Telemedicine with Urdu Support

Undertaken by
Muhammad Fayez Aziz
Muhammad Zaheer Aziz

Supervised by
Prof. Dr. Khalid Rashid

DEPARTMENT OF COMPUTER SCIENCE
INTERNATIONAL ISLAMIC UNIVERSITY
ISLAMABAD
2003

International Islamic University, Islamabad
Department of Computer Science

Acc. No. (TM4) T833

Dated: November 15, 2003

FINAL APPROVAL

It is certified that we have read the thesis submitted by **Mr. Mohammad Zaheer Aziz & Mr. Muhammad Fayez Aziz** and it is our judgment that this report is of sufficient standard to warrant its acceptance by

INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD
for the
MS DEGREE IN COMPUTER SCIENCE

COMMITTEE

1. **External Examiner**
   Dr. Muhammad Afzal,
   Director, Information Technology,
   ARID Agriculture University,
   Rawalpindi.

   [illegible]
   4/12/03

2. **Internal Examiner**
   Dr. Sikandar Hayat Khiyal,
   Head,
   Department of Computer Science,
   International Islamic University,
   Islamabad.

   [illegible]

3. **Supervisor**
   Prof. Dr. Khalid Rasheed,
   Dean,
   Faculty of Applied Sciences,
   International Islamic University,
   Islamabad.

   [illegible]
   15.11.03

A thesis submitted to

International Islamic University Islamabad

as a partial fulfillment of the requirements

for the award of the degree of

MS in Computer Science

Declaration

We hereby declare that this thesis and accompanying software, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that we have conducted this research, developed the experimental software, and written this thesis entirely on the basis of our efforts made under guidance of our supervisor. If any part of this thesis is proved to be copied out or found to be reported, we shall standby the consequences. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Muhammad Fayez Aziz
Regn. No: 07-CS/MS/02
Muhammad Zaheer Aziz
Regn. No: 53-CS/MS/02

Dedication

To Ummi and Abbu

Muhammad Fayez Aziz

To every one who reads this

Muhammad Zaheer Aziz

Acknowledgement

First of all we thank and praise Allah Subhanuhu-wa-Ta'ala for His special mercy on us and the energy He gave us by which we could accomplish this work. We acknowledge the efforts of the Prophet Muhammad Salla-Allah-u-Alaihe-wa-Sallam due to which we are Muslims.

We salute the patience and prayers of our parents that kept our spirits high and motivated us to keep up till the completion of this work.

We appreciate the sincere guidance of our supervisor Prof. Dr. Khalid Rashid. He took out time from his busy schedule to supervise this research and facilitated us with all the resources we needed.

We are also grateful to our colleagues and friends who helped in keeping our morale high with their appreciations and positive criticism.

Muhammad Fayez Aziz
Regn. No: 07-CS/MS/02
Muhammad Zaheer Aziz
Regn. No: 53-CS/MS/02

Abstract

This research is related to improvement in effectiveness, utility, and efficiency of telemedicine systems. In order to make telemedicine more effective, a structured doctor-patient consultancy session is proposed which can be transformed easily into web-based applications. This methodology will allow patients to get medical advice from doctors through internet while the doctors will be able to remotely investigate and diagnose the disease of the patient. For bringing the benefits of the web applications in general and telemedicine in particular to wider range of population of our country, tools and technologies are designed and presented for allowing user interaction in Urdu. The interaction in native language extends from static web pages to fully interactive dynamic web applications. These Urdu interaction tools create opportunity for non-English speaking public to get benefit from information highway. For the sake of improved performance with multilingual web applications, a character conversion mechanism is introduced to optimize server load and network traffic. The methodologies are tested by implementing in an experimental web application for Telemedicine in Urdu. The results show that the methodologies developed can work successfully not only in telemedicine but in other web applications as well.

Table of Contents

Chapter 1

Introduction

1. Introduction

Internet has influenced lives of every category of people. The big bang of information technology has changed life styles. But unknowingly there has been a severe cultural problem. English, as a language, and western culture has started dominating other languages and cultures with a great speed. Many countries for instance China, Japan, and Arab world have started efforts for providing all the information available on the Internet in their native languages. This effort is useful not only in preserving their own culture but it is also opening doors of information for people unfamiliar to English language. This project addresses two domains in one go, i.e., bringing Urdu in the scenario of Internet and developing a Tele-Medicine platform for Pakistan.

1.1 Urdu in Internet Applications

Literacy rate in our country is 47% [33] – 51.6% [34] which is extremely low considering this age of information and knowledge. Further, a small percentage out of the literate population is fluent or comfortable with English. The oceans of information and knowledge available through the information highway are of no use when the major part of population cannot understand it. Few efforts have been made by different developers to display news, information, and entertainment related material in Urdu in web pages but almost all of these web sites concentrate on displaying static web pages. Most of these websites are displaying bitmaps of Urdu text instead of character/string display. The main requirement for bringing Urdu in the main stream of Internet is to use Urdu in dynamic web applications with same ease and efficiency as that of English. The first aim of this project is to develop the tools that will be helpful in bringing Urdu on the Internet with its due glory. The research intends to identify the "missing links" that have prevented our national language from participating in the cyber world. Currently identified issues are as under:

- Non-existence of input controls that support Urdu.
- There are a few standard Urdu fonts that support Unicode.
- There are no online web-building systems for Urdu.
- Unicode supported web sites have to transfer '7' bytes per character for static page displays, which is a burden on server data transfer quota and network traffic.

1.2 Telemedicine

The communication power of Internet has enabled its user to select between options available all over the world. The geographical boundaries are loosing their significance. Similar to the area of E-Commerce, the emerging application of Telemedicine is now dominating the cyber world. A patient can get advice of his/her choice of doctor. Similarly there are a large number of areas on the globe where qualified and experienced medical practitioners are not available. By putting the Internet in between, it has become possible for people living in such remote areas to avail the medical facilities right inside their homes.

The second aim of the project is to provide groundwork for Tele-Medicine in Pakistan. Our country has a totally different life pattern and cultural habits as compared to western world. Hence there is a need to look for solutions to online health system according to our own environment.

1.3 Existing Systems

The project under discussion addresses two issues at the same time. First is the existence of Urdu in dynamic Internet applications and other is introduction of telemedicine in Pakistan. There are a number of web sites developed in Pakistan and other countries related to these topics. Let us discuss each category separately.

1.3.1 Urdu Web Sites

Urdu is a widely spoken language in the sub-continent and due to immigration on large scale to developed countries; it is also popular in other areas of the world. Software for Urdu has been a center of attention since early 1980's. Different developers have developed many utilities and applications. The main problem in these software applications is the diversity of standards. Every developer has concentrated on his/her own abilities and available tools. As a result, the products cannot communicate with each other in any way. Same problem persists in case of Urdu web sites. Some developers have contented themselves by making scanned images of Urdu text and add these to the web page. Some others who prefer character-based pages developed their own character encoding standards and tools to display information in Urdu. This lack of standards has kept Urdu web-application development behind.

A few popular Urdu web sites with a brief description are listed below:

Jang [1]: This web site is a news resource of Jang Group. It displays Urdu text using bitmaps. The headlines consume smaller image being short in size while details news occupy significantly large images.

Jasarat [2]: It a news web site of a daily newspaper 'Jasarat'. This web site uses two methods to display Urdu contents. First is the image method as that of Jang and other is by use of ActiveX controls.

Urdu ActiveX Controls [3]: Professional level Urdu ActiveX controls are produced and sold by Pakistan Data Management Services. The ActiveX control can be implanted in any Windows based application or web page. Urdu is displayed using vendor designed codes for alphabets and a Nastaliq font.

Urdu Point [4]: This web site presents a variety of contents in Urdu for example news, poetry, cards, jokes etc. The approach to display Urdu text is again with the help of scanned or computer generated images.

Urdu Email [5]: There are some web sites that allow users to type and send their emails in Urdu. The reader at the other end mostly has to use the same web site to view the message in Urdu. A typical example is mail facility in www.ApniUrdu.com

Urdu Poetry [6]: Urdu poetry is available on many web sites. Some display Urdu verses using images, some use ActiveX controls, and some use dual approach. In the dual approach smaller text like title is shown in bitmaps while the larger part like the whole poem is displayed using character based storage utilizing a vendor developed font. A poetry web site www.shairy.com is one of the examples for the third category.

1.3.2 Telemedicine Systems

There are two angles to view the existing systems of telemedicine. First is the international scenario and the other is situation of Pakistani telemedicine systems. The international viewpoint converges mostly to developed countries that have much educated population and high-end technology affordable by most of their citizens. Hence their telemedicine related systems exploit these advantages. On the other hand in Pakistan and other such developing countries literacy rate and poverty are major national problems so the systems developed (or being developed) had to keep these factors in consideration. A few of the popular web systems related to telemedicine are discussed below:

Doctor Search: There are a number of web sites available for developed countries that allow searching a doctor in a particular city. The doctor may also be searched by area of expertise or by name. A typical example of such web site is the search facility of American Medical Association [7].

Telemedicine [8]: There has been great progress in field of telemedicine internationally. There a associations, forums, research centers, and journal for telemedicine. www.telemed.org can be considered as an example and resource center for telemedicine activities in developed countries.

Telemedicine in Pakistan [9]: There has not been much of development in field of telemedicine in Pakistan. The only Pakistani telemedicine web site having some influence and information is www.telmedpak.com. It provides information about health care, allows asking questions from doctors, and many other health related features. There is some information available in Urdu but the display is done by using scanned images of Urdu text.

1.4 Objectives

The objective of the project under discussion are set keeping in view the needs of the users of the system being developed, and the drawbacks of the existing systems as discussed in section 1.3. The set of general objectives in developing this project is listed below:

- Develop a set of standard tools that would enable data entry using the same input controls that are used for English.
- Develop the technology to display Urdu language text on dynamic and static web sites on the same pattern as that of English.
- Develop a mechanism that would allow use of available and new standards of data interchange (code plates) and data entry (keyboards) dynamically.
- Develop a methodology that would make the data transmission of Urdu web applications as efficient as that of English.

- Develop a dynamic telemedicine web application using the above tools and technologies to prove the possibility of existence of Urdu on the Internet on equal grounds as English.
- Provide all facilities and features that doctors and patients require to establish an online medical consultation session, to communicate efficiently and effectively, and to conclude successfully.

1.5 Scope of Work

In order to achieve the objectives mentioned in section 1.4 in a limited time, a boundary has to be drawn for the scope of work to be done in the specified time. The enhancements and meeting of ever-growing demands of the system can be done in future versions of the project. The tasks to be completed as part of this project are listed below:

- **Standard Urdu Input Controls:** The main input controls that need to be Urdu-Enabled are edit box, text area, list box, check box, radio buttons, and submit buttons. The Urdu edit box and text area will be the standard HTML elements that will enable the user to type and edit Urdu text. This will be done at the browser layer, without requiring operating system's support enabled for Urdu. Other User Interface controls mostly involve display of Urdu text using character-based technology.

- **Multiple Keyboard Options:** Urdu controls will be fabricated to work with the user's choice of keyboard. Some people may prefer 'phonetic keyboard', others may use official standard. The controls will be adaptive to user's selection of keyboard.

- **Unicode supported Urdu Font:** A complete Urdu font supporting Unicode standard will be developed so that the text display in input controls and web pages works like character based data according to international standards.

- **Efficient Data Transfer method for Unicode:** An algorithm will be developed to transfer Urdu Unicode characters as single-byte instead of 7-bytes per character as presently done in case of Urdu in HTML. A mechanism will be developed to encode the 7-byte data of Unicode character into a single byte before transmission from server to client, and to convert it back to 7-byte HTML acceptable code before display in browser.

- **Multiple Code Plate options:** The data storage in databases, ***whois*** registries, and system level files is ultimately in ASCII format. In this project it is planned to allow user to make choice of basic code plate to encode the Urdu data. There are several standards available. The choice of code plate will make the data portable and transformable to desirable formats.

- **Online Tele-Medicine Platform:** Applying the results of the research and the developed tools, a platform for Tele-Medicine will be constructed. This platform will allow doctors to give consultancy to patients via Internet and patients will be able to search relevant doctors and medical facilities on the Internet. All facilities of data display, searching, and feeding will be available in both English and Urdu.

Chapter 2

Formulation of the Problem

2. Formulation of the Problem

In this chapter required output of the research will be formulated. The objectives will be transformed into concrete targets, each of which will lead to the development of a module of the system. The main objective is to develop an Urdu enabled telemedicine system. The requirements of this objective involve development of many modules and components that will work together to form a complete system. Each section of this chapter identifies some component or module of the system and discusses and analyses it in detail.

2.1 Display in Urdu

For the Urdu text display, the system under discussion considers the objective to use character-based display on the browser. This is a basic necessity for dynamic web applications in which information to be displayed is generated by a program or retrieved from a database engine. The first step is to select an internationally recognized standard for displaying non-English text. Unicode has emerged as the accepted standard for representation of characters of all languages of the world. It represents one character in a two-byte code that enables it to encode all character sets of languages used on earth [10]. Unicode is a standard for encoding the characters whereas a Unicode supported font is needed to display the characters in their natural shapes. A Unicode supported Urdu font will have to be developed in order to complete the life cycle of character-based Urdu display.

2.2 Input in Urdu

The most important set of components needed for Urdu-enabled web applications are the standard input controls for Urdu that will work in ordinary HTML documents as they do for English. The widely used input controls in HTML forms are edit box, text area, list box, check box, radio buttons, and submit/reset button. Each of these input controls has to be modified such that they become able to accept user input in Urdu along with English. Detailed requirement specification of these controls and other supporting gadgets is provided and discussed in subsequent sub-sections.

2.2.1 Edit Box

This is the most important input control that provides a basic mechanism eminent for other controls. This control provides facilities of text typing, editing, deleting and appending. It also involves display of data while being edited, as the user needs to see the text present in the control. When converting the edit box to accept Urdu text, the editing facilities available as for English are to be completely retained. The Urdu text in the control is able to be displayed in available Urdu fonts.

2.2.2 Text Area

This control is very similar to edit box as far as basic functionality is concerned. The main difference is the ability to handle larger text in multiple lines. This includes ability to wrap the text in which the word that does not completely fit in one line is shifted to the next line. The requirement for Urdu enablement would be same as that of edit box.

2.2.3 List Box

The List Box input control opens a list of options on mouse-click on its button. The options are text strings that can be selected (as single or multiple ones) by using mouse-clicks. For conversion into Urdu, this control needs display of the options list in Urdu. The rest of the mechanism will remain same as that of English control. This control returns the value or text of the selected item to the server, requiring encoding in later case.

2.2.4 Check Box and Radio Button

The check box works alone as well as in groups. The user enables the check box using mouse-click to select the option represented by it. In case of check boxes all of the available options may be selected, some may be selected and other left, or all may be left un-selected. On the other hand Radio Buttons work always in groups. It allows selecting at least and only one item out of the available options. The only language dependant part of Check Boxes and Radio Buttons is their caption. The conversion of these controls to Urdu will require writing the captions of these controls in Urdu.

2.2.5 Submit and Reset (other Ordinary) Buttons

These are push button controls. The Submit button sends all data of the form to the web server while the Reset button clears the values of all controls in the form. To convert these controls to Urdu, their caption has to be Urdu-enabled. Moreover the submit button has to be reprogrammed to do the necessary conversions of Urdu data present in the form elements so that the data transmission stays efficient and manageable at receiving end.

2.2.6 Multiple Keyboards Option

Different users prefer different keyboard layout for keypunching the text. There are more than one standard keyboard layouts available for Urdu, like phonetic keyboard, IBM standard keyboard, MQZ keyboard [11] etc. For feeding Urdu data in the edit box and text area controls, the keyboard should be selectable by the user at runtime rather than fixing only one keyboard. The user should also be able to see the keyboard map, on demand, in order to know the configuration of keys.

2.3 Data Storage and Code Plate

Databases and document management systems work on coding standards like ASCII and Unicode for storage of text. Each character is assigned a unique numeric code that becomes its identity. Standard information interchange code plate, like the ASCII, has recently been developed for Urdu. Such code plates play a vital role in allowing different applications to interchange their data. The application under discussion is being designed to work with Unicode standard but in order to make its data interchangeable with other Urdu applications features of exporting data in a required coding format will have to be provided. As the newly developed Urdu code plate is expected to go through changes before becoming a widely accepted standard so this system should be able to adapt any coding standard at runtime without needing to alter the software.

2.4 Client-Server Communication

Data transmission becomes heavy when a web page containing non-English contents is to be displayed on a browser using Unicode based data. For each Urdu character at least 7 bytes are transmitted from the web server to the browser. For example to display the Urdu character 'Chey' on the web browser the HTML code will contain the character sequence &#1640; in which the four digits form the Unicode number of the said Urdu character and the rest are requirements of HTML for displaying Unicode characters [13]. Individually, the isolated, initial, medial and final forms of Urdu characters have five digit Unicode number. For these characters the HTML code will be of 8 bytes. This mechanism bears many disadvantages. First disadvantage is regarding the data-upload quota of web servers. Most of the web sites are hosted on hired servers. The hired servers usually allocate a limited monthly data upload quota. With the above-mentioned drawback for Urdu data transfer the limited quota would exhaust a lot earlier than expected. The second disadvantage is burden on the network traffic. Urdu web sites would open taking much more time than the English ones. Similarly the server would be working to transfer more data for less information and efficiency will be affected. Third disadvantage or problem is regarding the database storage of Urdu data. It would not be wise to store the 7 or 8 byte data for Urdu characters in databases.

The above discussion leads to need of a *code conversion routine*. The routine should be able to convert the 7 or 8 byte Unicode data into a single byte representation in the files to be uploaded from client to server. When these stored files will be retrieved on client side through the browser the data would be converted back into 7 or 8 byte code as per requirement of the HTML browser. Similarly when the client would send the values of input forms to the server, the routine should perform the conversion from Unicode to one-byte representation.

2.5 Doctor's Signup & Login

For the basic requirement of Telemedicine system, the doctor needs to be online on Internet with the necessary information that would help patients to know his/her area of expertise. Hence a web-based signup form needs to be developed in which user friendly facilities should be provided to feed and save following information about the doctor:

- Doctor's Name
- Login ID & Password
- Country, City, and Street Address
- Area of specialization
- Education and Experience
- Time for online availability and fee charges

In order to facilitate the users (patients) who can understand Urdu only, the doctor will have to feed the information in both English and Urdu.

A login facility will also be needed using what the doctors could sign in to this web application after proving their login ID and correct password. A dynamic home page for each doctor will be the main part of this component in which the doctor will have different options including an option to update the above-mentioned information (not in scope as yet). For details of this component's functionality, consult Section 2.8 *Doctor-Patient Interaction.*

2.6 Patient's Signup & Login

The patient requires storing such information on the system that will help a doctor to identify the patient's problem. Apart from current symptoms, the social, economical, and professional background of the patient also plays an important role in diagnosis of the illness [14]. The information related to the patient belongs to three categories. First is the permanent information that would not change with time such as name, gender, and date of birth of patient. Second category is the information that can change with time but has long-term life. This category includes information like inherited diseases, illness or accidents suffered by patient, treatments given in prior life, and allergies etc. The information regarding these two categories needs a web-based form in the language easily understandable by patient (English or Urdu). The list of fields should be as under:

- Patient Name
- Login ID and Password
- Country, City, and Street Address
- Gender
- Date of Birth
- Profession
- Religion
- Existence of diseases like diabetes, blood pressure, and heart problems in family.
- Allergies
- History of previous illnesses and medications

The third category includes information about current illness of patient that will be provided at the time of getting or attending an appointment. Although this information has a very short life span but some part of it will be saved as medical history of the patient.

A sign-in system will also be needed that would lead to a dynamic home page specific to each patient comprising of different options related to the patients including an option to alter the information of second category (not in scope as yet).. For details of this component's functionality, consult Section 2.8 *Doctor-Patient Interaction.*

2.7 Service Provider's Signup & Login

According to the system under discussion service providers include medical clinics, hospitals, medical test laboratories, and medical stores. These service providers need facilities to upload information about their services and products available with them. The information will be available for customers / patients by searching the database on a given criteria. Information that must be provided by the service providers includes the following:

- Name of service provider
- Country, city, and street address
- Hints to reach the location
- Web site and email address
- Type of service provided (medical consultancy or supplies)
- List and details of products / services categories available

A sign-in system will also be needed that would lead to a dynamic home page specific to each service providers comprising mainly of an option to alter the above-mentioned information (not in scope as yet).

2.8 Doctor-Patient Interaction

The system must provide ample facilities to establish an online consultancy session between doctor and patient. This requires transforming the real life interaction procedure into a computer based interaction such that the effectiveness and quality of communication is maintained. In order to achieve this target the real life interaction scenario has to be carefully understood and then requirements for the online sessions may be established.

In real life when a patient needs services of a doctor he / she searches the appropriate medical practitioner related to their problem. Then usually an appointment is taken from the doctor. At the time of appointment (or time of visit without appointment) the patient has to wait in a queue while the doctor attends the patients who have come before this one. On reaching the turn in the queue, the doctor attends the patient in which the first part of conversation comprises of some basic questions regarding the patient and the illness suffered by the patient. With this the doctor also examines the patient to collect further information about the illness. In some cases the doctor suggests some tests to be conducted before reaching a conclusion about the disease. In the end the doctor writes a prescription comprising of medicines and instructions about their use.

In a computer-based session on Internet there will be obviously some natural constraints like emergency treatments (as in case of accidents and heart attacks etc) cannot be provided through Internet. Similarly the patient may not be able to convey the symptoms that can only be detected by a doctor via instruments (Thermometer, stethoscope, blood pressure machine etc). Electronic medical instruments have been developed that can be connected to a computer and the readings may be transmitted via the Internet [15]. Scope of the telemedicine consultancy broadens in presence of such instruments. This project is limited as yet to doctor-patient sessions in which the doctor can understand the problem with a structured question-answer session or by provision of results of medical tests such as blood sugar, RBC count, microscopy report etc.

The requirements of establishing an online consultancy session between doctor and patient can be specified in three portions. First one is according to doctors' perspective that is the doctor should be able to prepare the questions according to his / her area of specialization. Second part is according to the perspective of patients who should be able to easily answer the questions asked by the doctor. The last part is to make this question-answer session dynamic and flexible enough so that the doctor or patient may be able to communicate effectively and with ease. These three requirements are described in the following subsections:

2.8.1 Doctors to Design Questionnaires

After carefully observing the attending of patients by doctors and through the study of medical books [14] it becomes evident that it is possible to portrait the doctor's interrogation with patient utilizing a set of predefined questionnaire forms. There is a noticeable pattern of questions that doctors ask while opening an investigation with the patient. Although the questions may vary for different kinds of investigation sessions but taking a union of these sets leads to a relatively larger set of questions that a patient has to answer every time when consulting a doctor. These questions lead to specification of a particular category or group of diseases. After asking this set of questions the doctor starts queries about the current problem of patient. These questions are selected on basis of the answers given for the first set of questions and are targeted to diagnose the exact nature and intensity of the disease. These sets of questions have to be carefully designed by specialists because a specific sequence of questionnaires corresponds to pinpointing a specific disease.

It is obvious from above discussion that the job of designing the question sets should be handled by doctors rather than computer programmers. Hence the requirements of the system converge to providing a user-friendly questionnaire design facility for doctors. A doctor will be able to design many questionnaires, each aimed to diagnose a suspected disease. A patient will be presented questionnaires of basic and general nature at the time of getting appointment from a doctor. After reviewing the given answers to these queries, the doctor will be able to narrow down to the nature of illness and will send one or more specialized questionnaires to reach a conclusion. The questions should be either multiple-choice type or requiring just one value to be fed. This is needed due two reasons. Firstly the patient is not expected to type long answers as he / she may not be an expert computer user. Secondly long descriptions may become misleading, as the patient may not be able to write what the doctor needs to know.

While designing a questionnaire the doctor must have facilities to give a meaningful name to the questionnaire so that other doctors are also able to easily recognize it and use it for inquiring the patients. Hence following facilities must be available in the questionnaire design module:

- Give a name to questionnaire
- Purpose of questionnaire
- Add questions
- Select type of question: multiple choice or some value to be fed
- Add possible answers to be used as available choices

2.8.2 Patients to Get Appointments and Wait in Queue

The system should cover three steps before a patient is able to actually discuss a problem with the doctor. First step is to facilitate a patient to find a doctor related to the illness. Second is to get an appointment in which a time, in convenience to both doctor and patient, is selected for the meeting to occur. The last step before the meeting session starts is the 'real sense simulation' of waiting of a patient in queue for his/her appointment time or turn, while the doctor attends the patient(s) with prior appointments to this one. In the computer-based system, symptoms of the current illness may also be fed by the patient within the second step. Hence required facilities of this module can be listed as follows:

- Search doctor with a specific specialization in a particular country and city
- Display the availability time for online consultation for selected doctor
- Get appointment for a particular date and join the queue of patients when the time arrives
- Fill the answers of basic level questionnaire about symptoms felt
- Be able to feed results of tests which were advised by the doctor in previous visit
- Be able to see the queue of patients being attended by doctor while online and be acknowledged at the time of appointment tempting the patient to stay online until his / her turn arrives

2.8.3 Doctor-Patient Consultancy Session

This module of the system involves features and facilities for both kinds of clients. Communication between doctor and patient will need to be established in which they will exchange information in structured format through questionnaires. Doctors will be selecting and sending the already designed question sets and patients will be returning those back with filled answers. This Question-Answer session may continue for some period of time. Requirements of this module may be listed as under:

- Doctor should be able to see the online patients queue and select a patient to attend
- Doctor should be able to view all related data regarding the patient so that an initial approximation may be built about the disease and also the answers given to the basic questionnaire about current illness by the patient
- Doctor should be able to send specialized questionnaires, laboratory test advices, and prescription to the patient
- Patient should be able to interactively view and answer the latest questionnaire sent by doctor

- Patient should be able to receive advice of laboratory tests from doctor in shape of a form, which can be filled and returned after the tests are done.
- Patient should be able to view final prescription of, or to send any request/comment to, the doctor

2.9 General Search Facilities

In addition to facilities for searching doctors in the related specialization area, the system under discussion should also allow the users to find information about health related products and services. For example some patient may need to know where he/she could find a particular medicine in a specific city. Similarly some one may require the knowledge where he/she could find a particular medical service such as ultrasound or some lab test in a given city. The data collection about available services and products was handled in the service provider signup module discussed in section 2.7. In the search module users need following search facilities:

- Search availability of a particular medicine in a given city
- Find out from where a particular medical service may be availed in a given city
- Find out available doctors specialized in a particular area in a given city
- Find out location of a specific medical store, clinic, or hospital

Chapter 3

Proposed Methods

3. Proposed Methods

After formulation of problem, the areas of knowledge that need exploration in order to reach a solution have to be clearly identified. The study of the problem in the previous chapter reveals that many aspects of the system demand extensive study before actually conducting the task development. The project under discussion requires research for integrating three areas of knowledge, namely, clinical methods in medicine, multilingual user interfaces, and optimization of data transfer rate. Subsequent sections will illustrate the research work done in these areas.

3.1 Structuring Doctor – Patient Interaction

Clinical medicine is believed to be the most important part of medical science. It is an essential part of the curriculum for students of medicine. It is related to simple and systematic ways of case taking, brief but complete methods of examination of patient, and laboratory methods of investigation [14]. Doctors have to keep a well-defined sketch of symptoms and their reasons while examining a patient. The list of symptoms that are initially noted is given in figure 3.1. To elaborate a symptom fully and proceed towards diagnosis, the doctor must ask relevant questions from the patient about the complaints.

1. Fever
2. Pain
3. Dyspnoea
4. Cough
5. Vomiting
6. Diarrhea
7. Fits
8. Weakness
9. Headache
10. Mass

Figure 3.1 List of Symptoms

3.1.1 Clinical Questions

The questions used in the clinical consultancy either demand information regarding feelings of the patient or feeding results of laboratory tests. For the case of telemedicine, questions to be asked from the patient should be answered by either selection of a choice from a given list of options or providing an absolute value. Sample questions showing this nature are given in figure 3.2. This nature of questions will lead to many advantages. Firstly the patients will be confined to provide only that information that the doctor actually needs to know. Irrelevant details and stories will be avoided. Secondly all patients are not expected be expert computer users or even good typists. Hence selection of given choices will minimize requirement of data entry and also avoid errors due to mistyping. The doctor will also have a better opportunity to know and measure the complaint of the patient from discrete answers rather than textual explanations. The advantage with respect to telemedicine is that the information is easy to exchange between computers and efficient to transfer.

Q. What is the mode of onset of fever

(A) *Sudden* (B) *Gradual*

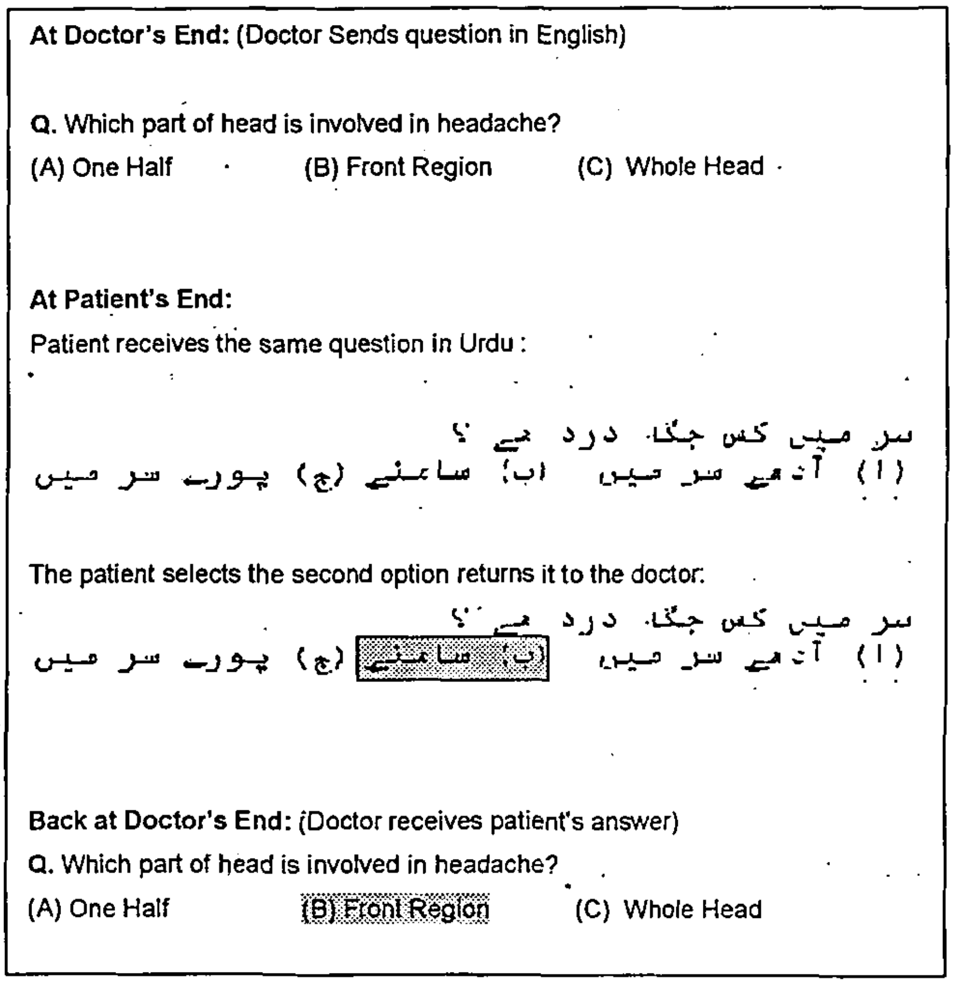
Q. What is the pattern of fever?

(A) Continuous (B) Remittent (C) Intermittent

Q. What is the degree of fever in Fahrenheit? ________

Figure 3.2 A Sample of Clinical Questions

Another advantage that can be gained from this approach is that a doctor may put the questions in one language, say English, while the patient could receive and answer them in another language such as Urdu. These answers could be shown to the doctor again in the first language. This can be done without going into any language translation because the patient is just selecting a given choice instead of feeding text. The captions of available choices can be shown to the doctor in one language and to the patient in the other. The choice will be highlighted when the patient selects it and the same choice will be shown highlighted to the doctor when the answers reach him/her. Hence the language barriers would fade away for doctors and patients residing in different regions of the globe. A sample exchange of a question between doctor and patient is shown in figure 3.3.



At Doctor's End: (Doctor Sends question in English)

Q. Which part of head is involved in headache?
(A) One Half (B) Front Region (C) Whole Head

At Patient's End:
Patient receives the same question in Urdu :

سر میں کس جگہ درد ہے ؟
(ا) آدھے سر میں (ب) سامنے (ج) پورے سر میں

The patient selects the second option returns it to the doctor:

سر میں کس جگہ درد ہے ؟
(ا) آدھے سر میں [✓ (ب) سامنے] (ج) پورے سر میں

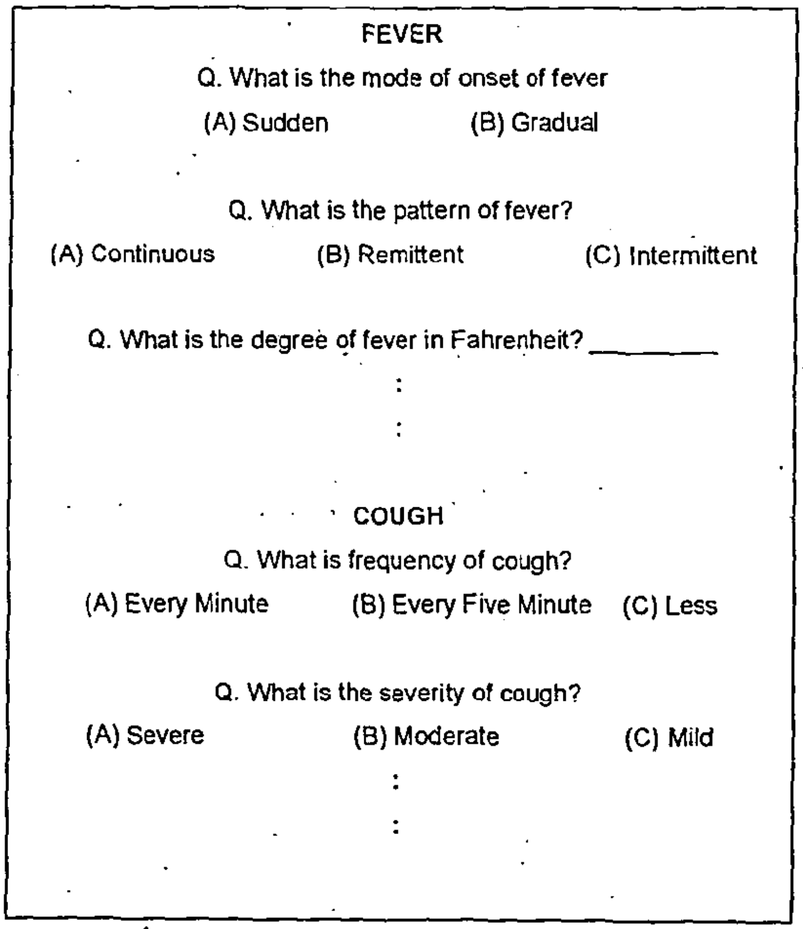
Back at Doctor's End: (Doctor receives patient's answer)
Q. Which part of head is involved in headache?
(A) One Half [✓ (B) Front Region] (C) Whole Head

Figure 3.3 Question-Answer Exchange in Different Languages

3.1.2 The Questionnaires

Generally investigation will be conducted using three levels questioning. In the first level a set of general questions asking about existence and nature of symptoms will be given to the patient. The inherent problem in case of telemedicine is the limitation in directly viewing the condition of patient and the clinical examination. Hence the initial set of questions to be asked from the patient may be large in order to cover all possibilities and provide the doctor enough information to proceed further in diagnosis. A sample questionnaire for this initial level, extracted from basic clinical medicine [14], is shown in figure 3.4.

In the second level there will be questionnaires comprising of queries to divert the route of investigation towards a specific group of diseases. Sets of questions will be prepared in advance each confirming the existence of disease belonging to a specific group. One or more of these question-sets will be selected to be asked from the patient based upon the answers given by him/her in the initial questionnaire. Figure 3.4 shows sample a question set extracted from clinical medicine that would be asked to find details about cough[15].



FEVER

Q. What is the mode of onset of fever

(A) Sudden (B) Gradual

Q. What is the pattern of fever?

(A) Continuous (B) Remittent (C) Intermittent

Q. What is the degree of fever in Fahrenheit? ________

:

:

COUGH

Q. What is frequency of cough?

(A) Every Minute (B) Every Five Minute (C) Less

Q. What is the severity of cough?

(A) Severe (B) Moderate (C) Mild

:

:

Figure 3.3 A Sample First Level Questionnaire

Q. Any special time in the day for occurrence of cough?
(A) All day (B) Morning (C) Noon (D) Evening
(E)Night

Q. Is cough dry or accompanied by expectoration?
(A) Dry (B) Little expectoration (C) Lot of expectoration

Q. Color of sputum ?
(A) Colorless (B) Greenish (C) Grayish (D) Reddish

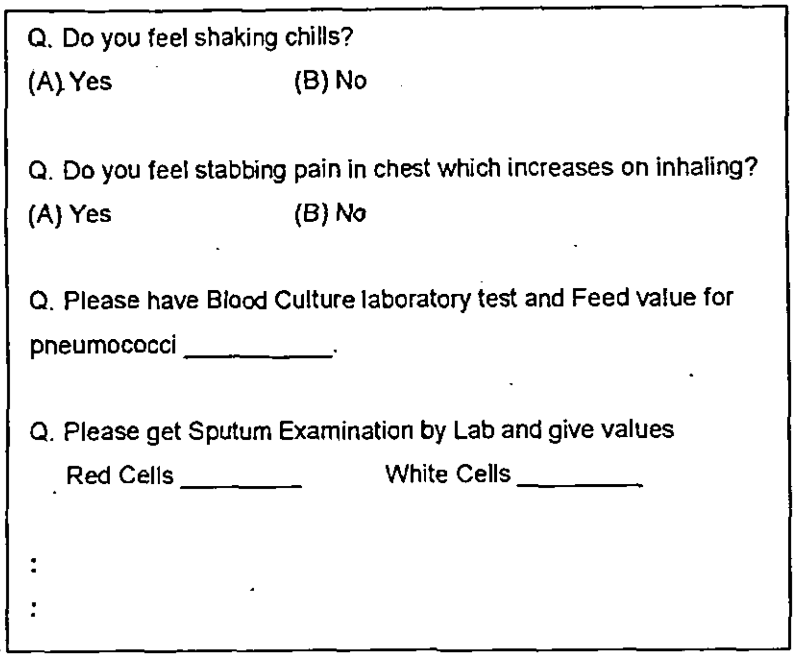
Q. If sputum has blood it color then describe color of blood
(A) Red (B) Dark Brown

:
:

Figure 3.4 A Sample Second Level Questionnaire

In the third level the questionnaires will be specialized to identify a particular disease from symptoms of patient. This may include specialized questions or an advice to have some laboratory test and feed the results. Figure 3.5 shows an example questionnaire when the doctor suspects Pneumonia after seeing answers of previous question sets and would like to confirm his/her diagnosis[16].

Preparation of these question sets for each level and placing them in suitable level of investigation can obviously be done by experienced doctors according to their professional requirements.


Q. Do you feel shaking chills?
(A) Yes (B) No

Q. Do you feel stabbing pain in chest which increases on inhaling?
(A) Yes (B) No

Q. Please have Blood Culture laboratory test and Feed value for pneumococci ________.

Q. Please get Sputum Examination by Lab and give values
Red Cells ________ White Cells ________

:
:

Figure 3.5 A Sample Third Level Questionnaire

3.1.3 Structured Session Between Doctor and Patient

Question sets prepared according to the mechanism discussed in the preceding subsections will form a basis to establish a remote medical consultancy session. Both doctor and patient will be communicating using computers connected to each other via some network. Hence a structured approach is proposed to make this counseling effective and useful. More than one questionnaire would be exchanged between the doctor and patient in order to enable the doctor to reach a correct diagnosis.

Initially a basic level questionnaire will be given to the patient to know the major symptoms. The next set of questions will be a branch route of investigation based upon answers given at this level. Similarly answers for this branch will decide the selection of route from the many branches available at this level. Hence we may consider the structured medical consultancy as a tree in which the first level basic questionnaire acts as the root and the session continues towards a particular leaf by selection of a specific branch at each level. The leaf node finally reached is the question set meant for confirmation of diagnosis for the disease. After this diagnosis the doctor will provide a prescription to the patient to complete the medical session.

The tree structure of reformed medical session is shown in figure 3.6. The root is associated with the first level questionnaire and in the next level there are a number of nodes available each assigned to a particular category of diseases. In the next level nodes representing specific disease emit from each category. The figure shows only three levels of nodes for simplicity while in actual diagnosis session the levels could be higher in number. Similarly at each level it is likely to happen that the doctor may have to jump back to previous level for exploring possibility of disease lying in some other branch. Dotted arrows in the figure show this provision of jumping back to previous level.

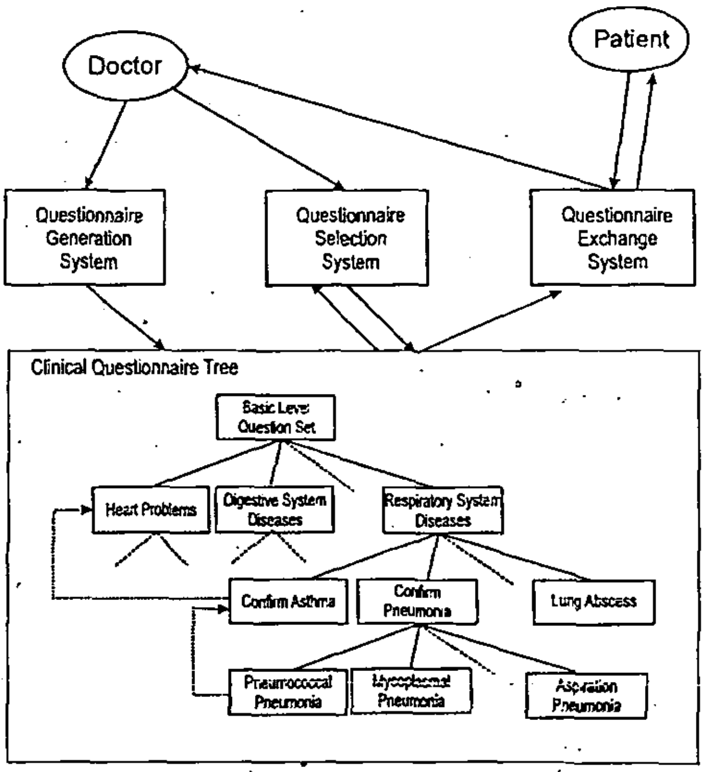

Patient
Doctor
Questionnaire Generation System
Questionnaire Selection System
Questionnaire Exchange System
Clinical Questionnaire Tree
Basic Level Question Set
Heart Problems
Digestive System Diseases
Respiratory System Diseases
Confirm Asthma
Confirm Pneumonia
Lung Abscess
Pneumococcal Pneumonia
Mycoplasmal Pneumonia
Aspiration Pneumonia

Figure 3.6 Architecture of Tree Structured Clinical Session

3.2 Building Urdu User Interfaces

Importance of local languages for computing and information technology is an established fact. In case of telemedicine, this importance has to be more emphasized due to the reason that a patient could belong to rural areas with insufficient level of education. Hence it is expected that a majority of users would not be fluent in English. Hence the computer interface for the system under discussion should essentially support Urdu. User interface involves two types of elements. First are the display elements and other are input controls. This section will discuss tools and technology available for the said purpose and the research work conducted to reach a conclusive solution for providing Urdu user interface. Telemedicine systems are usually implemented on internet; hence focus of research will be on web-based interfaces.

3.2.1 Displaying Urdu Text

Software development for Urdu has been a center of attention since early 1980's. Programmers have developed many utilities and applications. The main problem in these software applications is the diversity of standards. Every developer has concentrated on his/her own abilities and available tools. The result being the product cannot communicate with other products in any way. The same problem persisted in case of Urdu web sites. Some developers have contented themselves by making scanned images of the Urdu text and add the image to the web page. Some examples of such web sites are that of Urdustan (www.urdustan.com), Urdu Point (www.urdupoint.com), and Urdu Classic (www.urduclassic.com). In these sites the documents of Urdu are displayed in form of embedded bitmaps. Another website that can be placed in the same category is for English to Urdu dictionary (www.UrduWord.com) in which bitmaps of individual alphabets are used to form the Urdu word against the given English word.

The more appropriate mechanism for displaying Urdu on web pages is character based display. Most of the developers of such web sites have used their own set of rules and tools for display if information in Urdu. This lack of standards had kept the Urdu web-application development behind. Some web sites that lie in this category are that of Jang newspaper (www.jang-group.com/jang) and daily Jasarat (www.jasarat.com). These are news reporting web sites. They use some word processing software to type Urdu text and put it into web page as non-English characters. Urdu font is used to give these characters the shape of individual Urdu alphabets. These sites work using proprietary tools and technique of Pak Data Management Systems (www.pakdata.com), which is not a globally accepted standard. The user has to download their customized font in order to see the Urdu contents on the web page. These web sites do not have any input facilities in Urdu language.

In the third category of Urdu web sites, we put those that display Urdu as character based data using internationally accepted coding and font standards. Such web sites enjoy the feature of software platform independence. These web sites display Urdu text in standard multinational coding schemes such as Unicode (www.unicode.org). The Urdu contents will automatically be displayed on any computer that has support for Unicode. One example of such web site is that of Urdu service of BBC (www.bbc.co.uk). Another example of such system is the web building software to create Unicode based Urdu sites (Shehzad, 2002). Input for Urdu data is still missing in this category.

For the system under discussion, the third approach is adopted with a minor modification. The modified approach transmits text consisting of characters from both English and Urdu in a stream of single byte ASCII data. Hence network traffic for Urdu web content is minimized. This byte stream is converted into Unicode stream at client side. This Unicode data is displayed on the browser with help of standardized Unicode supported font [24]. This architecture is shown in figure 3.7.

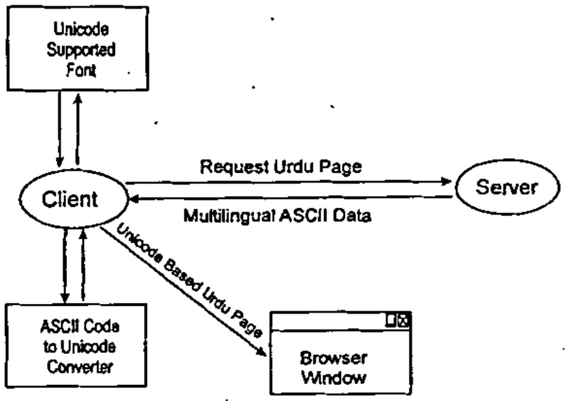


Figure 3.7 Architecture of Urdu Display System

3.2.2 Input in Urdu Language

The most important feature required to build a dynamic web application is the ability to input data from user. The web application then generates a web page according to the user input. Unfortunately such feature has been almost non-existent for Urdu web sites. There have been some local attempts at academic level but no professional products have been found so far. The web sites found so far use ActiveX controls to input Urdu data. Urdu database with Oracle and ASP (Irfan, 2001) is one of the examples. The controls of this system can accept input of Urdu but allow limited editing facility. Similarly a set of Urdu ActiveX controls was developed for input in Urdu in Department of Computer Science, International Islamic University, Islamabad (Ghulam Qadir, 2001). Such controls do not get popular because of their large size, requirement of permission to execute from user, and little complex to deal with at programmer level.

HTML Input Controls

In order to meet the requirements of the problem. research was carried out for development of standard input controls for Urdu that would work in ordinary HTML documents as they do for English. The widely used input controls in HTML forms are edit-box, text area, list box, check box, radio buttons, and submit/reset button. Each of these input controls has to be modified such that they become able to accept user input in Urdu in addition to English.

The input of check box, radio button, and submit/reset button does not have any language dependant processing involved as the only part that has a text is their caption. The input values are either integer or boolean numbers[23]. Hence the conversion process only involves displaying the caption in Urdu. In case of list box, all options of list have to be displayed in Urdu. The input values are either the index of selected item or its text. Hence the Urdu text will have to be managed when sending data back to the server. The most complex edit controls that require much work in conversion to Urdu are edit box and text area. In these controls user needs facility to enter and edit text while viewing what is being entered. Hence input and display are involved both at the same time. Another associated problem is the keyboard, i.e., which key should be assigned to which letter of Urdu. Similarly another issue is requirement of feeding English and Urdu mixed in the same field.

Keyboard Options

An important issue that needs to be resolved before making an input control to accept Urdu language is the association of keyboard keys to letters of the language. There are many versions of keyboard available for Urdu and each of them claims to be the standard in some aspect. The commonly known keyboards include IBM, MQZ, and Phonetic [11]. IBM standard has been in practice since many years while the MQZ standard has recently been developed for improving typing speed. On the other hand a general user feels more comfortable with phonetic keyboard in which the Urdu letter is associated with the closely sounding English letter.

In order to accommodate user's choice, it is decided to make the keyboard dynamically selectable. Associations between keys and the related letters will be saved in separate files. The file for the selected keyboard will be invoked when a user opts for it and that keyboard will be active right from the next key-press. Role of this keyboard file in the input system is shown in figure 3.8.

Role of Code plate

Code plates are universally accepted standards to represent, save, and interchange data. Exchange of data between diverse applications becomes possible when all developers save their data according to a common standard. The widely accepted standard for English is ASCII, whereas Unicode is emerging as a multilingual standard. A standard code plate has been developed for Urdu as well but it is subject to modifications for some time before it comes into final shape.

To resolve the issue of varying coding standards for Urdu the best option from available ones was to allow dynamic encoding scheme. It was done by keeping the code plates separate from main system instead of hard coding it. The association of characters and their codes will be kept in independent files that could be associated to the main program one at a time. This mechanism will allow accommodating alterations in the encoding schemes and also any new emerging standards. Import and export of data from one standard to the other will also be possible. Role of this dynamic code plate loader can be seen in figure 3.8.

Conversion Methodology

As discussed earlier conversion of two main HTML input controls, i.e. Edit Box and Text Area, is the major task of research. Other controls are dependant upon displaying mechanism that was discussed in the previous section. Edit box and Text area primarily accept ASCII codes that are fed into them on each keystroke from the keyboard. In Unicode supported web browsers, these controls are capable of storing Unicode characters as well. Shape of the inserted Unicode characters is formulated by using the default Unicode font installed in the computer. Hence our research now narrows down to devise a mechanism that would insert a Unicode of Urdu range when the users presses a key being in Urdu mode. Pressing of same key should feed the related Unicode of English range while in English Mode.

Above discussion leads to the fact that key and mouse button events will have to be trapped and reprogrammed for the edit control being used. The HTML 4.0 standard mandates a set of intrinsic events and these are supported to a lesser or greater degree in version 4.0 of both Netscape and Microsoft browsers [25]. Internet Explorer 4.0x features a property that indicates what mouse button was pressed (for a mouse event), and another property that indicates what key was pressed (for a keyboard event). In Navigator 4.0x, one propery (which) serves both cases [26]. Here the event of key press will be handled to reprogram these two edit controls. Other internet applications have been developed using key handling like the Doc Dialer which invokes different tasks on press of different keys [27]. Based on the study of available literature, the architecture of the system is designed as shown in figure 3.8. The system has to be configured by selecting a language and a keyboard. Now on pressing of a key inside the input control will invoke the system to use the keyboard manager. The keyboard manager will evaluate the key and replace it with an appropriate Unicode value using a predefined keyboard file. This code causes to display the key in the selected language (such as Urdu). In order to send this data to the server, this Unicode has to be converted into a 1-byte code according to the standard code plate for the language. When the value of the control consists of mixed text if English and Urdu, the system demarcates the languages by a special code at beginning of language toggle. This conversion back to single byte code is necessary because internal processing of database engines, domain registries, and other servers are based on ASCII style data coding. Detailed algorithm of this architecture will be discussed in chapters of design and implementation.

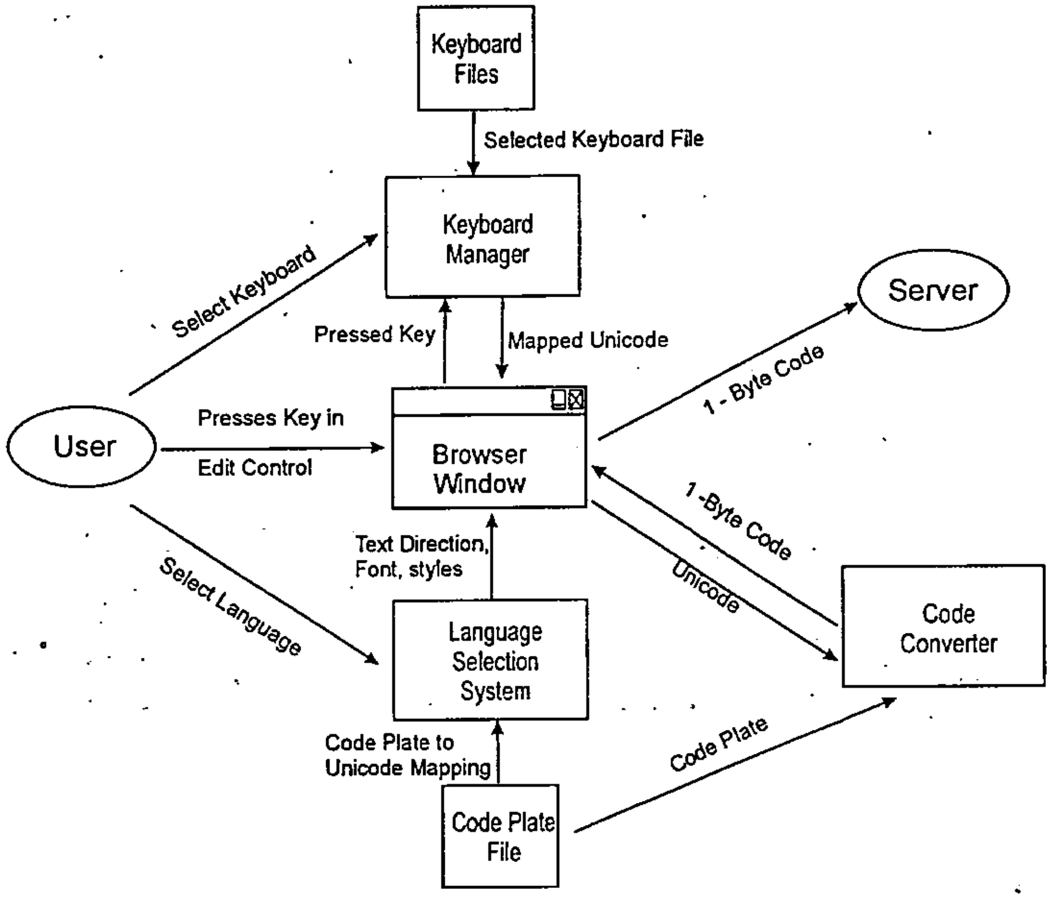


Figure 3.8 Architecture of Input Control Conversion

3.3 Optimizing Data Transfer Rate for Urdu Web Pages

Emergence of Unicode [10] for multilingual computing has made it quite easy to manipulate non-English languages in computer programs. Internet pages in native languages have become feasible to develop and view due this international encoding standard. The Unicode Standard encodes the basic characters of the Arabic alphabet, as well as a multitude of compatibility characters and presentation forms. The repertoire of Arabic characters in the Unicode Standard is sufficient for the representation of the three major languages using Arabic script (Arabic, Persian, Urdu), in addition to many other languages such as Sindhi, Kurdish, Jawi, Baluchi, and Pashto, among others [28].

The simple concept of separating user interface from program code seems to have cast aside in the typical web application. Mixing program code and user interface is the defacto standard for HTML and it is not uncommon to see a Java Servlet writing HTML code, which in turns contains Javascript. Web applications may also have the added complexity of trying to present information to multiple clients in multiple locales simultaneously [29]. This problem directly relates to using Unicode in multilingual web interfaces. The browser may be supporting Unicode standard and displaying every international character properly while the data base server could be supporting only one-byte text strings. Similarly search engines, 'whois' registries, and DNS servers may not be compatible with Unicode.

Another major problem with Unicode based web pages is its data size. At least 7 bytes are transmitted from the web server to the browser for each Urdu character. For example, in order to display the Urdu character 'Chey' on the web browser the HTML code will contain the character sequence "&#1640;". In this code the four digits 1640 are the Unicode number of the said Urdu character [13] while the first two characters and the last semicolon are requirements of HTML for displaying Unicode characters [23].

This research is focused on analyzing consequences of existing approaches of saving and transmitting multilingual data on internet and propose an appropriate methodology to solve the above mentioned problems.

3.3.1 Survey of Existing Approaches

A survey was conducted to collect data about storage space utilization and use of communication bandwidth by Urdu characters. Three different approaches were found for transferring Urdu text form server to client. First is the image approach [1 & 2], second in the static Unicode approach, and third is approach of using ActiveX Controls.

In the first approach, bitmaps of Urdu text are created and included in the web page as images. In this approach the amount of data transferred between the server and client includes the image itself with the few extra bytes of HTML code that embeds the image in the web page. The data size for the image with respect to the amount of text is so large that we can simply ignore the extra bytes needed for HTML tag for the image. The tag contains name of the image file which can vary unpredictably hence it is not possible to formulate a mathematical model for the data size.

In the second approach of embedding Urdu text as Unicode, each character is inserted using &# as prefix and ; as suffix around the Unicode number written as four byte text. Hence each character requires minimum of 7 bytes. Hence if there are N characters to be transferred then the number of bytes transmitted from the server to the client will be 7 x N. Similar approach has been observed in German and French web pages where they display their non-English characters using 5 – 6 bytes long codes [31 & 32].

The third approach uses ActiveX controls to show Urdu text on the browser [3]. The HTML code contains an 'object' tag for each chunk of text with the text string passed to the object as parameter. Each Urdu character is encoded in a two byte code (not according to Unicode standard). According to the survey, each 'object' tag requires 135 bytes and the text of N characters will need additional 2 x N bytes. Hence for a text of N characters, the number of bytes stored and transmitted by the server will be 135 + 2 x N.

A variety of Urdu web sites belonging to the categories mentioned above were visited during the survey and data was collected about the relationship between amount of text and space utilized for that text. Obviously the same amount of data ill be transmitted between the web server and client. Table 3.1 shows some sample values collected for the three existing approaches. It is evident that space and network bandwidth is over utilized and un-optimized in all of these methodologies.

Sr. No	Number of Urdu Characters (N)	Number of bytes stored / transferred		
		Image Transfer Approach	Unicode Transfer Approach (7 x N)	ActiveX Control Approach (135 + 2 x N)
1	22	527	154	179
2	35	681	245	205
3	43	748	301	221
4	64	2048	448	263
5	113	3072	791	361

Table 3.1 Relationship Between Text and Data Size for Existing Approaches

3.3.2 Drawbacks of Existing Methods

The first and main disadvantage of the existing methods for storing and transferring multilingual text is apparent from the analysis of survey results given in the previous section. These schemes are wasting storage space and causing an extra burden on network traffic. The image transfer method is the worst case in the available techniques. Significance of this problem magnifies for case of internet bandwidth available for small cities and rural areas of countries like Pakistan. It can be easily imagined that need for web pages in native languages is more intense in rural areas as more of the population is not conversant in English.

Apart from the apparent disadvantages there are some hidden drawbacks that add to severity of the issue of extra space utilization. Delivering heavier web pages not only burdens the communication media but the web and / or data server as well. It needs more time to access its devices and convey it to the network. This load amplifies when a large number of clients are accessing the same server. Ultimately the user gets more sluggish response.

Another devastating effect of over use of data space appears when the servers have a limited data transfer quota. Most of the hired internet servers have a restricted quota of bytes that they can transmit through the communication channel per day or per month. These servers stop serving any requests after exceeding the limit of allowed quota. For English language pages if a server could serve U users, the same machine would serve U / 7 users with Urdu pages using static Unicode methodology. The number of users would be significantly less for the other two approaches.

The client of such web pages has to pay a extra cost too. When a page in native language contains larger number of bytes, it would need more internet hours to download them. Again keeping in view the bandwidth for rural areas it can be easily established that the user will be paying lot of extra cost for accessing localized pages in shape of internet access charges and electricity bills for running the computer.

3.3.3 Proposed Methodology

At the time of when this research is being carried out, it is difficult to design and implement a multilingual system that would work on computers spread all over the world with same performance. As said in [30] "Due to the nascent nature of many Open Source projects their developers frequently do not have a strong internationalization background or experience. This leads to a certain evolutionary approach that is repeated over many projects. The end point of this evolution is usually a strongly internationalized application with a good Unicode foundation but the path can be sometimes very rocky." Hence the proposed methodology starts with separating the front end or user interface from processing part of the system. It is decided to use Unicode only for the browser for displaying the international character set in a suitable font. This decision will be valid until all web servers and data base engines not only recognize Unicode but apply optimization algorithms to maintain best possible performance level with the international encoding scheme.

The second difference in the proposed approach is to use single-byte code for transfer of data between servers and clients. Similarly multilingual data will be stored in database files as single-byte format. One major problem faced in this methodology is the existence of characters belonging to different languages in the same string. In order to deal with mixed language strings we have adopted to insert short delimiter codes when the language switches from one to another. In order to keep the data exchangeable with other application the standard code plate of each language will be used to represent the characters in 1 – byte codes. For example the data of English language will be in ASCII while the data of Urdu language will be in standard Urdu code plate. Figure 3.9 shows a sample multilingual string and its equivalent string encoded according to the proposed scheme. The English (or ASCII) characters are displayed here for the Urdu part because the shown characters have the ASCII value equivalent to that of the Urdu characters present in the string respectively.

Multilingual String : آپ کا Telegram ملا
Encoded in 1- Byte String : ~u/ٰ(K4(~eTelegram~u(ML4
~u = delimiter code to start Urdu ~e = delimiter code to start English

Figure 3.9 : Sample of multilingual character encoding

Now the remaining tasks include parsing of the multilingual strings and conversion algorithms to transform these strings for client and server. Architecture of this mechanism is shown in figure 3.10.

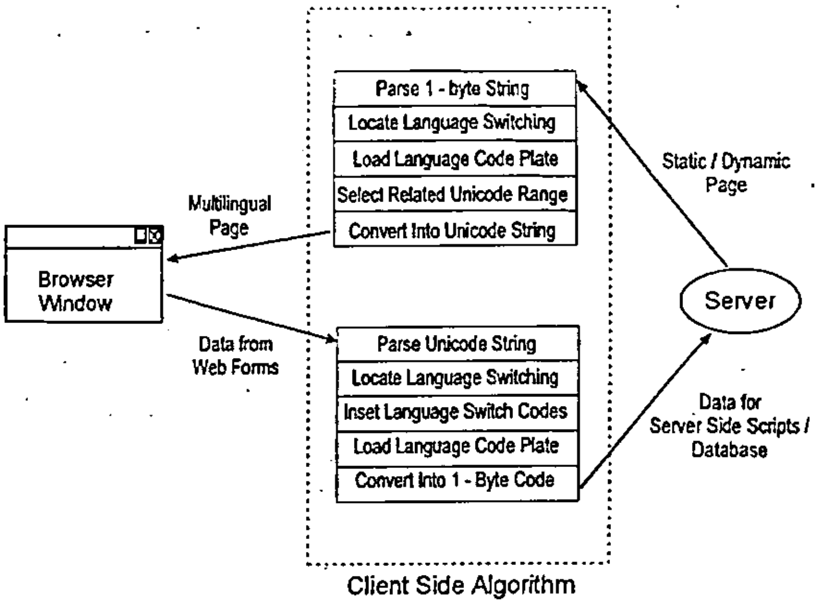

Parse 1 - byte String
Locate Language Switching
Load Language Code Plate
Select Related Unicode Range
Convert Into Unicode String
Static / Dynamic Page
Multilingual Page
Browser Window
Server
Data from Web Forms
Parse Unicode String
Locate Language Switching
Inset Language Switch Codes
Load Language Code Plate
Convert Into 1 - Byte Code
Data for Server Side Scripts / Database
Client Side Algorithm

Figure 3.10 : Architecture of Conversion Algorithm

The algorithm has to be implemented at the client side as it is the point where Unicode is necessarily needed for display of multilingual data. The web server will have the multilingual data in the said single-byte encoding scheme or producing the same using server side script. The client side algorithm will parse the input and locate the language switching codes. For each such code the algorithm will activate the code plate of the related language. Each character will be converted into relevant Unicode until encounter of end of string or next language switch code. On the reverse path, the Unicode data entered into the web forms of will be converted into single-byte strings before sending it to the web server, which in turn may be communicating with a database server and / or producing dynamic page in response to the request.

This scheme can easily be implemented in client side scripts for all famous web-browsers. For testing purpose a sample application was created using Java Script on Internet Explorer. The application executed successfully and worked exactly according to expectations and significantly decreased the network traffic between client and server.

3.3.4 Performance Analysis

In this section we evaluate the proposed methodology by comparing it with the existing techniques. Table 3.10 presents the complete comparison of available techniques and the propose method for sample strings of different lengths. Number of bytes that move between client and server for existing methodologies are copied from table 3.9 and bytes needed for the same set of data using the proposed method is inserted. The proposed character conversion mechanism uses 2 + N bytes for a string of N characters where the two bytes are for the language switching code. This number would increase if the same string contains more language switches. The values in the table are for strings that have three language switches. Such situation is likely to happen when a string of English is inserted in between Urdu text. Hence the sample data contains 6 extra bytes apart from actual data bytes. The comparison of storage and transmission space utilization is given in graphical form in figure 3.11.

Sr. No	Number of Urdu Characters (N)	Number of bytes stored / transferred			
		Image Transfer Approach	Unicode Transfer Approach (7 x N)	ActiveX Control Approach (135 + 2 x N)	Character Conversion Approach (6 + N)
1	22	527	154	179	28
2	35	681	245	205	41
3	43	748	301	221	49
4	64	2048	448	263	70
5	113	3072	791	361	119

Table 3.2 Relationship Between Text and Data Size for Existing Approaches

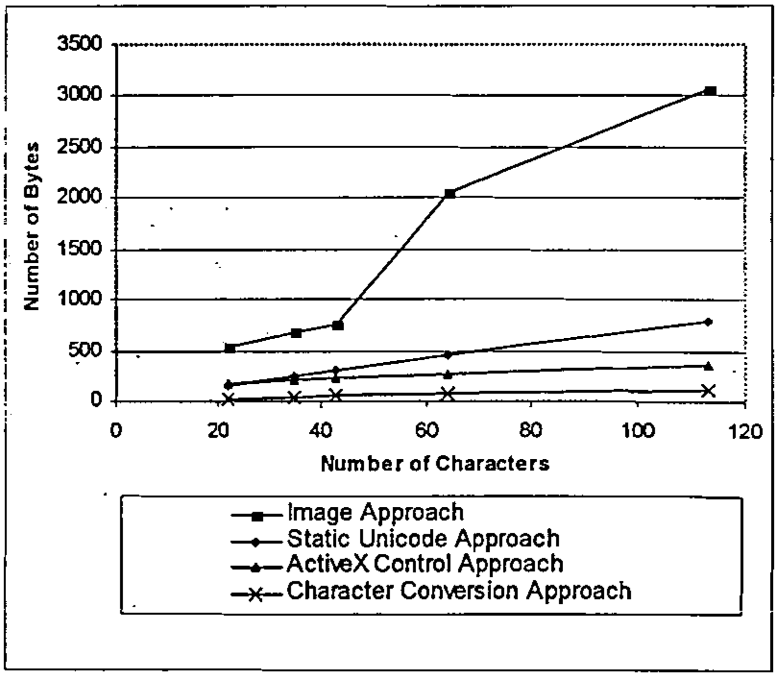


Figure 3.11 : Comparison of Performance

Chapter 4

Requirements Analysis

4. Requirements Analysis

The problem formulated in the second chapter will be technically analyzed in this chapter in order to move one step forward towards understanding and design of the system. Object-oriented analysis and design approach is adopted to carry out the requirement analysis phase. UML will be used as the tool for documentation purpose. UML formally includes the notion of use cases and use case diagrams [16]. UML utilizes the partial understanding of the system requirements, expressed in requirement specification document. In the following sections, complete analysis of the system will be presented in terms of actors, uses cases, and use case diagram [17] and also the conceptual model (ERD) for database organization [18].

4.1 Actors

Entities external to the system that stimulate the system with input events are included in actors list. Each actor has a specific role in the system. Actors identified for the system under discussion are:

- Web Browser
- Web Server
- Patient
- Doctor
- Service Provider (e.g. Clinic / Hospital or Medical Store Representative)
- General User
- System Administrator

Each actor can initiate many processes in the system. This is also called the role of actor in the system. The list of roles or performed functions for each actor is listed below:

Web Browser	Receive Unicode data in one-byte codes Display data in English and Urdu using 7 byte Unicode data Display Urdu input in HTML form elements in Urdu font Accept Urdu and English input according to selected keyboard

	Send Urdu and English data to the web server after code conversion / encoding Facilitate user interface by dynamically displaying links and other user interface elements Store user preferences of language, font, keyboard and scroll location
Web Server	Compose and send dynamically created web pages in both Urdu and English Accept Urdu and English one-byte converted/encoded data from the web browser Save encoded Urdu and English data into Database Open, use, manage and close database connections Support and help utilizing Server-Script language constructs and elements
Patient	Select language (English / Urdu) and font Choose preferred keyboard layout Fill out signup form in selected language Login Edit signup information (not in scope as yet) Search doctor Get appointment from selected doctor Continue in a follow-up appointment Fill out initial symptoms form for current illness Join patient queue at time of appointment Fill out real-time questionnaires and medical test results forms Logout
Doctor	Fill out signup form in both English and Urdu Login Edit signup information (not in scope as yet) Design Medical Questionnaires View Patient Queue as on today's date Select Patient to attend View initial information provided by patient Select one or more questionnaire and send to patient View answers to returned questionnaires Write prescription Logout
Service Provider	Fill out signup form Select service type, category and products (e.g. medical service or medicines available in the clinic / hospital or store) from lists to enable future search Add services to list if not already listed Login Edit signup and services information (not in scope as yet) Logout

General User	Search information about a hospital/clinic Search availability of a specific service Search availability of a specific medicine Search doctor with a specific specialization Find location of a known service provider
System Administrator	(not in scope as yet) Login View list of newly added items in database Edit / delete required database entry from list Modify / Delete any user record Logout

4.2 Use Cases

Use cases may be expressed in different degrees of detail, and commitment to design decisions. It means each use case may be written in different formats with different levels of detail, with respect to elaboration or technicality. There are two basic levels of use cases' format: *high level* and *expanded*. A high-level use case describes the process very briefly, only giving surface level details. On the other hand an expanded use case describes a process in more elaborately. The expanded format includes a section called 'Typical Course of Action' in which step-by-step events are described [16]. Furthermore, expanded use cases can be *essential* or *real* based on the technical level of expression while describing the story in the use case.

In the following subsections, use cases of the system under discussion are given, in expanded format. All of these use cases are of the type primary and essential as they discuss major common processes (primary, unlike secondary use case) and are free from technological and implementation details (essential, unlike real).

4.2.1 Display Urdu Text

Use Case:	Display Urdu Text
Actors:	Web Server (initiator), Web Browser, and General User
Purpose:	Display Urdu text in an appropriate Urdu Unicode supported font.
Overview:	The Web Server sends 1-byte codes for Urdu and English characters. This use case decodes it to allow multilingual character display using an appropriate font. It targets web pages and other than 'typed' form elements like selection lists, radio button captions etc.

	This Use Case is *implicitly* <<*included*>> in all other use cases (all involve Urdu Text display).
Type:	Primary and Essential

Typical Course of Events

Actor Action	**System Response**
1. This use case begins when the Web Server sends one-byte encoded data to the Web Browser, after initially reading from database or uploaded files.	
	2. Retrieves tactfully the English-Urdu ASCII encoded data from the database and system files, then composes and structures a dynamic web page containing required user interface elements.
3. The *Browser* receives the data and runs the client-side script over it.	
	4. Converts each one-byte data string into Urdu Unicode text using the Code Plate which consists of Unicode values' map, and echoes it to the Browser. (This step can be expressed as an <<included>> *real* use case).
5. The Browser displays the Urdu text using a Unicode supported font, to the General User.	

Alternative Courses

- Line 4: One-byte character not found in the code plate. Abort the operation.

4.2.2 Input Urdu Text

Use Case:	Input Urdu Text
Actors:	General User (initiator) and Web Browser
Purpose:	Allow input of Urdu text in standard HTML form 'typed' elements.
Overview:	Edit box and text area replace input characters to non-English Unicode characters pertaining to User's keyboard selection. **This Use Case is *implicitly* <<*included*>> in all other use cases (all involve User input fields).**
Type:	Primary and Essential

Typical Course of Events

Actor Action	System Response
1. This use case begins when General User types in an editable HTML form element (edit box or text area).	
	2. Converts input characters into non-English Unicode character based on selected keyboard layout and Urdu-English typing choice. 'Keyboard' is a Unicode values' pattern in a keyboard format.
3. The Web Browser displays the characters using Unicode supported font.	

Alternative Courses

- Line 2: Corresponding key match in the keyboard is null or absent. Return the character as it is.

4.2.3 Send Forms' Data

Use Case:	Send Forms' Data
Actors:	General User (initiator), Web Browser, and Web Server
Purpose:	Send English-Urdu data in Browser's form to the Server after code conversion / encoding.
Overview:	The Web Browser sends its form's data to the Web Server on submission by the General User. Before exactly submitting, the English-Urdu Unicode data is converted or encoded to one-byte code while keeping 'pure' English field values as it is. **This Use Case is *implicitly* <<*included*>> in all other use cases (all involve form data to be submitted).**
Type:	Primary and Essential

Typical Course of Events

Actor Action	System Response
1. This use case begins when the General User has filled the form on the Web Browser and selects a specific list's value, hits enter or presses a submit button.	
2. The Browser acknowledges the event and starts over the conversion operation.	
	3. Scans through all the form fields retrieving each one's value tactfully, while simply passing on

	the 'pure' English values.
	4. Converts each English-Urdu Unicode data string, character by character, into a single-byte encoded ASCII data string, utilizing an efficient Code Plate search mechanism in the process.
	5. Stores the ASCII encoded values separately, ready to be submitted. (Steps 3-5 can be expressed as an <<included>> *real* use case).
6. The Browser accepts a target location and submits the stored data to the Web Server.	7. Processes the data as required.
1. The Web Server notices the target location: a. If contemporary, stores the data in database. b. If same as was at the time of submission, initiate *Receive Forms' Data.*	

Alternative Courses
- Line 2: Any one form field value is empty or unselected. Prompt a message to the User and abort submission.
- Line 3: Form field's type unmatched. Return null or a blank value.

Related Use Cases
- Includes *Receive Forms' Data.*

4.2.4 Receive Forms' Data

Use Case:	Receive Forms' Data
Actors:	Web Server (initiator) and Web Browser
Purpose:	Receive back Browser's form data in encoded one-byte form, from the Server when target location is the same.
Overview:	The Web Server sends the form's processed data back to the Web Browser, the target location being the same as at the time of submitting. The English-Urdu ASCII encoded data is converted back / decoded to Unicode data and then these and the 'pure' English data values are assigned to their respective form fields.
Type:	Primary and Essential

Typical Course of Events

Actor Action	System Response
1. This use case begins when the target location is the same as it was at the time of submission and the Web Server sends back the form's data to the Web Browser after its processing.	
	2. Receives the form's data as a form-name sequenced structure.
3. The Browser starts the conversion back operation over the received data.	
	4. Retrieves each of the form field's received English-Urdu encoded ASCII data from the form-name sequenced structure, by iterating through the form field names correspondingly.
	5. Decodes each data string to a Unicode data value, partially <<using>> *Display Urdu Text: Step4 real* use case with an expectedly included difference of 'returning' the converted back value, while keeping the 'pure' English values intact.
	6. Passes through all of the Browser's form fields and assigns the Unicode values to their respective form fields tactfully.

Alternative Courses

- Line 4: The retrieved value is null or not defined on first loading. Skip the remaining steps for it.

4.2.5 Signup

Use Case:	Signup
Actors:	General User (initiator)
Purpose:	Gather necessary information for future use.
Overview:	The General User selects a User type and follows a signup procedure feeding required information, to be utilized when the User logs-in in future.
Type:	Primary and Essential

Typical Course of Events

Actor Action	System Response
1. This use case begins when General User chooses a User type and clicks on the signup link.	2. Examines User type and opens respective signup page: a. If signing up as a Doctor, initiate *Signup as Doctor.* b. If signing up as a Patient, initiate *Signup as Patient.* c. If signing up as a Service Provider, initiate *Signup as Service Provider.* 3. Stores the submitted data temporarily while checking database for a match with the User's chosen ID. 4. Feeds User's data to database after doing necessary processing.

Alternative Courses
- Line 3: The chosen User ID already exists. Inform and provide for the User with a form to provide another ID.

Related Use Cases
- Includes *Signup as Doctor.*
- Includes *Signup as Patient.*
- Includes *Signup as Service Provider.*

4.2.6 Signup as Doctor

Use Case:	Signup as Doctor
Actors:	Doctor
Purpose:	Gather necessary information of Doctor in English as well as in Urdu.
Overview:	The Doctor requests a doctor's signup form by choosing user-type as Doctor and is provided with a comprehensive one, in which he/she can feed information pertaining to a Doctor like name, user ID, password, city, country, area of specialization, education, experience, time of online availability, avg. time for a patient and fee charges etc., in both English and Urdu to cater users/patients of both the languages.
Type:	Primary and Essential

Typical Course of Events

Actor Action	System Response
1. This use case begins when a Doctor requests a doctor's signup form choosing user-type a Doctor and clicking on signup link.	
	2. Displays the form containing input elements for required information both in Urdu and English, loading necessary lists' data dynamically.
3. Doctor feeds the needed information, adding missing or required data into the dynamic lists.	
	4. The data is submitted to be saved in database.

Alternative Courses

- Line 3: Added element is empty or already exists. Prompt to Doctor and discontinue the operation.
- Line 4: Password, Time or 'numeric' fields fail to comply. Inform the Doctor and abort submitting.

4.2.7 Signup as Patient

Use Case:	Signup as Patient
Actors:	Patient
Purpose:	Gather necessary information of Patient in English or Urdu, according to Patient's choice.
Overview:	The Patient requests a patient's signup form by choosing user-type as Patient and is provided with a comprehensive one, in which he/she can feed information pertaining to a Patient like name, user ID, password, street address, city, country, medical history, professional and social background etc., in English or Urdu, catering Patients of both the languages.
Type:	Primary and Essential

Typical Course of Events

Actor Action	System Response
1. This use case begins when a Patient requests a patient's signup form choosing user-type a Patient and clicking on signup link.	
	2. Asks the Patient to choose a language (English or Urdu).
3. Patient proceeds to the signup form in accordance with the language chosen.	
	4. Displays the form containing

input elements for required information, loading necessary lists' data dynamically.

5. Patient feeds the needed information, adding missing or required data into the dynamic lists.

6. The data is submitted and saved in database according to the language specified.

Alternative Courses

- Line 5: Added element is empty or already exists. Prompt to Patient and discontinue the operation.
- Line 6: Password fields fail to comply. Inform the Patient and abort submitting.

4.2.8 Signup as Service Provider

Use Case:	Signup as Service Provider
Actors:	Service Provider
Purpose:	Gather necessary information of Service Provider in English as well as in Urdu.
Overview:	The Service Provider requests a service provider's signup form by choosing user-type as Service Provider and is provided with a comprehensive one, in which it can feed information pertaining to a Service Provider like name, user ID, password, street address, city, country, location, available services and products etc., in both English and Urdu to cater both languages' users.
Type:	Primary and Essential

Typical Course of Events

Actor Action	***System Response***
1. This use case begins when a Service Provider requests a service provider's signup form choosing user-type a Service Provider and clicking on signup link.	
	2. Displays the form containing input elements for required information both in Urdu and English, loading necessary lists' data dynamically.
3. Service Provider feeds the general information, adding missing or required data into the dynamic lists.	
4. It proceeds to specify the details of provided services	
	5. Stores the general information

temporarily and displays a dynamic services specification form that allows multiple selections and addition of data elements unspecified or required by the Service Provider.

6. The data is submitted to be saved in database.

Alternative Courses

- Line 3, 5: Added element is empty or already exists. Prompt to Service Provider and discontinue the operation.
- Line 4: Password fields or specified location fails to comply. Inform the Service Provider and abort submitting.

4.2.9 Login

Use Case:	Login
Actors:	General User (initiator)
Purpose:	Authenticate the User when signing in.
Overview:	The General User selects a User type and follows a login procedure feeding ID and Password, to be verified for sign-in to take place.
Type:	Primary and Essential

Typical Course of Events

Actor Action	**System Response**
1. This use case begins when General User chooses a User type and submits the ID and Password.	
	2. Examines User type and steers to respective login page.
	3. Stores the submitted data temporarily while checking the ID and Password against database for User's authentication.
	4. Escorts User to the login page according to selected user-type: a. If signing in as a Doctor, initiate *Login as Doctor*. b. If signing in as a Patient, initiate *Login as Patient*. c. If signing in as a Service Provider, initiate *Login as Service Provider*.

Alternative Courses

- Line 3: The ID and Password do not match. Inform and provide for the User with a form to provide ID and Password again for approved authentication.

Related Use Cases

- Includes *Login as Doctor.*
- Includes *Login as Patient.*
- Includes *Login as Service Provider.*

4.2.10 Login as Doctor

Use Case:	Login as Doctor
Actors:	Doctor
Purpose:	Display patients' queue and facilitate attending them.
Overview:	The Doctor is steered to his/her login page when chosen user-type is Doctor and correct ID and Password are provided. The login page displays patients' queue showing their names, appointment times and status. The Doctor is able to attend each one by clicking on their names or pressing 'next' button.
Type:	Primary and Essential

Typical Course of Events

Actor Action	System Response
1. This use case begins when a Doctor is steered to his/her login page on choosing user-type a Doctor and providing correct ID and Password.	
	2. Says Hello to the Doctor and displays that day's patients' queue showing their names, appointment times and status.
3. The Doctor attends the patients by clicking on their names or pressing 'next' button.	
	4. Initiate *Investigate the Patient.*
	5. Manages 'patient name' links and shows their current status, with the passage of time.

Alternative Courses

- Line 5: It passes Doctor's online time. Just say 'thanks' and let doctor to continue.

Related Use Cases

- Includes *Investigate the Patient.*

4.2.11 Investigate Patient

Use Case:	Investigate the Patient
Actors:	Doctor (initiator), Patient
Purpose:	View Patient's information and communicate with to reach a diagnosis.
Overview:	The Doctor attends a Patient by viewing his/her personal and medical information, any previous visits and also any doctors' remarks. The Doctor communicates with the Patient by sending, or designing then sending questionnaires. Depending on the investigated diagnosis the Doctor suggests some tests or gives prescription and any remarks.
Type:	Primary and Essential

Typical Course of Events

Actor Action	System Response
1. This use case begins when a Doctor attends a patient by clicking on the name or pressing 'next' button.	
	2. Displays Patient's personal info like name, calculated age, language etc. and medical info including any previous visits and doctors' remarks.
3. The Doctor proceeds to investigation session.	
4. The Doctor communicates with the Patient by sending, or designing then sending questionnaires from a separate pane and waits for Patient's response.	
5. The Patient responds by answering the questionnaires or sending any comment.	
6. Depending on the investigated diagnosis, the Doctor suggests some tests or gives prescription and any remarks, and completes Patient's visit.	
	7. Facilitates and manages the whole thing by changing Patient's status, enabling sending of questionnaire, prescription and remarks, displaying Patient's response as answered questionnaires and Patient's comment, and controlling the overall session.

4.2.12 Design Questionnaire

Use Case:	Design Questionnaire
Actors:	Doctor
Purpose:	Plan and construct medical questionnaire/inquiry list.
Overview:	The Doctor is facilitated to make new questionnaires or append questions to questionnaires, under a specific disease category and subgroup. The questions in the questionnaires and the multiple choice options must be provided in both English and Urdu to cater patients of both the languages.
Type:	Primary and Essential

Typical Course of Events

Actor Action	**System Response**
1. This use case begins when a Doctor initiates designing of a questionnaire by selecting a disease category and its subgroup or adding these to the dynamic lists if not present already.	
	2. Displays any existing questionnaires showing their names, description and no. of questions, under the chosen disease category and subgroup.
3. The Doctor selects a questionnaire to view or edit it, or clicks a button to create a new one.	
	4. Opens and displays the questionnaire under the chosen disease category and subgroup, examining if its being edited or created: a. If being viewed or edited, displays the name and description, and any existing questions, with facility to append these to the selected questionnaire. b. If being created, provides input fields for name and description, and for the questions and their single or multiple choice answers, in both English and Urdu.
5. Gives questionnaire's name and description (for newly created) and appends questions including their answer types, specialized of guiding towards a diagnosis.	

Alternative Courses

- Line 1: Added element is empty or already exists. Prompt to Doctor and discontinue the operation.
- Line 5: The appended question already exists. Prompt to Doctor and abort appending.
- Line 5: For multiple choice answer type, no of choices is less than two or the choices are haphazard among English-Urdu fields. Inform the Doctor and abort the append operation.

4.2.13 Login as Patient

Use Case:	Login as Patient
Actors:	Patient
Purpose:	Displays Patient's appointments and previous visits, in the language specified at signup time.
Overview:	The Patient is steered to his/her login page, in the language previously specified, when chosen user-type is Patient and correct ID and Password are provided. The login page displays current, missed or future appointments of the Patient and also any previous visits, showing the doctor names, dates and times. The Patient is able to attend current appointments or answer questionnaires in 'pending' previous visits by clicking on the shown link.
Type:	Primary and Essential

Typical Course of Events

Actor Action	System Response
1. This use case begins when a Patient is steered to his/her login page in the language specified previously, on choosing user-type a Patient and providing correct ID and Password.	
	2. Says Hello to the Patient updating the status and displays Patient's appointments, showing the doctor names, dates and times, and indicating the appointments as current, missed or of future.
	3. Displays any previous visits indicating if there are 'pending' questionnaires to be answered.
5. The Patient clicks on a current appointment to attend.	6. Initiate *Wait in Queue.*
5. The Patient clicks on any previous visit.	6. Expands it to show questionnaires as a report or as a form to be filled if it's 'pending', also shows the doctor's prescription etc.

Related Use Cases

- Includes *Wait in Queue.*

4.2.14 Get Appointment

Use Case:	Get Appointment
Actors:	Patient
Purpose:	Facilitate Patient to get appointments from doctors of specific specializations.
Overview:	The Patient is facilitated to search a doctor of particular specialization and after viewing information showing doctor's caliber, chooses a date for the appointment. The system automatically specifies the appointment time keeping in view the appointments of other patients with the same doctor and doctor's online time. Patient is allowed to change the date if he/she finds the appointment time not suiting.
Type:	Primary and Essential

Typical Course of Events

Actor Action	**System Response**
1. This use case begins when a Patient chooses to search doctor of a particular specialization or the Patient knows the name and thus chooses to get the appointment directly.	
	2. Escorts Patient on basis of the choice: a. If searching on specialization or other criteria, initiate *Search Doctor.* b. If searching by name, displays information showing doctor's caliber and provides means to choose date for appointment.
3. The Patient after viewing the information selects a date for appointment.	
	4. Automatically allocates the appointment time keeping in view the appointments of other patients with the same doctor, doctor's online time and other considerations.
5. Patient can change the date if it doesn't suits or may consult another doctor.	

Alternative Courses

- Line 3: Appointment date chosen is a previous one. Alert the Patient.
- Line 4: A missed appointment with the same doctor exists. Delete from it database.
- Line 4: A current appointment with the same doctor already exists. Inform the Patient.
- Line 4: Time slot not available, or is available owing to less no. of patients but doctor's online time for 'today' is over. Suggest Patient to change the date or to consult another doctor.

Related Use Cases

- Includes *Search Doctor.*

4.2.15 Wait in Queue

Use Case:	Wait in Queue
Actors:	Patient (initiator), Doctor
Purpose:	Display patients' queue and facilitates consulting with the Doctor in terms of questionnaires, prescription / Doctor's comments and Patient's comments.
Overview:	Displays Doctor's name and his/her Patients' queue showing their names, appointment times and status. The system highlights current Patient's name, and facilitates Patient's waiting for turn, i.e. when the Doctor attends the Patient by viewing his/her information and starts investigation session by sending questionnaires or prescription/comments. The Patient is able to answer the sent questionnaires and also to send any comments.
Type:	Primary and Essential

Typical Course of Events

Actor Action	System Response
1. This use case begins when a Patient attends a current appointment by clicking on the doctor's name link.	
	2. Opens and displays Patients' queue for the Doctor, showing their names, appointment times and status.
	3. Changes status and highlights current Patient's name in the queue, while waiting for turn.
4. The Doctor attends the Patient and views his/her personal and medical information.	
	5. Makes the Patient attentive by changing status and asking to wait for Doctor's questionnaires.
6. The Doctor starts investigation	

session by sending questionnaires or prescription/comments.

7. The Patient answers the sent questionnaires or sends any comments.
8. Depending on the investigated diagnosis, . the Doctor suggests some tests or gives prescription and any remarks, and completes Patient's visit.
9. Facilitates and manages the whole thing by changing Patient's status, enabling sending of questionnaire, prescription and remarks, displaying Patient's response as answered questionnaires and Patient's comment, and controlling the overall session.

Alternative Courses

- Line 9: It passes Doctor's online time. Inform the Patient.

4.2.16 Search Doctor

Use Case:	Search Doctor
Actors:	General User, Patient
Purpose:	Search doctor of a particular specialization.
Overview:	General User and Patient are made able to search doctors specialized in the specific disease category. Other criteria include country, city (which can be unspecified) and fee limit. After searching database on the given criteria, the system displays a list of doctors showing their name and specifics which can lead to a decision like education, affiliation and experience. Only a Patient is allowed to proceed into viewing doctor's detailed info and getting an appointment.
Type:	Primary and Essential

Typical Course of Events

Actor Action	System Response
1. This use case begins when a General User or Patient requests to search doctor. of a particular specialization or disease category.	
	2. Displays a form containing input elements equipped with dynamic loading of data, for specifying the criteria that also includes country, city and fee limit, in General User's chosen or Patient's

specified language.

3. General User or Patient feeds the needed criteria to search on.

4. Uses the given criteria to search database and displays a list of doctors showing their name and specifics which can lead to a decision.

5. The Patient can proceed into getting an appointment by clicking on doctor's name link.

Alternative Courses

- Line 3: 'Numeric' fields fail to comply. Inform the General User or Patient and abort searching procedure.

4.2.17 Search Service Provider

Use Case:	Search Service Provider
Actors:	General User
Purpose:	Search service provider with particular service details.
Overview:	General User is made able to search service providers ascertaining provision of certain services, their categories and 'products' within. Other criteria include country, city (which can be unspecified) and specific area within the city (which can be unspecified). After searching database on the given criteria, the system displays a list of service providers showing their name, address and website.. General User may click on name link to view full detail e.g. reaching hint and all service products. General User is also able to view current info of a service provider by giving search on name.
Type:	Primary and Essential

Typical Course of Events

Actor Action	**System Response**
1. This use case begins when a General User requests to search service provider ascertaining provision of certain services and categories within.	
	2. Displays a form containing input elements equipped with dynamic loading of data, for specifying the criteria that also includes country, city and sub area, in General User's chosen language.
3. General User feeds the needed criteria to search on.	
	4. Uses the given criteria to search database and displays a list of

5. General User can click on name link to view full detail.	service providers showing their initial info.
6. General User is able to view current info of a service provider giving search by name.	

4.2.18 Login as Service Provider

Use Case:	Login as Service Provider (not in scope as yet)
Actors:	Service Provider
Purpose:	Provide editing facility of service details.
Overview:	The Service Provider is steered to his/her login page when chosen user-type is Doctor and correct ID and Password are provided. The login page should provide links for service details editing but its not in scope as yet. Right now it just displays all the information captured during Service Provider's signup.
Type:	Primary and Essential

Typical Course of Events

Actor Action	System Response
1. This use case begins when a Service Provider is steered to its login page on choosing user-type a Service Provider and providing correct ID and Password.	
	2. Says Hello to the Service Provider and displays all the information captured during Service Provider's signup including organization specific and details of services provided.

4.3 Use Case Diagrams

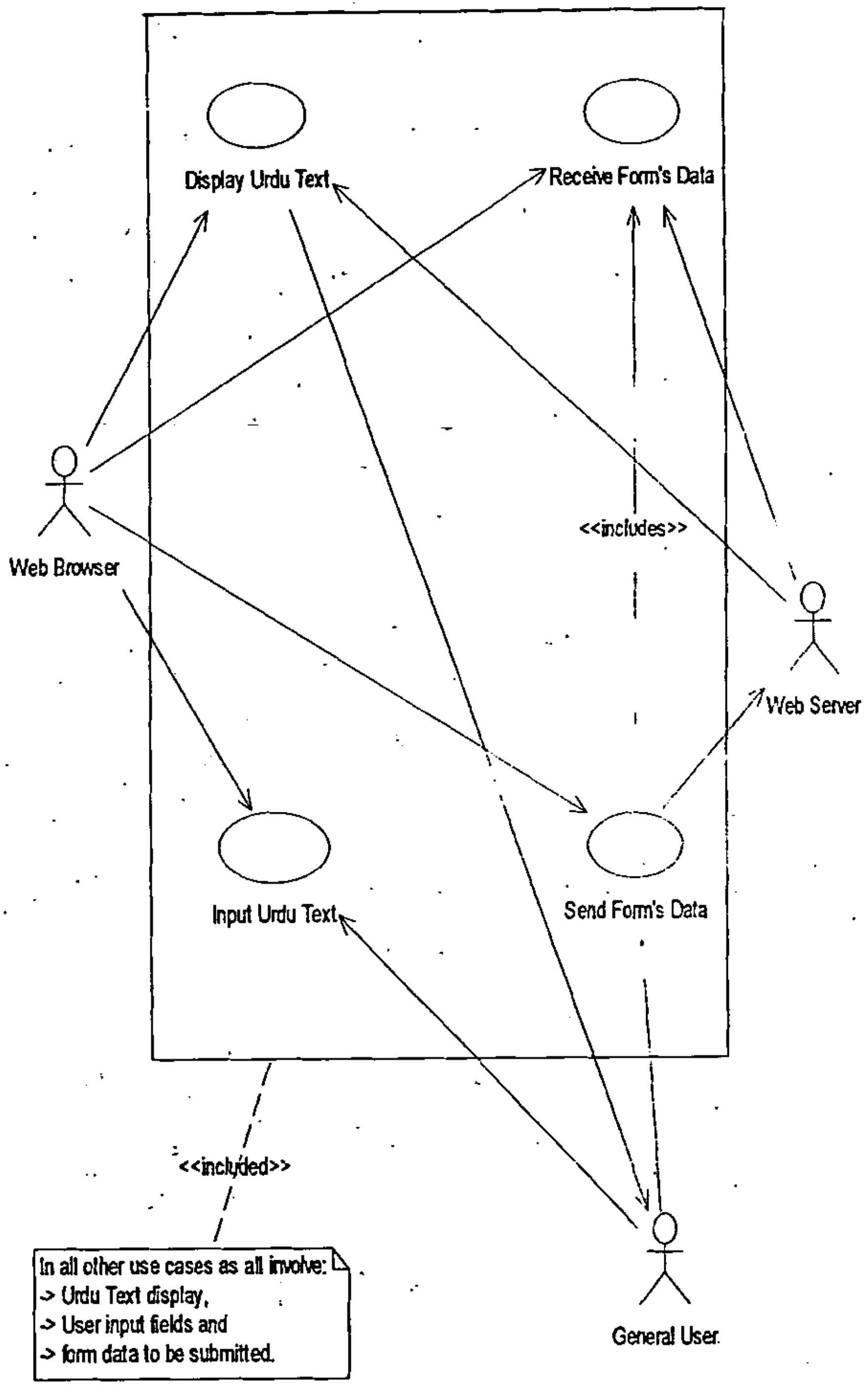


Figure 4.1 : Backbone for Urdu Processing

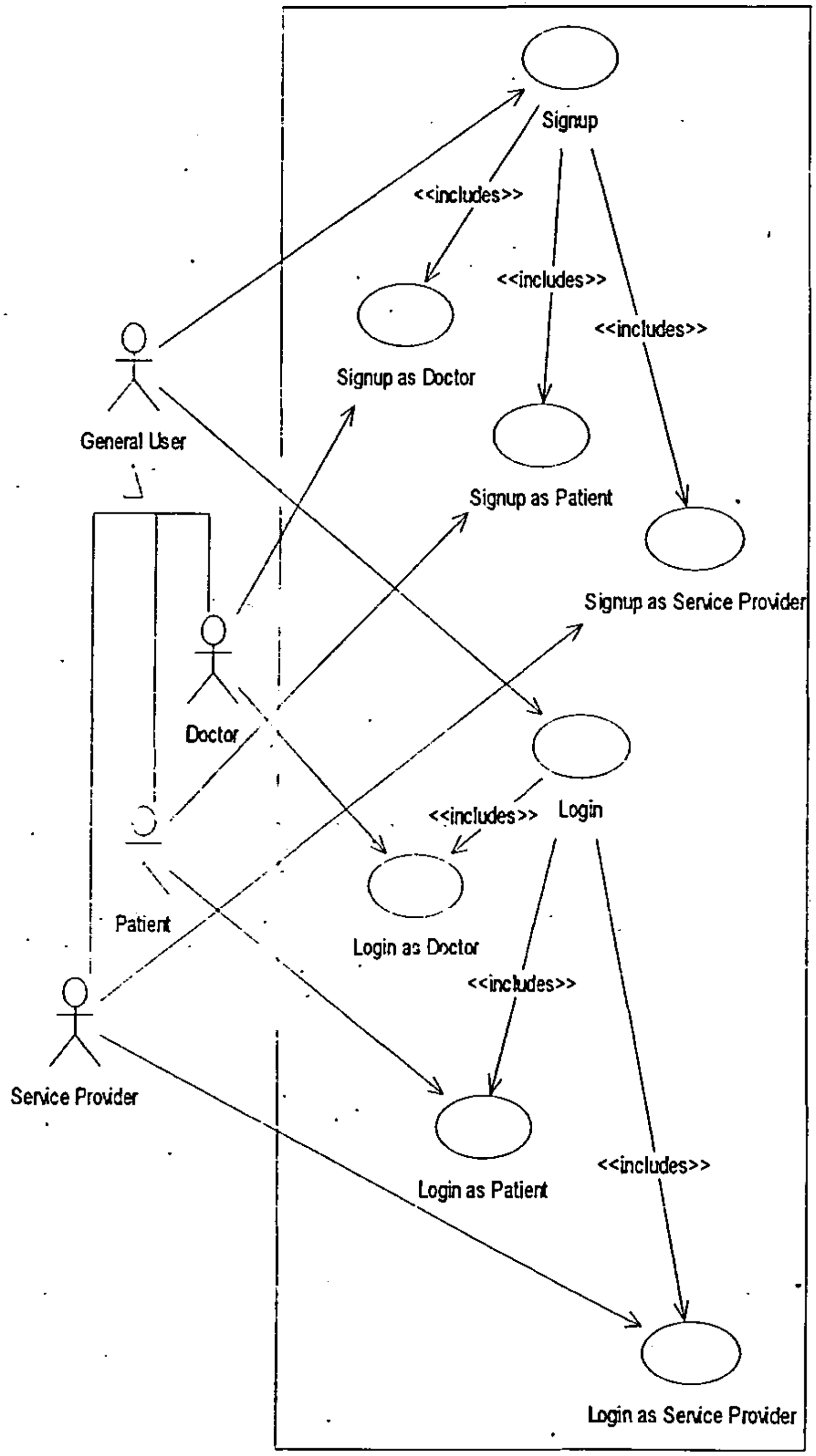


Figure 4.2 : Signup and login into Telemedicine System

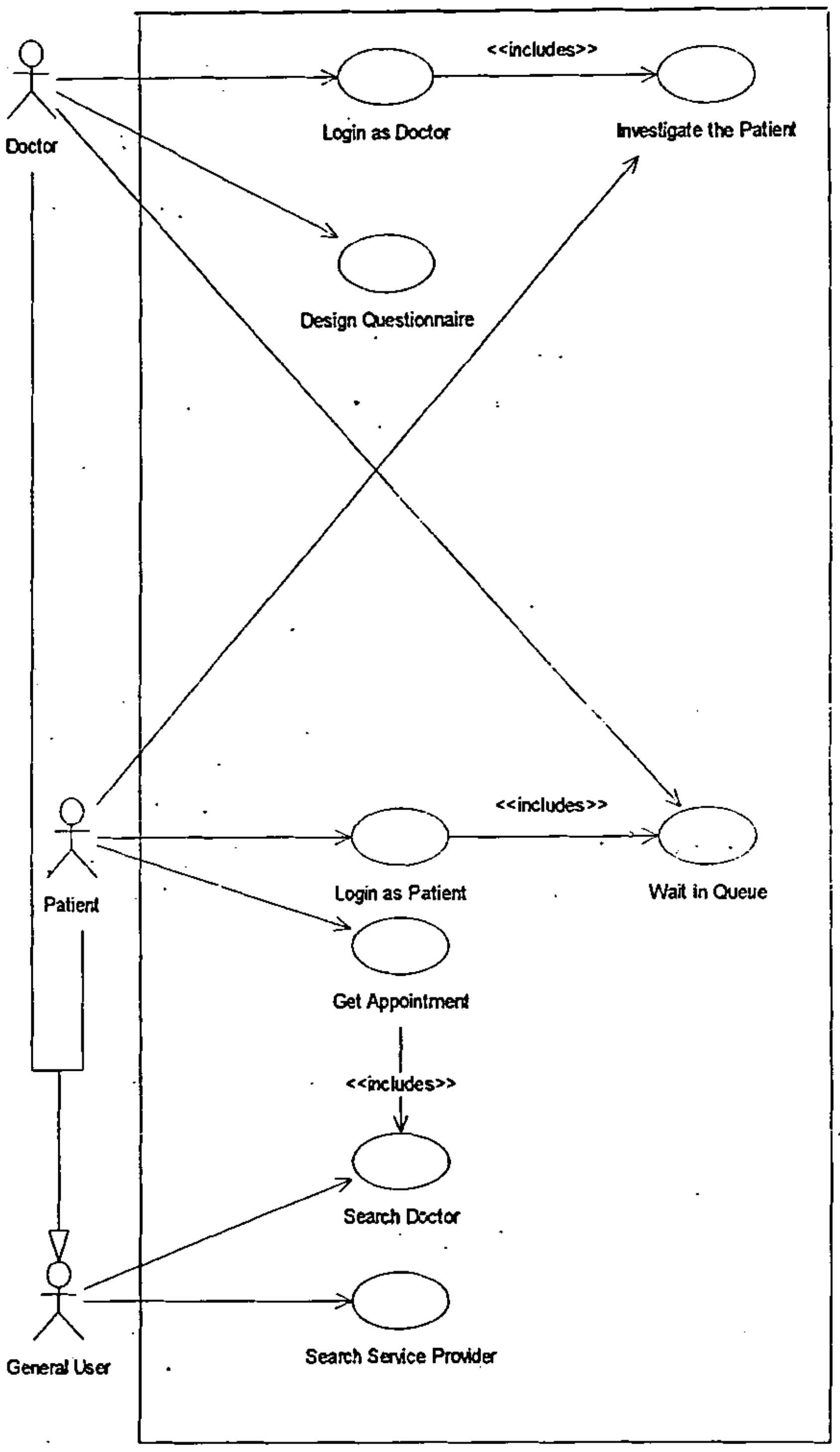
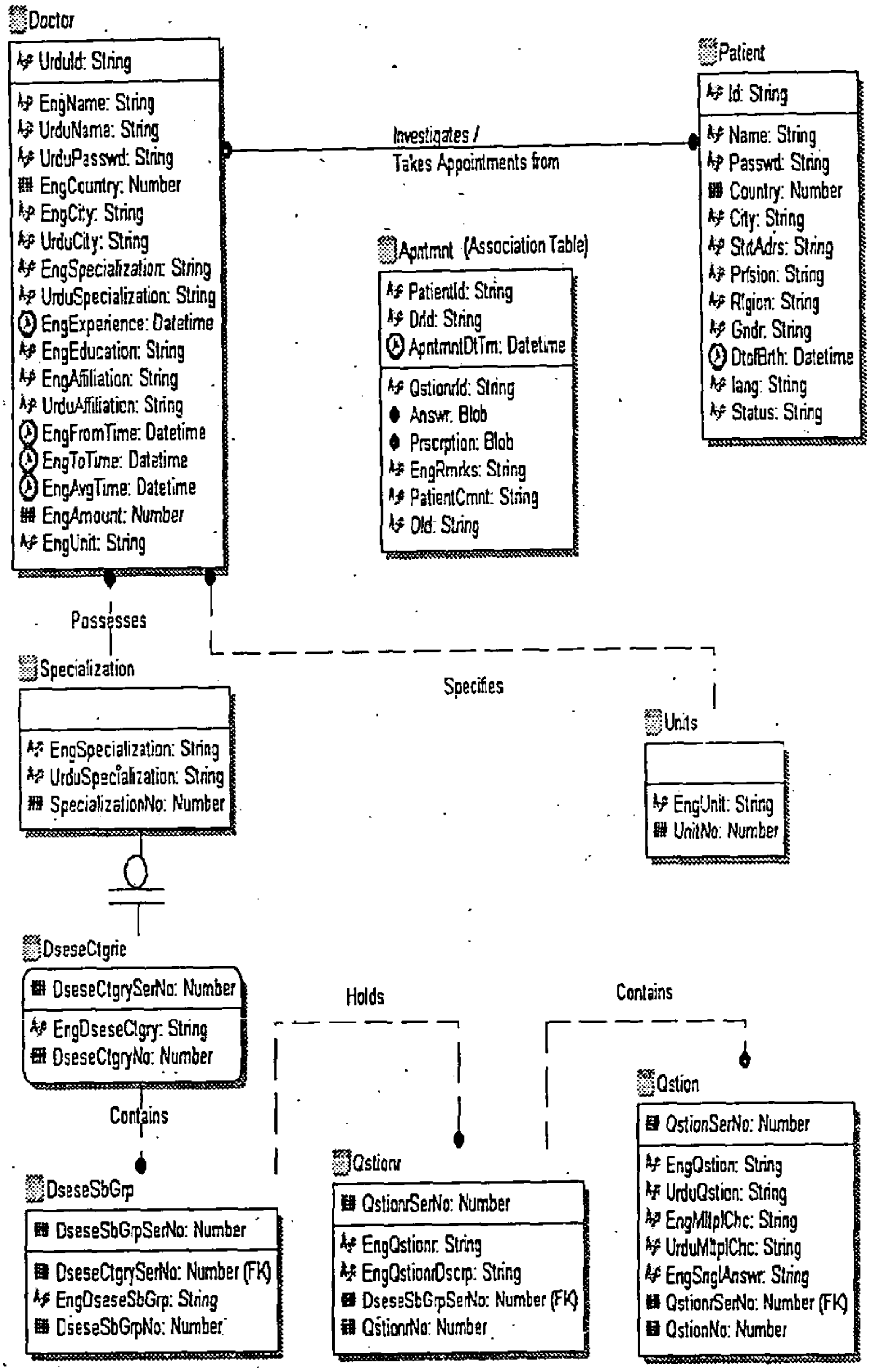
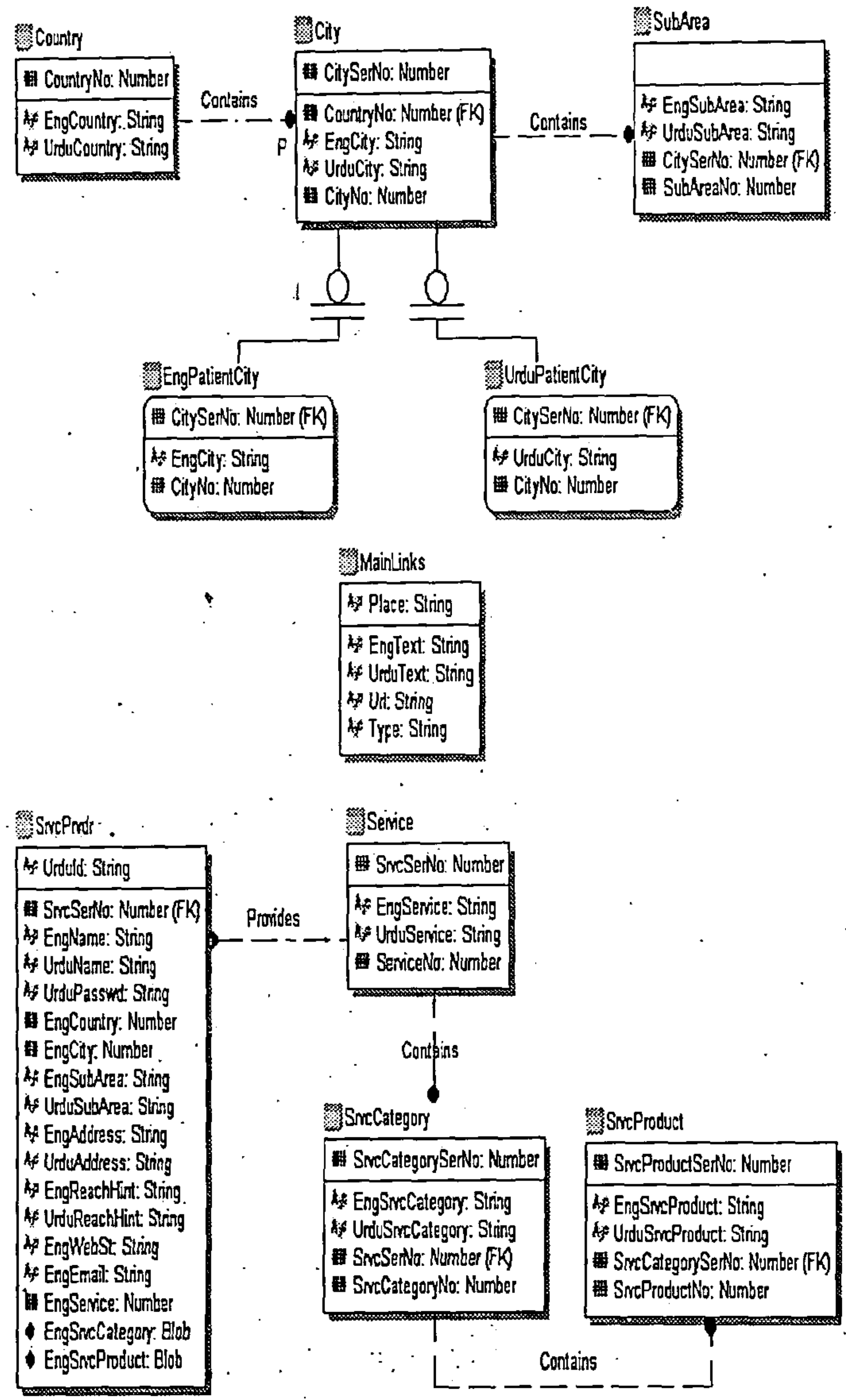


Figure 4.3 : System Operations of Urdu Telemedicine

4.4 Conceptual Model (Entity Relationship Diagram)




Country
CountryNo: Number
EngCountry: String
UrduCountry: String
Contains
P
City
CitySerNo: Number
CountryNo: Number (FK)
EngCity: String
UrduCity: String
CityNo: Number
Contains
SubArea
EngSubArea: String
UrduSubArea: String
CitySerNo: Number (FK)
SubAreaNo: Number
EngPatientCity
CitySerNo: Number (FK)
EngCity: String
CityNo: Number
UrduPatientCity
CitySerNo: Number (FK)
UrduCity: String
CityNo: Number
MainLinks
Place: String
EngText: String
UrduText: String
Url: String
Type: String
SrvcPrvdr
UrduId: String
SrvcSerNo: Number (FK)
EngName: String
UrduName: String
UrduPasswd: String
EngCountry: Number
EngCity: Number
EngSubArea: String
UrduSubArea: String
EngAddress: String
UrduAddress: String
EngReachHint: String
UrduReachHint: String
EngWebSt: String
EngEmail: String
EngService: Number
EngSrvcCategory: Blob
EngSrvcProduct: Blob
Provides
Service
SrvcSerNo: Number
EngService: String
UrduService: String
ServiceNo: Number
Contains
SrvcCategory
SrvcCategorySerNo: Number
EngSrvcCategory: String
UrduSrvcCategory: String
SrvcSerNo: Number (FK)
SrvcCategoryNo: Number
SrvcProduct
SrvcProductSerNo: Number
EngSrvcProduct: String
UrduSrvcProduct: String
SrvcCategorySerNo: Number (FK)
SrvcProductNo: Number
Contains

Chapter 5

Algorithm Design

5. Algorithm Design

This chapter presents the design of the algorithms that could be implemented in any computer language for testing or utilizing this research. Requirements for these methods have been established in the previous chapter. According to Object-oriented design approach, UML notations and tools [18] will be used to draw diagrams that will present the overall structure, functionality and operation of the system under discussion. Keeping in view the need and scope of this research, two of the UML usual notions of diagrams for design are presented, namely: Activity diagrams and Sequence diagrams. UML utilizes the understanding of the system, developed in the requirements analysis phase in the form of *expanded use cases*, as a basis to further elaborate the tasks accomplish by the system. In the following sections, each use case will be presented diagrammatically according to the two notions given above. In order to complete the design of the system the physical model for database will be presented.

5.1 Activity Diagrams

An activity diagram represents the expanded form of use cases as a series of activities or operations, as per the system performs these. This way one can easily comprehend the functionality of the application and the sequence within, in simple terms.

Following subsections present the <<traces>> of use cases discussed in chapter 4 corresponding to their *activity diagrams*. Traces are provided in figures with self explanatory actions with arrows showing their sequence.

5.1.1 Display Urdu Text

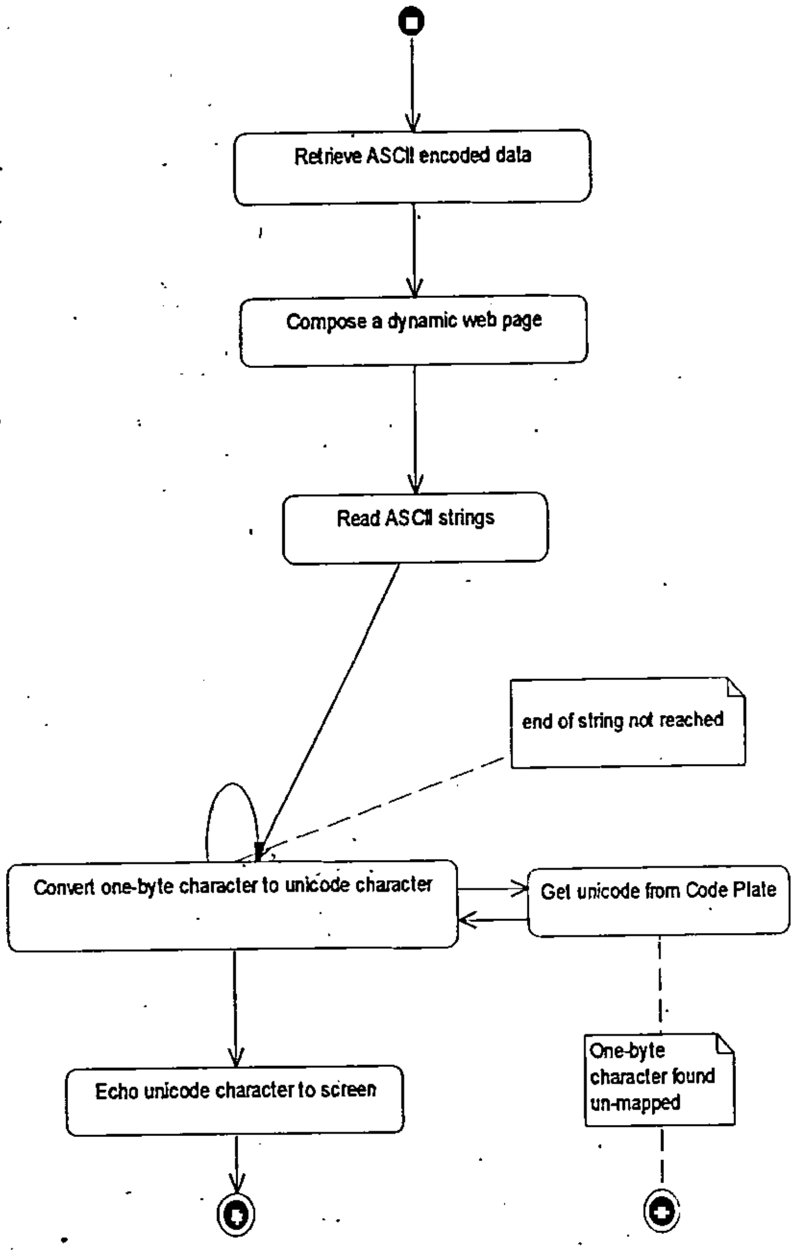

Retrieve ASCII encoded data
Compose a dynamic web page
Read ASCII strings
end of string not reached
Convert one-byte character to unicode character
Get unicode from Code Plate
Echo unicode character to screen
One-byte character found un-mapped

Figure 5.1 Traces for Displaying Urdu Text

5.1.2 Input Urdu Text

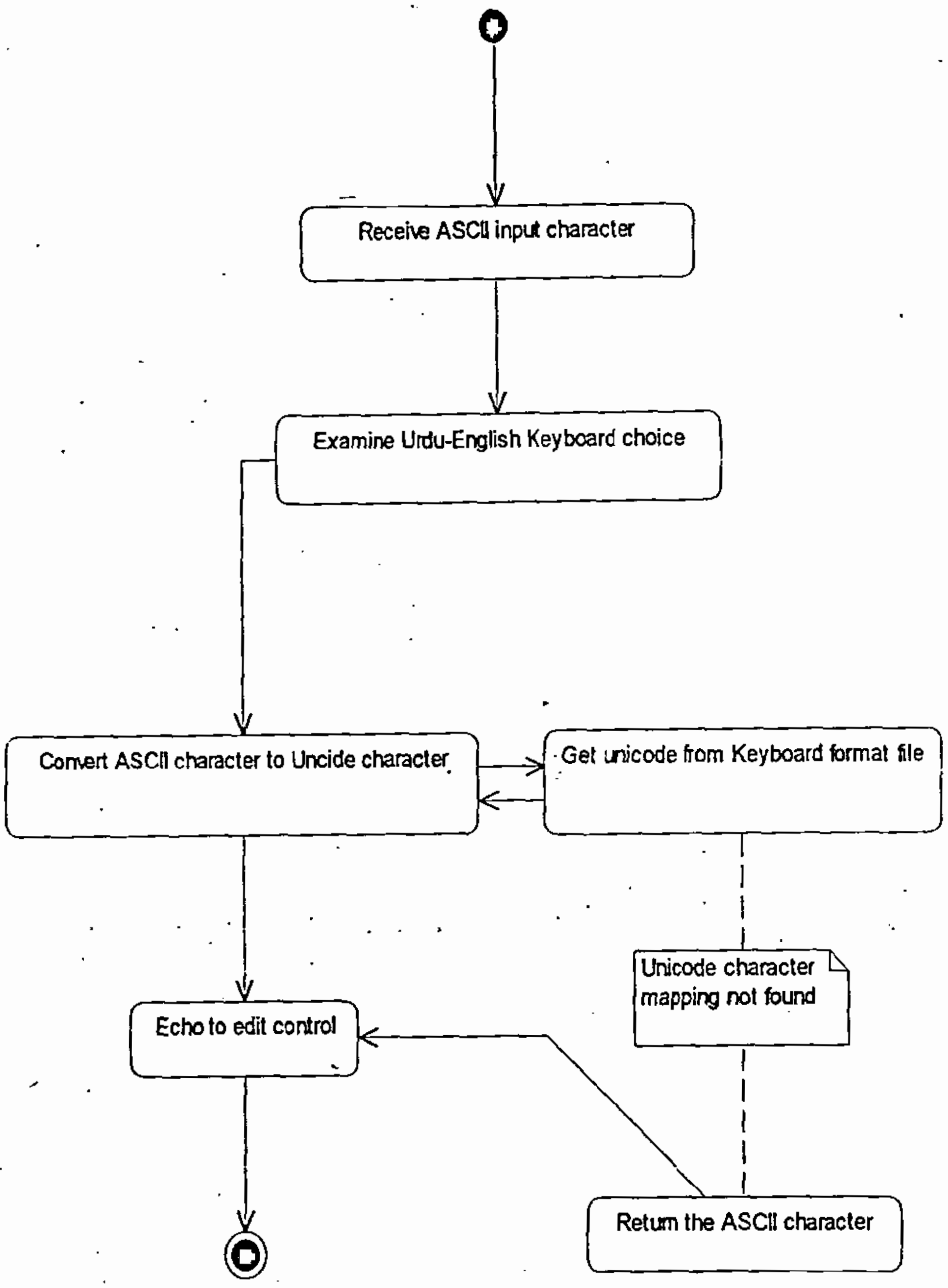

Receive ASCII input character
Examine Urdu-English Keyboard choice
Convert ASCII character to Uncide character
Get unicode from Keyboard format file
Unicode character mapping not found
Echo to edit control
Return the ASCII character

Figure 5.2 Traces for Input Urdu and English Text

5.1.3 Send Form Data (Client End)

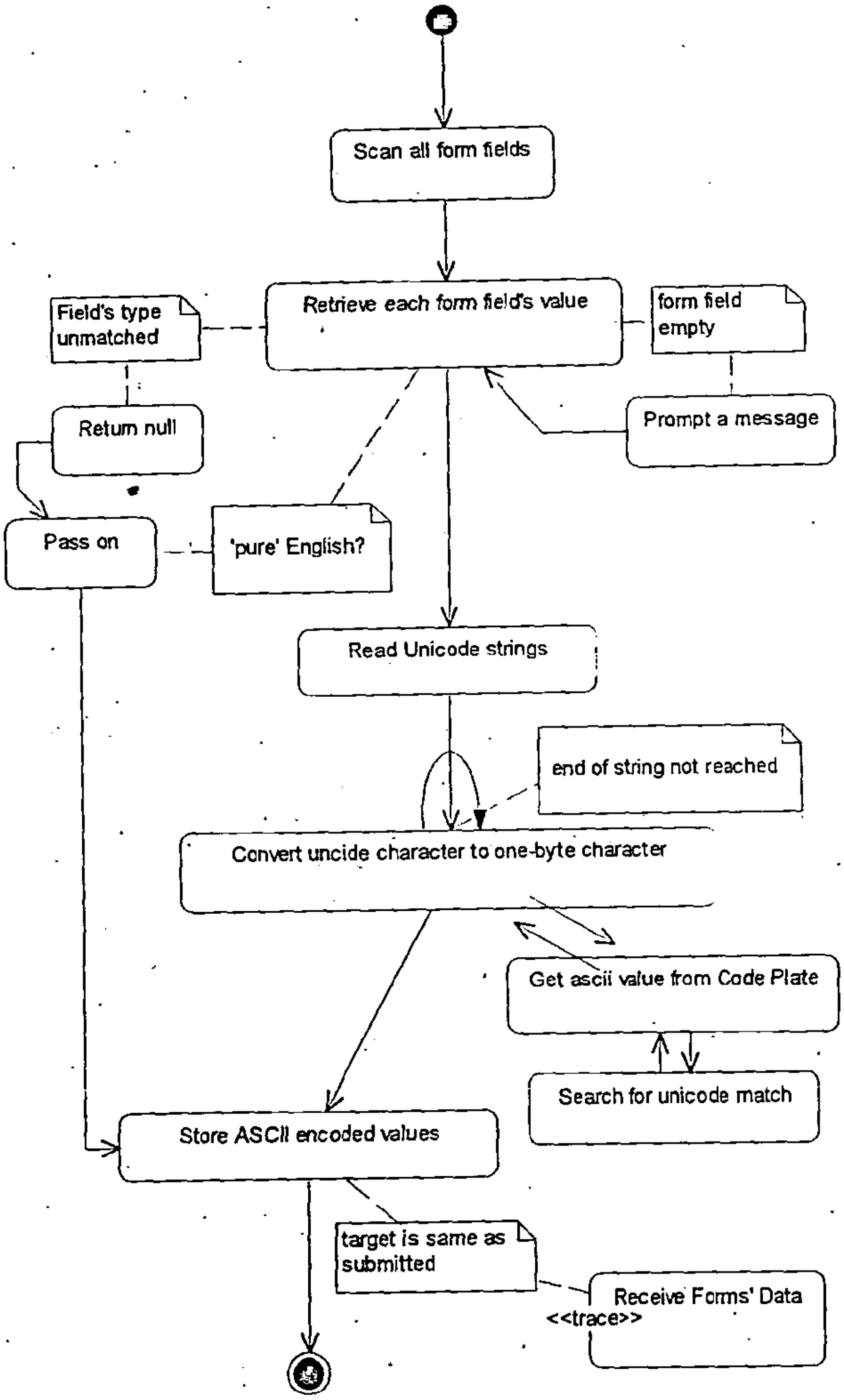

Scan all form fields
Retrieve each form field's value
Field's type unmatched
form field empty
Return null
Prompt a message
Pass on
'pure' English?
Read Unicode strings
end of string not reached
Convert uncide character to one-byte character
Get ascii value from Code Plate
Search for unicode match
Store ASCII encoded values
target is same as submitted
Receive Forms' Data
<<trace>>

Figure 5.3 Traces for Form Data

5.1.4 Receive Forms' Data (Server End)

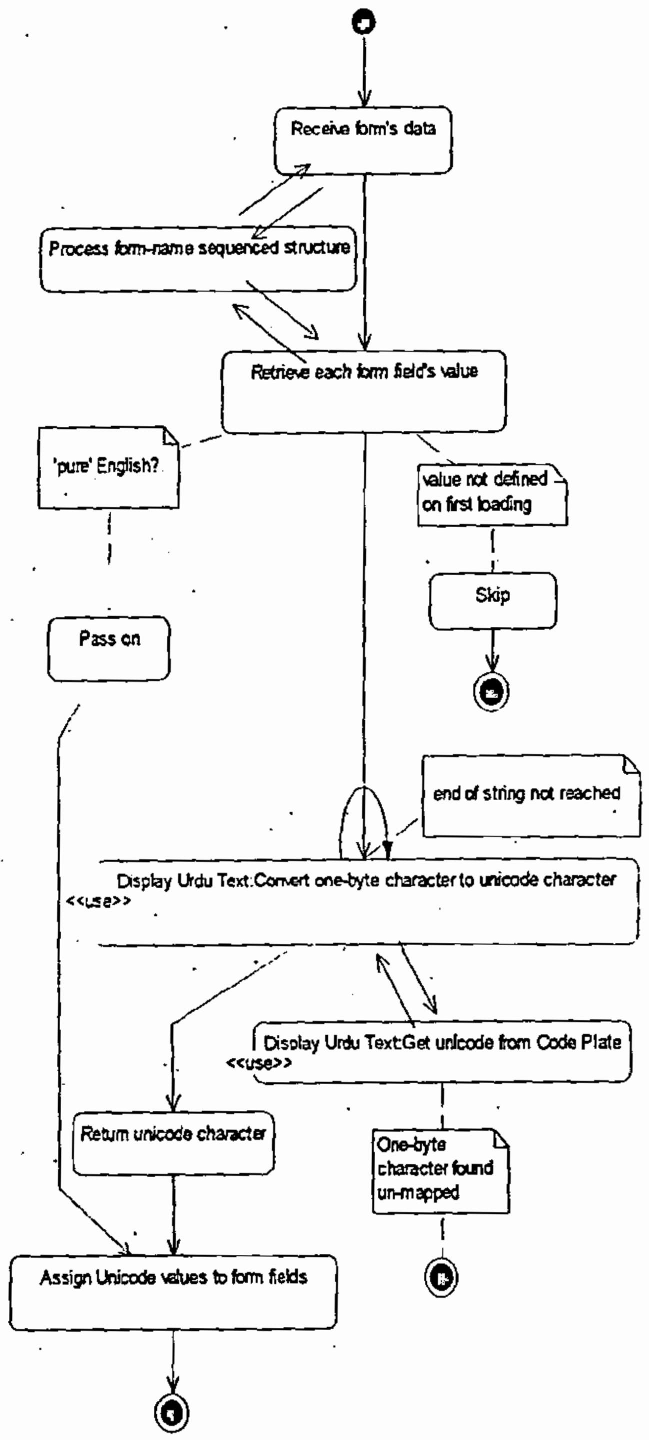

Receive form's data
Process form-name sequenced structure
Retrieve each form field's value
'pure' English?
value not defined on first loading
Skip
Pass on
end of string not reached
Display Urdu Text:Convert one-byte character to unicode character
<<use>>
Display Urdu Text:Get unicode from Code Plate
<<use>>
Return unicode character
One-byte character found un-mapped
Assign Unicode values to form fields

Figure 5.4 Traces for Receive Form Data

5.1.5 Common Procedure of Signup

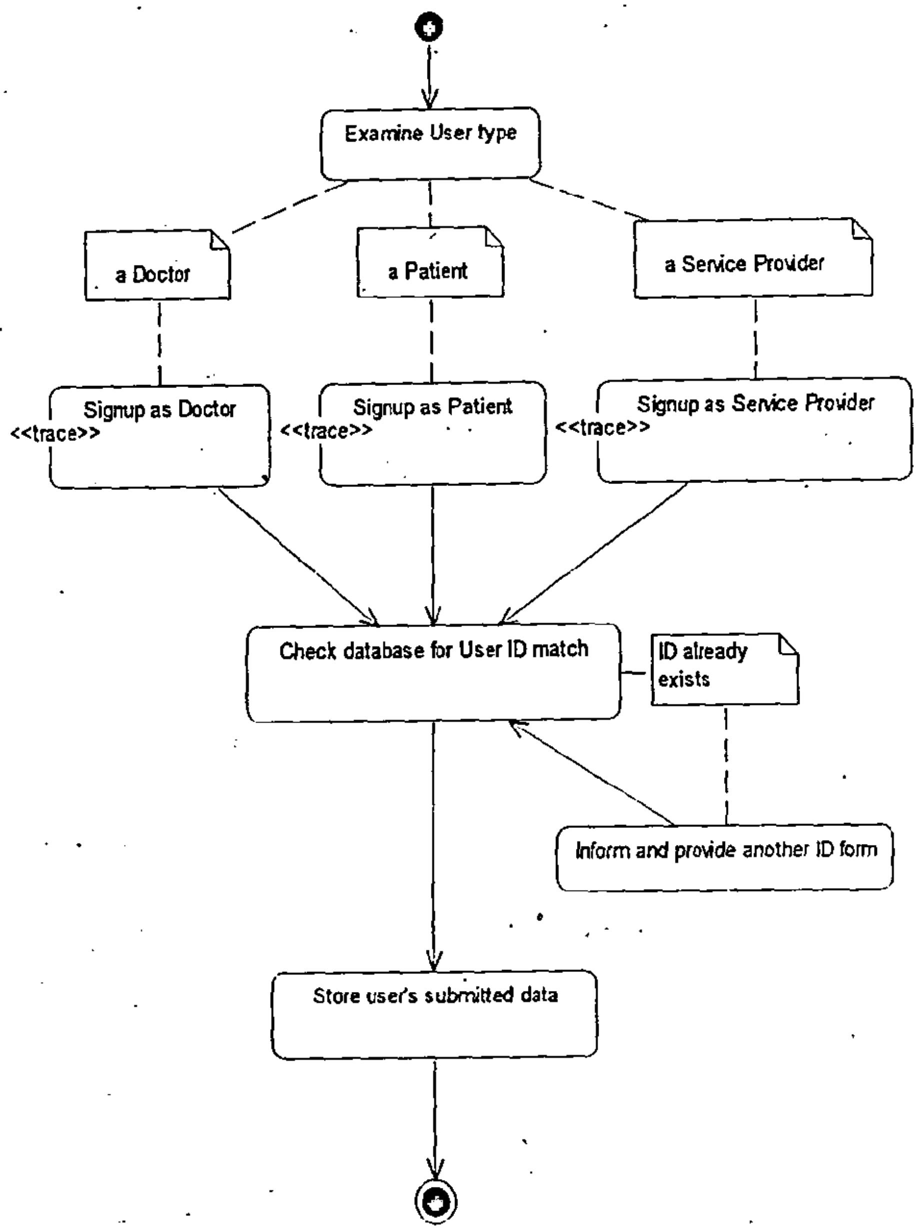


Figure 5.5 Traces for Common Signup Procedure for All Users

5.1.6 Signup as Doctor

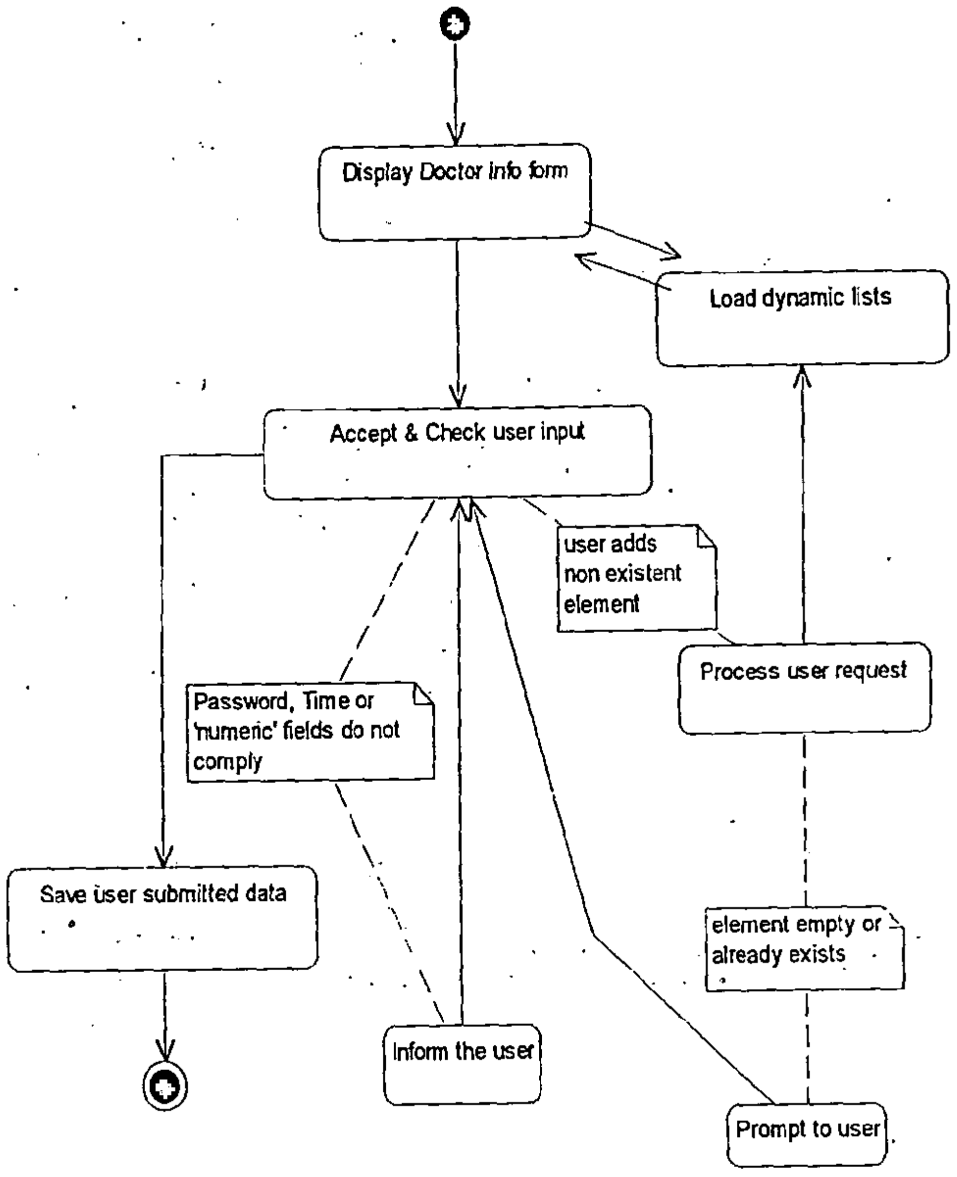

Display Doctor info form
Load dynamic lists
Accept & Check user input
user adds non existent element
Process user request
Password, Time or 'numeric' fields do not comply
Save user submitted data
element empty or already exists
Inform the user
Prompt to user

Figure 5.2 Traces for Signup as Doctor

5.1.7 Signup as Patient

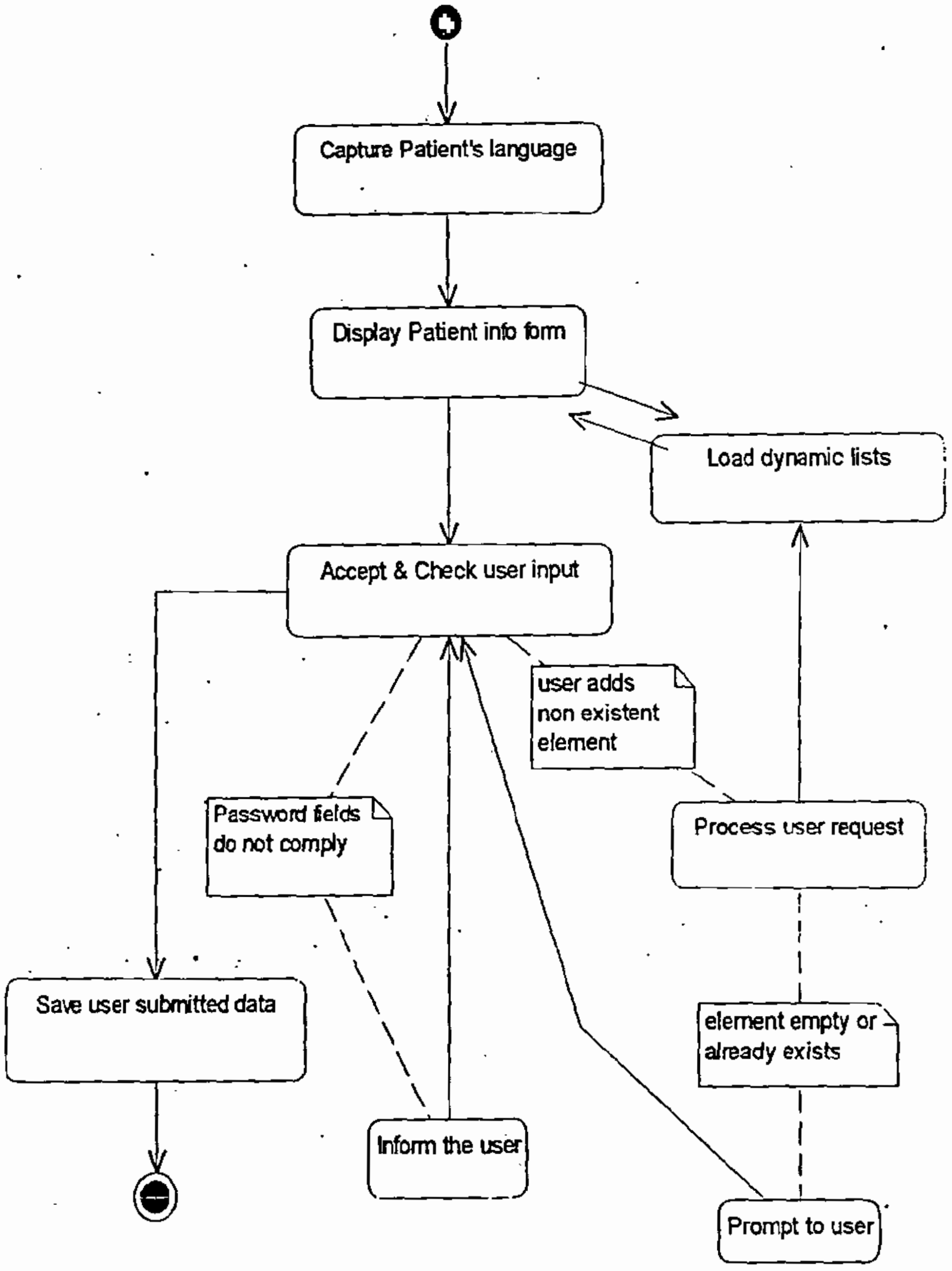

Capture Patient's language
Display Patient info form
Load dynamic lists
Accept & Check user input
user adds non existent element
Password fields do not comply
Process user request
Save user submitted data
element empty or already exists
Inform the user
Prompt to user

Figure 5.7 Traces for Sigup as Patient

5.1.8 Signup as Service Provider

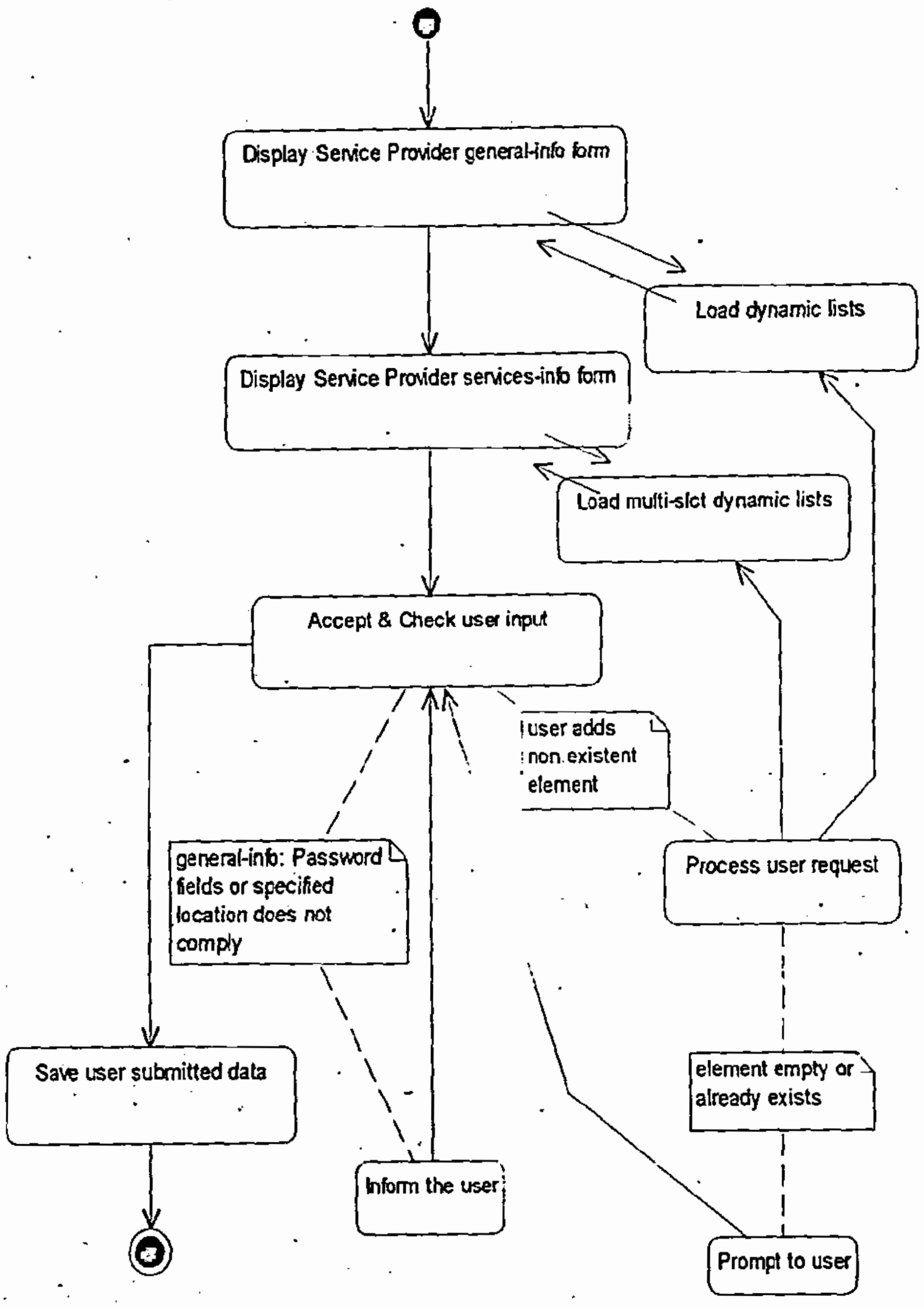

Display Service Provider general-info form
Load dynamic lists
Display Service Provider services-info form
Load multi-slct dynamic lists
Accept & Check user input
user adds non existent element
general-info: Password fields or specified location does not comply
Process user request
Save user submitted data
element empty or already exists
Inform the user
Prompt to user

Figure 5.8 Traces for Signup as Service Provider

5.1.9 Common Login Procedure

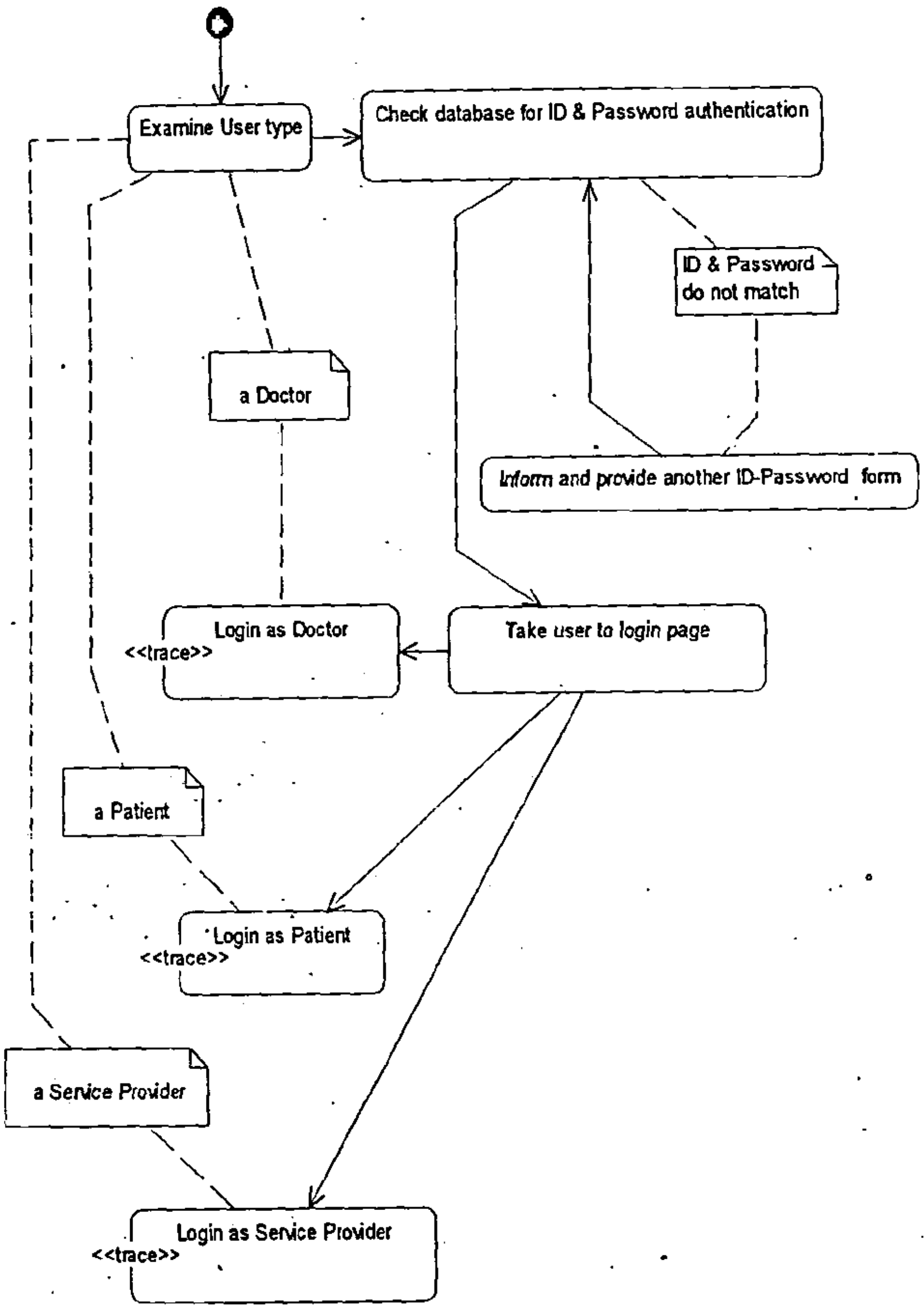

Examine User type
Check database for ID & Password authentication
ID & Password do not match
a Doctor
Inform and provide another ID-Password form
Login as Doctor
<<trace>>
Take user to login page
a Patient
Login as Patient
<<trace>>
a Service Provider
Login as Service Provider
<<trace>>

Figure 5.9 Traces for Login Procedure Common to All Users

5.1.10 Login as Doctor

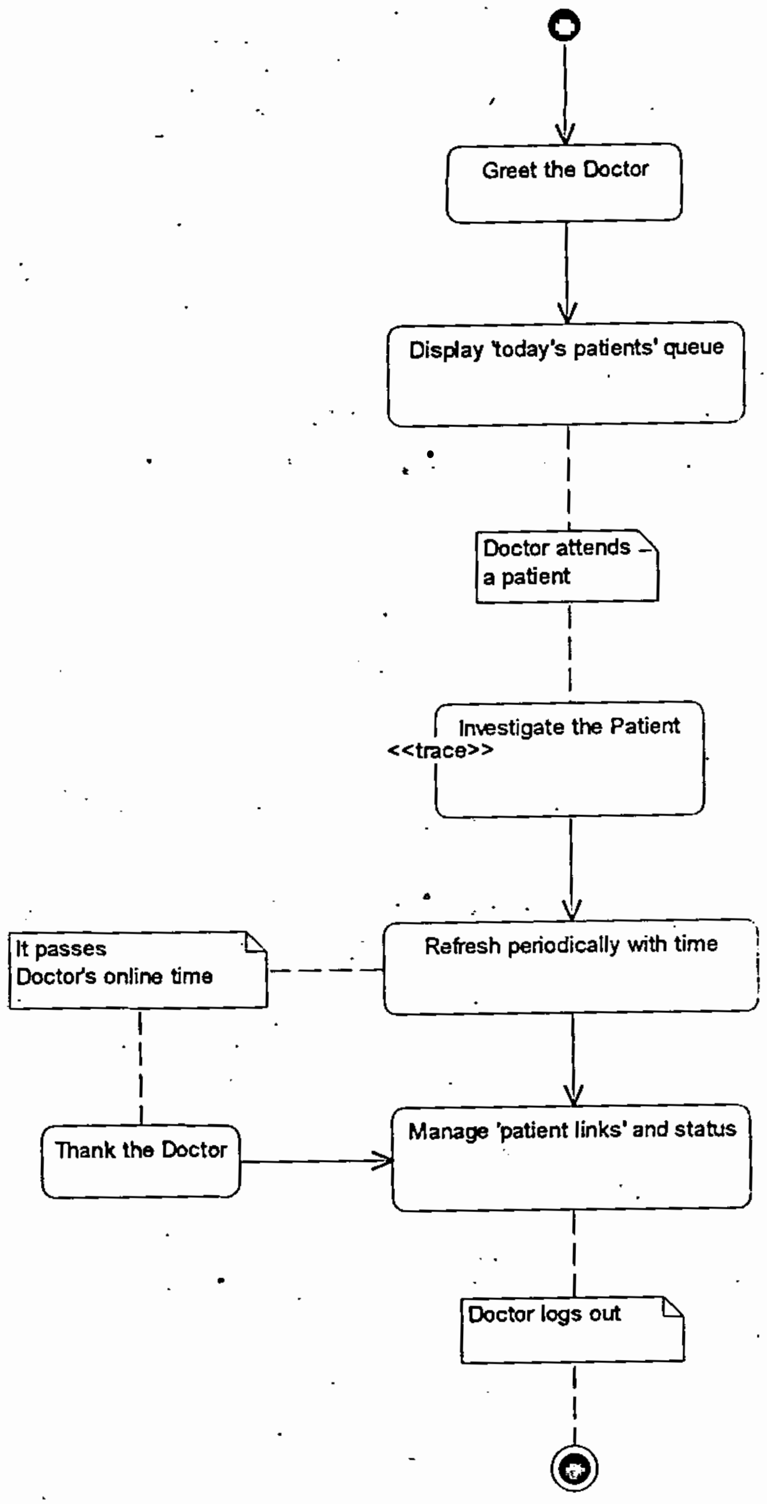

Greet the Doctor
Display 'today's patients' queue
Doctor attends a patient
Investigate the Patient
<<trace>>
Refresh periodically with time
It passes Doctor's online time
Thank the Doctor
Manage 'patient links' and status
Doctor logs out

Figure 5.10 Traces for Login as Doctor

5.1.11 Investigate the Patient

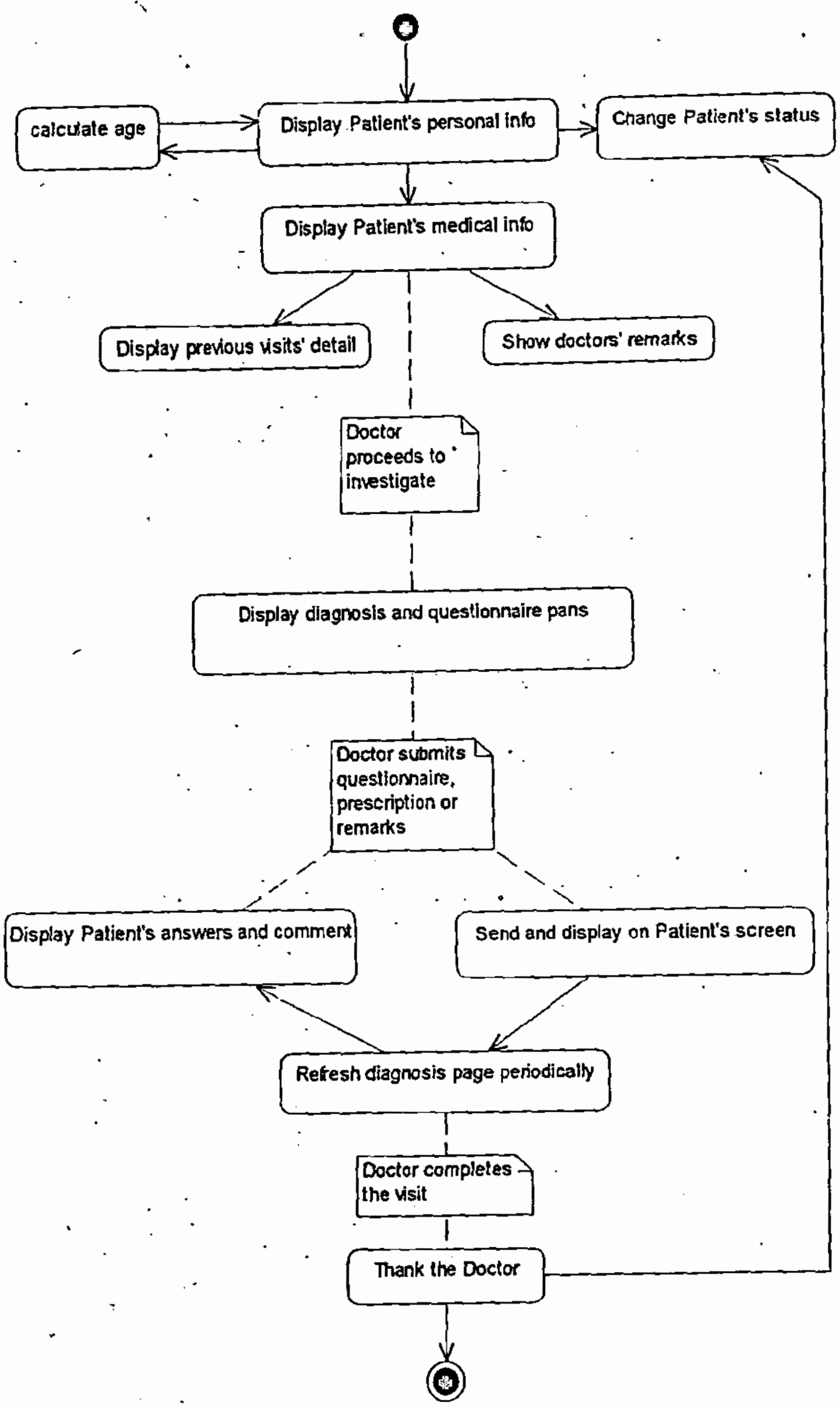

calculate age
Display Patient's personal info
Change Patient's status
Display Patient's medical info
Display previous visits' detail
Show doctors' remarks
Doctor proceeds to investigate
Display diagnosis and questionnaire pans
Doctor submits questionnaire, prescription or remarks
Display Patient's answers and comment
Send and display on Patient's screen
Refresh diagnosis page periodically
Doctor completes the visit
Thank the Doctor

Figure 5.11 Traces for Investigate Patient

5.1.12 Design Questionnaire

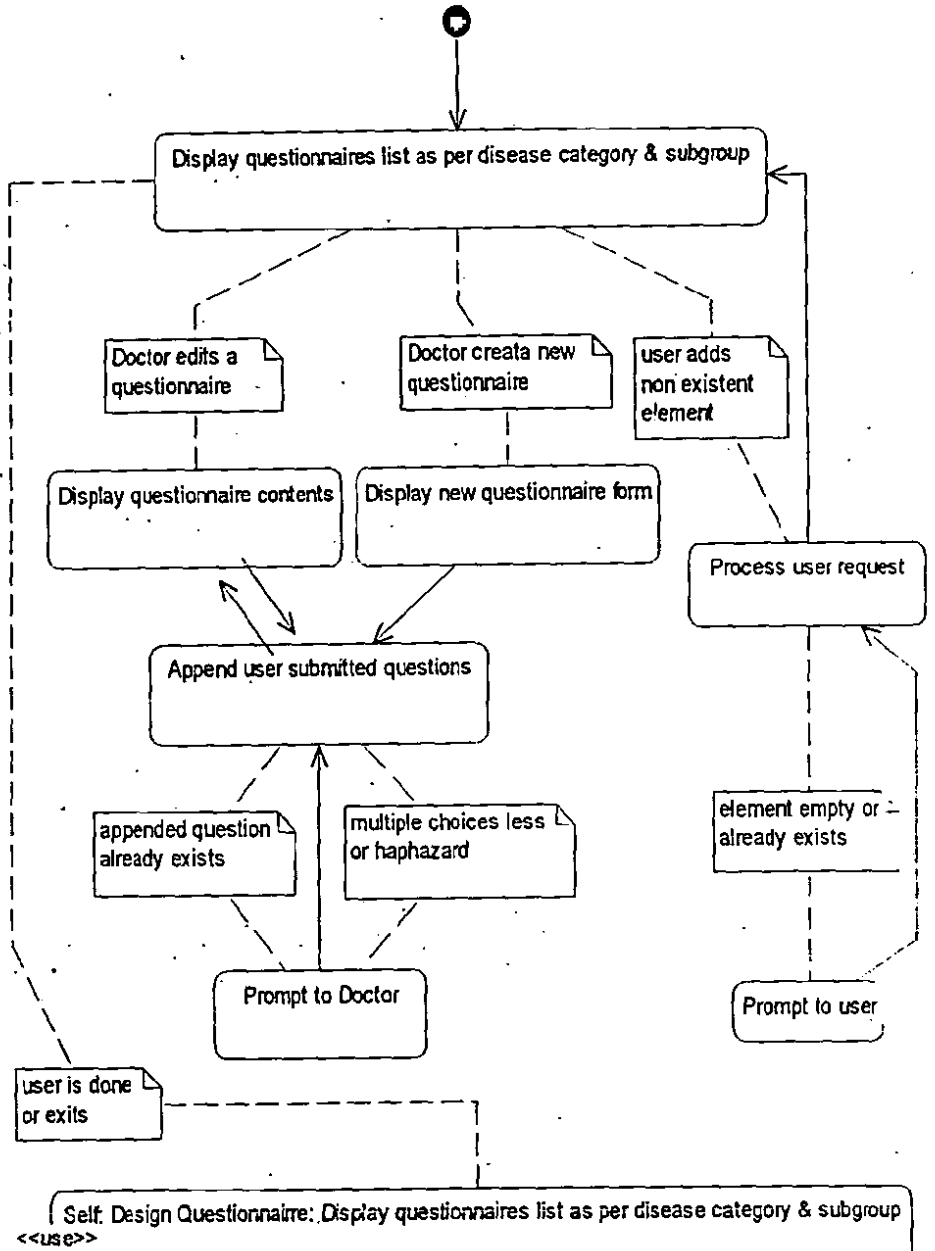

Display questionnaires list as per disease category & subgroup
Doctor edits a questionnaire
Doctor creata new questionnaire
user adds non existent element
Display questionnaire contents
Display new questionnaire form
Process user request
Append user submitted questions
appended question already exists
multiple choices less or haphazard
element empty or already exists
Prompt to Doctor
Prompt to user
user is done or exits
Self: Design Questionnaire: Display questionnaires list as per disease category & subgroup
<<use>>

Figure 5.12 Traces for Design Questionnaire

5.1.13 Login as Patient

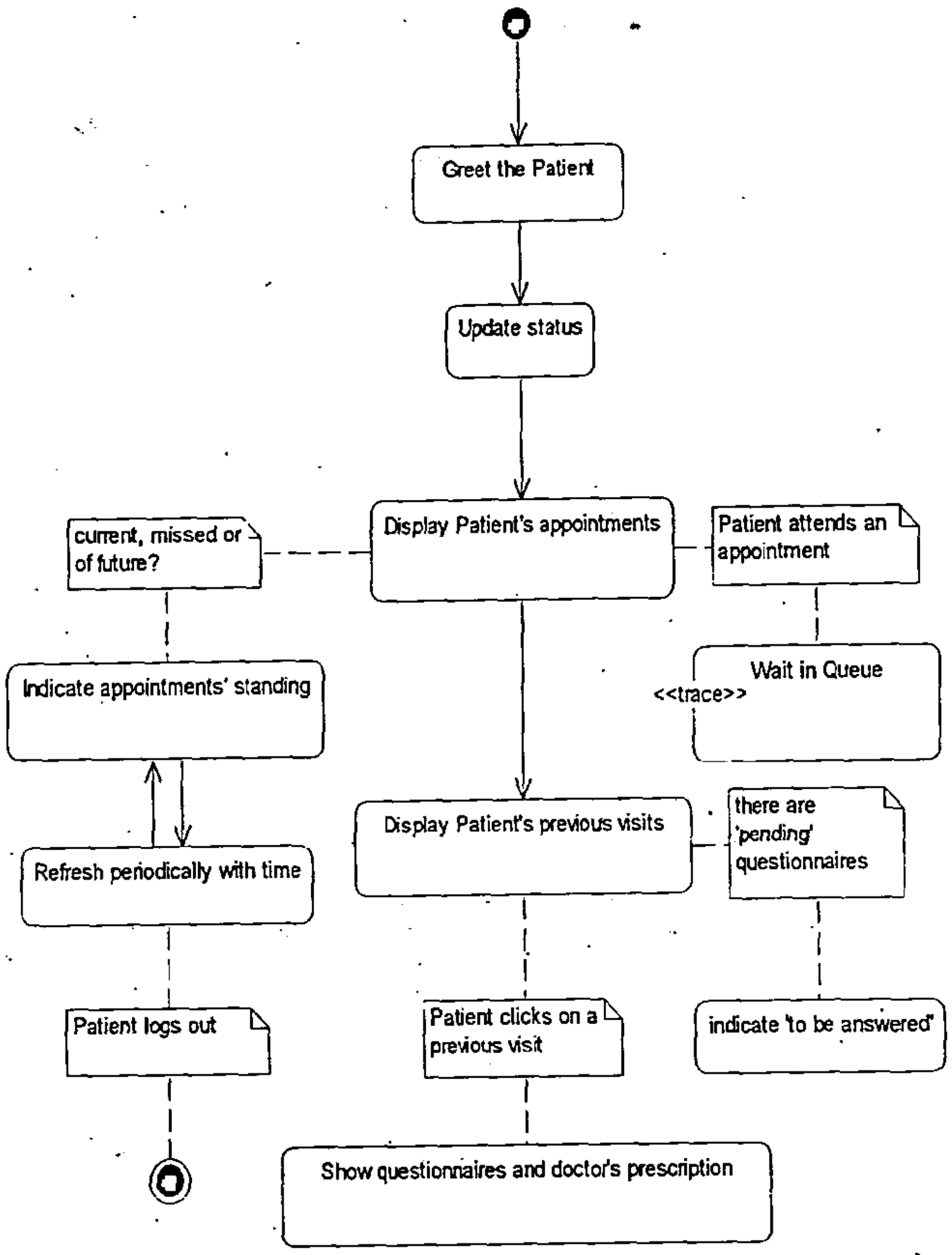

Greet the Patient
Update status
Display Patient's appointments
current, missed or of future?
Patient attends an appointment
Indicate appointments' standing
Wait in Queue
<<trace>>
Refresh periodically with time
Display Patient's previous visits
there are 'pending' questionnaires
Patient logs out
Patient clicks on a previous visit
indicate 'to be answered'
Show questionnaires and doctor's prescription

Figure 5.13 Traces for Login as Patient

5.1.14 Get Appointment

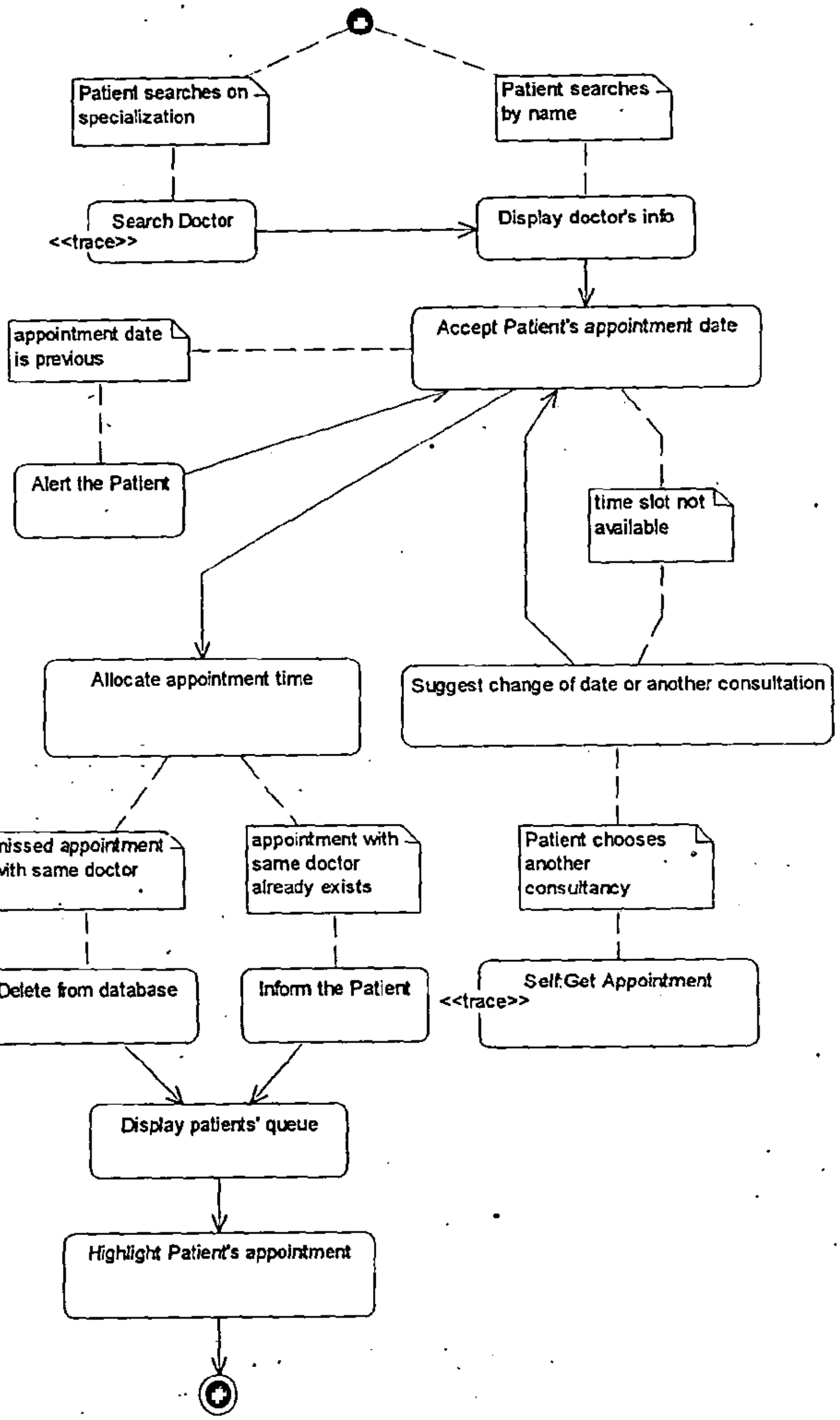

Patient searches on specialization
Patient searches by name
Search Doctor
<<trace>>
Display doctor's info
Accept Patient's appointment date
appointment date is previous
Alert the Patient
time slot not available
Allocate appointment time
Suggest change of date or another consultation
missed appointment with same doctor
appointment with same doctor already exists
Patient chooses another consultancy
Delete from database
Inform the Patient
<<trace>>
Self:Get Appointment
Display patients' queue
Highlight Patient's appointment

Figure 5.14 Traces for Get Appointment

5.1.15 Wait in Queue

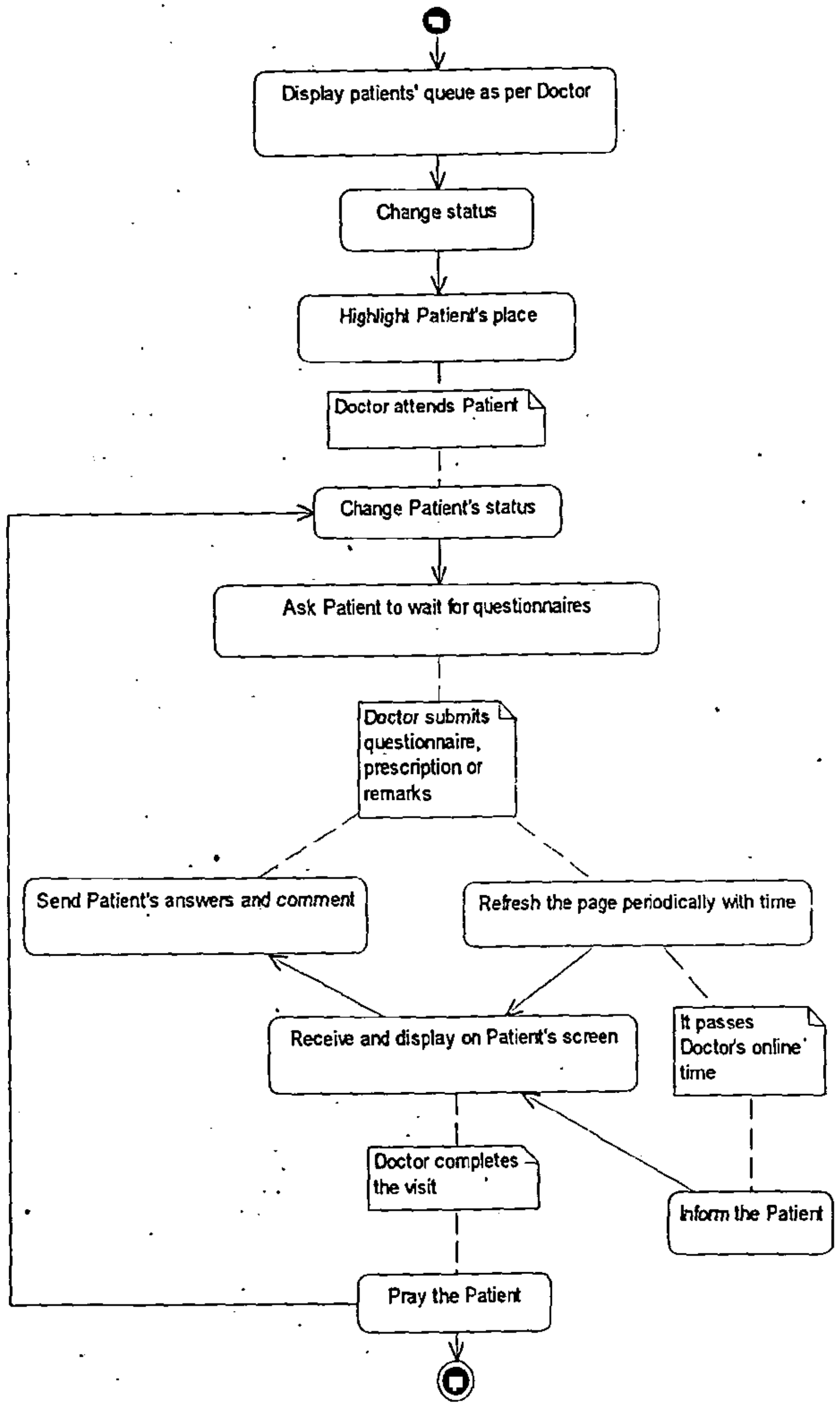

Display patients' queue as per Doctor
Change status
Highlight Patient's place
Doctor attends Patient
Change Patient's status
Ask Patient to wait for questionnaires
Doctor submits questionnaire, prescription or remarks
Send Patient's answers and comment
Refresh the page periodically with time
Receive and display on Patient's screen
It passes Doctor's online time
Doctor completes the visit
Inform the Patient
Pray the Patient

Figure 5.15 Traces for Wait on Queue

5.1.16 Search Doctor

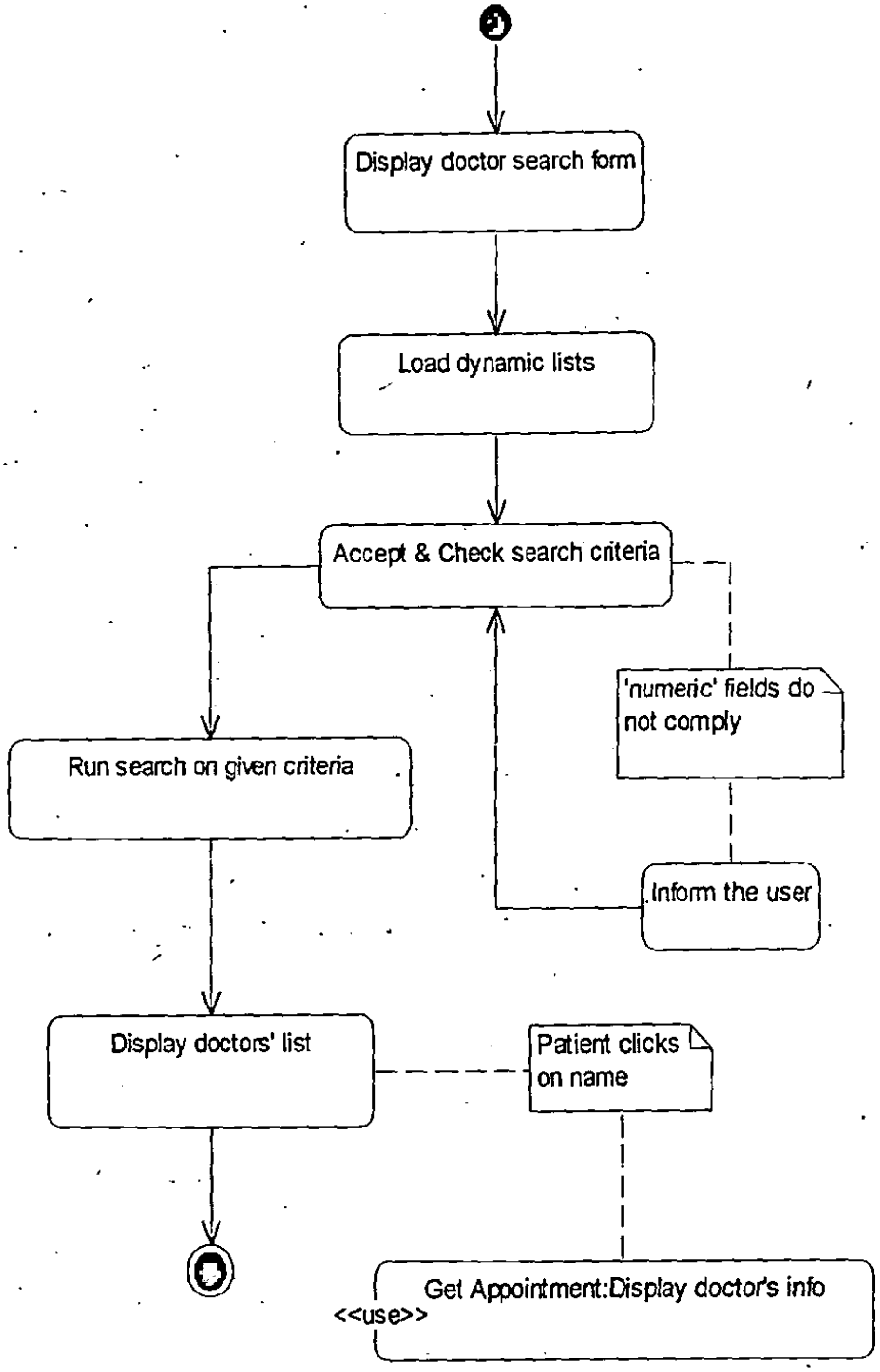

Display doctor search form
Load dynamic lists
Accept & Check search criteria
'numeric' fields do not comply
Inform the user
Run search on given criteria
Display doctors' list
Patient clicks on name
Get Appointment:Display doctor's info
<<use>>

Figure 5.12 Traces for Search Doctor

5.1.17 Search Service Provider

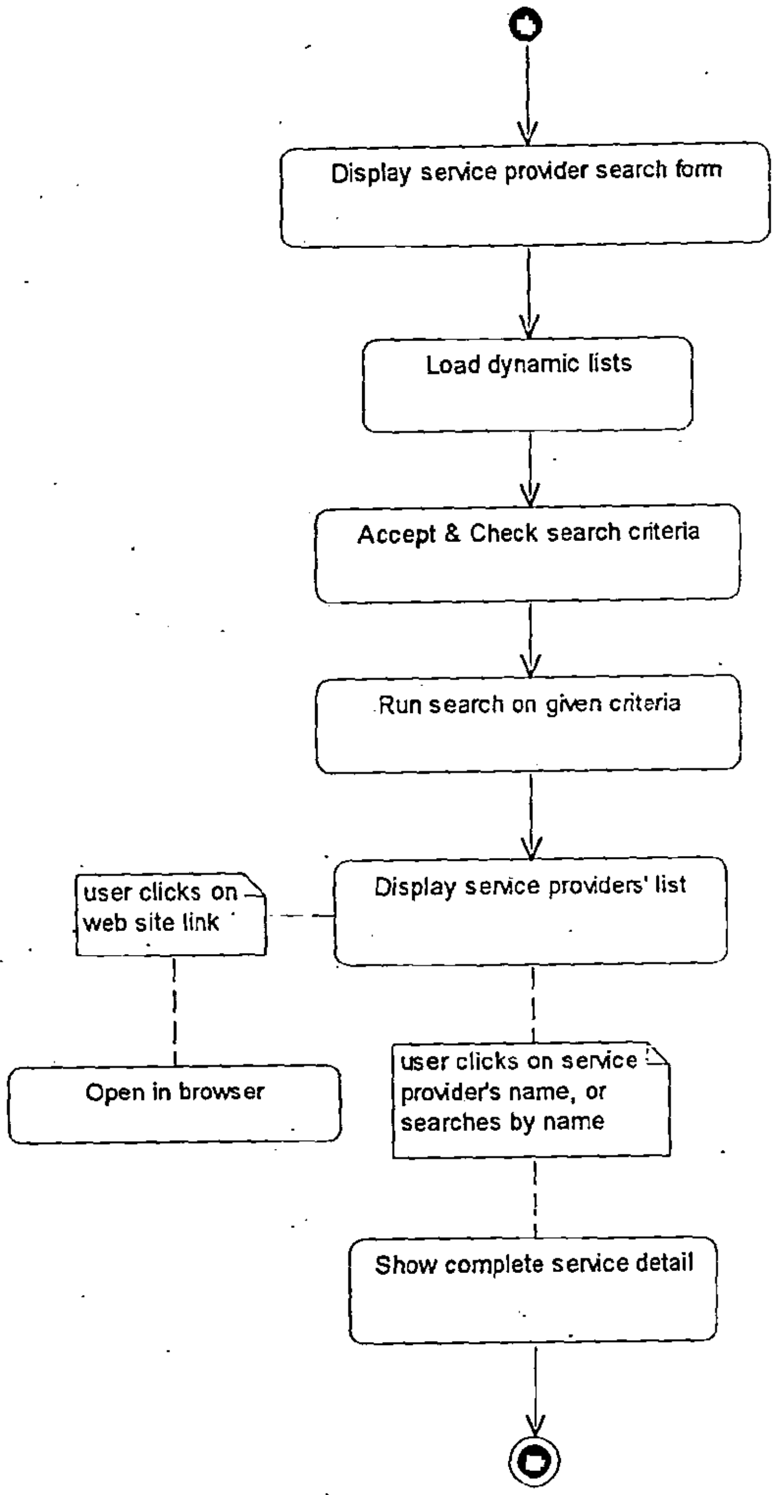

Display service provider search form
Load dynamic lists
Accept & Check search criteria
Run search on given criteria
Display service providers' list
user clicks on web site link
Open in browser
user clicks on service provider's name, or searches by name
Show complete service detail

Figure 5.17 Traces for Search Service Provider

5.1.18 Login as Service Provider

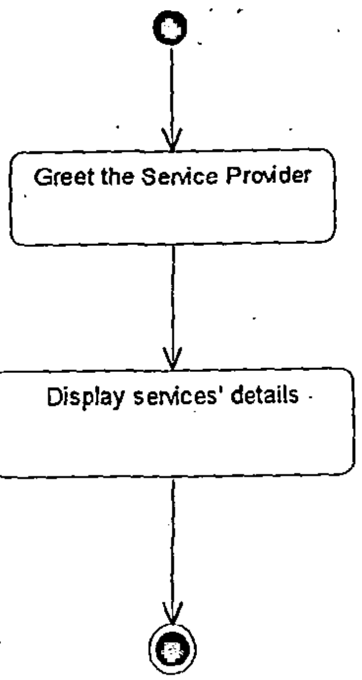


Figure 5.12 Traces for Login Procedure Specific to Service Provider

5.2 Sequence Diagrams

A sequence diagram represents the expanded form of use cases as a sequence of exchange of messages and parameters between different objects or elements, within the system. This way one can easily comprehend how different objects of the application coordinate among themselves while carrying out different functions, and in what sequence the tasks are performed to attain success. It describes the system in more technical and implemental terms. Following is each use case <<tracing>> its corresponding *sequence diagram.*

5.2.1 Display Urdu Text

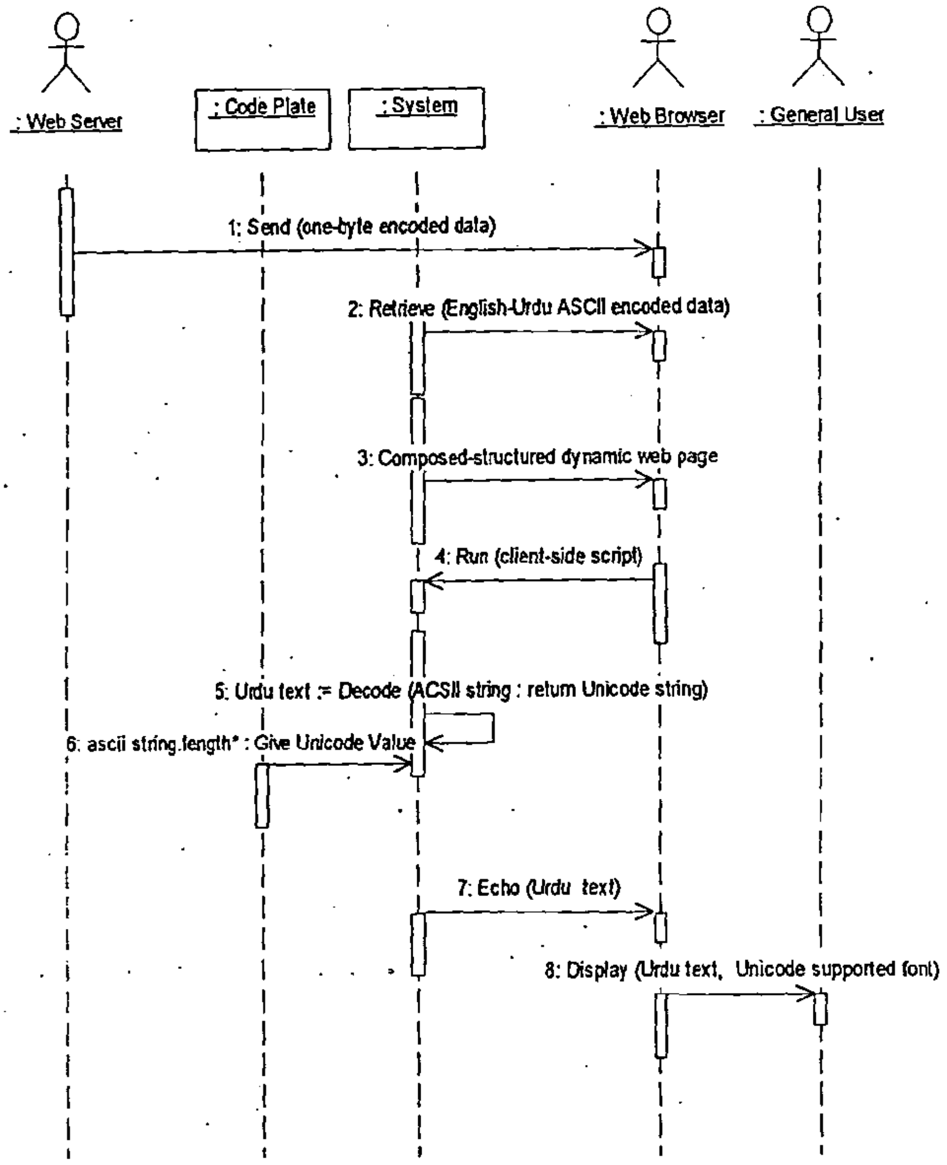

: Web Server
: Code Plate
: System
: Web Browser
: General User
1: Send (one-byte encoded data)
2: Retrieve (English-Urdu ASCII encoded data)
3: Composed-structured dynamic web page
4: Run (client-side script)
5: Urdu text := Decode (ACSII string : return Unicode string)
6: ascii string.length* : Give Unicode Value
7: Echo (Urdu text)
8: Display (Urdu text, Unicode supported font)

Figure 5.13 Sequence Diagram for Displaying Urdu Text

5.2.2 Input Urdu Text

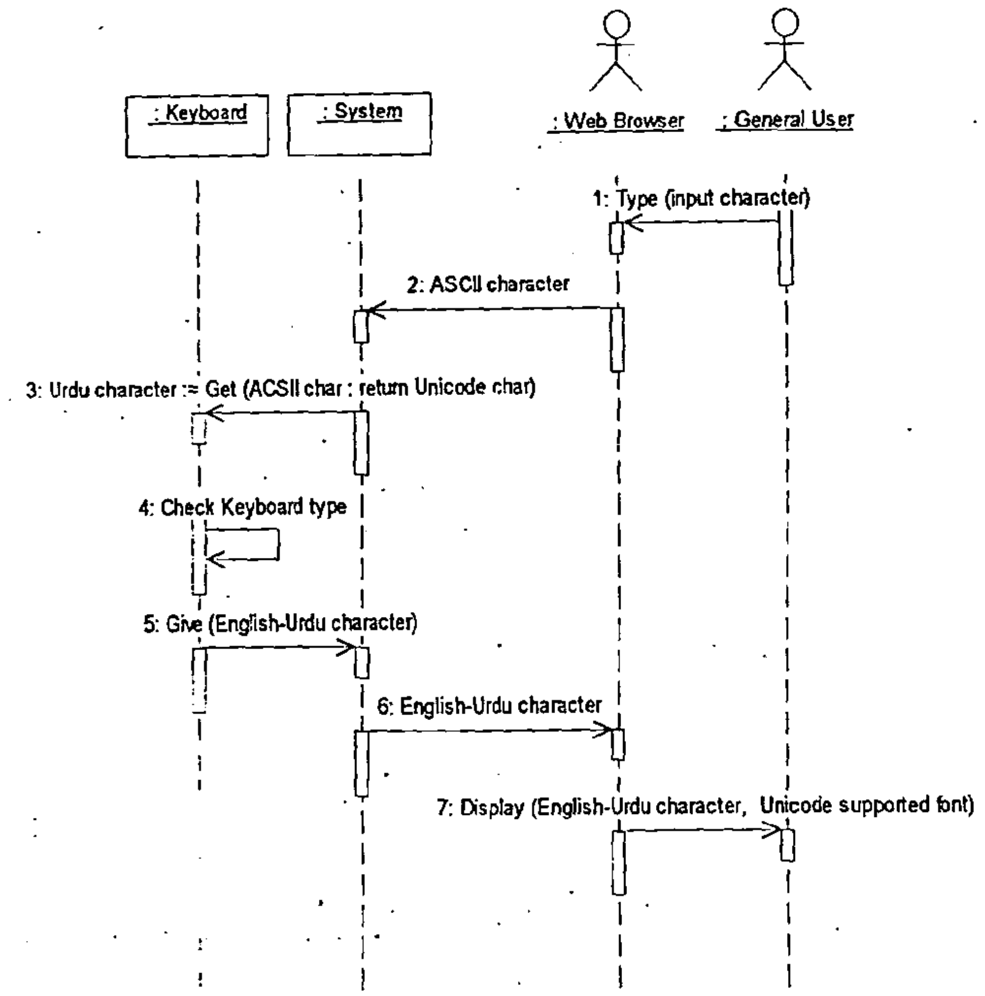

: Keyboard
: System
: Web Browser
: General User
1: Type (input character)
2: ASCII character
3: Urdu character := Get (ACSII char : return Unicode char)
4: Check Keyboard type
5: Give (English-Urdu character)
6: English-Urdu character
7: Display (English-Urdu character, Unicode supported font)

Figure 5.14 Sequence Diagram for Input Urdu and English Text

5.2.3 Send Form Data

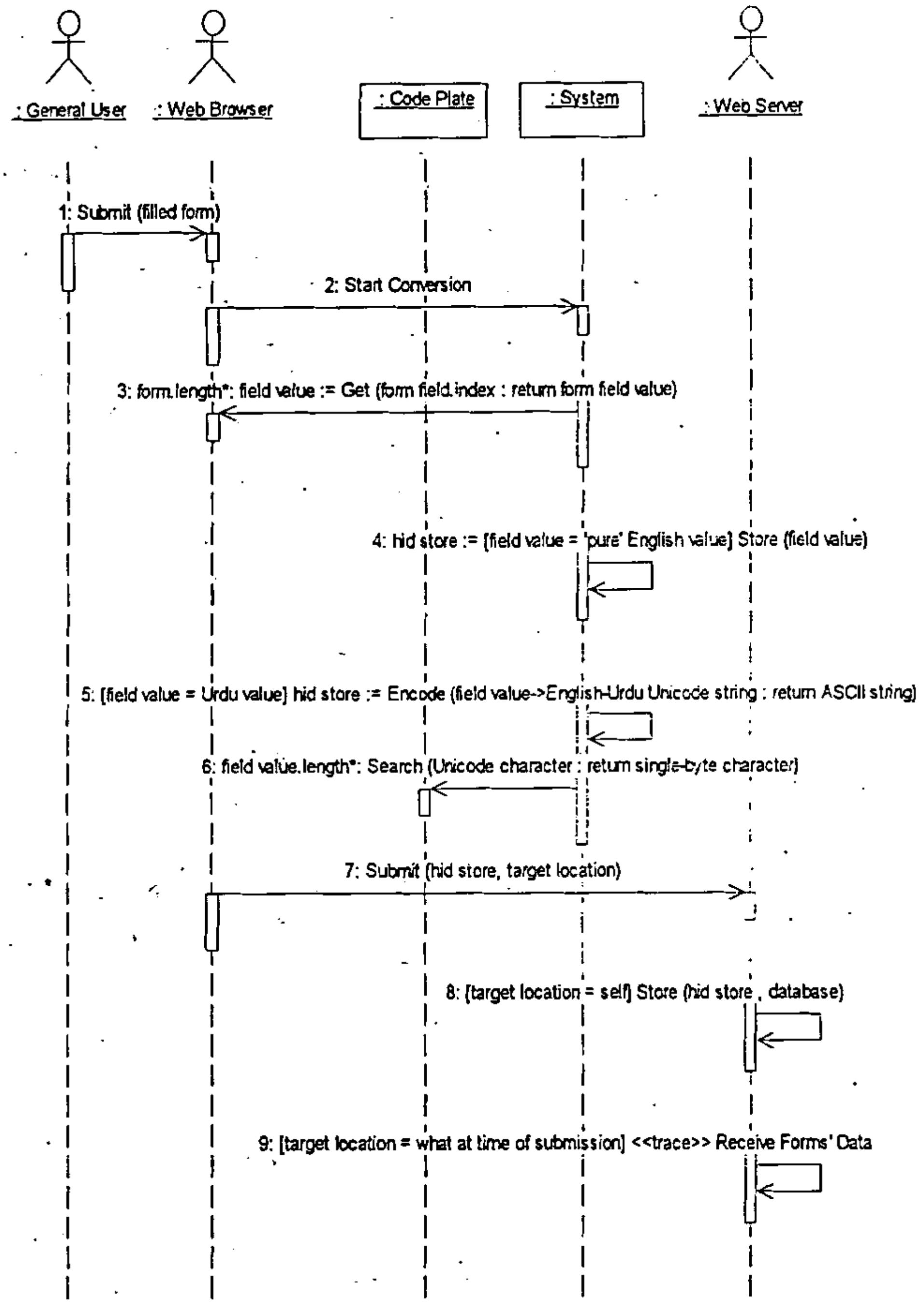

: General User
: Web Browser
: Code Plate
: System
: Web Server
1: Submit (filled form)
2: Start Conversion
3: form.length*: field value := Get (form field.index : return form field value)
4: hid store := [field value = 'pure' English value] Store (field value)
5: [field value = Urdu value] hid store := Encode (field value->English-Urdu Unicode string : return ASCII string)
6: field value.length*: Search (Unicode character : return single-byte character)
7: Submit (hid store, target location)
8: [target location = self] Store (hid store , database)
9: [target location = what at time of submission] <<trace>> Receive Forms' Data

Figure 5.15 Sequence Diagram for Sending Data of Web Forms

5.2.4 Receive Form Data

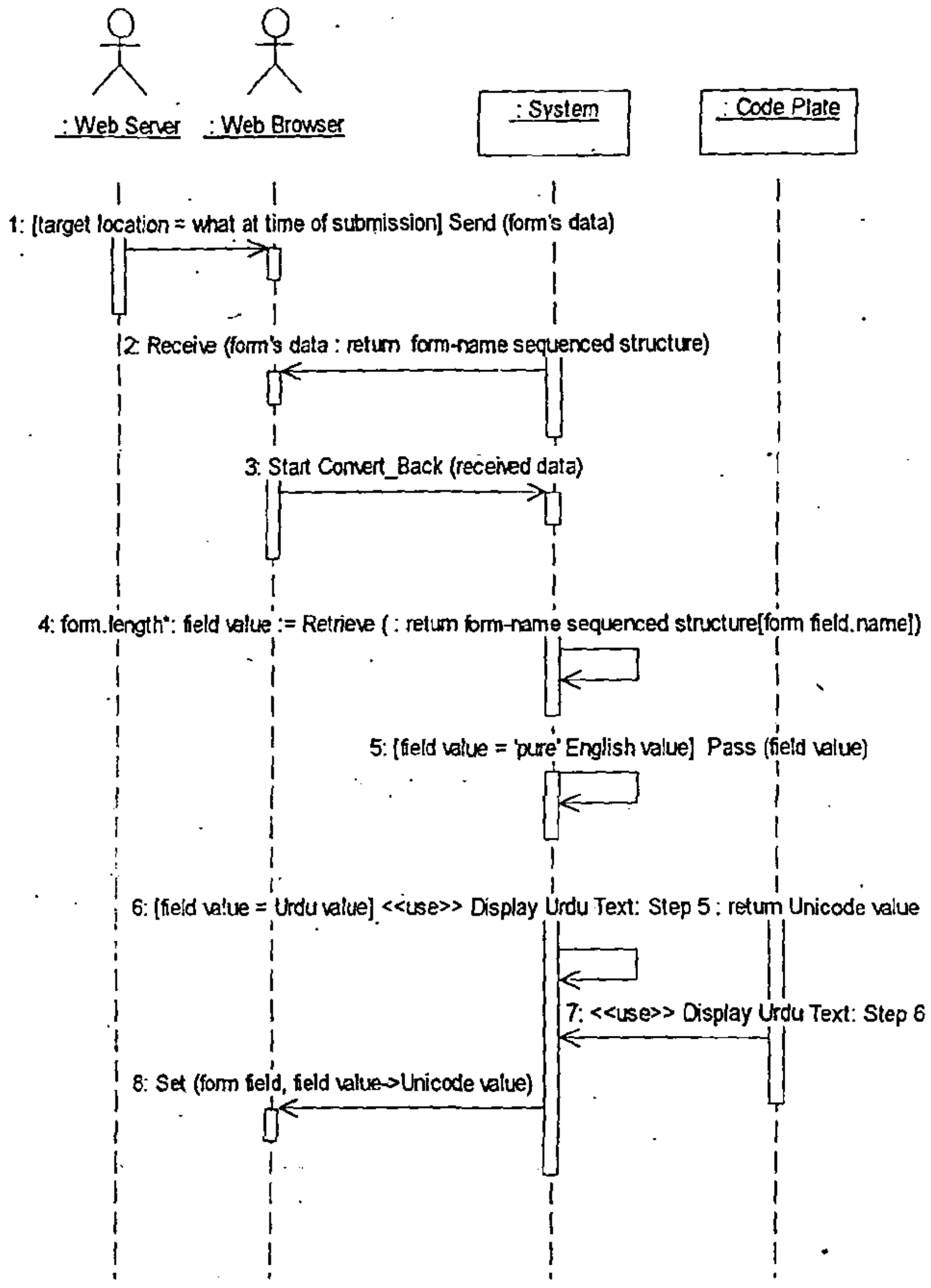

: Web Server
: Web Browser
: System
: Code Plate
1: [target location = what at time of submission] Send (form's data)
2: Receive (form's data : return form-name sequenced structure)
3: Start Convert_Back (received data)
4: form.length*: field value := Retrieve (: return form-name sequenced structure[form field.name])
5: [field value = 'pure' English value] Pass (field value)
6: [field value = Urdu value] <<use>> Display Urdu Text: Step 5 : return Unicode value
7: <<use>> Display Urdu Text: Step 6
8: Set (form field, field value->Unicode value)

Figure 5.15 Sequence Diagram for Receiving Web Form Data

5.2.5 Common Procedure of Signup

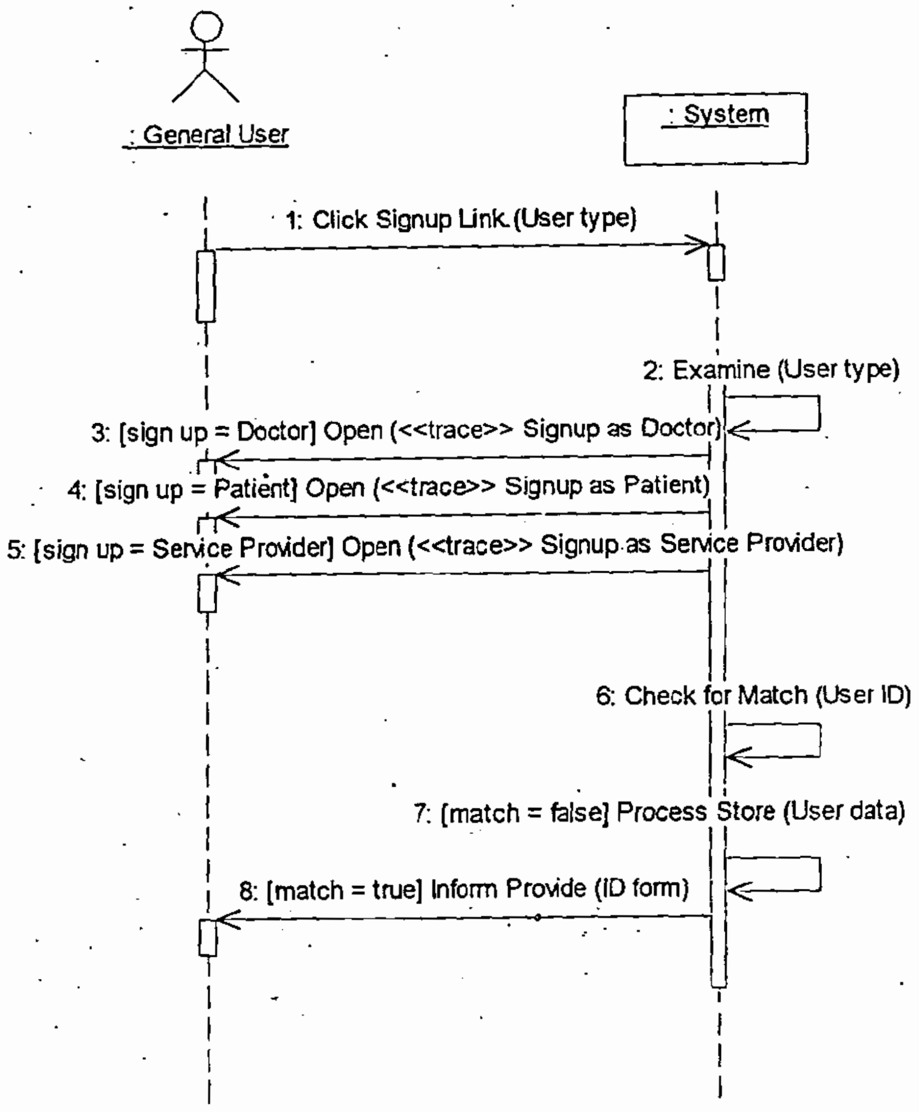

: General User
: System
1: Click Signup Link (User type)
2: Examine (User type)
3: [sign up = Doctor] Open (<<trace>> Signup as Doctor)
4: [sign up = Patient] Open (<<trace>> Signup as Patient)
5: [sign up = Service Provider] Open (<<trace>> Signup as Service Provider)
6: Check for Match (User ID)
7: [match = false] Process Store (User data)
8: [match = true] Inform Provide (ID form)

Figure 5.16 Sequence Diagram for Common Sigup Procedure

5.2.6 Signup as Doctor

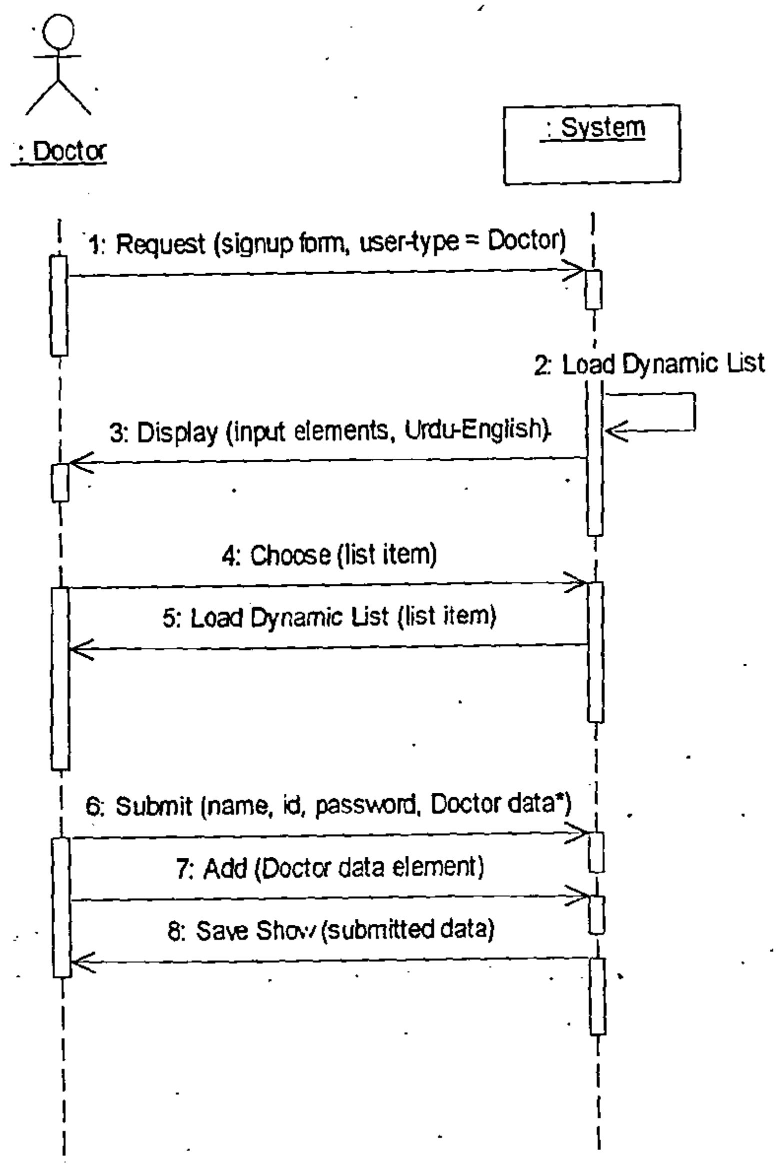

: Doctor
: System
1: Request (signup form, user-type = Doctor)
2: Load Dynamic List
3: Display (input elements, Urdu-English)
4: Choose (list item)
5: Load Dynamic List (list item)
6: Submit (name, id, password, Doctor data*)
7: Add (Doctor data element)
8: Save Show (submitted data)

Figure 5.17 Sequence Diagram for Doctor Specific Signup Procedure

5.2.7 Signup as Patient

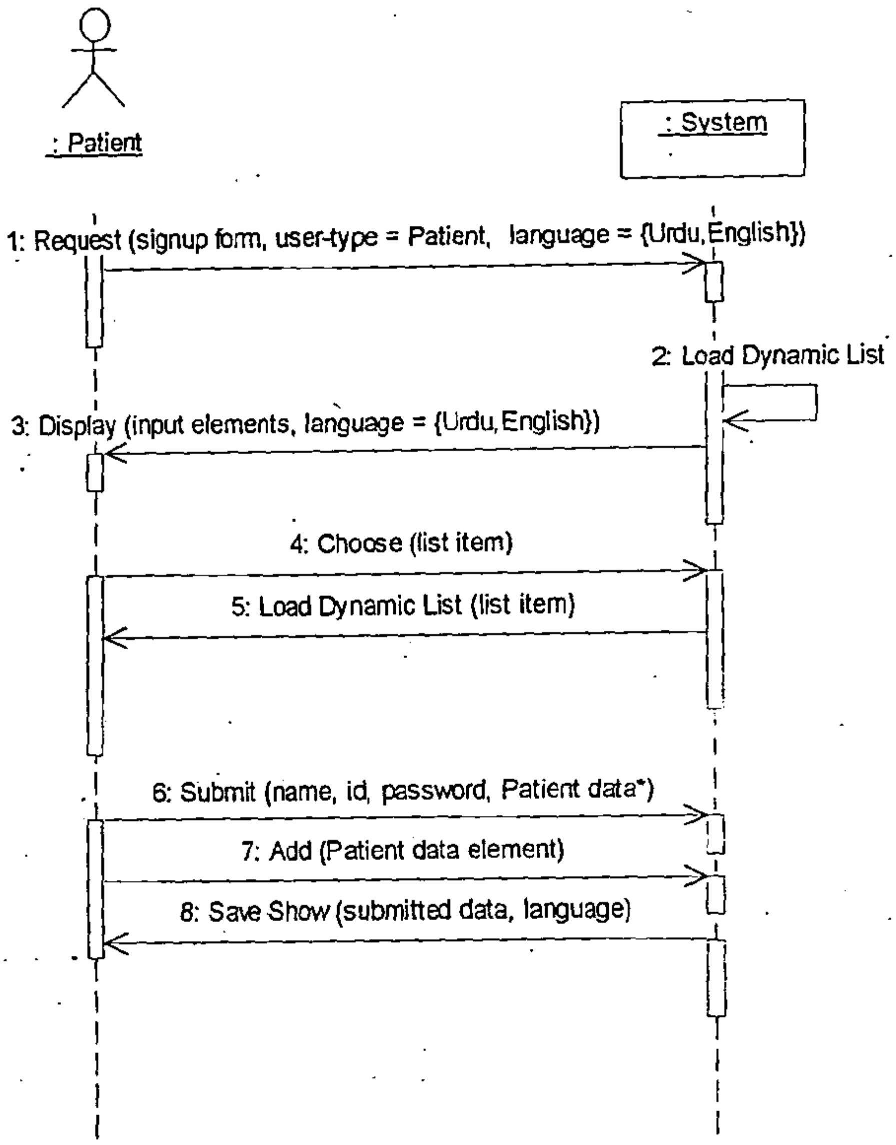

: Patient
: System
1: Request (signup form, user-type = Patient, language = {Urdu, English})
2: Load Dynamic List
3: Display (input elements, language = {Urdu, English})
4: Choose (list item)
5: Load Dynamic List (list item)
6: Submit (name, id, password, Patient data*)
7: Add (Patient data element)
8: Save Show (submitted data, language)

Figure 5.18 Sequence Diagram for Patient Specific Signup Procedure

5.2.8 Signup as Service Provider

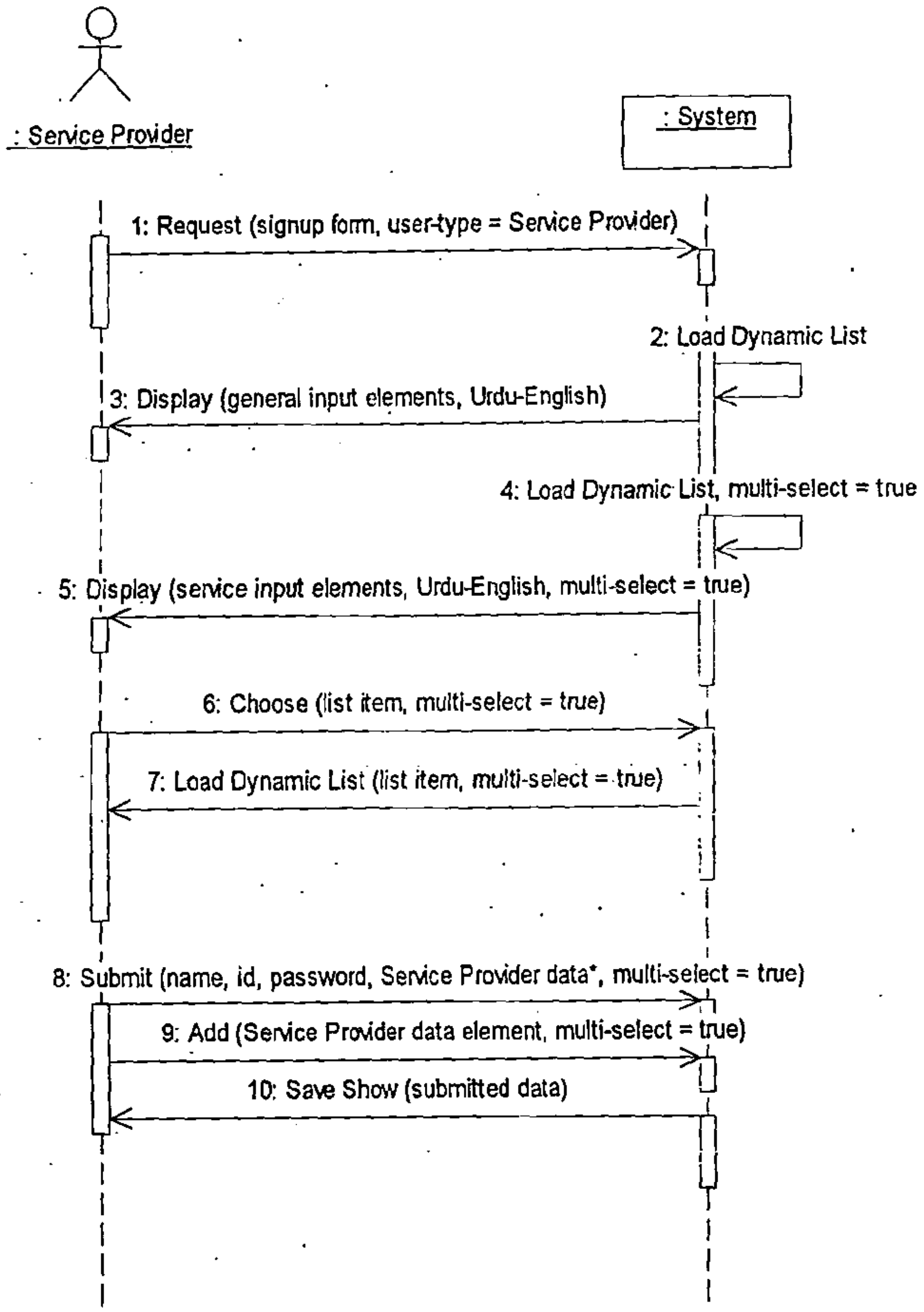

: Service Provider
: System
1: Request (signup form, user-type = Service Provider)
2: Load Dynamic List
3: Display (general input elements, Urdu-English)
4: Load Dynamic List, multi-select = true
5: Display (service input elements, Urdu-English, multi-select = true)
6: Choose (list item, multi-select = true)
7: Load Dynamic List (list item, multi-select = true)
8: Submit (name, id, password, Service Provider data*, multi-select = true)
9: Add (Service Provider data element, multi-select = true)
10: Save Show (submitted data)

Figure 5.19 Sequence Diagram for Service-Provider Specific Signup Procedure

5.2.9 Common Login Procedure

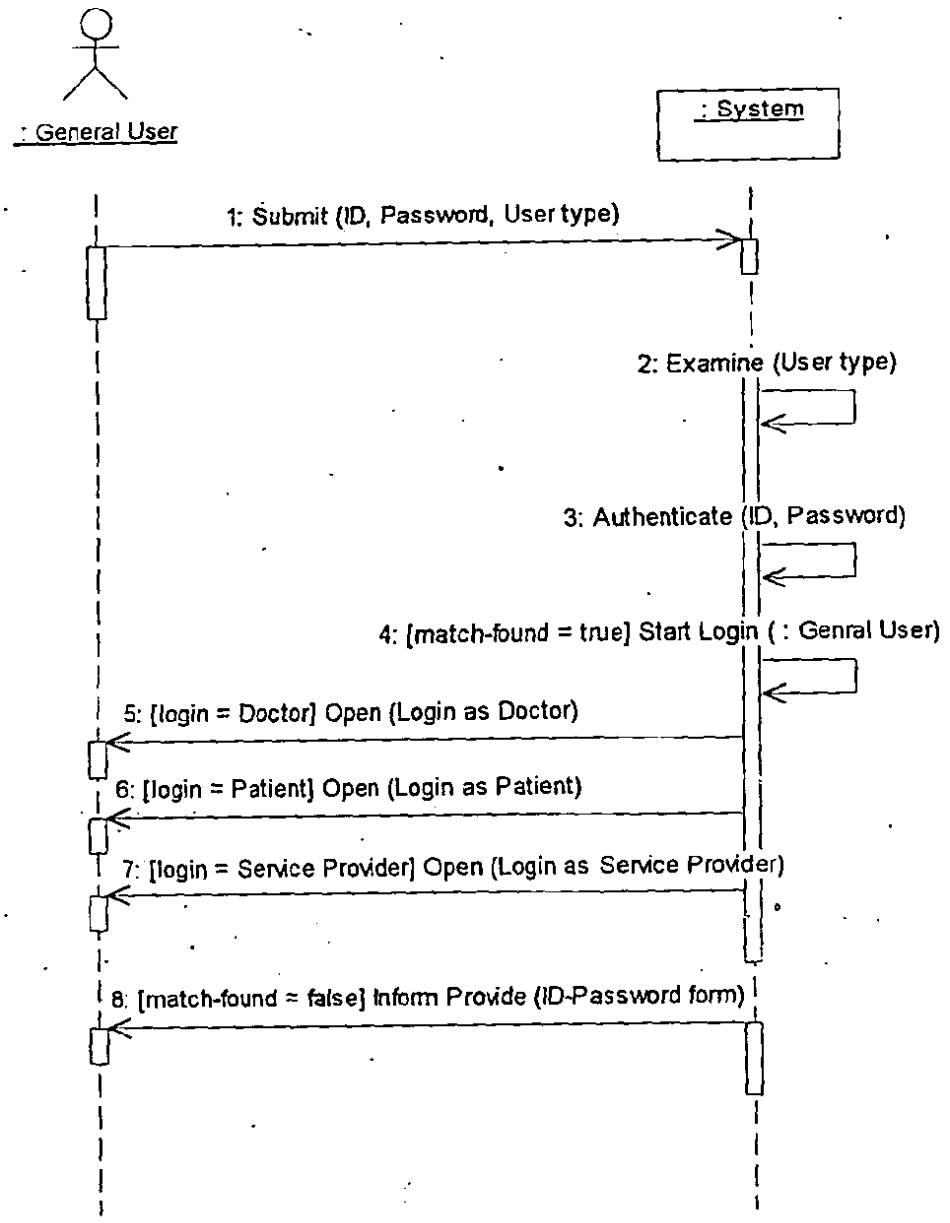

: General User
: System
1: Submit (ID, Password, User type)
2: Examine (User type)
3: Authenticate (ID, Password)
4: [match-found = true] Start Login (: Genral User)
5: [login = Doctor] Open (Login as Doctor)
6: [login = Patient] Open (Login as Patient)
7: [login = Service Provider] Open (Login as Service Provider)
8: [match-found = false] Inform Provide (ID-Password form)

Figure 5.20 Sequence Diagram for Login Procedure Common to All Users

5.2.10 Login as Doctor

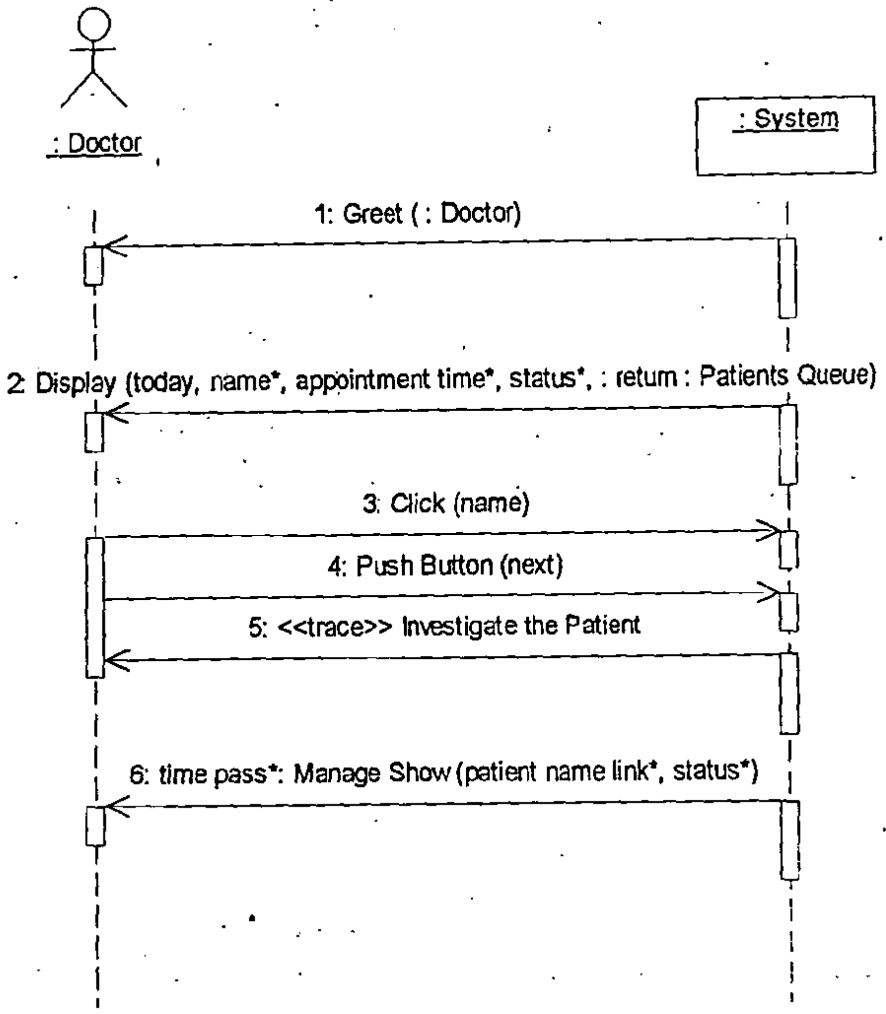

: Doctor
: System
1: Greet (: Doctor)
2: Display (today, name*, appointment time*, status*, : return : Patients Queue)
3: Click (name)
4: Push Button (next)
5: <<trace>> Investigate the Patient
6: time pass*: Manage Show (patient name link*, status*)

Figure 5.21 Sequence Diagram for Doctor Specific Login Procedure

5.2.11 Investigate the Patient

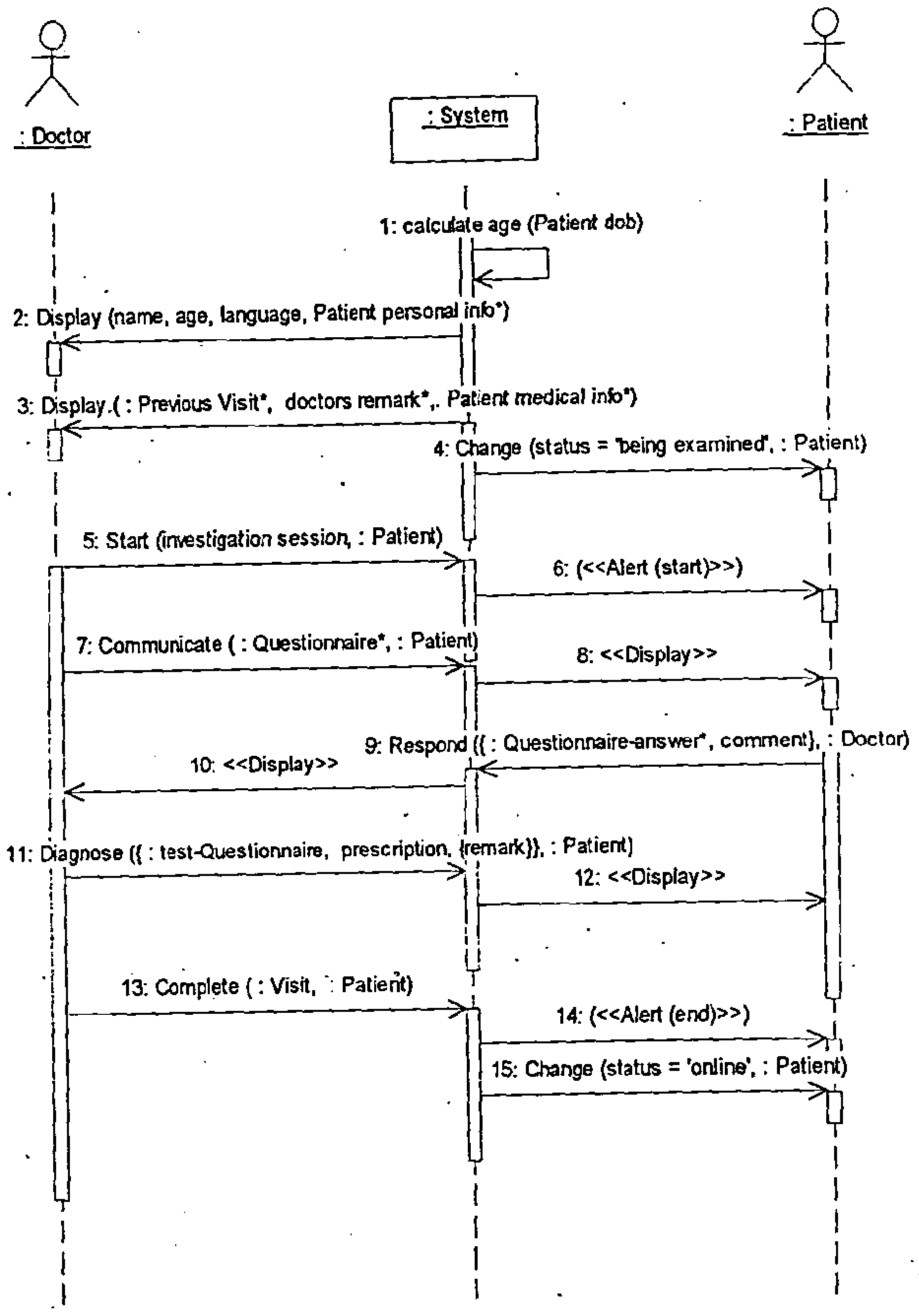

: Doctor
: System
: Patient
1: calculate age (Patient dob)
2: Display (name, age, language, Patient personal info*)
3: Display.(: Previous Visit*, doctors remark*,. Patient medical info*)
4: Change (status = 'being examined', : Patient)
5: Start (investigation session, : Patient)
6: (<<Alert (start)>>)
7: Communicate (: Questionnaire*, : Patient)
8: <<Display>>
9: Respond ({ : Questionnaire-answer*, comment}, : Doctor)
10: <<Display>>
11: Diagnose ({ : test-Questionnaire, prescription, {remark}}, : Patient)
12: <<Display>>
13: Complete (: Visit, : Patient)
14: (<<Alert (end)>>)
15: Change (status = 'online', : Patient)

Figure 5.22 Sequence Diagram for Patient Investigation Procedure

5.2.12 Design Questionnaire

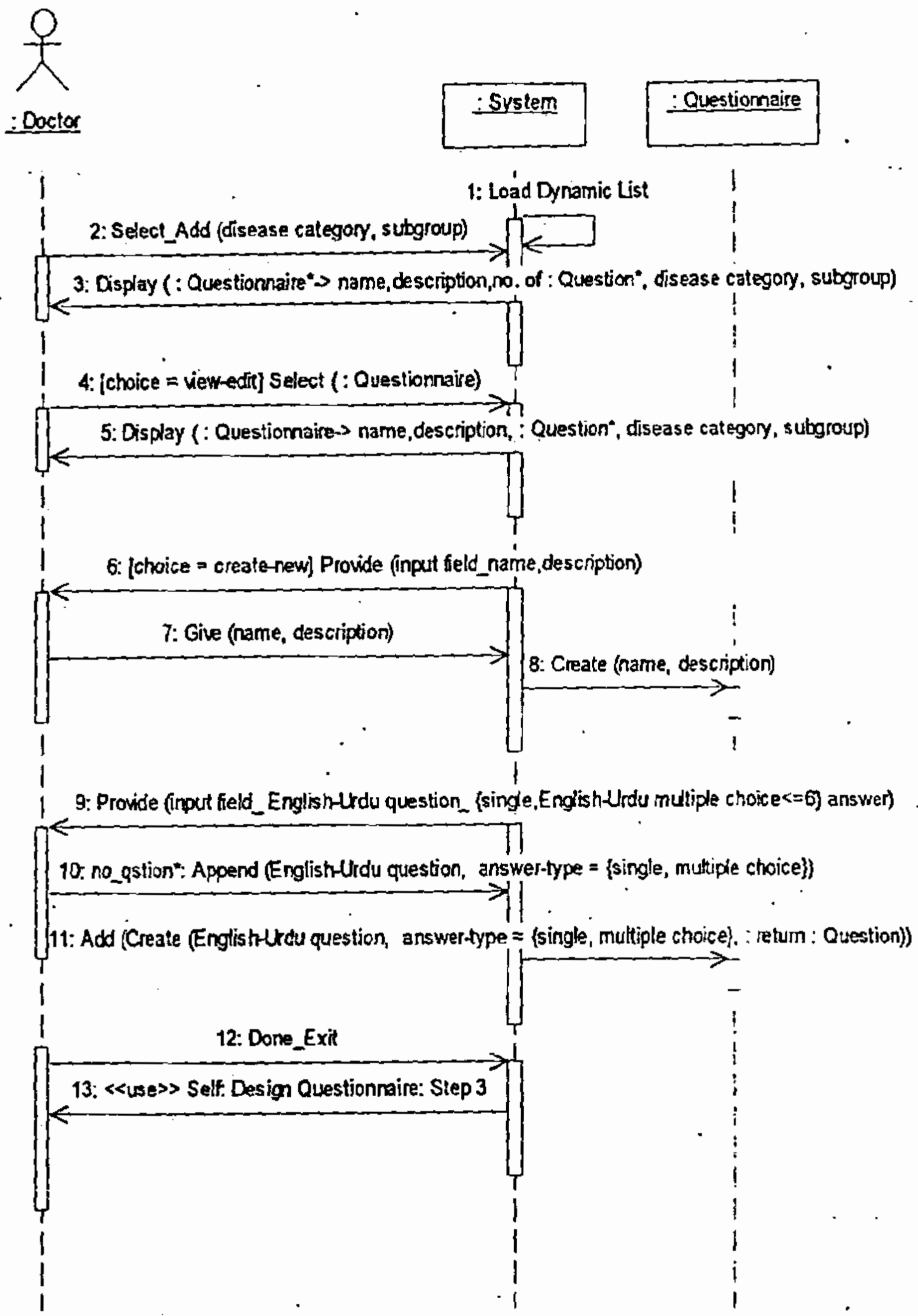

: Doctor
: System
: Questionnaire
1: Load Dynamic List
2: Select_Add (disease category, subgroup)
3: Display (: Questionnaire*-> name,description,no. of : Question*, disease category, subgroup)
4: [choice = view-edit] Select (: Questionnaire)
5: Display (: Questionnaire-> name,description, : Question*, disease category, subgroup)
6: [choice = create-new] Provide (input field_name,description)
7: Give (name, description)
8: Create (name, description)
9: Provide (input field_ English-Urdu question_ {single,English-Urdu multiple choice<=6} answer)
10: no_qstion*: Append (English-Urdu question, answer-type = {single, multiple choice})
11: Add (Create (English-Urdu question, answer-type = {single, multiple choice}, : return : Question))
12: Done_Exit
13: <<use>> Self: Design Questionnaire: Step 3

Figure 5.23 Sequence Diagram for Questionnaire Design Procedure

5.2.13 Login as Patient

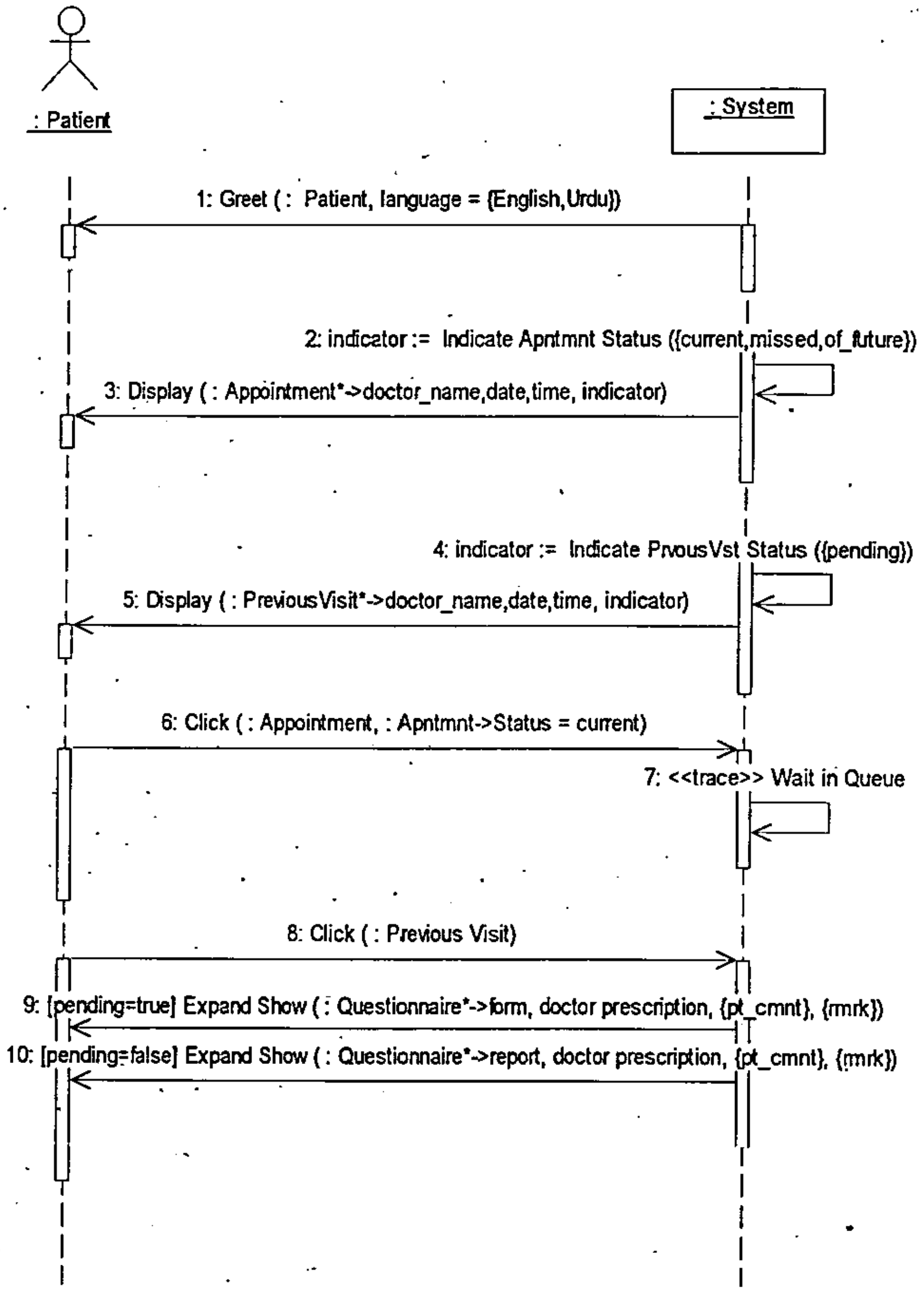

: Patient
: System
1: Greet (: Patient, language = {English,Urdu})
2: indicator := Indicate Apntmnt Status ({current,missed,of_future})
3: Display (: Appointment*->doctor_name,date,time, indicator)
4: indicator := Indicate PrvousVst Status ({pending})
5: Display (: PreviousVisit*->doctor_name,date,time, indicator)
6: Click (: Appointment, : Apntmnt->Status = current)
7: <<trace>> Wait in Queue
8: Click (: Previous Visit)
9: [pending=true] Expand Show (: Questionnaire*->form, doctor prescription, {pt_cmnt}, {rmrk})
10: [pending=false] Expand Show (: Questionnaire*->report, doctor prescription, {pt_cmnt}, {rmrk})

Figure 5.24 Sequence Diagram for Patient Specific Login Procedure

5.2.14 Get Appointment

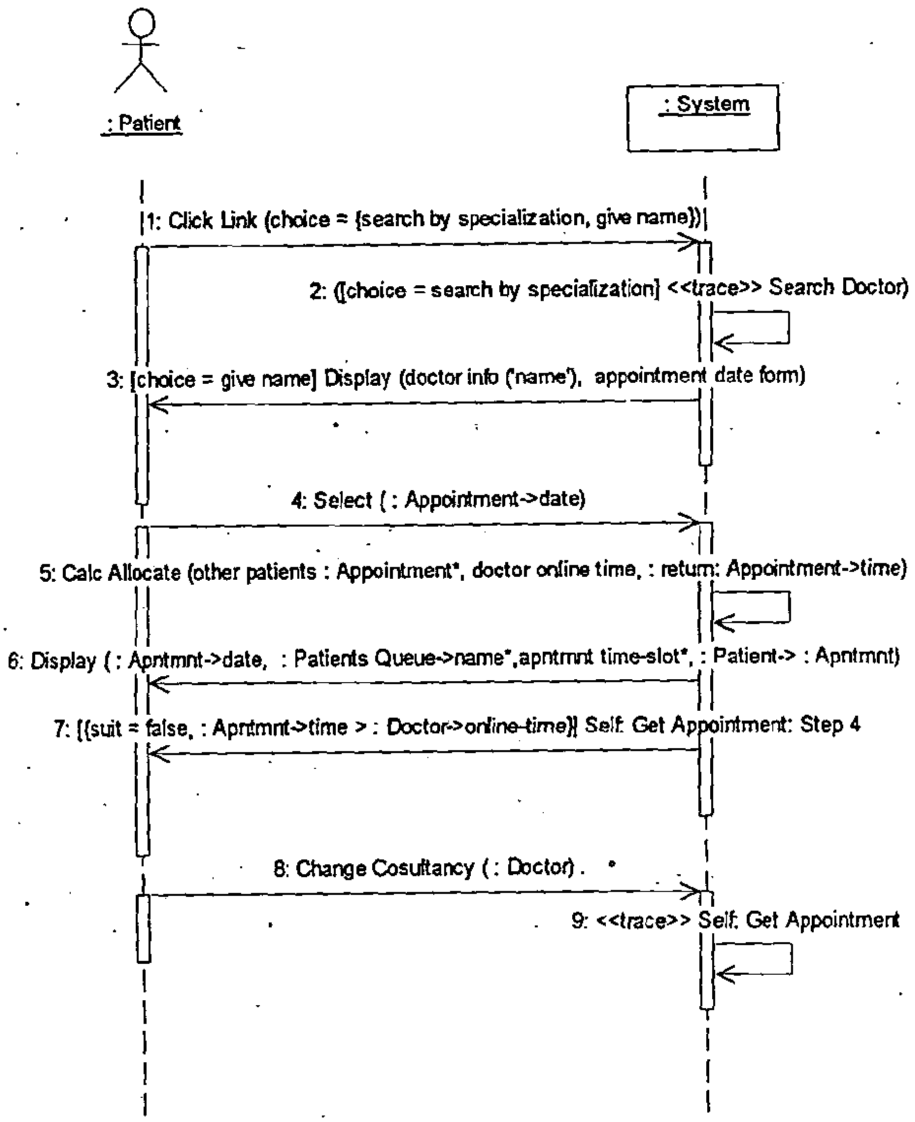

: Patient
: System
1: Click Link (choice = {search by specialization, give name})
2: ([choice = search by specialization] <<trace>> Search Doctor)
3: [choice = give name] Display (doctor info ('name'), appointment date form)
4: Select (: Appointment->date)
5: Calc Allocate (other patients : Appointment*, doctor online time, : return: Appointment->time)
6: Display (: Apntmnt->date, : Patients Queue->name*,apntmnt time-slot*, : Patient-> : Apntmnt)
7: [(suit = false, : Apntmnt->time > : Doctor->online-time)] Self: Get Appointment: Step 4
8: Change Cosultancy (: Doctor)
9: <<trace>> Self: Get Appointment

Figure 5.25 Sequence Procedure of Getting Appointment

5.2.15 Wait in Queue

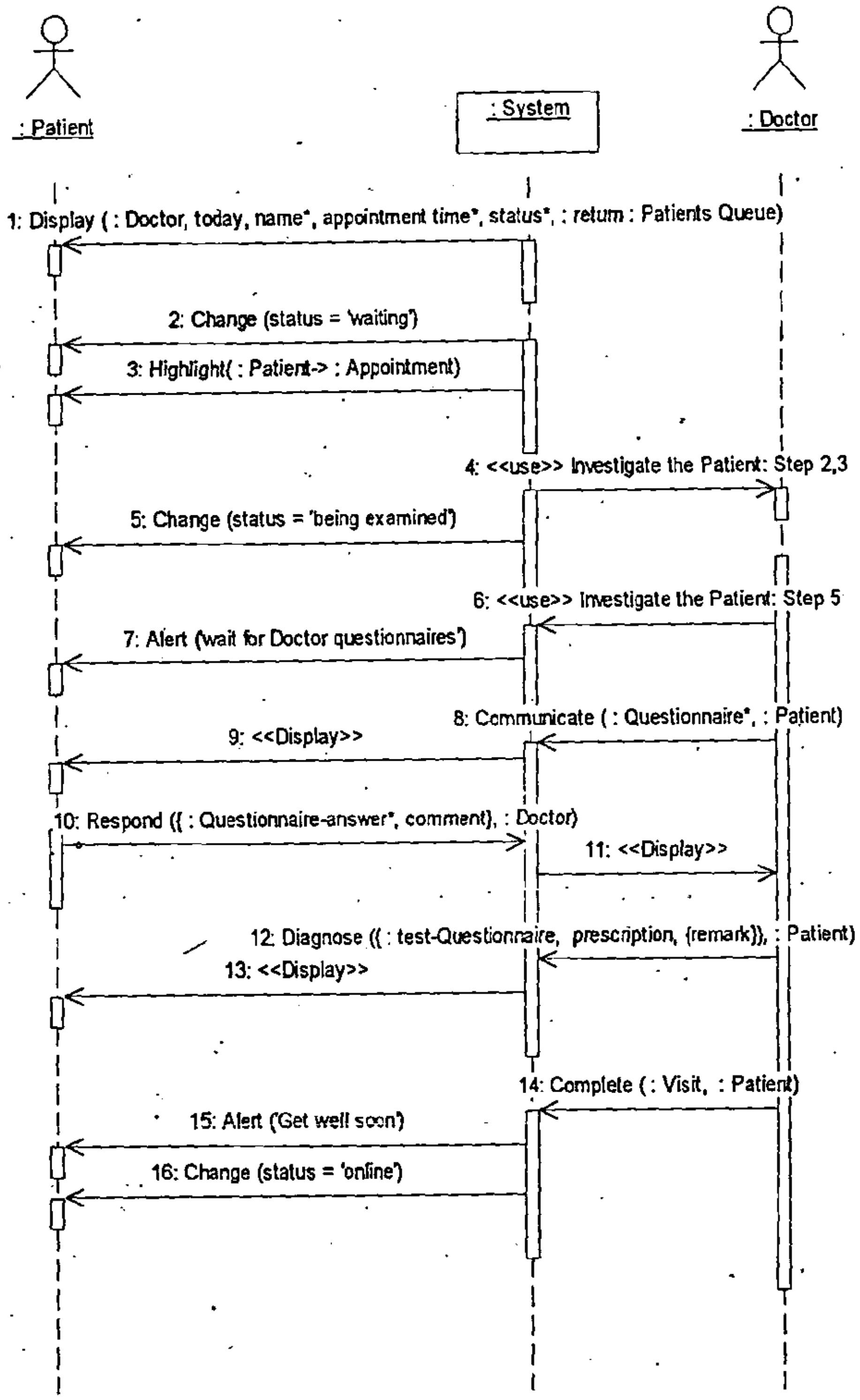

: Patient
: System
: Doctor
1: Display (: Doctor, today, name*, appointment time*, status*, : return : Patients Queue)
2: Change (status = 'waiting')
3: Highlight(: Patient-> : Appointment)
4: <<use>> Investigate the Patient: Step 2,3
5: Change (status = 'being examined')
6: <<use>> Investigate the Patient: Step 5
7: Alert ('wait for Doctor questionnaires')
8: Communicate (: Questionnaire*, : Patient)
9: <<Display>>
10: Respond ({ : Questionnaire-answer*, comment}, : Doctor)
11: <<Display>>
12: Diagnose ({ : test-Questionnaire, prescription, {remark}}, : Patient)
13: <<Display>>
14: Complete (: Visit, : Patient)
15: Alert ('Get well soon')
16: Change (status = 'online')

Figure 5.26 Sequence Diagram for Patient Waiting in Queue

5.2.16 Search Doctor

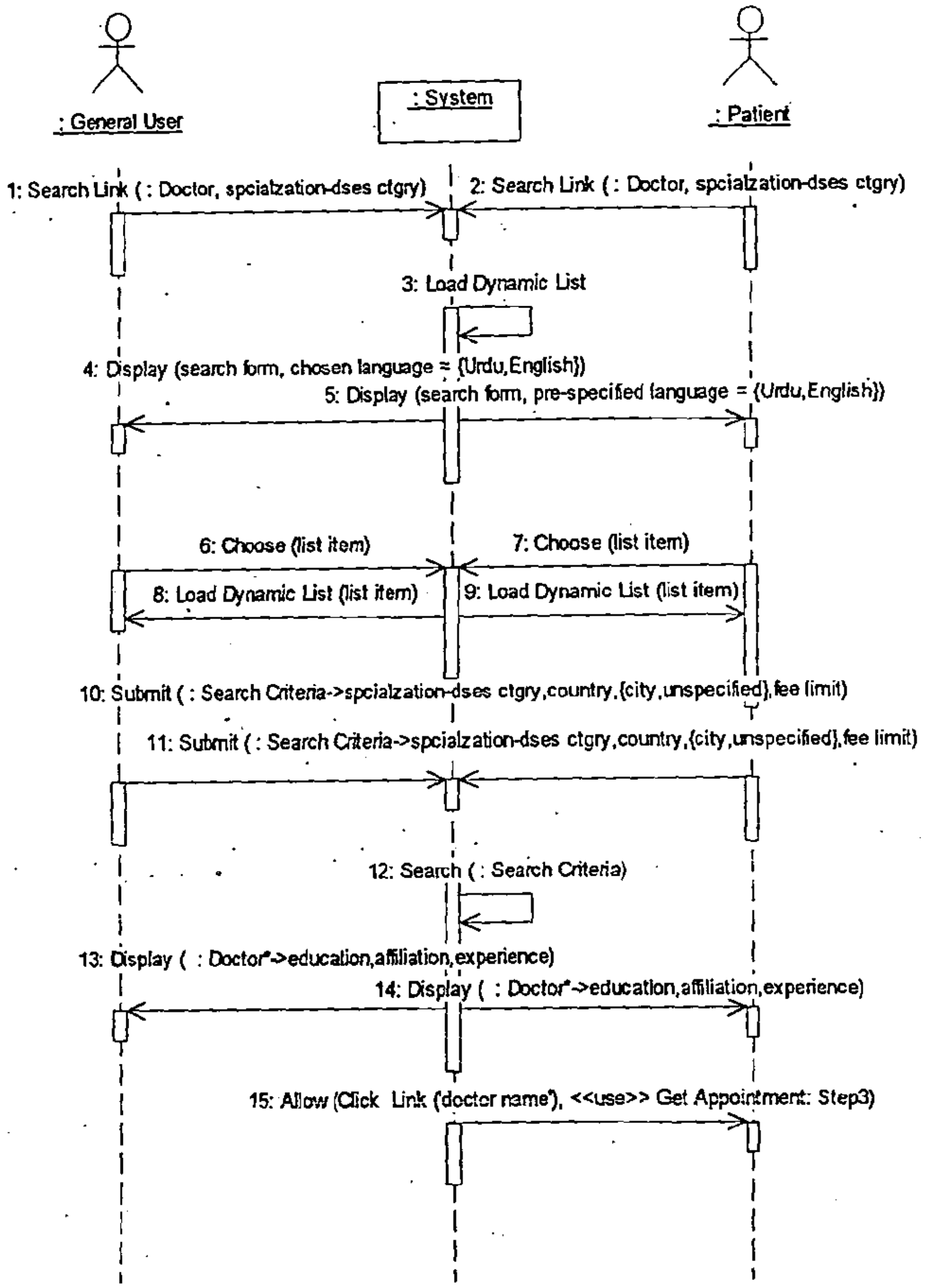

: General User
: System
: Patient
1: Search Link (: Doctor, spcialzation-dses ctgry)
2: Search Link (: Doctor, spcialzation-dses ctgry)
3: Load Dynamic List
4: Display (search form, chosen language = {Urdu,English})
5: Display (search form, pre-specified language = {Urdu,English})
6: Choose (list item)
7: Choose (list item)
8: Load Dynamic List (list item)
9: Load Dynamic List (list item)
10: Submit (: Search Criteria->spcialzation-dses ctgry,country,{city,unspecified},fee limit)
11: Submit (: Search Criteria->spcialzation-dses ctgry,country,{city,unspecified},fee limit)
12: Search (: Search Criteria)
13: Display (: Doctor*->education,affiliation,experience)
14: Display (: Doctor*->education,affiliation,experience)
15: Allow (Click Link ('doctor name'), <<use>> Get Appointment: Step3)

Figure 5.27 Sequence Diagram for Searching a Doctor

5.2.17 Search Service Provider

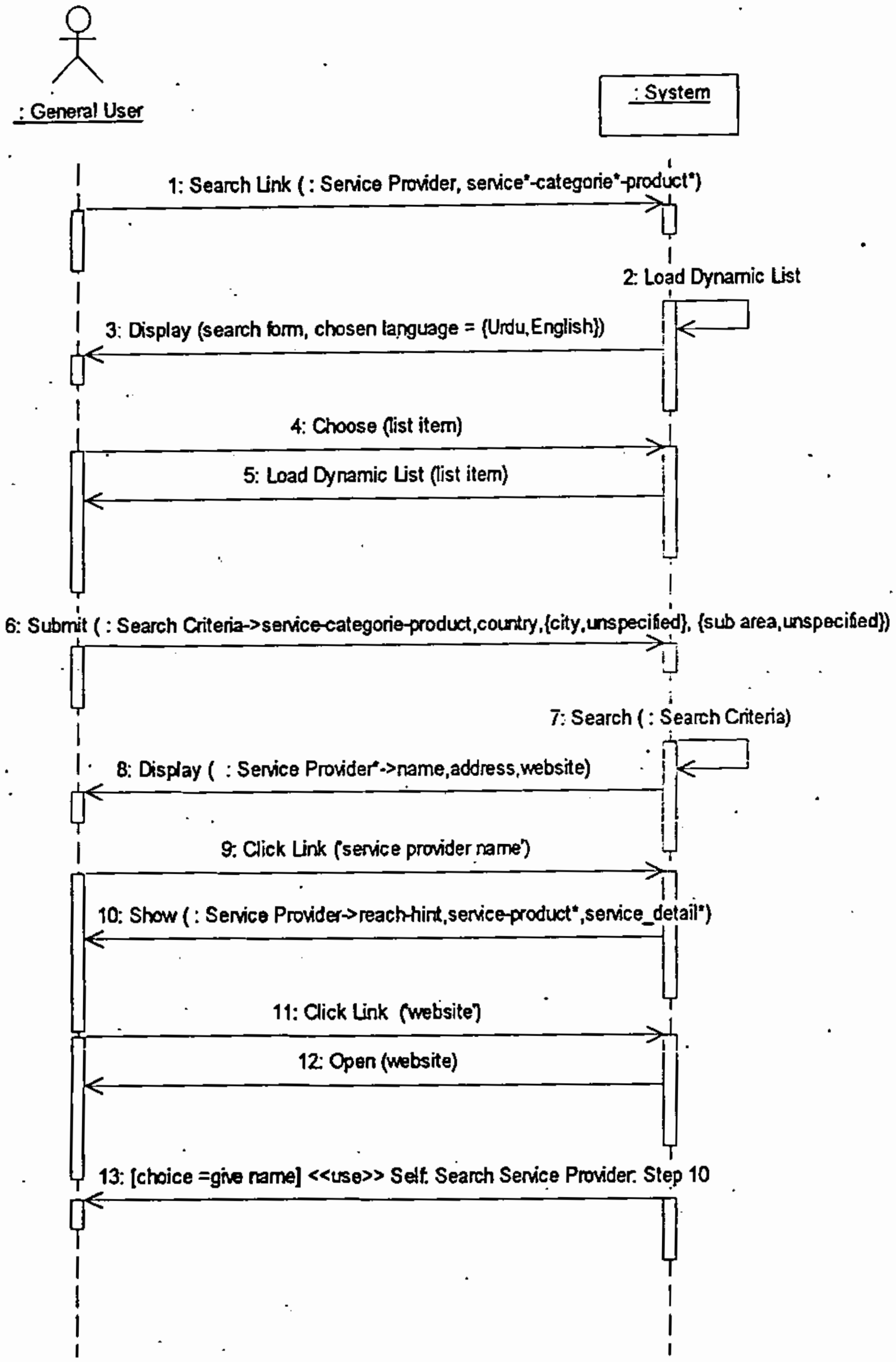
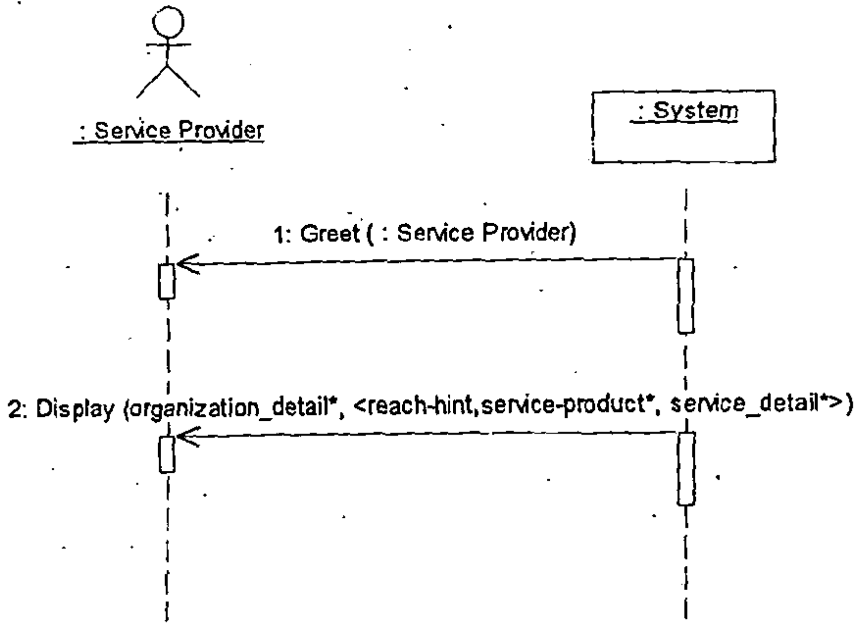

: General User
: System
1: Search Link (: Service Provider, service*-categorie*-product*)
2: Load Dynamic List
3: Display (search form, chosen language = {Urdu,English})
4: Choose (list item)
5: Load Dynamic List (list item)
6: Submit (: Search Criteria->service-categorie-product,country,{city,unspecified}, {sub area,unspecified})
7: Search (: Search Criteria)
8: Display (: Service Provider*->name,address,website)
9: Click Link ('service provider name')
10: Show (: Service Provider->reach-hint,service-product*,service_detail*)
11: Click Link ('website')
12: Open (website)
13: [choice =give name] <<use>> Self: Search Service Provider: Step 10

Figure 5.28 Sequence Diagram for Searching Service Provider

5.2.18 Login as Service Provider

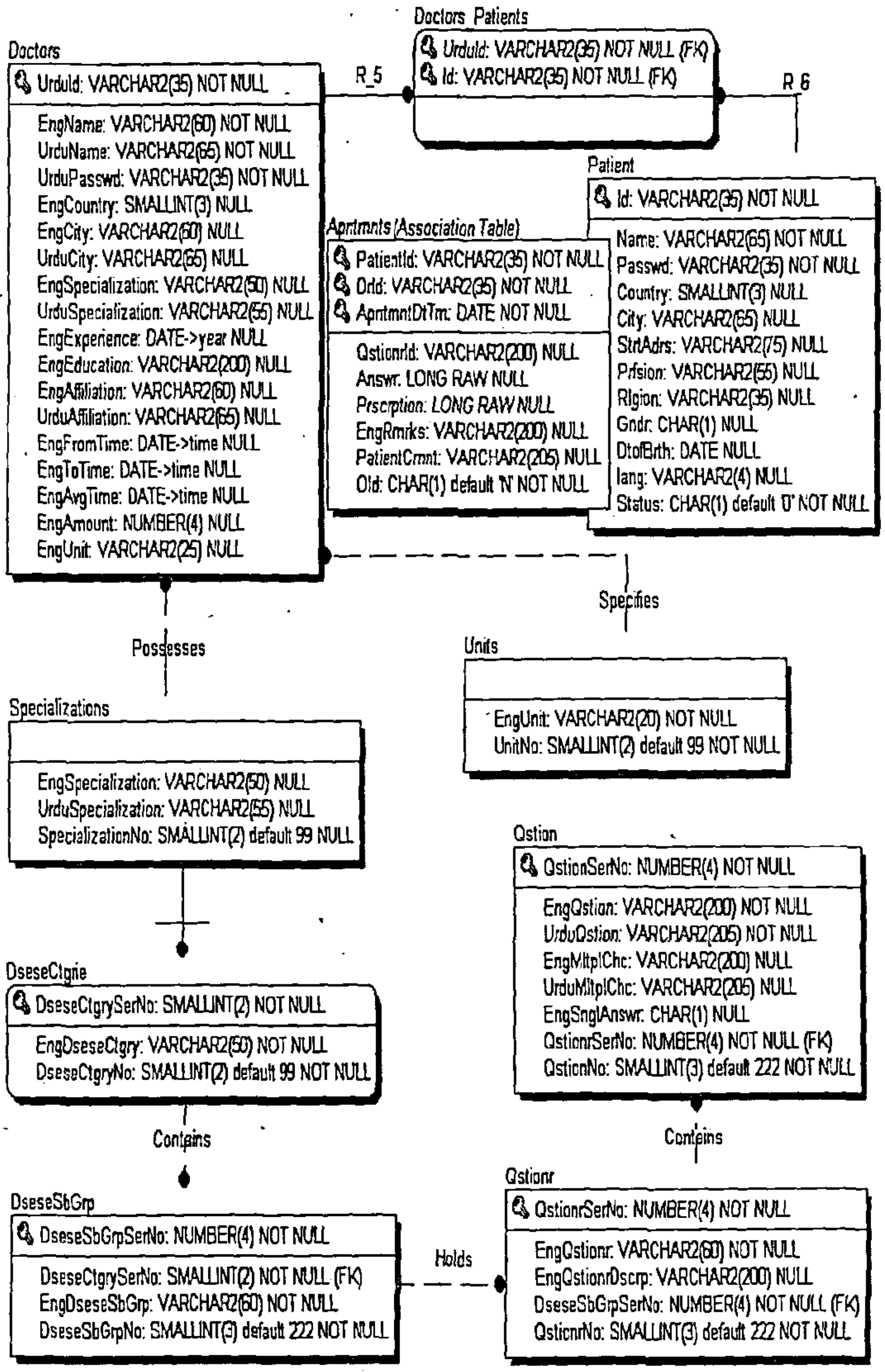


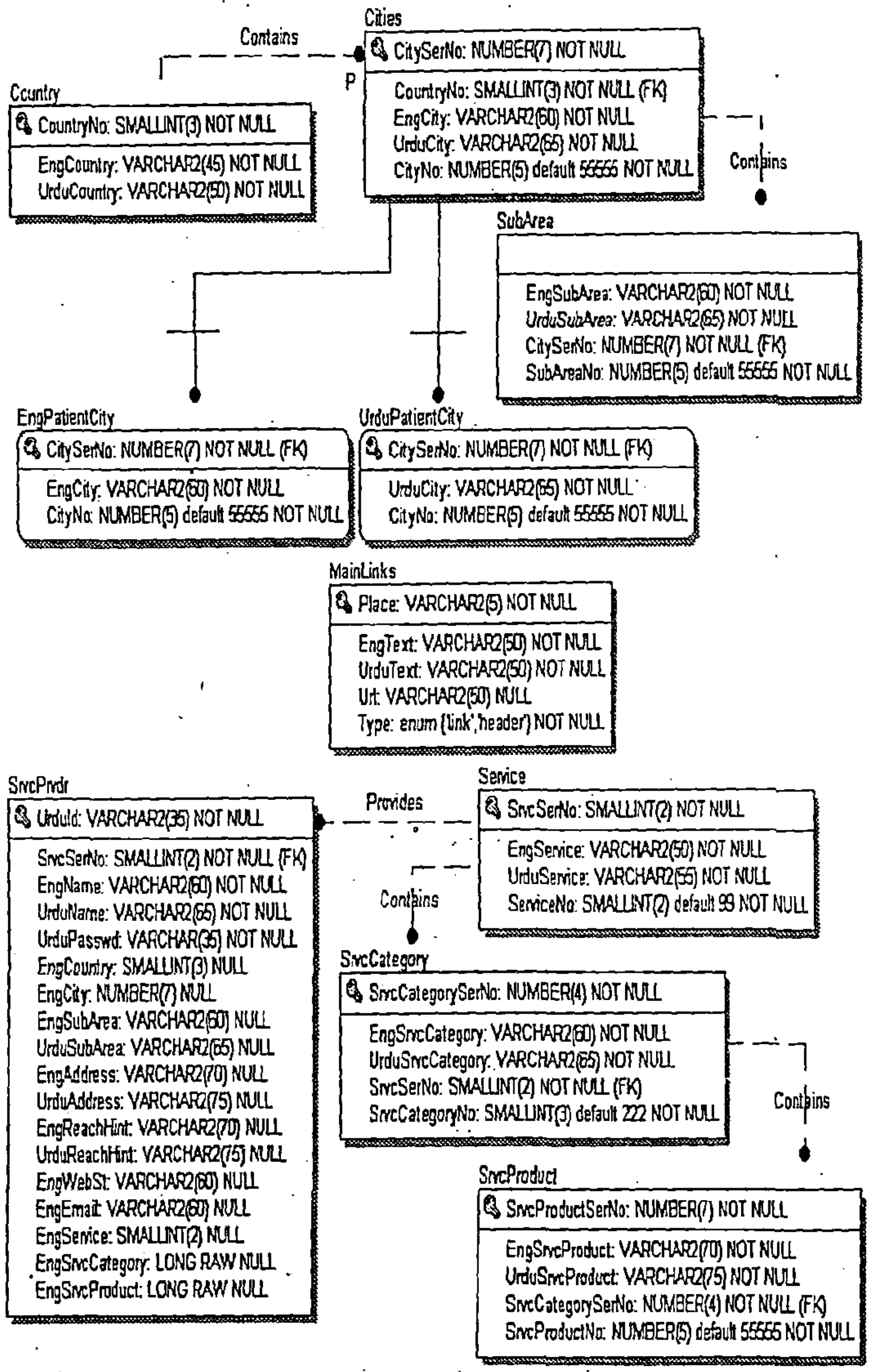
: Service Provider
: System
1: Greet (: Service Provider)
2: Display (organization_detail*, <reach-hint,service-product*, service_detail*>)

Figure 5.29 Sequence Diagram for Service-Provider Specific Login Procedure

5.3 Physical Model for Database

The physical model for database of the application under discussion is presented in pictorial form. Details of attributes of each table are visible. Similarly indication of primary, foreign, and composite key are shown pictorially. Some important information like whether an attribute can be left NULL *or not*, are also included in the diagram. The diagram being very large is spread over many continous pages.


Doctors Patients
UrduId: VARCHAR2(35) NOT NULL (FK)
Id: VARCHAR2(35) NOT NULL (FK)
R_5
R_6
Doctors
UrduId: VARCHAR2(35) NOT NULL
EngName: VARCHAR2(60) NOT NULL
UrduName: VARCHAR2(65) NOT NULL
UrduPasswd: VARCHAR2(35) NOT NULL
EngCountry: SMALLINT(3) NULL
EngCity: VARCHAR2(60) NULL
UrduCity: VARCHAR2(65) NULL
EngSpecialization: VARCHAR2(50) NULL
UrduSpecialization: VARCHAR2(55) NULL
EngExperience: DATE->year NULL
EngEducation: VARCHAR2(200) NULL
EngAffiliation: VARCHAR2(60) NULL
UrduAffiliation: VARCHAR2(65) NULL
EngFromTime: DATE->time NULL
EngToTime: DATE->time NULL
EngAvgTime: DATE->time NULL
EngAmount: NUMBER(4) NULL
EngUnit: VARCHAR2(25) NULL
Patient
Id: VARCHAR2(35) NOT NULL
Name: VARCHAR2(65) NOT NULL
Passwd: VARCHAR2(35) NOT NULL
Country: SMALLINT(3) NULL
City: VARCHAR2(65) NULL
StrAdrs: VARCHAR2(75) NULL
Prfsion: VARCHAR2(55) NULL
Rlgion: VARCHAR2(35) NULL
Gndr: CHAR(1) NULL
DtofBrth: DATE NULL
lang: VARCHAR2(4) NULL
Status: CHAR(1) default '0' NOT NULL
Apntmnts (Association Table)
PatientId: VARCHAR2(35) NOT NULL
DrId: VARCHAR2(35) NOT NULL
ApntmntDtTm: DATE NOT NULL
QstionrId: VARCHAR2(200) NULL
Answr: LONG RAW NULL
Prscrption: LONG RAW NULL
EngRmrks: VARCHAR2(200) NULL
PatientCmnt: VARCHAR2(205) NULL
Old: CHAR(1) default 'N' NOT NULL
Specifies
Possesses
Units
EngUnit: VARCHAR2(20) NOT NULL
UnitNo: SMALLINT(2) default 99 NOT NULL
Specializations
EngSpecialization: VARCHAR2(50) NULL
UrduSpecialization: VARCHAR2(55) NULL
SpecializationNo: SMALLINT(2) default 99 NULL
Qstion
QstionSerNo: NUMBER(4) NOT NULL
EngQstion: VARCHAR2(200) NOT NULL
UrduQstion: VARCHAR2(205) NOT NULL
EngMltplChc: VARCHAR2(200) NULL
UrduMltplChc: VARCHAR2(205) NULL
EngSnglAnswr: CHAR(1) NULL
QstionrSerNo: NUMBER(4) NOT NULL (FK)
QsticnNo: SMALLINT(3) default 222 NOT NULL
DseseCtgrie
DseseCtgrySerNo: SMALLINT(2) NOT NULL
EngDseseCtgry: VARCHAR2(50) NOT NULL
DseseCtgryNo: SMALLINT(2) default 99 NOT NULL
Contains
Contains
DseseSbGrp
DseseSbGrpSerNo: NUMBER(4) NOT NULL
DseseCtgrySerNo: SMALLINT(2) NOT NULL (FK)
EngDseseSbGrp: VARCHAR2(60) NOT NULL
DseseSbGrpNo: SMALLINT(3) default 222 NOT NULL
Holds
Qstionr
QstionrSerNo: NUMBER(4) NOT NULL
EngQstionr: VARCHAR2(60) NOT NULL
EngQstionrDscrp: VARCHAR2(200) NULL
DseseSbGrpSerNo: NUMBER(4) NOT NULL (FK)
QsticnrNo: SMALLINT(3) default 222 NOT NULL


Country
CountryNo: SMALLINT(3) NOT NULL
EngCountry: VARCHAR2(45) NOT NULL
UrduCountry: VARCHAR2(50) NOT NULL
Contains
P
Cities
CitySerNo: NUMBER(7) NOT NULL
CountryNo: SMALLINT(3) NOT NULL (FK)
EngCity: VARCHAR2(60) NOT NULL
UrduCity: VARCHAR2(65) NOT NULL
CityNo: NUMBER(5) default 55555 NOT NULL
Contains
SubArea
EngSubArea: VARCHAR2(60) NOT NULL
UrduSubArea: VARCHAR2(65) NOT NULL
CitySerNo: NUMBER(7) NOT NULL (FK)
SubAreaNo: NUMBER(5) default 55555 NOT NULL
EngPatientCity
CitySerNo: NUMBER(7) NOT NULL (FK)
EngCity: VARCHAR2(60) NOT NULL
CityNo: NUMBER(5) default 55555 NOT NULL
UrduPatientCity
CitySerNo: NUMBER(7) NOT NULL (FK)
UrduCity: VARCHAR2(65) NOT NULL
CityNo: NUMBER(5) default 55555 NOT NULL
MainLinks
Place: VARCHAR2(5) NOT NULL
EngText: VARCHAR2(50) NOT NULL
UrduText: VARCHAR2(50) NOT NULL
Url: VARCHAR2(50) NULL
Type: enum {'link','header'} NOT NULL
SrvcPrvdr
UrduId: VARCHAR2(35) NOT NULL
SrvcSerNo: SMALLINT(2) NOT NULL (FK)
EngName: VARCHAR2(60) NOT NULL
UrduName: VARCHAR2(65) NOT NULL
UrduPasswd: VARCHAR(35) NOT NULL
EngCountry: SMALLINT(3) NULL
EngCity: NUMBER(7) NULL
EngSubArea: VARCHAR2(60) NULL
UrduSubArea: VARCHAR2(65) NULL
EngAddress: VARCHAR2(70) NULL
UrduAddress: VARCHAR2(75) NULL
EngReachHint: VARCHAR2(70) NULL
UrduReachHint: VARCHAR2(75) NULL
EngWebSt: VARCHAR2(60) NULL
EngEmail: VARCHAR2(60) NULL
EngService: SMALLINT(2) NULL
EngSrvcCategory: LONG RAW NULL
EngSrvcProduct: LONG RAW NULL
Provides
Service
SrvcSerNo: SMALLINT(2) NOT NULL
EngService: VARCHAR2(50) NOT NULL
UrduService: VARCHAR2(55) NOT NULL
ServiceNo: SMALLINT(2) default 99 NOT NULL
Contains
SrvcCategory
SrvcCategorySerNo: NUMBER(4) NOT NULL
EngSrvcCategory: VARCHAR2(60) NOT NULL
UrduSrvcCategory: VARCHAR2(65) NOT NULL
SrvcSerNo: SMALLINT(2) NOT NULL (FK)
SrvcCategoryNo: SMALLINT(3) default 222 NOT NULL
Contains
SrvcProduct
SrvcProductSerNo: NUMBER(7) NOT NULL
EngSrvcProduct: VARCHAR2(70) NOT NULL
UrduSrvcProduct: VARCHAR2(75) NOT NULL
SrvcCategorySerNo: NUMBER(4) NOT NULL (FK)
SrvcProductNo: NUMBER(5) default 55555 NOT NULL

Chapter 6

Experimentation

6. Experimentation

This chapter will present the experimentation of the designed algorithms by implementing them in a complete web application. This will begin by first looking at how different parts and parcels of the system are connected and collaborate to each other, and how different components are organized. Up to this level the experimental application will be shown in form of *component diagrams.* Subsequently, each component will be discussed separately to capture how these components in themselves conform to the analysis and design theme of the system. Each component is elaborated by giving the 'course of events' taken from the previously conducted research process and followed by the code that implements the developed algorithm. Any alternative courses taken in case of errors and user faults are described separately for better readability.

6.1 Component Diagrams

A component diagram describes the organization and structuring of all the system segments. This helps understanding of system by explaining the sequence between these parts and segments and the flow of control among them. As the experimental application is a web based program, the component diagram also serves the purpose of a *site-map.* Following sections present component diagrams of the modules developed.

6.1.1 Multilingual Interface

This component is <<included>> in all other components. It provides initialization tasks to set up the multilingual interfaces in Urdu and English. It is implemented in three files, namely start.php, mndyrlang.php, and end.php. The last file, as the name reflects, contains code to clean up the settings done by the code. Figure 6.1 shows the relationship between these files.

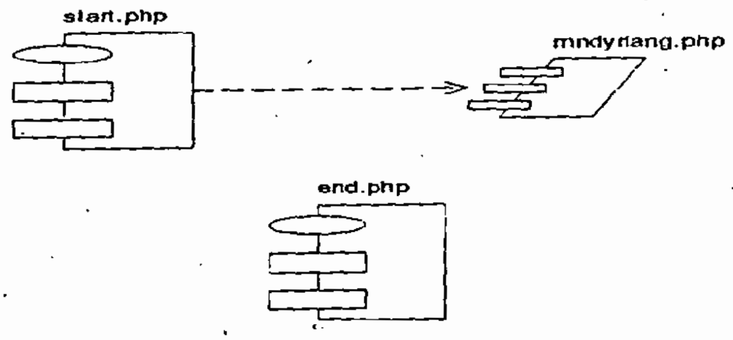


Figure 6.1 Setup Component for Multiligual Interface

6.1.2 Code Plate and Selectable Keyboard

This componenent is related to keyboard management allowing to select a keyboard layout at run time. It is implemented in many files and it also interacts with the code plate component for assigning character codes to the punched key strokes. Figure 6.2 shows this system in diagrammatic form.

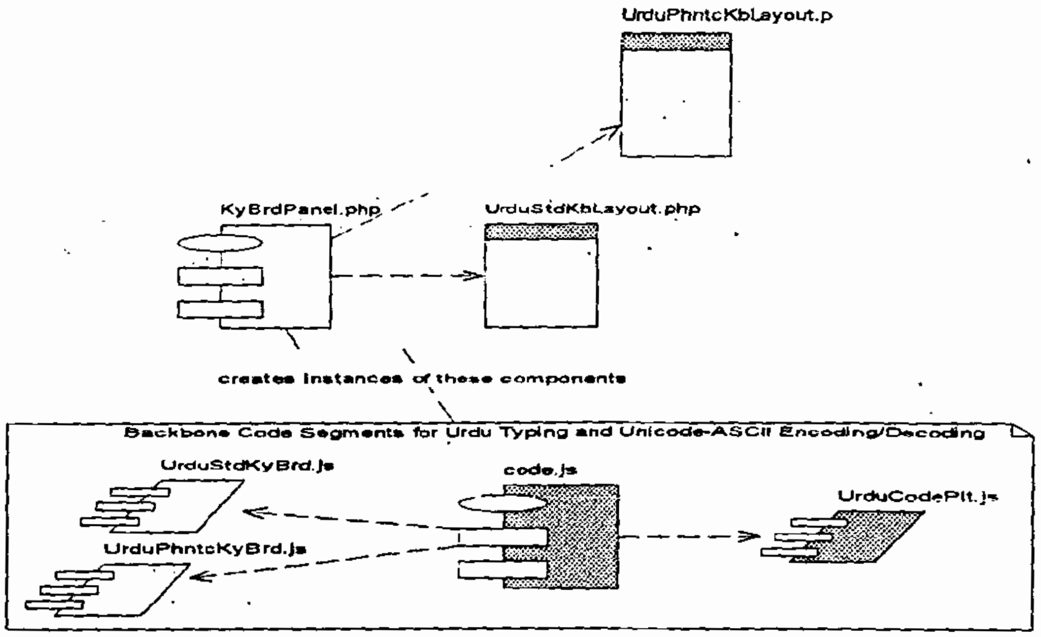

UrduPhntcKbLayout.p
KyBrdPanel.php
UrduStdKbLayout.php
creates instances of these components
Backbone Code Segments for Urdu Typing and Unicode-ASCII Encoding/Decoding
UrduStdKyBrd.js
code.js
UrduCodePlt.js
UrduPhntcKyBrd.js

Figure 6.2 Code Plate and Multiple Keyboads

6.1.3 Signup Process

This component includes the features that different types of users require for registering themselves in the system. Sign up process of each user is according the nature of tasks that will be performed by him / her. This nature of tasks has been dicussed in detail in the design process. Requrements of each user are separately coded in distinct files. As the data during the signup process has to be saved in the database, hence the database management code is also introduced in this component. Figure 6.3 shows the files and their interaction in the experimental application.

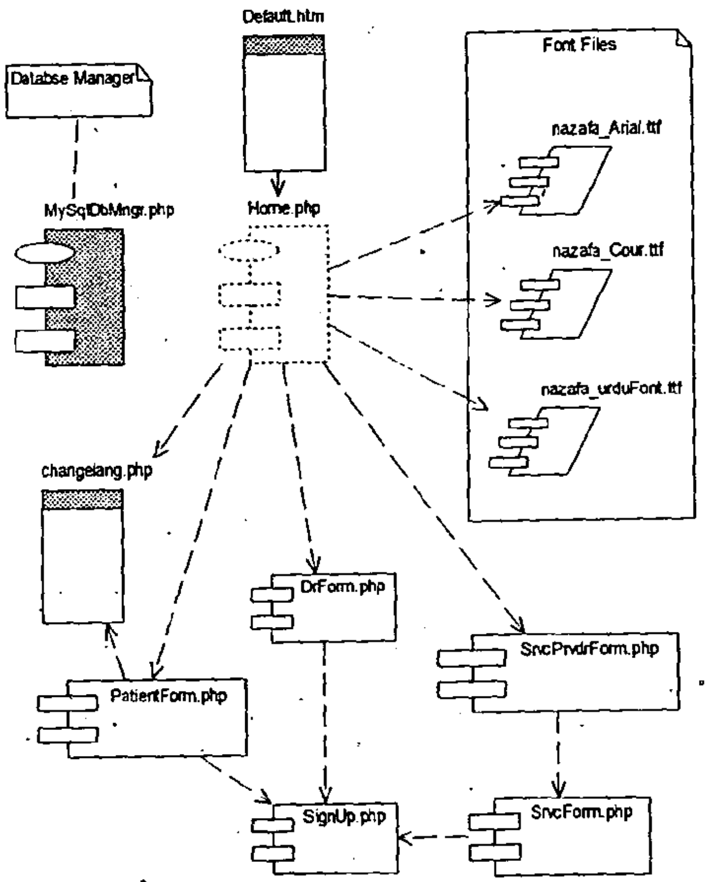


Figure 6.3 Signup of different users

6.1.4 Navigation and Search

Navigation in the web based systems is done by hyperlinks to the target parts of the systems. This component groups the navigational structure into one file. It also implements the facility to search the record of a particular doctor or service provider. Figure 6.4 shows the involved files and the relation between them.

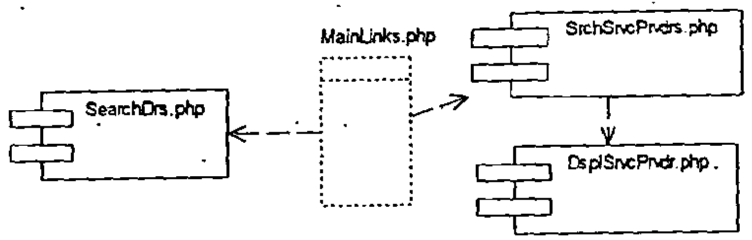


Figure 6.4 Navigation and Search Component

6.1.5 User Login Process

Login process for each of the three types of users leads to different parts of the system. Each part is customized according the needs of the users. More features are implemented for a doctor because he / she has to design the investigation questions and initiate the diagnosis process. Figure 6.5 shows the files involved and flow of data between them.

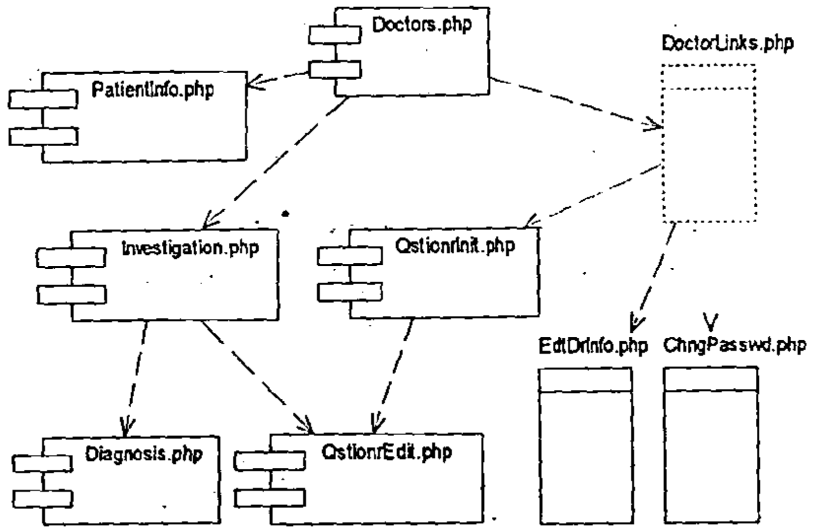


Figure 6.5 Login of Users

6.1.6 Doctor-Patient Interaction

This is the most important component in context of telemedicine. The manual procedure of doctor attending a patient in real life is simulated on the internet with a structured method of discussion. Figure 6.6 shows the files that implement this algorithm and their interation with each other.

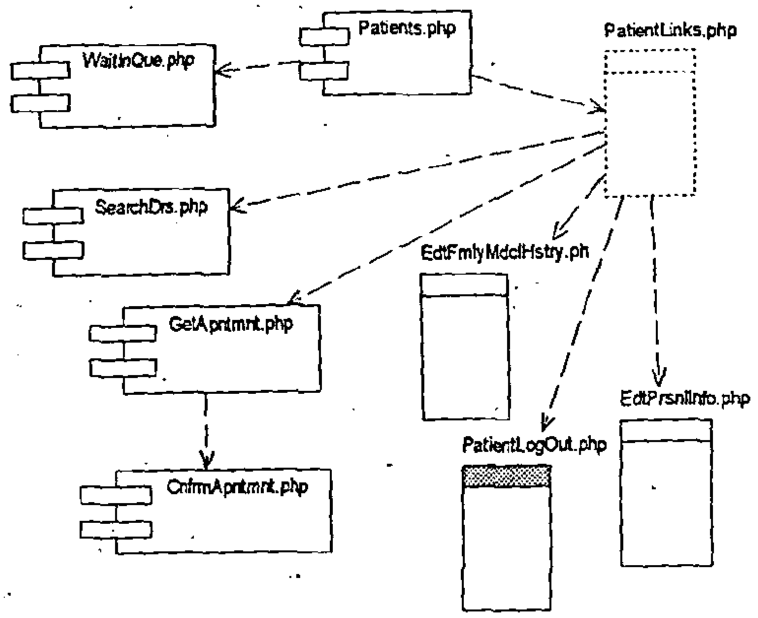


Figure 6.6 **Doctor – Patient Interaction**

6.2 Code Description

This section describes the above components in a format that cross-references the 'Typical Course of Events' and 'Alternative Courses' of each Use-Case from system's perspective. It provides the code written for implementation of the developed algorithms. One code segment may implement more than one Use-Case and one Use-Case may be implemented by more than one code segments.

6.2.1 Display Urdu Text

It is implemented in the files "code.js" and "UrduCodePlt.js". The following code handles the typical course of events including retrieve the English-Urdu ASCII encoded data from the database and system files, then composes and structures a dynamic web page containing required user interface elements.

```
if($_GET["lang"]=='Eng') {
        $langslct="Display the page in";
        $maintext =
        "<p align=center><font $hsize color=$hclr><i>Welcome</i><br>".
        "<font $sbhsize>to</font><br>".
        "<font color=$sbhclr><b>e-Health in Urdu.com</b></font></font></p>".
        "<p align=$dir2><font $sbhsz2 color=$sbhclr>How are you feeling
        today?</font></p>";
```

```
          $texts = array("Visit as", "Patient" ,"Doctor" ,"Service Provider", //0,1,2,3
          "LoginID", "Password", "Go", "New Users", "SIGN UP" ,"font" ); //4,5,6,7,8,9

} else {  $style="$ustyle"; $langslct="~u<Kg43Qi(BI:h(5Qa(?54N";//'>"
          $maintext =
          "~e<p align=center><font $hsize color=$hclr><i>~u;OA(/M<Q<~e</i><br>".
          "<font color=$sbhclr><b>~u5>JP~e-~uB:7(4>&lt;OMQf~e</b></font><br>".
          "<font $sbhsize>~u`>~e</font></font></p>".
          "~e<p align=$dir2><font $sbhsz2 color=$sbhclr>~u/9(/`KP(D5QF7(KQ@P(hi-
          ~e<br><br>".
          "<blockquote>(~u7O(KQ4+(K>(Li(e4k~e!)</blockquote></font></p>";

          $texts = array("~u<O>h(~uKQ9Q3i(5DO>",
          "~uM>QC","~ub4K_>","~u;<M7(I>4hM(KVNWN<h", //0,1,2,3
          "~u<4;Lh(AN4;7","~ueV?U>(LIE","~u943Qi",
          "~uN3i(4@7FM4L(KVNWN<e4N","~uAMOLQX7(K>Qf", //4,5,6,7,8
          "~uION_"); //9
}
<font face="$face"><p dir=$dir align=$dir2>$startdelim$texts[9]$enddelim <select
name="SlctFont" size="1" dir="ltr"

OnChange="KyBrdPanel.font.value=SlctFont.options[SlctFont.selectedIndex].text;location.rel
oad()">
<option>NaZaFa Arial<option>NaZaFa Courier New<option selected>NaZaFa Urdu
Font</select>
  <script>ValueSetter(SlctFont,"$font")</script></p>
$startdelim$maintext$enddelim</font>
```

Given below is the code that converts each one-byte data string into Urdu Unicode text using the Code Plate which consists of Unicode values' map, and echoes it to the Browser.

```
function D(textVal,wr)
{
          EscpChar="";
          for(j=0;j < textVal.length;j++)
          {
                    if(textVal.substr(j,2)==codePlate[0])
                    { EscpChar=codePlate[0]; j+=2;} // Two escpchars skipped
                    else if(textVal.substr(j,2)=='~e')
                    {         EscpChar='~e'; j+=2;} // This is more efficient than using 'continue'
                    slice1=textVal.slice(0,j);
                    slice2=textVal.slice(j+1);
                    if(EscpChar==codePlate[0])
                              textVal=slice1+String.fromCharCode(codePlate[textVal.charCodeAt
                              (j)])+slice2;
                    else if(EscpChar=='~e')
                              textVal=slice1+textVal.charAt(j)+slice2;
          }
          textVal=textVal.replace(new RegExp(codePlate[0]+"|~e","g"),"");
          if(typeof(wr)=="undefined") document.write(textVal);     /*return textVal;*/
}
"UrduCodePlt.js":
var urdu = new Array('~u', 0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,32,46,1524,1548,1563,1567,1569,1570,1571,1572,1573,1574,1575,1576,15
77,1578,1579,1580,1581,1582,1583,1584,1585,1586,1587,1588,1589,1590,1591,
1592,1593,1594,1600,1601,1602,1603,1604,1605,1606,1608,1609,1610,1611,
1612,1613,1614,1615,1616,1617,1618,1642,1643,1643.5,1645,1648,1657,1662,1670,1
672,1681,1688,1711,1722,1726,1729, 1746,1747,1748,1764,1776,1777,1778,1779,
1780,1781,1782,1783,1784,1785,8216,8217,8219,9679,65275);
codePlate=urdu;
```

The alternative course of events includes when One-Byte character not found in the code plate. Abort the operation is code as.

```
        else if(textVal.substr(j,2)=='~e')
                {        EscpChar='~e'; j+=2;} // This is more efficient than using 'continue'
```

6.2.2 Input Urdu Text

It is implemented in the files "code.js", "KyBrdPanel.php", "UrduStdKyBrd.js", and "UrduPhntcKyBrd.js". The code given below handles the typical course of events that converts input characters into non-English Unicode character based on selected keyboard layout and Urdu-English typing choice. 'Keyboard' is a Unicode values' pattern in a keyboard format.

```
function WriteUrdu()
{
        if (KyBrdPanel.chkbxWrEng.checked) return;
        UrduCode = keyBoard[event.keyCode];
        event.keyCode = arabCode ? UrduCode:event.keyCode;
}
```

Following code is implemented in "KyBrdPanel.php":

```
echo <<< Page
<form method="POST" name="KyBrdPanel">
<table border="0" width="100%">
  <tr>
        <td width="10%" bgcolor="#FFFF00"><p><i><b><u>Keyboards</u></b></i>:-
        <td width="20%" bgcolor="#33CCFF"><p><input type="radio" name="rdoKyBrd"
        value="UrduStd" checked
        onClick="keyBoard=UrduStd"><b>Urdu Standard</td>
        <td width="20%" bgcolor="#33CCFF"><p><input type="radio" name="rdoKyBrd"
        value="UrduPhntc"  onClick="keyBoard=UrduPhntc"><b>Urdu Phonetic</td>
        <td width="10%" align="center" bgcolor="#FF0000"><p><input type="button"
        value="Display" name="B1" onClick="for(i=0;i<KyBrdPanel.rdoKyBrd.length;i++) if(
        KyBrdPanel.rdoKyBrd[i].checked==true)
        window.open("$prextn/images/"+ rdoKyBrd[i].value +"KbLayout.php",
        "","toolbar=0,status=0,resizable=1,scrollbars=0");'></td>
        <td width="40%" align="left" bgcolor="#FFCC66">
        <input type="checkbox" name="chkbxWrEng" value="WrEng"> <b>Normal</td>
  </tr>
</table>
        <input type="hidden" name="pageId"><input type="hidden" name="y_cordn"><input
        type=hidden name="hmpglang">
        <input type=hidden name="font" value="$_COOKIE[font]">
</form>
"UrduStdKyBrd.js":
var UrduStd = new Array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
        0,0,0, ,1600,0,1574,1610,0,46,0,0,1612,1570,1548,1571,1748,1618,
        1776,1777,1778, 1779,1780,1781,1782,1783,1784,1785,0,1563,1616,1572,1614,
        1567,1643,1764,0,1579,1672,  0,0,1594,1581,1648,1590,1582,0,1573,1722,
        1577,1615,1615, 1681,1589,1657,8216,1592,0, 688,0,1584,0,0,0,0,1617,1613,1575,
        1576, 1670,1583,1593, 1601,1711,1726,1609,1580, 1603,1604,1605,1606,
        1729,1662, 1602,1585,1587,1578,1569,1591,1608,1588,1746,1586,
        8216,0,8217,1611);
keyBoard=UrduStd;
"UrduPhntcKyBrd.js":
var UrduPhntc = new Array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
        0,0, 0,1524,0,0,0,0,1612,0,0,0,0,1548,0,0,0,1776,1777,1778,1779,1780,1781,1782,
        1783,1784,1785,0,1563,1616,0,1614,1567,0,1570,1613,1579,1672,1645,1688,1594,
        1581, 1609,9679,1582,65275,1642,1606,1577,0,1611,1681,1588,1657,0,1592,
        1572,1590,1747, 1584,1617,1615,0,0,1600,0,1575,1576,1670,1583,1593,1601,1711,
        1729,1610,1580,1603, 1604,1605,1722,1726,1662,1602,1585,1587,1578,1569,1591,
        1608,1589,1746,1586,0,1618, 1648,8219);
keyBoard=UrduPhntc;
```

The Alternative Courses of events that perform corresponding key match in the keyboard is null or absent then return the character as it is.

```
        event.keyCode = arabCode ? arabCode:event.keyCode;
```

6.2.3 Send Form Data

It is implemented in the files "code.js" and "UrduCodePlt.js". The following code handles the typical course of events to Scan through all the form fields retrieving each one's value tactfully, while simply passing on the 'pure' English values.

```
function ConvertCode(PhoneyForm)
{
          for(i=0;i < PhoneyForm.length;i++) {
                  textVal=ValueRetriever(PhoneyForm[i],PhoneyForm);
                  if(PhoneyForm[i].name.substr(0,3)!='Eng') textVal=C(textVal);
                  HidForm[i].value=textVal;
          }
}
function ValueRetriever(field,form)
{
 switch (field.type) {     // Convert multiple line textarea string to single, for server variable
        case 'textarea':  return field.value.replace( new RegExp("\r\n","g"),'<br>');
        case 'text': case 'password': case 'hidden':          return field.value;
        case 'select-one':
                return field.options[field.selectedIndex].value?
                field.options[field.selectedIndex].value:
                field.options[field.selectedIndex].text;
        case 'select-multiple':
                vals=""; for(j=0;j < field.length;j++)
                if (field.options[j].selected==true) {
                vals+=(field.options[j].value?field.options[j].value:field.options[j].text)+"#"; }
                return vals;
        case 'radio': radioVal=''; for(k=i;k < form.length;k++) if(form[k].name==field.name)
                { if(form[k].checked==true) radioVal=form[k].value; i=k; } else break;
                return radioVal;
        case 'checkbox': return field.checked?field.value:"";
        default: return "";
   }
}
```

The part that converts each English-Urdu Unicode data string into a single-byte encoded ASCII data string character by character, utilizing an efficient Code Plate search mechanism in the process.

```
function C(textVal)
{
        prevEscpChar="";
        for(j=0;j < textVal.length;j++) {
                slice1=textVal.slice(0,j);
                slice2=textVal.slice(j+1);
                textVal=slice1+GetChar( textVal.charCodeAt(j) )+slice2;
        }
        return textVal;
}
function GetChar(charCode)
{
        if(charCode>255 || charCode==32 || charCode==46)  return
        TheChar(String.fromCharCode(SearchBinary(charCode)),codePlate[0]);
        return TheChar(String.fromCharCode(charCode),'~e');
}
function TheChar(Char,EscpChar)
{
        if(prevEscpChar==EscpChar)     return Char;
        else {
                prevEscpChar=EscpChar; j+=2;
                return EscpChar+Char;  // start index for slice1
        }
}
function SearchBinary(charCode)
{
        low1=1,high1=codePlate.length/2-1, mid=parseInt(high1+1);
        if(charCode < codePlate[mid]) // Most likely typed letters lie in first half
                while(low1<=high1)      {
                        mid1=parseInt((low1+high1)/2);
                        if(charCode>codePlate[mid1]) low1=mid1+1;
                        else if(charCode < codePlate[mid1])      high1=mid1-1;
                        else return mid1; // ('==' should be in last as rare chance)
                }
        low2=mid+1,high2=codePlate.length-1;
        while(low2<=high2){mid2=parseInt((low2+high2)/2);
                if(charCode < codePlate[mid2]) high2=mid2-1;
                else if(charCode>codePlate[mid2])        low2=mid2+1;
                else return mid2;}
        return mid; // of second one!
}
```

6.2.4 Receive Form Data

It is implemented in the files “Keybrdpanel.php”, “code.js” and “UrduCodePlt.js”. The following code handles the typical course of events that receives the form’s data as a form-name sequenced structure.

```
echo '<script>Post=new Array();';
reset ($_POST);
while (list ($key, $val) = each ($_POST))
echo <<< FillArray
        Post["$key"]="$val";
FillArray;
echo '</script>';
```

The part that retrieves each of the form field’s received English-Urdu encoded ASCII data from the form-name sequenced structure, by iterating through the form field names correspondingly.

```
function ConvertBack(PhoneyForm)
{
        for(i=0;i < PhoneyForm.length;i++) {
                textVal=Post[PhoneyForm[i].name];
                if(typeof(textVal)=="undefined") continue;
                if(PhoneyForm[i].name.substr(0,3)!='Eng') textVal=D(textVal,0);
                ValueSetter(PhoneyForm[i],textVal); // Resetting the Phoney form
        }
}
```

Following piece of program decodes each data string to a Unicode data value, partially <<using>> *Display Urdu Text:* Step4 *real* use case with an expectedly included difference of ‘returning’ the converted back value, while keeping the ‘pure’ English values intact.

```
function D(textVal,wr)
{
        EscpChar="";
        for(j=0;j < textVal.length;j++)
        {
                if(textVal.substr(j,2)==codePlate[0])
                { EscpChar=codePlate[0]; j+=2;} // Two escpchars skipped
                else if(textVal.substr(j,2)=='~e')
                {       EscpChar='~e'; j+=2;} // This is more efficient than using 'continue'
                slice1=textVal.slice(0,j);
                slice2=textVal.slice(j+1);
                if(EscpChar==codePlate[0])
        textVal=slice1+String.fromCharCode(codePlate[textVal.charCodeAt(j)])+slice2;
                else if(EscpChar=='~e')
```

```
                    textVal=slice1+textVal.charAt(j)+slice2;
        }
        textVal=textVal.replace(new RegExp(codePlate[0]+"|~e","g"),"");
        /*if(typeof(wr)=="undefined") document.write(textVal);*/  return textVal;
}
```

This code passes through all of the Browser's form fields and assigns the Unicode values to their respective form fields tactfully.

```
        ValueSetter(PhoneyForm[i],textVal); // Resetting the Phoney form
 }
}
function ValueSetter(field,val)
{
 switch (field.type) {
        case 'textarea': val=val.replace( new RegExp("<br>","g"),'\r\n'); // back2newline
        case 'text': case 'password': case 'hidden':          field.value=val; break;
        case 'select-one': for(j=0;j < field.length;j++) if(field.options[j].value==val ||
                field.options[j].text==val) field.options[j].selected=true; break;
        case 'select-multiple': vals=val.split("#"); for(j=0;j < vals.length-1;j++) for(k=0;k <
                field.length;k++)
                if(vals[j]==field.options[k].value || vals[j]==field.options[k].text)
                        { field.options[k].selected=true; break; }
                break;
        case 'radio': case 'checkbox': if(field.value==val) field.click(); break;
 };
}
```

The alternative course of events activates when the retrieved value is null or not defined on first loading. Skip the remaining steps for it.

```
        textVal=Post[PhoneyForm[i].name];
        if(typeof(textVal)=="undefined") continue;
```

6.2.5 Signup of Users

It is implemented in the files "home.php" and "signup.php". The code given in this section handles the typical course of events that examines User type and opens respective signup page:

a. If signing up as a Doctor, initiate *Signup as Doctor*.

b. If signing up as a Patient, initiate *Signup as Patient*.

c. If signing up as a Service Provider, initiate *Signup as Service Provider*.

```
sgnuplnks = new
Array("signup/PatientForm.php","signup/DrForm.php","signup/SrvcPrvdrForm.php");
i=0; usrtyp=ValueRetriever(UsrTypForm[i],UsrTypForm);
```

```
            else lnktyp.href=sgnuplnks[usrtyp];
```

Following code Stores the submitted data temporarily while checking database for a match with the User's chosen ID.

```
<form method="POST" action="" name="HidForm">
  <input type="hidden" name="UrduId"><input type="hidden" name="filler">
  <input type="hidden" name="EngName" value="$_POST[EngName]">
  <input type="hidden" name="UrduName" value="$_POST[UrduName]">
  <input type="hidden" name="UrduPasswd" value="$_POST[UrduPasswd]">
  <input type="hidden" name="EngCountry" value="$_POST[EngCountry]">...
</form>
switch ($_GET["User"]) {
  case 'Doctor':
        mysql_query( "Insert Doctors values
        ('$_POST[EngName]','$_POST[UrduName]','$_POST[UrduId]',
                password('$_POST[UrduPasswd]'), $_POST[EngCountry],...) ");
        if(mysql_errno()==1062)echo <<< Page
                $script
                The Id you choosed already exists. Kindly modify your DoctorID.
        else {  echo "Welcome Doctor <font $sbhsize
                color=$sbhclr>$_POST[EngName]</font>, ".
                "<font $sbhsize
                color=$sbhclr>$startdelim$_POST[UrduName]$enddelim</font><br>";
                }       break;
```

This is the piece of code that feeds User's data to database after doing necessary processing.

```
switch ($_GET["User"]) {
        case 'Doctor': mysql_query( "Insert Doctors values
                ('$_POST[EngName]','$_POST[UrduName]','$_POST[UrduId]',
                password('$_POST[UrduPasswd]'), $_POST[EngCountry],
                '$_POST[EngCity]', '$_POST[UrduCity]',
                '$_POST[EngSpecialization]','$_POST[UrduSpecialization]',$_POST[EngExp
                erience],'$_POST[EngEducation]',
                '$_POST[EngAffiliation]', '$_POST[UrduAffiliation]',
                '$_POST[EngFromHr]:$_POST[EngFromMin]',
                '$_POST[EngToHr]:$_POST[EngToMin]','$_POST[EngAvgTimeHr]:$_POST[
                EngAvgTimeMin]', $_POST[EngAmount], '$_POST[EngUnit]') ");
  case 'Patient': if($_POST[lang]=='Eng') {
                        echo "<script>for(i=0;i<KyBrdPanel.length;i++)
                        KyBrdPanel[i].disabled=true;</script>";
                        mysql_query( "Insert Patients values
                        ('$_POST[EngName]','$_POST[UrduId]',
                        password('$_POST[UrduPasswd]'), $_POST[EngCountry],
                        '$_POST[EngCity]', '$_POST[EngStrtAdrs]','
                        $_POST[EngPrfsion]','$_POST[EngRlgion]', '$_POST[EngGndr]',
                        '$_POST[EngYear]-$_POST[EngMonth]-$_POST[EngDay]',
                        '$_POST[lang]', '0') ");
                } else if($_GET["lang"]=='Urdu') {
                        mysql_query( "Insert Patients values
                        ('$_POST[UrduName]','$_POST[UrduId]',
                        password('$_POST[UrduPasswd]'),
                        $_POST[EngCountry],'$_POST[UrduCity]','$_POST[UrduStrtAdrs]',
                        '$_POST[UrduPrfsion]', '$_POST[UrduRlgion]', '$_POST[EngGndr]',
                        '$_POST[EngYear]-$_POST[EngMonth]-$_POST[EngDay]',
                    '$_POST[lang]', '0') ");
case 'SrvcPrvdr':  mysql_query( "Insert SrvcPrvdrs values ('$_POST[EngName]',
            '$_POST[UrduName]', '$_POST[UrduId]',
            password('$_POST[UrduPasswd]'), $_POST[EngCountry], $_POST[EngCity],
            '$_POST[EngSubArea]', '$_POST[UrduSubArea]', '$_POST[EngAddress]',
            '$_POST[UrduAddress]', '$_POST[EngReachHint]',
            '$_POST[UrduReachHint]', '$_POST[EngWebSt]', '$_POST[EngEmail]',
            $_POST[EngService], '$_POST[EngSrvcCategory]',
            '$_POST[EngSrvcProduct]') ");
```

Alternative course of events will be activated when the chosen User ID already exists. Inform and provide for the User with a form to provide another ID.

```
$script
        The Id you choosed already exists. Kindly modify your DoctorID.
 <form method="POST" action="SignUp.php?User=Doctor" name="IdForm" onSubmit="return
Check()">
<table border="0" width="100%" id="AutoNumber1">
 <tr>    <td width="30%">DoctorID</td>
         <td width="50%"><input type="text" name="UrduId" size="25" dir="rtl"
                 onKeyPress='WriteUrdu()' style="$ustyle"></td>
         <td width="20%"><input type="submit" value=" OK " name="B1"></td>
</tr> </table> </form>
<script>ConvertBack(IdForm);</script>
```

6.2.6 Signup as Doctor

It is implemented in the file "DrForm.php". Code given in this section handles the typical course of events that displays the form containing input elements for required information both in Urdu and English, loading necessary lists' data dynamically.

```
<body onLoad="ConvertBack(SignUpForm)">
<p align="center"><font $sbhsz2 color="$sbhclr">Please use duplicate fields to
specify your information in both English and Urdu</font></p>
<form method="POST" action="SignUp.php?User=Doctor" name="SignUpForm"
onSubmit="return Check()">
 <table border="0" width="75%" id="AutoNumber1">
  <tr>
   <td width="25%">Doctors Name</td>
   <td width="25%">Dr.<input type="text" name="EngName" size="30"></td>
   <td width="25%"><input type="text" name="UrduName" size="30" dir="rtl"
               onKeyPress="WriteArabic()" style="$ustyle"><font $sbhsz2
               face="$face">$startdelim~ub4K_>$enddelim</td>
  </tr>
...
   <td width="5%">City</td>
   <td width="17%"><select size="1" name="EngCity"
        onChange="UrduCity.options[EngCity.selectedIndex].selected=true">
Page;

$result=mysql_query("SELECT EngCity,UrduCity FROM Cities where
CountryNo='$_POST[EngCountry]' ". "order by CityNo");
while ($row = mysql_fetch_row($result)) echo"<option>$row[0]";
```

```
echo <<< Page
        </select></td>
        <td width="18%"><select size="1" name="UrduCity" style="$ustyle" dir="rtl"
        onChange="EngCity.options[UrduCity.selectedIndex].selected=true">
<script>
Page;
mysql_data_seek($result,0);
while ($row = mysql_fetch_row($result)) echo "D(\"~e<option>$row[1]\");";
echo <<< Page
</script></select></td>
    <td width="12%">(Not Listed?)</td>
    <td width="48%"><input type="text" name="EngCityAdd" size="20">
        <input type="text" name="UrduCityAdd" size="20" dir="rtl"
                            onKeyPress="WriteArabic()" style="$ustyle">
        <input type="button" value="Add" name="B3"
        onClick="Add(EngCityAdd,EngCity,UrduCityAdd,UrduCity)">
    </td>
  </tr>
  </table>
  <table border="0" width="100%" id="AutoNumber3">
   <tr>
    <td width="5%">Specialization</td>
    <td width="17%"><select size="1" name="EngSpecialization"
     onChange="UrduSpecialization.options[EngSpecialization.selectedIndex].selected=true">
    <option>Select a Specialization

Page;
$result=mysql_query("SELECT EngSpecialization,UrduSpecialization FROM Specializations ".
        "order by SpecializationNo");
while ($row = mysql_fetch_row($result)) echo "<option>$row[0]";
echo <<< Page
    </select></td>
    <td width="18%">
    <select size="1" name="UrduSpecialization" style="$ustyle" dir="rtl"
    onChange="EngSpecialization.options[UrduSpecialization.selectedIndex].selected=true">
    <option>
        &#1582;&#1589;&#1608;&#1589;&#1609;, &#1588;&#1593;&#1576;&#1729;
        &#1670;&#1606;&#1610;&#1746;
<script>
Page;

mysql_data_seek($result,0);
while ($row = mysql_fetch_row($result)) echo "D(\"~e<option>$row[1]\");";
echo <<< Page
</script></select></td>
    <td width="5%">(Un Listed?)</td>
    <td width="55%"><input type="text" name="EngSpecializationAdd" size="20">
        <input type="text" name="UrduSpecializationAdd" size="20" dir="rtl"
                            onKeyPress="WriteArabic()" style="$ustyle">
        <input type="button" value="Add" name="B4"
         onClick="Add(EngSpecializationAdd,EngSpecialization,
        UrduSpecializationAdd,UrduSpecialization)"></td>
        </tr>
  </table>...
  <table border="0" width="100%" id="AutoNumber5">
        <tr>
   <td width="55%">Time slot for online consultation (your country's time in 24 Hrs.)</td>
   <td width="45%">From <select size="1" name="EngFromHr">...
   </select> : <select size="1" name="EngFromMin"><option selected>00...
   </select> To <select size="1" name="EngToHr">...
```

```
            </select> : <select size="1" name="EngToMin"><option selected>00...
        </select></td>
            </tr>
    </table>
    <table border="0" width="100%" id="AutoNumber5_">
            <tr>
              <td width="35%">Average Time needed for a Patient</td>
              <td width="65%"><select size="1" name="EngAvgTimeHr"
            onChange="if(EngAvgTimeHr.selectedIndex==1) {
            zOption=new Option(); zOption.text='00'; zOption.selected=true;
            EngAvgTimeMin.add(zOption,0); }
            else if(EngAvgTimeHr.selectedIndex==0) {
                    for (i=0;i<EngAvgTimeMin.options.length-1;i++)
                            EngAvgTimeMin.options[i].text = EngAvgTimeMin.options[i+1].text;
                     EngAvgTimeMin.options.length = EngAvgTimeMin.options.length-1;
            }" dir="rtl"> <option selected>00<option>01
        </select>hr. & <select size="1" name="EngAvgTimeMin" dir="rtl">
         <option>05<option selected>10 <option>15<option>20<option>25<option>30
        <option>35<option>40<option>45<option>50<option>55
        </select>mins.</td>
      </tr>
    </table><p></p>
function ChangeCity(AList,BList)
{
            BList.options[AList.selectedIndex].selected=true;
            window.status="Please wait while Cities are loaded";
            HidForm.action='$PageID.php'; ConvertCode(SignUpForm); HidForm.submit();
}
```

Part of code that submits the data in database.

```
  <p><input type="submit" value="Submit" name="B1">
  <input type="reset" value="Reset" name="B2"></p>
</form>
<form method="POST" action="" name="HidForm">
  <input type="hidden" name="EngName"><input type="hidden" name="UrduName">
  <input type="hidden" name="UrduId">
  <input type="hidden" name="UrduPasswd"><input type="hidden"
name="UrduCnfrmPasswd">
  <input type="hidden" name="EngCountry"><input type="hidden" name="UrduCountry">
  <input type="hidden" name="EngCity"><input type="hidden" name="UrduCity">
  <input type="hidden" name="EngCityAdd"><input type="hidden" name="UrduCityAdd">
  <input type="hidden" name="filler">
  <input type="hidden" name="EngSpecialization"><input type="hidden"
name="UrduSpecialization">
  <input type="hidden" name="EngSpecializationAdd"><input type="hidden"
name="UrduSpecializationAdd">
  <input type="hidden" name="filler">
  <input type="hidden" name="EngExperience">
  <input type="hidden" name="EngEducation">
  <input type="hidden" name="EngAffiliation"><input type="hidden" name="UrduAffiliation">
  <input type="hidden" name="EngFromHr"><input type="hidden" name="EngFromMin">
  <input type="hidden" name="EngToHr"><input type="hidden" name="EngToMin">
  <input type="hidden" name="EngAvgTimeHr"><input type="hidden"
name="EngAvgTimeMin">
  <input type="hidden" name="EngAmount">
  <input type="hidden" name="EngUnit"><input type="hidden" name="EngUnitAdd">
  <input type="hidden" name="filler"><input type="hidden" name="filler"><input type="hidden"
name="filler">
```

```
</form>
 HidForm.action=SignUpForm.action; ConvertCode(SignUpForm); HidForm.submit();
 return false;
```

Alternative course of events is triggered when added element is empty or already exists. Prompt to Doctor and discontinue the operation.

```
function Add(Engitem,Englist,Urduitem,Urdulist)
{
        relnsert=false;
        for(j=0;j<Englist.length;j++)
                if(Englist.options[j].text.toLowerCase()==Engitem.value.toLowerCase() ||
                Urdulist.options[j].text==Urduitem.value) relnsert=true;
                ListName=Englist.name.substr(3,Englist.name.length);
                if(ListName=='City' && SignUpForm.EngCountry.selectedIndex==0)
                        alert('Please Select a Country first');
                else if( Engitem.value=='' || Urduitem.value=='')
                        alert("Can't add empty value to "+ ListName +"");
                else if (relnsert) alert("The value already exists in "+ ListName +"");
                else {
                        HidForm.action='$PageID.php?add='+ListName;
                        window.status="Please wait while "+ ListName +" is added";
                        ConvertCode(SignUpForm);
                        HidForm.submit();
        }
}
Page;

if($_GET["add"]!='')
 switch ($_GET["add"]) {
        case 'City': mysql_query( "Insert Cities (EngCity,UrduCity,CountryNo) ".
                "values ('$_POST[EngCityAdd]','$_POST[UrduCityAdd]',
                $_POST[EngCountry])" );
                echo 'Post["EngCity"]=Post["EngCityAdd"];
                Post["UrduCity"]=Post["UrduCityAdd"];'; break;
        case 'Specialization': mysql_query( "Insert Specializations
                (EngSpecialization,UrduSpecialization) "."values
                ('$_POST[EngSpecializationAdd]', '$_POST[UrduSpecializationAdd]')" );
                echo 'Post["EngSpecialization"]=Post["EngSpecializationAdd"];'.
                'Post["UrduSpecialization"]=Post["UrduSpecializationAdd"];';      break;
```

Another Alternative course of event is when Password, Time or 'numeric' fields fail to comply. Inform the Doctor and abort submitting.

```
function Check()
{
 for(i=0;i<SignUpForm.length;i++) {
        if( (SignUpForm[i].type=="text" && SignUpForm[i].value=='' &&
                SignUpForm[i].name.substr(SignUpForm[i].name.length-3,3)!='Add') ||
                (SignUpForm[i].type=="select-one" &&  SignUpForm[i].options
                [SignUpForm[i].selectedIndex].text.substr(0,6) =='Select') )
        {
                alert("The field "+SignUpForm[i].name.substr(3,SignUpForm[i].name.length)
                +'" is empty"); return false;
        }
 }
 if(SignUpForm.UrduPasswd.value!=SignUpForm.UrduCnfrmPasswd.value)
        { alert("Password fields do not match"); return false; }
 if( (SignUpForm.EngFromHr.selectedIndex+SignUpForm.EngAvgTimeHr.selectedIndex >
          SignUpForm.EngToHr.selectedIndex) ||
          ( (SignUpForm.EngFromHr.selectedIndex +
SignUpForm.EngAvgTimeHr.selectedIndex ==  SignUpForm.EngToHr.selectedIndex) &&
(SignUpForm.EngFromMin.selectedIndex+(SignUpForm.EngAvgTimeMin.selectedIndex+1)*5
>=  SignUpForm.EngToMin.selectedIndex) ) )
 { alert("FromTime+AvgTime should be Less than ToTime"); return false; }

 if(isNaN(SignUpForm.EngAmount.value))
        { alert("Amount must be numeric"); return false; }
```

6.2.7 Signup as Patient

It is implemented in the files "changelang.php" and "PatientForm.php". The Typical Course of Events that Asks the Patient to choose a language (English or Urdu) is handled by the following code:

```
if($_GET["lang"]=='Init') {
        echo "<script>for(i=0;i<KyBrdPanel.length;i++)
        KyBrdPanel[i].disabled=true;</script>".
        "<br><font $sbhsz2 color=$hclr>Please first select a Language<p align='right'>
        <font "."face='$face'>$startdelim~u5>4h(Mh>54NP(`hLi(?54N(MN7;5(KQ9Qi.
        $enddelim</font></p>";
        include '../includes/changelang.php';
}
"changelang.php":
echo <<< LangSelect
<table width="100%" dir=ltr>
        <tr>
          <td width="50%" align="center" bgcolor="#00FFFF">
        <font $sbhsz2 face="Courier New"> <a href="$PageID.php?lang=Eng"
        style="text-decoration: none">English>>></a></td>
         <td width="50%" align="center" bgcolor="#00FF00"><font $sbhsz2
        face="Courier New"><a href="$PageID.php?lang=Urdu" style="text-decoration:
        none"><<<&#1575;&#1585;&#1583;&#1608;</a></td>
        </tr>
</table>
LangSelect;
```

Code that displays the form containing input elements for required information, loading necessary lists' data dynamically is written in "PatientForm.php":

```
else if($_GET["lang"]=='Urdu') {
<form method="POST" action="SignUp.php?User=Patient" name="SignUpForm"
onSubmit="return Check()" dir="rtl">
<table border="0" width="75%" style="$ustyle" id="AutoNumber1">
   <tr><td width="20%">$startdelim~uM>QC(K4(N4M$enddelim</td><td width="55%">
          <input type="text" name="UrduName" size="30" onKeyPress="WriteArabic()"
        style="$ustyle;"></td>
```

```
          </tr>...
  <table border="0" width="100%" style="$ustyle" id="AutoNumber2_">
          <tr><td width="20%">$startdelim~uAh>$enddelim</td><td width="30%">
            <select size="1" name="UrduCity" style="$ustyle">
          <script>
Page;

$result=mysql_query("SELECT UrduCity FROM UrduPatientCities where
CountryNo='$_POST[EngCountry]' ". "order by CityNo");
while ($row = mysql_fetch_row($result)) echo "D(\"~e<option>$row[0]\");";

echo <<< Page
</script></select></td>
<td width="15%">($startdelim~uMO9O<NhQf-$enddelim)</td><td width="35%">
<input type="text" name="UrduCityAdd" size="20" onKeyPress="WriteArabic()"
style="$ustyle"> <input type="button" name="B3" style="$ustyle;"
OnClick="Add(UrduCityAdd,UrduCity)"></td>
  </tr>
  </table>.
  <table border="0" width="73%" style="$ustyle" id="AutoNumber1"><tr>
   <td width="33%">$startdelim~ueLP(K4(`7h$enddelim</td><td width="40%">
   <input type="text" name="UrduStrtAdrs" size="40" onKeyPress="WriteArabic()"
style="$ustyle"> </td></tr> </table><p></p>
  <table border="0" width="100%" style="$ustyle" id="AutoNumber1_">
    <tr>
  <td width="21%">$startdelim~u`QAh$enddelim</td><td width="41%">
<input type="text" name="UrduPrfsion" size="25" onKeyPress="WriteArabic()"
style="$ustyle"> </td><td width="12%">$startdelim~u<QN~e/~uM=h5$enddelim</td><td
width="26%"> <input type="text" name="UrduRIgion" size="20" onKeyPress="WriteArabic()"
style="$ustyle"> </td></tr>
  </table>
  <table border="0" width="100%" style="$ustyle"><tr>
    <td width="20%">$startdelim~uBNI$enddelim</td><td width="30%">
          <input type="radio" name="EngGndr" value="M">$startdelim~uM><4Nh($enddelim
          <input type="radio" name="EngGndr" value="F"> startdelim~u?N4Nh$enddelim</td>
   <td width="15%">$startdelim~u74>Q;(`Q<43QA$enddelim</td> <td width="35%">
          <select name="EngDay" size="1" style="$ustyle">
          <option  value="0">&#1583;&#1606;      </select>
          <select name="EngMonth" size="1" onChange="ChangeDays()" style="$ustyle">
                  <option value="0">&#1605;&#1575;&#1729;...   </select>
          <select name="EngYear" size="1" onChange="ChangeDays()" style="$ustyle">
                  <option value="0">&#1587;&#1575;&#1604;...
          </select></td></tr>
  </table><p></p>
<p><input type="submit" name="B1" style="$ustyle">
<input type="reset" name="B2" style="$ustyle"></p>
</form><script>SignUpForm.B3.value=String.fromCharCode(1588,1575,1605,1604,32,1603,
1585,1608);...
</script>
function ChangeCity()
{
        window.status="Please wait while Cities are loaded";
        HidForm.action='$PageID.php?lang=$_GET[lang]'; ConvertCode(SignUpForm);
HidForm.submit();
}
```

Following code handles data submission and saving it in database according to the language specified:

```
<form method="POST" action="" name="HidForm">
  <input type="hidden" name="EngName"><input type="hidden" name="UrduId">
  <input type="hidden" name="UrduPasswd"><input type="hidden"
name="UrduCnfrmPasswd">
  <input type="hidden" name="EngCountry"><input type="hidden" name="EngCity">
  <input type="hidden" name="EngCityAdd"><input type="hidden" name="filler">
  <input type="hidden" name="EngStrtAdrs">
  <input type="hidden" name="EngPrfsion"><input type="hidden" name="EngRlgion">
  <input type="hidden" name="filler"><input type="hidden" name="EngGndr">
  <input type="hidden" name="EngDay"><input type="hidden" name="EngMonth"><input
type="hidden" name="EngYear">
  <input type="hidden" name="filler"><input type="hidden" name="filler">
  <input type="hidden" name="lang" value="$_GET[lang]">
</form>
  HidForm.action=SignUpForm.action; ConvertCode(SignUpForm); HidForm.submit();
  return false;
```

Alternative Courses includes the situation when added element is empty or already exists. Prompt to Patient and discontinue the operation.

```
function Add(item,list)
{
reInsert=false;
for(j=0;j<list.length;j++) if(list.options[j].text.toLowerCase()==item.value.toLowerCase())
reInsert=true;
ListName=list.name.substr(3,list.name.length);
if(SignUpForm.EngCountry.selectedIndex==0) alert('Please Select a Country first');
else if( item.value=='') alert("Can't add empty value to '"+ ListName +"'");
else if (reInsert) alert("The value already exists in '"+ ListName +"'");
else {
        HidForm.action='$PageID.php?lang=$_GET[lang]&add='+ListName;
        window.status="Please wait while "+ ListName +" is added";
        ConvertCode(SignUpForm);
        HidForm.submit();
        }
}
```

Another alternative course invokes when password fields fail to comply. Inform the Patient and abort submitting.

```
function Check()
{
 for(i=0;i<SignUpForm.length;i++) {
        if( (SignUpForm[i].type=="text" && SignUpForm[i].value=='' &&
                SignUpForm[i].name.substr(SignUpForm[i].name.length-3,3)!='Add') ||
                (SignUpForm[i].type=="password" && SignUpForm[i].value=='') ||
                (SignUpForm[i].type=="select-one" &&
                SignUpForm[i].options[SignUpForm[i].selectedIndex].value=="0") ||
                (SignUpForm[i].type=="radio" && SignUpForm[i].checked==false &&
                SignUpForm[i+1].checked==false) )
        {
                alert ("The field '"+ SignUpForm[i].name.substr(3,SignUpForm[i].name.length)
                +"' is empty");    return false;
        }
```

```
  }
  if(SignUpForm.UrduPasswd.value!=SignUpForm.UrduCnfrmPasswd.value)
        { alert("Password fields do not match"); return false; }
}
function ChangeDays()
{
        mnth = SignUpForm.EngMonth.selectedIndex;
        bgnYr=1904;    yr = bgnYr+SignUpForm.EngYear.selectedIndex;
        numDays = NumDaysIn(mnth,yr);
        SignUpForm.EngDay.options.length = numDays;
        for (i=27;i<numDays;i++)        SignUpForm.EngDay.options[i].text = i+1;
}
function NumDaysIn(mnth,yr)
{
        switch(mnth) {
                case 4: case 6: case 9: case 11: return 30; break;
                case 2: if(LeapYear(yr)) return 29; else return 28; break;
                default: return 31;
        };
}
function LeapYear(yr)
{
        if (((yr % 4 == 0) && yr % 100 != 0) || yr % 400 == 0) return true;  else return false;
}
```

6.2.8 Signup as Service Provider

This part is implemented in “SrvcPrvdrForm.php” and “SrvcForm.php”. Typical course of events includes the code that displays the form containing input elements for required information both in Urdu and English, loading necessary lists’ data dynamically. The portion of code included in “SrvcPrvdrForm.php” is given below:

```
<body onLoad="ConvertBack(SignUpForm)">
<p align="center"><font $sbhsz2 color="$sbhclr">Please use duplicate fields to
specify your information in both English and Urdu</font></p><form method="POST"
action="SrvcForm.php" name="SignUpForm" onSubmit="return Check()">
 <table border="0" width="75%" id="AutoNumber1">
   <tr>
    <td width="25%">Name</td>
    <td width="25%"><input type="text" name="EngName" size="30"></td>
    <td width="25%"><input type="text" name="UrduName" size="30" dir="rtl"
                onKeyPress="WriteArabic()" style="$ustyle"></td>
  </tr>...
 <table border="0" width="100%" id="AutoNumber2_1">
 <tr>
    <td width="5%">City</td>
    <td width="17%"><select size="1" name="EngCity"
onChange="ChangeSubArea(EngCity,UrduCity)">
Page;

$result=mysql_query("SELECT EngCity,UrduCity,CitySerNo FROM Cities where
CountryNo='$_POST[EngCountry]' ". "order by CityNo");
while ($row = mysql_fetch_row($result)) echo"<option value=$row[2]>$row[0]";
echo <<< Page
        </select></td>
```

```
          <td width="18%"><select size="1" name="UrduCity" style="$ustyle" dir="rtl"
          onChange="ChangeSubArea(UrduCity,EngCity)">
                    <script>
Page;

mysql_data_seek($result,0);
while ($row = mysql_fetch_row($result)) echo "D(\"~e<option>$row[1]\");";

echo <<< Page
          </script></select></td>
          <td width="12%">(Not Listed?)</td>
          <td width="48%"><input type="text" name="EngCityAdd" size="20">
          <input type="text" name="UrduCityAdd" size="20" dir="rtl"
                    onKeyPress="WriteArabic()" style="$ustyle">
          <input type="button" value="Add" name="B3"
          onClick="Add(EngCityAdd,EngCity,UrduCityAdd,UrduCity)">
     </td>
  </tr>
  </table>
  <table border="0" width="100%" id="AutoNumber2_2">
  <tr>
     <td width="5%">SubArea</td>
     <td width="17%"><select size="1" name="EngSubArea"
          onChange="UrduSubArea.options[EngSubArea.selectedIndex].selected=true">
     <option>Select SubArea
Page;

if($_POST[EngCity]=='' || $_POST[CityChanged])
{         mysql_data_seek($result,0); $row=mysql_fetch_row($result);
$_POST[EngCity]=$row[2];       }
$result=mysql_query("SELECT EngSubArea,UrduSubArea FROM SubAreas where
CitySerNo='$_POST[EngCity]' ".
          "order by SubAreaNo");
if(mysql_num_rows($result)==0) echo "<option value='-1'>SubArea Name";
else while ($row = mysql_fetch_row($result)) echo"<option>$row[0]";

echo <<< Page
          </select></td>
          <td width="18%"><select size="1" name="UrduSubArea" style="$ustyle" dir="rtl"
          onChange="EngSubArea.options[UrduSubArea.selectedIndex].selected=true">
  <option>&#1586;&#1610;&#1585;&#1610;&#1722; &#1593;&#1604;&#1575;
          &#1602;&#1729; &#1670;&#1606;&#1610;&#1746;
<script>
Page;

if(mysql_num_rows($result)==0) echo "D(\"~e<option>~u?Q>Qf(FL4Jh(K4(N4M\");";
else {    mysql_data_seek($result,0);
while ($row = mysql_fetch_row($result)) echo "D(\"~e<option>$row[1]\");";          }

echo <<< Page
          </script></select></td>
           <td width="12%">(Not Listed?)</td>
          <td width="48%"><input type="text" name="EngSubAreaAdd" size="20">
          <input type="text" name="UrduSubAreaAdd" size="20" dir="rtl"
                    onKeyPress="WriteArabic()" style="$ustyle">
           <input type="button" value="Add" name="B3"
                    onClick="Add(EngSubAreaAdd,EngSubArea,
          UrduSubAreaAdd,UrduSubArea)">
    </td>
 </tr>...
```

```
function ChangeCity(AList,BList)
{
        BList.options[AList.selectedIndex].selected=true;
        window.status="Please wait while Cities are loaded";
        HidForm.CityChanged.value=true; // for change of subareas
        HidForm.action='$PageID.php'; ConvertCode(SignUpForm); HidForm.submit();
}
function ChangeSubArea(AList,BList)
{
        BList.options[AList.selectedIndex].selected=true;
        window.status="Please wait while SubAreas are loaded";
        HidForm.action='$PageID.php'; ConvertCode(SignUpForm); HidForm.submit();
}
```

The portion of code that stores the general information temporarily and displays a dynamic services specification form that allows multiple selections and addition of data elements unspecified or required by the Service Provider is included in code file "SrvcForm.php":

```
echo <<< prevForm
  <input type="hidden" name="filler"><input type="hidden" name="SrvcCategorySerNo">
  <input type="hidden" name="EngSrvcProductAdd"><input type="hidden"
name="UrduSrvcProductAdd">
  <input type="hidden" name="EngSrvcProduct">
  <input type="hidden" name="EngName" value="$_POST[EngName]">
  <input type="hidden" name="UrduName" value="$_POST[UrduName]">
  <input type="hidden" name="UrduId" value="$_POST[UrduId]">
  <input type="hidden" name="UrduPasswd" value="$_POST[UrduPasswd]">
  <input type="hidden" name="EngCountry" value="$_POST[EngCountry]">
  <input type="hidden" name="EngCity" value="$_POST[EngCity]">
  <input type="hidden" name="EngSubArea" value="$_POST[EngSubArea]">
  <input type="hidden" name="UrduSubArea" value="$_POST[UrduSubArea]">
  <input type="hidden" name="EngAddress" value="$_POST[EngAddress]">
  <input type="hidden" name="UrduAddress" value="$_POST[UrduAddress]">
  <input type="hidden" name="EngReachHint" value="$_POST[EngReachHint]">
  <input type="hidden" name="UrduReachHint" value="$_POST[UrduReachHint]">
  <input type="hidden" name="EngWebSt" value="$_POST[EngWebSt]">
  <input type="hidden" name="EngEmail" value="$_POST[EngEmail]">
</form>
prevForm;
<p align="center"><font $sbhsz2 color="$sbhclr">Please use duplicate fields to
specify your information in both English and Urdu</font></p>
<form method="POST" action="SignUp.php?User=SrvcPrvdr" name="SignUpForm"
onSubmit="return Check()">
  <table border="0" width="100%" id="AutoNumber3">
    <tr>
      <td width="5%">Service Provided</td>
      <td width="17%"><select size="1" name="EngService"
onChange="ChangeSrvcCategory(EngService,UrduService)">
      <option value="99">Select a Service
Page;

$result=mysql_query("SELECT EngService,UrduService,SrvcSerNo FROM Services order by
ServiceNo");
while ($row = mysql_fetch_row($result)) echo "<option value=$row[2]>$row[0]";

echo <<< Page
```

```
    </select></td>
    <td width="18%">
    <select size="1" name="UrduService" style="$ustyle" dir="rtl"
         onChange="ChangeSrvcCategory(UrduService,EngService)">
    <option>
         &#1588;&#1593;&#1576;&#1729;&#1569; &#1582;&#1583;&#1605;&#1578;
         &#1670;&#1606;&#1610;&#1746;
         <script>
Page;

mysql_data_seek($result,0);
while ($row = mysql_fetch_row($result)) echo "D(\"~e<option>$row[1]\");";

echo <<< Page
                 </script></select></td>
    <td width="5%">(UnListed?)</td>
    <td width="55%"><input type="text" name="EngServiceAdd" size="20">
         <input type="text" name="UrduServiceAdd" size="20" dir="rtl"
                 onKeyPress="WriteArabic()" style="$ustyle">
         <input type="button" value="Add" name="B4" onClick="Add(EngServiceAdd,
         EngService, UrduServiceAdd,UrduService)"></td>
   </tr>
  </table>
  <table border="0" width="100%" id="AutoNumber2_1">
  <tr>
    <td width="5%">Select Service Categories</td>
    <td width="17%"><select size="3" name="EngSrvcCategory" multiple
         onChange="OnChangeSrvcCategory(EngSrvcCategory,UrduSrvcCategory)">
Page;

if($_POST[EngService]=='') $_POST[EngService]=99;
$result=mysql_query("SELECT EngSrvcCategory,UrduSrvcCategory,SrvcCategorySerNo
FROM SrvcCategories ". "where SrvcSerNo='$_POST[EngService]' order by
SrvcCategoryNo");
if(mysql_num_rows($result)==0) echo "<option value='-1'>Service Categories";
else while ($row = mysql_fetch_row($result)) echo"<option value=$row[2]>$row[0]";

echo <<< Page
         </select></td>
         <td width="18%"><select size="3" name="UrduSrvcCategory" style="$ustyle" dir="rtl"
multiple
         onChange="OnChangeSrvcCategory(UrduSrvcCategory,EngSrvcCategory)">
                 <script>
Page;

if(mysql_num_rows($result)==0) echo "D(\"~e<option>~u4J@4M(AF5h.;<M7\");";
else {   mysql_data_seek($result,0);
while ($row = mysql_fetch_row($result)) echo "D(\"~e<option>$row[1]\");";        }

echo <<< Page
                 </script></select></td>
    <td width="12%">(Not Listed?)</td>
    <td width="48%"><input type="text" name="EngSrvcCategoryAdd" size="20">
         <input type="text" name="UrduSrvcCategoryAdd" size="20" dir="rtl"
                 onKeyPress="WriteArabic()" style="$ustyle">
      <input type="button" value="Add" name="B3"

         onClick="Add(EngSrvcCategoryAdd,EngSrvcCategory,UrduSrvcCategoryAdd,UrduSr
vcCategory)">
    </td>
  </tr>
  </table>
  (Use mouse-drag or cntrl+mouse-button to select/de-select more than one Service Types)<p
align="center">
  <input type="button" value="  Show Service / Product Lists  " name="B2"
onClick="ShowSrvcProductLists()">
  </p><p></p><table border="0" width="100%" style="$ustyle" id="AutoNumber3">
Page;

///////////////////////////////////////
$srnmbr=explode("#",$_POST["EngSrvcCategory"]);
for($i=0;$i < count($srnmbr)-1;$i++) { if(mysql_num_rows($result)!=0)
mysql_data_seek($result,0);
        while ($row = mysql_fetch_row($result)) if($srnmbr[$i]==$row[2]) { $s=true; // enble
submit btn

echo <<< Page
  <tr>
    <td width="8%">$row[0] <script>D("$row[1]");</script></td>
    <td width="20%"><select size="5" name="EngSrvcProduct$row[2]" multiple
        onChange="ChangeMltplSelList(EngSrvcProduct$row[2],UrduSrvcProduct$row[2])">
Page;

$res=mysql_query("SELECT EngSrvcProduct,UrduSrvcProduct,SrvcProductSerNo FROM
SrvcProducts ".
        "where SrvcCategorySerNo='$row[2]' order by SrvcProductNo");
if(mysql_num_rows($res)==0) echo "";
else while ($r = mysql_fetch_row($res)) echo"<option value=$r[2]>$r[0]";

echo <<< Page
                </select></td>
        <td width="21%">
        <select size="5" name="UrduSrvcProduct$row[2]" style="$ustyle" dir="rtl" multiple

        onChange="ChangeMltplSelList(UrduSrvcProduct$row[2],EngSrvcProduct$row[2])">
                <script>
Page;

if(mysql_num_rows($res)==0) echo "";
else {   mysql_data_seek($res,0);
while ($r = mysql_fetch_row($res)) echo "D(\"~e<option>$r[1]\");";        }

echo <<< Page
                </script></select></td>
    <td width="51%"><input type="text" name="EngSrvcProductAdd$i" size="20">
        <input type="text" name="UrduSrvcProductAdd$i" size="20" dir="rtl"
                onKeyPress="WriteArabic()" style="$ustyle">
      <input type="button" value="Add" name="B2_$i"
onClick="Add(EngSrvcProductAdd$i,EngSrvcProduct$row[2],
        UrduSrvcProductAdd$i,UrduSrvcProduct$row[2])">
    </td>
  </tr>
Page;

$crntForm .= <<< form
  <input type="hidden" name="EngSrvcProduct$row[2]"><input type="hidden"
name="UrduSrvcProduct$row[2]">
  <input type="hidden" name="EngSrvcProductAdd$i"><input type="hidden"
name="UrduSrvcProductAdd$i">
  <input type="hidden" name="filler">
form;

break;}}

echo '</table><p align="center"><input type="submit" value="Thankyou!" name="B1"
disabled="true"></p></form>';
if($s) echo "<script>SignUpForm.B1.disabled=false;</script>";
echo $crntForm;
function ChangeSrvcCategory(AList,BList)
{
        BList.options[AList.selectedIndex].selected=true;
        window.status="Please wait while Service Categories are loaded";
        HidForm.action='$PageID.php'; ConvertCode(SignUpForm); HidForm.submit();
}
function OnChangeSrvcCategory(AList,BList)
{
        ChangeMltplSelList(AList,BList);
        if(SignUpForm.B1.disabled==false)
        {       SignUpForm.B1.disabled=true; alert("You need to press the button again to
Show the Lists");       }
}
function ChangeMltplSelList(AList,BList)
{
        for(i=0;i<AList.length;i++) BList.options[i].selected=AList.options[i].selected;
}
function ShowSrvcProductLists()
{
        if(SignUpForm.EngSrvcCategory.selectedIndex==-1)
        { alert("Please first Select any Service Categories"); return false; }
        else
if(SignUpForm.EngSrvcCategory.options[SignUpForm.EngSrvcCategory.selectedIndex].value
=="-1")
        { alert("You need to Add a 'Service Category' under chosen 'Service'"); return false; }
        window.status="Please wait while Service/Product Lists are loaded";
        HidForm.action='$PageID.php'; ConvertCode(SignUpForm); HidForm.submit();
}
```

Following code performs the task in which data is submitted to be saved in database:

```
$crntForm = <<< form
<form method="POST" action="" name="HidForm">
  <input type="hidden" name="EngService"><input type="hidden" name="UrduService">
  <input type="hidden" name="EngServiceAdd"><input type="hidden"
name="UrduServiceAdd">
  <input type="hidden" name="filler">
  <input type="hidden" name="EngSrvcCategory"><input type="hidden"
name="UrduSrvcCategory">
  <input type="hidden" name="EngSrvcCategoryAdd"><input type="hidden"
name="UrduSrvcCategoryAdd">
  <input type="hidden" name="filler"><input type="hidden" name="filler">
form;

echo <<< Page
<html><head><title>Service Providers' SignUp Page</title>
<script>
function Check()
{
 for(i=0;i<SignUpForm.length;i++) {
        if( (SignUpForm[i].type=="select-one" &&

        SignUpForm[i].options[SignUpForm[i].selectedIndex].text.substr(0,6)=='Select') ||
                (SignUpForm[i].type=="select-multiple" && SignUpForm[i].selectedIndex==-1)
){
         alert("The field "+ SignUpForm[i].name.substr(3,SignUpForm[i].name.length) +"" is
empty");
         return false;
        }
 }
 HidForm.action=SignUpForm.action; ConvertCode(SignUpForm);
 for(i=11;i<SignUpForm.length-5;i++)
        if(SignUpForm[i].name.substr(0,14)=='EngSrvcProduct')
        {       HidForm.EngSrvcProduct.value+=HidForm[i].value;
HidForm[i].value=HidForm[i+1].value=''; }
 HidForm.submit(); return false;
}
```

The Alternative Courses that handle the situation in which added element is empty or already exists. Prompt to Service Provider and discontinue the operation is also implemented in "SrvcPrvdrForm.php":

```
function Add(Engitem,Englist,Urduitem,Urdulist)
{
        reInsert=false;
        for(j=0;j<Englist.length;j++)
        if(Englist.options[j].text.toLowerCase()==Engitem.value.toLowerCase() ||
                Urdulist.options[j].text==Urduitem.value) reInsert=true;
        ListName=Englist.name.substr(3,Englist.name.length);
        if( (ListName=='City' || ListName=='SubArea') &&
                SignUpForm.EngCountry.selectedIndex==0)
                alert('Please Select a Country first');
        else if( Engitem.value=='' || Urduitem.value=='')
                alert("Can't add empty value to "+ ListName +"");
        else if (reInsert) alert("The value already exists in "+ ListName +"");
        else { if(ListName=='City') Englist.options[Englist.selectedIndex].value=-1;
                HidForm.action='$PageID.php?add='+ListName;
                window.status="Please wait while "+ ListName +" is added";
                ConvertCode(SignUpForm);     HidForm.submit();
        }
}
Page;

if($_GET["add"]!='')
 switch ($_GET["add"]) {
        case 'City': mysql_query( "Insert Cities (EngCity,UrduCity,CountryNo) ".
                "values
('$_POST[EngCityAdd]','$_POST[UrduCityAdd]',$_POST[EngCountry])" );
                echo 'Post["EngCity"]=Post["EngCityAdd"];
Post["UrduCity"]=Post["UrduCityAdd"];';  break;
        case 'SubArea': mysql_query( "Insert SubAreas
(EngSubArea,UrduSubArea,CitySerNo) ".
                "values
('$_POST[EngSubAreaAdd]','$_POST[UrduSubAreaAdd]',$_POST[EngCity])" );
                echo 'Post["EngSubArea"]=Post["EngSubAreaAdd"];
Post["UrduSubArea"]=Post["UrduSubAreaAdd"];';       break;
};
"SrvcForm.php":
```

```
function Add(Engitem,Englist,Urduitem,Urdulist)
{
        reInsert=false;
        for(j=0;j<Englist.length;j++)
if(Englist.options[j].text.toLowerCase()==Engitem.value.toLowerCase() ||
                Urdulist.options[j].text==Urduitem.value) reInsert=true;
        ListName=Englist.name.substr(3,Englist.name.length);
        if( (ListName=='SrvcCategory' || ListName.substr(0,11)=='SrvcProduct') &&
                SignUpForm.EngService.selectedIndex==0)              alert('Please Select
a Service first');
        else if( Engitem.value=='' || Urduitem.value=='')
                alert("Can't add empty value to '"+ ListName +"'");
        else if (reInsert) alert("The value already exists in '"+ ListName +"'");
        else {  HidForm.action='$PageID.php?add='+ListName;
                window.status="Please wait while "+ ListName +" is added";
                ConvertCode(SignUpForm);
                if(ListName.substr(0,11)=='SrvcProduct') {
                        HidForm.action='$PageID.php?add='+ListName.substr(0,11);

        HidForm.SrvcCategorySerNo.value=Englist.name.substr(14,Englist.name.length);
                        HidForm.EngSrvcProductAdd.value=Engitem.value;
                        HidForm.UrduSrvcProductAdd.value=C(Urduitem.value);
                }
                else if(ListName=='Service') HidForm.EngService.value=-1;
                HidForm.submit();
        }
}
Page;

if($_GET["add"]!='')
  switch ($_GET["add"]) {
        case 'Service': mysql_query( "Insert Services (EngService,UrduService) ".
                "values ('$_POST[EngServiceAdd]','$_POST[UrduServiceAdd]')" );
                echo 'Post["EngService"]=Post["EngServiceAdd"];'.
                        'Post["UrduService"]=Post["UrduServiceAdd"];';  break;
        case 'SrvcCategory': mysql_query( "Insert SrvcCategories
(EngSrvcCategory,UrduSrvcCategory,SrvcSerNo) ".
                "values
('$_POST[EngSrvcCategoryAdd]','$_POST[UrduSrvcCategoryAdd]',$_POST[EngService])" );
                $_POST["EngSrvcCategory"] .= mysql_insert_id()."#"; // to output empty
product lists,to be added prdcts
                echo 'Post["EngSrvcCategory"]+=Post["EngSrvcCategoryAdd"]+"#";'.
                        'Post["UrduSrvcCategory"]+=Post["UrduSrvcCategoryAdd"]+"~e#";';
        break;
        case 'SrvcProduct': mysql_query( "Insert SrvcProducts
(EngSrvcProduct,UrduSrvcProduct,SrvcCategorySerNo) ".
                "values
('$_POST[EngSrvcProductAdd]','$_POST[UrduSrvcProductAdd]',$_POST[SrvcCategorySerN
o])" );
                echo
'Post["EngSrvcProduct'.$_POST["SrvcCategorySerNo"].'"]+=Post["EngSrvcProductAdd"]+"#";'
.

        'Post["UrduSrvcProduct'.$_POST["SrvcCategorySerNo"].'"]+=Post["UrduSrvcProduct
Add"]+"~e#";'; break;
};
```

Another alternative curse in which password fields or specified location fails to comply. Inform the Service Provider and abort submitting is available in "SrvcForm.php":

```
function Check()
 for(i=0;i<SignUpForm.length;i++) {
        if( (SignUpForm[i].type=="select-one" &&

        SignUpForm[i].options[SignUpForm[i].selectedIndex].text.substr(0,6)=='Select') ||
                (SignUpForm[i].type=="select-multiple" && SignUpForm[i].selectedIndex==-1)
){
         alert("The field "'+ SignUpForm[i].name.substr(3,SignUpForm[i].name.length) +'" is
empty");
         return false;
        }
 }
"SrvcPrvdrForm.php":
if(SignUpForm.UrduPasswd.value!=SignUpForm.UrduCnfrmPasswd.value)
        { alert("Password fields do not match"); return false; }
 else
if(SignUpForm.EngSubArea.options[SignUpForm.EngSubArea.selectedIndex].value=="-1")
 { alert("You need to Add a 'SubArea' under chosen 'City'"); return false; }
```

6.2.9 Login

Login procedure for different types of users is implemented in separate files namely "Home.php", "Doctors.php", "Patients.php", and "SrvcPrvdrs.php". The main page "Home.php" implements the common login tasks, examines User type and steers to respective login page and escorts the user to the login page according to selected user-type:

```
lognlnks = new Array("login/Patients.php","login/Doctors.php","login/SrvcPrvdrs.php");
function Check(lnktyp)
{
        i=0; usrtyp=ValueRetriever(UsrTypForm[i],UsrTypForm);
        if(usrtyp=='') {alert("Kindly Choose a User Type");return false;}
        else {
                if(lnktyp==1) {
                        if(LoginForm[0].value=='' || LoginForm[1].value=='')
                        {alert("Provide both LoginID & Password");return false;}
                        else {
                                HidForm.action=lognlnks[usrtyp]; ConvertCode(LoginForm);
                                HidForm.submit(); return false;
```

The submitted data is temporarily stored while checking the given ID and Password against available entries in the database for User's authentication.

```
<form method="POST" action="" name="HidForm">
  <input type="hidden" name="UrduId"><input type="hidden" name="UrduPasswd"><input
type="hidden" name="filler">
</form>
```

```
$result=mysql_query("select EngName,UrduName,EngAvgTime,EngToTime from Doctors
where UrduId='$_POST[UrduId]' ".
        "and UrduPasswd=password('$_POST[UrduPasswd]') ");

if( mysql_num_rows($result)==0 )        echo <<< Page
<font $hsize color=$hclr>
        Please re-enter your Id and Password to identify yourself as Doctor</font>
```

In case the ID and Password do not match any available database entry then inform the user and provide a form to enter ID and Password again for approved authentication.

```
<form method="POST" action="$PageID.php" name="LoginForm" onSubmit="return
Check()">
 <table border="0" width="100%">
    <tr>
      <td width="25%"><p align="center">DoctorID:</td>
      <td width="25%"><input type="text" name="UrduId" size="15" dir="rtl"
                onKeyPress="WriteArabic()" style="$ustyle"></td>
      <td width="25%"><p align="center">Password:</td>
      <td width="25%"><input type="text" name="UrduPasswd" size="15" dir="rtl"
                onKeyPress="WriteArabic()" style="$ustyle"></td>
    </tr>
        </table>
   <p align="center"><input type="submit" value=" OK " name="B1"></td>
        </form>
<script>ConvertBack(LoginForm);</script>
```

6.2.10 Doctor Home Page

After successful login, the doctor is taken to a customized dynamic page specific to that doctor. It is implemented by "Doctors.php". It greets the Doctor and displays patients' queue for that day showing their names, appointment times and status.

```
else {  $drrow = mysql_fetch_row($result);       $stts =
array("Offline","Online","Waiting","Being Examined");
                $fc="$font"; $strt='<script>D("'; $endd='");</script>';

echo <<< Page
        <table width="100%"><tr><td><font $hsize color=$hclr><b>Hello Doctor </b>
          <font color=$sbhclr>$drrow[0]</font>,<font color=$sbhclr
face="$fc">$strt$drrow[1]$endd</td>
          <td align="right"><font $hsize color=$hclr><b>LoginID: </b>
                <font color=$sbhclr face="$fc">$strt$_POST[UrduId]$endd</td></tr>
        </table><p></p>
Page;
// Queue
echo "<p><font $sbhsize color=$hclr>Patients' Queue, ".date('l d-F-Y g:i A')."</font></p>";
<table width=100% style="font-size:$tblsz">
  <tr><td><font $sbhsize color=$hclr><i>Patient Name</td>
        <td align="center"><font $sbhsize color=$hclr><i>Appointment Time</td>
        <td align="center"><font $sbhsize color=$hclr><i>Status</td>
  </tr>
Page;

$result=mysql_query("select PatientId,ApntmntDtTm,Old from Apntmnts ".
        "where (DrId='$_POST[UrduId]' and date_format(ApntmntDtTm,'%Y-%m-
%d')=current_date()) ".
        "order by ApntmntDtTm");

$lnkcnt=0;
while($row=mysql_fetch_row($result)) { echo "<tr><td>";
  $rp = mysql_fetch_row(mysql_query("select Name,Status from Patients where
Id='$row[0]'"));
  if($rp[1]==2 && $row[2]=='N')   { $lnkcnt++; $nxt=true; echo <<< Link
        <a href="javascript:Apnt('$row[0]','$row[1]')">
Link;
  }
  echo "<font face='$fc'>$strt$rp[0]$endd</td><td align=center>";
  $rt = mysql_fetch_row( mysql_query(" select time_format('$row[1]','%r'),".
        "time_format( ('$row[1]' + interval '$drrow[2]' hour_second) - interval 1 minute,'%r' )
"));
  echo "$rt[0] To $rt[1]</td><td align=center>".$stts[$rp[1]]."</td></tr>";
}
echo "</table>";

echo <<< Page
<form method="POST" action="" name="NextForm"
onSubmit="NextForm.action=links[links.length-$lnkcnt].href">
  <p align="center"><input type="submit" value="Next >>>" name=B1>
</form><script>if ($lnkcnt==0) NextForm.B1.disabled=true;</script>
Page;
} // outermost else
```

The facility to initiate *Investigate the Patient* is implemented on the this page via the following code:

```
<script>
function Apnt(patientId,apntmntdtm)
{
        ApntForm.PatientId.value=patientId; ApntForm.ApntmntDtTm.value=apntmntdtm;
        ApntForm.target+=patientId;ApntForm.submit();
}
<form method="POST" action="doctors/PatientInfo.php" name="ApntForm"
target="$_POST[UrduId]DrApntnewin">
  <input type="hidden" name="UrduId" value="$_POST[UrduId]"><input type="hidden"
name="UrduPasswd">
  <input type="hidden" name="PatientId"><input type="hidden" name="ApntmntDtTm">
</form>
```

Following code manages 'patient name' links and shows their current status, with the passage of time.

```
window.setTimeout("ApntForm.action='$PageID.php';ApntForm.target='';"+
        "ApntForm.UrduPasswd.value='$_POST[UrduPasswd]';
ApntForm.submit();",1000*60*1);
</script>
```

6.2.11 Investigate the Patient

This part of code is related to establishing a medical consultancy session between doctor and patient, It is implemented in "PatientInfo.php", "Investigation.php", "Diagnosis.php", and "QstionrInit.php". Initially it displays patient's personal info like name, calculated age, language etc. and medical info including any previous visits and doctors' remarks.

```
mysql_query(" update Patients set Status='3' where Id='$_POST[PatientId]' ");

$row=mysql_fetch_row(mysql_query(" select
Name,Country,City,StrtAdrs,Prfsion,Rlgion,Gndr,".
        "@age:=(to_days(curdate())-to_days(DtofBrth))/365, lang, ".
        "@yrs:=truncate(@age,0), truncate((@age-@yrs)*12,0) from Patients where
Id='$_POST[PatientId]' "));
$r=mysql_fetch_row(mysql_query("select $row[8]Country from Countries where
CountryNo=$row[1]"));

if($row[8]=='Eng') echo "<script>for(i=0;i<KyBrdPanel.length;i++)
KyBrdPanel[i].disabled=true;</script>";
else {   $_GET["lang"]=$row[8] ;include '../../includes/mndyrlang.php';      }

echo <<< Page
<html><head><title>$PageID Page</title></head><body>
<table width=100% style="font-size:$tblsz">
  <tr><td align="center" colspan=3><font $hsize color=$hclr>Attended Patient:
        <font color=$sbhclr face="$face">$startdelim$row[0]$enddelim</td></tr>
...
        <td>Age: <font color=$sbhclr>$row[9] years $row[10] months</td>
  </tr>
</table><hr>
<form method="POST" action="Investigation.php" name="HidForm"
        target="$_POST[PatientId]$_POST[UrduId]PtInfonewin">
  <p align=center><input type="submit" value="Proceed with Investigation Session >>>"></p>
  <input type="hidden" name="PatientId" value="$_POST[PatientId]">
  <input type="hidden" name="ptname" value="$row[0]">
  <input type="hidden" name="ptlang" value="$row[8]">
  <input type="hidden" name="ApntmntDtTm" value="$_POST[ApntmntDtTm]">
  <input type=hidden name="UrduId" value="$_POST[UrduId]">
</form>
Page;

// Previous Visits
echo <<< Page
<p><font $sbhsize color=$hclr><u>Previous Visits</u></font></p><table width=100%
style="font-size:$tblsz">
Page;

$result=mysql_query("select date_format(ApntmntDtTm,'%W %D-%M-%Y'),ApntmntDtTm, ".
        "QstionrId,Answr,Prscrption,EngRmrks,PatientCmnt from Apntmnts ".
```

```
 "where (PatientId='$_POST[PatientId]' and DrId='$_POST[UrduId]' and Old='Y') order by
ApntmntDtTm desc");

while($row=mysql_fetch_row($result)) {
        $r=mysql_fetch_row(mysql_query("select EngAvgTime from Doctors where
UrduId='$_POST[UrduId]'"));
        $rt = mysql_fetch_row( mysql_query(" select time_format('$row[1]','%r'),".
          "time_format( ('$row[1]' + interval '$r[0]' hour_second) - interval 1 minute,'%r' ) "));
        echo "<tr><td colspan=2><font color=$hclr><b>$row[0], $rt[0] To
$rt[1]</b></td></tr>";

        $qstionrid=explode('#',$row[2]); $answr=explode('#',$row[3]);
        for($i=0,$answrindx=0;$i < count($qstionrid)-1;$i++) {
          $r=mysql_fetch_row(mysql_query(" select EngQstionr from Qstionrs where
QstionrSerNo='$qstionrid[$i]' "));
          echo "<tr><td colspan=2><font color=$hclr>Questionnaire: </font>$r[0]</td></tr>";

          if( is_integer(strpos($qstionrid[$i],"!")) )
                echo "<tr><td align=center><font color=$sbhclr>unAnswered</td></tr>";

          // report...
        echo <<< Prscrption...
For detailed code of report & Prscrption, consult "Wait in Que – implementation"
// Remarks
echo <<< Page
<p><font $sbhsize color=$hclr><u>Doctors' Remarks</u></font></p><table width=100%
style="font-size:$tblsz">
Page;

$result=mysql_query("select DrId,date_format(ApntmntDtTm,'%d-%m-%y'),EngRmrks from
Apntmnts ".
 "where (PatientId='$_POST[PatientId]' and Old='Y' and EngRmrks!='') order by
ApntmntDtTm desc");

while($row=mysql_fetch_row($result)) {
        $r=mysql_fetch_row(mysql_query("select EngName from Doctors where
UrduId='$row[0]'"));
        echo"<tr><td>$row[1]: <font color=$sbhclr>Dr. $r[0]</font>: $row[2]</td></tr>";
}echo "</table>";
```

The code below facilitates and manages tasks of updating Patient's status, enabling sending of questionnaire, prescription and remarks, displaying Patient's response as answered questionnaires and Patient's comment, and controlling the overall session.

```
<form method="POST" action="" name="HidForm">
  <input type="hidden" name="PatientId" value="$_POST[PatientId]">
  <input type="hidden" name="ptname" value="$_POST[ptname]">
  <input type="hidden" name="ptlang" value="$_POST[ptlang]">
  <input type="hidden" name="ApntmntDtTm" value="$_POST[ApntmntDtTm]">
  <input type=hidden name="UrduId" value="$_POST[UrduId]">
</form>
<FRAMESET COLS="80%,20%"
 onLoad="HidForm.target='DiagnosisFrame'; HidForm.action='Diagnosis.php';
HidForm.submit();
```

```
        HidForm.target='QstionrInitFrame'; HidForm.action='QstionrInit.php';
HidForm.submit();">
  <FRAME NAME="DiagnosisFrame">
  <FRAME NAME="QstionrInitFrame">
</FRAMESET>
"Diagnosis.php":
if ($_POST["ptlang"]=='Eng')
        echo "<script>for(i=0;i<KyBrdPanel.length;i++)
KyBrdPanel[i].disabled=true;</script>";
else {
        $_GET["lang"]=$_POST["ptlang"]; include '../../includes/mndyrlang.php';
        $style=$ustyle; $keypress='WriteArabic()';
}

if ($_POST["vstcmplt"]=="ok") {  $thanks="<br><font color=$sbhclr>says Thanks
Alot!</font>";
        mysql_query(" update Patients set Status='1' where Id='$_POST[PatientId]' ");
}
else if($_POST["prscrpdone"])
  mysql_query(" update Apntmnts set Prscrption='".$_POST["$_POST[ptlang]Prscrption"]."', ".
                   "EngRmrks='$_POST[EngRmrks]', Old='Y' ".
        "where (PatientId='$_POST[PatientId]' and DrId='$_POST[UrduId]' and
ApntmntDtTm='$_POST[ApntmntDtTm]') ");

echo <<< Page
<html><head><title>$PageID Page</title><script>
function Check(vstcmplt)
{
        if(vstcmplt==1) HidForm.vstcmplt.value="ok";
        else if(PrscrpForm[0].value=='') { alert("Specify a Prescription, Please");   return false;
}
        ConvertCode(PrscrpForm);
HidForm.prscrpdone.value=true;HidForm.action=PrscrpForm.action; HidForm.submit();
        return false;
}
timer=window.setTimeout("HidForm.submit();",1000*60*1);
</script></head><body onLoad="ConvertBack(PrscrpForm);window.status='';">
<p align="center"><font $hsize color=$hclr>Attended Patient: <font color=$sbhclr
face="$face">
        $startdelim$_POST[ptname]$enddelim</font>$thanks</p>
<p align="center"><font $sbhsize>Language: <font
color=$sbhclr>$_POST[ptlang]</font></font></font></p>
<p><font $sbhsize color=$hclr><u>Patient's Response</u></font></p><table width=100%
style="font-size:$tblsz">
Page;

// Qstions_Answrs
$r=mysql_fetch_row(mysql_query("select QstionrId,Answr,PatientCmnt from Apntmnts ".
        "where (PatientId='$_POST[PatientId]' and DrId='$_POST[UrduId]' and
ApntmntDtTm='$_POST[ApntmntDtTm]')"));
echo "<tr><td><font $sbhsize>Patient Comment: ".
        "<font face='$face' color=$sbhclr><font
color=$sbhclr>$startdelim$r[2]$enddelim</td></tr>";
if($r[0]!='') {

        $qstionrid=explode('#',$r[0]); $answr=explode('#',$r[1]);
        for($i=0,$answrindx=0;$i < count($qstionrid)-1;$i++) {
          $r=mysql_fetch_row(mysql_query(" select EngQstionr from Qstionrs where
QstionrSerNo='$qstionrid[$i]' "));
          echo "<tr><td><font color=$hclr>Questionnaire: </font>$r[0]</td></tr>";
```

```
            if( is_integer(strpos($qstionrid[$i],"!")) )
                    echo "<tr><td align=center><font color=$sbhclr>unAnswered</td></tr>";
}
else echo "<tr><td><font color=$sbhclr>&nbsp;No Questions as Yet.</td></tr>";

echo <<< Page
</table><hr><table width=100% style="font-size:$tblsz">
<form method="POST" action="$PageID.php" name="PrscrpForm" onSubmit="return
Check()">
<tr>
  <td><font color=$hclr>Precscription, Advice, Next Appointment Description or any Comment
          <textarea rows="10" cols="50" name="$_POST[ptlang]Prscrption" dir=$dir
onKeyPress="$keypress"
            style="$style" onFocus="window.clearTimeout(timer);
                    window.status='Refreshal has Stopped! Continue by Sending
Prescription.'"></textarea></td>
  <td align="center"><font color=$hclr>Remarks (Next Visit Reminders for all Doctors)
          <input type="text" name="EngRmrks" size=30><br>
          <input type="submit" value="Send Prescription/Remarks" name=B2><br>
          <input type="button" value="<<<Visit Completed?>>>" name=B1 disabled=true
onClick="return Check(1)"></td>
</tr></form>
<script>if ('$_POST[prscrpdone]' && '$_POST[vstcmplt]'=='') PrscrpForm.B1.disabled=false;
          if ('$_POST[vstcmplt]'=='ok') { PrscrpForm.B2.disabled=true;
          window.clearTimeout(timer); }</script>
</table>
<form method="POST" action="" name="HidForm">
<input type="hidden" name="$_POST[ptlang]Prscrption" value="
Page;
echo $_POST["$_POST[ptlang]Prscrption"];
echo <<< Page
  "><input type="hidden" name="EngRmrks" value="$_POST[EngRmrks]">
  <input type="hidden" name="filler"><input type="hidden" name="filler">
  <input type="hidden" name="prscrpdone" value="$_POST[prscrpdone]">
  <input type="hidden" name="PatientId" value="$_POST[PatientId]">
  <input type="hidden" name="ptname" value="$_POST[ptname]">
  <input type="hidden" name="ptlang" value="$_POST[ptlang]">
  <input type="hidden" name="ApntmntDtTm" value="$_POST[ApntmntDtTm]">
  <input type=hidden name="UrduId" value="$_POST[UrduId]">
  <input type="hidden" name="vstcmplt">
</form>
```

Following code is included in the file 'QstionrInit.php':

```
  <input type="button" value="Send Questionnaire" name="B3" onClick="return
Check('Edt',1)">&nbsp;&nbsp;&nbsp;
function Check(New_Edt,snd)
  if(snd==1) {
          if (HidForm.SntQstionrId.value.indexOf(HidForm.EngQstionrSerNo.value)!=-1)
          {          alert("This Questionnaire has already been sent! Send another or create new
one."); return false;        }
          HidForm.SntQstionrId.value+=HidForm.EngQstionrSerNo.value+'#';//accumulating
SntQstionrIds in this hidfield
          HidForm.QstionrId.value=HidForm.EngQstionrSerNo.value;
          HidForm.q.value=true;HidForm.action='$PageID.php';
  }
  HidForm.submit();       return false;
}
if ($_POST["QstionrId"]!='') {
  mysql_query(" update Apntmnts set QstionrId=Concat('$_POST[QstionrId]!#',QstionrId) ".
        "where PatientId='$_POST[PatientId]' and DrId='$_POST[UrduId]' and
ApntmntDtTm='$_POST[ApntmntDtTm]' ");
echo <<< Page
  <input type="hidden" name="q"><input type="hidden" name="QstionrId">
  <input type="hidden" name="SntQstionrId" value="$_POST[SntQstionrId]">
  <input type=hidden name="UrduId" value="$_POST[UrduId]">
  <input type="hidden" name="PatientId" value="$_POST[PatientId]">
  <input type="hidden" name="ApntmntDtTm" value="$_POST[ApntmntDtTm]">
</form>
Page;
```

6.2.12 Design Questionnaire

This part of the system is implemented in "QstionrInit.php" and "QstionrEdit.php". The Typical Course of Events includes the task that displays any existing questionnaires showing their names, description and no. of questions, under the chosen disease category and subgroup.

```
function ShowQstionrs(dssbgrp)
{
        if(dssbgrp.options[dssbgrp.selectedIndex].value=="0") return false;
        else if(dssbgrp.options[dssbgrp.selectedIndex].value=="-1")
        { alert("You need to Add a 'Disease SubGroup' under chosen 'Disease Category'");
return false; }
        window.status="Please wait while Questionnaires are loaded";
        HidForm.q.value=true; // to indicate change/addition to DseseSbGrp list
        HidForm.action='$PageID.php'; ConvertCode(QstionrForm); HidForm.submit();
}
$crntForm = <<< form
<form method="POST" action="" name="HidForm">
  <input type="hidden" name="EngDseseCtgry"><input type="hidden"
name="EngDseseCtgryAdd">
  <input type="hidden" name="filler">
  <input type="hidden" name="EngDseseSbGrp"><input type="hidden"
name="EngDseseSbGrpAdd">
  <input type="hidden" name="filler"><input type="hidden" name="filler">
form;
//extra 'filler' for extra '0-value 0-position' radio button, put2avoid presence of single
'undefined' radiobtn
<font $sbhsz2 color="$sbhclr">Design/Modify a Questionnaire/Test Form to diagnose a
disease</font></p>
<form method="POST" action="QstionrEdit.php" name="QstionrForm">
  <table border="0" width="100%">
        <tr>
          <td width="30%">•› Category in which disease lies</td>
          <td width="30%"><select size="1" name="EngDseseCtgry"
onChange="ChangeDseseSbGrp()">
    <option value="99">Select Disease Category
Page;

$result=mysql_query("SELECT EngDseseCtgry,DseseCtgrySerNo FROM DseseCtgries
order by DseseCtgryNo");
while ($row = mysql_fetch_row($result)) echo "<option value=$row[1]>$row[0]";
```

```
echo <<< Page
                    </select></td>
    <td width="13%">(Not Listed?)</td>
    <td width="27%"><input type="text" name="EngDseseCtgryAdd" size="20">
      <input type="button" value="Add"
onClick="Add(EngDseseCtgryAdd,EngDseseCtgry)"></td>
   </tr>
  </table>
  <table border="0" width="100%">
  <tr>
        <td width="30%">•› SubGroup in above category to which the disease
belongs</td><td width="30%">
        <select size="1" name="EngDseseSbGrp"
onChange="ShowQstionrs(EngDseseSbGrp)">
          <option value='0'>Select Disease SubGroup
Page;

if($_POST[EngDseseCtgry]=='') $_POST[EngDseseCtgry]=99;
$result=mysql_query("SELECT EngDseseSbGrp,DseseSbGrpSerNo FROM DseseSbGrps ".
        "where DseseCtgrySerNo='$_POST[EngDseseCtgry]' order by DseseSbGrpNo");
if(mysql_num_rows($result)==0) echo "<option value='-1'>Disease SubGroup Name";
else while ($row = mysql_fetch_row($result)) echo"<option value=$row[1]>$row[0]";

echo <<< Page
                    </select></td>
        <td width="13%">(Not Listed?)</td>
        <td width="27%"><input type="text" name="EngDseseSbGrpAdd" size="20">
          <input type="button" value="Add"
onClick="Add(EngDseseSbGrpAdd,EngDseseSbGrp)">
        </td>
  </tr>
  </table><p></p>
Page;

////////////////////
if($_POST[q]) { echo <<< Page
<p>•› Available Questionnaires/Test Result Form in <script>function ListText(l,n) {
for(i=1;i<l.length;i++)
        if(l.options[i].value==n) {return l.options[i].text; break;} }
 document.write("<b>"+ ListText(QstionrForm.EngDseseCtgry,$_POST[EngDseseCtgry])
+"</b> and <b>"+
        ListText(QstionrForm.EngDseseSbGrp,$_POST[EngDseseSbGrp]) +"</b>")
</script></p>
 <table border="0" width="100%">
 <tr><td width="5%"><i>slctd<input type="radio" name="EngQstionrSerNo"
value="0">?</i></td>
        <td width="30%"><i>Questionnaire Name</i></td>
        <td width="60%"><i>Description</i></td><td width="5%"><i>Qns.</i></td>
 </tr>
 </table><hr>
 <table border="0" width="100%" cellspacing="0">
Page;

$result=mysql_query("SELECT EngQstionr,EngQstionrDscrp,QstionrSerNo FROM Qstionrs ".
        "where DseseSbGrpSerNo='$_POST[EngDseseSbGrp]' order by QstionrNo");
        // to set edt btn disabled
if(mysql_num_rows($result)==0) { echo "None. Kindly create new Questionnaire.";
$nedt=true; }
else while ($row = mysql_fetch_row($result)) {
 $r=mysql_fetch_row(mysql_query(" select count(QstionrSerNo) from Qstions where
QstionrSerNo=$row[2] "));
 echo <<< Page
        <tr><td width="5%" align=center><input type="radio" name="EngQstionrSerNo"
value="$row[2]"></td>
         <td width="30%">$row[0]</td><td width="60%" style="border-left-
style:solid">$row[1]</td>
         <td width="5%" align=center style="border-left-style:solid">$r[0]</td>
        </tr>
Page;

 if($_POST["QstionrId"]==$row[2])
        echo "<tr><td colspan=3><blockquote><font color=$hclr>Sent to Patient. Wait for
Response!</td></tr>";

 $crntForm .= '<input type="hidden" name="filler">';
}
$crntForm = substr_replace($crntForm,'<input type="hidden" name="EngQstionrSerNo">',-
35);

echo <<< Page
</table><p align="center">
 <input type="button" value="Create New" name="B1" onClick="return Check('New')">
 <input type="button" value="View/Edit selected one" name="B2" onClick="return
Check('Edt')"></p>
 <script>if('$nedt') // conditions kept mostly false reducing code execution
         { QstionrForm.B2.disabled=true;          QstionrForm.B3.disabled=true; }
        if('$_POST[PatientId]'=='') QstionrForm.B3.disabled=true;</script>
Page;
$crntForm.="<input type=hidden name=filler><input type=hidden name=filler><input
type=hidden name=filler>";

} //end if q

echo "</form>$crntForm";
echo <<< Page
 <input type="hidden" name="q">
```

Following code relates to the course of events that opens and displays the questionnaire under the chosen disease category and subgroup, examining if its being edited or created:

a. If being viewed or edited, displays the name and description, and any existing questions, with facility to append these to the selected questionnaire.

b. If being created, provides input fields for name and description, and for the questions and their single or multiple choice answers, in both English and Urdu.

```
if (New_Edt=='Edt') { t=false;
        for(i=6;i<QstionrForm.length-3;i++)
                if(QstionrForm[i].checked) { t=true; break; }
                if(!t) { alert("Questionnaire has not been selected"); return false; }
 }
```

```
HidForm.action=QstionrForm.action+'?New_Edt='+New_Edt; ConvertCode(QstionrForm);
'QstionrEdit.php':
<font $sbhsz2 color="$sbhclr">Add Questions to Questionnaire</font></p>
Page;

$result=mysql_query("SELECT EngDseseCtgry FROM DseseCtgries where
DseseCtgrySerNo='$_POST[EngDseseCtgry]'");
$row = mysql_fetch_row($result); echo "•› Category of disease to diagnose:
<b>$row[0]</b><br>";
$result=mysql_query("SELECT EngDseseSbGrp FROM DseseSbGrps where
DseseSbGrpSerNo='$_POST[EngDseseSbGrp]'");
$row = mysql_fetch_row($result); echo "•› SubGroup in this category: <b>$row[0]</b>";

echo <<< Page
<form method="POST" action="$PageID.php" name="QstionrForm" onSubmit="return
Check()">
 <table border="0" width="100%">
        <tr><td width="40%">•› Questionnaire's/Test Result Form's name</td>
          <td width="60%"><input type="text" name="EngQstionr" size="30"></td>
        </tr>
        <tr><td width="40%">•› Brief Description</td>
          <td width="60%"><input type="text" name="EngQstionrDscrp" size="60"></td>
        </tr>
 </table><p></p>
Page;

$result=mysql_query("SELECT EngQstionr,EngQstionrDscrp FROM Qstionrs ".
        "where QstionrSerNo='$_POST[EngQstionrSerNo]'");
$row = mysql_fetch_row($result);
if($_GET["New_Edt"]!='New')
 echo <<< Dsabl
        <script>QstionrForm.EngQstionr.value="$row[0]";
        QstionrForm.EngQstionr.disabled=true;
        QstionrForm.EngQstionrDscrp.value="$row[1]";
        QstionrForm.EngQstionrDscrp.disabled=true;</script>
Dsabl;

echo <<< Page
<p>•› Available Question(s):</p>
 <table border="0" width="100%">
 <tr><td align="center" width="45%"><i>Question</i></td>
        <td width="45%"><i>Multiple Choices</i></td><td width="10%"><i>Single
Answer?</i></td>
 </tr>
 </table><hr>
 <table border="0" width="100%" style="font-family:$face" cellspacing="0">
Page;

$qreInsert="";
$result=mysql_query("SELECT
EngQstion,UrduQstion,EngMltplChc,UrduMltplChc,EngSnglAnswr FROM Qstions ".
        "where QstionrSerNo='$_POST[EngQstionrSerNo]' order by QstionNo");
if(mysql_num_rows($result)==0 || $_GET["New_Edt"]=='New') echo "None. Kindly add
Question(s).";
else while ($row = mysql_fetch_row($result)) {
        echo "<tr><td width=45% style=border-right-style:ridge;border-bottom-
style:dotted>$row[0]".
                "<p align=right>$startdelim$row[1]$enddelim</td><td width=45%
style=border-bottom-style:dotted>";
        echo substr(str_replace("#",", ",$row[2]),0,-2)."<p align=right>$startdelim";
        echo substr(str_replace("#","+(",$row[3]),0,-2)."$enddelim&nbsp;</td><td
width=10%".
                " style=border-left-style:dashed>";
        echo $row[4]=='Y'?'<p align=center><font
face='Wingdings'>ü </font></td></tr>':"&nbsp;</td></tr>";
        $qreInsert.=$row[0].'#'.$row[1].'#';
}

echo <<< Page
  </table><script>qreInsert="$qreInsert";</script><hr><p align="center">
        <input type="button" value="  Done/Exit  "
onClick="BackForm.action='QstionrInit.php';BackForm.submit();">
</p><p>•› Add new Question (Please add in both English & Urdu)</p>
  <table border="0" width="100%">
        <tr><td width="90%">Question&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text"
name="EngQstion" size="65">
          <p><input type="text" name="UrduQstion" size="65" dir="rtl"
onKeyPress="WriteArabic()" style="$ustyle">
                &nbsp;&nbsp;&nbsp;&#1587;&#1608;&#1575;&#1604;</p></td>
          <td width="10%"><input type="submit" value=" OK "></td>
        </tr>
  </table>
  <table border="0" width="100%">
        <tr><td width="20%">Answer's Type:</td>
          <td width="25%"><input type="radio" name="EngSnglAnswr" value="N"
                onClick="for(i=QstionrForm.length-12;i<QstionrForm.length;i++)
QstionrForm[i].disabled=false">
                        Multiple Choice</td>
          <td width="55%"><input type="radio" name="EngSnglAnswr" value="Y"
                onClick="for(i=QstionrForm.length-12;i<QstionrForm.length;i++) {
QstionrForm[i].disabled=true;
                        QstionrForm[i].value=''; }"> Single Answer</td>
        </tr>
  </table>
<blockquote>Possible Options (If no. of choices are less, leave last ones blank)</blockquote>
```

An alternative course of events is triggered when added element is empty or already exists. Prompt to Doctor and discontinue the operation.

```
function Add(Engitem,Englist)
{
        reInsert=false;
        for(j=0;j<Englist.length;j++)
if(Englist.options[j].text.toLowerCase()==Engitem.value.toLowerCase())
                reInsert=true;
        ListName=Englist.name.substr(3,Englist.name.length);
        if(ListName=='DseseSbGrp' && QstionrForm.EngDseseCtgry.selectedIndex==0)
                alert('Please Select a Disease Category first');
        else if(Engitem.value=='')        alert("Can't add empty value to '"+ ListName +"'");
        else if (reInsert) alert("The value already exists in '"+ ListName +"'");
        else {  if(ListName=='DseseSbGrp') HidForm.q.value=true; // to indicate
change/addition to this list
                Englist.options[Englist.selectedIndex].value=-1;
                HidForm.action='$PageID.php?add='+ListName;
                window.status="Please wait while "+ ListName +" is added";
                ConvertCode(QstionrForm);     HidForm.submit();
        }
}
Page;
```

```
if($_GET["add"]!='')
  switch ($_GET["add"]) {
          case 'DseseCtgry': mysql_query("Insert DseseCtgries(EngDseseCtgry)
values('$_POST[EngDseseCtgryAdd]')" );
                    echo 'Post["EngDseseCtgry"]=Post["EngDseseCtgryAdd"];';        break;
          case 'DseseSbGrp': mysql_query("Insert
DseseSbGrps(EngDseseSbGrp,DseseCtgrySerNo) ".
                    "values('$_POST[EngDseseSbGrpAdd]',$_POST[EngDseseCtgry])" );
                    $_POST["EngDseseSbGrp"] = mysql_insert_id(); // to output added SbGrp's
name
                    echo 'Post["EngDseseSbGrp"]=Post["EngDseseSbGrpAdd"];';     break;
};
```

Another alternative course of events is taken when the appended question already exists in the database. Prompt to Doctor and abort appending.

```
  if ( qreInsert.indexOf(QstionrForm.EngQstion.value.toLowerCase())!=-1 ||
          qreInsert.indexOf(C(QstionrForm.UrduQstion.value))!=-1 )
          { alert("Sorry, this Question already exists"); return false; }
          $qreInsert.=$row[0].'#'.$row[1].'#';
if($_GET["add"]!='')
  switch ($_GET["add"]) {
          case 'Qstion': if($_POST["New_Edt"]=='New')
           {        mysql_query("Insert
Qstionrs(EngQstionr,EngQstionrDscrp,DseseSbGrpSerNo) ". "values('$_POST[EngQstionr]',
'$_POST[EngQstionrDscrp]', $_POST[EngDseseSbGrp])" ); $_POST["EngQstionrSerNo"] =
mysql_insert_id();
           }
           mysql_query("Insert
Qstions(EngQstion,UrduQstion,EngMltplChc,UrduMltplChc,EngSnglAnswr,QstionrSerNo) ".
          "values('$_POST[EngQstion]','$_POST[UrduQstion]',
          '$_POST[EngMltplChc]','$_POST[UrduMltplChc]',".
          "'$_POST[EngSnglAnswr]',$_POST[EngQstionrSerNo])" );          break;
};
```

For multiple choice answer type, no of choices is less than two or the choices are haphazard among English-Urdu fields. Inform the Doctor and abort the append operation.

```
  if (QstionrForm.EngSnglAnswr[0].checked)        { count=0;
          for(i=QstionrForm.length-12;i<QstionrForm.length-1;i+=2)         {
            engfil=QstionrForm[i].value?1:0; nengfil=QstionrForm[i+1].value?1:0;
            if (engfil ^ nengfil)        {          alert("Both fields required for "+
                    QstionrForm[i].name.substr(3,QstionrForm[i].name.length));       return false;
}
            else if (engfil & nengfil) {          count++;
                    HidForm.EngMltplChc.value+=HidForm[i].value+"#";
                    HidForm.UrduMltplChc.value+=HidForm[i+1].value+"#";
            }
          }
          if (count<2)       { alert("Provide atleast two choices"); return false; }
  }
```

```
 $r=mysql_fetch_row(mysql_query("select $_GET[lang]Name,EngToTime from Doctors
where UrduId='$row[0]'"));
 echo "<tr><td width=45%>&nbsp;$startdelim$texts[3]$enddelim ";        $late="";

 if(date("Y-m-d H:i:00")>="$row[7]-$row[6]-$row[5] $r[1]")
         $late="<tr><td align=$dir2><font color=$hclr
size=3>$startdelim$texts[9]$enddelim</td></tr>";
 else if(date("dmY")=="$row[5]$row[6]$row[7]")  echo <<< Link
         <a href="javascript:Apnt('$row[0]','$row[1]')">
Link;

 echo "<i>$startdelim$r[0]$enddelim</i></a></td><td width=35%>". //+0 to cnvrt char-
>numeric
         "$startdelim".$days[$row[4]]."$enddelim $row[5]-
$startdelim".$months[$row[6]+0]."$enddelim-$row[7]".
         "</td><td width=20%>$row[8]</td></tr>$late";
} echo "</table><p></p>";
```

3. Displays any previous visits indicating if there are 'pending' questionnaires to be answered.

```
// Previous Visits
echo <<< Page
<font $sbhsize color=$hclr><u>$startdelim$texts[4]$enddelim</u></font><p></p>
Page;

if( mysql_num_rows($result)==0 ) echo "<p
align=$dir1>&nbsp;&nbsp;&nbsp;$startdelim$texts[11]$enddelim</p>";
else {    mysql_data_seek($result,0); echo "<table width=100% dir=$dir style='font-
size:$tblsz'>";

while($row=mysql_fetch_row($result)) if($row[3]=='Y') {
 $r=mysql_fetch_row(mysql_query("select $_GET[lang]Name from Doctors where
UrduId='$row[0]'"));
 echo "<tr><td width=45%>&nbsp;$startdelim$texts[3]$enddelim ";        $star="";
 echo <<< Link
         <a href="javascript:Apnt('$row[0]','$row[1]',1)">
Link;

 // indicate pending form fill
 if( is_integer(strpos($row[2],"|")) )        { echo "<font color=$hclr>"; $star="*"; }

 echo "<i>$startdelim$r[0]$enddelim</i></a>$star</td><td width=35%>".
         "$startdelim".$days[$row[4]]."$enddelim $row[5]-
$startdelim".$months[$row[6]+0]."$enddelim-$row[7]".
         "</td><td width=20%>$row[8]</td></tr>";
```

Following code allows a patient to activate his / her *Wait in Queue* status.

```
function Apnt(drid,apntmntdtm,prvst)
{
 ApntForm.DrId.value=drid;       ApntForm.ApntmntDtTm.value=apntmntdtm;
         ApntForm.target+=drid;
 if(prvst==1) { ApntForm.action='$PageID.php?lang=$_GET[lang]';        ApntForm.target="";
         ApntForm.UrduPasswd.value='$_POST[UrduPasswd]';
 }
 ApntForm.submit();
}
timer=window.setTimeout("Apnt('$_POST[DrId]','$_POST[ApntmntDtTm]',1);",1000*60*2);
</script>
<form method="POST" action="patients/WaitInQue.php?lang=$_GET[lang]"
name="ApntForm"
```

```
        target="$_POST[Urduld]PtApntnewin">
  <input type="hidden" name="Urduld" value="$_POST[Urduld]"><input type="hidden"
name="UrduPasswd">
  <input type="hidden" name="Drld"><input type="hidden" name="ApntmntDtTm">
  <input type="hidden" name="notfrstm" value="yes">
```

Following code shows questionnaires as a report or as a form to be filled if it's 'pending', also shows the doctor's prescription etc.

```
 // expand the clicked one
 if ($_POST["Drld"]==$row[0] && $_POST["ApntmntDtTm"]==$row[1]) {

$r=mysql_fetch_row(mysql_query("select Answr,Prscrption,EngRmrks,PatientCmnt from
Apntmnts ". "where (Patientld='$_POST[Urduld]' and Drld='$row[0]' and
ApntmntDtTm='$row[1]')"));
        echo "<tr><td colspan=2><font $sbhsz2
color=$sbhclr>$startdelim$texts[5]$enddelim</td></tr>";

        $qstionrid=explode('#',$row[2]); $answr=explode('#',$r[0]);
        for($i=0,$answrindx=0;$i < count($qstionrid)-1;$i++) {
         $rsqn=mysql_query("select
$_GET[lang]Qstion,$_GET[lang]MltplChc,EngSnglAnswr,QstionSerNo ".
        "from Qstions where QstionrSerNo='$qstionrid[$i]' order by QstionSerNo");
 if( is_integer(strpos($qstionrid[$i],"!")) ) {         if($i==0) { //bgn form tags for first one
                echo <<< BgnForm
                if( !is_integer(strpos($qstionrid[$i+1],"!")) )
                 echo <<< EndForm
         // report
         else {
                while($rqn=mysql_fetch_row($rsqn)) {
For detailed code of form & report, consult "Wait in Que – implementation"
        echo <<< Prscrption...
```

6.2.14 Get Appointment

This part of system is implemented by in "PatientLinks.php", "GetApntmnt.php", and "CnfrmApntmnt.php". The Typical Course of Events implemented in "PatientLinks.php" includes the code that allows the user to search a doctor on basis of specialization or name. This part of the code is given below:

```
$ptlinks=array("EdtPrsnlInfo.php","EdtFmlyMdclHstry.php","SearchDrs.php","GetApntmnt.php
","LogOut()");
function Link(link)
{ PatientLink.action='$prextn/login/patients/'+link+'?lang=$_GET[lang]';PatientLink.submit(); }
</script><form method="POST" action="" name="PatientLink" target="PtLinknewin">
  <input type=hidden name="Urduld" value="$_POST[Urduld]">
</form><table width=100% style="font-family:$face" dir=$dir border=2 bgcolor="#00FFCC"
bordercolor="#FFCC99">
Page;

for($i=0;$i < count($ptlinks)-1;$i++)        echo <<< Links
        <td><a href="javascript:Link('$ptlinks[$i]')" style="text-decoration: none">
                $startdelim$pttexts[$i]$enddelim</a></td>
Links;
```

After finding a related doctor the patient is provided with information about doctor's expertise and experience. Next step is to allow the user to choose date for appointment. This portion is implemented in "GetApntmnt.php":

```
if ($_GET["lang"]=='Eng') {
        echo "<script>for(i=0;i<KyBrdPanel.length;i++)
KyBrdPanel[i].disabled=true;</script>";

$texts=array("Dr ","~eSearch by Name","Name","Specialization","Online Time","To",
        "Average Time per Patient","Fee for Visit","Select Date for Appointment",
                "Day","Month","Year","~eConfirm Appointment",
                "No Results. Kindly provide Correct Name.","","hr","mins", "City");
$months = array("","January", "February", "March", "April", "May", "June", "July", "August",
                "September", "October", "November", "December");
} else {
        $style=$ustyle;  $keypress="WriteArabic()"; $C="C";.
function NameCheck()
{
        if(DrNameForm[0].value=='') { alert("Specify a Name, Please");    return false; }
        HidForm.$_GET[lang]DrName.value=$C(DrNameForm[0].value);
        HidForm.action=DrNameForm.action; HidForm.submit(); return false;
}
function ChangeDays()
{...
<form method="POST" action="$PageID.php?lang=$_GET[lang]" name="DrNameForm"
onSubmit="return NameCheck()">
        <p dir=$dir align=$dir2>$startdelim$texts[0]$enddelim
        <input type="text" name="$_GET[lang]DrName" size=30 onKeyPress="$keypress"
style="$style"><br>
        <input type=submit name=B1
style="$style"></p><script>DrNameForm.B1.value=D('$texts[1]',1);</script>
</form>
Page;.

if($_POST["$_GET[lang]DrName"]!='') {
$result=mysql_query("SELECT UrduId FROM Doctors where
$_GET[lang]Name='".$_POST["$_GET[lang]DrName"]."'");
if(mysql_num_rows($result)!=0) { $row = mysql_fetch_row($result); $_POST["DrId"]=$row[0];
}
else $_POST["DrId"]=''; //to avoid any previous dr's info
}

$result = mysql_query(" select $_GET[lang]Name,$_GET[lang]Specialization,".

"time_format(EngFromTime,'%r'),time_format(EngToTime,'%r'),hour(EngAvgTime),minute(En
gAvgTime),".
 "EngAmount,EngUnit, $_GET[lang]City from Doctors where UrduId='$_POST[DrId]' ");

//////////////////
if($_POST["DrId"]!='') {  $row = mysql_fetch_row($result);          $row[7]=substr($row[7],0,1);
echo <<< Page
<blockquote><p align=$dir1><font $sbhsize color=$sbhclr>
        <font $hsize color=$hclr>$startdelim$texts[2]$enddelim:</font>
<b>$startdelim$texts[0]$row[0]$enddelim</b>...
<form method="POST" action="CnfrmApntmnt.php?lang=$_GET[lang]" name="DtApntForm"
onSubmit="return Check()">
 <table width="100%" dir="$dir" style="font-size:$tblsz">
        <tr>
```

```
            <td align="center">$startdelim$texts[8]$enddelim &nbsp;
      <select name="EngDay" size="1" style="$style">
                    <option value="0">$startdelim$texts[9]$enddelim...
            <select name="EngMonth" size="1" onChange="ChangeDays()" style="$style">
                    <option value="0">$startdelim$texts[10]$enddelim...
            <select name="EngYear" size="1" onChange="ChangeDays()" style="$style">
                    <option value="0">$startdelim$texts[11]$enddelim
                    <script>for(i=0;i<3;i++) document.write( "<option>"+ (new Date().getYear()+i)
);</script>
            </select></td>
  </tr>
  </table><p align="center"><input type=submit name=B2 style="$style"></p>
  <script>DtApntForm.EngDay.options[new Date().getDate()].selected=true;
            DtApntForm.EngMonth.options[new Date().getMonth()+1].selected=true;
            DtApntForm.EngYear.options[1].selected=true;
            DtApntForm.B2.value=D('$texts[12]',1);</script>
</form>
Page;

} else echo "<p align=$dir1>$startdelim$texts[13]$enddelim";

echo <<< Page
<form method="POST" action="" name="HidForm">
  <input type="hidden" name="EngDay"><input type="hidden" name="EngMonth">
  <input type="hidden" name="EngYear"><input type="hidden" name="filler">
  <input type="hidden" name="$_GET[lang]DrName" value="$row[0]">
  <input type="hidden" name="DrId" value="$_POST[DrId]">
  <input type=hidden name="UrduId" value="$_POST[UrduId]">
</form>
```

After the patient opts for a date, the system automatically allocates the appointment time keeping in view the appointments of other patients with the same doctor, doctor's online time and other considerations. This part is implemented in "CnfrmApntmnt.php":

```
if ($_GET["lang"]=='Eng') {
            echo "<script>for(i=0;i<KyBrdPanel.length;i++)
KyBrdPanel[i].disabled=true;</script>";

$texts=array("Appointment with Dr","Doctor's existing Appointments, Dated","Patient Name",
            "Time","To","","Your ","~eConfirm","~eChange Date", "None. You are the First one.",
            "Sorry, is not available. Change the Date or Consult another Doctor.",
            "Sorry, You already have an ", "Sorry, Try Next Dates, as Passed Time can't be
Alotted for ");
$months = array("","January", "February", "March", "April", "May", "June", "July", "August",
                    "September", "October", "November", "December");
} else {
            $style=$ustyle;

            $texts=array("~u:4BL(K><h(OJ7+(@47g(b4K_>","~ub4K_>@iMO9O<h(:4BL(AV<h(4
OJ47+(MO>;h","~uM>QC(K4(N4M",//0,1,2...
echo <<< Page
            <html><head><title>$PageID Page</title></head><body><font face="$face"><p
align="center">
            <font $hsize color=$hclr>$startdelim$alrdy$texts[0]$enddelim <font
color=$sbhclr>$startdelim
Page;
```

```
echo $drname=$_POST["$_GET[lang]DrName"];          echo <<< Page
        $enddelim</font></font></p>
        <p align="$dir1"><font $sbhsize color=$hclr>$startdelim$texts[1]$enddelim <font
color=$sbhclr>
        $_POST[EngDay]-$startdelim
Page;
echo $months[$_POST["EngMonth"]+0]."$enddelim-$_POST[EngYear]</font></font></p>";
//+0 to conver char->number

$result=mysql_query("select PatientId,ApntmntDtTm from Apntmnts where
(DrId='$_POST[DrId]' and ".
        "date_format(ApntmntDtTm,'%d%m%Y')='$_POST[EngDay]$_POST[EngMonth]$_P
OST[EngYear]') ".
 "order by ApntmntDtTm");

echo "<table width=100% dir=$dir style='font-size:$tblsz'>";
if(mysql_num_rows($result)==0) {
 echo "<tr><td>$startdelim$texts[9]$enddelim</td></tr>";
 $t=mysql_fetch_row(mysql_query(" select @t:='$_POST[EngYear]-$_POST[EngMonth]-
$_POST[EngDay] $drrow[0]' + ".
        "interval 1 minute, time_format(@t,'%r'), time_to_sec(@t) "));

} else {  echo <<< Page
        <tr><td align="center"><font
color=$sbhclr><b>$startdelim$texts[2]$enddelim</td><td align="center">
          <font color=$sbhclr><b>$startdelim$texts[3]$enddelim</td>
        </tr>
Page;
 while($row=mysql_fetch_row($result))  {
        if($_POST["ApntmntDtTm"]==$row[1]) $f="<font color=$hclr>"; else $f="";
        $r = mysql_fetch_row(mysql_query("select Name,lang from Patients where
Id='$row[0]'"));
        if($r[1]!='Eng') {$fc="$font";$strt='<script>D("';$endd='");</script>';}         else
{$fc=$strt=$endd="";}
        echo "<tr><td align=center>$f<font face='$fc'>$strt$r[0]$endd</td><td align=center>";
        $r = mysql_fetch_row( mysql_query(" select time_format('$row[1]','%r'),".
                 "time_format( ('$row[1]' + interval '$drrow[1]' hour_second) - interval 1
minute,'%r' ) "));
        echo "$f$r[0] $startdelim$texts[4]$enddelim $r[1]
$startdelim$texts[5]$enddelim</td></tr>";
 }
 mysql_data_seek($result,mysql_num_rows($result)-1); $row=mysql_fetch_row($result);
 $t = mysql_fetch_row(mysql_query(" select @t:='$row[1]' + interval '$drrow[1]'
hour_second,".
        "time_format(@t,'%r'), time_to_sec(@t) "));

}
echo "</table><hr>";

if ($_POST["ApntmntDtTm"]==''){

echo "<p align=center><font $hsize color=$hclr>$startdelim$texts[6]$texts[3]$enddelim <font
color=$sbhclr>";
if($t[2]>$drrow[2]) { echo "$startdelim$texts[10]$enddelim"; $dsablcnfrm=true; }
else echo $t[1];
echo <<< Page
</font></font></p><form method="POST" action="$PageID.php?lang=$_GET[lang]"
name="CnfrmForm">
 <p align="center"><input type=button name=B1 style="$style"
onClick="CnfrmForm.submit()">
```

```
  <input type=button name=B2 style="$style" onClick="history.back()"></p>
  <script>CnfrmForm.B1.value=D('$texts[7]',1); CnfrmForm.B1.disabled='$dsablcnfrm';
          CnfrmForm.B2.value=D('$texts[8]',1);</script>
  <input type="hidden" name="ApntmntDtTm" value="$t[0]">
  <input type="hidden" name="EngDay" value="$_POST[EngDay]">
  <input type="hidden" name="EngMonth" value="$_POST[EngMonth]">
  <input type="hidden" name="EngYear" value="$_POST[EngYear]">
  <input type="hidden" name="$_GET[lang]DrName" value="$drname">
  <input type="hidden" name="DrId" value="$_POST[DrId]">
  <input type=hidden name="UrduId" value="$_POST[UrduId]">
</form>
Page;
}

if ($_POST["ApntmntDtTm"]!='') {else     mysql_query("Insert
Apntmnts(PatientId,DrId,ApntmntDtTm, QstionrId,Answr) ".
                    "values('$_POST[UrduId]','$_POST[DrId]','$_POST[ApntmntDtTm]', '','')" );
}
```

The error handling part designed as alternative courses of events include the situation when appointment date chosen is a previous one. The Patient is properly informed about this by the code in "GetApntmnt.php":

```
function Check()
{...
 y=DtApntForm.EngYear.options[DtApntForm.EngYear.selectedIndex].text;
 m=DtApntForm.EngMonth.selectedIndex; d=DtApntForm.EngDay.selectedIndex;
 if( (y<new Date().getYear()) || (y==new Date().getYear() && m<new Date().getMonth()+1) ||
        (m==new Date().getMonth()+1 && d<new Date().getDate()) )
        {
                alert("Appointment Date cannot be a Previous one..."); return false;
        }
...
}
```

In case a patienr does not establish a session with the doctor in appointment time then the entry is deleted from the database by the code in "CnfrmApntmnt.php":

```
// any Existing 'New' Appointment?
if (mysql_num_rows($result)!=0) {        $row=mysql_fetch_row($result);
          if(date("Ymd H:i:00")>="$row[1] $drrow[3]") mysql_query("delete from Apntmnts ".
                   "where (PatientId='$_POST[UrduId]' and DrId='$_POST[DrId]' and Old='N')");
        else {
          $_POST["ApntmntDtTm"]=$row[0]; $_POST["EngYear"]=substr($row[1],0,4);
          $_POST["EngMonth"]=substr($row[1],4,2); $_POST["EngDay"]=substr($row[1],6);
$alrdy=$texts[11];
        }
}
```

Another alternative course of events arises when time slot is not available, or is available owing to less number of patients but doctor's online time for 'today' is over. In this case patient is suggested to change the date or to consult another doctor.

```
else if ($_POST["ApntmntDtTm"]!='')     {
```

```
if(date("Y-m-d H:i:00")>"$_POST[ApntmntDtTm]")
{ $alrdy=$texts[12];echo "<script>window.setTimeout('history.go(-2);',12000);</script>"; }
```

6.2.15 Wait in Queue

The procedure of patient's waiting in queue for the doctor is implemented in "WaitInQue.php". Typical course of events is invoked when the patiento logs in and opens queue of patient for the Doctor on the particular day. Names of patients, appointment times and status (online / offline / attended) is shown.

```
if ($_GET["lang"]=='Eng') {
        echo."<script>for(i=0;i<KyBrdPanel.length;i++)
KyBrdPanel[i].disabled=true;</script>";
        $texts = array("Dr","Patients' Queue for ", "Patient Name","Time","To","","Status",
                "Questionnaires/Suggested Tests","Prescription","Dr's Remark",
                "Kindly wait for Dr's Questionnaires.", "Your Comment","~eSend",
                "Info, Doctor's Online Time is over!","Better Health next time InshaAllah");
        $stts = array("Offline","Online","Waiting","Being Examined");
} else {
        $style=$ustyle;  $keypress="WriteArabic()"; $C="C";
        $texts = array("~ub4K_>","~uM>QCOf(KP(JD4>(5>43i(",
        "~uM>QC(K4(N4M","~uOJ7","~u@i","~u7K",
$result=mysql_query(" select $_GET[lang]Name,EngAvgTime,EngToTime from Doctors
where UrduId='$_POST[DrId]' ");
$drrow = mysql_fetch_row($result); echo <<< Page
<html><head><title>$PageID Page</title></head><body><font face="$face"><p
align="$dir1">
        <font $hsize color=$hclr>$startdelim$texts[1]$texts[0]$enddelim <font color=$sbhclr>
        $startdelim$drrow[0]$enddelim</font>&nbsp;:</font></p>
Page;
// Queue
echo <<< Page
<script>timer=window.setTimeout("QueueForm.submit();",1000*60*1);</script>
<form method="POST" action="$PageID.php?lang=$_GET[lang]" name="QueueForm">
  <input type="hidden" name="UrduId" value="$_POST[UrduId]">
  <input type="hidden" name="DrId" value="$_POST[DrId]">
  <input type="hidden" name="ApntmntDtTm" value="$_POST[ApntmntDtTm]">
  <input type="hidden" name="PatientCmnt" value="$_POST[PatientCmnt]">
  <input type="hidden" name="cmntdone">
</form>
<table width=100% dir=$dir style="font-size:$tblsz">
  <tr><td><font $sbhsize color=$sbhclr>$startdelim$texts[2]$enddelim</td>
        <td align="center"><font $sbhsize color=$sbhclr>$startdelim$texts[3]$enddelim</td>
        <td align="center"><font $sbhsize color=$sbhclr>$startdelim$texts[6]$enddelim</td>
  </tr>
Page;

$result=mysql_query("select PatientId,ApntmntDtTm from Apntmnts ".
        "where (DrId='$_POST[DrId]' and ".

"date_format(ApntmntDtTm,'%d%m%Y')=date_format('$_POST[ApntmntDtTm]','%d%m%Y'))
order by ApntmntDtTm");

while($row=mysql_fetch_row($result))  {
 if($_POST["UrduId"]==$row[0]) $f="<font color=$hclr>"; else $f="";
 $rp = mysql_fetch_row(mysql_query("select Name,Status,lang from Patients where
Id='$row[0]'"));
 if($rp[2]!='Eng') {$fc="$font";$strt='<script>D("';$endd='");</script>';}       else
{$fc=$strt=$endd="";}
 echo "<tr><td>$f<font face='$fc'>$strt$rp[0]$endd</td><td align=center>";
$rt = mysql_fetch_row( mysql_query(" select time_format('$row[1]','%r'),".
"time_format( ('$row[1]' + interval '$drrow[1]' hour_second) - interval 1 minute,'%r' ) "));
 echo "$f$rt[0] $startdelim$texts[4]$enddelim $rt[1] $startdelim$texts[5]$enddelim</td>";
 echo "<td align=center>$f$startdelim".$stts[$rp[1]]."$enddelim</td></tr>";
```

The part of code that changes status of patient and highlights current Patient's name in the queue, while waiting for turn is given below:

```
if($_POST["notfrstm"]=="yes") mysql_query(" update Patients set Status='2' where
Id='$_POST[UrduId]' ");
if($_POST["UrduId"]==$row[0]) $f="<font color=$hclr>"; else $f="";
 $rp = mysql_fetch_row(mysql_query("select Name,Status,lang from Patients where
Id='$row[0]'"));
 if($rp[2]!='Eng') {$fc="$font";$strt='<script>D("';$endd='");</script>';}       else
{$fc=$strt=$endd="";}
 echo "<tr><td>$f<font face='$fc'>$strt$rp[0]$endd</td><td align=center>";
```

Following code makes the patient ready for medical session by changing status and asking to wait for Doctor's questionnaires.

```
if($_POST["UrduId"]==$row[0] && $_POST[ApntmntDtTm]==$row[1]){

if ($rp[1]==3 || $rp[1]==1) {
 $r=mysql_fetch_row(mysql_query("select QstionrId,Answr,Prscrption,EngRmrks from
Apntmnts ". "where (PatientId='$_POST[UrduId]' and DrId='$_POST[DrId]' and
ApntmntDtTm='$_POST[ApntmntDtTm]')"));

 if($r[0]!='') {
        echo "<tr><td colspan=3><font $sbhsize
color=$sbhclr><u>$startdelim$texts[7]$enddelim</td></tr>";
```

Following code facilitates and manages the medical consultancy session by changing patient's status, enabling sending of questionnaire, prescription and remarks, displaying patient's response as answered questionnaires and Patient's comment, and controlling the overall session.

```
$qstionrid=explode('#',$r[0]); $answr=explode('#',$r[1]);
for($i=0,$answrindx=0;$i < count($qstionrid)-1;$i++) {
 $rsqn=mysql_query(
"select $_GET[lang]Qstion,$_GET[lang]MltplChc,EngSnglAnswr,QstionSerNo ".
                "from Qstions where QstionrSerNo='$qstionrid[$i]' order by QstionSerNo");

      if( is_integer(strpos($qstionrid[$i],"!")) ) {
             if($i==0) { echo <<< BgnForm
             <script>
```

```
                function Check()
                {       for(i=0;i<AnswrForm.length;i++)
                         if (AnswrForm[i].type=="text") { if(AnswrForm[i].value=='') {
                                alert(
"The Qustion"+AnswrForm[i].name.substr(8,AnswrForm[i].name.length)+
" is un-Answered");  return false;}
                         }
                         else if (AnswrForm[i].type=="radio") {    t=false;
nm=AnswrForm[i].name;
                        while(AnswrForm[i].name==nm) if (AnswrForm[i++].checked) t=true;
                        i--;
                        if(!t) {
                        alert("The Qustion"+ nm.substr(8,nm.length) +" is un-Answered");
                        return false; }
                         }

HidForm.action=AnswrForm.action; ConvertCode(AnswrForm);
for (i=0;i<HidForm.length-8;i++)  if(HidForm[i].name.substr(0,8)=="EngAnswr") {
                                HidForm.Answr.value+=HidForm[i].value+"#";
HidForm[i].value="";
                        }
HidForm.submit(); return false;
}
if('$_POST[cmntdone]'=='cmntaftrform' || '$_POST[cmntdone]'=='false')
        QueueForm.cmntdone.value=false;
else    window.clearTimeout(timer);</script>
<form method=POST action="$PageID.php?lang=$_GET[lang]" name="AnswrForm"
onSubmit="return Check()">
BgnForm;
$hidform="<form method=POST action='' name='HidForm'>";
} //end form tags on the first one
while($rqn=mysql_fetch_row($rsqn)) {
 if ($rqn[2]=='Y') {
                        echo "<tr><td colspan=3 style='border-bottom-
style:dashed'>$startdelim$rqn[0]$enddelim ".
 "<input type='text' name='EngAnswr$rqn[3]' size=10></td></tr>";
$hidform.="<input type='hidden' name='EngAnswr$rqn[3]'>";
  }
else { echo "<tr><td colspan=3 style='border-bottom-
style:dashed'>$startdelim$rqn[0]$enddelim ";
$mltplchc=explode('#',$rqn[1]);
for ($k=0;$k < count($mltplchc)-1; $k++) {
        echo "<input type='radio' name='EngAnswr$rqn[3]'
        value=$k>$startdelim$mltplchc[$k]$enddelim ";
        $hidform.="<input type='hidden' name='filler'>";
} echo "</td></tr>";
        $hidform = substr_replace($hidform,"<input type='hidden'
        name='EngAnswr$rqn[3]'>",-35);
}
}

if( !is_integer(strpos($qstionrid[$i+1],"!")) )
  echo <<< EndForm
  <tr><td colspan=3 align=center><input type=submit name=B1
  style="$style"></td></tr></form>
<script>AnswrForm.B1.value=D('$texts[12]',1);</script>
$hidform<input type="hidden" name="filler"><input type="hidden" name="Answr">
<input type="hidden" name="QstionrId" value="$r[0]">
<input type="hidden" name="UrduId" value="$_POST[UrduId]">
<input type="hidden" name="DrId" value="$_POST[DrId]">
```

```
        <script>if(typeof(AnswrForm)!="undefined") AnswrForm.B1.disabled=true;
CmntForm.B1.disabled=true;
                window.clearTimeout(timer);</script>
End;

} //end patientId if

}echo "</table>";
if ($_POST["cmntdone"] || $_POST["cmntdone"]=="cmntaftrform")
  mysql_query(" update Apntmnts set PatientCmnt='$_POST[PatientCmnt]' ".
        "where (PatientId='$_POST[UrduId]' and DrId='$_POST[DrId]' and
ApntmntDtTm='$_POST[ApntmntDtTm]') ");

if ($_POST["Answr"]!='')
  mysql_query(" update Apntmnts set Answr=Concat('$_POST[Answr]',Answr), ".
                "QstionrId=Replace(
QstionrId,'$_POST[QstionrId]',Replace('$_POST[QstionrId]','!','') ) ".
        "where (PatientId='$_POST[UrduId]' and DrId='$_POST[DrId]' and
ApntmntDtTm='$_POST[ApntmntDtTm]') ");

if (date("H:i:00")>"$drrow[2]")
        echo "<p align=$dir1><font $hsize
color=$sbhclr>$startdelim$texts[13]$enddelim</font></p>";
```

Chapter 7

Results and Conclusion

7. Results and Conclusion

This chapter will present the results obtained from the experimental application that was developed to test the research. These results will be compared with the objectives of the research in order to evaluate the level of success. Finally the thesis will be concluded with an assessment of work done and possible future directions of research in this area.

7.1 Results of Experimentation

Objectives of this research were described in section 1.4 of this thesis. In order to show and evaluate the results finally obtained we use the same sequence of objectives as portrayed in the beginning. The obtained results related to each objective will be discussed under the heading of that objective and its level of success will be assessed.

7.1.1 Standard Urdu Tools

First objective of this research was to develop a set of standard tools for Urdu data entry in dynamic web applications. Figure 7.1 shows the signup of page of doctor that allows input in both English and Urdu at the same time. Figure 7.2 shows the signup page of patients that allows data entry in Urdu using the developed input controls. Almost all of the standard input controls for web applications are represented in these pages. These tools can be included in any web page for Urdu input as other standard HTML controls. They have successfully communicated Urdu data to and from standard database engines and server side scripts.

7.1.2 Mechanism for Urdu Displays and Data Transmission

The first objective "Urdu Displays" was related to development of a mechanism to display Urdu text on web sites as that of English or other languages. The second one was related to optimized data transmission of Urdu. The methodology of character based Urdu display developed in this research has satisfied both of these objectives. It has rendered three main advantages. Firstly it has set Urdu free of the vendor dependant tools like ActiveX controls. Secondly it has reduced data traffic for the Urdu pages to the level of normal English pages by replacing the Urdu characters with single-byte codes while sending to the network. Thirdly the methodology has opened ways for use of Urdu in all standard database engines, whois registries, and domain name servers. Figure 7.3 shows a page that was generated dynamically using a server side script including values from database.

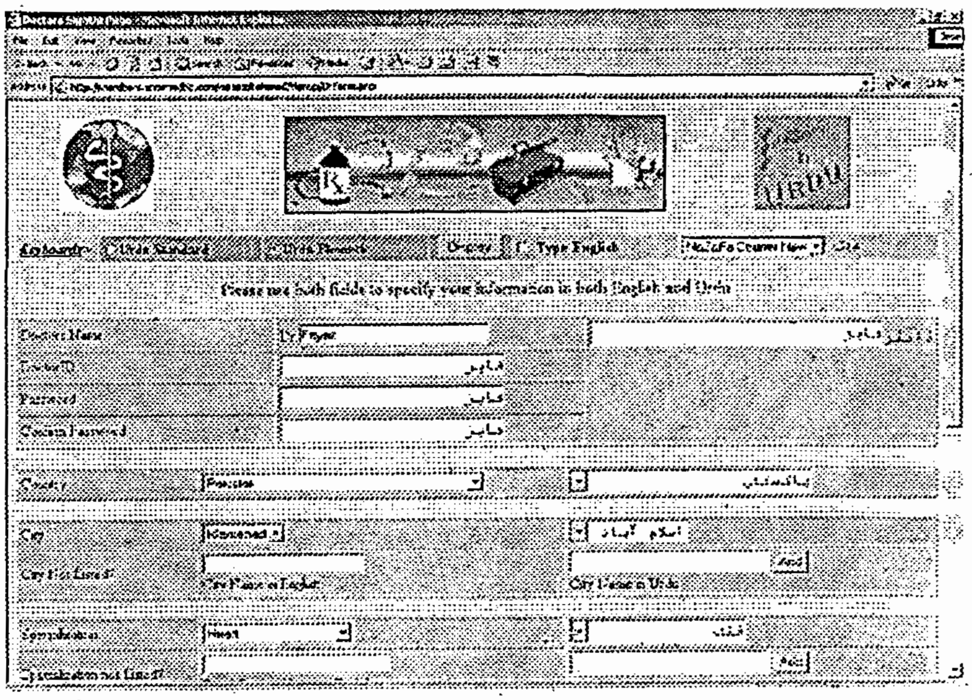


Figure 7.1 Doctor Signup Page

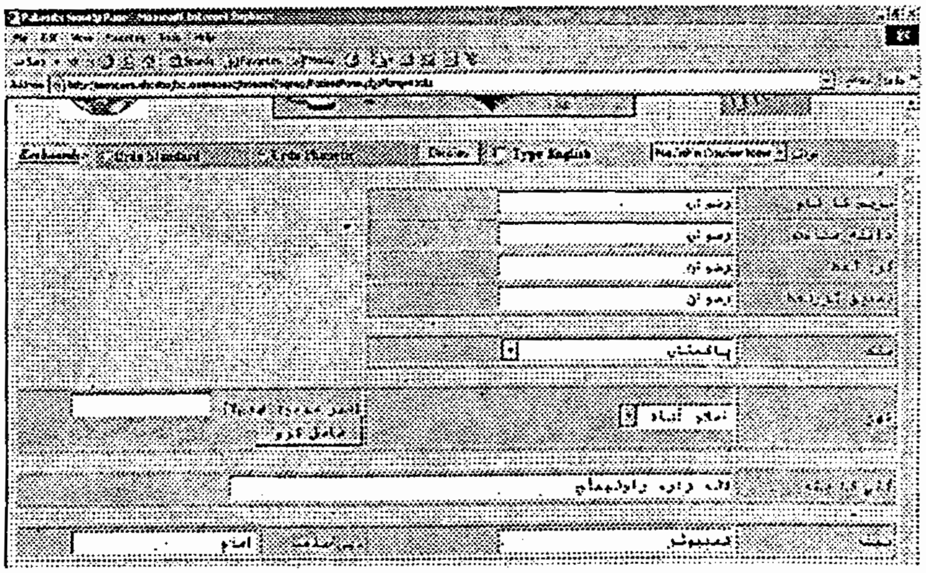
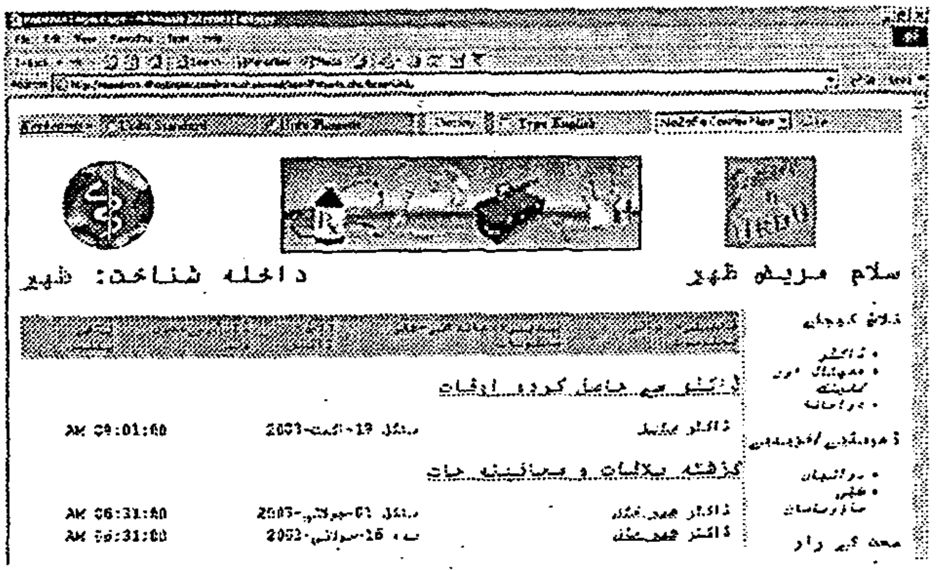


Figure 7.2 Patient Signup Page


سلام مریض ظہیر

داخلہ شناخت: ظہیر

ڈاکٹر سے حاصل کردہ اوقات

ڈاکٹر [illegible]	[illegible] 19-[illegible]-2003	AM 09:01:00

گزشتہ پیغامات و معائنہ جات

ڈاکٹر [illegible]	[illegible] 01-[illegible]-2003	AM 06:31:00
ڈاکٹر [illegible]	[illegible] 26-[illegible]-2002	AM 06:31:00

Figure 7.3 Patient Login Page

7.1.3 Code Plate and Keyboard Standards

According to the third objective, available and emerging code plate and keyboard standards were to be supported by the developed Urdu technology. This objective was met by keeping the keyboard and code plate separate from the core algorithms. Users are allowed to change the keyboard layout at any point and code plate can be dynamically altered by the web programmer. This was done by keeping the code plate and keyboard data in separate files that were included in the main stream program dynamically. It can be observed in top lines of Figures 7.1 to 7.3 that the user is given a choice to select from two available standard keyboards.

7.1.4 Establishing Telemedicine Sessions

This objective targeted for improving feasibility of telemedicine through internet for medical consultancy. The research has proposed a structured approach for Doctor-Patient dialogue in order to facilitate decision-making by the doctor about the disease of patients. As a first step the doctor and patient have to sign up with the system by providing necessary information about them. Figure 7.1 has already shown the signup page for doctors and figure 7.2 shows the same for patients. As the second step the patient has to search the database for a doctor closely related to his / her problem. The page developed for this purpose is shown in figure 7.4.

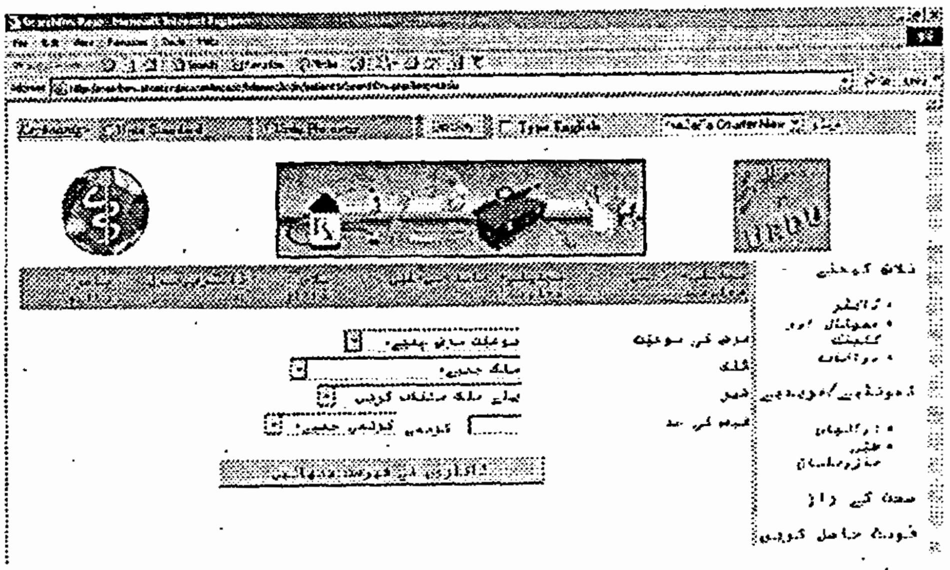
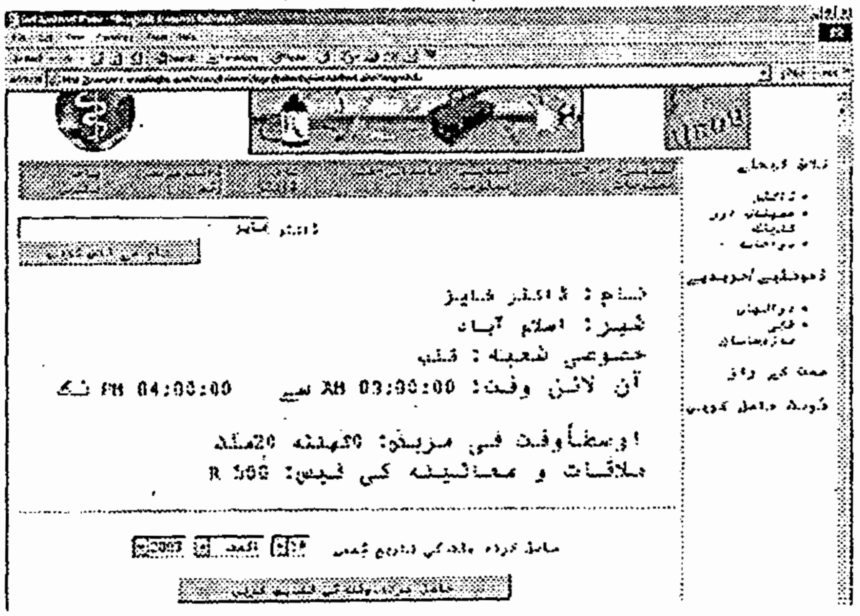


Figure 7.4 Search Doctor Page

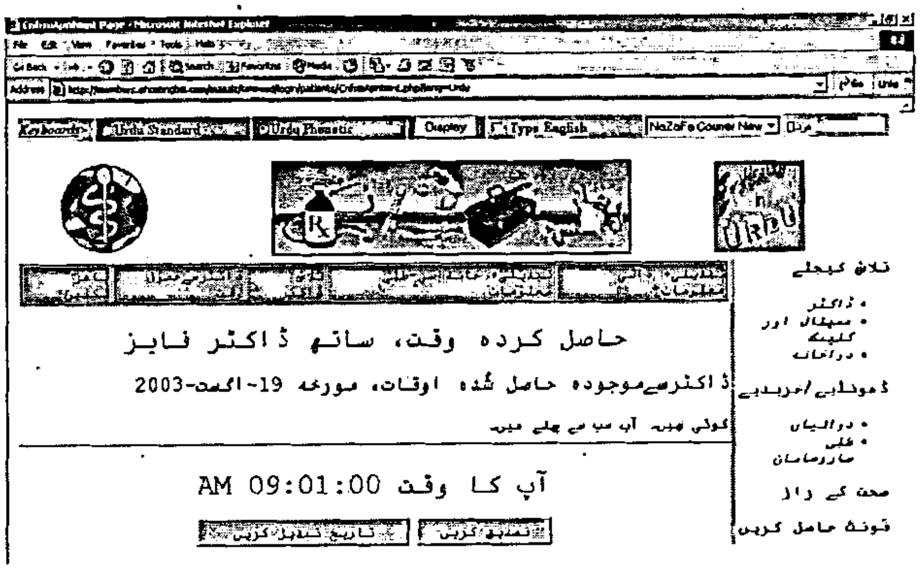
After finding a related doctor the patient is presented with a summary of information about the doctor and the times for which the doctor is available for online consultancy. This page is shown in figure 7.5. Now the patient can get an online appointment from the doctor. The system automatically assigns a time by using the start time of doctor's online avaialability (given by the doctor at time of signup) and adding the times given to patients who have already taken appointments on the same day. The system response to the patient after these calculations is shown in figure 7.6.



نام : ڈاکٹر فائز
شہر : اسلام آباد
خصوصی شعبہ : قلب
آن لائن وقت: 03:00:00 PM سے 04:00:00 PM تک

اوسطاً وقت فی مریض: 20منٹ
ملاقات و معائینہ کی فیس: 500 R

Figure 7.5 Doctor Information for Patient



حاصل کردہ وقت، ساتھ ڈاکٹر فایز

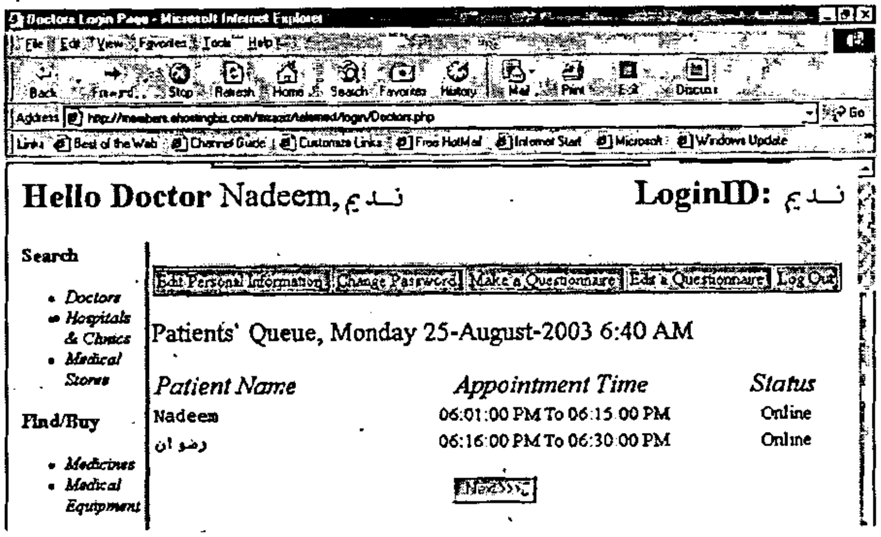
ڈاکٹرسےموجودہ حاصل شدہ اوقات، مورخہ 19-اگست-2003

کوئی نہیں آپ سب سے پہلے ہیں

آپ کا وقت AM 09:01:00

Figure 7.6 Information of Appointment for Patient

Name of the patient appears in the queue for the related doctor after the patient takes appointment from the doctor using the automatic time allocation system. Similarly this appointment also appears on the home page of patient. This information remains on display for information purposes but the actual tele-medical session is established when the doctor clicks on name of a patient from the queue for that day. This can only be done in the time allocated for the appointment. The informative page of patient queue for a doctor is shown in figure 7.7(a). It can be seen that name of the patient who opted for English language appears in English while the other in Urdu. Figure 7.7 (b) shows the situation when the doctor clicks on name of a patient to attend him / her. It is noticeable that basic information about the patient is shown to the doctor with the age calculated using the birth date as given by the patient at time of signup. The appointment information according to patient's perspective is displayed in home page of patient as shown in figure 7.3.

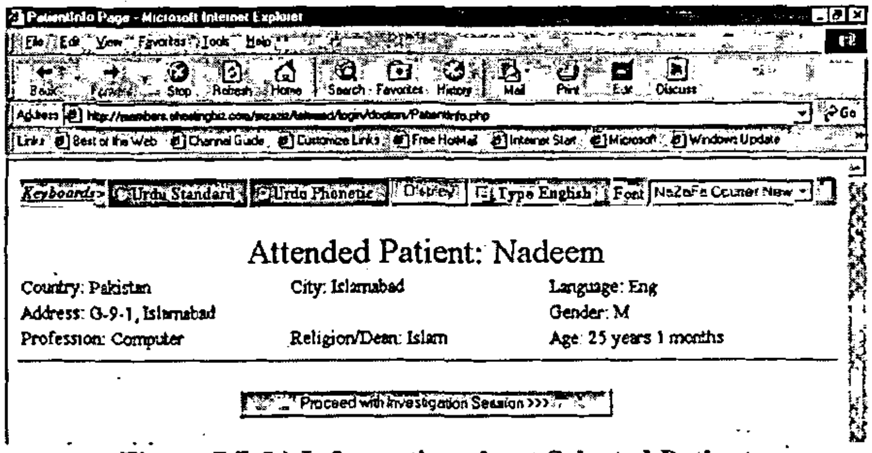


Hello Doctor Nadeem, ندیم **LoginID:** ندیم

Patients' Queue, Monday 25-August-2003 6:40 AM

Patient Name	*Appointment Time*	*Status*
Nadeem	06:01:00 PM To 06:15:00 PM	Online
رضوان	06:16:00 PM To 06:30:00 PM	Online

Figure 7.7 (a) Patient Queue on Doctor Home Page


PatientInfo Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://members.ehostingbiz.com/.../login/doctors/PatientInfo.php
Links Best of the Web Channel Guide Customize Links Free HotMail Internet Start Microsoft Windows Update
Keyboards> Urdu Standard Urdu Phonetic Disney Type English Font NaZaFa Courier New

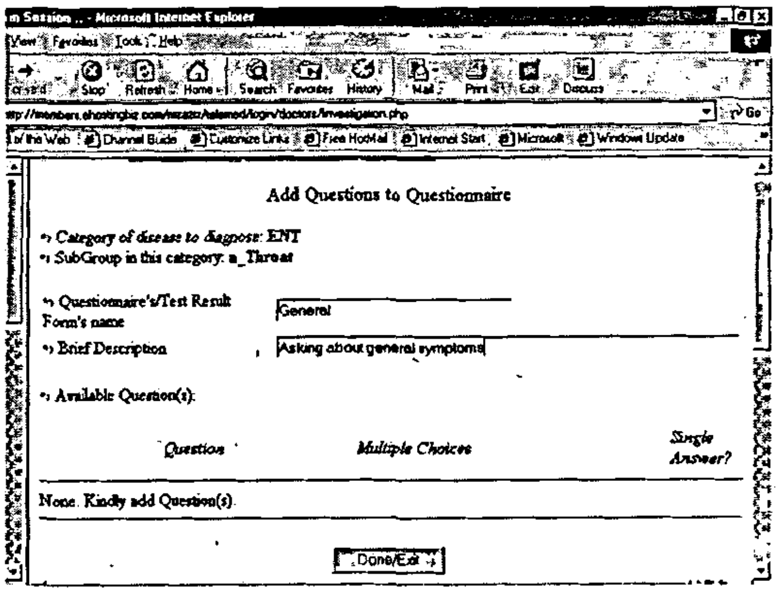
Attended Patient: Nadeem

Country: Pakistan City: Islamabad Language: Eng
Address: G-9-1, Islamabad Gender: M
Profession: Computer Religion/Dean: Islam Age: 25 years 1 months

Proceed with Investigation Session >>>

Figure 7.7 (b) Information about Selected Patient

The tele-medical consultancy session becomes active when the patient is in "wait in queue" status and the doctors selects the patient for attending him / her by clicking on patient's name. This structured medical dialogue starts with the level – 1 questionnaire as discussed in section 3.1.2. This would be a set of medical diagnosis questions aiming to inquire general condition of the patient. Usually this questionnaire will be same for every patient. The second level question set will be selected on basis of answers given for the first level. Similarly based upon the answers of second level questions the doctor will send a third level specialized question set aiming to pin-point the nature of disease. For all this setup a questionnaire design module is provided for the doctors. This facility allows doctors to prepare question sets for diagnosing diseases falling into their field of specialization. Figure 7.8 shows a snapshot of this mechanism. Each question has to be given in both English and Urdu in order to accommodate all types of patients. Patients who select English during their signup will see the questions in English and others in Urdu.


...Session ... - Microsoft Internet Explorer
View Favorites Tools Help
Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
http://members.ehostingbiz.com/.../login/doctors/investigation.php
Best of the Web Channel Guide Customize Links Free HotMail Internet Start Microsoft Windows Update

Add Questions to Questionnaire

Category of disease to diagnose: ENT
SubGroup in this category: a_Throat

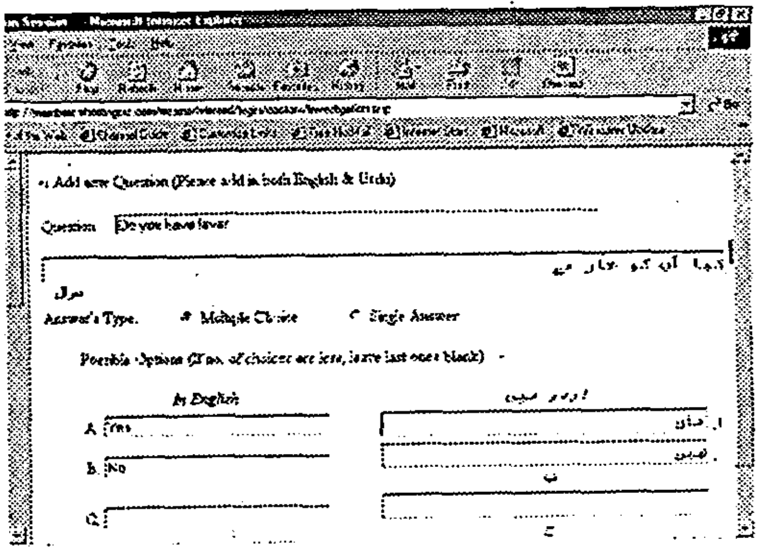
Questionnaire's/Test Result Form's name: General
Brief Description: Asking about general symptoms

Available Question(s):

Question	*Multiple Choices*	*Single Answer?*
None. Kindly add Question(s).		

Done/Exit

Figure 7.8 (a) Mechanism for Adding Medical Questionnaire



Add new Question (Please add in both English & Urdu)

Question: Do you have fever

کیا آپ کو بخار ہے

سوال

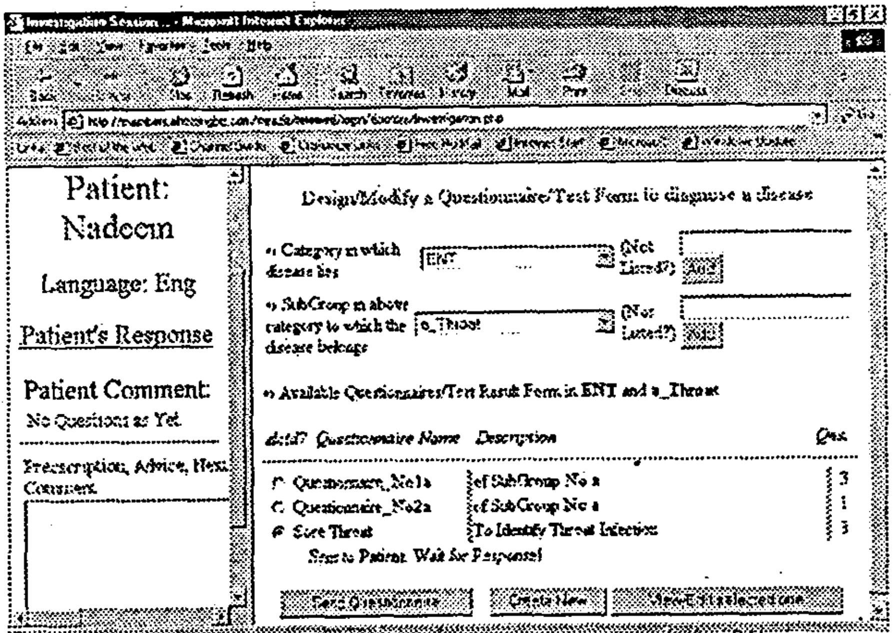
Answer's Type. ● Multiple Choice ○ Single Answer

Possible Options (If no. of choices are less, leave last ones blank)

In English	اردو میں
A. Yes	ا. ہاں
B. No	ب. نہیں
C.	ج

Figure 7.8 (b) Mechanism for Adding Questions to a Questionnaire

A database of the questionnaires is maintained and standard question sets are available for every doctor registered to the system. While attending a patient the doctor can select a required questionnaire and send to his / her patient. Figure 7.9 shows the mechanism available at doctor's end to select a diagnosis questionnaire and send it to the patient being attended. The snapshot is for the situation when the doctor is attending a patient named "Nadeem" as shown in left panel and questionnaire to confirm sour throat is sent to the patient, which belongs to ENT category and throat diseases group.



Patient: Nadeem

Language: Eng

Patient's Response

Patient Comment:
No Questions as Yet.

Prescription, Advice, Next Comment.

Design/Modify a Questionnaire/Test Form to diagnose a disease

Category in which disease lies: ENT (Not Listed?) Add

SubGroup in above category to which the disease belongs: a_Throat (Not Listed?) Add

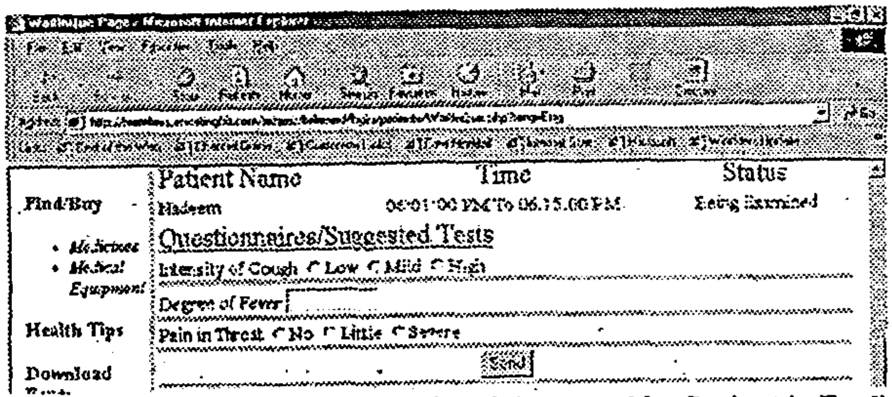
Available Questionnaires/Test Result Form in ENT and a_Throat

Add?	Questionnaire Name	Description	Qns.
○	Questionnaire_No1a	of SubGroup No a	3
○	Questionnaire_No2a	of SubGroup No a	1
●	Sore Throat	To Identify Throat Infection	3

Sent to Patient. Wait for Response!

Figure 7.9 Questionnaire Selection and Sending Mechanism

The interface is in English at the doctor's end as all doctors are expected to be fluent in the language. At patient's end the interface is according to choice of language selected by the patient time of signup. The questionnaire sent by the doctor is displayed in English for patients who opted for this language while the same questionnaire will be shown in Urdu for patients requiring it. Figure 7.10 shows the interface for a patient in English while figure 7.11 shows the same interface for a Urdu speaking user. The patient is expected to fill in the questionnaire with appropriate answers and send it back to the doctor.


Find/Buy
- *Medicines*
- *Medical Equipment*

Health Tips

Download

Patient Name	Time	Status
Nadeem	06:01:00 PM To 06:15:00 PM	Being Examined

Questionnaires/Suggested Tests

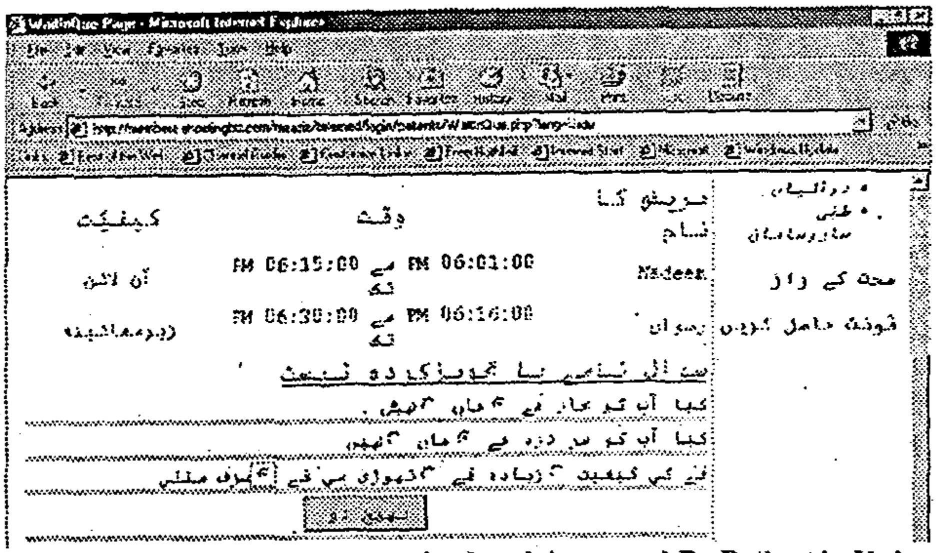
Intensity of Cough ◯ Low ◯ Mild ◯ High

Degree of Fever

Pain in Throat ◯ No ◯ Little ◯ Severe

Send

Figure 7.10 Questionnaire Received and Answered By Patient in English


- ادویات
- طبی ساز و سامان

صحت کے راز

فونٹ حاصل کریں

مریض کا نام	وقت	کیفیت
Nadeem	PM 06:01:00 سے PM 06:15:00 تک	آن لائن
رضوان	PM 06:16:00 سے PM 06:30:00 تک	(زیر معائنہ)

سوالنامہ یا تجویز کردہ ٹیسٹ

کیا آپ کو بخار ہے ◯ ہاں ◯ نہیں

کیا آپ کو سر درد ہے ◯ ہاں ◯ نہیں

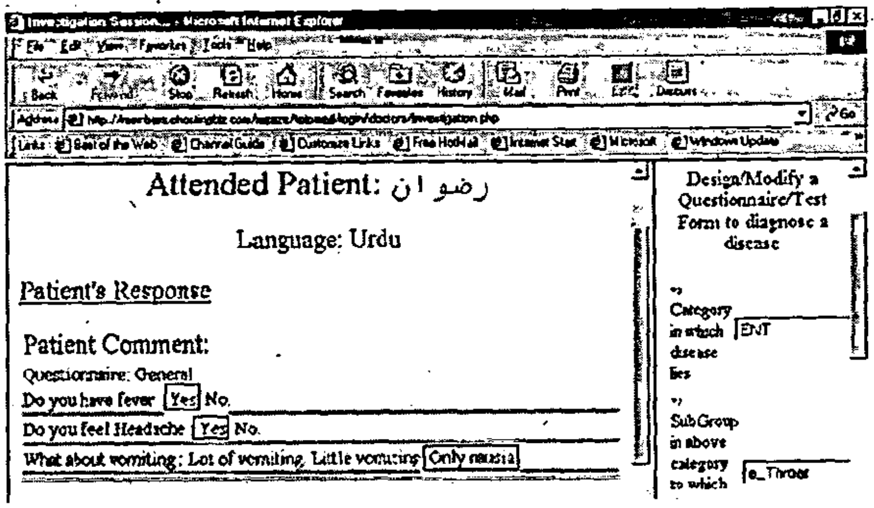
قے کی کیفیت ◯ زیادہ ہے ◯ تھوڑی سی ہے ◯ صرف متلی

بھیج دو

Figure 7.11 *Questionnaire* Received and Answered By Patient in Urdu

At doctor's end the language in which the questions were answered does not matter. The answers are shown to the doctor back in English. Figure 7.12 shows the interface when a questionnaire answered by the patient in figure 7.11 is returned to the doctor. It can be observed that the answers in the English version are same as given by the patient in Urdu.

This process of questions and answers is expected to completed in two or three cycles in most of the cases. Longer sessions may be required for complicated cases. Finally the doctor will diagnose the disease and send a prescription to the patient to conclude the consultancy with this patient. Figure 7.13 shows a sample prescription sent by the doctor in English while Figure 7.14 shows prescription for another patient which was sent in Urdu. After this the doctor may move to the next patient in queue.


Investigation Session... - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address Go
Links Best of the Web Channel Guide Customize Links Free HotMail Internet Start Microsoft Windows Update

Attended Patient: رضوان

Language: Urdu

Patient's Response

Patient Comment:

Questionnaire: General

Do you have fever [✓ Yes] No.

Do you feel Headache [✓ Yes] No.

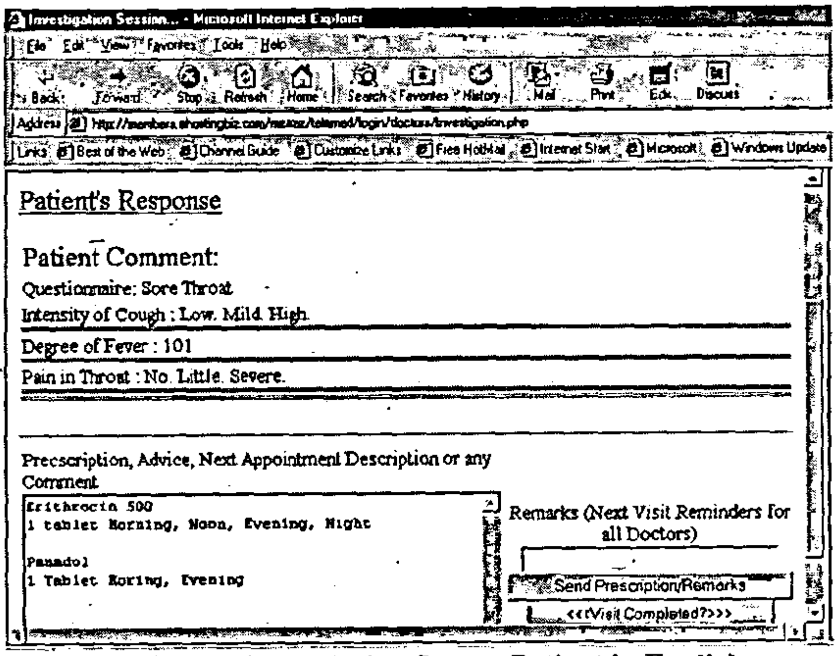
What about vomiting: Lot of vomiting, Little vomiting [✓ Only nausia]

Design/Modify a Questionnaire/Test Form to diagnose a disease

Category in which disease lies: ENT

SubGroup in above category to which: e_Throat

Figure 7.12 Questionnaire Returned to Doctor in English


Investigation Session... - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://members.hostingbiz.com/.../telemed/login/doctors/investigation.php
Links Best of the Web Channel Guide Customize Links Free HotMail Internet Start Microsoft Windows Update

Patient's Response

Patient Comment:

Questionnaire: Sore Throat

Intensity of Cough : Low. Mild. High.

Degree of Fever : 101

Pain in Throat : No. Little. Severe.

Precscription, Advice, Next Appointment Description or any Comment

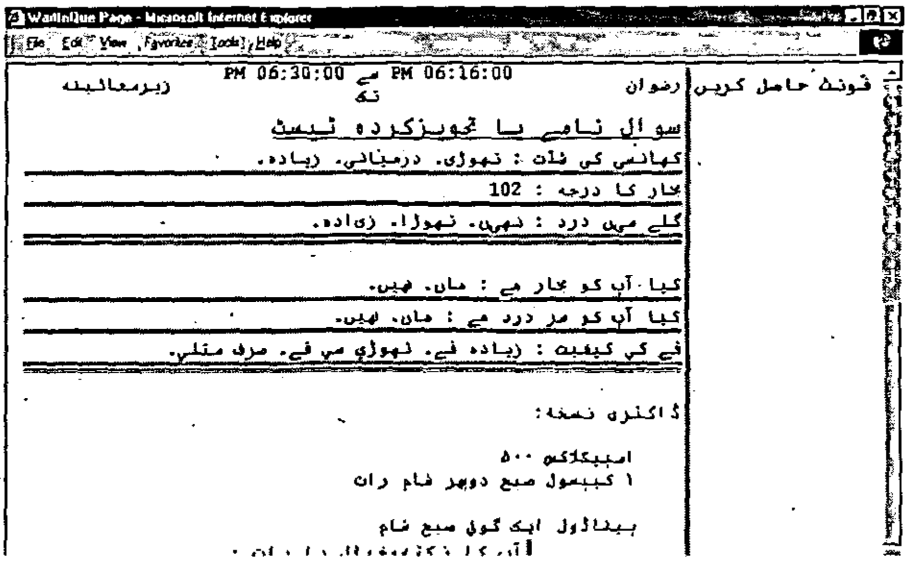
```
Erithrocin 500
1 tablet Morning, Noon, Evening, Night

Panadol
1 Tablet Moring, Evening
```

Remarks (Next Visit Reminders for all Doctors)

Send Prescription/Remarks
<<<Visit Completed?>>>

Figure 7.13 Prescription Sent to Patient in English


WadinIque Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

زیرمعائینہ PM 06:30:00 سے PM 06:16:00 تک رضوان | فونٹ حاصل کریں

سوال نامے یا تجویزکردہ ٹیسٹ

کھانسی کی شات : تھوڑی۔ درمیانی۔ زیادہ۔

بخار کا درجہ : 102

گلے میں درد : نہیں۔ تھوڑا۔ زیادہ۔

کیا آپ کو بخار ہے : ہاں۔ نہیں۔

کیا آپ کو سر درد ہے : ہاں۔ نہیں۔

قے کی کیفیت : زیادہ قے۔ تھوڑی سی قے۔ صرف متلی۔

ڈاکٹری نسخہ:

امپیکلاکس ۵۰۰
۱ کیپسول صبح دوپہر شام رات

پیناڈول ایک گولی صبح شام

Figure 7.14 Prescription at Patient's End in Urdu

7.2 Conclusion

As per discussion in section 7.1 it is clear that all objectives initially set for this research are properly satisfied. Urdu controls are properly functioning just like their English counterparts. Data storage for Urdu data and its transmission traffic was minimized by applying the conversion algorithms between client side and transmission media. Also a basic level telemedicine consultancy system was developed that can prove to be a milestone for further research / development.

It has been proved that Urdu can exist on dynamic web application as any other language of the world. The success of Urdu input controls and data conversion algorithms have opened door for web application developers to produce fully automated Urdu applications on the internet. This achievement is along with the advantage that no extra storage space or complex controls will be needed both at client and server ends.

As far as telemedicine is concerned, it is yet in infancy around the world. The methodology of establishing remote medical consultancy session can be considered as a ground work. The part in which the language barrier was removed between doctor and patient requires special attention. In this research the facility was limited to showing the questionnaire in English at doctor's end while in Urdu at patient's end without going into translation mechanisms. Any how, the research has presented an approach that can be useful for further research in the area.

7.3 Future Directions

In this research the concentration was focused on the widely used operating system and browser platform of Microsoft. Although the algorithms were implemented in Java Script, which is largely platform independent but there is still room for modification and customization according to different browser platforms. This is because each browser hosts different privileges and features at programming level. A suitable direction in this context could be standard input controls for UNIX based web browsers.

An interesting enhancement in the current research could be a full fledge language translation between doctor and patient. It is expected that people from one country may be getting medical services from doctors of some other countries. The discussion has to be translated for one another so that a useful session is achieved.

Different electronic equipment for reading human symptoms are being developed and expected to be developed. Research work will be needed to make efficient use of such devices connected to patient's computer for more effective tele-medical consultancy.

The method proposed in this research to represent Urdu data as single-byte characters can be enhanced and applied in to develop Urdu Domain Name Server. This will allow users to keep their internet and email names in Urdu as well.

A. ASCII to Unicode Map (NaZaFa Urdu Font)

Ascii		urduStd		urduPhntc		urduNLA	32		40
!	33		0		0	null	46	.	41
"	34	ـ	1600	″	1524		1524	″	42
#	35		null		0		1548	،	43
$	36	ئ	1574		null		1563	؛	44
%	37	ي	1610		null		1567	؟	45
&	38		null	O	1750		1569	ء	46
'	39	.	46	O	1612		1570	آ	47
(	40		0		0		1571	أ	48
)	41		0		0		1572	ؤ	49
•	42	O	1612		0		1573	إ	50
+	43	آ	1570		0		1574	ئ	51
,	44	،	1548	،	1548		1575	ا	52
-	45	أ	1571		0		1576	ب	53
.	46	۔	1748		0		1577	ة	54
/	47	O	1618		0		1578	ت	55
0	48	۰	1776	۰	1776		1579	ث	56
1	49	۱	1777	۱	1777		1580	ج	57
2	50	۲	1778	۲	1778		1581	ح	58
3	51	۳	1779	۳	1779		1582	خ	59
4	52	۴	1780	۴	1780		1583	د	60
5	53	۵	1781	۵	1781		1584	ذ	61
6	54	۶	1782	۶	1782		1585	ر	62
7	55	۷	1783	۷	1783		1586	ز	63
8	56	۸	1784	۸	1784		1587	س	64
9	57	۹	1785	۹	1785		1588	ش	65

:	58		0		0	1589	ص	66
;	59	؛	1563	؛	1563	1590	ض	67
<	60	O	1616	O	1616	1591	ط	68
=	61	ؤ	1572		0	1592	ظ	69
>	62	O	1614	O	1614	1593	ع	70
?	63	؟	1567	؟	1567	1594	غ	71
@	64	،	1643		null	1600	ـ	72
A	65	O	1764	آ	1570	1601	ف	73
B	66		null	O	1613	1602	ق	74
C	67	ث	1579	ث	1579	1603	ک	75
D	68	ڈ	1672	ڈ	1672	1604	ل	76
E	69		null	*	1645	1605	م	77
F	70		0	ژ	1688	1606	ن	78
G	71	غ	1594	غ	1594	1608	و	79
H	72	ح	1581	ح	1581	1609	ى	80
I	73	O	1648	ى	1609	1610	ي	81
J	74	ض	1590	●	9679	1611	O	82
K	75	خ	1582	خ	1582	1612	O	83
L	76		null	لا	65275	1613	O	84
M	77	إ	1573	٪	1642	1614	O	85
N	78	ں	1722	ن	1606	1615	O	86
O	79	ة	1577	ة	1577	1616	O	87
P	80	O	1615		0	1617	O	88
Q	81	O	1615	O	1611	1618	O	89
R	82	ڑ	1681	ڑ	1681	1642	٪	90
S	83	ص	1589	ش	1588	1643	،	91
T	84	ٹ	1657	ٹ	1657	1643.5		92
U	85	‘	8216		null	1645	*	93

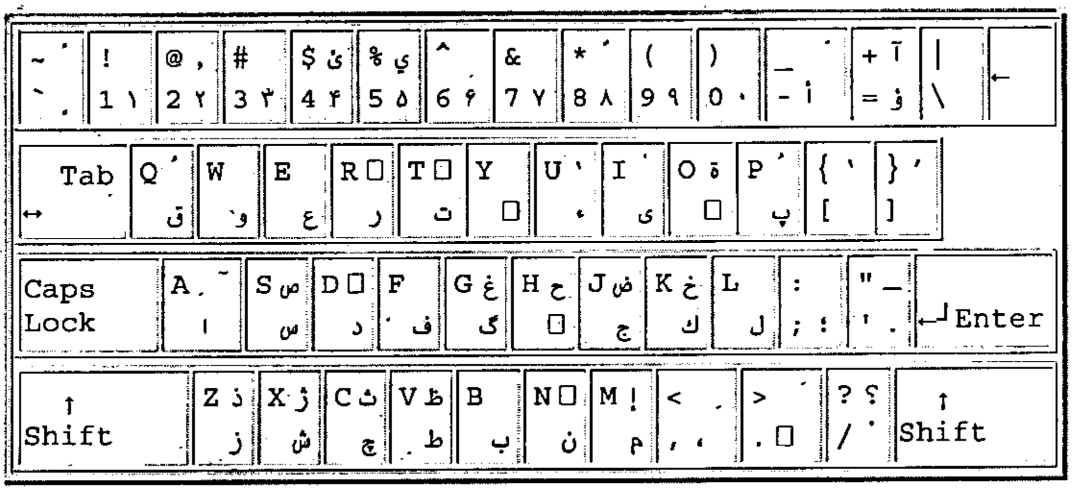
V	86	ظ	1592	ظ	1592
W	87		null	ؤ	1572
X	88	ژ	1688	ض	1590
Y	89	O	1619	ۓ	1747
Z	90	ذ	1584	ذ	1584
[	91		0	O	1617
\	92		null	O	1615
]	93		0		null
^	94	O	1750		null
_	95	O	1617	ـ	1600
`	96	O	1613		0
a	97	ا	1575	ا	1575
b	98	ب	1576	ب	1576
c	99	چ	1670	چ	1670
d	100	د	1583	د	1583
e	101	ع	1593	ع	1593
f	102	ف	1601	ف	1601
g	103	گ	1711	گ	1711
h	104	ھ	1726	ہ	1729
i	105	ى	1609	ي	1610
j	106	ج	1580	ج	1580
k	107	ك	1603	ك	1603
l	108	ل	1604	ل	1604
m	109	م	1605	م	1605
n	110	ن	1606	ں	1722
o	111	ہ	1729	ھ	1726
p	112	پ	1662	پ	1662
q	113	ق	1602	ق	1602

1648	O	94
1657	ٹ	95
1662	پ	96
1670	چ	97
1672	ڈ	98
1681	ڑ	99
1688	ژ	100
1711	گ	101
1722	ں	102
1726	ھ	103
1729	ہ	104
1746	ے	105
1747	ۓ	106
1748	۔	107
1764	O	108
1776	۰	109
1777	۱	110
1778	۲	111
1779	۳	112
1780	۴	113
1781	۵	114
1782	۶	115
1783	۷	116
1784	۸	117
1785	۹	118
8216	‘	119
8217	’	120
8219	‛	121

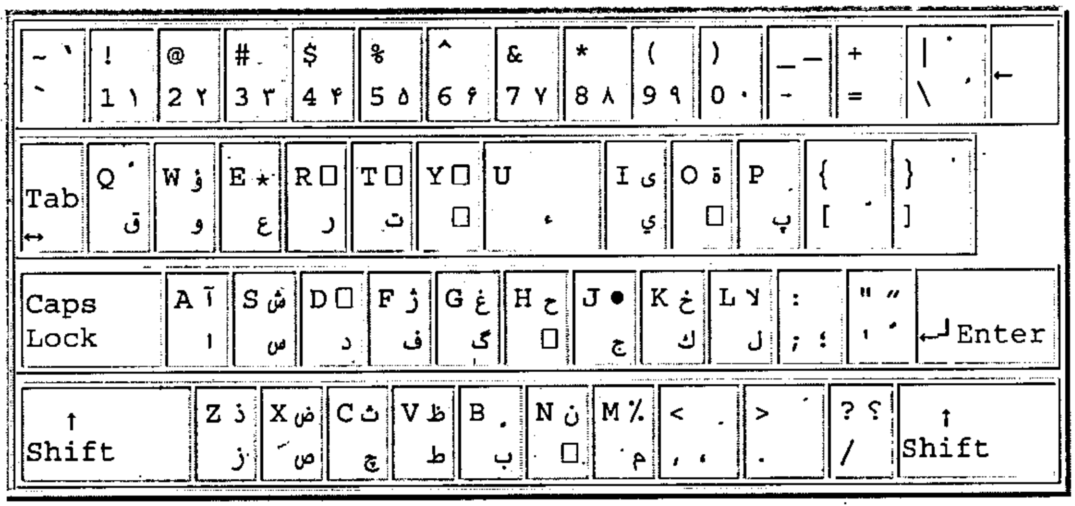
r	114	ر	1585	ر	1585		9679	●	122
s	115	س	1587	س	1587		65275	ﻻ	123
t	116	ت	1578	ت	1578				
u	117	ء	1569	ء	1569				
v	118	ط	1591	ط	1591				
w	119	و	1608	و	1608				
x	120	ش	1588	ص	1589				
y	121	ے	1746	ے	1746				
z	122	ز	1586	ز	1586				
{	123	‘	8216		null				
|	124		null	◌ْ	1618				
}	125	’	8217	◌ٰ	1648				
~	126	◌ً	1611	‛	8219				

B. Urdu Keyboard Layouts

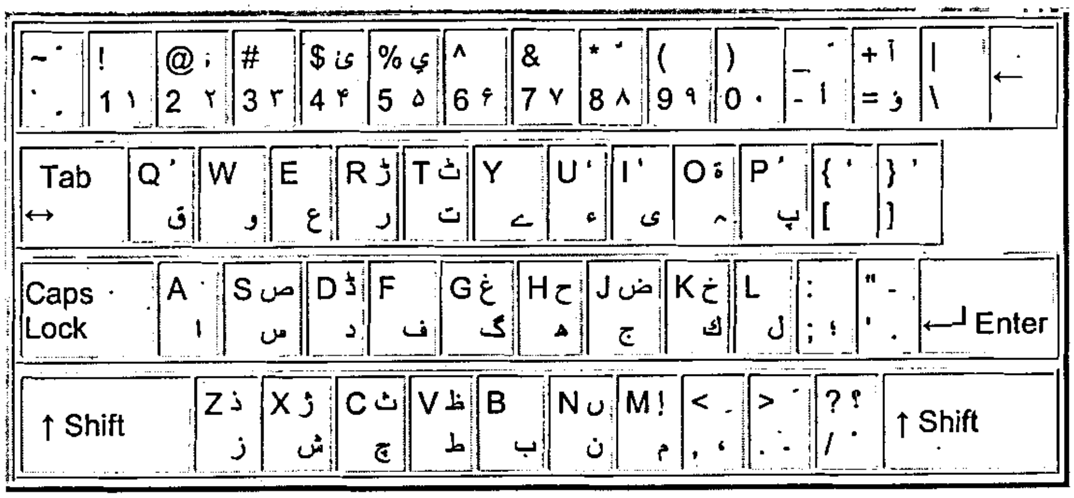
B.1 Standard Keyboard (NaZaFa Courier New)



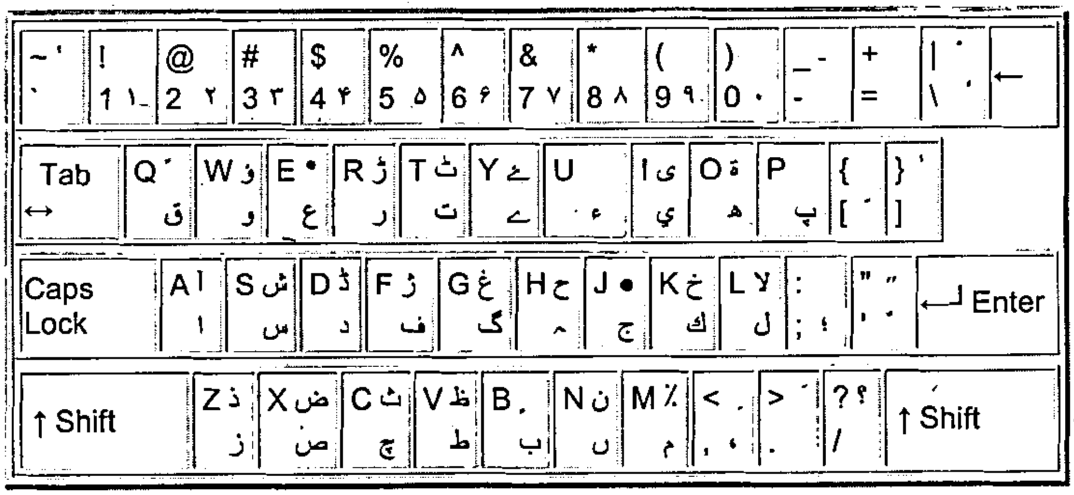
B.2 Phonetic Keyboard (NaZaFa Courier New)



B.3 Standard Keyboard (NaZaFa Arial)



B.4 Phonetic Keyboard (NaZaFa Arial)



31. http://www.tu-darmstadt.de/index.html
32. http://www.edf.
33. Human Development Foundation "Pakistan in Search of its Essence->Literacy and Education->Table 5", www.yespakistan.com/hdf/whywedoit/hdinpak.asp
34. paknews.com "Literacy Rate Reaches To 51.6 Percent This Year", paknews.com/flash.php?id=24&date1=2003-06-06