Centralized Medical Practice Management Software System

Supervisor: Sadia Hamid Kazi

Khaled Mohammad Ali	**11201006**
Mohammad Ashikus Salehin	**14141014**
Tahmid Hassan	**14241009**
Omar Hassan Khan	**11301012**

Department of Computer Science and Engineering

School of Engineering and Computer Science

BRAC University

Submitted on: August 2015

DECLARATION

We hereby declare that this work was carried out by us under the guidance and Supervision of **Sadia Hamid Kazi,** Assistant Professor, Department of Computer Science and Engineering, BRAC University. The period of project work is from September 2014 to August 2015. This project work is submitted to the Department of Computer Science and Engineering, School of Engineering and Computer Science, BRAC University in partial fulfillment for the requirement of Degree in Bachelor of Science in Computer Science. We declare that this work has not been submitted anywhere else for the award of any other degree.

Signature of Supervisor

Mrs. Sadia Hamid Kazi

Assistant Professor, Department of Computer Science and Engineering, BRAC University

Signature of the Authors

Khaled Mohammad Ali

Mohammad Ashikus Salehin

Tahmid Hassan

Omar Hassan Khan

ABSTRACT

The purpose of this project was to construct a centralized patient management system that would serve as a consolidated platform to provide an effective and efficient mechanism for patient management and means for availing various healthcare and health related services. The entire system is designed to streamline the patient management system by primarily targeting small and medium sized hospitals, diagnostic centres as well as individual practices of physicians by centralizing the process maintaining schedules, queues and appointments. Direct call ambulance service, locating nearest ER facilities, blood bank and pharmacy with respect to user location have also been encompassed to provide a holistic approach. Currently operations of these procedures are of an erratic where due to delays and other factors of inefficiency management services can neither be properly delivered nor availed. Using this system the patients can locate services which are best suited for their need, based on different criteria including specialty, location, queue and other aspects which are deemed to be of convenience. The service providers; such as doctors or their assistants, diagnostic centre staff, etc. can manage appointment schedule using the token system; patients can be made aware in real time of their current position in queue and be informed of delays or situations of cancellation.

ACKNOWLEDGEMENT

We would like to start by expressing our gratitude to our supervisor Ms. Sadia Hamid Kazi, Assistant Professor, Department of Computer Science and Engineering, School of Engineering and Computer Science, BRAC University for encouraging and supporting us through the development of this project.

TABLE OF CONTENTS

Section	Title	Page No.
Chapter 1	Introduction	1
1.1	Motivation	1
1.2	Primary Objectives	1
1.3	Key Features and Impact	1
1.4	Thesis Structure	2
Chapter 2	Background Study	4
2.1	The Current System and Problems Faced	4
2.2	New Technologies	5
2.3	Existing Platforms	5
2.4	Literature Review	5
Chapter 3	Requirements Analysis	9
3.1	Fact Finding	9
3.1.1	Interviews	9
3.1.2	Research	9
3.1.3	Observation	9
3.2	Problems faced by current system	10
3.3	Functional Requirements	11
3.4	Non-Functional Requirements	12
Chapter 4	System Design	23
4.1	Activity Diagram	14
4.2	Use Case Diagram	18
4.3	Data Dictionary	20
4.4	Mobile Based Application	26

4.4.1	**System Architecture**	**27**
Chapter 5	**System Implementation**	**28**
5.1	**ER Diagram/Schema**	**29**
5.2	**Programming Languages Used**	**30**
5.3	**Programming**	**30**
5.3.1	**Algorithms and Optimization**	**31**
5.4	**Tools Used**	**33**
5.5	**System Infrastructure**	**34**
5.5.1	**Hardware Specifications**	**34**
5.5.2	**Web Application Requirements**	**36**
5.5.3	**System Infrastructure Diagrams**	**37**
Chapter 6	**Business Model**	**38**
6.1	**Model Details**	**38**
Chapter 7	**Conclusion**	**40**
7.1	**Limitations**	**40**
7.2	**Future Work**	**40**
7.3	**Personal Reflection**	**41**
	References	**42**
	Appendices	**44**

LIST OF FIGURES

Code	Title	Page No.
Figure 4.1	Making an appointment	14
Figure 4.2	Appointment queue maintenance by doctor	15
Figure 4.3	Signaling ambulance service operator in case of emergency	16
Figure 4.4	Responding to an emergency call	17
Figure 4.5	Appointment Management	18
Figure 4.6	Billing Management	19
Figure 4.7	System architecture of EMS	27
Figure 5.1	Database Schema Design	29
Figure 5.2	Improvements in amount of space required	31
Figure 5.3	Improvements in parse time	32
Figure 5.4	Service Side Infrastructure	37
Figure 5.5	Client Side Infrastructure	37
Figure 6.1	Break-even point of CBA	39
Figure 7.1	Future Work	41

LIST OF TABLES

Code	Title	Page No.
Table 4.1	Appointment Details	20
Table 4.2	Doctor Appointments	20
Table 4.3	Doctor Qualifications	20
Table 4.4	Doctor Services	21
Table 4.5	Doctor Workplan	21
Table 4.6	ER Information	21
Table 4.7	ER Operator	22
Table 4.8	Basic Patient Information	22
Table 4.9	Patient History	23
Table 4.10	User Details	23
Table 4.11	User Login	23
Table 4.12	Profile Pictures	24
Table 4.13	Medicine Table	24
Table 4.14	Diagnostic Test Table	24
Table 4.15	Prescription Details Table	24
Table 4.16	Prescription Medicine Table	25
Table 4.17	Prescription Test Table	25
Table 4.18	Bill Information Table	25
Table 4.19	Bill Details Table	26
Table 5.1	Service Side Requirements	34
Table 5.2	Client Side Requirements	35
Table 5.3	Mobile Based Application Requirements	35
Table 5.4	Web Application Requirements	36
Table 6.1	Cost Benefit Analysis	39
Table A.1	Change in amount of space required with increase in number of users	55
Table A.2	Change in parse time with increase in number of users	55

Chapter 1: Introduction

Quality and easily obtainable access to healthcare services is considered to be a fundamental right in any modern society. The need to conveniently avail healthcare services cannot be underestimated by any means. Through absorbing modern digital elements of modular structure into the essential healthcare management system, revolutionary progress can be made in this sector. Automation of the system will reduce the hassles otherwise associated with using services, more efficient management and allocation of resources will be possible. Infrastructure and knowledge such as desktop/laptop computers, cell phones, and the internet that are already existing and are in use by the stakeholders will be used in this regard, thus requiring limited technological and skill/know how upgrades on their part.

1.1 Motivation

Provision of healthcare services, particularly patient management is a largely neglected area in Bangladesh. Any meaningful improvement made in this field will therefore be beneficial for all stakeholders involved. General people attempting to receive services will be able to experience the advantages provided by a system that is efficient, effective, easy to use and affordable. In order to minimize suffering of the affected people and introduce efficiency in the system that will provide greater utility, work on this project was commenced.

1.2 Primary Objectives

The primary objectives of the system are:

- Modernize and streamline and patient management system
- Provide a consolidated platform for improving ease of access to healthcare and related services

1.3 Key Features and Impact

The primary features of this system are as follows:

- Front end web (and mobile based) interface that is user-friendly and has been logically designed.

- A specifically designed database, with the purpose of storing doctor and patient details, availability information of pharmacy and blood bank, diagnostic center services, doctor appointment schedule, diagnostic center services schedule, ER services alongside access levels for the intended specific user types of the system as the main aspects.
- Create and store reports, prescription, diagnosis and medication for further use.
- Management of system users, i.e. administrative capabilities.
- Closest facility detection for availing healthcare and/or other services.
- Mechanism for creation, updating and searching of medical records with ease.
- Maintaining a patient queue for individual physician's chamber (or other facility) for the aim of attaining efficiency in the appointment management process.
- Necessary security later to allow access to data based only on specific user status within the system.
- The capability to add, view and update (as required) doctor appointment schedules.
- Online and offline registration system, with encryption of user passwords to ensure security and privacy.

The target of the project is to computerize if not all, then as much as possible all relevant and necessary information regarding patients, doctors, appointments, etc. The anticipated and hoped for impact of this project is the development and implementation of a system that exhibits the required stability and the ability to perform and deliver services of a computerized central health care management system to doctors and other personnel at small and medium clinics, hospitals as well as private practices. By updating from the current system which is paper-based backed by use of telephone communications, the new proposed system will introduce greater efficiency in the providing of services, resulting in improved health care given and streamlining service provider operations which shall be undoubtedly beneficial to them for a number of reasons.

1.4 Thesis Structure

The thesis has been broken down into a number of sections illustrating different aspects and stages of the project, each section is itself divided into a number of subsections for the purpose of convenience and conveying information in a manner that is structurally organized and easy to read.

Chapter 1 consists of an introduction to the project along with the motivation, primary objectives, key features and impact associated with it.

Chapter 2 shows how things are presently being carried out in the sector of providing healthcare services in Bangladesh, incorporated within this section is a background study of the mentioned sector, a basic overview of new technologies and existing platforms, and a literature review of related work looked at during the development of the project.

Chapter 3 illustrates the requirements analysis stage of the projects, including a fact finding subsection (reliant on interviews, research and observation), specific identification of problems faced by current system, list of functional and non-functional requirements of the proposed system.

Chapter 4 describes the system design of the project, it follows Chapter 3 (system requirements stage) to outline the planned structure and design of the project according to the requirements and some details of the system modules or subsystems in terms of activity and use case diagrams, data dictionary and an overview of the mobile based system.

Chapter 5 demonstrates the implementation phase of the working system, including operational details and specifics of the project, optimization activities and an overview of the system infrastructure.

Chapter 6 presents a small business model making a basic cost benefit analysis (CBA) on a preliminary level regarding deployment of the project.

Chapter 7 specifies the limitations of the project as per our findings, planned future work that may be possible from our perspective and a conclusion. That is followed by the list of references and appendices at the end.

Chapter 2: Background Study

Advances have been made in the healthcare industry at an increasing rate not only in the developed world, but also in the developing nations. Alongside improvements in the technology and techniques used for treatment in terms of variety, an expansion can also be witnessed in terms of the extent to which such services are provided as more and more people now have access to healthcare services. Bangladesh is no stranger to this phenomenon for a number of reasons which will not be discussed here since they are not relevant to the project. At the same time another industry has significantly grown in Bangladesh which is the information technology industry; there has been a widespread introduction of devices like personal computers, laptops and smart phones throughout the country, particularly in the urban centers like Dhaka. In terms of connectivity internet penetration rate in the country has increased as a variety of internet services in the form of broadband, WiMax and 3G are being offered at gradually decreasing prices.

2.1 The Current System and Problems Faced

In the current mode of healthcare services appointment management system, a paper-based system is used where the individual doctor's list of appointments is maintained by a staff member or secretary. Time slots are assigned by the secretary and when patients arrive they have to often wait for long hours at the queues, not only because they have been possibly given an incorrect time, but also because of the delays in the doctor starting to examine the patients at his chamber in a hospital or clinic or at his private practice. Similar problems are also present in the case of diagnostic center services; an additional problem here is the lack of efficient allocation to the services and resources that are available, there are excessively long queues and pressures at certain centers to avail services, causing congestion.

Furthermore in the case of blood bank, pharmacy and ER services; there is no central platform available which easily provides access to obtaining these services. In the case of requiring blood for purposes like medical operations, patients and their relatives have to frantically ask around for blood from donors or visit blood banks to get what they require on an urgent basis, causing hassles for them and wasting time. Also there is no way for patients to discover the nearest ER facility or ambulance service in case of any emergency. These are the predominant problems discovered in the existing system currently being used.

2.2 New Technologies

Current trends in the use of the internet are very diverse in terms of catering to the needs of the different kinds of users. From making information available and for communication purposes, the internet is being increasingly used for delivery of services. Mobile phones are becoming more sophisticated and powerful by the day, far surpassing conventional computers of not so long ago in terms of performance, specifications and capabilities. The point of mentioning all of this is to illustrate the direction towards which technology is advancing and its extent.

2.3 Existing Platforms

Internet penetration rate, mobile phone usage, smart phone usage, ownership of devices (phones, laptops, PCs)

According to data from the World Bank; the number of internet users (per 100 people) in Bangladesh increased from 3.7% in 2010 to 9.6% in 2014, which is an increase of 160%. The number of mobile subscriptions in Bangladesh was 121.86 million as of January 2015 (BTRC statistics); according to a GSMA report the number of unique mobile subscribers is 70% and the mobile penetration rate is nearly 100% for Dhaka city. In the urban areas like Dhaka the ownership of devices (phones, desktop computers, laptop computers, etc.) is increasing as cheaper devices are now increasingly more available in the form of locally produced and/or assembled devices and higher incomes (Bangladesh became a lower middle income country in 2015).

2.4 Literature Review

A systematic research is taken to perform relevant result of our purpose. We take two main steps to search and evaluate the results:

- Gathering sources: which contains a widespread search using the most accurate key words and moreover, a good literature review requires to keep apart key themes or issues related to your own research interests, so more restriction were added to clarify the articles.
- Evaluating Sources: which consists of reading all the eventual articles with some questions kept in wind: Does the information support the conclusion? Is the argument complete? Does all the research arrive at the same conclusion or are there differing?

Finally, eliminating inconvenient articles with these merits and classifying all the remained articles.

A Health Information System (HIS) is a useful entity within the structure of a broad health system to elevate the health of individuals and the population. Many definitions of a management information system have been schemed but the one proposed by Hurtubise (1984): “a system that provides specific information support to the decision-making process at each level of an organization” (Hurtubise 1984 cited Lippeveld 2001, page 15). The HIS structure ought to permit era of fundamental data for utilization in choice making at every level of the health framework with a given measure of assets. This includes the procedures for gathering, preparing and dispersing data in a health framework (Shrestha and Bodart 2000). [1]

Public health choice making is basically subject to exact, auspicious and dependable data. There is a far reaching conviction that a large portion of the national and sub-national health data frameworks fall flat in giving greatly required data backing to confirmation based health arranging and intercessions. Medical data framework disappointment is broadly rebuked for this circumstance. [2]

As time passes by, therapeutic consideration is getting more multifaceted and as new innovations are found, there is a requirement for the restorative group to think of better structures of keeping up the patients' data. Fitting and precise documentation comes close by close by with better restorative consideration and usage strategies. The electronic therapeutic record (EMR) is one of the restorative devices that try to enhance medicinal consideration by furnishing healing centers with the sort of stage that takes into account new administrations and new usefulness. The patient data can then be upgraded as the patient experiences new treatment and more up to date health information is found. [4]

Patients expect more than just analysis and medicine. They require data about their health status, investment during the time spent nurture, their security, straightforwardness in the treatment system, foresight of the result of the treatment and collaboration in every one of the phases of consideration. There is however the requirement for sober minded exertion by administration to strengthen coordination between the phases of consideration and the experts and patients cooperation over the span of consideration conveyance. An upright exertion ought to be made at streamlining and facilitating the medicine and medication organization

inside of the involving so as to heal center specialists, attendants, drug specialists and patients. [3]

A modernized therapeutic record carries with it numerous points of interest. It displays information in an extremely composed way so that every healing center office discovers the obliged data without troubles. This progression the way medicinal services is polished in that it is unrealistic to ignore essential discoveries. The computer based record is then upgraded as new patient information is benefited. This can maintain a strategic distance from miscommunication among doctor's facilities and human services staff. The data can be utilized as a part of future to evaluate the historical backdrop of sicknesses and how powerful the medicine and treatment embraced is. [4]

People are increasingly viewing their connection to the internet, including their gadgets, as their lifeline. Your patients keep their address book, calendar, and more on their mobile devices. They rely on them. Ensuring that your website is easily accessible and properly optimized for mobile devices just makes sense. The trend towards patients using their mobile devices to access information will only continue to grow. According to BTRC, 43.167 million people use internet via mobile devices. With such a trend, it makes sense to consider responsive web application and mobile app for your healthcare system.

Our Centralized Medical Practice Management Software System allows people to store large amount of information in different place. The E-health record may include the patient personal information, like telephone number age and so on. Sometimes the patients just want to share their relative information only to their physician. So our job is to secure all the information they share with us. We found that there is a set of national existing standards named Act prepared to provide legal recognition and security of Information and Communication Technology and rules of relevant subjects by Bangladesh government to offer administrative, technical and physical standards to ensure the security of information in the IT system. Moreover the limitations for physical equipment access to the facilities where health information system is housed, to ensure that the authorized personnel are allowed to access the system.

Kathi E. Hanna stated that the recognition of the vast benefits Electronic Health Record systems wil lprovide in the healthcare delivery enterprise continues to grow. Benefits are expected to include:

- Fewer medical errors;
- The arrangement of more efficient healthcare services;
- better ability to manage chronic disease;
- Advanced health status;
- Consolidated work processes; and
- More accurate and complete medical records.

Patrick Wong stated that e-healthcare environment would focus on patient-centric systems that reduced complexity, improve efficiency, and provide better patient outcomes. The transition towards e-healthcare, a vision of personalized healthcare that enclosed everything from patient empowerment to having a single slice-through view of the patient, would require a move away from an acute delivery platform to one that will focus more on managing the patient for life. He also add this was a move towards healthcare 'without walls', where a connected healthcare delivery platform would be accepted by greater use of patient-controlled data, the advancement of health data through analytics, and empowered patient and physician communities. The adoption of disruptive technologies would also move treatment and patient management beyond the confines of the traditional institution. The increase use of analytics layered over disparate data sources would help to transform the data mountain into actionable information.

During our project work we gave preference to JavaScript code optimization because according to ***Gregory Baker***, Client-side scripting can make an application dynamic and operative though the browser's interpretation of this code can itself introduce failure, and the work rate of different constructs differs from client to client.

Chapter 3: Requirements Analysis

This section consists of the requirements analysis of the project. The requirements analysis stage is to identify and understand the requirements that would be expected from and required of the project. Included within the requirements specification are the techniques or methods to identify user requirements, functional requirements and the non-functional requirements of the system.

3.1 Fact Finding

Interviews, research and observation

There are a number of methods or techniques that may be used by system analysts to conduct fact-finding in order to investigate and identify system requirements. It is important to understand the functioning of the system to design and develop the proposed system.

3.1.1 Interviews

Information about current systems may be obtained by analysts from speaking with users of the current system who are also in fact potential users of the proposed system. There are a number of reasons justifying this methodology; acquiring information, verification and clarification of information, identifying requirements, collecting opinions and ideas, getting end-user involvement, etc. In this regard as permitted by constraints, we were able to speak with several people involved in delivery of healthcare services as well as a number of service seekers to help us understand problems with the existing system.

3.1.2 Research

A standard and widely used method is to research available resources and material regarding the problem and application. Journal articles, publications and newspaper articles, documents and reports of existing similar and/or proposed systems are possible sources for this methodology of requirements analysis. Thus we have carried out our research accordingly as described in more detail in the relevant section.

3.1.3 Observation

An effective methodology for this stage is the observation technique; where it is possible to carry out investigation by either directly participating or closely examining the focus of

study. The technique may be generally used when other methodologies are not deemed to be satisfactory in terms of thoroughness and degree, or it may be used augmentation to them as well. In our case we observed for a relatively short duration how the existing system (of healthcare service delivery) is functioning and what the major limitations apparent as per observation.

3.2 Problems faced by current system

The current system through which operations are carried out is a manual paper-based system back by some use of telephones to make calls. Some problems associated with this system were mentioned earlier in previous section, they are listed categorically in this subsection for the convenience purposes.

Time consuming

A major complain of patients regarding service providing and management is the lengthy and almost invariably unnecessary waiting at queues and delays which may result from causes of incorrect time being relayed to the doctor starting late in examining patients. Making appointments and managing schedules also consumes some time. In a world where time is a precious resource, a modern computerized system can avert such problems.

Deterioration in quality of services

As mentioned earlier the records and appointment schedules are currently maintained manually on paper, this is particularly true for small and medium sized facilities; it is time consuming and rather cumbersome to operate like this. Appointments are made and confirmed via phone calls, with the records written on paper. As the system is manual, patients are required to make appointments by telephone or by showing up the healthcare providing facility and waiting in queues. Filling up patient details, checking time schedule is performed manually and it wastes time, causing efficiency losses.

Unavailability, Inconvenience and lack of flexibility

There is no central or consolidated platform that can provide patients with information about the matters of pharmacy, blood bank, ER and ambulance services at present. This can be considered a failure to easily provide access to consumable services. The existing system has its limitations in it that it cannot be used irrespective of location and timing, e.g. make appointments online as they might like to do so.

3.3 Functional Requirements

Appointment Scheduling: Waiting in the hospital lobby is a very boring and unpleasant experience when you are sick. With the Patient Queue one will approximately know to a great degree of accuracy his time of meeting the doctor. We are assuming each patient will be diagnosed for 10 minutes and supposing the doctor practice time starts from 12.00 PM. A patient at number 8 position will see meeting time 1200HRS+70Mins= 1310HRS.

If anyone misses his appointment he will be moved to a position of 3patients further down the queue,so that he should not wait for the queue to finish.

Patient Management: We are thinking about the system keeping in mind that it will help both doctors and patients. It will keep records for the doctor and the patient can access it with the app.

The primary objective here is to provide easy access to all the previous medical reports. Most often we lost the medical reports at the end of the diagnosis. The reports and diagnostics will be shared by a unique key to doctor and patient.

Emergency Service: In the context of our country there is no single app/service that can help to find an ambulance in situations of medical emergency. Using GPS and Google Maps API we are trying to implement point to point solution for ambulance service. Each call will detect the user points and send it to the nearest Emergency Center.

Doctors Profile: Doctors profile will help the visitors to know more about them. It will also emphasize on one’s expertise and practice hours, patient’s comments and recommendation and other relevant details and information as per requirements.

Patients Profile: The patients profile will be open to visiting doctors only. Only the patient chooses his/her record visibility here. The profile will be private to maintain privacy of information.

Prescription/Medication: The feature will help the patients find the medicines they need. The patient may forget to take the prescription while going to buy the medicines. Additionally, each time the patients may need the prescription to have medicines with proper timings and dosages. The app will have a complete medication file for current medicines

3.4 Non-Functional Requirements

There are a number of non-functional requirements that are expected from the system. They are the following:

- **Consistency** – The system should perform consistently to provide services to the users, the interfaces of the system are to be consistent and standardized such that end users are comfortable with using it.
- **Convenience** – Convenience in using the system to carry out operations is a necessary requirement to avail the services.
- **Availability** – It is an essential requirement, the system has to be available and operational 24 hours in order to ensure that there is no disruption in the operations.
- **Usability** –The system needs to be user-friendly and easy to use so that users are encouraged to use it and do not confused or discouraged away from it otherwise.
- **Security** – Ensuring security and privacy of the users and their personal data is very important; user passwords to allow authorized access only and mechanisms to prevent unauthorized access shall be in place.
- **Reliability** – The healthcare management system must be reliable in terms of its functions, it must deliver the functional requirements efficiently.

Chapter 4: System Design

Following the system analysis stage in the project development cycle is the system design stage. This stage is vital in the sense that the proposed procedures of operations described in the form of activity diagrams will eventually result in software. In this section the activity diagrams, use case diagrams and data dictionary for the system will be presented. The reason for giving activity diagrams and use case diagrams is that they are an efficient and effective means to convey how particular actions on the system are carried out to the reader which can be comprehended by quick glance. The data dictionary provides a list of the tables needed, which for a system of this size and scope is rather numerous, including details such as field name, data type, data description, key information (i.e. if the field is a primary key or not) and reference.

4.1 Activity Diagram

Appointment Module

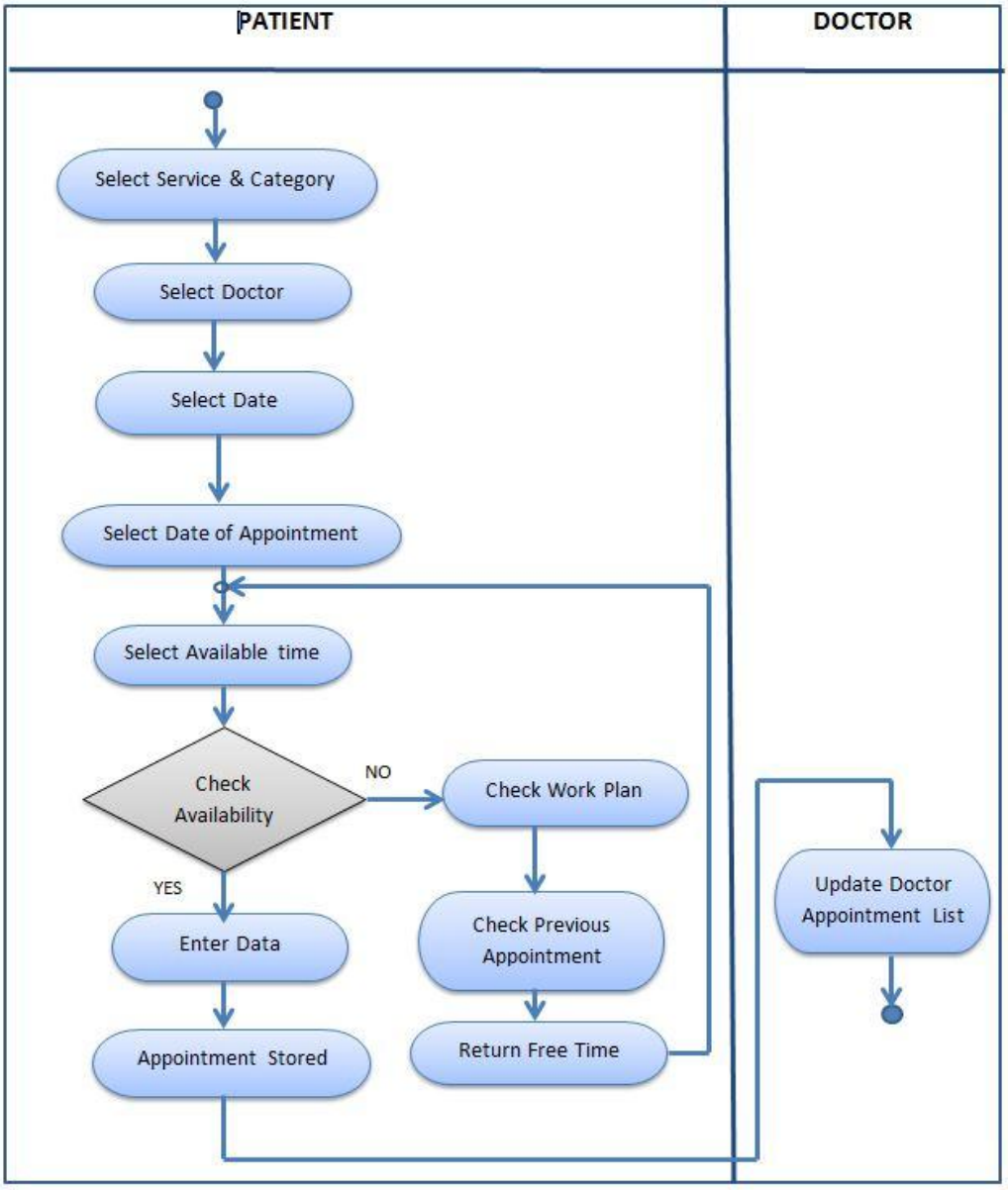


FIGURE 4.1: Making an appointment

This figure shows the process of making doctor's appointment. At first, the patient will choose a particular location (or choose any location), service category and doctor and book a suitable time for his/her checkup. If doctor is not available during that time, then he/she can make appointment according to doctor's free time. Finally the appointment will be added to the appointment list and it will go to the doctor's end.

Appointment Queue Maintenance Module

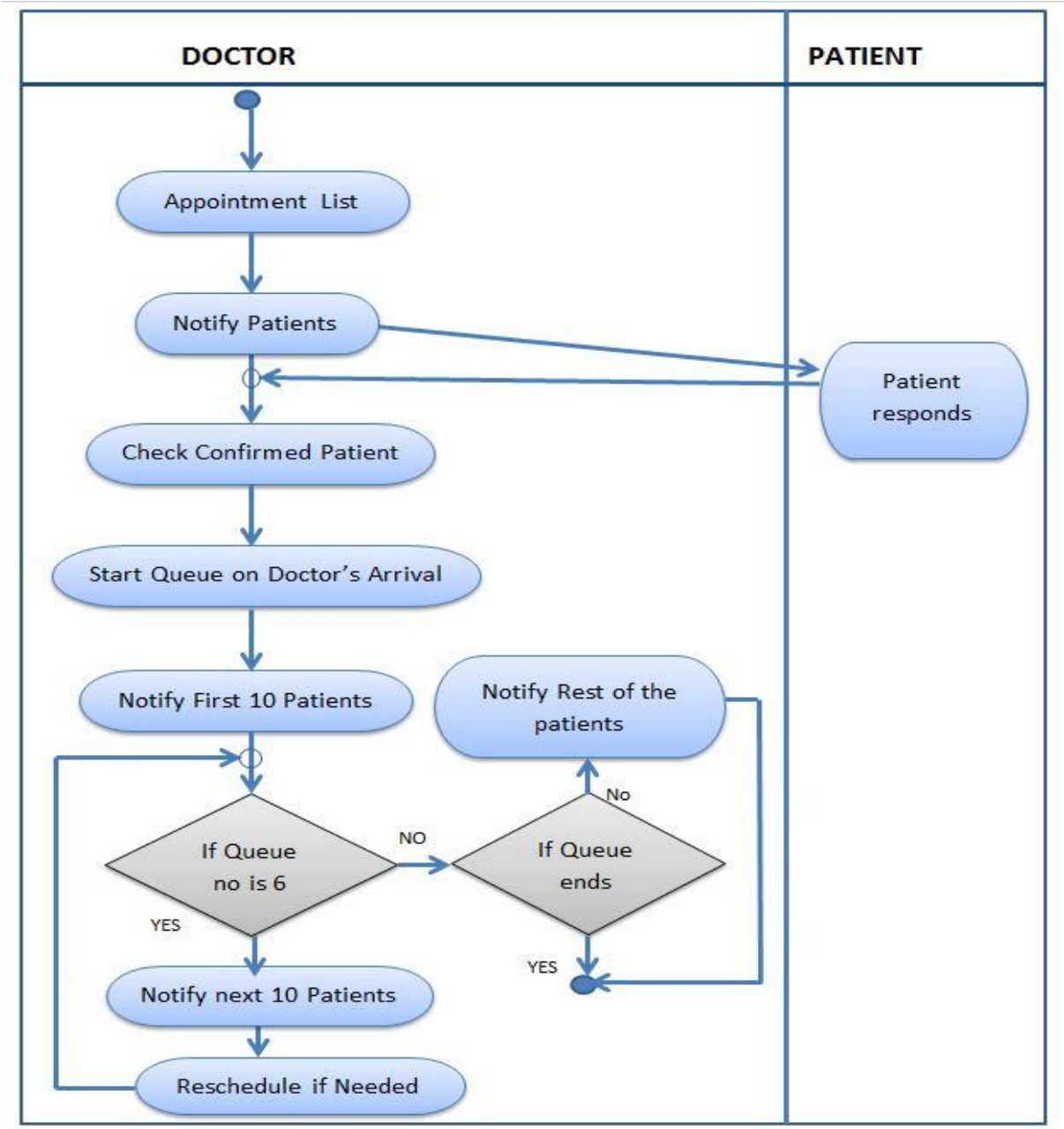


FIGURE 4.2: Appointment queue maintenance by doctor

This figure shows the procedure of sending SMS alert to the patients. At first, the doctor (or attendant) will check the appointment list and notify all patients about their appointment. He/she will update the list according to the response. Secondly, the attendant will send SMS to first 10 patients at the time of doctor's arrival via this system. The system will generate SMS to next 10 patients when the queue reaches position no 6. Appointment can be rescheduled if needed.

Map Module 1: Emergency Ambulance Calling

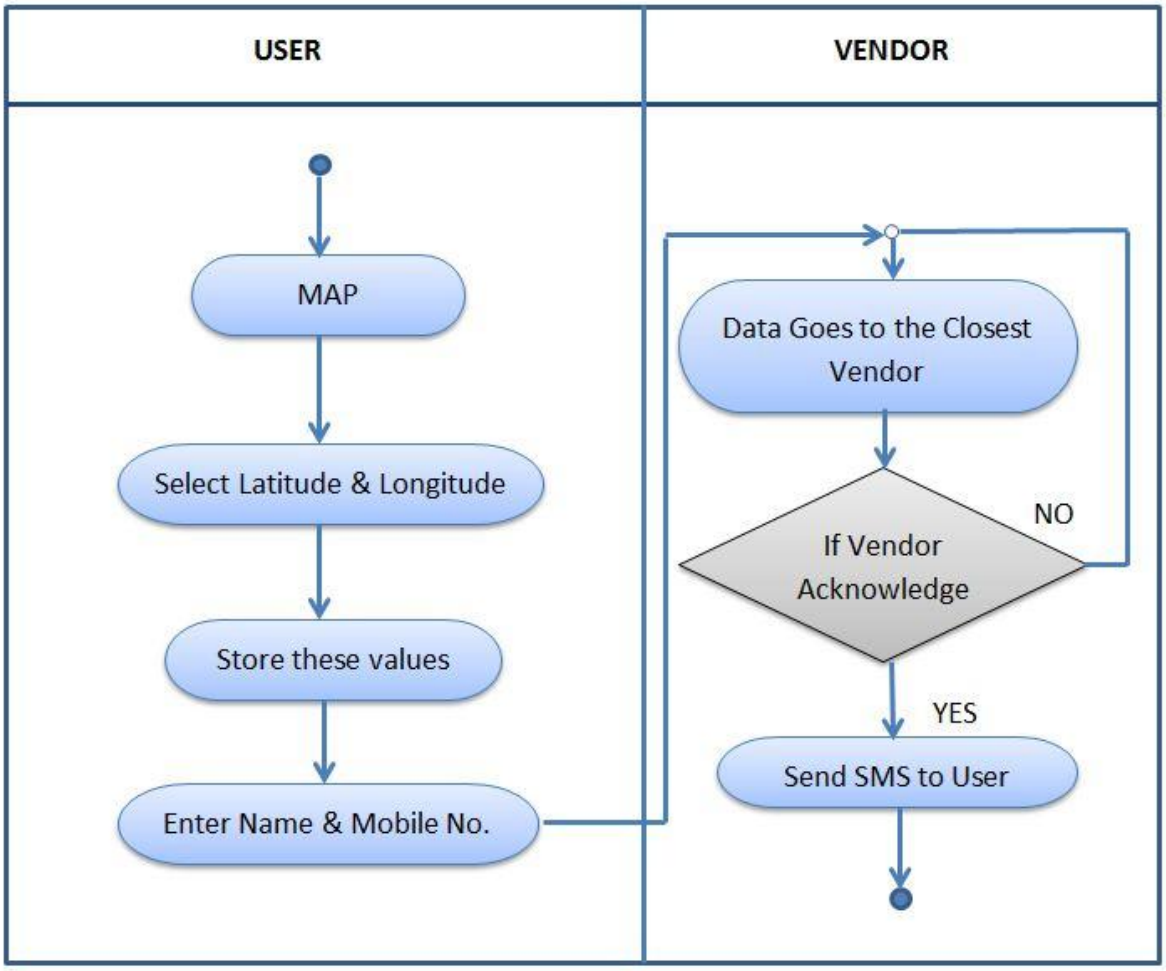


FIGURE 4.3: Signaling ambulance service operator in case of emergency

This figure shows the procedure of how emergency assistance calling system works. First of all user will select the address via map (by moving a point to his/her exact location). The system will generate the latitude and longitude values through the map and store the value in the database. Then, the ER request will be sent to the closest ER service providers which are linked with the system. If they acknowledge, then the user will get the confirmation message or else the ER request will be available there for the other service providers.

Map Module 1: Ambulance Operator Response

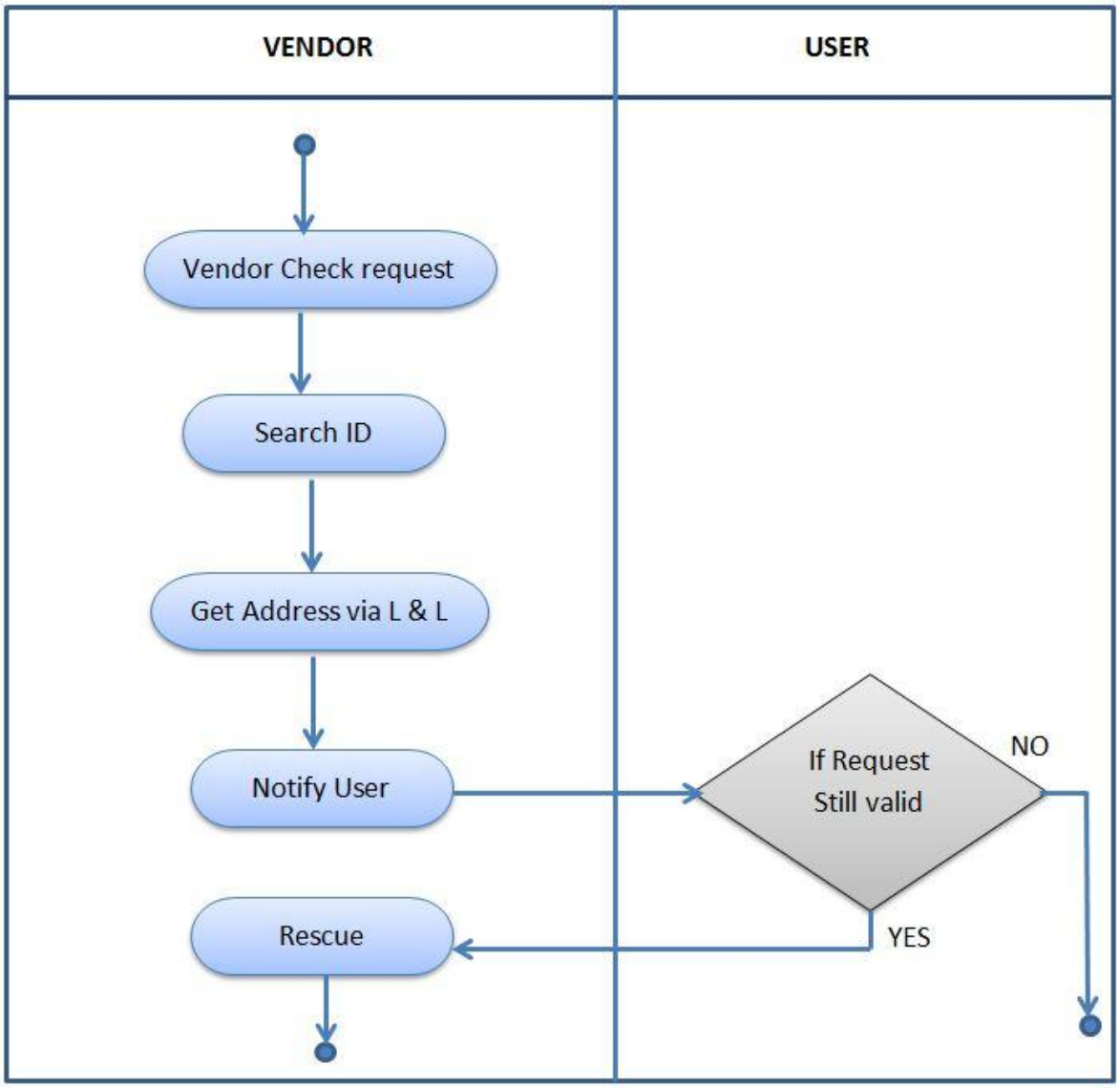


FIGURE 4.4: Responding to an emergency call

This figure shows how the ER service provider acknowledges user request. As soon as the vendor gets an ER request, they will select the user from a table that is viewable to them to search for that user to get the latitude and longitude values of his/her location. Then they will notify the user and according to user's reply they will give ER service.

4.2 Use Case Diagram

Appointment Management

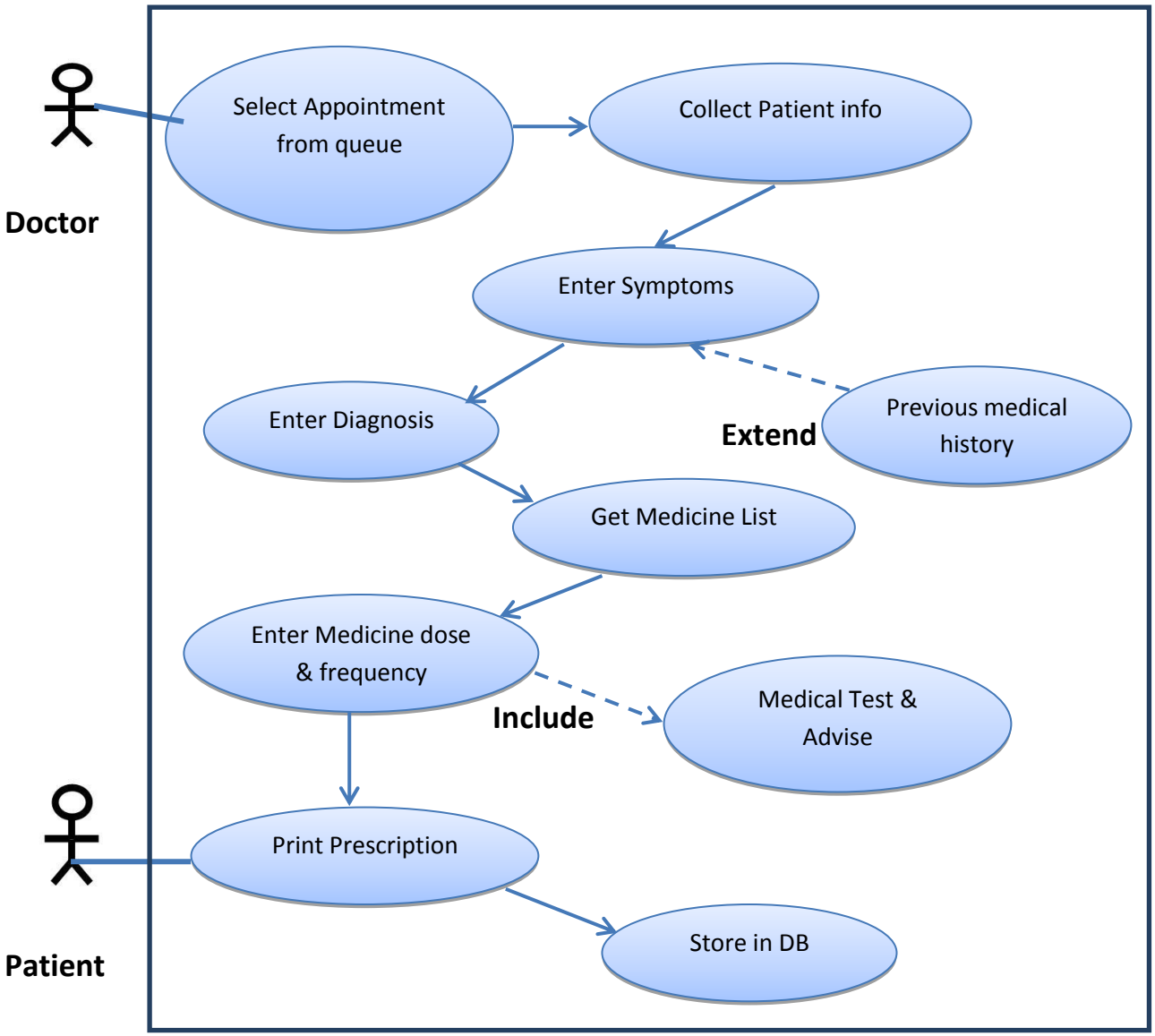


FIGURE 4.5: Appointment Management

The Use Case Diagram above shows the procedure of creating and printing the prescription. The doctor after consulting with the patient and making a diagnosis; will give a medicine list according to patient's symptoms and previous medical history. He or she can include some medical tests if needed. Finally the prescription will be printed and stored in the database for future reference.

Billing Management

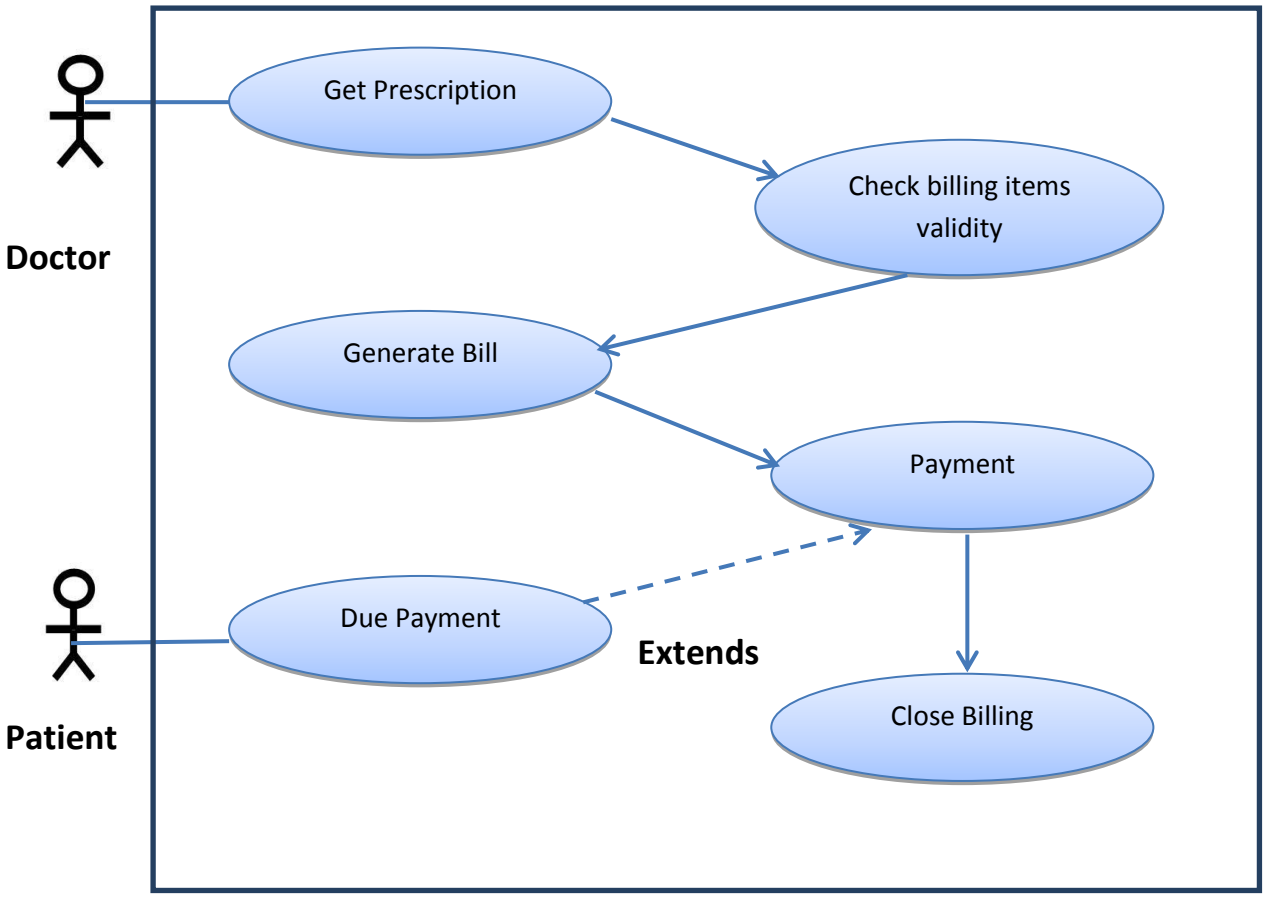


FIGURE 4.6: Billing Management

The Use Case Diagram above shows the billing procedure of the system. After generating the bill of a patient, attendant will check whether there is a due payment or not. If everything is fine, then he/she will print the billing copy and close the patient's billing page.

4.3 Data Dictionary

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Appointment ID	PRI	
app_fname	varchar(255)	Name		
app_email	varchar(255)	Email address		
app_phone	varchar(255)	Telephone No.		
app_reason	text	Appointment Reason		

Table 4.1: Appointment Details

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Appointment ID	PRI	
book_date	varchar(255)	Date of appointment		
start_time	varchar(255)	Appointment start time		
end_time	varchar(255)	Appointment end time		
notes	text	Notes		
id_doctor	int(10)	Doctor ID	FR Key	
id_patient	int(10)	Patient ID	FR Key	
services_name	varchar(255)	Name of service		
id_appdetails	int(10)	Appointment ID	FR Key	
app_status	varchar(10)	Status of the appointment		

Table 4.2: Doctor Appointments

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Doctor ID	PRI	
qualifications	varchar(255)	Qualifications		
category	varchar(255)	Category		
specialities	varchar(255)	Speciality		
hospitals	varchar(255)	Hospitals work for		
languages	varchar(255)	preferable languages		
professional_det	text	professional details		
shortbio	text	short biography		
user_id	int(10)	User ID	FR Key	

Table 4.3: Doctor Qualifications

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Doctor service ID	PRI	
name	varchar(255)	Doctor's name		
duration	int(11)	Consultancy duration		
price	decimal(10,2)	Fee		
currency	varchar(32)	Currency		
description	text	Description		
service_categories	varchar(255)	Professional details	FR key	
user_id	int(10)	Used ID	FR key	
location	varchar(20)	Location		

Table 4.4: Doctor Services

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Workplan ID	PRI	
user_id	int(10)	User ID	FR key	
workplan	text	Working plan		

Table 4.5: Doctor Workplan

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Emergency ID	PRI	
name	varchar(100)	Name of the caller		
email	varchar(100)	Email address		
number	int(10)	Phone No.		
lat	varchar(100)	Latitude		
lng	varchar(100)	Longitude		

Table 4.6: ER Information

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Emergency Operator ID	PRI	
name	varchar(255)	Name of the Operator		
address	varchar(255)	Address		
phone	int(20)	Phone No.		
other_phone	int(20)	Other Contact no.		
email	int(255)	Email ID		
status	int(20)	Status		
lat	varchar(100)	Latitude		
lng	varchar(100)	Longitude		

Table 4.7: ER Operator

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Patient ID	PRI	
dob	varchar(10)	Date of Birth		
gender	varchar(10)	Gender		
bloodgroup	text	Bloodgroup		
weight	int(5)	Weight		
height	int(5)	Height		
user_id	int(10)	User ID		

Table 4.8: Basic Patient Information

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Patient ID	PRI	
diabetes_own	varchar(5)	Diabetes status of patient		
diabetes_fam	varchar(5)	Diabetes status of family		
heart_own	varchar(5)	Heart disease status of patient		
heart_fam	varchar(5)	Heart disease status of family		
cholestrol_own	varchar(5)	Cholesterol level of patient		
cholestrol_fam	varchar(5)	Cholesterol level of family		
bp_own	varchar(5)	Blood pressure of patient		
bp_fam	varchar(5)	Blood pressure of family		

heartack_own	varchar(5)	Heartattack history of patient		
heartack_fam	varchar(5)	Heart attack history of family		
stroke_own	varchar(5)	Stroke history of patient		
stroke_fam	varchar(5)	Stroke history of family		
user_id	int(10)	User ID		

Table 4.9: Patient History

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	User ID	PRI	
first_name	varchar(20)	First Name		
last_name	varchar(20)	Last Name		
email	varchar(255)	Email ID		
mobile_number	varchar(15)	Mobile phone number		
phone_number	varchar(15)	Telephone number		
address	varchar(255)	Adress		
city	varchar(50)	City		
state	varchar(50)	State		
zip_code	varchar(8)	Zip_ Code of the city		
notes	text	Notes about everything		
login_id	int(10)	Stroke history		

Table 4.10: User Details

Field Name	Data Type	Data Description	Key	Reference
uid	int(10)	User ID	PRI	
u_name	varchar(255)	User Name		
u_email	varchar(255)	User Email ID		
u_password	varchar(255)	User Password		
u_type	varchar(10)	User Type		
first	varchar(5)	First time or not		

Table 4.11: User Login

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Picture ID	PRI	
path	varchar(255)	Directory path of the picture		
u_id	int(10)	User ID	FR Key	

Table 4.12: Profile Pictures

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Medicine list ID	PRI	
name	varchar(100)	medicine name		
price	int(6)	price of the medicine		

Table 4.13: Medicine Table

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Test list ID	PRI	
name	varchar(100)	Name of the test		
price	int(6)	Fee of the Test		

Table 4.14: Diagnostic Test Table

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Prescription details ID	PRI	
symptoms	text	Symptoms		
diagnosis	text	Diagnosis		
advice	text	Medical Advice		
doc_id	int(10)	Doctor ID	Foreign Key	
patient_id	int(10)	Patient ID	Foreign Key	
app_details	int(10)	Appointment ID	Foreign Key	
date	varchar(10)	Current Date		

Table 4.15: Prescription Details Table

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Prescription Medicine ID	PRI	
med_name	varchar(50)	Name of the Medicine		
dose	varchar(10)	Dosage		
freq	varchar(10)	Frequency		
doc_id	int(10)	Doctor ID	Foreign Key	
patient_id	int(10)	Patient ID	Foreign Key	
app_details	int(10)	Appointment ID	Foreign Key	

Table 4.16: Prescription Medicine Table

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Prescription Test ID	PRI	
name	varchar(20)	Name of the test		
notes	text	Notes		
doc_id	int(10)	Doctor ID	Foreign Key	
patient_id	int(10)	Patient ID	Foreign Key	
app_details	int(10)	Appointment ID	Foreign Key	

Table 4.17: Prescription Test Table

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Bill info ID	PRI	
date	varchar(13)	Date		
time	varchar(10)	Time		
doc_id	int(10)	Doctor ID	Foreign Key	
patient_id	int(10)	Patient ID	Foreign Key	
total_bill	decimal(10,2)	Total Bill		
vat	decimal(6,2)	Value Added Tax (VAT)		
grand_total	decimal(6,2)	Grand total		
paid	varchar(4)	Payment status		

Table 4.18: Bill Information Table

Field Name	Data Type	Data Description	Key	Reference
id	int(10)	Bill Det ID	PRI	
ser_name	varchar(15)	Service Name		
ser_price	decimal(5,2)	Service price		
ser_qty	int(3)	Quantity		
total_amm	decimal(10,2)	Total amount		

Table 4.19: Bill Details Table

4.4 Mobile Based Application

The establishment and improvement of doctor-patient interaction system is a very important requirement, especially now when the mobile communication technology is developing rapidly. The advantages of mobile web can be used to make up the time and reduce the distance gap between doctors and patients and to provide fast and adequate medical services. Through the connection between mobile terminals and specific service, both doctors and patients are able to obtain required data to achieve a better interaction. Android is a Linux based open source operating system which is mainly used in portal devices with excellent performance thus making its market share growing. The platform, Web services and database technology are all gradually maturing, so that we can develop a doctor patient interaction system on Android platform to meet the needs of the patient and provide doctors more efficient and convenient means of communication with patients.

4.4.1 System Architecture

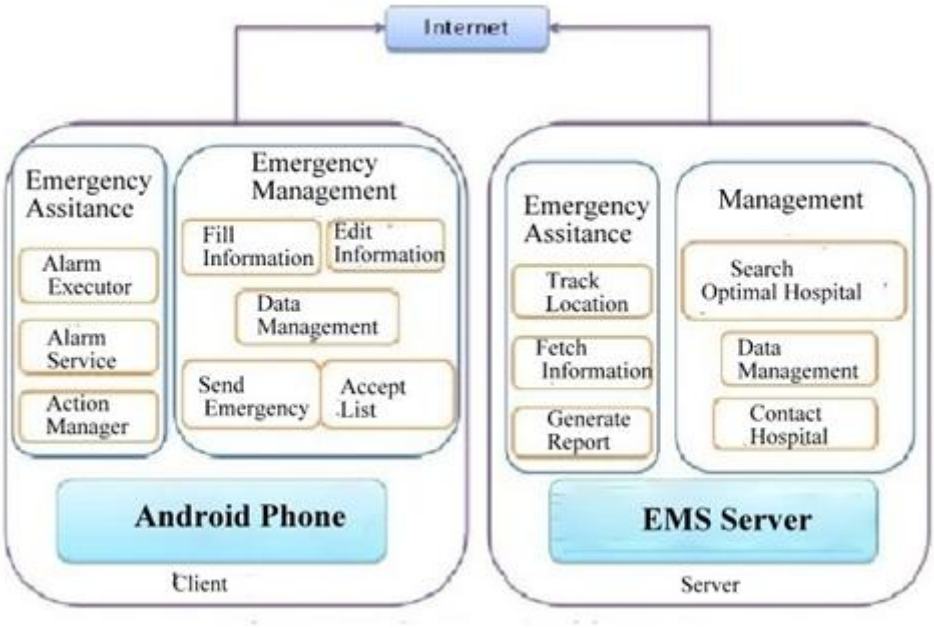


FIGURE 4.7: System architecture of EMS

System Architecture is divided into two parts client (or user) side and server side. At user side all user function takes place like filling personal and medical details of users end emergency, editing user information, etc. User android phone is connected with server by internet. Searching optimal hospital, creating and viewing appointment, finding its distance from user, informing hospital about the patients' location and emergency and informing patient about the hospital is done by server. Database management is also done by EMS server.

Chapter 5: System Implementation

The system implementation stage is the part of the software development process where the system designs planned and developed in the previous section materialize as the system is implemented by coding the individual module or subsystems. The separate modules are developed according to the system specifications and system designs that were determined in the previous sections. The front end user interfaces and the back end database were made and functionality was provided by creating and establishing links between the two. The important parts for this section are a database schema of the database system developed, a list of the programming languages used, some specific details about the coding of the project, a description of the algorithms devised and optimization activities, tools used and an overview of the system infrastructure.

5.1 Database Schema Design

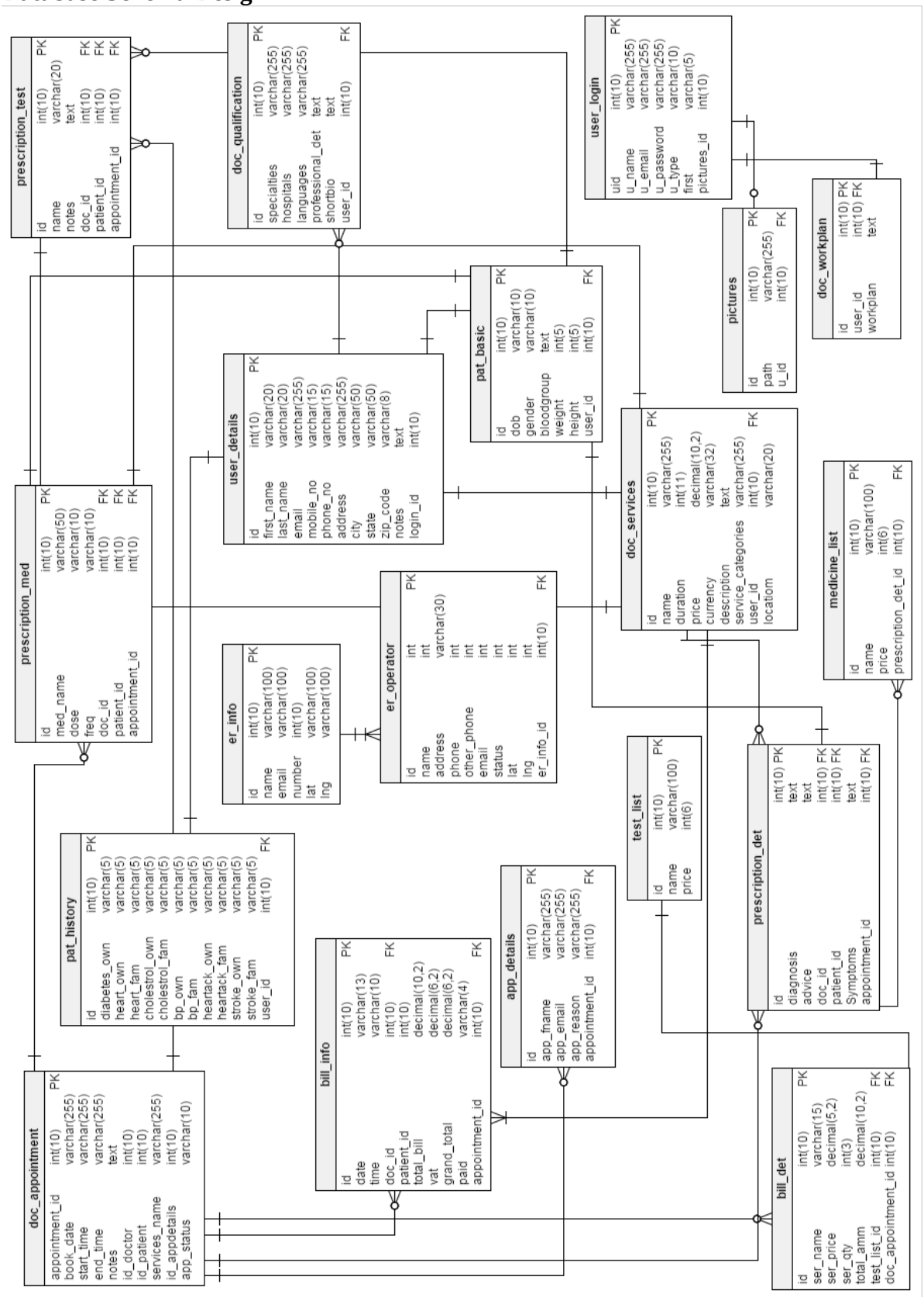


FIGURE 5.1: Database Schema Design

5.2 Programming Languages Used

Development of the system required programming of the individual components or modules in the most appropriate programming language or scripts available, further work had to be done to integrate the different modules or sub-systems to one another to provide full functionality. There was also need for smooth integration between the wed based and the mobile based applications. So overall it did involve a substantial amount of programming for which a number of suitable programming technologies were utilized. To develop the front-end user interfaces the familiar HTML5 and CSS3 technologies were used with JSON being used as one of the means of linking the front-end interfaces and the back-end databases. The client side programming includes use of JavaScript and Jquery, while server side programming involves use of PHP5. The back-end work predominantly involves use of MySQL and Ajax.

The mobile-based application is an Android system. The mobile application or Android app, as such software system are commonly called, has been developed using Java and the view screens created making use of XML. The local database in the mobile application was created using SQLite. Additionally; plug-ins and libraries used for making this project includes Bootstrap, JS datepicker (for the calendars used in the scheduling and appointment management system), JS tiny mice (for the editors used in the prescription making process), etc.

5.3 Programming

A standard procedure that has been used is to include files to reduce redundant codes within the system files. In most of the cases the PHP include function has been used for accessing and using codes written in files like the header, footer and connection files.

JQuery has been used in the front end development, where it is advantageous in view manipulation purposes, it is simple and efficient, additionally it supports Asynchronous JavaScript and XML. Due to using JQuery, use of a large number of libraries was made possible as opposed to other JavaScript libraries. Using JQuery enables any DOM (Document Object Model) objects to be parsed within a single line.

AJAX allows web pages to be updated asynchronously as small volumes of data is exchanged between the client and server (such as in our case where it is necessary to show numerous small changes in data).

Using foreign keys in MySQL; the fundamental reason behind this is simple; to reduce the size of the data. Also to minimize the size of data in term of the scope of test data, we have kept the field sizes at minimal levels, but enough to cover sufficient important information.

Implementing libraries; one of the libraries that we have used provides enhanced user experience by giving a calendar of a better view than what would have been available on default. Also a justification of using available libraries wherever appropriate is that rather than re-developing programs to provide functionality of their own, the libraries can be used efficiently with the opportunity to give a greater focus on implementing and extending from the functionality provided by the libraries to develop better programs for the tasks at hand.

Some sample codes of the system have been provided in the Sample System Codes of the Appendix.

5.3.1 Algorithms and Optimization

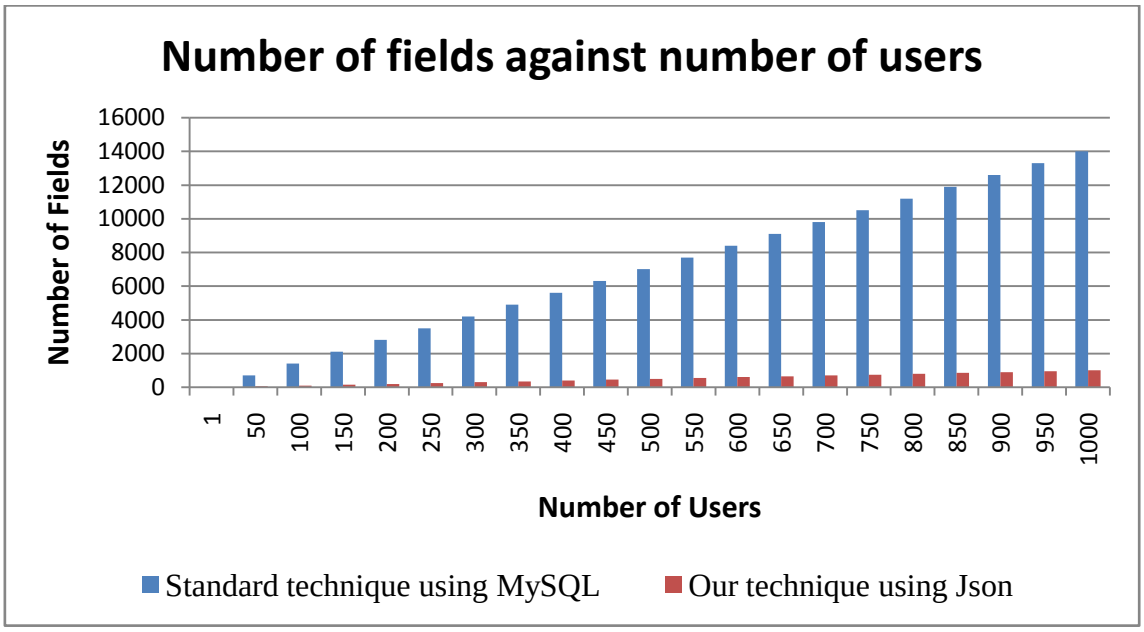


FIGURE 5.2: Improvements in amount of space required

The main aim of this optimization technique is to reduce the redundant data size that would otherwise have emerged while storing this kind of data in the database using standard MySQL and PHP techniques. In that case the number of fields for storing the work plan of the doctor consisting of 7working days with 7 break times in a database would have involved 14 fields. Whereas the technique we have devised and implemented is encapsulation of those 14 fields in a single string that gives us the benefit storing all that information in a single

string, not only has this drastically reduced the number of fields, it has also reduced the total size of the table as MySQL uses a different type of tag to store each field. The figure shown above has been reached using model data for 1000 users for all 14 fields that would exist in standard MySQL structure and that for our technique based on the same number of users.

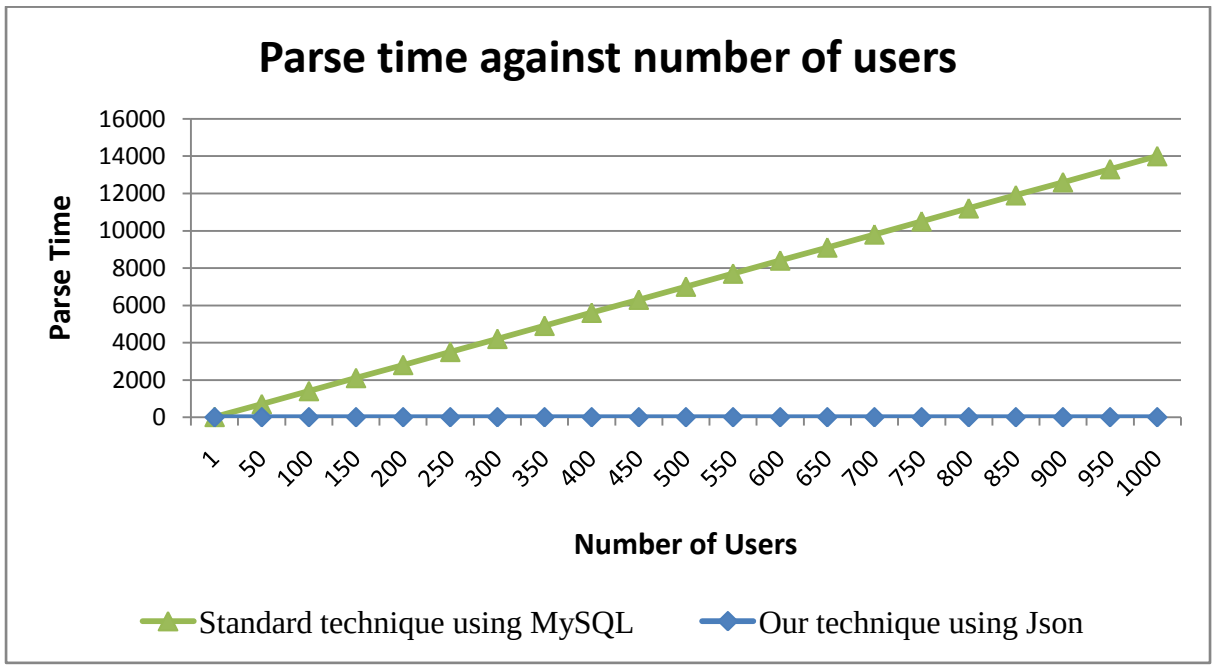


FIGURE 5.3: Improvements in parse time

This figure is basically an extension of the above mentioned optimization strategy showing the improvements (i.e. reductions) in parse time that is possible to bring out by means of implementing our technique. This technique has been used in the doctor appointment time availability selection system where there is need to parse all the available times slots from which patients may choose a slot of their preference. In the more traditional and standard approach a greater number of fields would have to be parsed, whereas using the improved techniquethat number is reduced.

However it must be mentioned that there is a limitation to this technique, that is if this is toused for dynamic data which is constantly or regularly changed; the parse time required for building the Json string will be comparable to that of the normal technique making use of MySQL. If this can be elaborated, it can be said that although this technique may be great for manipulating individual JSON objects, such as here in the case of a specific object which is an individual doctor's work plan, it no longer yields the mentioned benefits in cases where entire tables can be used.

Appointments

In the case of appointment scheduling problem we have made attempts to address as many kinds of clashes or conflicts as possible:

i). Clash with break time

ii). Clash with other appointments

iii) Check of clashes in three different ways; right overlap, left overlap and full overlap

iv) For rescheduling the appointments in a case where a doctor might be delayed in starting to examine patients, rather than push back all appointments by the specified delay time, the rescheduled times have been calculated by taking the gaps (time slots where the doctor is taking a break or is free) into account, where appointments which have a free time slot ahead of them do not get pushed back down the queue, rather they remain at their position.

End User Messages

A key aim of this project was to reduce the suffering of patients as they often have to wait for long hours waiting for a doctor to examine them, the waiting period can be very tiresome and difficult particularly for a sick person when long hours pass as they wait while the doctor is examining other patients or he/she is yet to arrive and start examining patients in the first place. For that our system will periodically send notification updates in the form of SMS (Short Message Service) via a server and gateway.

5.4 Tools Used

The text-editor chiefly used was Sublime Text due to its cross platform text and source code editing ability and compatibility to support numerous APIs which was a useful factor considering the variety of programming languages used in making the project system. Another justification for opting to use Sublime Text was our extensive familiarity and experience with the software. The system was uploaded and maintain in a localhost server using XAMPP for our development and internal testing purposes, XAMPP is a free to use and open-source cross-platform web server package system within which is incorporated Apache (file server), PHP 5.5 (server-side programming) and MySQL (database) technologies. As a version control tool in a team development environment Git had been chosen to serve as an online repository for storing and merging our programs. From the

mobile based application side Android Studio (based on IntelliJ IDEA) was the IDE (integrated development environment) of choice, with Genymotion being the preferred Android emulator for application testing and presentation purposes. Additional systems used include Google Drive for storing and sharing relevant documents during the period of word and Adobe Photoshop for image editing tasks. For design of diagrams like the schema diagram, we have made use of Edraw Max and Vertabelo.

5.5 System Infrastructure

Deployment of the system for operational use definitely requires certain specific and as well as some generic elements or components which collectively can be labeled as the system infrastructure which for a project of this scope and dimension is somewhat considerable. This subsection will describe the requirements necessary for deployment of the system including hardware and system requirements for both client and server sides.

5.5.1 Hardware Specifications

For the purpose of providing continued and quality service to the target group the system needs to be deployed into web servers that will be capable of hosting system and sustaining the daily traffic resulting from patient management operations.

Server-Side Requirements

Server Components	**Specifications**
Processor	Intel Xeon E3-1270 (4 cores 8 threads), 3.50Ghz(8M Cache)
RAM	32GB (DDR3)
Space	1000 GB, SATAIII 7200 RPM
Bandwidth	10,000GB (per Month)
IPAdresses	05

Table 5.1: Service Side Requirements

Since the system is primarily a web based one and the users have to be constantly connected with the server there is need for a dedicated server. Having such a server will enable a minimum load of 500 regular doctors and 1000 registered patients to be handled.

Client Side Requirements

On the client side (doctor) our system may be used in 3 (three) tiers or levels of devices in term of specifications based on user preference or requirements as applicable depending on their circumstances. The client side local computer with minimum requirements can work with our system.

Components	**Basic**	**Semi**	**Full**
Processor	Intel Quad Core	Intel i3	Intel i5
RAM	1GB	2GB	2GB
Space	10GB	10GB	10GB
Internet Bandwidth	1MBPS	1.5MBPS	2MBPS
Printer	--	Yes	Yes
POS Printer*	--	Yes	Yes
Waiting Display	--	--	Yes

Table 5.2: Client Side Requirements

Mobile Based Application Requirements

In order to cover around 80% of the devices available and used in the market at present we have made the application with compatible with minimum Gingerbread Android Operating System. A more detailed specification is provided below.

Mobile Components	**Specifications**
Operating System	Android 3.0 (Ginger Bread)
Processor	1.0 GHz Dual Core
RAM	512 MB
ROM	4 GB

Table 5.3: Mobile Based Application Requirements

5.5.2 Web Application Requirements

In order to maintain the operations of the web based application there is need to have at least the following components installed in the dedicated server.

Components	**Specifications**
Operation System	Linux / Ubuntu / Windows Server 2012
PHP	Version 5.5.x
My SQL	Version 2.4.x
CURL	Enabled

Table 5.4: Web Application Requirements

5.5.3 System Infrastructure Diagrams

Service Side Infrastructure

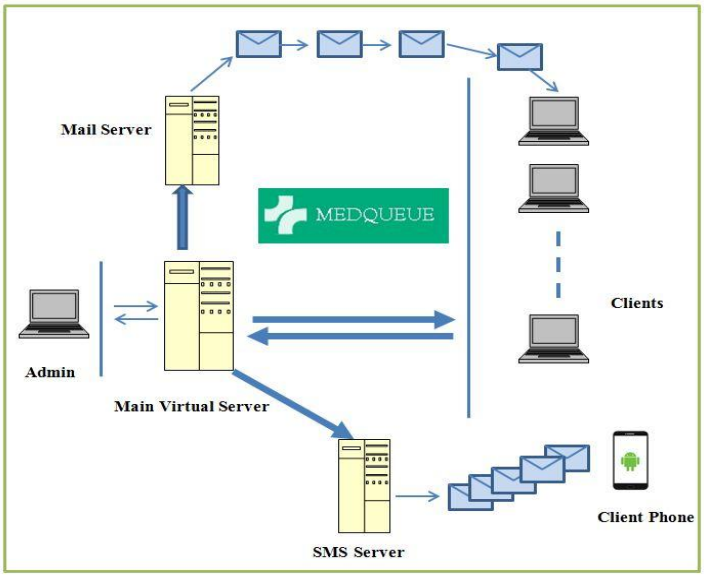


FIGURE 5.4: Service Side Infrastructure

The figure above provides a graphical representation of the service side infrastructure. A main server will be maintained that will host the MedQueue system which shall be maintained by an administrator of the service. The client devices will communicate with the main server through the internet. Updates and notification messages will be transmitted to the clients in the form of emails and SMS, the service for which will be provided by email and SMS servers with which the main server shall also communicate.

Client Side Infrastructure

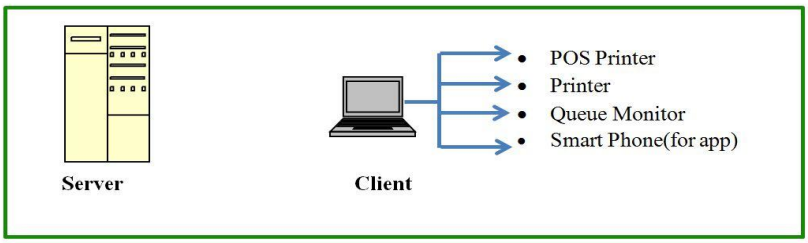


FIGURE 5.5: Client Side Infrastructure

The figure above gives a graphical representation of the client side infrastructure. The client computers will communicate with the main server, the client computer can also be connected to printers for printing prescriptions and bills, a display unit to show relevant information and a smart-phone (to run the web based application).

Chapter 6: Business Model

The doctor-patient management software systems available in the market at present are of the range Tk. 300,000 to Tk.500,000 in terms of cost. They are tailor made by software development firms to meet the requirements of a single large client such as a big hospital which provides doctors for consultation as well as diagnostics tests among other services. A single such system is used by a single client institution to cater to their own requirements, from our study we have found there is no centralized doctor-patient management system that may be used by doctors who operate their own private practice chamber or for small to medium sized clinics. The expensive commercial software systems in the market are not a feasible solution for them as the costs involved simply to not pay back in terms of benefits. However a system developed specifically according to local requirements and available for use by those who cannot have access to the above mentioned systems will provide the payoff which is roughly equal to their willingness to pay for such a system, allowing both parties, the doctors using the service and the providers of the service to benefit from such an arrangement.

6.1 Model Details

Cost Benefit Analysis (CBA) of MedQueue:

Year	**2016**	**2017**	**2018**	**2019**	**2020**	**Total**
Increased sale	30000	37000	48000	60000	75000	
Reduction in customer complaint calls	500	500	490	490	485	
Reduced inventory cost	5000	5000	5000	5000	5000	
Total Benefits:	35500	42500	53490	65490	80485	
PV of Benefits:	32272	35124	40188	44730	49975	202289
PV of All Benefits:	32272	67397	107585	152315	202290	
Server	5000					
Software licenses	2000					
Development labor	28000					
Total Development Costs:	35000					
Hardware	7000	10000	10000	10000	10000	

Software	1500	1500	1500	1500	1500	
Operational Labor	6000	8000	10000	12500	15000	
Total Operational Costs:	14500	19500	21500	24000	26500	
Total Costs:	49500	19500	21500	24000	26500	
PV of Costs:	45000	16115	16153	16392	16454	110114
PV of all Costs:	45000	61124	77277	93669	110123	
Total Project Benefits- Costs:	-14000	23000	31990	41490	53985	
Yearly NPV:	-14000	19008	24034	28338	33520	100900
Cumulative NPV:	-12728	15373	39408	67746	101267	
Return on Investment:	81.878%					

Table 6.1: Cost- Benefit Analysis

The table above shows a breakdown of the calculations done for the purpose of a cost benefit analysis to explore the commercial success and viability of the service once it has been deployed for operational use. While the figure below is a graphical representation of the information to show the projected costs and benefits for a period of the next 4 years to show where a break-even point is reached. On the bases of the preliminary analysis that has been carried out, the break even-point is expected to be within a time of around 1.5 years.

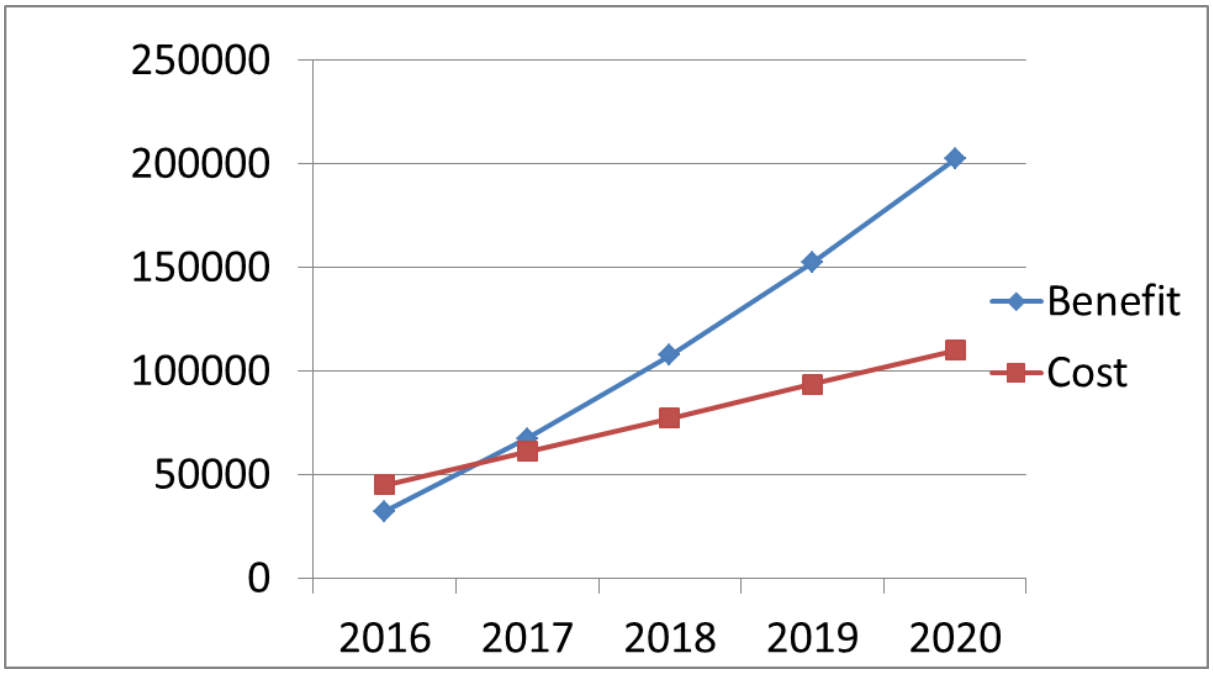


Figure 6.1: Break- Even Point of CBA

Chapter 7: Conclusion

Although the concept of a doctor-patient management system is not a new one, however there is no such general system in existence that can be used as a service by individual doctors and patients irrespective of their institutional or organizational affiliations and preferences. The system is low cost, easy to use and efficient in delivering functionality according to its intended objectives. Moreover a system of this type is the first of its kind in Bangladesh; it has been specifically designed according to local environment, requirements and specifications.

7.1 Limitations

The project currently being in the state of a model can be considered a limitation, in its present state it can serve as a demonstration of the capabilities that the full system can deliver. Additionally due to the project being developed for the partial fulfillment for the requirement of our degree, it was not possible to deploy the system to an external server with which the client devices could connect to for testing purposes, making us reliant on using a local host and connected the devices locally using our own network. The concept of testing brings lead to the next limitation, with the system not being deployed on a server, testing and evaluation using a sample of users with real time data generated could not be carried out. Also from the doctor side, the system will be operated on a daily basis by the doctor assistance/attendant, although there is no provision right now for distinguishing between a doctor and his/her attendant, it can be addressed relatively easily creating a new user type.

7.2 Future Work

The future development work that can be perceived for this project is deploying it in the field; at first for testing purposes with an appropriately sized and distributed sample to test its performance and obtain feedback which is an important part of the software development process, and finally into operational use as a viable service being provided to those who seek it.

Furthermore the system can be improved and expanded into a consolidated platform to provide easier access to healthcare and healthcare related services; such as including diagnostic centers, blood banks and pharmacies into the system as well to create a single platform to provide access to a multitude of useful and necessary services along the lines of the centralized system that has been envisioned.

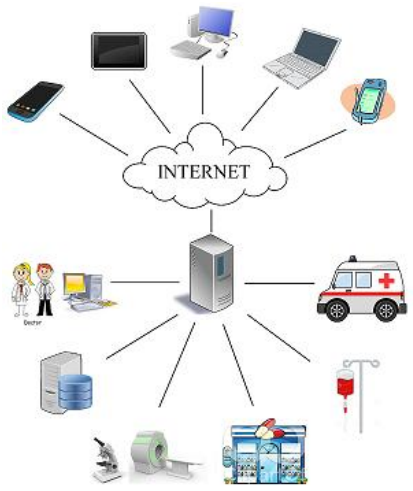


FIGURE 7.1: Future Work

Other future modifications that can be made include a system of user ratings, providing additional features and services in the mobile based application, etc. which provide a better user experience.

7.3 Personal Reflection

Working on a complete system like this one has provided an opportunity for development of relevant experience and skills on our side. While working on the design and development stages of making this system we can say that we have learned a lot, we feel that the growth in human capital that has come as a result of it will be advantageous to us in our lives in the near future. Additionally from the external payoff point of view, our system has direct implications in the sense that it is something from which the society as a whole can be benefitted as soon as it is implemented. This fact along with a sense of achievement associated with development of the project is in our perspective an important matter to reflect on.

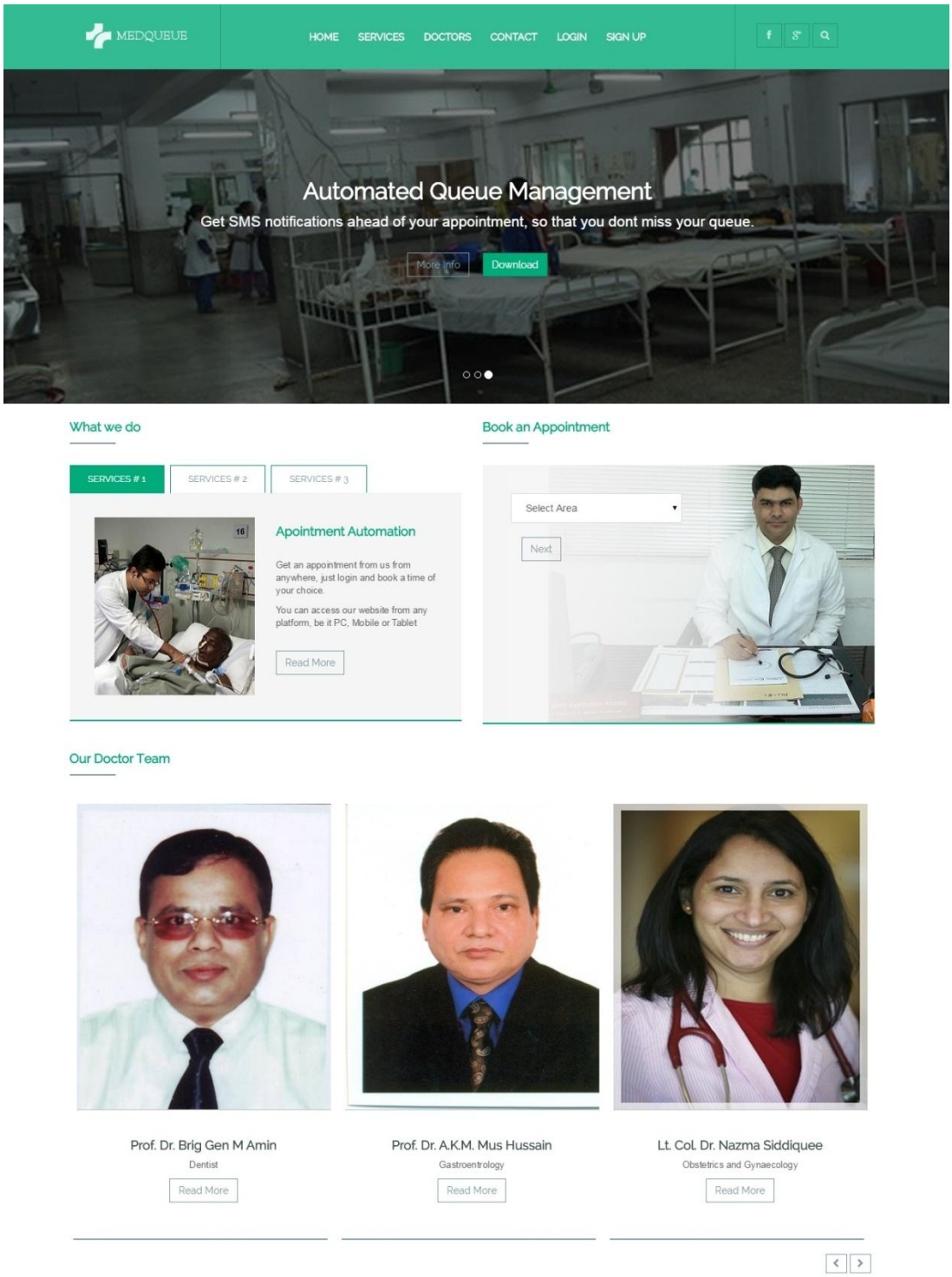
References

1. Mukama, F. (2013). A study of the health information systems at local levels in Tanzania and Mozambique: Improving use and management of information in health districts. Retrieved from www.ub.uib.no/elpub/Norad/2003/uio/thesis02.pdf
2. Kaduruwane, I. (2012). An Empirical Investigation of Health Information System Failure in Regional Sri Lanka. Retrieved from http://eprints.qut.edu.au/50663/1/Indika_Kaduruwane_Thesis.pdf
3. Kwame, A. (n.d.). The occurrence of medication errors and the occurrence of risk factors for medication errors in state hospitals in Ghana: Patients' safety improvements in focus. Retrieved from http://www.utwente.nl/bms/htsr/education/completedassignments/Kwame.pdf
4. Mwangi, C. (2013). Computerization of the Kenyan Health Care Records. Retrieved from https://www.theseus.fi/bitstream/handle/10024/56572/sherrythesis.pdf
5. Allen, S. L. (2009). *The E-Health Patient Record Management System*.
6. Electronic Patient Records. (n.d.). Retrieved from http://patient.info/doctor/electronic-patient-records
7. Fichman, R. G., Kohli, R., & Krishnan, R. (2011). The Role of Information Systems in Healthcare: Current Research and Future Trends. *Information Systems Research*,*22*(3), 419-428. Retrieved from http://www.btrc.gov.bd/content/internet-subscribers-bangladesh-june-2015
8. Hanna, K. E., Anderson, S. M., & Maddox, S. D. (2005). *Think Research Using Electronic Medical Records to Bridge Patient Care and Research*. Retrieved from Faster Cures website: http://www.fastercures.org/assets/Uploads/PDF/ThinkResearch-red.pdf
9. Kemkar, O. S. (2015). Knowledge Management and Web: The Clinical Diagnostic Process Scenario for Ehealth Delivery Model. *American Journal of Computer Science and Information Technology*, *3*(1), 007-013. Retrieved from http://pubicon.info/index.php/AJCSIT/article/view/10
10. Lee, D. H., Rabbi, A., Choi, J., &Rezai, R. F. (2012). Development of a Mobile Phone Based e-Health Monitoring Application. *International Journal of Advanced Computer Science and Applications*, *3*(3), 38-43. Retrieved from www.ijacsa.thesai.org
11. Lu, J. (2013). *E-health Web Application Framework and Platform Based On The Cloud Technology* (Master's thesis, Kristianstad University, Kristianstad, Sweden). Retrieved from http://www.diva-portal.org/smash/get/diva2:647835/FULLTEXT01.pdf

12. Luo, S., Hong, Y., Liu, Q., Wang, L., &Dssouli, R. (n.d.). *Implementing Web-based e-Health Portal Systems*. Retrieved from Department of Computer Science and CIISE, Concordia University website: http://www.albany.edu/faculty/hong/pub/eHealthPortal.pdf
13. Maddox, E. N., & Heine, R. P. (n.d.). Hospital Management Reform: a Step to Healthcare Reform. *Journal of Management and Marketing Research*. Retrieved from http://www.aabri.com/manuscripts/10443.pdf
14. Perlman, M., & Davis, B. (n.d.). Healthcare Without Walls - Delivering the Future Paradigm. *e Healthcare - Patient Management Without Walls*, 8-11. Retrieved from http://www.oracle.com/us/industries/healthcare/ehealthcare-patient-management-wp-173073.pdf
15. Yen, T. Y. (2011). *Restaurant Management System*. School of Arts and Science Tunku Abdul Rehman College.
16. Yen, W. X. (2011). *Restaurant Management System*. School of Arts and Science Tunku Abdul Rehman College.
17. Reaz, Z. (2015, January 7). Market Insight: The Smartphone Industry in Bangladesh. Retrieved from http://lightcastlebd.com/blog/2015/01/market-insight-market-smartphones-bangladesh
18. Optimizing JavaScript code. (2015, August 18). Retrieved from https://developers.google.com/speed/articles/optimizing-javascript
19. World Bank Country Data: Bangladesh. Retrieved from http://data.worldbank.org/country/bangladesh
20. Kawnine, N., Guinness, L., Rahman, M. A., Killingworth, J. R., Wong, Y. H., &Hussain, N. (1998). Private Sector Medical Clinics and Hospital Survey (15). Retrieved from Ministry of Health and Welfare website: http://heu.gov.bd/phocadownload/Research%20Paper%20No.15.pdf

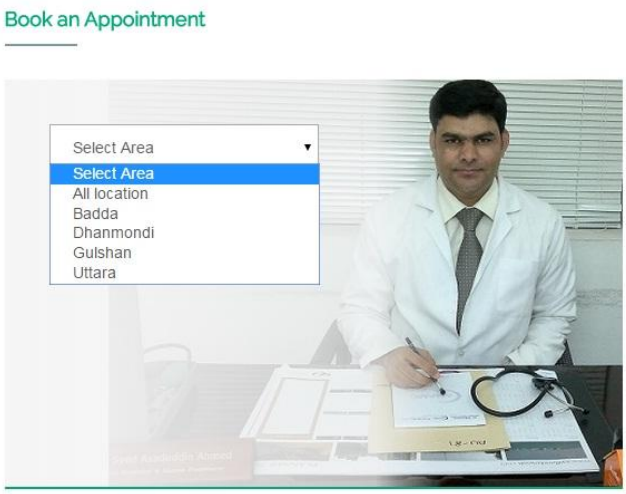
Appendix

Screenshots of the system

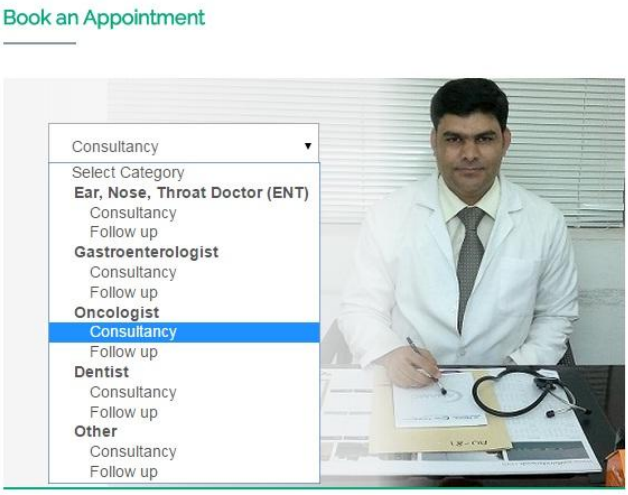


The home page of the system showing content such as the login and sign up options on the top, the box for booking an appointment, some doctors using the service.

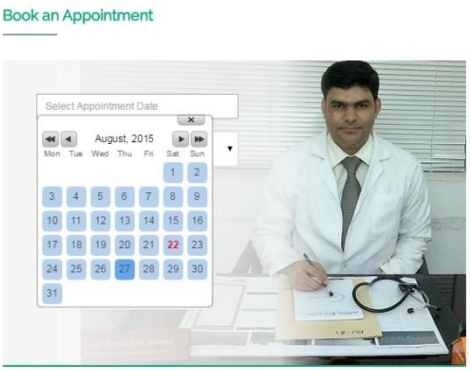
Process for booking an appointment:



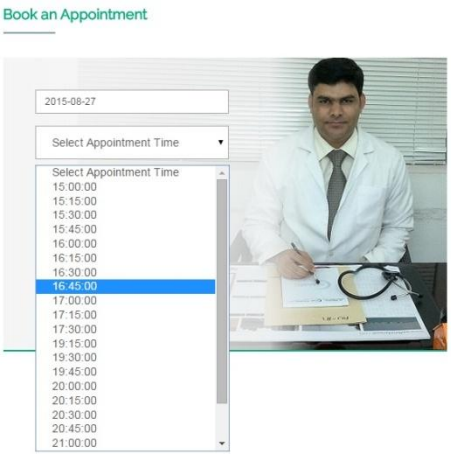
Step 1: Select a specific area or zone such that only doctors giving services in that area will be included in the search results. Or selecting all locations will include all doctors in the search.



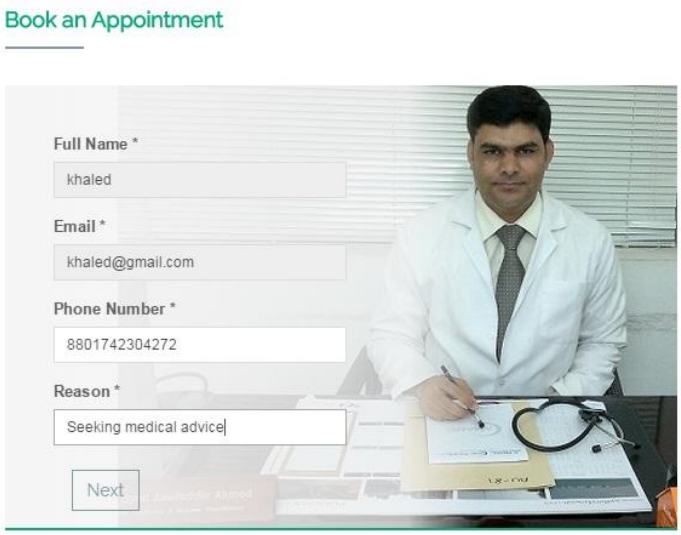
Step 2: Selecting a category or medical field; e.g. Oncologist and then seeking consultancy or follow up services.



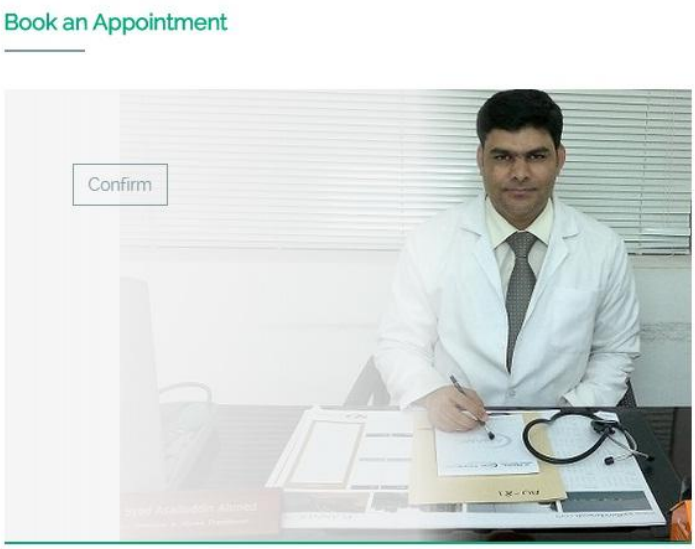
Step 3: Selection of date from a calendar



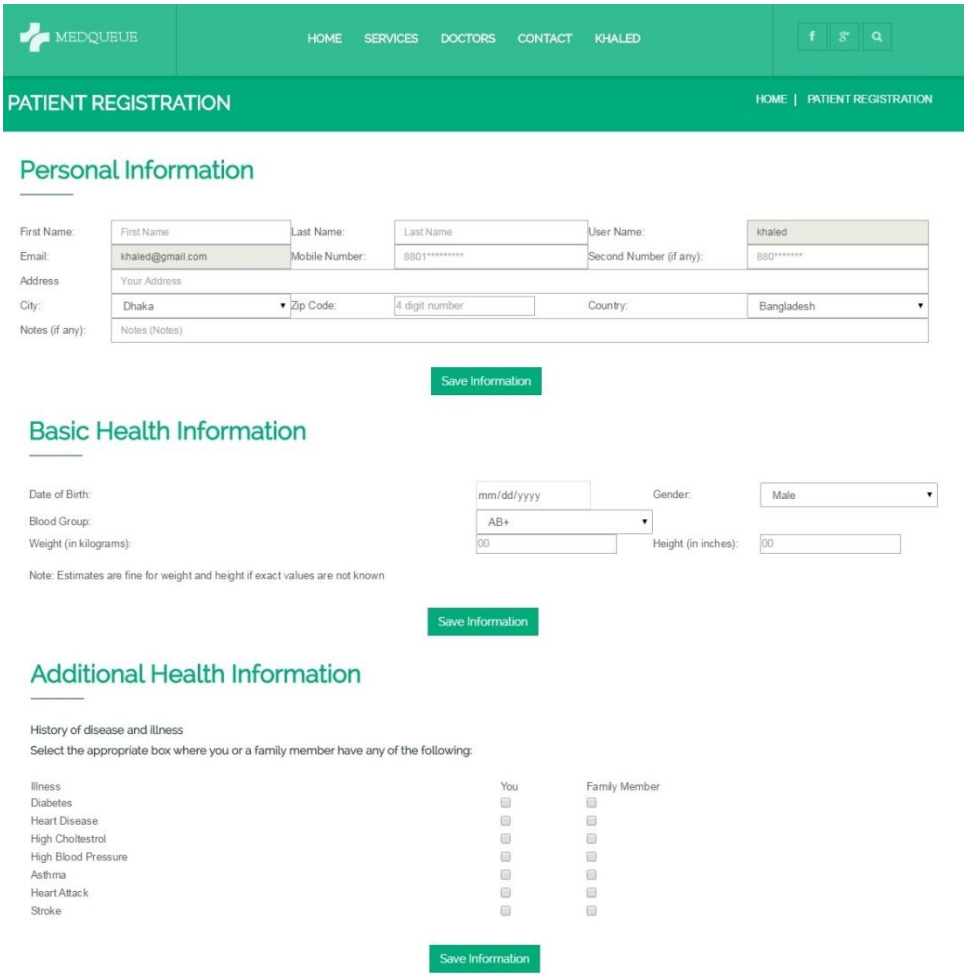
Step 4: Selection of time slot from the available time slots.



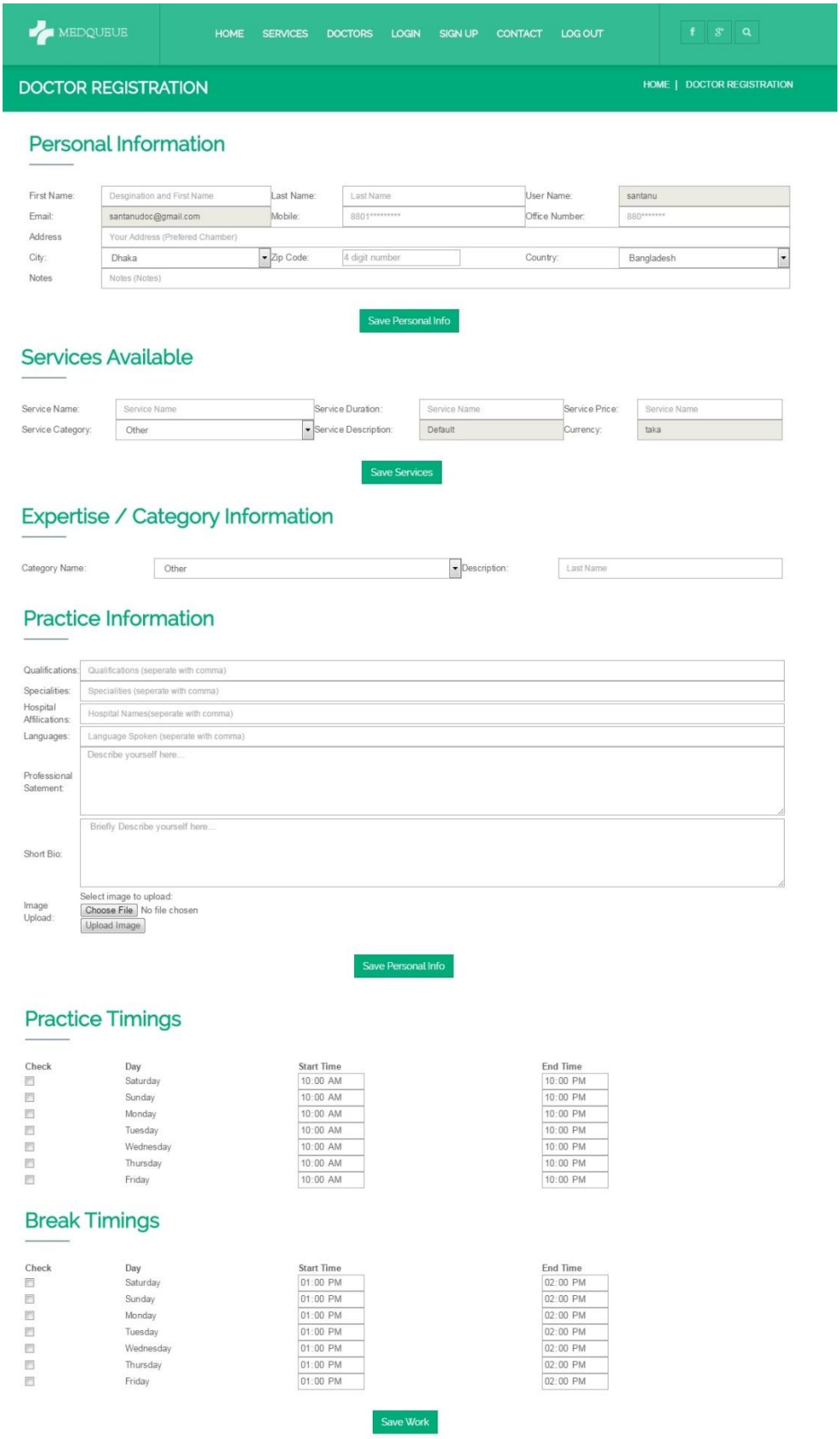
Step 5: Filling up some basic information, user already registered will have some of their information pulled from the database automatically; others will have to fill it up.



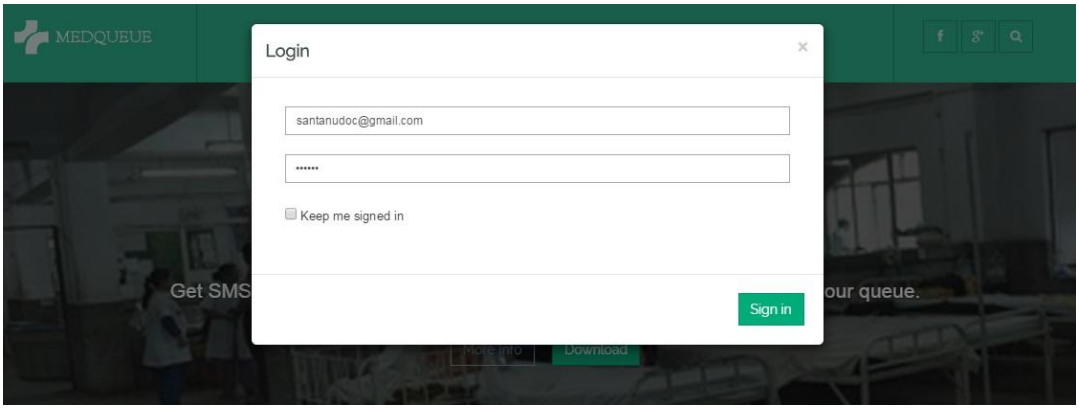
Step 6: The appointment making process has been completed.



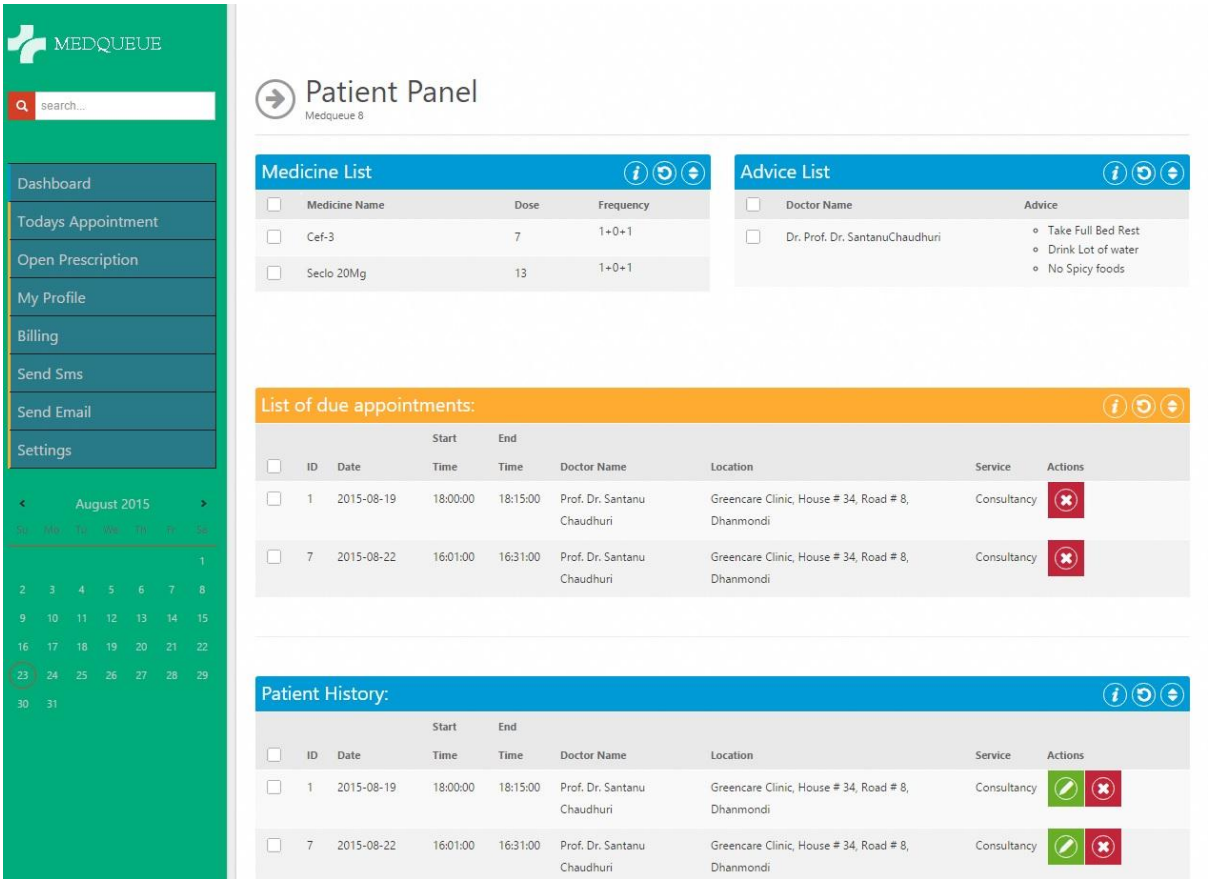
New Patient Registration Page.A new patient registering to the system will have to fill in their details like personal information, illness history, etc. as can be seen in the page above.



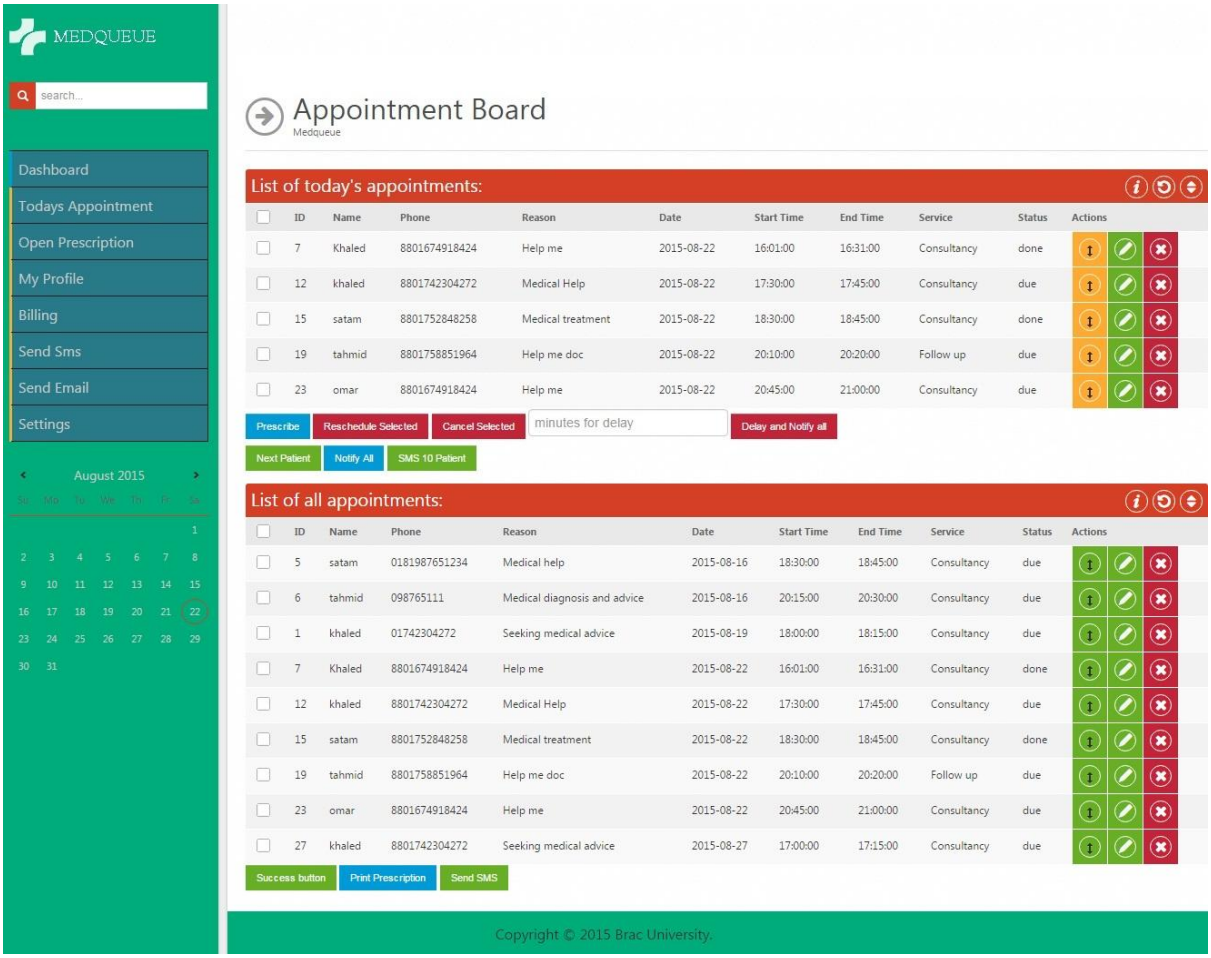
New Doctor Registration Page. A new doctor registering to the system will have to fill in their details like specialty, work plan, qualifications, etc. as can be seen in the page above.



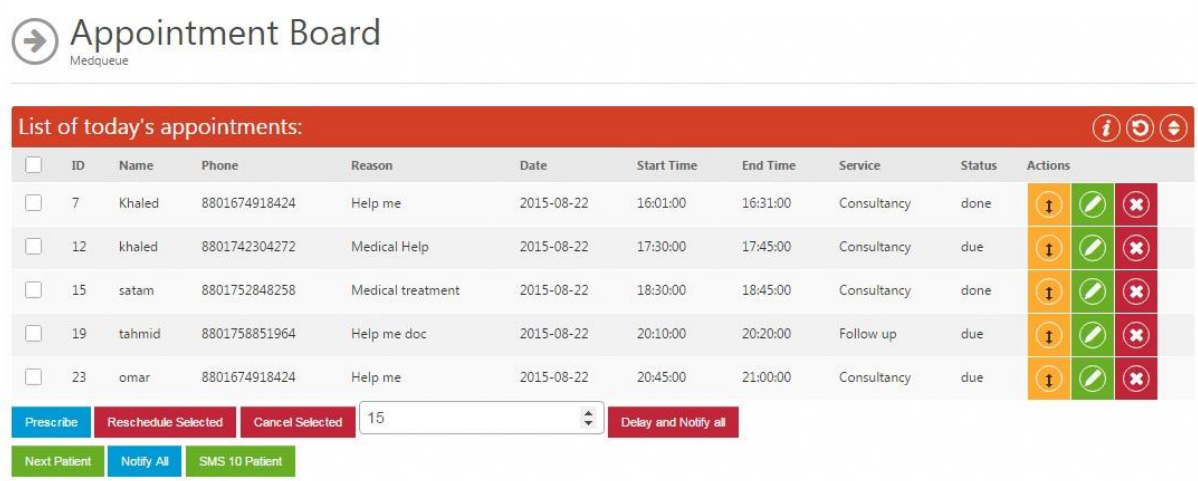
Login Panel: The login panel is shown above. Users (both patients and doctors) will login from here and will be redirected to their specific pages.



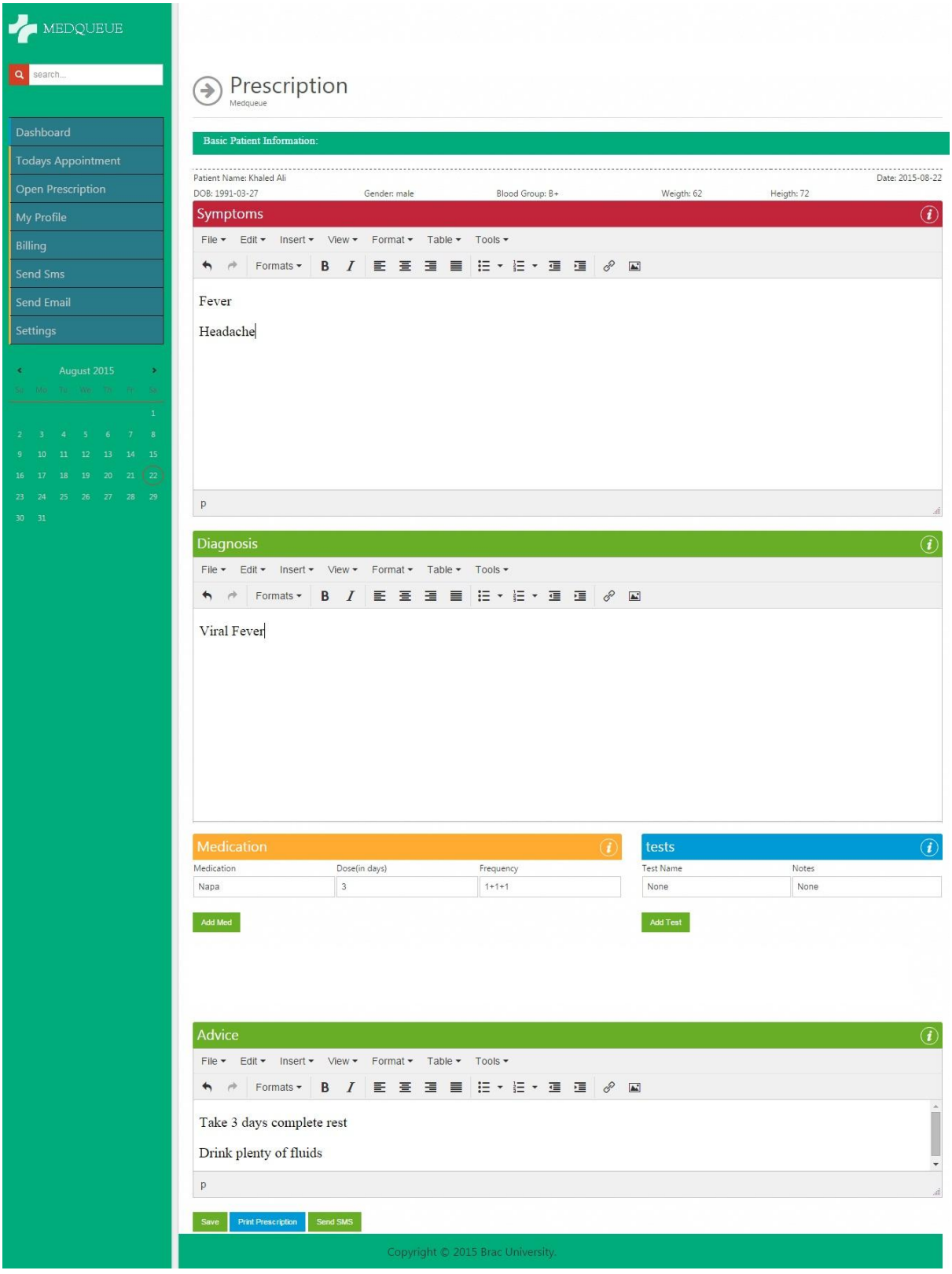
Patient Panel: The screen capture above shows the Patient Panel to which registered patients will be redirected to after they have logged in; relevant information such as medicine list, medical advice, list of due appointments and patient history can be viewed dynamically.



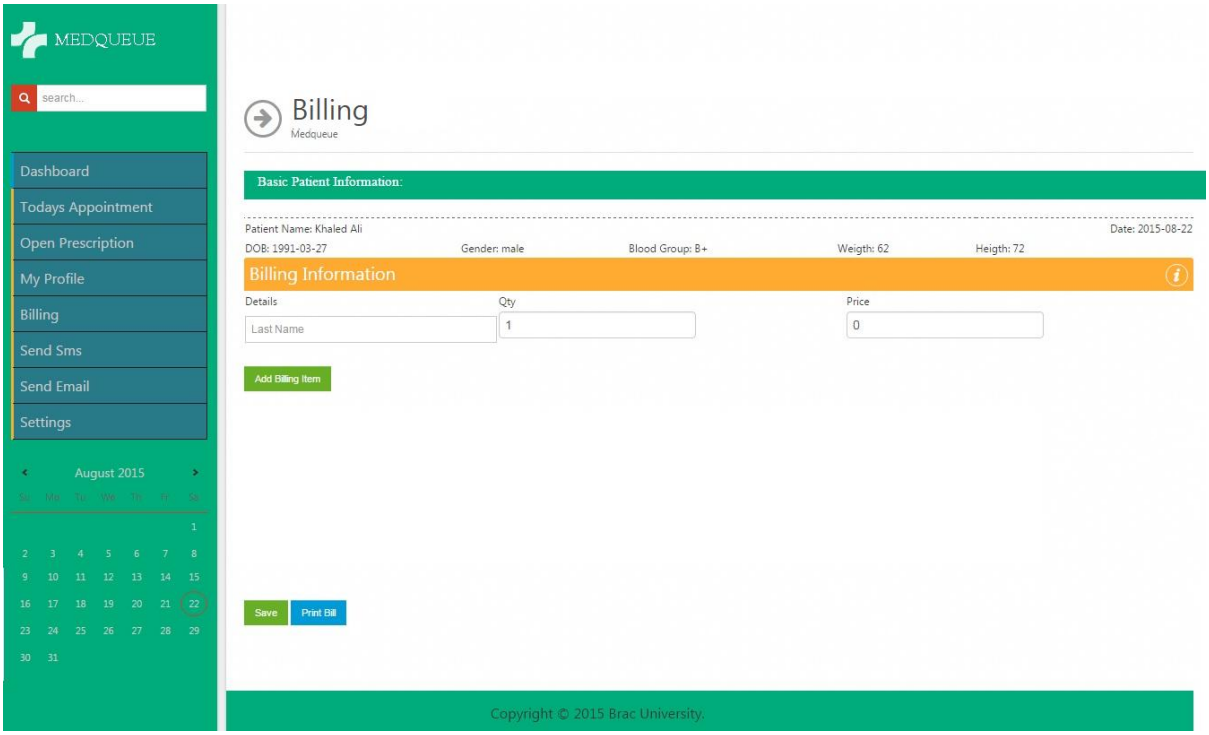
Doctor Panel: Registered doctors will be redirected to their own appointment board where their relevant information will be displayed for them. Firstly there a list showing the present date appointments, below that list is a set of buttons which provide functionality such as after entering a delay time (in minutes) all appointments will be rescheduled as described in the main body text, notification SMS will then be sent. Doctor can also send SMS notifications to the next 10 patients regarding their appointment time. Also buttons are present to call forth the next patient and go the prescription creating page for a selected patient (or current patient).



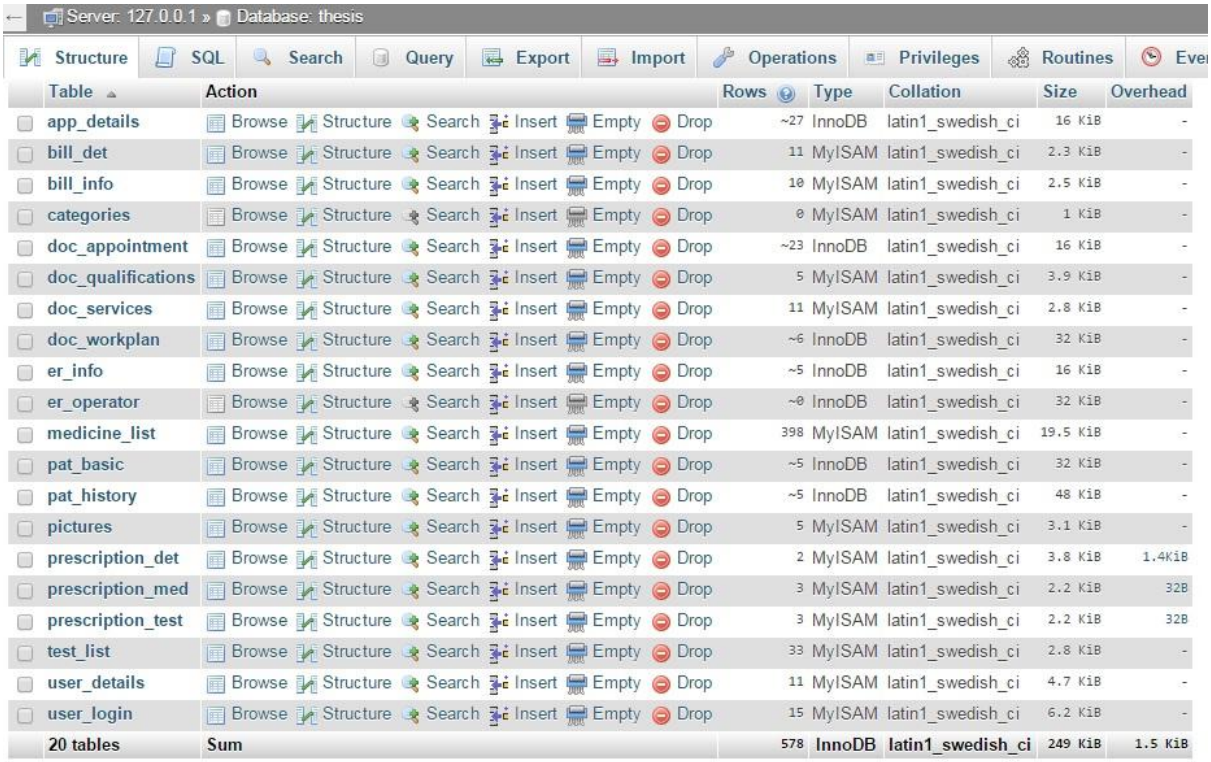
Doctor Panel: The screen above shows (zoomed in) the options available to the doctor at his her panel.



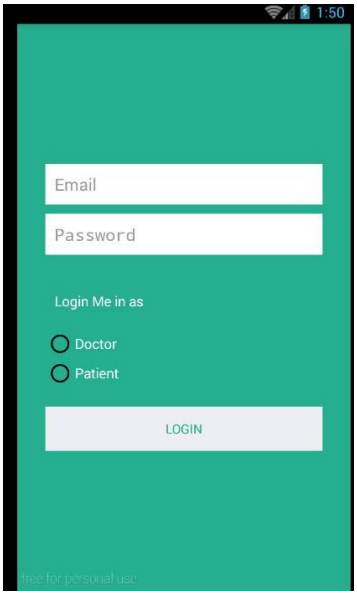
Prescription Panel: The above page is used to fill up the information to complete the prescription digitally; listing symptoms, diagnosis, assigning medication and tests, sending advice. Buttons are for saving the prescription (and medication and test data) in the database. The prescription is computer generated.The next step of the process is the billing stage.



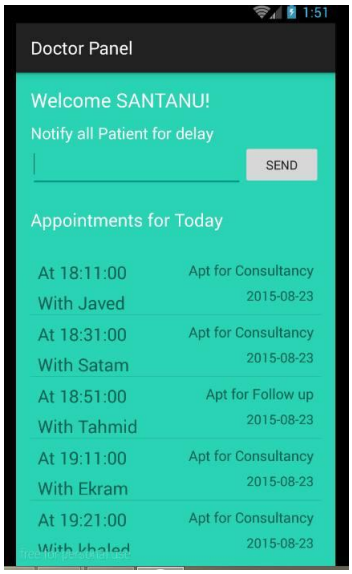
Billing Panel: The page to fill up information for the bill.



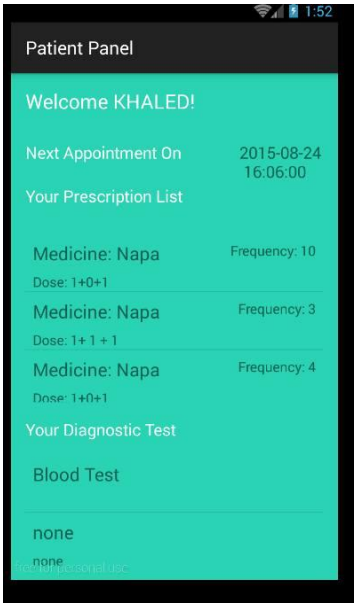
The database system stored in the localhost server.



Mobile based application home screen; email ID, password and user type selection for logging in.



Mobile based application doctor panel showing the appointments scheduled for the present date along with the provision to delay the appointments by a chosen amount of time.



Mobile based application patient panel showing the date and time of the next appointment the patient has and some recently prescribed medicine and tests.

The data provided in the tables below have been used to populate the graphs shown as Figure 5.2 and Figure 5.3. It is to be noted that this data is test data simplified on the basis of our model.

Amount of fields required		
Number of users	**Standard technique using MySQL**	**Our technique using Json**
1	14	1
50	700	50
100	1400	100
150	2100	150
200	2800	200
250	3500	250
300	4200	300
350	4900	350
400	5600	400
450	6300	450
500	7000	500
550	7700	550
600	8400	600
650	9100	650
700	9800	700
750	10500	750
800	11200	800
850	11900	850
900	12600	900
950	13300	950
1000	14000	1000

Table A.1: Change in amount of space required with increase in number of users

Parse time required		
Number of users	**Standard technique using MySQL**	**Our technique using Json**
1	14	1
50	700	1
100	1400	1
150	2100	1
200	2800	1
250	3500	1
300	4200	1
350	4900	1
400	5600	1

450	6300	1
500	7000	1
550	7700	1
600	8400	1
650	9100	1
700	9800	1
750	10500	1
800	11200	1
850	11900	1
900	12600	1
950	13300	1
1000	14000	1

Table A.2: Change in parse time with increase in number of users

Sample System Codes

```
function getDocApp(d_id, day, date) {
	$('#select_appoint_slot').empty();
	$.ajax({
		method: "POST",
		url: "forms/getTime.php",
		data: {
		   d_id: d_id,
		   date: xdate
		}
	   })
	   .done(function(text) {
		var app_data = JSON.parse(text);
		var msg = JSON.parse(app_data[0]);
		//console.log("msg > "+ msg['Sunday']['start']);
		if (msg[day] == "") {
		   $('#select_appoint_slot').hide();
		   $('#appoint_msg').append("<p id='a_msg'>Doctor unavailable!</p>");
		} else {
		   $('#a_msg').remove();
		   $('#select_appoint_slot').show();
		   $.ajax({
			method: "POST",
			url: "forms/getDuration.php",
			data: {
			   d_id: d_id,
			   service_selected: service_selected
			}
		      })
		      .done(function(text) {
			var result = eval("(function(){return " + text + ";})()")
			console.log(result[1].name + ' ' +
			   result[1].duration + ' ' +
			   result[1].price + ' ' +
			   result[1].currency);
			service_duration = result[1].duration;
			var regExp = /(\d{1,2})\:(\d{1,2})\:(\d{1,2})/;
			var doc_startTime = msg[day]['start'];
			var doc_endTime = msg[day]['end'];
			var a_startTime = msg[day]['start'];
			var a_endTime;
			var doc_break_start = msg[day]['break']['start'];
			var doc_break_end = msg[day]['break']['end'];
			var all_app = [];
			var optionHtml = "<option>Select Appointment Time</option>";
			while (parseInt(a_startTime.replace(regExp, "$1$2$3")) <
parseInt(doc_endTime.replace(regExp, "$1$2$3"))) {
			   var flag = true;
			   a_endTime = addMinute(a_startTime, result[1].duration);
```

```
                        console.log("time 1");
                        if (checkOverLap(a_startTime, a_endTime, doc_break_start,
doc_break_end) == true) {
                            for (var i = 1; i < Object.keys(app_data).length; i++) {
                                if (checkOverLap(a_startTime, a_endTime, app_data[i].start_time,
app_data[i].end_time) == true) {} else {
                                    flag = false;
                                    break;
                                }
                            };
                            if (flag) {
                                all_app.push(a_startTime);
                                optionHtml += '<option value="' + a_startTime + '">' + a_startTime
+ '</option>';
                            }
                        };
                        a_startTime = addMinute(a_startTime, result[1].duration);
                    }
                    $('#select_appoint_slot').append(optionHtml);
                });
            }
        });
    }

    function checkOverLap(time1, time2, time3, time4) {
        var regExp = /(\d{1,2})\:(\d{1,2})\:(\d{1,2})/;
        var stime = parseInt(time1.replace(regExp, "$1$2$3"));
        var etime = parseInt(time2.replace(regExp, "$1$2$3"));
        var bkStart = parseInt(time3.replace(regExp, "$1$2$3"));
        var bkEnd = parseInt(time4.replace(regExp, "$1$2$3"));
        if (stime < bkStart || stime > bkEnd) {
            if (etime < bkStart || etime > bkEnd) {
                if (stime < bkStart && etime > bkEnd) {
                    return false;
                } else {
                    return true;
                }
            } else {
                return false;
            }
        } else {
            return false;
        }
    }
```

The JavaScript code above is used for the appointment time selection process

```
$("#delayAll").click(function()
   {
      var docName= $("#form_u_name").val();
      var tablee = checktable('omar');
      var min= $("#dmin").val();
      //alert("enetring while");

      //var i=tablee.length-1;
      var i=0;
      while(i<tablee.length){
         if (min <=0 ){
            break;
         }
         var prstrt= tablee[i][6];
         var prend= tablee[i][7];
         console.log("start_time "+ prstrt+ "       end_time: "+prend);
         if (i<tablee.length-1){
            var diff= checkDiffMin(prend , tablee[i+1][6]);
            console.log("diff: "+diff+" min: "+ min);
         }
         var newstrt= addMinute(prstrt, min);
         var newend= addMinute(prend, min);
         if(diff>0){
            min= min-diff;
         }
         var number=[];
         number.push(tablee[i][3]);
         var msg= "Your Appointmet with Dr. "+ docName+" has been rescheduled to "+
newstrt+"(previous time: " + prstrt+")";
         console.log(msg);
         send_SMS(msg, number);
         $.ajax({ method: "POST", url: "forms/updateDelay.php",data: {
            app_id: tablee[i][1], newstrt: newstrt, newend:newend
         }
         }).done(function(text) {
            //alert("Delay msg:   "+ text);
         });

         i++;
      }
   });
```

The JavaScript code given above has been used for queue management purpose.

```
private void checkLogin(final String email, final String password, final int selectedId) {
        String tag_string_request = "req_login";

        pDialog.setMessage("Loading..");
        showDialog();

        String type;
        if(selectedId == R.id.radioBtnDoctor){
           type = "doctor";
        }else{
           type = "patient";
        }
        String uri =
AppConfig.URL_LOGIN+"?tag=login&email="+email+"&password="+password+"&type=
"+type;

        StringRequest strReq = new StringRequest(Method.GET,
                                        uri,
                                        new Response.Listener<String>() {
           @Override
           public void onResponse(String response) {
              Log.d(TAG, "Login Response " + response.toString());
              hideDialog();
              try {
                 JSONObject jObj = new JSONObject(response);
                 String name = jObj.getString("name");
                 String mail = jObj.getString("email");
                 String type = jObj.getString("type");
                 String uid = jObj.getString("uid");
                 Log.d(TAG, "uid:"+uid);
                 boolean error = jObj.getBoolean("error");
                 if (!error) {
                    db.addUser(name,mail,type,uid);
                    if (selectedId == R.id.radioBtnDoctor) {
                       session.setLogin(true, true);
                       Intent intent = new Intent(LoginActivity.this, DoctorActivity.class);
                       startActivity(intent);
                       finish();
                    } else {
                       session.setLogin(true, false);
                       Intent intent = new Intent(LoginActivity.this, PatientActivity.class);
                       startActivity(intent);
                       finish();
                    }
                 } else {
                    String errorMsg = jObj.getString("error_msg");
                    Toast.makeText(getApplicationContext(), errorMsg+" "+"Error",
Toast.LENGTH_LONG).show();
                 }
              } catch (JSONException e) {
```

```
                e.printStackTrace();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError volleyError) {
            Log.d(TAG, "Login Error: " + volleyError.getMessage());
            Toast.makeText(getApplicationContext(), volleyError.getMessage()+" "+"Error",
Toast.LENGTH_LONG).show();
            hideDialog();
        }
    });
    AppController.getInstance().addToRequestQueue(strReq, tag_string_request);
  }
```

The Java code shown above was used for the login system in the mobile based application.

```
private void fetchPrescription(final String uid) {
        String tag_request = "getPrescription";
        pDialog.setMessage("Fetching Prescription");
        showDialog();
        String uri = AppConfig.URL_LOGIN+"index.php?tag=fetchPres&uid="+uid;

        StringRequest strRequest = new StringRequest(Method.GET, uri,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String s) {
                        Log.d(TAG, "Response" + s.toString());
                        hideDialog();
                        try{
                            JSONObject jObj = new JSONObject(s);
                            boolean error = jObj.getBoolean("error");
                            if(!error){
                                JSONArray responseArray = jObj.getJSONArray("response");
                                for(int i=1; i < responseArray.length(); i++){
                                    JSONObject a = responseArray.getJSONObject(i);
                                    PrescriptionModel pm = new PrescriptionModel("Medicine:
"+a.getString("med_name"), "Frequency: "+a.getString("dose"), "Dose:
"+a.getString("freq"));
                                    prescription.add(pm);
                                }
                            }else{
                                String errorMsg = jObj.getString("error msg");
                                Toast.makeText(getApplicationContext(),errorMsg,
Toast.LENGTH_LONG).show();
                            }
                        }catch (JSONException e){
                            e.printStackTrace();
                        }
                    }
                }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError volleyError) {
                Toast.makeText(getApplicationContext(), volleyError.toString(),
Toast.LENGTH_SHORT).show();
                hideDialog();
            }
        });
        AppController.getInstance().addToRequestQueue(strRequest,tag_request);
    }
```

Java code used in the mobile based application for retrieving patient information like prescription and medication details from the database.