

---

# **Integrated Appointment Management**

**version 1.0**

## **Project Team**

Md. Mohayeminul Islam

Tajkia Rahman Toma

Abu Bakar Siddique

## **Project supervisor**

Dr. Kazi Muheymin-Us-Sakib

**Institute of Information Technology,**

**University of Dhaka**

**23 June, 2011**

The software is developed under course Software project lab-II(SE-505). The developers are the students of IIT 1st batch. The students have been guided by project supervisor Dr. Kazi Muheymin-Us-Sakib.

Project group members:

Md. Mohayeminul Islam (BIT0102) \_\_\_\_\_

Tajkia Rahman Toma (BIT0107) \_\_\_\_\_

Abu Bakar Siddique (BIT0126) \_\_\_\_\_

---

Dr. Kazi Muheymin-Us-Sakib  
Assistant Professor  
Institute of Information Technology  
University of Dhaka

## Table of Contents

<b>Abstract.....</b>	<b>7</b>
<b>Acknowledgement.....</b>	<b>7</b>
<b>Project Plan and Feasibility Study.....</b>	<b>8</b>
<b>    1.1.Background.....</b>	<b>9</b>
<b>    1.2.The Project Idea.....</b>	<b>9</b>
<b>    1.3.Proposed Time line.....</b>	<b>9</b>
<b>    1.4.Feasibility Study.....</b>	<b>10</b>
<b>Software Requirement Specification.....</b>	<b>11</b>
<b>    2.1.Introduction.....</b>	<b>12</b>
2.1.1.PURPOSE.....	12
2.1.2.PROJECT SCOPE.....	12
<b>    2.2.Overall Description.....</b>	<b>13</b>
2.2.1.PRODUCT PERSPECTIVE.....	13
2.2.2.USER CLASSES AND CHARACTERISTICS.....	13
2.2.2.1.The System Administrator.....	13
2.2.2.2.Doctors.....	13
2.2.2.3.Patients.....	13
2.2.2.4.Appointment Manager.....	13
2.2.3.OPERATING ENVIRONMENT.....	14
2.2.4.ASSUMPTIONS AND DEPENDENCIES.....	14
2.2.4.1.Communication Protocols.....	14
2.2.4.2.Server/Hardware Performance.....	14
2.2.4.3.Browser Dependency.....	14
<b>    2.3.Functional Requirements.....</b>	<b>14</b>
2.3.1.SMS MODULE.....	14
2.3.1.1.Receiving SMS.....	14
2.3.1.2.Parsing and Processing SMS.....	15
2.3.1.3.Replying SMS.....	15
2.3.2.WEB MODULE.....	15
2.3.2.1.Registration.....	15
2.3.2.2.Login.....	15
2.3.2.3.View Schedule.....	15
2.3.2.4.Update Schedule.....	15
2.3.3.SMS MANAGEMENT TOOL.....	15
2.3.3.1.Search an Appointment.....	15
2.3.3.2.Mark an Appointment Done.....	15
<b>    2.4.Non-functional Requirements.....</b>	<b>16</b>
2.4.1.PERFORMANCE REQUIREMENTS.....	16
2.4.2.SECURITY REQUIREMENTS.....	17
2.4.3.SOFTWARE QUALITY ATTRIBUTES.....	17
<b>The Design.....</b>	<b>18</b>

<b>3.1.Software Design.....</b>	<b>19</b>
3.1.1.DATABASE MANAGEMENT.....	19
3.1.2.SMS PROCESSING.....	20
3.1.3.APPOINTMENT MANAGEMENT TOOL.....	21
<b>3.2.Database Design.....</b>	<b>22</b>
3.2.1.DATA DICTIONARY.....	22
3.2.1.1.Register.....	22
3.2.1.2.Schedule.....	22
3.2.1.3.Appointment.....	22
3.2.2.ENTITY RELATIONSHIP DIAGRAM.....	22
<b>3.3.Design Tools.....</b>	<b>24</b>
<b>Implementation and Testing.....</b>	<b>25</b>
<b>4.1.The Technology.....</b>	<b>26</b>
4.1.1.SMS SERVER.....	26
4.1.1.1.Attention(AT) Command[1].....	26
4.1.1.2.Java: Communications API.....	26
4.1.2.WEB SERVER.....	26
4.1.2.1.Apache.....	26
4.1.2.2.PHP.....	26
4.1.3.DATABASE.....	26
4.1.4.WEB USER INTERFACE.....	27
4.1.4.1.JavaScript: JQuery.....	27
4.1.4.2.Cascading Style Sheet (CSS).....	27
4.1.4.3.Hyper Text Markup Language (HTML).....	27
4.1.5.APPOINTMENT MANAGEMENT TOOL.....	27
<b>4.2.Design Patterns.....</b>	<b>27</b>
4.2.1.STRATEGY.....	27
4.2.2.NULL OBJECT.....	28
4.2.3.SINGLETON.....	28
<b>4.3.Implementation Tools.....</b>	<b>28</b>
4.3.1.ECLIPSE IDE FOR JAVA.....	28
4.3.2.ECLIPSE IDE FOR WEB DEVELOPMENT.....	29
4.3.3.XAMPP FOR MySQL.....	29
<b>4.4.Testing.....</b>	<b>29</b>
<b>User Manual.....</b>	<b>30</b>
<b>5.1.SMS Module.....</b>	<b>31</b>
5.1.1.SMS SERVER.....	31
5.1.1.1.Server Errors.....	31
5.1.2,SMS CLIENT.....	32
<b>5.2.Web Module.....</b>	<b>33</b>
5.2.1.HOME PAGE.....	33
5.2.2.FIND A DOCTOR BY NAME.....	34
5.2.3.PATIENT SERVICES.....	35
5.2.3.1.Find Doctor.....	35
5.2.3.2.View Free Slots.....	36
5.2.3.3.Make Appointment.....	37

5.2.4.THE DOCTORS' PORTAL .....	39
5.2.4.1.Registered Doctors.....	40
5.2.4.2.Enrol New Doctor.....	40
5.2.4.3.Doctor's Login.....	42
5.2.5.DOCCTOR'S PROFILE.....	43
5.2.5.1.View Schedule.....	43
5.2.5.2.Add Schedule.....	43
5.2.5.3.View Appointment.....	44
5.2.5.4.Log Out.....	45
5.2.6.DEVELOPERS INFORMATION.....	45
5.2.7.PROCESS OF APPOINTMENT THROUGH SMS.....	46
<b>Conclusion.....</b>	<b>47</b>
<b>6.1.The Obstacles.....</b>	<b>48</b>
<b>6.2.The Achievements.....</b>	<b>48</b>
<b>6.3.Limitations and Future Works.....</b>	<b>48</b>
<b>6.4.The Failures.....</b>	<b>48</b>
<b>6.5.Last few words.....</b>	<b>49</b>
<b>Appendix A:References.....</b>	<b>50</b>
<b>Appendix B:Abbreviations and Acronyms.....</b>	<b>51</b>

## List of Figures

<b>Figure 2.4.1: Use Case.....</b>	<b>16</b>
<b>Figure 3.1.1: Class Diagram of SMS Server Database Management Classes.....</b>	<b>19</b>
<b>Figure 3.1.2: Class Diagram for SMS processing classes.....</b>	<b>20</b>
<b>Figure 3.1.3: Class Diagram of Appointment Management Tool.....</b>	<b>21</b>
<b>Figure 3.2.1: ER Diagram.....</b>	<b>24</b>
<b>Figure 5.1.1: Could Not Find Port Error.....</b>	<b>31</b>
<b>Figure 5.1.2: Port In Use Error.....</b>	<b>31</b>
<b>Figure 5.1.3: Port Not Found Error.....</b>	<b>32</b>
<b>Figure 5.1.4: Could Not Connect Error.....</b>	<b>32</b>
<b>Figure 5.1.5: Appointment by SMS.....</b>	<b>32</b>
<b>Figure 5.2.1: Home Page.....</b>	<b>33</b>
<b>Figure 5.2.2: Find doctor by name step 1.....</b>	<b>34</b>
<b>Figure 5.2.3: Find doctor by name step 2.....</b>	<b>34</b>
<b>Figure 5.2.4: Patient Service.....</b>	<b>35</b>
<b>Figure 5.2.5: Find A Doctor Step 1.....</b>	<b>35</b>
<b>Figure 5.2.6: Find A Doctor Step 2.....</b>	<b>36</b>
<b>Figure 5.2.7: View Free Slots Step 1.....</b>	<b>36</b>
<b>Figure 5.2.8: View Free Slots Step 2.....</b>	<b>37</b>

<b>Figure 5.2.9: Making Appointment Step 1.....</b>	<b>37</b>
<b>Figure 5.2.10: Making Appointment Step 2.....</b>	<b>38</b>
<b>Figure 5.2.11: Appointment Making Error.....</b>	<b>38</b>
<b>Figure 5.2.12: Successful Appointment.....</b>	<b>39</b>
<b>Figure 5.2.13: Doctors' Portal.....</b>	<b>39</b>
<b>Figure 5.2.14: Registered Doctors.....</b>	<b>40</b>
<b>Figure 5.2.15: Admin Varification.....</b>	<b>40</b>
<b>Figure 5.2.16: Admin Varification Failed.....</b>	<b>41</b>
<b>Figure 5.2.17: Doctor Entry Form.....</b>	<b>41</b>
<b>Figure 5.2.18: Successful Enrollment.....</b>	<b>41</b>
<b>Figure 5.2.19: Doctor's LoginPage.....</b>	<b>42</b>
<b>Figure 5.2.20: Successful Login.....</b>	<b>42</b>
<b>Figure 5.2.21: Doctor's Profile.....</b>	<b>43</b>
<b>Figure 5.2.22: View Schedule.....</b>	<b>43</b>
<b>Figure 5.2.23: Add Schedule.....</b>	<b>44</b>
<b>Figure 5.2.24: View Schedule After Schedule Added.....</b>	<b>44</b>
<b>Figure 5.2.25: View Appointment Step 1.....</b>	<b>44</b>
<b>Figure 5.2.26: View Appointment Step 2.....</b>	<b>45</b>
<b>Figure 5.2.27: Log Out.....</b>	<b>45</b>
<b>Figure 5.2.28: Developers' Info.....</b>	<b>46</b>
<b>Figure 5.2.29: Appointment From Mobile.....</b>	<b>46</b>

## List of Table

<b>Table 1.3.1: Proposed Project Time Line.....</b>	<b>10</b>
<b>Table 3.1.1: Database Management Classes.....</b>	<b>19</b>
<b>Table 3.1.2: SMS Processing Classes.....</b>	<b>20</b>
<b>Table 3.1.3: Appointment Management Tool Classes.....</b>	<b>21</b>
<b>Table 3.2.1: Data Dictionary .....</b>	<b>23</b>

---

## Abstract

This document is aimed to cover the complete description of the software Integrated Appointment Management. The software will provide a way for any mobile phone user to make an appointment to a doctor of a hospital registered to the system. The developers believe that this software can play a significant role in the field of medical appointment and SMS technology.

---

## Acknowledgement

We are grateful to the project supervisor Dr. Kazi Muheymin-Us-Sakib for his supervision throughout the project time. Dr. Md. Shariful Islam is also acknowledged for his kind advises even without being part of the team.

A of the team of BIT batch two are working with the same project in a limited domain. Thanks to the member of that team: Md. Mostafizur Rahman, Md. Habibur Rahman and Iftekhar Ahmed. Without the knowledge they shared with us, the development phase of the software could be much more difficult to us.

# Chapter 1

## Project Plan and Feasibility Study

This chapter covers the project proposal and the feasibility of the proposal along with background study and some preliminary idea about the project.

## 1.1. Background

Medical appointment management has been a traditional old school system relies on pen and paper process. In some hospitals phone calls are also introduced for making appointment to a doctor. But what will happen if there are only 2-3 phone number available for appointment and 10-15 patients are trying to make phone calls at a time? Most of them will find the number busy and may not be able to make appointment at last. In order to reduce the sufferings of the patients, a system needs to be drawn where an automated scheme will replace the existing conventional procedure.

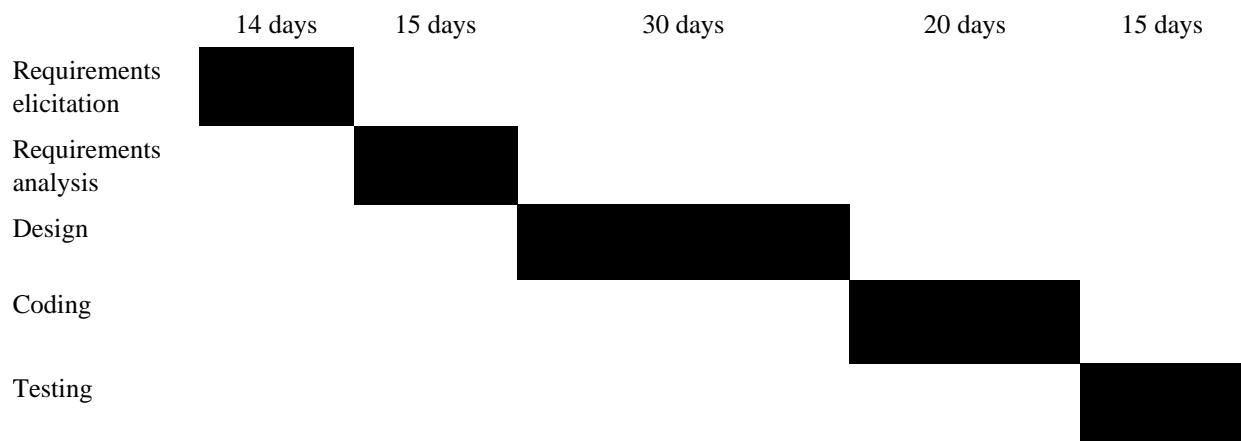
## 1.2. The Project Idea

Automation of appointment management system is the main concern of our project. Focusing on the current practice, we decided to construct an application that demands no labor and minimum time from patients within lowest cost. In the circumstances above, our project team decided to develop an SMS based appointment management system. It facilitates the patients to confirm their appointment with the doctor. A phone number will be issued for the patients for this purpose. They just need to sent a text message containing doctor's ID, appointment date, time and name to that number. An automatic notification will be sent to them. If the appointment is successful then they will be provided by a serial no. which is the verification ID. The system will also notify him if there is no free slot available.

An integrated web solution will also be provided by our team for similar purpose. The web application will allow the patient to find a doctor by their specialized category or name. The patients can formulate appointment by the doctor's registration number mentioned in a chart on that page. On the success it returns serial no. and time to the patient.

## 1.3. Proposed Time line

Project State	Start Time	End Time
Requirements elicitation	20/03/11	02/04/11
Requirements analysis	03/04/11	17/04/11
Design	18/04/11	17/05/11
Coding	18/05/11	06/06/11
Testing	07/06/11	21/06/11



*Table 1.3.1: Proposed Project Time Line*

## 1.4. Feasibility Study

Use of SMS technology is growing faster than ever. Now a days, common people are habituated with this technology. Even in the rural areas, many people are used to with SMS services. Therefore, the service we are trying to provide can reach every corner of the country.

SMS is cheap. Maximum cost of an SMS is set to taka 2.00 per SMS<sup>[8]</sup> in Bangladesh. So, there is no way that we can say the service will be costly.

The reply SMS will be sent immediately after the request SMS is received. Considering the mobile phone operators are providing service properly, the patient should receive reply SMS within 5 minutes of the request is sent.

The Internet service is not very good in Bangladesh. However, it is available in most of the portion of the country. Specially, mobile Internet is available wherever mobile phone service is available.

The popularity of the service highly depend on the promotional activity. If the hospital availing the service promote the service properly, the service is expected to be popular.

The service can be made extremely useful with support of a mobile phone operator. If the operator consider the service as their value added service, there is no doubt that it will be popular.

The server of the service has no special hardware requirement. Any modern personal computer can be used as the server machine.

## Chapter 2

# Software Requirement Specification

## 2.1. Introduction

In the Software Requirement Specification, the overall needs of the proposed software will be defined. This document will help the developers by setting up specific goals to develop the software.

### 2.1.1. Purpose

The purpose of the Software Requirements Specification document is to provide a detailed analysis of all the requirements to design and implement the full system. This document discusses the human and system interfaces and behaviors as it relates to using the system. It includes the use-case diagrams as the supporting information.

### 2.1.2. Project Scope

The software will provide the following features through different media.

- A patient can make appointment to a doctor of the hospital availing our service. The appointment can be made via either SMS or the Internet.
- A user can see full schedule of a particular doctor in a particular date.
- A user can see available schedule of a particular doctor in a particular date to make an appointment to the doctor.
- The users can view detail information about the doctors.
- The administrator can update database through a web interface.
- The doctors can register to the server with administrators approval.
- The doctors can update their own schedules.

## **2.2. Overall Description**

### **2.2.1. Product Perspective**

Several appointment systems are currently observed in different hospitals. In public hospitals and health care centers usually there is no advance appointment system. That means, one cannot make an appointment for some other day except the current one. The popular system in public sectors is that a token and a serial number is given to the patients and they have to wait until their turn comes.

In some hospitals, appointment can be made with phone call. The phone number is stored in the hospitals appointment list and the patients must prove their phone numbers to visit the doctor.

In traditional system a patient must go to a doctor's chamber and get an appointment with the doctor.

No automated system is available anywhere in Bangladesh.

Our system is intended to meet the need of automation in the patient appointment system. It will replace the current manual skim of appointment and introduce a complete automated system. It will be a completely new and self-contained product.

### **2.2.2. User Classes and Characteristics**

#### **2.2.2.1. The System Administrator**

The system administrator has the authority to make a doctor registered to the system. This person will be provided by a unique user-name and a password to administer the system.

#### **2.2.2.2. Doctors**

The doctors will be able to login with their user-name and password via the web interface. After logging in, they can insert and update their schedules.

#### **2.2.2.3. Patients**

The patients will be able to make appointment with the doctor using the web interface or by sending a fixed formatted SMS.

#### **2.2.2.4. Appointment Manager**

This user is the hospital staff who will be able to manage the appointments using a desktop interface, which will be connected to the server.

### **2.2.3. Operating Environment**

The system is entirely based on server-client communication. Therefore there are two major components: the server and the client.

The SMS server will process client SMS with AT command which is a Microsoft Windows technology. Therefore, the SMS server must run in a Windows environment. Windows XP is preferred.

The preferred database is MySQL and web server should be in PHP. Neither depend on operating environment, any common environment will work.

The SMS client needs a mobile phone that can send SMS.

The web client is a JavaScript enabled web browser.

### **2.2.4. Assumptions and Dependencies**

#### **2.2.4.1. Communication Protocols**

The SMS module will be developed depending on the standard protocols followed in SMS communications. The web module will be on TCP/IP protocol.

#### **2.2.4.2. Server/Hardware Performance**

The project team is not responsible for providing servers or hardware. However, the performance of a software sometimes depends on the hardware or server machine in which it is running.

#### **2.2.4.3. Browser Dependency**

The web module of the software will be compatible with any JavaScript enabled open standard browsers, and it will also support Internet Explorer (IE) compatible browsers.

## **2.3. Functional Requirements**

Functional requirements of a system are the requirements those defines the system. This section describes the functional requirements of the system.

### **2.3.1. SMS Module**

#### **2.3.1.1. Receiving SMS**

The request from the clients will be in form of SMS. So, the server must be able to receive SMS.

### **2.3.1.2. Parsing and Processing SMS**

The received SMS must be parsed by the server to decrypt the request. Then the request should be processed and database should be updated if necessary.

### **2.3.1.3. Replying SMS**

The module must be able to reply to a message. This reply message will contain the information about a the appointment the patient has made or an failure message if appointment is not possible or error message if the request message has syntactical error.

## **2.3.2. Web Module**

### **2.3.2.1. Registration**

The doctors cannot register themselves to the system, however, the system administrator can register a doctor in the system

### **2.3.2.2. Login**

The system administrator and the doctors will be able to login the system.

### **2.3.2.3. View Schedule**

Doctors will be able to view their own schedule if logged in.

### **2.3.2.4. Update Schedule**

Doctors will be able to update their own schedule if logged in.

## **2.3.3. SMS Management Tool**

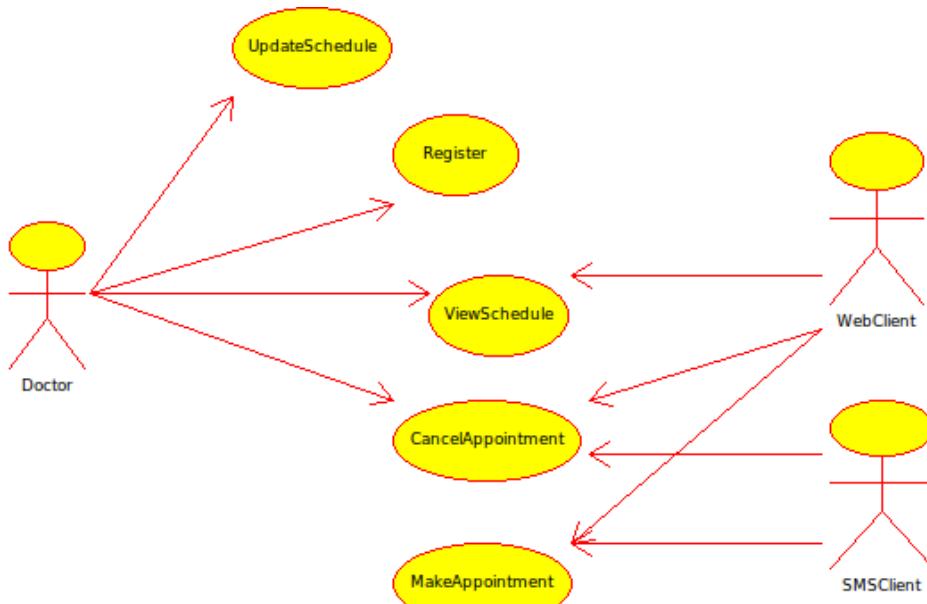
A management tool will be provided to the hospital to manage the appointments. The tool will provide the following features.

### **2.3.3.1. Search an Appointment**

The tool will provide a GUI to search an appointment with doctor's ID and serial number. The result will be a detailed information about the appointment on success, or a failure message if the appointment is not valid.

### **2.3.3.2. Mark an Appointment Done**

This tool will provide a way to mark an appointment visited.



*Figure 2.4.1: Use Case*

## 2.4. Non-functional Requirements

### 2.4.1. Performance Requirements

- Server software does not require any special hardware other than the minimum hardware required for running enterprise OS, PHP 5, MySQL 5.5.8 database, Apache 2.2.17.
- The SMS server must have Java 6 installed.
- Extra disk storage will be required for archives and electronic documents. Increases of memory enables efficient query processing.
- Processors with clock speed 3.0 Ghz, at least 4GB RAM and 300 GB hard disk is recommended for the server.
- The web client will need a personal computer with any common operating system.
- The SMS client will require a mobile phone that can send SMS.

### **2.4.2. Security Requirements**

The system may require digital certificate for some restricted access such as updating schedule, canceling appointment, administrative activities, verification of user accounts etc. Viewing schedule does not require digital certificates. For all private access, a user must be authenticated by the system using secure channel.

### **2.4.3. Software Quality Attributes**

- The system should have capabilities to adapt new releases of Operating System, database and server side development tools.
- The service should be available all the time, aside from the failure of network or communication media.
- System should respond to the user with the correct result in a correct way. If security policy depicted that a result should go through a secured channel it should go through a secured channel.
- All communications such http(s) must be abide by ISO Z39.5 information exchange standard.
- Well documentation and standard annotation will make the software easily maintainable and manageable.
- The software will be a implementation of a collection of standard OOP design which will be reusable to add new components to the software.
- The art of design and implementation should make the software easily testable at component level and as a whole.
- It should be user friendly and will not take too much crawling to reach the targeted user interface.

## Chapter 3

### The Design

This chapter discusses the design of the software in non-language-specific diagrams. These diagrams are known as Unified Modeling Language (UML).

The design includes the software design and the database design.

## 3.1. Software Design

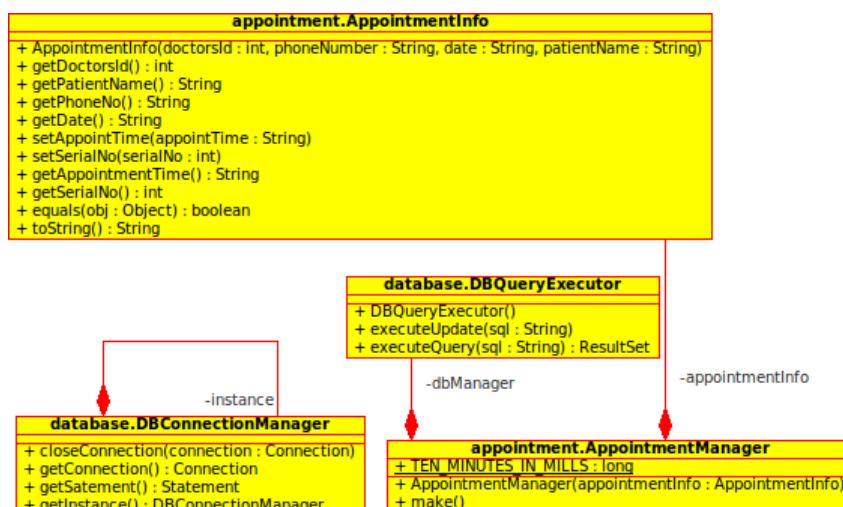
The designs will be described using class diagrams. The implementation technology is expected to be an Object Oriented Programming(OOP) language and this document will result a speed up in the development face.

Each of the modules may have several responsibilities. These responsibilities will be described separately with class description and class diagram. Also, the way in which one uses another will also be covered.

### 3.1.1. Database Management

Classes	Responsibilities
DBConnectionManager	Establishes connection between the database and the program.
DBQueryExecutor	Executes the queries to the database.
AppointmentManager	Processes an appointment request and provides a possible appointment time.
AppointmentInfo	Encapsulates the data of an appointment.

*Table 3.1.1: Database Management Classes*



*Figure 3.1.1: Class Diagram of SMS Server Database Management Classes  
Page 19 of 51*

### 3.1.2. SMS Processing

Classes	Responsibilities
PortConnectionManager	Establishes connection between the port (in which the modem is connected) and the program.
SMSManager	Receives, processes and replies messages received from the client.
SMSParser	SMSManager uses this class to process SMS.
Sender	SMSManager uses this class to reply SMS to the clients.
Receiver	SMSManager uses this class to receive SMS from the clients.

Table 3.1.2: SMS Processing Classes

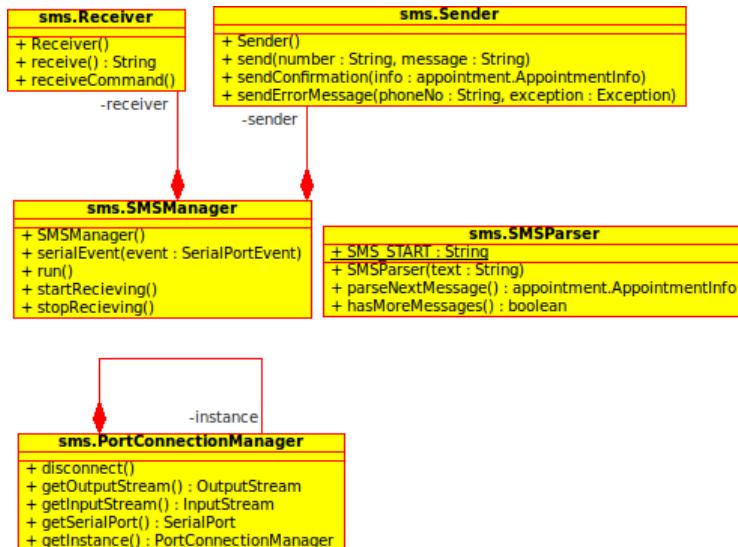


Figure 3.1.2: Class Diagram for SMS processing classes

### 3.1.3. Appointment Management Tool

Classes	Responsibility
ControlWindow	Manages the window of the tool.
ControlPanel	Manages all UI to be displayed to the
Display	Super class for searching and displaying search result.
SearchPanel	Takes input from the user to search an appointment.
ResultPanel	Shows the search result.
LabelTextPair	Encapsulates a label and an input field together to be used in Display.

Table 3.1.3: Appointment Management Tool Classes

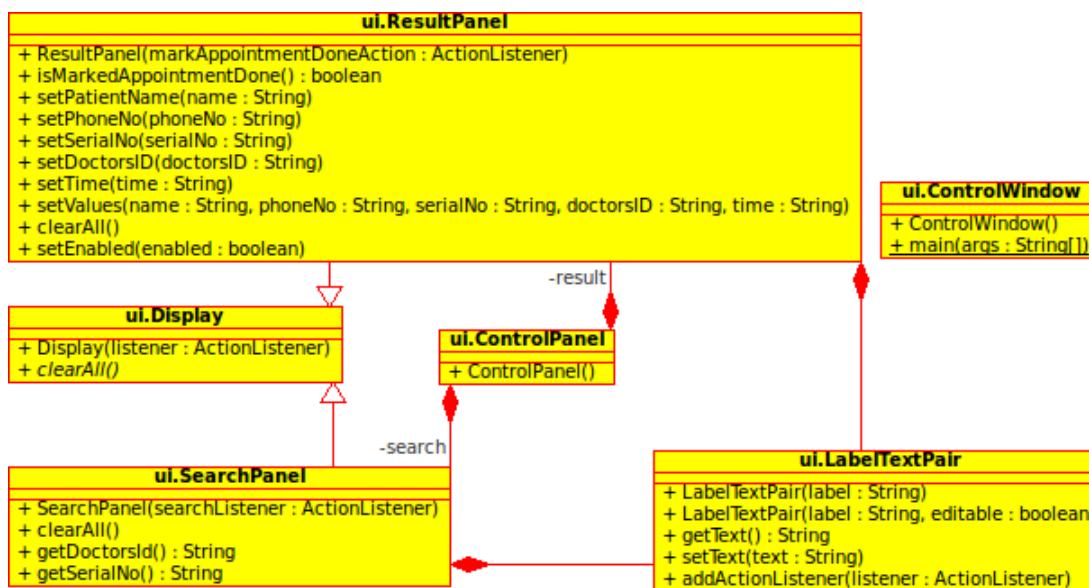


Figure 3.1.3: Class Diagram of Appointment Management Tool

## 3.2. Database Design

Most softwares needs a database to manage the data and our software is no different. The database design will be described by Data dictionary and Entity Relationship (ER) Diagrams.

### 3.2.1. Data Dictionary

The data dictionary describes the database with its entities and the data types of the attributes of the entities. The database for IAM needs three entities. Register, Schedule and Appointment.

#### 3.2.1.1. Register

The register entity is responsible for storing the information about a registered doctor. The primary attribute of this entity is the doctors identification number (doctors\_id). The other attributes of this entity are the name, login name (user\_name), password and specialization field of the doctor (category).

#### 3.2.1.2. Schedule

Each entry of this entity encapsulates a schedule of a doctor. A schedule is uniquely defined with combination of doctor's ID, date and start time() of the schedule. The doctor's id is the foreign key to the register entity.

#### 3.2.1.3. Appointment

This entity is stores the data about an appointment. An appointment can be identified by a doctor's ID, a date and a serial number or the patient's phone number. Like in the schedule entity, here also the doctor's ID is the foreign key to the Register entity.

### 3.2.2. Entity Relationship Diagram

The ER diagram shows the relationship between the entities in the database system with some universal symbols.

Entity Name	Attribute Name	Data type	Size	Null	Primary Key	Description
Register	category	Text	20	no	no	Specialization field of the doctor
	doctors_id	Integer	10	no	yes	The government provided identification number of the doctor
	name	Text	50	no	no	Name of the doctor
	password	Text	20	no	no	System login password of the doctor
	user_name	Text	20	no	no	Login name of the doctor
Schedule	doctors_id	Integer	10	no	yes	Foreign key to Register entity
	date	Date	-	no	yes	Date of the schedule
	start_time	Time	-	no	yes	Start time of the schedule
	end_time	Time	-	no	no	End time of the schedule
	is_packed	Boolean	-	no	no	Specifies whether the schedule is filled up with appointments. If the value is true, no more appointments can be made in this schedule.
Appointment	doctors_id	Integer	10	no	yes	Foreign key to Register entity
	date	Date	-	no	yes	Date of the appointment
	serial_number	Integer	4	no	yes	Serial number of the appointment
	time	Time	-	no	no	Time at which the appointment is expected to be done
	patient_phone	Text	14	yes	no	Phone number of the patient. This attribute is used to check if the correct patient is visiting the doctor
	patient_name	Text	50	yes	no	An optional field. The patients may or may not specify their names. Phone number is the identification of a patient because it is unique.
	booked	Boolean	-	no	no	If an appointment is booked, no other appointments can be made at that time.
	visited	Boolean	-	no	no	This field specifies whether or not an appointment is done. That means this value is true if the patient has visited the doctor.

Table 3.2.1: Data Dictionary

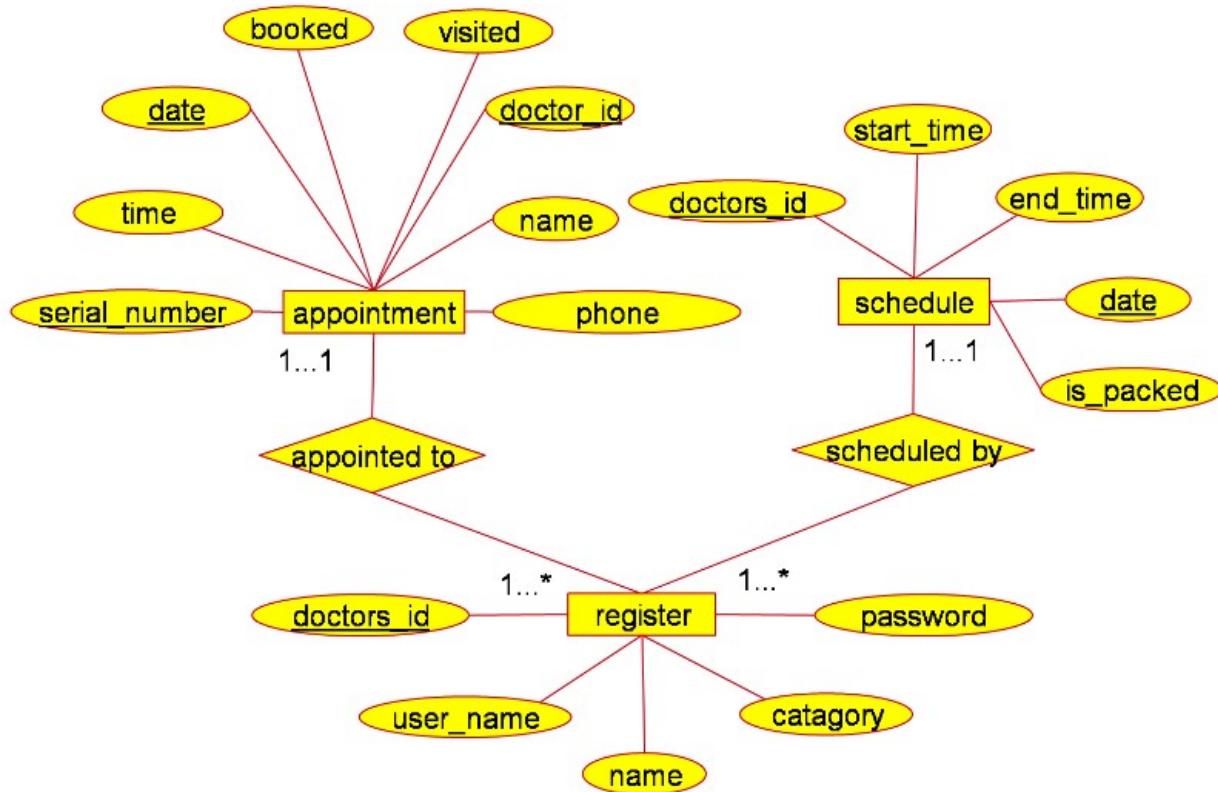


Figure 3.2.1: ER Diagram

### 3.3. Design Tools

We selected Umbrello UML Modeler as our design tool. Umbrello UML Modeler is a Unified Modeling Language diagram editor for KDE. It can create diagrams of software and other systems in the industry-standard UML format, and can also generate code from UML diagrams in a variety of programming languages.

There are several reason why we selected Umbrello.

- It is free and open source.
- The tool is available in the repositories of most Linux operating system. That means its very easy to find.
- The symbols used in this tool are the standard symbols understood by most developers and designers.

## Chapter 4

### Implementation and Testing

## 4.1. The Technology

Development technologies are growing everyday with the same speed as the requirement is growing. We have tried to use the most appropriate technologies to serve different purposes.

### 4.1.1. SMS Server

#### 4.1.1.1. Attention(AT) Command<sup>[1]</sup>

AT command is a technology to manage SMS using a modem from computer. AT commands are executable in windows environment only that is why our SMS server has to be run in a windows machine.

#### 4.1.1.2. Java: Communications API

The SMS server is programmed using Java communications API (commonly known as Java Comm) to execute AT command from the server code. The Comm API is a Java extension that facilitates developing platform-independent communications applications for various technologies.<sup>[2]</sup> We have used it to access GSM/GPRS modem or phone connected by a serial port.

Our system used Java-6 and Java comm 3.0.

### 4.1.2. Web Server

#### 4.1.2.1. Apache

We have used apache for our web server. The version is 2.2.17.

#### 4.1.2.2. PHP

PHP was our server side programming language for the web server. PHP provides very flexible and easy ways to program for server. Our preferred version was PHP-5.3.4. Unfortunately, we could not use Object-Oriented version of PHP because we have not learned that yet. Our target for the next version is to convert the PHP codes to Object-Oriented PHP.

### 4.1.3. Database

The most common database MySQL has been used for our system's data management. Our preferred version was MySQL 5.5.8.

#### **4.1.4. Web User Interface**

##### **4.1.4.1. JavaScript: JQuery**

JavaScript is claimed to be most powerful technology to develop user interface for web sites. We have also chosen this technology to develop the interfaces for our web clients.

JQuery is a JavaScript API that provides flexible way to create Graphical User Interface for web pages.<sup>[3]</sup> We have used JQuery version 1.5 in JavaScript version 1.8.0.

##### **4.1.4.2. Cascading Style Sheet (CSS)**

We have used CSS to design our pages with custom settings instead of default HTML look and feel.

##### **4.1.4.3. Hyper Text Markup Language (HTML)**

Every web pages have HTML in the bottom level of user interface. We have HTML-4 in our web pages.

#### **4.1.5. Appointment Management Tool**

The appointment management tool is built on Java swing API. The version is the latest stable one: Java 6.

### **4.2. Design Patterns**

We have tried to follow the design patterns in the development whenever applicable. Design patterns help a software implementation to be easily understandable to other developers. Also, they increase the standard of the software in case of security, adaptability and expandability.

All design patterns listed here are defined according to [4].

#### **4.2.1. Strategy**

Strategy pattern is one of the most common design patterns used by the developers. This pattern helps the software in case of expandability. With a good use of it, we can easily add new features to the software without any major change in the existing code.

The pattern is used in the SMS client in the exception handling classes.

### 4.2.2. Null Object

Null object design pattern is used to avoid null reference passing in the program. Null references can create null references error in the program because of invalid user input. Null Object pattern passes default object known as *null object* to avoid this error.

ApplicableInfo class uses this pattern.

### 4.2.3. Singleton

Some classes in the systems should not have more than one instances. Singleton pattern is used to manage these classes.

A *singleton* class usually have a private contractor and a global instance created within that class. All other classes will use this global object instance of creating new instances.

In our program, PortConnectionManager and DBConnectionManager classes are singletons.

## 4.3. Implementation Tools

Software requirements are being more challenging everyday. Therefore the life of the developers are also facing more challenges. The challenges can be won much easily by choosing the right development tool for the each part of the work. We have tried to utilize maximum of the benefits from these tools.

### 4.3.1. Eclipse IDE for Java

Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system.<sup>[5]</sup> We have chosen eclipse because:

- It is lightweight with respect to its ability.
- Its free and open source.
- It supports a variety of plugins. We could rely the fact that we will get whatever we need.
- Its suggestions and auto complete features are excellent.
- The user interface is simple and very friendly.
- We are used to work in eclipse platform and we never felt like eclipse is not enough to feed our hunger.

### **4.3.2. Eclipse IDE for Web development**

For the web module, we needed to work in PHP, JavaScript and HTML. In these cases also, we relied on Eclipse. Eclipse has a PHP version which has HTML and JavaScript editors integrated within it. Eclipse for PHP also has a web browser to test the development.

The reasons behind choosing Eclipse for PHP are the same as those are for Java.

### **4.3.3. XAMPP for MySQL**

For database management, we have used XAMPP. XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages. [6]

## **4.4. Testing**

We tried to do the modern approach of testing, the Test Driven Development (TDD)<sup>[7]</sup>. Unfortunately, we found ourselves not to be skilled enough to implement such a high level of development idea.

We went through informal testing throughout the development phase. However, there is no formal testing evidence we can deliver or document. This implies that the current version is released without any formal testing.

# Chapter 5

## User Manual

## 5.1. SMS Module

### 5.1.1. SMS Server

The server has no interface to be handled. It will do its work in an automated process.

#### 5.1.1.1. Server Errors

Whenever an unexpected error occurs it will show an error message and the server will be closed to avoid failure in any other processes running in the machine. Each message will indicate a particular error. The error messages are described with screen shots.

- The server is going to be started but the GSM/GPRS modem is not connected to the server or the modem is connected to a port other than the default port.



Figure 5.1.1: Could Not Find Port Error

- The server is already running and someone is trying to run the server again or any other device is connected to the default port.

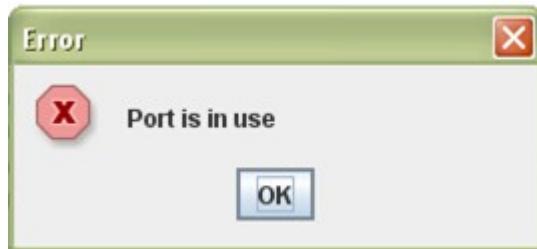
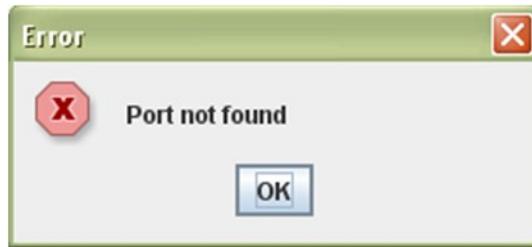


Figure 5.1.2: Port In Use Error

- The server cannot find the default port in the system it shows an error message.



*Figure 5.1.3: Port Not Found Error*

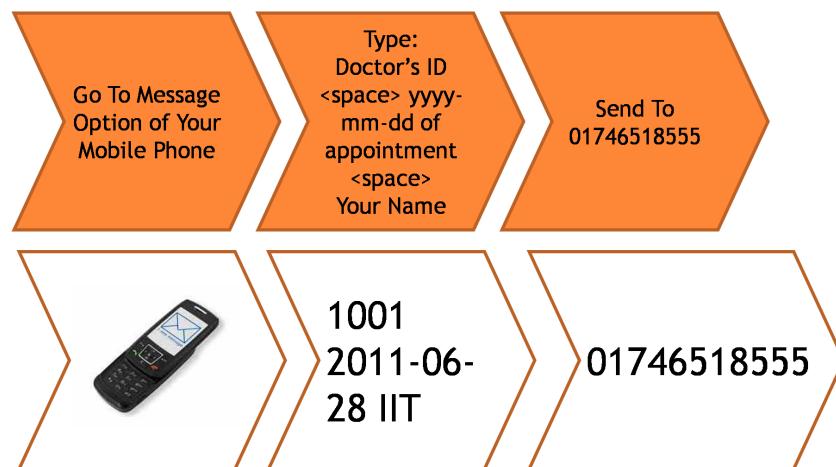
- If the server cannot write or read the port it will get an input-output error.



*Figure 5.1.4: Could Not Connect Error*

### 5.1.2. SMS Client

The patient will have to request an appointment by maintaining a specific format. The format is described with a picture.



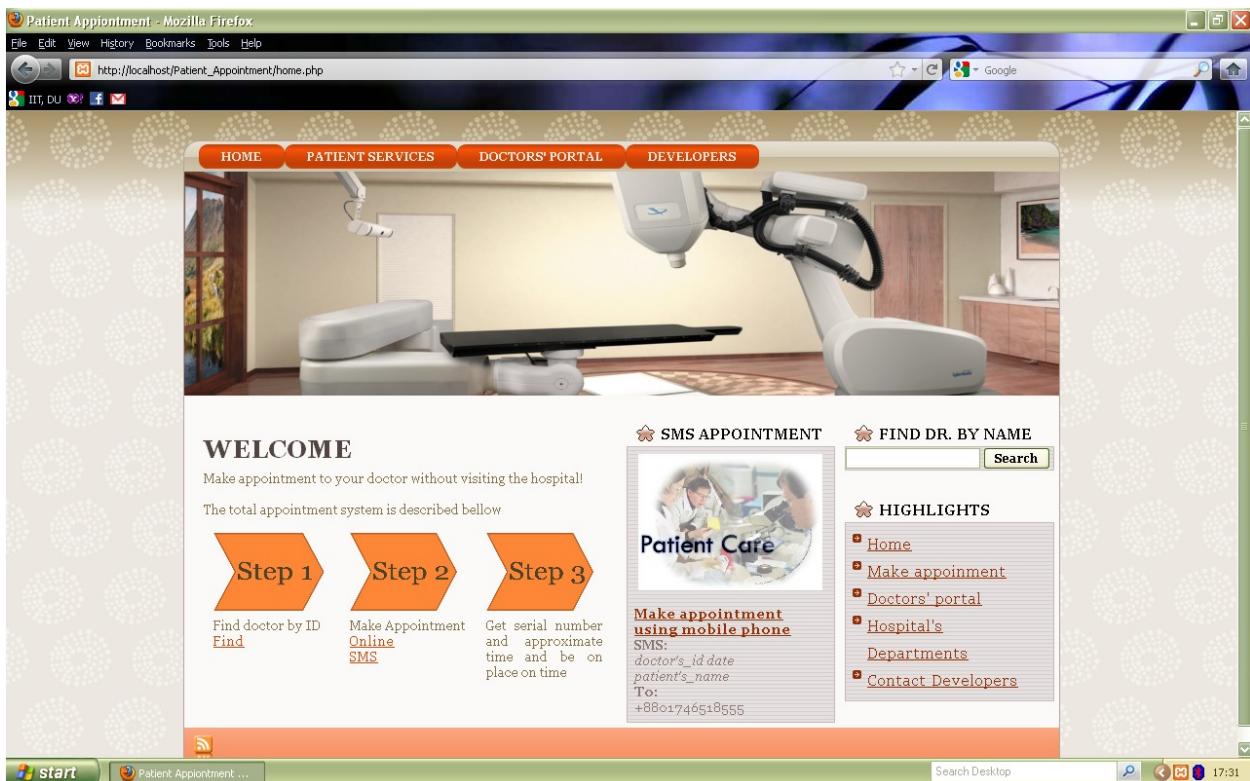
*Figure 5.1.5: Appointment by SMS*

## 5.2. Web Module

The web pages provide facilities to the patients to make appointment online. The doctors can also maintain and view their schedule from web pages.

### 5.2.1. Home page

The home page provides basic information of the total appointment system. It also provides a quick view of the facilities provided by the system from the highlights.



*Figure 5.2.1: Home Page*

The home page will take a patient to a number of other sections with other facilities. The sections are:

- Patient services
- Doctors' portal
- Developers

The services of these sections are described below.

### 5.2.2. Find A Doctor by Name

A patient can find a doctor by name. The search option is at the right side of the home page.



Figure 5.2.2: Find doctor by name step 1

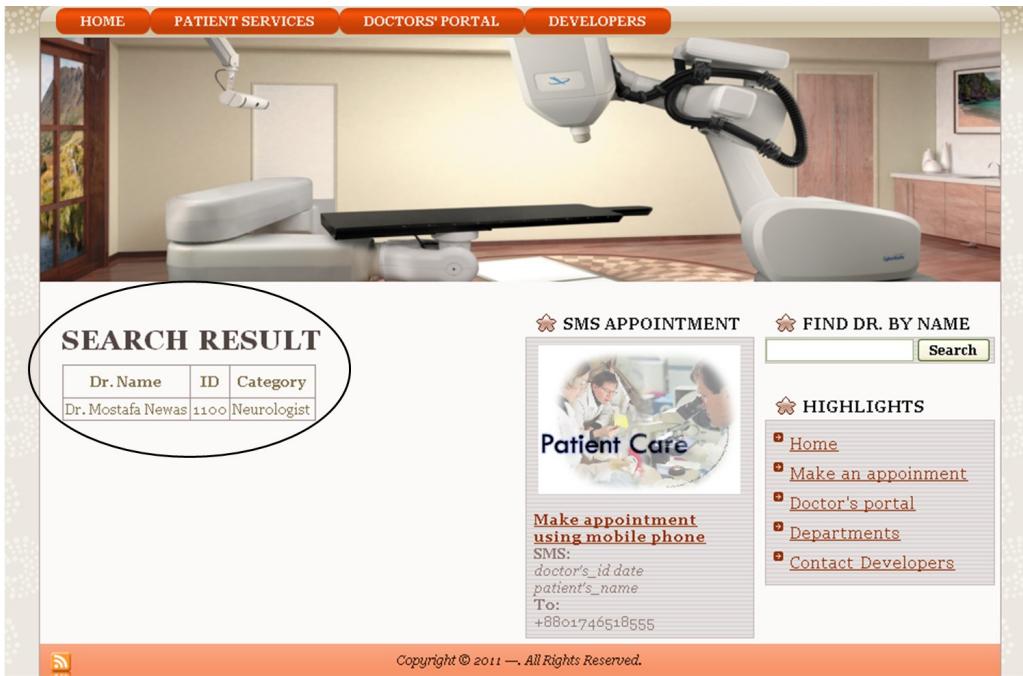


Figure 5.2.3: Find doctor by name step 2

### 5.2.3. Patient Services

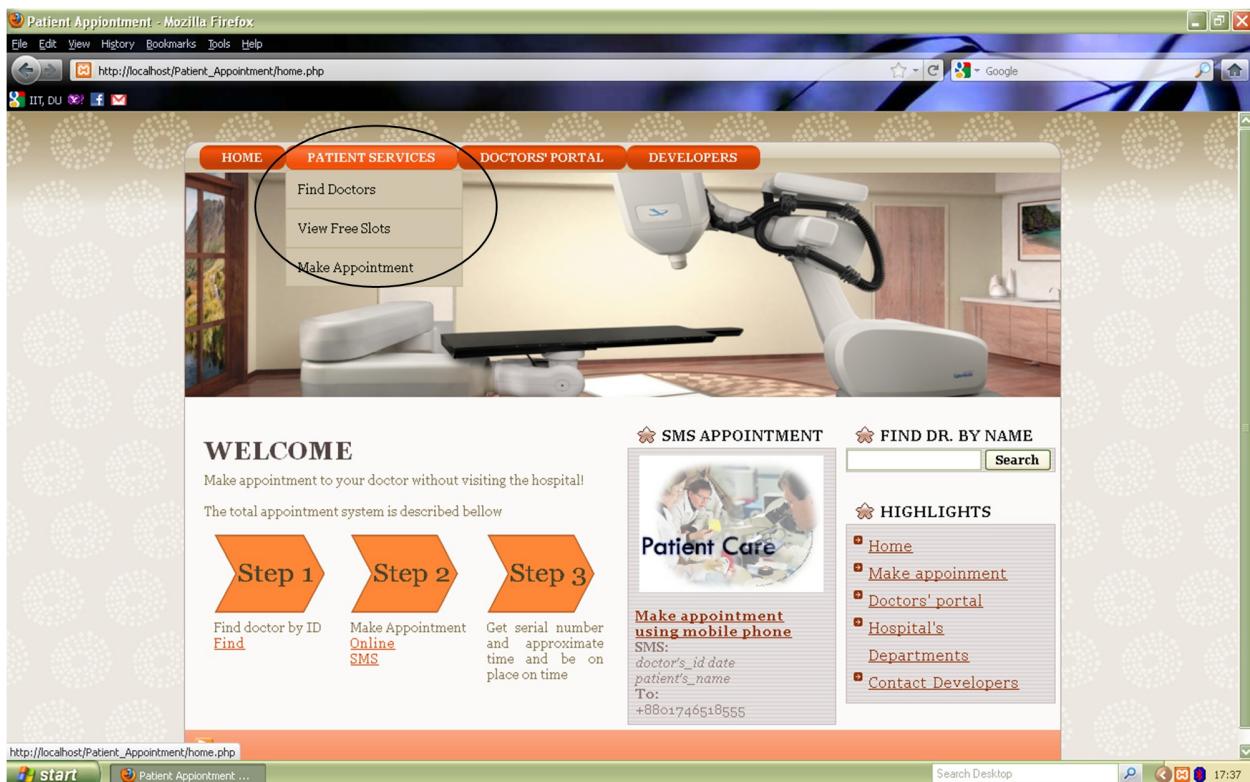


Figure 5.2.4: Patient Service

#### 5.2.3.1. Find Doctor

A patient can find a doctor to make appointment by the department of the doctor from the menu item.

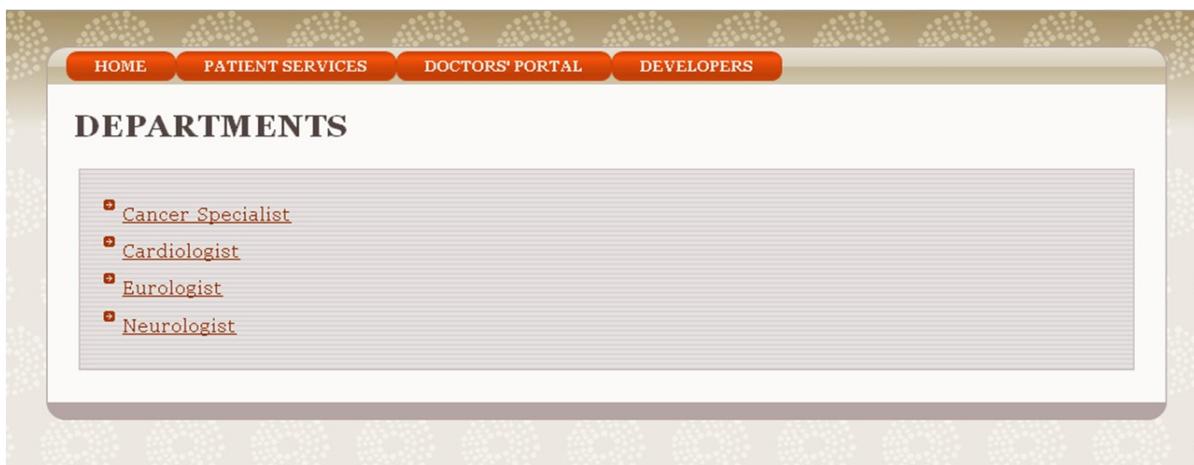


Figure 5.2.5: Find A Doctor Step 1



**CANCER SPECIALISTS**

Please enter desired Doctor's ID to get appointment

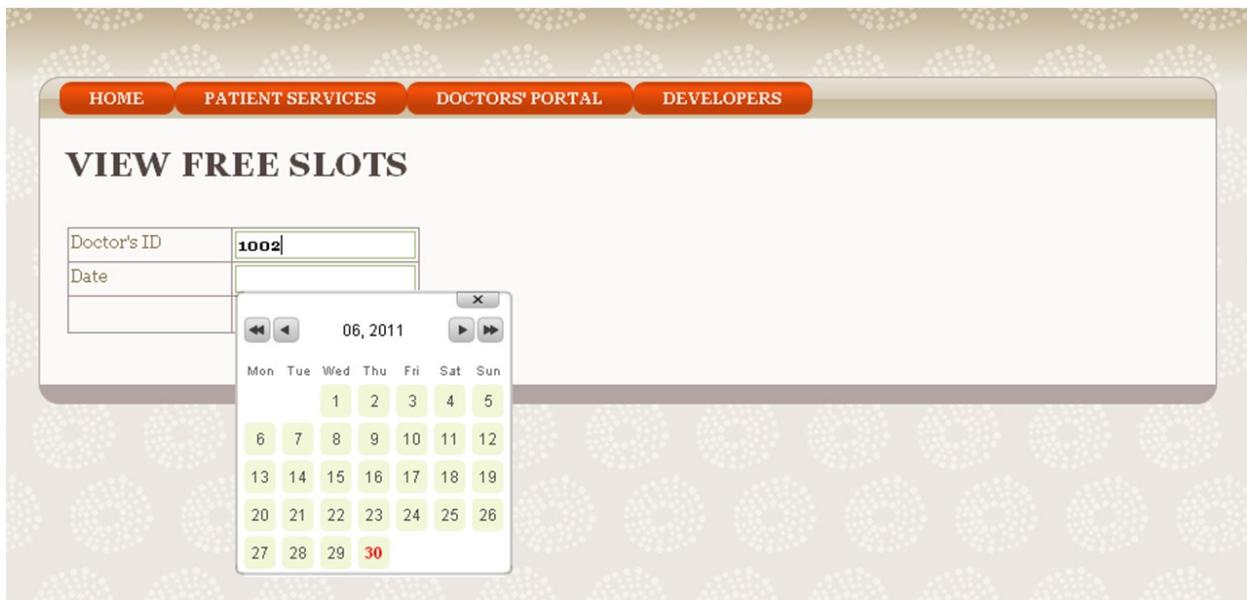
1002	Submit
------	--------

Doctor's name	ID
Dr. Abul Kashem	1002
Dr. Al Amin	1111

Figure 5.2.6: Find A Doctor Step 2

### 5.2.3.2. View Free Slots

A patient may want to know a doctor's free slots on a particular day to take appointment. This can be known from the menu item under the "patient services" menu.



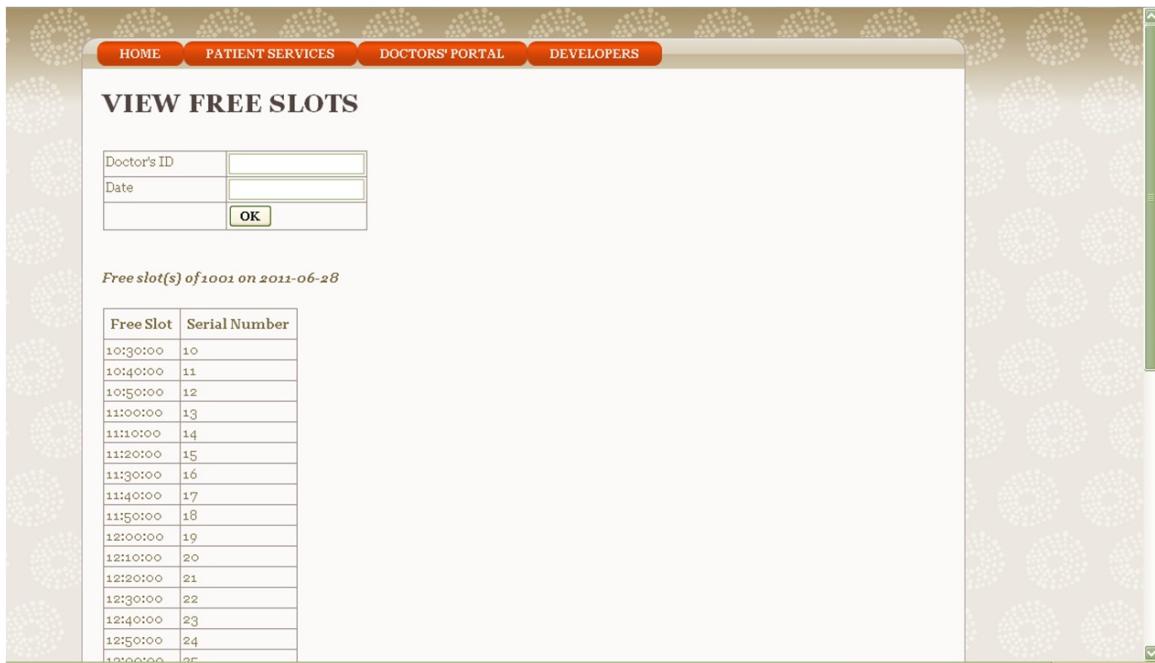
**VIEW FREE SLOTS**

Doctor's ID	1002
Date	06, 2011

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Figure 5.2.7: View Free Slots Step 1

At first the patient will have to fill a form with the field "Doctor's ID" and the date to see free slots on that day. According to the doctor's ID and date, the system will show the free slots like below:



**VIEW FREE SLOTS**

Doctor's ID	<input type="text"/>
Date	<input type="text"/>
<input type="button" value="OK"/>	

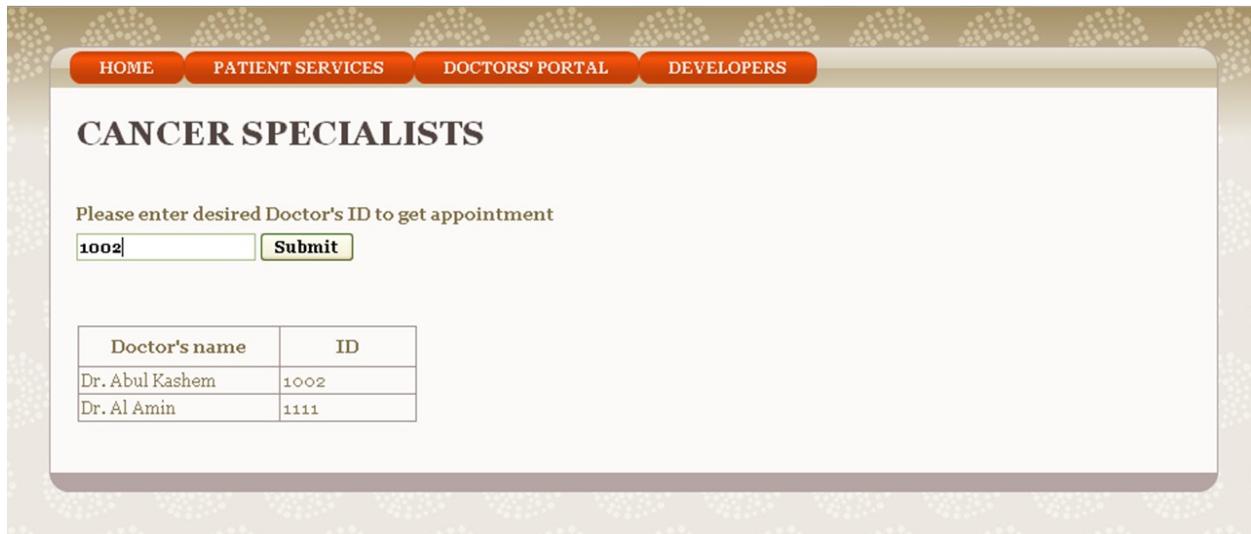
*Free slot(s) of 1001 on 2011-06-28*

Free Slot	Serial Number
10:30:00	10
10:40:00	11
10:50:00	12
11:00:00	13
11:10:00	14
11:20:00	15
11:30:00	16
11:40:00	17
11:50:00	18
12:00:00	19
12:10:00	20
12:20:00	21
12:30:00	22
12:40:00	23
12:50:00	24
12:55:00	25

Figure 5.2.8: View Free Slots Step 2

### 5.2.3.3. Make Appointment

To make an appointment, the patient will have to go through the steps for finding a doctor except the list of particular department's will be with a field to input the expected doctor's ID.



**CANCER SPECIALISTS**

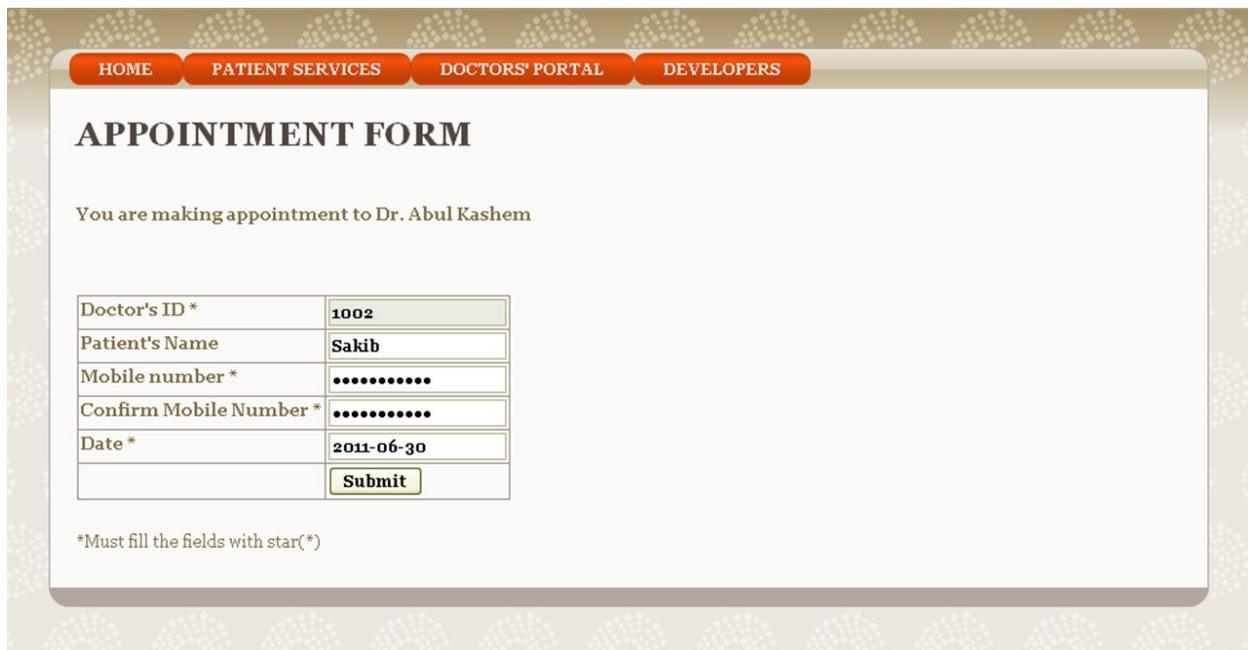
Please enter desired Doctor's ID to get appointment

<input type="text" value="1002"/>	<input type="button" value="Submit"/>
-----------------------------------	---------------------------------------

Doctor's name	ID
Dr. Abul Kashem	1002
Dr. Al Amin	1111

Figure 5.2.9: Making Appointment Step 1

After the patient has submitted the doctor's ID, there will be a form to fill to take appointment.



The screenshot shows a web-based appointment form. At the top, there is a navigation bar with links: HOME, PATIENT SERVICES, DOCTORS' PORTAL, and DEVELOPERS. Below the navigation bar, the title "APPOINTMENT FORM" is displayed. A message below the title states: "You are making appointment to Dr. Abul Kashem". The form contains several input fields:

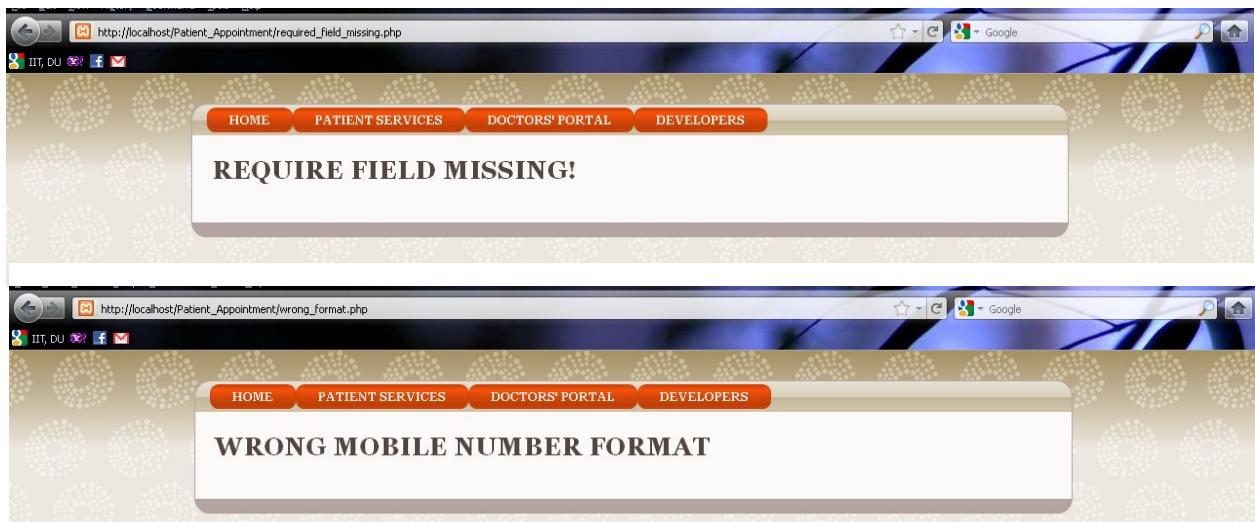
Doctor's ID *	<b>1002</b>
Patient's Name	<b>Sakib</b>
Mobile number *	<b>*****</b>
Confirm Mobile Number *	<b>*****</b>
Date *	<b>2011-06-30</b>
<b>Submit</b>	

A note at the bottom left of the form area says: "\*Must fill the fields with star(\*)".

*Figure 5.2.10: Making Appointment Step 2*

The star (\*) marked filled are mandatory to be filled. Leaving them blank will give an error and no appointment will be made for that patient.

Any error in filling the appointment form will direct the patient to a page with specific error message.



*Figure 5.2.11: Appointment Making Error*

Getting such type of error will prevent patient to take appointment. She/he will have to go to the back page (appointment form) and fill the form correctly.

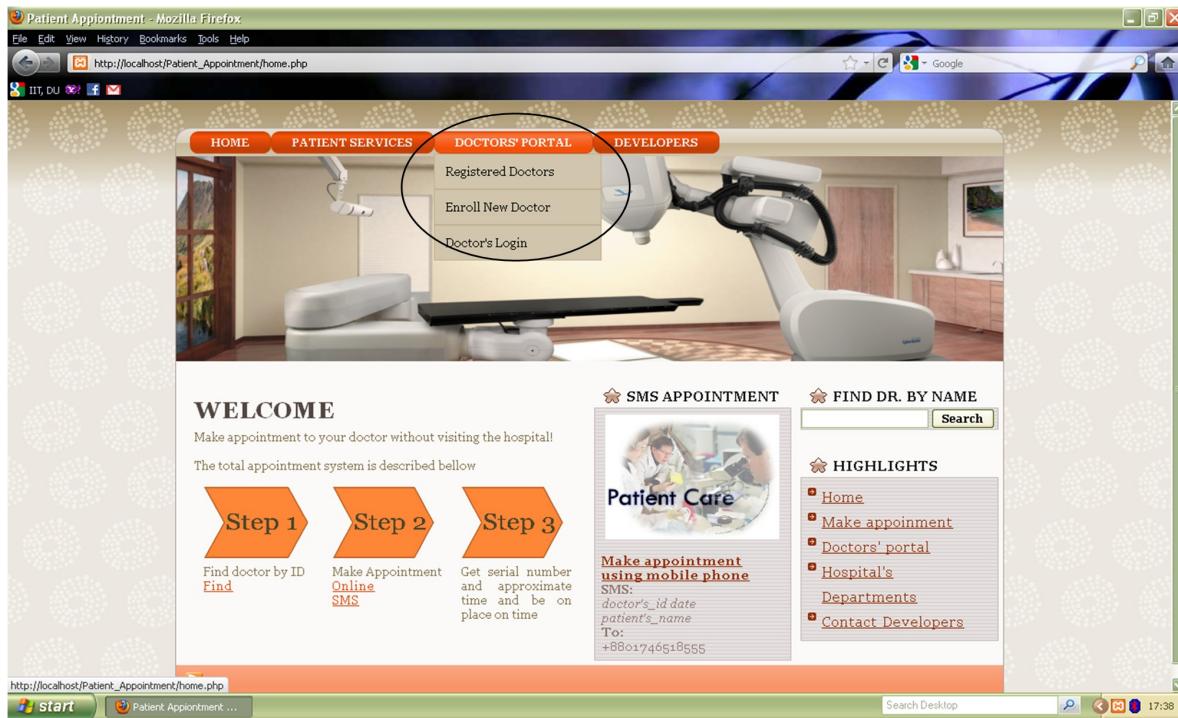
After a successful form fill up the patient will given a serial number and approximate time to visit the doctor in the hospital.



*Figure 5.2.12: Successful Appointment*

#### 5.2.4. The doctors' portal

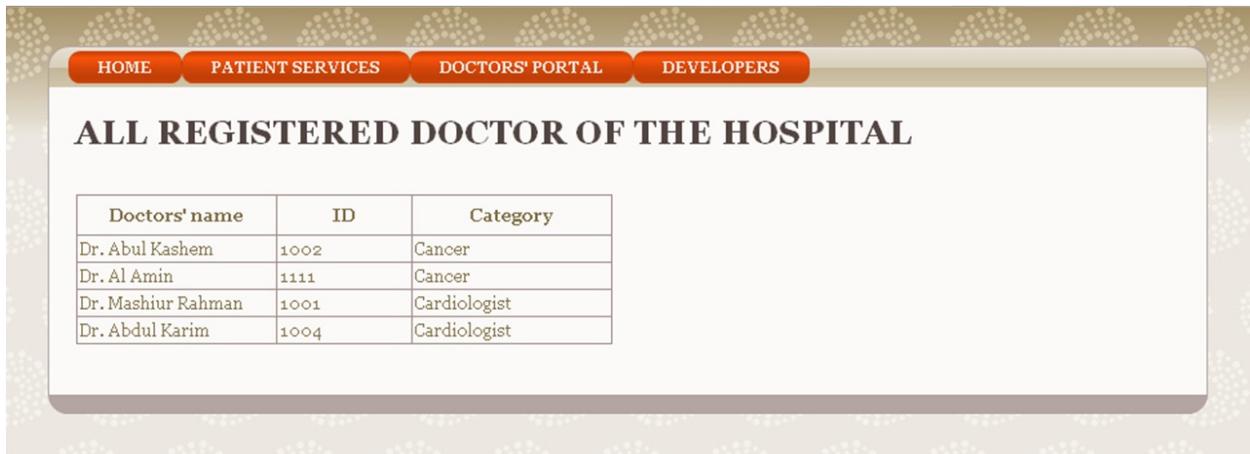
- Contains information of the doctors
- Let administer to enroll new doctors of the hospital and
- Let a doctor to login to his/her account to manage his/her own information



*Figure 5.2.13: Doctors' Portal*

#### **5.2.4.1. Registered Doctors**

Any user of the system can see all the registered doctors of the hospital from the menu item “Registered Doctors”.



The screenshot shows a web page titled "ALL REGISTERED DOCTOR OF THE HOSPITAL". At the top, there is a navigation bar with four tabs: "HOME", "PATIENT SERVICES", "DOCTORS' PORTAL", and "DEVELOPERS". Below the title, there is a table with three columns: "Doctors' name", "ID", and "Category". The table contains five rows of data:

Doctors' name	ID	Category
Dr. Abul Kashem	1002	Cancer
Dr. Al Amin	1111	Cancer
Dr. Mashirul Rahman	1001	Cardiologist
Dr. Abdul Karim	1004	Cardiologist

Figure 5.2.14: Registered Doctors

#### **5.2.4.2. Enrol New Doctor**

Enrolling a new doctor is an administrative work. Only administrators of the system can do this job. Verification for the administrators is done before enrolling a doctor to the hospital.

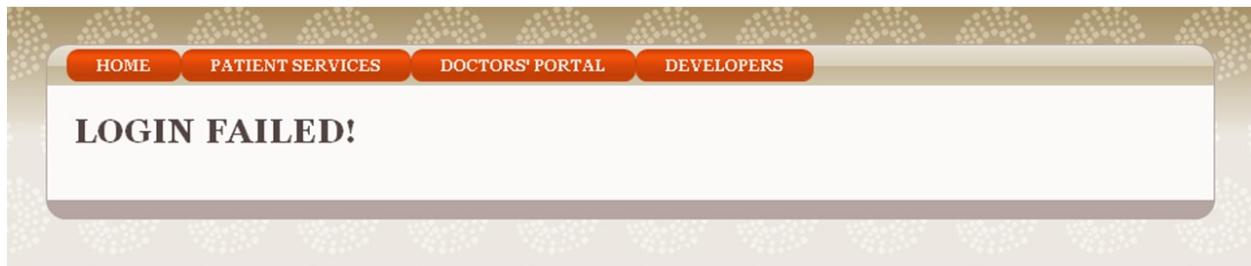


The screenshot shows a web page titled "ADMIN VERYFICATION". At the top, there is a navigation bar with four tabs: "HOME", "PATIENT SERVICES", "DOCTORS' PORTAL", and "DEVELOPERS". Below the title, there is a login form with two fields: "Username" and "Password", and a "Login" button.

Username	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

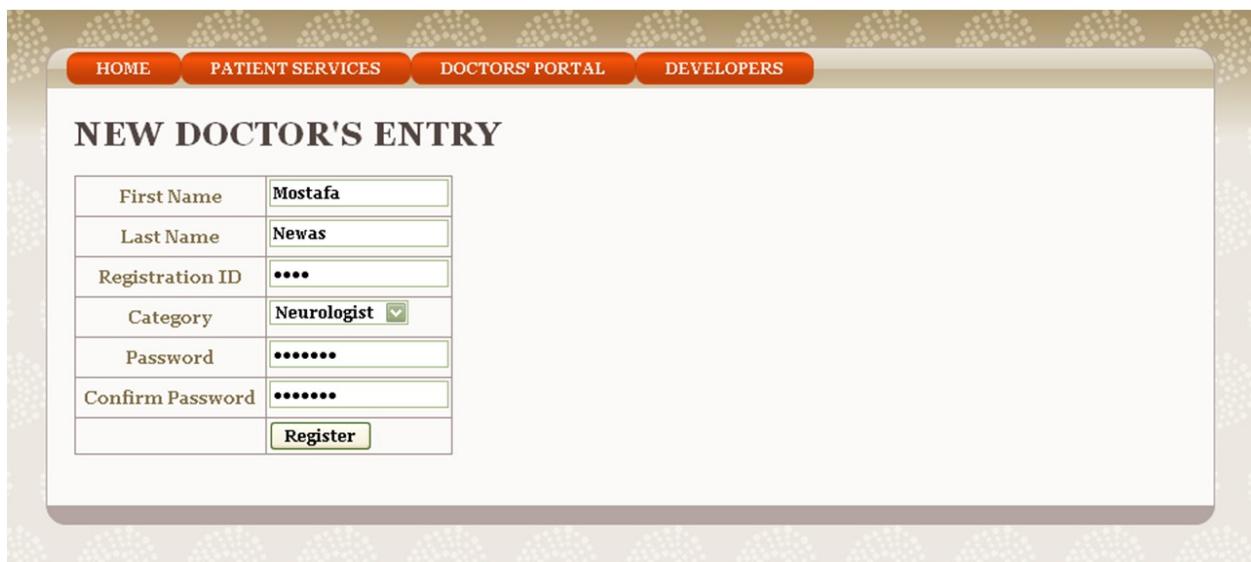
Figure 5.2.15: Admin Varification

Wrong username or password will prevent administrators to perform enrollment.



*Figure 5.2.16: Admin Varification Failed*

After a administrator has been verified as valid, an input form will appear in the page to fill will the information of a new doctor.



First Name	Mostafa
Last Name	Newas
Registration ID	****
Category	Neurologist <input type="button" value="▼"/>
Password	*****
Confirm Password	*****
<input type="button" value="Register"/>	

*Figure 5.2.17: Doctor Entry Form*

Submitting the form with valid information will show a message of success.



*Figure 5.2.18: Successful Enrollment*

#### 5.2.4.3. Doctor's Login

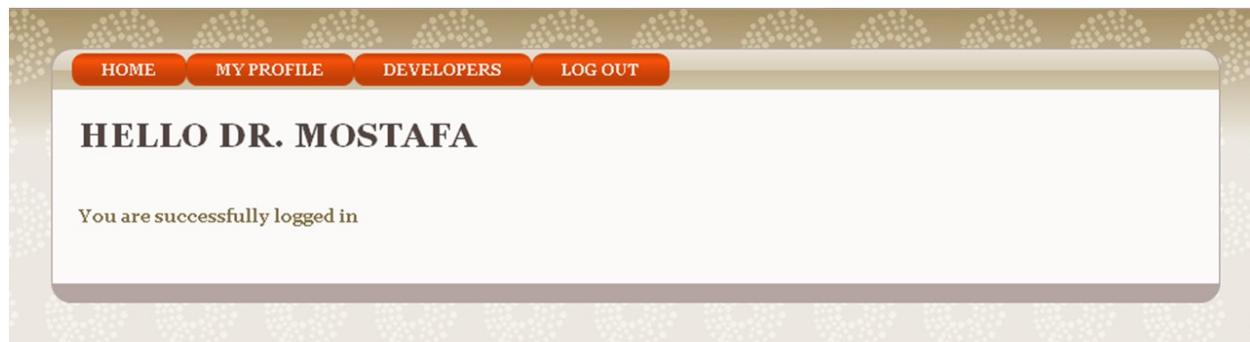
A doctor can log in to her/his account by his ID and password. Wrong username or password will not let to log in.



The image shows the 'DOCTOR'S LOG IN' page. At the top, there is a navigation bar with four buttons: 'HOME', 'PATIENT SERVICES', 'DOCTORS' PORTAL', and 'DEVELOPERS'. Below the navigation bar, the title 'DOCTOR'S LOG IN' is displayed. There are two input fields: 'Registration ID' containing '1100' and 'Password' containing '\*\*\*\*\*'. A green 'Login' button is located below the password field.

Figure 5.2.19: Doctor's LoginPage

As a doctor login to do his/her own maintenance work, the services for the patient are not visible after a doctor has logged in.

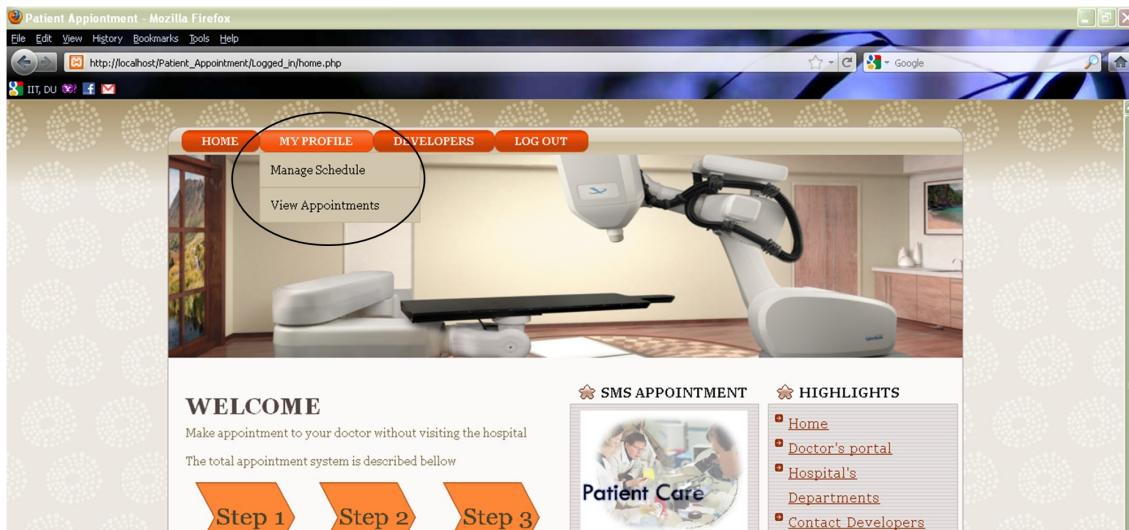


The image shows the 'MY PROFILE' page after successful login. At the top, there is a navigation bar with four buttons: 'HOME', 'MY PROFILE', 'DEVELOPERS', and 'LOG OUT'. The title 'HELLO DR. MOSTAFA' is displayed prominently. Below the title, a message 'You are successfully logged in' is shown.

Figure 5.2.20: Successful Login

### 5.2.5. Doctor's Profile

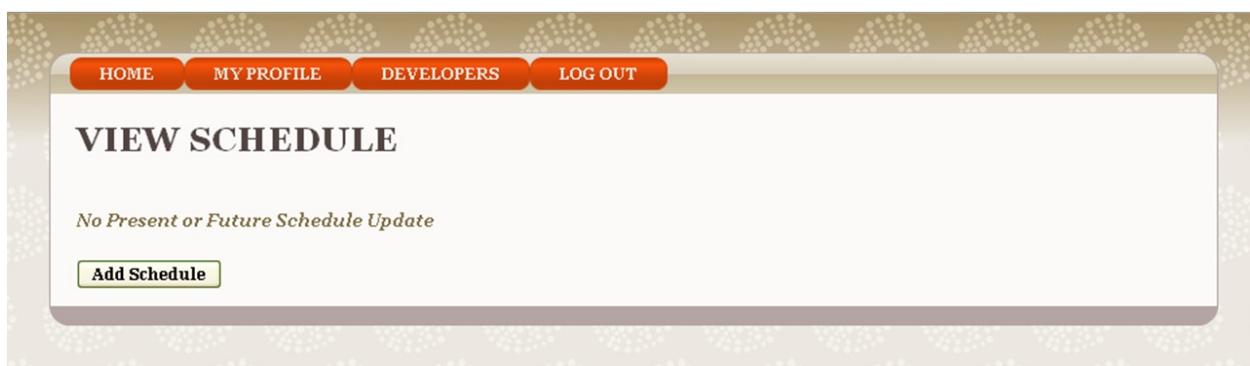
The doctor's profile contains the basic task to be done by the doctor.



*Figure 5.2.21: Doctor's Profile*

#### 5.2.5.1. View Schedule

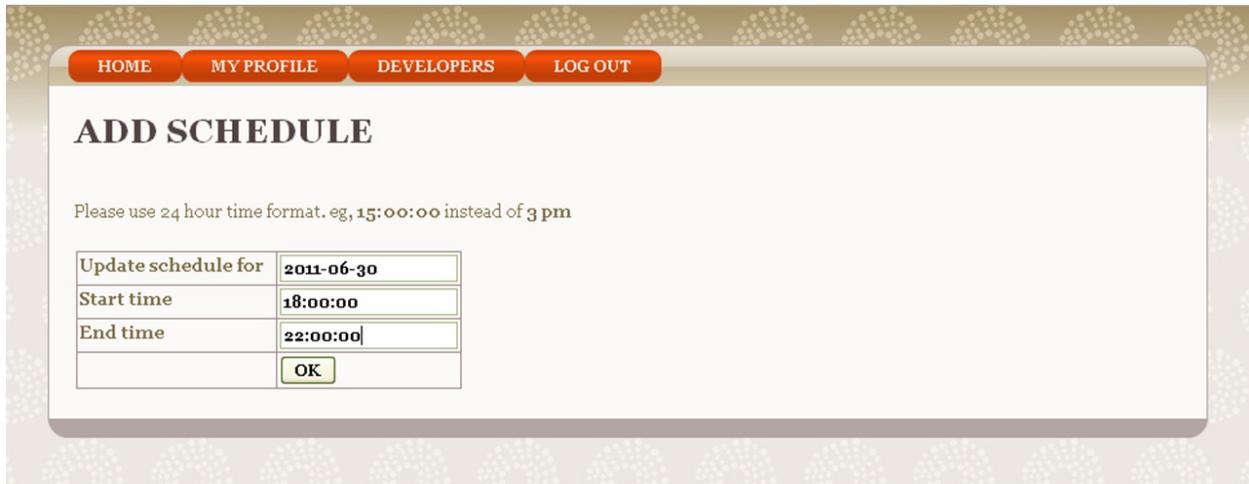
The schedule of the present day and the coming days are shown in this section.



*Figure 5.2.22: View Schedule*

#### 5.2.5.2. Add Schedule

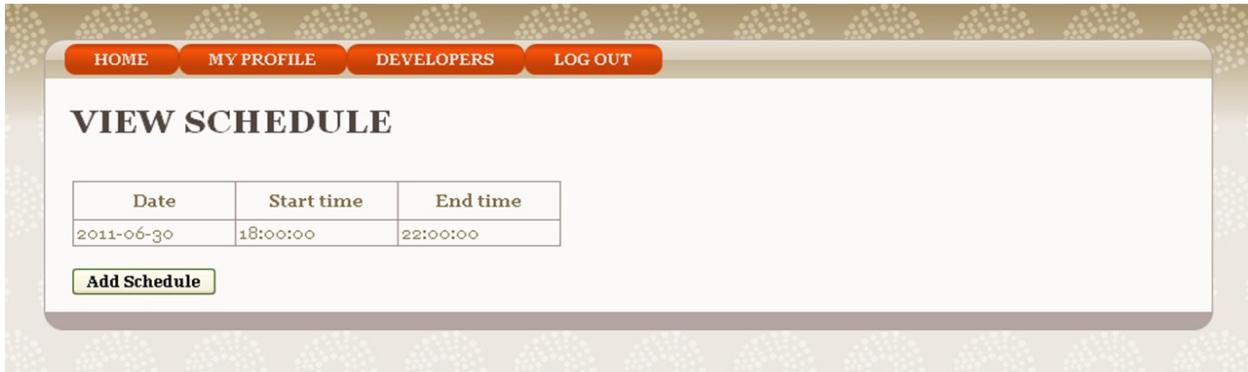
From the “View Schedule” page, clicking the “Add Schedule” button, a doctor can add new schedule for a specific date. She/he will set the start time and the end time to see patients on that day.



The screenshot shows a web-based appointment management system. At the top, there is a navigation bar with links for HOME, MY PROFILE, DEVELOPERS, and LOG OUT. Below the navigation bar, the title "ADD SCHEUDLE" is displayed in bold capital letters. A note below the title says "Please use 24 hour time format. eg, 15:00:00 instead of 3 pm". Below this note is a form with four rows. The first row has a label "Update schedule for" followed by a text input field containing "2011-06-30". The second row has a label "Start time" followed by a text input field containing "18:00:00". The third row has a label "End time" followed by a text input field containing "22:00:00". The fourth row contains a green "OK" button.

*Figure 5.2.23: Add Schedule*

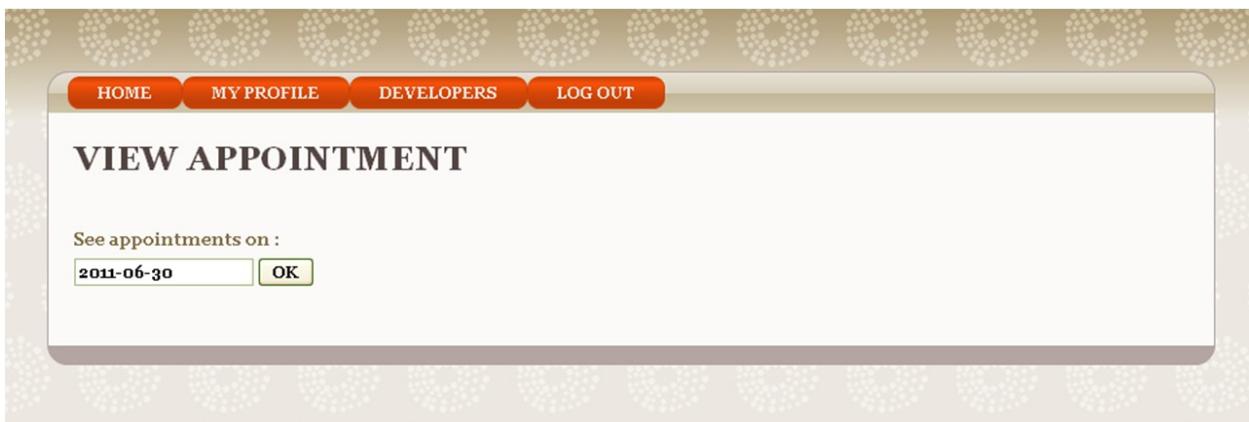
After adding the schedule system will direct the doctor to the page of “View Schedule” with the updated schedule.



The screenshot shows the "VIEW SCHEUDLE" page. The navigation bar at the top includes links for HOME, MY PROFILE, DEVELOPERS, and LOG OUT. The main content area displays a table with three columns: Date, Start time, and End time. The table shows the following data: Date is 2011-06-30, Start time is 18:00:00, and End time is 22:00:00. Below the table is a green "Add Schedule" button.

*Figure 5.2.24: View Schedule After Schedule Added*

### 5.2.5.3. View Appointment



The screenshot shows the "VIEW APPOINTMENT" page. The navigation bar at the top includes links for HOME, MY PROFILE, DEVELOPERS, and LOG OUT. The main content area has a heading "VIEW APPOINTMENT". Below it is a note "See appointments on :" followed by a text input field containing "2011-06-30" and a green "OK" button.

*Figure 5.2.25: View Appointment Step 1*

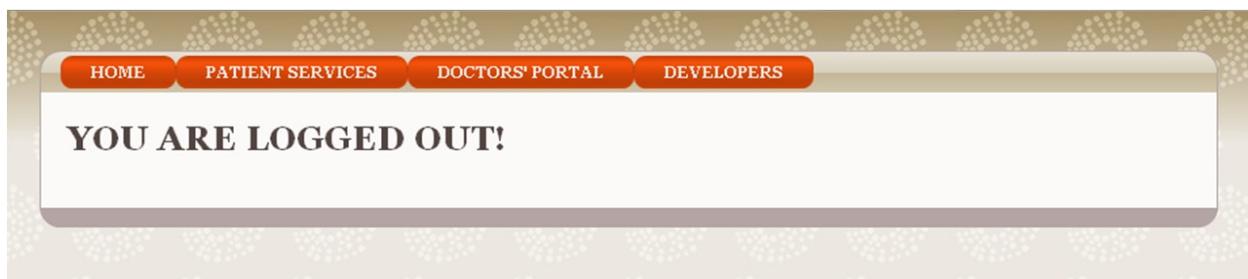


Patient's Name	Mobile Number	Appointment Time	Serial Number
Mohaimin	+8801913387553	18:00:00	1
Tajkia	+8801921192514	18:10:00	2
Linkon	+8801917079684	18:20:00	3

*Figure 5.2.26: View Appointment Step 2*

#### 5.2.5.4. Log Out

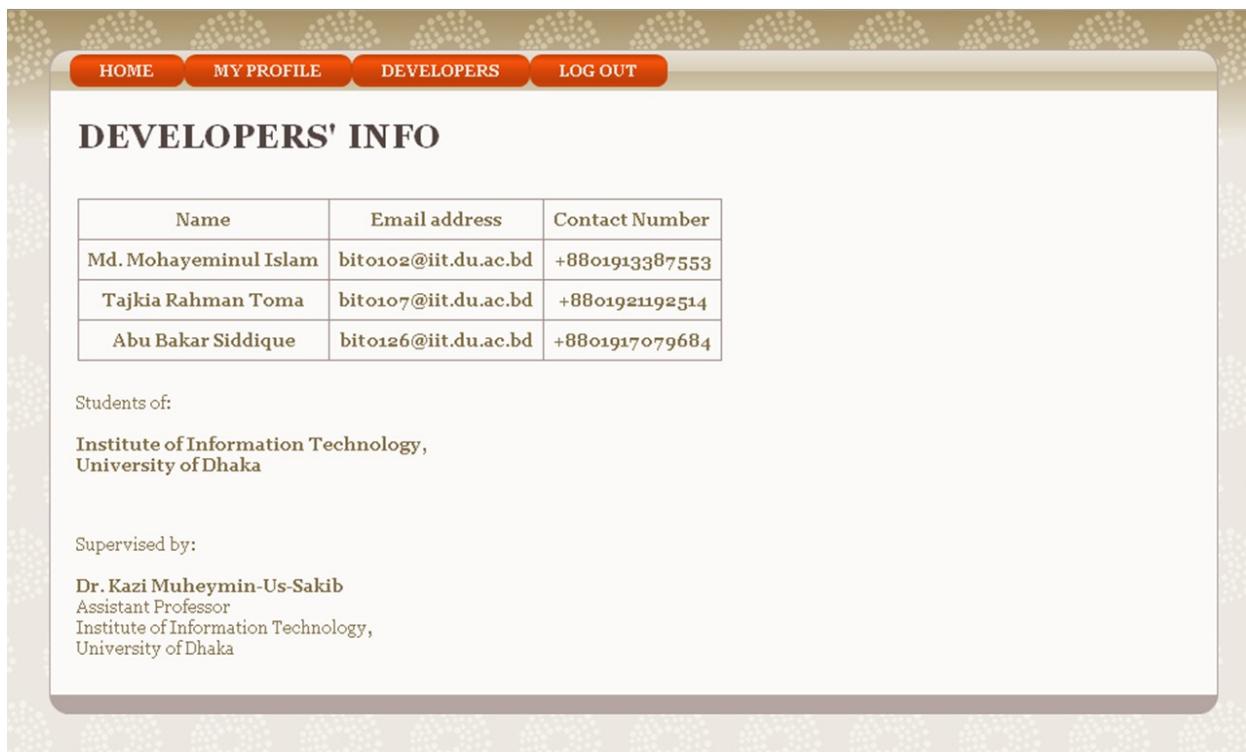
After the doctor has done his management tasks, he can now safely logout from the system from the menu “Log out”



*Figure 5.2.27: Log Out*

#### 5.2.6. Developers Information

The system contains information of the developers with their contact information for any need in the system.



The screenshot shows a web page titled "DEVELOPERS' INFO". At the top, there is a navigation bar with four buttons: "HOME", "MY PROFILE", "DEVELOPERS", and "LOG OUT". Below the title, there is a table with three columns: "Name", "Email address", and "Contact Number". The table contains four rows of data:

Name	Email address	Contact Number
Md. Mohayeminul Islam	bito102@iit.du.ac.bd	+8801913387553
Tajkia Rahman Toma	bito107@iit.du.ac.bd	+8801921192514
Abu Bakar Siddique	bito126@iit.du.ac.bd	+8801917079684

Below the table, there is a section labeled "Students of:" followed by the text "Institute of Information Technology, University of Dhaka". Further down, there is a section labeled "Supervised by:" followed by the text "Dr. Kazi Muheymin-Us-Sakib, Assistant Professor, Institute of Information Technology, University of Dhaka".

*Figure 5.2.28: Developers' Info*

### 5.2.7. Process of Appointment through SMS

The appointment making system through SMS is also provided in the web version. The link to know the process is provided at the home page.



The screenshot shows a web page titled "APPOINTMENT MAKING PROCEDURE USING MOBILE PHONE". At the top, there is a navigation bar with four buttons: "HOME", "MY PROFILE", "DEVELOPERS", and "LOG OUT". Below the title, there is a diagram illustrating the appointment-making process using a mobile phone. The diagram consists of several orange arrows and boxes:

- An arrow pointing right contains the text "Go To Message Option of Your Mobile Phone".
- An arrow pointing right contains the text "Type: Doctor's ID <space> yyyy-mm-dd of appointment <space> Your Name".
- An arrow pointing right contains the text "Send To 01746518555".
- A separate box contains an image of a mobile phone.
- A separate box contains the text "1001 2011-06-28 IIT".
- A separate box contains the text "01746518555".

*Figure 5.2.29: Appointment From Mobile*  
*Page 46 of 51*

# Chapter 6

## Conclusion

A software project means a lot of experience. This chapter summarizes the experience gained by the project team during development of the IAM.

## 6.1. The Obstacles

The project is a academic project which is intended to have some market values. At the beginning of the project schedule, the team selected another project, gathered some requirements, and started preliminary documentation. However, the project was then rejected because of unmet hardware dependencies.

We had to learn three new technologies: PHP, JavaScript and AT command for development of the software. The team members had very little experience with PHP and JavaScript, no prior experience regarding AT command. Learning new technologies cost a lot of work hour of the project team.

## 6.2. The Achievements

We have successfully learned the new technologies we needed. Specially, the knowledge of AT command is expected to help us in future.

We have gained experience with a new software design tool: Umbrello.

The work was a team work. For every software engineer, being good in team work is not only good, also necessary. The success in distribution of workload and compiling the respective parts together proves that we have learned the habit of team work.

## 6.3. Limitations and Future Works

We are considering to recover all known limitations in next versions. Therefore, future works is pointed along with Limitations.

The login mechanism is unsecured.

Some checking in the web module are done in the server side. However, those checking does not need any database access thus can be done in the client side to optimize server load.

An appointment cannot be canceled by anyone.

## 6.4. The Failures

We consider some of the facts as failure rather than limitations. We are determined to work hard next time so that such failures do not come again.

The requirement engineering was inappropriate. We could not gather enough requirement from reliable sources.

The design was not good enough to be unchanged after development started. There were few changes made in the design after the development was started.

## 6.5. Last few words

The piece of software we developed is intended to serve people by reducing time and cost. The success of this project may come to help to many of the people, specially those who live far away from proper health care. Therefore, we expect that, any one using the software will help us reporting any problems/bugs found in it.

Also, help us improve the software with constructive comments. For any kind of assistance, feel free to contact any of the developers. The contact information will be found in the Appendix.

---

## Appendix A: References

1. AT Command - <http://www.developershome.com/sms/atCommandsIntro.asp>
2. Java Comm - <http://www.oracle.com/technetwork/java/index-jsp-141752.html>
3. JQuery - <http://en.wikipedia.org/wiki/JQuery>
4. Design Pattern - [http://www.mydeveloperconnection.com/html/gof\\_design\\_patterns.htm](http://www.mydeveloperconnection.com/html/gof_design_patterns.htm)
5. Eclipse IDE - [http://en.wikipedia.org/wiki/Eclipse\\_IDE](http://en.wikipedia.org/wiki/Eclipse_IDE)
6. XAMPP - <http://en.wikipedia.org/wiki/Xampp>
7. TDD - [http://en.wikipedia.org/wiki/Test-driven\\_development](http://en.wikipedia.org/wiki/Test-driven_development)
8. BTRC - <http://www.btrc.gov.bd/index.php>

---

## Appendix B: Abbreviations and Acronyms

API	Abstract Programming Interface
AT Command	Attention Command
BTRC	Bangladesh Telecommunication Regulatory Commission
CSS	Cascading Style Sheet
ER	Entity Relationship
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HTML	Hypertext Markup Language
IAM	Integrated Appointment Management
IDE	Integrated Development Environment
ISO	International Organization for Standardization
KDE	Common Desktop Environment
OOP	Object Oriented Programming
OS	Operating System
PHP	PHP Hypertext Preprocessor (Recursive)
RAM	Random Access Memory
SQL	Standard Query Language
SRS	Software Requirement Specification
TDD	Test Driven Development
UI	User Interface
UML	Unified Modeling Language