# Virtual Doctor

**Software Project I**

**Submitted By**

| | |
|---|---|
| **Bhuiyan, Md. Mursalat** | **14-26584-2** |
| **Rahman, Tanvir** | **14-26571-2** |
| **Roy, Arnab** | **14-26473-2** |
| **Shihab, Ahmed** | **14-26284-1** |

**MD. Shiefuzzaman**

Faculty
Department of Computer Science
American International University-Bangladesh

**August 2017**

# Declaration

This is to certify that this project is our original work. No part of this has been submitted elsewhere partially or fully for the award of any other degree. Any material reproduced in this project has been properly acknowledged.

**Bhuiyan, MD. Mursalat**
14-26584-2
CSSE

**Rahman, Tanvir**
14-26571-2
CSSE

**Roy, Arnab**
14-26473-2
CSSE

**Ahmed , Shihab**
14-26284-1
CSSE

# Approval

The Software Project “VIRTUAL DOCTOR” has been submitted to the following respected members of the Board of Examiners of the Faculty of Science and Information Technology in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Software Engineering on 22 August 2017 by the following students and has been accepted satisfactory.

**MD. Shiefuzzaman**
Faculty
Department of Computer Science
American International University-Bangladesh

**Sabbir Ahmed**
Assistant Professor
Department of Computer Science
American International University-Bangladesh

**Dr. Dip Nandi**
Associate Professor & Head (Undergraduate)
Department of Computer Science
American International University-Bangladesh

**Professor Dr. Tafazzal Hossain**
Dean in Charge
Faculty of Science and Information Technology
American International University-Bangladesh

**Dr. Carmen Z. Lamagna**
Vice Chancellor
American International University-Bangladesh

# Acknowledgement

Working on the project “VIRTUAL DOCTOR” was a source of immense knowledge to us. The completion of this undertaking could not have been possible without the participation and assistance of the group members of this project. Their contributions are sincerely appreciated and gratefully acknowledged. The projects as well as letting us understand the concept and the ways to interpret it in real life situation.

We would like to take the chance to express our gratitude to our honorable teacher & Supervisor MD. Shiefuzzaman for his continuous guidance and support regarding this report. Besides this, we have found so many sincere and productive advices from many people that we would like pay homage to them. We convey our gratitude to our honorable Vice Chancellor, Dr. Carmen Z. Lamagna for encouragement.

# Table of Contents

# List of Tables

# List of Figures

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present a detailed usage and functionality of Virtual Doctor. The release no of the document is 1.0. It will explain the features and the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to an external behavior. This document is intended for both the stakeholders and the developers of the system.

## 1.2 Document Conventions

This document follows MLA Format. The bold-faced text has been used to emphasize section and sub-section headings. Highlighting is to point out words in the glossary and italicized text is used to label and recognize diagrams.

## 1.3 Intended Audience and Reading Suggestions

This document is intended for the development team, the project managers, marketing staff, testing team and documentation writers. Our stakeholders and distributors, who market the finished product, may review the document to learn about the project and to understand the requirements. The SRS has been organized approximately in order of increasing specificity. The developers and project managers need to become intimately familiar with the SRS. Others involved need to review the document as such:

**Overall Description** – Marketing staff have to become accustomed to the various product features in order to effectively advertise the product.

**System features** – Testers need an understanding of the system features to develop meaningful test cases and give useful feedback to the developers.

**Nonfunctional and Functional Requirements** – For Developers.

### 1.4 Product Scope

This software system will be web based. It will help patients to identify their real problems or disease and suggest appropriate doctors for the specific problem. It will minimize people's searching time and pain. To find the best specialist with the help of statistics data and AI, this would otherwise have to be performed manually.

More specifically, this system is designed to allow a patient to identify their problem and provide a list of doctors by their category related to patient's problem. The Later patient can also request a schedule from the doctor through this system. The software will facilitate communication between patient and doctor via Video Chat/Email. The system also contains a relational database containing a list of Patients, Doctors, and Symptoms of patients.

# 2. Overall Description

### 2.1 Product Perspective

Medicare System is a web based application which will work as patient's helper to find appropriate doctor if patients feel sick and it will do all the things a patient need when they try to go to doctor and get treatment. It is a system which uses some statistical data and AI and gives patients a quick suggestion about a doctor according to disease and help to find the specialists near to the patient and book their schedule for treatment.

### 2.2 Product Functions

With the application the users will be able to find the doctor that he/she needs. The result will be based on user's input and criteria. It will be possible for the administrator of the system to manage the options for those criteria that have that. The result of the search will be viewed either in a list view. The list view will have one list item for each doctor matching the search criteria and show some information about the doctor so that the user can identify and select the doctor that he/she needs. Beside this it will suggest some primary medicines name depending on the disease he/she have.

## 2.3 User Classes and Characteristics

There are three kinds of users that interact with the system: patient, doctor and administrator. Each of these three types of users has different use of the system so each of them has their own requirements. It is considered that the user should have the basic knowledge of operating the internet and to have access to it. The administrator is expected to be familiar with the interface of the tech support system.

The patient users can only use the application to find a doctor according to his/her disease. This means that the user have to be able to search for doctors, choose a doctor according to his/her disease from that search and then navigate to it. In order for the users to get a relevant search result there are multiple criteria the users can specify and all results matches all of those.

The doctor will manage the information about them, for example his qualification, contact information and the time he will be available.

The administrators will manage the overall system so there is no incorrect information within it. The administrator can manage the information for each doctor as well as the options for both the patient and the doctors.

## 2.4 Operating Environment

Medicare is a mobile based and web based application which will work online. Our Application will run on Android, IOS device having 512 MB RAM, 300MB ROM minimum and for better performance we recommend 1GB of RAM and 500MB of ROM.

## 2.5 Design and Implementation Constraints

The software package should be designed so as to handle the access by minimum —20 users and 1 Admin concurrently.

## 2.6 User Documentation

There will be a basic tutorial of this application that will show how to use this when the user will run this first time. A proper guidance will be given in that tutorial. There will be a help option that will help the user face any kind of problem while using it.

# 3. System Design

## 3.1 User Stories

The system consists of three types of users. They are patient, doctor and admin. There is also an Artificial intelligence system which will try to identify problem of patient and find out which types of doctor he/she should consult with.

### 3.1.1 Patient Perspective:

After login to the system patient will get access to his/her homepage. In home page there will 4 different areas (profile, contact list, Bot contact, Mail box) which he/she can access. In profile area user can modify their information. In contact list recent communication list with doctor will be shown. User can search and add new contact in the contact panel and communicate with doctor using it. The bot area provides basic services to the user when doctor isn’t available. There will be a panel where user will type what problem they are having. According to statistical data bot will provide what problem user might be facing and with which doctor they should communicate with. But to consult with doctor he/she have to make an appointment request. Doctor will then approve/deny the request and give user a possible free time. From mail box area the user will be able send and get email from doctor.

### 3.1.2 Admin Perspective:

Admin will monitor the system and has full access to the database. Admin can also ban user if someone violate the rule.

### 3.1.3 Doctor Perspective:

Doctors have to login first to get access of his her/her profile. Doctor has a schedule which he/she can manage and appoint patient. Using systems feature doctor can communicate with patient. Doctor can send prescription through email. Doctor will have contact list of patient. Doctor will be able to update system database such as adding disease name, symptoms and medicine information which will improve the AI with its statistic data.

# 3.2 UML

## 3.2.1 Use Case Diagram:

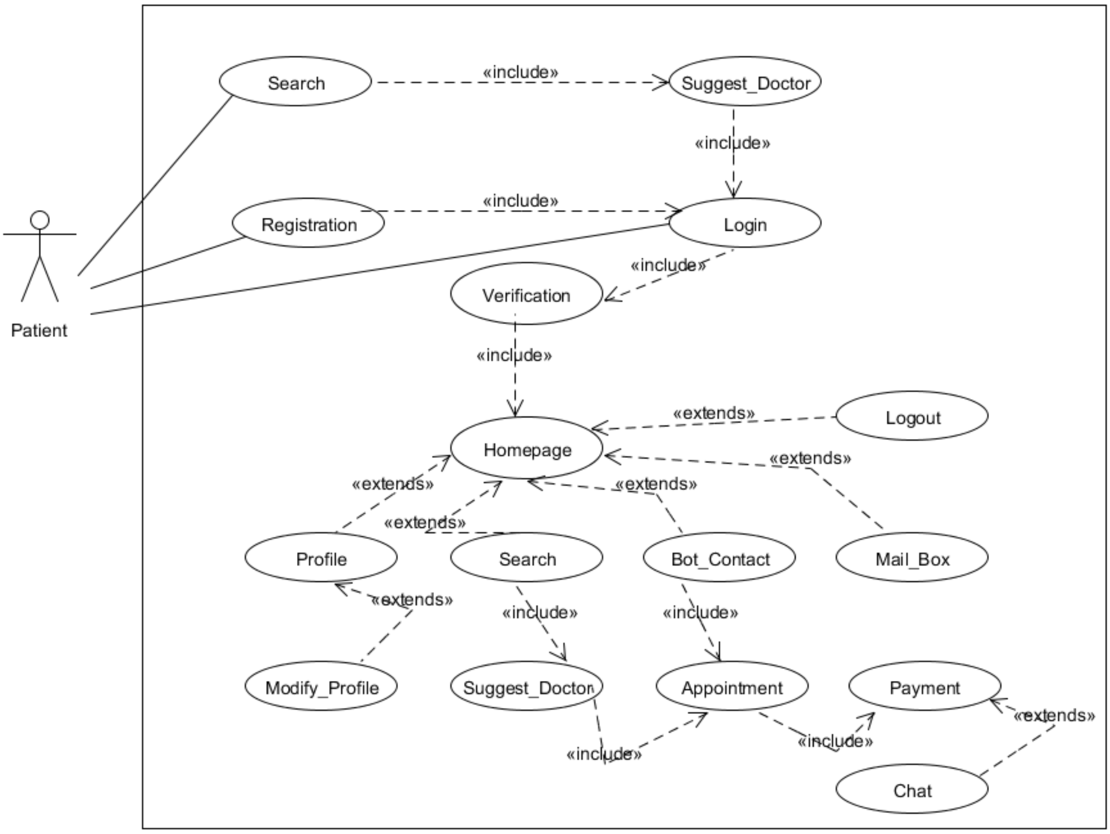


**Figure 3.1:** Patient Perspective

**Scenario:**

- Patient can search for a problem then system will suggest the patient a Doctor.
- Patient can do a registration and log in to the system and it will be verified by the system.
- After logged in patient can check the profile, search for a problem, can contact with admin, and can send mail. Patient can also request for appointment from Doctor.
- Patient can also modify information.

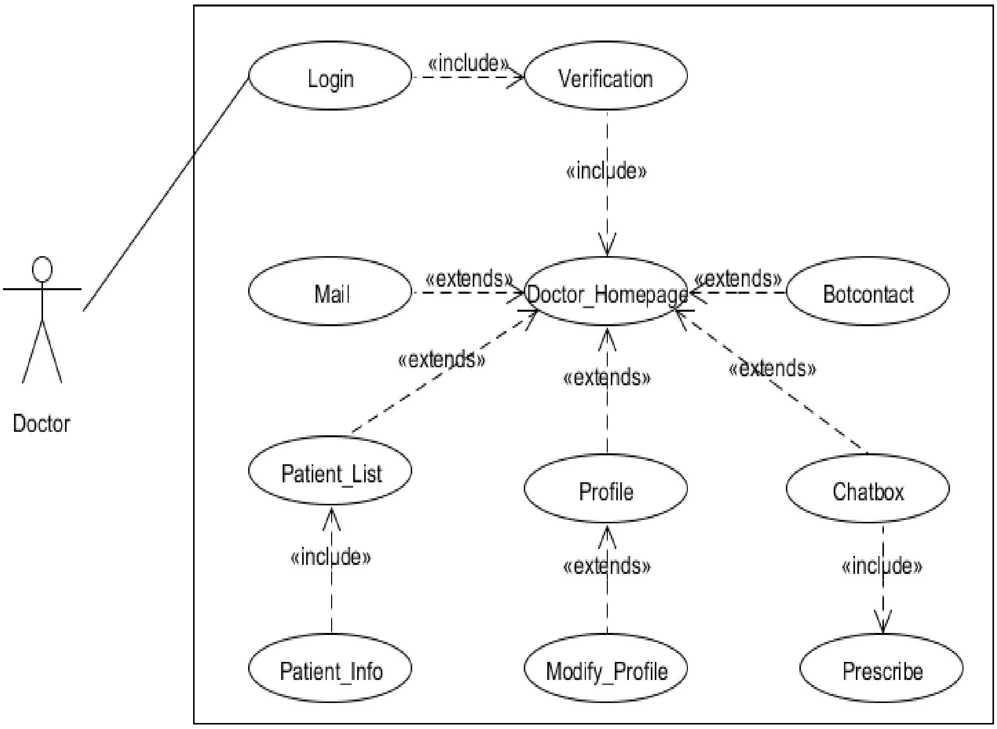


**Figure 3.2:** Doctor Perspective

## Scenario:

- Doctor can log in to the system and it will be verified.
- In Doctor homepage doctor can send mail, can check the profile, can see patient, can contact with the admin, can chat with patients.
- Doctor can modify profile, can see patient's info, can prescribe via chat box.

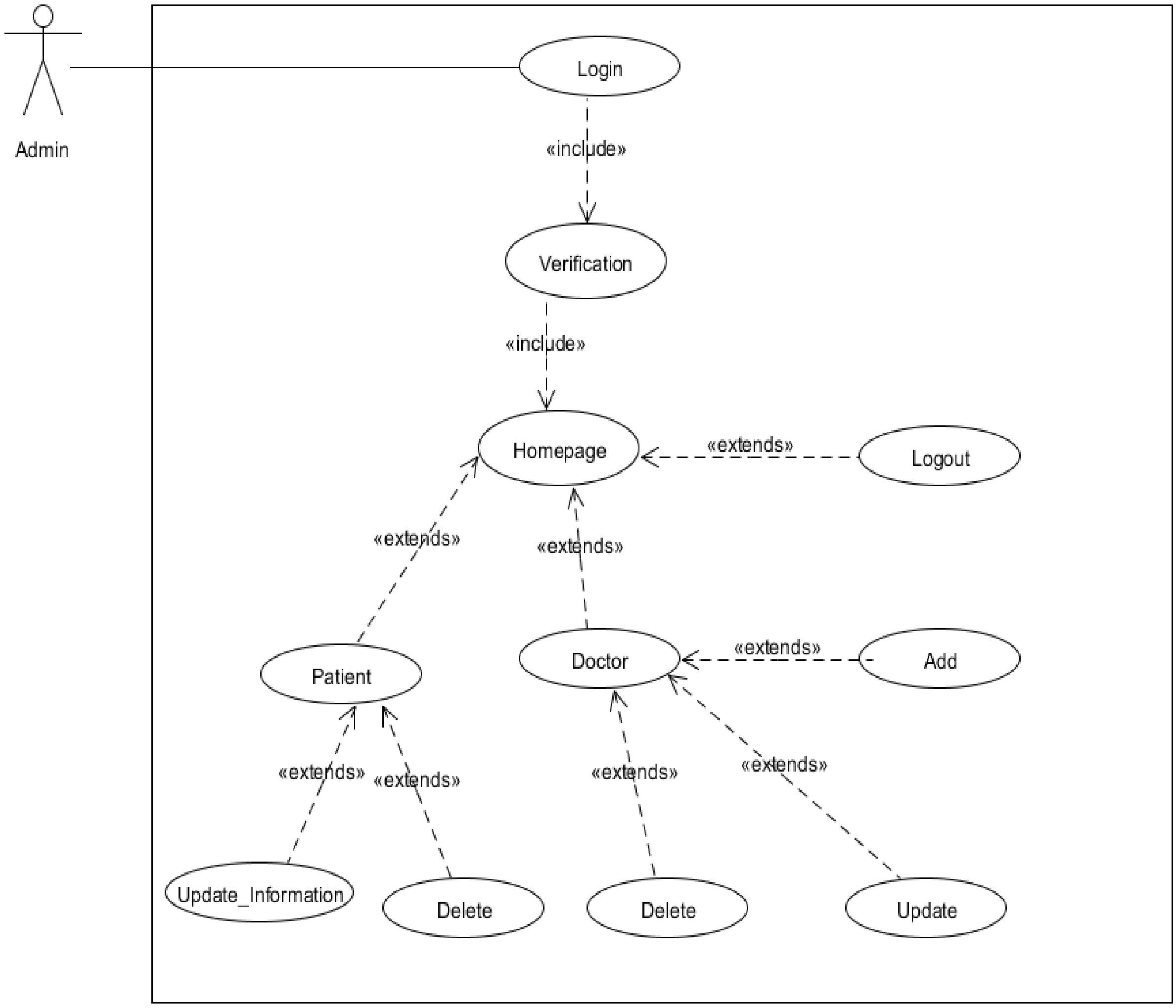


**Figure 3.3:** Admin Perspective

## Scenario:

- Admin can log in to the system and it will be verified by the system.
- In Homepage, admin can give a payment slip, can see the info of patients and doctors. Admin can also add or delete the info of doctors and patients.

## 3.2.2 Class Diagram:

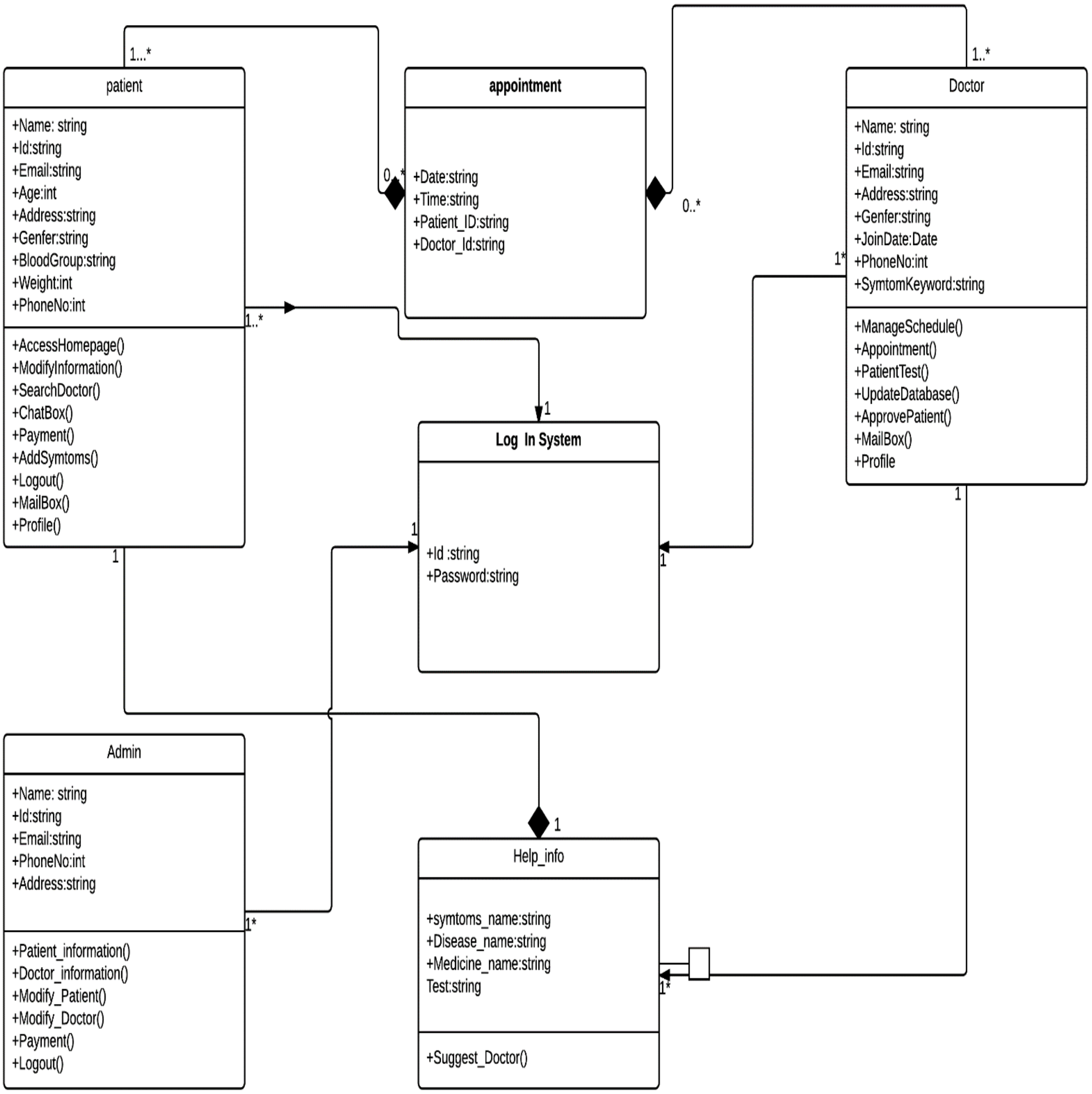


**Figure 3.4:** Class Diagram

### 3.2.3 Activity Diagram:

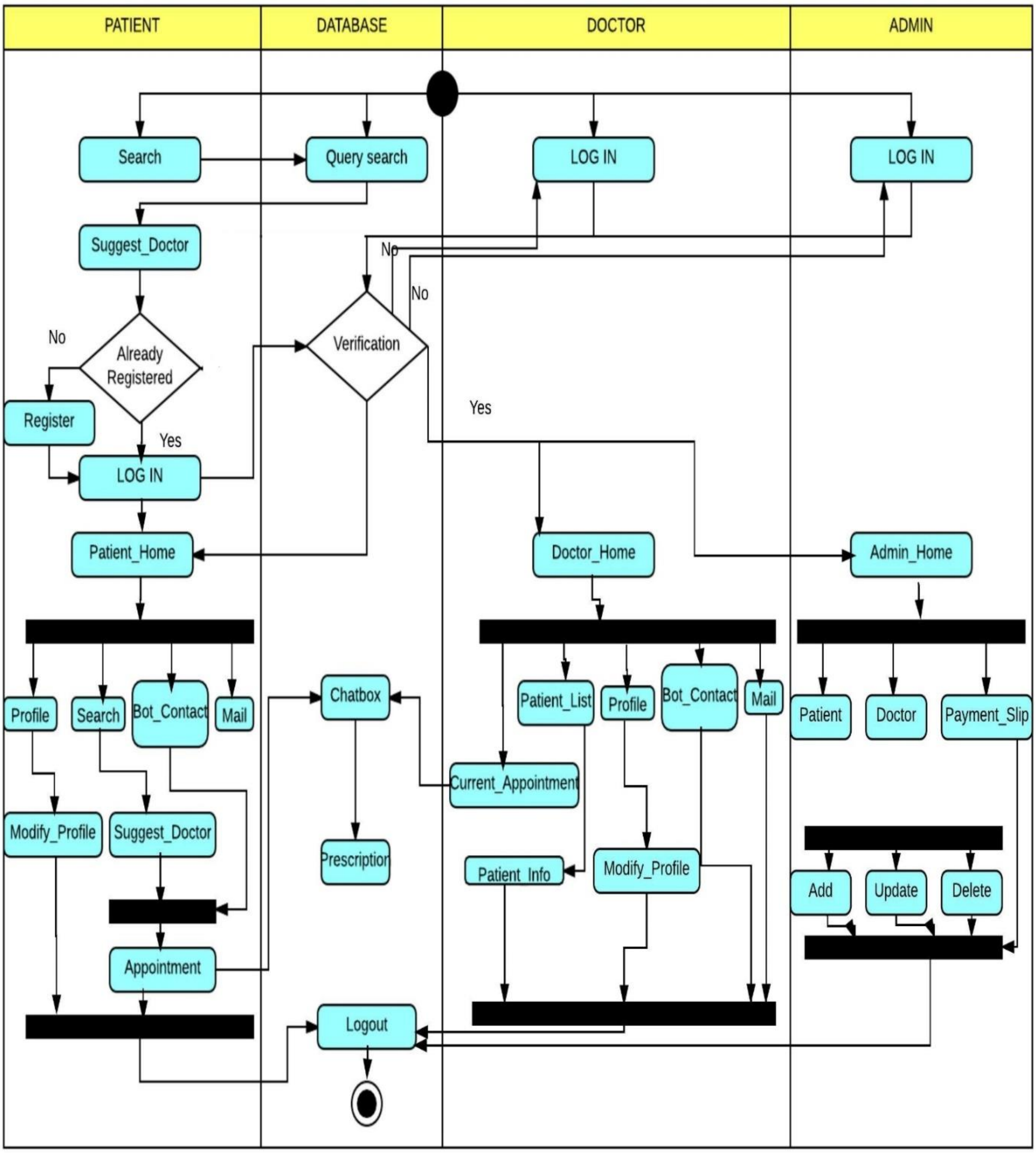


**Figure 3.5:** Activity Diagram

## 3.2.4 Entity Diagram:

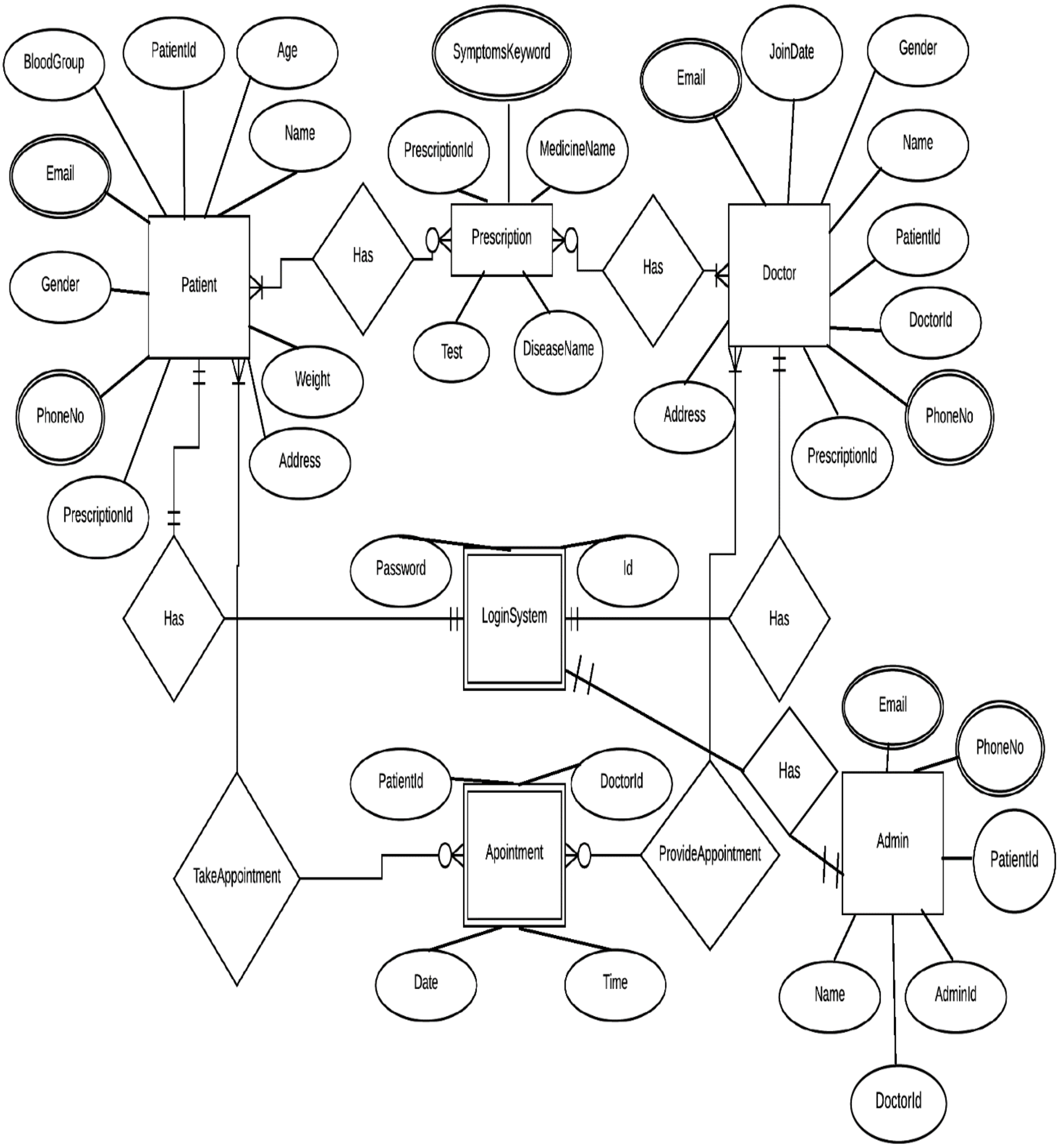


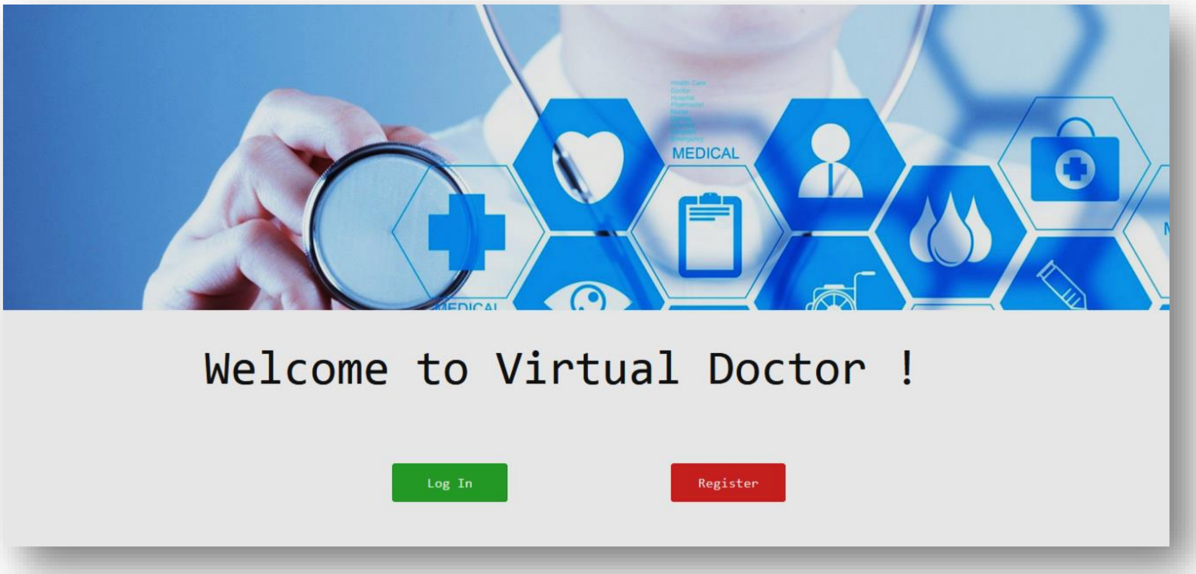
**Figure 3.6:** Entity Diagram

## 3.2.5 Data Dictionary

**Table 3-A:** A table of data dictionary

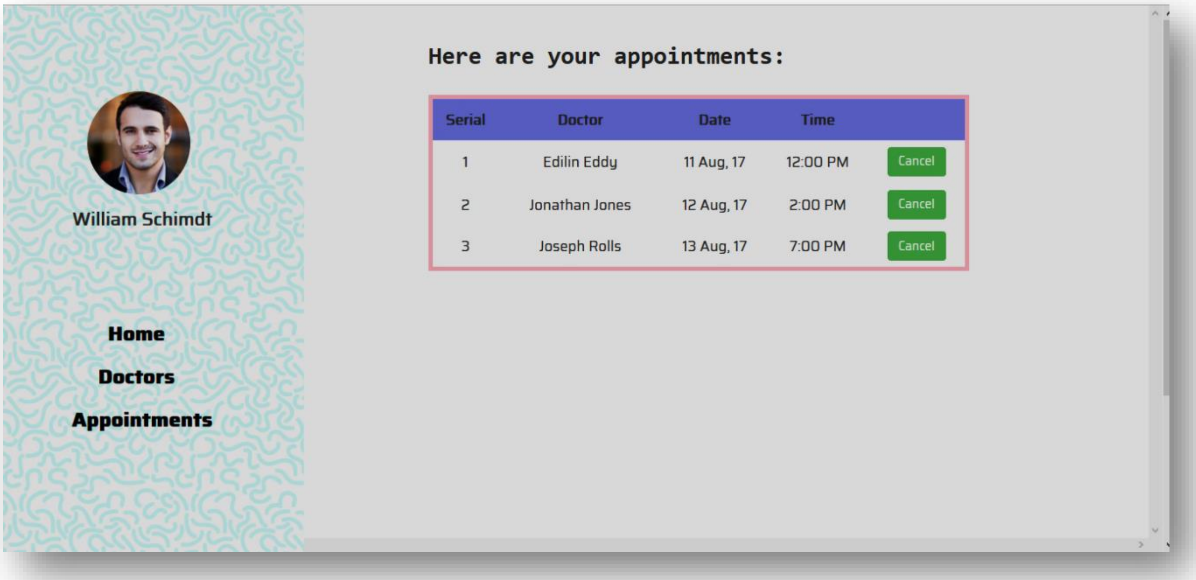
| Entity | Attribute | Type/Size | Validation | Key |
|---|---|---|---|---|
| Patient | Patient_ID | Number(5) | 10000-99999 | Primary |
| Patient | Patient_Name | Text(10) | Required | |
| Patient | Email | Text(10) | Optional | |
| Patient | Age | Number(2) | Required | |
| Patient | DOB | Date(8) | Valid Date | |
| Patient | Address | Text(15) | Required | |
| Patient | Blood_Group | Text(5) | Required | |
| Patient | Gender | Text(1) | Required | |
| Doctor | Doctor_ID | Number(5) | 01-50 | Primary |
| Doctor | Doctor_Name | Text(10) | Required | |
| Doctor | Email | Text(10) | Optional | |
| Doctor | Age | Number(2) | Required | |
| Doctor | NID | Number(13) | Required | |
| Doctor | Gender | Text(1) | Required | |
| Doctor | Address | Text(15) | Required | |
| Admin | Admin_ID | Number(5) | 01 | Primary |
| Admin | Admin_Name | Text(10) | Required | |
| Admin | Email | Text(10) | Optional | |
| Admin | Phone | Text(14) | Optional | |

## 3.3 UI Design Specification



Welcome to Virtual Doctor !

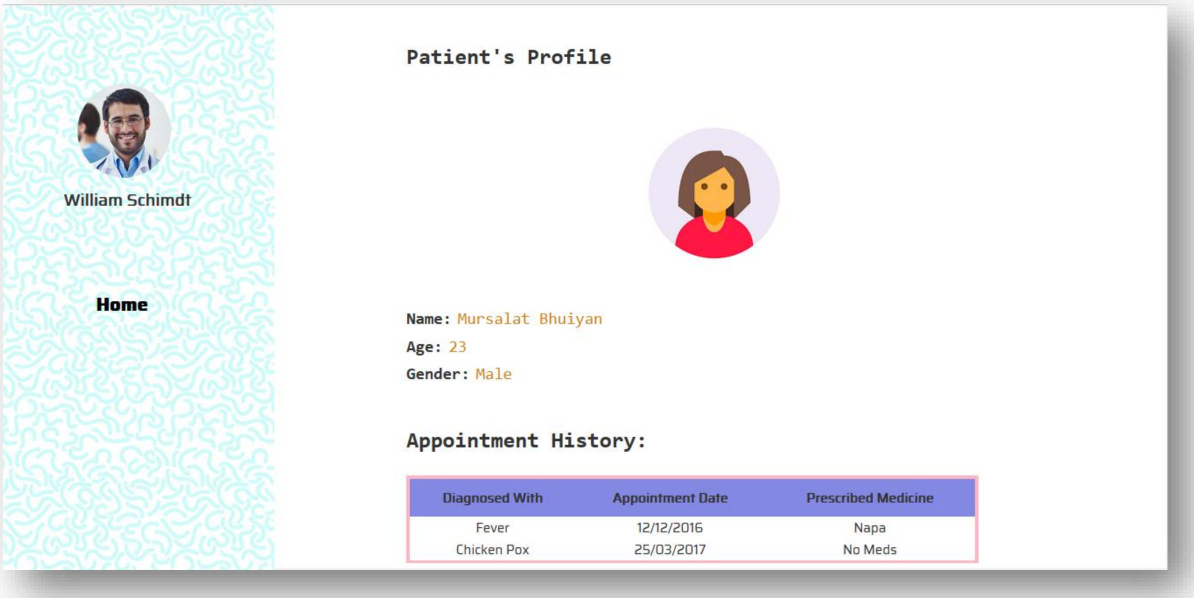
**Figure 3.7:** Home Page



Here are your appointments:

| Serial | Doctor | Date | Time | |
|---|---|---|---|---|
| 1 | Edilin Eddy | 11 Aug, 17 | 12:00 PM | Cancel |
| 2 | Jonathan Jones | 12 Aug, 17 | 2:00 PM | Cancel |
| 3 | Joseph Rolls | 13 Aug, 17 | 7:00 PM | Cancel |

**Figure 3.8:** Doctor’s Appointments List (For Patient)



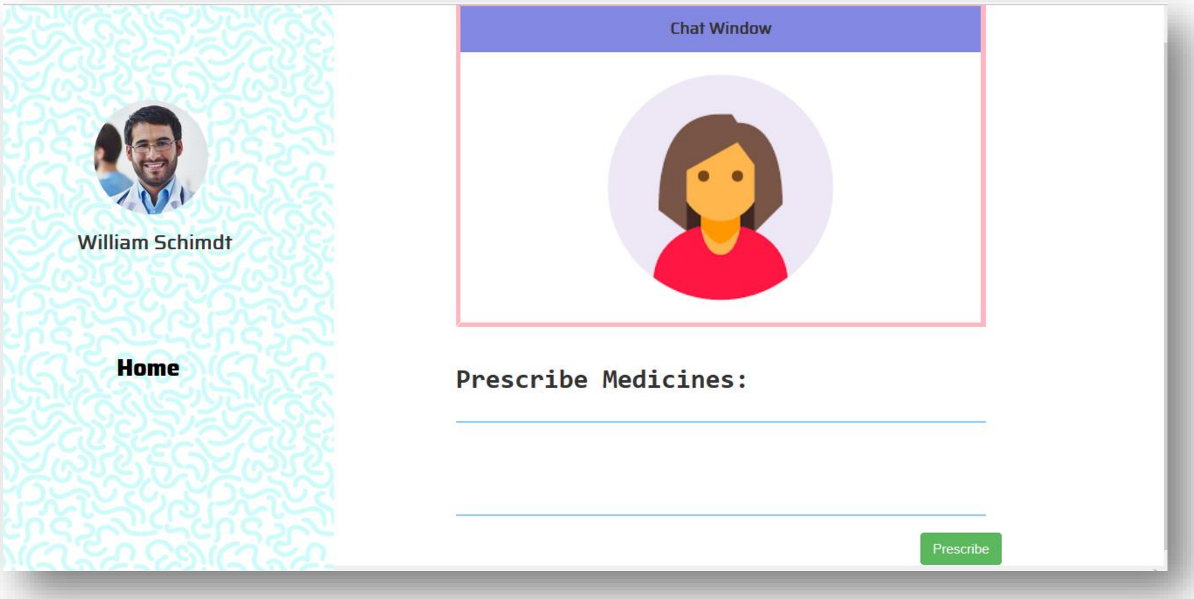
Patient's Profile

Name: Mursalat Bhuiyan
Age: 23
Gender: Male

Appointment History:

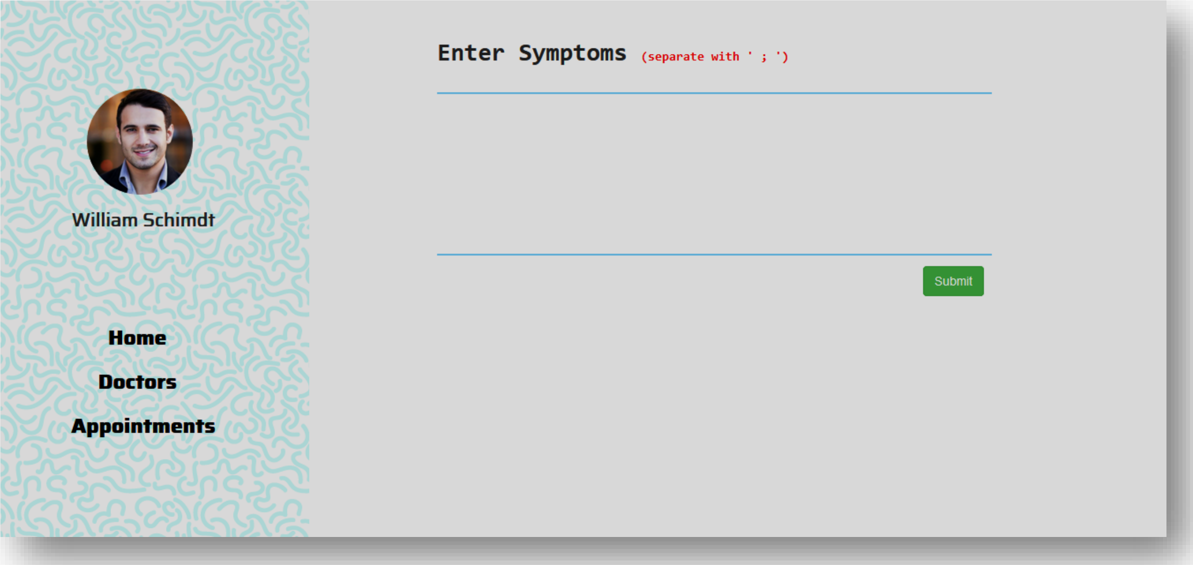
| Diagnosed With | Appointment Date | Prescribed Medicine |
|---|---|---|
| Fever | 12/12/2016 | Napa |
| Chicken Pox | 25/03/2017 | No Meds |

**Figure 3.9:** Patient Profile



Prescribe Medicines:

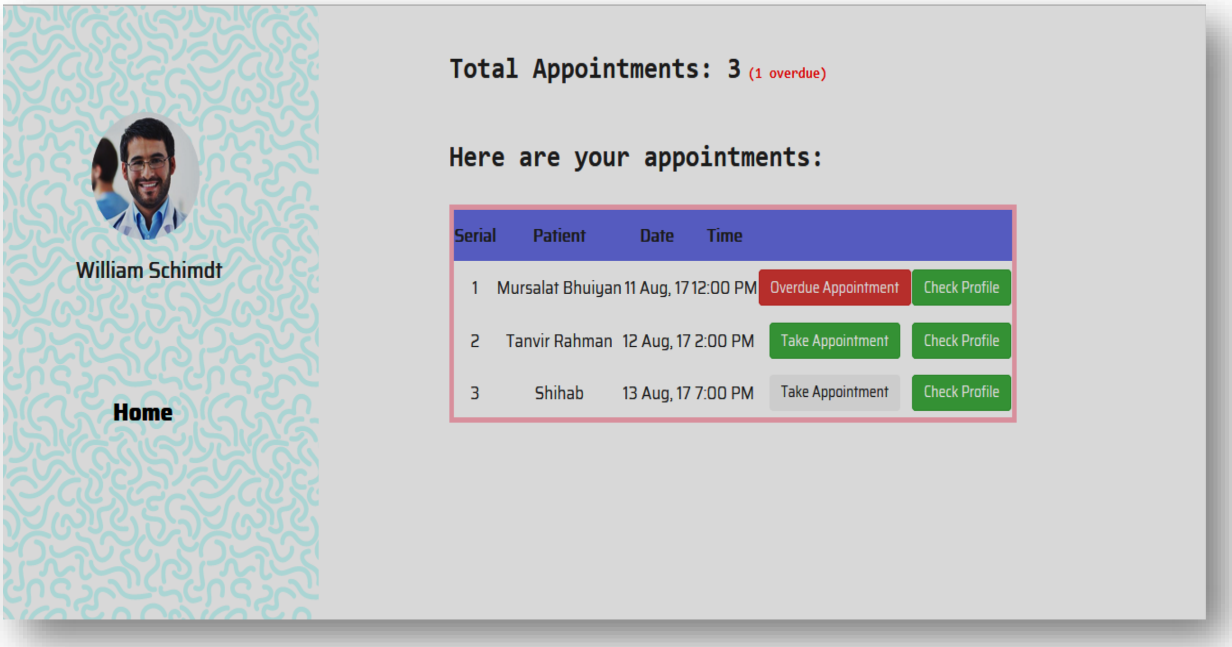
**Figure 3.10:** Chat Page



William Schimdt

Enter Symptoms (separate with ' ; ')

**Figure 3.11:** **AI**



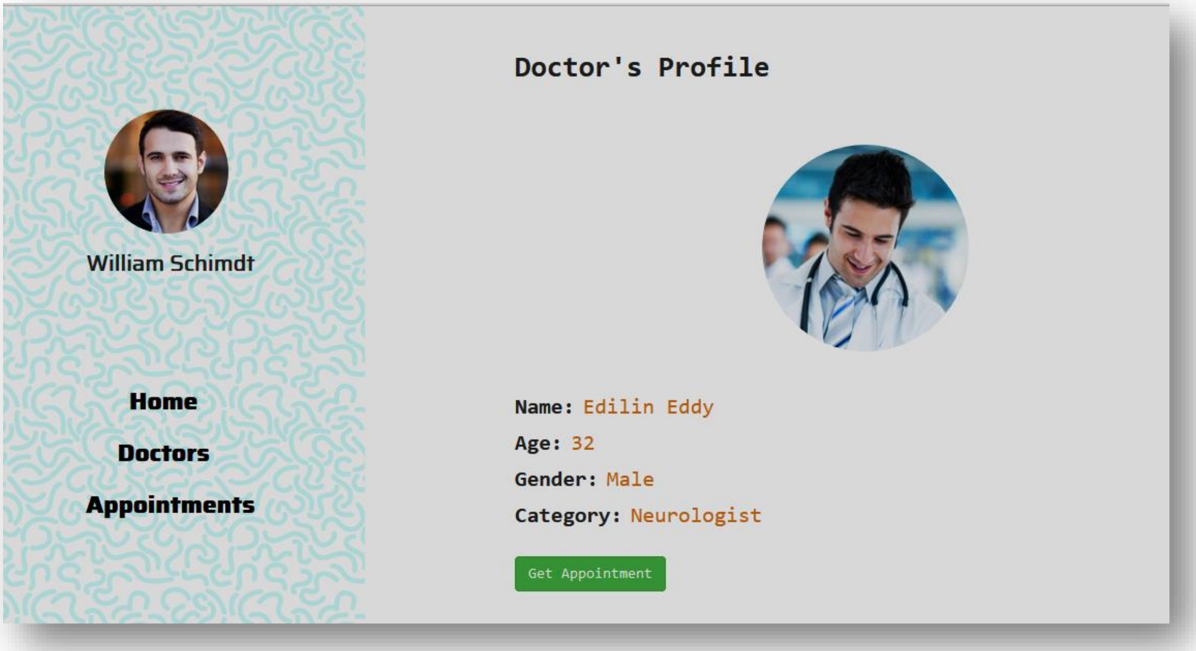
William Schimdt

Total Appointments: 3 (1 overdue)

Here are your appointments:

| Serial | Patient | Date | Time | | |
|---|---|---|---|---|---|
| 1 | Mursalat Bhuiyan | 11 Aug, 17 | 12:00 PM | Overdue Appointment | Check Profile |
| 2 | Tanvir Rahman | 12 Aug, 17 | 2:00 PM | Take Appointment | Check Profile |
| 3 | Shihab | 13 Aug, 17 | 7:00 PM | Take Appointment | Check Profile |

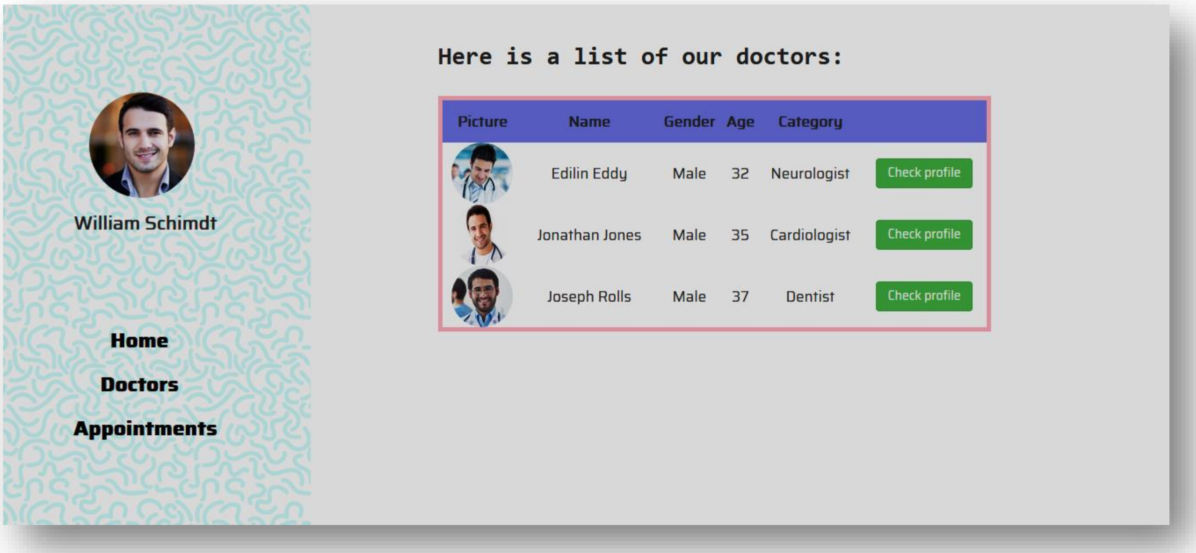
**Figure 3.12:** Patients Appointment List (For Doctor)



Doctor's Profile

Name: Edilin Eddy
Age: 32
Gender: Male
Category: Neurologist

**Figure 3.13:** Doctors Profile



Here is a list of our doctors:

| Picture | Name | Gender | Age | Category | |
|---|---|---|---|---|---|
| | Edilin Eddy | Male | 32 | Neurologist | Check profile |
| | Jonathan Jones | Male | 35 | Cardiologist | Check profile |
| | Joseph Rolls | Male | 37 | Dentist | Check profile |

**Figure 3.14:** Doctors list

# 4. System Requirements/Features

Virtual Medicare offers you the ultimate feature in- medical service system by providing easy access to your profile information using the internet through online. You have complete control of all your medical activities through online at your home or office or anywhere. Making profile inquiries, suggesting appropriate doctors, looking for the doctor's information, finding Perfect doctors for your disease and online payment system.

## 4.1 System features

- Provide online free registration facilities for both patients and doctors.

- Provide Log in services for user security.
- Provide Edit profile option.
- Give the symptoms input capability for parents.
- Search option for available online doctors in case of emergency.
- Provide Email sent and receive facilities for both patients and doctors.
- Provide efficient Search option for both patients and doctors.
- Give admin control over the full system like upload notices, AD, configure AI.
- View available doctor's profile and profile activities.
- Verify patients and Doctor's profile.
- See if a specific activity has done.
- Show available information between profiles.
- Show suggestions of doctors which is given by AI.
- Check doctors response.
- Check prescription information given by doctors.
- Check appointment time information given by doctors.
- Check suggested medicine information given by doctors.
- Verify sufficient availability of patients and doctors.
- Change your password online.
- Free online support.

## 4.2 Description and Priority

**Table 4-A:** Table of Description and Priority

| No | Features Name | Benefit (2) | Penalty (1) | Cost (1) | Risk (0.5) | Priority |
|---|---|---|---|---|---|---|
| 1 | Registration | 8 | 8 | 5 | 5 | 7 (High) |
| 2 | Log in | 8 | 8 | 5 | 3 | 8(High) |
| 3 | Edit profile | 7 | 5 | 4 | 2 | 5 (Medium) |
| 4 | Symptoms input | 5 | 7 | 6 | 4 | 6 (Medium) |
| 5 | Available online Doctors | 6 | 6 | 5 | 6 | 6 (Medium) |
| 6 | Email Service | 6 | 5 | 8 | 7 | 7 (High) |
| 7 | Upload notices, AD, configure AI. | 7 | 8 | 8 | 7 | 7 (High) |
| 8 | Available Doctor’s profile | 6 | 6 | 7 | 6 | 5 (Medium) |
| 9 | Available Patients profile | 6 | 5 | 5 | 4 | 4 (Low) |
| 10 | Suggestions of doctors by AI | 7 | 8 | 9 | 9 | 9 (High) |
| 11 | Doctors response Info | 5 | 4 | 6 | 5 | 6 (Medium) |
| 12 | Prescription Information | 8 | 7 | 8 | 7 | 8 (High) |
| 13 | Appointment Time information | 8 | 7 | 8 | 7 | 8 (High) |
| 14 | Suggested medicine information | 8 | 7 | 8 | 7 | 8 (High) |
| 15 | Change password online | 7 | 4 | 5 | 6 | 4 (Low) |
| 16 | Online support | 6 | 6 | 6 | 5 | 5 (Medium) |
| 17 | Search Option | 6 | 5 | 5 | 6 | 6 (Medium High) |
|  | Total |  |  |  |  |  |

## 4.3 Functional Requirements

The system has the following set of functional requirements.

1. **REQ_FUNC_1** The system shall have a dynamic web interface for the user of the system.

2. **REQ_FUNC_2** The system shall have a Graphical User Interface for Admins.

3. **REQ_FUNC_3** The system shall have a Graphical User Interface for Patients/Doctors.

4. **REQ_FUNC_4** Clients shall be able to sign up for free accounts for online Medical services using their appropriate information and the password will be randomly generated and emailed to them. A verified Email has to be added by Clients. Users can change the password later through Email verification.

5. **REQ_FUNC_5** The system shall have a secure login system using mid-level encryption for the Patients and Doctors and high-level encryption for the Admins.

6. **REQ_FUNC_6** The system shall support multiple logs in i.e. more than one Doctors and Patients can login/sign in into the system at the same time and the same goes for the Admins.

7. **REQ_FUNC_7** The Patients and Doctors will have a “restricted” access to the system depending on the rights granted by the System authorities.

8. **REQ_FUNC_8** While creating an account, a user’s shall be given required personal information.

9. **REQ_FUNC_9** Users must be able to give or take information related to their medical conditions or doctors’ information’s. (Appropriate)

## 4.4 Non-functional Requirements

### 4.4.1 Performance Requirement

The requirement in the section provides a detailed specification of the user interface interaction with the software and measurements placed on the system performance.

### 4.4.2 Response time

The maximum response time for the submission of a job will be 2sec. The system must be interactive and delays involved must be less. So in every action response of the system, there are no immediate delays. When connecting to the server the delays are based editing on the distance of the 2 systems and the configuration between them so there is high probability that there will be or not a successful connection is less than 20sec for good communication.

### 4.4.3 Prominent search feature

The search feature should be prominent and easy to find for the user.

### 4.4.4 Usage of the result in the list view

The result displayed in the list view should be user friendly and easy to understand. Selection an element is the result should only take one click.

### 4.4.5 System dependability

If the system loses the connection to the internet or to the GPS device or the system gets some strange input, the user should be informed.

### 4.4.6 Security Requirement

The program uses object oriented mechanisms to protect its data passed using methods.

### 4.4.7 Data transfer

The system shall use secure socket in all transaction that include any confidential patient information.

The system shall automatically log out all users after a period of inactivity. The system shall confirm all transactions with the patients and doctors web browser.

The system shall not leave any cookies on the patient's device containing the user's password and confidential information.

### 4.4.8 Communication security

This is the security of the communication between the system and server. The messages should be encrypted for log in communication. So other cannot get users name and password from those messages.

### 4.4.9 Data storage

The system's back-end server shall only be accessible to authenticated administrators. The system's back-end database shall be encrypted.

## 4.5 Other software Quality attributes

### 4.5.1 Availability

The system will only run infrequently. The system will allow the user to restart the application after a crash. The user will be able to load his or her data file after the system has been restarted and continue using the system. If the internet service gets disrupted while sending information to the server, the information can be sends again for verification.

### 4.5.2 Safety

Information transmission should be securely transmitted to server without any changes in information which will improve the AI with its statistics data. The system will be built using components that are as independent as possible to make the system easily modifiable. Ai component of the system will be modular and be as independent as possible.

### 4.5.3 Usability

Checking that the system is easy to handle and navigate in the most expected way with no delays .In that case the system program reacts accordingly and transverses quickly between its states.

### 4.5.4 Reliability

The system shall give the right result on a search .When a patient describes his symptom; the system should suggest him a correct and related doctor's name as well as information.

### 4.5.5 Portability

The system will be able to run a Mac OS X, Android and windows .The software will be written in a platform independent programming language for portability, there will be no platform specific code.

### 4.5.6 Maintainability

The doctor will be able to update the system with new disease name, symptoms and medicine information, which will improve AI with its statistics data. The system will be build using components that are as independent as possible to make the system easily modifiable. AI components of the system will be modular and be as independent as possible.

# 5. References

- IEEE. IEEE STD 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
- OMG. “Unified Modeling Language Specification”, Superstructure Version 2.1.1, Febrer 2007
- [ChC90] E.J.Chikofsky, J.H.Cross. “Reverse Engineering and Design Recovery: A Taxonomy”, IEEE Software, Gener-Febrer 1990 Vol.7 Núm. 1 pp.13-17
- [ChL99] L.L.Constantine, L.A.D.Lockwood. “Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design”, Addison-Wesley Professional, 1999
- [CST03] D.Costal, M.R. Sancho, E.Teniente. “Enginyeria del Software: Especificació. Especificació de sistemes orientats a objectes amb la notació UML”, Edicions UPC, 2003
- [FAT03] E.Faivre, L.Abbal, T.Murail. “EasyPHP”, 2003 http://www.easyphp.org/
- [GNU91] Free Software Foundation, Inc. “GNU General Public License”, Version 2, Juny 1991 http://www.gnu.org/licenses/gpl.html
- [Oli02] A.Olivé. “Modelització conceptual de Sistemes d’Informació. L’estructura”, Edicions UPC, 2002
- [Oli07] A.Olivé. “Conceptual Modeling of Information Systems”, Springer, 2007

- [OMG06] OMG. “Object Constraint Language Specification”, Version 2.0, 2006 http://www.omg.org/technology/documents/formal/ocl.htm
- [OMG07] [USE07] University of Bremen. “A UML-based Specification Environment”, 2007 http://www.db.informatik.uni-bremen.de/projects/USE/