Software Requirements Specification

Introduction

This project deals with the development and implementation of a smart-phone application that lets you make a pre-appointment with your doctor, digitally stores all our medical records on a secure cloud server and gives our loved ones peace of mind by reminding us to take our medicine on time. Users can easily make appointments with a doctor of their preference which is suitable to their convenience of time and date. As well as, users can input the symptoms that he/she is suffering from, and will be able to get an idea of the disease what he/she is suffering from. Medical records of the specific user will be uploaded by the doctor or hospital which can be easily accessed and downloaded by the user whenever needed. Using our phones facilities the application can enable reminders to take the specific medicine at the specific time according to our medical prescription which is also uploaded by our doctors. Additional features of calling emergency services, giving location based details of hospitals and medical care facilities and general tips for health care are also incorporated in the application.

Purpose

The main aim of our project is:

- To provide easiest way to deal with disease prediction, pre-appointments, reminder and scheduling system.
- To maintain distinct accounts this can be used by multiple users and would be independent of the devices. The user can log in through any mobile device having the app installed.
- To ease the process of taking appointment.
- Store information securely in data base.
- The core concept for this project is to avoid waiting in the queue to take appointment.

Project Scope:

- Proposed application can be used by all and is user friendly.
- This application will allow the user to take appointment, cancel appointment with the doctor of his choice.
- This application will be innovative in its own aspect as it will accommodate the whole process of adding reminders for taking medicines and will minimize the dependency of user on other available options.

- The application will also consist of facility to predict the disease just by entering the symptoms.
- This app would combine a number of functionalities into one, so the user need not download a number of applications for performing different tasks.
- The future scope would be to use the GPS system to deal with some emergency services in case of accidents.

Product Features

- The idea of our app differs as it can be used by multiple users. For example, our app would require the user to create his account which he would access from any android device.
- The data provided by the user in this app during signup would be saved in the database and would be independent of the devices.
- Our app would work like a social networker where the user would be able to create an account and use this app from any device which supports android OS, to book the ticket.

Use Classes and Characteristics

1. **List of actors and their details:** Actors in this case are the users/patients which would interact with the system to book the tickets. Also the doctor to confirm the Appointment, update schedule, and upload patient history and the hospital just registers the user/patient.

2. **Use case description:** The main purpose of the use case diagram is to make the user understand the basic functionalities of the particular project. The use case describes how the user uses the application to fix an appointment with the doctor or to check his medical history, etc. For instance, the user should sign up first that is user registration should be done and that information of user will be stored in the database. After user registration, user can perform following operations like fix an appointment with the doctor, check his/her medical history, view the prescription to be taken, set a reminder for his medicines. The doctor here confirms the appointment, updates the daily schedule, uploads the patient history and uploads the prescription to be taken by the patient. With all these we are also providing the delete, update and validate operations on the server side

3. **Preconditions:**
   - There should be Internet connectivity available.
   - The user should have android compatible smart device.

- The application should be installed by the user.
- Servers should be available at the time of request.

A. Use Case Suite

I. User To-Do list management

- User registers with the system.
- User logs into the system.
- User views the hospital details.
- User takes an appointment.
- User views his medical history.
- User views the prescription.
- User can set an alert/reminder.
- User can cancel the appointment.
- User logs out from the system.

II. Hospital To-Do list management

- Registers the patient.

III. Doctor To-Do list management

- Doctor confirms the appointment.
- Doctor updates daily schedule.
- Doctor uploads patient history.
- Doctor makes urgent schedule changes.
- Doctor uploads prescription.

B. User Story

User enters his username and password to log in to the system. User fixes an appointment with the doctor. Doctor confirms the appointment and updates the schedule. User sets an alert of this appointment. User logs out of the system.

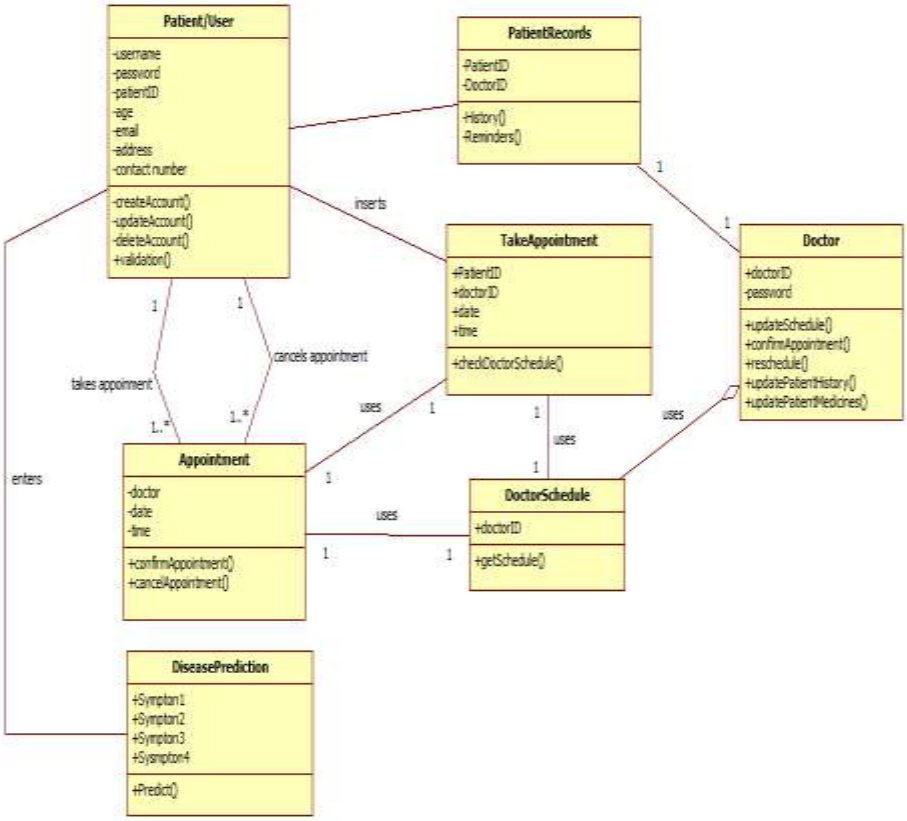

Figure 1: Class Diagram

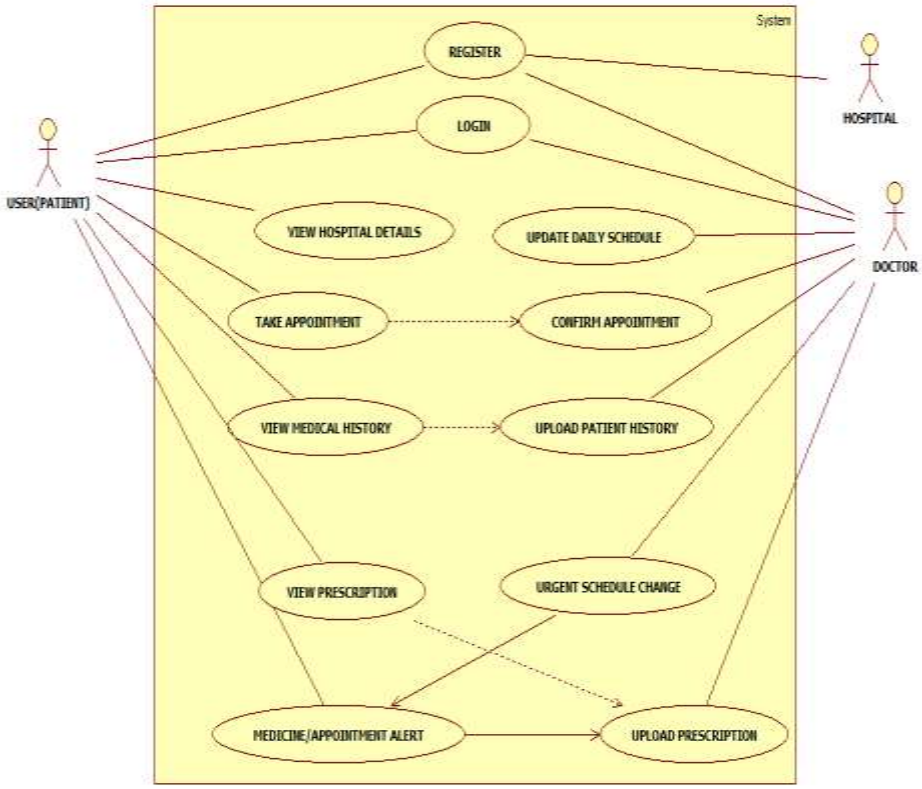

Figure 2: Use Case Diagram

Operating Environment

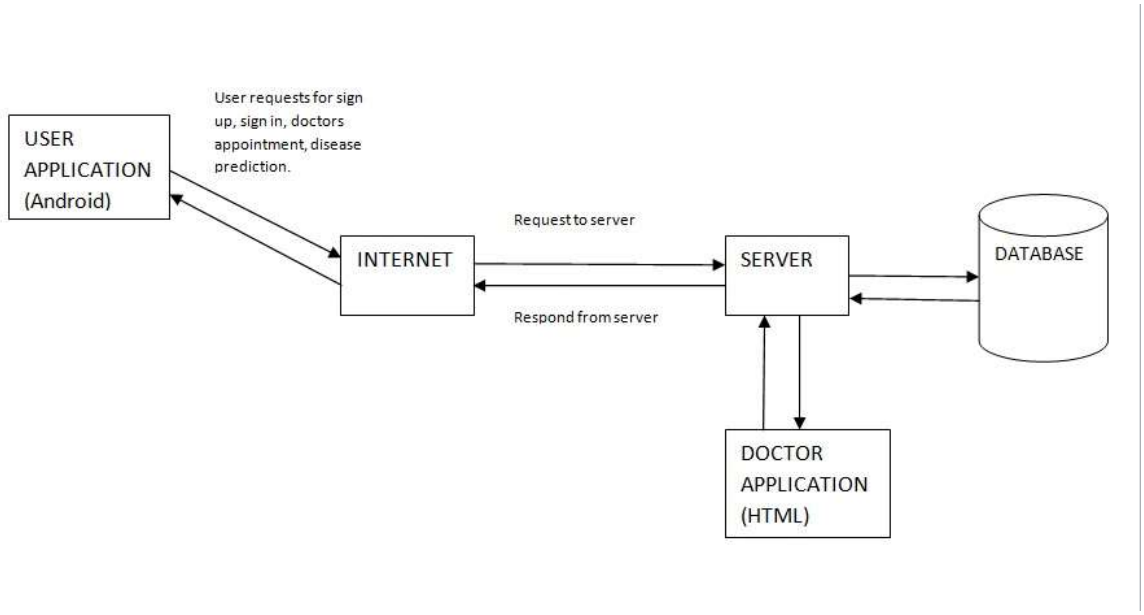

Figure 3: Operating Environment 1

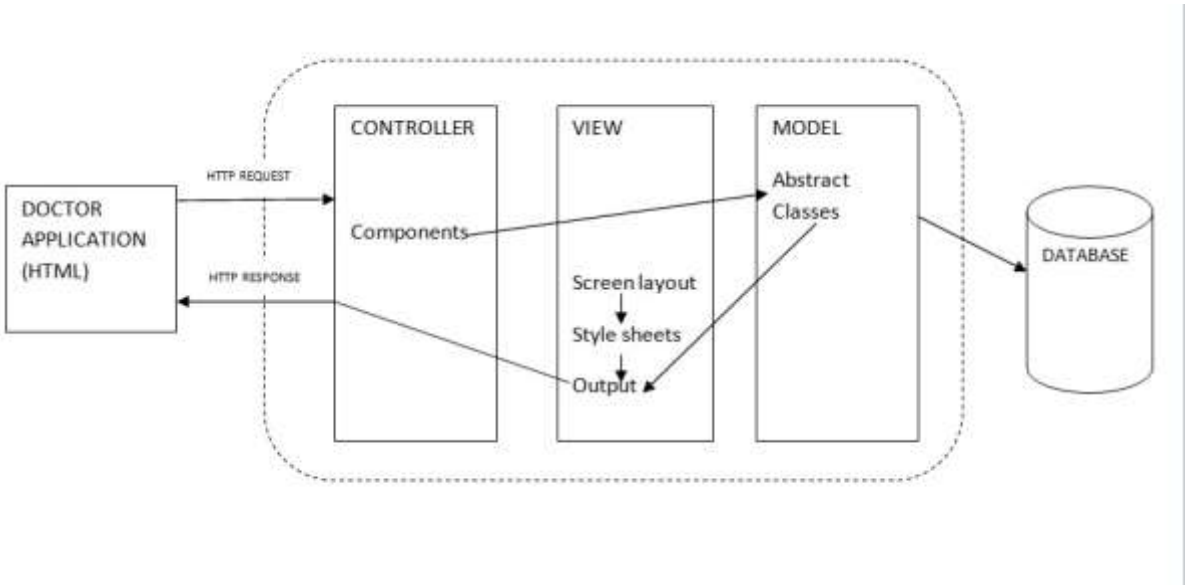

Figure 4: Operating Environment 2

Step 1: The work here starts during the first time installation of our application where the user has to sign up. During sign up the patients personal and basic details need to be entered like the first name, last name, date of birth, any genetic disease, or any other health ailments. These details will be gathered and will be stored into the database. Once the user has an account he can sign in directly. Thus the user can use different android phones or the webpage to log in and will not be restricted to only his phone.

Step 2: Once the user registers, he can sign in anytime. The user can check the hospital details and perform various actions. The user can fix an appointment with the doctor of his choice by checking the doctor's daily updated schedule. With this schedule he can check for the doctor's availability.

Step 3: If the doctor is available, then he acknowledges the request by the user and confirms the appointment. The user gets a confirmation and can set a reminder/alert as to when the appointment is scheduled.

Step 4: The doctor can make urgent schedule changes also which will lead to cancellation or confirmation of the appointment. The doctor can also upload the prescription of the medicines to be taken by the user.

Design and Implementation Constraints

- Regulatory Policies: There are no regulatory policies.
- Hardware limitations: There are no hardware limitations.
- Interfaces to other applications: There shall be no interfaces.
- Parallel operations: There are no parallel operations.
- Audit Functions: There shall be no audit functions.
- Control Functions: There shall be no control functions.
- Higher order Language Functions: SQL shall be used for the database information.
- Signal handshake protocols: There are no signal handshake protocols.
- Reliability Requirements: Total number of bugs in the system shall not exceed 1% of the total line number of code, except connection reliability which is out of our range.
- Criticality of application: The server application shall be available 365 days.
- Safety and Security issues: The password and a valid user name are the security issues. Data protection shall be satisfied by the backup process at the server side.

Assumptions and Dependencies

- The user must have the ability to use the Internet.
- The user must be connected to the Internet to use the system.
- The users' mobile phone must be equipped with the Android Operating System.
- The accuracy of information of the users is the responsibility of all users.

System Features

It consists of the features which includes the following:

- ❖ View Hospital Details
- ❖ Take Appointment
- ❖ View Medical History
- ❖ View Prescription
- ❖ Alert/Reminder

System Feature

ID	*View Hospital Details*
Description	*This includes information related to the hospital and the various doctors practising in that hospital.*
Actors	*The user will be the actor.*
Preconditions	*To view the hospital details the user must be registered on the application only then he will be able to use this service.*
Basic Steps	*The user will select the view hospital details option. The database checks for the details related to the selected hospital and provide the user with the information.*
Exceptions	*Exception which could arise would be the user selects a hospital which is not available in the systems database.*

System Features: View Hospital Details

ID	*Take Appointment*
Description	*Here the user takes an appointment with the doctor of his choice after checking that doctor's schedule for availability.*
Actors	*The user will be the actor.*
Preconditions	*To take an appointment, the user must be registered on the application only then he will be able to use this service. Also the doctor should be available to confirm the appointment.*

Basic Steps	*The user checks the doctor's schedule, checks for availability and then takes an appointment with that doctor if he is available.*
Exceptions	*Exception occurs if the doctor hasn't updated his schedule then it would lead to clashes with the appointment timings.*

System Feature: Take Appointment

ID	*View Medical History*
Description	*This includes the medical history of the patient which will be uploaded by the doctor.*
Actors	*The user will be the actor.*
Preconditions	*To view the medical history, the user must be registered on the application only then he will be able to use this service. Also the medical history has to be uploaded by the doctor.*
Basic Steps	*After logging in to the application the user can view his medical history which is uploaded by his doctor.*
Exceptions	*It wouldn't display the medical history if it is not uploaded in the database by the doctor.*

System Feature: View Medical History

ID	*View Prescription*
Description	*The medicines prescribed by the user's doctor are displayed here.*
Actors	*The user will be the actor.*
Preconditions	*To view the prescription the user must be registered on the application only then he will be able to use this service. Also the doctor should have prescribed some medicines to the patient.*
Basic Steps	*The user selects the view prescription option and checks what medicines the*

	doctor has prescribed for him.
Exceptions	*If the doctor has not prescribed any medicines for the user then it will not display anything.*

System Feature: View Prescription

ID	*Alert/Reminder*
Description	*The user can set a reminder/alert as to when his next appointment is or what time he has to take his medicines.*
Actors	*The user will be the actor*
Preconditions	*To set an alert/reminder the user must be registered on the application only then he will be able to use this service. Also either the user must have an appointment scheduled or some medicines prescribed to set an alert for the above.*
Basic Steps	*Once the users' appointment is confirmed he can set a reminder to ensure that he doesn't miss it. Similarly he can set an alert for his medicine ingestion.*
Exceptions	*If the user has no appointment scheduled or no medicines prescribed he cannot use this service.*

System Feature: Alert/Reminder

External Interface Requirements

E-health care facility system provides a group of works with interface environments. Also there will be a database which will keep all the records of the user while visiting the page.

1. **Hardware Interfaces:** There is no need of any hardware interface for this System.

2. **Software Interfaces:**
   a) **Two product options for viewing**
      i) **Name: Android device**
      Version number: Android GingerBread or later versions.

Source: Google inc.
Purpose: The operating system specified above is required as the container of the client software at the client site in order to execute the client site of User interface.
Definition of the interface: Android is a smart phone OS which provides an interface to work on any smartphone.

ii) **Name: BlueStacks**
Version Number: BlueStacks version 0.7.7.813
Source: Opensource Application developed by Silicon Valley-based software company BlueStacks
Purpose: The software specified above enables users to use this app on desktops and computers that are not equipped with Android.
Definition of the interface: BlueStacks is a software that simulates an Android OS.

b) **Name: Apache HTTP Server**
Version Number: 2.0.5.5
Source: The Apache Software Foundation.
Purpose: In order to execute the client site part, the web server specified above is required as the provider of the client software at the server site.
Definition of the Interface: The Apache Server Project is an effort to develop and maintain an open source server for modern OS. The goal of this project is to provide a secure, efficient and extensible server that provides services in sync with the current standards.

c) **Name: MySQL**
Version Number: 5.0
Source: Oracle Corporation.
Purpose: Required as database server.
Definition of the interface: MySQL is the world's most popular open source DB software. With superior speed, and ease of use, MySQL has become the preferred choice of corporate IT managers because it eliminates the major problems associated with downtime, maintenance, administration and support.

Communication Interfaces

The default communication protocol for data transmission between the server and the client is the Transmission Control Protocol/ Internet Protocol (TCP/IP). At the upper level, Hyper Text Transfer Protocol (HTTP, default port=80, default of Apache port=8080) will be used for communication between the web server and the client.

Non-functional Requirements

Performance and Scalability Requirements

The application can sustain long periods of continuous usage by one user or multiple users. The database that will be accessible by the server will hold tables, tables that will contain information about the users, detailed information about locations and fares. Authentication information like username and password needs to be stored. The database will grow as number of users increase. The scalability requirements of the system are another important issue as well as the performance requirements. The application should work efficiently with 1 thousand users approximately using the application simultaneously. The system will have ability to provide all users with efficient support, which will not be broken down.

Safety Requirements

Any safety problem will not take place throughout the lifecycle of the software system. Every data can be accessed and seen just after data entrance. Safety factors will be supplied through:

- Physical server security.
- Disaster recovery plan.
- Back up of data.

Software Quality Attributes

1. **Usability:** This project deals with the development and implementation of a smart-phone application for E-Health care facilities. To use this application we need to use a smart phone with android as operating system.
2. **Maintainability and Upgradeability:** Making changes or upgradeability in the system will not be that much difficult. By having some knowledge of programming, some features of the system might be converted to a new version. According to the needs of upgrade, system requirements might change such as change in operating system or not.
3. **Supportability and Operate ability**: Supportability will be provided over the whole product life of the system. System will be quite easy to use but educational support will be given if needed. The application is a multi-user web based app, can be run on every smart phone (Android) and Internet connection has to be established before using this application. The user is expected to be comfortable using android OS and have basic knowledge of English.
4. **Business Lifecycle:** The application is designed for everyone. It can be used by business professionals or students. Hence, this system is feasible for a range group of business and great number of people in any sector. Some innovations in the system may be performed and can have a greater range of business life

Other Requirements

System Hardware Requirements

- Smart Phone (Android).
- A server to process all the functions having high power and multiple core processors.

System Software Requirements

- **Language: Java J2SE and JDK**
  J2SE (Java 2 Standard Edition) Java would be the required as language for development of the project. JDK is the development kit used to compile java programs.

- **IDE: Eclipse, SDK**
  Just like visual studio provides development environment for VB and .Net, Eclipse provides an integrated development environment (IDE) for Java.

- **Database, Data Library**
  Serialized Objects / Serialization - Database in Java
  In case the project needs database this is how it is handled in java.
  - First step is to use data structures like Vectors and Lists. These come under Java Collections API.
  - Secondly we declare our own classes using these data structures. E.g. a class Student to hold all the student information. Now these classes need to be pre-compiled and called within Java application as libraries. This is called as a Java Class Library
  - Now class objects cannot be saved to hard drive directly. We need to convert these objects to bytes so that they can be saved to hard drive. To do this we must use a concept called as Serialization. Basically it is a concept where in objects are converted to byte streams so that they can be saved to hard drive or sent via internet and vice versa. The reverse process is called as deSerialization.
  - Finally to save these bytes to hard drive or to send them via network we need Java I/O.

- **GUI**
  AWT and SWING are used for GUI design.

- **Computing Architecture**
  In order to implement architecture or a Software As A Service (SaaS architecture) we need. Web Service – we need to implement a web

service. GlassFish Server – to host web service. SOAP API – to be able to call web service at client side we need to use SOAP API or even XML.

- **Rich Client Side Applications**
  When implementing client-server applications or even based applications, the client side applications can be implemented using architecture of java called as (JWS) Java Web Start. This allows us to create applications with rich GUI's which are also called as Rich Internet Applications (RIA). These are smarter than implementing conventional web pages.

Performance Expectations

The application can sustain long periods of continuous usage by one user or multiple users. It will load the home page quickly within 2 seconds. The database that will be accessible by the server will hold tables, tables that will contain information about the users, detailed information about locations and fares. The database will grow as number of users increase.

User and human factors

The user is expected to be comfortable using a smart phone with android OS, Internet and have basic knowledge of English.

Physical Environment

The application will be a multi user application. The user can use this application once downloaded from hosted site. The Internet connection should be established before using this application.

Interface Requirements

No, we do not need an interface with any other system for it to function properly.

Security Requirements

Security breaches on these kinds of application and are of a major concern because it can involve both enterprise information and private customer data. Security includes processes for authentication, authorization and information handling. Building security into the app from the beginning can be more effective and less disruptive in the long run. In our application, security is required for maintaining privacy and preventing illegal use so that the authentic user should be able to access his own account. This is provided by authenticating username and password. If the user enters the wrong password three times then his account will be kept on hold and he won't be able to access it for 24hrs.

Quality Requirements

Application Quality (Non-Functional) Requirements

The application will be user friendly so the user will be able to use it efficiently. Also it will work efficiently on all android platforms. The system is expected to sustain long periods of continuous usage by one user or multiple users. The app will provide good security to the user so that the user feels secure using this app and providing information. The database will grow as number of users increase. It will load the home page quickly within 2 seconds.

Availability

The application can sustain long period of continuous usage by one or multiple users.

Priority

Quality Parameter	**Priority**	**Statement of Req.**	**Yes/No**
Compatibility	1	Should be executed in expected time	Yes
Completeness	1	Expected i/p should get relative and complete o/p	Yes
Correctness	2	Conversion should be efficient	Yes
Cost of ownership	1	Should be below 1000	Yes
Environmental	1	Should not contribute to excess use of environmental harming	Yes
Extensibility	1	Advancement acceptable	Yes
Installation Complexity	1	Easy installable in local machine	Yes
Parallel Processing	3	Concept of multitasking is utilized	Yes
Performance	1	90per. of appl. should be converted	Yes
Portability	1	Should be installable in various common platform	Yes
Regulatory	1	Supervise proper seq. execution	Yes
Reusability	1	Main purpose is reusability	Yes
Scalability	1	Should be expandable in features	Yes
Security	1	Not accessible to external world so is safe	Yes
Time to Market	1	Should be published in Apple store or android store by completion of this year	Yes
Training Complexity	1	Should be easy to learn and maintain	Yes
Usability	1	Handy to use and execute	Yes

System Analysis Model

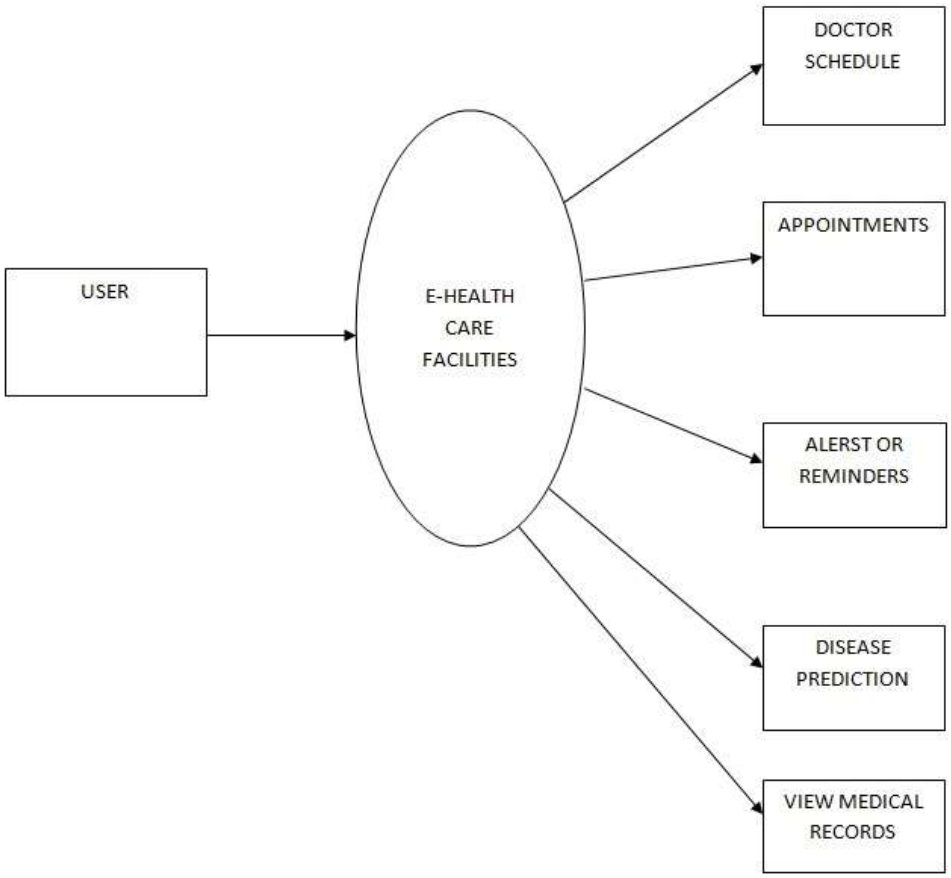

Figure 5: Data Flow Diagram-Level 0

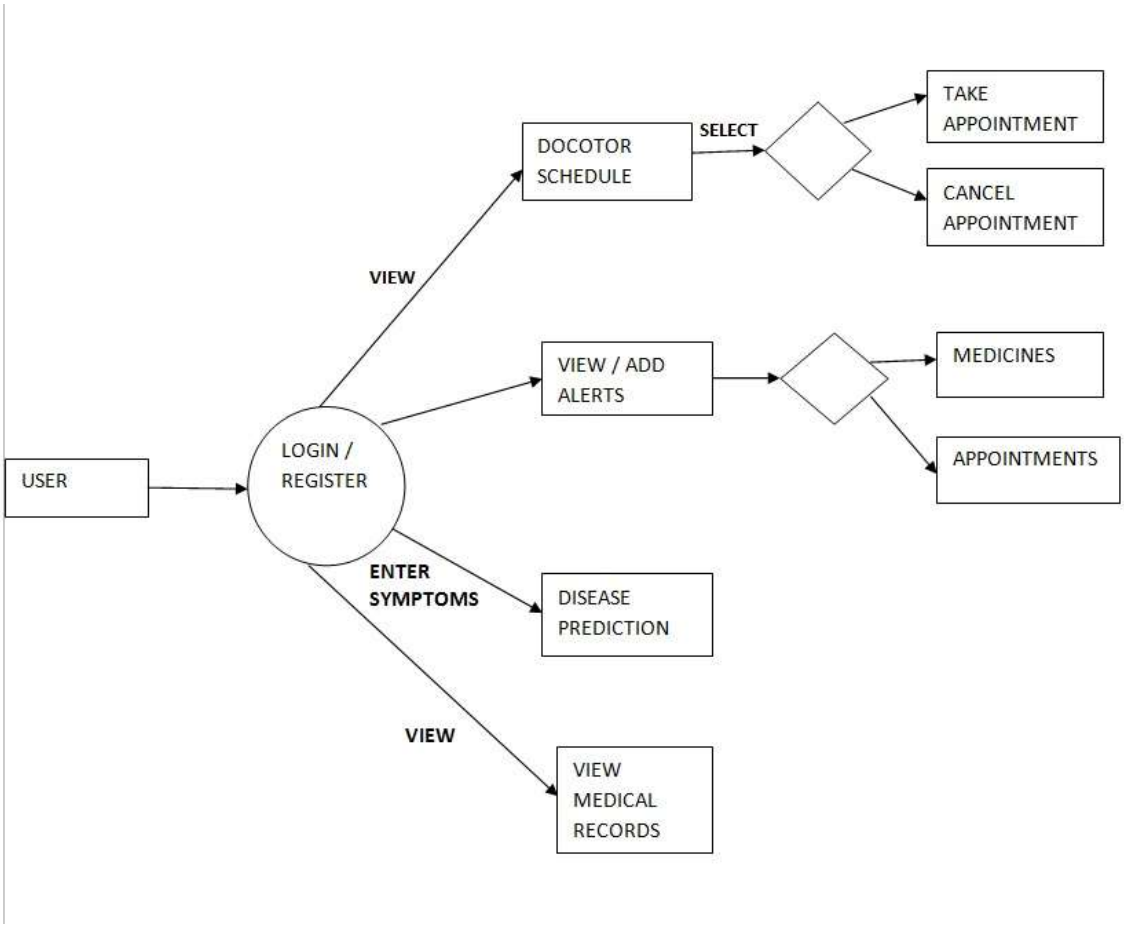

Figure 6: Data Flow Diagram-Level 1

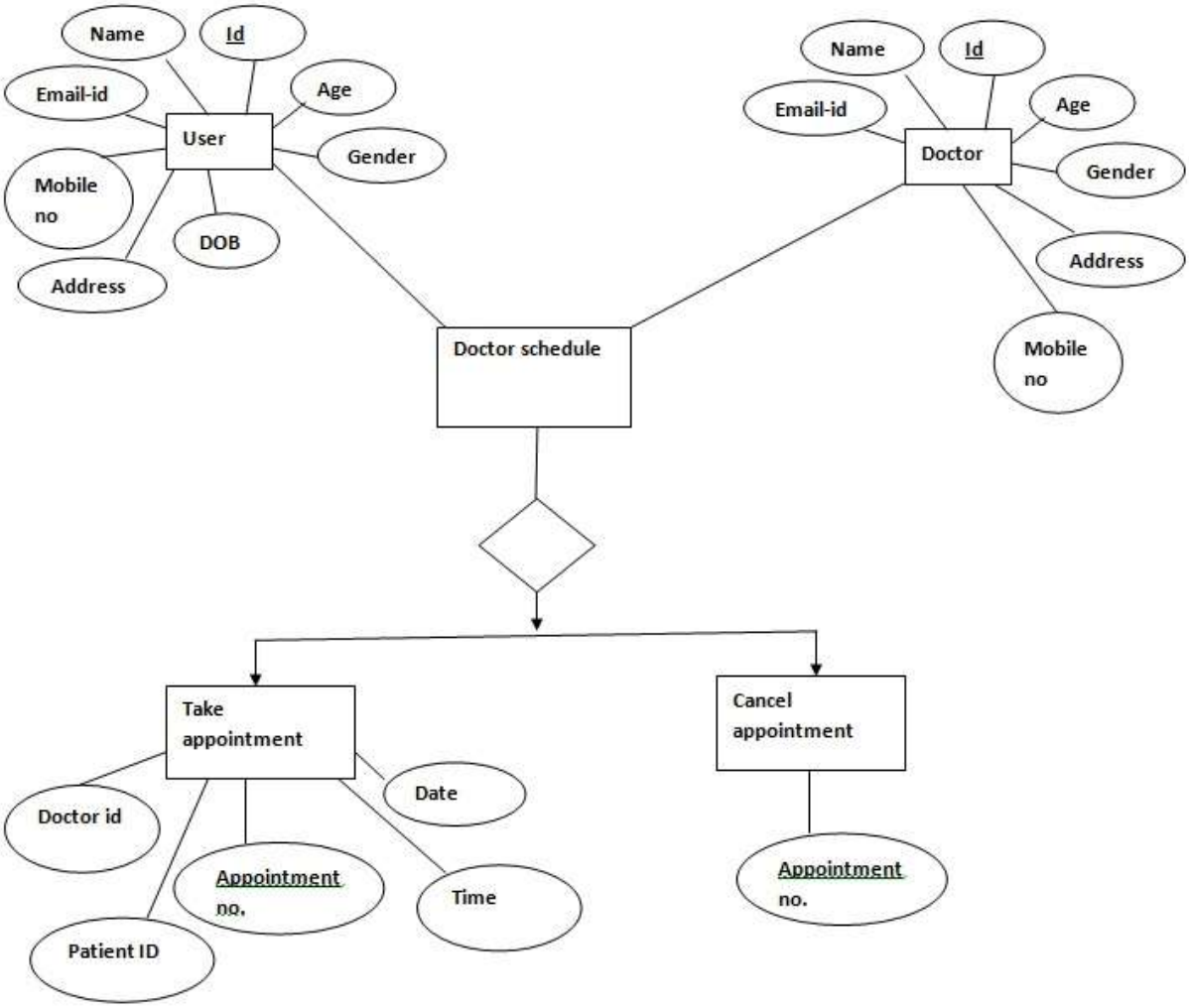

Figure 7: Entity Relationship Diagram

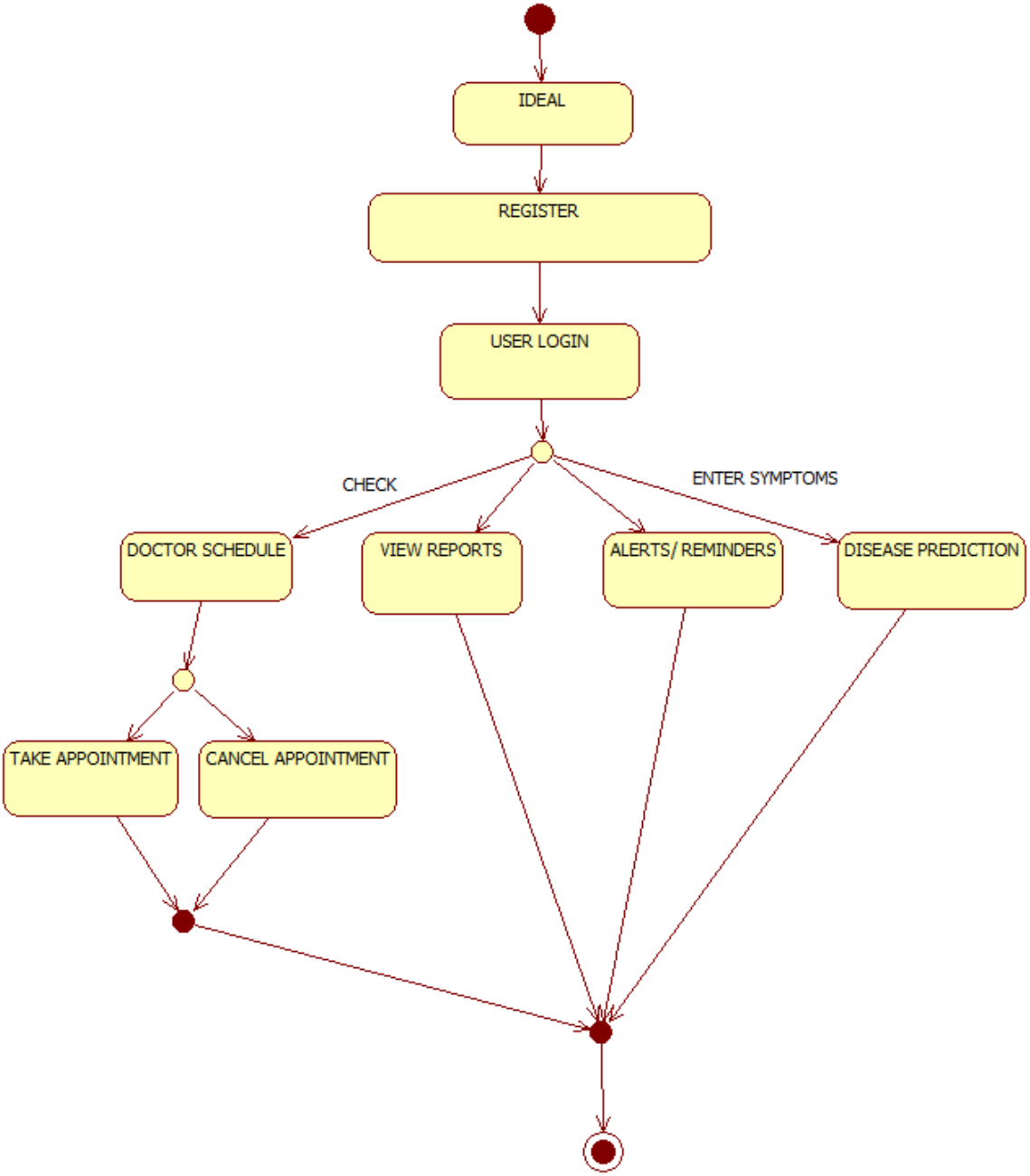

Figure 8: State Transition Diagram