
Software Requirements Specification

for

Online Medical Consultation System

Version 1.0 approved

Prepared by :-

Bratin Mondal

Swarnabh Mandal

Somya Kumar

Group: 18

Date: 27/03/2023

Table of Contents

1. Introduction.....	2
1.1 Purpose.....	2
1.2 Document Conventions.....	2
1.3 Intended Audience and Reading suggestions.....	2
1.4 Product scope.....	2
1.5 References.....	3
2. Overall Description.....	3
2.1 Product perspective.....	3
2.2 Product functions.....	3
2.2.1 Patient use cases.....	4
2.2.2 Doctor Use cases.....	6
2.2.3 Admin Use cases.....	8
2.3 User classes and characteristics.....	9
2.4 Operating Environment.....	9
2.5 Design and Implementation constraints.....	10
2.6 User documentations.....	11
2.7 Assumptions and dependencies.....	11
3. External interface requirements.....	12
3.1 User interfaces.....	12
3.2 Hardware Interfaces.....	12
3.3 Software Interfaces.....	12
3.4 Communication Interface.....	13
4. System features.....	13
4.1 Login page.....	13
4.1.1 Description and priorities.....	13
4.1.2 Stimulus/Response sequences.....	14
4.1.3 Functional requirements.....	14
4.2 Pin code search.....	14
4.3 Sending and receiving appointment requests.....	14
4.3.1 Description and priority.....	14
4.3.2 Stimulus/response sequences.....	14
4.3.3 Functional requirements.....	15
4.4 Feedback.....	15
5. Other non functional requirements.....	15
5.1 Performance requirements.....	15
5.2 Safety Requirements.....	16
5.3 Security requirements.....	16
5.4 Software quality attributes.....	16

5.5 Business Rules.....	16
6. Other requirements.....	17
7. Appendix A.....	17
8. Appendix B.....	18
9. Appendix C.....	18

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the **Online Medical Consultation System (OMCS)**. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate, and how the system will react to a user. This document is intended for the system's evaluators and developers (i.e., the students of group 18).

1.2 Document Conventions

Standard Conventions are followed.

1.3 Intended Audience and Reading Suggestions

The medical management system prototype is limited to the medical field and is a valuable tool for the medical management team and patients.

1.4 Product Scope

The proposed software system is an online medical consultation platform for patients seeking medical advice. The primary aim of this system is to simplify the process of obtaining medical consultations, appointments, and recommendations for healthcare facilities and practitioners, which has traditionally been a laborious and challenging task due to the scarcity of reliable information. The proposed system will offer ease of use, comprehensibility, and credibility while facilitating swift access to appropriate medical guidance for patients.

The system will comprise a user-friendly interface for patients to access available doctors based on location and a comprehensive list of specialisations, such as ophthalmology, dermatology, etc. Patients will also be able to update their personal information as desired. Additionally, healthcare professionals can upload their

credentials, including specialisation, schedule, location, and other relevant details, to establish connections with potential patients, schedule appointments, and receive automatic feedback.

Moreover, the system will feature a robust relational database with a comprehensive list of doctors, hospitals, clinics, and their schedules. This database will enable patients to view available healthcare providers and select those that meet their needs, ensuring effective and efficient medical consultation.

1.5 References

1. [IEEE-830](#)
2. [Class and User case diagram](#)
3. [Sample SRS Document](#)

2. Overall Description

2.1 Product Perspective

This software solution has been designed specifically for the healthcare industry, allowing hospitals to streamline their daily operational processes through automation. Implementing this technology eliminates the need for manual, repetitive tasks, thus mitigating the risk of human error. As a result, productivity increases, and hospital staff can more efficiently manage and monitor scheduling and resource allocation.

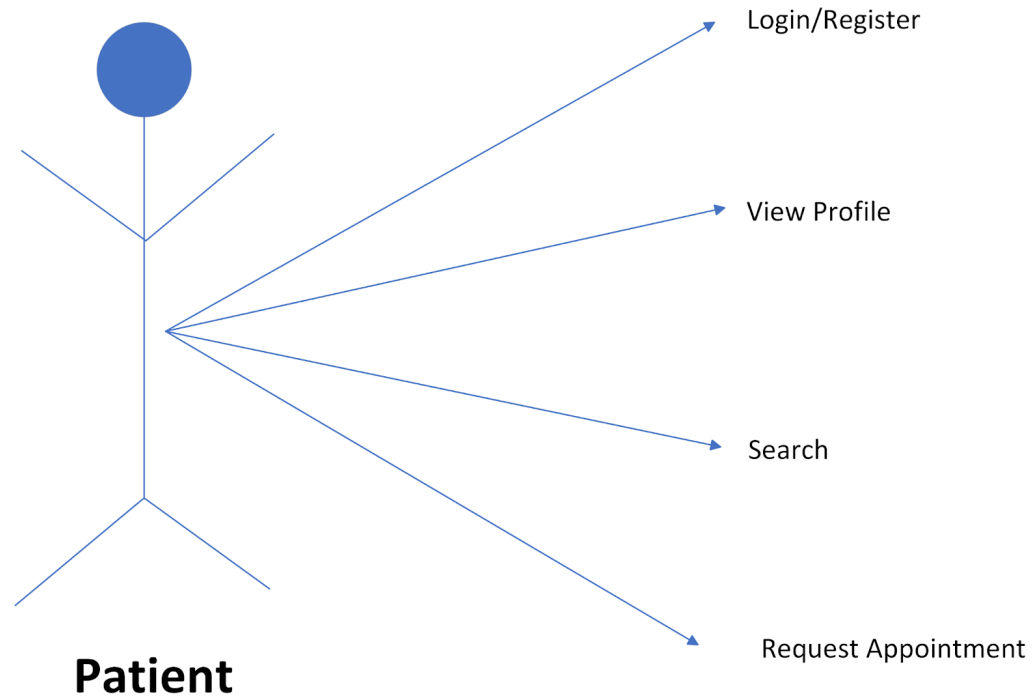
2.2 Product Functions

The following features are proposed for this application-

- Patient dashboard to check for available doctors by category
- User-input pin code to select doctors based on location
- Doctor dashboard to manage profiles and appointment slots
- Option to accept or reject patient requests
- Feedback portal to enhance patient engagement and transparency

- Automatic confirmation email sent to the patient upon doctor's approval

2.2.1 Patient Use Case



Use Case 1: Portal Registration and Login

Brief Description:

Patients can access the portal by registering and logging in with their credentials.

Step-By-Step Description:

- The patient opens the website
- For login, the patient enters their email ID and password
- For registration, the patient enters the following credentials: First name, Last name, Contact number, Email-ID, pin code, and address.

Use Case 2: Profile Viewing

Brief Description:

Patients can view their profile information

Step-By-Step Description:

- The patient clicks on the profile box at the web page's top left.

Use Case 3: Doctor Search

Brief Description:

Patients can search for specialized doctors in their location and receive a list of all doctors in the area along with relevant data like Hospital/Clinic name, etc.

Step-By-Step Description

- The patient logs into their account.
- The patient enters the pin code of their location.
- The patient enters a date for scheduling the appointment.
- The software lists available hospitals, doctors, and their specializations.

Use Case 4: Appointment Scheduling

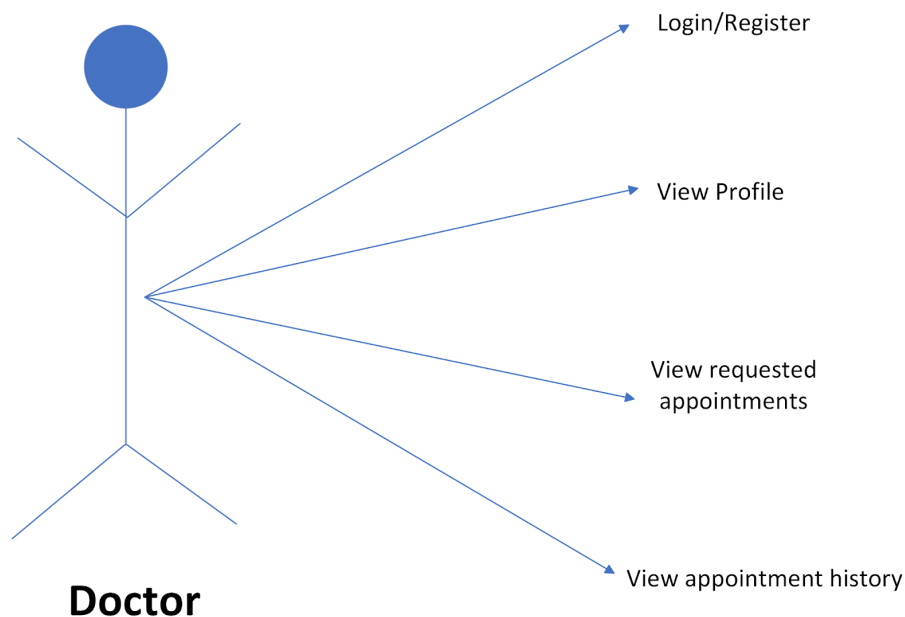
Brief Description:

Patients can schedule an appointment with a doctor from the list of available doctors after filtering available hospitals.

Step-By-Step Description:

- The patient chooses a date and time for appointment
- The patient receives a list of doctors, hospitals, and specializations.
- The patient selects a doctor that meets their requirements.
- The patient schedules the appointment.

2.2.2 Doctor Use Case



Use case 1: Portal Registration and Login

Brief description:

The doctor can access the portal by registering and logging in with their credentials.

Step-By-Step Description:

- The doctor first needs to open the website
- For login, the doctor provides his/her email ID and the password of his/her account.
- For registering for the first time, the doctor must provide the following credentials: First name, Last name, Contact number, Email-ID, pin code, specialization, Hospital/clinic of work and address.

Use case 2: Profile Viewing

Brief Description:

The doctor can view his profile information

Step-By-Step Description:

- The doctor needs to click on the profile box located at the top left of the web page

Use case 3: View pending appointments

Brief description:

The doctor can view his/her pending appointments and accept/reject new appointment requests.

Step-By-Step Description:

- The doctor enters the website first
- The doctor then clicks on the pending appointment option at the top of the website.
- The doctor is then shown a list of pending appointments and new requests.
- The doctor can then accept new appointments based on his/her convenience.

Use case 4: View appointment history

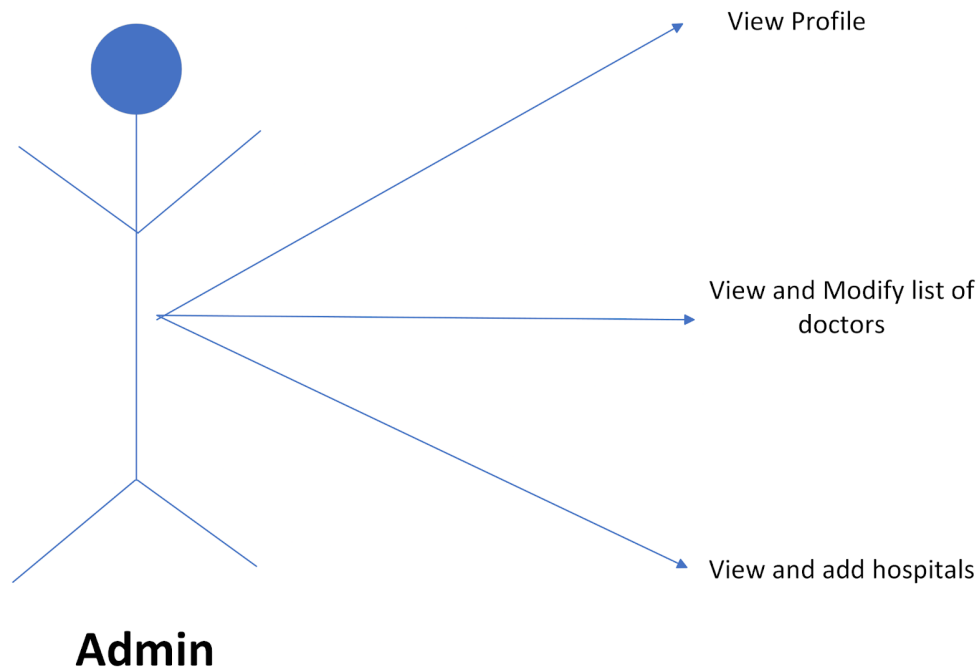
Brief description:

The doctor can view his/her appointment history.

Step-By-Step Description:

- The doctor enters the website first
- The doctor then clicks on the appointment history option present at the top of the website.
- The doctor is then shown his/her appointment history, which includes upcoming and completed appointments

2.2.3 Admin Use Case



Use case 1: Profile Viewing

Brief Description:

The admin can view his profile information

Step-By-Step Description:

- The admin needs to click on the profile box located at the top left of the web page.

Use case 2: View and modify list of doctors

Brief Description:

The admin can view a list of doctors registered to the website and also approve registration requests of new doctors.

Initial Step-By-Step Description:

- The admin after logging in can choose the list of doctors options from the top of the page.
- The website then shows the list of registered doctors and new registration requests. The admin can then choose to accept new requests and also remove registered doctors from the portal.

Use case 3: View and add list of hospital

Brief Description:

The admin can view a list of hospitals registered to the website and also approve registration requests of new doctors.

Step-By-Step Description:

- The admin after logging in can choose the list of hospital options from the top of the page.
- The website then shows the list of registered hospitals and new registration requests. The admin can then choose to accept new requests.

2.3 User Classes and Characteristics

The system will feature three distinct login roles, each with their respective responsibilities:

- **Admin:** This role will be responsible for managing the doctor database of hospitals. Specifically, Admins will be able to add or remove doctors as needed based on updates to the database.
- **Doctor:** This role will have access to a profile dashboard, allowing them to manage their profile information and review patient requests and feedback comments.

- **Patient:** This role will make appointment and consultation requests and provide feedback on their experiences.

2.4 Operating Environment

Application will be deployed as a web app

- Operating system: Windows / Linux / MacOS
- Deployed using local server
- Database can be SQLite (currently) or hosted PostgreSQL Database
- Frontend: ReactJS, SASS, CSS, Webpack (for building)
- Frontend package manager: Yarn
- Backend: Django, DjangoRestFramework
- Backend package manager: Poetry
- Emailing using Sendgrid python package
- Login feature will be accomplished using JWT (JSON Web Tokens)

2.5 Design and Implementation Constraints

The system's design follows a decoupled approach, where the frontend is developed using React and the backend using Django. This design approach provides the following benefits:

- **Flexibility:** Decoupling the frontend and back end allows for easier maintenance and updates to each part of the system without impacting the other.
- **Scalability:** Separating the front and back end allows each part of the system to scale independently based on its needs, resulting in better overall performance.
- **Reusability:** The decoupled architecture enables the reusability of components, leading to faster development times and reduced costs.

2.6 User Documentation

A README documentation (Markdown) will be included for delivering the instructions for using and deploying the app. A quick reference on the

patient's side documenting all the features and usability will be very useful.

Software Requirements Specification for OMCS Page 6

2.7 Assumptions and Dependencies

The system includes the following features and requirements:

- **Computer literacy:** Users are expected to possess basic computer skills to operate the system effectively.
- **English proficiency:** The system is implemented in English, and users are expected to be able to operate it in this language.
- **Appointment finalization:** Once an appointment has been approved by the doctor, a confirmation email is sent to the patient containing all relevant appointment details. After this point, the appointment cannot be cancelled.
- **Patient feedback:** Patient feedback will be visible to the respective doctors, allowing them to improve their services.

3. External Interface Requirements

3.1 User Interfaces

- **Admin Interface:** Based on the Django-admin portal, this interface allows the admin to manage the doctor database directly. Admins can add or remove doctors as necessary.
- **Doctor Interface:** This interface allows doctors to create and manage their credentials and available hours. Additionally, doctors can review patient requests and feedback.
- **Patient Interface:** This dashboard allows patients to view available doctors in their entered pin code and make appointment requests to their chosen doctor. Patients will receive a notification via email if their appointment request is rejected.

3.2 Hardware Interfaces

Not applicable

3.3 Software Interfaces

- **Pin Code Search:** The system shall provide a pin code search interface to enable filtering of the list of doctors based on the pin code of the hospital.
- **Doctor Login:** The system shall provide a role-based login interface for doctors to redirect them to their interface. The doctor interface shall provide the following features:
 - Profile details of the doctor
 - Available hours in the week for the doctor
 - Pending requests by patients
 - Completed requests, feedback, and comments from patients.
- **Patient Login:** The system shall provide a role-based login interface for patients to redirect them to their interface. The patient interface shall provide the following features:
 - Profile details of the patient
 - Pin code search interface for filtering out doctors based on the pin code of the hospital
 - Option to send an appointment request for a selected doctor
 - Feedback portal for providing feedback on the consultation.
- **Contact List:** The system shall display a drop-down list of contacts of doctors. The system shall show the doctor's contact details when clicking a name.
- **Admin Login (Superuser Mode):** The system shall provide an admin login interface to redirect the admin to the superuser mode. Through this portal, the admin shall have access and control over the doctor's portal and change it according to the need.

3.4 Communications Interfaces

- **Login Management:** The system shall manage login credentials by implementing password encryption in the database.
- **Patient Portal:** Patients can make appointment requests to specific doctors by logging into their portal.
- **Doctor Portal:** Doctors shall be able to view the appointment requests made by patients and accept or reject them accordingly.
- **Notification and Feedback:** The patient shall receive notifications via

email regarding the status of their appointment request. Once the appointment is completed, the patient can provide feedback through the portal.

- **Implementation:** The software shall be implemented using the HTTP protocol.

4. System Features

4.1 Login Page

4.1.1 Description and Priority

The Online Medical Consultation System (OCMS) manages the login process for the Admin, Doctor, and Patient roles, allowing them to operate independently without any interference from one another. The system shall store user passwords in an encrypted form using a hashing algorithm to ensure data security. The Admin Login, inherited from the Django-admin framework, grants the user direct access to the database for modification purposes. The Doctor and Patient login processes are based on role-based login mechanisms and shall redirect users to their respective interfaces upon successful authentication.

4.1.2 Stimulus/Response Sequences

- Admin will have the read and write access to doctor database
- Doctor and patient login will redirect to respective interfaces

4.1.3 Functional Requirements

Whole login will be extensively managed by Django and its dependencies hence Python and Django will be required.
Software Requirements Specification for OMCS Page 8

4.2 Pin code search

The list of doctors will be filtered according to the latest database based on entered pin code by the patient in their portal.

This will be managed by an SQL query.

4.3 Sending and receiving appointment requests

4.3.1 Description and Priority

This feature will be used by the patients and doctors for booking/approving / cancelling appointments.

4.3.2 Stimulus/Response Sequences

- **Appointment Booking:** The system shall allow patients to initiate the appointment booking process by requesting the relevant doctor.
- **Request Management:** The system shall enable the doctor to view incoming appointment requests from patients via their portal to manage appointment scheduling.
- **Request Response:** The system shall allow doctors to accept or reject incoming appointment requests and send an appropriate message to the patient for effective communication.

4.3.3 Functional Requirements

This needs a database for managing the data of requests. So a database management system is needed, which will be implemented with Django.

Required: PostgreSQL database

4.4 Feedback

- **Feedback and Comments:** The system shall enable patients to provide feedback and comments after the successful completion of an appointment for better service quality.
- **Database Integration:** The system shall integrate with the database to store and retrieve patient feedback and comments.
- **SQL Query:** The system shall use a SQL query to fetch the corresponding feedback of a specific doctor, allowing for efficient data management and analysis.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- **Network Requirements:** The system shall require a high-speed internet connection to ensure optimal website loading and functionality.
- **Browser Compatibility:** The system shall support access from any browser with a recent version to provide a flexible and user-friendly experience.

5.2 Safety Requirements

- The user must remember their login credentials.
Software Requirements Specification for OMCS Page 9

5.3 Security Requirements

- **User Information Storage:** The system shall store patient and doctor information in their respective portals, with password-protected access to ensure data security.
- **Role-Based Login:** The system shall provide separate logins for doctors and patients to ensure appropriate access and functionality for each role.
- **Doctor Management:** The system shall allow only the admin to add or remove doctors from the system, ensuring accurate and authorized updates to the database.

5.4 Software Quality Attributes

- **Reliability:** The software shall support multi-user access for patients and doctors to ensure reliable operation.
- **Maintenance:** The system shall allow administrators to access and modify doctor portals according to changing requirements to support efficient maintenance.
- **Portability:** The software shall be accessible on various devices such as PCs, tablets, and mobiles, ensuring ease of use and portability.
- **Availability:** The OMCS website shall be available to users throughout the day, providing uninterrupted service access.
- **Robustness:** The website shall undergo thorough testing before deployment to ensure robustness and to eliminate any possibility of bugs that may impact user experience.

5.5 Business Rules

The OMCS software(website) can be used by doctors and patients and managed by the admins (like Medical Store etc.) to which it is sold.

6. Other Requirements

Database:

A database is very much required for purposes like storing data of Patients, Doctors. Hence, it stores a lot of data in it, so it is suggested that the platform in which it is hosted has sufficient memory reserve.

Appendix A:

Glossary

Glossary

- SQL: Structured Query Language
- CSS: Cascading Style Sheets

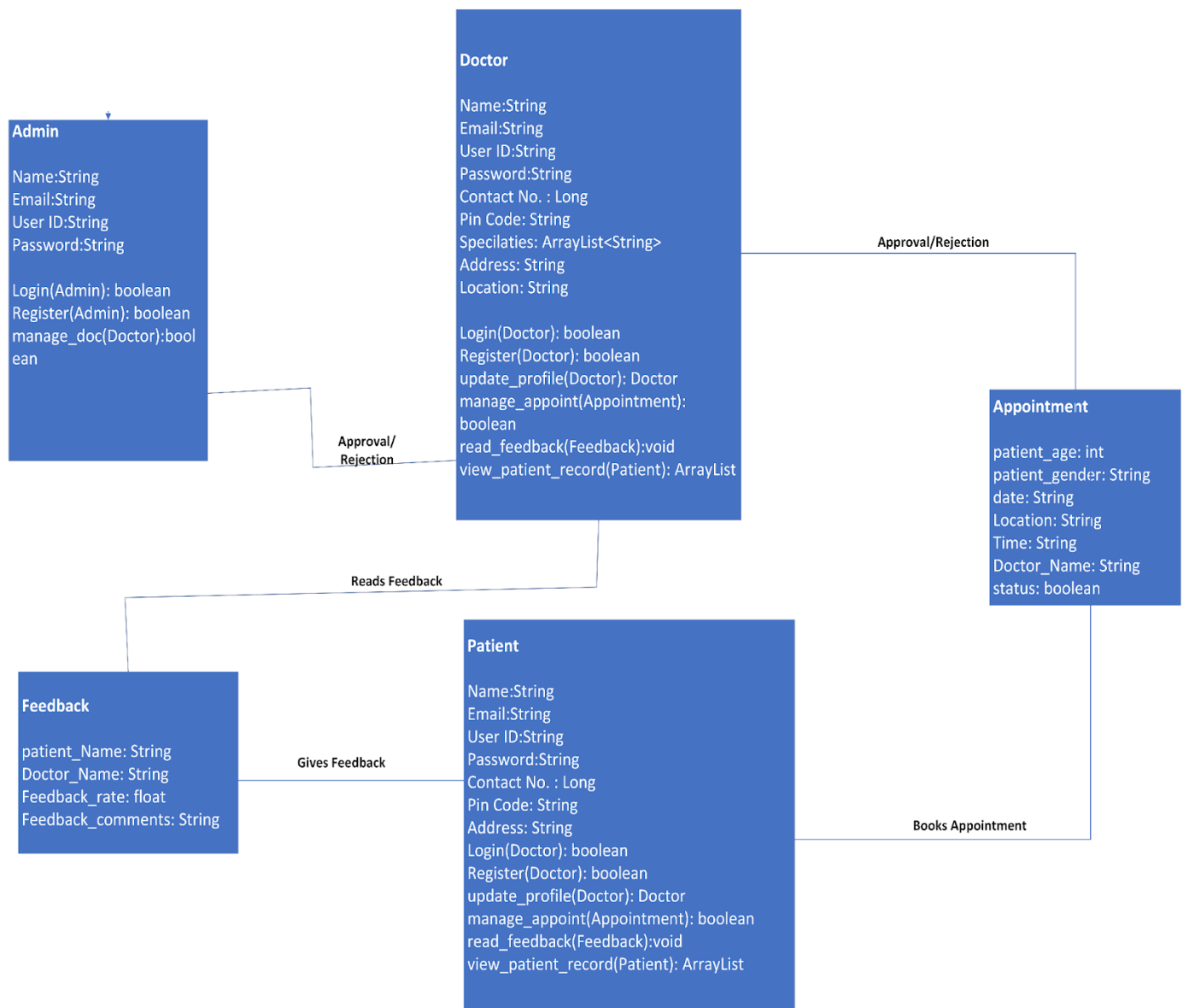
- SASS: Syntactically Awesome Style Sheets
- JS: JavaScript
- DB: Database

Software Requirements Specification for OMCS Page 10

Appendix B:

Analysis Models

Class Diagram:



Appendix C:

To Be Determined List