Software Requirements Specification

for

Online Doctor Appointment System (ODAS)

Version 1.0.1 approved

Prepared by Kayra Polat, Baturalp Kızıltan,

Ted University

02.01.2022

Table Of Contents

Revision History

Name	Date	Reason for Changes	Version
Kayra Polat Baturalp Kızıltan	27.12.2021	Detailed changes had to be made to the user interface and functional and non-functional requirements.	ODAS 1.0
Kayra Polat Baturalp Kızıltan	02.01.2022	UML Class diagram extension	ODAS 1.0.1

1. Introduction

1.1 Purpose

This document describes the online doctor appointment system. It also describes the general features, requirements, user features, system features and some limitations This system, which will enable every citizen to make an appointment easily through a website, works through several stakeholders. These; system administrator, doctor, patient, and developers. However, it is not limited to these. At the same time, this document not only explains the objectives at all stages of the project, but also details the functional and non-functional requirements.

For developers, this document is very important. All the information in this document explains everything developers need to do.

1.2 Document Conventions

While preparing this SRS document, Microsoft Word became the program we use. The font of the descriptions is set as 12. Times New Roman is used as the type of the font. The font of titles is set as 18 and bold. Also, there are subtitles which are divided by titles. The font of the subtitles is set as 14.

REQ	Requirement
SRS	Software Requirements Specification
ODAS	Online Doctor Appointment System
FAQ	Frequently Asked Questions

The abbreviation of some words is given in right:

1.3 Intended Audience and Reading Suggestions

This SRS document has been created for the stakeholders involved in the project. These stakeholders are doctors, patients, developers, project managers, testers, and SRS authors. The document consists of many parts. First part introduction, second part overall description, third part external interface requirements, fourth part system features, fifth part other nonfunctional requirements, sixth part other requirements and seventh part UML Diagrams. Reading the document in detail ensures that all stakeholders have accurate information about the project. Therefore, it is not recommended to start the project without examining every stage of the document.

1.4 Product Scope

The online doctor appointment system (ODAS) will make a great contribution to the field of health. With the technological developments, it is no longer important for a patient to go to the hospital or make an appointment by calling the hospital by phone. Through this web application, the patient will be able to make an appointment at the desired time from the doctor of the hospital he/she wants. Our main goal is to ensure that both the doctor and the patient get rid of the confusion they experience during the appointment process. The doctor will be able to determine the hours which he is available and get information about his appointments from his special profile screen. The patient will also be able to open a user account very quickly via the web application and choose the most suitable time for his appointment. Our goal is to make this application easy to use by many users. However, our other goal is to make this app the only app used by everyone.

1.5 References

1) CMPE 313 SRS Template File: We use this SRS-template-IEEE (The Institute of Electrical and Electronics Engineers) file to apply SRS structure.

2. Overall Description

2.1 Product Perspective

ODAS is a system that allows patients to make an appointment. However, it does not only enable patients to make an appointment. Doctors also determine the appointment times from this application. The system administrator has access to the database. It ensures that all necessary patient/doctor/appointment information is stored and managed in the database on a regular basis. A simple schematic representation of the system is shown in Figure 1.

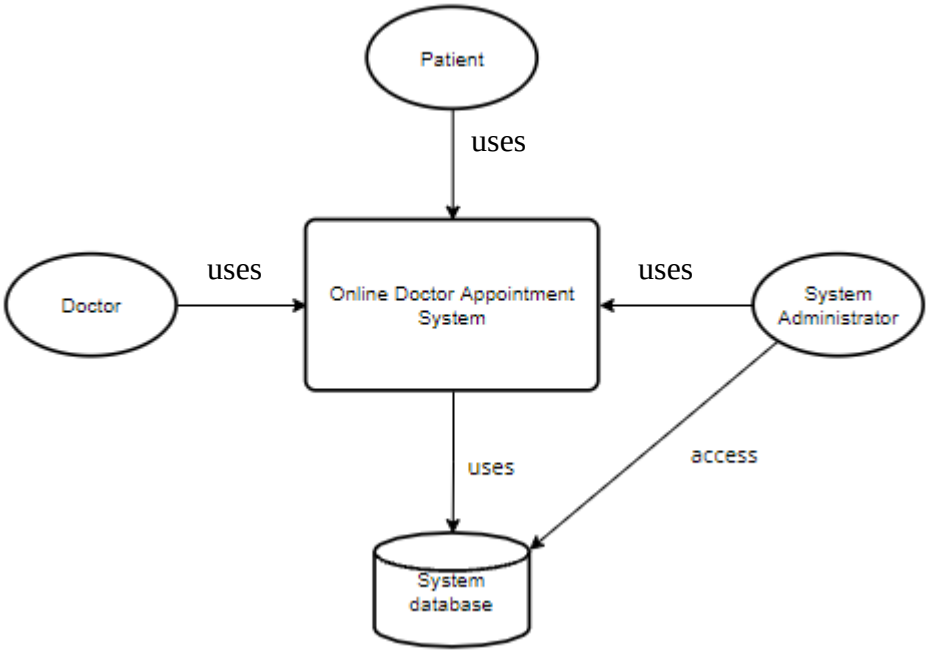
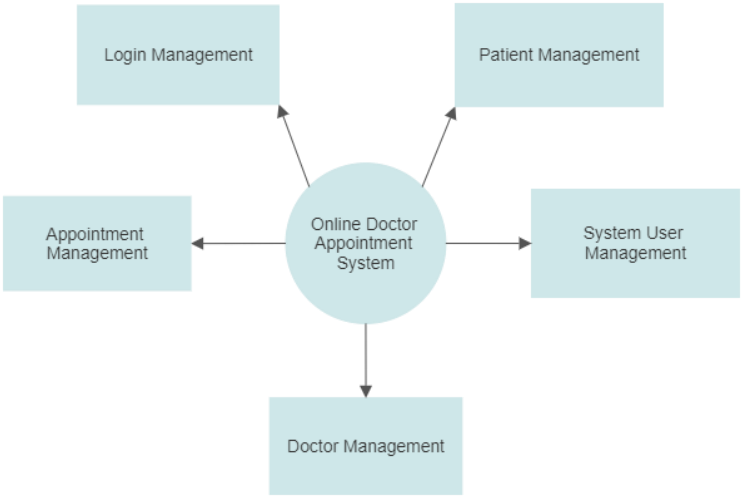


Figure 1: This figure provides a preview of how the system works. The patient actor accesses and uses the application. The doctor also accesses and uses the application in the same way. The system administrator also has access to the application but can also access and manage the system's database.

2.2 Product Functions

The basic functions that will be created during the design process of our application provide us great benefits. You can see a top-level data flow diagram below. To increase the understandability of our application, we will explain through this diagram. At this stage, it should be noted that this system has different functions.



a. Log in management is provided uninterruptedly so that users can easily and quickly log in.
b. Patients can manage all the information necessary for the appointment. However, all patient record information is managed by the system administrator.
c. Appointment management is provided by both doctors and patients. Doctors can set appointment times and change or delete them when they are available. Patients can also manage their appointments as they wish (make an appointment, change, delete).
d. Only doctors who can make an appointment are listed in the system. Since every doctor of a hospital may be closed to appointments at that moment, the system administrator provides the doctor's management according to such situations.
e. Although the system administrator has the main access to the application, there are also system users working under the system administrator. System users play a role in the visual design of the web application. The system administrator manages everything as well as provides system user management.

2.3 User Classes and Characteristics

This system will be used for almost all hospitals. The main users of this system are doctors and patients. In addition, since the management of the system is an important factor, the system administrator is among those who use this system. Our application is a user-friendly application as it will appeal to every person. For people who are far from computer language, the whole working logic of the application can be taught with a very simple training.

- **Patient:** Patients are people who want to see doctors.
  - They may be familiar with internet applications.
  - They may not be familiar with internet applications. They must learn it.
  - Wants to be able to open an account in the application and see and manage their own personal information.
  - If he/she has not made an appointment so far, he/she wants to make an appointment, if he/she has made an appointment, he/she wants to be able to manage (change, delete) his/her appointments as he/she wants.

- **Doctor:** Doctors are people who have undergone long-term training, have a lot of knowledge and are responsible for treating patients.
    - Previously used software such as patient record management.
    - Little knowledge of Internet applications.
    - They are the people who can control the appointments they give.
    - Wants to change the available time intervals according to the situation.
    - Want to be able to make changes to their user accounts.
- **System Administrator:** A person who has general knowledge of the software, is an expert in storing and managing data in the database and has a high level of software knowledge.
    - It keeps user information entered by patients and doctors in its database. It uses this information when necessary.
    - It ensures that the appointment information is kept correctly in the database and shows it to the patient and the doctor on a regular basis.

2.4 Operating Environment

The software will be platform-agnostic in terms of operating system, is able to run on either Unix-like (macOS, Linux) or Windows. Because the software will be containerized using Docker to ensure cross-platform operability. However, Windows Subsystem for Linux (WSL) layer is required for the special case of Windows OS in terms of full compatibility with the latest Docker engine. Additionally, our programming technologies are cross-platform supported themselves, therefore they can optionally work without virtualization too. From the infrastructure perspective, the software is planned to serve on bare-metal hardware platform.

2.5 Design and Implementation Constraints

The most important requirement of the application is that it must work 24 hours a day. Patients should be able to make an appointment at any time of the day. Therefore, a fully functional and stable internet-server connection is required. Since this web application will run on the web, providing this connection correctly to the user and reducing the possibility of disconnection is an element that the software developer should pay attention to. However, the interface of this web application needs to be user-friendly because users find it difficult to quickly get things done in a complex interface.
The application must be running fully functional and server connection must be provided until the specified deadline.

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

Assumptions are:

- Coding must be free of errors.
- The system should be user-friendly, making it simple for people to utilize.

- All users' login information should be saved in an internet-accessible database.
- The system should also allow for quick access.
- The system should make it simple to create, simulate, and change appointments, as well as facilitate quick transactions.
- The appointment system should be available 24 hours a day, seven days a week.
- Users should be able to log in from any computer with an Internet connection.

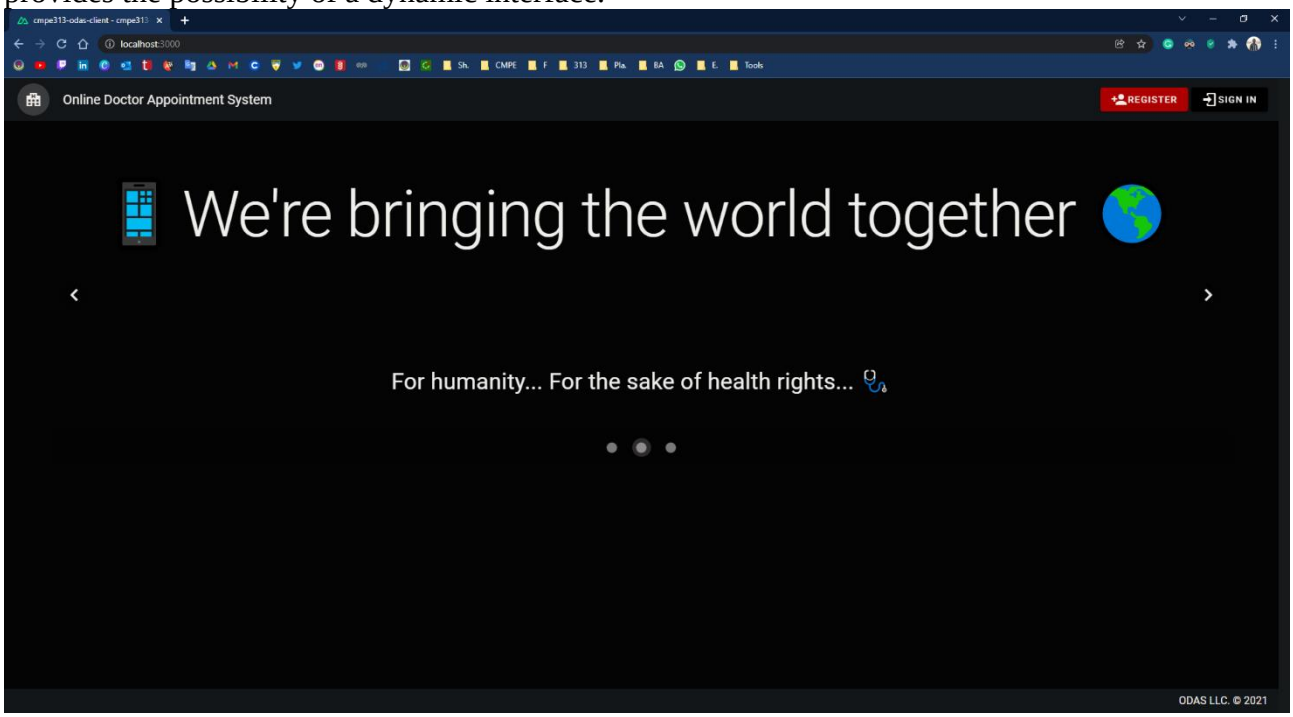
Dependencies are:

- Specialized hardware and software to execute the product.
- The project will be built and run based on a set of requirements and standards.
- The product's end-users (administrators) must have a thorough comprehension of it.
- General information should be kept in the system.
- The Appointment System should keep all users' information in a user-accessible database.
- Any pertinent appointment updates will be recorded in the database and entered data must be accurate.

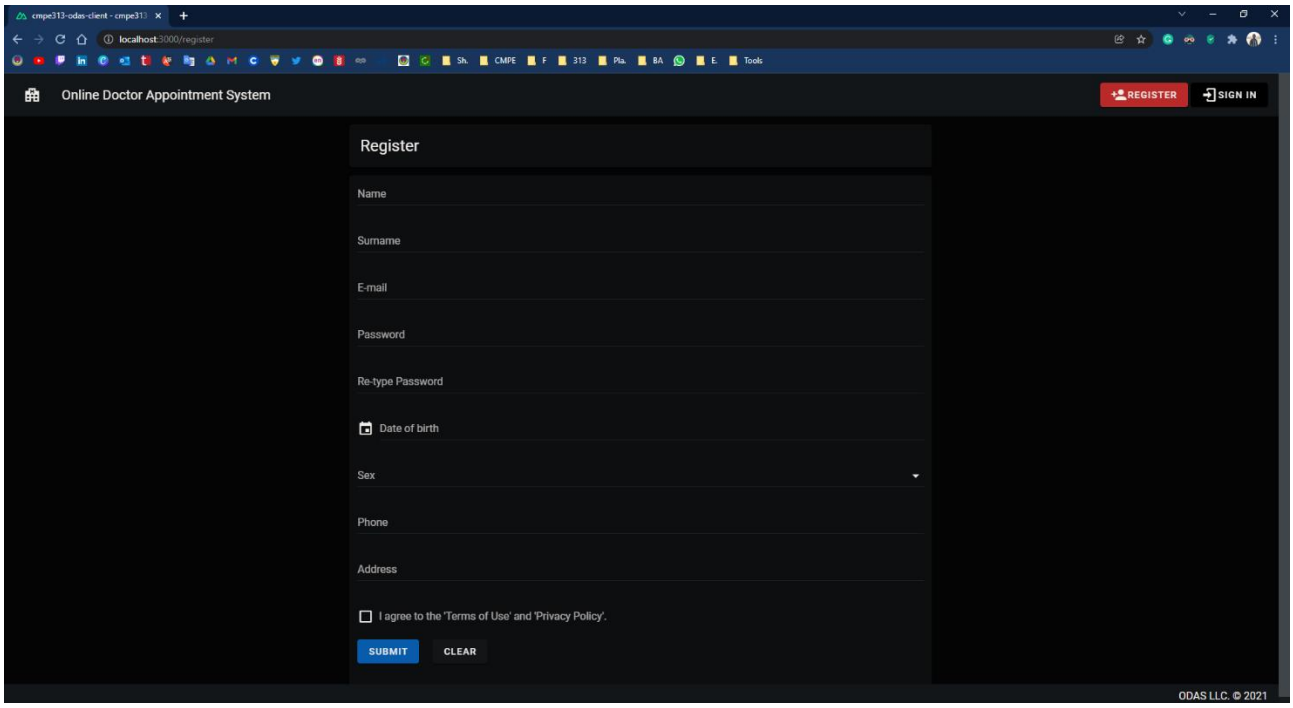
3. External Interface Requirements

3.1 User Interfaces

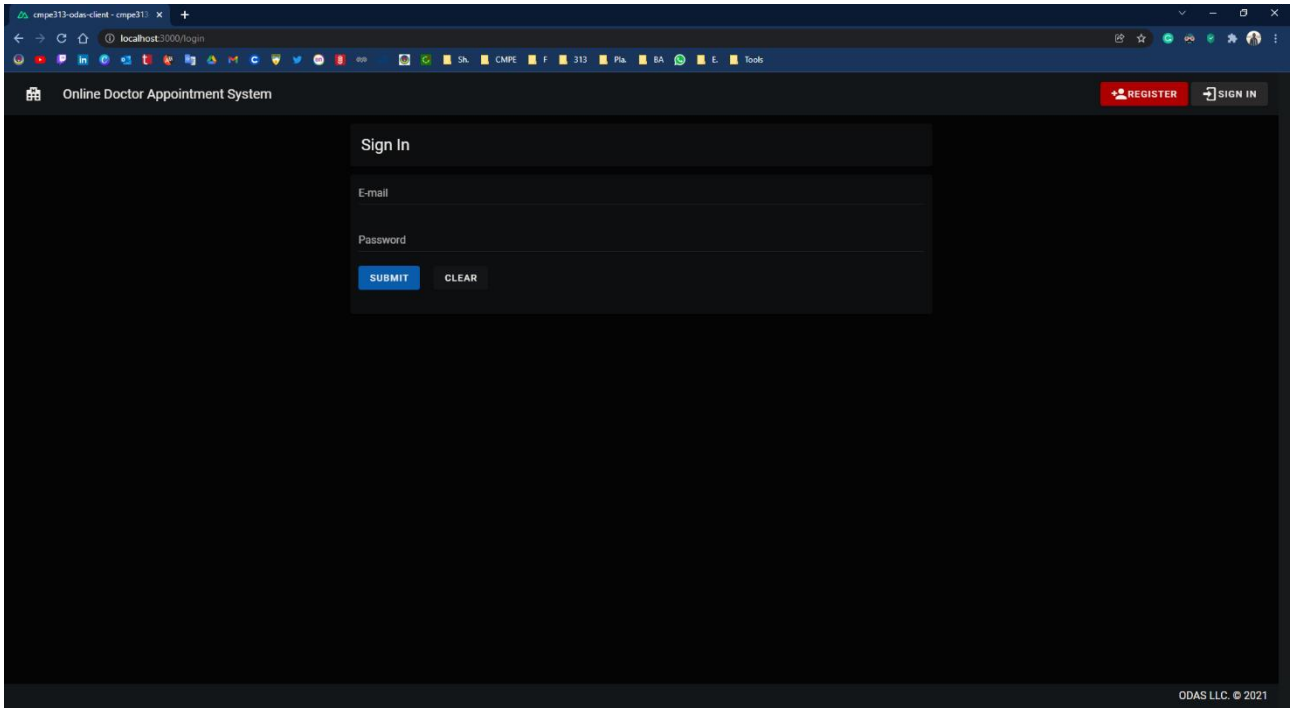
ODAS has a project design with its own user interface. The user interface designs of the web application were designed over the "Nuxt.js"(Single Page Application(SPA). This application provides the possibility of a dynamic interface.



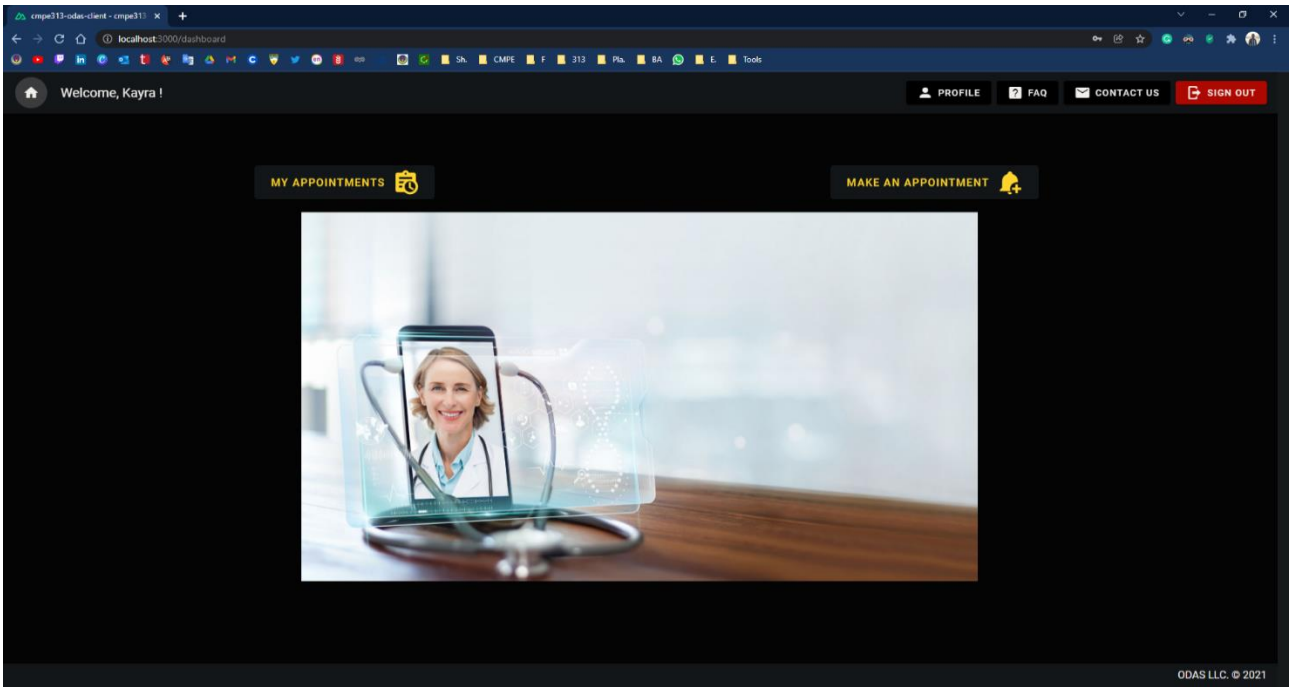
[1] Home Screen: Users can login or register from the main home screen of the application.



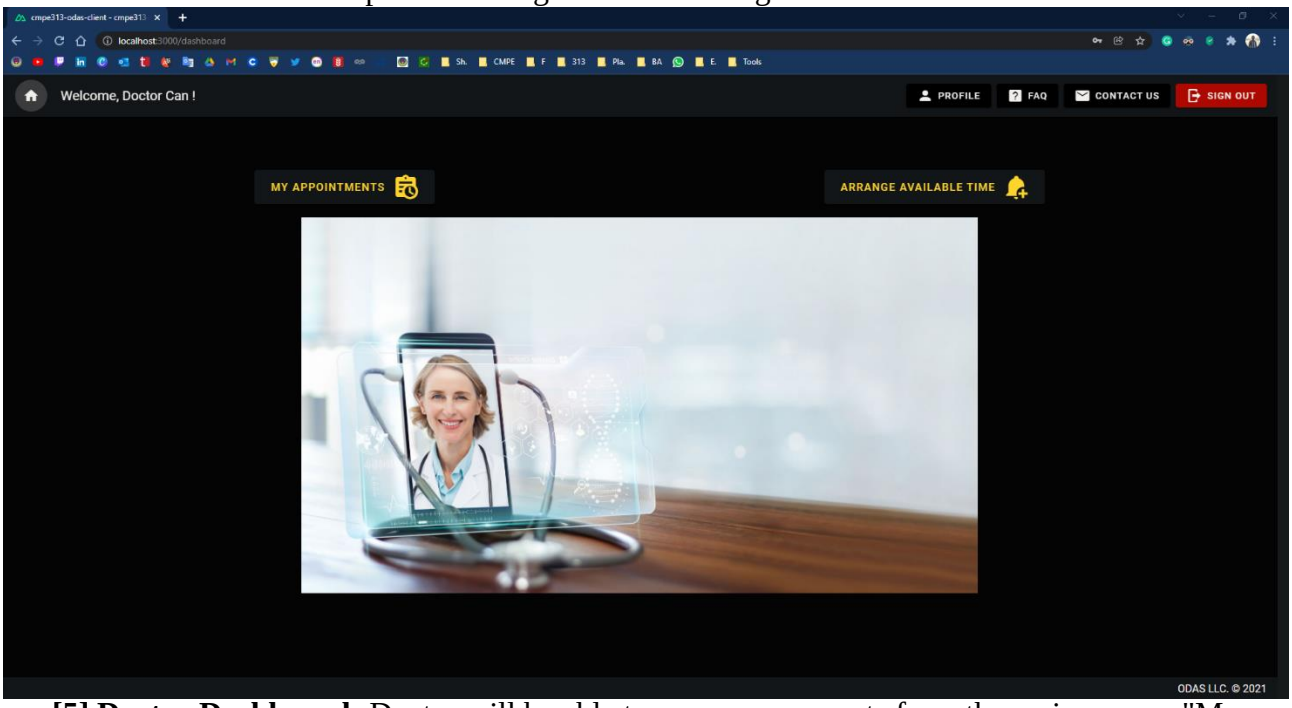
[2] Registration Screen: The part where data such as username, surname, residence address, telephone number, e-mail address, gender, date of birth are entered. Password is created here.



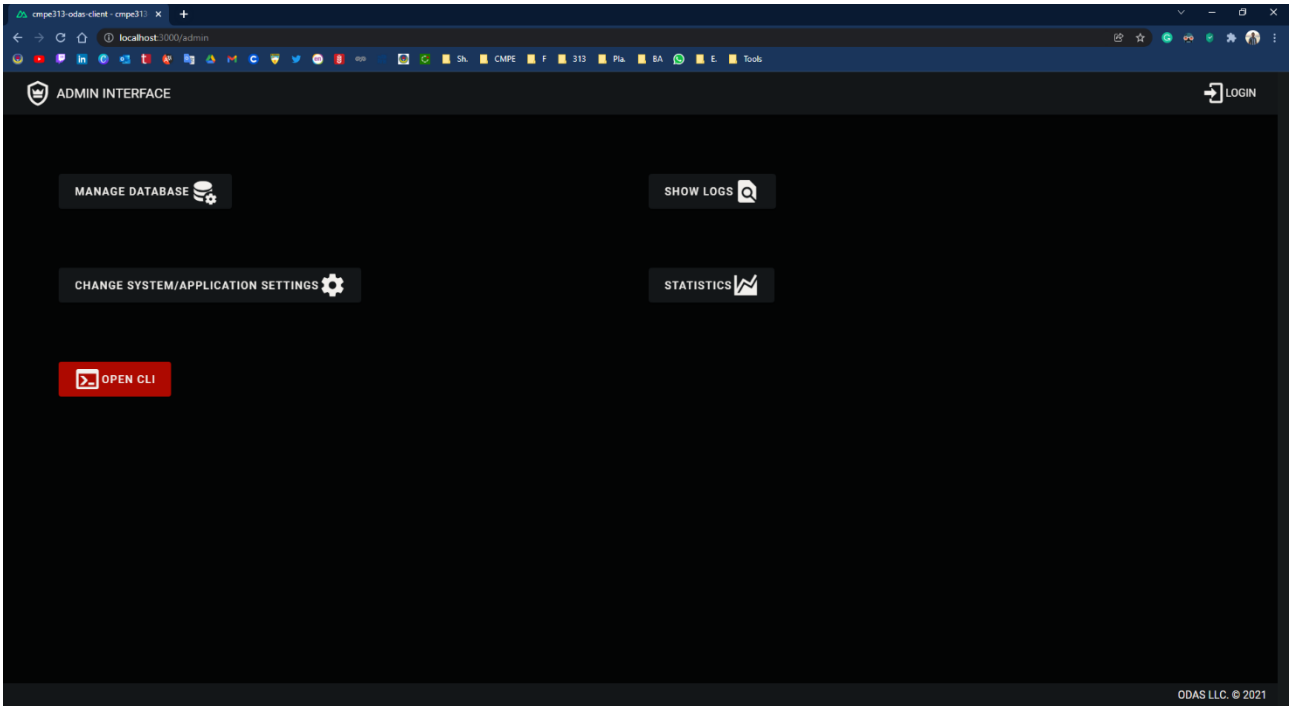
[3] Login Screen: Username and password writing part is shown here. On the screen of the log in, there are two fields where the email is entered, and the password is entered. After filling in the blanks, the user logs in according to their status.



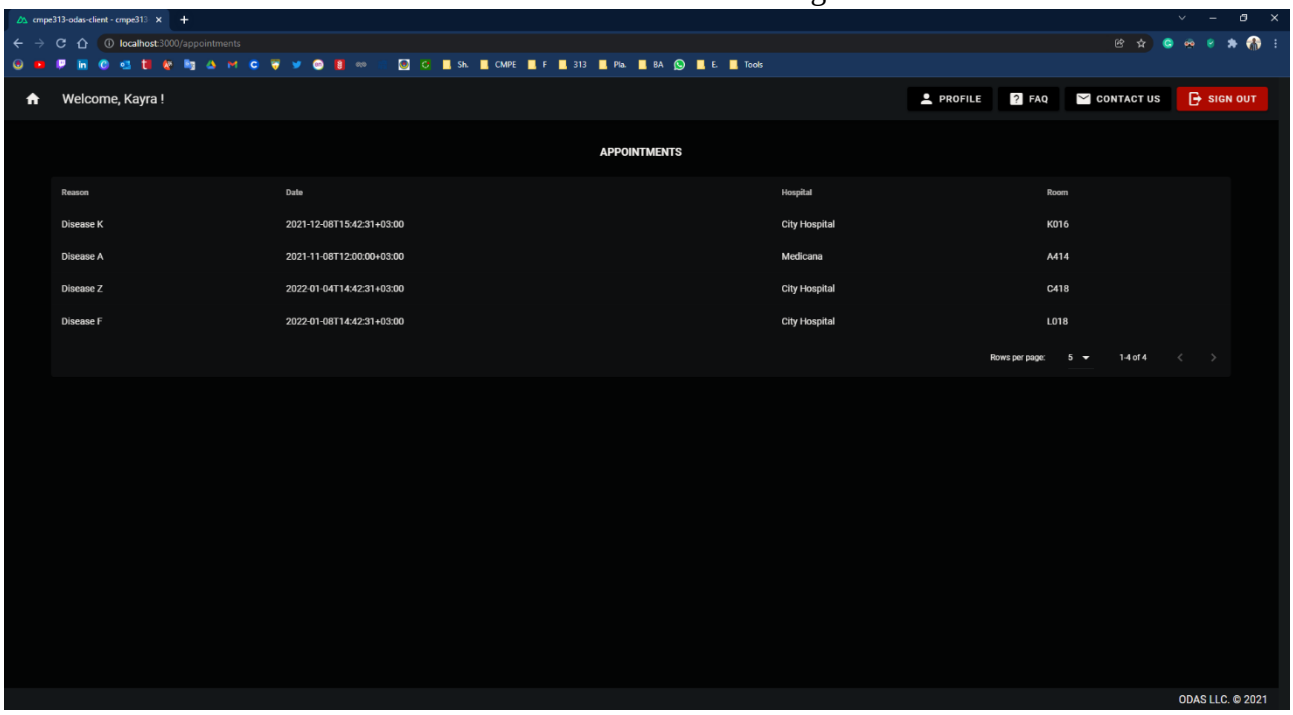
[4] Patient Dashboard: Patients will be able to access many parts from the main screen. "My appointments", "Make appointment", "Contact us", "Profile", "Frequently Ask Questions" sections will appear here, and the user will be directed to the relevant page with one click. At the same time, it is possible to sign out with the sign out button.



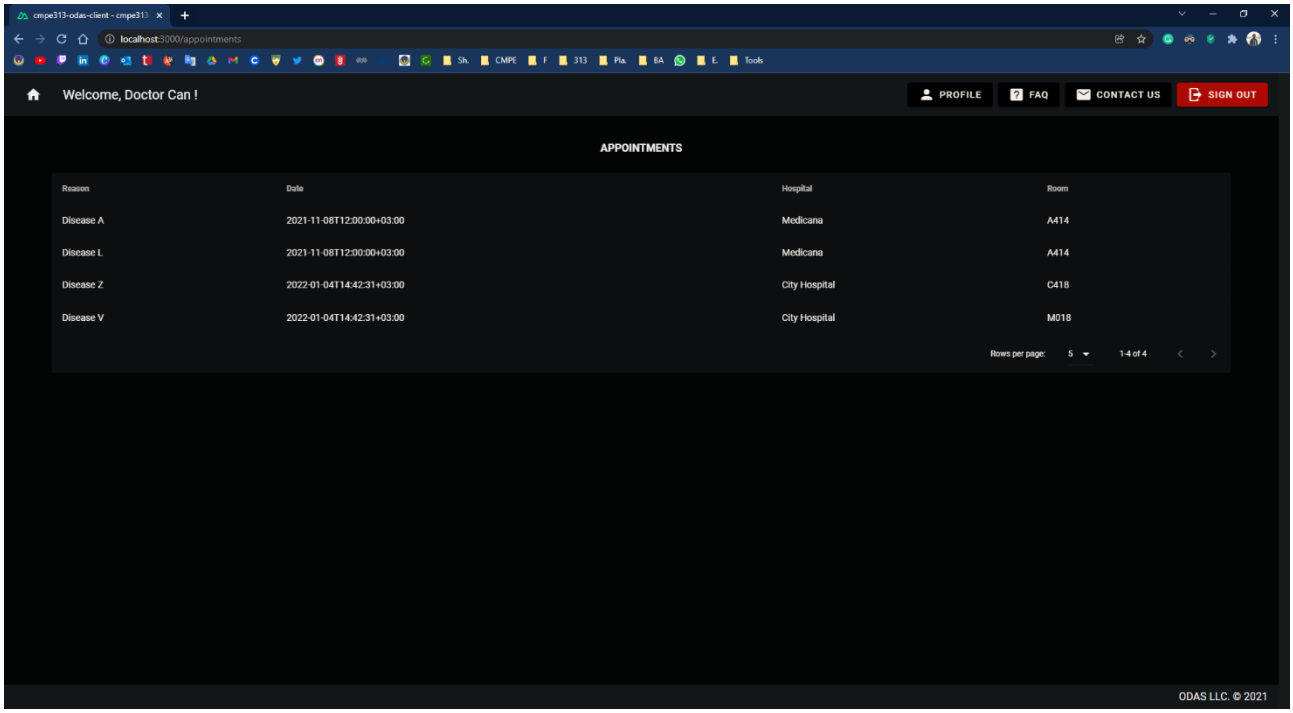
[5] Doctor Dashboard: Doctor will be able to access many parts from the main screen. "My appointments", "Arrange Available Time", "Contact us", "Profile", "Frequently Ask Questions" sections will appear here, and the user will be directed to the relevant page with one click. At the same time, it is possible to log out with the log out button.



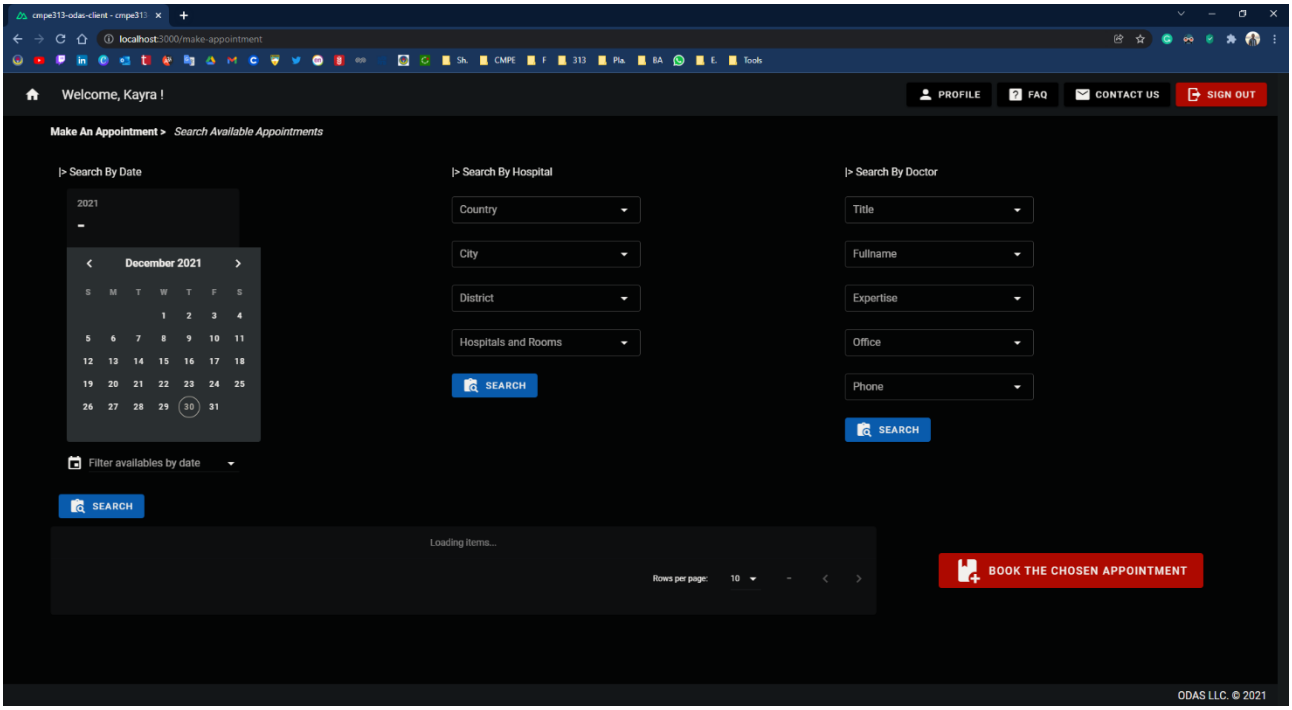
[6] Admin Screen: The system administrator can log in to a different interface of the web application, log in to the database where user data is kept, and make some changes on the website. During website changes, users are warned that the system is under maintenance. Admin can view user statistics and view log records.



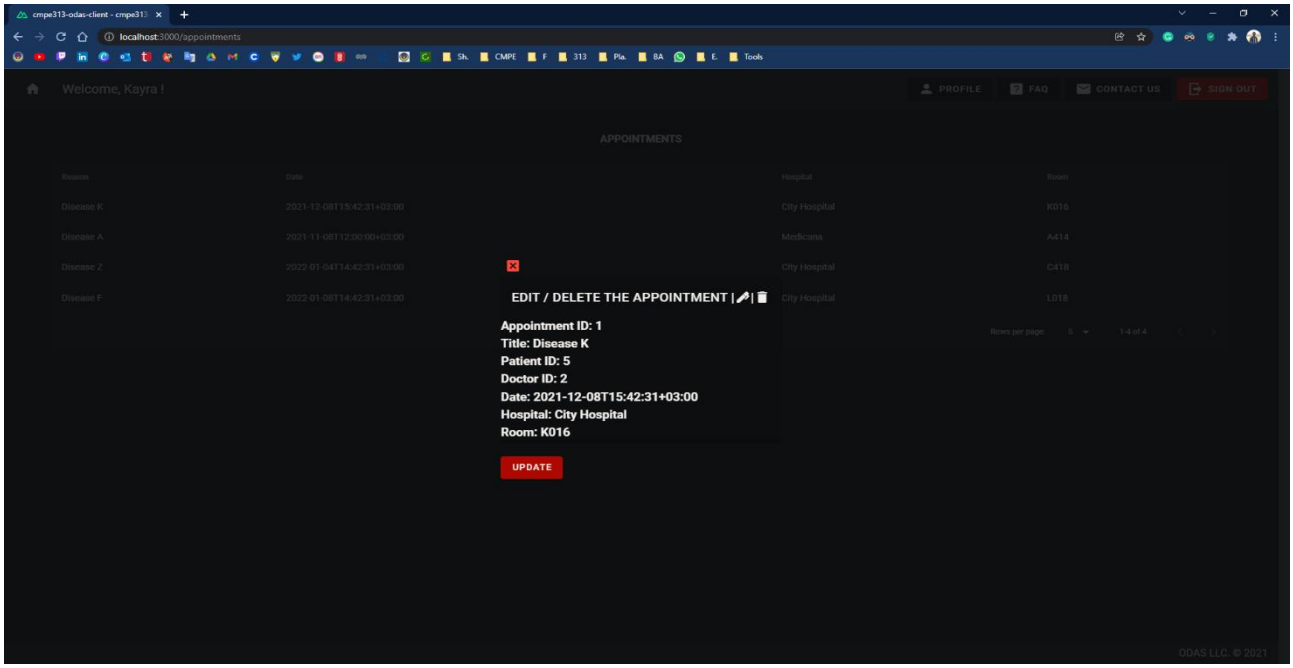
[7] My Appointment Screen for Patient: Existing appointments appear here. Patient can click on each appointment to see its details.



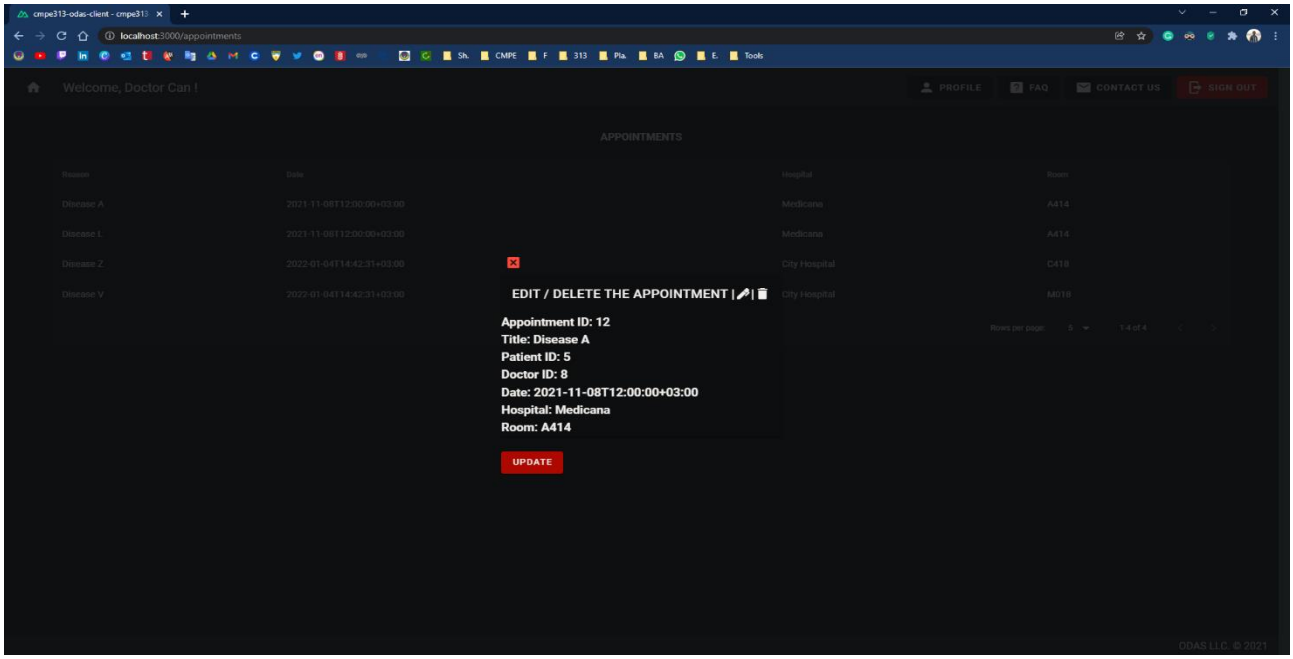
[8] My Appointment Screen for Doctor: Existing appointments appear here. Doctor can click on each appointment to see its details. Doctor can cancel the appointment with the cross button next to each appointment.



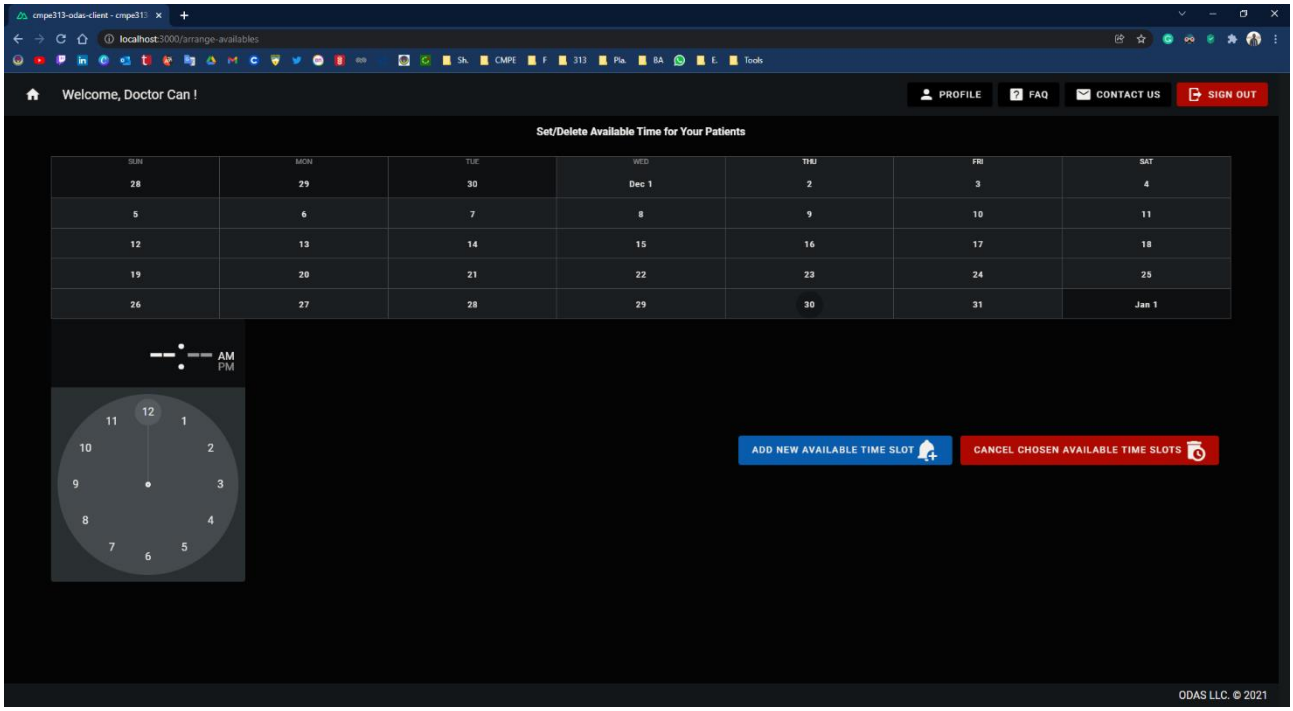
[9] Search and Make an Appointment Screen: Doctor, hospital, date, and time shown here. The pre-selected hours will be crossed out and the button cannot be pressed by the patient. When the patient clicks the "submit" button, an appointment is created. Cancel the operation with the "HOME ICON" button and return to the patient dashboard.



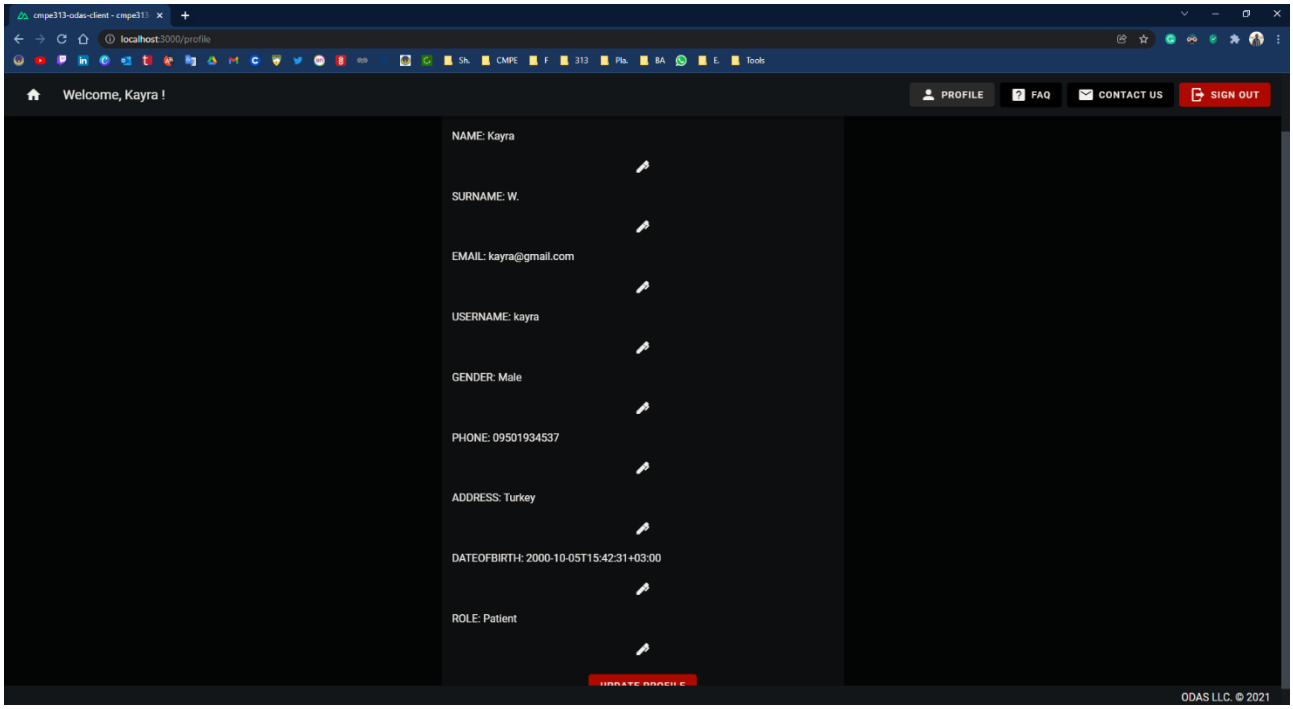
[10] Modify/Cancel Appointment for Patient Screen: The patient can make changes on his appointments thanks to this interface screen. When clicks an appointment, page opens to make some changes or see the relevant details. When the "Update" button is pressed, the operation takes place. At the same time, users can cancel their appointments through this screen. Appointments are canceled by pressing the "Trash" icon.



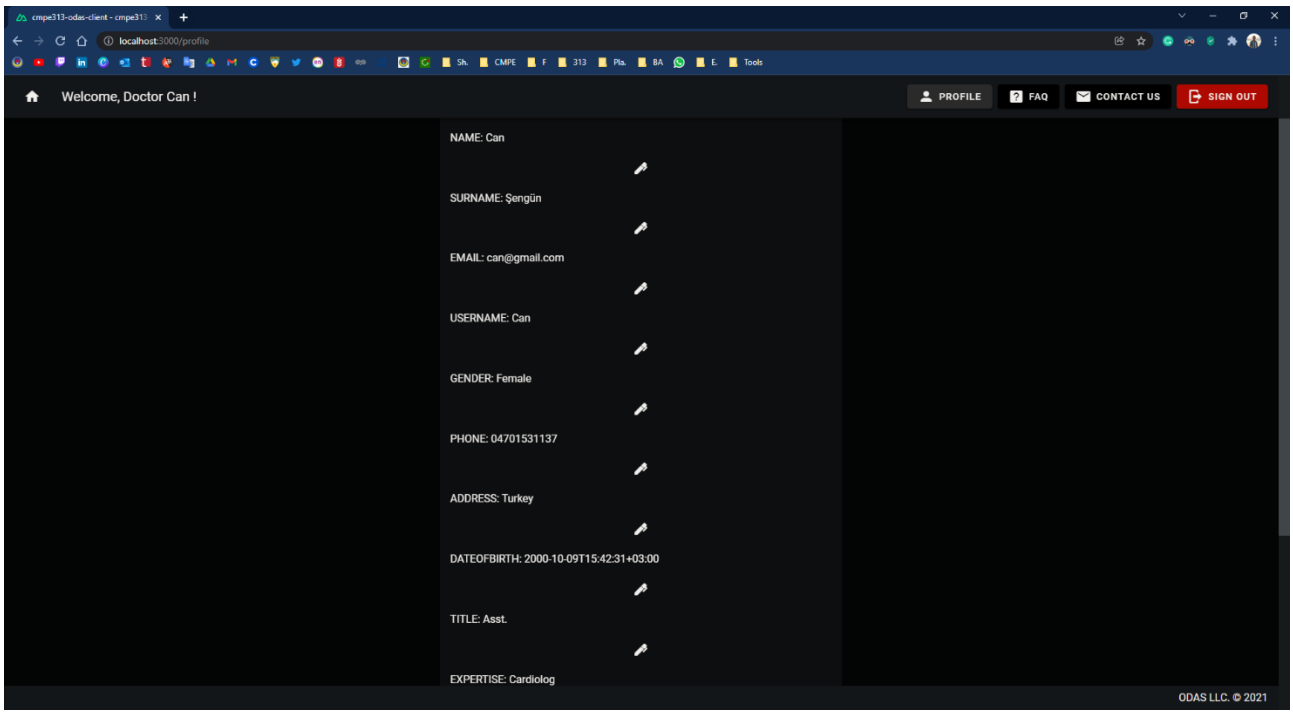
[11] Modify/Cancel Appointment for Doctor Screen: The doctor can make changes on his appointments thanks to this interface screen. When clicks an appointment, page opens to make some changes or see the relevant details. When the "Update" button is pressed, the operation takes place. At the same time, users can cancel their appointments through this screen. Appointments are canceled by pressing the "Trash" icon.



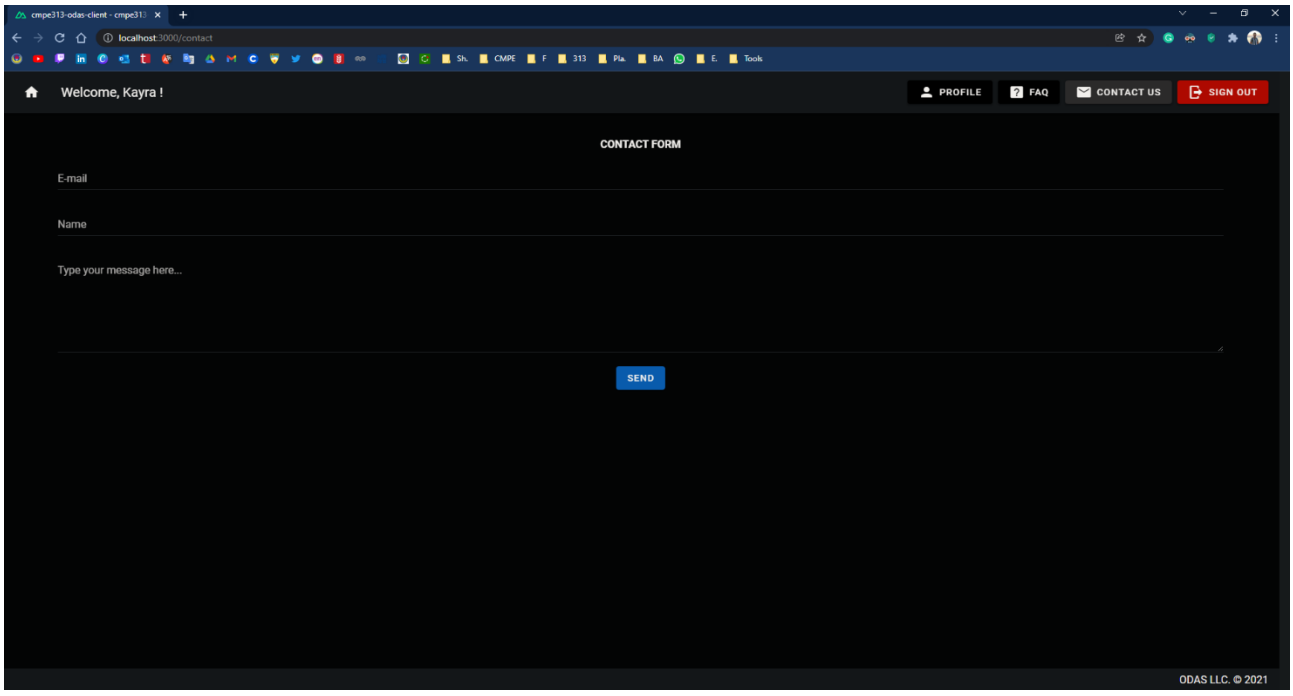
[12] Available time arranging screen: Arranging doctors available time and date by doctors. When pressing "Add new available Time Slot" button, available time is created. When pressing "Cancel Chosen Available Time Slot" button, available time is canceled.



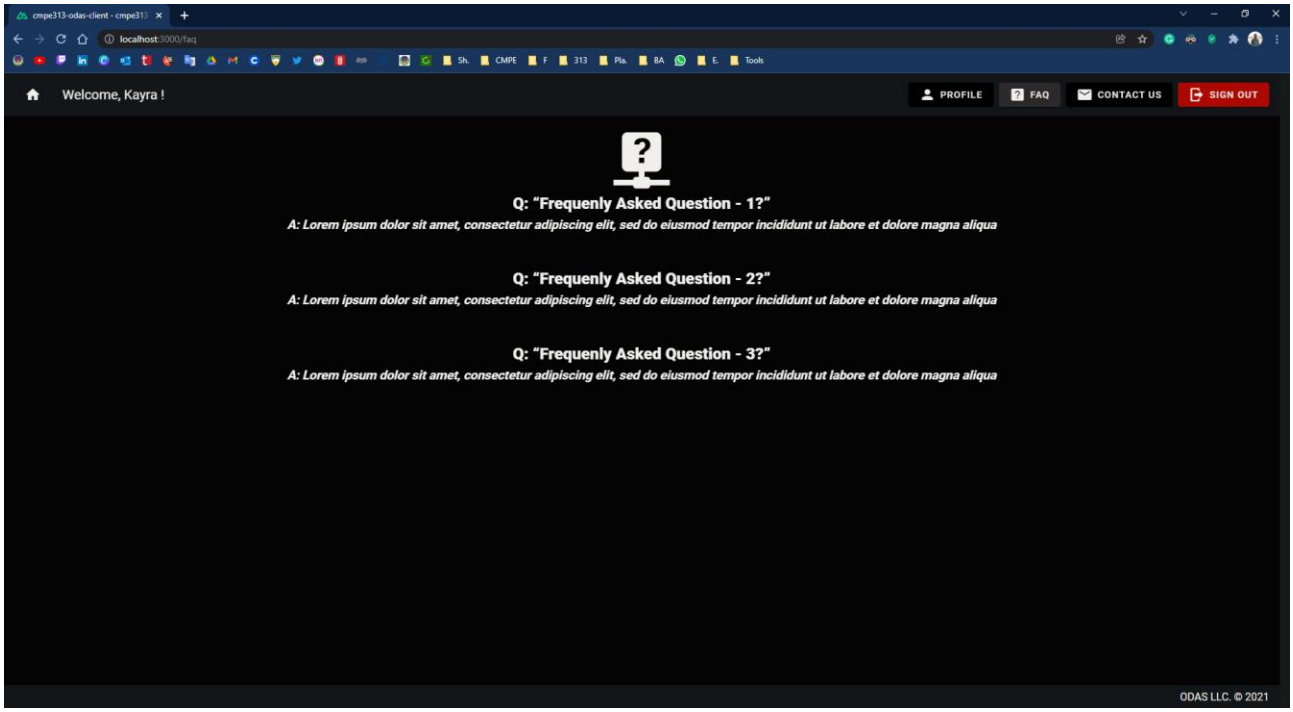
[13] Profile settings Screen for Patient: The patient can change the user information on this screen. The "Update" button changes the information, and the application directs the patient to the patient's dashboard.



[14] Profile settings Screen for Doctor: The doctor can change the user information on this screen. The "Update" button changes the information, and the application directs the doctor to the doctor's dashboard.



[15] Contact us screen: Users will be able to access this page by clicking "Contact us" on the main screen. A window will appear in this section. In this window, the user's name, surname, and e-mail information will be requested. However, there will be a box to write the message. The reason for requesting this information is to enable a user who is not registered to the system to communicate.



[16] Frequently Asked Questions Screen: Access to the "FAQ" screen is provided from the user's dashboard. This screen allows users to view previously asked questions.

3.2 Hardware Interfaces

The server software requires a computer system with 8 GB RAM & 64-bit supported 2 GHz Dual-core CPU as recommended hardware platform. For storage purposes, an SSD with 100 GB free space is suggested, but replacing with HDDs is also considerable. Moreover, no additional hardware devices (e.g., biometric scanners, NAS devices, …) are required for overall software and any web browser supported device will be enough to reach out the client application.

3.3 Software Interfaces

"Go" SDK (version 1.7+) and its dependent libraries & tools are required for the server side. PostgreSQL (version 14.0+) will be used as one-instance main relational database alongside the server application. Server app, frontend app and PostgreSQL will be optionally containerized via OS-level virtualization (*Docker*). Backend and database will talk each other using TCP adapter over *localhost* interface. The frontend application requires Node.js (version 16+ LTS) & NPM be pre-installed, and it will be based on Nuxt.js. Furthermore, the client will be able to talk to server application either over internet or LAN. And the data will be exchanged in JSON format through RESTful API.

3.4 Communications Interfaces

The client application can be reached from a web browser. Client and server will communicate over HTTP (TCP/IP) standard. The communication must be actualized using HTTP Secure (HTTPS) to protect users' sensitive data over network as a security constraint. Additionally, permanent user credentials (e.g., passwords) will be stored as hashed format (via ***bcrypt*** – Blowfish cipher implementation –) into the database behalf of privacy and security considerations.

4. System Features

4.1 Sign Up

4.1.1 Description and Priority

This screen is one of the most important screens that form the basis of the system. First, a user must create a user account to be able to perform operations on the system. The user enters the requested username, surname, identity number, residence address, telephone number, e-mail address, gender, date of birth information on this screen. Then he creates his account by clicking the "create account" button.

Prioritization Level: 1

4.1.2 Stimulus/Response Sequences

Input: Requested information is given by user and there is no wrong input. There is no collision of same e-mail address.

Output: Registration is successful. Check your e-mail address to confirm your account and start using the application.

Input: Wrong or missing input given by user.

Output: Sign up is failed. Please try again with correct information.

4.1.3 Functional Requirements

REQ-1: Both patient and doctor shall create an online account.

REQ-2: Both patient and doctor shall give correct information while creating an account.

REQ-3: If user give wrong e-mail address, the system shouldn't accept the signing up processes.

REQ-4: While creating an account, if there is a mismatching between two passwords, system should give an error message which is "The first password you entered does not match the second password."

REQ-5: Since the user is required to fill in all the fields marked with '*', the system should not record the user and give a warning when it sees a missing field.

REQ-6: User registration will not take place without accepting the Terms of Use and Privacy Policy.

4.2 Log In

4.2.1 Description and Priority

On this screen, the user logs in with his own username or e-mail account and password. After checking the accuracy of both given information on the system, the user's login process is completed.

Prioritization Level: 2

4.2.2 Stimulus/Response Sequences

Input: User's e-mail/username and password.

Output: "Log in is successful" message is the output message and system conduct user to homepage.

Input: Wrong information for e-mail/username or password.

Output: "Log in failed". "Please try again."

4.2.3 Functional Requirements

REQ-1: The user who enters the e-mail/username and password shall be able to press the button.

REQ-2: The system should be able to authenticate and authorize the user.

REQ-3: System administrator, doctor and patient shall log in to own accounts by using correct username/e-mail and password information.

REQ-4: System should give an output which is "Your password or e-mail is wrong, please try again...", when user give a wrong information.

4.3 Profile Settings

4.3.1 Description and Priority

On this screen, both the patient and the user can view the profile settings and make changes on them. The system also warns about information that users cannot change.

4.3.2 Stimulus/Response Sequences

Input: Press the button of "Profile"

Output: Profile screen appears.

Input: Update user's information.

Output: "User information updated". If user try to change identity number, system give an error message which is "This field doesn't changeable."

4.3.3 Functional Requirements

REQ-1: The system should provide the opportunity for both patients and doctors to view their profiles and change some of their information.
REQ-2: When user try to change his/her information's, system will show an error message, or it will show success message according to user inputs.
REQ-3: The system should not allow the user's ID to be changed.
REQ-4: The system shall be able to bring different information depending on whether the user is a doctor or a patient.

4.4 Search Available Appointments (Patient)

4.4.1 Description and Priority

This screen allows us to implement one of the most important functions of the application. From this screen, the patient chooses the hospital, doctor, date, and time according to his/her wishes. As a result of these selections, patients press "Submit" button and he/she make an appointment.
Prioritization Level: 4

4.4.2 Stimulus/Response Sequences

Input: As every patient should be able to get an appointment, he/she give information about his/her hospital, doctor, date, and time choice
Output: A new screen opens, and appointment information shown in this screen with a message: "Your appointment taken successfully."

Input: The patient cannot find the available appointment as a result of filtering.
Output: A warning message shown the user. (There isn't existing appointment according to your given data.)

4.4.3 Functional Requirements

REQ-1: Every patient searches an available appointment according to his/her choice.
REQ-2: The system shall show available time and date for specific doctor and hospital.
REQ-3: The system should enable users to search for available doctors on the screen where they can make an appointment.

4.5 Make Appointment

4.5.1 Description and Priority

After finding the appropriate appointment type, the user performs the appointment process. This screen allows users to book appointments.
Prioritization Level: 5

4.5.2 Stimulus/Response Sequences

Input: User press make appointment button to make an available appointment.
Output: A screen where available doctor, hospital and date and time selection is made appears in front of the user.

Input: The user presses the "Submit" button after making the desired selections.
Output: The message "Your appointment has been created" appears and the user is directed to his/her dashboard.

Input: The user presses the "Submit" button after making the desired selections.
Output: The message "Could not create an appointment at the specified date or time, please try again" appears.

4.5.3 Functional Requirements

REQ-1: Every patient should make an appointment and system should give a message due to user entry.
REQ-2: The system shall send SMS and e-mail for confirm his/her appointment.
REQ-3: The system should write the user-created appointment to the database.

4.6 Modify/Cancel Appointment

4.6.1 Description and Priority

For this screen to appear, users press any of their existing appointments. They can change or cancel their appointments on the screen that appears. It can also view the details of the appointment information.
Prioritization Level: 7

4.6.2 Stimulus/Response Sequences

Input: User press the modify button.
Output: The fields that he/she wants to change appear. The change is made, and the confirmation message appears.

Input: User press the modify button and change date and time.
Output: An error message shown user screen. The user is expected to enter another date and time.
Input: User press the cancel button.
Output: Cancellation process successful.

Input: User press the cancel button.

Output: “Cancellation process could not be done” shown user screen because of some server connection issues.

4.6.3 Functional Requirements

REQ-1: The system shall allow reservations to be modified or canceled.
REQ-2: Both patient and doctor shall modify or cancel their appointment with the system providing.
REQ-3: In modifying processes, the system should be making changes successfully.
REQ-4: The system shall give the users on my appointments screen the option to change or cancel an appointment.
REQ-5: The system shall be able to enable the user who presses the "Update" button to change the desired feature of his/her appointment.

4.7 View Appointment/My Appointments

4.7.1 Description and Priority

On this screen, both patients and doctors can see their appointments. With using this screen, users can access modifying screen by using one single button. Cancellation can be done directly here.
Prioritization Level: 6

4.7.2 Stimulus/Response Sequences

Input: User select “My Appointment” button.
Output: If there is an appointment in here, appointment/s is/are shown in this page. Otherwise, it shown an empty page and it shown a message:” There isn’t no appointment has been made.

4.7.3 Functional Requirements

REQ-1: The system shall provide a clear looking for appointments screen.
REQ-2: Users shall look their appointments to do important processes.
REQ-3: The system shall display users' appointments with their details.

4.8 Setting Available Time/Arrange Available Time (Doctor)

4.8.1 Description and Priority

This screen only exists for doctors. Doctors determine appointment dates and times through the application according to their availability. The doctor should determine the time so that the appointments can be made by the patients.
Prioritization Level: 3

4.8.2 Stimulus/Response Sequences

Input: Available date and time set by doctor.
Output: “Date and time set successfully” message shown doctor’s screen.

Input: Available date and time set by doctor.
Output: “This date and time can’t arrange. Please try again.”

4.8.3 Functional Requirements

REQ-1: Doctors shall create available date and time for give an appointment and system should create this information in application successfully.
REQ-2: The system should save date and time information to database.

4.9 Admin Interface

4.9.1 Description and Priority

This screen is the screen that only the system administrator sees, and all kinds of critical operations are performed. The system administrator adds, deletes, and updates users to the database. At the same time, it ensures that the appointments made by the users are stored correctly on the database. It deletes very old user-approved appointment records in order not to take up space.

4.9.2 Stimulus/Response Sequences

Input: Admin selects “manage users” button.
Output: Admin access whole user information through the database.
Input: Admin selects “statistics”
Output: Whole statistics about the application appears. For example, how many users create an account etc.
Input: Admin selects “Make change in App.” Button.
Output: The screen where you can make some changes appears.

4.9.3 Functional Requirements

REQ-1: The system should enable system admin to add/delete/update operation separately.
REQ-2: The system should enable system admin to make changes in patient and doctor information.
REQ-3: The system should enable system admin to delete past appointment in database.
REQ-4: The system should enable system admin to manage all database related to this application.
REQ-5: The system should be able to present the data where the logs are kept to the admin.

4.10 FAQ

4.10.1 Description and Priority

This screen allows users to view frequently asked questions. This prevents users from asking the same questions.

4.10.2 Stimulus/Response Sequences

Input : User chose a question.
Output: Answer is appeared to related question.

4.10.3 Functional Requirements

REQ-1: The system should provide user to see the faq questions and theirs answers.

4.11 Contacting with Relevant Persons (Contact Us)

4.11.1 Description and Priority

This screen is the screen that allows patients to communicate with the relevant people. Patients can contact this relevant employee. This person (relevant employee) is only assigned at this stage. Other than that, it has no other role.

4.11.2 Stimulus/Response Sequences

Input: User fill in the blank boxes which want some information like name and e-mail.
Output: Information send.

Input: User press “Send Anonymous” button and send message.
Output: Information successfully send.

4.11.3 Functional Requirements

REQ-1: The system should send these messages to relevant person.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system must have a very fast response time. It is not a problem if there will be a slight slowdown in the system when too much work is overloaded on the system. However, how late users get their work done reduces the reliability of the application proportionally. There should be no loss of speed during both the registration and login processes. All procedures performed by patients and doctors should be carried out as quickly as possible. The system must be built on an efficient database. The system administrator should not have problems accessing the data. At the same time, the client-server connection should be at the best level so that transactions can be made over the system for 24 hours.

5.2 Safety Requirements

Since sometimes unavoidable reasons can crash the system, taking a good backup of all existing data will increase users' confidence in the application. The fact that the data is always backed up will make it much easier for the system administrator to repair the damage in the application in case of a possible crash.

5.3 Security Requirements

Data security is the most important requirement of the application. While keeping the private information of the users in the database, it is most important that the access to them is prevented by external people. All communications between client-server and all user data must be stored in an encrypted form. Only the system administrator should make critical changes to the application. At the same time, only the system administrator can grant full access to the system.

5.4 Software Quality Attributes

Flexibility, maintainability, and testability are important characteristics especially for developers, since there are always on-demand business requirements (e.g., additional features, changes to existing ones). Therefore, codebase should be able to be easily refactorable to meet these needs. And the code must be testable to prevent from possible errors in production environment.

5.5 Business Rules

Any action made on this application does not affect others. Patients and doctors cannot take an action that will damage the system or change the system because of the operations they perform on the application. They can only use functions that fulfill the main purpose of the application, such as making an appointment, canceling an appointment, changing it, determining a suitable time. Important actions can only be done by the system administrator.

6. Other Requirements

6.1 Usability

- The System should be user friendly.
- There should be a detailed user manual describing the use of the application.
- The system should be user friendly.
- Users should not experience confusion while using the application.
- Descriptive error messages of the system should be shown to the user when necessary, so that the user can easily understand what to do.
- The graphical interface of the application should be understandable and interesting.
- The performance of the application should be at high levels so that the user does not force the application.

6.2 Reliability

- The system must work 24 hours a day. The user should be able to use this application whenever he wants.
- The failure rate of the system should be at very low levels. Errors that may occur independently of the user should be minimized.

6.3 Supportability

- The system must be able to run on all operating systems (Microsoft Windows, LINUX, UNIX, Mac OS)
- The system should be prepared for possible updates.
- The system should provide some international conventions which are languages, time zone.
- The system should be maintainable.

6.4 Durability

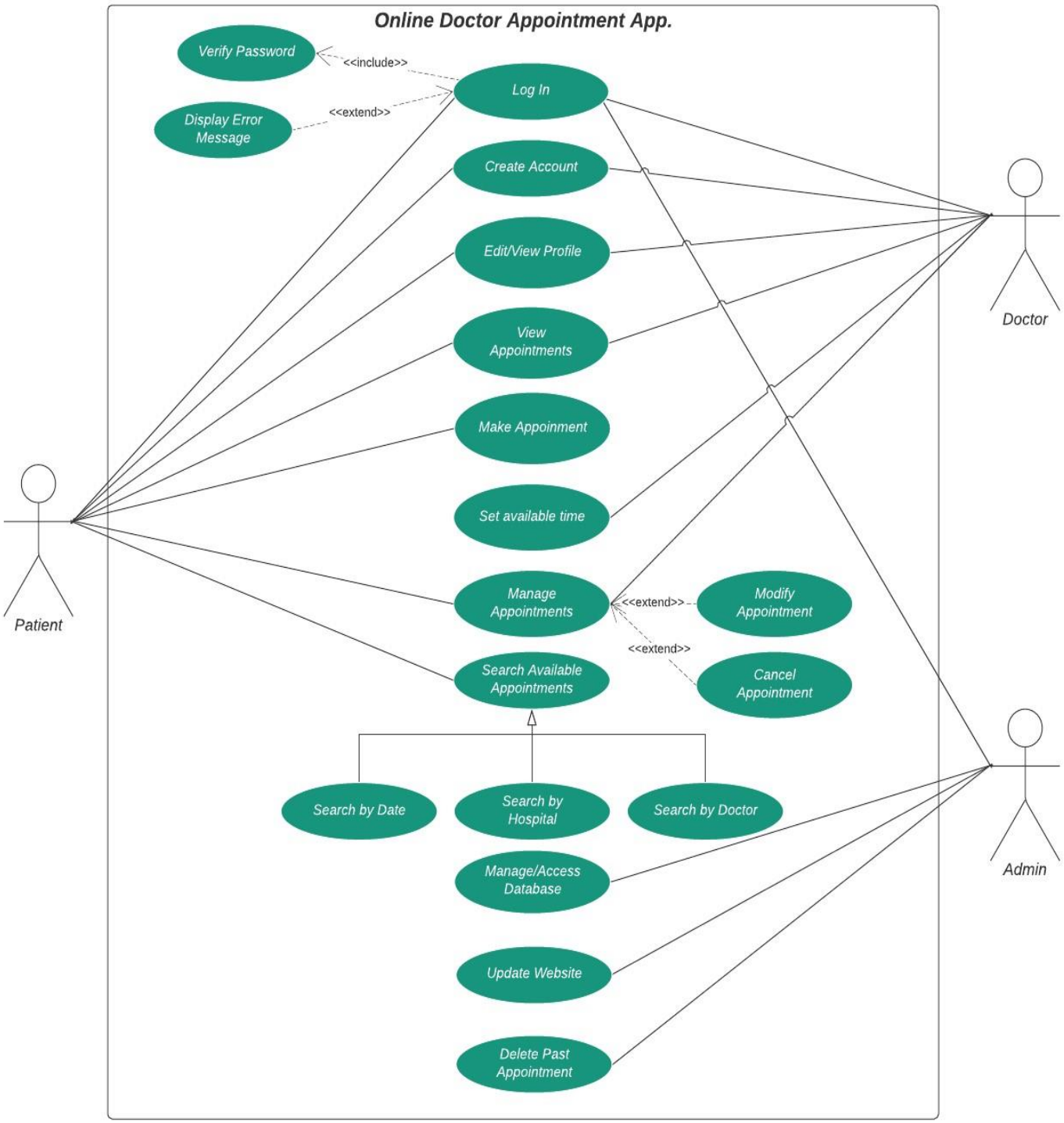
- The system must be prepared for overload.
- The system should provide lifetime use.
- The system must be powerful during critical operations performed by the system administrator.

6.5 Privacy

- Due to the law on the protection of personal data, users' data shall be stored in an encrypted manner.
- Users can only change their personal information with the help of the system.
- The system shall ensure the confidentiality of the data as a priority.

7. UML Diagrams

7.1 Use Case Diagram



7.1.1 High Level Descriptions of Each Use Cases

Use Case	Verify Password
Actor	Patient, Doctor, System Administrator
Description	**Verify Password** use case is not a use case accessed directly by an actor. It checks the correctness of the password entered in the background when any user enters the username and password and enters the application. Log in cannot be performed without the password validation use case.

Use Case	Log In
Actor	Patient, Doctor, System Administrator
Description	**Log In** process allows all users to login to the web application. Patients log in using their username and password. Likewise, doctors enter the web application by entering their username and password. The system administrator logs in with the admin username, unlike everyone else. Specific to the system administrator, the interface that appears after the login process is different from everyone else. We have one more use case included in the log in use case. There is also a use case called Displaying an Error message.

Use Case	Display Error Message
Actor	Patient, Doctor, System Administrator
Description	**Display Error Message** is extended to Log in use case. This use case may not happen every time. An error message is displayed when the user logs in with the wrong username or password. The user is prompted to re-enter their password or username.

Use Case	Create Account
Actor	Patient, Doctor
Description	**Create Account** process allows all users to create an Account that will allow them to log in to the web application. Patients perform this process by entering their personal user information into the system. Information such as username, password, e-mail address, home address, telephone number are mandatory information. Likewise, doctors do this by creating an account in the web app. Some of the information required to be entered in the Create Account process is also used later in the login process.

Use Case	Edit/View Profile
Actor	Patient, Doctor
Description	**Edit/View Profile** action is used to modify and review the previously entered information in the account created by the patient or doctor. Users can access the data and make some changes through the "Profile" section. All information given when creating an account can be changed in this section.

Use Case	View Appointments
Actor	Patient, Doctor
Description	**View Appointments** allows the doctor and the patient to receive information before, during and after the appointment. In the previously created appointment, the patient can get information about the doctor and access details such as date and time. Before the appointment, the doctor can access information about the patient through the system. Appointments are displayed in the "My Appointments" section.

Use Case	Make Appointment
Actor	Patient
Description	**Make appointment** allows the patient to easily make an appointment via the web application. After logging in to the patient's account, he can view the appropriate appointments and get his appointment as he wishes (date, hospital, doctor).

Use Case	Set Available Time
Actor	Doctor
Description	**Set Available Time** process allows doctors to determine the time that suits them through their profile accounts in the web application. Patients can look at these times and choose a date and time according to them.

Use Case	Manage Appointment
Actor	Patient, Doctor
Description	**Manage Appointment** allows users to act on their existing appointments. After logging into the system, users can make operations from the "My Appointments" section. Patients may want to change or cancel their existing appointments for special reasons. Likewise, doctors may have to change the time of that appointment or cancel the appointment after they make an appointment. For all cases, change processes are provided through the application. If the doctor wants to change the appointment time, he marks another available time on the system and a notification is sent to the patient in this way. If the patient accepts, the appointment appears in its new form in the "My Appointments" section.

Use Case	Cancel Appointment
Actor	Patient, Doctor
Description	**Cancel Appointment** allows doctors or patients to cancel the appointment according to their wishes. The patient can cancel the previous appointment by entering the "My appointment" section. The doctor has the authority to cancel the appointment by entering the section where the appointments are made.

Use Case	Modify Appointment
Actor	Patient, Doctor
Description	**Change appointment** section allows both doctors and patients to make changes to their appointments. The patient can change the previous appointment for a different date and time by going to the "My Appointment" section. Likewise, the doctor has the authority to change the appointment time. After clicking on "My Appointment", he clicks on the "modify" button and can change the appointment according to the date and time he wants.

Use Case	Manage/Access Database
Actor	System Administrator
Description	**Manage/Access Database** operation allows the system administrator to access and edit database data. The system administrator can delete users, namely doctor or patient, from the system and add them to the system in the same way. They can not only delete and add data, but also make changes on the data. After gaining access to the database, they can make the right queries and complete the operation. Finally, the message "Changes saved successfully" is displayed to the system administrator. In addition, it can organize the system and delete unused data that takes up space and free up space in the system.

Use Case	Search Available Appointments
Actor	Patient
Description	**Search available appointments** allows patients to find appointments which are available, mean these had not taken by other patients yet. A search form with multiple dropdown menus and bars is shown to patients for letting them filter and display available free slots. These filters consist of 3 different groups: “search by date”, “search by hospital”, “search by doctor”. Then patients can create appointment records for themselves individually.

Use Case	Search by Date
Actor	Patient
Description	**Search by Date** allows patients to filter available appointment slots by date. In the webpage, a calendar toolbox let patients to display all records at the chosen day, month, and year.

Use Case	Search by Hospital
Actor	Patient
Description	**Search by Hospital** allows patients to filter available appointment slots by chosen hospital. In the webpage, 4 dropdown menus are provided to patients. Each patient can search appointments by country, city, district, and hospital respectively.

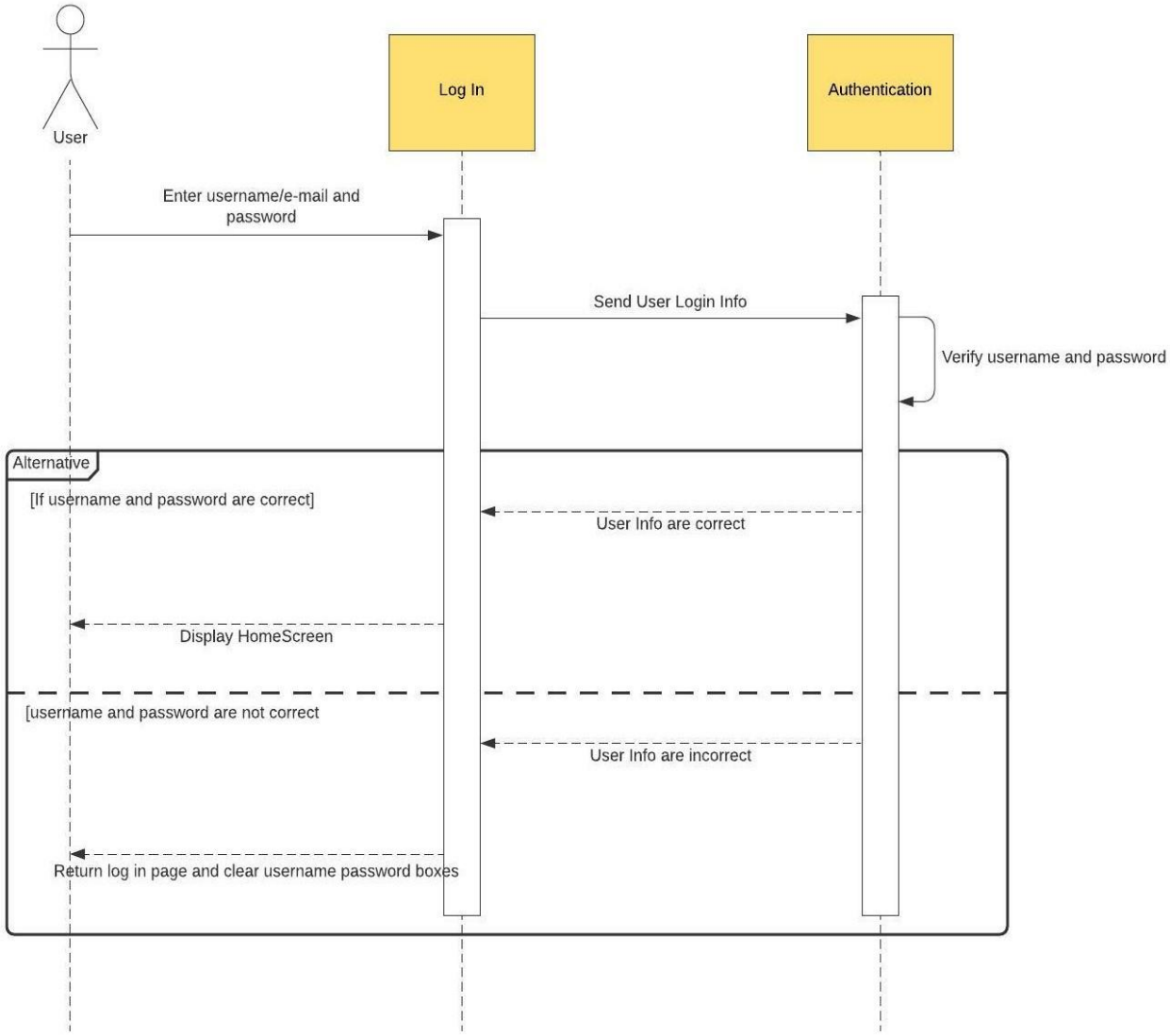
Use Case	Search by Doctor
Actor	Patient
Description	**Search by Doctor** allows patients to filter available appointments by doctor. Patients can type title, first name or last name of the doctor(s) that they desired to get appointment from. In the doctor selection part, there is provided a search bar to list all doctors that have similar name or title depending on the patient's query. This is also available to use advanced search parameters to find doctor(s) by their office, phone number or expertise.

Use Case	Update Website
Actor	System Administrator
Description	**Update website** operation allows the System administrator to update the information available in the web application, and to make edits and innovations in the web application. After logging into the server administration panel with the username and password, the system administrator can change the design and security features of the web application.

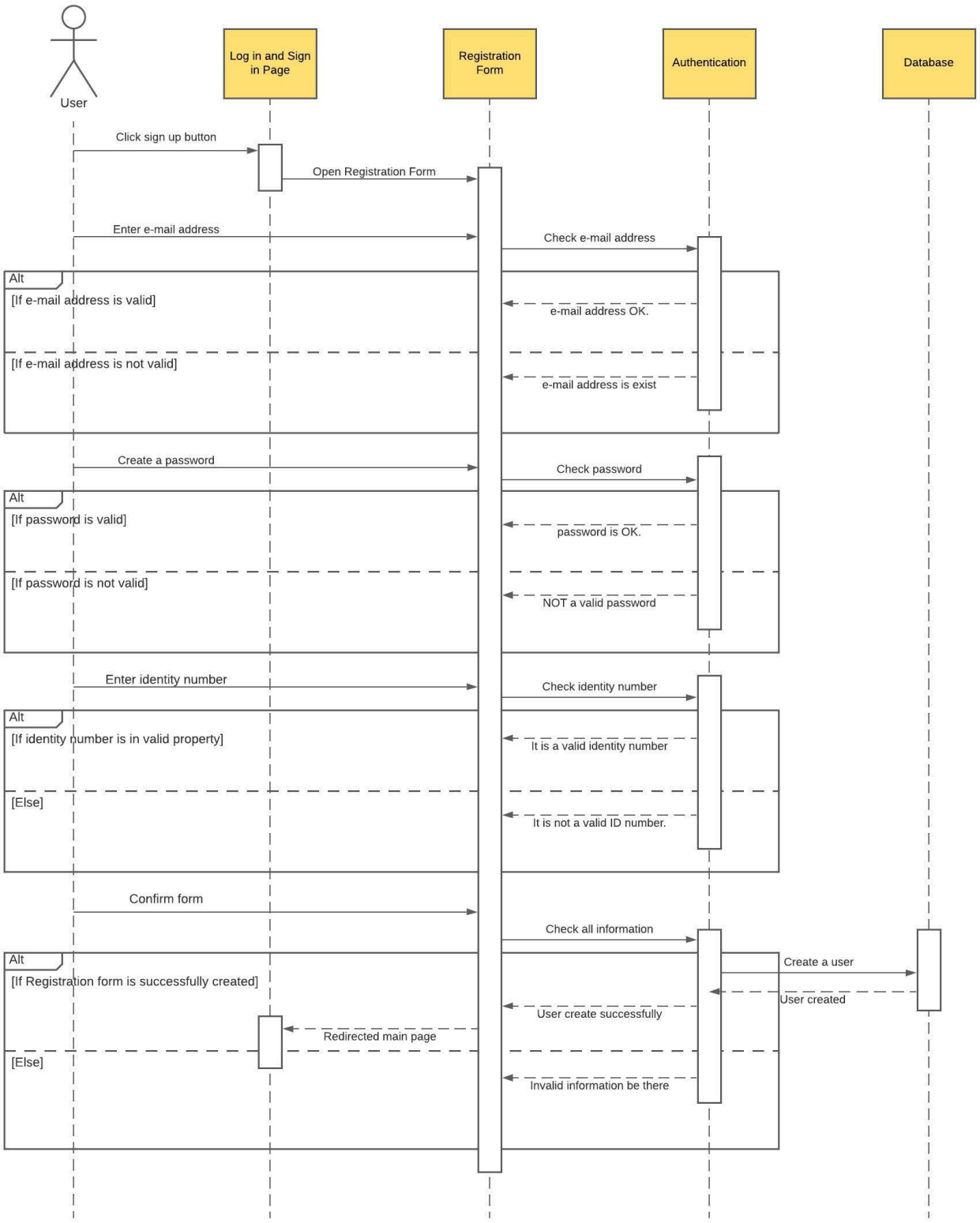
Use Case	Delete Past Appointment
Actor	System Administrator
Description	**Delete past appointment** operation is a process that allows the system administrator to delete the data that is kept idle in the system, that is, past appointments that have lost their function do not take up space anymore, as the system administrator deletes the data from the system. During this deletion process, the patient is notified.

7.2 Sequences Diagrams

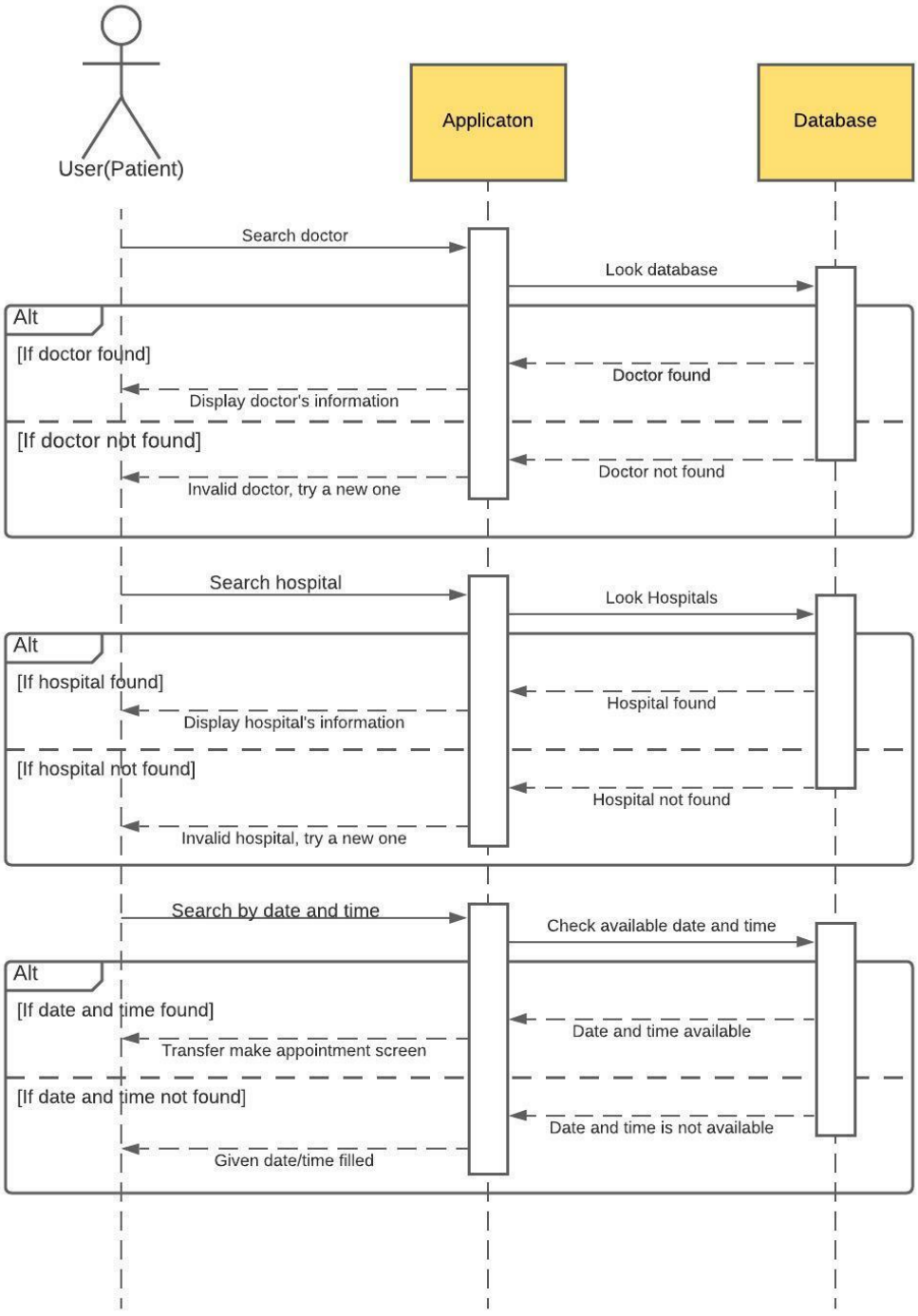
7.2.1 Sequence Diagram on Log In



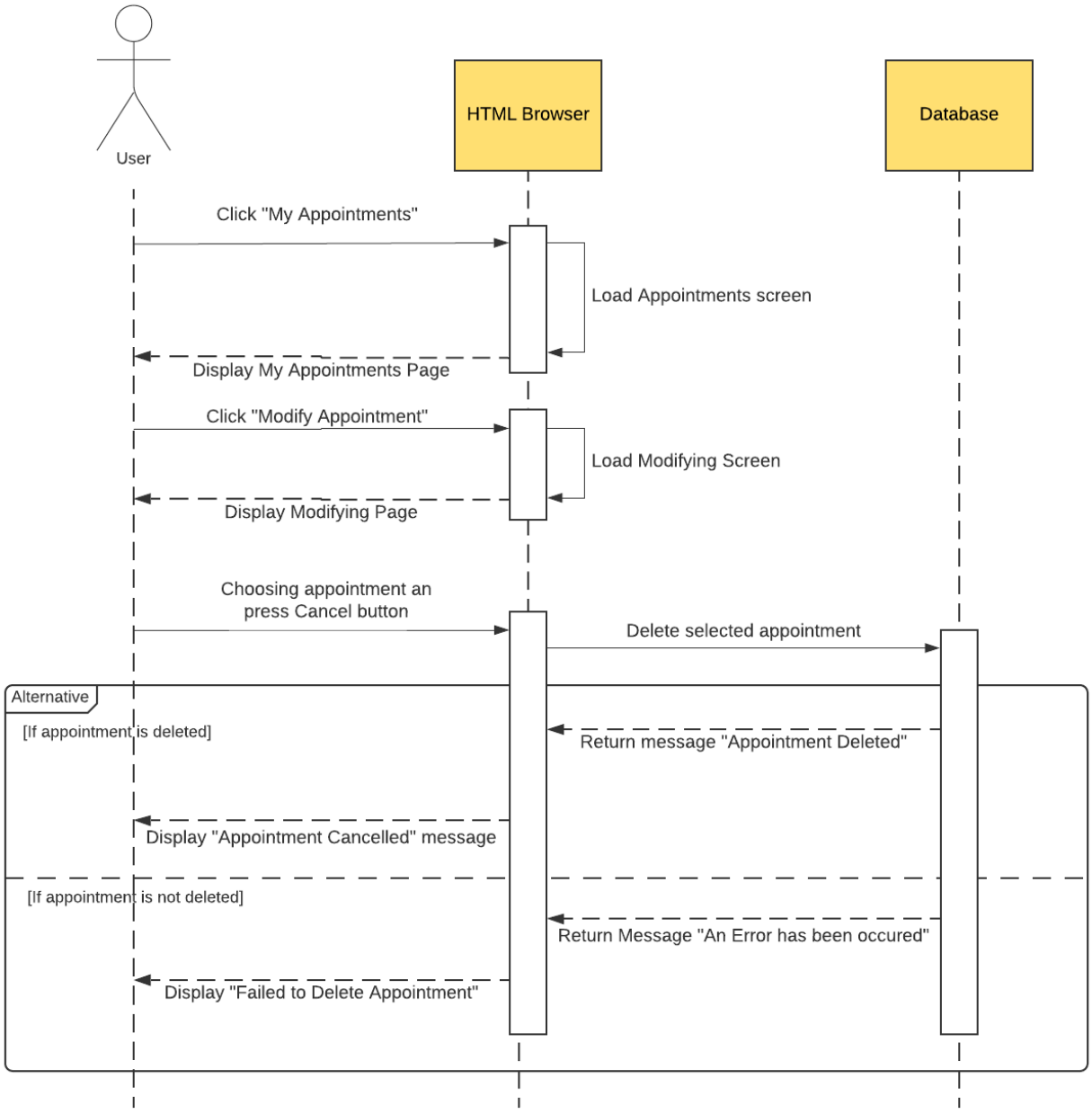
7.2.2 Sequence Diagram for Registration



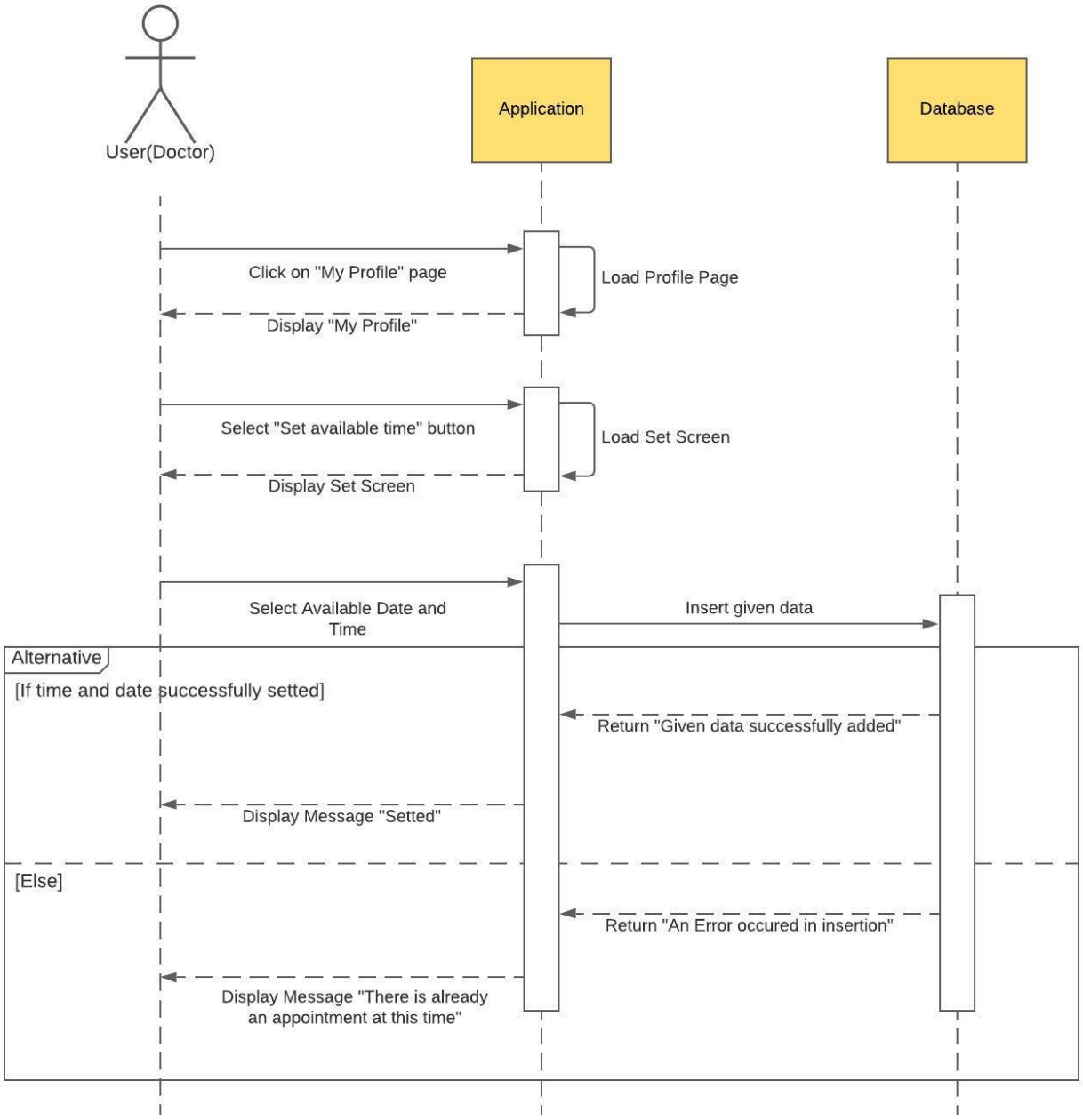
7.2.3 Sequence Diagram for Search



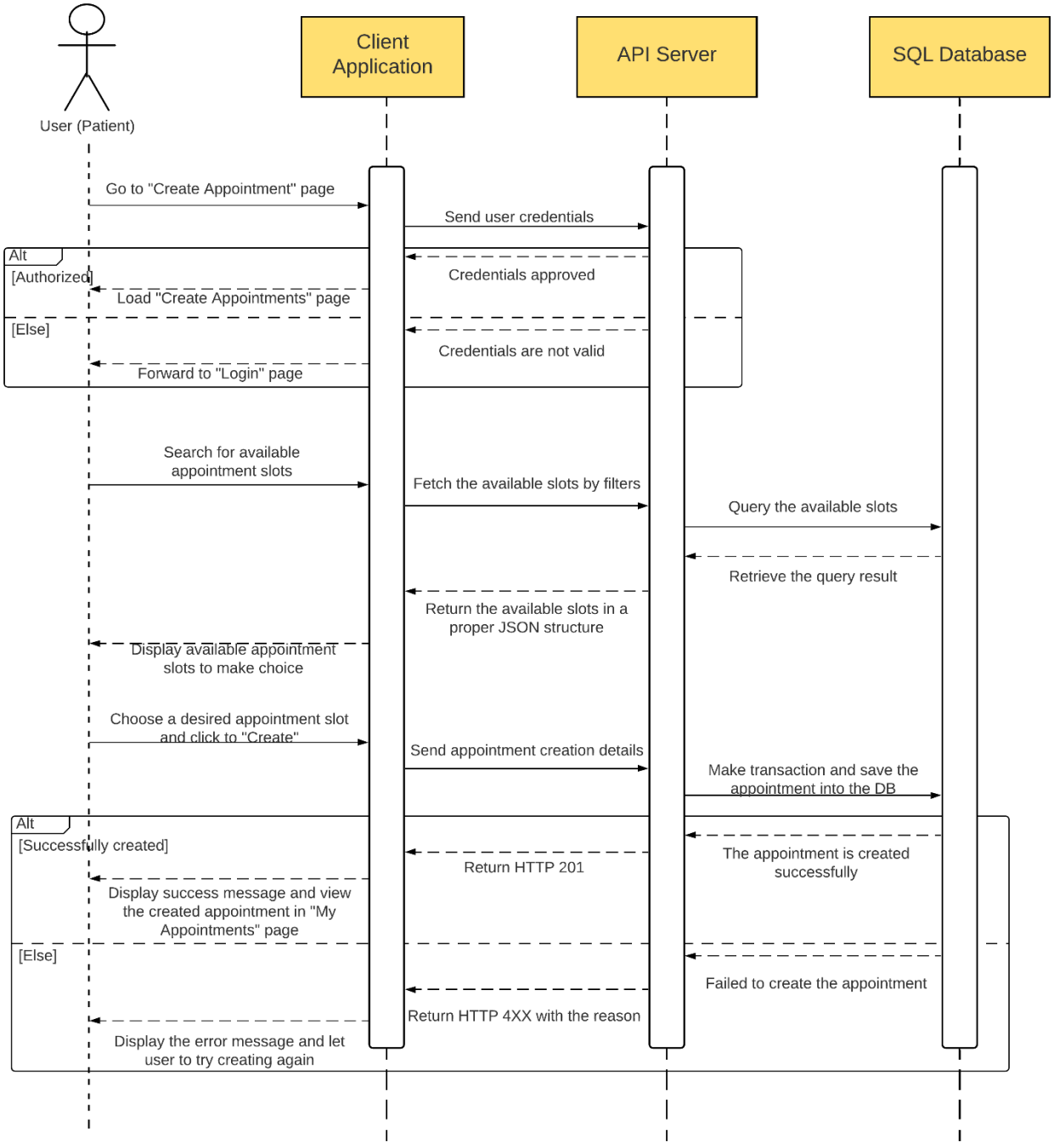
7.2.4 Sequence Diagram for Cancel Appointment



7.2.5 Sequence Diagram for Set Available Time (Doctor)

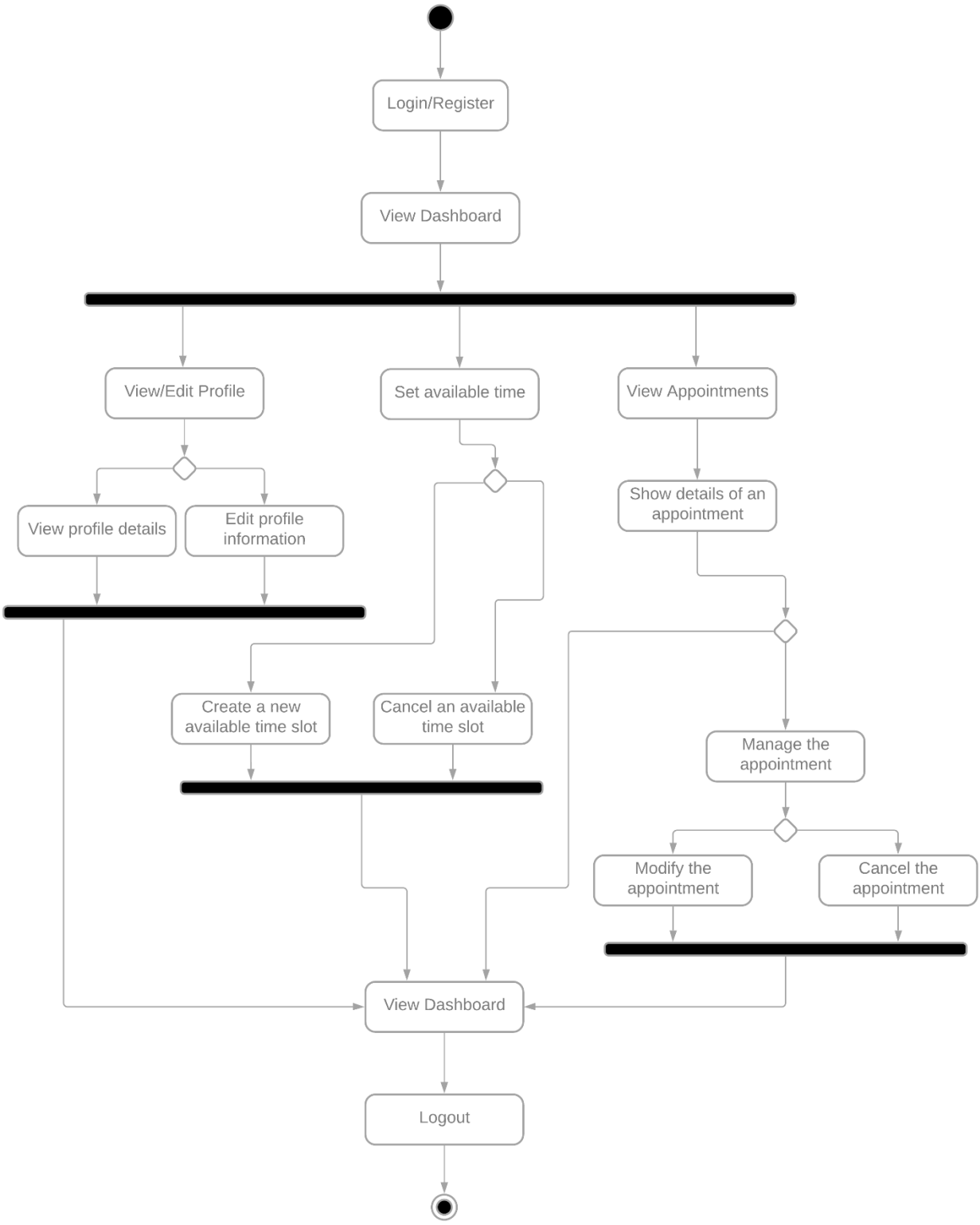


7.2.6 Sequence Diagram for Make Appointment (Patient)

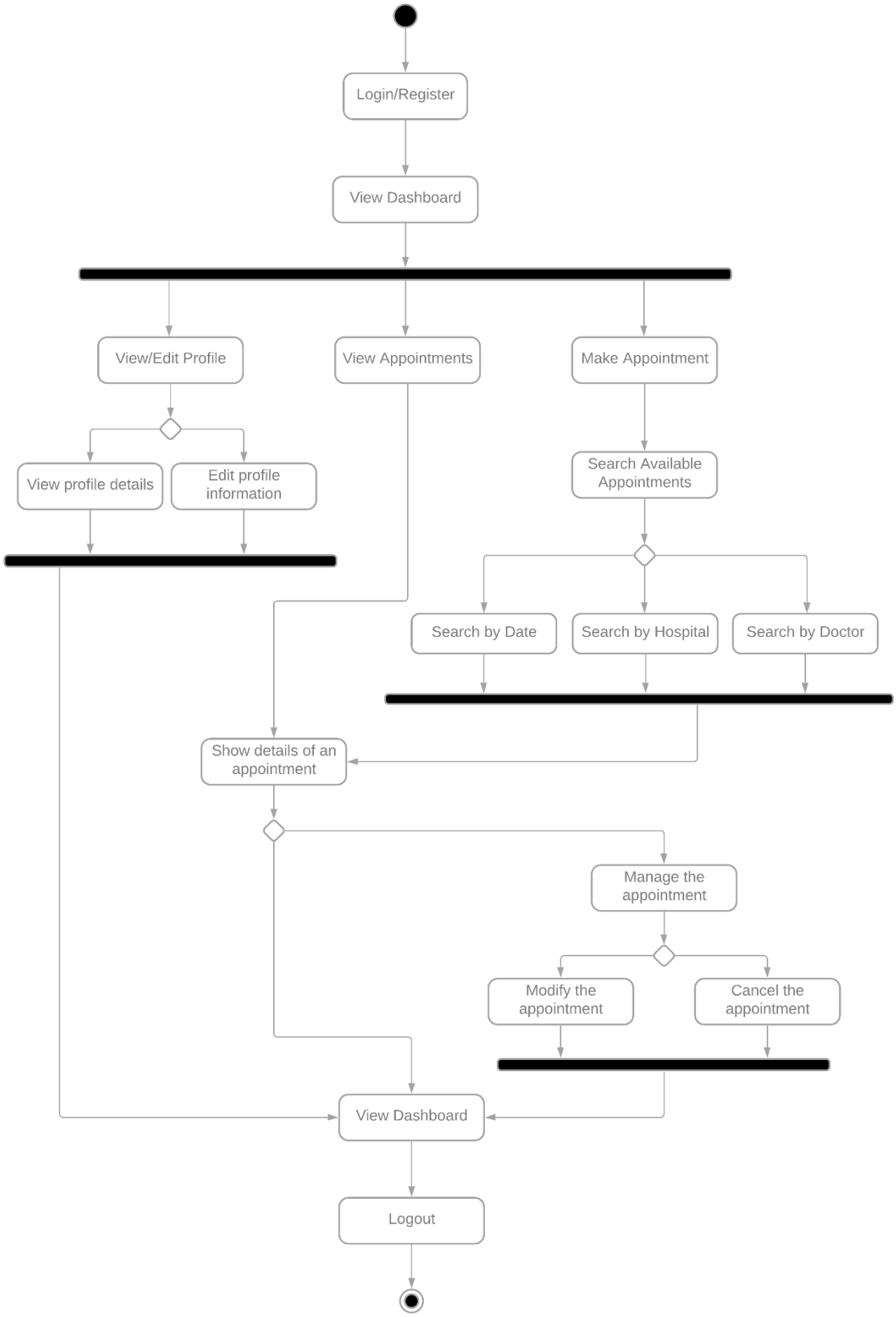


7.3 Activity Diagrams

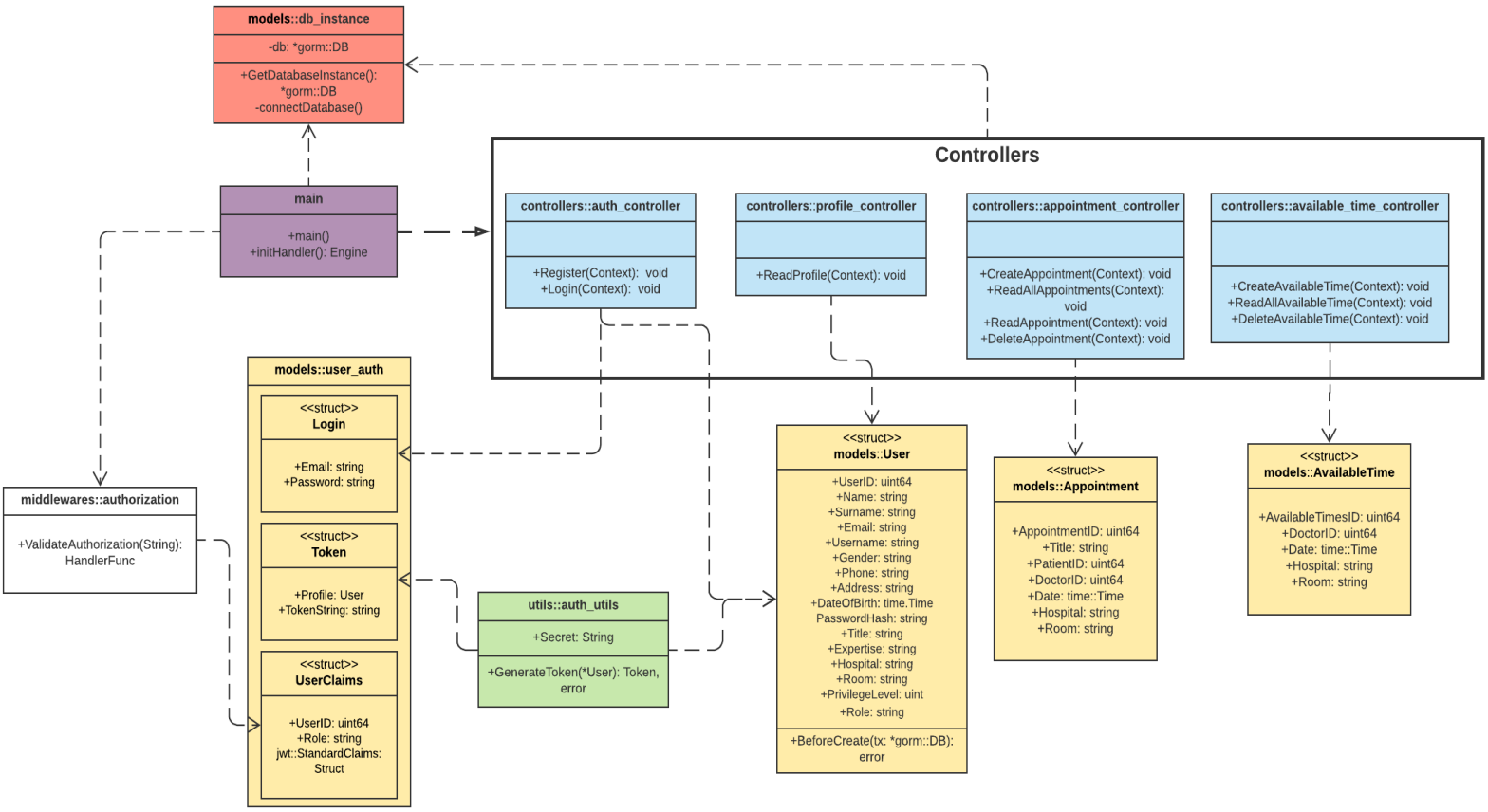
7.3.1 Activity Diagram for Doctor



7.3.2 Activity Diagram for Patient



7.4 Class Diagram



Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

We used these diagrams under the title of "UML Diagrams" because too many diagrams were created for the application. (see section 7)

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>