Software Requirements Specification

for

DocBook

Version 1.0 approved

Prepared by Roudranil Kar

Heritage Institute of Technology

21st January, 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
DocBook	20.01.23		1.0

1. Introduction

1.1 Purpose

The SRS describes the working of the DocBook application. The objective of this software is to provide to its users an easy access to every kind of doctors available and booking of ambulances in no time if the user is a patient and also shows the various appointments scheduled if the user is a doctor. This software can be used by both a patient and a doctor as per the needs. In this SRS we will be discussing regarding the various target specifications and the system features of the software on which we will be working.

1.2 Document Conventions

The document is created by keeping in mind the IEEE template for system requirement.

The system features are written based on the functional point of view.

1.3 Intended Audience and Reading Suggestions

Intended Audience

- All kinds of users can use this application, may it be a normal patient or a doctor. A person looking for an ambulance in an emergency situation or a doctor who is checking his appointments for the day.
- This software is made keeping in mind the immediate requirements of a normal patient, that is if he/she requires an ambulance or a doctor in the middle of the night also, he can use this software to fulfill his/her requirements.
- For doctors, who are starting afresh can create their accounts and use this software so that the patients can reach out to them. Ambulances of various hospitals, even the private ambulance owners can use this app to track where the ambulance needs to be send.
- This software is also open to programmers and open source developers if they want to contribute to this by updating any of the system features.

Reading suggestions

- This SRS contains all the information regarding the software starting from basic to advanced designs.
- Users should read beginning with the overview sections and proceeding through the system features to understand the proper way of using the app.
- Developers should read beginning with the overview sections and proceeding through the overall description, system features, non-functional features.

1.4 Product Scope

DocBook is an application that lets its users, if he/she is a patient, book a specific doctor or ambulance, and if he/she is a doctor, the application lets him to check the appointments for the day. To make things easier for every section of people, we have included various payments method that is, online payment and offline payment. The user interface has been kept simple so that no one finds it difficult to use it. Since, nowadays, everyone needs a doctor but the booking scenario for every doctor is not the same, therefore, this software provides an easy access to doctors and the booking procedure remains same for every doctor. Moreover, the doctors in their busy schedule cannot keep a count of the appointments, so the notification feature lets him/her know about the appointments.

1.5 References

None

2. Overall Description

2.1 Product Perspective

This software is a connection that is being made between doctors and patients and services like ambulances. The patient can choose which doctor to consult, and the application will generate the time and date as per the doctor's schedule. The doctor on the other hand can accept/reject the appointment, view all his current and past appointments and upload detailed prescriptions for the patient enriching the communication between the patient and doctors.

2.2 Product Functions

- A patient can register himself/herself to the software with the details and his aadhar card will be the document that can be used for verification. He can log in and log out at any time. He can also update his credentials at any time.
- A doctor can register himself/herself to the software with the details and his registration number will be the document that can be used for verification. He can log in, log out or update his/her credentials at any time.
- The UI has been kept simple so that the patients as well as the doctors find it easy to handle the application.
- Search has been made simpler, with filters so that the patient can find a doctor of any field without any difficulty.
- The software will be available in all regional languages so that no one finds it difficult to read.
- An ambulance (private) can also register itself to the system in the same way. The ambulance can start the journey and update the details which will be visible to the appointee.
- The feature of viewing past appointments or bookings is also added.
- The feature of online consultation is also available, that is the patient can upload his/her problems using voice note or in message format and the doctor can issue the prescription according to the problems he/she has mentioned.

2.3 User Classes and Characteristics

The various user classes and their characteristics have been listed below.

Sl. no	Users	Characteristics
1	Patients	This class can access log-in/log-out or update feature, can choose which doctor to book, can choose the payment option and also can book an ambulance and has the access to view past bookings.
2	Doctors	This class has the access to log-in/log-out/update credentials feature, can accept or reject a booking and also can issue the prescription using the online consultation feature
3	Ambulances	This class has the access to log-in/log-out/update credentials feature, can update the location details to the appointee.
4	Admin	This class has access to all functionalities of the application/software

2.4 Operating Environment

The product will work on environments:

- Web-based application which can be accessed from any device. It can run on any web browsers such as Chrome, Mozilla, Opera, and Brave.
- It can also run on Linux-based operating system such as Ubuntu.
- Android and IOS based application.

2.5 Design and Implementation Constraints

The various tools that can be used to build this application are:

- React, mongo DB, expressjs, nodejs for application and data related work.
- Java, Kotlin, C# can be used as programming languages.

The software will be build keeping in mind the Microsoft GUI standards to avoid any discrepancies.

2.6 User Documentation

A document containing the step-by-step manual of how to use the application for different functions should be provided in the app. A document for developers will also be provided where various functions and technologies to be used will also be mentioned.

2.7 Assumptions and Dependencies

- The assumption has been made that the users that is doctor, patients or ambulance using our application has the required space in his/her device with a proper internet connection so that he/she can install the app or can access it through web browsers.

- It is also assumed that during the payment if the patient is paying the fees online, after the payment has been made, the third party payment portal, redirects the user to the app or to the website displaying whether or not the payment is done.

- It is assumed that the user knows to handle the devices on which the application is accessed.

- The software will be responsible for any discrepancies that has been made during online payment, that is if the money is debited by mistake, the software will make sure the user gets his/her money back within one day.

3. External Interface Requirements

3.1 User Interfaces

Keeping in mind the different sections of people who will access our application, the web based graphical user interface (GUI) will be provided. The user interface will be made menu-driven, user-friendly so that it can look into all the requirements of the user may it be a patient, a doctor or an ambulance. The different sections and elements of the app will be made simple, so as to achieve good user experience and the application will be hosted in such a way so that the pages can load quickly. The only requirement from the user is to have a good internet connection.

3.2 Hardware Interfaces

- Devices: Phone or Computer.
- Processor: Pentium or higher.
- RAM: 100MB or higher.
- A proper internet connection of more than 1kbps.

3.3 Software Interfaces

- Operating System: Linux, windows, Mac etc.
- Tools for Development: HTML, ReactJS, ExpressJS, nodeJs, MongoDB etc.
- Programming languages: JAVA, Kotlin and C#.
- Since the application will also be web-based, a proper server is required to host the web-application.
- In addition to all the above mentioned software interfaces, separate software can be required for application frameworks, SMS and email servers and much more to come.

3.4 Communications Interfaces

All the communications between the user and the application will be done through messages to the registered mobile number and email (if required). The user can take screenshot of the payment to show to the doctors and the receipt of the same will also be send to the email. A payment confirmation message will be send to the user’s registered mobile number.

4. System Features

This section will walk you through the various system features that the application will have along with their respective functionalities.

4.1 Language Selection

4.1.1 Description and Priority

The feature of language selection allows you to select any language that the user might prefer in which they want to access the application.

4.1.2 Stimulus/Response Sequences

1. **Stimulus** – The user will open the application for the first time.
   Response – A drop down list containing all the available languages will appear.

2. **Stimulus** – The user chooses his/her preferred language.
   Response – The registration window will open and a welcome message will be shown above.

4.1.3 Functional Requirements

REQ-1: Language Selection – The software will allow its users to run the application in their preferred regional language.

REQ-2: Regional Language Choice – The software will show all the available languages according to the location of the user detected by the application. Suppose the user is from Assam, then the drop down will be, Assamese, Hindi, English, Telgu, Tamil, etc.

4.2 Registration

4.2.1 Description and Priority

The feature of the application allows the user to create his/her account and register himself as a doctor or patient or an ambulance.

4.2.2 Stimulus/Response Sequences

1. **Stimulus** – The register now button pops with the options of doctor/patient/ambulance which asks the user to register himself/herself.
   Response – A form will open which contains the details according to the choice of doctor/patient/ambulance with confirm phone number and OTP.

2. **Stimulus** – After the registration has been done, the application asks for “Sign up”.
   Response – The user can skip the option of sign-up and can get an overview of the homepage.

3. **Stimulus** – User, if he/she creates an account after pressing sign-up
   Response 1 – An “Account has been successfully created” message will pop up and the application will automatically redirect the user to the homepage.
   Response 2 – In case of any wrong details or any parameter has been left unchecked, a proper error message will be displayed.

4.2.3 Functional Requirements

REQ-1: Registering details – The software will first ask the user that if he/she is a doctor/patient/ambulance and then will take him for the necessary details filling along with phone number confirmation and OTP.

REQ-2: Sign-up/Skip for now – The user after the registration will have the option of signing up and creating an account or skipping the creation and going to the homepage directly.

4.3 Verification

4.3.1 Description and Priority

The feature of verification asks for the proper documents from the user to identify him as the doctor/patient/ambulance. This feature has been made keeping in mind the security of the application so that no one takes unfair advantage.

4.3.2 Stimulus/Response Sequences

1. **Stimulus** – The user selects the option of “Verify Yourself”.
   Response – An option of doctor/patient/ambulance appears on the screen.

2. **Stimulus** – The user chooses doctor.
   Response – The application asks the user to upload the registration certificate.

3. **Stimulus** – The user chooses patient.
   Response – The application asks the user to upload any government verified document for verification.

4. **Stimulus** – The user chooses ambulance.
   Response – The application asks the user to upload the details of the vehicle along with the registration details.

5. **Stimulus** – The user provides the necessary documents.
   Response – A message telling “You have submitted successfully, your details will be verified within few minutes” will appear.

4.3.3 Functional Requirements

REQ-1: Verification – The software will allow its users to verify themselves as doctor/patient/ambulance so as to avoid malpractice and unfair advantage.

REQ-2: Option of doctor/patient/ambulance – The user chooses what kind of person he/she is and the process of verification continues

REQ-3: Uploading documents – The application for the required documents according to the choice of user given by the person.

REQ-4: Access to bookings – The application will not allow its users to book or view or accept any bookings or appointments unless he/she is verified as doctor/ambulance/patient.

4.4 Login

4.4.1 Description and Priority

The feature of login allows the verified users to access their own accounts and the features that are associated with login.

4.4.2 Stimulus/Response Sequences

1. **Stimulus** – The user selects the option of “Log-in” after the verification is done.
   Response – A form with username/phone-number/email and password will be displayed on the screen.

2. **Stimulus** – The user enters the respective verified username/email/phone number.
   Response 1 – The user is send to the home page if the credentials are correct.
   Response 2 – An error message will be displayed below the area where the credentials are wrong.

3. **Stimulus** – The user chooses “Forgot password” option.
   Response – The application will send a verification code and a reset password link to the user’s email and phone number.

4.4.3 Functional Requirements

REQ-1: Entering the credentials – The software will allow its users to enter their respective username/phone-number/email along with the password to successfully access the facilities of the application.

REQ-2: Login – The software will redirect the users to the homepage if the credentials entered are correct or an error message will displayed below the credential which has been entered wrong.

REQ-2: Forgot Password – The software will send a verification code along with a reset link to the user’s respective email and phone number so that he can set his/her new password.

4.5 Profile

4.5.1 Description and Priority

This feature allows the users to view their profile and edit the personal information if he/she intends to do so.

4.5.2 Stimulus/Response Sequences

1. **Stimulus** – The user selects the small icon on the top right corner of home page.
   Response – A window with all the personal information regarding the user opens up.

2. **Stimulus** – The user selects the pen-like symbol to edit the information.
   Response – The user updates or changes any personal information as per his/her choice.

3. **Stimulus** – The user chooses “Go to home” option.
   Response – The application will redirect the user to the homepage once all the details are updated by him.

4.5.3 Functional Requirements

REQ-1: Viewing Profile – The software will allow its users to view their created profiles by clicking on the icon on the top right corner of the home page.

REQ-2: Updating/Changing details – The software will allow its users to update or change his/her personal details if he/she finds it important to do so.

REQ-3: Redirecting to Home – The application will have the option going back to the home page once the details are being changed or updated by the user.

4.6 Home

4.6.1 Description and Priority

The Home is the main feature of the application that contains all the major features of the app together. The options included in the home page are settings, search using filter, the information of top doctors and ambulances, booking a doctor/ambulance, checking the appointments for the day, download receipts, and many more.

4.6.2 Stimulus/Response Sequences

1. **Stimulus** – The user selects the option of settings.
   Response – A window with all the default settings will open and user can make necessary changes in the settings.

2. **Stimulus** – The user selects the profile icon.
   Response – A window opens up with all the personal information of the user.

3. **Stimulus** – The user, if he/she is patient, chooses a trending doctor's profile.
   Response – A window opens up with all the necessary information on the doctor, with the option of "book".

4. **Stimulus** – The user, if he/she is a doctor, chooses the appointments for the day.
   Response – A window opens up with all the appointments scheduled for the day.

5. **Stimulus** – The user, if it is an ambulance, chooses the option of bookings.
   Response – A window opens up with the location and address of the pickup point.

6. **Stimulus** – The user, if he/she is patient, chooses the option of view bookings.
   Response – A window opens up showing all the past bookings made by him/her.

7. **Stimulus** – The user selects the search bar.
   Response – A window with various filters like type of doctor and ambulances, date of appointment, location for appointment, location for pickup will open.

8. **Stimulus** – The user, after choosing a doctor and date clicks on "book" option.
   Response – The payment portal opens up with two options of online booking and offline booking.

9. **Stimulus** – The user, if he/she is a doctor, chooses view past appointments.
   Response – A window opens up showing all the past appointments.

10. **Stimulus** – The user, if he/she is a doctor chooses manage appointments option in the appointments for the day menu.
    Response – A window opens up with all the details of the patient and the option of accept, reject and online appointment.

4.6.3 Functional Requirements

REQ-1: Viewing Settings – The application will allow the users to change the settings to suit their preferences such as change language, enable dark mode, and manage notifications, update profile and many more.

REQ-2: Profile Icon – The application will allow to view their profile and update any information if the user finds it necessary to do so.

REQ-3: Choosing trending doctor's profile – The application will show the doctors who are gaining popularity around the area in terms of feedback given by the patients and the user can choose his profile and book him/her.
REQ-4: Selecting appointments for the day – If the user is a doctor, he/she can view all the appointments that are scheduled for that particular day.

REQ-5: Checking the bookings for the day – If the user is an ambulance, he/she can view the bookings for that day along with the location and address.

REQ-6: Past bookings – The application allows its user to check all the past bookings/appointments that has been made previously.

REQ-7: Search Bar – The application has a feature of search bar where the user if he/she is a patient can filter out and search for specific doctors and can book the doctor.

REQ-8: Manage Appointments – If the user is a doctor he can accept, reject and has the option of online appointment, which will be based on the schedule of the doctor.

4.7 Search Bar

4.7.1 Description and Priority

This feature allows the user if he/she is a patient, to search for a specific doctor or ambulance according to the specifications given by him/her.

4.7.2 Stimulus/Response Sequences

1. **Stimulus** – The user clicks on the search bar on top of the home page.
   Response – A window with the option of doctor or ambulance opens up.

2. **Stimulus** – The user selects doctor.
   Response – The option of filtering appears where the user can choose the date, type of doctor.

3. **Stimulus** – The user selects ambulance.
   Response – The option giving location by map or manually entering appears.

4.7.3 Functional Requirements

REQ-1: Clicking on search bar – The application will have a search bar right on the top of the screen, where the user can search for doctor or ambulances if he/she is a patient. Although the UI will be different for users who are doctors and ambulances, they will not have search option to work with.

REQ-2: Choosing doctor – The application will show various options such as type of doctor which includes, orthopedic, pediatrician, ophthalmologist, and many more along with date of appointment.

REQ-3: Choosing Ambulance – The application will show various options such as type of vehicle, pickup location and whether or not it is an emergency or not.

4.8 Booking (Patients)

4.8.1 Description and Priority

This feature allows the user if he/she is a patient, to book a specific doctor or ambulance as per his/her requirements.

4.8.2 Stimulus/Response Sequences

1. **Stimulus** – The user clicks on the “book” option beside the name and details of the doctor.
   Response – A window with the location of the chamber or hospital opens up.

2. **Stimulus** – The user selects the option of “confirm booking”.
   Response – A certificate like document will be generated showing all the details of the appointment.

3. **Stimulus** – The user selects “Go to payment”.
   Response – The user will be redirected to the payment portal.

4.8.3 Functional Requirements

REQ-1: Clicking on Book – After the user has searched for the specific doctor along with the appointment details and date of appointment, the application will show an option of book on the bottom left corner of his name and contact details.

REQ-2: Confirm Booking – After the user selects “Book”, the application will take him to a page where all the details regarding the doctor is given such as, name of doctor, contact details, where he/she would be available on that date and when to visit the chamber. The user after viewing all the above details can press confirm booking.

REQ-3: Going for payment – After the user selects “Confirm booking”, the application will generate a certificate like document where every details will be displayed along with the patient name, which is also like a booking receipt. The user cannot take a screenshot here until the payment is completed.

4.9 Appointment (Doctors)

4.9.1 Description and Priority

This feature allows the user if he/she is a doctor, to go through the appointments scheduled for that particular day. The UI of the application will be different for a doctor and ambulance compared to a patient.

4.9.2 Stimulus/Response Sequences

1. **Stimulus** – The user if he/she is a doctor opens the app.
   Response – The options of past appointments and appointments for the day appears in front of him.

2. **Stimulus** – The user selects the option of “past appointments”.
   Response – A list of all the past appointments along with details of the patient comes up.

3. **Stimulus** – The user selects “Appointments for today”.
   Response – The list of the all the appointments scheduled for that particular day opens up along with the details of the patient.

4. **Stimulus** – The user selects “confirm appointment” beside the patient name.
   Response – A message will be displayed showing that the notification has gone to the patient.

5. **Stimulus** – The user selects “reject appointment” beside the patient name.
   Response – A box will appear where the doctor has to mention the reason behind his/her rejecting the appointment and also the option of online prescription will come up.

6. **Stimulus** – The user selects “online prescription” beside the patient name.
   Response – A form will open up where he will prescribe the medicines and the necessary steps to be done.

4.9.3 Functional Requirements

REQ-1: UI for Doctors – If the user is a doctor and he opens the application the only thing that he will be able to see is the past appointments and appointments scheduled for that particular day. He has an option of viewing the past appointments or the current appointments whichever he finds suitable or necessary.

REQ-2: Past Appointments – After the user selects the option of “Past Appointments”, a list of all his/her past appointments with the patient details and prescription will appear and he might check through any of the patient’s record.

REQ-3: Appointments for the day – After the user selects “Appointments for the day”, a list of all the appointments scheduled for that day will open in front of him with the option of confirm, reject and online prescription.

REQ-4: Managing Appointments – When the user selects the option of "Appointments for the day" a list of all the current appointments comes up and the user can reject, accept or upload prescription online. When he accepts the appointment, a notification will be sent to the patient, and when he rejects he either has to mention the reason and reschedule or post a online prescription for the same.

4.10 Booking (Ambulances)

4.10.1 Description and Priority

This feature allows the user if it is an ambulance, to go through the bookings made to it. The UI of the application will be different for the ambulance also as compared to a patient and the doctor.

4.10.2 Stimulus/Response Sequences

1. **Stimulus** – The user if he/she is an ambulance opens the app.
   Response – The options of past bookings, present bookings and those which needs an ambulance urgently appears.

2. **Stimulus** – The user selects the option of "past bookings".
   Response – A list of all the past bookings along with details of the patient and hospital to where the patient was taken appears.

3. **Stimulus** – The user selects "Bookings for today".
   Response – The list of the all the bookings scheduled for that particular day opens up along with the details of the patient and hospital.

4. **Stimulus** – The user selects "emergency booking".
   Response – The location of the patient will be shown and a message of "Quick Action" will be shown.

5. **Stimulus** – The user selects "accept booking".
   Response – The pickup location of the patient will appear along with the nearby hospitals where the patient can be taken.

6. **Stimulus** – The user selects "reject booking".
   Response – A form will open up where the reason for rejection has to be stated.

4.10.3 Functional Requirements

REQ-1: UI for Ambulances – If the user is a doctor and he opens the application the only thing that he will be able to see is the past appointments and appointments scheduled for that particular day. He has an option of viewing the past appointments or the current appointments whichever he finds suitable or necessary.

REQ-2: Past Bookings – After the user selects the option of “Past Bookings”, a list of all past bookings with the patient details and the hospital to where the patient was taken.

REQ-3: Bookings for the day – After the user selects the option of “Past Bookings”, a list of all past bookings with the patient details and the hospital to where the patient was taken.

REQ-4: Emergency Booking – The application will provide the option of emergency booking to the patient, and the ambulance user will have to reach the destination as soon as possible, if the user rejects an emergency booking then the reason has to be stated as to why the booking was rejected and there will be a cut in payment.

4.11 Payment Window

4.11.1 Description and Priority

This feature takes the user if he/she is a patient to the payment portal where he has the option of online payment via UPI/Debit Card/Credit Card or offline payment that is paying it directly to hospital or chamber.

4.11.2 Stimulus/Response Sequences

1. **Stimulus** – The user if he/she is a patient clicks on the “Book” option beside the details of the doctor and chamber.
   Response – A portal opens up with two options of online payment or offline payment.

2. **Stimulus** – The user selects the option of “online payment”.
   Response – A list of options comes up like UPI, debit or credit card.

3. **Stimulus** – The user selects “UPI”.
   Response – A form will open where he has to give his/her UPI details and confirm the payment from the UPI application.

4. **Stimulus** – The user selects “debit card”.
   Response – A form will open where the user has to give his debit card details and confirm the booking procedure.

5. **Stimulus** – The user selects “credit card”.
   Response – A form will open where the user has to give his credit card details and confirm the booking procedure.

6. **Stimulus** – The user selects “offline booking”.
   Response – A certificate like document will generate where a message will be written in red that ***“Payment is pending***”.

7. **Stimulus** – The user takes more time than the required time while paying.
   Response – The payment will be failed and the user will be taken to the booking portal again and he/she has to redo the process again.

4.11.3 Functional Requirements

REQ-1: Payment Portal – After the user clicks on the “confirm booking” option, he will be redirected to the payment portal where the options of online and offline booking will appear in front of him.

REQ-2: Online payment – After the user clicks on online payment, options of UPI, debit card and credit card will appear in front of him and the user can choose any one of the above options for the payment.

REQ-3: Offline payment – After the user clicks offline payment option, a certificate like document will be generated where a message will be written “***Payment is pending”.*** The user has to produce this document when he visits the doctor either to doctor or in the chamber or hospital.

REQ-4: Payment Failed – The user has to pay the amount within a specific amount of time, like 5 minutes or so, and if he/she fails to do so, then a message of “Payment failed” will be displayed and the user will be redirected to the booking page.

4.12 Receipt Download

4.12.1 Description and Priority

This feature will generate a pdf document where the details of payment and appointment will be there. This pdf document will be like a receipt and has to be presented in the chamber as a proof.

4.12.2 Stimulus/Response Sequences

1. **Stimulus** – The user pays the required amount.
   Response – A message is being displayed “Payment successful, download receipt”.

2. **Stimulus** – The user selects the option of “download receipt”.
   Response – A pdf will be downloaded that will be shown on the notification tab.

4.12.3 Functional Requirements

REQ-1: Payment – After the user pays the required amount using online methods or offline, a message will be displayed showing that “Payment is successful” and the option of download receipt will appear.

REQ-2: Downloading receipt – After the option of download receipt appears on the screen the user presses the download option and the receipt will be downloaded which has to be produced in front of the chamber or hospital or the doctor himself/herself.

4.13 Manage Accounts

4.13.1 Description and Priority

This feature allows the admin of the application to view the activities of all the users and update their accounts. This feature is only available to the ***admin*** of the app.

4.13.2 Stimulus/Response Sequences

1. **Stimulus** – The user logs in as admin and requests the details of user accounts.
   Response – A list is displayed showing all the users along with user-type, the time when they last used the app and all the personal information.

2. **Stimulus** – The user selects the option of “notify user”.
   Response – The system will automatically send a notification to the user and a message of “User notified successfully” will be displayed.

3. **Stimulus** – The user selects the option of “delete account”.
   Response – A pop-up will come where it will ask whether the admin really wants to delete the app. On clicking “yes” the account will be deleted.

4.13.3 Functional Requirements

REQ-1: Requesting details – After the admin has logged into the application, the UI which will be shown to him will be different and he/she can ask for the details of the particular user by clicking on his/her name.

REQ-2: Notify User – The admin can view that whether a particular account has been properly updated or not, if he/she finds that an account is not updated properly, the admin can notify the user. On clicking the option of “notify user” a message will go to that respective user and a message of “user notified successfully” will be displayed to the admin.

REQ-3: Delete Account – If the admin finds out, that an account is not properly updated and the details of an account does not seem relevant the admin can delete that account. The admin clicks on the option of “delete account” beside the account name and a pop-up will be displayed where he/she will be asked to confirm the deletion and then the account will be deleted.

4.14 Manage Transactions

4.14.1 Description and Priority

This feature allows the ***admin*** of the application to view the transactions that the user (patient only) has made recently or sometime back. This feature is only available to the ***admin*** of the app.

4.14.2 Stimulus/Response Sequences

1. **Stimulus** – The user logs in as admin and clicks on the option of “Manage Transactions”.
   Response – All the transactions will be displayed in front of him with the option present transactions and past transactions.

2. **Stimulus** – The user selects the option of “notify user”.
   Response – The system will automatically send a notification to the user regarding the transaction if it is not made, and a message of “User notified successfully” will be displayed.

3. **Stimulus** – The user selects the option of “present transactions”.
   Response – A list of all the transaction made in the last two days will be shown.

4. **Stimulus** – The user selects the option of “past transactions”.
   Response – A list of all the transaction made in the last six months will be shown.

4.14.3 Functional Requirements

REQ-1: Manage Transactions – After the admin has logged into the application, the UI which will be shown to him will be different and he/she can check the past and present transactions made through the application by clicking on the option of “Manage transaction”.

REQ-2: Notify User – The admin can view that whether a particular transaction has taken place successfully or not, in case of any discrepancy from the user side the admin has the option of notifying him about it.

REQ-3: Present Transactions – The admin can check all the transactions that are being made in the last two days by clicking on the option of “present transactions”.

REQ-4: Past Transactions – The admin can check all the transactions that are being made in the last six months by clicking on the option of “past transactions”.

4.15 Manage Feedback

4.15.1 Description and Priority

This feature allows the ***admin*** of the application to view the feedbacks that the user a patient/doctor/ambulance has given. This feature is only available to the ***admin*** of the app.

4.15.1 Stimulus/Response Sequences

1. **Stimulus** – The user logs in as admin and clicks on the option of "Manage Feedback".
   Response – All the feedbacks will be displayed in front of him that a particular user has made some time back.

2. **Stimulus** – The user selects the option of "view doctor's feedback".
   Response – All the feedbacks that the patient has made for a doctor will be displayed.

3. **Stimulus** – The user selects the option of "view patient's feedback".
   Response – All the feedbacks that the doctor or an ambulance has made for a patient will be displayed.

4. **Stimulus** – The user selects the option of "view ambulance feedback".
   Response – All the feedbacks that the patient has made for an ambulance will be displayed.

4.15.3 Functional Requirements

REQ-1: Manage Feedbacks – After the admin has logged into the application, the UI which will be shown to him will be different and he/she can check the feedbacks of the three users of his/her application that is the doctor, patient and the ambulance.

REQ-2: View Doctor's feedback – The admin can view the feedback that a patient has made for a particular doctor and if the feedbacks are repeatedly negative then the admin can delete the particular doctor's account from their application.

REQ-3: View Patient's feedback – The admin can view the feedback that a doctor or an ambulance has made for a particular patient and if the feedbacks are repeatedly negative then the admin can delete the particular patient's account from their application.

REQ-4: View Ambulance's feedback – The admin can view the feedback that a patient has made for a particular ambulance and if the feedbacks are repeatedly negative then the admin can delete the particular ambulance's account and its service from their application.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The application is web-based, therefore, along with downloading it in a device, a proper web browser is also needed for the proper working of the application.
- Since the application is only concerned with the booking and checking of appointments, a proper internet connection is also required for proper use of the application.
- The response time after the payment should also be reduced, so that the fear of losing money is avoided.
- The hardware components of a user's machine should be also be proper so that application can run properly without utilizing much of the resources.

5.2 Safety Requirements

- The system where the application is said to run should not have any viruses or malware in it.
- The device/system must maintain the activity logs.
- The information that is being provided by the user should be stored in any database and the security and privacy of that database must be maintained. End-to-end encrypted messages should be used so that the data of the user is not compromised.

5.3 Security Requirements

- The application does not let the users do anything on the application other than viewing unless his/her details have been verified.
- During the payment, if the user is paying online, after a time span of 2 minutes the system automatically takes the user back, displaying the message of payment unsuccessful.
- The servers and the databases of the application should only be made available to the admin.
- The login details of the user must be encrypted properly so as to avoid any unauthorized access.
- The system should not leave any cookies on the user's machine, containing in the user's password not until the user himself/herself wants the system to remember the credentials.

5.4 Software Quality Attributes

- The user interface of the application has been made simple so that any kind of user can have a proper experience on the app.
- The application is made keeping mind the Microsoft graphical user interface, that is the GUI is simple to understand and access without any complex designs.
- The pages should load with minimum latency and must recover from system failures.
- The database should be designed in a simple manner so that it can be easily maintained by the administrators.
- The elements of the application must be easily accessible corresponding to user's internet connection.

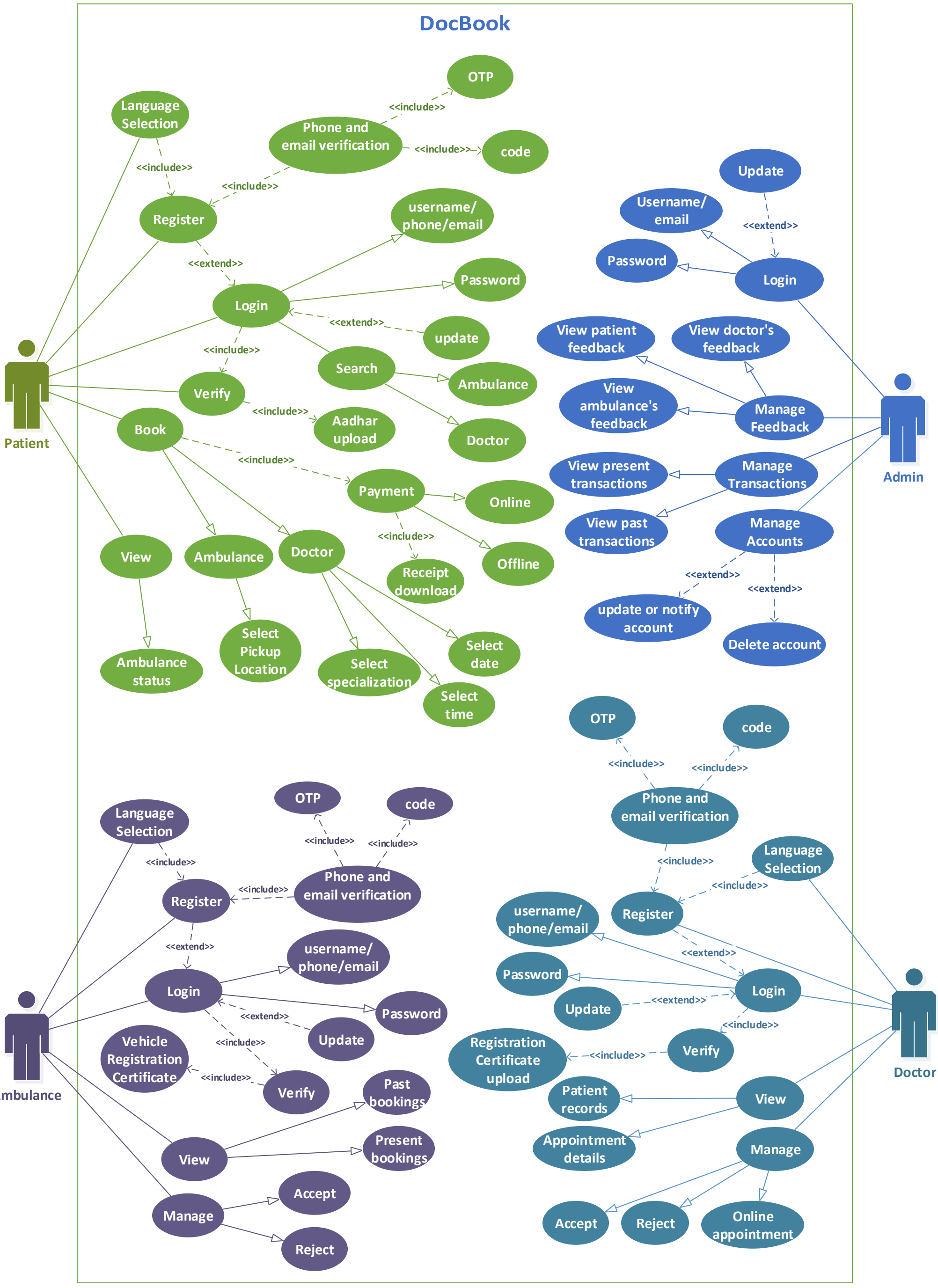
5.5 Business Rules

- B1 – Synchronization of Metadata
- B2 – Content Workflow
- B3 – Transactional Services
- B4 – Self Service

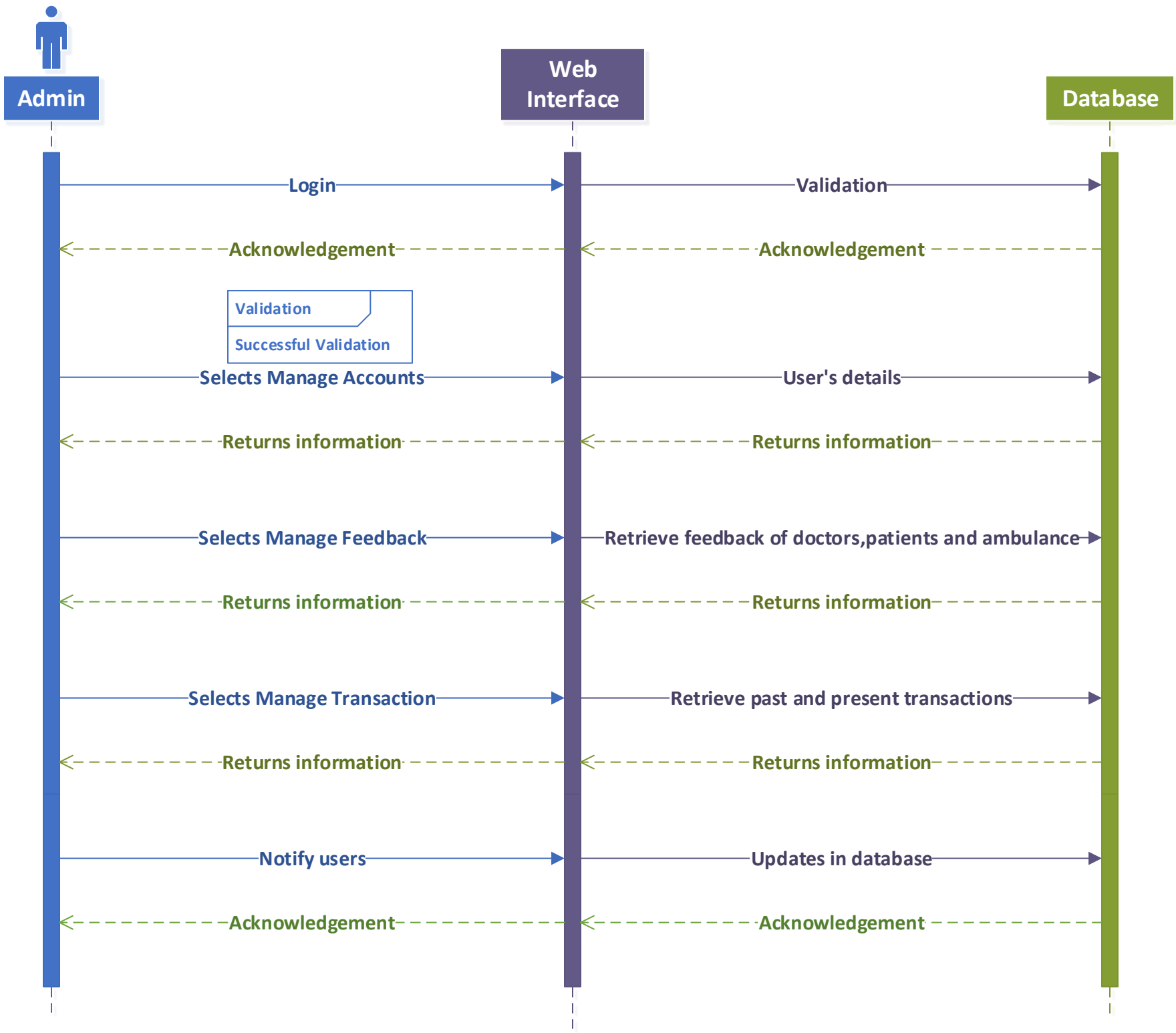
Appendix A: Glossary

- SRS – Software Requirement Specification.
- REQ – Requirements.
- B – Business Rules.

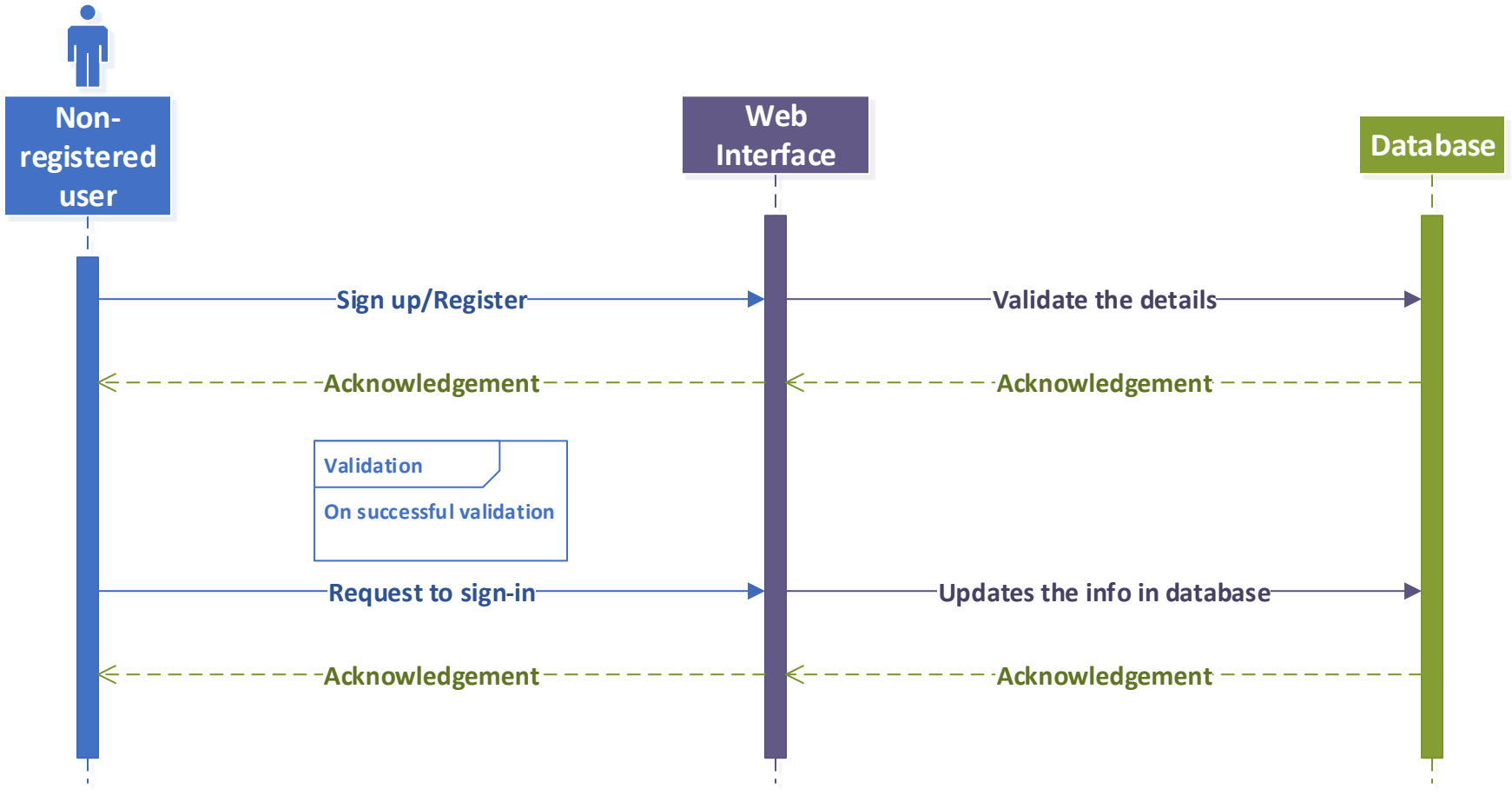
USE CASE DIAGRAM



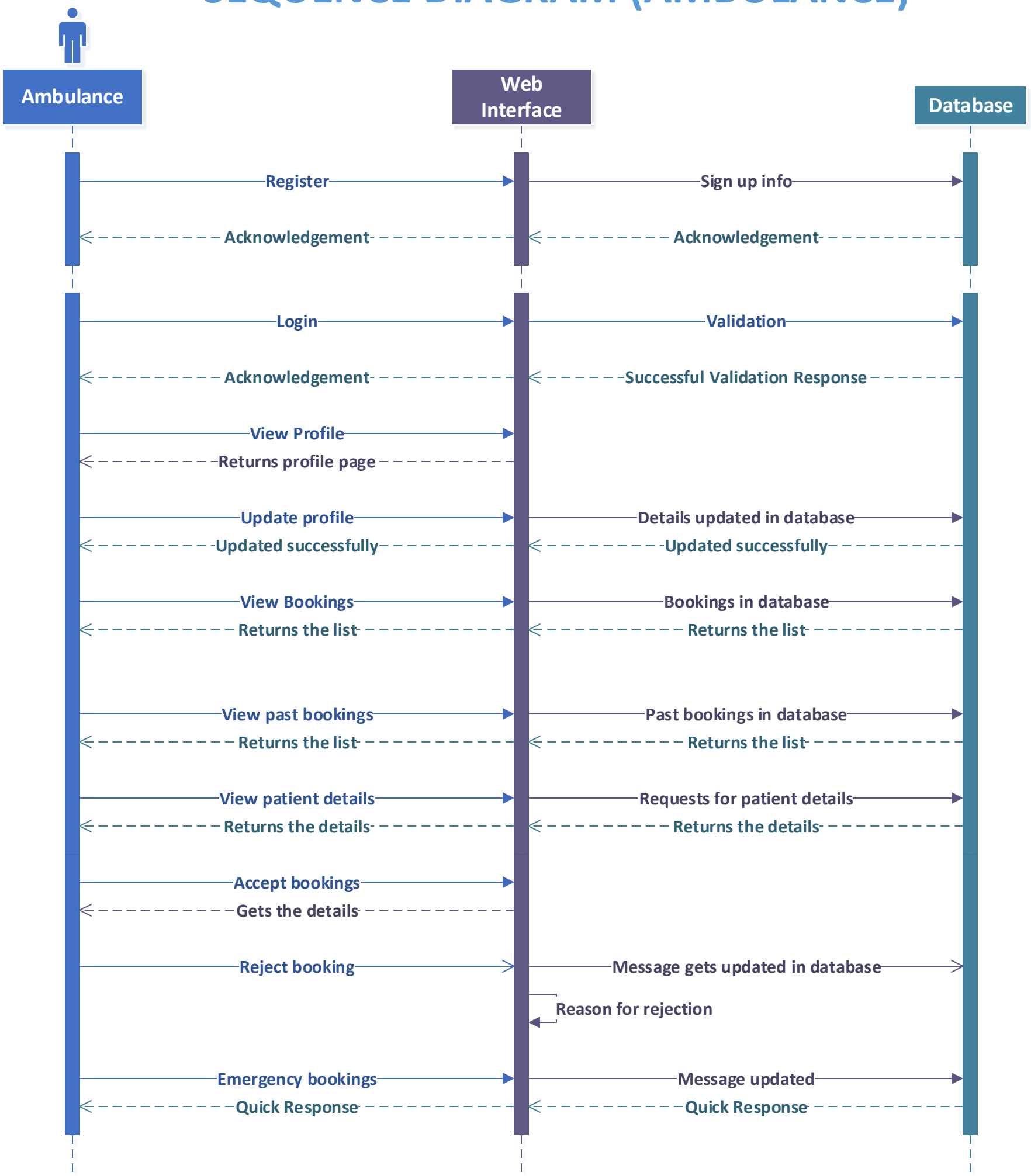
SEQUENCE DIAGRAM (ADMIN)



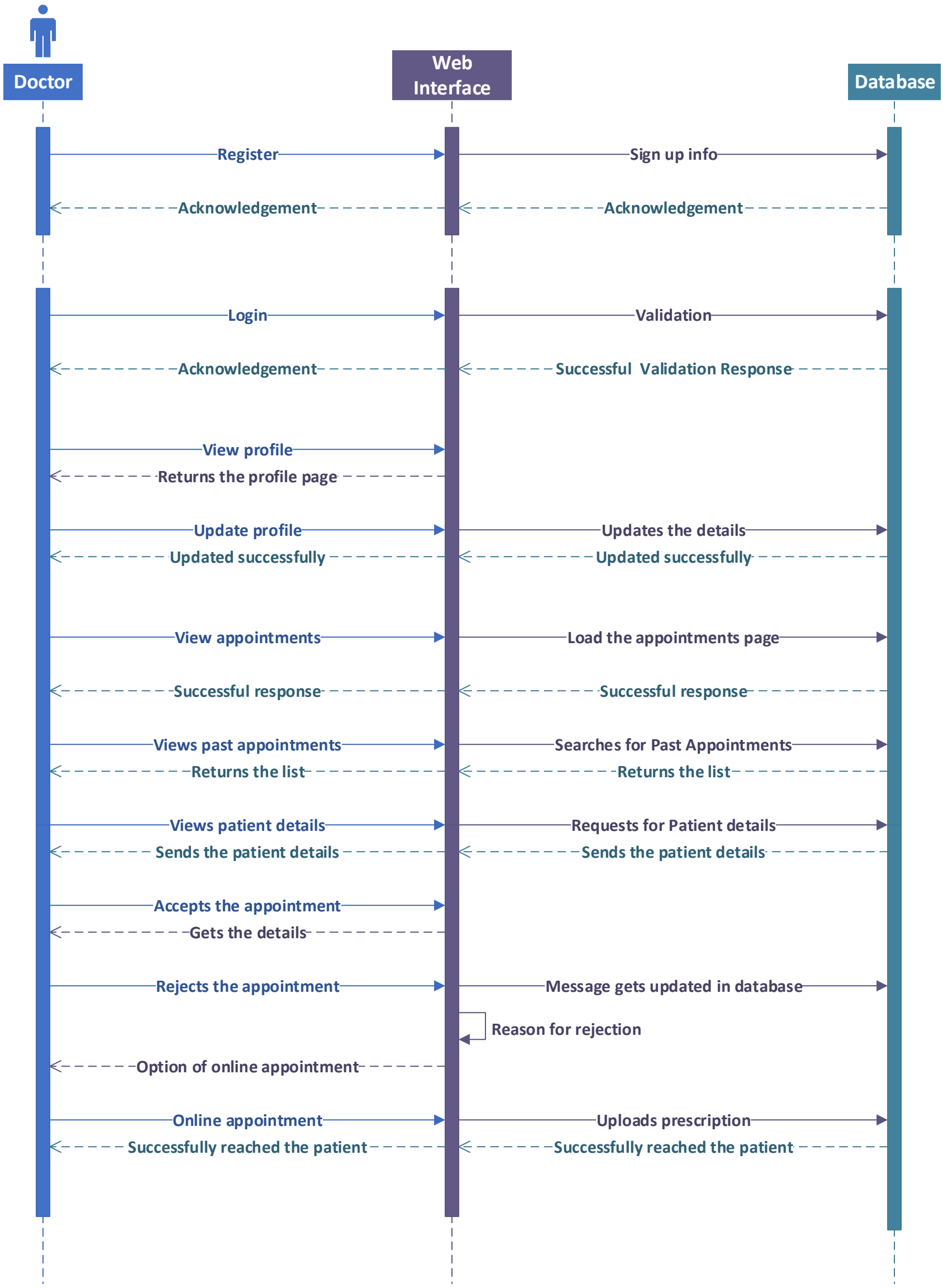
SEQUENCE DIAGRAM (NON-REGISTERED USERS)



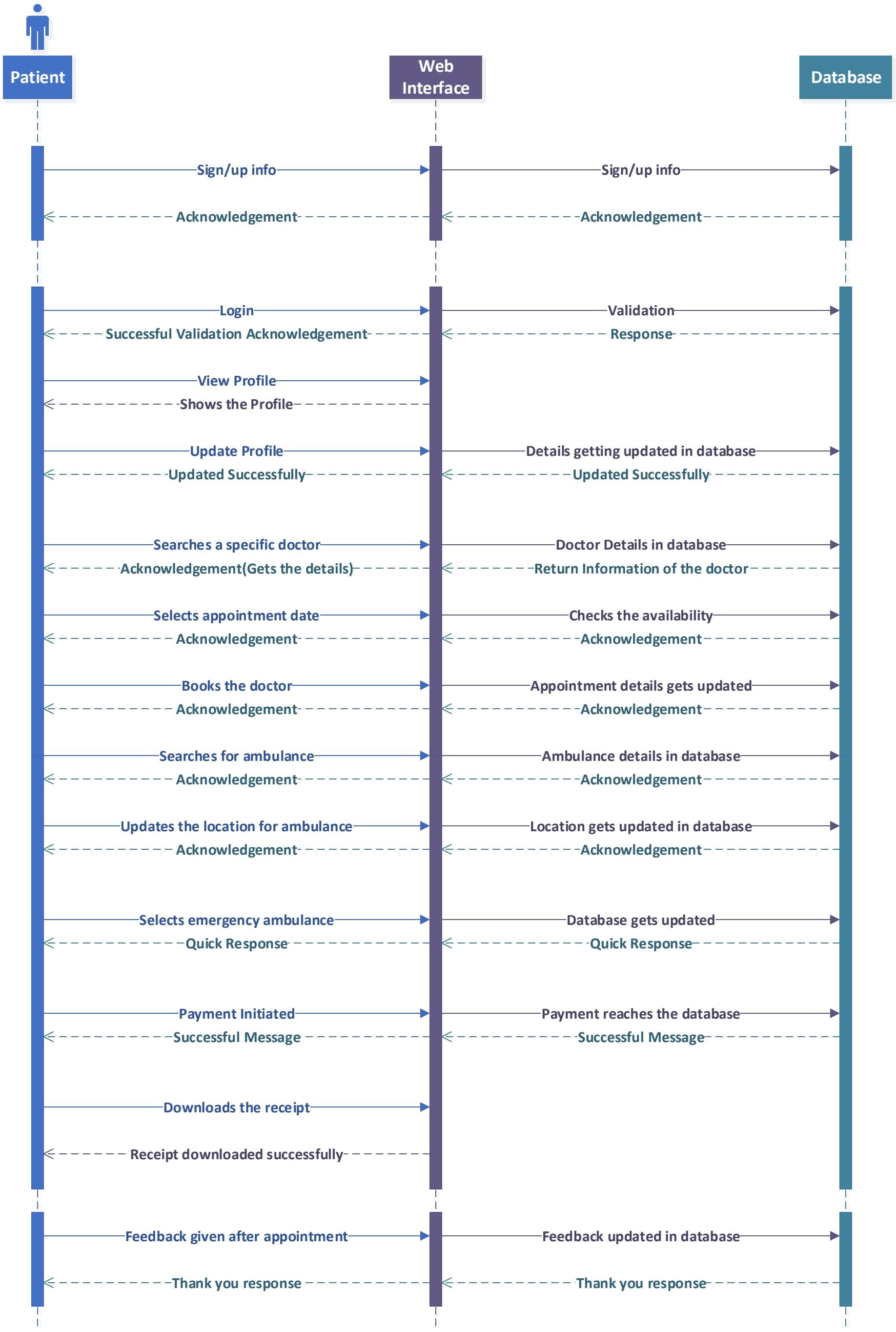
SEQUENCE DIAGRAM (AMBULANCE)



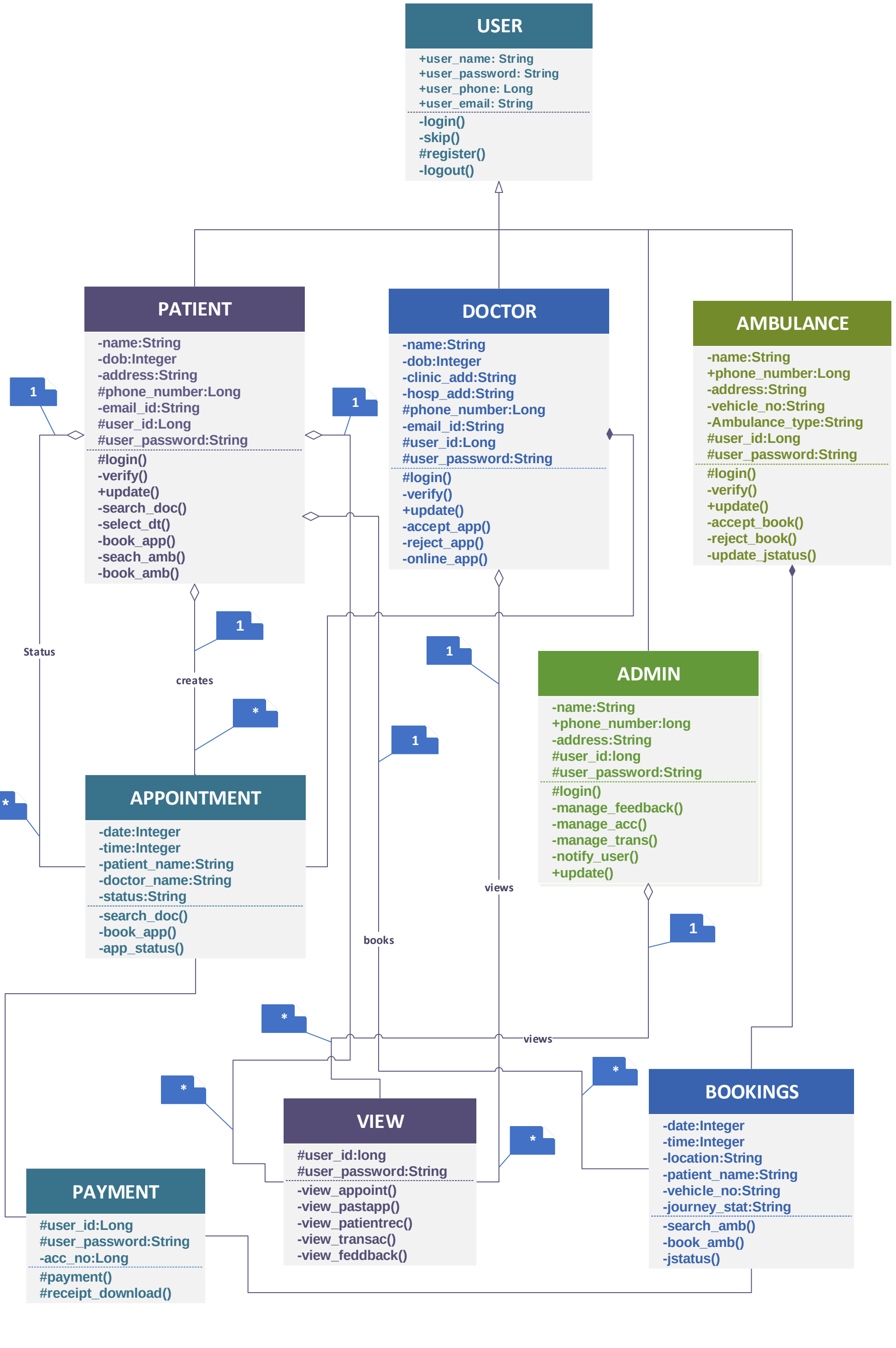
SEQUENCE DIAGRAM (DOCTOR)



SEQUENCE DIAGRAM (Patient)



CLASS DIAGRAM



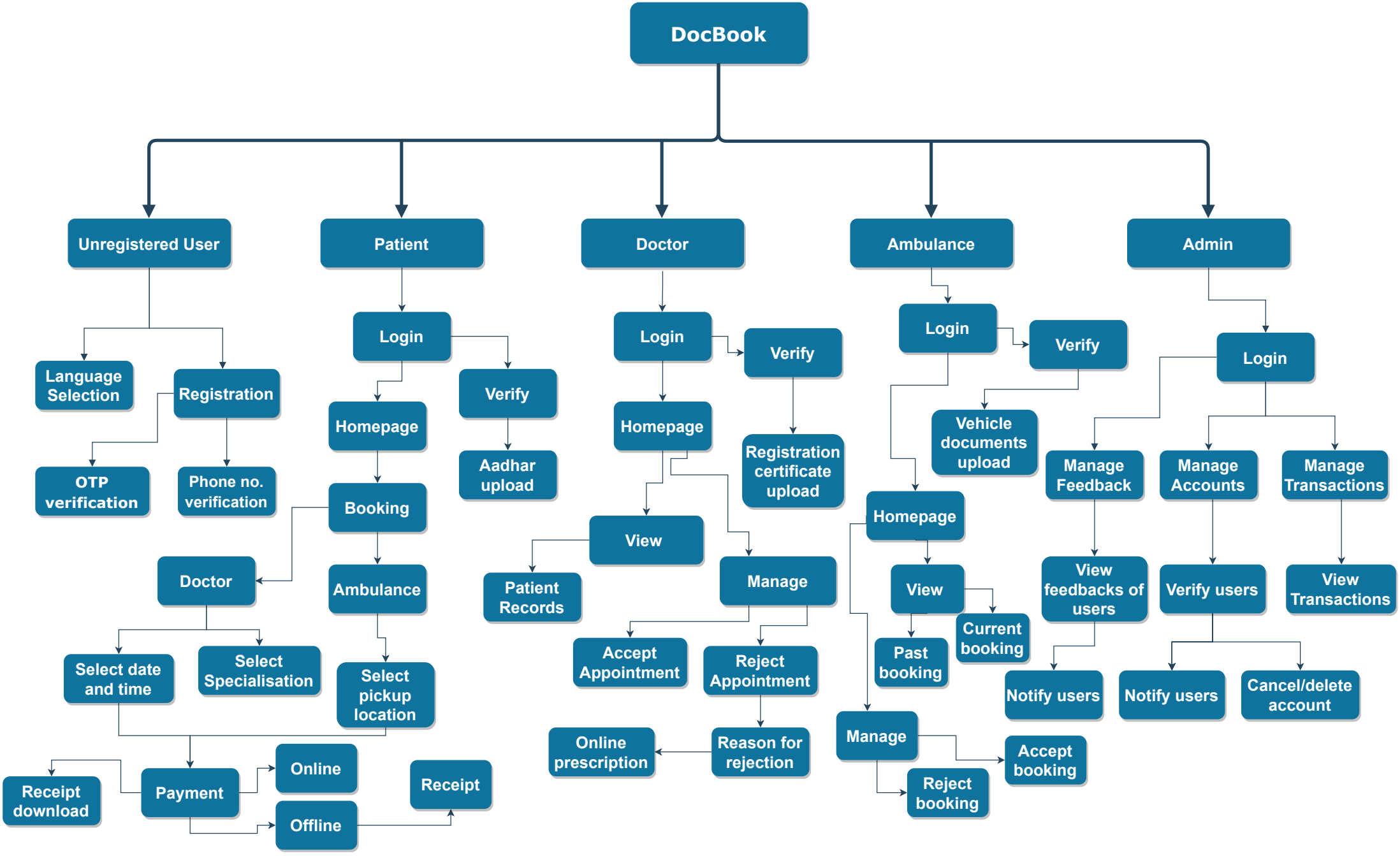
Test Cases for:	***DocBook***
Author	*Roudranil Kar*
Date Created	*20th January, 2023*
Last Updated	*Not Applicable*
Prerequisites:	*The user has a proper internet connection and a valid device to use the application*
System Module:	***Booking/Payment***

Test Case #	Test Case Description	Test Procedure	Input Data	Expected Result	Comments
1	Booking (Doctor)	1) User searches for a specific doctor. 2) User selects the required doctor. 3) User selects the "book" option.	The user gets the options of online or offline payment. User selects anyone.	Booking and payment done successfully.	Booking was done successfully.
2	Booking (Ambulances)	1) User searches for an ambulance. 2) User selects the ambulance. 3) User enters the pickup location.	The user gets the options of online or offline payment. User selects anyone.	Booking and payment done successfully.	Booking was done successfully.
3	Online payment (debit)	1) User selects the option of "online payment" 2) User selects the option of pay by debit. 3) User enters the debit card details. 4) User selects "pay".	The correct 6-digit otp sent to his registered mobile number.	Payment successful.	The payment was successful and the option of receipt download will be shown.
4	Online payment (UPI)	1) User selects the option of "online payment" 2) User selects the option of pay by UPI. 3) User enters the debit card details. 4) User selects "pay".	User pays from his upi app the amount which will be displayed correctly.	Payment successful.	The payment was successful and the option of receipt download will be shown.
5	Online payment (debit)	1) User selects the option of "online payment" 2) User selects the option of pay by debit. 3) User enters the debit card details. 4) User selects "pay".	Incorrect otp is entered.	Payment unsuccessful.	The payment was unsuccessful and the user has to repeat the whole process again.
6	Online payment (UPI)	1) User selects the option of "online payment" 2) User selects the option of pay by UPI. 3) User enters the debit card details. 4) User selects "pay".	User does not open the upi app within a specific time.	Payment unsuccessful.	The payment was unsuccessful and the user has to repeat the whole process again.
7	Offline payment.	User selects the option of "offline payment"	No input as such.	Receipt will be downloaded.	The payment will be made offline that is when the user visits the hospital.

Test Cases for:	***DocBook***
Author	*Roudranil Kar*
Date Created	*20th January, 2023*
Last Updated	*Not Applicable*
Prerequisites:	*The user has a proper internet connection and a valid device to use the application*
System Module:	***Verification***

Test Case #	Test Case Description	Test Procedure	Input Data	Expected Result	Comments
1	Verification (Doctor)	1) User successfully logs in. 2) User clicks on "verify" option.	User if he/she is a doctor uploads his/her registration certificate.	Verification of user successful	Verification is successful and the user can continue to use the application.
2	Verification (Patient)	1) User successfully logs in. 2) User clicks on "verify" option.	User if he/she is a doctor uploads his/her aadhar card for verification.	Verification of user successful	Verification is successful and the user can continue to use the application.
3	Verification (Ambulance)	1) User successfully logs in. 2) User clicks on "verify" option.	User if it is an ambulance user, uploads the vehicle registration certificate.	Verification of user successful	Verification is successful and the user can continue to use the application.
4	Verification (Patient)	1) User successfully logs in. 2) User clicks on "verify" option.	User if he/she is a patient enters wrong aadhar details.	Verification unsuccessful	Verification is unsuccessful and the option will freeze for few hours.
5	Verification (Doctor)	1) User successfully logs in. 2) User clicks on "verify" option.	User if he/she is a doctor enters wrong invalid registration certificate.	Verification unsuccessful	Verification is unsuccessful and the option will freeze for few hours.
6	Verification (Ambulance)	1) User successfully logs in. 2) User clicks on "verify" option.	User if it is an ambulance enters wrong vehicle details.	Verification unsuccessful	Verification is unsuccessful and the option will freeze for few hours.

MODULARISATION DOCUMENT



List of functions:

Language selection:

1. selectlang()
2. help_support()
3. autolangdetect()
4. display()
5. report()

Registration

1. accept_num()
2. accept_email()
3. verify_email()
4. verify_num()
5. otp_send()
6. resend_otp()
7. verify_again()
8. create account()

Login

1. login()
2. verify_yourself()
3. view_profile()
4. forgotpassword()
5. help_support()
6. search()
7. homepage()
8. daily_updates()

Homepage

1. search()
2. verify_yourself()
3. view_profile()
4. help_support()
5. forgotpassword()
6. search()
7. choose_service()
8. booking()
9. view_appointments()
10. view_records()
11. view_bookings()

12. manage_appointments()
13. manage_bookings()

Search

1. search_box()
2. filter()
3. sort()
4. suggestions()

Verify

1. choose_again()
2. upload_govt_cer()
3. upload_doc_regcer()
4. upload_vehicle_regdocs()
5. verify()
6. display()

Booking

1. choose()
2. select_specialisation()
3. select_date()
4. select_time()
5. select_location()
6. display_availability()
7. payment()

Payment

1. choose_mode()
2. onlinepay()
3. offlinepay()
4. display()
5. upi()
6. debit_card()
7. credit_card()
8. receipt_download()
9. repay()
10. report()
11. paybywallet()

View (Doctor)

1. view_patientrec()
2. pastrecords()
3. download_records()
4. view_prescriptions()
5. view_current_bookings()
6. manage_appointments()

Manage (Doctor)

1. view()
2. accept_appointment()
3. reject_appointment()
4. reason_reject()
5. online_appointment()
6. patient_query()
7. prescription_upload()

View (Ambulances)

1. view_past_bookings()
2. view_present_bookings()
3. view_emergency_bookings()
4. manage_bookings()
5. download_receipt()

Manage (Ambulances)

1. view()
2. accept_booking()
3. accept_emergency_booking()
4. reject()
5. reason_reject()
6. redirect_gmap()
7. update_location()

Admin

1. login()
2. checkupdates()
3. homepage()
4. manage_feedback()
5. manage_accounts()
6. manage_transactions()
7. view_feedbacks()
8. notify_user()
9. update_notify()
10. check_account()
11. delete_account()
12. view_transactions()

Feedback

1. topatient()
2. todoctor()
3. toAmbulance()
4. feedback_box()
5. feedback_email()
6. feedback_admin()

Function Point Analysis (FPA) for DocBook

Measurement Parameter	Count				Weightage				FP Count
	Simple	Average	Complex		Simple	Average	Complex		
# of External Inputs (EI)	92	40	10	X	3	4	6	=	496
# of External Outputs (EO)	120	50	10	X	4	5	7	=	800
# of External Inquiries (EQ)	45	25	5	X	3	4	6	=	265
# of Internal Logical Files (ILF)	100	65	15	X	7	10	15	=	1,575
# of External Interface Files (EIF)	100	70	20	X	5	7	10	=	1,190
Unadjusted Function Points (UFP)								**UFP =**	4,326

General System Characteristics (GSC): *- rate each factor below on a scale of 0 to 5 for DI*	0 = No Influence	1 = Incidental	2 = Moderate	**DI**
Degree of Influence (DI)	3 = Average	4 = Significant	5 = Essential	
01. Does the system require reliable backup and recovery?				4
02. Are data communications required?				3
03. Are there distributed processing functions?				2
04. Is performance critical?				4
05. Will the system run in an existing, heavily utilized operational environment?				4
06. Does the system require on-line data entry?				5
07. Does the on-line data entry require the input transaction to be built over multiple screens or operations?				4
08. Are the master files updated on-line?				1
09. Are the inputs, outputs, files, or inquiries complex?				0
10. Is the internal processing complex?				2
11. Is the code designed to be reusable?				5
12. Are conversion and installation included in the design?				4
13. Is the system designed for multiple installations in different organizations?				5
14. Is the application designed to facilitate change and ease of use by the user?				5
			Total DI =	48
Value Adjustment Factor (VAF)			**VAF =**	1.13
Adjusted Function Points (AFP)			**AFP =**	4,888

Any Assumptions made should be listed below.

Constructive Cost Effective Model (COCOMO)

Basic COCOMO:

Modules	No. of Functions
1. Language Selection	5
2. Registration	8
3. Login	8
4. Homepage	13
5. Search	4
6. Verify	6
7. Booking (Patient)	7
8. Payment (Patient)	11
9. View (doctor)	6
10. Manage (doctor)	7
11. View (Ambulance)	5
12. Manage (Ambulance)	7
13. Admin	12
14. Feedback	6
TOTAL	**105**

Type of Project: Semi-detached

Assumption (based on Coding Standards/Guidelines): Each function has approximately 10 lines of code (LOC)

Total lines of Code (LOC): 105*10 = 1050

Thus, KLOC = 1050/1000 = 1.05

Effort: 3.0 * (KLOC) ^1.12 PM = 3.0 * (1.05) ^1.12 PM = 3.16PM ~ 3PM

Development Time: 2.5 * (Effort) ^0.35 months = 2.5 * (3.16) ^0.35 months = 3.73 months ~ 4 months

Intermediate COCOMO:

Cost Drivers	**Ratings**
Product Attributes:	
Required Software Reliability (RELY)	High(1.08)
Size of Application Database (DATA)	Extra High(1.22)
Complexity of the Product (CPLX)	Normal(1.00)
Hardware Attributes:	
Run-time performance constraints (TIME)	High(1.08)
Memory Constraints (STOR)	Normal(1.00)
Volatility of the virtual machine environment (VIRT)	Low(0.87)
Required turnabout time (TURN)	Normal(1.00)
Personnel Attributes:	
Analyst Capability (ACAP)	High(1.08)
Applications experience (AEXP)	Normal(1.00)
Software engineer capability (PCAP)	Normal(1.00)
Virtual machine experience (VEXP)	Low(0.97)
Programming language experience (LEXP)	Very High(1.15)
Project Attributes:	
Application of software engineering methods (MODP)	Normal(1.00)
Use of software tools (TOOL)	High(1.08)
Required development schedule (SCED)	Normal(1.00)

Effort Adjustment Factor (EAF): (1.08 * 1.22 * 1.00 * 1.08 * 1.00 * 0.87 * 1.00 * 1.08 * 1.00 * 1.00 * 0.97 * 1.15 * 1.00 * 1.08 * 1.00) = 1.61

Effort = 2.4 * (KLOC) ^1.05 * EAF = 2.4 * (1.05) ^1.05 * 1.61 = 4.06 ~ 4PM