Software Requirements Specification

Version 3.0

MEDICO

Group 13

|  |  |
| --- | --- |
| **Member Name** | **Member Roll #** |
| Laiba Imran | 18L-1136 |
| Fatima Mehmood | 18L-0945 |
| Arooba Ahmad | 18L-0991 |
| Hizafa Nadeem | 18L-1105 |
| Memoona Ishaq | 18L-2131 |

**Table of Contents**

[Revision History iv](#_Toc75367365)

[1. Introduction 1](#_Toc75367366)

[1.1 Product 1](#_Toc75367367)

[1.2 Scope 1](#_Toc75367368)

[1.3 Business Goals 1](#_Toc75367369)

[1.4 Document Conventions 1](#_Toc75367370)

[1.5 References 1](#_Toc75367371)

[2. Functional Requirements 2](#_Toc75367372)

[2.1 Sign Up 2](#_Toc75367373)

[2.2 Login 3](#_Toc75367374)

[2.3 Edit profile 4](#_Toc75367375)

[2.4 Cancel Appointment 5](#_Toc75367376)

[2.5 View Appointment History 6](#_Toc75367377)

[2.6 Get Suggestion of highly rated doctors 6](#_Toc75367378)

[2.7 Selecting COVID cell 7](#_Toc75367379)

[2.8 Set medicine reminders 7](#_Toc75367380)

[2.9 Set Appointment Reminder 9](#_Toc75367381)

[2.10 Search doctor 10](#_Toc75367382)

[2.11 View Doctor’s Profile 11](#_Toc75367383)

[2.12 Book Appointment 11](#_Toc75367384)

[2.13 Rate Doctors 12](#_Toc75367385)

[2.14 Add Doctor’s Availability 12](#_Toc75367386)

[2.15 Cancel Doctor’s Availability 13](#_Toc75367387)

[2.16 COVID Cell Registration 14](#_Toc75367388)

[2.17 Book Appointment in COVID Cell 15](#_Toc75367389)

[3. Nonfunctional Requirements 16](#_Toc75367390)

[3.1 Performance Requirements 16](#_Toc75367391)

[3.2 Security Requirements 16](#_Toc75367392)

[3.3 Usability Requirements 16](#_Toc75367393)

[3.4 Reliability Requirements 16](#_Toc75367394)

[Appendix A: Data Flow Diagram 17](#_Toc75367395)

[1. Data Flow diagram 17](#_Toc75367396)

[Appendix B: Analysis Model 19](#_Toc75367397)

[1. Class Diagram 19](#_Toc75367398)

[2. Use Case Diagram 20](#_Toc75367399)

[Appendix C: Design Models 21](#_Toc75367400)

[1. Architectural Diagram 21](#_Toc75367401)

[2. Component-Level Diagram 22](#_Toc75367402)

[Appendix D: Screenshots 23](#_Toc75367403)

[Appendix E: Test Cases 28](#_Toc75367404)

[1. Flow-chart Diagram 28](#_Toc75367405)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name(s)** | **Date** | **Reason(s) For Change(s)** | **Version** |
| Software Requirements Specification MEDICO | 26th May, 2021. | Addition of Appendix B: Analysis Model  Addition of Appendix C: Design Models | 1 |
| Software Requirements Specification MEDICO | 23rd June, 2021. | Addition of Appendix D: Screenshots  Addition of Appendix E: Test Cases | 2 |

# 1. Introduction

## 1.1 Product

The purpose of this document is to present a detailed description of the Medico, an online Medical Facility application. This document will explain the purpose and features of the system, the interfaces of the system, what the system will do and the constraints under which it must operate. This document is intended for both the stakeholders and the developers of the system.

## 1.2 Scope

This application shall provide an online consultation platform that connects patients with the licensed doctors, 24/7.It will allow users to first register as patients and then interact with the doctors through booking an online appointment with them. This system will also allow users to register as doctors and hence, provide consultations to the patients while remaining easy to understand and use. It will facilitate communication between a doctor and a patient and allow doctors to give online prescriptions to their patients. The system shall also provide a dedicated section for COVID-19 patients to get urgent diagnosis andconsultation through any available doctor at the time.

Furthermore, the system shall allow patients to make payments for their appointments with the doctors.

## 1.3 Business Goals

This system will provide an opportunity to earn for freshly graduated unemployed doctors. It shall also allow old aged doctors who are unable to travel, to make money in the comfort of their homes. Due to the rising cases of COVID-19, this application shall provide doctors an opportunity to cater more quarantined patients during their hospital breaks or other free slots.

## 1.4 Document Conventions

This SRS document follows an Arial font style with a size 11 for text and Times font style with size 14 for headings.

For the tables, Verdana font style is used with a size 11 for text and Times font style with 14 size for headings.

## 1.5 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998

# 2. Functional Requirements

* The application shall provide a means to sign up for the user.
* The application shall provide a mean to login for the user.
* The application shall allow the user to edit his/her profile.
* The application shall allow the doctor to enter their availability status.
* The application shall allow the patient to book an appointment with a doctor.
* The application shall allow the user to cancel appointments made.
* The application shall show the user a list of appointments made by him/her.
* The application shall provide means for the patient to rate doctors.
* The application shall provide means for the patient to get suggestions of highest rated doctors.
* The application shall allow the patient to select the COVID cell.
* The application shall allow the patient to set up a reminder for taking medications.
* The application shall allow the patient to set up a reminder for his/her booked appointment.
* The application shall allow the patient to search for a doctor by the doctor's specialty or name.
* The application shall allow the patient to view the doctor's profile.

## 2.1 Sign Up

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | | UC-1 | | |
| **Name** | | Sign up | | |
| **Summary** | | The User can sign up to create account | | |
| **Priority** | | High | | |
| **Actors** | | Doctor, Patient | | |
| **Pre-condition(s)** | | User should have a working system with Internet access | | |
| **Post- condition(s)** | | User will be successfully login to the account and redirected to his/her homepage | | |
| **Typical Course of Action** | | | | |
| **S#** | **Actor Action** | | | **System Response** |
| **1** | User clicks the sign up button. | | |  |
| **2** |  | | | System will prompt a form. |
| **3** | User will enter his/her information in respective fields. | | |  |
| **4** |  | | | System will verify the information. |
| **Alternate Course of Action 1 (Email already exist)** | | | | |
| **S#** | **Actor Action** | | | **System Response** |
| **3** |  | | | System displays “an account with same email address already exists” message. |
| Go to step 2. | | | | |
| **Alternate Course of Action 2 (invalid fields)** | | | | |
|  | **Actor Action** | | **System Response** | |
| **4** |  | | System displays “particular field is invalid” message. | |
| Go to step 2. | | | | |

**Table 1: UC-1**

## 2.2 Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | | UC-2 | | |
| **Name** | | Login | | |
| **Summary** | | The user will login with email id and password | | |
| **Priority** | | High | | |
| **Actors** | | Doctor, Patient | | |
| **Pre-condition(s)** | | The user should have login credentials | | |
| **Post- condition(s)** | | User will successfully login on to the home page | | |
| **Typical Course of Action** | | | | |
| **S#** | **Actor Action** | | | **System Response** |
| **1** | User clicks the Login button. | | |  |
| **2** |  | | | System will prompt to enter account information. |
| **3** | User enters an email id. | | |  |
| **4** | User enters a password. | | |  |
| **5** |  | | | System will authenticate, identify  User and redirect user to home page. |
| **Alternate Course of Action 1 (invalid email)** | | | | |
| **S#** | **Actor Action** | | | **System Response** |
| **3** |  | | | System displays invalid email message. |
| Go to step 2. | | | | |
| **Alternate Course of Action 2 (invalid password)** | | | | |
|  | **Actor Action** | | **System Response** | |
| **4** |  | | System shall prompt if the password is incorrect. | |
| Go to step 2. | | | | |

**Table 2: UC-2**

## 2.3 Edit profile

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-3 | |
| **Name** | | Edit Profile | |
| **Summary** | | The user can edit his/her profile | |
| **Priority** | | low | |
| **Actors** | | Doctor, Patient | |
| **Pre-condition(s)** | | User will be logged in | |
| **Post- condition(s)** | | System will display the updated information to the user | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User clicks the setting option. | |  |
| **2** |  | | System will show his/her profile. |
| **3** | User can edit his/her details. | |  |
| **4** |  | | System will authenticate and will set new details. |
| **Alternate Course of Action 1 (invalid details)** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **3** |  | | System displays “invalid details” message. |
| Go to step 2. | | | |

**Table 3: UC-3**

## 2.4 Cancel Appointment

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-4 | |
| **Name** | | Cancel Appointment | |
| **Summary** | | This use case describes how user can cancel appointment. | |
| **Priority** | | low | |
| **Actors** | | Doctor, Patient | |
| **Pre-conditions** | | Login, appointment is already book | |
| **Post- conditions** | | If success:  Display appointment is successfully canceled  If failure:  User is presented with an error message | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Select “cancel appointment” | |  |
| **2** |  | | Display message” appointment is successfully canceled” |

**Table 4: UC-4**

## 2.5 View Appointment History

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-5 | |
| **Name** | | View Appointment History | |
| **Summary** | | User can see his/her Appointment history | |
| **Priority** | | Medium | |
| **Actors** | | Doctor, Patient | |
| **Pre-condition(s)** | | The user is logged in | |
| **Post- condition(s)** | | List of appointments will be displayed. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User clicks the appointment history button. | |  |
| **2** |  | | System will display the list of appointments. |

**Table 5: UC-5**

## 2.6 Get Suggestion of highly rated doctors

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-6 | |
| **Name** | | Get Suggestion of highly rated doctors | |
| **Summary** | | User can see list of highly rated doctors. | |
| **Priority** | | Medium | |
| **Actors** | | Patient | |
| **Pre-condition(s)** | | The user is logged in | |
| **Post- condition(s)** | | List of suggestion of highly rated doctors will be displayed will be displayed. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User clicks the Suggestion of Highly Rated Doctors button. | |  |
| **2** |  | | System will display the list of highly rated doctors. |

**Table 6: UC-6**

## 2.7 Selecting COVID cell

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-7 | |
| **Name** | | Selecting COVID cell: | |
| **Summary** | | User can enter into COVID cell. | |
| **Priority** | | High | |
| **Actors** | | Patient | |
| **Pre-condition(s)** | | The user is logged in | |
| **Post- condition(s)** | | List of doctors who are available for COVID patients will be displayed. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User clicks the Emergency button. | |  |
| **2** |  | | System will display the COVID cell. |

**Table 7: UC-7**

## 2.8 Set medicine reminders

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-8 | |
| **Name** | | Set Medicine Reminder | |
| **Summary** | | User can set reminder for medicine. | |
| **Priority** | | Low | |
| **Actors** | | Patient | |
| **Pre-condition(s)** | | User is logged in | |
| **Post- condition(s)** | | System sends a confirmation message to the user | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User clicks the Set Medicine Reminder button. | |  |
| **2** |  | | System will display the list of booked appointments. |
| **3** | User selects the appointment from the list. | |  |
| **4** | User enters the reminder time. | |  |
| **5** |  | | System displays the confirmation  Message. |
| **Alternate Course of Action 1 (invalid time)** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **4** |  | | System displays “invalid time” message. |
| Go to step 4 | | | |
| **Alternate Course of Action 2 (No appointments)** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** |  | | System displays “No appointments booked” message. |
| Go to step 1. | | | |

**Table 8: UC-8**

## 2.9 Set Appointment Reminder

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-9 | |
| **Name** | | Set Appointment Reminder | |
| **Summary** | | User can set reminder for his/her booked appointment | |
| **Priority** | | Low | |
| **Actors** | | Patient | |
| **Pre-condition(s)** | | User is logged in | |
| **Post- condition(s)** | | System sends a confirmation message to the user | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User clicks the Set Appointment Reminder button. | |  |
| **2** |  | | System will display the list of booked appointments. |
| **3** | User selects the appointment from the list. | |  |
| **4** | User enters the reminder time. | |  |
| **5** |  | | System displays the confirmation message. |
| **Alternate Course of Action 1 (invalid time)** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **4** |  | | System displays “invalid time” message. |
| Go to step 4 | | | |
| **Alternate Course of Action 2 (No appointments)** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** |  | | System displays “No appointments booked” message. |
| Go to step 1. | | | |

**Table 9: UC-9**

## 2.10 Search doctor

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-10 | |
| **Name** | | Search Doctor | |
| **Summary** | | The User can search doctors by their specialty or name. | |
| **Priority** | | High | |
| **Actors** | | Patient | |
| **Pre-condition(s)** | | User is logged in. | |
| **Post- condition(s)** | | User will output the resulted list of doctors for patients search. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User clicks the search button. | |  |
| **2** |  | | System will prompt a search bar. |
| **3** | User will enter doctor’s specialty or name. | |  |
| **4** |  | | System will show the list of doctors in the result of the keywords user entered. |
| **Alternate Course of Action 1 (Doctor does not exists)** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **3** |  | | System displays “No doctor found.” message. |
| Go to step 2. | | | |

**Table 10: UC-10**

## 2.11 View Doctor’s Profile

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-11 | |
| **Name** | | View Doctor’s Profile | |
| **Summary** | | This use case describes how user can view doctor’s profile. | |
| **Priority** | | Medium | |
| **Actors** | | Patient | |
| **Pre-conditions** | | User should be logged in | |
| **Post- conditions** | | Doctor’s profile will be displayed. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
|  | Include UC-10 | |  |
| **1** | User will select any of the doctor to view his/her profile. | |  |
| **2** |  | | System will display the profile of the doctor. |

**Table 11: UC-11**

## 2.12 Book Appointment

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-12 | |
| **Name** | | Book Appointment | |
| **Summary** | | To allow a patient to book an appointment with a doctor | |
| **Priority** | | High | |
| **Actors** | | Patient | |
| **Pre-conditions** | | The patient is registered. | |
| **Post-conditions** | | The patient is in the doctor’s appointment list. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
|  | Include UC-11 | |  |
| **1** | User clicks Appointment button | |  |
| **2** |  | | System displays the doctor’s available times for appointment |
| **3** | Patient chooses the desired timings for the appointment | |  |
| **4** |  | | System adds the patient to the doctor’s appointment list |

**Table 12: UC-12**

## 2.13 Rate Doctors

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-13 | |
| **Name** | | Rate Doctors | |
| **Summary** | | To give feedback to the doctors | |
| **Priority** | | Low | |
| **Actors** | | Patient, Doctor | |
| **Pre-conditions** | | The user must have logged in and have taken an appointment from the respective doctor | |
| **Post-conditions** | | Ratings of the doctor successfully updated | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User selects “Rate” option | |  |
| **2** |  | | System displays the doctors whose appointment the patient has taken. |
| **3** | User rates the doctor from the scale of 1 to 5 and selects “Done” option | |  |
| **4** |  | | System records the rating and displays a message |
| **Alternate Course of Action (No Doctor Found)** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** |  | | System displays error message “No appointment from any doctor found” |

**Table 13: UC-13**

## 2.14 Add Doctor’s Availability

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-14 | |
| **Name** | | Add Doctor’s Availability | |
| **Summary** | | To allow the doctor to enter their availability. | |
| **Priority** | | Medium | |
| **Actors** | | Doctor | |
| **Pre-conditions** | | The doctor is registered and logged in. | |
| **Post-conditions** | | Doctor’s availability status is updated successfully. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Doctor selects the ‘Availability Status’ option | |  |
| **2** |  | | System displays a drop down list having options to enter time and date |
| **3** | Doctor enters the desired date and time. | |  |
| **4** |  | | System updates the doctor’s availability status |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **3** |  | | System displays invalid date/time. |
| Go to step 3. | | | |

**Table 14: UC-14**

## 2.15 Cancel Doctor’s Availability

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-15 | |
| **Name** | | Cancel Doctor’s Availability | |
| **Summary** | | To allow the doctor to remove their availability. | |
| **Priority** | | Medium | |
| **Actors** | | Doctor | |
| **Pre-conditions** | | The doctor is registered and logged in. | |
| **Post-conditions** | | Doctor’s availability status is updated successfully. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Doctor selects the ‘Availability Status’ option. | |  |
| **2** |  | | System displays a status of availability of doctor. |
| **3** | Doctor selects ‘Cancel Availability’ button. | |  |
| **4** |  | | System updates the doctor’s availability status. |
| **Alternate Course of Action (No Availability Found)** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** |  | | System displays ‘No Availability Found’. |

**Table 15: UC-15**

## 2.16 COVID Cell Registration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | | UC-16 | | |
| **Name** | | COVID Cell Registration | | |
| **Summary** | | The user can register himself/herself for COVID cell. | | |
| **Priority** | | High | | |
| **Actors** | | Doctor | | |
| **Pre-condition(s)** | | User should have a working system with Internet access and should be logged in. | | |
| **Post- condition(s)** | | User will be successfully login to the account and redirected to his/her homepage | | |
| **Typical Course of Action** | | | | |
| **S#** | **Actor Action** | | | **System Response** |
| **1** | User clicks the Register For COVID Cell button. | | |  |
| **2** |  | | | System will prompt a form. |
| **3** | User will enter his/her information in respective fields. | | |  |
| **4** |  | | | System will verify the information. |
| **Alternate Course of Action 1 (Already Registered)** | | | | |
| **S#** | **Actor Action** | | | **System Response** |
| **1** |  | | | System displays “You have Already registered to COVID cell.” message. |
| **Alternate Course of Action 2 (invalid fields)** | | | | |
|  | **Actor Action** | | **System Response** | |
| **4** |  | | System displays “particular field is invalid” message. | |
| Go to step 2. | | | | |

**Table 16: UC-16**

## 2.17 Book Appointment in COVID Cell

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-17 | |
| **Name** | | Book Appointment in COVID Cell | |
| **Summary** | | To allow a patient to book an appointment with a doctor in COVID cell. | |
| **Priority** | | High | |
| **Actors** | | Patient | |
| **Pre-conditions** | | The patient is logged in. | |
| **Post-conditions** | | The patient will get appointment from a doctor in COVID cell. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
|  | Include UC-7 | |  |
| **1** | User selects the doctor from he want to get appointment. | |  |
| **2** |  | | System displays the doctor’s available times for appointment |
| **3** | Patient chooses the desired timings for the appointment | |  |
| **4** |  | | System adds the patient to the doctor’s appointment list |

**Table 17: UC-17**

# 3. Nonfunctional Requirements

## 3.1 Performance Requirements

* The Medico application’s load time should not be more than one second for users so that the users do not have to wait for a long time and do not suffer from delayed responses.
* The application needs to support at least 20 people at once so that the users could simultaneously access their profiles.
* The application must acknowledge the user's command within five seconds in order to increase the efficiency of the system.

## 3.2 Security Requirements

As security is the main concern for the users and helps in gaining user’s trust and confidence. Therefore,

* The application must use a secure database.
* The application must hold a user id and password for a user to log in.
* The application must ensure that the user’s data is protected.

## 3.3 Usability Requirements

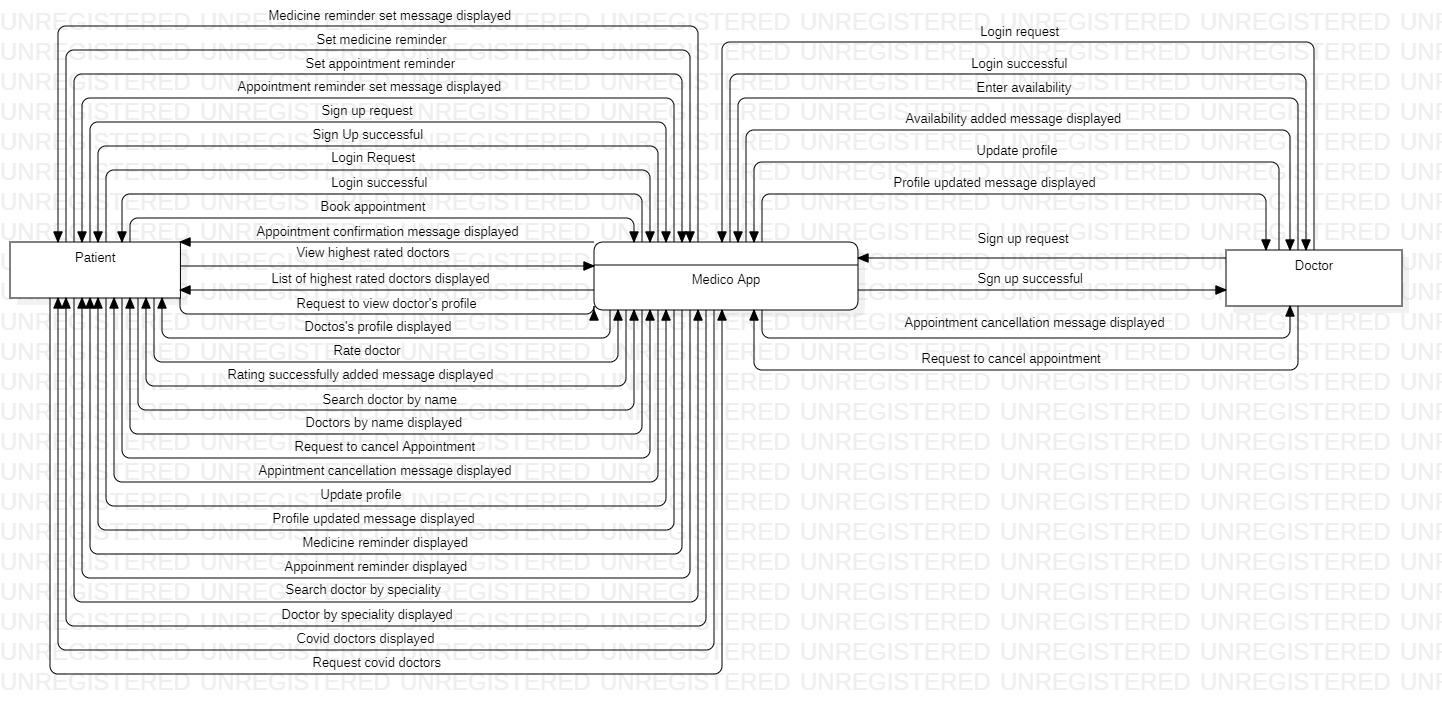
* The application must ensure that users can access their profiles 98% of the time without failure to ensure a smooth working of system.
* The application must ensure that the user interface is user-friendly and easy to use. Fit Criteria: “90% of the users should be able to create a new account within 10 minutes without using any help.
* The application must provide interface for users which can be used by any age group.

## 3.4 Reliability Requirements

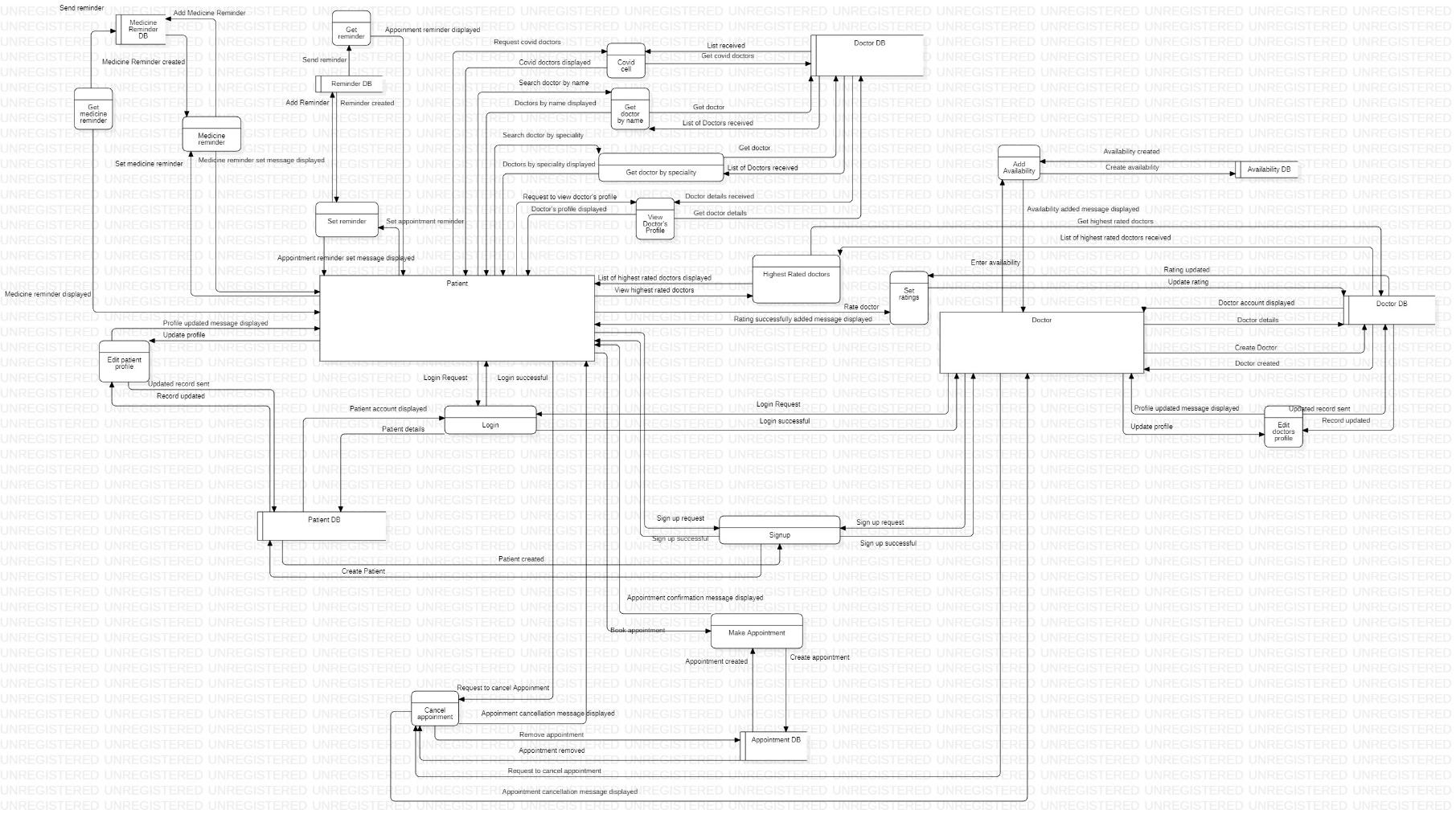
* The application must ensure that it is available 24 hours because our system is online based and requires data retrieval and update.
* The application must make sure that more than three failed attempts at log in will display a message so that the user could change its password and access the account easily.

# Appendix A: Data Flow Diagram

## Data Flow diagram

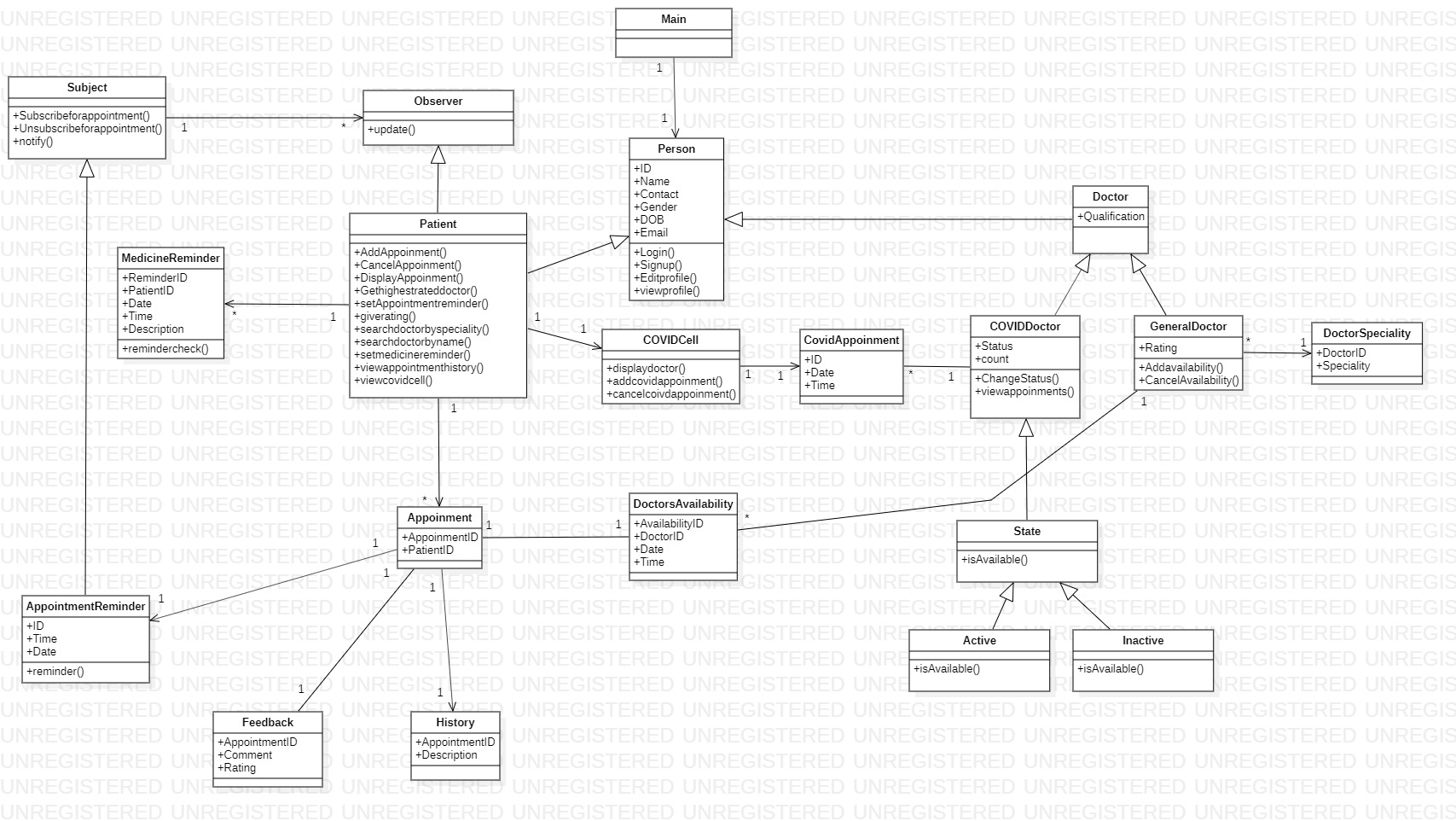


**1.1 Level 0 Diagram**

**1.2 Level 1 Diagram**

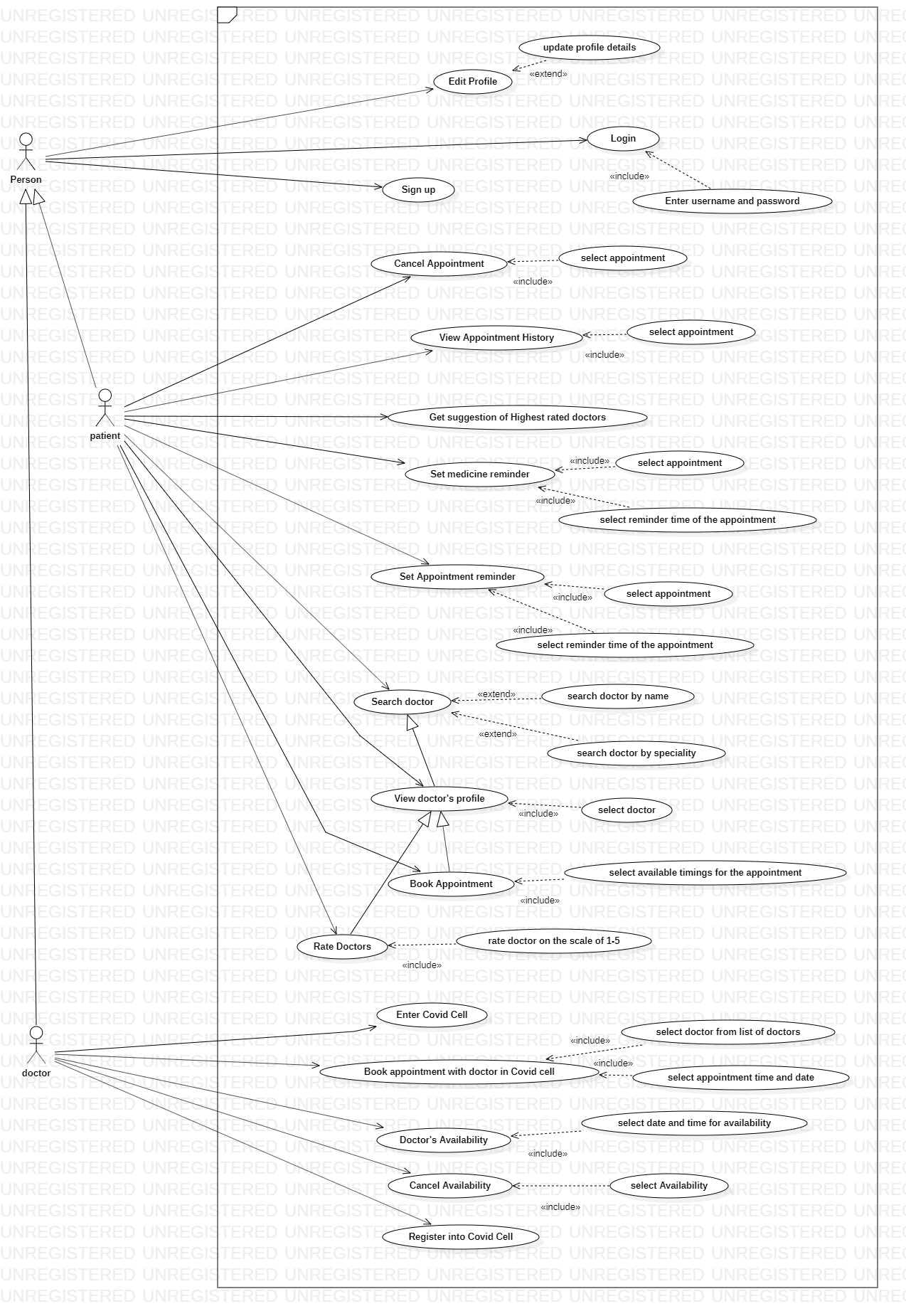
# Appendix B: Analysis Model

## Class Diagram



**1.1 Class Diagram**

## Use Case Diagram

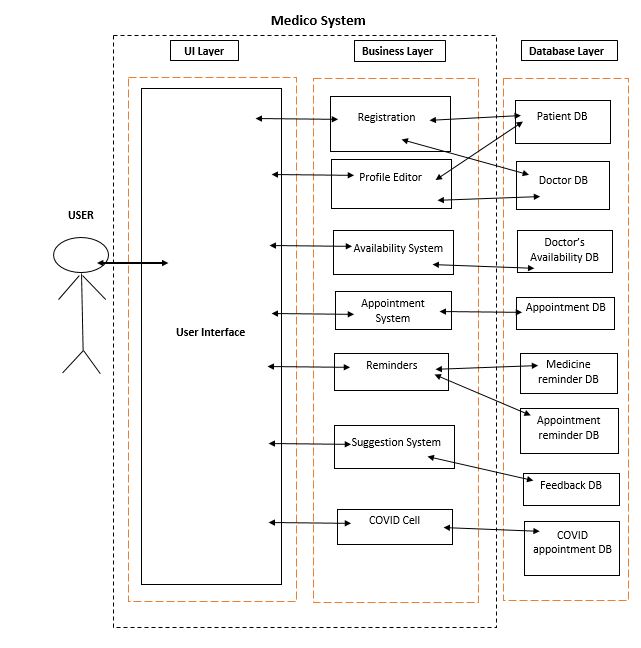


**2.1 Use Case Diagram**

# Appendix C: Design Models

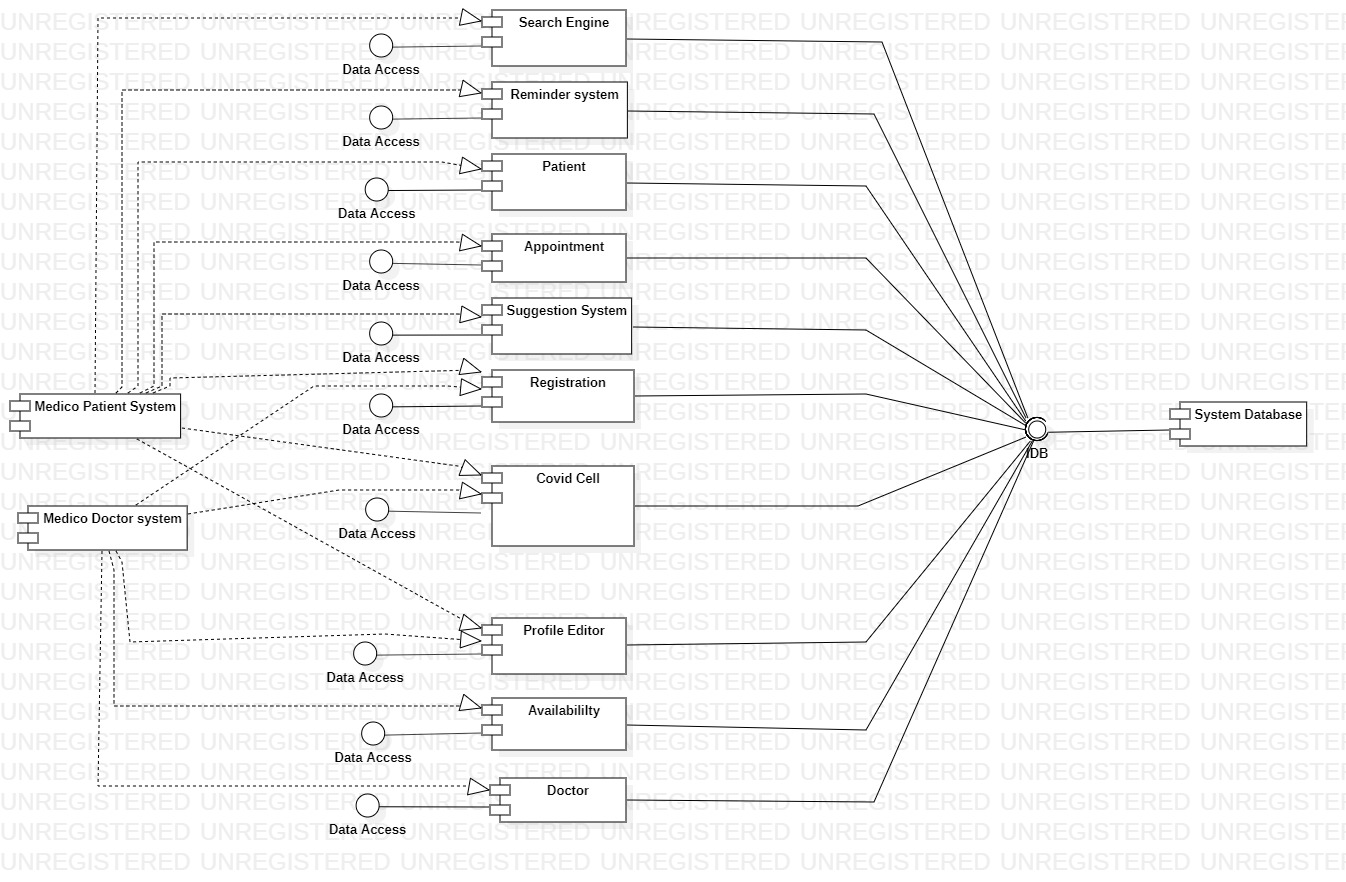
## Architectural Diagram

**Architectural Style:** Layer Architecture/ Call and Return



**1.1 Architectural Diagram**

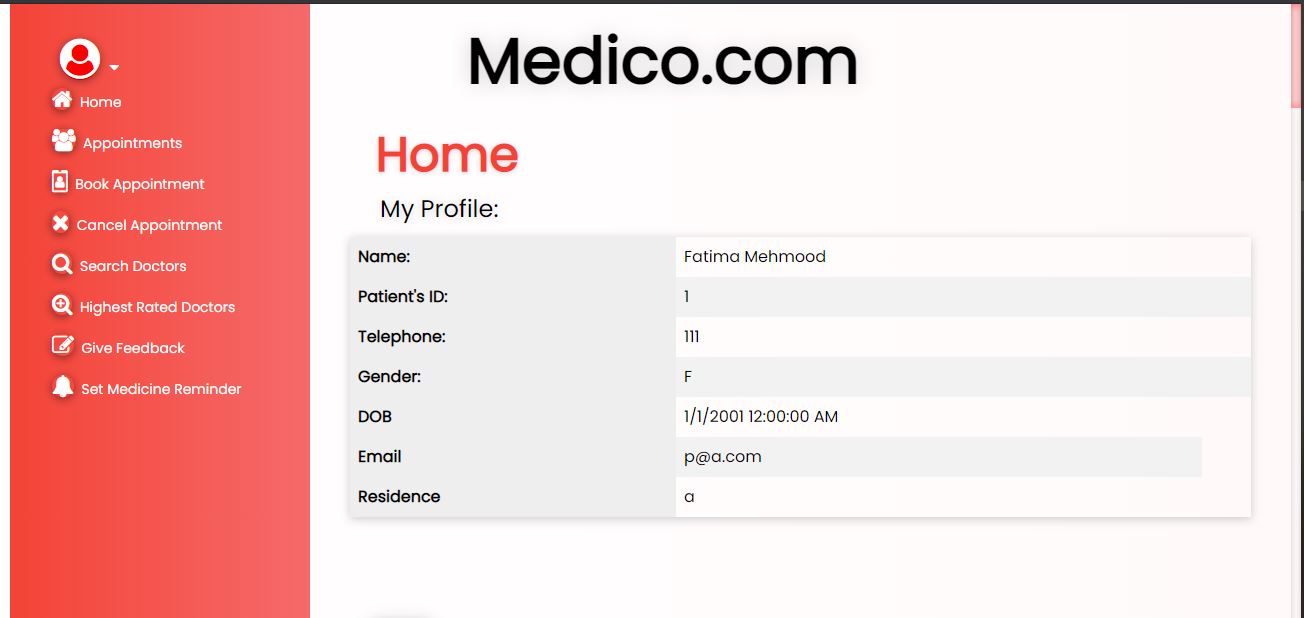
## Component-Level Diagram



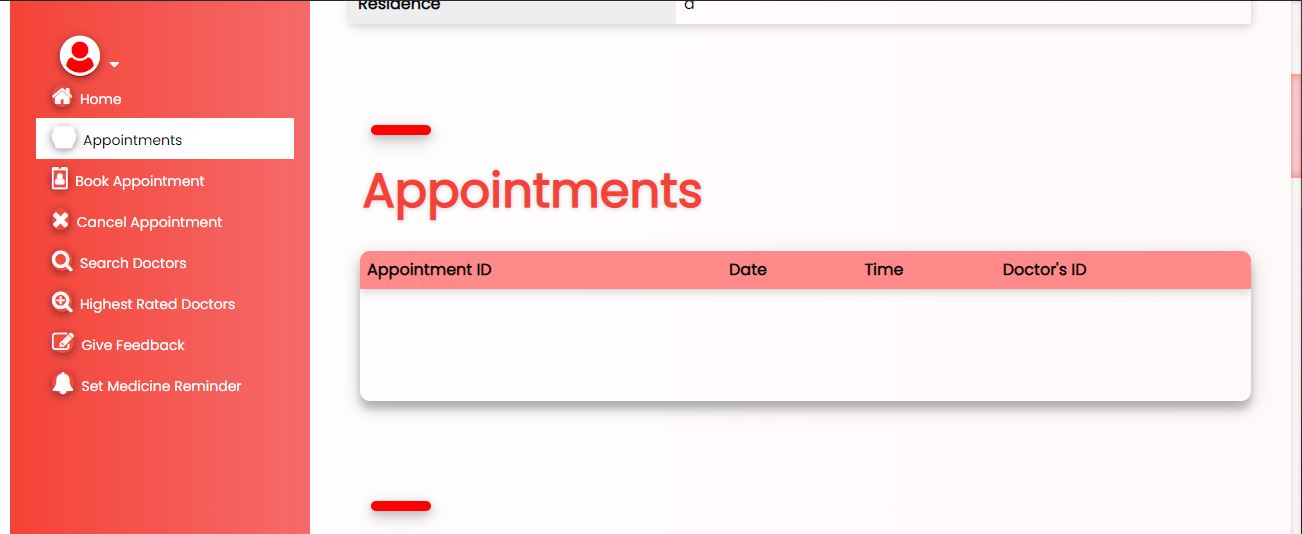
**2.1 Component-Level Diagram**

# Appendix D: Screenshots

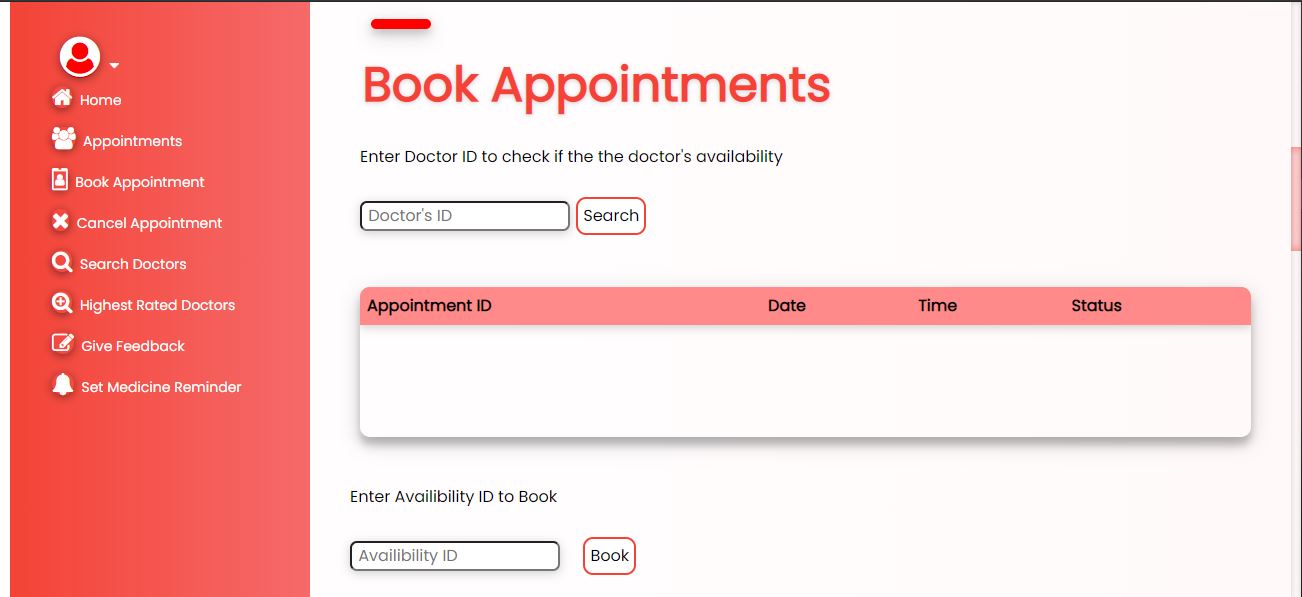
**Patient’s Home Page**



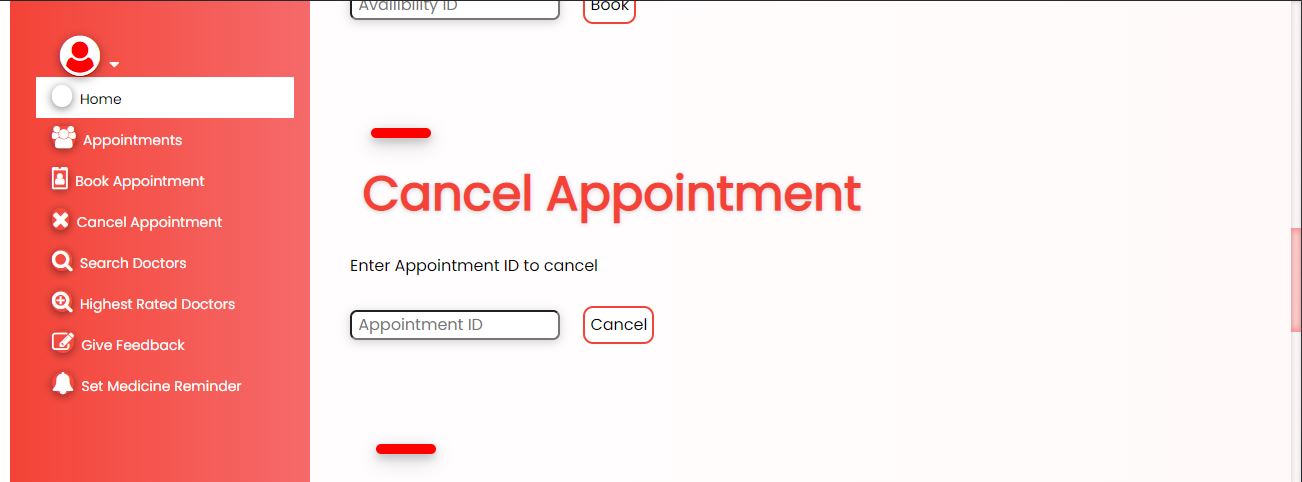
**View Booked Appointments**



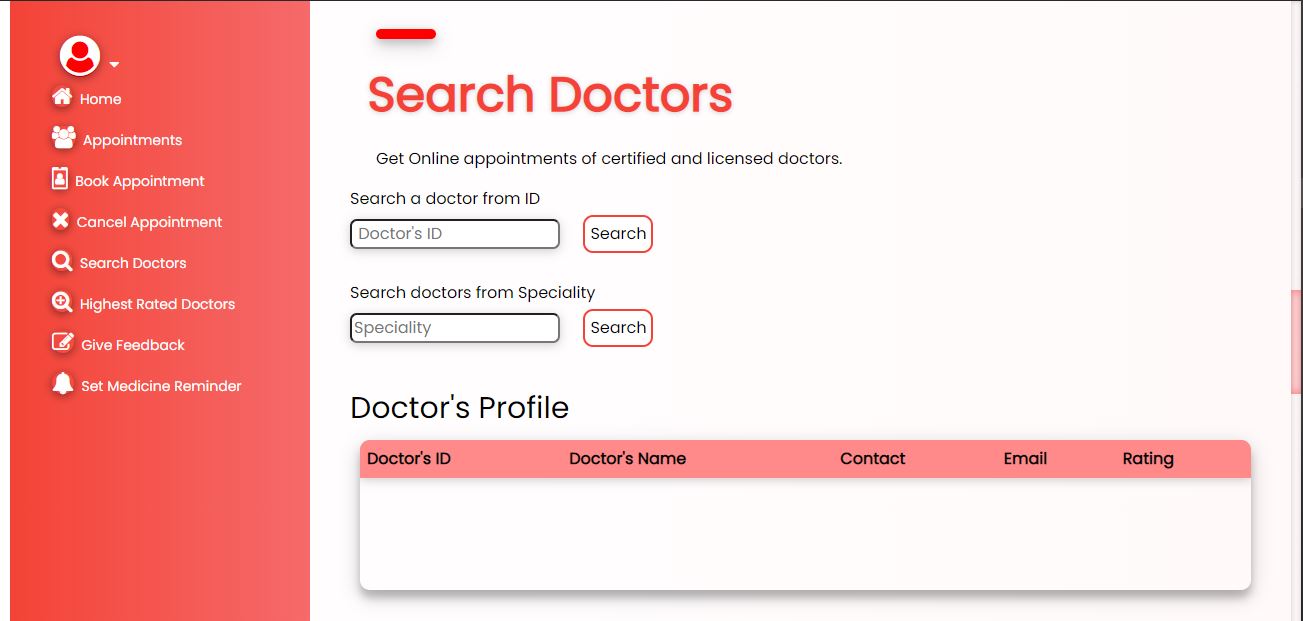
**Book Appointments**



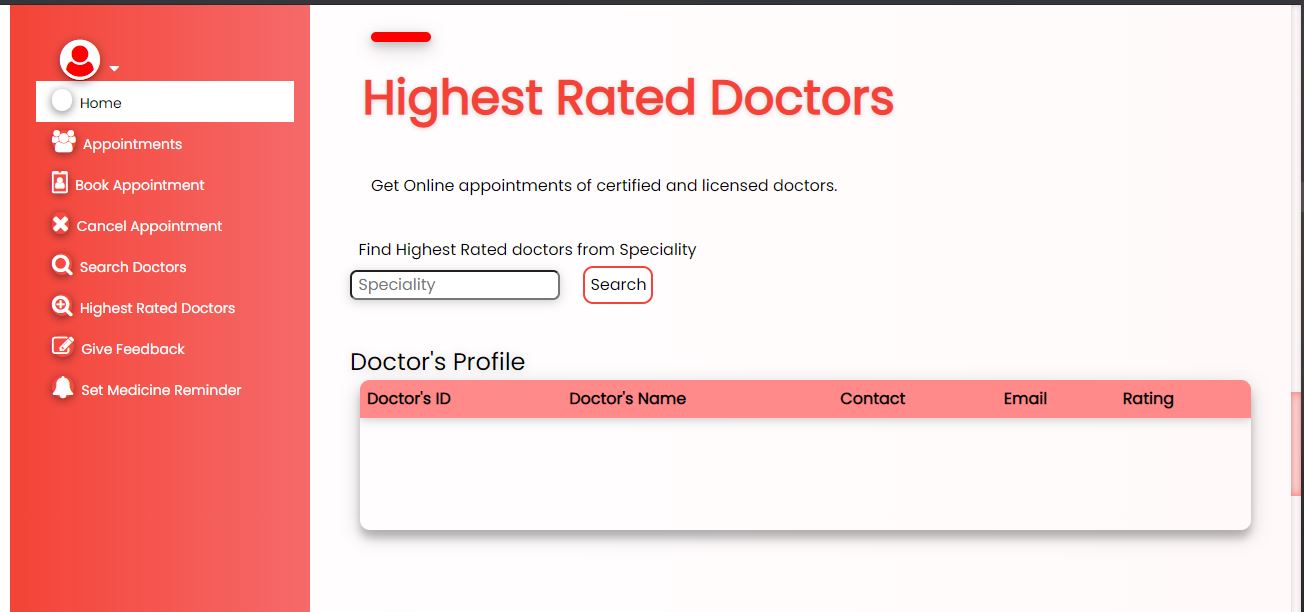
**Cancel Appointments**



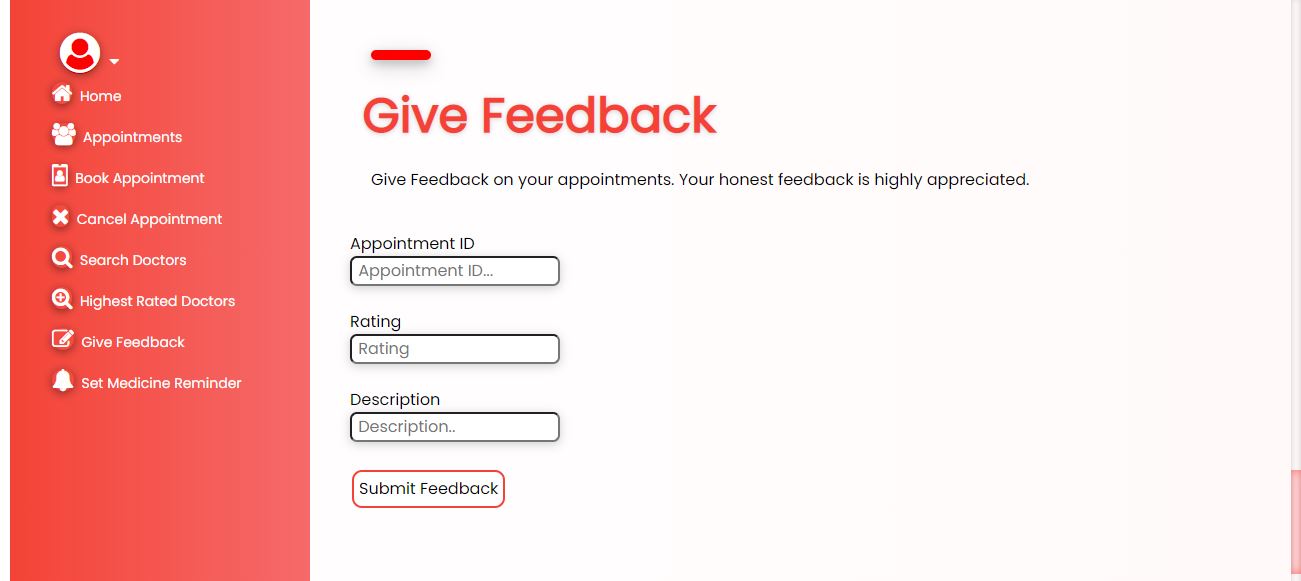
**Search Doctor by Specialty or ID**



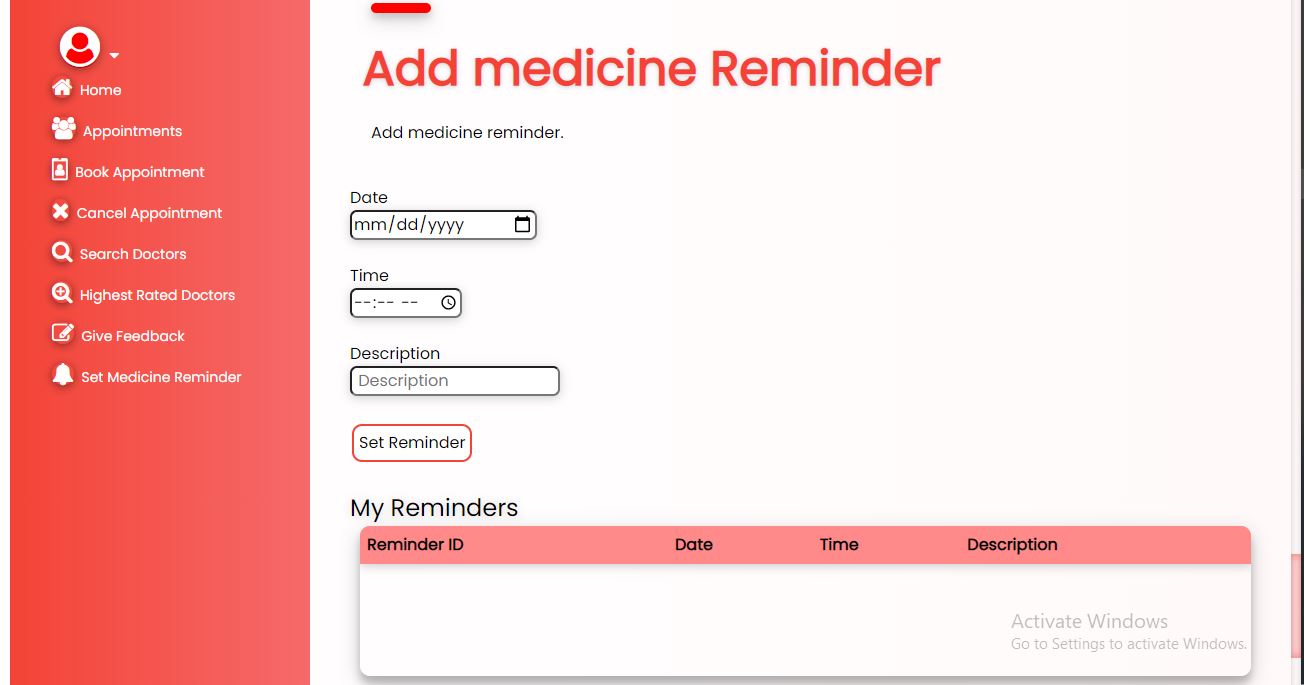
**Get Suggestion of Highest Rated Doctors**



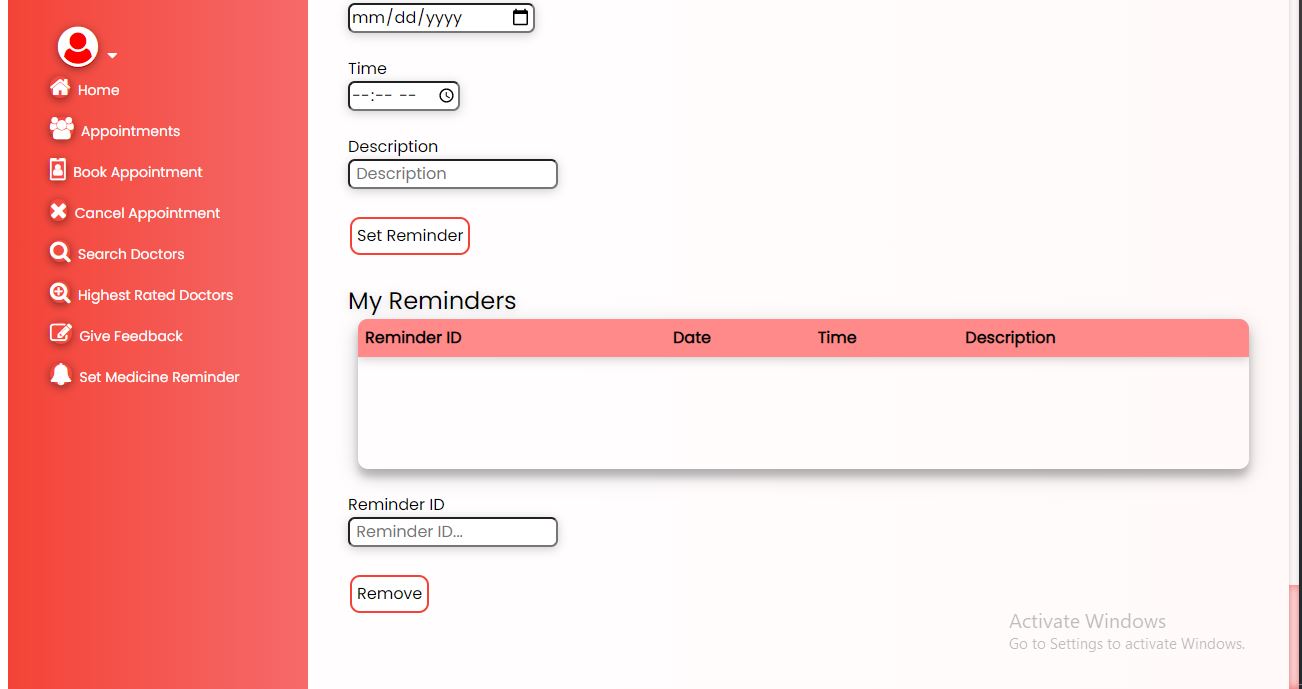
**Give Feedback for an Appointment**



**Set a Medicine Reminder**



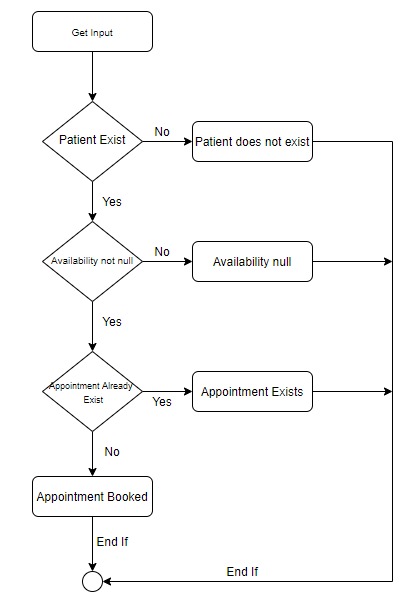
**Delete a Medicine Reminder**



# Appendix E: Test Cases

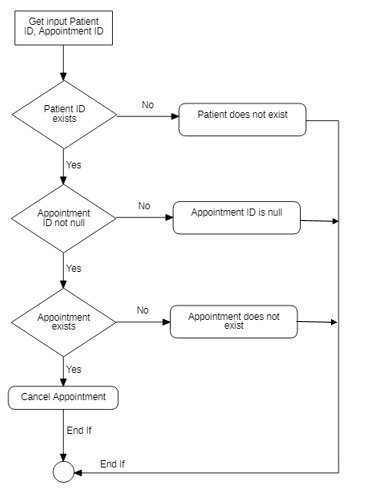
## Flow-chart Diagram

**Book Appointment**



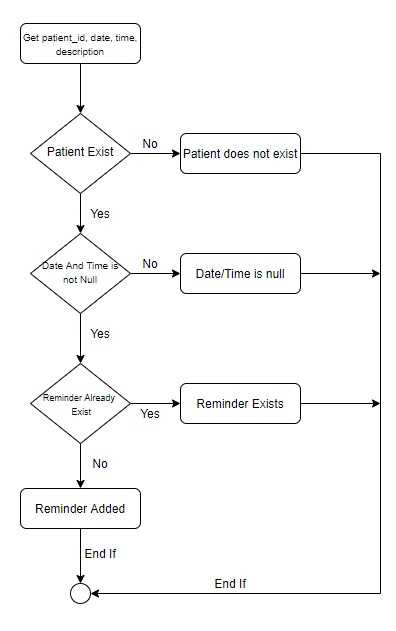
**1.1 Book Appointment**

**Cancel Appointment**



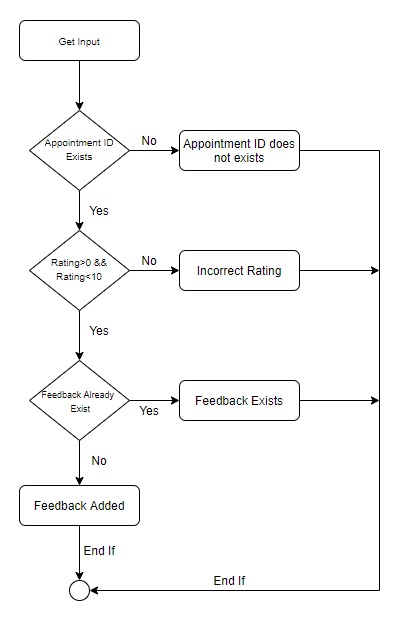
**1.2 Cancel Appointment**

**Add Reminder**



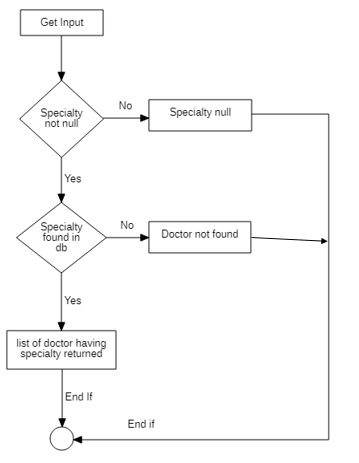
**1.3 Add Reminder**

**Give Feedback**



**1.4 Give Feedback**

**Doctor Specialty**



**1.5 Doctor Specialty**

|  |  |
| --- | --- |
| **Identifier** | TC-1 |
| **Priority** | High |
| **Related Requirement(s)** | UC-11,SRS-3.3 |
| **Short description** | Patient searches for the doctor using his id. |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Doctor id = 1 |
| **Detailed step(s)** | Doctor id is searched in the database .Doctor with the input id = 1 is found then doctor’s profile is displayed. |
| **Expected Result(s)** | Doctor profile is displayed |
| **Post Condition(s)** | Doctor’s profile is displayed on screen |

**Table 18: TC-1**

|  |  |
| --- | --- |
| **Identifier** | TC-2 |
| **Priority** | High |
| **Related Requirement(s)** | SRS-3.1,SRS-3.2 |
| **Short description** | Patient searches for the doctor using his id. |
| **Pre- condition(s)** | User has logged in |
| **Input data** | Doctor Id = -1 |
| **Detailed step(s)** | Doctor id is searched in the database. Doctor with id =-1 is not in database. Message displayed that doctor is not found |
| **Expected Result(s)** | Doctor’s profile or Doctor not found message is displayed |
| **Post Condition(s)** | Message is displayed on screen |

**Table 19: TC-2**

|  |  |
| --- | --- |
| **Identifier** | TC-3 |
| **Priority** | medium |
| **Related Requirement(s)** | SRS-3.1,SRS-3.2 |
| **Short description** | Doctors having a specific specialty are searched by patients |
| **Pre- condition(s)** | User has logged in |
| **Input data** | Specialty = ‘abds’ |
| **Detailed step(s)** | Doctor’s specialty is searched in the database. List of doctors having specialty ‘abds’ is displayed. If no such doctor is found the message, ‘No doctor found’ is displayed |
| **Expected Result(s)** | Message that ‘No doctor found’ is displayed |
| **Post Condition(s)** | Message will be displayed |

**Table 20: TC-3**

|  |  |
| --- | --- |
| **Identifier** | TC-4 |
| **Priority** | medium |
| **Related Requirement(s)** | SRS-3.1,SRS-3.2 |
| **Short description** | Doctors having a specific specialty are displayed |
| **Pre- condition(s)** | User has logged in |
| **Input data** | Specialty = ‘mbbs’ |
| **Detailed step(s)** | Doctors’ specialty is searched in the database. List of doctors having specialty ‘mbbs’ is displayed. If no such doctor is found the message, ‘No doctor found’ is displayed |
| **Expected Result(s)** | List of Doctors having specialty, ‘mbbs’ are displayed |
| **Post Condition(s)** | List of doctors displayed on screen |

**Table 21: TC-4**

|  |  |
| --- | --- |
| **Identifier** | TC-5 |
| **Priority** | medium |
| **Related Requirement(s)** | SRS-3.1, SRS-3.2 |
| **Short description** | User checks all the specialties of the doctor with a specific id. |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Doctor id =1 |
| **Detailed step(s)** | User inputs doctor’s id and list of specialty of doctor with id =1 is retrieved from database and displayed. |
| **Expected Result(s)** | List of specialties of doctor with id =1 s displayed |
| **Post Condition(s)** | List of specialties are displayed on screen. |

**Table 22: TC-5**

|  |  |
| --- | --- |
| **Identifier** | TC-6 |
| **Priority** | medium |
| **Related Requirement(s)** | SRS-3.1,SRS-3.2 |
| **Short description** | User checks all the specialties of the doctor with a specific id. |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Doctor id =-1 |
| **Detailed step(s)** | User inputs doctor’s  id =-1 is searched in database |
| **Expected Result(s)** | ‘Doctor with id =1 is not found’ |
| **Post Condition(s)** | Message is displayed on the screen. |

**Table 23: TC-6**

|  |  |
| --- | --- |
| **Identifier** | TC-7 |
| **Priority** | Medium |
| **Related Requirement(s)** | UC-8 |
| **Short description** | A reminder for patients appointment is set |
| **Pre- condition(s)** | Patient has logged in the system |
| **Input data** | Date=’03-07-2021’, time = ‘12:30’, description=’Appointment with Dr. B’ |
| **Detailed step(s)** | User inputs data. If the date and time are valid and the reminder for the same appointment has not been already made, a new reminder is set. |
| **Expected Result(s)** | Message displayed, ‘Reminder added successfully’ |
| **Post Condition(s)** | None |

**Table 24: TC-7**

|  |  |
| --- | --- |
| **Identifier** | TC-8 |
| **Priority** | Medium |
| **Related Requirement(s)** | UC-8 |
| **Short description** | A reminder for patients appointment is set |
| **Pre- condition(s)** | Patient has logged in the system |
| **Input data** | Date=’03-07-2021’, time = ‘12:30’, description=’Appointment with Dr. B’ |
| **Detailed step(s)** | User inputs data. If the date and time are valid and the reminder for the same appointment has not been already made, a new reminder is set. |
| **Expected Result(s)** | Message displayed, ‘Reminder already added’ |
| **Post Condition(s)** | Message displayed on screen |

**Table 25: TC-8**

|  |  |
| --- | --- |
| **Identifier** | TC-9 |
| **Priority** | Medium |
| **Related Requirement(s)** | UC-8 |
| **Short description** | A reminder for patients appointment is set |
| **Pre- condition(s)** | Patient has logged in the system |
| **Input data** | Date=’’, time = ‘’, description=’Appointment with Dr. B’ |
| **Detailed step(s)** | User inputs data. If the date and time are valid and the reminder for the same appointment has not been already made, a new reminder is set. |
| **Expected Result(s)** | Message displayed, ‘Time not entered’, ‘Date not entered’ |
| **Post Condition(s)** | Message displayed on the screen |

**Table 26: TC-9**

|  |  |
| --- | --- |
| **Identifier** | TC-10 |
| **Priority** | Medium |
| **Related Requirement(s)** | SRS-3.4 |
| **Short description** | A reminder for patients appointment is deleted |
| **Pre- condition(s)** | Patient has logged in the system |
| **Input data** | Reminder id = 1 |
| **Detailed step(s)** | User inputs data. Reminder id is searched in the database. If Reminder id is found in the database, it is deleted. ‘No such reminder exists’ |
| **Expected Result(s)** | Remind id = 1 is deleted. Message displayed, ‘Reminder deleted successfully’’ |
| **Post Condition(s)** | Reminder  for patient is deleted from database |

**Table 27: TC-10**

|  |  |
| --- | --- |
| **Identifier** | TC-11 |
| **Priority** | Medium |
| **Related Requirement(s)** | SRS-3.4 |
| **Short description** | A reminder for patients appointment is deleted |
| **Pre- condition(s)** | Patient has logged in the system |
| **Input data** | Reminder id = 1 |
| **Detailed step(s)** | User inputs data. Reminder id is searched in the database. If Reminder id is found in the database, it is deleted. ‘No such reminder exists’ |
| **Expected Result(s)** | Message displayed, ‘No such reminder exists’’ |
| **Post Condition(s)** | Message displayed on screen |

**Table 28: TC-11**

|  |  |
| --- | --- |
| **Identifier** | TC-12 |
| **Priority** | Medium |
| **Related Requirement(s)** | SRS-3.4 |
| **Short description** | A reminder for patients appointment is deleted |
| **Pre- condition(s)** | Patient has logged in the system |
| **Input data** | Reminder id = [] |
| **Detailed step(s)** | User inputs data. Reminder id is searched in the database. If Reminder id is found in the database, it is deleted. ‘No such reminder exists’ |
| **Expected Result(s)** | Message displayed, ‘Reminder id not added’ |
| **Post Condition(s)** | Reminder  for patient is deleted from database |

**Table 29: TC-12**

|  |  |
| --- | --- |
| **Identifier** | TC-13 |
| **Priority** | Medium |
| **Related Requirement(s)** | SRS-3.4 |
| **Short description** | A reminder for patients appointment is displayed |
| **Pre- condition(s)** | Patient has logged in the system |
| **Input data** |  |
| **Detailed step(s)** | Description of all the appointment reminders made by patient including Reminder ID, Date, Time and Description are displayed on the screen |
| **Expected Result(s)** | Reminder description (Reminder ID, Date, Time and Description)displayed on screen |
| **Post Condition(s)** | Reminder description displayed on screen |

**Table 30: TC-13**

|  |  |
| --- | --- |
| **Identifier** | TC-14 |
| **Priority** | medium |
| **Related Requirement(s)** | UC-6 |
| **Short description** | User searches high rated doctors with specific specialty |
| **Pre- condition(s)** | User is logged in |
| **Input data** | specialty=’mbbs’ |
| **Detailed step(s)** | Doctors with specialty = ‘mbbs’ is searched in the database and a list of high rated doctors are displayed. |
| **Expected Result(s)** | Details of high rated doctors with specialty is displayed |
| **Post Condition(s)** | List of details of high rated doctors with specialty = ‘mbbs’ is displayed. |

**Table 31: TC-14**

|  |  |
| --- | --- |
| **Identifier** | TC-15 |
| **Priority** | medium |
| **Related Requirement(s)** | UC-6 |
| **Short description** | User searches high rated doctors with specific specialty |
| **Pre- condition(s)** | User is logged in |
| **Input data** | specialty=’abcds’ |
| **Detailed step(s)** | Doctors with specialty = ‘abcds’ is searched in the database and no doctor with specialty abcds exists  message is displayed |
| **Expected Result(s)** | ‘No doctor with specialty abcds exists ‘ |
| **Post Condition(s)** | Message is displayed on screen |

**Table 32: TC-15**

|  |  |
| --- | --- |
| **Identifier** | TC-16 |
| **Priority** | High |
| **Related Requirement(s)** | UC-12 |
| **Short description** | User books an appointment according to the availability of a doctor. |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Patient id =1 ,availability id = 1 |
| **Detailed step(s)** | Patient books an appointment with a doctor who is available with availability id = 1 |
| **Expected Result(s)** | ‘Appointment has been booked’ |
| **Post Condition(s)** | Message is displayed on screen and appointment for that patient is added in database for patient and doctor. |

**Table 33: TC-16**

|  |  |
| --- | --- |
| **Identifier** | TC-17 |
| **Priority** | High |
| **Related Requirement(s)** | UC-12 |
| **Short description** | User books an appointment according to the availability of a doctor. |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Patient id =1 ,availability id = -1 |
| **Detailed step(s)** | Patient books an appointment with a doctor who is available with availability id = -1 |
| **Expected Result(s)** | ‘No such availability is found’ |
| **Post Condition(s)** | Message is displayed on screen |

**Table 34: TC-17**

|  |  |
| --- | --- |
| **Identifier** | TC-18 |
| **Priority** | High |
| **Related Requirement(s)** | UC-12 |
| **Short description** | User books an appointment according to the availability of a doctor. |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Patient id =1 ,availability id = 2 |
| **Detailed step(s)** | Patient books an appointment with a doctor who is available with availability id = 1 |
| **Expected Result(s)** | ‘Time of availability has passed’ |
| **Post Condition(s)** | Message is displayed on the screen. |

**Table 35: TC-18**

|  |  |
| --- | --- |
| **Identifier** | TC-19 |
| **Priority** | High |
| **Related Requirement(s)** | UC-4 |
| **Short description** | User cancels an appointment made earlier with a doctor |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Appointment id  =1 ,availability id = 1 |
| **Detailed step(s)** | Patient cancels an appointment with a doctor who is  with availability id = 1 |
| **Expected Result(s)** | ‘Appointment has been canceled’’ |
| **Post Condition(s)** | Message is displayed on screen and appointment for that patient is deleted from database tables for patient and doctor. |

**Table 36: TC-19**

|  |  |
| --- | --- |
| **Identifier** | TC-20 |
| **Priority** | High |
| **Related Requirement(s)** | UC-4 |
| **Short description** | User cancels an appointment made earlier with a doctor |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Appointment id  =-1 ,availability id = 1 |
| **Detailed step(s)** | Patient cancels an appointment with a doctor who is  with availability id = 1 |
| **Expected Result(s)** | ‘No Appointment with id = -1 exists’ |
| **Post Condition(s)** | Message is displayed on the screen. |

**Table 37: TC-20**

|  |  |
| --- | --- |
| **Identifier** | TC-21 |
| **Priority** | High |
| **Related Requirement(s)** | UC-4 |
| **Short description** | User cancels an appointment according to the availability of a doctor. |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Patient id =1 ,availability id = 2 |
| **Detailed step(s)** | Patient cancels an appointment with a doctor who is available with availability id = 2 |
| **Expected Result(s)** | ‘Date of availability has passed’ |
| **Post Condition(s)** | Message is displayed on the screen. |

**Table 38: TC-21**

|  |  |
| --- | --- |
| **Identifier** | TC-22 |
| **Priority** | Low |
| **Related Requirement(s)** | UC-4 |
| **Short description** | User cancels an appointment according to the availability of a doctor. |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Patient id =1 |
| **Detailed step(s)** | List of appointments made by patient is displayed |
| **Expected Result(s)** | List and details of appointment made by patient is displayed |
| **Post Condition(s)** | Details are displayed on screen |

**Table 39: TC-22**

|  |  |
| --- | --- |
| **Identifier** | TC-23 |
| **Priority** | Low |
| **Related Requirement(s)** | UC-4 |
| **Short description** | User cancels an appointment according to the availability of a doctor. |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Patient id =3 |
| **Detailed step(s)** | List of appointments made by patient is displayed |
| **Expected Result(s)** | Patient has no appointments |
| **Post Condition(s)** | Message is displayed on the screen. |

**Table 40: TC-23**

|  |  |
| --- | --- |
| **Identifier** | TC-24 |
| **Priority** | Low |
| **Related Requirement(s)** | UC-04 |
| **Short description** | User cancels an appointment according to the availability of a doctor. |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Doctor id =1 |
| **Detailed step(s)** | List of appointments made by patient is displayed |
| **Expected Result(s)** | List of appointments of doctors are displayed |
| **Post Condition(s)** | Message is displayed on the screen. |

**Table 41: TC-24**

|  |  |
| --- | --- |
| **Identifier** | TC-25 |
| **Priority** | Low |
| **Related Requirement(s)** | SRS-3.3 |
| **Short description** | Patient gives feedback about appointment |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Appointment id= 1, Rating = 4, Comment = ‘The doctor was very good’ |
| **Detailed step(s)** | The patient gives the input data. If the input is valid, the rating is stored in the database |
| **Expected Result(s)** | Rating 4 and comment is stored in database for appointment 1 |
| **Post Condition(s)** | Rating and comment for appointment is stored in database |

**Table 42: TC-25**

|  |  |
| --- | --- |
| **Identifier** | TC-26 |
| **Priority** | Low |
| **Related Requirement(s)** | SRS-3.3 |
| **Short description** | Patient gives feedback about appointment |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Appointment id= 1, Rating = 2, Comment = ‘The doctor was very good’ |
| **Detailed step(s)** | The patient gives the input data. If the input is valid, the rating is stored in the database |
| **Expected Result(s)** | Message displayed, ‘Feedback already given’ |
| **Post Condition(s)** | Message displayed on screen |

**Table 43: TC-26**

|  |  |
| --- | --- |
| **Identifier** | TC-27 |
| **Priority** | Low |
| **Related Requirement(s)** | SRS-3.3 |
| **Short description** | Patient gives feedback about appointment |
| **Pre- condition(s)** | User is logged in |
| **Input data** | Appointment id= y6y, Rating = abc, Comment = ‘ ’ |
| **Detailed step(s)** | The patient gives the input data. If the input is valid, the rating is stored in the database |
| **Expected Result(s)** | Message displayed ‘Invalid appointment ID entered’, ‘Invalid rating entered’ |
| **Post Condition(s)** | Error message displayed on screen |

**Table 44: TC-27**