Version 2.0

Prepared by Vy Vu

Flinders University

24 December 2021

Software requirements specifications

**MedCare – Appointment Booking System & Patient portal**

[Revisions 2](#_Toc91494596)

[1. Introduction 2](#_Toc91494597)

[1.1. Purpose 2](#_Toc91494598)

[1.2. Intended Audience 2](#_Toc91494599)

[1.3. Project Scope 2](#_Toc91494600)

[1.4. Definitions & Acronyms 2](#_Toc91494601)

[1.5. References 3](#_Toc91494602)

[2. Overall description 3](#_Toc91494603)

[2.1. Product Perspective 3](#_Toc91494604)

[2.2. Product Features 3](#_Toc91494605)

[2.3. User Classes and Characteristics 3](#_Toc91494606)

[2.4. Operating Environment 4](#_Toc91494607)

[2.5. Design and Implementation Constraints 4](#_Toc91494608)

[2.6. User Documentation 4](#_Toc91494609)

[2.7. Assumptions and Dependencies 4](#_Toc91494610)

[3. Specific requirement system 4](#_Toc91494611)

[3.1. Users (Common behaviors of classes of users) 4](#_Toc91494612)

[3.2. Patient 5](#_Toc91494613)

[3.3. Medical staff 6](#_Toc91494614)

[3.4. Doctor 7](#_Toc91494615)

[3.5. Administrator 7](#_Toc91494616)

[4. External interface required 8](#_Toc91494617)

[4.1. User Interfaces 8](#_Toc91494618)

[4.2. Hardware Interfaces 8](#_Toc91494619)

[4.3. Software Interfaces 8](#_Toc91494620)

[4.4. Communication Interfaces 8](#_Toc91494621)

[5. Other Nonfunctional Requirements 9](#_Toc91494622)

[5.1. Performance Requirements 9](#_Toc91494623)

[5.2. Safety Requirements 9](#_Toc91494624)

[5.3. Security Requirements 9](#_Toc91494625)

[5.4. Software Quality Attributes 9](#_Toc91494626)

[Appendix A. Analysis model 9](#_Toc91494627)

# Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary author** | **Description of Version** | **Date Completed** |
| 1.0 | Vy Vu | Initial draft of SRS as description of final MedCare patient portal |  |
| 2.0 | Vy Vu | Remove EHR function | February 2, 2022 |

# Introduction

## Purpose

This document proposes the software functionalities and requirements for version 1.0 of MedCare, a patient portal for accessing electronic medical records and managing medical appointments. It will explain detailed description of the system, including features, system constraints and its interfaces.

## Intended Audience

Intended reader group for this document are customers, project managers and developers. The customer will be able to have a clear idea of how the system behave with detailed description of the functionalities. The system development team should be able to understand the functional and non-functional requirements and the design constraints of the system which is going to be developed.

## Project Scope

This project will be a web-based application for any clinical settings wishing to implement online medical records and online appointment scheduling services. The project is separated into two main uses:

* The health portal for users to gain access to their health records in electronic formats and make medical appointments with specific doctors at given date and time from anywhere and anytime, instead of using traditional phone calls during working hours.
* The appointment management system which allows medical staffs to manage system users and have full control over medical bookings, including create, view, update and delete appointments.

## Definitions & Acronyms

|  |  |
| --- | --- |
| Terms | Definitions |
| EHR | Electronic Health Record is the digital version of patient medical record. |
| HTTP | Hypertext Transfer Protocol is the communication protocol between web browser and web server. |
| REST | Representational State Transfer is an architectural style for an application program interface (API) that uses HTTP requests to access and use data. |
| SendGrid | SendGrid is cloud-based SMTP solution providing email delivery service. |
| SMTP | Simple Mail Transfer Protocol, is a standard communication protocol for email transmission, being used by email server to send and receive emails between senders and receivers. |

## References

# Overall description

## Product Perspective

This project MedCare is a web application, acting as the prime connection between patients and the clinic. The product is seen as the transition from traditional communication mean of phone call bookings and physical health information to a more convenient and modern approach with technological involvement. MedCare can be adopted by any medical center and peacefully connected with existing databases of the clinic to handle patient and doctor information. However, in the current scope of initial development, this is a new, stand-alone product that is not extended from any existing system and has its own separate database.

## Product Features

MedCare will have the following core functionalities:

* ***Appointments management***: Patient users have the ability to make appointments with doctors and delete the registered appointments when needed. Medical staffs are able to control the appointments in case the patient makes phone call or traditional request for booking/canceling appointments. Any changes on a registered appointment shall be notified to the medical staffs and corresponding doctor. Doctors can manage their available hours for appointment time selection. Medical staff can create new patient profile, which is not linked to a system user, for the purpose of appointment registration only.
* ***Medical records management***: Doctors have the ability to add patients’ health information such as medical records and prescriptions. And patients can safely access and view their EHR.
* ***User management***: Admin have the ability to register new users including admin role, doctor role, and medical staff role. Admin manages users’ activities through log recording.
* ***Content management***: Admin have the ability to edit website content, including managing branches, services, and doctors’ profiles.

## User Classes and Characteristics

There are four classes of users that interact with this system: patient of the clinic, medical staff, doctor, and system administrator. The users of this system are required to possess the technical know-how and a device having internet connection to use this system. Users can only access their corresponding functionalities with a login to the system.

Patient can do the following key functions:

* Book/cancel appointment with a doctor
* View upcoming appointments

Medical staff can do the following key functions:

* Create appointment linking a patient with a doctor
* Cancel appointments
* Create and edit patient’s profile
* Manage doctor’s available hours

Doctor can do the following key functions:

* View related upcoming appointments
* Manage appointment availability

Administrator can do the following key functions:

* View logs
* Create system users (doctor, administrator and medical staff)
* Manage website content (doctor’s profile, branch, service)

## Operating Environment

The operating system for the project would be:

* Cloud based system where the server would reside in Cloud
* Client/server system
* Operating system used would be, but not limited to, Windows
* The use of SQL as database
* Browsing system used would be, but not limited to, Google Chrome

## Design and Implementation Constraints

* User’s privacy must be handled carefully. Since there will be retrieval of EHR, certain level of data encryption needs to be performed as an effort to protect user data.
* Languages that will be implemented is JavaScript, for both client- and server-side programming. The application front-end will be developed using ReactJS framework. The application backend will be built using Express – a NodeJS web application framework
* Database in used will be MySQL
* The email notification system will be implemented using a third-party service “sendgrid.com”

## User Documentation

## Assumptions and Dependencies

* Since email notification system depends on third-party service, our email function will be affected in case of any issue happens to “sendgrid.com”

# Specific requirement system

## Users (Common behaviors of classes of users)

* + 1. **Login to the system**

Actor: System user (already has an account – patient, doctor, staff, admin)

Purpose: Access the respective system functions according to the user’s role

Trigger: Press LOGIN button after providing valid credentials

From the homepage, users, once registered, can login to their account. The user provides his/her registered email address and password. The password field covers what the user types with secret replacements. After successful verification, patient can access respective system functionalities.

Input fields:

* Email (required)
* Password (required)

Require:

* Login button
  + 1. **Log out from the system**

Actor: Logged in user of the system

Purpose: User wants to sign out from the system and terminate the session of his/her account

Trigger: Press SIGN OUT button

This use case starts when a sign out button is pressed by signed in user. On completion, the user will be able to log out from the system successfully.

Require:

* Sign out button
  + 1. **Change account password**

Actor: Logged in user of the system

Purpose: User wants to update new account’s password

Trigger: User selects CHANGE PASSWORD function in their profile

The system allows users to change their account passwords. User provides current password for authentication and gives new password. On completion, the patient’s credential is updated.

Input fields:

* Old password (required)
* New password (required)
* Confirm new password (required)

Require:

* Save new password button

## Patient

* + 1. **Register for a new account as new patient**

Actor: New patient user that has not visited the clinic before

Purpose: Patient user wants to create a system account

Trigger: Press REGISTER button, then select ‘NEW PATIENT’ option

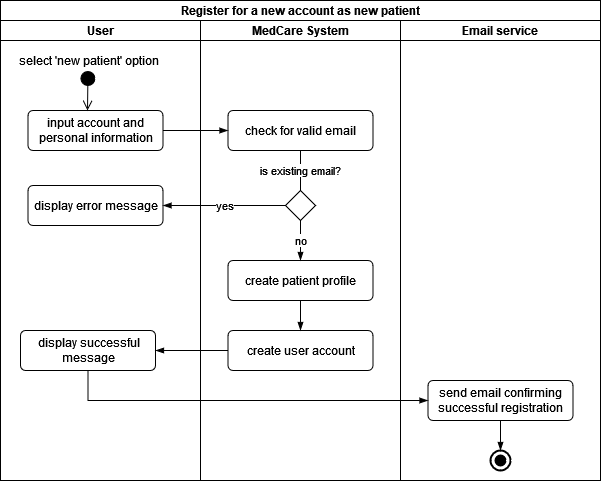
User musts specify: first name, last name, email address, mobile number, address, gender, password, date of birth. The system validates for newly-registered email address. Upon successfully validation, registration will be completed, patient receives email notification. Once completed, a new patient profile (with new patientID) will be created and linked to the account.

Input fields:

* First name (required)
* Last name (required)
* Email (required)
* Password (required)
* DOB (required)
* Address
* Phone number (required)
* Gender (required)
* Allergy

Require:

* Create user button



* + 1. **Register for a new account as existing patient**

Actor: Patient user that had visited the clinic before but has not had a system account

Purpose: Patient user wants to create a system account associating with their existing patient ID

Trigger: Press REGISTER button, then select ‘EXISTING PATIENT’ option

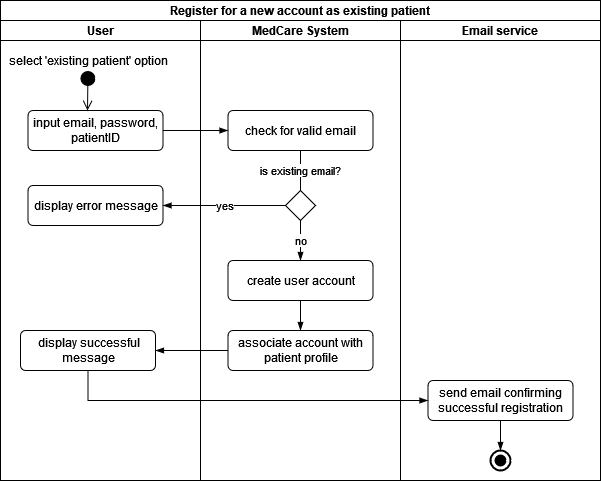
User musts specify: email address, password, and patientID. The system validates for newly-registered email address and patientID. Upon successfully validation, registration will be completed, patient receives email notification. Once completed, the account will be linked with the existing patient profile in the database.

Input fields:

* First name (required)
* Last name (required)
* Email (required)
* Password (required)
* Patient ID (required)

Require:

* Create user button



* + 1. **Search for a specific doctor**

Actor: Logged in patient user

Purpose: User wants to quickly look up a specific doctor using doctor’s name. This searching function can be used to view doctor’s profile or to book appointment with that doctor.

Trigger: Press SEARCH DOCTOR button on the home page (after filling the box with doctor’s name and selecting doctor’s gender)

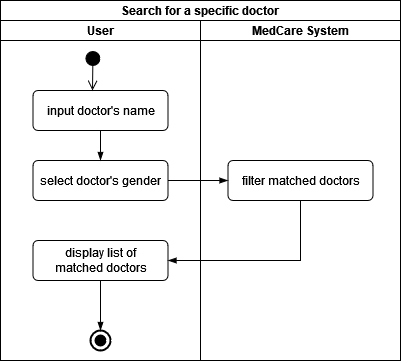
The patient user enters doctor’s name in search box. Results are displayed accordingly with brief descriptions of the matched doctors, including doctor’s working branch, provided service, and availability by date.

Input field:

* Search box to input doctor’s name
* Drop down menu to select doctor’s gender

Require:

* Search doctor button



* + 1. **Book an appointment**

Actor: Patient user

Purpose: User wants to make an appointment with a doctor.

Trigger:

The patient user wishes to make an appointment.

Case 1: The patient user selects specific hospital branch, the health service, the intended doctor, the date-time for appointment, and finally enters reason to visit.

Case 2: User selects doctor after search doctor (use case 3.2.3), then date-time for appointment, and finally enters reason to visit.

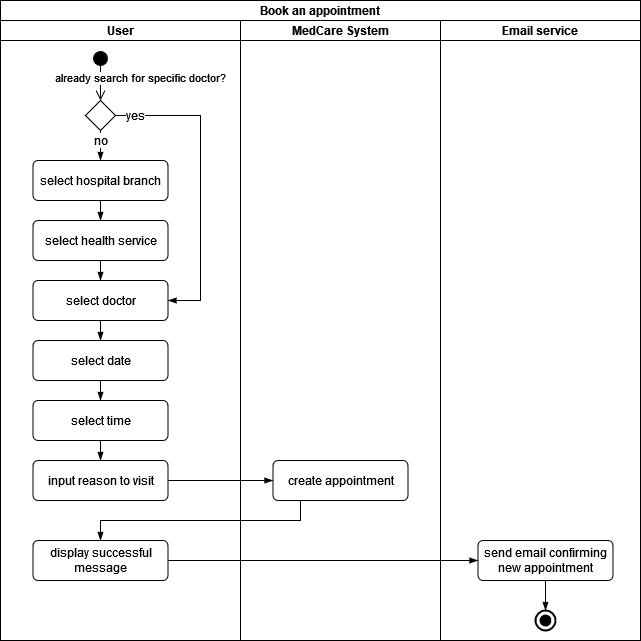
On completion, patient receives email confirmation for the registered appointment.

Input fields:

* Doctor ID (required)
* Date (required)
* Time (required)
* Reason to visit

Require:

* Book appointment button



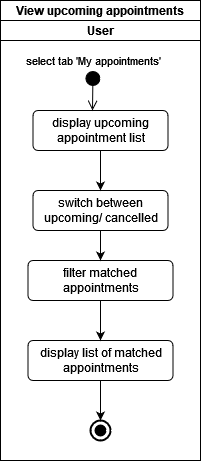
* + 1. **View appointments**

Actor: Patient user

Purpose: User wants to see all upcoming appointments booked by himself or by medical staffs

Trigger: Select the tab ‘My appointments’

Patient user navigates to profile information, then select tab ‘My appointments’. The system shall display the list of existing appointments associated with that patient user. User can switch between options of upcoming and cancelled filter to see the matched list of appointments.



* + 1. **Cancel an existing appointment**

Actor: Patient user

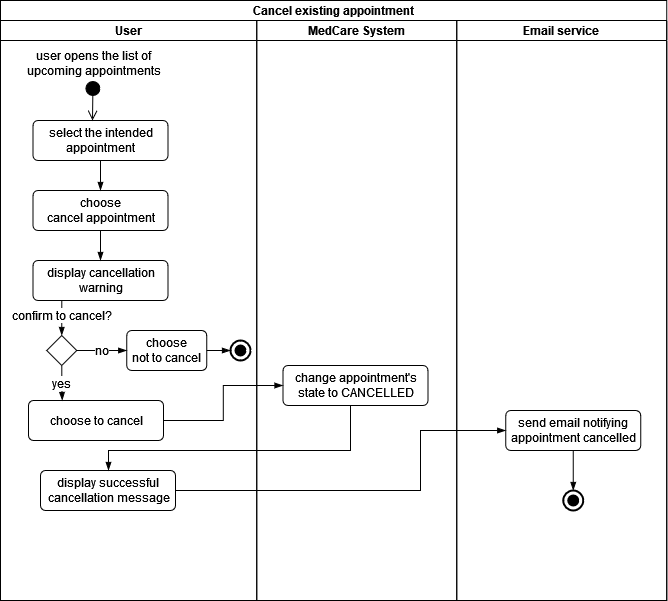
Purpose: User wants to cancel an existing appointment

Trigger: Press CANCEL button in appointment details page

The patient user navigates to his/her list of upcoming appointments. The user chooses the appointment that they wish to cancel. The user chooses cancel option for the selected appointment. On completion, the selected appointment is cancelled. The administrator and related doctor will be notified about the cancelation.

Require:

* Cancel button



* + 1. **Edit profile**

Actor: Patient user

Purpose: User wants to change his information registered on the system, including phone number, allergy and avatar image.

Trigger: Press ‘Edit profile’ button

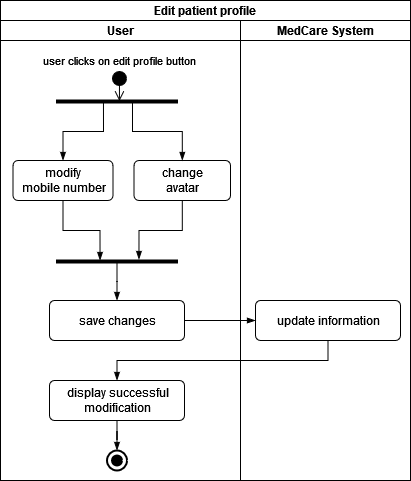
The system allows user patient to update their patient profile. For transparency, patient can only update their mobile number and avatar image. Other personal information including first name, last name, date of birth, address, gender will be modified upon request with a medical staff. On completion, user’s information has been changed.

Editable input fields (all optional):

* Phone number
* Allergy
* Avatar image

Require:

* ‘Edit profile’ button



## Medical staff

* + 1. **View all/filtered appointments**

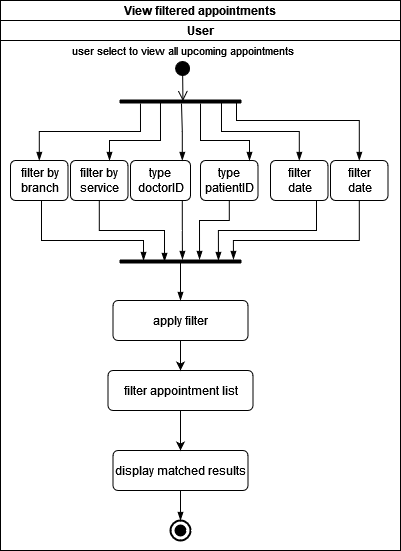
Actor: Medical staff user

Purpose: User wants to view the list of all appointments in the day, or wants to view selective list of appointments based on different date, specific branch, specific service, specific doctor, specific patient, or appointment status (active/ cancelled).

Trigger: Display by default once user logged into the system, or select the tab ‘Appointments’

This use case starts as the default when the user has logged into the system. The system displays all appointments for current and following dates. User can filter the list of appointments. On completion, the list of matched appointments shall be displayed correctly.

* User will be able to filter appointments by branch
* User will be able to filter appointments by service
* User will be able to filter appointments by doctor
* User will be able to filter appointments by patient
* User will be able to filter appointments by date
* User will be able to filter upcoming/cancelled appointments



* + 1. **Create an appointment**

Actor: Medical staff user

Purpose: User wants to make an appointment upon patient’s request.

Trigger: Press ‘Add new appointment’ button

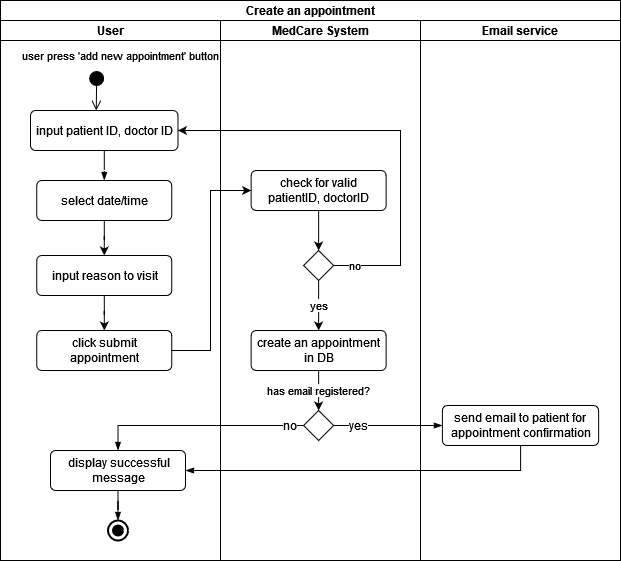
This use case starts when a patient requests an appointment using traditional booking methods and the patient has their profile registered. Medical staff user specifies patientID, branch, service, doctor, date/time, and reason to visit. On completion, if the patient has his/her account registered with an email address, patient receives email confirmation for the registered appointment.

Input fields:

* Patient ID
* Doctor ID
* Date/time
* Reason to visit

Require:

* ‘Submit appointment’ button



* + 1. **Create a patient profile**

Actor: Medical staff user

Purpose: User wants to add new patient profile because the patient is new to the clinic.

Trigger: Press ‘Add new patient’ button

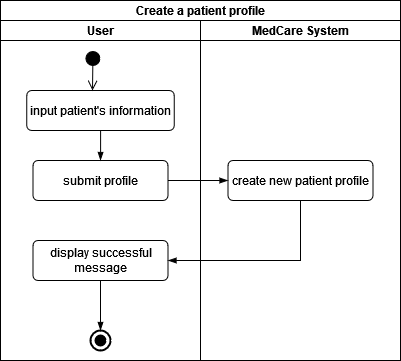
This use case starts when a new patient requests an appointment using traditional booking methods and the patient has not registered a profile. Medical staff user wants to create new patient profile. Medical staff user enters patient’s first name, last name, mobile number, address, gender, and date of birth. On completion, new patient profile is added to the system for appointment registration.

Input fields:

* First name (required)
* Last name (required)
* DOB (required)
* Address
* Phone number (required)
* Gender (required)
* Allergy

Require:

* ‘Add new patient’ button



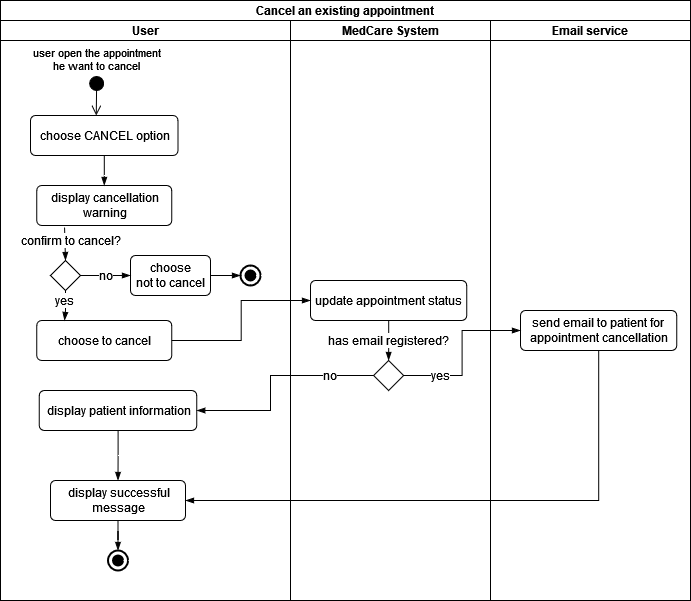
* + 1. **Cancel an existing appointment**

Actor: Medical staff user

Purpose: User wants to cancel an appointment due to changing doctor’s availability or upon patient’s request.

Trigger: Press ‘Cancel’ button in appointment details screen

This use case starts when a patient wants to cancel an existing appointment using traditional communication methods. Medical staff user navigates to the appointment she wants to cancel. Medical staff user selects option to cancel the intended appointment. The system sends email to patient if there is registered email.



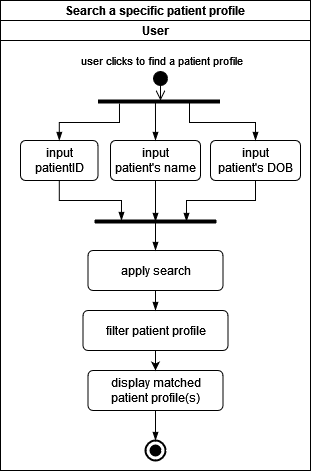
* + 1. **Search a specific patient profile**

Actor: Medical staff user

Purpose: User wants to look up a patient profile

Trigger: Select ‘Search patient’ tab

This use case starts when a medical staff want to look up a patient profile. The system allows user to input patientID or patient’s first name, last name. On completion, the system returns matched patient list.



* + 1. **Edit patient profile**

Actor: Medical staff user

Purpose: User wants to update information of a specific patient profile.

Trigger: Press ‘Edit profile’ button

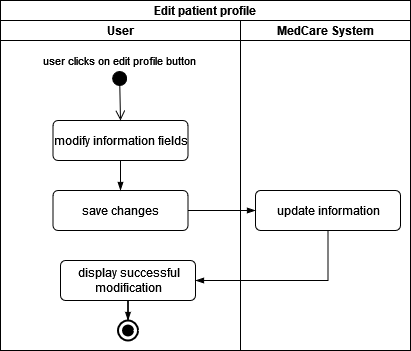
This use case starts when the medical staff want to change or update patient’s profile information. The system allows medical staff to change patient’s information. On completion, patient’s information has been updated.

Editable input fields (all optional):

* First name
* Last name
* DOB
* Address
* Phone number
* Gender
* Allergy

Require:

* ‘Edit profile’ button



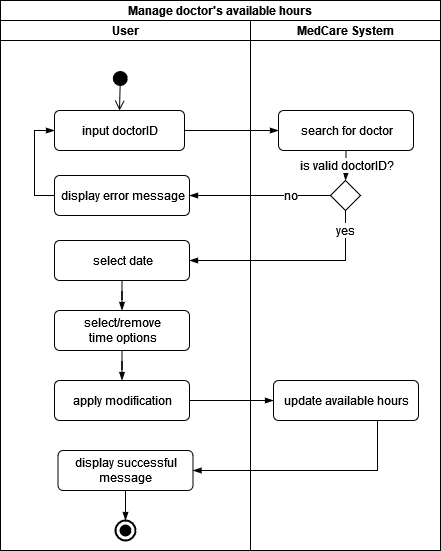
* + 1. **Manage doctor’s available hours**

Actor: Medical staff user

Purpose: User wants to manage appointment availability of a specific doctor

Trigger: Select ‘Manage doctor’ hours’ tab

The system allows medical staff to manage (add and remove) appointment availability. User find and select a doctor, the system allows user to select one- or multiple-time options for a specific date. Only the time option that does not have any appointment shall be able to deleted. On completion, the selected time frames of the date are available for appointment booking.



## Doctor

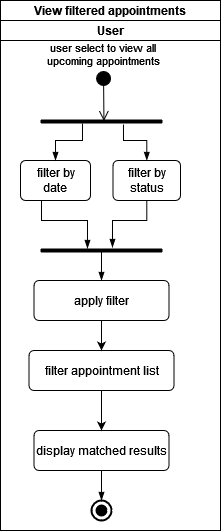
* + 1. **View all/filtered appointments**

Actor: Doctor user

Purpose: User wants to view the list of all appointments, or wants to view selective list of appointments based on different date, and (optional) on appointment status.

This use case starts as the default when the doctor user has logged into the system. The system displays all relevant appointments for current date and following dates. User can filter the list of appointments.

* User will be able to filter appointments by date
* User will be able to filter upcoming or cancelled appointments (optional)



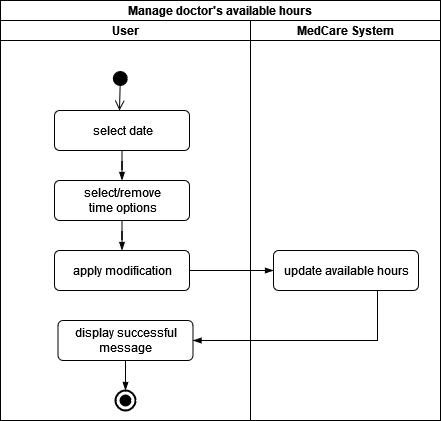
* + 1. **Manage appointment availabilities**

Actor: Doctor user

Purpose: User wants to add and/or remove their available hours for specific date.

Trigger: Select ‘Manage available hours’ tab

The system allows doctor user to select one- or multiple-time options for a specific date. Only the time frame that has no appointment shall be able to removed. On completion, the selected time frames of the date are available for appointment booking.



## Administrator

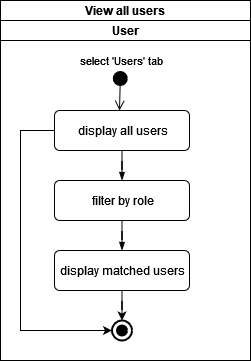
* + 1. **View all users**

Actor: Admin user

Purpose: User wants to view the list of all system users, or selective list of users based on role.

Trigger: Select ‘Users’ tab

The system shall show all system users on request of admin user. Admin is able to filter by role.



* + 1. **Edit doctor’s profile**

Actor: Admin user

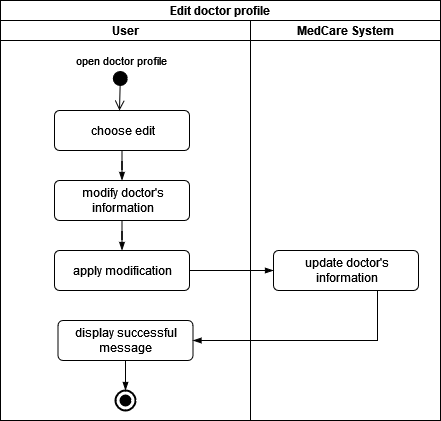
Purpose: User wants to edit doctor’s profile displayed on the web page.

Trigger: Press ‘Edit profile’ button in doctor’s details page.

The system allows administrator user to update doctor’s information including first name, last name, branch, service, gender, doctor’s about information, and avatar image. On completion, doctor user’s information has been changed.

Input fields:

* First name
* Last name
* Branch
* Service
* Gender
* About information
* Doctor image



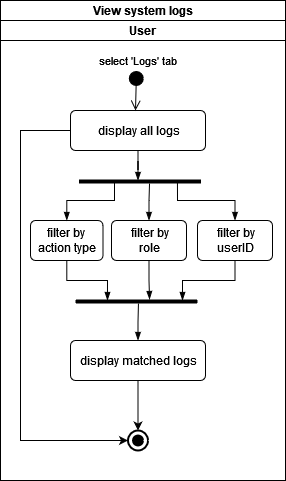
* + 1. **View system logs**

Actor: Admin user

Purpose: User wants to view activity logs of all the system users, or filter logs by specific action type, role, specific userID.

Trigger: Select ‘Logs’ tab

The system shall show all log items on request of admin user. Admin is able to filter logs by action type and by role, or search related logs of specific userId.



* + 1. **Add new branch**

Actor: Admin user

Purpose: User wants to create a new branch of the clinic.

Trigger: Press ‘Add new branch’ button

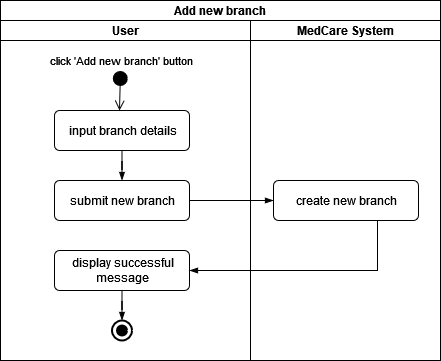
The system allows administrator user to add additional clinical branch. Administrator user specifies branch details. On completion, new branch is added.

Input fields:

* Branch name
* Address
* Description
* Image

Require:

* Add new branch button



* + 1. **Edit branch information**

Actor: Admin user

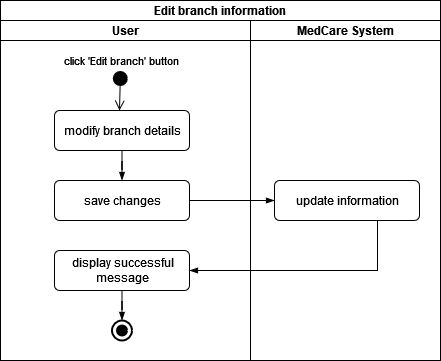
Purpose: User wants to update branch information, including branch name, address, description, and image.

Trigger: Press ‘Edit branch details’ button

The system allows administrator user to update information of existing branch. Administrator user edits branch details. On completion, branch details are updated.

Editable fields:

* Branch name
* Address
* Description
* Image



* + 1. **Add new provided service**

Actor: Admin user

Purpose: User wants to create new service the clinic provides.

Trigger: Press ‘Add new service’ button

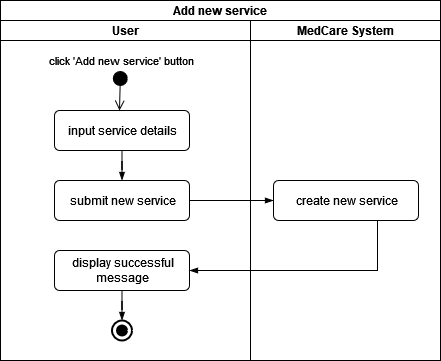
The system allows administrator user to add additional clinical service. Administrator user specifies service details. On completion, new service is added.

Input fields:

* Service name
* Description

Require:

* ‘Add new service’ button



* + 1. **Edit provided service**

Actor: Admin user

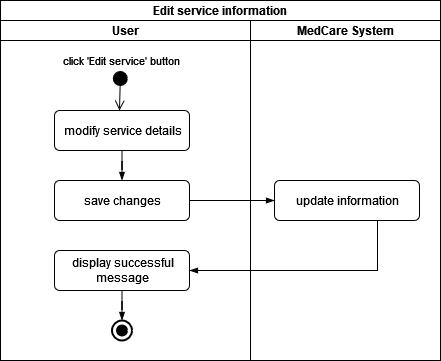
Purpose: User wants to update details of a service the clinic provides.

Trigger: Press ‘Edit service’ button

The system allows administrator user to update information of existing service. Administrator user edits service details. On completion, service details are updated.

Editable fields:

* Service name
* Description



* + 1. **Create new user (admin/ doctor/ staff)**

Actor: Admin user

Purpose: User wants to add new user to the system. User account role is among admin, doctor, medical staff.

Trigger: Press ‘Add new user’ in Users screen

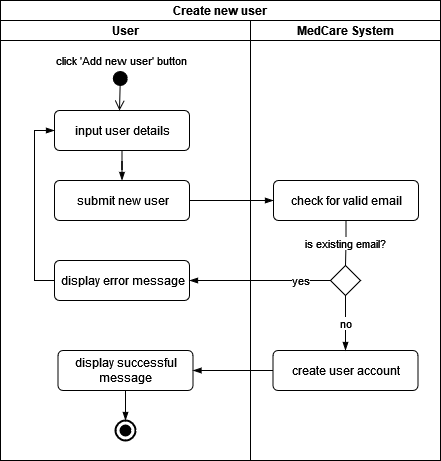
Administrator user wants to add new admin or doctor or medical staff user for the system. System requires specification of email address, password, full name, role (admin/ doctor/ staff), and avatar image. The system checks for newly-registered email address. Upon successful validation, new user is created. Registered email address and password can be used to login to the system.

Input fields:

* Email address
* Password
* First name
* Last name
* Role
* Avatar

Require:

* ‘Add new user’ button



# External interface required

## User Interfaces

* User interfaces for the patient shall be simple and easy to use so that a patient user or caregiver that is familiar with basic browser navigation skills can easily interact with the system.
* Notifications shall be displayed on the right-hand side of the screen with appropriate size.
* Once logged in, sign-out button will always be displayed for quick logout in all user interfaces.

## Hardware Interfaces

* User interfaces must be compatible with most of popular device screen sizes, including desktop, mobile, and tablet. Website layouts and components can be changed accordingly, however, the content shall remain the same.

## Software Interfaces

* The system shall be compatible with popular web browsers such as Google Chrome, Firefox, Safari and Microsoft Edge.

## Communication Interfaces

* Communication between client and server shall utilize REST-compliant web service and shall be served over HTTP secure.

# Other Nonfunctional Requirements

## Performance Requirements

* The system shall be interactive with quick response time and the delays involved shall be  
  minimized.
* The system shall support at least 200 concurrence users.

## Safety Requirements

* The system shall be highly resilient. In case of considerable database damage, the backed-up copy of the database shall be restored to recovery the state of the system up to the time failure  
  occurred

## Security Requirements

* The system shall guarantee information transmitted from and to the server is secure with no  
  change.
* The system shall maintain high level of security against unauthorized access.
* The system shall be protected against vulnerabilities
* User’s passwords shall be encrypted before storing in the database for enhanced security and user privacy.
* Changes in the database shall be logged.

## Software Quality Attributes

* Availability: The system should be available for 7 days and 24 hours
* Maintainability: All parts of the code should be easy to read. Most of the code shall be highly extensible for future development.

Appendix A. Analysis model

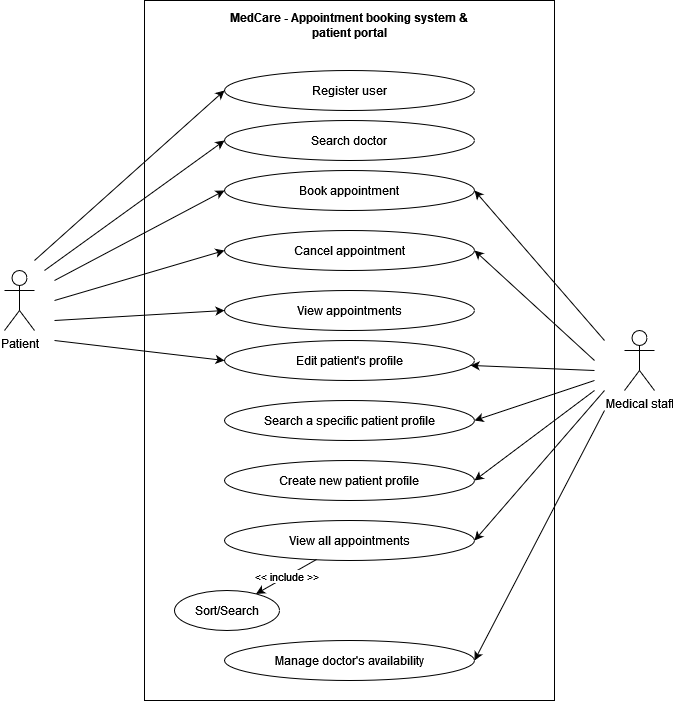


Figure 1. Use case diagram for Patient and Medical staff

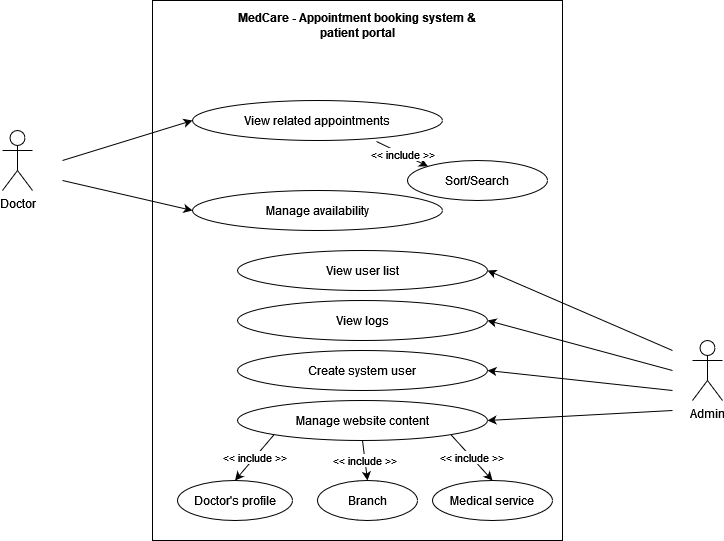


Figure 2. Use case diagram for Doctor and Administrator