DocCare

Use-Case Specification

Version 1.3

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 16/11/2023 | 1.0 | The initial version of Use-case Specification  Tran Le Quoc:Section 2.6 -> 2.10 Tran Kim Phat: 2.1 to 2.5, 2.24 Nguyen Mai Vinh: 2.6, 2.12 -> 2.14, Chu Hoang Phuc: 2.19 -> 2.22 Nguyen Trong Nghia:2.11, 2.15-> 2.18 | Tran Le Quoc  Tran Kim Phat  Nguyen Mai Vinh  Chu Hoang Phuc  Nguyen Trong Nghia |
| 17/11/2023 | 1.1 | Add Use-case Model | Tran Le Quoc  Nguyen Trong Nghia  Chu Hoang Phuc |
| 17/11/2023 | 1.2 | Draw Use-case Model | Nguyen Trong Nghia  Nguyen Mai Vinh |
| 18/11/2023 | 1.3 | Use-case Model Adjustments | Chu Hoang Phuc  Tran Le Quoc |

Table of Contents

[**1. Use-case Model 4**](#_heading=h.gjdgxs)

[**2. Use-case Specifications 4**](#_heading=h.30j0zll)

[2.1 Use-case: Register 4](#_heading=h.fq84e5wm0emh)

[2.2 Use-case: Login 6](#_heading=h.1fob9te)

[2.3 Use-case: Change password 7](#_heading=h.xfnd31hnfulk)

[2.4 Use-case: Send email reset password 8](#_heading=h.amcthzzrtkf)

[2.5 Use-case: Log out 9](#_heading=h.hsgifdag1psu)

[2.6 Use-case: Manage profile 10](#_heading=h.2e9v66fci5iy)

[2.7 Use-case: Handle Doctor Absent Request 11](#_heading=h.2et92p0)

[2.8 Use-case: View Intake Details and Rate Appointment 11](#_heading=h.tgncjzes8zdf)

[2.9 Use-case: Create Account for Doctor/Receptionist 12](#_heading=h.8o1j3yropex)

[2.10 Use-case: Delete Doctor/Receptionist Account 13](#_heading=h.z4b25xksc2nd)

[2.11 Use-case: Create Appointment for offline Patients 14](#_heading=h.vo8eittwvdx1)

[2.12 Use-case: View next appointment 15](#_heading=h.5tbo0nmqzxiq)

[2.13 Use-case: Intake daily reminder 15](#_heading=h.ps3rku8m3z2t)

[2.14 Use-case: Create monthly report 16](#_heading=h.j3pre823vik6)

[2.15 Use-case: View the appointment list 17](#_heading=h.fbsxicuuywxu)

[2.16 Use-case: Send absent request 17](#_heading=h.bzqi8hxngrjx)

[2.17 Use-case: Add a drug to intake 18](#_heading=h.r36brpwrwxg4)

[2.18 Use-case: Add statistics to prescription 19](#_heading=h.u6sn7kqld3n2)

[2.19 Use-case: Cancel an appointment 20](#_heading=h.bqshfzkduf07)

[2.20 Use-case: View notification 20](#_heading=h.jp7m0r20sv0n)

[2.21 Use-case: Messaging 20](#_heading=h.3vddzv6uq67z)

[2.22 Use-case: Search doctor 21](#_heading=h.c4lm7to1xpor)

[2.23 Use-case: Delete notification 22](#_heading=h.zapedwpup7ge)

[2.24 Use-case: Schedule an appointment 22](#_heading=h.l0xzuvr22pdk)

# Use-case Model

[*Put an image of the use-case diagram modeling all use-cases in this section*]



# Use-case Specifications

## Use-case: Register

| Use case Name | Register |
| --- | --- |
| Brief description | This use case describes how Customer (which is the patient) creates new account to use the application |
| Actors | Customer |
| Basic Flow | 1. At the landing screen, the user clicks on the “Get Started” button on the top right of the screen to open the bottom modal sheet for login. 2. The user clicks on the “Create an account” button 3. The user cusgoes to “Set up your profile” screen 4. The user enters Full name, Birthday, Email, Password, Confirm Password fields 5. The user ticks the “Agree with the Terms” 6. The user presses the “Sign up” button |
| Alternative Flows | **Alternative flow 1: User presses the “back” icon**   1. Users will be redirected to the landing screen.     **Alternative flow 2: User leaves some input sections blank**   1. The application alerts the user about the information requirements. 2. From #4 of the basic flow, the user continues to input the remaining fields.   **Alternative flow 3: User does not tick the checkbox of “Agree with the Terms”**   1. The application alerts the user about the confirmation of the terms in order to create an account. 2. From #5 of the basic flow, the user continues.   **Alternative flow 4: User has unstable Internet connection**   1. The application alerts this information to the user and refuses to send requests for authentication.   **Alternative flow 5: Information input by user has not met the criteria of safety or formating**   1. Users will be informed by notification and asked to re-input the information. 2. From #4, the user continues. |
| Pre-conditions | User goes to the “Set up your profile” screen.  User has a stable Internet connection when requesting signing up.  User enters the right, non-empty email format, and does not leave any section blank.  User does not enter the invalid birthday.  User enters the password that meets the safety criteria, and the Confirm Password section matches with the Password one.  User accepts the terms and conditions of the application.  The email must not be initially registered for any account. |
| Post-conditions | The system creates the new account with the given information.  System alerts the user if the email has been initially used to create an account and suggests using another one.  Users shall navigate to the Home screen provided they are successfully authenticated.  When low Internet connection happens during the process causing failures, alerts users to try again later when the connection is stable. |

## Use-case: Login

| Use case Name | Login |
| --- | --- |
| Brief description | This use case describes how Customer (which is the patient), Doctor, and Receptionist login to the application |
| Actors | Customer, Doctor, Receptionist |
| Basic Flow | 1. At the landing screen, the user clicks on the “Get Started” button on the top right of the screen to open the bottom modal sheet for login. 2. The user enters his/her email and password on the text form fields 3. The user clicks the button “Login” to request for authentication |
| Alternative Flows | **Alternative flow 1: User inputs wrong email format**   1. The application displays a notification on the screen about the wrong email format 2. From #2 of the basic flow, user reenters the email and password 3. Continue step 3 in the basic flow     **Alternative flow 2: The email input has not been registered**   1. The application displays a notification about the email not having been registered yet.   **Alternative flow 3: User inputs wrong password**   1. From #2 of the basic flow, user reenters the password 2. Continue step 3 in the basic flow   **Alternative flow 4: User has unstable Internet connection**   1. The application alerts this information to the user and refuses to send requests for authentication.   **Alternative flow 5: User presses the “Login with Google” button**   1. User will be navigated to another screen for Google authentication before being redirected back to the application   **Alternative flow 6: User presses the “I have forgotten my password” button**   1. Users will navigate to the “Forgot Password” screen to start the use case 2.4   **Alternative flow 7: User presses the “Create an account” button**   1. User will navigate to the “Set up your profile” screen for registration   **Alternative flow 8: User clicks outside of the modal**   1. The application shall close the bottom modal bottom sheet |
| Pre-conditions | User goes to landing screen  User has stable Internet connection when requesting logging in  User enters the right, non-empty email format, and password  The requested email must be initially registered for Customer. For Doctor and Receptionist, he/she must enter correctly the email which the administrator provided |
| Post-conditions | Users, of every role, shall navigate to the Home screen provided they are successfully authenticated.  When a wrong password error happens, the application displays the notification to inform users and ask them to reenter the correct one.  When the account has not been registered yet, the application requests them to try another one or create a new account.  When low Internet connection happens during the process causing failures, alerts users to try again later when the connection is stable. |

## Use-case: Change password

| Use case Name | Change password |
| --- | --- |
| Brief description | This use case describes how Customer (which is the patient), Doctor, and Receptionist change their accounts’ passwords |
| Actors | Customer, Doctor, Receptionist |
| Basic Flow | 1. At the Home screen, the user clicks on the drawer icon on the left of the app bar. 2. The user clicks on the “Profile” button in the drawer. 3. The user goes to the “Profile” screen, scrolling down to the “Change password” at the bottom of the screen, and presses that button.. 4. The user enters the current password and the new password, and then re-enter the new password in the pop-up dialog having the “Change password” title. 5. The user clicks the “Continue” button. 6. The user presses the “Confirm” button to change the password when the confirmation dialog appears. |
| Alternative Flows | **Alternative flow 1: User presses the “Cancel” button**   1. The “Change password” dialog will disappear.     **Alternative flow 2: User leaves some input sections blank when clicking “Continue”**   1. The application shows the alert messages to the user about the information requirements. 2. From #4 of the basic flow, the user continues to input the remaining fields.   **Alternative flow 4: User has unstable Internet connection**   1. The application alerts this information to the user and refuses to send requests for authentication.   **Alternative flow 5: Password input by user has not met the criteria of safety or formating**   1. Users will be informed by notification and asked to re-input the password. 2. From #4, the user continues.   **Alternative flow 6: User does not confirm the Change password action**   1. The confirmation dialog will disappear together with the “Change password” dialog.   **Alternative flow 7: User, when inputting, clicks outside of the dialog**   1. The current dialog will disappear. No information is modified. |
| Pre-conditions | User presses the “Change password” button on the “Profile” screen.  User has a stable Internet connection when requesting to change password.  User enters the right current password and does not leave any section blank.  User enters the new password that meets the safety criteria, and the Confirm Password section matches with the new password one.  User does not use the current password as the new password.  User confirms the change password action. |
| Post-conditions | The system changes the password of the user's account and notify the user when completed.  System alerts the user if the password has been initially used and suggests using another one.  Users shall stay at the Profile screen when completed.  When low Internet connection happens during the process causing failures, alerts users to try again later when the connection is stable. |

## Use-case: Send email reset password

| Use case Name | Send email reset password |
| --- | --- |
| Brief description | This use case describes how a Customer (which is the patient), Doctor, and Receptionist request emails to reset password when forgetting password. |
| Actors | Customer, Doctor, Receptionist |
| Basic Flow | 1. At the “Forgot Password” screen, the user enters his/her email. 2. The user presses on the “Send reset password email” button. 3. An email shall be sent to the email. The button is disabled for 10 seconds before allowing resending email. 4. The user can temporarily go to the link to change the password then go back to the current screen. 5. The user presses on the back button to navigate back to the entrance screen. |
| Alternative Flows | **Alternative flow 1: User presses the “Back” button**   1. The user will navigate back to the entrance screen.   **Alternative flow 2: User has unstable Internet connection**   1. The application alerts this information to the user. 2. From #2, the user continues. |
| Pre-conditions | User has a stable Internet connection when requesting to change password.  User enters the right, non-empty email format.  User must change the password through the link sent to his/her email.  Users must enter a new password which differs from the old ones and meets the safety criteria. |
| Post-conditions | The system changes the password of the user's account.  Users shall stay at the “Forgot password” screen when completed.  When low Internet connection happens during the process causing failures, alerts users to try again later when the connection is stable. |

## Use-case: Log out

| Use case Name | Log out |
| --- | --- |
| Brief description | This use case describes how Customer (which is the patient), Doctor, Receptionist, and Administrator log out of their accounts. |
| Actors | Customer, Doctor, Receptionist, Administrator |
| Basic Flow | 1. At the Home screen, the user clicks on the drawer icon on the left of the app bar. 2. The user clicks on the “Sign out” button in the drawer. |
| Alternative Flows | **Alternative flow 1: User closes the drawer**   1. The drawer will disappear. User is back at the Home screen.   **Alternative flow 2: User has unstable Internet connection**   1. The application alerts this information to the user and refuses to log out. |
| Pre-conditions | User presses the “Sign out” button on the drawer.  User has a stable Internet connection when requesting to log out. |
| Post-conditions | The system updates the authentication status of the user to logged out.  User is redirected to the waiting screen before being navigated to the Landing screen when successfully logged out.  When low Internet connection happens during the process causing failures, alerts users to try again later when the connection is stable. |

## Use-case: Manage profile

| Use case Name | Manage profile |
| --- | --- |
| Brief description | This use case describes how users of the application can view and customize their profile as they wish. |
| Actors | Customer, Doctor, receptionist |
| Basic Flow | 1. At the home screen, the user taps on the left corner of the header bar to open a drawer with several options. 2. The user taps on the profile button. A separate screen will show up with many text boxes containing their current profile information. 3. The user taps on the textbox containing information that they want to alter. 4. The user enters their new information 5. The user taps the “Save” button. The application shows a confirm popup. 6. Choose ‘Cancel’, and they can go back to basic step 3. 7. Choose ‘Confirm’, their profile will be updated, and the screen goes back to basic step 2. |
| Alternative Flows | **Alternative flow 1: The user input is not in the right format**   1. An error alert will be shown on top of the altered box. 2. The user made changes in basic step 3 until the box received valid input.     **Alternative flow 2: The user pressed ‘Save’ while there are still errors in their changes**   1. An error pop-up will appear. 2. After confirming, the user goes back to basic step 3.   **Alternative flow 3: The user confirms their changes while having a bad internet connection:**   1. An error pop-up will be shown if their information didn’t get updated correctly 2. After confirming, the user goes back to basic step 5.   **Alternative flow 4: The user presses the back button at the top left corner of the profile screen**   1. The user will be back to the screen that appears after basic step 1 2. Choose the “Profile” option again, and every change that has been made before will still be there, but it hasn’t been updated yet. 3. The user can choose to stay at basic step 3 or continue with step 4 |
| Pre-conditions | The user has successfully login to the application and got valid profile information |
| Post-conditions | The user can view detailed information of their profile. If they made any changes, it has to be valid and has to have a stable connection for the update to go through. |

## 

## Use-case: Handle Doctor Absent Request

| Use case Name | Handle Doctor Absent Request |
| --- | --- |
| Brief description | This use case involves the receptionist's actions to review and take action on absent requests submitted by doctors. |
| Actors | Receptionist |
| Basic Flow | 1. Receptionist navigates to the "Absent Requests" section. 2. System displays a list of pending absent requests from doctors. 3. Receptionist selects a specific absent request to review its details. 4. System presents details of the absent request including the doctor's name, date(s) of absence, reason, and any additional notes provided. 5. Receptionist evaluates the request and decides to:   a. Approve the absence:  i. Receptionist selects the option to approve the absence request.  ii. System updates the doctor's schedule accordingly, marking the requested dates as absent.  iii. System notifies the doctor about the approved absence.  b. Dismiss the absence request:  i. Receptionist selects the option to deny the absence request.  ii. System retains the doctor's regular schedule without marking the requested dates as absent.  iii. System notifies the doctor about the dismissal of the request.   1. System updates the status of the absent request (approved/dismissed). 2. Receptionist is redirected back to the list of pending absent requests. |
| Alternative Flows | **Alternative flow 1: If there are no pending absent requests:**   1. System displays a message indicating that there are no requests to handle. 2. Receptionist exits the "Absent Requests" section. |
| Pre-conditions | Receptionist has log-in to the app |
| Post-conditions | The absent request is either approved or dismissed by the receptionist.  Doctor's schedule is updated accordingly based on the receptionist's decision. |

## Use-case: View Intake Details and Rate Appointment

| Use case Name | View Intake Details and Rate Appointment |
| --- | --- |
| Brief description | This use case enables a customer to access and review their intake details, such as diagnosis, prescribed medications, doctor's notes, and provide a rating for the appointment experience. |
| Actors | Customer |
| Basic Flow | 1. Customer navigates to the "Intake History" section. 2. System displays a list of past appointments with their intake details associated with the customer's account. 3. Customer selects a specific intake session they want to review. 4. System presents detailed information related to the selected appointment, including:   a. Diagnosis provided by the doctor.  b. Prescribed medicines with dosage and instructions.  c. Notes from the doctor regarding the intake session.   1. Customer reviews the intake details provided. 2. Customer has the option to rate the appointment experience using a star-rating system from 1 to 5 stars 3. Customer submits the rating. 4. System records the provided rating, associating them with the specific appointment details. |
| Alternative Flows | **Alternative flow 1: If there are no past intake sessions available for the customer:**   1. System displays a message indicating no intake details are available. 2. Customer exits the "Intake History" section. |
| Pre-conditions | 1. The customer has an active account and is logged into the app. |
| Post-conditions | 1. The customer successfully views the intake details including diagnosis, prescribed medicines, and doctor's notes. 2. The customer provides a rating for the appointment. 3. System records the rating provided by the customer for future reference. |

## Use-case: Create Account for Doctor/Receptionist

| Use case Name | Create account for Doctor/Receptionist |
| --- | --- |
| Brief description | This use case describes the process by which an admin creates new accounts for doctors and receptionists within the system. |
| Actors | Admin |
| Basic Flow | 1. Admin navigates to the "Manage Accounts" section within the admin dashboard. 2. Admin selects the option to create a new account for a doctor or receptionist. 3. System displays a form or interface prompting the admin to enter necessary details for the new account, including:   a. Full name  b. Email address  c. Password  d. Birthday  e. Phone number  f. Role (doctor/receptionist)  g. Specialty (for doctors)  h. Working shift(Day of the week and period in that day)   1. Admin fills in the required information for the new account. 2. System display a form from what admin filled and ask for confirmation 3. Admin confirms and submits the form. 4. System validates the entered information for accuracy and completeness. 5. The form is valid and Admin receives a confirmation message indicating the successful creation of the new account. 6. Admin can either create another account or exit the account creation process. |
| Alternative Flows | **Alternative flow 1: If any mandatory fields are left empty or contain invalid information:**   1. At step 5, the system prompts the admin to fill in the missing or correct the invalid information before proceeding. 2. Continue the step 6   **Alternative flow 2:If there are existing accounts with similar information**   1. After step 8, the system notifies the admin to ensure uniqueness in the entered details before creating the new account. 2. Admin makes necessary changes to ensure uniqueness at step 5 and resubmits the form. |
| Pre-conditions | Admin is logged into the system using valid administrative credentials. |
| Post-conditions | A new account for the doctor or receptionist is successfully created in the system. |

## Use-case: Delete Doctor/Receptionist Account

| Use case Name | Delete Doctor/Receptionist Account |
| --- | --- |
| Brief description | This use case describes the process by which an admin deletes existing accounts of doctors or receptionists within the system. |
| Actors | Admin |
| Basic Flow | 1. Admin navigates to the "Manage Accounts" section within the admin dashboard. 2. Admin selects the option to view a list of existing doctor or receptionist accounts. 3. System displays a list of doctors or receptionists with their respective account details. 4. Admin selects the specific doctor or receptionist account they want to delete from the list. 5. System prompts the admin to confirm the deletion action for the selected account. 6. Admin confirms the deletion action. |
| Alternative Flows | **Alternative flow 1: If the selected account has associated data or ongoing appointments:**   1. System notifies the admin about the potential data loss or disruption in ongoing activities due to the account deletion. 2. Admin decides whether to proceed with the deletion or the action. |
| Pre-conditions | 1. Admin is logged into the system using valid administrative credentials. |
| Post-conditions | 1. The selected doctor or receptionist account is successfully deleted from the system. 2. Associated data (if any) is either removed or appropriately handled as per system specifications. |

## Use-case: Create Appointment for offline Patients

| Use case Name | Create Appointment for offline Patients |
| --- | --- |
| Brief description | This use case outlines the process by which a receptionist creates appointments for patients who visit the facility in person by selecting a doctor, choosing a suitable slot from the available schedule, and confirming the booking. |
| Actors | Receptionist |
| Basic Flow | 1. Receptionist accesses the appointment scheduling section. 2. System display a list of available doctors 3. Receptionist chooses the doctor either by searching for a specific doctor's name or his/her specialization. 4. System displays a calendar view for the selected doctor, showing their available slots. 5. Receptionist navigates through the calendar to view the available slots of the selected doctor for the desired date. 6. Receptionist selects a suitable available slot for the appointment. 7. System prompts the receptionist to confirm the selected slot. 8. Receptionist confirms the appointment slot selection. 9. System generates a unique appointment ID for the scheduled appointment. 10. Receptionist receives a confirmation message indicating successful appointment creation. 11. System also notifies the selected doctor about the new appointment. |
| Alternative Flows | **Alternative flow 1: If the selected doctor has no available slots for the desired date:**   1. System informs the receptionist about the unavailability and prompts to choose another date or doctor. 2. Receptionist selects an alternative available slot or doctor for the appointment.   **Alternative flow 2:If there are any technical issues when booking**   1. Receptionist receives a confirmation message indicating failed appointment creation. 2. Receptionist go back to step 1 and rebooking an appointment for offline patient |
| Pre-conditions | 1. Receptionist is logged into the system using valid credentials. 2. There is a need for an appointment for an offline patient. 3. Doctors' schedules and availability are properly updated in the system. |
| Post-conditions | 1. A new appointment for the offline patient is successfully scheduled in the system. 2. Receptionist and the selected doctor receive confirmation of the scheduled appointment. |

## Use-case: View next appointment

| Use case Name | View next appointment |
| --- | --- |
| Brief description | This use case describes how the doctor can see his/her next appointment with his/her patient. |
| Actors | Doctor |
| Basic Flow | 1. At the homepage or the doctor can navigate to home Page from another page, then he/she can see the next patient in the ‘Upcoming Appointment’ widget (including time and date). |
| Alternative Flows | **Alternative flow 1: Doctor has no appointment forward**   1. There must be a message for the doctors in this case.     **Alternative flow 2: Pending approval for appointment cancellation**   1. If a request for appointment cancellation is still pending approval by the Receptionist, the system indicates its status and provides an estimated time for approval. |
| Pre-conditions | Doctor must be logged-in/logged-up to the application  Doctor must be exactly at the home page to see the next appointment  There is an existing appointment with the patient of that doctor |
| Post-conditions | The edits or updates to the next appointment must be successfully applied to the ‘Upcoming Appointment’ widget |

## Use-case: Intake daily reminder

| Use case Name | Intake daily reminder |
| --- | --- |
| Brief description | This use case describes how a patient can see their schedule for taking medicines, and how they will get notified when the time comes. |
| Actors | Customer |
| Basic Flow | 1. At any screen that shows the bottom navigation bar, the patient taps the option with the pill icon or the second option from the left 2. The screen shows the list of medicines and the time to take them. 3. Medicine that has been notified is shown in the “Past Intake” section 4. At a fixed time, the application will send a notification to the patient’s phone |
| Alternative Flows | **Alternative flow 1: The user hasn’t got any prescription**   1. Nothing will be shown after basic step 1. 2. The user will not receive any notification from the application on intake-related issues.   **Alternative flow 2: The user doesn’t allow notification for the application**   1. The user can still see the intake details after basic step 1. 2. The patient will not receive any notification from the application on intake-related issues. |
| Pre-conditions | The user has successfully logged into the application. |
| Post-conditions | The user can view detailed information about their intake. If they give permission for notification, the patient will receive a notification every time they need to take their medicines. |

## 

## Use-case: Create monthly report

| Use case Name | Create monthly report |
| --- | --- |
| Brief description | This use case describes how the admin can create the monthly report based on information stored in the database, and under a printable format. |
| Actors | Admin |
| Basic Flow | 1. The admin taps the button on the upper left corner of the header bar. The drawer screen will show up along with some options. 2. Choose the “Generate Report” option, and a new screen appears with a few boxes. 3. Tap a textbox and fill in the needed information in that box. In the end, we will have the start, the final month, and the year. 4. Tap “Create” and a confirmation pop-up will be shown. The pop-up will also have a text field to enter. 5. Choose “Cancel’ to go back to basic step 3 6. The admin types a unique confirm code into the text field 7. Choose “Confirm ”. If the unique code is correct, after a brief loading screen, a preview of the report will be shown on the screen. 8. The admin taps the download icon to make a copy to their device. The application goes back to the drawer. |
| Alternative Flows | **Alternative flow 1: The textbox receives invalid information**   1. An error alert will be shown on top of the altered box 2. Admin stays in basic step 3 until the information is valid     **Alternative flow 2: The admin pressed ‘Create’ while there are still errors in their changes**   1. An error pop-up will appear. 2. After confirming, the user goes back to basic step 3.   **Alternative flow 3: The admin creates a report while having a bad internet connection:**   1. The report will be created much slower than usual 2. An error pop-up will be shown if the connection was cut while on the loading screen 3. After reconnecting, the admin goes back to basic step 4.   **Alternative flow 4: No record was found in the given time frame**   1. After the loading screen, the admin will receive a failed pop-up 2. Confirm will take them back to basic step 3. |
| Pre-conditions | The admin has successfully logged into the application. |
| Post-conditions | The admin can view the report after it is created. If there are issues during the generated process, every information is still being secure. |

## 

## Use-case: View the appointment list

| Use case Name | View appointment list |
| --- | --- |
| Brief description | This use case describes how the doctor can see their list of appointments with patient |
| Actors | Doctor |
| Basic Flow | 1. At the home screen, the doctor will see a calendar, and highlight their work day. There is also a separate section on top of the calendar showing their next appointment. 2. Doctor tap on the ‘Upcoming Appointment’ section.’ A separate screen will show their next appointment and past appointments during the day. |
| Alternative Flows | **Alternative flow 1: It’s not the doctor's work day.**   1. The separate section on top of the calendar in step 1 of the basic flow won’t show anything, hence no step 2 either. |
| Pre-conditions | The doctor has successfully logged into the app. |
| Post-conditions | The doctor got to see their appointments during the day |

## 

## Use-case: Send absent request

| Use case Name | Send absent request |
| --- | --- |
| Brief description | This use case describes how the doctor can send a request to the receptionist for a day cancellation because of his/her own business. |
| Actors | Doctor, Receptionist |
| Basic Flow | 1. The doctor navigates to the Home screen. 2. Then tap on the day where his/her desired cancellation appointment is 3. Doctor tap on the ‘Upcoming Appointment’ section. A separate screen will show his/her next appointments and past appointments during the day. 4. Doctor will tap on the ‘cancel’ button. 5. There will be a confirmed pop-up for her/him to think again. If he/she wants to cancel then tap on ‘yes’ else tap on ‘no’ |
| Alternative Flows | **Alternative flow 1: It’s not the doctor's working shift.**   1. The separate appearing screen in step 3 won’t show that shift, hence no remaining steps.   **Alternative flow 2: The cancellation operation of the doctor is down caused by the network.**   1. Everything that doctors operate won't be updated to the system as well as the request won’t be sent to the receptionist. 2. He will be notified when he is disconnected to the application. 3. But he won’t need to do it from scratch because the system temporarily saves what he did. |
| Pre-conditions | The doctor has successfully logged into the app.  The doctor must be at the exact screen to do the operation.  There must exist an appointment with his patient at his desired cancellation date and time. |
| Post-conditions | The doctor must receive a notification about the confirmation/decline from the receptionist.  The schedule will be updated correctly based on the result. |

## 

## Use-case: Add a drug to intake

| Use case Name | Add a drug to intake |
| --- | --- |
| Brief description | This use case describes how the doctor can add a medicine to the intake among the prescribing process for his/her current patient |
| Actors | Doctor |
| Basic Flow | 1. The doctor navigates to the Home screen. At this screen, it will default display the current patient for the doctor 2. Doctor tap on the ‘Upcoming Appointment’ section. A separate screen will show his/her next appointments and past appointments during the day. 3. Doctor will tap on the first patient of the list. 4. Doctor will tap on the ‘done’ button. 5. This will lead to another screen for adding medical statistics based on the measured figures that he/she takes after checking up on a patient. 6. Then he taps on the ‘Next’ button. 7. There will be a screen ‘Add Prescription’ and at there Doctor can edit by clicking on one of the medicine or add a new medicine by click on ‘Add a new medicine’ 8. There will be a screen for doctor to fill the field for drug information including: name, duration (days), dose per time, time of the day, to be taken 9. Doctor has to tap on ‘Add Prescription’ to complete the process |
| Alternative Flows | **Alternative flow 1: There is no medicine resource to add**   1. This could be at step 8 of basic flow when doctor types in wrong name of medicine 2. This will pop up an alert for doctors   **Alternative flow 2: The adding medicine operation of the doctor is down caused by the network.**   1. Everything that doctors operate won't be updated to the system as well as the request won’t be sent to the receptionist. 2. He will be notified when he is disconnected to the application. 3. But he won’t need to do it from scratch because the system temporarily saves what he did then he could continue at step 8   **Alternative flow 3: There is an error in api call**   1. This could be caused at step 8 when system try to call api to check whether that drug typed in is correct 2. The system must catch the exception and notify for the doctor for retrying then doctor just need to continue at step 8 |
| Pre-conditions | The doctor has successfully logged into the app.  The doctor must be at the exact screen to do the operation.  There must exist a medicine that satisfies the prescription for the patient. |
| Post-conditions | The patient must receive a notification about the prescription and information of medicines from the receptionist.  The patient’s prescription will be updated correctly based on the result. |

## Use-case: Add statistics to prescription

| Use case Name | Add statistics to prescription |
| --- | --- |
| Brief description | This use case describes how the doctor can add statistics to the prescription among the prescribing process for his/her current patient |
| Actors | Doctor |
| Basic Flow | 1. The doctor navigates to the Home screen. At this screen, it will default display the current patient for the doctor 2. Doctor tap on the ‘Upcoming Appointment’ section. A separate screen will show his/her next appointments and past appointments during the day. 3. Doctor will tap on the first patient of the list. 4. Doctor will tap on the ‘done’ button. 5. This will lead to another screen for adding medical statistics based on the measured figures that he/she takes after checking up on a patient including: 6. Then he would add the information of needed medicines prescribed to patient 7. He/she would tap on ‘Next’ to complete the adding statistics. |
| Alternative Flows | **Alternative flow 1: The adding statistics operation of the doctor is down caused by the network.**   1. Everything that doctors operate won't be updated to the system as well as the request won’t be sent to the receptionist. 2. He will be notified when he is disconnected to the application. 3. But he won’t need to do it from scratch because the system temporarily saves what he did then he could continue at the down network step.   **Alternative flow 2: There is an error in api call**   1. This could be caused at the final submit step when system try to load the statistics figures to the database 2. The system must catch the exception and notify for the doctor for retrying then doctor just need to resubmit |
| Pre-conditions | The doctor has successfully logged into the app.  The doctor must be at the exact screen to do the operation.  There must exist a statistics on the database side that satisfies the prescription for the patient. |
| Post-conditions | The patient must receive a notification about the prescription and information of medicines from the receptionist.  The patient’s prescription will be updated correctly based on the result. |

## Use-case: Cancel an appointment

| Use case Name | Cancel an appointment |
| --- | --- |
| Brief description | This use case describes how the doctor cancels the initially scheduled appointment of a patient. |
| Actors | Doctor |
| Basic Flow | 1. At every screen, there will be a header bar that has a notification icon. 2. The actor taps onto that icon and there will be a notification screen that will slide from the left, replacing the current screen with a screen that will display a list of notifications. 3. Here, the actor can scroll up or down to view the notifications of the nearest 7 days. |
| Alternative Flows | **Alternative flow 1: Actor does not have any notification**   1. In step 2, instead of displaying a list of notifications, the screen will display a line of text saying that the actor does not have any notification |
| Pre-conditions | The actor have logged on to the app |
| Post-conditions | The actor can view the list of notifications from the nearest 7 days. |

## 

## Use-case: View notification

| Use case Name | View notification |
| --- | --- |
| Brief description | This use case describes how the doctors, receptionists and customers can see their notifications |
| Actors | Doctor, Receptionist, Customer |
| Basic Flow | 1. At every screen, there will be a header bar that has a notification icon. 2. The actor taps onto that icon and there will be a notification screen that will slide from the left, replacing the current screen with a screen that will display a list of notifications. 3. Here, the actor can scroll up or down to view the notifications of the nearest 7 days. |
| Alternative Flows | **Alternative flow 1: Actor does not have any notification**   1. In step 2, instead of displaying a list of notifications, the screen will display a line of text saying that the actor does not have any notification |
| Pre-conditions | The actor have logged on to the app |
| Post-conditions | The actor can view the list of notifications from the nearest 7 days. |

## Use-case: Messaging

| Use case Name | Messaging |
| --- | --- |
| Brief description | This use case describes how the doctors, receptionists and customers can use the messaging service to send and receive messages |
| Actors | Doctor, Receptionist, Customer |
| Basic Flow | 1. At every screen, there will be a bottom navigation bar with a message icon. 2. The actor taps onto that icon, and they will be redirected to the messaging screen. 3. Here, the screen will display conversations with other users of the app, and the conversation will be displayed with the priority that the nearest will be on top. 4. If there are new messages in any conversation, that conversation will be highlighted by bold texts and a number indicating how many new messages there are. 5. The actor taps on to any conversation and will be redirected to another screen where messages in that conversation will be displayed. 6. The actorcan scroll up to see older messages, or they can start messaging by typing into the text field displayed near the bottom of the screen. After finishing typing, The actor can tap the send button next to the text field to send the message. |
| Alternative Flows | **Alternative flow 1: Actor does not have any previous conversation**   1. In step 2, instead of tapping to the message icon, the actor will tap the pill icon to redirect to the prescription screen. 2. At the end of the prescription, there will be the name of the doctor, and there will be a message icon next to that doctor’s name. Actor will tap that message icon to redirect to the conversation screen with that doctor. 3. Actor can start by typing into the text field and send messages, the conversation will then be tracked and be accessed by the initial method |
| Pre-conditions | The actor have logged on to the app |
| Post-conditions | The actor can send and receive messages as well as view older messages |

## 

## Use-case: Search doctor

| Use case Name | Search doctor |
| --- | --- |
| Brief description | This use case describes how customers can search for doctor to book an appointment |
| Actors | Customer |
| Basic Flow | 1. At every screen, there will be a bottom navigation bar with a doctor icon. 2. The customer taps on that icon and will be redirected to a screen displaying doctors with information of their specialization. 3. The customer can search by name or by choosing a category or by both. There will be a text field near the top of the screen. The customer will type in the name of the doctor they want to find. Or they can choose the specialization of the doctor by a filter on top of the text field. 4. The matching doctors will be displayed. The customer can tap the doctor to make a booking with that doctor. 5. A popup dialog will appear, asking for confirmation if the customer wants to book an appointment with that doctor. 6. If tap yes, the customer will be redirected to the booking screen. |
| Alternative Flows | **Alternative flow 1: Customer cannot find the doctor by name**   1. In step 4, if there is no matching doctor, then the screen will display a text saying that there is no doctor matching the requirements. |
| Pre-conditions | The customer have logged on to the app |
| Post-conditions | The customer can find the desired doctor for their specific needs |

## Use-case: Delete notification

| Use case Name | Delete notification |
| --- | --- |
| Brief description | This use case describes how doctors, customers, receptionists can delete a notification |
| Actors | Doctor, Customer, Receptionist |
| Basic Flow | 1. At every screen, there will be a header bar that has a notification icon. 2. The actor taps onto that icon and there will be a notification screen that will slide from the left, replacing the current screen with a screen that will display a list of notifications. 3. Here, the actor can scroll up or down to view the notifications of the nearest 7 days. 4. After having viewed, the actor can delete the notification by tapping on that notification and then swipe left, the notification will be pushed to the left direction 5. After swiping for a specific range, the notification will be deleted, and the other notifications will be pushed up for 1 position. |
| Alternative Flows | **Alternative flow 1: The actor does not swipe long enough**   1. In step 4, if the actor does swipe long enough, the notification will be pushed back to the normal position, and it will not be deleted.   **Alternative flow 2: The actor swipe right instead of left**   1. In step 4, if the actor swipes to the right, then the notification will not respond to the action. |
| Pre-conditions | The customer have logged on to the app |
| Post-conditions | The customer can delete the notification |

## Use-case: Schedule an appointment

| Use case Name | Schedule an appointment |
| --- | --- |
| Brief description | This use case describes how a customer schedules an appointment with a doctor |
| Actors | Customer |
| Basic Flow | 1. At ‘Find Doctor’ Screen, customer can search for his favorite doctor for the check-up 2. At the “Booking” screen, the user taps on his/her desired date to schedule. 3. The user selects one available time in the slot, and chooses a time for the application to remind him/her about the appointment. 4. (Optional) The user enters his symptoms. 5. The user presses the “Continue” button, then the confirmation dialog will appear. 6. The user confirms the request. |
| Alternative Flows | **Alternative flow 1: The actor presses any button on the navigation bar**   1. The actor will navigate back to the desired screen.   **Alternative flow 2: The actor taps outside of the confirmation dialog or click the “Cancel” button**   1. The actor will be back on the “Booking” screen without the loss of data. |
| Pre-conditions | The customer has a stable Internet connection.  The customer has no conflicted appointment. |
| Post-conditions | The database system receives the transaction and operates it.  The application shall inform the customer the result of booking with a detailed explanation when it fails. |

## 