

Project name: MediGram

Team 22

red parts are meant probably to be modified later

blue parts -- Jay(Xiaohui) Liu

purple parts -- Jiaqi Liu

Use case ID naming convention:

**UC -- Use Case, S -- Sign up, P -- Patient, C -- Care provider, PC -- P and C, SA --
System Admin**

A general template for use cases (you can copy this to save some time):

UC-

Use case name: verb + noun

Actors:

-

Goal:

Triggers:

-

Preconditions:

-
-

Postconditions:

-
-

Basic flow:

1. ...
2. ...
3. ...

Exception:

UC-01

Use case name: Sign in

Actors:

- patient
- care provider

Goal: log the user into APP and take them to their corresponding page

Triggers:

- user starts the APP

Preconditions:

- user has the APP installed

Postconditions:

- take patient to [UC-P-01](#)
- take care provider to [UC-C-01](#)
- take unregistered user to [UC-S-01](#)

Basic flow:

1. starts APP
2. enter user ID
 - 2.1. if user has no id, they can choose to sign up
 - 2.2 click “sign up” button
3. click “Sign in” button

Exception:

- Exception 3: user ID not found
- 2.1: take user to sign up page

[UC-S-01](#) ([US 03.01.01](#), [US 03.01.02](#), [US 03.01.03](#))

Use case name: sign up new account

Actors:

- new user

Goal: user creates a new account

Triggers: “sign up” button in [UC-01](#)

Preconditions:

- new user does not already have a user ID

Postconditions:

- new user ID created and saved by the APP

Basic flow:

1. enter new user ID
2. enter email
3. enter phone number
4. click “Create an account” button

Exception:

Exception 1: user ID not valid (too short or already taken)

1.1: shows warning message to user

1.2: go back to 1.

UC-P-01 (US 03.02.01)

Use case name: select problem

Actors:

- patient

Goal: see and edit patient profile, patient can select a specific body part to see its problems

Triggers:

- “Sign in” button in UC-01
- “Create an account” button in UC-S-01

Preconditions:

- patient is signed in

Postconditions:

- if patient search a problem, take patient to UC-P-02
- if patient sign out, take patient to UC-01
- if patient edit profile, a keyboard pops up and user can type in new information

Basic flow:

1. patient reads the body map
2. patient clicks one of the buttons
 - a. click “sign out” button to log out and go back to sign in page
 - b. click “edit” button to edit user profile
 - c. enter a body part number/keyword and click “search” button to search specific problems

Exception:

Exception 2: user input invalid (no such number/keyword)

2.1: enter again

UC-P-02 (US 01.02.01)

Use case name: view problems

Actors:

- patient

Goal: show list of problems with titles, dates, and number of records

Triggers:

- “Search” button in UC-P-01

Preconditions:

- patient entered valid body part number/keyword

Postconditions:

- search an existing problem: UC-P-05
- add new problem: UC-P-03
- edit existing problem: UC-P-04
- view records of a problem UC-P-06
- delete a problem by clicking on a problem and choose “delete problem” option
UC-P-07

Basic flow:

1. read the history of problems
2. scroll down/up to see more

Exception:

UC-P-03 (US 01.01.02, US 01.01.03)

Use case name: add problem

Actors:

- patient

Goal: add a new problem with title, date & time, and description

Triggers:

- “Add problem” button in [UC-P-02](#)

Preconditions:

- the list of problems is not full (is there a limit of # of problems?)

Postconditions:

- a new problem is added to history
-

Basic flow:

1. enter a problem title
2. set date & time with pop-up calendar and clock
3. write description of the problem
4. confirm changes
5. new problem added

Exception:

Exception 1: problem title already exists or too long (limit 30 chars)

1.1: go back to 1, choose another title

Exception 3: description too long (limit 300 chars)

3.1: go back to 3, write description again

[UC-P-04](#) ([US 01.01.02](#), [US 01.01.03](#))

Use case name: edit problem

Actors:

- patient

Goal: make change on an existing problem

Triggers:

- “edit” button for a problem

Preconditions:

- The patient has already found the problem he wants to change

Postconditions:

- problem editing page displayed

-

Basic flow:

1. edit title
2. edit date & time
3. edit description
4. confirm changes
5. problem edited

Exception:

Exception 1: **problem title already exists** or too long

1.1: go back to 1, choose another title

Exception 3: description too long

3.1: go back to 3, write description again

UC-P-05 (US 04.01.01, US 04.02.01-02)

Use case name: search problem

Actors:

- patient

Goal: search a problem from list of problems

Triggers:

- enter the problem's keyword in the text input box at the top

Preconditions:

-

Postconditions:

- problems matching with keywords show in the list

Basic flow:

1. user taps on search box
2. enter keywords
3. press "enter" key on virtual keyboard
4. search results show up (nothing shows up if no matching problems)

Exception:

UC-P-06 (US 02.01.01)

Use case name: view record

Actors:

- patient

Goal: view list of records of a problem

Triggers:

- “view record” button in UC-P-02

Preconditions:

- there is at least one problem
- user chose a problem to add record

Postconditions:

- user can see the records of the problem
- user can use slideshow mode by clicking “slideshow” button
- user can add a record by clicking add record button (UC-P-08)
- user can

Basic flow:

1. user reads the list of records
2. user can scroll down/up to view more records

Exception:

UC-P-07 (US 01.04.01)

Use case name: delete problem

Actors:

- patient

Goal: delete a problem from the list of problems

Triggers:

- “delete problem” option in the pop up menu after click on a problem in UC-P-02

Preconditions:

- at least one problem is in the list

Postconditions:

- the problem selected is deleted from the list

Basic flow:

1. click on a problem on the list
2. click “delete problem” option in the pop up menu

Exception:

Exception 1: no problem in list

1.1 nothing happens

UC-P-08 (US 01.05.01, US 02.02-06.01, US 02.08.01)

Use case name: add record

Actors:

- patient

Goal: add a detailed record to a problem

Triggers:

- add button in UC-P-06

Preconditions:

-

Postconditions:

- a new record with title, body location, picture, geo-location, date & time, and description is added to record list

Basic flow:

1. click add record button
2. enter title
3. add picture(s)
4. take a photo
5. add geo-location

6. add date & time
7. write a description
8. confirm changes
9. new record added

Exception:

Exception 7: description too long

7.1: do 7 again

UC-P-09

Use case name: edit record

Actors:

- patient

Goal: make change(s) to an existing record

Triggers:

- click on a record in UC-P-06

Preconditions:

- there is at least one record in record list

Postconditions:

- edited record with new information is updated

Basic flow:

1. click on a record in record list
2. edit title (user may leave it unchanged)
3. edit picture/photo (user may leave it unchanged)
4. edit geo-location (user may leave it unchanged)
5. edit date & time (user may leave it unchanged)
6. edit comment (user may leave it unchanged)
7. confirm changes
8. record updated

Exception:

Exception 2: title too long

2.1: do 2 again

Exception 6: comment too long

6.1: do 6 again

UC-C-01

Use case name: **view patient(6.01.01)**

Actors:

- Care providers

Goal: The care provider can view the list of patients that he assigned to, also he is able to find the add button and searching bar on the screen

Triggers:

- **Care provider log into the app**

Preconditions:

- the care provider need to connect to the internet, know his own id and the list of patient assign to him

Postconditions:

- If the add button is clicked, take the care provider to UC-C-03
- If some key words are typed in the searching bar, take the care provider to UC-C-02

Basic flow:

1. The care provider logged in
2. The system will show the searching bar at the top of the screen and provide the list of patient names with the edit button and a add patient button on the button to the user

Exception:1.if there is no patient assign to this care provider display “none”

UC-C-02

Use case name: add patient 7.01.01

Actors:

- care provider

Goal: care provider can add a new patient to his list of assigned patient

Triggers:

- care provider click the add button on the right side of the screen(UC-C-01)

Preconditions:

- The system will pop up a new activity that allow the care provider to add a patient

Postconditions:

- After the system received the input from the user, it append the new information into care provider's list.
- The care provider will have the the patient USER ID showing on view patient(UC-C-01)

Basic flow:

- 1.The care provider click on the add button in view patient (UC-C-01)
- 2. The system generate a new activity that allow the care provider to input Patient User ID,Dates and a Description of the patient.
- 3. The care provider input these informations
- The system generate a new tag for the new patient in the view patient screen (UC-C-01)

Exception:

- 1.The minimum user id should be at least 8 characters

- 2. The description should be no more than 300 words otherwise the information will not be solved

US-C-03

Use case name: browse patient problems (US 07.01.02)

Actors:

- care provider

Goal: The care provider will find the target patient by typing the user id in the searching bar and browse the patient problems

Triggers:

- The care provider starts to type in the searching bar

Preconditions:

- There is a list of patient who have already assign to the care provider

Postconditions:

- Display the target patient

Basic flow:

- 1.The care provider type the target patient's user id in the searching bar
- 2.The system finds the target patient and display him on the top of the list
- 3.The care provider click the target patient
- 4.The system will show a list of the problem that the patient has
- 5.For specific problems the care provider will type the key words for the problem
6. The system will find the specific problem
- 7.The care provider click into the problem
- 8.The system display information includes gps location, picture of the past

records

Exception:

- 1.There could be no such patient so display "none"

US-C-04

Use case name: add comment records to patient problems (US 07.01.03)

Actors:

- care provider

Goal:

- The care provider will be able to add comment to patient's problem

Triggers:

- The care provider find the specific problem he wants to comment

Preconditions:

- The patient has new record updated

Postconditions:

- The record of the patient will be updated and the the system will pop a message to indicate the patient that his record has been commented

Basic flow:

- 1.The care provider noticed that there a new record to be commented after he find the specific problems
- 2.The care provider click the problem he needs to be commented
- 3.The system will show all the record of the problem in a list and pop up a window which ask the care provider to type in comment
- 4.The care provider type his comment

Exception:

- 1.There could be no such patient so display “none”