University of Texas at Arlington

Fall 2023

CSE 5324-002 SFWR ENG I ANLY, DSGN, TESTING

Healthcare Monitor System

TEAM 3: Mobile Mavericks

Shubham Bhatt

Harsh Shah

Saaj Raval

Nitya Vodela

Anvita Veeramallu

12/04/2023

ITERATION 3

Table of Contents

PROJECT DESCRIPTION (CSE 5324-002)

App Name: **Healthcare Monitor**

Description: A Personal Health Monitoring System is a high-tech alternative to the conventional methods of managing patients' health. It can be easily accessible through mobile applications. This application will be used for keeping track of the user's vital signs, daily medicine intake, or following a diet regimen (regimen is a scheduled plan for the diet of a patient to be followed strictly which will help recover and improve patient health). Patients can register by entering personal information such as name, gender, age, weight, height, doctor's name etc.

Currently in the USA only, there are much more patients than there are medics who can treat them. COVID-19 pandemic has shown us that there is a global shortage of qualified doctors, and that the world is not prepared for this kind of unprecedented situation. This App can support the available medical staff. This system will reduce the workload of doctors by managing the patient's health as per the doctor's guideline. Apart from that, this system will keep track of the user's daily activities like workouts, steps walked and physical work. Common questions like when to take medication, what to eat, what things to avoid, how to maintain a perfect regimen can be answered with our Healthcare Monitor.

Although it's a relatively untouched market there are some competitors in the form of healthifyme app which measures calorie intake, Body Mass index using given data and daily physical activities and provides incentives for health-related milestones which can be taken as positive reinforcement, but it does not offer doctor consultation when needed and it doesn't have functionalities like "medication reminder". In a nutshell it is not a replication of a monitoring system. There's an app called MDlive that offers doctor consultations, but it doesn't have any other functionalities. So, that makes our app one of its kind to offer all medical and health-related functionalities on one system.

Major Functionalities:

Objective 1: Login/signup – Health system will be secure with password. Users must register and login before accessing any information available. This will Work as user functionality and be overlooked by the admin as well.
Objective 2: Vital tracker – System will maintain all vital signs such as Blood pressure, glucose level, Cholesterol, SpO2, Body-Mass index and sleep cycle etc. Data collected from the users will be used to develop their regimen accordingly. This will Work as user functionality.
Objective 3: Patient profile View – It will be accessible on mobile app. This will Work as user and admin functionality.
Objective 4: Medication Reminder – Can be delivered to the patients according to the prescription if needed.
Objective 5: Physical Activity Tracker – Activities like daily steps, Workouts and elevation climbed etc. now can be measured in the app to see the patient is behaving in daily life.
Objective 6: SOS button – In a health emergency, this button will ensure to call 9-1-1 and the emergency contact given by the user to get immediate attention needed. This can prevent any delays in first aid.
Objective 7: Doctor's Appointment Scheduler – Now Patients Can schedule their telemedicine appointments from the app to reduce the load on doctors and minimize waiting times.
Objective 8: Diet Regimen – This function will keep track of the patient's diet throughout the day. Let's say if patient's on a specific diet, the monitor will maintain the track and tell patients what to include and not include in their diet.

Resources Used:

1. We will be using the Internet Service of the phone for User Authentication.
2. We will be using phone storage to store the Database entries of the records.
3. We will be using audio Services for some features.
4. We will be using below listed sensors as well:
   Pedometer, Proximity, Gyroscope, Barometer, GPS, Motion sensors.

Team Members' Description:

1. Nitya Vodela: I accomplished my bachelor's degree in computer science and engineering with distinction, furthermore worked as an application development associate for a year in Accenture. Completed courses on python and java and implemented them in projects like heart disease detection using ML with python and worked on project management in Jira and bitbucket which included Java coding for adding certain features. Additionally, I can contribute towards designing, testing, documentation, and programming with strong presentation skills.

2. Anvita Veeramallu: I have a bachelor's in computer science and engineering. I have studied software engineering, agile software development, and operating systems. I also participated in training sessions for IBM Data Analyst that uses SQL and data analytics and Image Classification using Machine Learning, where we carried out supervised classification using single and multisets. I might contribute to the project by using java programming in addition to authoring the documentation and testing.

3. Harsh Shah: I achieved a first-class distinction in my computer engineering bachelor's degree in India. This educational journey acquainted me with programming languages like Python, Java, C, and C++. I excelled in courses

focused on object-oriented programming systems and software engineering. While I possess a strong grasp of Android fundamentals, my practical experience is limited. During my time in India, I completed two internships, one in web development and the other in machine learning. These experiences equipped me with skills in testing, UI design, writing, and programming. Leveraging these abilities, I am confident in contributing significantly to my team's semester project.

4. Shubham Bhatt: I completed my undergraduate studies in computer engineering with a 3.85/4.00 GPA. I took classes in Software Engineering and testing Methodology while earning my bachelor's degree. I also participated in a state level hackathon where we created crypto wallet android application. I have experience of 2 years in using python, PHP, SQL and numerous databases. I understand Java and Android Studio fundamentals and also have worked on android applications for a while. I will be able to assist with the application's documentation, coding, and testing.

5. Saaj Raval: I have successfully completed my undergraduate studies in Computer Engineering, achieving a GPA of 3.8. During my bachelor's program, I pursued coursework in Software Engineering, encompassing languages such as C, C++, SQL, Java, and Android. Complementing my academic pursuits, I engaged in practical projects during my internship, primarily focused on Angular (Node Full Stack) development. Furthermore, I possess foundational proficiency in Android development using Java. In the upcoming project, I am poised to contribute comprehensively across all facets, with a particular emphasis on coding and developmental aspects.

Requirement Table:

R No.	Requirement Statement	Line Reference	Weightage
R1	The system shall provide signup functionality for new users.	30	1
R2	The system shall have a secure login functionality.	30	1
R3	The system shall have a logout functionality.	30	1
R4	All users shall see their profile details.	36	2
R5	All users shall update/edit their profile details.	36	2
R6	The System shall track the vital signs of users.	33	2
R7	The system shall allow users to create a medication reminder.	38	2
R8	The system shall track all the physical activities done by the users.	40	3
R9	In case of an emergency, the system shall notify the emergency contact.	42	3
R10	In case of an emergency, the system shall dial 911.	42	4
R11	The system shall provide appointment scheduler functionality.	45	4
R12	The system shall provide ‘diet monitor’ functionality to users.	47	3

Use-Case List:

Use case #	Use case Name
UC 1	Sign up
UC 2	Login
UC 3	Logout
UC 4	Check Profile
UC 4.1	View details
UC 4.2	Edit details
UC 4.2.1	Update details
UC 5	Create medication reminders
UC 6	Tracking details
UC 6.1	tracking Vital signs
UC 6.2	tracking Physical activities
UC 7	Notify upon Emergency
UC 7.1	Notify emergency contact
UC 7.2	Notify 911
UC 8	Schedule Appointments
UC 9	Diet Monitoring

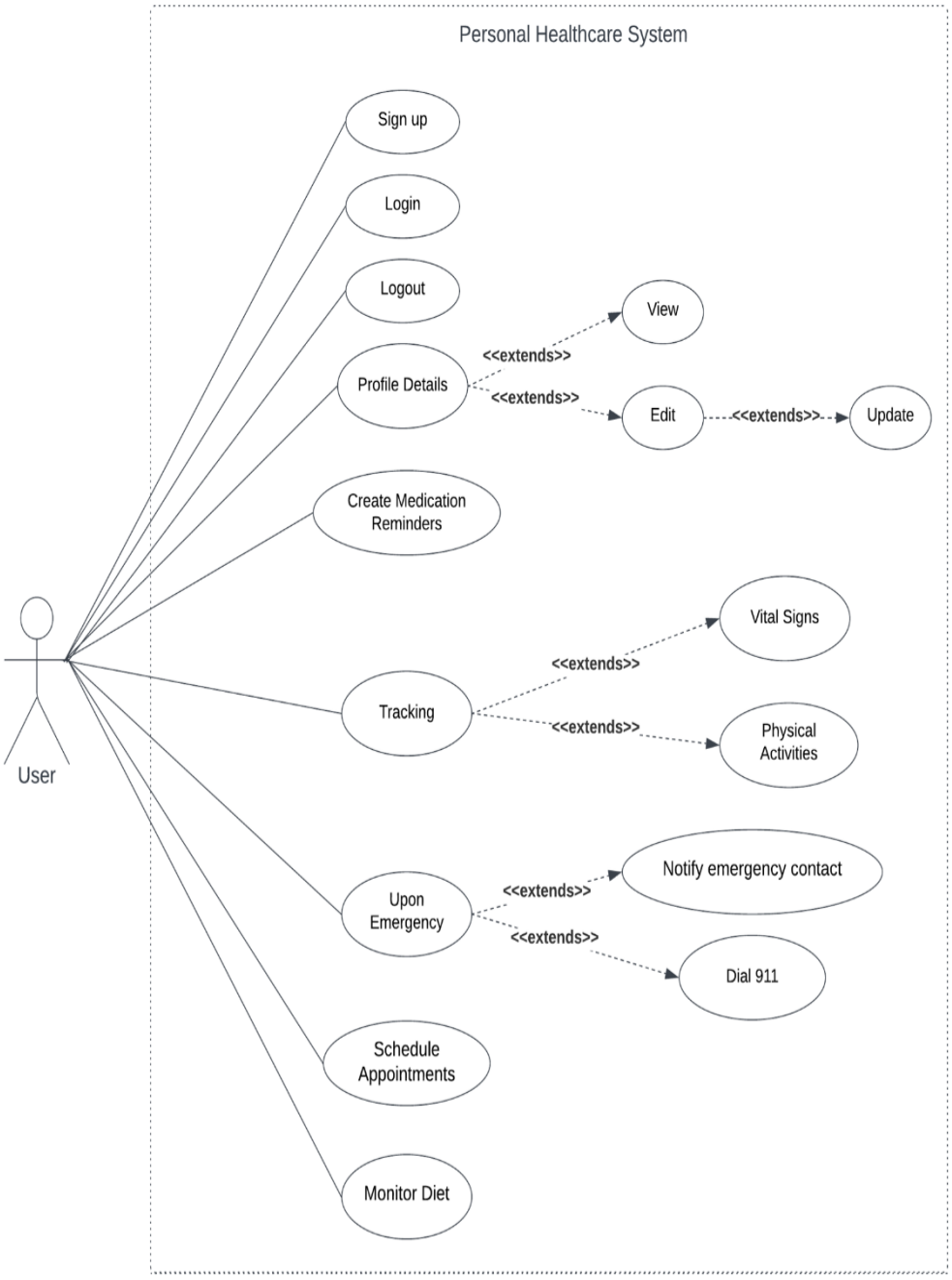
High Level Use-case:

TUCBW: *the use case begins with*
TUCEW: *the use case ends with*

- UC 1: Sign up
  - TUCBW user signing up for an account by clicking Sign-up
  - TUCEW user having an active account registered in the system
- UC 2: Login
  - TUCBW user giving their login credentials to access the account
  - TUCEW user logging in and viewing the homescreen
- UC 3: Logout
  - TUCBW user clicking on ‘Log Out’
  - TUCEW user logging out of application and viewing login page
- UC 4: check Profile
  - TUCBW clicking on ‘Profile’
  - TUCEW accessing profile options
- UC 4.1: View details
  - TUCBW clicking ‘View profile’
  - TUCEW user viewing their profile details
- UC 4.2: Edit details
  - TUCBW clicking ‘Edit profile’
  - TUCEW user able to edit the specific profile details
- UC 4.2.1: Update details
  - TUCBW clicking ‘Update profile’ after editing details
  - TUCEW user successfully updating their profile details
- UC 5: Create Medication reminders
  - TUCBW user creating reminders for medication
  - TUCEW user successfully getting reminders for each medication
- UC 6: Tracking details
  - TUCBW system keeping check of daily activities

    - TUCEW user being up to date with the tracker for daily activities
- UC 6.1: Tracking Vital Signs
    - TUCBW system tracking the vitals during an activity
    - TUCEW user is able to view their vitals reports page
- UC 6.2: Tracking Physical Activities
    - TUCBW system remind user to be active
    - TUCEW records being kept of user's physical activities in system
- UC 7: notify upon Emergency
    - TUCBW system activating necessary actions upon user facing an emergency situation
    - TUCEW user receiving medical attention
- UC 7.1: Notify emergency contact
    - TUCBW system sending an alert to emergency contact about a situation
    - TUCEW emergency contact being notified
- UC 7.2: Notify 911
    - TUCBW system placing an emergency call
    - TUCEW medical officials being notified
- UC 8: Schedule appointments
    - TUCBW system scheduling an appointment with a medical professional for user
    - TUCEW user getting information about their appointment details
- UC 9: Diet Monitoring
    - TUCBW system providing a diet plan according to the diagnosis of medical professional
    - TUCEW user accessing the diet plan.

Use case Diagram:



Requirements to Use Case Traceability Matrix

	Priority Weight	UC 1	UC 2	UC 3	UC 4	UC 4.1	UC 4.2	UC 4.2.1	UC 5	UC 6	UC 6.1	UC 6.2	UC7	UC 7.1	UC 7.2	UC 8	UC 9
R1	1	X															
R2	1		X														
R3	1			X													
R4	2				X	X											
R5	2				X		X	X									
R6	2									X	X						
R7	2								X								
R8	3									X		X					
R9	3												X	X			
R10	4												X		X		
R11	4															X	
R12	3																X
	Score	1	1	1	4	2	2	2	2	5	2	3	7	3	4	4	3

Note: Priority 1 is highest priority, work this first.

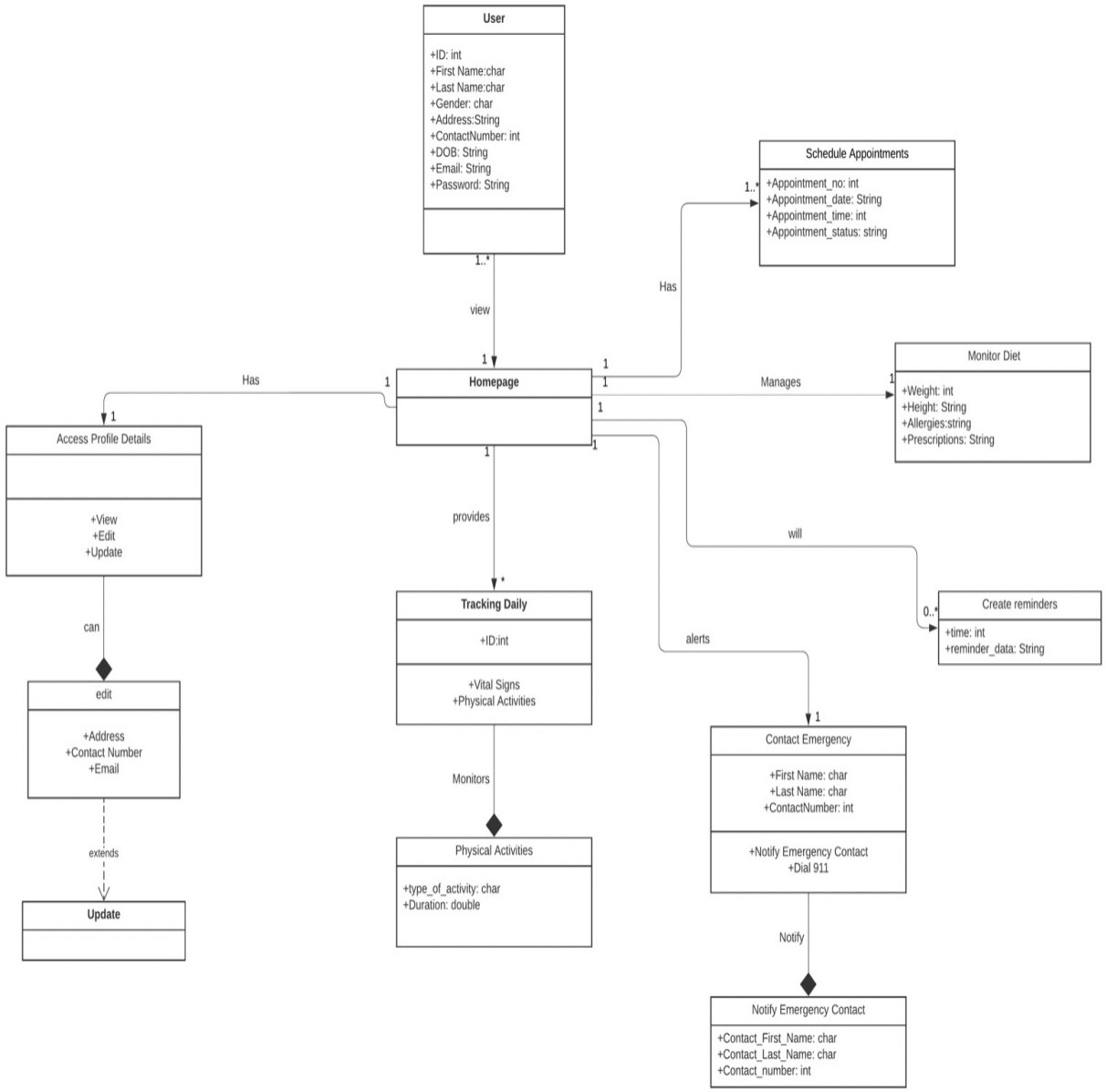
Use Case Increment Matrix

Use Case	Priority	Effort (Person-Weeks)	Depends On	Assigned To	Iteration 1 (10/06/23)	Iteration 2 (11/03/23)	Iteration 3 (12/04/23)
UC 1	1	2	None	SB, HS	2		
UC 2	1	1	UC 1	SR	1		
UC 3	1	1	UC 2	HS	1		
UC 4	4	2 (Actual 1)	UC 2	NV, AV	1		
UC 4.1	2	1	UC 4	SB	1		
UC 4.2	2	3	UC 4	SR, NV, AV	1	2	
UC 4.2.1	2	1	UC 4.2	HS	1		
UC 5	2	2	UC 2, UC 4, UC 6.1	SB, SR	1	1	
UC 6	5	4	UC 4	SB, HS, NV, AV	1	1	2
UC 6.1	2	2 (Actual 3)	UC 6	NV, AV	1	2	
UC 6.2	3	2	UC 6	SB, SR			2
UC 7	7	2	UC 2, UC 4	HS, SR		1	1
UC 7.1	3	1	UC 7	NV		1	
UC 7.2	4	1	UC 7	AV			1
UC 8	4	2	UC 4, UC 6.1	SB, HS			2
UC 9	3	3	UC 4, UC 6	SR, NV, AV		2	1
Total Effort		30			11	10	9

1-Person Weeks = 5 hrs.

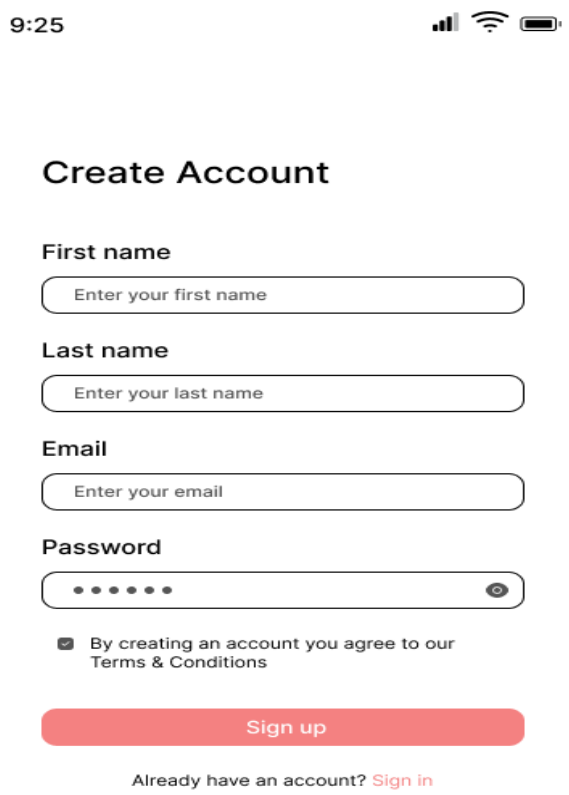
Team Members: Shubham Bhatt (SB), Harsh Shah (HS), Saaj Raval (SR), Nitya Vodela (NV), Anvita Veeramallu (AV).

Domain Model



Extended Use cases and UI:

1) Sign Up:

<table>
<tr><td colspan="2">Precondition: This use case assumes that the user is not registered yet.</td><td rowspan="5">Image 1.1
 
</td></tr>
<tr><td>Actor: User</td><td>System: Health Monitor</td></tr>
<tr><td>1. TUCBW the user clicking on "Sign Up" link to create an account.</td><td>0. System displays the Sign in screen.
(see image 1.1)

2. System displays Sign Up page UI.</td></tr>
<tr><td>3. The user clicks on "Sign up" button and inputs User details: First Name, Last Name, Email, Password, Checkbox for T&C to save the information in the database.

5. TUCEW user registering an account in application successfully.</td><td>*4. System displays a "User Registered Successfully" message and displays an error message for any missing information.</td></tr>
<tr><td colspan="2">Postcondition: The User can now Sign in to their account successfully.</td></tr>
</table>

2) Sign In:

<table>
<tr><td colspan="2">Precondition: This use case assumes that the user has an account to Sign in.</td></tr>
<tr><td>Actor: User</td><td>System: Health Monitor</td></tr>
<tr><td>1. TUCBW user enter their Email and Password, then clicks on "Sign in" button.

3. TUCEW the user successfully signing in and viewing the homepage.</td><td>0. System displays the Sign in screen. (Image2.1)

*2. System displays the Homepage UI and " User has signed in successfully" message upon receiving correct credentials else displays an "incorrect email/password" message. (image 2.2)</td></tr>
<tr><td colspan="2">Postcondition: The user can now access the healthcare monitor functionalities.</td></tr>
</table>

Image 2.1

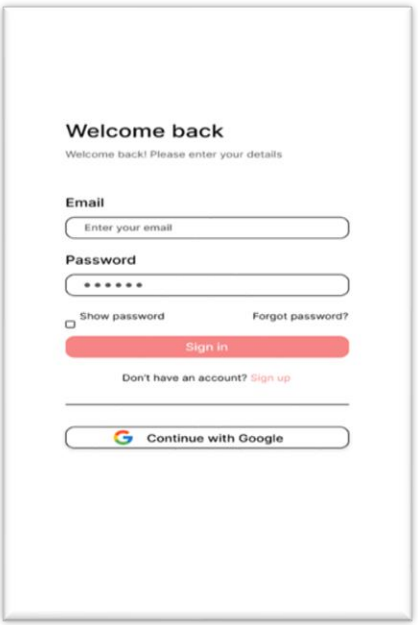
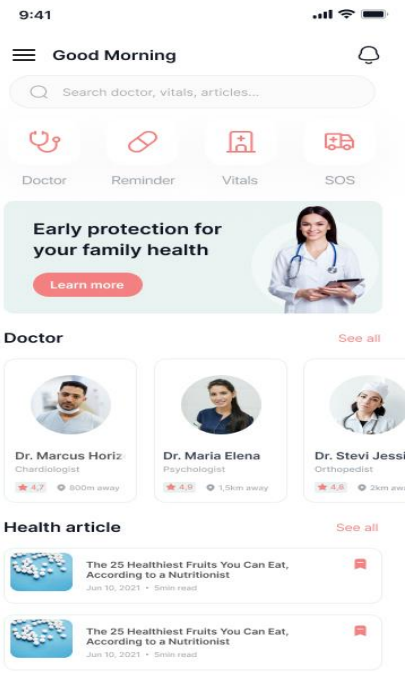
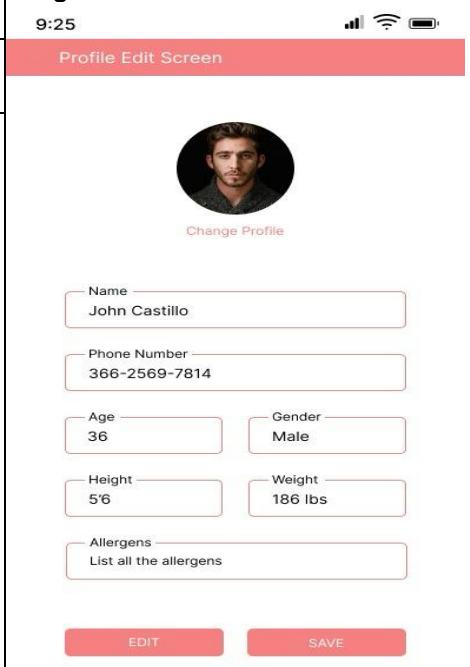


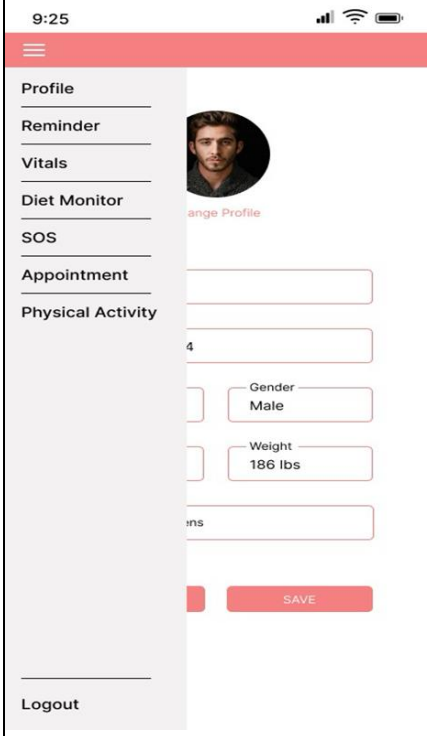
Image 2.2



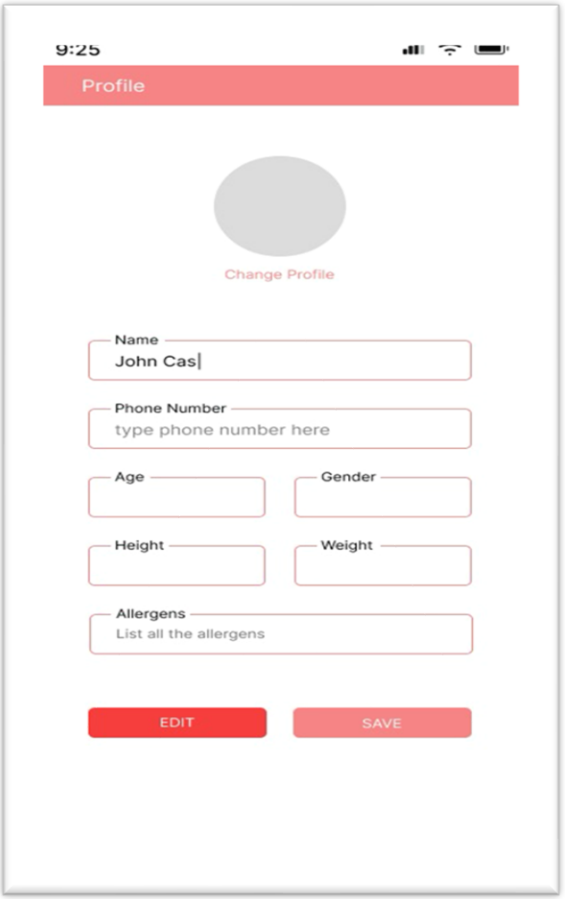
3) View Profile:

<table>
<tr><td colspan="2">Precondition: This use case assumes that user is viewing the Profile Screen</td><td rowspan="3">Image 3.1
 
</td></tr>
<tr><td>Actor: User</td><td>System: Health Monitor</td></tr>
<tr><td>1. TUCBW the user viewing their profile details: Name, Phone Number, Age, Gender, Height, Weight, Allergens. The user clicks on the Menu option.

3. TUCEW the user viewing the menu options.</td><td>0. System displays the Profile Screen. (image 3.1)

2. System displays Menu UI. (image 3.2)</td></tr>
<tr><td colspan="2">Postcondition: The user can view their profile details.</td><td>Image 3.2
 
</td></tr>
</table>

4) Edit Profile:

<table>
<tr><td colspan="2">Precondition: This use case assumes that user will edit profile</td><td rowspan="4">Image 4.1
 
</td></tr>
<tr><td>Actor: User</td><td>System: Health Monitor</td></tr>
<tr><td>1. TUCBW The user clicks on the "Edit" button to update profile details.

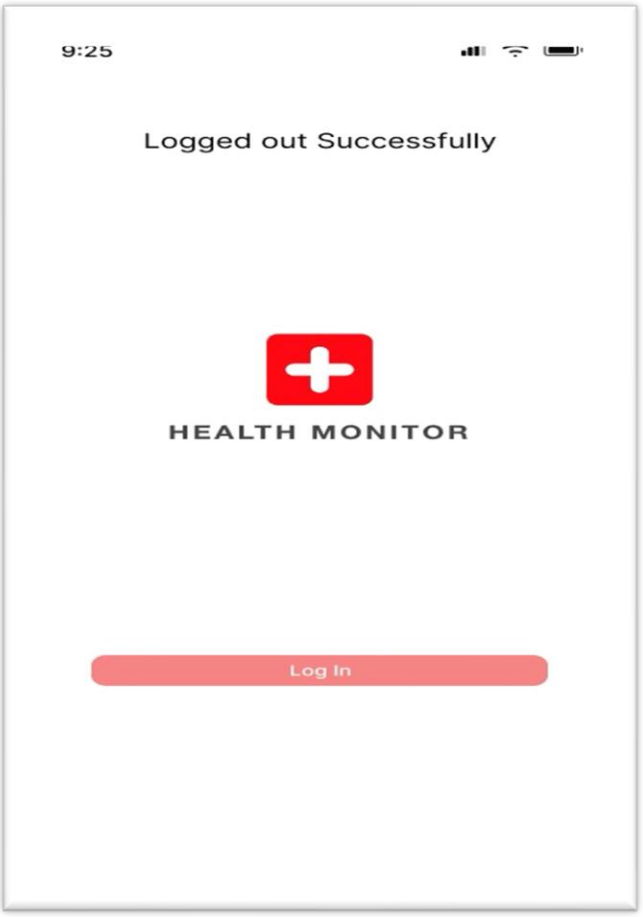
3. user updates required information and clicks on “Save”.

5. TUCEW the user updating profile details and clicking on the "Save" button.</td><td>0. System displays the Profile Screen. (Image 4.1)

2. System provides the Edit Profile UI.

*4. System displays the "Details updated successfully" message.</td></tr>
<tr><td colspan="2">Postcondition: The user can view their updated profile details.</td></tr>
</table>

5) Log Out:

<table>
<tr><td colspan="2">Precondition: This use case assumes that user is logging out</td><td rowspan="3">Image 5.1
 
</td></tr>
<tr><td>Actor: User</td><td>System: Health Monitor</td></tr>
<tr><td>1. TUCBW user clicking on "Logout".

3. TUCEW user logged out of account.</td><td>0. System displays Menu UI.

2. System displays a "User has logged out successfully" message. (Image 5.1)</td></tr>
<tr><td colspan="2">Postcondition: The user has logged out of the account successfully.</td></tr>
</table>

6) View/change Diet monitor:

Precondition: This use case assumes that user is viewing and adding to the diet monitor	
Actor: User	System: Health Monitor
1. **TUCBW** user viewing diet monitor details displaying the water intake and calories with the past records and will click on the plus symbol to add water/meal intake and time.	0. System displays the Diet Monitor Screen UI. (**Image 6.1)**
3. The user will input the amount of water ml/meal calories intake as per meal time.	2. System displays Add intake UI. **(Image 6.2)**
5. **TUCEW** user updating new record	4. System displays the user updated details
Postcondition: The user can view their new intake details and past records in Diet Monitor.	

Image 6.1

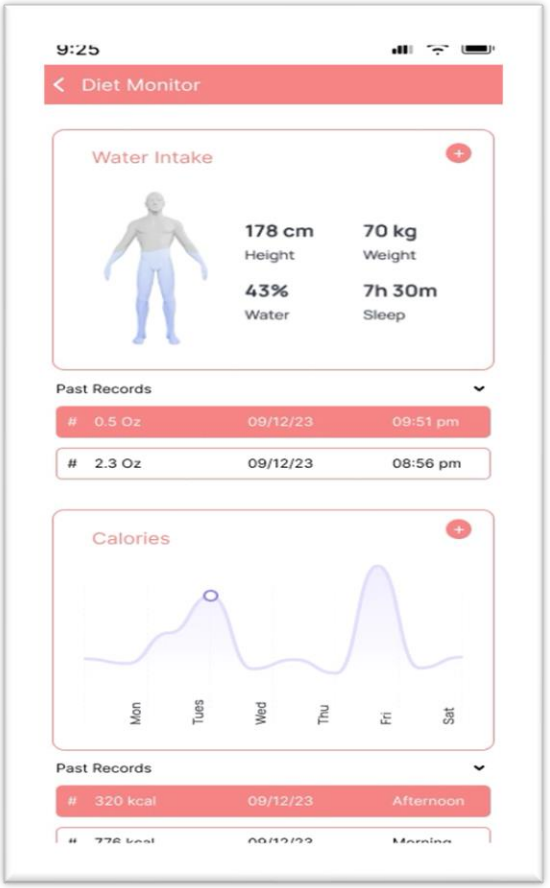
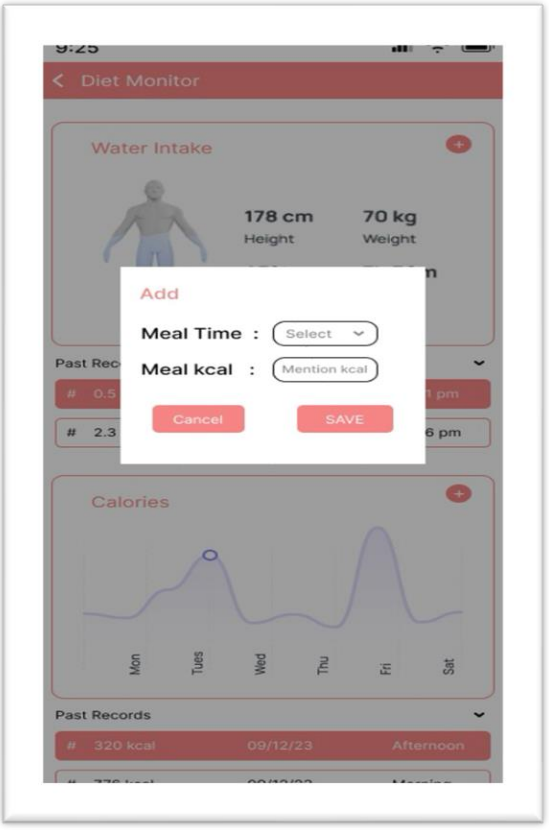
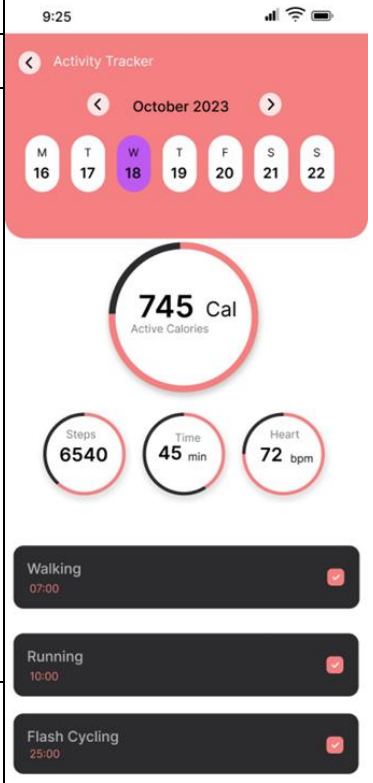


Image 6.2



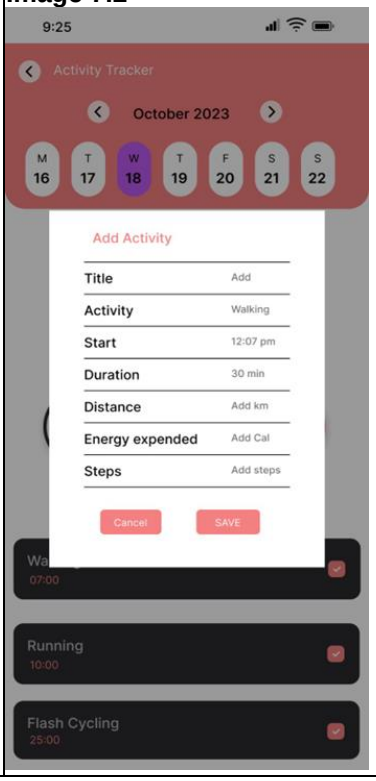
7) View/Update Physical activity:

<table>
<tr><td colspan="2">Precondition: This use case assumes that the user is updating their physical Event.</td><td rowspan="3">Image 7.1
 
</td></tr>
<tr><td>Actor: User</td><td>System: Health Monitor</td></tr>
<tr><td>1. TUCBW user viewing their physical Event and previous records, clicks on "add".

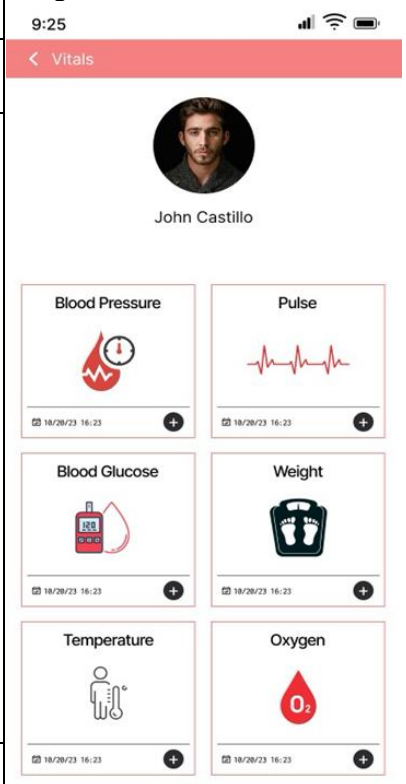
3. The user adds new event with information.

5. TUCEW user updating new record</td><td>0. System displays the Physical Event Screen UI. (Image 7.1)

2. The system displays add physical activity UI.

4. System displays the user updated details. (Image 7.2)</td></tr>
<tr><td colspan="2">Postcondition: The user can view updated physical Event.</td><td>Image 7.2
 
</td></tr>
</table>

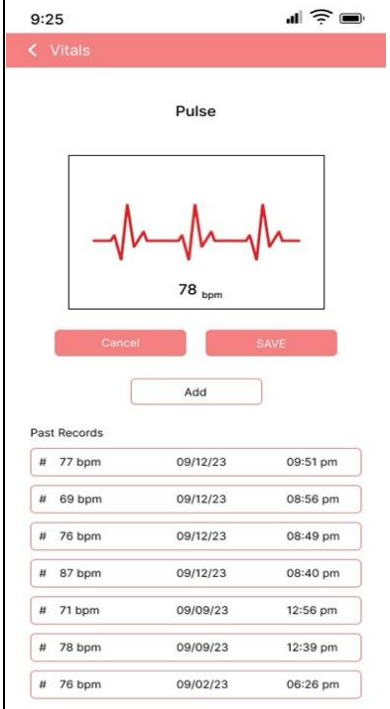
8) View/update Vitals:

<table>
<tr><td colspan="2">Precondition: This use case assumes that user is updating their vitals (blood pressure).</td><td rowspan="3">Image 8.1
 
</td></tr>
<tr><td>Actor: User</td><td>System: Health Monitor</td></tr>
<tr><td>1. TUCBW user viewing their vital and previous records, clicks on "add".

3. The user adds to vitals details

5. TUCEW user updating new record</td><td>0. System displays the pressure Screen UI.
(Image 8.1)

2. System displays Add vital UI

4. System displays the user updated details.
(image 8.2)</td></tr>
<tr><td colspan="2">Postcondition: The user can view their updated vitals.</td><td>Image 8.2
 
</td></tr>
</table>

9) Schedule Appointment:

<table>
<tr><td colspan="2">Precondition: This use case assumes that user wants to schedule an appointment</td></tr>
<tr><td>Actor: User</td><td>System: Health Monitor</td></tr>
<tr><td>1. TUCBW user viewing doctor information and clicks on the preferred doctor's name.

3. The user clicks on "Book Appointment" after choosing date and time

5. TUCEW user booking an appointment.</td><td>0. System displays Appointment Scheduler Screen UI. (image9.1)

2. System displays the Doctor Information UI. (image9.2)

4. System displays the booked appointment information. (image9.3)</td></tr>
<tr><td colspan="2">Postcondition: The user has a scheduled appointment with their preferred doctor.</td></tr>
</table>

Image9.1

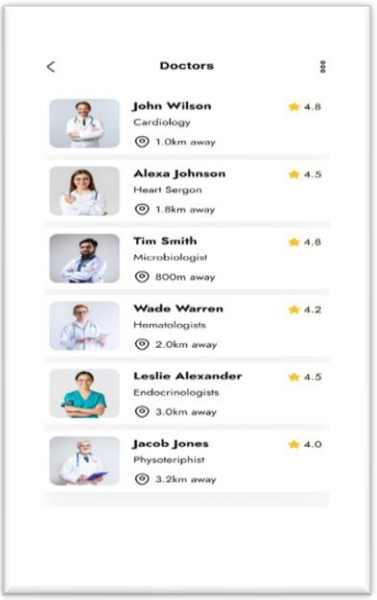


Image 9.2

Image9.3

10) SOS view/update:

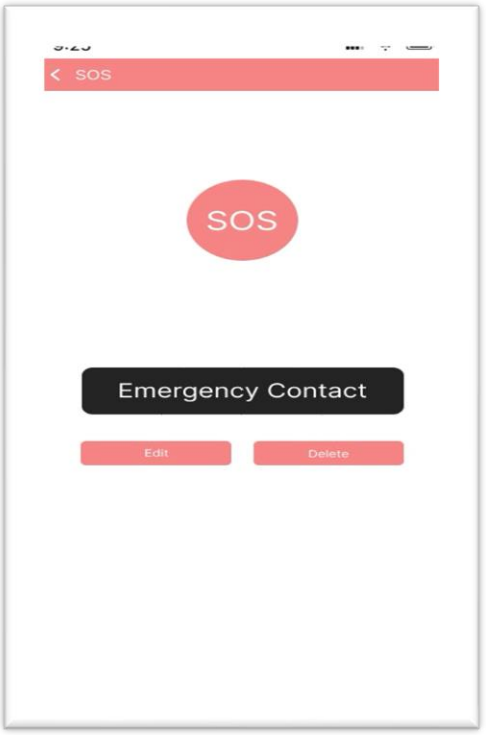
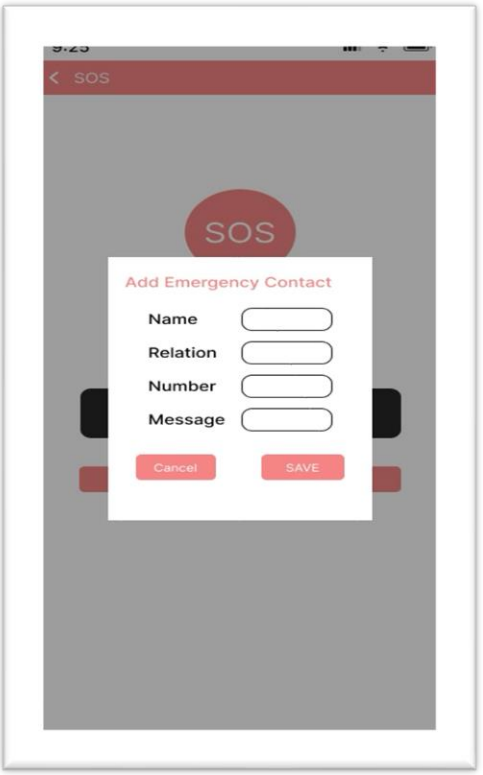
<table>
<tr><td colspan="2">Precondition: This use case assumes that user is Adding their Emergency contact information</td><td rowspan="4">Image 10.1
 

Image 10.2
 
</td></tr>
<tr><td>Actor: User</td><td>System: Health Monitor</td></tr>
<tr><td>1. TUCBW user updating their emergency contact details: Name, Relation, Number, Message.

3. The user clicks on the "SOS" button in case of emergency.

5. TUCEW user sending an alert.</td><td>0. System displays Emergency contact Screen UI. (Image 10.1)

*2. System displays user updated emergency contact details and saves it to the database displaying “Emergency contact saved” message. (Image 10.2)

4. System sends an alert to given contact.</td></tr>
<tr><td colspan="2">Postcondition: An alert is sent to the Emergency contact regarding the user.</td></tr>
</table>

11) View/update medicine Reminder:

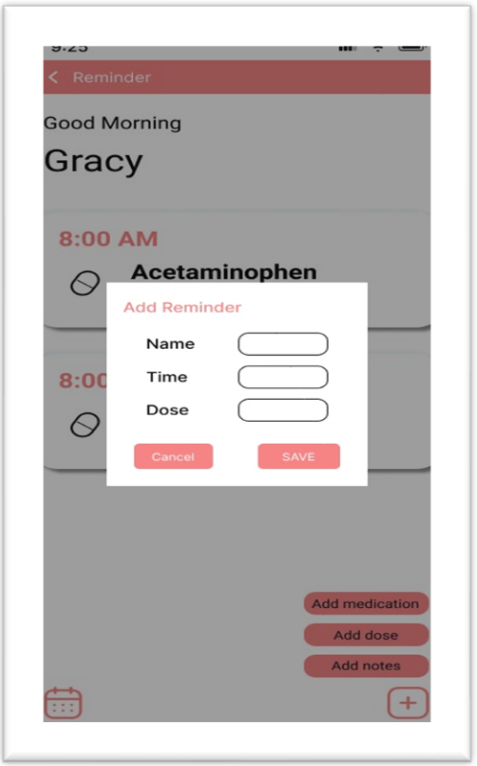
<table>
<tr><td colspan="2">Precondition: This use case assumes that user is creating a reminder for medicines</td></tr>
<tr><td>Actor: User</td><td>System: Health Monitor</td></tr>
<tr><td>1. TUCBW user creating a reminder by adding medicine details: Name, Time, Dose.

5. TUCEW user schedule a reminder for their medicine.</td><td>0. System displays Reminder Screen UI. (Image 11.1)

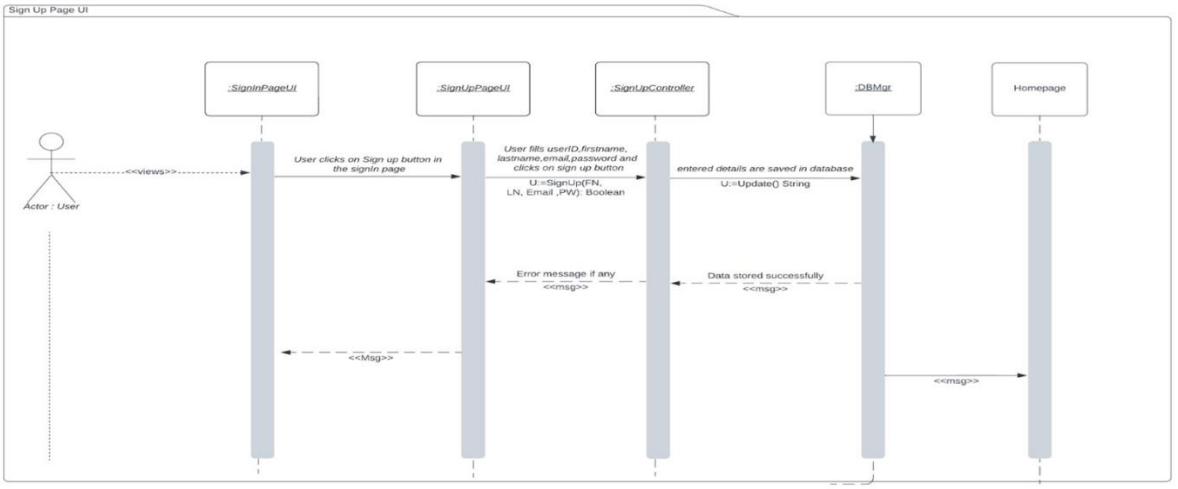
2. System creates a reminder with user given details. (Image 11.2)</td></tr>
<tr><td colspan="2">Postcondition: The user will be reminded about medication at a given time.</td></tr>
</table>

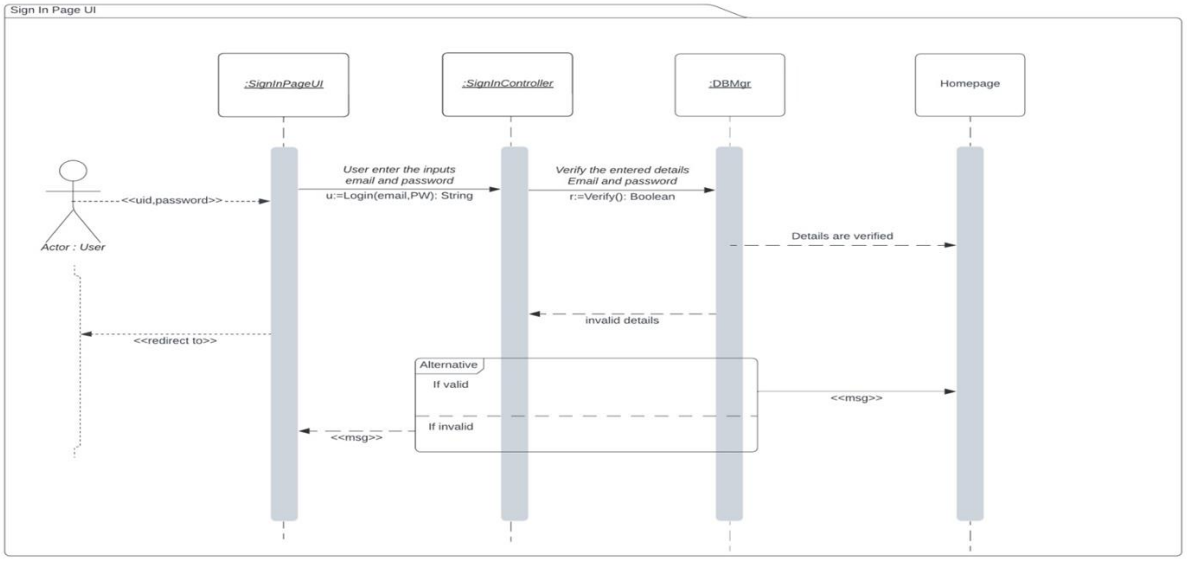
Image 11.1

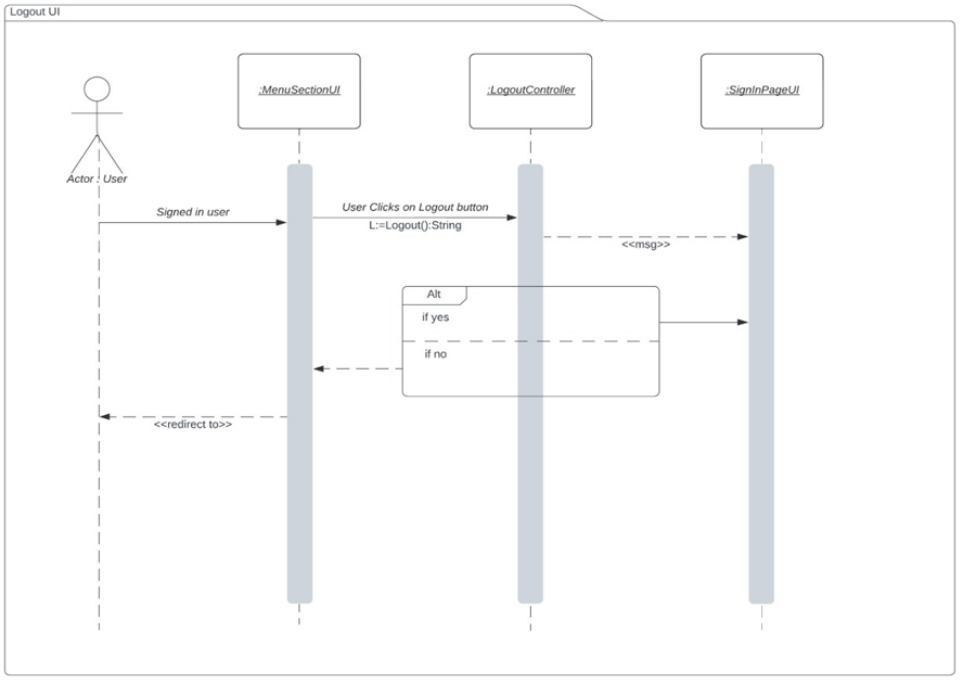
Image 11.2

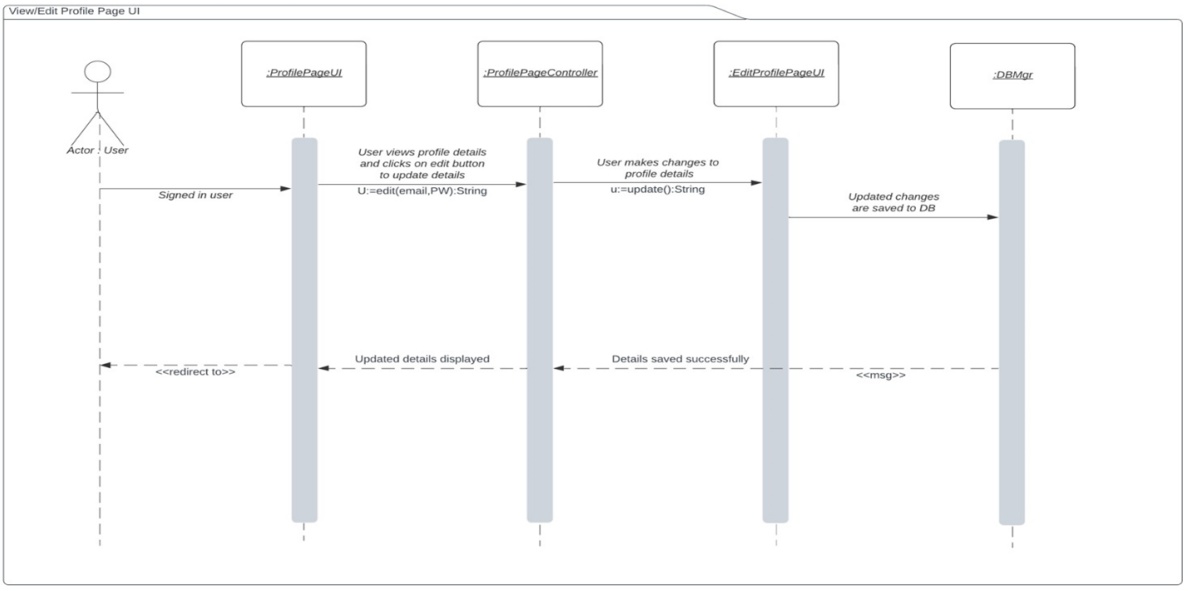


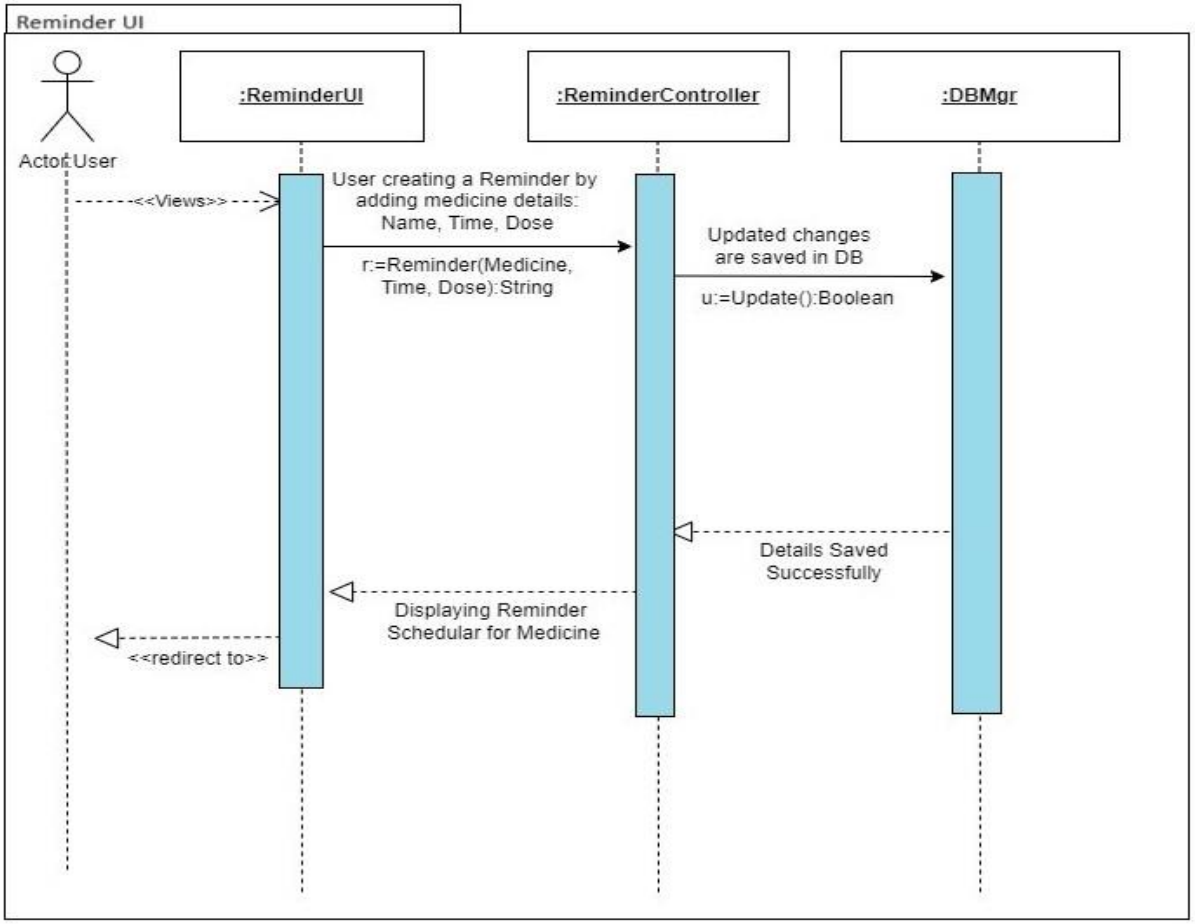
Sequence Diagram:

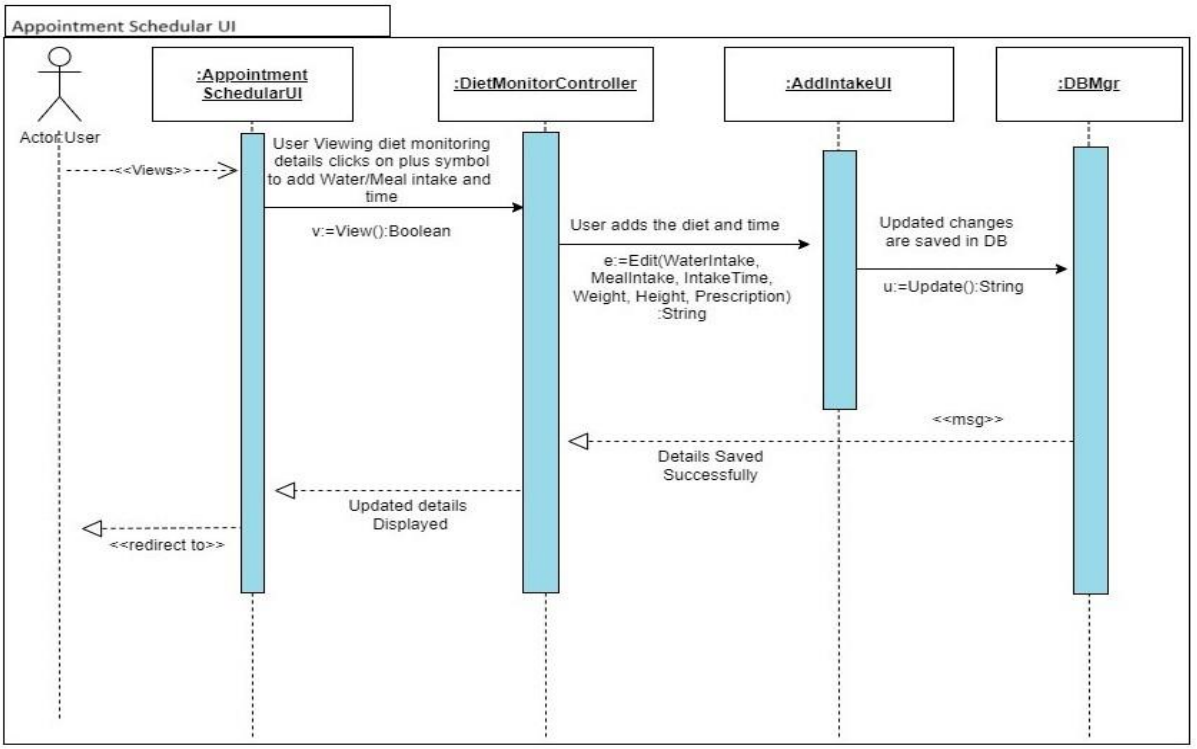


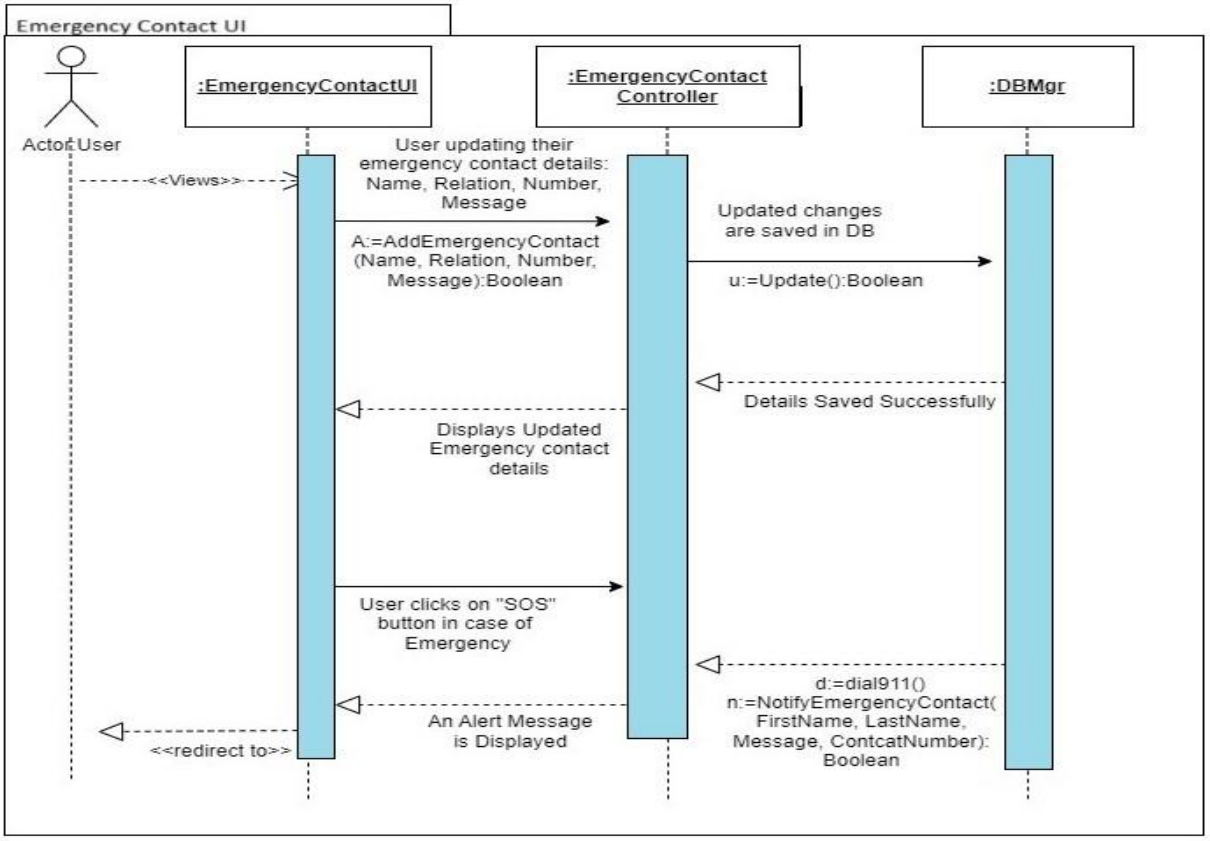


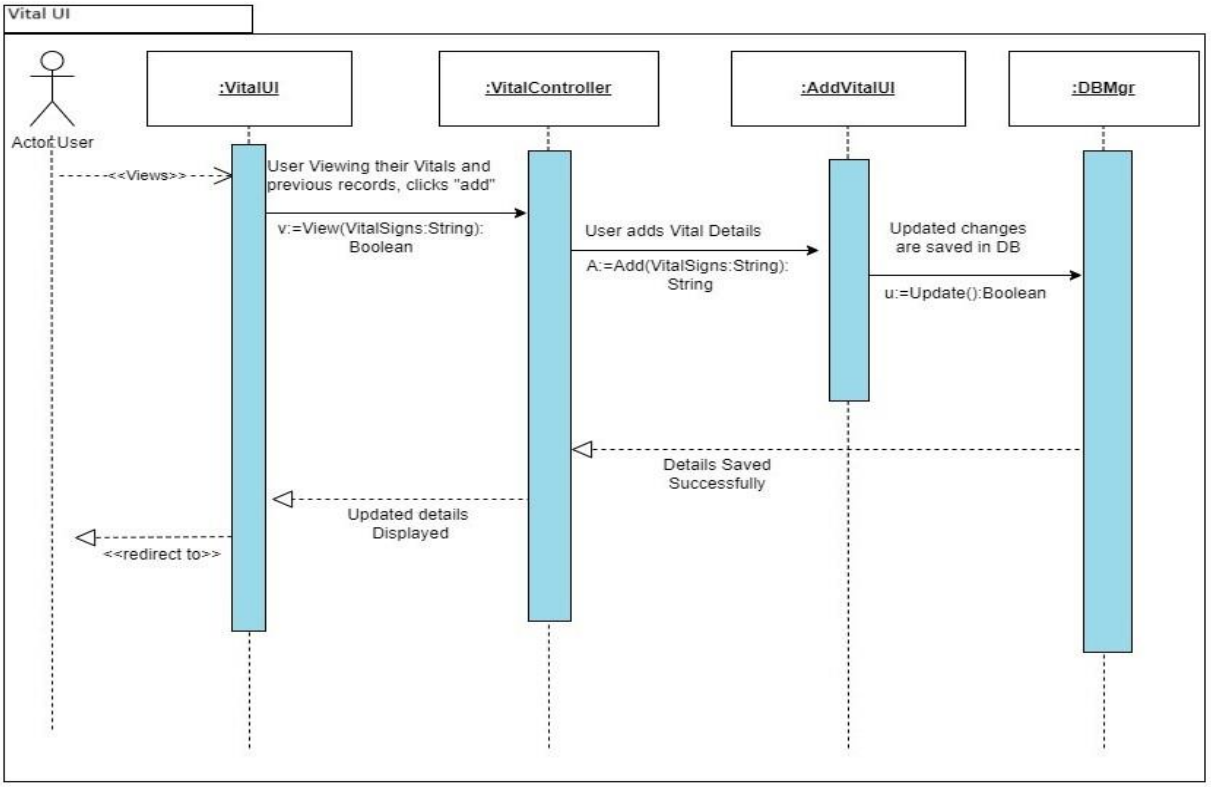

Logout UI
:MenuSectionUI
:LogoutController
:SignInPageUI
Actor : User
Signed in user
User Clicks on Logout button
L:=Logout():String
<<msg>>
Alt
if yes
if no
<<redirect to>>

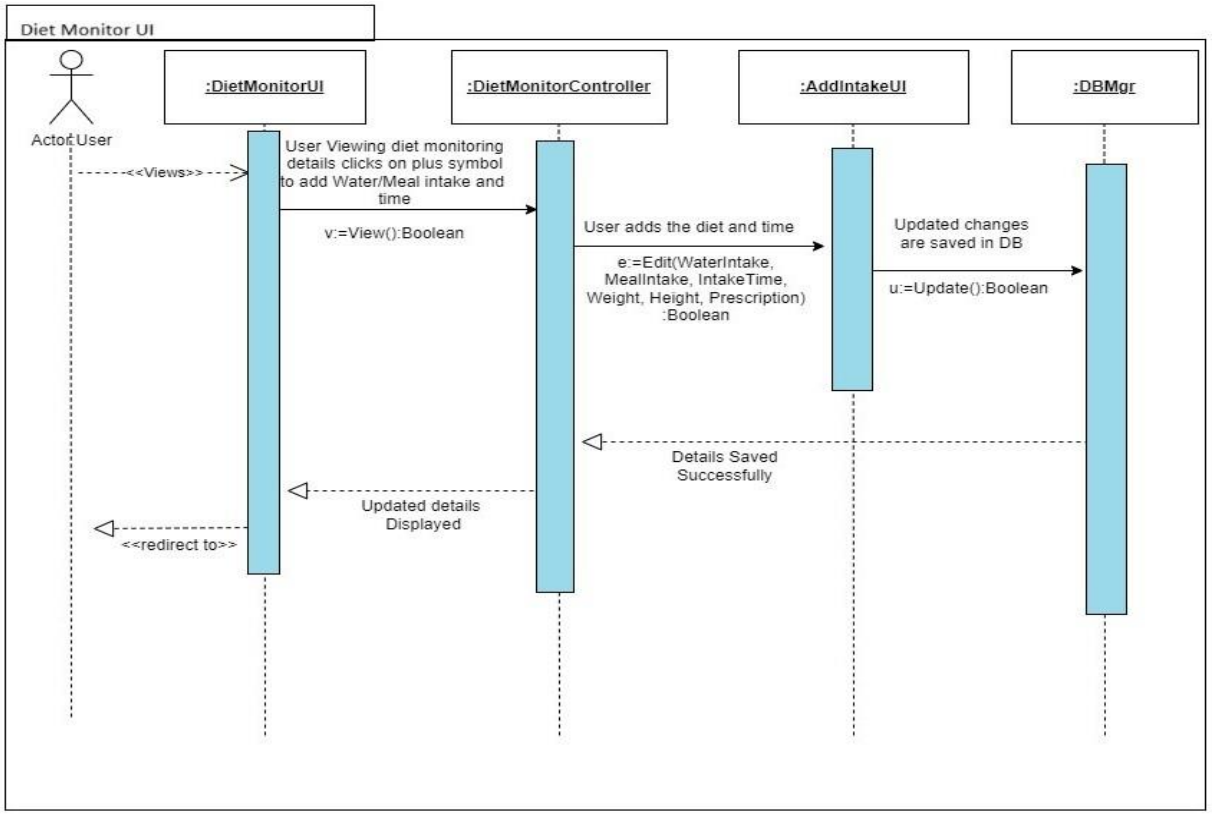

View/Edit Profile Page UI
:ProfilePageUI
:ProfilePageController
:EditProfilePageUI
:DBMgr
Actor : User
Signed in user
User views profile details and clicks on edit button to update details
U:=edit(email,PW):String
User makes changes to profile details
u:=update():String
Updated changes are saved to DB
Updated details displayed
<<redirect to>>
Details saved successfully
<<msg>>

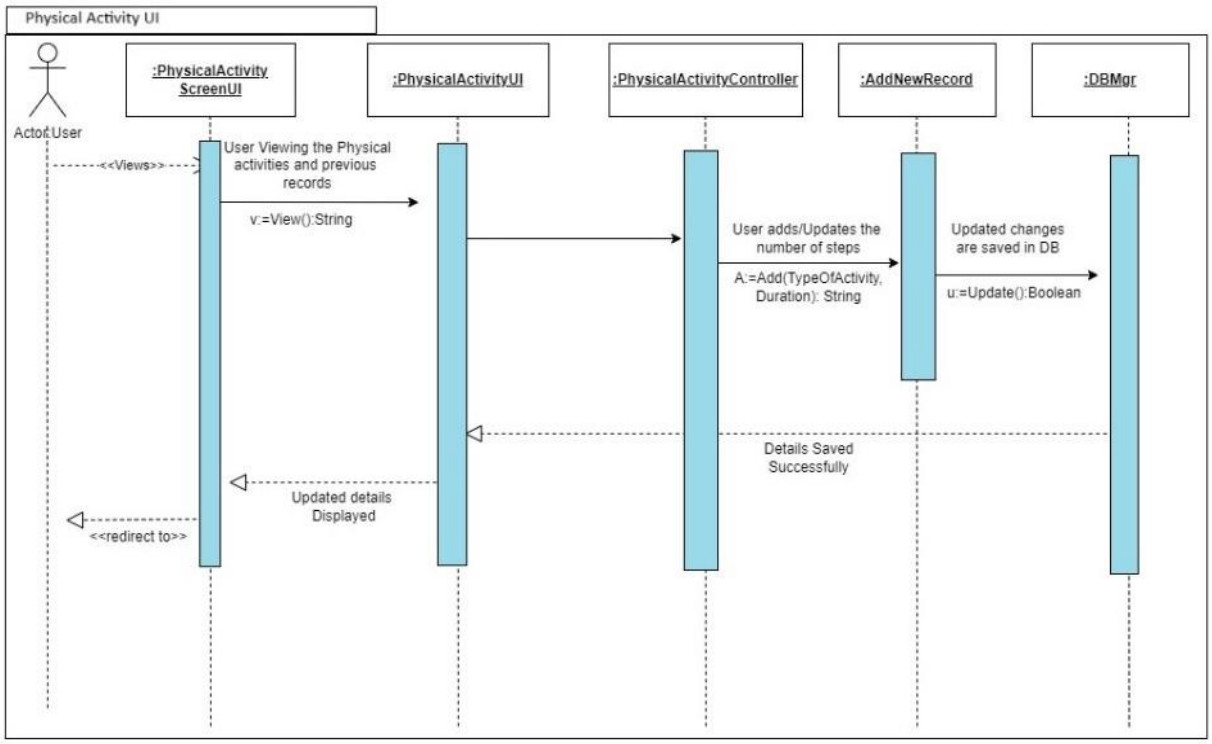

Reminder UI
:ReminderUI
:ReminderController
:DBMgr
Actor:User
<<Views>>
User creating a Reminder by adding medicine details: Name, Time, Dose
r:=Reminder(Medicine, Time, Dose):String
Updated changes are saved in DB
u:=Update():Boolean
Details Saved Successfully
Displaying Reminder Schedular for Medicine
<<redirect to>>


Appointment Schedular UI
:Appointment SchedularUI
:DietMonitorController
:AddIntakeUI
:DBMgr
Actor:User
<<Views>>
User Viewing diet monitoring details clicks on plus symbol to add Water/Meal intake and time
v:=View():Boolean
User adds the diet and time
e:=Edit(WaterIntake, MealIntake, IntakeTime, Weight, Height, Prescription) :String
Updated changes are saved in DB
u:=Update():String
<<msg>>
Details Saved Successfully
Updated details Displayed
<<redirect to>>

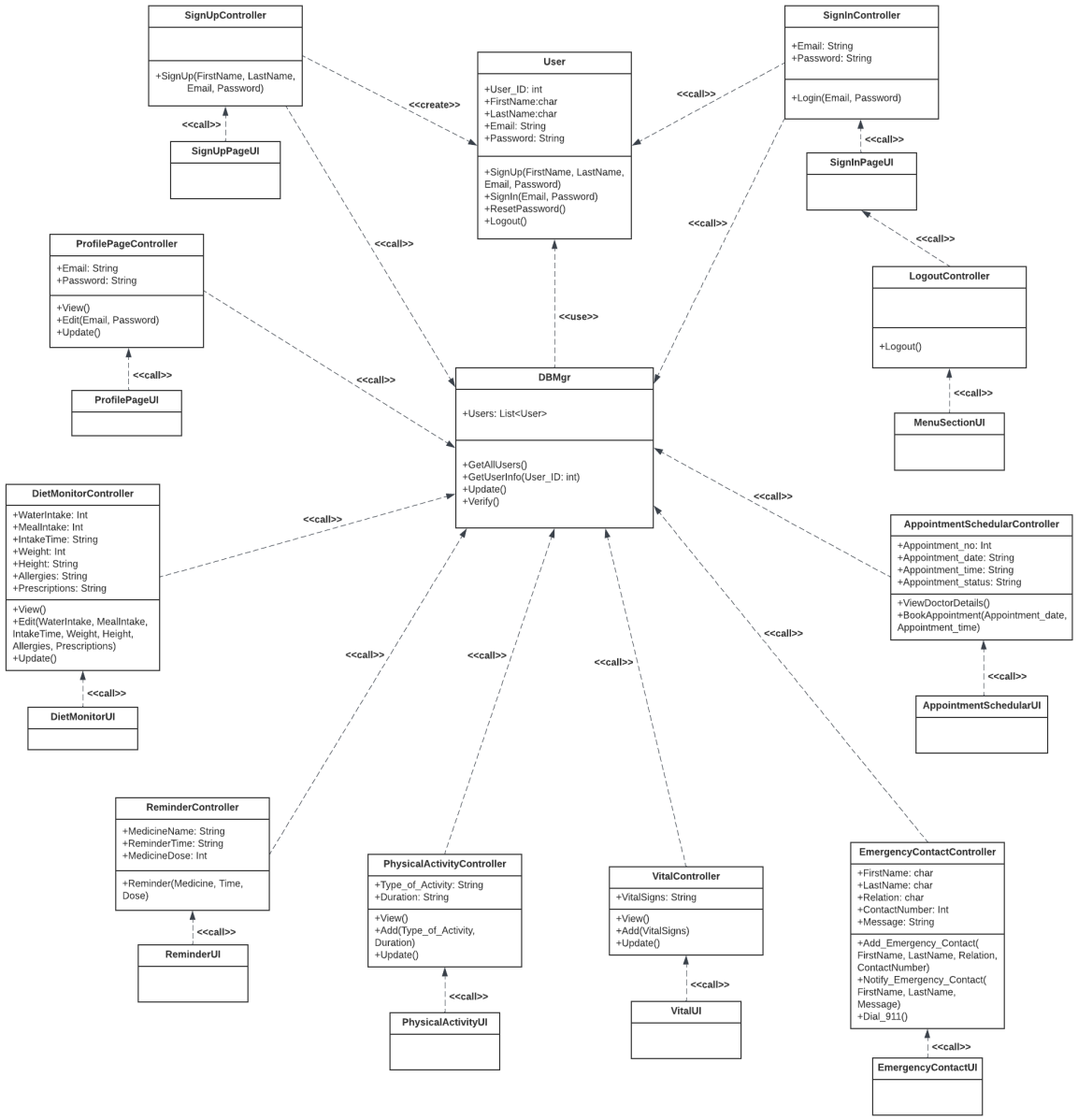

Emergency Contact UI
:EmergencyContactUI
:EmergencyContact Controller
:DBMgr
Actor:User
<<Views>>
User updating their emergency contact details: Name, Relation, Number, Message
A:=AddEmergencyContact (Name, Relation, Number, Message):Boolean
Updated changes are saved in DB
u:=Update():Boolean
Details Saved Successfully
Displays Updated Emergency contact details
User clicks on "SOS" button in case of Emergency
d:=dial911()
n:=NotifyEmergencyContact(FirstName, LastName, Message, ContcatNumber): Boolean
An Alert Message is Displayed
<<redirect to>>


Vital UI
:VitalUI
:VitalController
:AddVitalUI
:DBMgr
Actor:User
<<Views>>
User Viewing their Vitals and previous records, clicks "add"
v:=View(VitalSigns:String): Boolean
User adds Vital Details
A:=Add(VitalSigns:String): String
Updated changes are saved in DB
u:=Update():Boolean
Details Saved Successfully
Updated details Displayed
<<redirect to>>

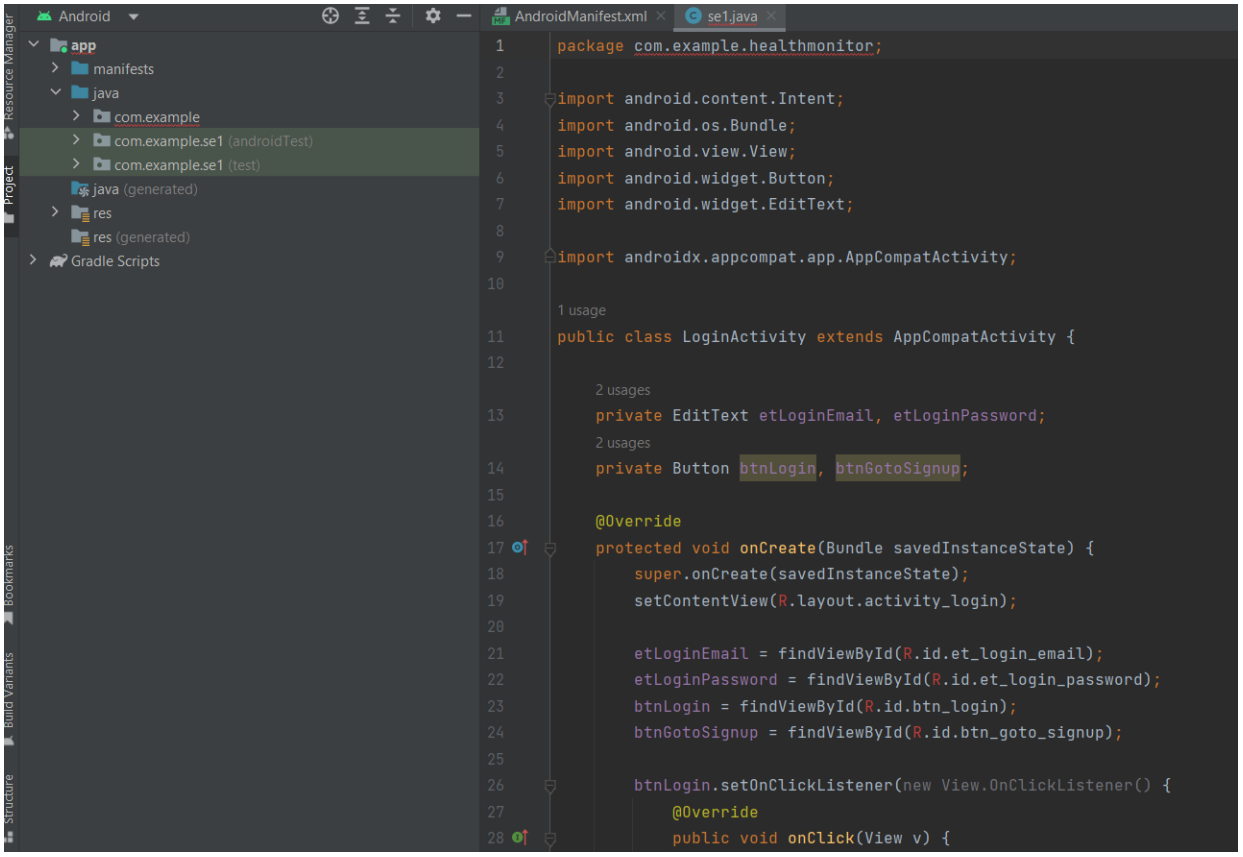

Diet Monitor UI
:DietMonitorUI
:DietMonitorController
:AddIntakeUI
:DBMgr
Actor:User
<<Views>>
User Viewing diet monitoring details clicks on plus symbol to add Water/Meal intake and time
v:=View():Boolean
User adds the diet and time
e:=Edit(WaterIntake, MealIntake, IntakeTime, Weight, Height, Prescription) :Boolean
Updated changes are saved in DB
u:=Update():Boolean
Details Saved Successfully
Updated details Displayed
<<redirect to>>

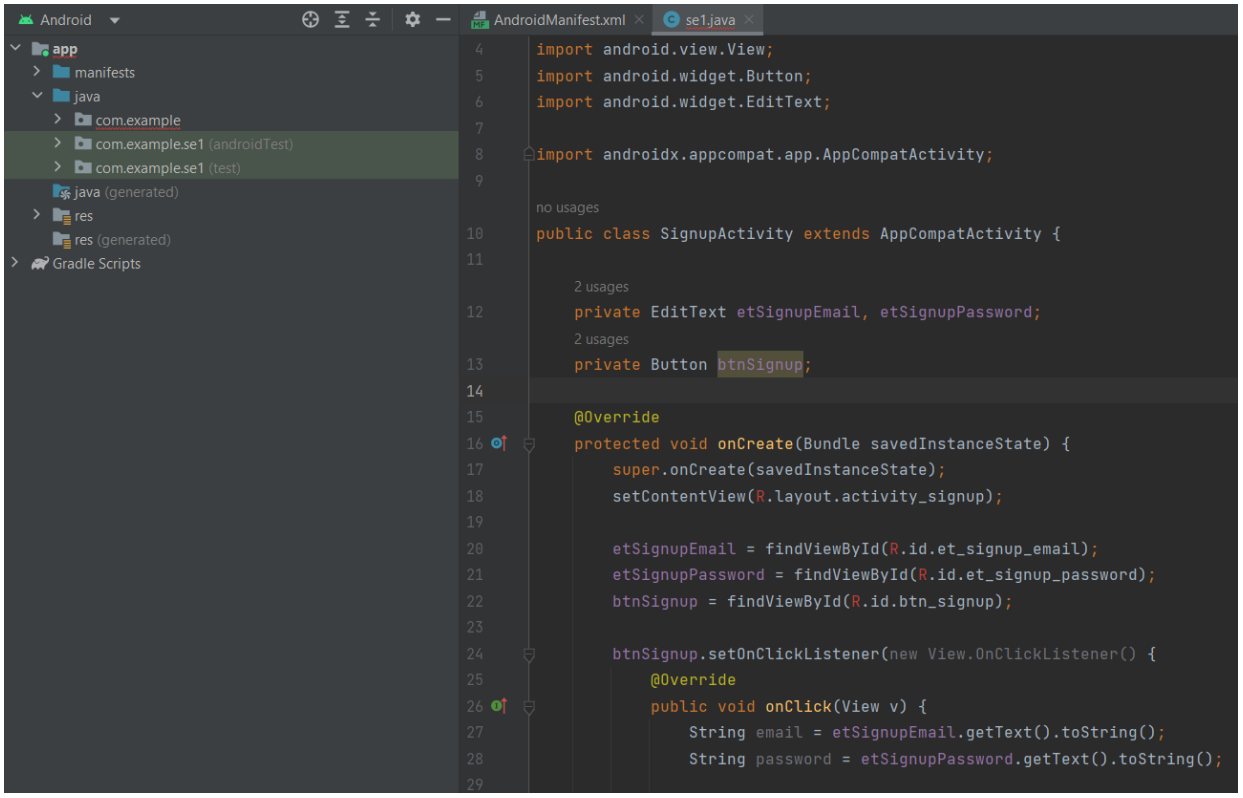

Physical Activity UI
:PhysicalActivity ScreenUI
:PhysicalActivityUI
:PhysicalActivityController
:AddNewRecord
:DBMgr
Actor:User
<<Views>>
User Viewing the Physical activities and previous records
v:=View():String
User adds/Updates the number of steps
A:=Add(TypeOfActivity, Duration): String
Updated changes are saved in DB
u:=Update():Boolean
Details Saved Successfully
Updated details Displayed
<<redirect to>>

Design Class Diagram:



Code Snippets:





YouTube Demo:

https://www.youtube.com/watch?v=_waXDRQVQVs&ab_channel=SHUBHAMBHATT