RU Healthy?

RU Healthy?

REPORT 1:

Class
ECE 567 – Software Engineering 1

Semester
Fall 2017

Group #2
Tahiya Chowdhury, Tina Drew,
George Koubbe, Aymen Al-Saadi,
Himabindu Paruchuri and Ramya Tadepalli

Website
RU Healthy?

Table of Contents

1 Effort Breakdown

Report	Aymen	George	Tahiya	Himabindu	Tina	Ramya
Customer Statement of work						
Problem Statement (5)	100%					
Glossary (4)						
System Requirements						
Functional Requirements (2)	100%					
Non-Functional Requirements (2)						
UI Requirements (2)						
Functional Requirements Specs						
Stakeholders Actors/Globals (2)				100%		
Use Case Casual Description (8)	33%	33%			33%	
Use Case Diagram (5)	33%	33%			33%	
Use Case Full Description (10)	33%	33%			33%	
System Sequence Diagram (5)				100%		
User interface						
Preliminary Design (11)			50%			50%
Effort Estimation (4)			50%			50%
Domain Analysis						
Concepts (10)	50%	50%				
Associations (5)			100%			
Attributions (5)						100%
Contracts (5)				100%		
Mathematical Model						
Plan of Work (5)					100%	
Project Management (10)	16%	16%	16%	16%	16%	16%
References						

2 Customer Statement of Work

2.1 Problem Statement

We live in a technology driven, quick paced, gluttonous society. We know that exercising has various benefits including: weight management, decreasing health conditions and diseases, improving mode, and boosting energy [2]. The issue is that we don't always receive these benefits because we don't regularly exercise. It is easy to get caught up in online profiles, games or social statuses and neglect physical exercise. It is also extremely convenient to purchase fast food from a local McDonald's rather than making a healthy home-cooked meal. Unfortunately, these habits lead to obesity and multitude of health issues. According to the 2016 Health Study by the United States Department of Health and CDC (Centers for Disease Control and Prevention), over 70% of Americans over the age of 20 are overweight [10]. Another study from the CDC showed that 80% of Americans don't get the recommended amount of exercise [3].

The similarities in these numbers between those that are overweight and those that do not get the recommended amount of exercise is not surprising. According to the Merriam Webster dictionary obesity is defined as: "a condition characterized by the excessive accumulation and storage of fat in the body" [6]. We know that body fat is essentially excess calories that have not been burned. Exercise provides a way to burn calories at a rapid rate. So the more the exercise, the more calories you burn; the more calories you burn, the more fat you lose.

According to the CDC, each adult should do a minimum of 150 minutes of moderate intensity activity or 75 minutes of vigorous activity each week [8]. Put simply, this amount is about 22 minutes of moderate active per day or 11 minutes of vigorous activity per day. A moderate activity is one that is able to raise your heart rate enough for you to sweat. Examples of moderate aerobic activity include [8]:

- Walking fast
- Doing water aerobics
- Riding a bike on level ground or with few hills
- Playing doubles tennis
- Pushing a lawn mower

Vigorous activity raises your heart rate moderately and involves heavy breathing. Some forms of vigorous activity include [8]:

- Jogging or running
- Swimming laps
- Riding a bike fast or on hills
- Playing singles tennis
- Playing basketball

Additionally, muscle strengthening activity is advised [8].

2.2 Current Market Solutions

As a way to promote activity and reduce obesity, various fitness devices and applications have been developed. Some of the fitness applications on the market include: Jawbone, Fitbit, Johnson and Johnson's 7 minute workout, FitStar, Lose It, Workout Trainer, JEFIT, Sworkit, and Strong Workout Tracker [7]. Most of the fitness applications offer activity tracking, calorie counting, and step tracking. Many of these apps, such as Fitbit, include web interfaces that allow you to join in with other workout groups. Some apps such as JEFIT and workout trainer, demonstrate how to do certain exercises [7]. Additional apps such as JEFIT also allow you to map workout routines and offer workout timers [7]. These advancements in technology have made is easier to track activity progress while also encouraging people to exercise by the positive cognitive reinforcements of social media groups.

Figure 2.2.1 - Applications of same category that are currently available in market [13]

2.3 RU Healthy? App Solution

The "RU Healthy?" app wants to continue this trend of using technology to motivate people to exercise. It will implement the basic fitness features such as activity tracking and step counting by using the motion sensors and camera of the smartphone. The app will also detect whether you are exercising according to your schedule. Furthermore, the app will store these statistics to an online database that both the physician and user can access.

No one enjoys the problems that come with being overweight and we know that exercise reduces body fat. So, why don't we exercise more? For many of us there are three main reasons that we don’t exercise: lack of accountability, lack of motivation, or scheduling inconsistencies. The RU Healthy? app addresses all these issues.

To address the issue of the lack of motivation, let’s consider the impact of accountability. Accountability motivates people to move forward. There is a friend who recently started a 30 day challenge to burn 1000 calories a day. The challenge was started because she wanted to buy a special shirt from her friend. The friend would not offer the shirt to her but stated that if she was able to complete the 1000 calories a day challenge, he would give it to her. He then opened up the challenge to other friends in their group in a fitness app. The one who does the best on the challenge gets the shirt and a $200 cash price. What is interesting about this group of people is that many of them are not very active and are not usually motivated to exercise. However, because they are part of the group and their progress is reported to the group in this fitness application, they are intentionally exercising. The group’s accountability motivated the non-active members to get up and move. The RU Healthy? app would essentially do the same thing except the user would be accountable to their physician.

Another reason that we don’t exercise consistently is because we have a very busy lifestyle. We never have the time and we simply “forget” or don’t put it in our schedule. The RU Healthy? app offers a possible solution for this issue. The app has a schedule reminder that sets off an alarm when it is time to exercise. The user can set up their own workout schedule, or be advised a schedule by a physician or personal trainer. This schedule will look different for each patient. For some, it may be 5 to 10 minutes of activity at certain points of the day. For others it may be a block of time setup specifically to work out. Once the schedule is set, it will sync with the phone calendar and timer to send an alert notifying the user that it is time to exercise. This alarm or alert gives the user a reminder to get up and move. The chances that we will forget to exercise are reduced. Additionally, because the schedule will be set up and agreed upon by the user, it should be at a pace that they can adhere to and follow.

From a physician's perspective, the RU Healthy? app makes it easier to hold their patient accountable to the exercise regiment. Think about it, everytime we go to the doctor they ask the question: are you exercising? The answers to this question often frustrates physicians because their patients are either not following their suggestions regarding exercise or not being honest about the exercise routine. What if the physician already had the answer to that question prior to the appointment? This would save time from the doctor trying to decipher the patient’s honesty and commitment to the exercise. Furthermore, this helps the patient to develop an action plan, a technique that increases the probability that the person will exercise and meet the goal [9]. In short this makes it easier for patients to be accountable to their physicians. This is one of the concepts that makes our app unique. Other fitness applications do not promote allowing physicians to access the patient's activity results.

Additionally, the doctor would have access to the patient’s basic health statistics like heart rate and analysis of “recovery rate”, which is a useful tool to track a patient’s progress. For the patients who are at risk this information could be extremely useful because it could make the physician aware of anomalies such as irregular heart rates. Since an irregular heart rate is often an indicator of other health issues, having this information could help physicians diagnose problems in early stages. The physician could also examine the heart rate levels of patients and detect possible issues such as thyroid problems or heart palpitations.

Copyright 2003 by Randy Glasbergen.
www.glasbergen.com
GLASBERGEN
“What fits your busy schedule better, exercising one hour a day or being dead 24 hours a day?”

Figure 2.3.1 - A Short comic from Glasbergen.com [18]

2.4 How will ‘RU Healthy?’ accomplish this?

Let’s say the patient, Mike, visits a physician and creates a user account, logs into the app and links it with the doctor’s account. Then, Mike will have to do an initial one-time configuration to enter his age, gender, height, weight. With the app, he will have the ability to check: when he started and stopped walking, and the total distance that he walked. Moreover, he will be able to see and keep track of the calories he burnt during the week and his activity history, along with a heart beat rate history.

At the other end, the physician will be able to see Mike’s activities in a web UI. The web UI will provide him/her with all the necessary information about the patient status. It will also contain the patient’s personal data such as:

- Name
- Age
- Gender
- Weight
- Height
- Activity history
- Heart rate history and average heart rate
- Number of the steps he/she walked
- How many miles he/she walked

- How many calories he/she burnt during his activity

The physician will have the ability to control all the patients from the web UI, and review their information. The data of this UI will be obtained from a database which regularly stores user activity (say, weekly). We can use Django or a similar platform to make the web UI for the doctor. Data from the Android smartphone can be sent to a server/database with the help of Java or Android libraries or this can be synced to the cloud. We can further scale this up to ensure that the physician is able to see all of his patient's records, if they have installed the app.

Every day, the person’s data will be stored. The weekly update will be sent to his/her physician. This way, the physician will get to know whether Mike is following his schedule or not. The doctor will have a web page with login information, and when he/she logs in, he/she can see the information updates coming from all their patients. And whichever patient is failing to follow the schedule, it will be mark by red and the device will give the patient a “warning” or a notification. The physician will set up an appointment with those patients to talk about it.

2.4.1 How does the proposed system count the steps?

The user just needs to carry his smartphone with him/her or leave it in their pocket. It will detect the unusual movement of the user and try to calculate the acceleration, based on three variables (x, y and z). The accelerometer sensor is found in every Android device (old and new). It is a device that measures the acceleration (or rate of change of velocity) of a body in its own instantaneous rest frame. Single and multi-axis models of accelerometer are available in Android devices to detect magnitude and direction of the proper acceleration, as a vector quantity, and can be used to sense orientation (because direction of weight changes), coordinate acceleration, vibration, shock, and falling in a resistive medium. That is one of the features that will give the advantages for the app , so it will not need any extra costs for the user to use this feature.

Our Footsteps Detector will collect the data from the sensors (x, y and z), velocity and device acceleration which is measured by m/s^2. After collecting these data we will process them and store the data in arrays. These arrays will try to help us to filter the data and normalize them based on normalization algorithms. These algorithms have a specific limit (which we will call sensor sensitivity) to detect the abnormal data that is coming from the sensors and store them in single dimension array or vector. We can retrieve this data, calculate the amount abnormal data obtained, and view it as a step.

Below is a specific description of the classes that we used for the Footsteps Detector:

Application Classes

- Step Listener Class Interface: This class-interface will listen to number of alerts about steps being detected.

- Step Detector Class: A class which will accept updates from accelerometer sensor and deploy the filter to detect if a step has been covered by the user.

- Sensor Filter Class: A class with an algorithm to filter out values that have a close approximation to steps sensitivity.

- Main Activity Class: This class will contain all the buttons on click activities and all the data returns from the main classes.

Application Functions

- On click Listener: This function will activate the Sensor Manager Listening to the abnormal acceleration from the Step Listener Class.

- On Sensor Changed: This function will update the sensor values from the Step Detector Class.

2.4.3 How does the proposed system measure heart rate?

For now, we would like to provide a general idea about how we intend to measure the heartbeat rate. We plan on using the smartphone’s flash and camera. All the user has to do is open the app and hold his/her index finger over the camera lens to start measuring. The camera is used to track color changes on the fingertip that are directly linked to the pulse. This is the same technique that medical pulse oximeters use. The whole process should not take more than 30 seconds.

2.5 Glossary of Terms

Table 2.5.1: Glossary

Term	Description
Accelerometer	Handles the axis-based motion sensing and is used to count the step taken or distance walked.
Account	Entity in database that contains specific information about the user including profile details, body measurements, progress, etc.
Alarms	User can set his personal alarms for exercise.
BPM	Beats Per Minute is the number of times the heart beats in one minute.
Calorie Counter	It will track the number of calories the user burnt based on the distance covered.
Database	Place to store information. Stores the data of the user as a structured set.
History	Shows the previous data of the calories burnt/steps covered and if the user is working as per schedule.
Messages	Users can check some links of various exercise available in this option.
Profile	This includes the user's name, age, height, weight, profile picture(optional), email, gender etc.
Register	User can give username and password and register to use the mobile application.
Scheduler	Feature in the app to display the schedule of exercises for each month for ease of user.
Settings	User can change the profile settings like steps/calories target, distance measurement units, etc.
Smartphone	Portable personal computer with a mobile operating system containing features useful for handheld use.

Step Tracker	A feature in app that tracks the number of steps completed and steps left to reach target.
Web Application	Web interface which the physician can use to access the patient data.

3 System Requirements

3.1 Functional Requirements

A requirement is defined as "a condition or capability to which a system must conform" [4]. A system may have a variety of requirements and these are often categorized to ensure better focus on each.

Functional requirements may be calculations, technical details, data manipulation, data processing and other specific functionality that define *what* a system is supposed to accomplish. These type of requirements are expressed in the form "system shall do <requirement>" for all the features that must be implemented, whereas features that are an optional addition are expressed in the form "system should do <requirement>". The priority weight assignation is based upon the customer's essential requirements that must be satisfied. A higher number indicates that a requirement is more crucial for the project.

Table 3.1.1: Enumerated functional requirements for 'RU Healthy?'

ID	Priority Weight	Requirements
REQ-1	5	The system shall be able to obtain and process the information from the Android device sensors.
REQ-1a	5	The system shall be able to obtain motion sensor data and count how many steps the user walked during his/her last activity.
REQ-1b	5	The system shall be able to obtain the data from camera and compute BPM.

REQ-2	4	The system shall be able to store the data collected from the sensors in an offline database.
REQ-3	4	The system shall be able to compute how many miles the user walked, how fast their paces were, and the duration of travel.
REQ-4	4	The system shall be able to compute how many calories the user burnt based on his/her traveled distance and speed.
REQ-5	4	The system shall allow the user to add/edit/remove accounts.
REQ-6	2	The system should be able to give the user alerts and notifications.
REQ-6a	2	The system should be able to send notifications about current progress.
REQ-6b	2	The system should send alerts when it is time for the user to exercise.
REQ-7	3	The system should be able to communicate to an online database.
REQ-7a	3	The system should be able to do online backups.
REQ-7b	3	The system should be able to do online restores.
REQ-8	5	The system shall be able to send a weekly activity summary to the physician.
REQ-9	4	The system shall be able to deliver the user all the important notes that the physician wants the user to follow.
REQ-10	2	The system should allow a workout schedule to be set by the physician in conjunction with the user.
REQ-11	1	The system should allow the user to manually turn off the alerts.
REQ-12	4	The system shall allow users to start and stop monitoring information upon the user’s request.

The first thing to do in RU Healthy? is to create a user profile (REQ-5). However, when it is time to link it with a doctor, how will the patient do that? Does he/she know a doctor? Does he/she prefer someone in particular? We thought about it, and we have come up with the following policy:

RUH-BP01: A prior connection between the patient/physician will need to be established through a physical visit that will allow the initial configuration of the app. That is, setting up initial information and linking the user's profile to the doctor's account.

We also analyzed the option in which the patient does not have a physician, and he/she could select the area and choose a physician of preference, or even the system could get the patient's current location and match him with a physician nearby. But what if after the blind selection, the user actually gets to know his/her physician and does not like him/her?. The user would have given the doctor records and private information that could expose him/her. So for now, we will be staying with our current business policy.

Once the physician selection through physical visit is complete, the primordial function of the system will be to obtain and process information from the camera and motion sensor every time the patient exercises. In order to satisfy this, we have separated REQ-1 into two different processes: activity and heart rate, that is REQ-1a and REQ-1b. Without this, the system has no point and no future development. The patient can turn this functionality on and off at will, so we need to introduce REQ-12.

Because the software will only use the internal sensors of the Android device, there is no way of continuously monitoring the heart rate. Therefore, we need to introduce our second policy:

RUH-BP02: The heart monitoring won't be a continuous feature. Instead, the patient will need to measure it manually every time he/she needs to.

The system will also provide additional metrics about the exercise, and we need REQ-3 and REQ-4 for this.

An optional feature that the system could have is to provide the user with updates on his/her workout every now and then while he/she is exercising. This is accomplished with REQ-6. Also, it sounds like a good idea to alert the user when is time to exercise, so he/she does not forget, as well as allow the same notification functionality to be turned off at any time. REQ-6 and REQ-11 go hand to hand.

This workout schedule should not be something to be set by the physician only. It should have feedback from the user regarding exercises, duration, days of the week, etc. For this, we introduce REQ-10.

In order for the system to be successful, a communication between patient/physician is crucial. The physician needs to be able to monitor the patient's activity, so he/she can recommend or give him/her further indications. For this, we have REQ-8 and REQ-9. But what if the patient, besides sending the data to the physician, wants to be able to backup everything online? Imagine he/she gets robbed, or ends up losing their smartphone? We have come up with REQ-7, which makes sure the patient feels safe in this matter.

One more thing. Because the system will have no administrators, nobody will control the creation or removal of user profiles. The physician needs to be able to access the user profile for reviewing purposes. The following needs to be established:

RUH-BP03: the Physician can terminate his participation at any point if he feels that the Patient does not need it anymore. The Patient can choose if he/she wants to keep using the app.

3.2 Non-Functional Requirements

Non-functional requirements are based on the environment and quality of the system-to-be. The acronym 'FURPS+' represents a widely used model that classifies software quality attributes. This set of attributes includes: Functionality, Usability, Reliability, Performance and Supportability. The '+' symbol denotes the requirements to include design constraints, implementation, physical and interface requirements [4, 5].

The attributes related to 'Functionality' of our proposed system is enlisted in Section 3.1. Based on our study on existing applications, previous related works and informal survey on target customer population, we specify the non-functional requirements for our proposed system 'RU Healthy?' in Table 3.

Table 3.2.1: Enumerated non-functional requirements for 'RU Healthy?'

ID	Priority Weight	Requirements
REQ-13	5	All information from sensors will be stored in the system database.

REQ-14	4	The interface of the mobile app will be easy to navigate. The patient shall be able to change between menus with minimal effort even when working out.
REQ-15	3	The system response shall be prompt to user's command. Patient shall not feel delay between action (click) and reaction (system response).
REQ-16	2	The app should not crash if running for a long period of time.
REQ-17	3	Both the mobile and web application shall be intuitive; so that general experience with Android app will suffice to use the basic features effectively.
REQ-18	3	The app (mobile and web) should provide a help document to guide the patient and physician into initial registry of the system.
REQ-19	2	The app should disclose the degree of access it will have to patient's phone and his information before actually starting to use it.
REQ-20	3	The patient information stored in the database of the application shall have a high-level security. Each patient will have access to his/her information only.
REQ-21	4	The physician will be required to protect the privacy of the patient. This means he/she will not disclose login information to his account to web application with others.
REQ-22	5	The system will prohibit manipulation of work-out data by the patient or the physician. All data stored in the database and presented by app will be collected from mobile sensors.
REQ-23	3	The patient should be able to stop the alarm/reminder from the app by starting his scheduled work-out on that time.
REQ-24	2	The app will stop running after sensing inactivity of the patient for a certain period to save battery life.

Usability Requirements

The concept of 'RU Healthy?' focus on providing a user-centered design and a satisfying user experience. The user interface, with its easy to navigate features and prompt response to command, will mark consistency for the user experience (REQ-14, REQ-15). The proposed requirement for an intuitive design on interface will ensure that anyone with experience of using an Android app will be able to accomplish their goal (REQ-17). Alongside that, the help document will be able to help a novice user (REQ-18).

Reliability Requirements

The app, with the amount of data processed during work-out, will be required to elicit adequate reliability. While the app is running for a long period of time, it should not crash with the data for the session being lost (REQ-16). Also, with all data obtained from the sensors and not from manual input, the integrity of the data will be ensured (REQ-22).

Performance Requirements

The performance of the system should be efficient, quick and accurate. The app thus needs to provide quick reaction to user command and operate as such (REQ-15). Once the user start working out, it should provide explicit feedback for the user to confirm response and availability (REQ-23).

Supportability Requirements

All information from the sensors will need to be stored for processing and analyzing of data. The processed data will be used for fathoming the progress of the user for a period of time (REQ-13). In addition to that, the resource usage by the app during idle time should be restricted (REQ-24).

Design and Implementation Restrictions

This portion of requirements sets the dimension for policies of database integrity, resource limit, etc. As a large amount of private information about the user will be available in the database, each user should only have access to their own information only (REQ-20). The physician is restricted to maintain confidentiality of the information regarding his/her patient as part of his professional confidentiality agreement (REQ-21. The integrity of available data in the database should be maintained by avoiding direct manipulation from any of the users (REQ-22)

3.3 On-Screen Appearance Requirements

The section below show hand-drawn sketches that demonstrate the user interaction.

3.3.1. Mobile Application

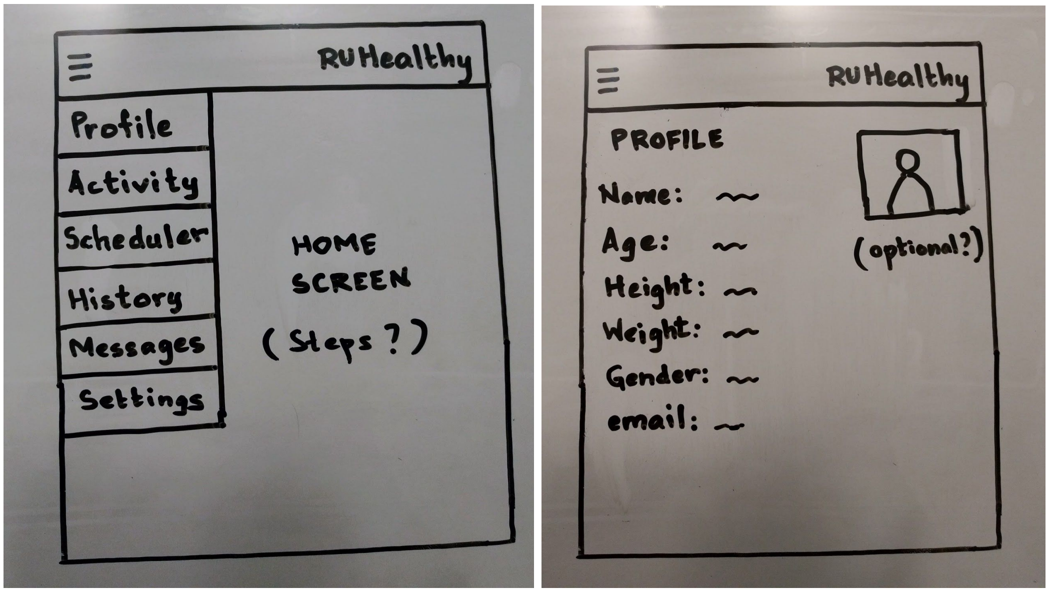


Fig 3.3.1.1: Options and User Profile

Figure 2 shows the available options on the mobile application. The user will be able to register and login. User’s details will be stored to a database. Once the user logs into their account, they will be able to add details to their profile. This details include: age, height, weight, and gender and will be used to set a reasonable target for their activity. They can also provide their email for further communication.

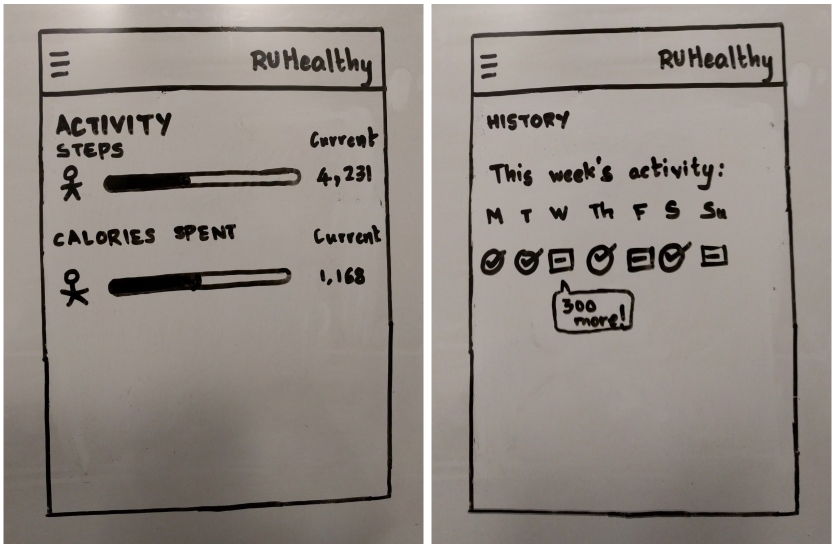


Fig 3.3.1.2: User Activity

Figure 3 shows how the User Activity will appear. The Users will be able to see their current progress for the day as a number and also as a graphic in comparison to their target. This includes the number of steps taken in miles or steps, the number of calories burnt (which will be calculated based on the profile details provided). Users will also be able to view a history of their activity as a timeline view, which highlights whether or not they have reached their target for each day.

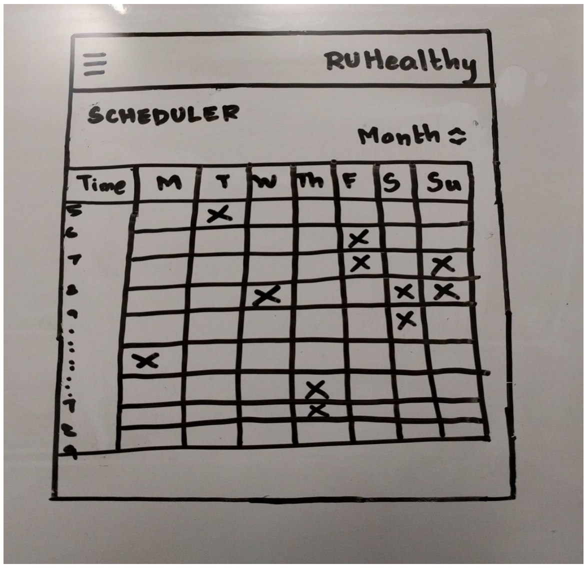
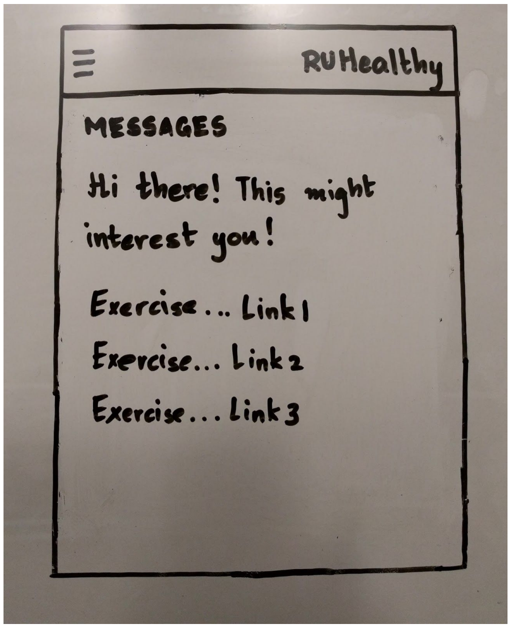


Fig 3.3.1.3: Scheduler

Figure 4 shows the Scheduler option. The Users will be able to mark some time of the day for exercise. The app then detects whether or not the User is performing any activity at that specified time. Since, we assume that the phone is on the user for the app to fully function, if the app detects the user being idle or lack of activity, it will remind the user that it’s time for exercise.



Hi there! This might interest you!

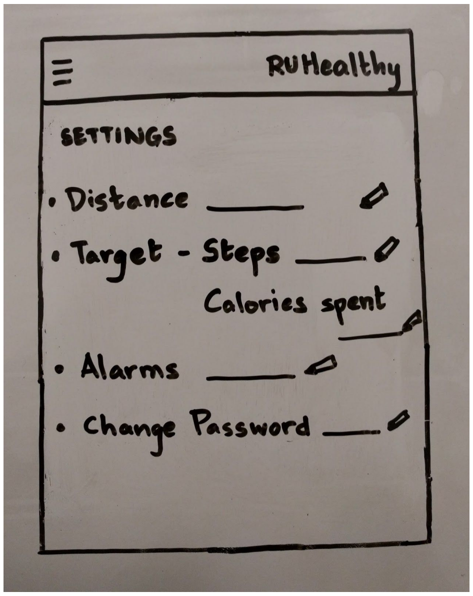
Exercise... Link1

Exercise... Link2

Exercise... Link3

Fig 3.3.1.4: Messages

Figure 5 shows the messages option for the app which could include messages that have been auto generated based on the activity the user performs or it could be messages sent to the user by their physician/personal trainer. This may include articles or links to some related exercise options or those advised for that particular user.



- Distance ____
- Target - Steps ____
  Calories spent ____
- Alarms ____
- Change Password ____

Fig 3.3.1.6: Settings

Figure 6 shows the Settings page of the app users may set their targets, reminder options etc.

3.3.2. Web Application

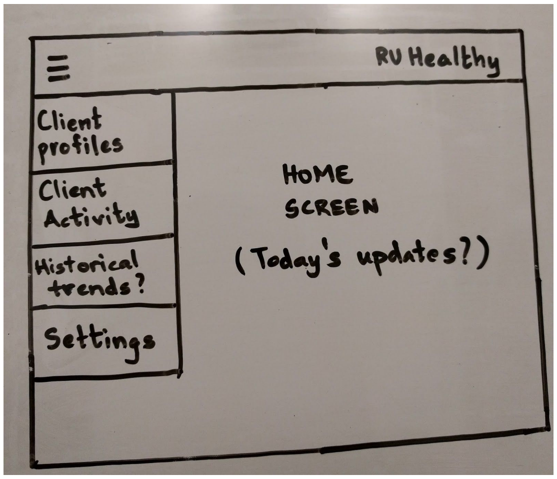


Fig 3.3.2.1: Options

We have also proposed a Web based application which will help the Users' physicians or personal trainers view their activity status. Let us call them admins. The admin will be able to register and login to the Web application. They can add or delete Client profiles i.e., details of their clients like, Name, age, gender, height, weight, BMI and possibly their health record. They will also be able to view Clients' activities. We envision that this information will help in better diagnosis and treatment and/or help clients lose weight.

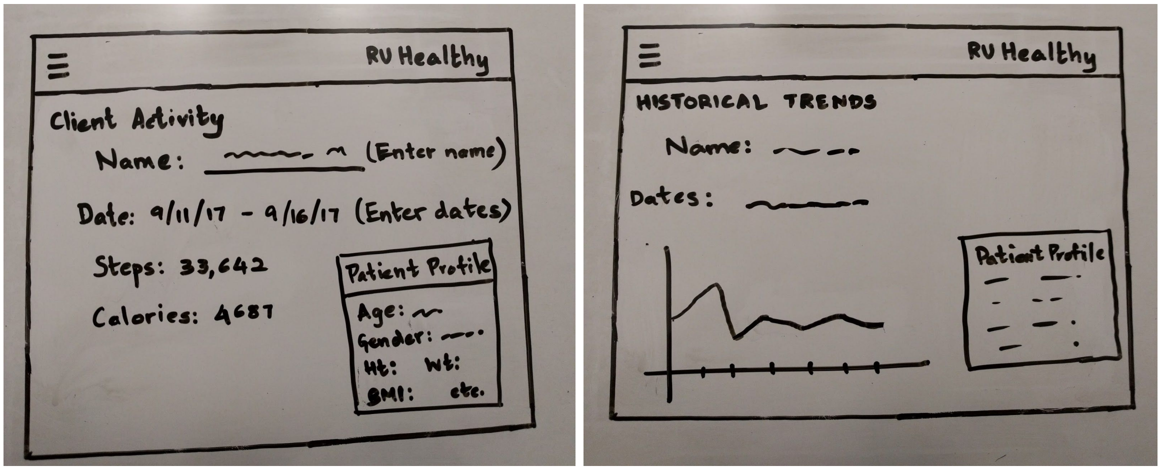


Fig 3.3.2.2: Client activity

Figure 8 shows how the admin will be able to view Client activity. This includes 2 pages, one to view activity and one to view trends in their activity. We may merge this into one page if that seems like a better and more user friendly view of the client's' activity. The admin is allowed to choose a specific set of dates and the User activity for that period is displayed on screen.

4 Functional Requirements Specification

4.1 Stakeholders

Stakeholders are those with any interest in project's outcome. Stakeholders are people who are invested in the project and who will be affected by your project at any point along the way, and their input can directly impact the outcome. Accordingly, the stakeholders in the project
RU Healthy? would include the following categories of people:

End Users or Customers: This project targets the people who aim to lose weight through exercising regularly and need some sort of push to achieve their goals. They can use this system to keep track of their health conditions and get easy way to keep healthy. So, the customers/end users would be the largest group of stakeholders using this system.

Physician/Personal trainer: This project will help the physicians/personal trainers to always keep track of their patient exercise routine. This would help save time in unnecessary appointments since he/she would have all the patient statistics (like heart rate) and it can be helpful in finding anomalies early.

Project team members: The team members also form a major stakeholder for the project who came up with an idea for it and have a major role in developing and executing it. They are also affected by the outcome and can, thus, be listed as stakeholders for the project.

4.2 Actors and Goals

Actors can be defined as people or devices that will directly interact with the product or the system. All these actors would have specific goals by interacting with the system. For this project, the following would be the actors:

User/Patient: The patient's goal is to access the mobile application to keep track of his/her exercise schedule and health conditions.

Physician: The physician's goal is to access the web application to keep track of his/her patient statistics for easy and early diagnosis.

Local Database: The local database (SQLite) would be used in the initial stages of the project to store the records of all patient data(like profile data). It will also store the health data for each patient separately.The mobile application will access the local database to retrieve all the required information.

Online Database: The online database would be used in the later stages of project for the same purpose as the local database. Firebase Real Time Database (cloud) would be used for the same which can be accessed by both the mobile application as well as the web application.

Mobile Application: The mobile application allows the user/patient to interact with the system and transfer of information between both. It acts as a participating actor in most of the use cases.

Web Application: The web application allows the physician to interact with the system and keep track of all patient records, manage their schedules.

4.3 Use Cases

Use case modeling is a technique used to represent system requirements. Each use case describes how the user and system interact to achieve business goals. [15] The sections below contain descriptions of our use cases and identify how they align with the system requirements.

4.3.1 Casual Description

Below is a brief summary of each of the use cases.

UC-1: Toggle Activity - Allows the Patient to start and stop the app's activity and/or heart rate monitoring.
Derived from REQ-12.

UC-2: Health Data - Allows the Patient to check and monitor activity using the smartphone app.
Derived from REQ-1.

We analyzed the idea of allowing the patient to monitor his/her activity without being logged in. That means not having/using an account to save all the information, and by implication, not communicating with the physician. There will be no point in doing that, since there are many apps that already have this functionalities. Because of this, it is implicit that the patient must be signed in to start using and monitoring his/her activity.

UC-3: Get Vital Signs Phone - Allows the Patient to retrieve his/her activity log, heart rate, calorie count, and progress information through the smartphone application interface.
Derived from REQ-2, REQ-3 and REQ-4.

UC-4: Get Vital Signs Web - Allows the Physician to retrieve the user activity log, heart rate, calorie count, and progress information through the web interface.
Derived from REQ-2, REQ-3, REQ-4 and REQ-7.

UC-5: Enable Notifications - Allows the Patient to receive notifications about his/her progress. (optional sub use case, «extend» UC-2: Health Data).
Derived from REQ-6.

UC-6: Get Notes - Allows the Patient to receive notes posted by the Physician. (mandatory sub use case, «include» from UC-3: Get Vital Signs Phone).
Derived from REQ-9.
UC-7: Manage Schedule - Allows the Physician to setup and control exercise schedule in conjunction with the Patient (optional sub use case, «extend» UC-2: Health Data).
Derived from REQ-10.

UC-8: Toggle Alarm - Allows the Patient to receive an alarm when it's time to exercise, or to turn it off (optional sub use case, «extend» UC-2: Health Data).
Derived from REQ-6 and REQ-11.

UC-9: Enable Weekly Updates - Allows the Physician to receive automatic weekly updates about the Patient's activity and heart rate status.
Derived from REQ-8.

UC-10: Send Notes - Allows the Physician to update the system with notes for the Patient.
Derived from REQ-9.

UC-11: Physician Sign-in - Allows the Physician to manage users accounts, doing operations like add/delete accounts.
Derived from REQ-5.

UC-12: User Sign-up - Allow the Patient to create his/her own account using her/his own personal information.
Derived from REQ-5.

UC-13: User Sign-in - Allow the Patient to access his/her own account using her/his own personal valid credentials, and will have the ability to modify his/her personal information.
Derived from REQ-5.

4.3.2 Use Case Diagram

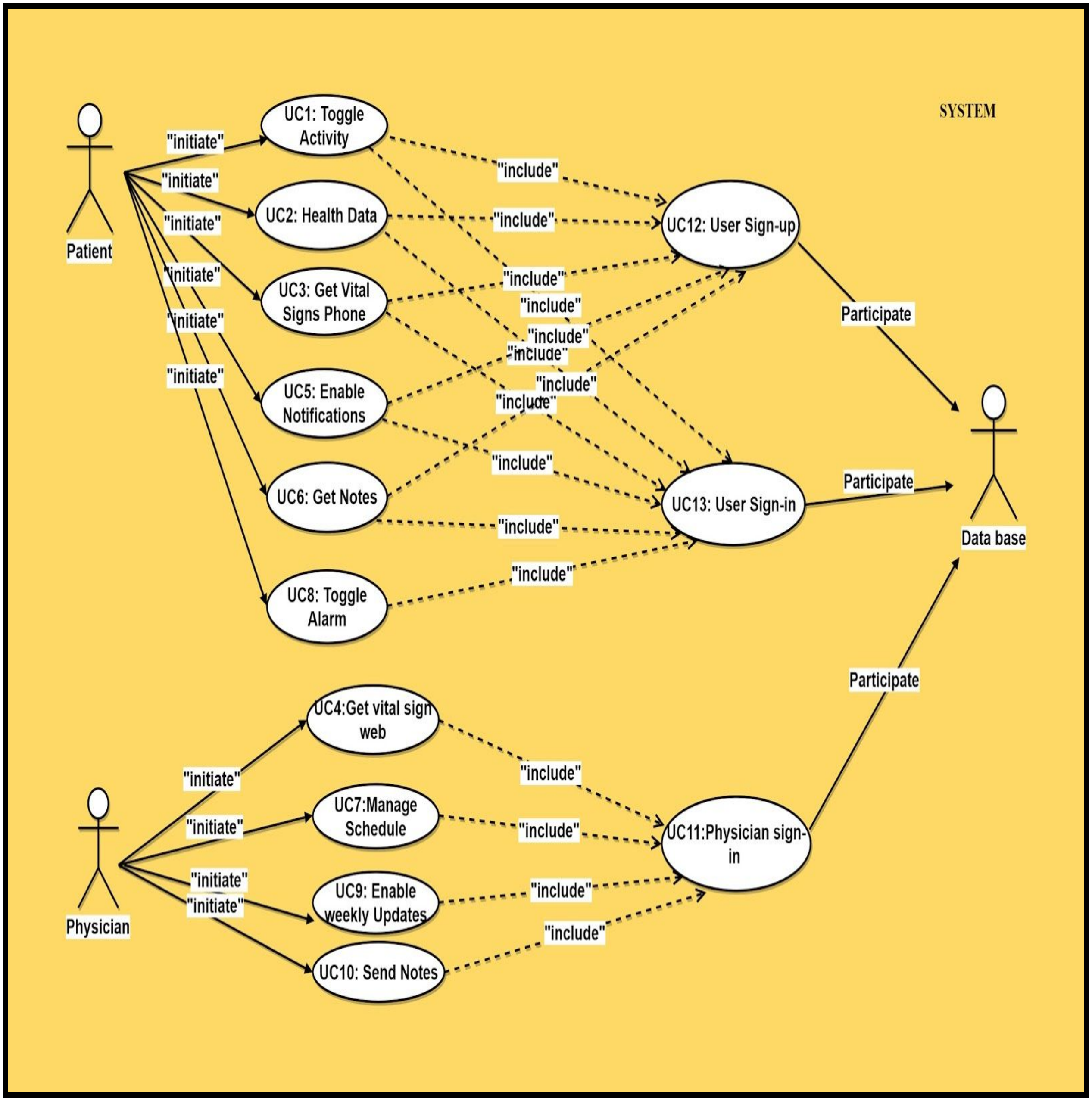


Figure 4.3.2.1- Use cases diagram showing the System Use case

4.3.3 Traceability Matrix

Table 4.3.3.1: Traceability matrix

Req't	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13
REQ-1	5		X											
REQ-2	4			X	X									
REQ-3	4			X	X									
REQ-4	4			X	X									
REQ-5	4											X	X	X
REQ-6	2					X			X					
REQ-7	3				X									
REQ-8	5									X				
REQ-9	4						X				X			
REQ-10	2							X						
REQ-11	1								X					
REQ-12	4	X												
Max PW		4	5	4	4	2	4	2	2	5	4	4	4	4
Total PW		4	5	12	15	2	4	2	3	5	4	4	4	4

4.3.4 Fully-Dressed Description

Below are the most important use cases. Without them, the system will be no longer functional.

Use Case UC-2	**Health Data**
Related Requirements	REQ-1 stated in the Table 4.3.3
Initiating Actor	Patient
Actor's Goals	To monitor his/her own activity
Participating Actors	Mobile Application, Local Database
Preconditions	● Smartphone has sufficient battery charge. ● The Patient has a valid registration credential (signed-in) in order to store/access his/her own data.
Post conditions	● Activity record was stored in Local Database.
Flow of Events for the Main Success Scenario	→ **1.** (a) **Patient** opens the app, (b) initiate the tracking function and (c) starts exercising. ← **2.** (a) **Mobile Application** counts the steps, (b) provides relevant data and (c) sends eventual notifications to the **Patient.** → **3. Patient** stops the tracking function. ← **4.** (a) **Mobile Application** shows the Patient his/her current Session and (b) stores it in Local Database.
Flow of Events for Extension (Alternative Scenario)	2a. **Patient** stops moving for a very long time. ← **1. Mobile Application** detects it. ← **2. Mobile Application** stops tracking activity.

Use Case UC-3	**Get Vital Signs Phone**
Related Requirements	REQ-2, REQ-3 and REQ-4 stated in Table 4.3.3
Initiating Actor	Patient
Actor's Goals	To retrieve the patient information and activity logs from the database using the Mobile Application.
Participating Actors	Mobile Application, Local Database
Preconditions	● Smartphone must have sufficient battery charge.
Post conditions	● **Mobile Application** displayed only the chosen record.
Flow of Events for the Main Success Scenario	→ **1.** (a) **Patient** opens the app and (b) presses the button corresponding to history logs. ← **2.** (a) **Mobile Application** retrieve information from **Local Database** and (b) shows the list of available activity records. → **3. Patient** selects the desired record.
Flow of Events for Extension (Alternative Scenario)	2b. **Mobile Application** shows an empty list because **Patient** has no previous activity and vital sign records. ← **1. Mobile Application** prompts **Patient** to start exercising for the first time.

Use Case UC-4	**Get Vital Signs Web**
Related Requirements	REQ-2, REQ-3, REQ-4 and REQ-7 stated in Table 4.3.3
Initiating Actor	Physician
Actor's Goals	To retrieve the Patient information and activity logs from the Online Database using the Web Page.
Participating Actors	Web Page, Online Database
Preconditions	● **Physician** has a valid registration credential (signed-in) in order to access the **Web Page**. ● **Patient** must have previous activities and vital signs record in order for the **Physician** to view his log.
Post conditions	● **Web Page** displayed the desired **Patient's** record.
Flow of Events for the Main Success Scenario	→ **1.** (a) **Physician** accesses the **Web Page** and (b) selects the desired **Patient**. ← **2.** (a) **Web Page** retrieves information from **Online Database** and (b) displays the **Patient's** records. → **3. Physician** selects a specific record to see.
Flow of Events for Extension (Alternative Scenario)	2b. **Web Page** shows an empty list because **Patient** has no previous activity and vital sign records.

Use Case UC-11	**Physician Sign-in**
Related Requirements	REQ-5 stated in the Table 4.3.3
Initiating Actor	Physician
Actor's Goals	To manage Patient accounts, doing operations like add/edit/remove
Participating Actors	Physician, Online Database
Preconditions	● **Physician** has a valid registration credential (signed-in) in order to access the **Web Page**.
Post conditions	● Patient was added/edited/removed to/in/from the database. ● In case of removal, **Online Database** no longer has Patient's information.
Flow of Events for the Main Success Scenario	→ **1.** (a) **Physician** accesses the **Web Page** and (b) selects the desired **Patient**. ← **2. Web Page** prompts the **Physician** with the available options. → **3. Physician** selects one of the options (add/edit/remove) ← **4. Web Page** updates the **Online Database**.
Flow of Events for Extension (Alternative Scenario)	3a. **Web Page** cannot add/edit/remove a specific **Patient** ← **1. Web Page** alerts **Physician** saying that **Patient** already is/is not in **Online Database**

4.4 System Sequence Diagrams

A **system sequence diagram** (SSD) is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate, their order, and possible inter-system events. System sequence diagrams are visual summaries of the individual use cases. Below are the System Sequence Diagrams for few important Use cases:

Use Case UC-2: Health Data

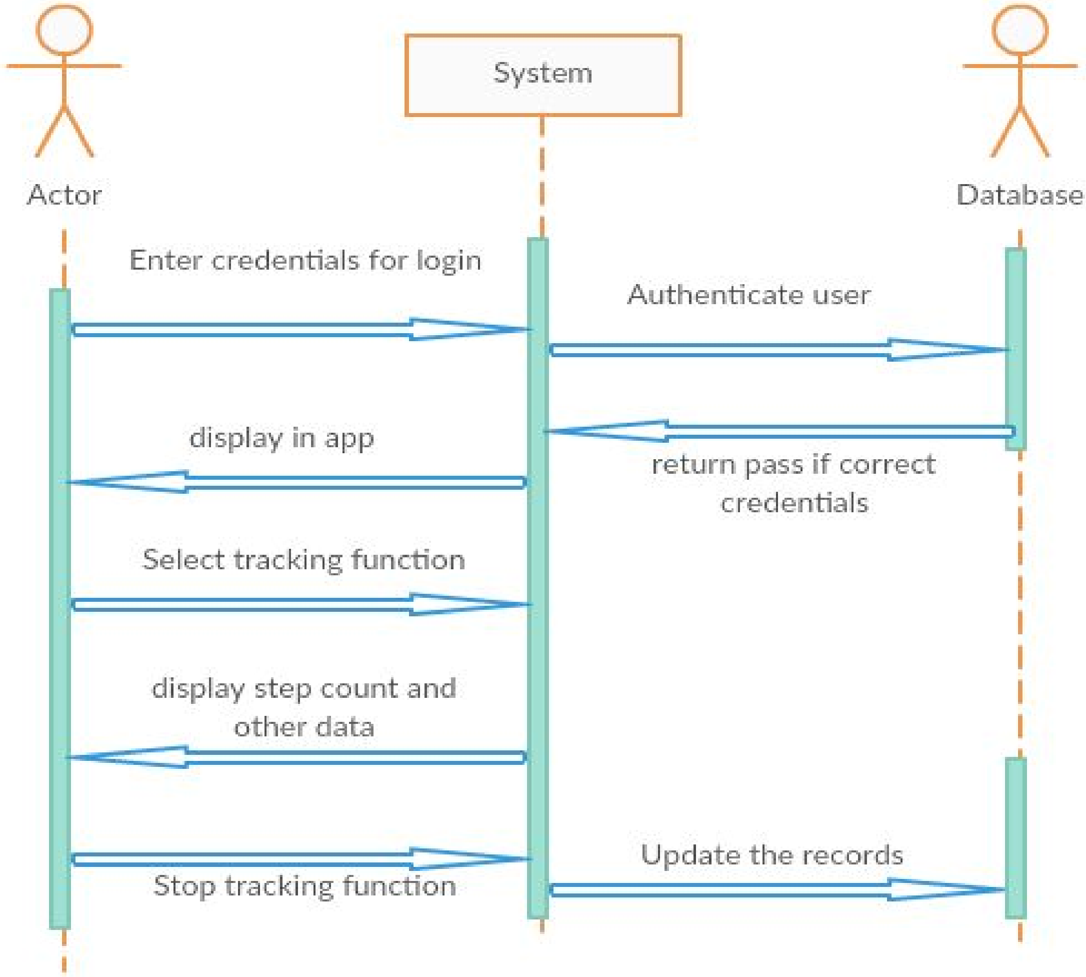


Figure 4.4.1- Sequence diagram for Use case UC-2:Health Data

Use Case UC-3 : Get vital signs Phone

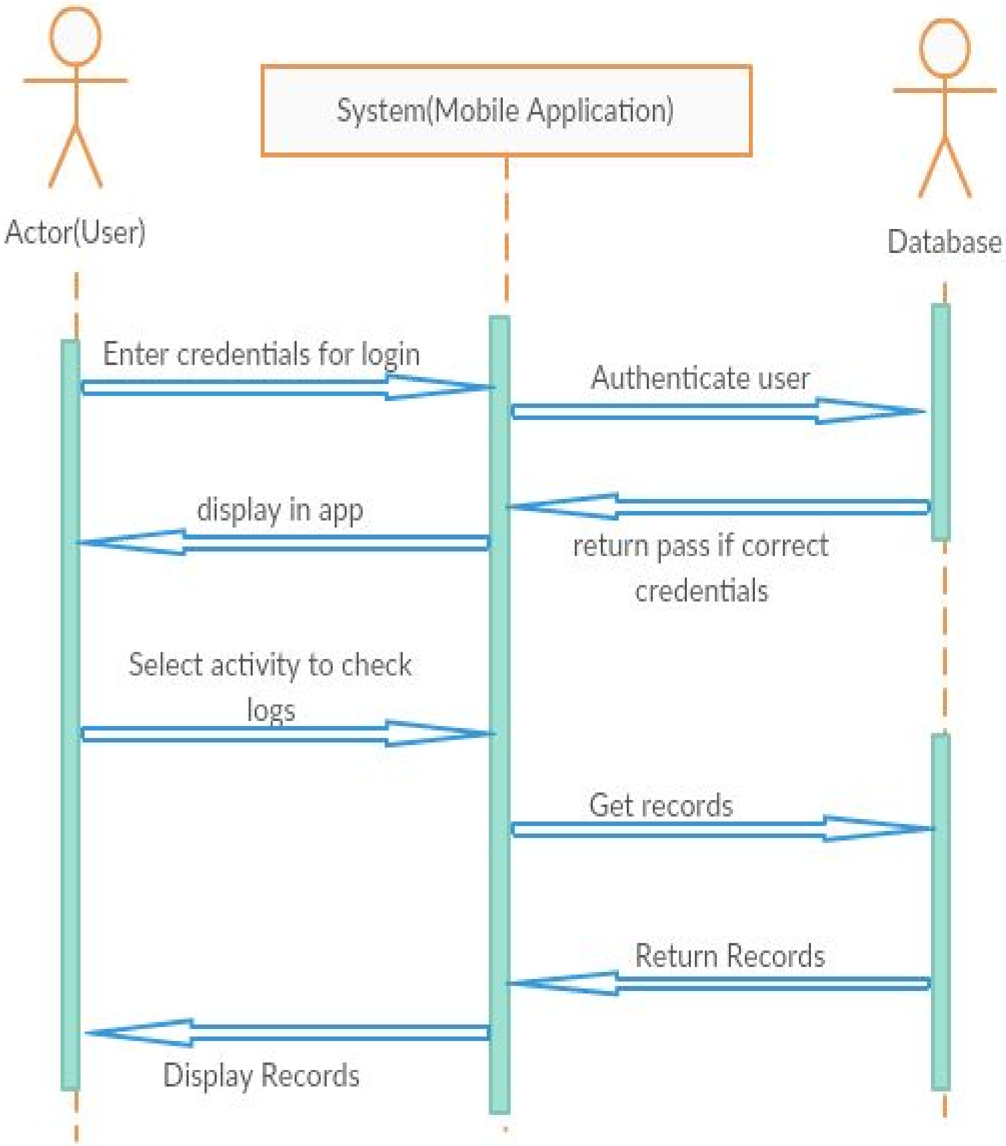

Actor(User)
System(Mobile Application)
Database
Enter credentials for login
Authenticate user
display in app
return pass if correct credentials
Select activity to check logs
Get records
Return Records
Display Records

Figure 4.4.2- Sequence diagram for Use case UC-3:Get Vital Signs Phone

Use Case UC- 4: Get Vital Signs Web

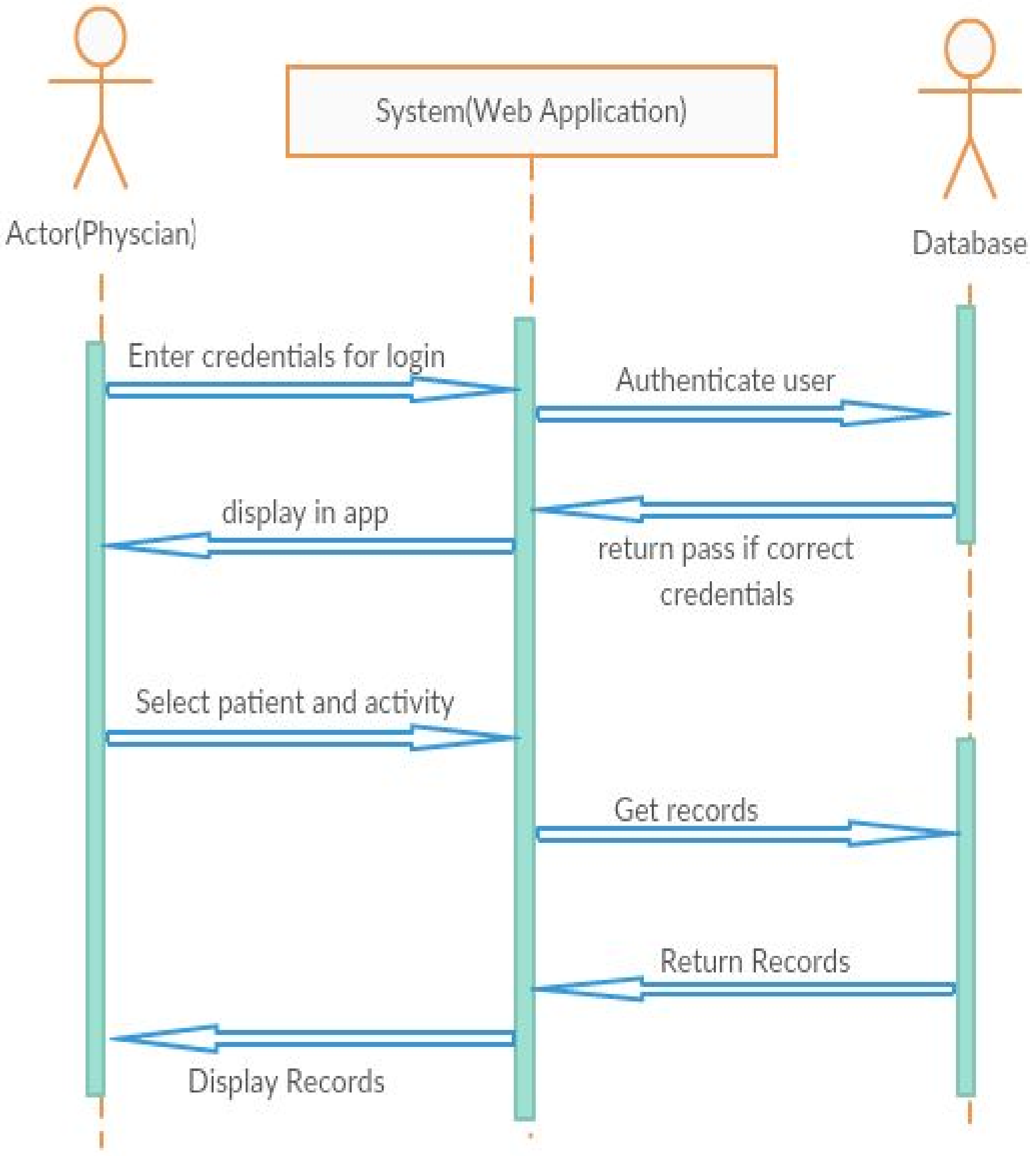


Figure 4.4.3- Sequence diagram for Use case UC-4:Get Vital Signs Web

Use Case UC-11: Physician Sign-in

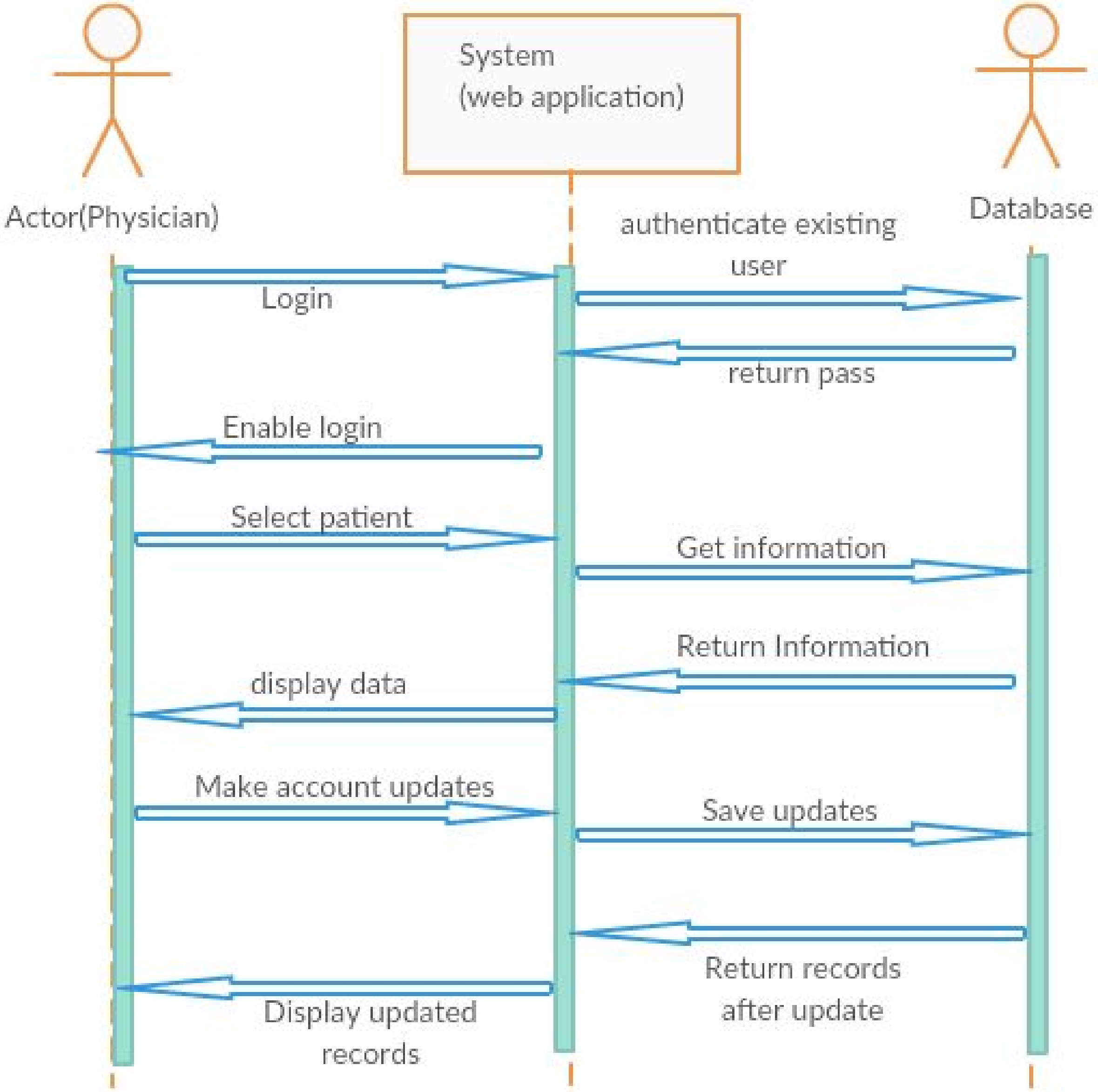


Figure 4.4.4- Sequence diagram for Use case UC-11:Physician Sign-In

5 User Interface Specification

5.1 Preliminary Design

5.1.1 Registration and Login (UC-12 and UC-13)

When the mobile app is installed, the user will be able to Register with minimal information. The details will be stored in a database which will help with authentication when the user tries to login. The login page will then verify the credentials provided and grant access if information is present in the database. All icons used for design are based on stock photos available online for reuse in non-commercial purpose.

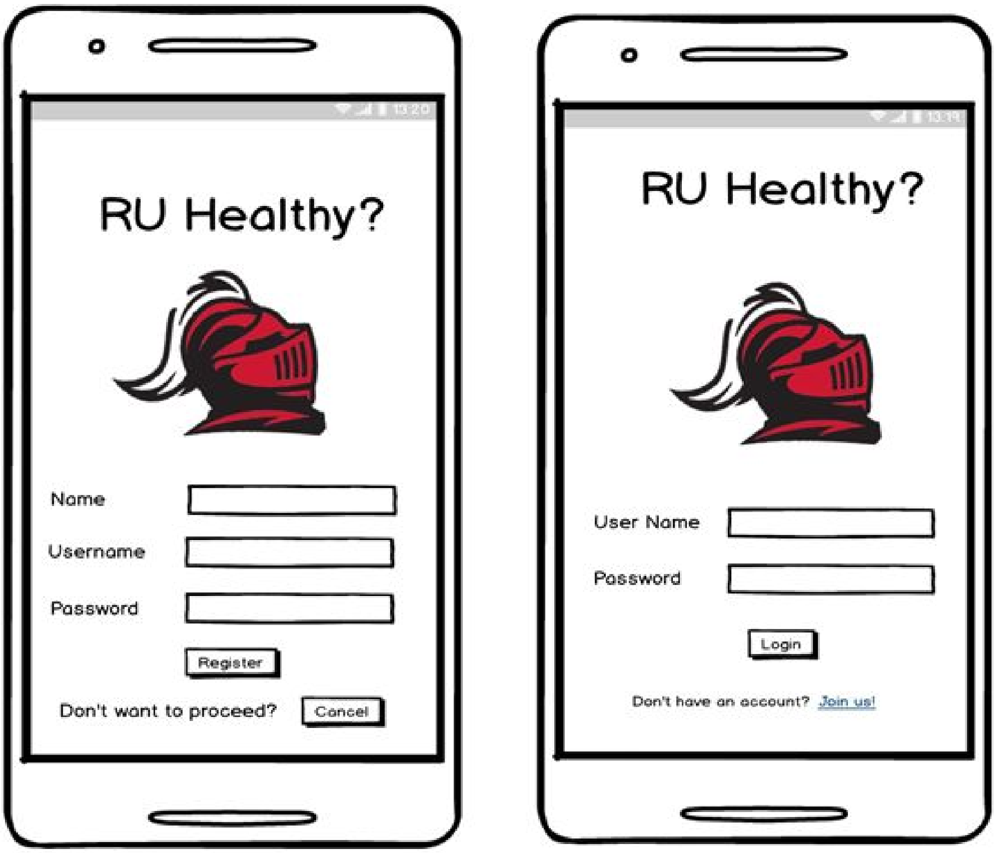


Figure 5.1.1- Registration and Login UI

5.1.2 Profile information (UC-13)

Once the User is logged into the application, they will be asked to enter some basic information. There will also be an optional page to enter the user's physician/personal trainer's information which will serve as authentication for the physician to be able to view their clients' information and activity. The activity is stored in an online database which will be shared in the admin's web application. This will then display the information to the authenticated physician.

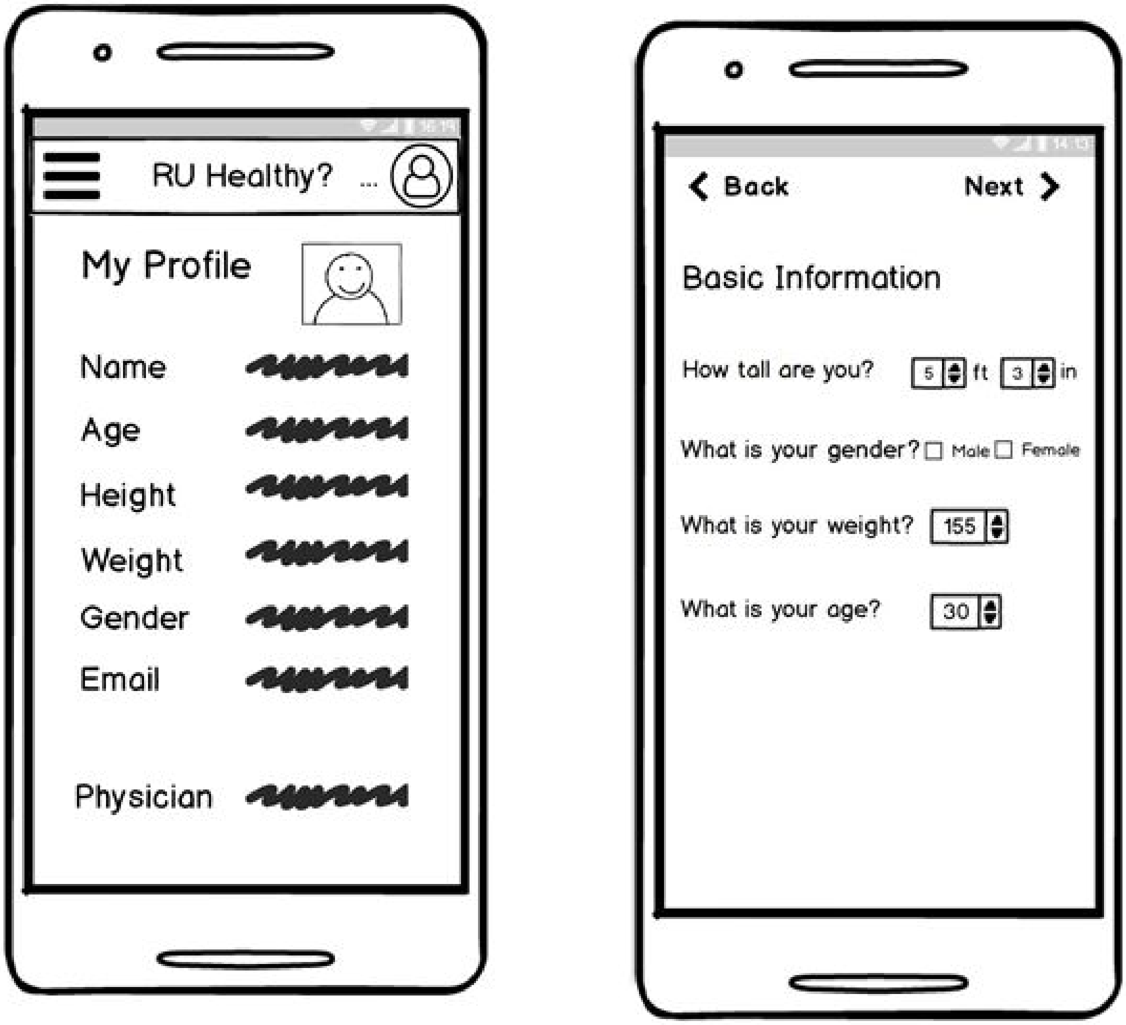
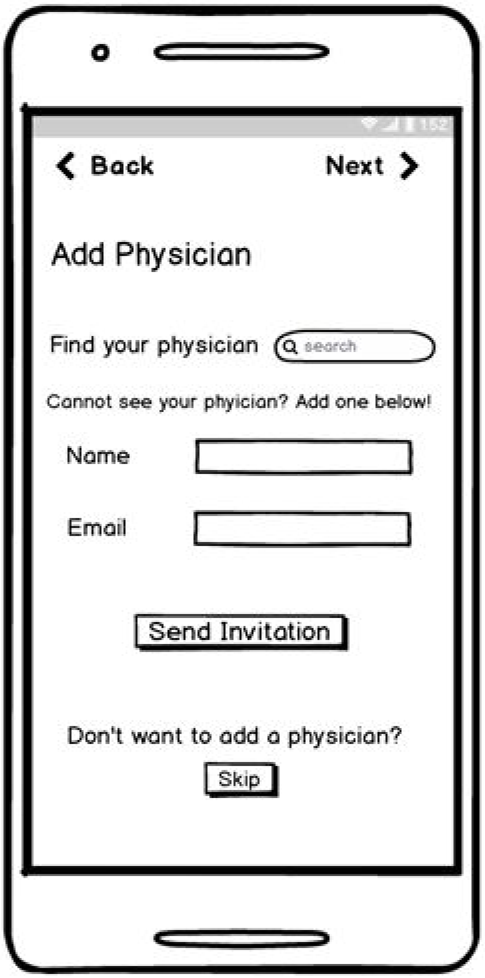


Figure 5.1.2- Profile Information and Basic Information UI



Cannot see your phyician? Add one below!

Don't want to add a physician?

Figure 5.1.3 - Add Physician App UI

5.1.4 Activity screen (UC-3)

User’s current activity is displayed on the home screen and includes steps taken and calories burnt. Additionally, we will also display heart rate information. These details are regularly stored in the online database.

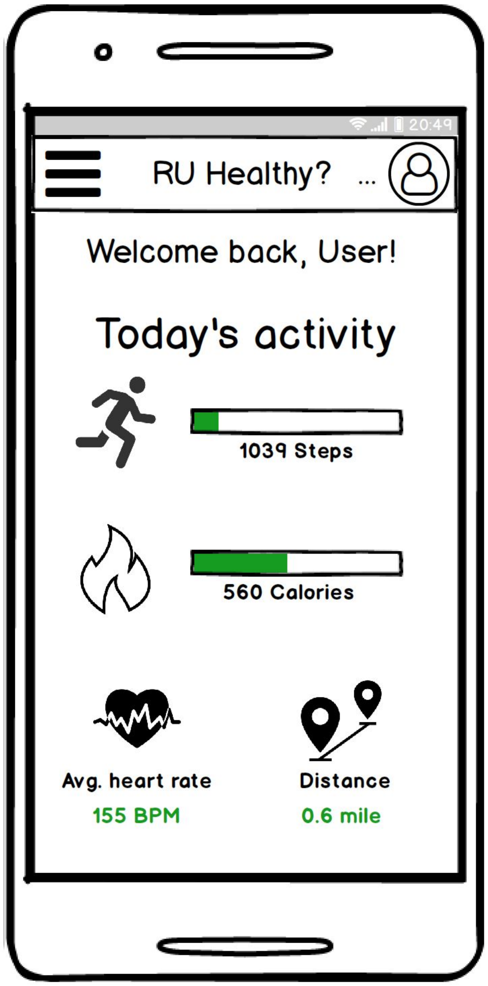
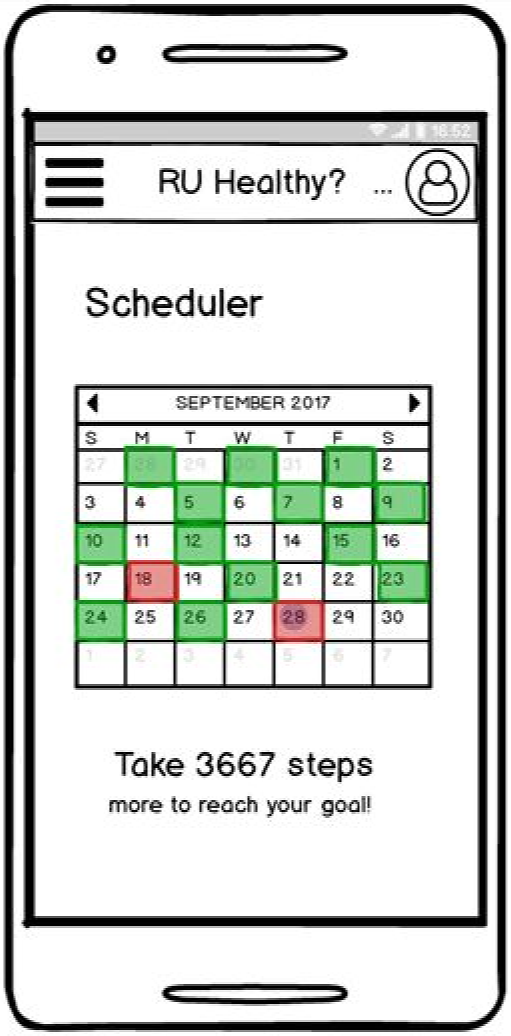


Figure 5.1.4 - Activity Monitor App UI

5.1.5 Scheduler (UC-7)

The User will be able to schedule or allocate some time for exercise every day. The app will then wait for the user to perform some kind of activity at the allotted time. When the app detects inactivity, using the mobile phone sensors, it will remind the User that it is time to exercise by sending a notification to the User.



Scheduler

SEPTEMBER 2017

S	M	T	W	T	F	S
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

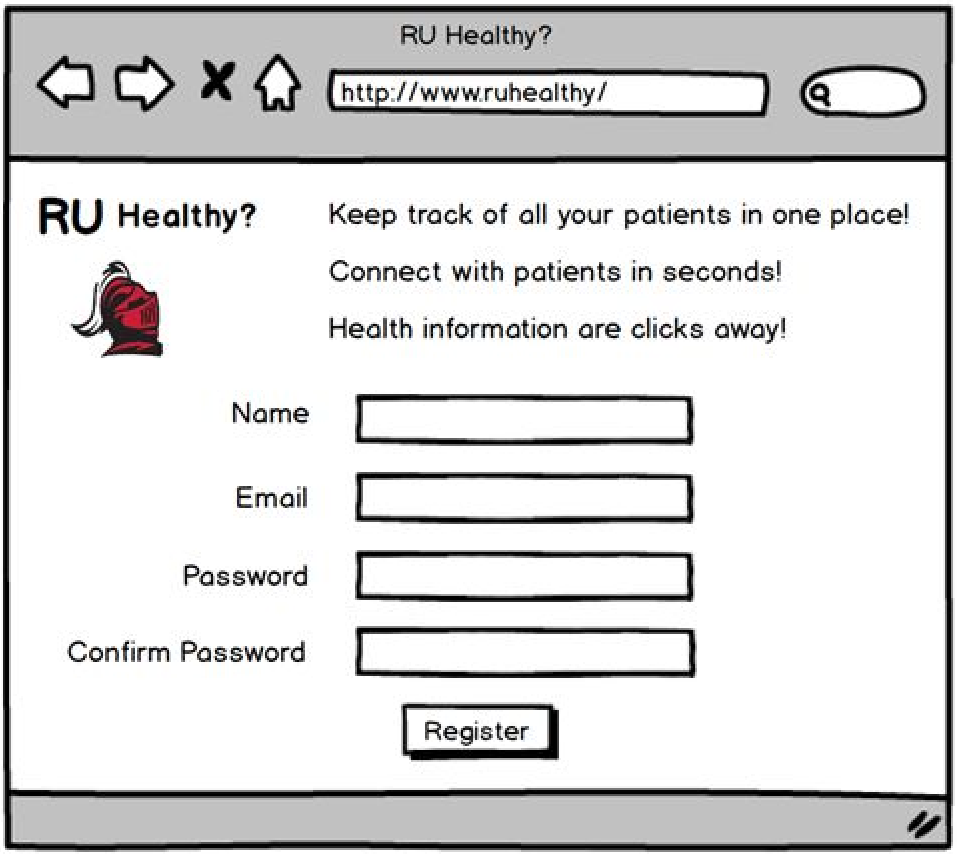
Take 3667 steps
more to reach your goal!

Figure 5.1.5 - Scheduler App UI

5.1.6 Web Application - View Client profiles and Activity (UC-4)

The admin (physician or personal trainer) will be able to view client profiles after sue authentication. Admin with these privileges can view client profiles and their activity on a particular day or set of days.

The User can login using his credentials to ensure privacy of data.

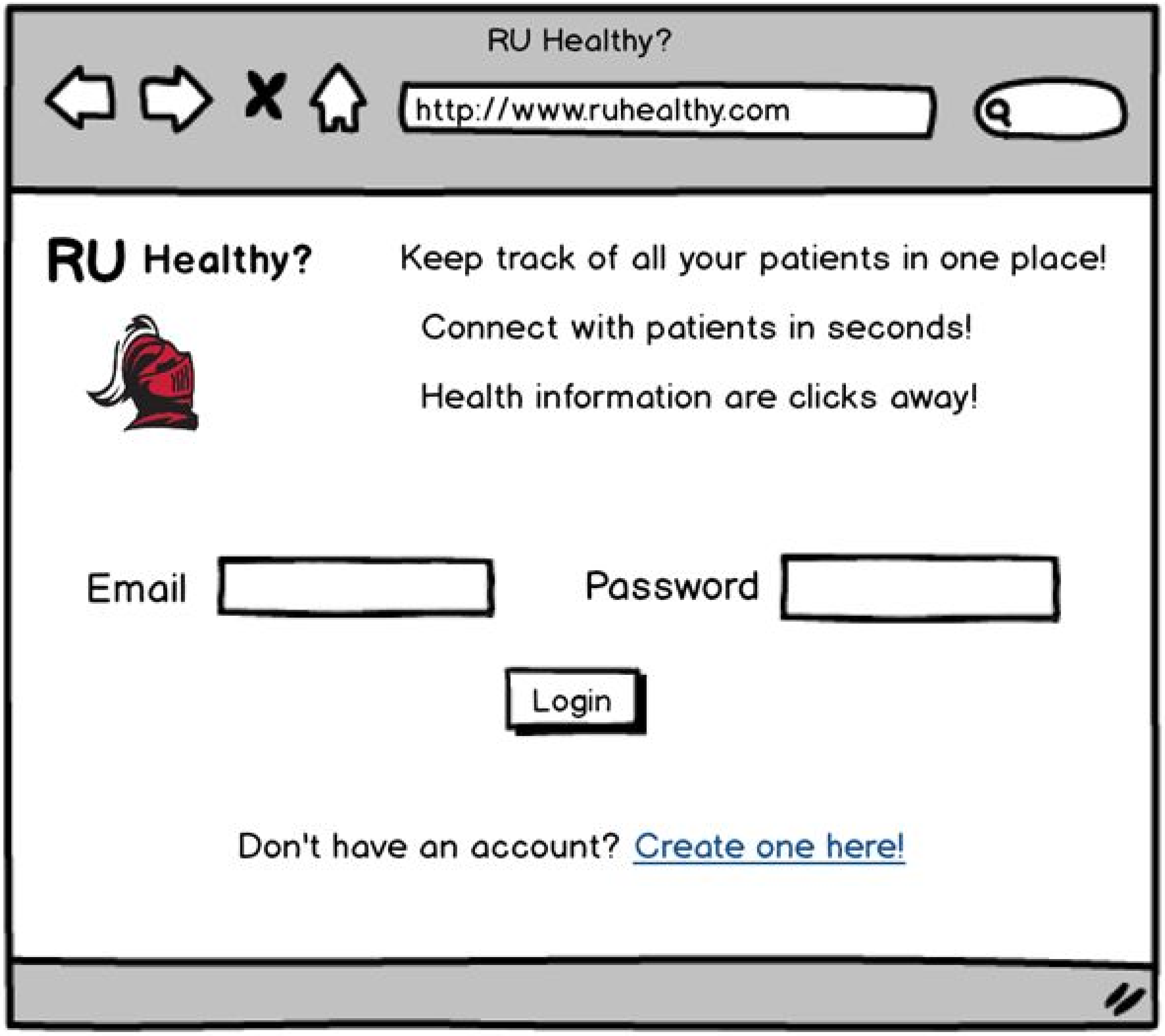


Keep track of all your patients in one place!

Connect with patients in seconds!

Health information are clicks away!

Figure 5.1.6 - Register Web UI



Keep track of all your patients in one place!

Connect with patients in seconds!

Health information are clicks away!

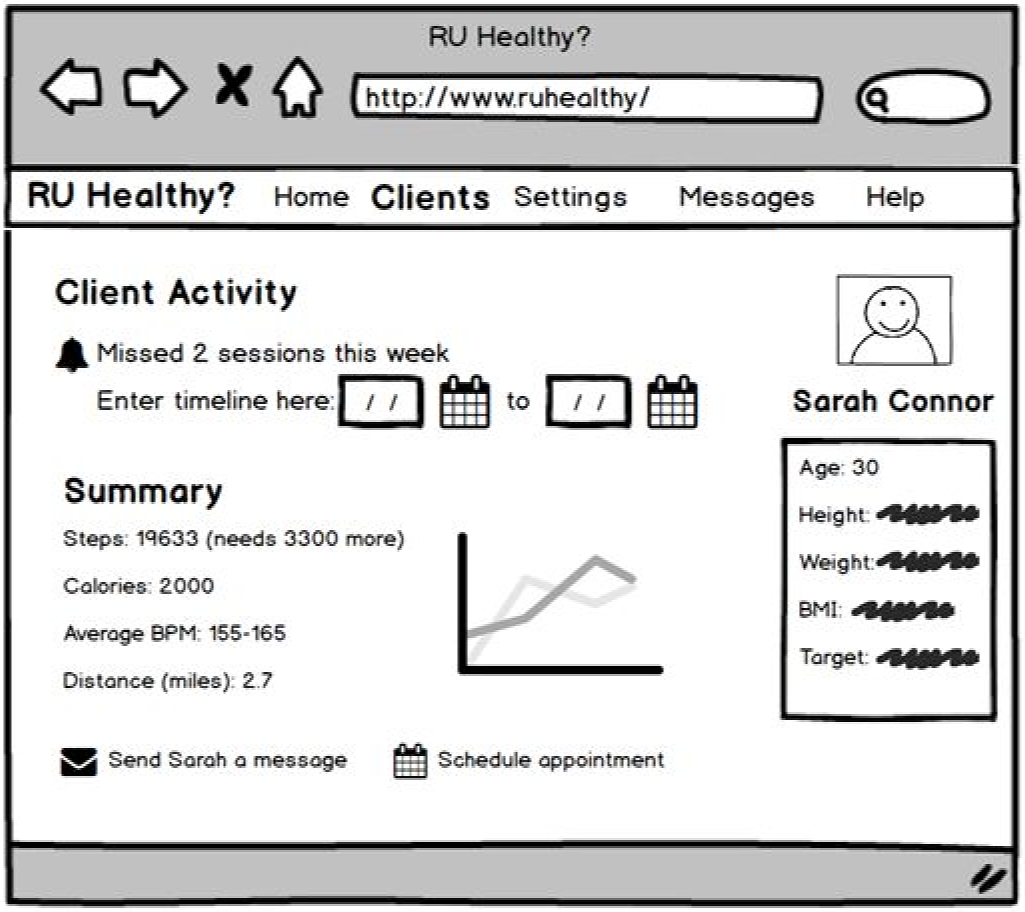
Don't have an account? Create one here!

Figure 5.1.7 - Login Web UI

The admin will be able to individually view each client’s profile which will include a number of options such as:

- the client’s basic information
- the client’s current activity
- a health record
  (This section will allow the admin to get a quick understanding of the client’s status)
- a message tool
  (This section will let the admin send important messages to that particular client)
- an analysis section

(This section will let the admin see a graphical representation of the client's activity status for a particular period of time. Furthermore, it will include some notes for future reference.)



Client Activity

Missed 2 sessions this week

Enter timeline here: / / to / /

Summary

Steps: 19633 (needs 3300 more)

Calories: 2000

Average BPM: 155-165

Distance (miles): 2.7

Sarah Connor

Age: 30

Height:

Weight:

BMI:

Target:

Figure 5.1.8 - Summary Web UI

5.2 User Effort Estimation

We are going to compute User Effort Estimation by the method of Estimation with Use Case Points. It works on the basic principle that more the complexity of the use case more is the time taken to design, develop, test and implement.

5.2.1 Unadjusted Use Case Points (UUCP)

We calculate UUCP by taking a sum of Unadjusted Actor Weights and Unadjusted Use Case Weights.

Table 5.2.1.1- Unadjusted Actor Weight

ACTOR	COMPLEXITY	WEIGHT
Web Application	Medium	2
Database	Medium	2
RU Healthy Mobile App UI	Simple	1
Users	Complex	3

UAW = 2+2+1+3 = 8

We allocate points to each of our Use Cases using the following table:

Table 5.2.1.2 - Unadjusted Use Case Weights

Use Case complexity	Weight
Simple	5
Medium	10
Complex	15

Based on our casual description of Use cases, we allocate points to each of our Use Cases (13 in number.)

We then calculate the UUCW by summing up points to all Use cases.

Weights have been allocated in the order of mention is the section 4.2.3 Casual description of Use cases. (i.e., from UC-1 to UC-13 in that order)

UUCW = 5 + 10 + 15 + 15 + 5 + 5 + 10 + 5 + 10 + 5 + 10 + 5 + 5 = 105

Therefor, UUCP = UAW + UUCW = 8 + 105 = 113

5.2.2 Technical Complexity Factor (TCF)

Table 5.2.2- Technical Complexity Factor

TECHNICAL FACTOR	DESCRIPTION	WEIGHT	COMPLEXITY	TOTAL
T1	Interface that User finds easy to use and less cluttered	3	1	3
T2	Ease of access to historical activity record- Database	4	2	8
T3	Accurate tracking using sensors	3	2	6
T4	Web app Security is important as it concerns health records	3	1	3
T5	Adjustments to cater to battery saving, memory	1	2	2

	management - minor			

Technical Factor Total (TFT) = 3+8+6+3+2 = 22
TCF = C1+ C2 * TFT where C1 and C2 are constants with values, C1= 0.6, C2 = 0.01
TCF = 0.6 + 0.01 * 22 = 0.82

5.2.3 Environmental Complexity Factors (ECF)

Table 5.2.3- Environmental Complexity Factors

ENVIRONMENTAL FACTOR	DESCRIPTION	WEIGHT	IMPACT	TOTAL
E1	Familiarity with health monitoring and defined problem	1	1	1
E2	Beginner knowledge of Android development approach	2	2	4
E3	Knowledge of Web application development	1.5	2	3
E4	Teamwork and motivation	2	1	2
E5	Meeting specified requirements	3	2	6

Environmental Factor Total (EFT) = 16
ECF = C1 + C2*EFT

Where C1 and C2 are constants with the values C1=1.4, C2=-0.03
ECF = 1.4 + (-0.03 * 16) = 0.92

5.2.4 Total Use Case Points and Final Effort

Finally we calculate the total use case points by the equation,

UCP = UUCP * TCF * ECF = 113 * 0.82 * 0.92 = 85.25

Final Effort = UCP * Conversion factor

We assume a conversion factor of 20 hours (Java based application)

Therefore, Final Effort = 85.25 * 20 = 1705

6 Domain Analysis

6.1 Domain Model

For the most important fully-dressed use cases specified above, we first start extracting the responsibilities of each one, and we proceed to assign concept names. The following tables contain this information.

6.1.1 Concept Definitions

Table 6.1.1.1 - Concepts And Responsibilities Extraction For UC-2: Health Data

Responsibility Description	Type	Concept Name
Controls and coordinates of all other concepts associated with use cases.	D	Controller
Contains all the Patient's data that is, was, and will be detected during the Patient's activity.	K	Data Container
Operates the process of user authentication with the Database.	D	Database Authenticator
Operates the device tracking functionality by disabling and enabling it when it is necessary.	D	Tracking Enabler
Operates the process of managing Local Database entries and update them with the Online Database.	D	Database Manager
Operates the process of managing notifications	D	Notifications Enabler
Operates the process of data detecting from the device sensors with the help of "Tracking Enabler" concept.	D	Sensor Operator

Table 6.1.1.2 - Concepts And Responsibilities Extraction For UC-3: Get Vital Signs Phone

Responsibility Description	Type	Concept Name
Controls and coordinates of all other concepts associated with use cases.	D	Controller
Manages and specifies the Patient search parameters and the Patient's activity logs retrieval.	K	Search Manager
Provides the connections between the Patient search inputs and the Offline Database and retrieve the matched records from said Database	D	Database Operator
App interface shows available activity records of patient.	K	Interface
Renders and projects the retrieved records from the Database and sends it to the User Application GUI	D	Pagemaker
Filters and applies the specific Patient search criteria for the retrieved data from the Database .	D	Database Postprocessor
Operates the process of managing and sending notifications to Patient if he does not have valid vital signs.	D	Activity Notifier

Table 6.1.1.3 - Concepts And Responsibilities Extraction For UC-4: Get Vital Signs Web

Responsibility Description	Type	Concept Name
Control and coordinates of all other concepts associated with use cases.	D	Controller
Manages and specifies the Physician search parameters and the Patient's activity log retrieval.	K	Search Manager

Provides the connections between the Physician search inputs and the Online Database and retrieve the matched records from said Database	D	Database Operator
Renders and projects the retrieved records from the Database and sends it to the Web Page UI	D	Pagemaker
App interface shows available activity records of patient.	K	Interface
Filters and applies the specific Physician search criteria for the retrieved data from the Database .	D	Database Postprocessor
Operates the process of managing and sending notifications to Physician if Patient does not have valid vital signs.	D	Activity Notifier

Table 6.1.1.4 - Concepts And Responsibilities Extraction For UC-11: Physician Sign-in

Responsibility Description	**Type**	**Concept Name**
Control and coordinates of all other concepts associated with use cases.	D	Controller
Manages and specifies the Physician search parameters	K	Patient Searcher
Manages and specifies the operation (add/edit/remove) to be performed by the Physician	K	Patient Operator
Compares search input with Online Database to verify if user is/is not in said Database	D	Patient Comparator
Keeps the Online Database synced accordingly to the operation specified by the concept “Patient Operator”	D	Database Manager

Alerts the Physician if Patient already is/is not in the Online Database	D	Patient Notifier

6.1.2 Associations

Associations are used to *which concept* needs to work together and *why* do they need to work together. In the following tables, we derived associations for principal use cases of our project (UC-2, UC-3, UC-4, UC-11) based on the fully dressed descriptions in section 4.3.4.

Table 6.1.2.1 - Identifying associations for use case UC- 2: Health Data

Concept pair	**Association description**	**Association name**
Controller ↔ Tracking Enabler	Controller passes request to Tracking Enabler to start tracking user motion via sensor and receives confirmation on Tracking initiation.	Conveys tracking request
Tracking Enabler ↔ Sensor Operator	Tracking Enabler sends signal to sensor for activation and receives confirmation.	Conveys sensor activation request
Sensor Operator ↔ Database Manager	Sensor sends sensor data to Database Manager for processing and storage.	Provides data
Database Manager ↔ Data Container	Database Manager sends data received from sensor to Data Container for local data storage.	Stores data
Controller ↔ Notification Enabler	Controller passes request to Notification Enabler for notifications based data stored in Data container.	Requests notification
Notification Enabler ↔ Data Container	Notification Enabler requests for processed data from Data Container.	Requests processed data
Controller ↔ Database	Controller requests access to	Requests access

Authenticator	Online database for updating data and receives access permission.	
Controller ↔ Database Manager	Controller sends signal to Database manager to send current to Online Database for storage.	Requests save

Table 6.1.2.2 - Identifying associations for use case UC- 3: Get Vital Signs Phone

Concept pair	**Association description**	**Association name**
Controller ↔ Pagemaker	Controller passes request to Pagemaker and Pagemaker prepares available activity records for display.	Conveys display request
Pagemaker ↔ Database Operator	Pagemaker requests for available records and receives data retrieved by Database Operator.	Requests data
Pagemaker ↔ Interface	Pagemaker prepares the received data for display on the App Interface.	Display data
Pagemaker ↔ Search Manager	Pagemaker conveys patient's search request parameters to Search Manager and receives retrieved results for display.	Requests search operation
Search Manager ↔ Database Postprocessor	Search Manager sends specified search query based on search criteria to Database Postprocessor.	Conveys specified search query
Pagemaker ↔ Activity Notifier	Pagemaker requests notification on patent's vital sign and receives them based on retrieved data from database.	Requests notification

Table 6.1.2.3 - Identifying associations for use case UC- 4: Get Vital Signs Web

Concept pair	Association description	Association name
Controller ↔ Pagemaker	Controller passes request to Pagemaker and Pagemaker prepares available activity records for display.	Conveys display request
Pagemaker ↔ Database Operator	Pagemaker requests for available records and receives data retrieved by Database Operator.	Requests data
Pagemaker ↔ Interface	Pagemaker prepares the received data for display on the Web Interface.	Display data
Pagemaker ↔ Search Manager	Pagemaker conveys Physician's search request parameters to Search Manager and receives retrieved results for display	Requests search operation
Search Manager ↔ Database Postprocessor	Search Manager sends specified search query based on search criteria to Database Postprocessor.	Conveys specified search query
Pagemaker ↔ Activity Notifier	Pagemaker requests notification on patient's vital sign and receives them based on retrieved data from database.	Requests notification

Table 6.1.2.4 - Identifying associations for use case UC-11: Physician Sign-in

Concept pair	Association description	Association name
Controller ↔ Patient Searcher	Controller sends search query to database to retrieve information of specific Patient.	Requests search operation
Patient Searcher ↔ Patient Comparator	Patient searcher compares search query to available patient list to verify and sends update.	Requests verification
Controller ↔ Patient Operator	Controller requests selected operation (add, remove or delete) on patient information.	Requests editing permission
Patient Operator ↔ Database Manager	Based on Physician's requested operation, Patient operator sends editing commands to database manager.	Provides editing command
Patient Comparator ↔ Patient Notifier	Patient comparator sends notification to Patient Notifier in case of not finding desired patient in the available list.	Send notification

6.1.3 Attribute Definitions

Table 6.1.3 - Attribute Definitions

CONCEPT	**ATTRIBUTE**	**ATTRIBUTE DEFINITION**
Controller	Central coordinator	Controls and coordinated all operations performed for a certain action performed by users
Sensor Operator	Mobile Application UI and functions	Enables and collects tracking data via phone sensors
Search Manager		Allows user to find specific physician/trainer
Tracking Enabler		Allows user to enable tracking-GPS
Notifications Enabler		Allows user to enable and change notification settings
Data Container	Database Operations	Stores mobile user data – profile and activity
Database Authenticator		Verifies user authenticity
Database Manager		Ensures continuous update of user data
Database Operator		Provides a connection between the database and the Web user
Database Postprocessor		Allows for specific data outputs based on search criteria – querying

Interface	Web Application UI	User Interface for the Web user
Pagemaker		Data processing and display for Web user
Patient Searcher	Web Application functions	Allows physician to search for specific records
Patient Operator		Allows physician to add or delete patient records
Patient Notifier		Alerts the Physician of incorrect input/search criteria

6.1.4 Traceability Matrix

Below is the traceability matrix which specifies how the different cases map to the domain concepts:

DOMAIN CONCEPTS	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13
Controller	X	X	X	X	X	X	X	X	X	X	X	X	X
Sensor Operator	X	X											
Search Manager							X						
Tracking Enabler	X	X	X										
Notifications Enabler				X	X				X	X			
Data Container		X	X		X							X	X
Database Authenticator													X
Database Manager		X	X	X	X	X	X	X	X	X	X		X
Database Operator				X			X		X	X	X		
Database Postprocessor		X	X	X	X	X	X	X	X	X	X		X
Interface				X			X		X	X	X		
Pagemaker				X			X		X	X	X		
Patient Searcher				X			X		X		X		
Patient operator											X		
Patient Notifier											X		

6.2 System Operation Contracts

System Operation Contracts are a way of documenting complex system sequence designs. Each operation contract describes what each message will need to accomplish.

CO-1: Register User
Cross References: UC-2, UC-3

- **Preconditions:** None
- **Postconditions:** User profile is linked to database and all user information is stored in database.

CO-2: User Login
Cross References: UC-2, UC-3

- **Preconditions:** User is already registered.
- **Postconditions:** The account features are accessed by the user.

CO-3: View Records:
Cross References: UC-3, UC-4, UC-11

- **Preconditions:** User has an account.
- **Postconditions:** None.

CO-4: Edit User Account
Cross References: UC-4, UC-11

- **Preconditions:**

1. Physician is logged in his/her account.
2. User to be edited/deleted has an account.

- **Postconditions:** Database updated according to made changes.

CO-5: Feedback by Physician
Cross Reference: UC-4

- **Preconditions:**

1. Physician is signed into the web application.
2. Physician has received data of requested user from the online database.

- **Postconditions:**

1. Doctor writes comments for the user based on his/her report.
2. User is notified about the comments.

6.3 Mathematical Models

6.3.1 Calorie Calculations

The System will have the capability to find the number of calories burned by the user based on certain factors:

1. Gender
2. Body Weight.
3. Average Heart Rate
4. Time

One of the more standard and most accurate ways to calculate the equation is to use the calorie expenditure formula below. It comes from the Journal of Sports Sciences and provides a formula for each gender.

We use the following equations to calculate calories burnt by the user:

Men:

$Calories\ Burned\ =\ [(Age\ *\ 0.2017)\text{—}(Weight\ *\ 0.09036)\ +\ (Heart\ Rate\ *\ 0.6309)\text{—}55.0969]\ *\ Time\,/\,4.184.$

Women:

$Calories\ Burned = [(Age\ *\ 0.074) - (Weight\ *\ 0.05741) + (Heart\ Rate\ *\ 0.4472) - 20.4022]\ *\ Time\,/\,4.184.$

Note : Heart Rate = average heart rate.

A less accurate Formula would be:

$Calories\ Burned = 0.75\ *\ your\ weight\ (in\ lbs.)$

These equations are based on the article titled “Energy Expenditure of Walking and Running” published in Medicine & Science in Sport & Exercise in 2004 [20].

6.3.2. Equivalent distance Calculation

The System will be able to calculate the distance in miles based on the number of Steps that the user did perform ,On the basis that a person of average height has a stride length of about 2.1 to 2.5 feet we can say that approximately 2,000 steps for 1 Mile so we can say that

*Number of Miles = Number of Steps performed *0.0005. []*

These calculations was published on 2004 by the ACSM(American College of Support and Medicine) and was certified by the American Nutrition Association.

6.3.3 Heart Rate

Normal heart rate can be in the range of *50% * Maximum heart rate – 85% * Maximum heart rate*

Where Maximum heart rate is approximately equal to 220 – *Age.*

6.3.4. Heart Recovery Rate

Heart rate is an important factor to assess the health of the user. We plan to use the heart rate data collected by the app to detect anomalies in heart rate after exercise. It is a documented medical fact that heart rate, once up during exercise, will to come down to the user's normal heart rate as a sudden drop before stabilizing. This drop is usually about 20 beats per minute in the first minute and about 15 beats per minute in the later couple of minutes. Anything less than 12 beats in the first minute, is a potential heart risk.

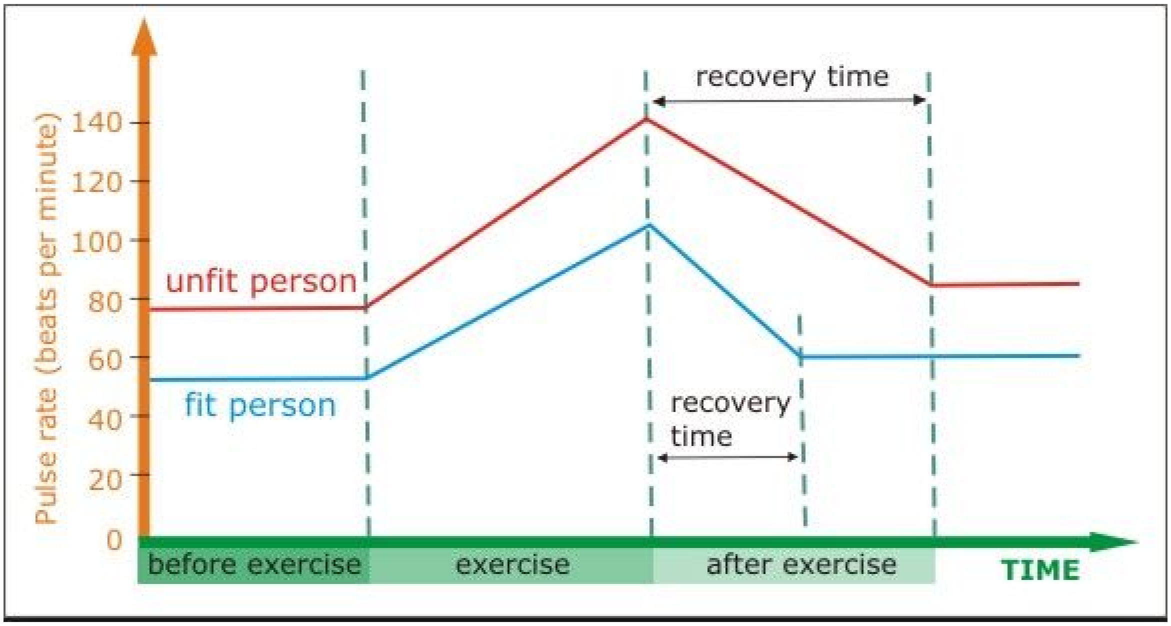


Fig.6.4.1 Heart rate pattern during and after exercise

The more fit a user is, faster his heart rate will get back to normal after exercise. Since the amount and intensity of exercise is an important factor in this recovery rate (the more you exercise, more time it takes to recover to normal rate), we make the assumption that the User's exercise pattern will indicate his heart recovery rate. Moreover, if a physician wants to monitor a patient who is recovering from a surgery or illness, this will be a useful analytical tool to show the user's recovery rate pattern and alert the doctor/user when there has been a series of abnormalities. We plan to use Bayesian curve fitting to obtain this pattern and predict the next heart rate. Bayesian Prediction is a useful statistical tool for short term prediction.

Bayesian prediction takes into account a trend that is being followed, and predicts the next values.

We have a data set of n points, $\mathbf{X} = \{x_1,x_2,\ldots\ldots..,x_n\}$, Bayesian prediction will fit the given data points into a curve and predict the next data point x_{n+1} based on the pattern. Since we assume that the user's exercise patterns are different every day, Bayesian prediction would help predict the next day's recovery rate.

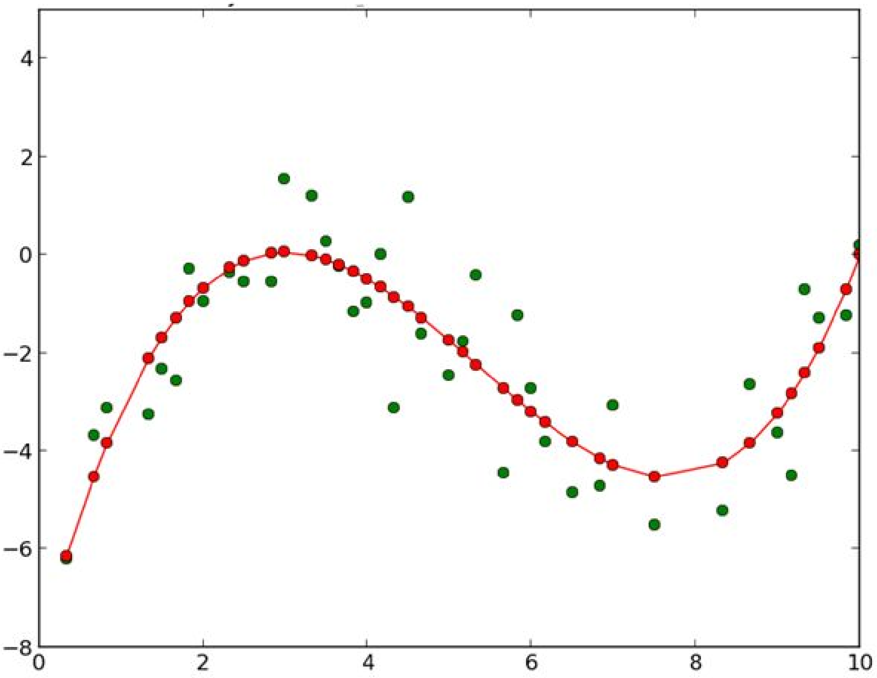


Fig.6.4.2 Bayesian curve fitting with a set of data points

Bayesian inference derives the posterior probability as a consequence of two antecedents, a prior probability and a "likelihood function". Bayesian inference computes the posterior probability according to Bayes' theorem:

$P(Y \mid X) =$, where $p(X) = \sum p(X|Y)\, p(Y)$

The physician can then use this analysis to determine whether the patient's progress is as expected.

7 Plan of Work

7.1 Product Ownership

Below is our product ownership table. This has changed slightly since the initial report and is likely to be altered as we continue throughout the process.

Tasks	Aymen	George	Tahiya	Himabindu	Tina	Ramya
Customer Registration			X	X		X
Data Capture and Storage	X	X				
Data Processing	X	X	X	X		
Web Development			X		X	
Data management				X		X
Sending data to Android server					X	X
Non functional features	X	X			X	
Documentation	X	X	X	X	X	X

7.2 Completed tasks

7.2.1 Project Management and Documentation

As a group much of our time has been devoted to completing documentation required for the class assignments. So far we have finished the proposal and parts 1 and 2 of report 1. The submission of this version will fulfill the requirements for the full report 1. Additionally, we have worked on the project management portion by mapping out task assignments. As a group we have all contributed to these sections. We have meet weekly in person or by What's app to delegate tasks and share information about our current coding status. The bulleted list outlines the non-coding tasks that we have completed:

- Proposal - **All**
- Report 1, Part 1 - **All**
- Report 1, Part 2 - **All**
- Coordination of team meetings - **Tahiya**
- Delegation of report sections - **All**
- Delegation of coding assignments - **All**

- Gantt chart - **Tina**

7.2.2 Software Coding

We have also made some progress on coding. Below is a list of some of the tasks that have been completed for the phone app:

- Review and practiced with test code in android studio - **All**

Customer Registration

- Created the mobile application with the basic login features - **Ramya, Hima**

Data capture and storage

- Created code to obtain motion information from phone sensors - **Aymen, George**
- Added algorithms to calculate number of steps taken - **Aymen**, **George**

Data Management

- Created database tables to store patient/user information and their records - **Ramya, Hima**

Non-functional tasks

- Created user interface for customer registration - **Ramya, Hima**
- Created user interface that allows access to different pages - **Tahiya, Ramya, Hima** (i.e. login/registration, profile information)

7.3 Short Term Projected tasks

In the next few weeks we will prepare the documentation and demonstration for the class assignments. The list below details the tasks and projected dates for these items:

- Report 2, Part 1 (Interaction diagrams) - **Oct. 15**
- Report 2, Part 2 (Class Diagram & System Architecture) - **Oct. 22**
- Report 2, Full - **Oct. 29**
- Demo 1 **- Nov. 1**

As far as coding we will work of the following tasks in the next weeks:

Web Application

- Create base structure for web application - **Oct. 18**
- Add user registration page - **Oct. 27**
- Add user interface for registration page - **Oct. 31**

Android Application

- Add user interface for customer registration page - **Oct 16**
- Add profile pages - **Oct 24**
- Add user interface for the profile page - **Oct 30**

- Combine code sections for customer registration and motion sensors - **Oct 31**

7.4 Project Gantt Chart

Additional tasks, projected dates, and task assignees are included in the Gantt chart in available in Microsoft project. We discovered that the software does not easily allow users to output in a format that would be easily readable in this report. As a result, we added the Microsoft Project PDF output as Appendix A at the end of this report.

8 Project Management

8.1 Basic Management

Team Leader: Tahiya
Primary Communication Method: WhatsApp
Meeting Time: Mondays, 5:00 pm to 6:30 pm
Meeting Location: Alexander LIbrary
Document Repository:
https://drive.google.com/drive/folders/0B0NuBl5TDP7_N2RaRk9BaUEyOTA?usp=sharing
Project Repository: https://github.com/karahbit/RU_Healthy

9 References

1) http://answers.google.com/answers/threadview?id=758572%22

2) http://www.mayoclinic.org/healthy-lifestyle/fitness/in-depth/exercise/art-20048389

3) https://www.cbsnews.com/news/cdc-80-percent-of-american-adults-dont-get-recommended-exercise/

4) http://www.upedu.org/process/gcncpt/co_req.htm

5) https://en.wikipedia.org/wiki/FURPS

6) https://www.merriam-webster.com/dictionary/obesity

7) https://www.tomsguide.com/us/pictures-story/482-best-fitness-apps.html#s10

8) https://www.cdc.gov/physicalactivity/basics/adults/index.htm

9) Sullivan, Alycia N. and Lachman, Margie E. (2017), Kostkova, Patty, ed., "Behavior Change with Fitness Technology in Sedentary Adults: A Review of the Evidence for Increasing Physical Activity", Front Public Health 4: 289, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5225122/

10) National Center for Health Statistics. Health, United States, 2016: With Chartbook on Long-term Trends in Health. Hyattsville, MD. 2017.

11) http://myphonefactor.in/2012/04/sensors-used-in-a-smartphone/

12) https://healthyceleb.com/wp-content/uploads/2012/10/Health-app-android-iOS.jpeg

13) https://www.google.com/search?biw=1240&bih=572&tbm=isch&sa=1&q=health+apps&oq=he&gs_l=psy-ab.3.0.0i67k1l4.73873.76835.0.78238.16.13.1.0.0.0.264.1860.0j5j4.10.0....0...1.1.64.psy-ab..10.5.874.0..0.147.QsrScBdLFfs#imgrc=C6IWjX4kIqVArM:

14) https://www.wrike.com/project-management-guide/faq/what-is-a-stakeholder-in-project-management/

15) Ivan Marsic, (2012). Written at Rutgers University, New Brunswick, New Jersey, *Software Engineering*, http://www.ece.rutgers.edu/~marsic/books/SE/

16) https://www.cmcrossroads.com/sites/default/files/article/file/2013/XUS234033087file1_0.pdf

17) https://en.wikipedia.org/wiki/System_sequence_diagram

18) http://photos-ak.sparkpeople.com/7/0/b707669348.jpg

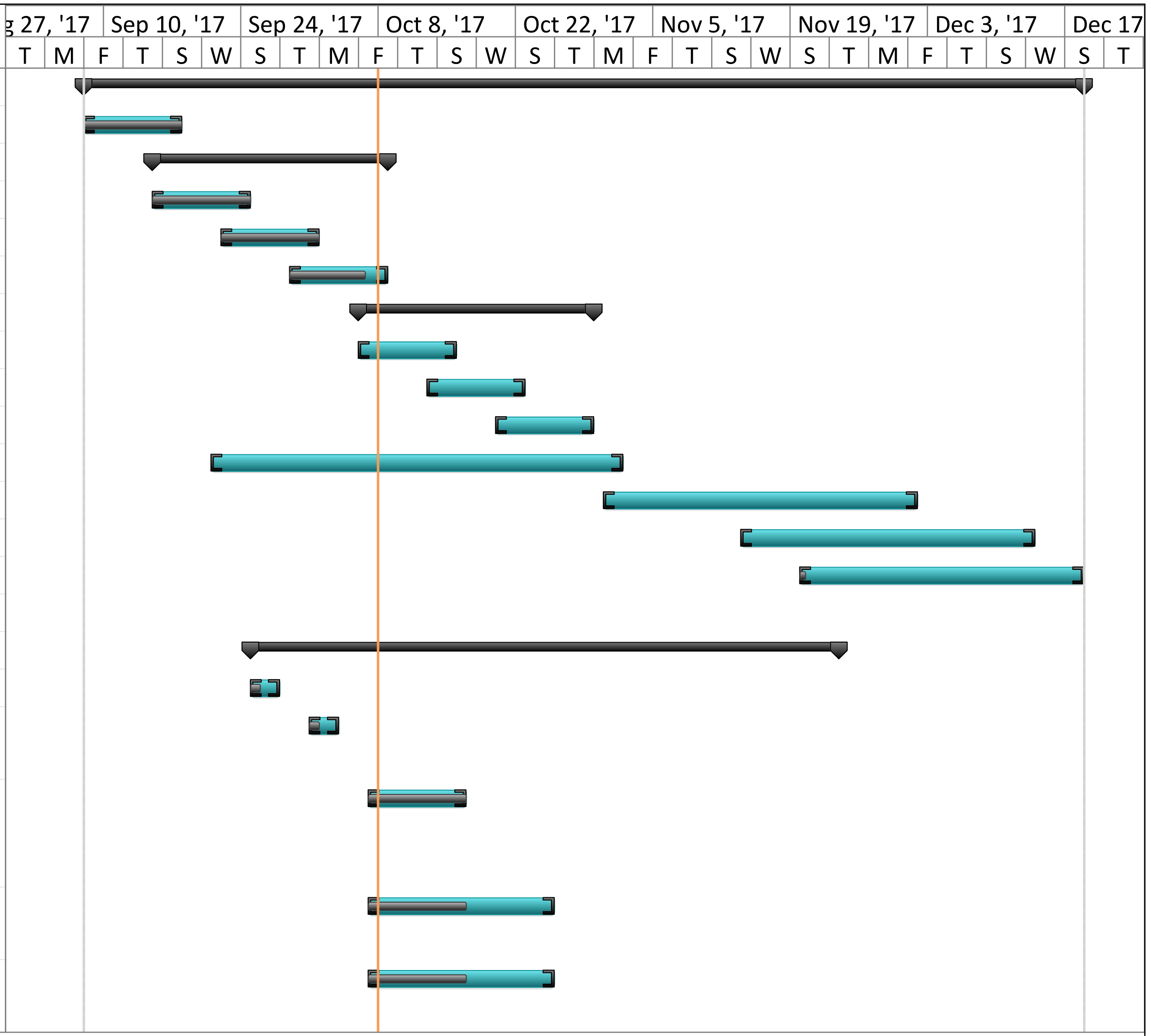
19) http://stg-tud.github.io/eise/WS14-EiSE-12-System_Sequence_Diagrams.pd

20) Cameron et al, (2004), "Energy Expenditure of Walking and Running", Medicine & Science in Sport & Exercise.

Appendix A

This Appendix contains the Gnatt chart of the project tasks for the RU Healhy? App

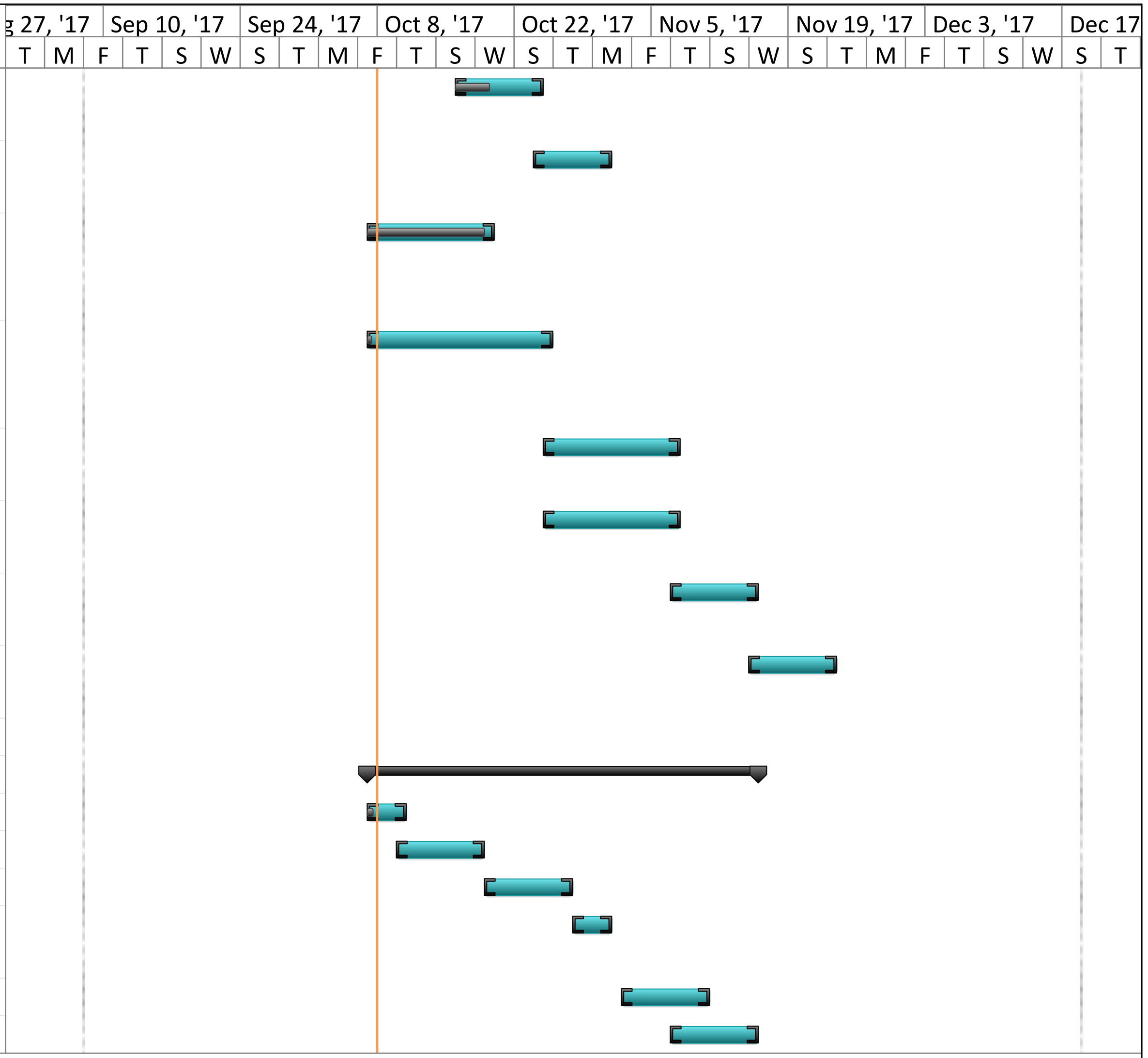
ID	Task Name	Duration	Start	Finish	Notes	% Complete
1	**Class Assignments**	**72 days**	**Fri 9/8/17**	**Mon 12/18/17**	**All**	**19%**
2	Proposal	7 days	Fri 9/8/17	Sun 9/17/17	All	100%
3	**Report 1**	**16 days**	**Fri 9/15/17**	**Sun 10/8/17**	**All**	**95%**
4	Part 1	7 days	Fri 9/15/17	Sun 9/24/17	All	100%
5	Part 2	7 days	Fri 9/22/17	Sun 10/1/17	All	100%
6	Full Report	7 days	Fri 9/29/17	Sun 10/8/17	All	85%
7	**Report 2**	**16 days**	**Fri 10/6/17**	**Sun 10/29/17**	**All**	**0%**
8	Part 1	7 days	Fri 10/6/17	Sun 10/15/17	All	0%
9	Part 2	7 days	Fri 10/13/17	Sun 10/22/17	All	0%
10	Full Report	7 days	Fri 10/20/17	Sun 10/29/17	All	0%
11	Demo 1	30 days	Thu 9/21/17	Wed 11/1/17	All	0%
12	**Report 3**	24 days	Tue 10/31/17	Fri 12/1/17	All	0%
13	Demo 2	22 days	Tue 11/14/17	Wed 12/13/17	All	0%
14	Electronic Project Archive	21 days	Mon 11/20/17	Mon 12/18/17	All	3%
15						
16	**Android App**	**44 days**	**Mon 9/25/17**	**Thu 11/23/17**		**32%**
17	Research Similar Apps	3 days	Mon 9/25/17	Wed 9/27/17	All	33%
18	Create base structure for app in android studio	3 days	Sun 10/1/17	Tue 10/3/17	All	33%
19	Create Mobile app with basic features (Login, registration, etc.)	7 days	Sat 10/7/17	Mon 10/16/17	Ramya, Hima	100%
20	Create database to store patient info	14 days	Sat 10/7/17	Wed 10/25/17	Ramya, Hima	50%
21	Add user interface for customer registration	14 days	Sat 10/7/17	Wed 10/25/17	Ramya, Hima, Tahiya	50%



Project: Software Engineering_RU
Date: Sun 10/8/17

Task	▬	External Tasks	▬	Manual Task	▬	Finish-only	]
Split	┈	External Milestone	◆	Duration-only	▬	Deadline	⇩
Milestone	◆	Inactive Task	▭	Manual Summary Rollup	▬	Progress	▬
Summary	▬	Inactive Milestone	◇	Manual Summary	▬		
Project Summary	▬	Inactive Summary	▭	Start-only	[		

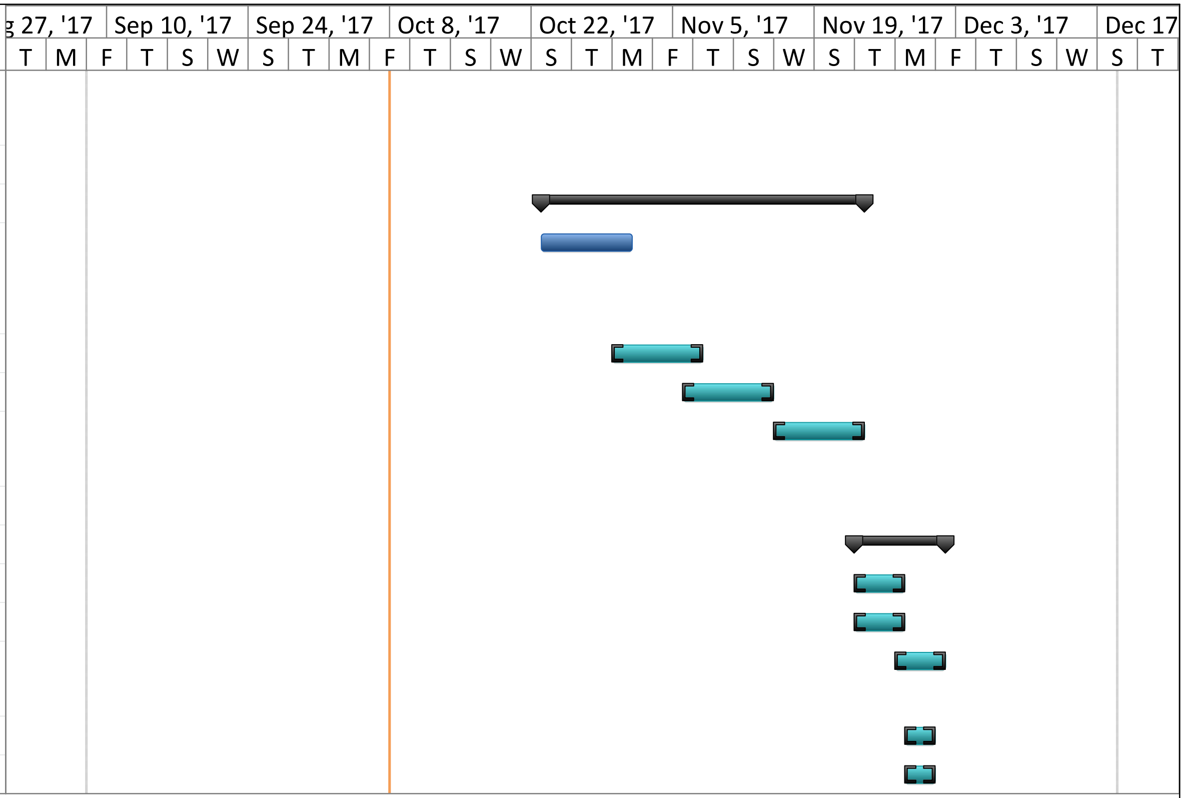
ID	Task Name	Duration	Start	Finish	Notes	% Complete
22	Add profile pages	7 days	Mon 10/16/17	Tue 10/24/17	Ramya, Hima, Tahiya	50%
23	Add user interface for profile pages	6 days	Tue 10/24/17	Tue 10/31/17	Ramya, Hima, Tahiya	0%
24	Create functionality to detect motion from phone sensors	10 days	Sat 10/7/17	Thu 10/19/17	Aymen	90%
25	Create functionality detect heart rate from phone sensors	14 days	Sat 10/7/17	Wed 10/25/17	George	2%
26	Create algorithms to process and store heart rate data	10 days	Wed 10/25/17	Tue 11/7/17	Aymen, George	0%
27	Create algorithms to process and store motion rate data	10 days	Wed 10/25/17	Tue 11/7/17	Aymen, George	0%
28	Add functionality for exercise scheduling	7 days	Tue 11/7/17	Wed 11/15/17	Tina, Aymen, George	0%
29	Add functionaility for notifications	7 days	Wed 11/15/17	Thu 11/23/17	Tina, Aymen, George	0%
30						
31	**Web Application**	**28 days?**	**Sat 10/7/17**	**Wed 11/15/17**		**2%**
32	Reseach similar applications	3 days	Sat 10/7/17	Tue 10/10/17	Tina, Tahiya	20%
33	Create base structure	7 days	Tue 10/10/17	Wed 10/18/17	Tina	0%
34	Add user registration page	7 days	Thu 10/19/17	Fri 10/27/17	Tina	0%
35	Add user interface for registration	3 days	Sat 10/28/17	Tue 10/31/17	Tina, Tahiya	0%
36	Add profile view page	7 days	Thu 11/2/17	Fri 11/10/17	Tina, Tahiya	0%
37	Add data view capability	7 days	Tue 11/7/17	Wed 11/15/17	Tina, Tahiya	0%



Project: Software Engineering_RU
Date: Sun 10/8/17

Task	▬	External Tasks	▬	Manual Task	▬	Finish-only	]
Split	┈┈┈┈	External Milestone	◆	Duration-only	▬	Deadline	⇩
Milestone	◆	Inactive Task	▭	Manual Summary Rollup	▬	Progress	▬
Summary	▼▬▼	Inactive Milestone	◇	Manual Summary	▼▬▼		
Project Summary	▼▬▼	Inactive Summary	▽▭▽	Start-only	[		

ID	Task Name	Duration	Start	Finish	Notes	% Complete
38						0%
39						
40	**Socket Programming**	**24 days**	**Mon 10/23/1**	**Thu 11/23/17**		**0%**
41	Combine code sections for customer registration and motion sensors	7 days	Mon 10/23/17	Tue 10/31/17	Ramya, Hima	0%
42	Incorporate heart rate	7 days	Mon 10/30/17	Tue 11/7/17	Aymen, Georg	0%
43	Link web app to android app	7 days	Mon 11/6/17	Tue 11/14/17	Tina, Tahiya	0%
44	Incorporate other code sections	7 days	Wed 11/15/17	Thu 11/23/17	All	0%
45						
46	**Quality Testing**	**7 days**	**Thu 11/23/17**	**Fri 12/1/17**		**0%**
47	Test phone app	3 days	Thu 11/23/17	Mon 11/27/17	TBD	0%
48	Test web app	3 days	Thu 11/23/17	Mon 11/27/17	TBD	0%
49	Make adjustments to the applications	5 days	Mon 11/27/17	Fri 12/1/17	All	0%
50	Retest phone app	3 days	Tue 11/28/17	Thu 11/30/17	TBD	0%
51	Retest web app	3 days	Tue 11/28/17	Thu 11/30/17	TBD	0%



Task	▬	External Tasks	▬	Manual Task	▬	Finish-only	]
Split	┅	External Milestone	◆	Duration-only	▬	Deadline	⇩
Milestone	◆	Inactive Task	▭	Manual Summary Rollup	▬	Progress	▬
Summary	▬	Inactive Milestone	◇	Manual Summary	▬		
Project Summary	▬	Inactive Summary	▭	Start-only	[		