

Software Engineering (14:332:452)

Group 3

**Programming Project: Blockchain-Based Safe Sharing of Population
Descriptors**

Submission Date: October 13, 2019

**Team Members: Sasan Hakimzadeh, Pradyumna Rao, Nithyasree Natarajan,
Jack Dulin, Shruthi Sureshkrishnan, Nithya Kandappan, Breanna Higgins,
Sean Kearns**

Table of Contents

Responsibility Matrix	3
Customer Problem Statement	5
Problem Statement	5
Glossary of Terms	8
System Requirements	9
Enumerated Functional Requirements	9
Enumerated Nonfunctional Requirements	10
User Interface Requirements	11
Functional Requirements Specification	12
Stakeholders	12
Actors and Goals	12
Use Cases	13
Use Case Descriptions	13
Use Case Diagram	16
Traceability Matrix	17
Fully-Dressed Description	18
System Sequence Diagrams	20
User Interface Specification	23
Preliminary Design	23
User Effort Estimation	29
Domain Analysis	35
Concept Definitions	35
Association Definitions	37
Attribute Definitions	39
Domain Model	40
Traceability Matrix	41
System Operation Contracts	42
Project Size Estimation	44
Plan of Work	47
Plan of Work	47
Product Ownership	48
References	50

Responsibility Matrix

	Nithyasree	Pradyumna	Sasan	Shruthi	Nithya	Breanna	Sean	Jack
Project Management	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Customer Problem Statement	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Glossary of Terms	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
System Requirements	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Stakeholders	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Actors & Goals	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Use Cases & Detailed Use Cases	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Use Case Diagram	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Traceability Matrix	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
System Sequence Diagrams	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
UI Mockup	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Concept Definitions	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Association Definitions	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Attribute Definitions	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%

Traceability Matrix	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
System Operation Contracts	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Project Size Estimation	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Plan of Work	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
References	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%

All team members contributed equally.

Section 1: Customer Problem Statement

a. Problem Statement

The customers find that it is difficult to stay in shape and lead healthy lives. With everyone living busy lives focused on education, work, and families, one's health is not given a priority. As a result, people's well-being and fitness levels deteriorate over time. Therefore, the customers want a web-based application that provides its users with a consolidated, one-stop platform to input all of their health-related data. Such data would include users' hourly heart rate, amount of hours slept, their daily activities, their daily activities, etc.

There are many aspects that go into making sure an individual is healthy both physically and mentally, especially since the two go hand in hand. Although customers would like to be healthy, they find it overwhelming to keep track of all the different aspects of good health. For instance, it is difficult to remember everything one ate throughout the day, and then also recall it the next day or even later in the week. Therefore, users would want the application to track several different features, including heart rate, sleep, food, weight, activity, and exercise. This way, people can use the application to review the various aspects for the current week as well as look at how much they have progressed throughout the weeks.

Being able to look at data from previous weeks and months is very important to customers because they want to be able to compare the data gathered from various weeks. This will also allow users to see the improvements in their fitness. Many times it is difficult to see progress in one's health just by looking in the mirror. This results in being discouraged from working out and eating healthy. However, having the physical data from each week will support the developments being made and provide the encouragement needed to continue with one's regimen or make the necessary changes to gain the results desired.

Users want to be able to create a profile that will have information about them, such as date of birth, sex, height, and weight. They can then input information for each of the different aspects, every day. For the heart rate aspect, customers want to upload their heart rate and check if it is at a healthy level, based on age and other information. For the sleep aspect, users want to put in the time they went to bed and the time they got up to see how many hours of sleep they got per night. The application would then give feedback to the person, telling him/her to get more or less sleep, based on the person's age.

For the nutrition aspect, users want to input everything they ate and drank for the day for breakfast, lunch, and dinner as well as any snacks eaten throughout the day. This will allow customers to keep track of what they ate, if it kept them full, if they liked it, and if they would eat it again. For example, if they reviewed what they ate for the day and felt that they were still hungry throughout the day, they know to increase the food intake. However, if they felt too full throughout the day, then they know to eat less food. Users would also like to have a calorie option, so that when they input the food, the amount of calories shows up, or a person can manually input the amount of calories for that food. This way, those who calorie count can know they are on track for their daily amount of calories.

For the weight aspect, customers want to input their daily weight so that they can check if they are staying on track to reach their goal of losing weight, gaining weight, or staying the same weight. For the activity aspect, they can for instance, upload their daily step count. For the exercise aspect, users want to input each of the exercises that were done that day, how long each exercise was done, and how many times each exercise was done. The application should then be able to show how many calories were lost from doing these exercises. After inputting data every

day, users also want to see how they stand when compared to the average population within a similar demographic, such as age, gender, race, etc. This way, users can track their relative standing on a secure platform. Many times, the statistics provided by the government are outdated and too general, so having reliable information in real time is important for users.

Customers want to use the application to inspire themselves to set personal goals such as reaching a higher step count the following week or drinking more water tomorrow. As a result, users need pop-up messages to remind them of their weekly tasks and to notify them if they are close to reaching their goal. Users also want a message to congratulate them once they have reached their goal, so they continue to stay motivated. Users also need a daily reminder system to remind the user to input their data for the day and should be customizable to when and how many times a day users get reminders.

Users want an application that is free and easy to use. A web-based application enables the user to input data from any device they own. The customers also want this to be accessible for people of all ages. They want this application to be simple to use so that the elderly can also use it to track their health. Applications on phones are sometimes more confusing to use for older people, so a web-based application, which has bigger words and icons, will allow people of all ages to watch over their well-being. Users will also be able to use the pop-up messages to remind them to take their daily medication or do their other daily health related tasks. Customers would also like to use this application to make sure that they are healthy in general. If there is a potential abnormality based on the data uploaded by the user, then the users want a pop-up message that urges them to go visit a doctor.

b. Glossary of Terms

Encryption: The method by which plaintext or any other type of data is converted from a readable form to an encoded version that can only be decoded by another entity if they have access to a decryption key

Blockchain: A mathematical structure for storing digital transactions (or data) in an immutable, peer-to-peer ledger that is incredibly difficult to fake and yet remains accessible to everyone.

Primary Key - A specific choice of a minimal set of attributes that uniquely specify a row in a database. In our case, the username would be the primary key, as it needs to be unique per user so as to not duplicate accounts.

Health Profile - A holistic set of physiological measurements and features such as weight, height, BMI, caloric intake, calories burned, heart rate, sleep, etc. These set comprised of these features make up a single user's health profile and can be used as a data point.

Tracking Device - A wearable health tracker that may be part of a "smart" watch, such as a FitBit, Apple Watch, Samsung Gear. It should be able to track all the basic features that can be tracked automatically under the Health Profile. It should also have support to manually input the features that cannot be automatically tracked, such as caloric intake, weight, and height.

Health Rank - A numerical ranking statistic that serves to show a user's parameters in comparison to a given population. There may be multiple health ranks for any given user based on the population against whom he/she is querying. I.E. A user may have a different rank against only other male users, only other female users, only local users, etc

Section 2: System Requirements

a. Enumerated Functional Requirements

Label	Priority Weight (1-10)	Description
REQ-1	10	The system must ensure that the communal health data must be encrypted and secured.
REQ-2	5	The system must ensure that there is a daily reminder for the user to put in their data.
REQ-3	8	The system must ensure that users must not be able to access other user data other than the general trends from the population
REQ-4	9	The system must ensure that users should be able to log into their account using a username as a primary key, and a secure password which adheres to certain security measures such as length and special character inclusion.
REQ-5	7	The system must ensure that users must be able to compare themselves against others using real-time data that is constantly updated.
REQ-6	6	Users must be able to see a page where they can analyze visualizations (graphs & charts) which show their data points compared to their demographic.
REQ-7	5	There must be a max threshold for certain parameters so as to avoid fraudulent input of data which would skew calculations/trends.

b. Enumerated Nonfunctional Requirements

Label	Priority Weight (1-10)	Description
REQ-8	5	Functionality: Users should be able to set goals for themselves to be able to achieve (number of steps taken, miles walked, etc.)
REQ-9	3	Functionality: Users can “friend” others on the platform and share their results with their network, securely and privately.
REQ-10	3	Functionality: Users can personalize their profiles by uploading images of themselves as their thumbnail.
REQ-11	4	Functionality: Users should be able to post their progress to their social media accounts (Facebook, Twitter, etc.)
REQ-12	8	Performance: The application shall be responsive to commands/clicks with a reasonable amount of loading time, based on the information being refreshed or loaded.
REQ-13	8	Performance: The application shall optimize data storage space to improve data retrieval speeds.
REQ-14	8	Usability: The application shall be human-centric and usability should be prioritized, with an emphasis on making the front-end application easy to use and navigate.
REQ-15	7	Supportability: The application shall be easy to maintain and update, with code written clearly with comments to ensure other developers can read and understand the functionality for future updates, bug fixes, etc.
REQ-16	7	Supportability: The application shall be reliable with little to no failure or crashing. If the application does crash, disaster recovery should be set in place to ensure all data remains and is consistent. The distributed and non-centralized storage of the data helps to ensure that a breach or malfunction in one location will not affect data stored in another location.
REQ-17	5	Scalability: Although the application will be on a relatively small scale for the purposes of this project, the system shall be able to be scaled up to support thousands of users on it at the same time.

c. User Interface Requirements

Label	Priority Weight (1-10)	Description
REQ-19	7	The user interface should allow for users to input data and access their profiles from both mobile devices and desktops.
REQ-20	6	The graphs and charts displayed to the user as visual aid should be interactive and show smaller breakdown information when clicked.
REQ-21	8	There should be a friendly, welcoming login page that should have two (2) fields for username and password, as well as a forgot password/reset password link.
REQ-22	4	There should be a friendly, reminder text message sent to the user's cell phone number at 10:00 P.M. reminding them to input their daily data if they have not already. Pop-up notifications are also an alternative.
REQ-23	9	There should be a menu that includes link to all the parts of the app, such as home, user profile, account and security, daily data input, health ranking and population querying, etc

Section 3: Functional Requirements Specification

a. Stakeholders

The following lists stakeholders who may have an interest in this application:

- General public: People who want to compare their fitness levels against others in the same demographic or want to know if they're in the range of what is considered healthy
 - This could apply to people who are already healthy and want to stay healthy, or those who are working toward obtaining a healthier lifestyle and want to know what that is
- Organizations that promote health and fitness, such as gyms
 - They could use this system as motivation and/or targets for their clients. They could also potentially use it to see the general trend of their clients
- Doctors could use this system to check if their patients are staying healthy
 - Doctors could check if their patients are following the regimen prescribed to them and if anything needs to be altered in order to prescribe a better course of action
- Research-based organizations: They could use the data to find potential patterns in certain populations
 - Patterns can show common occurrences that happen among different groups based on sex, age, or race, for instance
 - These common occurrences can identify norms for these groups or find health issues that need to be resolved
- Companies that sell medical devices and health care equipment
 - Companies can be sponsors to advertise their products on the website so that users can view these advertisements while inputting their data

b. Actors & Goals

The following lists all participating actors in the application and their goals

- User - The goal of the user is to be able to have a seamless and intuitive experience on the application where they can track their fitness metrics, compare it to the general population, set goals for themselves, build a social network, and visualize charts that show where they stand amongst others within their demographic.
- Database/Blockchain/Backend - The goal of the backend database & incorporated blockchain technology is to be able to store the user information as it changes in the app, and to ensure that all data is as secure as possible.
- UI - The goal of the UI is to offer the user a seamless experience on the application, without the need to question where to go in order to complete a task.
- Wearable Devices - The goal of the devices is to feed the data that we are tracking for the entire population. The data metrics we record for trends in various population demographics will be based on what is provided historically through these devices.

c. Use Cases

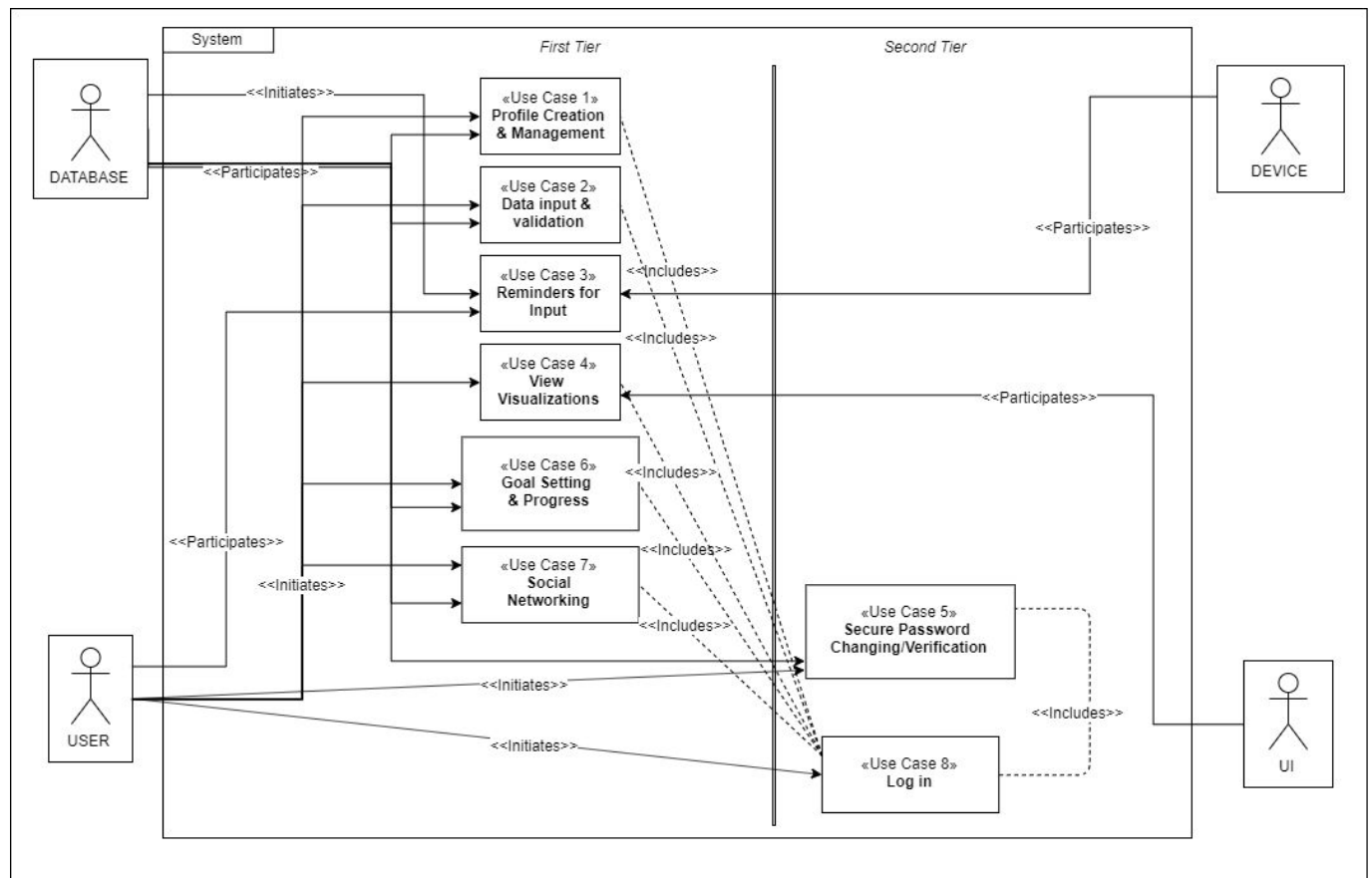
i. Casual Description

Actors and Goals	Use Case	Relevant Requirements	UC Number
Actor: User (Initiating), DBMS (Participating) Goal: Users can create & update their profiles as desired, and the DBMS stores these updates as they occur	Profile creation & management	REQ-10, REQ-19, REQ-13, REQ-16	1
Actor: for Data Input: User (Initiating), DBMS (Participating) Goal: User is able to input daily health metrics for analysis Actor: for Data range Validation: Database Management System (Initiating Actor) Goal: All of the data entered into the database is accurate/not skewed to ensure accurate visualizations and comparisons.	Data input & validation	REQ-7, REQ-1, REQ-12, REQ-13, REQ-14, REQ-15, REQ-18	2
Actor: Database (Initiating) User (Participating) Device (Participating) Goal: Ensure users are engaged with the platform Try to obtain as full of a data set as possible to avoid sparse/empty rows that would	Reminders for input	REQ-2,REQ-22	3

skew visualizations/calculations.			
Actor: User (Initiating) UI(Participating) Goal: To provide a friendly user interface. View statistical data in user-friendly graphical representation.	View visualizations	REQ-6, REQ-20, REQ-5, REQ-12, REQ-14, REQ-18	4
Actor: User (Initiating), DBMS (Participating) Goal: Users should be able to securely change their password (either through email link or verification questions) and the database should be able to reflect that change in the users' profile once the change has been made.	Secure Password Changing/Verification	REQ-21	5
Actor: User (Initiating), DBMS (Participating) Goal: Motivate the user to keep putting in data so that there is an abundance of it to ensure statistical accuracy due to increased sampling. Meanwhile, the DBMS should be able to store these goals as they are created by the user.	Goal Setting and Progress	REQ-8, REQ-14, REQ-16, REQ-18	6

<p>Actor: User (Initiating), DBMS (Participating)</p> <p>Goal: Users are able to curate their own user experience by sending friend requests, blocking users, denying requests, and sharing progress on social media to enhance user engagement with the application as well as increase motivation. Meanwhile, the database should keep records of social networking interactions and social media pages that are linked to each user.</p>	<p>Social Networking</p> <ul style="list-style-type: none"> -Friending/Unfriending -Blocking -Denying Requests -Share progress on social media 	<p>REQ-11, REQ-9, REQ-10, REQ-17</p>	<p>7</p>
<p>Actor: User (Initiating)</p> <p>Goal: The user is able to securely log into his/her account with the credentials he/she set with the account.</p>	<p>Log In</p>	<p>REQ-21, REQ-1, REQ-3, REQ-4, REQ-15</p>	<p>8</p>

ii) Use Case Diagram



iii. Traceability Matrix

The following traceability matrix shows the ability of the identified use cases to achieve all requirements set forth for the application. With the matched priority weighting of each related requirement, the matrix determines the most essential use cases. The three largest use cases are highlighted as they are the most critical to the system.

Priority Weight	Requirement	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8
5	REQ-1		x						x
10	REQ-2			x					
8	REQ-3								x
9	REQ-4								x
7	REQ-5				x				
6	REQ-6				x				
5	REQ-7		x						
5	REQ-8						x		
3	REQ-9							x	
3	REQ-10	x						x	
4	REQ-11							x	
8	REQ-12		x		x				
8	REQ-13	x	x						
8	REQ-14		x		x		x		
7	REQ-15		x						x
7	REQ-16	x					x		
5	REQ-17							x	
6	REQ-18		x		x		x		
7	REQ-19	x							
6	REQ-20				x				
8	REQ-21					x			x
4	REQ-22			x					

<i>Max PW</i>	8	8	10	8	6	8	5	9
<i>Max PW</i>	25	47	14	41	6	26	15	37

iv. Detailed Use Case Descriptions

Based on the given traceability matrix, the following detailed use cases are described as they have the highest priority weighting.

UC-2	Data Input/Validation
Related Requirements	REQ-7, REQ-1, REQ-12, REQ-13, REQ-14, REQ-15, REQ-18
Initiating Actor	User, Database Management System (DBMS)
Participating Actors	Database Management System (DBMS)
Preconditions	The user is logged into the system.
Postconditions	There has been a successful or unsuccessful attempt of adding more data into the system.
Event Flow (Success)	<ol style="list-style-type: none"> 1. The user has tried to enter a reasonable value. 2. The data point is entered into the system.
Event Flow (Failure)	<ol style="list-style-type: none"> 1. The user has entered an unreasonable value (e.g. it is above a certain threshold). 2. The data point is rejected from the system.

UC-4	View Visualizations
Related Requirements	REQ-6, REQ-20, REQ-5, REQ-12, REQ-14, REQ-18
Initiating Actor	User
Participating Actors	Data Visualization Software, DBMS
Preconditions	The user is logged into the system.

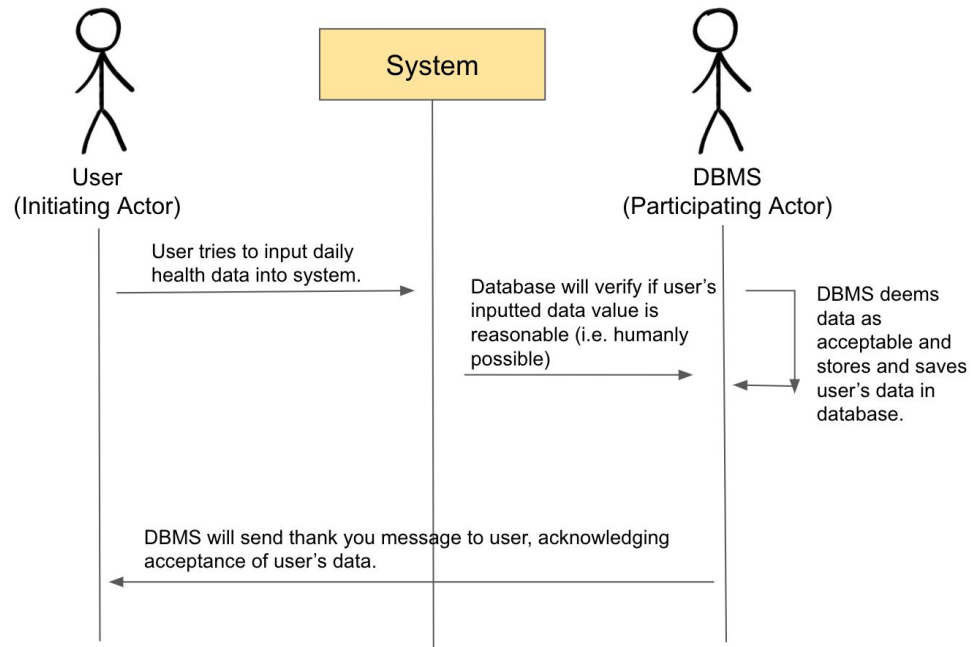
Postconditions	The user sees a collection of their data in the form of visuals.
Event Flow (Success)	<ol style="list-style-type: none"> 1. The user requests a visualization of their inputted data. 2. The data visualization software shows the user their data in a presentable way, whether through a pie chart or a bar graph.
Event Flow (Failure)	<ol style="list-style-type: none"> 1. The user requests a visual representation of their inputted data. 2. The data visualization software is unable to accomodate the user's request.

UC-8	Log In
Related Requirements	REQ-21, REQ-1, REQ-3, REQ-4, REQ-15
Initiating Actor	User
Participating Actors	Database Management System (DBMS)
Preconditions	The user is currently at the screen in which they are asked to enter his/her username and password.
Postconditions	There has been an attempt to get into the system through a correct or an incorrect username and password.
Event Flow (Success)	<ol style="list-style-type: none"> 1. The user types in the correct username and password. 2. The user is taken to the home page of our application.
Event Flow (Failure)	<ol style="list-style-type: none"> 1. The user types in the incorrect username and/or password. 2. They have another opportunity to type in the correct username and password. 3. They are asked if they have forgotten their username or password.

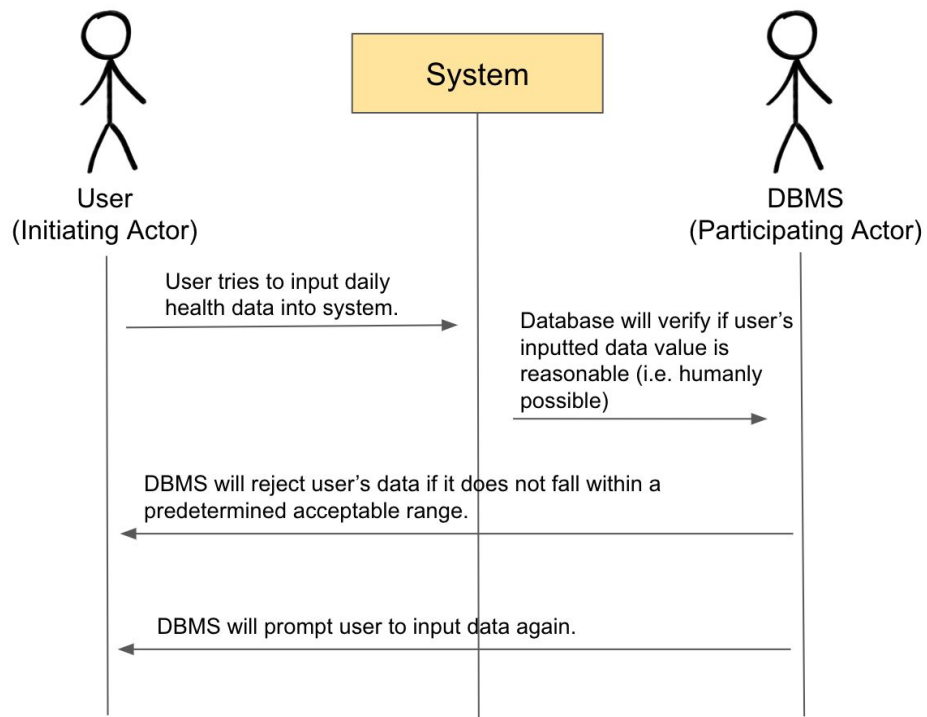
d. System Sequence Diagrams

The following diagrams model the event flows of the three fully-dressed use cases

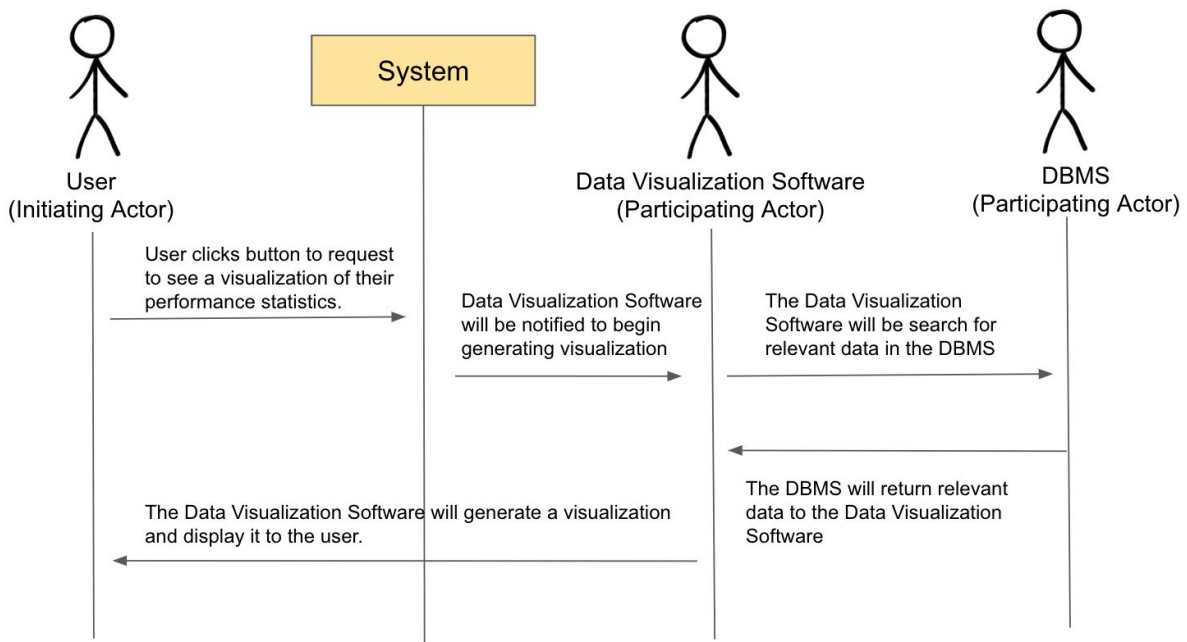
Use Case 2: Data Input / Validation (Success)



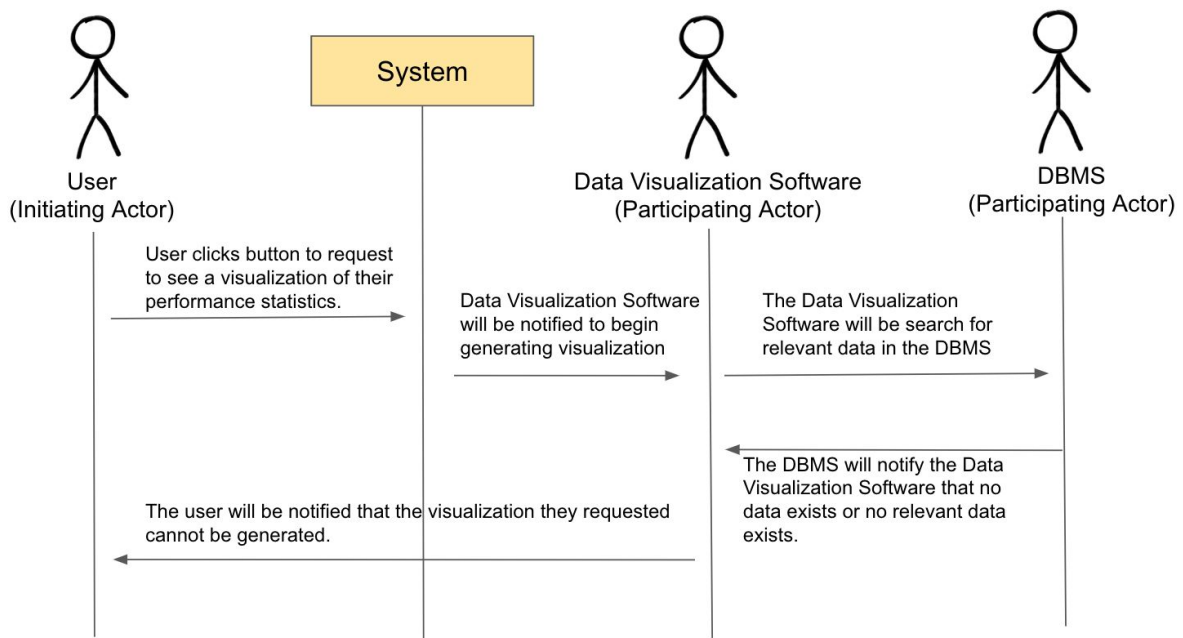
Use Case 2: Data Input / Validation (Failure)



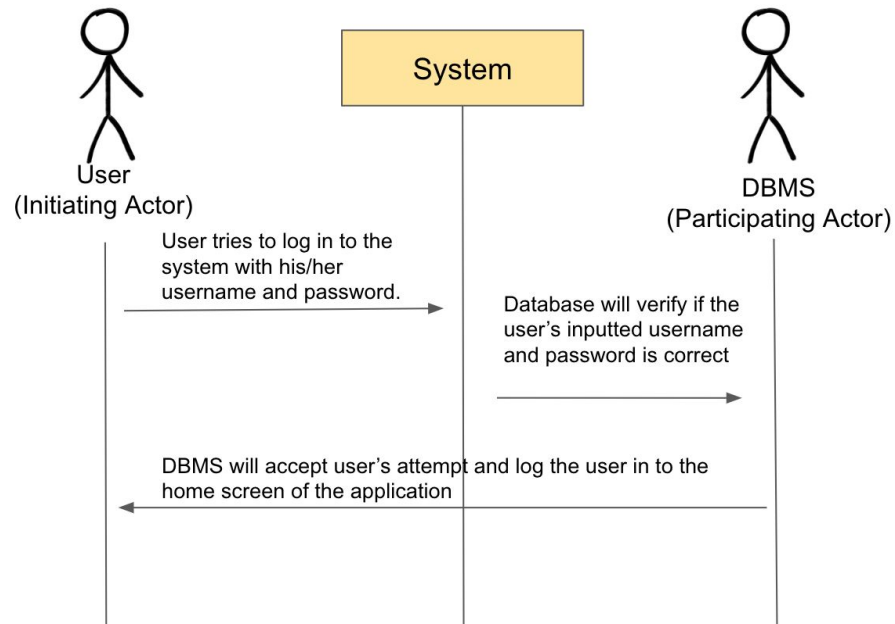
Use Case 4: View Visualizations (Success)



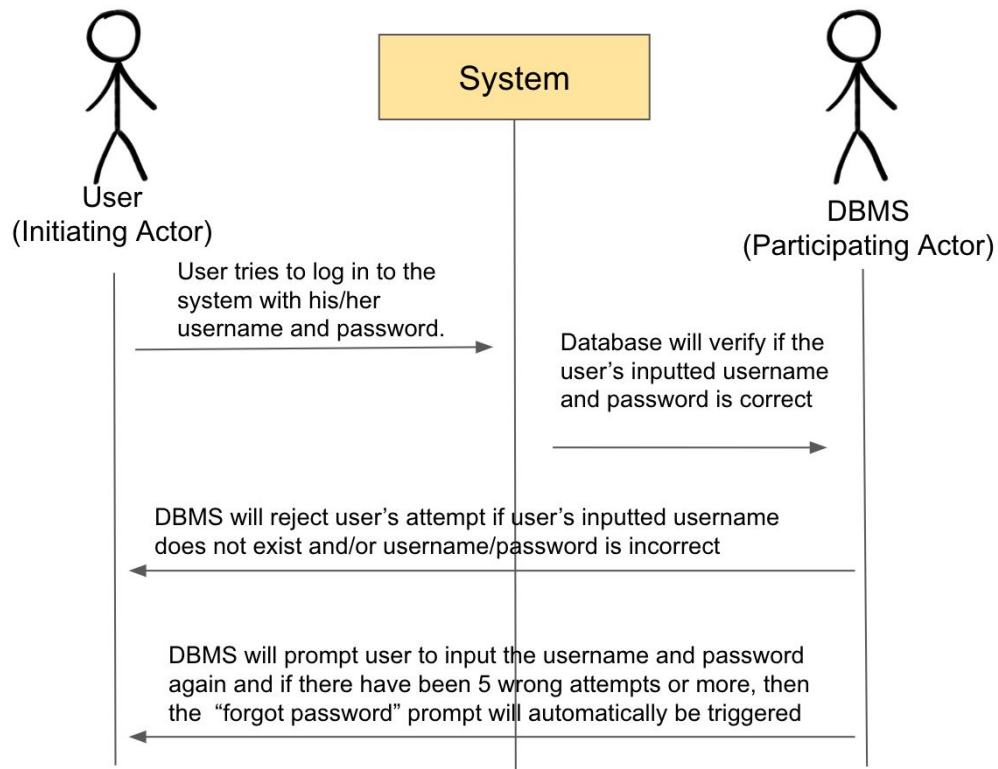
Use Case 4: View Visualizations (Failure)



Use Case 8: Log In (Success)



Use Case 8: Log In (Failure)



Section 4: User Interface Specification

a. Preliminary Design

i. Use Case 2

1. User is on the data entry page “Add Data.” He will enter any relevant information that he may wish to add for the day to continue tracking his progress. He will enter information in the appropriate fields and then submit it by clicking the “Add Data” button on the appropriate side of the screen.

Data Entry Page

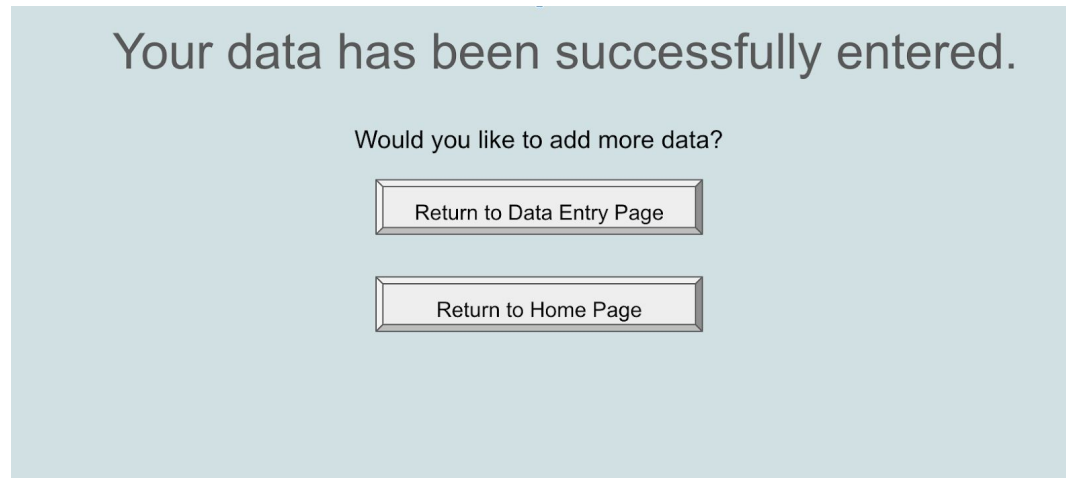
The screenshot shows a mobile application interface titled "Add Data". At the top left is a "Back" button. The interface is split into two main sections. The left section, with a dark teal background, contains input fields for "Weight:" and "Blood Pressure:", and an "Add Data" button at the bottom. The right section, with a light blue background, contains input fields for "Calories burned:" (with the value 500), "Steps:" (with the value 2000), "Distance:" (empty), and "Sleep:" (with a spinner showing 0:00). Below these fields are two "Add Data" buttons and an "Upload from device" button at the bottom.

2. The data will be processed for validity and make sure that it is simply not skewed, spam, or invalid data. If the data is recognized as invalid, it will be rejected, and the user will be notified with a pop-up notification in the application. The user will then be allowed to enter data again.

Data Entry Page

This screenshot shows the same "Add Data" page as the previous one, but with a validation error. The "Calories burned:" field now contains the value "-5". A blue pop-up notification box is centered over the form, displaying a yellow warning triangle icon, the text "Data entered is not valid.", and an "OK" button. The background interface elements, including the "Back" button, input fields, and "Add Data" buttons, are visible behind the notification.

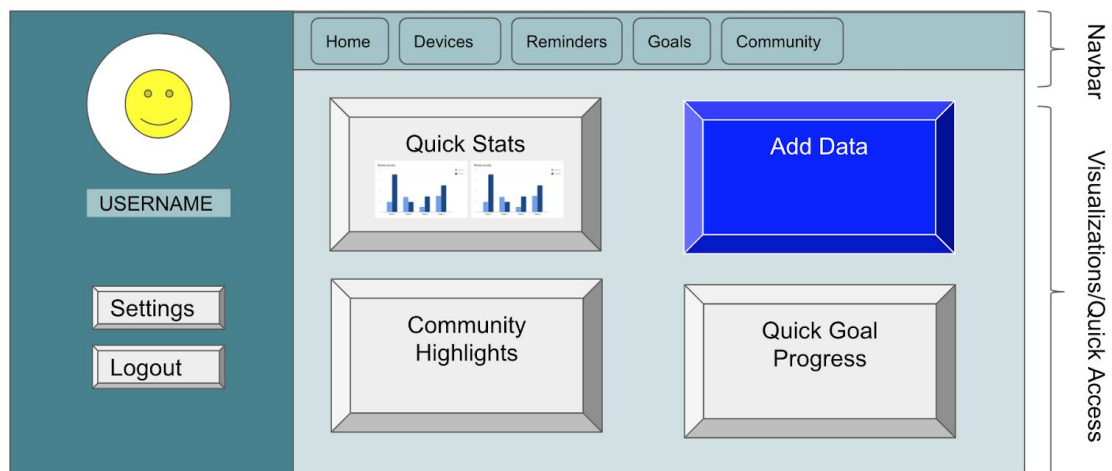
3. Once valid data is entered, the database will add it internally and associate it with the primary key, the user. A confirmation notification will appear for the user.



ii. Use Case 4

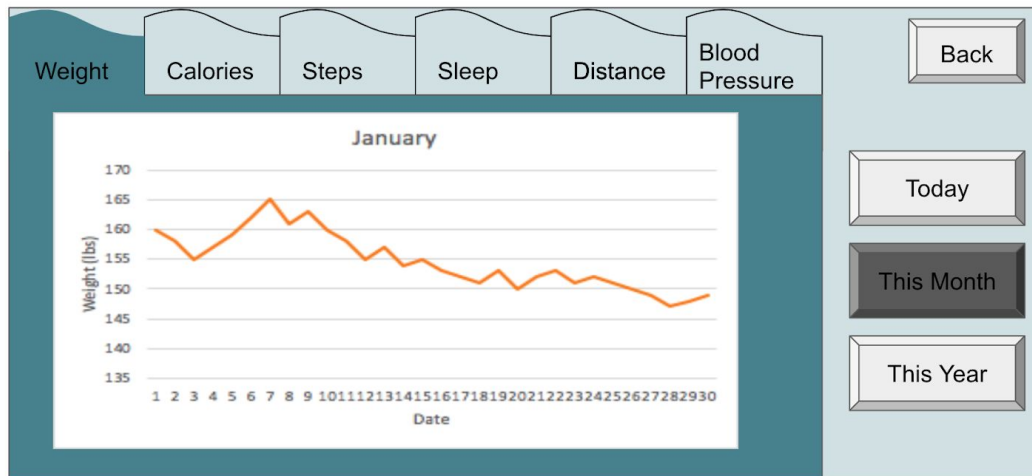
1. User will submit a request to visualize their performance and see their rankings and ratings in relation to other users or track their own progress over time. They will submit this request by pressing the “Quick Stats” button on the home page, and will not necessarily need to input numerical data. Rather, they are inputting a binary action (button press).

Home Page

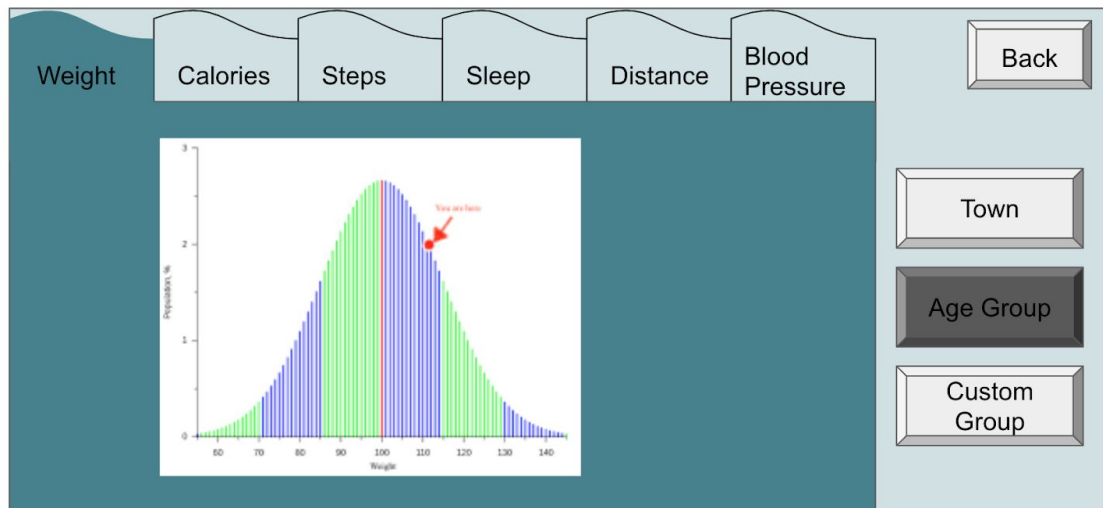


- The DB will pull all relevant information and the visualization module will format it accordingly based on what information was requested, to whom the data is in relation, how much time is in the scope of the query, etc.

User Statistics Page



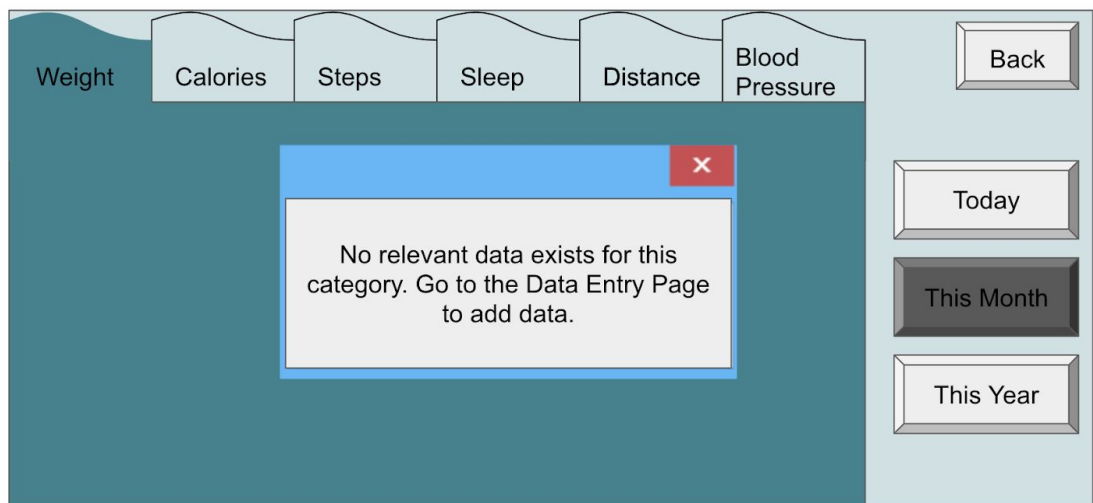
Population Statistics Page



3. The visualization will be displayed on the screen, and the user can freely look through the pictorial representations or return to the other pages of the application at their discretion. The user may also be informed no relevant data exists if that is the case.

For instance, if a user has never input data for their weight, then no relevant data would exist for that case.


User Statistics Page



iii. Use Case 8

1. User opens the application and enters his/her username in the “username” field.

Login Page

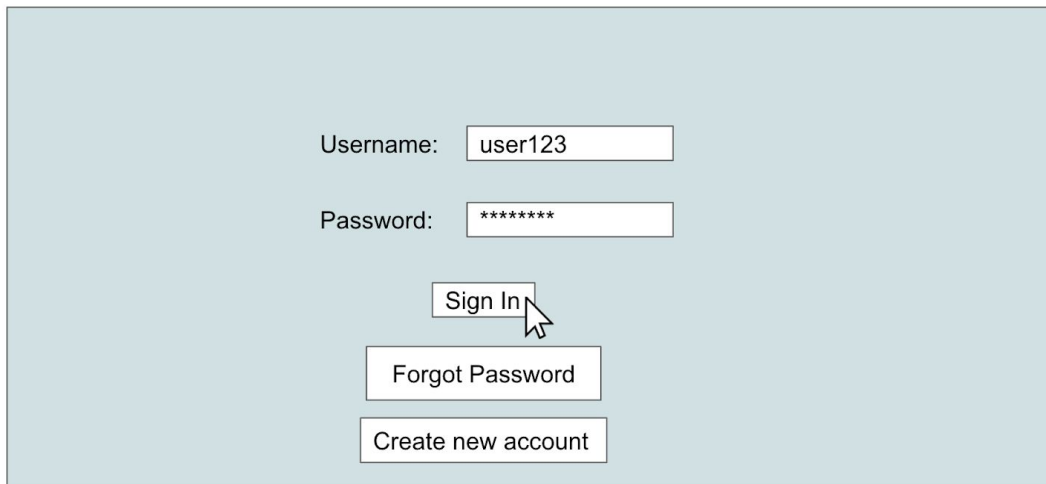


A screenshot of a login page with a light blue background. The page contains the following elements:

- Username:** A text input field containing the text "user123".
- Password:** An empty text input field.
- Sign In:** A button located below the password field.
- Forgot Password:** A button located below the "Sign In" button.
- Create new account:** A button located at the bottom of the form.

2. He then enters the password associated with that username in the “password” field. He then presses the “sign in” button.

Login Page

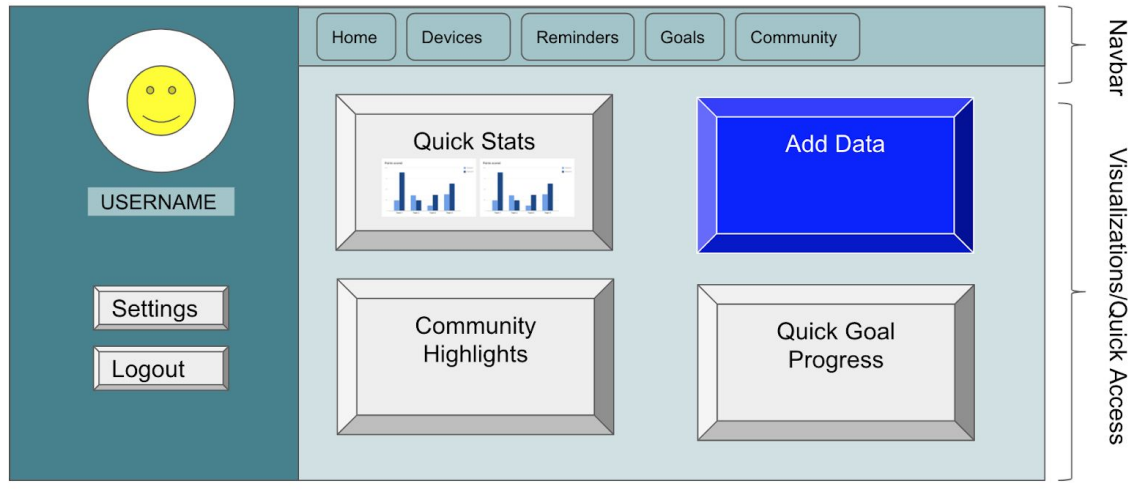


A screenshot of the same login page, but now the password field is filled with asterisks. A mouse cursor is hovering over the "Sign In" button.

- Username:** A text input field containing the text "user123".
- Password:** A text input field containing the text "*****".
- Sign In:** A button with a mouse cursor hovering over it.
- Forgot Password:** A button located below the "Sign In" button.
- Create new account:** A button located at the bottom of the form.

3. If the username exists and the password entered is the correct password associated with that account, the user is taken to the home screen of his application.


Home Page



4. Otherwise, he is prompted that the entered username/password is incorrect. After 5 incorrect attempts, the forgot password is automatically prompted to the user.

Forgot Your Password?

Enter your email and we'll email you instructions on how to reset your password.



Email:

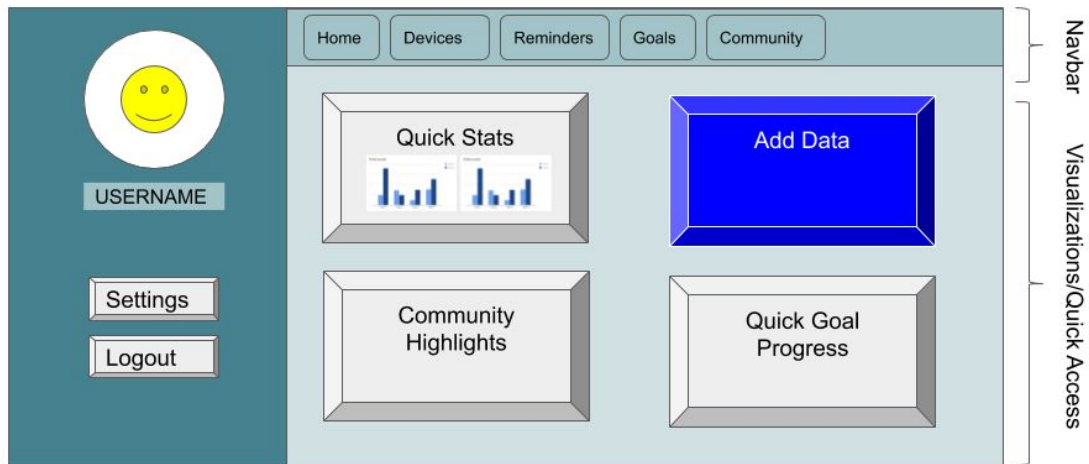
b. User Effort Estimation

Scenario 1: Inputting Daily Personal Data

Assume the application is open to the home page and that the user has logged in. What you need to do is (1) navigate to the Data Entry Page and (2) enter the data that corresponds to you

1. Navigation: 2 Total Mouse Clicks, as follows:
 - a. Click “Add Data” from the Home Screen-- after completing the data entry as referenced below --
 - b. Click “Add Data” on the Data Entry Page
2. Data Entry: 6 total mouse clicks and a variable number of keystrokes, depending on the length of the values you input, as follows:
 - a. Click cursor to the first data field, “calories burned”
 - b. Press the keys that correspond to the number of calories you burned that day
 - c. Press the “Tab” key to move to the next text field, “Steps”
 - d. Press the keys that correspond to the number of steps you took that day
 - e. Press the "Tab" key to move to the next text field, “Distance”
 - f. Press the keys that correspond to the number of steps you took that day
 - g. Click on the hours drop down menu for the number of hours you slept that night
 - h. Scroll on the drop down menu to select the number of hours
 - i. Click on the minutes drop down menu for the number of minutes you slept that night
 - j. Click “Add Data”

Home Page



↓Clicking “Add Data”↓

Data Entry Page

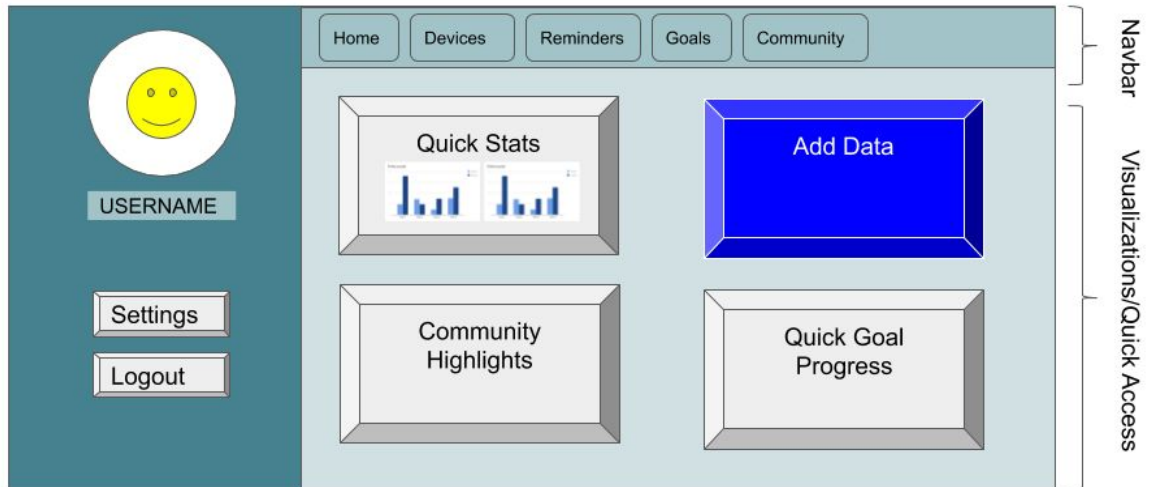
The 'Add Data' page has a 'Back' button in the top left. The form is divided into two columns. The left column contains 'Weight:' and 'Blood Pressure:' labels with text input fields, and an 'Add Data' button below them. The right column contains 'Calories burned:', 'Steps:', and 'Distance:' labels with text input fields. Below these is a 'Sleep:' label with two spinners (values 0 and 00) and a colon separator. At the bottom right are 'Add Data' and 'Upload from device' buttons.

Scenario 2: Updating Goals

Assume the application is open to the home page and that the user has logged in. What you need to do is (1) navigate to the Goals Page and (2) enter the data that corresponds to your goals

1. Navigation: 2 Total Mouse Clicks, as follows:
 - a. Click “Quick Goal Progress” from the Home Screen-- after completing the data entry as referenced below --
 - b. Click “Update Goals” on the Goal Management Page
2. Data Entry: 2 total mouse clicks and a variable number of keystrokes, depending on the length of the values you input, as follows:
 - a. On the Goal Management Page, click into the first field, “Blood Pressure”
 - b. Press the keys that correspond to the updated goal for that field
 - c. Press the “Tab” key to move to the next text field, “Steps”
 - d. Press the keys that correspond to the updated goal for your steps
 - e. Press the "Tab" key to move to the next text field, “Weight”
 - f. Press the keys that correspond to the updated goal for your weight
 - g. Click “Update Goals”

Home Page



↓Clicking “Quick Goal Progress”↓

Goal Management Page

Update Goals

Blood Pressure:

Steps:

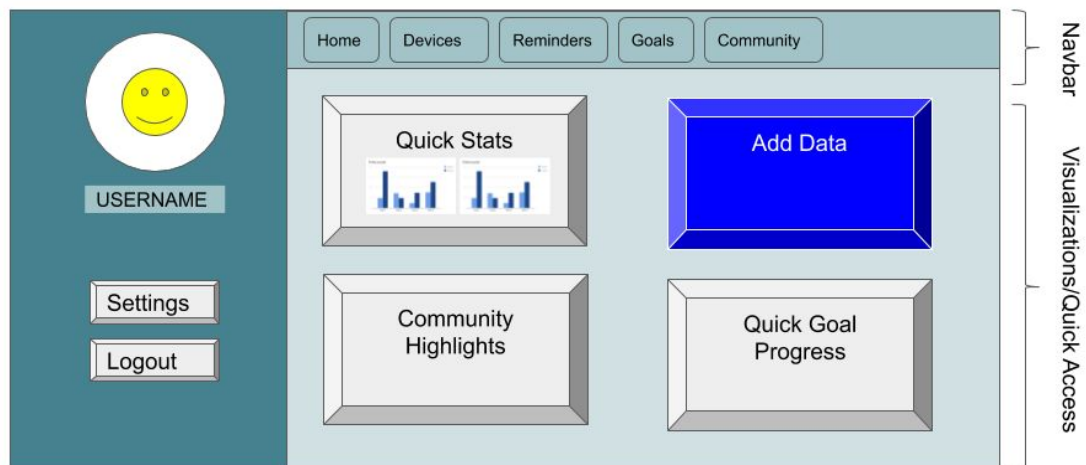
Weight:

Scenario 3: Viewing User Statistics

Assume the application is open to the home page and that the user has logged in. What you need to do is (1) navigate to the User Statistics Page and (2) View all metrics related to their health.

1. Navigation: 1 Total Mouse Click (to get to the User Statistics Page), as follows:
 - a. Click “Quick Stats” from the Home Screen
2. View all metrics: 5 total mouse clicks and 0 key strokes
 - a. Once you get to the User Statistics Page, you will automatically see the first chart, for weight.
 - b. Click on the next tab, Calories, to view your statistical data.
 - c. Click on the next tab, Steps, to view your statistical data.
 - d. Click on the next tab, Sleep, to view your statistical data.
 - e. Click on the next tab, Distance, to view your statistical data.
 - f. Click on the next tab, Blood Pressure, to view your statistical data.

Home Page



↓Clicking “Quick Stats”↓

User Statistics Page



Section 5: Domain Analysis

i. Concept Definitions

Responsibility	Type	Concept
Stores the user's profile picture, bio, list of posts, etc.	K	Profile
Allows the user to create, edit, or delete their profile.	D	ProfileChanger
Database that stores all user profiles and the information stored within them.	K	ProfileStorage
Attempts to place a user data point into the profile Database Management System (DBMS) if it's within a reasonable range.	D	DataEntry
Ensures that the data point is within a reasonable range. If so, the data point goes into the user's profile.	D	DataChecker
Database that stores all of the data points associated with each profile.	K	DataStorage
Checks the profile DBMS to see if the user put in any data at 10:00 p.m. every day.	D	TaskChecker
If the user hasn't put in their data in at 10:00 p.m., the system will remind the user to do so.	D	Reminder
Upon request, the image generating software creates an image for the user to see their data in their desired form (e.g. bar graphs, histograms, pie charts, etc.).	D	ImageGenerator
Displays the image on the screen after the image generating software creates it.	D	ImageDisplay
Sends an email to the user upon request of changing their password.	D	EmailNotifier
Changes the user's password after the user successfully confirms their email.	D	PasswordChanger
Allows the user to set goals and then store them into the goals DBMS (GoalStorage).	D	GoalSetter
Stores the list of goals associated with each user.	K	GoalStorage

Allows the user to add, block, or delete friends.	D	ModifyFriends
Takes the goals you have from the goals DBMS and then shares them with your friends.	D	ShareProgress
User attempts to enter a password to gain access to the system.	D	PasscodeEntry
Ensures that the user typed in the correct password.	D	PasscodeVerifier
Database that stores the password associated with each user.	K	PasscodeStorage
Check to make sure the user hasn't exceeded the maximum number of attempts to enter a password.	D	Controller

ii. Association definitions

Concept Pair	Association Description	Association Name
ProfileChanger <-> Profile	Changes made using ProfileChanger feature must be visually reflected on user's Profile.	update profile
DataEntry <-> DataChecker	Data user enters during DataEntry must be verified by DataChecker to ensure that the data is valid.	verify data
DataChecker <-> DataStorage	Data must be checked by DataChecker before it is stored in the Database Management System during DataStorage.	store data
TaskChecker <-> Reminder	Reminder will be sent out only if TaskChecker determines that the user has not inputted their daily data by 10:00 P.M.	verify message
ImageGenerator <-> ImageDisplay	ImageGenerator must generate visual before ImageDisplay displays it to the user.	prepare visualization
PasswordChanger <-> EmailNotifier	User must change password with PasswordChanger before EmailNotifier notifies them of their password change confirmation.	confirm password change
GoalSetter <-> GoalStorage	GoalStorage stores goals made by GoalSetter.	store goals
GoalStorage <-> ShareProgress	ShareProgress shares goals stored by GoalStorage.	share goals
PasscodeEntry <-> PasscodeVerifier	PasscodeVerifier verifies the validity of the passcode entered during	check passcode

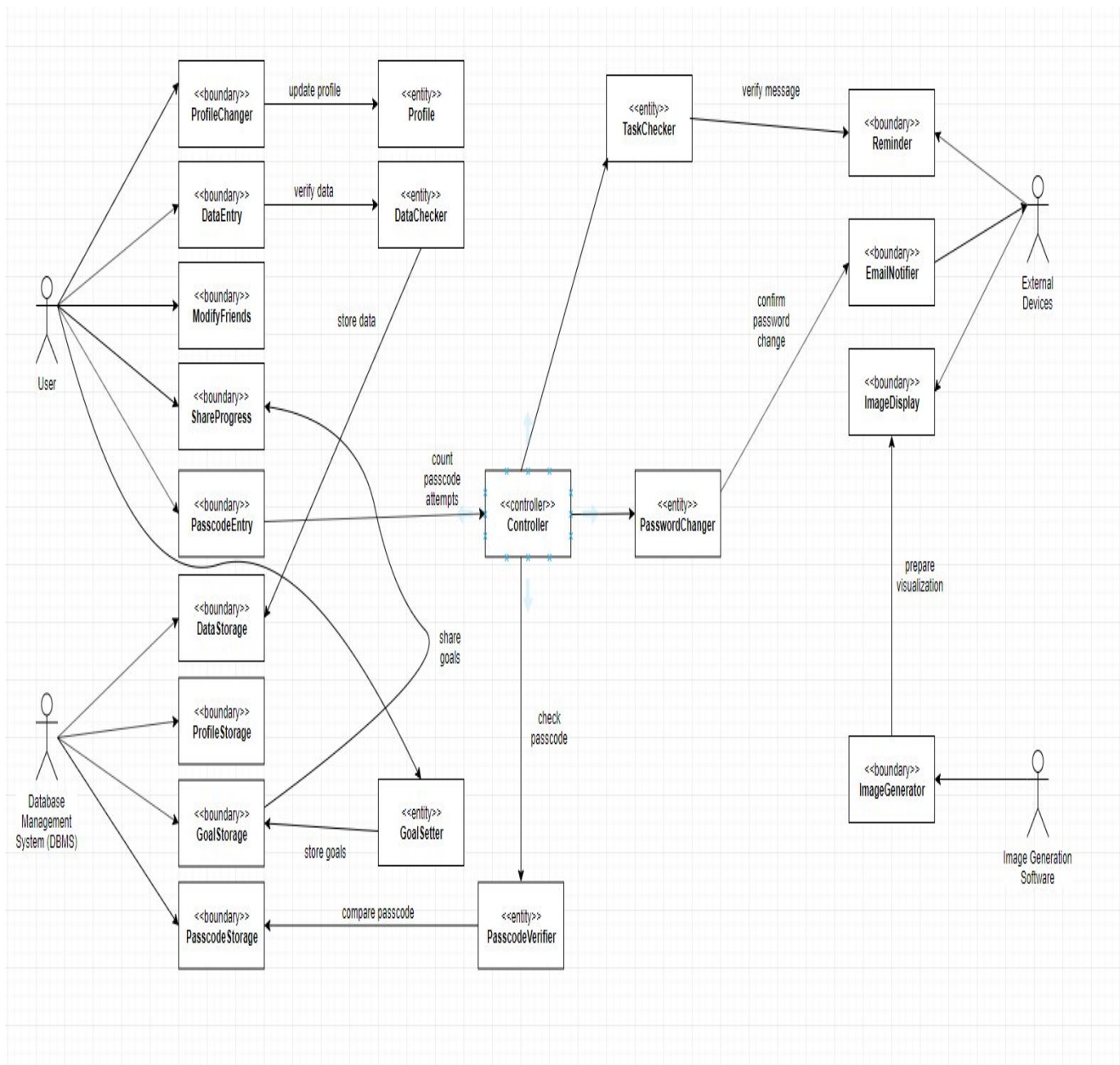
	PasscodeEntry.	
PasscodeVerifier <-> PasscodeStorage	PasscodeVerifier checks user-inputted password against correct password stored during PasscodeStorage.	compare passcode
PasscodeEntry <-> Controller	Controller verifies that count of user's passcode inputs during PasscodeEntry does not exceed a pre-set value.	count passcode attempts

iii. Attribute definitions

Concept	Attributes	Attribute Description
Profile	Cached user settings	Store a variety of user settings in one location which may affect operation characteristics of the UI. These diverse settings must share a common interface for later data persistence
ProfileStorage	Persisted user settings	Using the profile data entered into the Profile, store this data into the database for consistent UX between multiple login sessions
DataEntry	Data validation lookup table	Table of values that determines the acceptable range for data input
DataStorage	User fitness data	Data derived from a user's wearable device
TaskChecker	User activity log	Log within database to track activity in user sessions over time
Reminder	User activity log	This shares the activity log to generate reminder text for the expected user behavior
ImageGenerator	User fitness data	Access to this data required for generating graphics
ImageGenerator	Data visualization	Graphics representative of user data
EmailNotifier	User private information	Securely stored user private information used for access to user data outside of the main application. Mainly used to send emails.
PasswordChanger, PasswordVerifier, PasswordStorage	User credentials	Secure and encrypted user information
GoalStorage, ShareProgress	Goal	Common data structure in database extended by the user to create goals which affect the behavior of the UX for a number of login sessions for a

		single user
ModifyFriends, ShareProgress	Social Media Connector	API access to third party software for user to share information on other digital platforms

a. Domain Model



iv. Traceability Matrix

		Use Cases							
	Use Case	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8
	Priority Weight	25	47	14	41	6	26	15	37
Domain Concepts	Profile	x							
	ProfileChanger	x							
	ProfileStorage	x							
	DataEntry		x						
	DataChecker		x						
	DataStorage		x						
	TaskChecker			x					
	Reminder			x					
	ImageGenerator				x				
	ImageDisplay				x				
	EmailNotifier					x			
	PasswordChanger					x			
	GoalSetter						x		
	GoalStorage						x		
	ModifyFriends							x	
	ShareProgress							x	
	PasscodeEntry								x
	PasscodeVerifier								x
	PasscodeStorage								x
	Controller								x

b. System Operation Contracts

Operation	UC-1 : User Profile Management
Preconditions	<ul style="list-style-type: none">• User has been authenticated
Postconditions	<ul style="list-style-type: none">• User profile data has been updated in the database• Updated user profile information is immediately updated in the UI for this session

Operation	UC-2 : Data Input and validation
Preconditions	<ul style="list-style-type: none">• User has at least one device registered to the system which collects health and fitness data• Data being entered is deemed valid where validity is determined by the data falling within an acceptable range
Postconditions	<ul style="list-style-type: none">• If the data is valid<ul style="list-style-type: none">◦ New data is persisted to the database◦ Blockchain network updates population to include this data• If the data is invalid<ul style="list-style-type: none">◦ The UI tells the user that the data is valid◦ The application permits the user to try to input data again

Operation	UC-3 : Reminders for input
Preconditions	<ul style="list-style-type: none">• The current time of day is equal to the reminder time in the user's profile and settings
Postconditions	<ul style="list-style-type: none">• <None>

Operation	UC-4 : View visualizations
Preconditions	<ul style="list-style-type: none">• The user has selected the type of visualizations to view
Postconditions	<ul style="list-style-type: none">• The backend provides the UI with the data pertinent to the user's selection

Operation	UC-5 : Secure Password Changing/Verification
Preconditions	<ul style="list-style-type: none">• User has been authenticated

Postconditions	<ul style="list-style-type: none"> Database updates the tables containing login data to reflect changes
----------------	--

Operation	UC-6 : Goal Setting and Progress
Preconditions	<ul style="list-style-type: none"> The user has initiated at least one goal
Postconditions	<ul style="list-style-type: none"> <None>

Operation	UC-7 : Social Networking
Preconditions	<ul style="list-style-type: none"> User has been authenticated The user has connected at least one social media account the application The user has at least one element of data or visualizations which is marked as “shareable”
Postconditions	<ul style="list-style-type: none"> The shareable content is sent over the API for the selected social media platform

Operation	UC-8 : Login
Preconditions	<ul style="list-style-type: none"> <None>
Postconditions	<ul style="list-style-type: none"> If login is successful: <ul style="list-style-type: none"> User has been authenticated UI provides user with login success This login session is persisted to the database with the user credentials, timestamp of the attempted login, and details on rather or not the login was successful

Section 6: Project Size Estimation

The following table highlights the 8 identified use cases in terms of the size in Use Case Weights (UCW). The categories are assigned based on the complexity of achieving their success criteria and their interconnectedness in the Domain Model.

Use Case	Description	Category	Weight
UC-1 Profile creation & management	GUI for profile management and data persistence for user settings	Average	10
UC-2 Data Input and Validation	Controller for receiving user input and interfaces for validating data. Connection to Blockchain layer	Complex	15
UC-3 Reminders for input	GUI element that connects to the User's computer or handle device to display text	Average	10
UC-4 View Visualizations	Receive user selection for generating graphics and use Blockchain layer to display population statistics	Complex	15
UC-5 Password Verification	Securely handle user credentials	Simple	5
UC-6 Goal Setting and progress	UI interaction for storing data related to user interactions over time	Average	10
UC-7 Social Networking	Select certain data from the database to be shared over APIs for common social	Average	10

	media platforms		
UC-8 Login	Secure user authentication and encryption of password storage	Simple	5

Based on the UCW shown above, the Unadjusted Use Case Weight is as follows: $2 * 5 + 4 * 10 + 2 * 15 = 80$.

The following table calculates the technical complexity factor (TCF) based on the community standard constant values determined by the relevance of the technology listed below

Factor	Description	Weight	Relevance (0-5)
T1	Distributed system	2.0	5
T2	Response time/performance objectives	1.0	2
T3	End-user efficiency	1.0	4
T4	Internal processing complexity	1.0	4
T5	Code reusability	1.0	4
T6	Easy to install	0.5	1
T7	Easy to use	0.5	4
T8	Portability to other platforms	2.0	3
T9	System maintenance	1.0	3
T10	Concurrent/parallel processing	1.0	0

T11	Security features	1.0	5
T12	Access for third parties	1.0	4
T13	End user training	1.0	0
Technical Factor (TF)			40.5

Based on the formula of $TCF = 0.6 + TF/100$, the calculated TCF is 1.005.

With an assumed ECF = 1, the total size of the project as determined by $UCP = UUCW * TCF$
 *ECF is 80.4 UCP

Section 7: Plan of Work

a. Plan of Work

Oct 14 - 20:

- Complete 'peer review' evaluation
- Develop interaction diagrams for elaborated use cases
- Create UI mockups for all use cases
- Plan the system architecture and system design of the application
- Generate sample input data to use when testing the application
- Submit part 1 of Report 2

Oct 21 - 30:

- Begin coding, primarily focusing on the data interface
- Focus on creating the data interface
 - How user will input his/her data
 - Where user's data will be stored
 - How user can access his/her old data and data based on a certain demographic
- Go through the UI mockups and mark them up in HTML/CSS using the bootstrap library
- Make working navigation features to move through every page of the app and to test the functionality the app
- Create class diagrams
- Prepare and present demo 1

Oct 31 - Nov 3

- Continue coding and debugging
- Continue with implementing the system architecture and system design of the application
- Collect health and fitness data, such as heart rate, number of steps etc., to use for the application
- Submit part 2 of report 2

Nov 4 - 10

- Continue coding and debugging
- Improve UI if necessary, to make things clearer and/or easier for users
- Continue designing tests to check the functionality of the app
- Submit report 2

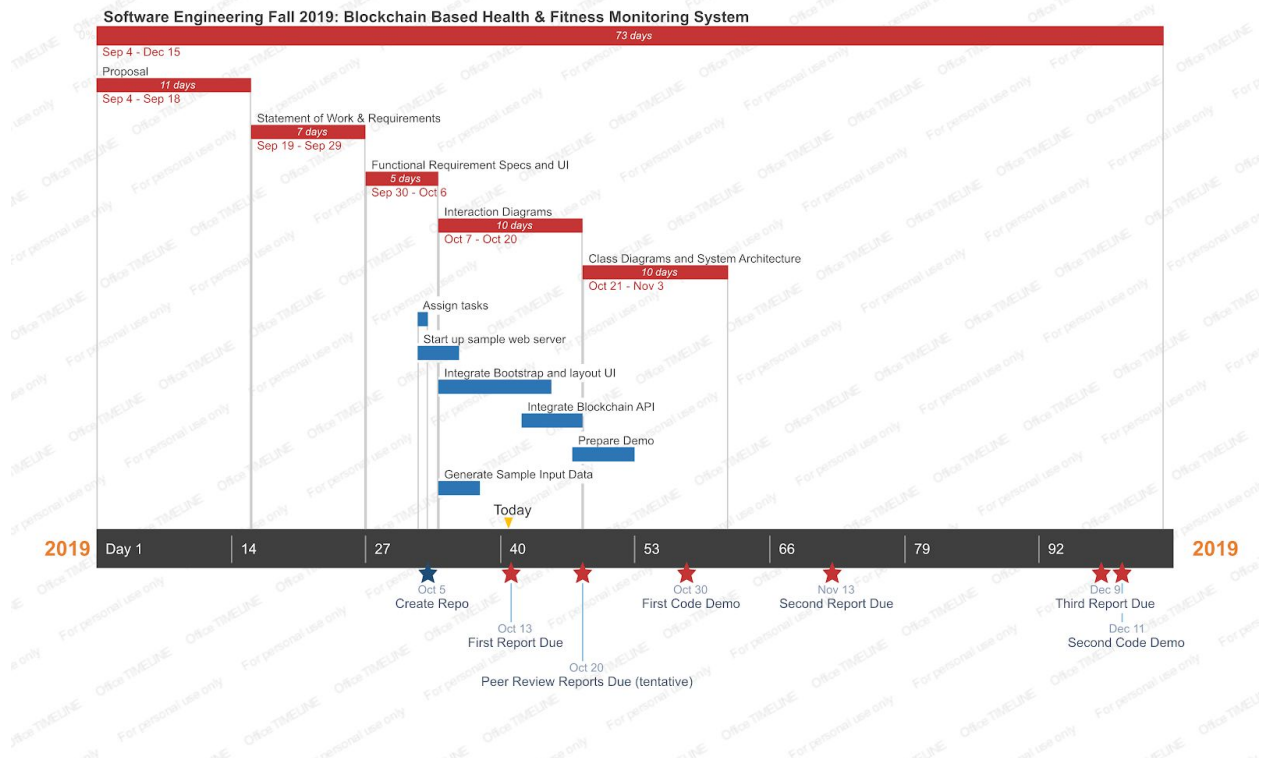
Nov 11 - Dec 9

- Continue coding and debugging
- Continue testing the application to see what needs to be changed or added
- Prepare for the second demo
- Submit report 3

Dec 10 - Dec 15

- Prepare and present demo 2
- Submit electronic archive

Software Engineering I Fall 2019 Group #3



The above Gantt chart is driving the team's deadlines and internal milestones.

b. Product Ownership

The team developing this application consists of 8 members collaborating to develop a web application. There is no elected team leader as the team communicates electronically. The team utilizes the Google Documents suite to facilitate the tracking of project status for written deliverables and code. The code is source controlled and hosted on GitHub. By tracking the multiple components of the software through an interactive spreadsheet, we are able to equally distribute the workflow to all team members to ensure a smooth Agile development environment. Each team member contributed equally to the progress of our project thus far. Till this point, we have mainly coordinated the thought process and work we have cut out ahead of us. To this end, it was often group member Sasan who delegated some responsibilities to ensure that we were all focused on what work needed to be done and did it in a timely fashion, although we have no explicit team leader. Group member Breanna was very useful in preparing the domain model, system operation contracts, and use case diagram. Group members Nithyasree and Nithya both ensured good looking UI mock-ups and diagrams. Nithyasree also outlined our plan of work that we have ahead of us and worked on the customer problem statement. Group member Shruthi was largely behind the system sequence design and worked with member Sasan on the casual description of the use cases. Group member Jack was responsible for the bulk of the domain

analysis section. Group member Sean helped set up our GitHub repository and reviewed many of the sections while they were worked on by fellow group members. Group member Pradyumna worked on the glossary and user interface specifications. Members Pradyumna, Sasan, and Shruthi worked on the detailed use case descriptions. Members Pradyumna and Nithyasree also were behind the formatting and aesthetics of the report. While these are some of the explicit agenda items that group members were mainly responsible for, it must be said that each group member did in fact contribute equally to the progress of our project. This can especially be pointed out by the fact that anything one person may have taken more responsibility for was reviewed by multiple other group members. It was always the case that each group member worked on a section with at least one other to ensure that it was not purely one mind behind a specific part. This was done to ensure cohesiveness of the project, and as a result of good team communication, we feel the parts of our report fit together quite well. Going forward, we plan to stick with this style of group effort, and we will self-delegate accordingly. As we move into more technical phases that involve coding and adhering to the use cases, we will determine more specifically which members are more proficient in front-end design or back-end development, for example.

References

“Safe Sharing of Population Descriptors”

Marsic, Ivan.

https://content.sakai.rutgers.edu/access/content/group/fbfe7282-89ce-4a4d-a619-943dee895665/Programming%20Project/Programming%20Project_%20Safe%20Sharing%20of%20Population%20Descriptors.pdf

“Report #1 - User Effort Estimation”

Marsic, Ivan

<https://www.ece.rutgers.edu/~marsic/Teaching/SE1/report1-appA.html>

“What Is Blockchain Technology? A Step-by-step Guide For Beginners”

Ameer Rosic- Blockgeeks-Marko @marko-ceki-Alissa Brandemuhl-Tiffany

Mccullar-Mark Silen-Alex @sashabakht -

<https://blockgeeks.com/guides/what-is-blockchain-technology/>

“What Is Encryption? - Definition from Whatis.com”

Margaret Rouse-Kevin Ferguson-Margaret Rouse-Margaret Rouse -

<https://searchsecurity.techtarget.com/definition/encryption>

“Global Glossary Of Blockchain Terms 2.0 in 5 Languages”

<https://blockchaintrainingalliance.com/pages/glossary-of-blockchain-terms>

“Text Message Template Ideas To Increase Customer Engagement: Direct Sms”

<https://www.directsms.com.au/text-message-template-ideas-improve-customer-engagement/>

“The World Of Web and Mobile Applications”

Admin - <https://pctechieguy.com/the-world-of-web-and-mobile-applications.html>