

# **Health Goals**

**Software Engineering (332:452)**

**Second Report Full**

**Group Number: 1**

**Visit Us At: <https://health-goals.herokuapp.com>**

**Submission Date: March 17, 2019**

**Group Members:**

Anushka Desai

Varun Bhandari

Niharika Mishra

Sai Singanamala

Arti Patel

Kishan Patel

Jose Cuentas

Eric Rivera

Varun Raghuraman

# Table of Contents

<b>Individual Contributions Breakdown:</b>	3
<b>1. Interaction Diagrams</b>	4
<b>2. Class Diagram and Interface Specification</b>	12
a. Class Diagram	12
b. Data Types and Operation Signatures	12
c. Traceability Matrix	16
<b>3. System Architecture and System Design</b>	18
a. Architectural Styles	18
b. Identifying Subsystems	18
c. Mapping Subsystems to Hardware	19
d. Persistent Data Storage	19
e. Network Protocol	20
f. Global Control Flow	20
g. Hardware Requirements	20
<b>4. Algorithms and Data Structures</b>	21
a. Algorithms	21
b. Data Structures	22
<b>5. User Interface Design and Implementation</b>	23
<b>6. Design of Tests</b>	<b>Error! Bookmark not defined.</b> 7
<b>7. Project Management and Plan of Work</b>	44
a. Merging the Contributions from Individual Team Members	44
b. Project Coordination and Progress Report	46
c. Plan of Work	47
d. Breakdown of Responsibilities	48
<b>8. References</b>	50

# Individual Contributions Breakdown:

Task	Kishan	Arti	Anushka	Niharika	Pranathi	Varun B.	Varun R.	Eric	Jose	Total
Sec 1a - UML Diagrams (10)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 1b - Prose Description of Diagrams (10)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 1c - Alternate Solution Description (10)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 2a - Class Diagram & Description(5)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 2b -Signatures (5)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 3a - Styles (5)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 3b- Package Diagram (2)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 3c - Map Hardware (2)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 3d- Database (3)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 3e-g (Other) (3)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 4 - Algorithms and Data Structures (4)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 5a - Appearance (6)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 5b - Prose Description (5)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 6 - Testing Design (12)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 7a - Document Merge (11)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 7b - Project Coordination Progress (5)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
Sec 7c - Plan of Work (2)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%
References (-5*)	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	11.11%	100%

Total(points)	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	100
---------------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-----

## 1. Interaction Diagrams

### UC - 1: GetMealPlan Sequence Diagram

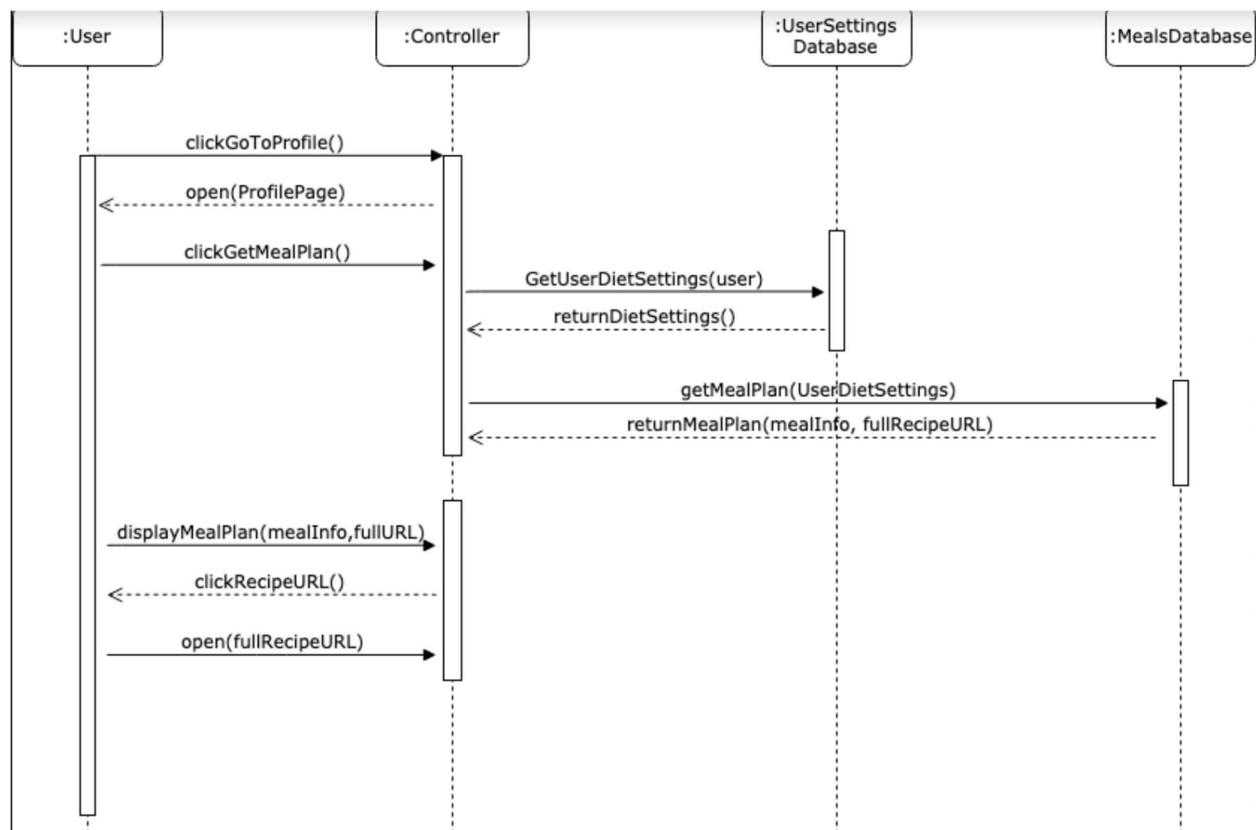
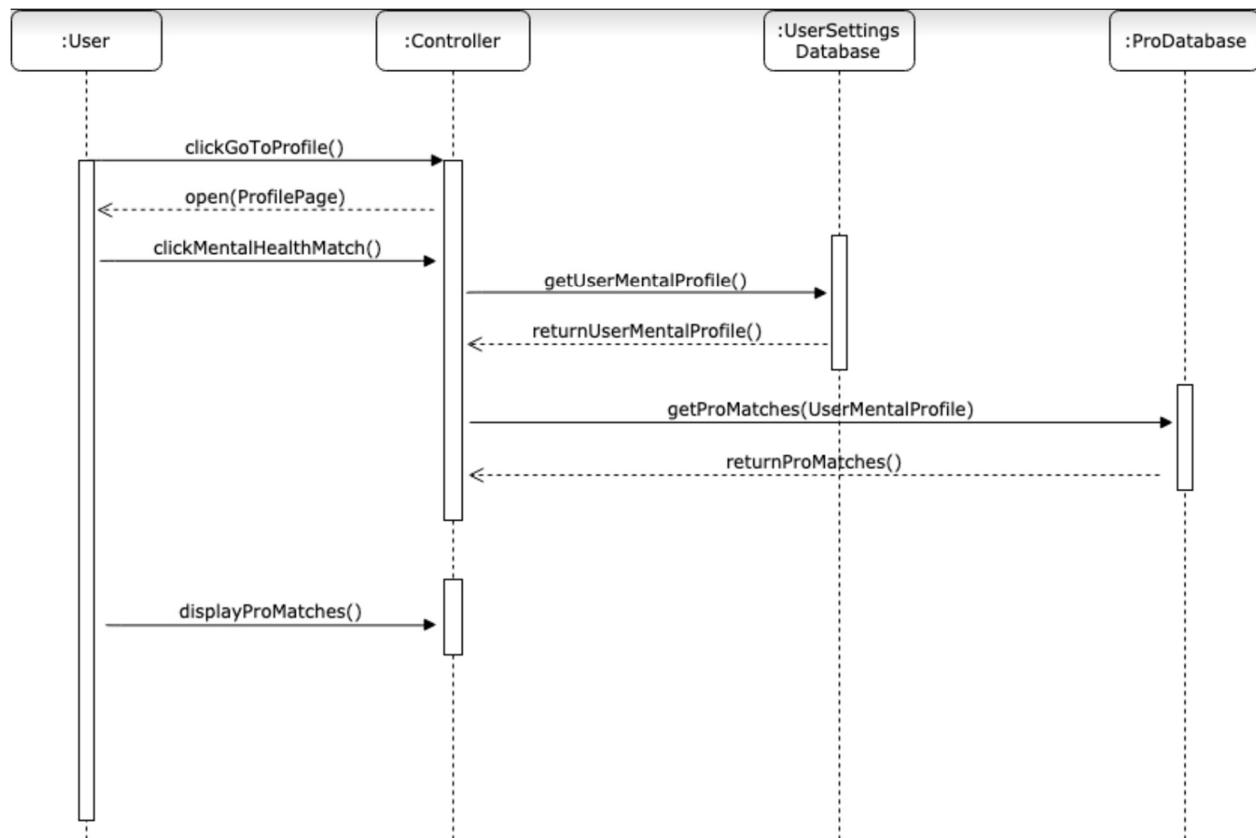


Diagram 1.1

The diagram above shows how each object will interact when a user is accessing their Meal Plan. From the home page, the user will click on the “Go to Profile” button. The controller will then open their Profile page and display it on the user’s screen. The user will then click on the “Get Meal Plan” button to obtain their day’s meal plan. The controller will then fetch the user’s diet preferences from the User Settings Database, which will send the controller the data back. The user’s settings will already be saved in the User Settings Database. (They edit that when signing up/ change settings.) The controller will then fetch a meal plan from the Meals Database, passing in the user’s settings as parameters. That database will return a meal plan based on the user’s

preferences to the controller. The controller will display the names of the breakfast, lunch, and dinner on the webpage, but not any other details. The user will click on “Full Recipe” button under the food name for the full recipe, and the controller will then open that webpage.

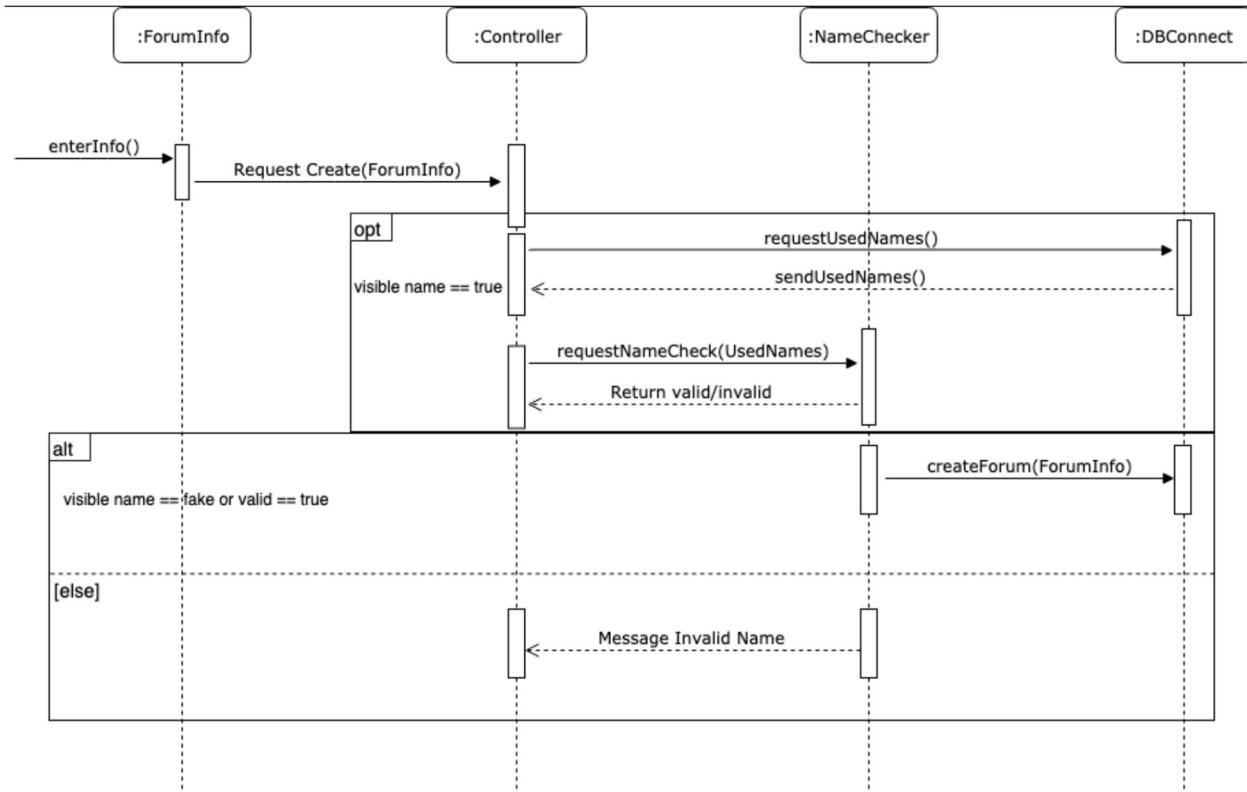
### UC - 2: MentalHealthMatch Sequence Diagram



**Diagram 1.2**

The diagram shows the responsibilities of each object whenever the use case Mental Health Match is happening. The user will start it off by clicking mental health matches from their profile, which then will direct that command to the controller. The controller will request the user profile from the user settings database, who then returns the user's mental profile to the controller and then to the user. Meanwhile also requesting to get the mental health professional matches from the professionals' database. The professional's database returns the matches to the controller who then displays them to the user.

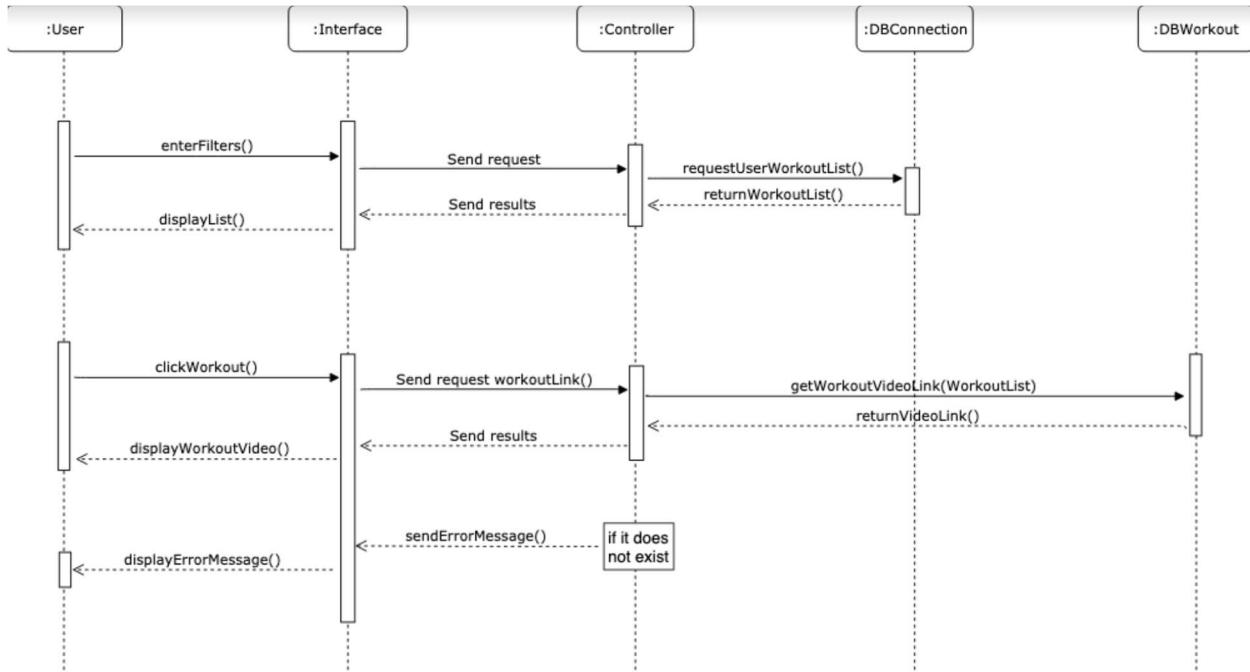
## UC - 4: StartForum Sequence Diagram



**Diagram 1.3**

When assigning responsibilities to the objects, the expert doer principle was kept in mind. As can be seen in the diagram, the `NameChecker` object is responsible only for verifying that a name is not already in use. The `Controller` is in charge of making requests and sending messages, the `DB Connection` receives and stores forum data, while the `ForumInfo` object contains the necessary information. Object responsibilities were kept to a minimum (highest responsibility count is 4 for the `Controller`), and communications chains were kept short. These decisions reflect the high cohesion and low coupling principles.

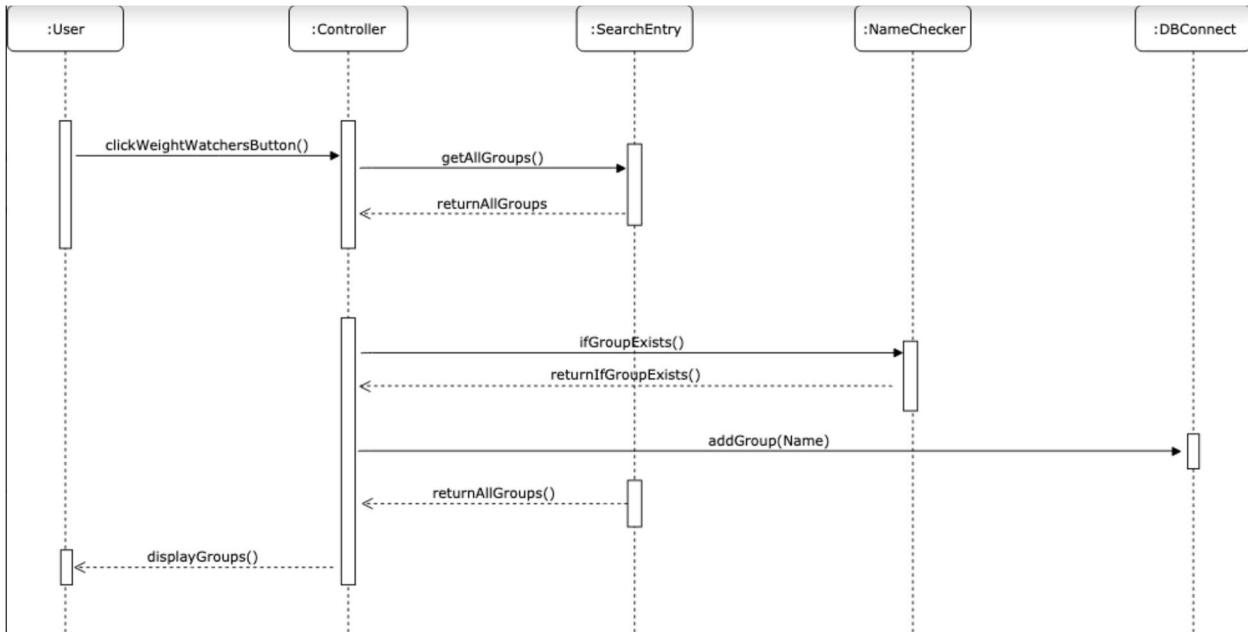
## UC - 6: GetWorkout Sequence Diagram



**Diagram 1.4**

The above interaction diagram is for UC-6: Get Workout. The user will first choose the filters given in the GetWorkout Page, the interface will send a request to the controller which in return will request the workout lists from the Database Connection. Then the DB will return the workout list to the controller which will send the results back to the interface. At this point, the interface will display the list of workouts most relevant to the input search filters to the user. The user can now choose a workout, and the interface will notify the controller of the request for workout link from the DB. Note that from this moment, if the workout link does not exist in the database then the controller will send an error message to the interface to notify the user that the workout link is no longer available. If it is, Workout DB will return the video link to the controller which sends the results to the interface which will display the resulting workout.

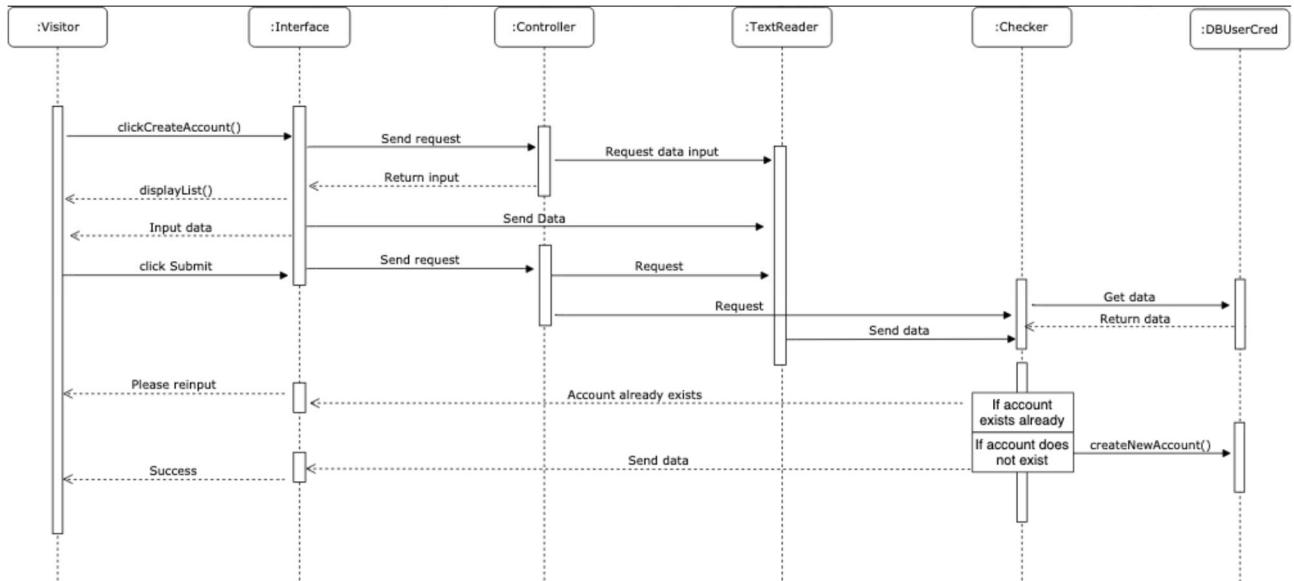
## UC - 9: JoinGroup Sequence Diagram



**Diagram 1.5**

The above picture shows the interaction diagram for UC - 9: JoinGroup Sequence Diagram. This interaction diagram goes in depth and shows how the system actually works and what calls are being made to which subsystem. The user first clicks the Weight Watchers button and is directed to a page that displays all the groups along with leaderboard of top users. The user can either select a group from the list or filter by providing a little information. This is done by the Controller sending a request to the Search Entry to get all groups and returning them back to the Controller. Next, if a user wants to create a new group, a request is sent by the Controller to Name Checker to see if the name has already been used. If it has not been used, the user is allowed to create a group. This sends a request to DB Connect which connects to the database and allows the user to add a group with a new name. Finally, all the groups, as well as the newly created one, are displayed on the page for the user to see.

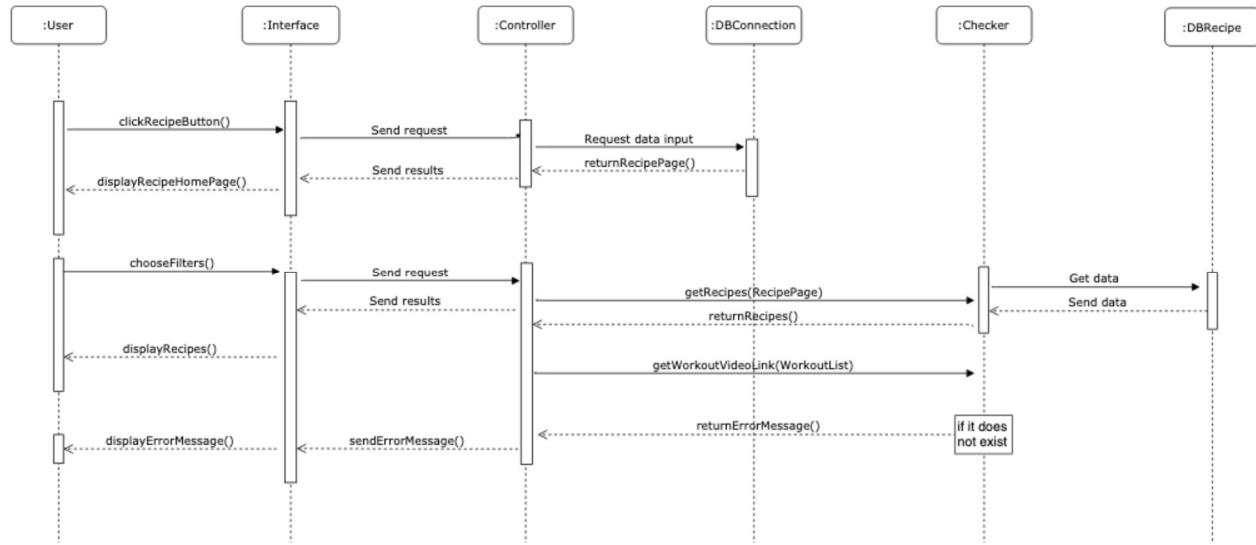
## UC - 10: CreateAccount Sequence Diagram



**Diagram 1.6**

The above picture shows the interaction diagram for UC-10, Create Account. Our earlier sequence diagram illustrates how the user, system and database all interact. The user clicks the create account button, inputs their information and the system creates an account by interacting with the database. The interaction diagram, however, goes in depth into how the system really operates and all the internal players involved. Firstly, the visitor clicks the create Account button and the Interface sends a request to the controller, which in turn requests information through dialog boxes. The controller sends this request to TextReader, so the visitor can input their data. This control then sends this data to the Checker, which interacts with the User Credentials Database to make sure the account has not already been created. If it has, the Interface sends a message that the account details already exist and to re-submit data. If it has not, the Checker will save the information to the Database. Then the success message is passed along so the Interface can display the new account to the visitor.

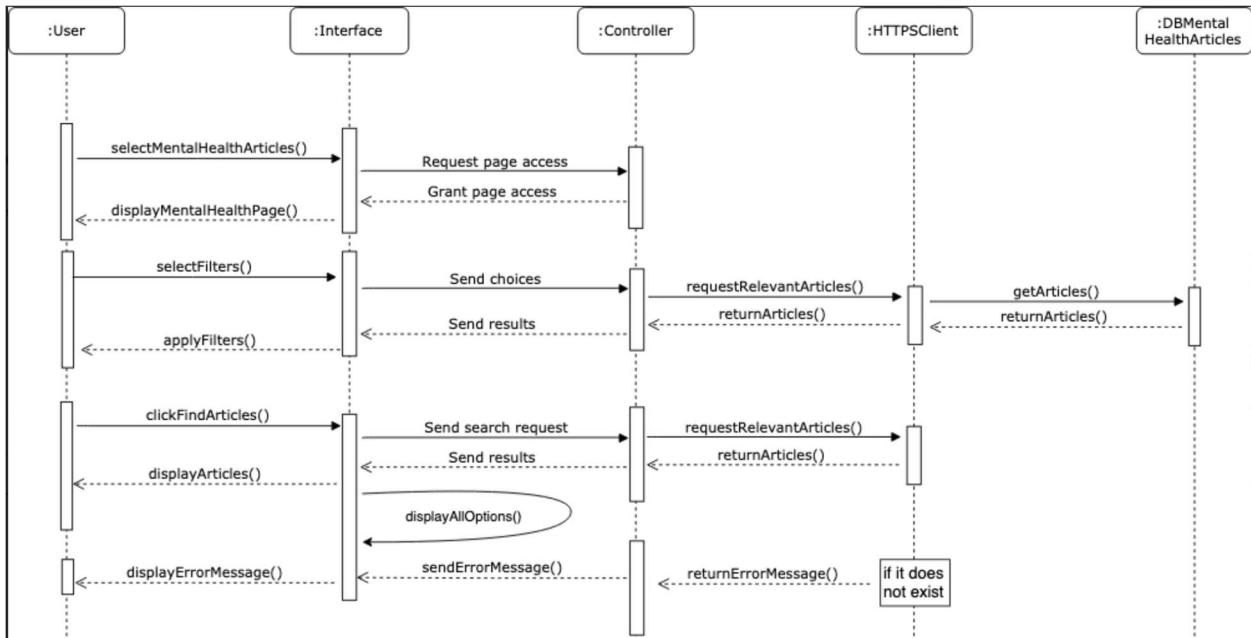
## UC - 14: RecipeMatch Sequence Diagram



**Diagram 1.7**

The above interaction diagram is for UC-14: Search Recipe. The user will start this process by clicking the recipe button. The interface will send this request to the controller which will request the webpage from the DB connection. After the recipe homepage is displayed to the user, he/she can now choose filters based on his/her preferences and dietary needs. The filtered choices will be sent to the controller by the interface where the controller will send this data to the checker. At this point, if the data (recipe) exists in the RecipeDB, the checker will obtain and return the data back to the controller. Now the controller can send the results back to the interface and display the filtered recipes to the user. Note that if the data (recipe) does not exist within the RecipeDB, an error message will be displayed to the user.

## UC - 15: SearchInfo Sequence Diagram



**Diagram 1.8**

This interaction diagram is for UC-15: SearchInfo. As shown above, the process starts with the user selecting the Mental Health Articles category on the main page. The user will land on the Mental Health Resources web page. On this page, the user will have several options: apply filters, search for new articles, view trending mental health topics/articles, etc. When the user applies filters, the current list of articles being displayed will be updated accordingly. Likewise, when the user enters a search query and clicks the search button for the search bar appropriate articles will be returned based on the filters and query entered by the user. There's also the possibility of nothing being returned if the HTTPS client was unable to fetch any relevant results from the mental health articles database; in which case, a simple "No results found" message is displayed to the user.

## 2. Class Diagram and Interface Specification

### a. Class Diagram

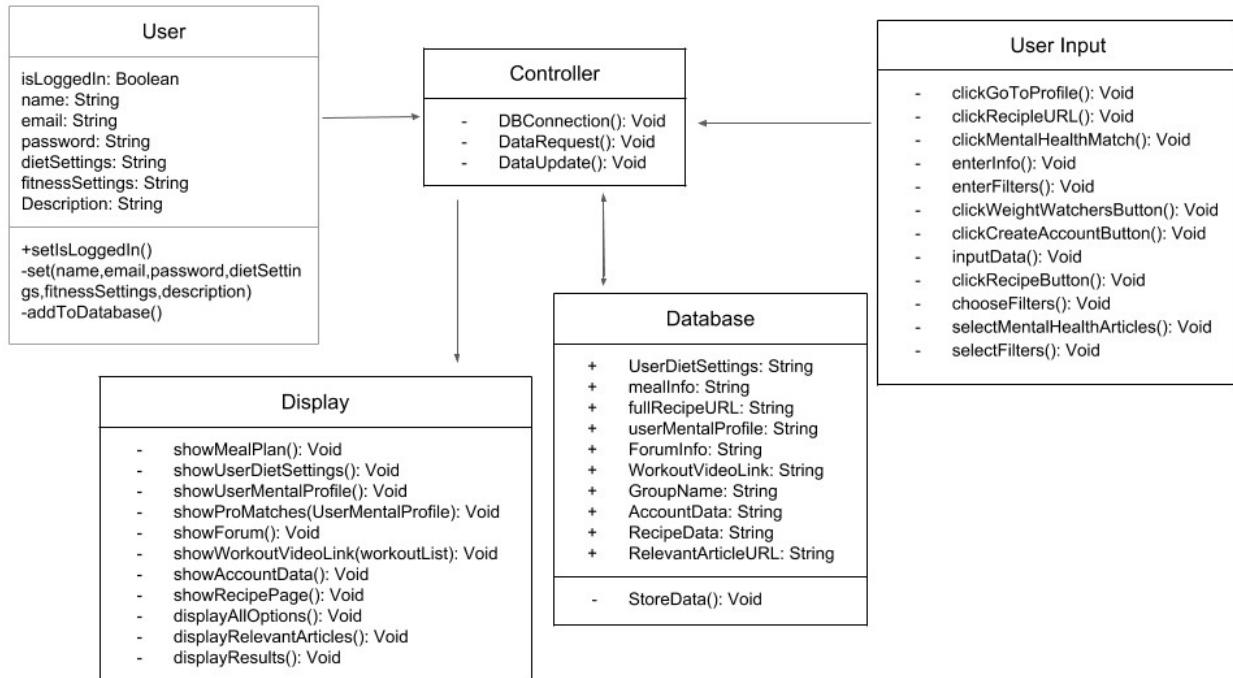


Diagram 2.a

### b. Data Types and Operation Signatures

#### User:

##### *Class Description*

- The user class is used to contain all the registered user's important information. It is also used to track if the user is logged in and to add an account to the database of users

##### *Attributes*

- IsLoggedIn: Boolean
  - Simple Boolean attribute that indicates whether the user is currently logged in or off
- Name: string

- This string contains the user's name and last name
- Email: string
  - String containing the user's email
- Password: string
  - String containing the user's password
- dietSettings: string
  - String containing the user's dietary settings and preferences
- fitnessSettings: string
  - String containing the user's fitness settings and information
- Description: string
  - A string containing a personal description entered by the user

### *Operations*

- setIsLoggedIn()
  - Changes the status of the isLoggedIn attribute upon being called
- set(name, email, password, dietSettings, fitnessSettings, description)
  - Updates the attributes of the user class given the string parameters entered by the user
- addToDatabase()
  - Adds a new user to the user database

## Display:

### *Class Description*

- The class represents the main display screen with which the user interacts. All of the options below are linked to icons on the display

### *Operations*

- showMealPlan():
  - Retrieves user's meal plan based on dietary condition selected earlier.
- showUserDietSettings():
  - Retrieves the user's dietary information, selected on prior occasions
- showUserMentalProfile():
  - Retrieves user's mental health profile
- showProMatches(UserMentalProfile):
  - Provides user with mental help tools based on their mental health profile
- showForum():
  - Gets a particular forum search from a user
- showWorkoutVideoLink(workoutList):

- Provides a list of Workout Videos in link form based on searches
- showAccountData():
  - Retrieves account data from the Database for a User
- showRecipePage():
  - Returns the recipe page based on user inputs
- displayAllOptions():
  - Displays all options that a user may select
- displayRelevantArticles():
  - Displays articles similar to the search entry
- displayResults():
  - Displays the results of various operations, such as searched

## **User Input:**

### *Class Description*

- The class represents the ways in which the icons from the display may be interacted with

### *Operations*

- clickGoToProfile():
  - The user may select this to go to their main profile
- clickRecipieURL():
  - The user may select their recipe's url to go to the linked page
- clickMentalHealthMatch():
  - The user may click the mental health match icon to match them with the tools they need
- enterInfo():
  - The user may enter their information into their profile
- enterFilters():
  - The user may select their filter and restrictions
- clickWeightWatchersButton():
  - The user may selected this icon to track their weight
- clickCreateAccountButton():
  - The user may create an account
- inputData():
  - The user may enter account information
- clickRecipeButton():
  - The user may select this to bring up recipe pages
- chooseFilters():
  - The user may choose filters for their search

- The user may select this to add filters for their searches from preset options
- selectMentalHealthArticles():
  - The user may select and view mental health articles
- selectFilters():
  - The user may apply the filters they chose

## **Controller:**

### *Class Description*

- The class manages and directs flow of data between subsections of the system

### *Operations*

- DBConnection():
  - The connection to the database when the data for the systems/users is stored. \ All stored information or information that must be stored is linked by this
- DataRequest():
  - Asks the database for stored information
- DataUpdate():
  - Updates data within the system based on inputs

## **Database:**

### *Class Description*

- The class represents the actual database of stored information

### *Attributes*

- UserDietSettings: String
  - A string representing the dietary information selected
- mealInfo: String
  - The information about a specific meal plan, stored in string form
- fullRecipeURL: String
  - The url to a particular recipe, stored in strings
- userMentalProfile: String
  - The mental health profile of a user, stored in a string
- ForumInfo: String
  - Forum information, stored in a string (title, posts, etc)
- WorkoutVideoLink: String

- The url for a workout video, stored in a string
- GroupName: String
  - The name of a group, stored as a string
- AccountData: String
  - The account information of a user (weight, name, age, location, passwords, and assorted health information) stored as a string
- RecipeData: String
  - The actual recipe information (ingredients, calories) stored in a string
- RelevantArticleURL: String
  - Health article information stored in a string

### *Operations*

- StoreData():
  - Stores the data in the database

## c. Traceability Matrix

	Display	User	User Input	Controller	Database
Controller		✓	✓		
Search Autofill				✓	✓
Search Entry	✓		✓	✓	
Logger	✓	✓		✓	
DB Connect		✓		✓	
User Choice	✓			✓	
Forum Info	✓			✓	✓
Name Checker			✓	✓	✓
Profile Info	✓	✓		✓	✓
ID Checker			✓	✓	✓
Group Info	✓			✓	✓

Table 2.c

- Display:
  - Search Entry - Displays articles similar to the search entry
  - Logger - Display account data if user enters correct account login or display create account page if user does not have an account
  - DB Connect - Retrieves account data from the Database for a User
  - User Choice - Displays all options that a user may select
  - Forum Info - Retrieves user's mental health profile and provides the user with their choice of mental help tools based on their mental health profile
  - Profile Info - The user can go to their main profile and view all account data
- User:
  - Controller - Take User Profile information and settings and request update to database
  - Logger - User is currently logged in or off
  - DB connect - Adds a new user to the user database
  - Profile Info - Have user input name, email, password, dietSettings, fitnessSettings, description
- User Input
  - Controller - User is able to interact with the website by clicking one of the many features as well as being able to update user profile information
  - Search Entry - User can choose filters provided in the website and narrow search results to their preference
  - Name Checker - Ensures that the username is not being used by a different user
  - ID Checker - Ensure that each user chooses a unique ID
- Controller:
  - Search Autocomplete - Controller receives autocomplete search parameters
  - Search Entry - Controller receives the user's search input
  - Logger - Controller receives logger data
  - DB Connect - Controller generates request to store/recieve to database
  - UserChoice - Controller receives user's choice

- Forum Info → Controller receives forum information from user and systems
  - Name Checker → Controller conveys a namecheck request and passes list of used names
  - Profile Info → Controller receives profile information from user
  - ID Checker → Controller conveys a id check request and passes list of used user IDs
- Database:
  - Search Entry → Database receives the user's search input and stores it
  - Forum Info → Database stores relevant forum information
  - Name Checker → Database stores used username list
  - Profile Info → Database stores relevant profile information from controller
  - ID Checker → Database stores list of used ID names

## 3. System Architecture and System Design

### a. Architectural Styles

The Health Goals app primarily uses the client-server architecture model. As the user makes use of the site resources, the system triggers a request to the server with setter and getter functions, which in turn responds by communicating to the centralized database and retrieving or storing the necessary information. The architecture is also data-centric as it relies heavily on database management. This combination of models supports the system well because we can have an easy and manageable front end that can be updated whenever and it allows multiple clients to access the services and data at any given time while maintaining the data efficiently organized.

### b. Identifying Subsystems

#### UML Package

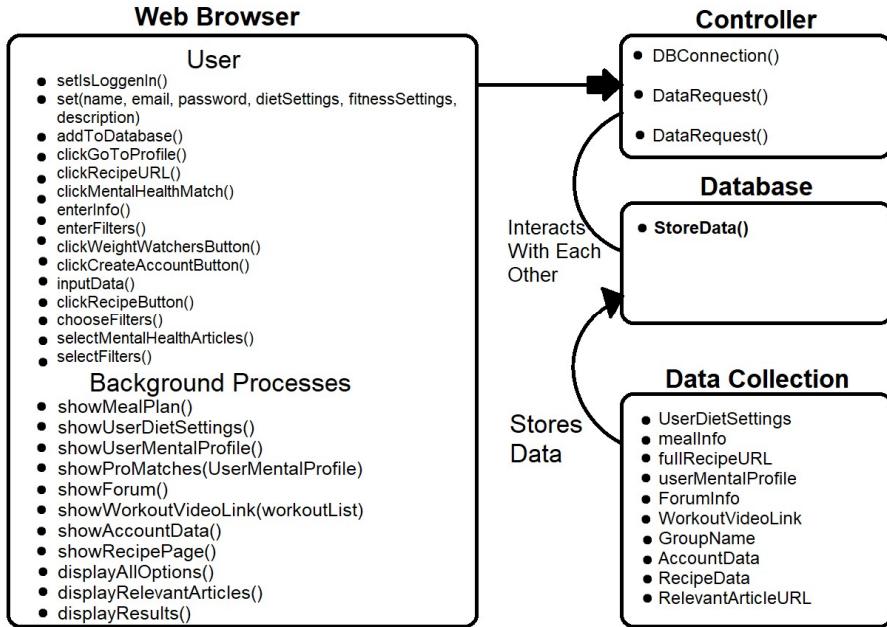


Diagram 3.b

## c. Mapping Subsystems to Hardware

Using the classes User, Controller, Database and the domain model for Use Cases 1, 2, 6, etc... we mapped the system in the following way. There are two subsystems: the client (web browser) and the server (web server). The client side contains the website and user interface for the user to interact and to look up any data. The server side contains the database which would be hosted on an AWS instance and would receive requests to process the data. This enables us to do the heavy processing of databases in our server and still be able to run the client side in most of the systems currently on the market.

## d. Persistent Data Storage

We will be storing data on a relational database, more specifically the MySQL database. We will be storing users' login information, diet/exercise preferences, and forum IDs. We will have a database of references for meals, exercise routines, therapists, trainors, articles, and uploaded forums; this means the database will contain information regarding these categories based on user preferences. References will be used rather than storing the raw information in the database to conserve space. These references will be used to fetch the information from the actual sources by our

website.

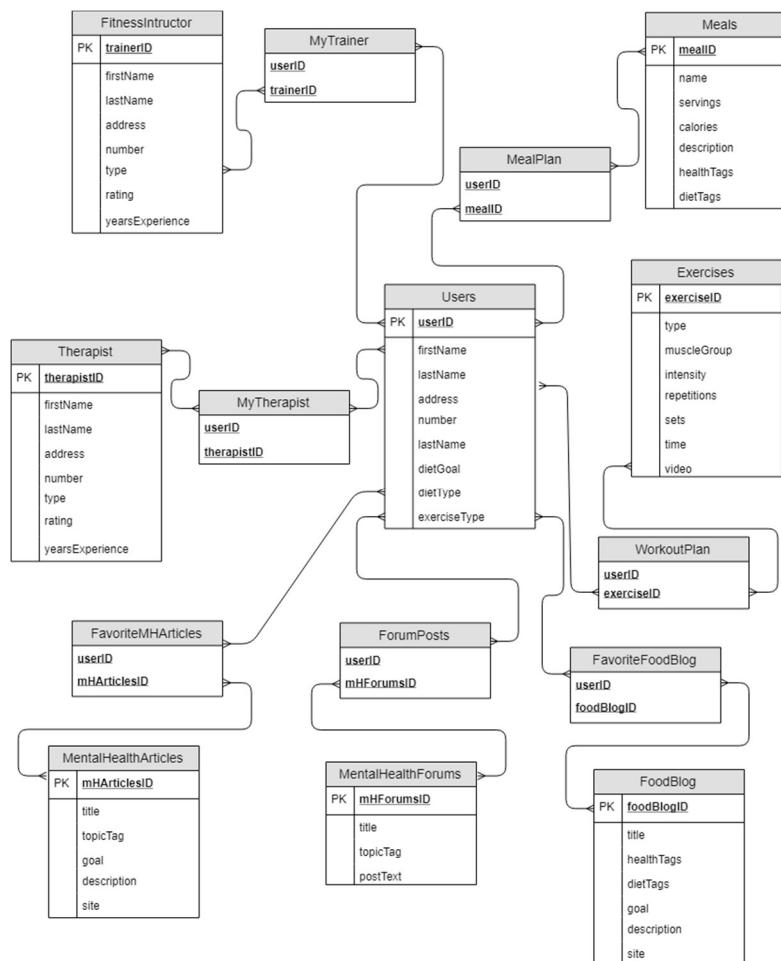


Diagram 3.d

## e. Network Protocol

This part does not apply to our project.

## f. Global Control Flow

The system is event-driven. It waits in a loop for events, and each user can use the system by clicking or typing information so they can generate whichever actions they want. There are no timers in our system. It is an event-response type.

## g. Hardware Requirements

Our system can use multiple different operating systems such as Windows 7, Windows 10, Linux or Mac os. The hard disk space needs to have a minimum of 4 Gbytes due to the size of our database and the minimum network bandwidth can be 56 kbps. There is

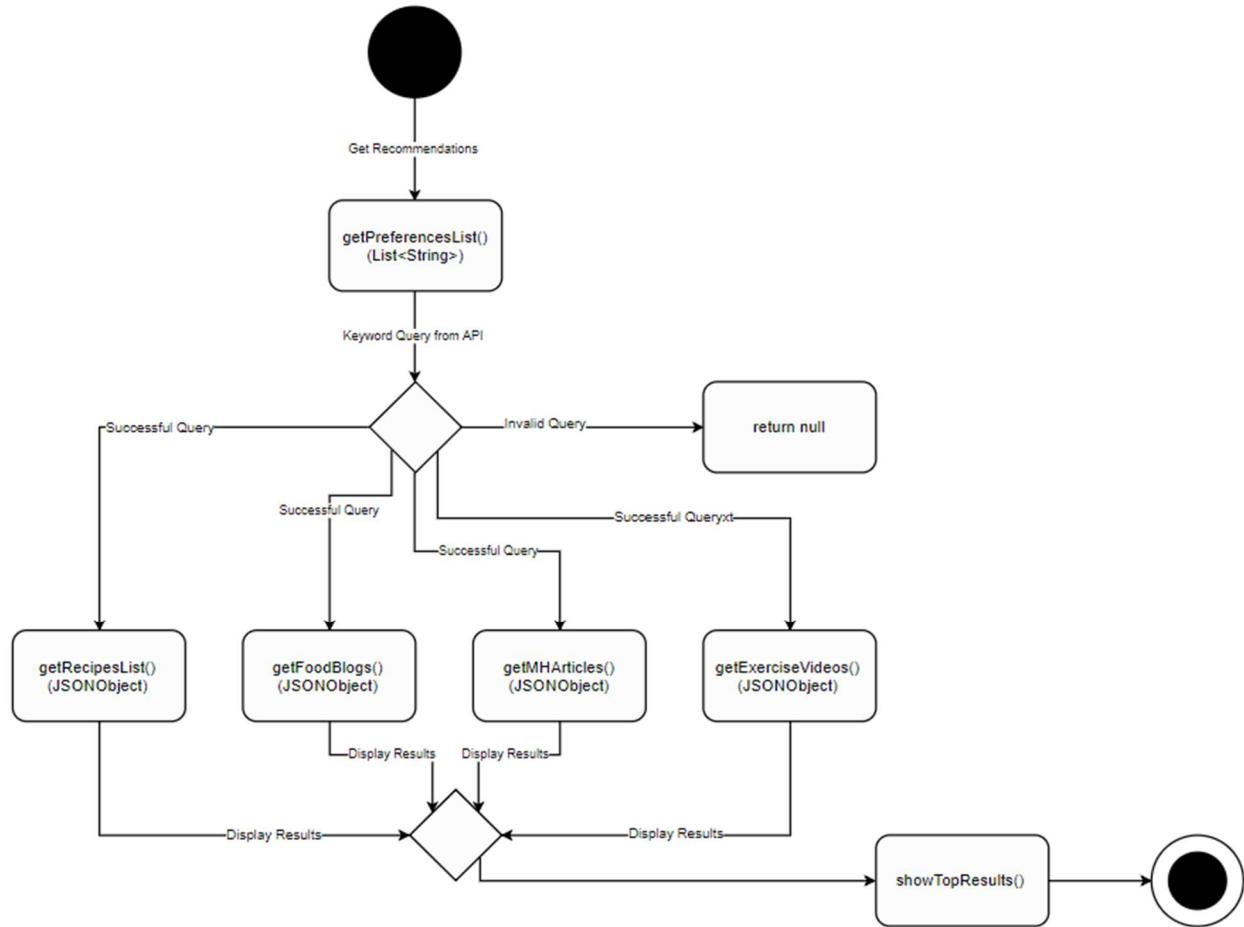
no minimum resolution required. Our backend will mainly use PHP and we will be using MySQL for our databases.

## 4. Algorithms and Data Structures

### a. Algorithms

For almost all use cases, the user receives a list of certain recommendations (links to sites, curated workout sets, etc) based on their input. For instance, UC-14 takes a user's profile, which has information such as dietary restrictions and food preferences, and search filters, such as types of recipes or meals, to create a recommended list of recipes that should be personalized for each user. Thus, our website will require some sort of recommendation or recommender system. This is best for when users are interacting with many different items of information. More specifically we will focus on using content-based filtering, which is a method that uses user preferences to determine whether the user would like a certain item. We can later expand to a more hybrid model that will start using the user's interactions with the website to generate more accurate recommendations. Recommendation systems use algorithms such as user clusters, tag affinity, item activity correlations, topic models, content similarity and more. The primary recommendation algorithm in our case will be the content similarity, which makes recommendations based on user profile preferences.

**Activity Diagram:**



**Diagram 4.a**

## b. Data Structures

The Weight Watcher's page will maintain a leaderboard of all active users to motivate them in achieving their goals. This leaderboard will display points; a sort of reward system for accomplishing challenges and personal goals which are publicly displayed on a leaderboard to encourage friendly competition. This leaderboard will use a binary search tree (BST) to maintain an order of scores. When a user gains points, the node reflecting the user's information will be deleted and re-inserted into the BST in the correct position. BST's will prove to be effective here because they maintain order and allow for  $O(\log n)$  deletions and insertions. An inorder traversal of the tree, that is right  $\rightarrow$  root  $\rightarrow$  left will be sufficient in displaying the leaderboard results in the appropriate order from first to last on the Weight Watcher's page.

# 5. User Interface Design and Implementation

## UC 1: Get a Meal Plan

This use case only requires a few clicks to get to the desired page and to lookup the recipes provided for their meal plan. First, the user clicks on Go to Profile and then clicks on Your Meal Plan. From there, the user selects the Full Recipe for either Breakfast, Lunch or Dinner and can view the detailed recipe. This use case is maximizing the ease of use for the user since it only requires 3 clicks to get to the recipe page.

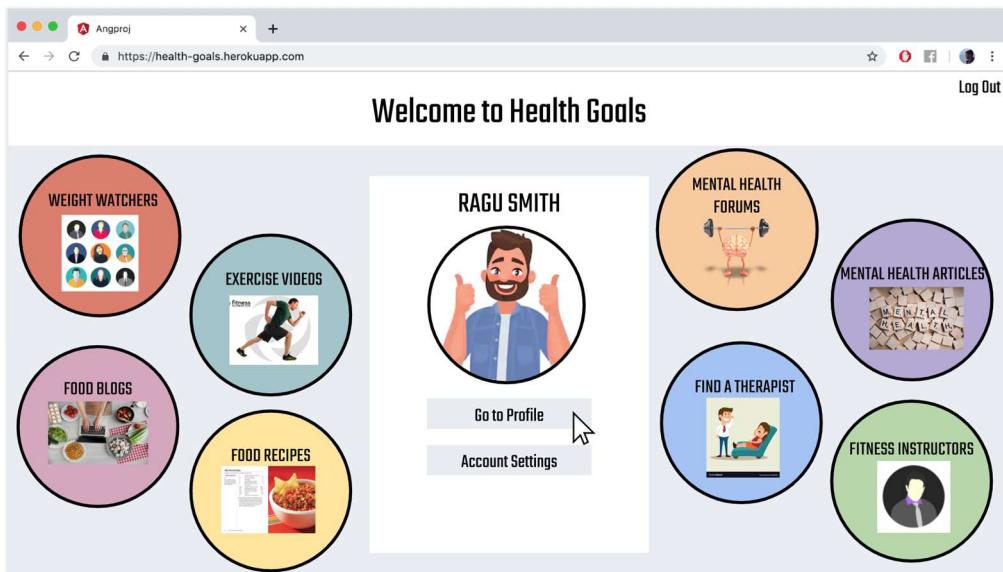


Figure 5.1.1 Welcome Page

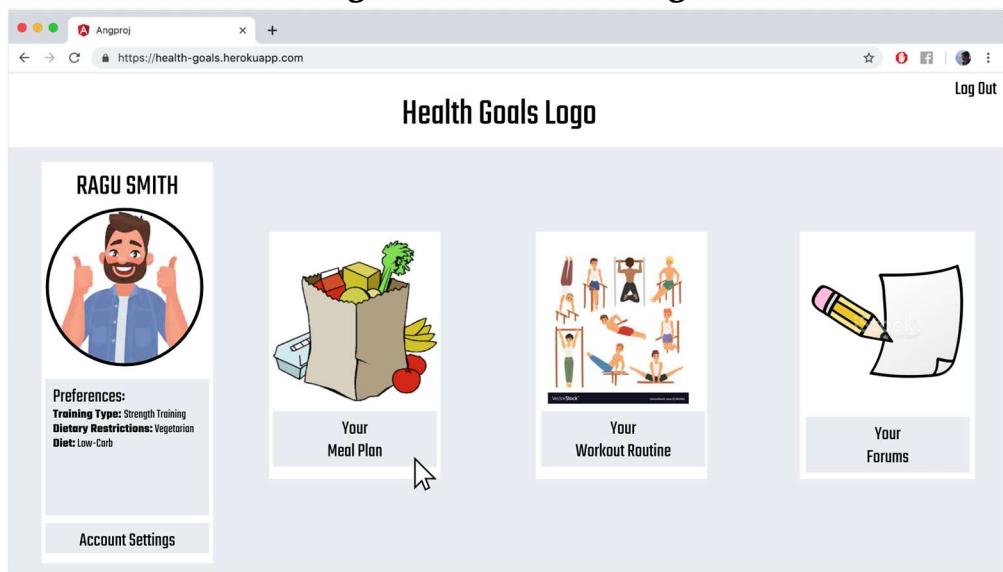


Figure 5.1.2 User Profile

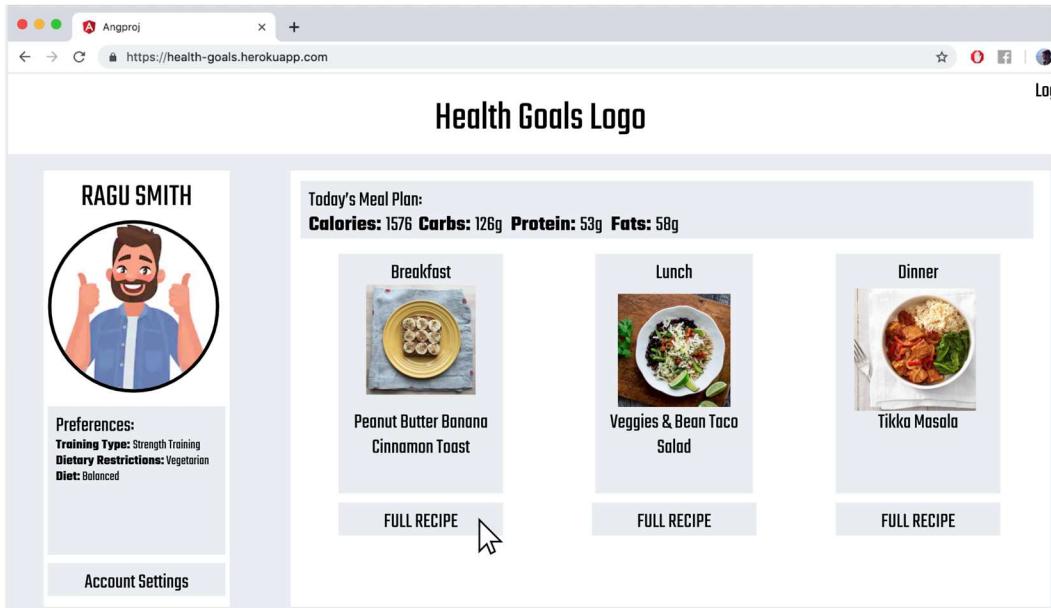


Figure 5.1.3 Meal Plan Page

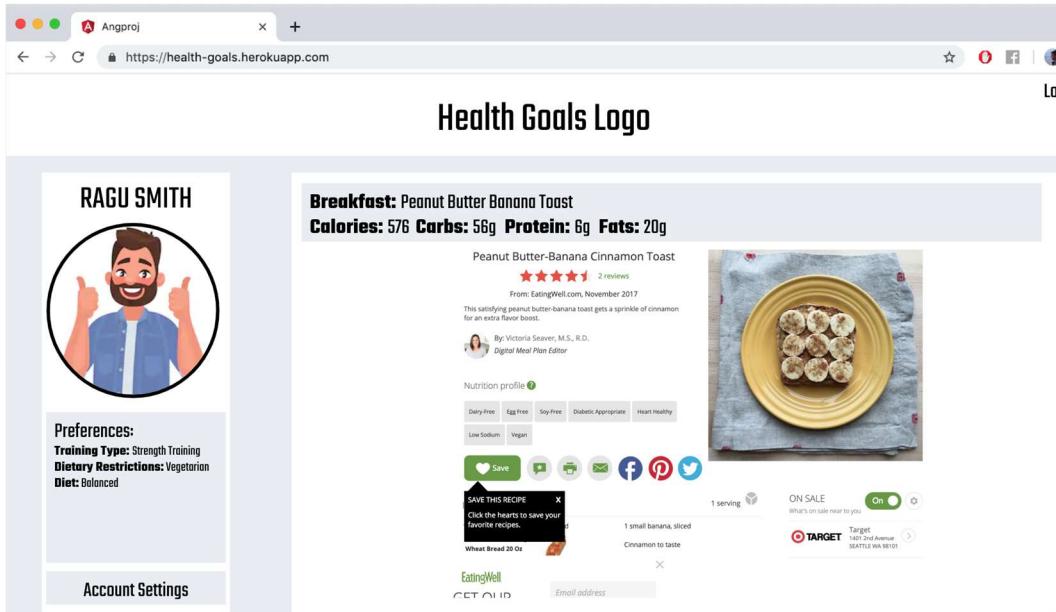


Figure 5.1.4 Recipe Page

## UC 2: Mental Health Match

This use case only requires a few clicks to find a therapist for a user. First, the user clicks on Find a Therapist icon on the main screen and then selects some filters which narrows down the number of therapists. From the list of therapists shown, the user will click on the therapist they prefer and can read more information about them. This

use case is maximizing the ease of use for the user since it only requires 3 UI Navigation clicks and a few Data Entry clicks to get to the preferred therapist's information page.

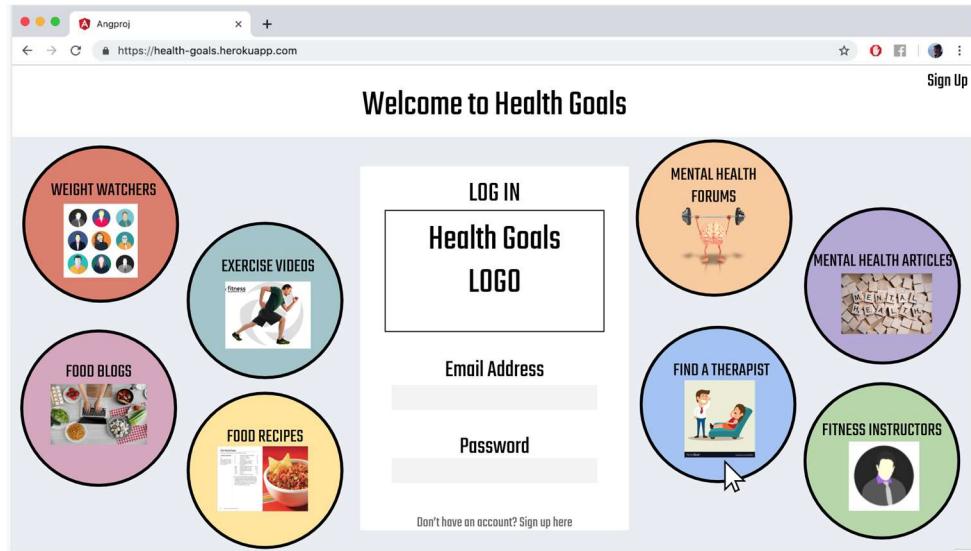


Figure 5.2.1 Login Page

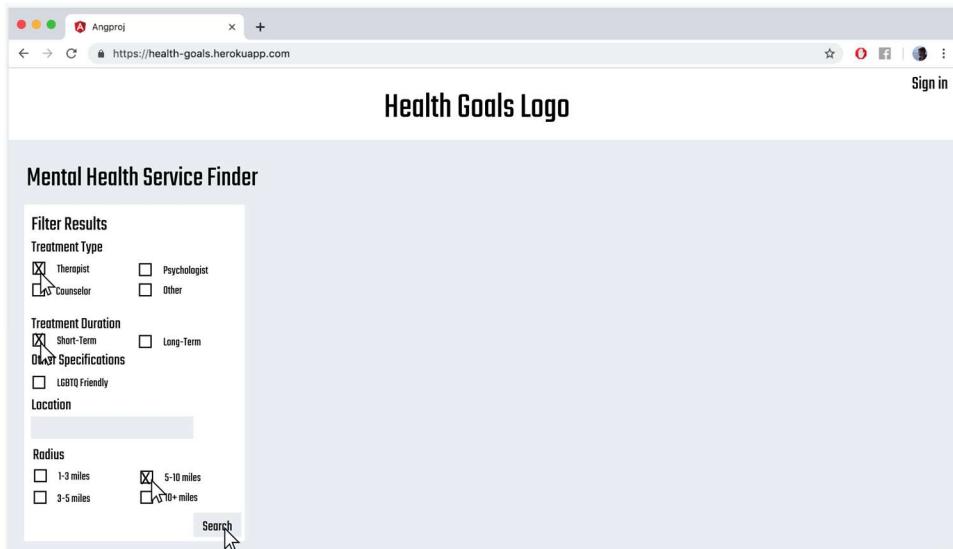


Figure 5.2.2 MH Service Page

**Health Goals Logo**

**Mental Health Service Finder**

**Filter Results**

**Treatment Type**

- Therapist       Psychologist
- Counselor       Other

**Treatment Duration**

- Short-Term       Long-Term

**Other Specifications**

- LGBTQ Friendly

**Location**

**Radius**

- 1-3 miles       5-10 miles
- 3-5 miles       10+ miles

**Search**

**Yonela G. Stephenson**

**Noreen Iqbal**

**Melissa Higgins**

**Figure 5.2.3 MH Service Finder Page**

**NOREEN IQBAL, LCSW**

Welcome to our group therapy practice. I am the owner and director of the Olive Branch Therapy Group, where I hope you are able to find a therapist to walk with you on your life journey.

Are you new to therapy? Rest assured, for most of my clients, this was their first time in therapy. Many times, we come from backgrounds, cultures, and religions that don't believe in therapy. The therapeutic process can be an integral tool in helping you help yourself and motivating you towards positive change. Therapy allows us that moment to invest in our healing. One day, life catches up with us and we may feel that our balance is off. This is when sadness and anxiety may set in.

Often, with our busy lives, we are not given the opportunity to heal from our past. We all need that moment. That time to heal our lifetime wounds. Whether it is childhood sexual abuse, physical abuse, trauma, an abusive marriage or even gaining self-confidence, make your healing a priority. Allow me the chance to help you believe that there are better days ahead.

Relationships can be challenging. Many of us feel we weren't properly prepared for the roles we take on. Parenthood can take a toll on us. Our intimate relationships- dating and marriage require work. Whether you are struggling with communication, dealing with infidelity, lack of intimacy or building an arranged marriage, therapy can help. I offer premarital therapy, and marriage counseling in a nonjudgmental, warm confidential space. Therapy can help you figure out what you want and need in a relationship and how to get it.

I deeply understand, honor and respect the South Asian and Middle Eastern religions and cultures. I am fluent in Urdu, Hindi, Hinko and Punjabi. Take the steps to live your best life. Your story isn't over yet. I look forward to speaking with you. Contact me, If you would like to schedule a complimentary 15-minute consultation, appointment or have insurance questions.

**Figure 5.2.4 Therapist Information Page**

## UC 4: Start Mental Health Forum

This use case only requires a few clicks for a user to start a forum. First, the user clicks on Mental Health Forums icon on the main screen and then clicks on Start a Forum. On the forum page, the user enters the title, the description and can chose to select relevant topics. This use case is maximizing the ease of use for the user since it only requires 3 UI Navigation clicks and some keyboard strokes for a user to start a forum.



Figure 5.3.1 Login Page

A screenshot of a web browser showing the 'Health Goals Logo' page. The top navigation bar includes a 'Sign in' link. The main content area has a heading 'Mental Health Forums: A safe discussion space for anyone who needs it'. Below this are several sections: 'Filter Results' (with a 'Mental Health Topics' dropdown containing options like Depression, Anxiety, PTSD, General, Men's Health, Women's Health, LGBTQ, and Suicide Prevention), a 'Start a Forum' button, a search bar, and links for 'Newest First' and 'Page 1 of 10'. There are also three forum posts: 'Should I See a Therapist?' (Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco), 'No Motivation' (Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco), and 'I Have a Story to Share' (Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco). On the right side, there is a 'Active Forums' sidebar with ten entries, each labeled 'Title'.

Figure 5.3.2 MH Forum Page

**Figure 5.3.3 MH creation Page**

### UC 6: Get Workouts

This use case only requires a few clicks to get various exercise videos. First, the user clicks on the Exercise Videos icon on the main screen and then selects some filters which narrows down the exercises they prefer. From the list of exercise videos shown, the user will click on the video they prefer and can view the video and read the steps needed for that exercise. This use case is maximizing the ease of use for the user since it only requires 2 UI Navigation clicks and a few Data Entry clicks to get to the preferred video's page.

**Figure 5.4.1 Login Page**

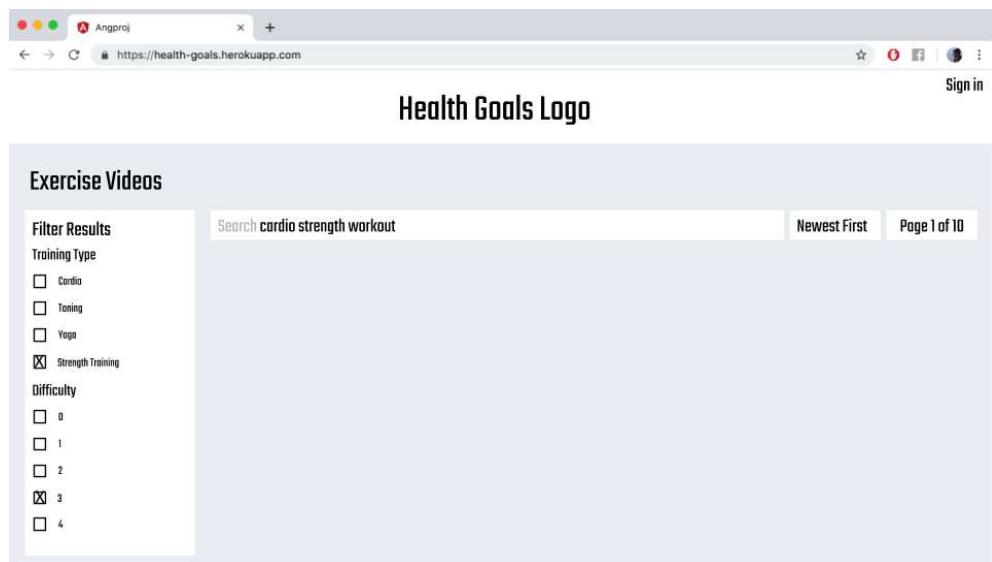


Figure 5.4.2 Exercise Search Page

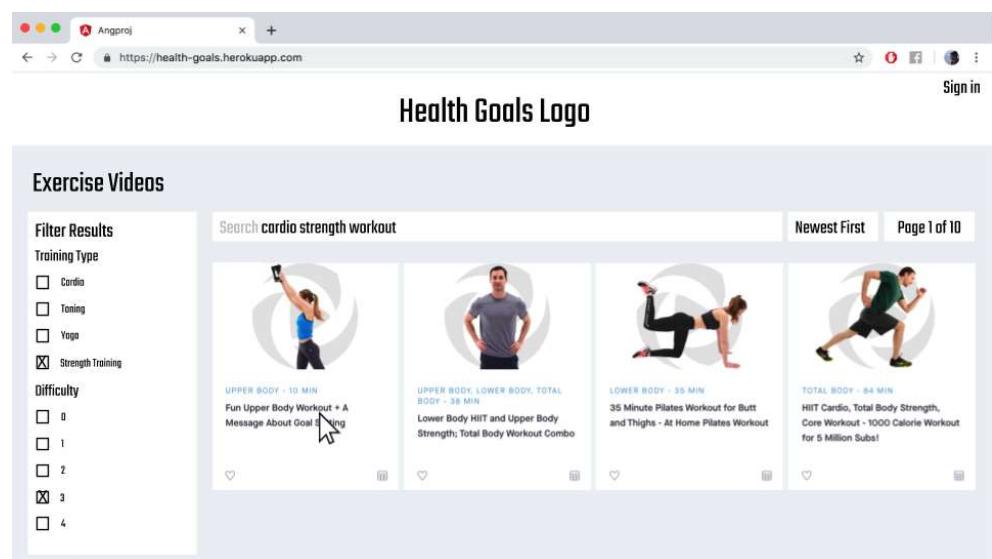
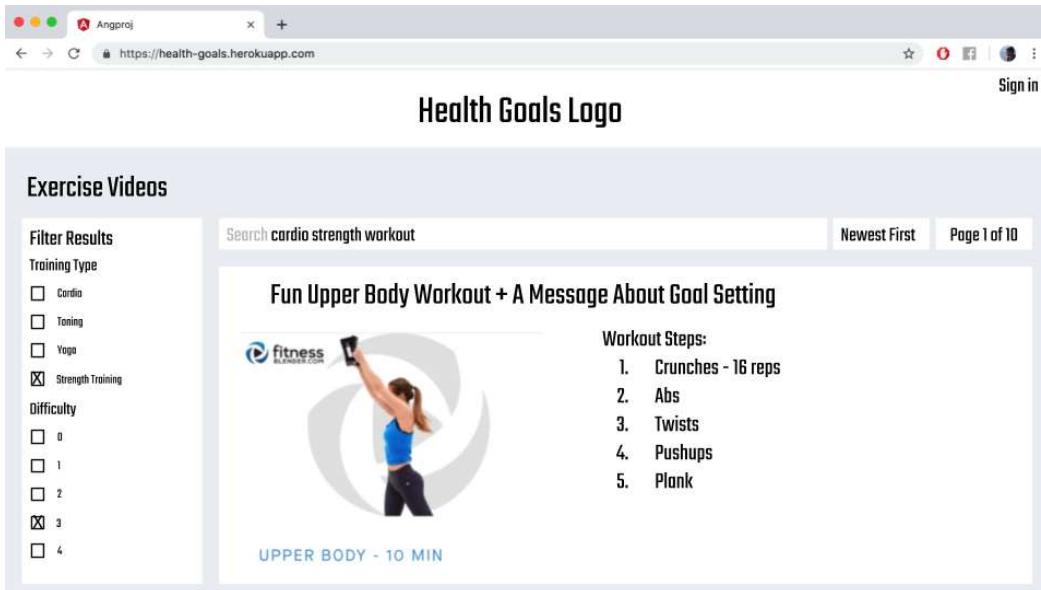


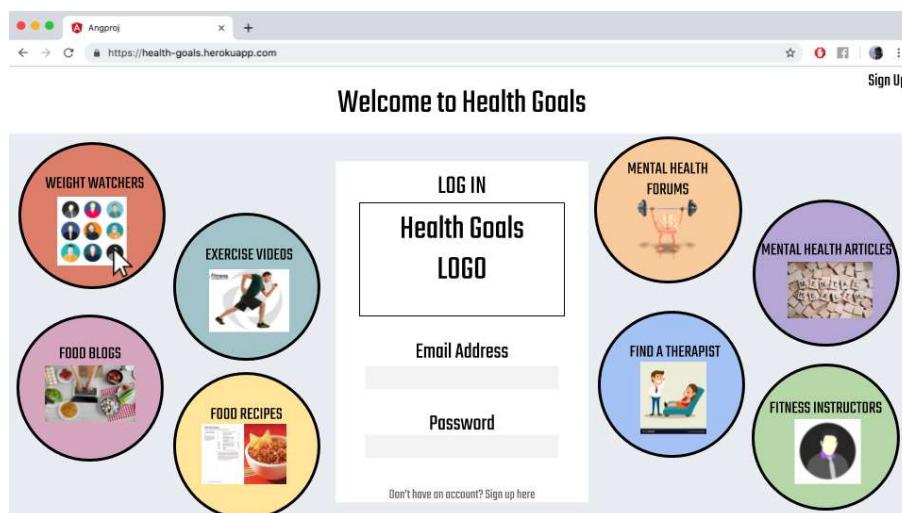
Figure 5.4.3 Exercise Options Page



**Figure 5.4.4 Exercise description Page**

### UC-9: Join Group

This use case only requires a few clicks for a user to join a group. First, the user clicks on the Weight Watchers icon on the main screen and then selects some filters which narrows down the groups they would be most interested in. From the list of groups, the user can view a specific group's information and can easily join that group by putting the group's name in the group entry row. This use case is maximizing the ease of use for the user since it only requires 2 UI Navigation clicks and a few Data Entry clicks to get to the preferred group's page and join it.



**Figure 5.5.1 Login Page**



**Figure 5.5.2 Find group Page**

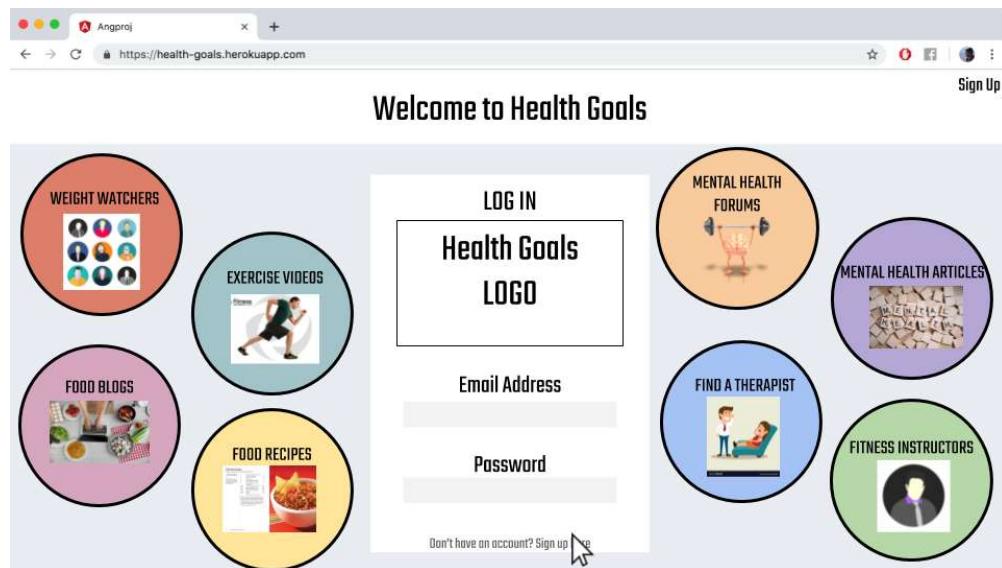
The screenshot shows a web browser window with the URL <https://health-goals.herokuapp.com>. The title bar says "Health Goals Logo". On the right, there is a "Sign in" link. The main content area has a header "Weight Watchers". To the left, there is a "Join Group" section with a "Type name here" input field, a "Find" button, and a "Join" button. To the right, there is a "Cardio Crew" section with a "Members" heading and a grid of 9 user icons. Below this is a "Top 10 Leaderboard" table with columns "Name", "Rank", and "% Goal Reached". The table currently has 10 empty rows.

Name	Rank	% Goal Reached

**Figure 5.5.3 Group description**

## UC-10: Create Account

This use case only requires a few clicks for a user to create an account. First, the user clicks on the Sign Up link either on the top right hand corner or on the main page. From there, the user fills out their personal information and can add a picture if they want. This use case is maximizing the ease of use for the user since it only requires 1 UI Navigation click and some keyboard strokes to create an account.



**Figure 5.6.1 Login Page**



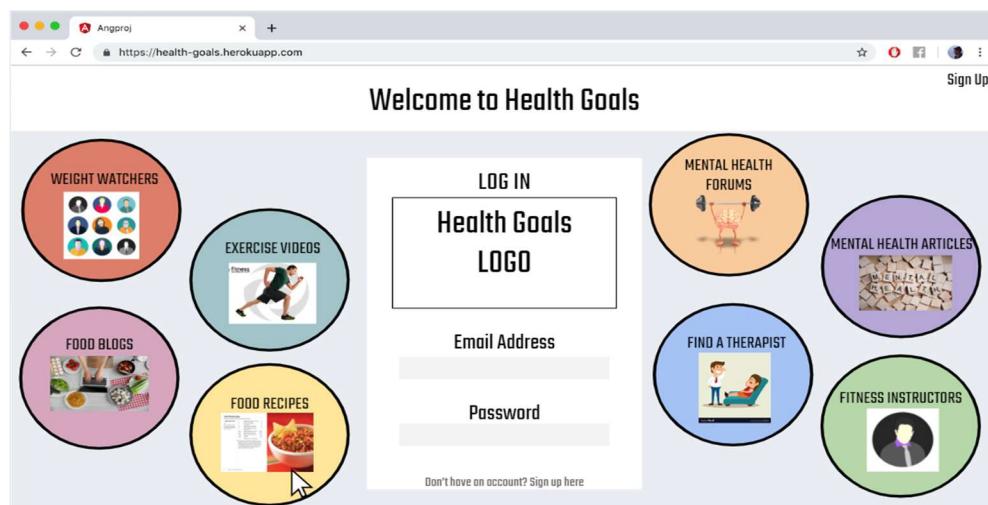
**Figure 5.6.2 Create account Page**



**Figure 5.6.3 Account Page**

#### UC-14: Search Recipes

This use case only requires a few clicks to get various food recipes. First, the user clicks on the Food Recipes icon on the main screen and then selects some filters which narrows down the recipes they prefer. The page will show the top 10 trending recipes as well. From the list of recipes shown, the user will click More Info next to the recipe they prefer and can view the ingredients and directions for that recipe. This use case is maximizing the ease of use for the user since it only requires 2 UI Navigation clicks and a few Data Entry clicks to get to the preferred recipes page.



**Figure 5.7.1 Login Page**

The screenshot shows a web browser window with the URL <https://health-goals.herokuapp.com>. The title bar says "Angproj". The main content area has a header "Health Goals Logo" and a sub-header "Recipes". On the left, there is a "Filter Results" sidebar with sections for "Meal Type" (Breakfast, Lunch, Dinner, Snack, Drink), "Additional" (Convenience, Cost, Ratings), and a "Custom Filters" button. The search bar contains the word "Chicken". Below the search bar, there are three recipe cards:

- Lemon Herb Chicken**: Basic Nutrition: Calories - 212, Servings - 2. Diet Labels: Lean. Health Labels: Low-Fat, High Protein.
- Grilled Rosemary Chicken**: Basic Nutrition: Calories - 1256, Servings - 2. Diet Labels: Lean. Health Labels: Low-Fat, High Protein.
- Summer Chicken Burgers**: Basic Nutrition: Calories - 630, Servings - 1. Diet Labels: Lean, High Protein. Health Labels: Multigrain, High Protein.

Each card has a "More Info" button. To the right of the cards is a "Trending Recipes" sidebar with a list of 10 items: Peaches, Fruit Salad, Bagel Avocado Sandwich, Tomato Basil Salmon, Lemon Herb Chicken, Grilled Rosemary Chicken, Lentil Soup, Easy Herb Potatoes, Fresh Green Bean Salad, and Roasted Garlic Cauliflower.

Figure 5.7.2 Recipes search Page

This screenshot shows the same web browser and URL as Figure 5.7.2. The main content area now displays a detailed recipe for "Simple Lemon Herb Chicken". The top part of the recipe card includes the title, a 5-star rating, the number of reviews (196), and the number of photos (22). Below this is a thumbnail image of the dish. The "Directions" section contains two numbered steps:

- Cut lemon in half, and squeeze juice from 1/2 lemon on chicken. Season with salt to taste. Let sit while you heat oil in a small skillet over medium low heat.
- When oil is hot, put chicken in skillet. As you saute chicken, add juice from other 1/2 lemon, pepper to taste, and oregano. Saute for 5 to 10 minutes each side, or until juices run clear. Serve with parsley for garnish.

The "Ingredients" section lists the following items with their quantities:

- 2 skinless, boneless chicken breast halves
- 1 tablespoon olive oil
- 1 lemon
- 1 pinch dried oregano
- salt and pepper to taste
- 2 springs fresh parsley, for garnish

The "Nutrition Facts" section provides nutritional information per serving: 212 calories, 8.6 g fat, 7.9 g carbohydrates, 28.8 g protein, 68 mg cholesterol, and 94 mg sodium. A "Full nutrition" link is also present. An "Add to Meal Plan" button is located at the bottom right of the recipe card. The right sidebar "Trending Recipes" is identical to Figure 5.7.2.

Figure 5.7.3 Recipes Page

### UC-15: Search Info

This use case only requires a few clicks to view mental health articles relating to the user. First, the user clicks on the Mental Health Articles icon on the main screen and then selects some filters which narrows down the articles they wish to read about. The page will show the top 10 trending articles as well. From the list of articles shown, the user will click Learn More next to the articles they wish to read and can view and read the entire article. This use case is maximizing the ease of use for the user since it only requires 2 UI Navigation clicks, a few Data Entry clicks and some keyboard strokes to get to the preferred article's page.

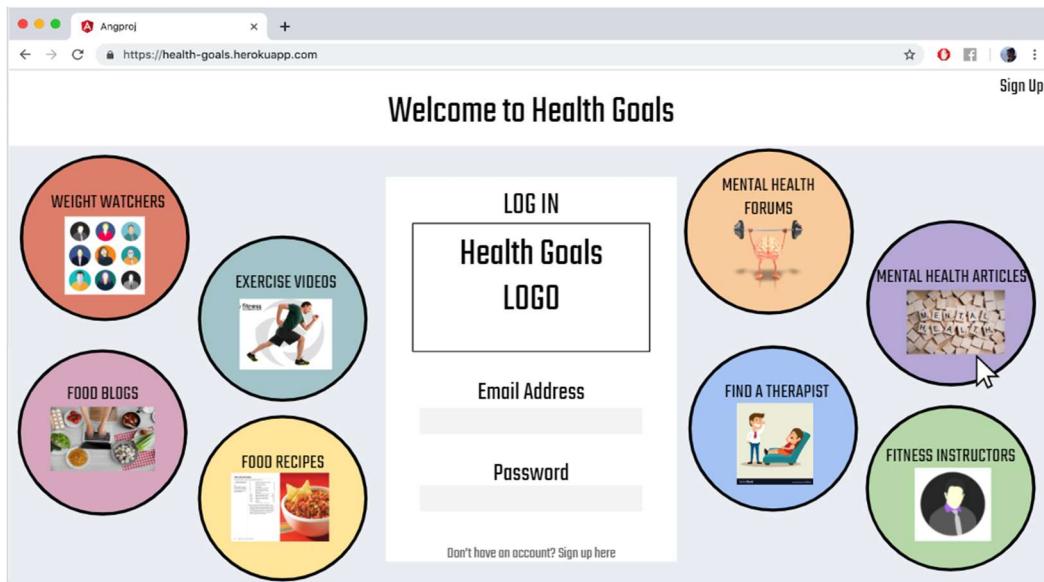


Figure 5.8.1 Login Page

The screenshot shows a web browser window with the URL <https://health-goals.herokuapp.com>. The page title is "Health Goals Logo". On the left, there's a sidebar titled "Mental Health Resources" with a "Filter Results" section containing a checkbox for "Anxiety" which is checked. Below it is a list of mental health topics: Depression, Anxiety, PTSD, General, Men's Health, Women's Health, LGBTQ, and Suicide Prevention. The main content area has a search bar with the query "How to overcome stress", sorting options "Newest First" and "Page 1 of 10", and a "Top 10 Articles" sidebar on the right. The first article in the list is about the effects of meditation.

Figure 5.8.2 MH Search Page

This screenshot shows the same web browser window as Figure 5.8.2, but the search results have been refined to show a single article. The article title is "How to overcome stress" and the abstract discusses the effects of Raja Yoga meditation on respiratory functions, cardiovascular parameters, and lipid profile. The sidebar on the right remains the same, listing the top 10 articles.

Figure 5.8.3 MH article Page

## **6. Design of Tests**

### **1. View Main Pages**

- In this test, each and every page is to be opened. All features should load on the main site.

### **2. Search for Mental Health Service**

- In this test, the search function for the mental health service will be tested. The treatment type, duration, specifications, locations, and/or name options in all configurations must be checked. The sort option should sort by newest, some form of rating, and most viewed. When selected, the site's page should load and the links off the site must function

### **3. Search for Workouts**

- In this test, the search function for the types of workouts will be tested. The training type and difficulty level options must be checked in all possible configurations. The sort option should sort by newest, some form of rating, and most viewed. A page with information on the workouts, with visuals of the workout and text instructions should appear.

### **4. Search Articles**

- In this test, the search function for the assorted health articles will be tested. The major topic filter and search bar must be checked. The sort option should sort by newest, some form of rating, and most viewed.

### **5. View Meal Plans**

- In this test, the basic meal plan page will be tested. The page should contain three meals, and have the sum of the calorie and nutrition information listed

### **6. Search Recipes**

- In this test, the search function for recipes will be tested. The filters must be tested in all combinations (search bar, meal type, etc.). The sort option should sort by newest, some form of rating, and most viewed. The actual instruction should appear when the information button is selected.

## **7. View Forums**

- In this test, forums should be accessed. The forum topics should be visible and a visitor may scroll through them. They should not have access to search, edit, and post options.

## **8. View Group Types**

- In this test, the groups should be accessed. The public group names should be visible and a visitor may scroll through them. They should not have access to search, create, and options.

### **The Test Coverage and Integration Testing Format**

Test: View Main Pages
Assumption: The homepage loads correctly
Steps:
<ol style="list-style-type: none"><li>1. Visitor clicks a page icon from the home screen</li><li>2. The indicated page loads</li></ol>
Success State: The page selected by the user loads correctly
Failure State: An incorrect page loads, or no page loads. All features should load on the main site.

**Table 6.1**

Test: Search for Mental Health Service

Assumption: A visitor has opened the page

Steps:

1. The visitor enters filters, and/or types a search option in the search bar
2. The options load
3. The visitor selects a sort option (note: this may occur at any time before or after a search)
4. The search options are displayed
5. The visitor selects a page
6. The page loads, and any URLs are active

Success State: The page loads, and the filters correctly restrict searches. The search bar delivers relevant results. The selected search items open new pages.

Failure State: The pages fail to load, the filters fail to restrict options, and the search bar doesn't bring relevant results. The selected search items do not open to the correct page, or fail to open. The sort icon also does not sort the options.

**Table 6.2**

**Test: Search Workouts**

Assumption:

Steps:

1. The visitor enters filters, and/or types a search option in the search bar
2. The options load
3. The visitor selects a sort option (note: this may occur at any time before or after a search)
4. The search options are displayed
5. The visitor selects a page

The page loads, and any URLs are active

Success State: The page loads, and the filters correctly restrict searches. The search bar delivers relevant results. The selected search items open new pages.

Failure State: The pages fail to load, the filters fail to restrict options, and the search bar doesn't bring relevant results. The selected search items do not open to the correct page, or fail to open. The sort icon also does not sort the options.

**Table 6.3**

**Test: Search Articles**

Assumption: A visitor has clicked the “article” icon and the page loaded

Steps:

1. The visitor enters filters, and/or types a search option in the search bar
2. The options load
3. The visitor selects a sort option (note: this may occur at any time before or after a search)
4. The search options are displayed
5. The visitor selects a page
6. The page loads, and any URLs are active

Success State: The page loads, and the filters correctly restrict searches. The search bar delivers relevant results. The selected search items open new pages.

Failure State: The pages fail to load, the filters fail to restrict options, and the search bar doesn't bring relevant results. The selected search items do not open to the correct page, or fail to open. The sort icon also does not sort the options.

**Table 6.4**

Test: View Meal Plans

Assumption: The meal plan icon has been clicked, and its page loaded

Steps:

1. The visitor may view a sample meal plan
2. The calorie and nutritional information is summed up and displayed above

Success State: The visitor may open the page and view the sample information.

Failure State: The page fails to load, or the visitor gets user-level access, which takes previous selections into account. The visitor is able to click the images and get more data on them

**Table 6.5**

Test: Search Recipes

Assumption: The recipe icon is selected and its page loads

<p>Steps:</p> <ol style="list-style-type: none"> <li>1. The visitor enters filters, and/or types a search option in the search bar</li> <li>2. The options load</li> <li>3. The visitor selects a sort option (note: this may occur at any time before or after a search)</li> <li>4. The search options are displayed</li> <li>5. The visitor selects a page</li> <li>6. The page loads, and any URLs are active</li> </ol>
--

Success State: The page loads, and the filters correctly restrict searches. The search bar delivers relevant results. The selected search items open new pages.

Failure State: The pages fail to load, the filters fail to restrict options, and the search bar doesn't bring relevant results. The selected search items do not open to the correct page, or fail to open. The sort icon also does not sort the options.

**Table 6.6**

<p>Test: View Forums</p>
<p>Assumption: The forum icon is selected</p>
<p>Steps:</p> <ol style="list-style-type: none"> <li>1. The forum names load</li> <li>2. The forums load in newest to oldest order</li> </ol>
<p>Success State: The forums load in the aforementioned form. The visitor cannot access a particular forum, nor may they post.</p>

Failure State: The forums fail to load, or the visitor gains access to the forum. The visitor creates new posts and edits. The visitor accesses the search function.

**Table 6.7**

Test: View Groups
Assumption: The group icon has been selected, and its page loads
Steps: <ol style="list-style-type: none"><li>1. The group names load</li><li>2. The visitor may use a small icon on the bottom right of the screen to see older groups</li></ol>
Success State: The visitor sees the newest group names. The visitor may click the icon to see older ones.
Failure State: The visitor creates, deletes, or joins a group. The visitor accesses the search function.

**Table 6.8**

## 7. Project Management and Plan of Work

### a. Merging the Contributions from Individual Team Members

	Project Report #2 - Part 1	Project Report #2 - Part 2	Project Report #2 - Part 3
Subgroup 1	Responsible for creating interaction diagram under Section 1 for Use Case - 1, Use Case - 10 & Use Case - 15. Also responsible for submitting the report.	Responsible for finishing the class diagrams under Section 2 and answering questions for persistent data storage and global control flow under section 3.	Responsible for completing algorithms and data structures of section 4 and cleaning up any inconsistencies present in section 1 with all the UML diagrams. Also responsible for cleaning up the document and submitting the report.
Subgroup 2	Responsible for creating interaction diagram under Section 1 for Use Case - 6, Use Case - 9 & Use Case - 14. Also responsible for cleaning up the format.	Responsible for finishing the Traceability Matrix and its description under Section 2 and mapping subsystems and describing hardware requirements under section 3. Also responsible for cleaning up the format.	Responsible for completing the user interface design and implementation of section 5 and cleaning up any inconsistencies present in section 2 of class diagrams and interface specification. Also responsible for finishing sections of project management and plan of work.
Subgroup 3	Responsible for creating interaction diagram under Section 1 for Use Case - 2, Use Case -	Responsible for finishing the data types and operation signatures under Section 2 and	Responsible for completing the design of tests of section 6 and cleaning up any

	4 & Project Management	describing architectural styles and subsystems under section 3. Also responsible for submitting the report.	inconsistencies present in section 3 of system architecture and design. Also responsible for finishing sections of project management and plan of work.
--	------------------------	---	---

**Table 7.1**

Our team of nine people is divided into three subgroups and during our weekly meetings, we assign different subgroups different sections to work on. This ensures that everyone gets a chance to work on our project report and through proper planning we meet our deadlines every week. There can be certain inconsistencies between sections on our report, however before submitting our report:

- We go through all the sections and make sure the information is accurate and we have answered all the questions
- Every week, one subgroup is assigned the task of cleaning up the formatting and appearance of the report to make it look uniform
- If there are any inconsistencies, the subgroups responsible for the two sections have a discussion
- We have group chat on which every member of the group actively participates and voices their concern or opinions to ensure clear communication within the group

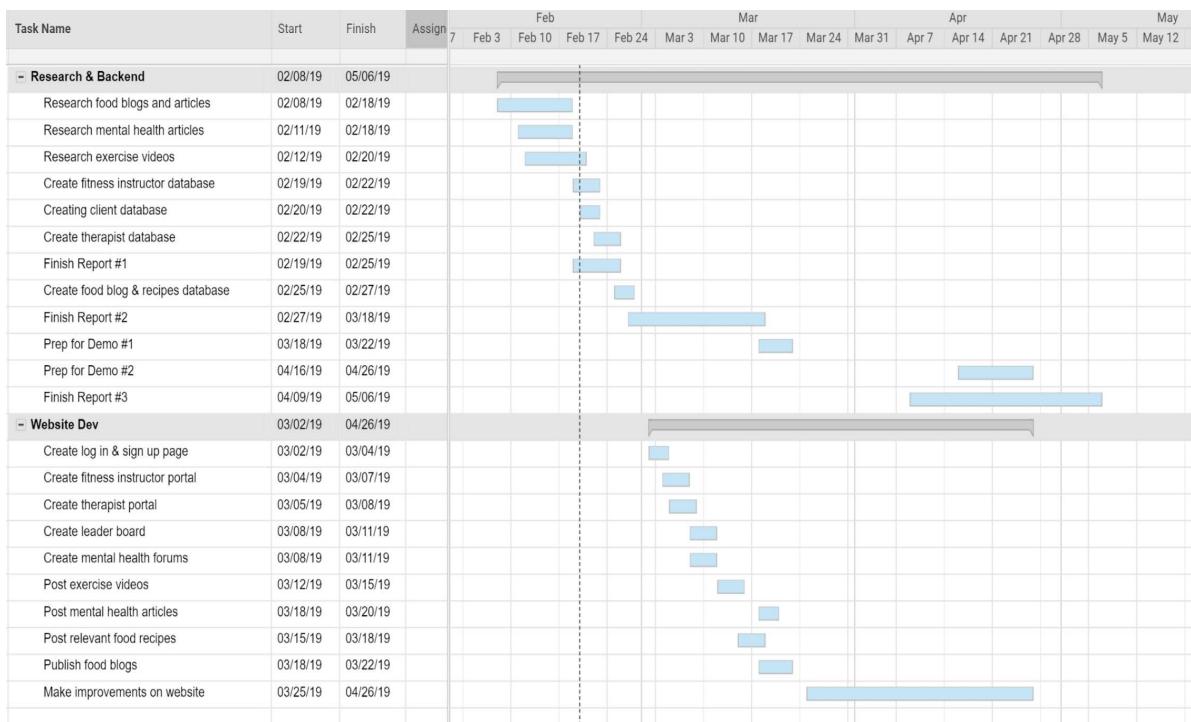
It is evident to us that clear communication is key when we try to work on a project of this scale because we are all working on the document remotely and on our own time. Our weekly meetings give us a chance to plan out our schedule beforehand and consider situations where one section might depend on another section being finished previously. Keeping all of this in mind our weekly meetings allow us to stay on track and finish our project reports in a timely fashion.

## b. Project Coordination and Progress Report

Currently, all the features are a work in progress however we are planning on implementing the use cases that the visitors would use since we are demonstrating only these features in demo #1. Use cases such as Get Meal Plan, Mental Health Match, Get Workout, Search Recipe, and Search Info will be developed and demonstrated in class. In order to get a working prototype by our demo date, we are working throughout this week to further develop and look over each subgroup features. Since it

is spring break it is harder to communicate, therefore we will need to set hard deadlines well before the demonstration date in that way we can prepare our presentation and we have already begun to do so. Our Gantt charts help us stay on track regarding our features. Using resources such as GitHub we have started pushing our code into the repository which allows us to stay on track although we are all working remotely.

## c. Plan of Work



**Figure 7.2**

All members will be actively working and helping each other on every part. After submitting report #2 until the semester ends, we decided to allocate responsibility for each of the following components to each of the three subgroups and the remaining subgroups will be helping where needed.

Nutrition Related Features(Food blogs & articles, Food Recipes): Subgroup 1

- Kishan
- Arti
- Anushka

Fitness Related Features(Exercise Videos, Fitness Instructor Portal/Database, WeightWatchers(Leaderboard): Subgroup 2

- Niharika
- Varun B.
- Sai

Mental Health Related Features(Mental Health Articles & Forum, Therapist Portal/Database): Subgroup 3

- Eric
- Varun R.
- Jose

## d. Breakdown of Responsibilities

We currently have a GitHub repository set up where all our team members/subgroups get a chance to develop their individual features and publish the changes to our website. We have no assigned group manager who looks over all the merges that are being made but we all notify each other before any major changes are being pushed to our master branch or website. Integration testing is performed by the team member responsible for making the changes, or in our case, subgroups as a whole. This way we have a chance to expose any defects in the interface and how it interacts with the system components. This ensures that we check how the unit test works together, as a whole system. By assigning one subgroup to look over it we ensure that we don't run into any other problems while further developing our website.

The following table reflects what research and attributes are reflected in each of the functional features as well as which subgroup is responsible for developing, coding, and testing each of the features. We will keep all subgroups updated on progress and any complications so that we can work together as a team to help any subgroup resolve any issues they may come across.

Functional Feature and Description	Main Subgroup Responsible
Welcome Page → Create a visually appealing and intriguing home page to attract users, reveal the site's underlying content, and provide easy navigation to access the other features.	Subgroup 1
Weight Watchers → Implement a community to maintain and reflect the progression of user goals. Top 10 people who have achieved the highest percentage of goal reached will be displayed in the group's leaderboard.	Subgroup 2
Fitness Exercises → Display exercises that will be filtered by type of training (cardio, toning, yoga, strength training) type of difficulty ranging from one to five, five being the hardest.	Subgroup 2
Mental Health Forum → Implement an online discussion topics feature where users can share and contribute to their experiences. Allow users to filter by depression, anxiety, PTSD, general, men's health, women's health, LGBTQ, suicide prevention.	Subgroup 3
Nutrition Blogs/Articles → Allow users to access nutrition relevant articles filtered based on the type of food(vegan, keto, kid-friendly, breakfast ) and dietary restrictions (dairy-free, gluten-free, vegetarian, low carb, non-gmo).	Subgroup 1
Find A Therapist → Users will be able to filter through therapists based on treatment type(therapist, psychologist, counselor, other), duration and location of treatment. Also, allow the user to make LGBTQ specification.	Subgroup 3
Mental Health Articles → Allow users to access mental health relevant articles filtered based on the type of mental health (depression, anxiety, PTSD, general, men's health, women's health, LGBTQ, suicide prevention)	Subgroup 3

<p>Nutrition Recipes - Implementing a portal of recipes for users to create an effective meal-plan based on the users' health and diet goals. Filter based on meal type (breakfast, lunch, dinner, snack, drink) and additionally cost, convenience, rating.</p>	Subgroup 1
<p>Create Account - Website will collect pertinent information, users will be able to create an account and are prompted to answer questions pertaining to their age, sex, gender identity, sexual orientation, race/ethnicity, weight loss goals, and personal workout goals.</p>	Subgroup 1
<p>Fitness Instructor Matchup - Users will be able to filter through fitness instructors based on training type(cardio, toning, yoga, strength training), trainer sex, and location.</p>	Subgroup 2

**Table 7.3**

## 8. References

Fitbit Health Monitoring Analytics:

<http://eceweb1.rutgers.edu/~marsic/books/SE/projects/HealthMonitor/2014-g5-report3.pdf>

UML Class Diagram Tutorial:

<https://www.youtube.com/watch?v=UI6lqHOVHic>

UML Activity Diagram Tutorial:

<https://www.lucidchart.com/pages/uml-activity-diagram>

Gantt Chart Assistance:

<https://www.smartsheet.com/blog/gantt-chart-excel>

Project Management Examples:

<https://marketbusinessnews.com/financial-glossary/project-management/>

Understanding Unit Testing:

<https://www.softwaretestingclass.com/what-is-difference-between-unit-testing-and-integration-testing/>