

Health Goals

Software Engineering (332:452)

Final Report 3

Group Number: 1

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

Submission Date: May 5th, 2019

Group Members:

Anushka Desai

Varun Bhandari

Niharika Mishra

Sai Singanamala

Arti Patel

Kishan Patel

Jose Cuentas

Eric Rivera

Varun Raghuraman

Table of Contents

Summary of Changes	3
1. Customer Problem Statement	4
a. Introduction	4
i. Health & Lifestyle Changes	4
ii. Mental Health Stats & Significance	5
iii. Our Approach	6
2. Glossary of Terms	8
3. System Requirements	9
a. Enumerated Functional Requirements:	9
b. Enumerated Nonfunctional Requirements:	11
c. On-Screen Appearance Requirements:	11
4. Functional Requirements Specification	177
a. Stakeholders	187
b. Actors and Goals	187
c. Use Cases	198
i. Casual Description	198
ii. Use Case Diagram	210
iii. Traceability Matrix	210
iv. Fully-Dressed Description	221
d. System Sequence Diagrams	276
5. Effort Estimation using Use Case Points	30
a. UUCW – Unadjusted Use Case Weight	30
b. UAW – Unadjusted Actor Weight	32
c. TCF – Technical Complexity Factors	33
d. Environment Complexity Factor	34
6. Domain Analysis	365
a. Domain Model – For UC 1, 2, 4, 6, 9, 10, 14, 15	35
b. System Operation Contracts	542
7. Interaction Diagrams	563
8. Class Diagram and Interface Specification	652
a. Class Diagram	652
b. Data Types and Operation Signatures	652
c. Traceability Matrix	696
d. Object Constraint Language(OCL) Contracts	717

9. System Architecture and System Design	69
a. Architectural Styles	69
b. Identifying Subsystems	69
c. Mapping Subsystems to Hardware	69
d. Persistent Data Storage	730
e. Network Protocol	741
f. Global Control Flow	741
g. Hardware Requirements	741
10. Algorithms and Data Structures	751
a. Algorithms	751
b. Data Structures	762
11. User Interface Design and Implementation	773
12. Design of Tests	Error! Bookmark not defined.
13. History of Work, Current Status, and Future Work	94
a. History of Work	94
b. Key Accomplishments	95
c. Future Work	95
14. Project Management	96
12. References	1027

Summary of Changes

1. In the Interaction Diagrams section we decided to revise a few of our diagrams to include some design patterns as they make our functions much more optimal, convenient and adaptable.
2. Similarly in the Class Diagram and Interface Specification, we followed through on our modifications of applying specific design patterns. We went more in depth on their importance and benefits. We also listed some important OCL contracts for classes and their operations.
3. We revised a few of our design tests to apply to our website's current and future goals.
4. Added Object Constraint Language(OCL) Contracts to our class diagrams to further explain their purpose.
5. Elaborated on the use cases implemented in the final demo and updated sequence diagrams in our functional requirements specification.
6. Updated and involved the use cases completed for the final demo in User Interface Design and Implementation.

7. Added Duration to our Effort Estimation Calculation
8. Added history of work, current status, key accomplishments and future work

1. Customer Problem Statement

a. Introduction

i. Health & Lifestyle Changes

There is no denying that health is possibly the most important part of our life. We have a duty to ourselves to not only keep ourselves alive but to make sure we are happy and thriving. Good health is the easiest way to achieve long term happiness and peace and is a guaranteed way to live longer and stronger. Good health has many benefits: healthier weight, reduced risk of diseases, strong bones and teeth, better mental health, better sleep, more energy, and more. It is also scientifically proven that maintaining good health can help people keep many painful and dangerous risks at bay, such as cancer, diabetes, strokes, and other diseases. The two best ways to maintain good health are a balanced diet and physical activity. Despite us knowing this, given the technological advances of our world today, it's no surprise that many people ignore healthy habits for more immediate gratification in the form of junk food or lazing around.

As they say, *good things in life do not come easy*. A good, healthy lifestyle does not come easy either. It's much easier to open a bag of processed foods than to buy fresh grocery and prepare a meal. Many times people are too lazy to question, much less read, the ingredients and nutrition labels of what they are eating. Processed foods are loaded in preservatives, sugar, sodium, and fat. They are designed to make us addicted and overeat. Too much junk can lead to serious health risks such as obesity, high blood pressure, diabetes, and cancer. As customers ourselves, we college students find it difficult to avoid eating these foods. We either don't make the time or have the motivation to exercise and prepare healthy meals. Between studying, socializing, and attending student organization meetings/events, it is much easier for us to eat out and be lazy during our time off. It also doesn't help that we're surrounded by other students just like us. Not only does this apply to students, but to people in general.

When it comes to lifestyle changes like dieting, it can be very daunting investing the time to adjust into this new habit. Many people have very tight schedules; dieting is the last thing most would think about in their spare time. The by-product of the lack of time can be many unhealthy habits for the sake of convenience. With that in mind, Health Goals offers users a convenient, effective meal-plan based on the users' health and diet goals. Meals will be conveniently chosen for the user. This alleviates the stress and time it would normally take someone to decide what to eat given their diet. The users will be able to pick from multiple options to take into account all sorts of factors: the convenience of the meal, cost, portion size, etc. Usually, the hardest part of any problem is the planning phase; with that out the way, it's just a matter of discipline and execution of that plan. For further motivation, users will have access to the weight watchers community, where friends and others alike pursue similar goals to that of the user's. The intention here is to promote a fair bit of competition; most don't like falling behind or sticking out, especially when it comes to friends. The leaderboard will maintain and reflect the progression of user goals. Being public, this feature will incentivize users to try their best because it's human nature to strive for first place and being the best at what they're trying to do.

ii. Mental Health Stats & Significance

We believe that a lifestyle change is necessary to maintain health and wellness at all levels. The poor behavior we exhibit and the poor choices we make with physical activity and eating manifest as detrimental mental behaviors and consequences. This also applies in the opposite direction: neglecting mental health affects physical wellness. According to the National Alliance on Mental Illness, "Approximately 1 in 5 adults in the U.S.—43.8 million, or 18.5%—experiences mental illness in a given year" [1]. Besides the scale of mental illness, it is important to acknowledge that although not everyone has a mental condition, everyone has mental health and it must be maintained. Despite its significance, mental health and illness remains a sensitive and controversial topic, which makes it more difficult to recognize and treat. In 2014, less than half (41%) of US adults with mental health conditions received the appropriate services [1]. In 2018, we've only seen a 3% increase in service administered [2].

With such a prevalence of poor mental health, it is important to push for more general awareness and for more emphasis on a balance of mental and physical health. As a Health Goals user, I should be able to understand this balance and find the appropriate resources and environments to foster the improvement of my health. As a general user of online resources, I can search for resources on my own. I can try to find a recipe, an exercise routine, or tips to maintain a better state of mind, but it is

difficult to find it all in one place and further difficult to connect the pieces. Different pieces may contradict directly or implicitly affect each other negatively. With Health Goals, I should be able to use the single platform to find all the resources that can help me make changes to my daily routines, such as eating habits or meditation exercises for example, which are specific to my current level of health and my general profile. Moreover, they will present a holistic approach at improving health. The platform should facilitate one of the biggest challenges customers face, relevancy and quick access of information.

There are many online and mobile resources targeted at improving health, and many have made great strides at their respective goals. In addition, with a simple google search, one can find an article on the topic of their choosing and get some generalized information. More personalized takes more effort. One can also find websites and apps for self-reporting health trackers, client-counselor matching services, recipes and diets, or mental health forums and chats. These approaches take a targeted focus at improving one or two aspects of health while being targeted at general users. Although it is widely accepted that physical and mental health are connected, a limited number of tools try to address both simultaneously. A further limited amount of tools provide a truly personalized experience. As a user, I would like to see a comprehensive, all-in-one site that is personalized to my health level and experience.

iii. Our Approach

As mentioned earlier it is not only difficult to begin a healthy lifestyle but it is also difficult to maintain one. People can often start a diet or start working out but more often than not, they fall out of these good habits and tend to become lazier when it comes to maintaining their physical as well as their mental health. It is difficult to stay up to date with all the trends about new fitness routines and recommended diets. The goal of our website is to offer diets and fitness routines for each individual which will promote a healthy lifestyle and gather a community. We hope that this entices our clients to try the exciting, new ways of keeping their mind and body healthy and also motivate them to remain a member of the community. The captivating factor of Health Goals is that it encompasses many different aspects of wellness and does not just focus on just one feature specifically. It provides them with dieting information, along with fitness and mental health sections which allows them to get all the information they need in one place. This can be helpful because many times while researching a single topic, we refer to multiple different websites or we use different apps to keep up with different aspects of our lives when it comes to mental and

physical health. This website offers a variety of resources in once place which helps promote health and wellness and a sense of community.

To execute our goal we decided to break our community into two aspects, mental health and physical health which was then further divided into different features. Because our platform encourages the importance of lifestyle changes and balance, Health Goals understands that individual experiences can be similar but can also be entirely different, especially when we consider marginalized and minority communities. People who find themselves in such communities may not benefit from a general health improvement approach. Some important health-affecting factors include age, sex, gender identity, etc. Which is why Health Goals accommodates for individual identities and lifestyles by collecting pertinent information, if the user chooses to provide, that affect health. Users will be able to create an account in which they will be prompted to answer questions pertaining to their age, sex, gender identity, sexual orientation, race/ethnicity, weight loss goals, and personal workout goals. As stated before, huge aspect of improving one's lifestyle starts with diet, for that we implemented food recipes and food blogs/articles features. Basically, users will have the option of filtering through thousands of recipes based on their ingredients and dietary needs and restrictions.

Health Goals places a strong importance on a service that allows for a personalized experience for each user. We realize the significance of being updated with current mental and physical health concerns, latest fitness exercises, newest diets that been studied, etc. which leads us to integrating news articles as another feature of this website. We want our clients to always be informed through scholarly health and wellness articles that have been written, reviewed, and published by experts in the field. These articles that are catered specifically for each of the aspects we are focusing on: food, fitness, and mental health. In each of these features, we offer a way to personalize their experience by providing users with different filters. The user can filter through food articles by type of food (breakfast, lunch, dinner, kid friendly, etc) and by dietary restrictions (vegan, gluten-free, etc). Fitness exercise articles can be filtered by type of training (cardio, toning, yoga, etc) as well as type of difficulty ranging from one to five, five being the hardest. Mental health articles can be filtered by type of mental health (depression, anxiety and panic disorders, etc). Health Goals also includes a mental health forum feature so that users can share and contribute their experiences dealing with mental illness, sometimes people don't even realize importance of being able to talk another person about their struggles.

There can only be so much online support offered and many people would prefer to get help in person. Therapy can promote one's self-esteem, relationships,

and outlook on life. Health Goals will filter through therapists based on location and sex, and provide educational details as well so users can know that they are experienced therapists. However, good diet and mental health is not all that it takes to attain a healthy lifestyle, one needs to be physically fit as well. Physical health is the motivator that encourages the balance of nutrition, physical wellness, and mental well-being. We know that it's not always easy to be stay motivated and leading a healthy lifestyle requires dedication and time, which is why we believe that having a partner or a group will really help motivate people.

Health Goals already includes a Weightwatchers community which as we described before stresses how having a group of people who are interested in similar fitness routines and weight loss goals. This will not only encourage people to do their best but it also provides for a sense of community and that really helps our goal to personalize our website to cater to each and every person. But, sometimes people may not always feel comfortable venturing into different fitness routines or they similar want a change of location. Add to that there are always risks of a possible injuries that come with physical activities. The best way to avoid both scenarios is to go a reputable and knowledgeable fitness trainer who will improve the exercise execution and skills to not only reduce your risk for injury but also get the most out of each activity. Through the use of Fitness Instructor Finder, users can filter their choice of trainer based on training type, sex and location. We provide them with not only names and addresses but also the timings and prices of said instructor. Changing a lifestyle is revamping how person lives day to day and we know the stress that comes along with it. Health Goals strives to be able to do everything within our power to motivate and encourage our users in the best ways possible.

2. Glossary of Terms

Center Searches - This application filters out the nearby fitness centers and yoga centers by the users location in case they prefer to physically attend classes. Each center that shows up will have a link which will redirect to the center's web page which will allow for the user to see information about the hours, fees and programs available.

Diet - A special course of food curated to achieve a certain goal, such as weight loss

Exercise Circuit - The completion of a curated set of exercises used to train and condition the body

Fitness Routines - This section will consist of workout videos which will depend on which muscle group the user wishes to work on. The different workout videos will show users how to perform certain exercises and this setting will also recommend certain workouts based on what they mention they wish to work out.

Group - User will be able to create/join groups with their friends in which they will motivate and compete against each other

Leaderboard - A list of top users who accomplished their fitness goals within a group, ordered by goal percentage reached

Meal-plan - A recommended or personalized set of meals with the purpose of achieving a health goal. Also, the intended purpose is to save the user some time, but at the same time offer options to meals that fall under this health goal.

Mental Health Forum - Forums allow users to write about their experiences or ask any questions they have which can be replied to by any other user. This fosters a sense of community and allows the user to communicate about their struggles and problems. The focus of the forum will be mental health topics.

Profile - Allows the user to add additional details about themselves, goals, diet and health info, etc. to personalize their portfolio.

User - Anyone who wishes to access our personalized resources will be required to create an account. This user has access to all features.

Non-Registered User - Visitor to the site who can access most resources, though not personalized. This user cannot join groups, be part of the leaderboard, nor can they take part of the forums.

Weight Watchers Ranking Platform - Each user will be allowed to create a “community” or group consisting of their friends with whom they wish to compete, if they wish to compete. At the beginning of the week, the user will be asked their weight goal for the week. At the end of the week, the platform will allow users to see their weekly standing compared to their friends which will be determined by whether or not they achieved their weekly goal/ how far off they are. This ranking platform promotes healthy competition between users and their friends and helps them stay motivated towards their goal.

3. System Requirements

a. Enumerated Functional Requirements:

Identifier	Priority	Requirement
REQ-1	5	System should receive user input from searches
REQ-2	4	System should store and receive page data
REQ-3	5	System should store user data
REQ-4	4	System should provide pertinent articles/recipes/workouts/information based on profile information
REQ-4a	2	System should provide general articles/recipes/workouts/information based on filters for non-registered users
REQ-5	3	System should provide filtering options for all information resources (blogs, articles, recipes, matching services)
REQ-6	3	System should provide updated resources and information
REQ-7	3	System should recommend therapists based on user profile and filters
REQ-7a	2	System should recommend therapists based on filters for non-registered users
REQ-8	2	System should locate local professionals
REQ-9	2	System should suggest more resources based on previous searches for registered users
REQ-10	4	System should recommend personalized set of meals for user based on their health goals.
REQ-11	2	System should offer intermittent “challenges” and “trophies” for their completion
REQ-12	4	System should allow the registered user to create or comment on a thread under the Mental Health Forums

REQ-12a	2	System should allow the registered user to anonymously create or comment on a thread under the Mental Health Forums
REQ-12b	2	System should issue a trigger warning before entering the health forums
REQ-13	3	System should display a list of top users who accomplished their fitness goals within a group, ordered by goal percentage reached
REQ-13a	3	System should allow user to join groups and invite others to the group
REQ-*	3	System should recommend fitness/yoga instructors based on user profile and filters
REQ-*a	2	System should recommend fitness/yoga instructors based on filters for non-registered users
REQ-\$	4	System should allow a visitor to create an account
REQ-#	3	System should allow user to update their profile information

Table 3.1: Enumerated Functional Requirements

b. Enumerated Nonfunctional Requirements:

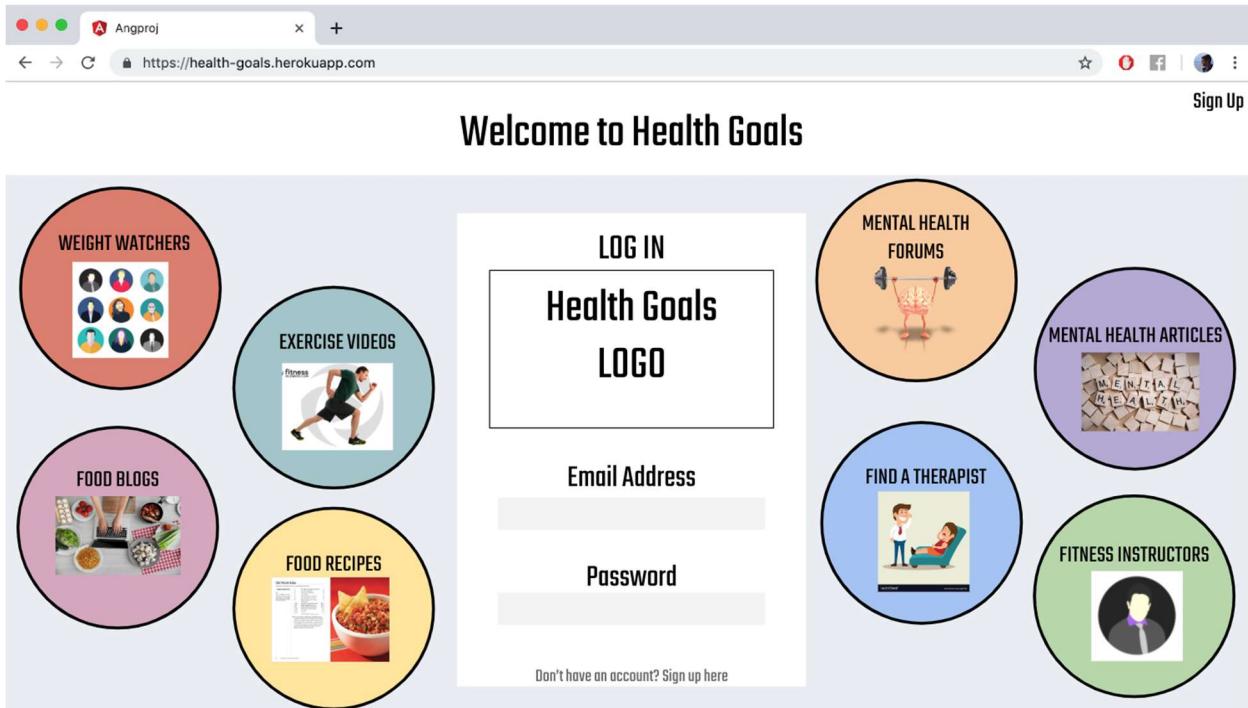
Identifier	Priority	Requirement
REQ-14	5	The site will require maintenance of at least one per week
REQ-15	2	The site will need updated articles. Every day there are many articles that are released and will prove another discovery wrong. We need to avoid this as much as possible
REQ-16	3	The forums must be regulated by staff in case of policy abuse
REQ-17	4	The site needs to have a strict security to protect people's personal data and information. Only admins may manage accounts, handle the server, and group settings. Moderators can only approve or disapprove posts, and manage group memberships.
REQ-18	3	The site needs to load articles from the database. This action will be implemented prioritizing time complexity (amount of

		time a program function takes to process a given input) so that articles will load efficiently.
REQ-19	4	The site needs to have weekly backups in order to restore in case of an error; Carbonite(online backup service) will be used to achieve this.
REQ-20	4	The site needs to be able to handle at least thousands of users and not crash if it gets close to the limit

Table 3.2: Enumerated Nonfunctional Requirements

c. On-Screen Appearance Requirements:

Welcome page - User login



Create Account (Registration)

Angproj

<https://health-goals.herokuapp.com>

Sign in

Health Goals Logo

Sign Up!



+ UPLOAD PHOTO

First Name	Last Name
Email Address	Password
Description	
Tell us about yourself...	
Health Goals	
Weight loss, get toned, healthy diet, etc...	
Dietary Restrictions	
Gluten free, vegetarian, vegan, etc...	

Food Recipes

Angproj

<https://health-goals.herokuapp.com>

Sign in

Health Goals Logo

Recipes

Filter Results

Meal Type

- Breakfast
- Lunch
- Dinner
- Snack
- Drink

Additional

- Convenience
- Cost
- Ratings

 [Custom Filters](#)



Image	Title
	Peaches Basic Nutrition: Cals, Servings, etc Diet Labels: Leon Health Labels: Fat-free, All-natural
	Fruit Salad Basic Nutrition: Cals, Servings, etc Diet Labels: Leon Health Labels: Fat-free, All-natural
	Bagel Avocado Sandwich Basic Nutrition: Cals, Servings, etc Diet Labels: Leon, High Protein Health Labels: Multigrain

[More Info](#)

Recommended | Page 1 of 10

Trending Recipes

1. Recipe Title
2. Recipe Title
3. Recipe Title
4. Recipe Title
5. Recipe Title
6. Recipe Title
7. Recipe Title
8. Recipe Title
9. Recipe Title
10. Recipe Title

Food Blogs / Relevant articles

Angproj

<https://health-goals.herokuapp.com>

Sign in

Health Goals Logo

Food blogs

Filter Results

Food

- Vegan
- Keto
- Kid friendly
- Breakfast

Restrictions

- Diary free
- Gluten free
- Vegetarian
- Low carb
- Non-GMO

Search for blogs and articles

Newest First

Page 1 of 10

- Website 1**
description
- Website 2**
description
- Website 3**
description
- Website 4**
description

Top 10 Articles

1. Article Title
2. Article Title
3. Article Title
4. Article Title
5. Article Title
6. Article Title
7. Article Title
8. Article Title
9. Article Title
10. Article Title

Angproj

<https://health-goals.herokuapp.com>

Sign in

Health Goals Logo

Find a Therapist

Mental Health Service Finder

Filter Results

Treatment Type

- Therapist
- Counselor
- Psychologist
- Other

Treatment Duration

- Short-Term
- Long-Term

Other Specifications

- LGBTQ Friendly

Location

Radius

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

Search



Name

Details

Address:

Time:

Price:



Name

Details

Address:

Time:

Price:



Name

Details

Address:

Time:

Price:

Mental Health Forums

The screenshot shows a web browser window for 'Angproj' at <https://health-goals.herokuapp.com>. The title bar says 'Health Goals Logo'. The top right has a 'Sign in' link. The main content area is titled 'Mental Health Forums: A safe discussion space for anyone who needs it'. It features a 'Filter Results' sidebar with 'Mental Health Topics' including Depression, Anxiety, PTSD, General, Men's Health, Women's Health, LGBTQ, and Suicide Prevention. The main content lists four forum topics: 'Coping with Anxiety and Depression', 'Should I See a Therapist?', 'No Motivation', and 'I Have a Story to Share', each with a short lorem ipsum description. A 'Start a Forum' button is in the top left of the main area. On the right, there's a 'Newest First' sorting option and a 'Page 1 of 10' indicator. A 'Active Forums' sidebar on the far right lists ten entries all labeled 'Title'.

Mental Health Articles / Information

The screenshot shows a web browser window for 'Angproj' at <https://health-goals.herokuapp.com>. The title bar says 'Health Goals Logo'. The top right has a 'Sign in' link. The main content area is titled 'Mental Health Resources'. It features a 'Filter Results' sidebar with 'Mental Health Topics' including Depression, Anxiety, PTSD, General, Men's Health, Women's Health, LGBTQ, and Suicide Prevention. The main content lists four website resources: 'Website 1' (description), 'Website 2' (description), 'Website 3' (description), and 'Website 4' (description). A 'Search for blogs and articles' bar is at the top. On the right, there's a 'Newest First' sorting option and a 'Page 1 of 10' indicator. A 'Top 10 Articles' sidebar on the far right lists ten entries numbered 1 through 10, all labeled 'Article Title'.

Weight Watchers

Angproj Sign in

<https://health-goals.herokuapp.com>

Health Goals Logo

Weight Watchers

Join Group

Name of group you want to join

Find Group

Filter Results to find best group

Training Type

<input type="checkbox"/> Cardio	<input type="checkbox"/> Toning
<input type="checkbox"/> Strength Training	<input type="checkbox"/> Yoga

Calorie Goal per week

<input type="checkbox"/> 1000-2000	<input type="checkbox"/> 3000-5000
<input type="checkbox"/> 2000-3000	<input type="checkbox"/> 5000+

Find

Group Name

Members	Top 10 Leaderboard	Sort: Rank

Angproj Sign in

<https://health-goals.herokuapp.com>

Health Goals Logo

Fitness Exercises

Exercise Videos

Filter Results

Training Type

<input type="checkbox"/> Cardio
<input type="checkbox"/> Toning
<input type="checkbox"/> Yoga
<input type="checkbox"/> Strength Training

Difficulty

<input type="checkbox"/> 0
<input type="checkbox"/> 1
<input type="checkbox"/> 2
<input type="checkbox"/> 3
<input type="checkbox"/> 4

Search

Newest First	Page 1 of 10		
UPPER BODY - 10 MIN Fun Upper Body Workout + A Message About Goal Setting	UPPER BODY, LOWER BODY, TOTAL BODY - 38 MIN Lower Body HIIT and Upper Body Strength; Total Body Workout Combo	LOWER BODY - 35 MIN 35 Minute Pilates Workout for Butt and Thighs - At Home Pilates Workout	TOTAL BODY - 84 MIN HIIT Cardio, Total Body Strength, Core Workout - 1000 Calorie Workout for 5 Million Subs!

Fitness Instructor Match up

Angproj

https://health-goals.herokuapp.com

Sign in

Health Goals Logo

Fitness Instructor Finder

Filter Results

Training Type

Cardio Toning
 Strength Training Yoga

Trainer

Male Female

Location

Radius

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

Search



Instructor Name

Details
Address:
Time:
Price:



Instructor Name

Details
Address:
Time:
Price:



Instructor Name

Details
Address:
Time:
Price:

4. Functional Requirements Specification

a. Stakeholders

Identify anyone and everyone who has interest in this system (users, managers, sponsors, etc.). Stakeholders should be humans or human organizations.

- Users
 - Fitness & Yoga Instructors
 - Mental Health Professionals
 - Healthcare Facilities

b. Actors and Goals

Identify the roles of people or devices that will directly interact with the system, their types (initiating vs. participating) and the goals of the initiating actors.

- Find Potential Clients

c. Use Cases

i. Casual Description

Use-Case Name	Actor	Actor's Goal	Functional Requirement
GetMealPlan (UC-1)	User	To receive a meal-plan based on profile and filters	REQ-3, REQ-10, REQ-5
MentalHealthMatch (UC-2)	User	To find a local mental health professional match based on profile and filters	REQ-3, REQ-7, REQ-8, REQ-5
FitnessYogaMatch (UC-3)	User	To find a local fitness/yoga instructor match based on profile and filters	REQ-3, REQ-*, REQ-8, REQ-5
StartForum (UC-4)	User	To start a mental health forum	REQ-1, REQ-2, REQ-12, REQ-12a
CommentForum (UC-5)	User	To comment on an existing mental health forums	REQ-1, REQ-2, REQ-12, REQ-12a
GetWorkout (UC-6)	User	To find personalized exercises/workouts based on profile and filters	REQ-3, REQ-4, REQ-5, REQ-9
GetRecipe (UC-7)	User	To find personalized recipe suggestions based on profile and filters	REQ-3, REQ-4, REQ-5, REQ-9

GetInfo (UC-8)	User	To find personalized and updated wellness articles and information	REQ-3, REQ-4, REQ-6, REQ-9
JoinGroup (UC-9)	User	To join a group and compete in the leaderboards	REQ-2, REQ-3, REQ- 13, REQ-13a
UpdateAccount (UC-18)	User	To update their profile	REQ-2, REQ-3, REQ-#
CreateAccount (UC - 10)	Visitor	To create an account to become a registered user	REQ-2, REQ-3, REQ-\$
SearchFitYoga (UC-11)	Visitor	To find a fitness/yoga instructor based on search & filters	REQ-1, REQ-*a, REQ-5
SearchMentalHealth Prof (UC-12)	Visitor	To find a mental health professional based on search & filters	REQ-1, REQ-7a, REQ-5
SearchWorkout (UC-13)	Visitor	To find general exercises/workouts based on search and filters	REQ-1, REQ-4a, REQ-5
SearchRecipe (UC-14)	Visitor	To search and find general recipes based on search & filters	REQ-1, REQ-4a, REQ-5
SearchInfo (UC-15)	Visitor	To find updated wellness articles and information based on search and filters	REQ-1, REQ-4a, REQ-6, REQ-5

Table 4.1: Use Case Casual Descriptions

ii. Use Case Diagram

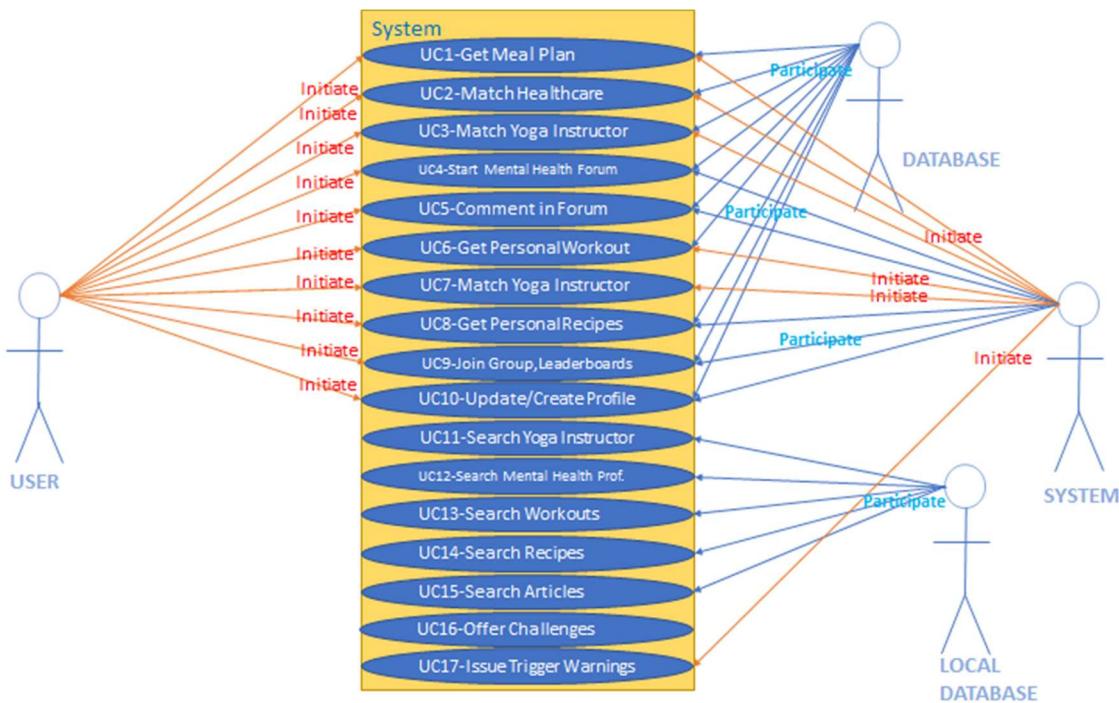


Figure 3.2: Use Case Diagram

iii. Traceability Matrix

	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15	UC16	UC17
REQ-1	5				X	X						X	X	X	X	X		
REQ-2	4				X	X				X	X							X
REQ-3	5	X	X	X			X	X	X	X	X							X
REQ-4	4						X	X	X									
REQ-4a	2													X	X	X		
REQ-5	3	X	X	X			X	X					X	X	X	X		
REQ-6	3								X									X
REQ-7	3		X															
REQ-7a	2																	
REQ-8	2		X	X														
REQ-9	2						X	X	X									
REQ-10	4	X																

REQ-11	2																	x	
REQ-12	4				x	x													
REQ-12a	3				x	x													
REQ-12b	2																		x
REQ-13	3									x									
REQ-13a	3									x									
REQ-*	3		x																
REQ-*a	2												x						
REQ-\$									x	x									
REQ-#																			

Table 3.3: Tracability Matrix

iv. Fully-Dressed Description

Use Case : UC-1 Related Requirements: Initiating Actor: Actor's Goal: Participating Actors:	Get Meal Plan REQ-3, REQ-10, REQ-5 User To receive a meal-plan based on profile and filters System, Database
Preconditions: Postconditions:	The website displays user's profile and their interests The website displays user's chosen section, in this case their meal plan
Failed End Condition:	Profile not available, please try again
Flow of Events for Main Success Scenario: -> 1. Users click on profile button <- 2. System displays all sections personalized to user's profile -> 3. Users choose meal plan section. <- 4. System displays list of recipes most relevant to user's profile	
Flow of Events for Extensions: 3(a). Users pick meal plan link that is unavailable or no longer relevant <- 1. System displays error message	

Table 4-1 UC-1

Use Case : UC-2	Mental Health Match
Related Requirements:	REQ-3, REQ-7, REQ-8, REQ-5
Initiating Actor:	User
Actor's Goal:	To find mental health places match based on profile
Participating Actors:	System, Database, Mental Health Professionals
Preconditions:	The website displays options of therapists from our database
Postconditions:	The website displays user's chosen therapist
Failed End Condition:	Therapist not available, please try again
Flow of Events for Main Success Scenario:	
-> 1. Users click on search button and type in therapy type and defines filters	
<- 2. System displays list of therapists most relevant to search	
-> 3. Users choose therapist.	
<- 4. System displays new therapist with information	
Flow of Events for Extensions:	
3(a). Users pick therapist link that is unavailable or no longer relevant	
<- 1. System displays error message	

Table 4-2 UC-2

Use Case UC-4:	Start Forum
Related Requirements:	REQ-1, REQ-2, REQ- 12, REQ-12a
Initiating Actor:	Users
Actor's Goal:	To start a mental health forum
Participating Actors:	System
Preconditions:	The website displays existing forums and option to start forum
Postconditions:	Users are able to write their own forum
Failed End Condition:	Unable to create forum, please try again
Flow of Events for Main Success Scenario:	
-> 1. Users click forum button.	
<- 2. System shows list of existing forums and option to create new forum.	
-> 3. Users choose to create their own forum	
<- 4. System displays new page to write own forum	
Flow of Events for Extensions:	
3(a). Users pick forum that already exists with similar title	
<- 1. System displays error message	

Table 4-3 UC-4

Use Case UC-6: Related Requirements: Initiating Actor: Actor's Goal: Participating Actors: Preconditions: Postconditions: Failed End Condition:	Get Workout REQ-3, REQ-4, REQ-5, REQ-9 Users To find personalized exercises/workouts based on profile and filters Database, System The website displays the search page The website displays the user's chosen workout Workout not found, please try again
<p>Flow of Events for Main Success Scenario:</p> <ul style="list-style-type: none"> -> 1. Users click on search button and type in workout keywords <- 2. System displays list of workouts most relevant to search -> 3. Users choose workout. <- 4. System displays new page with workout videos <p>Flow of Events for Extensions:</p> <p>3(a). Users pick workout link that is unavailable</p> <p><- 1. System displays error message</p>	

Table 4-4 UC-6

Use Case UC-9: Related Requirements: Initiating Actor: Actor's Goal: Participating Actors: Preconditions: Postconditions: Failed End Condition:	Join Group REQ-2, REQ-3, REQ-13, REQ-13a Users To join a group and compete in leaderboards Database, System The website displays join group page with group options Users are able to see their own group and participate Unable to find group please try again
<p>Flow of Events for Main Success Scenario:</p> <ul style="list-style-type: none"> -> 1. Users click Join Group button. <- 2. System shows list of available groups along with leaderboard of top users. -> 3. Users choose group or choose to create their own group <- 4. System adds user to group and displays new page <p>Flow of Events for Extensions:</p> <p>3(a). Users pick group that is unavailable or maximum capacity</p> <p><- 1. System displays error message</p>	

Table 4-5 UC-9

Use Case UC-10: Related Requirements: Initiating Actor: Actor's Goal: Participating Actors:	Create Account REQ-2, REQ-3, REQ- \$ Users, Visitors To create an account to become a registered user System
Preconditions: Postconditions:	The website displays personal information dialog boxes User fills in necessary information and creates an account
Failed End Condition:	Unable to find account, please try again
Flow of Events for Main Success Scenario: -> 1. Visitor clicks create account button. <- 2. System shows empty dialog boxes asking for visitor's personal information -> 3. User fills in the necessary information and creates an account <- 4. System creates a new account for the user.	
Flow of Events for Extensions: 3(a). User creates account that already exists <- 1. System displays error message	

Table 4-6 UC-10

Use Case UC-14: Related Requirements: Initiating Actor: Actor's Goal: Participating Actors:	Search Recipe REQ-1, REQ-4a, REQ- 5 Users, Visitors To find personalized recipes based on search & filters System
Preconditions:	The website displays list of existing recipes with filters on side
Postconditions:	Users are able to narrow their search by selecting filters
Failed End Condition:	Unable to find recipe, please try again

Flow of Events for Main Success Scenario:

- > 1. Users click recipe button.
- <- 2. System shows list of existing recipes and option to filter them based on needs
- > 3. Users select from the different option shown based on their search
- <- 4. System displays recipe of user's choice

Flow of Events for Extensions:

- 3(a). Users pick recipe that has been deleted
- <- 1. System displays error message

Table 4-7 UC-14

Use Case UC-15 :	Search Info
Related Requirements:	REQ-1, REQ-4a, REQ-6, REQ-5
Initiating Actor:	User, Visitor
Actor's Goal:	To find updated wellness articles and information based on search
Participating Actors:	Database, System
Preconditions:	The website displays existing articles and option to search
Postconditions:	Users are able to see other articles relating to their search
Failed End Condition:	Unable to find any relevant article, please try again

Flow of Events for Main Success Scenario:

- > 1. Users click on search button and type in article name or relevant keywords
- <- 2. System displays list of articles most relevant to search
- > 3. Users choose article.
- <- 4. System displays new article with information

Flow of Events for Extensions:

- 3(a). Users pick article that is unavailable, recently deleted or nonexistent
- <- 1. System displays error message

Table 4-8 UC-15

d. System Sequence Diagrams

UC-1

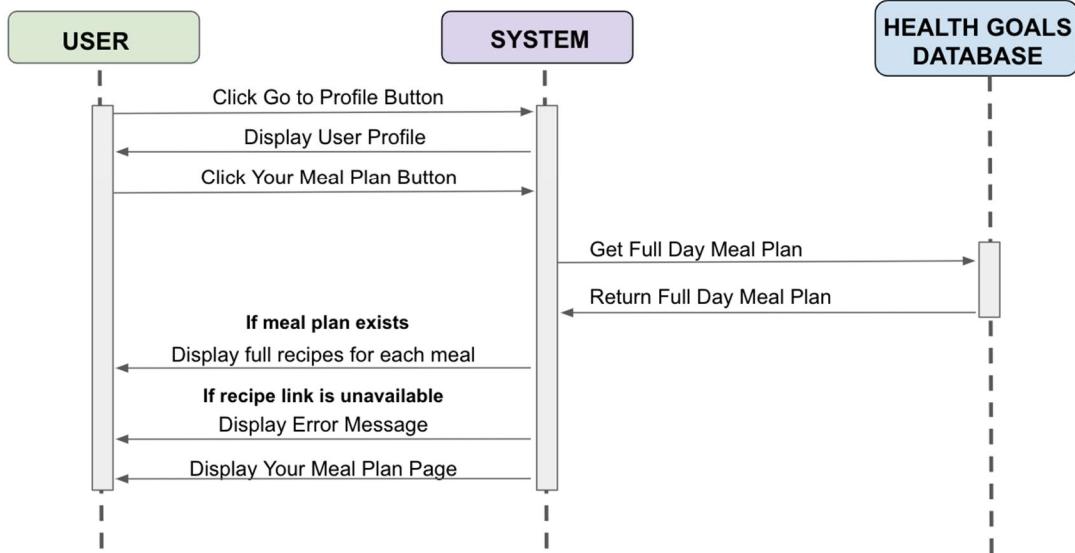


Figure 4-1 UC-1

UC-2

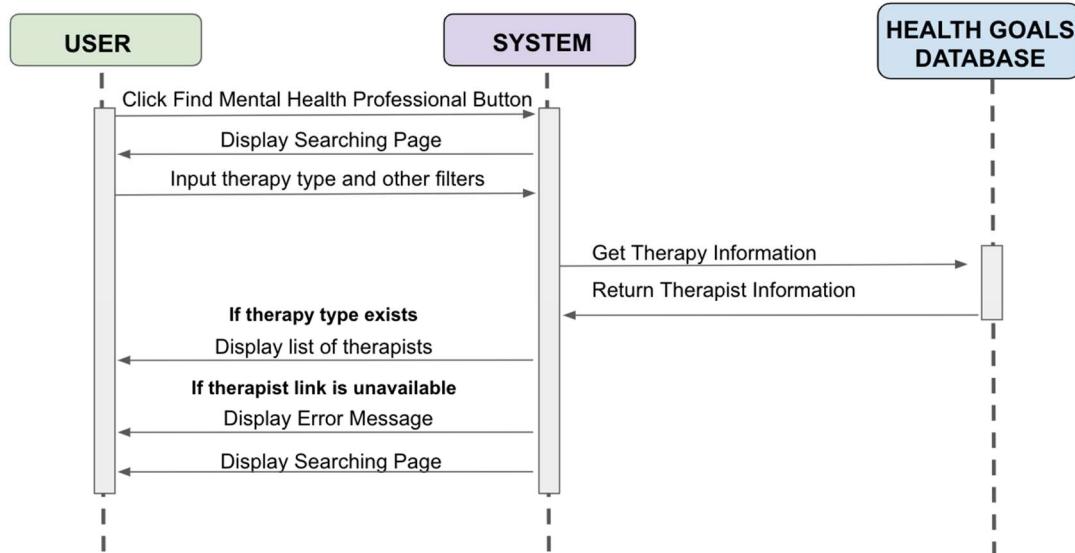


Figure 4-1 UC-2

UC-4

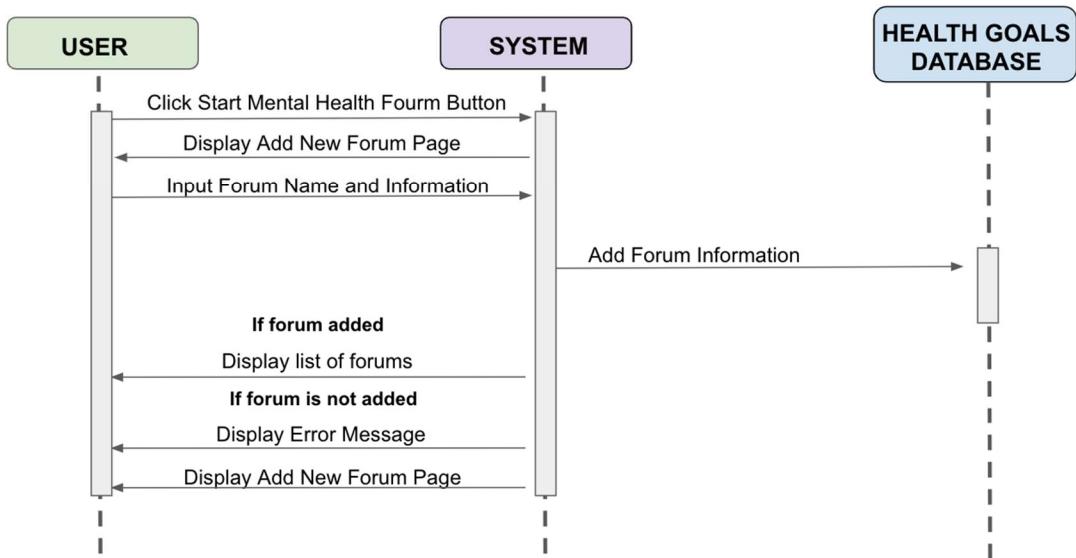


Figure 4-2 UC-4

UC-6

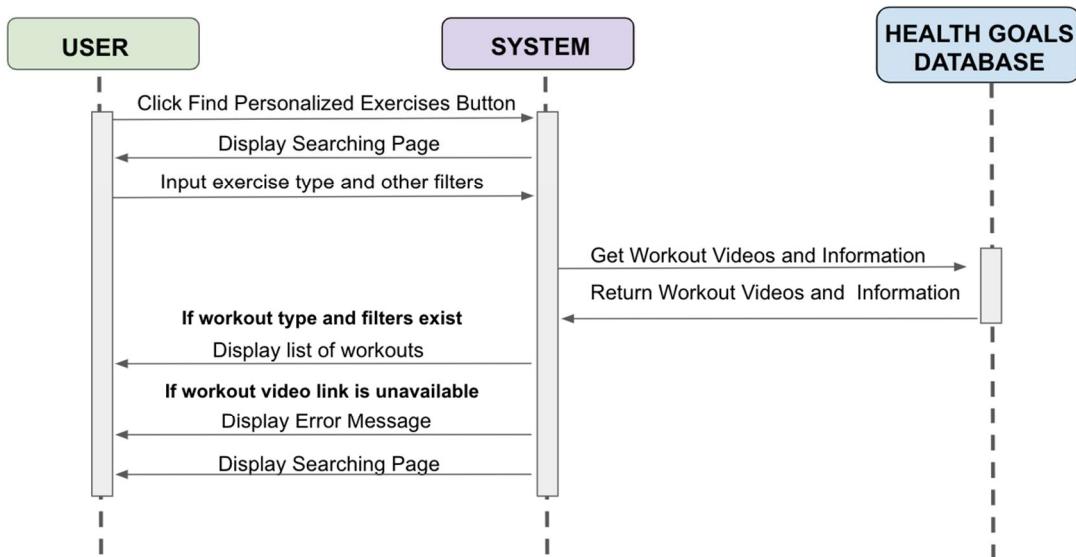


Figure 4-2 UC-6

UC-9

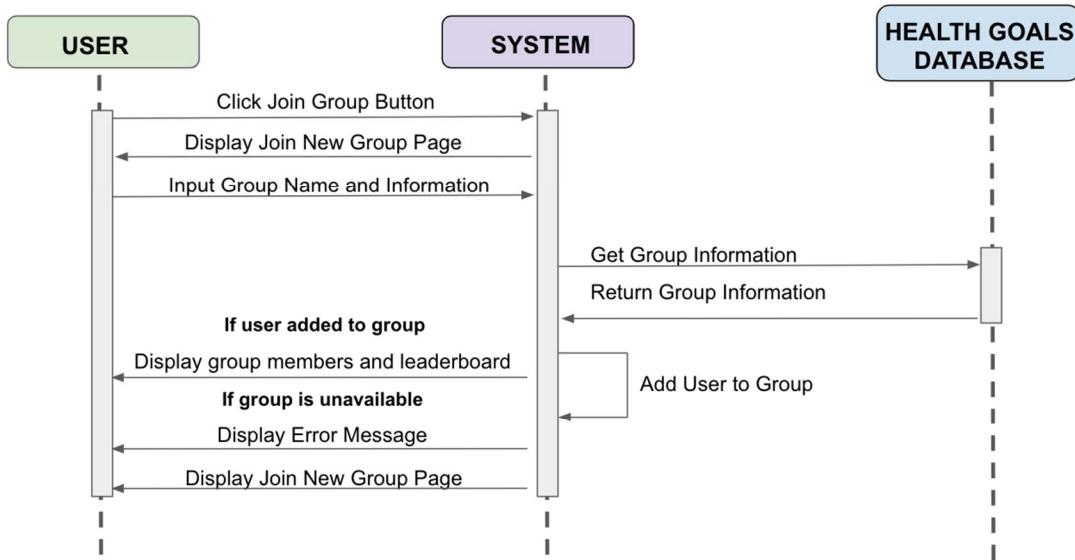


Figure 4-2 UC-9

UC-10

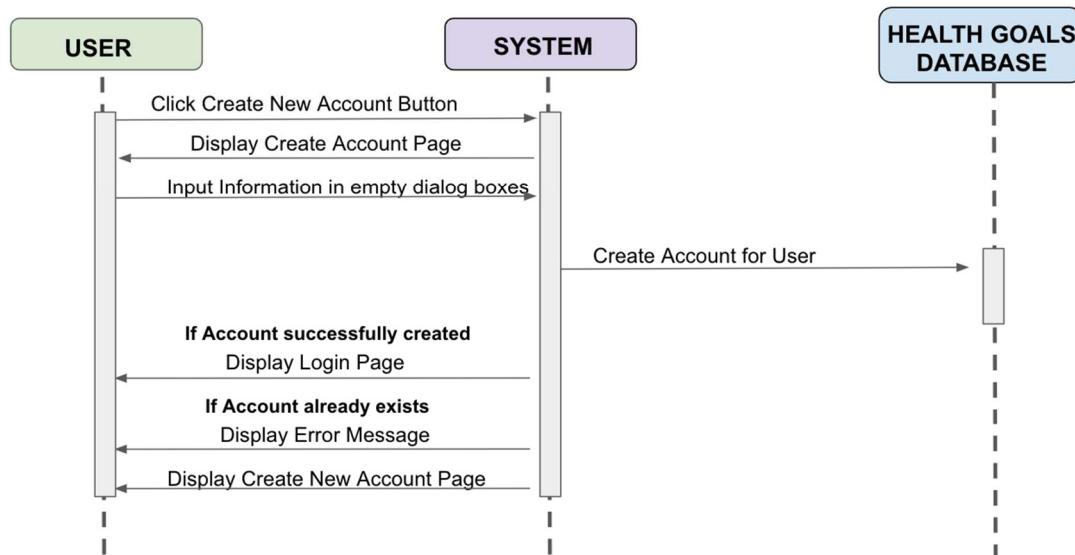


Figure 4-2 UC-10

UC-14

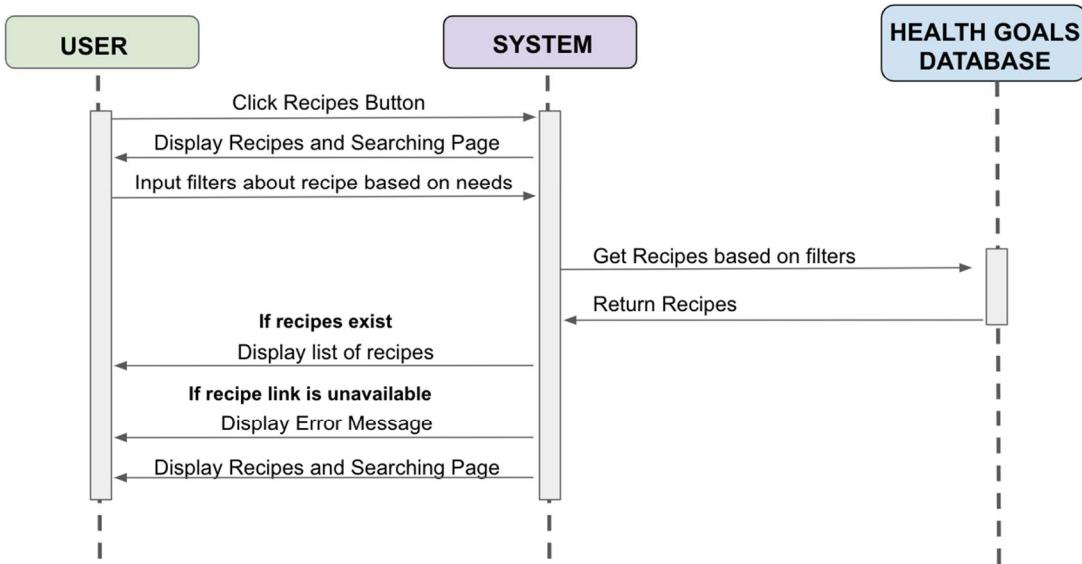


Figure 4-2 UC-14

UC-15

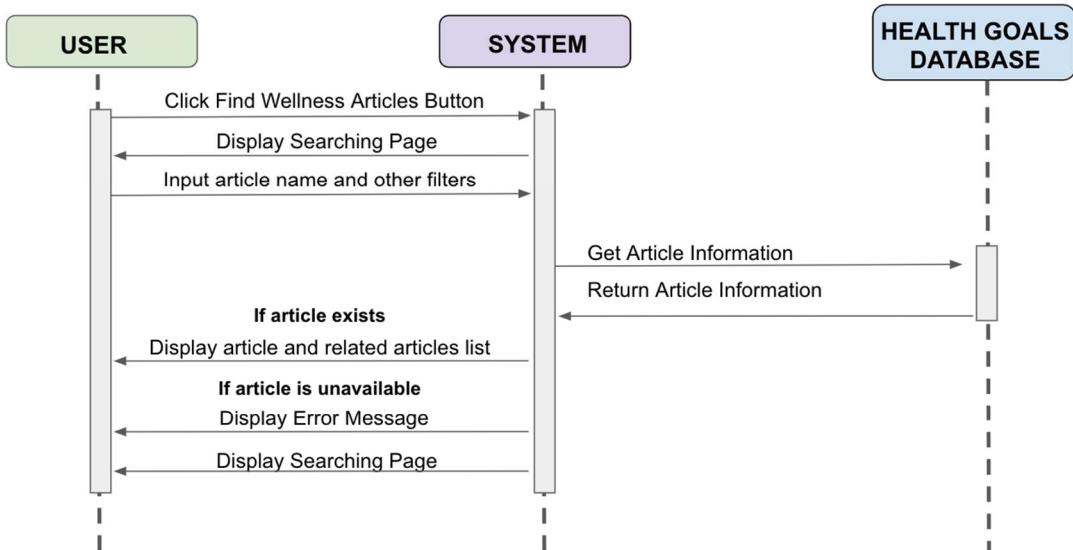


Figure 4-2 UC-15

5. Effort Estimation using Use Case Points

Use Case Points: (For UC 2, 4, 6, 9, 15)

To estimate the project size, we will calculate the use case points using this equation:

$$UCP = UUCP \times TCF \times ECF$$

(Unadjusted Use Case Points x Technical Complexity Factor x Environment Complexity Factor)

1. Unadjusted Use Case Points refers to the size of functional features involved in the project. We will find UUCP by calculating $UUCP = UUCW$ (Unadjusted Use Case Weight) + UAW (Unadjusted Actor Weight).

We will classify our use case weights as such:

[https://en.wikipedia.org/wiki/Use_Case_Points#Unadjusted_Use_Case_Weight_\(UUCW\)](https://en.wikipedia.org/wiki/Use_Case_Points#Unadjusted_Use_Case_Weight_(UUCW))

a. UUCW - Unadjusted Use Case Weight

Use Case Classification	No. of Transactions	Weight
Simple	1 to 3 transactions	5
Average	4 to 7 transactions	10
Complex	8 or more transactions	15

Table 5.1: Use Case Transaction Classification

Use Case	Description	Weight

UC-1	Get Meal Plan - Simple user interface - 4 steps for the main success scenario - 2 participating actors	10
UC-2	Mental Health Match - Average user interface - 4 steps for the main success scenario - 3 participating actors	10
UC-4	Start Mental Health Forum - Simple user interface - 4 steps for the main success scenario - 1 participating actor	10
UC-6	Get Workout - Simple user interface - 4 steps for the main success scenario - 2 participating actors	10
UC-9	Join Group - Simple user interface - 4 steps for the main success scenario - 2 participating actors	10

UC-10	Create Account - Simple user interface - 4 steps for the main success scenario - 1 participating actor	10
UC-14	Search Recipe - Simple user interface - 4 steps for the main success scenario - 1 participating actor	10
UC-15	Search Info - Simple user interface - 4 steps for the main success scenario - 2 participating actors	10

Table 5.2: Use Case Weight Classification

$$\text{UUCW} = (\text{Total No. of Simple Use Cases} \times 5) + (\text{Total No. Average Use Case} \times 10) + (\text{Total No. Complex Use Cases} \times 15) = 10(8)$$

Thus the UUCW is 80.

b. UAW - Unadjusted Actor Weight

Actor Classification	Type of Actor	Weight
Simple	The actor is another system which interacts with our system through a defined application programming interface (API).	1
Average	The actor is a person interacting through a text- or numeric-based user interface, or another system	2

	interacting through a protocol, such as a network communication protocol	
Complex	The actor is a person interacting via a graphical user interface (GUI).	3

Table 5.3: Actor Type Classification

Actor Name	Description of Characteristics	Classification	Weight
User	The user interacts with our system through a text-based user interface by creating an account, writing their own blogs, and searching through our data using keyboard and mouse	Complex	3
Database	The database interacts with our system by running and returning queries based on user's searches (text-based) when looking for therapists, exercise videos, etc.	Average	2
System	System displays results of the query returned by the database on the screen	Simple	1

Table 5.3: Actor Weight Classification

$$UAW = (\text{No. of Simple actors} \times 1) + (\text{No. Average actors} \times 2) + (\text{No. Complex actors} \times 3)$$

$$UAW = (1 \times 1) + (1 \times 2) + (1 \times 3) = 6$$

$$UUCP = UUCW + UAW = 80 + 6 = 86$$

The UAW is 6 and thus the UUCP is 86.

c. TCF – Technical Complexity Factors

Technical Complexity Factors are nonfunctional factors. They account for technical considerations, based on whether the features are irrelevant or not.

Factor	Description	Weight	Perceived Complexity	Factor
T1	Distributed system (able to run on multiple machines)	2.0	3	6
T2	Response time/performance objectives	1.0	3	3
T3	End-user efficiency	1.0	3	3
T4	Internal processing complexity	1.0	4	4
T5	Code reusability	1.0	3	3
T6	Easy to install	0.5	0	0
T7	Easy to use	0.5	5	2.5
T8	Portability to other platforms	2.0	1	2
T9	System maintenance (for existing or new features)	1.0	5	5
T10	Concurrent/parallel processing (multiple user requests and communication between users)	1.0	4	4
T11	Security features (for sensitive personal details)	1.0	3	3
T12	Access for third parties (None)	1.0	0	0
T13	End user training (None)	1.0	0	0
Technical Complexity Factor Total:				35.5

Table 5.4: Technical Complexity Factor

$$TCF = 0.6 + (TF / 100)$$

$$TCF = 0.6 + 35.5 / 100 = 0.955$$

d. Environment Complexity Factor

Environment Complexity Factor refers to the influence that different environmental factors may add to our project's evaluation. However we will assume that all of the members will work under the same environmental factors.

Thus we will take ECF = 1.

Deriving Project Size Estimation:

$$UCP = UUCP \times TCF \times ECF$$

$$UCP = 86 \times 0.955 \times 1$$

$$UCP \approx 82$$

Project Duration Estimation:

$$\text{Duration} = UCP * PF$$

PF=28 hrs (given in requirements)

$$\text{Duration} = 82 * 28 \approx 2296 \text{ hrs}$$

6. Domain Analysis

a. Domain Model - For UC 1, 2, 4, 6, 9, 10, 14, 15

UC-1: Get Meal Plan

Extracting Responsibilities (Concept Definitions)

Responsibility Description	Type	Concept Name
R1: Coordinate actions of concepts associated with this use case and delegate work to other concepts	D	Controller
R2: Form containing search parameters, obtained from profile information and selected filters (if selected)	K	Search AutoFill
R3: Contains user's search bar input and timestamp	K	Search Entry
R4: Keep track of user searches and their timestamp	D	Logger
R5: Prepare a database query that best matches the users search bar input and autofill information and retrieve the meal plan from the database	D	DB Connect

R6: Contains user's choice from presented options	K	User Choice
---	---	-------------

Table 6.1: Responsibility Concept Definitions

Extracting Associations

Concept Pair	Association Description	Association Name
Search Autofill - Controller	Controller receives autofill search parameters	Receive parameters
Search Entry - Controller	Controller receives the user's search input	Receive search
Search Entry - Logger	Logger receives the user's search input and stores it	Store search
Controller - DB Connect	Controller generates request from database	Request Data
Controller - DB Connect	Controller receives data from database	Receive Data
User Choice - Controller	Controller receives user's choice	Receive Choice

Table 6.2: Concept Associations

Extracting the Attributes

Concepts	Attributes	Attribute Description
Search Autofill	User's ID	Used to determine user's autofill search parameters
	Search Parameters	Existing health conditions, age, sex, etc. used as autofill search data
	User input	Used to form the database request

Search Entry	Timestamp	Used for logging search entries
User Choice	CurrentChoice	Used to display additional information on the chosen mental health professional

Table 6.3: Concept Attributes

Domain Model Sketch

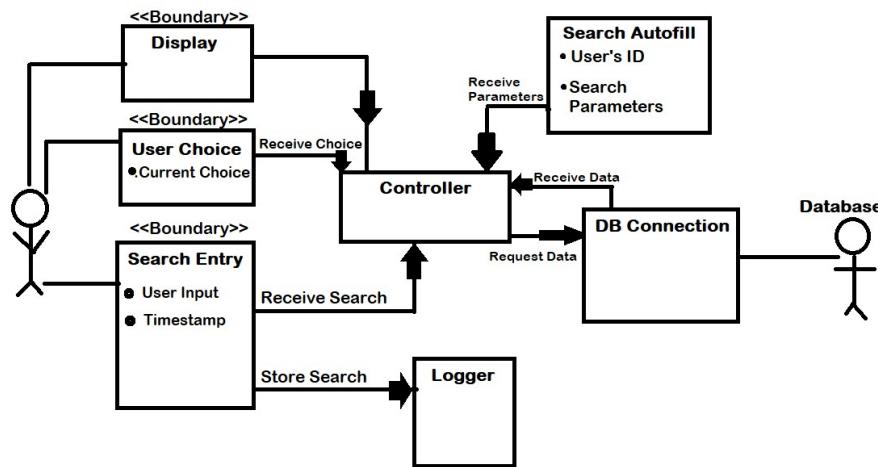


Figure 6.4: Domain Model

UC-2: Mental Health Match

Extracting Responsibilities (Concept Definitions)

Responsibility Description	Type	Concept Name
R1: Coordinate actions of concepts associated with this use case and delegate work to other concepts	D	Controller
R2: Form containing search parameters, obtained from profile information and selected filters (if selected)	K	Search AutoFill
R3: Contains user's search bar input and timestamp	K	Search Entry
R4: Keep track of user searches and their timestamp	D	Logger
R5: Prepare a database query that best matches the users	D	DB Connect

search bar input and autofill information and retrieve the therapists from the database		
R6: Contains user's choice from presented options	K	User Choice

Table 6.5: Responsibility Concept Definitions

Extracting Associations

Concept Pair	Association Description	Association Name
Search Autofill - Controller	Controller receives autofill search parameters	Receive parameters
Search Entry - Controller	Controller receives the user's search input	Receive search
Search Entry - Logger	Logger receives the user's search input and stores it	Store search
Controller - DB Connect	Controller generates request from database	Request Data
Controller - DB Connect	Controller receives data from database	Receive Data
User Choice - Controller	Controller receives user's choice	Receive Choice

Table 6.6: Concept Associations

Extracting the Attributes

Concepts	Attributes	Attribute Description
Search Autofill	User's ID	Used to determine user's autofill search parameters
	Search Parameters	Existing health conditions, age, sex, etc. used as autofill search data
Search Entry	User input	Used to form the database request
	Timestamp	Used for logging search entries
User Choice	CurrentChoice	Used to display additional information

		on the chosen mental health professional
--	--	--

Table 6.7: Concept Attributes

Domain Model Sketch

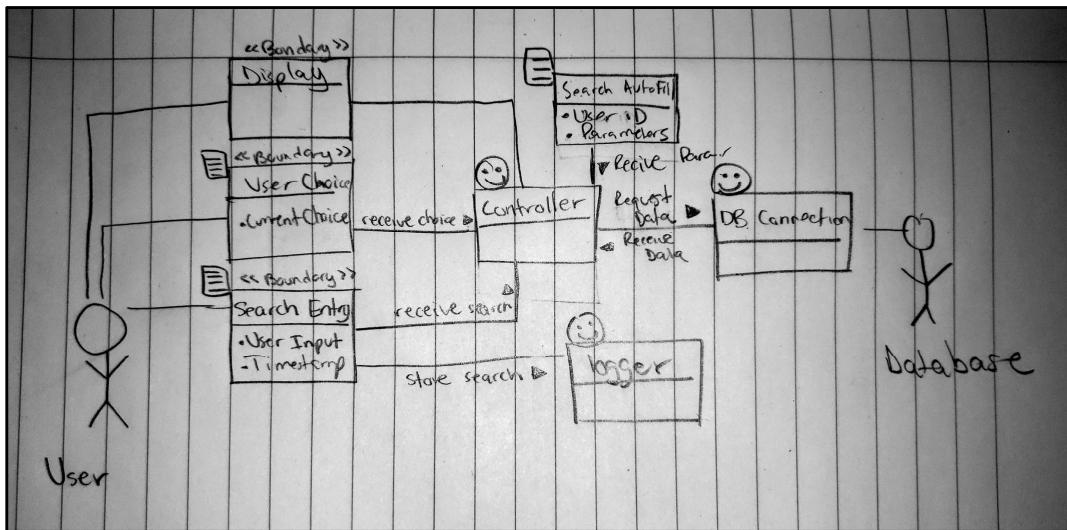


Figure 6.8: Domain Model

UC- 4: Start Forum

Extracting Responsibilities (Concept Definitions)

Responsibility Description	Type	Concept Name
R1: Coordinate actions of concepts associated with this use case and delegate work to other concepts	D	Controller
R2: Adds a new forum to the list of existing forums	D	DB Connection
R3: Contain the forum's information entered by the user, including title, description, and associated topics. Also contains system generated data	K	ForumInfo

R4: Ensures that a name is not being used in a different thread	D	NameChecker
---	---	-------------

Table 6.9: Responsibility Concept Definitions

Extracting Associations

Concept Pair	Association Description	Association Name
Controller - DB Connect	Controller generates request to store to database	Store Data
Controller - DB Connect	Controller generates request to retrieve data from database	Receive Data
Controller - ForumInfo	Controller receives forum information from user and systems	Receive Forum Info
Controller - NameChecker	Controller conveys a namecheck request and passes list of used names	Request Name check
Controller - NameChecker	Namechecker returns whether name is approved or denied	Check Outcome

Table 6.10: Concept Associations

Extracting the Attributes

Concepts	Attributes	Attribute Description
Forum Info	User's ID	Used to access created forums and to display name if attribute "visible name" is null
	Visible Name	Used for anonymity. If user does not want to display their real user id, they can create one for a given thread

	Forum Title	Title of the forum
	Forum Description	Detailed description of the forum
	Timestamp	Used for recordkeeping
	Associated Topics	Used for improved search abilities

Table 6.11: Concept Attributes

Domain Model Sketch

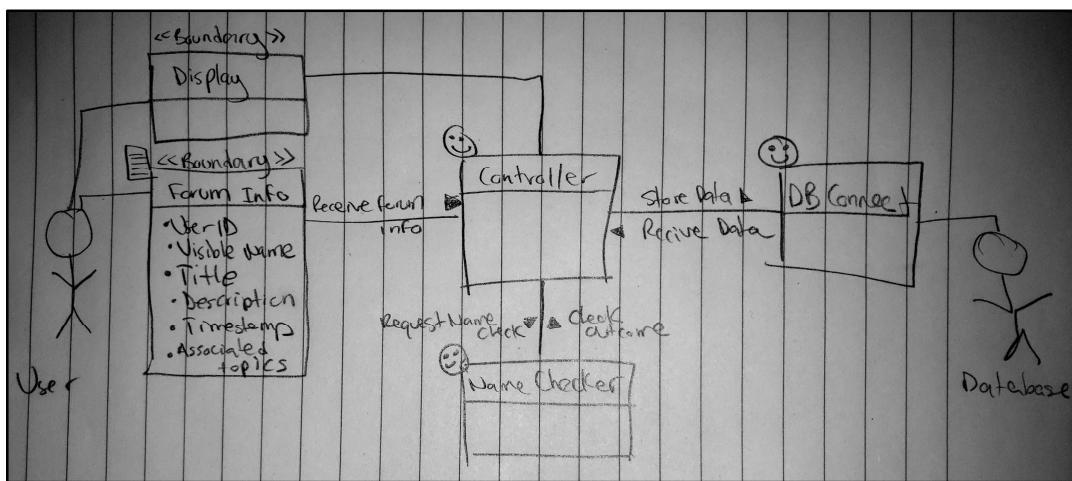


Figure 6.12: Domain Model

UC- 6: Get Workout

Extracting Responsibilities (Concept Definitions)

Responsibility Description	Type	Concept Name

R1: Coordinate actions of concepts associated with this use case and delegate work to other concepts	D	Controller
R2: Form containing search parameters, obtained from profile information and selected filters (if selected)	K	Search AutoFill
R3: Contains user's search bar input and timestamp	K	Search Entry
R4: Keep track of user searches and their timestamp	D	Logger
R5: Prepare a database query that best matches the users search bar input and autofill information and retrieve the workouts from the database	D	DB Connect
R6: Contains user's choice from presented options	K	User Choice

Table 6.13: Responsibility Concept Definitions

Extracting Associations

Concept Pair	Association Description	Association Name
Search Autofill - Controller	Controller receives autofill search parameters	Receive parameters
Search Entry - Controller	Controller receives the user's search input	Receive search
Search Entry - Logger	Logger receives the user's search input and stores it	Store search
Controller - DB Connect	Controller generates request from database	Request Data
Controller - DB Connect	Controller receives data from database	Receive Data
User Choice - Controller	Controller receives user's choice	Receive Choice

Table 6.14: Concept Associations

Extracting the Attributes

Concepts	Attributes	Attribute Description
Search Autofill	User's ID	Used to determine user's autofill search parameters
	Search Parameters	Existing health conditions, age, sex, etc. used as autofill search data
Search Entry	User input	Used to form the database request
	Timestamp	Used for logging search entries
User Choice	CurrentChoice	Used to display additional information on the chosen mental health professional

Table 6.15: Concept Attributes

Domain Model Sketch

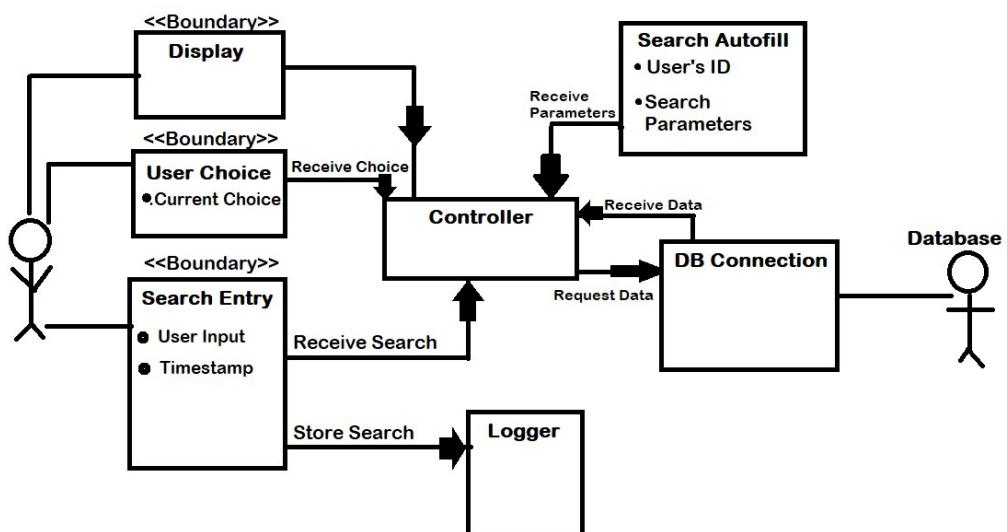


Figure 6.16: Domain Model

UC- 9: Join Group

Extracting Responsibilities (Concept Definitions)

Responsibility Description	Type	Concept Name
R1: Coordinate actions of concepts associated with this use case and delegate work to other concepts	D	Controller
R2: Adds a user to existing group	D	DB Connection
R3: Contain the group data including the assorted chats. Also contains system generated data	K	GroupInfo
R4: Ensures that a group exists to join	D	NameChecker

Table 6.17: Responsibility Concept Definitions

Extracting Associations

Concept Pair	Association Description	Association Name
Controller - DB Connect	Controller generates request to store to database	Store Data
Controller - DB Connect	Controller generates request to retrieve data from database	Receive Data
Controller - GroupInfo	Controller receives group information from user and systems	Receive Forum Info
Controller - NameChecker	Controller conveys a namecheck request and passes list of used names	Request Name check
Controller - NameChecker	Namechecker returns whether name is existent or not	Check Outcome

Table 6.18: Concept Associations

Extracting the Attributes

Concepts	Attributes	Attribute Description
Forum Info	User's ID	Used to access created forums and to display name if attribute “visible name” is null
	Visible Name	Used for anonymity. If user does not want to display their real user id, they can create one for a given thread
	Forum Title	Title of the group
	Forum Description	Rough description of the group
	Timestamp	Used for recordkeeping
	Associated Topics	Used for improved search abilities

Table 6.19: Concept Attributes

Domain Model Sketch

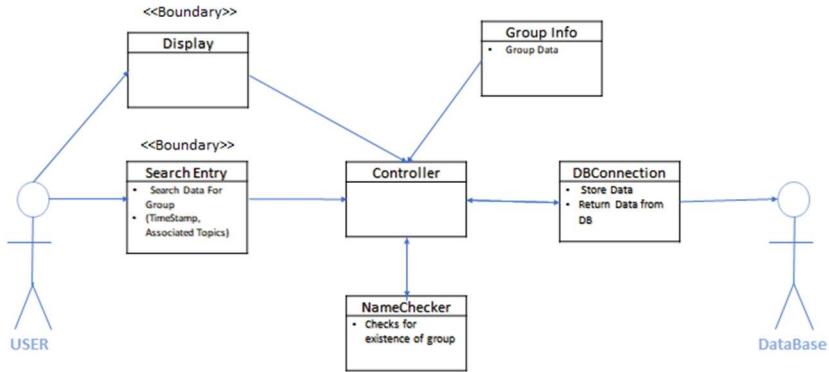


Figure 6.20: Domain Model

UC- 10: Create Account

Extracting Responsibilities (Concept Definitions)

Responsibility Description	Type	Concept Name
R1: Coordinate actions of concepts associated with this use case and delegate work to other concepts	D	Controller
R2: Adds a new account to the database and provides information about existing accounts	D	DB Connection
R3: Form containing the user's information (entered during the account creation process. Can be modified at later times)	K	ProfileInfo
R4: Ensures that the user id is not being used by a different user	D	IDChecker

Table 6.21: Responsibility Concept Definitions

Extracting Associations

Concept Pair	Association Description	Association Name
Controller - DB Connect	Controller generates request to store to database	Store Data
Controller - DB Connect	Controller generates request to retrieve data from database	Receive Data
Controller - ProfileInfo	Controller receives profile information from user	Receive Profile Info
Controller - IDChecker	Controller conveys a id check request and passes list of used user IDs	Request ID check
Controller - IDChecker	IDchecker returns whether id is approved or denied	Check Outcome

Table 6.22: Concept Associations

Extracting the Attributes

Concepts	Attributes	Attribute Description
Profile Info	User's ID	Used to identify users
	Name	First and Last Name
	Age	Use for tailoring information
	Gender	
	Sexual Orientation	
	Height & Weight	
	Existing Health Conditions (Physical)	

	Existing Health Conditions (Mental)	
	Dietary Restrictions	
	Fitness Goals	

Table 6.23: Concept Attributes

Domain Model Sketch

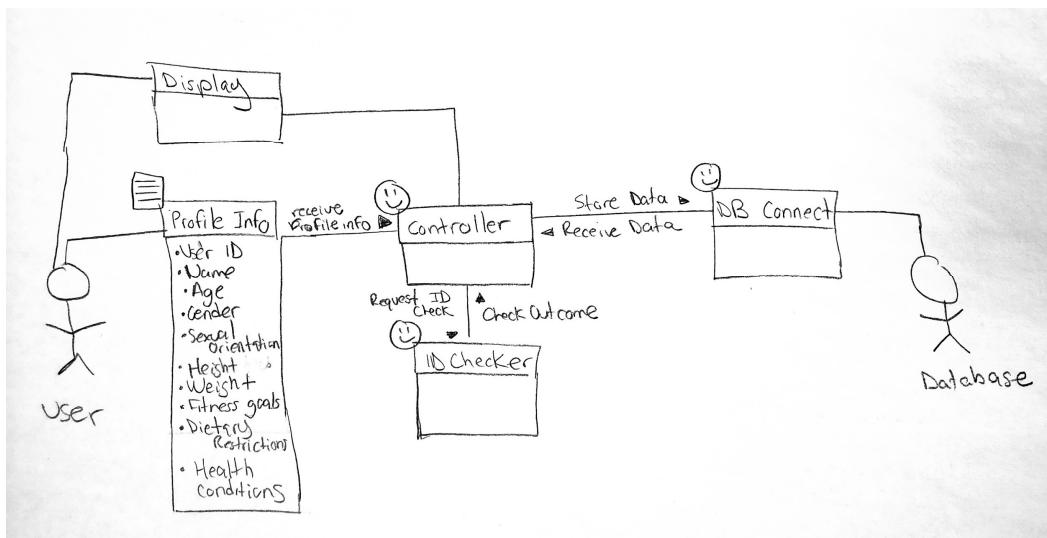


Figure 6.24: Domain Model

UC-14: Recipe Match

Extracting Responsibilities (Concept Definitions)

Responsibility Description	Type	Concept Name
R1: Coordinate actions of concepts associated with this use case and delegate work to other concepts	D	Controller

R2: Form containing search parameters, obtained from profile information and selected filters (if selected)	K	Search AutoFill
R3: Contains user's search bar input and timestamp	K	Search Entry
R4: Keep track of user searches and their timestamp	D	Logger
R5: Prepare a database query that best matches the users search bar input and autofill information and retrieve the recipes from the database	D	DB Connect
R6: Contains user's choice from presented options	K	User Choice
R7: Retrieve information from the user's profile to further aid searches	D	Profile Connect

Table 6.25: Responsibility Concept Definitions

Extracting Associations

Concept Pair	Association Description	Association Name
Logger - Controller	Controller receives logger data	Receive prior parameters
Search Entry - Controller	Controller receives the user's search input	Receive search
Search Entry - Logger	Logger receives the user's search input and stores it	Store search
Controller - DB Connect	Controller generates request from database	Request Data
Controller - DB Connect	Controller receives data from database	Receive Data
User Choice - Controller	Controller receives user's choice	Receive Choice

Table 6.26: Concept Associations

Extracting the Attributes

Concepts	Attributes	Attribute Description
Search Autofill	User's ID	Used to determine user's autofill search parameters by accessing profile
	Search Parameters	Calorie Count, ingredients, cost, and additional dietary requirements
Search Entry	User input	Used to form the database request
	Timestamp	Used for logging search entries
User Choice	CurrentChoice	Used to display additional information on the chosen recipes

Table 6.27: Concept Attributes

Domain Model Sketch

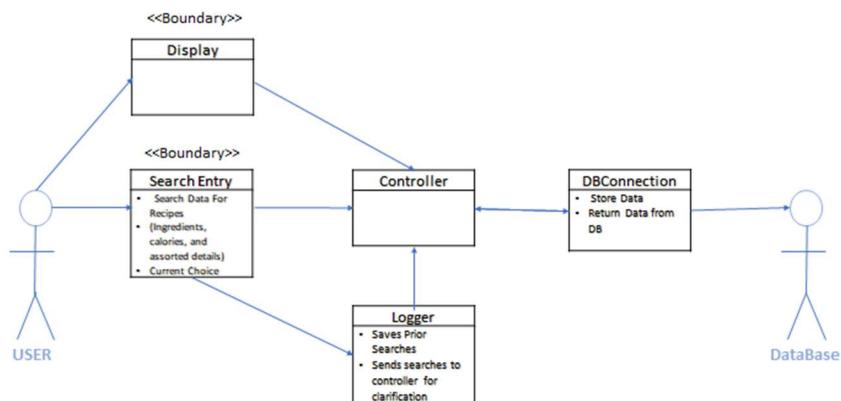


Figure 6.28: Domain Model

UC- 15: Search Info

Extracting Responsibilities (Concept Definitions)

Responsibility Description	Type	Concept Name
R1: Coordinate actions of concepts associated with this use case and delegate work to other concepts	D	Controller
R2: Form containing search parameters, obtained from profile information and selected filters (if selected)	K	Search AutoFill
R3: Contains user's search bar input and timestamp	K	Search Entry
R4: Keep track of user searches and their timestamp	D	Logger
R5: Prepare a database query that best matches the users search bar input and autofill information and retrieve the articles from the database	D	DB Connect
R6: Contains user's choice from presented options	K	User Choice

Table 6.29: Responsibility Concept Definitions

Extracting Associations

Concept Pair	Association Description	Association Name
Search Autofill - Controller	Controller receives autofill search parameters	Receive parameters
Search Entry - Controller	Controller receives the user's search input	Receive search
Search Entry - Logger	Logger receives the user's search input and stores it	Store search
Controller - DB Connect	Controller generates request from database	Request Data
Controller - DB Connect	Controller receives data from database	Receive Data
User Choice - Controller	Controller receives user's choice	Receive Choice

Table 6.30: Concept Associations

Extracting Attributes

Concepts	Attributes	Attribute Description
Search Autofill	User's ID	Used to determine user's autofill search parameters
	Search Parameters	Existing health conditions, age, sex, etc. used as autofill search data
Search Entry	User input	Used to form the database request
	Timestamp	Used for logging search entries
User Choice	CurrentChoice	Used to display additional information on the chosen mental health professional

Table 6.31: Concept Attributes

Domain Model Sketch

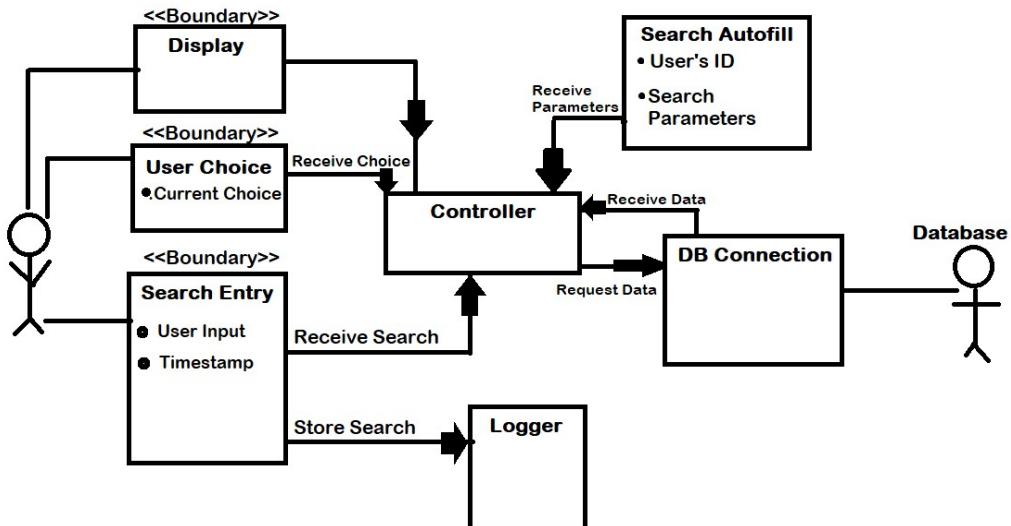


Figure 6.32: Domain Model

Traceability Matrix

Domain Model	UC-1	UC-2	UC-4	UC-6	UC-9	UC-10	UC-14	UC-15
Controller	X	X	X	X	X	X	X	X
Search Autofill	X	X		X				X
Search Entry	X	X		X			X	X
Logger	X	X		X			X	X
DB Connect	X	X	X	X	X	X	X	X
User Choice	X	X		X			X	X
ForumInfo			X		X			
NameChecker			X		X			
ProfileInfo						X		
IDChecker							X	

Table 6.33: Traceability Matrix

b. System Operation Contracts

Get Meal Plan - UC 1

Pre Condition	The website displays user's profile and their interests
Post Condition	The website displays user's chosen section, in this case their meal plan

Mental Health Match - UC 2

Pre Condition	The website displays options of therapists from our database
Post Condition	The website displays user's chosen therapist

Start Forum - UC 4

Pre Condition	The website displays existing forums and option to start forum
Post Condition	Users are able to write their own forum

Get Workout- UC 6

Pre Condition	The website displays the Search page
Post Condition	The website displays the user's chosen workout

Join Group - UC 9

Pre Condition	The website displays join group page with group options
Post Condition	Users are able to see their own group and participate

Create Account - UC 10

Pre Condition	The website displays personal information dialog boxes
Post Condition	User fills in necessary information and creates an account

Search Recipe - UC 14

Pre Condition	The website displays list of existing recipes with filters on side
Post Condition	Users are able to narrow their search by selecting filters

Search Info - UC 15

Pre Condition	The website displays existing articles and option to search
Post Condition	Users are able to see other articles relating to their search

7. Interaction Diagrams

UC - 1: GetMealPlan Sequence Diagram

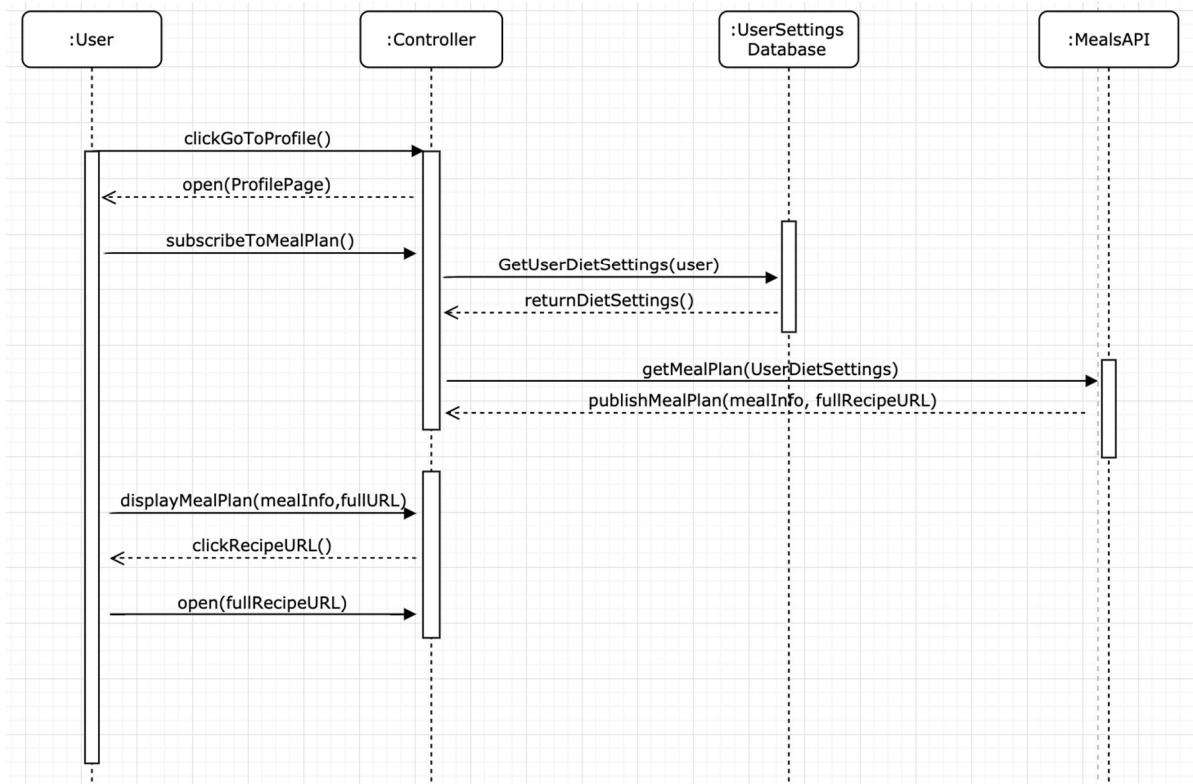


Diagram 7.1

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. This is a Subscriber–Publisher design pattern as the user is subscribed to the meal plan API.

Any changes in the API will not affect the subscriber, and vice versa. 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. 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: FindMentalHealthTherapists Sequence Diagram

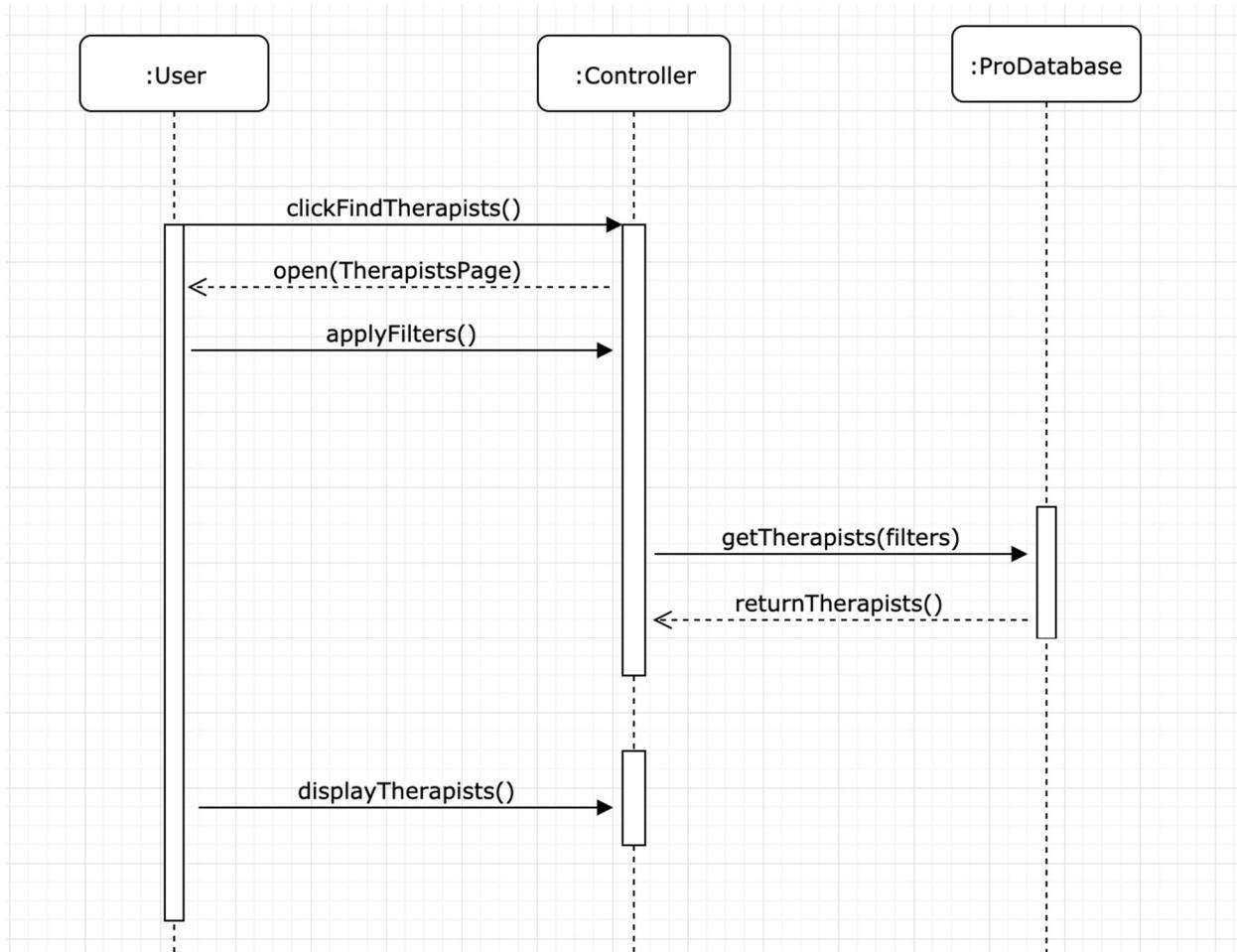


Diagram 7.2

This uses a Subscriber-Publisher design pattern as the user (subscriber) gets data from the Therapists database (publisher). The user first clicks on the FindTherapists icon, which leads them to the therapists page. There, they can apply filters bases on their personal preferences by checking them off. Once the click the apply button, these

filters are passed to our server which accesses the database to fetch the appropriate results. Therefore, the database and front end of our application are not directly connected. We would not have to alter the subscriber side if there are any changes in the publisher side.

UC - 4: StartForum Sequence Diagram

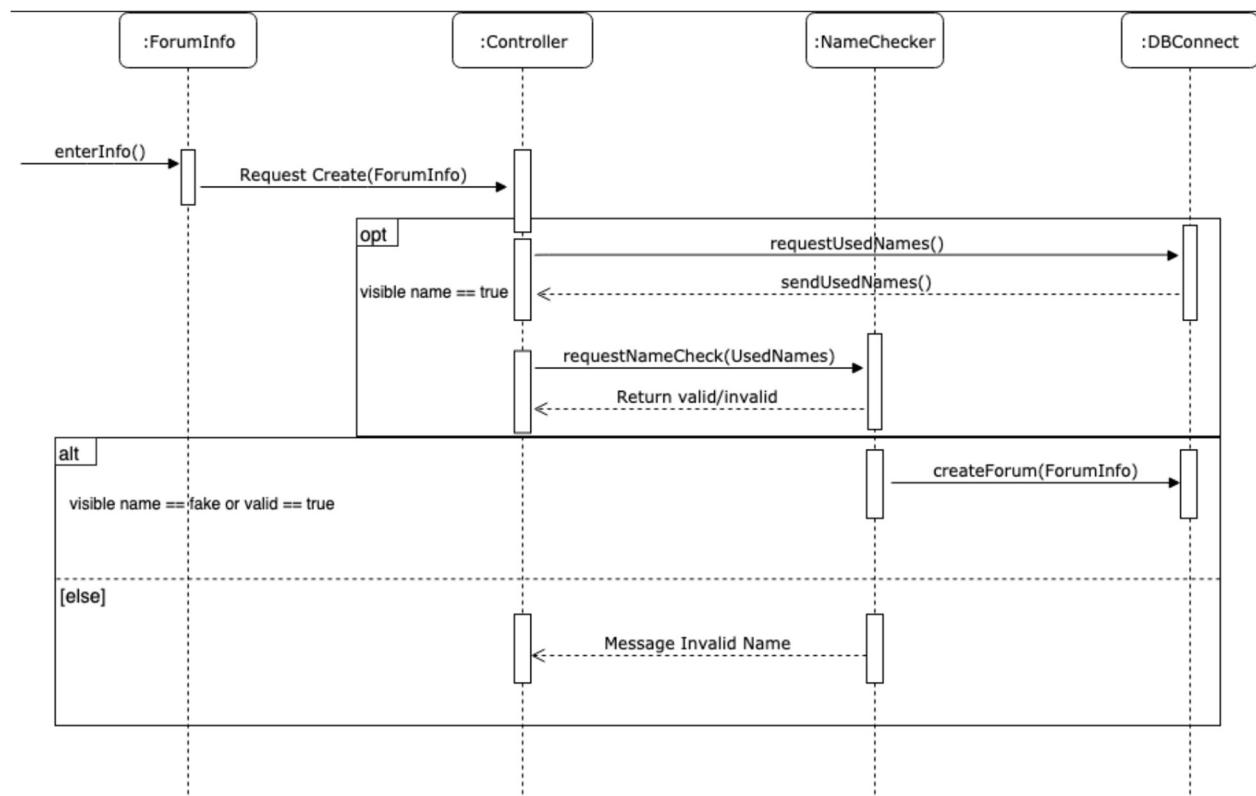


Diagram 7.3

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

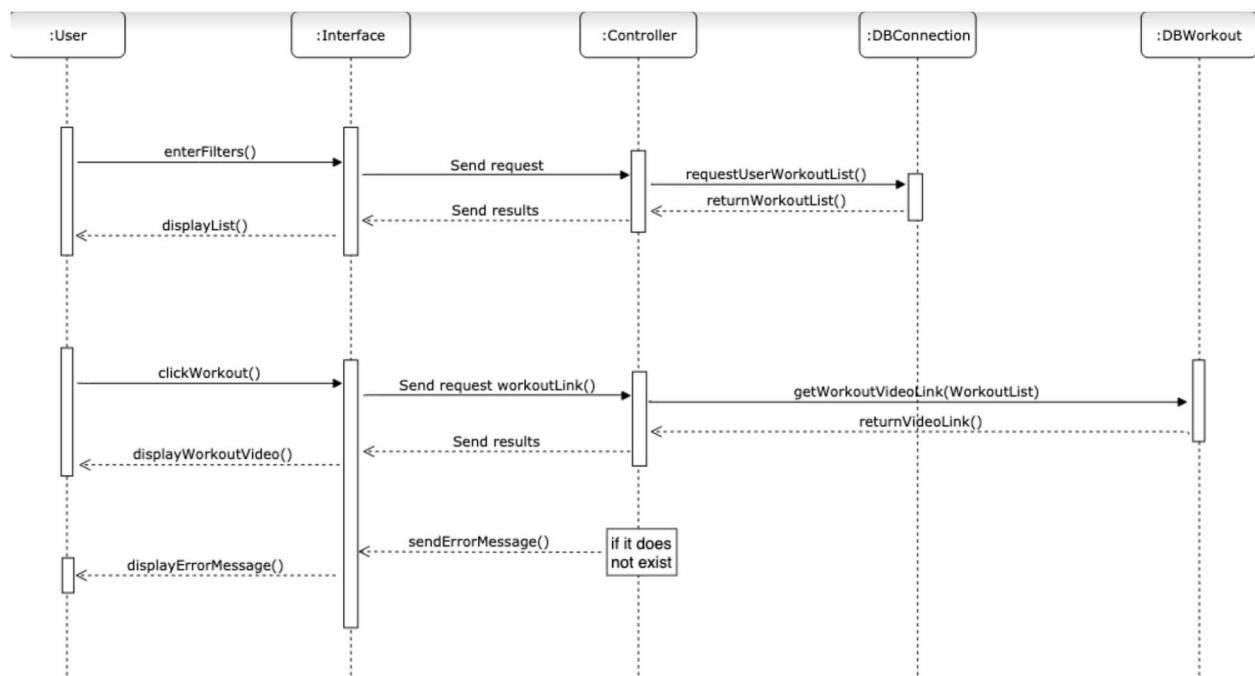


Diagram 7.4

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. This is another example of Subscriber-Publisher design pattern as the user is subscribes to the workout videos. The database and user indirectly communicate with each other.

UC - 9: JoinGroup Sequence Diagram

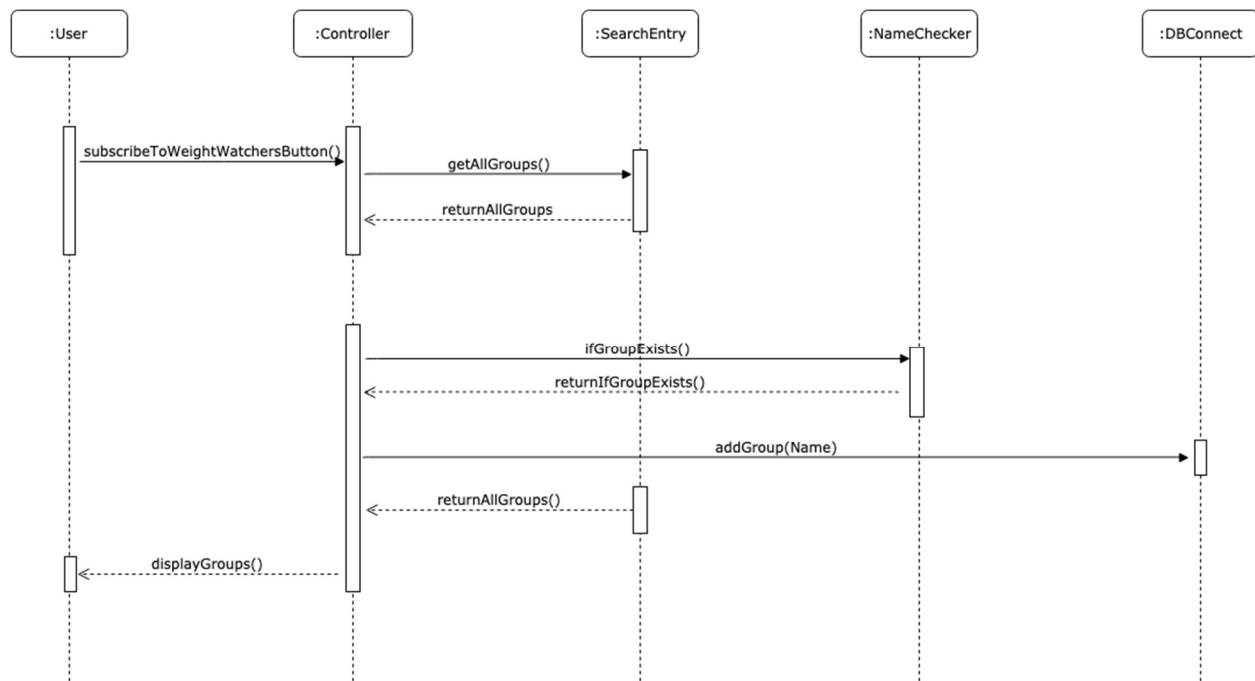


Diagram 7.5

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.

We see that this diagram displays the publisher-subscriber pattern. We wanted to use a well known and used software design pattern because design patterns are very helpful because it is best in adapting to changes, such as new customer requests. The publisher-subscriber pattern is helpful in simplifying the logic of a certain diagram and creates independence between the objects interacting. That way when the publisher needs to be somehow updated the subscriber's functions are intact. Thus we can use this pattern in the JoinGroup function between we want the user to be able to interact with the different elements on their group page. But if we want to update the different functions involved, we should be able to update the publisher without having to link all the subscribers involved. So essentially our controller is the publisher that will interact with events SearchEntry, NameChecker and DBConnect, but all subscribers (users) and the publisher will have no knowledge of each other.

UC - 10: CreateAccount Sequence Diagram

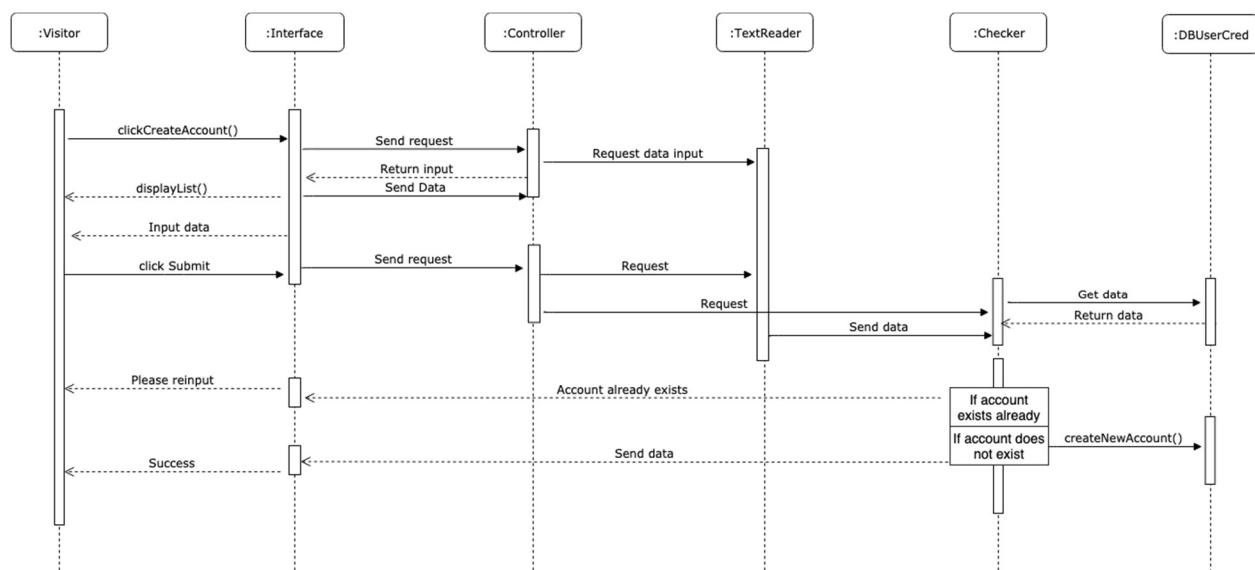


Diagram 7.6

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. We modified this diagram to display the Subscriber-Publisher design pattern because when creating accounts, there may be many elements that need to be changed, but we do not want that to affect the subscribers. Thus the Interface and Controller act as publishers while the visitors are subscribers that interact with the TextReader, Checker and DBUserCred events.

UC - 14: RecipeMatch Sequence Diagram

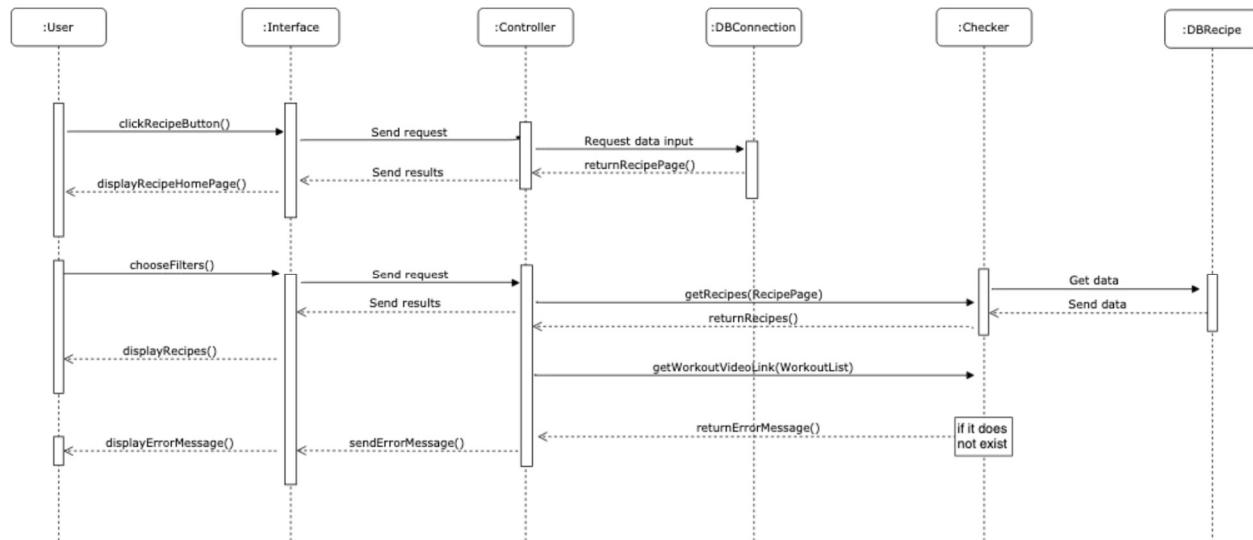


Diagram 7.7

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 send 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.

This use case and interaction diagram demonstrate the use of the following design patterns: publisher-subscriber, command, and decorator. The publisher-subscriber pattern is definitely the most prevalent. The user has access to an input messaging channel which in this case is the search bar and filters used to filter the search results. Based on the message from the input channel, relevant results are returns back to the user asynchronously. There is one output messaging channel for each of the subscribers or consumers depending on this data. When the user triggers an event like checking a filter and hitting enter to return a search query, the results are fetched for each of the subscribed methods from an http service that fetches information from an API (in this case Edamam's). The message from the output channel is parsed and organized accordingly to be displayed on the UI in a sensible way. This publisher subscriber pattern improves reliability because the asynchronous messaging helps the site run smoothly during increased load times. This also means the site and design is scalable and more responsive to the sender. In addition, the command pattern is used by limiting the user with access to the server. In this case, the filters and search bar have other business logic that allow the user to run a limited amount of queries to the Edamam's recipes database. The filters can even be undone by unchecking the checkboxes associated with the filters. Lastly, the decorator pattern is used by separating the main essential function from those that are non-essential. The service that fetches information regarding recipes in the main function in this case. A non-essential function is the auto-suggestions of words and ingredients as the user is typing in the search bar. This pattern makes it easier to add non-essential functions.

UC - 15: SearchInfo Sequence Diagram

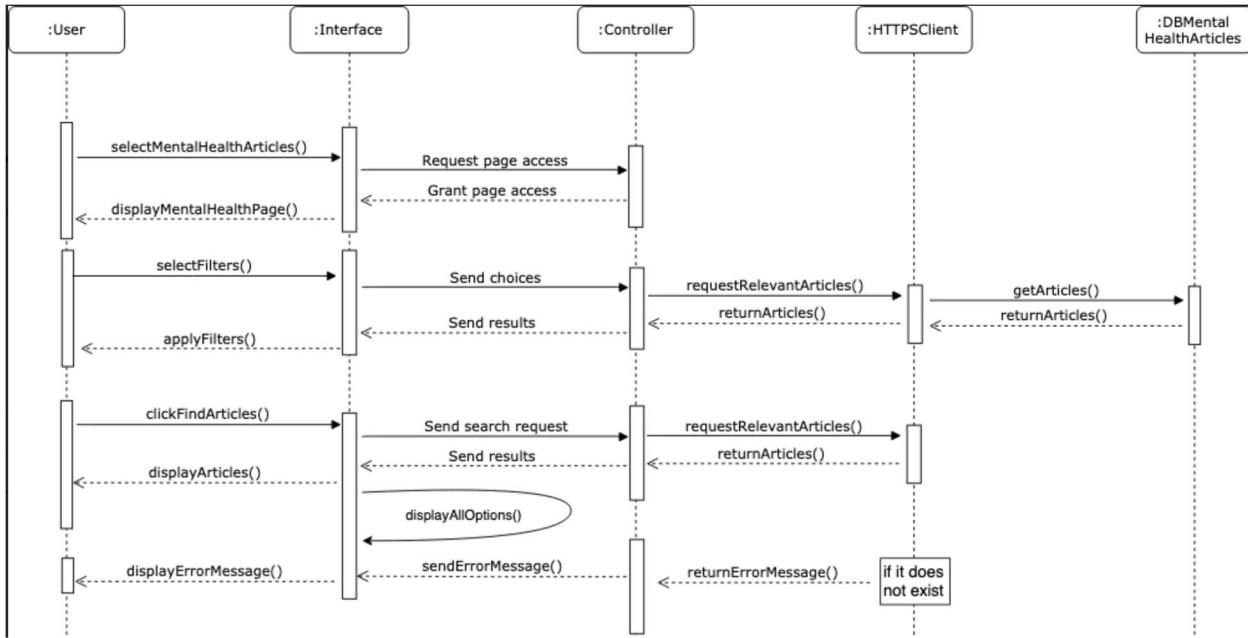


Diagram 7.8

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.

This use case and interaction diagram demonstrate the use of the publisher-subscriber pattern. The user has access to an input messaging channel which in this case is the search bar and filters used to filter the search results. Based on the message from the input channel, relevant results are returned back to the user asynchronously from a database of mental health articles and forums. There is one output messaging channel for each of the users waiting on this data. When the user triggers an event like checking a filter or hitting search, the results are fetched for each of the subscribed methods from an http service that fetches information from a MySQL database through a Python REST API. The message from the output channel is shown with a list of forums or articles depending on the mental health page. This publisher subscriber pattern improves reliability because the asynchronous messaging helps the site run

smoothly during increased load times. This also means the site and design is scalable and more responsive to the sender.

8. Class Diagram and Interface Specification

a. Class Diagram

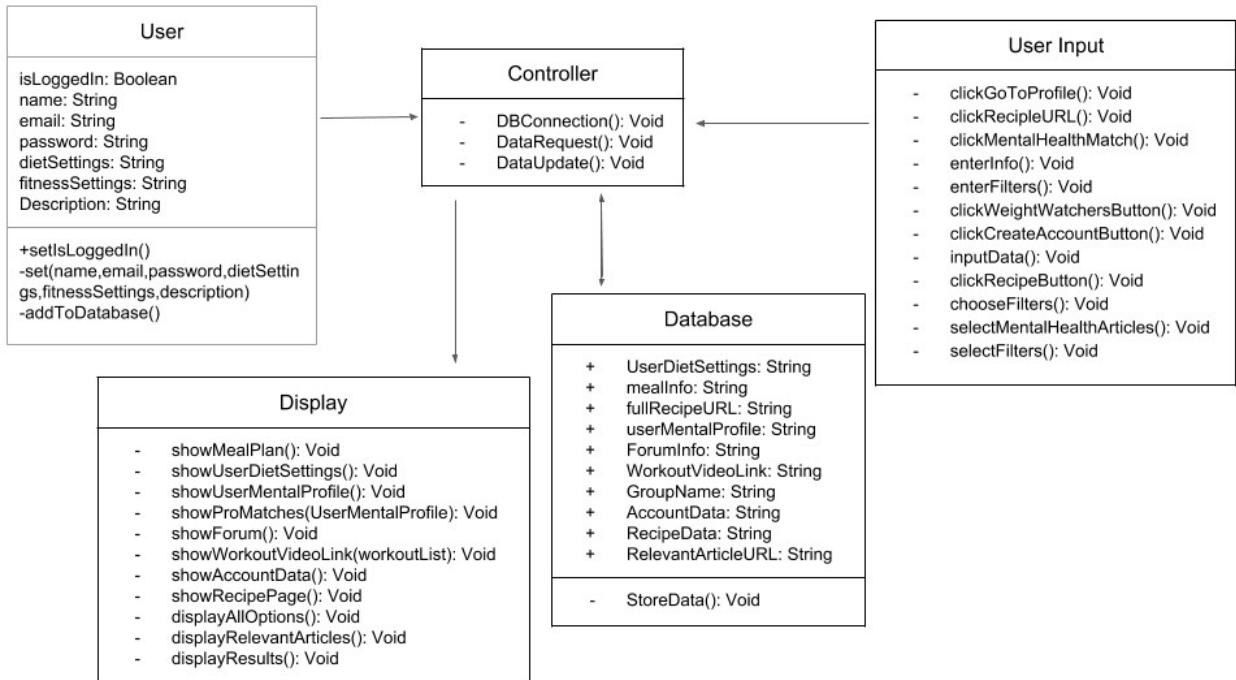


Diagram 2.a

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 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

Classes DomainConcept	Display	User	User Input	Controller	Database
Controller				✓	
Search Autofill				✓	✓
Search Entry	✓		✓	✓	
Logger	✓	✓		✓	
DB Connect		✓		✓	
User Choice	✓			✓	
Forum Info	✓			✓	✓
Name Checker			✓	✓	✓

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
- 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
- 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
- Controller:
 - Search Autofill - Controller receives autofill 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
- 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

Use of design patterns to improve our design is described in the previous section. In terms of **Object Constraint Language Contracts**, these are our invariants, preconditions, postconditions for classes and their operations:

d. Object Constraint Language(OCL) Contracts

1. Display

Invariants	Login credentials can be entered by the user; Not required
Precondition	Information has to be retrieved to display for every operation
Post Condition	Shows webpage to our user

Table 2.1.d

2. User

Invariants	User login credentials
Precondition	User has to be logged in
Post Condition	Shows user specific information on page

Table 2.2.d

3. User Input

Invariants	User operations or text entered
Precondition	User clicks on desired operation
Post Condition	Return user desired function or value

Table 2.3.d

4. Controller

Invariants	Retrieves data from database
Precondition	A request made by different operations in different classes
Post Condition	Returns a result or request made by operation or class

Table 2.4.d

5. Database

Invariants	User or visitor specific request to make changes
Precondition	Access to the database backend
Post Condition	Updates or creates new data in the backend & updates website

Table 2.5.d

9. 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

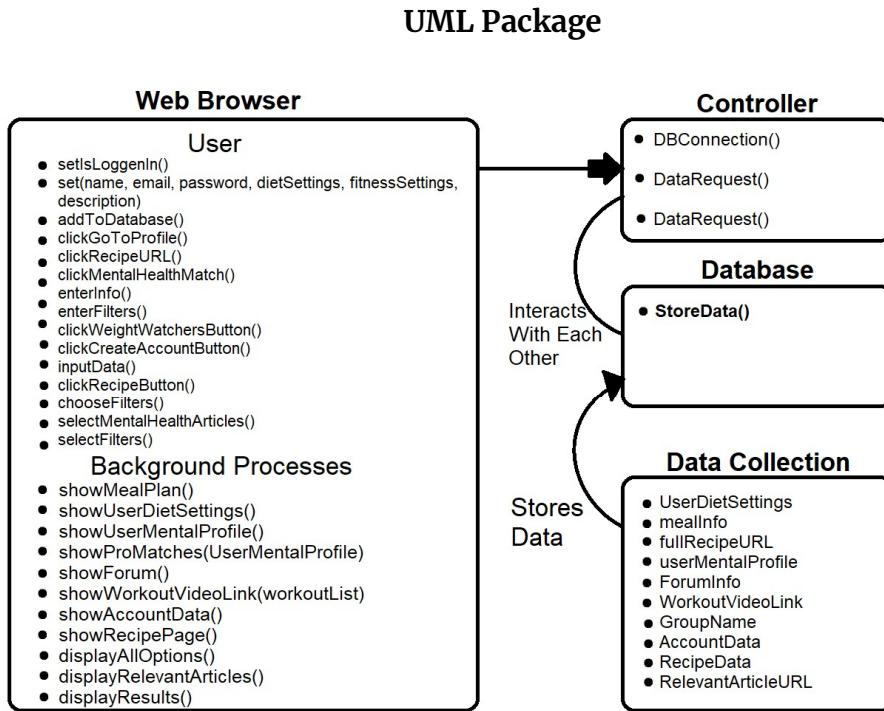


Diagram 3.b

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.

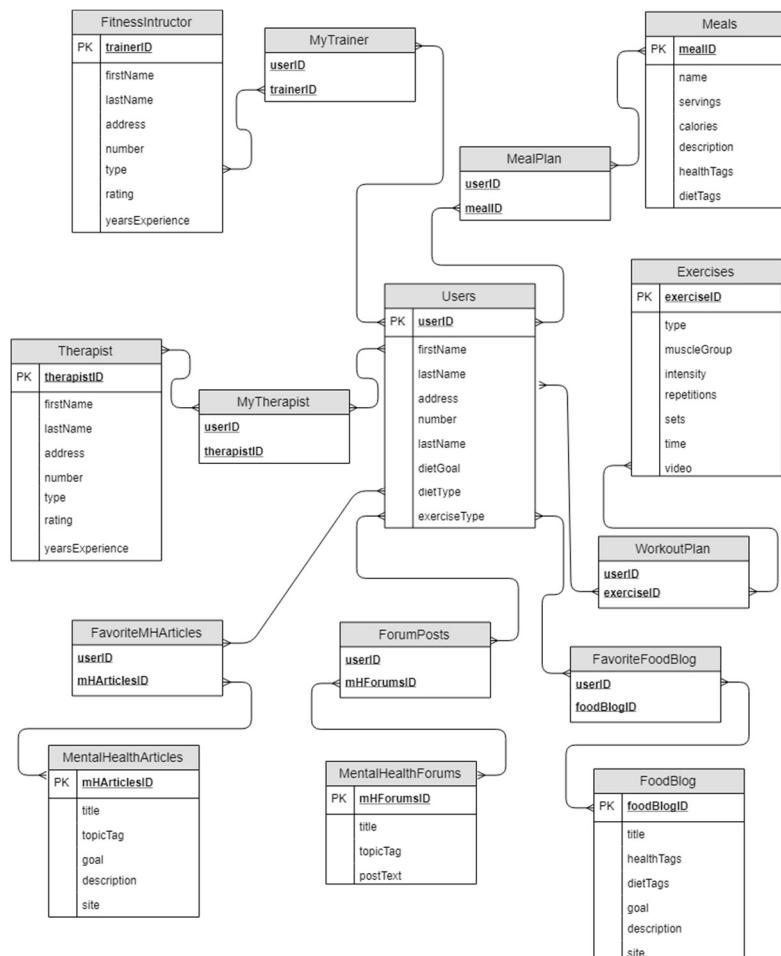


Diagram 3.d

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.

10. 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. These preferences are selected upon signing up for an account. Once logged in, upon landing on a page like recipes, a recommendation tailored to the user's preferences is made.

Activity Diagram:

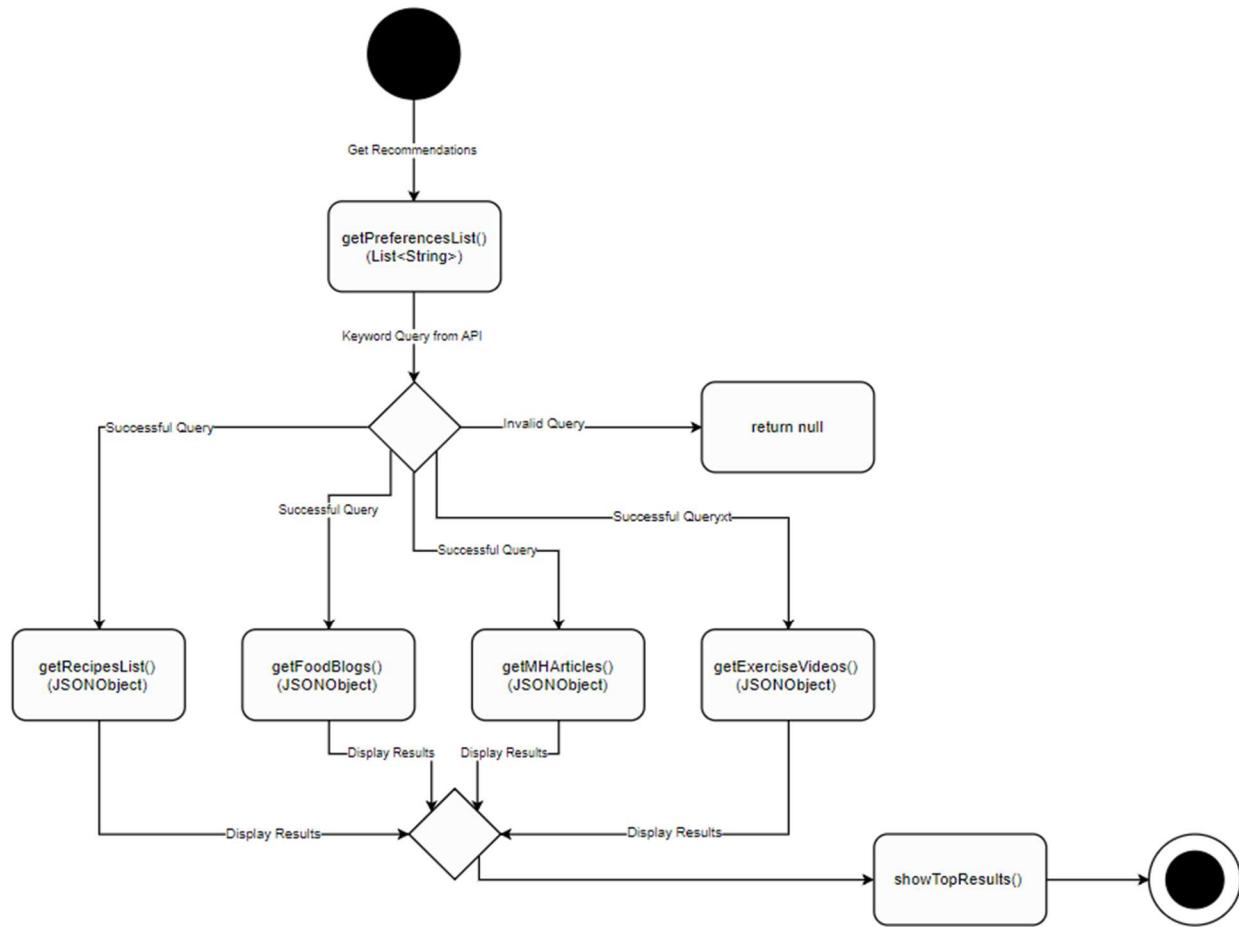


Diagram 4.a

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.

11. 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 Your Meal Plan in the Welcome Page. 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.

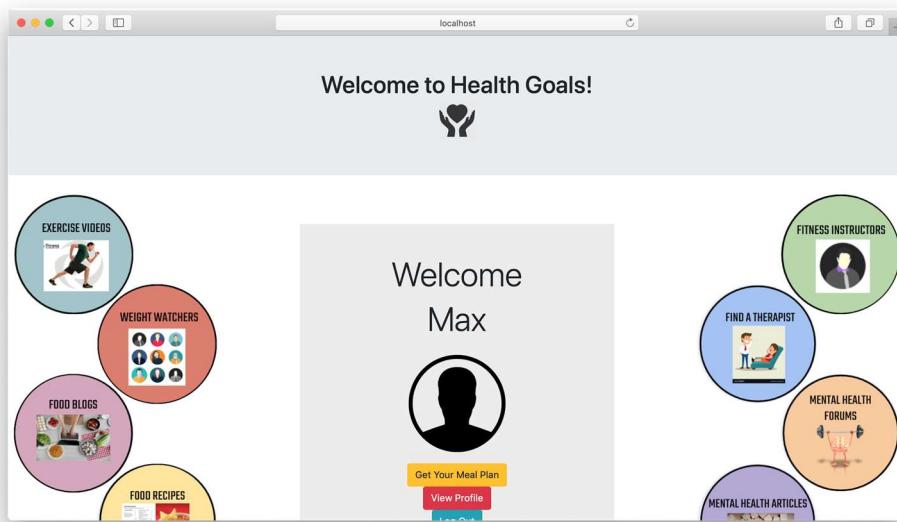


Figure 5.1.1 Welcome Page

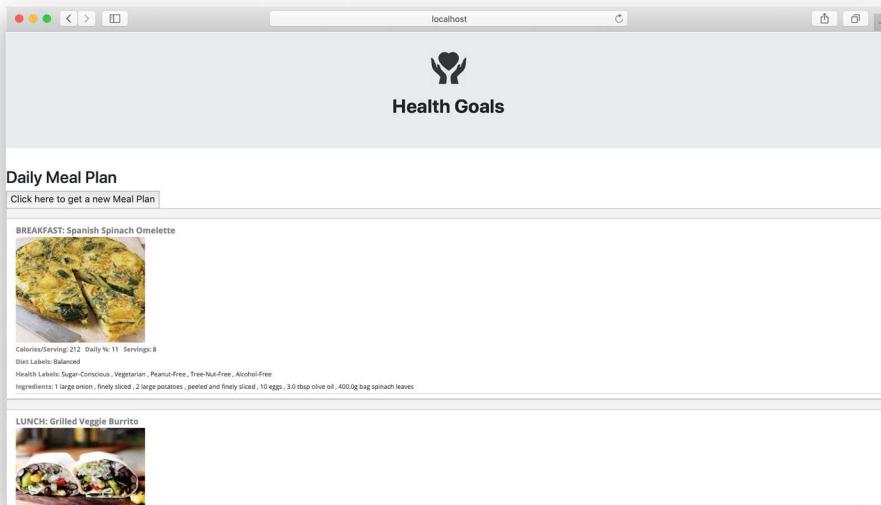


Figure 5.1.2 Meal Plan Page

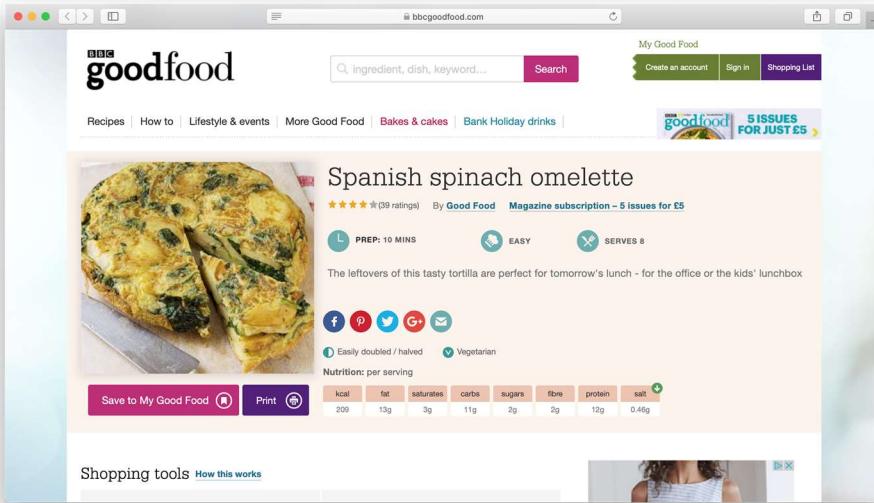


Figure 5.1.3 Recipe Page

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.

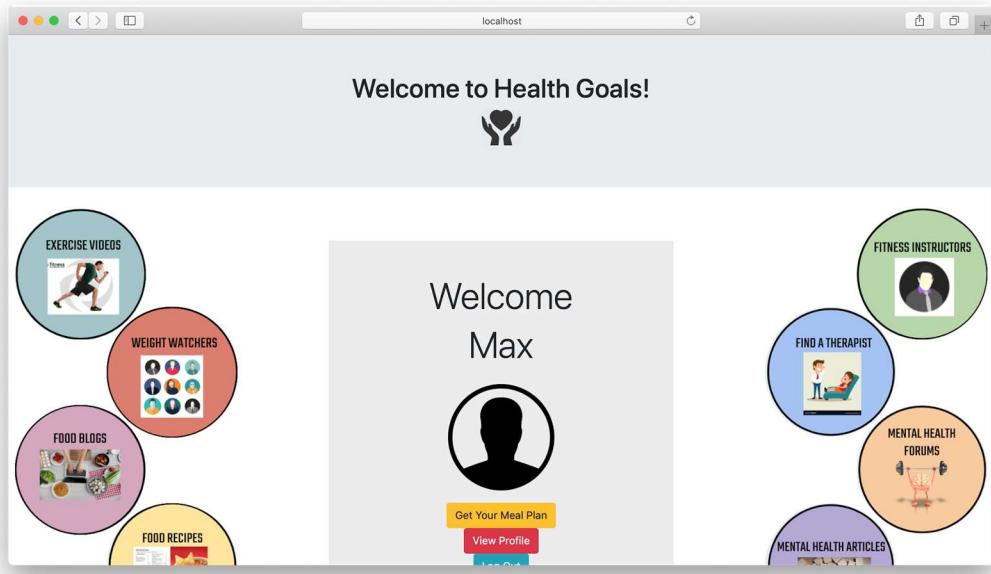


Figure 5.2.1 Login Page

Figure 5.2.2 MH Service Page

Find a Therapist!

Filter Results

Treatment Type

- Therapy
- Psychiatrist
- Psychology
- Counseling

Reset Filter

	David Z. Ritvo	2116 Sutter St, San Francisco, California, 94115	Phone Number: 4153465213				
	James W. Dimon	700 Lilly Rd NE, Olympia, Washington, 98506	Phone Number: 5092417349				
	Jennifer Williams	Specialty: Psychiatry Specializes in the diagnosis and treatment of mental disorders.					

Figure 5.2.3 MH Service Finder Page

Health Goals Exercise Videos Weight Watchers Food Blogs Food Recipes Fitness Instructors **Find Therapist** Mental Health Forums Mental Health Articles Log Out

Health Goals

[Back to list of Therapist](#)

David Z. Ritvo

1910 Olympic Blvd, Walnut Creek, USA, 94596
Phone: (925) 939-4159
Specialty: Psychiatry
Specializes in the diagnosis and treatment of mental disorders.
Title: MD

Dr. David Ritvo, MD, specialist in addiction psychiatry, child & adolescent psychiatry, clinical neurophysiology, cognitive & behavioral psychology, counseling psychology, counseling, health psychology, mental health counseling, psychiatry, psychology, psychiatry & neurology behavioral neurology & neuropsychiatry, psychiatry & neurology diagnostic neuroimaging, psychiatry & neurology forensic psychiatry, psychiatry & neurology hospice and palliative medicine, psychiatry & neurology psychosomatic medicine, psychiatry & neurology pain medicine, and occupational therapist mental health, currently sees patients in Walnut Creek, California.

Dr. Ritvo is licensed to practice medicine at California.

Ritvo has passed an automated background check which looked at elements including medical license status and malpractice screening (no history found).

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.

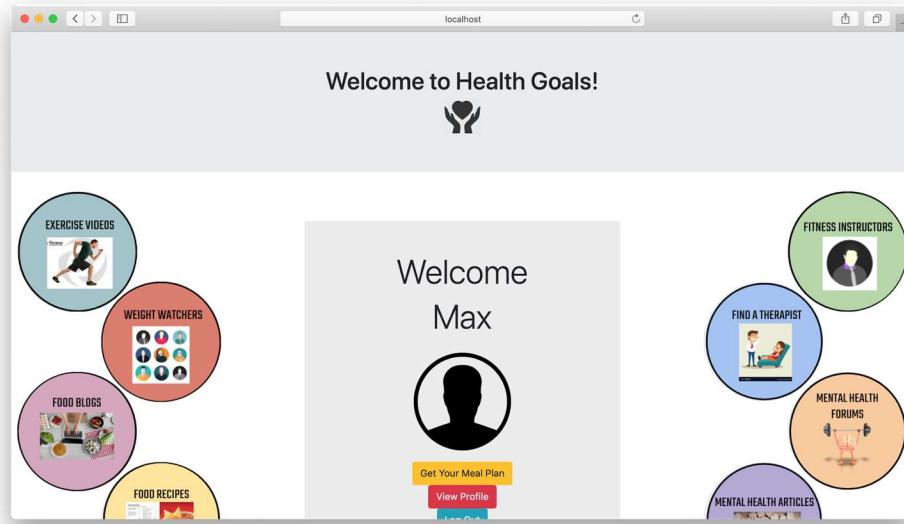


Figure 5.3.1 Login Page

A screenshot of the 'Mental Health Forums' section of the 'Health Goals' website. At the top, there's a navigation bar with links for Health Goals, Exercise Videos, Weight Watchers, Food Blogs, Food Recipes, Fitness Instructors, Find Therapist, Mental Health Forums, Mental Health Articles, and Log Out. Below the navigation is the 'Health Goals' logo. A heading reads 'Mental Health Forums: A safe discussion space for anyone who needs it'. There's a 'Start a Forum' button, a search bar, and a 'Filter Results' sidebar with checkboxes for various mental health topics like Depression, Anxiety, PTSD, Phobia, Eating Disorder, Self Harm, LGBTQ, and Panic Disorder, with a 'Save' button. The main content area shows a post titled 'Were you bullied at school? What happened and how did it affect you? Does it still affect you as an adult?' by 'srour@yahoo.ca'. The post discusses being bullied at school and its long-term effects. Another post below it is titled 'Bullying ruins lives' and discusses the impact of bullying. To the right, a 'Latest Forums' sidebar lists six recent posts with titles such as 'Were you bullied at school?', 'Bullying ruins lives', 'Your experience of being bullied?', 'Anxiety and Depression: Which one do I have?', 'Been a battle for 28 years with no medication', and 'Totally Lost with depressive thoughts'.

Figure 5.3.2 MH Forum Page

The screenshot shows a web browser window titled 'localhost' with a logo of two hands forming a heart at the top. Below it, the title 'Health Goals' is displayed. A sub-header reads 'Mental Health Forums: A safe discussion space for anyone who needs it'. On the left, there is a sidebar titled 'Filter Results' under 'Mental Health Topics' with checkboxes for Depression, Anxiety, PTSD, Phobia, Eating Disorder, Self Harm, LGBTQ, and Panic Disorder. It also includes 'Save' and 'Clear Filters' buttons. The main area has three input fields: 'Name of Forum:' (empty), 'Description:' (empty), and 'Topic:' (empty). At the bottom right is a 'Post Forum' button.

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 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.

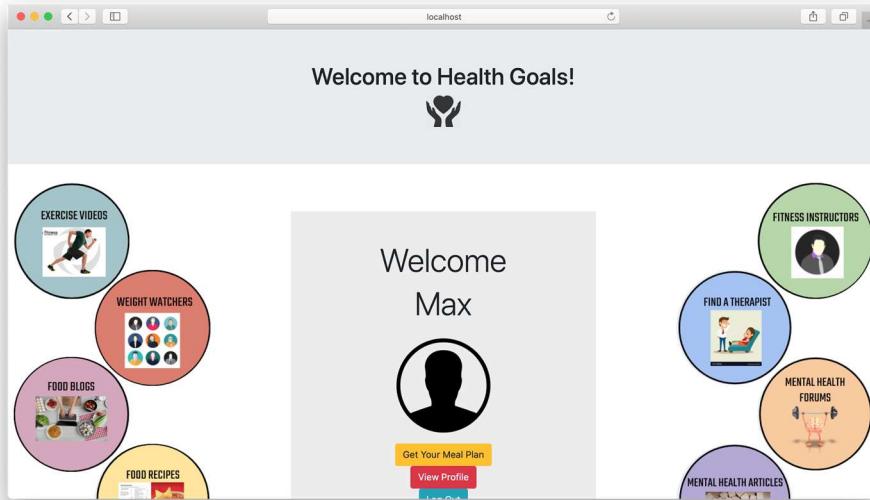


Figure 5.4.1 Login Page

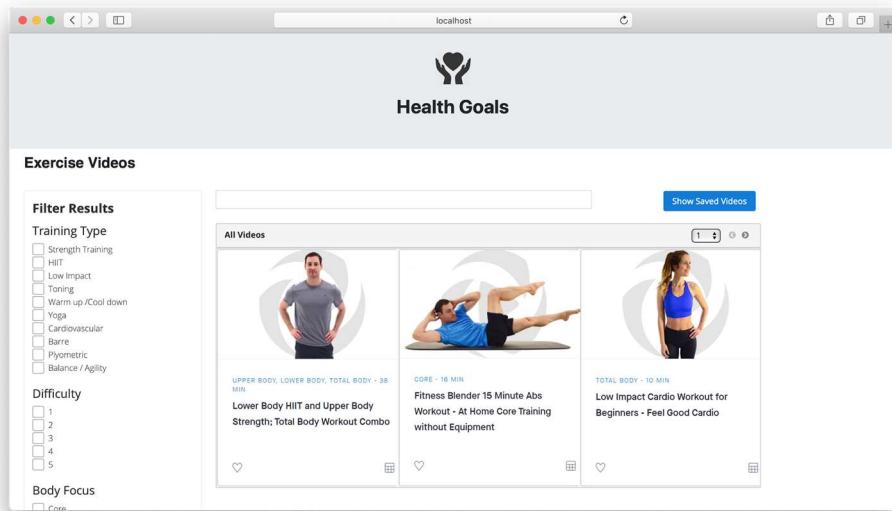


Figure 5.4.2 Exercise Search Page

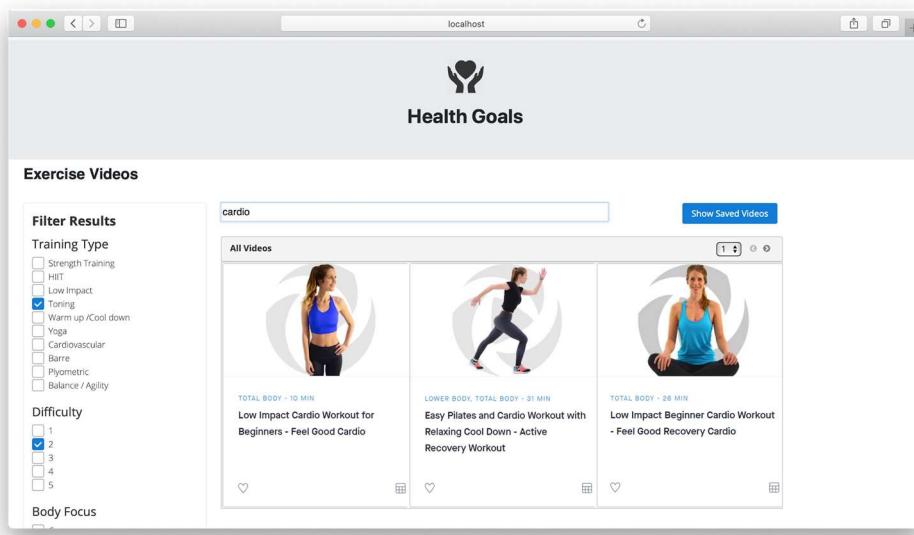


Figure 5.4.3 Exercise Options Page

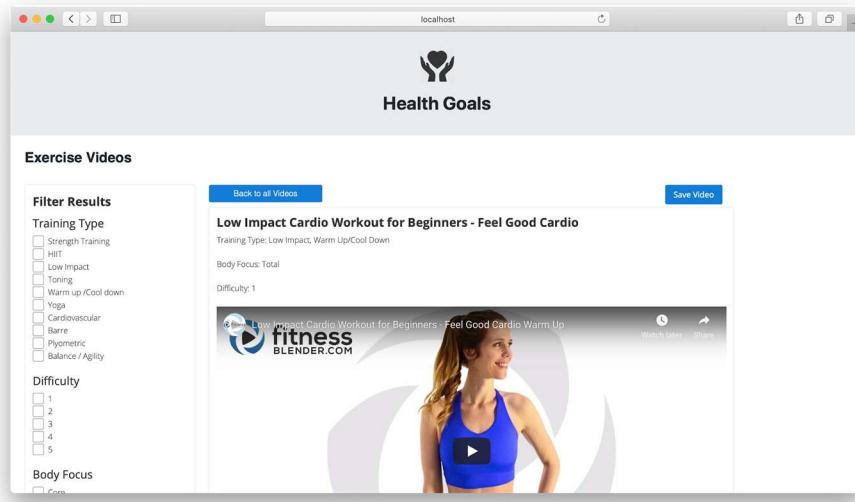


Figure 5.4.4 Exercise description Page

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.

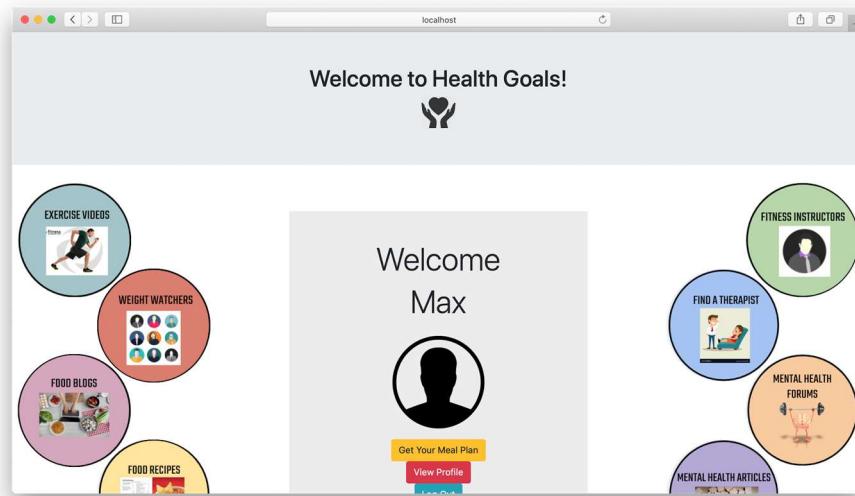


Figure 5.5.1 Login Page

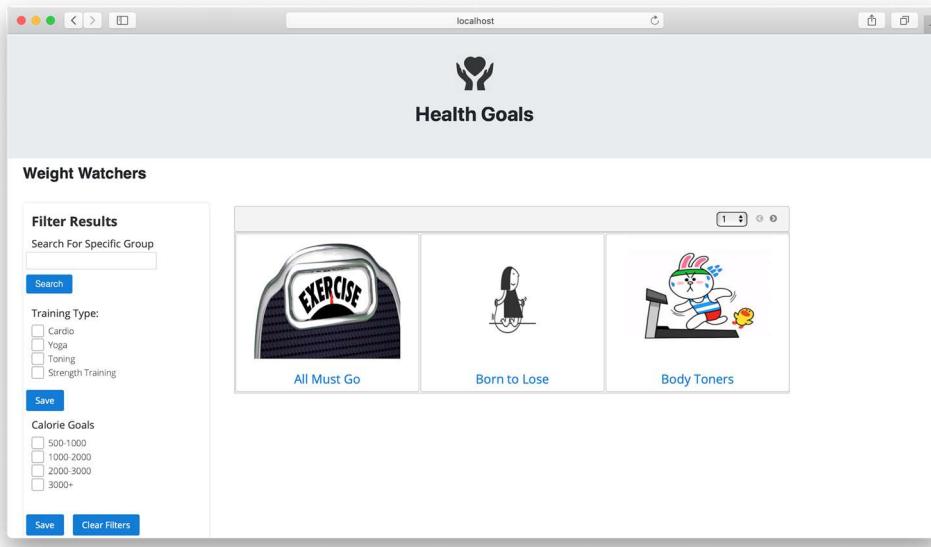


Figure 5.5.2 Find group Page

The screenshot shows a web browser window titled 'localhost'. The top navigation bar includes links for 'Health Goals', 'Exercise Videos', 'Weight Watchers', 'Food Blogs', 'Food Recipes', 'Fitness Instructors', 'Find Therapist', 'Mental Health Forums', 'Mental Health Articles', and 'Log Out'. The main content area is titled 'Health Goals' with a logo. Below this, there is a section for the 'All Must Go' group, featuring its image, Group Number (1), Calorie Goals (500-1000), and Training Type (Cardio). To the right of this is a table showing member statistics:

User ID	Member Name	Rank	% Goal Reached
1	Landon Dennis	2	95
10	Desiree Hampton	1	95
28	Georgette Powers	6	70
45	Shad Peters	7	65
115	Hoyt Everett	3	85
162	Kurt Lindsey	4	80
194	Willie Dorsey	5	75

A 'Join' button is located at the bottom left of the group description section.

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.

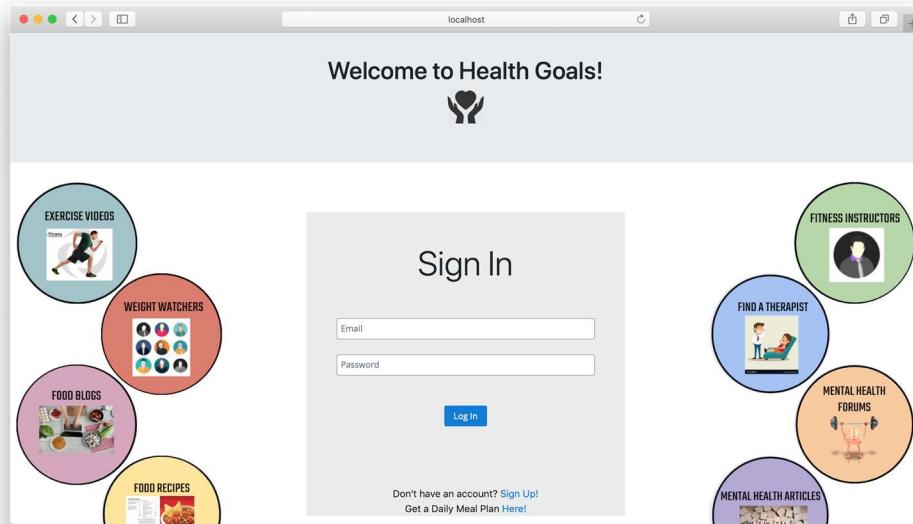


Figure 5.6.1 Login Page

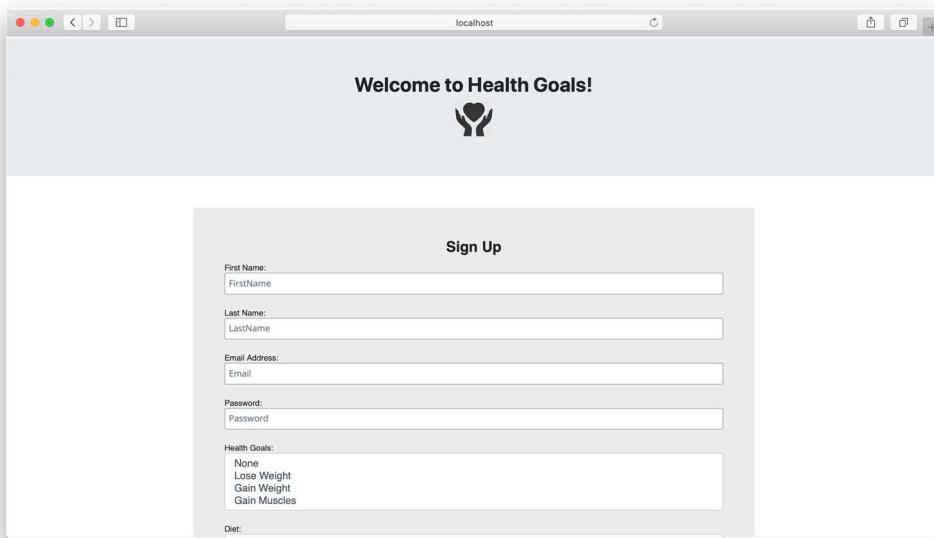


Figure 5.6.2 Create account Page

Sign Up

First Name: john

Last Name: Smith

Email Address: john.smith@gmail.com

Password: *****

Health Goals:

- None
- Lose Weight
- Gain Weight
- Gain Muscles**

Diet:

- None
- Balanced
- High Protein**
- Low Fat

Restrictions:

- None
- Vegan
- Vegetarian
- Sugar Conscious

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.

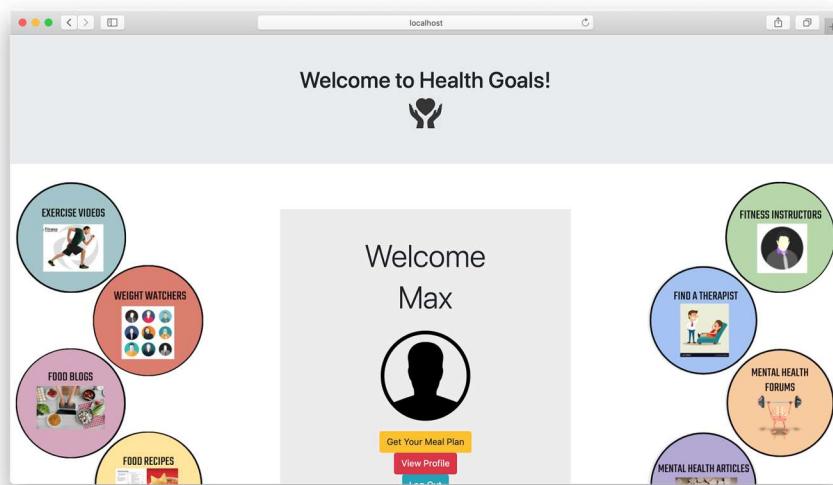


Figure 5.7.1 Login Page

Filter Results

Meal Type

- Breakfast
- Lunch
- Dinner
- Snack
- Drink

Diet

- Balanced
- High-Protein
- Low-fat
- Low-carb

Trending Recipes

1. Chicken Vesuvio
2. Chicken Paprikash
3. Chicken Gravy
4. Catalan Chicken
5. Citrus Roasted Chicken
6. Persian Chicken
7. Kreplach (Chicken Dumplings)
8. Twisting' Chicken
9. Chicken Piccata
10. Roast Chicken

Figure 5.7.2 Recipes search Page

SERIOUS EATS

RECIPES / INGREDIENT / MEATS AND POULTRY / CHICKEN

Chicken Vesuvio Recipe

DINNER TONIGHT / Your source for quick recipe every single week day.

BLAKE ROYER • PROFILE / TWITTER / FACEBOOK / CONTACT

PRINT-FRIENDLY VERSION

Published: December 15, 2011 | Last Updated: April 29, 2019

St. Ives ACNE CONTROL SCRUB WITH SALICYLIC ACID

CHECK IT OUT >

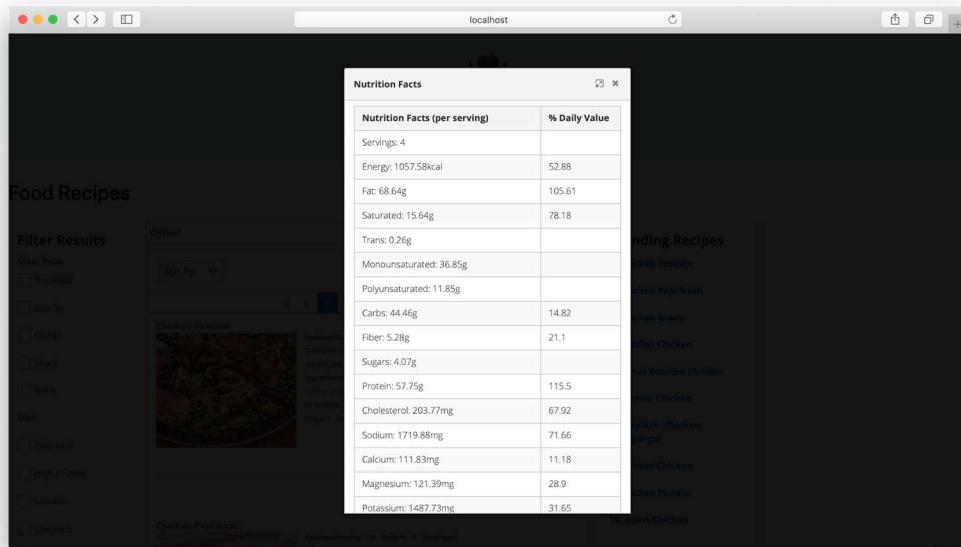


Figure 5.7.3 Recipes Page

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.

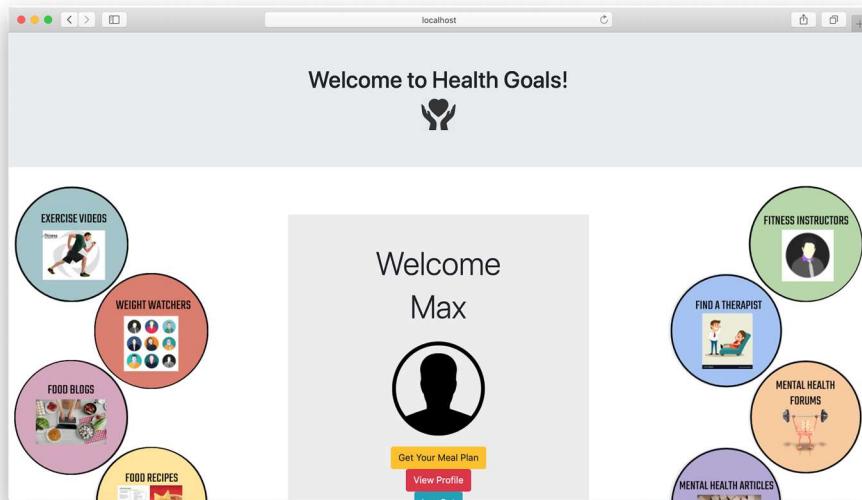


Figure 5.8.1 Login Page

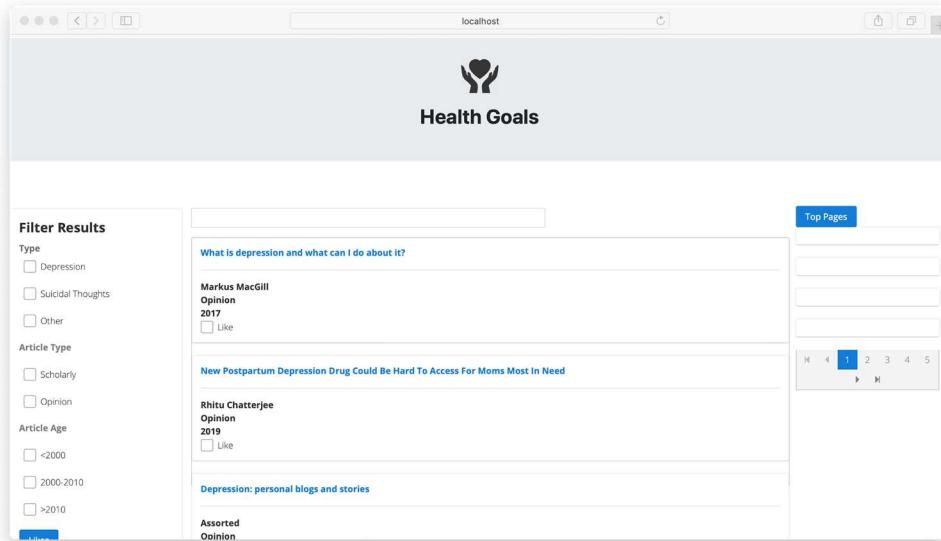


Figure 5.8.2 MH Search Page

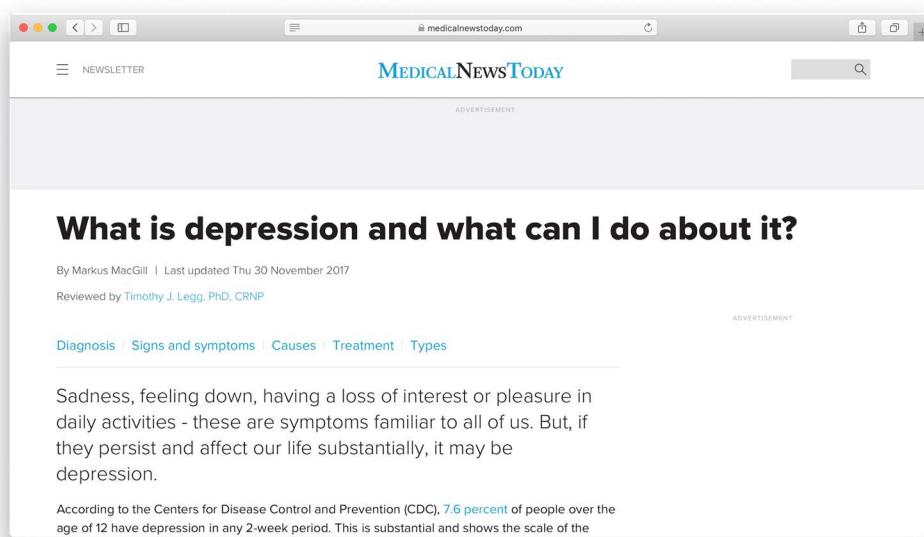


Figure 5.8.3 MH article Page

12. 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.

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">1. Visitor clicks a page icon from the home screen2. The indicated page loads
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:
<ol style="list-style-type: none"> 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 <p>The page loads, and any URLs are active</p>
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:
<ol style="list-style-type: none"> 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

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 6.6

Test: View Forums
Assumption: The forum icon is selected
Steps:
<ol style="list-style-type: none"> 1. The forum names load 2. The forums load in newest to oldest order
Success State: The forums load in the aforementioned form. The visitor cannot access a particular forum, nor may they post.
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">1. The group names load2. The visitor may use a small icon on the bottom right of the screen to see older groups
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

13. History of Work, Current Status, and Future Work

a. History of work

1. Comparing the first and second demo I believe there is a lot of change. For example at first we only had most of our cases hard coded in with HTML. However in the second demo we had most of our use cases features working within the web application.
2. When completing the report everyone did what they were supposed to however, we had different formats for each of our parts. In the domain models there was the most difference in formats some people hand drew them and others did it digitally which made the report look less adequate. By the second report we took the feedback that was given to us and fixed the report accordingly and made it look appropriate.
3. In the middle of the project, one of the struggles we found was including the filters for certain search pages. How the filters worked was that whenever the user would check off certain filters it would use those filters as strings and parameters to request the API to only search for those filters. However by doing this one had to already have the filter results loaded with the unfiltered results which generated too many requests toward the APIs and unable to get since we were using the free versions of those APIs. One way to overcome this was to put the results in an array and use a search functions so that if the result did not match with the filter it would dynamically be deleted.
4. Using the database was not an easy task for some of our group members, there was various topics one had to be knowledgeable about in order to successfully connect and make functions using the database. However using the resources the other group members had uploaded and with their own knowledge they were able to quickly catch up and complete their responsibilities.
5. There were certain user cases we were not able to develop the way we wanted to. For example “Find Instructors” we wanted to use an API in order to implement this feature, similarly to how “Find Therapists” worked, however we were not able to find an API that really did what we needed it to. So to address this we used professional trainers’ public profiles and implemented their data into our database.

b. Key Accomplishments

1. Developed a website application that can be used as health wellness social media
2. Implemented an aesthetically pleasing UI for our website application
3. Integrated a database that holds various user data and instructor data that can be used for certain features
4. Application can recommend user videos and recipes based on their profile data
5. Users can save videos and recipes to their profile
6. Application can serve both users and visitors
7. Integration between UI and Database

c. Future Work

We have successfully made a website application that will act as a social media for health and mental wellness. Our future plans are to integrate the therapists and find instructors with the users. We want to use the users goals and experience to recommend fitness instructors. Another thing we want to integrate is to use the user's workout history and eating habits to suggest certain mental conditions a user could be about to suffer. With these suggestions we can also recommend therapists based on their specialties.

With more users registered on our page we can also use their data recommend certain workouts that alike users have had success with. Another thing we could do with more users is develop the "Weight Watchers" feature more by having more challenges and also rewarding our members.

We would also like to integrate a special user for instructors so that they can use our page to advertise their services. And with more instructors in our database we would be able to reach more people from different locations. Additionally we want to add way for the instructors to be more active with our users by having things like live streams where they can talk with the audience and promote health and mental wellness.

14. Project Management

As our project has now come to an end, here is a brief outline and schedule of the project goals we had and how they were completed following the second demo:

Deadlines	Description of duties
April 26th	Revise our website use cases based on immediate feedback received from the demo, keeping in accordance with our business objectives.
May 3rd	Start finalizing our page elements, as well as discuss any final changes we want to make.
May 5th	Finish the last few sections of importance, such as Effort Estimation, Interaction Diagrams, and Class Diagram and Interface Specification. These sections are important in showing what features we implemented into our website.

Our team of nine people for this project was divided into three subgroups. Early on we established weekly meetings, where we assigned different subgroups different sections to work on. This ensured that everyone got a chance to work on our project report and through proper planning we were able to meet our deadlines every week. Additionally, there can be certain inconsistencies between sections on our report, however before submitting every report:

- We went through all the sections and made 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 gave 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 allowed us to stay on track and finish our project reports in a timely fashion.

15. References

1. Biometric Heart Monitoring:
<https://www.ece.rutgers.edu/~marsic/books/SE/projects/HealthMonitor/2012-g3-report3.pdf>
2. Fitbit Health Monitoring Analytics:
<http://eceweb1.rutgers.edu/~marsic/books/SE/projects/HealthMonitor/2014-g5-report3.pdf>
3. Gantt Chart Assistance:
<https://www.smartsheet.com/blog/gantt-chart-excel>
4. Mental Health Issues in America: <http://www.mentalhealthamerica.net/issues/state-mental-health-america-2018>
5. Mental Health Statistics: <https://www.nami.org/learn-more/mental-health-by-the-numbers>
6. Project Management Examples: <https://marketbusinessnews.com/financial-glossary/project-management/>
7. Publisher-Subscriber Pattern: <https://docs.microsoft.com/en-us/azure/architecture/patterns/publisher-subscriber>
8. Talking Heart:
<https://www.ece.rutgers.edu/~marsic/books/SE/projects/HealthMonitor/2012-g2-report3.pdf>
9. UML Activity Diagram Tutorial: <https://www.lucidchart.com/pages/uml-activity-diagram>
10. UML Class Diagram Tutorial:
<https://www.youtube.com/watch?v=UI6lqHOVHic>

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

12. Use Case Points:

https://en.wikipedia.org/wiki/Use_Case_Points

13. Workout With Friends:

<https://www.ece.rutgers.edu/~marsic/books/SE/projects/HealthMonitor/2012-g1-report3.pdf>