Health Tracking System

Group 'HONGKONG FUSION'
James Large, Pratik Gurung, Michal Zak, Mercury Aimnh

This document details the design of a health tracking system to meet a problem specification supplied by Aviva. The Health Tracker aims to help inform users of basic health information as well as enable them to track their diet and fitness regime with simple goals and reports of their progress over time.

Three central capabilities are required by the solution. The capability to:

1. Record on-going lifestyle details such as exercise taken and diet
2. Ability to set goals
3. And view a history of this information

Extra capabilities could include:

4. Ability to create and manage groups
5. A built-in social media / messaging system

Contents

MoSCoW

Must

- Allow users to create an account which persists over sessions, requiring a username, email, and password
- Allow users to securely log into their account with the username and password specified when the account was created
- Allow users to upload and log their meals, activities, mood, and weight
- Allow users to view their logs and generate reports from them
- Allow users to set goals for themselves, such as a target weight
- Allow users to check their progress towards their goals by comparing their log data against them
- Notify users when a goal is met or deadline passed
- Allow users to create new groups and join existing ones
- Allow group leaders to set goals for the entire group
- Allow members to leave a group

Should

- Allow users to edit their account profile with more personalised data about themselves, such as a biography, living location, etc.
- Allow group leaders to disband a group
- Have input validation across the site, e.g user should not be able to enter negative quantities

Could

- Allow users to privately message one another within the app
- Allow users to leave messages on another user's public profile
- Allow groups to release messages to all their members
- Implement an achievement system to reward users for, e.g, consistency in meeting goals

Won't

- Automatically enter data, e.g interface with a movement tracker, pedometer
- Be globally compatible, initial application will only be available on a limited collection of popular web-browsers
- Murder you in your sleep

Similar Systems Analysis

SuperTracker

Pros

- Gives targets and stores actual work time
- Stores daily food amount
- Gives Diet information(Daily/Weekly amounts) for different food groups
- Daily allowance related to personal height/weight, average activity time and weight goal

- Complex and covers everything you would like to know
- One ribbon to show important daily information(which you can constantly refer to)
- Stores nutrients (vitamins and minerals)
- Stores daily limits (empty calories, oils, sat.fat, sodium)
- Store your daily food group consumption and state if it's under or over
- Can create own recipe and can output to PDF, Excel, Word(with vitamins and minerals)
- Metric/Imperial
- Has reports for everything. Food Tracker, Physical Tracker

Cons

- Messy UI
- Not Automated
- Diet information layout is overly complex, too much information at once. Blocks of texts. More emphasis on minimalistic graphics would help.
- Throws too much as user, might find it overwhelming.
- Too many links and constants navigation hoops(pages have to be constantly reloaded)
- Food information is on different pages, should be better grouped
- Daily Limit, should be clickable and should give more information
- Have functions that pretty much do the same thing, i.e My Recipe and My Combo
- Doesn't allow to add your own exercise
- No social networking

Structure

- User doesn't have to log in(might not save personal goals/details)
- Formatting and layout is confusing and contains too much text
- Multiple pages and Links

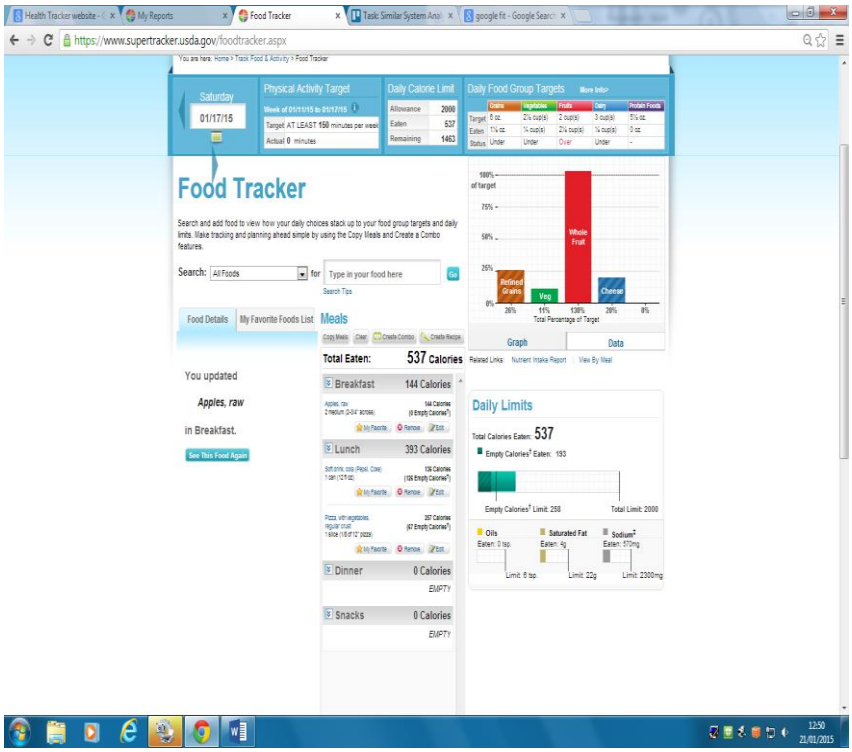


Figure 1: Food Tracker dashboard

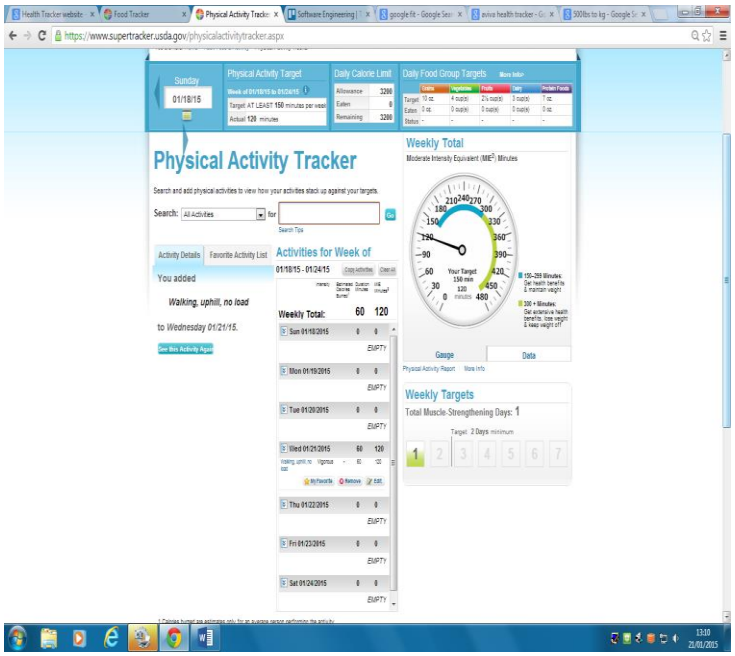


Figure 2: Physical Activity Tracker

Overview

This system is an all-in-one tracker. It stores every possible piece of information a user could need to keep track of their health. It accounts both for physical activity and diet which is similar to our project. The system focuses on the daily intakes and limits hence a ribbon at the top of the page which displays at all times important daily information. This approach makes it easier for a user to explore the website and at the same time refer to their own records. Daily limits/allowance based on personal height and weight, average activity time and weight goal would be a great feature to implement. The system has a big problem with the layout and overall complexity which results in anti-user-friendliness. The interaction is not straight-forward and requires time to get used to. The system does not provide a lot of automation either effectively forcing user to input every piece of data themselves. The UI creates a sense of confusion and discourages a potential user from signing up.

Our system will implement the top-bar/ribbon idea and the overall integration of physical activity and diet. We will aim to simplify the user experience and minimise the amount of hyperlinks and page reloading.

Google fit

Pros

- It is simple and accessible
- Google account to log in
- Shows daily activity time and compares to goal
- GPS
- Personal information, Height and Weight
- Counts steps
- Graphical View
- Colour coding

- Connected to mobile, works in background
- Set Goal
- Monitors Average Activity and modifies accordingly

Cons

- Only activities doesn't track diet
- Too Simple, not many features
- Navigation confusing when just starting off

Structure

- Purely graphical, not much text
- More user friendly
- Easy to reach
- Goal Based
- Not much Navigation (One page)

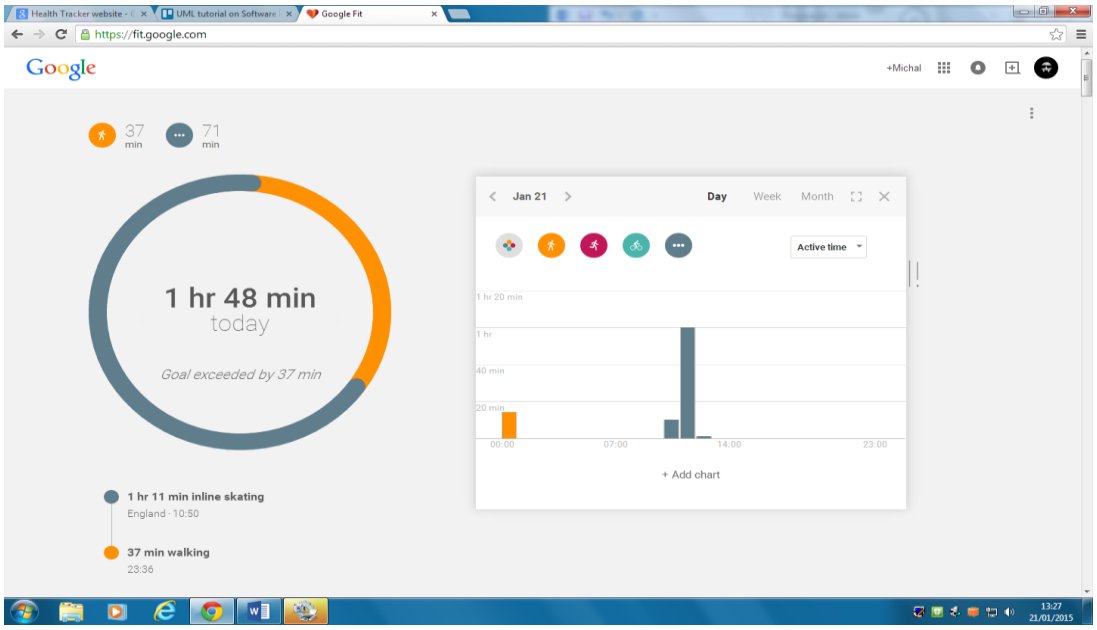


Figure 3: Home page

Overview

This system track only physical activity and it's time-based. Gamification of health tracking can be found here through the use of goals which are the most important part of the system. This approach enables Google Fit to be more focused and streamlined. Dynamic graphics help to convey data in an accessible way. The system is user-friendly and easy to set up. Additionally it helps that it's integrated into the Google infrastructure and acquires the data from the user's phone. The overall system is too simple but should be easily extendable.

The physical activity tracker of our system will be modelled after Google Fit's approach which focuses on the time spent doing exercises. We will also aim to keep our UI as uncluttered and user-friendly as possible because we find it makes the user experience better.

Myfitnesspal.com

Structure

- Account creation; enter details, height, weight, initial goals. Gives immediate feedback e.g target calorie intake per day. Immediate refer-a-friend type system, emphasis on group/social work for the get-go.
- Built in messaging system, peer to peer and messaging for goals met/missed
- Built-in public and group blogging
- Site takes a passive role, user enters information and visualises the data, graphs, live reports etc.
- Upload food eaten, exercises done – site has own very complete database e.g calories burnt while cycling at 10mph for 20 minutes, 14mph…
- Interface with many typical apps

Pros

- Lots of options
- Relatively clean look of the site itself

Cons

- Quite complicated to use – many different sections. 'Too many clicks'
- Passive role, doesn't 'pester' user if missing goals etc
- Ads in the way, distracting

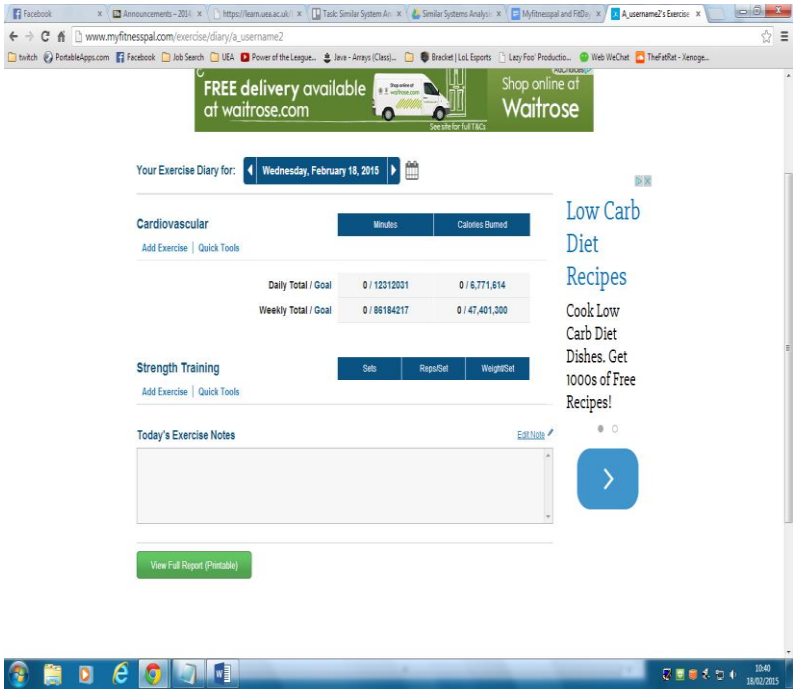


Figure 4: Inputting into Exercise Diary

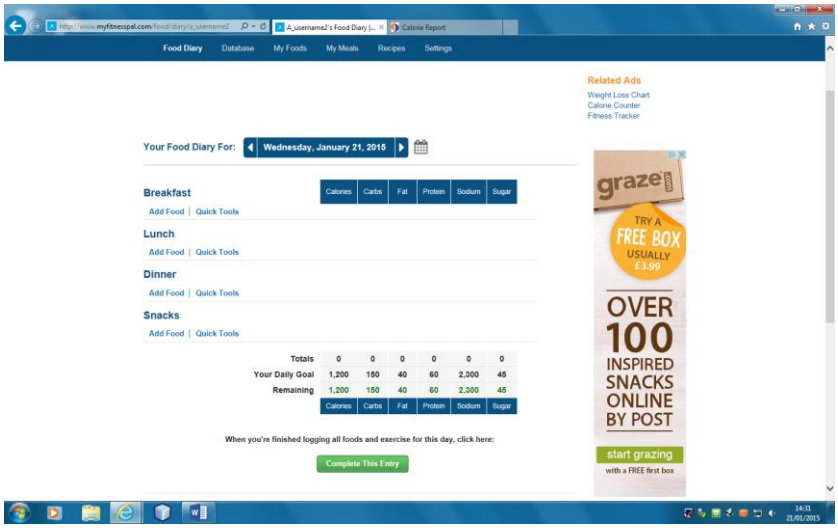


Figure 5:Inputting into Food Diary

Overview

Looking at this website the clean layout makes it easier to track information that the user would like to store, which is something we would like to stick to within our system.

When inputting Exercise into the Exercise Diary the website allows the user choose between Cardiovascular and Strength Training, this feature is something that will be added in our system as we would like to be accurate with what the user does specifically and how many calories he burns. Adding Meals are split into Breakfast, Lunch, Dinner and Snacks this reduces confusion as people have meals at different times. Another useful feature that myfitnesspal has is allowing the user to add meals and exercises from the pervious day, this is something we would like to implement as it may help to reduce the amount of times a user will have to look for his/her exercise or meal within the database.

Fitday.com

Structure

- Basic account creation
- Paid membership model
- Pairs user with 'expert dietician' for fee
- Personalised human contact to discover proper goals/methods

Pros

- Pairing with expert provides more personalised help, perhaps more motivation

Cons

- Actual content takes a few seconds to load, ads prioritised
- Site functionality is very basic, style over substance
- Feeds on a general lack of critical thinking skills

Figure 6: Graphical Output of Calories Eaten and Burnt

Overview

FitDay also had a simple layout but was too simple, not helpful to user but easy to understand. The website had a paid scheme, which seems unusual for the little amount of functionality it had.

The website showed reports in a graphical layout, which gives information to the user in a clear method, within our system we would like to have this feature so users can see their daily food intake and other piece of information easily.

Use Cases

User Registration

USE CASE NAME	User Registration
Goal in Context	Allow the Actor to record his Credentials onto the System, so he can later Log In
Scope & Level	Overall System
Preconditions	System must be running and active and awaiting Actor to input details

Success End Condition	Actors Credentials and details are stored onto the System's database	
Failed End Condition	Actors Credentials and details cannot be stored onto the System's database	
Primary Actor	User	
Trigger	Actor prompting System that he/she would like to store his Credentials	
SUCCESS SCENARIO	**Step**	**Action**
	1	Actor prompts System that he/she would like to store his Credentials
	2	System displays form for Actor to input his/her details
	3	Actor input all his/her details, including: Username, Password, Full name, Email Address etc
	4	System verifies that Username has not been taken by previous Actor and Email Address is not already being used(Actor is not attempting to make a second account)
	5	System checks Email Address is in correct format
	6	System saves Actors details onto System's database
ALTERNATIVE SCENARIO	**Step**	**Branching Action**
	4a	System notices Username has already been taken
	5a	System asks Actor to use a different Username
	4b	System notices Actor's Email address has already been taken
	5b	System prompts Actor that the Email address has been taken
	6b	System asks Actor if he has forgotten his Log In Details or if he/she would like to use another Email Address
	5c	System prompts Actor that Email Address is in wrong format
	6c	System asks Actor to re-enter Email Address
RELATED INFORMATION		
Priority	Top priority	
Performance Target	Actor will take 2-3 minutes to input his/her details, adding Actors details to System database should take 5 seconds	
Frequency	Frequent, Everything a new Actor wants to register	
Subordinate Use Cases	User Log In	
Channel to Primary Actor	-	

Secondary Actors	-
Channel to Secondary Actors	-
OPEN ISSUES	What happens when the Email address has already been taken?
SCHEDULE	Due date is version 1.0 release
AUTHOR	Pratik Gurung, 23/01/2014

User Log In

USE CASE NAME	User Log In	
Goal in Context	Allow the Actor to the System and he/she can get access to his/her environment	
Scope & Level	Overall System	
Preconditions	System is running and active and awaiting a log in by an Actor	
Success End Condition	The Actor is identified using his Authentication his/her credentials	
Failed End Condition	The Actor cannot be identified using his Authentication his/her credentials	
Primary Actor	User	
Trigger	Actor presses the log in button	
SUCCESS SCENARIO	**Step**	**Action**
	1	System displays a window requesting the Username and password of the Actor
	2	The Actor enter his/her Username and password into the system
	3	System verifies the Username and password
	4	System displays the Actors's environment
ALTERNATIVE SCENARIO	**Step**	**Branching Action**
	3a	System cannot verify Username and password
	4	System reports a Login error and restarts the user case
RELATED INFORMATION		
Priority	Top priority	
Performance Target	Log in verification should be completed within 5 seconds	
Frequency	Frequent, every time an Actor wants to Login	
Subordinate Use Cases	-	
Channel to Primary Actor	User Interface	

Secondary Actors	-
Channel to Secondary Actors	-
OPEN ISSUES	How many attempts is the user allowed to do?
SCHEDULE	Due date is version 1.0 release
AUTHOR	Pratik Gurung, 23/01/2014

Browse Another User's Environment

USE CASE NAME	Browse Another User's Environment	
Goal in Context	Allow the Actor to Browse the Environment of a Secondary Actor, without needing to Log In	
Scope & Level	Overall System	
Preconditions	System must be running, Secondary Actor must exist and Primary Actor must know Secondary Actors Username/Full Name	
Success End Condition	System displays Secondary Actor's Public Environment to Primary Actor	
Failed End Condition	System prompts Primary Actor that Secondary Actor's Environment cannot be displayed	
Primary Actor	User	
Trigger	Primary Actor prompting System that he/she would like to view Secondary Actor's Environment	
SUCCESS SCENARIO	**Step**	**Action**
	1	Actor prompts System that he/she would like to view Secondary Actor's Environment
	2	System searches for Secondary Actor's details
	3	System displays Secondary Actor's Environment
ALTERNATIVE SCENARIO	**Step**	**Branching Action**
	2a	System cannot find Secondary Actor's details
	3a	System prompts Primary Actor that Secondary Actor's details cannot be found
RELATED INFORMATION		
Priority	Top priority	
Performance Target	Primary Actor will take 10 second to input Secondary Actor's details, System searching and displaying Secondary Actor's details should take less than 5 seconds	

Frequency	Medium, when a User wants to view another User's details
Subordinate Use Cases	-
Channel to Primary Actor	-
Secondary Actors	-
Channel to Secondary Actors	-
OPEN ISSUES	-
SCHEDULE	Due date is version 1.0 release
AUTHOR	Pratik Gurung, 23/01/2014

Log Food Intake

USE CASE NAME	Log Food Intake	
Goal in Context	To log the food the user have eaten (type, amount, date)	
Scope & Level	Overall system	
Preconditions	• system is live/active • user has an account • user is logged in	
Success End Condition	The data is registered and the calorific count is recorded and displayed.	
Failed End Condition	The data is not registered.	
Primary Actor	User	
Trigger	Actor pressing the "LOG FOOD INTAKE" button	
SUCCESS SCENARIO	**Step**	**Action**
	1	The system displays a window requesting the type of food
	2	User chooses from a list by browsing or searching
	3	User chooses the amount of food they have eaten
	4	User chooses the date on which they have eaten the food
	5	User click on "LOG" button
	6	The system displays a window with the confirmation of entry, recap of what was input and its calorific value
	7	User clicks on "Return to Dashboard" and is returned to Dashboard
ALTERNATIVE SCENARIO	**Step**	**Branching Action**

	7b	User clicks on “Add more” and the process is repeated
RELATED INFORMATION		
Priority	High	
Performance Target	Logging the food intake should be completed within 10-15 seconds	
Frequency	Frequent, user	
Subordinate Use Cases	-	
Channel to Primary Actor	User interface	
Secondary Actors	-	
Channel to Secondary Actors	-	
OPEN ISSUES	Adding food intake can be made into a list that is updated and then persisted all at once	
SCHEDULE	Due date is version 1.0 release	
AUTHOR	Michal Zak, 23/01/2015	

Create Meal

USE CASE NAME	Create Meal	
Goal in Context	User creates a new Food Item that will be added to the Food Intake	
Scope & Level	User’s scope	
Preconditions	- system is live/active - user has an account - user is logged in	
Success End Condition	User’s has created a new Food Item to add to Food Intake	
Failed End Condition	Food Item not created	
Primary Actor	User	
Trigger	User presses ‘CREATE MEAL’ button	
SUCCESS SCENARIO	**Step**	**Action**
	1	System asks User to input ingredient
	2	System asks User to input the amount used for that ingredient
	3	System asks User if there is more ingredients to add, if so GOTO step 1

	4	User selects 'FINISH'
	5	System creates meal
ALTERNATIVE SCENARIO	**Step**	**Branching Action**
RELATED INFORMATION		
Priority	High	
Performance Target	System should take 5 seconds per ingredient	
Frequency	Minimal, Unless User creates unique meals reguarly	
Subordinate Use Cases	-	
Channel to Primary Actor	-	
Secondary Actors	-	
Channel to Secondary Actors	-	
OPEN ISSUES	What happens when User only selects one ingredient	
SCHEDULE	Due date is version 1.0 release	
AUTHOR	James Large, 28/01/2015	

Log Physical Activity

USE CASE NAME	Log Physical Activity	
Goal in Context	To log the physical activity the user has performed (i.e. running)	
Scope & Level	Overall system	
Preconditions	• system is live/active • user has an account • user is logged in	
Success End Condition	The data is registered and the physical activities chart is updated	
Failed End Condition	The data is not registered.	
Primary Actor	User	
Trigger	Actor pressing the "LOG ACTIVITY" button	
SUCCESS SCENARIO	**Step**	**Action**
	1	The system displays a window requesting the type of activity.
	2	User chooses from a drop down list by browsing or searching
	3	User inputs the duration of the activity in hours and minutes

	4	User confirms the input
	5	The system displays a window with confirmation of entry
	6	User clicks on “Return to Dashboard” and is returned to Dashboard
ALTERNATIVE SCENARIO	**Step**	**Branching Action**
	2b	User chooses “Other” and inputs their own name of the activity
	3b	User inputs type of exercise time, rep or speed based
	6c	User clicks on “Add more” and the process is repeated
RELATED INFORMATION		
Priority	High	
Performance Target	Logging the physical activity should take within 15-20 seconds	
Frequency	Frequent	
Subordinate Use Cases	-	
Channel to Primary Actor	User interface	
Secondary Actors	-	
Channel to Secondary Actors	-	
OPEN ISSUES	Converting distance and time to kCal? Taking more information?	
SCHEDULE	Due date is version 1.0 release	
AUTHOR	Michal Zak, 23/01/2015	

Log Weight

USE CASE NAME	Log Weight
Goal in Context	To log the user’s weight along with the date
Scope & Level	Overall system
Preconditions	• system is live/active • user has an account • user is logged in
Success End Condition	The data is registered, the weight is updated.
Failed End Condition	The data is not registered.
Primary Actor	User
Trigger	Actor pressing the “LOG WEIGHT” button

SUCCESS SCENARIO	Step	Action
	1	The system displays a window showing current weight and requesting new weight
	2	User inputs the numerical value (choose from lbs and kg)
	3	User chooses the date of the weight measurement (cannot be a future date)
	4	User clicks on the “LOG” button
	5	The system displays a window with the confirmation of entry and recap of what was input
	6	User clicks on “Return to Dashboard” and is returned to Dashboard
RELATED INFORMATION		
Priority	High	
Performance Target	Logging the weight should be completed within 10 seconds	
Frequency	Sporadic, depends on if the weight is important to the user	
Subordinate Use Cases	-	
Channel to Primary Actor	User interface	
Secondary Actors	-	
Channel to Secondary Actors	-	
OPEN ISSUES	-	
SCHEDULE	Due date is version 1.0 release	
AUTHOR	Michal Zak, 23/01/2015	

Set Personal Goal

USE CASE NAME	Set Personal Goal
Goal in Context	The user needs to be able to set themselves a goal to work towards; target weight, target 100m sprint time, etc.
Scope & Level	User’s scope
Preconditions	- system is live/active - user has an account - user is logged in
Success End Condition	Data about a goal is created for the user which can be checked for progress at a later time

Failed End Condition	No goal is set, invalid goal etc.	
Primary Actor	User	
Trigger	User presses 'SET GOAL' button	
SUCCESS SCENARIO	**Step**	**Action**
	1	System asks user for type of goal to set, update a previous goal, set a completely new one, etc
	2a	User selects a type of goal (new goal)
	3	System displays specialities of that goal, specific targets, e.g specific weights if a goal to lose weight is chosen
	4	User selects their specific goal
	5	User selects valid time-frame to meet goal
	6	System saves goal
ALTERNATIVE SCENARIO	**Step**	**Branching Action**
	2b	User selects one of their previous goals to update (requires precondition of goals being previously set, and goal having been met (? see OPEN ISSUES))
	3b	System displays that goal and the users progress in reaching it
	4b	User changes goal parameters to update the goal, e.g make it more challenging
RELATED INFORMATION		
Priority	Core	
Performance Target	Excluding user's time spent making decisions, system should take less than 5 seconds	
Frequency	Dependant on user, generally low frequency	
Subordinate Use Cases	In alternative scenario (b), Check Your Goals	
Channel to Primary Actor	-	
Secondary Actors	System	
Channel to Secondary Actors	User Interface	
OPEN ISSUES	How free should the user be to update their already set goals, e.g cancel goals before completion, make goals easier. Should they be 'forced' to continue, or be allowed to make unreasonably hard goals more reachable	
SCHEDULE	Due date is version 1.0 release	

AUTHOR	James Large, 26/01/2015

View Goal Progress

USE CASE NAME	View Goal Progress	
Goal in Context	To check the User's progress towards their personal goals	
Scope & Level	User's scope	
Preconditions	- System is live/active - User has an account - User is logged in - User has goal(s) set	
Success End Condition	User is able to compare their current progress with their goals	
Failed End Condition	User is not able to compare their current progress with their goals	
Primary Actor	User	
Trigger	User pressing 'VIEW GOALS' button	
SUCCESS SCENARIO	**Step**	**Action**
	1	System displays a list of goals that the user has currently set along with a brief overview of each
	2	User selects a goal to check on in more detail
	3	System displays more detailed reports (text, graphs) relating to particular goal and users progress towards it
ALTERNATIVE SCENARIO	**Step**	**Branching Action**
	4	User returns to step 2 to select a new goal to check on
RELATED INFORMATION		
Priority	High	
Performance Target	Once a goal is selected, reports for that goal should be formed and displayed within seconds	
Frequency	Dependant on user and nature of goal, generally low frequency	
Subordinate Use Cases	-	
Channel to Primary Actor	User interface	
Secondary Actors	System	

Channel to Secondary Actors	User Interface
OPEN ISSUES	The nature of the reports generated for each goal, diagrams, hard numbers, simple 'met'/'not met' message, etc.
SCHEDULE	Due date is version 1.0 release
AUTHOR	James Large, 26/01/2015

Create Group

USE CASE NAME	Create Group	
Goal in Context	Actor creates a group to share progress with other users	
Scope & Level	User management	
Preconditions	Actor is logged in	
Success End Condition	Group is created	
Failed End Condition	Group is not created	
Primary Actor	User	
Trigger	Actor presses the "create group" button	
SUCCESS SCENARIO	**Step**	**Action**
	1	**System displays page asking for details of group**
	2	**Actor enters a name for a group**
	3	**Actor sets privacy of group**
	4	**Actor clicks "Create group"**
	5	**Details of group are saved to system**
ALTERNATIVE SCENARIO	**Step**	**Branching Action**
	5a	**Actor fails to save group**
	6	**Actor relays error to user**
	7a	**Actor tries again**
	7b	**Actor contacts Admin**
RELATED INFORMATION		
Priority	Low	
Performance Target	-	
Frequency	Likely infrequent – investigate likeliness of users creating groups?	
Subordinate Use Cases	-	
Channel to Primary Actor	Java Script	

Secondary Actors	-
Channel to Secondary Actors	-
OPEN ISSUES	Yes
SCHEDULE	-
AUTHOR	Mercury Aimnh

Join Group

USE CASE NAME	Join Group	
Goal in Context	User adds self to currently existing group	
Scope & Level	User management/Group management	
Preconditions	User has an account, group exists	
Success End Condition	User is a member of the group	
Failed End Condition	User is not a member of the group	
Primary Actor	User	
Trigger	User clicks “join group”	
SUCCESS SCENARIO	**Step**	**Action**
	1	**User clicks “Join Group”**
	2	**Join request is sent to group owner**
	3	**Group owner accepts request**
	4	**User is notifed of acceptance, given link to group page**
ALTERNATIVE SCENARIO	**Step**	**Branching Action**
	3b	**Group owner rejects request**
	3c	**User is notified**
RELATED INFORMATION		
Priority	Yes	
Performance Target	100%	
Frequency	100%	
Subordinate Use Cases		
Channel to Primary Actor	JavaScript	

Secondary Actors	Group Owner
Channel to Secondary Actors	Automated E-mails
OPEN ISSUES	
SCHEDULE	
AUTHOR	Mercury Aimnh

Set Group Goal

USE CASE NAME	Set Group Goal	
Goal in Context	User creates a goal to be seen by all members of a group	
Scope & Level	User management/Group management/Goal management	
Preconditions	User is a member of a group, has permission to set goals	
Success End Condition	Goal is set for group members	
Failed End Condition	Goal is not set	
Primary Actor	User	
Trigger	User clicks “set goal”	
SUCCESS SCENARIO	**Step**	**Action**
	1	**User clicks “Set Goal”**
	2	**User enters details of goal**
	3	**Goal is set for group, or for members of group**
	4	**Members are notified**
ALTERNATIVE SCENARIO	**Step**	**Branching Action**
RELATED INFORMATION		
Priority	Yes	
Performance Target	100%	
Frequency	100%	
Subordinate Use Cases		
Channel to Primary Actor	JavaScript	

Secondary Actors	Group Owner
Channel to Secondary Actors	Automated E-mails
OPEN ISSUES	Group goal types, group goal handling
SCHEDULE	
AUTHOR	Mercury Aimnh

Initial Object Oriented Analysis Class Diagram

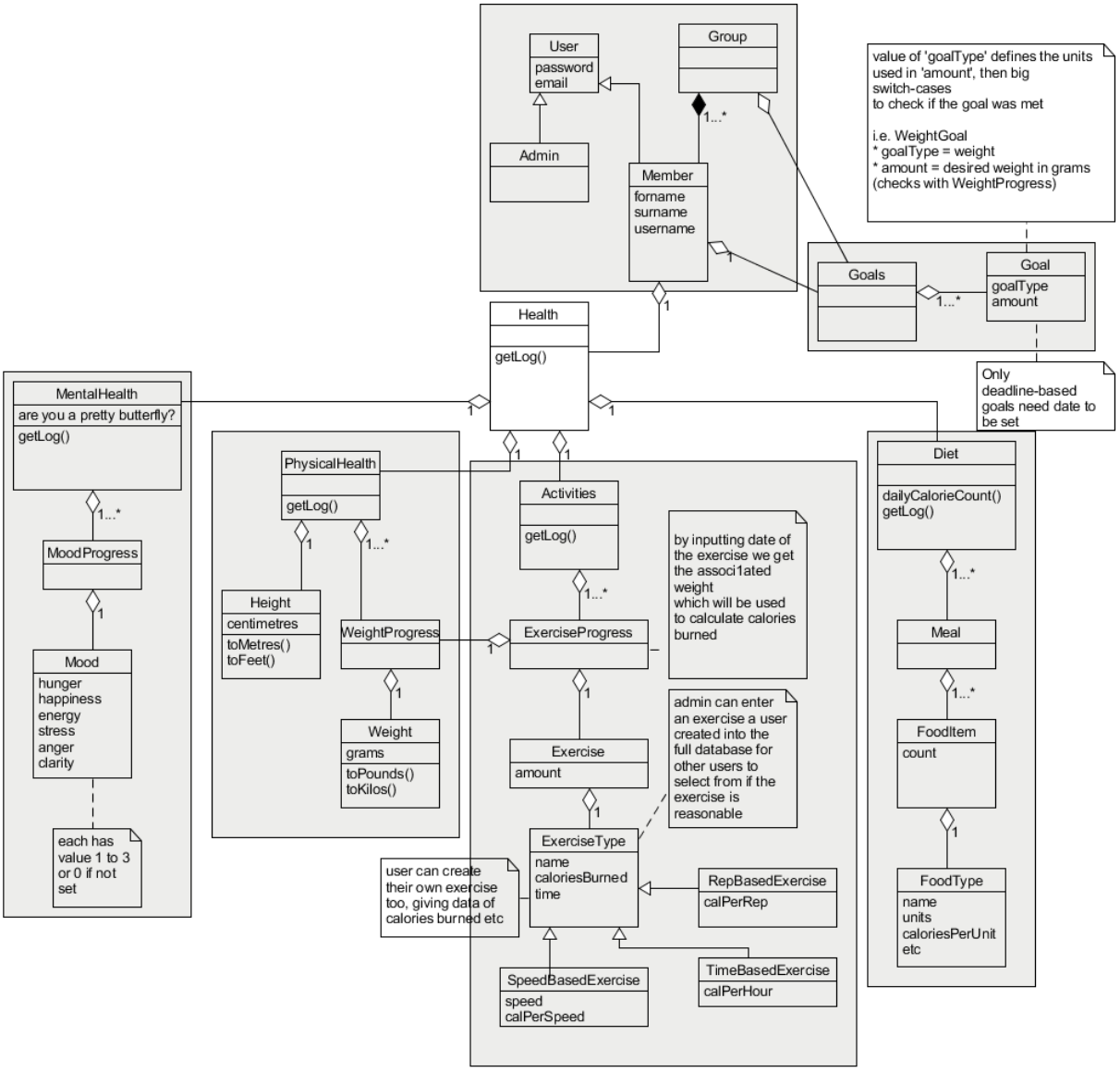


Figure 7: Initial Class Diagram

This diagram details our initial ideas for a model for the Health Tracking system. There are six general areas to be considered currently:

1. User area and groups
2. Goals
3. Mental Health
4. Physical Health
5. Activities
6. Diet

With a potential seventh being messages at a later date.

Archtecture Diagram

The diagram on the right of this page, shows the Model View Controller structure for the Health Tracker system. Where loggers for each of the types of the information's to be logged interact with the Database using a specialised Database Access controller, when receiving and responding to the requests from the client side.

Model - represents the entities found in the database.
View - creates web-pages and sends requests to the controllers
Controllers - contains controllers that link the view and the model.

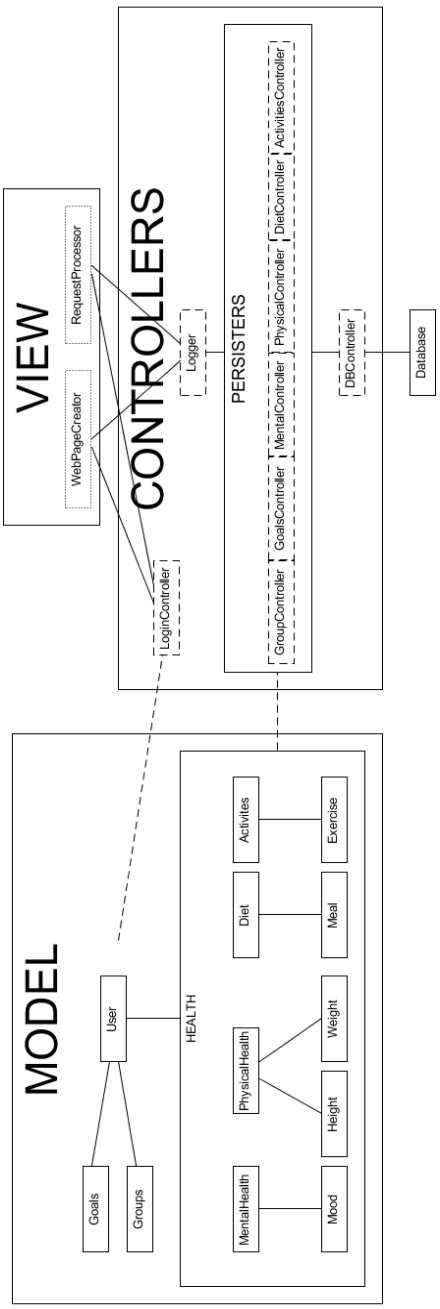
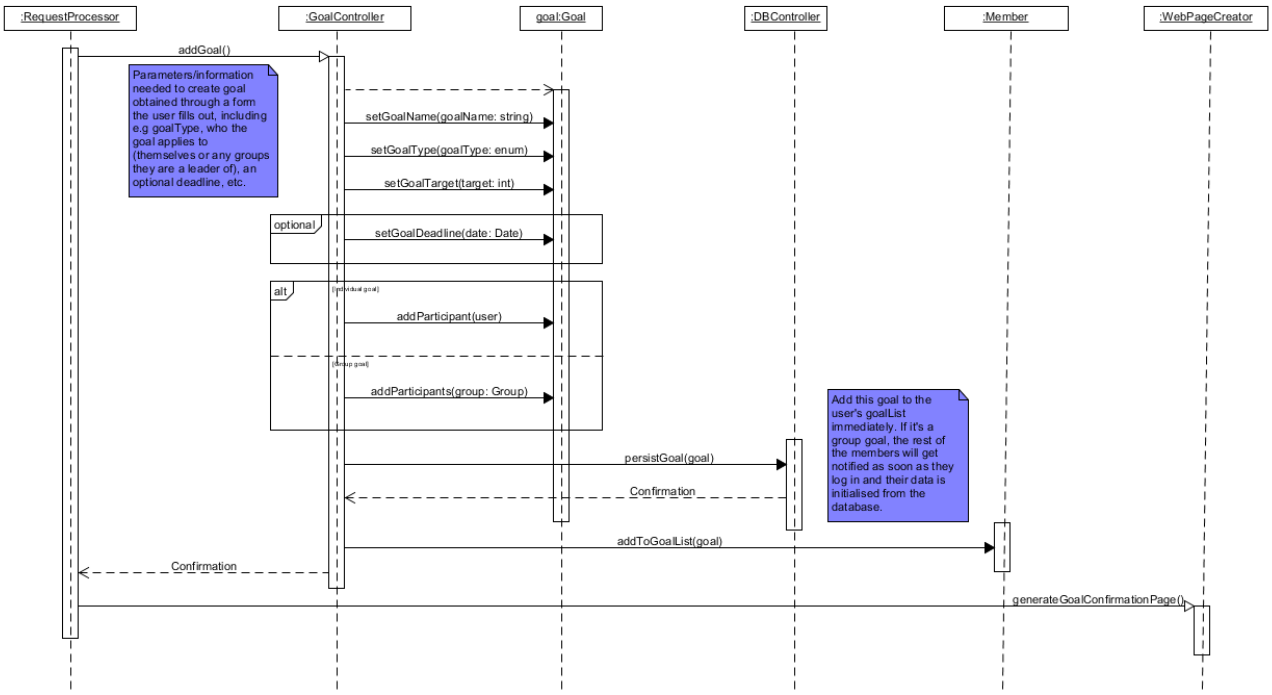


Figure 8: MVC Architecture Diagram

Sequence Diagram

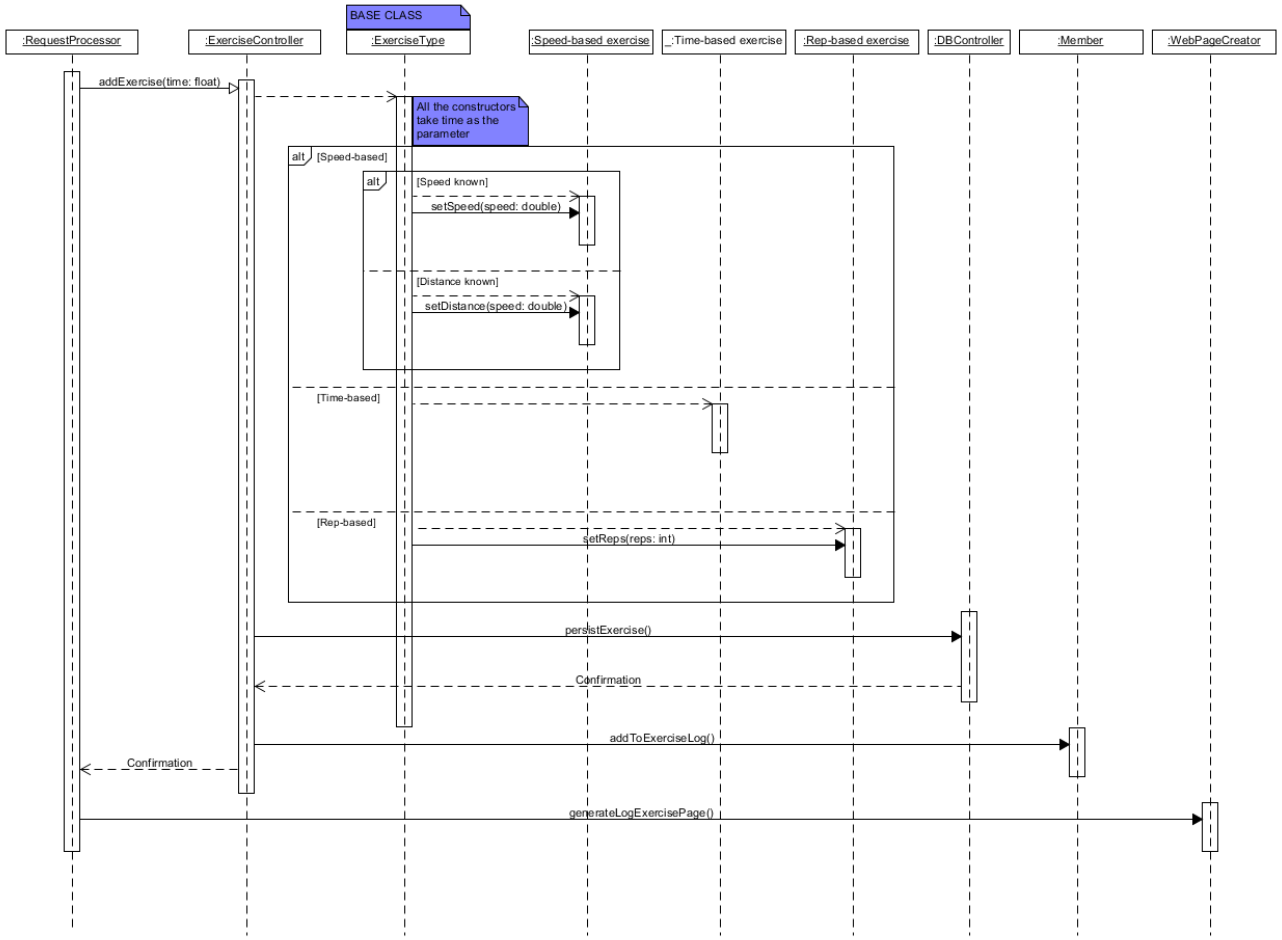
Goal Creation



The image above shows the interactions between Objects in the Health Tracker system, when trying to create a Goal.

To summarise the User will fill a form in the client application, and the RequestProcessor will pass that information to the GoalController, which will instantiate a Goal and will fill with the inputted data. The Goal will then be persisted to the DataBase via the DBController, a confirmation response will be sent to the GoalController who will then add the new Goal to the GoalList found in the Member Class. Finally a confirmation response will be sent to the RequestProcessor, which will call the method on the WebPageCreator to show confirmation to the user.

Logging Exercise

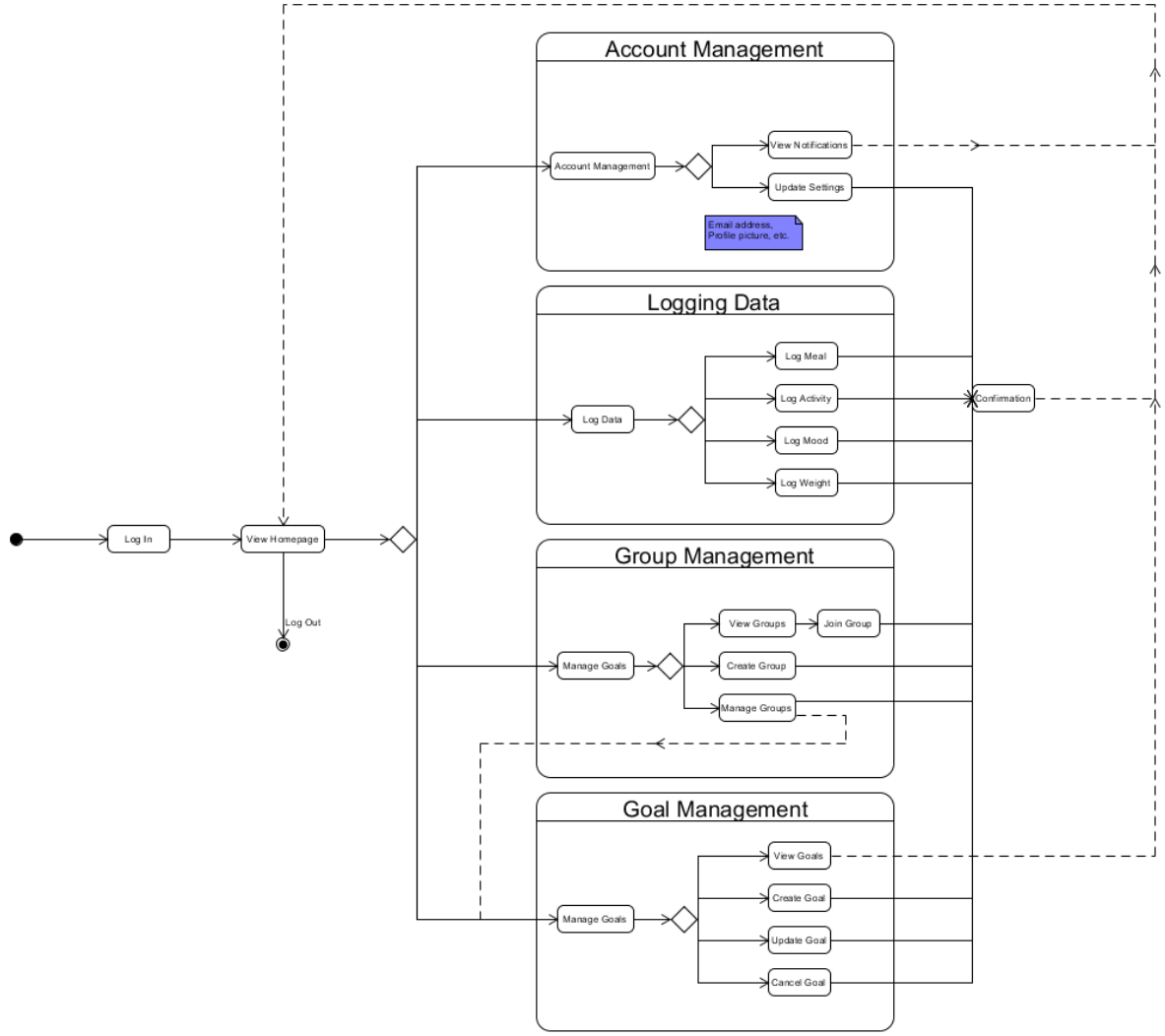


The image above shows the interactions between Objects in the Health Tracker system, when trying to log an Exercise.

To summarise the User will fill a form in the client application, and the RequestProcessor will pass that information to the ExerciseController, which will pass this to the ExerciseType class and check if the Exercise is a speed, time or rep based exercise type. After the type of the exercise has been found, the information is persisted to the Database via the DatabaseController, which then responds with a confirmation message. The ExerciseController then adds the Exercise to the Members list of Exercises done, and sends a Confirmation message to the RequestProcessor. Finally a confirmation response will be sent to the RequestProcessor, which will call the method on the WebPageCreator to show confirmation to the user.

State Charts

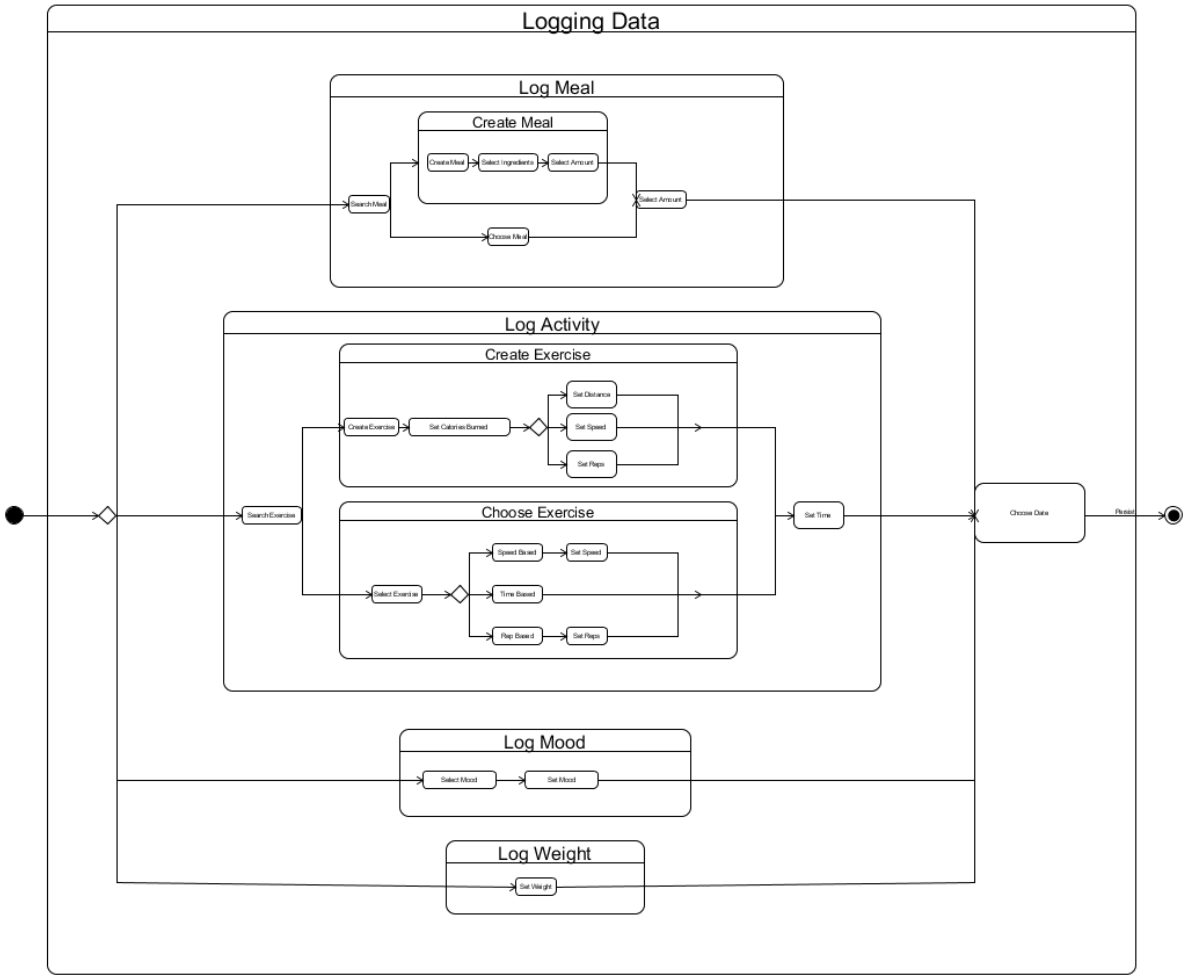
General Diagram



This state transition diagram represents the basic user interaction flow on the website.
After logging in the user is taken onto the homepage which works like a hub. From that point they can log out or go to one of the four sub-states.

Account Management sub-state lets the user view their notifications and update the settings such as email address, profile picture etc. Logging Data sub-state enables the user to log the particular piece of data (Meal, Activity, Mood, and Weight). This sub-state is explained in more detail in the second State Transition diagram. Group Management sub-state serves as a way to manage groups. The user can view the groups and then join one of them. They can also create groups. Managing groups links to goal management because a user can set up group goals. Goal Management provides a way to manage goals. The goals can be viewed, created, updated or cancelled. Every change and data input ends with the confirmation page and then the user gets redirected back to the hub (homepage).

Logging Data



If the User has decided to Log Data, he/she may choose from four options: Log Meal, Log Activity, Log Mood or Log Weight.

Within Log Meal, the User may choose to either Search for a Meal through the Database or choose to Create a new Meal, which lets the User select Ingredients that are used in the meal and the amount of each ingredient. The final part of Logging the Meal is for the user to input the amount of food he had in the meal.

Log Activity has a similar layout to Log Meal, User can decide to search for an exercise in the database or create his own exercise. If the User decided to create his own exercise, he/she must define how much calories it burns and what type of exercise it is, Distance, Speed or Rep based. If the User decides to search for a Exercise within the Database and states if it was Speed based, Time based or Rep based. For Speed based and Rep based, the User will need to define how quickly or how many reps they did. Finally the User will need to state for how long they did the exercise.

Log Mood will ask the User to select which mood to log, for instance happiness or Hunger, followed by a rating for that mood, which will be from 1 to 3.

Log Weight will require the user to set his current weight.

The final step is to add a date to the log, so that progress can be tracked.

Nouns

• **diet** • fitness regime • **goals** • **profile** • **user** • **user information** • physical details • basic information • **email communication component** • **email** • **social network platform** • lifestyle details • information • user name • real name • email address • format • personal information • **height** • **weight** • key factors • **feedback**	• **exercise** • duration • **workout routine** • **food** • **drink** • value • **meal** • ability • custom items • list • calorific count • meal type • **date** • **message** • **groups** • group name • email content • web link/code • member • **membership** • link • details • goal details

Verbs

• inform • add • define • record • creation • running • advise • configurable • share • join • select • distributed • meet • delete • accept • generate • using	• track • creation • comparison • capture • email communications • instant messaging • set • view • enter • prompted • validate • registration • proceed • collect • provide • selection

Glossary of Terms

- **activity** - an exercise a particular user performed on a particular date
- **admin** - a user with special permissions to, for example, remove inflammatory messages
- **biography** - a textual description of the item the bio is attached to, e.g a group or user
- **diet** - a history of a user's meals which have been logged in the system
- **exercise** - a type of physical activity which can be logged by a user
- **food item** - a particular type of food, with a number of calories per some unit defined by the type of food (e.g '1' apple, '100g' of rice)
- **goal** - a weight or timed activity target which may have a deadline date that the user works toward
- **group** - a collection of members who can share messages and set collective goals
- **group leader** - the creator of a group or someone promoted by another leader, who has special permissions within a group
- **log** - a list of past data sorted by date added on a particular topic, e.g a log of meals
- **meal** - a collection of food items eaten at a particular time
- **member** - a user able to log data, create goals and join groups, the target audience of the application
- **message** - textual data able to be sent from one user or group to another
- **profile** - a user's personal information and biography
- **private environment** - the section of a user's profile information which cannot be viewed by other users
- **public environment** - the section of a user's profile information which can be viewed by other users
- **user** - an entity which interacts with the system, either as an admin or member