Inf 43 – Spring Quarter, 2015 – Homework 1

Student Name: Jerry Anteater
Student Number:

Awarded Points	Maximum Points	Document Aspect
	15	Clarity of writing (spelling, grammar, sentence construction) and Clarity of expression (flow, structure, making logical arguments). Roughly 7.5 each.
	15	Introduction / Executive Summary (can be different sections or combined into one)
	7.5	Application Context / Environmental Constraints (can be different sections or combined into one)
	35	Functional Requirements, including use-case diagram and each use case (following a use-case template).
	7.5	Software Qualities and Non-functional Requirements
	5 (+5)	Other Requirements and Other Items. At least a Glossary of Terms. You can earn up to 5 points Extra Credit if you go beyond Glossary
	7.5	Assumptions / Risks (can be different sections or combined into one)
	7.5	Priorities / Implementation Phases; Future Directions and Expected Changes
	100	**TOTAL**

ZotMyHealth System Requirements

April 28, 2015

Jerry Anteater

Introduction

ZotMyHealth is a health app envisioned by Health Beacon Inc. that will act as a means of centralizing each individual's data from every health app on the market today. This app's mission is to avoid the unnecessary hassle of having to log on to multiple health apps, just to view information from a small subset of what attributes translate into one's well-being. Health is not measured by how many steps one takes or how much sleep one gets, but a collection of all these variables, specialized to each unique individual. ZotMyHealth will not only have the ability to accumulate and interpret this data, but also have the potential to improve the lives of its user.

ZotMyHealth is being developed by Jerry Is The Best, LLC. This document will describe the underlying requirements and overall process needed to design and engineer the app.

Table of Contents	
Overview / Executive Summary	A brief summary of the project, describes major features and overall goals of the application.
Application Context / Environmental Constraints	Details information on the platforms which the application will be compatible with, lays out constraints for various components of the project.
Functional Requirements	Goes into greater detail of how the application will work, identifies key steps in the design process.
Software Qualities and Non-functional Requirements	Describes requirements pertaining to the design of the system's implementation, in terms of measurable classifications such as performance or security.
Other Requirements	Miscellaneous requirements not described in the above sections, as well as a Glossary of Terms.
Assumptions / Risks	Various descriptions of assumptions and possible risks regarding each facet of building the app.
Priorities / Implementation Phases	Outlines the different priority levels of certain features as requested by the client.
Future Directions and Expected Changes	Grants further insight on the future of the app and possible changes proposed by the client.

Overview / Executive Summary

ZotMyHealth functions by extracting all health-related data from the user's pre-existing third party health apps and devices. This means that from this one app, the user can employ once separated functionalities into a single, continuous process. The user will be able to see how many calories are in a burger from his/her favorite calorie tracking app, then see what exercises s/he has completed from an exercise tracking app. Keep in mind that ZotMyHealth will not actually replace any one, individual health tracker, but will act as an interpreter by extracting, understanding, and synchronizing all the health related data into one, centralized location. In terms of the need for this software, the user will never again have to muddle through multiple interfaces, just to decipher the already complicated nuances of his/her health. It will be one app, for all things health.

The three main health mechanisms that will be featured through this app are the influences revolving around workouts, calorie intake, and sleep cycles, with other compatible factors/data sources implemented in the future. The data describing these health aspects can be imported from pre-existing third party apps, such as MyFitnessPal, and health devices, such as Fitbit. The user will also be provided the option to manually input data, and the app will have the ability to calculate accurate estimates of various quantifiable data given the user's queries. ZotMyHealth can then automatically sync this data to secure remote databases when necessary, upon completing a workout for example. This data is then analyzed and monitored through the app via interactive charts and specialized goals. These goals are customized by the user, ultimately serving as small milestones in the journey of improving overall health and wellness.

Application Context / Environmental Constraints

The mobile app will be available on Android, iOS, and Windows (phone) devices and must be compatible with all versions of each respective operating system. This means that the app will be available on the Google Play Store, The App Store, and the Windows Store, and must run on both phones and tablets. The web portal follows similar compatibility requirements, as it should run on all versions of all popular internet browsers, including Firefox, Chrome, Safari, and Internet Explorer. The web portal should be available while browsing the web from either a computer or a mobile device.

The user interface (UI) must be clean, concise, and intuitive for both the mobile app and the web portal. The UI must be relatively simple and data should be displayed in a non-overwhelming manner. The interface must also scale correctly between different devices, a problem that will likely exist between different sized phones and the differing sizes between a phone and tablet.

There are no software constraints in terms of what programming languages or libraries to use, however it is preferred that common languages are used to simplify the development process.

Functional Requirements

This section will describe how each user will interact with the app and web portal. Note that the handling of both the app and web portal will be functionally similar, however not all features will be available between each platform. In the case where both platforms support a given use case, assume that the term "system" is synonymous with each of the platforms. It is also assumed that ZotMyHealth is always connected to the internet and the user is currently logged in, unless otherwise stated.

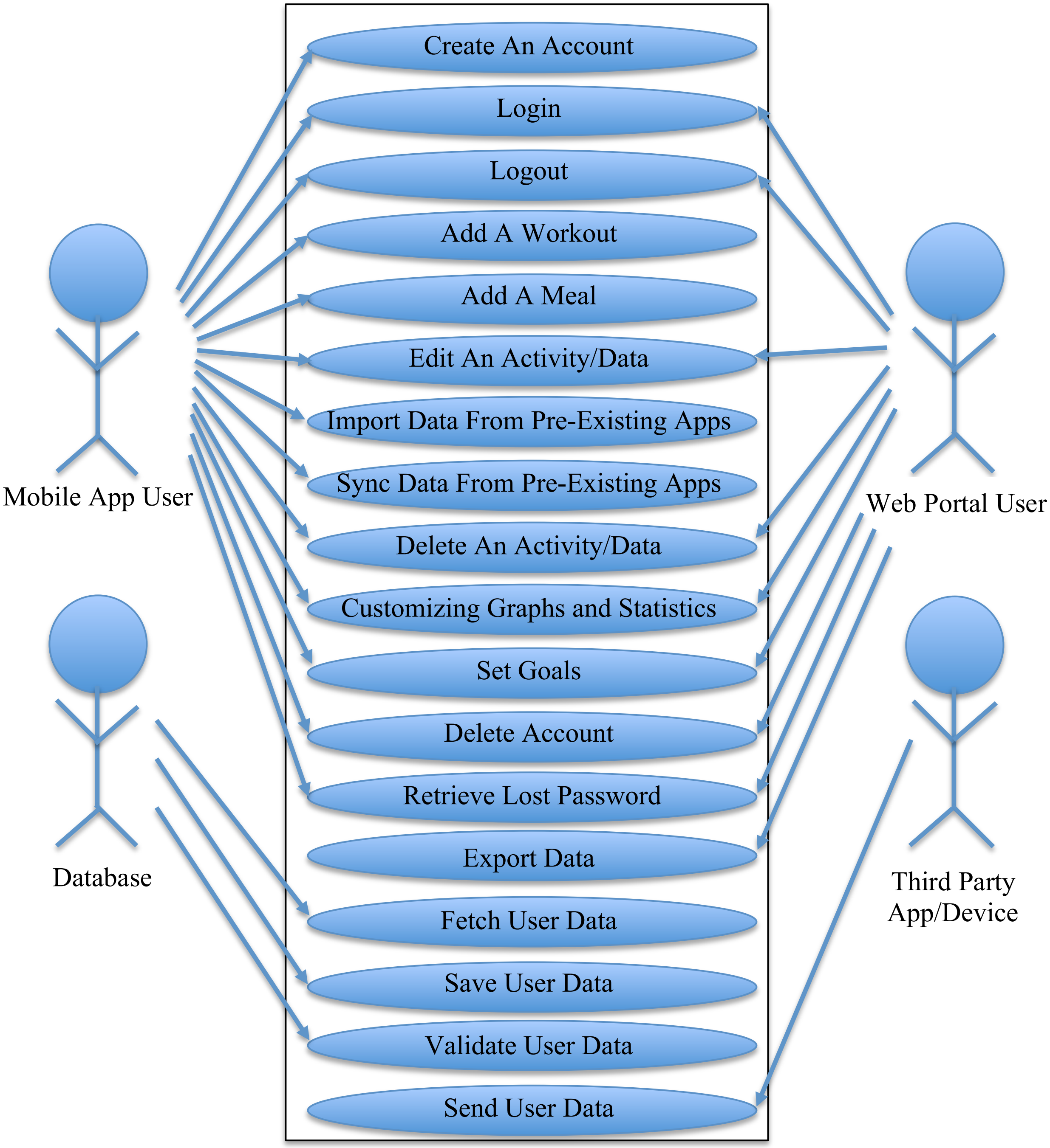

Figure 1: Use Case Diagram for ZotMyHealth

Mobile App and Web Portal Specific Use Cases

Login

Handle inputting login information, account verification, and updating of user data.

- Preconditions: User is currently logged out.

Basic Flow of Events:

1. User clicks the link to initiate the login process.
2. System prompts the user for username/email and password combination.
3. User fills in login credentials.
4. System validates login credentials.
5. System fetches the user's data from database and updates where appropriate.
6. System redirects to home screen.

Exception Flow:

5a. System detects the user entered invalid username/email and password combination.

  1. System displays appropriate error message.
  2. System clears password field.
  3. Return to basic flow at step 2.

5b. System detects the user's account is not verified (via verification email).

  1. System displays appropriate error message.
  2. System requests confirmation to resend verification email.
  3. System redirects to login screen.

Logout

Handle logout process.

Basic Flow of Events:

1. User clicks the link to initiate logout process.
2. System requests confirmation to logout.
3. System logs user out.

Alternative Flow:

3a. User indicates to not logout.

  1. System redirects to home screen.

Edit An Activity/Data

For each type of activity (workout, calorie intake, sleep cycle), allow the user to edit specific property(s) of the data.

Basic Flow of Events:

1. User clicks the link to initiate editing an activity.
2. System prompts the user with a list of all previous activities stored within the user's database.
3. User selects the activity to edit.
4. System prompts user with editable fields of every data property of the activity.
5. User edits fields as necessary.
6. User clicks the link to save changes to the activity.
7. System saves changes to Database.
8. System displays success message.

Alternative Flow:

3a. User indicates to stop editing.
    1. System redirects to home screen.

6a. User cancels changes to the activity.
    1. System redirects to home screen.

Delete An Activity/Data

For each type of activity (workout, calorie intake, sleep cycle), allow the user to delete specific properties of the data, given type of data and.

Basic Flow of Events:

1. User clicks the link to delete data.
2. System prompts the user with a list of all previous activities stored within the user's database and optional time span.
3. System requests confirmation to delete activity based on the user's query.
4. System displays success message.

Alternative Flow:

4a. User indicates to not delete activity.
    1. System returns to home screen.

Customizing Graphs and Statistics

For each type of activity (workout, calorie intake, sleep cycle), display various graphs and statistics to the user.

Basic Flow of Events:

1. User clicks the link to initiate viewing graphs and statistics.
2. System displays (default) graphs of workout, calories intake, and sleep cycle, as well as various statistics describing these categories.
3. User clicks the link to initiate customizing graph and statistics.
4. System prompts the user for multiple options, such as time span, specific workout, etc. Real-time autofill suggestions for queries, where applicable.
5. System fetches information from database based on user's query.
6. System displays graphs based on user's query.

Set Goals

Allows the user to set, maintain, or update quantifiable health goals.

Basic Flow of Events:

1. User clicks the link to initiate setting goals.
2. System prompts the user for type of goal and numerical value describing goal.
3. System prompts the user with notification options, such as reminders, progress completed, etc.
4. System requests confirmation to set goal.
5. System displays success message.

Alternative Flow:

5a. User indicates to cancel setting goals.

1. System redirects to home screen.

Delete Account

Handle deleting the user's account, including all data corresponding with the user.

Basic Flow of Events:

1. User clicks the link to initiate deleting an account.
2. System requests confirmation to delete account.
3. System deletes all user data from database.

Alternative Flow:

2a. User indicates to not delete account.

1. System redirects to home screen.

Retrieve Lost Password

Handle user retrieving lost password.

- Preconditions: User is currently logged out.

Basic Flow of Events:

1. User clicks the link to initiate changing password.
2. System prompts the user to enter username or email associated with account.
3. User fills in username or email.
4. System validates username/email in database.
5. System sends email with change password link.
6. User clicks change password link in email.
7. System confirms change password link.
8. System prompts the user for new password.
9. User fills in new password.
10. System changes associated user's password with new password in database.
11. System logs the user in.
12. System redirects to home screen.

Exception Flow:

4a. User entered invalid username/email (not in database).

1. System displays appropriate error message.
2. Return to basic flow at step 2.

Mobile App Specific Use Cases

Create An Account

Handle creating a new user account.

- Preconditions: User is currently logged out.

Basic Flow of Events:

1. User clicks the link to initiate the account creation process.
2. System prompts the user to enter his/her username, email, password, name, and gender. Age, height, and weight are optional fields.
3. User fills in fields.
4. System validates user input.
5. System confirms account has been created, verification link sent to account's email.
6. User opens link in verification email.
7. System verifies account.
8. System redirects to home screen.

Exception Flow:

5a. Non-optional fields are empty.
   1. System displays appropriate error message.
   2. Return to basic flow to step 2.

5b. Username/email is found in database.
   1. System displays appropriate error message.
   2. Return to basic flow to step 2.

Add A Workout

Allows the user to add a new workout. NOTE: this describes the process of manually inputting a workout, and not syncing with a different app.

Basic Flow of Events:

1. User clicks the link to initiate adding a workout process.
2. System prompts the user for workout name (required), time, calories burned, reps, weights, and distance. Real-time autofill suggestions for queries, where applicable.
3. System auto-propagates data into the empty fields, where applicable.
4. User clicks the link to confirm adding the workout.
5. System saves workout data to database.

Alternative Flow:

2a. Workout is not associated (synced) with pre-existing health app.
   1. System prompts the user to associate workout with pre-existing health app (optional).

4a. User indicates to not add the workout.
   1. System redirects to home screen.

Add A Meal

Allows the user to add a new meal. NOTE: this describes the process of manually inputting a meal, and not syncing with a different app.

Basic Flow of Events:

1. User clicks the link to initiate adding a meal process.
2. System prompts the user for name of meal. Real-time autofill suggestions for queries, where applicable.
3. User clicks the link to confirm the meal.
4. System saves meal data to database.

Alternative Flow:

2a. System does not recognize name of meal.

1. System prompts user for name of meal, list of ingredients of the meal, and/or number of calories in the meal/ingredients. Real-time autofill suggestions for queries, where applicable.
2. System saves new meal info into database.

Import Data From Pre-Existing Apps

Handle process of importing health data from pre-existing third party health apps.

- Preconditions: User has data from pre-existing health apps.

Basic Flow of Events:

1. User clicks the link to initiate the import process.
2. System prompts the user for the type of data to import and which health app to import the data from.
3. System saves imported data to database.
4. System prompts the user to remember choice for syncing data of this type with the pre-existing app in future (optional).
5. System saves user choice in database.

Exception Flow:

3a. Pre-existing (conflicting) data is found in the database, within the same time interval of the newly imported data.

1. System prompts user to delete existing (old) data and cancels importing new data.

Sync Data From Pre-Existing Apps

Handles process of syncing health data from pre-existing third party health apps.

- Preconditions: User has data from pre-existing health apps.

Basic Flow of Events:

1. User clicks the link to initiate the syncing process.
2. System prompts the user for the type of data to sync.
3. System prompts the user for which health app to sync the data from.
4. System saves choice to database.

Web Portal Specific Use Cases

Export Data

Handles process of exporting data. Expected format file type .xlsx, .xls, or .csv.

Basic Flow of Events:

1. User clicks the link to initiate the export process.
2. System prompts the user for what types of data to export, what format to write the data in.
3. System fetches user queries of data from database.
4. System writes data into proper format.
5. User downloads exported data in selected format.

Database Specific Use Cases

Fetch User Data

Handles process of fetching user data from database. This is used whenever the system requires the use of user data, such as displaying graphs or editing data.

Basic Flow of Events:

1. System requests the user's data from database.
2. Database fetches the user's data.
3. System retrieves the user's data.

Save User Data

Handles process of saving user data to database. For example, when the user adds a new workout, the data is saved in the database.

Basic Flow of Events:

1. System sends the user's data to database for saving.
2. Database retrieves the user's data.
3. Database writes/overwrites the user's data.

Validate User Data

Handles process of validating user data from database. This is used in processes such as validating login information.

Basic Flow of Events:

1. System sends the user's data for validation.
2. Database retrieves the user's data, checks validation.
3. Database sends validation (true/false) to system.

Third Party App/Device Specific Use Cases

Send User Data

Handles process when the system requests the user's health data from third party apps or devices.

Basic Flow of Events:

1. System requests the user's data from third party app/device.
2. Third Party App/Device receives request.
3. Third Party App/Device sends the requested user's data to system.
4. System receives the requested user's data.

Exception Flow:

1a. Third Party App/Device does not receive request.

1. System displays appropriate error message.

Software Qualities and Non-functional Requirements

Performance

The app must be efficient as possible, while consuming as little battery as possible, per the client's request. Expected target for average daily battery per moderate usage of the app is two percent.

Scalability

With heavy reliance on remote database calls, the database is expected to scale appropriately as the number of users increases (or decreases). All code must be written in a way that ensures code reuse to aid in the process of updating and bug fixes. Overall app maintenance must also be held in timely manner to ensure maximum user satisfaction with the app.

Availability

The app, specifically the calls to the remote database, must be made available for use always. The ability to load data from other health tracking apps must also be available, assuming the third party app is functioning as expected. User support must also be available, or at least handled, in a timely fashion.

Security

All data in the database must be secured with some sort of basic encryption. A user/password login system must be implemented to prevent any misuse or stolen data. Although there are no current plans of micro-transactions and the need for the user's credit card information, a strong foundation of security measures will prove to be useful later on.

Privacy

This app requires the agreement from other health tracking apps and devices to disclose user's data. This agreement must be honored and clearly disclosed. User data should not be shared without the consent of the user. Further details on the issue of privacy are discussed in the section "Assumptions / Risks".

Reliability

Health data calculated and interpreted must be error-free. Failure to ensure this fact can cause a negative impact on the user's health. In the event of a crash, data must preserved, most likely in the form of secure backup servers.

Portability

The app must be assessable from any standard Android, iOS, or Windows phone or tablet, or any standard computer, all equipped with an internet connection.

Other Requirements

There is one small detail that was not covered in the previous sections, regarding the use of the app with multiple devices. Upon using installing the app on a new device, if the user already has an account and logs in, all the data from his/her account will automatically sync with the user's new device, including all syncing preferences. If the new device does not have an exact third party health tracker app or device installed, and that app is included in one of the syncing preferences, the preference will be deleted, but only to that specific device.

One last detail described by the client regards sleep cycle data. The user should only be given the option to import or sync sleep cycle data. This means no manual input.

Other features deemed less important by the client (for the current state of this project) include the need for a tutorial for new users and support for languages other than English. Features that were deemed not necessary by the client, however may be applicable in future, include the collecting of user analytics and mobile widgets.

Glossary of Terms	
Activity	Describes a workout, calorie intake, or sleep cycle activity.
Availability	The ability of the system to be available.
Code Reuse	The ability to reuse code, instead of starting from scratch.
Interface/User Interface (UI)	How a certain page will be displayed to the user, including user interactions.
Operating System (OS)	The software that manages a certain device. For mobile devices, common OS's include Android, iOS, and Windows. For computers, common OS's include Windows, Macs, and Linux.
Platform	Device used to access the app, such as mobile phones, tablets, or computers.
Portability	Usability of software between different platforms.
Preconditions	Conditions that must be true before a given action can take place.
Scalability	The ability for the system to grow or shrink with the user base.
User Analytics	The collection and analyzing of user data for use in research.
Web Portal	A web-based application.
Widget	A small add-on application with limited functionalities, however is simply a means of commutating quick bits of information to the user.

Assumptions / Risks

A key assumption when using the app will be the availability of internet. It is assumed that the user will always be connected to the internet, whether it be via Wi-Fi or the phone carrier's data. ZotMyHealth will not be able to add/edit activities without it. This means that there is the chance of losing a small subset of users who rely on a device that is not always connected to the internet.

It is also assumed that the user already uses other health tracking apps and/or devices. Without these apps, the experience of ZotMyHealth may not be as desired. For example, instead of the automated syncing of workouts, the user will have to manually input the data and that can end up being tedious and less enjoyable.

The biggest risk involves the issue of privacy. Because of the multiple connections between different health tracking apps, it must be made certain that these connections only receive data and never send. In other words, all user data from ZotMyHealth must not be shared between any other apps, unless permission is granted by the user. This also means that if a certain health tracking app decides to terminate the agreement of sharing its data with ZotMyHealth, it is the job of the app to ensure no new data is ever used from that third party app again. Failure to abide to this request will likely lead to legal issues and eventual loss of users.

Priorities / Implementation Phases

The client has specified the following timeline for implementation priorities. It must be noted that some features in components of completed phases may still need to added, where appropriate.

Must Have (First) (Mobile App Only)

- Workout and calorie intake tracker support, including manual input of data, displaying data to the user, and saving to database, excluding importing and syncing of data.
- User preferences and settings, including creation of account, login system, setting goals, and editing data.

Should Have (Second) (Mobile App Only)

- Implementation of the connection to pre-existing third party health apps, including importing and syncing data.
- Sleep tracker support, including manual input of data, displaying data to the user, saving to database, and importing and syncing of data.

Nice to Have (Third):

- Web portal, which will include all appropriate facets as stated above.
- Windows Phone app, which will include all appropriate facets as stated above.

Future Directions and Expected Changes

The biggest change that will be expected in the future revolves around the monetization of this application. The client would like two versions of the app, the first version being free, but containing ads. Various advertisement methods will be interlaced within the app itself, whether it be static banner ads or video. The second version would be labeled as “premium,” having a one-time cost, but be ad-free. Many apps include this similar model, proving to be a good way for a user to try out the free app, and when satisfied, purchase the premium version. It is yet to be determined if the premium app will include any additional features or functionalities.

Another major feature planned will be the integration of social media, such as Facebook or Twitter. Users should be able to share their workouts with a simple click of a button. This will be a good way for users to communicate their personal health achievements with their friends and family, and also a great way to advertise the app to broader range of audiences.

Furthermore, the client insisted that there be no local database/storing of data within the device itself. This constraint must be evaluated in the future, since with this design, the app will not function without internet access.

Lastly, the integration and featuring of more types of health data should be added. In its current state, the three health mechanisms in workouts, calorie intake, and sleep cycle (albeit popular) may not be enough to entice the user who is looking for more. Giving the user more options will not only diversify the demographic of users, but improve the health of all users, which in turn, allow ZotMyHealth to gain further popularity and success.