Software Requirements Specification

FH Mobile Application

Version 2.0

Prepared by

 Omar Rivera
Andrew Poirier
Daven Amin
Rick Rejeleene

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Brian Strattard	February 4, 2014	Initial Draft	1.0
Andrew Poirier	February 11,2014	Draft Reconstruction	1.1
Omar X Rivera	February 12,2014	Updated Acronyms, Management Issues, Performance requirements, Review Sections, System Evolution	1.2
Daven Amin	February 13,2014	Added missing use case, added additional functional requirements, edited for improved clarity and consistency, updated definitions.	1.3
Team	March 18, 2014	Incorporated feedback from both group and professor.	2.0

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for creating Fitness Health, also referred to 'FH." This Fitness Software will consist of a downloadable mobile application for an Android operating system. This document will outline all of the necessary information to start development.

1.2 Intended Audience and Reading Suggestions

The intended audience for this document is CSC 505 course instructor and class peers. Throughout the rest of this document, the project will be broken up into sections for: Project Description, System Features, External Interface Requirements, and Non Functional Requirements. There is also a glossary of common terms found throughout the document.

1.3 Project Scope

The main purpose of this project is to create a mobile device application to make an end user's fitness goals attainable by providing financial motivation. The goal is to make it as easy as possible to setup profile, workout schedule, and fitness goals using the application. The profile shall be easy to view, as will the schedule and goals using the application. The end user shall be able to enter financial information and designate a penalty for missing scheduled workouts. The end user shall be able to designate a 'fitness coach' to engage the penalty for missing scheduled workouts.

The software being used for development is the Android SDK. Additionally, back end support services are managed by a server running Linux Debian 7.3 and MySQL server 5.5.35.

1.4 Definitions, Acronyms, and Abbreviations

Term	Definition
DFD	Data Flow Diagram
SDK	Software Developer Kit
GUI	Graphical User Interface
XML	Extensible Markup Language
HTML	Hyper Text Markup Language
HTTPS	Hyper Text Trans protocol secure
SSH	Secure Shell – Cryptographic Protocol
UML	Unified Modeling Language
SRS	Software Requirements Specification
FH	Fitness Health Application
Authenticator	A user account granted *authenticate* privileges on one or more *workout schedules* associated with another user account. Also referred to as a *Trainer*.
Authenticate	A user action confirming to the system that another user successfully completed a *workout metric* associated with one of their *workout*

	schedules.
Payment Information	The minimum required information to debit an external financial system. This could be a credit card number (along with cardholder name, expiration date, and security code), or login information to an external system, such as Amazon Payments.
Workout Metric	A numeric value and a textual description, representing a quantified exercise activity to occur over a week, i.e. "(30) sit-ups", "(3.5) walk/run miles", "(1) hour of swimming"
Workout Schedule	A start date, a number of weeks specifying duration, and a *workout metric* which collectively define the total length and quantity of an exercise regimen.

1.5 References

1. Developers, Android. "What is android." http://developer.android.com/guide/basics/what-is-android.html 02 Feb 2014.

2. Developers, Android. "Android Tutorial" http://www.tutorialspoint.com/android/index.htm 02 Feb 2014.

3. Creating Commons Atribution."Building Your First App" https://developer.android.com/training/basics/firstapp/index.html?hl=cn. 28 Jan 2014.

4. Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass. "Beginning PHP5, Apache, and MySQL Web Development". Wrox (Feb/2005)

2. Overall Description

2.1 Product Perspective

2.2 Product Functions

This program will allow users to be able to create a profile, enter financial information and set penalty amount, create workout schedules, track and review schedules, and designate their penalty mechanism from their mobile phone. Any phone that supports Android 4.3.1 or greater will be able to install the applications and run the application from their phone.

2.3 Operating Environment

The software will run on the Android operating system version 4.3.1 or higher. All devices that support this version of the Android operating system and possess an internet connection will be able to run the application. The application is developed with the Android SDK and SSH client respectively.

2.4 Design and Implementation Constraints

The software must run on the Android operating system. The mobile phone has existing hardware/software constraints. The supporting database used by the software shall be implemented using MySQL. The software must be developed in a language supported by the Android SDK. (Java,XML)

2.5 Assumptions and Dependencies

The system is dependent upon the server configuration. As part of the integration and testing the testers will download the app to the phone with a USB port; if the application is released to the public, the customers will have to download the application from Google Play Store.
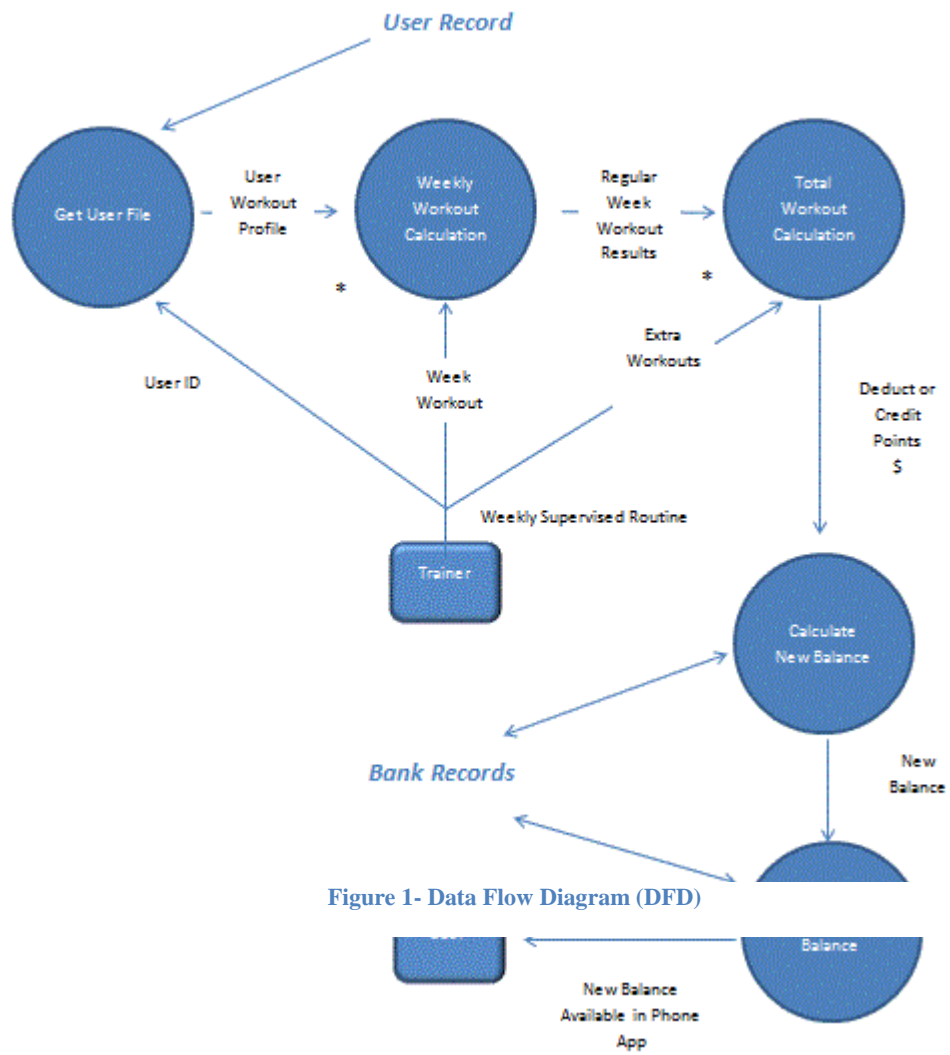
2.6 System Models

2.6.1 DFD

Figure 1- Data Flow Diagram (DFD)

3. Functional Requirements

This section outlines some of the system use cases.

3.1 Use Cases

3.1.1 Use Case: User Creates Account

3.1.1.1 Brief Description
The user begins the application, specifies that they wish to create a new account, and provides the necessary information to the system.

3.1.1.2 Step by Step Description
(Before this use case can be initiated, the system must have been started)
1. The user selects the option to create a new account.
2. The user inputs their full name, a login name, and a password.
3. The system prompts the user to read and accept the terms of service.
4. The user chooses to accept the terms of service.
5. The system creates an account with the provided user information, and informs the user.

3.1.1.3 Exceptions Scenarios
Exception Scenarios:
- 2. The user input login name conflicts with an existing account in the system.
  - o The system prompts the user to select a different login name.
- 3. The user declines the terms of service.
  - o The system does not create an account with the provided user information, and informs the user of this.

3.1.2 Use Case: User logs into System

3.1.2.1 Brief Description
The user provides login credentials to the system and is logged in.

3.1.2.2 Step by Step Description
(Before this use case can be initiated, the system must have been started)
1. The user selects the option to log into an existing account.
2. The user inputs their login name and password.
3. The system validates the credentials and logs in the user.

3.1.2.3 Exception Scenarios
- 3. The system cannot validate the user's credentials.

o   System informs user and does not log them into the system.

3.1.3 Use Case: User inputs new Workout Schedule

3.1.3.1 Brief Description

The user provides a workout schedule and a payment method for the schedule.

3.1.3.2 Step by Step Description

(Before this use case can be initiated, the system must have been started and the user must have logged in)

1. The user selects the option to create a new workout schedule.
2. The user inputs the type of workout metric ("time", "distance", "repetitions", etc.)
3. The user how many times a week the workout will take place.
4. The user specifies the number of weeks of the workout schedule.
5. The user specifies the login name of another user who will provide verification that workout metrics have been met.
6. The user provides a payment method (i.e. credit card number) for the system to use if workouts are not completed.
7. The user confirms the workout schedule information and payment information are correct.
8. The system validates and accepts the completed workout schedule information, stores it, and informs the user.

3.1.3.3 Exceptions Scenarios

- 2. The user specifies an unacceptable value for the workout metric (negative number, alphabetical character, decimal number for non-decimal metric, etc.)
  - o   The system informs the user of the error and prompts for valid input.
- 3. The user specifies an unacceptable value for the workout frequency (negative number, alphabetical character, decimal number, etc.)
  - o   The system informs the user of the error and prompts for valid input.
- 4. The user specifies an unacceptable value for the number of weeks (negative number, alphabetical character, decimal number, etc.)
  - o   The system informs the user of the error and prompts for valid input.
- 5. The user does not specify an existing user name of another user to verify workout metrics have been met.
  - o   The system informs the user of the error and prompts for a valid user name.
- 6. The user specifies invalid payment information (incorrect number of digits in credit card information, etc.)
  - o   The system informs the user of the error and prompts for valid input.

3.1.4 Use Case: User validates Workout

3.1.4.1 *Brief Description*

The user gives the system acknowledgement that another user has completed one of their workout metrics.

3.1.4.2 *Step by Step Description*

(Before this use case can be initiated, the system must have been started and the user must have logged in)

1. The user selects the option to validate another user's workout.
2. The user selects the user name of the other user.
3. The user selects the workouts which the other user has completed.
4. The system validates and accepts the selected list of workouts and informs user.

3.1.4.3 *Exceptions Scenarios*

- 1. The user has not been authorized to validate any other users' workouts.
  - ○ The system does not provide the option to validate another user's workouts.

3.2 Requirements

3.2.1 User Platform Requirements

- The app shall run on the Android 4.3.1 platform

3.2.2 Account Requirements

- Allow the user to log into the service with account credentials.

- Allow the user to create a new account in the service.

- Allow the user to delete their account in the service.

- Persist user account information when user is not logged into service.

- Update user account information when user is logged into service.

3.2.3 Workout Schedule Requirements

- Allow the user to create one or more *workout schedules* composed of a *workout metric*, a frequency per week, and a total number of weeks.
- Allow the user to specify a *workout metric* as a number (an amount of time, a distance, or a number of repetitions) and a textual description.
- Allow the user to associate a *payment method* with a *workout schedule*.
- Allow the user to provide a user name to *authenticate* that *workout metrics* have been met for a particular *workout schedule*.

- Workout Schedule information shall include a *workout metric*.

- Workout Schedule information shall include a start date and a duration in number of weeks.

3.2.4 **Validation Requirements**

- Allow a user with *authenticator* privileges to view *workout schedules* which they have been authorized to *authenticate* that *workout metrics* have been met.

- Allow a user with *authenticator* privileges to mark *workout metrics* in authorized *workout schedules* as *accomplished*.

3.2.5 **Server Platform Requirements**

- Server shall run on an internet-connected Linux platform.

3.2.6 **Server Requirements**

- Server shall allow for the creation, storage, and retrieval of user account information.

- Server shall allow for the creation, storage, and retrieval of *workout schedule* information associated with a specific user account.

- Server shall allow for the creation, storage, and retrieval of *workout metric* information associated with a specific user account and *workout schedule*.

- Server shall allow for many-to-one *authenticator* associations of one or more user accounts with a single user account.

- Server shall allow for debiting a financial account using *workout schedule* associated *payment information*.

- Server shall store every login event in a login table within the database for security purposes and maintenance.

3.2.7 **User Account Requirements**

- User Account information shall include user-provided full name.

- User Account information shall include a login name.

- User Account information shall include a login password.

- User Account information shall include account balance information.

4. External Interface Requirements

4.1 User Interface

UI-1:   All users shall be able to access the mobile application using a GUI provided by the application.

UI-2:   User Interface shall use secure connection to communicate with the software.

UI-3:   User Interface shall allow authorizing individuals to apply penalty or credit associated with the user accounts.

4.2 Hardware Interfaces

This will be an Android phone application, and as such, will be designed to interface with the hardware present on the mobiles phones that run the Android operating system. In theory, the application will be able to run by other devices that can emulate Android, but this will not be a consideration during design.

As this is a mobile device, it will be using a cellular network or WiFi to connect to the Internet, which will allow it to communicate with the database servers. This means that it will be using the wireless communication antennas infrastructure or physical lines, of the network in order to perform properly.   There will have to be some sort of error checking for if the network is down or inaccessible.

4.3 Software Interfaces

This product will be connecting remotely to a MySQL database that is already set up and is internet-connected. The operating system where the software will run is the Android OS, which comes with a software framework that will be utilized, including many prepackaged components to do things like create menus, hookup buttons, and other common functions expected of a mobile device. Communication will be between the phone and the server housing the database, which will be sending queries or updates and receiving the information back to the application.

4.4 Communications Interfaces

This will be an Android application, and as described above, this will be communicating with a database server, and so will be making use of the Android network and HTTPS in order to communicate. There is no email or messaging currently, but this may change. The primary form of communication will be database transactions or requests. The system will need to be able to interact with the server system in order for users to log in.

4.5 Database Requirements

4.5.1 Database Interface

- Users shall have access to input information into Use profile information fields
- Trainers shall have access to input information into Trainer profile information fields.
- Trainers shall have access to Penalty, Credit and Goals options.

4.5.2 Credit/Debit Requirements

- Correct data must be entered before a credit or penalty can be assessed.
- Users can select a Trainer to associate with their account.
- Selecting Penalty option engages withdrawal workflow.
- Selecting Credit option engages credit workflow.
- Selecting Goals option engages return of penalties to user account.

4.5.3 Compliance Requirements

- The database will have a functional audit trail.
- The system will limit access to authorized users.
- Selecting Credit option engages credit workflow.

4.5.4 Security Requirements

- Users can enter their user profile, schedule, goals, and financial information; however, cannot control when penalties or credits are accessed.
- Trainers can enter their trainer profile, but can only see the users' schedule.
- Trainers can control how penalties and credits are assessed to the user's account.
- Trainers control when the goals are met, which returns the penalties to the user account.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The primary performance requirement is speed of the network. The application itself will only have minimal logic and so there should be little to no issues with the computation required by the phone itself.

Booting up/ Authenticate login screen:

- Users shall be able to log in to the application and received confirmation in less than 5 seconds (< 5 sec).

Retrieving Data:

- Trainer shall be able to upload workout results and reieved confirmation in less than 5 seconds (< 5 sec).

Application Reporting Back:

- Application shall confirm data being updated in the database in less than 5 seconds (< 5 sec).

Database Reporting Back:

- The database server shall be able to sustained service to 100 multiple users at the same time without any performance issues.
- The database shall be able to handle multiple transactions at given time (e.g multiple database requests).

5.2 Safety Requirements

There are no safety requirements with this application, other than any normal hazards of a mobile device. The only hazard is a user using the device when they should not be, such as while driving.

5.3 Security Requirements

The application must be able to link up with the server database system in order for users to properly log in and be identified. This information must be kept secure and shall be store in special database secure fields.

5.4 Software Quality Attributes

The primary attribute of this application will be usability given the large amounts of data and information that will be presented on such a small screen, as well as the user's ability to input data

into the device in a reasonable manner that should not be that much more difficult than if they were at an actual computer.

As usability is hard to quantify, substantial user testing will be needed and feedback gathered in order to determine if the application can generally be considered usable. Because this application will be on a phone, portability is also important. We don't want it to take up so much space or be too slow causing the user's to not be able to fit it on the device.

6. Other Requirements

6.1   System Evolution Requirements

Adaptability to hardware/software evolution will be accomplished recompiling the source code with the same or a newer version of the Android SDK framework. The main approach will be changing the project configuration parameters to be compatible with newer version of the operating system or software APIs available in the future.

In terms of future user's needs, we will add a notification system feature. In this add-on feature the user will be notify via email or text message if they miss a workout routine and how many points are being deducted. This feature will also be used to reminder to inform to the user they have a pending routine that week or within the following days.

The system may also incorporate a "track user location feature" which will be used to identify if the user is on or the near the gym at the right time and sending a notification to the user about his/her pending workout routine. This will work in parallel with the user reminder features.

7. Management Issues

7.1 Milestone and Schedule

Project Milestones –

- Project organization – identify project requirements , deadline and team members
- Requirements Analysis – SRS development
- Develop Architecture Diagrams and Models
- Develop Class Models
- System Integration and Testing
- Complete project documentation – user manual , software manuals, sample test cases
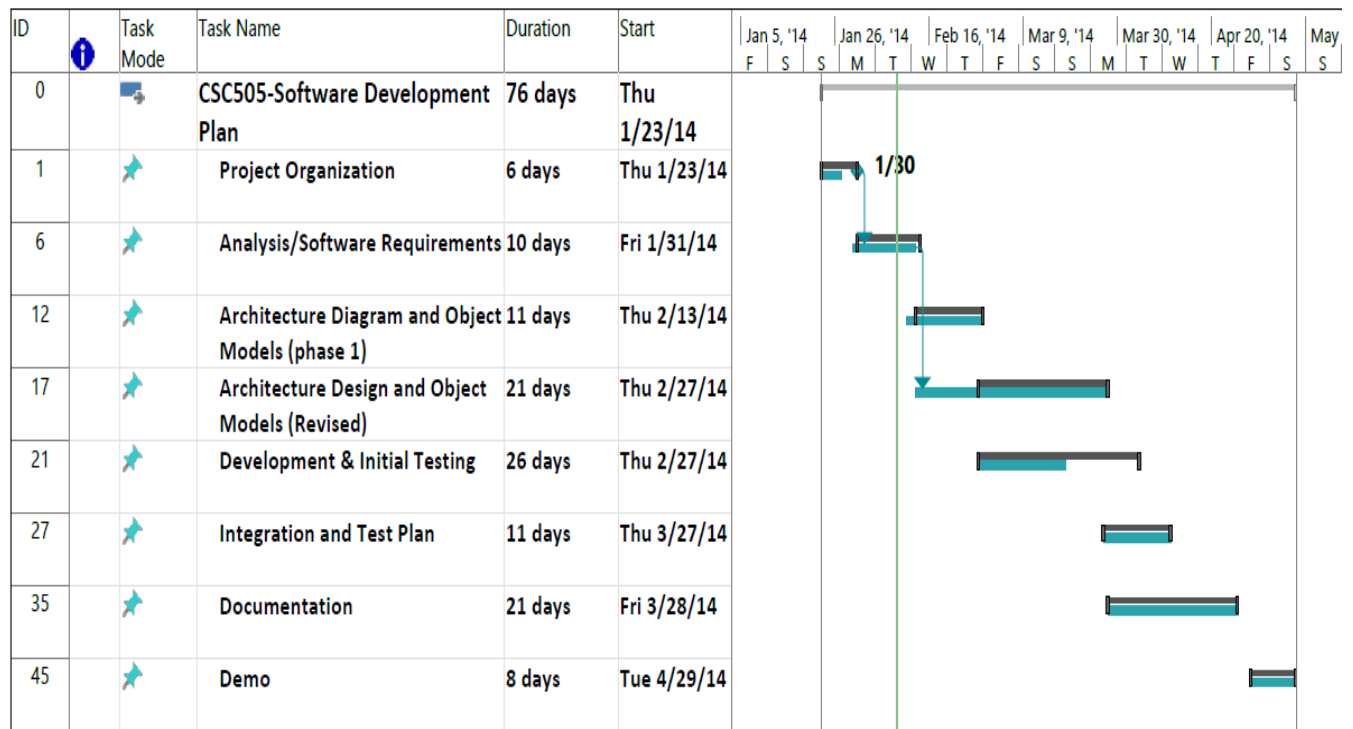- System Demonstration routine and run video.

ID	Task Mode	Task Name	Duration	Start
0		CSC505-Software Development Plan	76 days	Thu 1/23/14
1		Project Organization	6 days	Thu 1/23/14
6		Analysis/Software Requirements	10 days	Fri 1/31/14
12		Architecture Diagram and Object Models (phase 1)	11 days	Thu 2/13/14
17		Architecture Design and Object Models (Revised)	21 days	Thu 2/27/14
21		Development & Initial Testing	26 days	Thu 2/27/14
27		Integration and Test Plan	11 days	Thu 3/27/14
35		Documentation	21 days	Fri 3/28/14
45		Demo	8 days	Tue 4/29/14

Figure 2-Estimated Project Schedule with Milestones

7.2 Resources and Inventory

Inventory List

- Android Phones / Testing platforms- 2 Android Phones (Galaxy 3 and Galaxy Nexus) to test application in a real platform environment.

- Developers Laptops – 3 Developer Laptop with Eclipse IDE and Android SDK for development

- Server /Admin Laptops- 2 Laptops with tools (Putty-SSH and XMING) to connect to the server and manage files and Data base administration

- Server / Database - Remote server running Debian 7.3/3.2.51-1 x86_64 GNU/Linux with MySQL Server version 5.5.35

Team Members Background –

- Andrew – *Objects/classes, UML, MySQL, C++, Java, GUI, Web applications experience*

- Omar – *Schedules, Project management, integration and testing, C++, XML, Client/Server applications, Linux*

- Rick - *Class Models, PHP Python, server side programming, Android Emulator, XML*

- Daven – *Java, Python, Linux, programming experience*

- Brian – *XML, C++, MySQL, Databases experience*

8. Risk Management

8.1 Software Backup

With every increment or upgrade the software images/packages will be saved and controlled in a software repository. In case of losing or a repository corruption and alternate backup shall be available to restore the full system to a previous state.

8.2 Other Platform or Website

In case of losing one or more of the team member we may need to re architecture the mobile app to run as a mobile-accessible website. This backup plan doesn't require the Android OS framework to runs and in theory will work independently as a website in a regular browser. This will give us more flexibility in case of an emergency.

8.3 Demo Video

In case of Android emulator failure or not able to display the Android phone for the Demo day, the team shall have a Demo video that shows the multiple features of the application running in the emulator or the phone.