Requirements Specification

for

FH Mobile Application

Version 1.0

Prepared by

Omar Rivera
Andrew Poirier
Daven Amin
Rick Rejeleene
Brian Strattard

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

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 application. 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

This project is to create software and convert it to a mobile application. The benefits of this project are to be able to use a mobile device to make the fitness goals attainable by providing a financial motivation to the end user. 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' or utilize GPS coordinates to engage the penalty for missing scheduled workouts.

The software being used for development is the Apple and Android development kits, Xcode and Android SDK. The project is being managed by a server running INSERT SOFTWARE.

1.4 Definitions, Acronyms, and Abbreviations

Term	Definition

1.5 References

1.6 Overview

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 or computer. Any phone that supports Android 4.3.1 or greater will be able to install the applications and run the application from their phone. This app will allow users to have a seamless experience as if they were doing so on their pc through their web browser.

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 will be able to run the application. The application is developed with the Android SDK and SSH client respectively. The intent of this application is to utilize the images and ordering functions on a mobile device but with the functionality and interface designed for mobile phone. The application should be integrated with the existing database for easy updates.

2.4 Design and Implementation Constraints

The software must run on the Android operating system. The mobile phone has existing hardware/software constraints. The database used by the software needs to be the same one that is used by the existing database. The software must also use the language supported by the Android development environment, java plus the Android SDK.

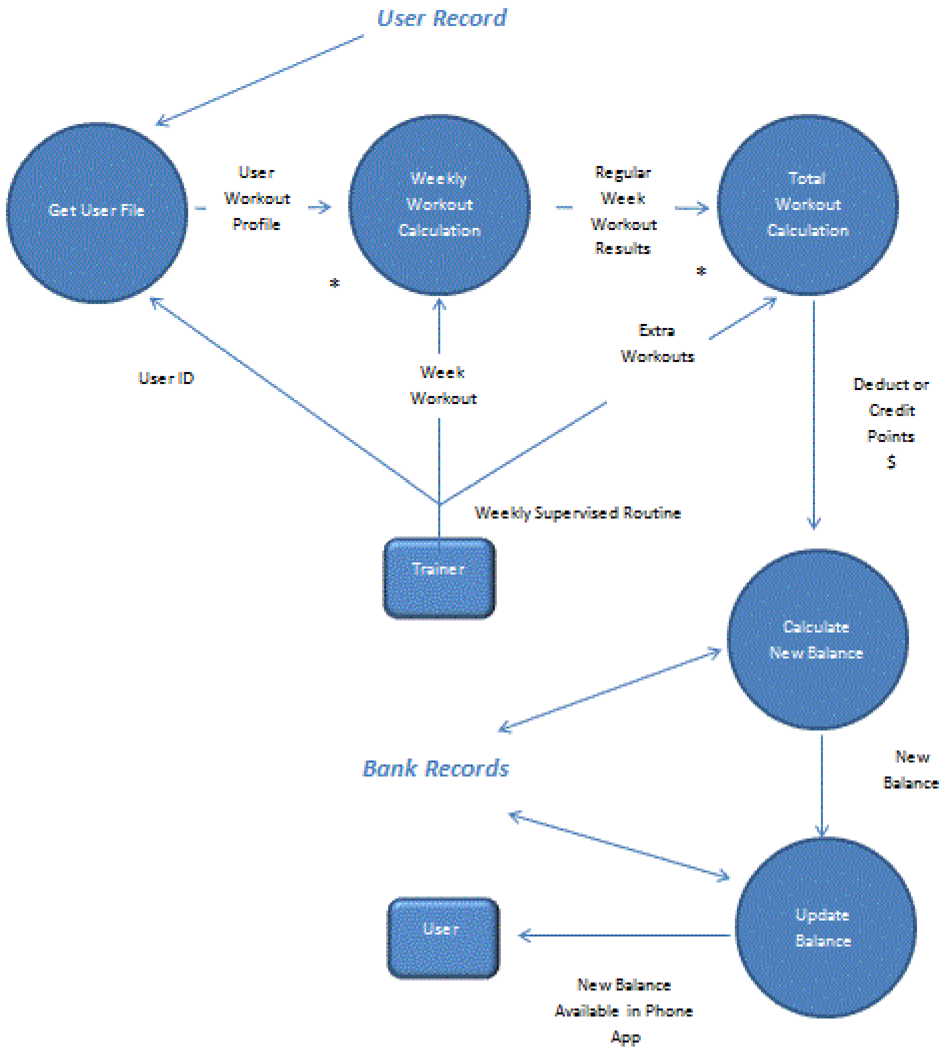
2.5 User Characteristics

2.6 General Constraints

2.7 Assumptions and Dependencies

The system is dependent upon the server. Customers will have to download the application from Google Play.

2.8 System Models


User Record
Get User File
User Workout Profile
Weekly Workout Calculation
Regular Week Workout Results
Total Workout Calculation
*
*
User ID
Week Workout
Extra Workouts
Deduct or Credit Points $
Weekly Supervised Routine
Trainer
Calculate New Balance
Bank Records
New Balance
Update Balance
User
New Balance Available in Phone App

2.8.1 DFD0

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 provide 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:

- The user input login name conflicts with an existing account in the system.
  - The system prompts the user to select a different login name.
- The user declines the terms of service.
  - 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

- The system cannot validate the user's credentials.
    - System informs user and does not log them into the system.

3.1.3 Use Case: User inputs new Workout Schedule

The system informs the user of the error and prompts for valid input

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

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

3.2 Requirements

3.2.1 User Platform Requirements

- Run on a “smart phone” 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.

3.2.3 Workout Schedule Requirements

- Allow the user to create a “workout schedule” 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”.

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.

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.

3.2.8 Workout Schedule Requirements

- Workout Schedule information shall include a “workout metric”.

4. External Interface Requirements

4.1 User Interface

UI-1: All users shall be able to access the website or mobile application.

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

UI-3: User Interface shall allow Authorizing Individuals to apply penalty or credit to associated User Accounts.

UI-4:

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 Android phones. In theory the application will be able to run by other devices that can emulate the Android, but this will not be a consideration during design.

As this is a mobile device, it will be using the 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 infrastructure, be it wireless communication points 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 the same one that the website connects to. This allows for use by users of both computers and the phone. The operating system the software runs on will be the operating system the Android runs on, 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. The only communication will be between the phone and the server housing the database, which will be sending queries or updates and receiving the information back. The logic associated with the website will be duplicated on the phone, so there will be little in the way of a server side component performing logic.

4.4 Communications Interfaces

This will be an Android application, but may still link to web pages that are not necessary to duplicate. 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 forms of communication will be database transactions or requests. The system will need to be able to integrate with the INTERNAL system in order for users to log in.

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.

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 INTERNAL system in order for users to properly log in and be identified. This information must be kept secure.

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

Appendix A: Issues List

Bug_Tracker.xls – This will be a list of the outstanding bugs that are left at the end of the project.

7. Management Issues

7.1 Milestone and Schedule

7.2 Recource Inventory

8. Disaster Issues

8.1 Software Backup

8.2 Other Platform or Website

8.3 Demo Video