CS4670 Coursework - Workout Recorder App

James Reed

February 27, 2013

Contents

1 Introduction

The purpose of this report is to produce a full and complete set of requirements and goals for a workout recorder app, along with a detailed scope and diagrams to support all of this. I will first analyse what the client's reqirements are and how these translate into high level requirements, then break these down into functional and non-functional requirements. I will then use these requirements to produce a set of goals for the system and the needed diagrams (e.g. UML, sequence, collaboration, etc) that will further explore the system. The goal of this report will be such that, when it is finished, a developer or team of developers will be able to take it and build the system it describes.

2 Overview of the product niche and the tasks involved

There are a lot of people who go to the gym or do workouts to keep themselves in shape and at points these people will want to know what the progress of their workouts is so that they can feel a sense of achivement and be able to quantify what they're doing. This app will aim to fill that market so that people who do workouts and similar will be able to record and track their accomplishments and decide how they need to adjust their schedules accordingly.

There are three main goals this Workout Recorder App will aim to achieve:

- Allow users to record and track workouts they've done
- Allow users to see progress over a time period of workouts
- Help users to improve or more easily adjust their training schedules depending on the results and analysis available with this app

These tasks provide the basis for the high-level requirements below, which then are further divided into more specific functional and non-functional requirements.

3 Requirements

3.1 High-level Requirements

The high level requirements of this project are as follows:

- Record which workouts are done or gym classes are followed that day
- Record progress made and goals achieved per workout
- Be able to visualise achivements/goals via graphs or other statistical methods
- App is able to work without being connected to the internet

- Non-technical and intuitive user interface
- Development time of three to six months
- Budget of £20,000-£30,000

3.2 Functional Requirements

The functional requirements for this system will be:

- `id:1` User will be able to record the type and number of workouts/gym classes done that day and be able to accept workout types that the app doesn't list from the user via a form or similar
- `id:2` User should be able to record progress (time, distance, crunches done, weight lifted) and tick off goals (ran for 20 minutes, lifted 120lb)
- `id:3` Progress should be able to be recorded in units of the user's choosing (i.e. weight should be able to be recored in lb, kg or stone, etc)
- `id:4` Progress should, after being entered, immediately be available for comparison and dissemination by the user via statistics (graphs, tables etc)
- `id:5` Statistical analysis of progress, achievments and goals should be available for all workouts and exercise types, provided enough information is available
- `id:6` The user should be able to access their workout statistics from anywhere that is able to launch the app via a login system (other android/iOS devices, pc, etc)
- `id:7` The app should (if possible) upload the user's statistics to their user account
- `id:8` (Optional) The app should provide facilities for daily/weekly targets and be able to alert the user if their average exercise so far isn't going to allow them to achieve that (say a user sets up so that they should burn 500kCal by 6pm, it should alert the user if they didn't input in their exercise that amounts to that by 6:15)

3.3 Non-functional Requirements

The non-functional requirements for this system will be:

- `id:9` The user interface will be non-technical and intuitive
- `id:10` The statistical analysis tools will be easy to disseminate, informative and quick to load (less than 3 seconds to load any particular graph/table/etc)
- `id:11` User's data should be easily accessible from anywhere, even if the app is not on the device or pc the user is currently using (maybe via some online login

which displays the statistical breakdowns and user data that can be viewed via a browser)

- `id:12` The app should be able to accept new data from the user at any point during its operation
- `id:13` The information the user inputs should (if possible) be able to be converted to different units (e.g. distance ran for x minutes is y kCal)
- `id:14` The app should be able to function without having access to the internet

3.4 Noun-Verb analysis

The full list of nouns is: user, progress, goal, statistic, graph, achievement, workout, information, user account, target, input, user interface, data, unit

The full list of verbs is: record, tick off, comparison, dissemination, analyse, access, provide, alert, load, accept, convert, function

4 Prioritising Requirements Using MoSCoW

The "must-have" requirements for this project are:

Requirement IDs: 1, 2, 4, 9, 14

The "should-have" requirements for this project are:

Requirement IDs: 3, 5, 10, 12

The "could-have" requirements for this project are:

Requirment IDs: 7, 8

The "won't-have" requirements for this project are:

Requirement IDs: 6, 11, 13

5 RUP elaboration

5.1 Project scope

The scope for the first release of this project will include the success factors for the release and the requirements that should be included in this first release. This are:

- To have not exceeded the budget by more than 5-10% (exact number decided by the client)

- To have attracted 10-15% of the 100,000 customers wanted by the end of the year, in the first month via advertising so that it can be spread more easily by word-of-mouth and similar techniques
- To have all of the 'must-have' and 'should-have' requirements realised in the first release, which are IDs 1, 2, 3, 4, 5, 9, 10, 12 and 14
- Achieve the three main goals of the project

5.2 Use Cases

Use case ID	1
Use case name	Entering a new workout
Description	User wants to enter a new workout, either pre-defined or custom
Pre-conditions	There is either a pre-defined template or form for creating a custom workout
Standard path	User chooses standard workout (e.g. running), fills out relevant data, including any targets if the user wishes to set them
Alternate path	User chooses non-standard workout (e.g. some unusual martial arts style, etc) and as well as the usual data (name, any targets) the user will have to fill out what units they want to use for the workout and how these units convert into other units (e.g. how many minutes of the workout = x kCals). Alternatively, the user can enter a super-category and the app will work out a rough conversion
Post conditions	A new workout is created and is available for use (e.g. viewing, new target setting, etc)
Exceptions	All fields of either form for the new workout are not filled out –¿ Error is thrown and user informed to fill out all necessary fields

Use case ID	2
Use case name	Updating a workout's progress
Description	User wants to update a workout's progress for the day/week
Pre-conditions	
Standard path	
Alternate path	
Post conditions	
Exceptions	