
Software Requirements Specification

for

Snappy Workout

Version 1.0 approved

Prepared by Team Snappy

Jianqiang Du, Peter Bui, Taylor Bui, Trung Nguyen

March 10, 2020

1. Introduction

1.1 Purpose

Snappy Workout Mobile Application is for Android Operating System 6.0 and above. The users are expected to be beginners to advance weight lifters. Snappy Workout is intended to aid and log workout progression through user inputted data.

1.2 Document Conventions

Each requirement statement has its own priority.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, marketing staff, and testers. The rest of the document contains specifications on the development of Snappy Workout Mobile Application.

Reading Sequence:

Developers: Start at Section 2 and read all sections in order.

Marketing Staff: Start at Section 2 then skip to Section 4.

Testers: Start at Section 2 then skip to Section 4.

1.4 Product Scope

Snappy Workout Mobile Application is a mobile application software intended to be used before, during, and after a workout session. The main point of this application is that it will aid in logging, organizing, and recording workout sessions. Revenue will be generated through advertisement throughout the software as well as premium membership that disables advertisement.

1.5 References

[RFP - Team Snappy - 02/05/20 - v.1.0](#)

[Use Cases - Team Snappy - 02/20/20 - v.1.0](#)

2. Overall Description

2.1 Product Perspective

This application is a new self-contained product that operates on the Android Operating System. All of the features will be newly implemented and developed by Team Snappy.

2.2 Product Functions

The major functions of this application includes:

- Registering and managing user information
- Creating, editing, exporting, importing, sharing workout plans
- Viewing and exporting workout data

2.3 Operating Environment

The software requires a Mobile device running Android Operating System 6.0 and above with at least 1GB RAM and 300MB of storage.

2.4 Design and Implementation Constraints

The design and content must follow professional standards.

2.5 User Documentation

There will be a tutorial presented when the application is first used.

2.6 Assumptions and Dependencies

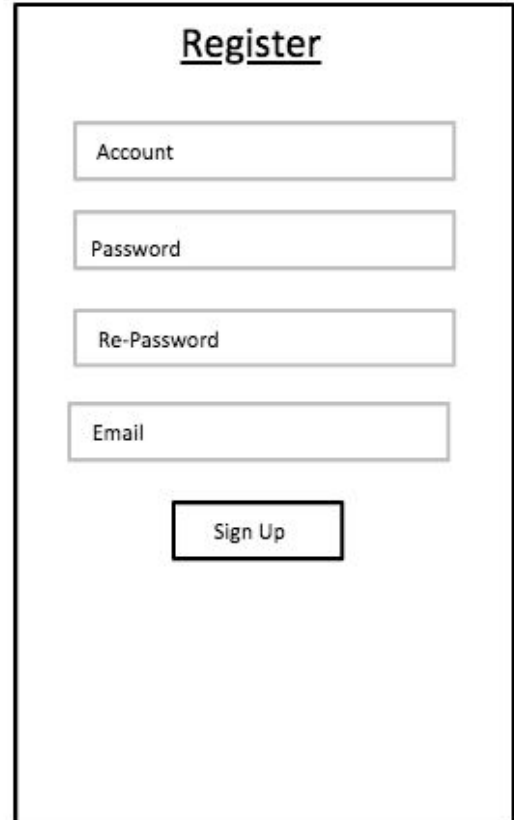
Google Authentication login will be assumed to be available and working. In addition Android Operating system is assumed to be functional and stable.

3. External Interface Requirements

3.1 User Interface



The login form for Snappy Workout features the app's logo at the top. Below the logo are two input fields for 'Account' and 'Password'. A 'Login' button is positioned below these fields. A link labeled 'Register new account!' is located just below the login button. At the bottom of the form is a 'Sign in with Google' button, which includes the Google logo.



The register form is titled 'Register' at the top. It contains four input fields: 'Account', 'Password', 'Re-Password', and 'Email'. A 'Sign Up' button is located at the bottom of the form.

Profile

Name

Weight

Age

Gender

Height

Goals: {Gain Mass, Loss Fat,...}

save

Plan

Import/Export

Plan Name

Number of Week

Start & End Date


List of Plan

Add Remove

Day: 1 **Shoulder** 08-12-2018

incline dumbbell bench press 10 sets 10 reps ☐

incline dumbbell bench press 10 sets 10 reps ☐

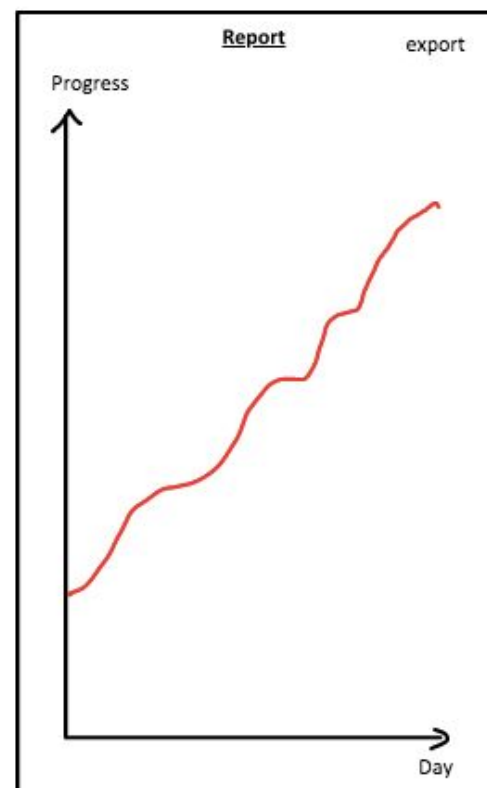


Peak: 145 lbs

Last: 15 lbs

Goal: +750 lbs

- 120 +



3.2 Hardware Interfaces

Smartphones that are able to run Android Operating System 6.0+ will be used to open up the Mobile Application.

3.3 Software Interfaces

Snappy Workout Mobile Application will work with google authentication for the login and registration. The application will also work with mongoDB in order to store the user workout and data. Flask framework will be used for the backend server.

3.4 Communications Interfaces

RESTful API will be utilized to deliver data to the user's client.

4. System Features

- Login
- Registration
- Workout Plan
 - View
 - Update
 - Export
 - Import
 - Add/Remove Exercise
- View Techniques
- View Statistics / progression
 - Export Stats

4.1 Use Case 1

Use Case 1	Register Account
Actors	User
Pre-Conditions	No User is logged in New User's Information is present Must Have Network Connection
Flow on Control	<ul style="list-style-type: none">• Clicks Register Button<ul style="list-style-type: none">◦ Display input fields for user

	<ul style="list-style-type: none"> ● Fill out field ● Click submit <ul style="list-style-type: none"> ○ Validate input field ○ Updates database showing new user information
Post-Conditions	The system now displays the new user information and grants them user privilege. Also saves data.
Error-Condition	Input field is not validated <ul style="list-style-type: none"> ● Empty username or email, password ● Character count exceeds max count (60)
Non-Functional Requirements	Data should be processed and updated within 5 seconds.

4.2 Use Case 2

Use Case 2	Disable Account
Actors	User
Pre-Conditions	User must be logged in Must be on the management page Must Have Network Connection
Flow on Control	<ul style="list-style-type: none"> ● Mark the disable action <ul style="list-style-type: none"> ○ System prompts for confirmation ● User clicks confirm button <ul style="list-style-type: none"> ○ System shows user is disabled
Post-Conditions	Account is disabled
Error-Condition	<ul style="list-style-type: none"> ● Network connection is lost during the process.

4.3 Use Case 3

Use Case 3	Update User Information
Actors	User

Pre-Conditions	User must be logged in Must be on the management page Must Have Network Connection
Flow on Control	<ul style="list-style-type: none"> • User clicks on update button <ul style="list-style-type: none"> ◦ System displays input fields • User modifies information • User clicks submit button <ul style="list-style-type: none"> ◦ System validates modifications ◦ System updates the database
Post-Conditions	User information in database is changed and updated
Error-Condition	Input field is not Validated <ul style="list-style-type: none"> • Empty field • Character count exceeds max count (60) for all fields
Non-Functional Requirements	Data should be processed and updated within 5 seconds.

4.4 Use Case 4

Use Case 4	Create Workout Plan
Actors	User
Pre-Conditions	Must be logged in.
Flow on Control	<ul style="list-style-type: none"> • User clicks Create Workout Plan button <ul style="list-style-type: none"> ◦ System transfers user to create workout plan input page
Post-Conditions	User information in the database is changed and updated.
Error-Condition	<ul style="list-style-type: none"> • There are already 20 defined Workout Plans. • The Workout Plan name cannot exceed 60 characters.
Non-Functional Requirements	System should process the instruction and update the database within 5 seconds.

4.5 Use Case 5

Use Case 5	Remove Workout Plan
------------	---------------------

Actors	User
Pre-Conditions	Must be logged in. Must have at least 1 Workout Plan. Must have network connection.
Flow on Control	<ul style="list-style-type: none"> • User selects the Workout Plan to be removed • User clicks submit button <ul style="list-style-type: none"> ◦ System prompts for confirmation • User clicks confirm <ul style="list-style-type: none"> ◦ System shows that Plan is removed.
Post-Conditions	System removes the Workout Plan from the database.
Error-Condition	There are no Workouts in the database.
Non-Functional Requirements	System should process the instruction and update the database within 5 seconds.

4.6 Use Case 6

Use Case 6	Update Workout Plan
Actors	User
Pre-Conditions	Must have network connection. Must have at least 1 Workout Plan.
Flow on Control	<ul style="list-style-type: none"> • User clicks update Workout button <ul style="list-style-type: none"> ◦ System transfers user to input page • User modifies Workout name and clicks submit <ul style="list-style-type: none"> ◦ System shows the updated data • User modifies Exercise and clicks submit <ul style="list-style-type: none"> ◦ System shows the updated data • User modifies Exercise date and clicks submit <ul style="list-style-type: none"> ◦ System shows the updated data
Post-Conditions	System saves and updates database
Error-Condition	Workout name exceeds 60 characters.
Non-Functional Requirements	System should process the instruction and update the database within 5 seconds.

4.7 Use Case 7

Use Case 7	Export User's Workout Plan
Actors	User
Pre-Conditions	User is logged in. User has at least 1 Workout Plan
Flow on Control	<ul style="list-style-type: none"> ● User clicks export User's Workout Plan <ul style="list-style-type: none"> ○ User selects the Workout plan to export ○ User selects submit
Post-Conditions	Program saves a file to the mobile device.
Error-Condition	No file is saved to the mobile device.
Non-Functional Requirements	Data should be processed and saved in 5 seconds.

5.

5.1 Use Case 8

Use Case 8	Import User's Workout Plan
Actors	User
Pre-Conditions	Must be logged in.
Flow on Control	<ul style="list-style-type: none"> ● User selects Import option <ul style="list-style-type: none"> ○ System prompts user to select file ● User selects file to import <ul style="list-style-type: none"> ○ System asks user to confirm ● User confirms selection
Post-Conditions	<ul style="list-style-type: none"> ● Updates User's database ● Adds imported Workout Plan to User's Workout Plan
Error-Condition	<ul style="list-style-type: none"> ● Invalid or corrupted file. ● User's Workout Plan is full.
Non-Functional Requirements	<ul style="list-style-type: none"> ● Data is processed and save within 5 seconds

5.2 Use Case 9

Use Case 9	Share Workout Plan
Actors	User
Pre-Conditions	User is logged in User must have at least 1 Workout Plan
Flow on Control	<ul style="list-style-type: none"> • User clicks Share Workout Plan • User selects Workout Plan <ul style="list-style-type: none"> ◦ System asks User for the user ID to send to. • User inputs another User's ID <ul style="list-style-type: none"> ◦ System confirms User ID • User selects submit.
Post-Conditions	<ul style="list-style-type: none"> • Another use receives an option to accept or reject the Workout Plan
Error-Condition	<ul style="list-style-type: none"> • System cannot find the user ID in database
Non-Functional Requirements	Data is processed and save within 5 seconds

5.3 Use Case 10

Use Case 10	Accept Workout Plan
Actors	User
Pre-Conditions	Users must be logged in. Another User must have sent a share Workout Plan.
Flow on Control	<ul style="list-style-type: none"> • User selects the prompt to accept Workout Plan <ul style="list-style-type: none"> ◦ System shows the Workout Plan accepted and asks User to confirm • User selects the confirm button.
Post-Conditions	Program saves and updates database with new Workout Plan
Error-Condition	Workout Plan is full.
Non-Functional Requirements	Data should be processed and updated within 5 seconds.

5.4 Use Case 11

Use Case 11	Mark as Complete
Actors	User
Pre-Conditions	User must be logged in. User must be in Workout menu.
Flow on Control	<ul style="list-style-type: none"> • User selects the box
Post-Conditions	The specific exercise box is filled.
Error-Condition	None
Non-Functional Requirements	System should process the instruction and update the database within 5 seconds.

5.5 Use Case 12

Use Case 12	Stopwatch
Actors	User
Pre-Conditions	Must be logged in
Flow on Control	<ul style="list-style-type: none"> • User selects the stopwatch option. • User selects start stopwatch. • User selects stop stopwatch.
Post-Conditions	System saves and updates database with the time.
Error-Condition	None
Non-Functional Requirements	System should process the instruction and update the database within 5 seconds.

5.6 Use Case 13

Use Case 13	Add Exercise
-------------	--------------

Actors	User
Pre-Conditions	Must be logged in. Must have created at least 1 Workout Plan.
Flow on Control	<ul style="list-style-type: none"> ● User selects Add exercise in Workout Plan <ul style="list-style-type: none"> ○ System shows list of all exercise ● User selects the exercise. ● User selects next. <ul style="list-style-type: none"> ○ System shows new menu asking how many. ● User selects a numeric choice from a dropdown menu. ● User clicks next. <ul style="list-style-type: none"> ○ System shows that the exercise is added
Post-Conditions	System updates the database and confirms the exercise is added.
Error-Condition	None
Non-Functional Requirements	System should process the instruction and update the database within 5 seconds.

5.7 Use Case 14

Use Case 14	Remove Exercise
Actors	User
Pre-Conditions	Must be logged in. Must have at least 1 Workout Plan. Must have at least 1 exercise added.
Flow on Control	<ul style="list-style-type: none"> ● User selects Remove Exercise <ul style="list-style-type: none"> ○ System shows exercise user has added ● User selects exercise. ● User selects next. <ul style="list-style-type: none"> ○ System removes exercise from the Workout Plan.
Post-Conditions	User information is updated to database. The exercise selected is removed.
Error-Condition	None
Non-Functional Requirements	Information is updated and saved within 5 seconds of confirming.

5.8 Use Case 15

Use Case 15	Update Exercise
Actors	User
Pre-Conditions	User must be logged in. User must have at least 1 Workout Plan. User must have at least 1 Exercise.
Flow on Control	<ul style="list-style-type: none"> ● User selects Update Exercise <ul style="list-style-type: none"> ○ System displays all information about exercise. ● User selects Amount <ul style="list-style-type: none"> ○ System asks how many. ● User selects a numerical value from a dropdown menu. ● User selects next. <ul style="list-style-type: none"> ○ Information is saved and database is updated.
Post-Conditions	User's database is updated.
Error-Condition	None
Non-Functional Requirements	Status is updated and saved within 5 seconds.

5.9 Use Case 16

Use Case 16	Export Progression
Actors	User
Pre-Conditions	Logged in as a user.
Flow on Control	<ul style="list-style-type: none"> ● User clicks on export. <ul style="list-style-type: none"> ○ Show alert "Saved"
Post-Conditions	Progression is saved into local storage.
Error-Condition	Local storage is full.
Non-Functional Requirements	System should process the instruction within 5 seconds.

6. Nonfunctional Requirements

All of the instructions and database update should process within 5 seconds.

7. Other Requirements

All materials here are copyrighted by Team Snappy and may not be redistributed or resold without permission.