Software Requirements Specifications

Heavid

1.1

April 21, 2021

MCS Department, Biola University

Revisions

Target Audience: Those interested in finding and creating workouts tailored to each muscle group, especially those with hindered gym access during the pandemic

Project Team Members: Luke Ebbinghaus, Misheel Bayarsaikhan, Keziah Bulseco, Vanessa Garcia

Version Control History:

Version	**Primary Author(s)**	**Description of Version**	**Date Completed**
0.1	Luke Ebbinghaus, Misheel Bayarsaikhan, Keziah Bulseco, Vanessa Garcia	Initial team additions	2/9/21
0.2	Luke Ebbinghaus	Added to User Interface Requirements and Software Interface Requirements	2/10/21
0.3	Luke Ebbinghaus, Misheel Bayarsaikhan, Keziah Bulseco, Vanessa Garcia	Added software Sofware Product Features, Software System Attributes, and Database Requirements	2/11/21
1.0	Luke Ebbinghaus	Final Revisions	2/12/21
1.1	Luke Ebbinghaus	Requirements modified to reflect Final Product Implementation. Dropped and Modified requirements indicated. Updated database requirements. Updated Project Title.	4/21/21

Table of Contents

1 INTRODUCTION 1
1.1 Project Overview 1

2 SPECIFIC REQUIREMENTS 1
2.1 External Interface Requirements 1
2.1.1 User Interfaces 1
2.1.2 Hardware Interfaces 2
2.1.3 Sofware Interfaces 2
2.1.4 Communications Protocols 2
2.2 Software Product Features 2
2.3 Sofware System Attributes 3
2.3.1 Security 3
2.3.2 Interoperability 3
2.3.3 Performance 3
2.4 Database Requirements 3
2.4.1 Data 3
2.4.2 High-level Entities 3
2.4.3 Business Rules 3

1 INTRODUCTION

1.1 Project Overview

A health app with built-in workout management features. The primary function will be to create an interface where the user can find workouts relevant to specific body parts through an intuitive diagram. Additionally, the user will be able to create their own workout plans with an input of difficulty and targeted muscle groups. The user can then modify this plan as desired using the same diagram interface.

2 SPECIFIC REQUIREMENTS

2.1 External Interface Requirements

2.1.1 User Interfaces

[Req 1] [Dropped]	Scalable window without distorting elements. There will be a fixed minimum window size to preserve visibility.
[Req 2]	One main window with pop-up functionality and dynamic sizing
[Req 3]	Clickable diagram of a person will show exercises depending on the muscle group chosen
[Req 4]	Person diagram on all screens
[Req 5] [Modified]	Listed exercises will be clickable with extra information available
[Req 6]	Clickable working-list of exercises in the home screen which shows the current exercises added by the user into their workout
[Req 7] [Modified]	Add to / start workout functionality which transitions to a workout plan section
[Req 8]	Difficulty filters for the exercises will be selectable when a muscle group is selected

2.1.2 Hardware Interfaces

[Req 1] [Modified]	The software would be supported only on desktop devices. Mac and Linux, but not Windows
[Req 2]	The data we would be working with would be the string data type names of the exercises, the classification of the exercises by the difficulty and muscle group(s) that the exercise falls under in MySQL
[Req 3] [Modified]	The user would be using a desktop device to access the application which is created using C++ through QT
[Req 4]	The communication protocol that we will incorporate in our project is TCP to connect to the database

2.1.3 Sofware Interfaces

[Req 1] [Modified]	The main application will be implemented with C++14 standards
[Req 2] [Modified]	The application will interface with Qt 6
[Req 3] [Modified]	The application will interface with MariaDB and MySQL

2.1.4 Communications Protocols

[Req 1]	Software application will be bundled with a graphical user interface by default
[Req 2]	Interface between application and GUI will be handled by Qt
[Req 3] [Modified]	Database will communicate with the software through MySQL Connector/C++ and TCP/IP
[Req 4]	Data on exercises and their attributes will be stored in the database to be queried by the application

2.2 Software Product Features

[Req 1]	Database of variety of exercises based on different muscle groups and the difficulty of each exercise
[Req 2]	Illustrations of each exercise for a better understanding of the motions (picture or animation)
[Req 3]	Local save functionality for workout plans

2.3 Sofware System Attributes

2.3.1 Security

[Req 1]	The saved workouts will be safely stored in the user's machine
[Req 2]	Database cannot be changed by the user

2.3.2 Interoperability

[Req 1] [Modified]	The application will be tested to function in 2 different operating systems: Linux and MacOS

2.3.3 Performance

[Req 1]	Workout changes will be done locally
[Req 2]	Database will not change size unless explicitly updated

2.4 Database Requirements

2.4.1 Data

- String tests of exercise names, muscle groups, and difficulty types
- images of exercises represented as Binary Large Objects

2.4.2 High-level Entities

- Exercise: A specific exercise workout (i.e. pushups, squats, etc.)
- Muscle Group: A targeted muscle group (i.e. back, legs, etc.)
- Exercise Image: Image(s) of a given exercise

2.4.3 Business Rules

- Each exercise must work one or more muscle group(s)
- A muscle group must be targeted by one or many exercises
- Each exercise can have one or many image(s)
- Each image can correspond to only one exercise