Software Requirements Specification

Version 1.2

<<Annotated Version>>

June 18, 2020

Gym Assistant app

***Muhammad Wazier***

***Suliman Alshami***

Class 3

Submitted in partial fulfillment

Of the requirements of

CS 310 Software Engineering

# Table of Contents

[Table of Contents i](#_Toc77487619)

[List of Figures ii](#_Toc77487620)

[1.0. Introduction 1](#_Toc77487621)

[1.1. Purpose 1](#_Toc77487622)

[1.2. Scope of Project 1](#_Toc77487623)

[1.3. Glossary 2](#_Toc77487624)

[1.4. References 2](#_Toc77487625)

[1.5. Overview of Document 2](#_Toc77487626)

[2.0. Overall Description 4](#_Toc77487627)

[2.1 System Environment 4](#_Toc77487628)

[2.2 Functional Requirements Specification 5](#_Toc77487629)

2.2.1 User..............................................................................................................................4 2.2.2 Create aAccount......................................................................................................4 2.2.3 Login............................................................................................................................5 2.2.4 Submit Exercise Program ..........................................................................................5 2.2.5 View Existing Programs..............................................................................................6 2.2.6 Rate a Program..........................................................................................................6 2.2.7 Save a Program to Personal List ..............................................................................7 2.2.8 Edit Personal Program ..............................................................................................7 2.2.9 Send Message...........................................................................................................7

[2.3 User Characteristics 8](#_Toc77487648)

[2.4 Non-Functional Requirements 10](#_Toc77487649)

[3.0. Requirements Specification 11](#_Toc77487650)

[3.1 External Interface Requirements 11](#_Toc77487651)

[3.2 Functional Requirements 11](#_Toc77487652)

Use Cases..........................................................................................................................11

Use Case Index ................................................................................................................11

Use Case #1 – Create Account.........................................................................................13

Use Case #2 – Login.........................................................................................................14

Use Case #3 – Submit Program .....................................................................................15

Use Case #4 – View Program..........................................................................................16

Use Case #5 – Rate a Program ..........................................................................................17

Use Case #6 – Save Program..........................................................................................19

Use Case #7 – Edit Program...........................................................................................21

Use Case #8 – Send Message.........................................................................................23

[3.3 Detailed Non-Functional Requirements 24](#_Toc77487665)

Portability..................................................................................................................24

Reliability...................................................................................................................24

Ease of Use ..............................................................................................................25

Speed........................................................................................................................25

Size.............................................................................................................................25

Class Diagram..........................................................................................................26

# List of Figures

[Figure 1 - System Environment 4](#_Toc77487669)

[Figure 2 - User Use Cases 12](#_Toc77487671)

[Figure 3 - Class Diagram ……………26](#_Toc77487672)

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to present a detailed description of the Gym Assistant application .Content includes research, requirements, use cases, data requirements, usability. This document outlines key technical details of the Gym Assistant application.

This document is intended for both the stakeholders and the developers of the system.

## 1.2. Scope

Gym Assistant is a community based health and fitness mobile application. It aims to offer a social environment for people interested in health and fitness. The application will be fueled by user generated content, users will be able to submit and view exercise programs within the application. All programs in the application will be submitted by users of the application, so if a user finds a certain program works really well for them or finds a useful workout on the internet, they can submit it to the application for other users to try out. Users will be able to rate programs submitted to the application, this will help ensure high quality content within the application. The main objectives of our project is to create an app which enables the user to effectively lose weight, maintain weight and gain muscle in a sustainable manner with clear and concise plans

## 1.3. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| User | Any person who will use this app. |
| Google Firebase | A No SQL, real-time, online database used to permanently store application data. |
| GUI | (graphical user interface) is a system of interactive visual components for computer software. A GUI displays objects that convey information, and represent actions that can be taken by the user. The objects change color, size, or visibility when the user interacts with them |
| API | An application programming interface (API) is a computing interface which defines interactions between multiple software intermediaries. |
| IOS | iOS is a mobile operating system created and developed by Apple Inc. exclusively for its hardware. It is the operating system that presently powers many of the company's mobile devices, including the iPhone, and iPod Touch; it also powered the iPad prior to the introduction of iPadOS in 2019 |

## 1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## 1.5. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

# 2.0. Overall Description

## 2.1 System Environment

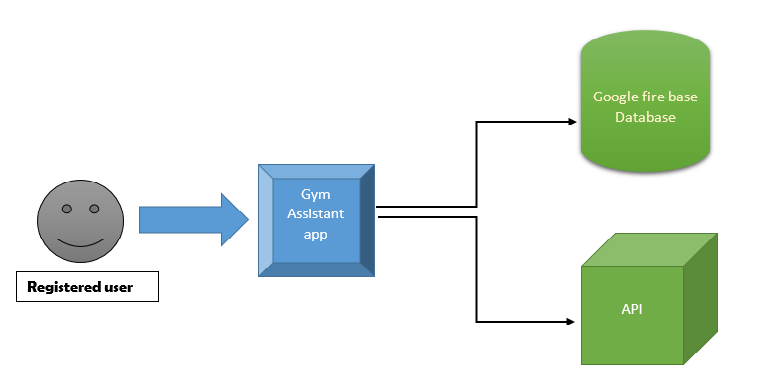


Figure 1 - System Environment

The application is designed to work on all devices that are running iOS version 8 or later. Currently this includes the iPhone.

The application will store all data in a Google Firebase database. The application will require an online connection at all times to allow for user sign on, user communication, and access to online user submitted content.

## 2.2 Assumptions & Dependencies

2.2.1 If no exercise routines are created by users, there will be no routines listed

in the application for users.

2.2.2 If no internet connection is available, users will be unable to log in or

access any functionality of the software.

2.2.3 Unregistered users will have no access to the software.

2.2.4 Users will only be permitted to register and gain access to the software

once they agree to the terms of service agreement.

## 2.**3** Functional Requirements

2.3.1 User

2.3.1.1 All users of the software shall have the ability to create an account

which is used to store user data and tie user actions to a user alias.

((User registration and login shall be mandatory)).

2.3.2 Create an Account

2.3.2.1 The system should provide the user with an easy to use GUI to facilitate

their creating an account.

2.3.2.2 The system shall ask for an email address and password.

2.3.2.3 The system shall notify the user if incorrect characters are used in the

email or password fields.

2.3.2.4 The system should notify the user if their email has already been used.

2.3.2.5 The system should notify the user if any required fields are left empty.

2.3.2.6 The system should not allow the user to create weak or unsecure

passwords.

2.3.2.6.1 The system should explain how the submitted password is

unsecure.

2.3.2.7 The system should prevent the user from completing registration if the

terms of service has not been agreed to.

2.3.3 Login

2.3.3.1 The system should provide a user friendly GUI to allow the user to login

when the application launches.

2.3.3.2 The system should prompt the user for their email address and

password.

2.3.3.2.1 The system should notify the user if submitted information is

incorrect.

2.3.4 Submit Exercise Program

2.3.4.1 The system should provide an intuitive UI for logged in users to allow

them to submit their exercise routine to the application.

2.3.4.1.1 The system should prevent the user submitting a blank or

empty routine.

2.3.4.2 The system shall add successfully submitted routines to the Google

Firebase Database.

2.3.4.3 The system should display user submitted routines from the Google

Firebase database in the appropriate section of the application.

2.3.5 View Existing Programs

2.3.5.1 The system should provide intuitive and user friendly navigation to allow

users to locate the current list of user submitted exercise routines.

2.3.5.2 Once selected, the system shall retrieve all user submitted programs

from the Google Firebase database and display them to the user.

2.3.5.3 The system shall display full details of an exercise routine once one is

selected by the user.

2.3.5.4 The system should allow quick and easy navigation between different

routines in the list.

2.3.6 Rate a Program

2.3.6.1 The system should provide a button to rate a program. Programs can be

rated up or down based on the level of success the user has with them

2.3.6.1.1 Programs will be displayed based on ratings. Lower rated

programs will be pushed to the bottom of the list before being removed.

2.3.7 Save a Program to Personal List

2.3.7.1 The system should provide a button to save a program from the public

list of user submitted programs to their own personal list.

2.3.7.1.1 Programs saved to a personal list are not visible to other

users.

2.3.8 Edit Personal Program

2.3.8.1 The system should provide an intuitive and user friendly UI to allow the

user to view and manage their personal list of workout programs.

2.3.8.2 The system should allow editing of programs saved to a user’s personal

list.

2.3.8.2.1 Multiple edits can be made and all changes must be

saved in real time

2.3.9 Send Message

2.3.9.1 The system shall provide an interface for sending messages between

users.

2.3.9.2 Messages should be sent in real time and have no delays.

***2.4* User Characteristics**

**Unregistered User**

Unregistered users will not be able to gain access to the application or use any of its features. The whole focus of the application is on user submitted content and social interactions, none of this is possible without an account alias to tie a user to. Additionally, the services offered by the application will be behind a ToS (Terms of Service) agreement, this model was chosen as users will be following programs submitted by other users completely at their own risk. No liability is accepted by the developer of the application or any parties in connection with the developer.

**Registered User**

Registered users will gain full, non admin access to the application. Once a user registers their details and agrees to the ToS they will be able to post workout routines, view workout routines, rate workout routines, save routines to their personal list for edits, chat with other users and use the gyms nearby feature.

**Administrator**

The administrator will maintain the data in the application behind the scenes. The administrator will deal with managing user accounts if any bans are levied against a user. They will also manage workout program data based on user feedback. If a program receives enough negative feedback it will need to be removed from the application by the administrator. The administrator will also carry out day to day housekeeping within the application and ensure the quality of the application content.

## 2.5 Non-Functional Requirements

Portability , Reliability , Ease of Use , Speed , Size

# 3.0 Requirements Specification

## 3.1 External Interface Requirements

## *User Interface*

## The user interface shall offer the user a logical representation of what the software is asking the user to do. Dropdown menus and buttons should be used where possible to aid the user. Input hints shall be used to aid the user when entering data.

## The application should have its logo present on each screen once a logo has been designed.

## A user friendly color scheme should be chosen , UI design should be carried out with visually impaired and color blind users in mind.

## The UI should have well defined constraints to ensure that the software displays correctly on the screens of all compatible devices. The UI should display in both portrait and landscape.

## The GUI should have continuity, all screens should have the same design and layouts should be consistent.

## *Hardware Interfaces*

The system shall be operated with a compatible Apple device using the

devices touch Screen && virtual keyboard .

## *API Interfaces*

The software must store user information and user submitted content in a

Google Firebase database using Cocoa Pod files to achieve

communications between the database and the application.

The software must show mapping information around the user’s current

location using the Google maps API.

## 3.2 Functional Requirements

**3.2.1 Use Cases**

**3.2.1.1 Use Case Index**

### 

