

**2022**

2013131 – Manodayrao Mahadoo

2010835-Louis Christopher Eliakim Mava

Fitness App

Report

Contents

[1. Introduction 3](#_Toc109400366)

[1.1 Purpose of document 3](#_Toc109400367)

[1.2 Scope 3](#_Toc109400368)

[1.3 Aims and Objectives 3](#_Toc109400369)

[2. Requirements 4](#_Toc109400370)

[2.1 Functional Requirements 4](#_Toc109400371)

[2.2 Non-Functional Requirements 5](#_Toc109400372)

[3. Overview of Screens 5](#_Toc109400373)

[3.1Design 6](#_Toc109400374)

[4. GUI 6](#_Toc109400375)

[4.1 GUI Components 6](#_Toc109400376)

[4.2 UI Layout 6](#_Toc109400377)

[4.3 System Panel 7](#_Toc109400378)

# Introduction

## 1.1 Purpose of document

The aim of this document is to describe the fitness app. This document provides information about the technical information and the implementation of the features for the app.

## 1.2 Scope

The application is targeted towards desktop platforms such as Windows or Mac OS. The application would consist of a front end along with a back end which would allow syncing across multiple devices. The app should use Java programming language and be implemented using the Swing framework. The application will be used by a wide variety of people and should have a user-friendly interface so that the user is able to navigate easily and seamlessly inputting his exercises done during the day and storing the data.

## 1.3 Aims and Objectives

The main objectives of this project are listed below:

• Develop Front-end interface using Java.

• Design a database using Mysql for the back-end.

• Store daily user activities in a text file.

• Design the different classes and controllers in Java.

# Requirements

## 2.1 Functional Requirements

1. The application should have an informative home screen describing the application to the user.
2. The application should have a register screen for new users.
3. 3The application should have a login screen for existing users.
4. The application should have a screen that allows users to enter their personal details.
5. The application should show a summary of all their details.
6. The application should validate all the data entered and prevent incorrect or empty data to be saved on the back-end.
7. The application should allow the user to input the duration of each workout done.
8. The application should allow the user to input the amount of reps done for a specific workout.
9. The application should allow the user to input the amount of sets done for a specific workout.
10. The application should allow the user to minimize and maximize it.

## 2.2 Non-Functional Requirements

1. The application should be implemented in Java.
2. The application should use a Mysql database for the back-end.
3. The application should not have a restricted screen size.
4. The system should use an online Mysql database hosted on a database server on the internet for viewing purposes.

# 3. Overview of Screens

1. Splash Screen

Provides the app name and the loading progress bar.

2. Login Screen

Requests the username and password for the user to be able to login into the application.

3. Register Screen

Allows new users to register into the application by entering their details in the text fields provided.

4. Main Screen

Allow the user to enter details of the workouts done in the predefined fields, also allow the user to enter a new workout now predetermined on the screen and record it in a file.

5. Database Review Screen

Details about all exercises performed in the past are recorded and the user is allowed to view the necessary details via the generated table.

6. Verification (Popup Message)

Informs the user whether the details entered are correct or not. Also provides indication of what is wrong.

## 3.1Design

The system has been modelled using Object Oriented Modelling techniques. The system has been decomposed into a handful of interacting objects which communicate via the use of methods and parameters. The application consists mainly consists of:

• GUI (Java.swing)

# 4. GUI

The GUI has been implemented in Java using the Swing framework.

The system has one main class which instantiates the controller objects and uses them to process data. The GUI uses Swing and makes use of J components such as JFrame/Jlabel.

The system has been designed as a single frame (window) system which uses Cards and the Card Layout to dynamically switch between cards to provide a mobile like UI with only one window. (hot reload effect achieved by the use of Card Layout)

## 4.1 GUI Components

The following Jcomponents have been used throughout the application.

• JFrame

• JPanel

• JButton

• JLabel

• JTextField

• JCheckbox

• JTable

## 4.2 UI Layout

The application makes use of Java Swing Layouts to arrange and organise Jcomponents within the frame(window). These layouts help to position the UI components on the screen and affect the look and feel of the application. This application has made use of several layouts for an optimal user experience. The following layouts have been used in the application:

1. SignUp Screen • Uses Container

2. Login Screen • Uses Container

2. Main Screen • Uses Grid Layout

3. Generate Screen • Uses ?  Layout

## 4.3 System Panel

The system panel at the top of the application. The system panel uses the Grid Layout and container Layout and is used to control the behaviour of the frame or window. The system panel contains the:

• Minimise button

• Maximise button (Which is disabled as changing size is set to false)

• Exit Button

The system panel also allows the user to click on it and drag the application around. This is implemented by using a listener on the system panel.