Names	Student 1 - Fawaz Alsafadi Student 2 - Ayman El Gendy
Project title	Gym Vision
Document	Functional specification
Student ID's	Student 1 - 15380871 Student 2 - 15395461
Date of submission	21/11/2020

Table of contents

Contents

1. Introduction

1.1 Overview

The system will be a software application that users will be able to download and use to judge whether they are performing a gym exercise correctly. The system will make use of computer vision and pose estimation to study a frame of a human doing an exercise and determine whether all their joints and limbs are in the correct position. This system is mainly aimed at beginners who are uncomfortable performing certain movements as they haven't had real guidance on how to perform the exercise. Performing exercises incorrectly is the number one cause of injuries in beginners/intermediates in the gym so eliminating the pressure on gym goers to seek help from unreliable sources and helping them avoid injuries is the inspiration behind this project.

The user will be required to stand in front of a camera and begin performing an exercise they chose. After the exercise has been completed the footage will be sent to our application at which time the analysis will occur. This will then return a result to the user. This result will include a 'correctness' metric which relates to how correct the users form was during the exercise. There will also be detailed feedback advising the user on how their form is incorrect with guidance on how to correct it with recommended exercises and stretches.

As beginners in the gym we noted a strong lack of support for new gym goers on how to perform gym exercises. There are also many unreliable sources online that has the potential to cause more harm than good and the option to hire a personal trainer can be very expensive and impractical. We hope that in developing this app we will solve these issues and provide a useful and highly accessible tool to assist anyone beginning their fitness journey.

1.2 Business Context

While we will not be developing our application in conjunction with a business, this application could potential be used by any professional personal trainer or any gym.

A personal trainer could user this app to determine whether a user's form in correct and help them modify it. They could also ask a client to share their exercise history with them with the 'add friend' functionality and help them correct any exercises they may be performing incorrectly.

A gym would be able to install booths, where a gym goes enters the booth and performs an exercise in front of the camera in the booth. With instant feedback and tips to correct form the gym could reduce injuries sustained while performing exercises incorrectly.

1.3 Glossary

- [1] *OpenCV Python* – Computer vision library for python
- [2] *Numpy* – Python library to assist with multi-dimensional arrays
- [3] *Docker* – Docker is a platform as a service product that delivers software in packages called containers
- [4] *Microsoft Azure* – Cloud computing service created by Microsoft for building, testing deploying and manging applications
- [5] *API* – Set of routines, protocols and tools for building software applications

2. General Description

2.1 Product / System Functions

The gym vision application will aim to assist users in ensuring that their stance and form is correct when performing various gym exercises for example “barbell squat” or “bent-over rows”. The analysis of the user’s pose will be carried out in the application and will be presented to the user in the application UI at which point the user will be able to build on the feedback from the application and improve the next time they carry out the exercise.

The application will work from a laptop utilising the camera which will feed footage of the user to the application. Once the footage has been captured using computer vision and pose estimation algorithms the user’s stance will be compared to data from correct form of athletes completing the same exercises. Based upon that result the user will begin to be guided to the correct form with adjustment messages informing them for example that their back needs to be straightened and suggest exercises to help them carry out the recommendation. The user can then select another exercise and begin to analyse their form and pose.

The user will be able to see a history of their previous workouts and see the list of suggestions and messages they received and when they were able to successfully complete the exercise. In addition, the user will also be able to maintain a friend list and add other users of the app to see what exercises the have successfully completed and ask them for help.

2.2 User Characteristics and Objectives

When the user initially accesses the software, they will be able to set up an account on our system. The account will provide them with the following features;

- Ability to estimate how well they are performing an exercise.
- Personalised guidance based on previous form estimation on how to improve your form.

It is not necessary for users to have a large amount of technical knowledge however, basic understanding of how to use a camera on their device, how to install an application and create an account is required.

From the user’s perspective, our system must be able to identify incorrect postures during the exercise. Our system must clearly indicate where users are going wrong and how they can adjust themselves accordingly. It should also show users how close they are to the correct form using either a percentage (e.g form 80% correct) or labels such as (Very bad, Bad, Average, Good, Very Good, Perfect).

2.3 Operational Scenarios

2.3.1 - Sign up

1. User clicks create account button
2. User is brought to account creation page and prompted for Email and secure password. User is prompted for information such as age, height and weight.
3. After the user clicks submit the app navigates back to the sign in page and the user is required to sign in with the new information.
4. User is prompted with error message if information on sign up is incorrect and the user must repeat step 2 again.

5. This use-case is complete when the user has successfully created an account.

2.3.2 - Sign in

1. User clicks the login button on the welcome page.
2. User is navigated to login page and asked for Email and password.
3. If information provided is incorrect the user will have to repeat step 2
4. This use-case is complete when the user has successfully logged in to the app.

2.3.3 - Tutorial

1. Before the user can begin using the app after successful login, they must complete the tutorial to be granted access as they must demonstrate they know how to use the app as to avoid injury.
2. The user will be presented with a sample exercise.
3. When the user completes the exercise, they will be taken to the app homepage however they may re-try the tutorial as many times as they want.
4. This use-case is complete when the user completes the tutorial.

2.3.4 - Homepage

1. The user will be taken to the homepage once all previous steps are completed.
2. The homepage will contain the dashboard for different exercises and app functions.
3. This use-case is complete when the user navigates to the home page and explores the different functionality.

2.3.5 - Select exercise

1. From the homepage once the user is successfully signed it they may navigate to the exercises tab and select an exercise they would like to perform.
2. The user is brought to another page where they will be shown their image through the camera.
3. The user then must stand in front of the camera and begin to perform the exercise.
4. The app will then capture the footage of the user and it will be sent to the API so that it can be analysed
5. The user will be presented with a summary page which will outline the result of the exercise and give the user advice on how to improve and what the user must do to successfully complete the exercises; where applicable.
6. The user then has the option to repeat the exercise or select a different one.
7. This use-case is complete when the user has successfully performed an exercise and received feedback.

2.3.6 - View exercise history

1. From the homepage once the user is successfully signed in they may navigate to profile tab to view their history.
2. In the history tab the user will be able to see all previous exercise attempts.
3. The user will also be able to see the previous feedback they received in case they may want to review it.
4. This use-case will be completed when the user reviews their exercise history and individual exercises.

2.3.7 - Add friends

1. Once the user has successfully signed in they may navigate through the homepage to the add friends tab.
2. From here the user will be able to supply the email of their friends to send a friend request.
3. The user will then receive a friend request to share their workout history with the other user.
4. The user can accept or deny the request.
5. This use-case is completed when the user has sent a friend request which is an optional feature.

2.3.8 - Viewing friends workout history

1. After successfully logging in and if the user has added a friend, they may view their workout history. The purpose of this use-case is so that the user can potentially as for help or advice from their friends.
2. The user can navigate to their friend's tab, here they can view friends completed exercise.
3. This optional use-case is complete when the user has viewed a friend completed exercise history.

2.3.9 - Signing out

1. The user must successfully log in using a created account.
2. Within the app the user can navigate over to the sign-out button to log off.

3. This use-case is completed when the user logs out successfully

2.4 Constraints

2.4.1 - Time

My partner and I will be diving into technologies we have not dealt with before such as Computer vision, image recognition, human recognition, motion detection(possibly) and human pose estimation. So, because we are starting from scratch with these technologies some aspects might take us a little longer than they would if we already had a previous understanding of the field we are developing in.

2.4.2 - Speed

Project using computer vision usually have a dependency on being fast and reliable in order to be efficient and worth using. If our software cannot analyse postures and relay back useful information almost instantly to the user, then it is not doing its job correctly. The idea is that users should be able to immediately correct their form if they are in danger of doing something potentially damaging. Speed may also vary depending on the quality of the hardware owned by the user.

2.4.3 - Accuracy

There may be constraints on how close the application can guide a user to perfect form based on factors that are outside of the software control such as Flexibility issues, Injuries, Mental blocks and so forth.

2.4.4 – User experience

The initial design is based on some assumptions about what a typical user entering the gym would expect from the gym, so there might be some features that may be more specific to some users that are needed/desired that we won't have initially because of foresight constraints.

2.4.5 - Scope

There are some constraints on fully deciding how much we can fit into the project in the recommended time such as; the number of different exercises we can provide guidance for before the deadline.

The software will be designed to analyse an image of the most important part of a movement (e.g Lower position in squat, Mid position in Deadlift), but is upgrading this to work on video for the full range of motion possible within our scope is an aspect we will have to investigate as the project begins to near completion.

3. Functional Requirements

3.1 Create account

Description: Our system must allow a user to create an account on our system that can store information and progress made by the user.

Criticality: This feature is only critical if the person using the software wants to keep track of personal progress.

Technical issues: We must be able to securely and safely store user emails and passwords.

Dependencies: None.

3.2 Log in

Description: After creating an account a user should be able to use their credentials to log in and use all the features of our software.

Criticality: This is critical if the person using the software wants to track their own progress and use other personal features that may be available.

Technical issues: Implementing adequate security to keep their account safe from intruders.

Dependencies: User creating an account successfully.

3.3 Tutorial

Description: The system must allow users to easily learn how to properly use the software for both safety and efficiency reasons.

Criticality: This is not critical to the overall functionality of the software but is still important for users to complete.

Technical issues: Making sure the tutorial is correct and easy to follow in order to avoid misleading the users.

Dependencies: User logging in.

3.4 Select exercise

Description: The user needs to be able to choose a desired exercise from a list and be able to train their form for that exercise.

Criticality: This is the second most critical feature of our system as it is the main purpose of the software.

Technical issues: Having a list of exercises to choose from.

Dependencies: None.

3.5 Analyse exercise

Description: After selecting an exercise, the system needs to be able to correctly assess and guide to fix form for the selected exercise.

Criticality: This is the most critical feature of our system as it is the main purpose of the software.

Technical issues: Being able to accurately assess user form and correct it.

Dependencies: Selecting an exercise.

3.6 View exercise history

Description: The user should easily be able to track their progress by viewing past attempts at correcting their form.

Criticality: Not critical to the functionality of the app but critical for personalization.

Technical issues: Storing user information in database.

Dependencies: Create Account, Select exercise.

3.7 Add friends

Description: The system should have the ability to add and follow other people that you are connected with (such as a friend that goes to the gym with you).

Criticality: Not critical to the functionality of the app but critical for personalization.

Technical issues: Setting up a social platform.

Dependencies: Create Account.

3.8 View friends progress

Description: The system should have the ability to share and follow other people's progress/history that you are friends with on the app in order to see how they are doing.

Criticality: Not critical to the functionality of the app but critical for personalization.

Technical issues: Setting up a social platform, View exercise history.

Dependencies: Create Account, select exercise, Add friend.

3.9 Sign out

Description: Users should be able to log out of the app when they are finished.

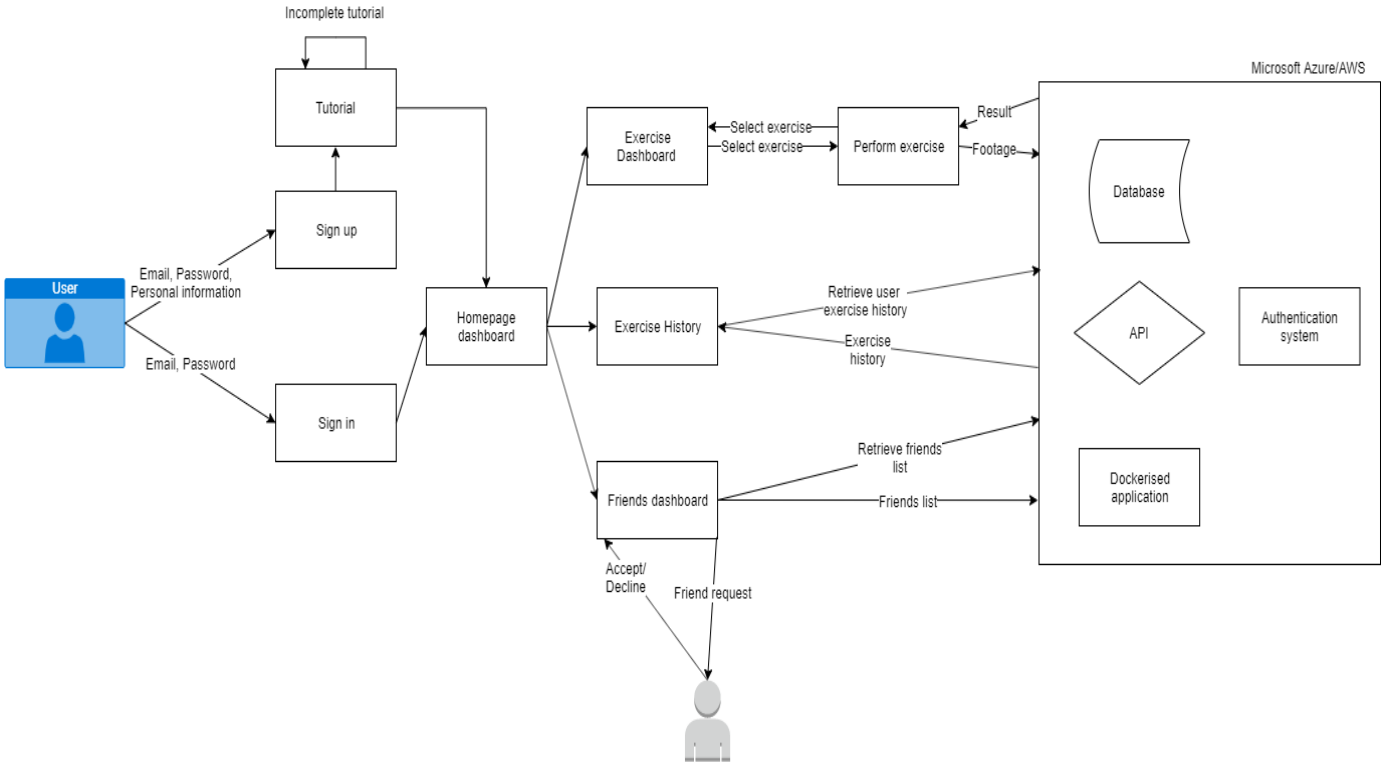
Criticality: Not critical to the functionality of the software but necessary for account safety.

Technical issues: None.

Dependencies: Create Account, Sign in.

4. System Architecture

Figure 4.1 - System architecture diagram



5. High-Level Design

Figure 5.1 – Context diagram

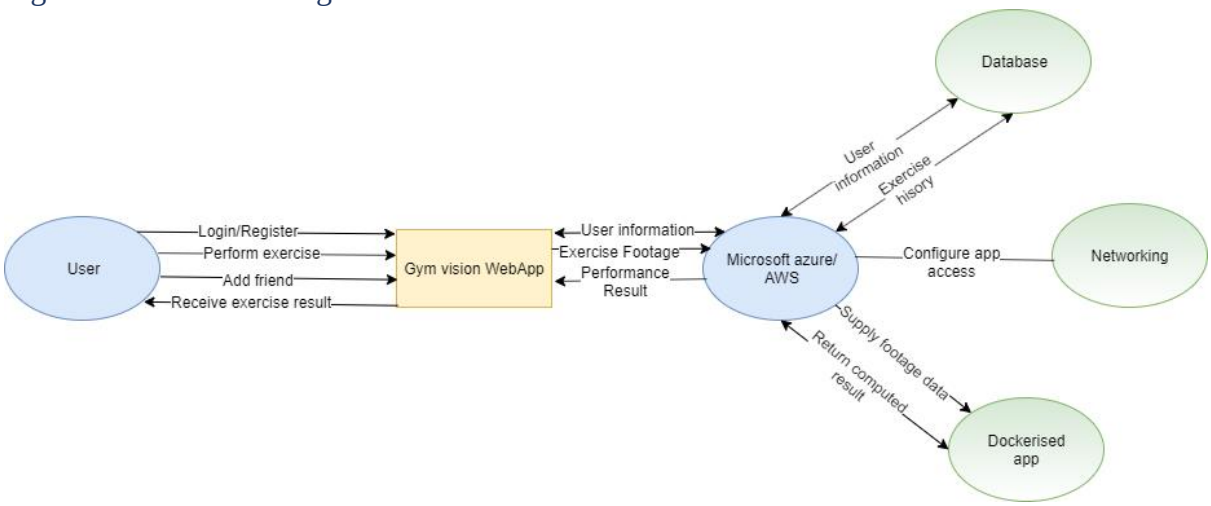
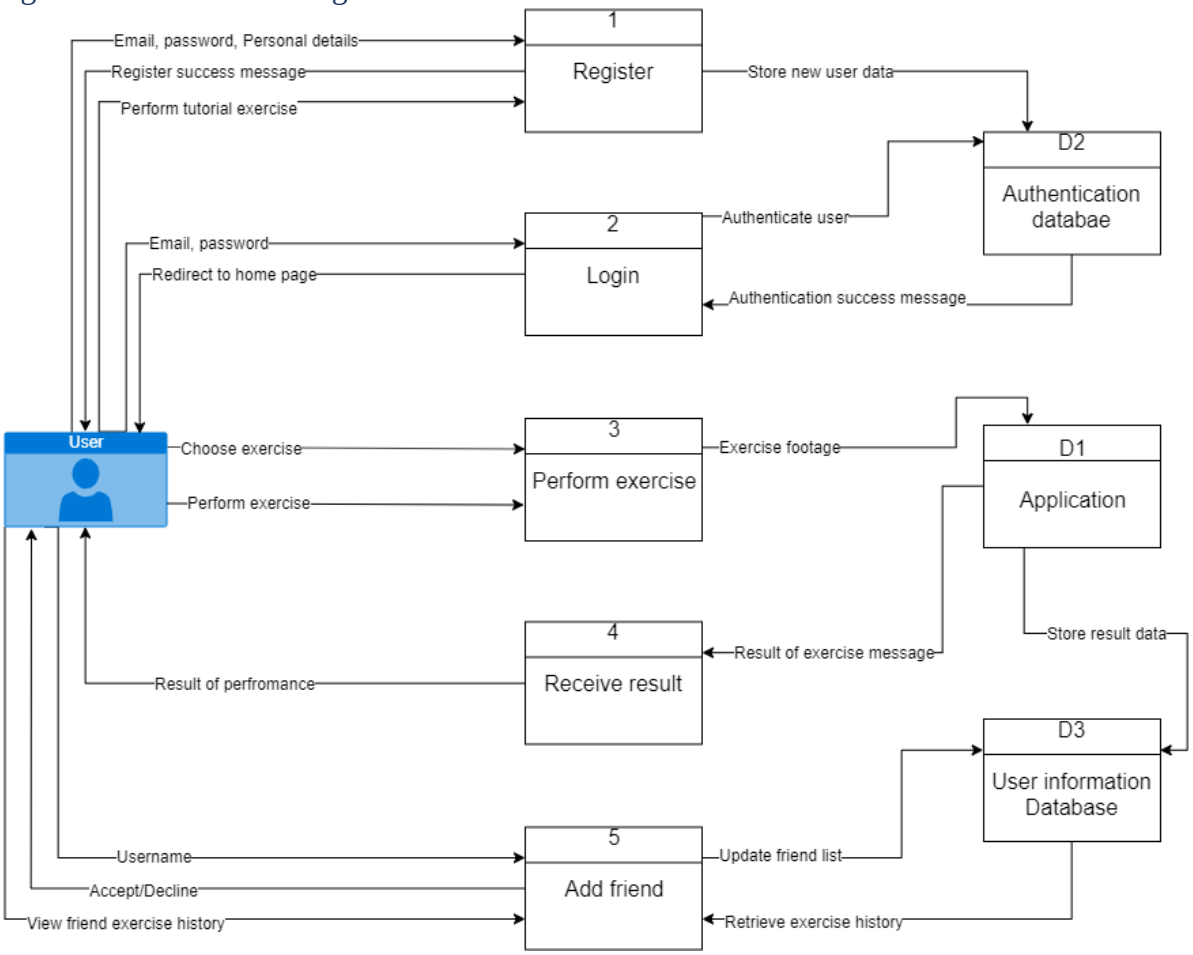
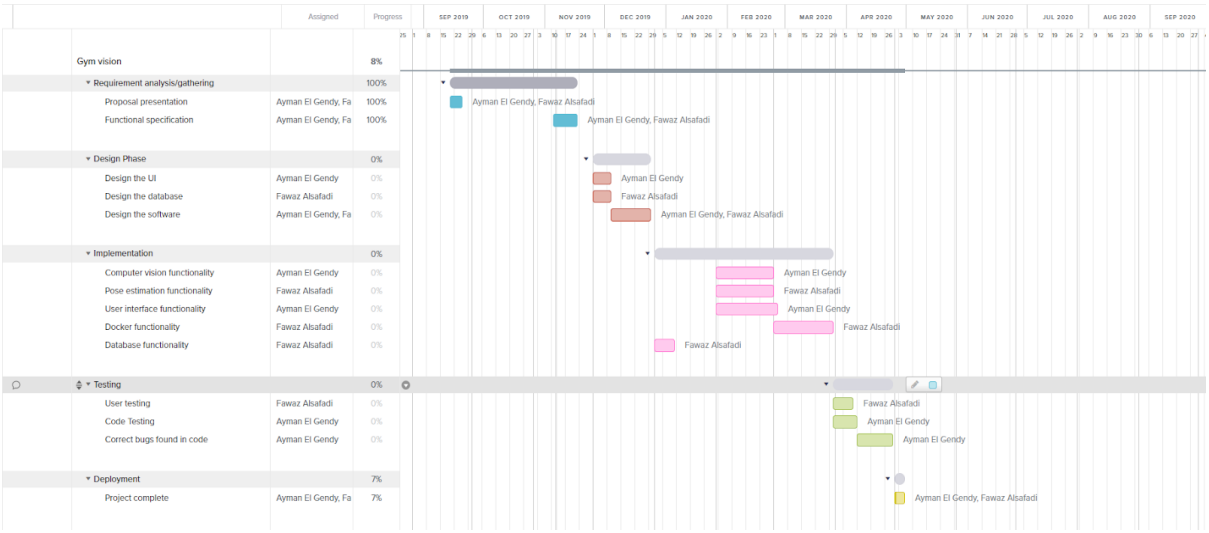


Figure 5.2 – Data flow diagram



6. Preliminary Schedule

Figure 6.1 – Gantt Chart



7. Appendices

- [1] https://opencv.org/
- [2] https://numpy.org/
- [3] https://www.docker.com/
- [4] https://azure.microsoft.com/en-us/
- [5] https://www.webopedia.com/TERM/A/API.html
- [6] https://www.mathworks.com/products/matlab/matlab-and-python.html