Bilkent University

Department of Computer Engineering

Senior Design Project

Gymtor

Final Report

Team Members : Emre Tolga Ayan, Umur Göğebakan, Cemal Arda Kızılkaya, Ömer Faruk Kürklü, Akın Parkan

Supervisor: Selim Aksoy

Innovation Expert: Veysi İşler

Jury Members: Can Alkan, Çiğdem Gündüz Demir

April 30, 2021

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

1 Introduction 4

2 Requirements Details 5

2.1 Functional Requirements 5

2.1.1 Application Requirements 5

2.1.2 User Requirements 5

2.2 Nonfunctional Requirements 5

2.2.1 Efficiency Requirements 5

2.2.2 Usability Requirements 6

2.2.3 Reliability Requirements 6

2.2.4 Security Requirements 6

2.2.5 Supportability 6

3 Final Architecture and Design Details 6

3.1 Overview 7

3.1.1 Mobile Application 7

3.1.1.1 View 7

3.1.1.2 Logic 7

3.1.2 Cloud Database 8

3.2 Application Architecture & Design Choices 9

4 Development/Implementation Details 9

4.1. User Interface & Cloud Database 9

4.2. Pose Detection 9

5 Testing Details 11

5.1 Integration 11

5.2 Testing the UI 12

5.3 Testing Sub-Modules 12

5.4 Image Processing Performance 12

6 Maintenance Plan and Details 12

7 Other Project Elements 13

7.1. Consideration of Various Factors in Engineering Design 13

7.1.1 Public Health 13

7.1.2 Public Safety 13

7.1.3 Public Welfare 13

7.1.4 Cultural Factors 14

7.1.5 Social Factors 14

7.1.6 Economical Factors 14

7.1.7 Environmental Factors 14

7.1.8 Evaluation of Factors 15

7.2. Ethics and Professional Responsibilities 16

7.3. Judgements and Impacts to Various Contexts 16

7.4. Teamwork Details 18

7.4.1. Contributing and functioning effectively on the team 18

7.4.2. Helping creating a collaborative and inclusive environment 19

7.4.3. Taking lead role and sharing leadership on the team 19
7.4.4. Meeting objectives 23
7.4.4.1 Functional Objectives 23
7.4.4.1.1. Application Functionality 23
7.4.4.1.2. User Functionality 23
7.4.4.2 Non-Functional Objectives 24
7.4.4.2.1 Efficiency Requirements 24
7.4.4.2.2 Usability Requirements 24
7.4.4.2.3 Reliability Requirements 24
7.4.4.2.4 Security Requirements 25
7.4.4.2.5 Supportability Requirements 25
8 Conclusion and Future Work 25
Conclusion 25
Future Work 26
9 User Manual 27
Glossary 45
References 46

1. Introduction

Exercising on a regular basis is essential for living a healthy life and one of the best practises that you can turn into a habit to take care of what had already been given to you: your body faced with slowly fading into mold, yet still full of energy and filled with the fresh water from the spring of youth; ready to move, willing to act; ready to prove, waiting to attack.

However, due to the way of living that most of today's society has to pursue, each day each person wastes their time on the way to work, at work and on the way back to home. Unwillingly paying hundreds of dollars to fitness clubs and personal trainers just to make you sweat for 45 minutes and listening to your excuses about how difficult it is to reach out the gym after work at rush hour. So we think that an inactive lifestyle, which is one of the most serious problems of the digital era, could and should be solved by an intelligent digital application: Gymtor, your artificially intelligent fitness mentor.

In order to make exercising accessible to more people, our team came up with the idea of a sports assistant, which is powered by cutting edge deep learning techniques to analyze the body pose of a person and provide relevant real-time feedback to the user about what can be improved about a certain exercise in order to help the users in making the most out of their workouts.

In the development of this idea, we realized that most of the mobile phones that we encounter today are capable of capturing sufficiently high resolution videos and have enough computational power to perform the tasks that our idea presupposes. Furthermore, smart phones have become an integrated essential part of our lives that everybody has always access to in nearly every condition. If we take into account that newly developing smart watch industry, some additional information about the state of the body can be collected and used to extract meaningful information.

In the remaining sections of this report, the development and overall life-cycle of the project will be evaluated. Certain features will be shown and emphasized. Furthermore, the details of the development stage will be explicitly presented and discussed. Finally, the factors that affected the project life-cycle and development will be introduced followed by some certain maintenance issues for the application and a future work section.

2. Requirements Details

2.1.1. Functional Requirements

2.1.2. Application Requirements

- Watch and analyze the users’ exercises.
- Give real time auditory or textual feedback based on the visual analysis of the exercise.
- Create statistics after the exercise plan and display it to the user.
- Warn users not to do exercises without break and not to do overtraining.
- Create overall statistics with all the exercise data and display them to the user when asked.

2.1.3. User Requirements

- Sign up, login and logout using an email and a password.
- Change password.
- See exercise plans, create new ones and modify them.
- Start an exercise plan.
- In an exercise plan, pause the exercise, quit from the exercise plan or skip to the next exercise.
- Enter weight, height, and age information and update them.
- Enter the information of any exercise tool (such as dumbbell) if the current exercise requires one.
- See the statistics.
- See the old exercises and statistics.
- Change the settings of the application.
- Delete account.
- Upload profile pictures.
- Search for other users and exercise plans.
- Add other users as friends.

2.2. Nonfunctional Requirements

2.2.1. Efficiency Requirements

- System response time must not exceed 30 seconds.
- Neural network must generate the feedback for the exercise with a maximum of 5 seconds delay.

2.2.2. Usability Requirements

- Buttons and menus must be seen easily.
- Button and menu names must be clear and understandable.

2.2.3. Reliability Requirements

- The application must be able to detect the angle of the body parts in exercise analysis with at most 5° of error acceptable.
- The application must be able to count the repeats of an exercise with 90% accuracy. In other words it can only miss at most one of the 10 repeats.
- The application must be able to roll-back to a safe state in case of any crashes and the user data must not stay in an unstable condition.
- The application must be available to 10,000 active users at a given time.

2.2.4. Security Requirements

- User data cannot be accessed by a third party application or a person.
- The servers that hold data must use RSA-256 encryption keys.

2.2.5. Supportability

- The application must run on any android device with Android 6.0 or higher.

3. Final Architecture and Design Details

In this section we will describe our architecture. In Figure 1 our deployment diagram can be seen. We do all the computations on the mobile device since our pose detection framework is lightweight and thus gives us zero time latency that would be caused by any computation made on a server communication otherwise.

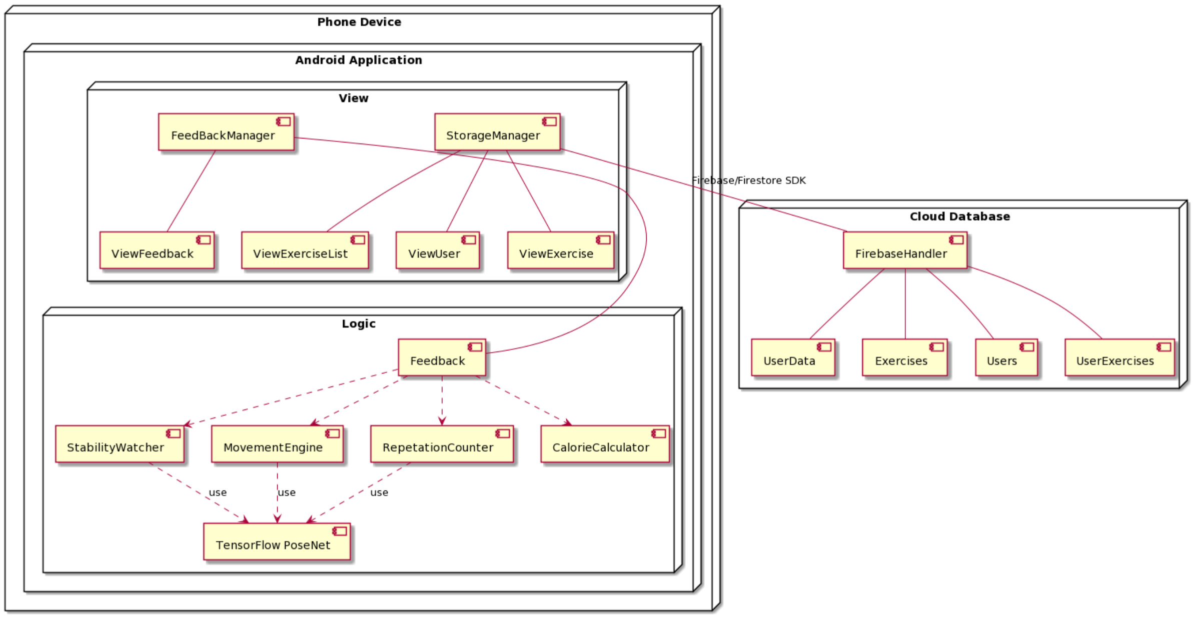


Figure 1. Architecture Diagram

3.1 Overview

Here we will describe the architecture and design overview under two main topics as Mobile Application and Cloud Database:

3.1.1 Mobile Application

We used Android Studio to develop the mobile application. The TensorFlow PoseNet framework was written in Kotlin and thus the Android Studio integration was possible. We have seperate views, view controllers and a logic tier where we do the logical computations regarding pose detection, correctness of movements etc. .

3.1.1.1 View

This node produces the UI an user sees while using our application. It uses FireBase [1] to manage a cloud storage system. It also has a Feedback controller which gets the feedback produced from the Logic tier.

3.1.1.2 Logic

In this node we do the computations. Here we have mainly 4 engines that help to produce feedback on user’s exercise performance.

1. ***StabilityWatcher***
   This component watches joints at all times to ensure they stay in the correct position given an exercise. For example while doing shoulder press exercise, the shoulder joint should stay still. The component uses TensorFlow PoseNet to get joints from the user's camera input and produces feedback when it detects joints are moving if they were not supposed to.

2. ***MovementEngine***
   This component checks the general correctness of an exercise given an exercise. For example while doing shoulder press exercise, the elbow should be 90 degrees while in rest and 170 degrees while on push; the connection between shoulder and elbow joints (upper arm) should be horizontal while in rest and should be as close to vertical while on push. The component also uses TensorFlow PoseNet to get joints from the user's camera input and produces feedback when it detects an error in movement for example the user does not keep their arms at 90 degrees while resting.

3. ***RepetitionCounter***
   Given the user doing the exercise correctly since the other components check the correctness, this component counts the repetition of that exercise. It uses certain thresholds to count repetitions using the pose estimation acquired from TensorFlow PoseNet. It then produces feedback on the count of repetitions.

4. ***CalorieCalculator***
   This component calculates the calories burnt given the exercise, weight and height and sex of the user and then produces feedback to be displayed later on.

3.1.2 Cloud Database

We used FireStore SDK [1] which essentially is a Cloud database system just like any other cloud-hosted real-time NoSQL database. Unlike Firebase Realtime Database, Cloud Firestore is designed for enterprise use, which entails scalability, complex data models, and advanced querying options. Firebase console can be used to view data in both databases. Another mutual point is that there are SDKs for working with server-side code of both databases. These are available for Python, Node.js, Golang, Ruby, PHP, Java, .NET, and C#.

3.2 Application Architecture & Design Choices

As we will describe in more detail in section 4, we decided to use TensorFlow PoseNet API to get pose estimations from video input in real time. This API was designed to be very light and to be used on a mobile device. It was written in Kotlin language thus creating integrability with java and an android application. In our project all our logical implementations are inside app/src/main/java/engine/. Our application first runs our general mobile application with all of its view functions, when the user requests to be monitored while doing exercises we then run the TensorFlow PoseNet application files and switch to an UI viewer integrated with the original posenet files. With the engines and the real time posenet outputs we then produce feedback on that view and display it.

4. Development/Implementation Details

In this section, we provide information about the development and implementation phases of the project. We discuss the reason behind some of the related decisions, and we explain the used technologies.

4.1. User Interface & Cloud Database

We write all the user interfaces majorly in Kotlin. We write some minor parts in Java. We follow the architecture and the design that is explained in the previous section. We store the user information such as email, password, or application-related information such as available exercises in Google Firebase [1]. It is a simple database that integrates well with Kotlin and Java and easy to learn.

4.2. Pose Detection

In the first place, we experimented with simple approaches for pose detection, such as applying edge detection on the image in real-time using OpenCV [2]. It was suitable for a real-time system since it is fast, but edge detection requires different color schemes in the image, which creates a problem when the user wears some clothes that hide the skin. Then, we experimented with convolutional neural network(CNN) based solutions. We implemented and trained simple CNN models for pose detection. Although the pose detection results are promising, they require high computational power that no mobile phone can handle. We tried deploying a model on cloud machines in Amazon AWS [3] and connecting the users to the cloud. Still, this method causes privacy issues, real-time delay issues, and the complexity of the project. Thus, we decided to use a pose detection architecture called PoseNet [4] implemented in Tensorflow Lite [5], a lite version of Tensorflow is suitable for mobile devices. Additionally, Google provides example mobile applications [6] for posenet and pose detection, which helped us increase our know-how.

About Posenet

Posenet model employs a convolutional network which learns to detect individual key points and predict their relative displacements, allowing us to group key points into person pose instances” [4]. The study reports that the study reports a 0.665 average precision value of the popular image recognition boilerplate COCO dataset [4].

We show the architecture of the Posenet in figure 2, which is taken from the original article. A CNN first processes the input image to predict several features of the image. Some of these features are combined to do human pose estimation. The pose estimation prediction then combined with other features to predict the body segmentation. However, we only use the pose estimation part in our project.

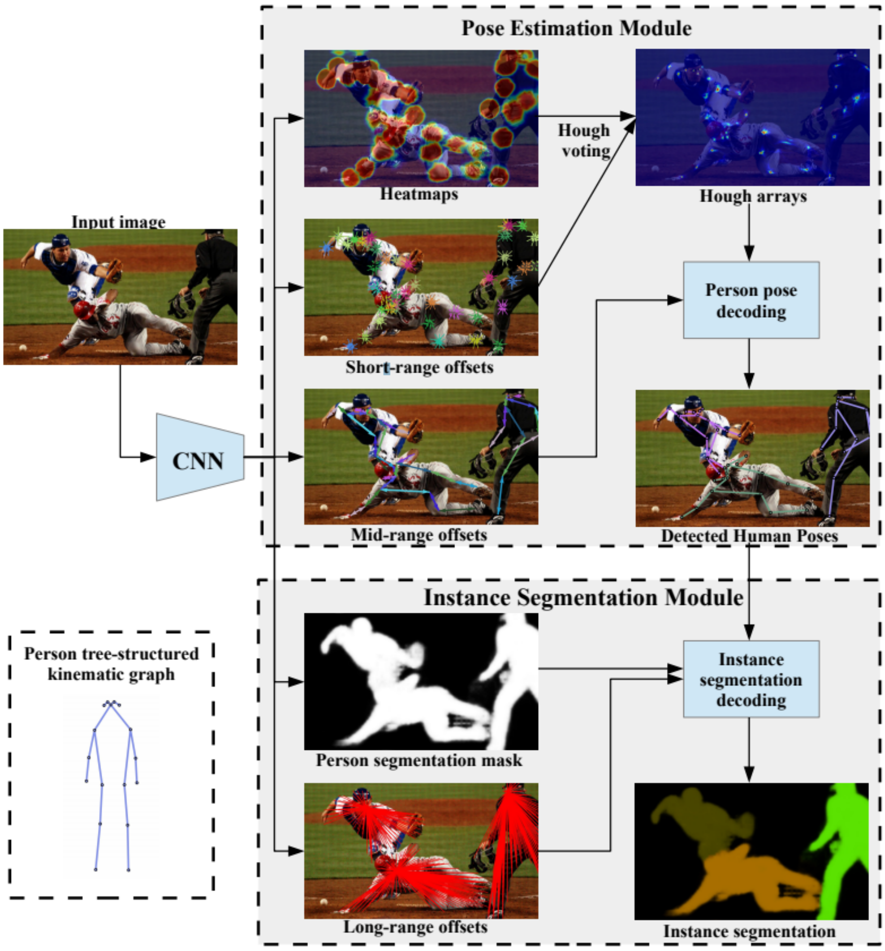


Figure 2: Posenet Architecture [4].

We extract the predicted body joints from pose estimation. Then, we generate exercise feedback using the angles between these joints. We create specific rules for each exercise type based on these angles. We also count the repetition of the exercise using these points. Each exercise has its reference point to measure the repetitions.

The PoseNet and TFLite parts are written in Kotlin since most of the examples of Google for PoseNet [6] are in Kotlin, and we majorly benefit from them. Feedback generation is written in Java for practical reasons.

5. Testing Details

5.1. Integration

Integration is a significant part of the software development process. As a team consisting of five people, everybody has to efficiently contribute to the development. In order to accomplish this goal, we have used GitHub. Especially during the COVID-19 days when physical communication was impossible, we utilized our all development and testing issues on GitHub.

5.2. Testing the UI

Testing UI was an important aspect of our development process. In order to better assess our UI, we have asked our close friends to use our application and give us feedback. As people who are not involved in the development process, they supplied us with beneficial information about the usability of our product. They have also noticed some bugs that were nearly impossible for us to detect in a limited time. This process significantly enabled us to make our UI design better in relatively short time.

5.3. Testing Sub-Modules

We have also used JUnit to test our classes and methods independently before integration. This enabled us to detect bugs before integration with other parts. Therefore, we were able to prevent severe bugs that would be harder to trace back due to the complexity.

5.4. Image Processing Performance

Since Gymtor processes images to build a skeletal model of the user continuously, performance is an important bottleneck. Therefore we tested our skeleton detection algorithm both on CPU and GPU. Our experiments showed that GPU performs four times better than CPU on average which enables us to process each image in less than quarter of a second, letting us increase the number of frames processed in constant time. Thus, we increased the sensitivity of our feedback algorithms and provided a better user experience.

6. Maintenance Plan and Details

The model used for pose detection is open to improvement. We plan to deploy federated learning, which is a method for training local models in user devices. At the same time, no data is shared with the center and only retrieves the trained local models to improve the global model. In this way, the application will perform better as the users continue to use the application.

We are going to add CI/CD pipelines to the project. Any deployment will be automatically checked against errors before merging to the project. We will add unit tests for each module so that we can expand the task easier.

In the future, we will optimize the application for different android versions and different mobile phones, which will improve the user experience. Also, we are planning to update the modules with better-structured, faster codes to make them more maintainable and refactorable.

7. Other Project Elements

7.1. Consideration of Various Factors in Engineering Design

Public Health

Gymtor is a sports assistant app that is directly connected to the body's health. Gymtor must not force users to do exercises that might be harmful to them. Furthermore, it must not motivate them to try harder exercises or to increase the repeats of exercises up to a limit that they cannot handle. Besides, properly doing the exercises also affects user health. Thus, the Gymtor must be able to give accurate instructions to the user and must not drive them to do the exercises in the wrong way which may harm them.

Public Safety

Gymtor will use image processing technologies that require users to allow camera access to the application. The visual materials of the users will be processed by the application and these materials may contain private information such as inappropriate visuals of a person, any material that might reveal the location of the person such as a specific building or banner in the background, or any personal information. In a situation where these materials are accessed from a third party person or organization without permission, this might create a potential safety problem for the users.

Public Welfare

Gymtor will be a free sports assistant application and its motivation is to help people to do exercise at home or wherever they want without paying any money or without going to the gym. Thus, Gymtor may satisfy some users' exercise needs without enrolling in a gym, which eventually benefits their welfare.

Cultural Factors

Gymtor has no major relationship with cultural factors. However, Using a mobile application with machine learning technologies, and getting instructions from it may make some people feel uncomfortable or unusual.

Social Factors

Sports activities are not only a body health matter but also it is a socializing tool for many people. Gymtor's motivation is to allow people to do sports at home, however, this might hinder users from socializing by going to the gym or attending outdoor sports. Thus, it is planned to add a friendship and following mechanism between users so that they will be connected to each other via the application.

Economical Factors

Gymtor will be a free application. However, using a server for the application for the database and for the computational power (if necessary) have a cost. Still, this cost will be minimal at the beginning of the application with fewer users and can be afforded easily. However, in the future, these fees can be afforded with sponsors and possible in-app purchases. Still, this future issue currently is not one of our primary discussions.

Environmental Factors

Gymtor has no major relationship with environmental factors.

Evaluation of Factors

Factor	LVL (/10)	Effect
Public Health	9	Gymtor is a sports assistant app, which is directly related with health.
Public Safety	7	Gymtor uses visual material of users which may contain private data or any data that is a possible threat to users' safety.
Public Welfare	6	Gymtor is a free app that may hinder users from spending money at the gym and motivate them to do exercises at home.
Cultural Factors	2	Only cultural factor of Gymtor for the users is to be assisted by an artificial intelligence which may make them feel unusual.
Social Factors	4	Sports is a way of socializing and this is not possible while doing sports at home, which Gymtor exists for. Thus, the social features of Gymtor will handle social factors.
Economical	3	Gymtor will be free, and any fee in the future are planned to be afforded using possible sponsorships.
Environmental	0	There is no major relationship between Gymtor and environmental factors.

7.2. Ethics and Professional Responsibilities

Gymtor collects data from users such as images or videos, personal information and preferences which are private and needs to be stored or used carefully. Any information that had been captured in that manner must be used without breaking the agreement between user and the application. A detailed and explicit User's Agreement to the Terms and Policies must be proposed to users before using the application and only the users who have accepted the policies should be subject to storage of private information. Our application requires analyzing videos and capturing the most of the information that is possible. Some personal information like weight, sex, age or other physically related information must be tracked. So we must ensure that users trust the application and do not try to give false information.

7.3. Judgements and Impacts to Various Contexts

Judgement	The app will be free to download	
	Impact Level (out of 10)	Impact Description
Impact in Global Context	10	Everyone using an Android device will be able to benefit from the app.
Impact in Economic Context	10	People won't have to subscribe to gyms.
Impact in Environmental Context	6	Less water bottles will be purchased.
Impact in Societal Context	10	People will have a healthier lifestyle.

Judgement	English is the language that is used in the app	
	Impact Level (out of 10)	Impact Description
Impact in Global Context	10	Since English is the most widely used language, most of the world’s population will be able to benefit from it.
Impact in Economic Context	0	-
Impact in Environmental Context	0	-
Impact in Societal Context	0	-

Judgement	Users will be able to create and share exercise plans	
	Impact Level (out of 10)	Impact Description
Impact in Global Context	4	Users can access the exercise plans of other users.
Impact in Economic Context	0	-
Impact in Environmental Context	0	-
Impact in Societal Context	0	-

7.4. Teamwork Details

7.4.1. Contributing and functioning effectively on the team

In order to make sure that every member is involved in the project, we held Discord meetings at least twice a week. The purpose of these meetings was to share our progress on the project and distribute the remaining work to be done among the team members. Also, we actively used Jira to monitor the progress of the project. All the members took an active role on deciding the future of the project, deciding on the features to be implemented, and on general project management. Additionally, the contribution of the members to the final product are as follows:

Emre Tolga Ayan

- Worked on the implementation of the PoseNet.
- Worked on generating the feedback based on the pose estimations.
- Worked on exercise session-related front-end modules.

Akın Parkan:

- Worked on Android-Kotlin development, UI Design, Firestore injection of the project.
- Worked on features related to Social segments of the project including subscribing, unsubscribing to users.
- Worked on creating Exercises, saving exercises and removing exercises from the saved ones.
- Worked on the search page which includes database injections.

Cemal Arda Kızılkaya:

- Worked on the front-end of the application, UI Design, Android-Kotlin Development.
- Worked on the recyclerview feature of the project.
- Worked on updating the user profile.
- Worked on uploading pictures to the database.

Umur Göğebakan:

- Worked on the development of PoseNet detection and giving respective feedback.
- Worked on metrics for evaluating exercises.
- Worked on general flow of the application like counting exercises, repetitions and producing respective results on UI.
- Worked on testing and further improvement on UI.

Ömer Faruk Kürklü:

- Took partition in the logical department and contributed to the engine.
- Developed engine functions that detect the correctness of a movement with angle and location of the joints as metrics.

7.4.2. Helping creating a collaborative and inclusive environment

We discussed the project and our progress in regular meetings every two weeks. In these meetings, everyone shared their ideas and opinions regarding the technologies to use, features to add or remove, or views about the project management. In these meetings, we created unit tasks and played scrum poker to assign story points to these tasks. After this step, we shared the duties such that everyone took a task that they are eager to do. We consider the story points such that every member of the team will get a similar workload. Thus, from deciding the features and creating the tasks to the stage of sharing them and finishing, all the members affected the progress.

7.4.3. Taking lead role and sharing leadership on the team

During the development of the project we have used different communication tools such as jira. For using the time more efficiently we have shared the responsibilities. Since our project requires collaborative working and for a better utilisation of time, these responsibilities are assigned to members then who leads these tasks. Whenever a question related to other tasks arised by a member, these task leaders share their knowledge with others and tell the members about feasibility of the task and if there exists a better alternative these leaders guide the members accordingly. The diagrams related how do the tasks are assigned to leaders can be seen below:

WP	Task Title	Leader	Members Involved
WP1	Documentation	Emre Tolga Ayan	All Members
WP2	Frontend Development	Umur Göğebakan	Ömer Faruk Kürklü
WP3	Backend Development	Ömer Faruk Kürklü	Akın Parkan, Emre Tolga Ayan
WP4	Machine Learning Development	Cemal Arda Kızılkaya	All Members
WP5	Testing	Akın Parkan	Cemal Arda Kızılkaya

WP1: Documentation
Start Date: 01-10-2020 **End Date:** 31-04-2021
Leader: Emre Tolga Ayan **Members Involved:** All Members
Objectives: Preparing necessary reports and product documentations in order to keep the project under control and design the project beforehand. The documentations such as high-level and low-level design reports will be the source documentation while working on the end product.
Sub-Tasks: **Task 1.1:** Writing High-level design report **Task 1.2:** Writing Low-level design report **Task 1.3:** Writing Final Report **Task 1.4:** Preparing necessary presentation documents and slide shows.
Deliverables **D1.1:** High-level design report **D1.2:** Low-level design report **D1.3:** Final report **D1.4:** Final report presentation

WP2: Frontend Development
Start Date: 21-11-2020 **End Date:** 20.01.2020
Leader: Umur Göğebakan **Members Involved:** Ömer Faruk Kürklü
Objectives: Implementing the UI design of Gymtor with user-friendly animations and effects.
Sub-Tasks **Task 2.1:** Researching user behaviors and best practises for UX/UI implementation. **Task 2.2:** Implementation of main page, signin/signup page. exercises page, statistics page and settings page. **Task 2.3:** Implementation of exercise session page.
Deliverables **D2.1:** Main page, signin/signup page. exercises page, statistics page and settings page **D2.2:** Exercise session page

WP3: Backend Development
Start Date: 21-11-2020 **End Date:** 20-02-2021
Leader: Ömer Faruk Kürklü **Members Involved:** Akın Parkan, Emre Tolga Ayan
Objectives: Implementing the connections between the machine learning system, database system and the frontend system. Implementing the UI based backend functionalities of Gymtor.
Sub-Tasks **Task 3.1:** Implementing the frontend based functionalities of Gymtor. **Task 3.2:** Implementing database system. **Task 3.3:** Implementing the connection between machine learning system, database system and the frontend system

Deliverables

D3.1: Database System

D3.2: Frontend Functionality

D3.3: Connections between different systems

WP4: Machine Learning Development

Start Date: 01-01-2021 **End Date:** 01-04-2021

Leader: Cemal Arda Kızılkaya

Members Involved: All members

Objectives: Implementing the product for detecting body gestures, human actions and movements and generating feedback accordingly.

Sub-Tasks:

Task 4.1: Implementing the machine learning based feedback system.

Task 4.2: Integration of machine learning system into mobile platform.

Deliverables

D4.1: ML based feedback system that generates real time feedback based on the visual input.

D4.2: Mobile integration of ML system.

WP5: Testing

Start Date: 01-04-2021 **End Date:** 29-04-2021

Leader: Akın Parkan

Members Involved: Cemal Arda Kızılkaya

Objectives: Testing the UI functionality, connections between different systems and feedback accuracy of the machine learning model.

Sub-Tasks

Task 5.1: Testing UI/UX design for usability

Task 5.2: Testing UI functionality.

Task 5.3: Testing connections between different systems.

Task 5.4: Testing machine learning accuracy in different conditions.
Deliverables **D5.1:** Comprehensive testing results report.

7.4.4. Meeting objectives

In this part of the report the requirements specified will be presented.

7.4.4.1 Functional Objectives

7.4.4.1.1. Application Functionality

- Check if the typed user email and password on the login page exist and matched
- Fetch the related data from the database and illustrate them on the appropriate pages
- Ask users for their permission to access their photo gallery and camera
- Display the photo gallery to users to pick the picture as their profile picture
- Watch and analyze the users' exercises
- Give real time auditory or textual feedback based on the visual analysis of the exercise
- Create statistics after the exercise plan and display it to the user
- Warn users not to do exercises without break and not to do overtraining
- Create overall statistics with all the exercise data and display them to the user when asked
- Display the subscribed users
- Display the created and saved exercises
- Display the other users that are searched by user
- Display the exercises that are searched by user
- Display the update screens to users for their profiles and exercises
- Update the informations to the database

7.4.4.1.2. User Functionality

- Sign up, login and logout using an email and a password
- Change password
- See exercise plans, create new ones and modify them
- Start an exercise plan

- In an exercise plan, pause the exercise, quit from the exercise plan or skip to the next exercise
- Enter weight, height, and age information and update them
- Enter the information of any exercise tool (such as dumbbell) if the current exercise requires one
- See the statistics
- See the old exercises and statistics
- Change the settings of the application
- Delete account
- Upload profile pictures
- Search for other users and exercise plans
- Subscribe to other users

7.4.4.2 Non-Functional Objectives

In this section, non-functional requirements of Gymtor will be discussed in following titles: Efficiency requirements, usability requirements, reliability requirements, security requirements, supportability requirements

7.4.4.2.1 Efficiency Requirements

Now the Gymtor's:

- System response time does not exceed 30 seconds.
- Neural networks generate the feedback for the exercise with a maximum of 5 seconds delay.

7.4.4.2.2 Usability Requirements

Now the Gymtor's:

- Buttons and menus can be seen easily.
- Button and menu names are clear and understandable.

7.4.4.2.3 Reliability Requirements

Now the Gymtor:

- is able to detect the angle of the body parts in exercise analysis with at most 5° of error acceptable.
- is able to count the repeats of an exercise with 90% accuracy. In other words it can only miss at most one of the 10 repeats.
- is able to roll-back to a safe state in case of any crashes and the user data must not stay in an unstable condition.

- is available to 10,000 active users at a given time.

7.4.4.2.4 Security Requirements

Now the Gymtor's:

- User data cannot be accessed by a third party application or a person.
- servers that hold data use RSA-256[1] encryption keys.

7.4.4.2.5 Supportability Requirements

Now the Gymtor:

- can be run on any Android device with Android 6.0 or higher.

7.5 New Knowledge Acquired and Applied

Through the development of this project we have used technologies that we did not have prior experience before. These technologies can be classified as:

- Image Processing
- Android Kotlin Development
- Testing
- Software Development Planning

For the image recognition part we have done a literature review and after finding the A Posture Evaluation System For Fitness Based on Recurrent Neural Networks [7] among various other papers related to posture detection topic, we implemented it using tensorflow lite. For developing mobile applications using Kotlin and Firebase Firestore from various websites that related tutorials can be found such as Coursera and Youtube moreover the documentation websites of these technologies. We have merged the Application part and tensorflow lite parts. After this merge, we have tested the application using Android test espresso framework. For reaching the deadlines, we applied software development planning which was beneficial for us to track which team member works on which part and if the team member requires help from another team member.

8. Conclusion and Future Work

8.1. Conclusion

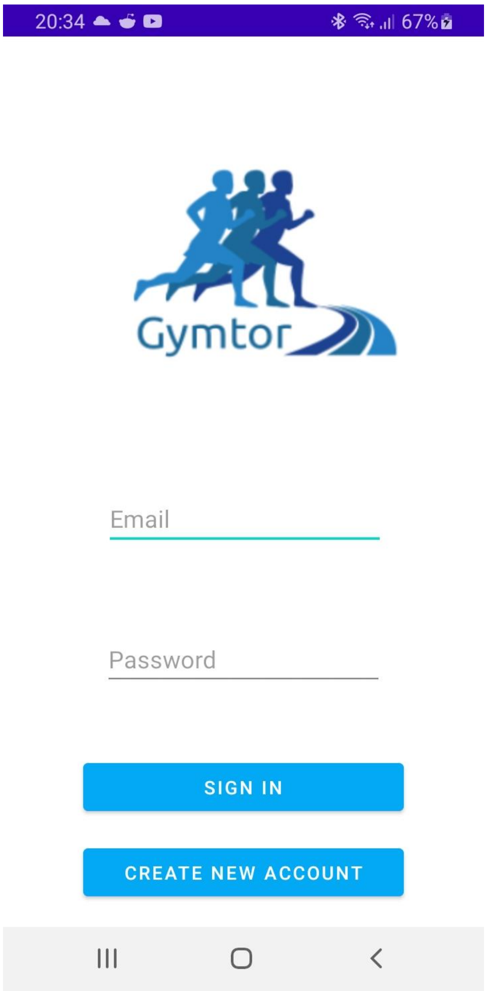
In the current version of Gymtor, we have managed to implement most of the features we had in mind during the requirements stage. The app currently supports a large set of exercises and provides appropriate feedback to the user. The users can create and share exercise plans with others. The only thing we haven't managed to implement is the additional equipment support for exercises due to time constraints. Other than that, we are happy about what we have achieved.

8.2. Future Work

In the future, we will publish Gymtor on Google Play Store and make it free to download. Then, we will update the application according to the user feedback and ratings. These updates may contain new features, bug fixes, and user experience improvements. If the application becomes popular, we might work on developing an iOS version of the application and publishing it on the Apple App Store to make the app accessible to even more users.

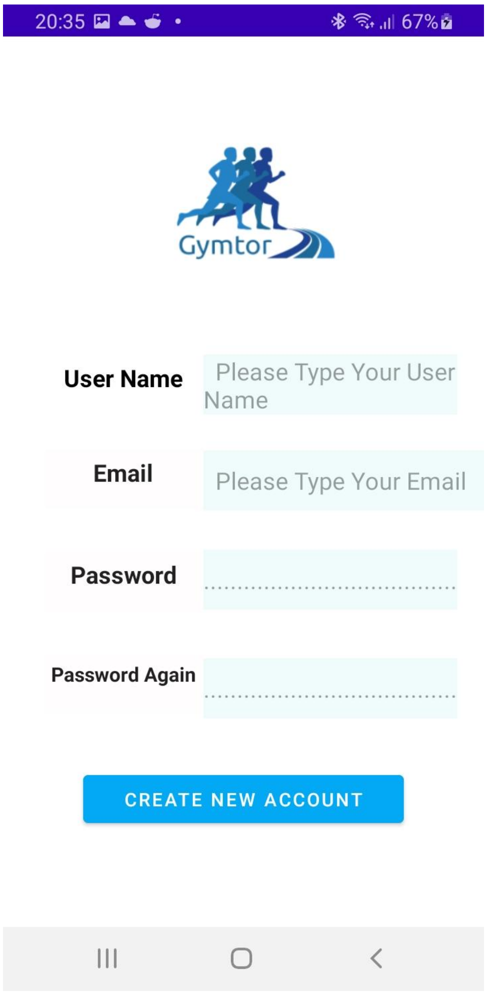
9. User Manual

Sign In Page



When the application is opened, first the sign in page is presented to the user. If the user has an account, he/she can sign in using her credentials. Otherwise, he/she can create an account by pressing the “Create New Account” button.

Sign Up Page



20:35 67%

Gymtor

User Name Please Type Your User Name

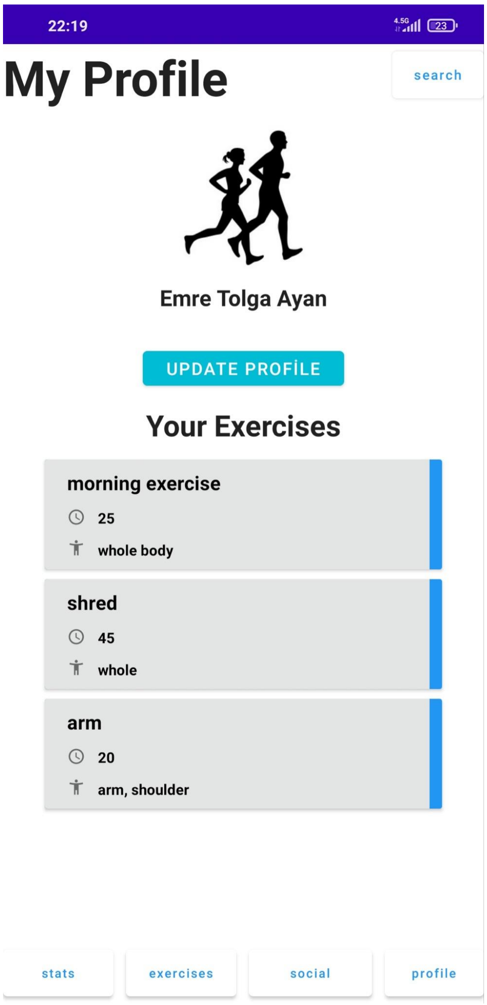
Email Please Type Your Email

Password

Password Again

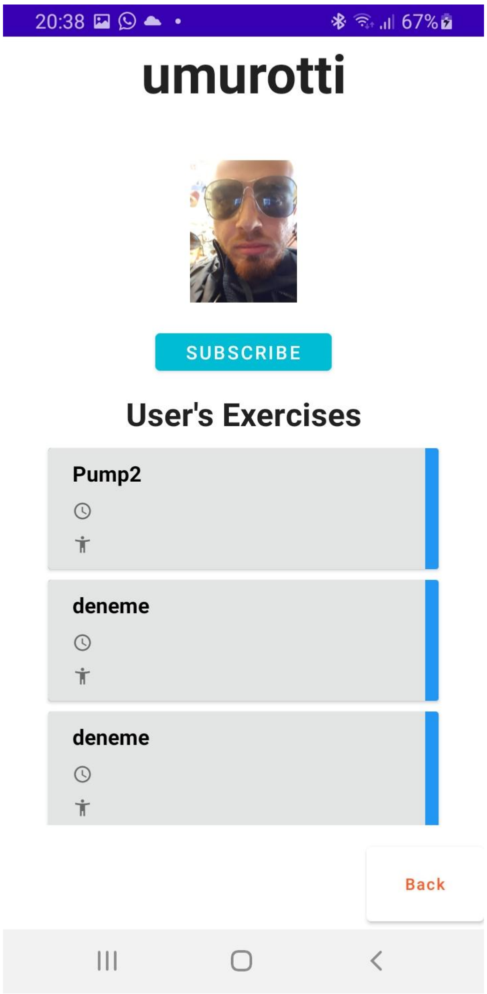
The user can sign up by providing a name, email, and a password.

Profile Page



A user’s profile displays the name, profile image, and exercises of the user. The user can update his/her profile page by pressing the “Update Profile” page. The user can also view an exercise in more detail by pressing on it.

User Profile Page



20:38 • 67%

umurotti

SUBSCRIBE

User's Exercises

Pump2

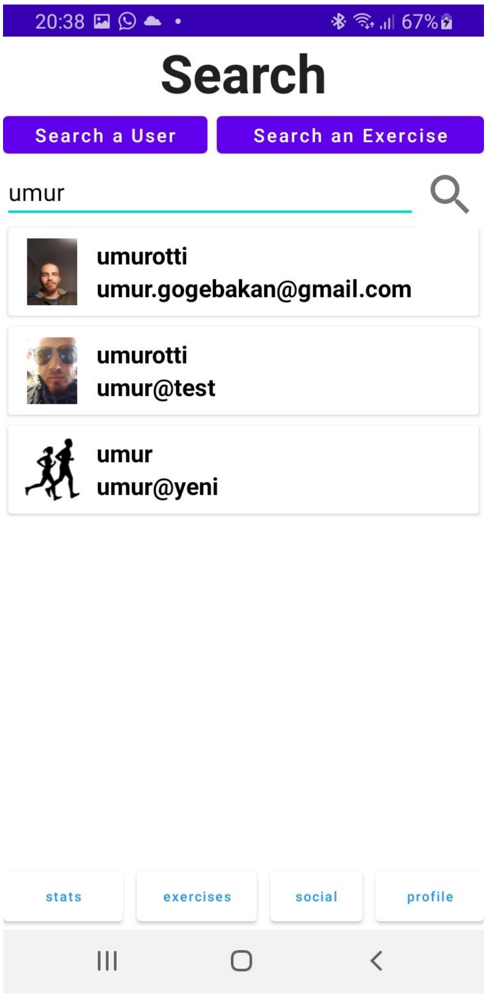
deneme

deneme

Back

A user’s profile page displays the exercises created by that user. The user can subscribe to that user by clicking the subscribe button. The user can also view an exercise of that user in detail by pressing on it.

Search An User Page



umur

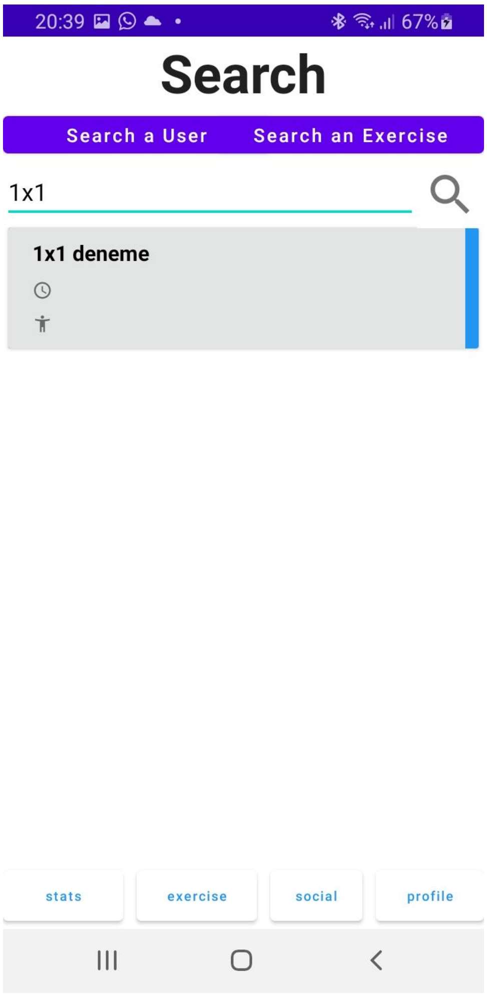
umurotti
umur.gogebakan@gmail.com

umurotti
umur@test

umur
umur@yeni

When the user presses on the Search button, Search an user page is displayed to the user. On this page, user can type the email of a user to search and click on the search button which is illustrated as a magnifying glass. After pressing to search button users whose email starts with the typed text are fetched and displayed to the user. If the user presses on a users view, he/she will be directed to their profile page. User can redirect to the stats page, exercises page, social page, profile page with pressing to the corresponding pages. If the user presses on Search An Exercise button he/she will be redirected to the Search an Exercise page.

Search An Exercise Page



20:39 67%

Search

Search a User

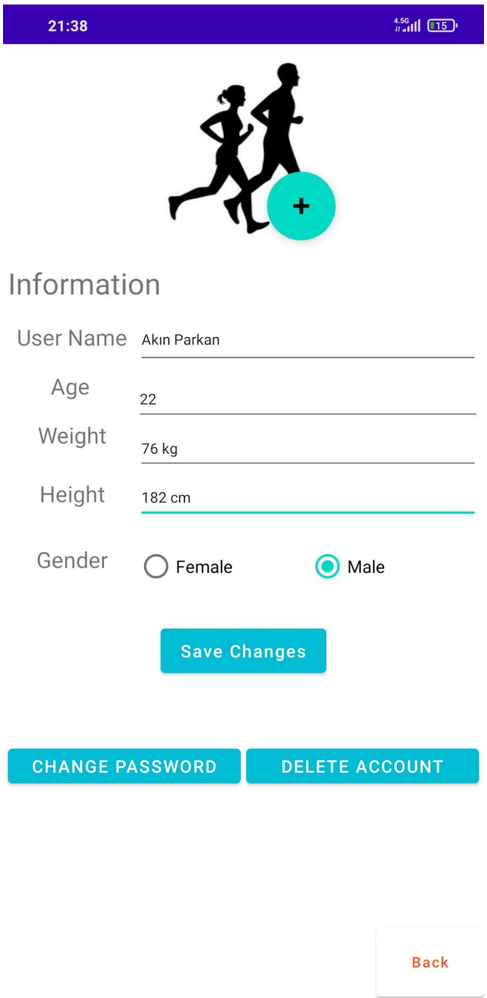
Search an Exercise

1x1

1x1 deneme

When the user presses on the “Search An Exercise” button on Search An User page, Search an Exercise page is displayed to the user. On this page, user can type the name of an exercise to search and click on the search button which is illustrated as a magnifying glass. After pressing to search button exercises whose name starts with the typed text are fetched and displayed to the user. If the user presses on an exercise’s view, he/she will be directed to the Exercise page. User can redirect to the stats page, exercises page, social page, profile page with pressing to the corresponding pages. If the user presses on Search An User button he/she will be redirected to the Search an User page.

Update Your Profile Page



Information

User Name: Akın Parkan

Age: 22

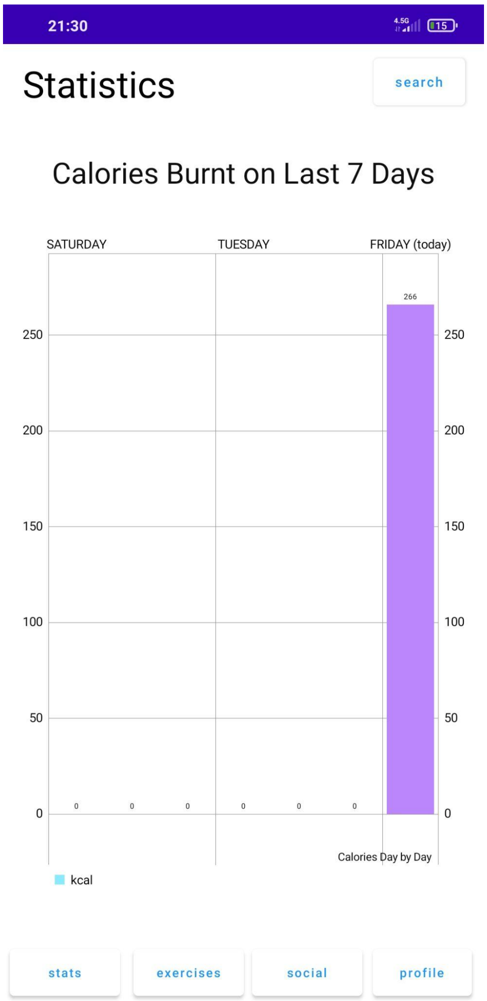
Weight: 76 kg

Height: 182 cm

Gender: ○ Female ● Male

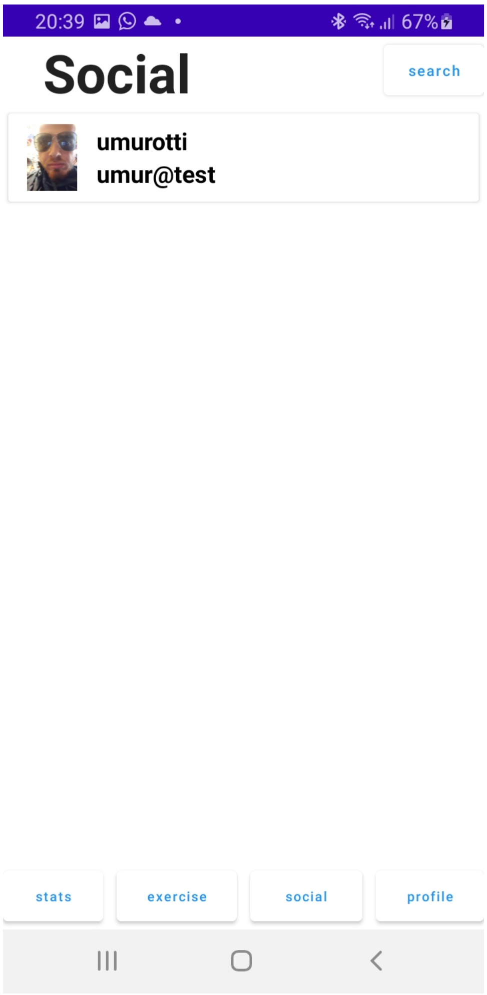
After pressing the Update Your Profile Page button on your profile page, the user is redirected to this page on which an user can update their information. If the user presses the Back button, he/she is redirected to the user's profile page back. If the user presses the Save Changes button, users information is updated on the database. If the change profile photo button is pressed, the user's permission is asked for opening the photo gallery at which the user can pick their new profile picture. If the user presses on Change Password button the user can change their password. If the user presses the Delete Account button, his/her profile is deleted from the database.

Statistics Page



After pressing the Stats button, the Stats page is displayed to the user. At which user can see the amount of kilocalories he/she burnt in the last 7 days. User can redirect to the search page, exercises page, social page, profile page with pressing to the corresponding pages.

Social Page

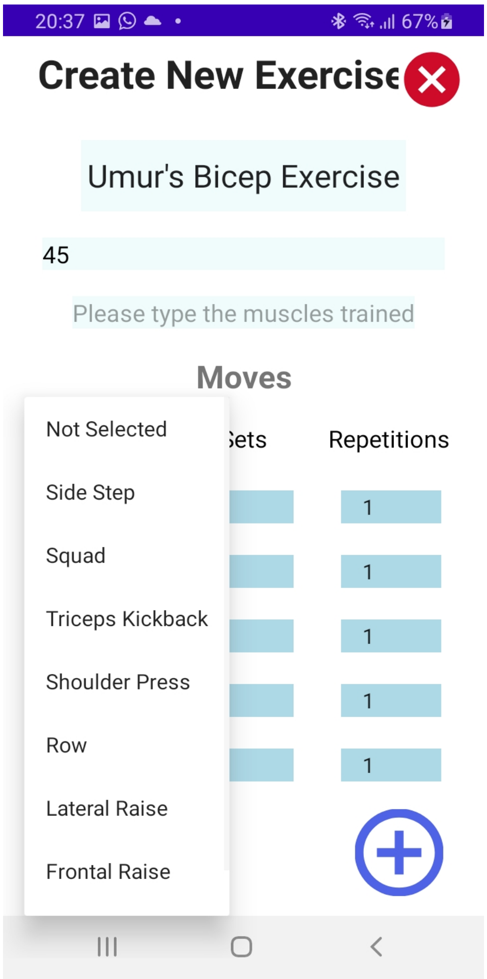


Social

umurotti
umur@test

In this page, the user can see the accounts he or she follows. If the user clicks on the profile, that person’s profile opens. Then, the user can see the exercises of that user, can add for himself/herself and follow that exercise.

Create New Exercise Page



Create New Exercise

Umur's Bicep Exercise

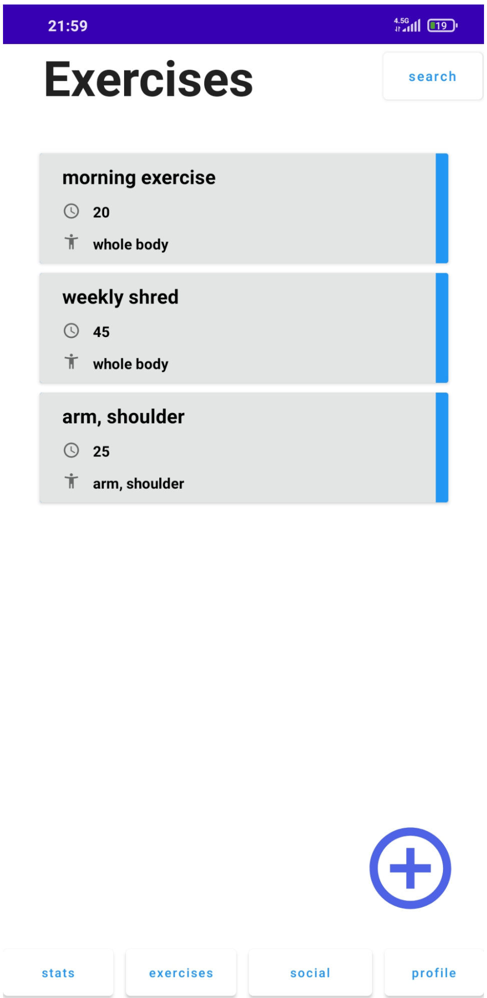
45

Please type the muscles trained

Moves

In this page, the user can enter the name for the exercise plan, the time required for the exercise, and the muscles that are trained in the exercise. Then, the user can select the exercises he or she desires to be in the exercise. The application will assist the user with the selected exercises in the exercise session. The user also must enter the number of sets and repetitions for the exercise.

Exercises Page



Exercises

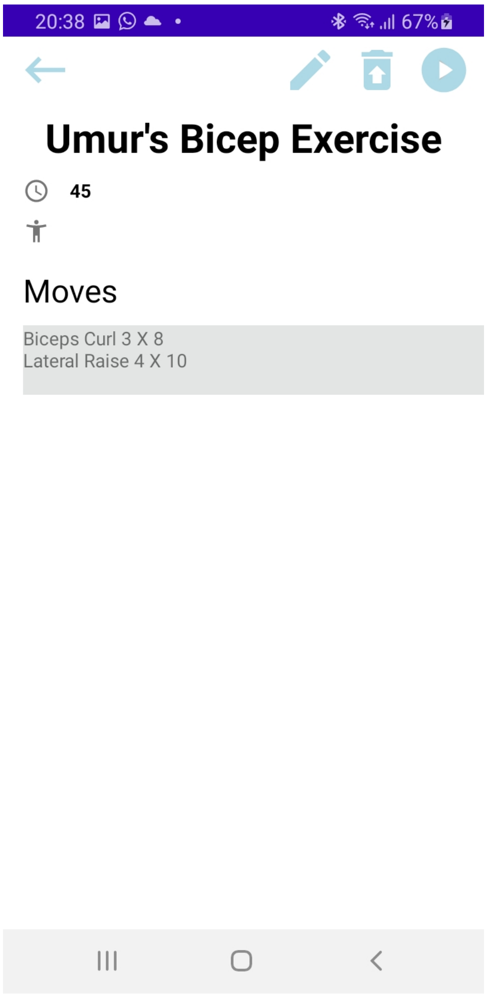
morning exercise
20
whole body

weekly shred
45
whole body

arm, shoulder
25
arm, shoulder

In the exercises page, the user can see his or her current exercises, and their info such as the duration and the focused muscles. The user can click on the plus button to create new exercises. If the user clicks one of these exercise objects, the exercise page opens and the user can start an exercise session.

Exercise Page



20:38

67%

Umur's Bicep Exercise

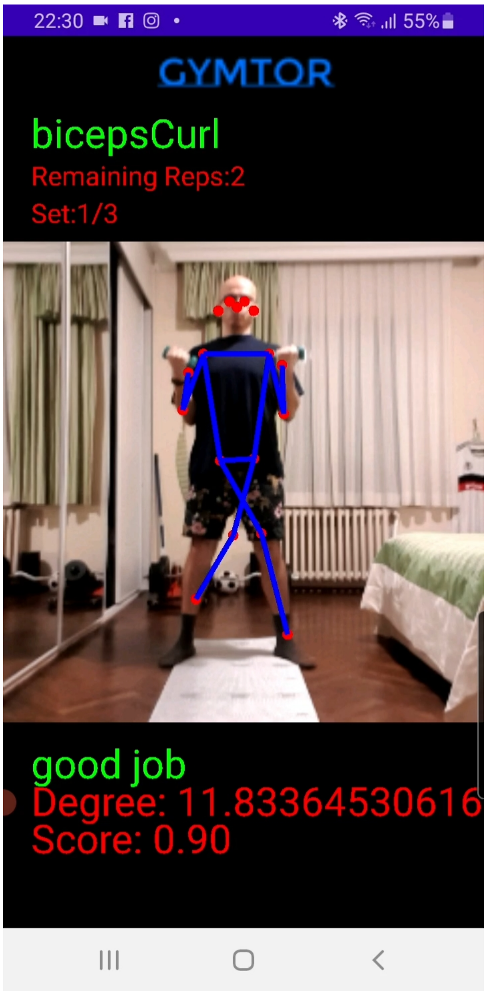
45

Moves

Biceps Curl 3 X 8
Lateral Raise 4 X 10

In this page, the user can see the details for the selected exercise. The user can change the settings for the exercise by clicking on the modify button which appears as a pencil icon, or the user can delete the exercise by clicking on the trash can icon. If the user clicks on the start icon, the exercise session will start.

Single Successful Repetition



bicepsCurl

Remaining Reps:2

Set:1/3

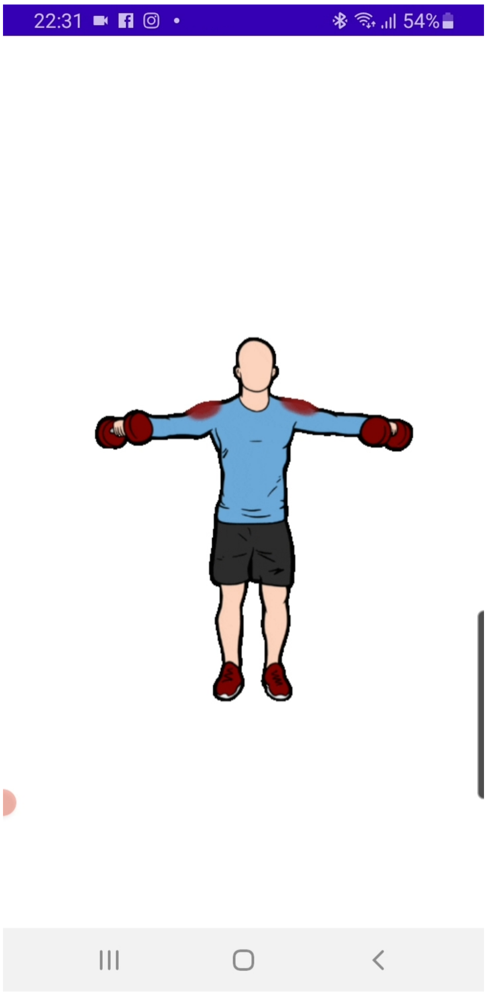
good job

Degree: 11.83364530616

Score: 0.90

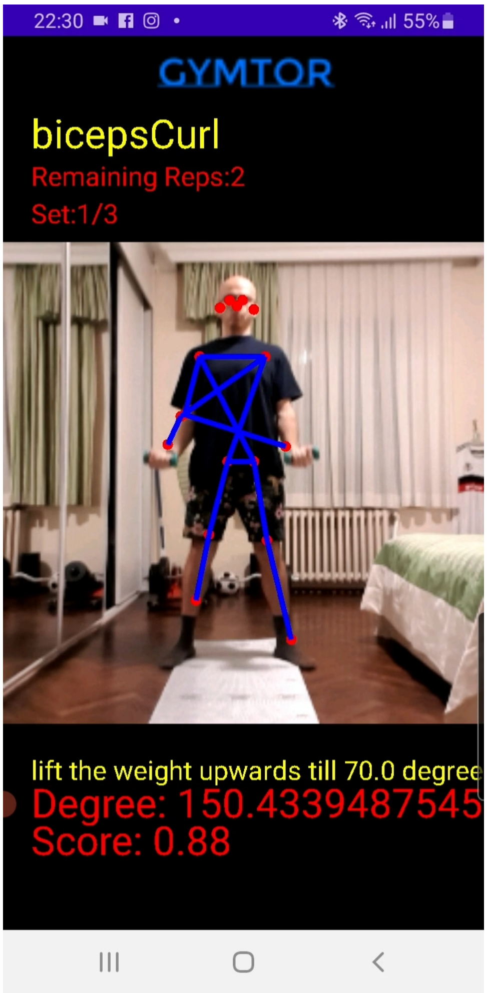
This page shows the input of the camera back to the user with feedback produced by our application. Here a correct completion of a single repetition for biceps curl exercise can be seen where the feedback “good job” is displayed in green color and the angle at the elbow is shown. Name of the exercise, remaining repetitions and the current set is also displayed at top of the page.

Exercise GIF Between Sets



This page shows the next exercise to be done between sets. GIFs of exercises are displayed for showing users the right way of completing the repetitions of exercises. This page automatically directs the user to the screen where the user performs the exercise in front of the camera.

Feedback for Single Exercise Instance: Biceps Curl



bicepsCurl

Remaining Reps:2

Set:1/3

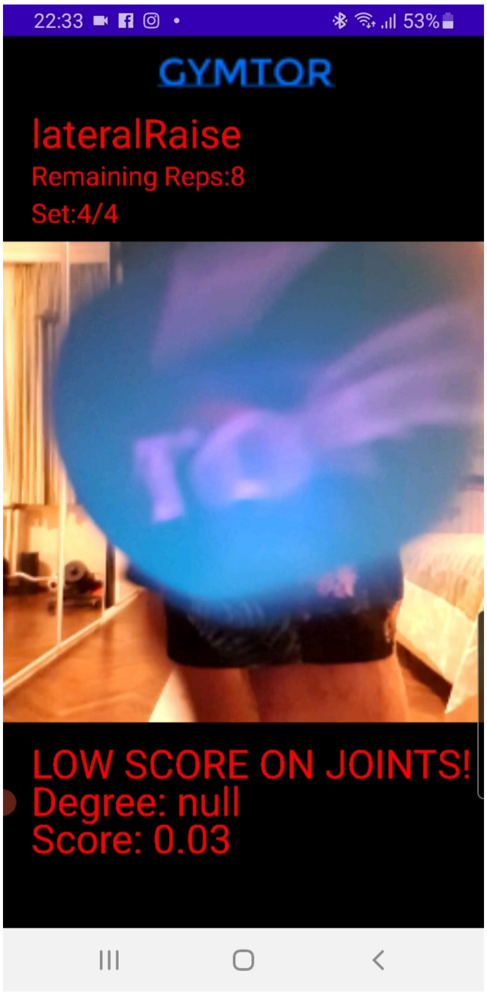
lift the weight upwards till 70.0 degree

Degree: 150.4339487545

Score: 0.88

This page shows the input of the camera back to the user with feedback produced by our application. Here a wrong completion of a single repetition for biceps curl exercise can be seen where the feedback “Lift the weight upwards till 70.0 degrees” is displayed in yellow color and the angle at the elbow is shown. Name of the exercise, remaining repetitions and the current set is also displayed at top of the page.

Failed Skeletal Detection



22:33 53%

GYMTOR

lateralRaise

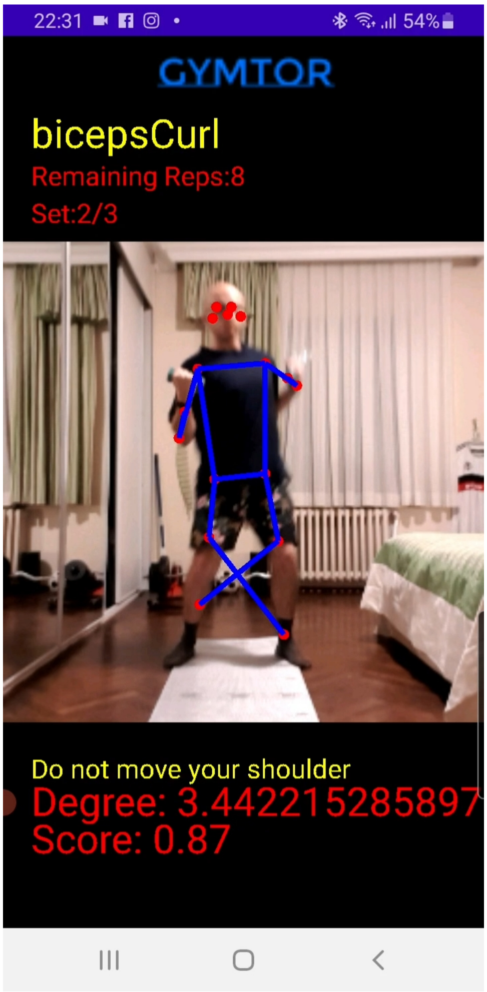
Remaining Reps:8

Set:4/4

LOW SCORE ON JOINTS!
Degree: null
Score: 0.03

This page shows the input of the camera back to the user with feedback produced by our application. Here an undetectable situation due to the environment (light intensity, position of the user etc.) can be seen where the feedback “LOW SCORE ON JOINTS!” is displayed in red color and the angle at the elbow is shown. Name of the exercise, remaining repetitions and the current set is also displayed at top of the page.

Feedback for Single Exercise Instance: Biceps Curl



GYMTOR

bicepsCurl

Remaining Reps:8

Set:2/3

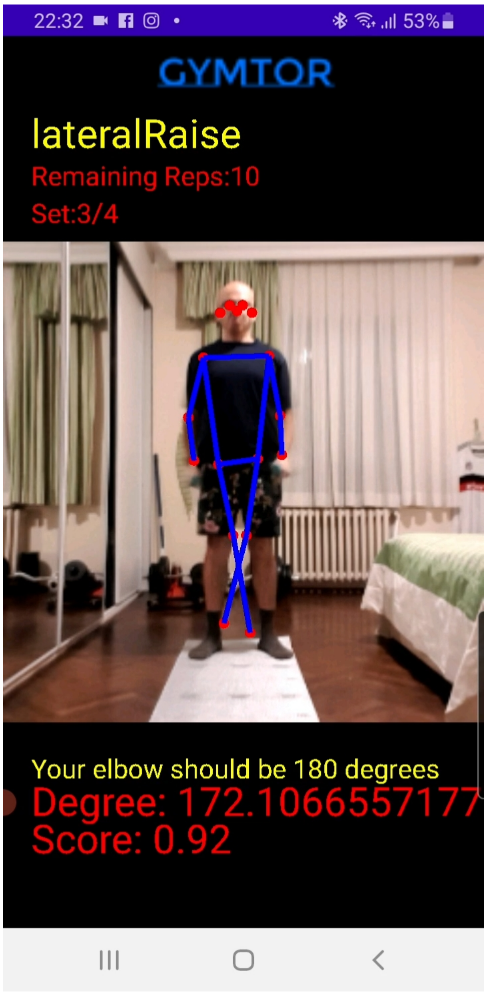
Do not move your shoulder

Degree: 3.442215285897

Score: 0.87

This page shows the input of the camera back to the user with feedback produced by our application. Here a wrong completion of a single repetition for biceps curl exercise can be seen where the feedback “Do not move your shoulder” is displayed in yellow color and the angle at the elbow is shown. Name of the exercise, remaining repetitions and the current set is also displayed at top of the page.

Feedback for Single Exercise Instance: Lateral Raise



22:32 53%

GYMTOR

lateralRaise

Remaining Reps:10

Set:3/4

Your elbow should be 180 degrees

Degree: 172.1066557177

Score: 0.92

This page shows the input of the camera back to the user with feedback produced by our application. Here a wrong completion of a single repetition for lateral raise exercise can be seen where the feedback “Your elbow should be 180 degrees” is displayed in yellow color and the angle at the elbow is shown. Name of the exercise, remaining repetitions and the current set is also displayed at top of the page.

10. Glossary

Cloud Storage: HTTP based request responder to retrieve information. Online and can be reached through the network.

Client: HTTP based request sender. Sends online requests to the backend to retrieve information as a response.

Movement Ground Truth: List<Float> of 3D point cloud sparse matrices to represent the accurate movement pattern of an exercise.

Machine Learning Model: Trained models to produce certain outputs from a given output according to the learnt/estimated parameters during the training phase.

JSON: JavaScript Object Notation as a lightweight communication format to store and transfer data.

GPU: Graphics Processing Unit. Designed for parallel processing. GPUs incorporate an extraordinary amount of computational capability, they can deliver incredible acceleration in workloads that take advantage of the highly parallel nature of GPUs, such as image recognition. Many of today's deep learning technologies rely on GPUs working in conjunction with CPUs.

Neural Network: A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates [8].

UI: User Interface. The intermediate visual module that connects the application functionality and the user interactions.

CNN: Convolutional Neural Networks, it is a class of Neural Networks that are used generally with images

11. References

[1] “Google.” [Online]. Available: https://firebase.google.com/. [Accessed: 30-Apr-2021].
[2] *OpenCV*, 26-Feb-2021. [Online]. Available: https://opencv.org/. [Accessed: 30-Apr-2021].

[3] C. Molter, “Amazon AWS,” *Amazon*, 2002. [Online]. Available: https://aws.amazon.com/tr/. [Accessed: 30-Apr-2021].

[4] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization,” *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[5] “TensorFlow Lite: ML for Mobile and Edge Devices,” *TensorFlow*. [Online]. Available: https://www.tensorflow.org/lite. [Accessed: 30-Apr-2021].

[6] Tensorflow, “tensorflow/tfjs-models,” *GitHub*. [Online]. Available: https://github.com/tensorflow/tfjs-models/tree/master/posenet. [Accessed: 30-Apr-2021].

[7] A.-L. Liu and W.-T. Chu, “A Posture Evaluation System for Fitness Videos based on Recurrent Neural Network,” *2020 International Symposium on Computer, Consumer and Control (IS3C)*, Apr. 2021.

[8] J. Chen, “Neural Network Definition,” *Investopedia*, 23-Dec-2020. [Online]. Available: https://www.investopedia.com/terms/n/neuralnetwork.asp. [Accessed: 30-Apr-2021].