



Bilkent University

---

# Senior Design Project

*Project short-name: GymFeat - AI Training Coach*

## High Level Design Report

Talha Burak uhadar, Mustafa aęrı Gngr, Aye Ezgi Yavuz, Gonca Yılmaz, Ravan Aliyev

Supervisor: Halil Altay Gvenir

Innovation Expert: Emin Okutan

Jury Members: Can Alkan, iędem Gndz Demir

Dec 27, 2020

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.

<b>1. Introduction</b>	<b>3</b>
1.1 Purpose of the system	3
1.2 Design goals	3
1.2.1. Usability	3
1.2.2. Reliability	4
1.2.3. Accessibility	4
1.2.4. Performance	4
1.2.5. Security	4
1.2.6. Maintainability	5
1.3 Definitions, acronyms, and abbreviations	5
1.4 Overview	6
<b>2. Proposed Software Architecture</b>	<b>6</b>
2.1 Overview	7
2.2 Subsystem Decomposition	8
2.3 Hardware/Software Mapping	10
2.4 Persistent Data Management	10
2.5 Access Control and Security	11
2.6 Global Software Control	11
2.7 Boundary Conditions	12
<b>3. Subsystem services</b>	<b>12</b>
3.1. Client	12
3.1.1. View Subsystem	12
3.1.2. Image Processing Subsystem	13
3.2. Server	13
3.2.1. Application Subsystem	13
3.2.1.1. Authentication Service	13
3.2.1.2. User Service	13
3.2.1.3. Workout Service	13
3.2.2. Database Subsystem	13
3.2.2.1. Authentication Database	14
3.2.2.2. User Database	14
3.2.2.3. Workout Database	14
3.2.2.4. Workout Media Bucket	14
<b>4. Consideration of Various Factors in Engineering Design</b>	<b>15</b>
4.1 Public Health	15
4.2 Public Safety	15
4.3 Public Welfare	15
4.4 Global Factors	15
4.5 Cultural Factors	16
4.6 Social Factors	16
<b>5. Teamwork Details</b>	<b>16</b>
5.1 Contributing and Functioning Effectively on the Team	16
5.2 Helping Creating a Collaborative and Inclusive Environment	17
5.3 Taking lead role and sharing leadership on the team	17
<b>6. Glossary</b>	<b>18</b>
<b>7. References</b>	<b>19</b>

# 1. Introduction

## 1.1 Purpose of the system

GymFeat is a training application for anyone who wants to work out alone without hiring a training coach. The main goals of this application are to encourage people doing exercises at home, to enhance the quality of training at home, and to prevent people from getting injured. In order to facilitate a work-out of high quality, we are offering a personalized training program, along with an AI-Trainer, which will compare your body lines with the ideal ones, and will tell you how you can enhance the quality of the movement.

## 1.2 Design goals

In order for our application to function as planned and be commercially desirable, we decided that the following properties are essential for its software design.

### 1.2.1. Usability

- The user should be able to use the application without logging in, or to keep logged in, to ease usability.
- There should be at most two pages before starting the training, and those pages should be the enrolment page and the initial setup page to generate a special training program.
- After the program is created, the user should be able to start training with only one click. To see the progress log, he should click at most two buttons.
- The user should be able to learn the use of UI with a tutorial video of at most thirty seconds.
- The user should be able to memorize the UI, in average, after fifth usage of the application.
- The user should be able to observe the fitness program along with the other activities within the same calendar, and to start training by clicking on the calendar event.

### 1.2.2. Reliability

- The professional fitness trainers should be consulted to design healthy and reliable fitness programs.
- Real-time movement analysis should be done in high precision, hence, the users should be able to count on the application that they are performing the movement correctly, if it does not show any warning about the enhancement of the movement.

### 1.2.3. Accessibility

- The system should be downloadable for free.
- It should be downloadable from the official website of the project for both iOS and Android, from Google Play Store, or from App Store.

### 1.2.4. Performance

- Screen loading and response times from the server side shouldn't take more than one second.
- The AI trainer will analyze the movements in real-time to give instant feedback.
- Creation of personalized fitness program, and Google Calendar Synchronization should not take longer than three seconds.

### 1.2.5. Security

- The user has the option to use the application without any internet connection and with having stored their progress logs in their local. Additionally, the user is also able to register and use the application from multiple mobile devices, therefore, the information about the user will also be stored in a database. Therefore, the system should provide a secure environment in order to keep the data about the user safe. The data that will be stored in the database are the progress log, e-mail, username, and

password. This data should not be shared by anyone and should be kept according to General Data Protection (GDPR) [1].

### 1.2.6. Maintainability

- The network will have a modular structure, so modules should not be strongly bounded and they should not be affected by changes in other modules.
- In addition to the automated checks about the new code changes, there will be a specified code standard. Each pull request will be checked by the reviewer to increase the maintainability of the code in future changes.

## 1.3 Definitions, acronyms, and abbreviations

- Back-end: The back end of a website consists of a server, an application, and a database [2].
- Computer Vision: The field of computer science that focuses on enabling computers to identify and process objects in images and videos in the same way that humans do [3].
- Deep Learning: Deep learning is the field that deals with imitating the human brain in processing data and creating patterns for decision making [4].
- Front-end: The front-end of a website is the part that users interact with. It is the interactable part of the application [2].
- HTTP: The Hypertext Transfer Protocol is a protocol that allows users to communicate data on the internet [5].
- Machine Learning: Machine Learning, or ML for short, is an application area of artificial intelligence and its goal is to provide the system the ability to automatically learn and improve from learnt data and experience without being explicitly programmed [6].
- UI: It is an abbreviation for User Interface, that is where the human-computer interaction and communication occurs [7].
- TCP/IP: IP, or in other words Internet Protocol, has the core function of delivering packets of information from source to target device. However, IP does not handle

packet ordering, nor error checking, thus, TCP is used in conjunction with IP in order to maintain a connection between the sender and the target and to ensure packet order [8].

## 1.4 Overview

GymFeat is an innovative application, in the sense that it brings a free training coach with one-click, and assesses your movements, helping you improve them. There is already one existing workout app that provides movement tracking [9], however it lacks giving people personalized programs unlike other training apps in the commercial market [10, 11]. Our mobile application offers both, while also keeping track of the overall progress.

Our AI-Trainer will apply advanced models such as neural networks [12] to give feedback of each movement to the user in real-time, and it will count the number of each movement, as it happens with a real trainer. For this reason, we will apply the techniques from sub-domains of Computer Vision, that are Image Processing and Deep Learning [13, 14]. As a result, this project will demonstrate how some practical problems can be addressed with such technologies.

With this project, we are aiming to have a broad audience. Therefore, our application will be executable both for iOS and Android. Because our idea is to encourage everyone to do more exercises at home and, hence, everyone, who has a phone, will be able to download and execute our application.

## 2. Proposed Software Architecture

GymFeat's software architecture is composed of multiple subsystems. In this section, subsystems will be described along with their interactions with each other. Also, hardware/software mapping, the data management, access control security, global software control and boundary conditions of GymFeat will be explained.

## 2.1 Overview

Before looking at the subsystems, first a general overview of the design must be defined. As shown in Figure 1, GymFeat's design consists of a client-server architecture which consists of a client and a server layer. The server has an application layer and a data layer. Application layer is where the logic happens and the data layer is where the information is stored. The client consists of two layers as well, view layer and image processing layer. As it is self-explanatory, the view layer is responsible for showing what the client is supposed to see. Image processing layer is responsible for processing the current image of the user's body position and responsible for deciding if the user's current body position is correct.

A general case study of this design can be explained as follows;

1. User logs in to the GymFeat.
  - a. Client sends a request to the server to check the user's credentials.
  - b. Client fetches user's appropriate data from the server.
2. User creates a new workout.
  - a. Client sends a request to the server to create a workout.
  - b. Application Layer handles the creation of workout.
  - c. Data Layer stores the new workout information.
3. User starts a workout.
  - a. Client fetches the workout data from the server.
  - b. Image Processing Layer analyzes the user's body and alerts the user if they have any form mistakes.

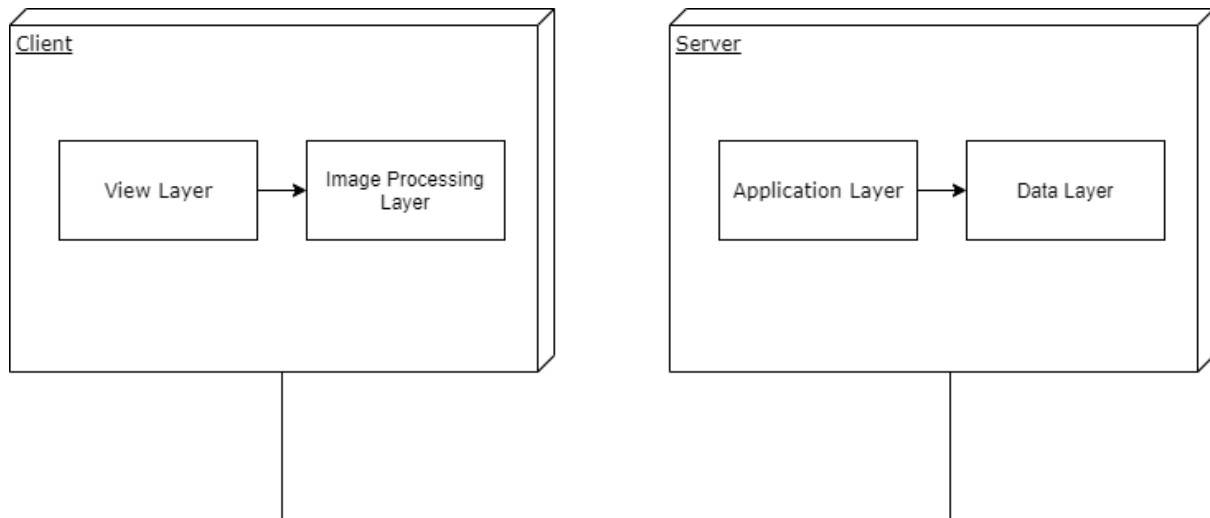


Figure 1: Overview of the Decomposition

## 2.2 Subsystem Decomposition

Details about the layers will be explained here. Since the client's layers do not have any subsystems, only the subsystems of the server's layers will be depicted. From the previous section, GymFeat's server consists of an Application Layer and a Data Layer. Application Layer is where the system does its logic, such as creating/editing a workout, adding/editing user information or logging the user in. As we looked at the use cases of our system, we realized the use cases often fall into 3 different categories. Therefore, we decided to create 3 services (subsystems) in the Application Layer to have a loosely coupled system.

Application Layer consists of 3 subservices, Authentication Service, User Service and Workout Service. A request to the server first hits the Authentication Service to make sure if the user is already logged in, or the user's credentials are correct to log in. Then, depending on what the user wants, the Authentication Service communicates with either the User Service or Workout Service to complete the user's request. Any request about user data (such as editing a profile) will be directed to the User Service. Any request about workouts (such as creating/editing a workout, completing a workout) will be directed to the Workout Service. Finally, requests about creating a new user will be handled by the Authentication Service.

As seen from Figure 2, the services communicate with databases in the Data Layer. Data Layer consists of 3 databases and a bucket storage. User Database is where user information



is stored. Authentication Database is where the user login information is stored. Workout information (when a workout is created, how long it is, what exercises it consists of) is stored in the Workout Database and media (videos, gifs of exercises) about workouts are stored in the Workout Media Bucket.

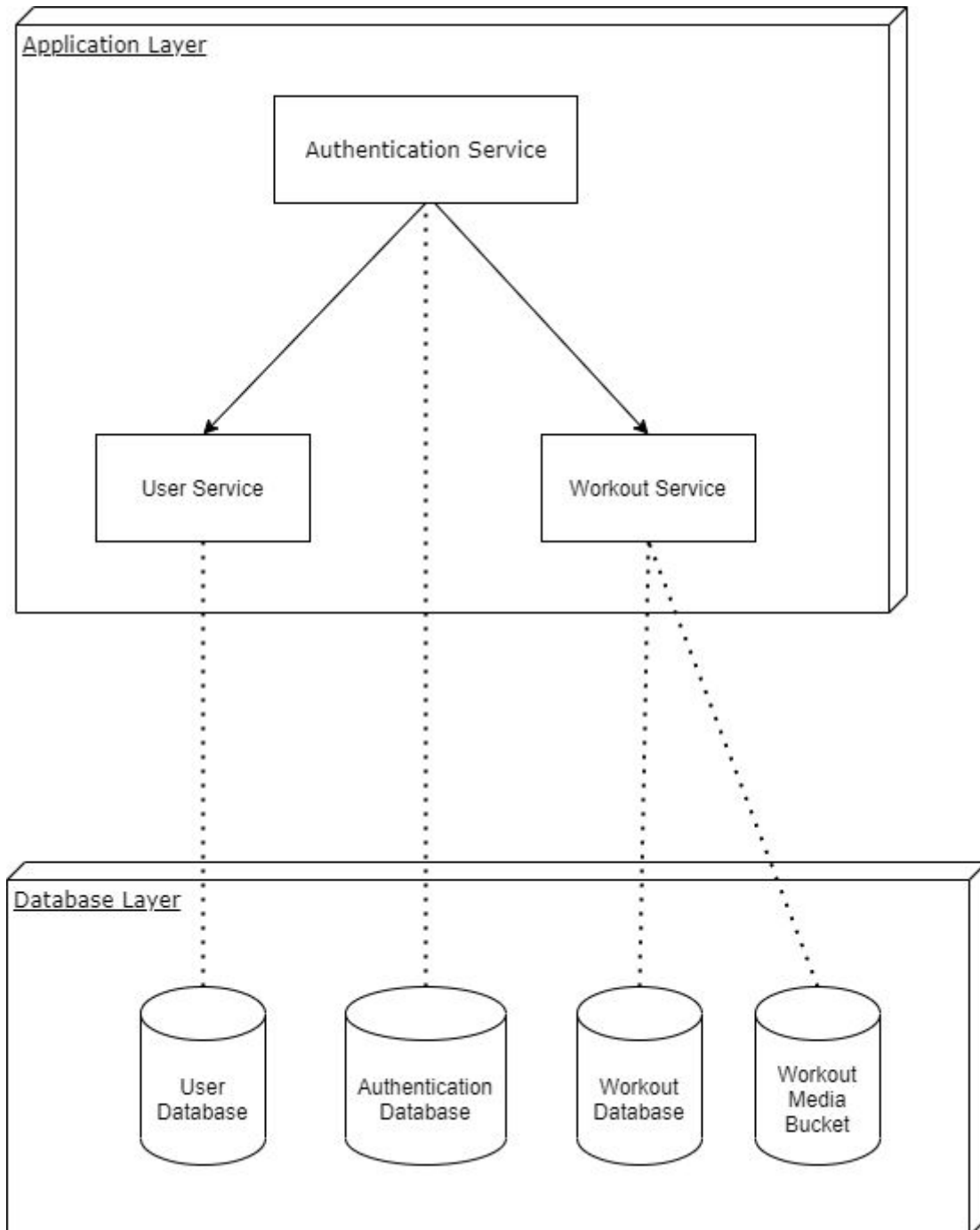
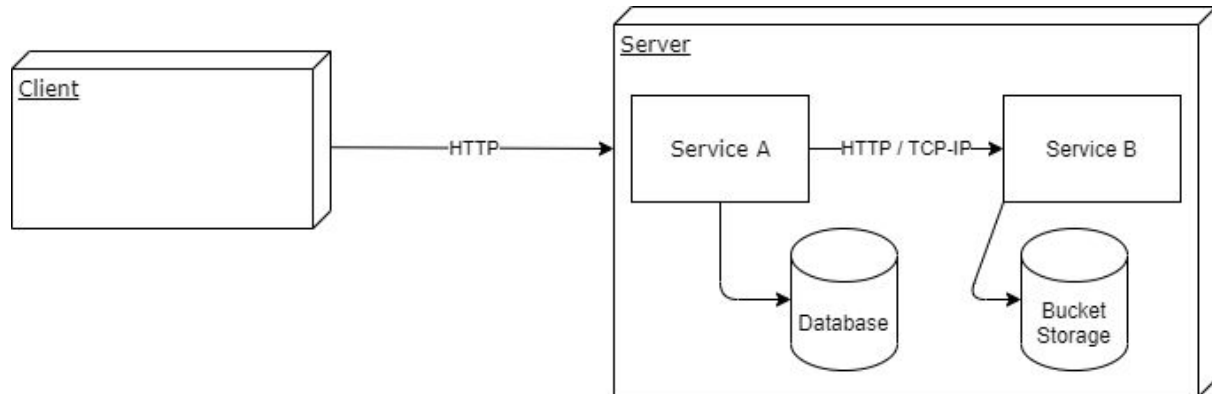


Figure 2: Subsystem Decomposition of GymFeat

## 2.3 Hardware/Software Mapping

GymFeat will run on mobile devices. So, about the hardware mapping of the client side, Client side of GymFeat will be compatible with Android and iOS devices. GymFeat will use the smartphone's resources like camera, processor, memory etc. Also, the client will communicate with the server via HTTP requests.

On the server, different softwares will be run on different services, therefore, these services will have to communicate with each other. Depending on the implementation, these services will communicate via HTTP or TCP/IP. There will also be databases and a storage system on the server. The services will communicate with the databases and the bucket storage system. The server will use Linux because it is open source and easy to use and quite common. Databases will be SQL and be implemented with MySQL. The Bucket Storage can be either the local file storage of the linux server or we can use an external service like Amazon S3 or Google Cloud Storage [15, 16].



## 2.4 Persistent Data Management

The data will be stored in the Data Layer which consists of databases and a file storage system. These databases and the file storage system will store the following type information.

- **User Database:** Data and metadata about the users. Personal information of the user; name, surname, age, height, weight etc. Metadata information about the user; the last the user was active, when the user's profile was created etc.

- **Authentication Database:** Credentials of users. Their username and passwords will be stored here. Hopefully, not the raw data but the hashed data will be stored.
- **Workout Database:** Information about the workouts along with their metadata will be stored here. Each workout that has been created, which user the workout belongs to, what exercises it consists of. Metadata of the workouts, when the workout is created etc.
- **Workout Media Bucket:** Visual information about the workouts will be stored here. We will use GIFs/Videos to teach the user how to do an exercise.

## 2.5 Access Control and Security

User credentials will be stored in the Authentication Database. To make it safe, even if the database contents are made public by hackers, raw passwords will not be stored. System will hash the passwords and store the hashed versions. Other than that, the requests will be transmitted via HTTPS [17] and not HTTP [5] to prevent man-in-the-middle attacks [18]. Also, we will use appropriate strategies to prevent Cross Site Scripting [19] and SQL Injections [20].

## 2.6 Global Software Control

In this section, the overall global control flow will be described. How requests are handled will be explained.

As described previously, GymFeat's design uses client-server architecture. The process starts with the client making a request to the server. The request is handled on the server. Then, the response is sent to the user. As explained here in three sentences, we can capture this process with 3 different steps.

1. **Request Phase:** This phase starts with the client deciding to make a request to the server. These requests can be anything from creating an account or logging in to the account to adding or removing a workout.
2. **Processing Phase:** This phase is where the request is processed. Here, the server receives the request and starts processing the request. This processing can be editing or adding data in the database.

- 3. Response Phase:** After the request processing is completed, the response of the request should be sent back to the user. The response can either carry the information that the request was successful or not, or it can carry the information of a workout and its visual representation (GIF/Video).

## 2.7 Boundary Conditions

GymFeat will have three boundary conditions. These are; Initialization, Termination, Failure. Initialization is starting the application. Termination is closing the application. Failure is a failure in the application.

- **Initialization:** First, GymFeat should be installed on the user's smartphone. If not, the user starts with downloading the app from the Play Store or App Store. When the user opens the application, internet connection will be needed only for the first time to download the required workout information. The workout information will be stored in the phone for the application to be used without internet connection.
- **Termination:** Termination is when the user closes the application. When the user closes the application, any on-going process will also be terminated.
- **Failure:** Failure can happen either on the client or server side. Failures on the server will be logged to a file. We will later check the logged failures and find a way to fix those. Failures on the client side should be reported to the server side to be stored as well (These failure storage will be ignored in the system design as it is a minor implementation).

## 3. Subsystem services

### 3.1. Client

#### 3.1.1. View Subsystem

In this subsystem the front-end logic and the user interfaces of all of the application will be present. This subsystem will work as the communication channel between the back-end logic, such as authentication, and the user. The main concern of this subsystem is to make the application as user-friendly as possible.

### 3.1.2. Image Processing Subsystem

In this subsystem, the user will be observed in real-time via camera whilst working out. The pose estimation algorithm will detect the body points continuously and the analysis algorithm will interpret the movements of the user from extracted body lines to give real-time feedback during the whole workout. Furthermore, the exercise will be rated after finished according to cumulative body points information.

## 3.2. Server

### 3.2.1. Application Subsystem

In this subsystem, the application's back-end logic will be described within 3 services that are Authentication Service, User Service and Workout Service.

#### 3.2.1.1. Authentication Service

Authentication service is responsible for communicating with the Authentication Database to enable users to make use of functionalities that require any type of authentication, such as registering, logging in.

#### 3.2.1.2. User Service

User service carries out all functionalities related to users. In our application, the user can create a personalized workout program, start a workout, collect scores during the workout, and finish the workout. User service accommodates these functionalities and communicates with the User Database to store these information.

#### 3.2.1.3. Workout Service

Workout service is the fundamental service for bringing all types of functionalities to the user whilst working out. In our workouts, there are introduction videos and certain warnings to enhance the quality of the workout and Workout Service is responsible for retrieving the media and collecting details of the workout.

### 3.2.2. Database Subsystem

GymFeat's database subsystems consist of 4 subsystems of which 3 of them are databases and a file storage system called media bucket.

### 3.2.2.1. Authentication Database

Authentication database is responsible for storing authentication data of the user.

Authentication database will store usernames (might be replaced with emails later) and hashed passwords. Passwords will be hashed in the Authentication Service. Hence, the table this database stores will more or less look like this tuple (username, hashed\_password). For this, we can use either a MySQL [21] database or use a 3rd party key-value database like AWS DynamoDB [22].

### 3.2.2.2. User Database

User database is responsible for storing user data. The user data consists of any data that can be mapped to users except the authentication data of the user. In short, user data can be first name, last name, height, weight, awards, history. Since these data will be mapped from the username, we can use a key-value database like DynamoDB [22] or a SQL database like MySQL [21] for this.

### 3.2.2.3. Workout Database

Workout database is responsible for storing workout data. The workout data can be anything from the workouts itself to the single exercises a workout can contain. Therefore, this database will store the workouts and the single exercises they are related with. Since there will be relations between workouts and single exercises, an SQL database like MySQL [21] will be a better fit here.

### 3.2.2.4. Workout Media Bucket

Workout media bucket will not be a database but it will store media (videos/GIFs) for single exercises. For this, we can use the file system of the linux and store the media directly, or we can use a 3rd party service like Amazon S3 [15]. Locations of these media (file location of the media if linux file storage is used, or URL of the media if S3 is used) will be stored in the Workout Database but the media itself will be stored here.

## 4. Consideration of Various Factors in Engineering Design

There are various limiting factors that must be addressed in the GymFeat. This section of the report aims to address these limiting factors in detail.

### 4.1 Public Health

Main use case of the GymFeat is to help people exercise more often and more safely. Therefore, public health will be increased by GymFeat as people using the app will be exercising more and without any dangers.

### 4.2 Public Safety

One of the main points of the GymFeat is helping people do exercises safely. While not directly related to public safety, with the help of GymFeat, people will be doing exercises more safely than otherwise they would be doing on their own without the help of GymFeat.

### 4.3 Public Welfare

GymFeat, essentially, is a free mobile app with in-app purchases. The in-app purchases are not related to the main use cases of the app therefore in-app purchases can be neglected in terms of welfare. Other than that, one can use and benefit from all of what GymFeat offers by only having a smartphone and a stable internet connection. Therefore, we can say only people at a certain welfare level who have smartphones and internet connection can use GymFeat.

### 4.4 Global Factors

GymFeat will be in English and the app will be the same for everyone. Thus, it can be said that it is a global app which can be used by anyone who knows English. Also, GymFeat will collect user's data to some extent, so, it will be made sure that GymFeat complies with certain data regulations of countries.

## 4.5 Cultural Factors

People from different cultures may want to have different exercises that are unknown to other cultures or to us. GymFeat will try to have a diverse set of exercises to target people from different cultures.

## 4.6 Social Factors

In GymFeat, users will only directly communicate with the AI trainer. Therefore, GymFeat can be said to have no effect in terms of social factors.

	Level	Effect
Public Health	9	Increasing public health by helping people exercise
Public Safety	4	Having people exercise safely
Public Welfare	1	People need smartphone and internet connection to use the GymFeat
Global Factors	1	GymFeat will comply with data regulations
Cultural Factors	1	Different cultures will be considered for exercises
Social Factors	0	Not affected

# 5. Teamwork Details

In this section, the details of teamwork is discussed and how the proper teamwork will be ensured is explained.

## 5.1 Contributing and Functioning Effectively on the Team

We assume that the key to a successful product is effective team-work and we do not want anyone to be solely responsible for a certain aspect of our project. We therefore strive to circulate duties and responsibilities so that everyone has sufficient knowledge of the sections of the project. Therefore, we first discussed the main roles of each member and how they can collaborate effectively and efficiently in the team. To do so, each member in the project group took part in certain aspects of the project. Since our project includes three main parts for the implementation, we assigned the members to each component in the following way.



- Front-End Engineer(s): Ezgi Yavuz and Ravan Aliyev
- Back-End Engineer: Talha Burak Cuhadar
- Machine Learning Engineer(s): Cagri Güngör and Gonca Yilmaz.

By dividing the project implementation in the aforementioned way, we have obtained an environment, in which everyone can take actions on some aspects of the project and everyone can contribute in the area that they have some prior knowledge about.

As for the reports, a similar outline is followed for the division of labor. For each report, the outline of the project report and the consistency were assigned to one specific member, which changed for each report. The sections were divided among the members equally in order to create an equal environment, which enables each member to function effectively.

## 5.2 Helping Creating a Collaborative and Inclusive Environment

To maintain proper teamwork, the key is to create an environment of collaborative creation, where every participant is involved in most aspects of the project. To ensure this, the team needed an environment in which it was easy to address project management and documentation in a systematic way. Asana and GitHub have been explored for this reason, and because everyone already has a GitHub account, the team has agreed to use it as a project management tool for the implementation part.

For the rest of the project deliveries, Asana is used as a project management tool and the leader of the current package discussed the work that needs to be done with each member and assigned those tasks in Asana to keep track of all tasks, which are in queue, in process and completed. The leader is responsible for giving a detailed explanation of each task and each member who was assigned to a certain task was responsible for updating the status of that task.

## 5.3 Taking lead role and sharing leadership on the team

In this project every member of the team has been appointed leader of a certain job kit. Using this strategy, we will ensure that all of the team members will lead in their respective job kits and will have the chance to experience leading a part of the project.

We circulated duties and responsibilities so that everyone shares leadership and that everyone has sufficient knowledge of the sections of the project.

The work package division among the project members is as follows:

- Gonca Yılmaz was responsible for the initialization of the project. Initial planning phase and the requirements specification as well as analysis of those requirements were done under her leadership.
- Ayşe Ezgi Yavuz is responsible for the site research and the beginning of the implementation phase of the project. The implementation specifications, frameworks to be used and implementation planning is done under her leadership.
- Ravan Aliyev is responsible for the implementation of the MVP and the integration of the work done, up to that point, into one working product. The first demo will also be led by him.
- Mustafa Çağrı Güngör as our lead ML specialist, will be responsible for the ML research and analysis by using real time data and integration of that analysis as an implemented algorithm into our application.
- Talha Burak Çuhadar will be the leader of the finalization process. Under his leadership, the final report and the final project demo along with the presentation will be distributed equally among the team members. Before the project demo, the implemented system and the connections should be finalized and tested very thoroughly. Hence, he will also ensure that all of these steps are done properly.

## 6. Glossary

Back-end 5, 12, 13, 17

Computer Vision 5, 6

Deep Learning 5, 6

Front-end 5, 12, 17

HTTP 5, 10, 11

Machine Learning 17

UI 3, 5

TCP/IP 5, 6, 10

## 7. References

- [1] "Official Legal Text," *General Data Protection Regulation (GDPR)*, 2019. [Online]. Available: <https://gdpr-info.eu/>. [Accessed: 27-Dec-2020].
- [2] "Front-End vs Back-End vs Full Stack Web Developers," *Udacity*, 10-Jan-2020. [Online]. Available: <https://blog.udacity.com/2014/12/front-end-vs-back-end-vs-full-stack-web-developers.html>. [Accessed: 27-Dec-2020].
- [3] I. Mihajlovic, "Everything You Ever Wanted To Know About Computer Vision. Here's A Look Why It's So Awesome.," *Medium*, 09-Feb-2020. [Online]. Available: <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e>. [Accessed: 27-Dec-2020].
- [4] M. Hargrave, "How Deep Learning Can Help Prevent Financial Fraud," *Investopedia*, 14-Dec-2020. [Online]. Available: <https://www.investopedia.com/terms/d/deep-learning.asp>. [Accessed: 27-Dec-2020].
- [5] "HTTP: Definition & How it Works: Protocol Support Library," *ExtraHop*. [Online]. Available: <https://www.extrahop.com/resources/protocols/http/>. [Accessed: 27-Dec-2020].
- [6] "What is Machine Learning? A definition - Expert System," *Expert.ai*, 21-Dec-2020. [Online]. Available: <https://www.expert.ai/blog/machine-learning-definition/>. [Accessed: 27-Dec-2020].
- [7] T. T. Contributors, "What is User Interface (UI)?," *SearchAppArchitecture*, 13-Aug-2019. [Online]. Available: <https://searchapparchitecture.techtarget.com/definition/user-interface-UI>. [Accessed: 27-Dec-2020].
- [8] "What is TCP/IP?," *Cloudflare*, 2020. [Online]. Available: <https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/>. [Accessed: 27-Dec-2020].
- [9] "Engadget is now a part of Verizon Media," *Engadget.com*, 2020. [Online]. Available: <https://www.engadget.com/2019-03-21-kaia-personal-trainer-app-iphone-ipad-camera>

-fitness.html?guccounter=1&guce\_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce\_referrer\_sig=AQAAAFEhpVIghX612lHa9E-KTd-Hn8y3gSv6UnHBOWfcT9DvoAC6YQw5HrEYWMI9iWwycAF76rOFYfr4vhMiYqJiVfDJZDejpsxKXecIDn-14oFEtkhou0xqYOAvg6WIwBbmejrIfSun9yvuko-ese0cS0IndNf6oM0McnZm3eO0qG77. [Accessed: 27-Dec- 2020].

- [10] “Best Fitness and Exercise Apps of 2020,” *Healthline*, 2020. [Online]. Available: <https://www.healthline.com/health/fitness-exercise/top-iphone-android-apps>. [Accessed: 27-Dec- 2020].
- [11] “The 7 best free workout & fitness apps for tracking & planning,” *The Sports Edit*, 2020. [Online]. Available: <https://thesportsedit.com/blogs/news/sweat-7-of-the-best-free-workout-apps-for-tracking-and-planning>. [Accessed: 27-Dec- 2020].
- [12] M. A. Nielsen, “Neural Networks and Deep Learning,” *Neural networks and deep learning*, 01-Jan-1970. [Online]. Available: <http://neuralnetworksanddeeplearning.com/chap1.html>. [Accessed: 27-Dec-2020].
- [13] Infrd, “Image Processing With Deep Learning- A Quick Start Guide,” *Medium*, 25-Aug-2019. [Online]. Available: <https://becominghuman.ai/image-processing-with-deep-learning-a-quick-start-guide-38e166340200>. [Accessed: 27-Dec-2020].
- [14] R. Kwok, “Deep learning powers a motion-tracking revolution,” *Nature News*, 30-Sep-2019. [Online]. Available: <https://www.nature.com/articles/d41586-019-02942-5>. [Accessed: 27-Dec-2020].
- [15] N. B. Gerson, “TR,” *Amazon*, 1970. [Online]. Available: <https://aws.amazon.com/tr/s3/>. [Accessed: 27-Dec-2020].
- [16] “Cloud Storage | Google Cloud,” *Google*. [Online]. Available: <https://cloud.google.com/storage?hl=tr>. [Accessed: 27-Dec-2020].
- [17] “HTTPS” *Google*. [Online]. Available: [https://en.wikipedia.org/wiki/Man-in-the-middle\\_attack](https://en.wikipedia.org/wiki/Man-in-the-middle_attack). [Accessed: 27-Dec-2020].

- [18] “Man-In-The-Middle Attack” *Wikipedia*. [Online]. Available: [https://en.wikipedia.org/wiki/Man-in-the-middle\\_attack](https://en.wikipedia.org/wiki/Man-in-the-middle_attack). [Accessed: 27-Dec-2020].
- [19] “HTTPS” *Wikipedia*. [Online]. Available: <https://en.wikipedia.org/wiki/HTTPS>. [Accessed: 27-Dec-2020].
- [20] “SQL Injection” *W3Schools*. [Online]. Available: [https://www.w3schools.com/sql/sql\\_injection.asp](https://www.w3schools.com/sql/sql_injection.asp). [Accessed: 27-Dec-2020].
- [21] “MySQL” *MySQL*. [Online]. Available: <https://www.mysql.com/>. [Accessed: 27-Dec-2020].
- [22] N. B. Gerson, “TR,” *Amazon*, 1970. [Online]. Available: <https://aws.amazon.com/tr/dynamodb/>. [Accessed: 27-Dec-2020].