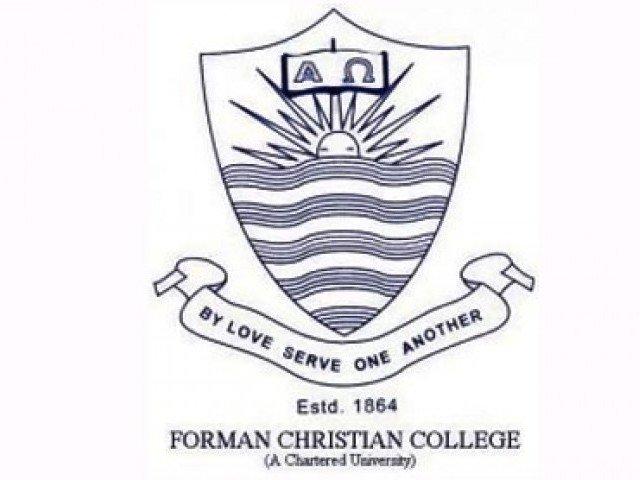
**Department of Computer Science**

**Forman Christian College (A Chartered University**



Presented by:

Roll no: 241546646, 231-487120

Name: Ammar Tariq, Rafay Monam

Body Analysis Tool

# Project Report

# Abstract

Across the world, there are an estimated 185+ million gym memberships. Most of these memberships belong to people who are new or still relatively new to fitness. Out of lack of knowledge and conflicting popular advice, novice fitness enthusiasts often find it challenging to understand where to begin their fitness journeys and what direction to go in, wasting months and years of progress. As such, we developed a mobile application for novice fitness enthusiasts to get a better idea of where to begin and what fitness goals to set for themselves. The ideology is to blend technology with scientifically determined ideal physique ratios to analyze each unique individual’s physique and give them customized feedback on what to improve upon in their body in order to look better. The application is designed using Android Studio and the image processing script is written in Python. In order to detect where the user’s body ends against the backdrop, we implemented MediaPipe. Using Machine Learning, MediaPipe has the functionality of detecting and mapping a person’s individual body parts no matter what position they are in. That feature made it the ideal solution for us. This application would have simple functionality where the user takes well-lit pictures of themselves in light-colored clothing with a black-colored background for contrast, or selects pictures they have already taken from their phone gallery, and submits them to the application for image processing and analysis. The application would then send the images to the backend Python script for analysis and receive the feedback to display to the user. The analysis will tell the user where they currently stand compared to how their physique should ideally look, and what changes they would need to make to their individual body in order to achieve that physique. We have used Firebase cloud database system for database storage, Google Cloud Platform for the image processing python script, and we connected the frontend and the backend via FastAPI.

Keywords: Image Processing, Mobile Application, Health and Fitness, Machine Learning.

# Introduction

**Introduction**

Our project is an Image Processing based Body Analysis Tool. It sets out to provide its users with guidelines as to how they can improve their physique, based on how their current physique compares to the scientifically determined ideal physique ratios. This is a common issue that fitness novices face when they are starting out on their fitness journey. They very often do not know what direction to take their body in - whether they should gain muscle, lose fat, or do both at the same time. We initially took on this project because we ourselves are passionate about health and fitness, and building good looking physiques.

**Objectives**

By the end of this project we would have a robust app that would give users quick and easy access to the insights of how their body looks and how they can improve it. It will efficiently provide the set of exercises best suited for the user based on their visual appearance. Our app will help users throughout their fitness journey.

**Problem Statement**

In a world increasingly conscious of health and fitness, individuals often face challenges in understanding and addressing specific aspects of their physical well-being. Many struggle with formulating personalized fitness plans that effectively target areas of their body requiring attention. The absence of accessible, tailored guidance hinders progress, leading to frustration and a lack of motivation.

To bridge this gap, our project addresses the need for a sophisticated Body Shape Analysis App. This app aims to empower users by employing advanced image processing algorithms to analyze their body shapes. By providing detailed insights into areas that require attention or improvement, the app seeks to guide users towards targeted fitness routines. The challenge lies in creating a seamless, secure, and user-friendly application that not only accurately analyzes body shapes but also offers personalized recommendations in terms of fitness goals. Addressing this challenge will contribute to a more informed and motivated user base, fostering a healthier and more confident society.

**Scope**

The Body Shape Analysis App endeavors to provide a comprehensive solution for individuals seeking personalized insights into their physical well-being and targeted fitness guidance. The scope of this project encompasses the following key aspects:

**Image-Based Body Analysis:**

The app will leverage advanced image processing and machine learning algorithms like MediaPipe to analyze user-uploaded body images.

Key body metrics, including but not limited to waist-to-shoulder ratio and height-to-waist ratio, will be extracted for accurate body shape classification.

**Personalized Fitness Recommendations:**

Based on the results of the body analysis, the app will generate tailored fitness recommendations.

Our app will give exercises according to the results from the image

**Mobile Application Development:**

The primary platform for the app will be mobile devices, supporting Android operating systems.

The app will be developed with a user-friendly interface, ensuring accessibility and engagement for users of varying technical backgrounds.

**Analysis and Feedback:**

The app will strive to provide body shape analysis and instant feedback to enhance user experience.

Users will receive immediate insights into their body shape and personalized fitness recommendations upon image upload.

**Continuous Improvement and Updates:**

The app will be designed with flexibility to accommodate future updates, incorporating new features, fitness trends, and advancements in image processing technology.

Regular updates will ensure the app remains aligned with evolving health and fitness standards.

The scope of the Body Shape Analysis App is designed to deliver a user-centric, technologically advanced solution that goes beyond traditional fitness apps. By combining image analysis with personalized fitness recommendations, the app aims to empower users on their health and fitness journeys.

# Requirements Analysis

**Literature Review**

This literature review explores advancements in body shape analysis using image processing, focusing on two key papers: "Body Shape Analysis via Image Processing" by Yang Jinyan et al. [1] and "Web-Based Digital Image Processing Tool for Body Shape Detection" [2]. The aim is to uncover innovative approaches in accurately analyzing body shapes through 2D images captured by conventional cameras.

Both papers categorize body shapes into archetypes such as thin-long, stout, motile, and female based on front body widths. There is a special emphasis placed on ratios of specific measurements, including height, width, perimeter, bust, waist, hip, and high hip width, enabling precise categorization.

Both papers extensively discuss image processing techniques critical for accurate body shape analysis. Topics include image segmentation, edge detection, color space conversion, thresholding, noise reduction, and width calculation. Robustness to noise, illumination, rotation invariance, and suitability for textured images are underscored, with detailed explanations of the algorithms and tools that were employed.

Results from the papers showcase the effectiveness of their methods. The first paper excels in athletics selection, which shows a high accuracy in body shape analysis. The second paper successfully categorized female body shapes, enhancing garment fitting in the clothing industry. Potential benefits include improved body image perception, personalized garment production in terms of bespoke sizing, and cost-effectiveness compared to 3D body scanning technology.

In conclusion, these two papers highlight the impact of image processing techniques on body shape analysis, offering innovative solutions and cost-effective alternatives. These approaches have implications for sports science, athletics selection, and the clothing industry, contributing valuable insights to the field and paving the way for further advancements in accurately analyzing body shapes.

**User Classes and Characteristics**

**Beginners in Fitness:**

Our app’s focus is on this user class since our app provides a starting point for fitness enthusiasts.

Characteristics:

1. Limited knowledge about fitness and body anatomy
2. Eager to start a fitness journey but unsure where to begin
3. Need guidance for targeted exercises

**Needs:**

1. User-friendly interface with clear exercise guidelines
2. Simple and clear recommendations for workouts

**Fitness Enthusiasts:**

These are people who are well into their fitness journey and need advanced guidance to correct physique proportionality ratios.

**Characteristics:**

1. Proactively engaged in fitness activities
2. Seek detailed insights into their body composition
3. Motivated to optimize workouts for specific body areas

**Needs:**

1. In-depth body analysis for tailored fitness and visualizations.

Design and Implementation Constraints

Platform Compatibility:

The app is compatible with a variety of mobile devices supporting Android platforms. Design considerations ensure a consistent and optimal user experience across different screen sizes and resolutions. The minimum version of android is Android 10.

**Image Processing Optimization:**

Due to the nature of MediaPipe machine learning model, it proved difficult to integrate the backend locally with the app, therefore we used FastAPI to create a server that hosts our image processing algorithm. It returns the results of images back to the app for the user to see.

**Technological Stack:**

We have used Python OpenCV to work on the image processing part of the project with the help of the MediaPipe machine learning algorithm. This consists of the backend which is stored on a server using Firebase and FastApi. The front end is made for Kotlin Android Studio.

**Maintainability and Upgrades:**

The design considers maintainability and ease of future upgrades.

**Educational Content Presentation:**

Presenting educational content in a way that is both informative and engaging imposes constraints on content format, multimedia integration, and accessibility for users with varying levels of health and fitness literacy.

**Assumptions and Dependencies**

We are assuming that the environment in which the user is taking pictures is well lit, and that the user is wearing black or dark clothes with a contrasting background without shadows. We are also assuming that the user has an active internet connection.

**Functional Requirements**

**Getting Pictures**

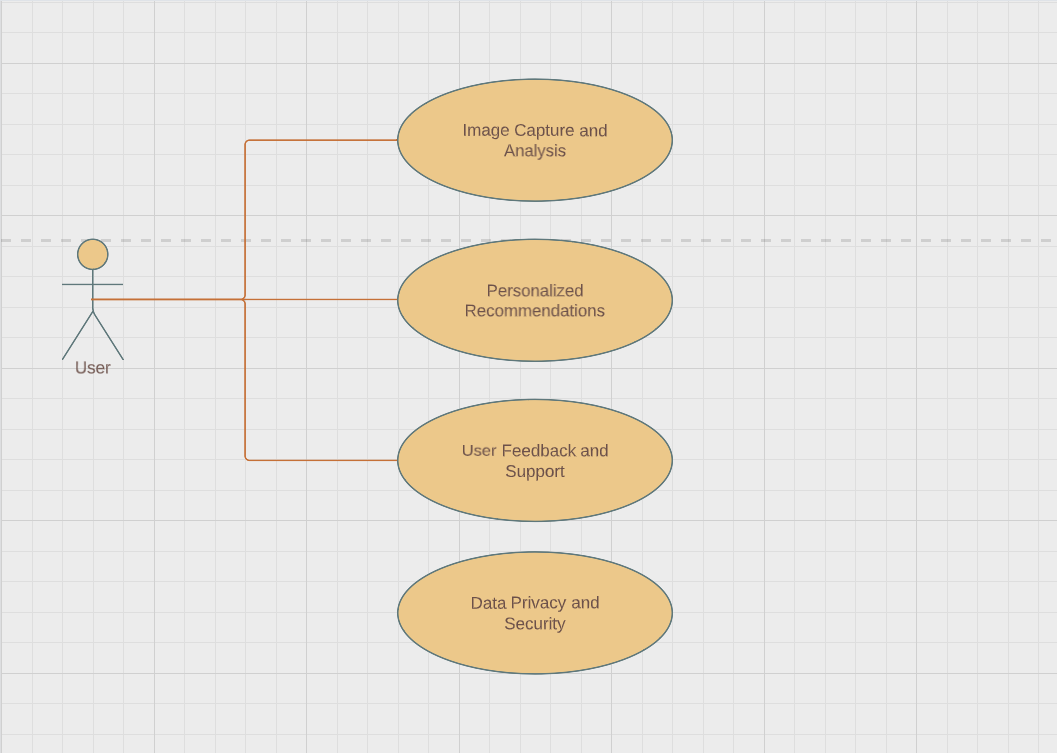
|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Getting Pictures | |
| **Purpose** | | Enable users to capture full-body images using the device's camera or upload images from the gallery.  Apply image analysis algorithms to extract relevant measurements and analyze the user's body shape accurately | |
| **Priority** | | High | |
| **Pre-conditions** | | Open Camera Icon is showed | |
| **Post-conditions** | | Camera is Opened | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Clicks The Camera Icon | | Opens Camera |
| **2** | Snaps front and then side pictures | | Stores them into Phone’s Gallery |
| **3** | Selects the relevant Pictures (front side first) | | Sends the selected pictures to the backend |

**Table 1: Getting Pictures**

**Personalized Recommendations**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Personalized Recommendations | |
| **Purpose** | | Generate personalized fitness recommendations based on the user's body shape analysis.Provide exercise routines, workout plans to target specific areas of the body for improvement. | |
| **Priority** | | High | |
| **Pre-conditions** | | User Needs to have selected the Pictures | |
| **Post-conditions** | | User gets Personalized exercise recommendations from the app | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Selects the Pictures | | Take those pictures and send it to the backend and backend does all the Image Processing |
| **2** | Receives the recommendations | | Backend Returns the results of body Analysis based on which the app will give exercise recommendations |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Selects the Pictures | | Take those pictures and send it to the backend and backend fails to detect the body sends back error message with pop up of guidelines on how to snap pictures |
| **2** | Returns to the main page to open Camera | |  |

**Table 2: Personalized Recommendations**

**Use Case Diagram**

**Figure 1: Use Case Diagram**

## 

**Nonfunctional Requirements**

**Performance Requirements**

The User’s device must at least support Android 10 and optimally should be Android Version 13 for the App to work properly. The Device must have a working Camera with at least 12 Megapixels Camera and a working flashlight for the clarity of the image.

**Safety Requirements**

In order to ensure that no one assesses a user’s account on our application, it is important that the user not share their login credentials with anyone.

**Security Requirements**

The image snapped of the user will be sent to the back end in order to process it but as soon as the processing is complete the algorithm will proceed to delete the image from its storage in order to ensure privacy of the user. The only place the image will remain stored is the gallery of the user's personal device.

**Additional Software Quality Attributes**

The body analysis algorithm is in its Alpha phase, hence it is sensitive to poor image quality and wearing loose clothes or having a non-contrasting color wall to the background will result in faulty values. Covering the contour of the body with things like hands can also affect the accuracy of the results.

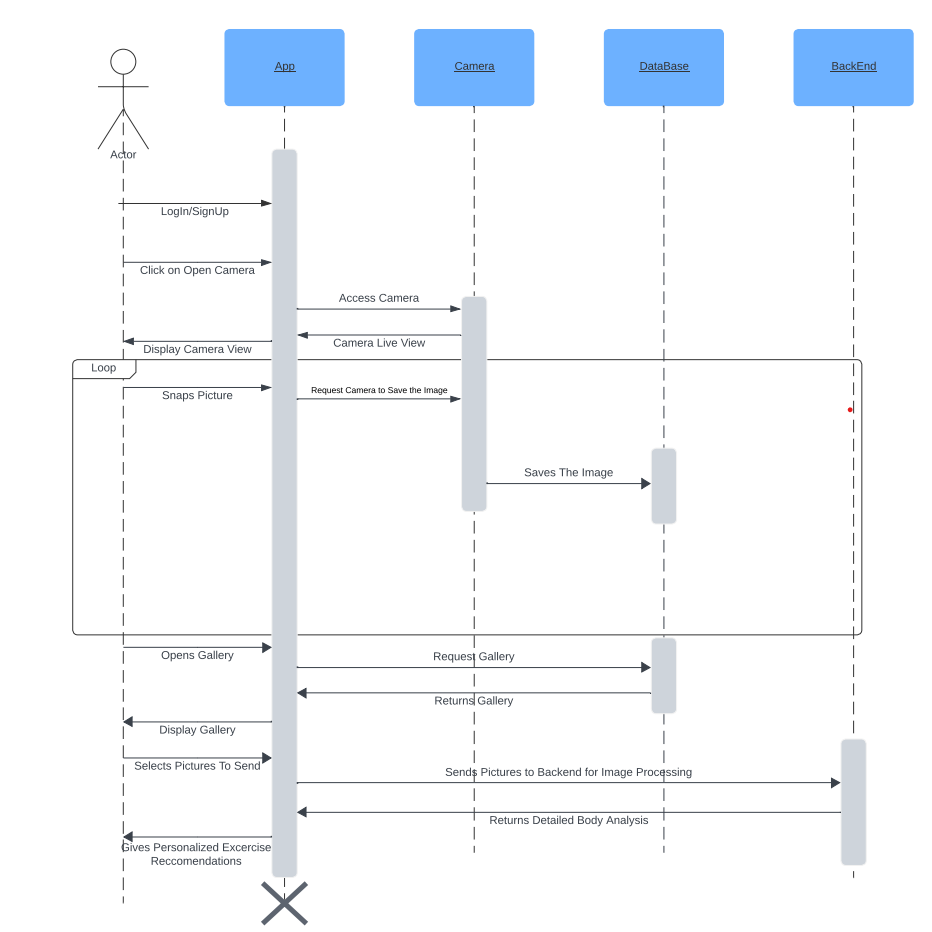
**Other Requirements**

User’s are required to have at minimum Android 10 running on their device, and preferably Android 13. This will ensure that the application works smoothly on their device.

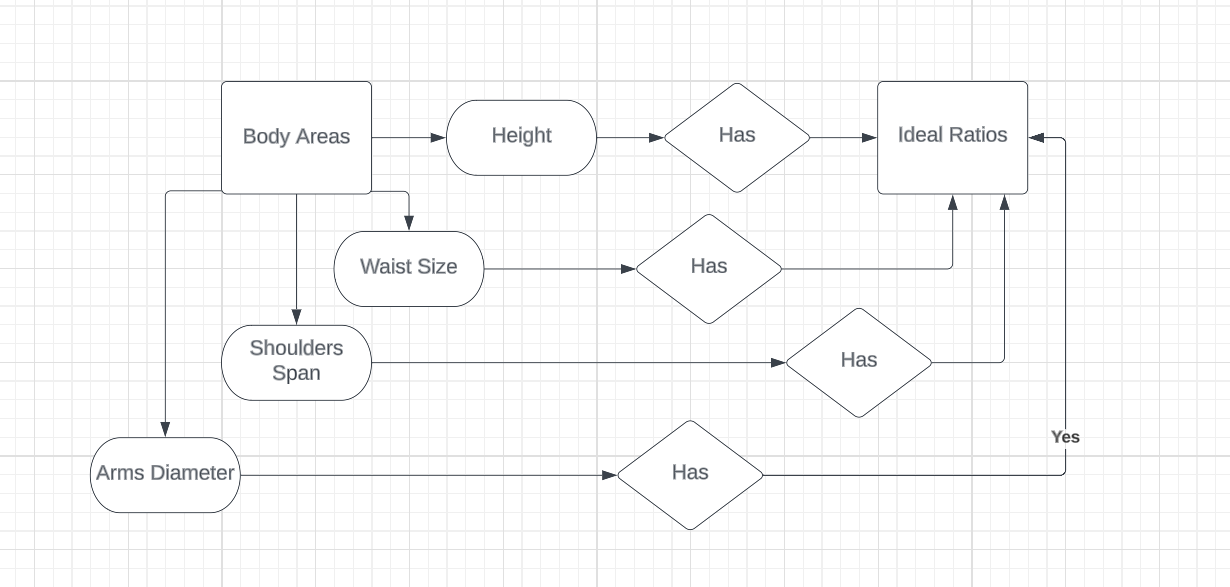
# System Design

**Application and Data Architecture**

**Sequence Diagram**

**Figure 2: Sequence Diagram**

**Entity Relationship Diagram**

****

**Figure 3: Entity Relationship Diagram**

**Component Interactions and Collaborations**

The way our application is setup is such that the app is in the user’s phone, and when they submit their pictures for the body analysis image processing model to process, the image is sent to a Firebase Storage, where the image receives a download URL which is provided to the Python script. From there, the Python script downloads the image, which is running on the FastAPI server that hosts the image processing algorithm. After the image has been processed, its analysis is sent back to the app by FastAPI, for the user to see. These are the interactions and collaborations between our system components and processing unit.

**System Architecture**

Our project is based on RESTfulAPI, which bridges the gap between our Android Mobile Application and our backend Python MediaPipe Machine Learning script.

**Architecture Evaluation**

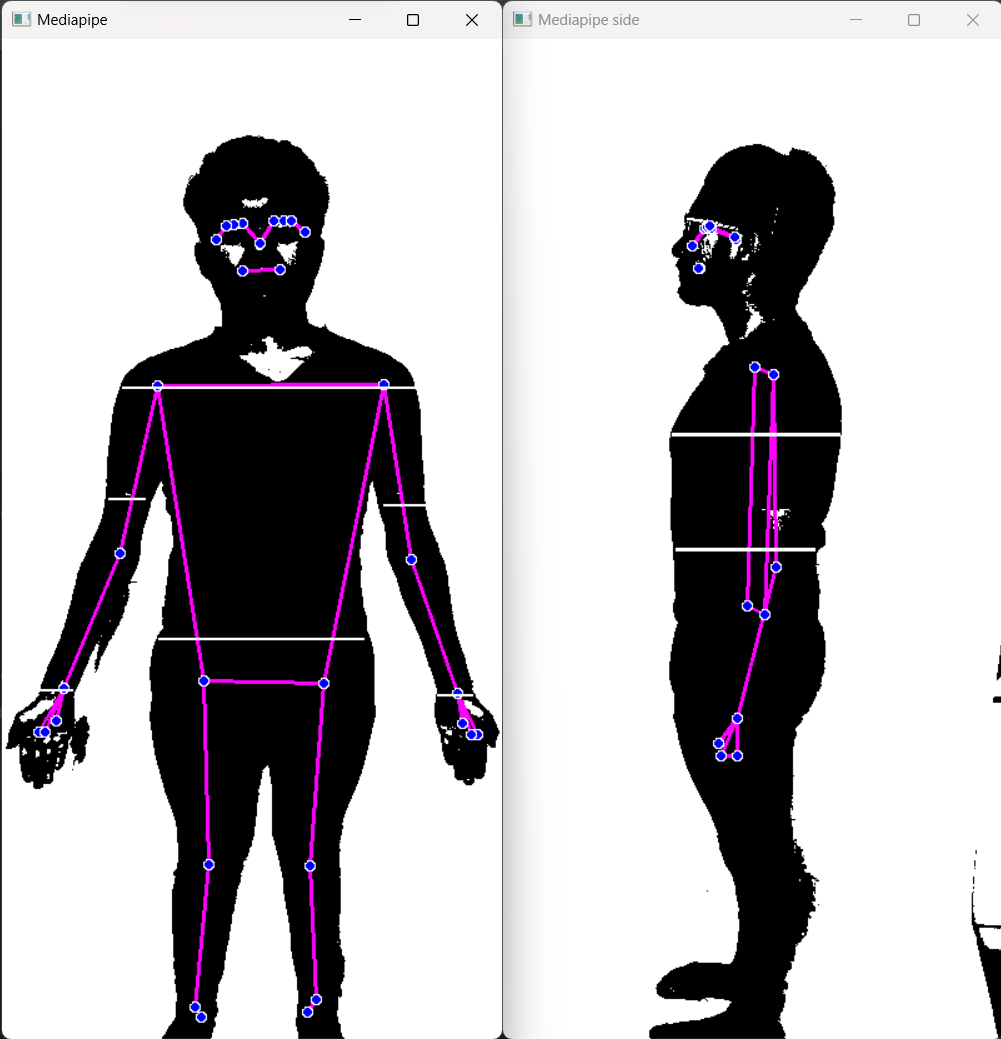
We primarily used MediaPipe as the foundation of our image processing model because of its ability to map out the human body accurately, because of its ability to detect body parts no matter what position they are in. This feature proved very useful to build our image processing model upon.

However, due to the way that MediaPipe’s machine learning model is designed, it was difficult to locally integrate the backend with the app. Therefore we used FastAPI to create a server that hosts our image processing algorithm. It returns the results of images back to the app for the user to see. Using a cloud-based architecture would also prove to be a long and complex task, so we decided to use FastAPI.We used Firebase to create a secure login portal, as well as a secure database for images to be stored.

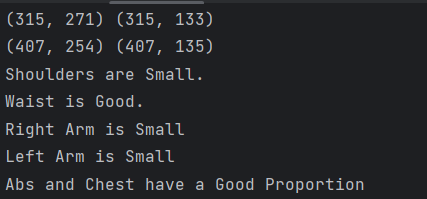
We have used Python OpenCV to work on the image processing part of the project, since OpenCV is necessary for python to work with images, with the help of the MediaPipe machine learning algorithm.

**Component-External Entities Interface**

Our app’s login is based on and interacts with the Firebase Authentication system

**Screenshots/Prototype** 

These are post-processed images sent by our app to the backend. It is a visual representation of how our backend detects the edges of the user’s body and determines the distance between each point to give us the measurements of each body part and compares them with the optimal body shape to give back the app string values on which app provides specific exercises to the User.



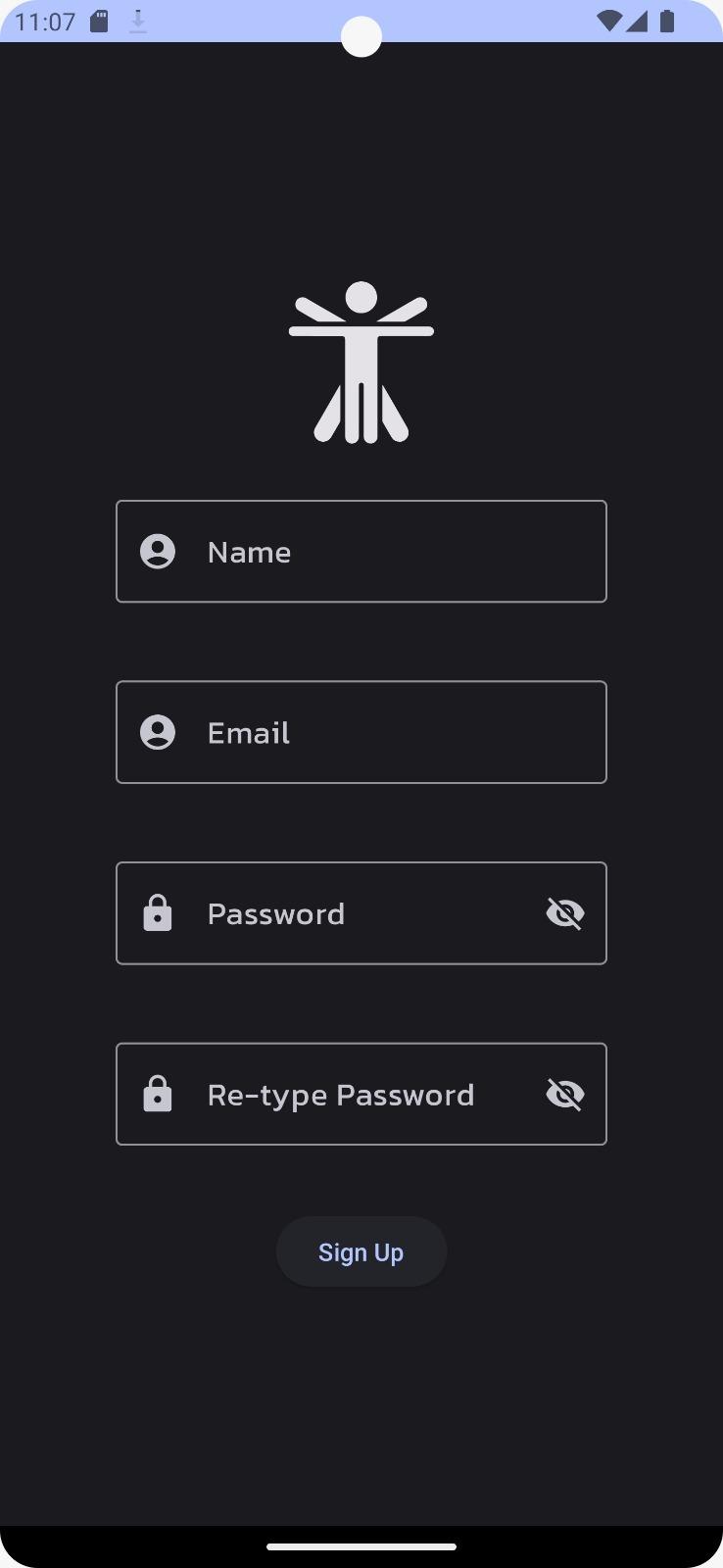
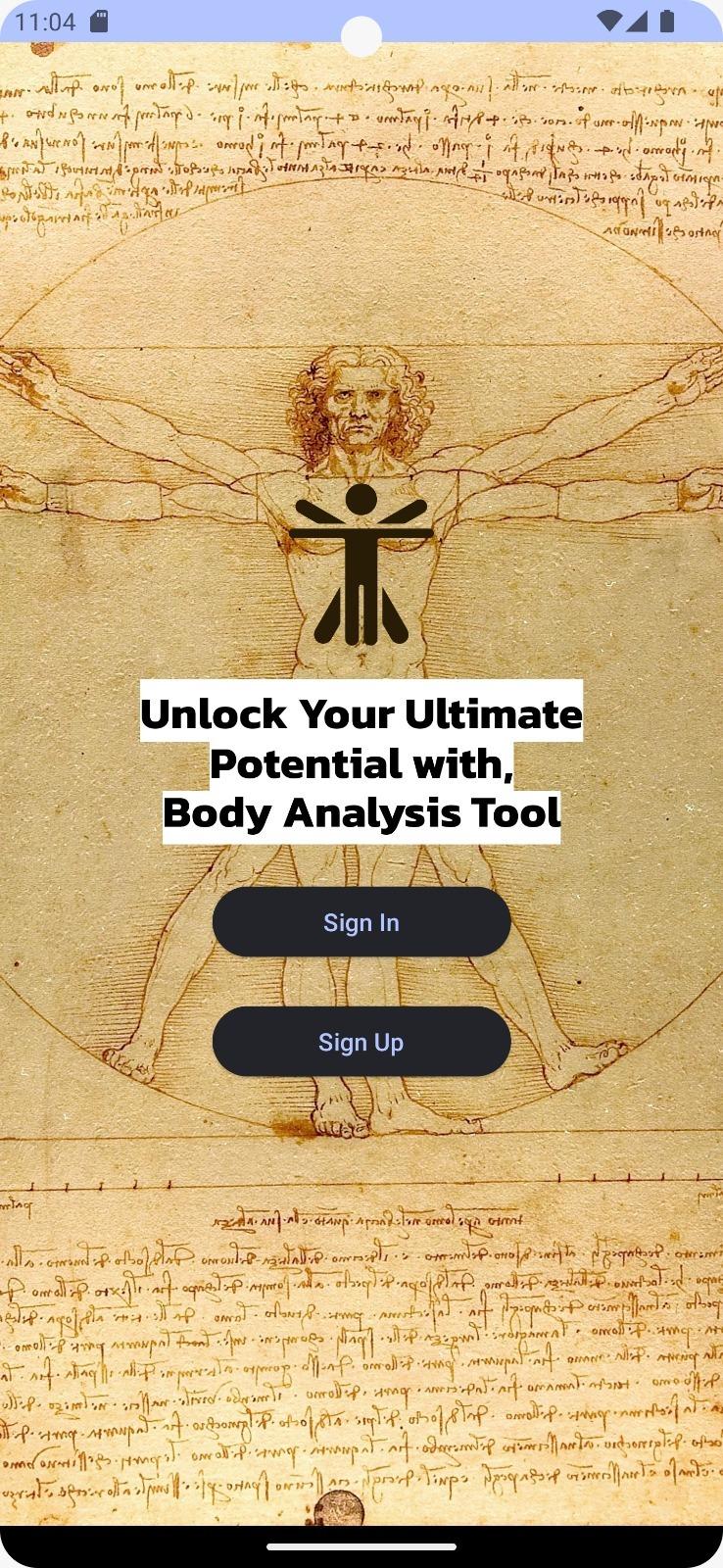
These are the string values sent back to the app by the backend on which further processing is done by the app.

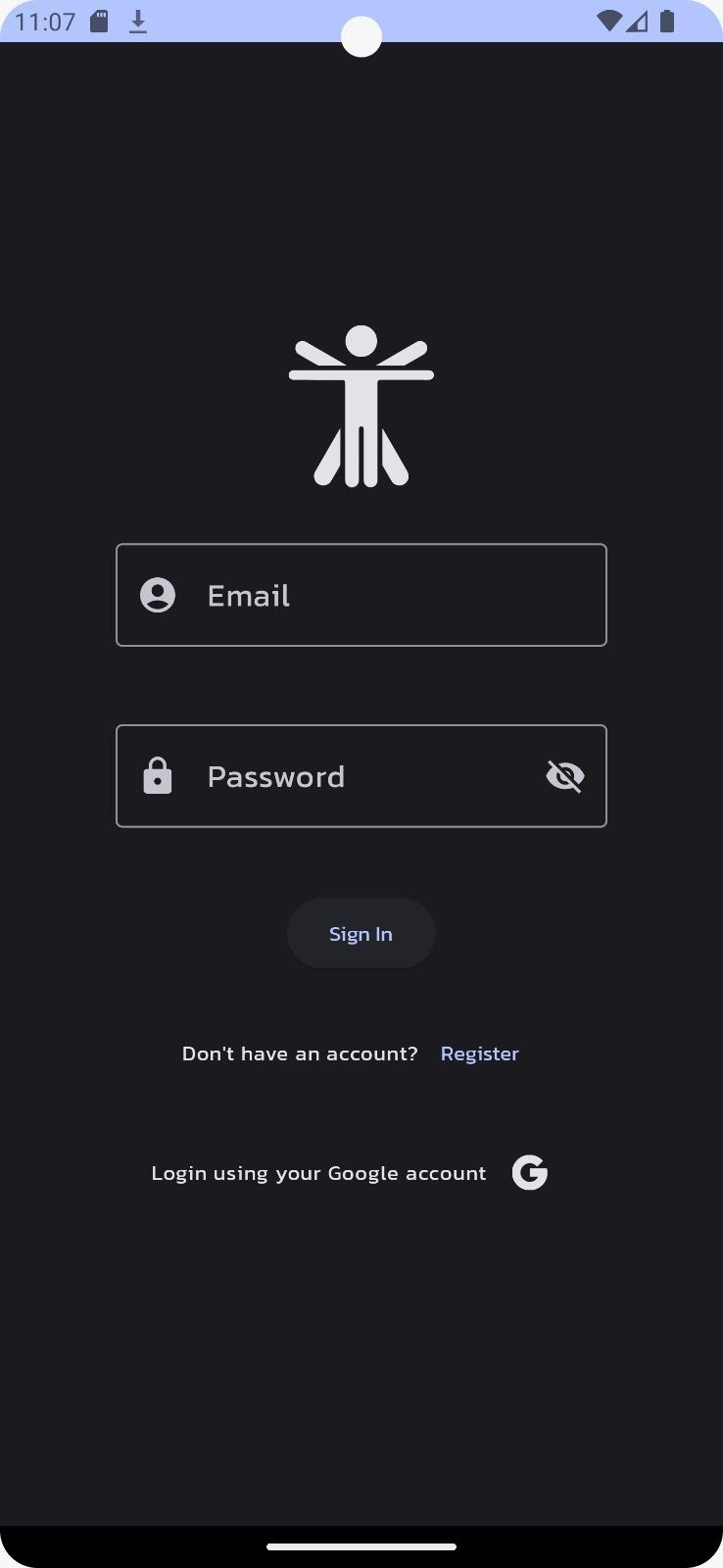
**Workflow**

After the user logs into their account on the app, they are welcomed in. Then the user takes new pictures by selecting the option to new pictures, or selects previously saved pictures from their gallery. Once the pictures are submitted to the app, it sends them to the Firebase database for safe storage. From there, the image is called upon by the FastAPI server that hosts the Image Processing model algorithm. Once FastAPI receives the images and image processing algorithm is applied on it, the analysis results are sent back to the application interface for the user to see the results.

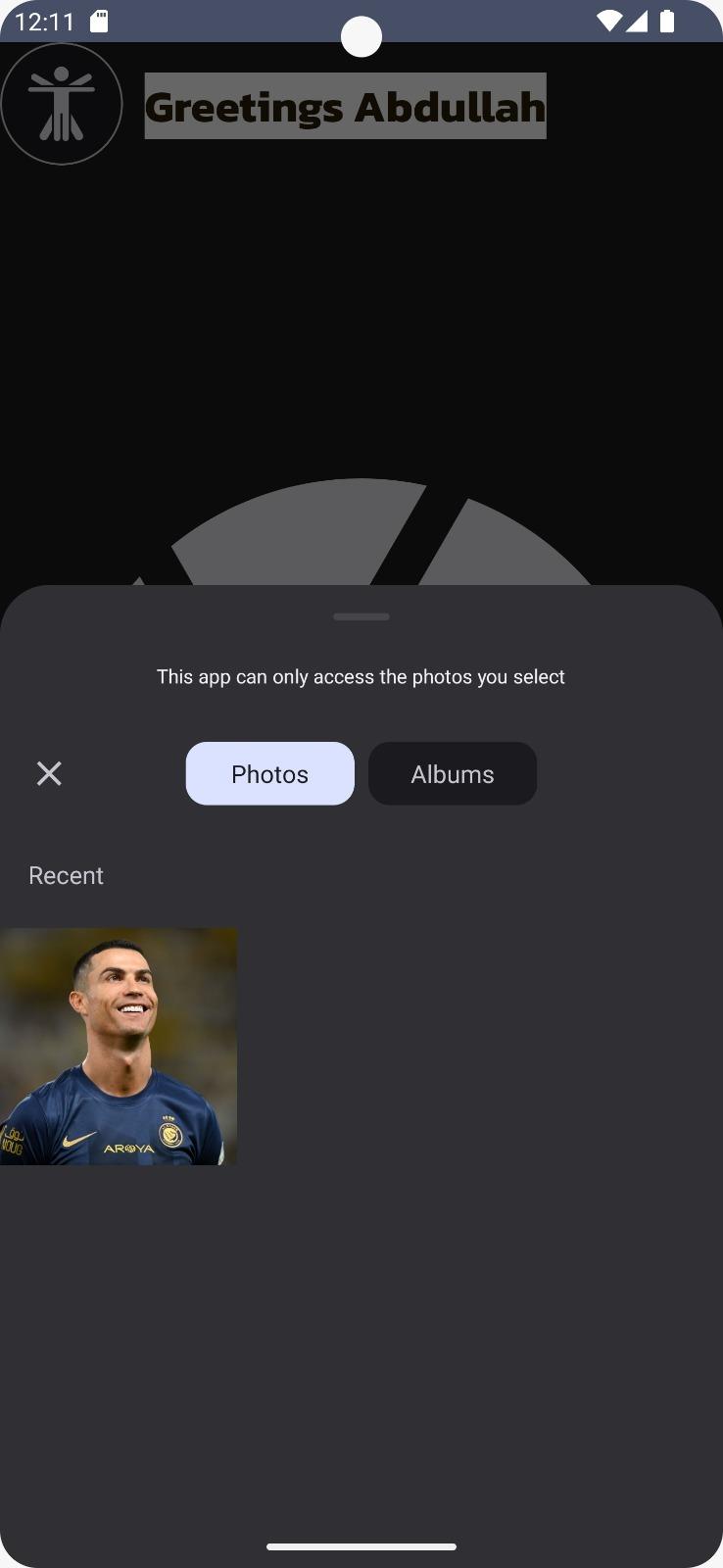
### 

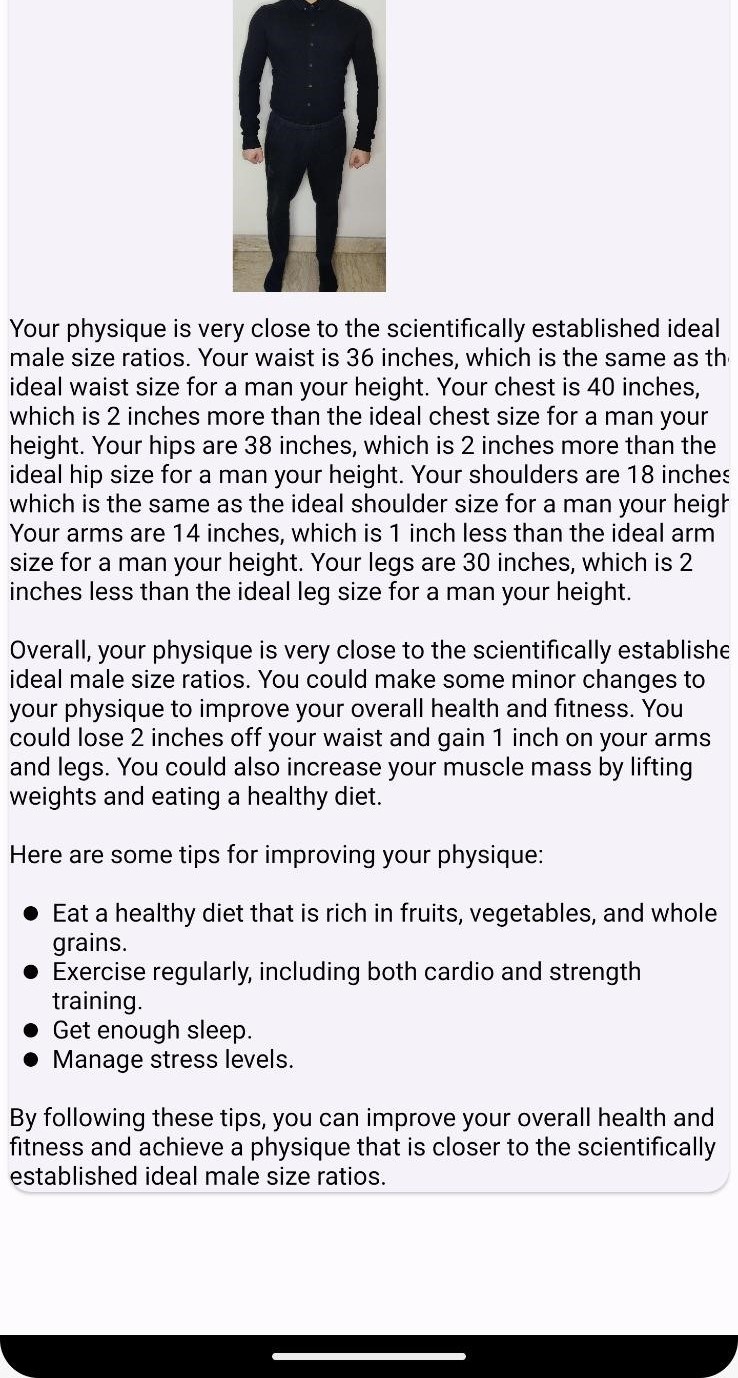
**Screenshot/UI**











# Test Specification and Results

# Test Case Specification

|  |  |
| --- | --- |
| **Identifier** | TC-1 |
| **Related requirements(s)** | A User with clothes that are not loose and a white background to snap pictures that is well lit. |
| **Short description** | We took an overweight person and gave his pictures to our model. |
| **Pre-condition(s)** | … |
| **Input data** | Two pictures of the user. Front and Side |
| **Detailed steps** | Snap his pictures. Select them to send to the backend. Receive exercises and detailed analysis of his body |
| **Expected result(s)** | Sensing the difference discrepancy between ratios and pointing them out |
| **Post-condition(s)** | … |
| **Actual result(s)** | Our model calculated that the chest and abs were not in line with the optimal ratios and pointed them out. But had a problem sensing it from the front view during waist and shoulder comparison. |
| **Test Case Result** | Semi-Success: Inaccuracy in front pose detection in case when fat is only present on front of abdomen instead of side (love handles) |

**Table 3.1: TC-1**

|  |  |
| --- | --- |
| **Identifier** | TC-2 |
| **Related requirements(s)** | A User with clothes that are not loose and a white background to snap pictures that is well lit. |
| **Short description** | We took an underweight person and gave his pictures to our model. |
| **Pre-condition(s)** | … |
| **Input data** | Two pictures of the user. Front and Side |
| **Detailed steps** | Snap his pictures. Select them to send to the backend for image processing. Receive exercises and detailed analysis of his body |
| **Expected result(s)** | Sensing the difference discrepancy between ratios and pointing them out |
| **Post-condition(s)** | … |
| **Actual result(s)** | Our model calculated that the shoulder to waist ratio was not in line with the optimal ratio and pointed it out. It also detected that the user had smaller arms than what would be proportionally ideal for them. |
| **Test Case Result** | Success: the model accurately made suggestions in line with the user’s manually calculated physique ratios, working as it should. |

**Table 3.2: TC-2**

**Summary of Test Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module Name** | Test cases run | Number of defects found | Number of defects corrected so far | Number of defects still need to be corrected |
| **Overweight User** | TC-1 | 1 | 1 | 0 |
| **Underweight User** | TC-2 | 0 | 0 | 0 |
| **Complete System** | TC1,TC2 | 1 | 1 | 0 |

**Table 3.3: Summary of All Test Results**

# 

# Conclusion and Future Work

**Project summary**

Over the course of our testing, we found that (as previously mentioned), 80% of our 10 test users gained a better idea of how to go about transforming their physique by identifying their physique’s strong and weak points. When their physique’s measurement ratios were evaluated manually via measuring tape to confirm what the app suggested, the app’s results turned out to be within 5-10% deviation at most compared to manual calculations.

**Problems faced and lessons learned**

It was challenging for us to figure out how to integrate our frontend with our backend, due to not having a full understanding of how MediaPipe works when we first decided to use MediaPipe for our project. Finding the right resources and guidance was a challenge due to fewer people working in the Image Processing domain. It was difficult to get guidance when we needed more specific direction on minute details.

It taught us the important lesson that when choosing technologies for any project/software development, it is crucial to have a solid understanding of how various technologies interact with each other, and how much effort is required to produce the desired outcome. We learned the importance of communicating with others who are more experienced than us in order to channel our efforts into a productive direcion, instead of working in a direction that is futile.

## 

**Future work**

For future work, the app can be worked on to add image processing for female users as well, based on scientifically ideal figure ratios. Furthermore, more work can be done to give users complete customized

workout routines generated by Artificial Intelligence, based on where each user currently stands in terms of their physique, as well as complete diet plans based on every user’s unique preferences.

**References**

* Paper Title: Web Based Digital Image Processing Tool for Body Shape Detection  
  Authors: Ozge Kart,Alp Kut, Arzu Vuruskan and Ender Bulgun  
  <https://proceedings.ictinnovations.org/attachment/paper/236/web-based-digital-image-processing-tool-for-body-shape-detection.pdf>
* Paper Title: Body Shape Analysis via Image Processing  
  Authors: Jinyan Yang, Yu Li, Tao Jiang, Yu Wei, Guanlei Xu  
  <https://www.atlantis-press.com/proceedings/erse-13/7426>

# Glossary

FastAPI: FastAPI is a modern web framework for building RESTful APIs in Python (Google definition)

RESTful API: RESTful API is an interface that two computer systems use to exchange information securely over the internet (Google definition)

Image Processing: Image processing involves performing operations on an image to make it better or to get important information from it.

# Deployment/Installation Guide

Simply clone the application from GitHub repository and install on system.

# User Manual

Manual for application users:

1. Signup
2. Login with your credentials
3. Select your images from your gallery or take news picture with a white background
4. Submit the pictures
5. Wait and receive the analysis from the application