

Contents

1 Introduction	4
2 Current System	4
3 Proposed System	5
3.1 Overview	5
3.2 Functional Requirements	6
3.2.1 Sign up & Log in	6
3.2.2 Pose/Exercise Selection	6
3.2.3 Pose/Exercise Detection & Feedback	6
3.2.4 Program Suggestion	7
3.2.5 Create Your Own Program	7
3.2.6 Meditation	8
3.2.7 Calorie Tracking	8
3.2.8 Progression Tracker	8
3.2.9 Notification	8
3.2.10 Miscellaneous	9
3.3 Non-functional Requirements	9
3.3.1 Response Time	9
3.3.2 Availability	9
3.3.3 Accuracy	9
3.3.4 Usability	9
3.3.5 Privacy	10
3.3.6 Extendibility	10
3.4 Pseudo Requirements	10
3.5 System Models	11
3.5.1 Scenarios	11
3.5.2 Use-Case Model	24
3.5.3 Object and Class Model	25
3.5.4 Dynamic Models	26

3.5.5 User Interface	30
4 Other Analysis Elements	48
4.1 Consideration of Various Factors in Engineering Design	48
4.1.1 Public Health	49
4.1.2 Public Safety	50
4.1.3 Public Welfare	50
4.1.4 Global Factors	50
4.1.5 Cultural Factors	50
4.1.6 Social Factors	51
4.2 Risks and Alternatives	51
4.3 Project Plan	53
4.4 Ensuring Proper Teamwork	59
4.5 Ethics and Professional Responsibilities	60
4.5.1 Professional Issues	60
4.5.2 Ethical Issues	60
4.6 Planning for New Knowledge and Learning Strategies	60
5 References	61

Analysis and Requirement Report

Exercaise

1 Introduction

Exercise is one of the crucial elements for a healthy life. However, nowadays, access to personal trainers is rather expensive and the process of getting a personalized exercise program is time consuming. While trying to exercise by themselves, people often get injured due to incorrect form and get mediocre results from their exercise.

Exercaise offers an easy solution in the form of an AI powered personal trainer. By using ML and joint detection technologies, Exercaise aims to increase the user's performance by creating a personalized exercise program and guiding the user through the process. Exercaise uses the phone's camera to observe the user during their workout and gives them real-time feedback on their form, both visually and verbally. With the sectional feedback feature, users will be able to learn which part of their body is in correct form and which part is not in a short amount of time. With features such as level-based program creation and form control, Exercaise will provide a safer exercise experience tailored to user needs.

This report consists of three main sections: Current system, Proposed System, and Other Analysis Elements. Current System section examines the existing system used in similar applications. In the Proposed System section, Functional, Non-functional, and Pseudo Requirements are explored in detail. System models are also given under this section with Scenarios, Use Case model, Object and Class models, Dynamic models and User Interface design. Last but not least, the section of Other Analysis Elements contains an examination of various factors in engineering design, risks and alternatives, and the project plan. Ensuring proper teamwork, ethics and professional responsibilities, and learning strategies are also discussed under this section.

2 Current System

In the current system, there are various applications with the concept of AI personal trainer. For example there is AIFit, it is a similar computer vision app to use with gym

equipment but it's scope is smaller and its implementation is not done [1]. There is skill yoga, where it is also a computer vision app but just for form-checking yoga poses and nothing else [2].

In addition to doing a form check on a wide variety of exercises, instead of focusing on one distinct area, we wanted to implement something that would act more as not just a personal trainer but as a companion. We wanted to combine multiple aspects that a person needs in their active life and we wanted to ease the use to motivate them and help them. We wanted to provide an application where users are motivated to use, where they can easily reach what they want and achieve perfect results in a single application. Hence, Exercaise is an application that combines the features that are needed for an active life and gives the opportunity to have everything in your hand in only one application.

3 Proposed System

3.1 Overview

Exercaise will be a mobile application that aims to optimize exercise quality and performance by providing personalized exercise programs and monitoring the form of the user. The application will generate a suitable program according to the information the user has provided, such as their current work-out routine, and the body parts they want to work on. Before starting the exercise routine, the user will position their mobile phone to capture their whole body. After activating the program on their phone, the user will get in front of the camera and start following their personalized routine. The application will detect the user's joints and limbs in the form of dots and lines. During the exercise, it will give feedback to the user when necessary, pointing out the incorrect form verbally and emphasizing the limbs and joints that are in incorrect position with the color red. Exercaise will also count repetitions for exercise activities that require counting, such as squats, push-ups, or sit-ups.

The most significant engineering challenge of developing a multi-functional and efficient application such as Exercaise will be regulating the response time for feedback feature, which is one of the distinguishing aspects of our application. The

response time should not exceed the expected limits in order to provide a fast and pleasant service. Optimized algorithms and cloud services will be utilized for this purpose.

Exercaise will bring innovation to the concept of home training with its product performance and process. With the application's many functionalities, users will be able to manage the programs of all their physical activities in one place. The process of the services provided by Exercaise will be operated by AI, making it the signature method of managing our application. Thus, Exercaise will implement and optimize sustainable innovations through its life cycle.

3.2 Functional Requirements

3.2.1 Sign up & Log in

The application should

- Allow users to register manually or using Open Authentication.
- Ask the user's age, height, weight and work-out routine while registering.
- Request permission for the user's camera.

3.2.2 Pose/Exercise Selection

The application should

- Offer the user default exercises or yoga poses to select from.
- Offer the manual of each pose/exercise.
- Allow the user to like/dislike a pose or a program.
- View the like/dislike of each pose or program.
- Filter the yoga poses or programs and exercise sets according to the body region they activate, duration, popularity or spiritual mindset.
- Allow the user to select one or more yoga poses to learn.
- Allow the user to select a default program that consists of different yoga poses or exercises.

3.2.3 Pose/Exercise Detection & Feedback

The application should

- Detect and view the joints of the user as a stick drawing.

- Ask the user to perform the current pose/exercise in a certain period of time.
- Scan the body sections of the user and decide if the user is performing the current pose/exercise properly.
- Detect which body section of the user is not matched with the current yoga pose or exercise movement.
- Give vocal feedback to the user according to the specific body regions and guide them through the pose or exercise.
- Highlight the joints of the body regions that do not fit the pose/exercise with red.
- Highlight the joints of the body regions that fit the pose/exercise with green.
- Display a countdown for the duration of the pose/exercise in order to inform the user of the exercise period.
- Show the success percentage of the user for the current pose/exercise.
- Show the success percentage of the user for the whole program.

3.2.4 Program Suggestion

The application should

- Suggest the user a program according to the body section and the difficulty level the user wants to exercise with.
- Allow the user to select one or more body sections to exercise with.
- Notify the user with newly added programs.
- Notify the user about the daily yoga/exercise challenges.

3.2.5 Create Your Own Program

The application should

- Allow the user to create their own program.
- Require the user to specify the poses/exercises they want to add to the program.
- Require the user to specify the arrangement of the poses/exercises in the program.

- Require the user to specify the number of sets and repetitions of the poses/exercises in the program.
- Require the user to give a title to their program.
- Allow the user to delete the program they created.
- Allow the user to update the program they created.

3.2.6 Meditation

The application should

- Present different types of audio meditations to the user.
- Notify the user when the meditation ends.

3.2.7 Calorie Tracking

The application should

- Calculate the calorie burn by the correctness of the pose, number of repetition and weight of the user.
- Show the calorie burned at the right side of the screen while doing an exercise (optional).
- Generate plots for weekly and monthly calorie burn.

3.2.8 Progression Tracker

The application should

- Show the user statistics about correctness rates (percentages) of poses that are worked on.
- Allow the user to track their progression specific to a pose/exercise.
- Show the user how many hours spent doing exercises both pose specific and general.

3.2.9 Notification

The application should

- Send notifications when new poses are added to the app.
- Send notifications to motivate the user to make exercises.

3.2.10 Miscellaneous

The application should

- Allow the user to delete their account.
- Allow the user to update their address, name, password, and email.
- Allow the user to change their weight, height, and work-out routine.
- Allow the user to reset the password if they forget it.

3.3 Non-functional Requirements

3.3.1 Response Time

The response time for recognizing the poses through the camera is one of the crucial features of Exercaise since it aims to give real-time audio feedback for the user. Retrieving and processing the visual of the users' pose should take at most 2 seconds. Sending notifications from the application should take at most 2 seconds as well.

3.3.2 Availability

The application's server should be available at all times since the user could want to exercise at any time of the day. Other than the service availability, since the application will be implemented to be compatible with Android and IOS devices, it would be fair to say that Exercaise aims for availability as one of the main requirements.

3.3.3 Accuracy

Many of the main features of the application heavily depend on accuracy. Some of the features which depend on accuracy are pose and joint recognition for the real-time feedback, and the progression tracking. In order to make these possible, the recognition accuracy of the joints should be at least 80%.

3.3.4 Usability

Exercaise aims to provide an application for a great range of stakeholders since exercising is an activity that could be done in any age group. There could be users who are not familiar with new technologies involving camera

usage and a new interface. Therefore the application should be user-friendly with informative but quick tutorials on how to use the application.

3.3.5 Privacy

The application will get permission to access the user's camera. However the videos that are taken by the application will not be shared with any third parties since it could be considered as sensitive information. Exercaise should provide a safe environment respecting the users' private information and data.

3.3.6 Extendibility

The application will give the feedback in English, therefore one of the future goals for Exercaise is to make it more extensible for other countries by including other languages for the user interface and also the audio feedback.

3.4 Pseudo Requirements

- The application will be a mobile application.
- The application will be available for both IOS and Android.
- The application needs a mobile phone with a camera.
- The application will need an internet connection.
- The application has to see the full body of the user for pose detection & feedback feature to work properly.
- The application will not share any private information or camera data with 3rd parties.

3.5 System Models

3.5.1 Scenarios

Scenario	Sign Up
Participating Actor	User
Entry Conditions	- User is not signed in
Exit Conditions	- Sign up process successful - Sign up process failed
Flow of Events	1- User pressed the “Sign Up” button from main menu. 2- User puts in their personal information to sign up. 2.a – User puts in their weight, height and activity level 3- User gets a confirmation mail, after confirmation is complete the sign up is successful.

Scenario	Enter Personal Information
Participating Actor	User
Entry Conditions	- User is not signed in - User is in the process of signing up
Exit Conditions	- User goes to the next part of sign up
Flow of Events	1- User puts in their weight, height and activity level

Scenario	Log In
Actors	User
Entry Conditions	- User is not signed in.
Exit Conditions	- Sign in process successful - Sign in process failed
Flow of Events	1- User pressed the “Sign in” button from main menu. 2- User puts their personal information to login. 3.a – If the credentials are correct, the use case is complete. 3.b – If the credentials are false, the user tries again.

Scenario	Forgot Password
Actors	User
Entry Conditions	- User is not signed in. - User has an account.
Exit Conditions	- User’s password is reset
Flow of Events	1- User clicks the “Forgot Password” button. 2- User puts in their personal information to reset the password. 3- User gets a confirmation email. 4- User resets the password using their confirmation email.

Scenario	View User Profile
Actors	User
Entry Conditions	- User is logged in.
Exit Conditions	- User leaves their profile page.
Flow of Events	1- User presses the profile button on the navigation bar. 2- User views their profile information which includes the progressions and calories burned.

Scenario	Track Calorie Burned
Actors	User
Entry Conditions	- User is viewing their profile.
Exit Conditions	- User leaves their profile page.
Flow of Events	1- User sees a chart of calorie burned over the last month in calendar view

Scenario	View Progression
Actors	User
Entry Conditions	- User is viewing their profile.
Exit Conditions	- User leaves their profile page.
Flow of Events	1- User sees their progression on accuracy of the movements and their workout routine for the past month with a calendar view and a plot.

Scenario	Edit Profile
Actors	User
Entry Conditions	- User is on the profile page.
Exit Conditions	- User successfully edited their profile or canceled the editing process.
Flow of Events	1- User presses the edit button on the profile page. 2- User edits the field they want to edit. 3- User clicks save to apply changes or cancel to exit edit screen.

Scenario	View Notifications
Actors	User
Entry Conditions	- User is logged in. - User is on the home page.
Exit Conditions	- User leaves notifications page.
Flow of Events	1- User presses the notifications icon in the navigation bar. 2- User views the notifications they have gotten in the form of a list.

Scenario	Disable Notifications
Actors	User
Entry Conditions	<ul style="list-style-type: none"> - User is logged in. - User is on the notifications page.
Exit Conditions	<ul style="list-style-type: none"> - User successfully disables the notifications.
Flow of Events	1- User presses the disable notifications button in the notifications screen.

Scenario	View Meditations
Actors	User
Entry Conditions	<ul style="list-style-type: none"> - User is logged in.
Exit Conditions	<ul style="list-style-type: none"> - User leaves the meditations page.
Flow of Events	<ul style="list-style-type: none"> 1- User presses the meditations button on the main screen. 2- User views the available meditations with information on the name, audio, and duration.

Scenario	View Programs
Actors	User
Entry Conditions	- User is logged in.
Exit Conditions	- User leaves the programs page.
Flow of Events	1- User presses the “Programs” button on the main screen. 2- User views the available, already created programs.

Scenario	Search Programs
Actors	User
Entry Conditions	- User is in the programs page.
Exit Conditions	- User successfully searched the programs with their desired query
Flow of Events	1- User presses the search bar in the programs screen to activate search functionality. 2- User enters a query for the name of the program they want to find in order to get what they are looking for.

Scenario	Like/Dislike Program
Actors	User
Entry Conditions	- User is in the programs page.
Exit Conditions	- User successfully liked a program or disliked an already liked program.
Flow of Events	1- User presses the heart button on the program they want to like/dislike.

Scenario	View Program Information
Actors	User
Entry Conditions	- User is in the programs page.
Exit Conditions	- User successfully viewed the program's details and pressed the back button to exit.
Flow of Events	<p>1- User presses on the card of the program that they want to see the details of.</p> <p>2- User views the program information such as difficulty level, exercises inside it, the time it takes.</p>

Scenario	Edit Program
Actors	User
Entry Conditions	- User is viewing a specific program.
Exit Conditions	- User successfully edited the program to their liking or cancelled.
Flow of Events	<p>1- User presses the edit button on the program details screen.</p> <p>2- User edits the fields they want to edit.</p> <p>3- User either saves or cancels to finish the edit operation.</p>

Scenario	Add/Remove Exercise
Actors	User
Entry Conditions	- User is editing or creating a program
Exit Conditions	- User successfully added/removed the exercise from the program
Flow of Events	<p>To add,</p> <ol style="list-style-type: none"> 1- User presses the add exercise button on the program screen. 2- User chooses the exercise they want to add 3- User selects the place of the exercise in the program <p>To remove,</p> <ol style="list-style-type: none"> 1- User selects the exercise they want to remove and presses minus button to remove it

Scenario	View Exercises
Actors	User
Entry Conditions	None
Exit Conditions	- User leaves the exercises page
Flow of Events	<ol style="list-style-type: none"> 1- User presses the Exercises button on the main page. 2- User views the available exercises.

Scenario	Filter Exercises
Actors	User
Entry Conditions	- User is in exercises page
Exit Conditions	- User successfully filtered the exercises
Flow of Events	1- User presses the filter button on the exercises screen. 2- User selects filters from available filtering options. 3- As the user selects filter options, the exercises on the page are renewed according to the filters.

Scenario	View Exercise Information
Actors	User
Entry Conditions	- User is in exercises page
Exit Conditions	- User successfully viewed exercise information and pressed back to exit
Flow of Events	1- User presses on the card of the exercise that they want to see the details of. 2- User views the exercise information such as how it is done, difficulty level, the time it takes, the average calorie burned.

Scenario	Search Exercises
Actors	User
Entry Conditions	- User is in exercises page
Exit Conditions	- User successfully searched the exercises with their desired query
Flow of Events	<p>1- User presses the search bar in the exercises screen to activate search functionality.</p> <p>2- User enters a query for the name of the exercise they want in order to find what they are looking for.</p>

Scenario	Like/Dislike Exercise
Actors	User
Entry Conditions	- User is in exercises page
Exit Conditions	- User successfully liked an exercise or disliked an already liked exercise
Flow of Events	<p>1- User presses the heart button on the exercise they want to like/dislike.</p>

Scenario	Start the Exercise
Actors	User
Entry Conditions	- User selected an exercise to view
Exit Conditions	- User successfully completed the exercise - User left the exercise
Flow of Events	<p>1- User selects the start button on the exercise page to start the exercise.</p> <p>2- User is presented with a video on how to do a pose before starting out.</p> <p>3- After giving camera permission, user must put their phone and position themselves in a way that their camera can understand their whole body.</p> <p>4- After the body recognition is complete, user starts doing the exercise.</p> <p>5- User will see their joints on the screen painted over their body in red or green light depending on whether they are doing the move correctly or not.</p> <p>6- User will hear vocal feedback from our application about their form and what they should fix to have a better form.</p>

Scenario	Mute the Voice Warnings
Actors	User
Entry Conditions	- User is doing an exercise now
Exit Conditions	- User successfully muted the voice warnings
Flow of Events	1- User presses the headphone button in the exercise screen to mute the voice warnings about their form

Scenario	Start the Program
Actors	User
Entry Conditions	- User selected a program to view
Exit Conditions	- User successfully finished the program - User exited the program
Flow of Events	<p>1- User selects the start button on the program page to start with the first exercise.</p> <p>2- User is presented with a video on how to do a pose before starting out.</p> <p>3- After giving camera permission, user must put their phone and position themselves in a way that their camera can understand their whole body.</p> <p>4- After the body recognition is complete, user starts doing the exercise.</p> <p>5- User will see their joints on the screen painted over their body in red or green light depending on whether they are doing the move correctly or not.</p> <p>6- User will hear vocal feedback from our application about their form and what they should fix to have a better form.</p> <p>7- After the time ran out for the first exercise, the user will go to the next until the program is finished.</p>

Scenario	Create Program
Actors	User
Entry Conditions	- User is in the programs page
Exit Conditions	- User successfully created their own program or canceled
Flow of Events	1- User presses the plus button on the programs screen to create their own program. 2- User is presented with an empty program where they can add the exercises they desire from the available exercises list. 3- User gives a name to their program. 4- User presses save to save their program

Scenario	Delete Program
Actors	User
Entry Conditions	- User is in the programs page
Exit Conditions	- User successfully deleted a program
Flow of Events	1- User holds a program's card to open the options menu for the program. 2- User selects the delete program option.

Scenario	View Favorites
Actors	User
Entry Conditions	- User is on the main page
Exit Conditions	- User leaves the favorites page by pressing the back button
Flow of Events	1- User presses the heart button in the navigation bar to see all their favorites 2- User can see their favorite exercises, and programs on different tabs on this screen.

3.5.2 Use-Case Model

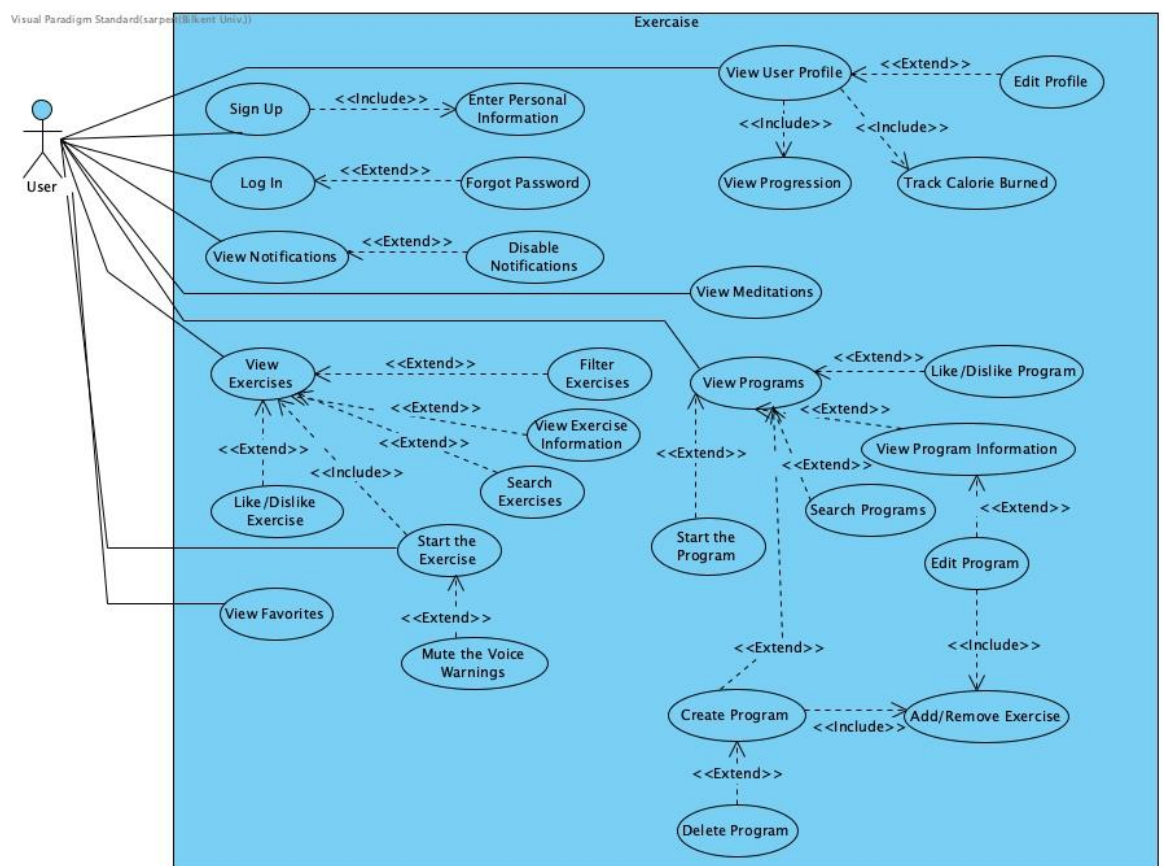


Figure 1. Use case diagram

3.5.3 Object and Class Model

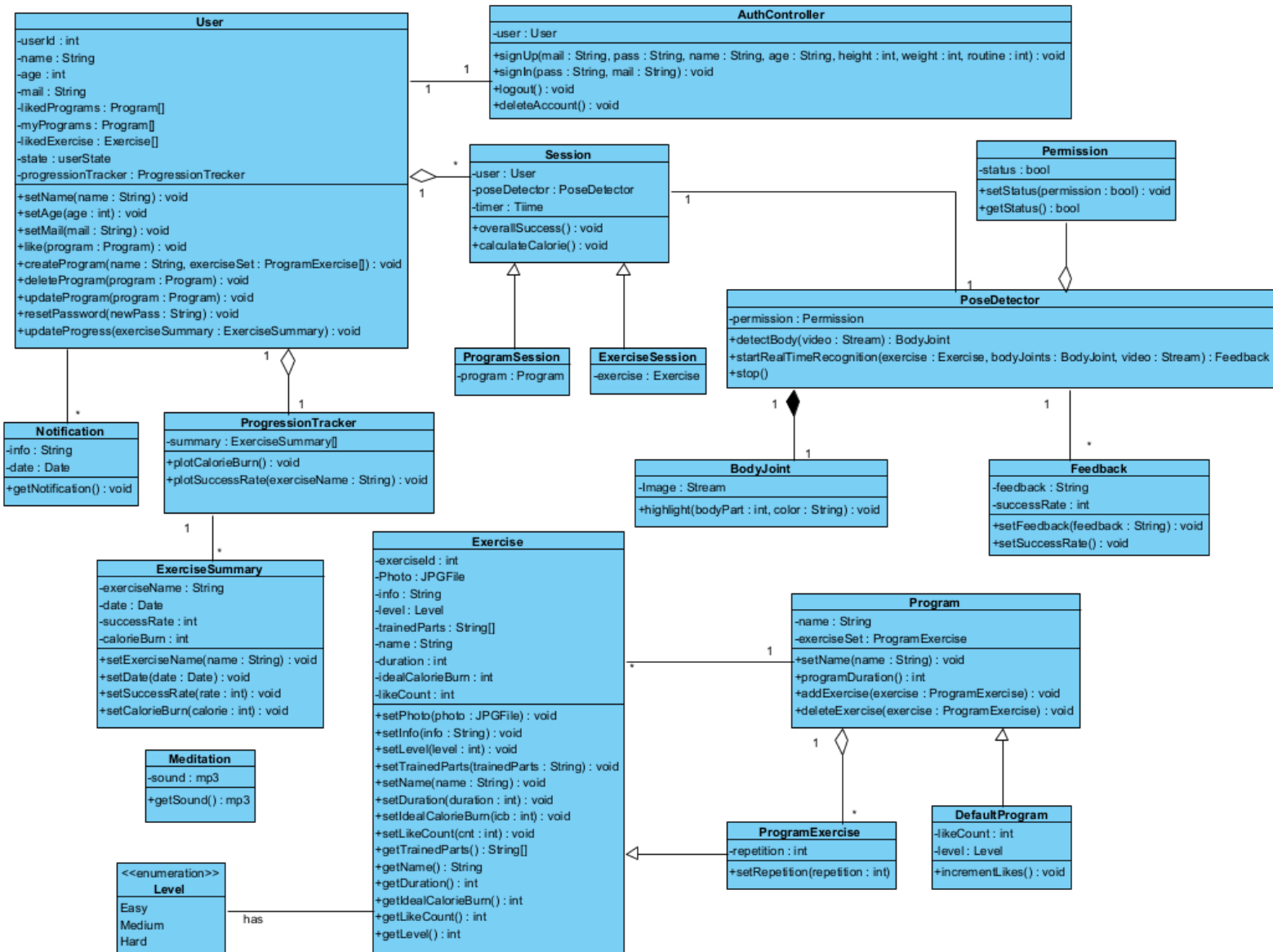


Figure 2: Class Diagram

3.5.4 Dynamic Models

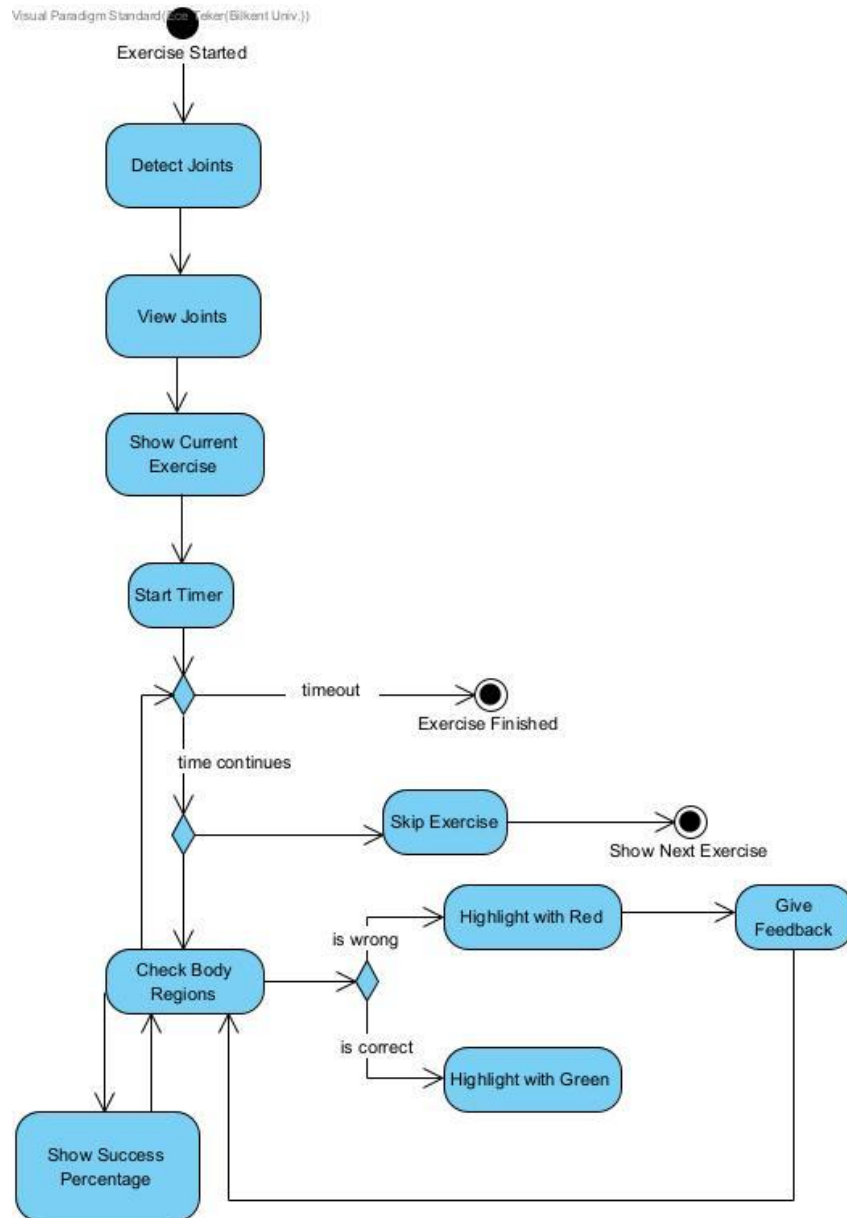


Figure 3. Exercise Activity Diagram

This activity diagram shows how the exercise screen works. When the exercise is started the joints will be detected and highlighted. After the detection, the current exercise will be displayed on the screen and hence the timer will start. If the user is doing the exercise wrong, the app will give feedback. Otherwise the user can see that they are doing it right by seeing the green colored joints. Also the user can see the success rate of their posture and/or skip the exercise. Else the exercise will finish after the timer runs out.

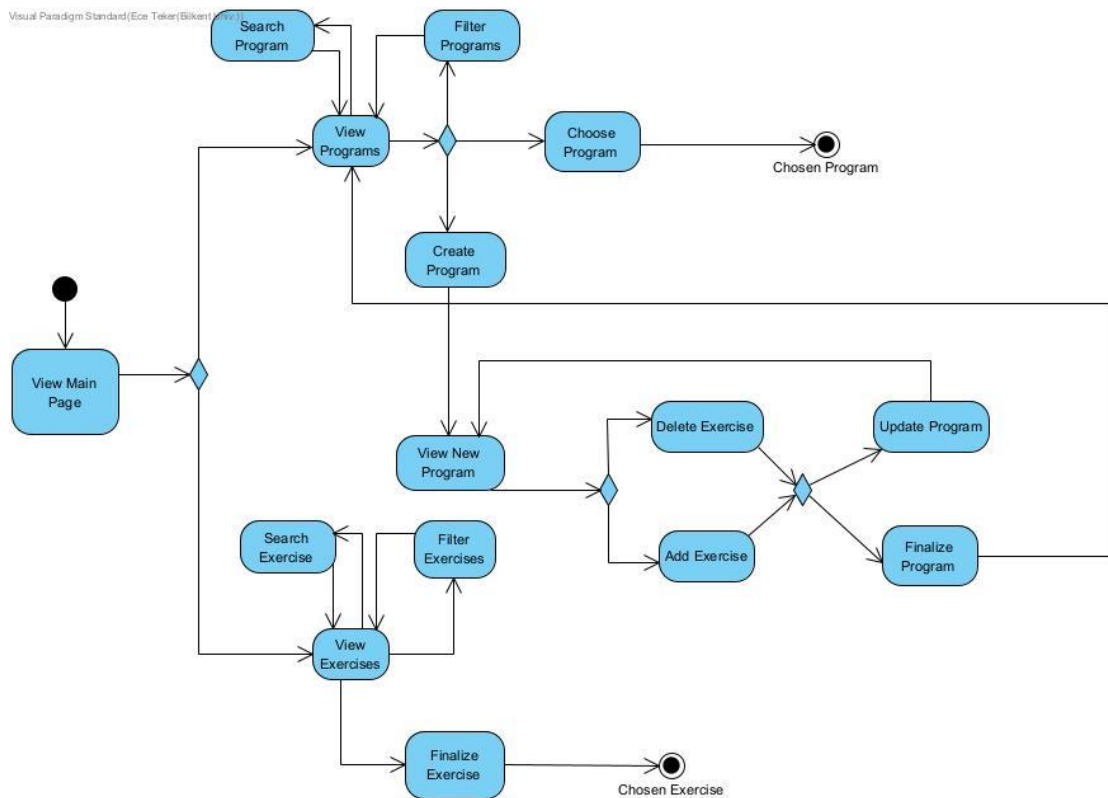


Figure 4. Program/Exercise Selection Activity Diagram

This activity diagram shows how a user can choose a program or an exercise. The user should navigate to the main page where they can either view the programs or the exercises. On both of the choices, they can filter the programs or use the search function to find a specific program/exercise. If the user views exercises, they can click on any exercise to perform it, hence the exercise will be chosen. If the user views the programs, they can choose any of the default programs or create a new program. If the user wants to create a new program, they can add exercises from the list or take any of them out as they like. After the user finalizes the program, it will be added to the programs page where they can choose the new program from.

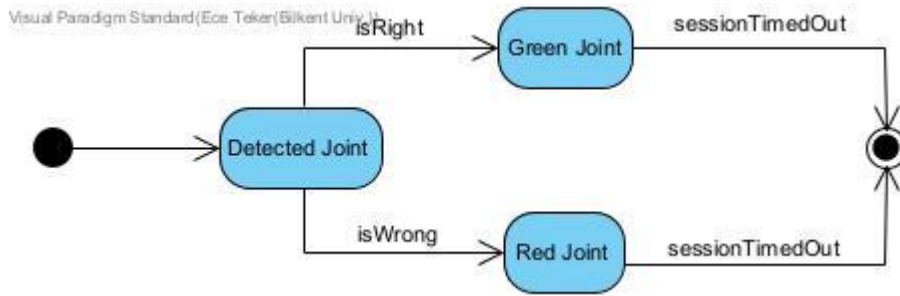


Figure 5. Body Joint State Diagram

This state diagram shows the state of a detected joint. If the user's joint is in the right position it will be recognized as a green joint, otherwise it will be recognized as a red joint. When the exercise session times out, the state will end.

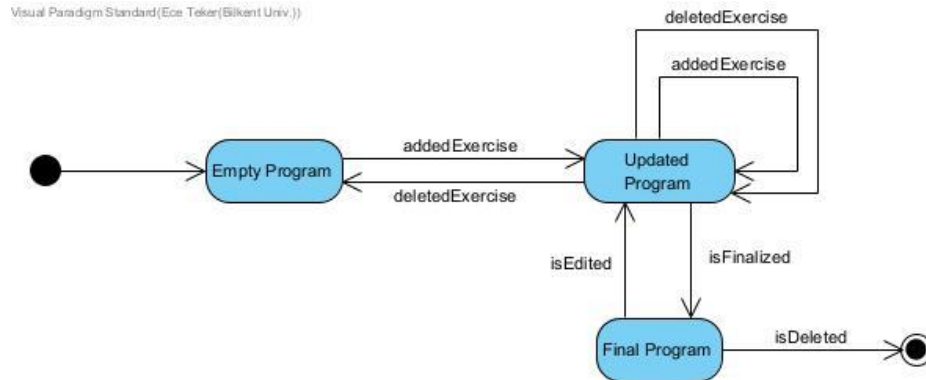


Figure 6. Program State Diagram

This state diagram shows the state of a program. A new program starts as an empty program. When the user adds new exercises or deletes the added exercise it will turn into an updated program. When the program is finalized it will turn into a final program which can be edited later. The state will end if the program is deleted.

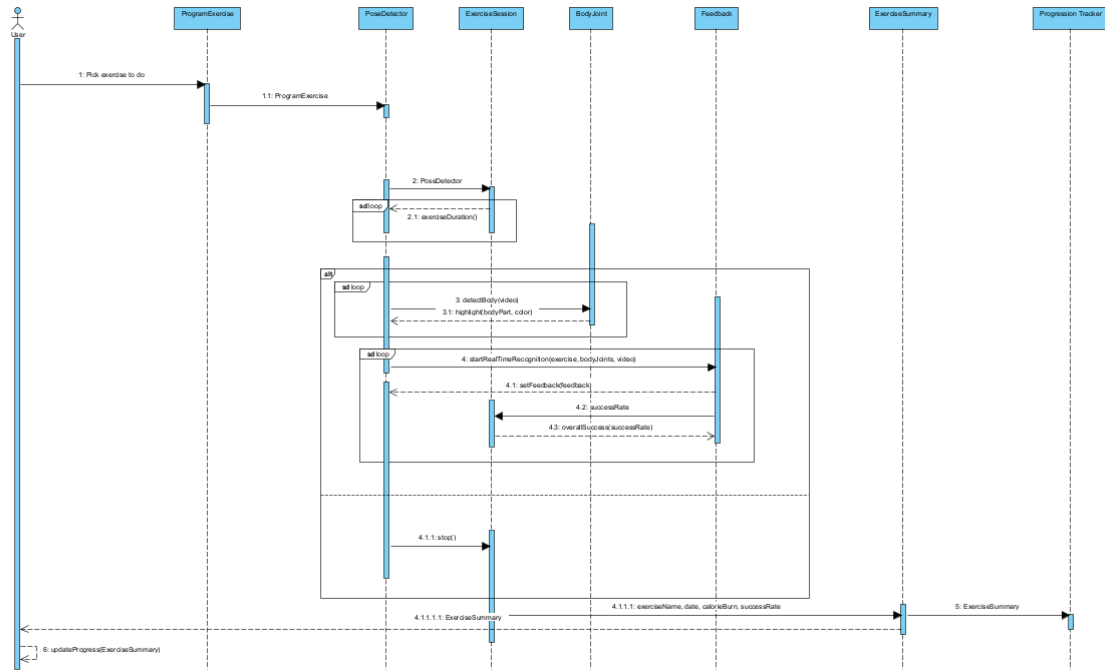


Figure 7: Exercise Sequence Diagram

This sequence diagram shows the process of doing an exercise. The diagram starts with the user choosing an exercise to do. After the user chooses the exercise, the body detection starts. Pose detector starts the detection and the exercise session starts. During the exercise session, the duration of the exercise is sent to the pose detector continuously. If the timeline of the specific exercise is not over, body joints of the user are detected and highlighted. If the user is doing the exercise correctly, the joints are highlighted green and red otherwise. Also, the success rate is continuously updated and shown to the user. After the exercise is over, an exercise summary is created and sent to the Progress Tracker class.

3.5.5 User Interface

Welcome Screen

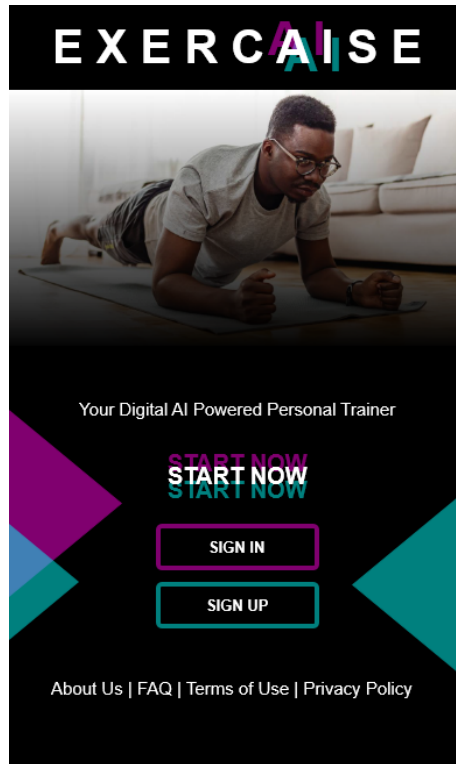
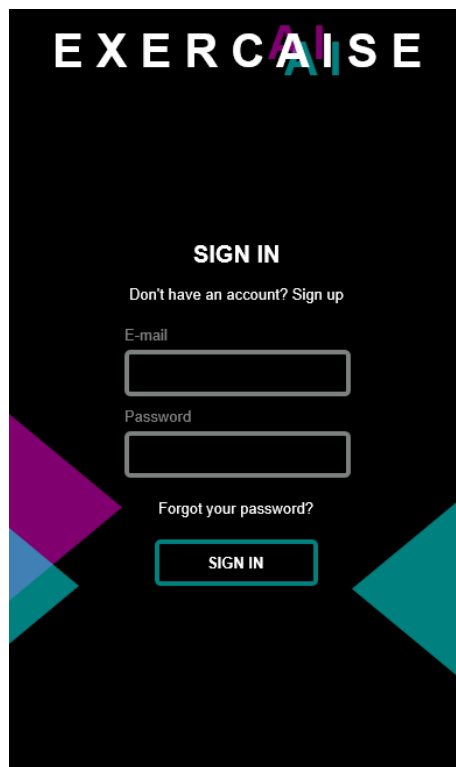


Figure 8: Welcome Screen

This is the screen user will first see when they open their app if they are not already logged in. This screen includes sign in and sign up buttons, about us, frequently asked questions, terms of use, and privacy policy links.

Sign In Screen

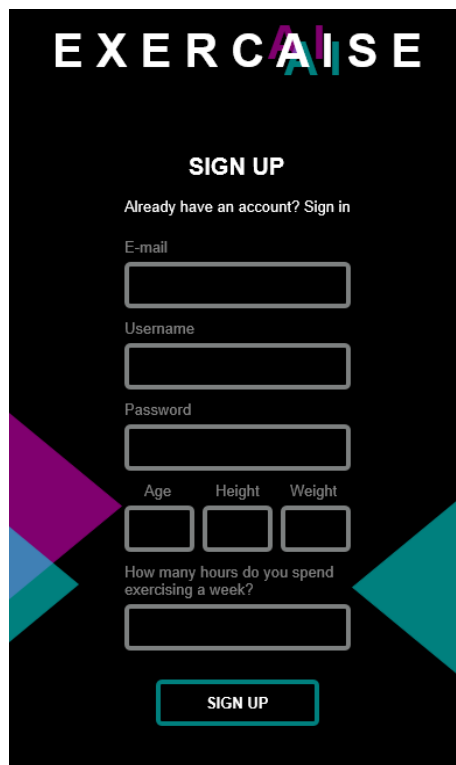


The image shows a sign-in screen for an application named "EXERCISE". The background is black. At the top, the word "EXERCISE" is written in white, with the "A" and "I" in the word "EXERCISE" being stylized with pink and teal colors. Below the title, the text "SIGN IN" is centered in white. Underneath, there is a link "Don't have an account? Sign up" in a smaller white font. The form consists of two input fields: "E-mail" and "Password", both with white borders and labels in white. Below the password field is a link "Forgot your password?" in a smaller white font. At the bottom of the form is a white button with the text "SIGN IN" in black. The screen is decorated with large, colorful geometric shapes: a pink triangle on the left and a teal triangle on the right, both pointing towards the center.

Figure 9: Sign in Screen

This is the sign in screen where the user will have to fill in the fields to sign in to our system. If a user has forgotten their password, they can restore it by clicking on the “Forgot your password?” link. Also, if the user does not have an account, they can navigate to the sign up screen via this page.

Sign Up Screen



The image shows a sign-up screen for an application titled "EXERCISE". The background is black with colorful geometric shapes (purple, teal, and blue) on the left and right sides. The text "EXERCISE" is at the top in white, with the "A" and "I" in purple and teal. Below the title, the text "SIGN UP" is centered. Underneath, there is a link "Already have an account? Sign in". The form consists of several input fields: "E-mail", "Username", "Password", "Age", "Height", "Weight", and "How many hours do you spend exercising a week?". Each field is a white rectangle with a thin border. At the bottom, there is a "SIGN UP" button with a teal border and white text.

EXERCISE

SIGN UP

Already have an account? Sign in

E-mail

Username

Password

Age Height Weight

How many hours do you spend exercising a week?

SIGN UP

Figure 10: Sign Up Screen

This is the sign up screen where the user will have to fill in their credentials to create an account on our system. Also, if the user already has an account, they can navigate to the sign in screen via this page.

Home Screen

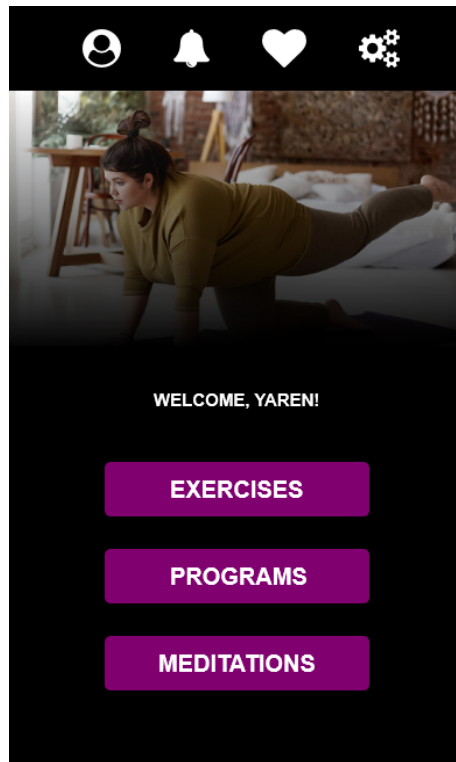


Figure 11: Home Screen

Home screen is the main navigation page of the application. It includes the links to the user profile, notifications, the user's liked exercises and programs section, and the settings. The home page also gives the user the options to view exercises, programs and meditations. The user can select to view and do a single exercise via the exercises option. Various workout programs are available in the programs section. The user can also choose to do meditations which are audio only.

Profile Screen

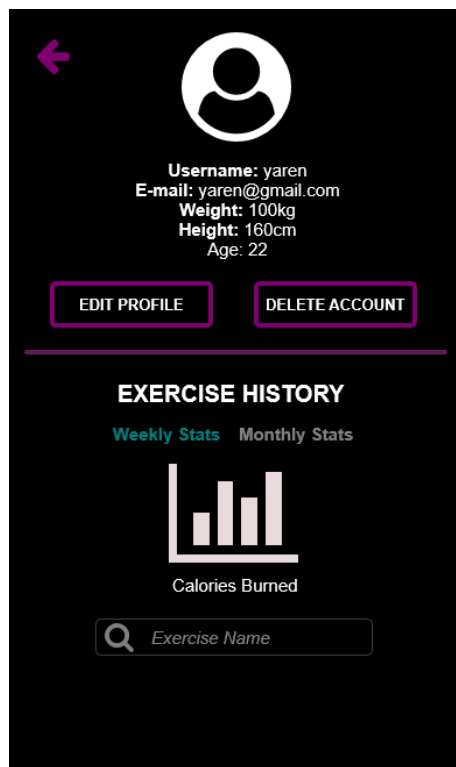


Figure 12: Profile Screen

This is the profile screen where the user can see their account information, exercise history and statistics. The user can navigate to edit their profile or delete their account via this screen. Exercise history can be viewed as weekly or monthly statistics with graphs concerning calories burned, accuracy, and time spent exercising. The user can also choose to view their statistics about a single exercise via search bar. The back button takes the user back to the home screen.

Notifications Screen

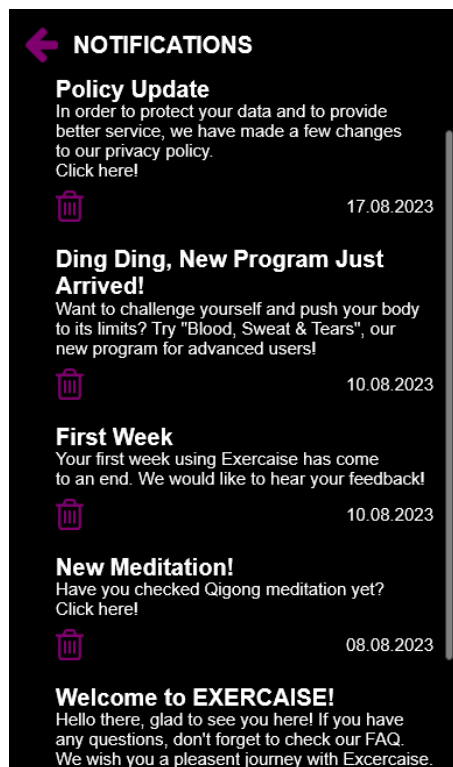


Figure 13: Notifications Screen

This is the notifications screen where the user can see their notifications and remove them if they want to. All types of notifications sent by the application regarding updates, policy changes, newly added programs and exercises drop into the user's notification board. The back button takes the user back to the home screen.

Liked Screen

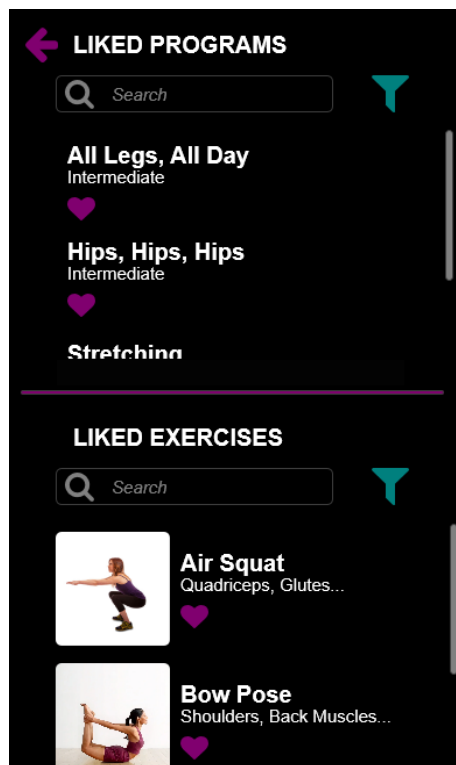


Figure 14: Liked Screen

This is the liked screen where the user can see their liked programs and liked exercises. The user can utilize the search and filter functionalities in order to find a specific program or exercise. Also, by clicking on the icon or the name of the liked item, the user can navigate to the details page of the said item. The back button takes the user back to the home screen.

Exercises Screen

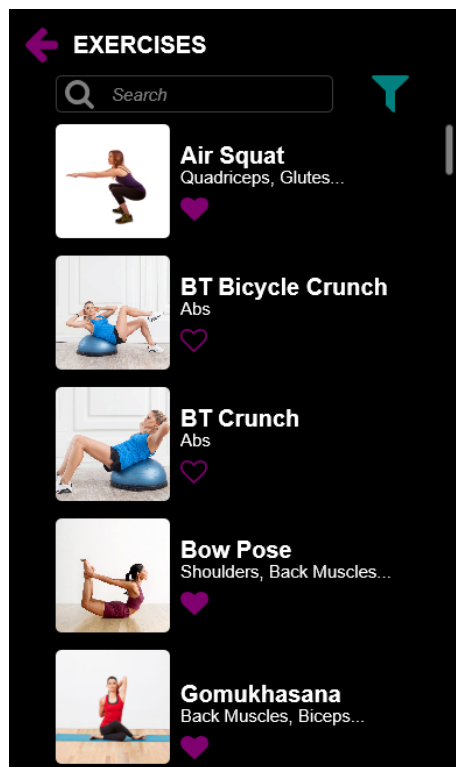


Figure 15: Exercises Screen

The user can see all of the exercises available in our application in the exercises screen. Every exercise is displayed with the name of the exercise, the target muscle groups, and an icon which visualizes the exercise. The user can choose to filter the exercises using the filter icon, and search for specific exercises using the search bar. The user can also choose to like an exercise by clicking the heart button corresponding to the said exercise. This way, the liked exercise can be found on the user's liked page. When the user clicks on the name or the icon of an exercise, they can view the details of the corresponding exercise. The back button takes the user back to the home screen.

Filter Exercises Screen

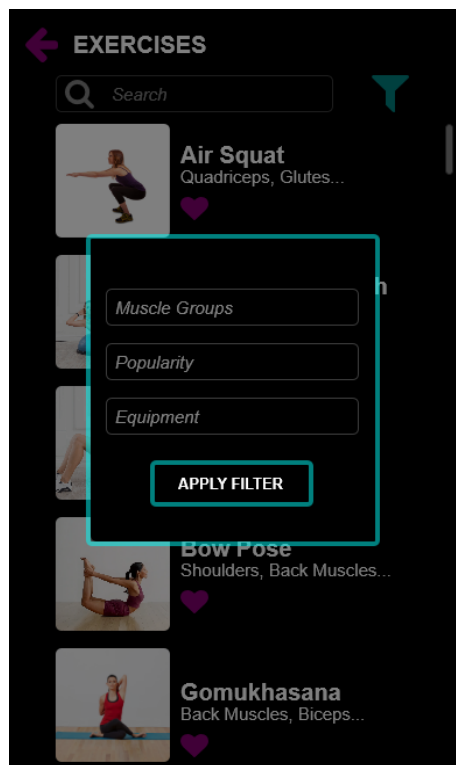


Figure 16: Filter Exercises Screen

This is the screen that appears when the user wants to filter the exercises. The user can filter the exercises according to their target muscle groups, their popularity or the equipment used. More than one filter can be applied at a time. Clicking on anywhere outside the filtering box takes the user back to the exercises screen.

Exercise Details Screen

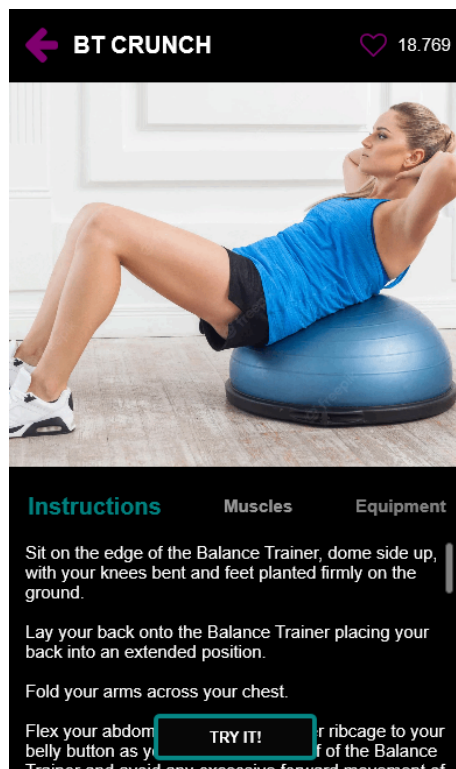


Figure 17: Exercise Details Screen

Details screen of an exercise gives detailed instructions on how to perform the exercise, the target muscle groups, and the equipment required to perform the exercise. The user can navigate between these informations by using the navbar below the exercise image. By using the “Try it!” button, the user can attempt to perform the exercise. The back button takes the user back to the exercises screen.

Programs Screen

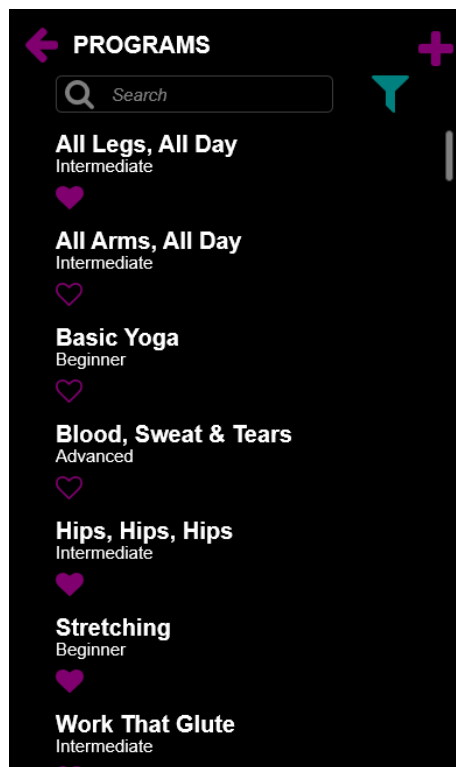


Figure 18: Programs Screen

This is the programs screen where the user can see all of the programs available in our application. The user may apply filters to the program list, search for a specific program and like a program to check it out later. The user can also select a program to see the details of it. The back button takes the user back to the home screen.

Filter Programs Screen

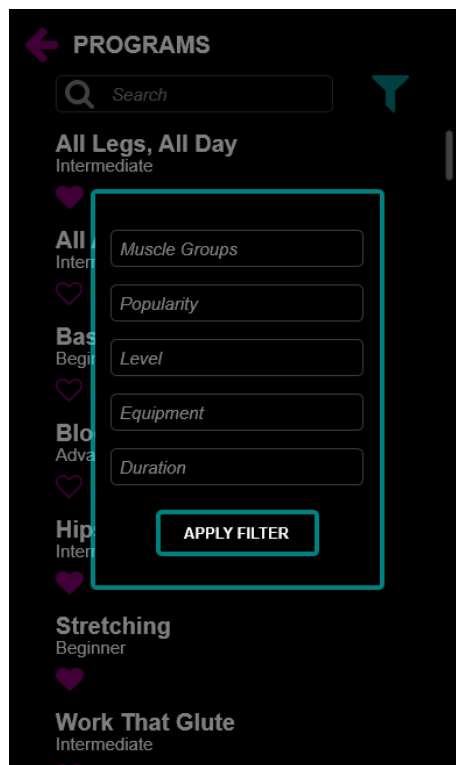


Figure 19: Filter Programs Screen

This is the filtering screen of the programs. Here, user can filter the programs according to the muscle group they work, the popularity of them, the level of the programs, the equipment required and lastly the duration of the program. After the user presses the Apply Filter button, the programs that fit to the criteria are returned on the screen. Clicking on anywhere outside the filtering box takes the user back to the programs screen.

Program Details Screen

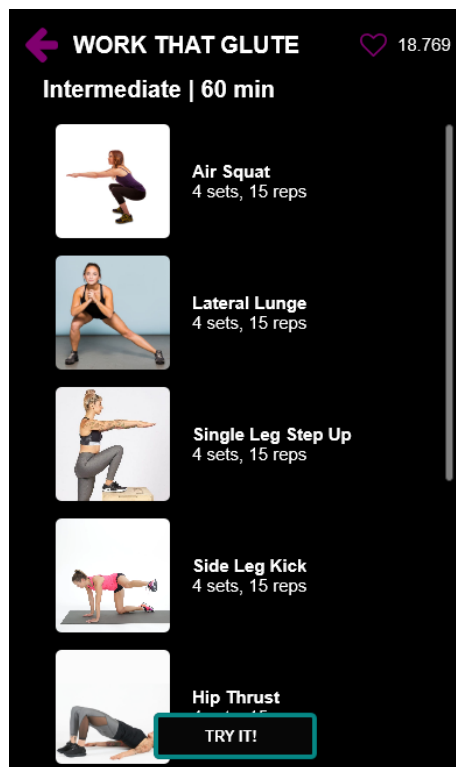


Figure 20: Program Details Screen

This is the program details screen to which the user navigates through clicking on a program in the programs screen. The user sees here the sequence of the program, the exercises in it and their set and repetition count. From here, the user can directly start performing the program.

Creating Program - Select Exercises Screen

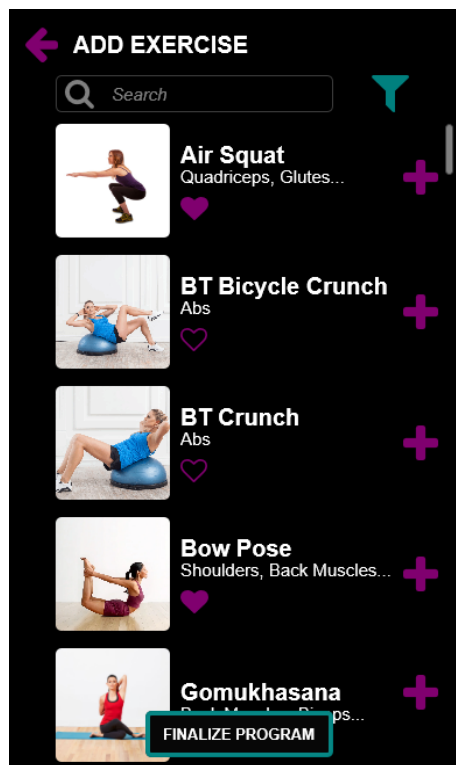


Figure 21: Creating Program - Select Exercises Screen

This is the screen where the user will select exercises to add their newly created program. The user can see the exercises available in our application and press the plus button near it to add to their program. The user can also filter exercises and search exercises here too.

Creating Program - Add Exercise Screen

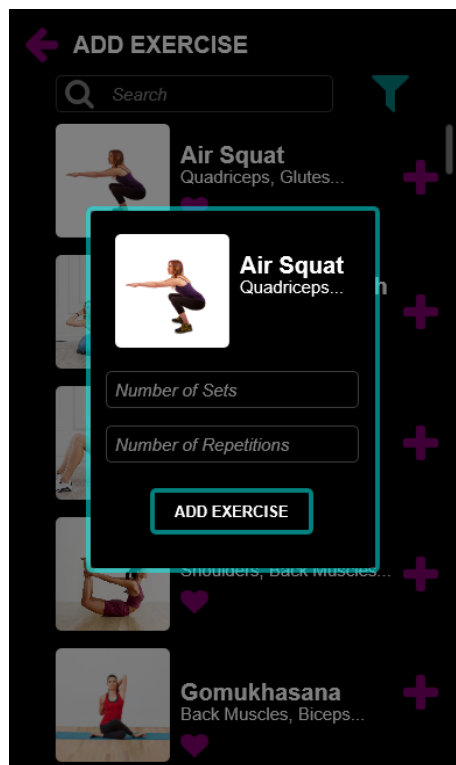


Figure 22: Creating Program - Add Exercise Screen

This is the screen where the user will see while they are trying to add an exercise to their program. They have to fill in the fields number of sets and number of repetitions and after pressing the add exercise button, user will add the exercise with the specified number of sets and repetitions to their program.

Doing Exercise - Wrong

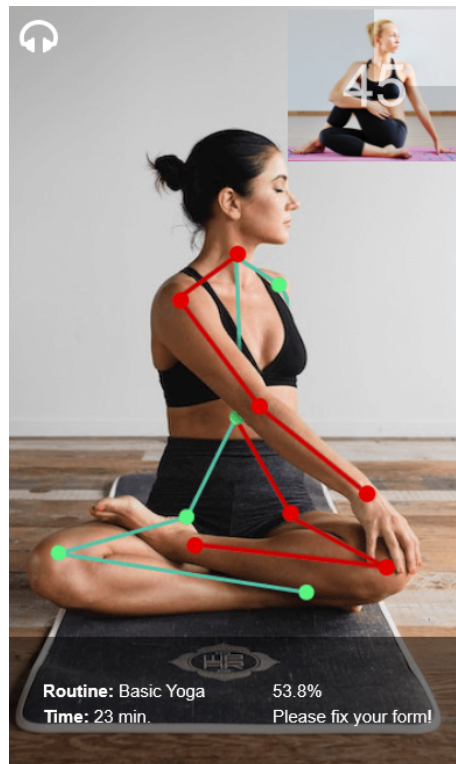


Figure 23: Doing Exercise - Wrong

This is the screen the user will see when they are doing the exercise wrong. The body parts that the user is performing wrong are colored to red while the correct ones are highlighted with green. The user will be given feedback audibly and furthermore they can see the written feedback on their screen to fix their posture. The remaining time for the pose can be seen at the right corner of the screen.

Doing Exercise - Correct

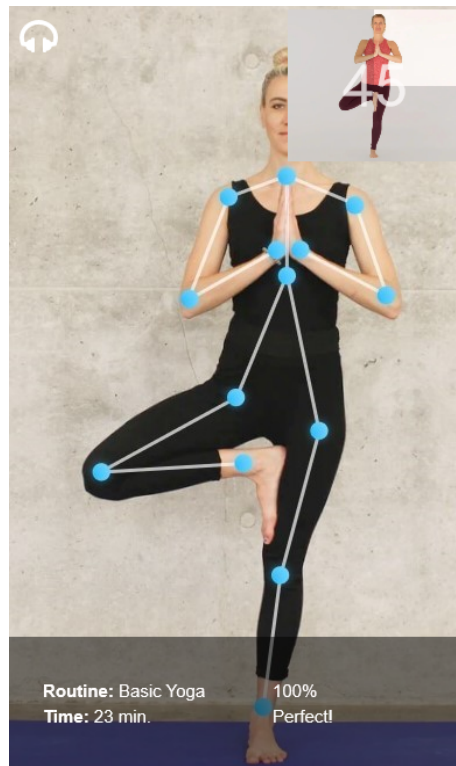


Figure 24: Doing Exercise - Correct

This is the screen the user will see while they are doing the exercise in the correct way. The body joints of the user will be colored green. The user can view time left at the right corner.

Program Summary

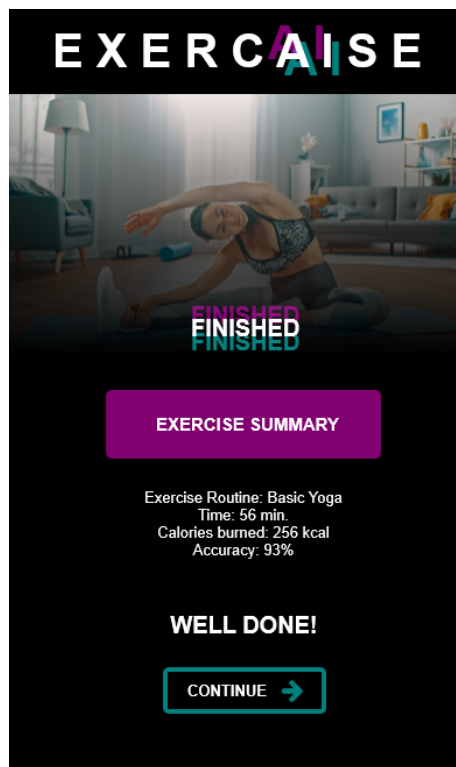


Figure 25: Program Summary

This is the program summary screen that the user will see after they complete the program. This screen indicated details such as the completed routine, the time passed, the calories burned and lastly the accuracy of the movements.

Meditation Screen

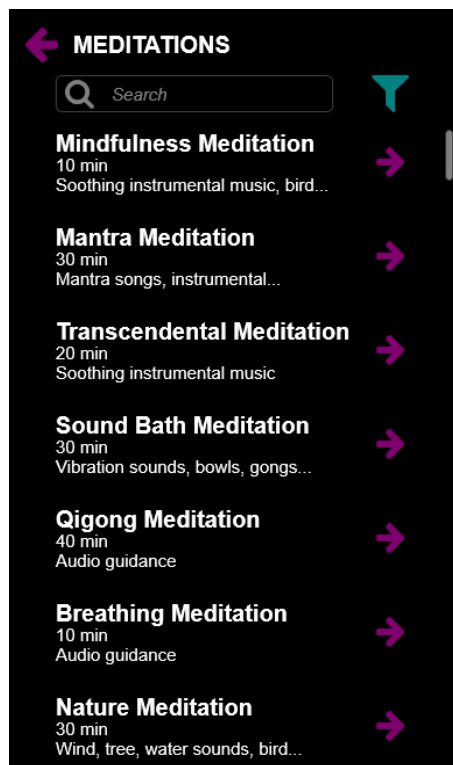


Figure 26: Meditation Screen

This is the meditation screen where the user can see the meditations available in our application. The user can filter the meditations, search for a specific meditation and start the meditation from this screen.

4 Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

Exercaise is mainly focused on providing a safer and healthier option for exercising. Since it involves tracking calories, body-region specialized programs and meditations; it has a massive effect on public health. It gives the user a chance to exercise without going to crowded places such as gyms and courses so it also affects public health by lowering the risk of pandemic. People with mental illnesses such as anxiety, agoraphobia and depression will be able to have a quality time exercising at home so the app also has social impacts. There are also some risks concerning public health, public safety, public welfare, global factors, cultural factors and social factors. Below are some of the impacts of Exercaise that will be taken into consideration on the design stage.

Table 01: Factors that can affect analysis and design.

	Effect level	Effect
Public health	8	Injuries, mental issues.
Public safety	0	No evident effect.
Public welfare	0	No evident effect.
Global factors	7	Privacy issues.
Cultural factors	1	Exploiting the ancient yoga culture.
Social factors	3	Decrease in social interactions.

4.1.1 Public Health

Exercaise is mainly an app that helps people to exercise wherever and whenever they want with its specialized programs. The app will have a feedback system which will correct the user while doing the exercises and minimizes the common workout injuries such as muscle pull and strain or sprained ankle so it is up to the user to follow the feedback of the program correctly. Because of the app's real-time feedback system, people will be able to correct their forms and track their progress. Each exercise or exercise set will have a difficulty level which will be easy, medium or hard. There will be warnings if a beginner is not suited for the exercise or flexible enough for a yoga pose. If the users chose to do a hard exercise program as a result of not taking notice of the warnings and the difficulty levels, they could be faced with injuries. In addition, even though the exercises or exercise sets have difficulty levels, the levels might not be suitable for each user. The users can also filter the exercises or exercise programs according to their duration so they will be able to choose the best exercise for their needs and desires. The app will also provide the information of which exercise works which body regions. This feature is very helpful for users with injuries in a specific part of their bodies.

The effect level of public health in our project is 8. There are many health risks to consider in the designing phase of the project. The risk of permanent

or temporal injuries may be high so the pose detection model will be designed accordingly.

4.1.2 Public Safety

In terms of public safety, Exercaise will not have any evident effect. The app does not take any actions considering the safety of the public so this factor will not be taken into consideration in the design phase of the project.

The effect level of public safety in our project is 0 since the app does not have any evident effect on public safety.

4.1.3 Public Welfare

Similar to public safety, Exercaise does not have any negative impacts on public welfare. As a matter of fact, the app will provide high quality and efficient exercise time for the users.

The effect level of public welfare in our project is 0 since the app does not have any evident effect on public welfare.

4.1.4 Global Factors

The app will comply with the Google Play Developer Distribution Agreement and follow the standards of the Google Policy Center which ensures that the applications do not violate the privacy, intellectual property, do not contain any kind of malware and do not include restricted content [1]. Privacy is important since the app requires personal information of each user such as their gender, age, height and weight. Exercaise uses the phone camera while the user is exercising so any exploitation to malwares might create a significant privacy risk.

The effect of global factors is 7 since the user privacy is a global problem that affects people more and more each day.

4.1.5 Cultural Factors

The app provides a yoga and meditation experience for the users besides the main exercises. The users will benefit from ancient yoga teachings and meditation techniques but there is a risk of exploiting the Indian culture. We

will use the guidance of commonly accepted yoga resources so the risk will be minimized but the dataset has to be chosen according to the cultural factors.

The effect of cultural factors is 1 since yoga and meditation has many official resources that will be taken into consideration on the design stage of the app.

4.1.6 Social Factors

Exercaise will provide the users with quality exercise options whenever and wherever they want. Even though this will help in reducing the effects of pandemic, people might become more dependent on their homes. Since the app removes the necessity of being in a gym or a yoga course with other people, users may tend to stay at home and exercise alone. This might create a risk of decrease in social interactions.

The effect level of social factors is 3 since the users can socialize even though they prefer to exercise alone in their homes.

4.2 Risks and Alternatives

Below are given the risk factors that have been considered and some alternatives as solutions if a problem regarding these risks were ever to occur.

Table 02: Risks

	Likelihood	Effect on the project	B Plan Summary
Positioning and Video Quality	%35	Increase in classification time and decrease in accuracy, so decrease in usability.	Resolution enhancer models will be used for lighting and quality issues. The user will be warned to change their position.
Real-Time Feedback	%30	Slow or inaccurate model.	The model will be developed with a lower scale or new datasets will be used. The user will be warned about the connection issues.
Data Privacy	%10	Privacy issues due to sensitive information.	Data will be encrypted and will not be shared with 3rd party applicants.

Computational Power	%40	Disruption on functionality.	Efficient models and algorithms will be used. The user will be advised to use a more powerful device.
---------------------	-----	------------------------------	---

Positioning and Video Quality

The positioning of the camera is very significant since our pose detection model works based on the similarities between the training data and the user. The user has to place the camera in a place where there is suitable lighting and frame quality so the body joints will be detected correctly and the accuracy of the model will be high.

Alternative: Before the exercise starts, the camera will open and the user will change his/her position until the app decides that it can detect the body joints accurately. There might be some warnings about which body part is not recognized or the user has to come forward/backwards. When the system detects the body joints, it will ask if the detection is made correctly and the user is ready to start. Since the model will be trained based on samples from many angles and lightning, the model will be able to recognize the user from different angles and with different lightning. Also, the datasets that will be used in training contain many people from different race, gender, age and body type so the user can be recognized no matter their background or identity.

Real-Time Feedback System

The app will have a real-time feedback system which will notify the user either vocally or verbally. This system has to work very fast and accurately since the user will be dependent on the feedback and could potentially injure themselves because of a wrong or slow detection.

Alternative: If the feedback system gives incorrect output because of connection issues between the user and the system, , the user will be alerted and the data will be stored in the local hardware which will be sent to the device as soon as the connection is reestablished. If the feedback system gives inaccurate results, the model will be retrained with different datasets and new methods. Also, we can reduce the scale of our trained model to have better performance.

Data Privacy

The app will ask the user for various personal data such as their age, name, height or weight. In addition, the exercise progression and the calorie burn of the user will be stored. The app will ask for permission to camera from the user and will record the user in order to give real-time feedback. It is a high priority task to protect the privacy of these data.

Alternative: Even though the app asks the user for personal information such as name, age, exercise progression, calorie burn etc., this information will not be shared with any 3rd party applications. While the real-time feedback system is working, body images of the users are used but it will not be stored on cloud.

Computational Power of the Used Device

Exercaise will compute visual data which requires an effective computational power. Also, the app will give real-time feedback and follow the progression of the user. Since the app is a mobile application, if the user's device does not have enough computational power, it may cause a bottleneck for the application and the computations might take too long to complete.

Alternative: If the used device is dealing with computational issues and it takes too long for the computations to complete, the user will be alerted and advised to move on to another device. In order to not encounter such problems, Exercaise will be developed using efficient models and algorithms.

4.3 Project Plan

Under this section, we separated the tasks into packages, and assigned a team with a team leader for each package. In order to make the plan more apprehensible, tables for the project plan were prepared and are given below.

Table 03: List of work packages

WP#	Work package title	Leader	Members involved
WP1	Reports	Sarper Turan	Ece Teker, Lara Özyeğen, Yaren Durgun, Selen Görgün
WP2	Frontend Development	Yaren Durgun	Sarper Turan
WP3	Backend Development	Ece Teker	Lara Özyeğen, Selen Görgün
WP4	Pose Detection	Selen Görgün	Sarper Turan
WP5	Visual Joint Feedback	Lara Özyeğen	Yaren Durgun
WP6	Audio Joint Feedback	Sarper Turan	Ece Teker, Selen Görgün
WP7	Demo Development 1	Lara Özyeğen	Ece Teker, Yaren Durgun, Selen Görgün, Sarper Turan
WP8	Demo Development 2	Yaren Durgun	Ece Teker, Lara Özyeğen, Selen Görgün, Sarper Turan
WP9	Testing	Ece Teker	Sarper Turan, Selen Görgün, Lara Özyeğen, Yaren Durgun

WP 1: Reports			
Start date: 03.10.22 End date: 15.05.23			
Leader:	Sarper Turan	Members involved:	Yaren Durgun, Lara Özyeğen, Ece Teker, Selen Görgün
<p>Objectives: The main goal of the reports is to put some constraints and make decisions about the project beforehand so that the project can be built without ambiguity and speed up the process. The objectives include the project's requirements, analysis of the requirements, and detailed design.</p>			
<p>Tasks:</p> <p>Task 1.1 Brainstorm Ideas: Try to understand the needs of the project and decide on specific functionalities for the project.</p> <p>Task 1.2 Requirements Elicitation: Research and decide on functional and non-functional requirements in order to meet the expectations of the application.</p> <p>Task 1.3 Specification Report: The report is a non ambiguous documentation of requirements of the project. It can be used as a guideline by the developers while building the application.</p> <p>Task 1.4 Analysis Report: The report includes the UI design, scenarios, use cases, models such as object, class, and dynamic. It also includes the project plan which will be helpful through the project.</p> <p>Task 1.5 Detailed Design Report: The report is to show class packages and interfaces.</p> <p>Task 1.6 Final Report: The report includes the packages, tools and technologies, frameworks that have been used while developing the project. Overall design will be included as well.</p>			
<p>Deliverables</p> <p>D1.1: Project Website</p> <p>D1.2: Project Specification Report</p> <p>D1.3: Analysis Report</p> <p>D1.4: Detailed Design Report</p> <p>D1.5: Final Report</p>			

WP 2: Frontend Development			
Start date: 07.11.22 End date: 15.05.23			
Leader:	Yaren Durgun	Members involved:	Sarper Turan
<p>Objectives: The purpose of frontend development is to create a user interface which satisfies all the functional, non-functional, and UI requirements. All visual elements of the application and user interaction tools are a part of the frontend development objectives.</p>			
<p>Tasks:</p> <p>Task 2.1 Welcome Screen: Brief introduction of the mobile application with the options to view frequently asked questions, terms of service, and privacy policy.</p> <p>Task 2.2 Sign Up and Sign In Screens: The interactive pages which will allow users to sign up and sign in by working with the backend logic.</p> <p>Task 2.3 Home Page: Personal home page of a user which will offer options to view personal exercise history, exercises, and programs.</p>			

Task 2.4 Exercises Screen: A list of all exercises registered in the system and the means to view the exercise details, like a particular exercise, and search and filter the exercises.

Task 2.5 Exercise Details Screen: The details of a particular exercise, with the instructions to perform the exercise, the target muscle groups, and the equipment needed for the exercise.

Task 2.6 Programs Screen: A list of all programs registered in the systems and the level of the program, with the means to view program details, like a particular program, and search and filter the programs.

Task 2.7 Program Details Screen: The details of a particular program, the exercises included in the program, number of sets and repetitions, with the expected duration of the program.

Task 2.8 Filtering Screen: The option to filter the exercises regarding the target muscle groups and equipment, and the programs regarding the level, the muscle groups, equipment, expected duration, and exercises included in the program.

Task 2.9 Exercise History Screen: A brief history of the user's progress while using the application, reinforced with weekly and monthly graphs.

Task 2.10 Workout Screen: The distinct feature of the application. Design and visuals of the screen that will monitor the user during workout.

Task 2.11 Profile: An interactive page that will allow the user to view and edit their information including username, password, height, weight, exercise habits.

Deliverables

D2.1: User Interface

WP 3: Backend Development

Start date: 07.11.22 **End date:** 15.05.23

Leader:	Ece Teker	Members involved:	Lara Özyeğen, Selen Görgün
----------------	-----------	--------------------------	----------------------------

Objectives: The main goal of backend development is to create the means to manage application data by designing and implementing a database model .

Tasks:

Task 3.1 User Data: Design and implement the database schema that will manage the creation, saving, update, retrieval, and deletion of the user data.

Task 3.2 Exercise Data: Design and implement the database schema that will manage the search operations, saving, update, and retrieval of the exercise data.

Task 3.3 Program Data: Design and implement the database schema that will manage the search operations, saving, update, and retrieval of the program data.

Task 3.4 Exercise Summary Data: Design and implement the database schema that will save, get the exercise summary information of a particular user.

Deliverables

D3.1: Database model

D3.2: Implementation of database model

D3.3: Database Management tools

WP 4: Pose Detection			
Start date: 14.11.22 End date: 27.11.22			
Leader:	Selen Görgün	Members involved:	Sarper Turan
Objectives: It is required to detect the whole body before the start of a workout session in order to give feedback afterwards. In this package joints of a full body will be detected accurately. Machine learning libraries will be used for this purpose.			
Tasks: Task 4.1 Train Pose Detection Model: A model will be trained using a labeled joints photos dataset. Movenet library will be used to detect the joints of the body. Task 4.2 Evaluate Model Performance : Model performance will be evaluated using the test dataset. Task 4.3 Detect poses live: The model's performance on a phone camera will be tested.			
Deliverables D4.1: Pose Detection Model D4.2: Pose Detection feature			

WP 5: Visual Joint Feedback			
Start date: 28.11.22 End date: 11.12.22			
Leader:	Lara Özyeğen	Members involved:	Yaren Durgun
Objectives: Using the pose detection model, the user should be able to see which joint of their body is in correct form or not by the different highlighted joints (Red indicates wrong joint form while green indicates it is correct) .			
Tasks: Task 5.1 Model Training: Train a model to calculate how the correct form should be and display the wrong and correct joints on the screen by highlighting the joints red or green. Task 5.2 Model Performance : Observe the performance of the model by calculating the accuracy by providing test data.			
Deliverables D5.1: Model Performance Report			

WP 6: Audio Feedback			
Start date: 30.01.23 End date: 01.03.23			
Leader:	Sarper Turan	Members involved:	Ece Teker, Selen Görgün
Objectives: Using the visual joint feedback, the model should be able to give audio feedback to users, telling them to re-do the pose by correcting the red highlighted body joints.			
Tasks: Task 6.1 Model Training: Train a model to analyze the visual joint feedback to output an audio feedback. Task 6.2 Model Performance: Observe the performance of the model by calculating the accuracy by providing test data.			
Deliverables D6.1: Model Performance Report			

WP 7: Demo Development 1			
Start date: 21.11.22 End date: 4.12.22			
Leader:	Lara Özyeğen	Members involved:	Sarper Turan, Yaren Durgun, Ece Teker, Selen Görgün
Objectives: In order to showcase the project and the development progress, a demo will be provided. This will consist of a presentation and a live demo. The aim is to make sure the demo is well executed and delivers the work in an interesting, professional manner.			
Tasks: Task 7.1 Demo Presentation : Create a powerpoint presentation that covers the description, completed features and the future work. Task 7.2 Demo Application: Have an application that has the features of pose detection and visual joint feedback. Create sample users to present in the demo and show all the completed features up to that point.			
Deliverables D7.1: Live demo of the working mobile application D7.2: Demo Presentation			

WP 8: Demo Development 2			
Start date: 01.05.23 End date: 15.05.23			
Leader:	Yaren Durgun	Members involved:	Sarper Turan, Lara Özyeğen, Ece Teker, Selen Görgün
Objectives: The second demo will involve all the features of the application. A demo will consist of a presentation and a live demo of the app. The aim is to make sure the demo is well executed and delivers the work in an interesting, professional manner.			
Tasks: Task 8.1 Demo Presentation: A powerpoint presentation that covers the description, completed features and the future work. Task 8.2 Demo Application: Have an application that has the features of pose detection, visual joint feedback and audio feedback. Create sample users to present in a demo and show all the features of the application.			
Deliverables D8.1: Live demo of the working mobile application D8.2: Demo Presentation			

WP 9: Testing			
Start date: 14.11.22 End date: 15.05.23			
Leader:	Ece Teker	Members involved:	Sarper Turan, Selen Görgün, Lara Özyeğen, Yaren Durgun
Objectives: Test the joint recognition, visual and audio feedback, ML models and recognition of poses. Furthermore, test the frontend and backend of the application to observe any possible errors and issues.			
Tasks: Task 9.1 Test Joint Recognition: The joint recognition will be tested through mobile application to see if it is giving accurate results. Task 9.2 Test Visual and Audio Feedback: The feedback features will be tested through mobile application to see if it is giving accurate results. Task 9.3 Test ML models: The models that have been used for the project will be tested. Task 9.4 Test Mobile Application: The mobile application will be tested to see if there are any bugs or issues within the frontend or the backend.			
Deliverables D9.1: Test results D9.2: Bug fixes			

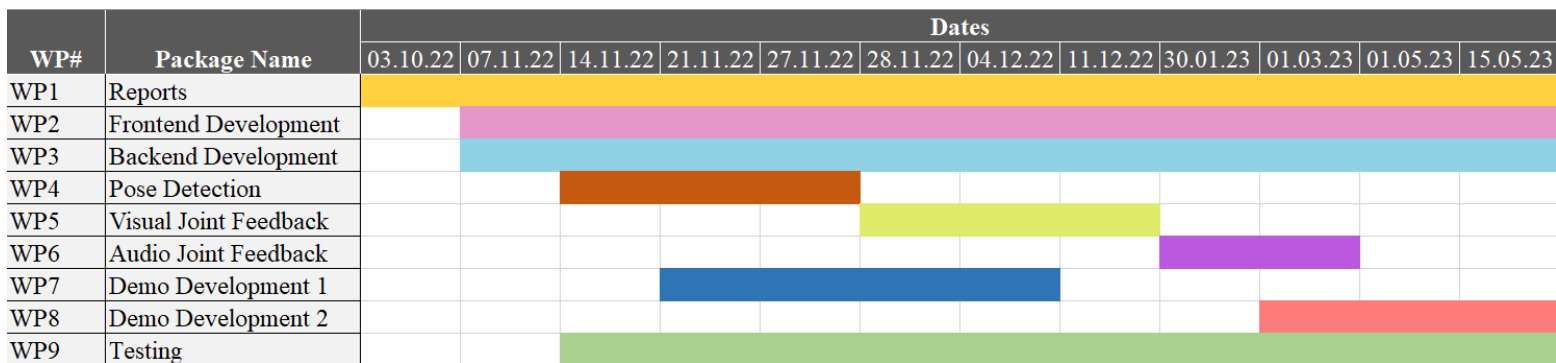


Figure 27: Gantt chart displaying the packages and their assigned time periods

4.4 Ensuring Proper Teamwork

First of all, to ensure proper teamwork and for everyone to be aware of each other's performances; we will use a Kanban Agile board using Github's Projects feature. As a team, we value communication and we believe that we should be able to communicate the responsibilities and do the distribution accordingly. We will have a meeting every 2 weeks to create tickets, then assign them democratically to each one of us so it'll be easier to keep track of our progress. As a team, it is our wish for everything to happen democratically and we want to create a safe and non-hostile environment as we believe it is important to have a productive and welcoming environment to achieve success.

4.5 Ethics and Professional Responsibilities

4.5.1 Professional Issues

- The application's repository will be a private one.
- There will be weekly meetings to make sure the project is on the track and then as the time progresses we will start to have more frequent meetings to be aligned with Agile methodologies and to have better progression.
- The project tasks will be distributed equally and democratically.
- Meeting logs will be kept in order to check the contribution of every member.

4.5.2 Ethical Issues

- The camera data from the application will not be stored or published in any way. Frames will only be sent to cloud for pose recognition and then they will be deleted.
- Profile information of the users will not be shared.
- Only necessary and pertinent data will be collected. The users will be informed about what data is collected and why.

4.6 Planning for New Knowledge and Learning Strategies

There are 3 main topics that we have to learn to implement this project. First and foremost, we will try to learn more about machine learning and how to train a model and use it. Currently 4 of us are taking machine learning and one of us has already taken it and has some experience working in machine learning. We will also do a lot of research in order to get better at the topic. Next, we will learn flutter by watching videos on it and reading the documentation. We also have to learn about the cloud as we want to deploy our model into the cloud.

5 References

- [1] “The Smartest Fitness App Yet,” AIFit.[Online]. Available: <https://aifitness.com.au/>. [Accessed: Nov. 8, 2022]
- [2] “Your Digital Yoga Coach,” Skill Yoga. [Online]. Available: <https://skill-yoga.com/>. [Accessed: Nov, 10, 2022]
- [3] “Developer Distribution Agreement,” Google play.Available: <https://play.google.com/about/developer-distribution-agreement.html>. [Accessed: Nov, 7, 2022]