



Prepared by:

Mustafa Amin Abbas Helmy

Mostafa Lotfy Mohamed Mohammed

Doaa Shaaban Mahmoud Syam

Supervisor

DR. Manal Showman

Abstract

Gym owners face many problems in their gyms like strangers enters and pretends to be a member of the gym, a member with expired subscription pretends that their subscription is still valid, new people at gym feels embarrassed for unknowing exercises, private trainers want to promote and display their skills and biography, trainers want to have a way of communicating with their trainees to provide them with help and appropriate diet to reach their goal.

Our system aims to overcome all these problems by using face recognition system to scan people at the entrance of the gym and check if they are members or not and check their subscription validation.

Admin can view all users of the gym.

Admin can Add Announcements to notify all users about any new event.

Users Sign up to have an account and use the system to view available trainers and available subscription plans then they can subscribe online using their credit card.

Users who have subscribed with a private trainer can use the chat in the system to communicate with the trainer using mobile application.

New people to the gym will not feel embarrassed anymore because they can now find exercises videos with steps that describes how to perform it on the system using mobile application.

Users can view Healthy meals and their nutrition to help them to find a suitable diet for their goal after using body calculators and consulting their trainer.

Table of Contents

Abstract	2
Abbreviations and Definitions	7
Chapter 1: Introduction	8
1. Introduction	9
1.1 Purpose	9
1.2 Overview	9
Chapter 2: Analysis and Requirements	10
2.1 System Users.....	11
2.1.1 System Stakeholders.....	11
2.1.2 Users Objectives	12
2.2 User Requirements Definitions	13
2.2.1 System Functions	13
2.2.2 Constraints	13
2.3 System Requirements	14
2.3.1 Functional Requirements	14
2.3.2 Non-Functional Requirements	17
2.4 Functional Requirements Specifications	18
2.4.1 Use Cases Description	18
Chapter 3: Software Design	31
3.1 Use case Diagram	32
3.2 Sequence Diagram	33
3.3 Entity relationship diagram (ERD)	40
3.4 Class Diagram	41
Chapter 4: Face Recognition	42
4. Face Recognition Model	43
4.1 Introduction	43
4.2 Model Architecture	44
4.3 Model Inputs	45
4.4 Model outputs	45
4.5 Model workflow	46

4.6 Model training	47
4.7 Model optimization.....	47
4.8 Model deployment.....	48
4.9 Testing cases:	49
4.10 Conclusion	52
Chapter 5: System Implementation	53
5. System Implementation	54
5.1 Website	54
5.2 Mobile Application	61
Chapter 6: Testing.....	77
6. Testing and Validation	78
Chapter 7: Development Tools and Technologies	91
7. Development tools and technologies	92
7.1 Tools and IDEs	92
7.2 Languages	93
7.3 Frameworks	95
7.4 Databases	97
Chapter 8: Conclusion	99
8. Conclusion	100

Table of Figures

Figure 1: Use case diagram	32
Figure 2.1: Registration sequence diagram	33
Figure 2.2: Login sequence diagram.....	34
Figure 2.3: Subscription sequence diagram	35
Figure 2.4: Private Subscription sequence diagram	36
Figure 2.5: Face Scan sequence diagram	37
Figure 2.6: Chat sequence diagram	38
Figure 2.7: Add Announcement sequence diagram	39
Figure 2.8: Add Income/Exercise/Recipe sequence diagram	39
Figure 3: Entity Relationship diagram (ERD)	40
Figure 4: Class diagram	41
Figure 5: Convolutional Neural Network	44
Figure 6.1: Case 1 Input	49
Figure 6.2: Case 1 Output	49
Figure 6.3: Case 2 Input	50
Figure 6.4: Case 2 Output	50
Figure 6.5: Case 3 Input	51
Figure 6.6: Case 3 Output	51
Figure 7.1: Website Empty Registration Page	54
Figure 7.2: Website Filled Registration Page	54
Figure 7.3: Website Empty Login Page	55
Figure 7.4: Website Filled Login Page.....	55
Figure 7.5: Website Home Page	56
Figure 7.6: Website Subscription Plans Page	56
Figure 7.7: Website Payment Page	57
Figure 7.8: : Website Payment Page	57
Figure 7.9: Website Profile Page	58
Figure 7.10: Website Trainers Page	58
Figure 7.11: Website Trainer's Profile Page	59
Figure 7.12: Website Admin Panel Page	59
Figure 7.13: Website Edit user Page (admin panel)	59
Figure 8.1: Mobile Application Login Screen	61
Figure 8.2: Mobile Application Registration Screen	61
Figure 8.3: Mobile Application Home Screen	62
Figure 8.4: Mobile Application App drawer	62
Figure 8.5: Mobile Application User's Profile Screen	63
Figure 8.6: Mobile Application Exercises Category Screen.....	64

Figure 8.7: Mobile Application Exercises Screen	64
Figure 8.8: Mobile Application Exercise Video Screen	65
Figure 8.9: Mobile Application Trainers Screen	66
Figure 8.10: Mobile Application Trainer's Profile Screen	66
Figure 8.11: Mobile Application Conversations Screen.....	67
Figure 8.12: Mobile Application Chat Pick Media Screen	67
Figure 8.13: Mobile Application Chat from Trainee to Trainer	68
Figure 8.14: Mobile Application Notification from Chat	68
Figure 8.15: Mobile Application Chat from Trainer to Trainee	69
Figure 8.16: Mobile Application Announcements Screen	69
Figure 8.17: Mobile Application BMI Calculator Screen	70
Figure 8.18: Mobile Application BMR Calculator Screen	70
Figure 8.19: Mobile Application All Recipes Screen	71
Figure 8.20: Mobile Application Recipe Screen	71
Figure 8.21: : Mobile Application Admin Panel Screen	72
Figure 8.22: Mobile Application Trainees Screen	72
Figure 8.23: Mobile Application unsubscribed Users Screen	73
Figure 8.24: Mobile Application Scan QR Screen.....	73
Figure 8.25: Mobile Application Result of QR Scan Screen	74
Figure 8.26: Mobile Application Result of QR Scan Screen	74
Figure 8.27: Mobile Application Create Exercise Screen	75
Figure 8.28: Mobile Application Create Recipe Screen	75
Figure 8.29: Mobile Application Add Income Screen.....	76
Figure 8.30: Mobile Application Add Announcement Screen.....	76
Figure 9.1: VS Code	92
Figure 9.2: Dart Language	93
Figure 9.3: HTML	93
Figure 9.4: CSS Language	94
Figure 9.5: JavaScript Language	94
Figure 9.6: Flutter Framework	95
Figure 9.7: Node.js Framework	96
Figure 9.8: Firebase	97
Figure 9.9: MongoDB	98

Abbreviations and Definitions

- **GMS:** Gym Management System
- **Gym Admin:** Gym's owner or manager
- **User:** The one who is registered to the website/app and use them
- **Subscriber or Trainee:** The customer who has subscribed to the gym whether normal subscription or private subscription with a trainer.

Chapter 1: Introduction

1. Introduction

1.1 Purpose

Gym Management System aims to help the owners of the gym to manage their gym and record all financial operations and help them to know who are the subscribers in the gym and when is their expiration date and to make sure that only subscribed people enter the gym by using face recognition.

Also, to help the Customers to explore and know the advantages of the gym, to make things easier for them by providing an online way to subscribe.

To provide exercise videos as guide for them so they can use it to do their exercises whether at gym or home.

To provide a way for communication between GYM admin and trainees and trainers

To provide an appropriate diet for them by their trainers and help them to know their required calories based on their goal.

To help trainers to promote themselves by making profile for them so that trainers can explore profiles of trainers and choose the suitable one for them to do private subscription with.

1.2 Overview

This document is organized as follows:

First, an overview description of the GMS and its high-level functions. Next, stating types of users who can use GMS and a list of general constraints that should be followed. Then, provide a detailed description of the system functions and requirements. Finally, present some helpful information and diagrams that will facilitate the understanding of this document.

Chapter 2: Analysis and Requirements

2.1 System Users

2.1.1 System Stakeholders

- System Engineers
 - Responsible for gathering requirements.
 - Responsible for developing the system.
 - Responsible for deployment and support.
- Gym Administrator
 - The main stakeholder of this system.
 - responsible for adding trainers.
 - responsible for doing subscription for users.
 - responsible for adding exercises videos.
 - responsible for adding healthy meals for diet.
 - View the trainers and trainees and check their subscription.
 - Use the system to check whether a person is subscribed to the gym or not when entering the gym.
 - Add income to be saved on the systems.
 - Send an announcement to all trainers and users regarding maintenance of the gym, Etc.
- Trainee
 - The most frequent user beside the trainer.
 - Signup to the system to use it.
 - View his profile.
 - View exercises videos to help him to do his exercises.
 - Use body calculators.
 - View Meal recipes.
 - View the available subscription plans and do online subscription.
 - View the available trainers and their profile.
 - Subscribe with a trainer and chat with him.
 - Ability to send pictures and videos to the trainer to ask for help in performing exercises.
 - View Gym announcements.

- Trainers
 - View his profile.
 - Chat with his subscribed trainees and provide help to them.
 - View exercises videos.
 - Use body calculators.
 - View Meal recipes.
 - View the available subscription plans.
 - View the available trainers and their profile.
 - View Gym announcements.

2.1.2 Users Objectives

- System Engineer
 - Gain Experience in developing software.
- Gym Administrator
 - Organize his gym and make work easier.
 - Avoid the problem of lost or fake subscription cards.
 - Verifying all people who are in the gym.
- Trainee
 - No need to keep a subscription card with them.
 - Find a guide to help him to explore the gym system and do exercises.
 - Facilitate subscription and purchasing by making it available online.
 - Find appropriate meals for reaching their goal by the help of trainers.
- Trainer
 - Promote himself by having a profile.
 - Communicate with their subscribers.

2.2 User Requirements Definitions

2.2.1 System Functions

1. Sign up and login
2. View user profile
3. View trainees and trainers
4. Scan User QR code
5. Add exercises
6. Add meals recipes
7. Add an income
8. Add an announcement
9. View exercises videos
10. View subscription plans
11. Online subscription
12. Private subscription with a trainer
13. Chat between trainee and trainer
14. Use Body Calculators (BMI & BMR)
15. View meals recipes
16. View available products for sale
17. Buy products online using credit card
18. View Announcements

2.2.2 Constraints

- Gym Policies
 - Each trainee/trainer has one account only
 - Any trainee can't enter gym without scanning face
 - All incomes must be entered to the system
- Technology limitations
 - No Technology Limitations
- Hardware Limitations
 - Computer (required)
 - Camera (required)
 - Internet (required)
 - Android Mobile (required)

2.3 System Requirements

Requirements can be divided into functional requirements and non-functional requirements.

2.3.1 Functional Requirements

Sign Up

Users can Sign up to the system using Website or Mobile Application.

Login

Users can Login to the system on the Website or Mobile Application

View User Profile

User can View his profile and information, use the generated QR code in his profile for authentication when entering the gym.

View Trainees

Only gym administrator can view all trainees that are registered on the system and their information.

View Trainers

Gym administrator and users can view the available trainers in the gym and information about them by viewing their profile.

Scan User QR code

Only gym administrator can scan the QR code of people when they are entering the gym to check whether they are subscriber or not and check their expiration date.

Add an Income

Only gym administrator is the one who can add an income using admin panel.

Add an Announcement

Only gym administrator can add an announcement in case he wanted to announce about something important like maintenance of the gym.

View exercises videos

Users can View exercises Videos and explanations steps to help them to perform it.

Videos are categorized by Body part.

View Subscription Plans

Users can view available subscription plans and choose the suitable one for them.

Online Subscription

User can use his credit card to subscribe to the plan he wants online.

Private subscription with a trainer

Users can make private subscription with a trainer to train them and chat with him to ask him for help

Chat between trainer and trainee

Trainer and trainee can chat each other by sending text messages and sending videos/images to ask for help and help them to have better way to illustrate their inquiry.

Use body calculators (BMI & BMR)

Users can use body calculators to check their body fatness and decide if they need to lose or maintain or gain weight.

View Meal recipes

Users can view healthy meal recipes and how to prepare them and the nutritional value of each meal.

By the help of trainers and body calculators they can make healthy diet to achieve their goal.

View Available Products for sale

Users can view the available products for sale in the gym and their prices.

Buy Products Online

Users can use their credit card to buy the available products like Protein powders and other dietary supplements, Sportswear, etc.

View Announcements

Users can view the announcements to check if the gym is closed today due to maintenance or if there are new changes in the gym system.

2.3.2 Non-Functional Requirements

Non-functional requirements can be categorized as availability, security, maintainability, usability, and flexibility.

Availability

- Ensures that the system will work on all browsers.
- Ensures that the mobile application will work on all mobiles with different screen sizes as long as it meets the requirements (like android version)
- System mobile application and web application will run all the time as all users have mobiles and laptops these days and want to use the system at any time.

Security

- The system will prevent unsubscribed people from entering the gym by scanning their face and checking whether they are allowed to enter or not.

Maintainability

- The maintenance requirements of the system are concerned with maintenance of the issues reported by users.

Usability

- The system should be fast and responsive.
- The system should be easy to use and have user-friendly UI/UX.

Flexibility

- User can log on to the system at any time as long as he has internet access.
- The system is flexible and can accommodate increasing number of users.

2.4 Functional Requirements Specifications

2.4.1 Use Cases Description

• Registration

Actor: User

Brief Description:

This use case describes how the user register to the system and submits his required information.

Basic Flow:

This use case starts when the user wants to register to the system using a web application or mobile application.

1. System requests from the user to enter his email and password and upload an image of the user.
2. Actor enters his email and password and uploads an image of himself.
3. Actor will press Register Button.
4. System check the entered email and password, if they meet the requirements, user will be registered successfully.

Alternative Flow:

[Username/password doesn't meet requirements.](#)

If a user entered an email or password that doesn't meet the requirements, the system will display an error message to tell the user what is wrong so that he modifies it to meet the requirements.

[Password and Password confirm doesn't match](#)

If a user entered a valid email but password and password confirm doesn't match, system will display an error message telling the user that password and password confirm doesn't match.

- **Login**

Actor: User/Trainer/Trainee/Gym Administrator

Brief Description:

This use case describes how the user/trainer/trainee/gym administrator login to the system.

Basic Flow:

This use case starts when an actor wants to login to the system.

1. System asks the user to enter his valid email and password.
2. User enters his valid email and password.
3. System validates the entered email and password and logs the actor into the system.

Alternative Flow:

Actor enters invalid email or password

If the actor in the basic flow entered invalid email and/or password, the system will show up an error message telling the user that the email/ password is incorrect, and he should check them.

Use case back to the beginning of the basic flow where it asks the user to enter a valid email and password.

- **Log Out**

Actor: User/Trainer/Trainee/Gym Administrator

Brief Description:

This use case describes how the user/trainer/trainee/gym administrator log out of the system.

Basic Flow:

This use case starts when an actor wants to log out of the system.

1. Actor will press on Logout button.
2. Actor will be logged out of the system and system will back to login use case.

- **Admin scans face for verification**

Actor: Admin

Brief Description:

This use case describes how to verify people using face recognition model

Admin uses his mobile with the mobile application to scan the QR code in the profile of the user before entering the gym to check if they are allowed to enter or not.

Basic Flow:

1. Gym administrator scans face of the person
2. The result on screen will be “valid” user so that person can enter the gym

Alternative flow:

1. Gym administrator scans face of the person
2. The result on screen will be “invalid” which means he is a member of gym, but his subscription is expired, or it will be “unknown” which means he isn’t a member of the gym. Both cases the person will not enter the gym.

- **Users subscribe to the gym**

Actor: User

Brief Description:

This use case describes how users subscribe to the gym

Basic Flow:

1. User logins to the website
2. User go to offers page from home page
3. User choose one of the available offers
4. User is redirected to payment page
5. User enters payment data
6. Payment data is sent to stripe API to check it
7. Payment is accepted, and user is redirected to his profile page and display his offer

Alternative Flow:

1. User logins to the website
2. User go to offers page from home page
3. User choose one of the available offers
4. User is redirected to payment page
5. User enters payment information
6. Payment information is sent to stripe API to check it
7. Payment is refused (invalid) and user is redirected to the payment page again to re-enter payment data

- **User subscribe with a trainer (private subscription)**

Actor: User

Brief Description:

This use case describes how users subscribe private with a trainer

Basic Flow:

1. User logs in to the website
2. User goes to trainers' page from home page
3. User selects a trainer and visits his profile
4. User views available offers for this trainer
5. User is redirected to payment page
6. User enters payment data
7. Payment data is sent to stripe API to check it
8. Payment is accepted, and user is redirected to his profile page and displays his offer

Alternative Flow:

1. User logs in to the website
2. User goes to trainers' page from home page
3. User selects a trainer and visits his profile
4. User views available offers for this trainer
5. User is redirected to payment page
6. User enters payment data
7. Payment data is sent to stripe API to check it
8. Payment is refused (invalid), and user is redirected to his profile page and displays his offer

- **Admin scan QR code for authentication**

Actor: Gym administrator

Brief Description:

In case of face recognition system has a problem or the camera isn't working, Admin uses his mobile with the mobile application to scan the QR code in the profile of the user before entering the gym to check if they are allowed to enter or not.

Basic Flow:

1. Gym administrator asks the person who is registered in the system to show up his QR code in his profile in the application.
2. Gym administrator uses his phone to scan the QR code and wait for the result.
3. Mobile application will show up a message saying that this user is found in the system and allowed to enter.

Alternative flow:

1. Gym administrator asks the person who isn't registered in the system to show up his QR code in his profile in the application.
2. Gym administrator uses his phone to scan the QR code and wait for the result.
3. Mobile application will show up a message saying that this user isn't allowed to enter the gym because he doesn't exist on the system or because his subscription has expired.

- **User uses BMI Calculator**

Actor: User/trainer/trainee

Brief Description:

The actor will enter the required information in BMI calculator screen and wait for the result.

Basic Flow:

1. Actor will enter his height and weight using slider bar in BMI calculator screen.
2. Actor will press Calculate button.
2. Actor will get a result displaying his BMI and his body fatness class.

- **User uses BMR Calculator**

Actor: User/trainer/trainee

Brief Description:

The actor will enter the required information in BMR calculator screen and wait for the result.

Basic Flow:

1. Actor will choose his goal (gain/loss) weight.
2. Actor will choose his gender (male/female).
3. Actor will enter his height in cm.
4. Actor will enter his weight in Kg.
5. Actor will enter his age.
6. Actor will choose his activity factor (choose one of 4 available factors each has its description).
7. Actor will press Calculate button.
7. Actor will get a result displaying the needed calories to (maintain/gain/loss) weight

- **Chat**

Actor: Trainer/Trainee

Brief Description:

Trainer and Trainee can chat with each other and send text messages and send images and videos.

Basic Flow:

1. The sender actor will type a text message in the message text field and press the send button or send an image or video from gallery/camera.
2. The recipient actor will receive the message from the sender and receive a notification with a vibration.

- **Create Exercise**

Actor: Gym administrator

Brief Description:

Gym administrator upload exercise's video with a description to the server database.

Basic Flow:

1. Gym administrator Enters the name of the exercise.
2. Gym administrator chooses to which type this exercise belongs.
3. Gym administrator adds the steps of the exercise (explanation of exercise).
4. Gym administrator picks a video for exercise to upload to server-side database.
5. Gym administrator press Add button to upload the exercise.

- **Create Recipe**

Actor: Gym administrator

Brief Description:

Gym administrator upload recipe image with ingredients and nutrition value to the server database.

Basic Flow:

1. Gym administrator Enters the name of the meal recipe.
2. Gym administrator adds the ingredients of the meal recipe.
3. Gym administrator adds nutrition meal recipe (Calories & Protein & Fats & Carbs)
4. Gym administrator picks an image for meal recipe to upload to server-side database.
5. Gym administrator press Add button to upload the meal recipe.

- **Add Income**

Actor: Gym administrator

Brief Description:

Gym administrator adds income to be saved on the server-side database.

Basic Flow:

1. Gym administrator Enters the income title.
2. Gym administrator Enters the amount of income (Price).
3. Gym administrator press Add button to upload the Income.

- **Add Announcement**

Actor: Gym administrator

Brief Description:

Gym administrator adds an announcement to all users that use the mobile application.

Basic Flow:

1. Gym administrator Enters the announcement.
2. Gym administrator press Add button send the announcement to all users.

Chapter 3: Software Design

3.1 Use case Diagram

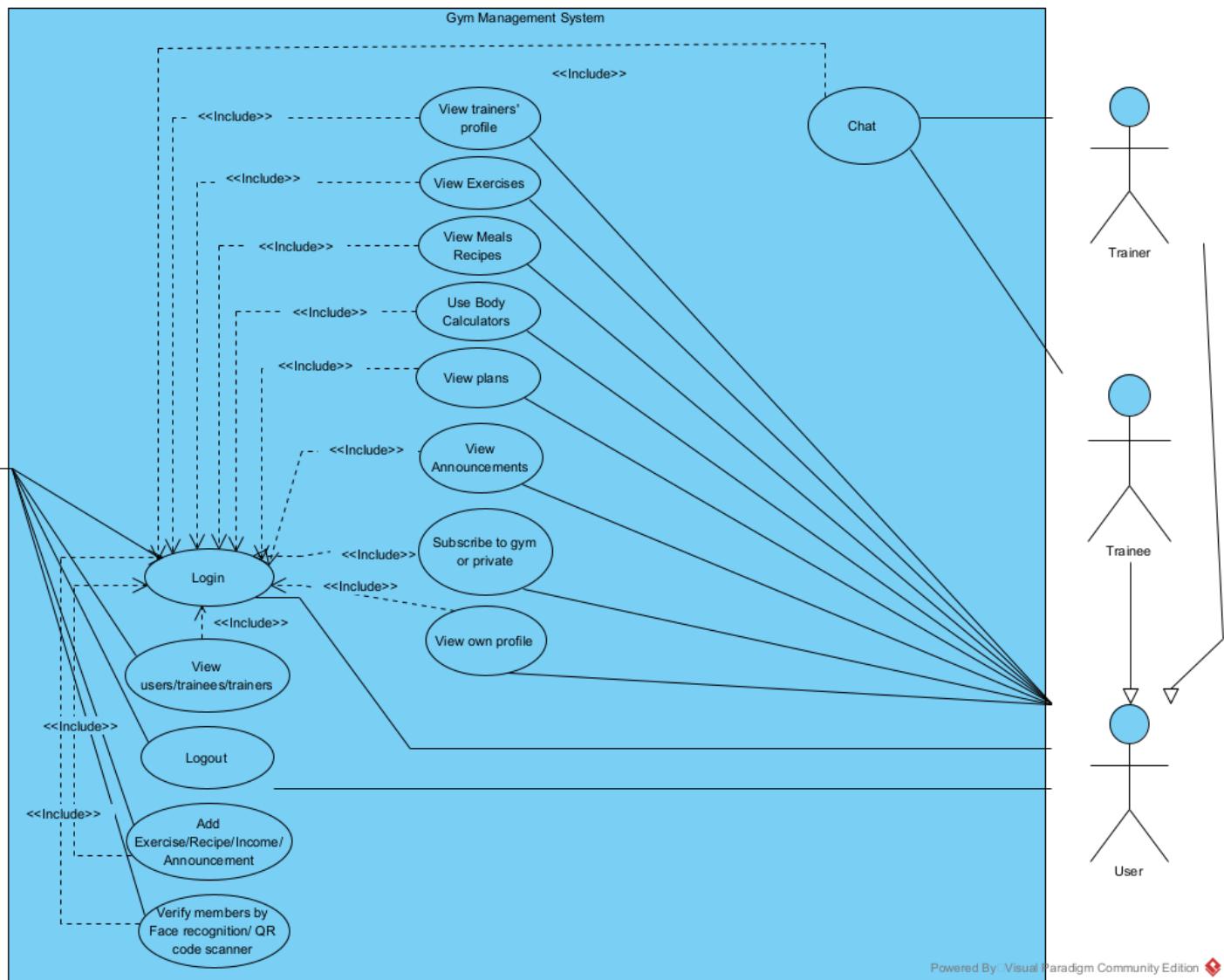


Figure 1: Use case diagram

3.2 Sequence Diagram

- User Registration

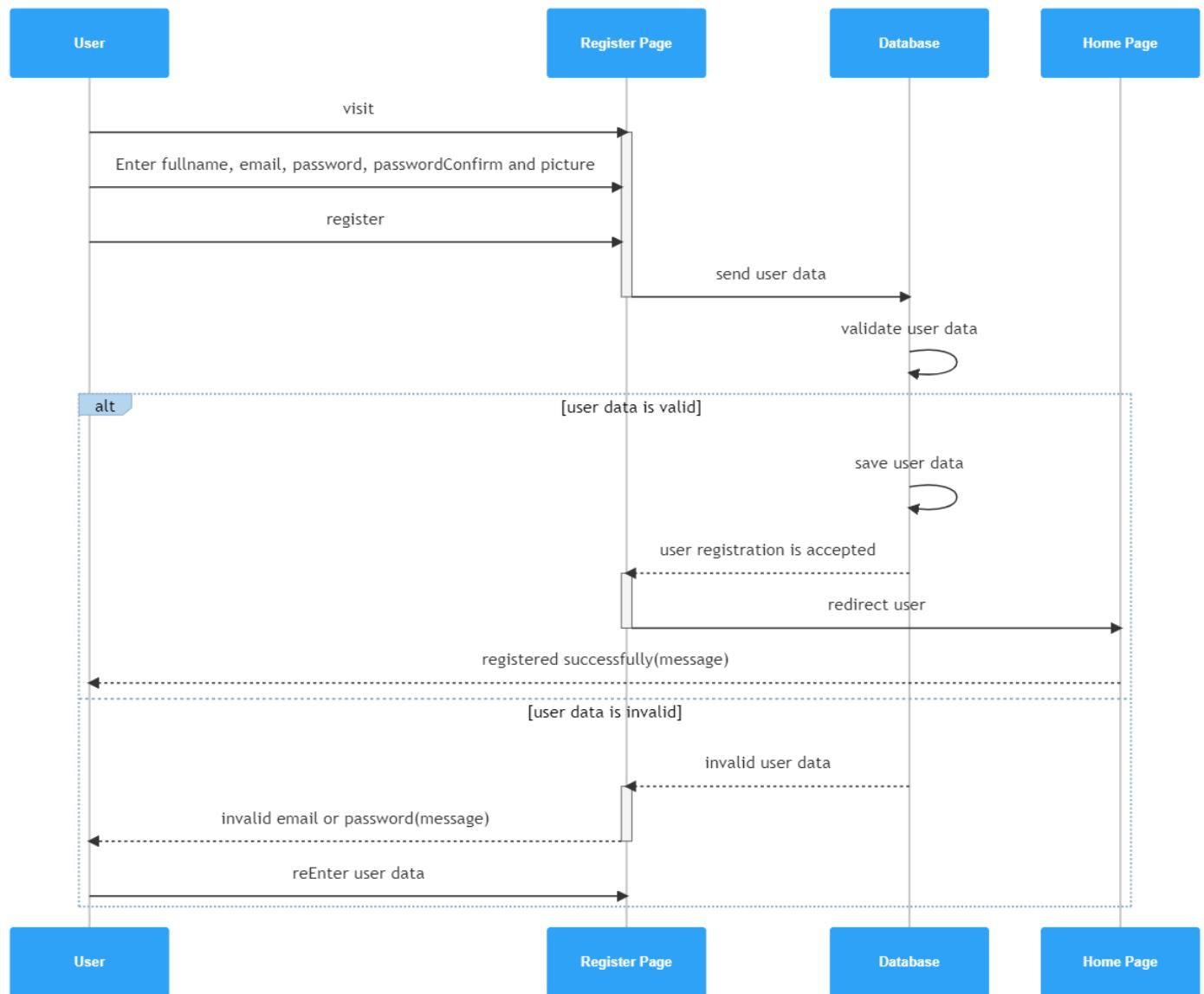


Figure 2.1: Registration sequence diagram

- User Login

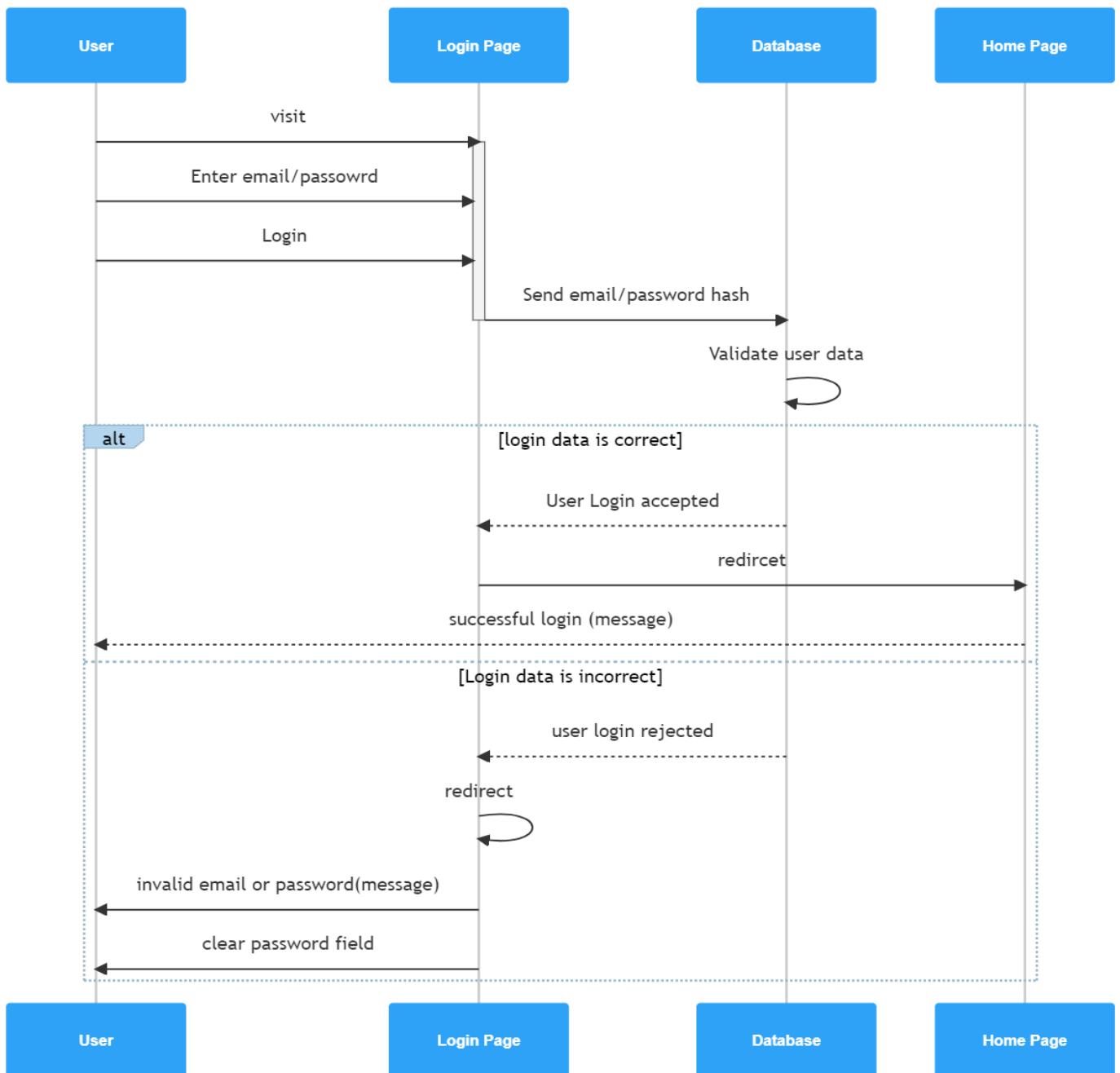


Figure 2.2: Login sequence diagram

- Gym Subscription

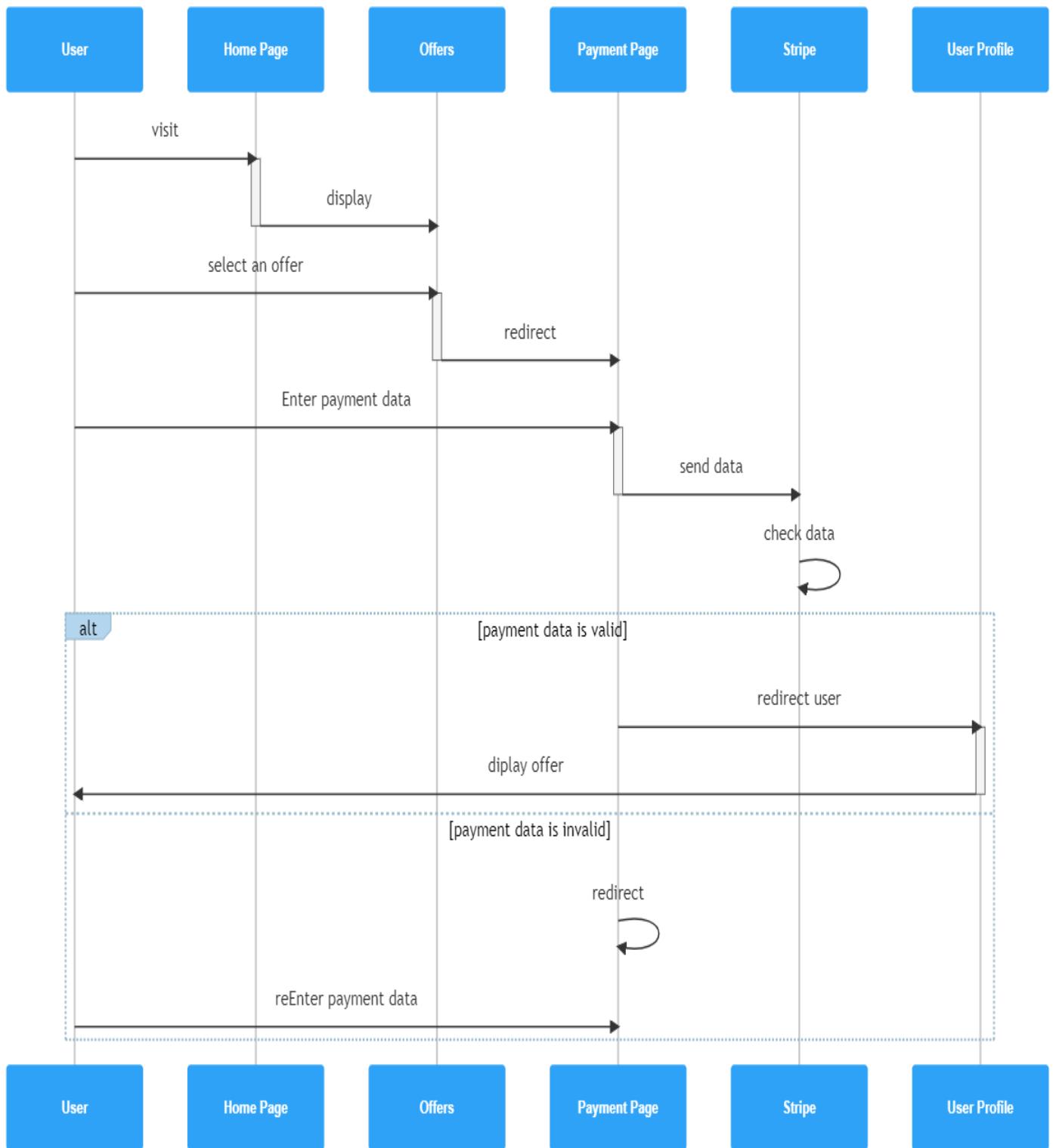


Figure 2.3: Subscription sequence diagram

- Trainer Subscription

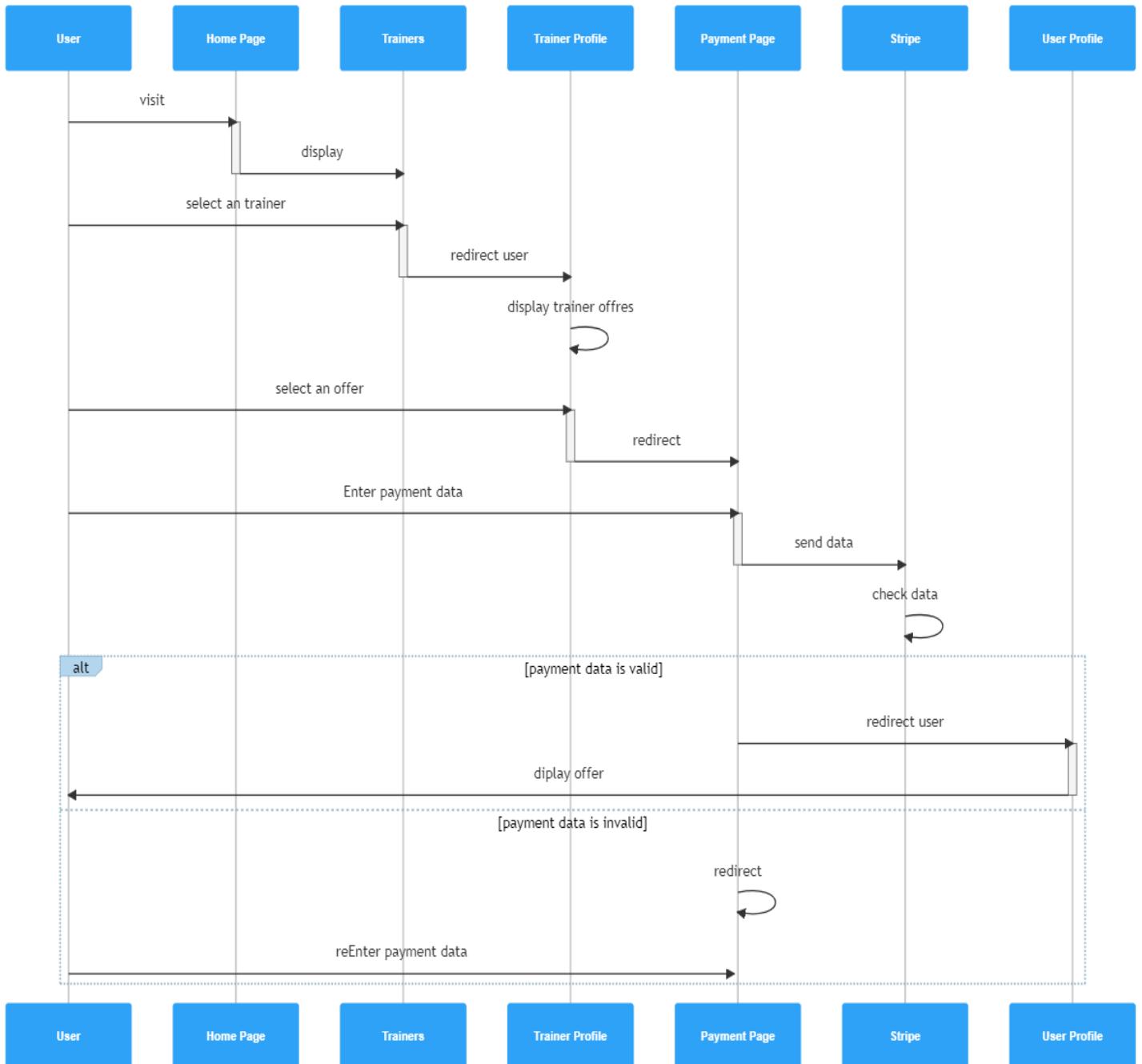


Figure 2.4: Private Subscription sequence diagram

- Face scan

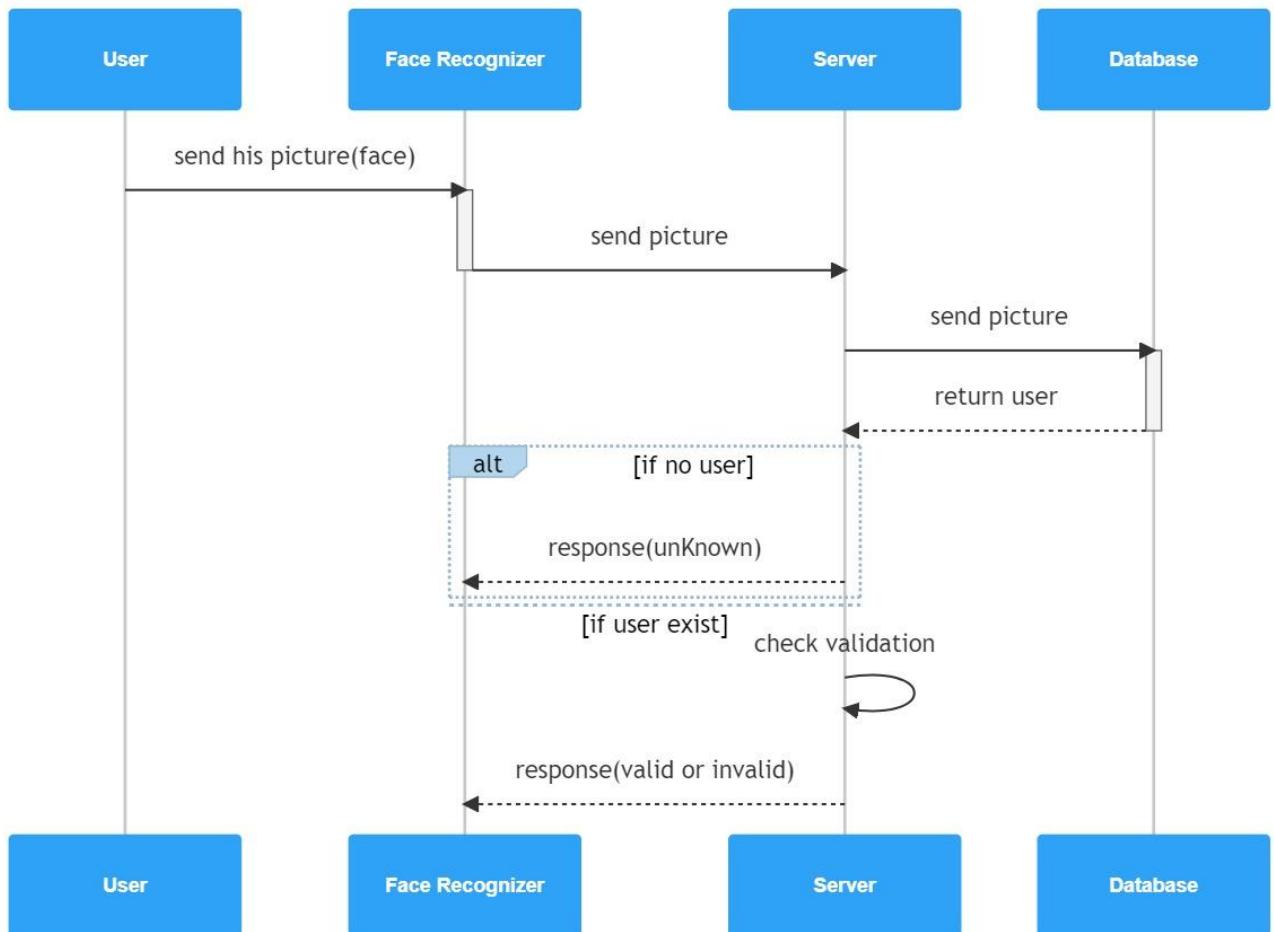


Figure 2.5: Face Scan sequence diagram

- Chat

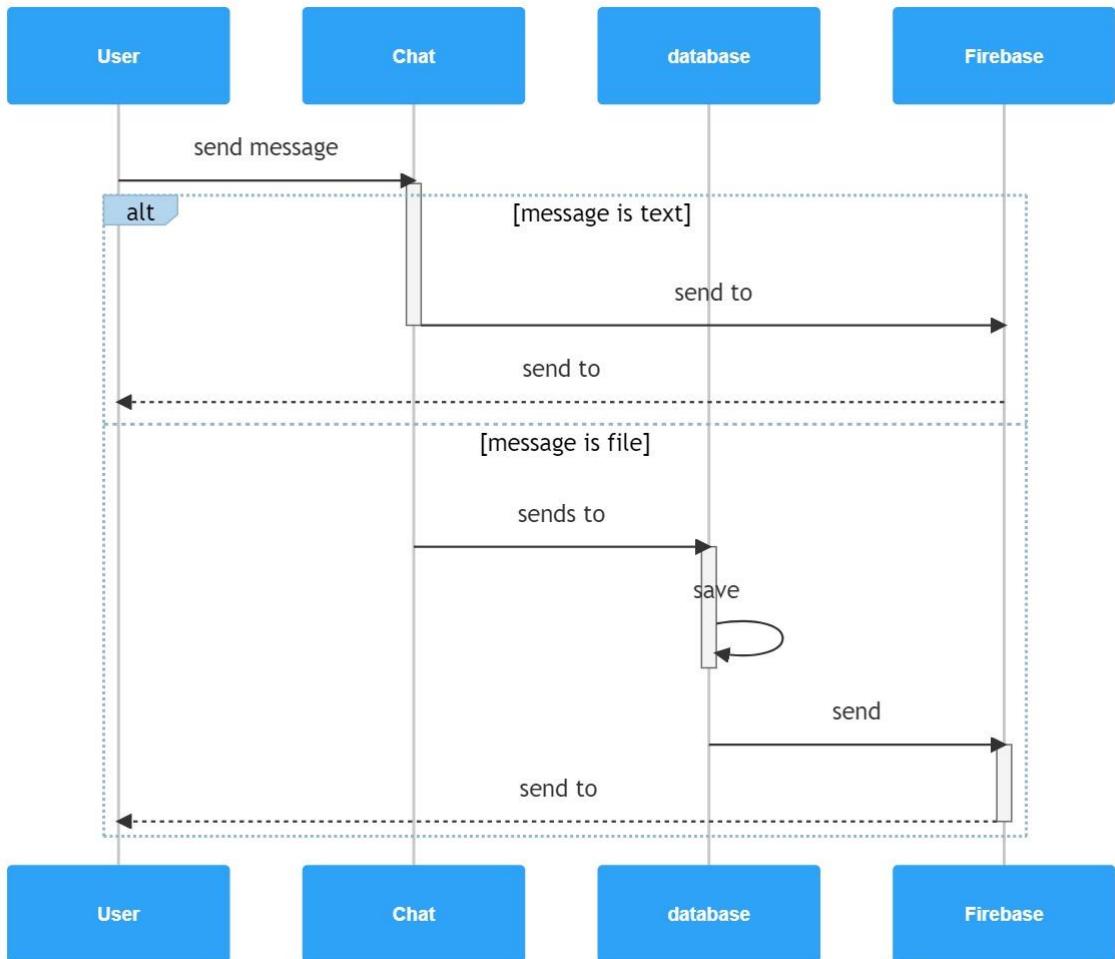


Figure 2.6: Chat sequence diagram

- Adding Announcement

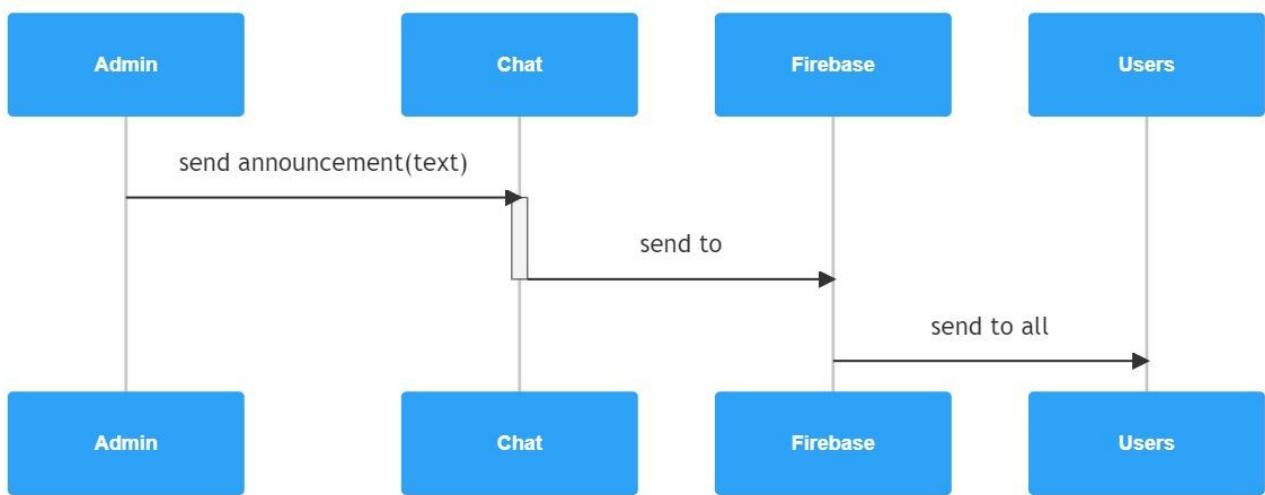


Figure 2.7: Add Announcement sequence diagram

- Adding Income/Exercise/Recipe

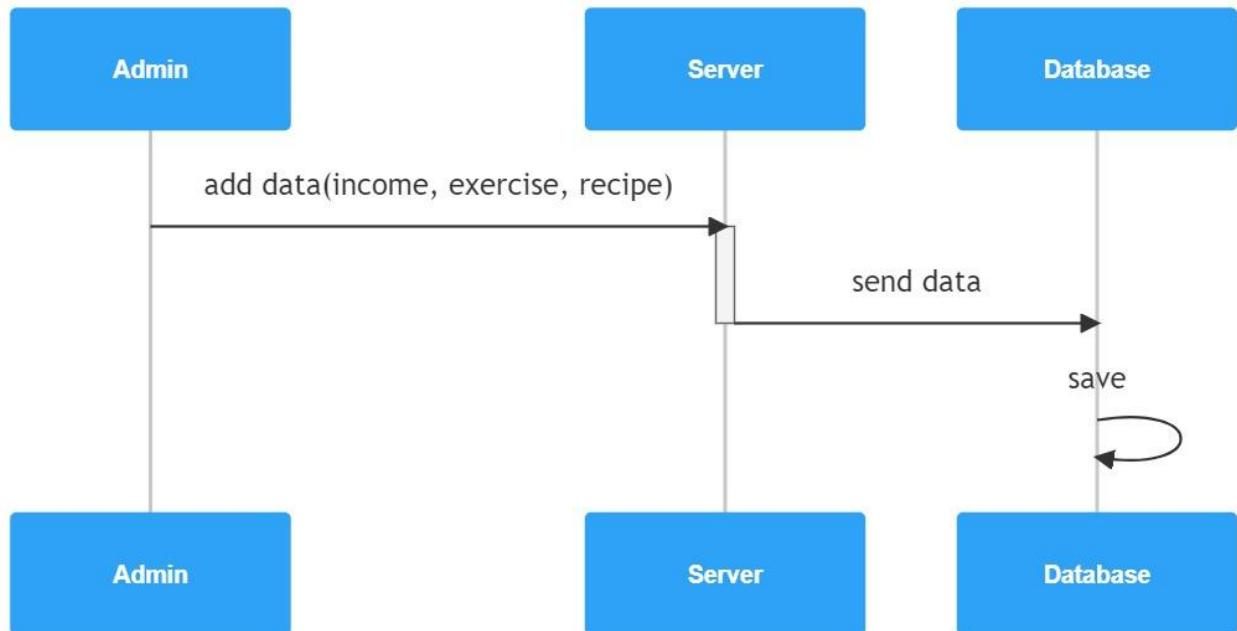


Figure 2.8: Add Income/Exercise/Recipe sequence diagram

3.3 Entity relationship diagram (ERD)

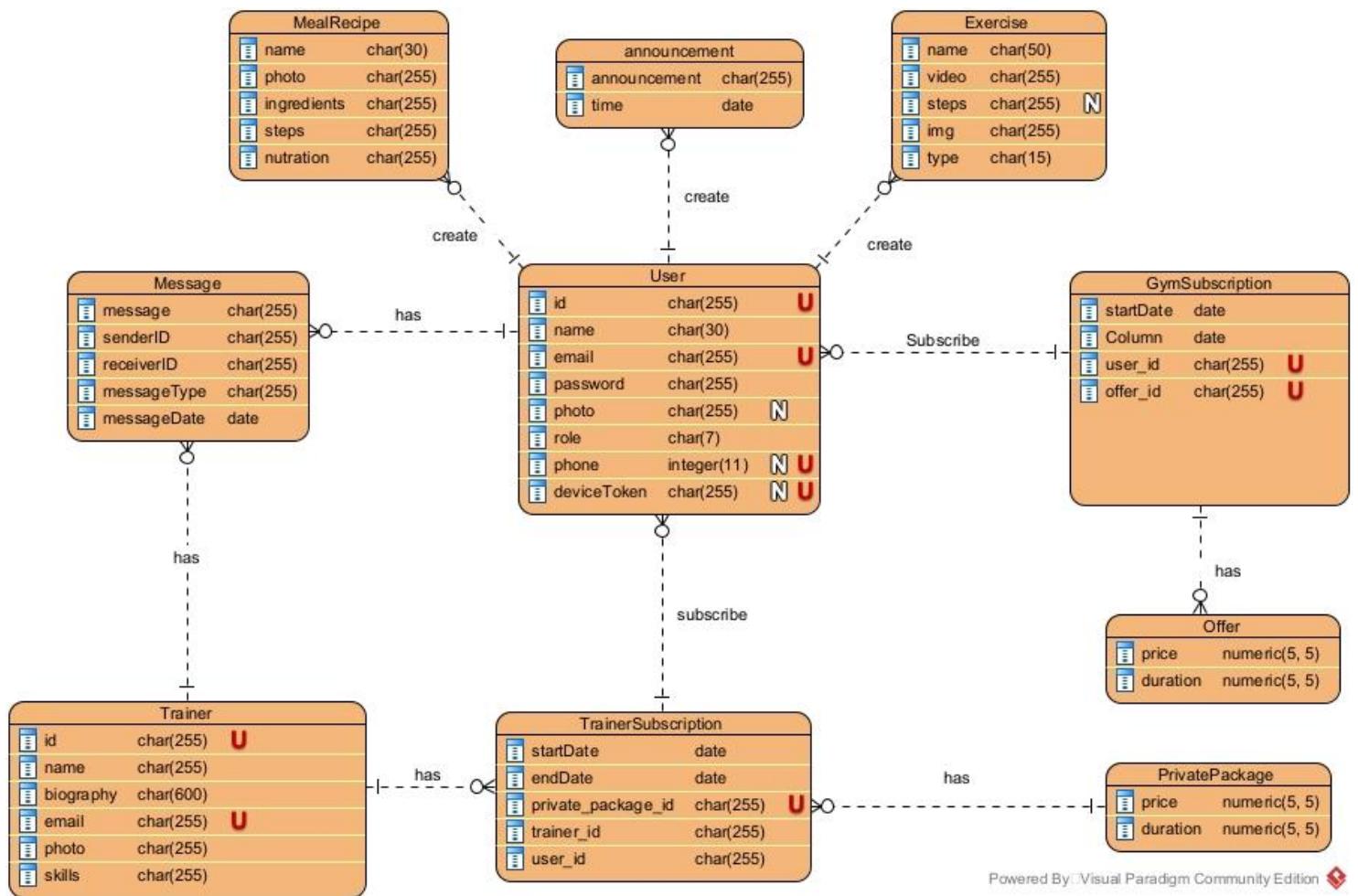


Figure 3: Entity Relationship diagram (ERD)

3.4 Class Diagram

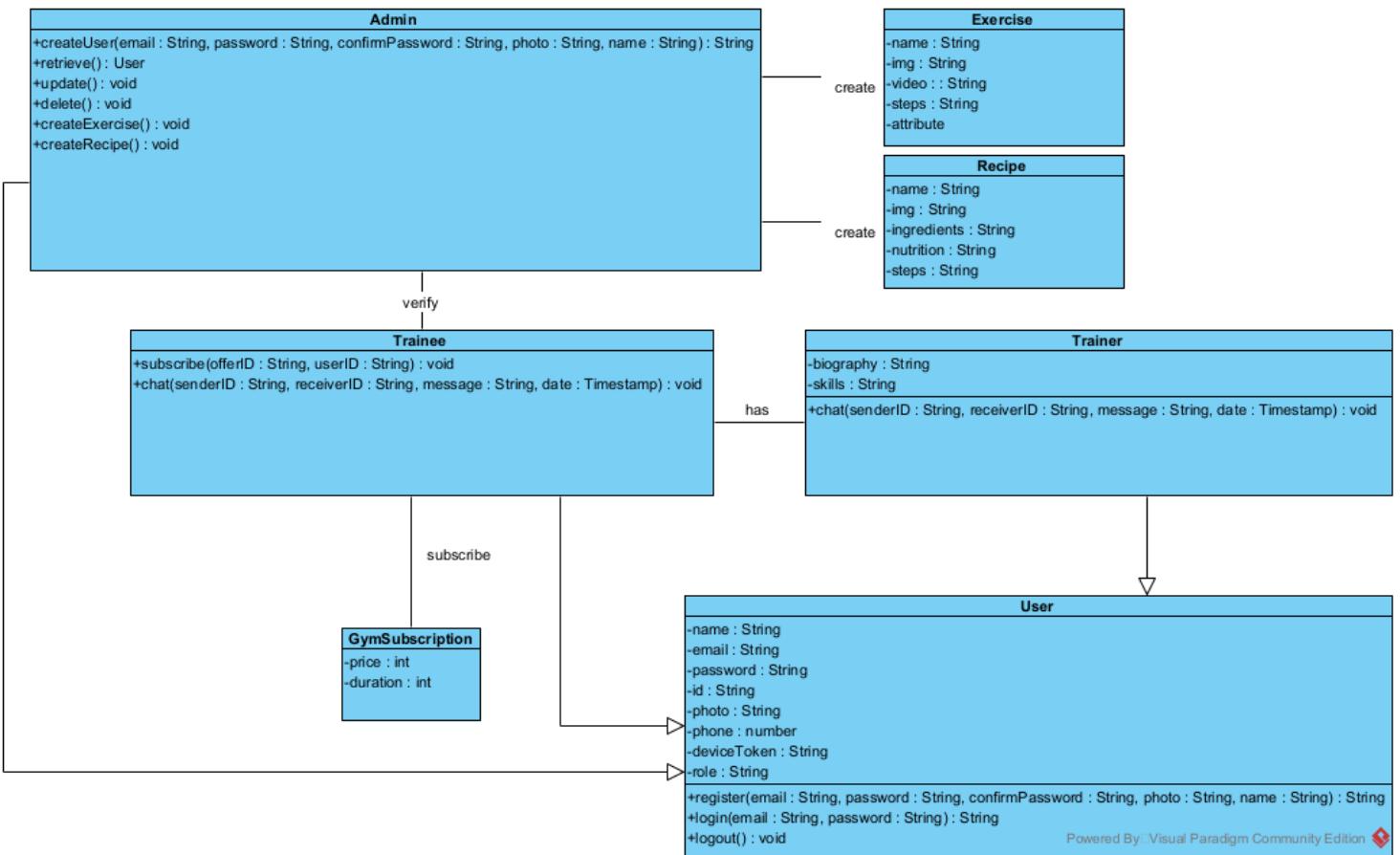
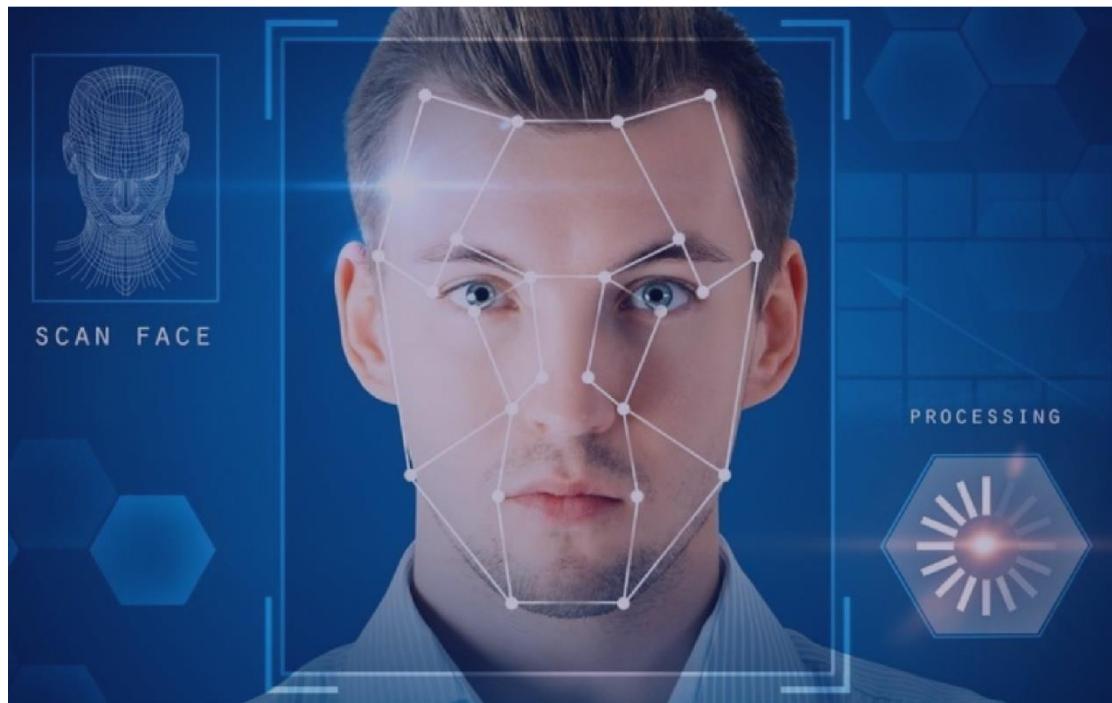


Figure 4: Class diagram

Chapter 4: Face Recognition

4. Face Recognition Model



4.1 Introduction

This face recognition model is designed to verify the identity of clients in a gym by analyzing their facial features.

The model takes an image of a client's face as input, compares it to a database of known faces, and outputs a response indicating whether the client is valid, invalid, or unknown.

This model can be used to automate the client verification process in the gym and save time for both the clients and the gym staff.

4.2 Model Architecture

The face recognition model is built using a deep learning architecture called Convolutional Neural Network (CNN).

CNN consists of multiple layers that are trained on a dataset of faces to learn the features that distinguish one face from another.

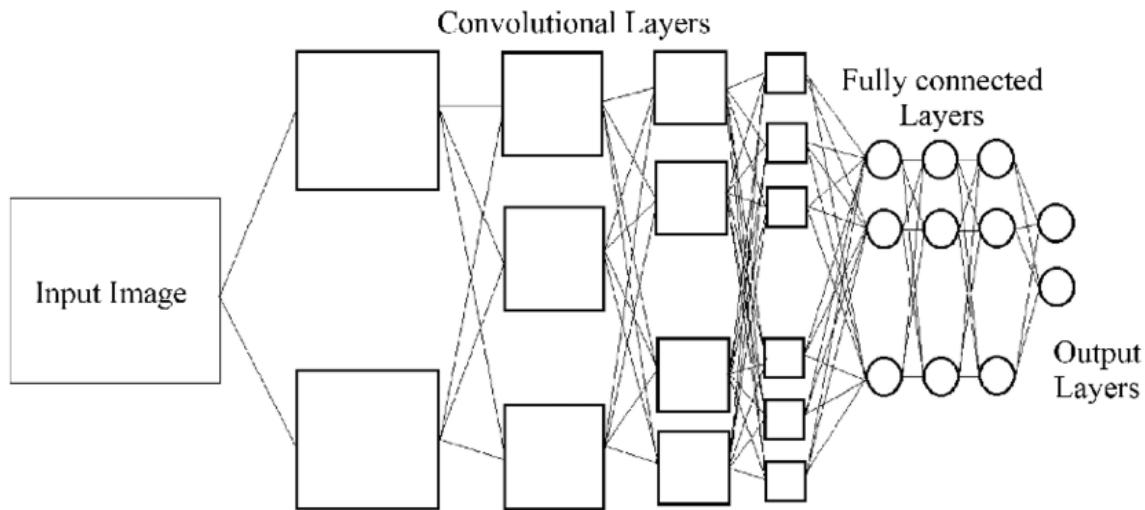


Figure 5: Convolutional Neural Network (CNN)

The model is then able to identify the unique features of each face and use them to recognize the individual.

The CNN architecture used in this model consists of several convolutional layers followed by pooling layers, and then fully connected layers.

The convolutional layers extract features from the input image, while the pooling layers reduce the size of the feature maps.

The fully connected layers combine the features and make the final prediction.

The model is trained using the backpropagation algorithm to minimize the difference between the predicted output and the actual label.

4.3 Model Inputs

The model requires an image of the client's face as input. The image can be captured using a camera or uploaded from a file.

The image should be clear and well-lit, with the face centered and facing forward.

The model can handle images of different sizes and orientations, but it is recommended to use images of similar size and orientation for better accuracy.

The model inputs may also include additional information about the client, such as their name, ID number, or subscription end date.

This information can be used to verify the identity of the client and determine their **subscription status**.

4.4 Model outputs

The model outputs a response indicating whether the client is **valid**, **invalid**, or **unknown**.

The response is sent to a server and displayed on the screen.

The response is **valid** if the client subscription end date is not come yet, **invalid** if the client subscription has expired, and **unknown** if the client is not in the database.

4.5 Model workflow

The face recognition model works by comparing the **input image** to a **database of known faces**.

The database contains images of all the clients in the gym, along with their subscription end dates.

When a new image is sent to the model, it first preprocesses the image to extract the **facial features**.

The model then compares these features to the database to find a match.

If a match is found, the model checks the subscription end date of the client in the database.

If the subscription end date has not expired, the response is **valid**.

If the subscription end date has expired, the response is **invalid**.

If **no match** is found in the database, the response is **unknown**.

In this case, the gym staff can manually verify the identity of the client and add their image and subscription end date to the database for future reference.

4.6 Model training

The face recognition model is trained on a dataset of labeled faces using supervised learning.

The dataset contains images of individuals labeled with their **names in data base**.

The model is trained to learn the unique features of each face and to recognize the individual based on those features.

The training process involves adjusting the weights of the CNN layers to minimize the difference between the predicted output and the actual label.

The dataset used to train the model should be diverse and representative of the population of clients in the gym.

It should also include images of clients with different orientations, hairstyles, and lighting conditions to improve the robustness of the model.

4.7 Model optimization

The face recognition model can be optimized in several ways to improve its accuracy and efficiency.

One way to optimize the model is to use transfer learning, which involves using a pre-trained CNN to extract features from the input image.

The pre-trained CNN has already learned to recognize low-level features such as edges and corners and can be fine-tuned on the specific task of face recognition.

Another way to optimize the model is to use ensemble learning, which involves combining multiple models to improve the overall performance.

The ensemble can include different types of CNN architectures, or different versions of the same architecture trained on different subsets of the data.

4.8 Model deployment

The face recognition model can be deployed in various ways depending on the application requirements.

One way to deploy the model is to use it as a web API, which can be accessed by other applications through RESTful endpoints.

This allows the model to be used in various platforms and devices, such as web browsers or mobile apps.

4.9 Testing cases:

- Case 1:

The client is not a gym member.



Figure 6.1: Case 1 Input



Figure 6.2: Case 1 Output

- **Case 2:**

The client is a gym member, but his subscription is expired.



Figure 6.3: Case 2 Input

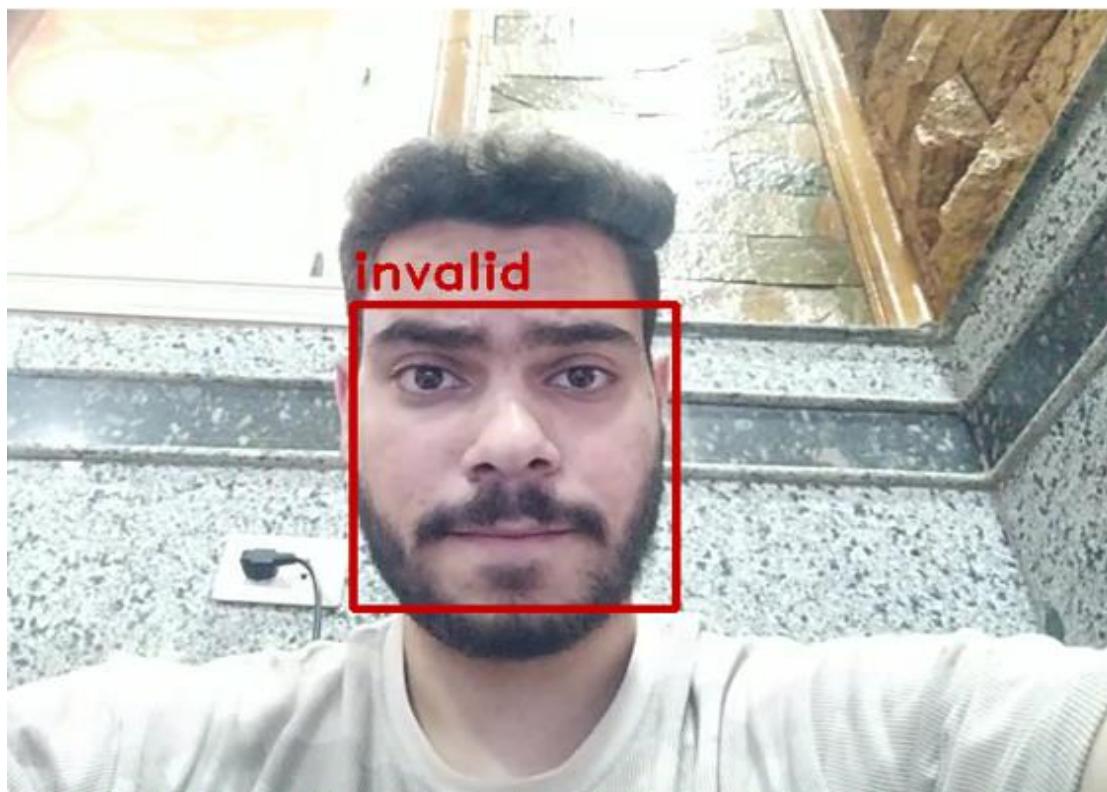


Figure 6.4: Case 2 Output

- **Case 3:**

The client is a gym member, and his subscription is valid (not expired).



Figure 6.5: Case 3 Input

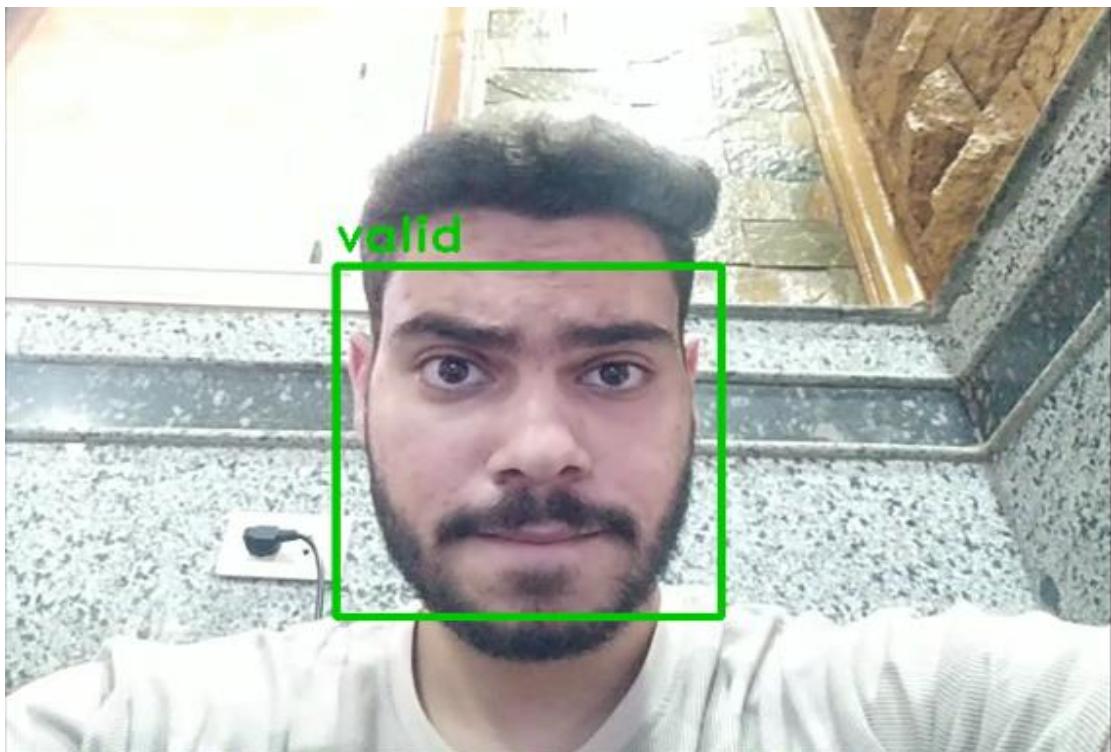


Figure 6.6: Case 3 Output

4.10 Conclusion

This face recognition model provides a reliable and efficient way to verify the identity of clients in a gym. By comparing the facial features of each client to a database of known faces, the model is able to accurately recognize the individual and determine their subscription status.

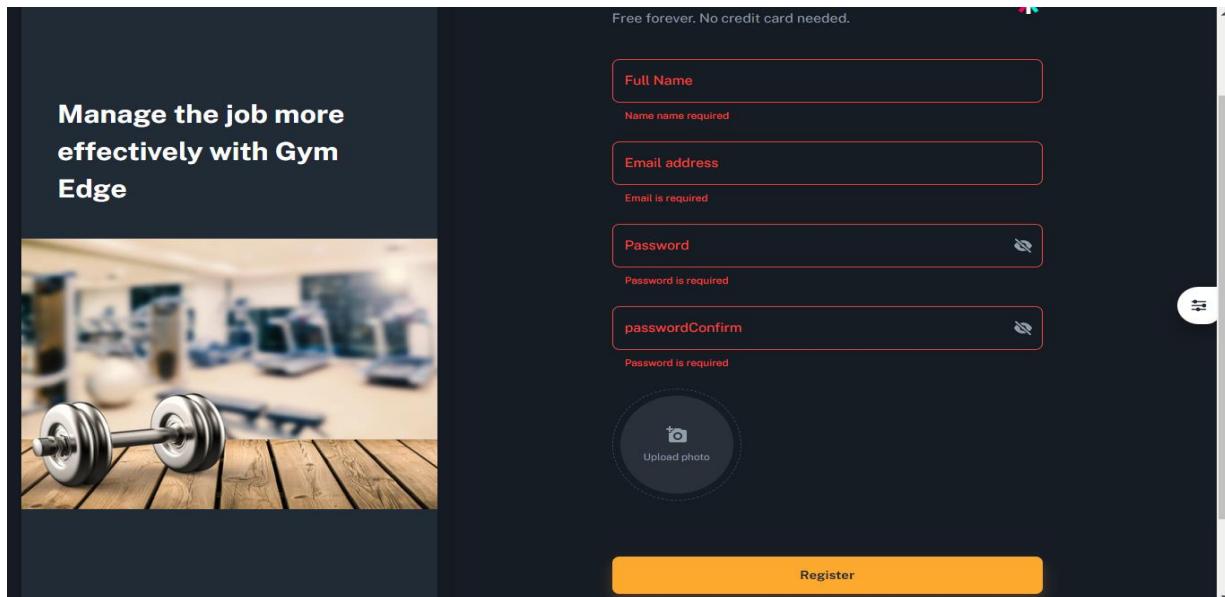
This model can help to automate the client verification process in the gym and save time for both the clients and the gym staff. With proper training and optimization, this model can achieve high accuracy and robustness in real-world scenarios and can be deployed in various platforms and devices.

Chapter 5: System Implementation

5. System Implementation

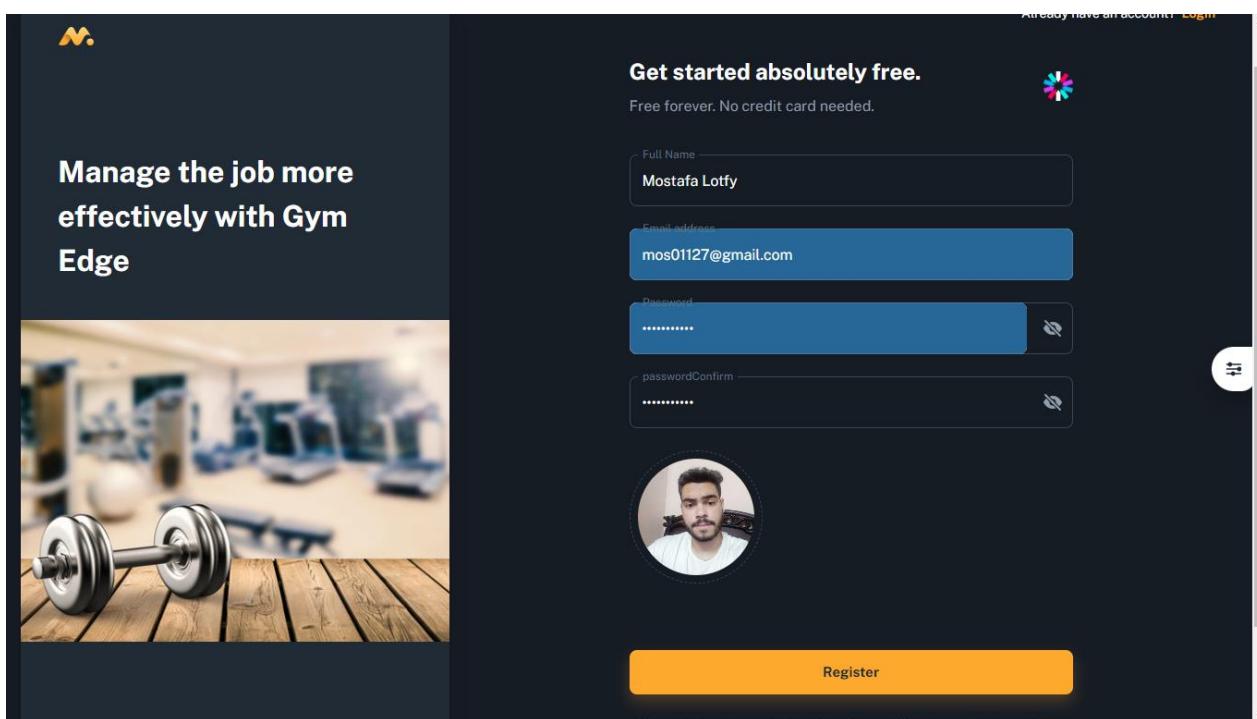
5.1 Website

- Registration Page



The screenshot shows the registration page for the Gym Edge website. The left side features a dark header with the text "Manage the job more effectively with Gym Edge" and a blurred background image of gym equipment. The right side has a dark background with white text and input fields. At the top right, it says "Free forever. No credit card needed." Below this are four input fields: "Full Name" (with error message "Name name required"), "Email address" (with error message "Email is required"), "Password" (with error message "Password is required"), and "passwordConfirm" (with error message "Password is required"). There is also a circular "Upload photo" button with a camera icon. At the bottom is a large orange "Register" button.

Figure 7.1: Website Empty Registration Page



The screenshot shows the same registration page but with filled input fields. The "Full Name" field contains "Mostafa Lotfy". The "Email address" field contains "mos01127@gmail.com". The "Password" and "passwordConfirm" fields both contain masked text. The "Upload photo" button now displays a circular profile picture of a man with a beard. The rest of the page, including the header text and the orange "Register" button, remains the same as in Figure 7.1.

Figure 7.2: Website Filled Registration Page

- Login Page

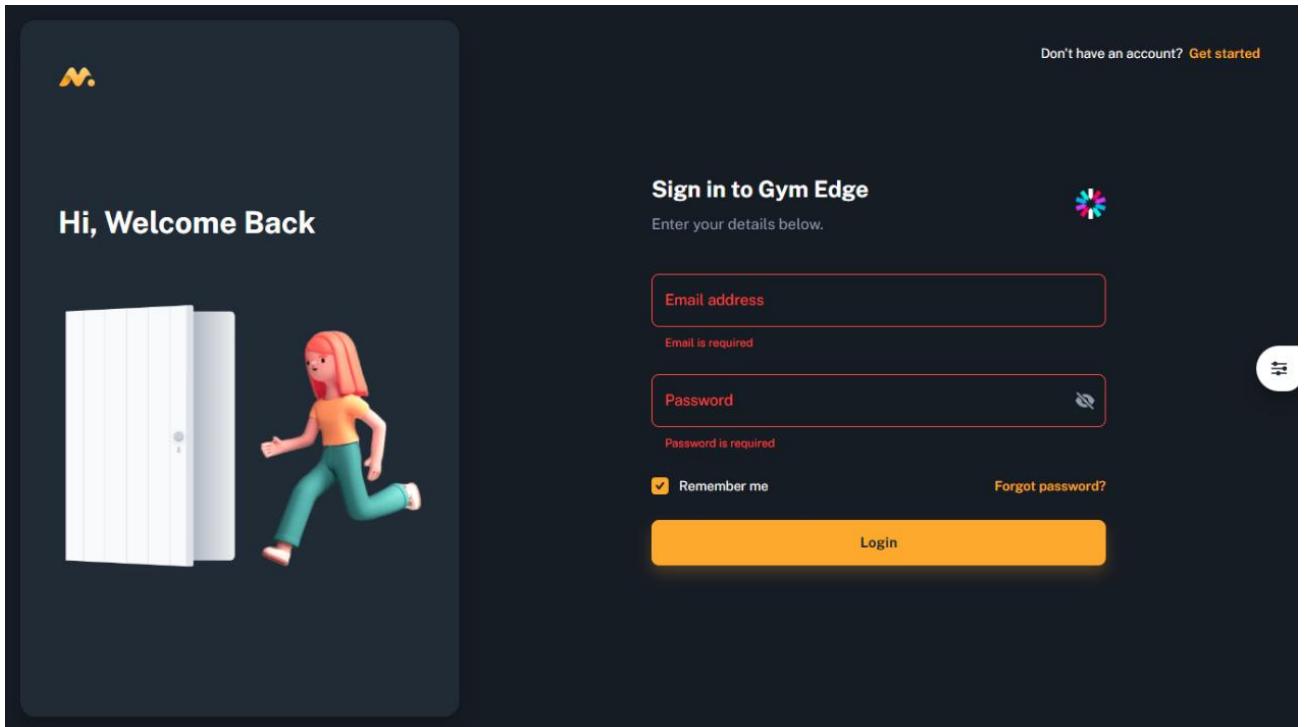


Figure 7.3: Website Empty Login Page

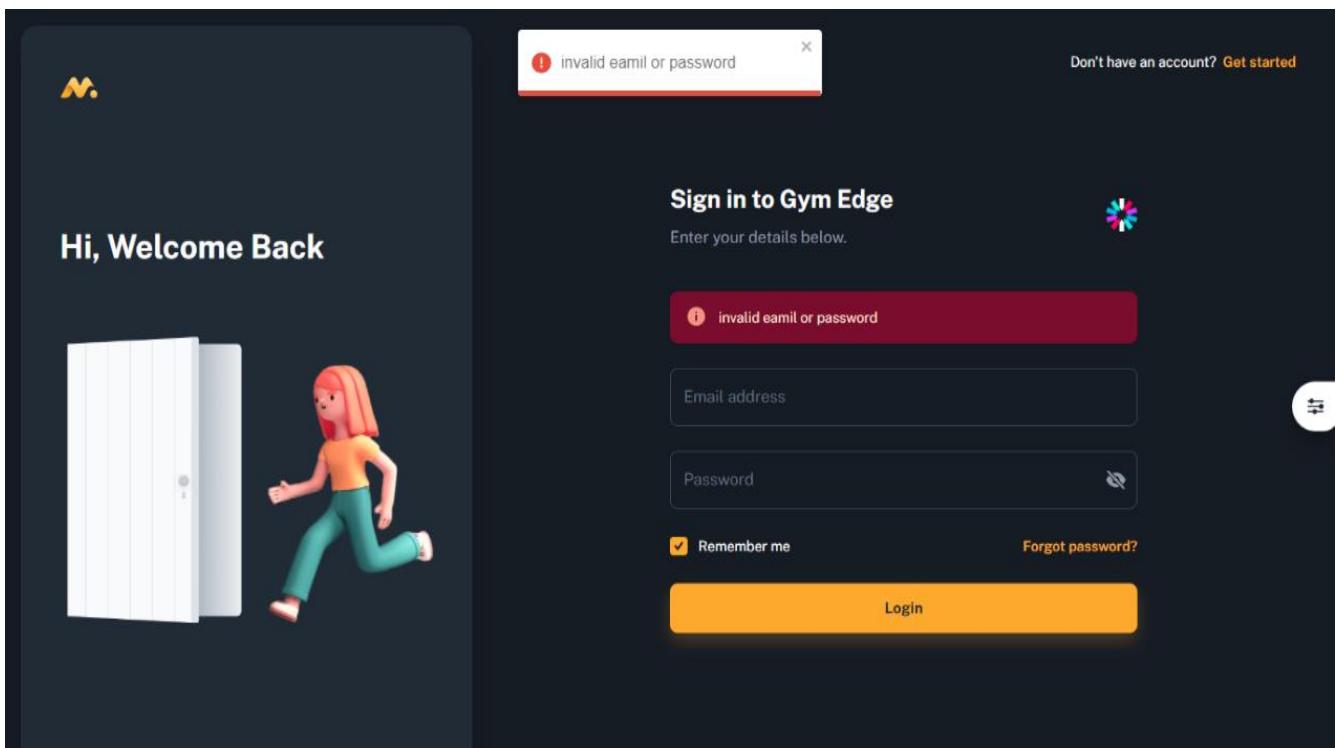


Figure 7.4: Website Filled Login Page

- Home Page

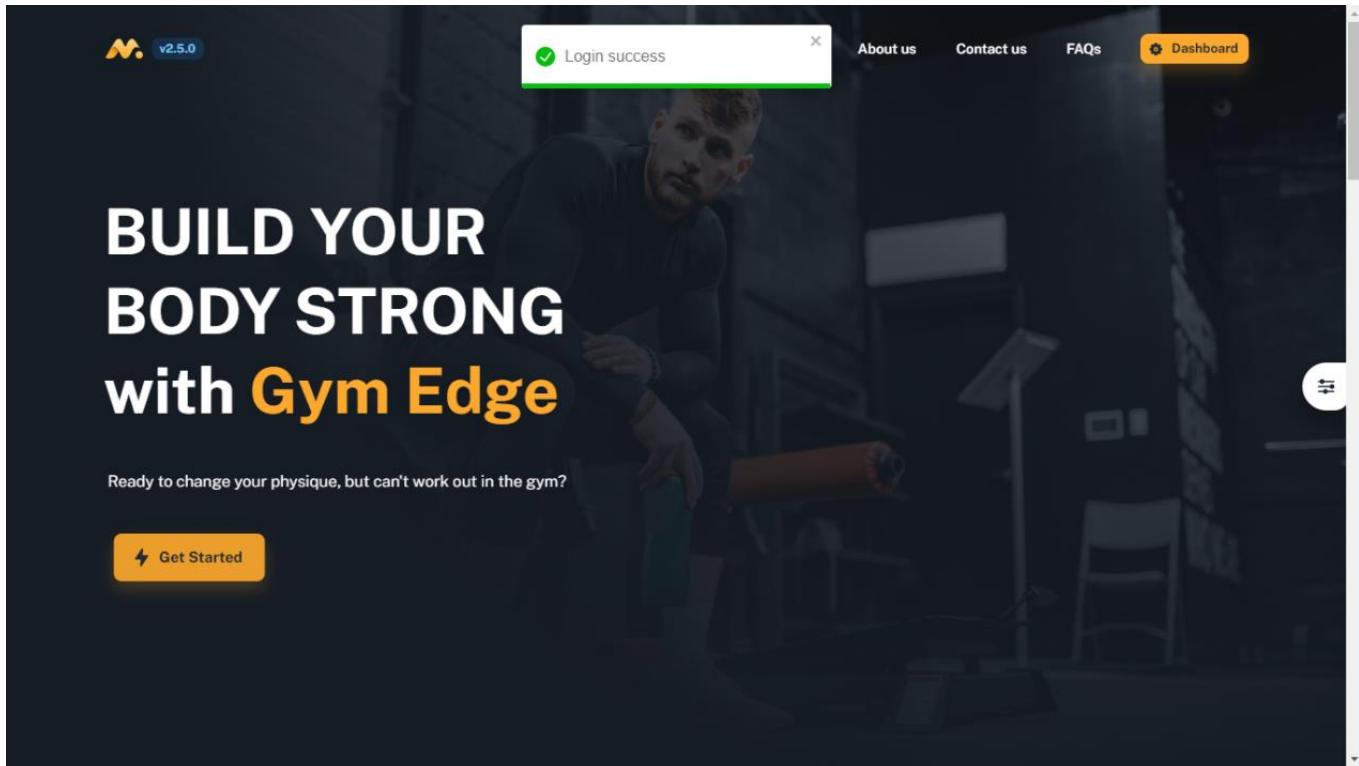


Figure 7.5: Website Home Page

- Subscription Plans Page

LICENSE	Standard	Standard Plus	Extended
Price	1500\$	1300\$	1800\$
Duration	3 months	5 months	8 months
Choose Plan	Choose Plan	Choose Plan	Choose Plan

Figure 7.6: Website Subscription Plans Page

- Payment Page

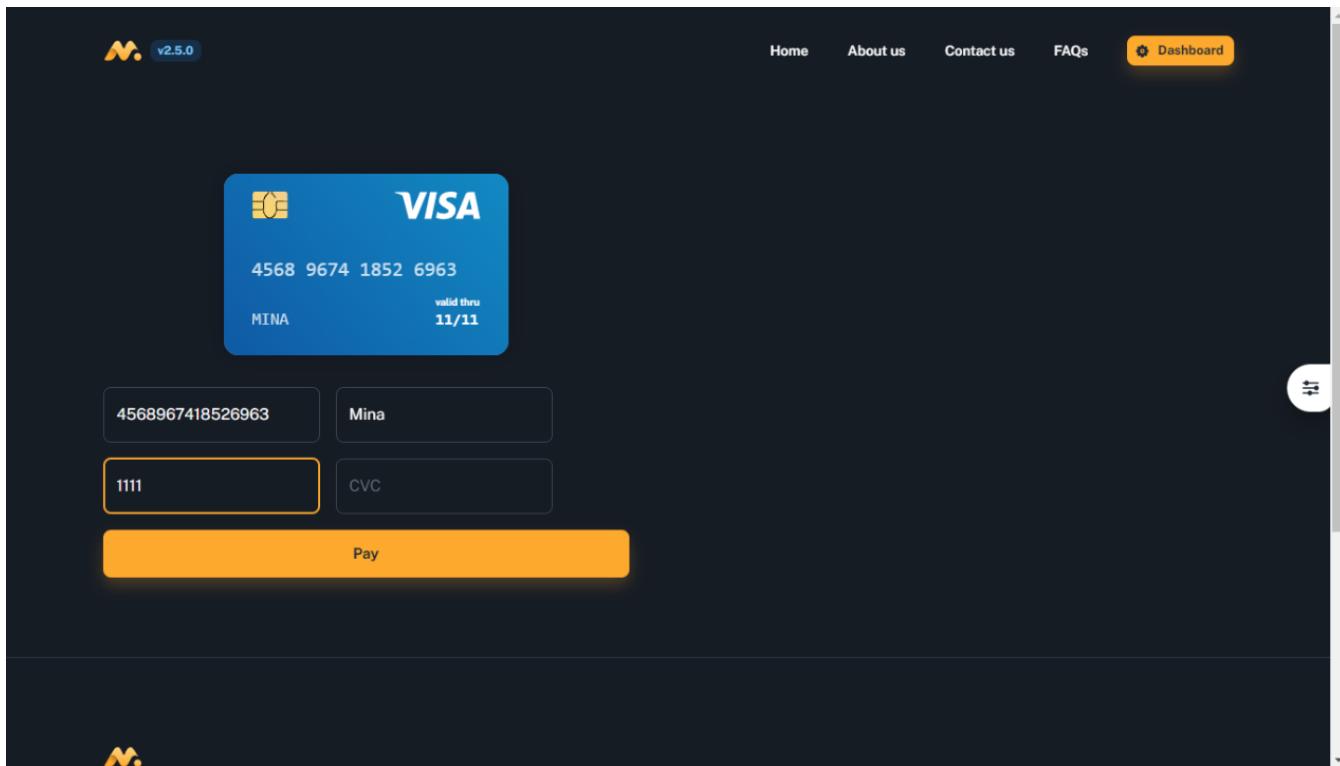


Figure 7.7: Website Payment Page

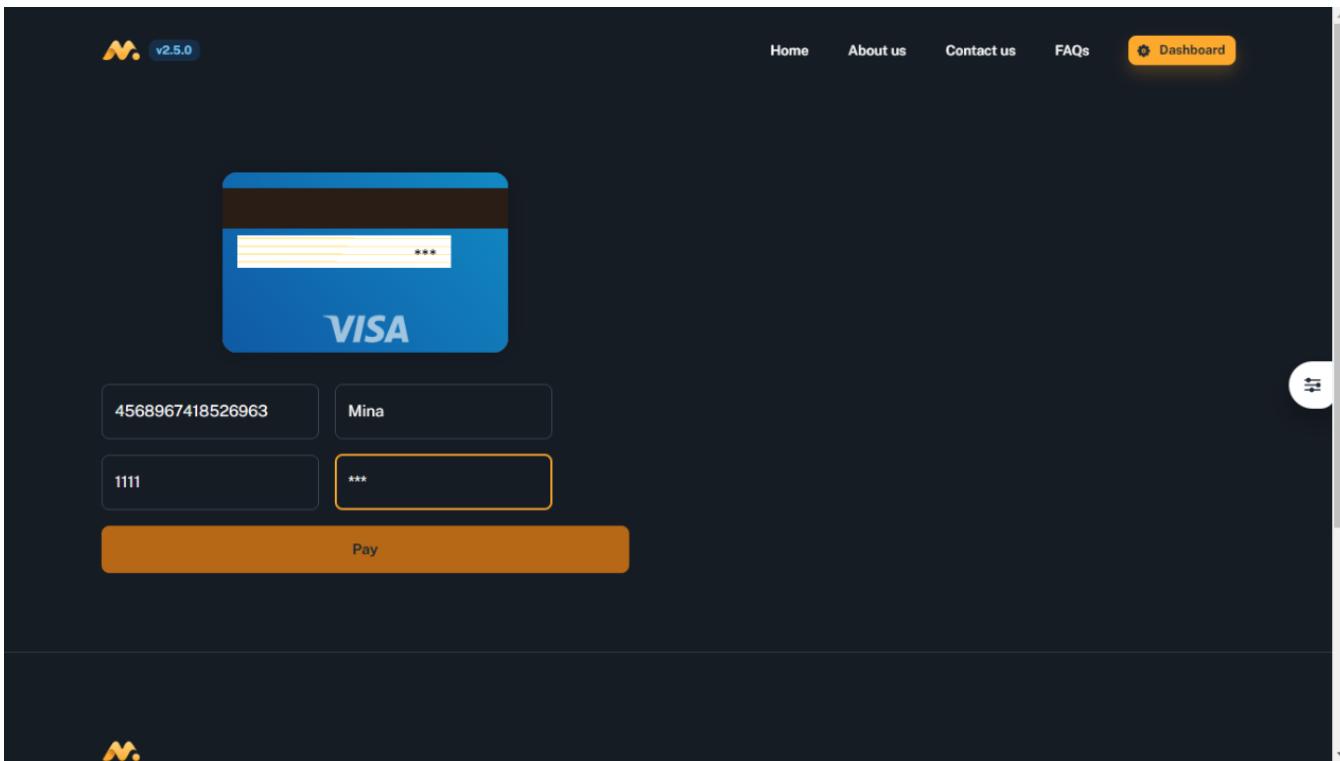


Figure 7.8: Website Payment Page

- Profile Page

The screenshot shows a dark-themed web application interface. On the left, there's a sidebar with sections for 'GENERAL' (App) and 'MANAGEMENT' (User, Profile, Users). The main area is titled 'Profile' and shows a user profile picture of a man with a beard. Below the picture, there's an 'About' section with the text 'Live at egypt' and 'mina@gmail.com'. To the right, there's a 'Subscribe' section with details: Price 1500\$, Duration 3 months, Start date 25/06/2023, and End date 25/09/2023. A green success message at the top says 'subscribe successfully'.

Figure 7.9: Website Profile Page

- Trainers Page

The screenshot shows a dark-themed web application interface for a gym. At the top, there's a navigation bar with links for Home, About us, Contact us, FAQs, and Dashboard. Below the navigation, a message reads: 'Gym Edge will provide you support if you have any problems, our support team will reply within a day.' The main content area displays four trainer profiles in cards:

- Yousef Ali** (Trainer): Shows a photo of a bald man with a tattooed arm, social media links (Facebook, Instagram, LinkedIn, Twitter), and a 'View Details' button.
- Karim Ali** (Trainer): Shows a photo of a smiling man in a blue t-shirt, social media links, and a 'View Details' button.
- Abdo Div** (Trainer): Shows a photo of a man in a blue shirt, a testimonial in Arabic, and a 'View Details' button.
- Ahmed Ali** (Trainer): Shows a photo of a man with a beard, social media links, and a 'View Details' button.

At the bottom, there's a button labeled 'View All Team Members →'.

Figure 7.10: Website Trainers Page

- Trainer's profile Page

Karim Ali
karim@gmail.com
011178758766
Skills
Youga Boxing Gym

LICENSE

Standard

- ✓ Price 5000\$
- ✓ 8 months

Choose Plan

LICENSE

Standard Plus

- ✓ Price 4000\$
- ✓ 3 months

Choose Plan

LICENSE

Extended

- ✓ Price 2000\$
- ✓ 1 months

Choose Plan

Figure 7.11: Website Trainer's Profile Page

- Admin Panel Page

User List

Dashboard • User • List

+ New User

#	Photo	Name	Role	Actions
1		Karim Ali	trainer	Delete Edit
2		jonas wael	trainee	Delete Edit
3		Moaz Mohamed	user	Delete Edit
4		Mina Youssef	trainee	Delete Edit
5		Ahmed Samir	trainee	Delete Edit
6		Chad Randall	trainee	Delete Edit
7		Randy Morris	trainee	Delete Edit
8		Henry Hansen	trainee	Delete Edit
9		Michael Paul	trainee	Delete Edit

Figure 7.12: Website Admin Panel Page

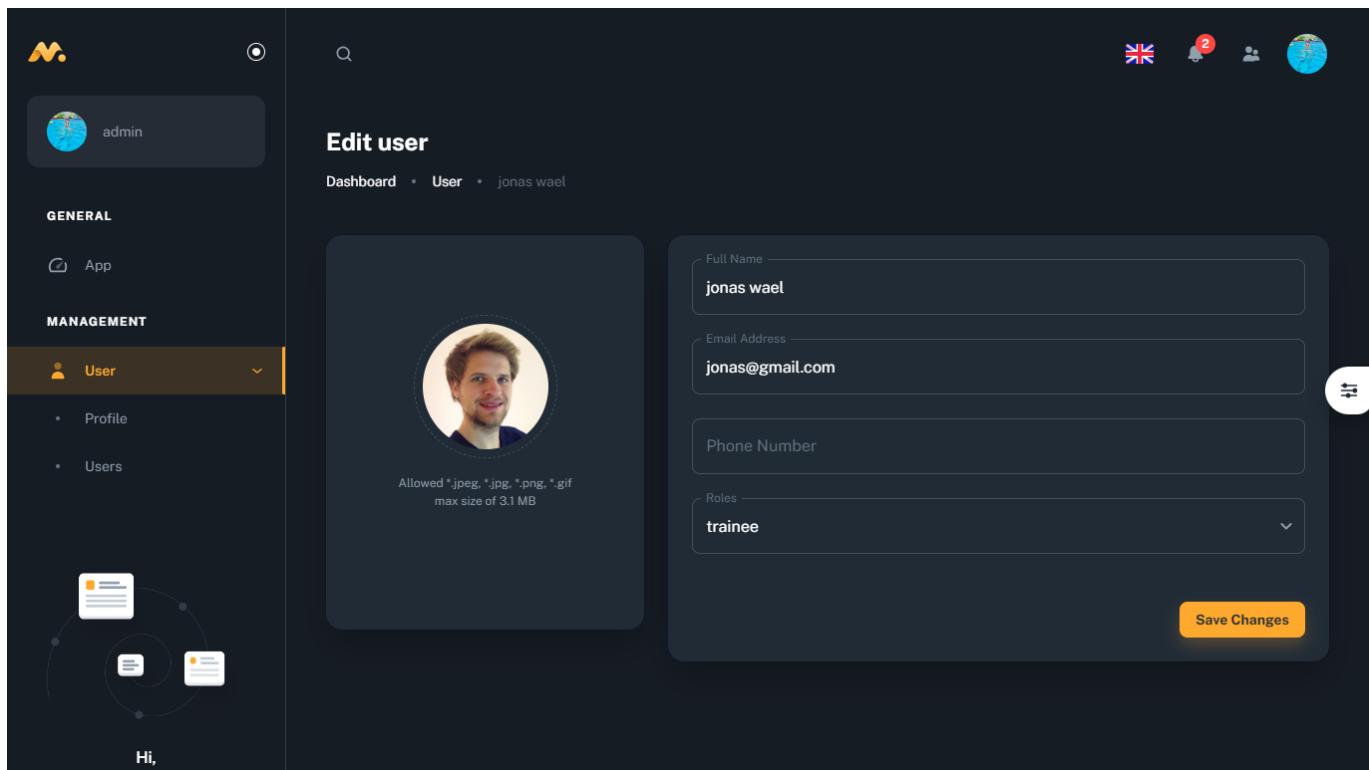
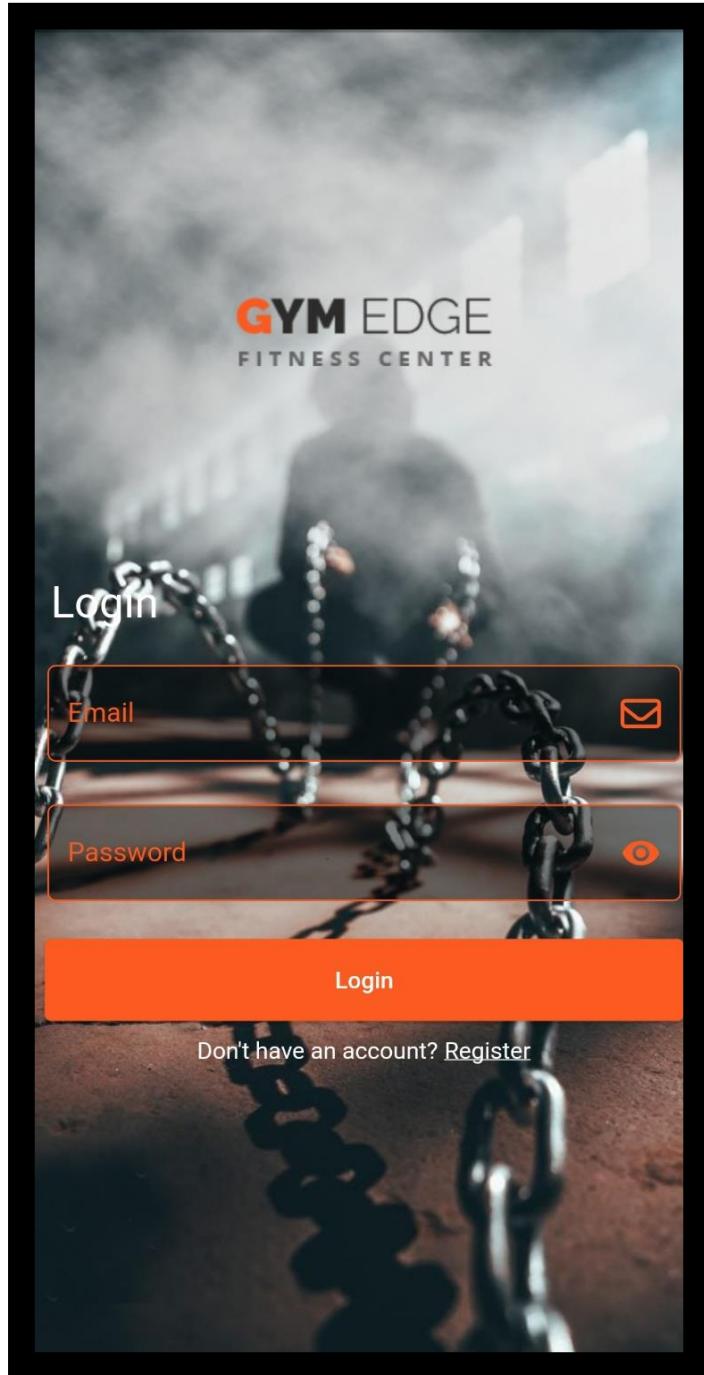


Figure 7.13: Website Edit user Page (admin panel)

5.2 Mobile Application

Login Screen



Registration Screen

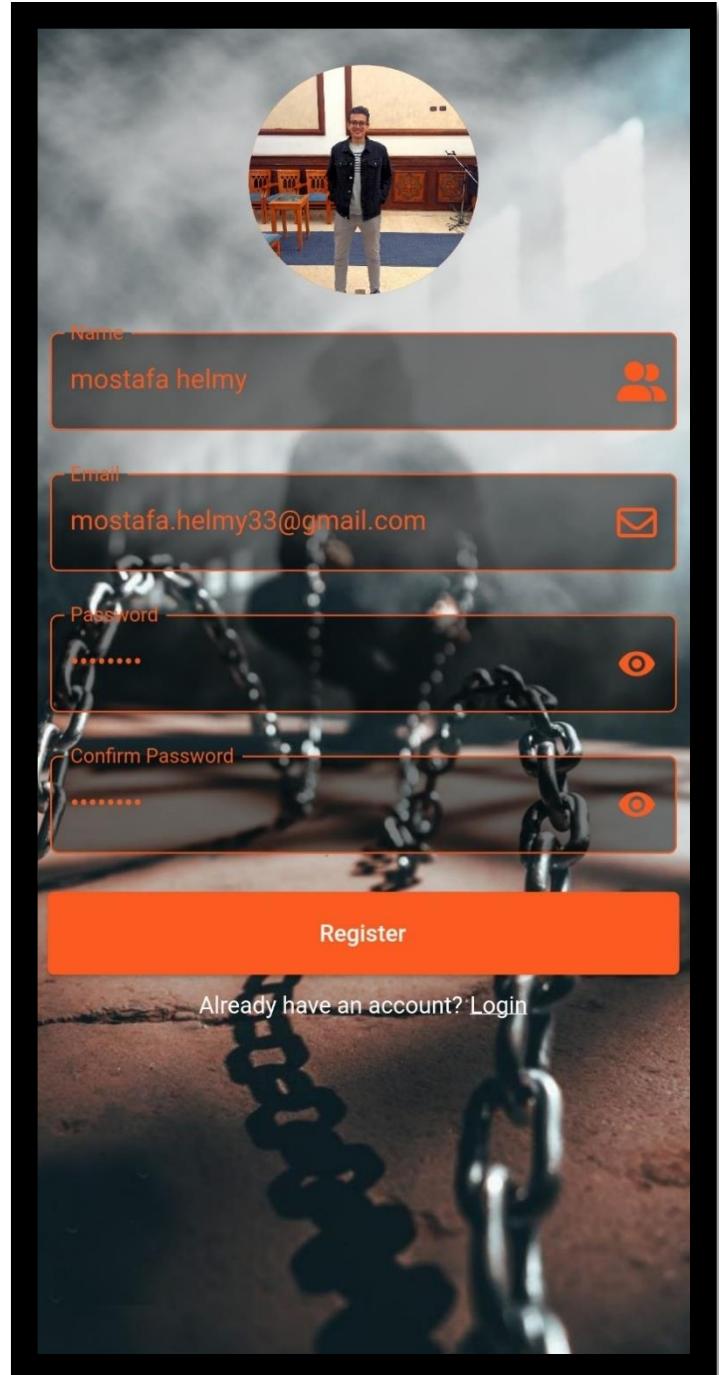
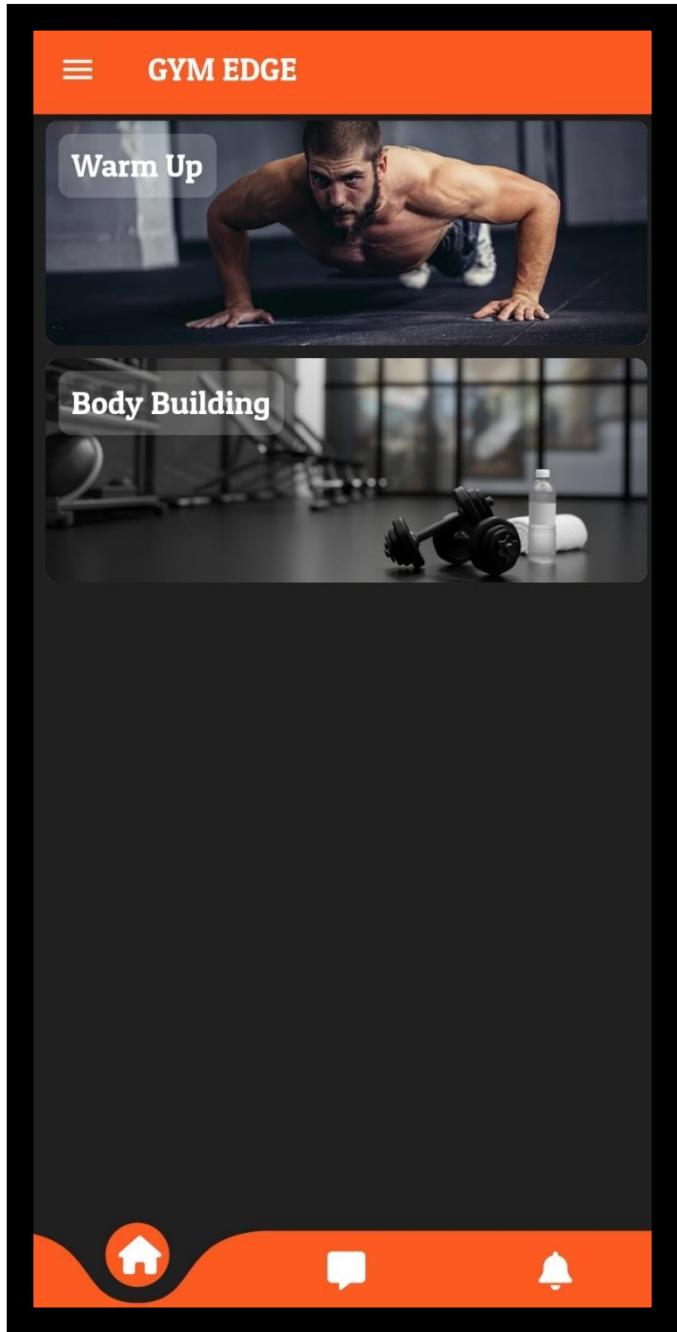


Figure 8.1: Mobile Application Login Screen

Figure 8.2: Mobile Application Registration Screen

Home Screen



App Drawer

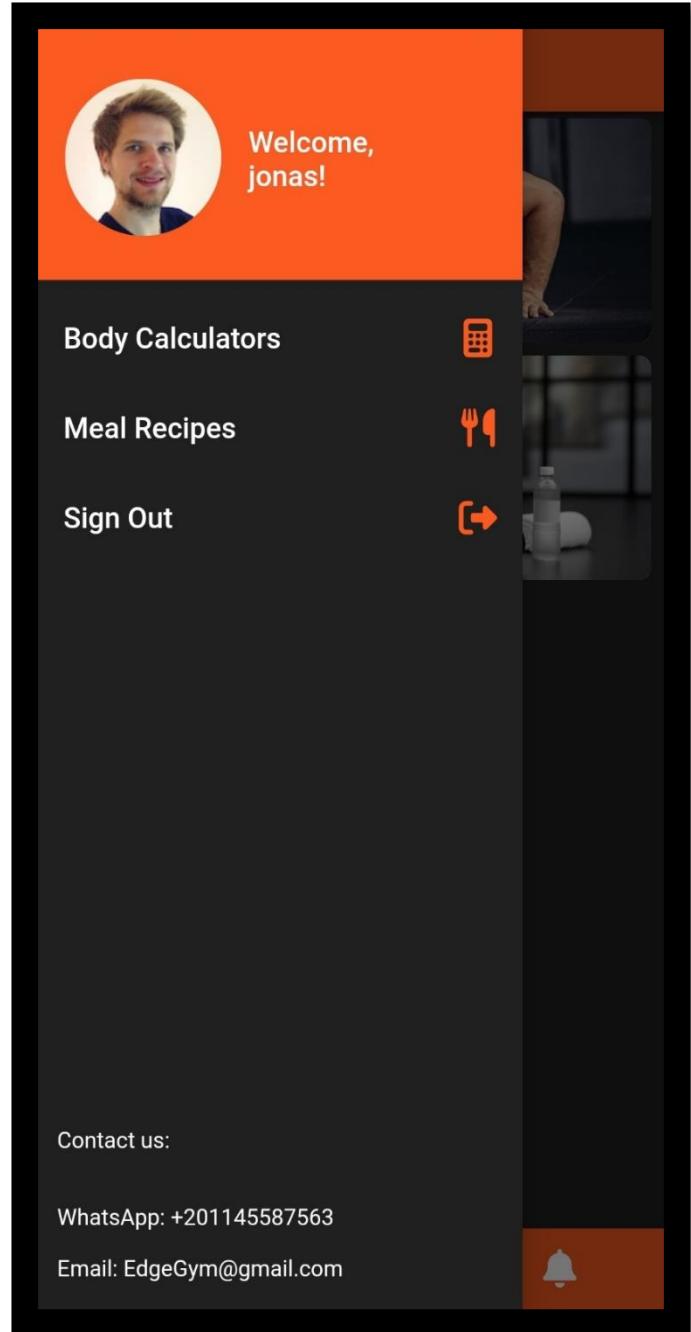


Figure 8.3: Mobile Application Home Screen

Figure 8.4: Mobile Application App drawer

User's profile Screen

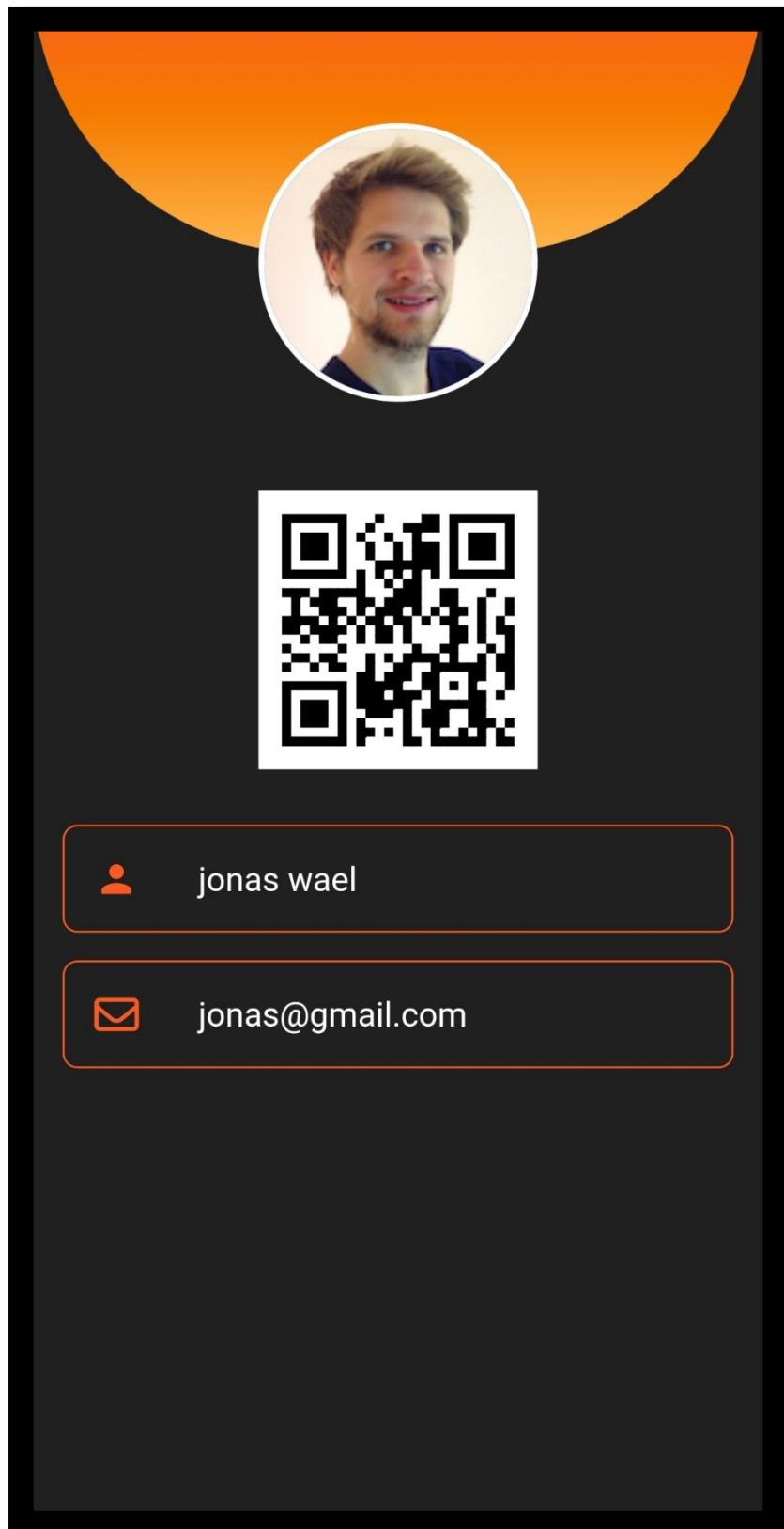


Figure 8.5: Mobile Application User's Profile Screen

Exercises Screens

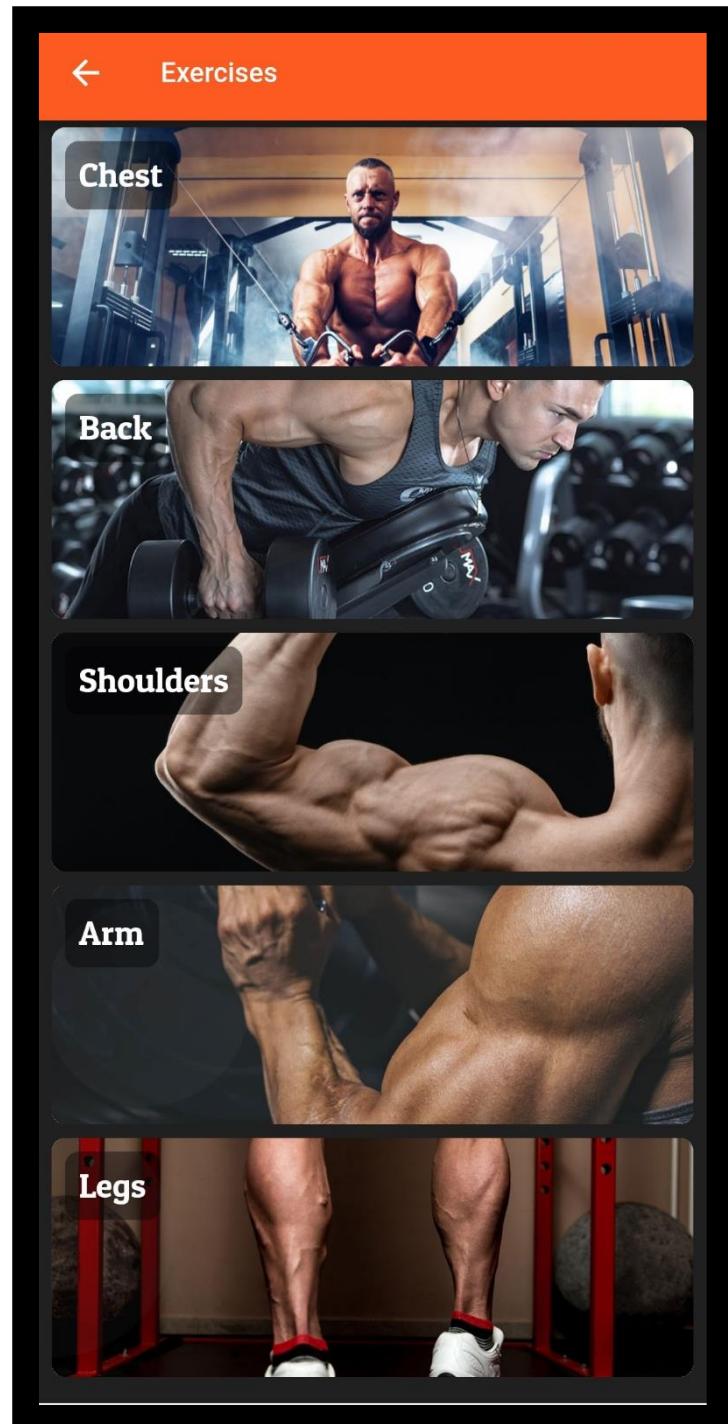


Figure 8.6: Mobile Application Exercises Category Screen

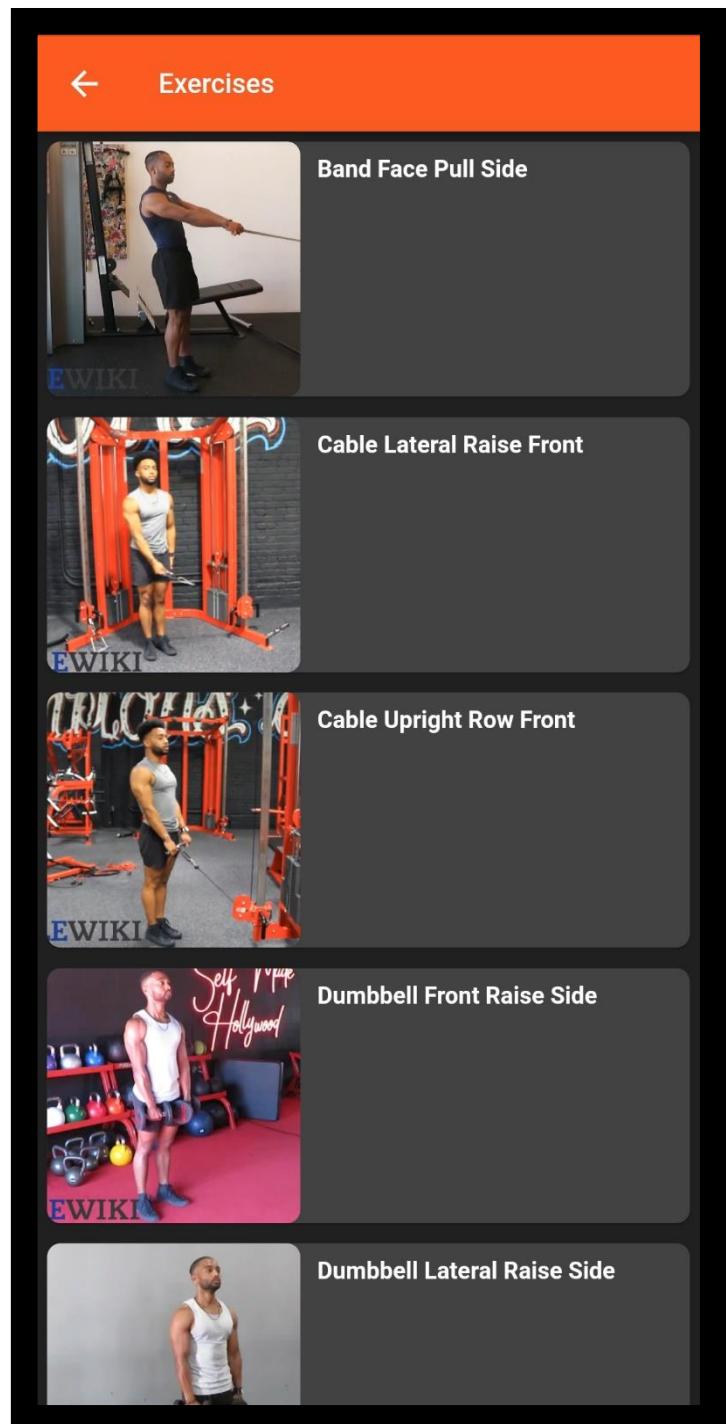


Figure 8.7: Mobile Application Exercises Screen

Exercise Video Screen

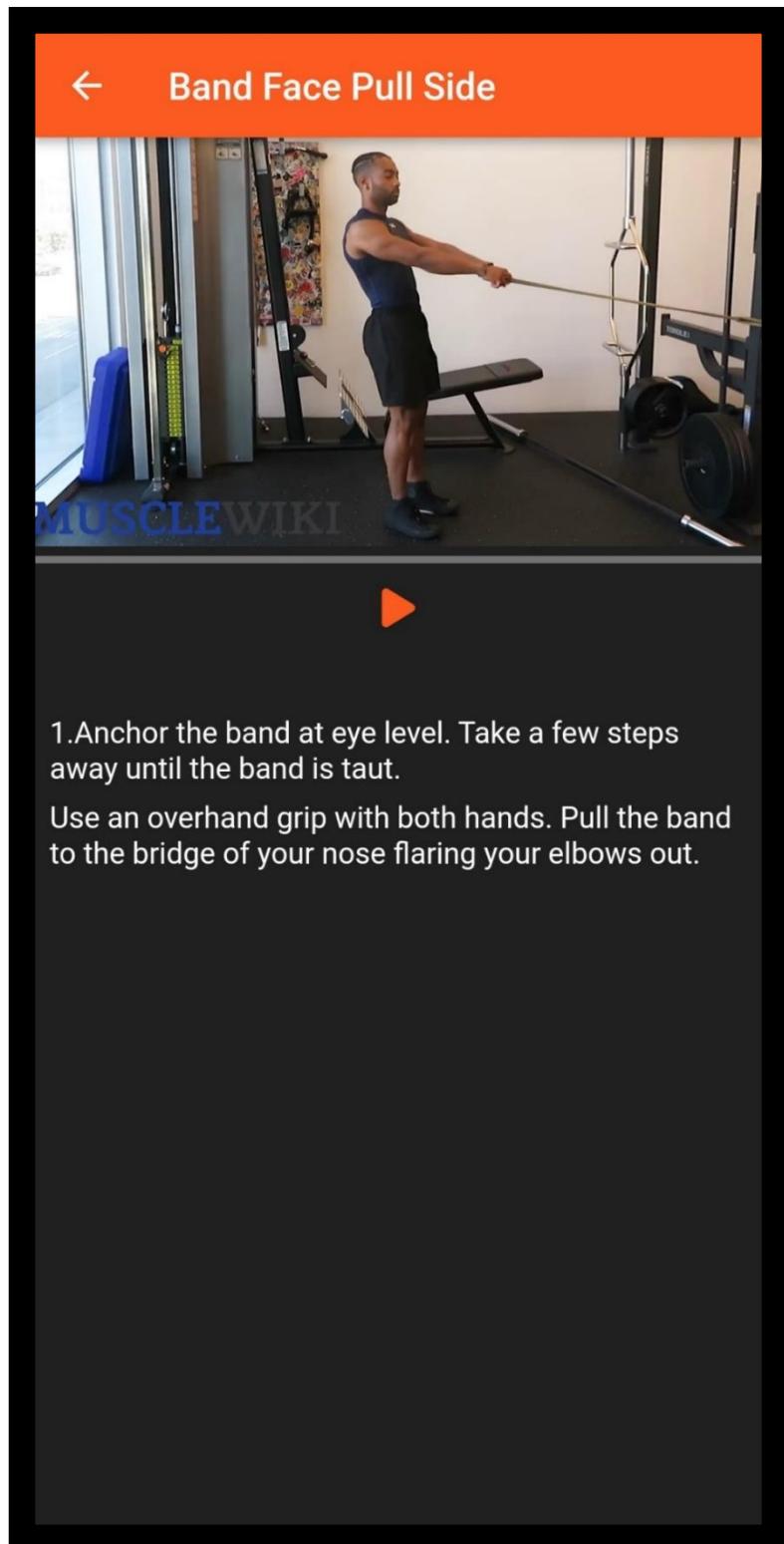
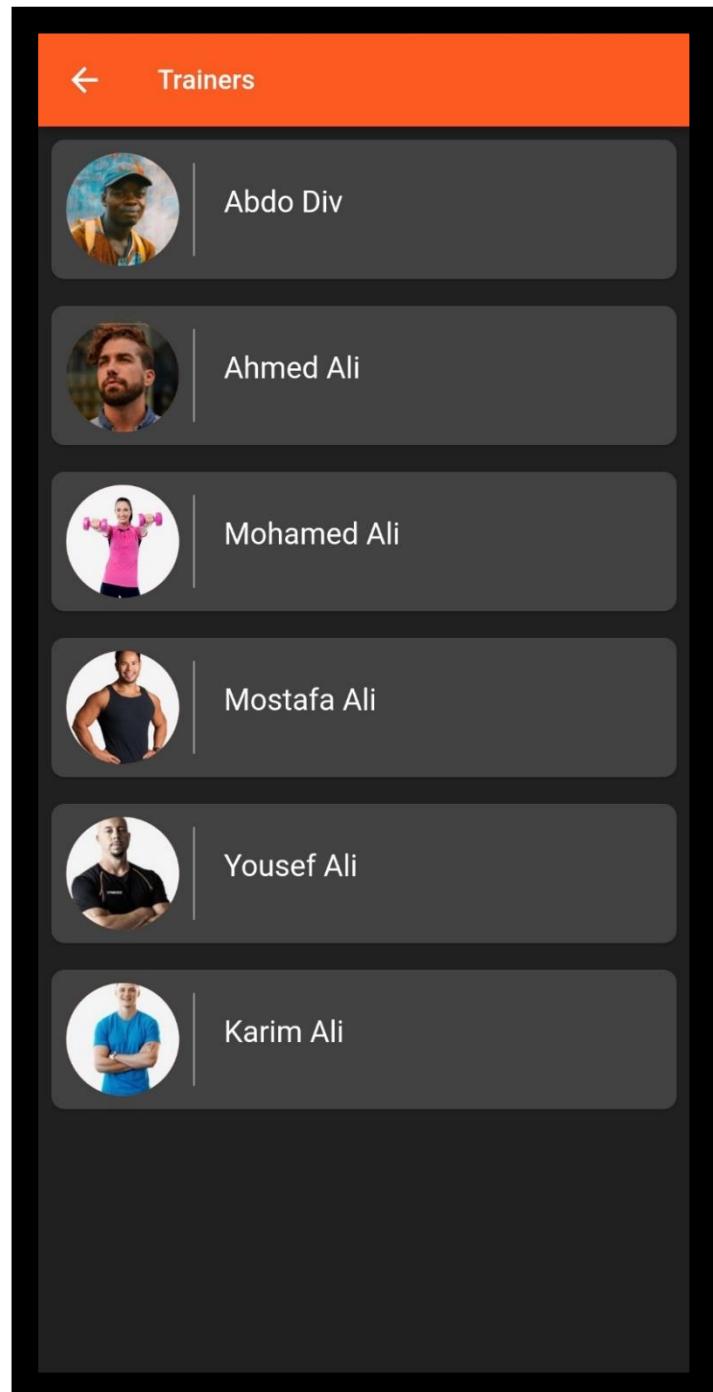


Figure 8.8: Mobile Application Exercise Video Screen

Trainers Screen



Trainer's profile Screen

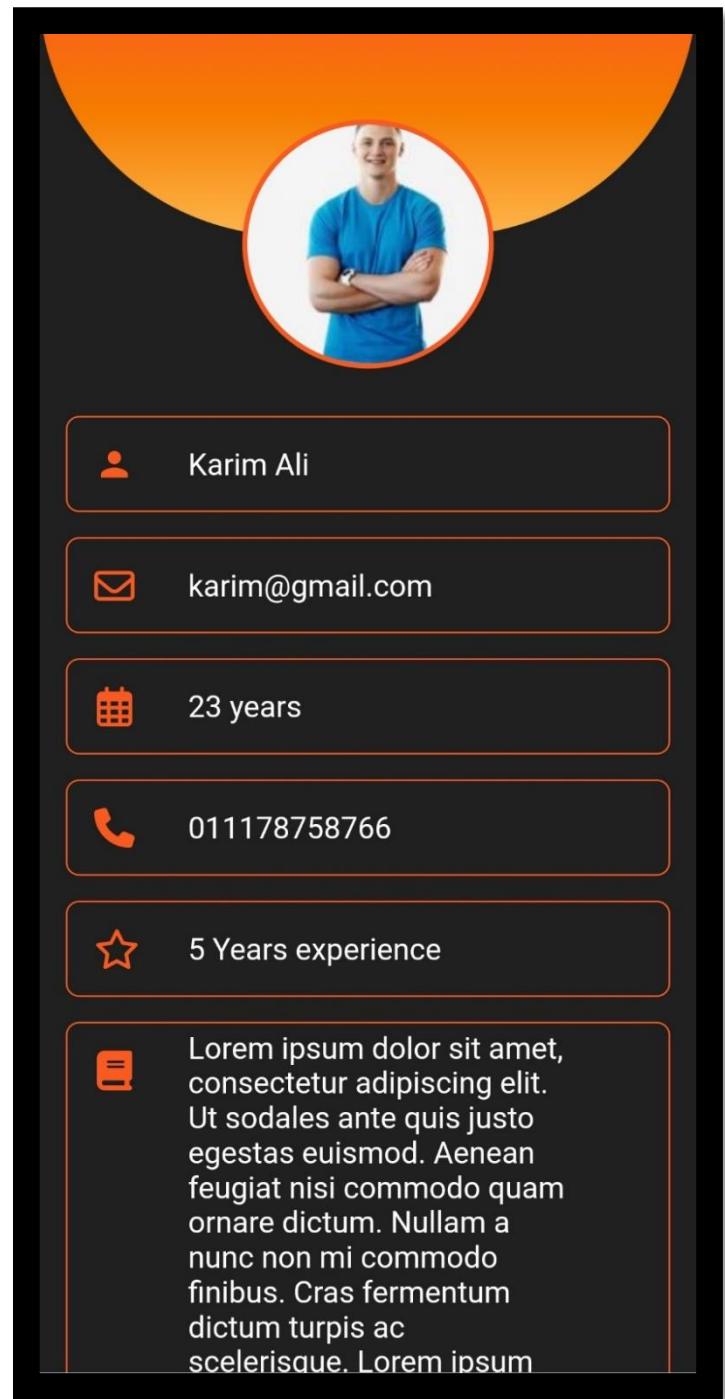


Figure 8.9: Mobile Application Trainers Screen

Figure 8.10: Mobile Application Trainer's Profile Screen

Chat Screens

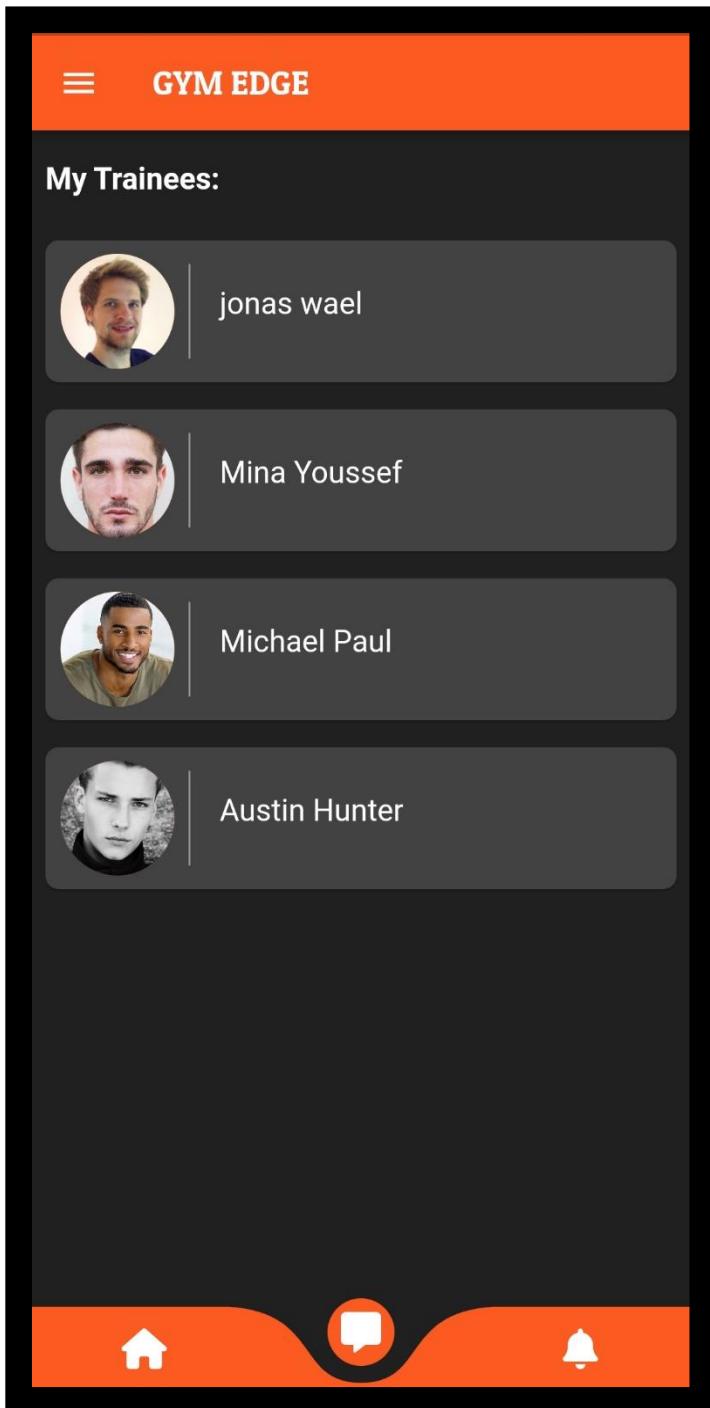


Figure 8.11: Mobile Application Conversations Screen

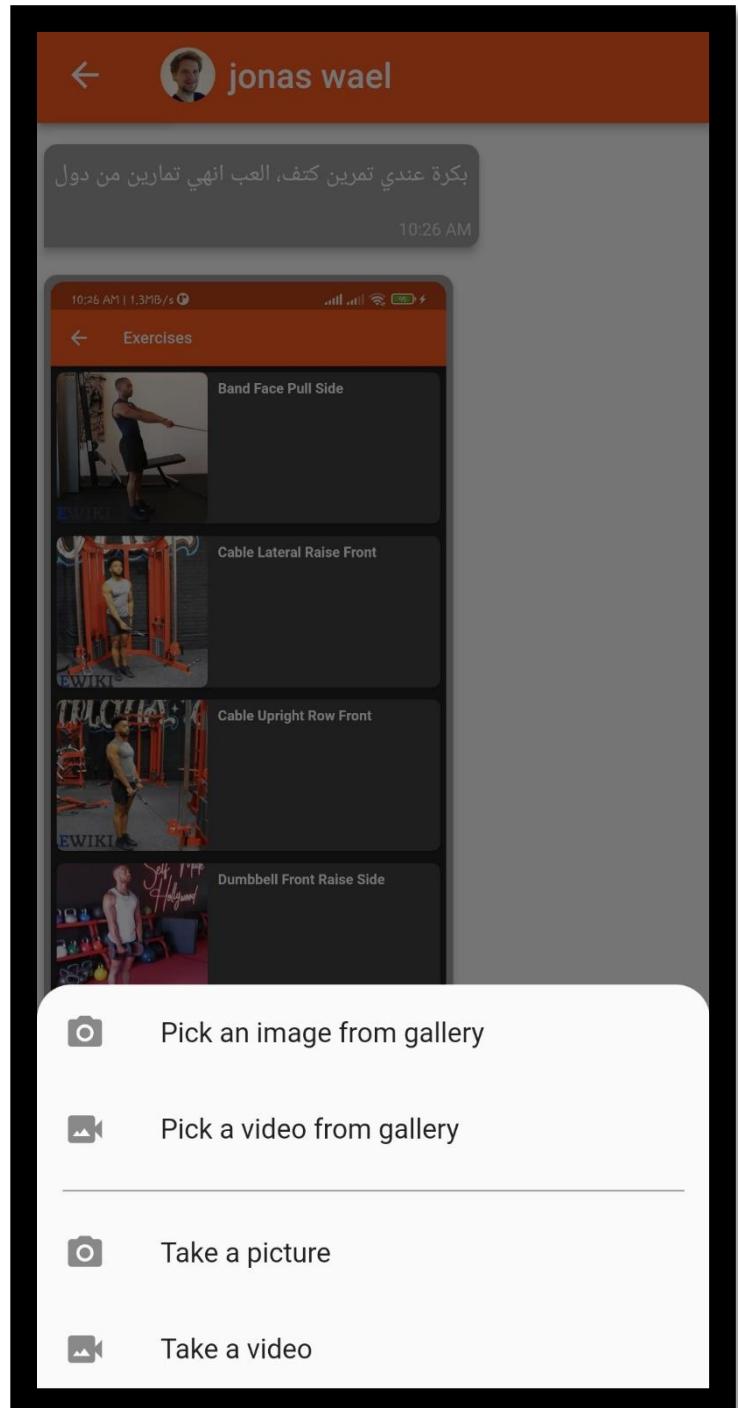


Figure 8.12: Mobile Application Chat Pick Media Screen

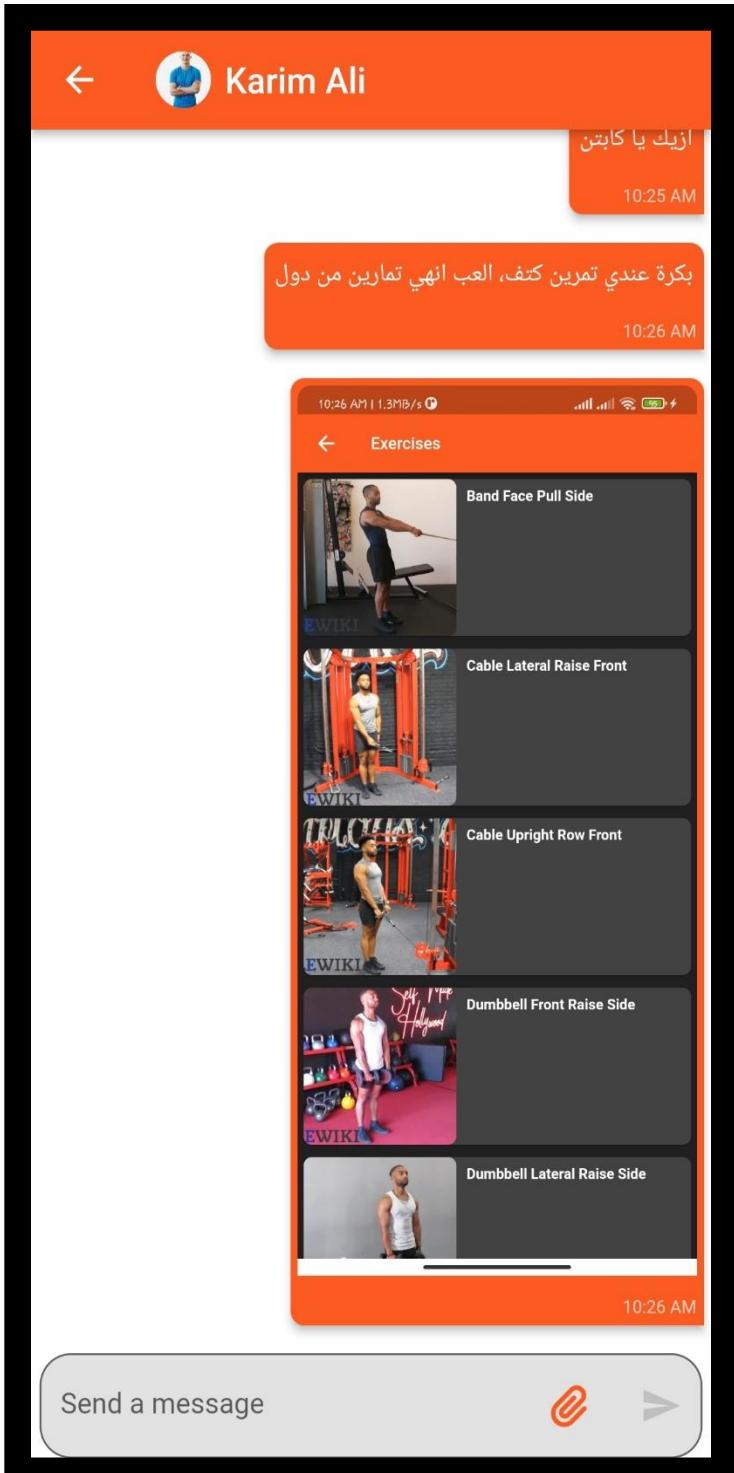


Figure 8.13: Mobile Application Chat from Trainee to Trainer

Figure 8.14: Mobile Application Notification from Chat

Announcement Screen



Figure 8.15: Mobile Application Chat from Trainer to Trainee

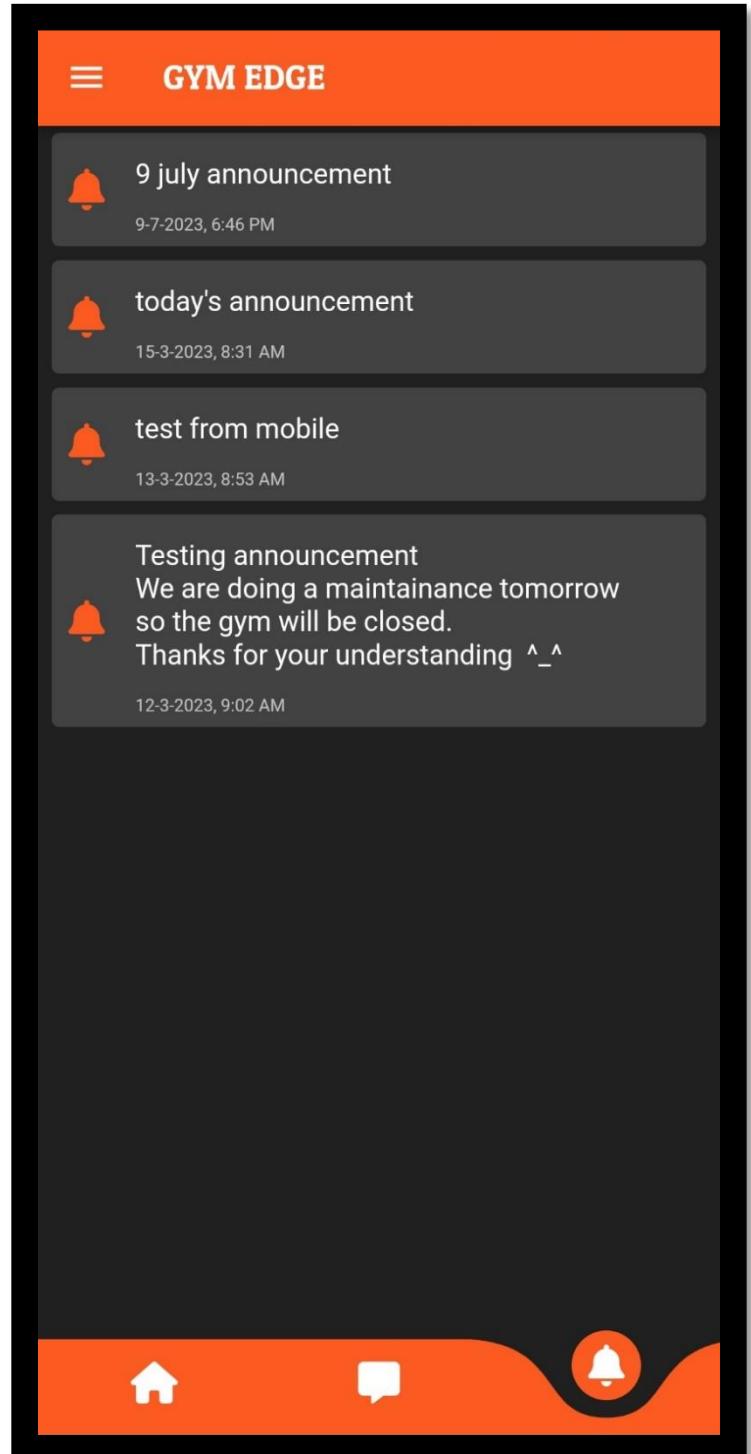
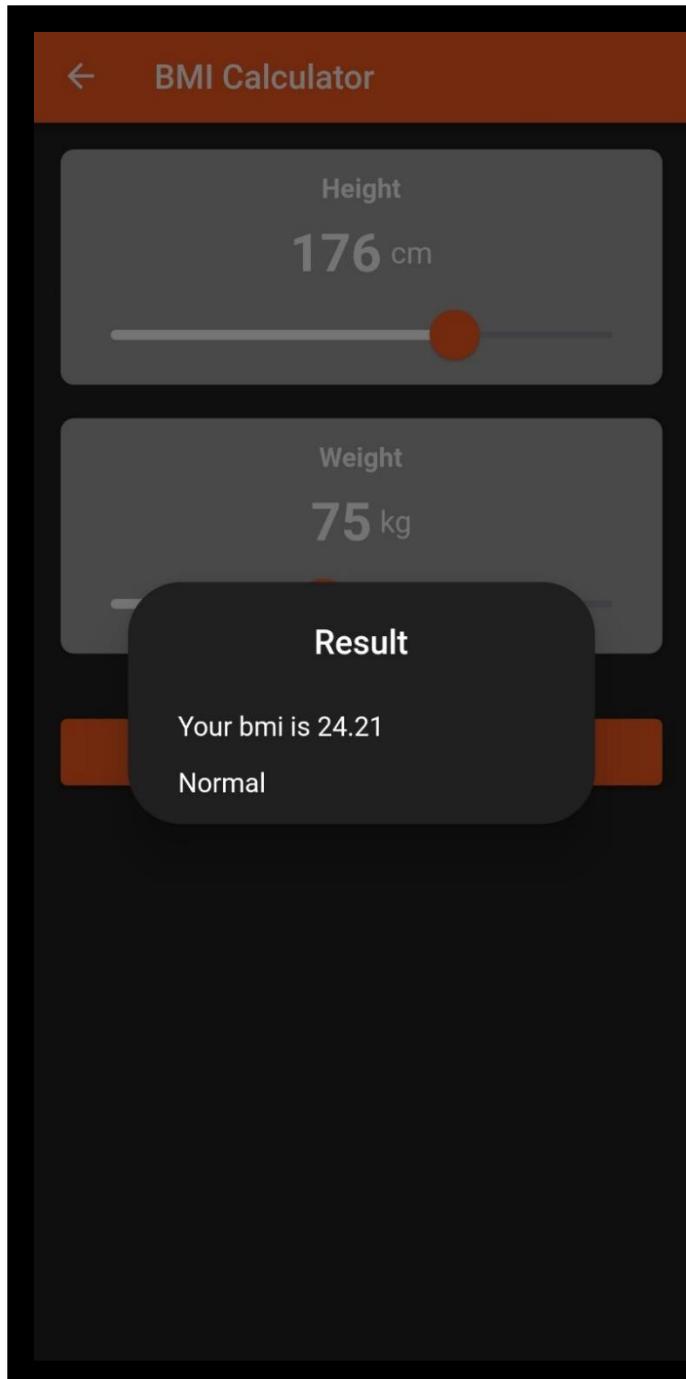


Figure 8.16: Mobile Application Announcements Screen

BMI Calc Screen



BMR Calc Screen

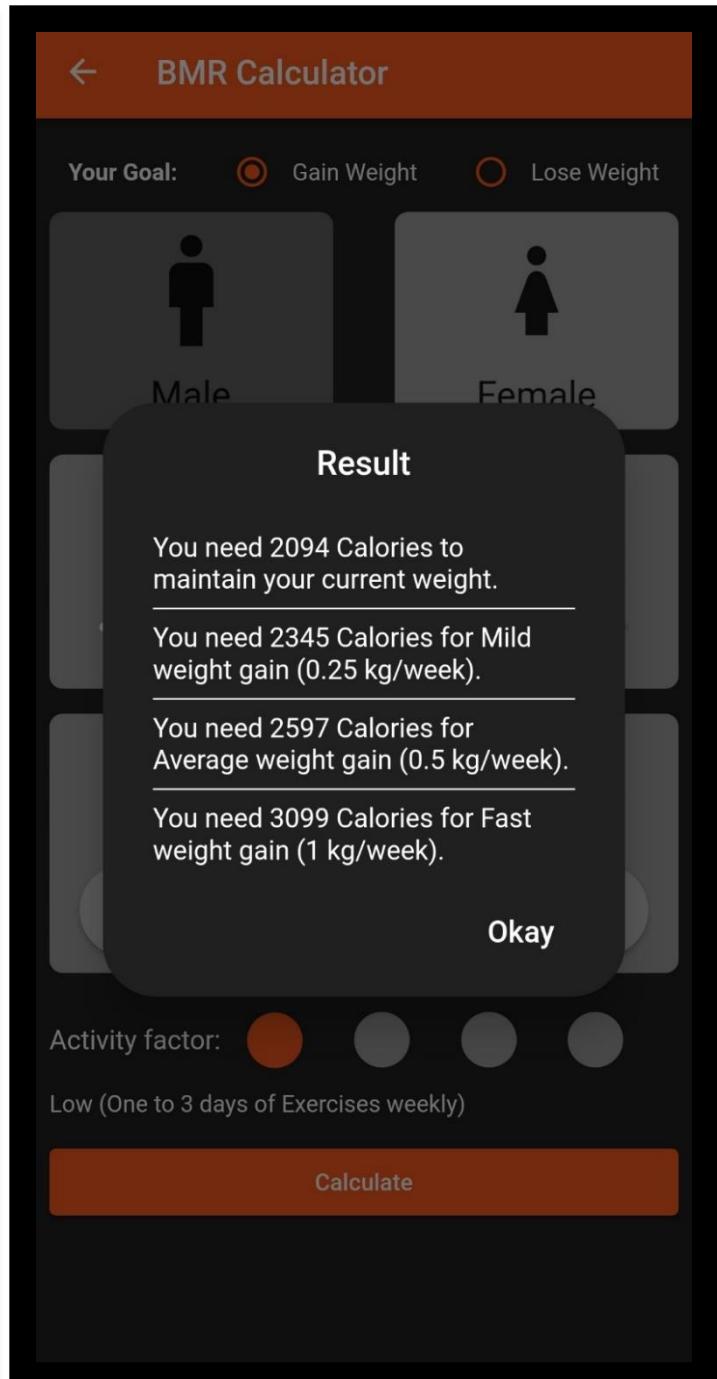


Figure 8.17: Mobile Application BMI Calculator Screen

Figure 8.18: Mobile Application BMR Calculator Screen

Meal Recipes Screen

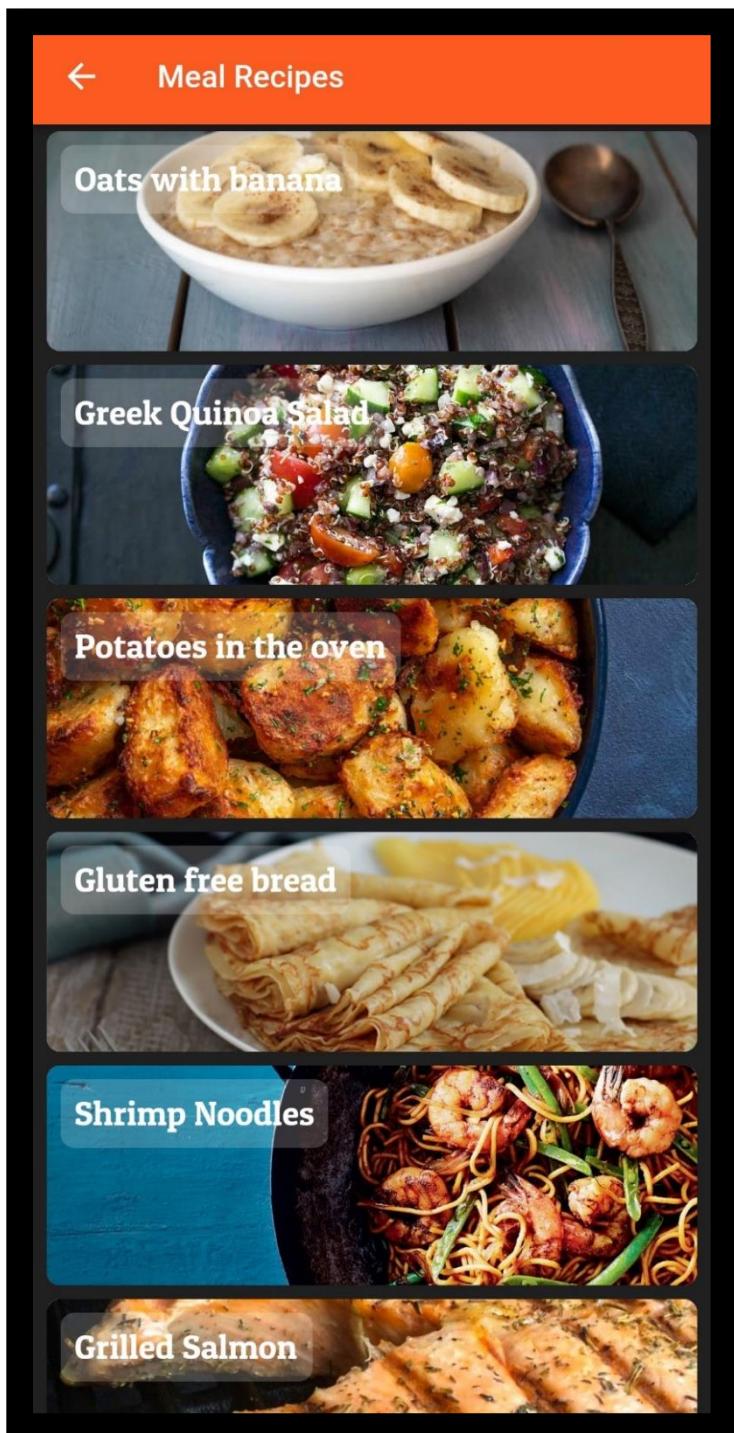


Figure 8.19: Mobile Application All Recipes Screen

This image shows a detailed view of a recipe card for "Oats with banana". At the top left is a fork and knife icon. The title "Oats with banana" is at the top right. The main content area includes nutritional information and ingredient lists.

القيمة الغذائية للوجبة:

- كالوريز (سعرات حرارية) : 478 كالوري
- بروتين : 19 جرام
- كربوهيدرات : 79 جرام
- دهون : 9 جرام

المكونات:

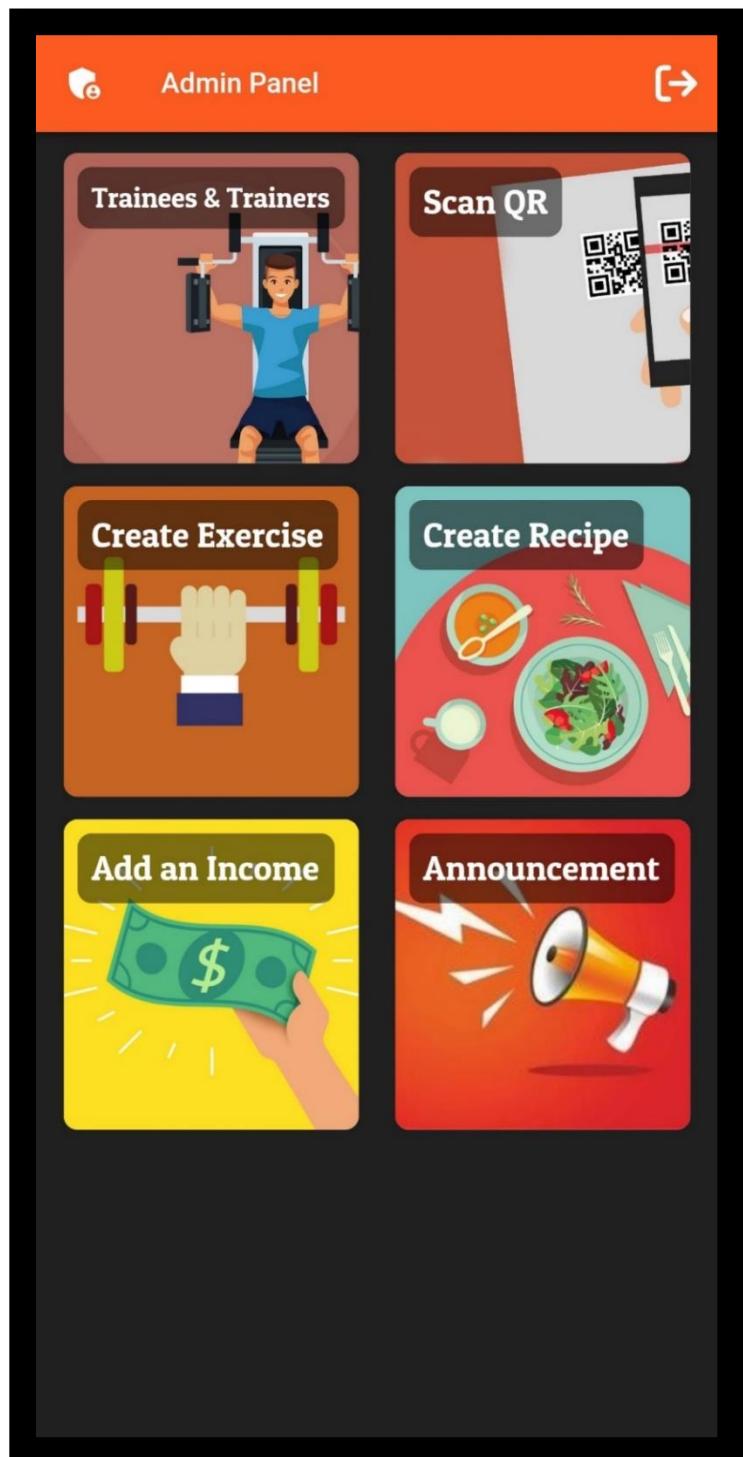
- 100 جرام شوفان
- 230 جرام حليب
- 50 جرام موز
- 15 جرام بذور الشيا
- رشة ملح

الخطوات:

- امزج الموز بكوب من الحليب
- ضع الشوفان في وعاء كبير، أضيف حليب الموز وباقى المكونات

Figure 8.20: Mobile Application Recipe Screen

Admin Panel Screen



Trainees Screen

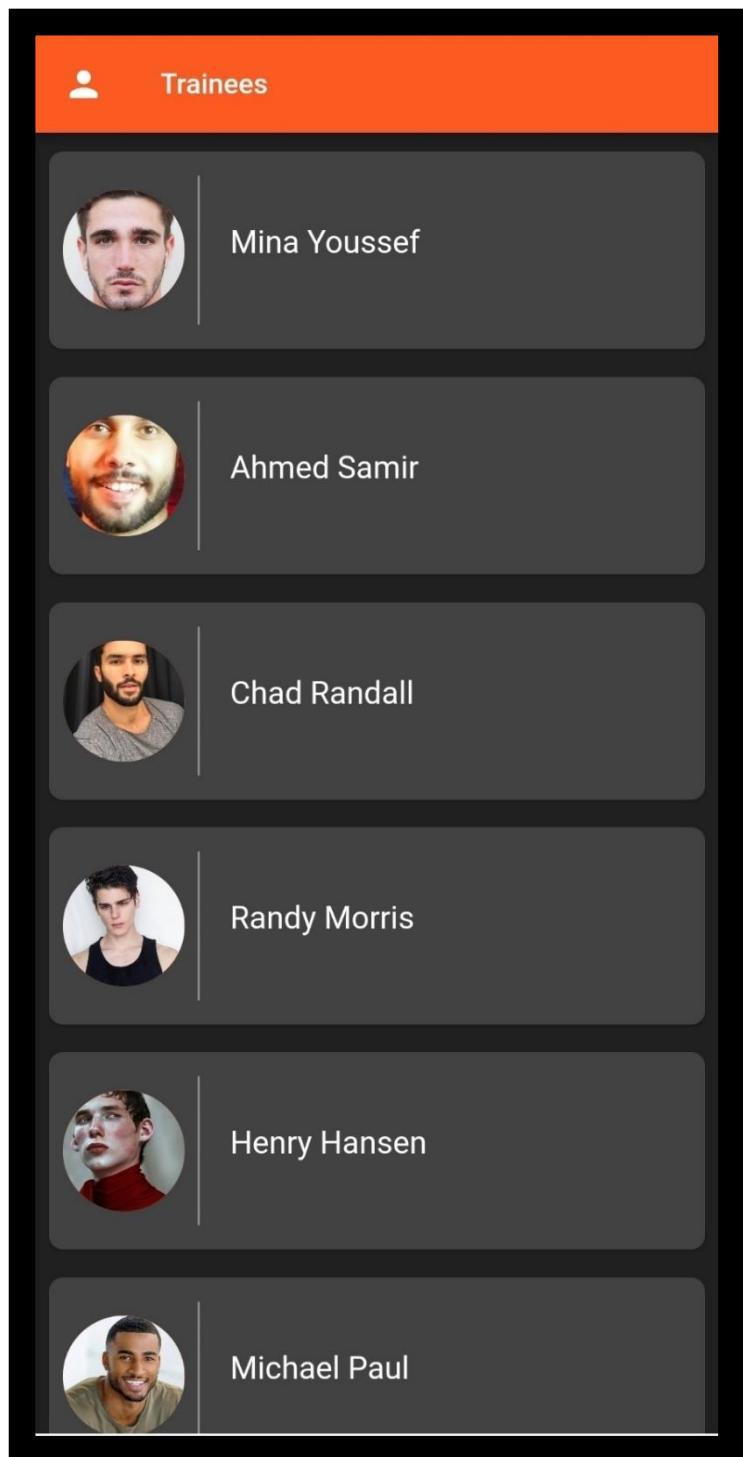
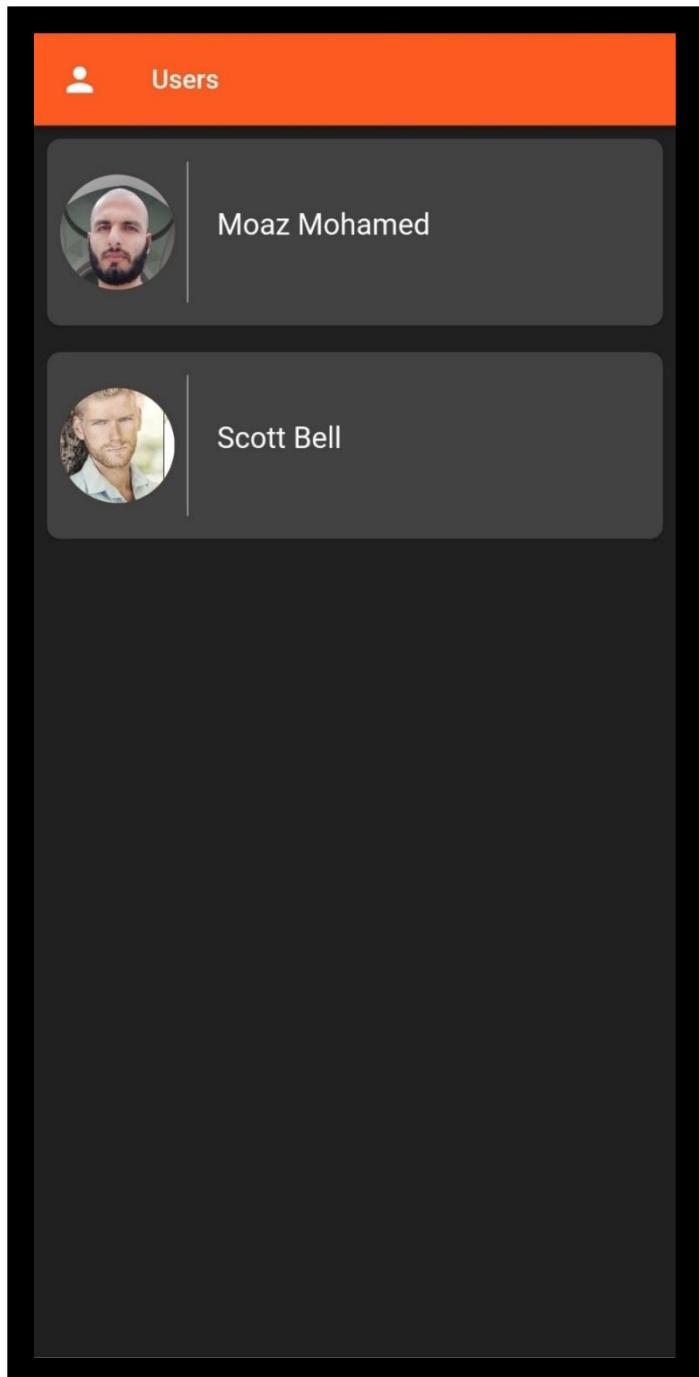


Figure 8.21: Mobile Application Admin Panel Screen

Figure 8.22: Mobile Application Trainees Screen

Users Screen



QR code Scan



Figure 8.23: Mobile Application unsubscribed Users Screen

Figure 8.24: Mobile Application Scan QR Screen

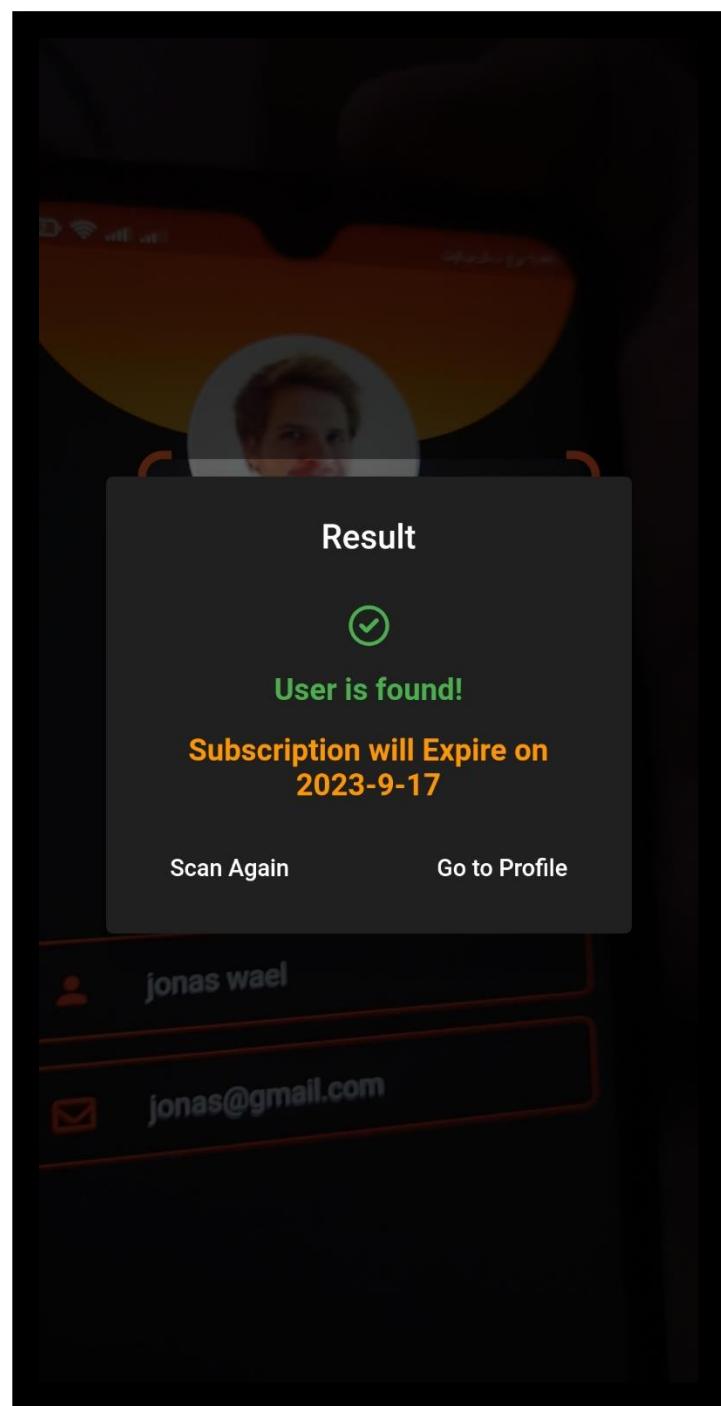


Figure 8.25: Mobile Application Result of QR Scan Screen (User is Found and Valid)

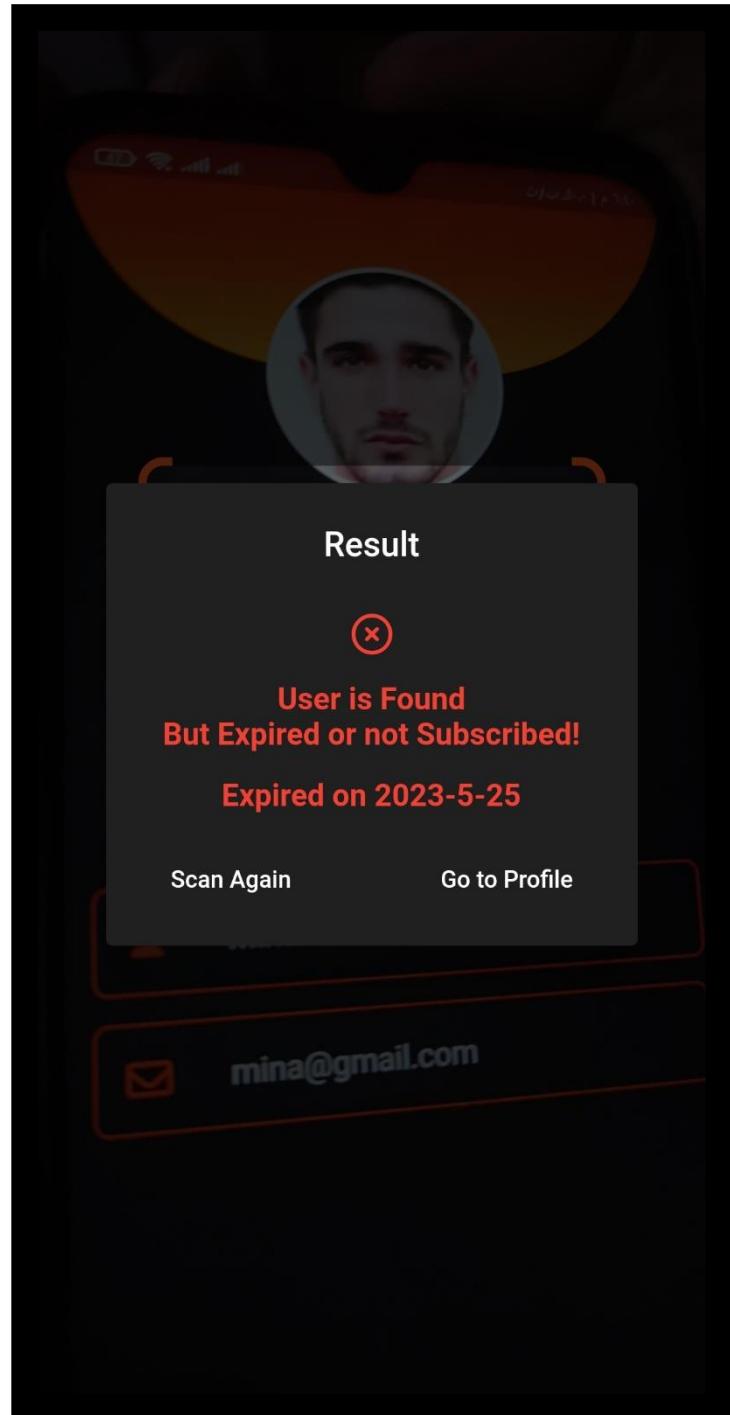
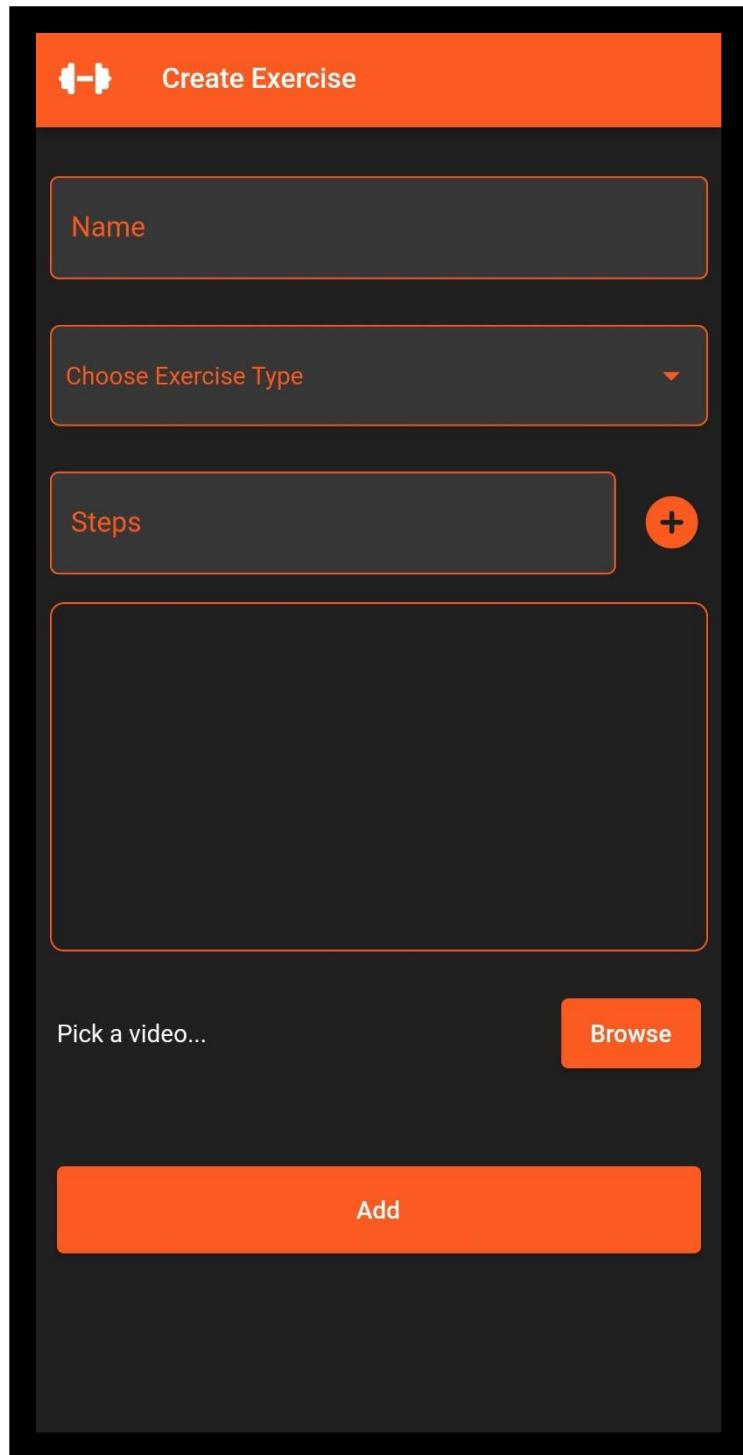


Figure 8.26: Mobile Application Result of QR Scan Screen (User is Found but not Valid)

Create Exercise Screen



Create Recipe Screen

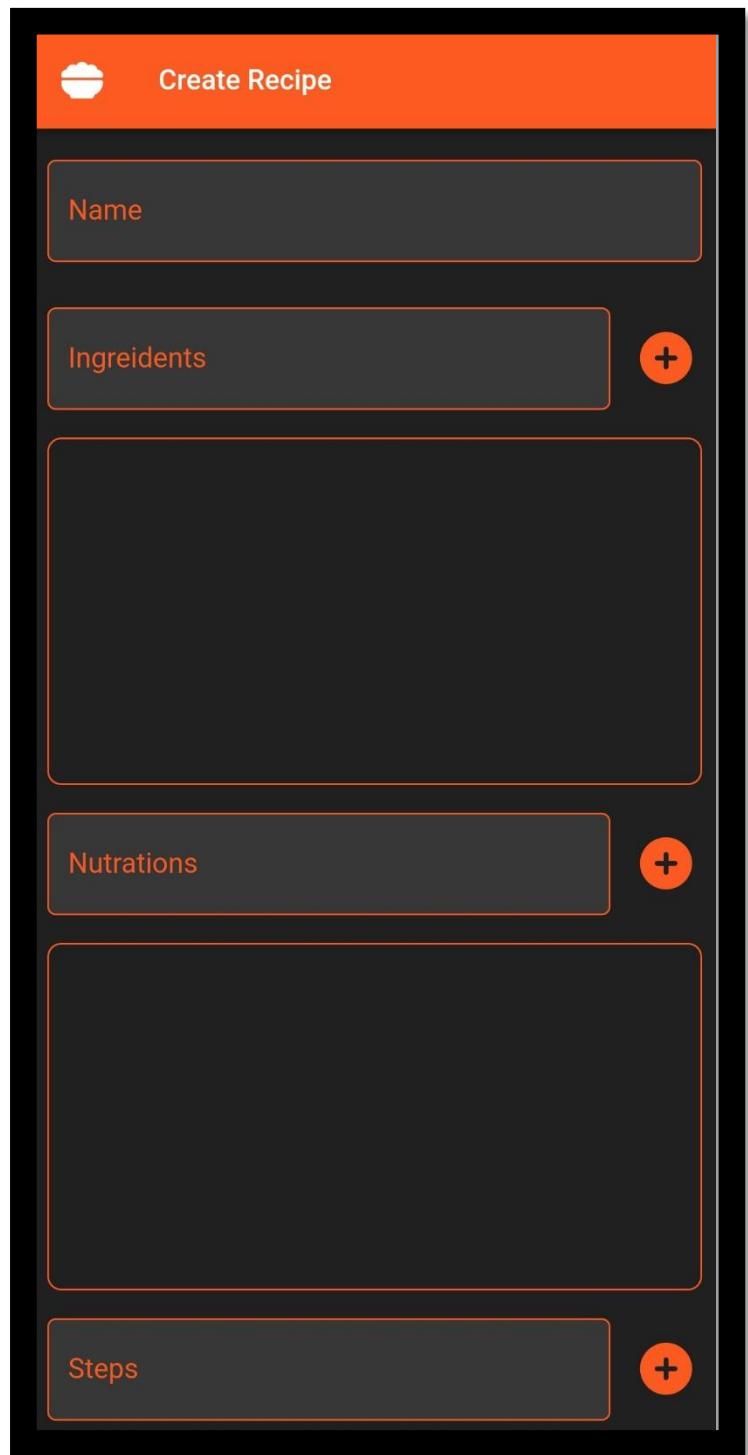
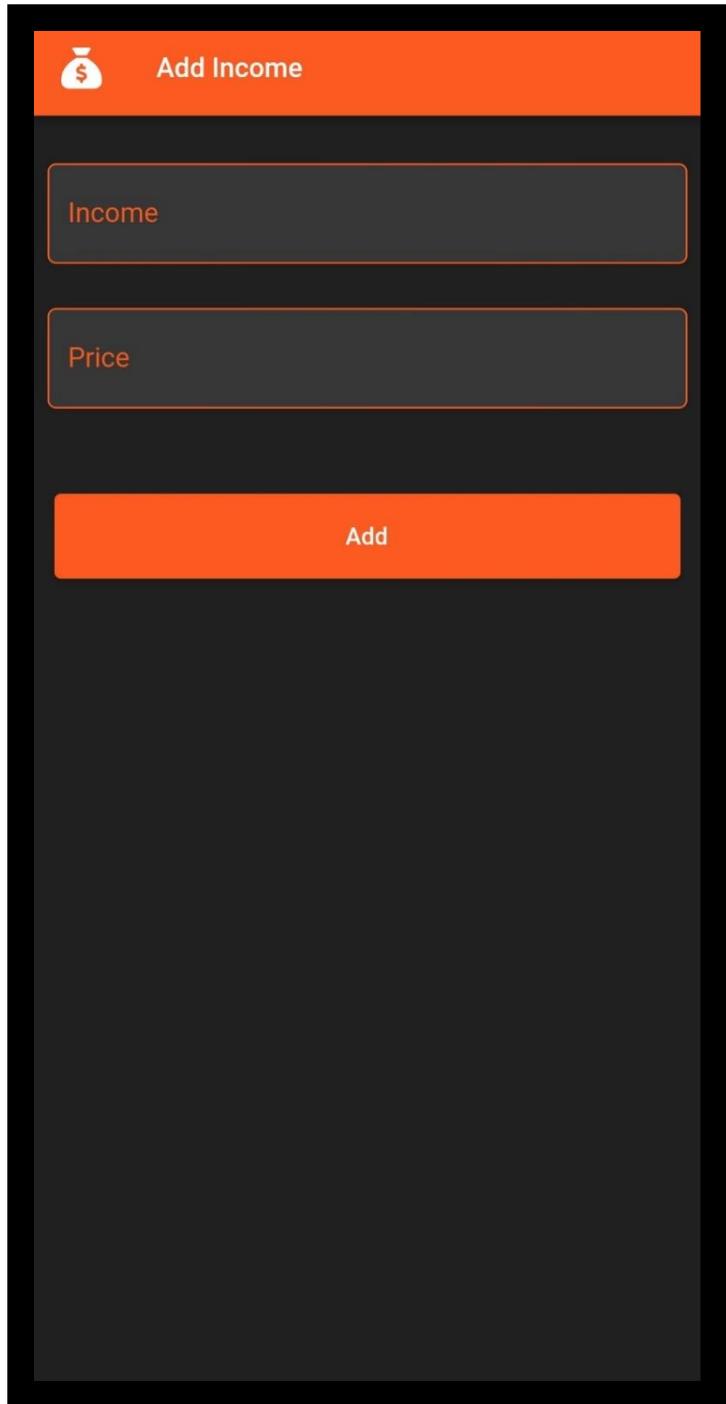


Figure 8.27: Mobile Application Create Exercise Screen

Figure 8.28: Mobile Application Create Recipe Screen

Add Income



Add Announcement

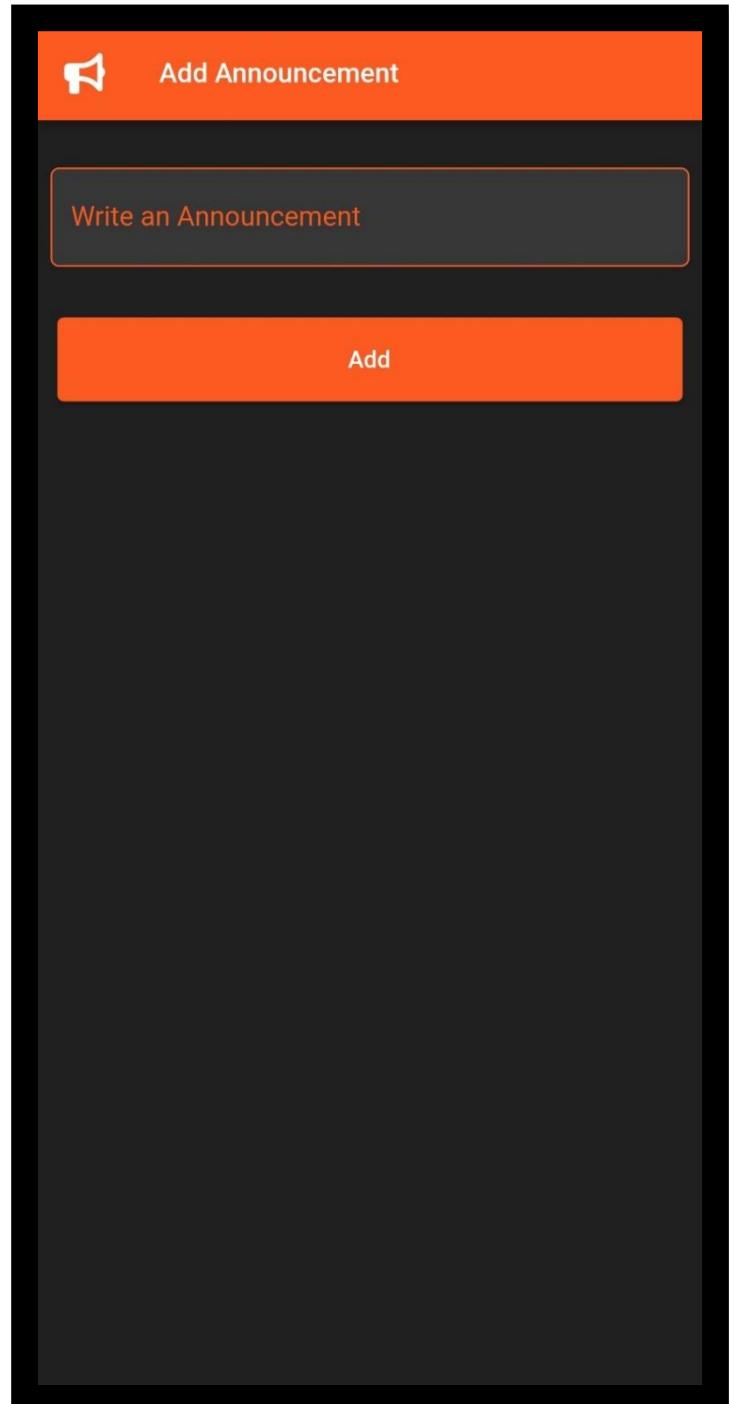


Figure 8.29: Mobile Application Add Income Screen

Figure 8.30: Mobile Application Add Announcement Screen

Chapter 6: Testing

6. Testing and Validation

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps, or missing requirements in contrast to actual requirements.

There are different types of software testing

Functional Testing:

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

Non-Functional Testing:

- Security Testing
- Performance Testing
- Usability Testing
- Compatibility Testing

Unit Testing

- Registration Testing

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_01	Successful Registered	Valid Email, password, password confirm, name, image	Email=Ali.Ahmed15@gmail.com Pass= 123789456 Pass Confirm= 123789456 Name= Ali Ahmed Image exists	Registration Success	Registration Success
TC_02	Invalid Message Appears for email	Invalid Email	Email= Ali.Ahmed.com	Message appears saying invalid email	Message appeared
TC_03	Invalid message appears for password	Password less than 8 characters	Pass = 123456 Pass Confirm = 123456	Message appears saying invalid password	Message appeared
TC_04	Invalid message appears for password and password confirm doesn't match	Password and Password confirm are different	Pass = 123789456 Pass Confirm = 11223344	Message appears saying password doesn't match	Message appeared
TC_05	Invalid Message appears for name	Name is less than 4 characters	Name = Ali	Message appears saying name is short	Message appeared

TC_06	Invalid Message appears for image	No image is uploaded	Image is Empty	Message appears saying image is required	Message Appeared
TC_07	Invalid Message Appears for Empty data	Any data is missing	Any of the 5 data is empty	Message appears saying specific data is required	Message appeared

- **Login Testing**

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_08	Successful Login	Valid Email and password	Email= Ali.Ahmed15@gmail.com Pass= 123789456	Login Success	Login Success
TC_09	Invalid Message Appears for email	incorrect Email	Email= Ali.Ahmed.com	Message appears saying invalid email	Message appeared
TC_10	Invalid message appears for password	Password is incorrect	Pass = 123456789	Message appears saying invalid password	Message appeared

- **Face Scan Testing**

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_11	User is Valid result	Scanned person is a member in gym and has a valid subscription	Scan a person's face that is a member in gym and has a valid subscription	User is Valid	User is Valid
TC_12	User is Invalid	Scanned person is a member in gym, but his subscription is expired	Scan a person's face that is a member in gym, but his subscription is expired	User is invalid	User is invalid
TC_13	User is unknown	Scanned person isn't a member in gym (stranger)	Scan person's face that isn't a member in gym (stranger)	User is unknown	User is unknown

- **Payment Testing**

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_14	Payment Successful	Valid Payment Data	Card number= 5338142814761209 Name=Ali Ahmed Date= 08/25 CVC= 172	Payment is accepted	Payment is accepted
TC_15	Message appears for Invalid Card number	Incorrect Card number	Card number=less than 16 digits Or Card number=more than 16 digits Or Card number= 16 digits but incorrect	Message appears saying invalid payment data	Message appeared
TC_16	Message appears for invalid Date	Incorrect Date	Date= 1325 (out of months range) Or Date = is different than one on card	Message appears saying invalid payment data	Message appeared
TC_17	Message appears for invalid CVC	Incorrect CVC	CVC = more or less than 3 digits Or CVC = incorrect (different than one on card)	Message appears saying invalid payment data	Message appeared

- **Admin Update User (Edit User) Testing**

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_18	Update is successful	Valid Name, Email, Phone, Image	Name = Aya Ahmed Email = Aya.Ahmed15@gmail.com Phone= 01145587362 Image = upload image	Update is successful	User Updated
TC_19	Message appears for Invalid Name	Invalid Name	Name = Aya (Less than 4 characters)	Message appears saying name	Message appeared
TC_20	Message appears for invalid Email	Invalid Email	Email = Aya.Ahemd15@gmail	Message appears saying invalid Email	Message appeared
TC_21	Message appears for phone number	Incorrect phone number	Phone = more or less than 11 digits	Message appears saying invalid phone number	Message appeared

- Chat Testing

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_22	Text Message Sent Successfully	Type a message in text field	Message = Any string in the text field (not empty) Example: Hi, How are you, أَنْتَ	Message is Sent	Message is Sent
TC_23	Media Message (Image, Video) is Sent Successfully	Pick an Image or Video from (Gallery/Camera)	Message = Picked Image/Video From Gallery / Camera	Media is Sent	Media is Sent
TC_24	Message is not Sent because it's Empty	Empty text field (no message)	Message = Empty	Message isn't sent (Send button isn't active)	Message isn't sent (Send button isn't active)
TC_25	Any other file than Image / Video isn't picked	Picking Any other File than Image / Video	Message = Any file other than Image / Video like PDF or Audio	Unsupported Files will not Appear for user to pick	Files didn't appear (only Images / Videos Appeared)

- **BMI Calculator Testing**

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_26	BMI Result is shown	Move Sliders of Input data	Move Height Slider between 80cm to 220cm, Move Weight Slider between 20kg to 180 kg	BMI Result Appears	BMI Result Appeared
TC_27	Testing out of range height and weight	Try to move Sliders of Input out of range	Move Height Slider less than 80cm or more than 220cm, Move Weight Slider less than 20kg or more than 180 kg	Slider won't move out of range	Slider didn't move out of range

- **BMR Calculator Testing**

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_28	BMR Result is shown	Choose valid Input Data	Gender = Choose Male or Female, Height slider between 80cm to 220cm, Weight between 20kg to 180kg, Age between 10 Years to 100 Years,	BMR Result Appears	BMR Result Appeared
TC_29	Message Saying pick a gender	Choose valid data but didn't choose gender	All data are valid but forgot to choose gender	Message appears saying choose a gender	Message Appeared

- Scan QR Testing

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_30	Scan QR of a valid member Result (trainee)	Scanned QR is a member in gym and has valid subscription	Scan QR that is a member in gym and has valid subscription	User is found and show his Subscription expiration date	User is found and show his Subscription expiration date
TC_31	Scan QR of a invalid member Result (trainee)	Scanned QR is a member in gym, but his subscription is expired	Scan QR that is a member in gym, but his subscription is expired	User is found but expired and show his Subscription expiration date	User is found but expired and show his Subscription expiration date
TC_32	Scan QR of User or Trainer (not trainee)	Scanned QR is unsubscribed user	Scan QR that is unsubscribed user	User is found but not subscribed	User is found but not subscribed
TC_33	Scan QR of unknown person	Scanned QR isn't a member in gym (stranger)	Scan QR that isn't a member in gym,	User is not found	User is not found

- **Create Exercise Testing**

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_34	Valid exercise is created	Enter valid data of exercise	Name = isn't empty, Video = isn't empty, Steps =isn't empty, Type = isn't empty	Exercise is created successfully	Exercise is created successfully
TC_35	Empty steps	Press add button to add empty step to the list	Steps = empty	A message saying you must enter at least one step	A message saying you must enter at least one step
TC_36	Invalid picked file	Pick invalid file (not a video)	Video = anything other than a video file	File will not appear to pick and upload (only videos will appear)	File will not appear to pick and upload (only videos will appear)
TC_37	invalid exercise data	Leave one or more of the 4 fields empty	Name = empty or / and Video = empty or /and Steps = empty or / and Type = empty	A message saying please fill all fields (exercise isn't created)	A message saying please fill all fields (exercise isn't created)

- **Create Recipe Testing**

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_38	Valid recipe is created	Enter valid data of recipe	Name = isn't empty, Image = isn't empty, Steps =isn't empty, Ingredients = isn't empty, Nutrition = isn't empty	Recipe is created successfully	Recipe is created successfully
TC_39	Empty steps	Press add button to add empty step to the list	Steps = empty	A message saying you must enter at least one step	A message saying you must enter at least one step
TC_40	Empty nutrition	Press add button to add empty nutrition to the list	Nutrition = empty	A message saying you must enter at least one nutrition	A message saying you must enter at least one nutrition
TC_41	Empty ingredients	Press add button to add empty ingredients to the list	Ingredients = empty	A message saying you must enter at least one ingredient	A message saying you must enter at least one ingredient
TC_42	Invalid picked file	Pick invalid file (not an image)	Image = anything other than an image file	File will not appear to pick and upload (only images will appear)	File will not appear to pick and upload (only images will appear)

TC_43	invalid recipe data	Leave one or more of the 5 fields empty	Name = empty or / and Image = empty or /and Steps = empty or / and Ingredients = empty or / and Nutrition = empty	A message saying please fill all fields (recipe isn't created)	A message saying please fill all fields (recipe isn't created)
-------	---------------------	---	---	--	--

- **Add Income Testing**

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_44	Valid income and price	Enter a valid Income and price	Income = “Daily subscription member” Price = 20	Income is added successfully	Income is added successfully
TC_45	Invalid price	Enter a character in price field	Price = not a number	Not allowed to enter character (number only keyboard appears)	Not allowed to enter character (number only keyboard appears)
TC_46	Empty income And / or price	Press add button with one or more empty fields	Income = empty or / and Price = empty	A message saying all fields must be filled	A message saying all fields must be filled

- **Add Announcement Testing**

Test Case ID	Test Objective	Preconditions	Test Data	Expected Result	Actual Result
TC_47	Valid Announcement	Enter a valid Announcement	Announcement = any string	Announcement is added successfully	Announcement is added successfully
TC_48	Invalid Announcement	Press add button with Announcement field empty	Announcement = empty	A message saying field cannot be empty	A message saying field cannot be empty

Chapter 7: Development Tools and Technologies

7. Development tools and technologies

7.1 Tools and IDEs

- **Visual Studio code IDE**

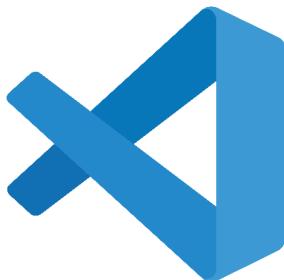


Figure 9.1: VS Code

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

It is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET).

It contains many available extensions and plugins to maximize your productivity.

- **Firebase Cloud Messaging**

Firebase Cloud Messaging (FCM) is a cross-platform messaging solution that lets you reliably send messages at no cost.

Using FCM, you can notify a client app that new email or other data is available to sync. You can send notification messages to drive user re-engagement and retention. For use cases such as instant messaging, a message can transfer a payload of up to 4000 bytes to a client app.

7.2 Languages

- **Dart (Programming Language)**



Figure 9.2: Dart Language

Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.

It is developed by Google and it's the foundation of Flutter.

- **HTML (Markup Language)**



Figure 9.3: HTML

The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

- **CSS (Stylesheet Language)**



Figure 9.4: CSS Language

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

- **Java Script (Programming Language)**



Figure 9.5: JavaScript Language

JavaScript, often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2023, 98.7% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

7.3 Frameworks

- **Flutter**

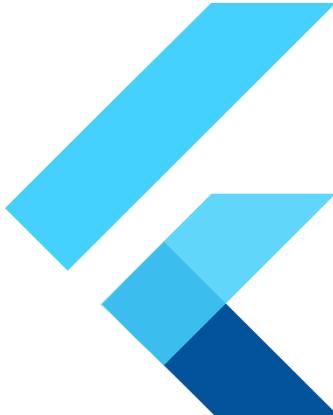


Figure 9.6: Flutter Framework

Flutter is an open-source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase.

It allows developer to build Android, IOS, and Web applications with a single code with dart programming language.

Flutter is:

Fast: Flutter code compiles to ARM or Intel machine code as well as JavaScript, for fast performance on any device.

Productive: Build and iterate quickly with Hot Reload. Update code and see changes almost instantly, without losing state.

Flexible: Control every pixel to create customized, adaptive designs that look and feel great on any screen.

- **NodeJS**



Figure 9.7: Node.Js Framework

Node.js is a cross-platform, open-source server environment that can run on Windows, Linux, Unix, macOS, and more. Node.js is a back-end JavaScript runtime environment, runs on the V8 JavaScript Engine, and executes JavaScript code outside a web browser.

Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The ability to run JavaScript code on the server is often used to generate dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, as opposed to using different languages for the server- versus client-side programming.

7.4 Databases

Firebase



Figure 9.8: Firebase

Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android, and web apps. Firebase provides tools for tracking analytics, reporting, and fixing app crashes, creating marketing and product experiment.

It offers various services to help you to build your application and monitor its performance and stability.

- **Firestore Database**

Firestore is a NoSQL, document-oriented database. Unlike a SQL database, there are no tables or rows. Instead, you store data in *documents*, which are organized into *collections*.

Each *document* contains a set of key-value pairs. Firestore is optimized for storing large collections of small documents.

All documents must be stored in collections. Documents can contain *subcollections* and nested objects, both of which can include primitive fields like strings or complex objects like lists.

Collections and documents are created implicitly in Firestore. Simply assign data to a document within a collection. If either the collection or document does not exist, Firestore creates it.

- **Firebase Cloud Storage**

Cloud Storage for Firebase lets you upload and share user generated content, such as images and video, which allows you to build rich media content into your apps. Your data is stored in a Google Cloud Storage bucket — an exabyte scale object storage solution with high availability and global redundancy. Cloud Storage for Firebase lets you securely upload these files directly from mobile devices and web browsers, handling spotty networks with ease.

- **MongoDB**



Figure 9.9: MongoDB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc.

Chapter 8: Conclusion

8. Conclusion

In Conclusion Gym Management System (GMS) is developed to overcome problems that faces gym owners and improve the management of the gym and provide services to facilitate management for gym owners by preventing strangers to enter the gym with efficient way, to view all members of the gym and track their subscription expiration date, training for members of the gym especially the people who are new to the gym, and trainers by making them appears to users so that they can promote themselves.

It also provides a way of communicating between trainer and trainee to make it easier for trainer to keep track of their trainees and guide them if they wanted any help.
