1.1 Introduction

The Gym Club App is an Android application designed to help gym club members manage their fitness routine and track their progress. The app provides users with a range of features that can be used to set fitness goals, track workouts, and monitor progress over time. The app allows users to create personalized workout plans by selecting specific exercises and setting goals for each one.

Users can also track their progress over time by logging the number of sets and repetitions completed, as well as the weight lifted for each exercise. In addition to tracking workouts, the app also includes features for tracking nutrition and setting reminders for upcoming workouts.

Users can input their daily food intake and track their macronutrient and micronutrient intake over time. The app also allows users to set reminders for upcoming workouts, ensuring that they stay on track with their fitness goals. The Gym Club App is designed with a clean and user-friendly interface that is easy to navigate. Users can quickly access their workout plans, track their progress, and set reminders using the app's intuitive menu system.

Overall, the Gym Club App is a powerful tool for gym club members who want to take control of their fitness routine and achieve their goals. With its range of features and user-friendly interface, the app is an essential tool for anyone looking to improve their fitness and overall health.

1.2 Abstract:

The Gym Club App is an Android application designed to help gym club members manage their fitness routine and track their progress. The app provides users with a range of features that can be used to set fitness goals, track workouts, and monitor progress over time. The app allows users to create personalized workout plans by selecting specific exercises and setting goals for each one.

Users can also track their progress over time by logging the number of sets and repetitions completed, as well as the weight lifted for each exercise. In addition to tracking workouts, the app also includes features for tracking nutrition and setting reminders for upcoming workouts.

Users can input their daily food intake and track their macronutrient and micronutrient intake over time. The app also allows users to set reminders for upcoming workouts, ensuring that they stay on track with their fitness goals. The Gym Club App is designed with a clean and user-friendly interface that is easy to navigate. Users can quickly access their workout plans, track their progress, and set reminders using the app's intuitive menu system.

The committed to providing high-quality software products that meet the needs of its clients and users. Overall, the Gym Club App is a powerful tool for gym club members who want to take control of their fitness routine and achieve their goals. With its range of features and user-friendly interface, the app is an essential tool for anyone looking to improve their fitness and overall health.

1.3 Existing System:

Currently, many gym club members track their fitness routines and progress manually using pen and paper or spreadsheets. This process can be time-consuming, inefficient, and prone to errors.

Without a centralized system, gym club members may also struggle to track their progress over time and adjust their workout plans and nutrition goals.

Need for System:

The Gym Club App was developed to address the limitations of the existing system and provide gym club members with a more efficient and effective way to manage their fitness routines and track their progress. The app provides users with a centralized system for tracking their workouts, nutrition, and progress over time. By using the Gym Club App, gym club members can easily create personalized workout plans and track their progress by logging the number of sets, repetitions, and weight lifted for each exercise.

The app also includes features for tracking nutrition, allowing users to input their daily food intake and track their macronutrient and micronutrient intake over time. In addition, the Gym Club App allows users to set reminders for upcoming workouts, ensuring that they stay on track with their fitness goals. The app's user-friendly interface makes it easy for users to access their workout plans, track their progress, and set reminders using the app's intuitive menu system.

Overall, the need for the Gym Club App is clear: to provide gym club members with a more efficient and effective way to manage their fitness routines and track their progress. By using the app, gym club members can achieve their fitness goals more easily and accurately, leading to better health and overall well-being.

1.4 Scope of System:

The Gym Club App is a comprehensive Android application designed to provide gym club members with a range of tools to manage their fitness routines and track their progress. The app includes the following key features:

Personalized workout plans: Users can create personalized workout plans by selecting specific exercises and setting goals for each one.

Workout tracking: Users can track their progress over time by logging the number of sets and repetitions completed, as well as the weight lifted for each exercise.

Nutrition tracking: Users can input their daily food intake and track their macronutrient and micronutrient intake over time.

Reminder system: Users can set reminders for upcoming workouts, ensuring that they stay on track with their fitness goals.

User-friendly interface: The app is designed with a clean and user-friendly interface that is easy to navigate.

Data visualization: The app provides users with graphs and charts to visualize their progress over time. The Gym Club App is intended for gym club members who want to take control of their fitness routine and achieve their goals. The app is suitable for users of all fitness levels, from beginners to advanced athletes.

The scope of the system includes the development of the Android application, as well as the backend infrastructure required to support the app's functionality. The system will be developed using the latest technologies and best practices in mobile app development to ensure that it is fast, responsive, and reliable.

Overall, the Gym Club App is a comprehensive tool for gym club members who want to improve their fitness and overall health. With its range of features and user-friendly interface, the app is an asset for anyone looking to take control of their fitness routine and achieve their goals.

1.5 Operating Environment – Hardware and Software

Hardware Requirements:

Android Version - 5.0 Lollipop or later
Operating System - Android
RAM 4GB or More

Software Requirements:

Front End - XML
Developing Tools - Android Studio
Back End - Java
Database - Firebase

2 Proposed System:

The proposed Gym Club App is an Android application that will provide gym club members with a range of tools to manage their fitness routines and track their progress. The app will be developed using the latest technologies and best practices in mobile app development to ensure that it is fast, responsive, and reliable.

Key Features of Proposed System:

Personalized workout plans: The app will allow users to create personalized workout plans by selecting specific exercises and setting goals for each one.

tracking: Users will be able to track their progress over time by logging the number of sets and repetitions completed, as well as the weight lifted for each exercise.

Nutrition tracking: The app will allow users to input their daily food intake and track their macronutrient and micronutrient intake over time.

Reminder system: Users will be able to set reminders for upcoming workouts, ensuring that they stay on track with their fitness goals.

User-friendly interface: The app will be designed with a clean and user-friendly interface that is easy to navigate.

Data visualization: The app will provide users with graphs and charts to visualize their progress over time.

Technical Features: The proposed system will be developed using Android Studio, Java programming language, and SQLite database. The app will be compatible with Android devices running Android OS version 5.0 and higher. The app will be designed with a responsive layout, optimized for different screen sizes and orientations.

Security Features: The app will be secured with user authentication, ensuring that only authorized users can access their data. The app will also encrypt user data using standard encryption algorithms to protect user privacy.

2.2 Feasibility Study:

The feasibility study is a critical aspect of any software development project, as it helps to determine whether the proposed system is viable and sustainable. The feasibility study for the Gym Club App includes the following key components:

Technical Feasibility: The Gym Club App is technically feasible, as it will be developed using well-established technologies and best practices in mobile app development. Android Studio and Java programming language will be used to develop the app, while SQLite will be used as the backend database. These technologies are widely used and have a vast developer community, ensuring that the app can be maintained and updated over time.

Operational Feasibility: The Gym Club App is operationally feasible, as it will be designed with a user-friendly interface that is easy to navigate. The app's features are also straightforward and easy to understand, ensuring that gym club members can quickly start using the app without any training.

Economic Feasibility: The Gym Club App is economically feasible, as the development costs are reasonable, and the app's potential benefits far outweigh the costs. The app will be developed by a team of experienced developers and designers, ensuring that the app is of high quality and meets user requirements.

Legal and Ethical Feasibility: The Gym Club App is legal and ethical, as it complies with all relevant laws and regulations, including data protection and privacy laws. The app will also be developed with the user's privacy in mind, ensuring that user data is encrypted and always secured.

Schedule Feasibility: The Gym Club App is schedule feasible, as the development team has a clear understanding of the project requirements and a well-defined development plan. The project timeline will be managed using Agile methodologies, ensuring that the project is delivered on time and within budget.

2.3 Objectives of Proposed System:

The main objectives of the proposed Gym Club App are:
To provide gym club members with a comprehensive tool for managing their fitness routines and tracking their progress.
To provide gym club members with a personalized workout plan that is tailored to their fitness goals and preferences.
To help gym club members track their progress over time, including the number of sets and repetitions completed and the weight lifted for each exercise.
To help gym club members track their daily food intake and macronutrient and micronutrient intake to ensure they are meeting their nutritional needs.
To provide gym club members with a reminder system that helps them stay on track with their fitness routine.
To provide gym club members with a user-friendly interface that is easy to navigate and understand.
To provide gym club members with data visualization tools that help them visualize their progress over time.
To ensure the app is secure, with user authentication and data encryption to protect user privacy.
To ensure the app is compatible with Android devices running Android OS version 5.0 and higher.
To ensure the app is developed using the latest technologies and best practices in mobile app development to ensure it is fast, responsive, and reliable.
In summary, the proposed Gym Club App aims to provide a comprehensive solution to gym club members, helping them manage their fitness routines, track their progress, and achieve their fitness goals. The app's user-friendly interface, personalized workout plans, and reminder system make it an invaluable tool for anyone looking to improve their fitness and overall health.

2.4 Users of system

1 Admin (Trainer)
2 User (Customer)

3 Analysis and Design

3.1 System Requirements (Functional and Non-Functional)

System Requirements: The Gym Club App Android is an application developed for gym enthusiasts who want to keep track of their fitness routine and progress. The application provides an easy-to-use interface for managing various activities such as workout routines, diet plans, and fitness goals.

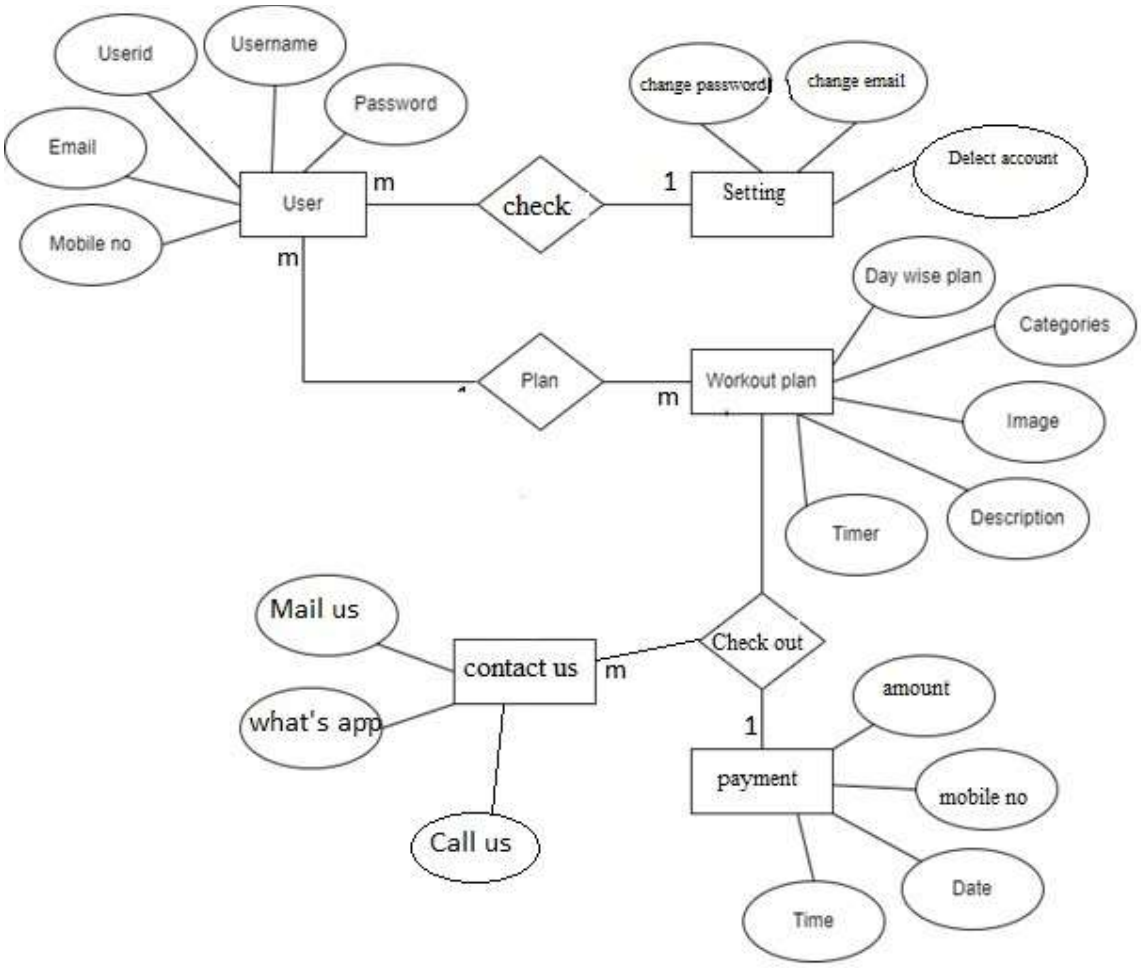
Functional Requirements:

1. User Registration and Login: Users can register and create an account using their email ID or social media profiles. They can then log in to the app using the same credentials.
2. Personal Profile: Users can create their personal profile with their basic information such as name, age, gender, weight, and height.
3. Workout Plans: Users can access pre-built workout plans or create their own workout plans based on their fitness goals. They can track their progress and view their history of completed workouts.
4. Exercise Tracking: Users can track their exercises using the app. They can log the number of sets, reps, and weights lifted for each exercise.
5. Diet Plans: Users can create their diet plans or access pre-built diet plans that suit their fitness goals. They can track their daily calorie intake and monitor their progress.

Non-functional Requirements:

1. User Interface: The user interface should be intuitive, user-friendly, and visually appealing.
2. Performance: The app should be responsive and fast. It should load quickly and perform all actions without any lag.
3. Security: The app should be secure and protect user data. User data should be encrypted, and access should be restricted to authorized users.
4. Compatibility: The app should be compatible with a wide range of Android devices and versions.
5. Availability: The app should be available on the Google Play Store and should be easily downloadable and installable.

3.2 Entity Relationship Diagram



3.3 Table Structure

User

Sr.No	Field	Data Type	Constraints	Description
1	Userid	Varchar	Primary key	UserID
2	Username	Varchar	Not null	Username
3	User_Pin	INT	Primary key	Userpin
4	User_no	INT	Primary key	User number
5	User_DOB	Date	Not null	User date of birth
6	User_Email	Varchar	Not null	User email
7	Gender	Varchar	Not null	User gender

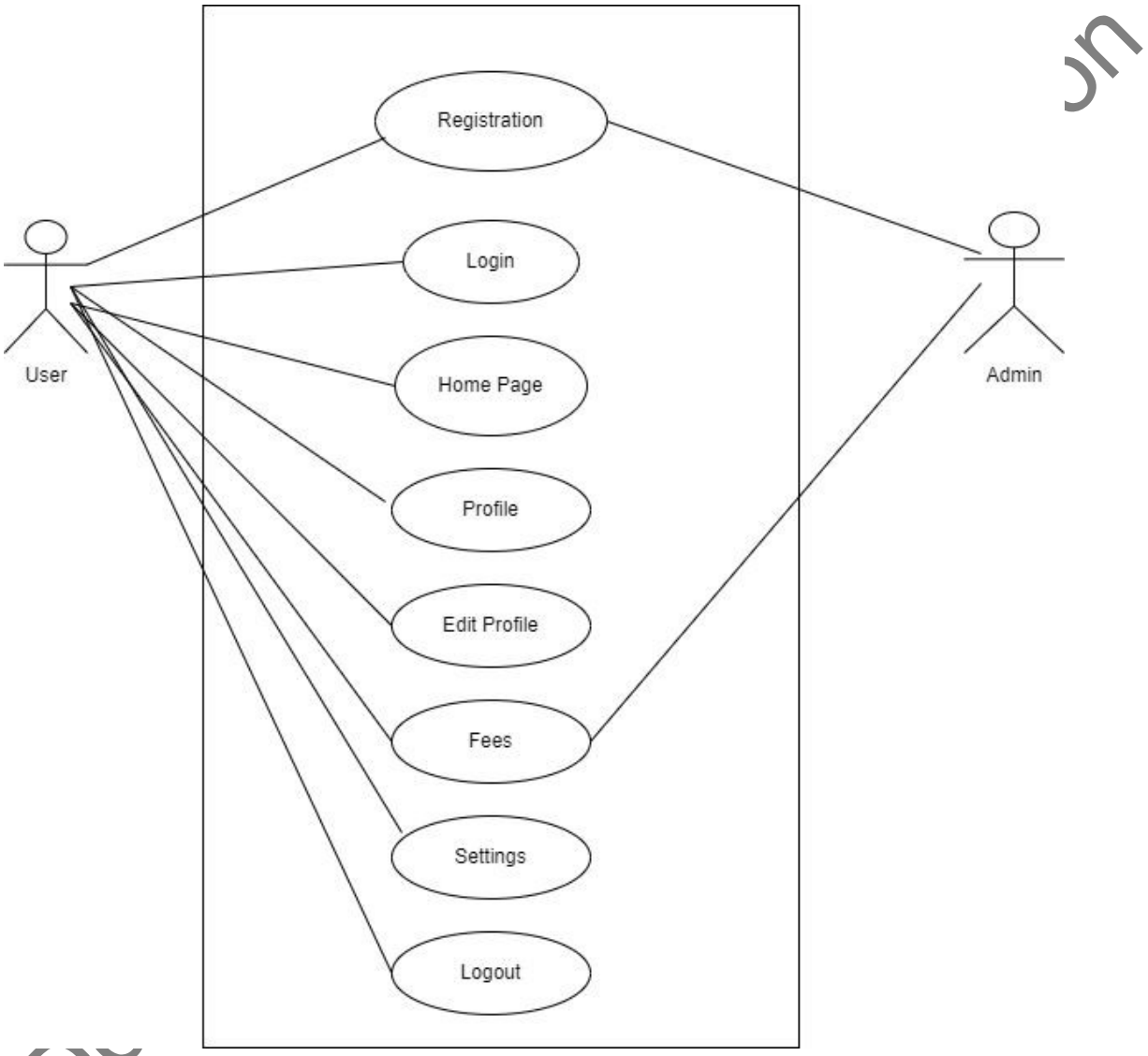
Setting

Sr.No	Field	Data Type	Constraints	Description
1	Change password	Varchar	Not null	Change password
2	Change email	Varchar	Not null	Change email
3	Delete account	INT	Not null	Delete account

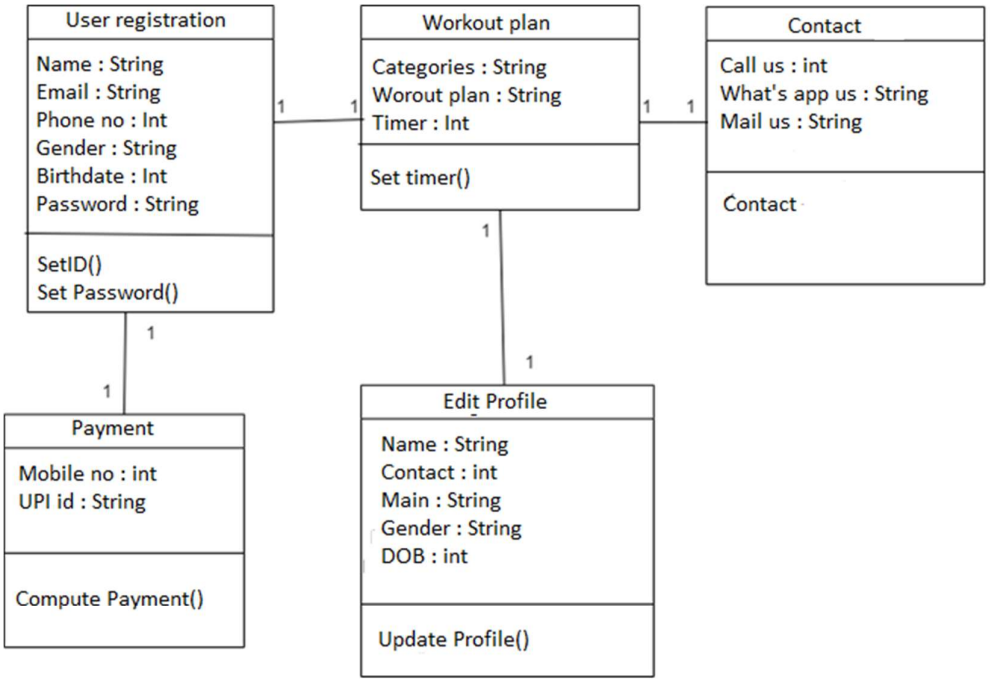
Edit Profile

Sr.No	Field	Data Type	Constraints	Description
1	Categories	Varchar	Not null	Workout_categories
2	Plan	Varchar	Not null	Workout_plan
3	Timer	INT	Primary key	Workout_time
4	Name	Varchar	Not null	Change name
5	Place	Varchar	Not null	Change location
6	Duration	Varchar	Not null	Change duration
7	Mobile no	INT	Not null	Change number
8	User DOB	INT	Not null	Change DOB

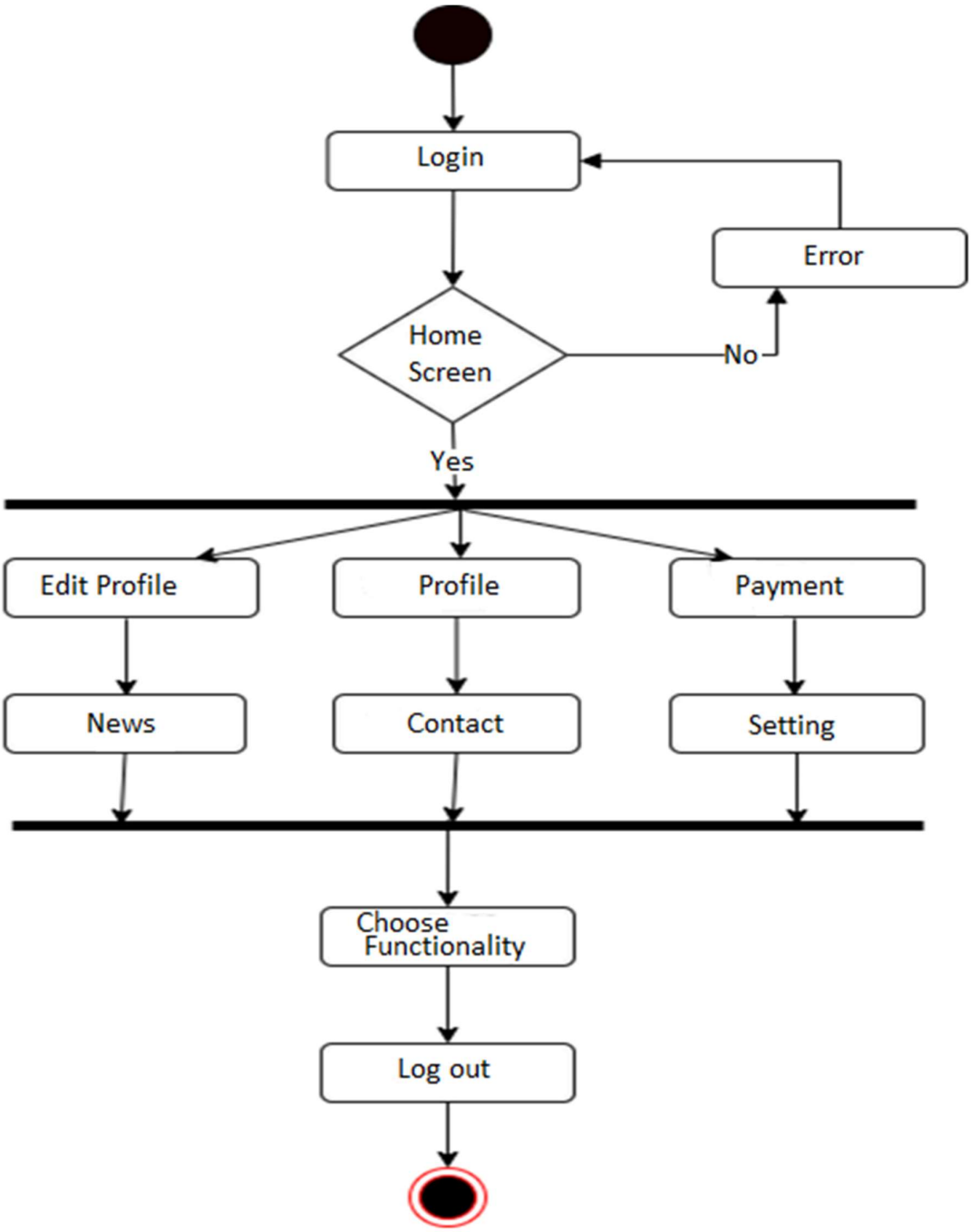
3.4 Use Case Diagram



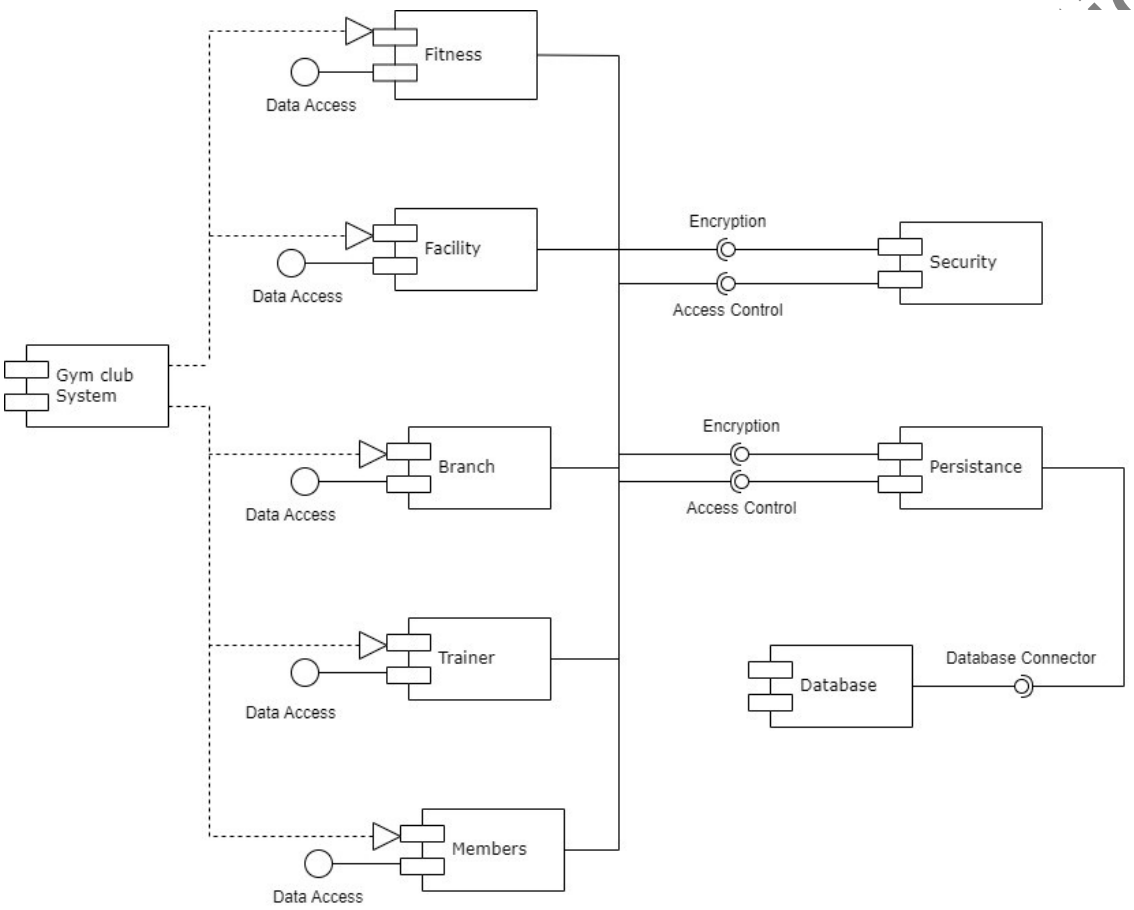
3.5 Class Diagram



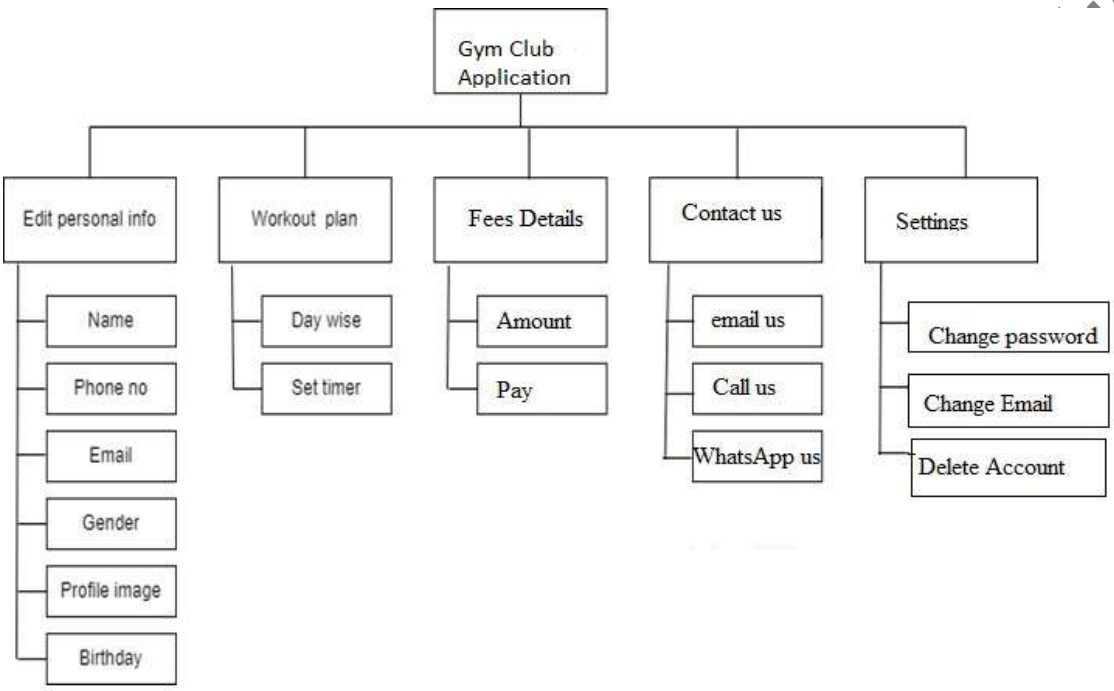
3.6 Activity Diagram



3.7 Deployment Diagram

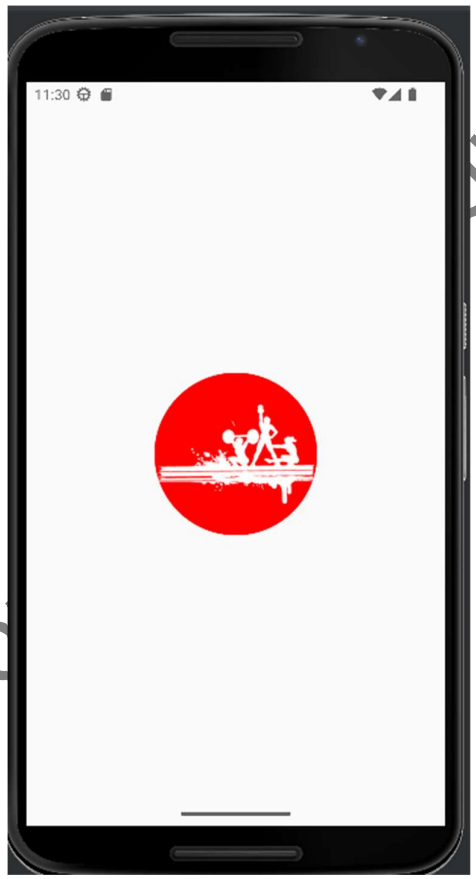


3.8 Module Hierarchy Diagram



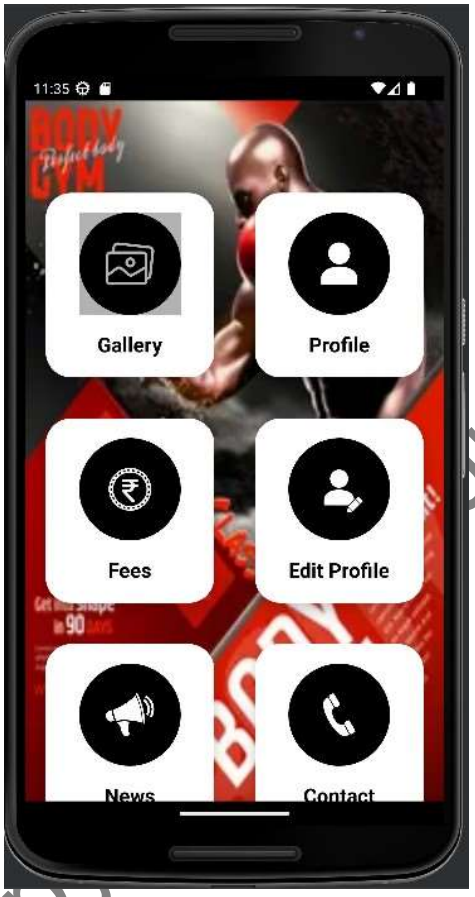
3.9 Sample Input and Output Screens

App Logo

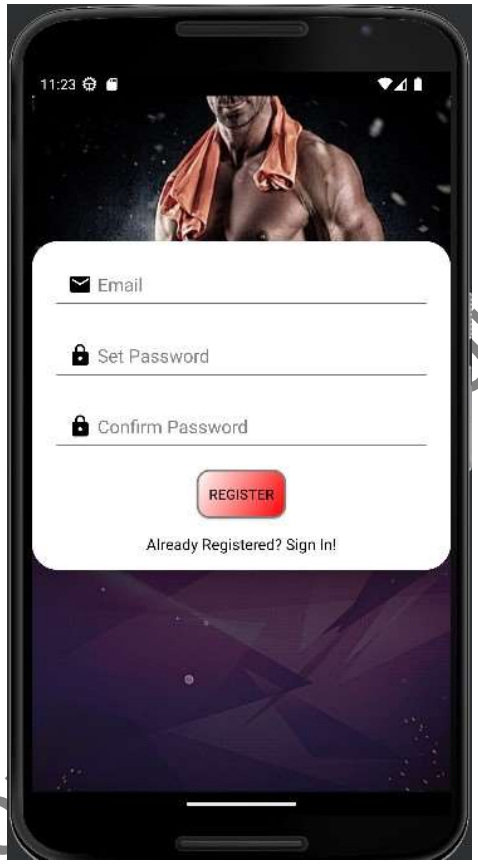


Welcome page

Home page

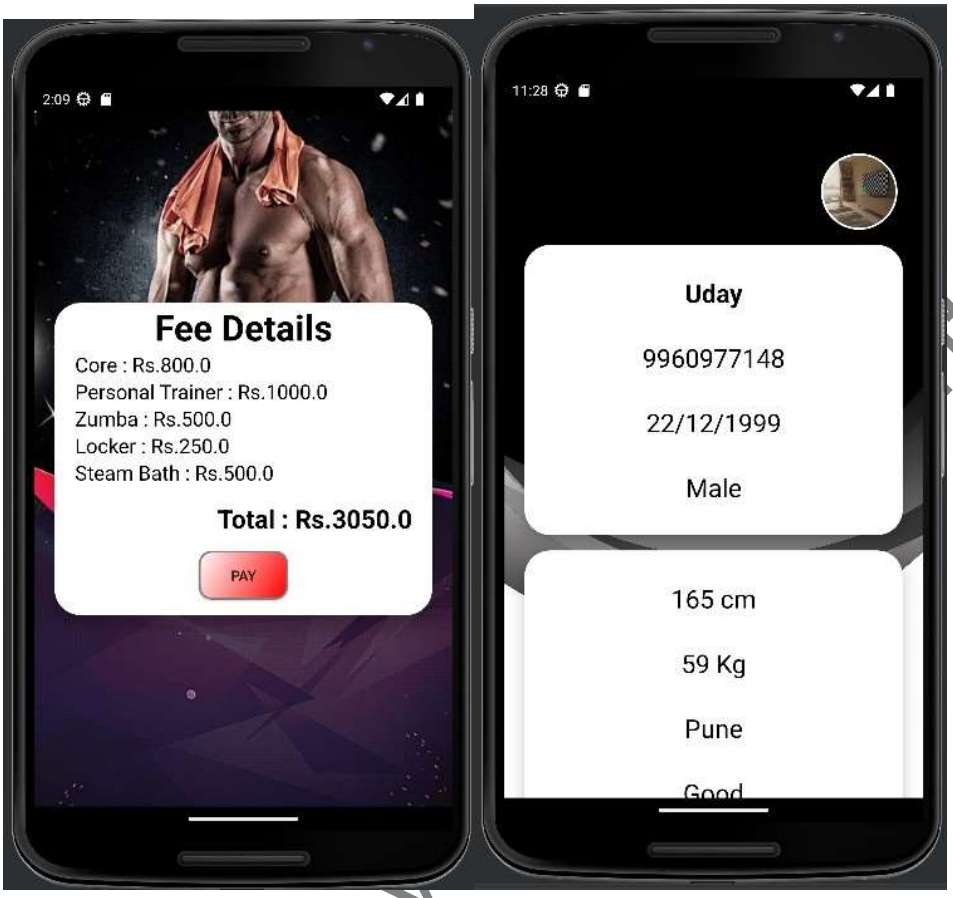


Registration page

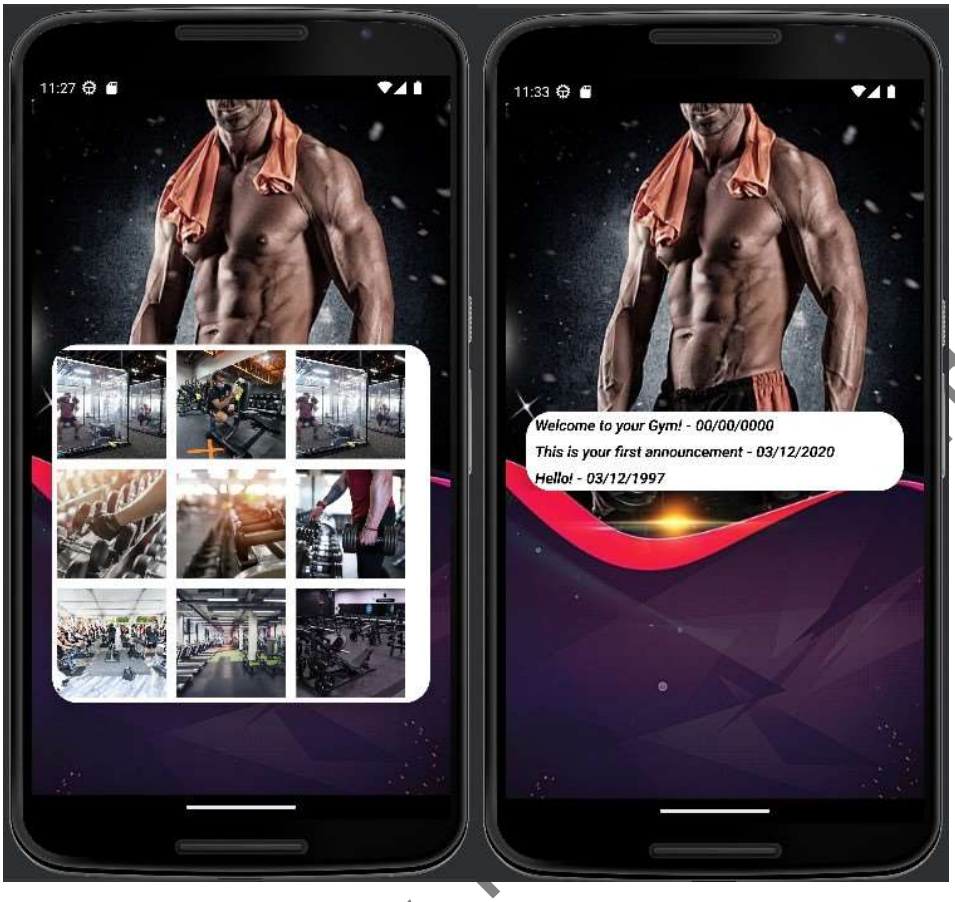


Login page

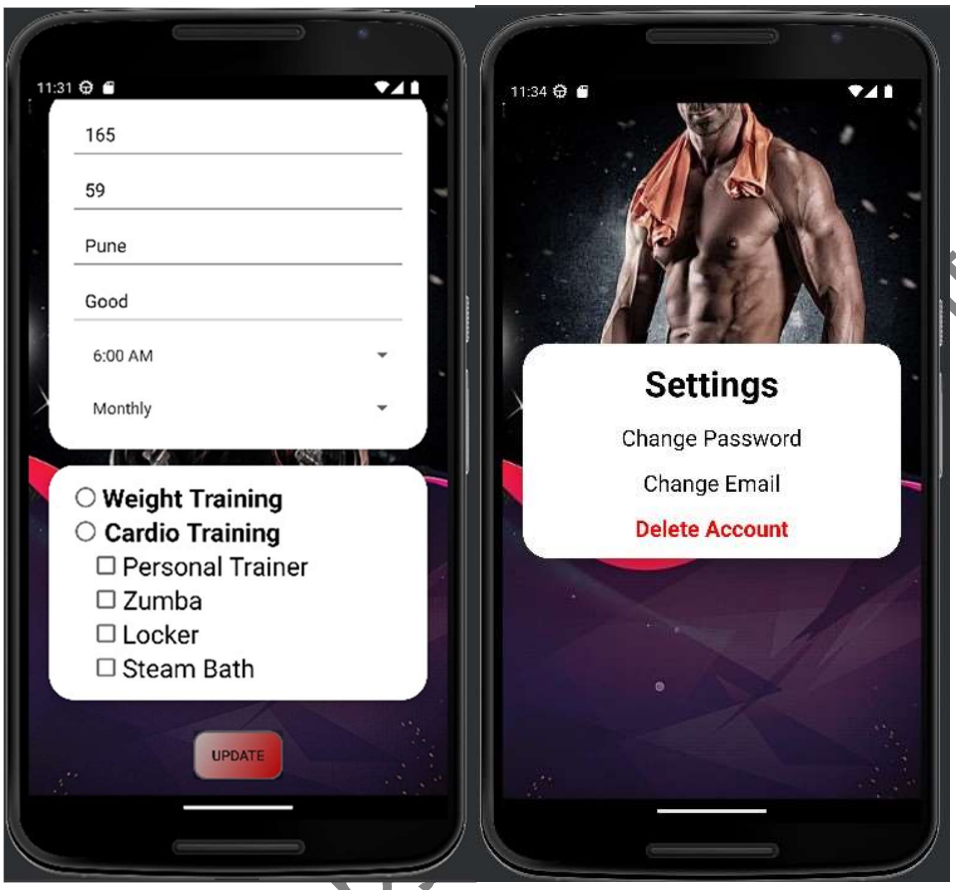
Profile page **Payment page**



Gallery page **News Page**



Profile Update page **Setting page**



4 Coding
4.1 Code Snippets

Add Profile Activity:

```
package com.rohansingh.gymmanagement;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.Spinner;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
```

```
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.io.IOException;

import de.hdodenhof.circleimageview.CircleImageView;

public class AddProfileActivity extends AppCompatActivity
implements AdapterView.OnItemSelectedListener {

   EditText
etAddName,etAddMobile,etAddDateOfBirth,etHeight,etWeight,etCit
y,etMedicalCondition;
   Button btAddProfile;
   Spinner spGender,spTimings,spPackages;
   CircleImageView ivProfilePic;
   String
spGenderLabel,userName,userDob,userMobile,spTimingsLabel,spPac
kageLabel,userHeight,userWeight,userCity,userMedicalCondition,use
rCore,userPersonalTrainer="No", userZumba="No", userLocker="No",
userSteamBath="No";
   FirebaseAuth fbAuth;
   CheckBox cbPersonalTrainer, cbZumba, cbLocker, cbSteamBath;
   RadioButton rbWeightTraining, rbCardioTraining;

   private static final int PICK_IMAGE = 123;
   Uri imagePath;

   private StorageReference storageReference;

   @Override
   protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
```

```
        if (requestCode == PICK_IMAGE && resultCode == RESULT_OK &&
data.getData() != null)
            imagePath = data.getData();
        try {
            Bitmap                          bitmap                          =
MediaStore.Images.Media.getBitmap(getContentResolver(),imagePa
th);
            ivProfilePic.setImageBitmap(bitmap);
        } catch (IOException e) {
            e.printStackTrace();
            super.onActivityResult(requestCode, resultCode, data);
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_profile);
        setupUI();
        fbAuth = FirebaseAuth.getInstance();
        FirebaseStorage firebaseStorage = FirebaseStorage.getInstance();
        storageReference = firebaseStorage.getReference();

        if(spGender!=null){
            spGender.setOnItemSelectedListener(this);
        }
        ArrayAdapter<CharSequence>              adapter                 =
ArrayAdapter.createFromResource(this,R.array.gender,android.R.lay
out.simple_spinner_item);

adapter.setDropDownViewResource(android.R.layout.simple_spinne
r_dropdown_item);
        if(spGender!=null){
            spGender.setAdapter(adapter);
        }
```

```
        if(spTimings!=null){
            spTimings.setOnItemSelectedListener(this);
        }
        ArrayAdapter<CharSequence> adTimings =
ArrayAdapter.createFromResource(this,R.array.timings,android.R.lay
out.simple_spinner_item);

adTimings.setDropDownViewResource(android.R.layout.simple_spin
ner_dropdown_item);
        if(spTimings!=null){
            spTimings.setAdapter(adTimings);
        }

        if(spPackages!=null){
            spPackages.setOnItemSelectedListener(this);
        }
        ArrayAdapter<CharSequence> adPackages =
ArrayAdapter.createFromResource(this,R.array.packages,android.R.l
ayout.simple_spinner_item);

adPackages.setDropDownViewResource(android.R.layout.simple_spi
nner_dropdown_item);
        if(spPackages!=null){
            spPackages.setAdapter(adPackages);
        }

        ivProfilePic.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent();
                intent.setType("image/*");
                intent.setAction(Intent.ACTION_GET_CONTENT);
                startActivityForResult(Intent.createChooser(intent,"Select
Image"),PICK_IMAGE);
```

```
        }
    });

    btAddProfile.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            userName = etAddName.getText().toString();
            userMobile = etAddMobile.getText().toString();
            userDob = etAddDateOfBirth.getText().toString();
            userHeight = etHeight.getText().toString();
            userWeight = etWeight.getText().toString();
            userCity = etCity.getText().toString();
            userMedicalCondition =
etMedicalCondition.getText().toString();
            if(imagePath==null){
                Toast.makeText(AddProfileActivity.this, "Add profile Pic!",
Toast.LENGTH_SHORT).show();
            } else {
                sendUserProfile();
            }
        }
    });
  }

  private void setupUI(){
    etAddName = findViewById(R.id.etAddName);
    etAddMobile = findViewById(R.id.etAddMobile);
    etAddDateOfBirth = findViewById(R.id.etAddDateOfBirth);
    btAddProfile = findViewById(R.id.btAddProfile);
    spGender = findViewById(R.id.spGender);
    ivProfilePic = findViewById(R.id.ivProfilePic);
    rbWeightTraining = findViewById(R.id.rbWeightTraining);
    rbCardioTraining = findViewById(R.id.rbCardioTraining);
    cbPersonalTrainer = findViewById(R.id.cbPersonalTrainer);
    cbZumba = findViewById(R.id.cbZumba);
```

```
        cbLocker = findViewById(R.id.cbLocker);
        cbSteamBath = findViewById(R.id.cbSteamBath);
        etHeight = findViewById(R.id.etHeight);
        etWeight = findViewById(R.id.etWeight);
        etCity = findViewById(R.id.etCity);
        etMedicalCondition = findViewById(R.id.etMedicalCondition);
        spTimings = findViewById(R.id.spTimings);
        spPackages = findViewById(R.id.spPackage);
    }

    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {
        Spinner spOne = (Spinner) parent;
        Spinner spTwo = (Spinner) parent;
        Spinner spThree = (Spinner) parent;
        if(spOne.getId()==R.id.spGender){
            spGenderLabel                                              =
parent.getItemAtPosition(position).toString();
        } else if(spTwo.getId()==R.id.spTimings){
            spTimingsLabel                                             =
parent.getItemAtPosition(position).toString();
        } else if(spThree.getId()==R.id.spPackage){
            spPackageLabel                                             =
parent.getItemAtPosition(position).toString();
        }
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }

    @SuppressLint("NonConstantResourceId")
    public void onCheckBoxClicked(View view) {
```

```
    Boolean checked = ((CheckBox) view).isChecked();
    switch (view.getId()) {
        case R.id.cbPersonalTrainer:
            if (checked) {
                userPersonalTrainer = "Yes";
            } else {
                userPersonalTrainer = "No";
            }
            break;
        case R.id.cbZumba:
            if (checked) {
                userZumba = "Yes";
            } else {
                userZumba = "No";
            }
            break;
        case R.id.cbLocker:
            if (checked) {
                userLocker = "Yes";
            } else {
                userLocker = "No";
            }
            break;
        case R.id.cbSteamBath:
            if (checked) {
                userSteamBath = "Yes";
            } else {
                userSteamBath = "No";
            }
            break;
        default:
            break;
    }
}
```

```
    private void sendUserProfile(){
        FirebaseDatabase fbDatabase = FirebaseDatabase.getInstance();
        DatabaseReference myRef
=fbDatabase.getReference(fbAuth.getUid());
        updateUserProfilePic();
        UserProfile userProfile = new
UserProfile(userName,userMobile,userDob,spGenderLabel,userHeig
ht,userWeight,userCity,userMedicalCondition,spTimingsLabel,spPack
ageLabel,userCore,userPersonalTrainer,userZumba,userLocker,userS
teamBath);

myRef.child("Profile").setValue(userProfile).addOnCompleteListener(
new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if(task.isSuccessful()){
                    Toast.makeText(AddProfileActivity.this, "Successful",
Toast.LENGTH_SHORT).show();
                    finish();
                } else {
                    Toast.makeText(AddProfileActivity.this, "Failed",
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    private void updateUserProfilePic(){
        StorageReference imageReference =
storageReference.child("Images").child("ProfilePic").child(fbAuth.get
Uid());
        UploadTask uploadTask = imageReference.putFile(imagePath);
        uploadTask.addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
```

```
            Toast.makeText(AddProfileActivity.this, "File Upload Failed",
Toast.LENGTH_SHORT).show();
          }
        }).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
          @Override
          public void onSuccess(UploadTask.TaskSnapshot taskSnapshot)
{
            Toast.makeText(AddProfileActivity.this,  "File  Successfully
Uploaded!", Toast.LENGTH_SHORT).show();
          }
        });
    }

    public void onRadioButtonClicked(View view) {
        boolean checked = ((RadioButton)view).isChecked();
        switch(view.getId()){
            case R.id.rbWeightTraining :
                if(checked){
                    userCore=getString(R.string.weight_training);
                }
                break;
            case R.id.rbCardioTraining :
                if (checked){
                    userCore=getString(R.string.cardio_training);
                }
                break;
            default:
                break;
        }
    }
}
```

5 Testing

5.1 Test Strategy:

The Gym Club App is a complex system that requires rigorous testing to ensure that it meets the requirements and works reliably across a range of devices and operating systems. The following test strategy will be employed to ensure that the app is thoroughly tested and meets the required quality standards.

Unit Testing:

Unit testing will be conducted to ensure that each individual component of the app works as expected. This includes testing individual functions and classes, as well as ensuring that the user interface is responsive and intuitive.

Integration Testing:

Integration testing will be conducted to ensure that all the components of the app work together as expected. This includes testing the communication between the front-end and back-end systems and ensuring that data is being transferred correctly.

Compatibility Testing:

Compatibility testing will be conducted to ensure that the app works reliably across a range of devices and operating systems. The app will be tested on various Android devices running different versions of the Android OS to ensure that it works reliably on all devices.

Performance Testing:

Performance testing will be conducted to ensure that the app is fast, responsive, and reliable. This includes testing the app's load times, response times, and the app's ability to handle multiple user requests simultaneously.

Security Testing:

Security testing will be conducted to ensure that the app is secure and user data is protected. This includes testing the app's user authentication system, data encryption, and ensuring that user data is not accessible to unauthorized parties.

User Acceptance Testing:

User acceptance testing will be conducted to ensure that the app meets user requirements and is easy to use. This includes testing the app with actual gym club members to gather feedback and identify areas for improvement.

5.2 Unit Test Plan:

The Gym Club App is a complex system that requires thorough testing at the unit level to ensure that each individual component of the app works as expected. The following is a sample unit test plan for the app:

User Interface Testing:

a. Verify that all buttons, text fields, and other UI elements are functional and responsive.
b. Verify that the UI elements are arranged logically and are easy to use.
c. Verify that the UI elements are consistent across different screens.

Functionality Testing:

a. Verify that each function of the app works as expected.
b. Verify that all user inputs are validated and sanitized.
c. Verify that all error messages are displayed correctly.

Database Testing:

a. Verify that data is being stored correctly in the database.
b. Verify that data can be retrieved from the database correctly.
c. Verify that the database is updated correctly when data is changed.

Network Testing:

a. Verify that the app can communicate with the server correctly.
b. Verify that data is being transferred correctly between the app and the server.
c. Verify that error messages are displayed correctly when network errors occur.

Performance Testing:

a. Verify that the app is fast and responsive.
b. Verify that the app can handle multiple user requests simultaneously.
c. Verify that the app does not crash or hang during use.

Security Testing:

a. Verify that the app's authentication system is working correctly.
b. Verify that user data is encrypted correctly.
c. Verify that user data is not accessible to unauthorized parties.

5.4 Test Case

Test Case ID	Test Case Description	Test Steps	Expected Results	Actual Results	Pass/Fail
TC001	Login Functionality	1. Open the app 2. Click on the login button 3. Enter valid credentials	User should be logged in successfully and redirected to the home screen	User is redirected to home page after successful login	Pass
TC002	Registration Functionality	1. Open the app 2. Click on the registration button 3. Enter valid details	User should be registered successfully and redirected to the home screen	User is redirected to login page after successful registration	Pass
TC003	Home Screen Display	1. Login to the app 2. Verify the home screen displays all necessary information	Home screen should display user's membership details, upcoming classes, and recommended workouts	User profile details are updated successfully	Pass
TC004	Book Gym plan	1. Login to the app 2. Click on the classes tab 3. Select an upcoming class 4. Book the class	User should receive a confirmation message and the class should be added to their schedule	User received a confirmation message and the class was added to their schedule	Pass

Test Case ID	Test Case Description	Test Steps	Expected Results	Actual Results	Pass/Fail
TC005	Cancel a Gym Plan	1. Login to the app 2. Click on the profile tab 3. Select a booked class 4. Cancel the class	User should receive a confirmation message and the class should be removed from their schedule	User received a confirmation message and the class was removed from their schedule	Pass
TC006	View Workout Plans	1. Login to the app 2. Click on the workouts tab 3. Select a workout plan	User should be able to view the details of the workout plan, including exercises, sets, and reps	User was able to view the workout plan details as expected	Pass
TC008	Logout Functionality	1. Login to the app 2. Click on the logout button	User should be logged out successfully and redirected to the login screen	User was logged out successfully and redirected to the login screen	Pass

5.5 Defect Report:

A defect report is used to track and document any issues or defects found during testing. The following is a sample defect report for the Gym Club App:

Defect ID: GC-001
Defect Name: Incorrect email validation message
Defect Description: When a user enters an incorrect email address during registration, the validation message displayed is "Invalid Password" instead of "Invalid Email Address".

Steps to Reproduce:
Open the Gym Club App.
Click on the "Register" button.
Enter an invalid email address (e.g., "test@.com") in the email address field.
Enter a valid password in the password field.
Click on the "Register" button.
Verify that the validation message displayed is "Invalid Password" instead of "Invalid Email Address".

Expected Result:
The app should validate the user's email address correctly.
The validation message should display "Invalid Email Address" when an incorrect email address is entered.

Actual Result:
The app validates the user's email address incorrectly.
The validation message displays "Invalid Password" when an incorrect email address is entered.
Severity: Medium

Priority: High
Status: Open
Assigned To: Developer X

6 Limitations of Proposed System:

Despite the many benefits and features of the proposed Gym Club App, there are also some limitations that need to be considered. These include:

Hardware Limitations: The app requires a smartphone or tablet to run, which means that users without access to this hardware will not be able to use the app.

Internet Connectivity: The app requires an internet connection to work correctly. Users without an internet connection will not be able to access the app's features.

Platform Compatibility: The app is designed for Android devices only, which means that iOS users will not be able to use the app.

Security: The app stores user data, including personal and financial information. This means that there is a risk of data breaches, and the app must be designed with security in mind to prevent these types of issues.

User Adoption: Even if the app is well-designed and functional, there is no guarantee that users will adopt it. The app must be marketed effectively to reach its target audience and encourage users to use it regularly.

App Maintenance: The app will require ongoing maintenance and updates to ensure that it continues to work correctly and meets user needs. This can be costly and time-consuming for the development team.

7 Proposed Enhancements:

The following are some proposed enhancements for the Gym Club App that can improve its functionality and user experience:

Integration with Wearable Devices: The app could be integrated with wearable devices such as fitness trackers and smartwatches, allowing users to track their fitness progress in real-time.

Meal Tracking: A meal tracking feature could be added to the app, allowing users to track their daily caloric intake and monitor their diet more effectively.

Group Fitness Classes: The app could offer group fitness class scheduling and booking, allowing users to sign up for fitness classes and reserve spots in advance.

Social Sharing: The app could be integrated with social media platforms, allowing users to share their fitness progress with friends and family.
Personalized Recommendations: The app could use machine learning algorithms to provide personalized fitness recommendations based on user data and fitness goals.

Virtual Training: The app could offer virtual personal training sessions, allowing users to work with a certified personal trainer from the comfort of their own homes.

8 Conclusion

In conclusion, the proposed Gym Club App is a valuable tool for individuals looking to achieve their fitness goals. The app offers a range of features, including workout tracking, progress monitoring, and personalized workout plans. The app is designed to be user-friendly, with an intuitive interface and easy-to-use features. The feasibility study indicates that the app is financially viable and has the potential to generate revenue through a variety of monetization strategies.

The proposed system is expected to meet the needs of the target audience and provide significant benefits in terms of convenience and effectiveness. The test strategy and plans ensure that the app will be thoroughly tested and free of defects before its release. Any issues identified during the testing phase will be addressed promptly, ensuring that the app meets its quality standards. While there are some limitations to the proposed system, such as hardware requirements and platform compatibility, these can be addressed through effective marketing and development strategies.

Proposed enhancements, such as integration with wearable devices and personalized recommendations, can provide significant benefits to users and improve the app's overall functionality and user experience. Overall, the proposed Gym Club App has the potential to be an asset for individuals looking to improve their fitness and achieve their goals. With ongoing development and support, the app can continue to evolve and meet the changing needs of its users.

9 Bibliography

Book Reference

The One-Minute Workout - by Steven Holzner
Glute Lab: The Art and Science of Strength and Physique Training – Audio Book
www.wikipedia.org
www.codeproject.com
Visual Basic 6.0 Resource Centre
http://msdn.microsoft.com/hi-in/vbrun/default(en-us).aspx