VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA

A Mini Project Report

On

"Fitness App"

Submitted in the partial fulfillment for the requirements for
the conferment of degree of

BACHELOR OF ENGINEERING

In

INFORMATION SCIENCE AND ENGINEERING

By

Ms. MANASA CB	**1BY20IS077**
Mr. MANISH KUMAR	**1BY20IS078**

Under the guidance of

Dr. Gireesh Babu CN
Asst. Professor

BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT
YELAHANKA, BENGALURU-560064
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

2022-2023

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA

BMS INSTITUTE OF TECHNOLOGY& MANAGEMENT

YELAHANKA, BENGALURU-560064

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Mini Project work entitled **"Fitness App"** is a bonafide work carried out by **Ms. Manasa C B (1BY20IS077) and Manish Kumar (1BY20IS078)** in partial fulfillment of Mobile Application Development Laboratory with Mini Project (18CSMP68) for the award of **Bachelor of Engineering Degree in Information Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report. The project report has been approved as it satisfies the academic requirements in respect of Mini Project work for the B.E Degree.

Signature of the Guide	**Signature of the HOD**
Dr Gireesh Babu CN	Dr. Pushpa S.K.
Associate Professor	Professor and Head
Department of ISE	Department of ISE

EXTERNAL EXAMINERS

Name of the Examiners	Signature with Date
1.	
2.	

ABSTRACT

The Android Trip Count project is a mobile application designed to help users keep track of their travel activities and calculate the number of trips they have taken. The application offers a user-friendly interface that allows users to input relevant information about each trip, such as the starting and ending locations, travel mode, and purpose of the trip.

The main objective of the project is to provide users with a convenient and efficient way to monitor and analyze their travel patterns. By recording trip details, the application can generate valuable insights and statistics, such as the total number of trips, average trip duration, most frequently visited destinations, and preferred modes of transportation. This information can be useful for personal travel management, expense tracking, and future trip planning.

The Android Trip Count project utilizes various features and technologies provided by the Android platform, including user interface design, database management, and data visualization. The application leverages the built-in GPS capabilities of Android devices to automatically capture the starting and ending locations of each trip, eliminating the need for manual entry.

ACKNOWLEDGEMENT

We heartily thank our Principal **Dr. Mohan Babu G N**, BMS Institute of technology and Management for his constant encouragement and Inspiration in doing the project.

We heartily thank our Head of the Department **Dr. Pushpa S K**, Dept of Information Science and Engineering, BMS Institute of technology and Management for her constant encouragement and Inspiration in doing the project.

We gracefully thank our project guide **Dr. Gireesh Babu CN,** Assistant Professor, Dept. of Information Science and Engineering for her constant support, advice, guidance, and inspiration for the project.

Nevertheless, we express our gratitude towards our family and friends for the encouragement and support which helped us to finish this project successfully.

TABLE OF CONTENTS

CHAPTER	TOPIC	Pg. No.
1	Introduction	1-6
	1.1 Introduction to mobile application development	
	1.2 What is Mobile App?	
	1.3 What is Mobile OS?	
	1.4 Introduction to Android Studio	
	1.5 Android Architecture	
	1.6 Android Application Components	
	1.7 Problem Statement	
	1.8 Objectives	
	1.9 Project Application	
2	System Design	7-9
	2.1 Application Components of the Project	
	2.2 User Interface Design	
3	Implementation	10-16
	3.1 Explanation of the modules with Java Code and XML Code	
4	Results	17-20
	4.1 Screenshots of the Application	
5	Application Testing	21
6	6.1 Conclusion	22
	6.2 Future Enhancements	
7	References	23

List of Figures/Tables

Figure No.	Name of Figures	Page No.
1.5	Android architecture	4
1.6	Android components	5
2.1	Flow of the Activities in Application	8
2.2	User Interface	9
3.1	Retrieval of Data	12
3.2	Design of the application	13-14
3.3	Activity_main.java code	15-16
4.1	Creating a profile	17
4.2	Profile	17
4.3	Creating new profile	18
4.4	Weight Track Page	18
4.5	Timer	19
4.6	Main Page Exercises	19
4.7	Workout History	20

CHAPTER-1

1. INTRODUCTION

The Android Trip Count project is a mobile application designed to simplify and enhance the process of tracking and analyzing travel activities for Android users. In today's fast-paced world, people often find themselves traveling frequently for various purposes such as work, leisure, or personal commitments. However, keeping track of these trips and managing the associated information can be a daunting task.

As The Android Trip Count application aims to address this challenge by providing users with a convenient and intuitive tool to record, organize, and analyze their travel data. With the widespread use of smartphones and the popularity of the Android platform, this application offers a user-friendly interface and leverages the capabilities of Android devices to streamline the trip counting process.

A By capturing essential trip details such as starting and ending locations, travel mode, and purpose, users can maintain a comprehensive record of their travel activities. The application utilizes the built-in GPS functionality of Android devices to automatically detect and record location information, eliminating the need for manual entry and ensuring accuracy.

1.1 INTRODUCTION TO MOBILE APPLICATION DEVELOPMENT

Mobile application development is the process to making software for Smartphone and digital assistants, most commonly for Android and iOS.

The software can be preinstalled on the device, downloaded from a mobile app store or accessed through a mobile web browser. The programming and markup languages used for this kind of software development include Java, Swift, C# and HTML5.

Mobile app development is rapidly growing. From retail, telecommunications and e- commerce to insurance, healthcare and government, organizations across industries must meet user expectations for real-time, convenient ways to conduct transaction and access information. Today, mobile devices- and the mobile applications that unlock their value-are the most popular way for people and business to connect to the internet. To stay relevant, responsive and successful, organizations need to develop the mobile applications that their customers, partners demand.

1.2 WHAT IS MOBILE APP?

A mobile application or mobile app is a computer program or software application designed to run on a mobile device such as a phone, tablet, or watch. Apps were originally intended for productivity assistance such as email, calendar, and contact databases, but the public demand for apps caused rapid expansion into other areas such as mobile games, factory automation, GPS and location-based services, order-tracking, and ticket purchases, so that there are now millions of apps available.

Apps are generally downloaded from application distribution platforms which are operated by the owner of the mobile operating system, such as the App Store (iOS) or Google Play Store.

Mobile applications often stand in contrast to desktop applications which are designed to run on desktop computers, and web applications which run in mobile web browsers rather than directly on the mobile device.

Mobile App has many advantages like within a short app we can communicate a lot of information to the client/customers and even it is an ease of access to client/customer for services update or sale/purchase activity.

1.3 WHAT IS MOBILE OS?

A mobile operating system is an operating system for mobile phones, tablets, smart watches, 2-in-1 PCs, smart speakers, or other mobile devices. While computers such as typical laptops are 'mobile', the operating systems used on them are generally not considered mobile ones, as they were originally designed for desktop computers that historically did not have or need specific mobile features. This distinction is becoming blurred in some newer operating systems that are hybrid made for both uses.

A mobile OS is responsible for identifying and defining mobile device features and functions, including keypads, application synchronization, email, thumbwheel and text messaging. A mobile OS is like a standard OS (like Windows, Linux, and Mac) but is relatively simple and light and primarily manages the wireless variations of local and broadband connections, mobile multimedia and various input methods.

1.4 INTRODUCTION TO ANDROID STUDIO

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools,

Android Studio offers even more features that enhance your productivity when building Androidapps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices.
- Apply Changes to push code and resource changes to your running app without restarting your app.
- Code templates and GitHub integration to help you build common app features and import sample code.
- Extensive testing tools and frameworks.
- Lint tools to catch performance, usability, version compatibility, and other problems.
- C++ and NDK support.
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

Android Studio provides a unified environment where we can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto.

1.5 ANDROID ARCHITECTURE

Fig1.5 shows architecture of android architecture contains different number of components to support any android device needs. Android software contains an open-source Linux Kernel having collection of number of C/C++ libraries which are exposed through an application framework service. Among all the components Linux Kernel provides main functionality of operating system functions to smartphones and Dalvik Virtual Machine (DVM) provide platform for running an android application.

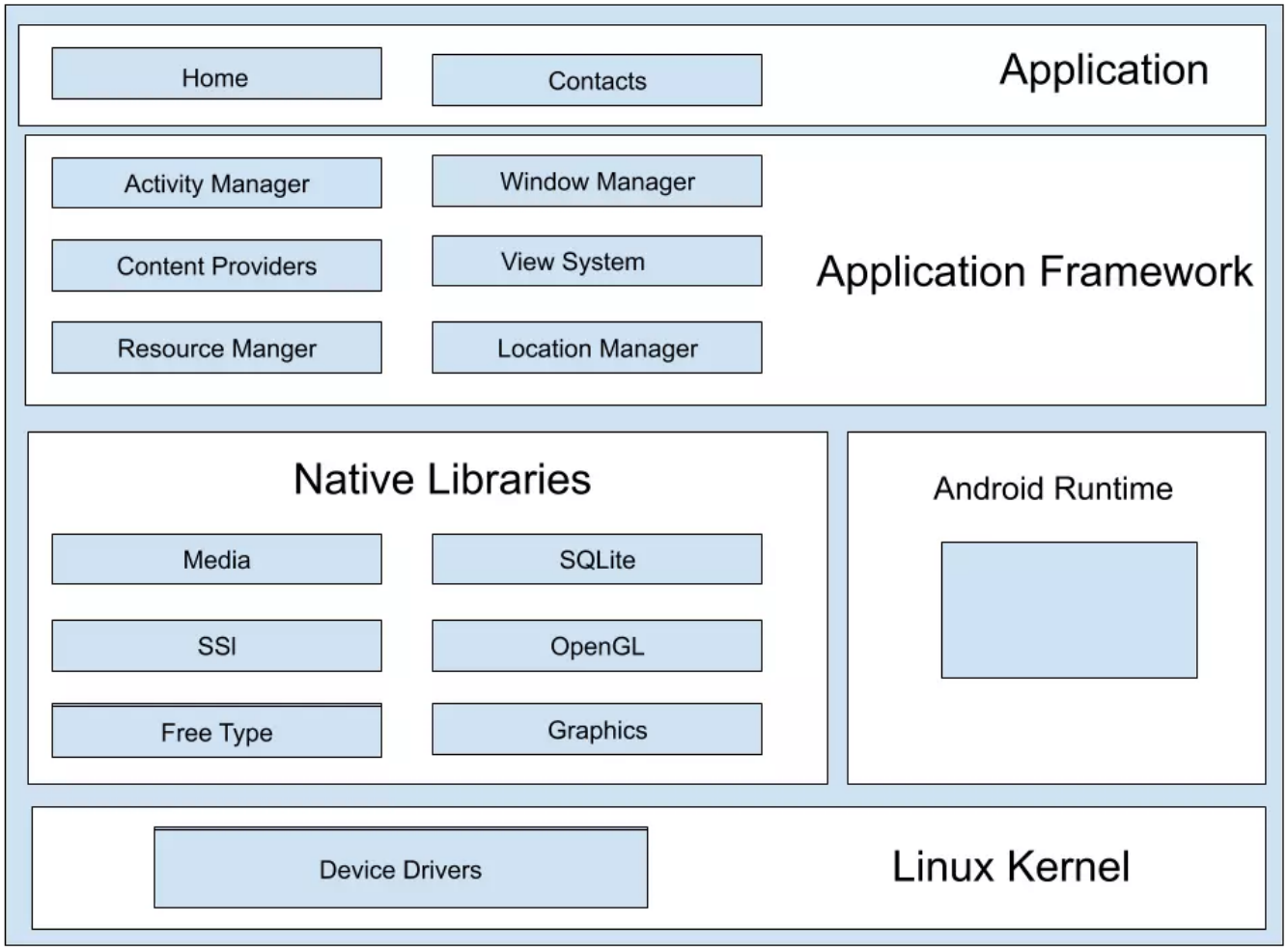


Fig 1.5: Android Architecture

1.6 ANDROID APPLICATION COMPONENTS

Application components are the essential building blocks of an Android application. These components are loosely coupled by the application manifest file AndroidManifest.xml that describes each component of the application and how they interact. Table 1.6 shows the Android Components.

Components & Description
Activities They dictate the UI and handle the user interaction to the smart phone screen.
Services They handle background processing associated with an application.
Broadcast Receivers They handle communication between Android OS and applications.
Content Providers They handle data and database management issues.
Fragments Represents a portion of user interface in an Activity.
Views UI elements that are drawn on-screen including buttons, lists forms etc.
Layouts View hierarchies that control screen format and appearance of the views.
Intents Messages wiring components together.
Resources External elements, such as strings, constants and drawable pictures.
Manifest Configuration file for the application.

Fig 1.6: Android components

1.7 PROBLEM STATEMENT

Fitness apps suffer from lack of automation features that allow users to input all required information together much quicker. It's hard to imagine users who would spend time on logging their food every single day for a more extended period. That's why many users abandon these apps after a month or two. Fitness apps are applications designed by companies to keep you fit and healthy. These apps can be downloaded on mobile phones quite easily. The aim of these apps is to make your lifestyle healthier by tracking your food intake, water intake and workout pattern. Some apps even keep a track of your heart rate and blood pressure, which is beneficial for individuals with high blood pressure. Some health and fitness apps even have a health coach, who help their clients to achieve their health goals effectively.

Our goal is to develop an innovative fitness app that addresses the challenges and limitations faced by individuals in achieving their fitness goals. The app aims to provide a comprehensive and personalized solution to promote physical activity, improve overall health, and enhance user engagement.

1.8 OBJECTIVES

- **User Engagement:** The primary objective is to develop a fitness app that captivates users and encourages them to engage actively in their fitness journey. This includes providing an intuitive user interface, personalized content, and interactive features that promote ongoing user interaction and participation.
- **Accountability and Motivation:** Establish mechanisms within the app to foster accountability and motivation. This includes features such as goal setting, progress tracking, achievement badges, social integration, and challenges. By creating a supportive environment and offering rewards and recognition, the app aims to motivate users to stay committed to their fitness goals.

1.9 PROJECT APPLICATION

Existing fitness apps often fail to provide personalized workout plans and recommendations tailored to individual needs, goals, and fitness levels. Users may feel overwhelmed or unmotivated due to generic content that does not cater to their specific requirements.

Without a supportive environment or accountability mechanisms, individuals often struggle to stay motivated and committed to their fitness goals. The absence of real-time feedback, progress tracking, and social interaction can lead to a lack of consistency and diminished results.

CHAPTER-2

SYSTEM DESIGN

System design is a critical phase in the development of a software application. It involves designing the architecture and components of the system to ensure its functionality, scalability, reliability, and performance. The system design phase focuses on creating a blueprint for how different modules, components, and services will work together to achieve the desired goals of the application.

2.1 APPLICATION COMPONENTS OF PROJECT

Application components are Core Building Blocks of an Android Application. It is an entry Pointfor System or Users from which they can enter in App.

The project consists of the following components:

1. Activities:

An Activity is a fundamental component of an Android application. It represents a single screen with a user interface where users can interact with the app. Activities play a crucial role in navigating between different screens and managing the app's lifecycle. Activities are essential building blocks of an Android app, allowing developers to create interactive and responsive user interfaces. Multiple activities can be combined to create complex app flows and user experiences. Activities work in conjunction with other components such as Fragments, Services, and Broadcast

The clicks on the views will lead to triggerof respective activities, the flow is represented as:

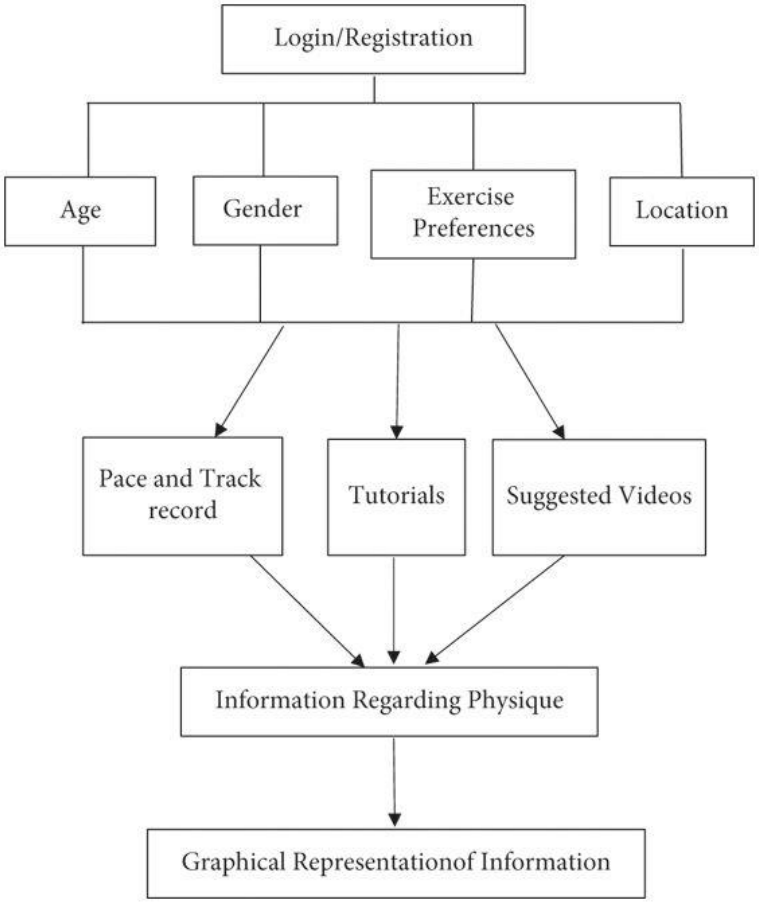


Fig 2.1: Flow of the Activities in Application

2. Views:

View is the basic building block of UI (User Interface) in android. View refers tothe android view. View class, which is the super class for all the GUI components.

The app contains Text View, Image View, Button, Edit Text, List View, Progress Bar, Card View which helps in achieving the flow of Activities and design the layout responsively.

Text Views are used to display the field labels and the details of the Admin. Edit Texts are used to take the details during login, register and vehicle entry. Image Views are used to display the icon of the app and List Views are used to display the slots for the users with details of number of available slots, area, name and display the admin list of parked vehicles and list of vehicles visited.

3. Layouts:

Android Layout is used to define the user interface that holds the UI controls or widgetsthat will appear on the screen of an android application or activity screen.The app is designed using constraint layout which shows the user options that is Trip Expenses and Day2Day expenses.

4. Intents, Resources:

The primary intent of the project is to develop a mobile application that allows users to track and record their travel expenses. Users can input various expense categories such as transportation, accommodation, meals, entertainment, and miscellaneous expenses.

2.2 USER INTERFACE DESIGN

User Interface (UI) design in Android Studio involves creating visually appealing and user-friendly interfaces for Android applications. Android Studio provides various tools and features to assist developers in designing and refining the UI of their apps. The Fig 2.2 shows the design of the application in the grocery app using class diagram.

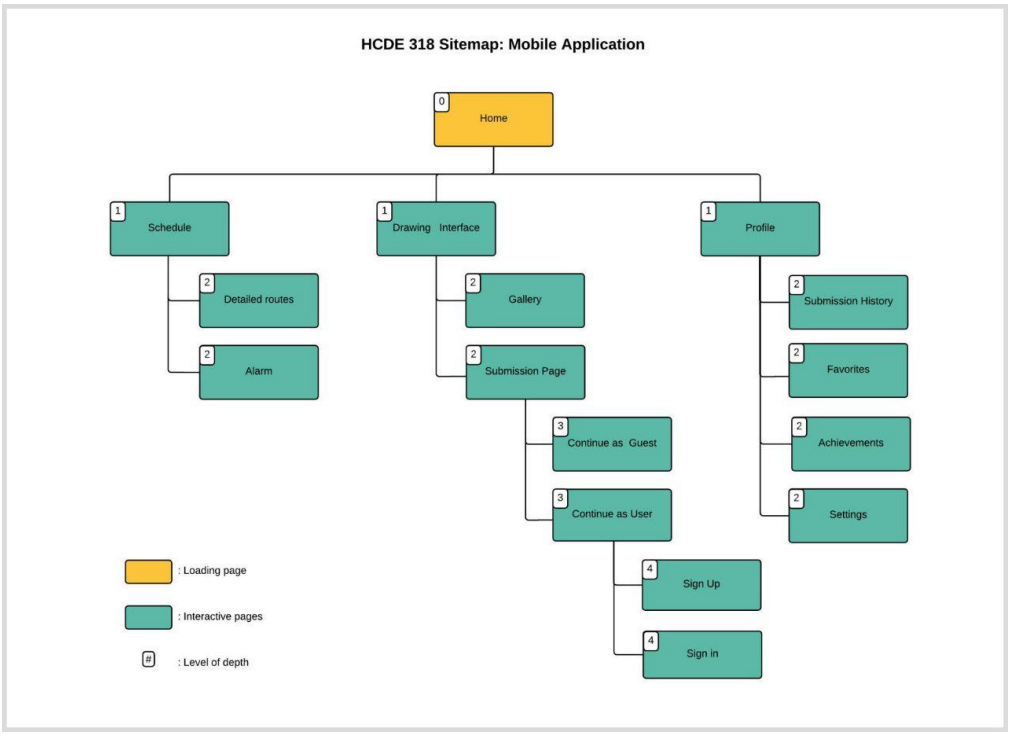


Fig 2.2: User Interface

CHAPTER-3

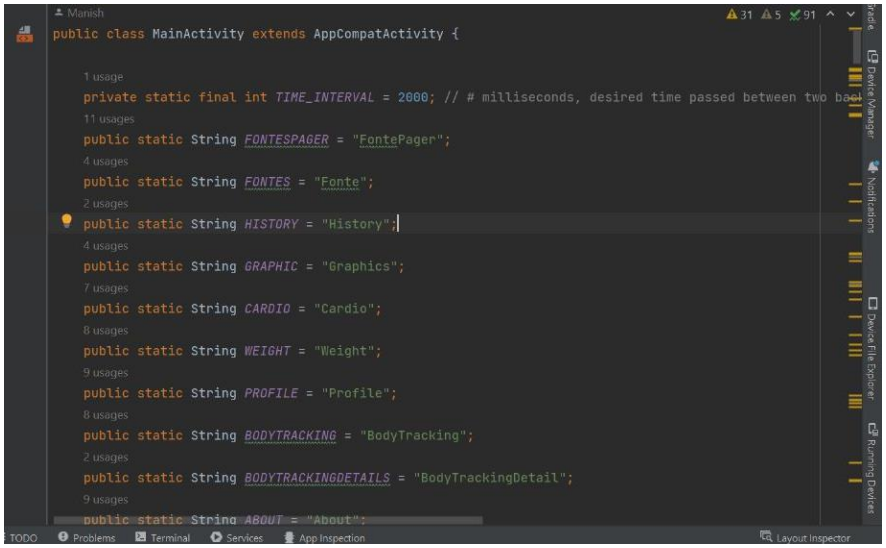
3. IMPLEMENTATION

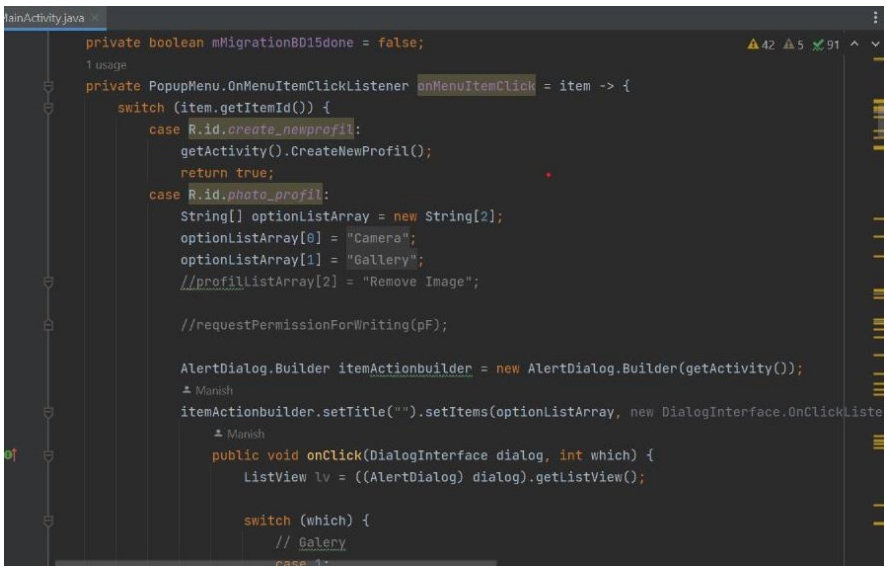
Implementing an Android app in Android Studio involves writing code, creating resources, and configuring various components to bring your app to life.

3.1 EXPLANATION OF THE MODULES WITH JAVA AND XML CODE

1. Main activity:

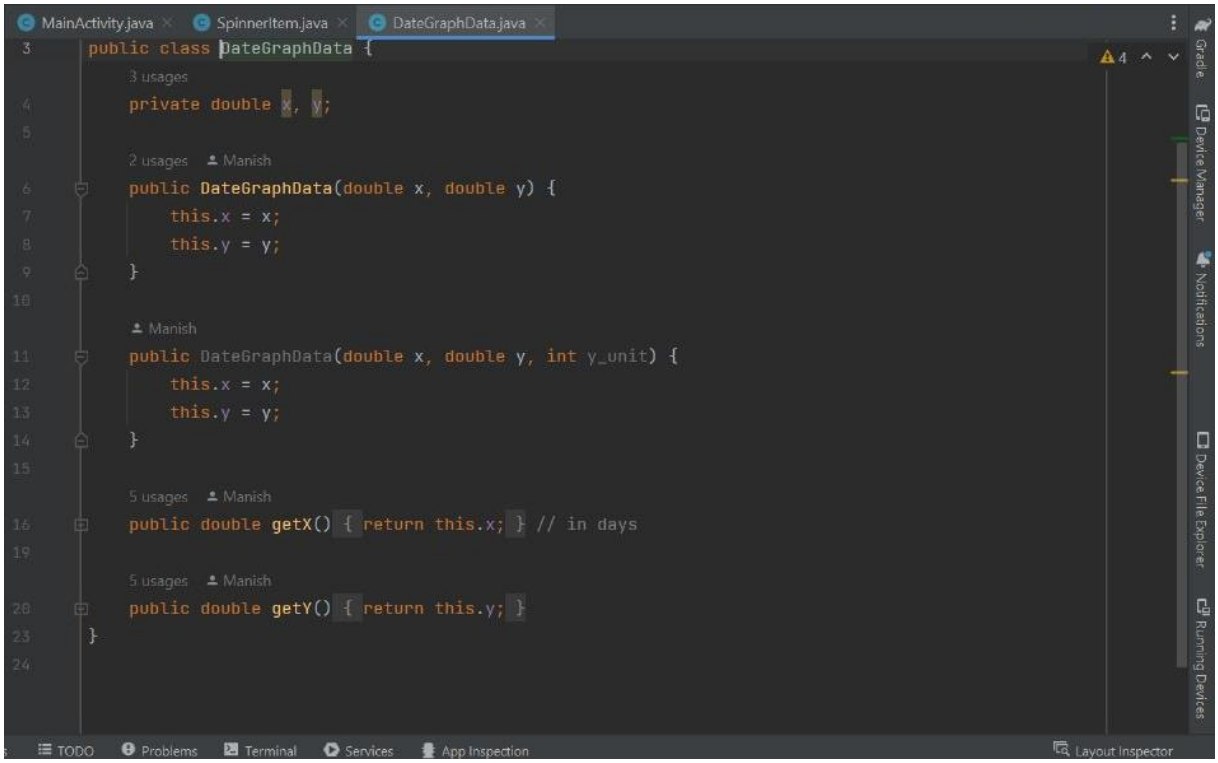
The main operation carried out throughout the app is adding the data. The main activity.java is the main code used in this application.

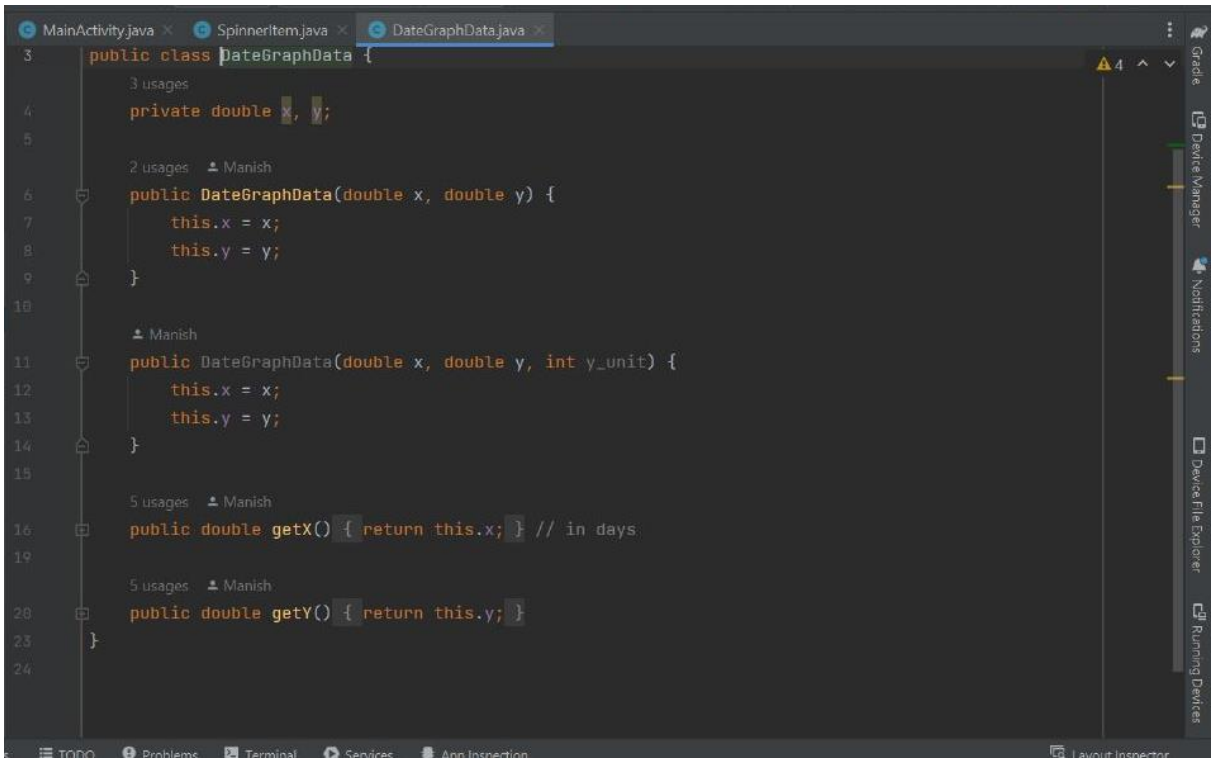




2. Retrieving the data from database:

Use the appropriate database driver or library provided by your programming language to execute the query. Pass the query as a parameter to the relevant method or function. After executing the query, you will receive a result set containing the data returned by the query.





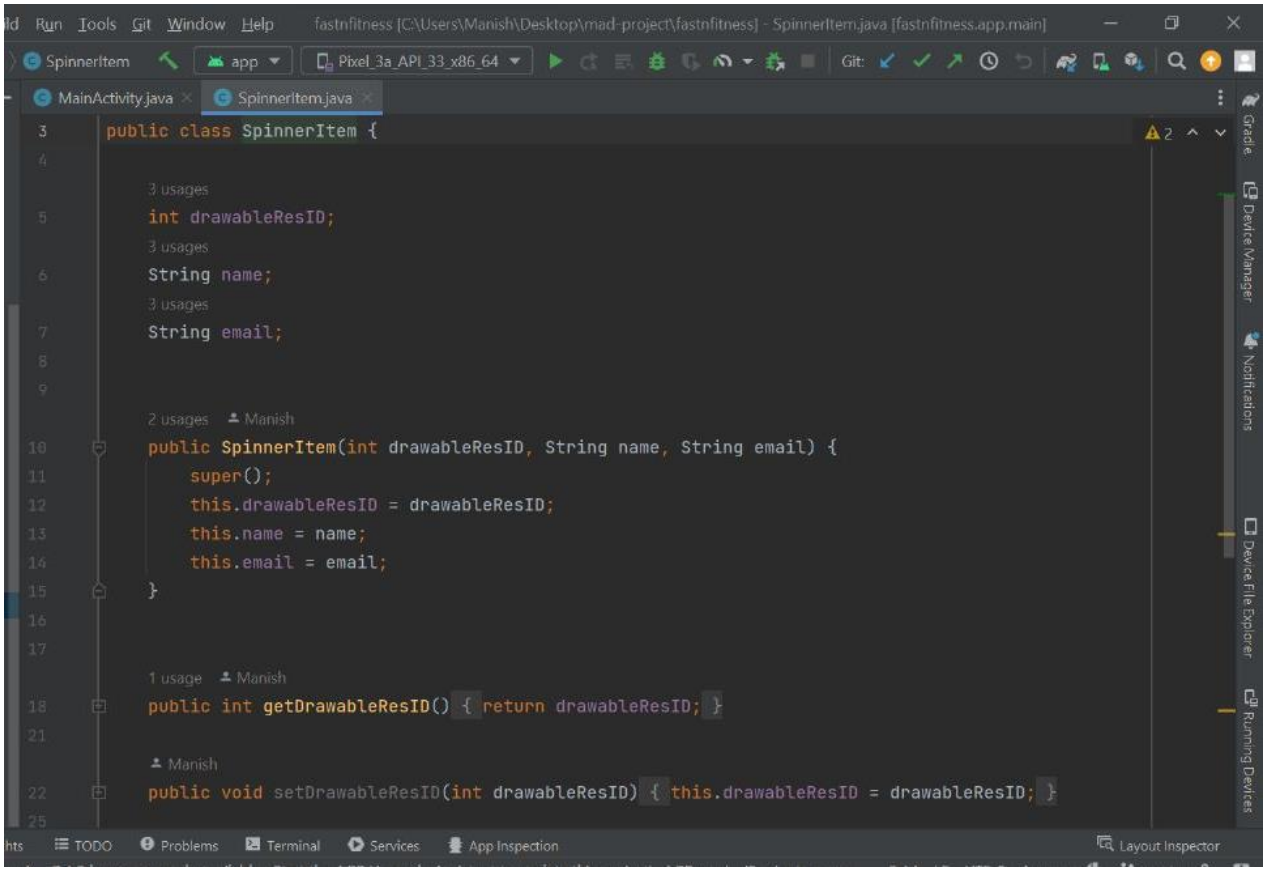


Fig 3.1 Retrieval of data

XML:

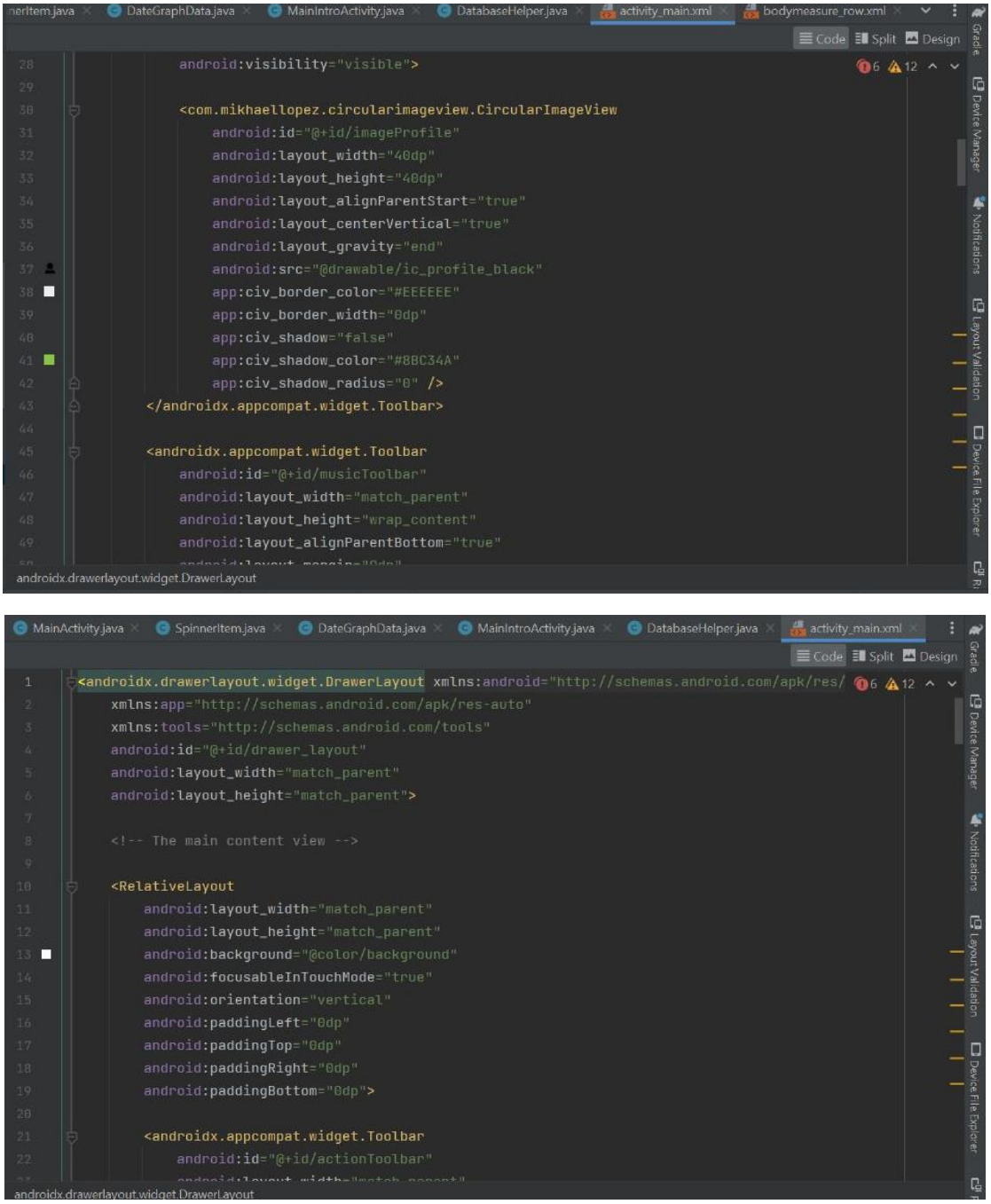
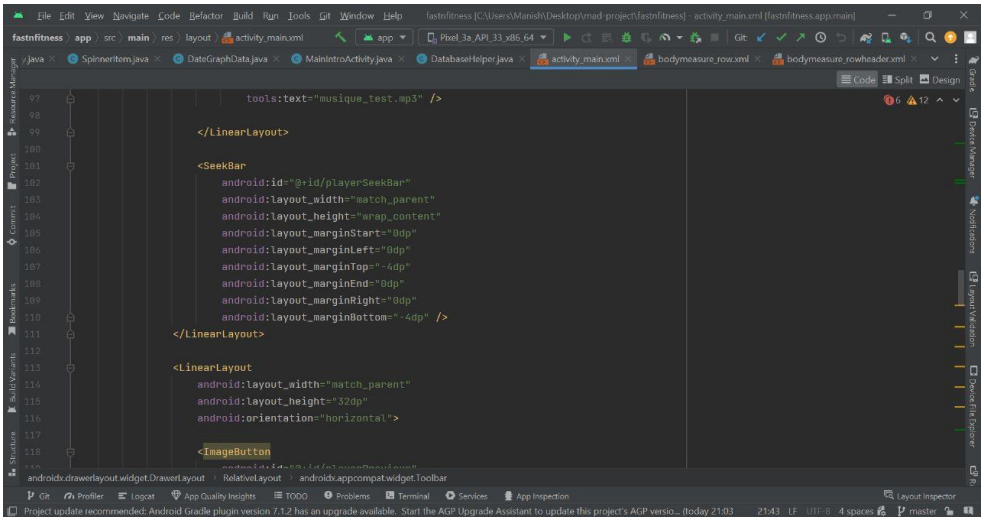
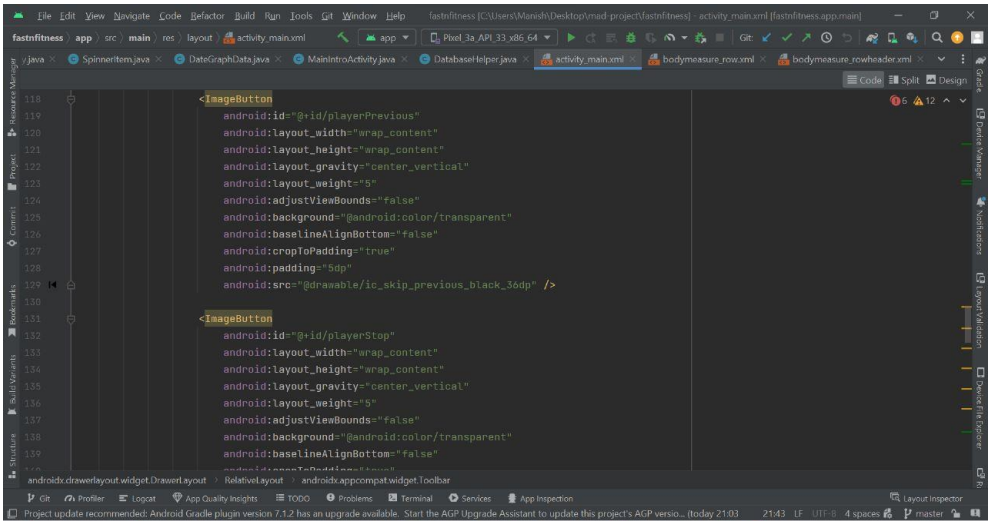


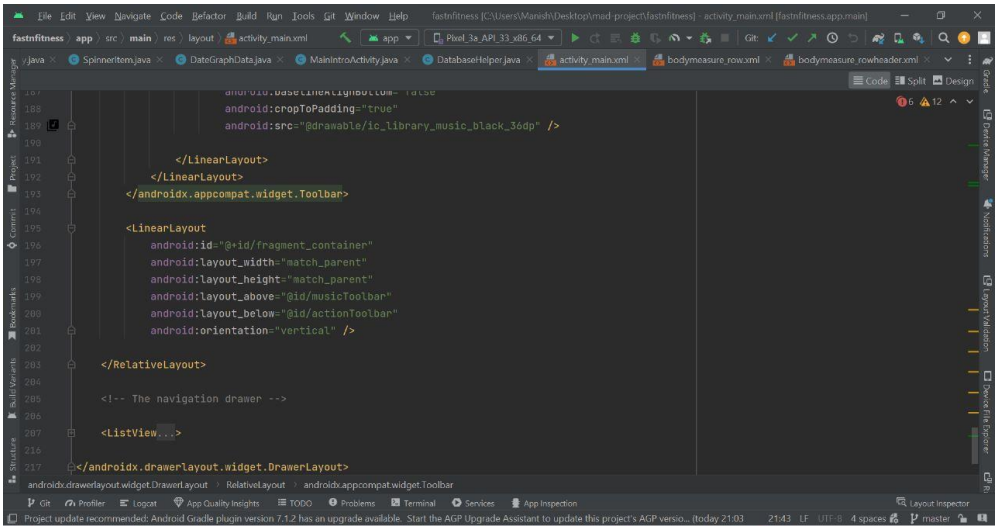
Fig 3.2 Design of the Application



```
                    tools:text="musique_test.mp3" />

            </LinearLayout>

            <SeekBar
                android:id="@+id/playerSeekBar"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginStart="0dp"
                android:layout_marginLeft="0dp"
                android:layout_marginTop="-4dp"
                android:layout_marginEnd="0dp"
                android:layout_marginRight="0dp"
                android:layout_marginBottom="-4dp" />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="32dp"
            android:orientation="horizontal">

            <ImageButton
```



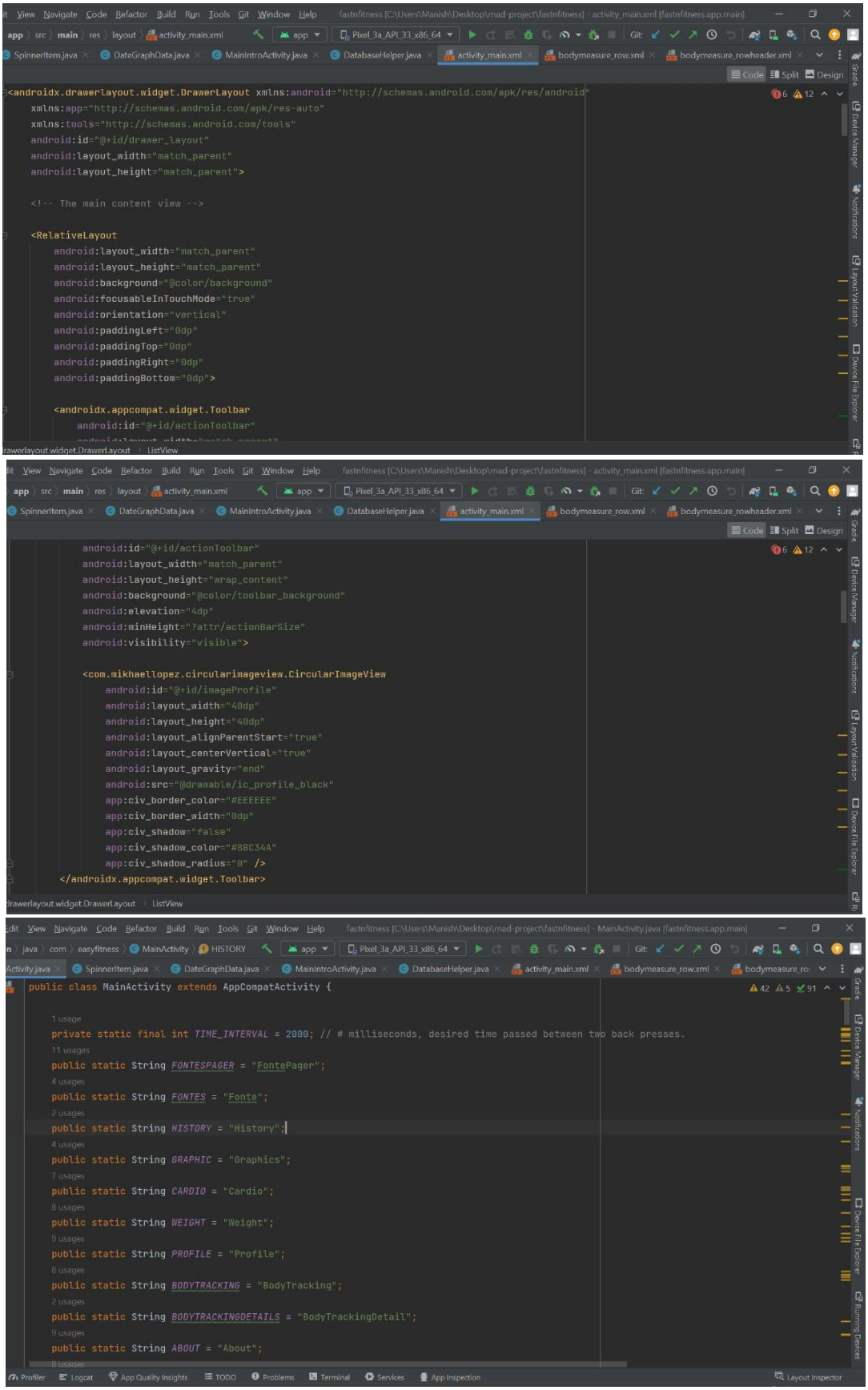
```
            <ImageButton
                android:id="@+id/playerPrevious"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center_vertical"
                android:layout_weight="5"
                android:adjustViewBounds="false"
                android:background="@android:color/transparent"
                android:baselineAlignBottom="false"
                android:cropToPadding="true"
                android:padding="5dp"
                android:src="@drawable/ic_skip_previous_black_36dp" />

            <ImageButton
                android:id="@+id/playerStop"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center_vertical"
                android:layout_weight="5"
                android:adjustViewBounds="false"
                android:background="@android:color/transparent"
                android:baselineAlignBottom="false"
```



```
                android:cropToPadding="true"
                android:src="@drawable/ic_library_music_black_36dp" />

            </LinearLayout>
        </LinearLayout>
    </androidx.appcompat.widget.Toolbar>

    <LinearLayout
        android:id="@+id/fragment_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@id/musicToolbar"
        android:layout_below="@id/actionToolbar"
        android:orientation="vertical" />

</RelativeLayout>

<!-- The navigation drawer -->

<ListView...>

</androidx.drawerlayout.widget.DrawerLayout>
```

Java:

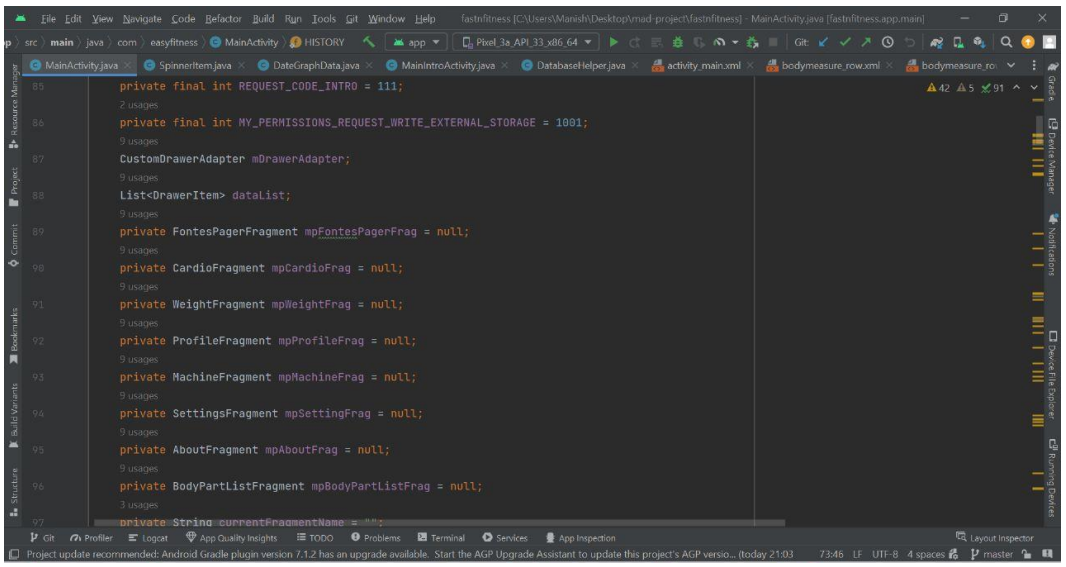
Fig 3.3: activity _main.java code



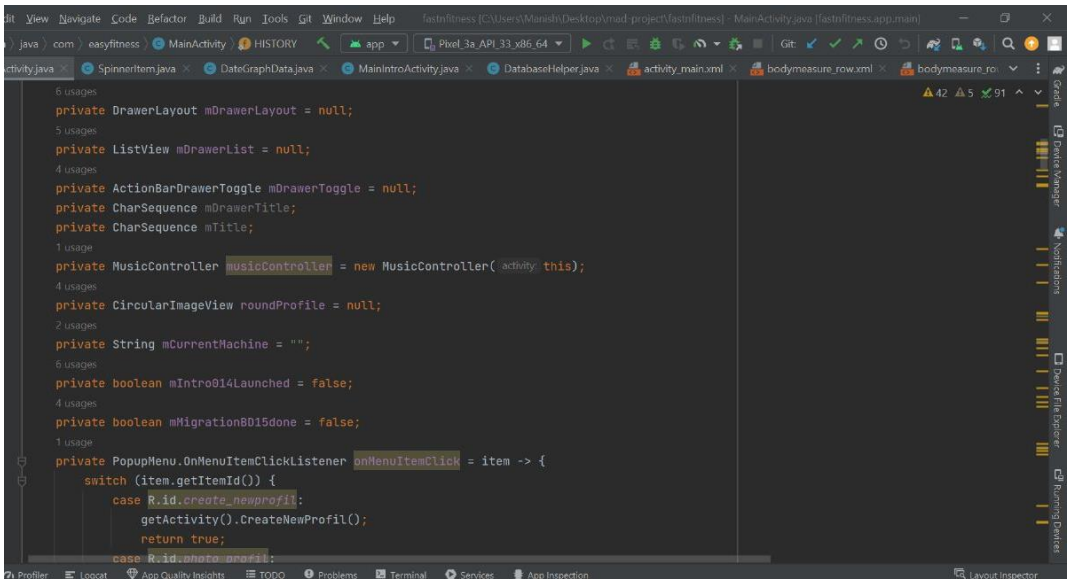
```
<androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- The main content view -->

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/background"
        android:focusableInTouchMode="true"
        android:orientation="vertical"
        android:paddingLeft="0dp"
        android:paddingTop="0dp"
        android:paddingRight="0dp"
        android:paddingBottom="0dp">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/actionToolbar"
```

```
            android:id="@+id/actionToolbar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/toolbar_background"
            android:elevation="4dp"
            android:minHeight="?attr/actionBarSize"
            android:visibility="visible">

            <com.mikhaellopez.circularimageview.CircularImageView
                android:id="@+id/imageProfile"
                android:layout_width="40dp"
                android:layout_height="40dp"
                android:layout_alignParentStart="true"
                android:layout_centerVertical="true"
                android:layout_gravity="end"
                android:src="@drawable/ic_profile_black"
                app:civ_border_color="#EEEEEE"
                app:civ_border_width="0dp"
                app:civ_shadow="false"
                app:civ_shadow_color="#8BC34A"
                app:civ_shadow_radius="0" />
        </androidx.appcompat.widget.Toolbar>
```

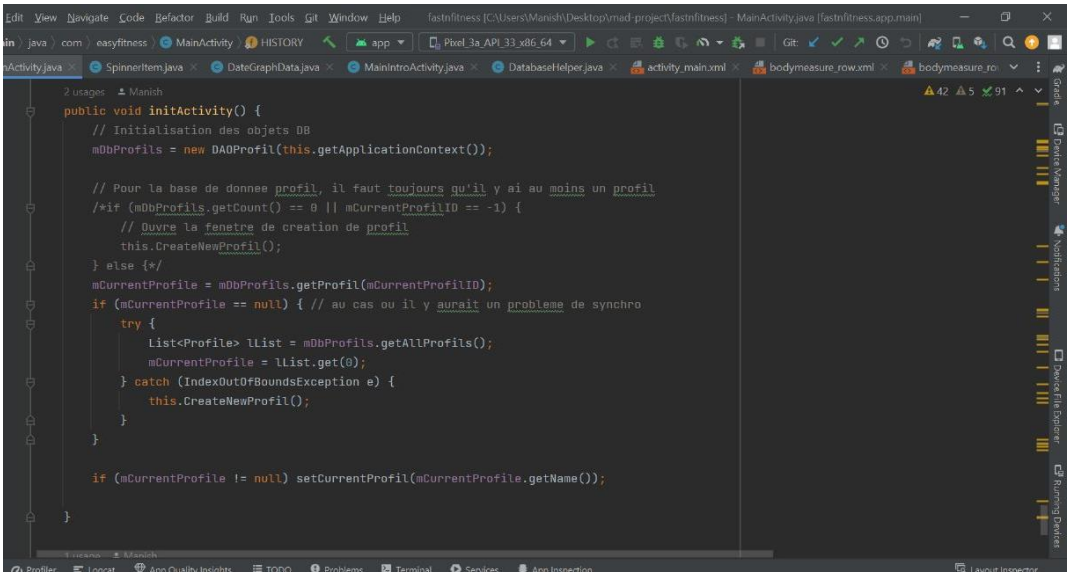
```
public class MainActivity extends AppCompatActivity {


    1 usage
    private static final int TIME_INTERVAL = 2000; // # milliseconds, desired time passed between two back presses.
    11 usages
    public static String FONTESPAGER = "FontePager";
    4 usages
    public static String FONTES = "Fonte";
    2 usages
    public static String HISTORY = "History";
    4 usages
    public static String GRAPHIC = "Graphics";
    7 usages
    public static String CARDIO = "Cardio";
    8 usages
    public static String WEIGHT = "Weight";
    9 usages
    public static String PROFILE = "Profile";
    8 usages
    public static String BODYTRACKING = "BodyTracking";
    2 usages
    public static String BODYTRACKINGDETAILS = "BodyTrackingDetail";
    9 usages
    public static String ABOUT = "About";
```



```
private final int REQUEST_CODE_INTRO = 111;
2 usages
private final int MY_PERMISSIONS_REQUEST_WRITE_EXTERNAL_STORAGE = 1001;
9 usages
CustomDrawerAdapter mDrawerAdapter;
9 usages
List<DrawerItem> dataList;
9 usages
private FontesPagerFragment mpFontesPagerFrag = null;
9 usages
private CardioFragment mpCardioFrag = null;
9 usages
private WeightFragment mpWeightFrag = null;
9 usages
private ProfileFragment mpProfileFrag = null;
9 usages
private MachineFragment mpMachineFrag = null;
9 usages
private SettingsFragment mpSettingFrag = null;
9 usages
private AboutFragment mpAboutFrag = null;
9 usages
private BodyPartListFragment mpBodyPartListFrag = null;
3 usages
private String currentFragmentName = "";
```



```
6 usages
private DrawerLayout mDrawerLayout = null;
5 usages
private ListView mDrawerList = null;
4 usages
private ActionBarDrawerToggle mDrawerToggle = null;
private CharSequence mDrawerTitle;
private CharSequence mTitle;
1 usage
private MusicController musicController = new MusicController( activity: this);
4 usages
private CircularImageView roundProfile = null;
2 usages
private String mCurrentMachine = "";
6 usages
private boolean mIntro014Launched = false;
4 usages
private boolean mMigrationBD15done = false;
1 usage
private PopupMenu.OnMenuItemClickListener onMenuItemClick = item -> {
    switch (item.getItemId()) {
        case R.id.create_newprofil:
            getActivity().CreateNewProfil();
            return true;
        case R.id.photo_profil:
```



```
2 usages  Manish
public void initActivity() {
    // Initialisation des objets DB
    mDbProfils = new DAOProfil(this.getApplicationContext());

    // Pour la base de donnee profil, il faut toujours qu'il y ai au moins un profil
    /*if (mDbProfils.getCount() == 0 || mCurrentProfilID == -1) {
        // Ouvre la fenetre de creation de profil
        this.CreateNewProfil();
    } else {*/
    mCurrentProfile = mDbProfils.getProfil(mCurrentProfilID);
    if (mCurrentProfile == null) { // au cas ou il y aurait un probleme de synchro
        try {
            List<Profile> lList = mDbProfils.getAllProfils();
            mCurrentProfile = lList.get(0);
        } catch (IndexOutOfBoundsException e) {
            this.CreateNewProfil();
        }
    }

    if (mCurrentProfile != null) setCurrentProfil(mCurrentProfile.getName());

}
```

CHAPTER-4

RESULTS

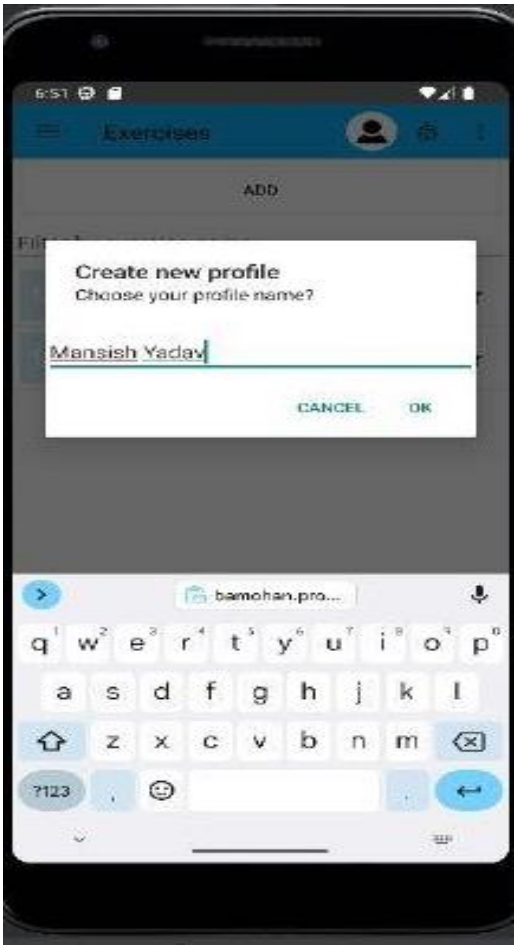


Fig 4.1: Creating a Profile

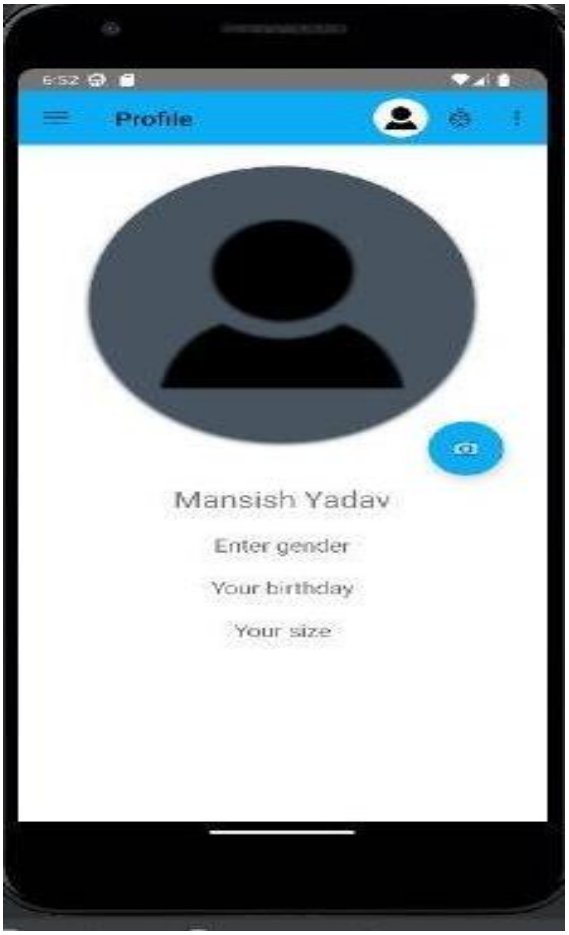


Fig 4.2: Profile

Fig 4.1 shows the user with his name and the user can press ok to create a profile

Fig 4.2 shows the frontend of the created profile.

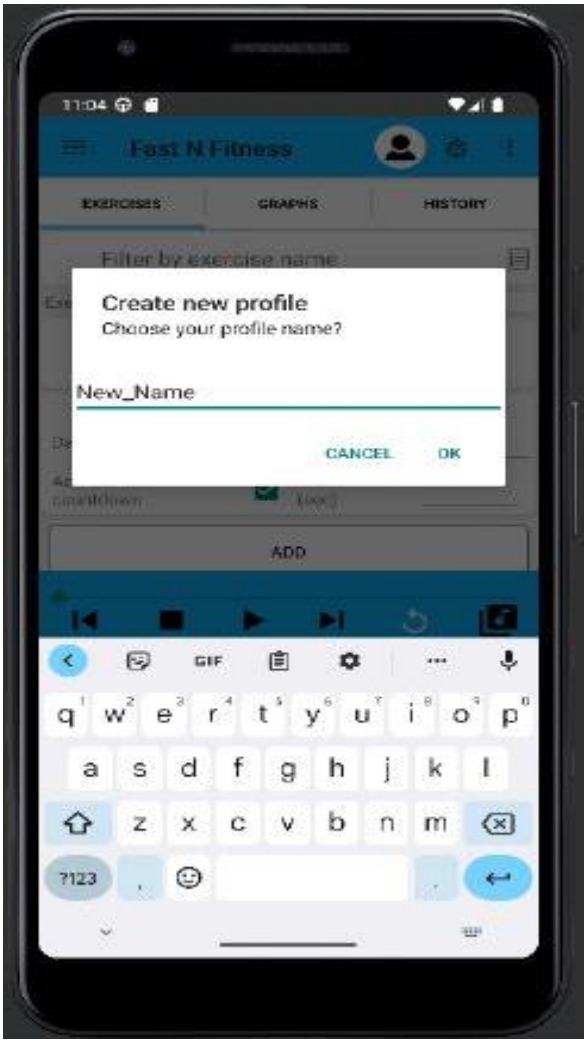


Fig. 4.3 Creating a new profile

Fig.4.3 allows the user to add new users

Fig. 4.4 Weight Track Page

Fig 4.4 allows the user to know about his weight ,fat, muscles and water to consume

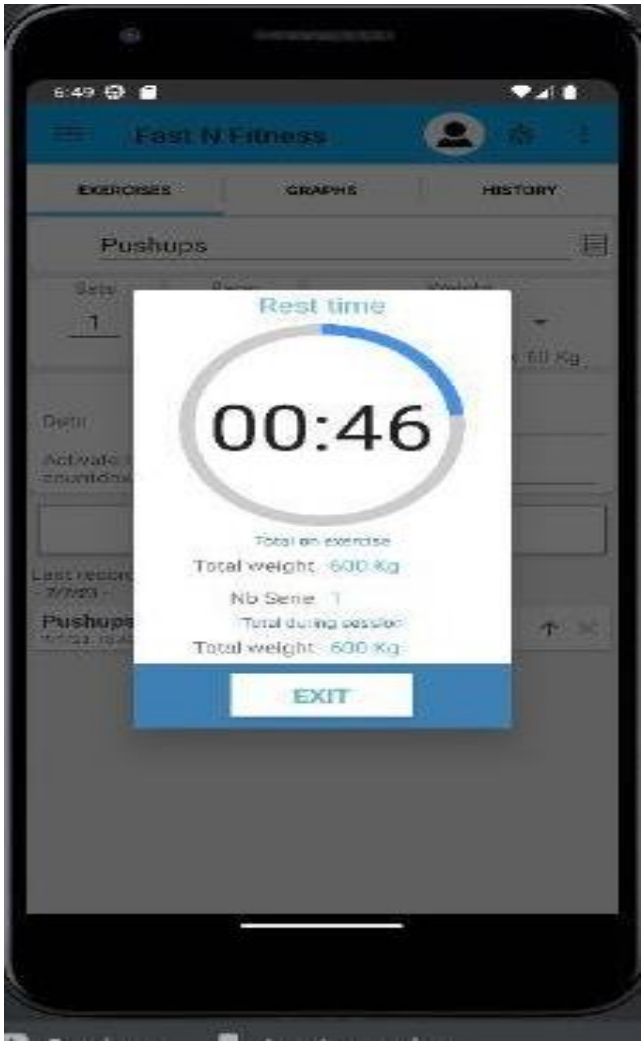


Fig 4.5 Timer

Fig 4.5: The above user interface provides a timer for the user to proceed with exercises.

Fig 4.6: Main Page Exercises

Fig 4.6 : The above page helps to keep track of exercises done.

Fig. 3.7: Workout History

Fig 3.7 shows all the exercises done by the user

CHAPTER-5

Application Testing

When it comes to testing a fitness app, there are several aspects you should consider to ensure its functionality, usability, and overall user experience. Here are some key areas to focus on during the testing process:

1. Functional Testing: Verify that all the features and functionalities of the fitness app are working as intended. Test various scenarios, such as creating a new user account, logging in/out, tracking workouts, setting goals, and any other core functionality the app offers.

2. Compatibility Testing: Test the app on different devices, operating systems (iOS, Android), and screen resolutions to ensure it works well across various platforms. Pay attention to how the app adapts to different screen sizes and orientations.

3. Performance Testing: Assess the app's performance under different conditions, including normal usage, heavy usage, and low network connectivity. Test the app's responsiveness, loading times, and the impact on device resources (CPU, memory, battery) to ensure it performs optimally.

4. Usability Testing: Evaluate the user interface (UI) and user experience (UX) of the app. Ensure that the app is intuitive, easy to navigate, and visually appealing. Test the app with real users to gather feedback on the app's usability and make improvements accordingly.

5. Security Testing: Verify that the app handles user data securely. Test authentication and authorization mechanisms to prevent unauthorized access. Check for vulnerabilities like data leakage, insecure data storage, and encryption protocols.

6. Compatibility with Fitness Devices: If the app integrates with fitness devices (such as smartwatches, heart rate monitors, or pedometers), test the compatibility and accuracy of data synchronization between the app and the devices.

7. Stress Testing: Simulate heavy user loads to determine how the app performs under high traffic conditions. Test scenarios such as simultaneous logins, concurrent workout tracking, or sharing workout updates on social media.

CHAPTER-6

CONCLUSION

6.1. CONCLUSION

The Fitness apps are applications designed by companies to keep you fit and healthy. These apps can be downloaded on mobile phones quite easily. The aim of these apps is to make your lifestyle healthier by tracking your food intake, water intake and workout pattern.

Some apps even keep a track of your heart rate and blood pressure, which is beneficial for individuals with high blood pressure. Some health and fitness apps even have a health coach, who help their clients to achieve their health goals effectively.

6.2 FUTURE ENHANCEMENTS

It has made the world smaller. And, the techies are becoming more and more engrossed with these new advancements in the field of science and technology. And thereby, neglecting their health widely.

This has created a havoc on the health-giving rise to the following disorders. Obesity is certainly one of the worst consequences. Also, abnormalities in blood pressure and level of blood sugar, depression, cardio vascular disorders, anxiety, different types of cancer and many more. The technology hasn't made our life complicated. Yes, there are some negative impacts on health. But it has provided a lot of alternatives to outnumber the negative ones. Thanks to the technological development that you can now start exercising at your desk right in your workplace.

Several under desk cycles and elliptical bikes have been developed, which no longer create the problem of restricted blood flow and constant inactivity even if you sit for more than 8 hours in your workstation.

CHAPTER-7

REFERENCES

[1]. Android Developers, Location APIs. URL: http://developer.android.com/google/playservice s/location.html

[2]. https://nodejs.org/en/docs

[3]. https://devcenter.heroku.com/categories/reference

[4]. https://devcenter.heroku.com/articles/getting-started-with-node