VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, Belgaum - 590 014.

2020 - 2021

A

Mini project report on

"newsMag"

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

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

Submitted by

DHANUSH K VIJAY

(1AT18IS027)

GOUTAM NARASIMHA HEGDE

(1AT18IS032)

Under the guidance of

Ms. Uzma Sulthana

Assistant Professor

Dept. of ISE, ATRIA I. T.

&

Ms. Prapulla G

Assistant Professor

Dept. of ISE, ATRIA I. T.

ATRIA INSTITUTE OF TECHNOLOGY

Department of Information Science and Engineering,

Bengaluru - 560 024

ATRIA INSTITUTE OF TECHNOLOGY
(Affiliated to Visvesvaraya Technological University)
ASKB Campus, Anandnagar,
Bengaluru – 560024

Department of Information Science and Engineering

CERTIFICATE

Certified that the project work entitled ”**newsMag**” carried out by **DHANUSH K VIJAY** bearing USN : 1AT18IS027 and **GOUTAM NARASIMHA HEGDE** bearing USN : 1AT18IS032, the bonafide students of Department of Information Science and Engineering, Atria I. T., in partial fulfillment for the award of **Bachelor of Engineering** in Information Science & Engineering of the Visvesvaraya Technological University, Belgavi during the year 2020-2021. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Ms. Uzma Sulthana/
Ms. Prapulla G
Asst. Professor - Project Guide
Department of I.S.E.,
Atria I. T.

Dr. Shanthi Mahesh
Head of Department
Department of I.S.E.,
Atria I.T.

Dr. T.N Sreenivasa
Principal
Atria I.T.

External Viva

Name of Examiners	**Signature with date**
1.	
2.	

DECLARATION

We, Dhanush K Vijay (USN : 1AT18IS027) & Goutam Narasimha Hegde (USN : 1AT18IS032), Students of sixth semester, Bachelor of Engineering, Atria Institute of Technology hereby declare that the mini project entitled **“newsMag”** has been carried out by us at Atria Institute of Technology, Bengaluru and submitted in partial fulfillment of the course requirements for the award of the degree of **Bachelor of Engineering** in **Information Science & Engineering** of **Visvesvaraya Technological University, Belgavi**, during the academic year 2020-2021.

We also declare that, to the best of our knowledge and belief, the work reported here doesn’t from part of any other dissertation on the basis of which a degree or award was conferred on an earlier occasion on this by any other student.

Place: **DHANUSH K VIJAY**

Date: **(USN : 1AT18IS027)**

GOUTAM NARASIMHA HEGDE

(USN : 1AT18IS032)

ABSTRACT

The application for viewing news ‘newsMag’ is helpful for the people to read news on the go with the busy life. Currently it is difficult to read pages of news each day as it is very time consuming. Our newsMag application helps them by providing the news on short. There are various activities like getting started to the app, showing the news articles, their category, sharing the app, about the app and the user can give feedback.

The application has a GET STARTED button which helps the user on how to use the app. It has registration page where the user can get himself an account created by entering a unique username and specifying a password satisfying the conditions: minimum length 8 containing numbers, special characters and combination of both uppercase and lowercase letters.

The user can use the application once logged in using the registered credentials. The user can have a good experience using the application that once logged in there is no need to log in again unless the user clicks on Log out.

ACKNOWLEDGEMENT

We are grateful to our institution, **Atria Institute of Technology**, for having provided us with the facilities to successfully complete this mini project on newsMag.

We thank **Dr. T. N. Sreenivasa, principal** and **Dr. Shanthi Mahesh, HOD, ISE** for providing us all the necessary facilities for successful completion of our mini project.

Deadlines play a very important role in successful completion of the academic project on time, efficiently and effectively. We take this opportunity to express our deep sense of gratitude to our guide and coordinators **Ms. Uzma Sulthana, Assistant Professor and Ms. Prapulla G, Assistant Professor, Department of ISE** for their valuable guidance and help throughout the course of the academic mini project. They have always been patient with us and helped immensely in completing the task on hand. We also thank them for the immense support, guidance, specifications & ideas without which seminar would have been completed without full merit.

Last but not least from the Department of Information Science and Engineering, teaching and non-teaching staffs for their constant encouragement, support, patience and endurance shown during the preparation of this report were remarkable. We also thank the management.

Finally, we thank our parents and friends for their motivation, moral and material support.

DHANUSH K VIJAY
(USN : 1AT18IS027)

GOUTAM NARASIMHA HEGDE
(USN : 1AT18IS032)

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction to Mobile Application Development

Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources. Like web application development, mobile application development has its roots in more traditional software development.

1.1.1 History:

- The first mobile phones were invented whose microchips required the most basic software to send and receive voice calls.
- On 3rd of April 1973, Martin Cooper of Motorola made the first call on the mobile phone to Dr. Joel S. Engel of the Bell Labs.
- The R&D department of IBM Simon came up with the first mobile app for Smartphones in 1993 exactly two decades after the first call was made.
- EPOC, first operating system developed by Psion, released in the early 90s, this was first of the recognizable apps.
- Palm OS, developed by Palm Inc. in the year 1996, these were mainly designed for personal digital assistants and were known as Garnet OS.
- The wireless markup language was specifically designed for devices that were dependent on XML and could be run across wireless application protocols.
- Java ME or J2ME or JME – it was first introduced as JSR 68. It was given various shapes and forms for use via Phones, embedded devices, and even PDAs.
- Symbian, developed by Symbian Ltd, which was a joint venture from Ericsson, Motorola, Nokia and PSION, this was a further developed version of PSION EPOC.
- Later on, the smartphones and iPhones that we use today evolved, making lives a lot easier for people.

1.1.2 Advantages:

- Improves Efficiency.
- Offers High Scalability.
- Secures the App Data.
- Integrates With Existing Software.
- Easy to Maintain.
- Improves Customer Relationship.
- Facilitates New Client Data Retrieval.
- Provides Real-time Project Access.
- Ease in Project Management.

1.2 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA software. It provides the fastest tools for building apps on every type of android device. It is a purpose-built for android to accelerate the development and helps to build the highest-quality apps for every android device. Features of Android Studio include,

- A flexible Gradle-based build system.
- A fast and feature-rich emulator.
- A unified environment where one can develop for all Android devices.
- Extensive testing tools and frameworks.

1.2.1 Android Studio SDK:

Android SDK performs all the tasks needed to develop apps for all versions of Android. This program is a necessary tool for any developer who wants to make smoothly running applications for the latest systems. It uses Java for development and relies on the Integrated Development Environment, to build the apps and test them.

1.2.2 Android Studio Emulator:

The Android Emulator simulates Android devices on your computer so that you can test your application on a variety of devices and Android API levels without needing to have each physical device. The emulator provides almost all of the capabilities of a real Android device. Simulation of incoming phone calls and text messages, specify the location of the device, simulate different network speeds, simulate rotation and other hardware sensors, access the Google Play Store, and much more are possible.

1.3 JAVA

Java is an object-oriented programming language created by James Gosling, Mike Sheridan, and Patrick Naughton in 1991. It is a high-level, class-based language that is designed to have a few implementation dependencies as possible. It is a general-purpose programming language intended to let android developers run the compiled Java code on all platforms that support Java without any need for recompilation. Features of Java include,

- **Simple:** Java is designed to be easy to learn.
- **Secure:** With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- **Architecture-neutral:** Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- **Portable:** Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- **Robust:** Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking.
- **Multithreaded:** With Java's multithreaded feature it is possible to write programs
- that can perform many tasks simultaneously.
- **Interpreted:** Java byte code is translated on the fly to native machine instructions and is not stored anywhere.

- **High Performance:** With the use of Just-In-Time compilers, Java enables high performance.
- **Distributed:** Java is designed for the distributed environment of the internet.
- **Dynamic**: Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time.

1.4 XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML focus on simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Features of XML include,

- XML focuses on data rather than how it looks.
- Easy and efficient data sharing.
- Compatibility with other markup language HTML.
- Supports platform transition.
- Allows XML validation.
- Adapts technology advancements.
- XML supports Unicode.

1.5 Structure of Report

This report is for our mini project newsMag using Mobile Application Development concepts. Our report consists of five chapters where in first chapter we are giving introduction to mobile application development with its history. In second chapter we have given brief description of our problem definition and literature survey made. Similarly in third chapter we have a brief document on the requirement specifications of hardware and software required along with purpose and scope of our project. The fourth and fifth module gives us the design and implementation knowledge. To conclude we have added our conclusions and further enhancement. We have also mentioned the references to our project. Last but not the least we have the screen shots of our output showing execution of our program in appendix.

CHAPTER 2

LITERATURE SURVEY

2.1. Problem Definition

newsMag is an application which is helpful for the people to read news on the go with the busy life. The application helps them by providing the news on short. There are various activities like getting started to the app, showing the news articles, their category, sharing the app, about the app and the user can give feedback.

2.2. Purpose & Scope

In the application we can view various news with respect to their category, we can share app, view about the app, contact the helpline, send mail to the developers and provide feedback. Another advantage of the application is that people can news in a particular category and it is very easy to view them. Our application has several advantages.

Advantages:

i. User friendly interface
ii. Fast access to news
iii. Less error
iv. Look and Feel Environment

2.3. Aim of the application

Our news application, titled newsMag is to help the user by providing news in short with their on the go busy schedule life so that they aren't left behind in getting updated to what is happening around them. As is difficult to read pages of news each day and is very time consuming newsMag plays a vital role in making the users aware of the current affairs in short. The users can also read the news in detailed if interested by clicking on the news view shown using our application.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1. Introduction

3.1.1. Purpose

In the application we can view various news with respect to their category, we can share app, view about the app, contact the helpline, send mail to the developers and provide feedback. Another advantage of the application is that people can news in a particular category and it is very easy to view them.

3.1.2. Scope

For people with busy schedule in life who cannot have time to study complete newspaper can use this application so that they do not miss to have knowledge of what is happening around them.

3.1.3. Definition, Acronyms and Abbreviations

1. XML - Extensible Markup Language
2. MS – Microsoft
3. IDE - Integrated Development Environment
4. SDK – Software Development Kit

3.2. Development Environment

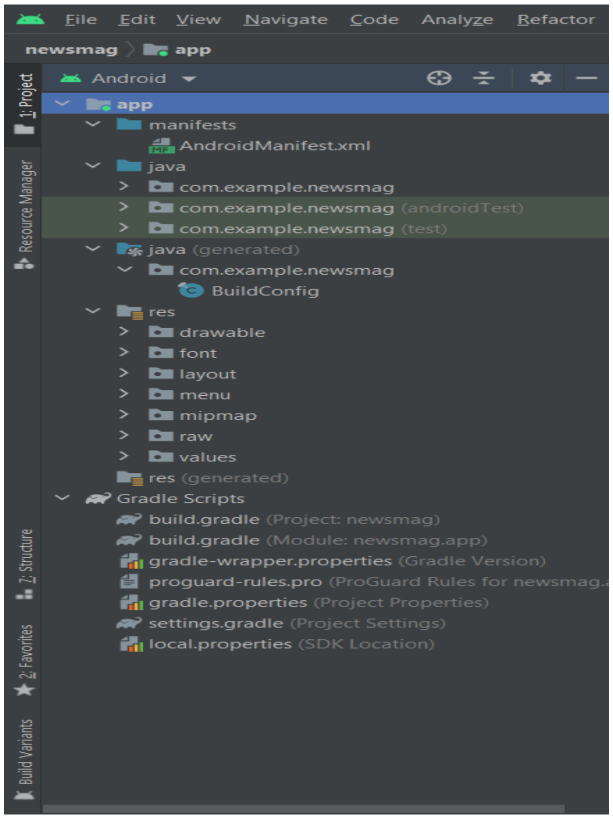
3.2.1. Android Programming Languages

In Android, programming is done in two languages JAVA or C++ and XML (Extension Markup Language). Nowadays KOTLIN is also preferred. The XML file deals with the design, presentation, layouts, blueprint, etc. (as a front-end) while the JAVA or KOTLIN deals with theworking of buttons, variables, storing, etc. (as a back-end).

3.2.2. Android Components

- **Activities**: It deals with the UI and the user interactions to the screen. In other words, it is a User Interface that contains activities.
- **Services**: Services are the background actions performed by the app; these might be long- running operations. A service might need other sub-services so as to perform specific tasks.
- **Content Provider:** Content Provider is used to transferring the data from one application to the others at the request of the other application.
- **Broadcast Receivers:** A Broadcast is used to respond to messages from other applications or from the System.

3.2.3. Structural Layout of Android



```
app
  manifests
    AndroidManifest.xml
  java
    com.example.newsmag
    com.example.newsmag (androidTest)
    com.example.newsmag (test)
  java (generated)
    com.example.newsmag
      BuildConfig
  res
    drawable
    font
    layout
    menu
    mipmap
    raw
    values
  res (generated)
Gradle Scripts
  build.gradle (Project: newsmag)
  build.gradle (Module: newsmag.app)
  gradle-wrapper.properties (Gradle Version)
  proguard-rules.pro (ProGuard Rules for newsmag.
  gradle.properties (Project Properties)
  settings.gradle (Project Settings)
  local.properties (SDK Location)
```

Figure 3.2.3 Structural Layout of Android Studio

- **Manifest Folder**: Android Manifest is an XML file that is the root of the project source set. It describes the essential information about the app and the Android build tools, the Android Operating System, and Google Play. It contains the permission that an app might need in order to perform a specific task. It also contains the Hardware and the Software features of the app, which determines the compatibility of an app on the Play Store.

- **Java Folder:** The JAVA folder consists of the java files that are required to perform the background task of the app. It consists of the functionality of the buttons, calculation, storing, variables, toast (small popup message), programming function, etc. The number of these files depends upon the type of activities created.

- **Resource Folder:** The res or Resource folder consists of the various resources that are used in the app. This consists of sub-folders like drawable, layout, mipmap, raw, and values. The drawable consists of the images. The layout consists of the XML files that define the user interface layout. These are stored in res.layout and are accessed as R.layout class. The raw consists of the Resources files like audio files or music files, etc. These are accessed through R.raw.filename.values are used to store the hardcoded strings (considered safe to store string values) values, integers, and colors.

- **Gradle Files:** Gradle is an advanced toolkit, which is used to manage the build process that drawable consists of the images. The layout consists of the XML files that define the user interface layout. These are stored in res.layout and are accessed as R.layout class. The raw consists of the Resources files like audio files or music files, etc. These are accessed through R.raw.filename.values are used to store the hardcoded strings (considered safe to store string values) values, integers, and colors.

3.3. Specific Requirements

3.3.1 Software Requirements

i. Windows 10 Operating System

ii. **Tool kit**: Android SDK (Software development kit), Java development kit (JDK)

iii. **IDE**: Android Studio

3.3.2. Hardware Requirements

i.1.8 GHz Processor

ii.8GB (IDE + Android SDK + Android Emulator)/ 4GB (minimum) RAM

iii.15 inches Monitor

iv.104 keys with keyboard and mouse

v.6.3 inches Physical Android Device

CHAPTER 4

DESIGN

4.1. Project Flow

In our project newsMag, we have used XML with different attributes and a code written in Java language for various activities.

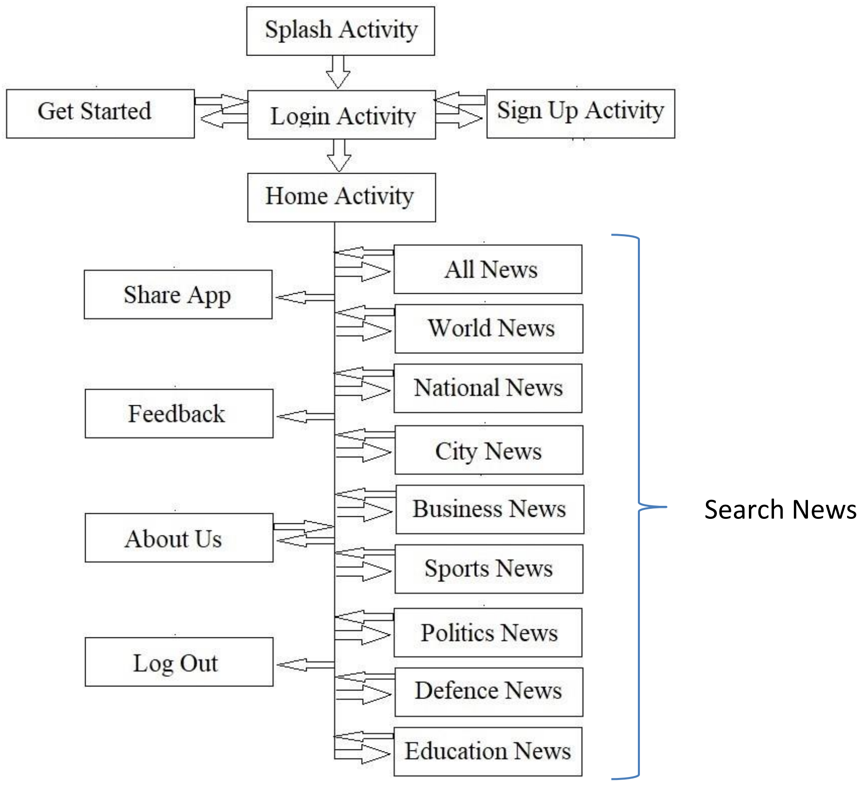


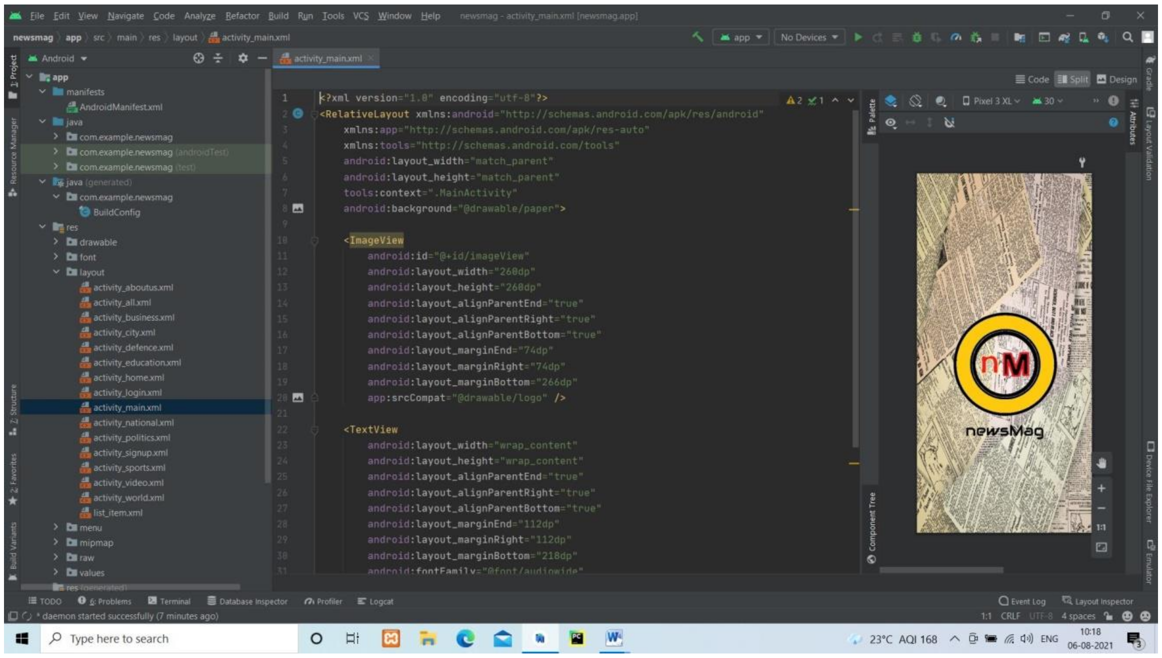
Figure 4.1 Flow diagram of newsMag application

Once the user opens the application on the android device, splash screen with the app logo is displayed for few seconds and is navigated automatically to the login activity. The user can register in case if not registered using the 'Sign Up' option provided and if the user is new to the app they can get the user manual by clicking on the 'Get Started' button provided where one can view the video get played. On successful login the 'Home Activity' Dashboard is displayed, the Home Activity has various categories of news based on users interest. The users are provided with few features on the navigation drawer through which one can 'Share App' through external messaging apps, provide ‘Feedback’ by filling a google form, view about the app through ‘About Us’ and can contact the helpline number provided and send mail to the technical team which gets connected to external mail services. The users can Log Out on clicking ‘LOG OUT’ option provided on the navigation drawer to exit from the app.

4.2. Design using XML

4.2.1 activity_main.xml

The application on opening displays the splash screen which is designed in the activity_main.xml.

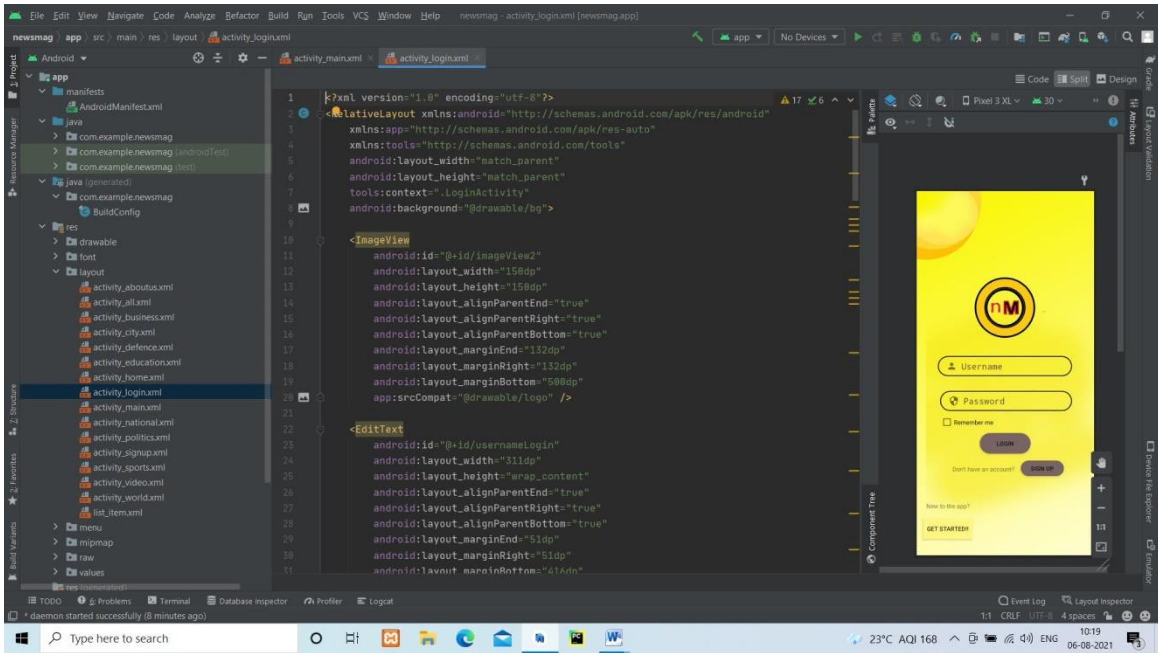


```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="@drawable/paper">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="260dp"
        android:layout_height="260dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="74dp"
        android:layout_marginRight="74dp"
        android:layout_marginBottom="266dp"
        app:srcCompat="@drawable/logo" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="112dp"
        android:layout_marginRight="112dp"
        android:layout_marginBottom="218dp"
        android:fontFamily="@font/audiowide"
```

Figure 4.2.1 Design Code screenshot for splash screen

4.2.2 activity_login.xml

The login page is designed in the activity_login.xml where the user can login by entering the registered username and password.

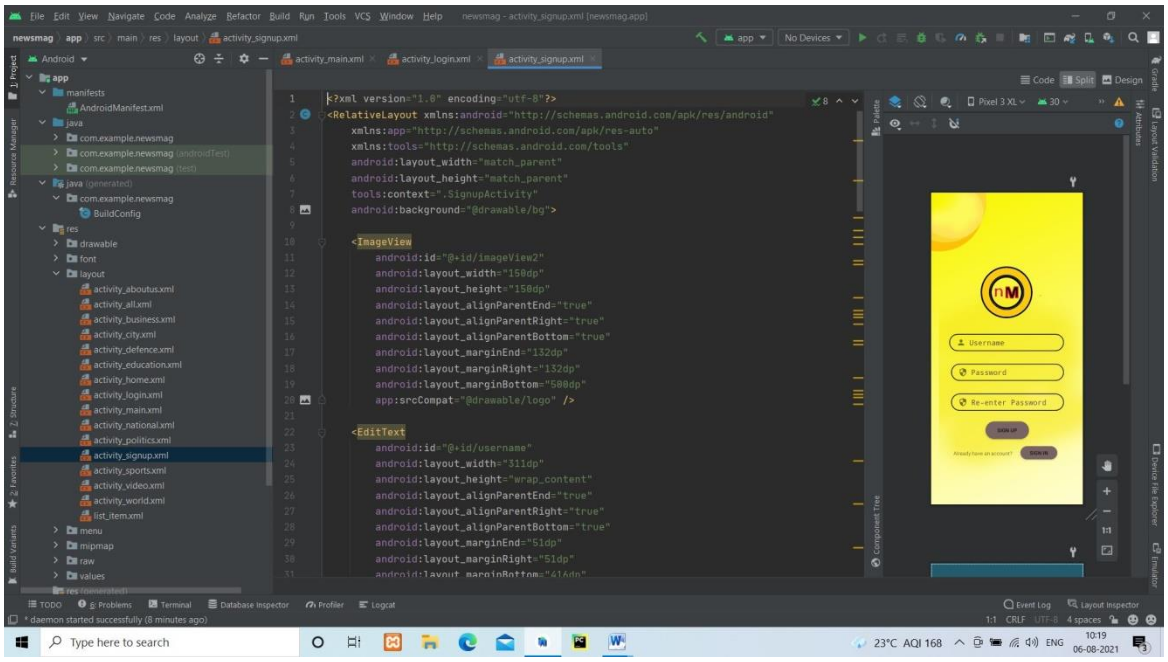


```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".LoginActivity"
    android:background="@drawable/bg">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="132dp"
        android:layout_marginRight="132dp"
        android:layout_marginBottom="500dp"
        app:srcCompat="@drawable/logo" />

    <EditText
        android:id="@+id/usernameLogin"
        android:layout_width="311dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="51dp"
        android:layout_marginRight="51dp"
        android:layout_marginBottom="416dp"
```

Figure 4.2.2 Design Code screenshot for Login

4.2.3 activity_signup.xml

The user can register through register screen which is designed in the activity_signup.xml.

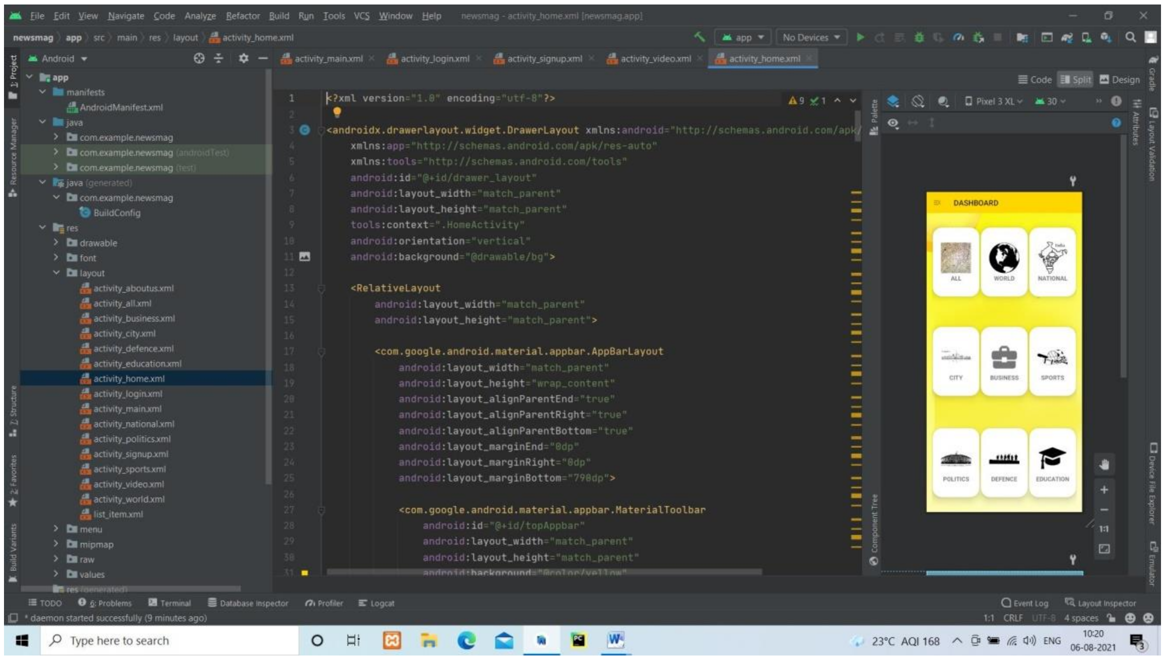


```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SignupActivity"
    android:background="@drawable/bg">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="132dp"
        android:layout_marginRight="132dp"
        android:layout_marginBottom="500dp"
        app:srcCompat="@drawable/logo" />

    <EditText
        android:id="@+id/username"
        android:layout_width="311dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="51dp"
        android:layout_marginRight="51dp"
        android:layout_marginBottom="416dp"
```

Figure 4.2.3 Design Code screenshot for Signup

4.2.4 activity_home.xml

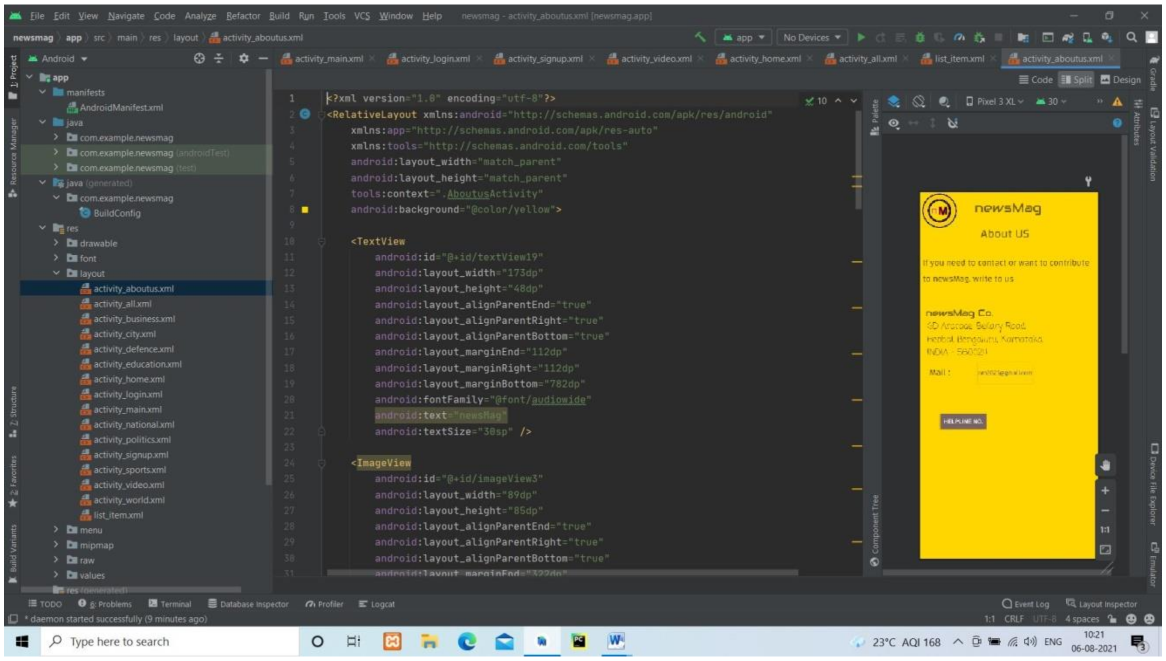
The activity_home.xml has various card views to display the news category and also contains the navigation drawer.



```
<?xml version="1.0" encoding="utf-8"?>

<androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/
    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"
    tools:context=".HomeActivity"
    android:orientation="vertical"
    android:background="@drawable/bg">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <com.google.android.material.appbar.AppBarLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true"
            android:layout_alignParentBottom="true"
            android:layout_marginEnd="0dp"
            android:layout_marginRight="0dp"
            android:layout_marginBottom="790dp">

            <com.google.android.material.appbar.MaterialToolbar
                android:id="@+id/topAppbar"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:background="@color/yellow"
```

Figure 4.2.4 Design Code screenshot of Home Dashboard

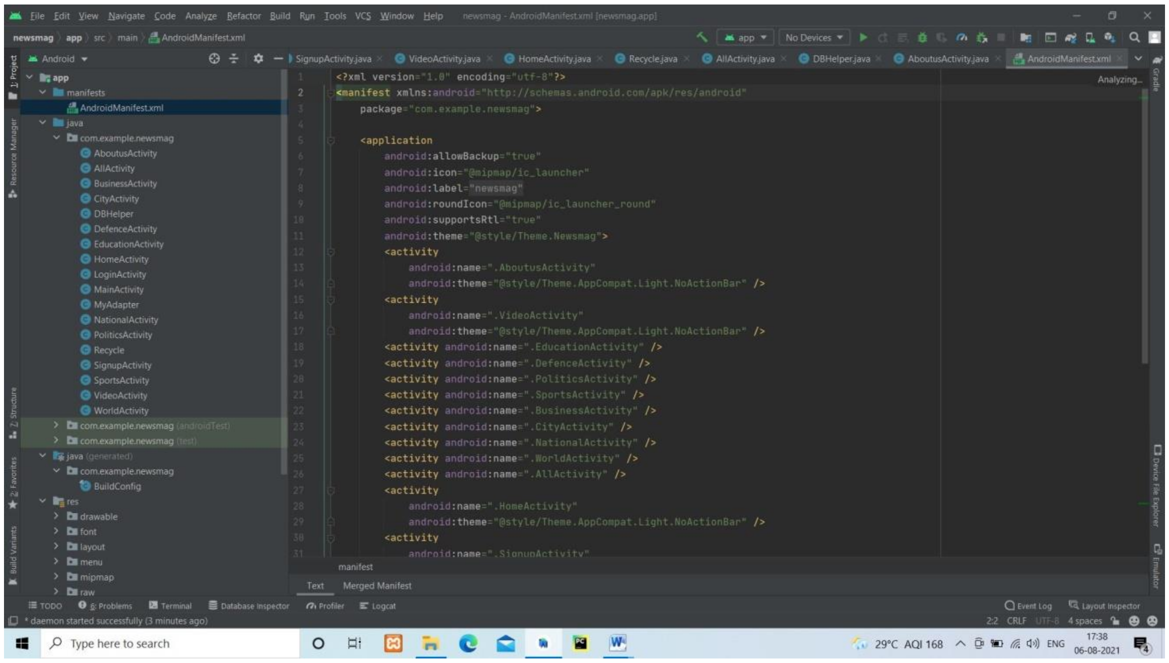
4.2.5 activity_aboutus.xml



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".AboutusActivity"
    android:background="@color/yellow">

    <TextView
        android:id="@+id/textView19"
        android:layout_width="173dp"
        android:layout_height="48dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="112dp"
        android:layout_marginRight="112dp"
        android:layout_marginBottom="782dp"
        android:fontFamily="@font/audiowide"
        android:text="newsMag"
        android:textSize="30sp" />

    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="89dp"
        android:layout_height="85dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="322dp"
```

Figure 4.2.5 Design Code screenshot of About us

4.2.6 AndroidManifest.xml

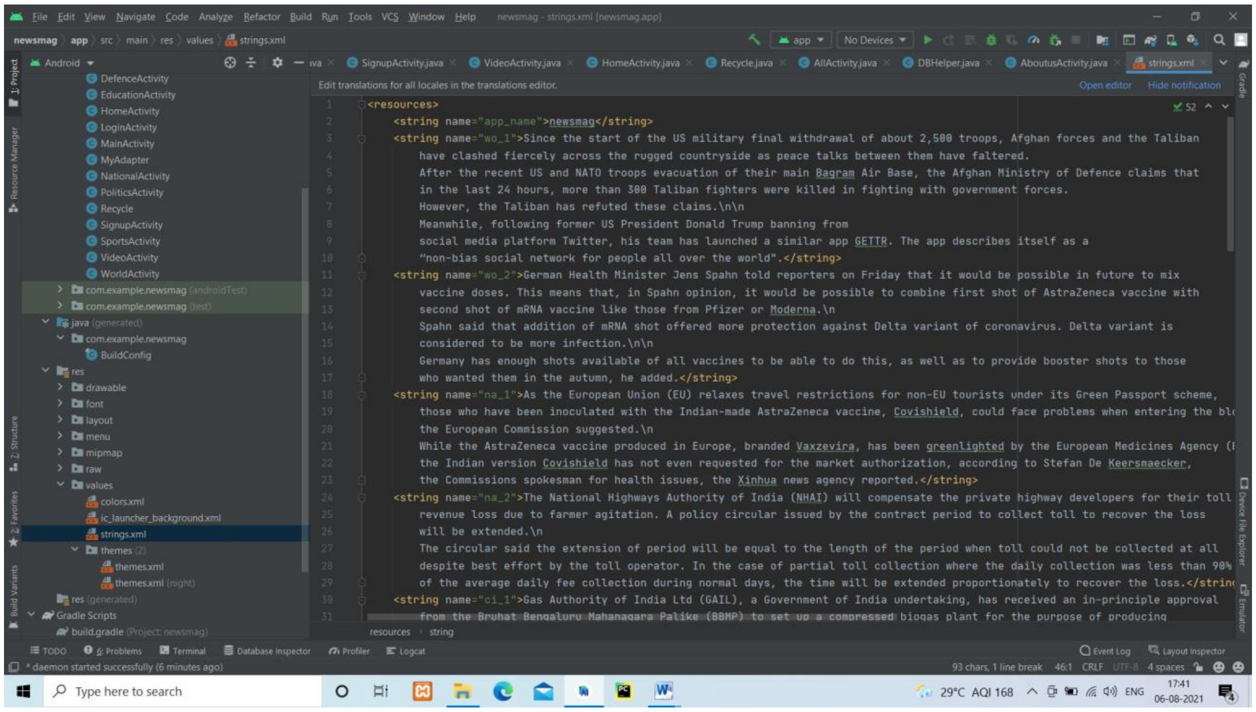


```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.newsmag">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="newsmag"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Newsmag">
        <activity
            android:name=".AboutusActivity"
            android:theme="@style/Theme.AppCompat.Light.NoActionBar" />
        <activity
            android:name=".VideoActivity"
            android:theme="@style/Theme.AppCompat.Light.NoActionBar" />
        <activity android:name=".EducationActivity" />
        <activity android:name=".DefenceActivity" />
        <activity android:name=".PoliticsActivity" />
        <activity android:name=".SportsActivity" />
        <activity android:name=".BusinessActivity" />
        <activity android:name=".CityActivity" />
        <activity android:name=".NationalActivity" />
        <activity android:name=".WorldActivity" />
        <activity android:name=".AllActivity" />
        <activity
            android:name=".HomeActivity"
            android:theme="@style/Theme.AppCompat.Light.NoActionBar" />
        <activity
            android:name=".SignupActivity"
```

Figure 4.2.6 Code screenshot of Android Manifest

4.2.7 strings.xml

The string file is used to initialized strings and fetch using it's id.



```xml
<resources>
    <string name="app_name">newsmag</string>
    <string name="wo_1">Since the start of the US military final withdrawal of about 2,500 troops, Afghan forces and the Taliban
        have clashed fiercely across the rugged countryside as peace talks between them have faltered.
        After the recent US and NATO troops evacuation of their main Bagram Air Base, the Afghan Ministry of Defence claims that
        in the last 24 hours, more than 300 Taliban fighters were killed in fighting with government forces.
        However, the Taliban has refuted these claims.\n\n
        Meanwhile, following former US President Donald Trump banning from
        social media platform Twitter, his team has launched a similar app GETTR. The app describes itself as a
        “non-bias social network for people all over the world".</string>
    <string name="wo_2">German Health Minister Jens Spahn told reporters on Friday that it would be possible in future to mix
        vaccine doses. This means that, in Spahn opinion, it would be possible to combine first shot of AstraZeneca vaccine with
        second shot of mRNA vaccine like those from Pfizer or Moderna.\n
        Spahn said that addition of mRNA shot offered more protection against Delta variant of coronavirus. Delta variant is
        considered to be more infection.\n\n
        Germany has enough shots available of all vaccines to be able to do this, as well as to provide booster shots to those
        who wanted them in the autumn, he added.</string>
    <string name="na_1">As the European Union (EU) relaxes travel restrictions for non-EU tourists under its Green Passport scheme,
        those who have been inoculated with the Indian-made AstraZeneca vaccine, Covishield, could face problems when entering the bl
        the European Commission suggested.\n
        While the AstraZeneca vaccine produced in Europe, branded Vaxzevira, has been greenlighted by the European Medicines Agency (
        the Indian version Covishield has not even requested for the market authorization, according to Stefan De Keersmaecker,
        the Commissions spokesman for health issues, the Xinhua news agency reported.</string>
    <string name="na_2">The National Highways Authority of India (NHAI) will compensate the private highway developers for their toll
        revenue loss due to farmer agitation. A policy circular issued by the contract period to collect toll to recover the loss
        will be extended.\n
        The circular said the extension of period will be equal to the length of the period when toll could not be collected at all
        despite best effort by the toll operator. In the case of partial toll collection where the daily collection was less than 90%
        of the average daily fee collection during normal days, the time will be extended proportionately to recover the loss.</strin
    <string name="ci_1">Gas Authority of India Ltd (GAIL), a Government of India undertaking, has received an in-principle approval
        from the Bruhat Bengaluru Mahanagara Palike (BBMP) to set up a compressed biogas plant for the purpose of producing
```

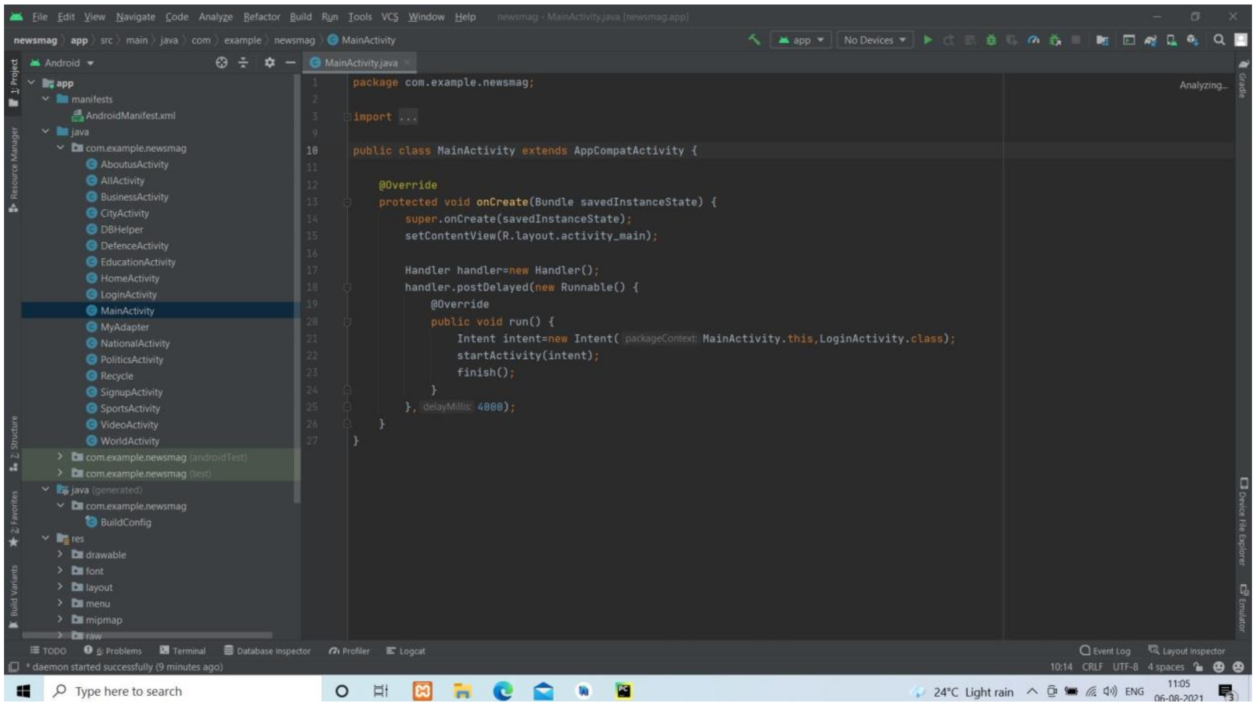
Figure 4.2.7 Design Code screenshot of Strings

CHAPTER 5

IMPLEMENTATION

5.1. MainActivity.java

The MainActivity.java file contains the splash screen where the logo is displayed for 4 seconds and is parsed to the login screen.

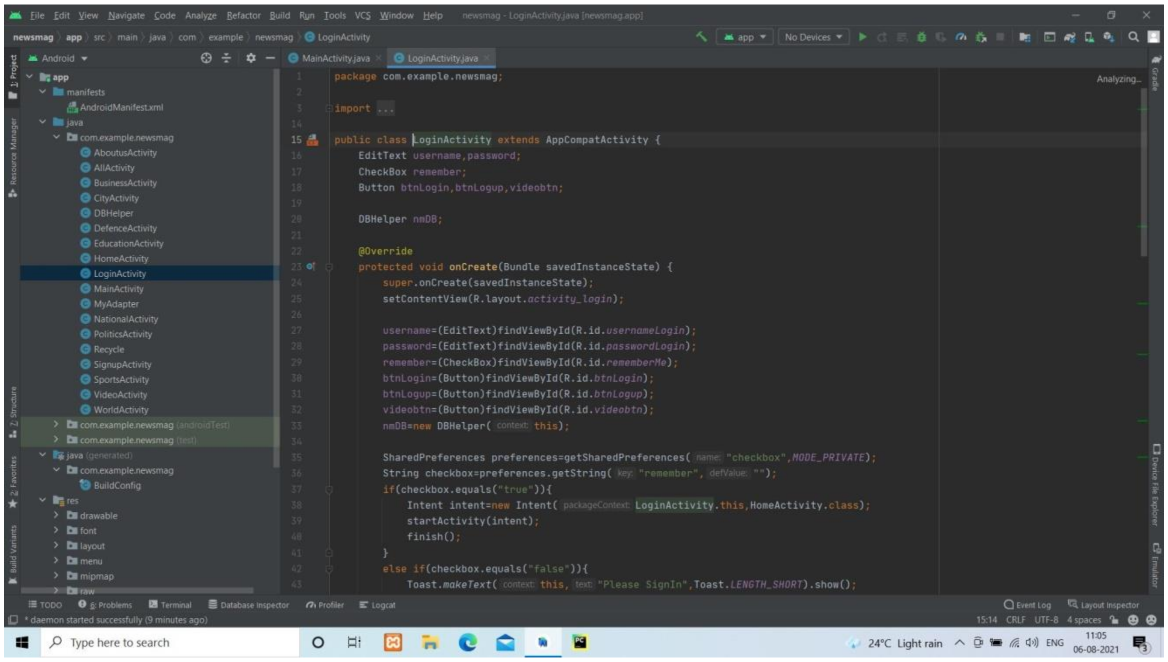


```java
package com.example.newsmag;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Handler handler=new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent=new Intent( packageContext: MainActivity.this,LoginActivity.class);
                startActivity(intent);
                finish();
            }
        }, delayMillis: 4000);
    }
}
```

Figure 5.1 Code screenshot for MainActivity.java

5.2. LoginActivity.java

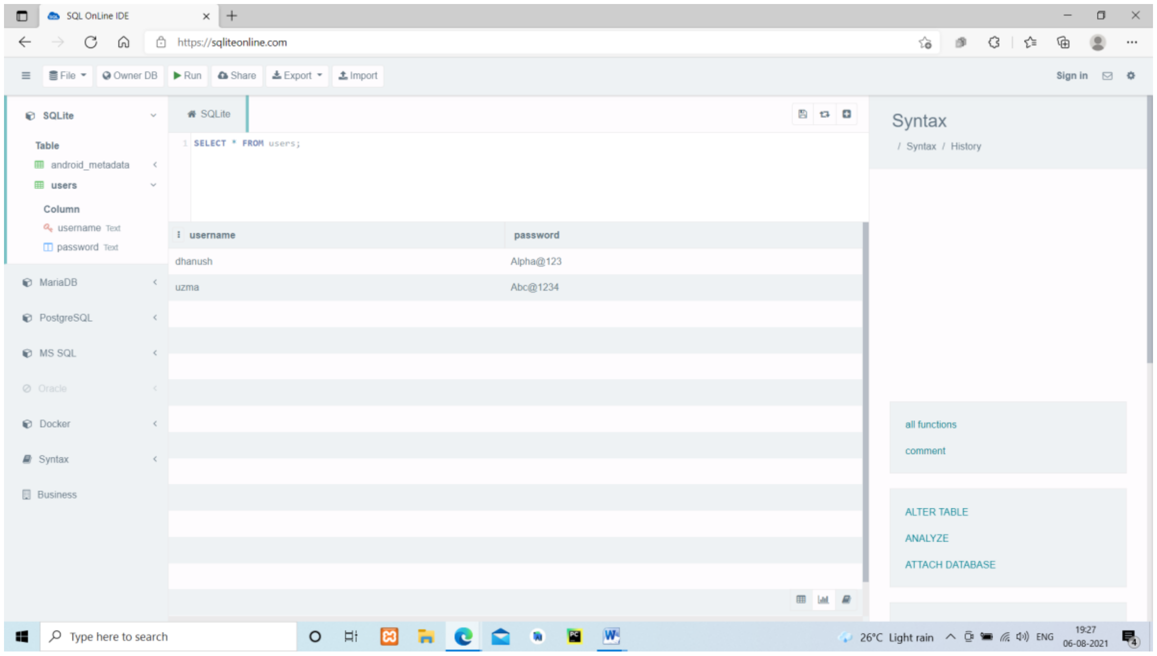
The Login screen contains the code to parse to sign up, get started and the home activity on successful login. Login credentials are connected to the DBHelper java file to save the credentials in the database.



```java
package com.example.newsmag;

import ...

public class LoginActivity extends AppCompatActivity {
    EditText username,password;
    CheckBox remember;
    Button btnLogin,btnLogup,videobtn;

    DBHelper nmDB;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        username=(EditText)findViewById(R.id.usernameLogin);
        password=(EditText)findViewById(R.id.passwordLogin);
        remember=(CheckBox)findViewById(R.id.rememberMe);
        btnLogin=(Button)findViewById(R.id.btnLogin);
        btnLogup=(Button)findViewById(R.id.btnLogup);
        videobtn=(Button)findViewById(R.id.videobtn);
        nmDB=new DBHelper( context: this);

        SharedPreferences preferences=getSharedPreferences( name: "checkbox",MODE_PRIVATE);
        String checkbox=preferences.getString( key: "remember", defValue: "");
        if(checkbox.equals("true")){
            Intent intent=new Intent( packageContext: LoginActivity.this,HomeActivity.class);
            startActivity(intent);
            finish();
        }
        else if(checkbox.equals("false")){
            Toast.makeText( context: this, text: "Please SignIn",Toast.LENGTH_SHORT).show();
```

Figure 5.2 Code screenshot for LoginActivity.java

5.3. Online SQLite DB viewer



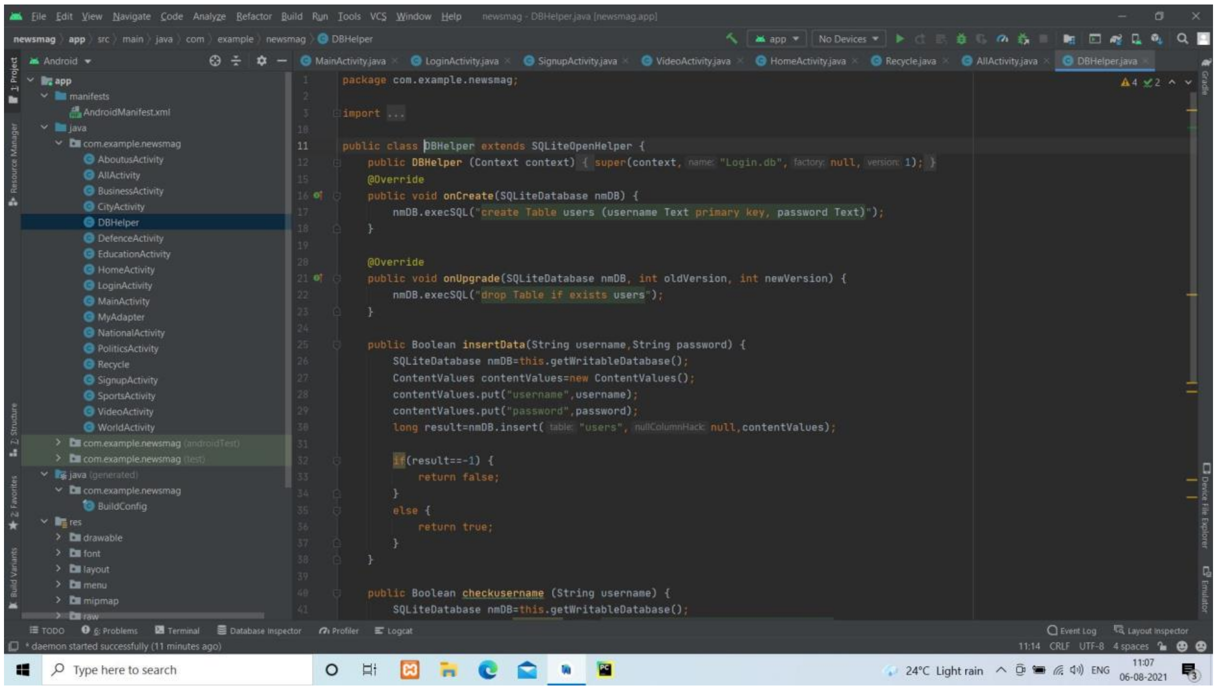
```sql
SELECT * FROM users;
```

username	password
dhanush	Alpha@123
uzma	Abc@1234

Figure 5.3 Screenshot showing the username and the password details

5.4. DBHelper.java

The DBHelper.java file helps in connecting the Login and check validity of the credentials.

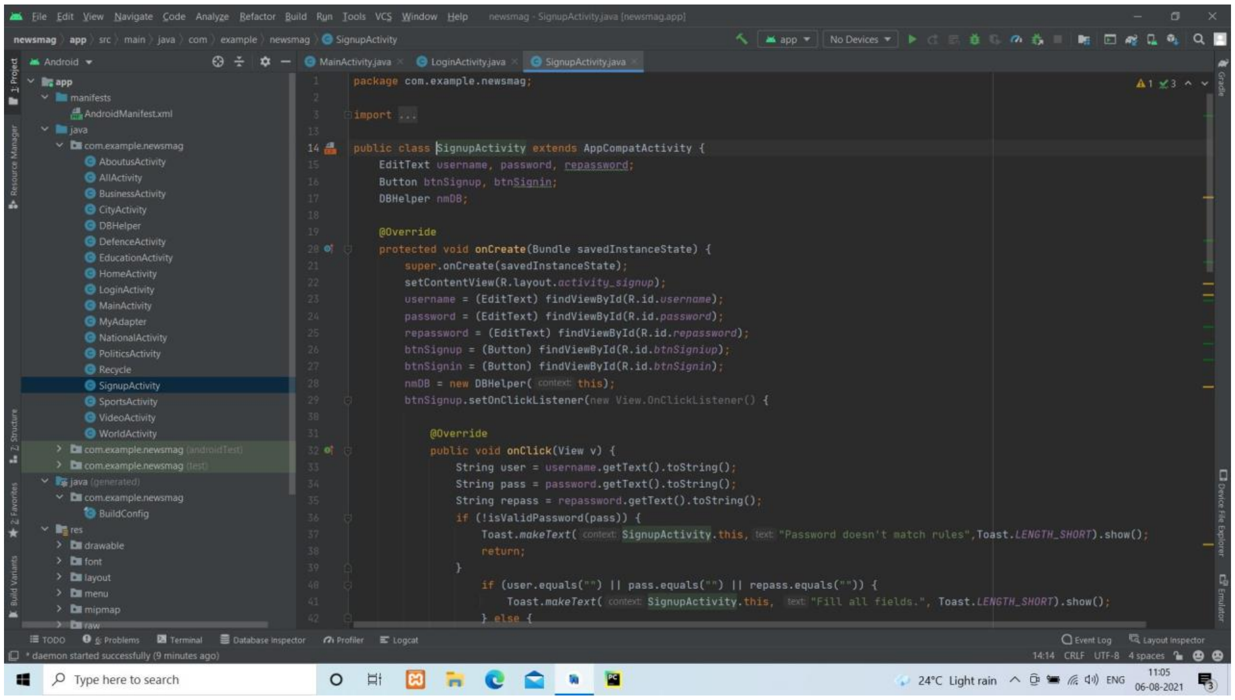


```java
package com.example.newsmag;

import ...

public class DBHelper extends SQLiteOpenHelper {
    public DBHelper (Context context) { super(context, name: "Login.db", factory: null, version: 1); }
    @Override
    public void onCreate(SQLiteDatabase nmDB) {
        nmDB.execSQL("create Table users (username Text primary key, password Text)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase nmDB, int oldVersion, int newVersion) {
        nmDB.execSQL("drop Table if exists users");
    }

    public Boolean insertData(String username,String password) {
        SQLiteDatabase nmDB=this.getWritableDatabase();
        ContentValues contentValues=new ContentValues();
        contentValues.put("username",username);
        contentValues.put("password",password);
        long result=nmDB.insert( table: "users", nullColumnHack: null,contentValues);

        if(result==-1) {
            return false;
        }
        else {
            return true;
        }
    }

    public Boolean checkusername (String username) {
        SQLiteDatabase nmDB=this.getWritableDatabase();
```

Figure 5.4 Code screenshot of DBHelper.java

5.5. SignupActivity.java

The Signup screen contains the code to parse to login screen after registration.

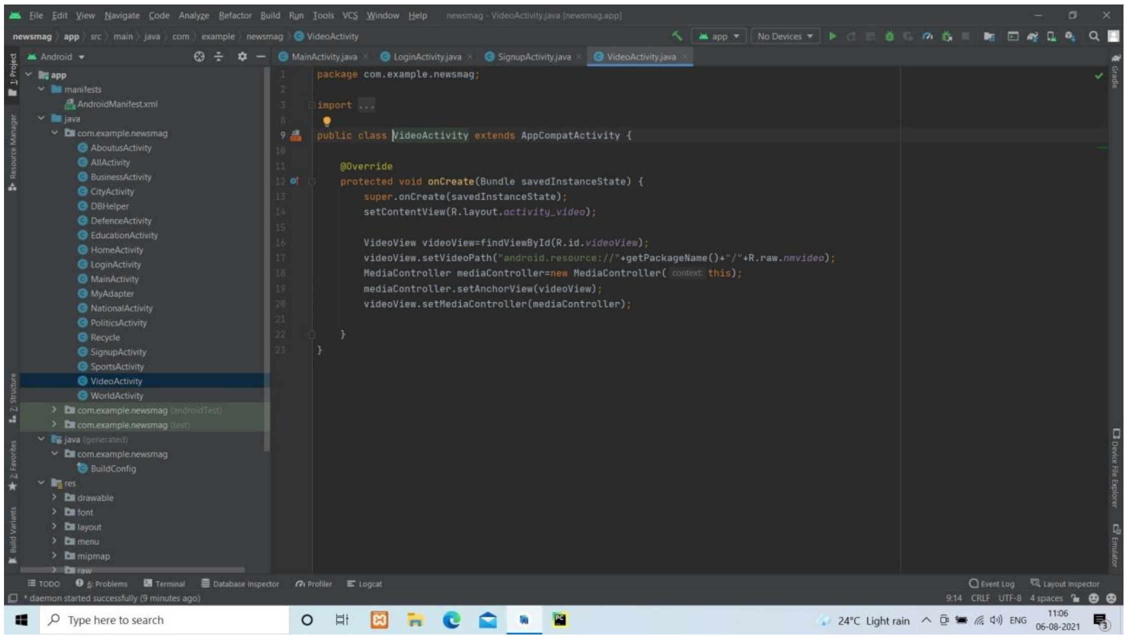


```java
package com.example.newsmag;

import ...

public class SignupActivity extends AppCompatActivity {
    EditText username, password, repassword;
    Button btnSignup, btnSignin;
    DBHelper nmDB;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
        username = (EditText) findViewById(R.id.username);
        password = (EditText) findViewById(R.id.password);
        repassword = (EditText) findViewById(R.id.repassword);
        btnSignup = (Button) findViewById(R.id.btnSigniup);
        btnSignin = (Button) findViewById(R.id.btnSignin);
        nmDB = new DBHelper( context: this);
        btnSignup.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                String user = username.getText().toString();
                String pass = password.getText().toString();
                String repass = repassword.getText().toString();
                if (!isValidPassword(pass)) {
                    Toast.makeText( context: SignupActivity.this, text: "Password doesn't match rules",Toast.LENGTH_SHORT).show();
                    return;
                }
                    if (user.equals("") || pass.equals("") || repass.equals("")) {
                        Toast.makeText( context: SignupActivity.this,  text: "Fill all fields.", Toast.LENGTH_SHORT).show();
                    } else {
```

Figure 5.5 Code screenshot for SignupActivity.java

5.6. VideoActivity.java

VideoActivity.java is used to play the get started video in the application.

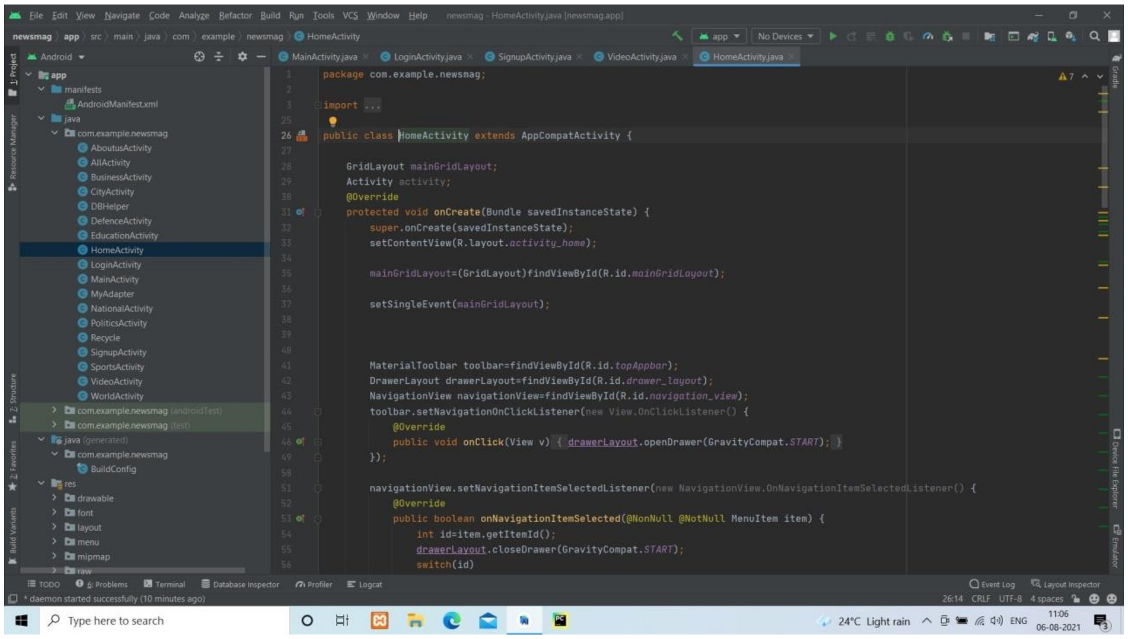


```
package com.example.newsmag;

import ...

public class VideoActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_video);

        VideoView videoView=findViewById(R.id.videoView);
        videoView.setVideoPath("android.resource://"+getPackageName()+"/"+R.raw.nmvideo);
        MediaController mediaController=new MediaController( context: this);
        mediaController.setAnchorView(videoView);
        videoView.setMediaController(mediaController);

    }
}
```

Figure 5.6 Code screenshot for ActivityVideo.java

5.7. HomeActivity.java

The HomeActivity is the heart of the application which is connected to all the activities of various news and also with the options provided in the navigation drawer which includes Share App, Feedback, About us features.

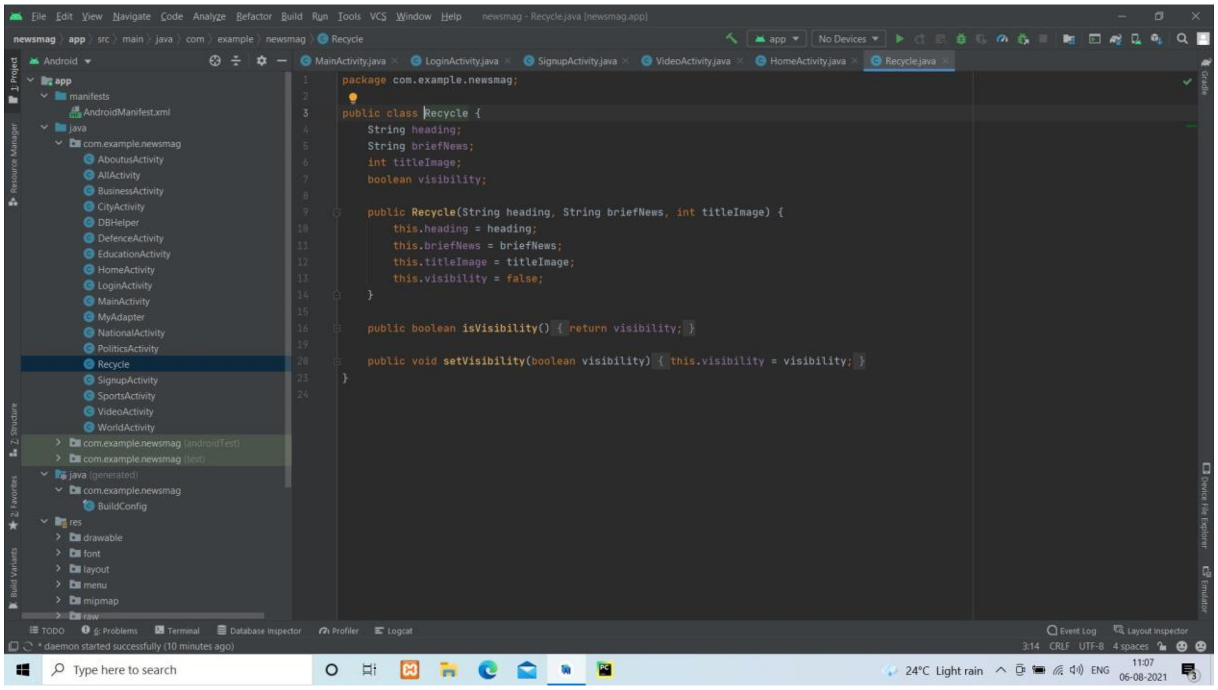


```
package com.example.newsmag;

import ...

public class HomeActivity extends AppCompatActivity {

    GridLayout mainGridLayout;
    Activity activity;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);

        mainGridLayout=(GridLayout)findViewById(R.id.mainGridLayout);

        setSingleEvent(mainGridLayout);



        MaterialToolbar toolbar=findViewById(R.id.topAppbar);
        DrawerLayout drawerLayout=findViewById(R.id.drawer_layout);
        NavigationView navigationView=findViewById(R.id.navigation_view);
        toolbar.setNavigationOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { drawerLayout.openDrawer(GravityCompat.START); }
        });

        navigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull @NotNull MenuItem item) {
                int id=item.getItemId();
                drawerLayout.closeDrawer(GravityCompat.START);
                switch(id)
```

Figure 5.7 Code screenshot for HomeActivity.java

5.8. Recycler.java

Recycler.java is used to specify what all entries should be displayed in the Recycler views.

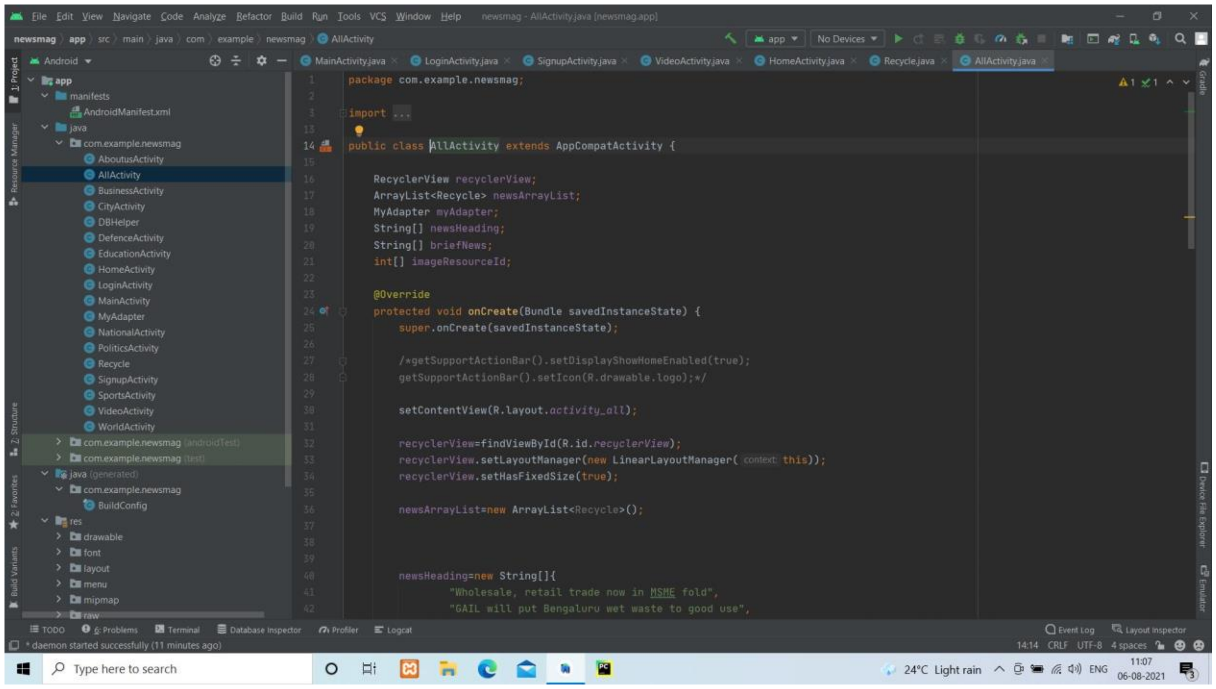


```java
package com.example.newsmag;

public class Recycle {
    String heading;
    String briefNews;
    int titleImage;
    boolean visibility;

    public Recycle(String heading, String briefNews, int titleImage) {
        this.heading = heading;
        this.briefNews = briefNews;
        this.titleImage = titleImage;
        this.visibility = false;
    }

    public boolean isVisibility() { return visibility; }

    public void setVisibility(boolean visibility) { this.visibility = visibility; }
}
```

Figure 5.8 Code screenshot for Recycler.java file

5.9. AllActivity.java

The various news categories are created similar to the AllActivity.java under recycler view.

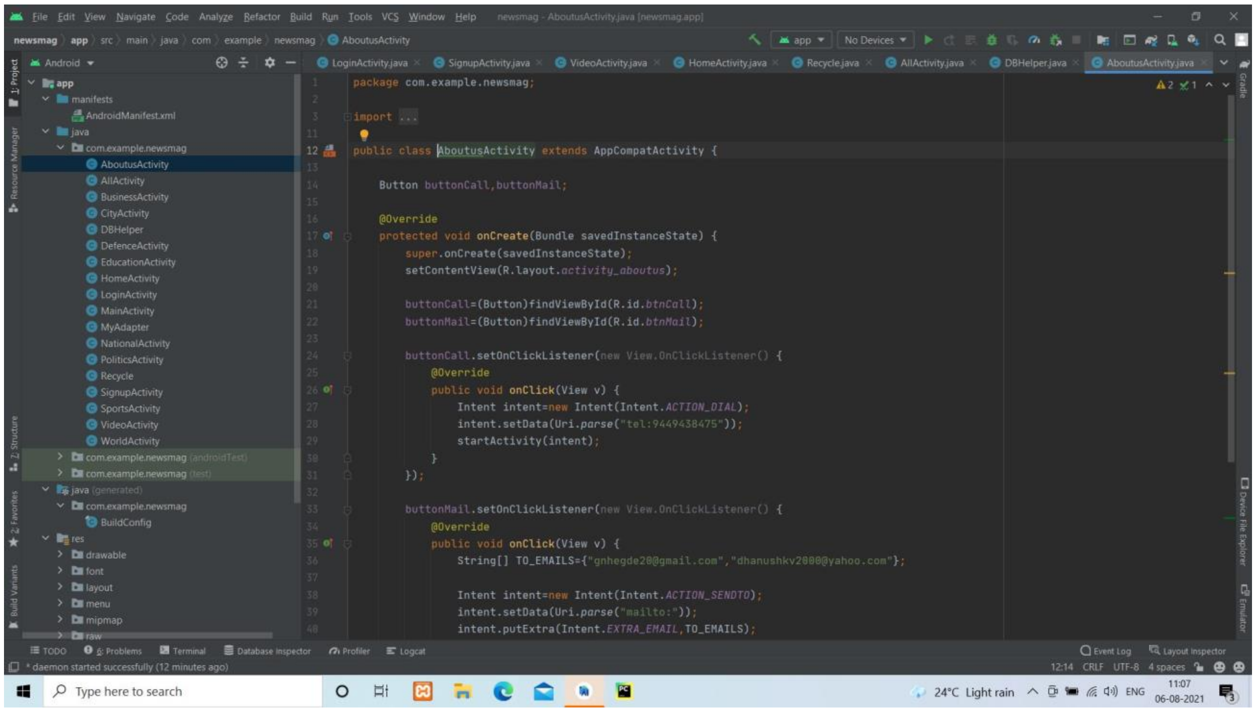


```java
package com.example.newsmag;

import ...

public class AllActivity extends AppCompatActivity {

    RecyclerView recyclerView;
    ArrayList<Recycle> newsArrayList;
    MyAdapter myAdapter;
    String[] newsHeading;
    String[] briefNews;
    int[] imageResourceId;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        /*getSupportActionBar().setDisplayShowHomeEnabled(true);
        getSupportActionBar().setIcon(R.drawable.logo);*/

        setContentView(R.layout.activity_all);

        recyclerView=findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager( context: this));
        recyclerView.setHasFixedSize(true);

        newsArrayList=new ArrayList<Recycle>();



        newsHeading=new String[]{
                "Wholesale, retail trade now in MSME fold",
                "GAIL will put Bengaluru wet waste to good use",
```

Figure 5.9 Code screenshot for AllActivity.java

5.10. AboutusActivity.java

The AboutusActivity.java is used to implement features of calling the helpline number and sending mail to the newsMag technical team using external mail sending applications.



```
package com.example.newsmag;

import ...

public class AboutusActivity extends AppCompatActivity {

    Button buttonCall,buttonMail;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_aboutus);

        buttonCall=(Button)findViewById(R.id.btnCall);
        buttonMail=(Button)findViewById(R.id.btnMail);

        buttonCall.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(Intent.ACTION_DIAL);
                intent.setData(Uri.parse("tel:9449438475"));
                startActivity(intent);
            }
        });

        buttonMail.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String[] TO_EMAILS={"gnhegde20@gmail.com","dhanushkv2000@yahoo.com"};

                Intent intent=new Intent(Intent.ACTION_SENDTO);
                intent.setData(Uri.parse("mailto:"));
                intent.putExtra(Intent.EXTRA_EMAIL,TO_EMAILS);
```

Figure 5.10 Code screenshot for AboutusActivity.java

CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion

Our generation relies mostly on phones to get through the day. Due to this, phones have become more of a personal assistant than a means to just communicate. Viewing news in short on the go with the busy schedule would make it easier for people. Henceforth to conclude, we've developed a mobile application based on Java using Android Studio. And for the efficient working of the ongoing system we've this application to help people to read news and stay updated easily in limited time.

Future Enhancements

i. We can use APIs to draw news from web to get displayed in the application.
ii. We can even further make it private and secured by enhancing login features.
iii. We can allow different users to write comments to news using their account which can be viewed by others.
iv. We can make it more space and resource efficient so that this application consumes lesser RAM and ROM.

REFERENCES

[1] IEEE Standard 830-1998: IEEE Recommended Practice for Software Requirements specifications.

[2] www.google.co.in

[3] Various videos from www.youtube.com

[4] https:// developer.android.com

[5] https://en.wikipedia.org/wiki

Appendix 'A' Screenshots

RESULTS

1. Main Activity (Splash Screen)

This is the screenshot of our mini-project's splash screen with logo which will be occurring for few seconds when the application is opened.

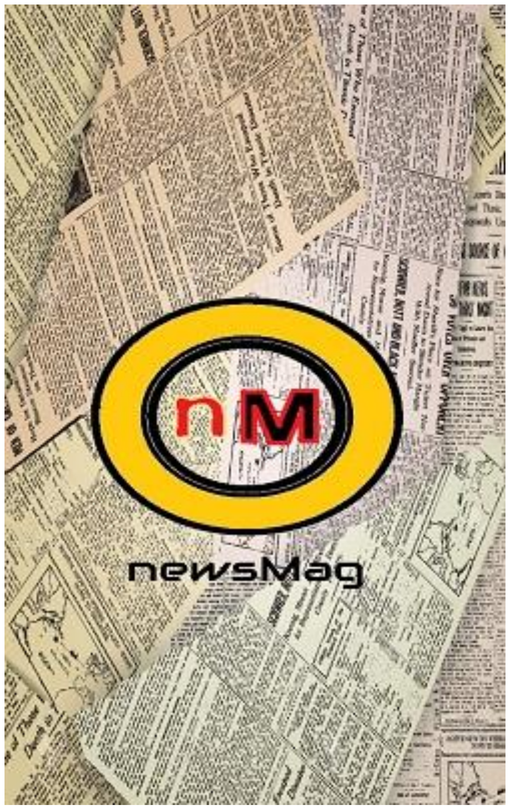


Figure A.1. Splash Screen of newsMag application

The splash screen is then directed to the Login page where the user of the application can login to view the news and use the app.

2. Login Activity

This is the screenshot of our mini-project's login screen where user can login by entering a registered username and password to use the app.

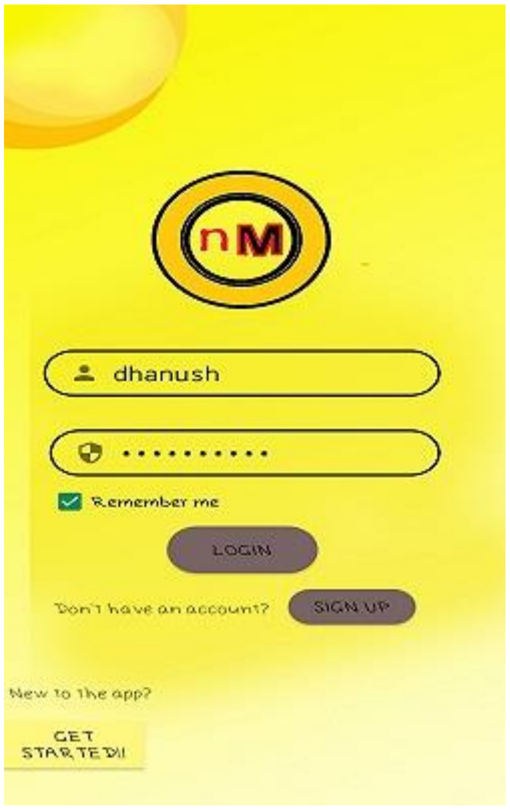


Figure A.2. Login Screen of newsMag application

The login screen is connected to the Signup Activity where the user can get registered and the Get Started Activity where user can play a video to get themselves aware on how to use the app.

3. Signup Activity

This is the screenshot of our mini-project's signup screen where user can get registered by entering a unique username and password validating the credentials to login further using the registered credentials.

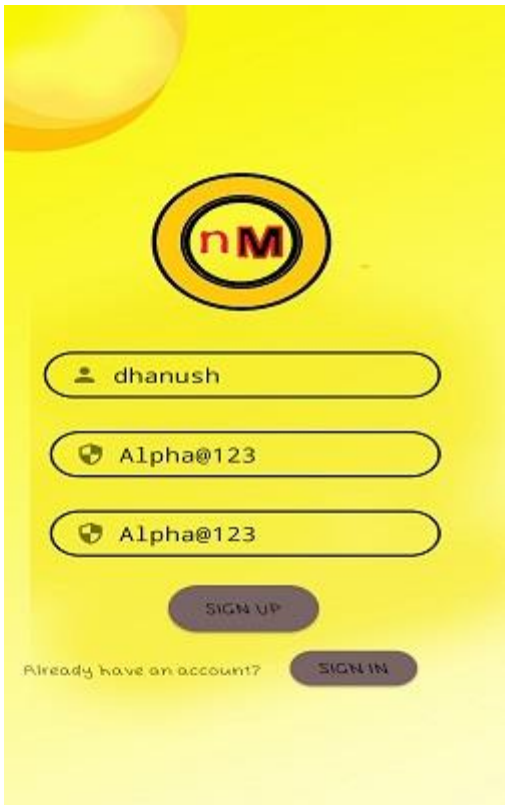


Figure A.3. Signup Screen of newsMag application

The signup screen is connected to the Login Activity where the user can login to use the application.

4. Video Activity

This is the screenshot of our mini-project's get started screen where user can play a video to get an idea on how to use the app. The video is pre-loaded by the technical team.

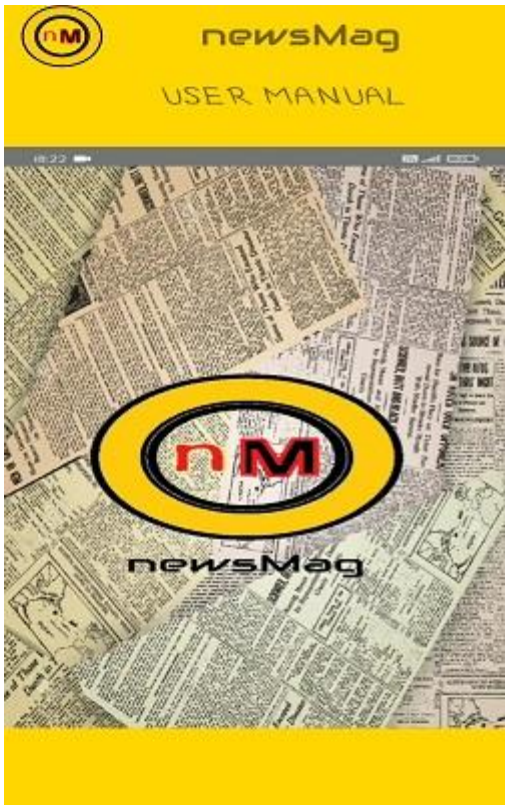


Figure A.4. Get Started Screen of newsMag application

Video in Get Started Activity can be played, paused, rewind, fast forwarded using the options associated with it. The get started screen is connected to the Login Activity on clicking back button.

5. Home Activity

This is the screenshot of our mini-project's home screen where user can select among various categories provided based on their interest.

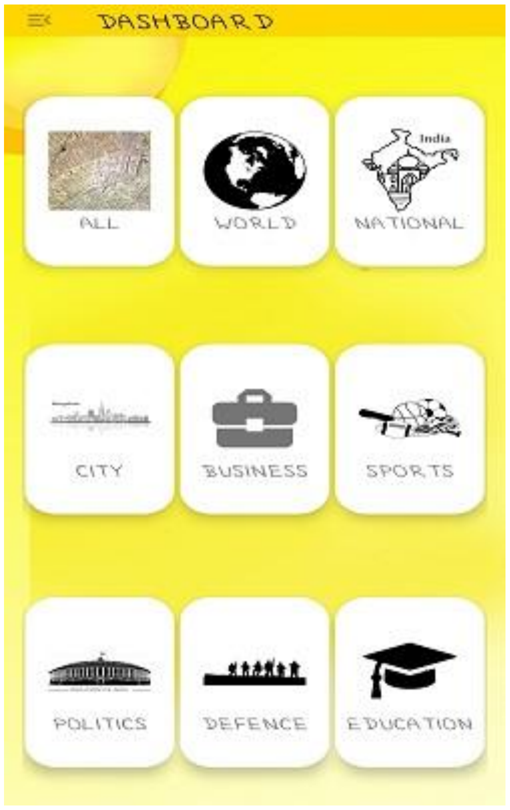


Figure A.5. Home Screen of newsMag application

The categories are displayed in the form of cards, on clicking the category the users are directed to the activities containing news articles present under that particular category.

6. Navigation Drawer in Home Activity

This is the screenshot of our mini-project's navigation drawer present in home screen where user can select among features provided in the drawer menu such as Share App, Feedback, About Us and can Log out of the app.

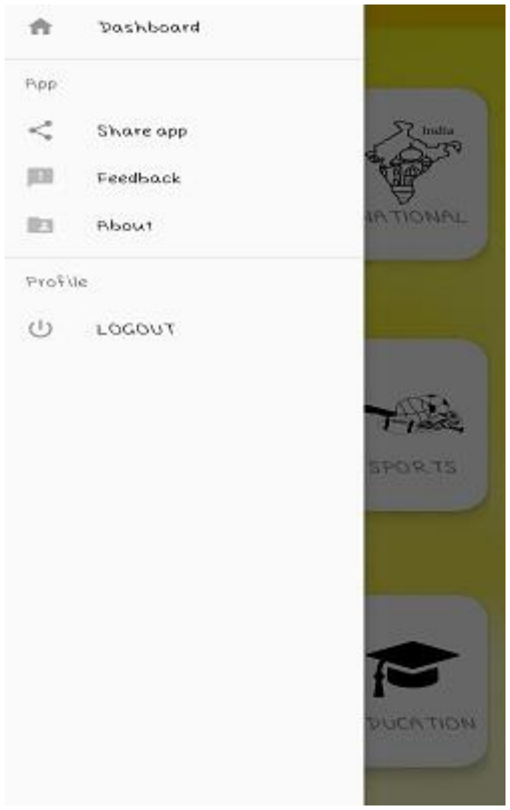
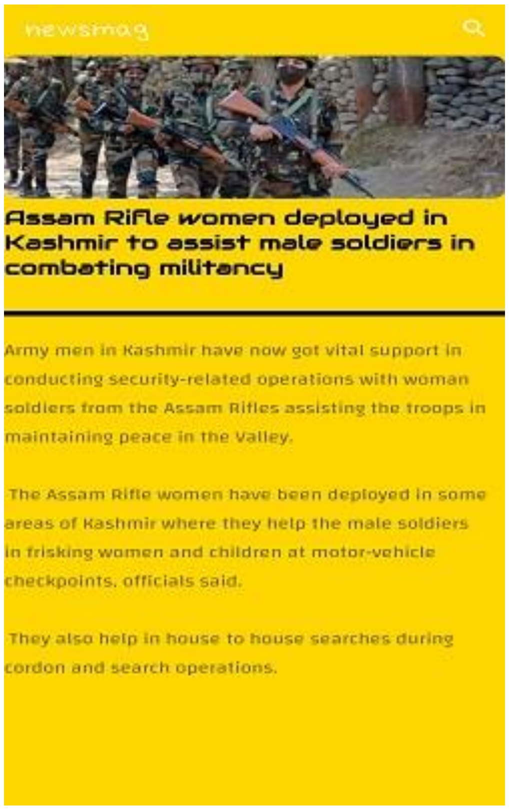


Figure A.6. Navigation Drawer in Home Screen of newsMag application

The navigation drawer is associated to the Home Activity on clicking on Dashboard, can share the newsMag application, go to about newsMag, help users provide feedback and lastly log out option to exit from the application.

7. News Activities

This is the screenshot of our mini-project’s news displaying screen where user can view the news article in brief and on clicking the news article will be able to read the news in detail.



Assam Rifle women deployed in Kashmir to assist male soldiers in combating militancy

Army men in Kashmir have now got vital support in conducting security-related operations with woman soldiers from the Assam Rifles assisting the troops in maintaining peace in the Valley.

The Assam Rifle women have been deployed in some areas of Kashmir where they help the male soldiers in frisking women and children at motor-vehicle checkpoints, officials said.

They also help in house to house searches during cordon and search operations.

Figure A.7. News displaying Screen of newsMag application

Users are also provided with the search bar where they can search for desired news to know about. The search bar filters all news with the keyword typed in the search box and matches with news and displays all the filtered news for the user.

8. Share App Activity

This is the screenshot of our mini-project's share app screen where user can share newsMag application to others using external messaging applications.

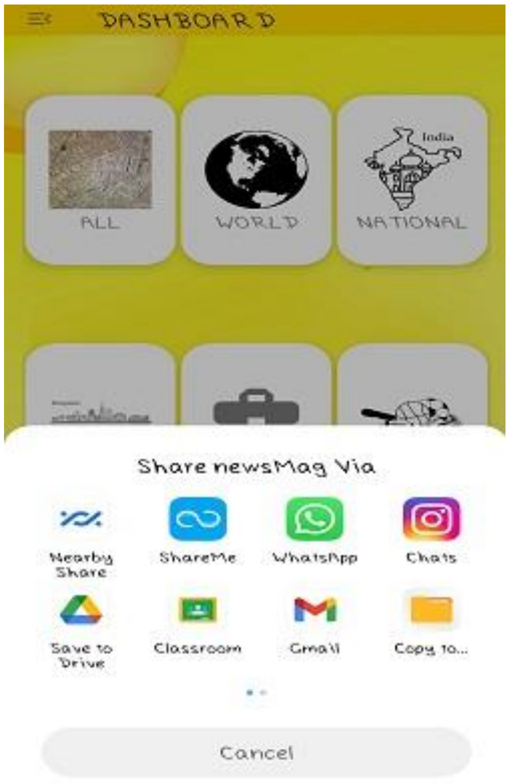


Figure A.8. Share App Screen of newsMag application

When the app is shared by a user the recipient gets a message with a google drive link using which they can download the apk file and install into their android device.

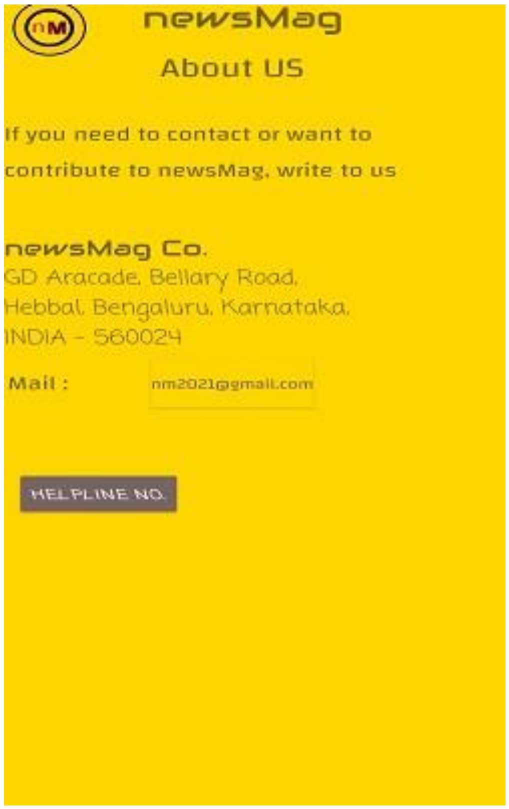
Check out newsMag, news on the go application!!

https://drive.google.com/drive/folders/1-zh3rEegg9zSSGKUIFmfyu56lnJp9VCn?usp=sharing

Use the above link for your news application

9. Aboutus Activity

This is the screenshot of our mini-project's about us screen where user can view the address of the organization.


newsMag

About US

If you need to contact or want to contribute to newsMag, write to us

newsMag Co.
GD Aracade, Bellary Road,
Hebbal, Bengaluru, Karnataka,
INDIA - 560024

Mail : nm2021@gmail.com

Figure A.9. Home Screen of newsMag application

Users are provided with option to send E-mail to the editor in case they want to contribute something or in case if they have major issues. A helpline button is also provided using which the users can contact the clustomer helpline service.

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANA SANGAMA, Belgaum - 590 014.

2020 - 2021

A
Mini project report on

"Daily Mood"

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

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

Submitted by

NAVYA SHREE PN
(1AT18IS057)

Under the guidance of
Ms. Uzma Sulthana
Assistant Professor
Dept. of ISE, ATRIA I. T.
&
Ms. Prapulla G
Assistant Professor
Dept. of ISE, ATRIA I. T.

ATRIA INSTITUTE OF TECHNOLOGY
Department of Information Science and Engineering,
Bengaluru - 560 024

ATRIA INSTITUTE OF TECHNOLOGY
(Affiliated to Visvesvaraya Technological University)
ASKB Campus, Anandnagar,
Bengaluru – 560024

Department of Information Science and Engineering

CERTIFICATE

Certified that the project work entitled ”**Daily Mood**” carried out by **Navya Shree PN** bearing USN : 1AT18IS057, the bonafide student of Department of Information Science and Engineering, Atria I. T., in partial fulfillment for the award of **Bachelor of Engineering** in Information Science & Engineering of the Visvesvaraya Technological University, Belgavi during the year 2020-2021. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Ms. Uzma Sulthana/
Ms. Prapulla G
Asst. Professor - Project Guide
Department of I.S.E.,
Atria I. T.

Dr. Shanthi Mahesh
Head of Department
Department of I.S.E.,
Atria I.T.

Dr. T.N Sreenivasa
Principal
Atria I.T.

External Viva

Name of Examiners	**Signature with date**
1.	
2.	

DECLARATION

I, Navya Shree PN (USN : 1AT18IS057), Student of sixth semester, Bachelor of Engineering, Atria Institute of Technology hereby declare that the mini project entitled **"Daily Mood"** has been carried out by me at Atria Institute of Technology, Bengaluru and submitted in partial fulfillment of the course requirements for the award of the degree of **Bachelor of Engineering** in **Information Science & Engineering** of **Visvesvaraya Technological University, Belgavi**, during the academic year 2020-2021.

I also declare that, to the best of my knowledge and belief, the work reported here doesn't from part of any other dissertation on the basis of which a degree or award was conferred on an earlier occasion on this by any other student.

Place: **NAVYA SHREE PN**

Date: **(USN : 1AT18IS057)**

ABSTRACT

The application for viewing ‘Daily Mood’ is helpful for the people to keep a track about there mental health on the go with the busy life. Currently it is much of the burden of mental ill-health is mediated by early onset. This app aims to track the early period of mental disorders among young people mental health services to facilitate more streamlined transdiagnostic processes. There are various activities like getting started to the app, showing the articles, their category.

The project focuses on building a mental health tracker. Doing so, we get an idea of the mental state of the user (in the least intrusive ways), and then suggest measures they can take to get out of their present condition. The user answers some questions and based on the answers that they provide, our app will provide resources to them and help them maintain a record of their mood.

ACKNOWLEDGEMENT

I am grateful to my institution, **Atria Institute of Technology**, for having provided me with the facilities to successfully complete this mini project on newsMag.

I thank **Dr. T. N. Sreenivasa, principal** and **Dr. Shanthi Mahesh, HOD, ISE** for providing me all the necessary facilities for successful completion of my mini project.

Deadlines play a very important role in successful completion of the academic project on time, efficiently and effectively. I take this opportunity to express my deep sense of gratitude to my guide and coordinators **Ms. Uzma Sulthana, Assistant Professor and Ms. Prapulla G, Assistant Professor, Department of ISE** for their valuable guidance and help throughout the course of the academic mini project. They have always been patient with me and helped immensely in completing the task on hand. I also thank them for the immense support, guidance, specifications & ideas without which seminar would have been completed without full merit.

Last but not least from the Department of Information Science and Engineering, teaching and non-teaching staffs for their constant encouragement, support, patience and endurance shown during the preparation of this report were remarkable. I also thank the management.

Finally, I thank my parents and friends for their motivation, moral and material support.

Navya Shree PN
(USN : 1AT18IS057)

TABLE OF CONTENTS

LIST OF FIGURES

APPENDIX ‘A’ – Screenshots

CHAPTER 1

INTRODUCTION

1.1 Introduction to Mobile Application Development

Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources. Like web application development, mobile application development has its roots in more traditional software development.

1.1.1 History:

- The first mobile phones were invented whose microchips required the most basic software to send and receive voice calls.
- On 3rd of April 1973, Martin Cooper of Motorola made the first call on the mobile phone to Dr. Joel S. Engel of the Bell Labs.
- The R&D department of IBM Simon came up with the first mobile app for Smartphones in 1993 exactly two decades after the first call was made.
- EPOC, first operating system developed by Psion, released in the early 90s, this was first of the recognizable apps.
- Palm OS, developed by Palm Inc. in the year 1996, these were mainly designed for personal digital assistants and were known as Garnet OS.
- The wireless markup language was specifically designed for devices that were dependent on XML and could be run across wireless application protocols.
- Java ME or J2ME or JME – it was first introduced as JSR 68. It was given various shapes and forms for use via Phones, embedded devices, and even PDAs.
- Symbian, developed by Symbian Ltd, which was a joint venture from Ericsson, Motorola, Nokia and PSION, this was a further developed version of PSION EPOC.
- Later on, the smartphones and iPhones that we use today evolved, making lives a lot easier for people.

1.1.2 Advantages:

- Improves Efficiency.
- Offers High Scalability.
- Secures the App Data.
- Integrates With Existing Software.
- Easy to Maintain.
- Improves Customer Relationship.
- Facilitates New Client Data Retrieval.
- Provides Real-time Project Access.
- Ease in Project Management.

1.2 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA software. It provides the fastest tools for building apps on every type of android device. It is a purpose-built for android to accelerate the development and helps to build the highest-quality apps for every android device. Features of Android Studio include,

- A flexible Gradle-based build system.
- A fast and feature-rich emulator.
- A unified environment where one can develop for all Android devices.
- Extensive testing tools and frameworks.

1.2.1 Android Studio SDK:

Android SDK performs all the tasks needed to develop apps for all versions of Android. This program is a necessary tool for any developer who wants to make smoothly running applications for the latest systems. It uses Java for development and relies on the Integrated Development Environment, to build the apps and test them.

1.2.2 Android Studio Emulator:

The Android Emulator simulates Android devices on your computer so that you can test your application on a variety of devices and Android API levels without needing to have each physical device. The emulator provides almost all of the capabilities of a real Android device. Simulation of incoming phone calls and text messages, specify the location of the device, simulate different network speeds, simulate rotation and other hardware sensors, access the Google Play Store, and much more are possible.

1.3 JAVA

Java is an object-oriented programming language created by James Gosling, Mike Sheridan, and Patrick Naughton in 1991. It is a high-level, class-based language that is designed to have a few implementation dependencies as possible. It is a general-purpose programming language intended to let android developers run the compiled Java code on all platforms that support Java without any need for recompilation. Features of Java include,

- **Simple:** Java is designed to be easy to learn.
- **Secure:** With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- **Architecture-neutral:** Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- **Portable:** Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- **Robust:** Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking.
- **Multithreaded:** With Java's multithreaded feature it is possible to write programs
- that can perform many tasks simultaneously.
- **Interpreted:** Java byte code is translated on the fly to native machine instructions and is not stored anywhere.

- **High Performance:** With the use of Just-In-Time compilers, Java enables high performance.
- **Distributed:** Java is designed for the distributed environment of the internet.
- **Dynamic**: Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time.

1.4 XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML focus on simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Features of XML include,

- XML focuses on data rather than how it looks.
- Easy and efficient data sharing.
- Compatibility with other markup language HTML.
- Supports platform transition.
- Allows XML validation.
- Adapts technology advancements.
- XML supports Unicode.

1.5 Structure of Report

This report is for our mini project Daily Mood using Mobile Application Development concepts. Our report consists of five chapters where in first chapter we are giving introduction to mobile application development with its history. In second chapter we have given brief description of our problem definition and literature survey made. Similarly in third chapter we have a brief document on the requirement specifications of hardware and software required along with purpose and scope of our project. The fourth and fifth module gives us the design and implementation knowledge. To conclude we have added our conclusions and further enhancement. We have also mentioned the references to our project. Last but not the least we have the screen shots of our output showing execution of our program in appendix.

CHAPTER 2

LITERATURE SURVEY

2.1. Problem Definition

Daily Mood is an application which is helpful for the people to track mental health on the go with the busy life. The application helps them by providing the resources on short. There are various activities like getting started to the app, showing the articles, their category.

2.2. Purpose & Scope

In the application we can view various resources with respect to their category, we can share app, view about the app. Another advantage of the application is that people can note in a particular category and it is very easy to view them. Our application has several advantages.

Advantages:

i. User friendly interface
ii. Fast access to resources
iii. Less error
iv. Look and Feel Environment

2.3. Aim of the application

Our mental health application, Daily Mood focuses on building a mental health tracker. Doing so, we get an idea of the mental state of the user (in the least intrusive ways), and then suggest measures they can take to get out of their present condition. The user answers some questions and based on the answers that they provide, our app will provide resources to them and help them maintain a record of their mood.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1. Introduction

3.1.1. Purpose

In the application we can track mental health of a person, Another advantage of the application is that people can note their feelings in a particular category and it is very easy to view them.

3.1.2. Scope

For people with busy schedule in life who cannot have time to track mental health can use this application so that they do not miss to have knowledge of what is happening with their mental health.

3.1.3. Definition, Acronyms and Abbreviations

1. XML - Extensible Markup Language
2. MS – Microsoft
3. IDE - Integrated Development Environment
4. SDK – Software Development Kit

3.2. Development Environment

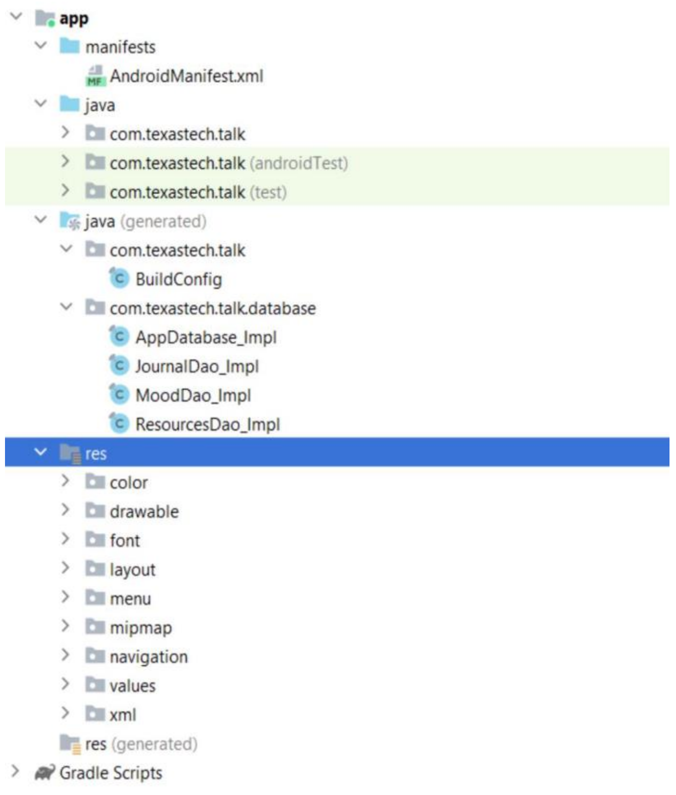
3.2.1. Android Programming Languages

In Android, programming is done in two languages JAVA or C++ and XML (Extension Markup Language). Nowadays KOTLIN is also preferred. The XML file deals with the design, presentation, layouts, blueprint, etc. (as a front-end) while the JAVA or KOTLIN deals with the working of buttons, variables, storing, etc. (as a back-end).

3.2.2. Android Components

- **Activities**: It deals with the UI and the user interactions to the screen. In other words, it is a User Interface that contains activities.
- **Services**: Services are the background actions performed by the app; these might be long- running operations. A service might need other sub-services so as to perform specific tasks.
- **Content Provider:** Content Provider is used to transferring the data from one application to the others at the request of the other application.
- **Broadcast Receivers:** A Broadcast is used to respond to messages from other applications or from the System.

3.2.3. Structural Layout of Android



```
app
  manifests
    AndroidManifest.xml
  java
    com.texastech.talk
    com.texastech.talk (androidTest)
    com.texastech.talk (test)
  java (generated)
    com.texastech.talk
      BuildConfig
    com.texastech.talk.database
      AppDatabase_Impl
      JournalDao_Impl
      MoodDao_Impl
      ResourcesDao_Impl
  res
    color
    drawable
    font
    layout
    menu
    mipmap
    navigation
    values
    xml
  res (generated)
Gradle Scripts
```

Figure 3.2.3 Structural Layout of Android Studio

- **Manifest Folder**: Android Manifest is an XML file that is the root of the project source set. It describes the essential information about the app and the Android build tools, the Android Operating System, and Google Play. It contains the permission that an app might need in order to perform a specific task. It also contains the Hardware and the Software features of the app, which determines the compatibility of an app on the Play Store.

- **Java Folder:** The JAVA folder consists of the java files that are required to perform the background task of the app. It consists of the functionality of the buttons, calculation, storing, variables, toast (small popup message), programming function, etc. The number of these files depends upon the type of activities created.

- **Resource Folder:** The res or Resource folder consists of the various resources that are used in the app. This consists of sub-folders like drawable, layout, mipmap, raw, and values. The drawable consists of the images. The layout consists of the XML files that define the user interface layout. These are stored in res.layout and are accessed as R.layout class. The raw consists of the Resources files like audio files or music files, etc. These are accessed through R.raw.filename.values are used to store the hardcoded strings (considered safe to store string values) values, integers, and colors.

- **Gradle Files:** Gradle is an advanced toolkit, which is used to manage the build process that drawable consists of the images. The layout consists of the XML files that define the user interface layout. These are stored in res.layout and are accessed as R.layout class. The raw consists of the Resources files like audio files or music files, etc. These are accessed through R.raw.filename.values are used to store the hardcoded strings (considered safe to store string values) values, integers, and colors.

3.3. Specific Requirements

3.3.1 Software Requirements

i. Windows 10 Operating System

ii. **Tool kit**: Android SDK (Software development kit), Java development kit (JDK)

iii. **IDE**: Android Studio

3.3.2. Hardware Requirements

i.1.8 GHz Processor

ii.8GB (IDE + Android SDK + Android Emulator)/ 4GB (minimum) RAM

iii.15 inches Monitor

iv.104 keys with keyboard and mouse

v.6.3 inches Physical Android Device

CHAPTER 4

DESIGN

4.1. Project Flow

In our project Daily Mood, we have used XML with different attributes and a code written in Java language for various activities.

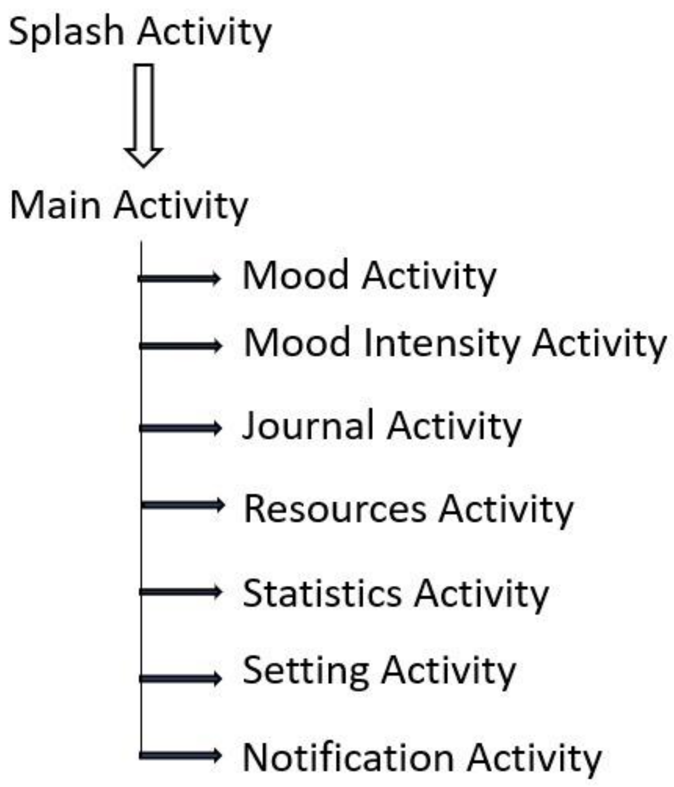


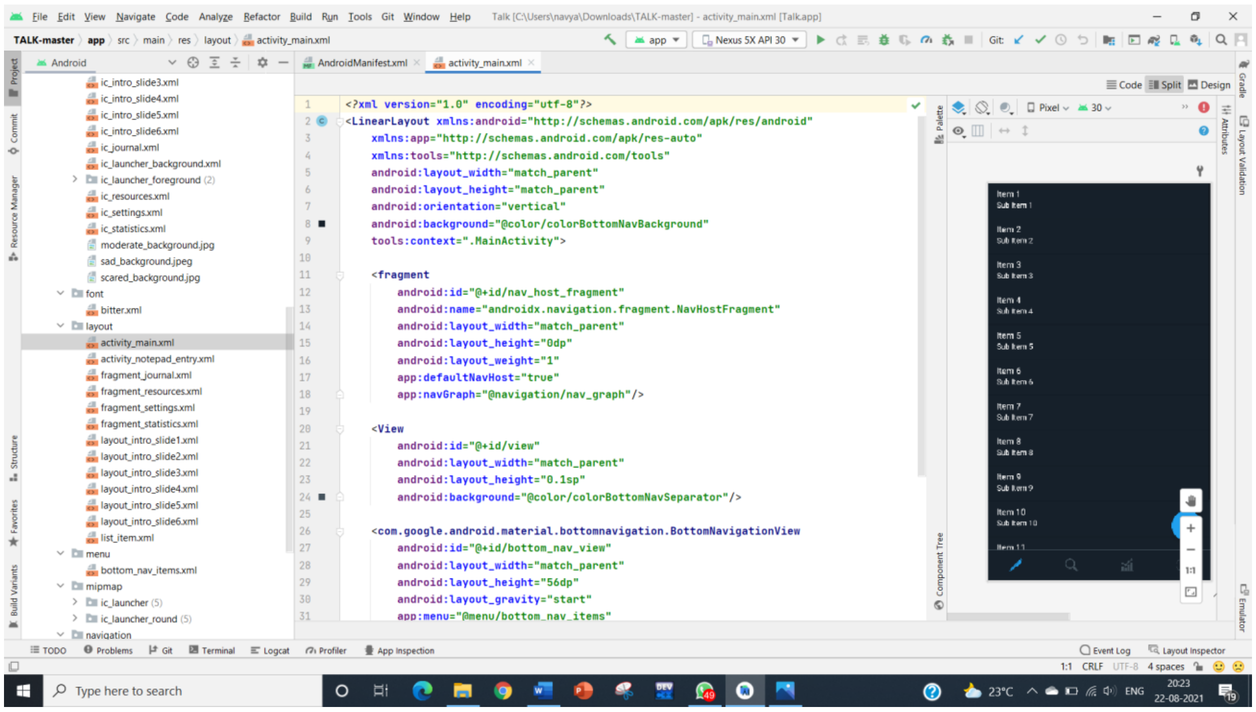
Figure 4.1 Flow diagram of DailyMood application

Once the user opens the application on the android device, splash screen with the app logo is displayed for few seconds and is navigated automatically to the main activity.It asks user few questions on their mood and also asks the intensity of the feeling of the user.Based on the input given by the user it suggests the resources.The icon of the resource is available in the notification panel which will be redirected to web once he click on it.User can also journal their thoughts.There is also the mood graph which tracks the mood of the user.In the settings panel he can enable or disable the notification.

4.2. Design using XML

4.2.1 activity_main.xml

The main activity, which is the first screen to appear when the user launches the app.

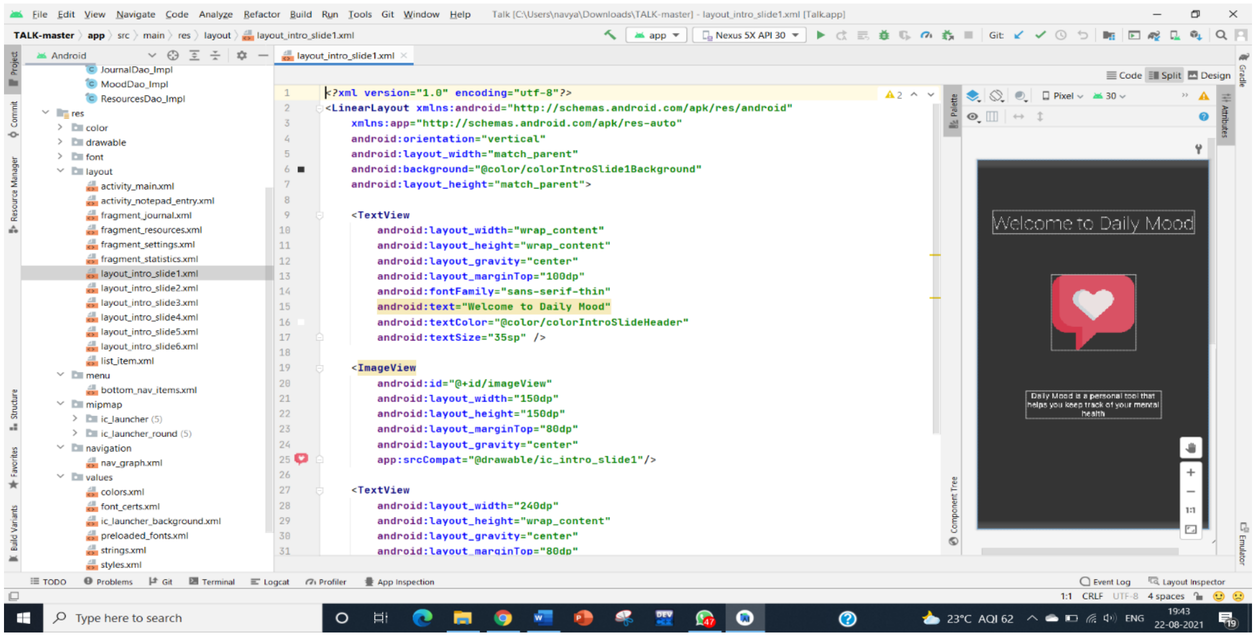


```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/colorBottomNavBackground"
    tools:context=".MainActivity">

    <fragment
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        app:defaultNavHost="true"
        app:navGraph="@navigation/nav_graph"/>

    <View
        android:id="@+id/view"
        android:layout_width="match_parent"
        android:layout_height="0.1sp"
        android:background="@color/colorBottomNavSeparator"/>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_nav_view"
        android:layout_width="match_parent"
        android:layout_height="56dp"
        android:layout_gravity="start"
        app:menu="@menu/bottom_nav_items"
```

Figure 4.2.1 Design Code screenshot for splash screen

4.2.2 Layout_intro_slide1.xml

The application on opening displays the splash screen which is designed in the Layout_intro_slide1_main.xml.

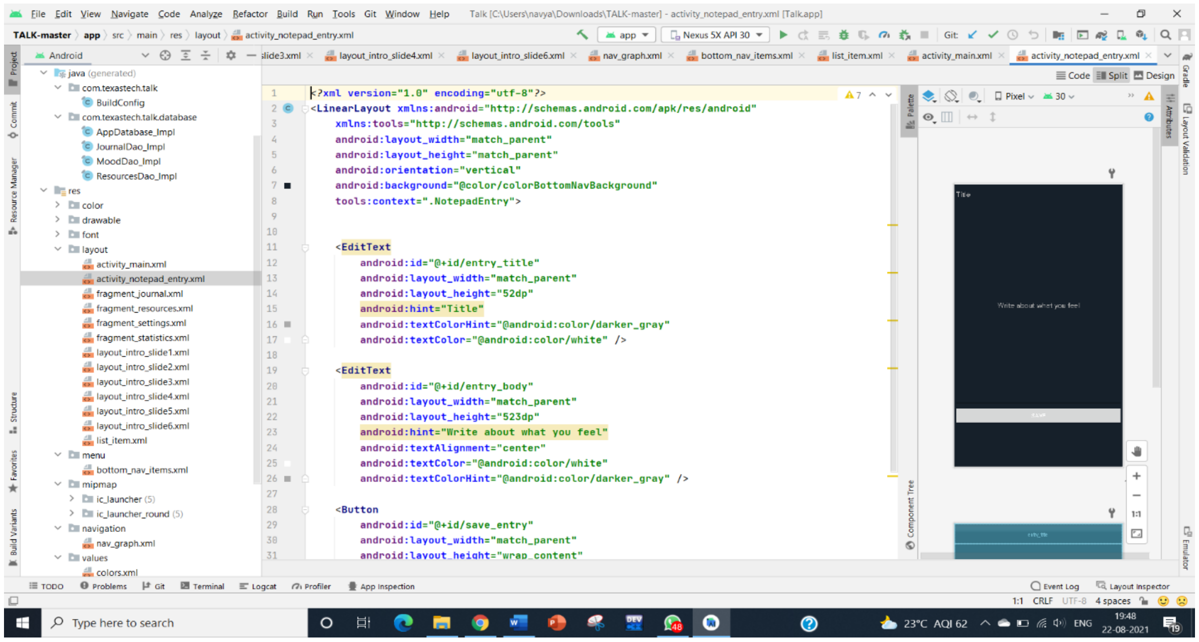


```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:background="@color/colorIntroSlide1Background"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="100dp"
        android:fontFamily="sans-serif-thin"
        android:text="Welcome to Daily Mood"
        android:textColor="@color/colorIntroSlideHeader"
        android:textSize="35sp" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_marginTop="80dp"
        android:layout_gravity="center"
        app:srcCompat="@drawable/ic_intro_slide1"/>

    <TextView
        android:layout_width="240dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="80dp"
```

Figure 4.2.2 Design Code screenshot for layout intro

4.2.3 activity_notepad_entry.xml

This activity is signed which allows the user to note down there feelings.

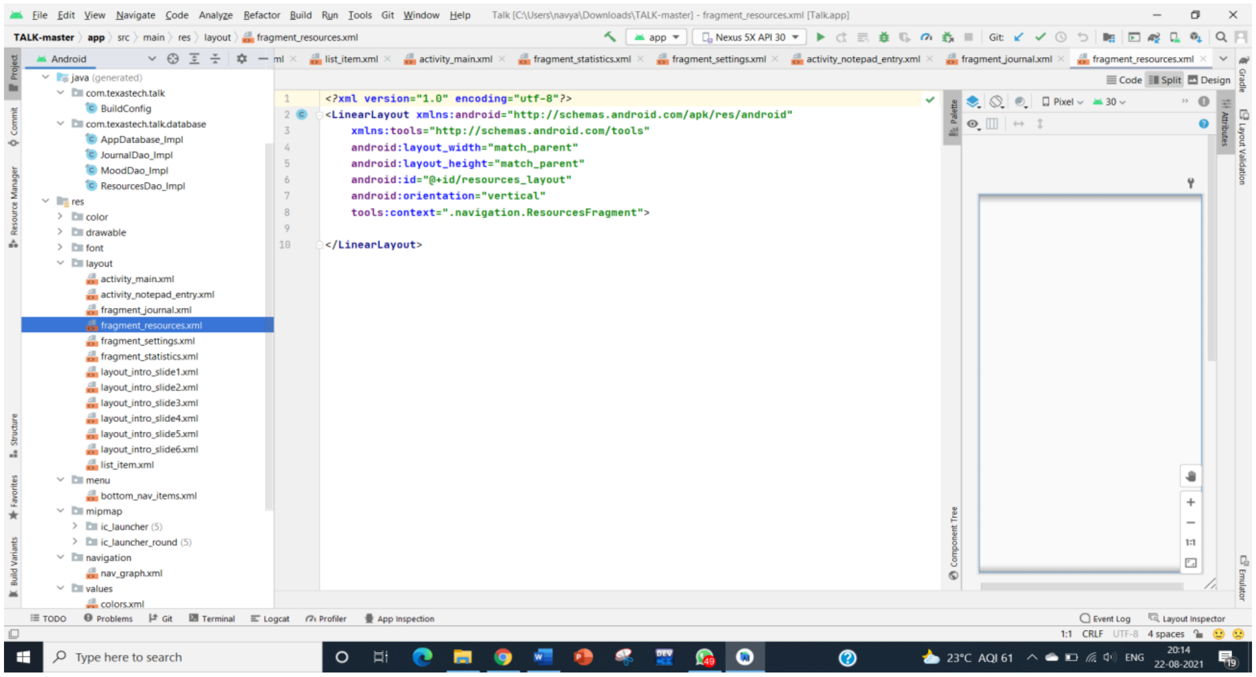


```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/colorBottomNavBackground"
    tools:context=".NotepadEntry">


    <EditText
        android:id="@+id/entry_title"
        android:layout_width="match_parent"
        android:layout_height="52dp"
        android:hint="Title"
        android:textColorHint="@android:color/darker_gray"
        android:textColor="@android:color/white" />

    <EditText
        android:id="@+id/entry_body"
        android:layout_width="match_parent"
        android:layout_height="523dp"
        android:hint="Write about what you feel"
        android:textAlignment="center"
        android:textColor="@android:color/white"
        android:textColorHint="@android:color/darker_gray" />

    <Button
        android:id="@+id/save_entry"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

Figure 4.2.3 Design Code screenshot for notepad entry

4.2.4 Fragmentation_resource.xml

The fragmentation_resource.xml has various card views to display the mood category and also contains Resouces.

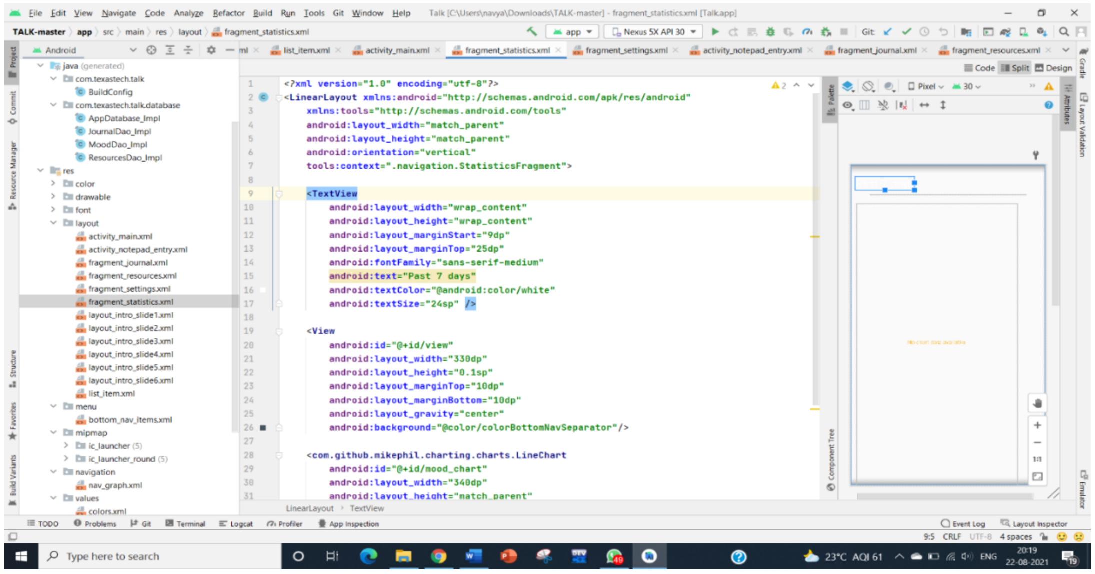


```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/resources_layout"
    android:orientation="vertical"
    tools:context=".navigation.ResourcesFragment">

</LinearLayout>
```

Figure 4.2.4 Design Code screenshot of fragmentation resource

4.2.5 Fragmentation_statistics.xml

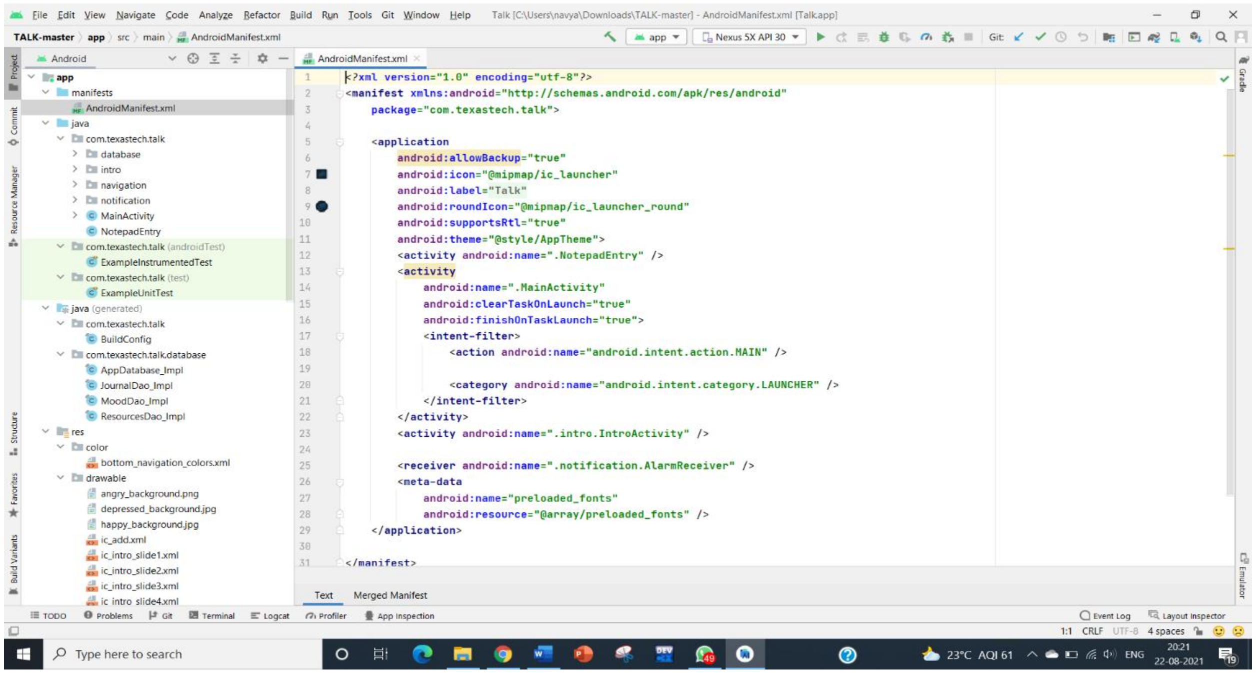
This activity is designed for the user to keep a track of their mood.



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".navigation.StatisticsFragment">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="9dp"
        android:layout_marginTop="25dp"
        android:fontFamily="sans-serif-medium"
        android:text="Past 7 days"
        android:textColor="@android:color/white"
        android:textSize="24sp" />

    <View
        android:id="@+id/view"
        android:layout_width="330dp"
        android:layout_height="0.1sp"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp"
        android:layout_gravity="center"
        android:background="@color/colorBottomNavSeparator"/>

    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/mood_chart"
        android:layout_width="340dp"
        android:layout_height="match_parent"
```

Figure 4.2.5 Design Code screenshot of fragmentation statistics

4.2.6 AndroidManifest.xml

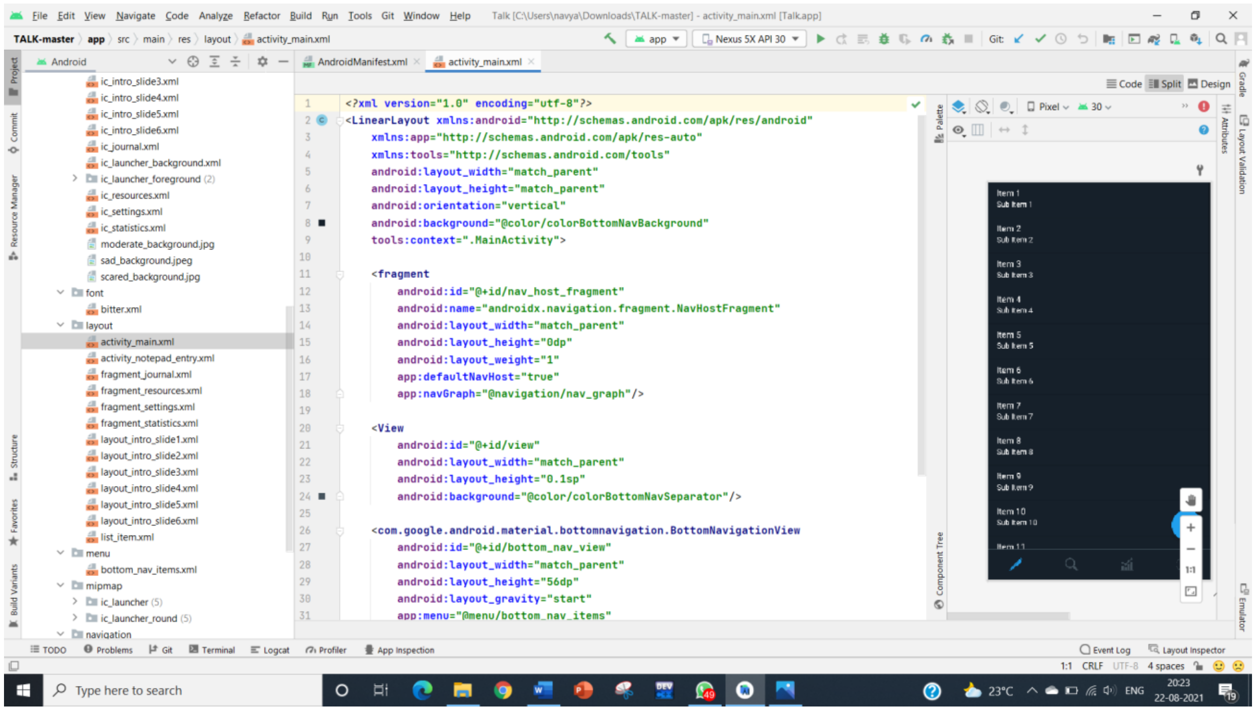


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.texastech.talk">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Talk"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".NotepadEntry" />
        <activity
            android:name=".MainActivity"
            android:clearTaskOnLaunch="true"
            android:finishOnTaskLaunch="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".intro.IntroActivity" />

        <receiver android:name=".notification.AlarmReceiver" />
        <meta-data
            android:name="preloaded_fonts"
            android:resource="@array/preloaded_fonts" />
    </application>

</manifest>
```

Figure 4.2.6 Code screenshot of Android Manifest

4.2.7 strings.xml

The string file is used to initialized strings and fetch using it’s id.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/colorBottomNavBackground"
    tools:context=".MainActivity">

    <fragment
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        app:defaultNavHost="true"
        app:navGraph="@navigation/nav_graph"/>

    <View
        android:id="@+id/view"
        android:layout_width="match_parent"
        android:layout_height="0.1sp"
        android:background="@color/colorBottomNavSeparator"/>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_nav_view"
        android:layout_width="match_parent"
        android:layout_height="56dp"
        android:layout_gravity="start"
        app:menu="@menu/bottom_nav_items"
```

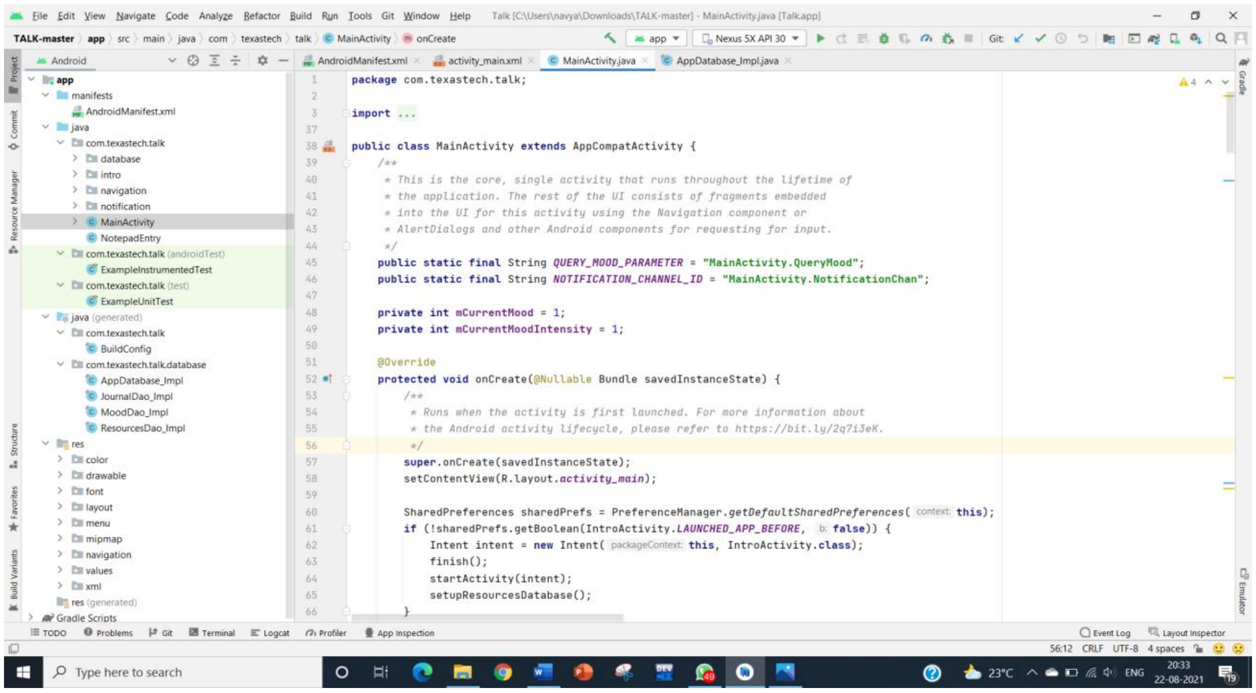
Figure 4.2.7 Design Code screenshot of Strings

CHAPTER 5

IMPLEMENTATION

5.1. MainActivity.java

The MainActivity.java file contains the splash screen where it is passed on to the next screen.

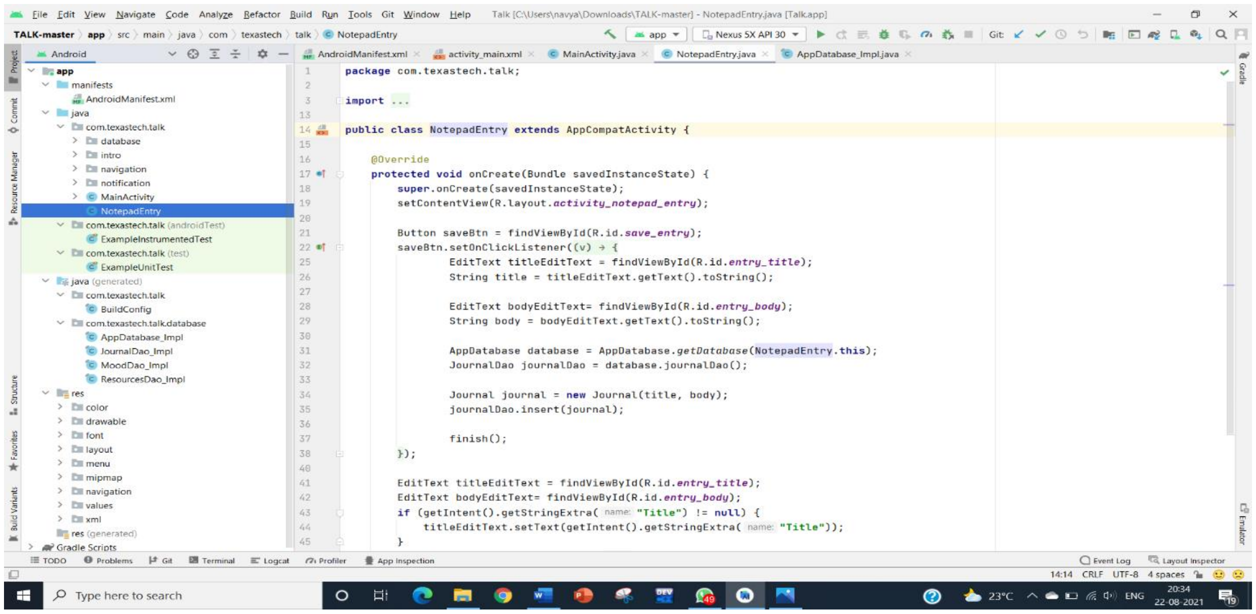


```java
package com.texastech.talk;

import ...

public class MainActivity extends AppCompatActivity {
    /**
     * This is the core, single activity that runs throughout the lifetime of
     * the application. The rest of the UI consists of fragments embedded
     * into the UI for this activity using the Navigation component or
     * AlertDialogs and other Android components for requesting for input.
     */
    public static final String QUERY_MOOD_PARAMETER = "MainActivity.QueryMood";
    public static final String NOTIFICATION_CHANNEL_ID = "MainActivity.NotificationChan";

    private int mCurrentMood = 1;
    private int mCurrentMoodIntensity = 1;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        /**
         * Runs when the activity is first launched. For more information about
         * the Android activity lifecycle, please refer to https://bit.ly/2q7i3eK.
         */
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        SharedPreferences sharedPrefs = PreferenceManager.getDefaultSharedPreferences( context: this);
        if (!sharedPrefs.getBoolean(IntroActivity.LAUNCHED_APP_BEFORE,  b: false)) {
            Intent intent = new Intent( packageContext: this, IntroActivity.class);
            finish();
            startActivity(intent);
            setupResourcesDatabase();
        }
```

Figure 5.1 Code screenshot for MainActivity.java

5.2. NotepadEntry.java

The Notepad screen contains the code which helps the user to track down their feelings

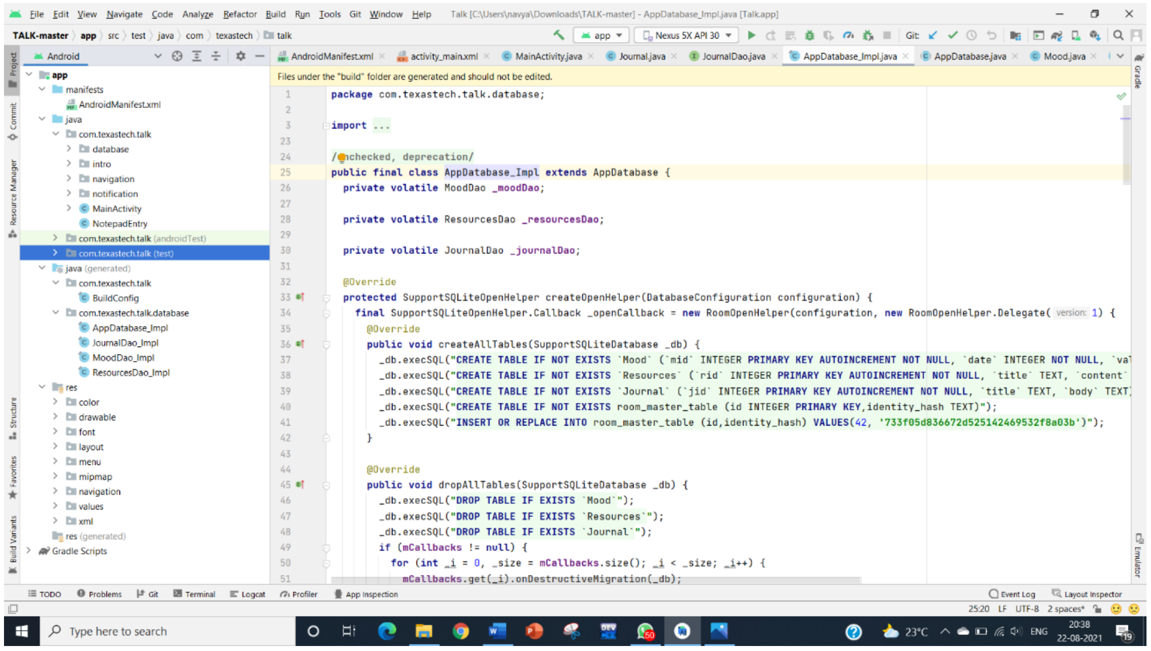


```java
package com.texastech.talk;

import ...

public class NotepadEntry extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_notepad_entry);

        Button saveBtn = findViewById(R.id.save_entry);
        saveBtn.setOnClickListener((v) → {
                EditText titleEditText = findViewById(R.id.entry_title);
                String title = titleEditText.getText().toString();

                EditText bodyEditText= findViewById(R.id.entry_body);
                String body = bodyEditText.getText().toString();

                AppDatabase database = AppDatabase.getDatabase(NotepadEntry.this);
                JournalDao journalDao = database.journalDao();

                Journal journal = new Journal(title, body);
                journalDao.insert(journal);

                finish();
        });

        EditText titleEditText = findViewById(R.id.entry_title);
        EditText bodyEditText= findViewById(R.id.entry_body);
        if (getIntent().getStringExtra( name: "Title") != null) {
            titleEditText.setText(getIntent().getStringExtra( name: "Title"));
        }
```

Figure 5.2 Code screenshot for NotepadEntry.java

5.3. AppDatabase_lmpLjava

It is used to store data inside the user's device in the form of a Text file. We can perform so many operations on this data such as adding new data, updating, reading, and deleting this data.

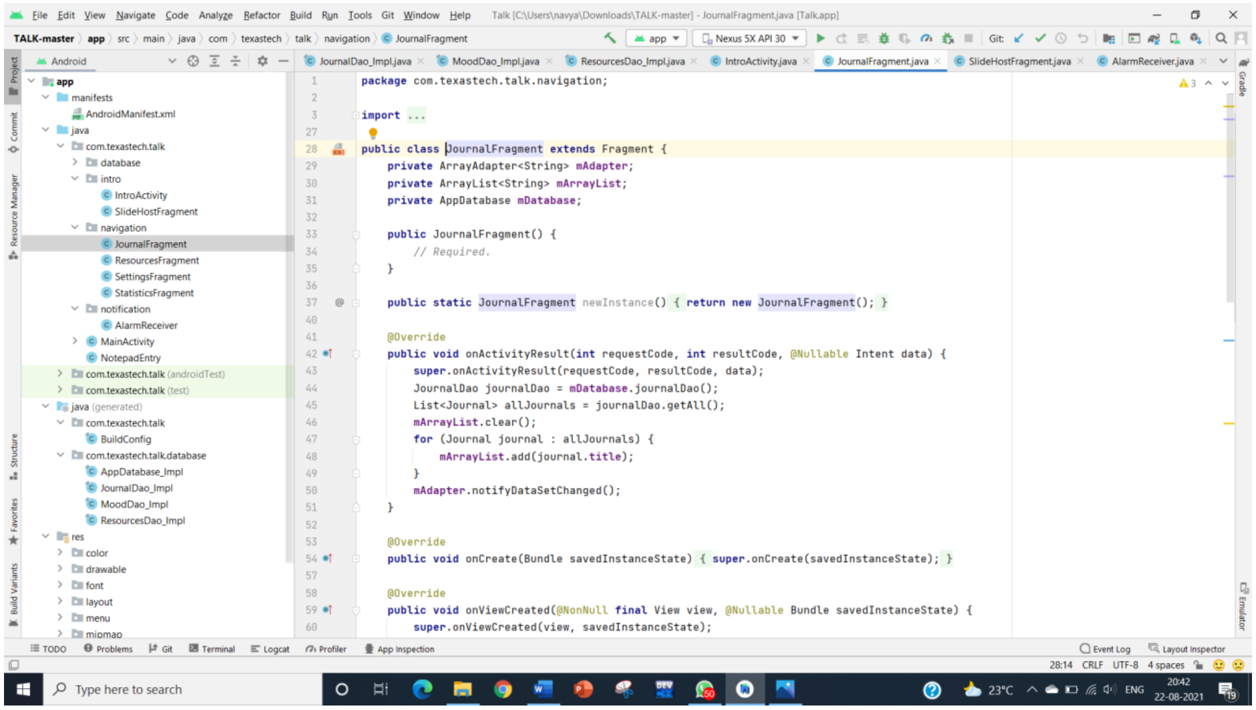


```java
package com.texastech.talk.database;

import ...

/unchecked, deprecation/
public final class AppDatabase_Impl extends AppDatabase {
  private volatile MoodDao _moodDao;

  private volatile ResourcesDao _resourcesDao;

  private volatile JournalDao _journalDao;

  @Override
  protected SupportSQLiteOpenHelper createOpenHelper(DatabaseConfiguration configuration) {
    final SupportSQLiteOpenHelper.Callback _openCallback = new RoomOpenHelper(configuration, new RoomOpenHelper.Delegate( version: 1) {
      @Override
      public void createAllTables(SupportSQLiteDatabase _db) {
        _db.execSQL("CREATE TABLE IF NOT EXISTS `Mood` (`mid` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `date` INTEGER NOT NULL, `va
        _db.execSQL("CREATE TABLE IF NOT EXISTS `Resources` (`rid` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `title` TEXT, `content`
        _db.execSQL("CREATE TABLE IF NOT EXISTS `Journal` (`jid` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `title` TEXT, `body` TEXT)
        _db.execSQL("CREATE TABLE IF NOT EXISTS room_master_table (id INTEGER PRIMARY KEY,identity_hash TEXT)");
        _db.execSQL("INSERT OR REPLACE INTO room_master_table (id,identity_hash) VALUES(42, '733f05d836672d525142469532f8a03b')");
      }

      @Override
      public void dropAllTables(SupportSQLiteDatabase _db) {
        _db.execSQL("DROP TABLE IF EXISTS `Mood`");
        _db.execSQL("DROP TABLE IF EXISTS `Resources`");
        _db.execSQL("DROP TABLE IF EXISTS `Journal`");
        if (mCallbacks != null) {
          for (int _i = 0, _size = mCallbacks.size(); _i < _size; _i++) {
            mCallbacks.get(_i).onDestructiveMigration(_db);
```

Figure 5.3 Screenshot showing AppDatabase

5.4. JournalFragment.java

The Journalfragment file helps in connecting to the resources fragment.

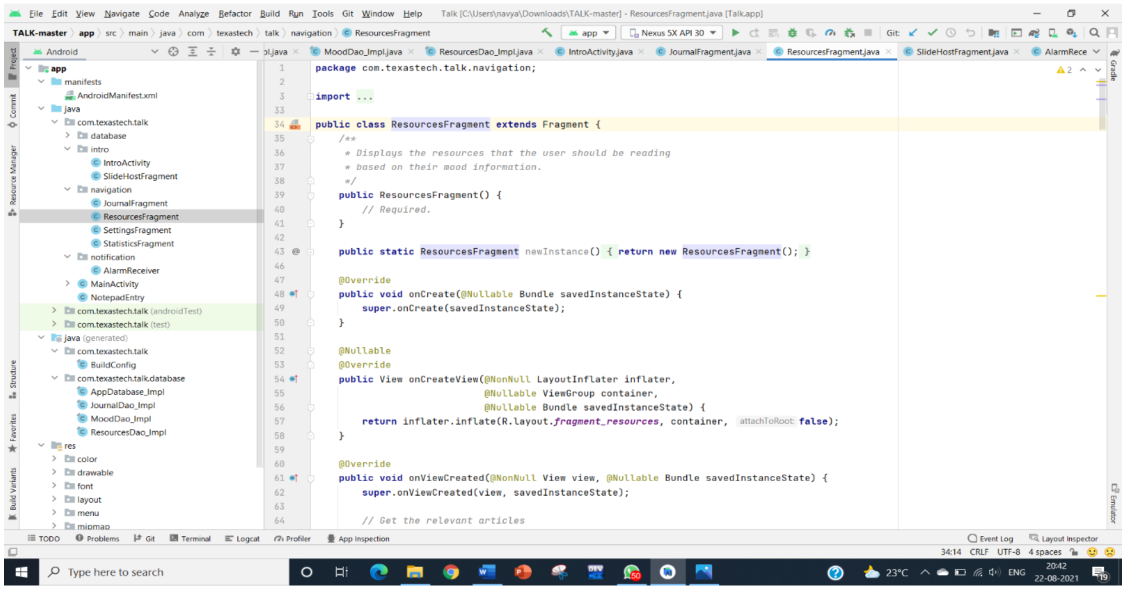


```java
package com.texastech.talk.navigation;

import ...

public class JournalFragment extends Fragment {
    private ArrayAdapter<String> mAdapter;
    private ArrayList<String> mArrayList;
    private AppDatabase mDatabase;

    public JournalFragment() {
        // Required.
    }

    public static JournalFragment newInstance() { return new JournalFragment(); }

    @Override
    public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        JournalDao journalDao = mDatabase.journalDao();
        List<Journal> allJournals = journalDao.getAll();
        mArrayList.clear();
        for (Journal journal : allJournals) {
            mArrayList.add(journal.title);
        }
        mAdapter.notifyDataSetChanged();
    }

    @Override
    public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); }

    @Override
    public void onViewCreated(@NonNull final View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
```

Figure 5.4 Code screenshot of JournalFragment.java

5.5. ResourcesFragment.java

Resources fragment provides the user with the resources based on the input provided by the user.

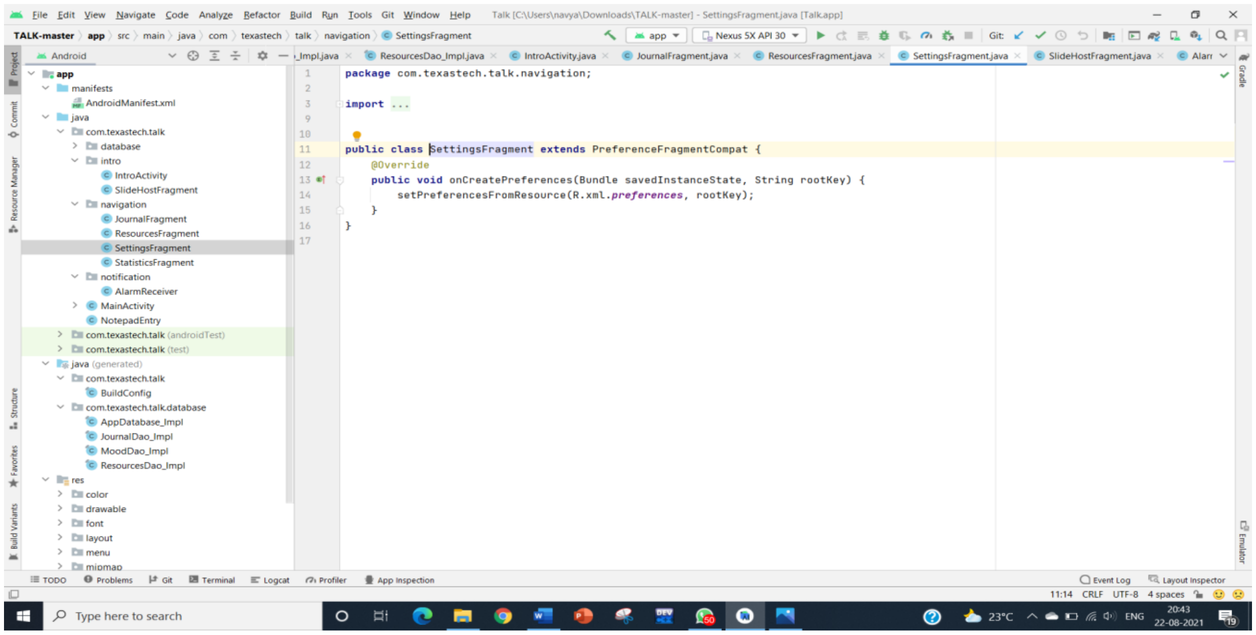


```java
package com.texastech.talk.navigation;

import ...

public class ResourcesFragment extends Fragment {
    /**
     * Displays the resources that the user should be reading
     * based on their mood information.
     */
    public ResourcesFragment() {
        // Required.
    }

    public static ResourcesFragment newInstance() { return new ResourcesFragment(); }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                             @Nullable ViewGroup container,
                             @Nullable Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_resources, container, attachToRoot: false);
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        // Get the relevant articles
```

Figure 5.5 Code screenshot for ResourcesFragment.java

5.6. SettingsFragment.java

It connects the user to the notification activity where the user can choose to enable or disable the activity.

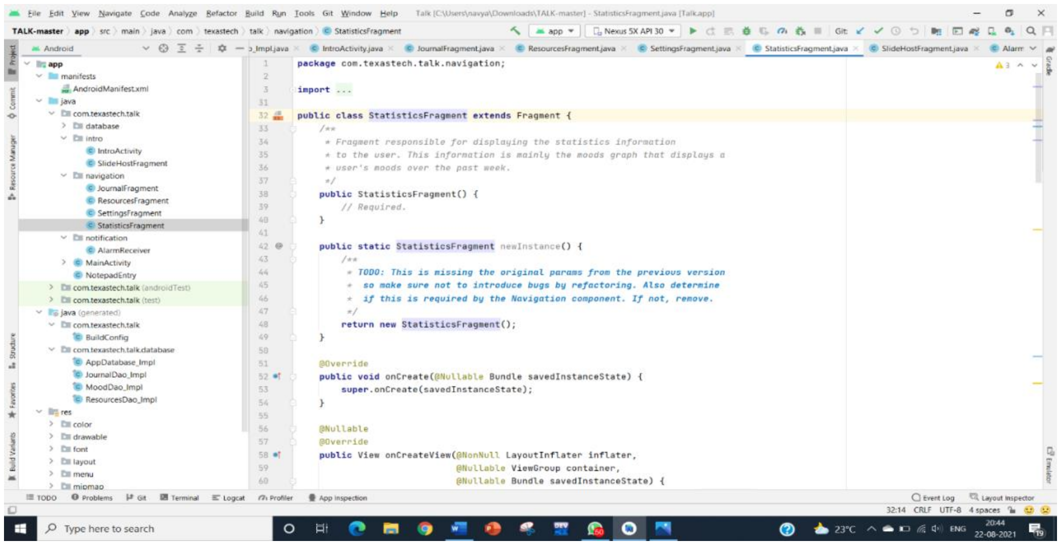


```java
package com.texastech.talk.navigation;

import ...

public class SettingsFragment extends PreferenceFragmentCompat {
    @Override
    public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {
        setPreferencesFromResource(R.xml.preferences, rootKey);
    }
}
```

Figure 5.6 Code screenshot for SettingsFragment.java

5.7. StatisticsFragment.java

Statistics fragment provides the user with a mood graph or provides a graph to the user to track there feelings.

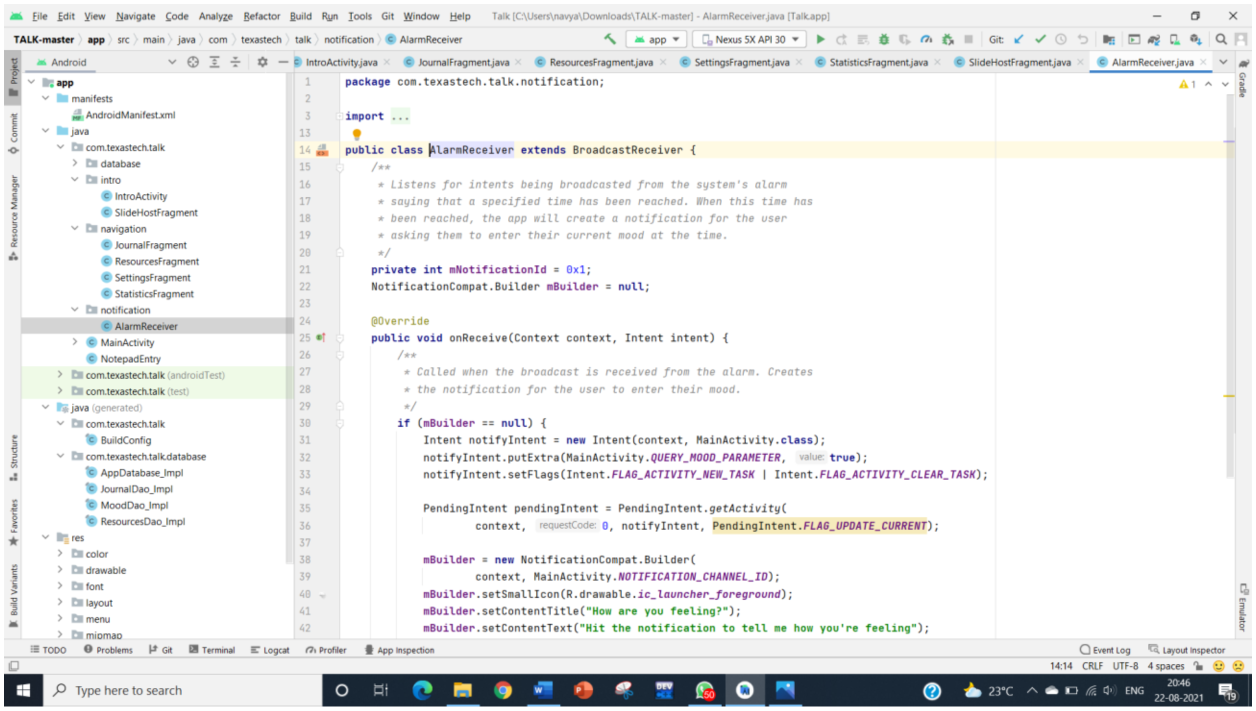


```java
package com.texastech.talk.navigation;

import ...

public class StatisticsFragment extends Fragment {
    /**
     * Fragment responsible for displaying the statistics information
     * to the user. This information is mainly the moods graph that displays a
     * user's moods over the past week.
     */
    public StatisticsFragment() {
        // Required.
    }

    public static StatisticsFragment newInstance() {
        /**
         * TODO: This is missing the original params from the previous version
         *  so make sure not to introduce bugs by refactoring. Also determine
         *  if this is required by the Navigation component. If not, remove.
         */
        return new StatisticsFragment();
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                             @Nullable ViewGroup container,
                             @Nullable Bundle savedInstanceState) {
```

Figure 5.7 Code screenshot for StatisticsFragment.java

5.8. AlarmReceiver.java

Alarm.java is used to specify what all entries should be displayed in the notification panel.

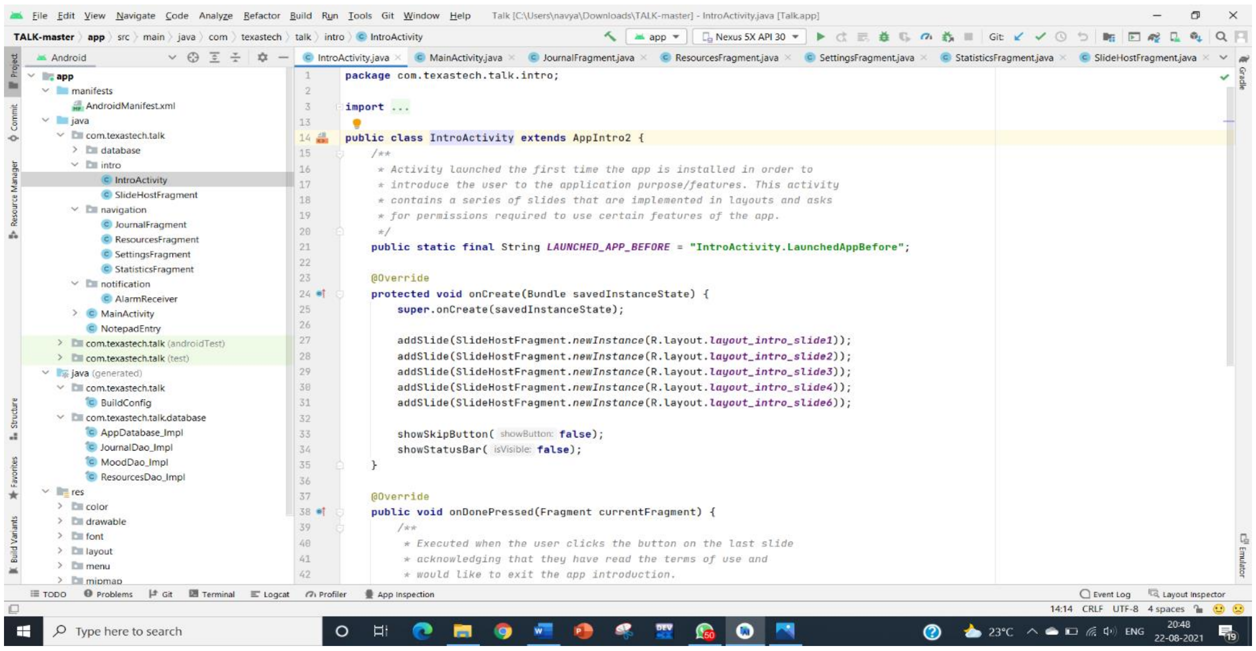


```
package com.texastech.talk.notification;

import ...

public class AlarmReceiver extends BroadcastReceiver {
    /**
     * Listens for intents being broadcasted from the system's alarm
     * saying that a specified time has been reached. When this time has
     * been reached, the app will create a notification for the user
     * asking them to enter their current mood at the time.
     */
    private int mNotificationId = 0x1;
    NotificationCompat.Builder mBuilder = null;

    @Override
    public void onReceive(Context context, Intent intent) {
        /**
         * Called when the broadcast is received from the alarm. Creates
         * the notification for the user to enter their mood.
         */
        if (mBuilder == null) {
            Intent notifyIntent = new Intent(context, MainActivity.class);
            notifyIntent.putExtra(MainActivity.QUERY_MOOD_PARAMETER, value: true);
            notifyIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);

            PendingIntent pendingIntent = PendingIntent.getActivity(
                    context, requestCode: 0, notifyIntent, PendingIntent.FLAG_UPDATE_CURRENT);

            mBuilder = new NotificationCompat.Builder(
                    context, MainActivity.NOTIFICATION_CHANNEL_ID);
            mBuilder.setSmallIcon(R.drawable.ic_launcher_foreground);
            mBuilder.setContentTitle("How are you feeling?");
            mBuilder.setContentText("Hit the notification to tell me how you're feeling");
```

Figure 5.8 Code screenshot for AlarmReceiver.java file

5.9. IntroActivity.java

MainActivity is connected to IntroActivity.java



```
package com.texastech.talk.intro;

import ...

public class IntroActivity extends AppIntro2 {
    /**
     * Activity launched the first time the app is installed in order to
     * introduce the user to the application purpose/features. This activity
     * contains a series of slides that are implemented in layouts and asks
     * for permissions required to use certain features of the app.
     */
    public static final String LAUNCHED_APP_BEFORE = "IntroActivity.LaunchedAppBefore";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        addSlide(SlideHostFragment.newInstance(R.layout.layout_intro_slide1));
        addSlide(SlideHostFragment.newInstance(R.layout.layout_intro_slide2));
        addSlide(SlideHostFragment.newInstance(R.layout.layout_intro_slide3));
        addSlide(SlideHostFragment.newInstance(R.layout.layout_intro_slide4));
        addSlide(SlideHostFragment.newInstance(R.layout.layout_intro_slide6));

        showSkipButton( showButton: false);
        showStatusBar( isVisible: false);
    }

    @Override
    public void onDonePressed(Fragment currentFragment) {
        /**
         * Executed when the user clicks the button on the last slide
         * acknowledging that they have read the terms of use and
         * would like to exit the app introduction.
```

Figure 5.9 Code screenshot for IntroActivity.java

CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion

Our generation relies mostly on phones to get through the day. Due to this, phones have become more of a personal assistant than a means to just communicate. Viewing news in short on the go with the busy schedule would make it easier for people. Henceforth to conclude, we've developed a mobile application based on Java using Android Studio. And for the efficient working of the ongoing system we've this application to help people to read news and stay updated easily in limited time.

Future Enhancements

i. We can use APIs to draw news from web to get displayed in the application.

ii. We can even further make it private and secured by enhancing login features.

iii. We can allow different users to write comments to news using their account which can be viewed by others.

iv. We can make it more space and resource efficient so that this application consumes lesser RAM and ROM.

REFERENCES

[1] IEEE Standard 830-1998: IEEE Recommended Practice for Software Requirements specifications.

[2] www.google.co.in

[3] Various videos from www.youtube.com

[4] https:// developer.android.com

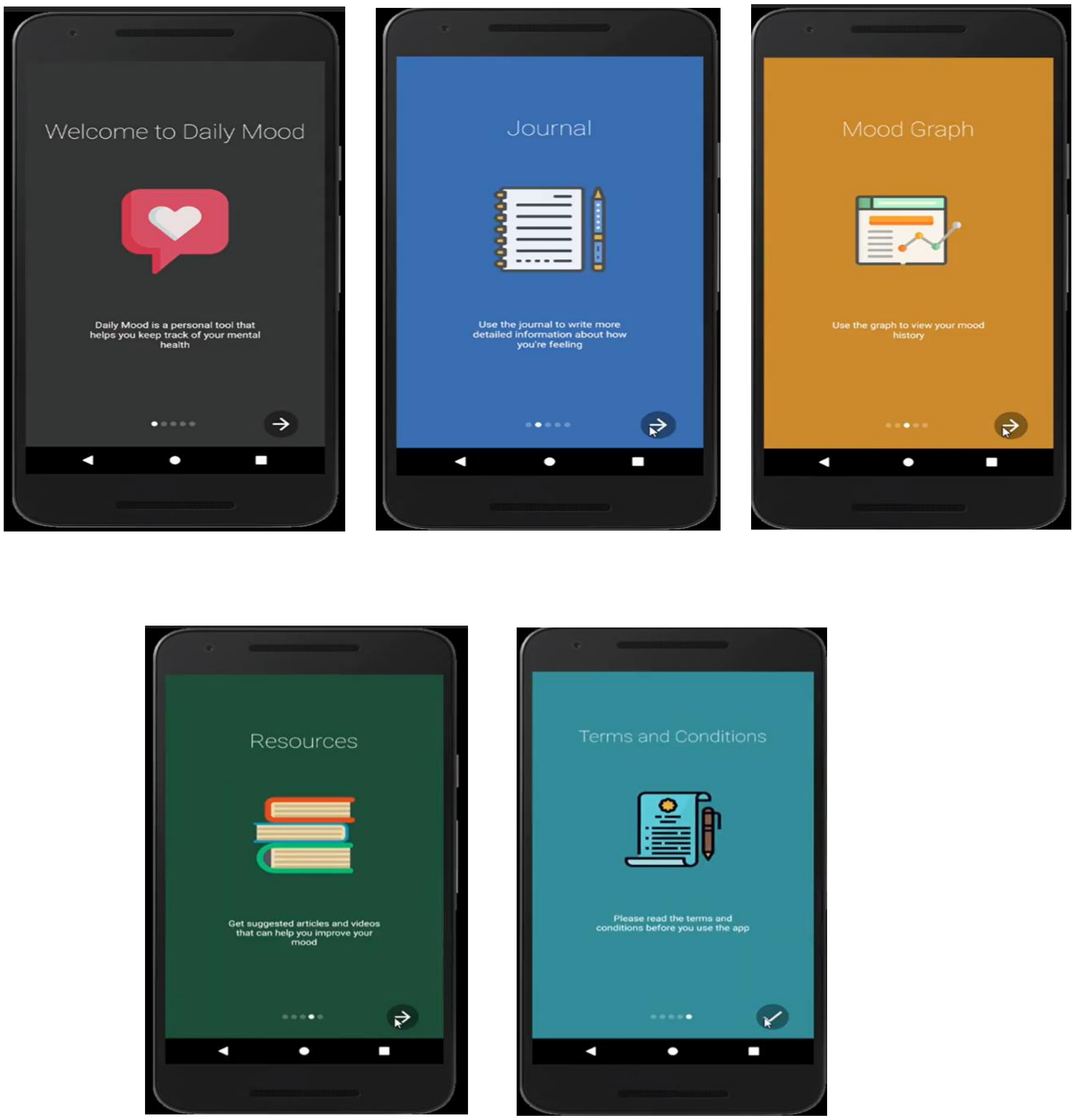
[5] https://en.wikipedia.org/wiki

Appendix ‘A’
Screenshots

RESULTS

1. Main Activity (Splash Screen)

This is the screenshot of our mini-project's splash screen with logo which will be occurring for few seconds when the application is opened.


Welcome to Daily Mood

Daily Mood is a personal tool that helps you keep track of your mental health

Journal

Use the journal to write more detailed information about how you're feeling

Mood Graph

Use the graph to view your mood history

Resources

Get suggested articles and videos that can help you improve your mood

Terms and Conditions

Please read the terms and conditions before you use the app

Figure A.1. Splash Screen of DailyMood application

2. Mood Activity

This is the screenshot of our mini-project's mood activity screen where the user can track their feelings.

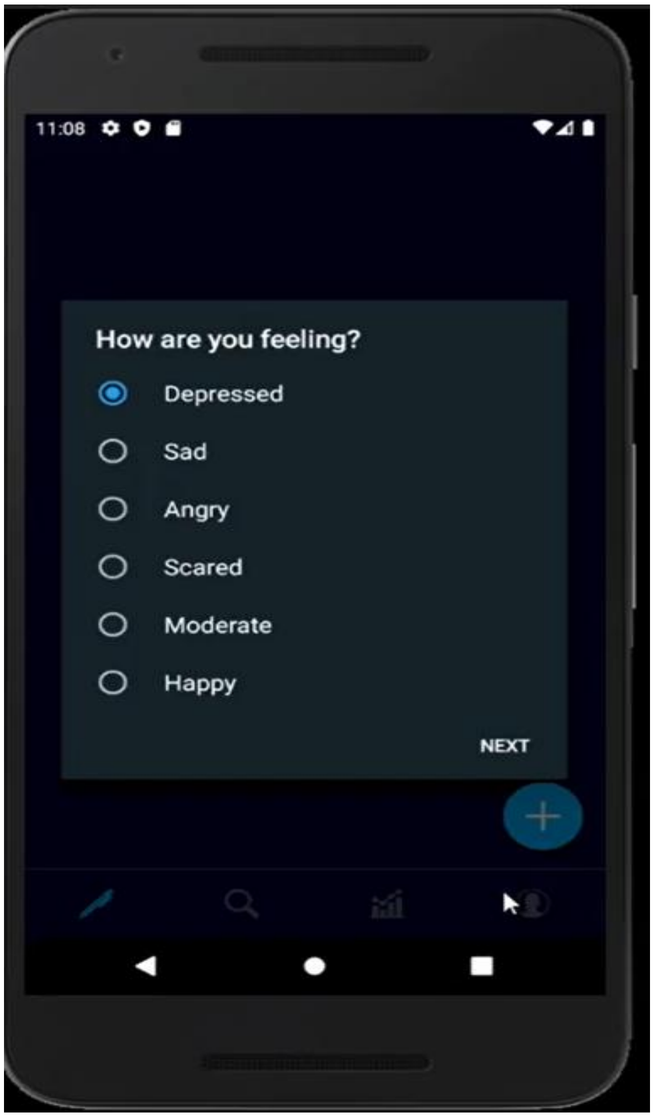


Figure A.2. Mood Activity of DailyMood application

3. Mood Intensity Activity

This is the screenshot of our mini-project's Mood intensity screen where users Emotions vary not only in type, but **also in intensity**. For example, people may not only feel happy or sad, they can also feel each along a continuum ranging from slightly happy or sad to extremely happy or sad.

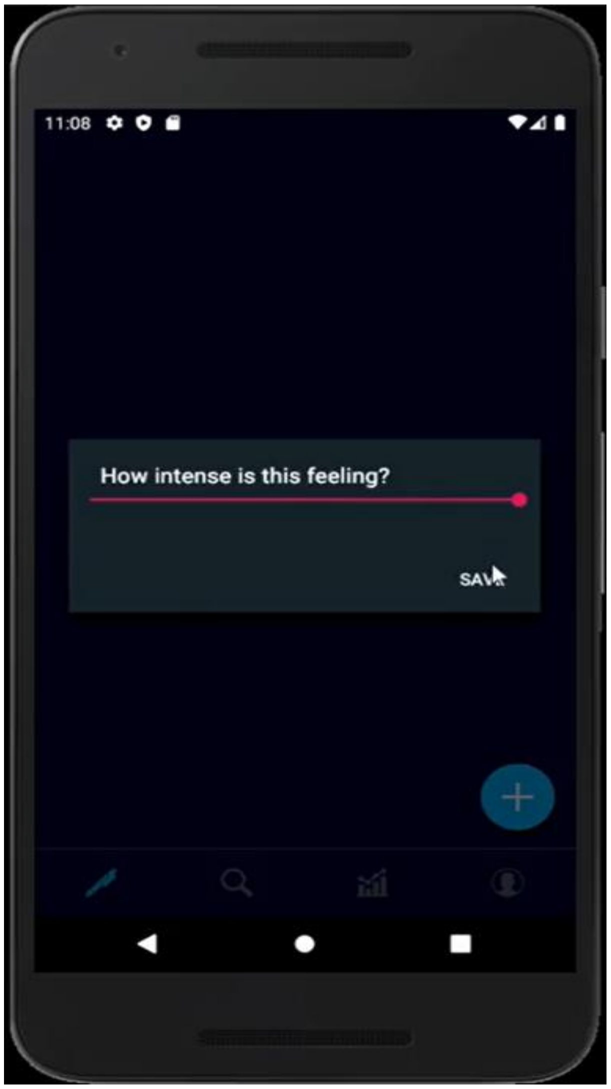


Figure A.3. Mood Intensity Screen of DailyMood application

4. Journal Activity

This is the screenshot of our mini-project's where user can journal their feelings.

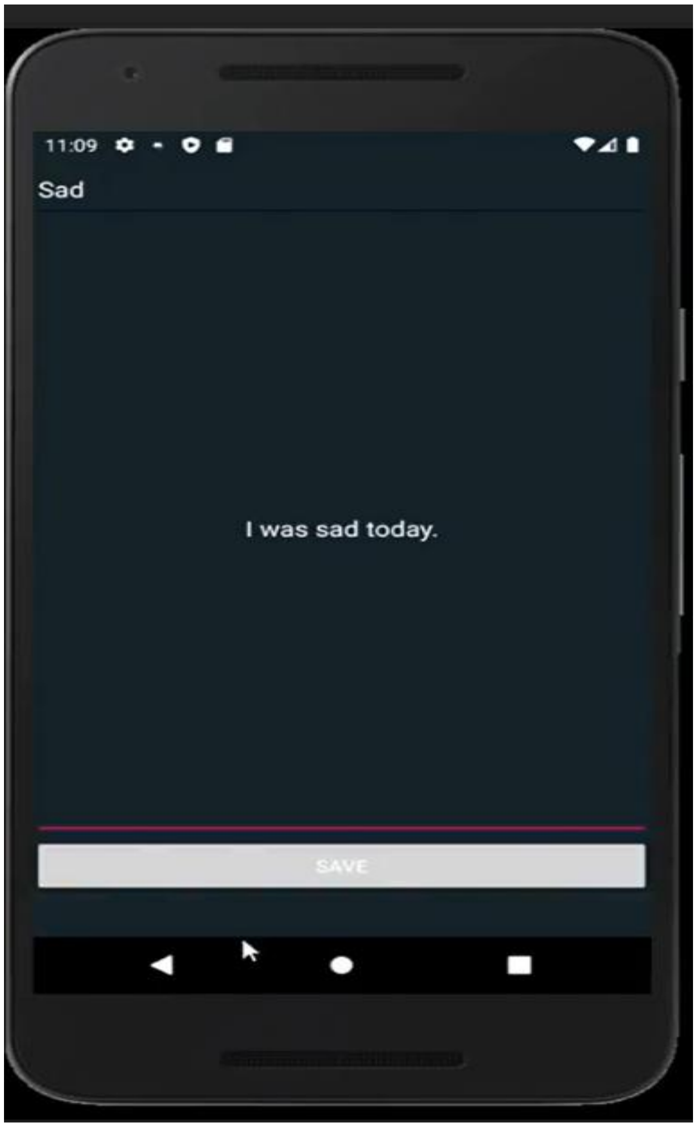
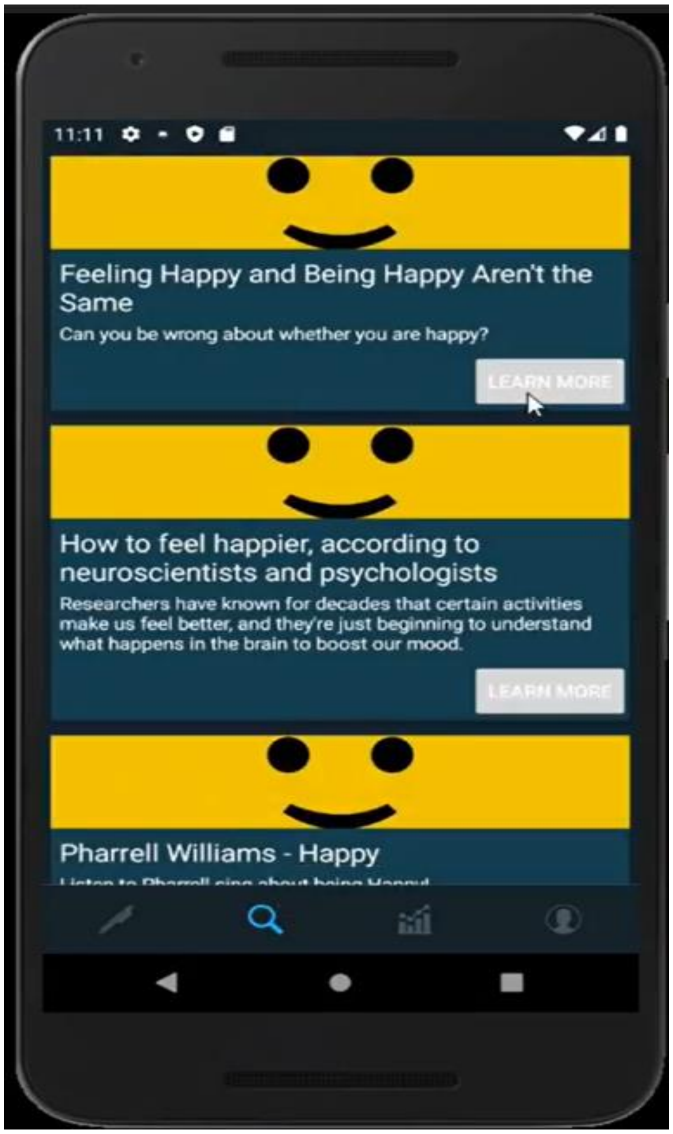


Figure A.4. Journal Screen of DailyMood application

5.Resources Activity

This is the screenshot of Resources activity of our mini-project's where user can select among various categories provided based on their interest and feeling.



Feeling Happy and Being Happy Aren't the Same

Can you be wrong about whether you are happy?

How to feel happier, according to neuroscientists and psychologists

Researchers have known for decades that certain activities make us feel better, and they're just beginning to understand what happens in the brain to boost our mood.

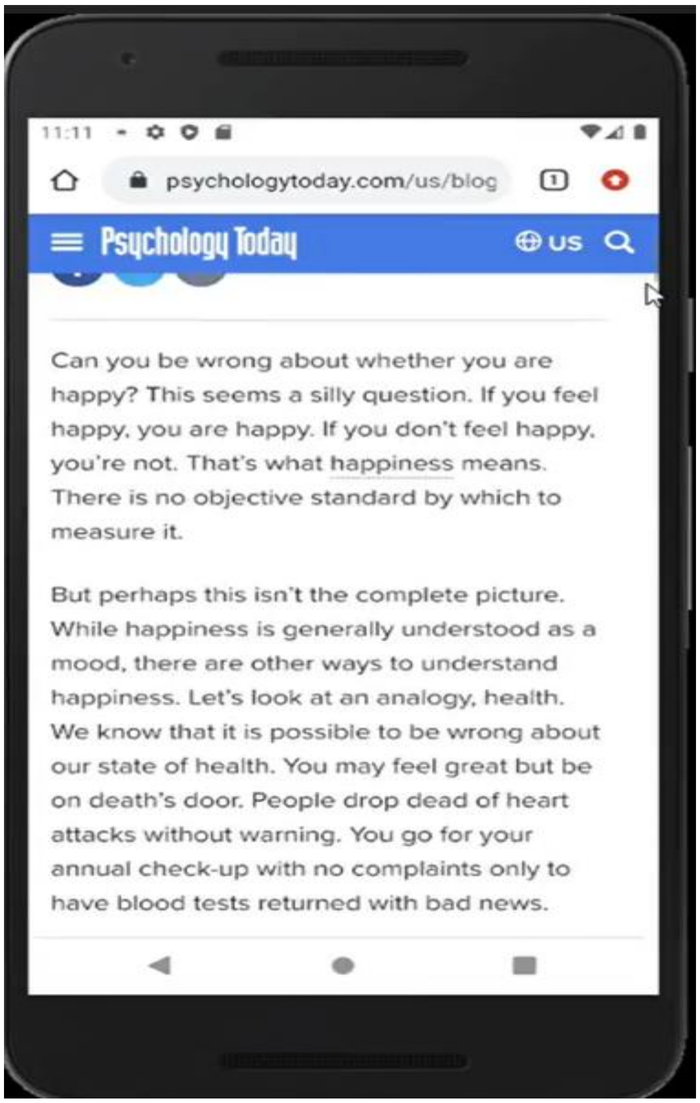
Pharrell Williams - Happy

Listen to Pharrell sing about being Happy!

Figure A.5. Resource Screen of DailyMood application

6. Resources Activity

This is the screenshot of Resouces Activity of our mini-project's where user can select among various categories provided based on their interest and feeling, it also takes you to web and provides the user with necessary information required.



Can you be wrong about whether you are happy? This seems a silly question. If you feel happy, you are happy. If you don't feel happy, you're not. That's what happiness means. There is no objective standard by which to measure it.

But perhaps this isn't the complete picture. While happiness is generally understood as a mood, there are other ways to understand happiness. Let's look at an analogy, health. We know that it is possible to be wrong about our state of health. You may feel great but be on death's door. People drop dead of heart attacks without warning. You go for your annual check-up with no complaints only to have blood tests returned with bad news.

Figure A.6. Resources Screen of DailyMood application

7.Statistic Activity

This is the screenshot of our mini-project's which provides the user with a mood graph or provides a graph to the user to track there feelings.

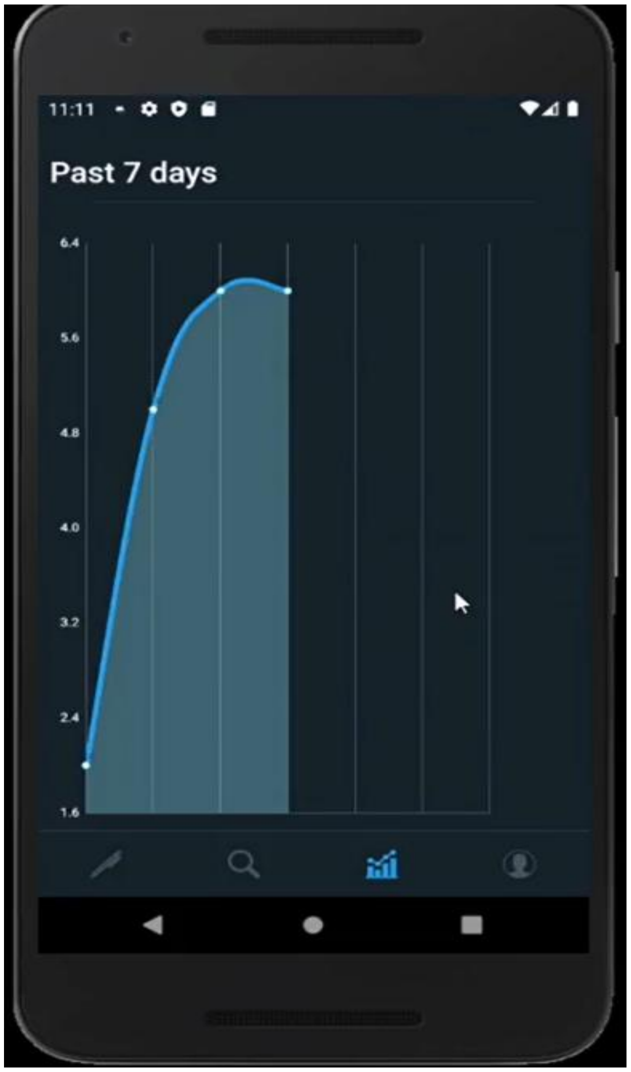
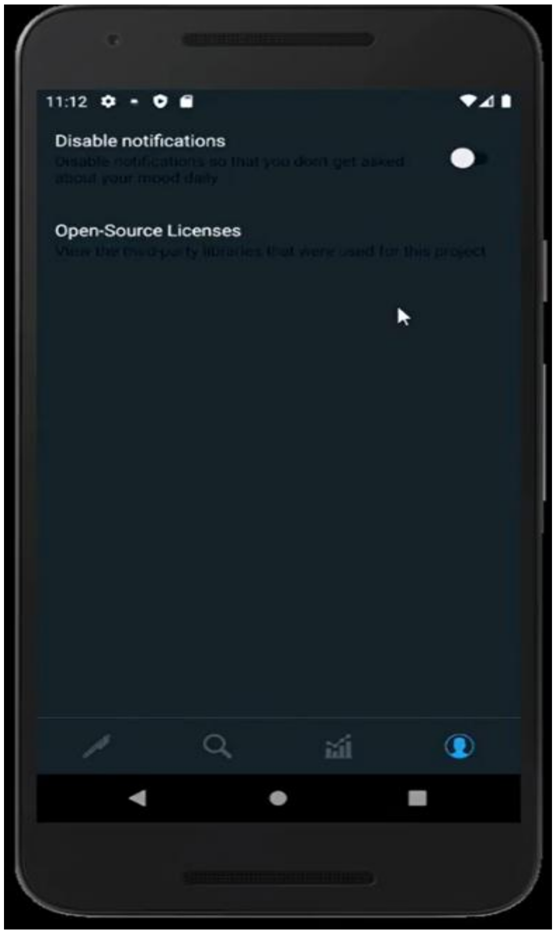


Figure A.7. Statistics Screen of DailyMood application

8.Setting Activity

It connects the user to the notification activity where the user can choose to enable or disable the activity.



11:12

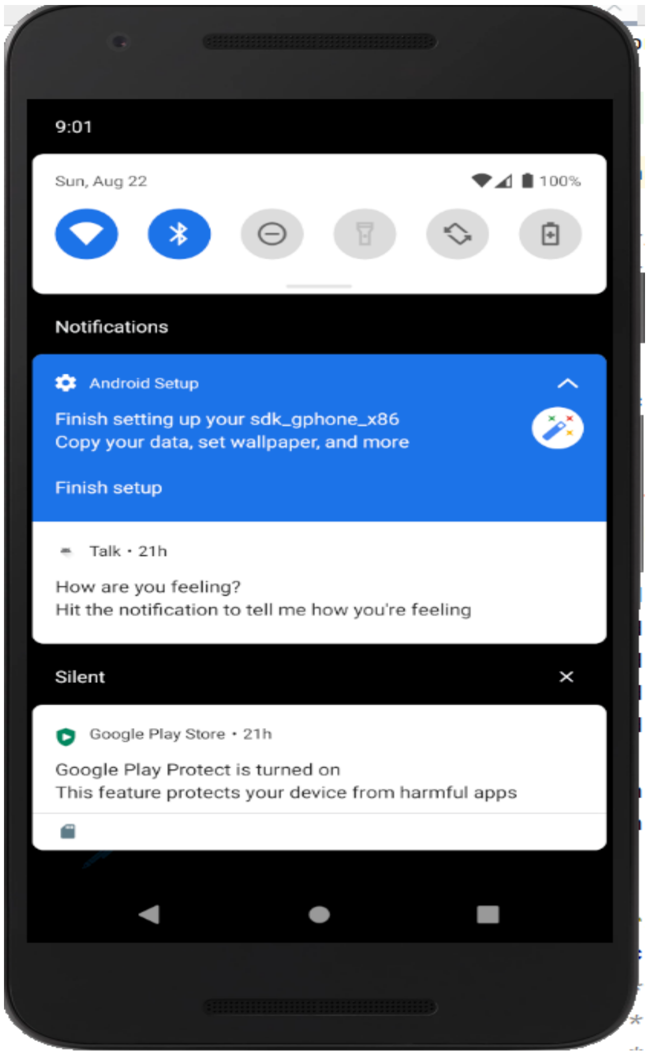
Disable notifications
Disable notifications so that you don't get asked about your mood daily

Open-Source Licenses
View the third-party libraries that were used for this project

Figure A.8. Setting Screen of DailyMood application

9.Notification Activity

This is the screenshot of our mini-project's notification activity which provides the user with notifications which helps the user to track down there feeling without fail. Its done by enabling notification activity in the setting panel.



9:01

Sun, Aug 22 100%

Notifications

Android Setup

Finish setting up your sdk_gphone_x86
Copy your data, set wallpaper, and more

Finish setup

Talk • 21h

How are you feeling?
Hit the notification to tell me how you're feeling

Silent

Google Play Store • 21h

Google Play Protect is turned on
This feature protects your device from harmful apps

Figure A.9. Notification Screen of DailyMood application

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANA SANGAMA, Belgaum - 590 014.

2020 - 2021

A
Mini project report on

"Piechester N Co"

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

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

Submitted by
BELINDA SHARYL BENJAMIN
(1AT18IS016)

MICHELLE MARIA THOMAS
(1AT18IS052)

Under the guidance of
Ms. Uzma Sulthana
Assistant Professor
Dept. of ISE, ATRIA I. T.
&
Ms. Prapulla G
Assistant Professor
Dept. of ISE, ATRIA I. T.

ATRIA INSTITUTE OF TECHNOLOGY
Department of Information Science and Engineering,
Bengaluru - 560 024

ATRIA INSTITUTE OF TECHNOLOGY
(Affiliated to Visvesvaraya Technological University)
ASKB Campus, Anandnagar,
Bengaluru – 560024

Department of Information Science and Engineering

CERTIFICATE

Certified that the project work entitled "**Piechester N Co**" carried out by **BELINDA SHARYL BENJAMIN** bearing USN : 1AT18IS016 and **MICHELLE MARIA THOMAS** bearing USN : 1AT18IS052, the bonafide students of Department of Information Science and Engineering, Atria I. T., in partial fulfillment for the award of **Bachelor of Engineering** in Information Science & Engineering of the Visvesvaraya Technological University, Belgavi during the year 2020-2021. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Ms. Uzma Sulthana/	**Dr. Shanthi Mahesh**	**Dr. T.N Sreenivasa**
Ms. Prapulla G	Head of Department	Principal
Asst. Professor - Project Guide	Department of I.S.E.,	Atria I.T.
Department of I.S.E.,	Atria I.T.	
Atria I. T.		

External Viva

Name of Examiners	**Signature with date**
1.	
2.	

DECLARATION

We, Belinda Sharyl Benjamin (USN : 1AT18IS016) & Michelle Maria Thomas (USN : 1AT18IS052), Students of sixth semester, Bachelor of Engineering, Atria Institute of Technology hereby declare that the mini project entitled **"Piechester N Co"** has been carried out by us at Atria Institute of Technology, Bengaluru and submitted in partial fulfillment of the course requirements for the award of the degree of **Bachelor of Engineering** in **Information Science & Engineering** of **Visvesvaraya Technological University, Belgavi**, during the academic year 2020-2021.

We also declare that, to the best of our knowledge and belief, the work reported here doesn't from part of any other dissertation based on which a degree or award was conferred on an earlier occasion on this by any other student.

Place: **BELINDA SHARYL BENJAMIN**

Date: **(USN : 1AT18IS016)**

MICHELLE MARIA THOMAS

(USN : 1AT18IS052)

ABSTRACT

The application for ordering pies: ‘Piechester N Co’; is helpful for the people to order customized pies on the go with the busy life. Currently it is difficult to physically go to a store and order food we like due to the rise in restrictions. Our app, Piechester N Co helps them by providing that option online. There are various activities like getting started to the app, showing the menu, their category, selecting various items, and the checkout feature.

It has a login page where the user can get themselves an account created by entering a unique username and specifying a password satisfying the conditions.

The user can use the application once logged in using the registered credentials. The user can have a smooth experience using the mobile application. Once logged in there is no need to log in again unless the user clicks on Check out.

ACKNOWLEDGEMENT

We are grateful to our institution, **Atria Institute of Technology**, for having provided us with the facilities to successfully complete this mini project on Piechester N Co.

We thank **Dr. T. N. Sreenivasa, principal** and **Dr. Shanthi Mahesh, HOD, ISE** for providing us all the necessary facilities for successful completion of our mini project.

Deadlines play a very important role in successful completion of the academic project on time, efficiently and effectively. We take this opportunity to express our deep sense of gratitude to our guide and coordinators **Ms. Uzma Sulthana, Assistant Professor and Ms. Prapulla G, Assistant Professor, Department of ISE** for their valuable guidance and help throughout the course of the academic mini project. They have always been patient with us and helped immensely in completing the task on hand. We also thank them for the immense support, guidance, specifications & ideas without which seminar would have been completed without full merit.

Last but not least from the Department of Information Science and Engineering, teaching and non-teaching staffs for their constant encouragement, support, patience and endurance shown during the preparation of this report were remarkable. We also thank the management.

Finally, we thank our parents and friends for their motivation, moral and material support.

BELINDA SHARYL BENJAMIN
(USN: 1AT18IS016)

MICHELLE MARIA THOMAS
(USN: 1AT18IS052)

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction to Mobile Application Development

Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources. Like web **application development, mobile application development** has its roots in more traditional software development.

1.1.1 History:

- The first mobile phones were invented whose microchips required the most basic software to send and receive voice calls.
- On 3rd of April 1973, Martin Cooper of Motorola made the first call on the mobile phone to Dr. Joel S. Engel of the Bell Labs.
- The R&D department of IBM Simon came up with the first mobile app for Smartphones in 1993 exactly two decades after the first call was made.
- EPOC, first operating system developed by Psion, released in the early 90s, this was first of the recognizable apps.
- Palm OS, developed by Palm Inc. in the year 1996, these were mainly designed for personal digital assistants and were known as Garnet OS.
- The wireless markup language was specifically designed for devices that were dependent on XML and could be run across wireless application protocols.
- Java ME or J2ME or JME – it was first introduced as JSR 68. It was given various shapes and forms for use via Phones, embedded devices, and even PDAs.
- Symbian, developed by Symbian Ltd, which was a joint venture from Ericsson, Motorola, Nokia and PSION, this was a further developed version of PSION EPOC.
- Later on, the smartphones and iPhones that we use today evolved, making lives a lot easier for people.

1.1.2 Advantages:

- Improves Efficiency.
- Offers High Scalability.
- Secures the App Data.
- Integrates With Existing Software.
- Easy to Maintain.
- Improves Customer Relationship.
- Facilitates New Client Data Retrieval.
- Provides Real-time Project Access.
- Ease in Project Management.

1.2 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA software. It provides the fastest tools for building apps on every type of android device. It is a purpose-built for android to accelerate the development and helps to build the highest-quality apps for every android device. Features of Android Studio include,

- A flexible Gradle-based build system.
- A fast and feature-rich emulator.
- A unified environment where one can develop for all Android devices.
- Extensive testing tools and frameworks.

1.2.1 Android Studio SDK:

Android SDK performs all the tasks needed to develop apps for all versions of Android. This program is a necessary tool for any developer who wants to make smoothly running applications for the latest systems. It uses Java for development and relies on the Integrated Development Environment, to build the apps and test them.

1.2.2 Android Studio Emulator:

The Android Emulator simulates Android devices on your computer so that you can test your application on a variety of devices and Android API levels without needing to have each physical device. The emulator provides almost all of the capabilities of a real Android device. Simulation of incoming phone calls and text messages, specify the location of the device, simulate different network speeds, simulate rotation and other hardware sensors, access the Google Play Store, and much more are possible.

1.3 JAVA

Java is an object-oriented programming language created by James Gosling, Mike Sheridan, and Patrick Naughton in 1991. It is a high-level, class-based language that is designed to have a few implementation dependencies as possible. It is a general-purpose programming language intended to let android developers run the compiled Java code on all platforms that support Java without any need for recompilation. Features of Java include,

- **Simple:** Java is designed to be easy to learn.
- **Secure:** With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- **Architecture-neutral:** Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- **Portable:** Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- **Robust:** Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking.
- **Multithreaded:** With Java's multithreaded feature it is possible to write programs
- that can perform many tasks simultaneously.
- **Interpreted:** Java byte code is translated on the fly to native machine instructions and is not stored anywhere.

- **High Performance:** With the use of Just-In-Time compilers, Java enables high performance.
- **Distributed:** Java is designed for the distributed environment of the internet.
- **Dynamic**: Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time.

1.4 XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML focus on simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Features of XML include,

- XML focuses on data rather than how it looks.
- Easy and efficient data sharing.
- Compatibility with other markup language HTML.
- Supports platform transition.
- Allows XML validation.
- Adapts technology advancements.
- XML supports Unicode.

1.5 STRUCTURE OF REPORT

This report is for our mini project Piechester N Co using Mobile Application Development concepts. Our report consists of five chapters where in first chapter we are giving introduction to mobile application development with its history. In second chapter we have given brief description of our problem definition and literature survey made. Similarly in third chapter we have a brief document on the requirement specifications of hardware and software required along with purpose and scope of our project. The fourth and fifth module gives us the design and implementation knowledge. To conclude we have added our conclusions and further enhancement. We have also mentioned the references to our project. Last but not the least we have the screen shots of our output showing execution of our program in appendix.

CHAPTER 2

LITERATURE SURVEY

2.1. Problem Definition

Piechester N Co is a mobile application for ordering pies of different flavours. There are plenty of bakeries and eateries in Bangalore, but somehow the one underrated food item seems to be pies. This is strange considering the range and variety that pies have to offer. Pies can be sweet, savoury, sweet-and-savoury, spicy, smoky and more; basically fit for any and every mood and occasion. The activities of this app include Sign in/ log in feature, well-designed menu and cart.

2.2. Purpose & Scope

In the application we can view various items available for ordering in an enhanced manner. The customer can also mention any additional toppings they may like on their pies. Another advantage of the application is that it is dynamic, allows easy viewing and the feature to delete items off the cart as per choice. Our application has several advantages.

Advantages:

i. User friendly interface
ii. Sign-In feature
iii. Scrollable menu
iv. Cart for storing items for later purchase

2.3. Aim of the application

Our mobile application, titled Piechester N Co aims to make popular this dish (pies), traced all the way back to the ancient Egyptians. ‘Piechester n Co’ aims to bring popularity to this dish in our current society, by providing a means of easy purchase of pies, that is, via mobile application. These rich, buttery dishes, that can serve as a mini-meal or dessert is guaranteed to find a place in many a mouth. This user friendly application is sure to help achieve this aim.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1. Introduction

3.1.1. Purpose

In the application we can view various pies available in the menu for purchase, at reasonable prices, and order them for personal or gifting purposes. The purpose of this app is to raise the popularity of these delicious baked goods, allowing customers to purchase them from the comfort and safety of their homes, and regardless of their location within the city.

3.1.2. Scope

This mobile application shows exactly what the pies you will purchase look like, with a vivid and accurate description of what it will taste like.

3.1.3. Definition, Acronyms and Abbreviations

1. XML - Extensible Markup Language
2. MS – Microsoft
3. IDE - Integrated Development Environment
4. SDK – Software Development Kit

3.2. Development Environment

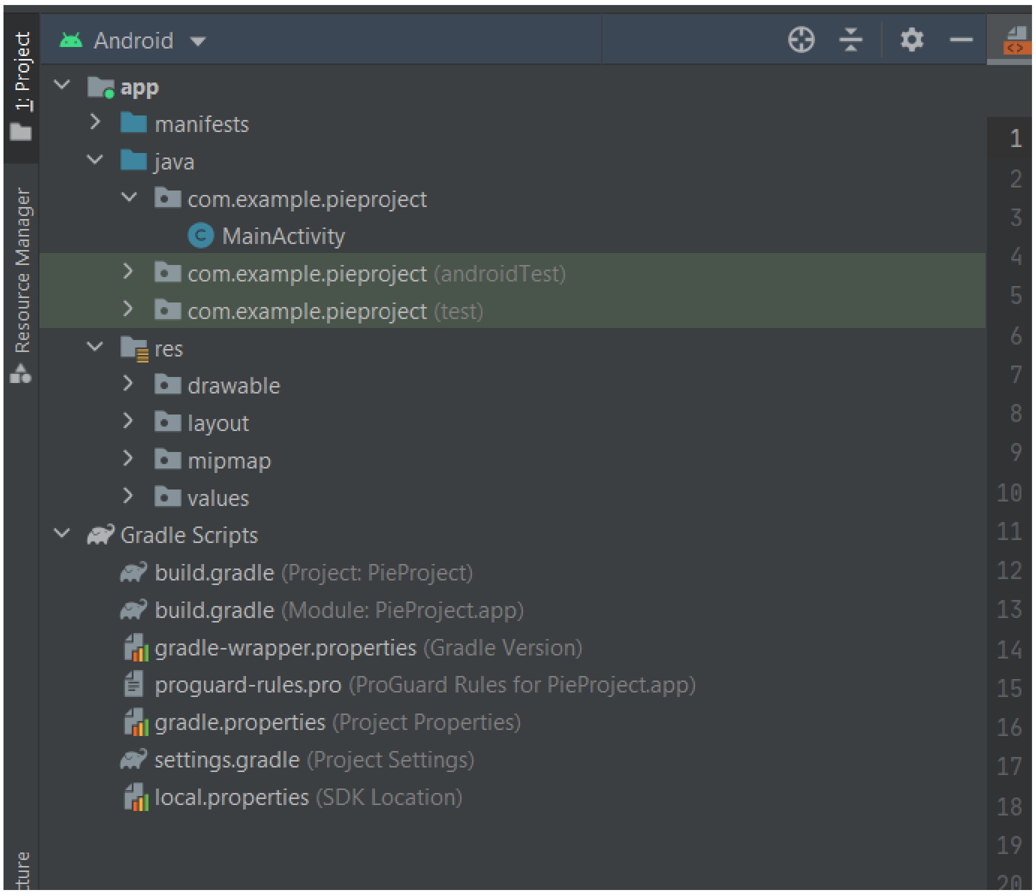
3.2.1. Android Programming Languages

In Android, programming is done in two languages JAVA or C++ and XML (Extension Markup Language). Nowadays KOTLIN is also preferred. The XML file deals with the design, presentation, layouts, blueprint, etc. (as a front-end) while the JAVA or KOTLIN deals with the working of buttons, variables, storing, etc. (as a back-end).

3.2.2. Android Components

- **Activities**: It deals with the UI and the user interactions to the screen. In other words, it is a User Interface that contains activities.
- **Services**: Services are the background actions performed by the app; these might be long- running operations. A service might need other sub-services so as to perform specific tasks.
- **Content Provider:** Content Provider is used to transferring the data from one application to the others at the request of the other application.
- **Broadcast Receivers:** A Broadcast is used to respond to messages from other applications or from the System.

3.2.3. Structural Layout of Android



```
Android
app
  manifests
  java
    com.example.pieproject
      MainActivity
    com.example.pieproject (androidTest)
    com.example.pieproject (test)
  res
    drawable
    layout
    mipmap
    values
Gradle Scripts
  build.gradle (Project: PieProject)
  build.gradle (Module: PieProject.app)
  gradle-wrapper.properties (Gradle Version)
  proguard-rules.pro (ProGuard Rules for PieProject.app)
  gradle.properties (Project Properties)
  settings.gradle (Project Settings)
  local.properties (SDK Location)
```

Figure 3.2.3 Structural Layout of Android Studio

- **Manifest Folder**: Android Manifest is an XML file that is the root of the project source set. It describes the essential information about the app and the Android build tools, the Android Operating System, and Google Play. It contains the permission that an app might need in order to perform a specific task. It also contains the Hardware and the Software features of the app, which determines the compatibility of an app on the Play Store.

- **Java Folder:** The JAVA folder consists of the java files that are required to perform the background task of the app. It consists of the functionality of the buttons, calculation, storing, variables, toast (small popup message), programming function, etc. The number of these files depends upon the type of activities created.

- **Resource Folder:** The res or Resource folder consists of the various resources that are used in the app. This consists of sub-folders like drawable, layout, mipmap, raw, and values. The drawable consists of the images. The layout consists of the XML files that define the user interface layout. These are stored in res.layout and are accessed as R.layout class. The raw consists of the Resources files like audio files or music files, etc. These are accessed through R.raw.filename.values are used to store the hardcoded strings (considered safe to store string values) values, integers, and colors.

- **Gradle Files:** Gradle is an advanced toolkit, which is used to manage the build process that drawable consists of the images. The layout consists of the XML files that define the user interface layout. These are stored in res.layout and are accessed as R.layout class. The raw consists of the Resources files like audio files or music files, etc. These are accessed through R.raw.filename.values are used to store the hardcoded strings (considered safe to store string values) values, integers, and colors.

3.3. Specific Requirements

3.3.1 Software Requirements

i. Windows 10 Operating System
ii. **Tool kit**: Android SDK (Software development kit), Java development kit (JDK)
iii. **IDE**: Android Studio

3.3.2. Hardware Requirements

i. 1.8 GHz Processor
ii. 8GB (IDE + Android SDK + Android Emulator)/ 4GB (minimum) RAM
iii. 15 inches Monitor
iv. 104 keys with keyboard and mouse
v. 6.3 inches Physical Android Device

CHAPTER 4

DESIGN

4.1. Project Flow

In our project Piechester N Co, we have used XML with different attributes and a code written in Java language for various activities.

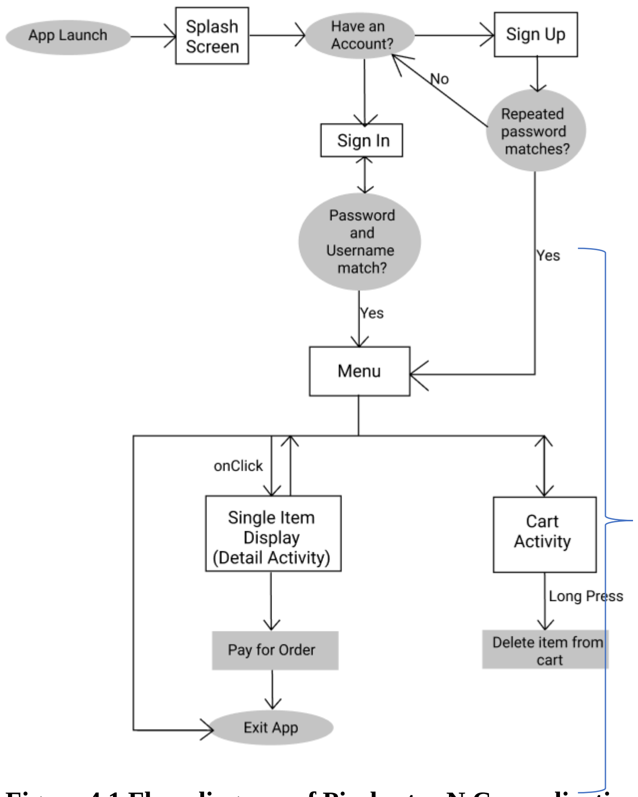


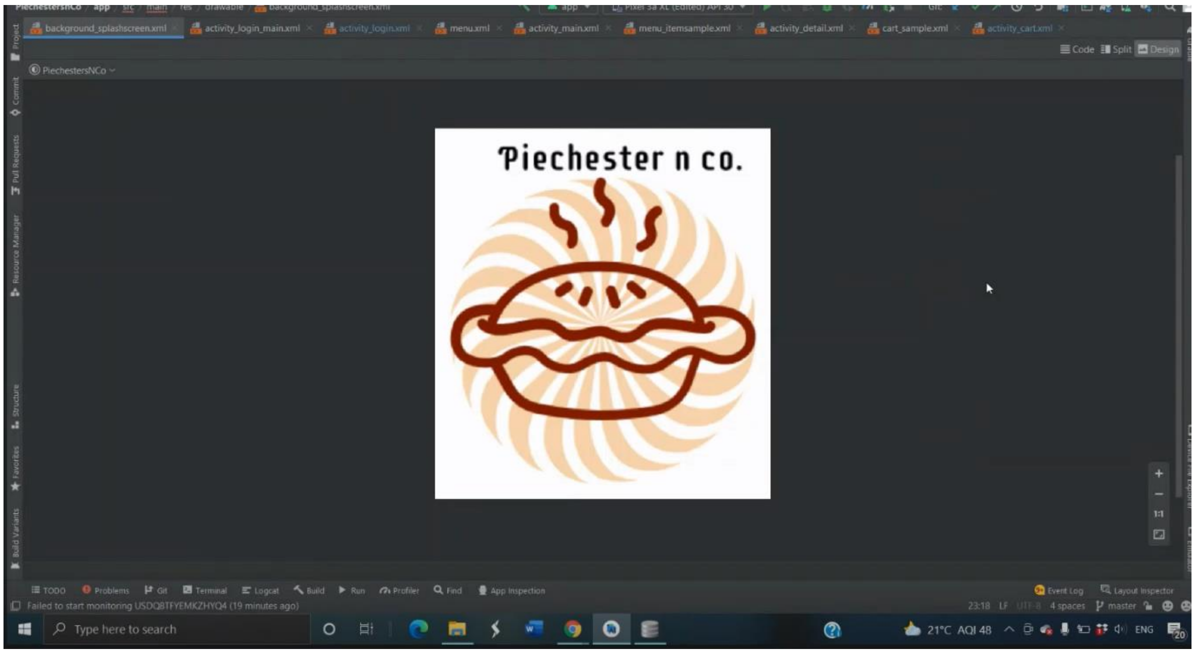
Figure 4.1 Flow diagram of Piechester N Co application

Once the user opens the application on the android device, splash screen with the app logo is displayed for few seconds and is navigated automatically to the login activity. The user can register in case if not registered using the ‘Sign Up’ option provided and if the user is new to the app. On successful login the menu is displayed. The users are provided with few features on the navigation drawer through which one can Select an item by scrolling through the menu, selecting and then placing orders. The users can Log Out after clicking on the ‘PAY’ button provided on the cart screen.

4.2. Design using XML

4.2.1 activity_main.xml

The application on opening displays the splash screen which is designed in the activity_main.xml.

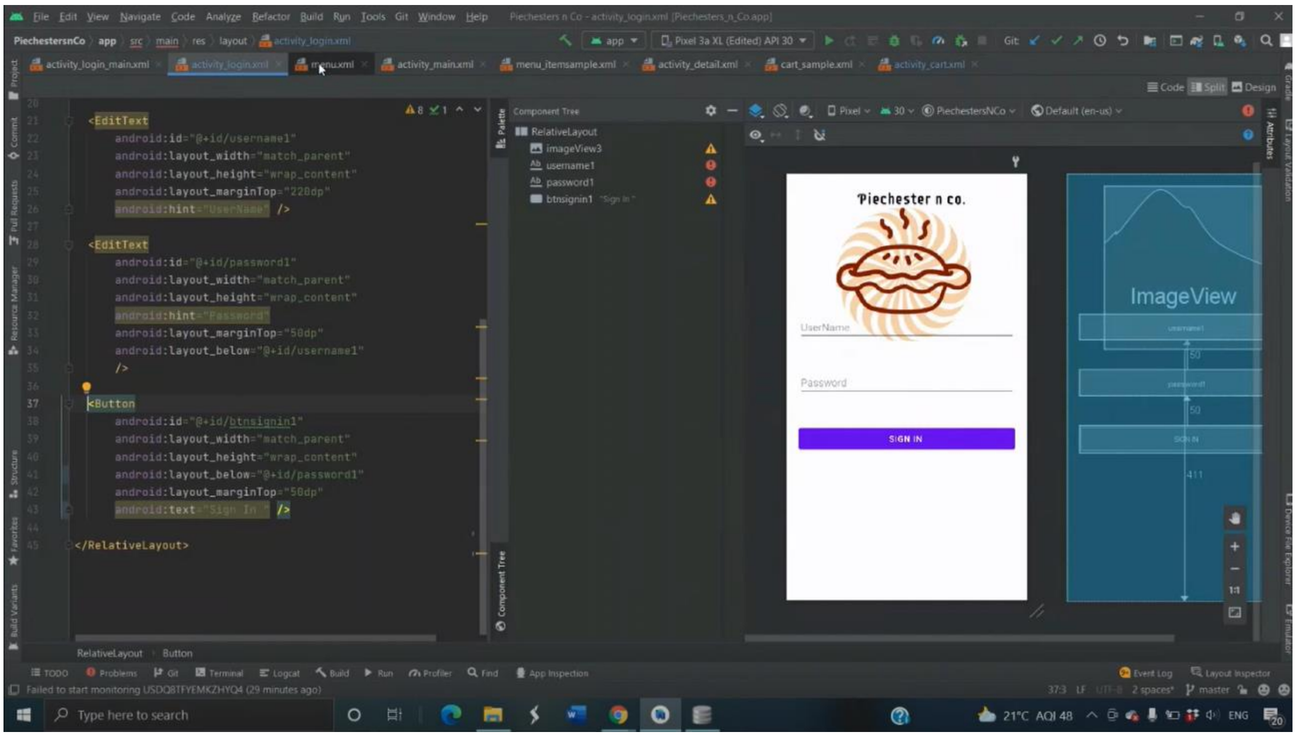


Failed to start monitoring USDQBTFYEMKZHYQ4 (19 minutes ago)

Figure 4.2.1 Design Code screenshot for splash screen

4.2.2 activity_login.xml

The login page is designed in the activity_login.xml where the user can login by entering the registered username and password.



```
    <EditText
        android:id="@+id/username1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="220dp"
        android:hint="UserName" />

    <EditText
        android:id="@+id/password1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:layout_marginTop="50dp"
        android:layout_below="@+id/username1"
        />

    <Button
        android:id="@+id/btnsignin1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/password1"
        android:layout_marginTop="50dp"
        android:text="Sign In " />

</RelativeLayout>
```

Figure 4.2.2 Design Code screenshot for Login

4.2.3 activity_login_main.xml

The user can register through register screen which is designed in the activity_login_main.xml.

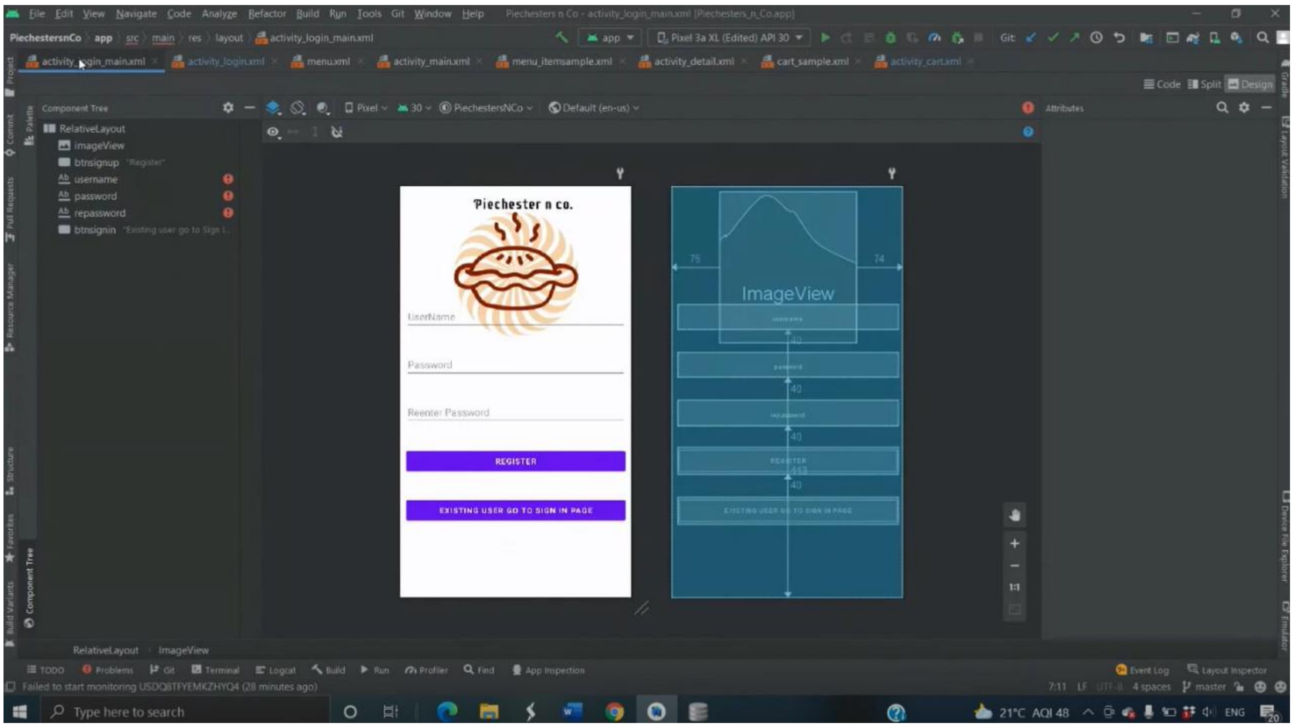
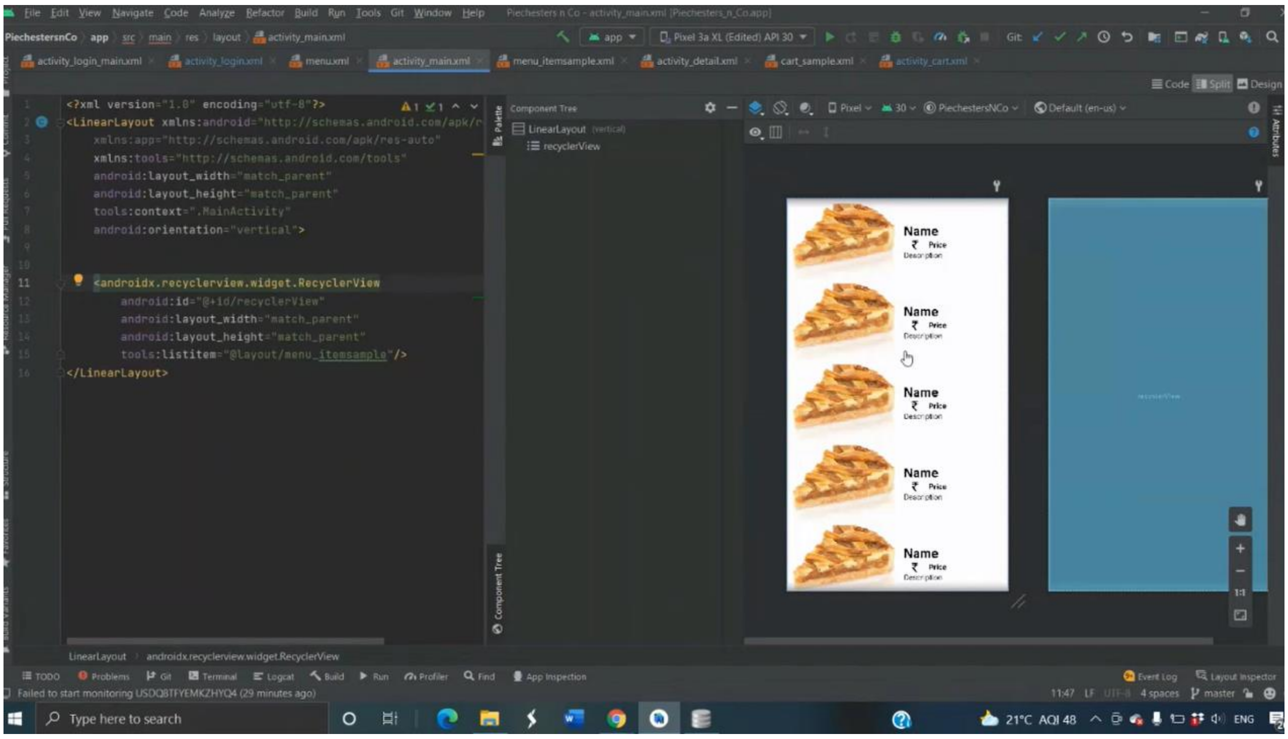


Figure 4.2.3 Design Code screenshot for Signup

4.2.4 menu.xml

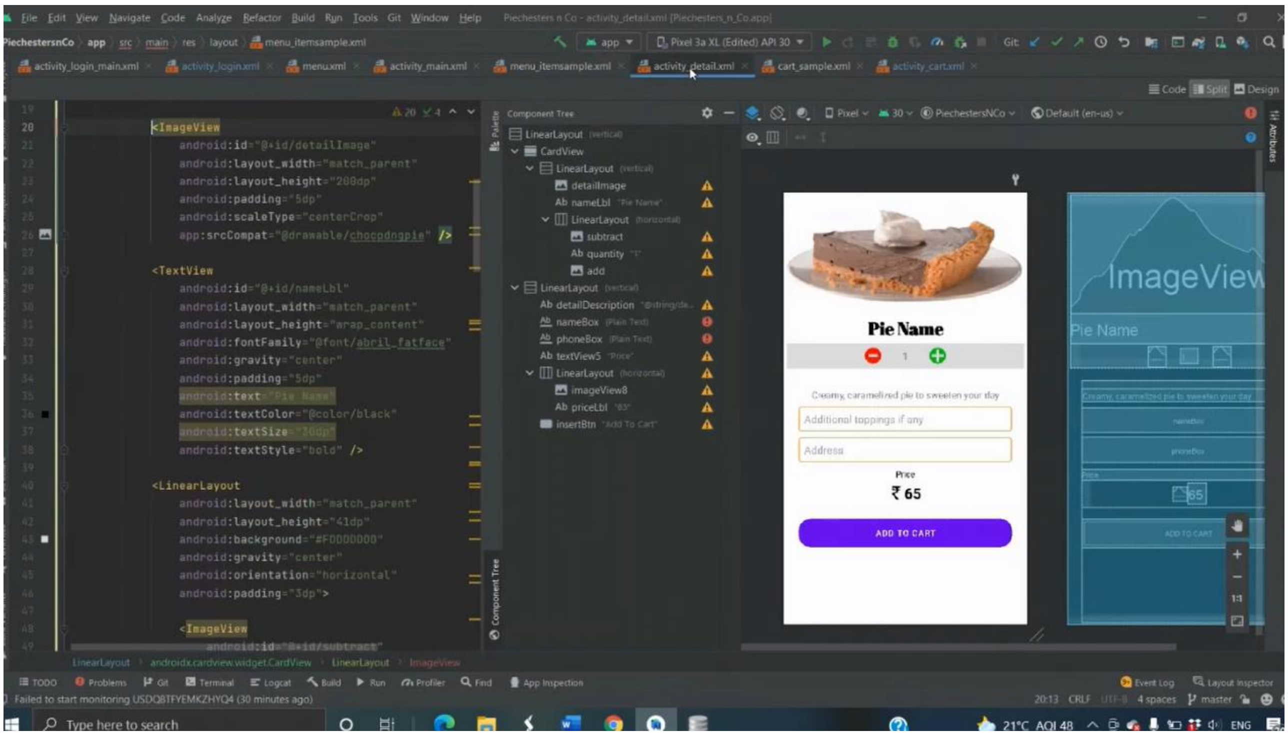
The menu.xml has various card views to display the items available and contains the navigation drawer.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/r
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical">


    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:listitem="@layout/menu_itemsample"/>
</LinearLayout>
```

Figure 4.2.4 Design Code screenshot of Menu

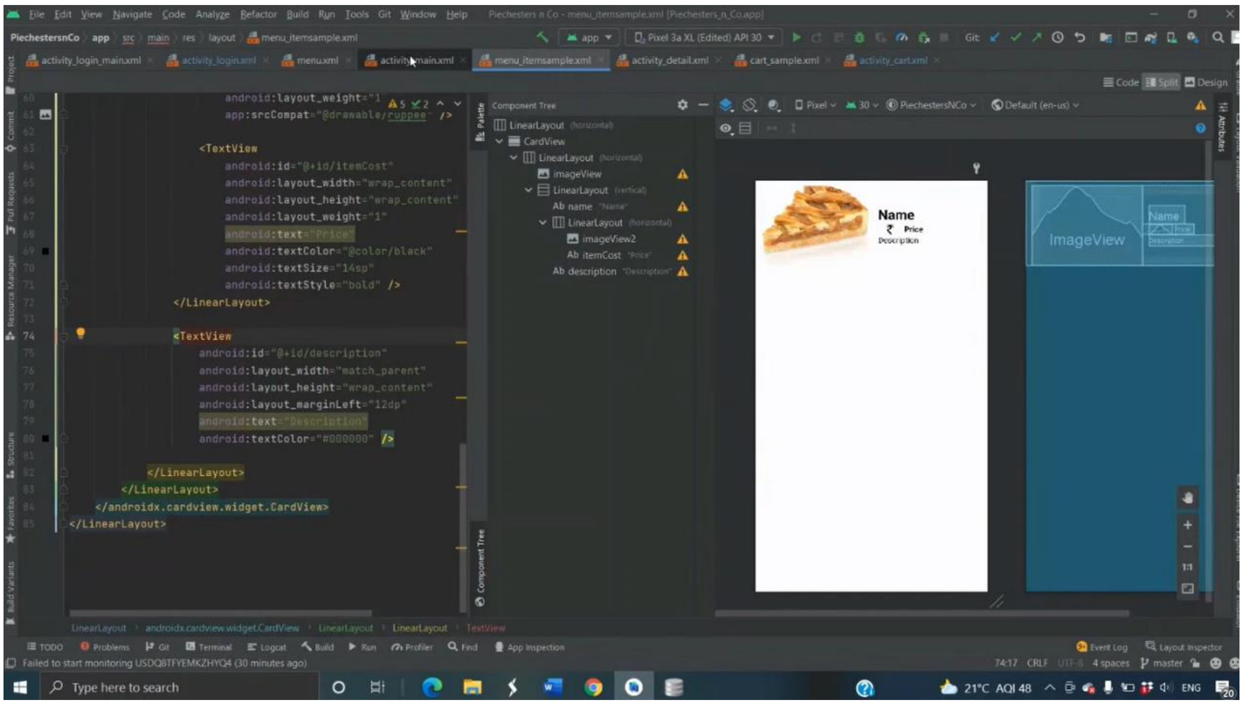
4.2.5 activity_detail.xml



```

            <ImageView
                android:id="@+id/detailImage"
                android:layout_width="match_parent"
                android:layout_height="200dp"
                android:padding="5dp"
                android:scaleType="centerCrop"
                app:srcCompat="@drawable/chocodngpie" />

            <TextView
                android:id="@+id/nameLbl"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:fontFamily="@font/abril_fatface"
                android:gravity="center"
                android:padding="5dp"
                android:text="Pie Name"
                android:textColor="@color/black"
                android:textSize="30dp"
                android:textStyle="bold" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="41dp"
                android:background="#FDDDDDDD"
                android:gravity="center"
                android:orientation="horizontal"
                android:padding="3dp">

                <ImageView
                    android:id="@+id/subtract"
```

Figure 4.2.5 Design Code screenshot of Details

4.2.6 Menu_itemsample.xml

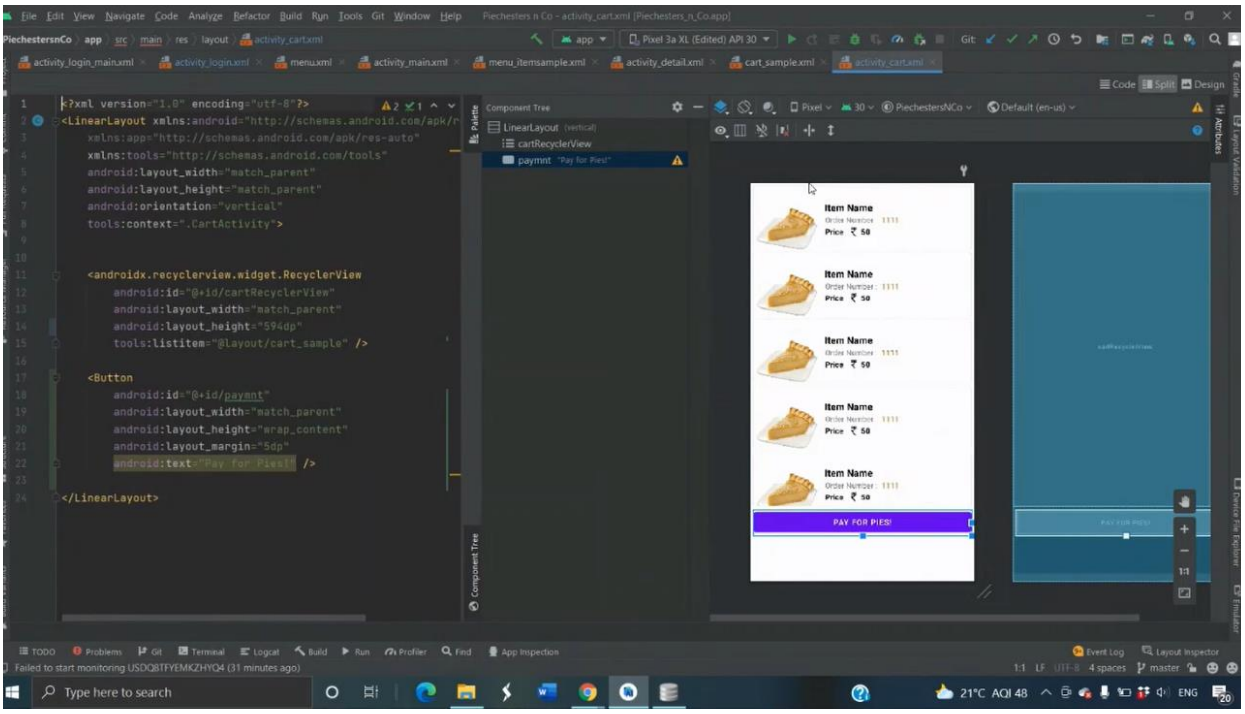


```
                    android:layout_weight="1
                    app:srcCompat="@drawable/ruppee" />

                <TextView
                    android:id="@+id/itemCost"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    android:text="Price"
                    android:textColor="@color/black"
                    android:textSize="14sp"
                    android:textStyle="bold" />
            </LinearLayout>

            <TextView
                android:id="@+id/description"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginLeft="12dp"
                android:text="Description"
                android:textColor="#000000" />

        </LinearLayout>
    </LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>
```

Figure 4.2.6 Code screenshot of menu item

4.2.7 activity_cart.xml

The string file is used to initialized strings and fetch using it's id.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/r
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".CartActivity">


    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/cartRecyclerView"
        android:layout_width="match_parent"
        android:layout_height="594dp"
        tools:listitem="@layout/cart_sample" />

    <Button
        android:id="@+id/paymnt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="Pay for Pies!" />

</LinearLayout>
```

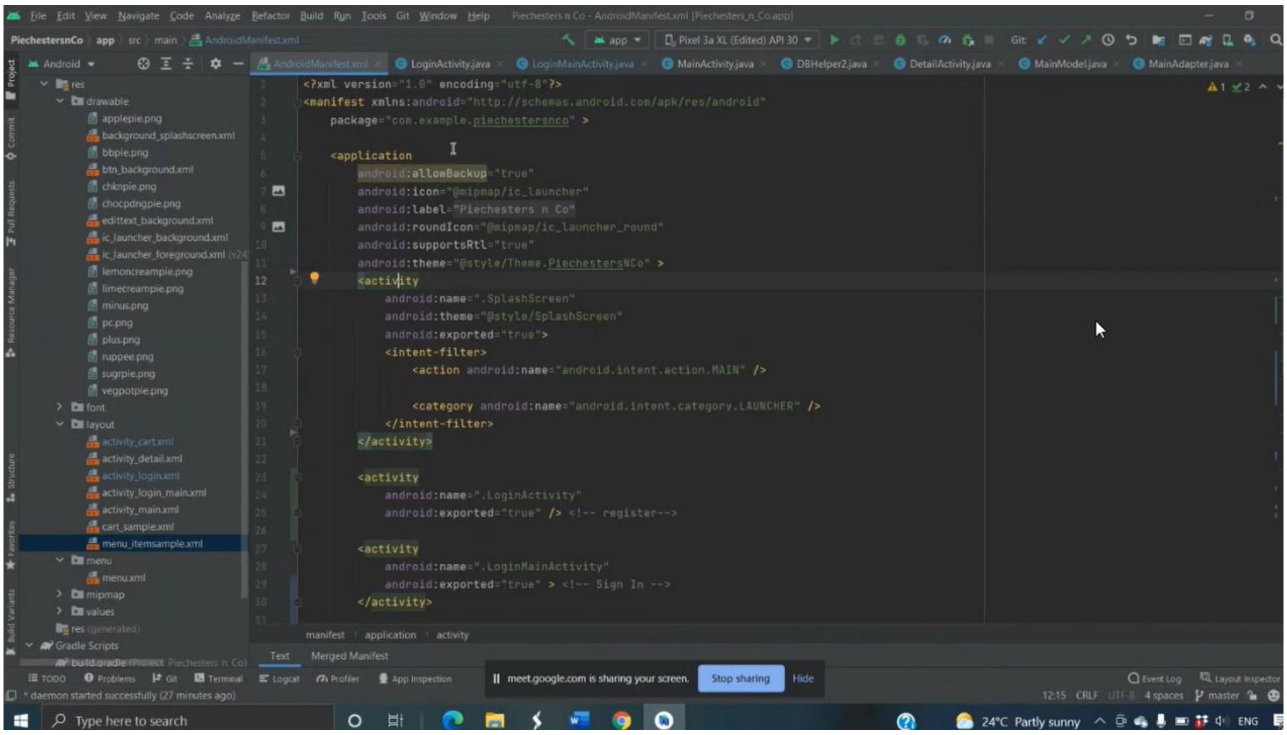
Figure 4.2.7 Design Code screenshot of Activity_cart

CHAPTER 5

IMPLEMENTATION

5.1. AndroidManifest.xml

The AnndroidManifest.xml file contains the splash screen where the logo is displayed for 2 seconds and is parsed to the login screen.

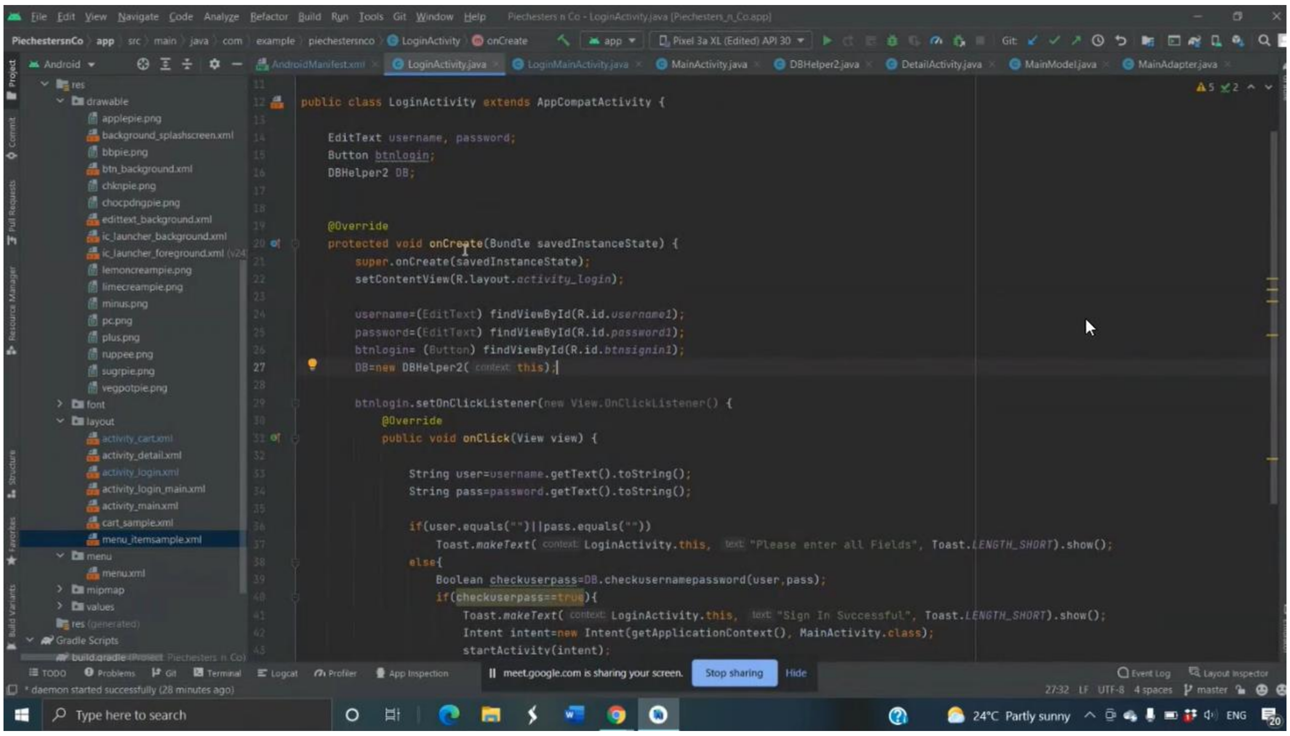


```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.piechestersnco" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Piechesters n Co"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.PiechestersNCo" >
        <activity
            android:name=".SplashScreen"
            android:theme="@style/SplashScreen"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
            android:name=".LoginActivity"
            android:exported="true" /> <!-- register-->

        <activity
            android:name=".LoginMainActivity"
            android:exported="true" > <!-- Sign In -->
        </activity>
```

Figure 5.1 Code screenshot for AndroidManifest.xml

5.2. LoginActivity.java

The Login screen contains the code to parse to sign up, get started and the home activity on successful login. Login credentials are connected to the DBHelper java file to save the credentials in the database.



```java
public class LoginActivity extends AppCompatActivity {

    EditText username, password;
    Button btnlogin;
    DBHelper2 DB;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        username=(EditText) findViewById(R.id.username1);
        password=(EditText) findViewById(R.id.password1);
        btnlogin= (Button) findViewById(R.id.btnsignin1);
        DB=new DBHelper2( this);

        btnlogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                String user=username.getText().toString();
                String pass=password.getText().toString();

                if(user.equals("")||pass.equals(""))
                    Toast.makeText( LoginActivity.this, "Please enter all Fields", Toast.LENGTH_SHORT).show();
                else{
                    Boolean checkuserpass=DB.checkusernamepassword(user,pass);
                    if(checkuserpass==true){
                        Toast.makeText( LoginActivity.this, "Sign In Successful", Toast.LENGTH_SHORT).show();
                        Intent intent=new Intent(getApplicationContext(), MainActivity.class);
                        startActivity(intent);
```

Figure 5.2 Code screenshot for LoginActivity.java

5.3. Online SQLite DB viewer

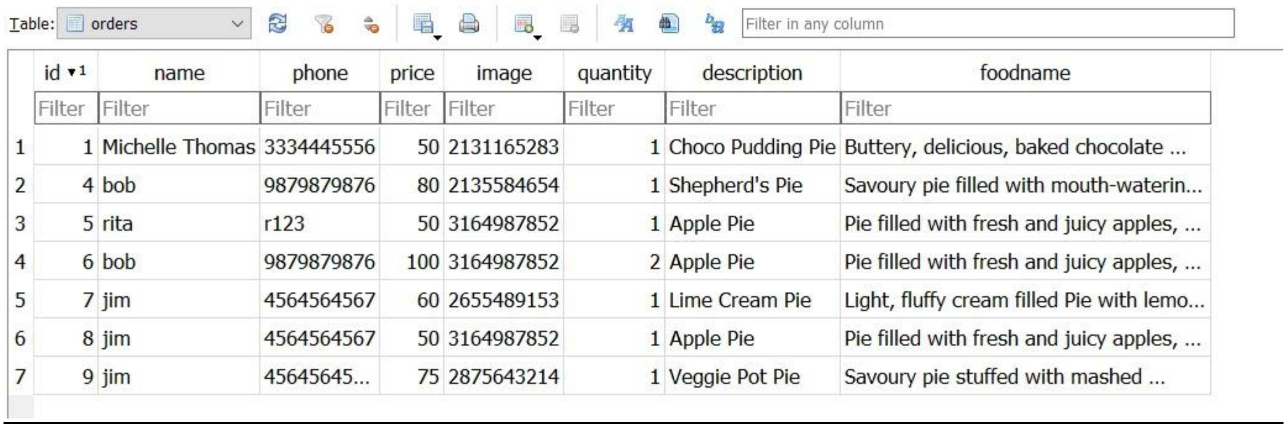


Table: orders

Filter in any column

	id	name	phone	price	image	quantity	description	foodname
1	1	Michelle Thomas	3334445556	50	2131165283	1	Choco Pudding Pie	Buttery, delicious, baked chocolate ...
2	4	bob	9879879876	80	2135584654	1	Shepherd's Pie	Savoury pie filled with mouth-waterin...
3	5	rita	r123	50	3164987852	1	Apple Pie	Pie filled with fresh and juicy apples, ...
4	6	bob	9879879876	100	3164987852	2	Apple Pie	Pie filled with fresh and juicy apples, ...
5	7	jim	4564564567	60	2655489153	1	Lime Cream Pie	Light, fluffy cream filled Pie with lemo...
6	8	jim	4564564567	50	3164987852	1	Apple Pie	Pie filled with fresh and juicy apples, ...
7	9	jim	45645645...	75	2875643214	1	Veggie Pot Pie	Savoury pie stuffed with mashed ...

Figure 5.3.1 Screenshot showing the order details

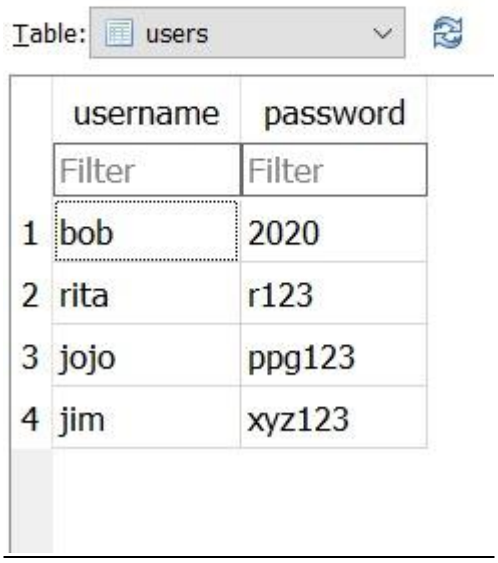


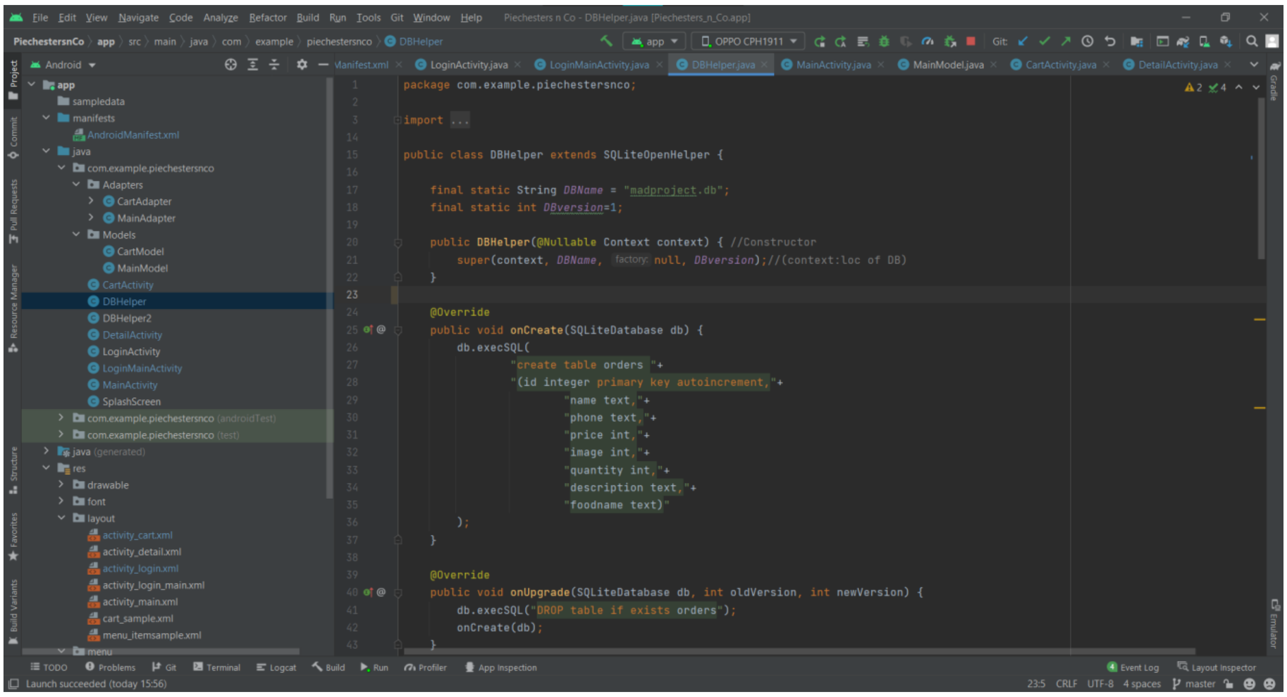
Table: users

	username	password
1	bob	2020
2	rita	r123
3	jojo	ppg123
4	jim	xyz123

Figure 5.3.2 Screenshot showing the username and the password details

5.4. DBHelper.java

The DBHelper.java file helps in connecting the Login and check validity of the credentials.

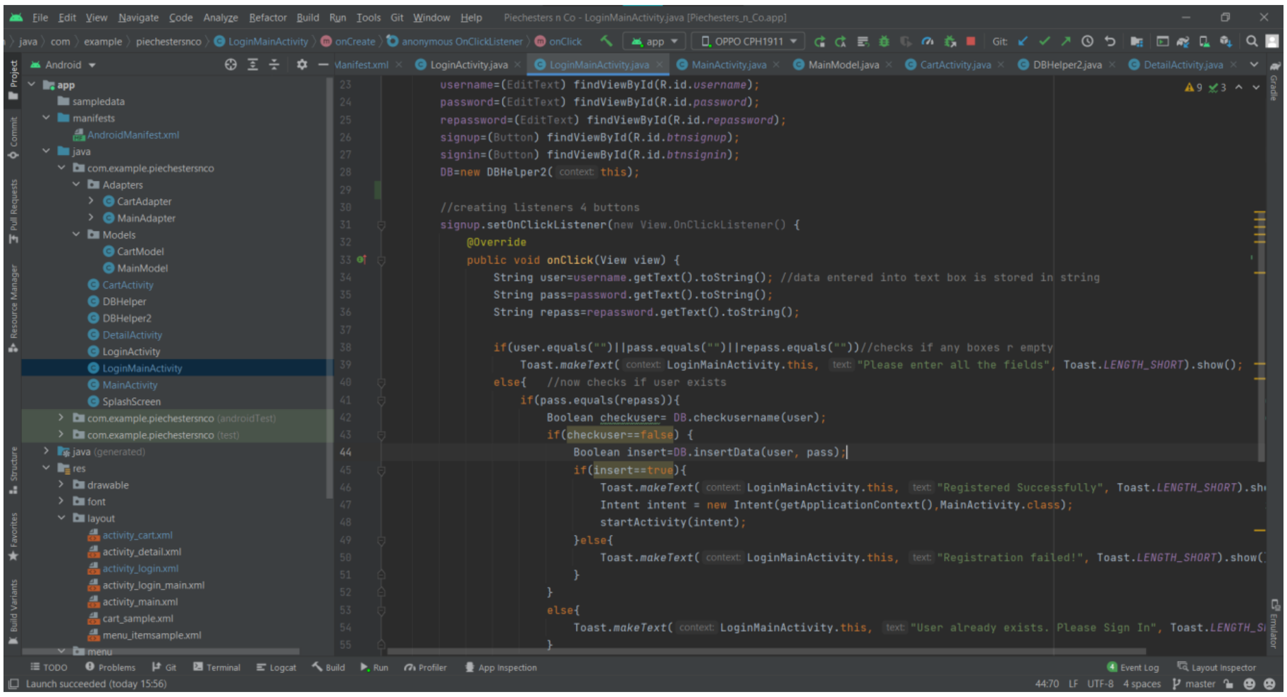


```
package com.example.piechestersnco;

import ...

public class DBHelper extends SQLiteOpenHelper {

    final static String DBName = "madproject.db";
    final static int DBversion=1;

    public DBHelper(@Nullable Context context) { //Constructor
        super(context, DBName,  factory: null, DBversion);//(context:loc of DB)
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(
                "create table orders "+
                "(id integer primary key autoincrement,"+
                        "name text,"+
                        "phone text,"+
                        "price int,"+
                        "image int,"+
                        "quantity int,"+
                        "description text,"+
                        "foodname text)"
        );
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP table if exists orders");
        onCreate(db);
    }
```

Figure 5.4 Code screenshot of DBHelper.java

5.5. LoginMainActivity.java

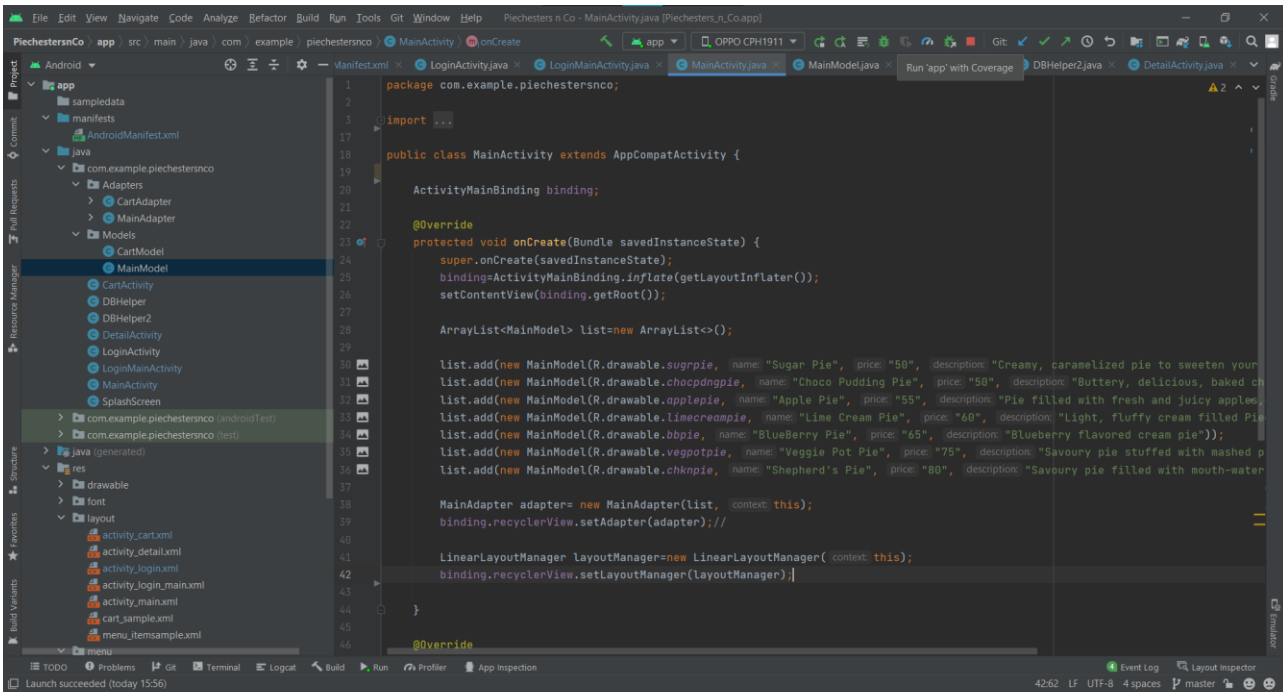
The Signup screen contains the code to parse to login screen after registration.



```
        username=(EditText) findViewById(R.id.username);
        password=(EditText) findViewById(R.id.password);
        repassword=(EditText) findViewById(R.id.repassword);
        signup=(Button) findViewById(R.id.btnsignup);
        signin=(Button) findViewById(R.id.btnsignin);
        DB=new DBHelper2( context: this);

        //creating listeners 4 buttons
        signup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String user=username.getText().toString(); //data entered into text box is stored in string
                String pass=password.getText().toString();
                String repass=repassword.getText().toString();

                if(user.equals("")||pass.equals("")||repass.equals(""))//checks if any boxes r empty
                    Toast.makeText( context: LoginMainActivity.this,  text: "Please enter all the fields", Toast.LENGTH_SHORT).show();
                else{   //now checks if user exists
                    if(pass.equals(repass)){
                        Boolean checkuser= DB.checkusername(user);
                        if(checkuser==false) {
                            Boolean insert=DB.insertData(user, pass);
                            if(insert==true){
                                Toast.makeText( context: LoginMainActivity.this,  text: "Registered Successfully", Toast.LENGTH_SHORT).sh
                                Intent intent = new Intent(getApplicationContext(),MainActivity.class);
                                startActivity(intent);
                            }else{
                                Toast.makeText( context: LoginMainActivity.this,  text: "Registration failed!", Toast.LENGTH_SHORT).show(
                            }
                        }
                        else{
                            Toast.makeText( context: LoginMainActivity.this,  text: "User already exists. Please Sign In", Toast.LENGTH_S
                        }
```

Figure 5.5 Code screenshot for LoginMainActivity.java

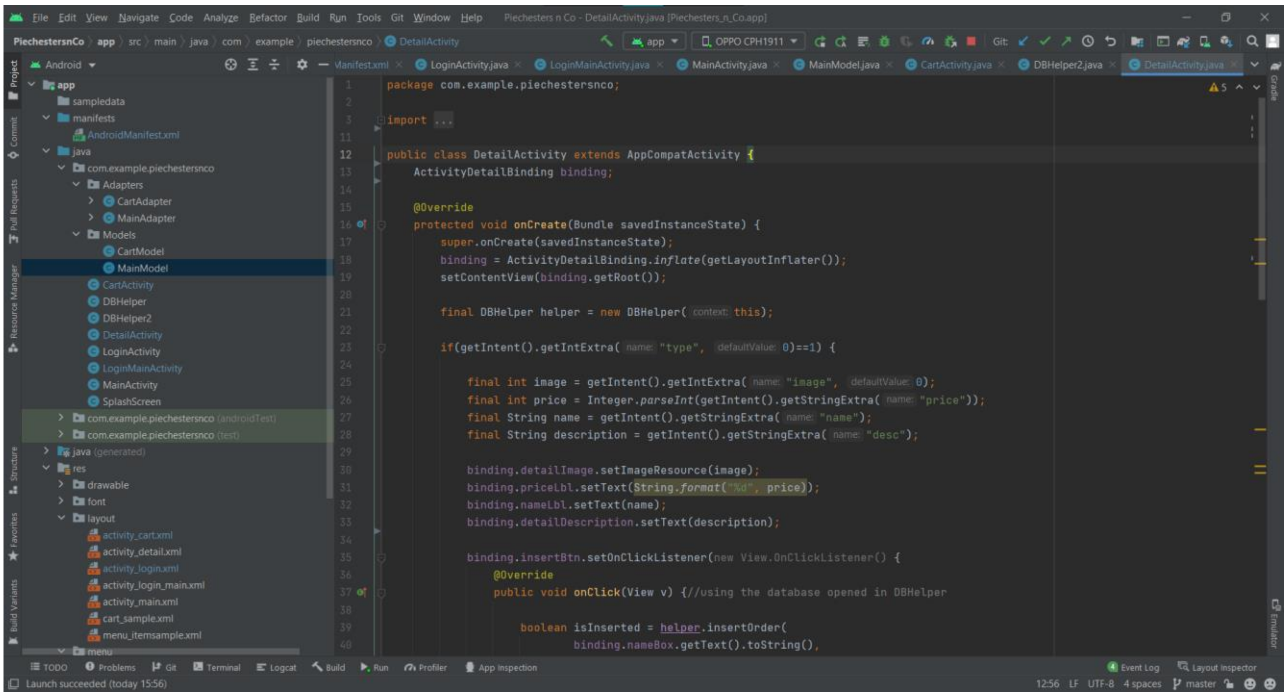
5.6. MainActivity.java



```
package com.example.piechestersnco;

import ...

public class MainActivity extends AppCompatActivity {

    ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding=ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        ArrayList<MainModel> list=new ArrayList<>();

        list.add(new MainModel(R.drawable.sugrpie,  name: "Sugar Pie",  price: "50",  description: "Creamy, caramelized pie to sweeten your
        list.add(new MainModel(R.drawable.chocpdngpie,  name: "Choco Pudding Pie",  price: "50",  description: "Buttery, delicious, baked ch
        list.add(new MainModel(R.drawable.applepie,  name: "Apple Pie",  price: "55",  description: "Pie filled with fresh and juicy apples,
        list.add(new MainModel(R.drawable.limecreampie,  name: "Lime Cream Pie",  price: "60",  description: "Light, fluffy cream filled Pie
        list.add(new MainModel(R.drawable.bbpie,  name: "BlueBerry Pie",  price: "65",  description: "Blueberry flavored cream pie"));
        list.add(new MainModel(R.drawable.vegpotpie,  name: "Veggie Pot Pie",  price: "75",  description: "Savoury pie stuffed with mashed p
        list.add(new MainModel(R.drawable.chknpie,  name: "Shepherd's Pie",  price: "80",  description: "Savoury pie filled with mouth-water

        MainAdapter adapter= new MainAdapter(list,  context: this);
        binding.recyclerView.setAdapter(adapter);//

        LinearLayoutManager layoutManager=new LinearLayoutManager( context: this);
        binding.recyclerView.setLayoutManager(layoutManager);

    }

    @Override
```

Figure 5.6 Code screenshot for MainActivity.java

5.7. DetailActivity.java

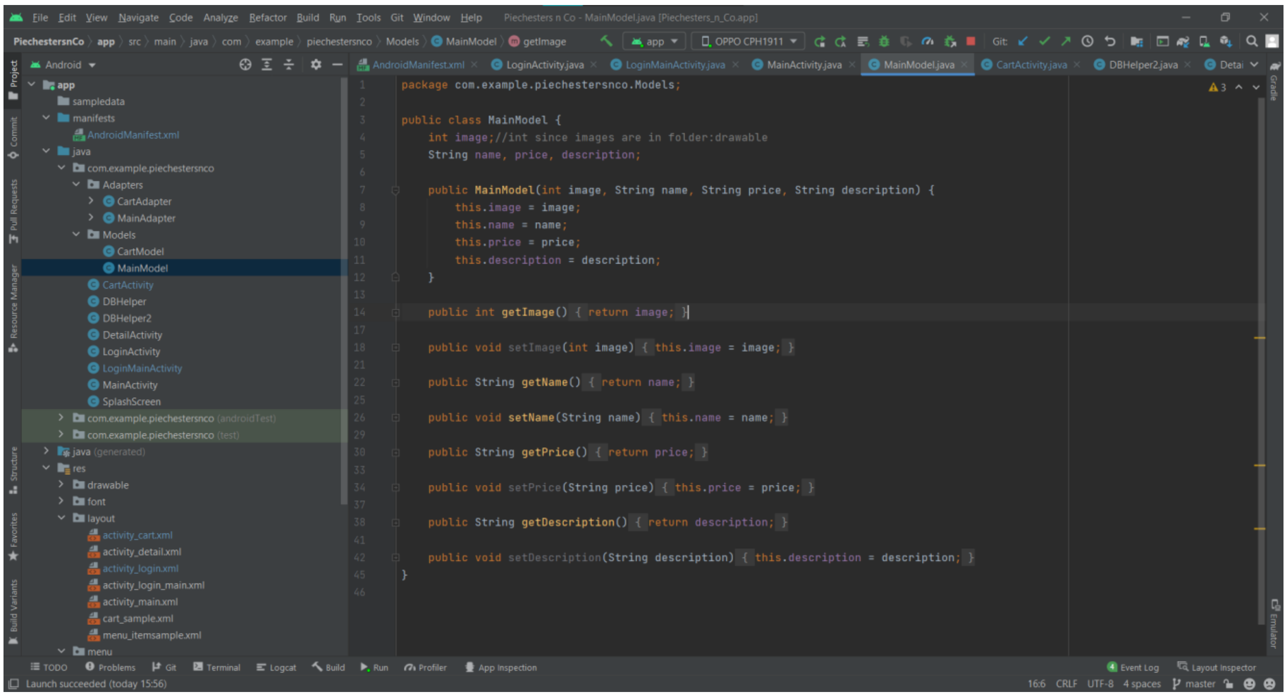


```java
package com.example.piechestersnco;

import ...

public class DetailActivity extends AppCompatActivity {
    ActivityDetailBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityDetailBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        final DBHelper helper = new DBHelper( context: this);

        if(getIntent().getIntExtra( name: "type",  defaultValue: 0)==1) {

            final int image = getIntent().getIntExtra( name: "image",  defaultValue: 0);
            final int price = Integer.parseInt(getIntent().getStringExtra( name: "price"));
            final String name = getIntent().getStringExtra( name: "name");
            final String description = getIntent().getStringExtra( name: "desc");

            binding.detailImage.setImageResource(image);
            binding.priceLbl.setText(String.format("%d", price));
            binding.nameLbl.setText(name);
            binding.detailDescription.setText(description);

            binding.insertBtn.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {//using the database opened in DBHelper

                    boolean isInserted = helper.insertOrder(
                            binding.nameBox.getText().toString(),
```

Figure 5.7 Code screenshot for DetailActivity.java

5.8. MainModel.java

MainModel.java is used to specify what all entries should be displayed in the Recycler view of the Menu on the LoginMainActivity activity.



```java
package com.example.piechestersnco.Models;

public class MainModel {
    int image;//int since images are in folder:drawable
    String name, price, description;

    public MainModel(int image, String name, String price, String description) {
        this.image = image;
        this.name = name;
        this.price = price;
        this.description = description;
    }

    public int getImage() { return image; }

    public void setImage(int image) { this.image = image; }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public String getPrice() { return price; }

    public void setPrice(String price) { this.price = price; }

    public String getDescription() { return description; }

    public void setDescription(String description) { this.description = description; }
}
```

Figure 5.8 Code screenshot for MainModel.java file

CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion

Our generation, and even the previous ones now rely mostly on phones to get through the day. Due to this, phones have become more of a personal assistant than a means to just communicate. Owing to the pandemic, people have become increasingly reliant upon their phones. People have become increasingly comfortable with ordering their bare necessities and cravings via mobile apps. Many now, even prefer it to the alternative. Keeping this in mind, we have developed a mobile app based on Java using Android Studio to bring to fruition our dream to bring pies into the mainstream choice of foods.

Future Enhancements

i. We can make this app more space and resource efficient so that this application consumes lesser RAM and ROM.

ii. We can even further make it private and secured by enhancing login features.

iii. We will be able to allow different users to write comments and feedbacks of their purchases which can be voted on by other customers.

REFERENCES

[1] IEEE Standard 830-1998: IEEE Recommended Practice for Software Requirements specifications.

[2] www.google.co.in

[3] Various videos from www.youtube.com

[4] https:// developer.android.com

[5] https://en.wikipedia.org/wiki

APPENDIX “A”
SCREENSHOTS

RESULTS

1. Main Activity (Splash Screen)

This is the screenshot of our mini-project's splash screen with logo which will be occurring for few seconds when the application is opened.

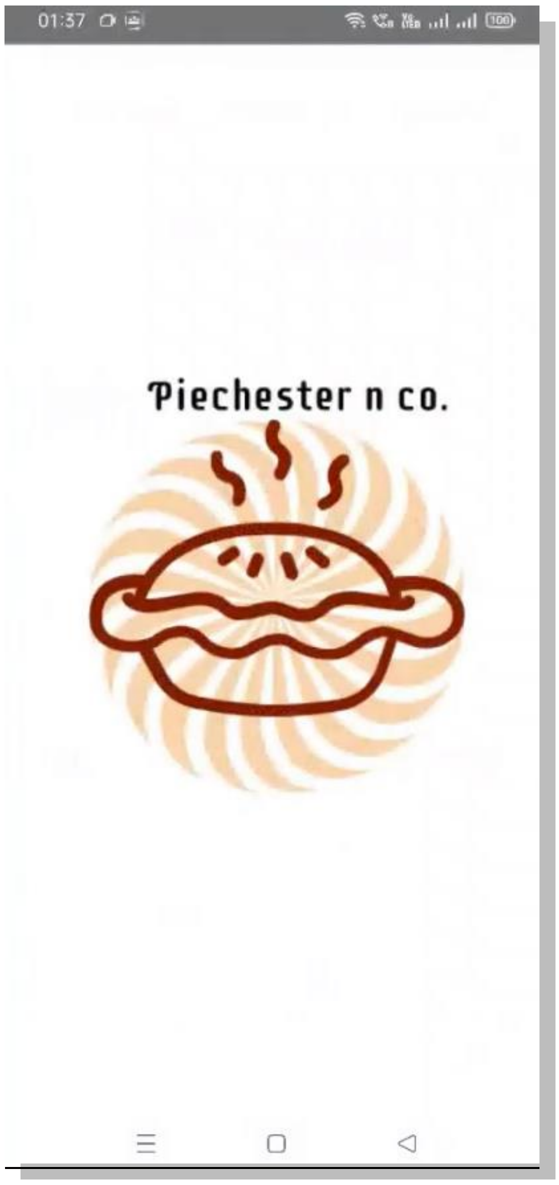


Figure A.1. Splash Screen of Piechester N Co application

The splash screen is then directed to the Registering Login activity where the user of the application can login to view the menu and save their purchases on the app.

2. Login Activity

This is the screenshot of our mini-project’s login screen where user can login by entering a registered username and password to use the app.

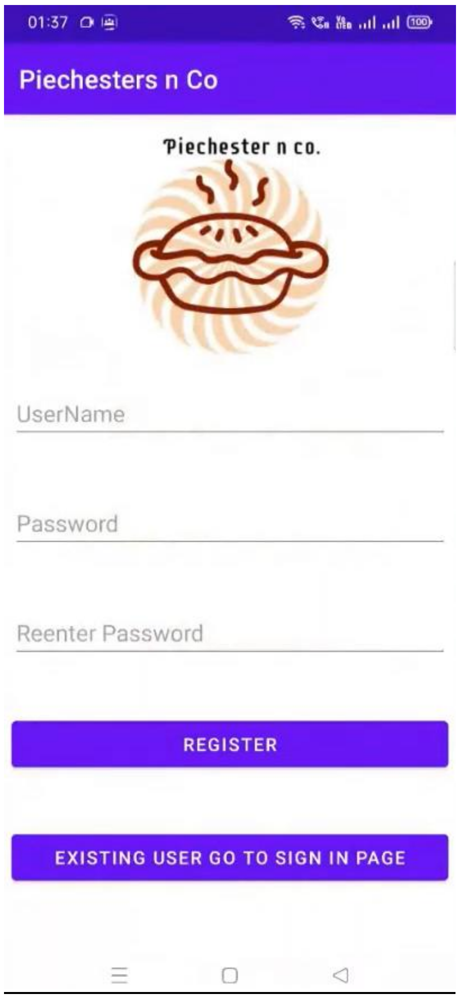
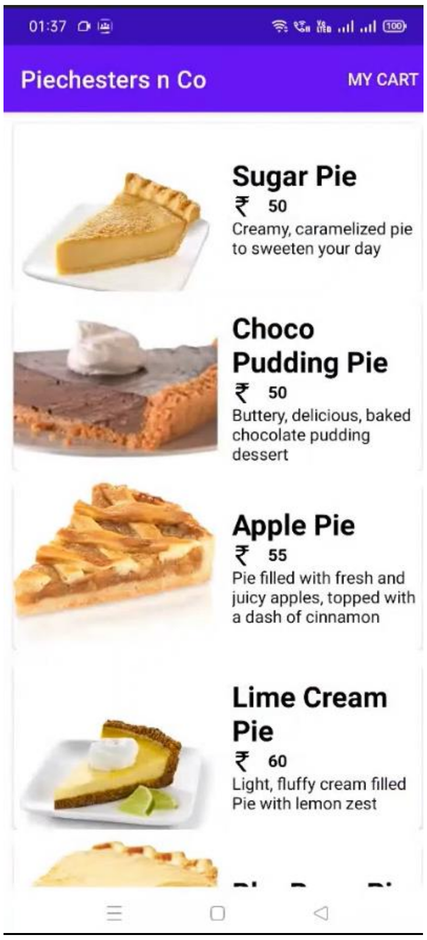


Figure A.2. Login Screen of Piechester N Co application

The login screen is connected to the Signup Activity where the user can get registered and the Get Started Activity where user can play a video to get themselves aware on how to use the app.

3. Menu Activity

This is the screenshot of our mini-project’s Menu screen where user can select among pies provided based on their interest.



Sugar Pie
₹ 50
Creamy, caramelized pie to sweeten your day

Choco Pudding Pie
₹ 50
Buttery, delicious, baked chocolate pudding dessert

Apple Pie
₹ 55
Pie filled with fresh and juicy apples, topped with a dash of cinnamon

Lime Cream Pie
₹ 60
Light, fluffy cream filled Pie with lemon zest

Figure A.3. Menu Screen of Piechester N Co application

The categories are displayed in the form of cards, on clicking the category the users are directed to the detailed activity.

4. Details Activity

This is the screenshot of our mini-project’s details screen where the user can see the details regarding that particular pie.

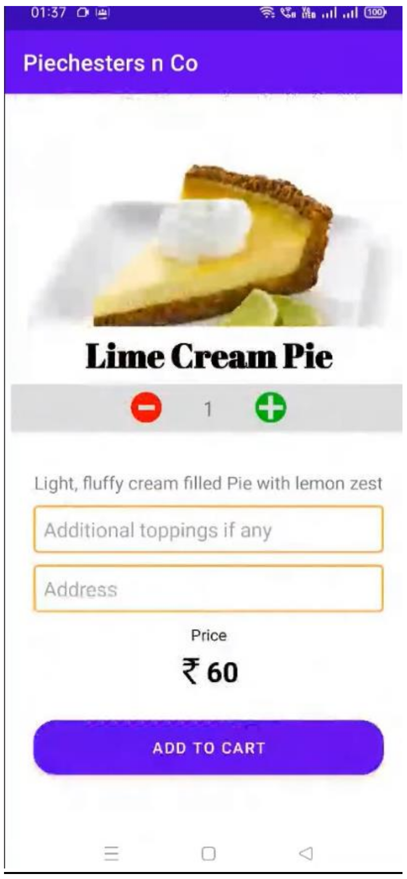
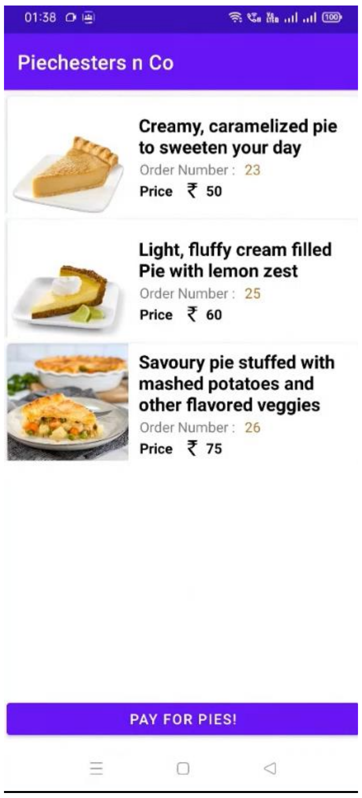

01:37
Piechesters n Co
Lime Cream Pie
1
Light, fluffy cream filled Pie with lemon zest
Additional toppings if any
Address
Price
₹ 60
ADD TO CART

Figure A.4. Details Screen of Piechester N Co application

5. Cart Activities

This is the screenshot of our mini-project's Cart displaying screen where user can view the items in the Cart.



Creamy, caramelized pie to sweeten your day
Order Number : 23
Price ₹ 50

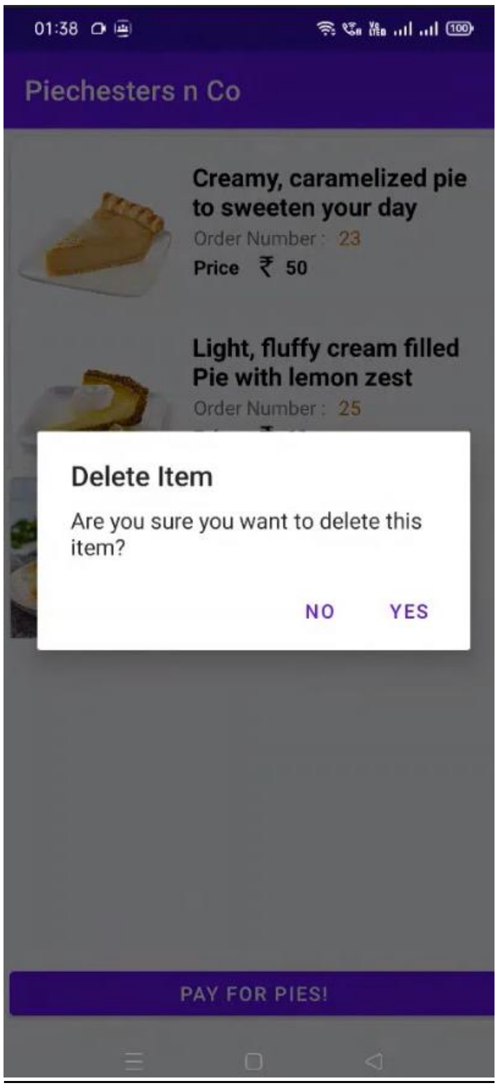
Light, fluffy cream filled Pie with lemon zest
Order Number : 25
Price ₹ 60

Savoury pie stuffed with mashed potatoes and other flavored veggies
Order Number : 26
Price ₹ 75

Figure A.5. Cart displaying Screen of Piechester N Co application

6. Delete Item

This is the screenshot of our mini-project's screen where user delete any item in the cart by pressing and holding for a while. After a pop-up asks for confirmation to delete a particular item, the item will be removed from cart.


01:38

Piechesters n Co

Creamy, caramelized pie to sweeten your day

Order Number : 23

Price ₹ 50

Light, fluffy cream filled Pie with lemon zest

Order Number : 25

Delete Item

Are you sure you want to delete this item?

NO YES

PAY FOR PIES!

Figure A.6. Delete item Screen of Piechester N Co application

THANKYOU