SMART BUS TICKET USING QR CODE

For the Degree of

Bachelor of Technology in

Computer Science and Engineering

By

Tanya Dixit (118)

Manthan Gadekar (120)

Rutuja Ganakwar (149)

Under the Guidance of

Prof. Kiran Chaudhary

Department of Computer Science and Engineering

Marathwada Institute of Technology, Aurangabad

Maharashtra State, India

(2020-2021)

Name & Sign of External:

A

Project Report

On

SMART BUS TICKET USING QR CODE

Submitted by

Tanya Dixit (118)

Manthan Gadekar (120)

Rutuja Ganakwar (149)

In partial fulfillment for the award of

Bachelor of Technology in

Computer Science and Engineering

Guided by

Prof. Kiran Chaudhary

Department of Computer Science and Engineering

Marathwada Institute of Technology, Aurangabad

Maharashtra State, India

2020-2021

Name & Sign of External:

CERTIFICATE

This is to certify that, the project entitled **“SMART BUS TICKET USING QR CODE”,** which has been submitted herewith in the partial fulfillment for the award of the **‘Bachelor of Technology’** in **‘Computer Science and Engineering’** of Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad (M.S.). This is the result of the original work and contribution by **Tanya Dixit, Manthan Gadekar** and **Rutuja Ganakwar under** my supervision and guidance.

Place: Aurangabad
Date:

Prof. Kiran Chaudhary	**Dr. Bhakti Ahirwadkar,**
Guide,	**Head,**
Department of computer and Engineering	Department of Computer Science and Engineering

Dr. Nilesh G. Patil
Principal
Marathwada Institute of Technology
Aurangabad (M.S.) - 431 010

Name & Sign of External:

Acknowledgement

I take this precious opportunity to express my gratitude towards **“SMART BUS TICKET USING QR CODE**” for granting permission for undergoing the training project. We are thankful to those people who helped me in shaping this project. Without their support, guidance and blessings this project would have not been completed. I have deep sense of gratitude in my heart for them.

We would give my sincere thanks to our Head of The Department Dr. Bhakti Ahirwadkar who have been and will be source of inspiration for us. We are thankful to our respected guide Prof. Kiran Chaudhary for his hearted support and affectionate encouragement without which we would have not been able to complete this project. We have great pleasure in submitting report for "Smart bus e-Ticket using QR code".

We are also thankful to Principal Dr. N. G. Patil who have always encouraged and inspired us because of whom we could learn to manage things in required time.

We are grateful to all staff members of Computer Science and Engineering department for their timely help.

THANK YOU

Name & Sign of External:

Abstract

Public transport is the cheapest and has therefore, always been popular with the masses. The advancement in transport system has been increasing in day-to-day life. The transport plays a vital role in individuals' life, in making it efficient we are introducing an android application. The android application has the bus ticket system using QR reader. The android mobile has a great part in human life, it helps the people to be stay connected with web.

In this project, we are proposing QR reader for bus ticketing system. The QR code (Quick Response code) becomes popular outside the automotive industry due to its fast readability and greater storage capacity compared to standard UPC barcodes. The proposed system provides web application as well as android application for the passengers to buy their tickets online. During the travel time, we can get the ticket by entering their location details and make payment. Message alert will be notified to the passenger. By this application, we can minimize the usage of paper (Tickets) and there will not be any problem in getting change.

Name & Sign of External:

\Table of Contents

Name & Sign of External:

Name & Sign of External:

CHAPTER 1

ANDRIOD APPLICATION DEVELOPMENT

1.1 Introduction about Andriod Programming Language

Before jumping over to actual programming lets first talk about a brief introduction to android programming. In this project I will give a short overview of android which include topics like history, features and different versions of android. Technically, you don't need any tools to complete this tutorial but you will certainly need them to develop Android applications. To develop Android applications (or any Java applications, for that matter), you need a development environment to write and build applications. Eclipse is a very popular development environment (IDE) for Java and the preferred IDE for Android development.

Android applications are developed using the Java language. As of now, that's really your only option for native applications. Java is a very popular programming language developed by Sun Microsystems (now owned by Oracle). Developed long after C and C++, Java incorporates many of the powerful features of those powerful languages while addressing some of their drawbacks. Still, programming languages are only as powerful as their libraries. These libraries exist to help developers build applications. Java is easy to learn for a variety of reasons. There's certainly no shortage of Java resources out there to help we learn the language. Java is one of the most widely discussed, taught, and used programming languages on the planet. It's used for many different types of programming projects, no matter their scale, from web applications to desktop applications to mobile applications.

1.2 Information about Andriod

Android is a Linux based and an open source operating system for devices such as smartphones, tablets, smartwatches, etc. It is the most popular mobile operating system in the world.

Android was developed by Open Handset Alliance(OHA) led by Google. Open Handset Alliance is a consortium of 84 companies such as Google, HTC, Dell, Motorola, Sony, Samsung, etc.

Although some other Langauges can be used but Java is the official language for Android development.

1.3 History of Android

Andy Rubin founded Android, Inc. in Palo Alto, California in October 2003.In July 2005, Google acquired Android Inc. and key employees including Andy Rubin, Rich Miner, Chris White and Nick Sears.

On November 5, 2007 Google revelead about development of Android OS.HTC Dream was the first android smartphone which was launched on October 22, 2008.

Android is the nick name of Andy Rubin which was given by his coworkers because of his love for robots.

Name & Sign of External:

1.4 Features of Android

Android has so many interesting features that makes it most popular mobile OS in the world. Few of them are given below.

User Interface – It provides very beautiful user interface.
Connectivity – Android supports various connectivity technologies like GSM/EDGE, WiFi, Bluetooth, CDMA, etc.
Storage – SQLite is a very light weight database which is used for storage purpose in android.
Messaging – It supports SMS and MMS.
Multitasking – Android supports multitasking feature which lets you to run multiple Applications simultaneously.

Name & Sign of External:

CHAPTER 2

JAVA PROGRAMMING LANGUAGE

2.1 Introduction about JAVA Programming Language

JAVA was developed by James Gosling at **Sun Microsystems** Inc in the year **1991**, later acquired by Oracle Corporation. It is a simple programming language. Java makes writing, compiling, and debugging programming easy. It helps to create reusable code and modular programs.

JAVA is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible. A general-purpose programming language made for developers to *write once run anywhere* that is compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code that can run on any Java Virtual Machine.

2.2 History

Java's history is very interesting. It is a programming language created in 1991. James Gosling, Mike Sheridan, and Patrick Naughton, a team of Sun engineers known as the **Green team** initiated the Java language in 1991. **Sun Microsystems** released its first public implementation in 1996 as **Java 1.0**. It provides no-cost -run-times on popular platforms. Java1.0 compiler was re-written in Java by Arthur Van Hoff to strictly comply with its specifications. With the arrival of Java 2, new versions had multiple configurations built for different types of platforms.

In 1997, Sun Microsystems approached the ISO standards body and later formalized Java, but it soon withdrew from the process. At one time, Sun made most of its Java implementations available without charge, despite their proprietary software status.

On November 13, 2006, Sun released much of its Java virtual machine as free, open-source software. On May 8, 2007, Sun finished the process, making all of its JVM's core code available under open-source distribution terms.

The principles for creating java were simple, robust, secured, high performance, portable, multi-threaded, interpreted, dynamic, etc. **James** Gosling in 1995 developed Java, who is known as the Father of Java. Currently, Java is used in mobile devices, internet programming, games, e-business, etc.

2.3 Java Terminology

One must be familiar with these common terms of Java:

2.3.1 Java Virtual Machine(JVM): This is generally referred to as JVM. There are three execution phases of a program. They are written, compile and run the program.
Writing a program is done by a java programmer like you and me.
The compilation is done by the **JAVAC** compiler which is a primary Java compiler included in the Java development kit (JDK). It takes Java program as input and generates bytecode as output. In the Running phase of a program, **JVM** executes the bytecode generated by the compiler. Now, we understood that the function of Java Virtual Machine is to execute the bytecode produced by the compiler. Every Operating System has a different JVM but the output they produce after the execution of bytecode is the same across all the operating systems. This is why Java is known as a **platform-independent language.**

2.3.2 Bytecode in the Development process: As discussed, the Javac compiler of JDK compiles the java source code into bytecode so that it can be executed by JVM. It is saved as **.class** file by the compiler. To view the bytecode, a disassembler like javap can be used.

Name & Sign of External:

2.3.3 Java Development Kit (JDK): While we were using the term JDK, when we learn about bytecode and JVM. So, as the name suggests, it is a complete Java development kit that includes everything including compiler, Java Runtime Environment (JRE), java debuggers, java docs, etc. For the program to execute in java, we need to install JDK on our computer in order to create, compile and run the java program.

2.3.4 Java Runtime Environment (JRE): JDK includes JRE. JRE installation on our computers allows the java program to run, however, we cannot compile it. JRE includes a browser, JVM, applet supports, and plugins. For running the java program, a computer needs JRE.

2.3.5 Garbage Collector: In Java, programmers can't delete the objects. To delete or recollect that memory JVM has a program called Garbage Collector. Garbage Collectors can recollect the of objects that are not referenced. So Java makes the life of a programmer easy by handling memory management. However, programmers should be careful about their code whether they are using objects that have been used for a long time. Because Garbage cannot recover the memory of objects being referenced.

2.3.6 ClassPath: The classpath is the file path where the java runtime and Java compiler look for **.class** files to load. By default, JDK provides many libraries. If you want to include external libraries they should be added to the classpath.

Name & Sign of External:

2.4 Main Features of Java

2.4.1 Platform Independent: Compiler converts source code to bytecode and then the JVM executes the bytecode generated by the compiler. This bytecode can run on any platform be it Windows, Linux, macOS which means if we compile a program on Windows, then we can run it on Linux and vice versa. Each operating system has a different JVM, but the output produced by all the OS is the same after the execution of bytecode. That is why we call java a platform-independent language.

2.4.2 Object-Oriented Programming Language: Organizing the program in the terms of collection of objects is a way of object-oriented programming, each of which represents an instance of the class.

The four main concepts of Object-Oriented programming are:

Abstraction
Encapsulation
Inheritance
Polymorphism

2.4.3 Simple: Java is one of the simple languages as it does not have complex features like pointers, operator overloading, multiple inheritances, Explicit memory allocation.

2.4.4 Robust: Java language is robust that means reliable. It is developed in such a way that it puts a lot of effort into checking errors as early as possible, that is why the java compiler is able to detect even those errors that are not easy to detect by another programming language. The main features of java that make it robust are garbage collection, Exception Handling, and memory allocation.

2.4.5 Secure: In java, we don't have pointers, and so we cannot access out-of-bound arrays i.e it shows **ArrayIndexOutOfBoundsException** if we try to do so. That's why several security flaws like stack corruption or buffer overflow is impossible to exploit in Java.

2.4.6 Distributed: We can create distributed applications using the java programming language. Remote Method Invocation and Enterprise Java Beans are used for creating distributed applications in java. The java programs can be easily distributed on one or more systems that are connected to each other through an internet connection.

2.5 Application of the Project

There are three application in this Project as below

1) User
2) Admin
3) Conductor

CHAPTER 3

USER

3.1 Process

1) Register or Login
2) Select source and destination
3) Check bus availability
4) A unique QR code is generated
5) User can travel in any bus on that route

Start screen will remain 2 to 3 seconds after the application is started. User is required to login first by entering their Email and Password. Firstly, user is required to sign up. Shows menu for users, here users required entering data of source location and destination location.

After the detailing and filling the required information the user details are now verified by the admin. The users' card and every information are verified by the admin in every sense and approved and rejected on the information given base. The verification process takes up to 24 hours. After successful verification user will get QR code. User can easily travel with the help of QR code only. Unique Id is generated for each and every user using the application.

Splash-screen Activity

```java
package com.alpha.epass_user.activity;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.Window;
import android.view.WindowManager;

import com.alpha.epass_user.R;

public class SplashscreenActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splashscreen);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                WindowManager.LayoutParams.FLAG_FULLSCREEN);

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent myIntent = new Intent(SplashscreenActivity.this,
LoginActivity.class);
                SplashscreenActivity.this.startActivity(myIntent);
                SplashscreenActivity.this.finish();
            }
        }, 1000);
    }
}
```

Name & Sign of External:

Login Activity

```java
package com.alpha.epass_user.activity;

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

import android.content.Intent;
import android.os.Bundle;
import android.text.method.PasswordTransformationMethod;
import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.alpha.epass_user.R;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class LoginActivity extends AppCompatActivity implements
View.OnClickListener {

    EditText email, password;
    Button loginButton;
    TextView createNewAccount, forgetPassword;
    TextView alertTV;

    AlertDialog.Builder builder;
    AlertDialog progressDialog;

    private FirebaseAuth mAuth;

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

        mAuth = FirebaseAuth.getInstance();

        progressDialog = getBuilder().create();
        progressDialog.setCancelable(false);


        loginButton = findViewById(R.id.loginButton);
```

Name & Sign of External:

```
        email = findViewById(R.id.username);
        password = findViewById(R.id.password);
        createNewAccount = findViewById(R.id.createNewaccount);
        forgetPassword = findViewById(R.id.forgotpassword);
        alertTV = findViewById(R.id.alertTV);

        loginButton.setOnClickListener(this);
        createNewAccount.setOnClickListener(this);
        forgetPassword.setOnClickListener(this);
    }

    @Override
    public void onClick(@NonNull View v) {
        alertTV.setVisibility(View.GONE);
        if (v.getId() == R.id.loginButton) {

            String mEmail = email.getText().toString();
            String mPassword = password.getText().toString();

            if (mEmail.equals("") || mEmail.isEmpty()) {
                alertTV.setVisibility(View.VISIBLE);
                alertTV.setText("please enter email");
                return;
            }
            if (mPassword.equals("") || mPassword.isEmpty()) {
                alertTV.setText("please enter password");
                alertTV.setVisibility(View.VISIBLE);
                return;
            }
            progressDialog.show();
            loginWithCredentials(mEmail, mPassword);
        }
        if (v.getId() == R.id.createNewaccount) {
            Intent myIntent = new Intent(LoginActivity.this,
CreateNewAccount.class);
            startActivity(myIntent);
        }
        if (v.getId() == R.id.forgotpassword) {
            PasswordResetDialog fileDialog = new
PasswordResetDialog(LoginActivity.this);
            fileDialog.show(getSupportFragmentManager(), "Reset Password");

        }

    }

    @Override
    public void onStart() {
        super.onStart();
        // Check if user is signed in (non-null) and update UI accordingly.
        FirebaseUser currentUser = mAuth.getCurrentUser();
        // updateUI(currentUser);
        if (currentUser != null) {
            updateUI();
        }
    }

    public void loginWithCredentials(@NonNull String email, @NonNull String
password) {
```

Name & Sign of External:

```
        mAuth.signInWithEmailAndPassword(email, password).addOnSuccessListener(new
OnSuccessListener<AuthResult>() {
            @Override
            public void onSuccess(AuthResult authResult) {
                progressDialog.dismiss();
                updateUI();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                progressDialog.dismiss();
                Toast.makeText(LoginActivity.this, "" + e.getMessage(),
Toast.LENGTH_LONG).show();
            }
        });
    }


    public AlertDialog.Builder getBuilder() {
        if (builder == null) {
            builder = new AlertDialog.Builder(LoginActivity.this);
            builder.setTitle("Checking details...");

            final ProgressBar progressBar = new ProgressBar(LoginActivity.this);
            LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
            progressBar.setLayoutParams(layoutParams);
            builder.setView(progressBar);
        }
        return builder;
    }

    public void updateUI() {
        Intent myIntent = new Intent(LoginActivity.this, RedirectActivity.class);
        startActivity(myIntent);
        LoginActivity.this.finish();
    }

}
```

Name & Sign of External:

Create New Account

```java
package com.alpha.epass_user.activity;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.alpha.epass_user.R;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class CreateNewAccount extends AppCompatActivity {


    AlertDialog.Builder builder;
    AlertDialog progressDialog;
    Button signup;
    EditText email, password;
    TextView alertTV;
    private FirebaseAuth mAuth;

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

        mAuth = FirebaseAuth.getInstance();
        progressDialog = getBuilder().create();
        progressDialog.setCancelable(false);

        signup = findViewById(R.id.signupButton);
        email = findViewById(R.id.username);
        password = findViewById(R.id.password);
        alertTV = findViewById(R.id.alertTV);
```

Name & Sign of External:

```
        signup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String mEmail = email.getText().toString();
                String mPassword = password.getText().toString();

                if (mEmail.equals("") || mEmail.isEmpty()) {
                    alertTV.setText("please enter email");
                    alertTV.setVisibility(View.VISIBLE);
                    return;
                }
                if (mPassword.equals("") || mPassword.isEmpty()) {
                    alertTV.setText("please enter password");
                    alertTV.setVisibility(View.VISIBLE);
                    return;
                }
                if (mPassword.length() < 9) {
                    alertTV.setText("password must be longer than 8 characters");
                    alertTV.setVisibility(View.VISIBLE);
                    return;
                }

                signupWithCredentials(mEmail, mPassword);

            }
        });


    }

    public void signupWithCredentials(@NonNull String email, @NonNull String
password) {
        progressDialog.show();
        mAuth.createUserWithEmailAndPassword(email,
password).addOnSuccessListener(new OnSuccessListener<AuthResult>() {
            @Override
            public void onSuccess(AuthResult authResult) {
                Intent intent = new Intent(CreateNewAccount.this,
StudentDetails.class);
                intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                startActivity(intent);
                progressDialog.dismiss();
                CreateNewAccount.this.finish();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                progressDialog.dismiss();
                Toast.makeText(CreateNewAccount.this, "" + e.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }

    public AlertDialog.Builder getBuilder() {
        if (builder == null) {
            builder = new AlertDialog.Builder(CreateNewAccount.this);
            builder.setTitle("Creating Account");
```

Name & Sign of External:

```
            final ProgressBar progressBar = new
ProgressBar(CreateNewAccount.this);
            LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
            progressBar.setLayoutParams(layoutParams);
            builder.setView(progressBar);
        }
        return builder;
    }


}
```

Name & Sign of External:

Profile Activity

```java
package com.alpha.epass_user.activity;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import com.alpha.epass_user.R;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

public class ProfileActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {

    private NavigationView mNavigationView;
    private DrawerLayout mDrawerLayout;
    private ActionBarDrawerToggle mToggle;
    private Toolbar mToolbar;
    private ChangeNavigationActivities mChangeActivityFromNavigation;
    private Context mContext;
    TextView name, email, gender, college, roll, accountID;
    ScrollView scrollView;
    AlertDialog.Builder builder;
    AlertDialog progressDialog;

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

        mContext = getApplicationContext();
        mDrawerLayout = findViewById(R.id.drawer_layout);
        mToolbar = findViewById(R.id.toolbar);

        progressDialog = getBuilder().create();
        progressDialog.setCancelable(false);

        mToggle = new ActionBarDrawerToggle(this, mDrawerLayout, mToolbar,
                R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        mDrawerLayout.addDrawerListener(mToggle);
        mToggle.syncState();
        mChangeActivityFromNavigation = new ChangeNavigationActivities();


        name = findViewById(R.id.name);
        college = findViewById(R.id.college);
        gender = findViewById(R.id.gender);
        email = findViewById(R.id.email);
        roll = findViewById(R.id.roll);
        accountID = findViewById(R.id.accountID);
        scrollView = findViewById(R.id.scrollViewLayout);

        scrollView.setVisibility(View.GONE);

        progressDialog.show();
        getDetails();

    }

    private void setNavigation() {
        mNavigationView = findViewById(R.id.nav_view);
        mNavigationView.getMenu().getItem(1).setChecked(true);
        mNavigationView.setNavigationItemSelectedListener(this);
        final TextView name =
mNavigationView.getHeaderView(0).findViewById(R.id.profileName);

        FirebaseAuth mAuth = FirebaseAuth.getInstance();
        try {
            name.setText("" + mAuth.getCurrentUser().getEmail());
        } catch (Exception e) {

        }
    }

    @Override
    protected void onStart() {
        super.onStart();
        setNavigation();
    }

    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        Log.d("PROA", "onNavigationItemSelected: " + id);
        if (id == R.id.nav_profile) {
```

Name & Sign of External:

```
            //     mDrawerLayout.closeDrawer(GravityCompat.START);
            //    mChangeActivityFromNavigation.startProfileActivity(mContext);
        } else if (id == R.id.nav_home) {
            mDrawerLayout.closeDrawer(GravityCompat.START);
            mChangeActivityFromNavigation.startHomeActivity(mContext);
        } else if (id == R.id.nav_new_application) {
            mDrawerLayout.closeDrawer(GravityCompat.START);
            mChangeActivityFromNavigation.startUserPassesActivity(mContext);
        } else if (id == R.id.nav_logout) {
            mDrawerLayout.closeDrawer(GravityCompat.START);
            AlertDialog.Builder builder = new
AlertDialog.Builder(ProfileActivity.this);
            builder.setMessage("Are you sure you want to logout
?").setPositiveButton("YES", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    FirebaseAuth.getInstance().signOut();
                    Intent intent = new Intent(ProfileActivity.this,
LoginActivity.class);
                    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                    startActivity(intent);
                }
            }).setNegativeButton("NO", null);
            AlertDialog alertDialog = builder.create();
            alertDialog.show();
        }
        return false;
    }


    public void getDetails() {
        FirebaseAuth mAuth = FirebaseAuth.getInstance();
        FirebaseFirestore db = FirebaseFirestore.getInstance();


db.collection(StudentDetails.STUDENT_COLLECTION).document(mAuth.getCurrentUser().g
etUid()).get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(@NonNull DocumentSnapshot documentSnapshot) {
                scrollView.setVisibility(View.VISIBLE);
                name.setText("" +
documentSnapshot.getString(StudentDetails.FIRST_NAME) + " " +
documentSnapshot.getString(StudentDetails.MIDDLE_NAME) + " " +
                        documentSnapshot.getString(StudentDetails.LAST_NAME));
                email.setText("" +
documentSnapshot.getString(StudentDetails.USER_EMAIL));
                college.setText("" +
documentSnapshot.getString(StudentDetails.COLLEGE_NAME));
                roll.setText("" +
documentSnapshot.getString(StudentDetails.COLLEGE_ID));
                accountID.setText("" + documentSnapshot.getId());

                gender.setText("" + documentSnapshot.getString("gender"));
                progressDialog.dismiss();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                progressDialog.dismiss();
                Toast.makeText(ProfileActivity.this, "" + e.getMessage(),
Toast.LENGTH_LONG).show();
            }
        });
    }

    public AlertDialog.Builder getBuilder() {
        if (builder == null) {
            builder = new AlertDialog.Builder(ProfileActivity.this);
            builder.setTitle("fetching details...");

            final ProgressBar progressBar = new ProgressBar(ProfileActivity.this);
            LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
            progressBar.setLayoutParams(layoutParams);
            builder.setView(progressBar);
        }
        return builder;
    }
}
```

Name & Sign of External:

Main Acitvity

```
package com.alpha.epass_user.activity;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.view.Menu;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.alpha.epass_user.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.FirebaseFirestoreException;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageException;
import com.google.firebase.storage.StorageReference;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AlertDialog;
import androidx.core.view.GravityCompat;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

public class MainActivity extends AppCompatActivity implements
```

```java
NavigationView.OnNavigationItemSelectedListener {

    private ProgressDialog mProgressDialog;
    private FirebaseFirestore db;
    private NavigationView mNavigationView;
    private DrawerLayout mDrawerLayout;
    private ActionBarDrawerToggle mToggle;
    private Toolbar mToolbar;
    private ChangeNavigationActivities mChangeActivityFromNavigation;
    private Context mContext;

    public static final String Remark1 = " Your account details is being verified
by the scrutiny officer \n ";
    public static final String Remark2 = " YOUR ACCOUNT IS VERIFIED! \n you can
now create e-Passes for your travel";
    public static final String Remark3 = " YOUR REQUEST IS REJECTED! \n Please
correct your details and resubmit your form \n Remark by verification officer:\n
";

    TextView accountid;
    ImageView imageView;
    ProgressBar progressBar;
    Button createPass, resubmit;
    AlertDialog.Builder builder;
    AlertDialog progressDialog;

    FirebaseAuth firebaseAuth;

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


        firebaseAuth = FirebaseAuth.getInstance();
        mContext = getApplicationContext();
        mDrawerLayout = findViewById(R.id.drawer_layout);
        mToolbar = findViewById(R.id.toolbar);

        mToggle = new ActionBarDrawerToggle(this, mDrawerLayout, mToolbar,
                R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        mDrawerLayout.addDrawerListener(mToggle);
        mToggle.syncState();
        mChangeActivityFromNavigation = new ChangeNavigationActivities();

        mProgressDialog = new ProgressDialog(this);
        progressDialog = getBuilder().create();
        progressDialog.setCancelable(false);


        accountid = findViewById(R.id.accountid);

        progressBar = findViewById(R.id.circularLoader);
        imageView = findViewById(R.id.qrcode);

        imageView.setVisibility(View.GONE);
        accountid.setText("ID: "+ firebaseAuth.getCurrentUser().getUid());
        fetchImages();


    }

    private void setNavigation() {
        mNavigationView = findViewById(R.id.nav_view);
        mNavigationView.getMenu().getItem(0).setChecked(true);
        mNavigationView.setNavigationItemSelectedListener(this);

        final TextView name =
mNavigationView.getHeaderView(0).findViewById(R.id.profileName);

        FirebaseAuth mAuth = FirebaseAuth.getInstance();
        try {
            name.setText("" + mAuth.getCurrentUser().getEmail());
        } catch (Exception e) {

        }
    }

    @Override
    protected void onStart() {
        super.onStart();
        setNavigation();
    }


    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        Log.d("PROA", "onNavigationItemSelected: " + id);
        if (id == R.id.nav_profile) {
            mDrawerLayout.closeDrawer(GravityCompat.START);
            mChangeActivityFromNavigation.startProfileActivity(mContext);
        } else if (id == R.id.nav_home) {
            mDrawerLayout.closeDrawer(GravityCompat.START);
            mChangeActivityFromNavigation.startHomeActivity(mContext);
        } else if (id == R.id.nav_new_application) {
            mDrawerLayout.closeDrawer(GravityCompat.START);
            mChangeActivityFromNavigation.startUserPassesActivity(mContext);
        } else if (id == R.id.nav_logout) {
            mDrawerLayout.closeDrawer(GravityCompat.START);
            AlertDialog.Builder builder = new
AlertDialog.Builder(MainActivity.this);
            builder.setMessage("Are you sure you want to logout
?").setPositiveButton("YES", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    FirebaseAuth.getInstance().signOut();
                    Intent intent = new Intent(MainActivity.this,
LoginActivity.class);
                    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                    startActivity(intent);
                }
            }).setNegativeButton("NO", null);
            AlertDialog alertDialog = builder.create();
            alertDialog.show();
        }
        return false;
    }




    private void fetchImages() {
        progressBar.setVisibility(View.VISIBLE);

        FirebaseStorage storage = FirebaseStorage.getInstance();


        StorageReference storageReference =
storage.getReference().child("qr_codes" + "/" +
firebaseAuth.getCurrentUser().getUid());
        storageReference.getBytes(1024 * 1024 * 5).addOnSuccessListener(new
OnSuccessListener<byte[]>() {
            @Override
            public void onSuccess(@NonNull byte[] bytes) {
                Bitmap bitmap = BitmapFactory.decodeByteArray(bytes, 0,
bytes.length);
                imageView.setImageBitmap(bitmap);
                progressBar.setVisibility(View.GONE);
                imageView.setVisibility(View.VISIBLE);
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(mContext, "" + e.getMessage(),
Toast.LENGTH_SHORT).show();
                progressBar.setVisibility(View.GONE);
                imageView.setVisibility(View.VISIBLE);
            }
        });

    }

    public AlertDialog.Builder getBuilder() {
        if (builder == null) {
            builder = new AlertDialog.Builder(MainActivity.this);
            builder.setTitle("Checking details...");

            final ProgressBar progressBar = new ProgressBar(MainActivity.this);
            LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
            progressBar.setLayoutParams(layoutParams);
            builder.setView(progressBar);
        }
        return builder;
    }
}
```

Name & Sign of External:

Student Activity

```java
package com.alpha.epass_user.activity;

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

import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;

import com.alpha.epass_user.R;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.io.ByteArrayOutputStream;
import java.util.HashMap;
import java.util.Map;

public class StudentDetails extends AppCompatActivity {

    public static final String FIRST_NAME = "firstName";
    public static final String MIDDLE_NAME = "middleName";
    public static final String LAST_NAME = "lastName";
    public static final String COLLEGE_NAME = "college";
    public static final String COLLEGE_ID = "rollNumber";
    public static final String ID_CARD_IMAGE = "idcard";
    public static final String SOURCE_LOCATION = "source";
    public static final String DESTINATION_LOCATION = "destination";
    public static final String USER_VERIFIED = "isVerified";
    public static final String USER_EMAIL = "email";
    public static final String STUDENT_COLLECTION = "student";

    EditText name, middlename, surname, college, rollnumber, source, destination;
    private ImageView idCardImage;
    private byte[] byteArray;
    @Nullable
    private Uri mImageHolder;
    Button selectImageBtn, submit;
    RadioGroup radioGroup;

    private FirebaseFirestore mFirebaseDb;
    private StorageReference mStorageReference;

    private ProgressDialog mProgressDialog;

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

        mProgressDialog = new ProgressDialog(this);

        name = findViewById(R.id.firstName);
        middlename = findViewById(R.id.middleName);
        surname = findViewById(R.id.lastName);
        college = findViewById(R.id.collegeName);
        rollnumber = findViewById(R.id.collegeID);
        source = findViewById(R.id.source);
        destination = findViewById(R.id.destination);

        radioGroup = findViewById(R.id.radioGroup);
        selectImageBtn = findViewById(R.id.selectImage);
        submit = findViewById(R.id.submitrequest);
        idCardImage = findViewById(R.id.idcardImage);

        mFirebaseDb = FirebaseFirestore.getInstance();
        mStorageReference = FirebaseStorage.getInstance().getReference();

        selectImageBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                selectImage();
            }
        });
        submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                checkAllFields();
            }
        });
    }

    public void checkAllFields() {

        String mName = name.getText().toString();
        String mMiddle = middlename.getText().toString();
        String mLast = surname.getText().toString();

        String mCollege = college.getText().toString();
        String mRoll = rollnumber.getText().toString();

        String mSource = source.getText().toString();
        String mDestination = destination.getText().toString();
```

Name & Sign of External:

```
        int radioButtonId = radioGroup.getCheckedRadioButtonId();
        RadioButton radioButton = findViewById(radioButtonId);
        String gender = radioButton.getText().toString();

        if (mName.equals("") || mName.isEmpty()) {
            name.setError("this field is missing");
            Toast.makeText(this, "Please enter all the details",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (mMiddle.equals("") || mMiddle.isEmpty()) {
            middlename.setError("this field is missing");
            Toast.makeText(this, "Please enter all the details",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (mLast.equals("") || mLast.isEmpty()) {
            surname.setError("this field is missing");
            Toast.makeText(this, "Please enter all the details",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (mCollege.equals("") || mCollege.isEmpty()) {
            college.setError("this field is missing");
            Toast.makeText(this, "Please enter all the details",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (mRoll.equals("") || mRoll.isEmpty()) {
            rollnumber.setError("this field is missing");
            Toast.makeText(this, "Please enter all the details",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (mSource.equals("") || mSource.isEmpty()) {
            source.setError("this field is missing");
            Toast.makeText(this, "Please enter all the details",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (mDestination.equals("") || mDestination.isEmpty()) {
            destination.setError("this field is missing");
            Toast.makeText(this, "Please enter all the details",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (mImageHolder == null) {
            Toast.makeText(this, "Please enter you ID PROOF IMAGE",
Toast.LENGTH_SHORT).show();
            return;
        }

        storeImage(mName, mLast, mMiddle, mCollege, mRoll, mSource, mDestination,
gender);
    }


    public void uploadDetails(String name, String surname, String middleName,
String college, String id, String source, String destination, String gender) {
```

Name & Sign of External:

```
        FirebaseAuth mAuth = FirebaseAuth.getInstance();
        String uuid = mAuth.getCurrentUser().getUid();
        String email = mAuth.getCurrentUser().getEmail();

        Map<String, Object> details = new HashMap<>();
        details.put(FIRST_NAME, name);
        details.put(LAST_NAME, surname);
        details.put(MIDDLE_NAME, middleName);
        details.put(COLLEGE_NAME, college);
        details.put(COLLEGE_ID, id);
        details.put(SOURCE_LOCATION, source);
        details.put(DESTINATION_LOCATION, destination);
        details.put(USER_VERIFIED, "0");
        details.put(USER_EMAIL, email);
        details.put("remark", "No Remarks");
        details.put("gender", gender);


mFirebaseDb.collection(STUDENT_COLLECTION).document(uuid).set(details).addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                hideProgressDialogWithTitle();
                updateUI();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                hideProgressDialogWithTitle();
                Toast.makeText(StudentDetails.this, "" + e.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }


    private void selectImage() {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Select Picture"), 1);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == 1 && resultCode == RESULT_OK && data != null &&
data.getData() != null) {
            mImageHolder = data.getData();
            idCardImage.setVisibility(View.VISIBLE);
            try {
                Bitmap bitmap =
MediaStore.Images.Media.getBitmap(getContentResolver(), mImageHolder);
                idCardImage.setImageBitmap(bitmap);
                ByteArrayOutputStream stream = new ByteArrayOutputStream();
                bitmap.compress(Bitmap.CompressFormat.JPEG, 100, stream);
//                bitmap.recycle();
```

```
                byteArray = stream.toByteArray();
            } catch (Exception e) {
                Toast.makeText(this, " " + e.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        }
    }

    private void storeImage(final String name, final String surname, final String
middleName, final String college, final String id, final String source, final
String destination, final String gender) {

        showProgressDialogWithTitle("Submitting your details...");
        FirebaseAuth mAuth = FirebaseAuth.getInstance();
        String uuid = mAuth.getCurrentUser().getUid();
        StorageReference reference = mStorageReference.child("IdCards" + "/" +
uuid);
        reference.putFile(mImageHolder).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                uploadDetails(name, surname, middleName, college, id, source,
destination, gender);
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                hideProgressDialogWithTitle();
                Toast.makeText(StudentDetails.this, "" + e.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });


    }

    public void updateUI() {
        Intent myIntent = new Intent(StudentDetails.this, RedirectActivity.class);
        startActivity(myIntent);
        StudentDetails.this.finish();
    }


    private void showProgressDialogWithTitle(String substring) {
        mProgressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        mProgressDialog.setCancelable(false);
        mProgressDialog.setMessage(substring);
        mProgressDialog.show();
    }

    private void hideProgressDialogWithTitle() {
        mProgressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        mProgressDialog.dismiss();
    }

}
```

Name & Sign of External:

User Passes

```
package com.alpha.epass_user.activity;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.cardview.widget.CardView;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.text.format.DateFormat;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.alpha.epass_user.R;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.auth.User;

import java.text.SimpleDateFormat;
import java.util.Date;

public class UserPasses extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {

    private NavigationView mNavigationView;
    private DrawerLayout mDrawerLayout;
    private ActionBarDrawerToggle mToggle;
    private Toolbar mToolbar;
    private ChangeNavigationActivities mChangeActivityFromNavigation;
    private Context mContext;

    AlertDialog.Builder builder;
    AlertDialog progressDialog;
```

Name & Sign of External:

```
    LinearLayout remarkLayout;

    public static final String REMARK = "Please create a new pass from Home
Section\n";

    TextView source, destination, validity, months, remark, status;
    Button createNewPass,createPass2;
    CardView cardView;

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

        mContext = getApplicationContext();
        mDrawerLayout = findViewById(R.id.drawer_layout);
        mToolbar = findViewById(R.id.toolbar);

        mToggle = new ActionBarDrawerToggle(this, mDrawerLayout, mToolbar,
                R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        mDrawerLayout.addDrawerListener(mToggle);
        mToggle.syncState();
        mChangeActivityFromNavigation = new ChangeNavigationActivities();

        progressDialog = getBuilder().create();
        progressDialog.setCancelable(false);

        status = findViewById(R.id.status);
        source = findViewById(R.id.source);
        destination = findViewById(R.id.destination);
        validity = findViewById(R.id.validity);
        months = findViewById(R.id.months);
        remark = findViewById(R.id.remark);
        remarkLayout = findViewById(R.id.remarkLayout);
        createNewPass = findViewById(R.id.createbtn);
        createPass2 = findViewById(R.id.createPass);

        cardView = findViewById(R.id.cardView);
        cardView.setVisibility(View.GONE);


        getData();

        createNewPass.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(UserPasses.this, CreateEPass.class);
                startActivity(intent);
            }
        });
        createPass2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(UserPasses.this, CreateEPass.class);
                startActivity(intent);
            }
        });
    }
    private void setNavigation() {
        mNavigationView = findViewById(R.id.nav_view);
        mNavigationView.getMenu().getItem(2).setChecked(true);
        mNavigationView.setNavigationItemSelectedListener(this);
        final TextView name =
mNavigationView.getHeaderView(0).findViewById(R.id.profileName);

        FirebaseAuth mAuth = FirebaseAuth.getInstance();
        try {
            name.setText("" + mAuth.getCurrentUser().getEmail());
        } catch (Exception e) {

        }
    }

    @Override
    protected void onStart() {
        super.onStart();
        setNavigation();
    }

    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        Log.d("PROA", "onNavigationItemSelected: " + id);
        if (id == R.id.nav_profile) {
            mDrawerLayout.closeDrawer(GravityCompat.START);
            mChangeActivityFromNavigation.startProfileActivity(mContext);
        } else if (id == R.id.nav_home) {
            mDrawerLayout.closeDrawer(GravityCompat.START);
            mChangeActivityFromNavigation.startHomeActivity(mContext);
        } else if (id == R.id.nav_new_application) {
            //  mDrawerLayout.closeDrawer(GravityCompat.START);
            //  mChangeActivityFromNavigation.startUserPassesActivity(mContext);
        } else if (id == R.id.nav_logout) {
            mDrawerLayout.closeDrawer(GravityCompat.START);
            AlertDialog.Builder builder = new
AlertDialog.Builder(UserPasses.this);
            builder.setMessage("Are you sure you want to logout
?").setPositiveButton("YES", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    FirebaseAuth.getInstance().signOut();
                    Intent intent = new Intent(UserPasses.this,
LoginActivity.class);
                    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                    startActivity(intent);
                }
            }).setNegativeButton("NO", null);
            AlertDialog alertDialog = builder.create();
            alertDialog.show();
        }
        return false;
    }

    public void getData() {
        progressDialog.show();
        FirebaseFirestore db = FirebaseFirestore.getInstance();
        FirebaseAuth mAuth = FirebaseAuth.getInstance();

db.collection("ePasses").document(mAuth.getCurrentUser().getUid()).get().addOnSucc
essListener(new OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(@NonNull DocumentSnapshot documentSnapshot) {

                try {
                    source.setText("" + documentSnapshot.getString("source"));
                    destination.setText("" +
documentSnapshot.getString("destination"));
                    months.setText("" + documentSnapshot.getString("months"));
                    validity.setText("" +
convertDate(documentSnapshot.getLong("expiryDate")));

                    int mstatus =
Integer.parseInt(documentSnapshot.getString("verificationStatus"));
                    if (mstatus == 1) {

                        status.setText("Approved");
                    } else if (mstatus == -1) {
                        status.setText("Rejected");
                        remarkLayout.setVisibility(View.VISIBLE);
                        remark.setText(REMARK +
documentSnapshot.getString("remark"));
                    } else {
                        status.setText("Pending");
                    }

                    cardView.setVisibility(View.VISIBLE);
                } catch (Exception e) {
                    Toast.makeText(mContext, "You dont have any ePass",
Toast.LENGTH_LONG).show();
                }
                progressDialog.dismiss();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(mContext, "" + e.getMessage(),
Toast.LENGTH_SHORT).show();
                progressDialog.dismiss();
            }
        });
    }

    @NonNull
    public String convertDate(long time) {
        String date = new SimpleDateFormat("dd/MM/yyyy").format(new Date(time));
        return date;
    }

    public AlertDialog.Builder getBuilder() {
        if (builder == null) {
            builder = new AlertDialog.Builder(UserPasses.this);
            builder.setTitle("fetching details...");

            final ProgressBar progressBar = new ProgressBar(UserPasses.this);
            LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
            progressBar.setLayoutParams(layoutParams);
            builder.setView(progressBar);
        }
        return builder;
    }

}
```

Name & Sign of External:

Create E-Passes

```
package com.alpha.epass_user.activity;

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

import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.os.StrictMode;
import android.provider.MediaStore;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.alpha.epass_user.R;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import org.apache.commons.net.ntp.NTPUDPClient;
import org.apache.commons.net.ntp.TimeInfo;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.util.HashMap;
import java.util.Map;

public class CreateEPass extends AppCompatActivity {


    EditText source, destination, months;
    ImageView imageView;

    private byte[] byteArray;
    @Nullable
    private Uri mImageHolder;
    Button selectImage, submit;

    private FirebaseFirestore mFirebaseDb;
    private StorageReference mStorageReference;

    AlertDialog.Builder builder;
    AlertDialog progressDialog;

    private static final Long THIRTY_DAY = 2592000000L;
    public static final String TIME_SERVER = "time-a.nist.gov";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_create_e_pass);
        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);


        progressDialog = getBuilder().create();
        progressDialog.setCancelable(false);

        source = findViewById(R.id.source_location);
        destination = findViewById(R.id.destination_location);
        months = findViewById(R.id.months);

        submit = findViewById(R.id.submit);
        selectImage = findViewById(R.id.selectpaymentproof);

        imageView = findViewById(R.id.paymentImage);

        mFirebaseDb = FirebaseFirestore.getInstance();
        mStorageReference = FirebaseStorage.getInstance().getReference();

        selectImage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                selectImage();
            }
        });

        submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                checkDetails();
            }
        });


    }

    public void checkDetails() {

        String sourceName = source.getText().toString();
        String destinationName = destination.getText().toString();
        String numberOfMonths = months.getText().toString();
        if (sourceName.equals("") || sourceName.isEmpty()) {
            source.setError("this field is missing");
            Toast.makeText(this, "Please enter all the details",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (destinationName.equals("") || destinationName.isEmpty()) {
            destination.setError("this field is missing");
            Toast.makeText(this, "Please enter all the details",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (numberOfMonths.equals("") || numberOfMonths.isEmpty()) {
            months.setError("this field is missing");
            Toast.makeText(this, "Please enter all the details",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (mImageHolder == null) {
            Toast.makeText(this, "Please select Payment Proof Image/Screenshot",
Toast.LENGTH_LONG).show();
            return;
        }

        long currentDate = getCurrentTime();
        long expiryDate = currentDate + Integer.parseInt(numberOfMonths) *
THIRTY_DAY;

        progressDialog.show();
        uploadImage(sourceName, destinationName, numberOfMonths, currentDate,
expiryDate);

    }


    private void selectImage() {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Select Picture"), 1);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == 1 && resultCode == RESULT_OK && data != null &&
data.getData() != null) {
            mImageHolder = data.getData();
            imageView.setVisibility(View.VISIBLE);
            try {
                Bitmap bitmap =
MediaStore.Images.Media.getBitmap(getContentResolver(), mImageHolder);
                imageView.setImageBitmap(bitmap);
                ByteArrayOutputStream stream = new ByteArrayOutputStream();
                bitmap.compress(Bitmap.CompressFormat.JPEG, 100, stream);
//                bitmap.recycle();
                //      byteArray = stream.toByteArray();
            } catch (Exception e) {
                Toast.makeText(this, " " + e.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        }
    }

    public void uploadImage(final String source, final String destination, final
String months, final long current, final long expiry) {
        FirebaseAuth mAuth = FirebaseAuth.getInstance();
        String uuid = mAuth.getCurrentUser().getUid();
        StorageReference reference = mStorageReference.child("PaymentProof" + "/"
+ uuid);
        reference.putFile(mImageHolder).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                storePassDetails(source, destination, months, current, expiry);
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                progressDialog.dismiss();
                Toast.makeText(CreateEPass.this, "" + e.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }

    public void storePassDetails(String source, String destination, String months,
long activationDate, long expiryDate) {

        FirebaseAuth mAuth = FirebaseAuth.getInstance();

        Map<String, Object> details = new HashMap<>();

        details.put("source", source);
        details.put("destination", destination);
        details.put("months", months);
        details.put("activationDate", activationDate);
        details.put("expiryDate", expiryDate);
        details.put("remark", "No Remarks");
        details.put("verificationStatus", "0");



mFirebaseDb.collection("ePasses").document(mAuth.getCurrentUser().getUid()).set(de
tails).addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Toast.makeText(CreateEPass.this, "Successfully Submitted!",
Toast.LENGTH_SHORT).show();
                progressDialog.dismiss();
                CreateEPass.this.finish();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                progressDialog.dismiss();
```

Name & Sign of External:

```
            }
        });

mFirebaseDb.collection(StudentDetails.STUDENT_COLLECTION).document(mAuth.getCurren
tUser().getUid()).collection("passesHistory").document().set(details).addOnSuccess
Listener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {

            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {

            }
        });
    }


    public AlertDialog.Builder getBuilder() {
        if (builder == null) {
            builder = new AlertDialog.Builder(CreateEPass.this);
            builder.setTitle("Submitting details...");

            final ProgressBar progressBar = new ProgressBar(CreateEPass.this);
            LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
            progressBar.setLayoutParams(layoutParams);
            builder.setView(progressBar);
        }
        return builder;
    }


    private Long getCurrentTime() {
        NTPUDPClient timeClient = new NTPUDPClient();
        InetAddress inetAddress = null;
        TimeInfo timeInfo = null;
        try {
            inetAddress = InetAddress.getByName(TIME_SERVER);
            timeInfo = timeClient.getTime(inetAddress);
        } catch (IOException e) {
            e.printStackTrace();
        }
        long returnTime = 0;
        if (timeInfo != null) {
            returnTime = timeInfo.getMessage().getReceiveTimeStamp().getTime();
        }
        return returnTime;
    }


}
```

CHAPTER 4

ADMIN

4.1 Process of Admin

1) Admin plays an important role in smooth functioning of the system
2) Managing data stored in database.
3) Adding new busses to the system

It starts with the logo of application and will stay for about 2 to 3 second on screen when the screen will start.

There are 2 functions which are account verification and E-pass applications in Admin.

Admin can view the passenger's details and applications given for the passes. Admin can approve and reject if the applications are not completely filled or the information are not complete by the user/passenger.

It also shows the list of users that has registered on the mobile application.

If the user forgot their password, they can contact the admin and the management, after that the admin will recover the data and ask the user to sign up again.

Splash-screen Activity

```
package com.alpha.epass_admin.activity;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.WindowManager;

import com.alpha.epass_admin.R;

public class SplashscreenActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splashscreen);
//        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                WindowManager.LayoutParams.FLAG_FULLSCREEN);

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent myIntent = new Intent(SplashscreenActivity.this,
LoginActivity.class);
                SplashscreenActivity.this.startActivity(myIntent);
                SplashscreenActivity.this.finish();
            }
        }, 1000);
    }
}
```

Name & Sign of External:

Login Activity

```java
package com.alpha.epass_admin.activity;

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

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.alpha.epass_admin.R;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class LoginActivity extends AppCompatActivity implements
View.OnClickListener{


    public static final String USER_NAME="busadmin@gmail.com";
    public static final String PASSWORD="busadmin123#";

    EditText email, password;
    Button loginButton;
    TextView alertTV;

    AlertDialog.Builder builder;
    AlertDialog progressDialog;

    private FirebaseAuth mAuth;

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

        mAuth = FirebaseAuth.getInstance();

        progressDialog = getBuilder().create();
        progressDialog.setCancelable(false);
        loginButton = (Button) findViewById(R.id.loginButton);
        email = (EditText) findViewById(R.id.username);
        password = (EditText) findViewById(R.id.password);
        alertTV = (TextView) findViewById(R.id.alertTV);

        email.setText(USER_NAME);
        password.setText(PASSWORD);

        loginButton.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        alertTV.setVisibility(View.GONE);

        if (v.getId() == R.id.loginButton) {

            String mEmail = email.getText().toString();
            String mPassword = password.getText().toString();



            if (mEmail.equals("") || mEmail.isEmpty()) {
                alertTV.setVisibility(View.VISIBLE);
                alertTV.setText("please enter email");
                return;
            }
            if (mPassword.equals("") || mPassword.isEmpty()) {
                alertTV.setText("please enter password");
                alertTV.setVisibility(View.VISIBLE);
                return;
            }
            progressDialog.show();
            loginWithCredentials(mEmail, mPassword);
        }
    }
    public void loginWithCredentials(String email, String password) {

      /*  if (email.equals(USER_NAME)&&password.equals(PASSWORD)){
            progressDialog.dismiss();
            updateUI();
        }else{
            progressDialog.dismiss();
            Toast.makeText(this, "Invalid username or password",
Toast.LENGTH_SHORT).show();
        }

       */
        mAuth.signInWithEmailAndPassword(email, password).addOnSuccessListener(new
OnSuccessListener<AuthResult>() {
            @Override
            public void onSuccess(AuthResult authResult) {
                progressDialog.dismiss();
                updateUI();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                progressDialog.dismiss();
                Toast.makeText(LoginActivity.this, "" + e.getMessage(),
Toast.LENGTH_LONG).show();
            }
        });


    }

    public AlertDialog.Builder getBuilder() {
        if (builder == null) {
            builder = new AlertDialog.Builder(LoginActivity.this);
            builder.setTitle("Checking details...");

            final ProgressBar progressBar = new ProgressBar(LoginActivity.this);
            LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
            progressBar.setLayoutParams(layoutParams);
            builder.setView(progressBar);
        }
        return builder;
    }

    public void updateUI() {
        Intent myIntent = new Intent(LoginActivity.this,
AccountVerification.class);
        startActivity(myIntent);
        LoginActivity.this.finish();
    }

    @Override
    protected void onStart() {
        super.onStart();

        FirebaseAuth auth= FirebaseAuth.getInstance();

        if (mAuth.getCurrentUser()!=null){
            updateUI();
        }
    }
}
```

Name & Sign of External:

Main Activity

```
package com.alpha.epass_admin.activity;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.widget.TextView;

import com.alpha.epass_admin.R;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.auth.FirebaseAuth;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;


public class MainActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {

    private NavigationView mNavigationView;
    private DrawerLayout mDrawerLayout;
    private ActionBarDrawerToggle mToggle;
    private Toolbar mToolbar;
    private ChangeNavigationActivities mChangeActivityFromNavigation;
    private Context mContext;


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

        mContext =getApplicationContext();
        mDrawerLayout = findViewById(R.id.drawer_layout);
        mToolbar = findViewById(R.id.toolbar);

        mToggle = new ActionBarDrawerToggle(this, mDrawerLayout, mToolbar,
                R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        mDrawerLayout.addDrawerListener(mToggle);
        mToggle.syncState();
        mChangeActivityFromNavigation = new ChangeNavigationActivities();

    }
```

Name & Sign of External:

```
    private void setNavigation() {
        mNavigationView = findViewById(R.id.nav_view);
        mNavigationView.getMenu().getItem(0).setChecked(true);
        mNavigationView.setNavigationItemSelectedListener(this);
    }

    @Override
    protected void onStart() {
        super.onStart();
        setNavigation();
    }




    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.nav_account_verification) {
            mDrawerLayout.closeDrawer(GravityCompat.START);

mChangeActivityFromNavigation.startAccountVerificationActivity(mContext);
        } else if (id==R.id.nav_new_application){
            mDrawerLayout.closeDrawer(GravityCompat.START);
            mChangeActivityFromNavigation.startEpassApplicationActivity(mContext);
        }else if (id==R.id.logout){

        }


        return false;
    }
}
```

Profile Activity

```
package com.alpha.epass_admin.activity;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import com.alpha.epass_admin.R;
import com.alpha.epass_admin.fragment.StudentData;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.google.zxing.BarcodeFormat;
import com.google.zxing.MultiFormatWriter;
import com.google.zxing.WriterException;
import com.google.zxing.common.BitMatrix;
import com.journeyapps.barcodescanner.BarcodeEncoder;

import java.io.ByteArrayOutputStream;
import java.util.HashMap;
import java.util.Map;

public class ProfileActivity extends AppCompatActivity  {
    private Context mContext;
    TextView name, email, gender, college, roll, accountID;
    ImageView imageView;
    ProgressBar progressBar;

    Button approve,reject;

    ScrollView scrollView;
    AlertDialog.Builder builder;
    AlertDialog progressDialog;

    StudentData studentData;
    String docID;
    String tag;

    LinearLayout linearLayout;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile2);

        mContext =ProfileActivity.this;


        progressDialog = getBuilder().create();
        progressDialog.setCancelable(false);

        name = findViewById(R.id.name);
        college = findViewById(R.id.college);
        gender = findViewById(R.id.gender);
        email = findViewById(R.id.email);
        roll = findViewById(R.id.roll);
        accountID = findViewById(R.id.accountID);
        imageView = findViewById(R.id.idproofimage);
        progressBar=findViewById(R.id.imageloader);


        approve=findViewById(R.id.approve1);
        reject=findViewById(R.id.reject);

        Intent intent=getIntent();
        studentData= (StudentData) intent.getSerializableExtra("studentdata");
        docID=intent.getStringExtra("docid");

        linearLayout=findViewById(R.id.approvalLayout);

        if(studentData.getIsVerified().equals("1")){
            linearLayout.setVisibility(View.GONE);
        }else{
            linearLayout.setVisibility(View.VISIBLE);
        }

        setDetails();



        approve.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AlertDialog.Builder builder=new
AlertDialog.Builder(ProfileActivity.this);
                builder.setTitle("Approve Request").setMessage("Are you sure you
want to approve this form?").setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        approveRequest();
                    }
                }).setNegativeButton("No", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {

                    }
                });
                builder.show();
            }
        });
        reject.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                EditText remark=(EditText) findViewById(R.id.remark);
                final String mRemark=remark.getText().toString();

                if (mRemark.equals("")||mRemark.isEmpty()){
                    remark.setError("Please enter remark");
                    Toast.makeText(mContext, "Please enter remark",
Toast.LENGTH_SHORT).show();
                    return;
                }

                AlertDialog.Builder builder=new
AlertDialog.Builder(ProfileActivity.this);
                builder.setTitle("Reject Request").setMessage("Are you sure you
want to reject this form?").setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        rejectRequest(mRemark);
                    }
                }).setNegativeButton("No", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {

                    }
                });
                builder.show();

            }
        });


      //  getDetails();

    }
```

Name & Sign of External:

```
    public void setDetails(){

        name.setText(studentData.getFirstName()+" "+studentData.getMiddleName()+"
"+studentData.getLastName());
        college.setText(studentData.getCollege());
        gender.setText(studentData.getGender());
        email.setText(studentData.getEmail());
        roll.setText(studentData.getRollNumber());
        accountID.setText(docID);

        loadImageFromDB(docID);

    }

/*

    public void getDetails() {
        FirebaseAuth mAuth = FirebaseAuth.getInstance();
        FirebaseFirestore db = FirebaseFirestore.getInstance();


db.collection(StudentDetails.STUDENT_COLLECTION).document(mAuth.getCurrentUser().g
etUid()).get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(@NonNull DocumentSnapshot documentSnapshot) {
                scrollView.setVisibility(View.VISIBLE);
                name.setText("" +
documentSnapshot.getString(StudentDetails.FIRST_NAME) + " " +
documentSnapshot.getString(StudentDetails.MIDDLE_NAME) + " " +
                        documentSnapshot.getString(StudentDetails.LAST_NAME));
                email.setText("" +
documentSnapshot.getString(StudentDetails.USER_EMAIL));
                college.setText("" +
documentSnapshot.getString(StudentDetails.COLLEGE_NAME));
                roll.setText("" +
documentSnapshot.getString(StudentDetails.COLLEGE_ID));
                accountID.setText("" + documentSnapshot.getId());

                gender.setText("" + documentSnapshot.getString("gender"));
                progressDialog.dismiss();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                progressDialog.dismiss();
                Toast.makeText(ProfileActivity.this, "" + e.getMessage(),
Toast.LENGTH_LONG).show();
            }
        });
    }


 */
    public AlertDialog.Builder getBuilder() {
        if (builder == null) {
            builder = new AlertDialog.Builder(ProfileActivity.this);
            builder.setTitle("Updating status...");

            final ProgressBar progressBar = new ProgressBar(ProfileActivity.this);
            LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
            progressBar.setLayoutParams(layoutParams);
            builder.setView(progressBar);
        }
        return builder;
    }


    public void loadImageFromDB(String imageName){

        FirebaseStorage storage = FirebaseStorage.getInstance();
        StorageReference storageReference =
storage.getReference().child("IdCards").child(imageName);
        storageReference.getBytes(1024 * 1024 * 5).addOnSuccessListener(new
OnSuccessListener<byte[]>() {
            @Override
            public void onSuccess(byte[] bytes) {
                Bitmap bitmap = BitmapFactory.decodeByteArray(bytes, 0,
bytes.length);
                imageView.setImageBitmap(bitmap);
                progressBar.setVisibility(View.GONE);
            }
        });

    }


    public void approveRequest(){
       uploadImage(docID);
    }

    public void rejectRequest(String remark){
        FirebaseFirestore db=FirebaseFirestore.getInstance();
        progressDialog.show();
        Map<String,Object> data=new HashMap<>();
        data.put("isVerified","-1");
        data.put("remark",remark);

db.collection("student").document(docID).update(data).addOnSuccessListener(new
OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Toast.makeText(mContext, "Successfully Rejected ! ",
Toast.LENGTH_SHORT).show();
                progressDialog.dismiss();
                ProfileActivity.this.finish();

            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(mContext, "Error: "+e.getMessage(),
Toast.LENGTH_SHORT).show();
                progressDialog.dismiss();
            }
        });
    }
```

Name & Sign of External:

```
    public Bitmap generateQR(String userUUID){

        Bitmap bitmap=null;

         // Whatever you need to encode in the QR code
        MultiFormatWriter multiFormatWriter = new MultiFormatWriter();
        try {
            BitMatrix bitMatrix = multiFormatWriter.encode(userUUID,
BarcodeFormat.QR_CODE,200,200);
            BarcodeEncoder barcodeEncoder = new BarcodeEncoder();
             bitmap = barcodeEncoder.createBitmap(bitMatrix);

          //  imageView.setImageBitmap(bitmap);
        } catch (WriterException e) {
            Log.d("QRCODE", "onFailure: bitmap: "+e.getMessage());
            e.printStackTrace();
        }
        return bitmap;
    }


    public void uploadImage(String docID){
        progressDialog.show();
        StorageReference
storageReference=FirebaseStorage.getInstance().getReference();
        StorageReference reference = storageReference.child("qr_codes" + "/" +
docID );
        Bitmap bitmap=generateQR(docID);
        if (bitmap==null){
            Log.d("QRCODE", "onFailure: Bitpmap Null");
            Toast.makeText(mContext, "Some error occured please try again",
Toast.LENGTH_SHORT).show();
            progressDialog.dismiss();

        }else {

            Log.d("QRCODE", "onSuccess: Bitpmap is NOT Null ,now uploading the
image!");

            ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream();
            bitmap.compress(Bitmap.CompressFormat.JPEG, 100,
byteArrayOutputStream);
            byte[] data = byteArrayOutputStream.toByteArray();

            reference.putBytes(data).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                    Log.d("QRCODE", "QR is uploaded Successfully!");
                    storeDataToFirestore();
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    progressDialog.dismiss();
                    Toast.makeText(mContext, "Firebase Error: "+e.getMessage(),
Toast.LENGTH_SHORT).show();
```

Name & Sign of External:

```
                        Log.d("QRCODE", "onFailure: " + e.getMessage());
                    }
                });
            }

        }

        public void storeDataToFirestore(){
            FirebaseFirestore db=FirebaseFirestore.getInstance();

    db.collection("student").document(docID).update("isVerified","1").addOnSuccessList
    ener(new OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void aVoid) {
                    Toast.makeText(mContext, "Successfully Approved ! ",
    Toast.LENGTH_SHORT).show();
                    progressDialog.dismiss();
                    ProfileActivity.this.finish();

                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    Toast.makeText(mContext, "Error: "+e.getMessage(),
    Toast.LENGTH_SHORT).show();
                    progressDialog.dismiss();
                }
            });
        }
    }
```

Name & Sign of External:

E-Pass Application

```java
package com.alpha.epass_admin.activity;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.viewpager.widget.ViewPager;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;

import com.alpha.epass_admin.R;
import com.alpha.epass_admin.fragment.EpassApproved;
import com.alpha.epass_admin.fragment.EpassPending;
import com.google.android.material.navigation.NavigationView;
import com.google.android.material.tabs.TabLayout;
import com.google.firebase.auth.FirebaseAuth;

public class EpassApplications extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener{

    private NavigationView mNavigationView;
    private DrawerLayout mDrawerLayout;
    private ActionBarDrawerToggle mToggle;
    private Toolbar mToolbar;
    private ChangeNavigationActivities mChangeActivityFromNavigation;
    private Context mContext;

    private TabLayout mTabLayout;
    private ViewPager mViewPager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_epass_applications);
        mContext =getApplicationContext();
        mDrawerLayout = findViewById(R.id.drawer_layout);
        mToolbar = findViewById(R.id.toolbar);

        mToggle = new ActionBarDrawerToggle(this, mDrawerLayout, mToolbar,
                R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        mDrawerLayout.addDrawerListener(mToggle);
        mToggle.syncState();
```

Name & Sign of External:

```
        mChangeActivityFromNavigation = new ChangeNavigationActivities();

        mTabLayout = findViewById(R.id.tabLayout);
        mViewPager = findViewById(R.id.myViewPager);
        setupViewPager(mViewPager);
        mTabLayout.setupWithViewPager(mViewPager);

    }

    private void setNavigation() {
        mNavigationView = findViewById(R.id.nav_view);
        mNavigationView.getMenu().getItem(1).setChecked(true);
        mNavigationView.setNavigationItemSelectedListener(this);
    }

    @Override
    protected void onStart() {
        super.onStart();
        setNavigation();
    }

    private void setupViewPager(ViewPager mViewPager) {
        ViewPagerAdapter viewPagerAdapter = new
ViewPagerAdapter(getSupportFragmentManager());
        viewPagerAdapter.addFragment(new EpassPending(), "Pending");
        viewPagerAdapter.addFragment(new EpassApproved(), "Approved");
        mViewPager.setAdapter(viewPagerAdapter);
    }



    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.nav_account_verification) {
            mDrawerLayout.closeDrawer(GravityCompat.START);

mChangeActivityFromNavigation.startAccountVerificationActivity(mContext);
        }else if (id==R.id.nav_new_application){
            mDrawerLayout.closeDrawer(GravityCompat.START);
            mChangeActivityFromNavigation.startEpassApplicationActivity(mContext);
        }else if (id==R.id.logout){

            mDrawerLayout.closeDrawer(GravityCompat.START);
            AlertDialog.Builder builder = new
AlertDialog.Builder(EpassApplications.this);
            builder.setMessage("Are you sure you want to logout
?").setPositiveButton("YES", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    FirebaseAuth.getInstance().signOut();
                    Intent intent = new Intent(EpassApplications.this,
LoginActivity.class);
                    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                    startActivity(intent);
                }
            }).setNegativeButton("NO", null);
            AlertDialog alertDialog = builder.create();
            alertDialog.show();
        }


        return false;
```

Name & Sign of External:

Account Verification

```java
package com.alpha.epass_admin.activity;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.viewpager.widget.ViewPager;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;

import com.alpha.epass_admin.R;
import com.alpha.epass_admin.fragment.ApplicationApproved;
import com.alpha.epass_admin.fragment.ApplicationPending;
import com.google.android.material.navigation.NavigationView;
import com.google.android.material.tabs.TabLayout;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class AccountVerification extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {

    private NavigationView mNavigationView;
    private DrawerLayout mDrawerLayout;
    private ActionBarDrawerToggle mToggle;
    private Toolbar mToolbar;
    private ChangeNavigationActivities mChangeActivityFromNavigation;
    private Context mContext;

    private TabLayout mTabLayout;
    private ViewPager mViewPager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_account_verification);
        mContext =getApplicationContext();
        mDrawerLayout = findViewById(R.id.drawer_layout);
        mToolbar = findViewById(R.id.toolbar);

        mTabLayout = findViewById(R.id.tabLayout);
        mViewPager = findViewById(R.id.myViewPager);
        setupViewPager(mViewPager);
        mTabLayout.setupWithViewPager(mViewPager);
        mToggle = new ActionBarDrawerToggle(this, mDrawerLayout, mToolbar,
                R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        mDrawerLayout.addDrawerListener(mToggle);
        mToggle.syncState();
        mChangeActivityFromNavigation = new ChangeNavigationActivities();

    }

    private void setNavigation() {
        mNavigationView = findViewById(R.id.nav_view);
        mNavigationView.getMenu().getItem(0).setChecked(true);
        mNavigationView.setNavigationItemSelectedListener(this);
    }

    @Override
    protected void onStart() {
        super.onStart();
        setNavigation();
    }


    private void setupViewPager(ViewPager mViewPager) {
        ViewPagerAdapter viewPagerAdapter = new
ViewPagerAdapter(getSupportFragmentManager());
        viewPagerAdapter.addFragment(new
ApplicationPending(AccountVerification.this), "Pending");
        viewPagerAdapter.addFragment(new
ApplicationApproved(AccountVerification.this), "Approved");
        mViewPager.setAdapter(viewPagerAdapter);
    }


    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.nav_account_verification) {
            mDrawerLayout.closeDrawer(GravityCompat.START);

mChangeActivityFromNavigation.startAccountVerificationActivity(mContext);
        } else if (id==R.id.nav_new_application){
            mDrawerLayout.closeDrawer(GravityCompat.START);

mChangeActivityFromNavigation.startEpassApplicationActivity(mContext);
        }else if (id==R.id.logout){

            mDrawerLayout.closeDrawer(GravityCompat.START);
            AlertDialog.Builder builder = new
AlertDialog.Builder(AccountVerification.this);
            builder.setMessage("Are you sure you want to logout
?").setPositiveButton("YES", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    FirebaseAuth.getInstance().signOut();
                    Intent intent = new Intent(AccountVerification.this,
LoginActivity.class);
                    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                    startActivity(intent);
                }
            }).setNegativeButton("NO", null);
            AlertDialog alertDialog = builder.create();
            alertDialog.show();
        }


        return false;
    }
}
```

Name & Sign of External:

Verify Account Details

```java
package com.alpha.epass_admin.activity;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import com.alpha.epass_admin.R;

public class VerifyAccountDetails extends AppCompatActivity {



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

Name & Sign of External:

Verify E-Passes

```
package com.alpha.epass_admin.activity;

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

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.alpha.epass_admin.R;
import com.alpha.epass_admin.fragment.EpassModel;
import com.alpha.epass_admin.fragment.StudentData;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;

public class VerifyEpass extends AppCompatActivity {


    TextView source,destination,validity,months,status;
    EditText remark;
    ImageView paymentProof;
    Button approve,reject,viewProfile;

    Context mContext;

    AlertDialog.Builder builder;
    AlertDialog progressDialog;
```

```java
ProgressBar progressBar;

EpassModel epassModel;
String docID;

StudentData data;



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

    mContext=VerifyEpass.this;
    progressDialog = getBuilder().create();
    progressDialog.setCancelable(false);

    source=findViewById(R.id.source);
    destination=findViewById(R.id.destination);
    validity=findViewById(R.id.validity);
    months=findViewById(R.id.months);
    status=findViewById(R.id.status);

    remark=findViewById(R.id.remark);
    paymentProof=findViewById(R.id.idproofimage);

    approve=findViewById(R.id.approve);
    progressBar=findViewById(R.id.imageloader);

    reject=findViewById(R.id.reject);
    viewProfile=findViewById(R.id.viewProfile);

    Intent intent=getIntent();
    epassModel= (EpassModel) intent.getSerializableExtra("epass");
    docID=intent.getStringExtra("docid");

    if (epassModel.getVerificationStatus().equals("1")){
        LinearLayout linearLayout=findViewById(R.id.remarkLayout);
        linearLayout.setVisibility(View.GONE);
    }

    viewProfile.setClickable(false);
    setdetails();
    fetchData();

    viewProfile.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent intent=new Intent(VerifyEpass.this, ProfileActivity.class);
            intent.putExtra("studentdata",data);
            intent.putExtra("docid",docID);
            startActivity(intent);
        }
    });
```

```
        approve.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AlertDialog.Builder builder=new
AlertDialog.Builder(VerifyEpass.this);
                builder.setTitle("Approve ePass").setMessage("Are you sure you
want to approve this ePass?").setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        approveRequest();
                    }
                }).setNegativeButton("No", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {

                    }
                });
                builder.show();

            }
        });
        reject.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                EditText remark=(EditText) findViewById(R.id.remark);
                final String mRemark=remark.getText().toString();

                if (mRemark.equals("")||mRemark.isEmpty()){
                    remark.setError("Please enter remark");
                    Toast.makeText(VerifyEpass.this, "Please enter remark",
Toast.LENGTH_SHORT).show();
                    return;
                }

                AlertDialog.Builder builder=new
AlertDialog.Builder(VerifyEpass.this);
                builder.setTitle("Reject ePass").setMessage("Are you sure you want
to reject this ePass?").setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        rejectRequest(mRemark);
                    }
                }).setNegativeButton("No", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {

                    }
                });

                builder.show();

            }
        });
    }


    public void approveRequest(){
        FirebaseFirestore db=FirebaseFirestore.getInstance();
        progressDialog.show();

db.collection("ePasses").document(docID).update("verificationStatus","1").addOnSuc
cessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Toast.makeText(mContext, "Successfully Approved ! ",
Toast.LENGTH_SHORT).show();
                progressDialog.dismiss();
                VerifyEpass.this.finish();

            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(mContext, "Error: "+e.getMessage(),
Toast.LENGTH_SHORT).show();
                progressDialog.dismiss();
            }
        });
    }

    public void rejectRequest(String remark){
        FirebaseFirestore db=FirebaseFirestore.getInstance();
        progressDialog.show();
        Map<String,Object> data=new HashMap<>();
        data.put("verificationStatus","-1");
        data.put("remark",remark);

db.collection("ePasses").document(docID).update(data).addOnSuccessListener(new
OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Toast.makeText(mContext, "Successfully Rejected ! ",
Toast.LENGTH_SHORT).show();
                progressDialog.dismiss();
                VerifyEpass.this.finish();

            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(mContext, "Error: "+e.getMessage(),
Toast.LENGTH_SHORT).show();
                progressDialog.dismiss();
            }
        });
    }
    public AlertDialog.Builder getBuilder() {
        if (builder == null) {
            builder = new AlertDialog.Builder(VerifyEpass.this);
            builder.setTitle("Updating status...");

            final ProgressBar progressBar = new ProgressBar(VerifyEpass.this);
            LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
            progressBar.setLayoutParams(layoutParams);
```

Name & Sign of External:

```
            builder.setView(progressBar);
        }
        return builder;
    }


    public void loadImageFromDB(String imageName){

        FirebaseStorage storage = FirebaseStorage.getInstance();
        StorageReference storageReference =
storage.getReference().child("PaymentProof").child(imageName);
        storageReference.getBytes(1024 * 1024 * 5).addOnSuccessListener(new
OnSuccessListener<byte[]>() {
            @Override
            public void onSuccess(byte[] bytes) {
                Bitmap bitmap = BitmapFactory.decodeByteArray(bytes, 0,
bytes.length);
                paymentProof.setImageBitmap(bitmap);
                progressBar.setVisibility(View.GONE);
            }
        });

    }

    public void fetchData(){
        FirebaseFirestore db=FirebaseFirestore.getInstance();
        db.collection("student").document(docID).get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                data=documentSnapshot.toObject(StudentData.class);
                viewProfile.setClickable(true);
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {

            }
        });
    }

    public String convertDate(long time) {
        String date = new SimpleDateFormat("dd/MM/yyyy").format(new Date(time));
        return date;
    }


    public void setdetails(){
        source.setText(""+epassModel.getSource());
        destination.setText(""+epassModel.getDestination());
        months.setText(""+epassModel.getMonths());
        validity.setText(""+convertDate(epassModel.getExpiryDate()));

        String mStatus=epassModel.getVerificationStatus();
        if (mStatus.equals("1")){
            status.setText("Approved");
        }else{
            status.setText("Pending");
        }
```

Name & Sign of External:

```
        loadImageFromDB(docID);
}}
```

Name & Sign of External:

Account Verification

```
package com.alpha.epass_admin.activity;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.viewpager.widget.ViewPager;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;

import com.alpha.epass_admin.R;
import com.alpha.epass_admin.fragment.ApplicationApproved;
import com.alpha.epass_admin.fragment.ApplicationPending;
import com.google.android.material.navigation.NavigationView;
import com.google.android.material.tabs.TabLayout;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class AccountVerification extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {

    private NavigationView mNavigationView;
    private DrawerLayout mDrawerLayout;
    private ActionBarDrawerToggle mToggle;
    private Toolbar mToolbar;
    private ChangeNavigationActivities mChangeActivityFromNavigation;
    private Context mContext;

    private TabLayout mTabLayout;
    private ViewPager mViewPager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_account_verification);
        mContext =getApplicationContext();
        mDrawerLayout = findViewById(R.id.drawer_layout);
        mToolbar = findViewById(R.id.toolbar);

        mTabLayout = findViewById(R.id.tabLayout);
        mViewPager = findViewById(R.id.myViewPager);
        setupViewPager(mViewPager);
        mTabLayout.setupWithViewPager(mViewPager);

        mToggle = new ActionBarDrawerToggle(this, mDrawerLayout, mToolbar,
                R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
```

Name & Sign of External:

```
        mDrawerLayout.addDrawerListener(mToggle);
        mToggle.syncState();
        mChangeActivityFromNavigation = new ChangeNavigationActivities();

    }

    private void setNavigation() {
        mNavigationView = findViewById(R.id.nav_view);
        mNavigationView.getMenu().getItem(0).setChecked(true);
        mNavigationView.setNavigationItemSelectedListener(this);
    }

    @Override
    protected void onStart() {
        super.onStart();
        setNavigation();
    }


    private void setupViewPager(ViewPager mViewPager) {
        ViewPagerAdapter viewPagerAdapter = new
ViewPagerAdapter(getSupportFragmentManager());
        viewPagerAdapter.addFragment(new
ApplicationPending(AccountVerification.this), "Pending");
        viewPagerAdapter.addFragment(new
ApplicationApproved(AccountVerification.this), "Approved");
        mViewPager.setAdapter(viewPagerAdapter);
    }


    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.nav_account_verification) {
            mDrawerLayout.closeDrawer(GravityCompat.START);

mChangeActivityFromNavigation.startAccountVerificationActivity(mContext);
        } else if (id==R.id.nav_new_application){
            mDrawerLayout.closeDrawer(GravityCompat.START);

mChangeActivityFromNavigation.startEpassApplicationActivity(mContext);
        }else if (id==R.id.logout){

            mDrawerLayout.closeDrawer(GravityCompat.START);
            AlertDialog.Builder builder = new
AlertDialog.Builder(AccountVerification.this);
            builder.setMessage("Are you sure you want to logout
?").setPositiveButton("YES", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    FirebaseAuth.getInstance().signOut();
                    Intent intent = new Intent(AccountVerification.this,
LoginActivity.class);
                    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                    startActivity(intent);
                }
            }).setNegativeButton("NO", null);
            AlertDialog alertDialog = builder.create();
```

Name & Sign of External:

```
            alertDialog.show();
        }


        return false;
    }
}
```

Name & Sign of External:

CHAPTER 5

CONDUCTOR

5.1 Working Process of the Conductor application

1) Scan Users QR code.
2) Validate Passenger details.
3) Automatically update database

The conductor will scan the QR-Code which are given to the verified user and approve the user to travel in the bus.

After the scanning of the QR-Code the conductor will be notified if the user/passenger is valid user or not.

- QR code contains:
  - User Details
  - Source & Destination
  - Date and Time of booking
  - Total Fare.

Name & Sign of External:

Scanned Barcode Activity

```
package com.epass_conductor;

import android.Manifest;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.util.SparseArray;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;


import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.android.gms.vision.CameraSource;
import com.google.android.gms.vision.Detector;
import com.google.android.gms.vision.barcode.Barcode;
import com.google.android.gms.vision.barcode.BarcodeDetector;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.io.IOException;

public class ScannedBarcodeActivity extends AppCompatActivity {


    private SurfaceView surfaceView;
    private BarcodeDetector barcodeDetector;
    private CameraSource cameraSource;
    private static final int REQUEST_CAMERA_PERMISSION = 201;
    private String intentData = "";
    private boolean isEmail = false;
    private ProgressDialog mProgressDialog;
    AlertDialog progressDialog;
    AlertDialog.Builder builder;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scan_barcode);
        mProgressDialog = new ProgressDialog(this);
        progressDialog = getBuilder().create();
        progressDialog.setCancelable(false);
        initViews();
    }

    private void showProgressDialogWithTitle(final String substring) {
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                mProgressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
                mProgressDialog.setCancelable(false);
                mProgressDialog.setMessage(substring);
                mProgressDialog.show();
            }
        });
    }

    private void hideProgressDialogWithTitle() {
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                mProgressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
                mProgressDialog.dismiss();
            }
        });
    }

    public AlertDialog.Builder getBuilder() {
        if (builder == null) {
            builder = new AlertDialog.Builder(this);
            builder.setTitle("Checking details...\n");

            final ProgressBar progressBar = new ProgressBar(this);
            LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
            progressBar.setLayoutParams(layoutParams);
            builder.setView(progressBar);
        }
        return builder;
    }

    private void initViews() {
        surfaceView = findViewById(R.id.qr_code_view);
    }

    private void initialiseDetectorsAndSources() {

        Toast.makeText(getApplicationContext(), "Barcode scanner started",
Toast.LENGTH_SHORT).show();

        barcodeDetector = new BarcodeDetector.Builder(this)
                .setBarcodeFormats(Barcode.ALL_FORMATS)
                .build();
        cameraSource = new CameraSource.Builder(this, barcodeDetector)
                .setRequestedPreviewSize(1920, 1080)
                .setAutoFocusEnabled(true) //you should add this feature
                .build();

        surfaceView.getHolder().addCallback(new SurfaceHolder.Callback() {
            @Override
            public void surfaceCreated(SurfaceHolder holder) {
                try {
                    if
(ActivityCompat.checkSelfPermission(ScannedBarcodeActivity.this,
Manifest.permission.CAMERA) == PackageManager.PERMISSION_GRANTED) {
                        cameraSource.start(surfaceView.getHolder());
                    } else {

ActivityCompat.requestPermissions(ScannedBarcodeActivity.this, new
                                String[]{Manifest.permission.CAMERA},
REQUEST_CAMERA_PERMISSION);
                    }

                } catch (IOException e) {
                    e.printStackTrace();
                }


            }

            @Override
            public void surfaceChanged(SurfaceHolder holder, int format, int
width, int height) {
            }

            @Override
            public void surfaceDestroyed(SurfaceHolder holder) {
                cameraSource.stop();
            }
        });


        barcodeDetector.setProcessor(new Detector.Processor<Barcode>() {
            @Override
            public void release() {
                Toast.makeText(getApplicationContext(), "To prevent memory leaks
barcode scanner has been stopped", Toast.LENGTH_SHORT).show();
            }

            @Override
            public void receiveDetections(Detector.Detections<Barcode> detections)
{
                final SparseArray<Barcode> barcodes =
detections.getDetectedItems();
                if (barcodes.size() != 0) {
                    String val = barcodes.valueAt(0).displayValue;
                    Log.d("Shubham", "receiveDetections: " + val);
                    showProgressDialogWithTitle("Checking Details!");
                    checkApplicationStatus(val);
                }
            }
```

Name & Sign of External:

```
        });
    }

    private void checkApplicationStatus(String val) {
        FirebaseFirestore db = FirebaseFirestore.getInstance();
        db.collection("ePasses")
                .document(val)
                .get()
                .addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>()
{
                    @Override
                    public void onComplete(Task<DocumentSnapshot> task) {
//                          progressDialog.dismiss();
                        hideProgressDialogWithTitle();
                        DocumentSnapshot snapshot = task.getResult();
                        if (snapshot != null && snapshot.exists()) {
                            long time = (long) snapshot.get("expiryDate");
                            if (time >= System.currentTimeMillis()) {
                                showApprovedOrRejectPage(1);
                            } else {
                                showApprovedOrRejectPage(0);
                            }
                        } else {
                            showApprovedOrRejectPage(0);
                        }
                    }
                }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(Exception e) {
                showApprovedOrRejectPage(0);
                hideProgressDialogWithTitle();
            }
        });
    }

    // 1 means success and 0 means failure
    private void showApprovedOrRejectPage(int i) {
        Intent intent = new Intent(this, SuccessOrFailureActivity.class);
        intent.putExtra("resultValue", i);
        this.startActivity(intent);
    }


    @Override
    protected void onPause() {
        super.onPause();
        cameraSource.release();
    }

    @Override
    protected void onResume() {
        super.onResume();
        initialiseDetectorsAndSources();
    }
}
```

Name & Sign of External:

Success or Fail Activity

```java
package com.epass_conductor;

import android.Manifest;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.util.SparseArray;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;


import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.android.gms.vision.CameraSource;
import com.google.android.gms.vision.Detector;
import com.google.android.gms.vision.barcode.Barcode;
import com.google.android.gms.vision.barcode.BarcodeDetector;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.io.IOException;

public class ScannedBarcodeActivity extends AppCompatActivity {


    private SurfaceView surfaceView;
    private BarcodeDetector barcodeDetector;
    private CameraSource cameraSource;
    private static final int REQUEST_CAMERA_PERMISSION = 201;
    private String intentData = "";
    private boolean isEmail = false;
    private ProgressDialog mProgressDialog;
    AlertDialog progressDialog;
    AlertDialog.Builder builder;
```

Name & Sign of External:

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scan_barcode);
        mProgressDialog = new ProgressDialog(this);
        progressDialog = getBuilder().create();
        progressDialog.setCancelable(false);
        initViews();
    }

    private void showProgressDialogWithTitle(final String substring) {
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                mProgressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
                mProgressDialog.setCancelable(false);
                mProgressDialog.setMessage(substring);
                mProgressDialog.show();
            }
        });
    }

    private void hideProgressDialogWithTitle() {
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                mProgressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
                mProgressDialog.dismiss();
            }
        });
    }

    public AlertDialog.Builder getBuilder() {
        if (builder == null) {
            builder = new AlertDialog.Builder(this);
            builder.setTitle("Checking details...\n");

            final ProgressBar progressBar = new ProgressBar(this);
            LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
            progressBar.setLayoutParams(layoutParams);
            builder.setView(progressBar);
        }
        return builder;
    }

    private void initViews() {
        surfaceView = findViewById(R.id.qr_code_view);
    }

    private void initialiseDetectorsAndSources() {

        Toast.makeText(getApplicationContext(), "Barcode scanner started",
Toast.LENGTH_SHORT).show();

        barcodeDetector = new BarcodeDetector.Builder(this)
                .setBarcodeFormats(Barcode.ALL_FORMATS)
```

Name & Sign of External:

```
                .build();

        cameraSource = new CameraSource.Builder(this, barcodeDetector)
                .setRequestedPreviewSize(1920, 1080)
                .setAutoFocusEnabled(true) //you should add this feature
                .build();

        surfaceView.getHolder().addCallback(new SurfaceHolder.Callback() {
            @Override
            public void surfaceCreated(SurfaceHolder holder) {
                try {
                    if
(ActivityCompat.checkSelfPermission(ScannedBarcodeActivity.this,
Manifest.permission.CAMERA) == PackageManager.PERMISSION_GRANTED) {
                        cameraSource.start(surfaceView.getHolder());
                    } else {

ActivityCompat.requestPermissions(ScannedBarcodeActivity.this, new
                                String[]{Manifest.permission.CAMERA},
REQUEST_CAMERA_PERMISSION);
                    }

                } catch (IOException e) {
                    e.printStackTrace();
                }


            }

            @Override
            public void surfaceChanged(SurfaceHolder holder, int format, int
width, int height) {
            }

            @Override
            public void surfaceDestroyed(SurfaceHolder holder) {
                cameraSource.stop();
            }
        });


        barcodeDetector.setProcessor(new Detector.Processor<Barcode>() {
            @Override
            public void release() {
                Toast.makeText(getApplicationContext(), "To prevent memory leaks
barcode scanner has been stopped", Toast.LENGTH_SHORT).show();
            }

            @Override
            public void receiveDetections(Detector.Detections<Barcode> detections)
{
                final SparseArray<Barcode> barcodes =
detections.getDetectedItems();
                if (barcodes.size() != 0) {
                    String val = barcodes.valueAt(0).displayValue;
                    Log.d("Shubham", "receiveDetections: " + val);
                    showProgressDialogWithTitle("Checking Details!");
                    checkApplicationStatus(val);
                }
```

Name & Sign of External:

```
            }
        });
    }

    private void checkApplicationStatus(String val) {
        FirebaseFirestore db = FirebaseFirestore.getInstance();
        db.collection("ePasses")
                .document(val)
                .get()
                .addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>()
{
                    @Override
                    public void onComplete(Task<DocumentSnapshot> task) {
//                        progressDialog.dismiss();
                        hideProgressDialogWithTitle();
                        DocumentSnapshot snapshot = task.getResult();
                        if (snapshot != null && snapshot.exists()) {
                            long time = (long) snapshot.get("expiryDate");
                            if (time >= System.currentTimeMillis()) {
                                showApprovedOrRejectPage(1);
                            } else {
                                showApprovedOrRejectPage(0);
                            }
                        } else {
                            showApprovedOrRejectPage(0);
                        }
                    }
                }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(Exception e) {
                showApprovedOrRejectPage(0);
                hideProgressDialogWithTitle();
            }
        });
    }

    // 1 means success and 0 means failure
    private void showApprovedOrRejectPage(int i) {
        Intent intent = new Intent(this, SuccessOrFailureActivity.class);
        intent.putExtra("resultValue", i);
        this.startActivity(intent);
    }


    @Override
    protected void onPause() {
        super.onPause();
        cameraSource.release();
    }

    @Override
    protected void onResume() {
        super.onResume();
        initialiseDetectorsAndSources();
    }
}
```

CHAPTER 6

SUMMARY

6.1 Summary

In this chapter, the interfaces of the system that help the user to interact with the system. Besides that, the testing approach is the methods that are used for test the functional and nonfunctional of the system for each main module on the system by using the test case.

CHAPTER 7

CONCLUSION

7.1 Conclusion of the Project

The paper summarizes the current issues in bus ticketing system. To overcome from this, we are working towards android platform. We have identified the current gaps and open research areas. Our research will focus on these open problems and propose effective solutions for the same.

This paper introduces on how to secure passenger information. To overcome the drawbacks of manual ticketing system we are using QR-Code for security purpose of passengers' information in the propose system.

Name & Sign of External:

CHAPTER 8

References

[1] “Predicting Transit Vehicle Arrival Times”. Kidwell,B, Geographic Laboratory, Bridgewater State College, Bridgewater, Mass., 2001.

[2] “Public Transport System Ticketing system using RFID and ARM processor Perspective Mumbai bus facility B.E.S.T”, SaurabhChatterjee, Prof. BalramTimande, International Journalof Electronics and Computer Science Engineering., 2012.

[3] “Taking an Electronic Ticketing System to the Cloud: Design and Discussion”. Filipe Araujo, Marilia Curado, Pedro Furtado, Raul Barbosa CISUC, Dept. of Informatics Engineering, University of Coimbra, Portugal filipius@uc.pt, marilia, pnf, rbarbosa@dei.uc.pt 2013.

Name & Sign of External:

Name & Sign of External: