SplitX

A PROJECT REPORT

in partial fulfillment for the award of the degree

of

Bachelor of Computer Application(BCA)

Department of Computational Science

BRAINWARE UNIVERSITY

398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

June,2022

Abstract

SplitX is hands down one of the best apps to split bills with roommates, friends, family etc. The app's super easy to use with its sleek user interface. The project describes and presents a application designed and built with the intention of reducing paper-based transactions by taking advantage of the technology and recent legislature which allows the service to eliminate deficiencies associated with financial transactions. This bill splitting app keeps a running total of who owes whom how much money. So, it’s much more convenient to settle the bills. Most importantly, this ensures you have an easy way to keep track of miscellaneous expenses. This split-bill app is a system flexible enough to perform bill splliting of many people or you can say a group of friends. The ultimate aim of this app is to split bills and show that who owes how much money to whom and also track the expenses or the usage of the specific user.

INDEX

1.INTRODUCTION

1.1 Introduction

It is a mobile application intended to run on android devices. It is design to fulfil the needs of the user by reducing their efforts for the settlement of the bill. The application encourages corresponding users help in who owes who, and for what. Aim to provide user the best approach to help user and their companion to share expenses easily. This application will let bunch users and their companions to have detailed view inside this application around the individual costs and to settle them by using payment gateways linked with the application or by using the e-wallets. This application allows its user to add remark to an expense, click on the expense name in any expense list. Bill posting will have space for comments. It will also have the notification option to notify each time somebody adds a remark to an expense user is on, or user can withdraw to posted bill. It will also have the graphical representation of the expenses weekly, monthly, quarterly or yearly. The additional feature that we are going to add in this application in the near future is that it enables user to settle the bill without disturbing the interaction with this application, that is by providing the payment gateway like paytm, paypal or by using the e-wallet for the settlement of the bill.

1.2 Modules of the Project

1.2.1 User Signup

- User need to provide an email address.
- User needs to type a password.
- User needs to click on signup.

1.2.2 Database Storing

- After the signup a data of the new user is stored in the database.

1.2.3 User Login

- Type the email id.
- Type the password.
- Click on login.

1.2.4 Adding of Members

- Click on add members.
- Add as many members as you want.

1.2.5 Data Entry

- Type the name of the members.
- Type the specific amount paid by each of them.

1.2.6 Splitting of Bills

- Click on Split bill.
- Output generation (Equally splitted amount among all members).

2. Requirement Specification

2.1 Introduction

While developing an android application some specific requirements are there which the API must have. If these are not there in a system the app will not run properly. For our application of bill splitting we need some hardware and software requirements which are to be met so that the app can run smoothly. This section highlights the functional and non function requirements.

2.2 Hardware Requirements

Hardware requirements like memory restrictions, cache size, processor, RAM size etc for the application to run:-

Minimum Hardware Requirements

- RAM - Minimum RAM of 3 GB
- Processor - Qualcomm Snapdragon (version 630 and higher), Samsung Exynos (version 5433 or higher), Mediatek(version MT6750 or higher) etc.

Preferred Hardware Requirements

- RAM – 4 GB of RAM
- Processor – Qualcomm Snapdragon (version 710 or higher), Samsung Exynos(version 7880 or higher), Mediatek (version Helio G70 or higher), etc.

2.2 Software Requirements

Software Requirements like Android versions which are needed for the application to run:-

Minimum Software Requirements

Android 4.1 or higher.

Preferred Software Requirements

Android 9 or higher.

3.ANALYSIS

3.1 Existing System

- Calculation Errors.
- Lots of paper usage.
- Time wastage.
- No specific expense shown.
- No data storage.
- Complications.

3.2 Proposed System

- No Errors
- No paper usage.
- Time Efficient.
- Specific Expenses.
- Data Storage Availability.
- Easy to use.

3.3 Feasibility Study

As per our research and project planning this android application named “SplitX” is a fully practical application which will be adapted by users easily as the necessity of the app in current period is very high and mostly in need for youth generations targeting the busy and hurry lifestyle of their hectic life. This application will indeed be a very simple and useful application for everyone.

3.4 Software Specifications

- RAM – 4 GB.
- Android Version – 4.1 or higher.
- Storage – 50 mb of free space.

4.DESIGN

4.1 System Design

4.1.1 Introduction to UML

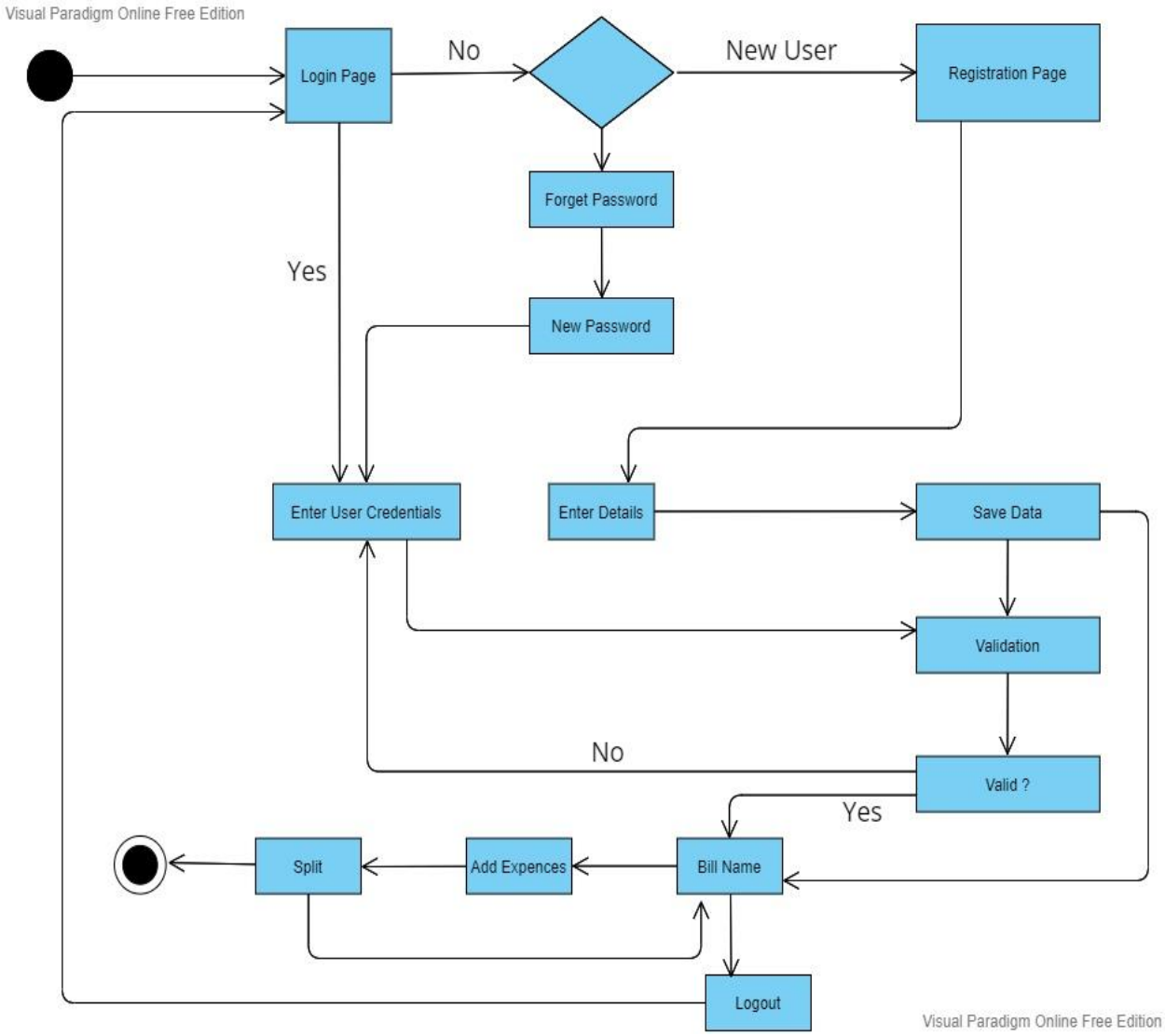
Unified Modeling Language (UML) is a general purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering. UML is not a programming language, it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. It' s been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.UML is linked with object oriented design and analysis. UML makes the use of elements and forms associations between them to form diagrams. Diagrams in UML can be broadly classified as:-

- Structural Diagrams – Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams, Object Diagrams, Class Diagrams and Deployment Diagrams.

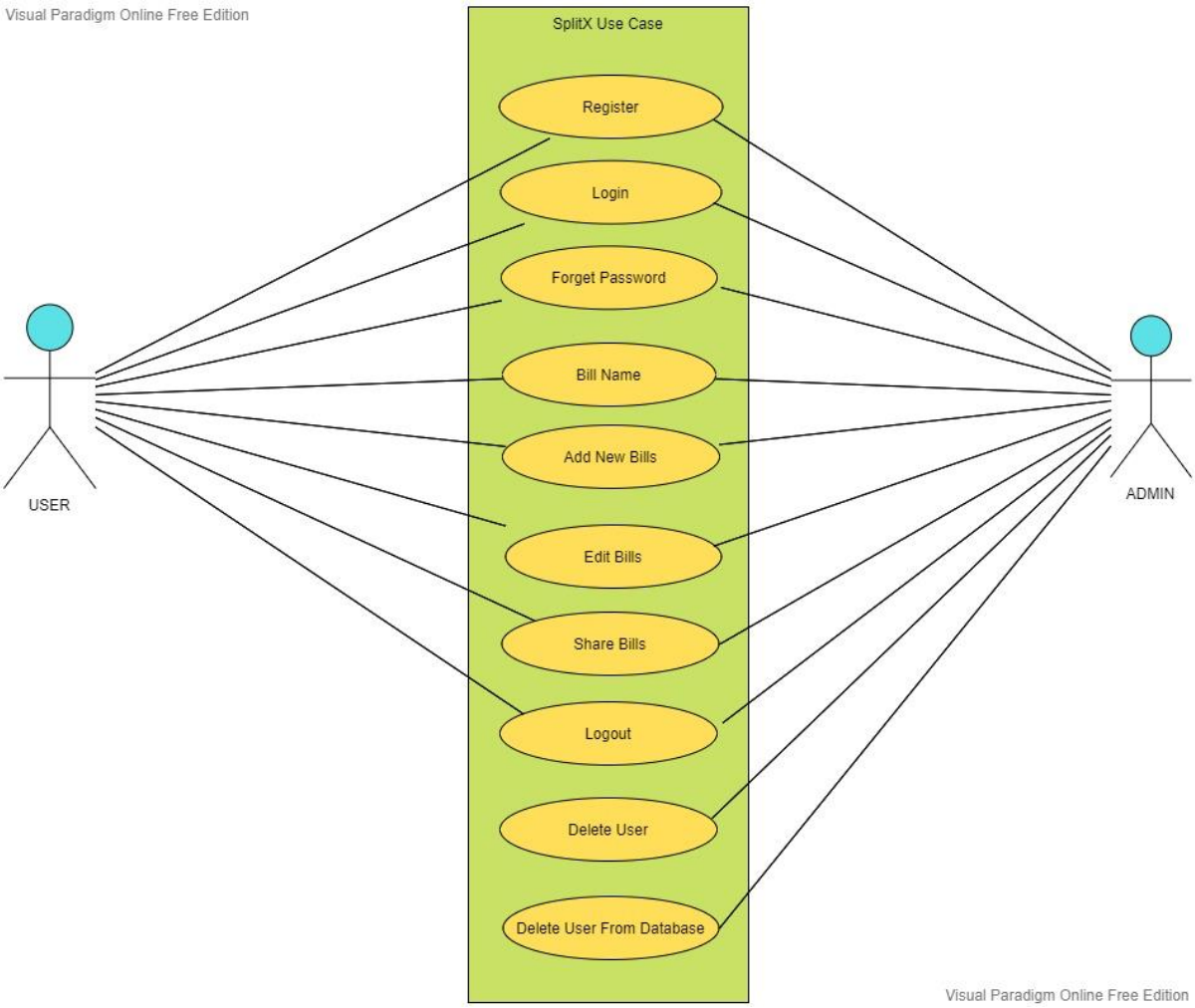
- Behavior Diagrams – Capture dynamic aspects or behavior of the system. Behavior diagrams include: Use Case Diagrams, State Diagrams, Activity Diagrams and Interaction Diagrams.

4.1.2 UML Diagrams of our Project

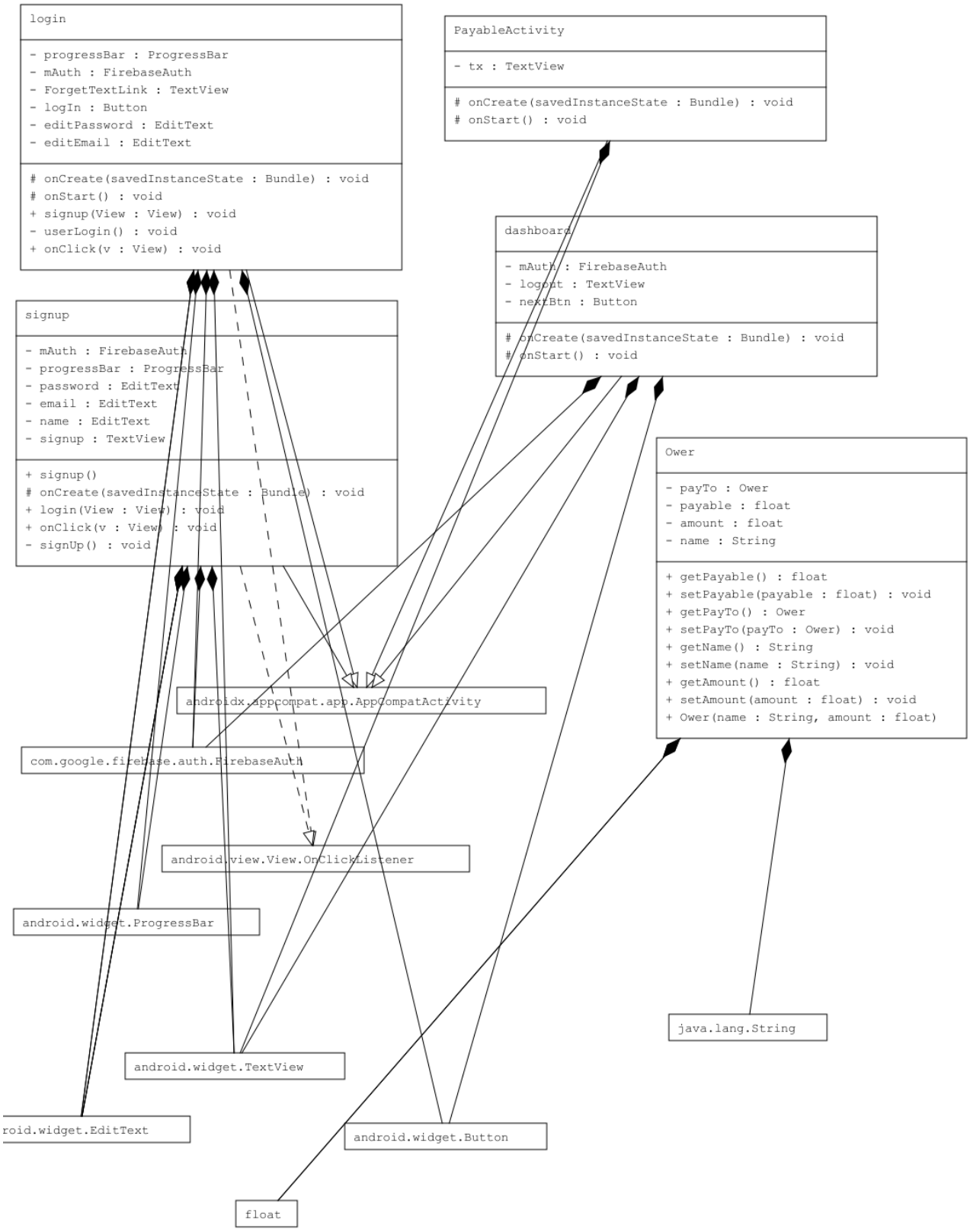
ACTIVITY DIAGRAM



USE-CASE DIAGRAM



SYSTEM GENERATED CLASS DIAGRAM



5.SYSTEM IMPLEMENTATION

5.1 Introduction

The purpose of System Implementation can be summarized as follows: making the new system available to a prepared set of users (the deployment), and positioning on-going support and maintenance of the system within the Performing Organization (the transition). At a finer level of detail, deploying the system consists of executing all steps necessary to educate the Consumers on the use of the new system, placing the newly developed system into production, confirming that all data required at the start of operations is available and accurate, and validating that business functions that interact with the system are functioning properly. Transitioning the system support responsibilities involves changing from a system development to a system support and maintenance mode of operation, with ownership of the new system moving from the Project Team to the Performing Organization.

A key difference between System Implementation and all other phases of the lifecycle is that all project activities up to this point have been performed in safe, protected, and secure environments, where project issues that arise have little or no impact on day-to-day business operations. Once the system goes live, however, this is no longer the case. Any miscues at this point will almost certainly translate into direct operational and/or financial impacts on the Performing Organization. It is through the careful planning, execution, and management of System Implementation activities that the Project Team can minimize the likelihood of these occurrences, and determine appropriate contingency plans in the event of a problem.

5.2 Sample Code

signUp.java

```
package com.example.splitx;

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

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

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.FirebaseDatabase;

import java.util.regex.Pattern;

public class signup extends AppCompatActivity implements
View.OnClickListener {

    private TextView signup;
    private EditText name,email,password;
    private ProgressBar progressBar;

    private FirebaseAuth mAuth;  // Initialize Firebase Auth

    public signup() {
    }

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

        mAuth=FirebaseAuth.getInstance();

        signup=(Button) findViewById(R.id.button4);
        signup.setOnClickListener(this);

        name=(EditText) findViewById(R.id.editTextTextPersonName2);
        email=(EditText) findViewById(R.id.editTextTextPersonName3);
        password=(EditText) findViewById(R.id.editTextTextPassword2);

        progressBar=(ProgressBar) findViewById(R.id.progressBar);
    }
```

```
    public void login(View View){
        Intent intent2= new Intent(this,login.class);
        startActivity(intent2);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.button4:
                signUp();
                break;

        }
    }

    private void signUp() {
        String name1=name.getText().toString().trim();
        String email1=email.getText().toString().trim();
        String password1=password.getText().toString().trim();

        if (name1.isEmpty()){
            name.setError("Name is Required");
            name.requestFocus();
            return;
        }

        if (email1.isEmpty()){
            email.setError("Email is Required");
            email.requestFocus();
            return;
        }

        if(!Patterns.EMAIL_ADDRESS.matcher(email1).matches()){
            email.setError("Please Provide valid email");
            email.requestFocus();
            return;
        }

        if (password1.isEmpty()){
            password.setError("Password is required");
            password.requestFocus();
            return;
        }
        if (password1.length() < 6){
            password.setError("Min password length should be 6
characters");
            password.requestFocus();
            return;
        }

        progressBar.setVisibility(View.GONE);
        mAuth.createUserWithEmailAndPassword(email1,password1)
                .addOnCompleteListener(new OnCompleteListener<AuthResult>()
{
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task)
{

                        if (task.isSuccessful()){
                           User user=new User(name1,email1);
```

```
FirebaseDatabase.getInstance().getReference("Users")

.child(FirebaseAuth.getInstance().getCurrentUser().getUid())

.setValue(user).addOnCompleteListener(task1 -> {
                                        if (task1.isSuccessful()){
                                            Toast.makeText(signup.this,
"User has been registered successfully", Toast.LENGTH_LONG).show();

progressBar.setVisibility(View.GONE);

                                            // redirect to login layout
                                        }else {
                                            Toast.makeText(signup.this,
"Failed to register, try again", Toast.LENGTH_LONG).show();

progressBar.setVisibility(View.GONE);
                                        }
                                    });

                        }
                        else {
                            Toast.makeText(signup.this, "Failed to
register, try again", Toast.LENGTH_LONG).show();
                            progressBar.setVisibility(View.GONE);
                        }
                    }
                });
    }
}
```

login.java

```
package com.example.splitx;

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

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class login extends AppCompatActivity implements
View.OnClickListener{
```

```
private EditText editEmail,editPassword;
private Button logIn;

private FirebaseAuth mAuth;
private ProgressBar progressBar;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.login);
    logIn=(Button) findViewById(R.id.button);

    editEmail=(EditText) findViewById(R.id.editTextTextPersonName);
    editPassword=(EditText) findViewById(R.id.editTextTextPassword);

    progressBar = findViewById(R.id.progressBar);

    mAuth=FirebaseAuth.getInstance();


}

@Override
protected void onStart() {
    super.onStart();
    logIn.setOnClickListener(it -> {
        userLogin();
    });

    if(FirebaseAuth.getInstance().getCurrentUser() != null){
        startActivity(new Intent(this, dashboard.class));
        finish();
    }
}

public void signup(View View){
    Intent intent=new Intent(this,signup.class);
    startActivity(intent);
}


private void userLogin() {
    Log.d("error_trap", "I am called!");
    String email2=editEmail.getText().toString().trim();
    String password2=editPassword.getText().toString().trim();

    if (email2.isEmpty()){
        editEmail.setError("Email is Required");
        editEmail.requestFocus();
        return;
    }

    if(!Patterns.EMAIL_ADDRESS.matcher(email2).matches()){
        editEmail.setError("Please Provide valid email");
        editEmail.requestFocus();
        return;
    }

    if (password2.isEmpty()){
        editPassword.setError("Password is required");
```

```
            editPassword.requestFocus();
            return;
        }
        if (password2.length() < 6){
            editPassword.setError("Min password length should be 6
characters");
            editPassword.requestFocus();
            return;
        }

        progressBar.setVisibility(View.GONE);


mAuth.signInWithEmailAndPassword(email2,password2).addOnCompleteListener(ne
w OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()){
                    // redirect to user profile
                    startActivity(new Intent(login.this,dashboard.class));

                }else {
                    Toast.makeText(login.this, "Failed to login, Please
check your credentials", Toast.LENGTH_SHORT).show();
                }
            }
        });

    }

    @Override
    public void onClick(View v) {

    }
}
```

payableActivity.java

```
package com.example.splitx;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.widget.TextView;

import com.example.splitx.models.Payable;

import java.util.ArrayList;

public class PayableActivity extends AppCompatActivity {

    private TextView tx;

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

```
        setContentView(R.layout.activity_payable);

        tx = findViewById(R.id.payable_activity_text);

    }

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

        Intent intent = this.getIntent();

        ArrayList<Payable> list = (ArrayList<Payable>)
intent.getSerializableExtra("payableList");

        StringBuilder message = new StringBuilder("Here is the ower list:
");

        for(int i = 0; i < list.size(); i++){
           message.append("\n\n").append(list.get(i).getName()).append(",
you owe:");
           for (String ower: list.get(i).getOwedList().keySet()){
               if(list.get(i).getOwedList().get(ower) > 0f){
                   message.append("\n").append(ower).append(" -
").append(list.get(i).getOwedList().get(ower));
               }
           }
           message.append("\n--------------\n");
        }

        tx.setText(message.toString());

    }
}
```

login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:onClick="onClick"
    tools:context=".login">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:scaleType="fitXY"
        android:src="@drawable/element_1"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
```

```
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/middle_rectange"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.49"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.63" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/interbold"
        android:text="Login"
        android:textColor="@color/black"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.179"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.197" />

    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/line_1"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.169"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.235" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/intersemibold"
        android:text="Email"
        android:textColor="@color/black"
        android:textSize="16sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.163"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.316" />

    <EditText
        android:id="@+id/editTextTextPersonName"
```

```
        android:layout_width="283dp"
        android:layout_height="50dp"
        android:drawableStart="@drawable/ic_baseline_email_24"
        android:drawablePadding="12dp"
        android:ems="10"
        android:fontFamily="@font/interregular"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.468"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.375" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/intersemibold"
        android:text="Password"
        android:textColor="@color/black"
        android:textSize="16sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.185"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.461" />

    <EditText
        android:id="@+id/editTextTextPassword"
        android:layout_width="283dp"
        android:layout_height="wrap_content"
        android:drawableStart="@drawable/ic_baseline_lock_24"
        android:drawablePadding="12dp"
        android:ems="10"
        android:inputType="textPassword"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.484"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.521" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/interregular"
        android:text="Forget password ?"
        android:textColor="@color/black"
        android:textSize="11sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.783"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.59" />

    <Button
        android:id="@+id/button"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="#3B3B3B"
        android:text="Login"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.683" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/intersemibold"
        android:text="or login with"
        android:textColor="@color/black"
        android:textSize="12sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.734" />

    <ImageView
        android:id="@+id/imageView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/phone"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.818" />

    <TextView
        android:id="@+id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/interregular"
        android:text="Don't have an account ?"
        android:textColor="@color/black"
        android:textSize="11sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.297"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.886" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="#3B3B3B"
        android:onClick="signup"
        android:text="Sign up"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
```

```
        app:layout_constraintHorizontal_bias="0.205"
        app:layout_constraintStart_toEndOf="@+id/textView6"
        app:layout_constraintTop_toTopOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.904" />

    <ProgressBar
        android:id="@+id/progressBar"
        style="?android:attr/progressBarStyle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="gone"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

List_input_card.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:padding="16dp"
        android:layout_height="wrap_content">

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/list_name_text_holder"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="8dp"
            app:layout_constraintEnd_toEndOf="parent"
            app:helperText="Name"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent">

            <com.google.android.material.textfield.TextInputEditText
                android:id="@+id/list_name_text"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="textPersonName"
                android:padding="8dp"
                android:textSize="18sp" />
        </com.google.android.material.textfield.TextInputLayout>
```

```
        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/list_amount_text_holder"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:layout_marginBottom="1dp"
            android:padding="8dp"
            app:layout_constraintBottom_toTopOf="@+id/list_remove_btn"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:helperText="Amount"

app:layout_constraintTop_toBottomOf="@+id/list_name_text_holder">

            <com.google.android.material.textfield.TextInputEditText
                android:id="@+id/list_amount_text"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="numberDecimal"
                android:padding="8dp"
                android:textSize="18sp" />
        </com.google.android.material.textfield.TextInputLayout>

        <Button
            android:id="@+id/list_remove_btn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:layout_marginEnd="24dp"
            android:layout_marginRight="24dp"
            android:backgroundTint="#615A5A"
            android:text="Remove"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintTop_toBottomOf="@+id/list_amount_text_holder" />

</androidx.constraintlayout.widget.ConstraintLayout>

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

6.TESTING

6.1 Introduction

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring the specification, functionality, and performance of a software program or application.

Software testing can be divided into two steps:

1. **Verification:** it refers to the set of tasks that ensure that the software correctly implements a specific function.

2. **Validation:** it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Different types of software testing:

Software Testing can be broadly classified into two types:

1. **Manual Testing:** Manual testing includes testing software manually, i.e., without using any automation tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected

behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

Testers use test plans, test cases, or test scenarios to test software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

2. **Automation Testing:** Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves the automation of a manual process. Automation Testing is used to re-run the test scenarios quickly and repeatedly, that were performed manually in manual testing.

Apart from regression testing, automation testing is also used to test the application from a load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money when compared to manual testing.

Different types of Software Testing Techniques:

Software testing techniques can be majorly classified into two categories:

1. **Black Box Testing:** The technique of testing in which the tester doesn't have access to the source code of the software and is conducted at the software interface without any concern with the internal logical structure of the software is known as black-box testing.

2. **White-Box Testing:** The technique of testing in which the tester is aware of the internal workings of the product, has access to its source code, and is conducted by making sure that all internal operations are performed according to the specifications is known as white box testing.

Black Box Testing	White Box Testing
Internal workings of an application are not required.	Knowledge of the internal workings is a must.
Also known as closed box/data-driven testing.	Also known as clear box/structural testing.
End users, testers, and developers.	Normally done by testers and developers.
This can only be done by a trial and error method.	Data domains and internal boundaries can be better tested.

Different levels of software testing:

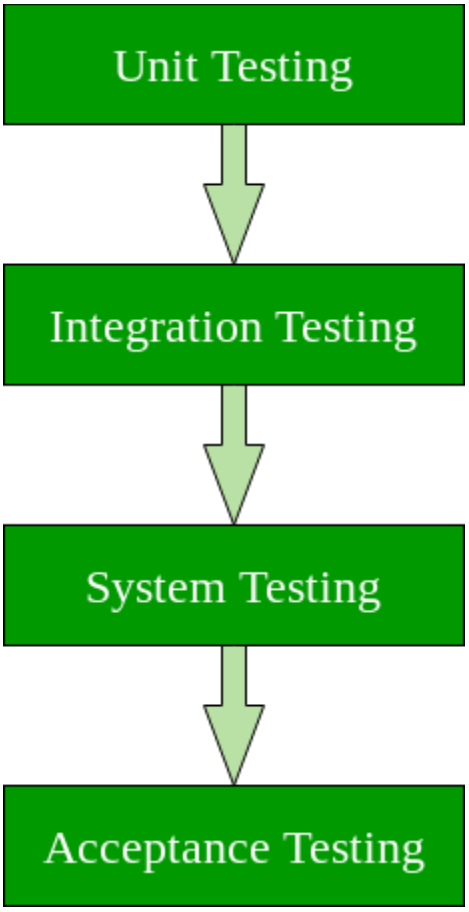
Software level testing can be majorly classified into 4 levels:

1. **Unit Testing:** A level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

2. **Integration Testing:** A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

3. **System Testing:** A level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

4. **Acceptance Testing:** A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.



6.2 Testing Methods

So as concerned with our project we have used 4 testing methods that are:

1.Alpha Testing

As we know what alpha testing means, basically in this testing method the application is tested by developers who have developed the code and have knowledge of understanding the errors faced.

So our group has used this method to see the possible errors and bugs that are there in the application before giving the application to the real world users.

2.Beta Testing

In this testing method as we know that its done by some real world users who have knowledge of testing and know that what are the problems that are faced while the execution of the application.

So we have shared our application to some of our friends who have knowledge on testing and can say about the errors faced on the application.

3.Black Box Testing

As we know in this testing method the application is checked by a person who does not know about the backend process, the only thing he knows is about the frontend process or we can what he sees in the application . It is also a part of alpha testing.

So our frontend designer has tested this application and said about the errors or problems which he faced as he does not have any knowledge about the backend process.

4.White Box Testing

In this testing method we know that it is done by a person who knows both about the frontend and the backend process. Moreover it is a part of alpha testing method also.

So our group has tested this application by checking both the design and the code that we have used to build this application and look for all possible errors and bugs.

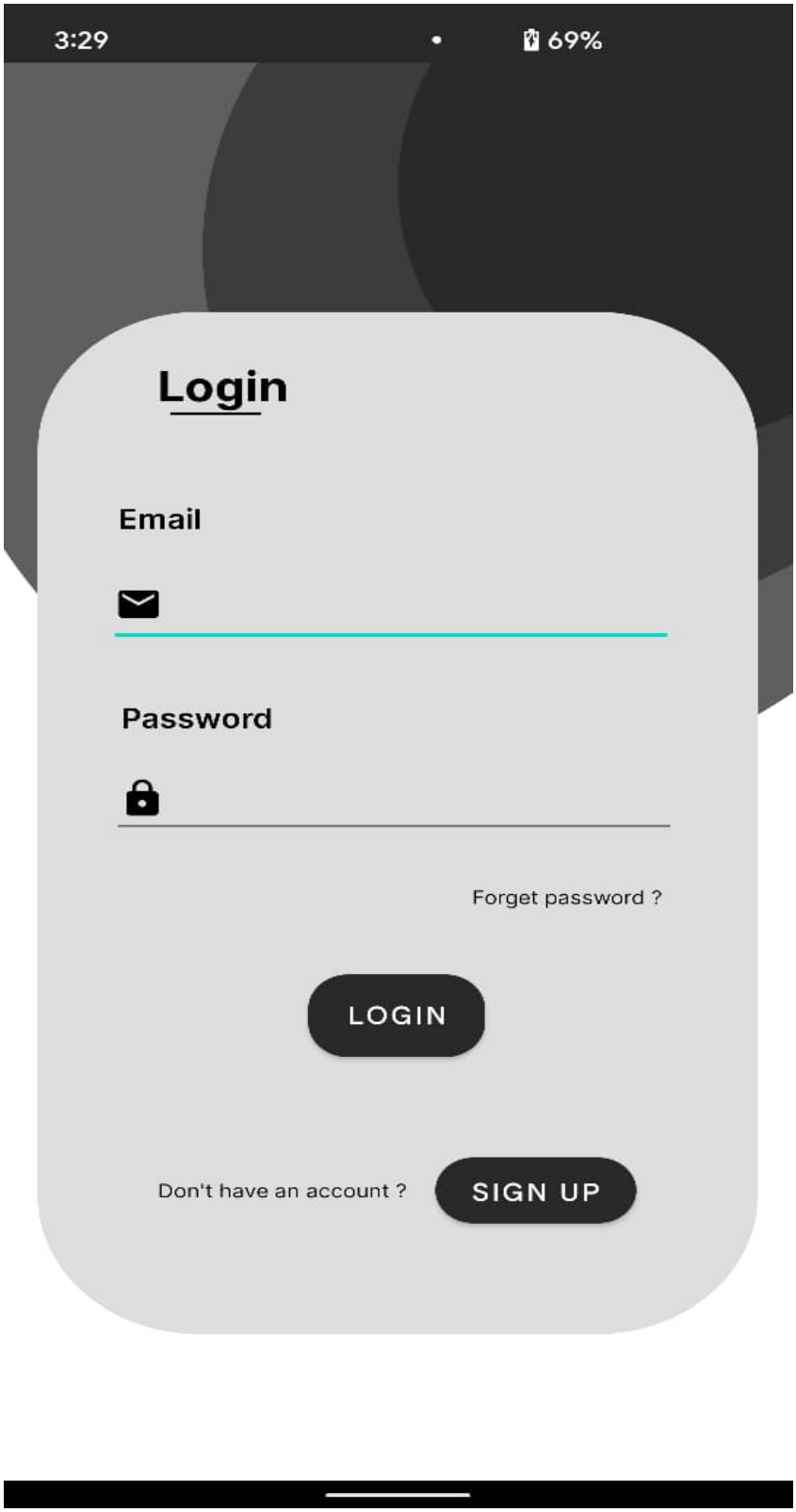
Testing Procedure and Results

SR.No	Test Case	Procedure	Pre-Condition	Result
1	App opening	Click on the app	App is closed	App is opened
2	Login	Appears automatically	App is getting opened	Login page opens
3	Forget Password	On clicking forget password	Login page	Forget Password opens
4	Signup	On clicking signup	Login page	Signup page opens
5	Home page	After clicking login	Login page	Home page opens
6	Member & Amount Entry	Data Entry	Home page	Member name and amount entry
7	Split	On clicking Split	Member and Amount Entry	Bill Splitting
8	Output	Output window	Split Calculation	Name with bill splitting

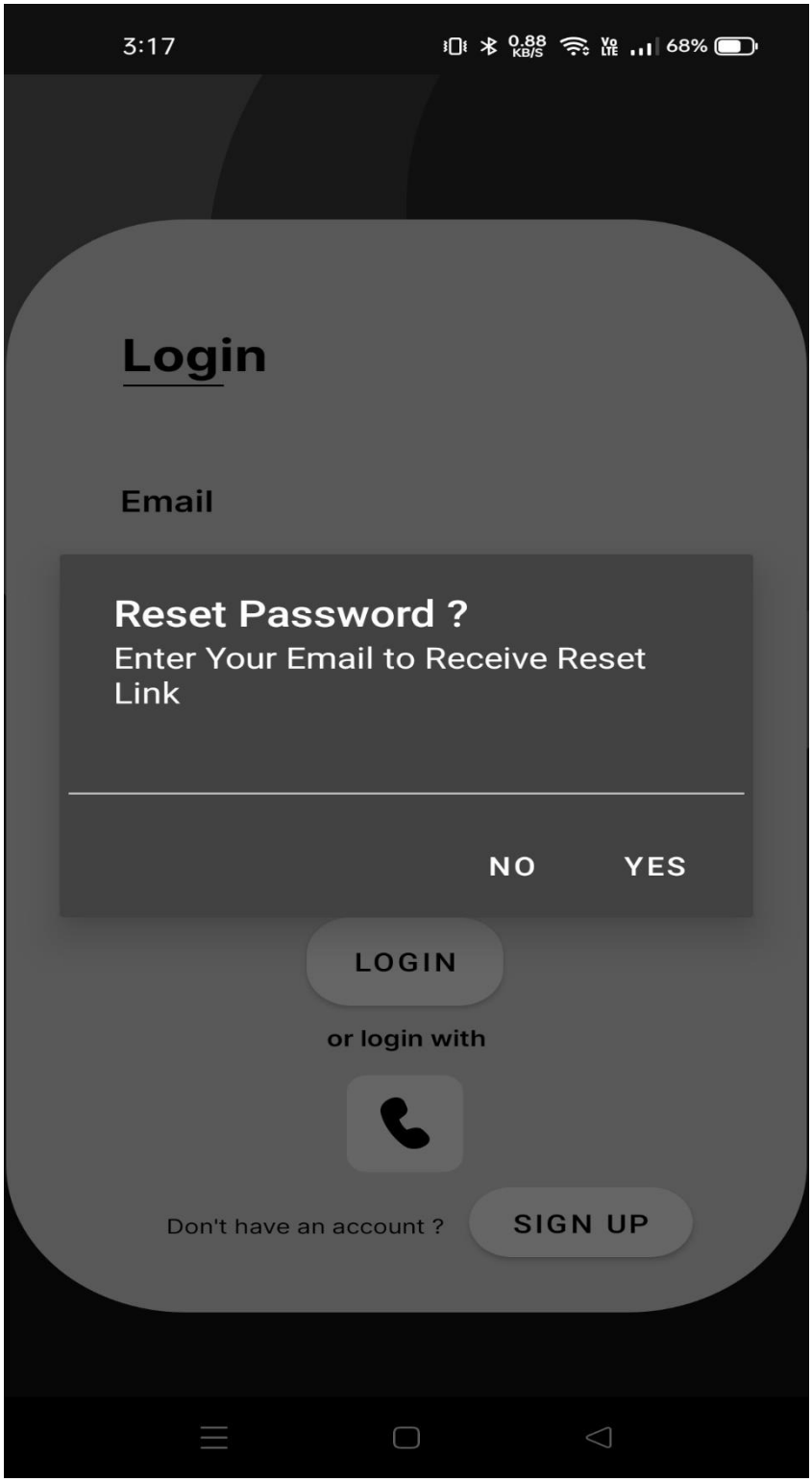
7. SAMPLE SCREENSHOTS

Splash Screen

LOGIN PAGE


3:29
69%
Login
Email
Password
Forget password ?
LOGIN
Don't have an account ?
SIGN UP

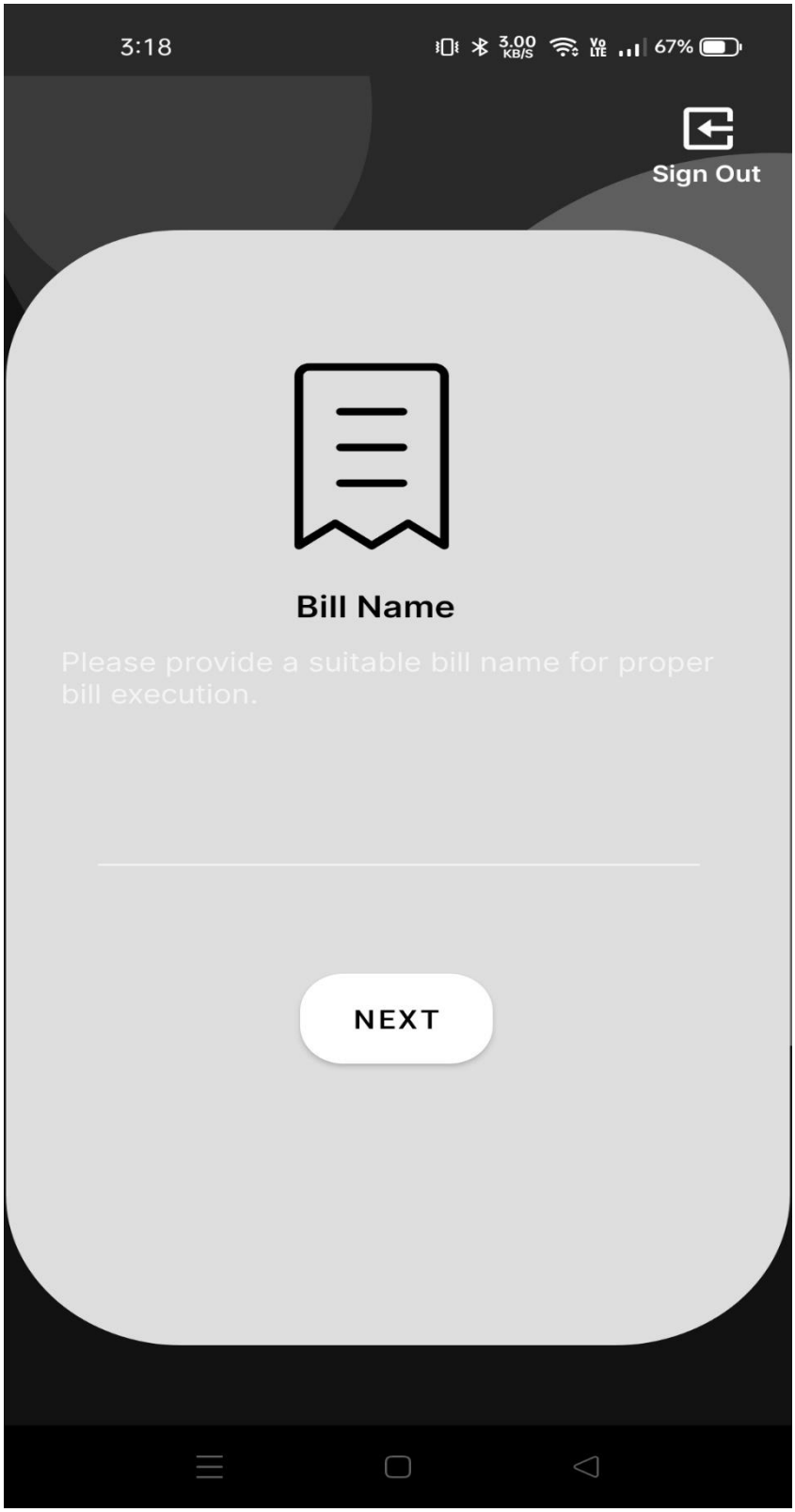
FORGOT PASSWORD



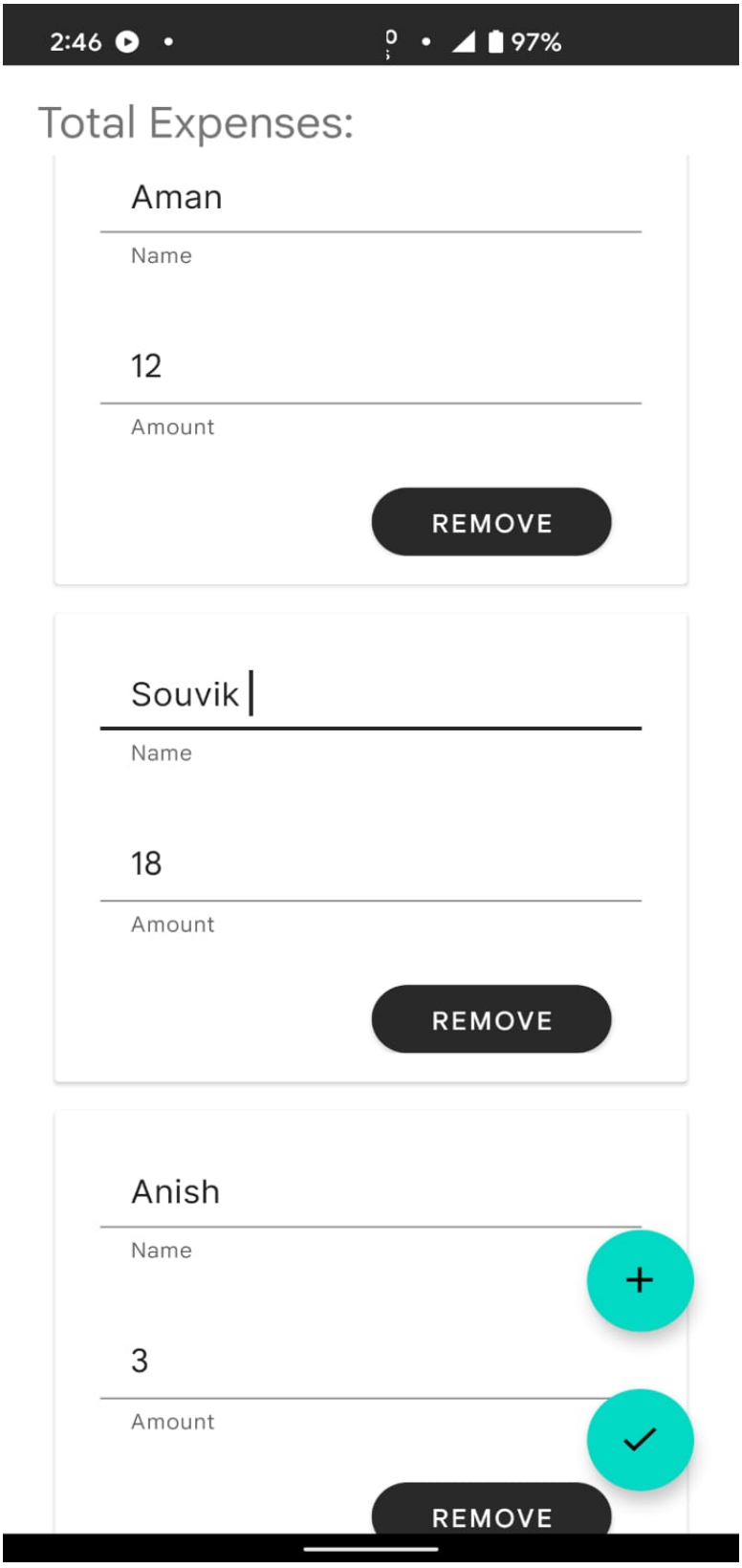
SIGNUP PAGE


3:17
68%
Sign Up
Name
Email
xyz@email.com
Password
Min 6 characters
SIGN UP
Already have an account ?
LOGIN

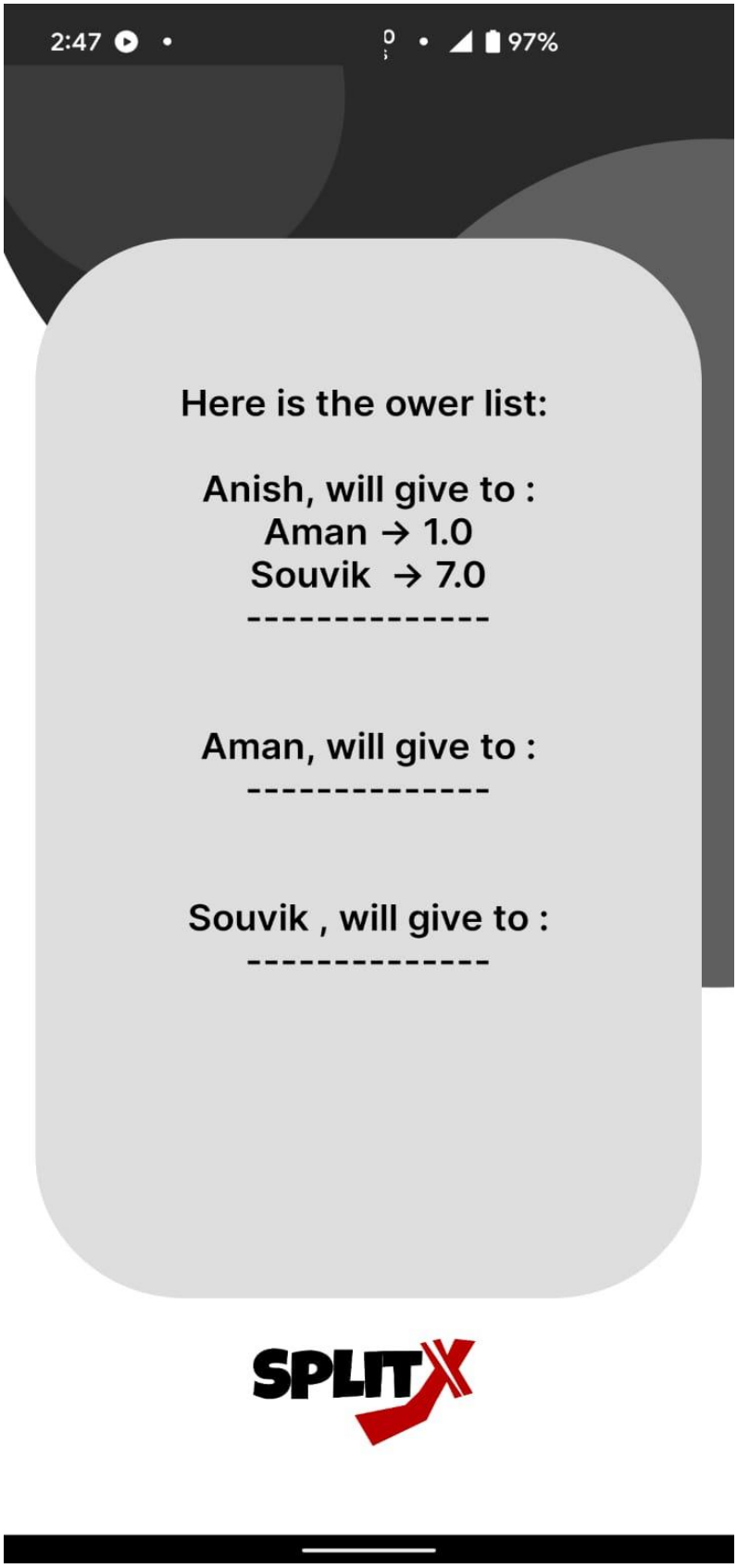
BILL NAME


3:18
67%
Sign Out
Bill Name
Please provide a suitable bill name for proper bill execution.
NEXT

DATA ENTRY


2:46
97%
Total Expenses:
Aman
Name
12
Amount
REMOVE
Souvik
Name
18
Amount
REMOVE
Anish
Name
3
Amount
REMOVE

OUTPUT

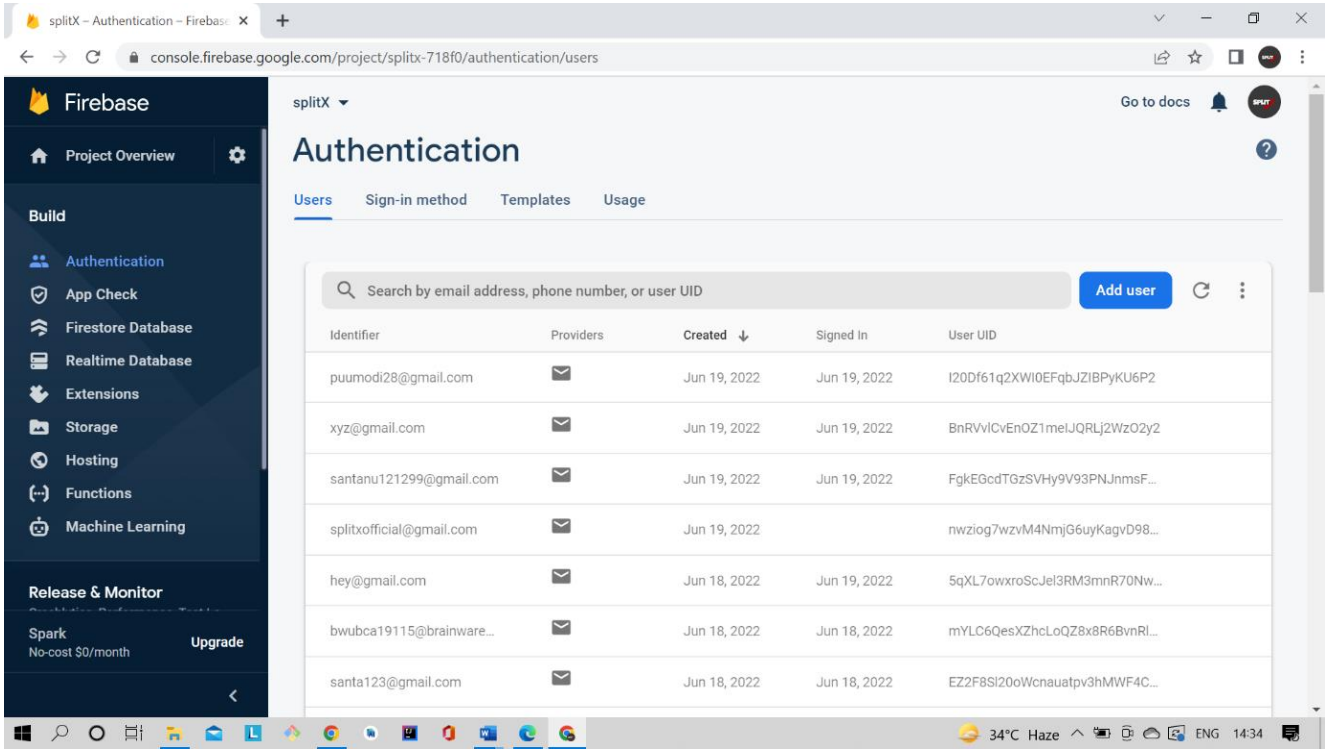

2:47
97%
Here is the ower list:
Anish, will give to :
Aman → 1.0
Souvik → 7.0

Aman, will give to :

Souvik , will give to :

SPLITX

Firebase


splitX – Authentication – Firebase
console.firebase.google.com/project/splitx-718f0/authentication/users
Firebase
Project Overview
Build
Authentication
App Check
Firestore Database
Realtime Database
Extensions
Storage
Hosting
Functions
Machine Learning
Release & Monitor
Spark
No-cost $0/month
Upgrade
splitX
Go to docs
Authentication
Users
Sign-in method
Templates
Usage
Search by email address, phone number, or user UID
Add user
Identifier
Providers
Created
Signed In
User UID
puumodi28@gmail.com
Jun 19, 2022
Jun 19, 2022
I20Df61q2XWI0EFqbJZIBPyKU6P2
xyz@gmail.com
Jun 19, 2022
Jun 19, 2022
BnRVvlCvEnOZ1meIJQRLj2WzO2y2
santanu121299@gmail.com
Jun 19, 2022
Jun 19, 2022
FgkEGcdTGzSVHy9V93PNJnmsF...
splitxofficial@gmail.com
Jun 19, 2022
nwziog7wzvM4NmjG6uyKagvD98...
hey@gmail.com
Jun 18, 2022
Jun 19, 2022
5qXL7owxroScJel3RM3mnR70Nw...
bwubca19115@brainware...
Jun 18, 2022
Jun 18, 2022
mYLC6QesXZhcLoQZ8x8R6BvnRl...
santa123@gmail.com
Jun 18, 2022
Jun 18, 2022
EZ2F8Sl20oWcnauatpv3hMWF4C...
34°C Haze
ENG
14:34

8. FUTURE SCOPE

There are some future ideas which we want to implement or rather we will include it in our Android application in the near future.

The ideas are:

- Payment Options
- Bill Sharing
- Phone Login/Google Login
- Storing Bills in Database for future use
- Contact Syncing
- Group creation & management
- User profile modification
- Monthly expense report

9. CONCLUSION

So as a conclusion we can say that as the world is developing and technology is getting upgraded day by day the change is very much needed. In context of our project named “SplitX” it can be said that this app is great for splitting bills.

This research has demonstrated the development of an Android Application which can be installed on Android devices with minimum android version 4.0 . The basic requirement of this application is to have Wi-Fi or cellular connection availability. Once the application is installed, it can be moved to SD - Card.

No major issues were found with the results of users' feedback. This feedback showed that everyone had found the application to be very much useful in calculating bills. Every participant agreed that the application can be easily installed and was quite easy to handle and operate. The UI is easy to use and simple. A few user interface issues were also faced and raised by the users which are expected to be delievered in the future.

So we can conclude that “SplitX” is a great app which will be very useful to people.

10. Bibliography

We the members of the project “SplitX” have used several technologies in this android application.

For this we got some references that are:

- Stack Overflow
- Google
- YouTube
- https://developer.android.com/docs