E-BANKING SYSTEM

Project Report

TCS-Remote Internship Program

Designed By:

Name: Mansi Yadav

Branch: Computer Science

College: Indira Gandhi Delhi Technical University for Women

CONTENTS

S.NO.	TITLE
1.	INTRODUCTION
2.	TECHNOLOGIES USED
3.	MODULE'S INFORMATION(SOURCE CODE AND UI)
4.	DATA FLOW DIAGRAM
4.	TEST CASES
5.	SCREENSHOTS
6.	DEMONSTRATION
7.	FUTURE ENHANCEMENTS

INTRODUCTION

The E-Banking System is an electronic payment system that enables customers of a bank or other financial institution to conduct a range of financial transactions through a mobile application. To access a financial institution's mobile banking facility, an individual would need to register with the institution for the service, and set up a password and other credentials for customer verification.

The customer can then, view his account information and transfer or retrieve money from other registered accounts.

FUNCTIONALITIES:

1. **PROFILE INFORMATION:** The customer can view his login credentials such as name, email-id, balance and password .

2. **TRANSFER:** The customer can transfer amount to some other registered account using his/her registered email-id.

3. **WITHDRAW:** The customer can withdraw amount from some other registered account using his/her registered email-id.

TECHNOLOGIES USED

ANDROID STUDIO : Android Studio is the official integrated development environment (IDE) for Android platform development.

SQLite Database: SQLite is a open-source SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation. SQLite supports all the relational database features.

In order to access this database, you don't need to establish any kind of connections for it like JDBC, ODBC etc.

MODULE'S INFORMATION

PROJECT TITLE: TCS

Package activity: This package contains all the activity classes.

1. **LoginActivity:** This class contains all the fields and methods required for login purposes.
2. **RegisterActivity:** This class contains all the fields and methods required for registration purposes.
3. **FirstActivity :** This class contains links to all other activity classes.
4. **ProfileActivity:** This class contains all the fields and methods required for viewing customer profile information.
5. **TransferActivity:** This class contains all the fields and methods required for transferring money from one registered account to another.
6. **WithdrawlActivity:** This class contains all the fields and methods required for withdrawing money from one registered account to another.

Package helper : This package contains all the helper classes.

1. **SQLiteHandler :** This class contains all the methods required for database implementation and manipulation.

Androidmanifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.mansi.tcs"
   android:versionCode="1"
   android:versionName="1.0" >

   <uses-sdk
      android:minSdkVersion="9"
      android:targetSdkVersion="21" />

   <uses-permission android:name="android.permission.INTERNET" />

   <application
      android:allowBackup="true"
      android:icon="@drawable/logo"
      android:label="@string/app_name"
      android:theme="@style/AppTheme" >
      <activity
         android:name="activity.LoginActivity"
         android:label="@string/app_name"
         android:launchMode="singleTop"
         android:windowSoftInputMode="adjustPan" >
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />

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

      <activity
         android:name="activity.FirstActivity"
         android:label="@string/app_name"
         android:launchMode="singleTop"
         android:windowSoftInputMode="adjustPan" />
      <activity
         android:name="activity.RegisterActivity"
         android:label="@string/app_name"
         android:launchMode="singleTop"
         android:windowSoftInputMode="adjustPan" />
      <activity
         android:name="activity.TransferActivity"
         android:label="@string/app_name"
         android:launchMode="singleTop"
         android:windowSoftInputMode="adjustPan" />
      <activity
         android:name="activity.WithdrawlActivity"
         android:label="@string/app_name"
         android:launchMode="singleTop"
         android:windowSoftInputMode="adjustPan" />
      <activity
         android:name="activity.ProfileActivity"
         android:label="@string/app_name"
         android:launchMode="singleTop"
         android:windowSoftInputMode="adjustPan" />

     </application
></manifest>
```

LoginActivity.java

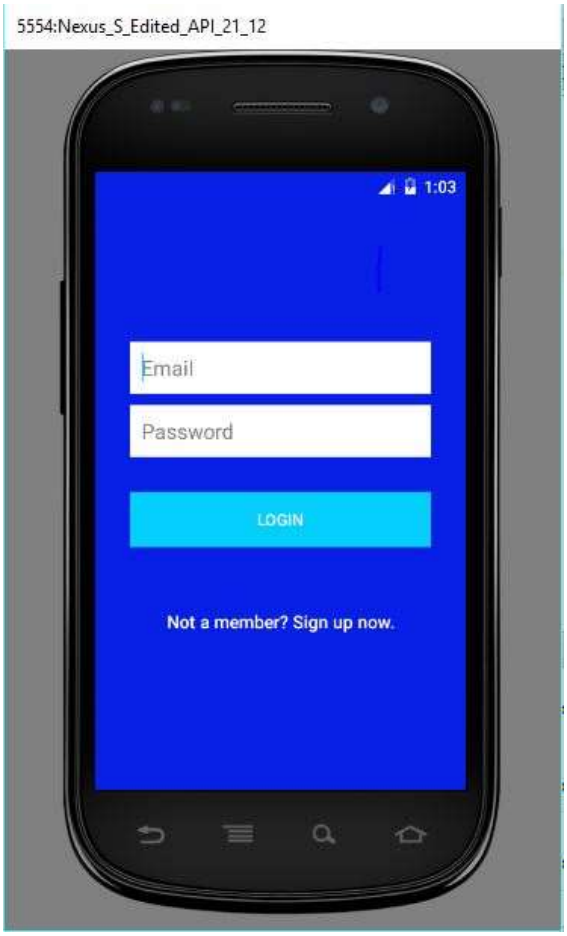
```
package activity;
import helper.*;
import com.example.mansi.tcs.R;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;


public class LoginActivity extends Activity {

    private Button btnLogin;
    private Button btnLinkToRegister;
    private EditText inputEmail;
    private EditText inputPassword;
    private ProgressDialog pDialog;
    private SQLiteHandler db;
    private String email = new String();
    String password = new String();

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

        inputEmail = (EditText) findViewById(R.id.email);
        inputPassword = (EditText) findViewById(R.id.password);
        btnLogin = (Button) findViewById(R.id.btnLogin);
        btnLinkToRegister = (Button) findViewById(R.id.btnLinkToRegisterScreen);

        // Progress dialog
        pDialog = new ProgressDialog(this);
        pDialog.setCancelable(false);

        // SQLite database handler
        db = new SQLiteHandler(getApplicationContext());



        btnLogin.setOnClickListener(new View.OnClickListener() {

            public void onClick(View view) {
                email = inputEmail.getText().toString().trim();
                password = inputPassword.getText().toString().trim();

               // Check for empty data in the form
               if (!email.isEmpty() && !password.isEmpty()) {
                   // login user
                   checkLogin(email, password);
               }
```

```
 else {
                // Prompt user to enter credentials
                Toast.makeText(getApplicationContext(),
                      "Please enter the credentials!", Toast.LENGTH_LONG)
                      .show();
             }
          }

       });

       // Link to Register Screen
       btnLinkToRegister.setOnClickListener(new View.OnClickListener() {

          public void onClick(View view) {
             Intent i = new Intent(getApplicationContext(),
                   RegisterActivity.class);
             startActivity(i);
             finish();
          }
       });

    }


    private void checkLogin(final String email, final String pwd) {


       pDialog.setMessage("Logging in ...");
       showDialog();
       boolean ans = db.login(email, pwd);

       if (ans == true) {
          hideDialog();
          Intent intent = new Intent(LoginActivity.this,
                FirstActivity.class);
          intent.putExtra("email", email);
          startActivity(intent);
          finish();
       }
else {
          hideDialog();
          Toast.makeText(getApplicationContext(), "INVALID CREDENTIALS", Toast.LENGTH_LONG).show();
       }
    }

    private void showDialog() {
       if (!pDialog.isShowing())
          pDialog.show();
    }


    private void hideDialog() {
       if (pDialog.isShowing())
          pDialog.dismiss();
    }

}
```

activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/bg_login"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="10dp" >

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:paddingLeft="20dp"
        android:paddingRight="20dp" >

        <EditText
            android:id="@+id/email"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="10dp"
            android:background="@color/white"
            android:hint="@string/hint_email"
            android:inputType="textEmailAddress"
            android:padding="10dp"
            android:singleLine="true"
            android:textColor="@color/input_login"
            android:textColorHint="@color/input_login_hint" />

        <EditText
            android:id="@+id/password"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="10dp"
            android:background="@color/white"
            android:hint="@string/hint_password"
            android:inputType="textPassword"
            android:padding="10dp"
            android:singleLine="true"
            android:textColor="@color/input_login"
            android:textColorHint="@color/input_login_hint" />

        <!-- Login Button -->

        <Button
            android:id="@+id/btnLogin"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dip"
            android:background="@color/btn_login_bg"
            android:text="@string/btn_login"
            android:textColor="@color/btn_login" />

        <!-- Link to Login Screen -->
```

```
 <Button
        android:id="@+id/btnLinkToRegisterScreen"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dip"
        android:background="@null"
        android:text="@string/btn_link_to_register"
        android:textAllCaps="false"
        android:textColor="@color/white"
        android:textSize="15dp" />
   </LinearLayout>

</LinearLayout>
```



RegisterActivity.java

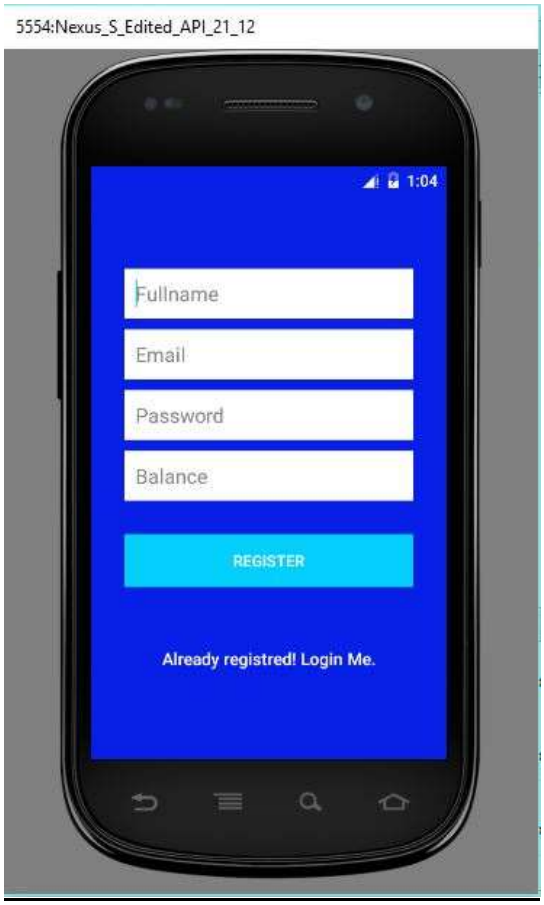
```
package activity;

import helper.*;
import com.example.mansi.tcs.*;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;



public class RegisterActivity extends Activity {

   private Button btnRegister;
    private Button btnLinkToLogin;
    private EditText inputFullName;
    private EditText inputEmail;
    private EditText inputPassword;
    private EditText inputBalance;
    private ProgressDialog pDialog;
    private SQLiteHandler db;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        inputFullName = (EditText) findViewById(R.id.name);
        inputEmail = (EditText) findViewById(R.id.email);
        inputPassword = (EditText) findViewById(R.id.password);
        inputBalance = (EditText) findViewById(R.id.balance);
        btnRegister = (Button) findViewById(R.id.btnRegister);
        btnLinkToLogin = (Button) findViewById(R.id.btnLinkToLoginScreen);

        // Progress dialog
        pDialog = new ProgressDialog(this);
        pDialog.setCancelable(false);



        // SQLite database handler
        db = new SQLiteHandler(getApplicationContext());


        btnRegister.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                String name = inputFullName.getText().toString().trim();
                String email = inputEmail.getText().toString().trim();
                String pwd = inputPassword.getText().toString().trim();
                String balance = inputBalance.getText().toString().trim();
```

```
if (!name.isEmpty() && !email.isEmpty() && !pwd.isEmpty() && !balance.isEmpty()) {
                registerUser(name, email,balance,pwd);
            } else {
                Toast.makeText(getApplicationContext(),
                        "Please enter your details!", Toast.LENGTH_LONG)
                        .show();
            }
        }
    });

    // Link to Login Screen
    btnLinkToLogin.setOnClickListener(new View.OnClickListener() {

        public void onClick(View view) {
            Intent i = new Intent(getApplicationContext(),
                    LoginActivity.class);
            startActivity(i);
            finish();
        }
    });
  }

  private void registerUser(final String name, final String email,
                    final String pwd, final String balance) {

    pDialog.setMessage("Registering ...");
    showDialog();
    boolean var= db.addUser(name, email, pwd, balance);
    if(var==true){
    hideDialog();
    Toast.makeText(getApplicationContext(), "User successfully registered. Try login now!",
Toast.LENGTH_LONG).show();

    Intent intent = new Intent(
            RegisterActivity.this,
            LoginActivity.class);
    startActivity(intent);
    finish();
  }

  else

  {
    hideDialog();
    Toast.makeText(getApplicationContext(), "UNABLE TO REGISTER", Toast.LENGTH_LONG).show();
  }
}

private void showDialog() {
    if (!pDialog.isShowing())
        pDialog.show();
  }

  private void hideDialog() {

    if (pDialog.isShowing())
        pDialog.dismiss();    }
}
```

activity_register.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/bg_register"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="10dp" >

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:paddingLeft="20dp"
        android:paddingRight="20dp" >

        <EditText
            android:id="@+id/name"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="10dp"
            android:background="@color/input_register_bg"
            android:hint="@string/hint_name"
            android:padding="10dp"
            android:singleLine="true"
            android:inputType="textCapWords"
            android:textColor="@color/input_register"
            android:textColorHint="@color/input_register_hint" />

        <EditText
            android:id="@+id/email"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="10dp"
            android:background="@color/input_register_bg"
            android:hint="@string/hint_email"
            android:inputType="textEmailAddress"
            android:padding="10dp"
            android:singleLine="true"
            android:textColor="@color/input_register"
            android:textColorHint="@color/input_register_hint" />

        <EditText
            android:id="@+id/password"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="10dp"
            android:background="@color/input_register_bg"
            android:hint="@string/hint_password"
            android:inputType="textPassword"
            android:padding="10dp"
            android:singleLine="true"
            android:textColor="@color/input_register"
            android:textColorHint="@color/input_register_hint" />
```

```
<EditText
        android:id="@+id/balance"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:background="@color/input_register_bg"
        android:hint="Balance"
        android:inputType="number"
        android:padding="10dp"
        android:singleLine="true"
        android:textColor="@color/input_register"
        android:textColorHint="@color/input_register_hint" />



    <Button
        android:id="@+id/btnRegister"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dip"
        android:background="@color/btn_login_bg"
        android:text="@string/btn_register"
        android:textColor="@color/white" />

    <!-- Link to Login Screen -->

    <Button
        android:id="@+id/btnLinkToLoginScreen"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dip"
        android:background="@null"
        android:text="@string/btn_link_to_login"
        android:textAllCaps="false"
        android:textColor="@color/white"
        android:textSize="15dp" />
  </LinearLayout>

</LinearLayout>
```


5554:Nexus_S_Edited_API_21_12
1:04
Fullname
Email
Password
Balance
REGISTER
Already registred! Login Me.

FirstActivity.java

```
package activity;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import com.example.mansi.tcs.R;


public class FirstActivity extends Activity{
   private Button btntoTransfer;
   private Button btntoWithdrawl;
   private Button btntoProfile;
   private Button Logoutfirst;


   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_first);

      btntoTransfer = (Button) findViewById(R.id.btntoTransfer);
      btntoWithdrawl = (Button) findViewById(R.id.btntoWithdrawl);
      btntoProfile = (Button) findViewById(R.id.btntoProfile);
      Logoutfirst = (Button) findViewById(R.id.Logoutfirst);

     btntoProfile.setOnClickListener(new View.OnClickListener() {

         public void onClick(View view) {
            Bundle extras = getIntent().getExtras();
            String data = extras.getString("email");
            Intent i = new Intent(getApplicationContext(),
                  ProfileActivity.class);
            i.putExtra("email", data);
            startActivity(i);
            finish();
         }
      });
      btntoTransfer.setOnClickListener(new View.OnClickListener() {

         public void onClick(View view) {
            Bundle extras = getIntent().getExtras();
            String data = extras.getString("email");
            Intent i = new Intent(getApplicationContext(),
                  TransferActivity.class);
            i.putExtra("email", data);
            startActivity(i);
            finish();
         }
      });
```

```
    btntoWithdrawl.setOnClickListener(new View.OnClickListener() {

            public void onClick(View view) {
                Bundle extras = getIntent().getExtras();
                String data = extras.getString("email");
                Intent i = new Intent(getApplicationContext(),
                        WithdrawlActivity.class);
                i.putExtra("email", data);
                startActivity(i);

                finish();
            }
        });

        Logoutfirst.setOnClickListener(new View.OnClickListener() {

            public void onClick(View view) {

                Intent i = new Intent(getApplicationContext(),
                        LoginActivity.class);

                startActivity(i);

                finish();
            }
        });



    }
}
```

activity_first.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/bg_register"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="10dp" >
    <Button
        android:id="@+id/btntoProfile"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dip"
        android:background="@color/btn_login_bg"
        android:text="@string/btn_profile"
        android:textColor="@color/white" />

    <Button
        android:id="@+id/btntoTransfer"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dip"
        android:background="@color/btn_login_bg"
        android:text="@string/btn_transfer"
        android:textColor="@color/white" />
    <Button
        android:id="@+id/btntoWithdrawl"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dip"
        android:background="@color/btn_login_bg"
        android:text="@string/btn_withdrawl"
        android:textColor="@color/white" />

    <Button
        android:id="@+id/Logoutfirst"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dip"
        android:background="@color/btn_logut_bg"
        android:text="Log Out"
        android:textAllCaps="false"
        android:textColor="@color/white"
        android:textSize="15dp" />

</LinearLayout>
```


5554:Nexus_S_Edited_API_21_12
1:09
PROFILE
TRANSFER
WITHDRAWL
Log Out

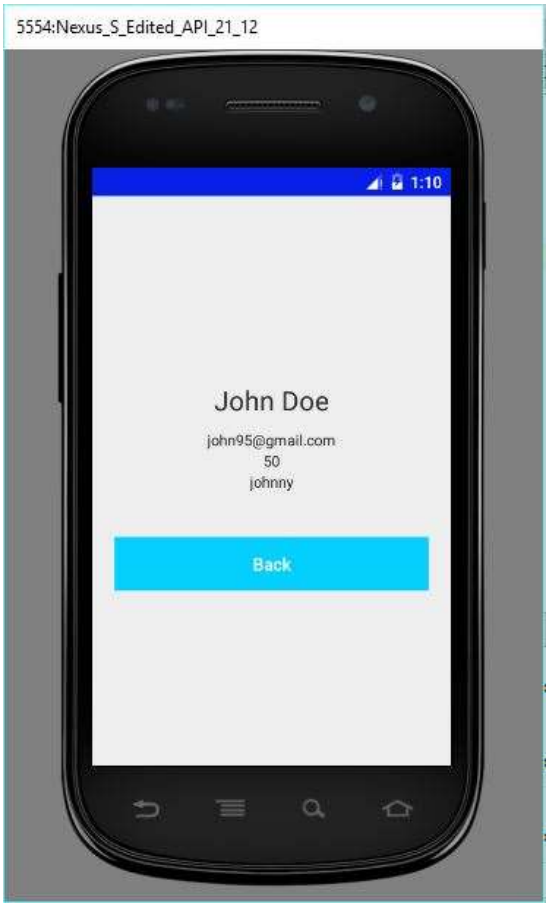
ProfileActivity.java

```
package activity;

import helper.*;
import java.util.HashMap;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import com.example.mansi.tcs.R;


public class ProfileActivity extends Activity {

    private TextView txtName;
    private TextView txtEmail;
    private TextView txtbalance;
    private TextView txtpwd;
    private Button backProfile;


    private SQLiteHandler db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile);
        final String myemail = getIntent().getExtras().getString("email");
        txtName = (TextView) findViewById(R.id.nameview);
        txtEmail = (TextView) findViewById(R.id.emailview);
        txtpwd = (TextView) findViewById(R.id.pwdview);
        txtbalance = (TextView) findViewById(R.id.balview);
        backProfile = (Button) findViewById(R.id.backProfile);

        // SqLite database handler
        db = new SQLiteHandler(getApplicationContext());


        // Fetching user details from sqlite
          HashMap<String, String> user = db.getUserDetails(myemail);

        String name = user.get("name");
        String email = user.get("email");
        String balance = user.get("balance");
        String pwd = user.get("pwd");

        // Displaying the user details on the screen
        txtName.setText(name);
        txtEmail.setText(email);
        txtbalance.setText(balance);
```

```
        txtpwd.setText(pwd);


    backProfile.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            Intent intent = new Intent(ProfileActivity.this, FirstActivity.class);
            intent.putExtra("email", myemail);
            startActivity(intent);

            finish();
        }
    });
  }


}
```

activity_profile.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="activity.ProfileActivity" >

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:gravity="center"
        android:orientation="vertical" >



        <TextView
            android:id="@+id/nameview"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="10dp"
            android:textColor="@color/lbl_name"
            android:textSize="24dp" />

        <TextView
            android:id="@+id/emailview"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="@color/lbl_name"
            android:textSize="13dp" />
        <TextView
            android:id="@+id/balview"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="@color/lbl_name"
            android:textSize="13dp" />
        <TextView
            android:id="@+id/pwdview"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="@color/lbl_name"
            android:textSize="13dp" />

        <Button
            android:id="@+id/backProfile"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="40dip"
            android:background="@color/btn_logut_bg"
            android:text="Back"
            android:textAllCaps="false"
            android:textColor="@color/white"
            android:textSize="15dp" />
    </LinearLayout>

</RelativeLayout>
```


5554:Nexus_S_Edited_API_21_12
1:10
John Doe
john95@gmail.com
50
johnny
Back

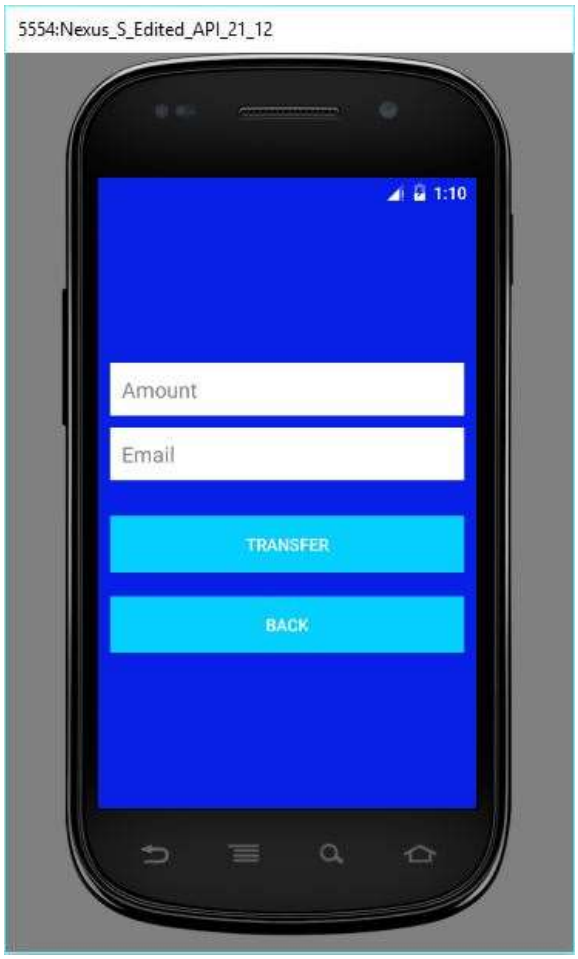
TransferActivity.java

```
package activity;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.example.mansi.tcs.*;
import helper.SQLiteHandler;

public class TransferActivity extends Activity {


    private Button btnTransfer;
    private Button backTransfer;
    private EditText inputEmail;
    private EditText inputAmount;
    private ProgressDialog pDialog;
    private SQLiteHandler db;


    @Override
    public void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_transfer);

       inputEmail = (EditText) findViewById(R.id.emailTransfer);
       inputAmount = (EditText) findViewById(R.id.amount);
       btnTransfer = (Button) findViewById(R.id.btnTransfer);
       backTransfer = (Button) findViewById(R.id.backTransfer);


       // Progress dialog
       pDialog = new ProgressDialog(this);
       pDialog.setCancelable(false);


       db = new SQLiteHandler(getApplicationContext());

       btnTransfer.setOnClickListener(new View.OnClickListener() {

          public void onClick(View view) {
             String emailTransfer = inputEmail.getText().toString().trim();
             String amount = inputAmount.getText().toString().trim();
             String emailfrom = getIntent().getExtras().getString("email");

             if (!emailTransfer.isEmpty() && !amount.isEmpty())

                transferamt(emailTransfer, amount, emailfrom);
             }

            else {

                Toast.makeText(getApplicationContext(),
```

```
                    "Please enter the details", Toast.LENGTH_LONG)
                    .show();
            }
        }

    });


    backTransfer.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            String emailfrom = getIntent().getExtras().getString("email");
            Intent intent = new Intent(TransferActivity.this, FirstActivity.class);
            intent.putExtra("email", emailfrom);
            startActivity(intent);

            finish();
        }
    });
}
private void transferamt(final String emailTransfer, final String amount, final String emailfrom) {


    pDialog.setMessage("Updating...");
    boolean ans = db.transfer(emailTransfer,amount,emailfrom);
    if(ans==true){
        hideDialog();
        Toast.makeText(getApplicationContext(), "Amount transferred successfully", Toast.LENGTH_LONG).show();
        Intent intent = new Intent(
                TransferActivity.this,
                FirstActivity.class);
        intent.putExtra("email", emailfrom);
        startActivity(intent);
        finish();
    }

    else

    {
        hideDialog();
        Toast.makeText(getApplicationContext(), "UNABLE TO TRANSFER", Toast.LENGTH_LONG).show();
    }

    showDialog();
    }

private void showDialog() {
    if (!pDialog.isShowing())
        pDialog.show();
}


private void hideDialog() {
    if (pDialog.isShowing())
        pDialog.dismiss();
}

}
```

activity_transfer.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/bg_register"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="10dp" >

<EditText
    android:id="@+id/amount"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:background="@color/input_register_bg"
    android:hint="Amount"
    android:inputType="number"
    android:padding="10dp"
    android:singleLine="true"
    android:textColor="@color/input_register"
    android:textColorHint="@color/input_register_hint" />
    <EditText
        android:id="@+id/emailTransfer"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:background="@color/input_register_bg"
        android:hint="Email"
        android:inputType="text"
        android:padding="10dp"
        android:singleLine="true"
        android:textColor="@color/input_register"
        android:textColorHint="@color/input_register_hint" />
    <Button
    android:id="@+id/btnTransfer"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dip"
    android:background="@color/btn_login_bg"
    android:text="@string/btn_transfer"
    android:textColor="@color/white" />
    <Button
        android:id="@+id/backTransfer"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dip"
        android:background="@color/btn_login_bg"
        android:text="Back"
        android:textColor="@color/white" />
</LinearLayout>
```


5554:Nexus_S_Edited_API_21_12
1:10
Amount
Email
TRANSFER
BACK

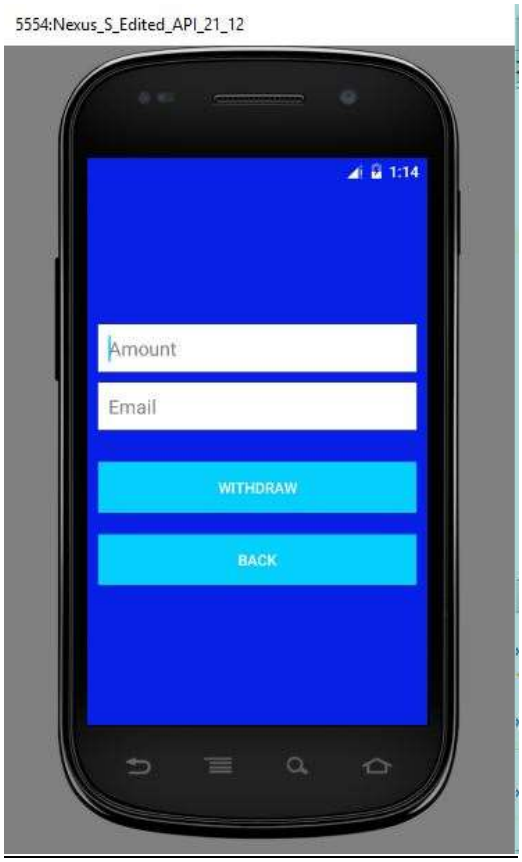
WithdrawlActivity.java

```
package activity;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.example.mansi.tcs.*;
import helper.SQLiteHandler;


public class WithdrawlActivity extends Activity {
    private Button btntoWithdrawl;
    private Button backWithdrawl;
    private EditText inputEmail;
    private EditText inputAmount;
    private ProgressDialog pDialog;
    private SQLiteHandler db;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_withdrawl);

        inputEmail = (EditText) findViewById(R.id.emailWithdrawl);
        inputAmount = (EditText) findViewById(R.id.amountWithdrawl);
        btntoWithdrawl = (Button) findViewById(R.id.btntoWithdrawl);
        backWithdrawl = (Button) findViewById(R.id.backWithdrawl);


        pDialog = new ProgressDialog(this);
        pDialog.setCancelable(false);

        // SQLite database handler
        db = new SQLiteHandler(getApplicationContext());

        btntoWithdrawl.setOnClickListener(new View.OnClickListener() {

            public void onClick(View view) {

                String amount = inputAmount.getText().toString().trim();
                String emailWithdrawl = inputEmail.getText().toString().trim();
                String emailfrom = getIntent().getExtras().getString("email");

if (!emailWithdrawl.isEmpty() && !amount.isEmpty()) {

                    withdrawamt(emailWithdrawl, amount, emailfrom);
            }
```

```
                else {

                    Toast.makeText(getApplicationContext(),
                            "Please fill the details", Toast.LENGTH_LONG)
                            .show();
                }
            }

        });
        backWithdrawl.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                String emailfrom = getIntent().getExtras().getString("email");
                Intent intent = new Intent(WithdrawlActivity.this, FirstActivity.class);
                intent.putExtra("email", emailfrom);
                startActivity(intent);

                finish();
            }
        });
    }

    private void withdrawamt(final String emailWithdrawl, final String amount, final String emailfrom) {
      pDialog.setMessage("Updating...");
        boolean ans = db.withdraw(emailWithdrawl,amount,emailfrom);
        if(ans==true){
            hideDialog();
            Toast.makeText(getApplicationContext(), "Amount Withdrawn successfully", Toast.LENGTH_LONG).show()


            Intent intent = new Intent(
                    WithdrawlActivity.this,
                    FirstActivity.class);
            intent.putExtra("email", emailfrom);
            startActivity(intent);
            finish();
        }

        else
        {
            hideDialog();
            Toast.makeText(getApplicationContext(), "UNABLE TO WITHDRAW", Toast.LENGTH_LONG).show();
        }

        showDialog();
    }

  private void showDialog() {
      if (!pDialog.isShowing())
          pDialog.show();
  }

  private void hideDialog() {
      if (pDialog.isShowing())
          pDialog.dismiss();
  }

}
```

activity_withdrawl.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/bg_register"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="10dp" >

    <EditText
        android:id="@+id/amountWithdrawl"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:background="@color/input_register_bg"
        android:hint="Amount"
        android:inputType= "number"
        android:padding="10dp"
        android:singleLine="true"
        android:textColor="@color/input_register"
        android:textColorHint="@color/input_register_hint" />
    <EditText
        android:id="@+id/emailWithdrawl"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:background="@color/input_register_bg"
        android:hint="Email"
        android:inputType="text"
        android:padding="10dp"
        android:singleLine="true"
        android:textColor="@color/input_register"
        android:textColorHint="@color/input_register_hint" />
    <Button
        android:id="@+id/btntoWithdrawl"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dip"
        android:background="@color/btn_login_bg"
        android:text="Withdraw"
        android:textColor="@color/white" />
   <Button
    android:id="@+id/backWithdrawl"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dip"
    android:background="@color/btn_login_bg"
    android:text="Back"
    android:textColor="@color/white" />
</LinearLayout>
```


5554:Nexus_S_Edited_API_21_12
1:14
Amount
Email
WITHDRAW
BACK

SQLiteHandler.java

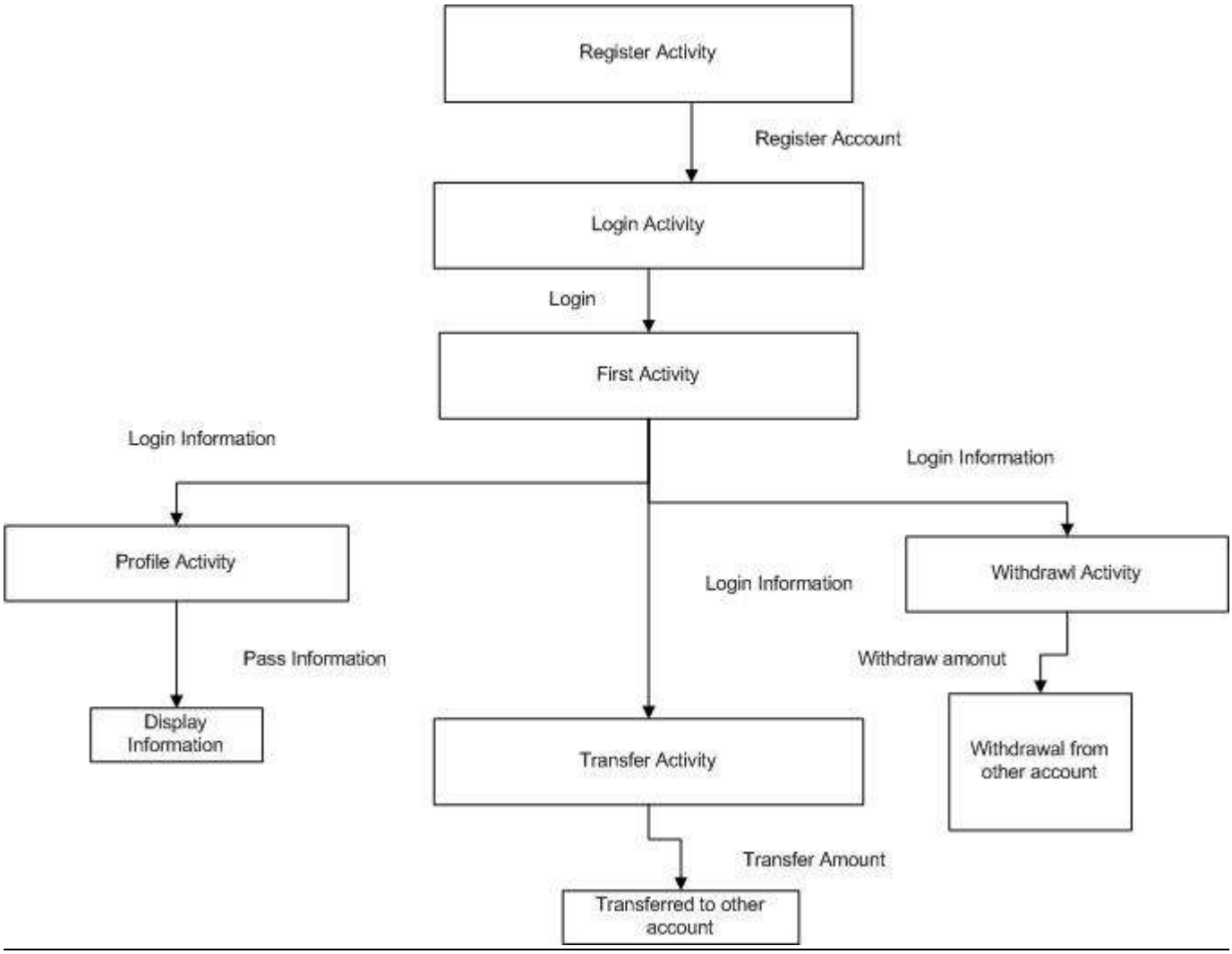
```
package helper;
import app.*;
import activity.*;
import helper.*;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

import java.util.HashMap;

public class SQLiteHandler extends SQLiteOpenHelper {

    private static final String TAG = SQLiteHandler.class.getSimpleName();

    // All Static variables
    // Database Version
    private static final int DATABASE_VERSION = 2;

    // Database Name
    private static final String DATABASE_NAME = "android_api";

    // Login table name
    private static final String TABLE_USER = "user1";

    // Login Table Columns names
    private static final String KEY_ID = "id";
    private static final String KEY_NAME = "name";
    private static final String KEY_EMAIL = "email";
    private static final String KEY_BALANCE = "balance";
    private static final String KEY_PASS = "pwd";


    public SQLiteHandler(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    // Creating Tables
    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_LOGIN_TABLE = "CREATE TABLE " + TABLE_USER + "("
                + KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," + KEY_NAME + " TEXT,"
                + KEY_EMAIL + " TEXT UNIQUE," + KEY_BALANCE + " INTEGER," + KEY_PASS + " TEXT UNIQUE );";
        db.execSQL(CREATE_LOGIN_TABLE);

        Log.d(TAG, "Database tables created");
    }
```

```
// Upgrading database
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // Drop older table if existed
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_USER);

        // Create tables again
        onCreate(db);
    }
    public boolean login(String email,String pwd){
        SQLiteDatabase db = this.getReadableDatabase();

        String selectQuery = "SELECT * FROM " + TABLE_USER + " WHERE " + KEY_EMAIL + "='" + email +
       "' AND " + KEY_PASS + "='" + pwd + "'" ;
        Log.d(TAG, selectQuery);

        Cursor cursor = db.rawQuery(selectQuery,null);

        cursor.moveToFirst();

        db.close();

      if(cursor.getCount()>0)
        {
            return true;}
        else
            return false;



    }
    public boolean withdraw(String emailWithdrawl ,String amount, String emailfrom) {
        SQLiteDatabase db = this.getReadableDatabase();
        String selectQuery = "SELECT "+ KEY_BALANCE +" FROM " + TABLE_USER + " WHERE "+ KEY_EMAIL
         + "='"+ emailWithdrawl+ "'";
        String selectQuery1 = "SELECT "+ KEY_BALANCE +" FROM " + TABLE_USER + " WHERE "+ KEY_EMAIL
         + "='"+ emailfrom+ "'";
        Cursor cursor = db.rawQuery(selectQuery, null);
        int current = 0;
        int to = 0;
        if (cursor.moveToFirst()) {
            current = Integer.parseInt(cursor.getString(0));

        }
        cursor.close();

        current = current - Integer.parseInt(amount);
        Cursor cursor1 = db.rawQuery(selectQuery1, null);
        if (cursor1.moveToFirst()) {
            to = Integer.parseInt(cursor1.getString(0));

        }
        to = to +Integer.parseInt(amount);
        cursor1.close();
        db.close();
```

```
try {
        SQLiteDatabase db1 = this.getWritableDatabase();
        String rawQuery = "UPDATE "+ TABLE_USER +" SET " + KEY_BALANCE + "='" + String.valueOf(current)
        + "'" + " WHERE " + KEY_EMAIL + "='" + emailWithdrawl + "'";

       db1.execSQL(rawQuery);
        String Query = "UPDATE "+ TABLE_USER +" SET " + KEY_BALANCE + "='" + String.valueOf(to) + "'"+
       "   WHERE " + KEY_EMAIL + "='" + emailfrom + "'";

        db1.execSQL(Query);
        return true;
    }
    catch (Exception ex) {
        ex.printStackTrace();
        return false;
    }
 }
 public boolean transfer(String emailTransfer ,String amount, String emailfrom) {
    SQLiteDatabase db = this.getReadableDatabase();
    String selectQuery = "SELECT "+ KEY_BALANCE +" FROM " + TABLE_USER + " WHERE "+ KEY_EMAIL
    + "='"+ emailTransfer+ "'";
    String selectQuery1 = "SELECT "+ KEY_BALANCE +" FROM " + TABLE_USER + " WHERE "+ KEY_EMAIL
    + "='"+ emailfrom+ "'";
    Cursor cursor = db.rawQuery(selectQuery, null);
    int current = 0;
    int frm = 0;
    if (cursor.moveToFirst()) {
        current = Integer.parseInt(cursor.getString(0));

    }
    cursor.close();

        current = current + Integer.parseInt(amount);
    Cursor cursor1 = db.rawQuery(selectQuery1, null);
    if (cursor1.moveToFirst()) {
        frm = Integer.parseInt(cursor1.getString(0));

    }
        frm = frm -Integer.parseInt(amount);
    cursor1.close();
    db.close();
    try {
        SQLiteDatabase db1 = this.getWritableDatabase();
        String rawQuery = "UPDATE "+ TABLE_USER +" SET " + KEY_BALANCE + "='" + String.valueOf(current)
        + "'" + " WHERE " + KEY_EMAIL + "='" + emailTransfer + "'";

        db1.execSQL(rawQuery);
        String Query = "UPDATE "+ TABLE_USER +" SET " + KEY_BALANCE + "='" + String.valueOf(frm) + "'"+
       " WHERE " + KEY_EMAIL + "='" + emailfrom + "'";

        db1.execSQL(Query);
        return true;

    }
    catch (Exception ex) {
        ex.printStackTrace();
        return false;
    }}
```

```
public boolean addUser(String name, String email, String balance, String pwd) {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();
        values.put(KEY_NAME, name);
        values.put(KEY_EMAIL, email);
        values.put(KEY_BALANCE, balance );
        values.put(KEY_PASS, pwd );
        // Inserting Row
        long id = db.insert(TABLE_USER, null, values);
        db.close(); // Closing database connection

        Log.d(TAG, "New user inserted into sqlite: " + id);
        if(id>0)
        return true;
        else
            return false;
    }

    public HashMap<String, String> getUserDetails(String myemail) {
        HashMap<String, String> user = new HashMap<String, String>();
        String selectQuery = "SELECT  * FROM " + TABLE_USER +" WHERE " + KEY_EMAIL + "='"+ myemail +"'";

        Log.d(TAG, selectQuery);
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery(selectQuery, null);
        // Move to first row
        cursor.moveToFirst();
        if (cursor.getCount() > 0) {
            user.put("uid",cursor.getString(0));
            user.put("name", cursor.getString(1));
            user.put("email", cursor.getString(2));
            user.put("balance", cursor.getString(3));
            user.put("pwd", cursor.getString(4));

        }
        cursor.close();
        db.close();
        Log.d(TAG, "Fetching user from Sqlite: " + user.toString());

        return user;
    }

    /**
     * Re crate database Delete all tables and create them again
     * */
    public void deleteUsers() {
        SQLiteDatabase db = this.getWritableDatabase();
        // Delete All Rows
        db.delete(TABLE_USER, null, null);
        db.close();

        Log.d(TAG, "Deleted all user info from sqlite");
    }

}
```

DATA FLOW DIAGRAM



TEST CASES

1. Registration:

Name	Email-id	Password	Balance	Output
John doe	John95@gmail.com	johnny	-	Please enter your details
John doe	John95@gmail.com	-	100	Please enter your details
John doe	-	johnny	100	Please enter your details
-	John95@gmail.com	johnny	100	Please enter your details
John doe	John95@gmail.com	johnny	100	User Registered successfully. Try login now

2. Login:

Email-id	Password	Output
John95@gmail.com	-	Please enter your details
-	johnny	Please enter your details
John95@gmail.com	johnny	Takes you to the main page

3.Transfer:

Amount	Email-id	Output
-	jane12@gmail.com	Please enter the details
10	-	Please enter the details
10	jane12@gmail.com	Amount transferred successfully

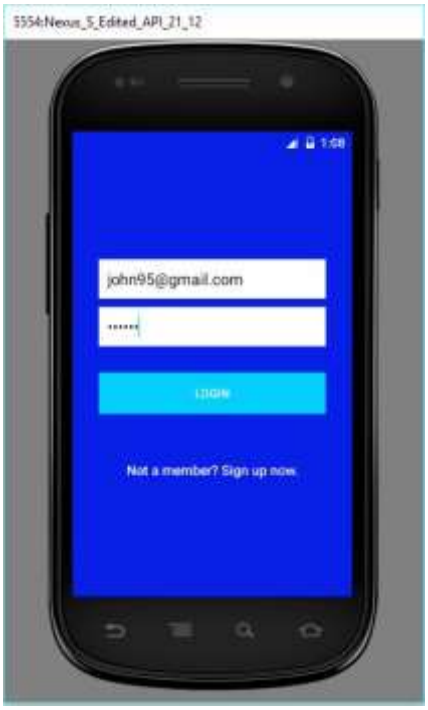
4. Withdraw:

Amount	Email-id	Output
-	jane12@gmail.com	Please enter the details
10	-	Please enter the details
10	jane12@gmail.com	Amount withdrawn successfully

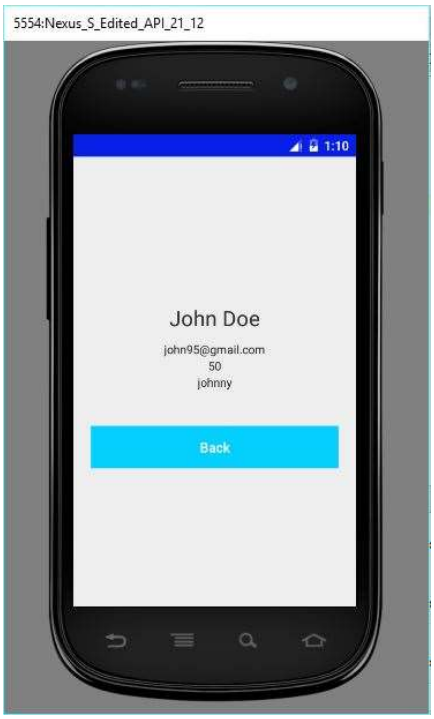
SCREENSHOTS

1. REGISTRATION

2. LOGIN

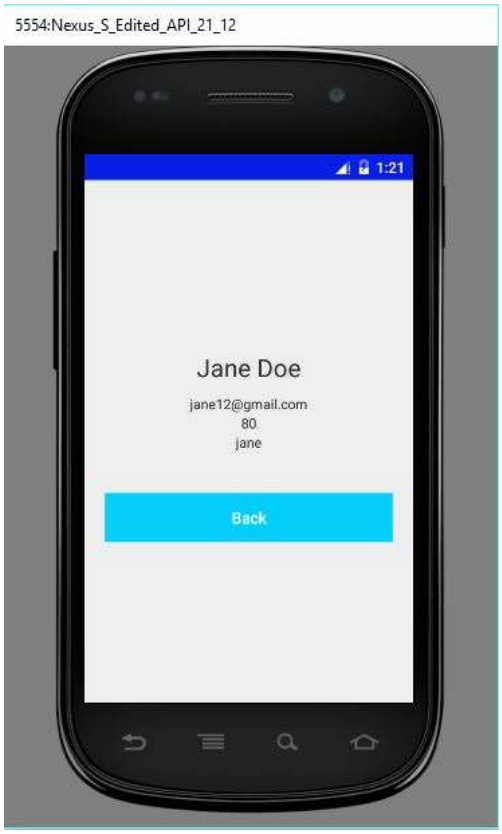


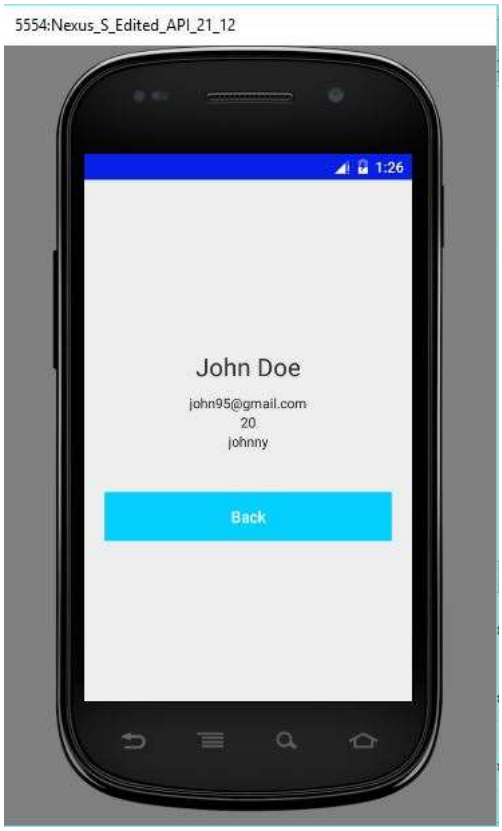
3. PROFILE



4. TRANSFER 10 RUPEES FROM JOHN'S ACCOUNT TO JANE'S ACCOUNT

5. WITHDRAW 20 RUPEES FROM JOHN'S ACCOUNT AND ADD TO JANE'S ACCOUNT





DEMONSTRATION

The application provides 3 basic functionalities:

1. View your account information.
2. Transfer amount from one registered account to another.
3. Withdraw amount from one registered account and add it to the other.

The application uses the SQLite database, provided with android studio and all android devices. The application performs all the operations from the internal database.

SQLite database can be updated, deleted and modified using MySQL queries.

SQLiteHandler class manages all the SQLite operations requested by all other classes.

1. Login:

   The class takes 2 input fields namely, email id and password. It then calls the login function of the SQLiteHandler Class that fetches the tuple from the database. If no such record exists it returns a false value otherwise, it returns True. The true value takes you the first activity page while, the false value forces you to stay on the login page. It also provides a link to the registration screen.

2. Registeration:

   The class takes 4 input fields namely, name, email-id, password and balance. It calls the adduser function of the SQLiteHandler class and stores the information in the database.
   If the record is stored successfully it returns a true value otherwise, a false value is returned.
   A true value takes you to the login page while, a false value forces you to stay on the registration page.

3. Main Page:

   This class does not take any input parameters. It provides 4 buttons.
   The profile button takes you to the account information page.
   The transfer button takes you to the transfer page.
   The withdraw button takes you to the withdrawal page.
   Lastly, the logout button takes you to the login page.

4. Profile:

   This class calls the getuserdetails method of the SQLiteHandler class the fetches the user information using the Logged in email id and displays the information on screen.
   Also, it has a back button that takes you back to the first activity screen.

5. Transfer:

   This class takes 2 parameters, the amount to be transferred and the account to which it is supposed to be Transferred. It calls the transfer method of the SQLiteHandler class and returns a true value on successful transfer.
   It returns a false value and forces you to stay on the same page if the transfer wasn't successful.
   The back button takes you to the first activity screen.

6. Withdrawal:

   This class takes 2 parameters, the amount to be withdrawn and the account from which it is supposed to be withdrawn. It calls the withdraw method of the SQLiteHandler class and returns a true value on successful withdrawal
   It returns a false value and forces you to stay on the same page if the transfer wasn't successful.
   The back button takes you to the first activity screen.

FUTURE ENHANCEMENTS

SECURITY FEATURES: To transfer money in between accounts there will be a pin number. The current version of the application doesn't provide this feature and any person with the other person's email id can withdraw money.

CENTRALISED DATABASE SYSTEM: The current version maintains a local database in the user's android device. The database vanishes as soon as the application is deleted. A server database is required for keeping the track of all the customers.