

Trapio
Theory - Object Oriented Software Engineering
Instructor Yella Mehroze
Assignment 1

Project Name		Trapio				
Section		B				
Submission Date		07/03/2021				
Project Group Members						
Sr.#	Reg. #	Student Name	CGPA	Email ID	Phone #	Signature
(i)	FA19-BSE-004	Areeb Shahzad	3.29	fa19-bse-004@cuilahore.edu.pk		
(ii)	FA19-BSE-014	Abdul Majid	3.20	fa19-bse-014@cuilahore.edu.pk		
(iii)	FA19-BSE-030	Hammad Naveed	3.20	fa19-bse-03@cuilahore.edu.pk		
(iv)	FA19-BSE-111	Abdul Raheem		fa19-bse-111@cuilahore.edu.pk		
(v)	SP19-BSE-136	Faraz Nawaz		sp19-bse-136@cuilahore.edu.pk		



Department of Computer Science
COMSATS University Islamabad, Lahore Campus
28 February 2021
Theory – Object oriented Software Engineering (CSC, 354)

Course Instructor – Yella Mehroze

Project– 1 Fa19-BSE (B)

Trapio

Introduction

Problem Statement:

The app lets you trade any product online providing flexibility and variety of products from a single seller.

Scope of the System:

The scope of the system is to buy, or sell any product with another one online.

It will let the users create an account and sell any item by uploading its details. It will also provide a category section to choose which products to buy or sell. The users will be able to choose between new and used products. The transaction will proceed after the agreement from both the parties.

Motivation for the system:

The idea behind the development of this system was to provide a platform for business-oriented people to sell a wider range of products than conventional stores and markets. This creates flexibility and ease of access for users and sellers at the same time. As a result, sellers will be able to sell whatever product they want providing greater chances for their overall growth in sales.

Proposed System

Functional Requirements:

FR01: Create Account

Req. No.	Functional Requirements
FR01-01	The system shall provide an interface to the user to fill signup credentials.
FR01-02	The system shall ask for confirmation of signup credentials.
FR01-03	The system shall send a verification email to the user's entered email address.
FR01-04	The system shall display account created successfully message to the user.
FR01-05	The system shall direct the user to login page.

FR02: Sell Product

Req. No.	Functional Requirements
FR02-01	The system shall display an interface to let the user to choose between buying and selling.
FR02-02	The system shall let the user to select an option between new and used products.
FR02-03	The system shall allow the user to choose category of the product.
FR02-04	The system shall display an interface to the user to enter details of the product.
FR02-05	The system shall let the seller to upload picture of the product.
FR02-06	The system shall ask the user for confirmation of posting the product
FR02-07	The system shall update the inventory of the seller.

FR02-08	The system shall notify the user of potential buyers.
FR02-09	The system shall provide an interface to let seller communicate with buyer.
FR02-10	The system shall ask the seller for confirmation of selling the product.
FR02-11	The system shall update the inventory of the seller.

FR03: Buy Product

Req. No.	Functional Requirements
FR03-01	The system shall display an interface to let the user to choose between buying and selling.
FR03-02	The system shall let the user to select an option between new and used products.
FR03-03	The system shall allow the user to choose category of the product.
FR03-04	The system shall display the related products to the buyer.
FR03-05	The system shall let the user select a product.
FR03-06	The system shall display the details of the product.
FR03-07	The system shall let the buyer contact the seller of the product.
FR03-08	The system shall display an interface for the buyer to select payment method.
FR03-09	The system shall ask for confirmation of purchase from buyer.
FR03-10	The system shall display estimated time for delivery of the product.
FR03-11	The system shall notify the buyer on arrival of the item.
FR03-12	The system shall provide an interface to the buyer to rate the seller.

FR04: Manage Accounts

Req. No.	Functional Requirements
FR04-01	The system shall let the admin register with admin account.
FR04-02	The system shall let the admin login to the system.
FR04-03	The system shall display buyer(s)/seller(s) to the admin.
FR04-04	The system shall let the admin search buyer/seller profile.
FR04-05	The system shall let the admin view buyer/seller profile.
FR04-06	The system shall let the admin block buyer/seller profile.
FR04-07	The system shall let the admin unblock buyer/seller profile.
FR04-08	The system shall let the admin log out of the system.

Non-functional Requirements:

Quality Requirements:

NFR01: Functionality

NFR01-01	The system shall let the user buy and sell any product online.
----------	--

NFR02: Usability

NFR02-01	The system shall be usable by non-technical staff.
----------	--

NFR03: Reliability

NFR03-01	The system shall send an email to the registered email address of the user in case of wrong password attempt.
----------	---

NFR04: Performance

NFR04-01	The system should have a response time of 3 seconds.
----------	--

NFR05: Supportability

NFR05-01	The system should be given maintenance update after every 3 weeks.
----------	--

Pseudo Requirements:

NFR06: Implementation Requirements

NFR06-01	Java shall be used as programming language for the system.
NFR06-02	There should be minimum 1GB RAM and 2GB internal memory.

NFR07: Interface Requirements

NFR07-01	The system should work with android 10, windows 10 and IOS 11.
----------	--

NFR08: Operation Requirements

NFR08-01	The system shall be manageable by any non-technical personnel.
----------	--

NFR09: Packaging Requirements

NFR09-01	The system shall be installed in only 1 day.
NFR09-02	The system shall be installable using internet connection.

NFR10: Legal Requirements

NFR10-01	The system shall be developed using licensed version of JetBrains IntelliJ IDE Community Edition.
NFR10-02	The system shall be developed using free to access modules available on GitHub.

Definition, Acronyms and Abbreviations

- **FR:** Functional Requirements.
- **NFR:** Non-functional Requirements.
- **IDE:** Integrated Development Environment

References

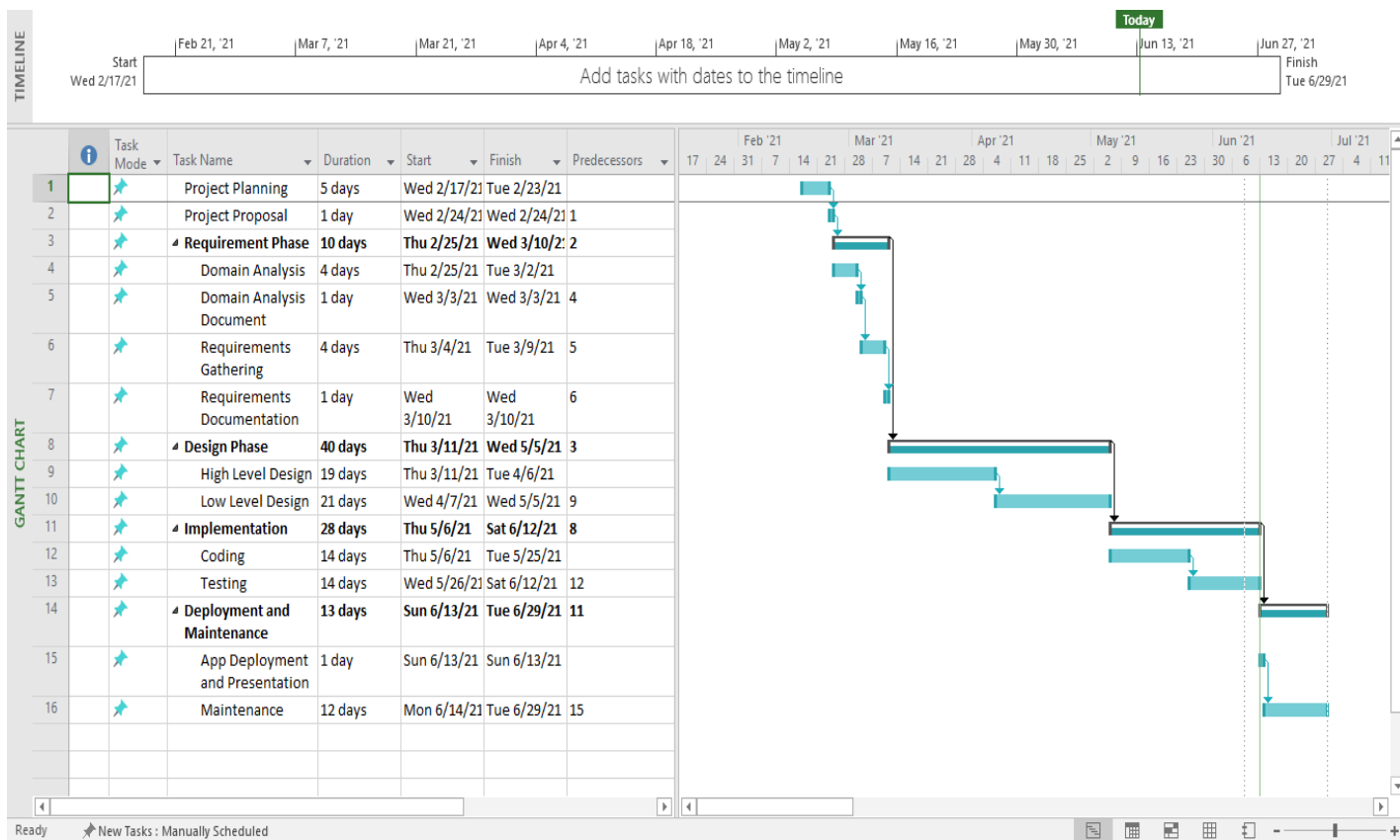
Documentation References:

- <https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>
- <https://www.projectmanager.com/training/write-scope-work>

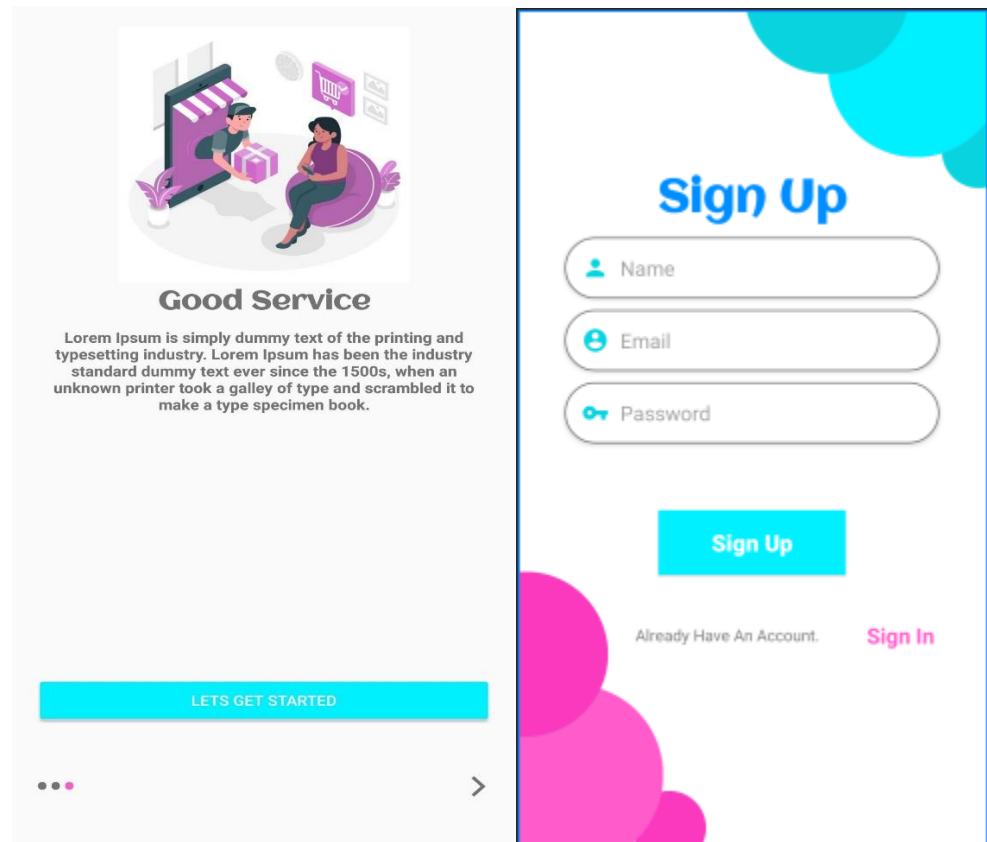
Glossary

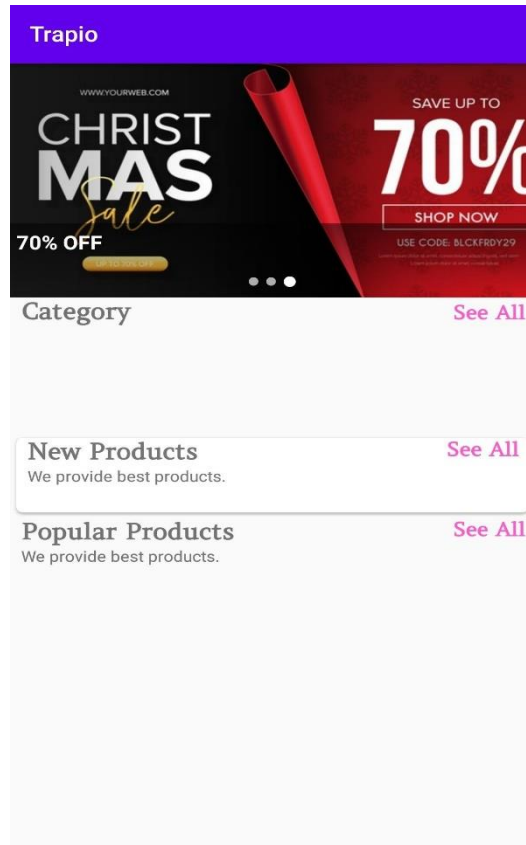
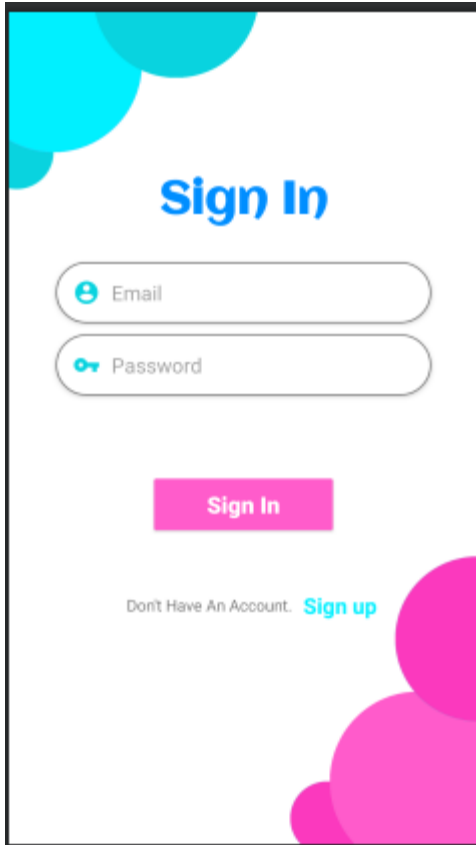
- **Functional Requirements:** The tasks and operations that are required by the system to perform.
- **Non-Functional Requirements:** The additional functionality that is not part of the system but enhances the overall experience of the system.
- **Integrated Development Environment:** The platform that lets you develop your software using coding language.

Gantt chart



User Interface:





Code:

Activities Files (Java Files):

LoginActivity:

```
package com.example.etrapiro.activities;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import com.example.etrapiro.R;

public class LoginActivity extends AppCompatActivity {

    EditText email, password;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        getSupportActionBar().hide();

        email = findViewById(R.id.email);
        password = findViewById(R.id.password);
    }

    public void signIn(View view) {

        String userEmail = email.getText().toString();
        String userPassword= password.getText().toString();

        if(TextUtils.isEmpty(userEmail)){
            Toast.makeText(this, "Enter Email Please",
                Toast.LENGTH_SHORT).show();
            return;
        }

        if(TextUtils.isEmpty(userPassword)){
            Toast.makeText(this, "Enter Password Please",
                Toast.LENGTH_SHORT).show();
            return;
        }

        if(userPassword.length()<6){
            Toast.makeText(this, "Password too Short. Enter minimum 6
characters", Toast.LENGTH_SHORT).show();
            return;
        }

        Toast.makeText(this, "Login Successfully", Toast.LENGTH_SHORT).show();
        startActivity(new Intent(LoginActivity.this, MainActivity.class));
    }

    public void singUp(View view) {
        startActivity(new Intent(LoginActivity.this,
RegistrationActivity.class));
    }
}

```

MainActivity:

```

package com.example.etrapiro.activities;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentTransaction;

import android.os.Bundle;

```



```
import com.example.etrapiro.R;
import com.example.etrapiro.fragments.HomeFragment;

public class MainActivity extends AppCompatActivity {

    Fragment homeFragment;

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

        homeFragment = new HomeFragment();
        loadFragment(homeFragment);
    }

    private void loadFragment(Fragment homeFragment) {
        FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();
        transaction.replace(R.id.home_container, homeFragment);
        transaction.commit();
    }
}
```

OnBoardingActivity:

```
package com.example.etrapiro.activities;

import androidx.appcompat.app.AppCompatActivity;
import androidx.viewpager.widget.ViewPager;

import android.content.Intent;
import android.os.Bundle;
import android.text.Html;
import android.view.View;
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.example.etrapiro.R;
import com.example.etrapiro.adapters.SliderAdapter;

public class OnBoardingActivity extends AppCompatActivity {

    ViewPager viewPager;
    LinearLayout dostoLayout;

    Button btn;
    SliderAdapter sliderAdapter;
```

```

    TextView[] dots;
    Animation animation;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //hide status bar
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_on_boarding);

        //hide tool bar
        getSupportActionBar().hide();

        viewPager = findViewById(R.id.slider);
        dostLayout = findViewById(R.id.dots);
        btn = findViewById(R.id.get_started_btn);
        addDots(0);

        viewPager.addOnPageChangeListener(changeListener);

        //call adapter
        sliderAdapter = new SliderAdapter(this);
        viewPager.setAdapter(sliderAdapter);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivities(new Intent[]{new Intent(OnBoardingActivity.this,
RegistrationActivity.class)});
                finish();
            }
        });
    }

    private void addDots(int position){
        dots =new TextView[3];
        dostLayout.removeAllViews();
        for(int i = 0; i < dots.length; i++){
            dots[i] = new TextView(this);
            dots[i].setText(Html.fromHtml("&#8226;"));
            dots[i].setTextSize(35);
            dostLayout.addView(dots[i]);
        }

        if(dots.length > 0){
            dots[position].setTextColor(getResources().getColor(R.color.pink));
        }
    }

    ViewPager.OnPageChangeListener changelister = new
ViewPager.OnPageChangeListener() {
        @Override
        public void onPageScrolled(int position, float positionOffset, int

```

```
positionOffsetPixels) {

    }

    @Override
    public void onPageSelected(int position) {
        addDots(position);

        if(position == 0){
            btn.setVisibility(View.INVISIBLE);
        }else if(position == 1){
            btn.setVisibility(View.INVISIBLE);
        }else{
            animation = AnimationUtils.LoadAnimation(OnBoardingActivity.this,
R.anim.slide_animation);
            btn.setAnimation(animation);
            btn.setVisibility(View.VISIBLE);
        }
    }

    @Override
    public void onPageScrollStateChanged(int state) {

    }

};
}
```

RegistrationActivity:

```
package com.example.etrapiro.activities;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import com.example.etrapiro.R;

public class RegistrationActivity extends AppCompatActivity {

    EditText name, email, password;
    //private FirebaseAuth auth
    SharedPreferences sharedPreferences;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_registration);

getSupportActionBar().hide();

name = findViewById(R.id.name);
email = findViewById(R.id.email);
password = findViewById(R.id.password);

sharedPreferences = getSharedPreferences("onBoardingScreen", MODE_PRIVATE);

boolean isFirstTime = sharedPreferences.getBoolean("firstTime", true);

if(isFirstTime){
    SharedPreferences.Editor editor= sharedPreferences.edit();
    editor.putBoolean("firstTime", false);
    editor.commit();

    Intent intent = new
Intent(RegistrationActivity.this, OnBoardingActivity.class);
    startActivity(intent);
    finish();
}

}

public void signup(View view){
    String userName = name.getText().toString();
    String userEmail = email.getText().toString();
    String userPassword= password.getText().toString();

    if(TextUtils.isEmpty(userName)){
        Toast.makeText(this, "Enter Name Please", Toast.LENGTH_SHORT).show();
        return;
    }

    if(TextUtils.isEmpty(userEmail)){
        Toast.makeText(this, "Enter Email Please", Toast.LENGTH_SHORT).show();
        return;
    }

    if(TextUtils.isEmpty(userPassword)){
        Toast.makeText(this, "Enter Password Please", Toast.LENGTH_SHORT).show();
        return;
    }

    if(userPassword.length()<6){
        Toast.makeText(this, "Password too Short. Enter minimum 6 characters",
Toast.LENGTH_SHORT).show();
        return;
    }

    Toast.makeText(this, "Successfully Register", Toast.LENGTH_SHORT).show();
    startActivity(new Intent(RegistrationActivity.this, MainActivity.class));

}
```

```

    public void signin(View view) {
        startActivity(new Intent(RegistrationActivity.this, LoginActivity.class));
    }
}

```

Design (Xml Files):

Login:

```

<?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"
    tools:context=".activities.LoginActivity"
    android:background="@drawable/bggg">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/aclonica"
        android:text="Sign In"
        android:textColor="@color/textHeading"
        android:textSize="40sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.2" />

    <EditText
        android:id="@+id/email"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="@drawable/edit_text_bg"
        android:drawableLeft="@drawable/ic_baseline_account_circle"
        android:drawablePadding="10dp"
        android:elevation="3dp"
        android:hint="Email"
        android:layout_marginTop="30dp"
        android:padding="15dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintWidth_percent=".8" />

    <EditText

```

```

    android:id="@+id/password"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:background="@drawable/edit_text_bg"
    android:layout_marginTop="10dp"
    android:drawableLeft="@drawable/ic_baseline_vpn_key_24"
    android:drawablePadding="10dp"
    android:elevation="3dp"
    android:hint="Password"
    android:inputType="numberPassword"
    android:padding="15dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/email"
    app:layout_constraintWidth_percent=".8" />

```

<Button

```

    android:backgroundTint="@color/pink"
    android:id="@+id/button"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:onClick="signIn"
    android:padding="15dp"
    android:text="Sign In"
    android:textAllCaps="false"
    android:textColor="@color/white"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/password"
    app:layout_constraintVertical_bias=".2"
    app:layout_constraintWidth_percent=".4" />

```

<TextView

```

    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Don't Have An Account."
    android:paddingLeft="10dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="@+id/password"
    app:layout_constraintHorizontal_bias=".3"
    app:layout_constraintStart_toStartOf="@+id/password"
    app:layout_constraintTop_toBottomOf="@+id/button"
    app:layout_constraintVertical_bias=".2" />

```

<TextView

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="singUp"
    android:text="Sign up"
    android:textColor="@color/blue"
    android:textSize="19sp"

```

```

        android:textStyle="bold"
        android:backgroundTint="@color/pink"
        app:layout_constraintHorizontal_bias="0.1"
        app:layout_constraintBottom_toBottomOf="@+id/textView2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/textView2"
        app:layout_constraintTop_toTopOf="@+id/textView2" />

```

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

Registration:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".activities.RegistrationActivity"
    android:background="@drawable/designlogin2"
    >

```

```

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/aclonica"
        android:text="Sign Up"
        android:textColor="@color/textHeading"
        android:textSize="40sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.2" />

```

```

    <EditText
        android:id="@+id/name"
        android:background="@drawable/edit_text_bg"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:hint="Name"
        android:padding="15dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintWidth_percent=".8"
        android:drawableLeft="@drawable/ic_baseline_person_24"

```

```
        android:drawablePadding="10dp"
        android:elevation="3dp"
    />

    <EditText
        android:id="@+id/email"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:background="@drawable/edit_text_bg"
        android:drawableLeft="@drawable/ic_baseline_account_circle"
        android:drawablePadding="10dp"
        android:elevation="3dp"
        android:hint="Email"
        android:padding="15dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/name"
        app:layout_constraintWidth_percent=".8" />

    <EditText
        android:id="@+id/password"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="@drawable/edit_text_bg"
        android:layout_marginTop="10dp"
        android:drawableLeft="@drawable/ic_baseline_vpn_key_24"
        android:drawablePadding="10dp"
        android:elevation="3dp"
        android:hint="Password"
        android:inputType="numberPassword"
        android:padding="15dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/email"
        app:layout_constraintWidth_percent=".8" />

    <Button
        android:id="@+id/button"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="@color/blue"
        android:onClick="signup"
        android:padding="15dp"
        android:text="Sign Up"
        android:textAllCaps="false"
        android:textColor="@color/white"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/password"
        app:layout_constraintVertical_bias=".2"
        app:layout_constraintWidth_percent=".4" />
```



```

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Already Have An Account."
    android:paddingLeft="30dp"
    android:paddingRight="30dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="@+id/password"
    app:layout_constraintHorizontal_bias=".3"
    app:layout_constraintStart_toStartOf="@+id/password"
    app:layout_constraintTop_toBottomOf="@+id/button"
    app:layout_constraintVertical_bias=".2" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="signin"
    android:text="Sign In"
    android:textColor="@color/pink"
    android:textSize="19sp"
    android:textStyle="bold"
    app:layout_constraintHorizontal_bias=".2"
    app:layout_constraintBottom_toBottomOf="@+id/textView2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/textView2"
    app:layout_constraintTop_toTopOf="@+id/textView2" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Slider:

```

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

    <ImageView
        android:id="@+id/slider_img"
        android:layout_width="0dp"
        android:layout_height="250dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/onboardscreen1" />

    <TextView
        android:id="@+id/heading"
        android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:fontFamily="@font/aclonica"
        android:text="@string/first_slide"
        android:textAlignment="center"
        android:textSize="25sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/slider_img"
        android:gravity="center_horizontal" />

<TextView
    android:id="@+id/description"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:text="@string/description"
    android:textAlignment="center"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/heading"
    android:gravity="center_horizontal" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Summary

The coding tool used to build the project is **Android Studio** and the coding language used is Java and XML. The app developed by us is an e-commerce app that lets you buy and sell products. The user can either register as a buyer or a seller. There is a separate account for admin registration. The modules of our app include **Registration, Login, Buyer, Seller and Admin**. The functionality for all these modules had to be implemented. The issues that were faced by us was the use of Gradle in Android Studio. This is the first and basic version of our app. Further improvements require changes in app GUI and implementation of newer modules into the system.
