A PROJECT REPORT

on

"SECOND HAND MART- SHM"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
INFORMATION TECHNOLOGY

BY

SPARSH SINHA	1606067
SIDDHARTH AGRAWAL	1606306
RISHI DHACHOLIA	1606373

UNDER THE GUIDANCE OF
PROF. DR. SUNEETA MOHANTY

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

April 2020

A PROJECT REPORT

on

"SECOND HAND MART- SHM"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
INFORMATION TECHNOLOGY

BY

SPARSH SINHA	1606067
SIDDHARTH AGRAWAL	1606306
RISHI DHACHOLIA	1606373

UNDER THE GUIDANCE OF
PROF. DR. SUNEETA MOHANTY

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

April 2020

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"SECOND HAND MART- SHM"

submitted by

SPARSH SINHA	1606067
SIDDHARTH AGRAWAL	1606306
RISHI DHACHOLIA	1606373

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci-ence & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2019-2020, under our guidance.

Date: / /

(Prof. Dr. Suneeta Mohanty)
Project Guide

Acknowledgements

We are profoundly grateful to Prof. Dr. Suneeta Mohanty for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. On the very outset of this report, we would like to extend our sincere and heartfelt obligation towards all the personages who have helped us in this endeavour. Without their active guidance, help, cooperation and encouragement, we would have not made headway to this project. We extend our sincere gratitude to our college Kalinga Institute of Industrial Technology for giving us this opportunity.

SPARSH SINHA
SIDDHARTH AGRAWAL
RISHI DHACHOLIA

ABSTRACT

In this project, our goal was to develop an Android Application that can be used for buying “second-hand” products online. We got the opportunity to implement the things we learned in our theoretical and lab sessions along with the concepts we got familiar with in our trainings and internships.

We have used Realtime Database for saving user and product details, as well as the transactions that a user has made. Along with this, we have added E-mail authentication to keep the application secure. We have used Android Studio 3.6.1 for creating out application. We used our knowledge of Android development, JAVA, Database Management and its various applications used in android development.

Keywords: Android application, “second-hand” products, Realtime Database, E-mail authentication, Android Studio 3.6.1, Android development, JAVA, Database Management.

Contents

List of Figures

Chapter 1

Introduction

1.1 INTRODUCTION TO THE PROJECT

This project is based on an Android Application that can be used for buying second-hand or used products online in a completely secure way. We've used Android Studio for creating our application. We used our knowledge of Android development, JAVA, Database Management and its various applications used in android development.

Fig 1.1: APPLICATION ICON LOGO

We have also used E-mail authentication to keep the user's details secure at all time. All the user's details like his Name, e-mail id, password along with his transactions are all stored using Realtime Database.

1.1.1 INTRODUCTION TO ANDROID

Android is a mobile operating system developed by Google. based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.[1]

1.1.2 INTRODUCTION TO ANDROID STUDIO

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It uses a flexible Gradle-based build system. We have used version 3.6.1 of Android Studio in our project.[2]

Chapter 2

Literature Survey

2.1 LITERATURE SURVEY

This section provides the technical context of the project. It gives the literature review of the work that has been done on applications whose main purpose was buying products online. There have been projects created for shopping purposes.

a. Some of those projects received world-wide recognition while some of them just came and went by without many people noticing them. The 2-best example in this line of applications are:

- Flipkart: The brand that started it all in India. Launched in Bengaluru in 2007[3], Flipkart quickly gained recognition and a major success. People realized how easy it can be to just sit at home and shop. Even though its usage has dropped with the introduction of new brands, without Flipkart, there would be no online shopping in India.

- Amazon: Although Flipkart started it, Amazon has taken over the throne of the most used and reliable shopping application among users. Almost everything available in the world can be bought on Amazon now.

b. While the above 2 applications were meant to purchase, we were introduced to applications like OLX () and Quikr (), which were to sell and purchase used products at a lower rate. Both had a brief period of success, but then they faded. Quikr later became so unsuccessful and they had to change it functionality completely.

c. While these applications could either be used to buy original products or to sell used product, ebay was the application where people could not only buy original and used product but also sell them.
The application found success all around the world, but due to the many frauds committed on this app, its usage decreased, especially in India.[4]

d. These are applications where almost all kinds of products are available. There are also applications where only a particular type of products are available. One such app is Myntra ( ) which focuses on only fashion products.

2.2 OUR PROJECT

What we have done with our application is that we are selling the Fashion products by ourselves. Unlike Amazon, Flipkart and Myntra, we are selling Second Hand but unused Fashion products at a cheaper rate than the standard rate. Here we sell fashion products that were bought but no use was found for them.
We can also sell products that other people sell to us but the major difference being that those people won't have to get involved in the functioning of this application. If anything happens, they'll not be held responsible. There will be only 2 sides- the customer and the developers, so that there is no fraud. This would clear the issues faced by ebay, OLX and quikr. Along with all this, the layout of our application is user-friendly. It is not complex for them to understand the functioning of our application.
Our project will also be a secure app which can be opened only by using a unique and password protected e-mail id.

Chapter 3

Software Requirements Specification

3.1 SOFTWARE REQUIREMENT

Our project is an Android application which can run on all android devices with the minimum Software Development Kit (SDK) being **23 API: Android 6.0 (Marshmallow)**. It was developed using **Windows 10 version 1909** Operating System.

3.1.1 SOFTWARE USED IN DEVELOPMENT

- Android Studio 3.6.1: The major part of development was done on Android Studio 3.6.1. The source codes and all the layouts and icons were designed using this software.
- Firebase: Firebase is a tool provided by Google. It is a tool that can found attached with the Android Studio. It was used for adding authentication to the application.
- XAMPP 7.4.3: XAMPP was used to access MySQL on the system. It helped us in accessing the database containing all the customer details as well as the customer details.

3.1.2 SOFTWARE USED IN TESTING

- Android Studio 3.6.1: Android studio was not only used for the development part, but also for the testing purposes. The software provides various emulators for running the applications. We used the Pixel 3 API 29 Emulator with *Android version 8.0 (Oreo)*.
- One UI version 1.0: Samsung Galaxy S8 was used for this test with *Android 9 (Pie)*.
- One UI version 2.1: Samsung Galaxy S20+ was used with *Android 10*.

Chapter 4

Requirement Analysis

4.1 SOFTWARE REQUIRED

As discussed in the previous chapter, there were different software programmes used in this project for development and testing purposes. The following programmes were required for this project to be complete:

- Android Studio: The most important requirement was Android Studio. It is the main software where the functioning and the design of the project was developed. It provided 2 languages to work on- JAVA and Kotlin. We have developed the application using JAVA.
  The version 3.6.1 of Android Studio was used in this project.

- Firebase: This project would not have been possible without the Google provided tool known as Firebase. Since we have used the term 'Firebase' a lot, let's understand what firebase actually is. It is a tool provided by Google. This tool is a part of Android Studio, i.e. it is embedded in Android Studio. It provides the user with multiple features like realtime database, authentication, cloud computing, storage, etc.[5] We used the authentication feature of this tool to add E-mail authentication to this application.

- XAMPP: The XAMPP suite of Web development tools makes it easy to run PHP scripts locally on your computer. Manual installation of a Web server and PHP requires in-depth configuration knowledge, but installing XAMPP on Windows only requires running an installer package. This package installs not only a Web server and PHP but also MySQL, FileZilla, Mercury, Perl and Tomcat. These applications allow testing of full websites on your own desktop without the need to upload everything to an online Web server.[6]

  Instead of using Firebase for creating the Database, we used SQL database. Using XAMPP, we were able to access that data. It helps the users to get the data on their local servers.

  The version 7.4.3 was used for this project.

- Android Studio's Emulator: It is another tool provided by the Android Studio. It is basically a Virtual Android Device which can be used to run the developed application for testing purposes. The developer is allowed to select the Virtual Device and the Android version it provides. The user has the access to almost all Virtual Android mobile phones to Virtual Tablets running the Android OS.

  Since the minimum SDK of our application was 23 API: Android 6.0 (Marshmallow), we used Pixel 3 API 29 with Android 8.0 (Pie).

4.2 HARDWARE REQUIRED

Following Hardware devices were used in this project:

- HP Pavilion x360 Convertible 14: This laptop with Windows 10 version 1909 was used for the development of this application. The 1 GB graphic card and 16.00 GB RAM was very big help in testing this application on the emulator smoothly.

- Samsung Galaxy S8: This mobile device was used to test run the application step by step. We used it for testing the multiple functions of our application one at a time and then all of them combined as well. This device runs on One UI version 1.0, Android 9 (Pie).

- Samsung Galaxy S20+: Another device was used to test run this application to check the working of the application on different devices using different android versions. This device runs on One UI version 2.1, Android 10.

Chapter 5

System Design

5.1 FLOW CHART

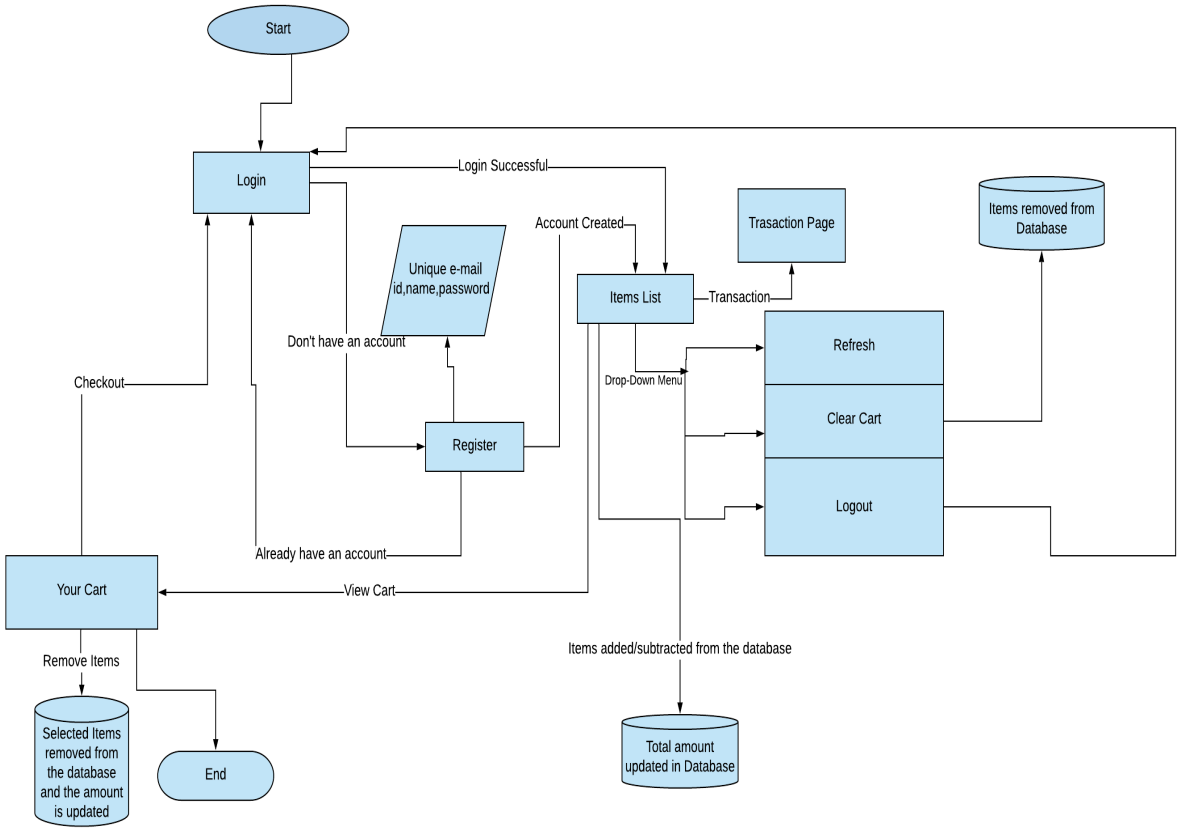

Fig 5.1: SYSTEM DESIGN FLOW CHART

5.2 EXPLANATION

- Enter the login details (e-mail id, password) in the ‘Login’ page to open the Items list. If the e-mail id and password are correct, only then the user will get to open the ‘Items list’.
- If the user hasn’t registered, he/she can do so by going to the ‘Register’ page. There the user details (name, e-mail id, password) are stored in the database.
- In the ‘Items list’, there will be multiple options:
  - Any addition or subtraction of items would update the cart and amount to be paid in the database.
  - The user can also go and view all the transactions made from his/her account previously.
  - The user will also get the option to refresh, or to ‘Clear the Cart’, which would delete all the items from the database. He/she can also logout from there, which would take him/her to the ‘Login’ page.
- In the ‘Your Cart’ page, the user will have the option to remove one or more of his/her items from the cart, which would remove the selected items from the database and update the amount accordingly.
- The ‘Checkout’ button would take the user back to the ‘Login’ page.

Chapter 6

System Testing

Various tests were performed manually during the development of this project.

6.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behaviour	Expected Result
T01	User Login Test-I	The correct E-mail id and password entered	Login successful and the Items list is opened.	A successful login and the Items list expected to open.
T02	User Login Test-II	Invalid E-mail address entered	Login unsuccessful.	Unsuccessful login.
T03	User Login Test-III	A valid E-mail address entered with a wrong password	Login unsuccessful.	Unsuccessful login.
T04	Registration Test-I	A valid and unused E-mail id is entered	Account created.	Creation of new Account.
T05	Registration Test-II	An already Active E-mail id is entered	Registration failed.	Failure in new account creation.
T06	Items Test	Addition/Subtraction of items	Cart and Amount updated.	Changes in cart and amount.
T07	Clear Cart Test	Clear Cart option was selected	All items were removed from cart.	Removal of items from the cart.
T08	Logout Test	Logout option was selected	User taken back to the Login page.	Login page was expected to open.
T09	Removal Test	Press Remove Items	All items removed.	Removal of items.
T10	Checkout Test	The Checkout button was pressed.	User taken back to the Login page.	Login page was expected to open.

Table 6.1: SYSTEM TESTING AND RESULTS

Chapter 7

Project Planning

7.1 REQUIREMENTS

For an e-commerce retail store application, it was very important to take note of the requirements below:

1. Secure purchase and secure application login
2. Robust and easy to use
3. Fast
4. Scalable
5. Customized experience

7.2 TECHNICAL ISSUES

While developing this project, there were certain technical issues that had to be looked into. They were as follows:

1. Security in all its form
2. Transaction & replications
3. Speed of Development and changes

7.3 BROKERAGE

For an e-commerce retail store application to be successful, it was very important to plan how the brokerage would work. It was planned as follows:

1. Bring buyers and sellers together that can be B2B or B2C
2. Usually, charge a minimal fee for the transaction

Chapter 8

Implementation

The whole project was developed using the Android Studio. We used JAVA programming language to generate the source codes for this project along with multiple XML codes for designing the layout of our application. Multiple JAVA classes and XML files were created for the various pages of this application.

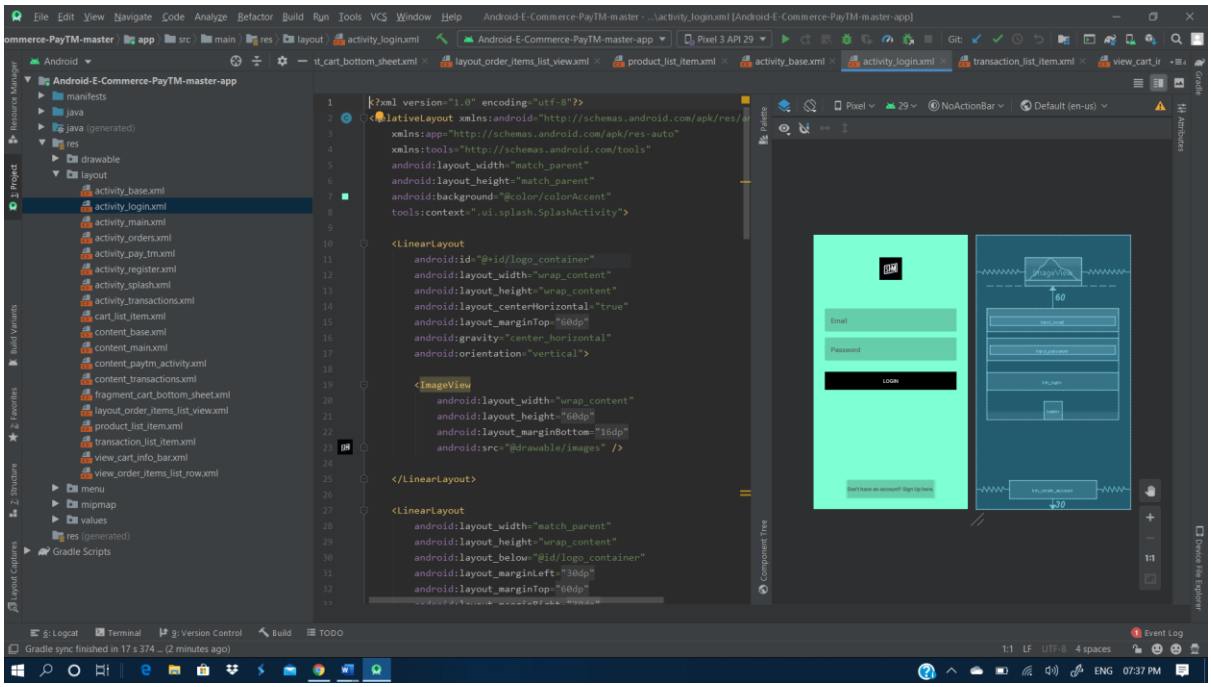

Fig 8.1: LOGIN ACTIVITY XML SCREENSHOT

All these pages were connected using ‘Intent’.

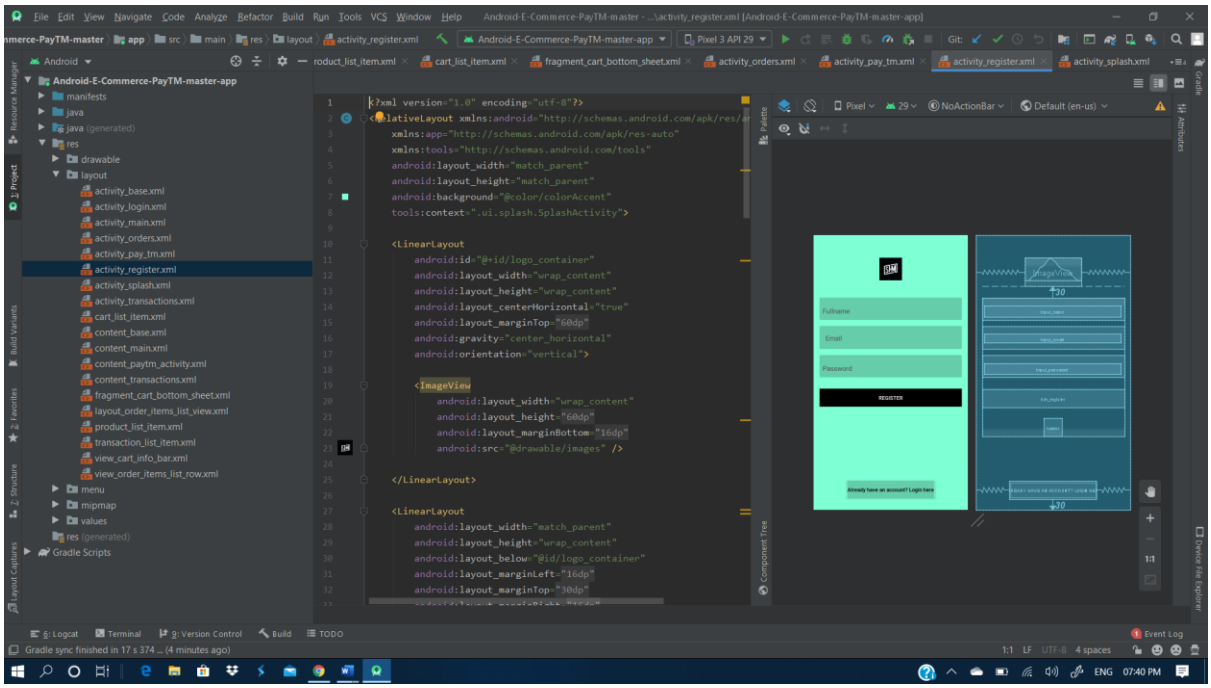

Fig 8.2: REGISTRATION PAGE XML SCREENSHOT

To create the items list, we added the real images taken from other e-commerce applications like Flipkart and Amazon. By adding the '+' and '-' buttons to the item layout, we were able to add the feature of selected one or more of the same products as well.

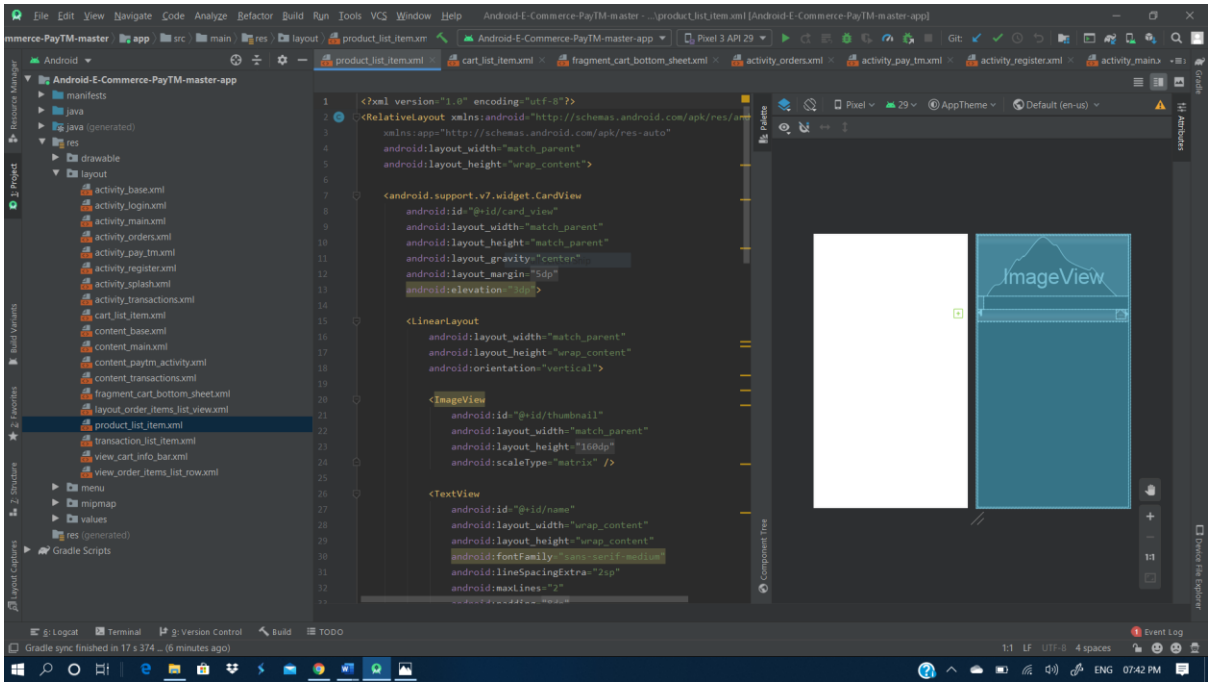

Fig 8.3: ITEMS LIST LAYOUT XML SCREENSHOT

In the main screen, a dropdown menu was added which contained the 3 options of:

- Refresh
- Clear Cart
- Logout

The functioning of these 3 features have been thoroughly discussed in System Design (Chapter 5), Section 5.2.
A transaction list was added to this application, which would display all the previous transactions made by the user.

This takes us to the final step of the project- The Cart.
The user can go and view the cart by using the ‘Checkout’ Button.

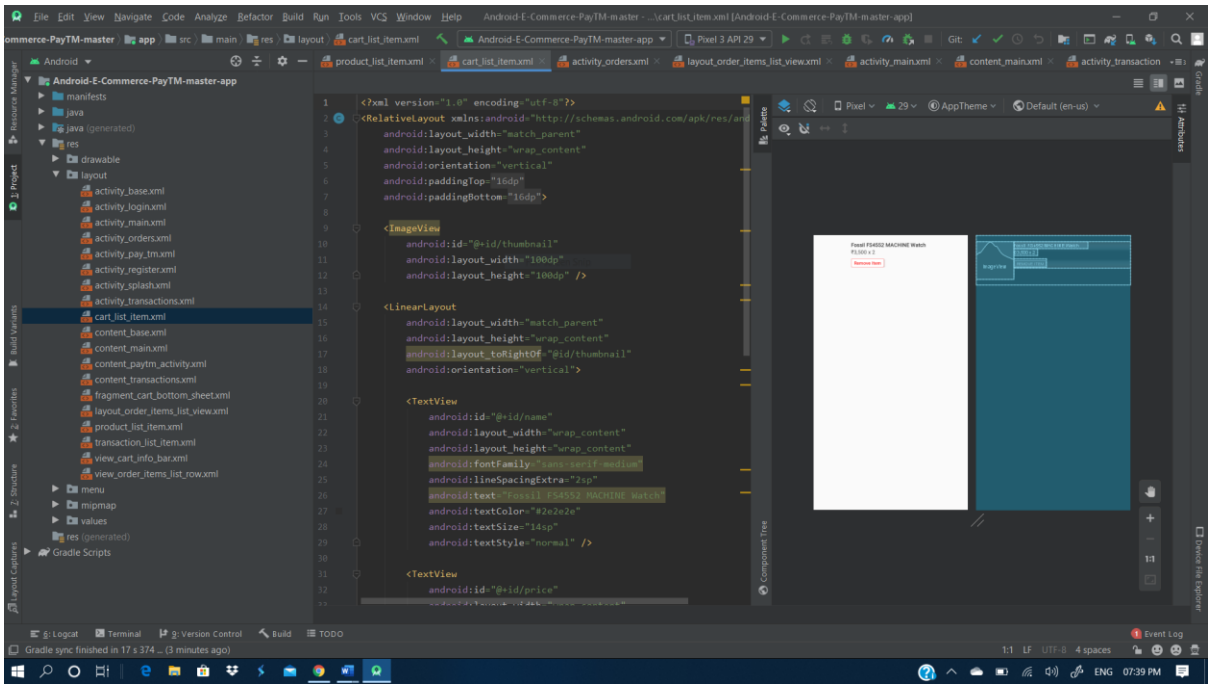

Fig 8.4: THE CART LAYOUT XML SCREENSHOT

In the cart, we have added the option to Remove Items as well. By using the ‘Remove Item’ button, the user can completely remove the selected item from the cart. The selected button would hence be deleted from the database along with the amount of the item being subtracted from the total amount.

The one thing we as Undergraduate students could not do was to add the payment option in this application. Since it requires to add our Bank Account details and to link our account to this project, we decided to remove this option altogether.

Without the payment option, Checkout would be the last step of this project. So once the user selects the option to Checkout, he/she would be taken back to the Login page. Again, we did it by using the Intent function.

Manifest:

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

  <application
      android:name=".app.MyApplication"
      android:allowBackup="true"
      android:icon="@mipmap/images"
      android:label="Second Hand Mart"
      android:roundIcon="@mipmap/images"
      android:supportsRtl="true"
      android:theme="@style/AppTheme">
<activity
         android:name=".ui.transactions.TransactionsActivity"
         android:label="@string/title_activity_transactions"
         android:screenOrientation="portrait"
         android:theme="@style/AppTheme.NoActionBar" />
<activity
         android:name=".ui.splash.SplashActivity"
         android:screenOrientation="portrait"
         android:theme="@style/AppTheme.NoActionBar">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
      <activity
         android:name=".ui.main.MainActivity"
         android:label="@string/app_name"
         android:screenOrientation="portrait"
         android:theme="@style/AppTheme.NoActionBar"></activity>
<activity
         android:name=".ui.login.LoginActivity"
         android:screenOrientation="portrait"
         android:theme="@style/AppTheme.NoActionBar" />

      <activity
         android:name=".ui.register.RegisterActivity"
         android:screenOrientation="portrait"
         android:theme="@style/AppTheme.NoActionBar" />
   </application>
```

BaseActivity.java:

```
public abstract class BaseActivity extends AppCompatActivity {
    private static ApiClient mApi;

    @BindView(R.id.progress_bar)
    ProgressBar progressBar;

    @BindView(R.id.coordinator_layout)
    CoordinatorLayout coordinatorLayout;

    @BindView(R.id.app_bar)
    AppBarLayout appBar;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        View view = getLayoutInflater().inflate(R.layout.activity_base, null);
        FrameLayout activityContainer = view.findViewById(R.id.activity_content_holder);
        DataBindingUtil.inflate(getLayoutInflater(), getLayoutId(), activityContainer, true);
    }

    public ApiClient getApi() {
        if (mApi == null) {
            mApi = ApiService.getClient().create(ApiClient.class);
        }
        return mApi;
    }

    public abstract
    @LayoutRes
    int getLayoutId();

    @Override
    public void setContentView(final int layoutResID) {
        View view = getLayoutInflater().inflate(R.layout.activity_base, null);
        FrameLayout activityContainer = view.findViewById(R.id.activity_content_holder);
        DataBindingUtil.inflate(getLayoutInflater(), getLayoutId(), activityContainer, true);
        super.setContentView(view);
        ButterKnife.bind(this);
    }

    public void showProgress() {
        progressBar.setVisibility(View.VISIBLE);
    }

    public void toggleProgress(boolean isLoading) {
        if (isLoading)
            showProgress();
        else
            hideProgress();
    }
    public void hideProgress() {
        progressBar.setVisibility(View.GONE);
    }
```

```
    public void handleError(Throwable throwable) {
        showErrorDialog(getString(R.string.msg_unknown));
    }

    public void handleUnknownError() {
        showErrorDialog(getString(R.string.msg_unknown));
    }

    public void handleError(ResponseBody responseBody) {
        String message = null;
        if (responseBody != null) {
            try {
                ErrorResponse errorResponse = new Gson().fromJson(responseBody.charStream(),
ErrorResponse.class);
                message = errorResponse.error;
            } catch (JsonSyntaxException e) {
            } catch (JsonIOException e) {
            }
        }

        message = TextUtils.isEmpty(message) ? getString(R.string.msg_unknown) : message;
        showErrorDialog(message);
    }

    public void showErrorDialog(String message) {
        AlertDialog.Builder builder;
        builder = new AlertDialog.Builder(this);
        builder.setTitle(getString(R.string.error))
                .setMessage(message)
                .setPositiveButton(android.R.string.ok, (dialog, which) -> {
                })
                .show();
    }

    public void setToolbar() {
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
    }

    public void hideToolbar() {
        appBar.setVisibility(View.GONE);
    }

    public void changeStatusBarColor() {
        changeStatusBarColor(Color.WHITE);
    }

    public void changeStatusBarColor(int color) {
        Window window = getWindow();
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
        }

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            window.getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
            window.setStatusBarColor(color);
```

```
        }
    }

    public void makeFullScreen() {
        if (Build.VERSION.SDK_INT >= 21) {
            getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LAYOUT_STABLE |
View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN);
        }
    }

    public void enableToolbarUpNavigation() {
        if (getSupportActionBar() != null) {
            getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        }
    }

    public void launchSplash(Activity activity) {
        Intent intent = new Intent(activity, SplashActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
        startActivity(intent);
        finish();
    }

    public void launchLogin(Activity activity) {
        Intent intent = new Intent(activity, LoginActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
        startActivity(intent);
        finish();
    }

    public void checkSession(Activity activity) {
        User user = AppDatabase.getUser();
        if (user == null) {
            Intent intent = new Intent(activity, LoginActivity.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
            startActivity(intent);
        }
    }
}
```

MainActivity.java:

```
public class MainActivity extends BaseActivity implements ProductsAdapter.ProductsAdapterListener {
    @BindView(R.id.recycler_view)
    RecyclerView recyclerView;

    @BindView(R.id.cart_info_bar)
    CartInfoBar cartInfoBar;

    private ProductsAdapter mAdapter;
    private Realm realm;
    private RealmResults<CartItem> cartItems;
    private RealmChangeListener<RealmResults<CartItem>> cartRealmChangeListener;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ButterKnife.bind(this);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        changeStatusBarColor();

        init();
        renderProducts();

        realm = Realm.getDefaultInstance();
        cartItems = realm.where(CartItem.class).findAllAsync();

        cartRealmChangeListener = cartItems -> {
            Timber.d("Cart items changed! " + this.cartItems.size());
            if (cartItems != null && cartItems.size() > 0) {
                setCartInfoBar(cartItems);
                toggleCartBar(true);
            } else {
                toggleCartBar(false);
            }

            mAdapter.setCartItems(cartItems);
        };
    }

    @Override
    public int getLayoutId() {
        return R.layout.activity_main;
    }

    private void setCartInfoBar(RealmResults<CartItem> cartItems) {
        int itemCount = 0;
        for (CartItem cartItem : cartItems) {
            itemCount += cartItem.quantity;
        }
        cartInfoBar.setData(itemCount, String.valueOf(Utils.getCartPrice(cartItems)));
    }
```

```
/**
 * Rendering the products from local db
 */
private void renderProducts() {
    RealmResults<Product> products = AppDatabase.getProducts();
    mAdapter = new ProductsAdapter(this, products, this);
    recyclerView.setAdapter(mAdapter);
    mAdapter.notifyDataSetChanged();
}

private void init() {
    RecyclerView.LayoutManager mLayoutManager = new GridLayoutManager(this, 2);
    recyclerView.setLayoutManager(mLayoutManager);
    recyclerView.addItemDecoration(new GridSpacingItemDecoration(2, dpToPx(10), true));
    recyclerView.setItemAnimator(new DefaultItemAnimator());
    cartInfoBar.setListener(() -> showCart());
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (item.getItemId() == R.id.transactions) {
        startActivity(new Intent(MainActivity.this, TransactionsActivity.class));
        return true;
    }

    if (item.getItemId() == R.id.clear_cart) {
        AppDatabase.clearCart();
    }

    if (item.getItemId() == R.id.logout) {
        AppDatabase.clearData();
        AppPref.getInstance().clearData();
        launchLogin();
    }

    return super.onOptionsItemSelected(item);
}

private void launchLogin() {
    Intent intent = new Intent(MainActivity.this, LoginActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
    startActivity(intent);
}

void showCart() {
    CartBottomSheetFragment fragment = new CartBottomSheetFragment();
    fragment.show(getSupportFragmentManager(), fragment.getTag());
}
```

```
    private int dpToPx(int dp) {
        Resources r = getResources();
        return Math.round(TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, dp,
r.getDisplayMetrics()));
    }

    @Override
    public void onProductAddedCart(int index, Product product) {
        AppDatabase.addItemToCart(product);
        if (cartItems != null) {
            mAdapter.updateItem(index, cartItems);
        }
    }

    @Override
    public void onProductRemovedFromCart(int index, Product product) {
        AppDatabase.removeCartItem(product);
        if (cartItems != null) {
            mAdapter.updateItem(index, cartItems);
        }
    }
    private void toggleCartBar(boolean show) {
        if (show)
            cartInfoBar.setVisibility(View.VISIBLE);
        else
            cartInfoBar.setVisibility(View.GONE);
    }

    @Override
    protected void onPause() {
        super.onPause();
        if (cartItems != null) {
            // cartItems.removeChangeListener(cartRealmChangeListener);
        }
    }

    @Override
    protected void onResume() {
        super.onResume();
        checkSession(MainActivity.this);
        if (cartItems != null) {
            cartItems.addChangeListener(cartRealmChangeListener);
        }
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        if (cartItems != null) {
            cartItems.addChangeListener(cartRealmChangeListener);
        }
        if (realm != null) {
            realm.close();
        }
    }
}
```

LoginActivity.java:

```
public class LoginActivity extends BaseActivity {

    @BindView(R.id.input_email)
    EditText inputEmail;

    @BindView(R.id.input_password)
    EditText inputPassword;

    @BindView(R.id.loader)
    AVLoadingIndicatorView loader;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        makeFullScreen();
        setContentView(R.layout.activity_login);
        ButterKnife.bind(this);
        changeStatusBarColor(ContextCompat.getColor(this, R.color.colorAccent));
        hideToolbar();
    }

    @Override
    public int getLayoutId() {
        return R.layout.activity_login;
    }

    @OnClick(R.id.btn_login)
    void onLoginClick() {
        String email = inputEmail.getText().toString();
        String password = inputPassword.getText().toString();

        if (TextUtils.isEmpty(email) || TextUtils.isEmpty(password)) {
            Toast.makeText(getApplicationContext(), getString(R.string.msg_enter_credentials),
Toast.LENGTH_LONG).show();
            return;
        }

        loader.setVisibility(View.VISIBLE);
        LoginRequest request = new LoginRequest();
        request.email = email;
        request.password = password;
        getApi().login(request).enqueue(new Callback<User>() {
            @Override
            public void onResponse(Call<User> call, Response<User> response) {
                loader.setVisibility(View.INVISIBLE);
                if (!response.isSuccessful()) {
                    handleError(response.errorBody());
                    return;
                }
                AppDatabase.saveUser(response.body());
                AppPref.getInstance().saveAuthToken(response.body().token);
                launchSplash(LoginActivity.this);
            }
```

```
        @Override
        public void onFailure(Call<User> call, Throwable t) {
            loader.setVisibility(View.INVISIBLE);
            handleError(t);
        }
    });
}

@OnClick(R.id.btn_create_account)
void onCreateAccountClick() {
    Intent intent = new Intent(LoginActivity.this, RegisterActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
    startActivity(intent);
    finish();
}
}
```

RegisterActivity.java:

```
public class RegisterActivity extends BaseActivity {
    @BindView(R.id.input_name)
    EditText inputName;

    @BindView(R.id.input_email)
    EditText inputEmail;

    @BindView(R.id.input_password)
    EditText inputPassword;

    @BindView(R.id.loader)
    AVLoadingIndicatorView loader;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        makeFullScreen();
        setContentView(R.layout.activity_register);
        ButterKnife.bind(this);
        changeStatusBarColor(ContextCompat.getColor(this, R.color.colorAccent));
        hideToolbar();
    }

    @Override
    public int getLayoutId() {
        return R.layout.activity_register;
    }

    @OnClick(R.id.btn_register)
    void onRegisterClick() {
        String name = inputName.getText().toString();
        String email = inputEmail.getText().toString();
        String password = inputPassword.getText().toString();
```

```
        if (TextUtils.isEmpty(name) || TextUtils.isEmpty(email) || TextUtils.isEmpty(password)) {
            Toast.makeText(getApplicationContext(), getString(R.string.msg_fill_the_form),
Toast.LENGTH_LONG).show();
            return;
        }

        loader.setVisibility(View.VISIBLE);
        RegisterRequest request = new RegisterRequest();
        request.name = name;
        request.email = email;
        request.password = password;
        request.confirmPassword = password;
        getApi().register(request).enqueue(new Callback<User>() {
            @Override
            public void onResponse(Call<User> call, Response<User> response) {
                loader.setVisibility(View.INVISIBLE);
                if (!response.isSuccessful()) {
                    handleError(response.errorBody());
                    return;
                }

                AppDatabase.saveUser(response.body());
                AppPref.getInstance().saveAuthToken(response.body().token);
                launchSplash(RegisterActivity.this);
            }

            @Override
            public void onFailure(Call<User> call, Throwable t) {
                loader.setVisibility(View.INVISIBLE);
                handleError(t);
            }
        });
    }

    @OnClick(R.id.btn_login_account)
    void onCreateAccountClick() {
        Intent intent = new Intent(RegisterActivity.this, LoginActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
        startActivity(intent);
        finish();
    }
}
```

OrderedItemsListView.java:

```
public class OrderItemsListView extends LinearLayout {
    @BindView(R.id.container)
    LinearLayout layoutContainer;

    private List<OrderItem> items = new ArrayList<>();
    private LayoutInflater inflater;
```

```
    public OrderItemsListView(Context context) {
        super(context);
        init(context, null);
    }

    public OrderItemsListView(Context context, @Nullable AttributeSet attrs) {
        super(context, attrs);
        init(context, attrs);
    }

    private void init(Context context, AttributeSet attrs) {
        inflate(context, R.layout.layout_order_items_list_view, this);
        ButterKnife.bind(this);
        if (inflater == null) {
            inflater = (LayoutInflater) getContext().getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        }
    }

    public void setOrderItems(List<OrderItem> items) {
        this.items.clear();
        this.items.addAll(items);

        renderItems();
    }

    private void renderItems() {
        layoutContainer.removeAllViews();
        for (OrderItem item : items) {
            View view = inflater.inflate(R.layout.view_order_items_list_row, null);
            if (item.product != null) {
                ((TextView) view.findViewById(R.id.name)).setText(item.product.name);
                ((TextView)
view.findViewById(R.id.price)).setText(getContext().getString(R.string.price_with_currency, item.product.price));
            }
            layoutContainer.addView(view);
        }
    }
}
```

CartInfoBar.java:

```
public class CartInfoBar extends RelativeLayout {
    private CartInfoBarListener listener;

    @BindView(R.id.cart_price)
    TextView cartInfo;

    public CartInfoBar(Context context) {
        super(context);
        init(context, null);
    }
```

```
    public CartInfoBar(Context context, @Nullable AttributeSet attrs) {
        super(context, attrs);
        init(context, attrs);
    }

    public void init(Context context, @Nullable AttributeSet attrs) {
        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View view = inflater.inflate(R.layout.view_cart_info_bar, null);
        ButterKnife.bind(this, view);
        addView(view);
    }

    public void setListener(CartInfoBarListener listener) {
        this.listener = listener;
    }

    @OnClick(R.id.container)
    void onContainerClick() {
        if (listener != null)
            listener.onClick();
    }

    public void setData(int itemCount, String price) {
        cartInfo.setText(getContext().getString(R.string.cart_info_bar_data, itemCount, price));
    }

    public interface CartInfoBarListener {
        void onClick();
    }
}
```

CartBottomSheetFragment.java:

```
public class CartBottomSheetFragment extends BottomSheetDialogFragment implements
CartProductsAdapter.CartProductsAdapterListener {

    @BindView(R.id.recycler_view)
    RecyclerView recyclerView;

    @BindView(R.id.btn_checkout)
    Button btnCheckout;

    private Realm realm;
    private CartProductsAdapter mAdapter;
    private RealmResults<CartItem> cartItems;
    private RealmChangeListener<RealmResults<CartItem>> cartItemRealmChangeListener;

    public CartBottomSheetFragment() {
        // Required empty public constructor
    }
```

```
    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        // Making bottom sheet expanding to full height by default
        BottomSheetDialog dialog = (BottomSheetDialog) super.onCreateDialog(savedInstanceState);
        dialog.setOnShowListener(dialog1 -> {
            BottomSheetDialog d = (BottomSheetDialog) dialog1;

            FrameLayout bottomSheet = d.findViewById(android.support.design.R.id.design_bottom_sheet);
            BottomSheetBehavior.from(bottomSheet).setState(BottomSheetBehavior.STATE_EXPANDED);
        });
        return dialog;
    }

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

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_cart_bottom_sheet, container, false);
        ButterKnife.bind(this, view);
        realm = Realm.getDefaultInstance();
        cartItems = realm.where(CartItem.class).findAllAsync();

        cartItemRealmChangeListener = cartItems -> {
            mAdapter.setData(cartItems);
            setTotalPrice();
        };

        cartItems.addChangeListener(cartItemRealmChangeListener);

        return view;
    }

    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        init();
    }
    private void init() {
        mAdapter = new CartProductsAdapter(getActivity(), this);
        RecyclerView.LayoutManager mLayoutManager = new LinearLayoutManager(getActivity());
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setItemAnimator(new DefaultItemAnimator());
        recyclerView.addItemDecoration(new DividerItemDecoration(getActivity(),
LinearLayoutManager.VERTICAL));
        recyclerView.setAdapter(mAdapter);
        mAdapter.notifyDataSetChanged();
        setTotalPrice();
    }
```

```
    private void setTotalPrice() {
        if (cartItems != null) {
            float price = Utils.getCartPrice(cartItems);
            if (price > 0) {
                btnCheckout.setText(getString(R.string.btn_checkout, getString(R.string.price_with_currency, price)));
            } else {
                // if the price is zero, dismiss the dialog
                dismiss();
            }
        }
    }

    @Override
    public void onResume() {
        super.onResume();
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        if (cartItems != null) {
            cartItems.removeChangeListener(cartItemRealmChangeListener);
        }

        if (realm != null) {
            realm.close();
        }
    }

    @OnClick(R.id.ic_close)
    void onCloseClick() {
        dismiss();
    }

    @OnClick(R.id.btn_checkout)
    void onCheckoutClick() {
//        startActivity(new Intent(getActivity(), PayTMActivity.class));

        Intent intent = new Intent(getActivity(), LoginActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
        startActivity(intent);

        return;

    }

    @Override
    public void onCartItemRemoved(int index, CartItem cartItem) {
        AppDatabase.removeCartItem(cartItem);
    }
}
```

TransactionsActivity.java:

```
public class TransactionsActivity extends BaseActivity {

    @BindView(R.id.recycler_view)
    RecyclerView recyclerView;

    @BindView(R.id.layout_empty_data)
    LinearLayout layoutEmptyData;

    private TransactionsAdapter mAdapter;
    private List<Transaction> transactions = new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_transactions);
        ButterKnife.bind(this);
        setToolbar();
        changeStatusBarColor();
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        init();

        fetchTransactions();
    }

    @Override
    public int getLayoutId() {
        return R.layout.activity_transactions;
    }

    private void init() {
        recyclerView.setLayoutManager(new LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false));
        DividerItemDecoration dividerItemDecoration = new DividerItemDecoration(recyclerView.getContext(),
                LinearLayoutManager.VERTICAL);
        recyclerView.addItemDecoration(dividerItemDecoration);
        mAdapter = new TransactionsAdapter(this, transactions);
        recyclerView.setAdapter(mAdapter);
    }
    private void fetchTransactions() {
        toggleProgress(true);
        getApi().getTransactions().enqueue(new Callback<List<Transaction>>() {
            @Override
            public void onResponse(Call<List<Transaction>> call, Response<List<Transaction>> response) {
                toggleProgress(false);
                if (!response.isSuccessful()) {
                    handleUnknownError();
                    return;
                }
                transactions.clear();
                transactions.addAll(response.body());
                mAdapter.notifyDataSetChanged();
                toggleEmptyData();
            }
```

```
        @Override
        public void onFailure(Call<List<Transaction>> call, Throwable t) {
            toggleProgress(false);
            handleError(t);
        }
    });
}

private void toggleEmptyData() {
    if (transactions.size() > 0) {
        layoutEmptyData.setVisibility(View.GONE);
        recyclerView.setVisibility(View.VISIBLE);
    } else {
        layoutEmptyData.setVisibility(View.VISIBLE);
        recyclerView.setVisibility(View.GONE);
    }
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (item.getItemId() == android.R.id.home) {
        finish();
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}
```

SplashActivity.java:

```
public class SplashActivity extends BaseActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        makeFullScreen();
        setContentView(R.layout.activity_splash);
        changeStatusBarColor(ContextCompat.getColor(this, R.color.colorAccent));
        hideToolbar();

        // checking for auth token in prefs
        if (TextUtils.isEmpty(AppPref.getInstance().getAuthToken())) {
            // user token is not present, take him to login screen
            Intent intent = new Intent(SplashActivity.this, LoginActivity.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
            startActivity(intent);
            finish();
            return;
        }
        fetchAppConfig();
    }
```

```
@Override
public int getLayoutId() {
    return R.layout.activity_splash;
}

/**
 * Fetching app configuration from server
 * This will get PayTM configuration required for payment related operations
 */
private void fetchAppConfig() {
    Call<AppConfig> call = getApi().getAppConfig();
    call.enqueue(new Callback<AppConfig>() {
        @Override
        public void onResponse(Call<AppConfig> call, Response<AppConfig> response) {
            if (!response.isSuccessful()) {
                handleError(response.errorBody());
                return;
            }

            // save app config to db
            AppDatabase.saveAppConfig(response.body());

            // fetch products
            fetchProducts();
        }

        @Override
        public void onFailure(Call<AppConfig> call, Throwable t) {
            handleError(t);
        }
    });
}

/**
 * Fetching products on splash screen before loading home screen
 */
private void fetchProducts() {
    Call<List<Product>> call = getApi().getProducts();
    call.enqueue(new Callback<List<Product>>() {
        @Override
        public void onResponse(Call<List<Product>> call, Response<List<Product>> response) {
            if (!response.isSuccessful()) {
                handleError(response.errorBody());
                return;
            }

            // store products in db
            AppDatabase.saveProducts(response.body());

            // start home screen
            launchHomeScreen();
        }
```

```
            @Override
            public void onFailure(Call<List<Product>> call, Throwable t) {
                handleError(t);
            }
        });
    }

    private void launchHomeScreen() {
        Intent intent = new Intent(SplashActivity.this, MainActivity.class);
        startActivity(intent);
        finish();
    }
}
```

Chapter 9

Screen shots of Project

9.1 SCREEN SHOTS OF THE WORKING APPLICATION

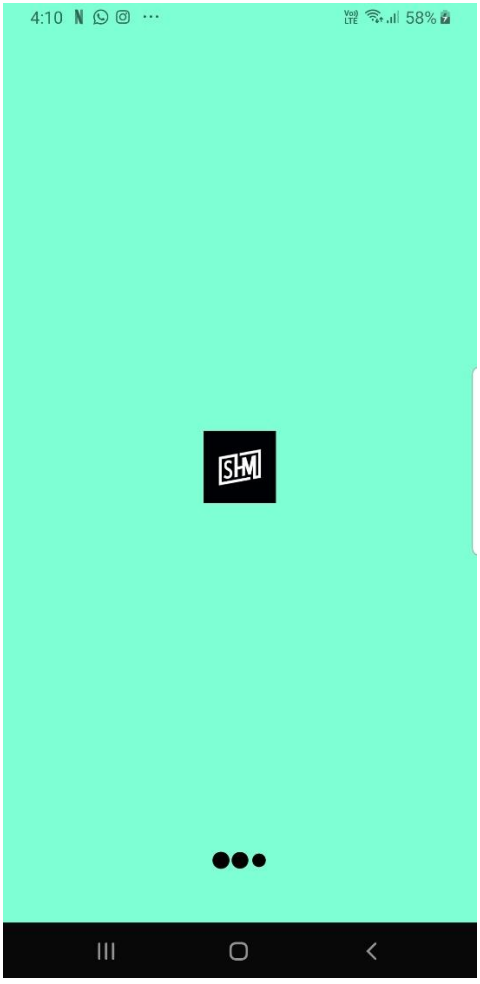

Fig 9.11: APPLICATION SPLASH SCREEN

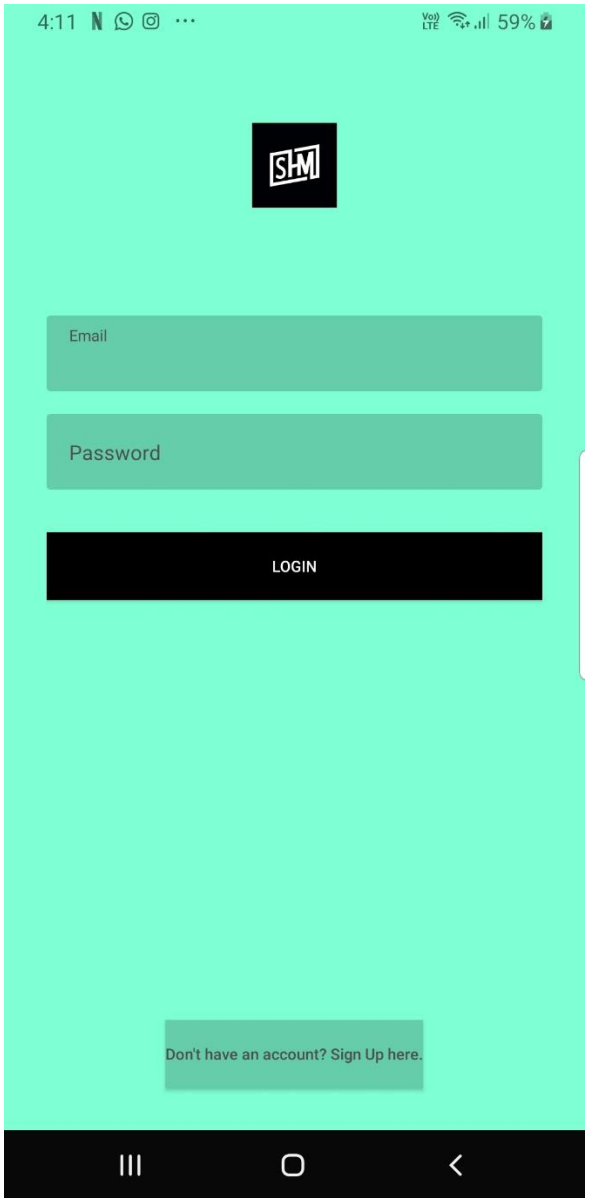

Fig 9.12: LOGIN PAGE

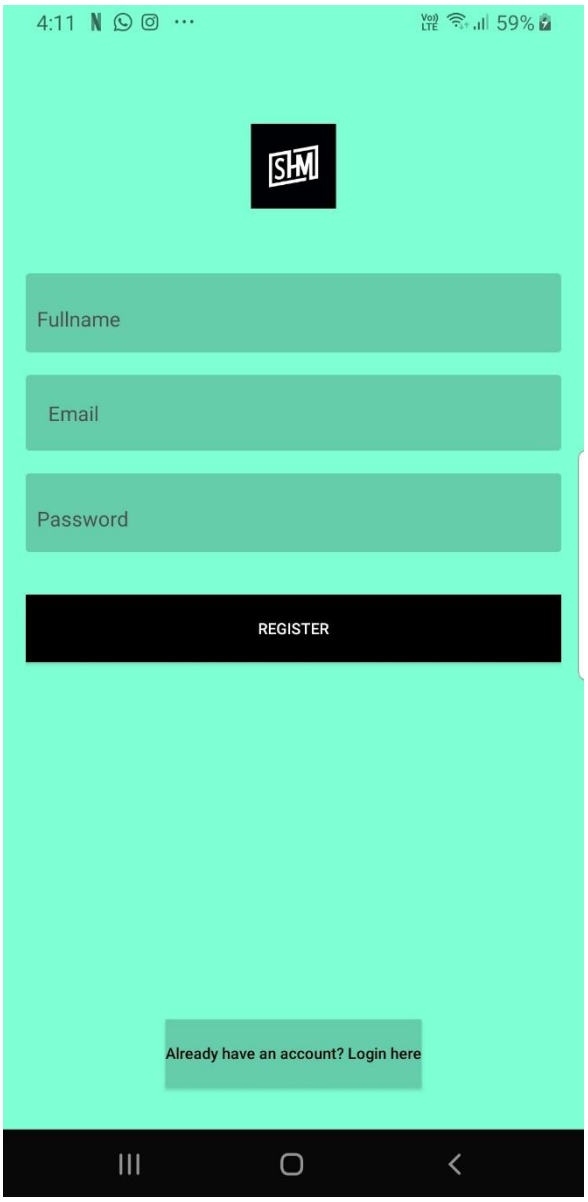

Fig 9.13: REGISTRATION PAGE

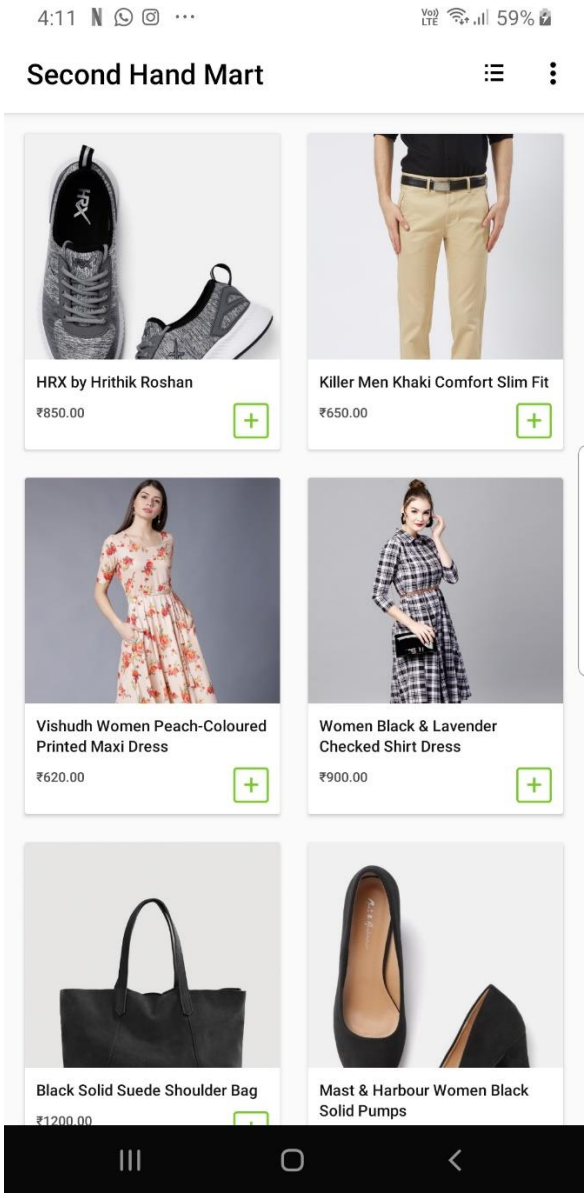

Fig 9.14: SHOPPING ITEMS

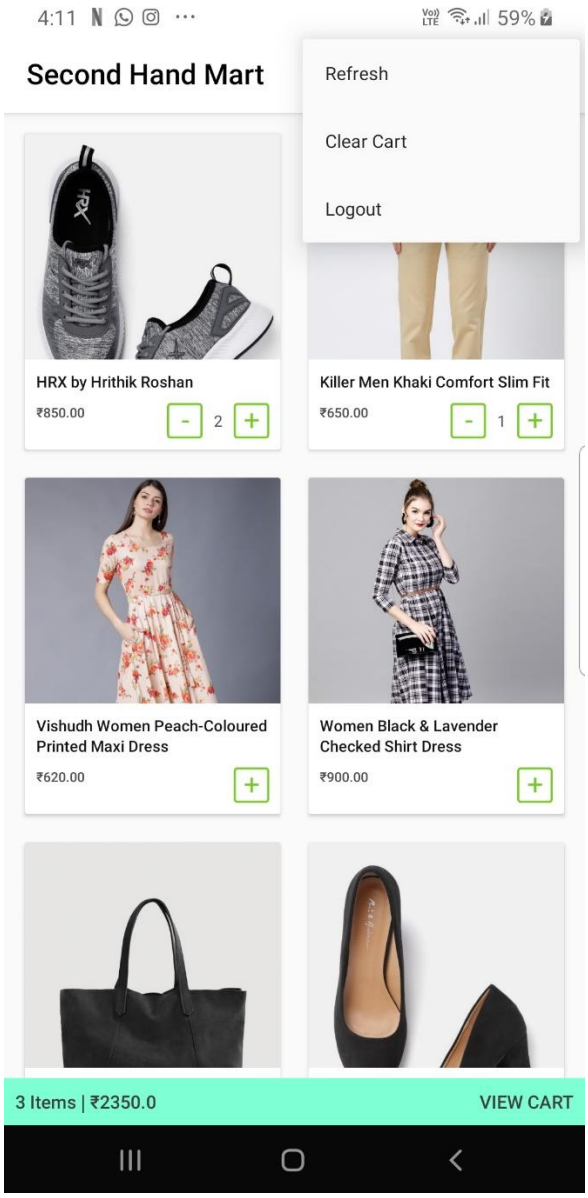

Fig 9.15: DROPDOWN MENU

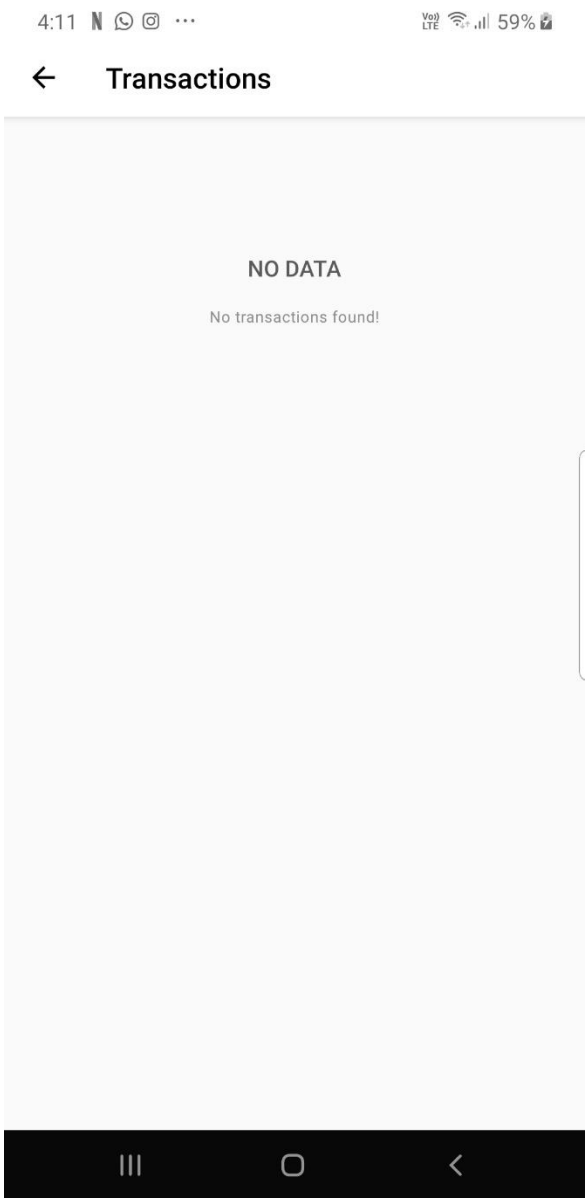

Fig 9.16: PREVIOUS TRANSACTIONS PAGE

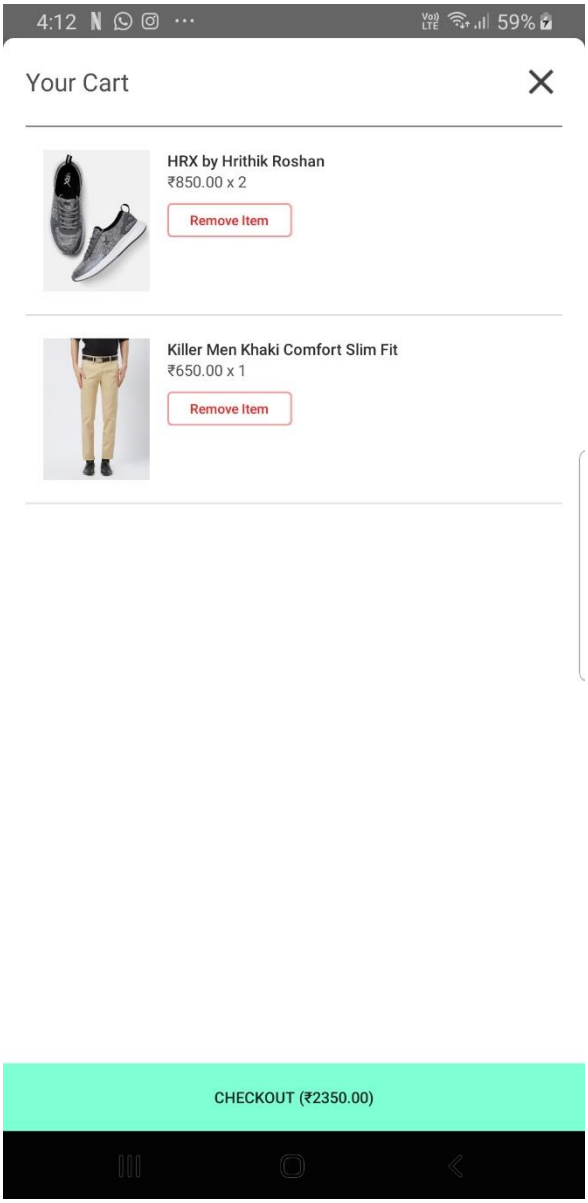

Fig 9.17: CART WITH CHECKOUT BUTTON

Chapter 10

Conclusion and Future Scope

10.1 Conclusion

In this project, we developed an Android Application that can be used for buying "second-hand" products online implementing the things we learned in our theoretical and lab sessions along with the concepts we got familiar with in our trainings and internships. We used Realtime Database for saving user and product details, as well as the transactions that a user has made. Along with this, we have added E-mail authentication to keep the application secure.

We used Android Studio 3.6.1 for creating out application. We got the chance to use our knowledge of Android development, JAVA, Database Management and its various applications used in android development. This project also helped us in learning about new things like Splash Screen and how to use the MySQL Database in Android Studio.

10.2 Future Scope

There is a lot of future scope in this project. Some of them are as follows:

Under a proper authorization, different payment methods like Cash on Delivery, Credit & Debit cards, Wallets and UPI can be implemented to this application.
Many new features can be added to this project like:

- Search option
- Customer Support Service
- Order Cancellation Option

Using this as a base application, different categories of products can be added to establish this application up to a wider range of customers.

References

[1] https://en.wikipedia.org/wiki/Android_(operating_system)

[2] https://developer.android.com/studio/intro

[3] https://en.wikipedia.org/wiki/Flipkart

[4] https://www.thehindubusinessline.com/info-tech/how-ebay-failed-to-gauge-indias-ecomm-scene/article9640407.ece

[5] https://howtofirebase.com/what-is-firebase-fcb8614ba442

[6] https://en.wikipedia.org/wiki/XAMPP

SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

SECOND HAND MART- SHM

SPARSH SINHA
1606067

Abstract: In this project, our goal was to develop an Android Application that can be used for buying "second-hand" products online. We got the opportunity to implement the things we learned in our theoretical and lab sessions along with the concepts we got familiar with in our trainings and internships. We have used Realtime Database for saving user and product details, as well as the transactions that a user has made. Along with this, we have added E-mail authentication to keep the application secure. We have used Android Studio 3.6.1 for creating out application. We used our knowledge of Android development, JAVA, Database Management and its various applications used in android development.

Individual contribution and findings: Did the whole implementation (which includes the source codes and layouts), i.e., developed the application using the knowledge of JAVA and Android Development. Also performed the System Testing by performing various tests after development.

Individual contribution to project report preparation: Introduction, Software Requirement Specification, System Testing, Implementation.

Individual contribution for project presentation and demonstration: Introduction, Implementation w/ Manifest, System Testing.

Full Signature of Supervisor: ……

Full signature of the student: ……..

SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

SECOND HAND MART- SHM

SIDDHARTH AGRAWAL
1606306

Abstract: In this project, our goal was to develop an Android Application that can be used for buying "second-hand" products online. We got the opportunity to implement the things we learned in our theoretical and lab sessions along with the concepts we got familiar with in our trainings and internships. We have used Realtime Database for saving user and product details, as well as the transactions that a user has made. Along with this, we have added E-mail authentication to keep the application secure. We have used Android Studio 3.6.1 for creating out application. We used our knowledge of Android development, JAVA, Database Management and its various applications used in android development.

Individual contribution and findings: Did the planning for the project by keeping various technicalities in mind. Created the whole System Design and did a survey on previous projects, similar to our project.

Individual contribution to project report preparation: Literature Survey, System Design, Project Planning.

Individual contribution for project presentation and demonstration: System Design w/ Explanation, Conclusion and Future Scope.

Full Signature of Supervisor: ……

Full signature of the student: ……..

SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

SECOND HAND MART- SHM

RISHI DHACHOLIA
1606373

Abstract: In this project, our goal was to develop an Android Application that can be used for buying "second-hand" products online. We got the opportunity to implement the things we learned in our theoretical and lab sessions along with the concepts we got familiar with in our trainings and internships. We have used Realtime Database for saving user and product details, as well as the transactions that a user has made. Along with this, we have added E-mail authentication to keep the application secure. We have used Android Studio 3.6.1 for creating out application. We used our knowledge of Android development, JAVA, Database Management and its various applications used in android development.

Individual contribution and findings: Did the study for all the requirements of the project, which includes all the software, as well as the hardware specifications. Also took all the Screenshots of the working application after its development was completed.

Individual contribution to project report preparation: Requirement Analysis, Screenshots, Conclusion and Future Scope.

Individual contribution for project presentation and demonstration: Requirement Analysis, Screenshots.

Full Signature of Supervisor: ……

Full signature of the student: ……..