ITRW 324 PHASE 2

Technical report: Mobile App

GROUP MEMBERS:

Marc Kishinkwa 24246603

Salmaan Vally 26049597

Adrian Van Kraayenburg 24165220

Mathapelo Ngoasheng 25955527

Dinah Lee Booysen 26186365

Contents

Overview

WeSellcc is an online clothing store which is home to a diverse variety of brands from all across the world, the company is aiming at reinventing the E-commerce market by keeping the platforms (website and app) simple, free flowing, easy to use and most importantly being home to unbeatable prices with shipping included. During this phase of the development the WeSell developers created a mobile application based on the website that was created in the first phase. Statistics from various research projects claim that 85% of users prefer a mobile application over a mobile website. The application is an android based application which allows users to:

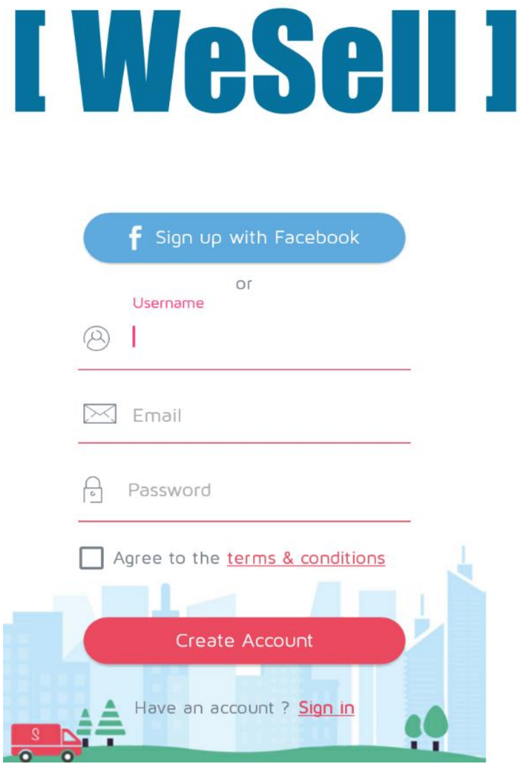
- Create an account
- Sign in(authentication)
- Browse product categories
- Add products to cart
- Checkout
- Pay for products using PayPal

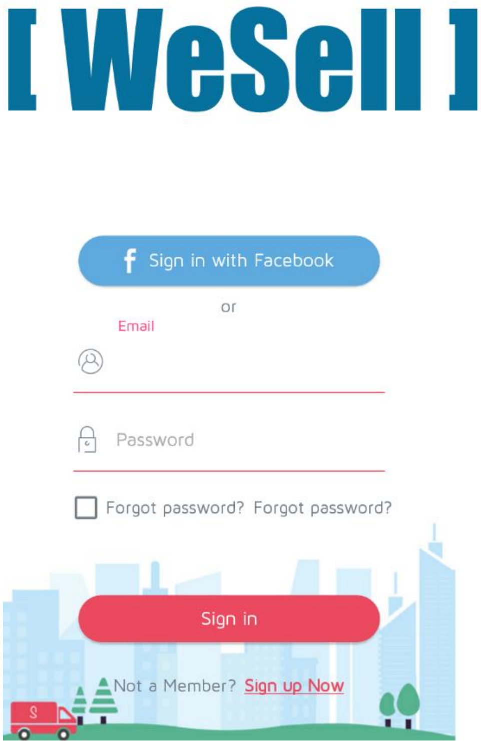
The mobile app does exactly what the website does and makes use of the same database. The app runs smoothly and is more convenient for android device users to shop using the app instead of the website.

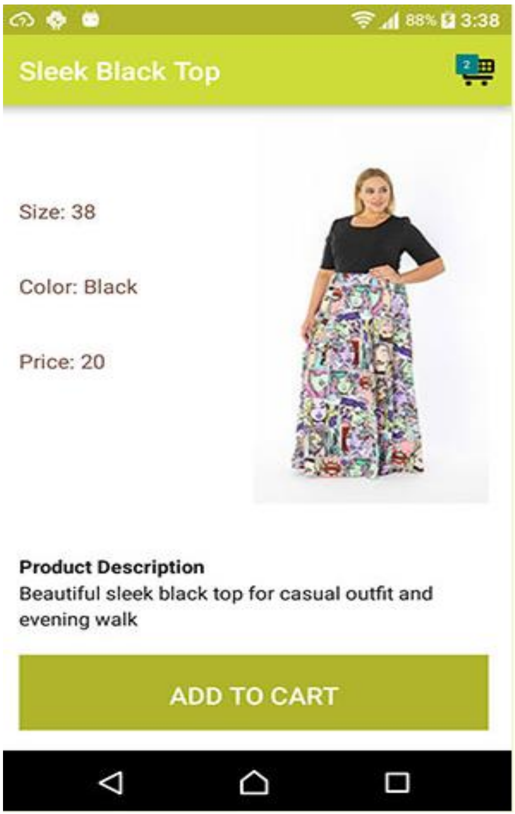
WeSell mobile application objectives

- Design of a simple vibrant interface
- Ease of use(hassle-free experience) for android users
- Provide seamless navigation between products
- Secure payment system using PayPal
- Easy registration process with minimum information required
- Be able to select quantity and sizes of inquired product before checkout

WeSell mobile app WireFrame




[WeSell]
Sign in with Facebook
or
Email
Password
□ Forgot password? Forgot password?
Sign in
Not a Member? Sign up Now

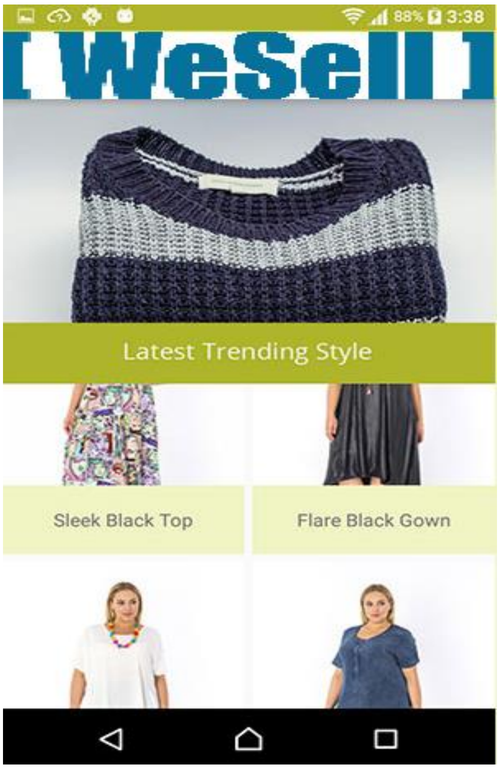


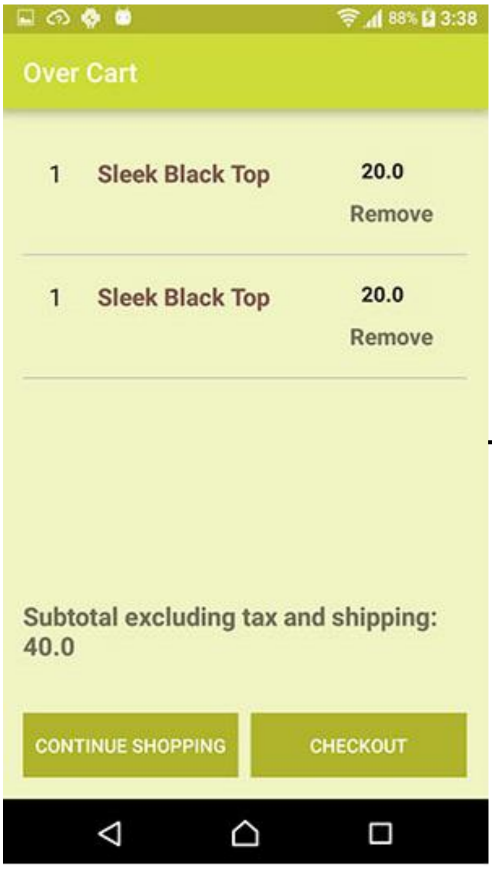
Size: 38

Color: Black

Price: 20

Product Description
Beautiful sleek black top for casual outfit and evening walk



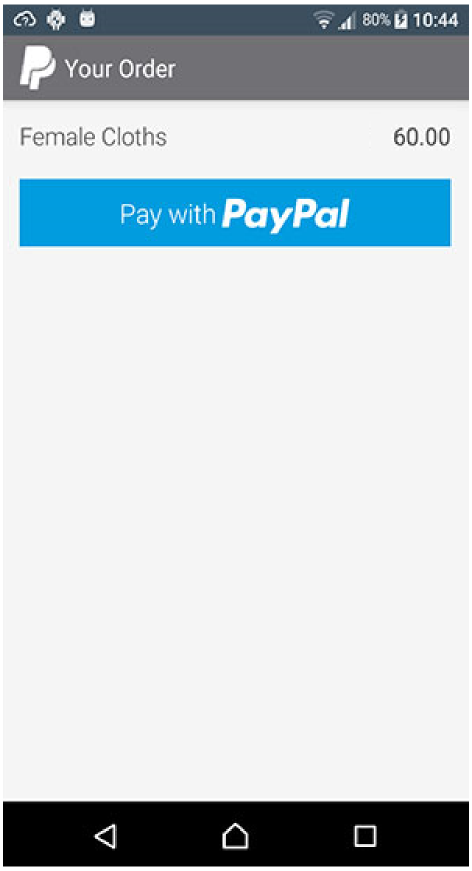


Over Cart

1 Sleek Black Top 20.0 Remove

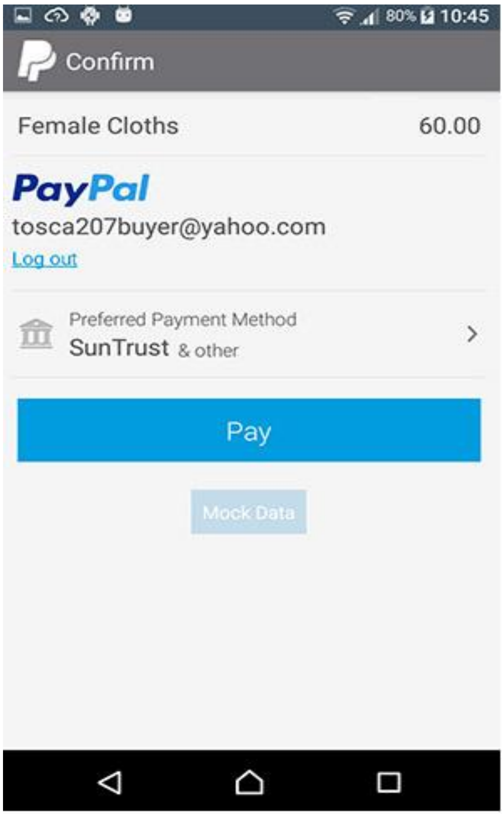
1 Sleek Black Top 20.0 Remove

Subtotal excluding tax and shipping: 40.0



Your Order

Female Cloths 60.00



Confirm

Female Cloths 60.00

PayPal

tosca207buyer@yahoo.com

Log out

Preferred Payment Method
SunTrust & other

Development procedure

The development procedure used by the WeSell developers consists of the following phases:

1. PHASE1 – Pre-planning and research

During this phase the WeSell developers brainstormed and did research to find out the following:

- What is the aim of the app?

The main aim of the app is to create a free flowing, easy to use platform where users can purchase clothes using their mobile devices instead of using the mobile site

- Who is the target audience?

The target audience is people with a taste for trending fashion that like to shop from the comfort of their homes along with people who do not have time on their hands to go to a store to purchase a product, instead they can just make use of WeSell.

- What platform will be used?

The application the WeSell developers designed is solely for android users.

2. PHASE2 – Mental prototyping

This phase involves visualising the application, to see the results of this phase, please refer to the WeSell mobile application objectives tab which was represented before the development procedure.

3. PHASE3 – Assessment of technical feasibility

The application will make use of google and android API's to instruct software components on how to asses data. It will also make use of a server to server API call for the integration of the PayPal system. The technical aspects will be discussed more in detail in the Technicalities of the apps functionality below.

4. PHASE4 – Building a prototype

During this phase the WeSell developers built a prototype shell which just illustrated some of the functions provided by the mobile application without any data, this helped give the developers an idea of the working of the application and where can we make any improvements.

5. PHASE5 – Designing and development of App

In this phase of development the WeSell developers divided into two group namely:

- User experience designer(To create the interaction architecture)
- User interface designer(To create the look of the app)

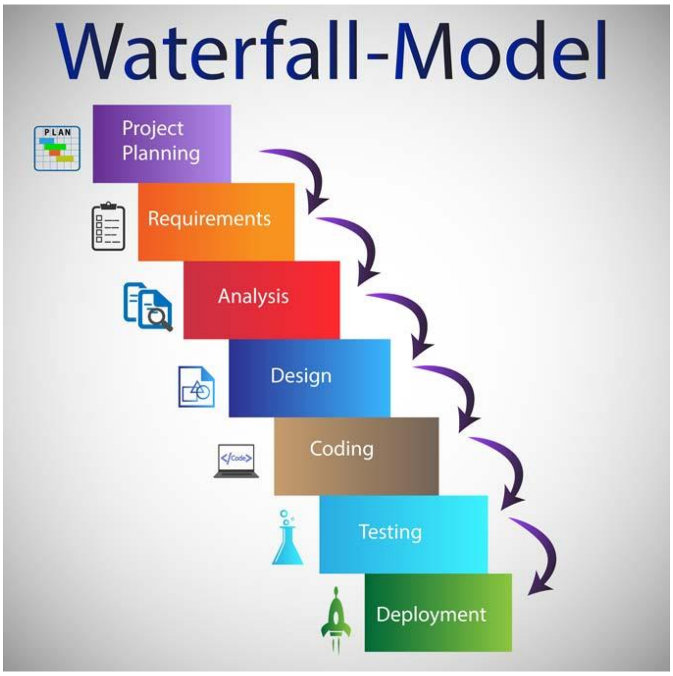
Furthermore the whole app was coded in android studio with making use of databases to store PayPal information, XML files were created for the layout of the app , PHP was used to create a connection between the application and the database.

6. PHASE6 – Testing the mobile app

During this phase the app was debugged and all functions were tested in all ways possible.

Development methodology

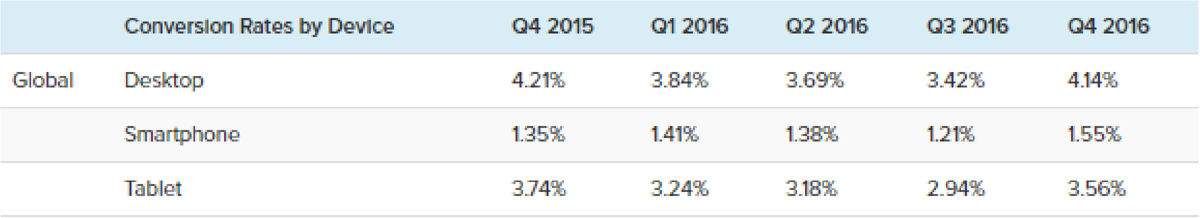
The whole mobile application development was based on the waterfall development methodology which consists of a sequential design mainly:



Each step of the methodology is explained below

Project Planning

Before starting the initial project of creating and application the need for such an application was identified by doing a feasibility study. This indicated that the growth and need for ecommerce is growing worldwide with access to ecommerce by means of tablet and smartphone combined scoring a higher percentage than that of computers.



	Conversion Rates by Device	Q4 2015	Q1 2016	Q2 2016	Q3 2016	Q4 2016
Global	Desktop	4.21%	3.84%	3.69%	3.42%	4.14%
	Smartphone	1.35%	1.41%	1.38%	1.21%	1.55%
	Tablet	3.74%	3.24%	3.18%	2.94%	3.56%

The decision to create an android based application was decided based on the most used mobile software in South Africa with android making up a total of 67% according to Vodacom South Africa.

Requirements

In order for the WeSell application to run correctly android version 3.1 and later are required. Enough space is required to download and install the application which can vary due to changes and updates. With the use of android 3.1 or higher user requirements are also met by ensuring fast and easy navigation of the app. Main requirements are to keep as much functionality of the original website but changing it to an easy to use and efficient but secure android application.

Analysis & Design

Upon initial analysis of the project a time schedule was created as to ensure the completion of the application within the given time. This gave an indication of what could be done in the given time period as well as what options were available in creating an android application based on the WeSell website. Once solutions were found in solving all the necessary requirements of the application and researching development of ecommerce applications the design and layout process was started.

Coding & Testing

The project was developed in sections creating and developing parts of the application from start to finish keeping all requirements in mind. During this further research was done during the coding process investigating different ways and techniques of reaching goals throughout the project.

Testing was done by creating beta versions and allowing users to test the application before it is published.

Deployment

The WeSell app is not published to google play but once the app is fully tested and approved by the development leaders the option of publishing the app is available.

Why the waterfall development methodology?

The advantage of using this methodology to develop the WeSell mobile application lies in the firmness of the scope and the consideration in planning. If all fundamentals of the application have been well thought out, and the guidelines are followed, it allows you to create a detailed evaluation of the cost and timeframe of the project. The strictness of the methodology in the sense of, if one function is developed and perfected and the developer cannot go back, helps a lot when it comes to debugging, since errors are resolved as soon as they occur.

Technicalities of the apps functionality

The following tools and environment was used:

- Windows 10
- Android Studio
- Sony Xperia ZL
- Min SDK 16
- Target SDK 25

About android studio

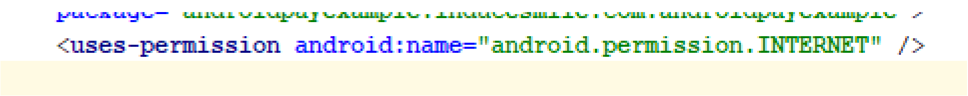
Android studio is and IDE for googles android OS, designed with the main object of android application development. It has automatic debugging feautures which aids in helpings develepors write error free code.

REQUIREMENTS version 2x(Version used in development):

- Windows 7 or later
- Minimum 3GB RAM.
- 500MB disk space
- Java development kit

Update androidManifest.xml file to allow internet access

Add the following code:

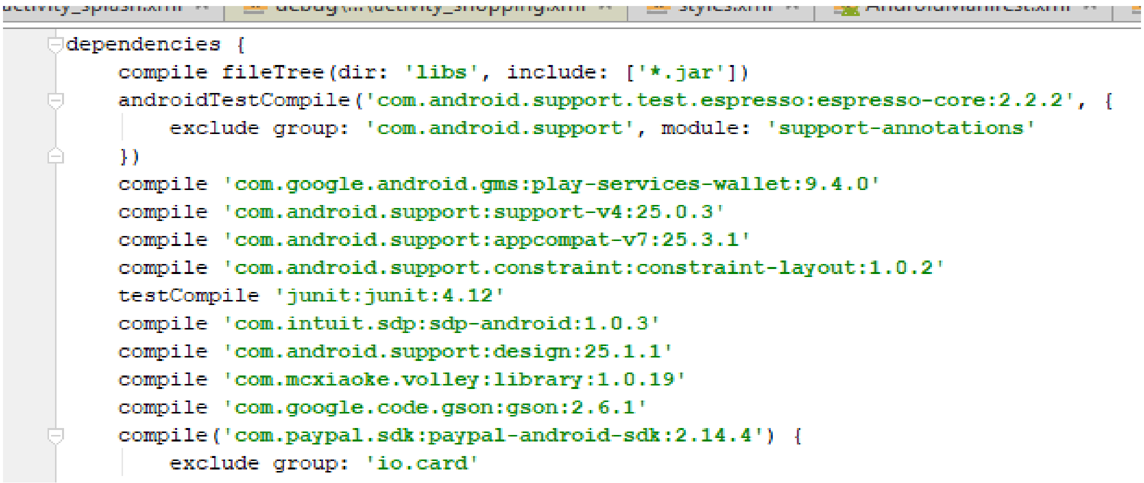


```
<uses-permission android:name="android.permission.INTERNET" />
```

Figure 1:androidmanifest.XML

Update build.gradle file to allow third party libraries under depencies

Add the following code:



```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.google.android.gms:play-services-wallet:9.4.0'
    compile 'com.android.support:support-v4:25.0.3'
    compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    testCompile 'junit:junit:4.12'
    compile 'com.intuit.sdp:sdp-android:1.0.3'
    compile 'com.android.support:design:25.1.1'
    compile 'com.mcxiaoke.volley:library:1.0.19'
    compile 'com.google.code.gson:gson:2.6.1'
    compile('com.paypal.sdk:paypal-android-sdk:2.14.4') {
        exclude group: 'io.card'
```

Figure 2:Build.gradle

Values for layout can be found in res folder

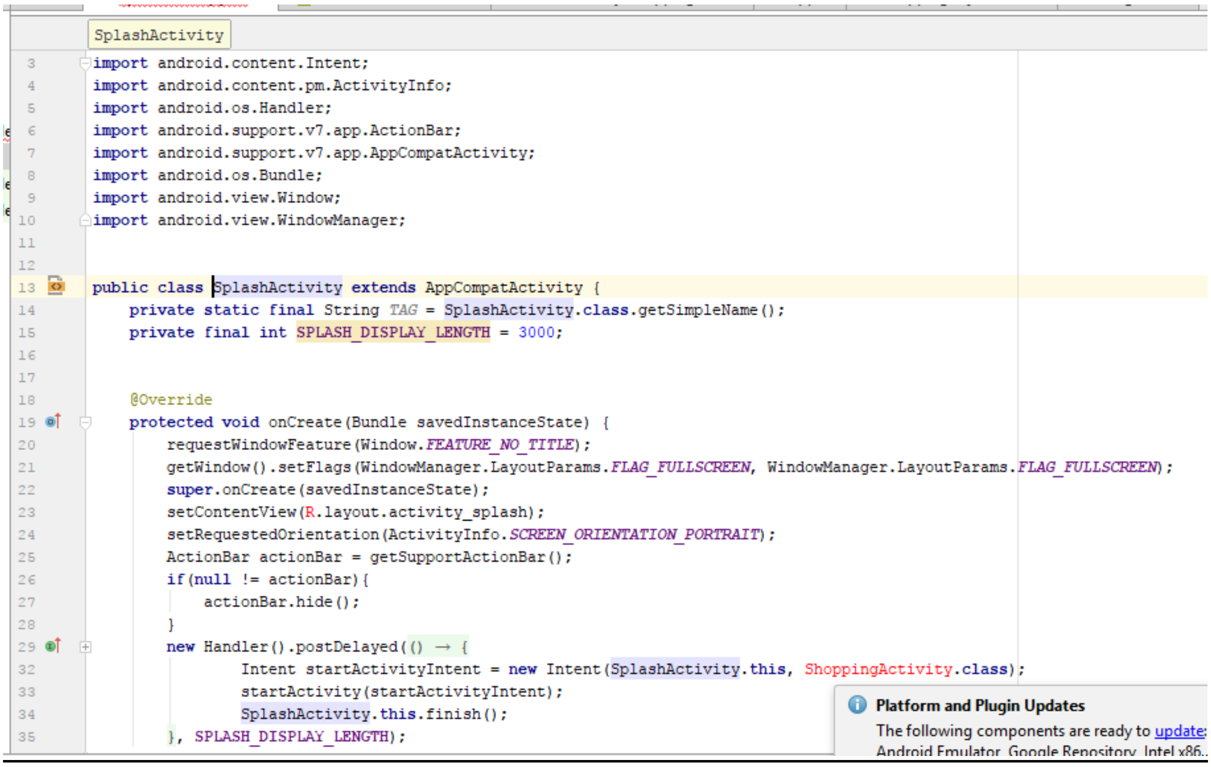
All methods make use of the following layout values in the files for color,string names,

- Colors.xml
- Strings.xml
- Styles.xml

About splashactivity.java file

Once this page is loaded, it will take three seconds before it redirects to the ShoppingActivity page.

Example of code is shown below:



```java
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.os.Handler;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Window;
import android.view.WindowManager;


public class SplashActivity extends AppCompatActivity {
    private static final String TAG = SplashActivity.class.getSimpleName();
    private final int SPLASH_DISPLAY_LENGTH = 3000;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        ActionBar actionBar = getSupportActionBar();
        if(null != actionBar){
            actionBar.hide();
        }
        new Handler().postDelayed(() → {
                Intent startActivityIntent = new Intent(SplashActivity.this, ShoppingActivity.class);
                startActivity(startActivityIntent);
                SplashActivity.this.finish();
        }, SPLASH_DISPLAY_LENGTH);
```

Platform and Plugin Updates

The following components are ready to update:

Android Emulator, Google Repository, Intel x86...

Figure 3:splashactivity.java

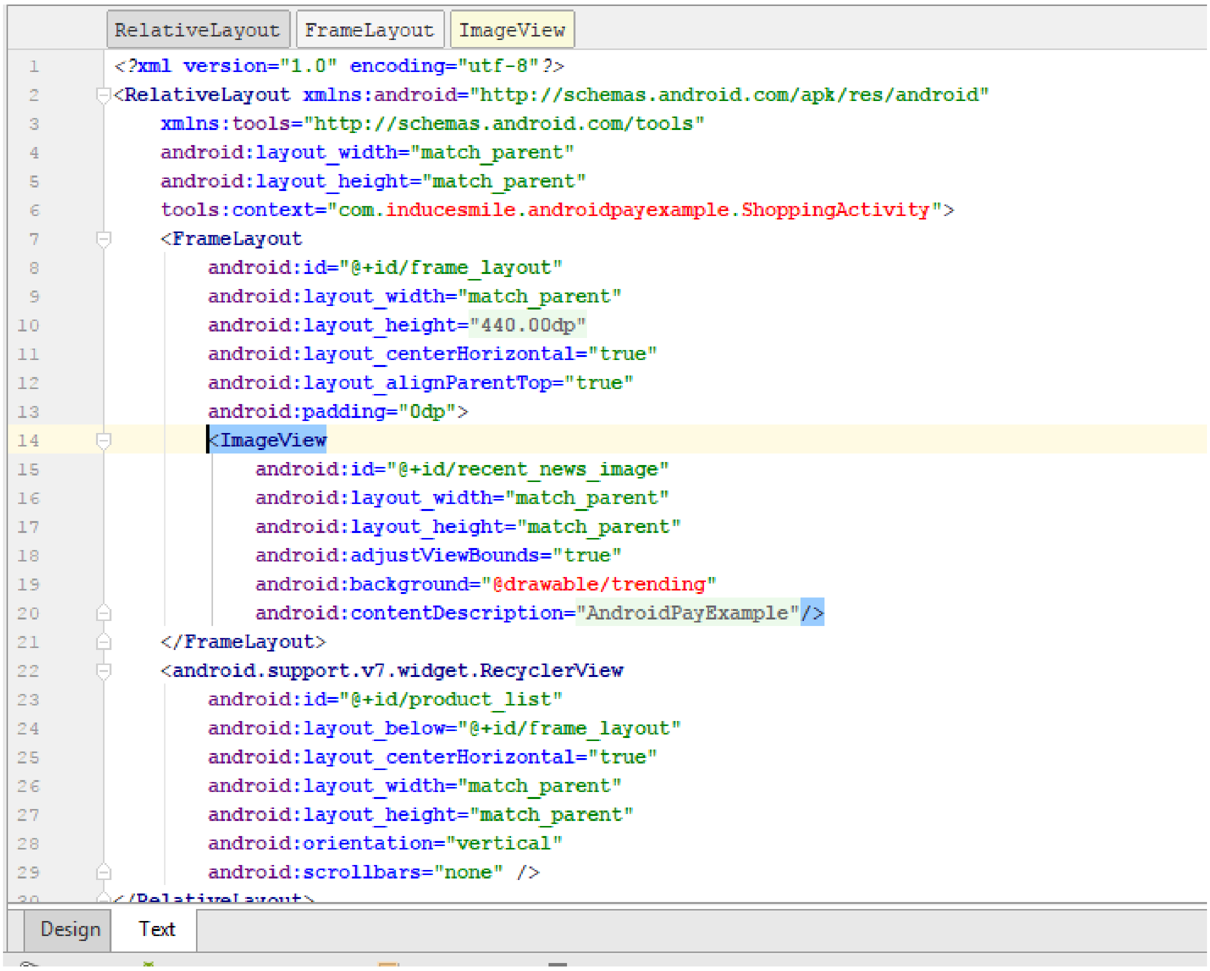
- Xml

Xml means Extensible Markup Language. Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses. The goal of using Android's XML vocabulary, is to quickly design UI layouts and the screen elements they contain, in the same way that creating web pages in HTML : with a series of nested elements.

Main front shop design

The activity_shopping.xml is the main front shop design. It will contain a top ImageView and below it will be RecyclerView that will hold products on sale.

Below is the code for activity_shopping.xml



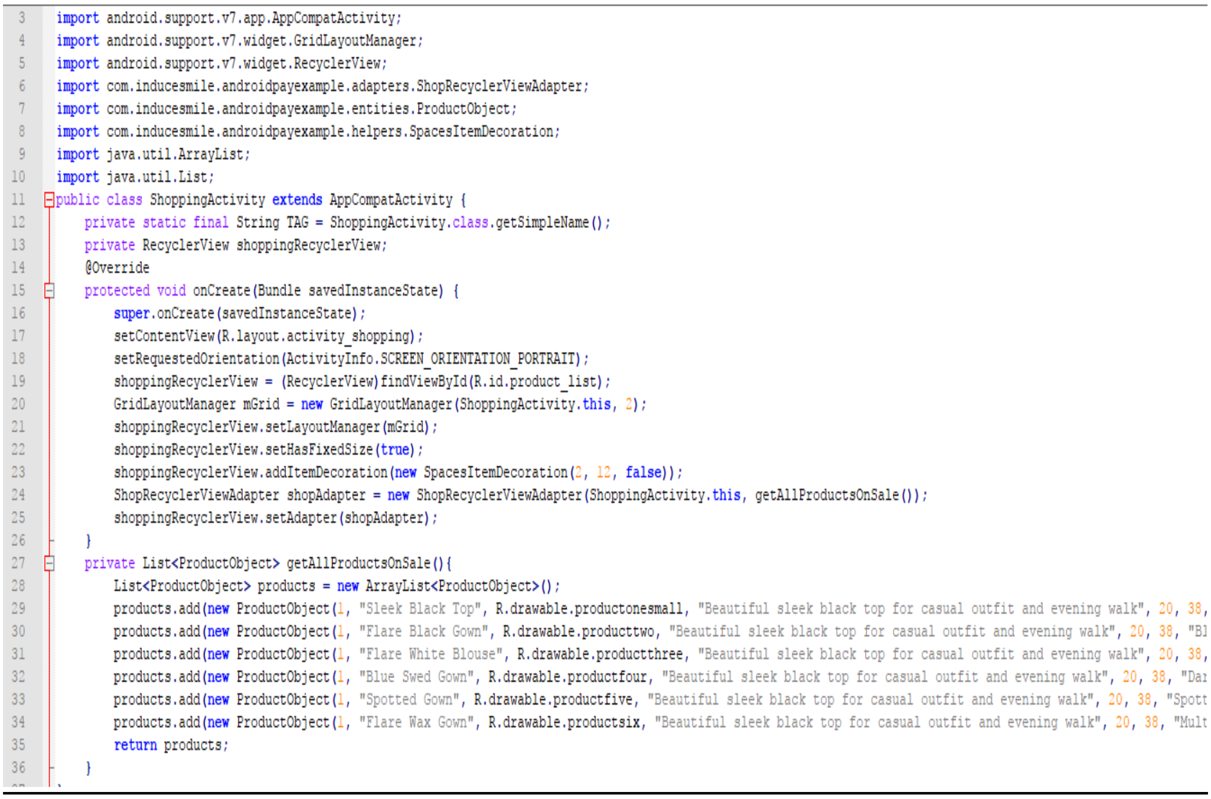
```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.inducesmile.androidpayexample.ShoppingActivity">
    <FrameLayout
        android:id="@+id/frame_layout"
        android:layout_width="match_parent"
        android:layout_height="440.00dp"
        android:layout_centerHorizontal="true"
        android:layout_alignParentTop="true"
        android:padding="0dp">
        <ImageView
            android:id="@+id/recent_news_image"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:adjustViewBounds="true"
            android:background="@drawable/trending"
            android:contentDescription="AndroidPayExample"/>
    </FrameLayout>
    <android.support.v7.widget.RecyclerView
        android:id="@+id/product_list"
        android:layout_below="@+id/frame_layout"
        android:layout_centerHorizontal="true"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:scrollbars="none" />
</RelativeLayout>
```

Figure 4: activity_shopping.xml

Binding data source to recyclerview

In ShoppingActivity class GridLayoutManager is used and we created a RecyclerView adapter that will bind the data source to the View. The data is objects which displays information about the products are stored in a ListView.

Below is a screenshot displaying shoppingactivity.java

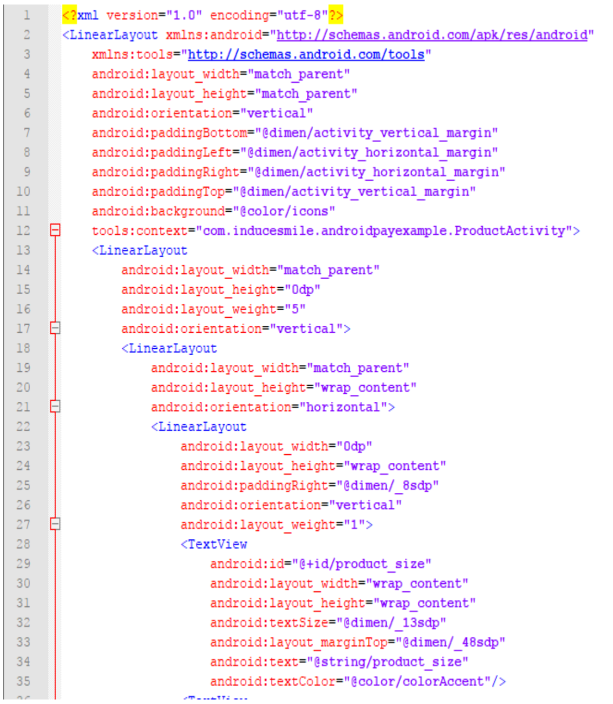


```java
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.RecyclerView;
import com.inducesmile.androidpayexample.adapters.ShopRecyclerViewAdapter;
import com.inducesmile.androidpayexample.entities.ProductObject;
import com.inducesmile.androidpayexample.helpers.SpacesItemDecoration;
import java.util.ArrayList;
import java.util.List;
public class ShoppingActivity extends AppCompatActivity {
    private static final String TAG = ShoppingActivity.class.getSimpleName();
    private RecyclerView shoppingRecyclerView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_shopping);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        shoppingRecyclerView = (RecyclerView)findViewById(R.id.product_list);
        GridLayoutManager mGrid = new GridLayoutManager(ShoppingActivity.this, 2);
        shoppingRecyclerView.setLayoutManager(mGrid);
        shoppingRecyclerView.setHasFixedSize(true);
        shoppingRecyclerView.addItemDecoration(new SpacesItemDecoration(2, 12, false));
        ShopRecyclerViewAdapter shopAdapter = new ShopRecyclerViewAdapter(ShoppingActivity.this, getAllProductsOnSale());
        shoppingRecyclerView.setAdapter(shopAdapter);
    }
    private List<ProductObject> getAllProductsOnSale(){
        List<ProductObject> products = new ArrayList<ProductObject>();
        products.add(new ProductObject(1, "Sleek Black Top", R.drawable.productonesmall, "Beautiful sleek black top for casual outfit and evening walk", 20, 38,
        products.add(new ProductObject(1, "Flare Black Gown", R.drawable.producttwo, "Beautiful sleek black top for casual outfit and evening walk", 20, 38, "Bl
        products.add(new ProductObject(1, "Flare White Blouse", R.drawable.productthree, "Beautiful sleek black top for casual outfit and evening walk", 20, 38,
        products.add(new ProductObject(1, "Blue Swed Gown", R.drawable.productfour, "Beautiful sleek black top for casual outfit and evening walk", 20, 38, "Dar
        products.add(new ProductObject(1, "Spotted Gown", R.drawable.productfive, "Beautiful sleek black top for casual outfit and evening walk", 20, 38, "Spott
        products.add(new ProductObject(1, "Flare Wax Gown", R.drawable.productsix, "Beautiful sleek black top for casual outfit and evening walk", 20, 38, "Mult
        return products;
    }
```

Figure 5:shoppingactivity.java

Detailed information about product

once the user selects a product it will take you to the product page where the detailed information about the product is shown..Below is the code for activity_product.xml



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="@color/icons"
    tools:context="com.inducesmile.androidpayexample.ProductActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="5"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <LinearLayout
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:paddingRight="@dimen/_8sdp"
                android:orientation="vertical"
                android:layout_weight="1">
                <TextView
                    android:id="@+id/product_size"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:textSize="@dimen/_13sdp"
                    android:layout_marginTop="@dimen/_48sdp"
                    android:text="@string/product_size"
                    android:textColor="@color/colorAccent"/>
```

Figure 6:activity_product.xml

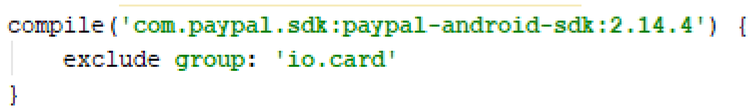
Android PayPal integration

- For android paypal integration we created a sandbox account

your can click this link (https://developer.paypal.com/developer/accounts/create)

Add PayPal SDK

The SDK was added in the build.gradle file



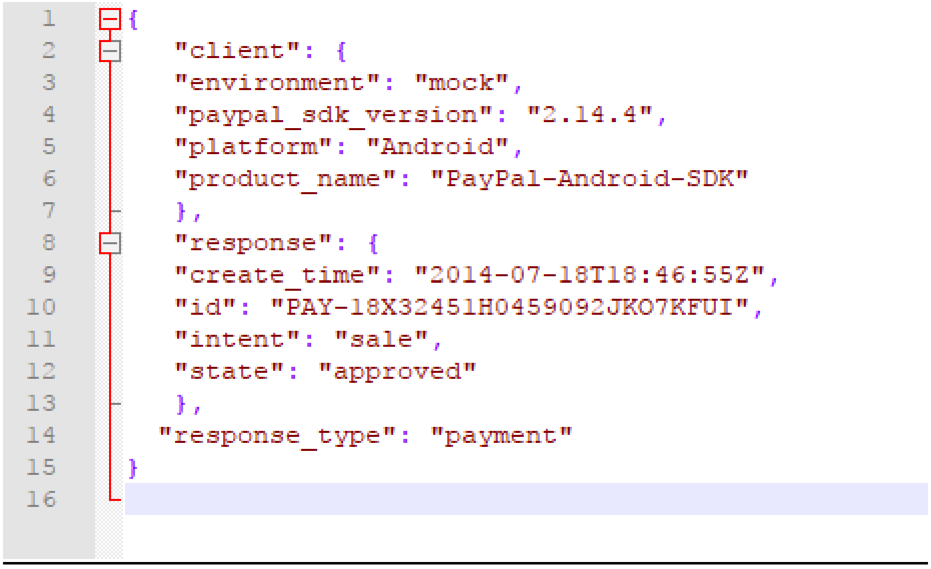
```
compile('com.paypal.sdk:paypal-android-sdk:2.14.4') {
    exclude group: 'io.card'
}
```

Figure 7:SDK for PayPal

JSON

Known as JavaScript object notation is a trivial data interchange format making it easy to read and write for users as well as quick and simple for computer machines to parse and create. JSON forms part of the subset of the java script programming language. It is used for the serializing and transmitting of structured data over network connections such as that between server and web application

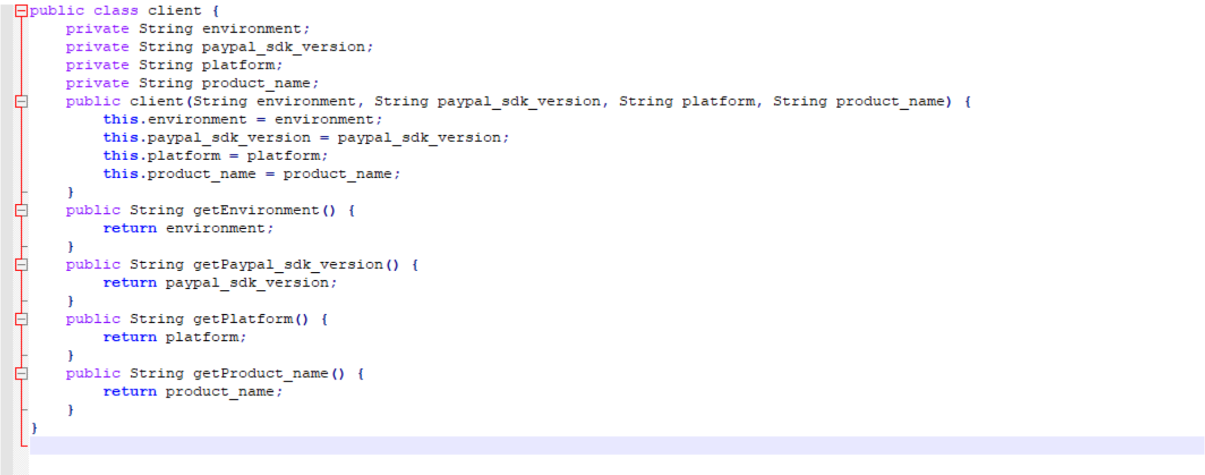
In our application paypal returns a JSON response after the PayPal payment is made



```
{
    "client": {
    "environment": "mock",
    "paypal_sdk_version": "2.14.4",
    "platform": "Android",
    "product_name": "PayPal-Android-SDK"
    },
    "response": {
    "create_time": "2014-07-18T18:46:55Z",
    "id": "PAY-18X32451H0459092JKO7KFUI",
    "intent": "sale",
    "state": "approved"
    },
   "response_type": "payment"
}
```

Figure 8:Json response from Paypal

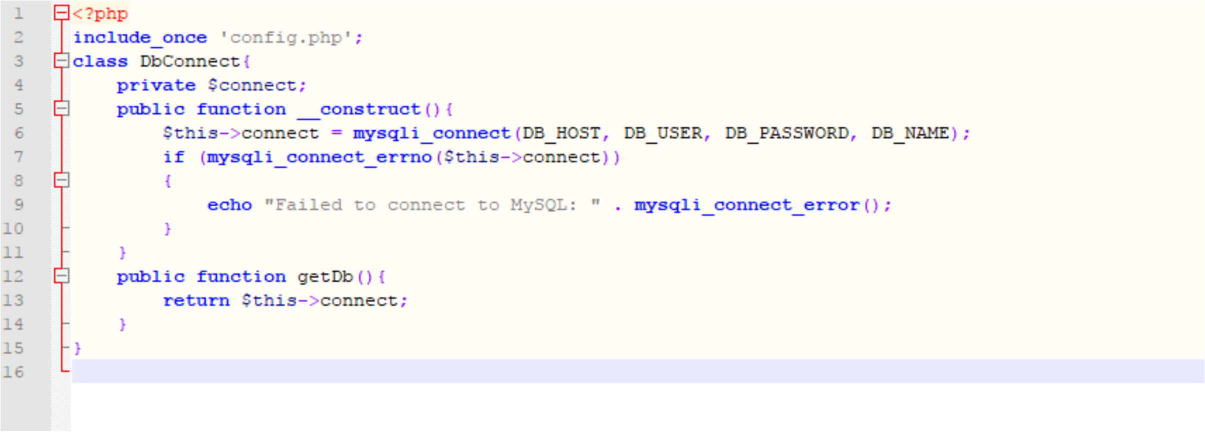
To convert the JSON response into plain java text the following class is used



```java
public class client {
    private String environment;
    private String paypal_sdk_version;
    private String platform;
    private String product_name;
    public client(String environment, String paypal_sdk_version, String platform, String product_name) {
        this.environment = environment;
        this.paypal_sdk_version = paypal_sdk_version;
        this.platform = platform;
        this.product_name = product_name;
    }
    public String getEnvironment() {
        return environment;
    }
    public String getPaypal_sdk_version() {
        return paypal_sdk_version;
    }
    public String getPlatform() {
        return platform;
    }
    public String getProduct_name() {
        return product_name;
    }
}
```

Figure 9:client.java

DB used in MySQLi for PayPal data



```php
<?php
include_once 'config.php';
class DbConnect{
    private $connect;
    public function __construct(){
        $this->connect = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
        if (mysqli_connect_errno($this->connect))
        {
            echo "Failed to connect to MySQL: " . mysqli_connect_error();
        }
    }
    public function getDb(){
        return $this->connect;
    }
}
```

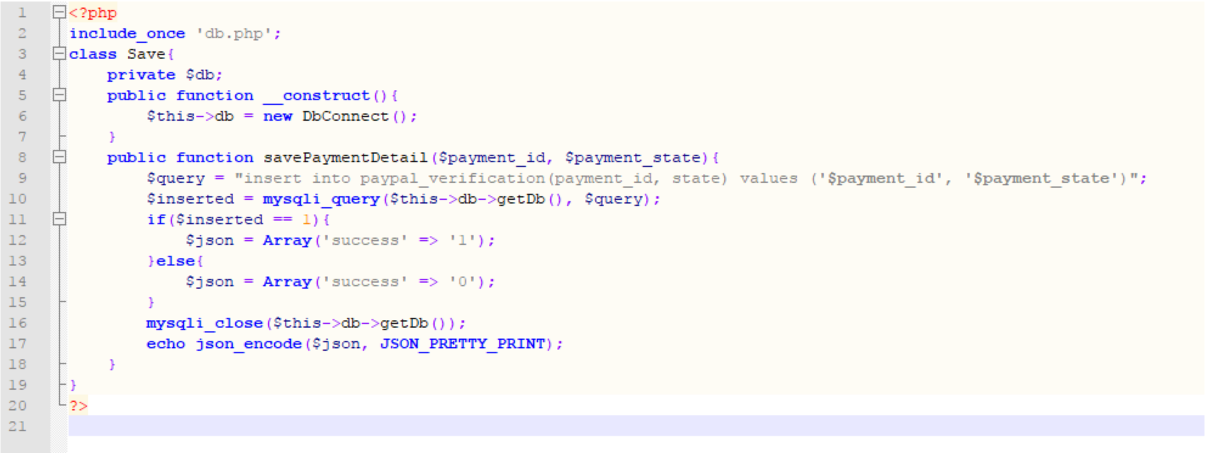
Figure 10 :DB.PHP

PHP

PHP acronym for Hypertext Processor used widely to open source general-purpose scripting language suited for web development and embedded into HTML –Hypertext markup language. Instead of using many commands to output the HTML, PHP pages contains this markup language with embedded code that is actually doing something. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer's

PHP was used throughout development

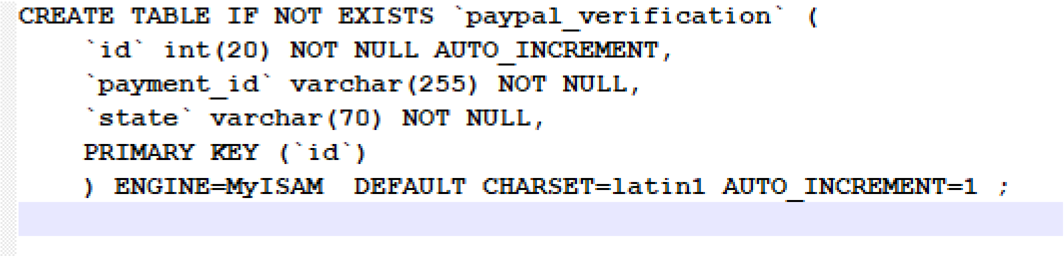
PHP code to write data from PayPal to database



```php
<?php
include_once 'db.php';
class Save{
    private $db;
    public function __construct(){
        $this->db = new DbConnect();
    }
    public function savePaymentDetail($payment_id, $payment_state){
        $query = "insert into paypal_verification(payment_id, state) values ('$payment_id', '$payment_state')";
        $inserted = mysqli_query($this->db->getDb(), $query);
        if($inserted == 1){
            $json = Array('success' => '1');
        }else{
            $json = Array('success' => '0');
        }
        mysqli_close($this->db->getDb());
        echo json_encode($json, JSON_PRETTY_PRINT);
    }
}
?>
```

Figure 11:Save.php

Database creation for paypal verification



```sql
CREATE TABLE IF NOT EXISTS `paypal_verification` (
    `id` int(20) NOT NULL AUTO_INCREMENT,
    `payment_id` varchar(255) NOT NULL,
    `state` varchar(70) NOT NULL,
    PRIMARY KEY (`id`)
    ) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Figure 12:Code to create table for PayPal database

Procedure and requirements of publishing application

As an android compatible application the processes that need to be followed in order to upload the We Sell application to google play can be put out into steps. Google play is our first choice as it provides an easy single platform to distribute, sell, advertise and monitor the sales of the application that is uploaded.

How to become a publisher

The requirements for any application to be distributed via google play include the creation of a developer account which is done using a valid google account. With the use of the developer account one can manage different phases of publishing and distribute an application among all active android users. Furthermore developer program policies need to be understood and accepted which can be found at https://play.google.com/about/developer-distribution-agreement.html. These policies ensure that google play remains a trusted source for applications to android users. If the application is sold to android users via the play store the creation of a merchant account is required otherwise this step can be skipped. Once the required registration fee is paid, a and the account is successfully created one can publish applications via google play.

Complete account information by providing:

- A developer name
- Application title
- Any further required details.

Once this has been submitted one has to wait for approval of the google play developer registration.

Uploading

Once logged into the google play developer console click on, and then follow:

- Add new application option under the all applications tab.
- Choose the desired default language from the drop down menu
- Add the application title.

To upload to a new page select the upload APK option which includes the application homepage. Choose your desired .apk file and then publish it.

Testing

Google play offers the great option of allowing the publisher to test the application before launching it with certain end users. This is called Beta testing whereas

Alpha testing creates a shareable URL address containing the necessary .apk files making it possible to share the application for testing.

Details

Using the store listing tab once all .apk files are uploaded different details of the application can be provided as to give necessary information to end users such as:

- Descriptions
- Screenshots
- Video links
- Contact information
- The category under which the application is classified.

Pricing, distribution and publishing.

If all tabs, APK, store listing, pricing and distribution have been completed successfully the application is ready for publishing. Select the *publish this app* option under the ready to *publish tab*. Once the application has been published it can be changed and edited later on.

Requirements for google play applications

These are guidelines that can be followed to ensure that the application stays published on google play.

Android apps are required to be written in Java by using android APIs, XML for arrangements and layout. The size of the application cannot exceed 100MB and the application should be able to fully use the whole screen portrait or landscape. The application should be tested to minimize crashes, bugs and any abnormal behavior. User interfaces should offer clear text and images and not allow for sound if the screen is off. When basic functionality requirements are not met the application is considered as a beta version.

Work breakdown structure

Salmaan Vally:

1. Create adaptor layout files

Each of the adapters inflates a layout file. The layout files below are associated with the adapter classes and the checkout menu icon.

- Check_layout.xml
- Product_listing.xml
- Shopping_layout.xml

2. Dependencies in build.gradle file

Dependencies needed to make use of third party android libraries

3. Layout files needed for strings,colours and java class to redirect login.

- String.xml
- Colors.xml
- Activity_splash.xml
- Activity_shopping.xml
- SplashActivity.java

Marc kishinkwa

4. Create a RecyclerView adapter that will bind the data source to the View.

- Shoppingactivity.java
- ShopRecycleviewAdapter.java

5. Create product page for shopping cart

- Activity_product.xml

6. Add to cart function

Once this button is clicked it will add the product to the cart. The cart item count increases or decreases if an item is added or removed from the cart.

- Productactivity.java

7. Android paypal integration

- Create sandbox account
- Add paypal SDK
- Paypalcheckoutactivity.java

Adriaan van kraayenburg

8. Create package folder for java helper classes

Holds all the constant variables used in android shopping cart.

- Constants.java
- Mysharedpreference.java

9. Add line break after each item and other decorations

- Simpledivideritemdecoration.java
- Spacesitemdecorator.java

10. Entities package

- Productobject.java

11. Convert JSON enity to java class

- Client.java

Mathapelo

12. Create classes used to make network calls

- Customapplication.java
- Gsonrequest.java
- Volleysingleton.java

13. Convert JSON enity to java class

- Client.java
- Response.java class
- Paymentresponseobject.java
- Serverobject.java

Dinah lee

14. Create php config file to connect to database

- Config.php

15. Create database

- Db.php

- Save.php
- Token.php

16. Database update
17. Server to API call.

What have we learnt?

- usage of android studio
- how to create databases
- XML coding
- Improved java knowledge
- Server to API call
- PHP coding
- Difference between layouts in android studio
- Debugging in android studio
- Use of dependencies
- Extensive XML coding to create mindblowing UI
- How to implement paypal webservice and SDK.

Problems faced?

- exporting MySql database in .db format
- XML coding
- Challenges designing interfaces in android studio
- Connecting to database
- Connecting PayPal SDK
- Server to API
- Usage of dependencies
- Slowness of android studio
- Creation of PHP connect files
- Debugging in android studio

BIBLIOGRAPHY

Duckett, J., 2014. *JavaScript and JQuery: interactive front-end web development*. Wiley Publishing.

Georgiev, M., Jana, S. and Shmatikov, V., 2014, February. Breaking and fixing origin-based access control in hybrid web/mobile application frameworks. In *NDSS symposium* (Vol. 2014, p. 1). NIH Public Access.

Katawetawaraks, C. and Cheng, L.W., 2013. Online shopper behavior: Influences of online shopping decision.

Lei, K., Ma, Y. and Tan, Z., 2014, December. Performance comparison and evaluation of web development technologies in php, python, and node. js. In *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on* (pp. 661-668). IEEE.

Duckett, J., 2014. *JavaScript and JQuery: interactive front-end web development*. Wiley Publishing.