**Mobile Application Development**

T.M.Savindu Pasingtha NG/HNDCSE/24/11



**Assignment Cover Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | | **Module Number and Title** | |
| Higher National Diploma in Computing & Software Engineering(Level 5) | | SED5211 - Mobile Application Development | |
| **Student Name & No.** | | **Assessor** | |
| T.M.Savindu Pasingtha NG/HNDCSE/24/11 | | Mr. Sanaka perera | |
| **Hand out date** | | | **Submission Date** |
| 11.02.2021 | | | 16/05/2021 |
| **Assessment type**  Reports 6000 Words | **Duration/Length of**  **Assessment Type**  End of module | | **Weighting of Assessment**  100% |

|  |
| --- |
| **Learner declaration** |
| I am T.M.Savindu Pasingtha NG/HNDCSE/24/11, certify that the work submitted for this assignment is my own and research sources are fully acknowledged. |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Marks Awarded** | | | |  | | First assessor | |  | |  | | IV marks | |  | |  | | Agreed grade | |  | |  | | Signature of the assessor |  | Date |  |  | |

**Feedback Form**

**International College of Business & Technology**

**Module/Title:** SED5211 - Mobile Application Development

**Student:** T.M. Savindu Pasingtha

**Assessor:**

**Assignment:**

**Strong features of your work:**

**Areas for improvement:**

**Marks Awarded:**

# ACKNOLEDGMENT

In preparation of my assignment, I had to take the help and guidance of some respected persons, who deserve my deepest gratitude. As the completion of this assignment give me much pleasure, I would like to show my gratitude Mr. Sanaka Perera, our lecture, on ICBT Campus for giving me good guidance for assignment throughout numerous consultations. I would also like to expand my gratitude to all those who have directly and indirectly guided me in making this assignment.

# ABSTRACT

This assignment comes under mobile application Development skills for module at Higher National Diploma. This assignment mainly focus, “Lola Cupcakes” to create new Mobile application using to improve their business among the world wide people.

To completing this assignment in the firstly I discussed about the Java Android Rapid application development. Not only that I requirement gather about the Lola cupcakes to develop that mobile application. So successfully cerate that application Before I created and design some diagrams. Such as User Interface, UML, Class diagram, ER diagram and other useful diagrams.

In the second task I completely design proper database with useful SQLite database queries. Not only that I used simple and very attractive user interfaces to that application. So I used control alignments, color thems, images, and appropriate icons to develop that system.

I the third task I completely design full system according to that Business logic. I used several very important classes and methods to complete that system. After I included mobile application to test document. I think this mobile application many help to improve “Lola’s cupcakes” sales in online.

Table of Contents

[ACKNOLEDGMENT 4](#_Toc71993415)

[ABSTRACT 5](#_Toc71993416)

[Task 1 10](#_Toc71993417)

[Diagrams 10](#_Toc71993418)

[ER-DIAGRAM 10](#_Toc71993419)

[UML for Lola’s cupcakes 11](#_Toc71993420)

[Class Diagram for Lola’s cupcakes 12](#_Toc71993421)

[UI-DESIGN 13](#_Toc71993422)

[Task 2 20](#_Toc71993423)

[XML UI for landing page 20](#_Toc71993424)

[XML UI for Single Product Full view 23](#_Toc71993425)

[XML UI for product Manage 24](#_Toc71993426)

[XML UI for Login and SignUp 29](#_Toc71993427)

[XML UI or Cart 30](#_Toc71993428)

[XML UI for All the Product Show 32](#_Toc71993429)

[Task 3 38](#_Toc71993430)

[Database Operations java file 38](#_Toc71993431)

[Product management Java file 46](#_Toc71993432)

[Product model java file 58](#_Toc71993433)

[Product Adater Java file 62](#_Toc71993434)

[LOGIN java file 65](#_Toc71993435)

[SignUp java file 74](#_Toc71993436)

[Cart model java file 82](#_Toc71993437)

[Cart Adapter java file 84](#_Toc71993438)

[Cart java file 88](#_Toc71993439)

[Task 4 96](#_Toc71993440)

[Test Plan 96](#_Toc71993441)

[Test case 1 – Install APK file 98](#_Toc71993442)

[Test - Application user Authentication 98](#_Toc71993443)

[Test - Admin functions 102](#_Toc71993444)

[Test – Product Buy process 113](#_Toc71993445)

[Test case – 9 Payment process to order testing 121](#_Toc71993446)

[User Manual Document 122](#_Toc71993447)

[1 - Sign up or Login 122](#_Toc71993448)

[2 - Product Manage 123](#_Toc71993449)

[3 - Product Buy 124](#_Toc71993450)

[CONCLUTION 125](#_Toc71993451)

[GANTT 126](#_Toc71993452)

# Table Of figure

[Figure 1-ER diagram for Lola's cupcakes 10](#_Toc71993332)

[Figure 2-UML diagram 11](#_Toc71993333)

[Figure 3-Class diagram 12](#_Toc71993334)

[Figure 4-Landing UI 13](#_Toc71993335)

[Figure 5-Login UI 14](#_Toc71993336)

[Figure 6-SignUp UI 15](#_Toc71993337)

[Figure 7-Product View All 16](#_Toc71993338)

[Figure 8-One Product VIew UI 17](#_Toc71993339)

[Figure 9-Cart Items View 18](#_Toc71993340)

[Figure 10-Product Mange UI 19](#_Toc71993341)

[Figure 11-XML UI for landing page 20](#_Toc71993342)

[Figure 12-XML codes for landing page 1 21](#_Toc71993343)

[Figure 13-XML codes for landing page 2 22](#_Toc71993344)

[Figure 14-XML UI for Product Show 23](#_Toc71993345)

[Figure 15-XML UI for Product Manage 24](#_Toc71993346)

[Figure 16-XML code for product manage 1 25](#_Toc71993347)

[Figure 17-XML code for product manage 2 26](#_Toc71993348)

[Figure 18-XML code for product manage 3 27](#_Toc71993349)

[Figure 19-XML code for product manage 4 28](#_Toc71993350)

[Figure 20-XML UI for LOGIN 29](#_Toc71993351)

[Figure 21-XML UI for SignUp 29](#_Toc71993352)

[Figure 22-XML UI for Cat View 30](#_Toc71993353)

[Figure 23-XML UI for Cat Layout 30](#_Toc71993354)

[Figure 24-XML code for cart view 31](#_Toc71993355)

[Figure 25-XML code for Cart LayOut 31](#_Toc71993356)

[Figure 26-XML UI for Product LayOut 32](#_Toc71993357)

[Figure 27-XML UI for All the Product Show 32](#_Toc71993358)

[Figure 28-XML code for Product view 1 33](#_Toc71993359)

[Figure 29-XML code for Product view 2 34](#_Toc71993360)

[Figure 30-XML code for Product view 3 35](#_Toc71993361)

[Figure 31-XML code for Product view 4 36](#_Toc71993362)

[Figure 32-XML code for Product view 5 37](#_Toc71993363)

[Figure 33-sign up test 99](#_Toc71993364)

[Figure 34-login test 101](#_Toc71993365)

[Figure 35-product add test 103](#_Toc71993366)

[Figure 36-product added alert 104](#_Toc71993367)

[Figure 37-product update test 106](#_Toc71993368)

[Figure 38-product update test alert 107](#_Toc71993369)

[Figure 39-product delete 110](#_Toc71993370)

[Figure 40-product delete alert 110](#_Toc71993371)

[Figure 41-products view test 112](#_Toc71993372)

[Figure 42-product add to cart test 114](#_Toc71993373)

[Figure 43-view cart items 116](#_Toc71993374)

[Figure 44-cart item delete test 118](#_Toc71993375)

[Figure 45-empty cart list 119](#_Toc71993376)

# Task 1

## Diagrams

### ER-DIAGRAM



Figure 1-ER diagram for Lola's cupcakes

In this diagram I included, all I identified entities and attributes to create this web application.

So I identify entity names are, customer payment, order product, login and signup, product data, and cart items.

* Order product and customer payment between have many to many relationship.
* Login and customer table between have one to one relationship.
* Cart and product data between have one to one relationship.

### UML for Lola’s cupcakes

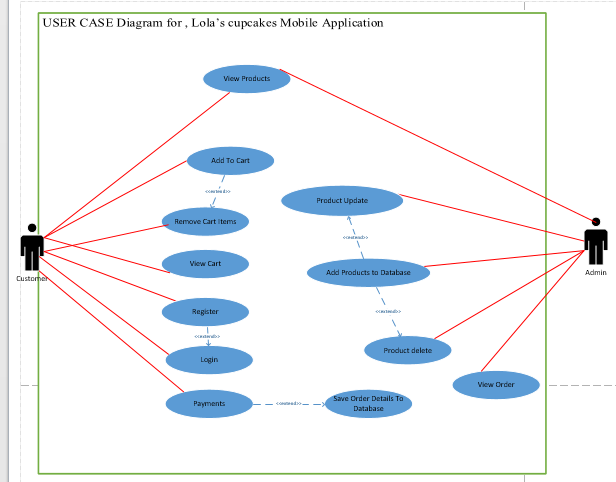


Figure 2-UML diagram

This UML diagram show us to, who the system actors and what they can doing in this system. So we can identify main two actor as customer and system admin. Customer can view products, Add item to card and remove card item, view cart items, do payment with buying product, system login and registration.

Not only that, Admin can manage the system database system. Such as product add, delete, update, view and view customers’ orders etc.

We can identify extends relationship between add to cart and remove cart items user cases. Not only that registration and login between have extend relationship. Add to product user case between and product delete and product update among have extends relationships.

### Class Diagram for Lola’s cupcakes

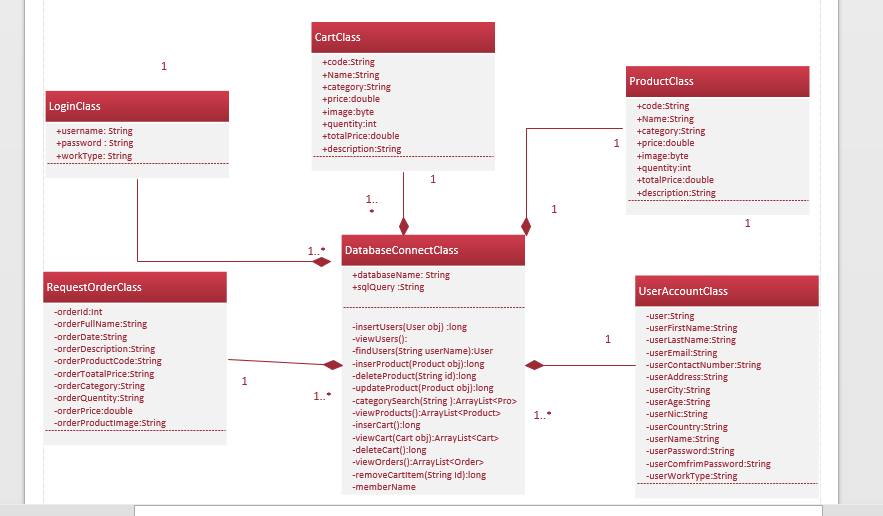


Figure 3-Class diagram

Assumptions

* I thought all DatabseConnectionClass with other all the class have “COMPOSITION” relationship.
* Not only that I thought DatabseConnectionClass multiplicity have “one to many” with other classes.
* DatabseConnectionClass have product insert, cart insert, user registration, product update, product delete, product view, cart all item delete, cart specific item delete, order save , category search and specific id find methods.

Not only that we can find order class, product class, user class, login class add cart class in this class diagram. This classes included all the encapsulated variables. That classes work as models.

### UI-DESIGN

All the User Interfaces Design to used Abobe XD Software.

#### Landig page UI



Figure 4-Landing UI

#### Login UI

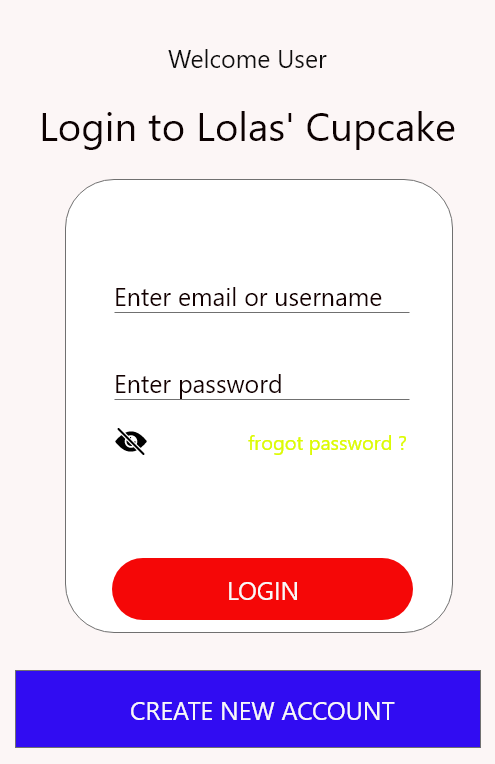


Figure 5-Login UI

#### Sign Up UI

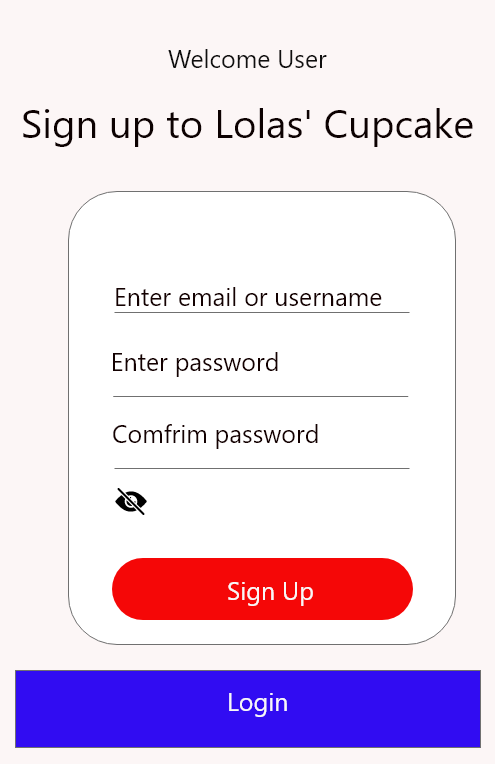


Figure 6-SignUp UI

#### Product Show UI

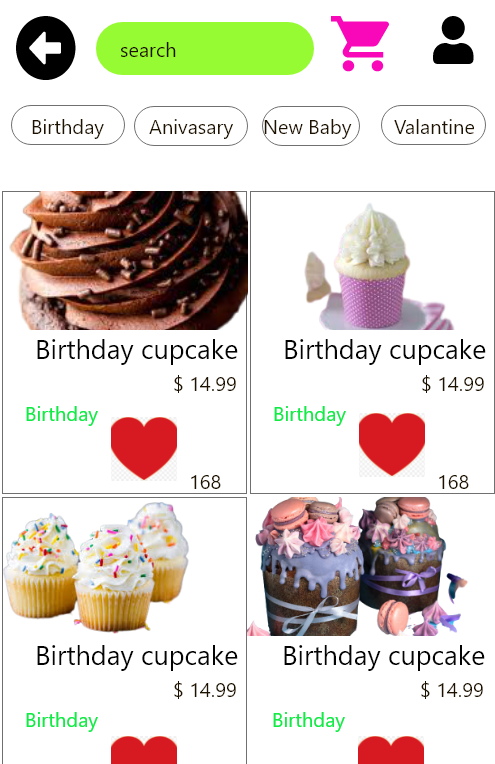


Figure 7-Product View All

#### Product Show UI

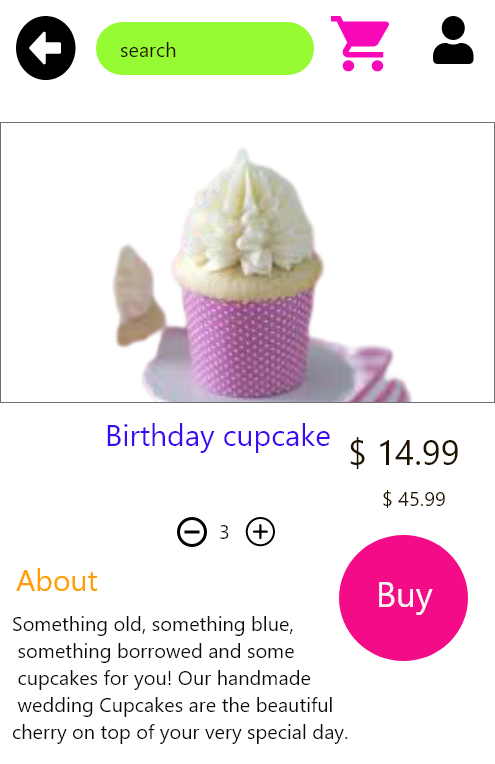


Figure 8-One Product VIew UI

#### Cart

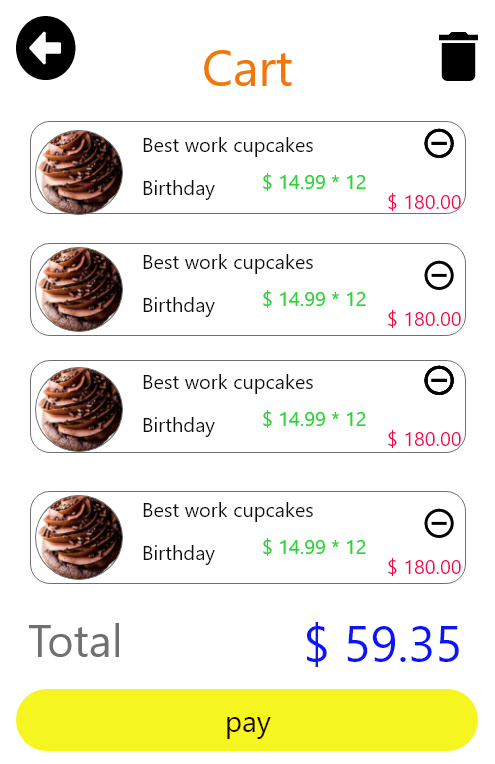


Figure 9-Cart Items View

#### Product Mange UI

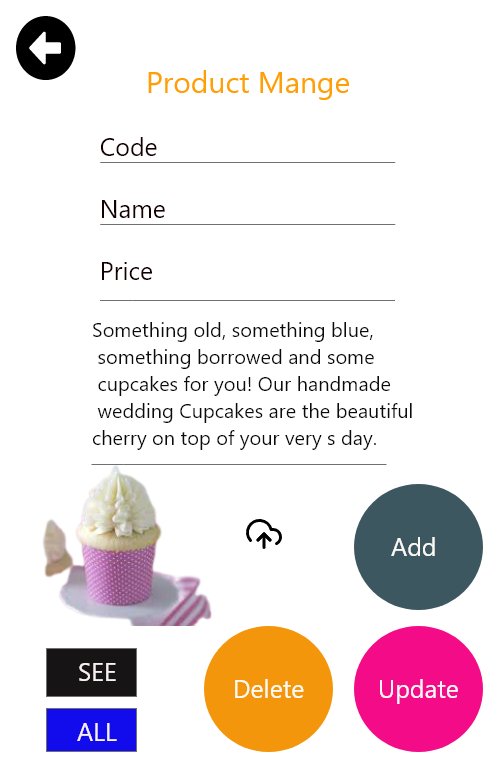


Figure 10-Product Mange UI

# Task 2

Bellow I included all the android XML UI faces in this Mobile application.

## XML UI for landing page



Figure 11-XML UI for landing page

This page include android **Image view , TextView** . All the components include inside the **constraint Layout.**

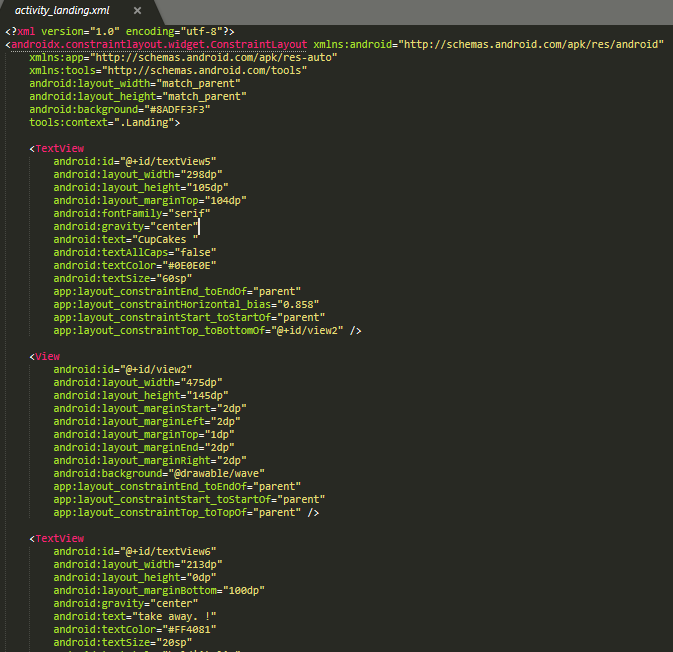


Figure 12-XML codes for landing page 1



Figure 13-XML codes for landing page 2

## XML UI for Single Product Full view

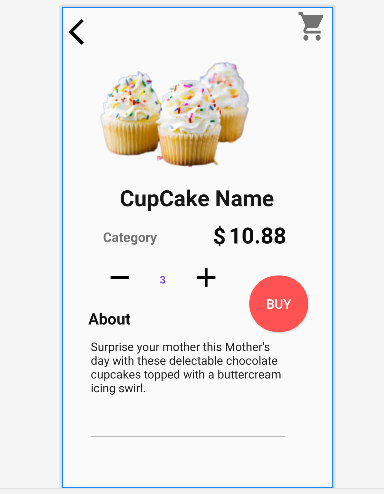


Figure 14-XML UI for Product Show

This page include android **Image view , TextView, Buttons and icons** . All the components include inside the **constraint Layout**. This page to data send by when click the product XML Product, that specific data through the **android intent class**.

## XML UI for product Manage

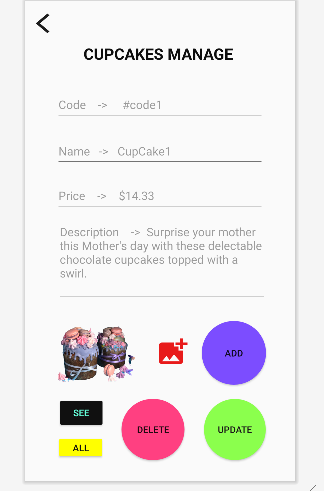


Figure 15-XML UI for Product Manage

This page include android **Image view , Image Browser TextView, EditText, Buttons and icons** . All the components include inside the **constraint Layout**. When click the Add, Update, Delete Buttons Execute the Database Operation methods.

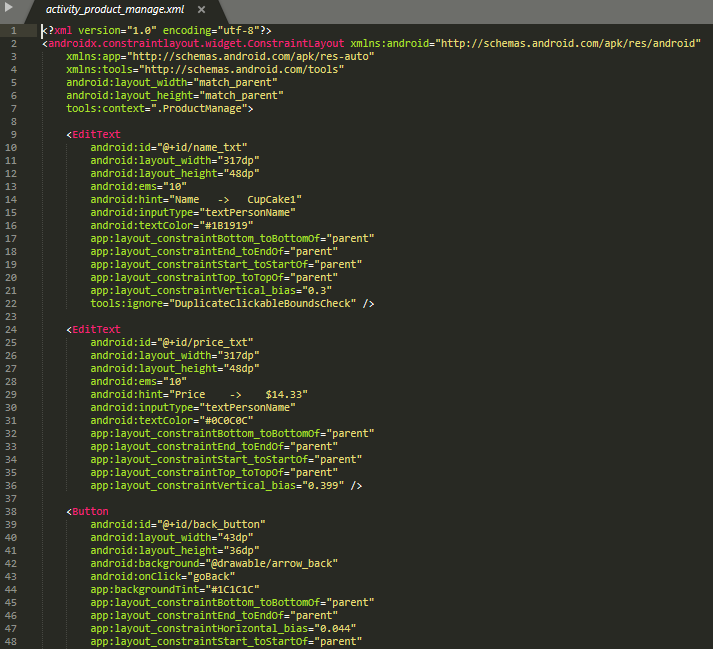


Figure 16-XML code for product manage 1

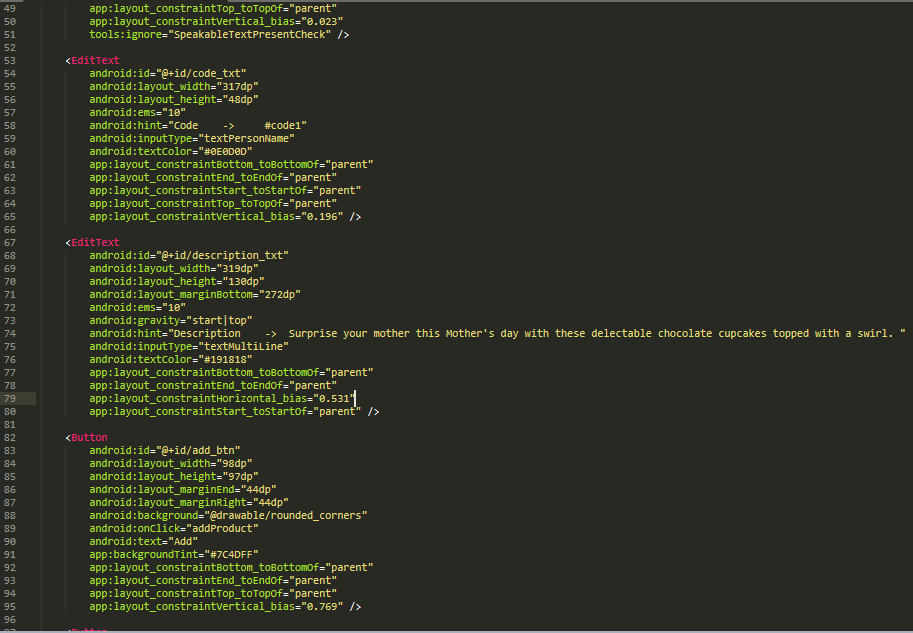


Figure 17-XML code for product manage 2

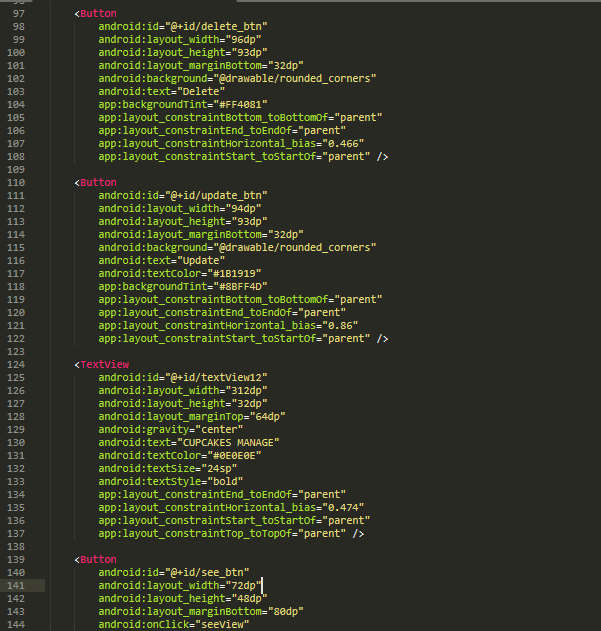


Figure 18-XML code for product manage 3

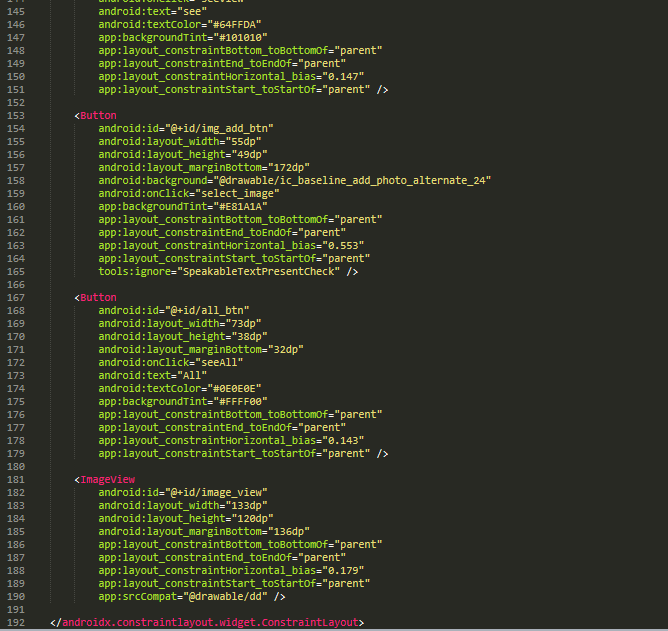


Figure 19-XML code for product manage 4

## XML UI for Login and SignUp

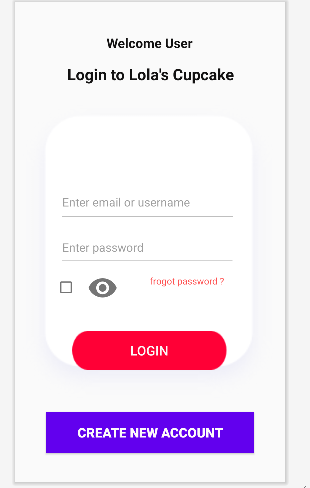
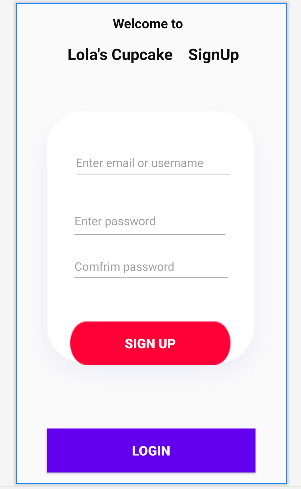
 

Figure 20-XML UI for LOGIN

Figure 21-XML UI for SignUp

This page include android **Image view , TextView, EditText, Buttons and icons** . All the components include inside the **constraint Layout**. When click the Login and SignUp button execute the login and sign upmethods.

## XML UI or Cart

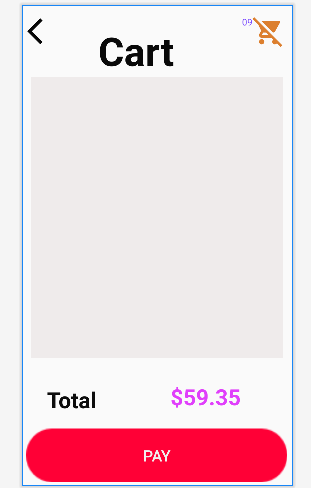
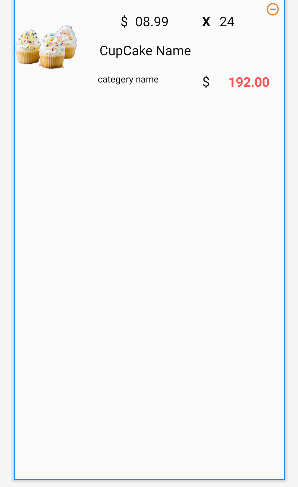
 

Figure 22-XML UI for Cat View

Figure 23-XML UI for Cat Layout

This page include android **List view**, **Image view , TextView, EditText, Buttons and icons** . All the components include inside the **constraint Layout**. User can delete all the cart items and order complete when click the pay button. **Array Adapetr** class using Cart Layout XML render to the Cart View Xml.

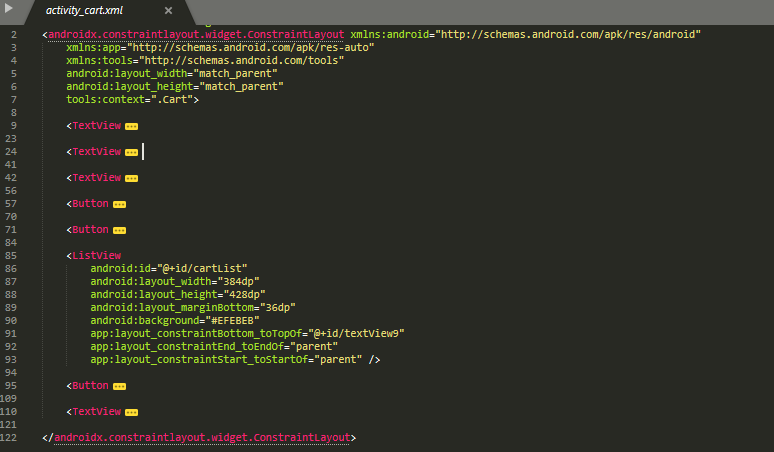


Figure 24-XML code for cart view

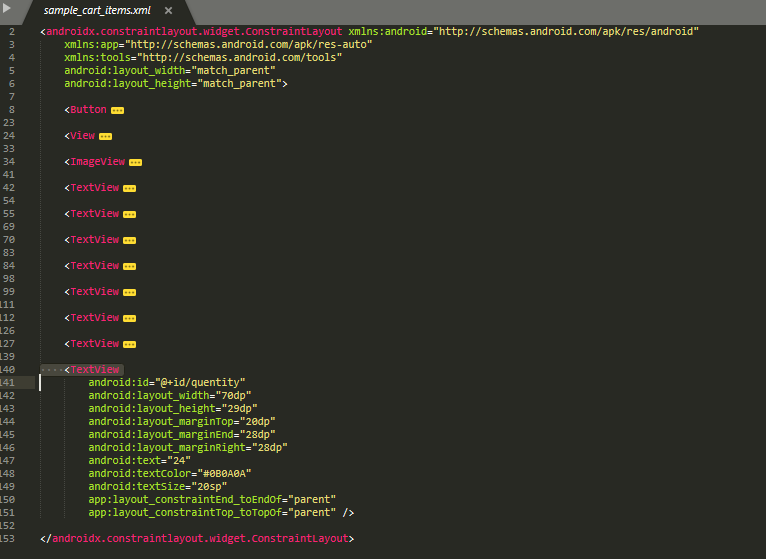


Figure 25-XML code for Cart LayOut

## XML UI for All the Product Show

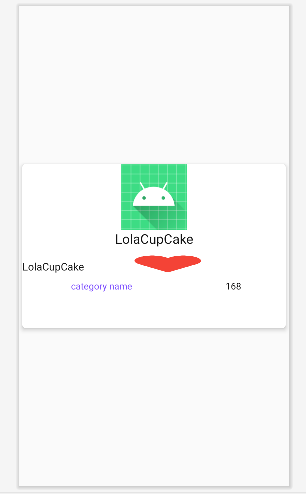
 

Figure 26-XML UI for Product LayOut

Figure 27-XML UI for All the Product Show

This page include android **Gride view, CardView**, **Image view , TextView, EditText, Buttons and icons** .Cart All the components include inside the **constraint Layout** and Layout Xml code include the **Linear Layout**. User can view all the product and select the productto buy. When click the product image auto redirect to the show product xml. **Array Adapetr** class using Product Layout XML render to the Product View Xml Grid View.

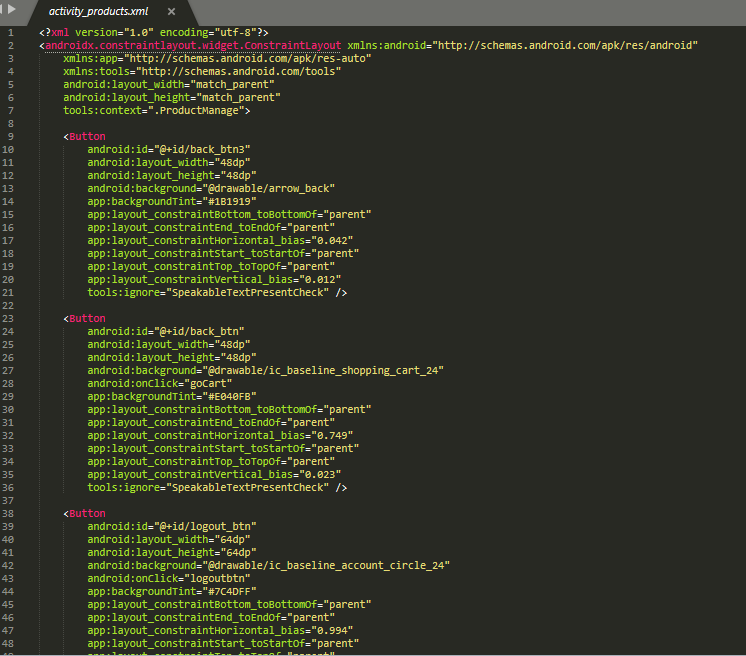


Figure 28-XML code for Product view 1



Figure 29-XML code for Product view 2

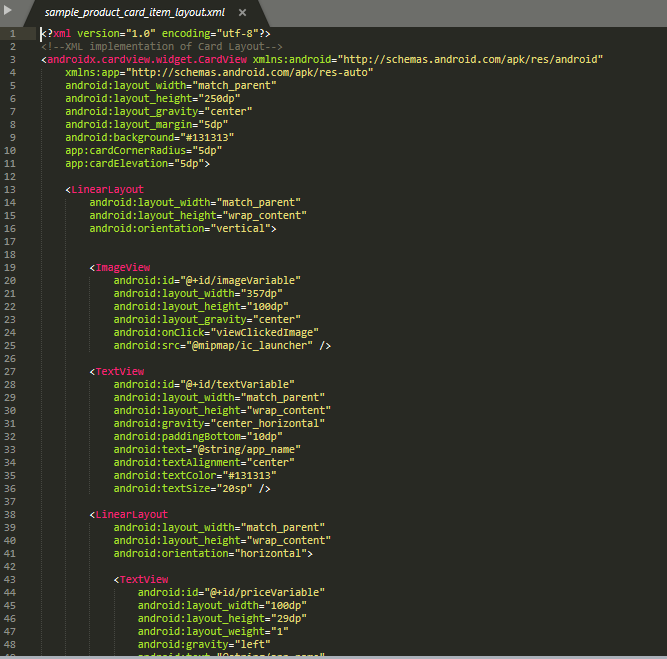


Figure 30-XML code for Product view 3

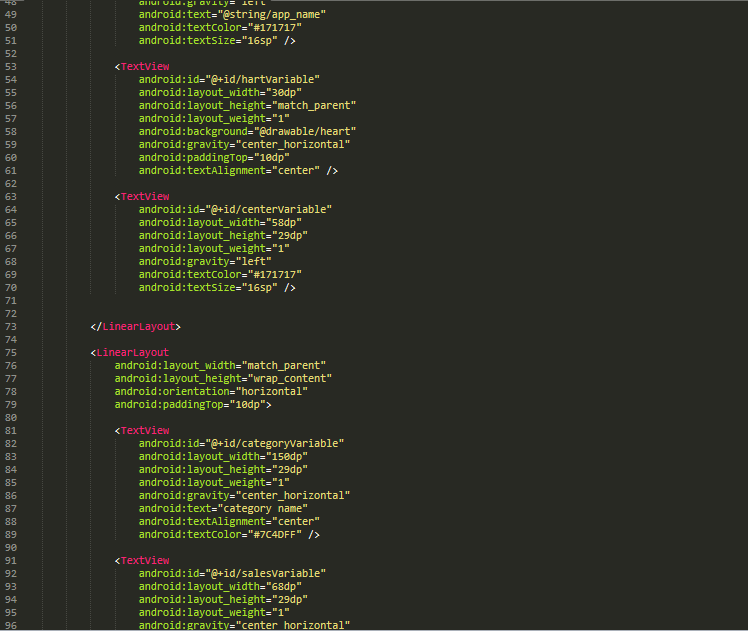


Figure 31-XML code for Product view 4

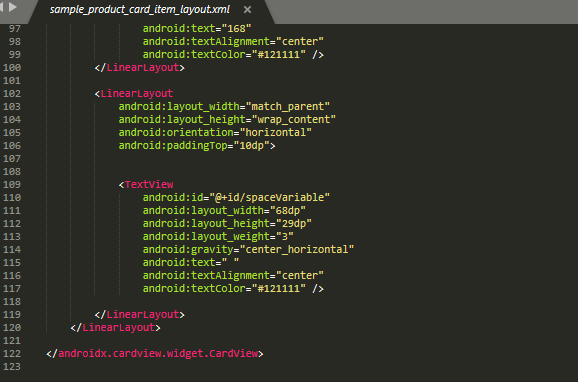


Figure 32-XML code for Product view 5

# Task 3

## Database Operations java file

package com.example.lolacupcake;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.util.Log;

import androidx.annotation.Nullable;

import java.util.ArrayList;

public class DatabaseOperation extends SQLiteOpenHelper{

String sql="";

//1-create database

public DatabaseOperation(@Nullable Context context) {

super(context, "db\_cupcake", null, 1);

}

//2-create table inside database

@Override

public void onCreate(SQLiteDatabase sqLiteDatabase) {

try{

//1- Table 1 create

String sql="CREATE TABLE registrationTable(ID INTEGER PRIMARY KEY AUTOINCREMENT, USERNAME VARCHAR(20), PASSWORD VARCHAR(15))";

sqLiteDatabase.execSQL(sql);

//2-Table 2 create

sql ="CREATE TABLE productTable(CODE VARCHAR(10) PRIMARY KEY,NAME VARCHAR(50),PRICE VARCHAR(10),CATEGORY VARCHAR(25), DESCRIPTION VARCHAR(30), IMG BLOG)";

sqLiteDatabase.execSQL(sql);

//3-Table 3 create

sql ="CREATE TABLE orderTable(ID INTEGER PRIMARY KEY AUTOINCREMENT,CODE VARCHAR(10) PRIMARY KEY,NAME VARCHAR(50),PRICE VARCHAR(10),CATEGORY VARCHAR(25), DESCRIPTION VARCHAR(30), IMG BLOG)";

sqLiteDatabase.execSQL(sql);

//3-Table cart

sql ="CREATE TABLE cartTabel(ID INTEGER PRIMARY KEY AUTOINCREMENT, CODE VARCHAR(10),NAME VARCHAR(50),PRICE VARCHAR(10),CATEGORY VARCHAR(25),QUENTITY VARCHAR(30),TOTAL VARCHAR(10),IMG BLOG)";

sqLiteDatabase.execSQL(sql);

}catch(Exception e){

Log.e("error->",e.getMessage());

}

}

//3-

@Override

public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

try{

//1-delete table 1

String sql ="DROP TABLE IF EXISTS registrationTable";

sqLiteDatabase.execSQL(sql);

//2-delete table 2

sql ="DROP TABLE IF EXISTS productTable";

sqLiteDatabase.execSQL(sql);

}catch(Exception e){

Log.e("error->",e.getMessage());

}

}

//4-Insert Query Execute method insertUserRegistration

public long insertUserRegistration(String username,String password){

long x=0;

try{

SQLiteDatabase obj = getWritableDatabase();

//option 1 execute insert command

ContentValues values = new ContentValues();

values.put("USERNAME",username);

values.put("PASSWORD",password);

x=obj.insert("registrationTable",null,values);//return the integer number 0 or 1

}catch(Exception e){

Log.e("error->",e.getMessage());

}

return x;

}

//4-select Query Execute method findUserRegistration

public int findUserRegistration(int username){

int x=0;

try{

String sql = "SELECT \* FROM registrationTable WHERE ID="+username;

SQLiteDatabase db = getReadableDatabase();

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

if(cursor.getCount()!=0){

userModel obj = new userModel();

obj.setId(cursor.getInt(0));//id auto increment feild

obj.setName(cursor.getString(1));//username colum

obj.setPass(cursor.getString(2));//password colum

x=1;

//if have the username with result

Log.i("h-----------",obj.getName());

}else{

x=0;//if not find result

}

}catch(Exception e){

Log.e("error->",e.getMessage());

}

return x;

}

//5-product insert

//public long insertProduct(PModel p){

public long insertProduct(String c,String n,String ca,String p,String d,byte im[]){

SQLiteDatabase obj = getWritableDatabase();

//option 1 execute insert command

ContentValues values = new ContentValues();

values.put("CODE",c);

values.put("NAME",n);

values.put("PRICE",p);

values.put("CATEGORY",ca);

values.put("DESCRIPTION",d);

values.put("IMG",im);

//return the integer number 0 or 1

return obj.insert("productTable",null,values);

}

//6-update table

public int updateProduct(String name, String price, String code, String category, String description, byte[] img,String columeNameValue){

int x=0;

try{

SQLiteDatabase db = getWritableDatabase();

ContentValues values = new ContentValues();

values.put("CODE",code);

values.put("NAME",name);

values.put("PRICE",price);

values.put("CATEGORY",category);

values.put("DESCRIPTION",description);

values.put("IMG",img);

x=db.update("productTable",values, "CODE="+columeNameValue,null);

}catch(Exception e){

Log.e("error->",e.getMessage());

}

return x;

}

//6-DELETE QUERY

public int delete(String tableName,String columNameWithValue){

int x=0;

try{

SQLiteDatabase db = getWritableDatabase();

x=db.delete(tableName,columNameWithValue,null);

// columeNameWithValue ="col1=10"

}catch(Exception e){

Log.e("error->",e.getMessage());

}

return x;

}

public ArrayList<userModel> viewAllUsers(){

ArrayList<userModel> data = new ArrayList<userModel>();

try{

SQLiteDatabase db= getReadableDatabase();

String sql = "SELECT \* FROM registrationTable";

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

int count = cursor.getCount();

//data[ 1,2,3]

if(count !=0){

while (cursor.moveToNext()){

userModel obj = new userModel();//set valus to array list

obj.setId(cursor.getInt(0));//id auto increment feild

obj.setName(cursor.getString(1));//username colum

obj.setPass(cursor.getString(2));//password colum

data.add(obj);//object value add as array list index

}

}

}catch(Exception e){

Log.e("error->",e.getMessage());

}

return data;

}

public ArrayList<PModel> viewAllProducts(){

ArrayList<PModel> productList = new ArrayList<PModel>();

try{

SQLiteDatabase db = getReadableDatabase();

String sql = "SELECT \* FROM productTable";

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

int count = cursor.getCount();

if(count!=0){

while (cursor.moveToNext()) {

PModel obj = new PModel();

obj.setPrdId(cursor.getString(0));

obj.setpName(cursor.getString(1));

obj.setpPrice(cursor.getString(2));

obj.setpCategory(cursor.getString(3));

obj.setDesc(cursor.getString(4));

obj.setImg(cursor.getBlob(5));

productList.add(obj);

Log.i("product view code -> ",""+ cursor.getString(0));

}

}else{

Log.i("product Error view -> ","ERROR");

}

}catch(Exception e){

Log.e("error->",e.getMessage());

}

return productList;

}

//8-Insert Query Execute method insertCart Item

public long insertCart(String total,String code){

long x=0;

try{

CartModel c = new CartModel();

SQLiteDatabase obj = getWritableDatabase();

//option 1 execute insert command

ContentValues values = new ContentValues();

values.put("NAME",c.getName());

values.put("CATEGORY",c.getCategiry());

values.put("IMG",c.getImgId());

values.put("PRICE",c.getPrice());

values.put("QUENTITY",c.getQuentity());

values.put("TOTAL",total);

values.put("CODE",code);

x=obj.insert("registrationTable",null,values);//return the integer number 0 or 1

}catch(Exception e){

Log.e("error->",e.getMessage());

}

return x;

}

}

## Product management Java file

package com.example.lolacupcake;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.graphics.Bitmap;

import android.net.Uri;

import android.os.Bundle;

import android.provider.MediaStore;

import android.text.Editable;

import android.text.TextWatcher;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import java.io.ByteArrayOutputStream;

import java.io.IOException;

public class ProductManage extends AppCompatActivity {

Button back\_button,see\_btn,add\_btn,update\_btn,delete\_btn,all\_btn;

EditText code\_txt,price\_txt,description\_txt,name\_txt,category\_txt;

ImageView image\_view;

byte imageByte[];

/\*----Database----\*/

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_product\_manage);

/\*----------------------\*/

back\_button=(Button)findViewById(R.id.back\_button);

see\_btn=(Button)findViewById(R.id.see\_btn);

add\_btn=(Button)findViewById(R.id.add\_btn);

delete\_btn=(Button)findViewById(R.id.delete\_btn);

update\_btn=(Button)findViewById(R.id.update\_btn);

/\*----------------------\*/

code\_txt=(EditText)findViewById(R.id.code\_txt);

price\_txt=(EditText)findViewById(R.id.price\_txt);

description\_txt=(EditText)findViewById(R.id.description\_txt);

name\_txt=(EditText)findViewById(R.id.name\_txt);

/\*---------------------\*/

image\_view=(ImageView)findViewById(R.id.image\_view);

//text watcher

code\_txt.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

if(!code\_txt.getText().toString().isEmpty()){

add\_btn.setEnabled(true);

update\_btn.setEnabled(true);

delete\_btn.setEnabled(true);

see\_btn.setEnabled(true);

}else{

add\_btn.setEnabled(false);

update\_btn.setEnabled(false);

delete\_btn.setEnabled(false);

see\_btn.setEnabled(false);

}

}

@Override

public void afterTextChanged(Editable editable) {

}

});

price\_txt.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

if(!price\_txt.getText().toString().isEmpty()){

add\_btn.setEnabled(true);

update\_btn.setEnabled(true);

see\_btn.setEnabled(true);

}else{

add\_btn.setEnabled(false);

update\_btn.setEnabled(false);

see\_btn.setEnabled(false);

}

}

@Override

public void afterTextChanged(Editable editable) {

}

});

description\_txt.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

if(!description\_txt.getText().toString().isEmpty()){

add\_btn.setEnabled(true);

update\_btn.setEnabled(true);

see\_btn.setEnabled(true);

}else{

add\_btn.setEnabled(false);

update\_btn.setEnabled(false);

see\_btn.setEnabled(false);

}

}

@Override

public void afterTextChanged(Editable editable) {

}

});

name\_txt.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

if(!name\_txt.getText().toString().isEmpty()){

add\_btn.setEnabled(true);

update\_btn.setEnabled(true);

see\_btn.setEnabled(true);

}else{

add\_btn.setEnabled(false);

update\_btn.setEnabled(false);

see\_btn.setEnabled(false);

}

}

@Override

public void afterTextChanged(Editable editable) {

}

});

}

public void goBack(View view){

Intent obj = new Intent(ProductManage.this,ShowProduct.class);

startActivity(obj);

}

public void seeAll(View view){

Intent obj = new Intent(ProductManage.this,Products.class);

startActivity(obj);

}

public void seeView(View view){

//data send to another page XML VIEW

try{

Intent obj = new Intent(ProductManage.this,ShowProduct.class);

if(name\_txt.getText().toString().length()!=0){

obj.putExtra("code",code\_txt.getText().toString());

obj.putExtra("name",name\_txt.getText().toString());

obj.putExtra("price",price\_txt.getText().toString());

obj.putExtra("description",description\_txt.getText().toString());

startActivity(obj);

}

//alertMethod("work ");

}catch(Exception e){

Log.e("error seeView()",e.getMessage().toString());

}

}

public void alertMethod(String title,String msg){

try{

//1-builder to a set alert values

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);

alertDialogBuilder.setTitle(title);

alertDialogBuilder.setMessage(msg);

//alertDialogBuilder.setMessage("Login successfully.");

//2- execute alert dialog

AlertDialog alertDialog = alertDialogBuilder.create();

alertDialog.show();

}catch(Exception e){

Log.e("error alertMethod()",e.getMessage().toString());

}

}

/\*----select image ---\*/

public void select\_image(View view){

Intent intent = new Intent();

intent.setType("image/\*");

intent.setAction(Intent.ACTION\_GET\_CONTENT);

intent.putExtra("crop","true");

intent.putExtra("aspectX",0);

intent.putExtra("aspectY",0);

intent.putExtra("outputX",150);

intent.putExtra("outputY",150);

intent.putExtra("return-data","true");

startActivityForResult(Intent.createChooser(intent,"SELECT IMAGE"),111);

}

/\*-----Selected image convert to byte[] view in ImageView-----\*/

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if(requestCode == 111){

if( data!= null){

Uri uri = data.getData();

try{

Bitmap bitmap = MediaStore.Images.Media.getBitmap(this.getContentResolver(),uri);

ByteArrayOutputStream arrayOutputStream = new ByteArrayOutputStream();

bitmap.compress(Bitmap.CompressFormat.PNG,0, arrayOutputStream);

imageByte = arrayOutputStream.toByteArray();

image\_view.setImageBitmap(bitmap);//front end view image

}catch(IOException e){

Log.e("Error image save : ", e.getMessage());

}

}

}

}

/\*----------product add-----\*/

public void addProduct(View view){

PModel productObj =new PModel();

String c=code\_txt.getText().toString();

String n=name\_txt.getText().toString();

String ca=category\_txt.getText().toString();

String p=price\_txt.getText().toString();

String d=description\_txt.getText().toString();

Log.e("out",c+n+ca+p+d);

byte im[]=imageByte;

try {

productObj.setpPrice(p);

productObj.setpCategory(ca);

productObj.setpName(n);

productObj.setDesc(d);

productObj.setImg(im);

if(!c.isEmpty()) {

DatabaseOperation dbObj=new DatabaseOperation(this);

if(dbObj.insertProduct(c,n,ca,p,d,im) > 0) {

alertMethod("Product Insert", "Successfully.");

Log.e("If product\_insert() ", "Added");

claer();

} else {

alertMethod("Product Insert", "Failed !.");

Log.e("Else1 product\_insert() ", "Failed");

}

}else{

alertMethod("", "Please enter values or Image select !.");

Log.e("Else2 product\_insert() ", "No image select");

}

}catch(Exception e){

Log.e("Catch product\_insert() ", e.getMessage());

claer();

}

}

public void claer(){

code\_txt.setText("");

name\_txt.setText("");

price\_txt.setText("");

description\_txt.setText("");

}

}

## Product model java file

package com.example.lolacupcake;

public class PModel {

private String prdId;

private String pName;

private String pPrice;

private String pCategory;

private String desc;

private byte[] img;

public String getPrdId() {

return prdId;

}

public void setPrdId(String prdId) {

this.prdId = prdId;

}

public String getpName() {

return pName;

}

public void setpName(String pName) {

this.pName = pName;

}

public String getpPrice() {

return pPrice;

}

public void setpPrice(String pPrice) {

this.pPrice = pPrice;

}

public String getpCategory() {

return pCategory;

}

public void setpCategory(String pCategory) {

this.pCategory = pCategory;

}

public String getDesc() {

return desc;

}

public void setDesc(String desc) {

this.desc = desc;

}

public byte[] getImg() {

return img;

}

public void setImg(byte[] img) {

this.img = img;

}

/\*

private String ppName;

private int imgId;

private String price;

\*/

// private byte[] img;

/\*

public PModel(String pName,String price, int imgID) {

this.imgId=imgID; this.ppName=pName; this.price=price;

}

public int getImgId() {

return imgId;

}

public void setImgId(int imgId) {

this.imgId = imgId;

}

public String getPpName() {

return ppName;

}

public void setPpName(String ppName) {

this.ppName = ppName;

}

public String getPrice() {

return price;

}

public void setPrice(String price) {

this.price = price;

}

}

## Product Adater Java file

package com.example.lolacupcake;

import android.content.Context;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ArrayAdapter;

import android.widget.ImageView;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import java.util.ArrayList;

public class ProductGrideViewAdapter extends ArrayAdapter<PModel> {

public ProductGrideViewAdapter(@NonNull Context context, ArrayList<PModel> ProductPModelArrayList) {

super(context,0, ProductPModelArrayList);

}

@NonNull

@Override

public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {

View listitemView = convertView;

if (listitemView == null) {

// Layout Inflater inflates each item to be displayed in GridView.

listitemView = LayoutInflater.from(getContext()).inflate(R.layout.sample\_product\_card\_item\_layout, parent, false);

}

PModel pModel = getItem(position);

TextView textVariable = listitemView.findViewById(R.id.textVariable);

ImageView imageVariable = listitemView.findViewById(R.id.imageVariable);

TextView priceText=listitemView.findViewById(R.id.priceVariable);

textVariable.setText(pModel.getpName());

//imageVariable.setImageResource(pModel.getImgId());

Bitmap bitmap = BitmapFactory.decodeByteArray(pModel.getImg(),0,pModel.getImg().length);

imageVariable.setImageBitmap(bitmap);

priceText.setText(pModel.getpPrice());

// imageVariable.setImageResource(obj.getImgid());

//select krpu image ike index ek access krnn

imageVariable.setTag(position);

return listitemView;

}

}

## LOGIN java file

package com.example.lolacupcake;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Bundle;

import android.text.Editable;

import android.text.TextWatcher;

import android.text.method.HideReturnsTransformationMethod;

import android.text.method.PasswordTransformationMethod;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.CheckBox;

import android.widget.CompoundButton;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import java.util.ArrayList;

public class Login extends AppCompatActivity {

//define ui componets id names as a variable

EditText usernamevalue,passwordvalue;

TextView lblname,lblpass,lbl\_error;

Button btnLogin ,showHiddenBtn;

CheckBox ck;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

//input feilds id values access

usernamevalue=(EditText)findViewById(R.id.username);

passwordvalue=(EditText)findViewById(R.id.password);

btnLogin=(Button) findViewById(R.id.login\_btn);

// showHiddenBtn=(Button) findViewById(R.id.showHiddenBtn);

ck=(CheckBox) findViewById(R.id.checkBox);

// lblname = (TextView) findViewById(R.id.lblname);

// lblpass = (TextView) findViewById(R.id.lblpass);

lbl\_error=(TextView)findViewById(R.id.lbl\_error);

//text watcher class

usernamevalue.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

if (!usernamevalue.getText().toString().isEmpty() && !passwordvalue.getText().toString().isEmpty()) {

btnLogin.setEnabled(true);

} else {

btnLogin.setEnabled(false);

}

}

@Override

public void afterTextChanged(Editable editable) {

}

});

passwordvalue.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

if(!usernamevalue.getText().toString().isEmpty() && !passwordvalue.getText().toString().isEmpty()){

btnLogin.setEnabled(true);

}else{

btnLogin.setEnabled(false);

}

}

@Override

public void afterTextChanged(Editable editable) {

}

});

}

public void showHidePassword(View view){

/\*

ck.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {

@Override

public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {

if(isChecked){

passwordvalue.setTransformationMethod(null);

}else{

passwordvalue.setTransformationMethod(new PasswordTransformationMethod());

}

}

});

\*/

if(ck.isChecked()){

passwordvalue.setTransformationMethod(null);

} else{

passwordvalue.setTransformationMethod(new PasswordTransformationMethod());

}

}

public void login\_btn\_click(View view){

String u="Abc";

String p="123";

String user = usernamevalue.getText().toString();

String pass = passwordvalue.getText().toString();

try{

DatabaseOperation db=new DatabaseOperation(this);

// db.findUserRegistration(2);

ArrayList<userModel> list=db.viewAllUsers();

ArrayList<String> arr=new ArrayList<>();

for(int x=0; x<list.size(); x++){

userModel o=list.get(x);

// arr.add(o.getId()+);

Log.i("val->",list.get(x).toString());

}

final ProgressDialog pd = new ProgressDialog(Login.this);

pd.setTitle("Login");

pd.setMessage("Authenticating...");

pd.setProgressStyle(ProgressDialog.STYLE\_SPINNER);

pd.show();

//asyncronous method ekk time out set kenn

new android.os.Handler().postDelayed(new Runnable() {@Override public void run()

{ pd.dismiss();

Log.i("error in Login","6");

userModel obj=new userModel();

if((user.equalsIgnoreCase(u) || user.equalsIgnoreCase(obj.getName())) && (pass.equals(obj.getPass()) || p.equals(pass))){

alertMethod("Login Successfully.");

clear();// claer the the text feils

//go to new xml

Intent objc = new Intent(Login.this,Products.class);

startActivity(objc);

}else if((user.equalsIgnoreCase("admin") && pass.equalsIgnoreCase("admin")))

{

alertMethod("Login Successfully to Admin panel.");

clear();// claer the the text feils

//go to new xml

Intent objc = new Intent(Login.this,ProductManage.class);

startActivity(objc);

}else{

// alertMethod("Login failed !.");

lbl\_error.setText("password length should be 8 or not match !");

}

}},2000);

}catch(Exception e){

Log.i("error in Login",e.getMessage());

clear();

}

}

public void gotoSignup(View view){

Intent obj = new Intent(Login.this,Signup.class);

startActivity(obj);

}

public void alertMethod(String msg){

//1-builder to a set alert values

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);

alertDialogBuilder.setTitle("");

alertDialogBuilder.setMessage(msg);

//alertDialogBuilder.setMessage("Login successfully.");

//2- execute alert dialog

AlertDialog alertDialog = alertDialogBuilder.create();

alertDialog.show();

}

public void clear(){

usernamevalue.setText(null);

passwordvalue.setText(null);

lbl\_error.setText(null);

}

}

## SignUp java file

package com.example.lolacupcake;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Bundle;

import android.text.Editable;

import android.text.TextWatcher;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

public class Signup extends AppCompatActivity {

//define ui componets id names as a variable

EditText usernamevalue,passwordvalue,comfrimpasswordvalue;

TextView lbl\_error,one,two,three;

Button btnSignUp;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_signup);

//input feilds id values access

usernamevalue=(EditText)findViewById(R.id.username\_reg);

passwordvalue=(EditText)findViewById(R.id.password\_reg);

comfrimpasswordvalue=(EditText)findViewById(R.id.comfrimpassword\_reg);

btnSignUp=(Button) findViewById(R.id.signup\_btn);

lbl\_error=(TextView)findViewById(R.id.lbl\_error);

one=(TextView)findViewById(R.id.one);

two=(TextView)findViewById(R.id.two);

three=(TextView)findViewById(R.id.three);

//text watcher

usernamevalue.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

if(!usernamevalue.getText().toString().isEmpty() && !passwordvalue.getText().toString().isEmpty()){

btnSignUp.setEnabled(true);

}else{

btnSignUp.setEnabled(false);

}

}

@Override

public void afterTextChanged(Editable editable) {

}

});

passwordvalue.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

if(!usernamevalue.getText().toString().isEmpty() && !passwordvalue.getText().toString().isEmpty()){

btnSignUp.setEnabled(true);

}else{

btnSignUp.setEnabled(false);

}

}

@Override

public void afterTextChanged(Editable editable) {

}

});

comfrimpasswordvalue.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

if(!usernamevalue.getText().toString().isEmpty() && !comfrimpasswordvalue.getText().toString().isEmpty()){

btnSignUp.setEnabled(true);

}else{

btnSignUp.setEnabled(false);

}

}

@Override

public void afterTextChanged(Editable editable) {

}

});

}

public void signUp\_btn\_click(View view){

String user = usernamevalue.getText().toString();

String pass = passwordvalue.getText().toString();

String comfrim = comfrimpasswordvalue.getText().toString();

try{

one.setText(user);

two.setText(pass);

three.setText(comfrim);

//progerss loading

final ProgressDialog pd = new ProgressDialog(Signup.this);

pd.setTitle("SignUp");

pd.setMessage("Authenticating...");

pd.setProgressStyle(ProgressDialog.STYLE\_SPINNER);

pd.show();

//asyncronous method ekk time out set kenn

new android.os.Handler().postDelayed(new Runnable() {@Override public void run() { pd.dismiss();

if( !user.isEmpty() && pass.equals(comfrim) && (pass.length()==8)){

saveUserRegistrations(user,pass);//saved db

gotologin(view);

clear();//clearthe text feils

}else{

lbl\_error.setText("password length should be 8 or not match !");

}

}},2000);

}catch(Exception e) {

Log.i("error in Login",e.getMessage());

clear();//clearthe text feilds

}

}

public void gotologin(View view){

Intent obj = new Intent(Signup.this,Login.class);

startActivity(obj);

}

public void alertMethod(String msg){

//1-builder to a set alert values

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);

alertDialogBuilder.setTitle("");

alertDialogBuilder.setMessage(msg);

//alertDialogBuilder.setMessage("Login successfully.");

//2- execute alert dialog

AlertDialog alertDialog = alertDialogBuilder.create();

alertDialog.show();

}

public void clear(){

usernamevalue.setText(null);

passwordvalue.setText(null);

comfrimpasswordvalue.setText(null);

lbl\_error.setText(null);

}

@SuppressLint("LongLogTag")

public void saveUserRegistrations(String username,String password){

try{

DatabaseOperation db=new DatabaseOperation(this);

if(db.insertUserRegistration(username,password)>0){

alertMethod("Registration Successfully Completed.");

}else{

alertMethod("Registration Failed !.");

}

}catch(Exception e){

Log.e("Error saveUserReg()->", e.getMessage());

}

}

}

## Cart model java file

package com.example.lolacupcake;

public class CartModel {

private String name,categiry,price,quentity;

private int imgId;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCategiry() {

return categiry;

}

public void setCategiry(String categiry) {

this.categiry = categiry;

}

public String getPrice() {

return price;

}

public void setPrice(String price) {

this.price = price;

}

public String getQuentity() {

return quentity;

}

public void setQuentity(String quentity) {

this.quentity = quentity;

}

public int getImgId() {

return imgId;

}

public void setImgId(int imgId) {

this.imgId = imgId;

}

}

## Cart Adapter java file

package com.example.lolacupcake;

import android.content.Context;

import android.content.DialogInterface;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import java.util.ArrayList;

public class CartItemAdapter extends BaseAdapter {

Context context;

TextView name,category,price,quentity,total;

ImageView imageView;

Button removeItemBtn;

ArrayList<CartModel> cartArrayList;

//constructor method 1

public CartItemAdapter(Context context, ArrayList<CartModel> list){

this.context = context;

this. cartArrayList = list;

}

@Override

public int getCount() {

return cartArrayList.size();

}

@Override

public Object getItem(int position) {

return cartArrayList.get(position);

}

@Override

public long getItemId(int position) {

return 0;

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

//externam lay out access

LayoutInflater inflater=(LayoutInflater)context.getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);

//mokakd view kal yutu lay out ek set kala

View v = inflater.inflate(R.layout.sample\_cart\_items, parent, false);

//external layout variable access and define

name=(TextView)v.findViewById(R.id.name);

category=(TextView)v.findViewById(R.id.category);

price=(TextView)v.findViewById(R.id.price);

quentity=(TextView)v.findViewById(R.id.quentity);

total=(TextView)v.findViewById(R.id.total);

imageView = (ImageView)v.findViewById(R.id.img);

removeItemBtn=(Button)v.findViewById(R.id.removeItemBtn);

// CartModel cartModel=new CartModel();

CartModel cartModel = cartArrayList.get(position);//arraylist index value

//external lay out variables to values set in array inside value assign

name.setText(cartModel.getName());

category.setText(cartModel.getCategiry());

price.setText(cartModel.getPrice());

quentity.setText(cartModel.getQuentity());

imageView.setImageResource(cartModel.getImgId());

//conversation

double p= Double.valueOf(price.getText().toString());

double q= Double.valueOf(quentity.getText().toString());

total.setText(String.valueOf(p\*q));

//selected bton index value access and pass to the click event

removeItemBtn.setTag(position);

// Bitmap bitmap = BitmapFactory.decodeByteArray(product.getImg(),0,product.getImg().length);

// imageView.setImageBitmap(bitmap);

return v;

}

}

## Cart java file

package com.example.lolacupcake;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.Gravity;

import android.view.View;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import java.util.ArrayList;

public class Cart extends AppCompatActivity {

ListView listView;

TextView cartNoOfItems,totalP;

ArrayList<CartModel> cartArrayList = new ArrayList<CartModel>();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_cart);

//1

listView=(ListView)findViewById(R.id.cartList);

cartNoOfItems=(TextView)findViewById(R.id.noitems);

totalP=(TextView)findViewById(R.id.totalP);

try{

// ArrayList<PModel> ProductModelArrayList = new ArrayList<PModel>();

// ArrayList<CartModel> cartArrayList = new ArrayList<CartModel>();

for(int i=0; i<20; i++){

//assign data inset model

CartModel obj=new CartModel();

obj.setCategiry("cat1");

obj.setName("name1-"+String.valueOf(i\*5));

obj.setPrice(String.valueOf(i\*10));

obj.setQuentity(String.valueOf(i\*5));

obj.setImgId(R.drawable.aa);

//save add to list to add model obj

cartArrayList.add(obj);

}

//view arrary list data

if(cartArrayList.size()>0){

CartItemAdapter adapeter = new CartItemAdapter(this,cartArrayList);

listView.setAdapter( adapeter);

//shopin cart items set

cartNoOfItems.setText(String.valueOf(cartArrayList.size()));

try{

//conversation

double p,q,t=0;

for(int i=0; i<cartArrayList.size(); i++){

CartModel cartModel = cartArrayList.get(i);//arraylist index value objext get and asign

//conversation

p=Double.valueOf(cartModel.getPrice().toString());

q=Double.valueOf(cartModel.getQuentity().toString());

t=t+(p\*q);

}

totalP.setText(String.valueOf(t));

}catch (Exception e){

Log.e("Error in Cart",e.getMessage());

}

}else{

Toast.makeText(this,"No Cart Items",Toast.LENGTH\_LONG).show();

}

}catch (Exception e){

Log.e("Error in Cart:" , e.getMessage());

}

}

//delete all the cart item

public void allCartItemClear(View view){

alertMethod("Do you want clear all the Cart items ?");

}

//delete a cart item one

public void clickedCartItemClear(View view){

//selected index value get in click the button this value pass by adapter btutton.setTag(position);

int index = (int)view.getTag();

String p,q,n,c;

p=cartArrayList.get(index).getPrice();

c=cartArrayList.get(index).getCategiry();

n=cartArrayList.get(index).getName();

q=cartArrayList.get(index).getQuentity();

//1-builder to a set alert values

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);

alertDialogBuilder.setTitle("Do you want clear all the Cart items ?");

alertDialogBuilder.setMessage("\n"+String.valueOf(index+1)+"\n"+n+"\n"+c+"\n"+p+"\n"+q+"\n");

alertDialogBuilder.setPositiveButton("Ok", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

cartArrayList.remove(index);

Toast.makeText(getApplicationContext(),"Removed.", Toast.LENGTH\_SHORT).show();

}

});

alertDialogBuilder.setNeutralButton("Cancel", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

Toast.makeText(getApplicationContext(), "Canceled !", Toast.LENGTH\_LONG).show();

dialog.cancel();

}

});

//2- execute alert dialog

AlertDialog alertDialog = alertDialogBuilder.create();

alertDialog.show();

}

//delete all the cart item

public void payBtn(View view){

}

//go back to view all the products

public void goBack(View view){

Intent obj = new Intent(Cart.this,Products.class);

startActivity(obj);

}

public void alertMethod(String msg){

//1-builder to a set alert values

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);

alertDialogBuilder.setTitle("");

alertDialogBuilder.setMessage(msg);

alertDialogBuilder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

Toast.makeText(getApplicationContext(),"Remove All the Cart Items In the Cart.", Toast.LENGTH\_SHORT).show();

cartNoOfItems.setText("00");

}

});

alertDialogBuilder.setNeutralButton("Cancel", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

Toast.makeText(getApplicationContext(), "Canceled !", Toast.LENGTH\_LONG).show();

dialog.cancel();

}

});

//2- execute alert dialog

AlertDialog alertDialog = alertDialogBuilder.create();

alertDialog.show();

}

}

# Task 4

## Test Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case number** | | **Test case name** | **Steps** | **Expected result** |
| TC1 |  | App Installation test | APK file install the android mobile devices | View the APK Icon in the mobile phone apps menu. |
| TC2 |  | Application user Authentication test |  |  |
|  | TC2.1 | Application user Registration Test | Input user name email password and click Signup button. | View the alert message |
|  | TC2.2 | Application user Login Test | Input user name email password and click Login button. | View the alert message |
| TC3 |  | User Interface Navigation Testing |  |  |
|  | TC3.1 | Arrow Back Button Icon Test | Click button | View the before xml screen. |
|  | TC3.2 | Cart Button Icon Test | Clicks button | View the cart xml screen |
| TC5 | Admin part | Product Manage Testing |  |  |
|  | TC5.1 | Product Add test | Input product details and click Add button. | View the alert message. |
|  | TC5.2 | Product update test | Input product details and click Add button. | View the alert message |
|  | TC5.3 | Product delete test | Input product ID and Click the delete Button | View the alert message |
|  | TC5.4 | Products view test | Show All the Products. | Show All the Products. |
| TC6 | Customer | Product Buying Process test |  |  |
|  | TC6.1 | Product view | Show All the Products. |  |
|  | TC6.2 | Product add to cart test | Click Buy Button | View the toast message |
|  | TC6.3 | Cart items view test | Show All the Cart Items. |  |
|  | TC6.4 | Cart item remove test | Click the remove button. | View the toast message |
|  | TC6.5 | Payment button test | Click the payment button. | View the alert message |
| TC7 |  | Validation check | Login and sign up validation check |  |
| TC8 |  | Filter and search check | Search the product | View the search product items. |
| TC9 |  | Payment complete | Click the payment button | View alert message |

### Test case 1 – Install APK file

|  |  |
| --- | --- |
| **Test Case NO** | 01 |
| **TEST DATE** | 2021/05/04 |
| **Test OBJECTIVE** | Install the APK file |
| **OBJECTIVE**  ID | 1 |
| **Test Data** | Install the APK file android mobile. (Huawei Y5 lite 2018) |
| **Expected Results** | View the install app icon inside the mobile apps menu. |
| **Actual Results** | View the install app icon inside the mobile apps menu. |
| **Pass/Fail** | pass |
| **Conclusion** | Expected actual output. |

|  |  |
| --- | --- |
| **Test Case NO** | 02 |
| **TEST DATE** | 2021/05/04 |
| **Test OBJECTIVE** | Registration check |
| **OBJECTIVE**  ID | 1 |
| **Test Data** | Email : [savindu@gmail.com](mailto:savindu@gmail.com)  Password : pass12345  Confirm password :pass12345  Click the Signup button |
| **Expected Results** | View the alert message. |
| **Actual Results** | View the alert message. |
| **Pass/Fail** | pass |
| **Conclusion** | Expected actual output. |

### Test - Application user Authentication

#### Test case 2.1 – User Registration

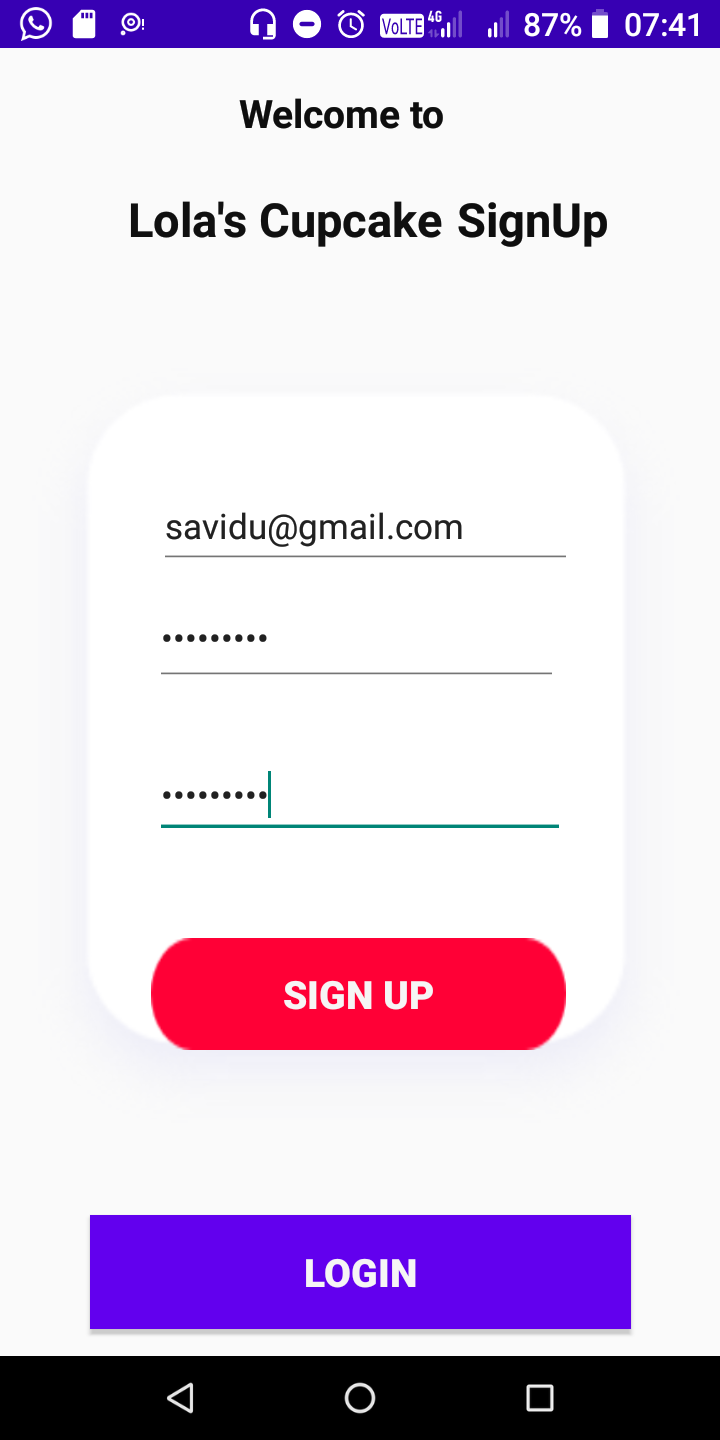


Figure 33-sign up test

#### Test case 2.2 – user Login

|  |  |
| --- | --- |
| **Test Case NO** | 01 |
| **TEST DATE** | 2021/05/04 |
| **Test OBJECTIVE** | Login check |
| **OBJECTIVE**  ID | 3 |
| **Test Data** | Email : [savindu@gmail.com](mailto:savindu@gmail.com)  Password : pass12345 |
| **Expected Results** | View the alert message success full. |
| **Actual Results** | View the alert message success full. |
| **Pass/Fail** | pass |
| **Conclusion** | Expected actual output. |

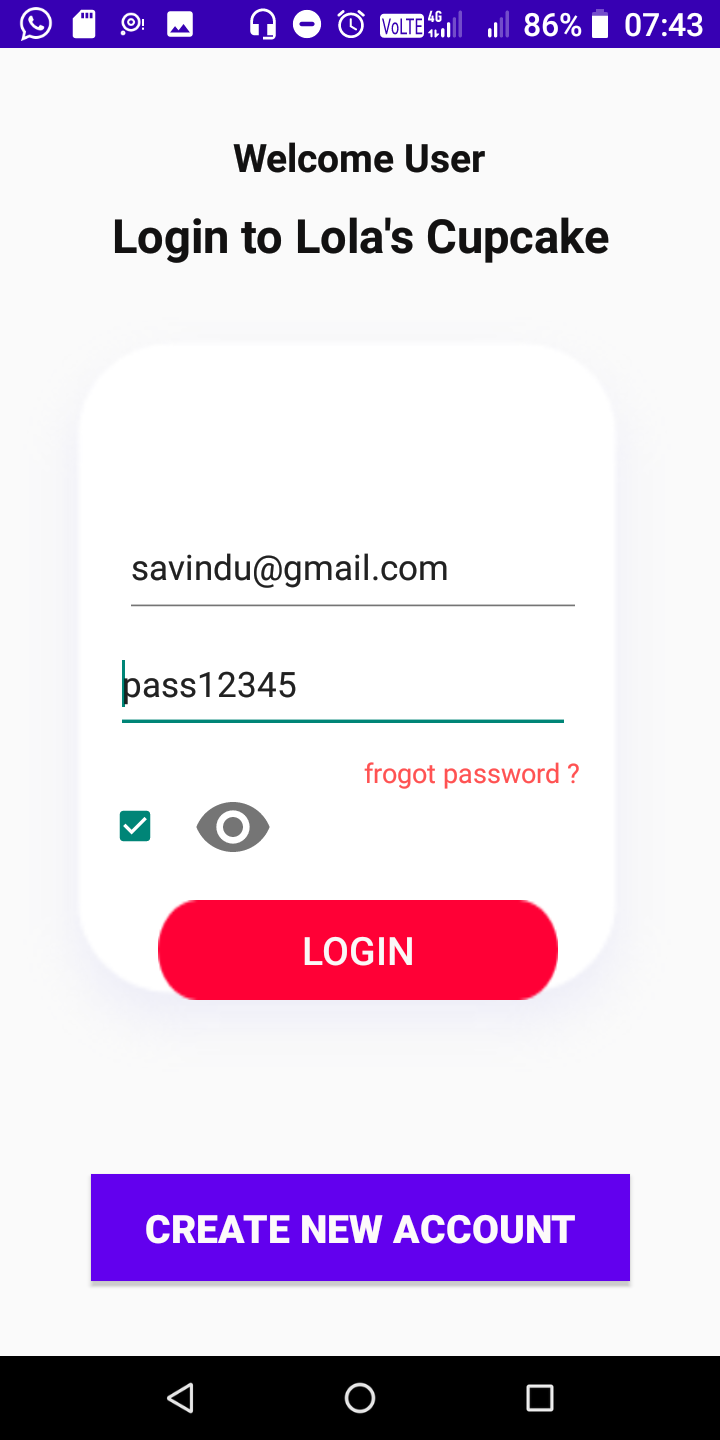


Figure 34-login test

### Test - Admin functions

#### Test case 3.1 – Product Add

|  |  |
| --- | --- |
| **Test Case NO** | 03 |
| **TEST DATE** | 2021/05/04 |
| **Test OBJECTIVE** | Product Items add check |
| **OBJECTIVE**  ID | 1 |
| **Test Data** | Input  Cupcake ID =1  name = ”Chocolate cupcake”  category = “Birthday”  price = “ 12”  Description = “It is very delicious.”  Add button click. |
| **Expected Results** | View success fully add alert message. |
| **Actual Results** | View success fully add alert message. |
| **Pass/Fail** | pass |
| **Conclusion** | Expected actual output. |

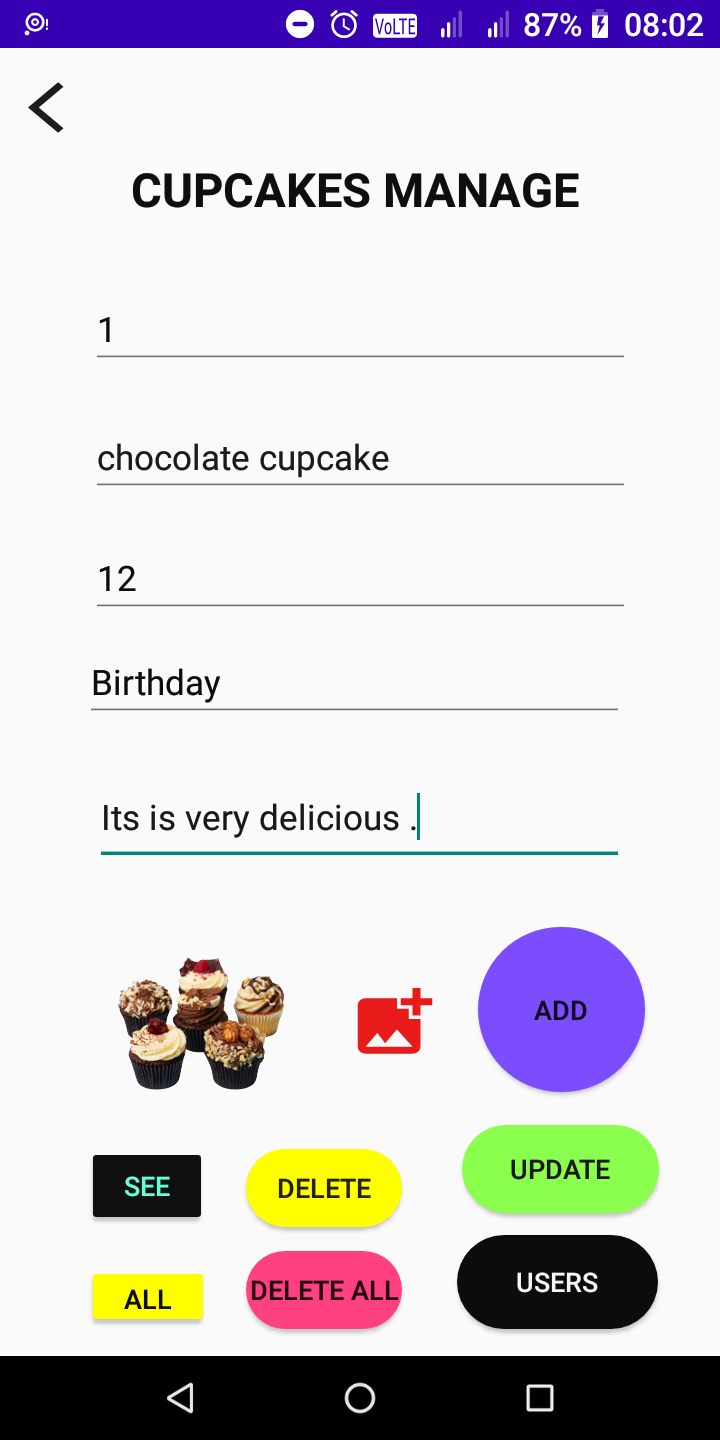


Figure 35-product add test

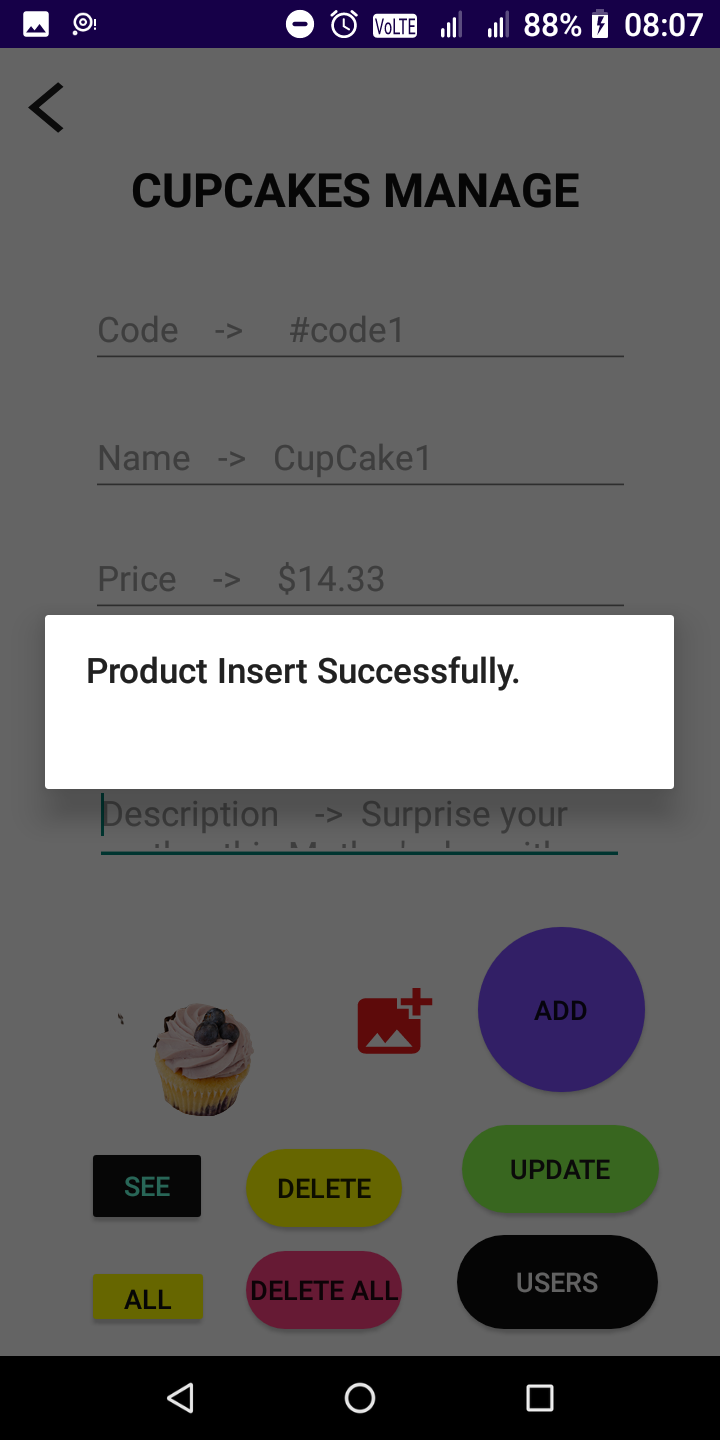


Figure 36-product added alert

#### Test case 3.2 – Product update

|  |  |
| --- | --- |
| **Test Case NO** | 03 |
| **TEST DATE** | 2021/05/04 |
| **Test OBJECTIVE** | Product Items update check |
| **OBJECTIVE**  ID | 2 |
| **Test Data** | Input  Cupcake ID =1  name = ”Vanilla cupcake”  category = “wedding”  price = “ 8”  Description = “It is very delicious.”  Update button click. |
| **Expected Results** | View success fully add alert message. |
| **Actual Results** | View success fully add alert message. |
| **Pass/Fail** | pass |
| **Conclusion** | Expected actual output. |

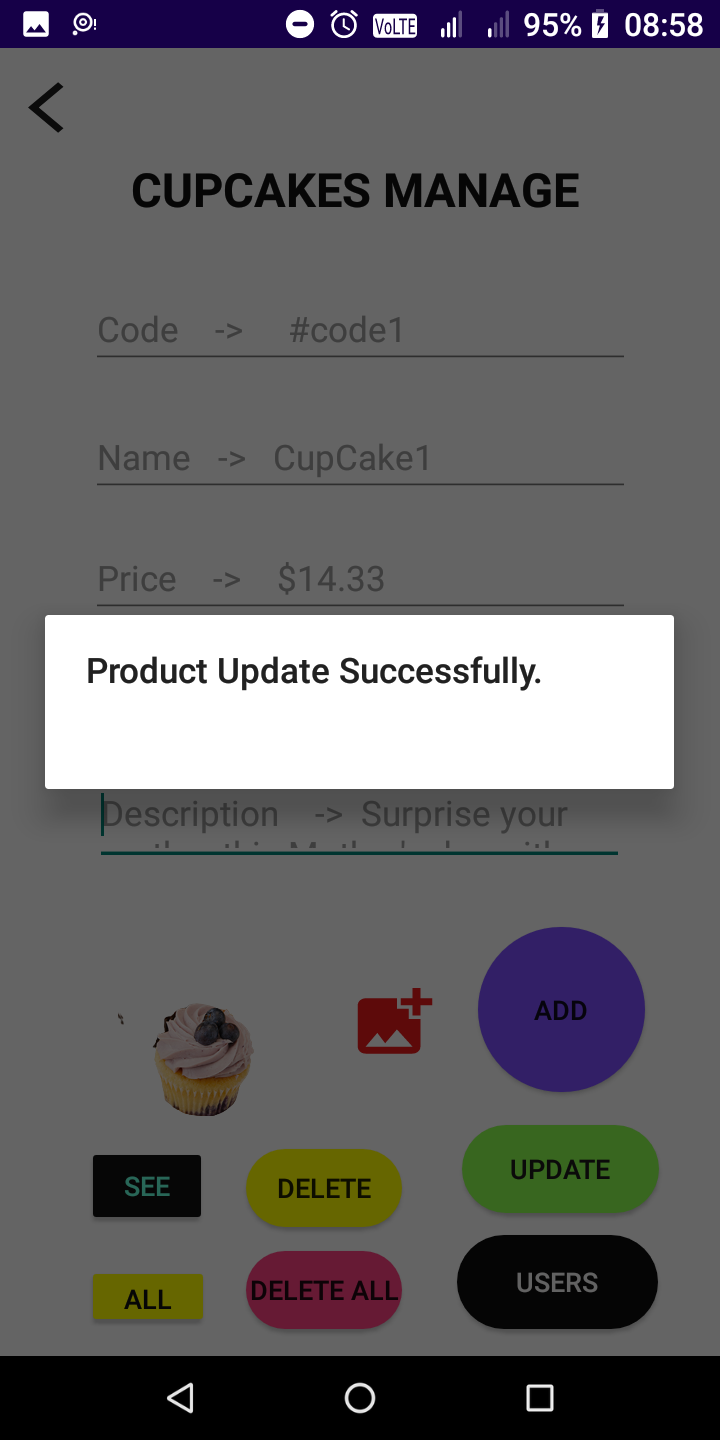


Figure 37-product update test

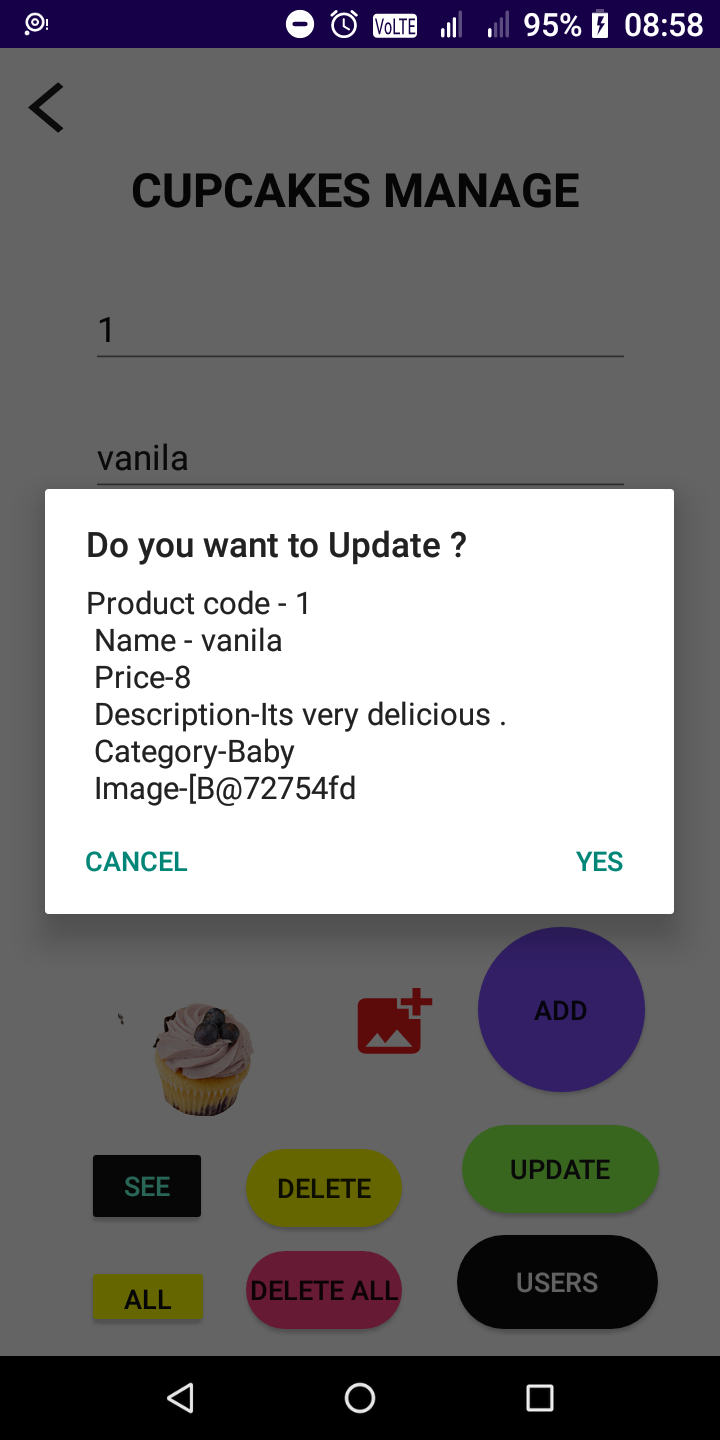


Figure 38-product update test alert

#### Test case 3.3 – Product Delete

|  |  |
| --- | --- |
| **Test Case NO** | 03 |
| **TEST DATE** | 2021/05/04 |
| **Test OBJECTIVE** | Product item delete checking |
| **OBJECTIVE**  ID | 3 |
| **Test Data** | Input  Cupcake ID =1  Delete button click. |
| **Expected Results** | View success fully deleted alert message. |
| **Actual Results** | View success fully deleted alert message. |
| **Pass/Fail** | pass |
| **Conclusion** | Expected actual output. |

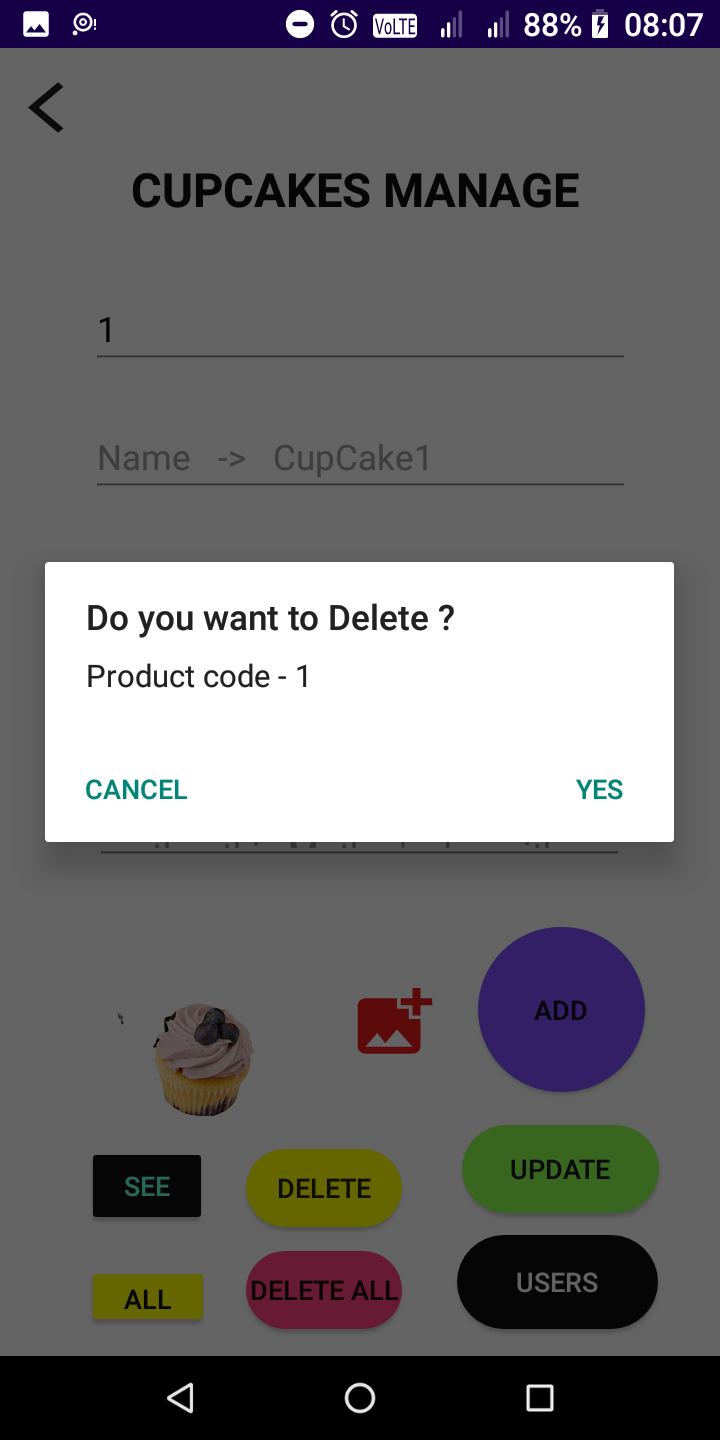


Figure 39-product delete

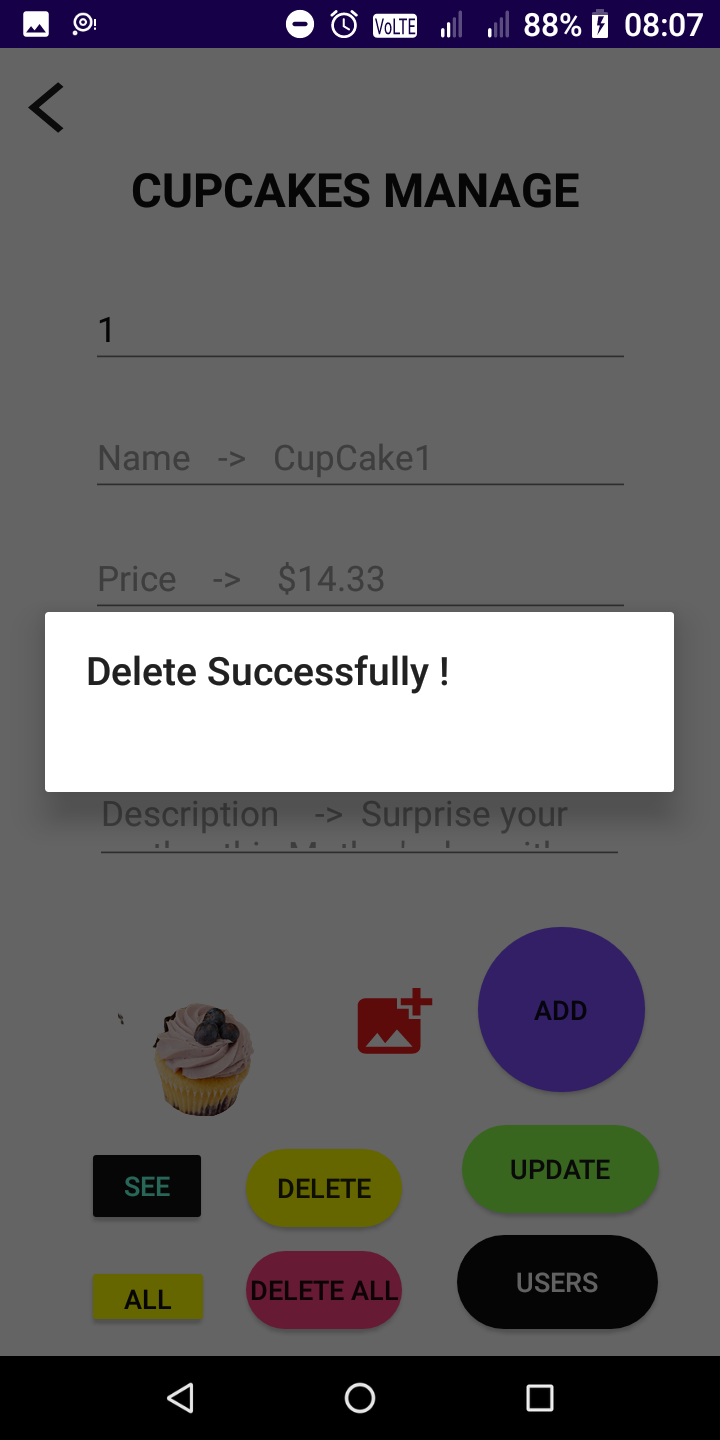


Figure 40-product delete alert

#### Test case 3.4 – Product view

|  |  |
| --- | --- |
| **Test Case NO** | 03 |
| **TEST DATE** | 2021/05/04 |
| **Test OBJECTIVE** | Product item view checking |
| **OBJECTIVE**  ID | 4 |
| **Test Data** | Show the all the product items |
| **Expected Results** | Display All items. |
| **Actual Results** | Display All items. |
| **Pass/Fail** | pass |
| **Conclusion** | Expected actual output. |

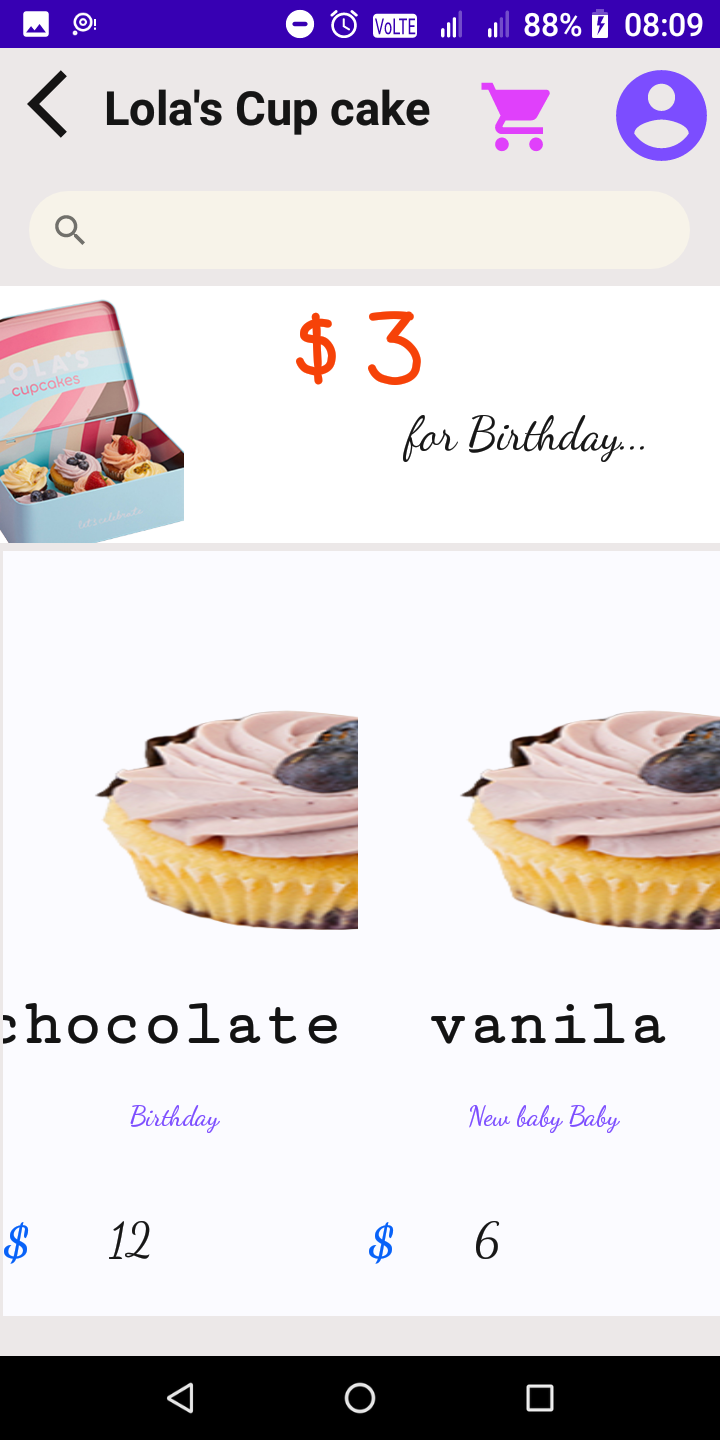


Figure 41-products view test

### Test – Product Buy process

#### Test case 6.2 – Product Add to cart

|  |  |
| --- | --- |
| **Test Case NO** | 06 |
| **TEST DATE** | 2021/05/04 |
| **Test OBJECTIVE** | Product Items add to cart check |
| **OBJECTIVE**  ID | 2 |
| **Test Data** | On click the product. Select the product quantity. Click add to cart button. |
| **Expected Results** | View success fully added alert message. |
| **Actual Results** | View success fully added alert message. |
| **Pass/Fail** | pass |
| **Conclusion** | Expected actual output. |

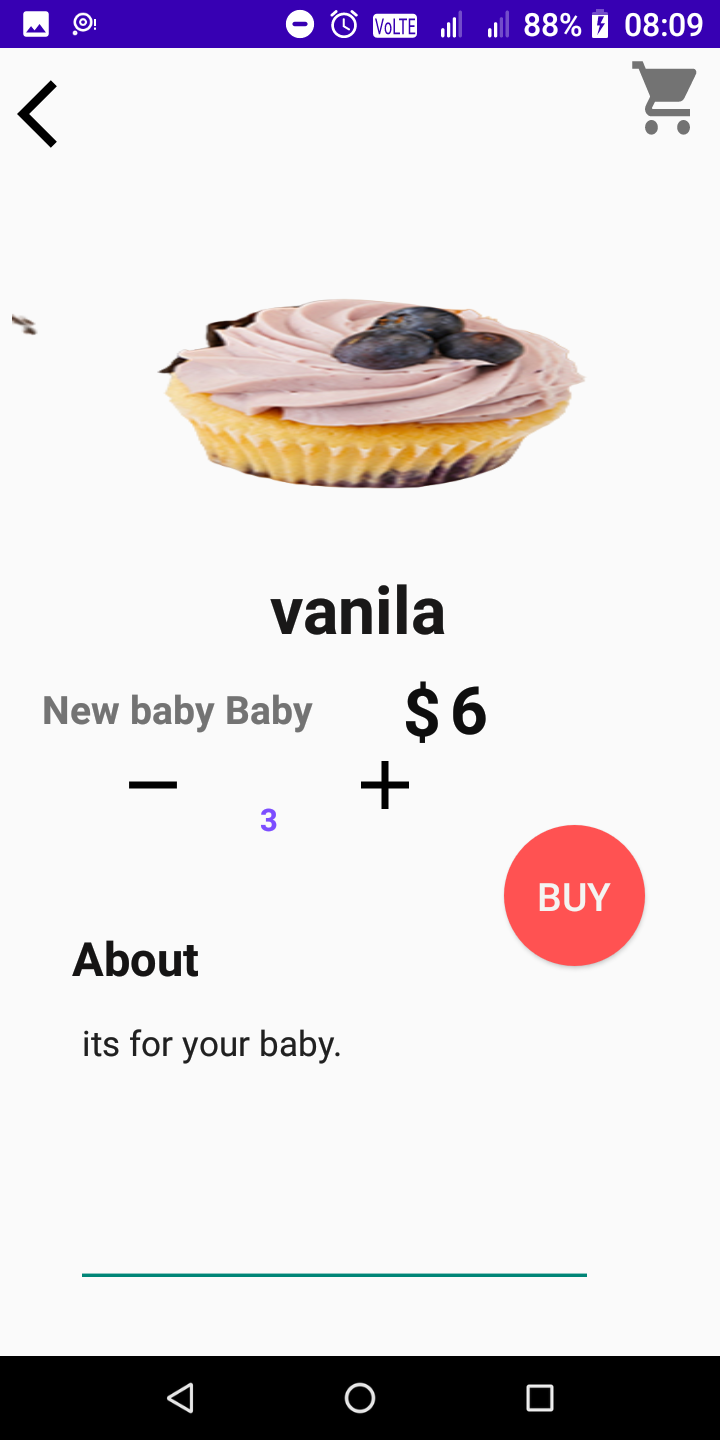


Figure 42-product add to cart test

#### Test case 6.3 – Cart items view

|  |  |
| --- | --- |
| **Test Case NO** | 06 |
| **TEST DATE** | 2021/05/04 |
| **Test OBJECTIVE** | View the cart checking |
| **OBJECTIVE**  ID | 3 |
| **Test Data** | Click the cart button. Display the list of cart items. |
| **Expected Results** | View all or toast the message. |
| **Actual Results** | View all or toast the message. |
| **Pass/Fail** | pass |
| **Conclusion** | Expected actual output. |

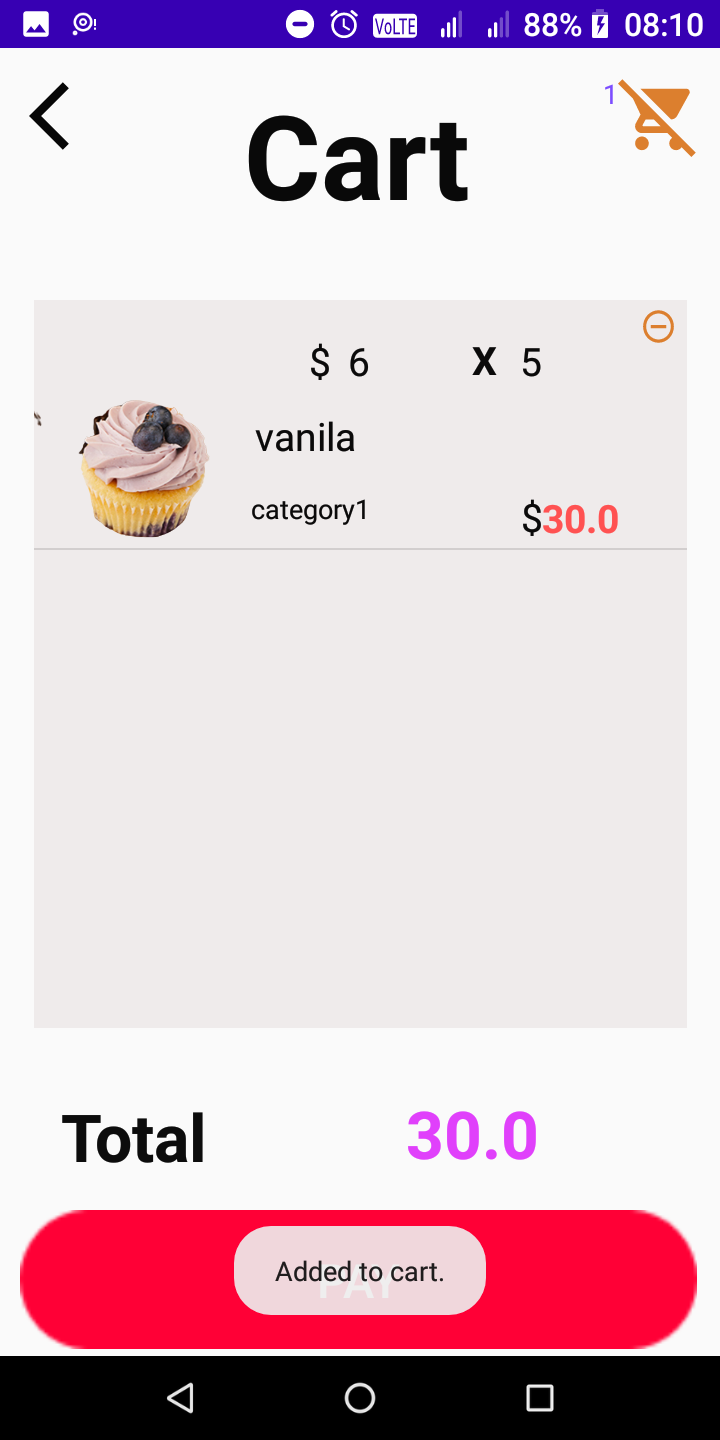


Figure 43-view cart items

#### Test case 6.4 – Cart item Delete

|  |  |
| --- | --- |
| **Test Case NO** | 06 |
| **TEST DATE** | 2021/05/04 |
| **Test OBJECTIVE** | Cart item Delete checking |
| **OBJECTIVE**  ID | 4 |
| **Test Data** | Click the specific cart item remove button. |
| **Expected Results** | View success fully removed alert message. |
| **Actual Results** | View success fully removed alert message. |
| **Pass/Fail** | pass |
| **Conclusion** | Expected actual output. |

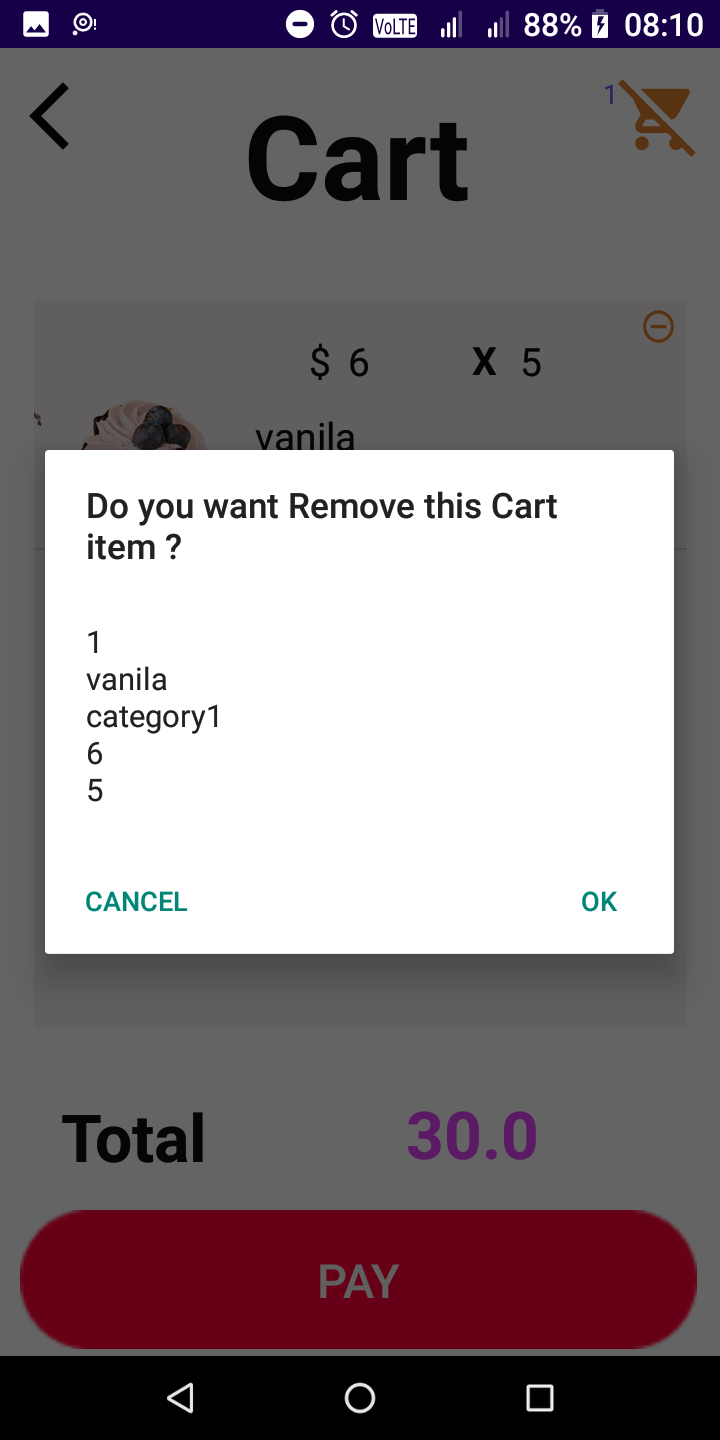


Figure 44-cart item delete test

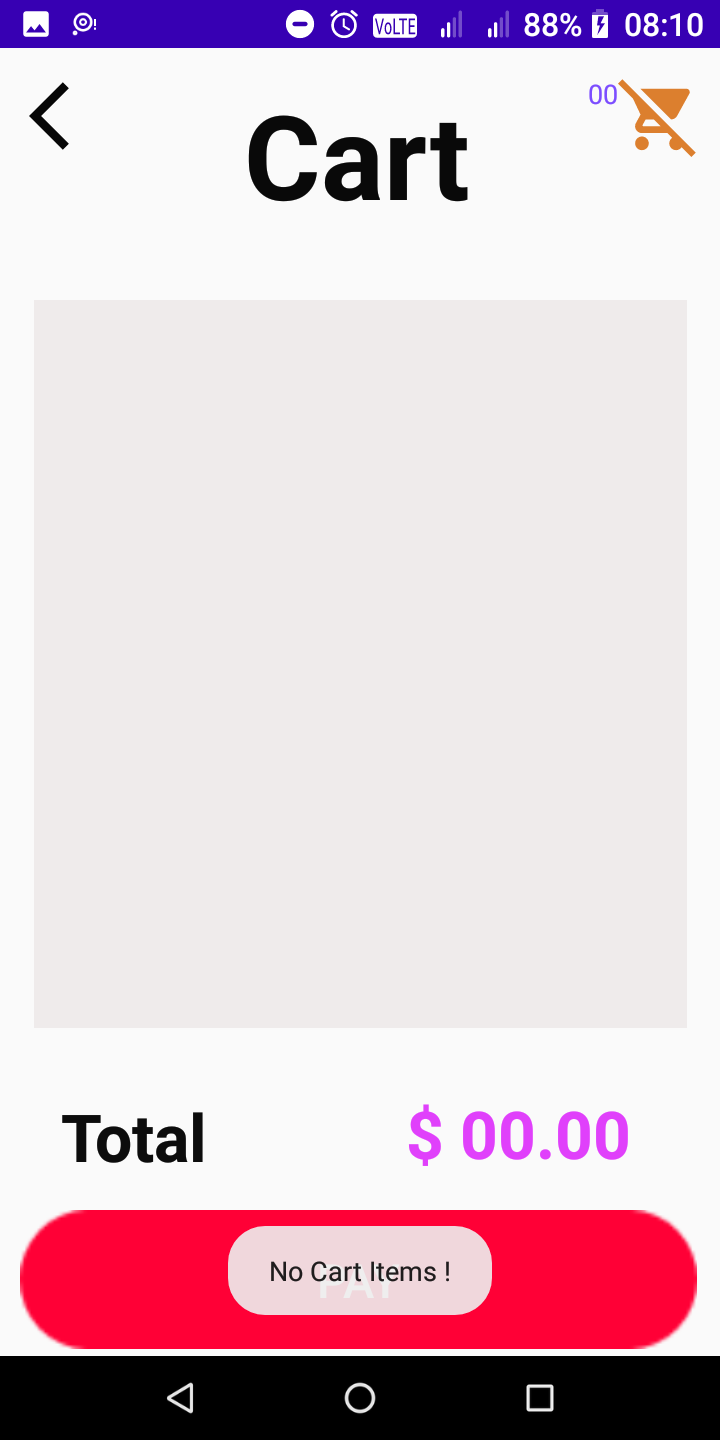


Figure 45-empty cart list

#### Test case 8 - Filter or search Product item view checking

|  |  |
| --- | --- |
| **Test Case NO** | 08 |
| **TEST DATE** | 2021/04/02 |
| **Test OBJECTIVE** | Filter or search Product item view checking |
| **OBJECTIVE**  ID | 1 |
| **Test Data** | Search bar write the products Show the all the product items |
| **Expected Results** | Display All search items. |
| **Actual Results** | Not working search items |
| **Pass/Fail** | Fail. |
| **Conclusion** | Expected actual not output. |

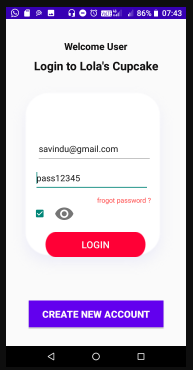
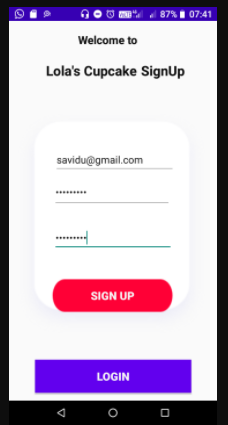
|  |  |
| --- | --- |
| **Test Case NO** | 09 |
| **TEST DATE** | 2021/04/02 |
| **Test OBJECTIVE** | Payment process to order testing |
| **OBJECTIVE**  ID | 1 |
| **Test Data** | Fill the form  First name : savindu  Last name :pasingtha  Id/passport :991452273v  Company: abs pvt ltd  Country : sri lank  Address : no 57, Ritiketiya, new colon, Meegama  Postal code : 12094  Town : Aluthgama  Province : western  Phone : +94768755787  Email : savindu@gmail.com  Fill all the check boxes.  Click the PayPal button  Redirect to PayPal payment method, when cancel the PayPal auto redirect to the home page in web site.  Save the data in database. |
| **Expected Results** | Payment success full result |
| **Actual Results** | Not complete payment. |
| **Pass/Fail** | Fail. |
| **Conclusion** | Expected actual Not output. |

### Test case – 9 Payment process to order testing

## User Manual Document

### 1 - Sign up or Login

* Application install after inside the mobile device automatically redirect to the login page. If you can enter you login user name and password system to login. If not you can click Create Account button and, now you can create new Username and password to login to this system. After then you can enter new username and password to enter to the system.
* Sign up user name should be only characters or email. Ex- email = “example@gmail.com”
* Sign up password should not be 8 length. Password = “pass12345”
* If you want to login as a system admin username = “admin”&& password =”admin”



Button

Button

Button

User name

Button

Reset password

Show/hide

Password

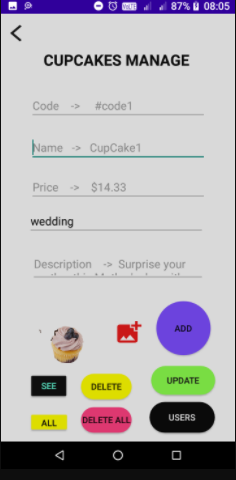
Comfrim password

password

User name

### 2 - Product Manage

* Product manage page support to product add, delete, view, user account manage and various options.
* If you want to product add, update so you must input product id, name, category name, price, cupcake image and product description. After click the add or update
* If you want product delete so you must input input only product id and then you should click delete button.



Button

Button

Button

Image

Button

Button

Input Description

Input Category

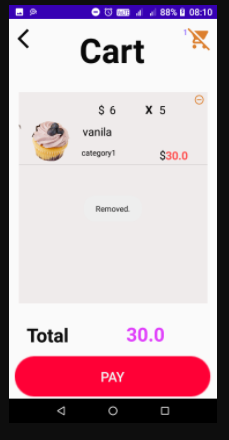
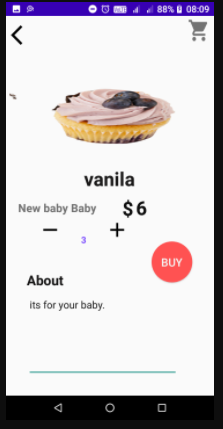
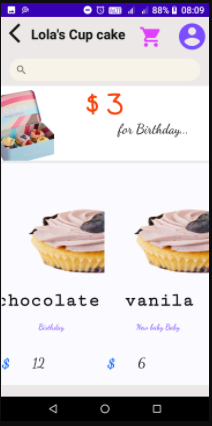
Input Price

Input Name

Input ID

### 3 - Product Buy

* When customer come to buy product, they can view the all product in the list.
* After choose some product and click on the product image then screen redirect to product quantity add screen. After the input product quantity you can click button. so now you can see cart item.
* If you want remove the cart item you can click remove button.
* If you can remove all the cart items clicking the button.



Button

# CONCLUTION

Lola’s aspiration can generate a lot of revenue though this mobile application. Not only that always, serving customers over the internet saves a lot of time. They can go to international market because of this mobile application. Play store have a lot of fake mobile ecommerce applications. So, its very disadvantage to Lola’s cupcakes. Because our customers will able to the vulnerable to fraudsters. Not only that this cupcake shop can immediately develop as an ecommerce mobile application. Sometimes they won’t daily update web site. So its bad affect to our business. Because customers reject their cupcakes. There for consumer confidence is failing. Not only that they can make a bond with their customers because they give to the most valuable cupcakes items for customers. Not only that many opportunities growth to sell their product in the internet or physically. Especially they want to internet to maintain the mobile application. So spent over money to it. Not only has that mobile phoned to spend more than our money. TAX will be pay to government. Some government regulation will be able to bad side to keep it up their mobile application exactly. I hope to their mobile app save in play store. So I think then we can get big Lola cupcakes users.

# GANTT

# References

(2021, May 02). Retrieved from Cursoradapter: https://developer.android.com/jetpack/androidx/releases/cursoradapter

(2021, May 02). Retrieved from Recyclerview: https://developer.android.com/jetpack/androidx/releases/recyclerview

(2021, May 02). Retrieved from Activity: https://developer.android.com/jetpack/androidx/releases/activity

(2021, May 03). Retrieved from Lifecycle: https://developer.android.com/jetpack/androidx/releases/lifecycle

(2021, May 04). Retrieved from AlertDialog: https://developer.android.com/reference/kotlin/androidx/appcompat/app/AlertDialog?hl=en

(2021, May 04). Retrieved from https://developer.android.com/reference/android/widget/Toast?hl=en

(2021, May 06). Retrieved from Bitmap: https://developer.android.com/reference/android/graphics/Bitmap?hl=en

(2021, May 05). Retrieved from Base64: https://developer.android.com/reference/java/util/Base64?hl=en

*sqlite.* (2021, May 2). Retrieved from developer.android.com: https://developer.android.com/jetpack/androidx/releases/sqlite?gclid=Cj0KCQjw4v2EBhCtARIsACan3nw7hED4tU6dOTvetEXqw\_S4tRIS3hFBmuoXaw9BRPf4j6yKLd-Y66YaAipzEALw\_wcB&gclsrc=aw.ds