

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology Degree in

Information Technology

By

KARTHIKEYANN
(Reg.No:-38120039)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMILNADU.
NOVEMBER 2021



**SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY**
(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119
www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **KARTHIKEYAN N** who underwent the professional training in “**ANDROID**” under our supervision from Aug 2021 to Nov 2021.

Internal Guide

Dr.P.JEYANTHI M.E., Ph.D.,

Head of the Department

Dr. R.SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **KARTHIKEYANN** hereby declare that the Professional Training Report on “**SUPER MARKET BILLING USING MOBILE APPLICATION**” done by me under the guidance of **Dr. P.JEYANTHI M.E., Ph.D.**, at Sathyabama Institute of Science and Technologies submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology .

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D.**, Dean School of Computing and **Dr.R.SUBHASHINI M.E., Ph.D.**, Head of the Department of information technology for providing me necessary support and details at the right time during the progressive reviews

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. P.JEYANTHI M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Information Technology who were helpful in many ways for the completion of the project.

ABSTRACT

The project is an Android based application where users can get instant electricity bill and pay them online via card. The project entitled Mobile payment system project is an application to assist mobile phones to access the information of different bills, send and receive queries requested by user from. In simpler words, mobile payment management software application helps customers to shop and retailers to do business without any difficulty. The system automates the conventional process of paying electricity bill by visiting the place. Users have to stand in queue for paying bill and wait for their turn. The process is tiresome and time consuming. Billing is an easy to use and comprehensive & Billing App for Retail and supermarket. It on mobile application. This application makes it easier for you to keep track of your business and pay more importance to your business growth. Surveys show that the billing systems are fast replacing the cash drawers. Unlike cash drawers, the billing systems has the option of maintaining your stock, keeping a tab on your employees, offer customer loyalty and do more in addition to billing. Furthermore, it also helps in reducing the cost of maintaining your business. Just Billing use this application lets you conveniently generate invoices, run customer loyalty program, manage and monitor sales, procurement activities, optimize inventory, manage accounts, and reports at the store level or consolidated analysis at the office anytime. Just Billing Mobile is a unique billing app for micro, small and medium size business. It is an intuitive business solution, which does not require you to have any technical knowledge and simply automates your business.!. This application is used for shop keeper for every supermarket used payments and products verification.

TABLE OF CONTENTS

Chapter No	Title	Page No.
	ABSTRACT	5
	LIST OF ABBREVIATIONS	7
1	INTRODUCTION	
1.1	OUTLINE OF THE PROJECT	8
1.2	SCOPE OF THE PROJECT	8
2	BACKGROUND AND RELATED WORK	
2.1	RELATED WORKS	9
3	SYSTEM IMPLEMENTATION	
3.1	OBJECTIVES	10
3.2	PROBLEM STATEMENT	10
3.2.1	ADVANTAGES	10
3.2.2	DISADVANTAGES	10
3.2.3	FEATURES	10
3.3	SYSTEM REQUIREMENTS	
3.3.1	SOFTWARE REQUIREMENTS	11
3.3.2	HARDWARE REQUIREMENTS & TECHNOLOGY USED	11
4	APPENDIX	
A	SOURCE CODE & DIAGRAMS	12
B	SCREENSHOTS & ARCHITECTURES	28
5	MAINTENANCE	

5.1	CONCLUSION&FUTURE WORK	31
6	REFERENCE	32

LIST OF SYMBOLS AND ABBREVIATIONS:

JDK - Java Development Kit

IDE - Integrated Development Environment

SDK - Software Development Kit

AVD - Android Virtual Device

1. Introduction

1.1 Outline of the Project.

Supermarket is the place where customers come to purchase their daily using products and pay for that. There is a need to calculate how many products are sold and to generate the bill for the customer. In our project and implementation, we have 3 users. First is the data entry operator who will enter the products in database. Second one is the administrator who will decide the taxes and commissions on the products and can see the report of any product. Third one is the bill calculating operator who will calculate the bill and print This software project is a traditional supermarket billing system with some added functionality. This system is built for fast data processing and bill generation for supermarket customers.

The billing system consists of ansql database and effective front end designed in Asp.net. The billing database is a vast collection of product name, price and other product specific data. A product when billed is searched from the database and its price is added to the bill based upon the product quantity. The system also contains discounts on various products so that the product is offered at discounted price while billing. The supermarket billing system is built to help supermarkets calculate and display bills and serve the customer in a faster and efficient manner. This software project consists of an effective and easy Gui to help the employee in easy bill calculation and providing an efficient customer service.

1.2 Scope of the Project

This billing system focus on the development of an information system that will automate manual transaction in Beatriz Food and Cafe.The proposed automated system should generate reports of daily and monthly sales including reservation transactions of Beatriz Food and Café. The system will store and recognize customer reservations. This billing system is very useful for the employees to reduce their work.

2. Background and Related Work

2.1 Related works

The existing system is not user-friendly because the retrieval of data is very slow and data is not maintained efficiently. The use of the same technology can be complicated and time-consuming. These systems need to handle by a specialist for maintaining and update the system which can again be very costly. Author KartikiDatakar, from International Journal of Computer Science and Mobile Computing, quoted that Online Attendance and Feedback System is software developed for daily student attendance in schools, colleges, and institutes. It facilitates to access the information of a particular student in a particular class. It is concluded that a graduated approach to result monitoring is the most effective response, in which sanctions have a place, although only as a last resort. Online Attendance and Feedback System are software developed for daily student attendance in schools, colleges, and institutes. Author 'Lalit Joshi', "A Research Paper on College Management System", International Journal of Computer Applications, referred that the system utilizes user authentication, displaying only information necessary for an individual's duties. Additionally, each sub-system has authentication allowing authorized users to create or update information in that subsystem. All data is thoroughly reviewed and validated on the server before actual record alteration occurs. In addition to a staff user interface, the system plans for student user interface, allowing users to access Information and submit requests online thus reducing processing time. Prof. 'S. R. Bharamagoudar', BasaveshwarEngg. College, Karnataka worked on "Web-Based Student Information Management System", from International Journal of Advanced Research in Computer and Communication Engineering referenced that the design and implementation of a comprehensive College Activity Management System and the user interface is to replace the current paper records. College Staff is able to directly access all aspects of a student's academic progress through a secure, online interface embedded in the college's website. The objective of this study is to design

and develop a customizable college management information system that provides detailed and summarized information on colleges critical areas being evaluated for funding and accreditations, that is accessible anywhere anytime as data is stored remotely and made available to users over the internet. This will help college administrators in making informed and timely decisions.

3.System Implementation:

3.1 objectives:

To make software fast in processing, with good user interface so that user can change it and it should be used for a long time without error and maintenance.

3.2 Problem Statements:

3.2.1Advantages:

- User does not have to wait in long queue and does not have to struggle with trolleys.
- User can coolly sit and pay bills for the products
- Administrator doesn't have to keep a manual track of the users.
- The system excludes manual bill calculation.
- Users don't have visit the office for bill payment.

3.2.2Disadvantages:

- We want buy products and pay using lines in super markets.
- it's take much more time.
- Sometime customers' have irritated.

3.2.3 Features:

- This Management System is fully computerized. We can store the information of each product. We can calculate the stock computerize system.

- We can generate various reports like purchase, sales and stock. Save expensive.

3.3 System Requirements:

3.3.1 Software Requirements:

Operating system	-	Windows XP
Technology Used	-	Android 2.2
IDE	-	Eclipse 3.4 (min)
Emulators	-	Micro emulator 5055
Plug-in	-	ADT plug-in
Tools used	-	Android SDK 1.2

3.3.2: Hardware Requirements:

Processor	-	Pentium P4
Motherboard	-	Genuine Intel
RAM	-	Min 1 GB
Hard Disk	-	80 GB

3.3.2: TECHNOLOGY USED:

- This project can be implemented only in JAVA because Android supports only JAVA for user applications.

JAVA

- Java is Platform Independent. Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. It implements a strong security model, which prevents compiled Java programs from illicitly accessing resources on the system where they execute or on the network. Popular World-Wide Web browsers, as well as some World-Wide Web servers and other systems implement Java interpreters. These are used to display interactive user interfaces, and to script behavior on these systems.

ANDROID

- Android is a complete set of software for mobile devices such as tablet computers, Smartphone's, electronic book readers, notebooks, set-top boxes etc. It contains an Linux-based OS, middleware and key mobile applications. It can be thought of as a mobile operating system. But it is not limited to mobile only. It is currently used in mobiles, tablets, televisions etc.

MYSQL SERVER:

- Microsoft SQL Server is an application used to create computer databases for the Microsoft Windows family of server operating systems. Microsoft SQL Server provides an environment used to generate databases that can be accessed from workstations, the Internet, or other media such as a personal digital assistant (PDA).
- MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python".
- MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, Mac OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

4. APPENDIX:

A. Source Code

1)admin home activity.java:

```
package com.example.admin;  
  
import com.example.expdetails.ExpiryActivity;
```

```

import com.example.newww.EmpDetailsActivity;
import com.example.newww.R;
import com.example.newww.R.layout;
import com.example.newww.R.menu;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.graphics.Typeface;
import android.view.Menu;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.TextView;

public class AdminHomeActivity extends Activity {
    Button b1, b2, b3;
    TextView t1;

    @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_admin_home);

        b1 = (Button) findViewById(R.id.adHomeBilling);
        b2 = (Button) findViewById(R.id.adHomeEmp);
        b3 = (Button) findViewById(R.id.adHomeExp);
        t1 = (TextView) findViewById(R.id.idTitle);

        Typeface typeface = Typeface.createFromAsset(getAssets(),
"seaside.ttf");
        t1.setTypeface(typeface);

        b1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                Intent i = new Intent(AdminHomeActivity.this,
Billing.class);
                startActivity(i);
            }
        });

        b2.setOnClickListener(new View.OnClickListener() {

```

```

        @Override
        public void onClick(View arg0) {
            Intent i = new Intent(AdminHomeActivity.this,
EmpDetailsActivity.class);
            startActivity(i);
        }
    });

    b3.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            Intent i = new Intent(AdminHomeActivity.this,
ExpiryActivity.class);
            startActivity(i);
        }
    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.admin_home, menu);
    return true;
}

}

```

2)bill adapter.java:

```

package com.example.admin;

import java.util.ArrayList;

import com.example.newww.R;
import com.example.newww.CustomAdapter.Holder;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.View.OnClickListener;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

```

```

public class BillAdapter extends BaseAdapter {
    private ArrayList<String>emp_name = new ArrayList<String>();
    private ArrayList<String>cus_name = new ArrayList<String>();
    private ArrayList<String> bill = new ArrayList<String>();
    private ArrayList<String>dat = new ArrayList<String>();
    Context context;
    private static LayoutInflater inflater = null;
    public BillAdapter(Context context, ArrayList<String>emp_nameee,
        ArrayList<String>cus_nameee, ArrayList<String>bille,
        ArrayList<String> date) {
        // TODO Auto-generated constructor stub
        emp_name = emp_nameee;
        this.context = context;
        cus_name = cus_nameee;
        bill = bille;
        dat = date;

        inflater = (LayoutInflater) context

        .getSystemService(Context.LAYOUT_INFLATER_SERVICE);

    }

    @Override
    public int getCount() {
        // TODO Auto-generated method stub
        return emp_name.size();
    }

    @Override
    public Object getItem(int position) {
        // TODO Auto-generated method stub
        return position;
    }

    @Override
    public long getItemId(int position) {
        // TODO Auto-generated method stub
        return position;
    }

    public class Holder {
        TextViewtEmpName, tCusName,tAmount,tdate;

    }

    @Override
    public View getView(final int position, View convertView, ViewGroup parent) {
        // TODO Auto-generated method stub
        Holder holder = new Holder();

```

```

        View rowView;
        rowView = inflater.inflate(R.layout.billing_model, null);
        holder.tEmpName = (TextView)
rowView.findViewById(R.id.adIdEmpName);
        holder.tCusName = (TextView)
rowView.findViewById(R.id.adIdCusName);
        holder.tAmount = (TextView) rowView.findViewById(R.id.adIdBilling);
        holder.tdate = (TextView) rowView.findViewById(R.id.adIdDate);

        holder.tEmpName.setText(emp_name.get(position));
        holder.tCusName.setText(cus_name.get(position));
        holder.tAmount.setText(bill.get(position));
        holder.tdate.setText(dat.get(position));

        return rowView;
    }
}

```

3)billing.java:

```

package com.example.admin;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import welcome.Ipaddress;
import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;

```

```

import android.util.Log;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Adapter;
import android.widget.ListView;
import android.widget.Toast;

import com.example.newww.R;

public class Billing extends Activity {
    ListView listView;
    HttpClient httpClient;
    HttpPost httpPost;
    HttpResponse httpResponse;
    BillAdapter adapter;
    Context context = this;
    private ArrayList<String> emp_name = new ArrayList<String>();
    private ArrayList<String> cus_name = new ArrayList<String>();
    private ArrayList<String> bill = new ArrayList<String>();
    private ArrayList<String> dat = new ArrayList<String>();
    private StringBuilder stringBuilder;
    String getData;

    @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_billing);

        Log.d("opening", "opening");
        adapter = new BillAdapter(context, emp_name, cus_name, bill, dat);
        listView = (ListView) findViewById(R.id.billList);
        listView.setAdapter(adapter);
        new Thread() {
            public void run() {
                try {
                    httpClient = new DefaultHttpClient();
                    httpPost = new HttpPost(Ipaddress.URL);
                    List<NameValuePair> nameValuePair = new
ArrayList<NameValuePair>();
                    nameValuePair.add(new
BasicNameValuePair("flag", "15"));
                    httpPost.setEntity(new
UrlEncodedFormEntity(nameValuePair));
                    httpResponse = httpClient.execute(httpPost);
                    InputStream inputStream =
httpResponse.getEntity()

```

```

        .getContent();

        InputStreamReaderinputStreamReader    =    new
InputStreamReader(
        inputStream);

        BufferedReaderbufferedReader    =    new
BufferedReader(
        inputStreamReader);

        stringBuilder = new StringBuilder();
        String bufferedStrChunk = null;

        while        ((bufferedStrChunk        =
bufferedReader.readLine()) != null) {
            stringBuilder.append(bufferedStrChunk);
            Log.d("first jjj", stringBuilder.toString());
            getData = stringBuilder.toString();
        }

        handler.sendMessage(1);
    } catch (Exception e) {
        handler.sendMessage(3);
    }
}
}.start();

}

Handler handler = new Handler() {
    public void handleMessage(Message msg) {
        switch (msg.what) {
            case 1:
                try {
                    JSONArrayj_arr = new JSONArray(getData);
                    JSONObjectj_obj = null;
                    for (int i = 0; i<j_arr.length(); i++) {
                        j_obj = j_arr.getJSONObject(i);
                        Log.d("sartry", j_obj.toString());

                        emp_name.add(j_obj.getString("Aemployee name"));

                        cus_name.add(j_obj.getString("Acustomer name"));
                        bill.add(j_obj.getString("Abill"));
                        dat.add(j_obj.getString("Adate"));
                    }
                    Log.d("college name size: ",
                        String.valueOf(emp_name.size()));
                    Collections.reverse(emp_name);
                    Collections.reverse(cus_name);
                }
            }
        }
    }
}

```

```

        Collections.reverse(bill);
        Collections.reverse(dat);
    } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (Exception e) {

    }
    adapter.notifyDataSetChanged();

    break;
case 2:

    Toast.makeText(getApplicationContext(),
        "You dont have a post here...",
Toast.LENGTH_LONG)
        .show();

    break;
case 3:

    Toast.makeText(getApplicationContext(),
        "Please check your internet connection!",
        Toast.LENGTH_LONG).show();

    break;

    }
    }
};
}

```

4)custom adapter.java:

```

package com.example.newww;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.os.Bundle;
import android.preference.Preference;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

```

```

import android.widget.Toast;

public class CustomAdapter extends BaseAdapter {
    String[] result, rate;
    Context context;
    int[] imageId;
    StringBuilder value = new StringBuilder();
    float amount;
    float tot, net;
    SharedPreferences pref;
    private static LayoutInflater inflater = null;
    Editor editor;

    public CustomAdapter(Context context, String[] prgmNameList,
        int[] prgmImages, String[] prgmNameRate) {
        // TODO Auto-generated constructor stub
        result = prgmNameList;
        this.context = context;
        imageId = prgmImages;
        rate = prgmNameRate;
        inflater = (LayoutInflater) context

        .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @Override
    public int getCount() {
        // TODO Auto-generated method stub
        return result.length;
    }

    @Override
    public Object getItem(int position) {
        // TODO Auto-generated method stub
        return position;
    }

    @Override
    public long getItemId(int position) {
        // TODO Auto-generated method stub
        return position;
    }

    public class Holder {
        TextView tv, tv1;
        ImageView img;
    }
}

```

```

@Override
public View getView(final int position, View convertView, ViewGroup parent) {
    // TODO Auto-generated method stub
    Holder holder = new Holder();
    View rowView;
    rowView = inflater.inflate(R.layout.program_list, null);
    holder.tv = (TextView) rowView.findViewById(R.id.textview1);
    holder.img = (ImageView) rowView.findViewById(R.id.imageView1);
    holder.tv1 = (TextView) rowView.findViewById(R.id.textview2);
    holder.tv.setText(result[position]);
    holder.img.setImageResource(imageId[position]);
    holder.tv1.setText(rate[position]);

    rowView.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            // net = 0;

            net = Float.parseFloat(rate[position]);
            net = amount + net;
            amount = net;

            value.append("\n" + result[position]);
            value.append(" : " + rate[position]);

            // amount.append(amount+rate[position]);
            Toast.makeText(context, "Added", 1000).show();

            pref          =          context.getSharedPreferences("cart",
context.MODE_PRIVATE);

            editor = pref.edit();

            editor.putString("1", String.valueOf(net));
            editor.putString("2", String.valueOf(value));

            editor.commit();

        }

    });
    return rowView;
}

}

```

5)home 21ctivity.java:

```

package com.example.newww;

import category_activity.Bakery;
import category_activity.Beauty;
import category_activity.Beverages;
import category_activity.Fruits;
import category_activity.HouseHold;
import category_activity.MainActivity;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;

public class HomeActivity extends Activity {

    @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_home);
    }

    public void foodgrains(View view) {
        Intent i = new Intent(HomeActivity.this, MainActivity.class);
        startActivity(i);
    }

    public void bakery(View view) {
        Intent i = new Intent(HomeActivity.this, Bakery.class);
        startActivity(i);
    }

    public void fruits(View view) {
        Intent i = new Intent(HomeActivity.this, Fruits.class);
        startActivity(i);
    }

    public void beverages(View view) {
        Intent i = new Intent(HomeActivity.this, Beverages.class);
        startActivity(i);
    }

    public void beauty(View view) {
        Intent i = new Intent(HomeActivity.this, Beauty.class);
        startActivity(i);
    }
}

```

```

    }

    public void household(View view) {
        Intent i = new Intent(HomeActivity.this, HouseHold.class);
        startActivity(i);
    }

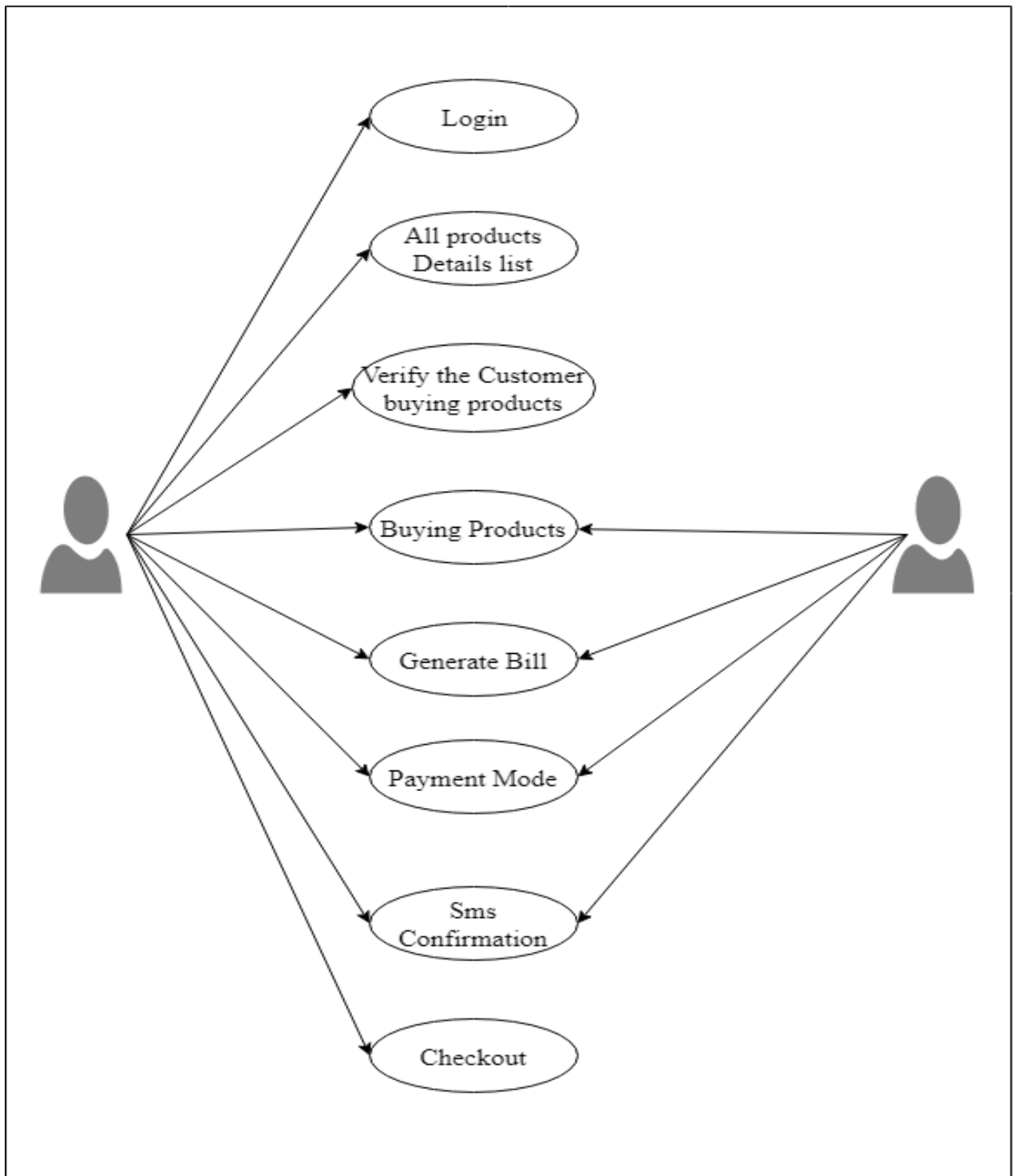
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.home, menu);
        return true;
    }
}

```

DIAGRAMS:

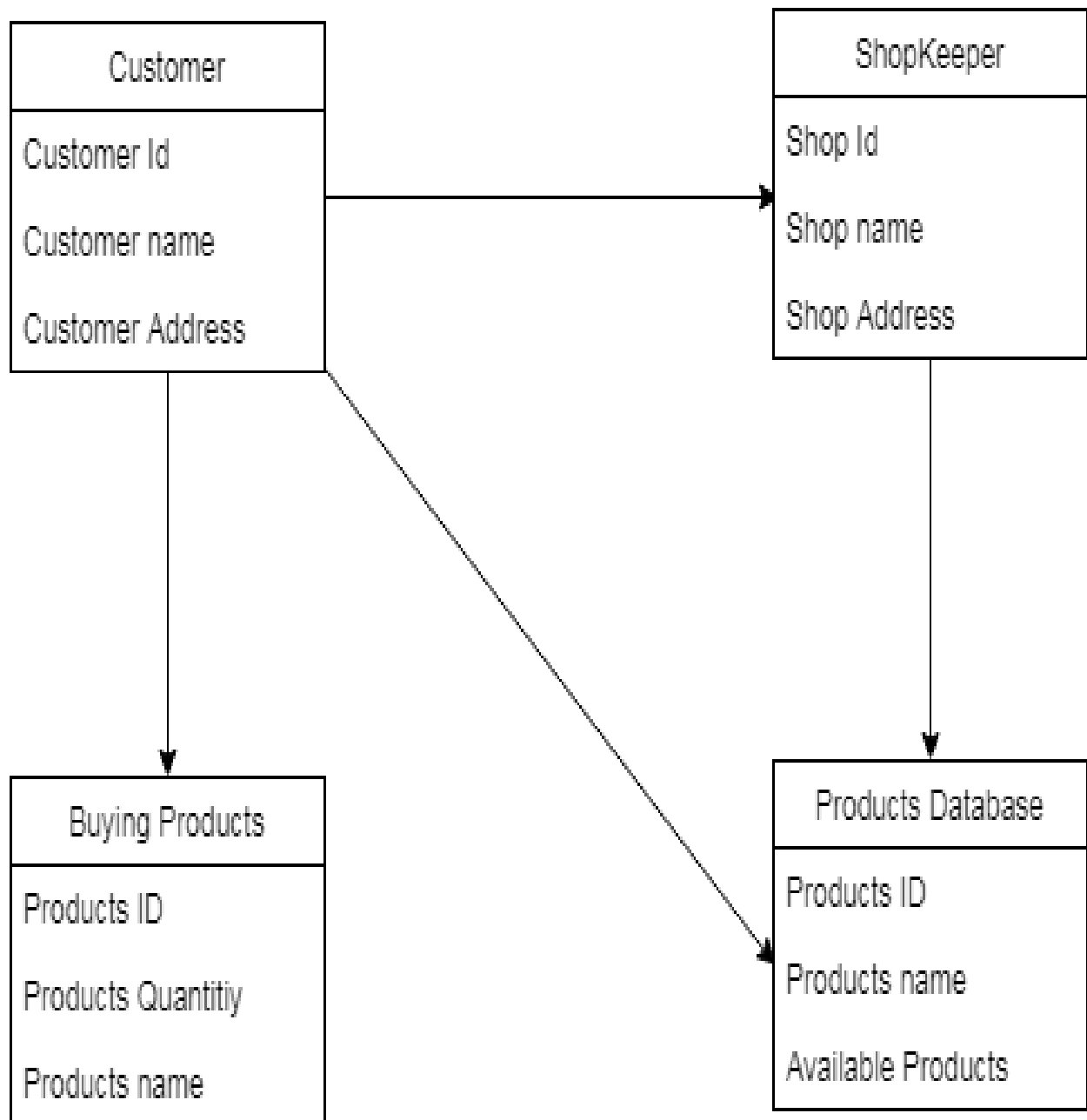
USE CASE DIAGRAM:

- **UML Use Case Diagrams.** Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system



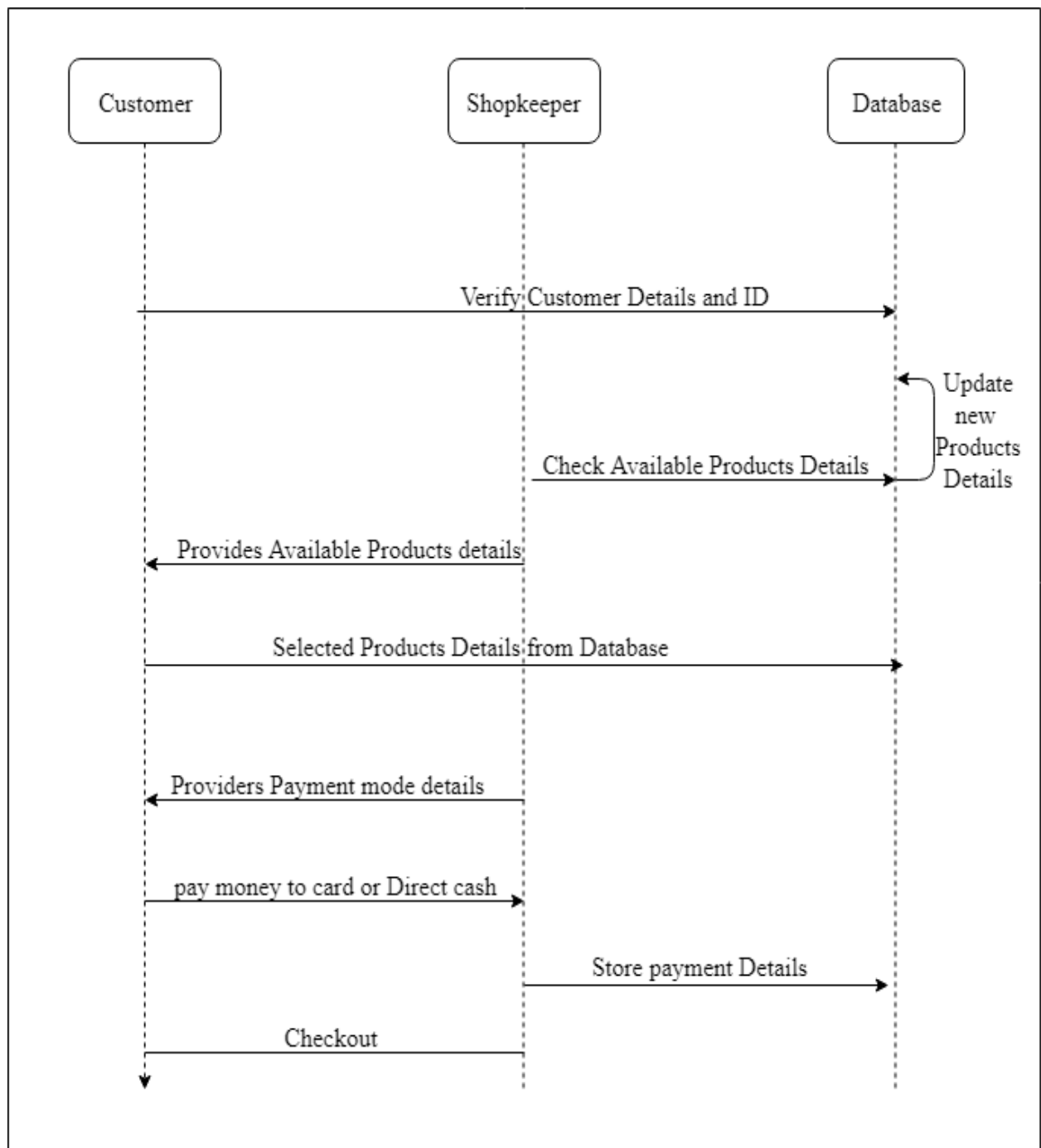
CLASS DIAGRAM:

- A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.



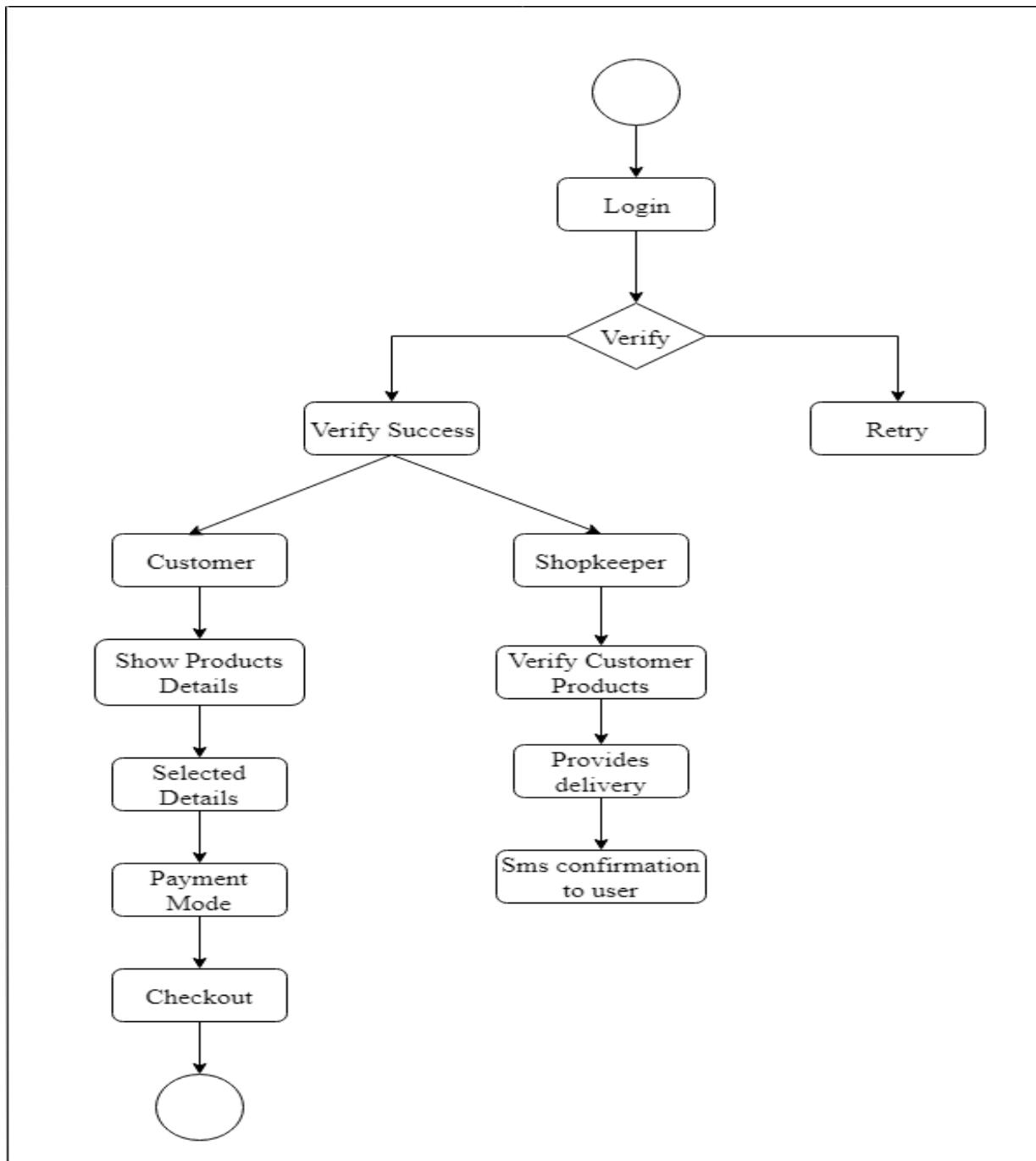
SEQUENCE DIAGRAM:

- A sequence diagram shows object interactions arranged in time **sequence**. It depicts the objects and classes involved in the scenario and the **sequence** of messages exchanged between the objects needed to carry out the functionality of the scenario. ... Sequence diagrams are sometimes called event diagrams or event scenarios.



ACTIVITY DIAGRAM:

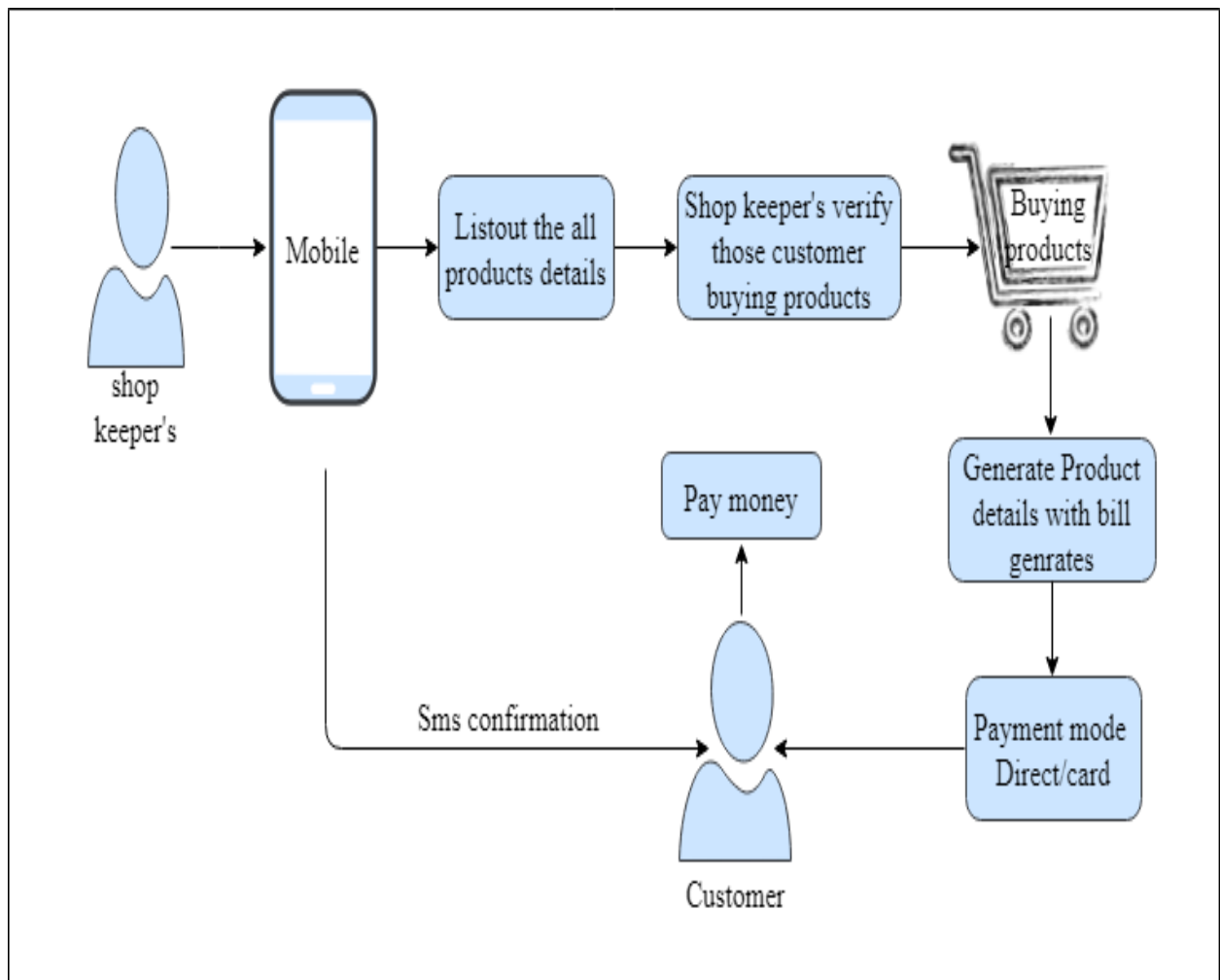
- In Unified Modeling Language (UML), an activity diagram is a graphical representation of an executed set of procedural system activities and considered a state chart diagram variation. **Activity diagrams** describe parallel and conditional activities, use cases and system functions at a detailed level.



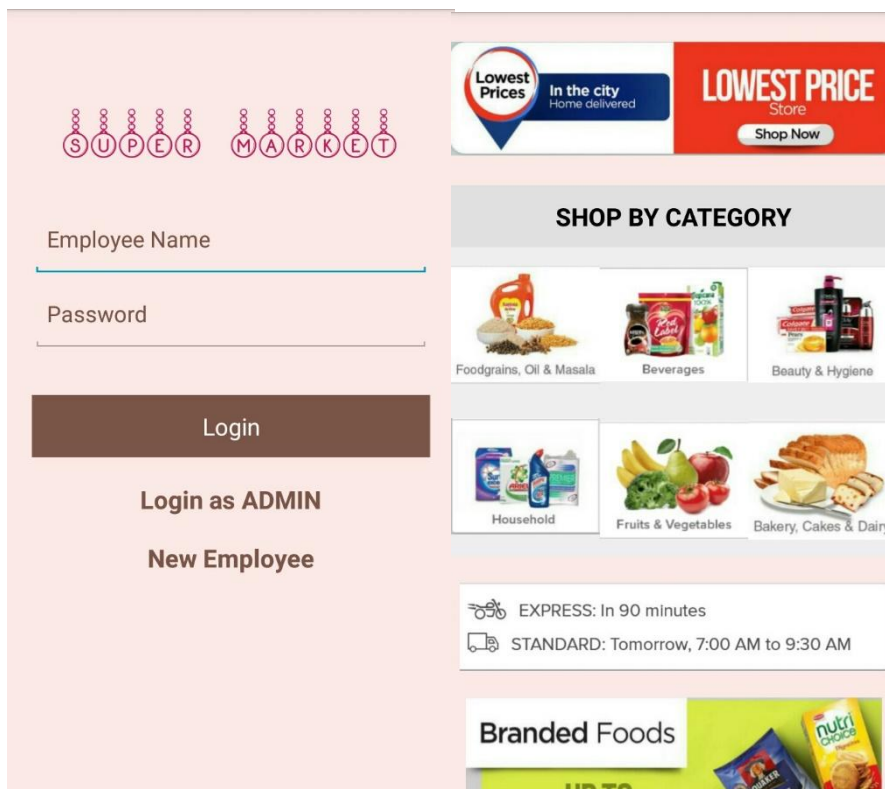
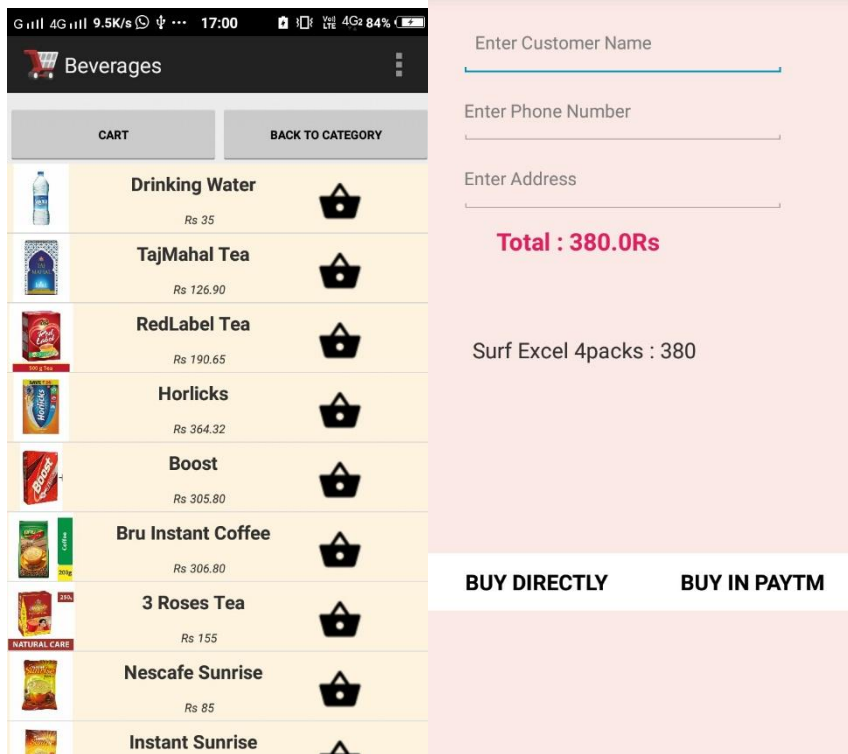
ARCHITECTURES:

Overall Architecture Diagram:

- An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components.



B. SCREENSHOTS:



Bakery			HouseHold		
CART		BACK TO CATEGORY	CART		BACK TO CATEGORY
	Cow Milk Rs 48			Harpic Rs 136.80	
	Malai Panner Rs 67.90			VIM Dishwash Bar Rs 30	
	Greek Yogurt Rs 40			Lizol 4ltr Rs 598.50	
	Musti Buttermilk Rs 52.25			Comfort Rs 188.60	
	Bacconcini Rs 135			Surf Excel 4packs Rs 380	
	Bread Rs 38			Plasto Bags Rs 60	
	Bun Rs 23			RIN Detergent Bar Rs 5	
	Pizza base Rs 30			A Added Rs 180	
	Plum Cake			Patanjali Dishwash Bar	

Super Market			Fruits		
CART		BACK TO CATEGORY	CART		BACK TO CATEGORY
	Sugar Rs 48			Amaranthus Rs 13	
	Salt Rs 15			Ash Guard Rs 25	
	Jaggery Round Rs 50			Banana Rs 17	
	Natura Rs 270			Banana Stem Rs 12	
	Coconut Palm Sugar Rs 399			Beans Rs 35	
	Atta Rs 390			Beetroot Rs 10	
	Bombay Ravai Rs 37			Bitterguard Rs 45	
	Maida Rs 35			Bottle Guard Rs 15	
	Besan Flour			Brinjal Rs 18	

5. MAINTENANCE:

- The objectives of this maintenance work are to make sure that the system gets into work all time without any bug. Provision must be for environmental changes which may affect the computer or software system.
- This is called the maintenance of the system. Nowadays there is the rapid change in the software world. Due to this rapid change, the system should be capable of adapting these changes. In our project the process can be added without affecting other parts of the system.
- Maintenance plays a vital role. The system will be able to accept any modification after its implementation.
- This system has been designed to favor all new changes. Doing this will not affect the system's performance or its accuracy.
- This is the final step in system life cycle. Here we implement the tested error-free system into real-life environment and make necessary changes, which runs in an online fashion.
- As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware.
- System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

5.1 CONCLUSION:

- Supermarket Management System has to do with making appropriate effort to stop the rising problem to all manual supermarket operation in order to enhance the operation of such supermarket. In this project, the software or system that can be used to aid all supermarkets that is still operating manually

have been successfully developed. The software can be implementing in all types of supermarket.

5.1.1 FUTURE WORK:

- The project has a very vast scope in future. In terms of future scope, a variety of data mining techniques can be used by researchers to simplify customer perceptions and attitudes. Every day, every hour and every minute, tera-bytes of data gets generated from millions of shoppers, yet, retail managers/ business executives always grapple with relevant information that can help retailers/ researchers design strategies to generate customer loyalty.

6. REFERENCES:

- [1] J. Qin, Y. Zhou, and H. Chen, "A multi-region empirical study on the Internet presence of global extremist organizations," *Inf. Sys. Frontiers*, vol. 13, no. 1, pp. 75–88, 2011.
- [2] Takayuki Nagai, Ryuichi Matsuba, Shinichirou Kubota, Toshihiro Kita, Shirou Kitamura, Yasuhiro Migita, Yasuo Musashi, Kenichi Sugitani, Masashi Toda, Hiroshi Nakano: Open implementation of FCF campus-card reader for attendance checking and integration with LMS, *Journal of Academic Computing and Networking*, 17, pp.67-76, 2013.
- [3] Ichimura Takumi, Kamada Shin: Attendance Management System with NFC(Near Field Communication) on Nexus, *Symposium on Mobile Interactions 2013*, pp.139-140, Mar. 2013. [4] Nippon Systems Development, Attendance management web system <http://www.systemgear.com/product/pdc300/school.html> (Accessed on Jan 16, 2015)
- [5] Seth Ladd, Darren Davison, StevenDevijver and Colin Yates, *Expert Spring MVC and Web Flow Apress*. Apress, 2006.
- [6] D. Y. Shi, *Management Information System*. Publishing House of Electronics Industry, 2006.
- [7] H. C. Bi and X. K. Li, "Design and research of the Web project based on

Spring+Struts+Hibernate framework”, Computer Development & Applications, Taiyuan, vol. 21, pp.51–52, Apl. 2008.

THANK YOU!!!