A

Project Report On

Food For You

Submitted To

IN

PARTIAL FULFILLMENT OF

B. Sc. COMPUTER SCIENCE

SUBMITTED BY

Mr. Dattaprasad Subhash Tambe

Under The Guidance Of

Prof. Miss. S.R. Joshi

THROUGH

DEPARTMENT OF COMPUTER SCIENCE
Kankavli College, Kankavli

2019-2020

DECLARATION

To,
The Head,
Department of Computer Science,
Kankavli College, Kankavli

Respected Madam,

I undersigned, hereby declare that the project on **"Food For You"** is developed under the guidance of the project guide **Prof. Miss. S.R. Joshi**.

The conclusion in this report is based on the data, which is collected by me. I am declaring that this is my original work. I have not copied any materials, which are used while developing this project, or other reports that are submitted to the Kankavli College, Kankavli.

I do undersign that if my work is found to be copied, then I am liable to punishment as per the university rule.

DATE: 21/09/2020

PLACE: Kankavli (Dattaprasad S. Tambe)

S. P. Mandal's

KANKAVLI COLLEGE, KANKAVLI

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that ***Mr. Tambe Dattaprasad Subhash*** *have satisfactorily carried out the project work entitled "****Food For You****" as per prescribed in the syllabus of T. Y. B.Sc. (Computer Science) of* ***Mumbai University, Mumbai****.*

It is also to certify that this is his own work completed during academic year 2019 – 2020. The work done is satisfactory and as per the specifications.

PROJECT GUIDE	EXTERNAL EXAMINER	HEAD OF DEPARTMENT	PRINCIPAL

Certificate of Approval

This is certify that the project titled

“Food For You”

Is bonafied record of project work done

by

Mr. Dattaprasad Subhash Tambe.

Seat No:_4038024

This Project is approved for the Bachelor Degree of T.Y.B.Sc. (Computer Science) of University of Mumbai

Examiners:

1) _____

2) _____

CERTIFICATE

LETTER OF AUTHORIZATION

This is to certify that, the product details and references used in project are provided and permitted by **Kankavli's Mint Leaf Hotel**.

Also **Mr.Dattaprasad Subhash Tambe** has satisfactorily completed Computerized Project "Food For You" during 2019-20 .

We find the project as per requirements and it is running successfully.

Date: 21/09/2020

Place: Kankavli

ACKNOWLEDGEMENT

My project **"Food For You"** was a very rewarding experience. It was my second experience to work on project updated and online database in a real life situation. The knowledge and experience gained during the project will help me immensely in days to come.

I would first of all like to thank for my project guide **Miss.S.R.Joshi, Mrs.P.M.Modak** for his kind and whole hearted guidance and inspiration as well as timely help for developing our project.

I have achieved satisfactory completion of my project only because of his continuously increasing despite all the practical difficulties we faced various phases of our project.

Also I will take this opportunity to thank all lectures of Computer Science department for giving valuable suggestion in completing my project.

At the end, I express our gratitude to my friends & my Parents who inspired me in this work without their inspiration the work was almost impossible.

Thank You.

PHASE COMPLETION TABLE

Class	**T.Y.Bsc**	**Roll No.**	**4038024**
Name Of Project		**Food For You**	

No	Task Performed	Proposed Date		Complete Date	Remark
1	**Analysis Phase**				
	1.System Analysis	20/11/2020	25/11/2020	25/11/2020	
	2. Study of Current System	25/11/2020	28/11/2020	28/11/2020	
	3. Documentation Current System	29/11/2020	03/12/2020	03/12/2020	
	4.CLass Diagram	04/12/2020	05/12/2020	05/12/2020	
	5. E-R Diagram	11/12/2020	12/12/2020	12/12/2020	
	6. Activity Digram	01/12/2020	01/12/2020	01/12/2020	
	7.Usecase diagram	02/12/2020	02/12/2020	02/12/2020	
	8.System requirements	10/12/2020	10/12/2020	10/12/2020	
2	**Design Phase**				
	1.System Design	12/12/2020	14/12/2020	14/12/2020	
	2 Database Design	15/12/2020	15/12/2020	15/12/2020	
	3.Form Design	18/12/2020	18/01/2020	18/12/2020	
	4. List and Described Class	21/12/2020	23/12/2020	23/12/2020	
	Implementation				
3	**Coding Phase**				
	1.Coding	26/12/2020	30/12/2020	30/12/2020	
	2.Construct Database	01/01/2020	02/01/2020	02/01/2020	
	3.Construct Screen Layout	03/01/2020	03/01/2020	11/01/2020	
	4. Construct Report	12/01/2020	10/01/2020	10/01/2020	
4	**Testing Phase**				
	1.Testing Phase	11/01/2020	13/01/2020	13/01/2020	
	2.Design and Plane Test Data	14/01/2020	16/0/2020	16/01/2020	
	3.Conduct Unit Testing	17/01/2020	19/01/2020	19/01/2020	
	4.Conduct Integrity Testing	20/01/2020	21/01/2020	21/01/2020	
	5.Conduct Test Data & Output	22/01/2020	24/01/2020	24/01/2020	

5	**Implementation Phase**				
	1.Implementation	25/01/2020	28/01/2020	28/01/2020	
	2.Install test version System	29/01/2020	31/01/2020	31/01/2020	
	3.Conduct System Testing	01/02/2020	04/02/2020	04/02/2020	
	4.Collect Feedback from User	05/02/2020	01/02/2020	01/02/2020	
	5.Developed Final System	02/02/2020	12/02/2020	12/02/2020	

Miss.S.R.Joshi
(Project Guide)

Mrs.P.M.Modak
(Project Head)

INDEX

CHAPTER-I

PROBLEM DOMAIN ANALYSIS

INTRODUCTION

The proposed system is for **"Food For You".** As present this system is not computerized. The need for computerization is in order to do overcome the human inefficiency and process/work of system is doing easily & fastly than old system.

I am trying my level best while developing this project report such that it will highlight each & every aspect of the system. I hope that the reader will try to make the best use of it.

EXISTING SYSTEM

Peoples go out to eat the food which they love. Sometimes due to lack of space in hotels and also because of crowd they have to wait in their to get their turn and order the food and enjoy the meal.

The drawbacks of the existing system are as follows:

- **Process is slow.**
- **Time consuming.**
- **Have to wait outside.**
- **Cant eat peacefully.**
- **Have to finish in time.**
- **Have to eat in very low space.**
- **There is no privacy while eating.**

Because of the above points the existing system is slowly irradiated. And in existing system cant tell how much time will get waste to just our favourite food.
Hence by my proposed system, everybody can enjoy their favourite meal at their home without any disturbance and full of privacy.

NEED OF COMPUTERIZATION

Computers are the devices that are the heart most of the automated circuits and networks. Computer technology has been under continuous improvement and has resulted in verification of very useful and fast devices. Better visual quality is also available. High capacity storage devices are available and they also take very small space. Software that is available now a day are providing very advanced feature like very advanced database structure and front end that provide excellent graphical interface.

1. All transaction is done on computer.
2. Information is stored on the computer aided storage devices
Like hard disk cassette.
3. Very high security is being provided to the information.
4. In case of loss of data it can be achieved by using backup.
5. Storage of hard disk, cassette is very easy and safe.
6. Operating is very fast.
7. Checking of any person status is very fast and easy.
8. All manual limitation was overcome in this system.
9. Paper work was considerably reduced to less than 10% of previous.

FACT FINDING TECHNIQUES

To study the system we require facts. Fact is also termed as data or information. A specific method used for collecting data is called fact finding techniques.

Interview:-

It is used to collect information from individuals or from groups. The interviewer must plan in advance and should know the problem under consideration. There are two types of interview.

1) Structured Interview
2) Unstructured Interview

1) Structured Interview:

In this type of technique fixed type of question are asked and that questions covered in specific area. In this type of technique interview period may be short.

2) Unstructured Interview:

In this type of technique different type of question are asked and there are specific area. In this type of fact finding technique extra information may be collected.

Questionnaire:-

Questionnaire is nothing but a question which asks to employees or to manager of the organization. Questionnaire may be used suppliments to interview. There are two types of questions which are considered in Questionnaire.

1) Open ended questions
2) Closed ended questions

Open ended questions:

Open ended questions are used to learn about options, feelings and general experience about the problem.

Closed ended questions:-

Close ended questions contain specific questions and responses from which respondent as to choose the best one.

Questions:

- When was the system established?
- Who is the owner of the system?
- Where is the main centers of the system?
- How many departments are present in your system?
- How many menus and staff are there are?
- From which city these distributors are belongs to?
- How many sub centers are available?
- How much the cost of every item varies?
- How many reports are generated?
- Do you have software?

Record review:-

Information related with the system may be present in the form of records, registers like books, magazines, newspaper, historical documents, letters, manually etc.

This kind of record review provided very valuable information to me about system, organization at various procedures and rules.

Observation:-

Is observed that the flow of documents, the process is carried out, steps followed by person involved and so on.

FEASIBILITY STUDY

When a project is initiated it is assumed that a new system is feasibility to develop & install.

Economic Feasibility:-

A determination of the economic feasibility of the project always requires a cost/benefit analysis.

Organizational & cultural feasibility:-

The proposed system is computer based software. During developing the system we follow all rules of computer based software. The system is developing under the phases of SDLC. So there is straight way for developing system. Thus the system is cultural & social feasibility.

Technological feasibility:-

In the current system all the work is manually done. This technical process has some problems like making mistakes during writing, data, losing some files etc.

For avoiding this problem they need one software to handle data. This software works as saving, adding, deleting the files & terms.

Schedule feasibility:-

The development of project schedule is always on activity with high risk. Schedule feasibility means the project can be completed on time.

Resource feasibility:-

The project management team must assess the availability of resources for the project.

SYSTEM REQUIREMENTS

Software:-

- Front End: Android Studio
- Back End : SQLite
- Operating System: Windows 10

Hardware:-

- Processor: Intel(R)Core (TM) i3
- Memory: 4 GB RAM
- Storage Device: Hard Disk 1 TB

TECHNICAL ANALYSIS

- **Merits, Performance, Reliability, Maintenance :**

1. This new System will able to satisfy most of the queries of eating food at home.

2. The system would have a much enhanced response time compared to exiting system

3. System could easily update all information of menues, their prices and new items.

4. Department it would be more reliable and cost required to update information would be negligible as compared to existing system.

5. Maintaining your order record becomes easier then exiting system. While going for answering a query the data become more efficiently processed.

- **Technology**

The use of newly computerized data entry & information retrieval will encourage the manager also he will be able to get relief from the Tedious job to maintain the all information. The security provided by this newly computerized system will be much higher then that of the existing system.

CHAPTER-II

SOFTWARE SPECIFICATION

PROPOSED SYSTEM

Proposed System:-

The use of newly proposed system is computerized. Data entry & information retrieval done on computer. This system will maintain all the oders of the customers in queued format and will safely deliver the food.

The security provided by this newly computerized system will be much higher than that of existing system.

Advantages of Proposed system:-

- **Daily Record:-**

By using proposed system we can maintain the users record and activities as of their likes and dislikes, Payments, reviews.

Accuracy:-

Accuracy is very important advantage of proposed system. The daily records can be stored in the system with accuracy and easily.

- **Security:-**

There is no any possibility of losing any important record or files. So this system gives high security of data.

- **Reducing time period:-**

The proposed system can easily deliver the food as per the scheduled time.

- **Easy to understand:-**

The proposed system will be easy to understand to every customer and user.

LIMITATIONS OF THE PRESENT SYSTEM

The current system is available in some specific areas.

- **Wastage of time:-**
  Customers have to wait until the table is empty and after that till their food arrives.

- **Lack of Proper Preservation:-**
  Everyone have to after the table gets empty with lack of security and precautions.
- **Mistakes:-**
  Anybody can receive wrong order due to lack of proper arrangement.

- **Less accuracy:-**
  It is not possible to tell any information at a time .They cannot improve the mistakes in previous record because it's very difficult to found previous records because there are many files of previous records.

- **Less security:-**
  In existing system security and precautions are very less.

PROGRAM FUNCTION EXPLANATION

This program is made to carry out the process. In This program the customer's information is taken at front office and room is allowed to him. Throw this program user can do following program.

LOGIN –

By clicking this button user can log in to their accound and order the food which they want.

JOIN US -

This allows new users to register to the application and enjoy the service which we provide. They just need to feel some information so that we can maintain the record of the user.

CHECKOUT-

By clicking checkout, users have to confirm their orders.

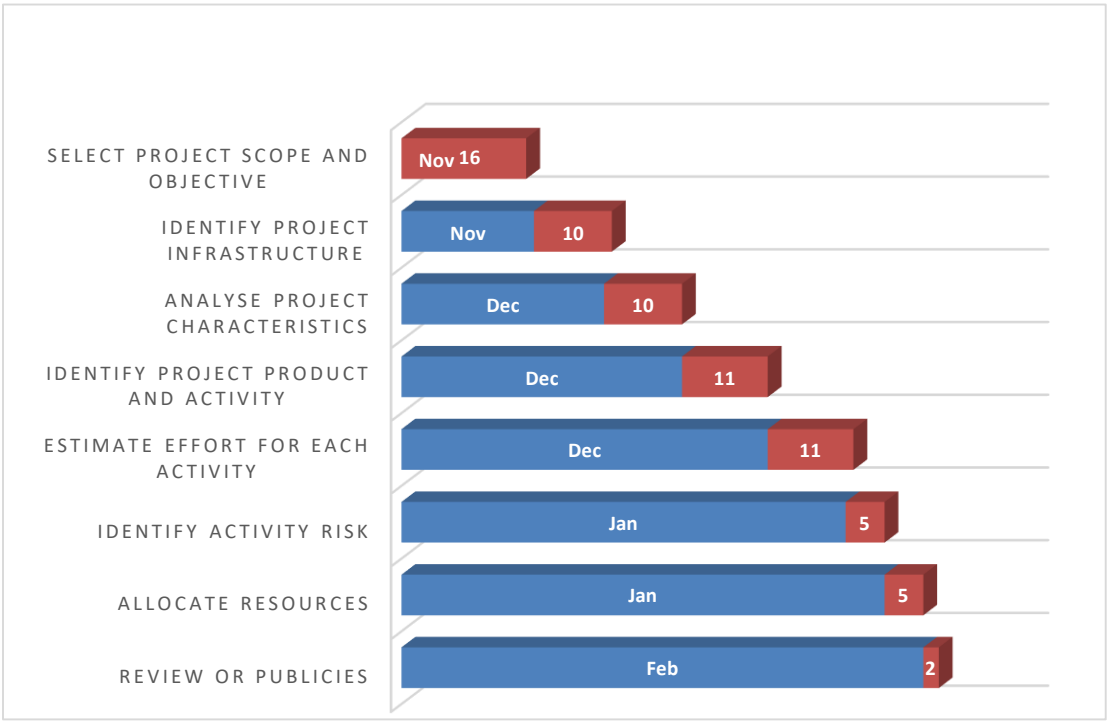
UPDATE-

If user wants to edit certain record from data grid & register then it is allowed .For this first user have to select particular record. Also we can save it.

EXIT -

To quit the program.

GANTT CHART


SELECT PROJECT SCOPE AND OBJECTIVE
Nov 16
IDENTIFY PROJECT INFRASTRUCTURE
Nov
10
ANALYSE PROJECT CHARACTERISTICS
Dec
10
IDENTIFY PROJECT PRODUCT AND ACTIVITY
Dec
11
ESTIMATE EFFORT FOR EACH ACTIVITY
Dec
11
IDENTIFY ACTIVITY RISK
Jan
5
ALLOCATE RESOURCES
Jan
5
REVIEW OR PUBLICIES
Feb
2

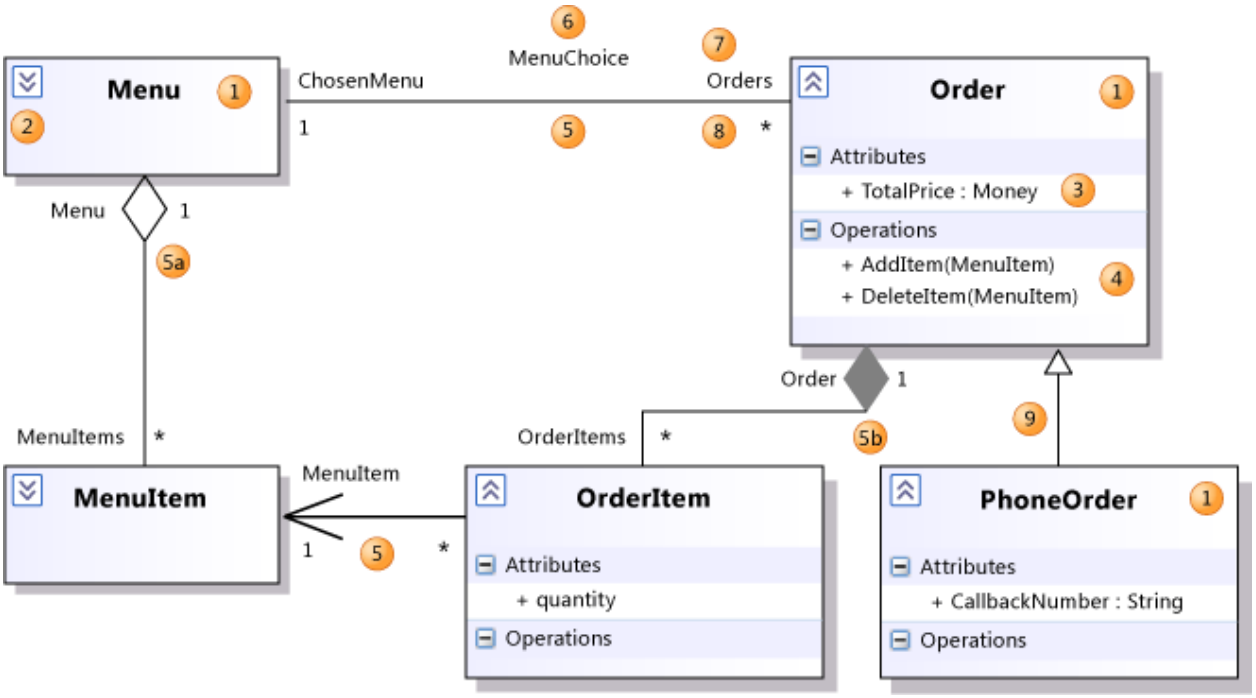
Event Table

Event	Trigger	Source	Activity	Response	Destination
Join us	User creates it account	Admin	Add new User	Users data gets saved	New user
Admin Adds a new menu to the menu card	Adds a new item	Menu	Adds a extra menu for the users	New menu	Menu card
Select Menu	Let you select a item	From menu card	Adds your orders to checkout	Checkout	Confirm order

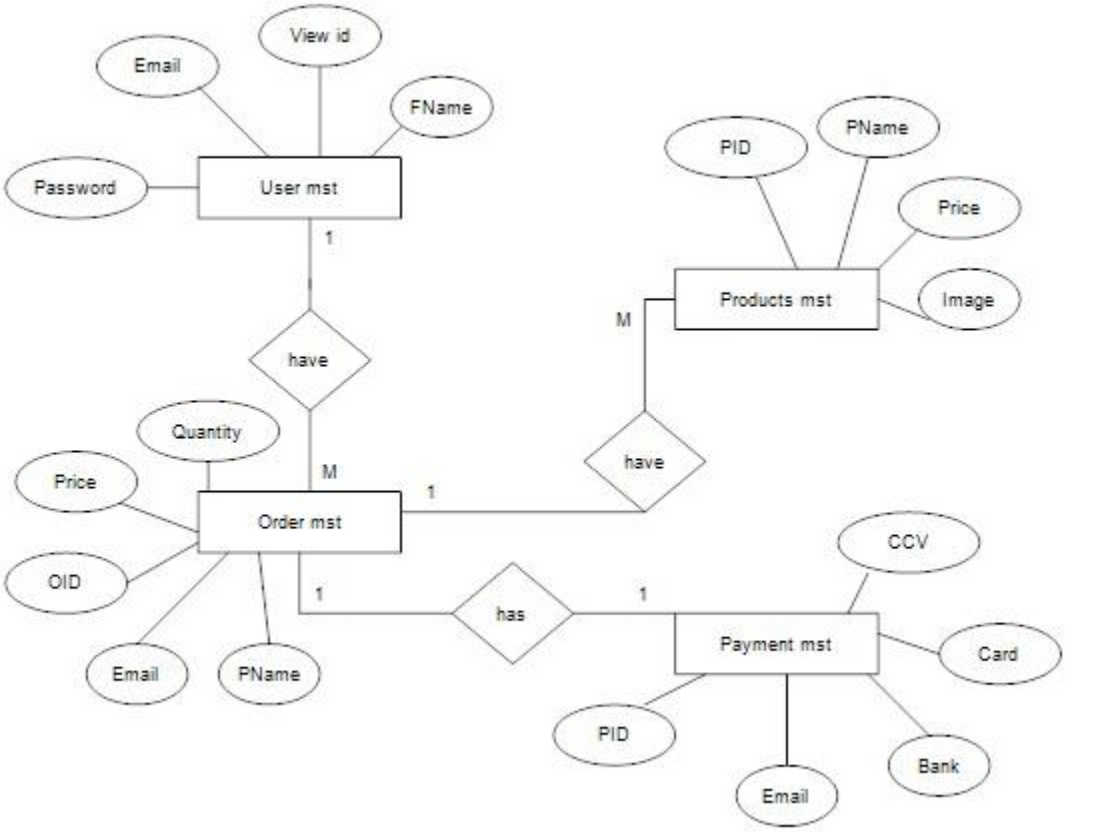
CHAPTER-III

ANALYSIS

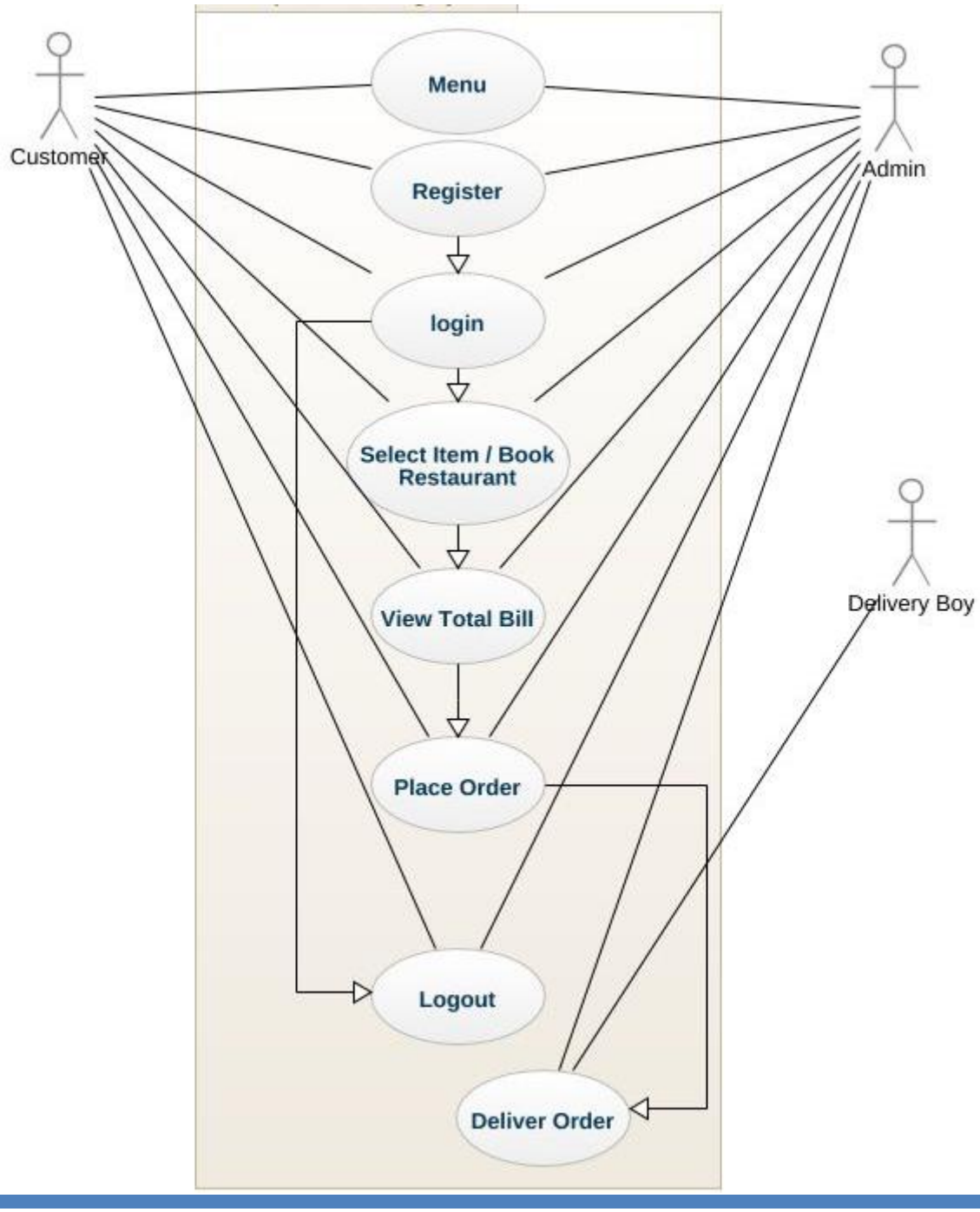
Class Diagram



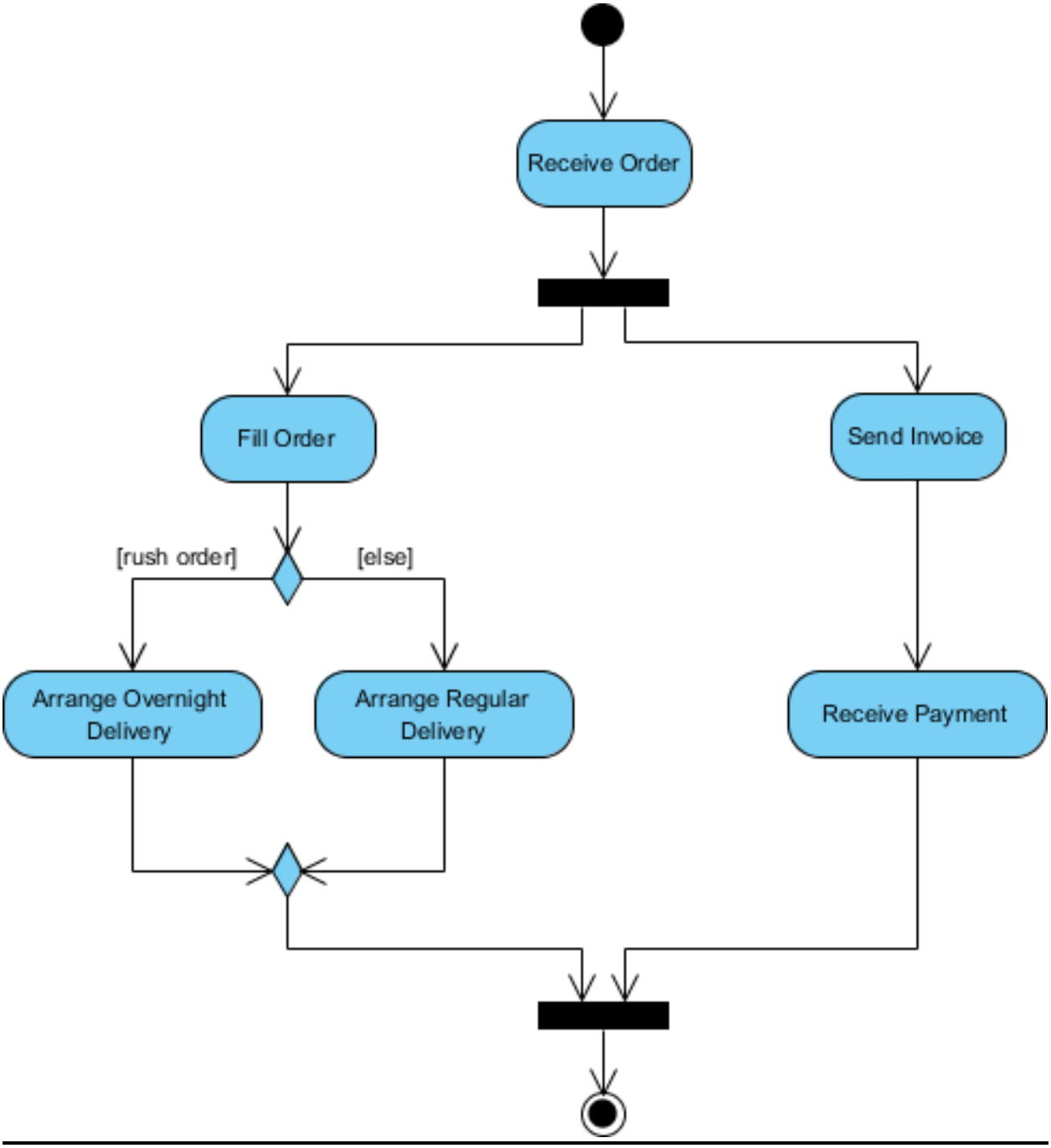
E-R Diagram


View id
Email
FName
Password
User mst
1
have
M
Products mst
PID
PName
Price
Image
Quantity
Price
M
1
Order mst
have
OID
Email
PName
1
has
1
Payment mst
CCV
Card
PID
Email
Bank

Use case Diagram



Activity Diagram



CHAPTER-IV

DESIGN

Table Design-

1.User Table-

Table: userTable

	userId	userFirstName	userLastName	userMobile	userEmail	userPassword	userAddress
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Datta	Ta	8308878765	dattapras...	dp21	204, empire ...
2	2	Rehan	Nav	1478523690	rehan@g...	rehan29	Nandgaon
3	3	Shubham	Patil	1234567890	chepya@...	chepya123	Olvan, Dajipur

2.Product table-

	productId	productTitle	productCategory	productPrice	productDescription	**productImage**
	Filter	Filter	Filter	Filter	Filter	Filter
1	0	Chicken ...	Biryani	120	Combination of Spic...	chickenbiryani....
2	1	Mutton Biryani	Biryani	180	Combination of Spic...	muttonbiryani....
3	2	Veg Thali	Thali	80	Pure veg meal	vegthali.png
4	3	Chicken Thali	Thali	120	Non-veg meal	chickenthali.png
5	4	Mutton Thali	Thali	160	Mutton thali with roti	muttonthali.png

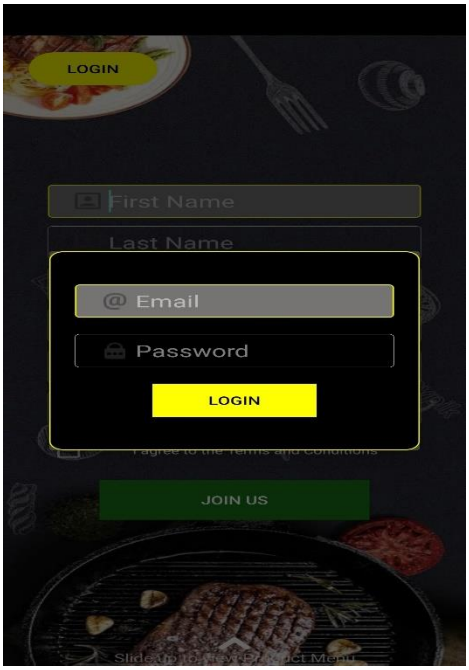
3. Order Table-

	orderId	memberId	idList	quantityList	totalPrice	orderStatus
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	50129038...	1	2	240	Delivered
2	2	50129038...	2	4	380	Delivered
3	3	50129038...	3	2	300	Pending

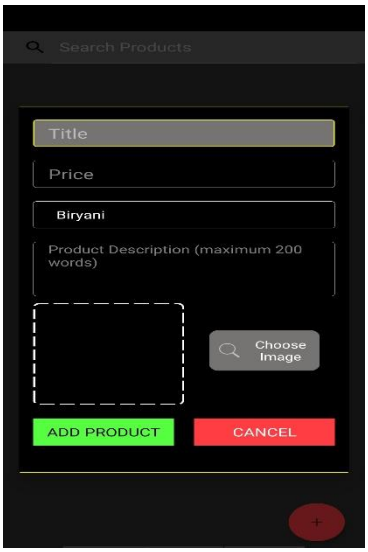
Screen Outputs-

1.Home Page

2.Log in page



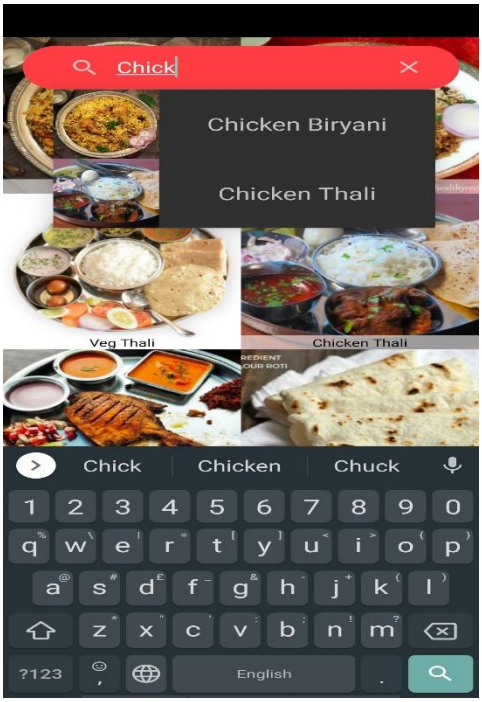
3.Add menu



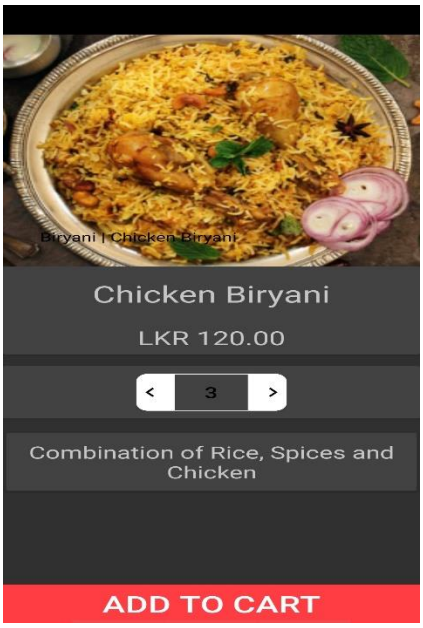
4.Menu admin panel

5.User menu panel

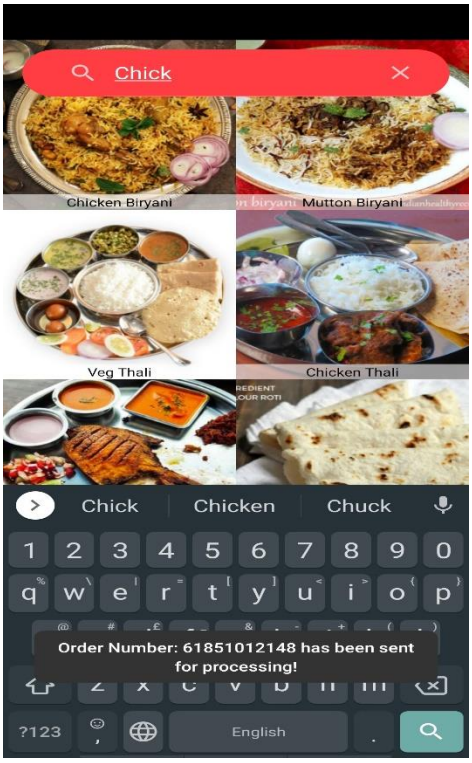
6.Search panel



7.User Cart panel

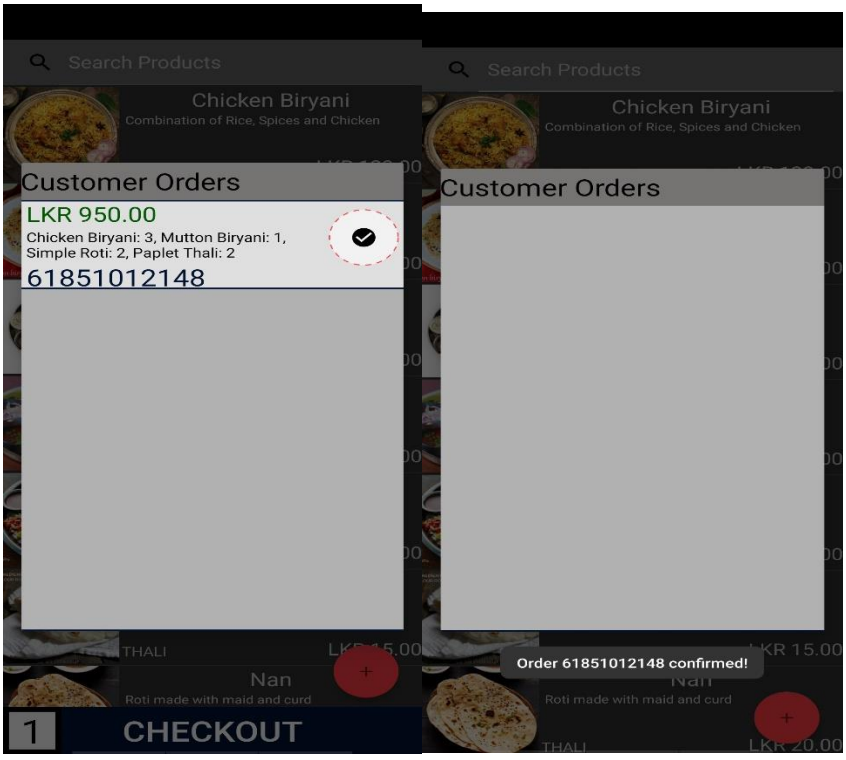


8.Order Processing prompt



9.Admin checkout option

10.Admin Confirm order



Code-

1.Login page-

```
package com.cmbpizza.razor.colombopizza;

import android.app.Activity;
import android.app.Dialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Point;
import android.graphics.drawable.Animatable;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.os.Handler;
import android.util.Patterns;
import android.view.Display;
import android.view.GestureDetector;
import android.view.MotionEvent;
import android.view.View;
import android.view.WindowManager;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

public class LoginScreen extends Activity implements View.OnClickListener,
GestureDetector.OnGestureListener{

    Button LoginAccessButton, MainLoginButton, RegisterButton;
    AutoCompleteTextView LoginEmail;
    EditText FirstName, LastName, Mobile, Email, Password, ConfirmPassword,
LoginPassword;
    CheckBox CheckTnC;
    ImageView LoginBackgroundImage, LoginBottomImage;
    GestureDetector gestureDetector;
    private static SQLLiteHelperProducts sqLiteHelper;
    public Dialog loginDialog; //creating a new instance of the Dialog class object,
and specifying its context as the current class

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login_screen);
        //this is to prevent the keyboard from showing when the app starts

this.getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_
```

```
HIDDEN);
        initializeComponents();
        initializeListeners();
        initializeAnimators();
        initializeGestures();
        sqlLiteDB();
        setBackgroundImage();
    }

    //initializing the buttons and views that is being used in this class
    private void initializeComponents(){
        //initializing and declaring button to this java class by referring to the
button ID specified in the XML
        LoginAccessButton = findViewById(R.id.btnLogin);
        RegisterButton = findViewById(R.id.btnRegister);
        LoginBottomImage = findViewById(R.id.login_bottom_image);
        LoginBackgroundImage = findViewById(R.id.loginBackgroundImage);

        FirstName = findViewById(R.id.txtFirstName);
        LastName = findViewById(R.id.txtLastName);
        Mobile = findViewById(R.id.txtMobileNumber);
        Email = findViewById(R.id.txtEmail);
        Password = findViewById(R.id.txtPassword);
        ConfirmPassword = findViewById(R.id.txtConfirmPassword);
        CheckTnC = findViewById(R.id.chkTermsAndConditions);

        loginDialog = new Dialog(LoginScreen.this);
        //specifying the layout/content of the dialog by referring to the XML file
        loginDialog.setContentView(R.layout.login_dialog_screen);
        MainLoginButton = loginDialog.findViewById(R.id.btnLoginMain);
        LoginEmail = loginDialog.findViewById(R.id.txtLoginEmail);
        LoginPassword = loginDialog.findViewById(R.id.txtLoginPassword);
        loginDialog.setCancelable(true); //specifying that whether the dialog box is
able to be cancelled
        loginDialog.setCanceledOnTouchOutside(true);//specifying that the dialog box
should close when and if the user touches outside it
    }

    //used to initialize the animations that is used in this activity
    private void initializeAnimators(){
        Drawable drawable =  LoginBottomImage.getDrawable();
        if(drawable instanceof Animatable){
            ((Animatable)drawable).start();
        }
    }

    //setting a click event listener to these buttons
    private void initializeListeners(){
        LoginAccessButton.setOnClickListener(this);
        RegisterButton.setOnClickListener(this);
        MainLoginButton.setOnClickListener(this);
    }

    //used to initialize the gestures used in this activity
```

```
    private void initializeGestures(){
        //this gesture detector will monitor for any gestures performed in this
activity
        //the on touch listener is used to restrain the gesture monitoring activity
to
        //be constrained to a particular view in this case the bottom of the screen
layout)
        gestureDetector = new GestureDetector(LoginScreen.this ,this);
        findViewById(R.id.login_bottom_image).setOnTouchListener(new
View.OnTouchListener() {
            @Override
            public boolean onTouch(View view, MotionEvent motionEvent) {
                return gestureDetector.onTouchEvent(motionEvent);
            }
        });
    }

    //set login's background image
    private void setBackgroundImage(){
        /* adapt the image to the size of the display */
        Display display = getWindowManager().getDefaultDisplay();
        Point size = new Point();
        display.getSize(size);
        Bitmap bmp = Bitmap.createScaledBitmap(BitmapFactory.decodeResource(
                getResources(),R.drawable.pizza_main_background),size.x,size.y,true);

    /* fill the background ImageView with the resized image */
        LoginBackgroundImage.setImageBitmap(bmp);
    }

    //this method is used to set the onClick listeners that is used in this activity
    @Override
    public void onClick(View v){
        switch (v.getId()){
            case R.id.btnLogin:
                loginDialog.show();
                break;
            case R.id.btnRegister:
                registerUser();
                break;
            case R.id.btnLoginMain:
                checkUser();
                break;
        }
    }

    //this event will fire when the user touches any point inside a certain
element(view) on the screen
    @Override
    public boolean onTouchEvent(MotionEvent event) {
        // TODO Auto-generated method stub
        return gestureDetector.onTouchEvent(event);
    }
```

```
    @Override
    public boolean onDown(MotionEvent motionEvent) {
        return true;
    }

    @Override
    public void onShowPress(MotionEvent motionEvent) {

    }

    @Override
    public boolean onSingleTapUp(MotionEvent motionEvent) {
        return false;
    }

    @Override
    public boolean onScroll(MotionEvent motionEvent, MotionEvent motionEvent1, float
v, float v1) {
        return false;
    }

    @Override
    public void onLongPress(MotionEvent motionEvent) {

    }

    //this event will fire when the user does the fling motion on the screen
    //when a user drags and lifts their finger quickly
    @Override
    public boolean onFling(MotionEvent start, MotionEvent end, float v, float v1) {
        if(start.getY() > end.getY()){
            //if the user flings from bottom to top (slide up)
            Intent newIntent = new Intent(LoginScreen.this, UserMenu.class);
            startActivity(newIntent);
            overridePendingTransition(R.anim.layout_slide_up,
R.anim.back_layout_slide_up);
            return true;
        } else if(start.getX() > end.getX() || start.getY() < end.getY() ||
start.getX() < end.getX()){
            //ignore other fling motions in other directions
            return false;
        } else{
            return false;
        }
    }

    //this method is used to create a database
    private void sqlLiteDB() {
        sqLiteHelper = new SQLLiteHelperProducts(LoginScreen.this, "ProductDB", null,
1);
        sqLiteHelper.dataQuery("CREATE TABLE IF NOT EXISTS userTable (userId INTEGER
PRIMARY KEY AUTOINCREMENT, userFirstName VARCHAR, userLastName VARCHAR, userMobile
VARCHAR, userEmail VARCHAR, userPassword VARCHAR)");
```

```
    }

    private void checkUser(){
        String AdminEmail = "admin";
        String AdminPass = "pass";
        if(inputLoginValidation()){
            String Email = LoginEmail.getText().toString();
            String Password = LoginPassword.getText().toString();
            if(Email.equals(AdminEmail) && Password.equals(AdminPass)){
                Intent newIntent = new Intent(LoginScreen.this, AdminMenu.class);
                startActivity(newIntent);
                clearEditTextValues();
                loginDialog.cancel();
            } else {
                if(sqLiteHelper.checkEmail(Email)){
                    if(sqLiteHelper.checkPassword(Email, Password)){
                        int UserId = sqLiteHelper.getUserId(Email);
                        Intent newIntent = new Intent(LoginScreen.this,
UserMenu.class);
                        newIntent.putExtra("userId", UserId);
                        startActivity(newIntent);
                        clearEditTextValues();
                        loginDialog.cancel();
                    } else {
                        Toast.makeText(LoginScreen.this, "Your password is incorrect,
please try again", Toast.LENGTH_SHORT).show();
                        LoginPassword.requestFocus();
                    }
                } else {
                    Toast.makeText(LoginScreen.this, "Your email does not exist",
Toast.LENGTH_SHORT).show();
                    LoginEmail.requestFocus();
                }
            }
        }
    }

    private void clearEditTextValues(){
        LoginEmail.getText().clear();
        LoginPassword.getText().clear();
    }

    private void registerUser(){
        if(inputValidation()){
            String UserFirstName = FirstName.getText().toString();
            String UserLastName = LastName.getText().toString();
            String UserMobile = Mobile.getText().toString();
            String UserEmail = Email.getText().toString();
            String UserPassword = Password.getText().toString();
            sqLiteHelper.registerUser(UserFirstName, UserLastName, UserMobile,
UserEmail, UserPassword);
            Toast.makeText(this, "Registration Successful",
Toast.LENGTH_SHORT).show();
            clearRegistrationForm();
```

```
            int UserId = sqLiteHelper.getUserId(UserEmail);
            Intent newIntent = new Intent(LoginScreen.this, UserMenu.class);
            newIntent.putExtra("userId", UserId);
            startActivity(newIntent);
            overridePendingTransition(R.anim.layout_slide_up,
R.anim.back_layout_slide_up);
        }
    }

    private void clearRegistrationForm() {
        FirstName.getText().clear();
        LastName.getText().clear();
        Mobile.getText().clear();
        Email.getText().clear();
        Password.getText().clear();
        ConfirmPassword.getText().clear();
        CheckTnC.setSelected(false);
    }

    //this method will check the input fields and validate them before submitting
them
    private boolean inputValidation() {
        boolean valid;
        String UserFirstName = FirstName.getText().toString();
        String UserLastName = LastName.getText().toString();
        String UserMobile = Mobile.getText().toString();
        String UserEmail = Email.getText().toString();
        String UserPassword = Password.getText().toString();
        String UserConfirmPassword = ConfirmPassword.getText().toString();
        Boolean TnCConfirmed = CheckTnC.isChecked();
        if (UserFirstName.isEmpty()) {
            valid = false;
            FirstName.requestFocus();
            Toast.makeText(this, "Please enter your First Name",
Toast.LENGTH_SHORT).show();
        } else if (UserLastName.isEmpty()) {
            valid = false;
            LastName.requestFocus();
            Toast.makeText(this, "Please enter your Last Name",
Toast.LENGTH_SHORT).show();
        } else if (UserMobile.isEmpty()) {
            valid = false;
            Mobile.requestFocus();
            Toast.makeText(this, "Please enter Mobile Number",
Toast.LENGTH_SHORT).show();
        } else if (UserEmail.isEmpty()) {
            valid = false;
            Email.requestFocus();
            Toast.makeText(this, "Please enter your Email Address",
Toast.LENGTH_SHORT).show();
        } else if (!Patterns.EMAIL_ADDRESS.matcher(UserEmail).matches()) {
            valid = false;
            LoginEmail.requestFocus();
            Toast.makeText(this, "Please enter a valid Email Address",
```

```
Toast.LENGTH_SHORT).show();
        } else if (UserPassword.isEmpty()) {
            valid = false;
            Password.requestFocus();
            Toast.makeText(this, "Please provide a Password",
Toast.LENGTH_SHORT).show();
        } else if (UserConfirmPassword.isEmpty()) {
            valid = false;
            ConfirmPassword.requestFocus();
            Toast.makeText(this, "Re-enter your Password in the Confirm Password
field", Toast.LENGTH_SHORT).show();
        } else if (!UserPassword.equals(UserConfirmPassword)) {
            valid = false;
            ConfirmPassword.requestFocus();
            Toast.makeText(this, "Your Confirm Password value does not match your
Password value", Toast.LENGTH_SHORT).show();
        } else if (!TnCConfirmed) {
            valid = false;
            Toast.makeText(this, "You have to accept out Terms and Conditions before
registering", Toast.LENGTH_SHORT).show();
        } else {
            valid = true;
        }
        return valid;
    }

    private boolean inputLoginValidation() {
        boolean valid;
        String Email = LoginEmail.getText().toString();
        String Password = LoginPassword.getText().toString();
        if (String.valueOf(Email).isEmpty()) {
            valid = false;
            LoginEmail.requestFocus();
            Toast.makeText(this, "Please enter your Email Address",
Toast.LENGTH_SHORT).show();
        } else if (String.valueOf(Password).isEmpty()) {
            valid = false;
            LoginPassword.requestFocus();
            Toast.makeText(this, "Please enter your Password",
Toast.LENGTH_SHORT).show();
        } else if(Email.equals("admin")) {
            valid = true; //this is to ensure that when the admin enters the admin
username (which is not an email), that no validation error occurs
        } else if (!Patterns.EMAIL_ADDRESS.matcher(Email).matches()) {
            valid = false;
            LoginEmail.requestFocus();
            Toast.makeText(this, "Please enter a valid Email Address",
Toast.LENGTH_SHORT).show();
        } else {
            valid = true;
        }

        return valid;
```

```
    }
}
```

2.Register New User-

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.cmbpizza.razor.colombopizza.LoginScreen">

    <ImageView
        android:id="@+id/loginBackgroundImage"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <LinearLayout
        android:id="@+id/logo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:id="@+id/btnLogin"
            android:layout_width="100dp"
            android:layout_height="40dp"
            android:layout_marginEnd="40dp"
            android:layout_marginLeft="40dp"
            android:layout_marginRight="40dp"
            android:layout_marginStart="20dp"
            android:layout_marginTop="20dp"
            android:background="@drawable/login_button_style"
            android:text="@string/btnLogin"
            android:textAllCaps="true"
            android:textColor="#000000" />
    </LinearLayout>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="140dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:layout_marginEnd="35dp"
            android:layout_marginLeft="35dp"
            android:layout_marginRight="35dp"
            android:layout_marginStart="35dp"
            android:layout_marginTop="40dp"
            android:orientation="vertical">
```

```
        <EditText
            android:id="@+id/txtFirstName"
            android:layout_width="fill_parent"
            android:layout_height="40dp"
            android:background="@drawable/text_background_style"
            android:drawableStart="@drawable/ic_action_user_name"
            android:gravity="center_vertical"
            android:hint="@string/txtFirstName_hint"
            android:inputType="textCapWords"
            android:paddingEnd="5dp"
            android:paddingStart="15dp"
            android:textColor="@color/white"
            android:textSize="20sp"
            android:textStyle="normal"
            android:typeface="normal" />

        <EditText
            android:id="@+id/txtLastName"
            android:layout_width="fill_parent"
            android:layout_height="40dp"
            android:layout_marginTop="10dp"
            android:background="@drawable/text_background_style"
            android:drawableStart="@drawable/ic_action_user_name"
            android:gravity="center_vertical"
            android:hint="@string/txtLastName_hint"
            android:inputType="textCapWords"
            android:paddingEnd="5dp"
            android:paddingStart="15dp"
            android:textColor="@color/white"
            android:textSize="20sp"
            android:textStyle="normal"
            android:typeface="normal" />

        <EditText
            android:id="@+id/txtMobileNumber"
            android:layout_width="fill_parent"
            android:layout_height="40dp"
            android:layout_marginTop="10dp"
            android:background="@drawable/text_background_style"
            android:drawableStart="@drawable/ic_action_mobile"
            android:gravity="center_vertical"
            android:hint="@string/txtMobileNumber_hint"
            android:inputType="phone"
            android:maxLength="10"
            android:paddingEnd="5dp"
            android:paddingStart="15dp"
            android:textColor="@color/white"
            android:textSize="20sp"
            android:textStyle="normal"
            android:typeface="normal" />

        <AutoCompleteTextView
            android:id="@+id/txtEmail"
            android:layout_width="fill_parent"
```

```
        android:layout_height="40dp"
        android:layout_marginTop="10dp"
        android:background="@drawable/text_background_style"
        android:drawableStart="@drawable/ic_action_email"
        android:gravity="center_vertical"
        android:hint="@string/txtEmail_hint"
        android:inputType="textEmailAddress"
        android:paddingEnd="5dp"
        android:paddingStart="15dp"
        android:textColor="@color/white"
        android:textSize="20sp"
        android:textStyle="normal"
        android:typeface="normal" />

    <EditText
        android:id="@+id/txtPassword"
        android:layout_width="fill_parent"
        android:layout_height="40dp"
        android:layout_marginTop="10dp"
        android:background="@drawable/text_background_style"
        android:drawableStart="@drawable/ic_action_password"
        android:fontFamily="sans-serif"
        android:gravity="center_vertical"
        android:hint="@string/txtPassword_hint"
        android:inputType="textPassword"
        android:paddingEnd="5dp"
        android:paddingStart="15dp"
        android:singleLine="false"
        android:textColor="@color/white"
        android:textSize="20sp"
        android:textStyle="normal"
        android:typeface="normal" />

    <EditText
        android:id="@+id/txtConfirmPassword"
        android:layout_width="fill_parent"
        android:layout_height="40dp"
        android:layout_marginTop="10dp"
        android:background="@drawable/text_background_style"
        android:drawableStart="@drawable/ic_action_confirm_password"
        android:fontFamily="sans-serif"
        android:gravity="center_vertical"
        android:hint="@string/txtConfirmPassword_hint"
        android:inputType="textPassword"
        android:paddingEnd="5dp"
        android:paddingStart="15dp"
        android:textColor="@color/white"
        android:textSize="20sp"
        android:textStyle="normal"
        android:typeface="normal" />

    <CheckBox
        android:id="@+id/chkTermsAndConditions"
        android:layout_width="match_parent"
```

```
                android:layout_height="match_parent"
                android:layout_marginTop="15dp"
                android:gravity="center"
                android:text="@string/rdbAccept_hint"
                android:textSize="12sp" />

            <Button
                android:id="@+id/btnRegister"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginEnd="40dp"
                android:layout_marginLeft="40dp"
                android:layout_marginRight="40dp"
                android:layout_marginStart="40dp"
                android:layout_marginTop="20dp"
                android:background="@drawable/register_button_style"
                android:text="@string/btnRegisterUser"
                android:textAllCaps="true"
                android:textColor="#ffffff" />
        </LinearLayout>
    </ScrollView>

    <ImageView
        android:id="@+id/login_bottom_image"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="20dp"
        android:contentDescription="@string/imgSlideArrowAnimation"
        android:scaleType="fitCenter"
        android:src="@drawable/swipe_arrow_animation" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="20dp"
        android:layout_alignParentBottom="true"
        android:gravity="center"
        android:text="@string/txtSlideArrowLabel" />
</RelativeLayout>
```

3.Add new product-

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:theme="@style/Theme.AppCompat.NoActionBar"
    tools:context="com.cmbpizza.razor.colombopizza.AdminMenu">

    <SearchView
        android:id="@+id/adminSearchView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:background="@color/white"
        android:backgroundTint="@color/greyTransparent"
        android:closeIcon="@drawable/ic_close_black_24dp"
        android:focusable="false"
        android:goIcon="@drawable/ic_commit_search_black_24dp"
        android:iconifiedByDefault="false"
        android:inputType="textCapWords"
        android:queryHint="@string/txtSearchViewHint"
        android:searchIcon="@drawable/ic_search_black_24dp" />

    <ListView
        android:id="@+id/adminListView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="50dp" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/add_product_fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@id/layoutRelativeOrderCheckButton"
        android:layout_alignParentEnd="true"
        android:layout_gravity="bottom|end"
        android:layout_marginBottom="60dp"
        android:layout_marginEnd="20dp"
        android:elevation="6dp"
        android:src="@drawable/ic_action_add_product"
        app:backgroundTint="@color/pastelRed"
        app:backgroundTintMode="src_over"
        app:fabSize="normal"
        app:pressedTranslationZ="12dp"
        app:rippleColor="@color/white"
        app:tint="@color/black" />

    <RelativeLayout
        android:id="@+id/layoutRelativeOrderCheckButton"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="0dp">

        <LinearLayout
            android:id="@+id/layoutLinearOrderCheckButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:orientation="horizontal"
            android:weightSum="1">

            <Button
                android:id="@+id/adminOrderCheckButton"
                android:layout_width="0dp"
                android:layout_height="50dp"
                android:layout_gravity="center"
                android:layout_weight="1"
                android:background="@drawable/admin_order_check_button_style"
                android:gravity="center"
                android:text="@string/txtCheckoutButton"
                android:textAllCaps="true"
                android:textColor="@color/white"
                android:textSize="30sp" />
        </LinearLayout>

        <RelativeLayout
            android:id="@+id/layoutRelativeOrderButtonBadge"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignStart="@+id/layoutLinearOrderCheckButton"
            android:layout_alignTop="@id/layoutLinearCheckoutButton"
            android:visibility="visible">

            <ImageView
                android:layout_width="50dp"
                android:layout_height="50dp"
                android:layout_gravity="center"
                android:background="@drawable/order_counter_badge_background_style"
/>


            <TextView
                android:id="@+id/txtOrderButtonBadge"
                android:layout_width="50dp"
                android:layout_height="50dp"
                android:gravity="center"
                android:textColor="@color/black"
                android:textSize="35sp" />
        </RelativeLayout>
    </RelativeLayout>
</RelativeLayout>
```

CHAPTER-V

SYSTEM IMPLEMENTATION

Implementation and Maintainance

An important phase in the system life cycle is the successful implementation of the new system design. Implementation simply means converting a new system design into operation. This involves creating computer compactible files, training the operating task, & installing hardware, terminals before the system is up & running.

An important phase in the system life cycle is the successful implementation of the new system design. Implementation simply means converting a new system design into operation. This involves creating computer compatible files, training the operation staff, and installing hardware, terminals before the system is up and running in our project implementation is done in following manner.

1.Installation of Software:

First of all, I prepared apk file with the help of android studio menu. Then this apk file is uploaded to the playstore or send to the other user. On this disk I also copy Database of all the procedures that are required to retrieves the data from database. Then this all files are installed on client machine.

2.Training The Operating Staff:

All the users are given demonstration after copying all the files. In this demonstration all the input variables are taken and project is executed. After this I give them chance to run project by themselves.

CHAPTER-VI

BIBLOGRAPHY

References and Bibliography

Books refered for this Project are:

④★ Android Studio Technologies,Prof.JaylaxmiShrinivasan , VipulPublication,Mumbai.

④★ Prof. S. Parthasarthy, Prof .B.W.Khalkar,System Analysis,

Design and Introduction To Software Engineering (SADSE)

④★ Satzinger,Jakson and Burd ,System Analysis And Design in a

Changing World.

④★ Julia Case Bradley & Anita C.Millspaugh,Programming in

Visual Basic 6.0,TATA McGraw-HILL.

④★ Software Engineering a Practitioners Approach, TATA McGraw

Hill , 5'th Edition ,Rogre S Pressman.

④★ System Analysis And Design ,Galgotia 2'nd

Edition,EliasM.Awad.

④★ Front End: Android Studio

④★ Back End : SQL- SQLite database

④★ Operating System: Windows 10

THANK YOU!