Food Zone System

College of Science, Al Zulfi
Department of Computer Science and Information

Graduation Project
Submitted in partial fulfillment of the requirements for the award of
Bachelor degree of the Majmaah University
(Semester 2, 2020)

Submitted by:

Student: Faisal Mohammed Al Anezi   ID: 361101872

Student: Nawaf Mana Al Shammari   ID: 361102529

Under the Supervision of:
Dr. Khalid Nazim S.A
Assistant Professor, Department of CSI
College of Science, Az-Zulfi campus
Majmaah University, KSA

Abstract

Many restaurants is storing all of their data in manual way. They have huge number of customers daily.

Food Zone System is an application which will help restaurant to optimized and control over their restaurants. For the management delivery, table reservation.

This application helps the restaurants to do all functionalities more accurately and faster way. Food Zone System reduces manual works and improves efficiency of restaurant. This application is helping Food Zone System to maintain the stock and cash flows and there are many more functionalities.

Acknowledgements

Thanks for all, thanks for all staff in college, special thanks for our supervisor.

When we need help always we find the person standing by our side, and give us all

information and guidance to achieve this project, this person is our supervisor.

Certificate by student

MAJAMAAH UNIVERSITY

COLLEGE OF SCIENCE AL ZULFI

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION

(CERTIFICATE BY STUDENT)

This is to certify that the project titled "Title of the project" submitted by   me

,Faisal Mohammed Al Anezi , 361101872, under the supervision of Dr. Khalid

Nazim S.A., for award of Bachelor degree of the Majmaah University carried out

during the Semester 1, 2019-20 embodies my original work.

Signature in full: -------------------------------

Name in block letters: FAISAL MOHAMMED AL ANEZI

Student ID   : 361101872

Date:

Table of Content

Table of Figures

Abstract

Many restaurants are storing all of their data in manual way. They have huge number of customers daily.

Food Zone System is an application which will help restaurant to optimized and control over their restaurants. For the management delivery, table reservation.

This application helps the restaurants to do all functionalities more accurately and faster way. Food Zone System reduces manual works and improves efficiency of restaurant. This application is helping Food Zone System to maintain the stock and cash flows and there are many more functionalities.

Summary

Food Zone System is an application which will help restaurant to optimized and control over their restaurants.

For the management delivery, table reservation, We have been to many restaurants, to understand their process of maintaining database and the level of efficiency they have in their system and drawbacks of their existing systems. After visiting many such centers and stores we thought of developing an application which will overcome the drawbacks of the existing systems.

1 INTRODUCTION

Food Zone System is an application which will help restaurant to optimized and control over their restaurants. For the management delivery, table reservation.

This application helps the restaurants to do all functionalities more accurately and faster way. Food Zone System reduces manual works and improves efficiency of restaurant. This application is helping Food Zone System to maintain the stock and cash flows and there are many more functionalities.

- To store records.

- Control orders and services.

- Billings.

- Helps Manager to control delivery.

- Reserve table.

1.1 Goals

The main goal is to maintain the restaurant's functions in an effective and accurate manner and also it is reducing the use of manual entries. This software helps food orders to maintain day to day records in system. It is keeping a proper record of the database.

1.1 PROBLEM STATEMENT:

Many restaurants are storing all their data in manual way. They have huge number of customers daily. So because large number of customers, they need the help of some features so they can maintain and stores the records accurately. For managers it is difficult to view the tables, orders.

They need full-fledged software to maintain their day to day transactions, orders and also regular update on records, cash transaction, customers feedbacks etc. In the existing system, entering all the details are done manually, it is taking lots of time and also there are chances for mistakes.

1.2 Research Objectives

Food Ordering is the profession of managing a restaurant. The management is mainly divided into three broader section.

- Floor management: 'Floor management' includes managing staff who give services to customers and allocate the duties of opening and closing restaurant. The manager is responsible for making sure his or her staff is following the service standards and health and safety regulations. The manager is the most important person in the front-of-the-house environment, since it is up to him or her to motivate the staff and give them job satisfaction. The manager also looks after and guides the personal well-being of the staff, since it makes the work force stronger and more profitable.

- Kitchen Management: 'Kitchen Management' includes the managing staff working in the kitchen, especially the head chef. The kitchen is the most important part of the business and the main reason customers patronize the restaurant. Managing the kitchen staff helps to control food quality. As most commercial kitchens are a closed environment, the staff may become bored or tired from the work. Without proper management, this often results in an inconsistent food product. Kitchen management

involves most importantly, cost control and budgeting. Meeting KPI's are a must for a restaurant to survive. Head chefs must instill and teach money management to apprentices.

This is as important as teaching the art and skills of cookery.

- Administration: 'Administration' includes stock controlling, scheduling rotations, budgeting the labor costs, balancing cost and profit according to seasonality, surveying and hiring staff, and maintenance of the commercial kitchen equipment.

2 SYSTEM REQUIREMENTS

2.1 LITERATURE REVIEW

Sources of Information

We have been to many restaurants, to understand their process of maintaining database and the level of efficiency they have in their system and drawbacks of their existing systems. After visiting many such centers and stores we thought of developing an application which will overcome the drawbacks of the existing systems.

Primary sources:

- Web Sites
- Discussion with owners of many restaurants.
- Suggestions from friends.

Secondary sources:

- Reference Materials

2.2 EXISTING SYSTEM:

2.3 Description:

Many Restaurants stores and maintain their day to day transactions manually. But some of them are having automation system which is helping them to store the data. But such restaurants are storing the information about the orders and the customer information. They don't have facility to store the information of feedbacks and favorite orders of customers over some period of time.

Restaurants are having standalone applications so at one time, they have the facility of many screens or many operations which is happening at one time. So they are storing them and then at last, the restaurant managers will able to see the data of last day.

The software which restaurants are using is very costly and their maintenance which is very high.

The software which we tried and get the information was called "Food Delivery ". This software which free for download but the restaurant has to pay after sometime.

- Food Delivery:

Food Delivery aims to save trees by letting you manage reservations on a computer and, in the process, lets you keep track of regular customers. For such a high price tag, we weren't that impressed by its design and brief trial period; however, it gets the job done.

The user interface is pretty plain, but it's intuitive and easy to navigate even without the Help feature. Should you need it, it also contains helpful tutorials and tips for getting started and pinpointing any issues you might have. Command buttons for viewing and making reservations line the top of the window, along with buttons for viewing customer history and reports. We were able to jump right in and quickly create a new reservation. It was simply a matter of clicking the Make Reservation button and inputting all the required information, including name, time of reservation, and duration. There are also fields for entering the customer's address, phone, e-mail, and business information. It's not required, but will certainly come in handy for taking advantage of the customer history and report features. Our reservation immediately appeared in the program's main screen. Using the buttons at the bottom of the window, we were able to edit the entry, change the status (arrived, departed, seated, at bar, etc.), and even add a message.

Though we would have liked to have seen a longer trial period than the 10 uses offered, we were still able to get a grip on the program's functions and navigation. If you're looking for a way to enter and manage reservations, this is a good tool to have on-hand.

Drawbacks:

- This software is basically used only for reservation means table booking. So if we want to just order some food or store any feed backs then it wont

  be any helpful. At last the restaurants have to store by themselves which will became no use of software.

- The user interface of the application is also not that much attractive. It is made in .net so the restaurant must have to install any windows operating system to install this application.

- So from the restaurants point of view, they are able to store only one kind of information. There is no security feature also.

- If any of party order is cancelling at the last moment, it will make a variation in the already created records and also will causes the wastage of foods.

- Many of the systems will not store the budget details for a long time.

- This will creates lot of mistakes like misspellings, calculation problems, duplicate entries etc.

- It is difficult for Managers to supervise all the sides of restaurants like kitchen, floor, and counter simultaneously.

- There is no functionality to get the updated details at all time to the Owners and Managers from all the branches.

2.4 PROPOSED SYSTEM:

The proposed system helps in many ways. It helps to do billing very easily. Account maintenance also becomes easier. They can keep track of their purchases of inventories, customer feedback, sales of foods, and account details ,table reservation etc. The software is provided with the facilities to find out the favorite food of the customers, or customers to add or modify and delete their feedbacks and suggestions. It helps in managing data of different types of orders like party order, home delivery or the normal order. Managing data of daily customers, managing

data of daily expenses. It eliminates the drawbacks of existing system and also includes some more features.

2.4.1 **Advantages**:

FZ will stores all the records permanently and also gives the raises and deviations in the status of the restaurant in the markets.

- Managers can view the floor side and the counter side details simultaneously.
- Any cancelations of parties are also updated.
- Also keeps the record of food items prepared and the sales of food and also the record of balance food.
- Less use of manual work.
- Profits and losses will be updated every month and also shows the variations.
- All the food order details like the order types (normal, home delivery, party order etc.) are stored daily.

2.5   FUNCTIONAL SPECIFICATION

Food Zone is having many modules, which make the software more efficient and user friendly. The modules make the maintenance of the database easier. Every module is divided on the basis of the scenarios. The main three scenarios are

- Manager
- Floor
- Table

The different modules in this project are described

below:

- Food Items
- Order.
- Tables.
- Cancellations.
- Home delivery

Menu Module:

In this module the number of food, their prices, offers etc. like details will stores.

- Insert food item.
- Insert price.

   Delete food item.
- Delete price.
- Modify price.

Order Module:

In this module the type of order, the amount etc. like details will stores.

- Insert Order.
- Insert type of order.
- Insert price.
- Delete order.
- Delete price.

- Modify price.

Cancelations Module:

In this module the canceled foods details are stores.

- Insert Canceled order.
- Delete canceled order.

Home Delivery:

In this module, there will section which will store the data about home delivery orders.

Add order.

- Add bill.
- Delete order.
- Delete bill.

Tables:

In this module, there will section which will store the data about Tables orders.

Add Tables.

- Add bill.
- Delete Table.
- Delete bill.

3   Research Methodology:

XAMPP

XAMPP is an easy to install Apache distribution containing MySQL, PHP.

XAMPP is really very easy to install and to use - just download, extract and start.

XAMPP for Windows

This version contains: Apache, MySQL, PHP + PEAR, Perl, mod_php, mod_perl, mod_ssl, OpenSSL phpMyAdmin,

Android Studio with java Support .

- Apache 2.4.4

- MySQL 5.5.32

- PHP 5.4.16

- phpMyAdmin 4.0.4

- FileZilla FTP Server 0.9.41

3.1 HARDWARE REQUIREMENT:

- Processor: 3.2MHz Intel Pentium i5 or equivalent or new

- Disk Space: 1000 MB or more

3.2 SOFTWARE REQUIREMENT:

- Operating System: No particular OS required because PHP is platform independent.

- Software: XAMPP, MySQL

3.3 Programming Language

- HTML
- CSS
- Jquery
- PHP
- MYSQL
- Java

3.4 Management Plan

Gantt Chart

	Conclusion and deliver final report	Complet chapter 3 and deliver it	System design	Complete chapter 2 and deliver it	System analysis and specificat...	Deliver Chapter 1	problem statement and objectives
■ Duration (days)	8	13	3	5	5	6	6

PROBLEM STATEMENT AND OBJECTIVES
COMPLETE CHAPTER 2 AND DELIVER IT
CONCLUSION AND DELIVER FINAL REPORT

3.5   FEASABILTY STUDY:

FOOD ZONE

إجمالي الإجابات	نسبة الإجابات			هل الفكرة قابلة للتنفيذ ام لا 1.
1	100.00%		نعم	1
0	0.00%		لا	2
0	0.00%		ربما	3

1	تم الإجابة عليه	0	معدل الرضا:	0	انحراف معياري:	1	متوسط:	**تحليل**
0	تم تخطيه			0	خطأ معياري:	0	اختلاف:	

إجمالي الإجابات	نسبة الإجابات			هل يمكن نشر ثقافة طلب الطعام عن طريق التطبيق 2.
1	100.00%		نعم	1
0	0.00%		لا	2
0	0.00%		بنسبة ضئيلة	3

1	تم الإجابة عليه	0	معدل الرضا:	0	انحراف معياري:	1	متوسط:	**تحليل**
0	تم تخطيه			0	خطأ معياري:	0	اختلاف:	

بنسبة كم سوف ترشح التطبيق للاستخدم لدى اصدقائك 3.

			نسبة الإجابات	إجمالي الإجابات
1	10%		0.00%	0
2	20%		0.00%	0
3	30%		0.00%	0
4	40%		0.00%	0
5	50%		0.00%	0
6	100%		100.00%	1
7	لن انصح احد باستخدامه		0.00%	0

تحليل	متوسط:	6	انحراف معياري:	0	معدل الرضا:	83.33	تم الإجابة عليه	1
	اختلاف:	0	خطأ معياري:	0			تم تخطيه	0

---

4. هل يتشابه التطبيق مع باقى التطبيقات الموجودة

			نسبة الإجابات	إجمالي الإجابات
1	نعم		0.00%	0
2	لا		100.00%	1
3	لا اعرف		0.00%	0

تحليل	متوسط:	2	انحراف معياري:	0	معدل الرضا:	50	تم الإجابة عليه	1
	اختلاف:	0	خطأ معياري:	0			تم تخطيه	0

---

5. كم تتوقع تقريبا تتوقع نسبة الاقبال على استخدام التطبيق من قبل الطلاب

			نسبة الإجابات	إجمالي الإجابات
1	نسبة كبيرة		100.00%	1
2	نسبة ضئيلة		0.00%	0

5. كم تتوقع تقريبا تتوقع نسبة الاقبال على استخدام التطبيق من قبل الطلاب

إجمالي الإجابات	نسبة الإجابات			
0	0.00%		اعتقد انه لن يجد اقبال	3
1	تم الإجابة عليه	0 معدل الرضا:	0 انحراف معياري:	1 :متوسط **تحليل**
0	تم تخطيه		0 خطأ معياري:	0 :اختلاف

6. هل انت ممن يطلبوا الطعام اون لاين بكثرة

إجمالي الإجابات	نسبة الإجابات			
1	100.00%	▮▮▮▮▮	نعم	1
0	0.00%		لا	2
0	0.00%		غير متفاعل مع طلب الطعام اون لاين	3
1	تم الإجابة عليه	0 معدل الرضا:	0 انحراف معياري:	1 :متوسط **تحليل**
0	تم تخطيه		0 خطأ معياري:	0 :اختلاف

7. من وجهة نظرك هل يقبل الطلاب على طلب الطعام اون لاين ام انهم يفضلون الذهاب الى المطعم مباشرة

إجمالي الإجابات	نسبة الإجابات			
1	100.00%	▮▮▮▮▮	الكثير يطلبه اون لاين	1
0	0.00%		الكثير يذهب الى المطعم	2
0	0.00%		لا اعرف	3
1	تم الإجابة عليه	0 معدل الرضا:	0 انحراف معياري:	1 :متوسط **تحليل**
0	تم تخطيه		0 خطأ معياري:	0 :اختلاف

			نسبة الإجابات	إجمالي الإجابات
8. هل تفضل حجز طاولة اون لاين ام تفضل الذهاب الى المطعم مباشرة				
1	الحجز اون لاين		0.00%	0
2	الذهاب الى المطعم		0.00%	0
3	لا شئ مماسبق		0.00%	0
			تم الإجابة عليه	0
			تم تخطيه	1

18

4 DESIGN SPECIFICATION

4.1 MODULES DESIGN:

Food Zone System, we have divided mainly in three modules.

.Manager Side:

- It includes stock controlling, scheduling rotations, labor costs, balancing cost and profit according to seasonality, surveying and hiring staff, and maintenance of the commercial kitchen equipment.

1. Table Side

- There will be details about orders, details about waiter and billing facility will be there.

4.2   SYSTEM DESIGN

4.3   DATAFLOW DIAGRAM:
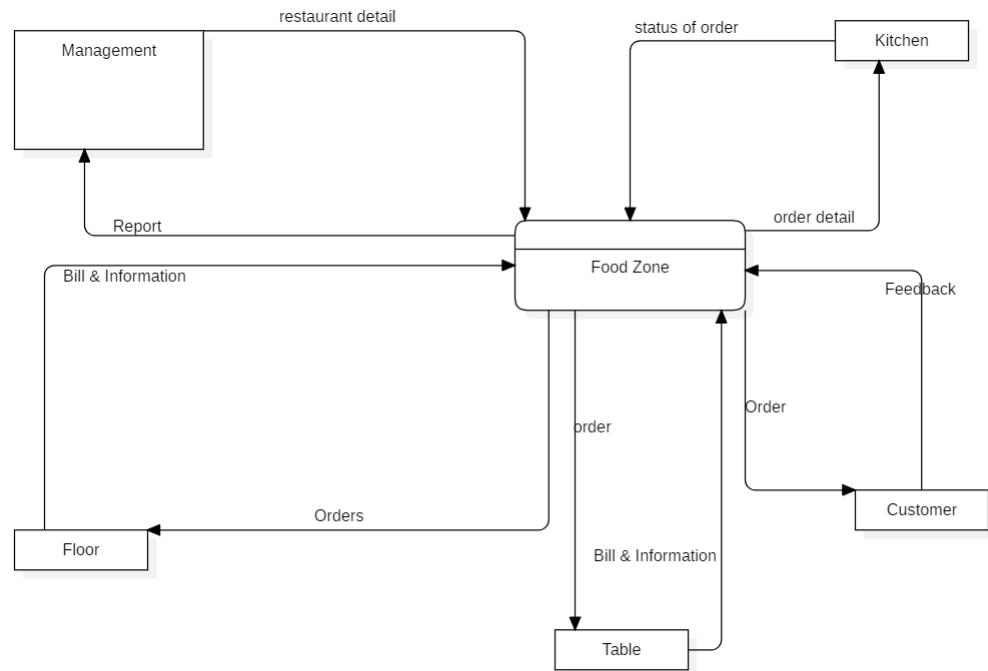
Figure 4-1

4.4   Login Process:
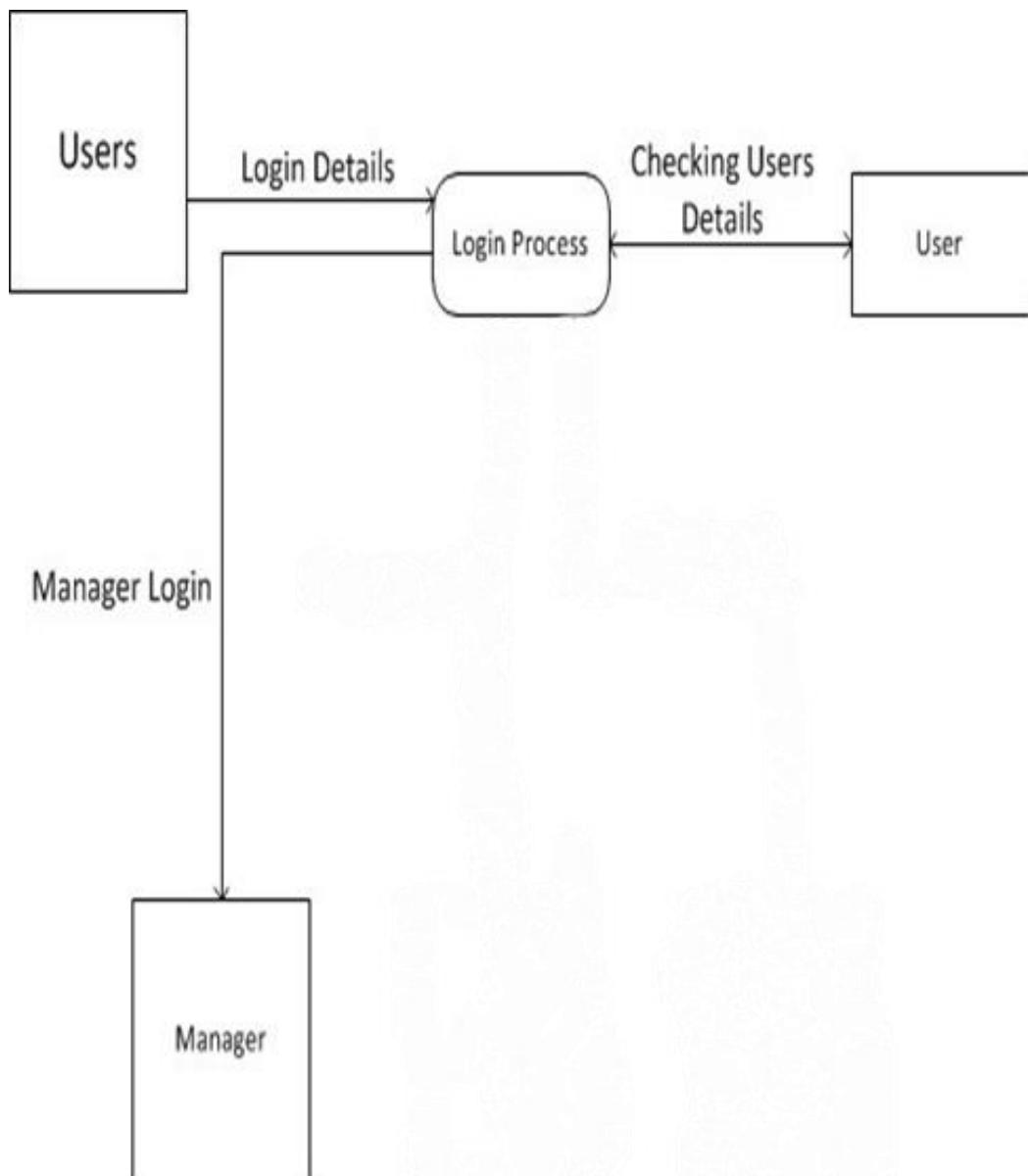
Figure 4-2

4.5   Manager Side:
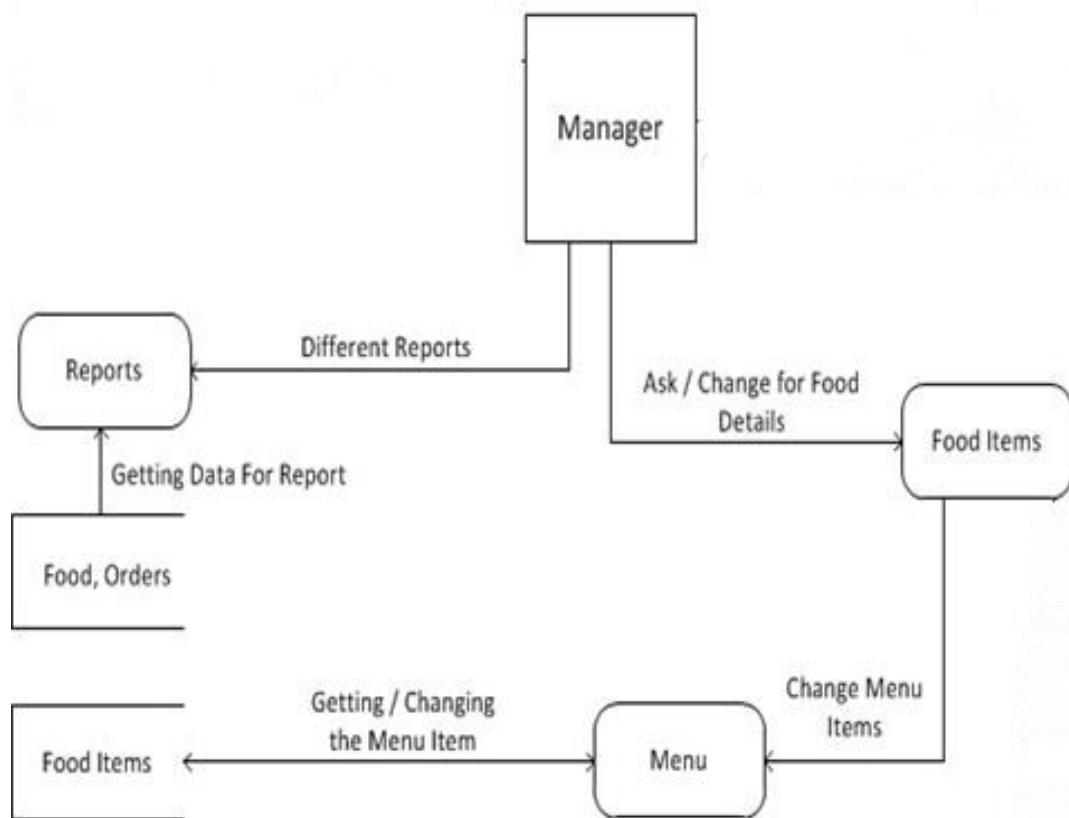
Figure 4-3

4.6   order Side:
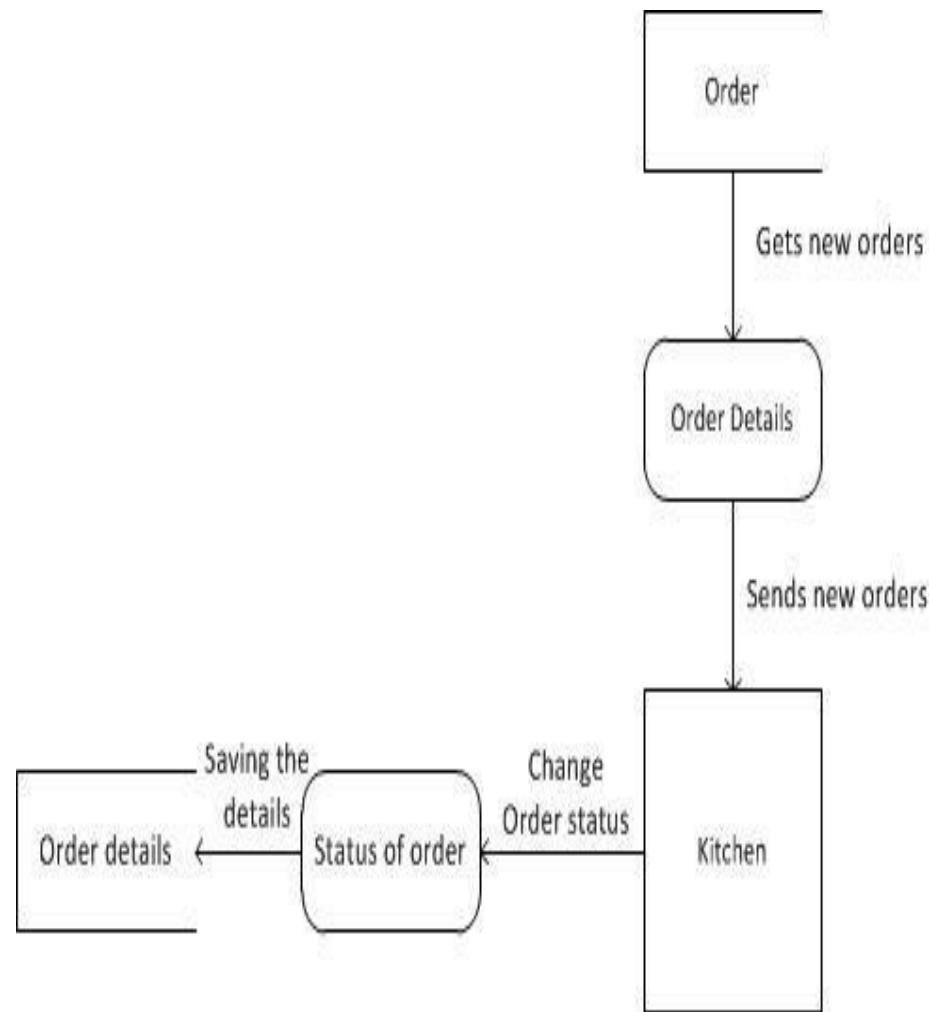
Figure 4-4

4.7　Table Side
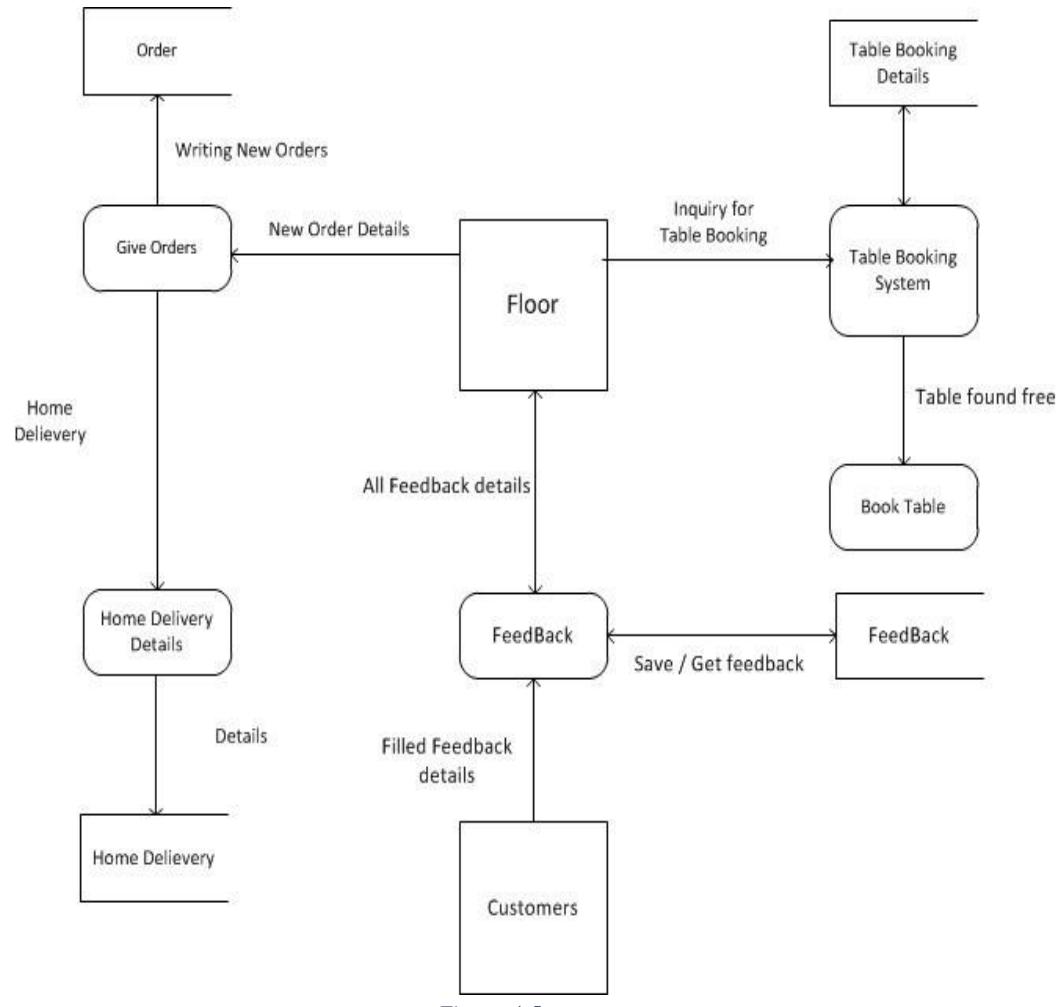
Figure 4-5

4.8 ER DIAGRAM
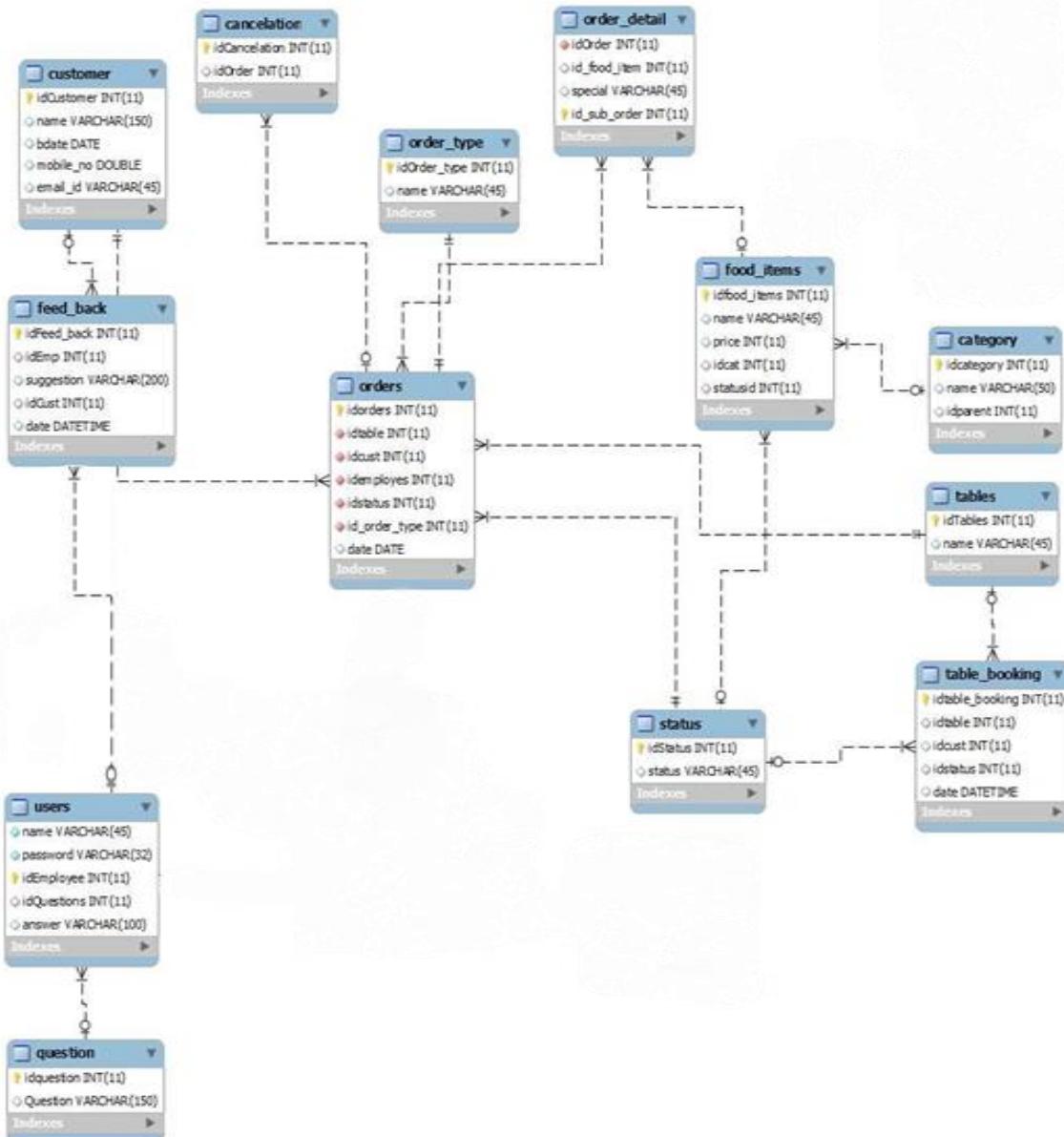
Figure 4-6

4.9   DATA DICTIONARY

1. Users:

Field Name	Data Type	Size
Name	Varchar	20
Password	Varchar	32
idEmployee	INT	
idQuestion	INT	
Answer	VARCHAR	100

2. Cancelation:

Field Name	Data Type	Size
idCancelation	INT	
idOrder	INT	

3. Category:

Field Name	Data Type	Size
idCategory	INT	
Name	Varchar	32
idParent	INT	

4. Customer:

Field Name	Data Type	Size
idCustomer	INT	
Name	Varchar	150
Bdate	Date	

Field Name	Data Type	Size
Mobile_no	Double	
Email_id	Varchar	45

5. Feed_Back:

Field Name	Data Type	Size
idFeed_Back	INT	
Suggestion	Varchar	200
idEmployee	INT	
idCustomer	INT	

6. Food Items:

Field Name	Data Type	Size
idFood_item	INT	
Name	Varchar	45
Price	INT	
idCategory	INT	

7. Order Type:

Field Name	Data Type	Size
idOrder_type	INT	
Name	Varchar	45

8. Order:

Field Name	Data Type	Size

idOrder	INT	
idTable	INT	
idCustomer	INT	
idEmployee	INT	
idStatus	INT	
idOrder_type	INT	
Date	Date	

9. Order Details:

Field Name	Data Type	Size
idSub_Order	INT	
idOrder	INT	
idFood_item	INT	
Special	Varchar	45

10. Status:

Field Name	Data Type	Size
idStatus	INT	
Status	Varchar	45

11. Tables:

Field Name	Data Type	Size
idTable	INT	
Name	Varchar	45

12. Table Booking

Field Name	Data Type	Size
idTable_booking	INT	
idTable	INT	
idCustomer	INT	
idStatus	INT	
Date	Date	

13. Questions:

Field Name	Data Type	Size
idQuestion	INT	
Questions	Varchar	100

4.10 interfaces
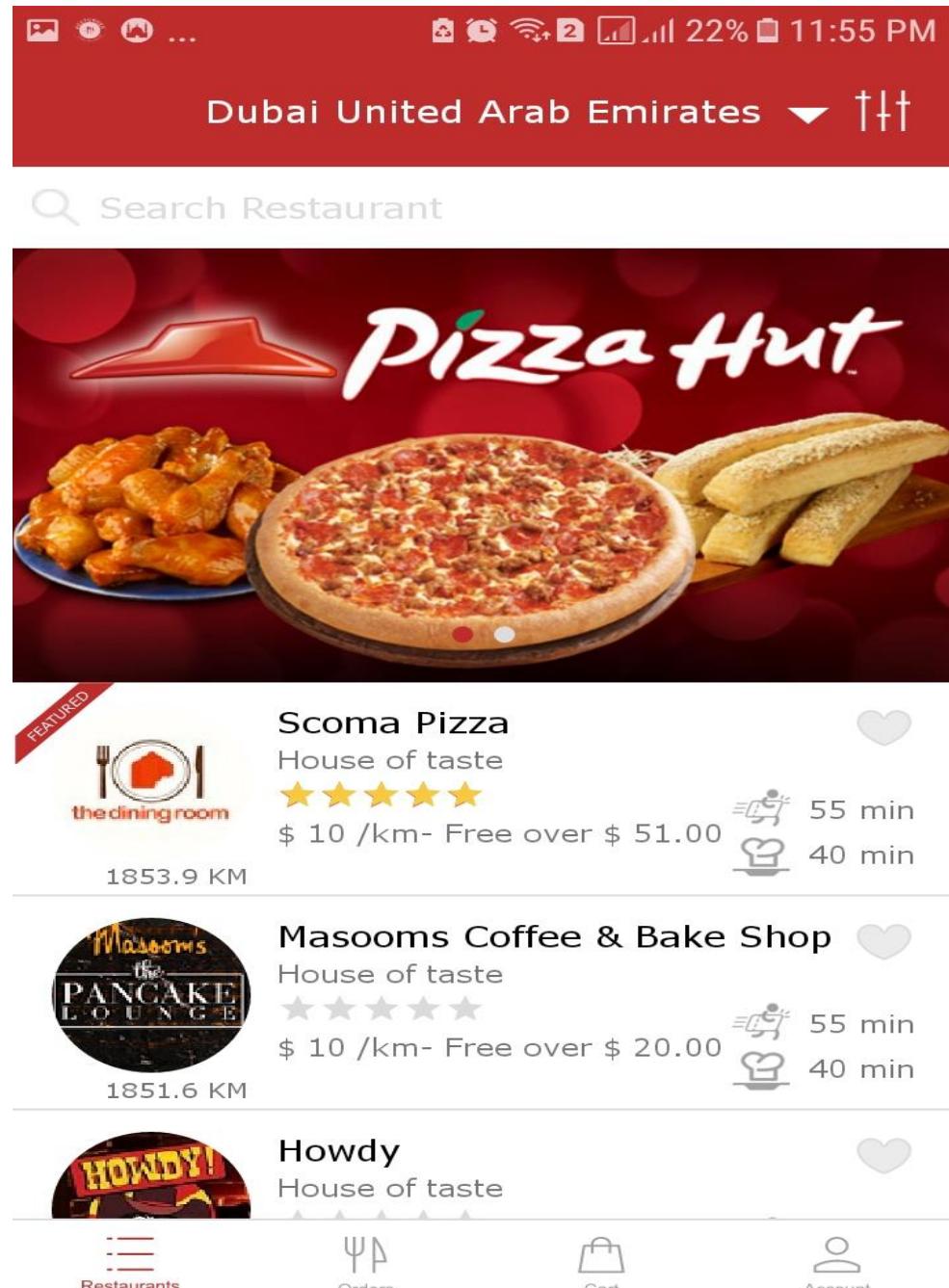
Figure 4-7

FOOD ITEMS SCREEN
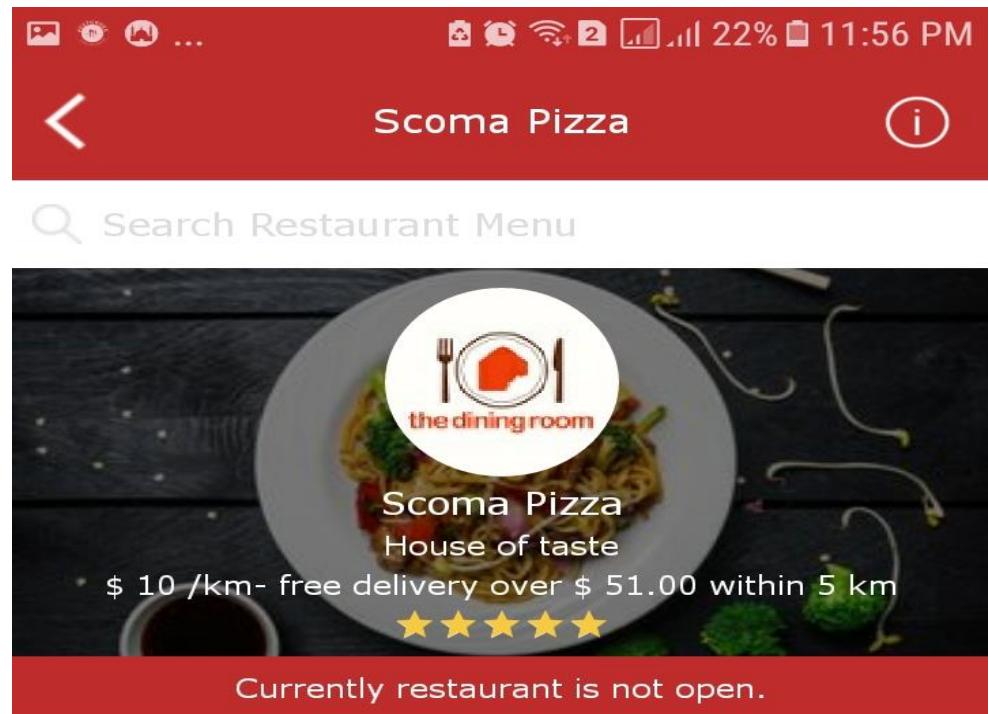
Figure 4-8

ORDERS SCREEN
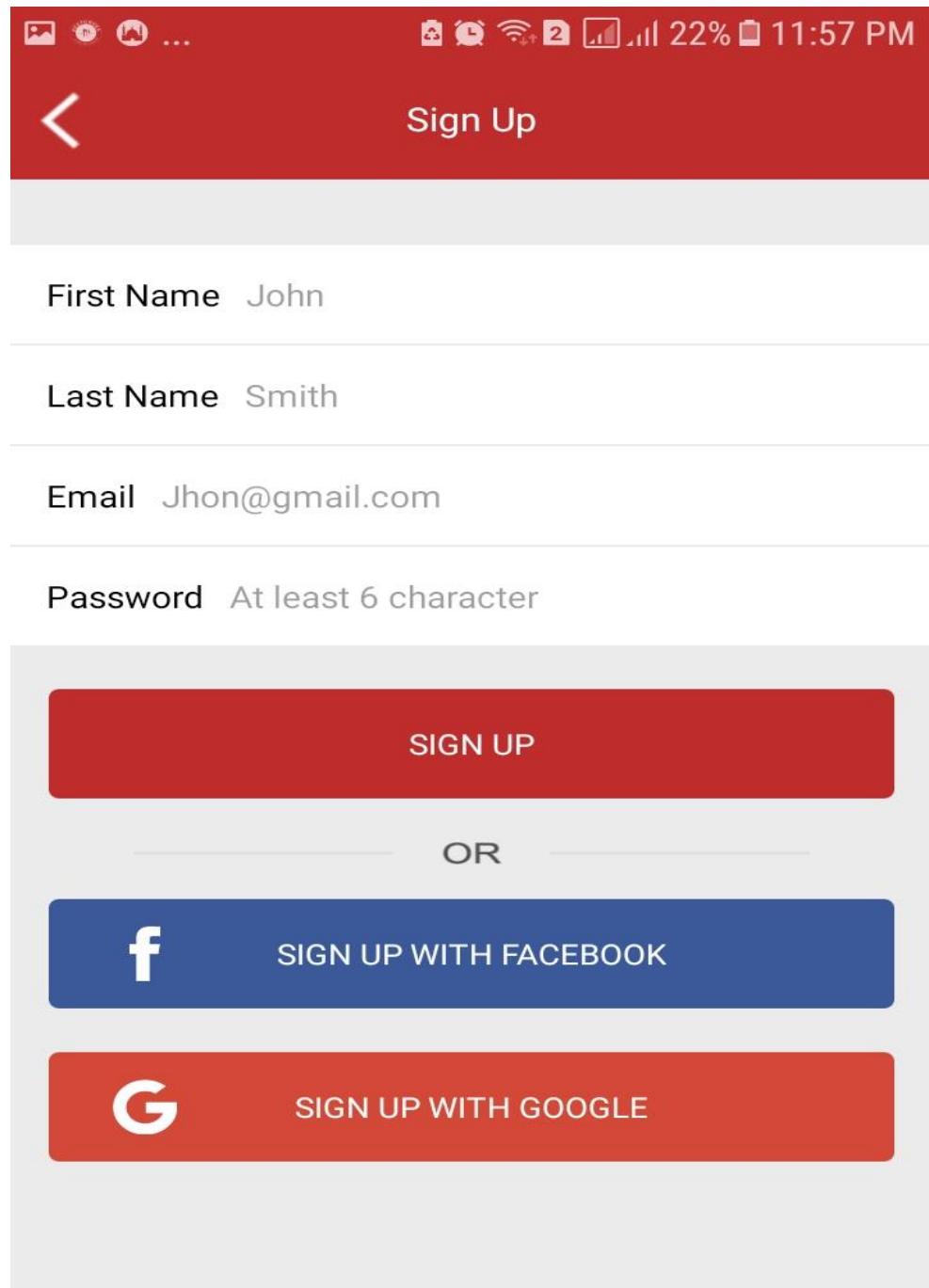
Figure 4-9

REGISTER SCREEN

Figure 4-10
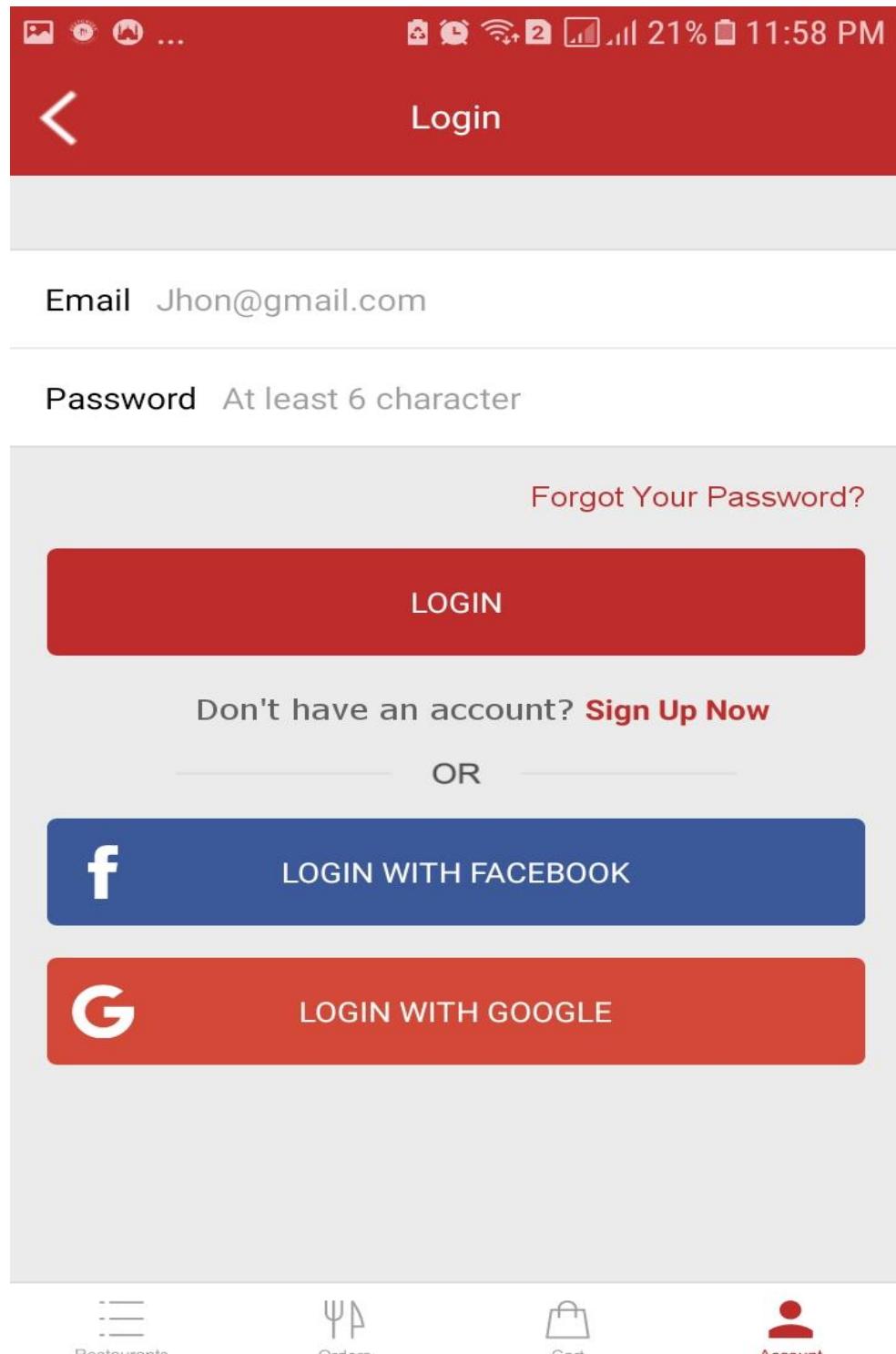
LOGIN SCREEN

5   Implementation

5.1.   Introduction

5.2.   Procedures

Connection to database procedures :

```
public          static          String          baseURL          =
"https://api.foodzone.ofertasbooking.com/mobileapp api/api/
";
 public          static          String          imgBaseURL          =
"https://api.foodzone.ofertasbooking.com/mobileapp_api/";
```

```
    public static final String LOGIN_URL = baseURL+"login";

    public     static     final     String     showCountries     =
baseURL+"showCountries";
    public     static     final     String     Verify_URL     =
baseURL+"verifyPhoneNo";
    public     static     final     String     SignUp_URL     =
baseURL+"registerUser";


    public     static     final     String     forgotPassword     =
baseURL+"forgotPassword";
    public static final String verifyforgotPasswordCode   =
baseURL+"verifyforgotPasswordCode";
    public     static     final     String     changePasswordForgot     =
baseURL+"changePasswordForgot";


    public     static     final     String     SHOW RESTAURANTS     =
baseURL+"showRestaurants";
    public     static     final     String     SHOW RESTAURANT MENU     =
baseURL+"showRestaurantsMenu";
    public     static     final     String     ADD FAV RESTAURANT     =
baseURL+"addFavouriteRestaurant";
    public     static     final     String     SHOW FAV RESTAURANT     =
baseURL+"showFavouriteRestaurants";
```

```java
        public    static    final    String    SHOW_SLIDER    =
baseURL+"showAppSliderImages";
        public    static    final    String    SHOW_ORDERS    =
baseURL+"showOrders";
        public    static    final    String    SHOW_DEALS    =
baseURL+"showDeals";

        public    static    final    String    CHANGE_PASSWORD    =
baseURL+"changePassword";
        public    static    final    String    EDIT_PROFILE    =
baseURL+"editUserProfile";


        public    static    final    String    ADD_PAYMENT_METHOD    =
baseURL+"addPaymentMethod";
        public    static    final    String    GET_PAYMENT_METHID    =
baseURL+"getPaymentDetails";
        public    static    final    String    ADD_DELIVERY_ADDRESS    =
baseURL+"addDeliveryAddress";
        public    static    final    String    GET_DELIVERY_ADDRESES    =
baseURL+"showDeliveryAddresses";

        public    static    final    String    SHOW COUNTRIES LIST    =
baseURL+"showCountries";
        public    static    final    String    SHOW_ORDER_DETAIL    =
baseURL+"showOrderDetail";
        public            static            final            String
SHOW_RIDER_LOCATION_AGAINST_LATLONG                    =
baseURL+"showRiderLocationAgainstOrder";

        public    static    final    String    SHOW_MENU_EXTRA_ITEM    =
baseURL+"showMenuExtraItems";
        public    static    final    String    SHOE_TOTAL_RATINGS    =
baseURL+"showRestaurantRatings";
        public    static    final    String    SHOW_RESTAURANT_DEALS    =
baseURL+"showRestaurantDeals";


        public static final String SHOW_REST_AGAINST_SPECIALITY
= baseURL+"showRestaurantsAgainstSpeciality";
        public static final String SHOW_REST_SPECIALITY_LIST =
baseURL+"showRestaurantsSpecialities";

        public    static    final    String    VERIFY COUPAN    =
baseURL+"verifyCoupon";
        public    static    final    String    PLACE ORDER    =
baseURL+"placeOrder";
```

```
    public      static      final      String      ORDER_DEAL      =
baseURL+"orderDeal";



    /// All Admin Apis

    public    static    final    String    ACCEPT_DECLINE_STATUS    =
baseURL+"updateRestaurantOrderStatus";



    // Reviews API

    public    static    final    String    AddRestaurantRating    =
baseURL+"addRestaurantRating";
    public    static    final    String    GiveRatingsToRider    =
baseURL+"giveRatingsToRider";



    // Api For GetPlaces

    public     static     final     String     GET_CITY_BOUNDRIES     =
"http://maps.google.com/maps/api/geocode/json?address=";




    public static final String TOPIC_GLOBAL = "global";

    // broadcast receiver intent filters
    public    static    final    String    REGISTRATION_COMPLETE    =
"registrationComplete";
    public     static     final     String     PUSH_NOTIFICATION     =
"pushNotification";

    public static final int NOTIFICATION_ID = 100;
```
/////////////////////////////////////

RegisterActivity
```
 @SuppressLint("RestrictedApi")
   @Nullable
   @Override
   public    View    onCreateView(@NonNull  LayoutInflater  inflater,  @Nullable
ViewGroup container, @Nullable Bundle savedInstanceState) {
     v = inflater.inflate(R.layout.activity_sing_up, container, false);
     context=getContext();

     progressBar = v.findViewById(R.id.signUpProgress);
     progressBar.start();
```

```java
progressDialog = v.findViewById(R.id.progressDialog);
transparent_layer = v.findViewById(R.id.transparent_layer);

preferences            =            getContext().getSharedPreferences(PreferenceClass.user,
Context.MODE_PRIVATE);



// Configure Google Sign In
GoogleSignInOptions              gso              =              new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
        .requestIdToken(getString(R.string.google_gmail_release))
        .requestEmail()
        .build();
mGoogleSignInClient = GoogleSignIn.getClient(getContext(), gso);

// End//


callbackManager = CallbackManager.Factory.create();

/// End


spinner = (Spinner) v.findViewById(R.id.spinner);



back_sign_up = (ImageView) v.findViewById(R.id.back_icon);
countryCode = (TextView) v.findViewById(R.id.country_code);
phone = (MaskedEditText) v.findViewById(R.id.country_phone);
verification_div = v.findViewById(R.id.verification_screen);
main_sign_up = v.findViewById(R.id.main_sign_up);
btn_signup = v.findViewById(R.id.btn_signup);
btn_resend = v.findViewById(R.id.resend_btn);



// ed_progress = (LinearLayout) v.findViewById(R.id.linlaHeaderProgress);
back_icon_verification = v.findViewById(R.id.back_icon_verification);
verification_main_screen = v.findViewById(R.id.verification_main_div);

confirmation_main_screen = v.findViewById(R.id.confirmation_main_screen);

div = v.findViewById(R.id.div);
div.setOnTouchListener(new View.OnTouchListener() {
```

```java
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
      return true;
    }
  }
});

div2 = v.findViewById(R.id.div2);
div2.setOnTouchListener(new View.OnTouchListener() {
  @Override
  public boolean onTouch(View view, MotionEvent motionEvent) {
    return true;
  }
});

// EditText Boxes

editText1 = (EditText) v.findViewById(R.id.edit_text1);
editText2 = (EditText) v.findViewById(R.id.edit_text2);
editText3 = (EditText) v.findViewById(R.id.edit_text3);
editText4 = (EditText) v.findViewById(R.id.edit_text4);

e_first = (EditText) v.findViewById(R.id.ed_fname);
e_last = (EditText) v.findViewById(R.id.ed_lname);
e_email = (EditText) v.findViewById(R.id.ed_email);
e_password = (EditText) v.findViewById(R.id.ed_password);

fb_div = v.findViewById(R.id.fb_div);
fb_btn = (TextView) v.findViewById(R.id.btn_fb);
login_button_fb = (LoginButton) v.findViewById(R.id.login_button_fb);
login_button_fb.setReadPermissions(Arrays.asList("email"));

// If using in a fragment
login_button_fb.setFragment(this);

fb_btn.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {

    login_button_fb.performClick();
    /// Login Method
    FB_LOGIN();

    ////
  }
});
fb_div.setOnClickListener(new View.OnClickListener() {
  @Override
```

```
        public void onClick(View v) {
            login_button_fb.performClick();
            /// Login Method
            FB_LOGIN();
        }
    });

    google_sign_in_div = v.findViewById(R.id.google_sign_in_div);
    google_sign_in_div.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            GoogleSignInAccount                acct                =
GoogleSignIn.getLastSignedInAccount(getContext());
            if (acct != null) {
                String Fname = acct.getGivenName();
                String Lname = acct.getFamilyName();
                String Email = acct.getEmail();

                e_first.setText(Fname);
                e_last.setText(Lname);
                e_email.setText(Email);

            }
            else {
                Intent signInIntent = mGoogleSignInClient.getSignInIntent();
                startActivityForResult(signInIntent, 123);
            }
        }
    });


    main_sign_up_div = v.findViewById(R.id.main_sign_up_div);
```

>>
        Login procedures

```
        public static final Pattern EMAIL_ADDRESS_PATTERN =

        Pattern.compile(

            "[a-zA-Z0-9\\+\\.\\_\\%\\-\\+]{1,256}" +

                "\\@" +
```

```java
            "[a-zA-Z0-9][a-zA-Z0-9\\-]{0,64}" +

            "(" +

            "\\." +

            "[a-zA-Z0-9][a-zA-Z0-9\\-]{0,25}" +

            ")+"
    );


    CallbackManager callbackManager;
    public static GoogleSignInClient  mGoogleSignInClient;


    View view;
    Context context;


    @SuppressWarnings("deprecation")
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
ViewGroup container, @Nullable Bundle savedInstanceState) {


        view = inflater.inflate(R.layout.login_activity, container, false);
        context=getContext();
```

```java
sPref =
getContext().getSharedPreferences(PreferenceClass.user,Context.MODE_P
RIVATE);



ed_email  = (EditText)view.findViewById(R.id.ed_email);

ed_password =(EditText)view.findViewById(R.id.ed_password);

log_in_now  = (Button)view.findViewById(R.id.btn_login);




GoogleSignInOptions  gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
        .requestEmail()
        .build();

mGoogleSignInClient = GoogleSignIn.getClient(getContext(),  gso);



FacebookSdk.sdkInitialize(getContext());

AppEventsLogger.activateApp(getContext());



callbackManager  = CallbackManager.Factory.create();
```

```
/// End


tv_sign_up  = view.findViewById(R.id.tv_sign_up);

tv_signed_up_now  = view.findViewById(R.id.tv_signed_up_now);

FontHelper.applyFont(getContext(),tv_sign_up,  AllConstants.verdana);


fb_btn = view.findViewById(R.id.fb_btn);

fb_div = view.findViewById(R.id.fb_div);

fb_div.setOnClickListener(this);

fb_btn.setOnClickListener(this);

login_button_fb  = (LoginButton)

view.findViewById(R.id.login_button_fb);

login_button_fb.setReadPermissions(Arrays.asList("email"));


// If using in a fragment

login_button_fb.setFragment(this);

// Callback registration


back_icon = view.findViewById(R.id.back_icon);

back_icon.setOnClickListener(new View.OnClickListener() {

  @Override

  public void onClick(View view) {


    LoginManager.getInstance().logOut();
```

```java
        try {

            InputMethodManager imm =

(InputMethodManager)getContext().getSystemService(INPUT_METHOD_

SERVICE);


imm.hideSoftInputFromWindow(getActivity().getCurrentFocus().getWindo

wToken(), 0);

        } catch (Exception e) {


        }


         getActivity().onBackPressed();

        LoginManager.getInstance().logOut();


    }
    });



    login_button_fb.registerCallback(callbackManager, new

FacebookCallback<LoginResult>() {
        @Override
        public void onSuccess(LoginResult loginResult) {
```

```java
        final AccessToken accessToken = loginResult.getAccessToken();

        GraphRequest request =

GraphRequest.newMeRequest(accessToken, new

GraphRequest.GraphJSONObjectCallback() {

        @Override

        public void onCompleted(JSONObject user, GraphResponse

graphResponse) {

            String useremail = user.optString("email");



            ed_email.setText(useremail);



        }

    });

    Bundle parameters = new Bundle();

    parameters.putString("fields", "last_name,first_name,email");

    request.setParameters(parameters);

    request.executeAsync();



}
```

```java
        @Override

        public void onCancel() {

            // App code


Toast.makeText(getContext(),"Cancle",Toast.LENGTH_SHORT).show();

        }



        @Override

        public void onError(FacebookException exception) {

            // App code


Toast.makeText(getContext(),"Error",Toast.LENGTH_SHORT).show();

        }

    });



    login_main_div = view.findViewById(R.id.login_main_div);

    login_main_div.setOnTouchListener(new View.OnTouchListener() {

        @Override

        public boolean onTouch(View view, MotionEvent motionEvent) {

            InputMethodManager imm = (InputMethodManager)

getContext().getSystemService(

                Activity.INPUT_METHOD_SERVICE);
```

```java
            imm.toggleSoftInput(InputMethodManager.HIDE_IMPLICIT_ONLY, 0);


               return false;

            }

         });


      google_sign_in_div = view.findViewById(R.id.google_sign_in_div);

      google_sign_in_div.setOnClickListener(new View.OnClickListener() {

         @Override

         public void onClick(View v) {

          GoogleSignInAccount acct =

GoogleSignIn.getLastSignedInAccount(getContext());

            if (acct != null) {

               String Email = acct.getEmail();


               ed_email.setText(Email);



            }

            else {

               Intent signInIntent = mGoogleSignInClient.getSignInIntent();

               startActivityForResult(signInIntent, 123);

            }
```

```java
            }
        });



        tv_email = (TextView)view.findViewById(R.id.tv_email);

        tv_pass = (TextView)view.findViewById(R.id.tv_password);

        sign_up_txt = (TextView)view.findViewById(R.id.tv_sign_up);


        logInProgress = view.findViewById(R.id.logInProgress);

        logInProgress.start();

        progressDialog = view.findViewById(R.id.progressDialog);

        transparent_layer = view.findViewById(R.id.transparent_layer);


        loginText = (TextView)view.findViewById(R.id.login_title);

        tv_forget_password = view.findViewById(R.id.tv_forget_password);

        tv_forget_password.setOnClickListener(this);

        FontHelper.applyFont(getContext(),tv_forget_password,

AllConstants.arial);


        //


        log_in_now.setOnClickListener(new View.OnClickListener() {
            @Override
```

```java
        public void onClick(View view) {

            boolean valid = checkEmail(ed_email.getText().toString());

            if (ed_email.getText().toString().trim().equals("")) {

                Toast.makeText(getContext(), "Enter Email!",
Toast.LENGTH_SHORT).show();

            } else if (ed_password.getText().toString().trim().equals("")) {

                Toast.makeText(getContext(), "Enter Password!",
Toast.LENGTH_SHORT).show();
            }else if (ed_password.getText().toString().length()<6) {

                Toast.makeText(getContext(), "Enter Password Atleat 6
Charaters!", Toast.LENGTH_SHORT).show();
            }
            else if (!valid) {

                Toast.makeText(getContext(), "Enter Valid Email!",
Toast.LENGTH_SHORT).show();
            }else {
```

```java
                            String this_email = ed_email.getText().toString();

                            String this_password = ed_password.getText().toString();

                            AddressListFragment.CART_NOT_LOAD = true;

                            login(this_email,this_password);



                    }

                }

            });
```

>>
Forget Password this method using to validate the password

```java
ViewFlipper viewFlipper;
    EditText recover_email,code_edit, ed_new_pass,ed_confirm_pass;
    SharedPreferences sharedPreferences;
    CamomileSpinner progressBar;
    RelativeLayout transparent_layer, progressDialog;

    @Nullable
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_recover_password);

        sharedPreferences =
getSharedPreferences(PreferenceClass.user,Context.MODE_PRIVATE);

        FrameLayout frameLayout = findViewById(R.id.main_recover_pass);
        FontHelper.applyFont(RecoverPasswordActivity.this,frameLayout,
AllConstants.verdana);
        initUI();

    }

    public void initUI(){

        progressBar = findViewById(R.id.signUpProgress);
        progressBar.start();
        progressDialog = findViewById(R.id.progressDialog);
        transparent_layer =findViewById(R.id.transparent_layer);
```

```java
        viewFlipper=findViewById(R.id.viewflliper);

        recover_email  = findViewById(R.id.recover_email);
        findViewById(R.id.btn_recover).setOnClickListener(this::onClick);


        code_edit=findViewById(R.id.code_edit);
        findViewById(R.id.sendcode_btn).setOnClickListener(this::onClick);


        ed_new_pass=findViewById(R.id.ed_new_pass);
        ed_confirm_pass=findViewById(R.id.ed_confirm_pass);
        findViewById(R.id.btn_change_pass).setOnClickListener(this::onClick);


        findViewById(R.id.Goback1).setOnClickListener(this::onClick);
        findViewById(R.id.Goback2).setOnClickListener(this::onClick);
        findViewById(R.id.Goback3).setOnClickListener(this::onClick);

}

public void CallApi_sendemail(){

    JSONObject jsonObject = new JSONObject();
    try {
        jsonObject.put("email",recover_email.getText().toString());

    } catch (JSONException e) {
        e.printStackTrace();
    }


    progressDialog.setVisibility(View.VISIBLE);
    transparent_layer.setVisibility(View.VISIBLE);

    ApiRequest.Call_Api(this, Config.forgotPassword, jsonObject, new Callback() {
        @Override
        public void Responce(String resp) {

            progressDialog.setVisibility(View.GONE);
            transparent_layer.setVisibility(View.GONE);

            try {
                JSONObject jsonResponse = new JSONObject(resp);
```

```java
            int code_id = Integer.parseInt(jsonResponse.optString("code"));
            if (code_id==200){

                Toast.makeText(RecoverPasswordActivity.this,"Password sent to your
given email",Toast.LENGTH_LONG).show();
                viewFlipper.showNext();

            }
            else {
                Toast.makeText(RecoverPasswordActivity.this,"Your email is not
correct",Toast.LENGTH_LONG).show();
            }

        } catch (JSONException e) {
            e.printStackTrace();

        }

      }
    });


  }
```

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Create the menus

```java
  miles_desc_tv = v.findViewById(R.id.miles_desc_tv);
   searchView = v.findViewById(R.id.floating_search_view);
   String txt="<font color = #dddddd>" + "Search Restaurant Menu" +
"</font>";
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.N) {
        searchView.setQueryHint(Html.fromHtml(txt,
Html.FROM_HTML_MODE_LEGACY));
    } else {
        searchView.setQueryHint(Html.fromHtml(txt));
    }
   TextView searchText = (TextView)
        v.findViewById(R.id.search_src_text);
   searchText.setTextSize(TypedValue.COMPLEX_UNIT_SP,14);
   searchText.setPadding(0,0,0,0);
   LinearLayout searchEditFrame = (LinearLayout)
searchView.findViewById(R.id.search_edit_frame); // Get the Linear
Layout
```

```
   ((LinearLayout.LayoutParams)
searchEditFrame.getLayoutParams()).leftMargin = 5;
   search(searchView);

   upper_header = v.findViewById(R.id.upper_header);
   about_div = v.findViewById(R.id.about_div);

   about_icon = v.findViewById(R.id. about_icon);

   res_menu_item_progress =
v.findViewById(R.id.res_menu_item_progress);
   res_menu_item_progress.start();
   rating = v.findViewById(R.id.rating);
   rastaurant_menu_item_title_tv =
v.findViewById(R.id.rastaurant_menu_item_title_tv);
   restaurant_name_tv = v.findViewById(R.id.restaurant_name_tv);
   salogon_tv = v.findViewById(R.id.salogon_tv);
   restaurant_img = v.findViewById(R.id.restaurant_image);
   cover_image = v.findViewById(R.id.cover_image);


   rating.setRating(Float.parseFloat(item_model.restaurant_avgRating));

   rastaurant_menu_item_title_tv.setText(item_model.restaurant_name);
   restaurant_name_tv.setText(item_model.restaurant_name);
   salogon_tv.setText(item_model.restaurant_salgon);

   if(item_model.deliveryFee_Range.equalsIgnoreCase("0")){
      miles_desc_tv.setText(item_model.restaurant_currency + " " +
item_model.delivery_fee_per_km + " /km");
   }
     else {
      miles_desc_tv.setText(item_model.restaurant_currency + " " +
item_model.delivery_fee_per_km + " /km- free delivery over " +
item_model.restaurant_currency + " " + item_model.min_order_price + "
within " + item_model.deliveryFee_Range + " km");
   }
```

```java
      Uri uri =
Uri.parse(Config.imgBaseURL+item_model.restaurant_image);
      restaurant_img.setImageURI(uri);



    about_icon.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        new MaterialDialog.Builder(getContext())
              .title(item_model.restaurant_name)
              .content(item_model.restaurant_about)
              .positiveText("OK")
              .show();
      }
    });

    backIcon = v.findViewById(R.id.back_icon_menu_option);
    backIcon.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View  view) {

        getActivity().onBackPressed();

      }
    });

    about_restaurant_div =
(RelativeLayout)v.findViewById(R.id.about_restaurant_div);
    about_restaurant_div.setOnClickListener(new  View.OnClickListener()
{
      @Override
      public void onClick(View  view) {

        FLAG_RES_MENU_FRAG = true;
        RestaurantDealsFragment restaurantDealsFragment  = new
RestaurantDealsFragment();
        FragmentTransaction transaction =
getChildFragmentManager().beginTransaction();
        Bundle bundle=new Bundle();
        bundle.putSerializable("data",item_model);
```

```
            restaurantDealsFragment.setArguments(bundle);
            transaction.addToBackStack(null);
            transaction.replace(R.id.resaurant_menu_items_main_layout,
restaurantDealsFragment,"ParentFragment_MenuItems").commit();


         }
      });
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Create the cart
public void initUI(View view){
      sPref = getContext().getSharedPreferences(PreferenceClass.user,
Context.MODE_PRIVATE);
      udid = sPref.getString(PreferenceClass.UDID,"");
      getLoINSession = sPref.getBoolean(PreferenceClass.IS_LOGIN,false);
      user_id = sPref.getString(PreferenceClass.pre_user_id,"");



      free_delivery_tv = view.findViewById(R.id.free_delivery_tv);
      progressDialog = view.findViewById(R.id.progressDialog);
      transparent_layer = view.findViewById(R.id.transparent_layer);

      delivery_address_tv = view.findViewById(R.id.delivery_address_tv);
      cartProgress = view.findViewById(R.id.cartProgress);
      cartProgress.start();
      cart_main_container = view.findViewById(R.id.cart_main_container);
      cart_main_container.setOnTouchListener(new View.OnTouchListener() {
        @Override
        public boolean onTouch(View view, MotionEvent motionEvent) {
           return true;
        }
      });


      cart_main_container.invalidate();
      credit_card_number_tv = view.findViewById(R.id.credit_card_number_tv);


      extraItemArray = new ArrayList<>();



      firebaseDatabase = FirebaseDatabase.getInstance();
      mDatabase =
firebaseDatabase.getReference().child(AllConstants.CALCULATION).child(udid);
```

```
riderTip = "0";

no_cart_div = view.findViewById(R.id.no_cart_div);
mainCartDiv = view.findViewById(R.id.mainCartDiv);
promo_tv = view.findViewById(R.id.promo_tv);
total_promo_tv = view.findViewById(R.id.total_promo_tv);


cart_check_out_div = view.findViewById(R.id.cart_check_out_div);

clear_btn = view.findViewById(R.id.clear_btn);
rest_name_tv = view.findViewById(R.id.rest_name_tv);

discount_tv = view.findViewById(R.id.discount_tv);
promo_code_div = view.findViewById(R.id.promo_code_div);
rider_tip = view.findViewById(R.id.rider_tip);
total_sum_tv = view.findViewById(R.id.total_sum_tv);

total_delivery_fee_tv= view.findViewById(R.id.total_delivery_fee_tv);
rider_tip_price_tv = view.findViewById(R.id.rider_tip_price_tv);

tip_div = view.findViewById(R.id.tip_div);
total_tex_tv = view.findViewById(R.id.total_tex_tv);
tax_tv = view.findViewById(R.id.tax_tv);


sub_total_price_tv = view.findViewById(R.id.sub_total_price_tv);
decline_div = view.findViewById(R.id.decline_div);
accept_div = view.findViewById(R.id.accept_div);
decline_tv = view.findViewById(R.id.decline_tv);
accept_tv = view.findViewById(R.id.accept_tv);
selected_item_list = view.findViewById(R.id.selected_item_list);
```

5.3. Reports

Users Table

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
☐	1	id 🔑	int(11)			No	*None*		AUTO_INCREMENT	🖊 Change	⊝ Drop	▾ More
☐	2	email	varchar(255)	utf8_general_ci		No	*None*			🖊 Change	⊝ Drop	▾ More
☐	3	password	varchar(255)	utf8_general_ci		No	*None*			🖊 Change	⊝ Drop	▾ More
☐	4	salt	varchar(255)	utf8_general_ci		No	*None*			🖊 Change	⊝ Drop	▾ More
☐	5	active	int(11)			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	6	block	int(11)			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	7	created	datetime			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	8	token	varchar(255)	utf8_general_ci		Yes	*NULL*			🖊 Change	⊝ Drop	▾ More
☐	9	role	varchar(255)	utf8_general_ci		No	*None*	user,admin,hotel,rider		🖊 Change	⊝ Drop	▾ More

Restaurant Table

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
☐	1	id 🔑	int(11)			No	*None*		AUTO_INCREMENT	🖊 Change	⊝ Drop	▾ More
☐	2	name	varchar(255)	utf8_general_ci		No	*None*			🖊 Change	⊝ Drop	▾ More
☐	3	slogan	varchar(255)	utf8_general_ci		No	*None*			🖊 Change	⊝ Drop	▾ More
☐	4	about	varchar(500)	utf8_general_ci		No	*None*			🖊 Change	⊝ Drop	▾ More
☐	5	speciality	varchar(200)	utf8_general_ci		No	*None*			🖊 Change	⊝ Drop	▾ More
☐	6	phone	varchar(20)	utf8_general_ci		No	*None*			🖊 Change	⊝ Drop	▾ More
☐	7	timezone	varchar(100)	utf8_general_ci		No	*None*			🖊 Change	⊝ Drop	▾ More
☐	8	menu_style	int(11)			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	9	promoted	int(11)			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	10	image	varchar(255)	utf8_general_ci		No	*None*			🖊 Change	⊝ Drop	▾ More
☐	11	preparation_time	tinyint(3)			No	*None*	minutes		🖊 Change	⊝ Drop	▾ More
☐	12	min_order_price	decimal(10,2)			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	13	delivery_free_range	int(11)			No	*None*	in km		🖊 Change	⊝ Drop	▾ More
☐	14	currency_id	int(11)			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	15	tax_id	int(11)			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	16	tax_free	int(3)			No	*None*	1 - tax free		🖊 Change	⊝ Drop	▾ More

Server: localhost:3306 » Database: ofernhpp_foodzone » Table: restaurant

Browse	Structure	SQL	Search	Insert	Export	Import	Operations	Triggers

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
☐	19	user_id 🔑	int(11)			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	20	added_by	int(11)			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	21	created	datetime			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	22	updated	datetime			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	23	google_analytics	int(20)			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	24	block	int(11)			No	*None*			🖊 Change	⊝ Drop	▾ More
☐	25	single_restaurant	int(11)			No	*None*			🖊 Change	⊝ Drop	▾ More

Order Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id 🔑	int(11)			No	None		AUTO_INCREMENT	🖉 Change ⊘ Drop ▾ Mo
2	deal_id	int(11)			No	None			🖉 Change ⊘ Drop ▾ Mo
3	created	datetime			No	None			🖉 Change ⊘ Drop ▾ Mo
4	price	float(10,2)			No	None			🖉 Change ⊘ Drop ▾ Mo
5	status	int(11)			No	None	1- active , 2-completed ,3 accepted , 4 rejected		🖉 Change ⊘ Drop ▾ Mo
6	delivery_fee	float(10,2)			No	None			🖉 Change ⊘ Drop ▾ Mo
7	user_id 🔑	int(11)			No	None			🖉 Change ⊘ Drop ▾ Mo
8	address_id	int(11)			No	None			🖉 Change ⊘ Drop ▾ Mo
9	payment_method_id	int(11)			No	None			🖉 Change ⊘ Drop ▾ Mo
10	quantity	int(11)			No	None			🖉 Change ⊘ Drop ▾ Mo
11	delivery	int(11)			No	None	0 - pickup, 1 - delivery		🖉 Change ⊘ Drop ▾ Mo
12	rider_tip	int(11)			No	None			🖉 Change ⊘ Drop ▾ Mo
13	tax	float(10,2)			No	None			🖉 Change ⊘ Drop ▾ Mo
14	sub_total	float(10,2)			No	None			🖉 Change ⊘ Drop ▾ Mo
15	restaurant_id	int(11)			No	None			🖉 Change ⊘ Drop ▾ Mo
16	instructions	text	utf8_general_ci		No				🖉 Change ⊘ Drop ▾ Mo

Server: localhost:3306 » Database: ofernhpp_foodzone » Table: order

Browse Structure SQL Search Insert Export Import Operations Triggers

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
19	cod	int(2)			No	None			🖉 Change ⊘ Drop ▾ Mo
20	notification	int(11)			No	None			🖉 Change ⊘ Drop ▾ Mo
21	rejected_reason	text	utf8_general_ci		No				🖉 Change ⊘ Drop ▾ Mo
22	restaurant_delivery_fee	int(11)			No	None	restaurent will pay us this delivery fee		🖉 Change ⊘ Drop ▾ Mo
23	total_distance_between_user_and_restaurant	int(11)			No	None	total distance between user address and hotel address		🖉 Change ⊘ Drop ▾ Mo
24	delivery_fee_per_km	int(11)			No	None	resturent delivery fee per km		🖉 Change ⊘ Drop ▾ Mo
25	delivery_free_range	int(11)			No	None	resturent free delivery range		🖉 Change ⊘ Drop ▾ Mo
26	tracking	int(11)			No	None			🖉 Change ⊘ Drop ▾ Mo
27	stripe_charge	varchar(200)	utf8_general_ci		No	None			🖉 Change ⊘ Drop ▾ Mo
28	device	varchar(10)	utf8_general_ci		No	None			🖉 Change ⊘ Drop ▾ Mo
29	version	varchar(25)	utf8_general_ci		No	None			🖉 Change ⊘ Drop ▾ Mo

restaurant_menu_item Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id 🔑	int(11)			No	None		AUTO_INCREMENT	🖉 Change ⊘ Drop ▾ More
2	name	varchar(255)	utf8_general_ci		No	None			🖉 Change ⊘ Drop ▾ More
3	description	text	utf8_general_ci		No				🖉 Change ⊘ Drop ▾ More
4	price	float			No	None			🖉 Change ⊘ Drop ▾ More
5	image	varchar(255)	utf8_general_ci		No	None			🖉 Change ⊘ Drop ▾ More
6	created	datetime			No	None			🖉 Change ⊘ Drop ▾ More
7	active	int(11)			No	1			🖉 Change ⊘ Drop ▾ More
8	restaurant_menu_id	int(11)			No	None			🖉 Change ⊘ Drop ▾ More
9	out_of_order	int(11)			No	None			🖉 Change ⊘ Drop ▾ More

5. CONCLUSIONS

5.1 ADVANTAGES

- FZ makes works easy and faster than current applications.

- All the data will be saved in the database. So the administer can view all the data on time.

- This system reduces manual works.

- FZ will stores all the records permanently and also gives the raises and deviations in the status of the restaurant in the markets.

- All the records of the current and ex-staffs will be stored in the database.

- All the shift timings of staffs, salary, extra working time and charge etc. are also updated daily.

- Any cancelations of parties are also updated.

- All the expenses per day will updated in the database daily.

- Also keeps the record of food items prepared and   the sales of food and also the record of balance food.

- All the food order details like the order types (normal, home delivery, party order etc.) are stored daily.

LIMITATIONS:

- Application won't be able to send some notification about any report or any delivery system to owner or customer.

- Application will required the java, MySQL and three different system which should be connected to one network for using application to its full functionality.

- User won't able to change the skin or any functionality of application.

- User can only get the reports in some formats only.

- Delivery Options: Add delivery option

- Restaurant Locator: Allow to find and choose a nearby restaurant

- User can reserve table.

5.2   Future Work:

The following section describes the work that will be implemented with future releases of the software.

- Customize orders: Allow customers to customize food orders

- Payment Options: Add different payment options such as PayPal, Cash, Gift Cards etc. Allow to save payment details for future use.

- Allow to process an order as a Guest

- Order Process Estimate: Provide customer a visual graphical order status bar

- Order Status: Show only Active orders to Restaurant Employees.

- Order Ready notification: Send an Order Ready notification to the customer

6   References

1. "A Secular Shift to Online Food Ordering". TechCrunch. 2015-05-07.

2. Associated Press. "Papa John's hits online ordering milestone." 5 May 2008.

3. Soder, Chuck. "Online Ordering System Will Get Bigger Slice of Case Students' Pie." Crane's Cleveland Business News. 14 May 2007.

4. The retailer, EY's publication in consumer products and retail sector (Report). EY. January–March 2015.

5. "Why Pizza Giants Want Customers to Click, Not Call, for Delivery". Adage.com. Retrieved January 10, 2016.

6. "Delivery Start-Ups Are Back Like It's 1999". The New York Times. Retrieved January 10, 2016.

7. "Online food delivery ordering is about to overtake phone ordering in the US - Quartz". Qz.com. Retrieved January 10, 2016.

8. "Restaurant food delivery heating up". Columbian.com. Retrieved January 10, 2016.

9. "Peapod". Retrieved January 8, 2016.

10. "Pizza Hut Celebrates 20th Anniversary of World's First Online Purchase With 50 Percent Off Online Deal for Hut Lovers Members". Restaurantnews.com. Retrieved January 8, 2016.