

1. INTRODUCTION

Order It is a revolutionary app to ease the order taking during rush hours at restaurant. This particular product makes the mismatch of orders potentially zero and manpower required also gets nullified instantly. It will be pretty effective for restaurants of every scale as it minimizes the amount that has to spend on workers and unique experience makes the customers to feel the meal in elite manner.

With Order It, customers can use a self-ordering terminal at a table to access the restaurant menu and order a meal. This ensures prompt order processing at peak times, improves the overall service speed, and reduces waiting times to ultimately increase customer satisfaction. Managers or restaurant owners can use this platform to edit and dispatch interactive advertisements and to configure content delivery schedules for targeting specific customers at certain times. For example, the Order It can be configured to display discount information of mealtimes to attract customers.

Every time when customers comes to any hotel, they will be waiting for waiters or bearers and gets hesitated to wait to have food. In this Scenario, the customers are leaving the hotel or restaurants without getting any service from them. Every table present in a restaurant will be provided with a tablet preinstalled with e-menu from the app. whenever the customer orders some dish, the data will be received by the one who is using admin interface. After receiving order, the admin will share the order details with cook and clocks the time required for the dish to prepare to the customer. After the dish gets prepared the waiter will serve the dish to the ordered table. Once the user decides to opt for bill he will press the checkout button and submits the feedback as well. Then the admin gets notification so that he will print the bill receipt

Order It Tablet Menu does not only offer visually rich digital menu but we also often listen our customers, get their feedbacks and develop new features to improve our app. There are so many advantages of using tablet menus over paper menus. These advantages can immediately help you to improve customer satisfaction and increase your revenue. Order It advertising feature allows restaurant owners to host advertisements on Android to generate additional revenue.

Order It- Android Based Food Ordering System

The main purpose of the app is to deliver the food items to the customer without any overhead problems. This app will deliver in digital format that replaces the traditional format such as menu cards and the physical presentations. Keeping up with Trends a Order It is a great method of keeping your customers and staff engaged while making more sales

2. BACKGROUND STUDY

Every hotel management wants their customer satisfaction and wants to make their services better and attractive. Every time when customers comes to any hotel, they will be waiting for waiters or bearers and gets hesitated to wait to have food. In this Scenario, the customers are leaving the hotel or restaurants without getting any service from them. After seeing such situations, the idea of making digital menu came into existence. That is my project, Order It.

Here, the servers/waiters generally take the order from the customer and head onto a terminal, where they can feed the order into a computer. The order can then be transmitted to the kitchen automatically via the terminal through a network, or it may even be delivered manually by the server to the kitchen.

Existing System

Restaurant services such as making reservations, processing orders, and delivering meals generally require waiters to input customer information and then transmit the orders to kitchen for meal preparation. When the customer pays the bill, the amount due is calculated by the cashier. Although this procedure is simple, it may significantly increase the workload of waiters and even cause errors in meal ordering or in prioritizing customers, especially when the number of customers suddenly increases during busy hours, which can seriously degrade the overall service quality.

Limitations

- At rush times at hotel or restaurants, customers are not completely satisfied with the services and offerings provided by the hotel. In this case, hotel management lose their benefits or yield from their customers.
- Even reputation matters over there. Hotel management cannot assign extra manpower for those particular times.

Proposed Methods

A very commonly implemented system, currently being used by numerous restaurants and chains all over the world, is the electronic point-of-sale (POS) terminal system.

Every table present in a restaurant will be provided with a tablet preinstalled with e-menu from the app. whenever the customer orders some dish, the data will be received by the one who is using admin interface.

After receiving order, the admin will share the order details with cook and clocks the time required for the dish to prepare to the customer. After the dish gets prepared the waiter will serve the dish to the ordered table. Once the user decides to opt for bill he will press the checkout button and submits the feedback as well. Then the admin gets notification so that he will print the bill receipt.

Advantages

- Cost-effective solution by reducing reliance on manpower.
- Improves order accuracy and minimizes cost of error.
- Enticing and interactive menu display for convenience to the customers.
- Real-time data analysis to plan for your business enhancement.
- Investment appraisal with less cost.

3. PROBLEM IDENTIFICATION

3.1 PROBLEM DEFINITION

Although a huge improvement over the pen and paper still prevalent over the world, this does not have much value addition for the customer and mostly only benefits the establishment and the administration of the establishment. It may significantly increase the workload of waiters and even cause errors in meal ordering or in prioritizing customers, especially when the number of customers suddenly increases during busy hours, which can seriously degrade the overall service quality.

At rush times at hotel or restaurants, customers are not completely satisfied with the services and offerings provided by the hotel. In this case, hotel management lose their benefits or yield from their customers. Even reputation matters over there. Hotel management cannot assign extra man power for those particular times. To avoid this kind of problems, Order It has been developed to make food serving, digitized. Using this application reduces the man power and also give a great look for any hotel or restaurant. Even though there are many applications like e-menu, ezze etc. Making it as user-friendly is the main criteria over this application.

3.2 REQUIREMENTS

ANDROID APPLICATION

Development Tools

Tools & Editor	:	Android Studio
API Level	:	19
Server	:	XAMPP
Drawing tools	:	Adobe XD, edraw max

Development Environment

- Software Requirements**

Operating system	:	Windows 10
Languages	:	Java, XML, Json
Databases	:	MYSQL, SQLite

- Hardware Requirements**

Processor	:	Processor Intel(R) Core(TM) i3.
Ram	:	8 GB (Recommended)
Hard Disk	:	1 TB
Monitor	:	15.6'' Anti-glare screen
Input Devices	:	Keyboard, Mouse

WEB ADMIN PANEL

Development Tools

Tools & Editor : Sublime Text3
Server : XAMPP
Drawing tools : Adobe XD, edraw max

Development Environment

• Software Requirements

Operating system : Windows 10
Languages : HTML, CSS, Bootstrap, PHP
Databases : MYSQL

• Hardware Requirements

Processor : Processor Intel(R) Core(TM) i3.
Ram : 4 GB (Recommended)
Hard Disk : 1 TB
Monitor : 15.6'' Anti-glare screen
Input Devices : Keyboard, Mouse

4. DESIGN

4.1 SYSTEM ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can comprise system components, the expand systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs)

The below diagram explains about the System Application structure.

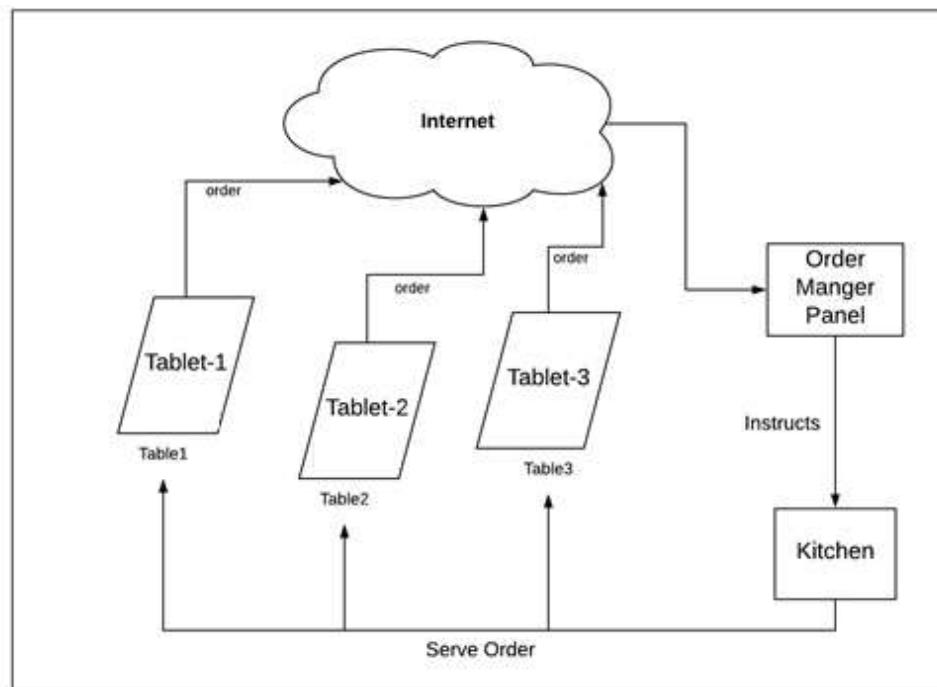


Figure 4.1: System Architecture

4.2 DATA DESIGN

Data design is the first design activity, which results in less complex, modular and efficient program structure. The information domain model developed during analysis phase is transformed into data structures needed for implementing the software. The data objects, attributes, and relationships depicted in entity relationship diagrams and the information stored in data dictionary provide a base for data design activity. During the data design process, data types are specified along with the integrity rules required for the data. For specifying and designing efficient data structures, some principles should be followed. These principles are listed below.

1. The data structures needed for implementing the software as well-as the operations that can be applied on them should be identified.
2. A data dictionary should be developed to depict how different data objects interact with each other and what constraints are to be imposed on the elements of data structure.
3. Stepwise refinement should be used in data design process and detailed design decisions should be made later in the process.
4. Only those modules that need to access data stored in a data structure directly should be aware of the representation of the data structure.
5. A library containing the set of useful data structures along with the operations that can be performed on them should be maintained.
6. Language used for developing the system should support abstract data types.

Order It- Android Based Food Ordering System

Database Model will depicts the all the requirements of the Data design. A database model is a type of data model that determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized and manipulated. The most popular example of a database model is the Relational model, which uses a table-based format.

The Relational model for my project will contains 5 tables namely:

- Users
- Tabs
- Food_Item
- Placed_order
- Placed_order_items

The Relational model for my project will be as follows:

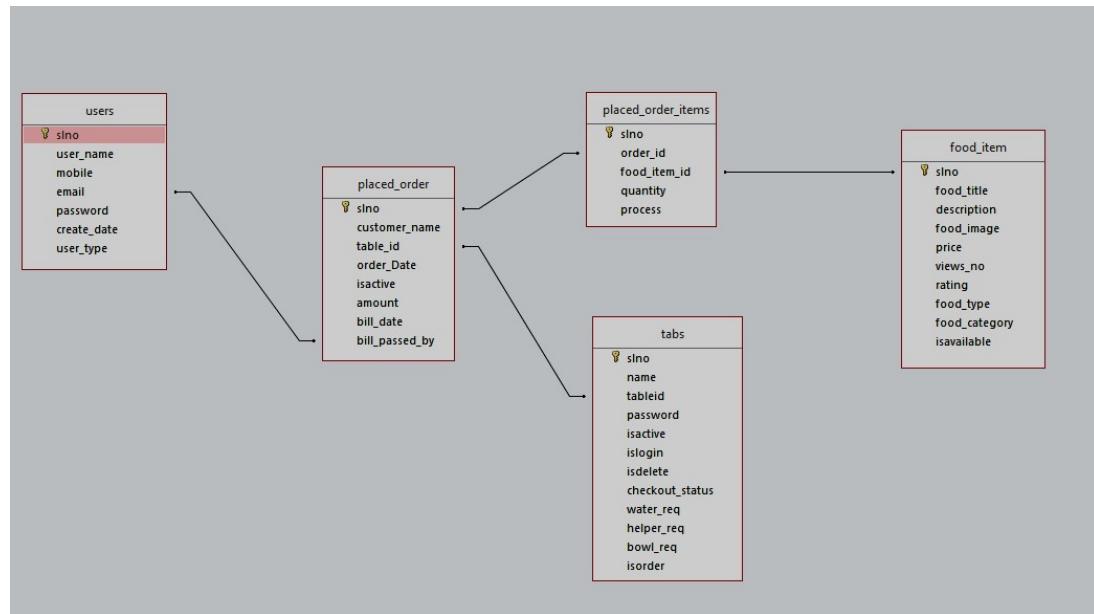


Figure 4.2: Relational model

4.3 COMPONENT DESIGN

The Component Design Activity is an activity of the Product Design Activity for creating a Component Design. The Product Architecture identifies a set of Adaptable Components that may be used to implement a work product family. A Component Design is a design specification for one of these Adaptable Components. Application engineers, using the Generation Procedure, may adapt and compose a set of these components to implement certain work products, or portions thereof. Each component must be designed to satisfy relevant aspects of the Product Requirements and all design structures of the Product Architecture.

The objective of the Component Design Activity is to produce a design for an Adaptable Component that satisfies applicable Product Requirements in accordance with its role in the Product Architecture.

4.4 INTERFACE DESIGN

User interface is the front-end application view to which user interacts in order to use the software. User can manipulate and control the software as well as hardware by means of user interface. Today, user interface is found at almost every place where digital technology exists, right from computers, mobile phones, cars, music players, airplanes, ships etc.

User interface is part of software and is designed such a way that it is expected to provide the user insight of the software. UI provides fundamental platform for human-computer interaction.

UI can be graphical, text-based, audio-video based, depending upon the underlying hardware and software combination. UI can be hardware or software or a combination of both.

The software becomes more popular if its user interface is:

- Attractive
- Simple to use
- Responsive in short time
- Clear to understand
- Consistent on all interfacing screens

My project has two user interfaces namely:

- Customer Interface
- Admin Interface

Customer Interface

Here User interface or Customer interface in an Android Application through which he can perform following actions:

- View The Food Items/offers
- Place an order multiple times
- Request for water, Finger bowl and Helper
- Checkout and give ratings to his orders

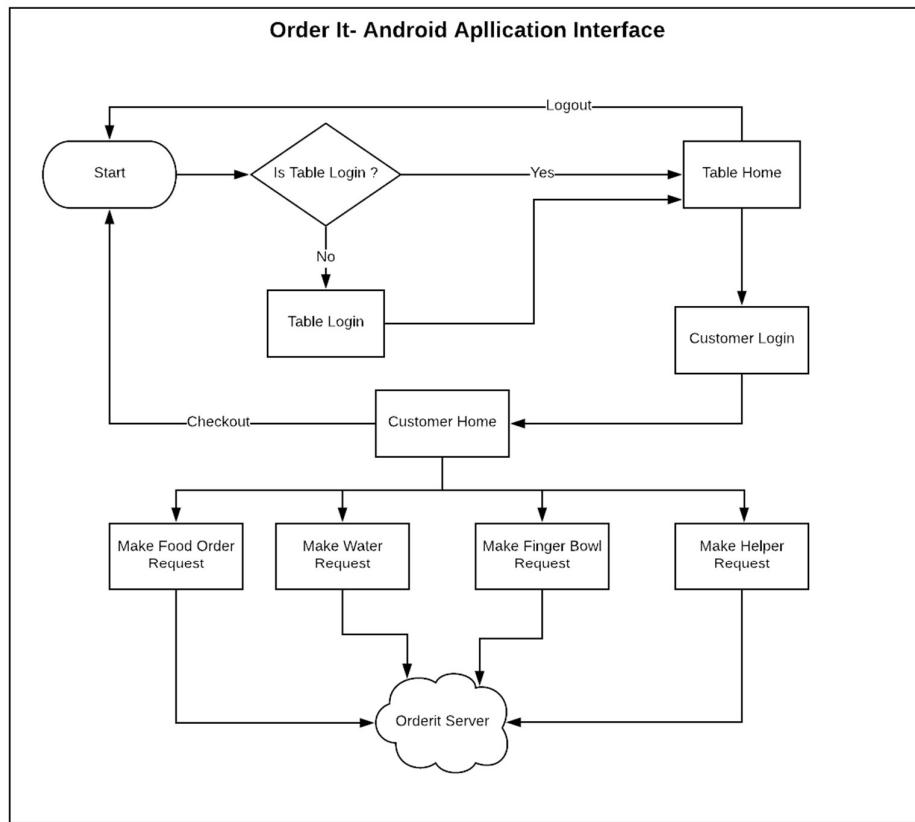


Figure 4.4.1: Customer Interface

Admin Interface

Here Admin interface or Admin Panel is a Real-time Web Application through which organization persons can perform following actions:

- Manage Customers Orders and Requests
- Managing Food Items/offers
- Add or Removing Others Users
- Add or Removing Others Tables
- Analyzing sales and Ratings

Order It- Android Based Food Ordering System

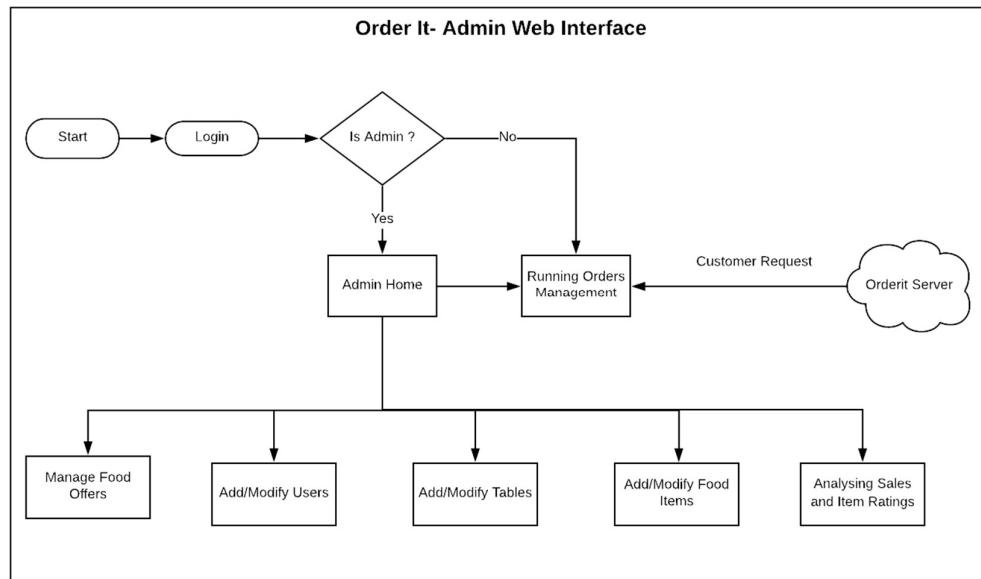


Figure 4.4.2: Admin Interface

4.5 SYSTEM SPECIFIC DESIGNS

UML Diagrams

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system. The current UML standards call for 13 different types of diagrams: class, activity, object, use case, sequence, package, state, component, communication, composite structure, interaction overview, timing, and deployment.

1. Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes

In my application **Order It** contains four classes namely Users, Order, Items and Admin and my class diagram will be as Follows.

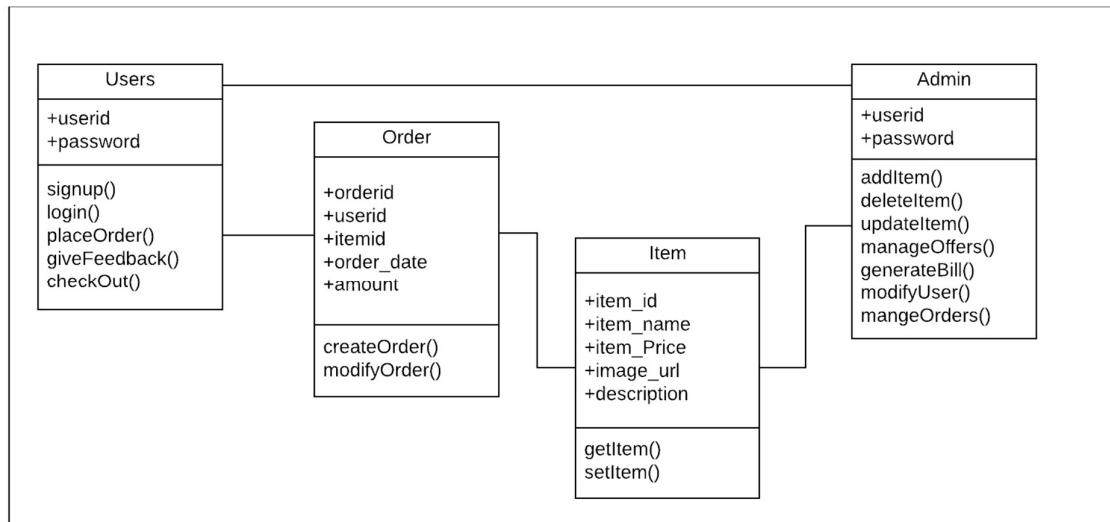


Figure 4.5.1: Class Diagram

2. Use case Diagram

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system. In my application **Order It** The Following Actors and Use cases.

Actors:

- Admin
- Users

Use cases:

- View offers
- Modify offers
- Place order
- Give feed back
- Checkout
- Generate bill
- View bill
- Analyses Feed back
- Logout

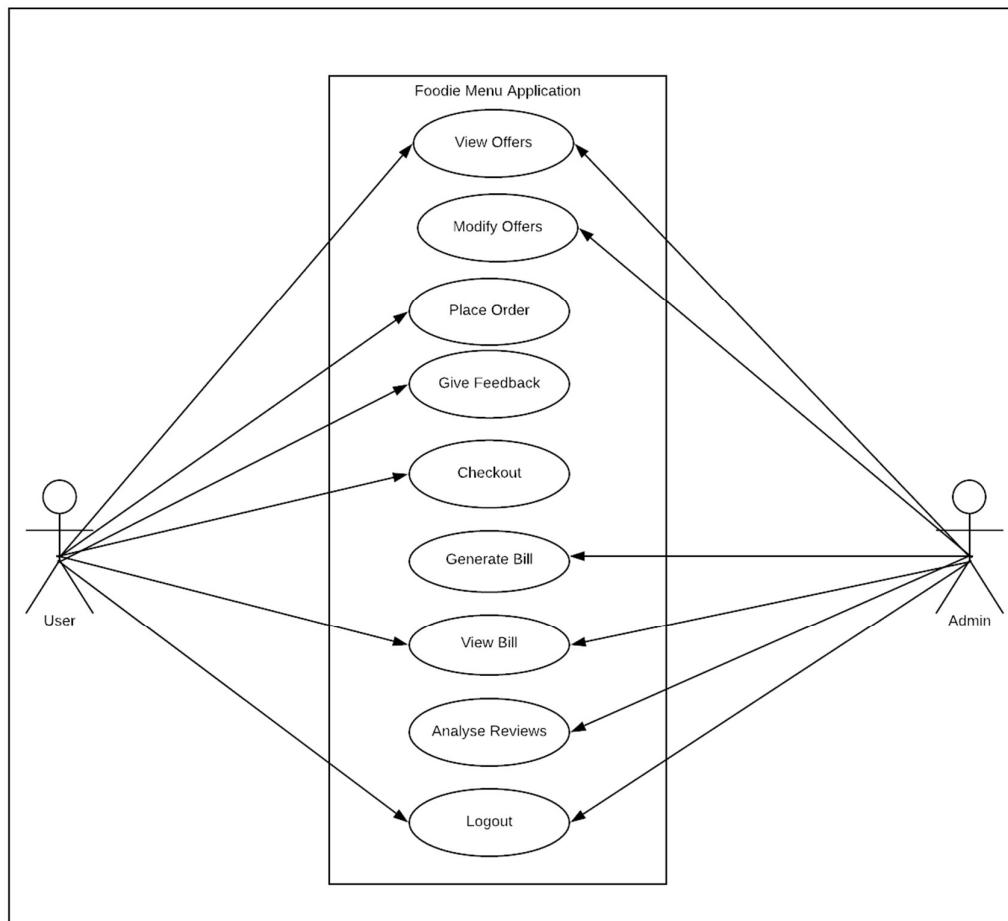


Figure 4.5.2: Use case Diagram

3. Sequence Diagram

In software engineering, a system sequence diagram (SSD) is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate, their order, and possible inter-system events. Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

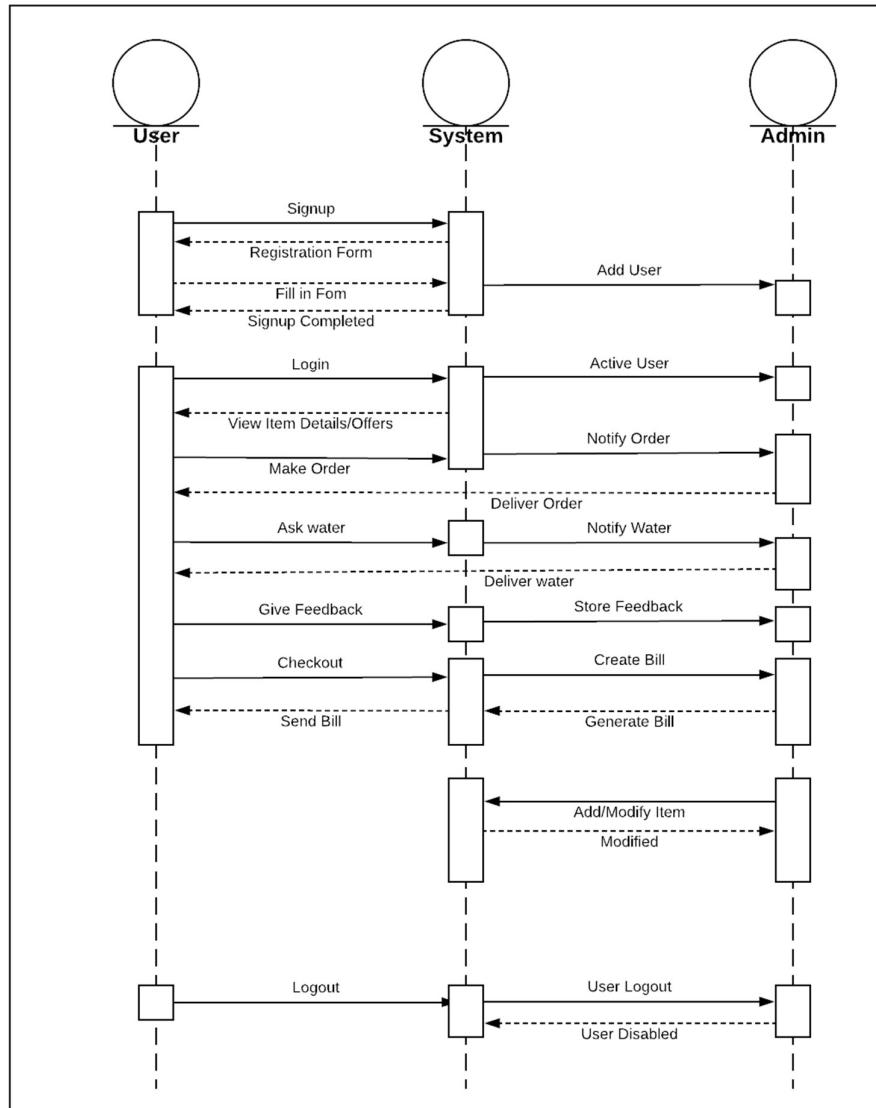


Figure 4.5.3: Sequence Diagram

4. Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

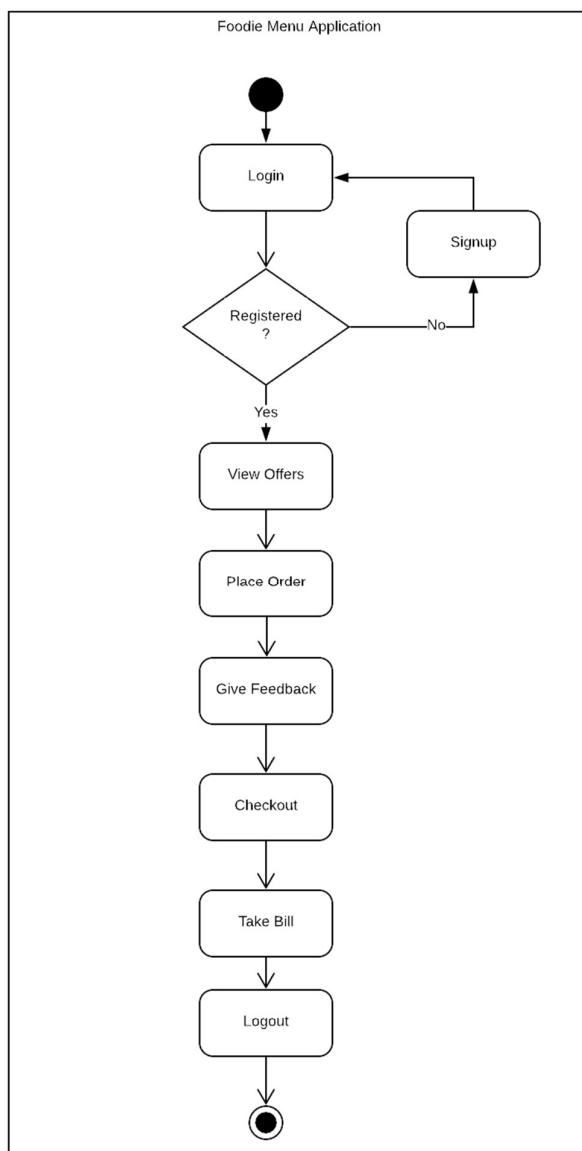


Figure 4.5.4: Activity Diagram

5. TESTING

5.1 INTRODUCTION OF TESTING

Testing is a process of checking whether the developed system is working according to the original objectives and requirements. It is a set of activities that can be planned in advance and conducted systematically. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate testing if not testing leads to errors that may not appear even many months. This creates two problems, the time lag between the cause and the appearance of the problem and the effect of the system errors on the files and records within the system. A small system error can conceivably explode into a much larger problem. Effective testing early in the process translates directly into long term cost savings from a reduced number of errors. Another reason for system testing is its utility, as a user-oriented vehicle before implementation. The best programs are worthless if it produces the correct outputs.

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Taking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

Strategic Approach to Software Testing

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirements analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decreases the level of abstraction on each item.

A Strategy for software testing may also be viewed in the context of the spiral. Testing will progress by moving outward along the spiral to integration testing, where the focus on the design and the concentration of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

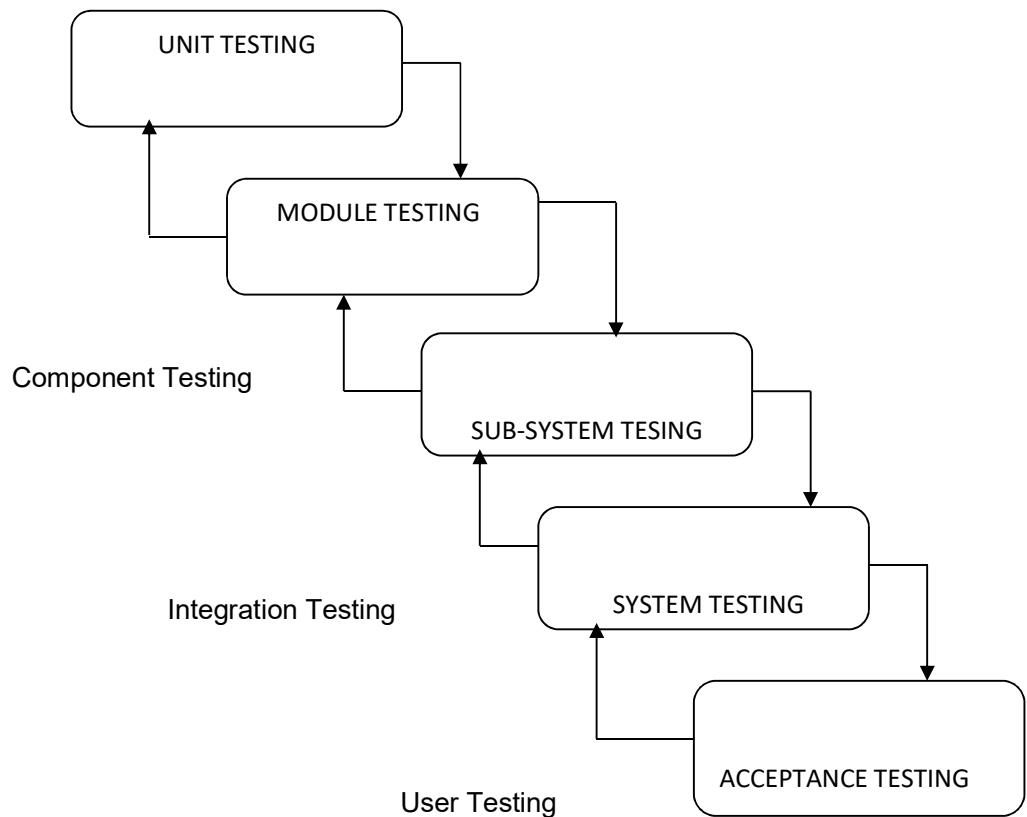


Figure 5.1.1: Software Testing

Test plans

A test plan is a document detailing the objectives, resources, and processes for a specific test for a software or hardware product. The plan typically contains a detailed understanding of the eventual workflow.

Developers use test plans that check a range of types of data under different circumstances. A test plan is a list of requirements designed to ensure that the coded solution works as expected. The test plan will include specific instructions about the data and conditions the program will be tested with.

Testing Strategies

A testing strategy is a general approach to the testing process rather than a method of devising particular system or component tests.

- Testing is a set of activities which are decided in advance i.e. before the start of development and organized systematically.
- In the literature of software engineering various testing strategies to implement the testing are defined.
- All the strategies give a testing template.

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs.

Unit testing for Android can be classified into:

- **Local unit tests** - tests which can run on the JVM. If possible, you should prefer to use local tests. Local tests run much faster compared to the time required to deploy and run the test on an Android device.
- **Instrumented unit tests** - tests which require the Android system. If you want to test code which uses the Android API, you need to run these tests on an Android device. Unfortunately, this makes the execution time of the tests longer.

Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

Black box testing

This method is used when knowledge of the specified function that a product has been designed to perform is known. The concept of black box is used to represent a system whose inside workings are not available to inspection. In a black box the test item is a "Black" , since its logic is unknown , all that is known is what goes in and what comes out , or the input and output.

Black box testing attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structure
- Performance errors
- Initialization and termination errors

As shown in the following figure of Black box testing, we are not thinking of the internal workings, just we think about

What is the output to our system?

What is the output for given input to our system?

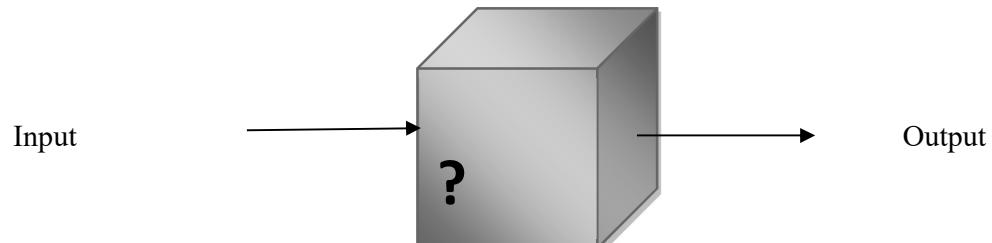


Figure 5.1.2: Black box Testing

The Black box is an imaginary box that hides its internal workings

White box testing

White box testing is concerned with testing the implementation of the program. The intent of structural is not to exercise all the inputs or outputs but to exercise the different programming and data structure used in the program. Thus structural testing aims to achieve test cases that will force the desire coverage of different structures. Two types of path testing are statement testing coverage and branch testing coverage.

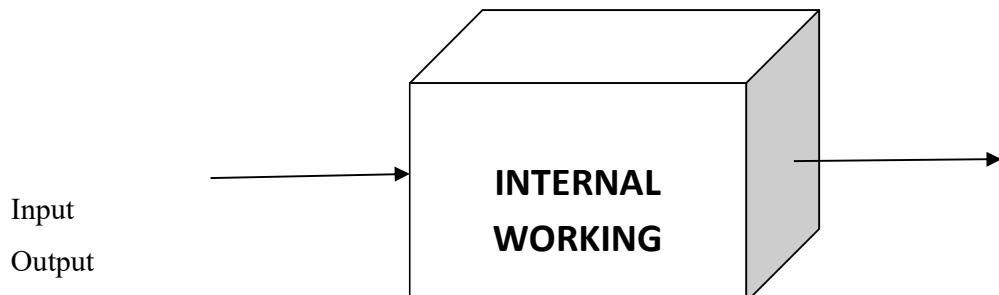


Figure 5.1.3: White box testing

The White Box testing strategy, the internal workings

5.2 TEST CASES

A test case is an individual set of variables or conditions that is used to see if features of the software work as expected. In the test plan, various test cases may be planned and then tested. When designing test cases, it is a good idea to design some test cases that will check if the software will fail and some that will check if it succeeds.

Order It: Android Application (User Interface)

Feature/ Module	Test cases/ Inputs	Expected Result	Actual Results	Results
Login	1.Table Id: tab1, Password: 4123	Successful	Successful	True
	2.Table Id: tab1, Password: 4123	Successful	Error 404	False
	3.Table Id: tab1, Password: 123	Invalid credentials	Invalid credentials	True
Loading Items	1.Categories: Main Courses, Item Type: Veg	Item List loaded	Item list Loaded	True
	2. Categories: Courses, Item Type: Non-veg	Item list not loaded	Item list not loaded	False
Generate Bill	1.Order id:25	Bill generated	Bill generated	True
	2. Order id: 25	Bill generated	Error 404	False
Check Out	1.Table Id:tab2, Order Id:25	Successful	Successful	True
	2.Table Id:tab2, Order Id:25	Successful	Error 404	False
Water Request/Helper Request/Finger bowl Request	1.needWater:true	Done	Done	True
	2.needWater:true	Done	Error 404	False

Table 5.2.1: TEST CASES (User Interface)

Order It: Web Admin panel (Admin Interface)

Feature/ Module	Test cases/ Inputs	Expected Result	Actual Results	Results
Login	1. User Id: jai@mine.com, Password: abc@123.O	Successful	Successful	True
	2. User Id: jai@mine.com, Password: abc@123.O	Successful	Error 404	
	3. User Id: jai@mine.com, Password: abc@	Invalid credentials	Invalid credentials	True
Add User	1. Name: Jai Mobile:9000204595 Email:jai@mine.com Password:Aa@12345 Re type Password: Aa@12345 User Type: Admin	Successful	Successful	True
	2. Name: Jai Mobile:900020495 Email:jai@mine.com Password:Aa@12345 Re type Password: Aa@12345 User Type: Admin	Invalid Mobile number	Invalid Mobile number	True
	3. Name: Jai Mobile:900020495 Email:jai Password:Aa@12345 Re type Password: Aa@12345 User Type: Admin	Invalid Mail id	Invalid Mail id	True
	4. Name: Jai Mobile:900020495 Email:jai	Password Must have At least one	Password Must have At least one	False

Order It- Android Based Food Ordering System

	Password:Aa@1234 Re type Password: Aa@1234 User Type: Admin	capital letter, small letter & one symbol	capital letter, small letter & one symbol	
Add Table	1. Table Name: Table1 Tab Id:tab Password:Aa@12345 Re type Password: Aa@12345	Successful	Successful	True
	2. Table Name: Table1 Tab Id:9000204595 Password:1234 Re type Password: 123	Password not match	Password not match	True
Add Food Offers	1. Food Title: Dosa Description: nice one Price:150 Type: Veg Category: Main course Pic: dosa.jpg	Added Successful	Added Successful	True
	2. Food Title: Dosa Description: nice one Price:150 Type: Veg Category: Main course Pic: dosa.mp4	Pic should be of jpg and png type only	Pic should be of jpg and png type only	False
Generate Bill	1.Table Id:tab2, Order Id:25	Bill generated	Bill generated	
	2.Table Id:tab2, Order Id:25	Bill generated	Error 404	False

Table 5.2.2: TEST CASES (Admin Interface)

5.3 TEST REPORTS

Order It: Android Application (User Interface)

Test case1 Login to the app	Priority (H,L) High	
Test Objective To accept the Login credentials such as username and password.		
Test Description User creates his details with registered credentials.		
Requirements verified Yes		
Test Environment Mobile Application		
Test setup/Pre-conditions Enter the valid username and password in formatted way.		
Actions	Expected	
The user details will be successfully registered and can login into the application.	Formatted values. Valid username and password.	
Pass Yes	Condition Pass	No Fail No
Problems / Issues NIL		
Notes Successfully Executed.		

Table 5.3.1: TEST REPORTS (User Interface)

Order It: Web Admin panel (Admin Interface)

Test case1 Registration& login to the app	Priority (H,L) High	
Test Objective To accept the Registered credentials such as username and password.		
Test Description User creates his details with registered credentials.		
Requirements verified Yes		
Test Environment Web Application		
Test setup/Pre-conditions Enter the valid username and password in formatted way.		
Actions	Expected	
The user details will be successfully registered and can login into the application.	Formatted values. Valid username and password.	
Pass Yes	Condition Pass No	Fail No
Problems / Issues NIL		
Notes Successfully Executed.		

Table 5.3.2: TEST REPORTS (Admin Interface)

5.4 TEST CONCLUSION

All the test cases mentioned above passed successfully. No defects encountered. After structural testing and functional testing we get error free modules. These modules are to be integrated to get the required results of the system. After checking a module, another module is tested and is integrated with the previous module. After the integration, the test cases are generated and the results are tested. Unit testing allows parallelism in the testing activities that is each component can be tested independently of one another. Hence the goal is to test the internal logic of the module.

6. IMPLEMENTATION

MODULES: Digital Menu-Order It contains following modules:

- **Customer Interface Modules**

1. Login and Signup
2. Home
3. Veg and Non-Veg
4. Offers
5. Contact
6. About-Us
7. Logout
8. Cart
9. Confirm-order

- **Admin Interface Modules**

1. Add an item
2. Update an item
3. Delete an item
4. Check the order and process it
5. Generate the bill
6. Analyses the user review
7. Manage offers

Modules Description

Customer Interface Module

The app consists of seven sections. Each section has its specific criteria. All users of the system, can Access these sections as follows

1. Login and Signup

Each Table will have one Tablet where Order It is installed and each must be sign up with unique User name and password for each table then we can login to Order It with these login details

2. Home

This section is the main page of the app after the user logins. This section delivers the main content of the app and the working of the app. This section will explain you detail about the app working from selecting the order to placing the order in a description way. In this section on the bottom it contains the demo of the app. It explains the way how the app works.

3. Veg and Non-Veg

In Veg section it will display all the categories of the veg items it contains such as soups, biryani items, beverages, veg-starters, main courses, staples etc. Non-veg section it will display all the categories of the non-veg items it contains such as soups, biryani items, main courses, starters, staples, restaurant specials, tandoor-se, from the Chinese etc.

For every item it contains an image, description, rating of the item and the price of the item. On clicking the item, it will display image, description, rating of the item and the price of the item. On the bottom of this page it contains the order button where you can checkout of the order of that specific item. It also contains the button such adding the items. After adding the items, it will automatically deliver to the cart section.

4. Offers

In this section it displays the offers that are available in the app of the items that are added by the admin. The admin has the ability to display the offer at any time. In this section it will display the offers with the images and the text of the offer whatever the admin fixes the offers of the specific categories and the items.

5. Contact

In this section it will deliver the support of the app, so that the customer can feel free to contact the super admin. It will display the logo of the company that product developed and also the address of the company.

6. About-Us

This section will display the description such as the features of the app and what are the advantages of the app, usage of the app, uniqueness of the other apps that are existed.

7. Logout

This section will have logged of the session of the user. It will display the alert box such as are you sure to logout the app or not.

8. Cart

In this section after selecting all the items there will be cart button for every section except the home page section. After selecting the items by the customer, he can directly goes to the car page. In the cart page it will display all the items that are ordered by the customers. It will show the total amount of the order, so that you can proceed out to the confirm order.

9. Confirm-order

In this section it will displays the table id, Order number, And the items that are ordered by the customer. At the bottom it contains the feedback rating and then you can check-out the order.

All of the above modules can be functioned as follows

- A User can check all the available items in the Order It.
- In the Order It we have two classifiers veg and non-veg. A User can select either of the classifiers.
- If the User selects veg, a list of veg categories are displayed to the user.
- If the User selects non-veg, a list of non-veg categories are displayed to the user.
- A User can choose the respective item that he wish to eat from the set of categories.
- A User can also check the available offers, prices and discounts of a particular item.
- A User can add the items to the cart. This makes the User easier to check his order list.
- A User can also update the quantity of the item that has been added to the cart.
- A User can also give feedback about the particular item by clicking on the emoji's given below.

Admin Interface Modules

The manager of the hotel is the admin of the app. He has the authorities to modify the menu specific items in the app that he intends to display in the app.

1. Add an item

When the hotel plans to add a new dish to its menu, the admin will have an interface to add the new item (a dish or a beverage or anything to add in the menu). He will also define all the details of the new item.

2. Update an item

When there is a modification needed in the currently available items in the menu, the admin will look into that and perform necessary changes in the interface.

3. Delete an item

If an item shown in the menu is not available in the hotel at current time or if there are items that are seasonal then the admin can either fade the respective item in the menu or delete from the list.

4. Check the order and process it

Here if any order received by the table then they can process for the next step for the kitchen and after completion of the recipe then food will deliver to the specific user of the table.

5. Generate the bill

Hotel accountant can generate the bill for specific user as per their table and food which ordered by him.

6. Analyses the user review

After submit the order the app will request for the review and the user and place review for their food. This help to analyses the demand of food from the hotel.

7. Manage offers

The admin can add, update or delete offers on all or specific items from the list whenever required. There are times when hotels offer discounts to the customers and they also offer their valuable customers with complementary dishes during festive seasons. These all are the actions to be performed on admin's part.

All the above actions need proper authority and an admin, who is the in-charge of the hotel, is in an appropriate position to handle these. So the admin is also the in-charge of the app and decides on what is to be displayed on the app for his/her hotel.

7. FUTURE ENHANCEMENT

- Currently, this application is working with the functionality like replacing menu cards, can be extended with online payment gateway for the ordered food.
- This app can be used for food sales report if the restaurant have many branches.
- It can make the order environment completely digitalized. And this application has a flexibility to change as per the client requirements as it is private app.
- Rating analysis can also be enhanced in this application using statistical algorithms to make restaurant management to easily analyze their business.
- The attractive feature of this application is, a user can know the offers of the day in restaurants. In future, we are planning to make this application offline.
- It's an application developed with functionality alone, this can be added with animations and some other effects to make it more attractive.

8. CONCLUSION

This application is completely developed for restaurants to develop their business. 100% digitized process for ordering food. Efficient estimation of food order based on budget. Instead of waiter or server this Order It will help you. Order It integrates with your POS system, sending orders straight from customer to kitchen and creating a seamlessly integrated platform for both the user and you. Order It Tablet Menu does not only offer visually rich digital menu but we also often listen our customers, get their feedbacks and develop new features to improve Hotels.

9. APPENDICES

9.1 APPENDIX- I: SOURCE CODE

Android Application (User Interface):

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.jai.orderit">
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:name=".utils.AppController"
        android:allowBackup="true"
        android:icon="@drawable/ot_round"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".SplashActivity"
            android:configChanges="orientation|keyboardHidden">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".LoginActivity" />
        <activity
            android:name=".HomeActivity"
```

Order It- Android Based Food Ordering System

```
        android:label="@string/title_activity_home"
        android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".CustomerActivity"
    android:label="@string/title_activity_customer"
    android:theme="@style/AppTheme.NoActionBar" />
<activity android:name=".CartActivity" />
<activity android:name=".ViewOrdersActivity" />
<activity android:name=".CheckOutActivity" />
<activity android:name=".ItemActivity"></activity>
</application>
</manifest>

activity_splash.xml:
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:id="@+id/mybackground"
    android:gravity="center"
    android:background="@drawable/sample"
    tools:context="com.example.jai.orderit.SplashActivity">

<ImageView
    android:scaleType="fitXY"
    android:layout_centerInParent="true"
    android:src="@drawable/logo"
    android:layout_width="275dp"
    android:layout_height="130dp" />

</LinearLayout>
SplashActivity.java
package com.example.jai.orderit;

import android.content.Intent;
import android.content.res.Configuration;
```

Order It- Android Based Food Ordering System

```
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.LinearLayout;

import com.example.jai.orderit.utils.SessionManager;

public class SplashActivity extends AppCompatActivity {
    private static int SPLASH_TIME_OUT = 3000;
    final int sdk = android.os.Build.VERSION.SDK_INT;
    SessionManager sessionManager;
    LinearLayout layout;
    Intent i;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        sessionManager=new SessionManager(getApplicationContext());
        layout=findViewById(R.id.mybackground);

        new Handler().postDelayed(new Runnable() {

            /*
             * Showing splash screen with a timer. This will be useful when you
             * want to show case your app logo / company
             */
            @Override
            public void run() {
                // This method will be executed once the timer is over
                // Start your app main activity
                if(sessionManager.isLoggedIn()){
                    if(sessionManager.isCustomerLoggedIn()){
                        i = new Intent(SplashActivity.this, CustomerActivity.class);
                    }
                    else {
                        i = new Intent(SplashActivity.this, HomeActivity.class);
                    }
                }
                else
            }
        }, SPLASH_TIME_OUT);
    }
}
```

Order It- Android Based Food Ordering System

```
{  
    i = new Intent(SplashActivity.this, LoginActivity.class);  
  
    }  
    startActivity(i);  
    finish();  
    // close this activity  
    finish();  
}  
, SPLASH_TIME_OUT);  
  
}  
  
@Override  
public void onConfigurationChanged(Configuration newConfig) {  
    super.onConfigurationChanged(newConfig);  
    // Checks the orientation of the screen  
    if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {  
  
        Log.d("Land", "ORIENTATION_LANDSCAPE");  
  
    } else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {  
  
        Log.d("Port", "ORIENTATION_PORTRAIT");  
    }  
}
```

activity_login.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<ScrollView 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"  
    tools:context="com.example.jai.orderit.LoginActivity">  
  
<RelativeLayout  
    android:layout_width="match_parent"
```

```
    android:layout_height="500dp"
    android:layout_margin="10dp"
    android:gravity="center">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <android.support.design.widget.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            app:hintTextAppearance="@style/CustomTextAppearance">

            <EditText
                android:id="@+id/tab_id"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Table ID"
                android:inputType="textEmailAddress"
                android:textSize="18sp" />
        </android.support.design.widget.TextInputLayout>

        <android.support.design.widget.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            app:hintTextAppearance="@style/CustomTextAppearance">

            <EditText
                android:id="@+id/tab_pass"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Password"
                android:inputType="textPassword"
                android:textSize="18sp" />
        </android.support.design.widget.TextInputLayout>

        <Button
            android:id="@+id/tab_login"
```

```
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginBottom="5dp"
        android:layout_marginLeft="25dp"
        android:layout_marginRight="25dp"
        android:layout_marginTop="10dp"
        android:background="@drawable/round_back"
        android:text="Login"
        android:textColor="#FFF" />
    </LinearLayout>
```

```
    </RelativeLayout>
</ScrollView>
```

LoginActivity.java:

```
package com.example.jai.orderit;

import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Typeface;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.VolleyLog;
import com.android.volley.toolbox.StringRequest;
import com.example.jai.orderit.utils.AppController;
import com.example.jai.orderit.utils.SessionManager;
import com.example.jai.orderit.utilsUrls;
import com.example.jai.orderit.utils.Utils;

import org.json.JSONException;
import org.json.JSONObject;
```

```
import java.util.HashMap;
import java.util.Map;

public class LoginActivity extends AppCompatActivity {
    EditText tab_id, tab_pass;
    Button tab_login;
    private ProgressDialog pDialog;
    SessionManager sessionManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        tab_id = findViewById(R.id.tab_id);
        tab_pass = findViewById(R.id.tab_pass);
        tab_login = findViewById(R.id.tab_login);
        pDialog = new ProgressDialog(this);
        sessionManager = new SessionManager(LoginActivity.this);
        Utils.hideKeyBoard(LoginActivity.this);

        tab_pass.setTypeface(Typeface.DEFAULT);
        pDialog.setMessage("Loading... ");
        pDialog.setCancelable(false);

        tab_login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showpDialog();
                loginTable();
            }
        });
    }

    private void showpDialog() {
        if (!pDialog.isShowing())
            pDialog.show();
    }
}
```

```
private void hidepDialog() {
    if (pDialog.isShowing())
        pDialog.dismiss();
}

private void loginTable() {

    StringRequest stringRequest = new StringRequest(
        Request.Method.POST,
       Urls.requestAPi,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Log.d("res", response.toString());
                hidepDialog();
                try {
                    JSONObject jsonresponse = new JSONObject(response);
                    String message = jsonresponse.getString("message");
                    if (jsonresponse.getBoolean("status")) {
                        Log.d("info", jsonresponse.getJSONObject("tableinfo").toString());
                        JSONObject info = jsonresponse.getJSONObject("tableinfo");

                        sessionManager.createLoginSession(info.getString("table_id"),info.getString("name"));
                    });
                }
                Toast.makeText(getApplicationContext(),message,Toast.LENGTH_SHORT).show();

                startActivity(new
                Intent(getApplicationContext(),HomeActivity.class));
                finish();
            }
            else{
                //
                Toast.makeText(getApplicationContext(),message,Toast.LENGTH_SHORT).show();
                Utils.setAlertMsg(LoginActivity.this,message,"Try
again",false,true);
            }
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```

```
        Utils.setAlertMsg(LoginActivity.this,"Database Error","Try
again",false,true);

    }

    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            hidepDialog();
            VolleyLog.d("valley ", "Error: " + error.getMessage());
            Utils.setAlertMsg(LoginActivity.this,"Database not Connected","Try
again",false,true);

        }
    }
) {
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        params.put("userLogin", "true");
        params.put("table_id", ""+tab_id.getText());
        params.put("password", ""+tab_pass.getText());
        return params;
    }
};

// Adding request to request queue
AppController.getInstance().addToRequestQueue(stringRequest);
}

}
```

activity_customer.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
```

Order It- Android Based Food Ordering System

```
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/main_content"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
tools:context="com.example.jai.orderit.CustomerActivity">
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

<android.support.design.widget.AppBarLayout
    android:id="@+id/appbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.AppBarOverlay">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:layout_weight="1"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/AppTheme.PopupOverlay"
        app:title="@string/app_name">

    </android.support.v7.widget.Toolbar>

    <android.support.design.widget.TabLayout
        android:id="@+id/tabs"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
```

```
<android.support.design.widget.TabItem
    android:id="@+id/tabItem"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/tab_text_1" />

<android.support.design.widget.TabItem
    android:id="@+id/tabItem2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/tab_text_2" />

<android.support.design.widget.TabItem
    android:id="@+id/tabItem3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/tab_text_3" />

</android.support.design.widget.TabLayout>
</android.support.design.widget.AppBarLayout>
<FrameLayout
    android:layout_below="@+id/appbar"
    android:id="@+id/container"
    android:layout_marginBottom="60dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

<RelativeLayout
    android:id="@+id/relativeLayout"
    android:layout_width="200dp"
    android:layout_height="200dp"
```

Order It- Android Based Food Ordering System

```
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true">

<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="40dp"
    android:src="@drawable/water"
    android:visibility="invisible"
    app:backgroundTint="@color/colorAccent"
    app:pressedTranslationZ="12dp" />

<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_centerVertical="true"
    android:src="@drawable/bowl"
    android:visibility="invisible"
    app:backgroundTint="@color/colorAccent"
    app:elevation="6dp"
    app:pressedTranslationZ="12dp" />

<android.support.design.widget.FloatingActionButton
    android:id="@+id/Helpfab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
```

```
        android:layout_centerVertical="true"
        android:layout_marginStart="12dp"
        android:layout_marginTop="20dp"
        android:src="@drawable/order"
        app:backgroundTint="@color/colorAccent"
        app:elevation="6dp"
        app:pressedTranslationZ="12dp" />

<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_marginRight="40dp"
    android:src="@drawable/helper"
    android:visibility="invisible"
    app:backgroundTint="@color/colorAccent"
    app:elevation="6dp"
    app:pressedTranslationZ="12dp" />
</RelativeLayout>
<com.andremion.counterfab.CounterFab
    android:id="@+id/fab"
    android:layout_above="@+id/bottom_navigation"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:src="@drawable/ic_shopping_cart_black_24dp"
/>
```

```
<android.support.design.widget.BottomNavigationView  
    android:id="@+id/bottom_navigation"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:layout_weight="1.5"  
    app:menu="@menu/bottom_nav_menu"  
    app:itemBackground="@color/colorPrimary"  
    app:itemIconTint="@android:color/white"  
    app:itemTextColor="@android:color/white" />  
  
</RelativeLayout>  
</android.support.design.widget.CoordinatorLayout>
```

CustomerActivity.java:

```
package com.example.jai.orderit;  
  
import android.content.Intent;  
  
import android.support.annotation.NonNull;  
  
import android.support.design.widget.BottomNavigationView;  
  
import android.support.design.widget.FloatingActionButton;  
  
import android.support.design.widget.TabLayout;  
  
import android.support.v4.app.FragmentTransaction;  
  
import android.support.v7.app.AppCompatActivity;  
  
import android.support.v7.widget.Toolbar;  
  
import android.support.v4.app.Fragment;  
  
import android.support.v4.app.FragmentManager;  
  
import android.os.Bundle;  
  
import android.util.Log;  
  
import android.view.MenuItem;  
  
import android.view.View;  
  
import android.view.animation.Animation;  
  
import android.view.animation.AnimationUtils;  
  
import android.widget.Toast;  
  
import com.andremion.counterfab.CounterFab;
```

Order It- Android Based Food Ordering System

```
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.VolleyLog;
import com.android.volley.toolbox.StringRequest;
import com.example.jai.orderit.myfragments.items.NonVegFragment;
import com.example.jai.orderit.myfragments.items.OffersFragment;
import com.example.jai.orderit.myfragments.items.VegFragment;
import com.example.jai.orderit.utils.AppController;
import com.example.jai.orderit.utils.BottomNavigationViewHelper;
import com.example.jai.orderit.utils.DatabaseHandler;
import com.example.jai.orderit.utils.SessionManager;
import com.example.jai.orderit.utilsUrls;
import com.example.jai.orderit.utils.Utils;
import java.util.HashMap;
import java.util.Map;

public class CustomerActivity extends AppCompatActivity implements
View.OnClickListener {

    BottomNavigationView btnov;
    SessionManager sessionManager;
    CounterFab myfab;
    DatabaseHandler mycart;

    private Boolean isFabOpen = false;
    private FloatingActionButton helpfab, fab1, fab3, fab2;
    private Animation fab_open, fab_close, rotate_forward, rotate_backward;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

Order It- Android Based Food Ordering System

```
setContentView(R.layout.activity_customer);

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

helpfab = findViewById(R.id.Helpfab);
fab1 = findViewById(R.id.fab1);
fab2 = findViewById(R.id.fab2);
fab3 = findViewById(R.id.fab3);
fab_open = AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.fab_open);
fab_close = AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.fab_close);
rotate_forward = AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.rotate_forward);
rotate_backward = AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.rotate_backward);
helpfab.setOnClickListener(this);
fab1.setOnClickListener(this);
fab2.setOnClickListener(this);
fab3.setOnClickListener(this);

TabLayout tabLayout = (TabLayout) findViewById(R.id.tabs);
replaceFragment(new VegFragment());
btnov=findViewById(R.id.bottom_navigation);
BottomNavigationViewHelper.disableShiftMode(btnov);
sessionManager=new SessionManager(CustomerActivity.this);
toolbar.setTitle("Mr. / Mrs."+sessionManager.getCustomerName());

mycart=new DatabaseHandler(CustomerActivity.this);
tabLayout.setOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
    @Override
```

Order It- Android Based Food Ordering System

```
public void onTabSelected(TabLayout.Tab tab) {  
    setCurrentTabFragment(tab.getPosition());  
}  
  
@Override  
public void onTabUnselected(TabLayout.Tab tab) {}  
  
@Override  
public void onTabReselected(TabLayout.Tab tab) {}  
});  
  
btnov.setOnNavigationItemSelected(new  
BottomNavigationView.OnNavigationItemSelectedListener() {  
  
    @Override  
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
        switch (item.getItemId()) {  
            case R.id.action_order:  
                startActivity(new Intent(CustomerActivity.this,CartActivity.class));  
                break;  
            case R.id.action_help:  
                Toast.makeText(CustomerActivity.this,"Help  
place",Toast.LENGTH_SHORT).show();  
                break;  
            case R.id.action_view_order:  
                startActivity(new  
Intent(CustomerActivity.this,ViewOrdersActivity.class));  
                break;  
            case R.id.action_checkout:  
                startActivity(new Intent(CustomerActivity.this,CheckOutActivity.class));  
                finish();  
                break;  
        }  
        return true;  
    }  
});
```

```
myfab= findViewById(R.id.fab);
myfab.setCount(mycart.getCartSize());
myfab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(CustomerActivity.this,CartActivity.class));
        //      Snackbar.make(view, "Replace with your own action",
        Snackbar.LENGTH_LONG)
        //      .setAction("Action", null).show();
    }
});
private void setCurrentTabFragment(int position) {
    switch (position)
    {
        case 0:
            replaceFragment(new VegFragment());
            break;
        case 1:
            replaceFragment(new NonVegFragment());
            break;
        case 2:
            replaceFragment(new OffersFragment());
            break;
    }
}
public void replaceFragment(Fragment fragment) {
    FragmentManager fm = getSupportFragmentManager();
    FragmentTransaction ft = fm.beginTransaction();
    ft.replace(R.id.container, fragment);
```

```
        ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_OPEN);
        ft.commit();
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        myfab.setCount(mycart.getCartSize());
    }

    @Override
    public void onClick(View v) {
        int id = v.getId();
        switch (id) {
            case R.id.Helpfab:
                animateFAB();
                break;
            case R.id.fab1:
                fabRequest("iswater");
                animateFAB();
                break;
            case R.id.fab2:
                fabRequest("isbowl");
                animateFAB();
                break;
            case R.id.fab3:
                fabRequest("ishelper");
                animateFAB();
        }
    }
}
```

Order It- Android Based Food Ordering System

```
public void animateFAB() {  
    if (isFabOpen) {  
        helpfab.startAnimation(rotate_backward);  
        fab1.startAnimation(fab_close);  
        fab2.startAnimation(fab_close);  
        fab3.startAnimation(fab_close);  
        fab1.setClickable(false);  
        fab2.setClickable(false);  
        fab3.setClickable(false);  
        isFabOpen = false;  
        Log.d("Raj", "close");  
  
    } else {  
        helpfab.startAnimation(rotate_forward);  
        fab1.startAnimation(fab_open);  
        fab2.startAnimation(fab_open);  
        fab3.startAnimation(fab_open);  
        fab1.setClickable(true);  
        fab2.setClickable(true);  
        fab3.setClickable(true);  
        isFabOpen = true;  
        Log.d("Raj", "open");  
    }  
}  
  
private void fabRequest(final String req) {  
    StringRequest stringRequest = new StringRequest(  
        Request.Method.POST,  
       Urls.fabAPI,  
        new Response.Listener<String>() {  
            @Override  
            public void onResponse(String response) {  
                // Handle response  
            }  
        }  
    );  
    // Add request to the queue  
    RequestQueue requestQueue = Volley.newRequestQueue(this);  
    requestQueue.add(stringRequest);  
}
```

```
        Log.d("msg","sent");
    }
},
new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        VolleyLog.d("valley ", "Error: " + error.getMessage());
        Log.e("Error Req", "Error: " + error.getMessage());
        Utils.setAlertMsg(CustomerActivity.this,"DB not Connected","Try
again",false,true);
    }
    // hide the progress dialog
}
)
{
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        params.put(req, "true");
        params.put("tab_id",sessionManager.getTableId());
        return params;
    }
};

// Adding request to request queue
AppController.getInstance().addToRequestQueue(stringRequest);
}
}
```

Connection API (appconnections.php):

```
<?php
define('__ROOT__', dirname(dirname(__FILE__)));
require_once(__ROOT__.'/DbConnection.php');
//require_once 'DbConnection.php';
$response = array();
$ipadd='orderit.000webhostapp.com';
//table login validation
if(isset($_POST['userLogin'])){
    if(isset($_POST['table_id']) AND isset($_POST['password'])){
        $username = $_POST['table_id'];
        $password = $_POST['password'];
        $query_find = "SELECT * FROM tabs WHERE (table_id = '$username'
AND password= '$password') AND isdelete=0";
        $query_execute = mysqli_query($connection, $query_find);
        if(mysqli_num_rows($query_execute) > 0){
            $row = $query_execute->fetch_assoc();
            if(!$row['islogin']){
                $tableinfo = array(
                    'table_id'=>$row['table_id'],
                    'name'=>$row['name'],
                );
            }
            $response['message'] = 'Login successfull';
            $response['status']=true;
            $response['tableinfo'] = $tableinfo;
            $query_insert = "UPDATE tabs SET isLogin=1 WHERE
table_id='".$row['table_id']."' ";
            $query_status = mysqli_query($connection, $query_insert);
            if($query_status){
                $response['error'] = false;
            }
        }
    }
}
```

```
        else{
            $response['error'] = true;
            $response['status']=false;
        }
    }
else{
    $response['error'] = true;
    $response['status']=false;
    $response['message'] = 'Table Already Logged in
Someother Device';
}
}else{
    $response['error'] = false;
    $response['status']=false;
    $response['message'] = 'Invalid username or password';
}
}

else{
    $response['error'] = true;
    $response['status']=false
    $response['message'] = 'insufficient parameters supplied';
}
}

else if(isset($_POST['userLogout'])){ //Logout table
if(isset($_POST['table_id'])){
    $username = $_POST['table_id'];
    $query_find = "UPDATE tabs SET isLogin=0 WHERE
table_id='$username'";
    $query_status = mysqli_query($connection, $query_find);
    if($query_status){
        $response['error'] = false;
    }
}
```

```
$response['message']="Logout successfull";  
}  
else{  
    $response['error']= true;  
    $response['message']="Logout Failed";  
}  
}  
else{  
    $response['error']= true;  
    $response['message']= 'insufficient parameters supplied';  
}  
}  
}  
else if (isset($_POST['viewItems'])) {  
    if(isset($_POST['food_type']) AND isset($_POST['food_category'])){  
  
        $ftype=$_POST['food_type'];  
        $fcat=$_POST['food_category'];  
        $query_find = "SELECT * FROM food_item WHERE (food_type =  
'$ftype' AND food_category= '$fcat') AND (isavailable = 1 AND isdelete = 0)";  
        $query_result = mysqli_query($connection, $query_find);  
  
        if($query_result){  
            $response['error']= false;  
        }  
        else{  
            $response['error']= true;  
            $response['message']= 'DB error';  
        }  
        $items_info= array();  
        $i=0;  
  
        while($row =mysqli_fetch_assoc($query_result)) {

---


```

```
$temp = array(
    'item_id'=>$row['slno'],
    'title'=>$row['food_title'],
    'item_type'=>$ftype,
    'price'=>$row['price'],
    'pic_url'=>'http://'.$ipadd.'/orderit/uploads/'.$row['food_image'],
    'views_no'=>$row['views_no'],
    'rating'=>round($row['rating'],1),
    'description'=>$row['description'],
);

array_push($items_info,$temp);
$i++;
}

$response['message'] = 'Retrival successfull';
$response['items_count']=$i;
$response['items_info'] = $items_info;
}

else{
    $response['error'] = true;
    $response['message'] = 'insufficient parameters supplied';
}
}

else if(isset($_POST['customerLogin'])){
    if(isset($_POST['cust_name']) AND isset($_POST['tab_id'])){
        $cust_name = $_POST['cust_name'];
        $tab_id = $_POST['tab_id'];
        $query_test = "SELECT checkout_status FROM tabs WHERE table_id =
        '$tab_id' AND isdelete = 0";
        $query_test_execute = mysqli_query($connection, $query_test);
    }
}
```

Order It- Android Based Food Ordering System

```
if(mysqli_num_rows($query_test_execute) > 0){
    $trow =mysqli_fetch_assoc($query_test_execute);
    if($trow['checkout_status']==0)
        $order_id=0;
    $query_find = "SELECT * FROM placed_order";
    $query_execute = mysqli_query($connection,
$query_find);
    if(mysqli_num_rows($query_execute) == 0){
        $query_insert = "INSERT INTO placed_order
(slno, customer_name, table_id, order_date, isActive) VALUES (1, '$cust_name',
'$tab_id', NOW(), 0)";
        $order_id=1;
    }else{
        // $query_find = "SELECT MAX(slno) FROM placed_order";
        $query_find ="SELECT slno FROM placed_order
ORDER BY slno DESC LIMIT 1";
        $query_execute = mysqli_query($connection,
$query_find);
        $row =mysqli_fetch_assoc($query_execute);
        $order_id=($row['slno']+1);
        $query_insert = "INSERT INTO placed_order
(slno, customer_name, table_id, order_date, isActive, amount) VALUES ($order_id,
'$cust_name', '$tab_id', NOW(), 0,0.0)";
        }
        $query_status = mysqli_query($connection,
$query_insert);
        $query_insert = "UPDATE tabs SET isactive=1 WHERE
table_id='$tab_id' AND isdelete = 0";
        $query_change= mysqli_query($connection,
$query_insert);
        if($query_status AND $query_change){
```

Order It- Android Based Food Ordering System

```
$response['error'] = false;  
$response['message'] = 'Customer Login  
successfull';  
$response['order_id']=$order_id;  
}  
else{  
    $response['error'] = true;  
    $response['message'] = 'Data base error';  
}  
}  
else{  
    $response['error'] = true;  
    $response['message'] = 'Checkout Pending';  
}  
}  
else{  
    $response['error'] = true;  
    $response['message'] = 'Data base error';  
}  
}  
}  
else{  
    $response['error'] = true;  
    $response['message'] = 'insufficient parameters supplied';  
}  
}  
}  
else if(isset($_POST['insertItem'])){  
    if(isset($_POST['order_id']) AND isset($_POST['item_id']) AND  
    isset($_POST['quantity'])){  
        $order_id = $_POST['order_id'];  
        $item_id = $_POST['item_id'];  
        $quantity= $_POST['quantity'];  
    }
```

```
$query_insert = "INSERT INTO placed_order_items (order_id,
food_item_id, quantity) VALUES ($order_id, $item_id,$quantity)";

$query_status = mysqli_query($connection, $query_insert);

if($query_status){

    $response['error'] = false;

    $response['message'] = 'Item Inserted';

}

else{

    $response['error'] = true;

    $response['message'] = 'DB error';

}

else{

    $response['error'] = true;

    $response['message'] = 'insufficient parameters supplied';

}

}

else if(isset($_POST['orderNow'])){

    if(isset($_POST['order_id']) AND isset($_POST['tab_id'])) ){

        $order_id = $_POST['order_id'];

        $tab_id = $_POST['tab_id'];

        $query_find = "UPDATE placed_order SET isActive=1 WHERE
slno='$order_id'";

        $query_status = mysqli_query($connection, $query_find);

        $query_tab = "UPDATE tabs SET isorder=1 WHERE
table_id='$tab_id'";

        $query_tab_status = mysqli_query($connection, $query_tab);

        if($query_status AND $query_tab_status ){

            $response['error'] = false;

            $response['message']="Order sent";

        }

}
```

```
        else{
            $response['error'] = true;
            $response['message']="DB error";
        }
    }
else{
    $response['error'] = true;
    $response['message'] = 'insufficient parameters supplied';
}
}

else if(isset($_POST['orderStatus'])){
if(isset($_POST['order_id'])){

    $order_id = $_POST['order_id'];
    $query_find = "SELECT * FROM placed_order WHERE
sIno=$order_id";
    $query_status = mysqli_query($connection, $query_find);
    if(mysqli_num_rows($query_status) > 0){

        $response['error'] = false;
        $row =mysqli_fetch_assoc($query_status);
        if($row['isActive']){
            $response['message']="Previous Order is Processing";
            $response['isActive']=true;
        }
        else{
            $response['message']="Available";
            $response['isActive']=false;
        }
    }
    else{
        $response['error'] = true;
        $response['message']="DB error";
    }
}
```

```
    }

    else{
        $response['error'] = true;
        $response['message'] = 'insufficient parameters supplied';

    }

}

else if(isset($_POST['viewOrder'])){

    if(isset($_POST['order_id'])){
        $order_id = $_POST['order_id'];
        $query_find = "SELECT * FROM placed_order_items WHERE
order_id=$order_id";
        $query_status = mysqli_query($connection, $query_find);
        if(mysqli_num_rows($query_status) > 0){
            $order_info= array();
            $response['error'] = false;
            $i=0;
            $sum=0;
            while($row =mysqli_fetch_assoc($query_status)) {
                $num=$row['quantity'];
                $id=$row['food_item_id'];
                $query_item= "SELECT * FROM food_item WHERE
slno=$id";
                $query_result = mysqli_query($connection,
$query_item);
                if(mysqli_num_rows($query_result) > 0){
                    $rows =mysqli_fetch_assoc($query_result);
                    $total=($rows['price']*$num);
                    $temp = array(
                        'item_id'=>$rows['slno'],
                        'title'=>$rows['food_title'],

```

Order It- Android Based Food Ordering System

```
'price'=>$rows['price'],
'quantity'=>$num,
'amount'=>$total,
);
$i++;
$sum=$sum+$total;
array_push($order_info,$temp);
}

}

$response['message']="All orders Received";
$response['order_info']=$order_info;
$response['item_count'] = $i;
$response['total_bill'] = $sum;
}

else{
    $response['error'] = true;
    $response['message']="DB error";
}

}

else{
    $response['error'] = true;
    $response['message'] = 'insufficient parameters supplied';
}

}

else if(isset($_POST['checkoutUser'])){
    if(isset($_POST['tab_id']) ){
        $tab_id = $_POST['tab_id'];
        $query_find = "UPDATE tabs SET checkout_status=1 WHERE
table_id='$tab_id' AND isdelete = 0";
        $query_status = mysqli_query($connection, $query_find);
    }
}
```

```
if($query_status){  
    $response['error'] = false;  
    $response['message']="sent to Checkout";  
    $response['status']=true;  
}  
  
else{  
    $response['error'] = true;  
    $response['message']="DB error";  
}  
  
}  
  
else{  
    $response['error'] = true;  
    $response['message'] = 'insufficient parameters supplied';  
}  
  
}  
  
}  
  
else if(isset($_POST['reviewItems'])){  
    if(isset($_POST['order_id'])) {  
        $order_id = $_POST['order_id'];  
        $query_find = "SELECT DISTINCT food_item_id FROM  
placed_order_items WHERE order_id=$order_id";  
        $query_status = mysqli_query($connection, $query_find);  
        if(mysqli_num_rows($query_status) > 0){  
            $review_info= array();  
            $response['error'] = false;  
            $i=0;  
            while($row =mysqli_fetch_assoc($query_status)) {  
                $id=$row['food_item_id'];  
                $query_item= "SELECT * FROM food_item WHERE  
sIno=$id";  
                $query_result = mysqli_query($connection,  
$query_item);  
                $rows=mysqli_fetch_assoc($query_result);  
    
```

```
$temp = array(
    'item_id'=>$rows['slno'],
    'title'=>$rows['food_title'],
    'price'=>$rows['price'],
);

$i++;
array_push($review_info,$temp);

}

$response['message']="All Review Items Received";
$response['review_info']=$review_info;
}

else{
    $response['error']= true;
    $response['message']="DB error";
}

}

else{
    $response['error']= true;
    $response['message']= 'insufficient parameters supplied';
}

}

else if(isset($_POST['updateReview'])){
    if(isset($_POST['item_id']) AND isset($_POST['review_value'])) {
        $item_id = $_POST['item_id'];
        $review_value = $_POST['review_value'];
        $query_item= "SELECT * FROM food_item WHERE slno=$item_id";
        $query_result = mysqli_query($connection, $query_item);
        if(mysqli_num_rows($query_result) > 0){
            $rows =mysqli_fetch_assoc($query_result);
            $old_rate=$rows['rating'];
            $old_views=$rows['views_no'];
        }
    }
}
```

```
$value=$old_rate*$old_views;  
$no_views=$old_views+1;  
$rating_value=($value+$review_value)/$no_views;  
$query_find = "UPDATE food_item SET rating='$rating_value',  
views_no='$no_views' WHERE slno='$item_id"';  
$query_status = mysqli_query($connection, $query_find);  
if($query_status){  
    $response['error'] = false;  
    $response['message']="Rating Updated";  
}  
else{  
    $response['error'] = true;  
    $response['message']="DB error";  
}  
}  
else{  
    $response['error'] = true;  
    $response['message']="DB error";  
}  
}  
  
}  
  
else{  
    $response['error'] = true;  
    $response['message'] = 'insufficient parameters supplied';  
}  
}  
else{  
    $response['message'] = 'Invalid Operation Called';  
}  
  
echo json_encode($response);
```

?>

App fabs Connection API (fabconnect.php):

```
<?php
```

```
    require_once 'DbConnect.php';
```

```
    if(isset($_POST['iswater']) AND isset($_POST['tab_id'])) {
```

```
        $tab_id = $_POST['tab_id'];
```

```
        $query_find = "UPDATE tabs SET water_req=1 WHERE
```

```
        table_id='$tab_id' AND isdelete = 0";
```

```
        $query_status = mysqli_query($connection, $query_find);
```

```
        if(!$query_status){
```

```
            echo '<script> alert("Update Failed.") </script>';
```

```
        }
```

```
}
```

```
    if(isset($_POST['isbowl']) AND isset($_POST['tab_id'])) {
```

```
        $tab_id = $_POST['tab_id'];
```

```
        $query_find = "UPDATE tabs SET helper_req=1 WHERE
```

```
        table_id='$tab_id' AND isdelete = 0";
```

```
        $query_status = mysqli_query($connection, $query_find);
```

```
        if(!$query_status){
```

```
            echo '<script> alert("Update Failed.") </script>';
```

```
        }
```

```
}
```

Order It- Android Based Food Ordering System

```
if(isset($_POST['ishelper']) AND isset($_POST['tab_id'])) {  
    $tab_id = $_POST['tab_id'];  
    $query_find = "UPDATE tabs SET bowl_req=1 WHERE  
    table_id='$tab_id' AND isdelete = 0";  
    $query_status = mysqli_query($connection, $query_find);  
    if(!$query_status){  
        echo '<script> alert("Update Failed.") </script>';  
    }  
}  
?>
```

Web panel (Admin Interface):

Header.php:

```
<div class="row">
    <nav class="navbar navbar-default navbar-fixed-top" style="background-color: #D80000;">
        <div class="container">
            <div class="navbar-header">
                <a href="home.php">
                    <span class="h3" style="position: relative; top: 11px; left: 10px; margin-right: 10px; color: #FFFFFF">Digital Menu</span>
                </a>
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#mynav">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
            </div>
            <div class="collapse navbar-collapse" id="mynav">
                <ul class="nav navbar-nav">
                    <li class="dropdown">
                        <a class="dropdown-toggle" data-toggle="dropdown" style="color: #000000; font-weight: bold" href="#">Activities<span class="caret"></span></a>
                        <ul class="dropdown-menu">
                            <li class="dropdown-header">Order Status</li>
                            <li><a href="orders.php">Running Orders</a></li>
                        </ul>
                    </li>
                <?php
                if($_SESSION['type'] == "Admin")
```

```
echo '  
  
<li class="dropdown">  
    <a class="dropdown-toggle" data-toggle="dropdown" href="#" style="color: #000000; font-weight: bold">Management  
        <span class="caret"></span></a>  
        <ul class="dropdown-menu">  
            <li class="dropdown-header"><span class="glyphicon glyphicon-user"></span> User Activity</li>  
                <li><a href="UserAdd.php"><span class="glyphicon glyphicon-plus-sign"></span> Add User</a></li>  
                <li><a href="UserList.php"><span class="glyphicon glyphicon-th-list"></span> List User</a></li>  
                <li><a href="Userremove.php"><span class="glyphicon glyphicon-minus-sign"></span> Remove User</a></li>  
            <li class="divider"></li>  
            <li class="dropdown-header">Table Activity</li>  
                <li><a href="TableAdd.php"><span class="glyphicon glyphicon-plus-sign"></span> Add Table</a></li>  
                <li><a href="TableList.php"><span class="glyphicon glyphicon-th-list"></span> List Table</a></li>  
                <li><a href="tableremove.php"><span class="glyphicon glyphicon-minus-sign"></span> Remove Table</a></li>  
            <li class="divider"></li>  
            <li class="dropdown-header"><span class="glyphicon glyphicon-cutlery"></span> Food & Offer Item Activity</li>  
                <li><a href="FoodOfferAdd.php"><span class="glyphicon glyphicon-plus-sign"></span> Add Food/Offer</a></li>  
                <li><a href="FoodOfferList.php"><span class="glyphicon glyphicon-th-list"></span> List Food/Offer</a></li>  
            </ul>  
        </li>  
    '>?>
```

```
</ul>

<ul class="nav navbar-nav navbar-right">
    <div class="row"><?php setAdminLink() ?></div>
</ul>
</div>
</div>
</nav>
</div>
```

Footer.php

```
<div>
    <hr>
    <p align="center">Copy Right Reserved.</p>
</div>
```

orders. php:

```
<?php include('DBConnection.php') ?>
<?php if(!isset($_SESSION['email'])) :?>
<?php header("location: index.php"); ?>
<?php elseif($_SESSION['type'] == "User") :?>
<?php header("location: orders.php"); ?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>E-Menu</title>
<link href="css/bootstrap.min.css" rel="stylesheet">
<link rel="stylesheet" type="text/css" href="styles.css">
<link rel="stylesheet" type="text/css" href=".css/animate.css">
<link rel="stylesheet" type="text/css" href=".css/tab_btn.css" />
```

Order It- Android Based Food Ordering System

Order It- Android Based Food Ordering System

```
$water_color="my_btn_red";
if($row["bowl_req"])
$bowl_color="my_btn_green";
else
$bowl_color="my_btn_red";
if($row["helper_req"])
$helper_color="my_btn_green";
else
$helper_color="my_btn_red";
if($row["bowl_req"])
$bowl_color="my_btn_green";
else
$bowl_color="my_btn_red";
if($row["isorder"])
$order_color="my_btn_green";
else
$order_color="my_btn_red";
if($row["checkout_status"])
$tab_color="btn_pink";
echo '<th><BUTTON class="'.$tab_color.'" type="submit" height="10px" name="'.$row["table_id"].'" id="'.$row["table_id"].'">'.$row["name"].'</BUTTON><br>
<button class="'.$order_color.'" disabled></button>
<button class="'.$water_color.'" disabled></button>
<button class="'.$bowl_color.'" disabled></button>
<button class="'.$helper_color.'" disabled></button> </th>';
}
echo '</form></tr>';
```

```
}

echo '</table>';

}

?>

</div></div></div>

<?php include('footer.php');?>

<script src="js/bootstrap.min.js"></script>

</body>

</html>

viewFullTable.php:

<?php include('DBConnection.php') ?>

<?php

$order_id="";

$result =getTabs();

while($row =mysqli_fetch_assoc($result)) {

if(isset($_POST[$row['table_id']]))

$tab_id=$row['table_id'];

$tab_name=$row['name'];

$isactive=$row['isactive'];

$islogin=$row['islogin'];

$checkout_status=$row['checkout_status'];

$water_req=$row['water_req'];

$helper_req=$row['helper_req'];

$bowl_req=$row['bowl_req'];

$isorder=$row['isorder'];

$query_ordered = "SELECT sno FROM placed_order WHERE table_id = '$tab_id' AND

isActive=1 ";

$query_ordered_status = mysqli_query($connection, $query_ordered);

if(mysqli_num_rows($query_ordered_status) == 1){

$rows =mysqli_fetch_assoc($query_ordered_status);

$order_id=$rows['sno'];

}

}
```

```
else{
    $order_id=-999;
}
if($row["islogin"]){
    if($row["isactive"]){
        $tab_color="btn btn-warning btn-block";
    }
    else
        $tab_color="btn btn-success btn-block";
}
else
    $tab_color="btn btn-danger btn-block";
if($row["water_req"]){
    $water_color="btn btn-success btn-block";
}
else
    $water_color="btn btn-danger btn-block";
if($row["bowl_req"]){
    $bowl_color="btn btn-success btn-block";
}
else
    $bowl_color="btn btn-danger btn-block";
if($row["helper_req"]){
    $helper_color="btn btn-success btn-block";
}
else
    $helper_color="btn btn-danger btn-block";
if($row["bowl_req"])
    $bowl_color="btn btn-success btn-block";
else
    $bowl_color="btn btn-danger btn-block";
if($row["isorder"]){
    $order_color="btn btn-success btn-block";
}
else
    $order_color="btn btn-danger btn-block";
if($row["checkout_status"])
    
```

Order It- Android Based Food Ordering System

```
$btn_type="btn btn-primary btn-block";
else
$btn_type="my_btn_hidden";
}
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>E-Menu</title>
<!-- Bootstrap -->
<link href="css/bootstrap.min.css" rel="stylesheet">
<link rel="stylesheet" type="text/css" href="styles.css">
<link rel="stylesheet" type="text/css" href=".css/animate.css">
<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
<link rel="stylesheet" type="text/css" href=".css/tab_btn.css" />
<style type="text/css">
.my_btn_back{
margin-top: 20px;
background-color: #f4511e;
text-align: center;
border-radius: 10px;
font-size: 18px;
color: #FFFFFF;
width:200px ;
height: 50px;
}
</style>
</head>
```

```
<body>
<div class="container body-height">
<?php include('header.php') ?>
<div class="row" style="background-image: url('img/fade.jpg');">
<div class="col-sm-9">
<h1 style="color:#FFFFFF;font-size: 54px; font-weight: 200; ""><br><?php
echo"".$tab_name;?> Status</h1>
</div> </div> <div>
<a href="orders.php"><input type="button" name="back" value="Back to Orders"
class="my_btn_back"></a> </div>
<div class="row">
<div class="container-fluid mt-10" style="border: 2px solid; margin-top: 15px">
<div class="row" style="margin-top:15px">
<div class="col-md-12 text-center"><button type="button" class="<?php echo
$tab_color ?>"><?php echo"".$tab_name; ?></button></div> </div>
<div class="row" style="margin: 20px">
<div class="col-sm-3" style="margin: 0px"><button type="button" class="<?php echo
$order_color ?>"></button></div>
<div class="col-sm-3" style="margin: 0px"><button type="button" class="<?php echo
$water_color ?>"></button></div>
<div class="col-sm-3" style="margin: 0px"><button type="button" class="<?php echo
$bowl_color ?>"></button></div>
<div class="col-sm-3" style="margin: 0px"><button type="button" class="<?php echo
$helper_color ?>"></button></div></div>
<div class="row">
<div class="col-md-12">
<button type="button" class="btn btn-primary btn-block">Items Ordered</button>
<table class="table table-hover">
```

Order It- Android Based Food Ordering System

```
<thead>
<th>S.NO</th><th>Title</th><th>Price</th><th>Type</th><th>Category</th><th>Q
uantity</th><th>Process</th>
</thead>
<tbody>
<?php
$query_order = "SELECT * FROM placed_order_items WHERE order_id='\$order_id'";
$query_result = mysqli_query($connection, $query_order);
if(mysqli_num_rows($query_result) >0){
$i=1;
while($rows =mysqli_fetch_array($query_result)) {
$query_item = "SELECT * FROM food_item WHERE sno=". $rows['food_item_id']. "";
$result = mysqli_query($connection, $query_item);
$row =mysqli_fetch_array($result);
echo '<tr>
<td>' . $i .
<td>' . $row['food_title'] .
<td>' . $row['price'] .
<td>' . $row['food_type'] .
<td>' . $row['food_category'] .
<td>' . $rows['quantity'] . <td>';
if($rows['process'] == 1)
echo 'Done';
else
echo 'Not Done';
echo'</td></tr>';
$i++;} }
?>
</tbody>
</table>
</div> </div>
<div class="row" style="margin: 25px">
```

Order It- Android Based Food Ordering System

```
<div class="col-sm-3" ><form method="POST" action="orders.php">
<input type='hidden' name='tab_id' value='<?php echo "$tab_id";?>'/>
<input type='hidden' name='order_id' value='<?php echo "$order_id";?>'/>
<button type="submit" name="isOrderSent" class="btn btn-info btn-block">Order
sent</button></form> </div>

<div class="col-sm-3" ><form method="POST" action="orders.php"><input
type='hidden' name='tab_id' value='<?php echo "$tab_id";?>'> <button
type="submit" name="iswaterSent" class="btn btn-info btn-block">Water
sent</button></form></div>

<div class="col-sm-3" ><form method="POST" action="orders.php"><input
type='hidden' name='tab_id' value='<?php echo "$tab_id";?>'> <button
type="submit" name="isbowlSent" class="btn btn-info btn-block">FingerBowl
sent</button></form></div>

<div class="col-sm-3" ><form method="POST" action="orders.php"><input
type='hidden' name='tab_id' value='<?php echo "$tab_id";?>'> <button
type="submit" name="ishelperSent" class="btn btn-info btn-block">Helper
sent</button></form> </div>

<div class="row" style="margin-bottom: 30px">
<div class="col-md-12 text-center">
<form method="POST" action="billGenerator.php">
<input type='hidden' name='tab_id' value='<?php echo "$tab_id";?>'>
<input type='hidden' name='order_id' value='<?php echo "$order_id";?>'>
<button type="submit" class="<?php echo $btn_type ?>" name='send_bill' >Send
Bill</button>
</form> </div> </div> </div>

<div class="col-xs-12"></div> </div> </div>

<!--Footer Area-->
<?php include('footer.php');?>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

billgenerator.php

```
<?php include('DBConnection.php') ?>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <!-- The above 3 meta tags *must* come first in the head; any other head content
        must come *after* these tags -->
        <title>E-Menu</title>
        <!-- Bootstrap -->
        <link href="css/bootstrap.min.css" rel="stylesheet">
        <link rel="stylesheet" type="text/css" href="styles.css">
        <link rel="stylesheet" type="text/css" href=".css/animate.css">
        <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
        queries -->
        <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
        <!--[if lt IE 9]>
            <script
                src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
            <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
        <![endif]-->
        <link rel="stylesheet" type="text/css" href=".css/tab_btn.css" />
        <style type="text/css">
            table{
                width: 50%;
                border: 2px solid black;
                padding: 10px
            }
            th,td{
                padding: 10px;
```

```
text-align: left;
vertical-align: middle;
height: 25px;
}

@media print{
    .noprint{display: none;}
    .my_btn_bill{display: none;}
}

.my_btn_bill{
background-color: #4CAF50;
text-align: center;
font-size: 18px;
color: #FFFFFF;
width:200px ;
height: 50px;
}

</style>
</head>
<body>
<div class="container body-height" id="print_info">
<?php
$order_id= $_POST['order_id'];
$tab_id=$_POST['tab_id'];
$query_main = "SELECT * FROM placed_order WHERE slno=$order_id";
$query_main_status = mysqli_query($connection, $query_main);
$data=mysqli_fetch_assoc($query_main_status);

?>
<br><br><br>
<h3 align="center">Bill Receipt</h3>
```

```
<table border="1" width="50%" align="center">

<tr>
    <th colspan="2">Customer Name</th><td colspan="4"><?php echo
$data['customer_name'];?></td>
</tr>

<tr>
    <th>Date</th> <td><?php echo date('d/M/Y') ;?></td>
    <th>Table Id</th> <td><?php echo $data['table_id']; ?></td>
    <th>Receipt No</th><td><?php echo $data['slno']; ?></td>
</tr>

</table>

<table border="1" width="50%" align="center">

<tr>
    <th>S.NO</th><th>Item
Name</th><th>Price</th><th>Quantity</th><th>Ammount</th>
</tr>';

<?php
$query_bill = "SELECT * FROM placed_order_items WHERE
order_id=$order_id";
$query_bill_status = mysqli_query($connection, $query_bill);
if(mysqli_num_rows($query_bill_status) > 0){
    $i=1;
    $sum=0;
    while($row =mysqli_fetch_assoc($query_bill_status)) {
        $num=$row['quantity'];
        $id=$row['food_item_id'];
        $query_item= "SELECT * FROM food_item WHERE slno=$id";
        $query_result = mysqli_query($connection, $query_item);
        if(mysqli_num_rows($query_result) > 0){
```

Order It- Android Based Food Ordering System

```
$rows =mysqli_fetch_assoc($query_result);

$total=($rows['price']*$num);

echo "<tr><td>".$i."</td><td>".
$rows['food_title']."'</td><td>".
$rows['price']."'</td><td>".
$num."'</td><td>".
$total."'</td></tr>";

$i++;

$sum=$sum+$total;

}

}

echo "<tr><th colspan='4'>Total Bill(Including GST)</th><td>".$sum."</td></tr>";

echo "<tr><th colspan='4'>Bill Passed By </th><td>".$_SESSION['user_name']."'</td></tr>";

}

else{
    echo '<script> alert("Not Ordered any Thing") </script>';
}

?>

</table>
<br><br>
<center><input type="button" class="noprint" value="Print"
onclick="window.print()">
<br><br>
<form method="POST" action="orders.php">
<input type='hidden' name='tab_id' value='<?php echo "$tab_id";?>'/>
```

Order It- Android Based Food Ordering System

```
<input type='hidden' name='order_id' value='<?php echo  
"$order_id";?>'>  
<button type="submit" class="my_btn_bill" id="noprint" name='pay_bill'  
>Payment Done</button>  
</form>  
</center>  
</div>  
<!--JavaScript code-->  
<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->  
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js""></script>  
<!-- Include all compiled plugins (below), or include individual files as needed -->  
<script src="js/bootstrap.min.js"></script>  
</body>  
</html>
```

9.2 APPENDIX – II: SCREEN SHOTS

Order It: Android Application (User Interface)

Splash Screen & Table Login Screen

Splash screen is the first screen of app usually contains App logo for few seconds it will be redirected to table login screen

Table login is used to login the table with is id and password if it is valid it will be redirected to Table home screen

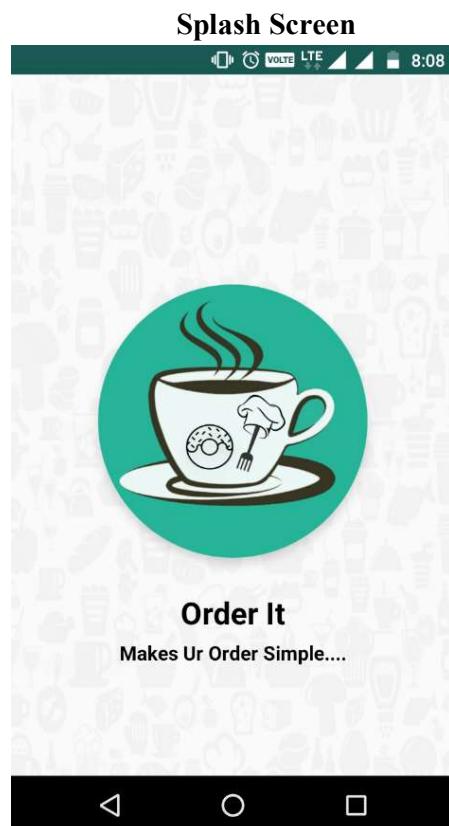


Figure 9.2.1: Android App Screenshot-1

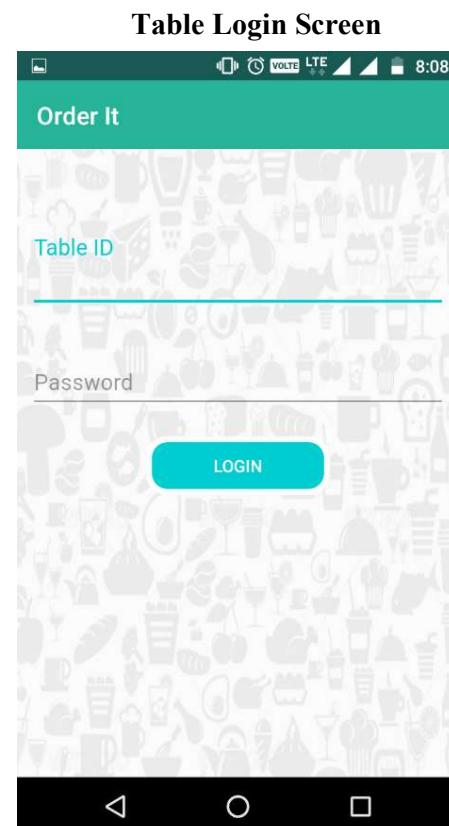


Figure 9.2.2: Android App Screenshot-2

Table Home Screen & Customer Login Screen

Table Home screen is navigation menu screen it contains options to home, contact us, help, Logout default was home which contains the customer login screen where customer will give his name to login. It will be redirected to Customer Home Screen.

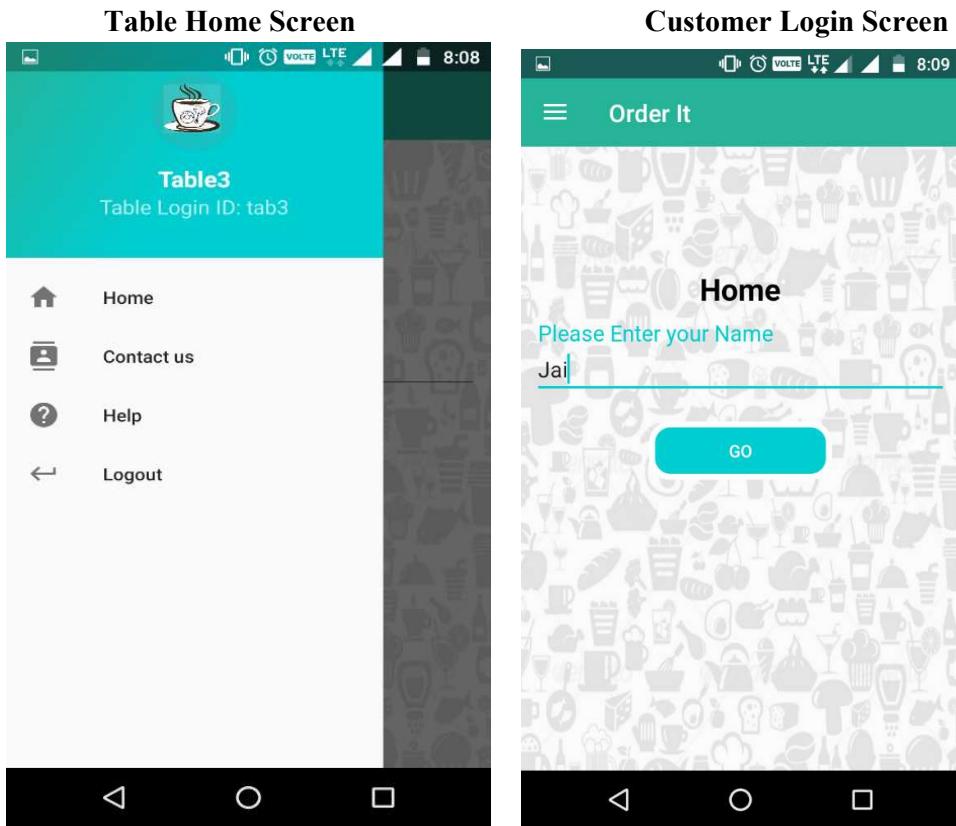


Figure 9.2.3: Android App Screenshot-3

Figure 9.2.4: Android App Screenshot-4

Order It- Android Based Food Ordering System

Customer Home Screen & Food Category Screen

Customer home screen is tabbed screen with three screens Veg, Non-Veg, Offers which contains the list of subcategories and food items lists etc. In additionally it has options to water/ helper/ finger bowl requests

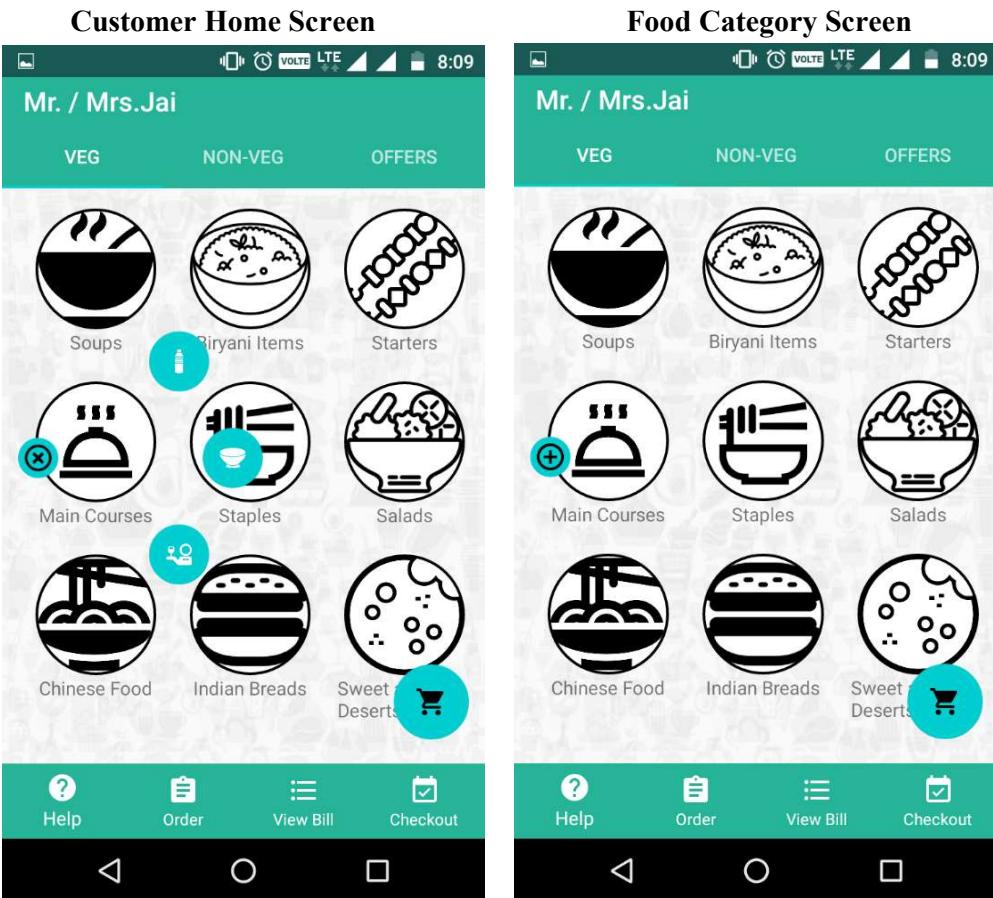


Figure 9.2.5: Android App Screenshot-5

Figure 9.2.6: Android App Screenshot-6

Item List Screen & Item Full Description Screen

Item list screen is a list of item list on clicking on the item will redirected to Item full view. We can add the items to screen using + button on Item list and add cart in full item view

Item List Screen

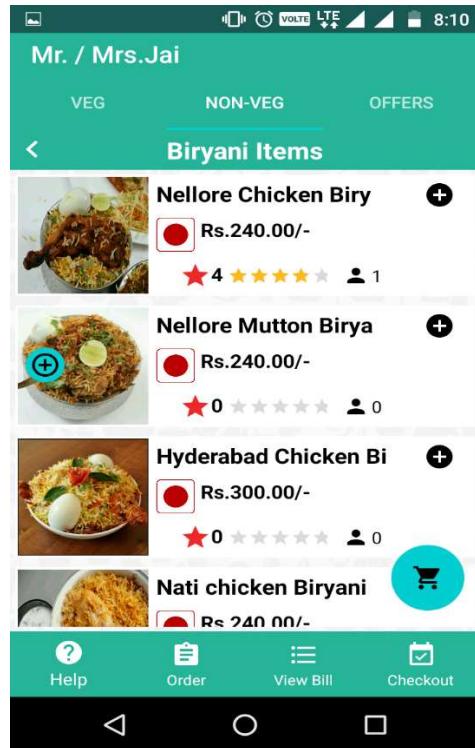


Figure 9.2.7: Android App Screenshot-7

Item Full Description Screen

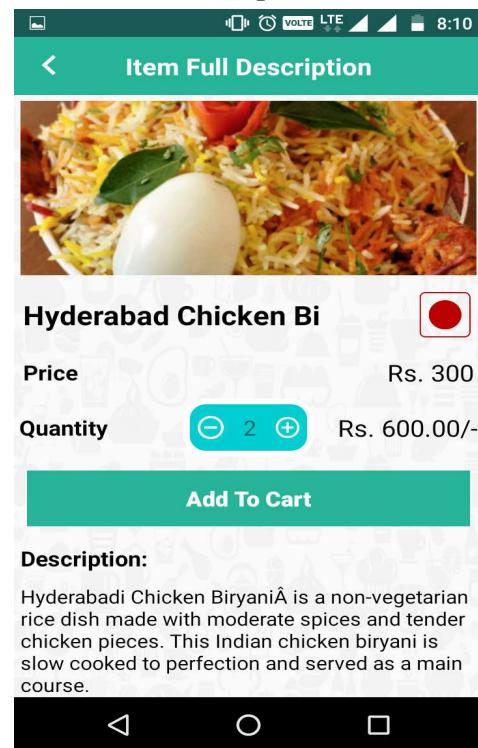


Figure 9.2.8: Android App Screenshot-8

Items in Cart Screen & Bill Generating Screen

Cart screen contains the items that are selected by user on clicking on the order now will send request to admin

View bill option in customer home will opens bill generating screen

Items in Cart Screen

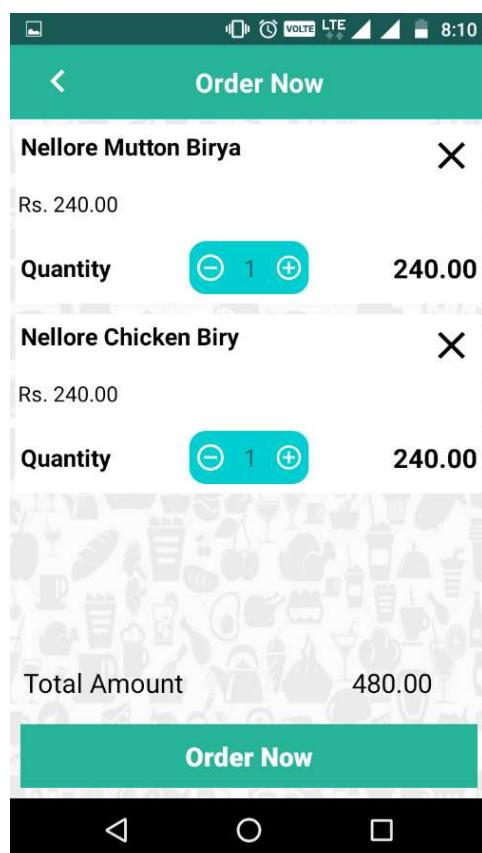


Figure 9.2.9: Android App Screenshot-9

Bill Generating Screen

Table1			
Name	Price	Quantity	Amount
Jai		26	
Nellore Mutton Birya	Rs. 240.00	1	Rs. 240.00
Nellore Chicken Biry	Rs. 240.00	1	Rs. 240.00

Total Amount: Rs. 480/-

Figure 9.2.10: Android App Screenshot-10

Checkout Screen & Contact Us Screen

Checkout screen will be open to take reviews and to checkout from table and contact us will display contact information of developers.

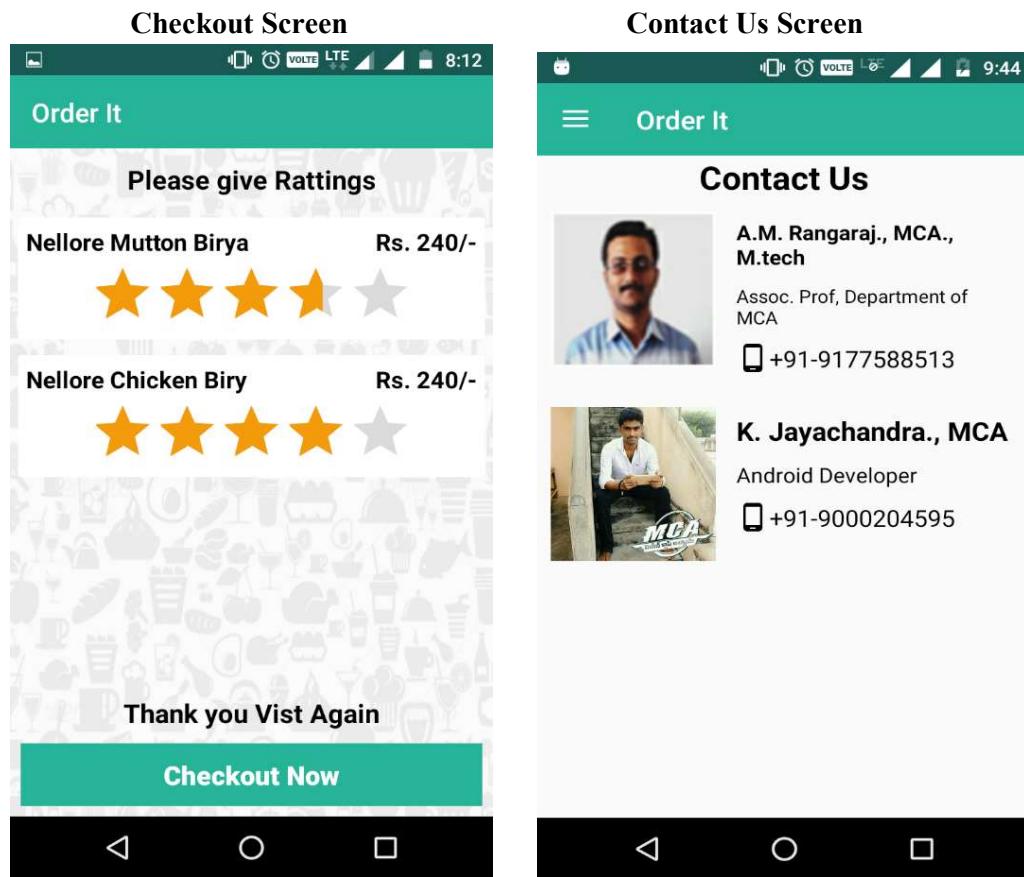


Figure 9.2.11: Android App Screenshot-11

Figure 9.2.12: Android App Screenshot-12

Order It: Web Admin Panel (Admin Interface) Login Page

It is a user login page if the user is admin it will redirect to admin page otherwise it will be redirected to others user page

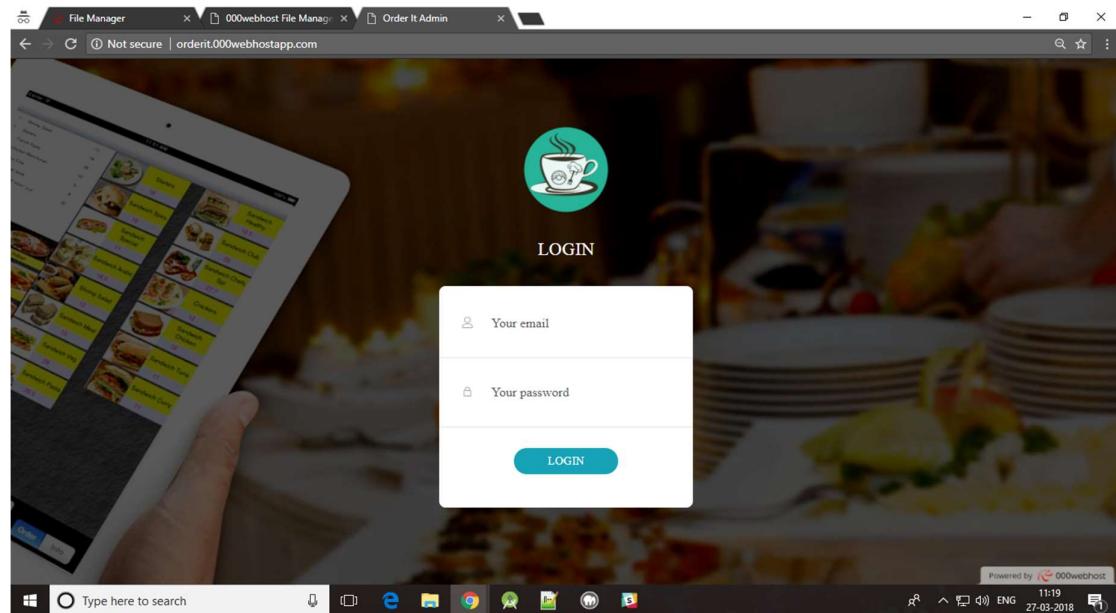


Figure 9.2.13: Web panel Screenshot-1

Other Users Home Page

Other user home page have only orders panel which will be redirected to full table view page

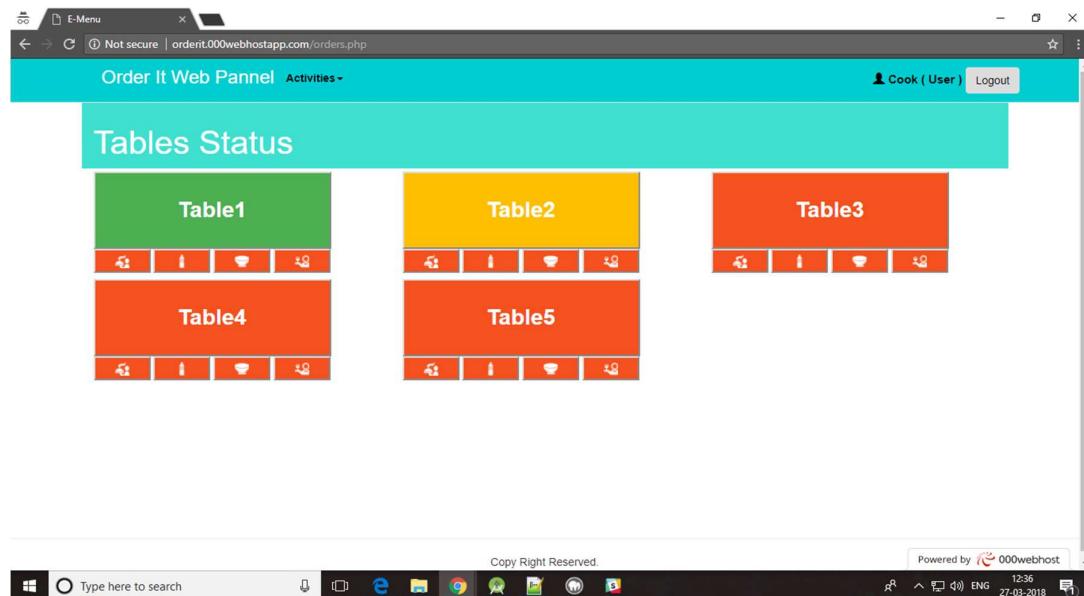


Figure 9.2.14: Web panel Screenshot-2

Order It- Android Based Food Ordering System

Full Table description Page

It contains all table information including items information & water, helper, bowl request and bill generation etc.

The screenshot shows a web browser window titled 'E-Menu' with the URL 'orderit.000webhostapp.com/viewFullTable.php'. The page header includes 'Order It Web Panel' and 'Activities'. A user profile 'Cook (User)' and a 'Logout' button are visible. The main content area is titled 'Table1 Status' and displays a table of ordered items. The table has columns for S.NO, Title, Price, Type, Category, Quantity, and Process. The items listed are Chilli Chicken, Chicken Manchurian, and Chowmein, all marked as 'Done'. Below the table are buttons for 'Order sent', 'Water sent', 'FingerBowl sent', and 'Helper sent', followed by a 'Send Bill' button. At the bottom of the page are copyright and activation notices for Windows, along with a taskbar showing various application icons and system status.

Figure 9.2.15: Web panel Screenshot-3

Bill Generation Page

It generates the complete bill based on table

The screenshot shows a web browser window titled 'E-Menu' with the URL 'orderit.000webhostapp.com/billGenerator.php'. The page header includes 'Order It Web Panel' and 'Activities'. A user profile 'Cook (User)' and a 'Logout' button are visible. The main content area is titled 'Bill Receipt' and displays a receipt for a customer named 'Chandu'. The receipt includes fields for Customer Name, Date (27/Mar/2018), Table Id (tab1), and Receipt No (27). Below this is a table of ordered items with columns for S.NO, Item Name, Price, Quantity, and Amount. The items listed are Chilli Chicken, Chicken Manchurian, and Chowmein, with a total amount of 1580. The receipt also includes a 'Total Bill(Including GST)' row and a 'Bill Passed By' row. At the bottom are 'Print' and 'Payment Done' buttons. At the very bottom of the page are copyright and activation notices for Windows, along with a taskbar showing various application icons and system status.

Figure 9.2.16: Web panel Screenshot-4

Order It- Android Based Food Ordering System

Admin Home Page

Admin has additional features such as management which involves all dashboard information

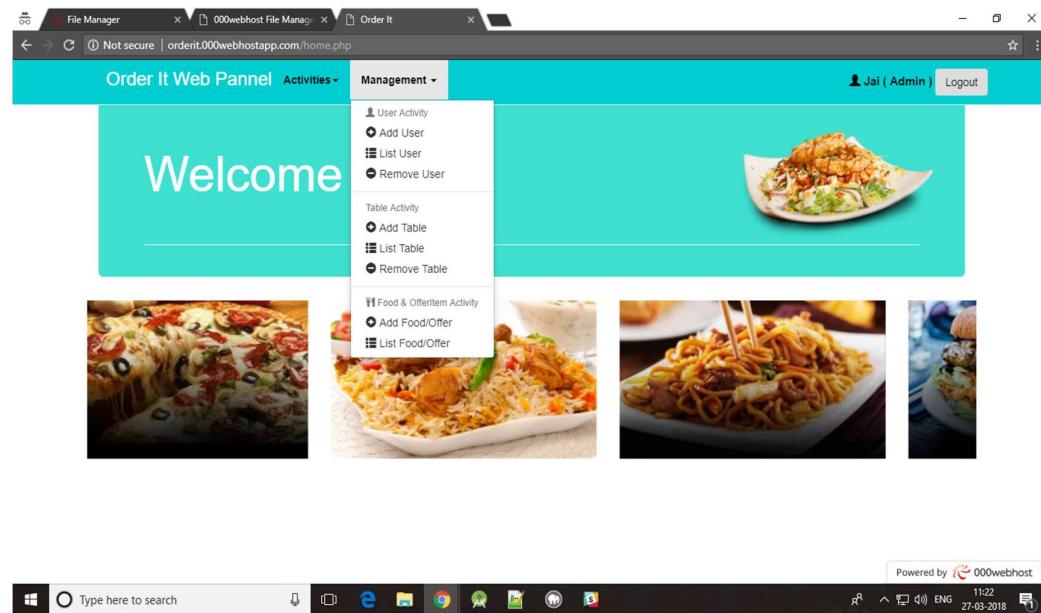


Figure 9.2.17: Web panel Screenshot-5

Add User Page

To add the new user. It is possible for admin user only.

A screenshot of a web browser displaying the 'UserAdd.php' page. The title bar shows 'E-Menu' and 'orderit.000webhostapp.com/UserAdd.php'. The main content area has a teal header with 'Add a new User'. Below it are input fields for Name, Mobile, Email, Password, Re-Password, and User Type (with 'Others' selected). A 'Submit' button is at the bottom. The status bar at the bottom shows system information like battery level, signal strength, and date/time.

Figure 9.2.18: Web panel Screenshot-6

Order It- Android Based Food Ordering System

List Users Page

To view the all user information. It is possible for admin user only.

The screenshot shows a web browser window titled 'E-Menu' with the URL 'orderit.000webhostapp.com/UserList.php'. The page header includes 'Order It Web Panel', 'Activities - Management -', and a user session 'Jai (Admin) | Logout'. The main content area is titled 'Users List' and displays a table of user data:

Sl No	Name	Mobile	Email	Password	Create Date	User Type
1	Jai	9000204595	jai@mine.com	4123	2018-03-21 05:19:51	Admin
2	keerthi	9701841415	keerthi@gmail.com	123#@abcD	2018-03-14 16:02:00	User
3	Cook	9000204595	cook@orderit.com	Orderit@c	2018-03-27 05:59:54	User

At the bottom of the page, there is a copyright notice 'Copy Right Reserved.' and a Windows taskbar at the bottom with various icons and system status.

Figure 9.2.19: Web panel Screenshot-7

Remove User Page

To remove the existing user. Only we can remove other users. It is possible for admin user only.

The screenshot shows a web browser window titled 'E-Menu' with the URL 'orderit.000webhostapp.com/Userremove.php'. The page header includes 'Order It Web Panel', 'Activities - Management -', and a user session 'Jai (Admin) | Logout'. The main content area is titled 'Remove User' and displays a form to select a user for deletion:

Select user to be Deleted
keerthi
Submit

At the bottom of the page, there is a copyright notice 'Copy Right Reserved.' and a Windows taskbar at the bottom with various icons and system status.

Figure 9.2.20: Web panel Screenshot-8

Order It- Android Based Food Ordering System

Add Table Page

To add new hotel table. It is possible for admin user only.

Table Name

Table ID

Password

Re-Password

Figure 9.2.21: Web panel Screenshot-9

List Tables Page

To view the all table information. It is possible for admin user only.

Sl No	Name	Table Id	Is Active	Is Login	checkout_status
1	Table1	tab1	No	No	Done
2	Table2	tab2	No	No	Done
3	Table3	tab3	No	No	Done
4	Table4	tab4	No	No	Done
5	Table5	tab5	No	No	Done

Figure 9.2.22: Web panel Screenshot-10

Remove Table Page

To remove exiting table. We can only remove non active tables only It is possible for admin user only.

Select Table to be Deleted

Note:
Please Logout the table to Delete

Copy Right Reserved.

Figure 9.2.23: Web panel Screenshot-11

Order It- Android Based Food Ordering System

Add Foods Offers Page:

To add item offer to customers. It is possible for admin user only.

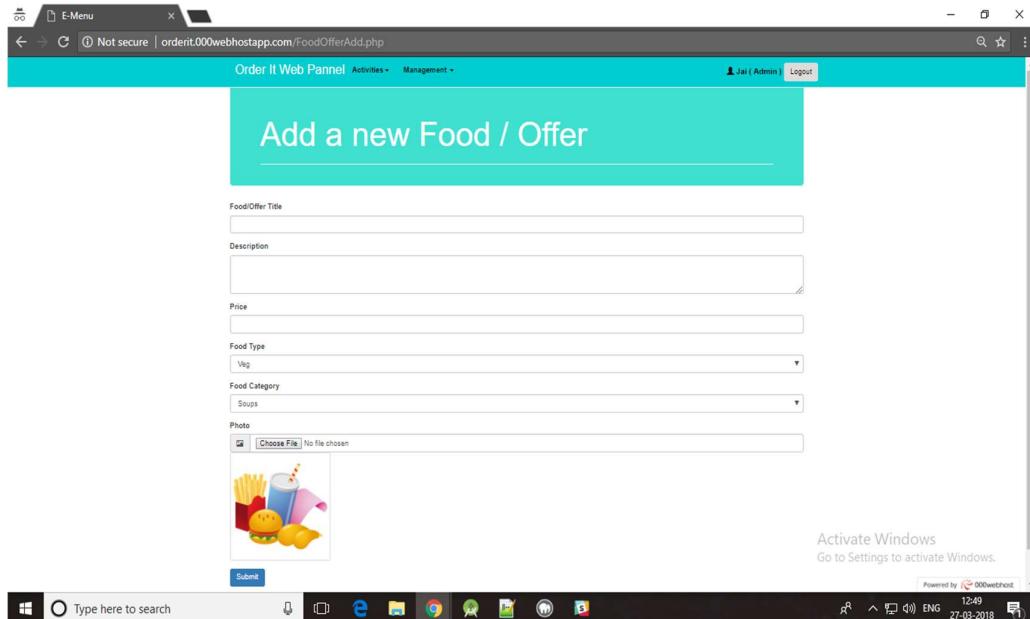


Figure 9.2.24: Web panel Screenshot-12

List Food offers Page:

To view the all Item information and also to update it. It is possible for admin user only.

The screenshot shows a Windows desktop environment with a web browser window open to 'FoodOfferList.php'. The browser title bar says 'E-Menu' and the address bar shows 'Not secure | orderit.000webhostapp.com/FoodOfferList.php'. The main content area has a teal header 'Food Offers List'. Below it is a table listing food offers with columns: Image, Title, Desc, Price, Views, Rating, Type, Category, IsAvailable, and Action. The table contains five rows of data. At the bottom right of the table, there is an 'Activate Windows' message. The bottom right corner of the screen shows a Windows taskbar with various icons and a system status bar indicating '12:49 27-03-2018'.

Image	Title	Desc	Price	Views	Rating	Type	Category	IsAvailable	Action
	Cabbage And Paneer R	Paneer Roll is a sumptuous treat with a peppery flavor. The crunch of grated cabbage and spring onions together with the succulence of crumbled paneer gives these rolls an awesome mouth-feel, while a dash of hot chili sauce	150	2	4.35	Veg	Soups	Yes	<button>Update</button>
	babycorn pudhina	mint and coriander add an interesting color and flavor to the crunchy baby corn... Boil the baby corn in salted water till they become soft and are cooked... Tarla Dalal delighted you loved the Baby Corn A Phudina recipe.	120	1	5	Veg	Starters	Yes	<button>Update</button>
	American Chop Suey	American chop suey is an American pasta dish popular in New England. It is related to other popular and similarly regional pasta dishes, like chili mac. Despite its name, it has only a very distant relation to Chinese and American Chinese cuisines.	180	0	0	Veg	Starters	Yes	<button>Update</button>
	Aloo Aur Shakarkand	Also our Shakarkand ki Chatat, most of us consider sweet potatoes (Shakarkand) to be a very boring vegetable, only to be eaten during religious fasts and then too, it is only baked or boiled before it is peeled and eaten.	150	0	0	Veg	Starters	Yes	<button>Update</button>
	Achari paneer tikka	achari paneer tikkaA recipe with step by step photos & another	200	1	4	Veg	Starters	Yes	<button>Update</button>

Figure 9.2.25: Web panel Screenshot-13

Order It- Android Based Food Ordering System

Sales Report Page:

To view the all sales information between selected dates. It is possible for admin user only.

The screenshot shows a web browser window titled "Order It". The URL is https://orderit.000webhostapp.com/sales.php. The page header includes "Order it Web Panel" and "Management". A "Logout" link is visible in the top right. The main content is titled "Sales Report". It features two date input fields: "Select From Date" (10-April-2018) and "Select To Date" (10-April-2018). Below these is a "Submit" button. The table displays the following data:

S.no	Date	Customer name	Bill Passed	Amount
1	2018-04-01 11:20:13	MamaJ	Jai	1700
2	2018-04-01 08:00:11	Jai	Jai	800
3	2018-04-07 03:11:05	Jai	Jai	540
4	2018-04-04 09:50:39	kaethi	Jai	810
5	2018-04-04 08:52:12	sri	Jai	810

Below the table, it says "From 30-March-2018 to 10-April-2018". At the bottom, it shows "Total Orders : 5" and "Total Sales : Rs.4320/-". There is also a "Print" button. The status bar at the bottom right shows "Copy Right Reserved.", "05:20 PM", "10-04-2018", and a battery icon.

Figure 9.2.26: Web panel Screenshot-14

APPENDIX – III : REFERENCES

1. Introduction to Android: <http://developer.android.com/guide/index.html>.
2. Android API: <http://developer.android.com/reference/packages.html>
3. Java 6 API: <http://docs.oracle.com/javase/6/docs/api/>
4. Android
Fundamentals: <http://developer.android.com/guide/components/fundamentals.html>
5. The Java Tutorials: <http://docs.oracle.com/javase/tutorial/>
6. Android User
Interfaces: <http://developer.android.com/guide/topics/ui/index.html>
7. Layout: <http://developer.android.com/guide/topics/ui/declaring-layout.html>
8. Common
Tasks: <http://developer.android.com/guide/appendix/faq/commontasks.html>
9. Google Maps: <http://code.google.com/android/add-ons/google-apis/maps-overview.html>
10. Iconography: http://developer.android.com/guide/practices/ui_guidelines/icon_design.html
11. Sample Source
Code: <http://developer.android.com/resources/samples/get.html>
12. Android Training: <http://developer.android.com/training/index.html>.
13. Android Developer's Blog: <http://android-developers.blogspot.com/>
14. Developer FAQ: <http://developer.android.com/resources/faq/>
15. Developer Forums: <http://developer.android.com/resources/community-groups.html>
16. Android Developer's Group: <http://groups.google.com/group/android-developers?lnk=gtp>
17. XDA-Developers Forums: <http://forum.xda-developers.com/>
18. *Android Power*: <http://blogs.computerworld.com/raphael>
19. *The Droid Guy*: <http://thedroidguy.com/>
20. *Android Guys*: <http://www.androidguys.com/>