



Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer

Engineering

B.Tech CE Semester – VI

Subject: “Software Development Project”

Project Title:

“Everfino”

Submitted By

Prashant Ramani (18CEUOD011) (CE110)

Krupal Shah (18CEUOD005) (CE123)

Akash Suvagiya (18CEUOD006) (CE131)

Guided by

Prof. Ankit Vaishnav



Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer

Engineering

CERTIFICATE

This is to certify **Software Development Project** entitled “**Everfino**” is the bonofied report of work carried out by **Prashant Ramani (18CEUOD011) (CE110), Krupal Shah (18CEUOD005) (CE005) and Akash Suvagiya (18CEUOD006) (CE131)** of Computer Engineering, Semester VI, academic year 2019-2020, under our supervision and guidance.

Guide

Prof. Ankit Vaishnav

Assistant Professor of Department of
Computer Engineering,

Dharmsinh Desai

University, Nadiad

HOD

Dr. C. K. Bhensdadia

Head of Department of
Computer Engineering,

Dharmsinh Desai

University, Nadiad

INDEX

Sr. No.	Content	Page No.
1.	Abstract	1
2.	Introduction of Project	3
	2.1 Brief Introduction	4
	2.2 Technology and Tools Used	5
3.	Software Requirement Specification	8
	3.1 Scope	9
	3.2 System Functional Requirements	9
	3.3 Other Non-Functional Requirements	16
4.	Design	17
	4.1 Use Case Diagram	18
	4.2 Sequence Diagram	19
	4.3 Activity Diagram	22
	4.4 E-R Diagram	25
	4.5 Data Dictionary	26
5.	Implementation Details	30
	5.1 Modules Description	31
6.	Testing	32
7.	Screenshots	35
8.	Conclusion	44
9.	Limitations and Future Extension	46
10.	Bibliography	48

1. Abstract

Abstract

Everfino is Client-Server Android Based Application. Gameplet is develop in Node.js Api and Android Application, Node.js used for server side Api to provide services and client are made by using Android Application.

Everfino has very variety of Restaurant that can be Managed by Admin and Restaurant Manager.

Everfino can be used by various Restaurant to easily manage their record and provide transparent service to user. Using our system EndUser can easily order any item in Restaurant and track live status of item by only scanning the QR code of the table.

2. Introduction of Project

2.1 Brief Introduction

Project Name:	Gameplet
Project Definition/Aim:	The main aim of project is to provide better maintainability to Restaurant and transparent service to User
Develop For :	Restaurant and Users
Front End :	Android Application
Back End :	Node.js
Other Technologies :	> MySQL
Documentation tools:	MS Word
IDE:	Android Studio

2.2 Tools and Technology Used

- **Android Studio**

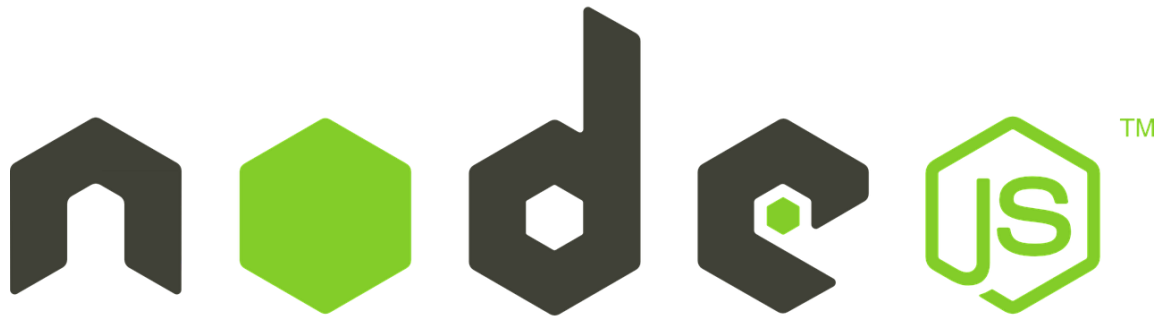


Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

- **Node.js**



Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts.

Though .js is the standard filename extension for JavaScript code, the name "Node.js" doesn't refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

The Node.js distributed development project was previously governed by the Node.js Foundation, and has now merged with the JS Foundation to form the OpenJS Foundation, which is facilitated by the Linux Foundation's Collaborative Projects program.

- **MySql**



MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Wideness's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

3. Software Requirement Specifications

3.1 Scope

The scope of this system is very large so it used anywhere is the world to provide gamers to get their favourite game and publisher to get free and great platform to publish.

3.2 System Functional Requirements

Everfino

➤ There are Five Roles in Our System

- Admin
- Manager
- Chef
- Receptionist
- End-User

1) Admin

R.1 : Manages Manager's Information

Admin can verify, update and delete the manager of the Restaurant.

R1.1 Verify Manager

Input : Add manager details

Output: Manager will be verified

R1.2 Update Manager

Input: Add manager details

Output: Manager Updated

R1.3 Delete Manager

Input: Add manager details

Output: Manager Deleted

R.2 : Manages End-User Information

Admin can verify, update and delete the end-user.

R2.1 Verify End-User

Input: Enter End-User Information

Output: End-User Verified

R2.2 Update End-User

Input: Enter End-User Information

Output: End-User Updated

R2.3 Delete End-User

Input: Enter End-User Information

Output: End-User Deleted

R.3 : View Statistics

Admin can view the general statistics related to all restaurants

R3.3 Show Statistics

Output: General Statistics

2) **Manager**

R.1 : Manages Staff's Information

Manager can add, update and delete the staff of the restaurant(Chef and Receptionist).

R1.1 Add Staff

Input: Enter Staff Details

Output: Staff Added successfully

R1.2 Update Staff

Input: Enter Staff Details

Output: Staff Updated Successfully

R1.3 Delete Staff

Input: Enter Staff Details

Output: Staff Deleted Successfully

R.2 : Manages Bill's Information

Manager can view, update and delete the bills of the Restaurant.

R2.1 View Bill

Input: Enter Bill Id

Output: Display Bill

R2.2 Update Bill

Input: Enter Bill Id

Output: Bill Updated Successfully

R2.3 Delete Bill

Input: Enter Bill Id

Output: Bill Deleted Successfully

R.3 : View Statistic

Manager can view the statistics of the Restaurant.

R3.1 Show Statistics

Output: Statistics of the Restaurant.

R.4 : Manages Tables(Dining Table) Information

Manager can add, update and delete the table of the Restaurant.

R4.1 Add Tables

Input: Enter Table Information.

Output: Table added Successfully.

R4.2 Update Tables

Input: Enter New information

Output: Table Information Added Successfully.

R4.3 Delete Tables

Input: Enter Table Information

Output: Table Deleted Successfully.

3) **Chef**

R.1 : Manages Order's Information

The Chef will get the order request generated by the end user.

R1.1 Accept Order

Output: Order Accepted Successfully

R1.2 Reject Order

Output: Order Rejected Successfully.

4) **Receptionist**

R.1 : Manages Bill's Information

Receptionist can update and manage payment status at the Restaurant.

R1.1 Update Bill

Input: Enter Bill Information

Output: Bill Updated Successfully.

R1.2 Accept Payment

Input: Pay via Card or Cash

Output: Payment Done Successfully.

5) **End-User**

R.1 : Browse Restaurant

The End-User can browse the restaurant registered with everfino.

Output: List of all Restaurants

R.2 : Manages Own Orders

The customer can view, add, update and delete the order.

R2.1 View Order

Output: List of Orders

R2.2 Add Order

Input: Select Item

Output: Ordered Successfully

R2.3 Update Order

Input: Select Order

Output: Order Updated

R2.4 Delete Order

Input: Select Order

Output: Order Deleted Successfully.

R.3 : Make Bill Payment

The end-user can pay the bill through card or Upi.

R3.1 Pay Via Card

Input: Card Details

Output: Payment done Successfully.

R3.2 Pay Via Upi

Input: Enter Upi Id

Output: Payment done Successfully

R.4 : Manages Feedback

The End-User can Add, Update and delete their feedback for their order.

R4.1

Input: Enter Feedback

Output: Feedback Successfully

R.5 : Table Reservation

The End-user can Reserve table in advance using our system

Input: Select Table and Time

Output: Reservation Status

3.3 Other Non-Functional Requirements

Performance and scalability. How fast does the system return results. How much will this performance change with higher workloads.

Portability and compatibility. Which hardware, operating systems, browsers, and their versions does the software run on? Does it conflict with other applications and processes within these environments.

Reliability, availability, maintainability. How often does the system experience critical failures? and how much time is it available to users against downtimes.

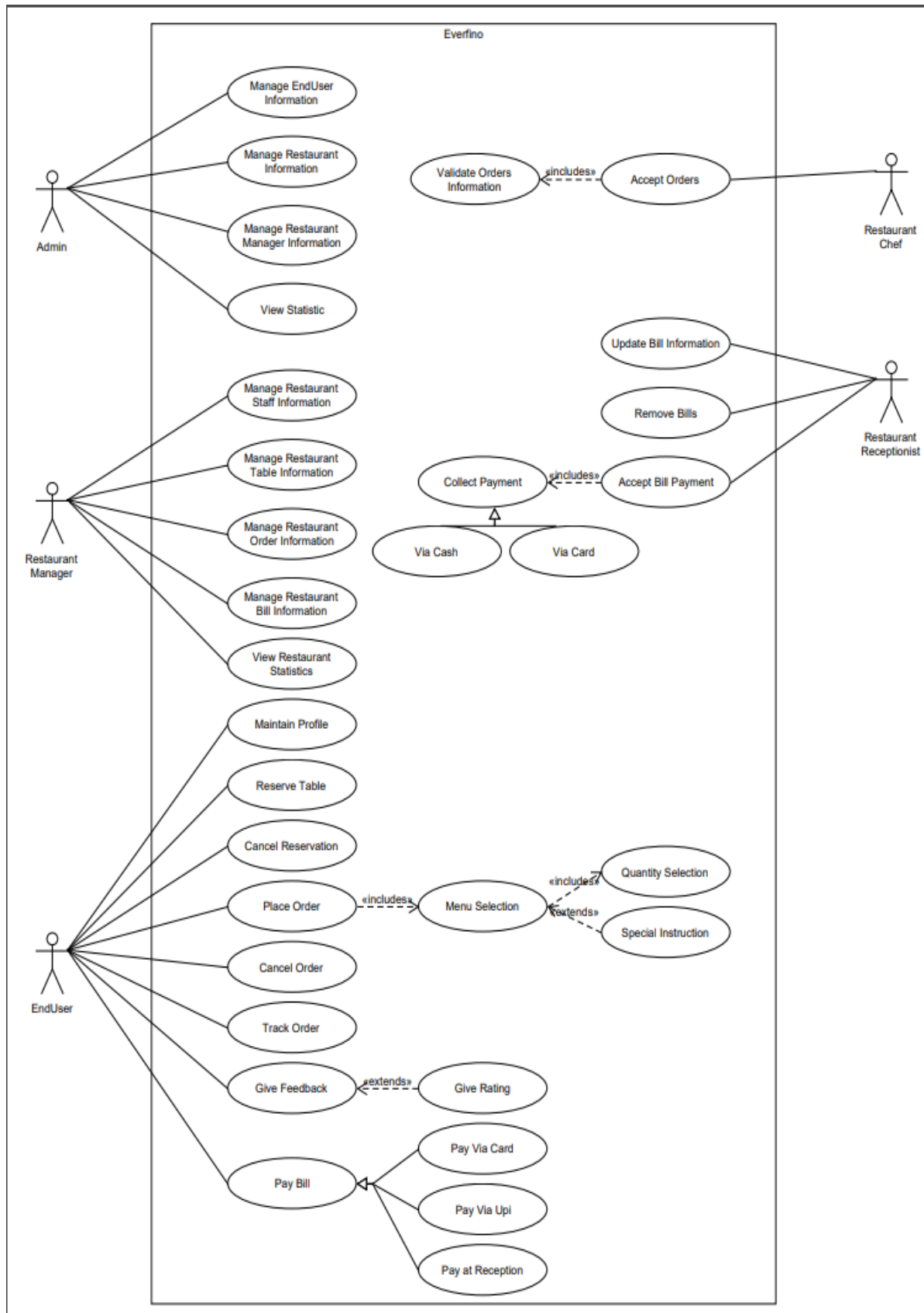
Security. How are the system and its data protected against attacks.

Localization. Does the system match local specifics.

Usability. How easy is it for a customer to use the system.

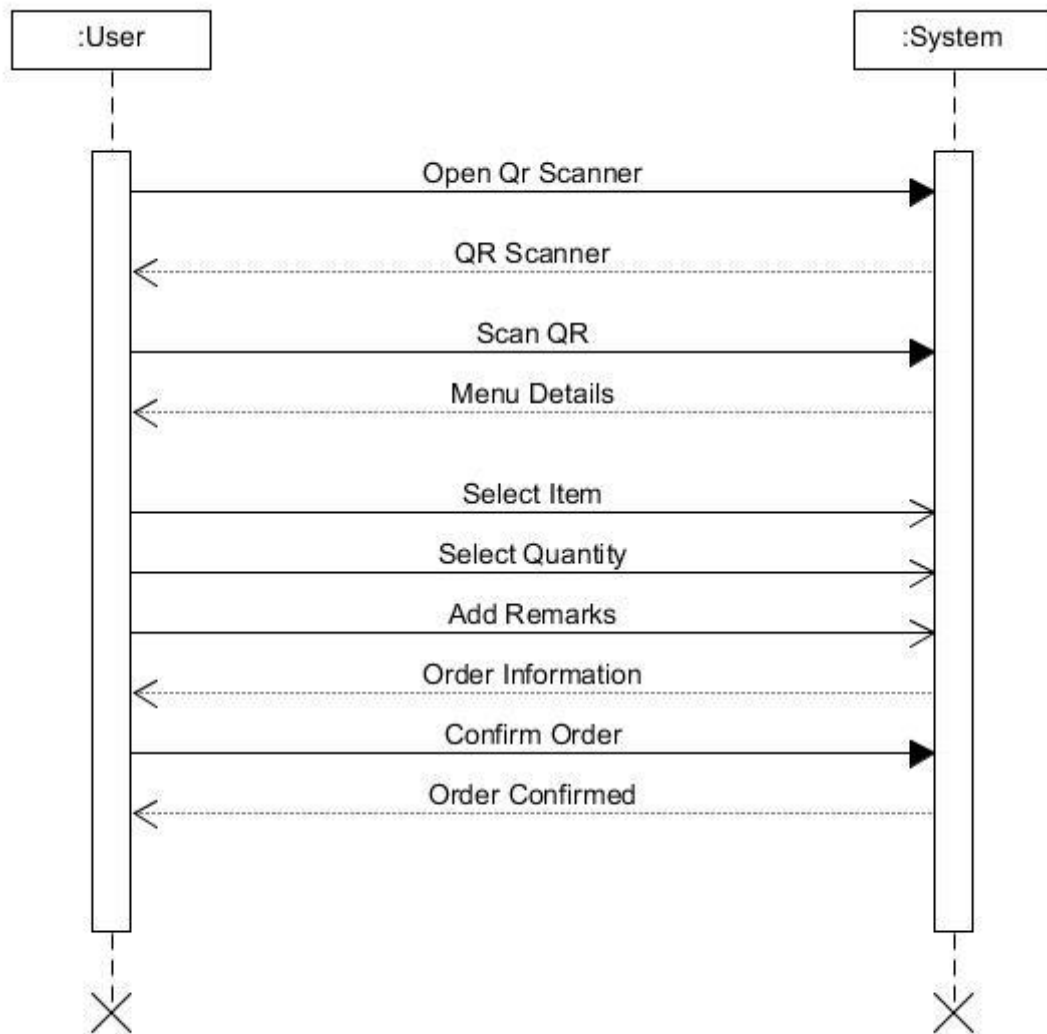
4. Design

4.1 Use Case Diagram

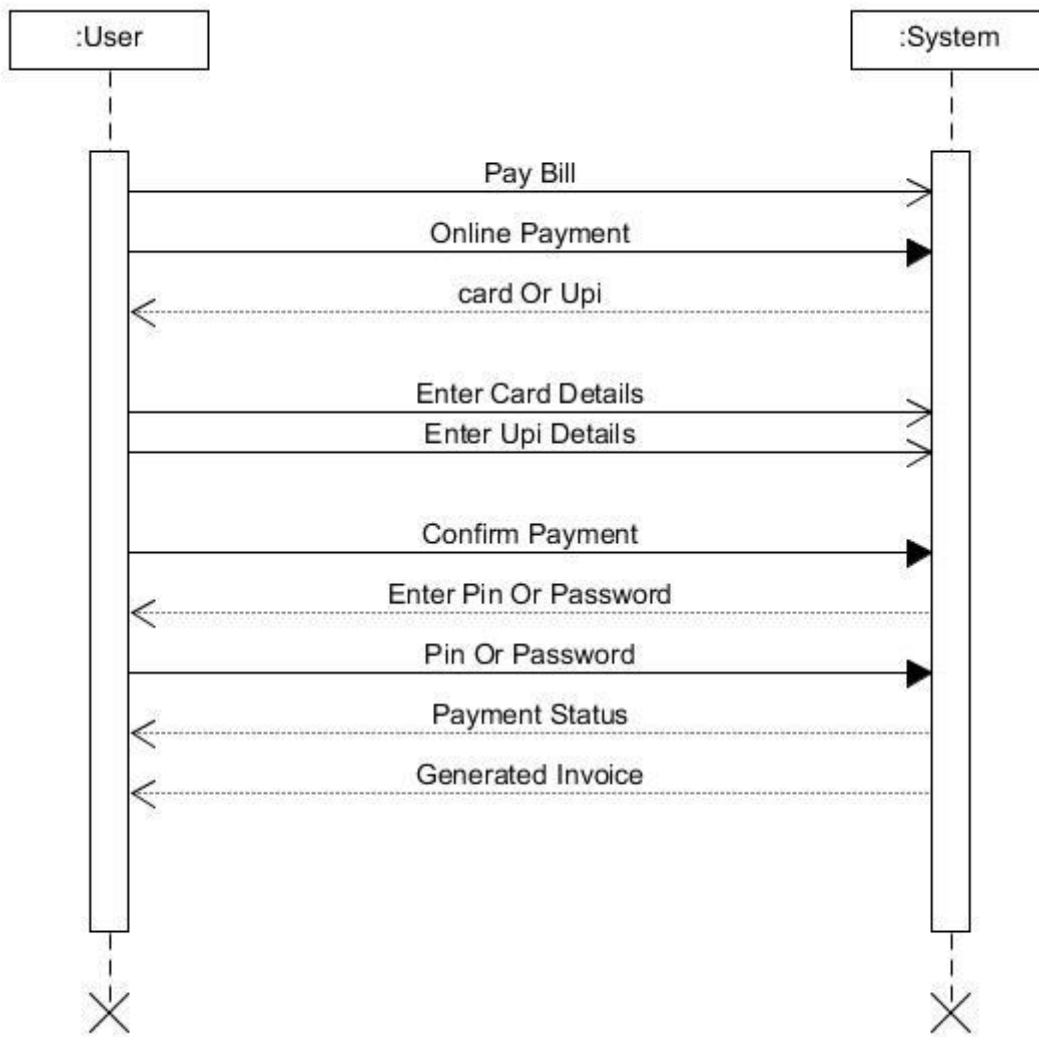


4.2 Sequence Diagram

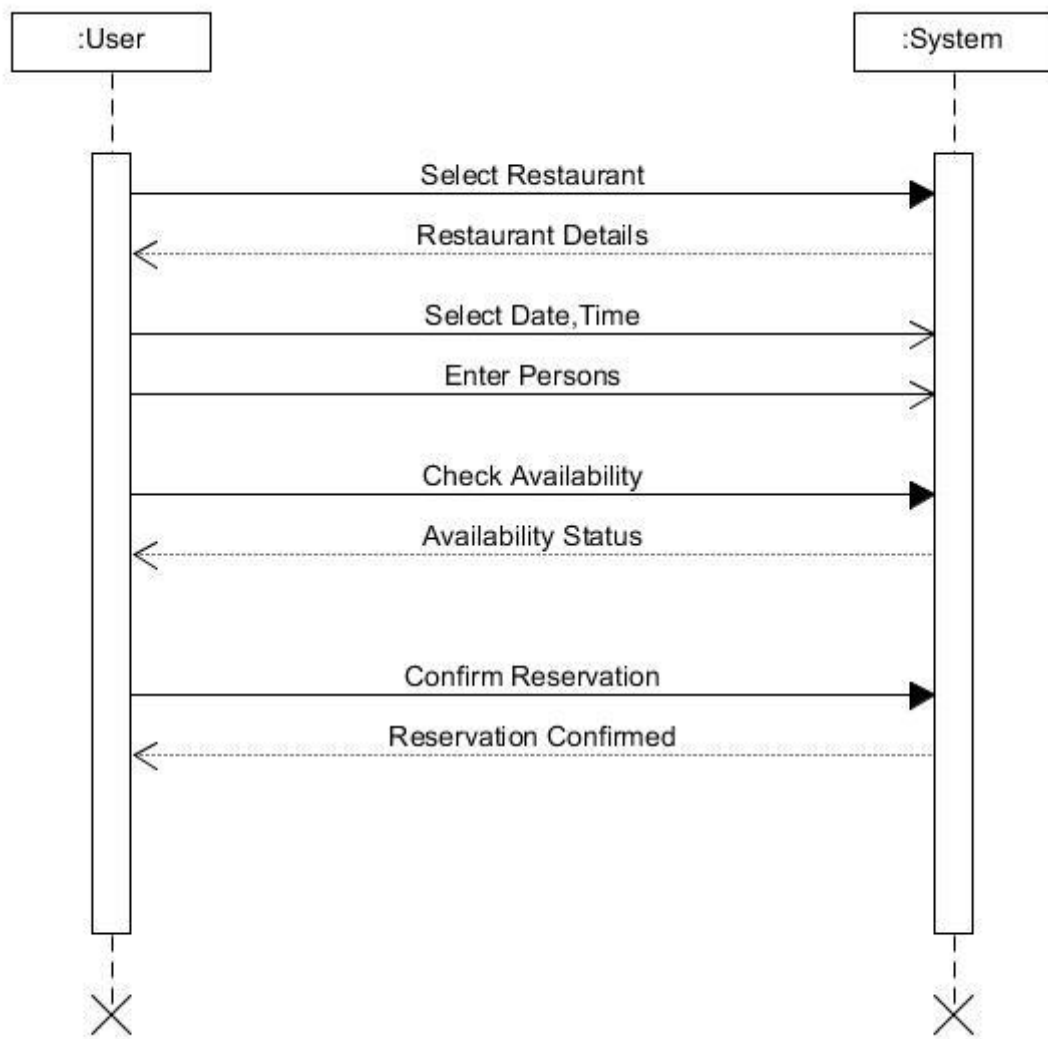
1. Place Order



2. Online Payment

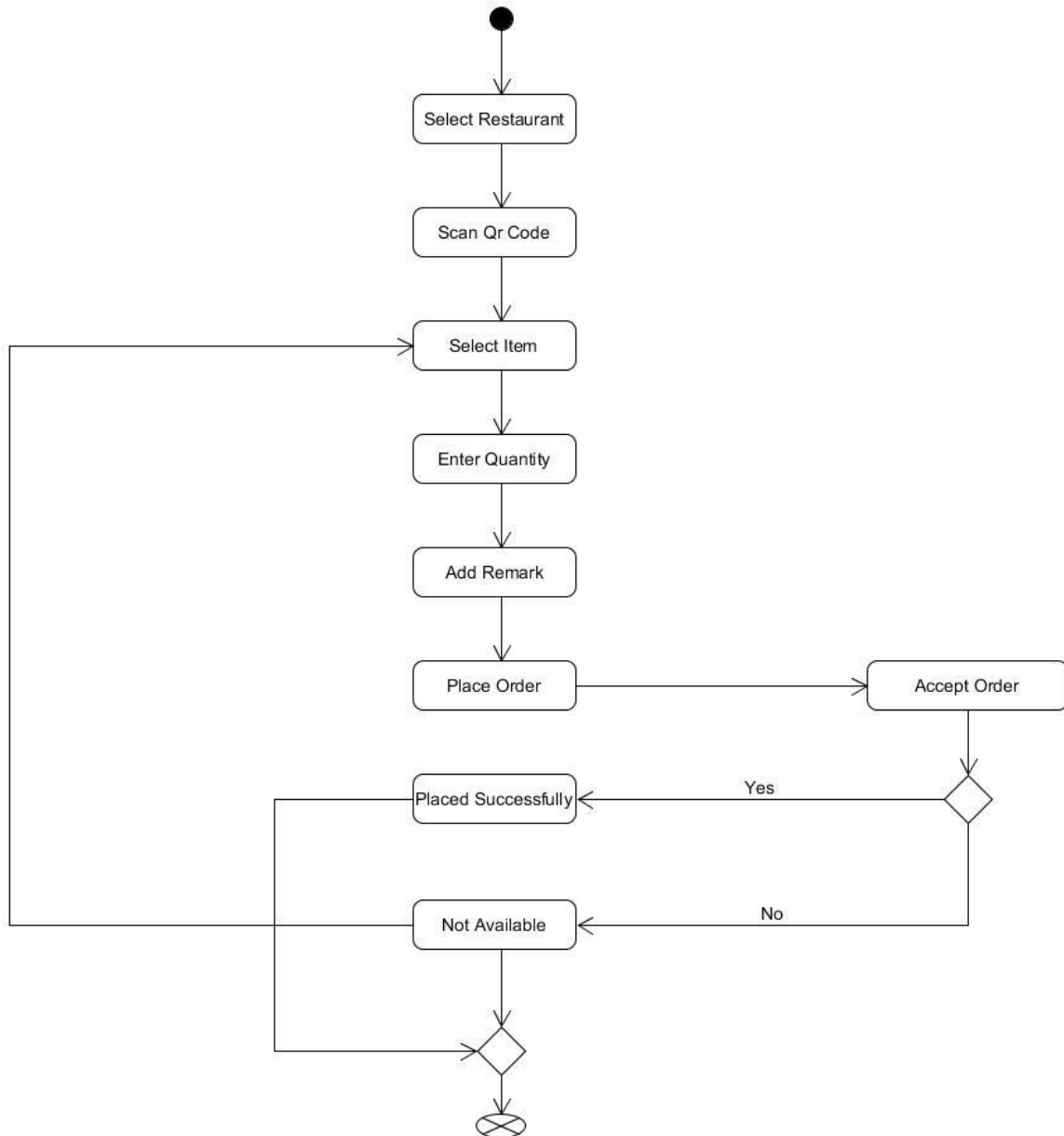


3. Table Reservation

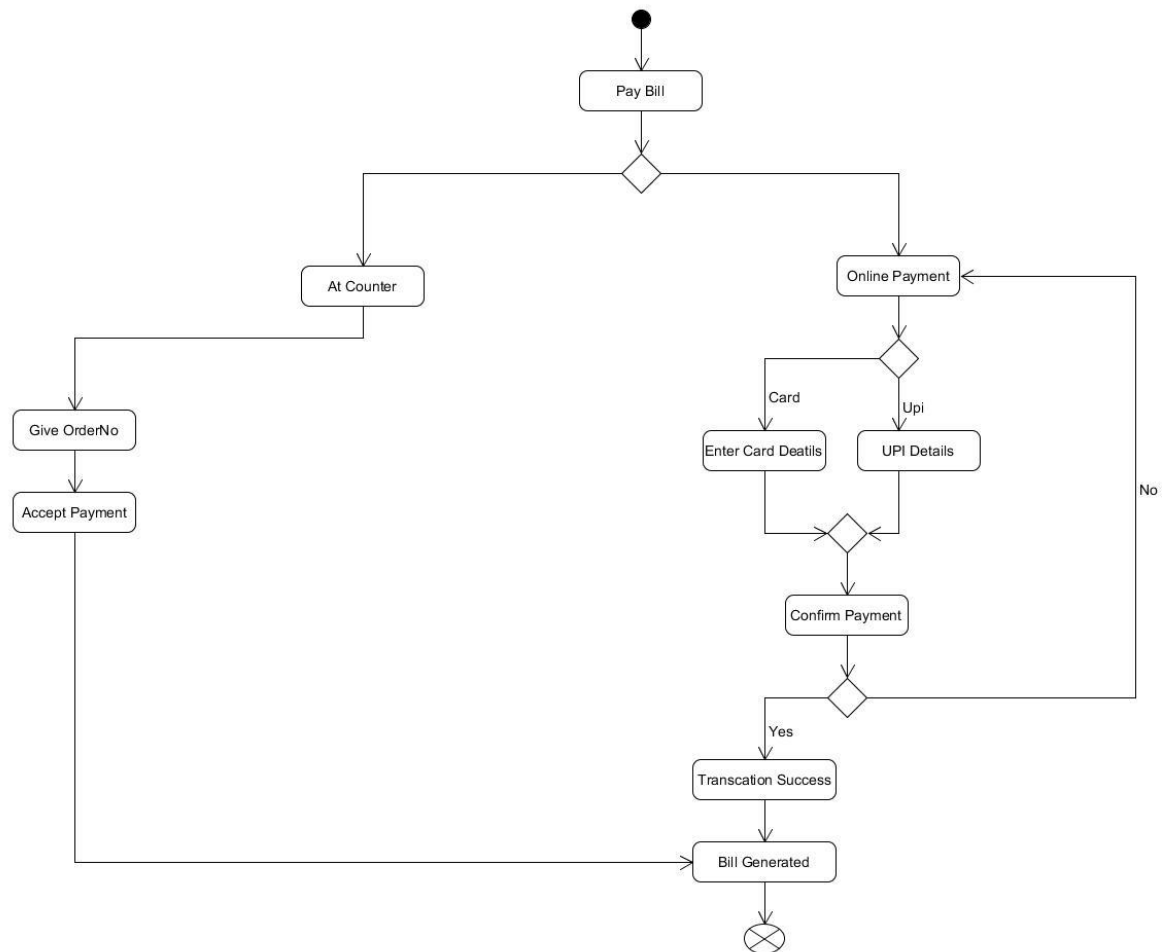


4.3 Actitivity Diagram

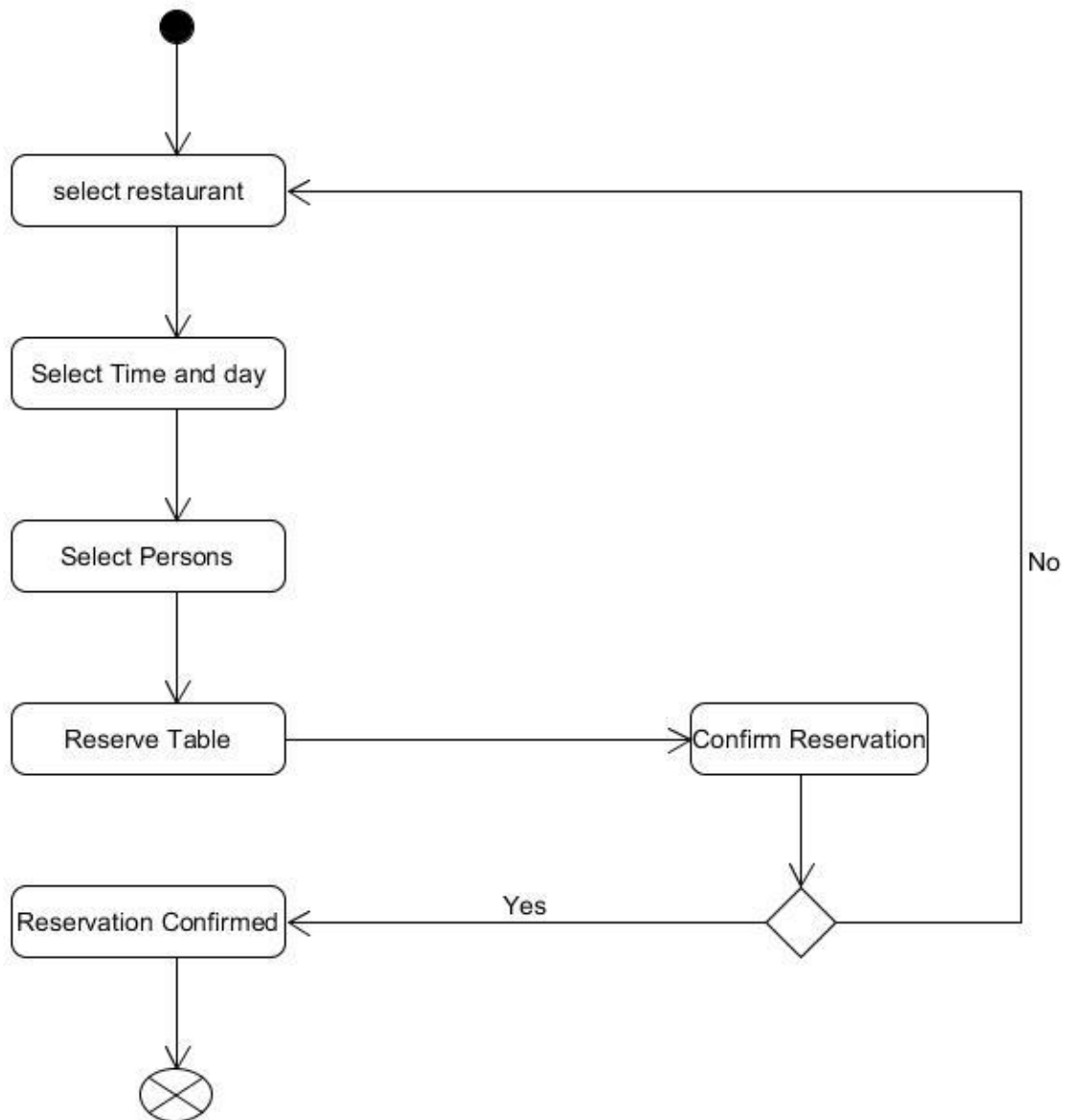
1. Place Order Activity



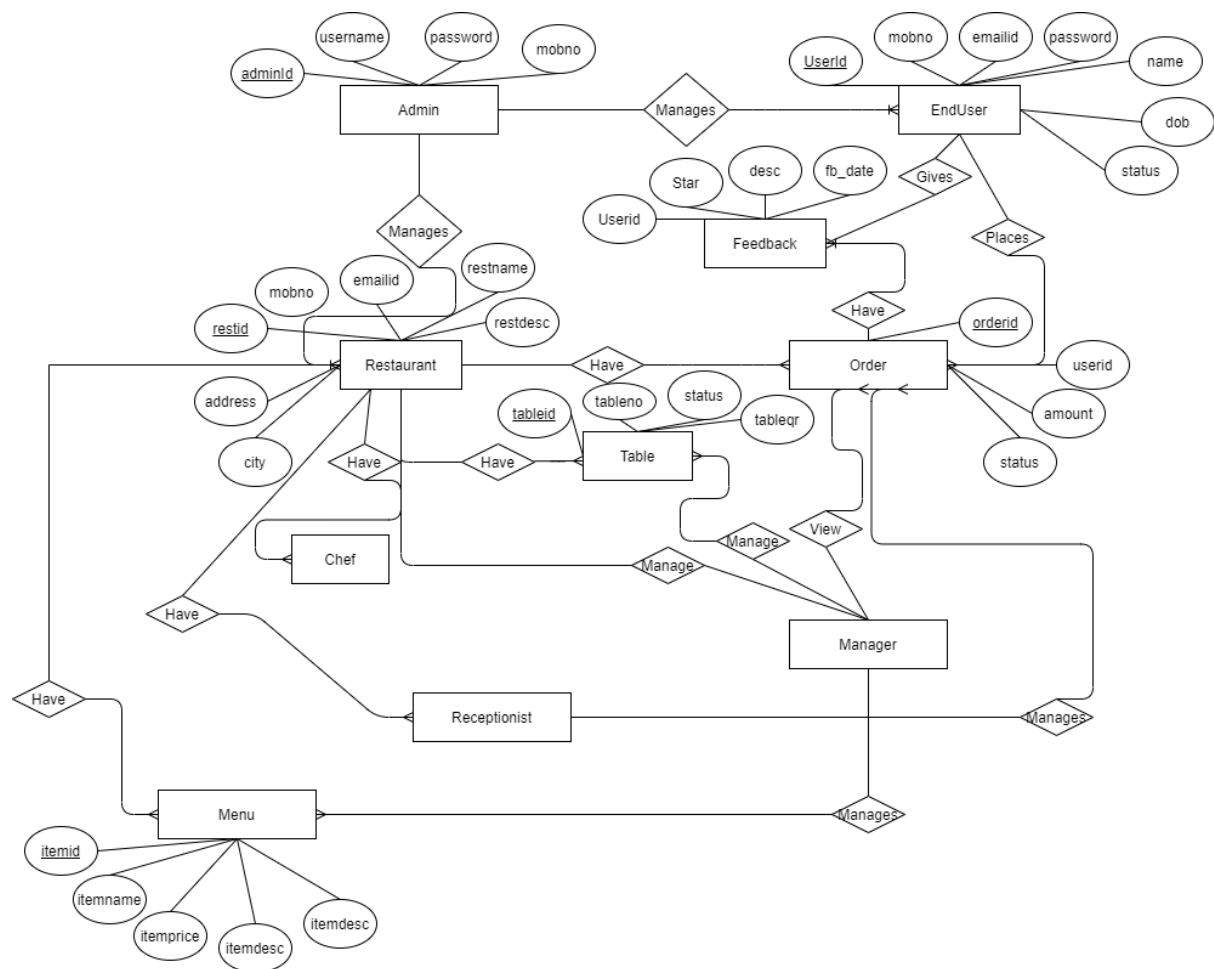
4 Accept Bill Activity



5 Reserve Table Activity



4.4 ER Diagram



4.5 Data Dictionary

1) Admin :

Name	Datatype	Size	Extra
adminid	int	6	Primary key
username	varchar	20	
password	varchar	20	
mobno	int	10	

2) Rest :

Name	Datatype	Size	Extra
restid	int	6	Primary key
mobno	int	10	
emailid	varchar	40	
restname	varchar	20	
restdesc	varchar	50	
address	varchar	100	
city	varchar	20	

3) RestUser[restid] :

Name	Datatype	Size	Extra
userid	int	6	Primary key
password	varchar	20	
mobno	int	10	
emailid	varchar	40	
role	varchar	10	Values(Manager, Receptionist, Chef)

4) EndUser :

Name	Datatype	Size	Extra
userid	int	6	Primary key
mobno	int	10	
emailid	varchar	40	
password	varchar	20	
name	varchar	20	
dob	date		
status	varchar		Values(Activated, Deactivated)

5) Menu[restid] :

Name	Datatype	Size	Extra
itemid	int	6	Primary key
itemname	varchar	30	
itemprice	int	6	
itemdesc	varchar	100	
itemtype	varchar	20	

6) Dining Table[restid] :

Name	Datatype	Size	Extra
tableid	int	6	Primary key
tableno	int	2	
status	varchar	10	Values(Reserved, Available, Not Available)
tableqr	varchar	10	

7) Orders[restid] :

Name	Datatype	Size	Extra
orderid	int	6	Primary key
userid	int	6	Ref(EndUser)
amount	int	6	
status	varchar	10	Value(Paid, Unpaid)
order_date	date		

8) Order Item[restid]:

Name	Datatype	Size	Extra
orderid	int	6	Ref(Orders)
itemid	int	6	Ref(Menu)
quantity	int	2	

9) EndUserOrders[userid] :

Name	Datatype	Size	Extra
orderid	int	6	Ref(Orders)
itemid	int	6	Ref(Menu)
quantity	int	2	
restid	int	6	Ref(Rest)

10) Reservation[restid] :

Name	Datatype	Size	Extra
reserveid	int	6	
tableid	int	6	Ref(Dining Table)
userid	int	6	Ref(EndUser)
reserve_date	date		

11) LiveOrders[restid] :

Name	Datatype	Size	Extra
liveid	int	6	
orderid	int	6	Ref(Orders)
tableid	int	6	Ref(Dining Table)
itemid	int	6	Ref(Menu)
userid	int	6	Ref(EndUser)
quantity	int	2	
status	varchar	10	
order_date	date		

12) Feedback[restid] :

Name	Datatype	Size	Extra
userid	int	6	Ref(EndUser)
star	int	1	
desc	varchar	100	
fb_date	date		

5. Implementation Details

5.1 Modules Description

1. Admin

In this module we are provide functionality like manage Restaurant, manage End User and also display statistic of all over system.

2. Restaurant Manager

In this module we are provide functionality like Manage Menu, Staff like Chef, Receptionist and also display statistic of particular Restaurant.

3. Chef

In this module we are provide functionality like Manage Live Order of customer and able to change the status of order.

4. Receptionist

In this module we are provide functionality like manage all the bills, old record and accept bill payment from End User.

5. End User

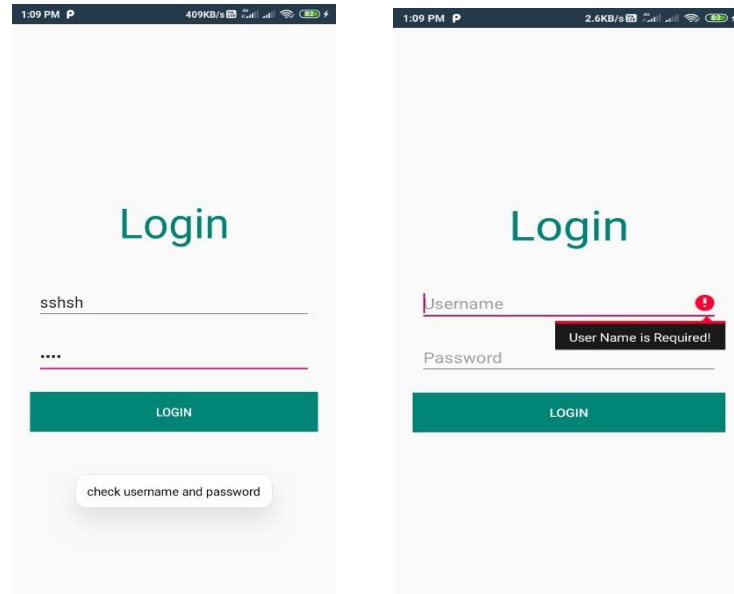
In this module End User can Go to any Restaurant and order any food from menu and get the live status of order by only scanning the QR code on the table.

6. Testing

Testing

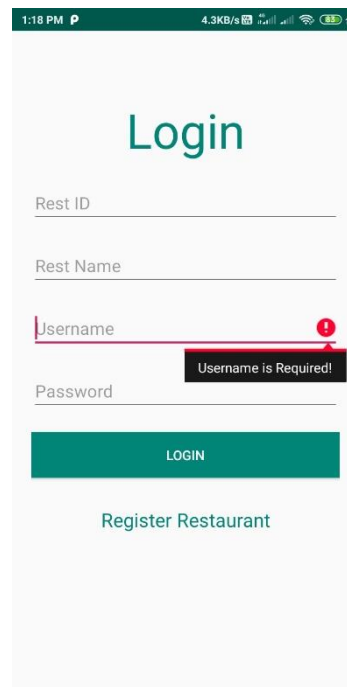
We are using black box testing for test out project.

1. Login Page of Admin



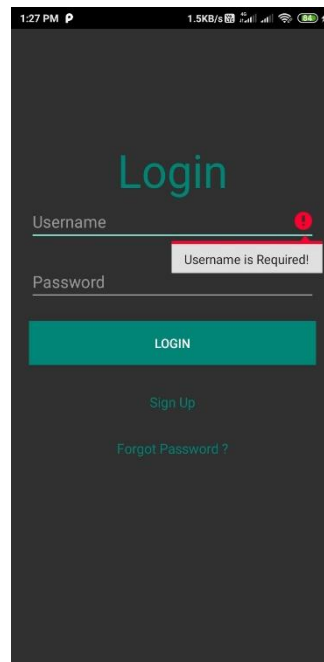
Without enter any detail if admin wants to login than admin to able to login system provide error message that fields are required. And if Admin enter wrong details than also system provide message for check password and username.

2. Login page for Restaurant User



Without providing correct credential no one can login to the restaurant app.

3. User Login page



If user need to login to the system user need to enter all details and correct credentials than only system allow user to login.

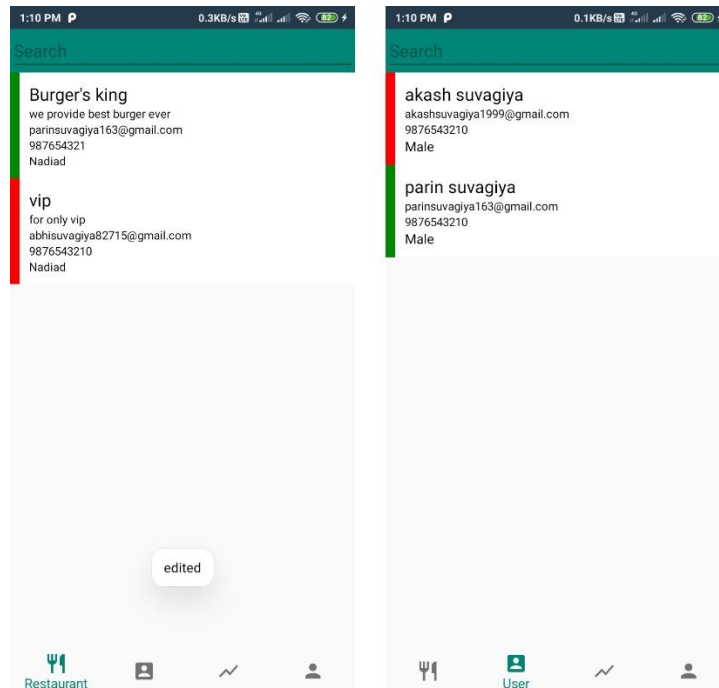
4. Required Field Error

We provide user friendly error in our all apps whenever any user forgot to enter some details than system provide error message to the user. For reference you can see above snapshots.

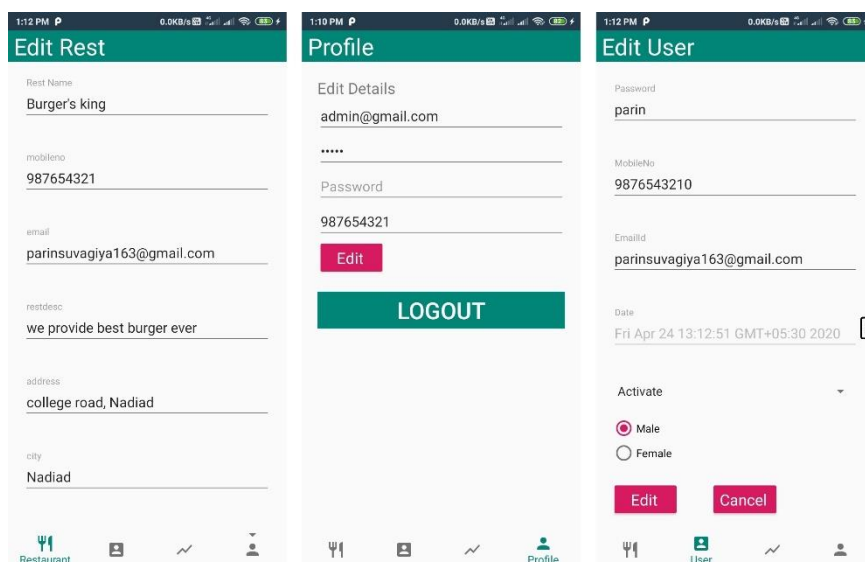
7. Screenshots

1. Admin

Admin can manage the user and restaurant and admin can change it's own profile.



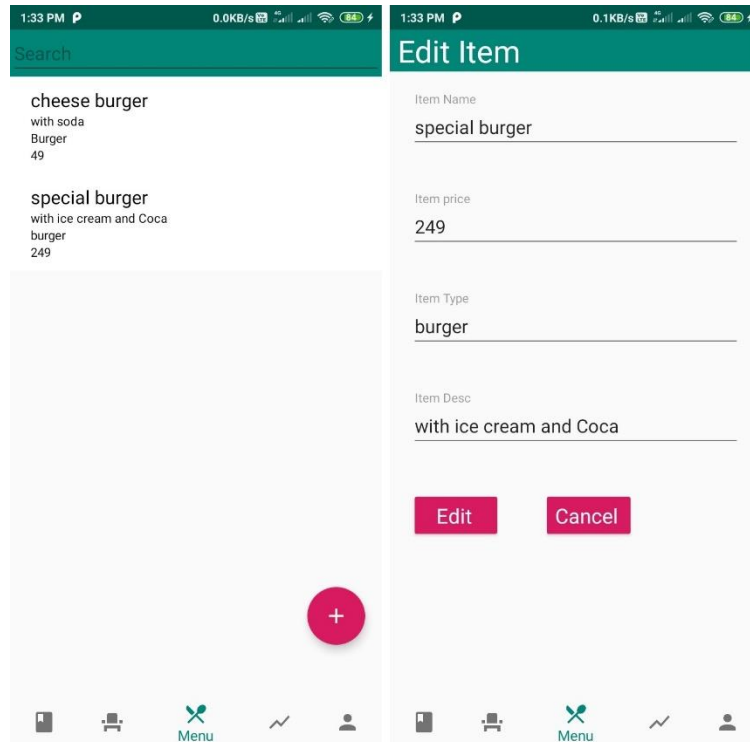
Here we provide snapshot of restaurant and User view . The restaurant and user have green left border they all are Active and which have red left border they all are Deactivate.



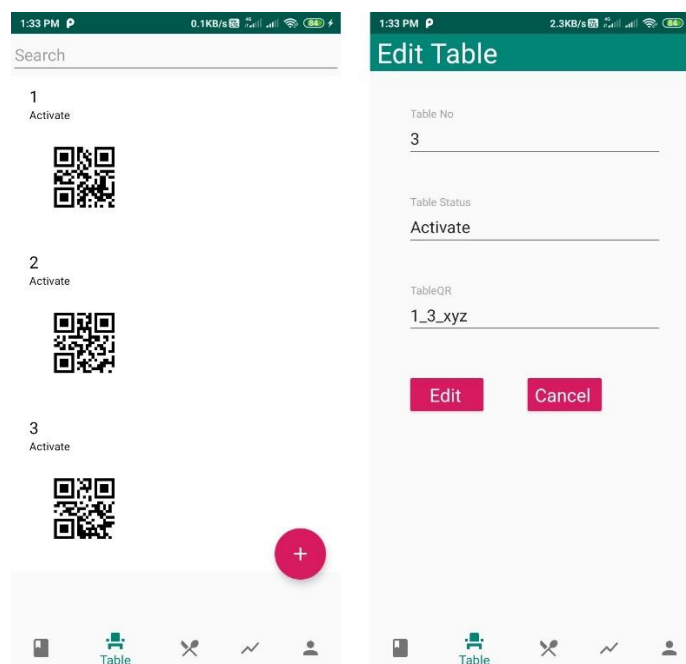
Here in above snapshot you can see the view of edit restaurant ,edit user and admin's profile edit screens.

2. Restaurant Manager

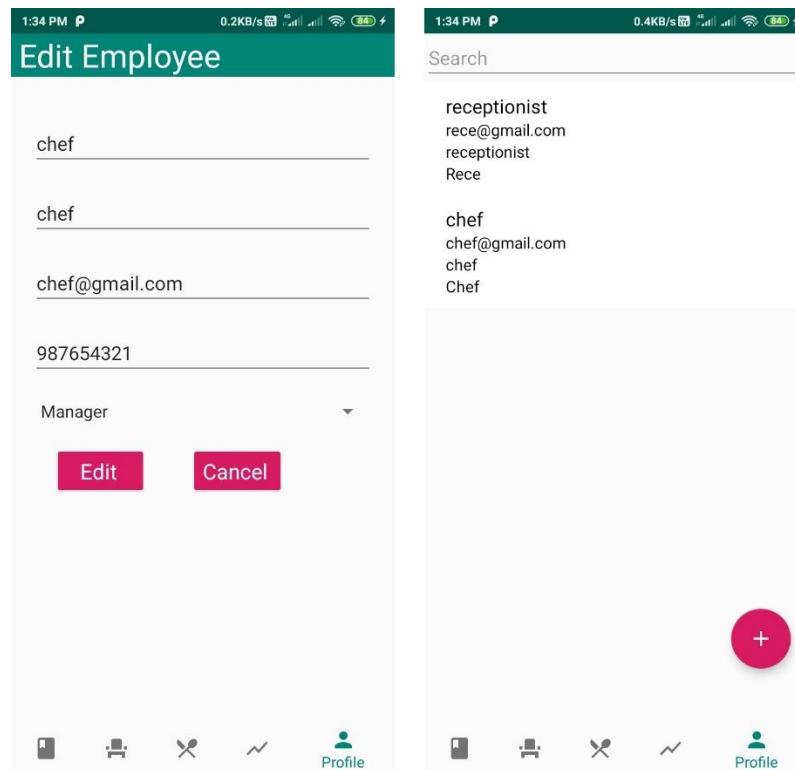
Restaurant Manager can manage Menu of restaurant, dining table in restaurant and manage staff of the restaurant. And also view order and live order of the restaurant.



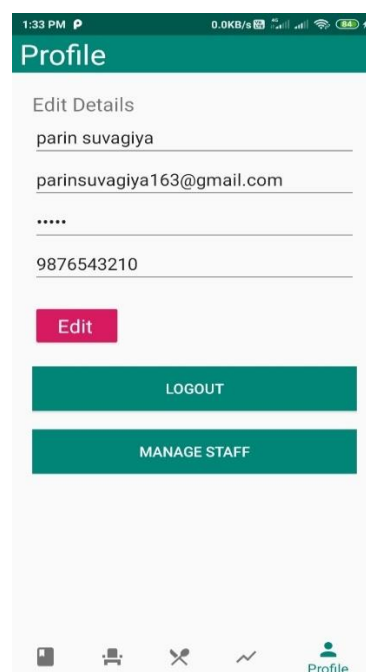
Here you can see the snapshot of manage Menu in restaurant in snapshot we have edit and list of available menu and also one floating button to add new Menu item in Menu of restaurant. In same way manager can manage dining table also.



Manager can also manage the staff of the restaurant and add, modify and remove employees from restaurant.

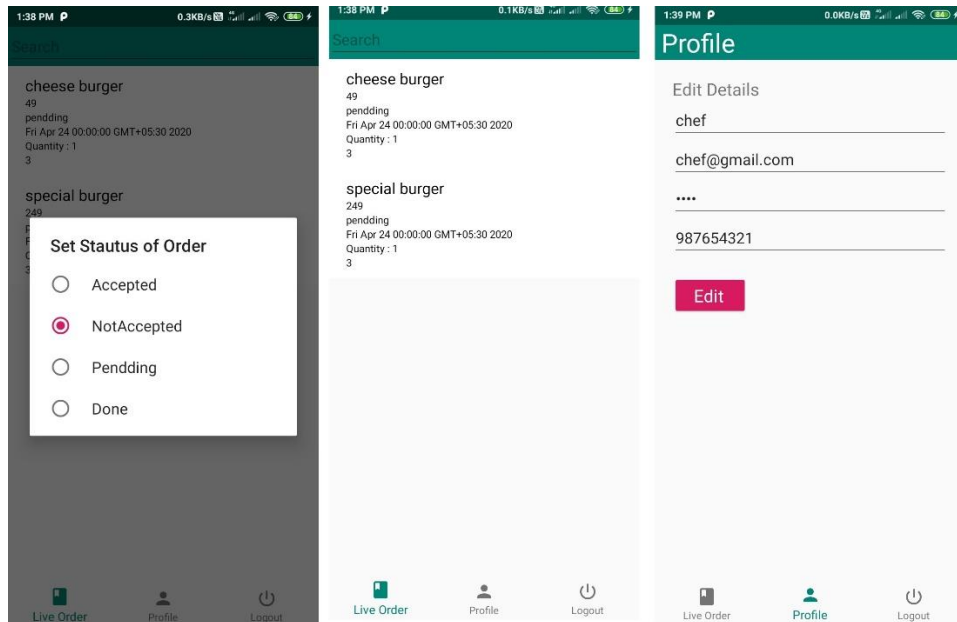


Here in above snapshot you can see list of all available staff and by clicking any staff member you can able to edit details of staff member. And also we have floating button to add new staff member. And manager can also edit it's own profile.



3. Chef

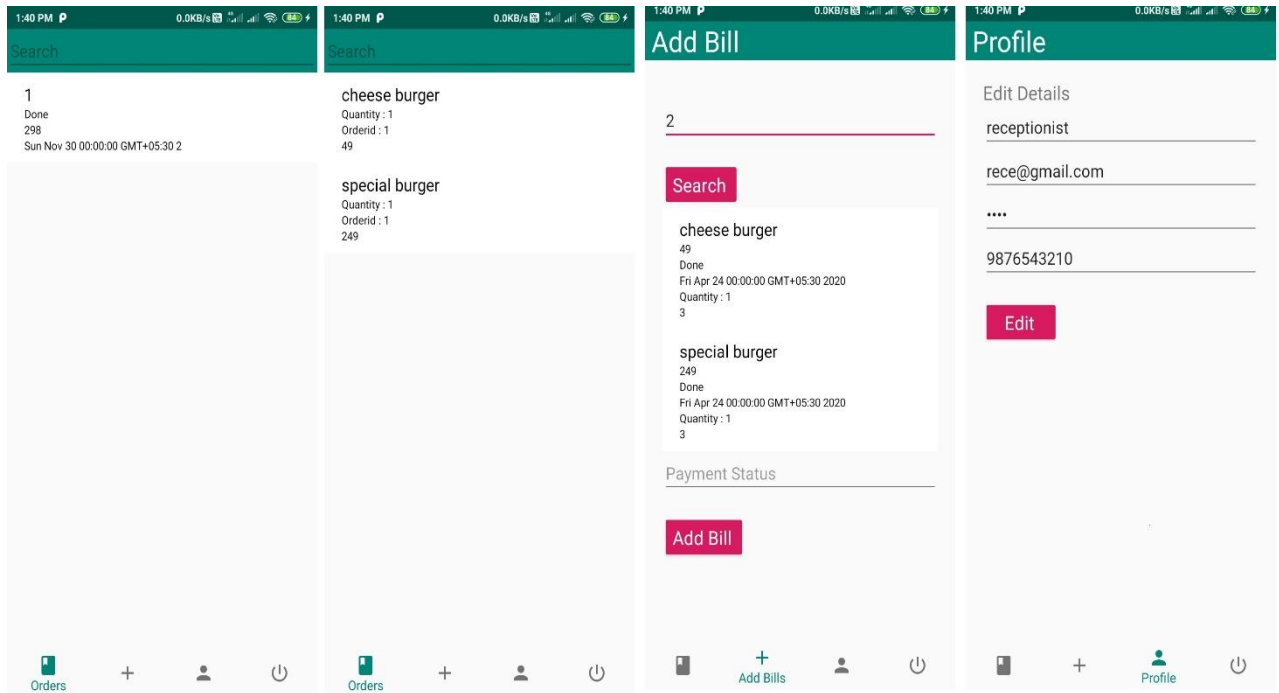
Chef is staff member of restaurant it can work under the manager. Chef can accept the order of the restaurant given by user and also manage it's own profile.



Here above snapshot is of chef accepting order and edit profile.

4. Receptionist

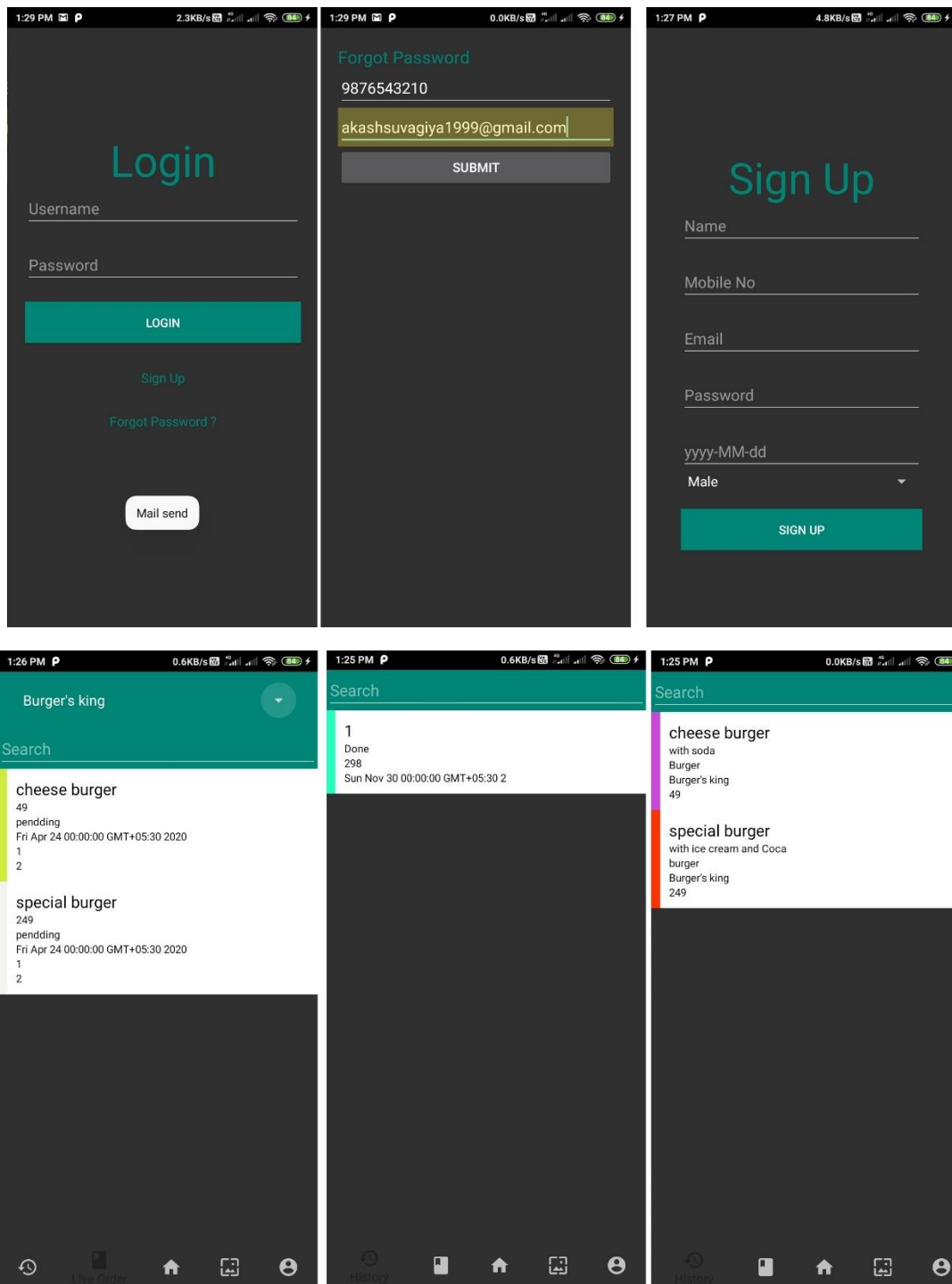
The receptionist is member of restaurant staff just like as chef. It can main bills of restaurant via our app and also edit It's own profile. And also view all order of the restaurant.



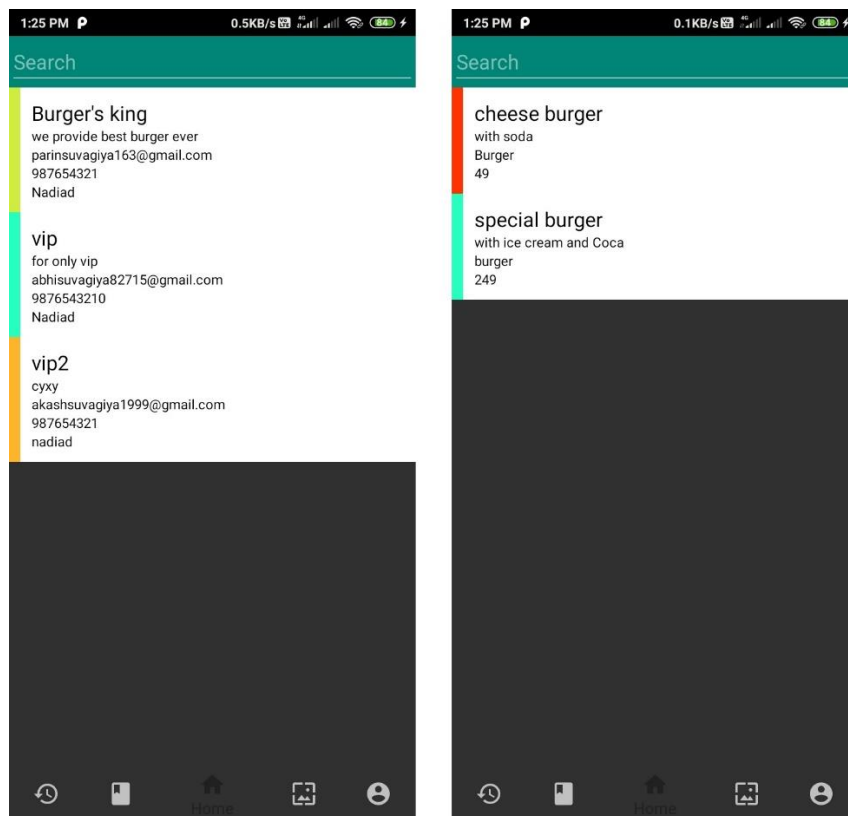
Here you can see above snapshot for bill and bill details. Add bill screen and edit profile screen of receptionist.

5. Enduser

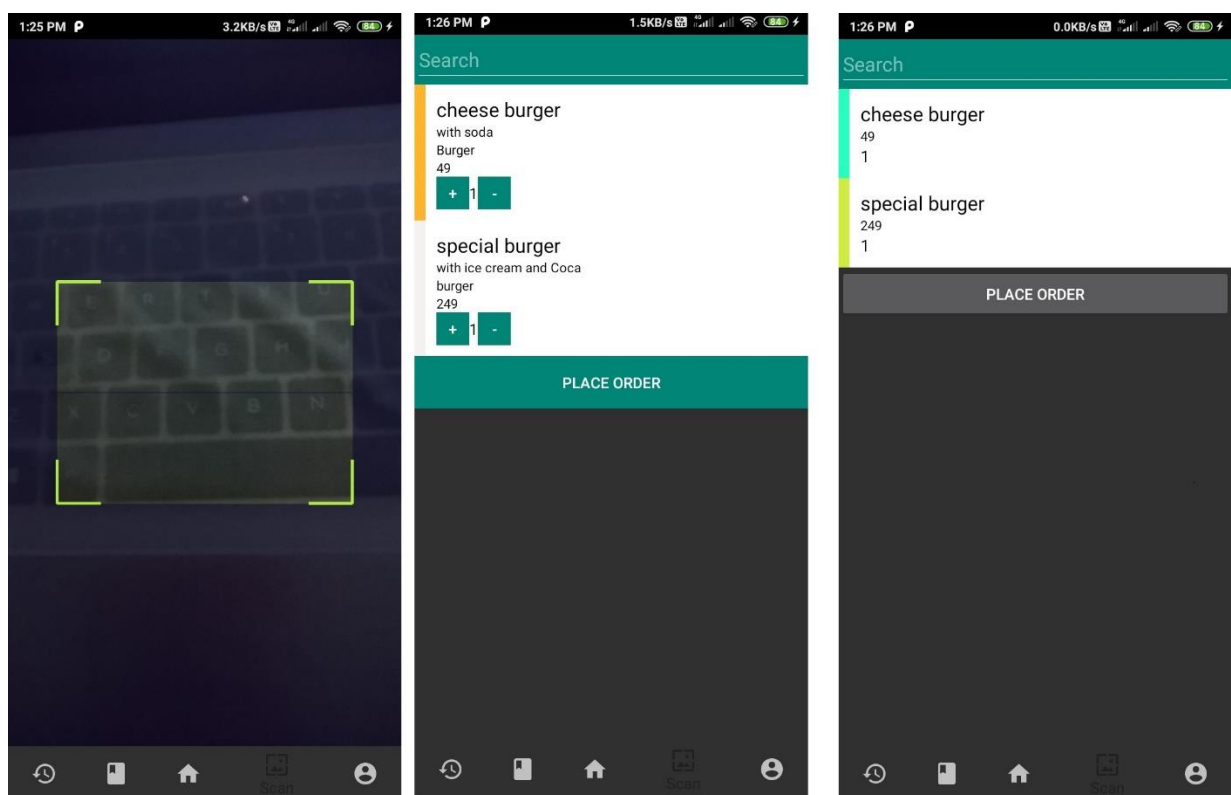
Enduser can register by it self and used our app. For add user we provide facility to forgot password in which we used mail system.



And enduser can view the all order which is placed earlier. And also see current live order and it's status.

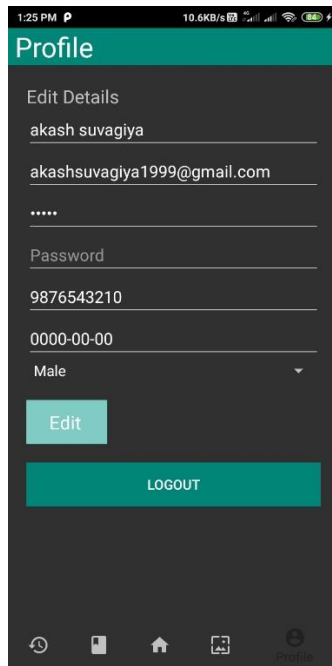


Enduser can also view the restaurant and their menu.



For enduser we are provide facility of scanning qr code. By scanning qr code user directly able to see restaurant menu and user can select item from menu and place order. System is also provide order preview for confirmation.

At the end enduser can also maintain is it's profile.



Here we also have one splash screen whenever any system user start app this splash screen is visible for a while.



8. Conclusion

Conclusion

The Functionality implemented in the system was done after understanding all the system modules according to the requirements.

Functionalities that are successfully implemented are:-

Login

Open Particular Restaurant on QR scan

Order Food item

Accept Bill

Manager, Chef and Receptionist Application

After the implementation and coding of system, Comprehensive testing was performed on the system to determine the loopholes and possible flaws in the system.

9. Limitations and Future Extension

Limitations and Future Extension

Limitations:-

Internet Connectivity is required

Admin is required to handle the app

Admin should have basic knowledge of handling app

Future Extensions:-

Payment Gateway need to implement

Forget Password using mobile no or email

Restaurant feedback

10. Bibliography

Bibliography

Web:-

<https://stackoverflow.com>

<https://www.c-sharpcorner.com>

https://en.wikipedia.org/wiki/Windows_Forms

<https://docs.microsoft.com/en-us/dotnet/framework/wcf/whats-wcf>