

# **CSE 5324-003: SOFTWARE ENGINEERING: ANALYSIS, DESIGN, AND TESTING**

## **ONLINE FOOD DELIVERY APPLICATION**

### **Team 4:**

Shubash Muniyappa (1001915563)  
Shreethika Sivakumaran Sathyama (1002042661)  
Bhargava Manikanta Tummalapenta (1001965491)  
Radha Yashwanth Uppuganti (1001964009)

# Project Description

Food ordering System is an application which will help restaurants to optimize and have a good control over the whereabouts of their restaurants. From a managerial perspective, the manager will be able to oversee the restaurants by having access to all reports and being able to check the records of each employee's orders.

The major objective is to keep the restaurant running properly and effectively while minimizing the need for manual entries. Food orders may be tracked more readily using the system's daily data.

One of the most comprehensive and user-friendly apps is the Food Delivery app, which allows users to look up nearby eateries and cafés, place online meal orders, and have their food quickly delivered to their home. As it offers menus, reviews, and ratings, you can also find factual information about restaurants on this website. Users can use that information to place orders and enjoy delectable food at home.

Our app's primary function is to recommend nearby eateries to users and take their orders. Based on ratings and reviews posted by previous customers, users can place orders from their preferred restaurants. It provides people with comfort and aids entrepreneurs in expanding their small-town food industry.

Our food delivery program will assist the users by providing them with food right where they are at during these kind of unforeseen crises. Recently, we experienced a significant outbreak known as Covid, which was unexpected. Instead of going out and hunting for restaurants, they would like to stay at home and place orders for whatever they desire. This is beneficial for both those who are consuming the food and those who are wanting to earn by registering and delivering it.

By offering opportunities to workers who are looking for opportunities, this will also create job opportunities for many people in both technical and non-technical ways.

# Overall Design Approach

The methodology which we are going to adapt ourselves in developing the application is “Agile Methodology” with scrum based. By adapting the agile methodology, we are going to break the project by multiple phases, and we also have a constant collaboration with stakeholders about the application we are developing for every enhancement we are developing. There is also an implementation of the cycle process in our project like process planning, executing, and evaluating.

An agile software development process always starts by defining the users for a particular product and documenting a vision statement for the scope of problems, opportunities, and values to be addressed. Similar way we also implemented the process of adapting ourselves in those kinds of roles developing in our process of adapting agile methodology.

## **Users:**

According to the agile process we always have a end user whom is our perspective. So the application “Food delivery “is also based on the user perspective.

## **Product Owner:**

The product owner is tasked to be the voice of the customer, including any internal stakeholders. By adapting that we also developed our application based upon that.

## **Software Development Team:**

According to the agile methodology we adapted there would be a software development team. So, in our way of developing an application we have a software development team which is further divided into sub teams

**Software Developer** – Who looks on the development of the application

**Business Analyst** – who develop user stories in from the product owner for the application development

**Software Tester** – Software tester who is responsible for verifying and validating the quality of the application.

**Scrum Master** – who is responsible for conducting the Scrum calls to know the daily update, who also conduct backlog grooming and also responsible for improving the agile team’s velocity.

In the application development process in agile methodology, we adapted the “Scrum” based methodology which organizes the work in cadences called *sprints*,

usually lasting one or two weeks. The product owner writes the requirements as user stories, then prioritizes them in a backlog based on their business value. The team reviews the backlog and commits to the top user stories they can complete during the sprint.

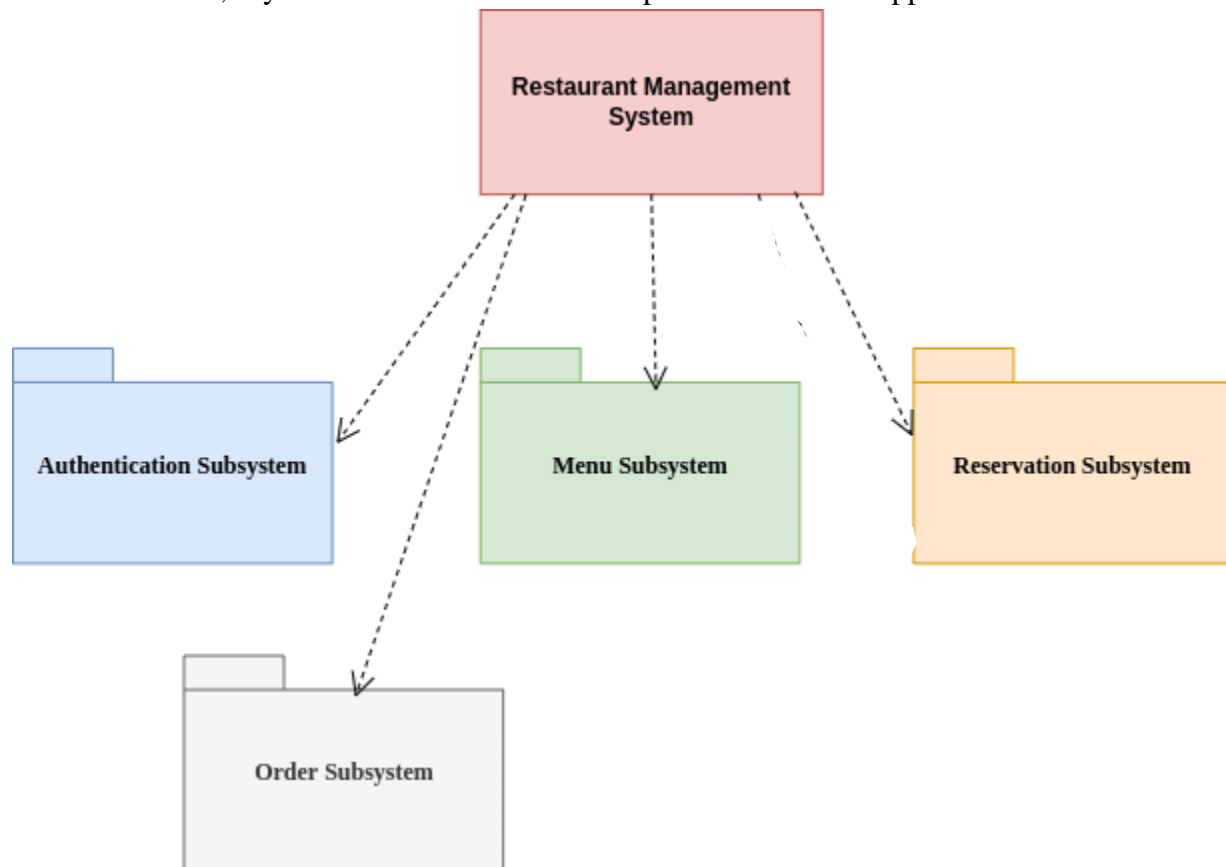
So according to the scrum methodology of agile we have following agile ceremonies implemented in our application

Sprint Planning, Daily Stand up, Sprint Reviews and Retrospective meetings .

# System Design

The simulation starts with the customer entering his/her credentials (name, ID and password). After verification, the customer can place the order by specifying the amount of food required. Now we get a window displaying the order number, customer ID, food name, price and quantity. Once the customer completes his/her order, they are redirected to a payment window that shows the total price, the customer can select the payment method of their choice, and then the customer receives a message of order confirmation.

Now, if you are an administrator, you can select the login option and enter Administrator Credentials (email ID and password). Once you enter you can manage the portal; where you can choose to add food, delete food or refund food. Any option you select will lead you to the food menu. Once the selected action has been performed, the end result, i.e. added foods or updated food lists shows, if you delete a certain food that particular food disappears from the main menu.



The Above figure is the System design diagram with Sub-systems.

## Function 1: Search food categories

Subsystem 1: Pop the search bar

Subsystem 2: Search for food item

Subsystem 3: Display related food or display not available (if no food Item found)

## Function 2: Add food item to cart

Subsystem 1: View full details of the food item like price, picture etc.

Subsystem 2: Add the preferred food item to the cart.

**Function 3: Cart items:**

Subsystem 1: Navigate to cart.

Subsystem 2: View items in cart.

Subsystem 2: Make changes to cart.

**Function 4: Place order:**

Subsystem 1: Navigate to order screen from cart.

Subsystem 2: Enter/Edit shipping address.

Subsystem 2: Confirm to order.

**Function 5: Order Id**

Subsystem 1: Display Order confirmation ID to admin.

Subsystem 2: Display Order confirmation ID to customer.

**Function 6: Bill**

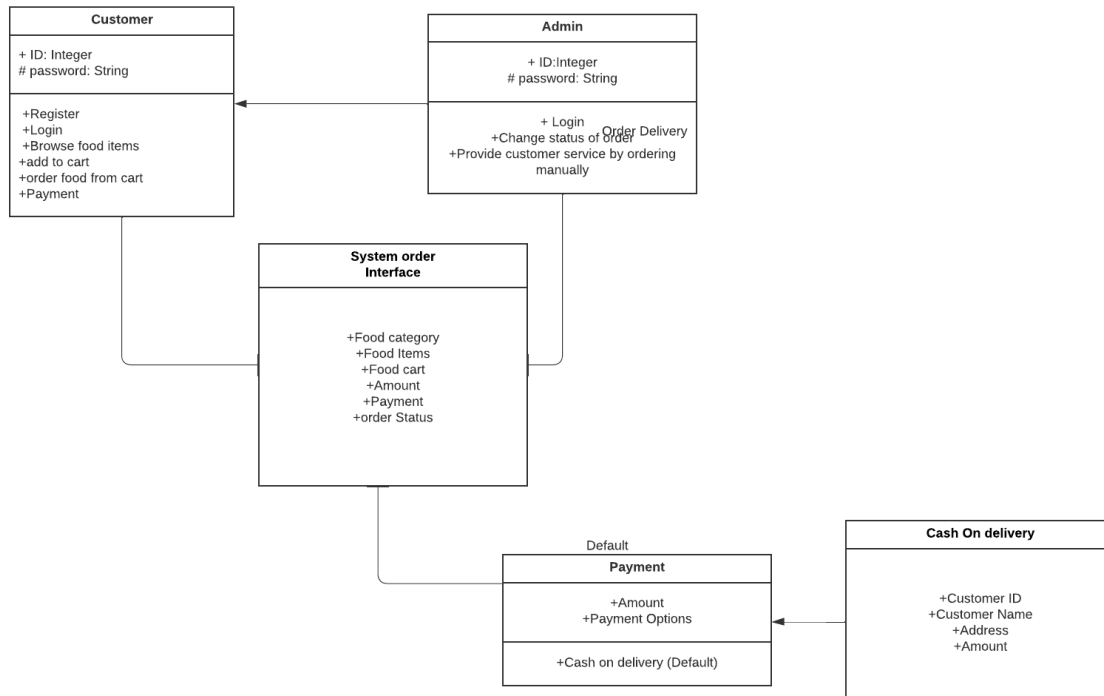
Subsystem 1: Print Order ID, Price, food Item to admin.

Subsystem 2: Print Order ID, Price, food Item to customer.

**Function 7: Delivery address**

Subsystem 1: Edit/Enter the delivery address while ordering or independently in home screen.

# Class Diagram



# PROJECT REQUIREMENTS

## FUNCTIONAL REQUIREMENTS :

Requirement ID	Requirement Statement
R1	The application shall provide a form for registering the user with fields such as name, Email ID, password, confirm password, address.
R2	The application shall allow the authenticated user to login as customer or admin.
R3	The application shall provide an authenticated user to logout.
R4	The application shall provide a reset password function for all registered users.
R5	The application shall provide a form for seller to add a food Item for sale with fields such as item name, description, price, picture.
R6	The application shall display food items list available for the customer to order.
R7	The application shall allow the admin to edit/update the post which is already created.
R8	The application shall notify both the admin and customer after the food item is purchased with details such as order confirmations, order details.
R9	The application shall provide a search bar for customers to search for a preferred food item with price.
R10	The application shall show a list of orders received on the date and allow him to calculate the expenditure and earnings.
R11	The application shall display all the customer details to the admin.



R12	The application shall allow customer to add items to cart.
R13	The application shall allow customer to view and make changes in the cart.
R14	The application shall allow customer to place order to restaurant with the selected items.
R15	The application shall provide the bill and store expenses it keep track.
R16	The application shall allow customer to track his order.
R17	The application shall provide customer to provide feedback on the restaurant.
R18	The application shall allow admin to make payment to staff
R19	The application shall allow admin to edit status of the order
R20	The application shall allow admin to schedule time for staff to work in shifts (Manage staff).

#### NON-FUNCTIONAL REQUIREMENTS :

- The System is designed to be **secure** by maintaing state properly
- The End product is an Android Application making it portable as 70% of global smartphones run on Android.
- The System is designed to have high perdomance and low responce time

## Actor System Interaction Model. - Food Ordering

	Actor: Customer	System
1	TUCBW ordering food from a restaurant application	2 It shows the restaurant options to place an order
3	Customer navigates thru the menu from a restaurant	4 System shows suggestions regarding the orders
5	Customers add the food into the cart	6 System estimates the total amount and displays it
7	Customer places an order	8 system displays the food tracking
9	TUCEW customer receives the food.	

In the above actor system interaction model, we can see how a customer can interact with the system and make a food order online.

- User can search in the application for a specific restaurant and food regarding his choice.
- System shows the promotions or offers for the person also shows the suggestions for the customers.

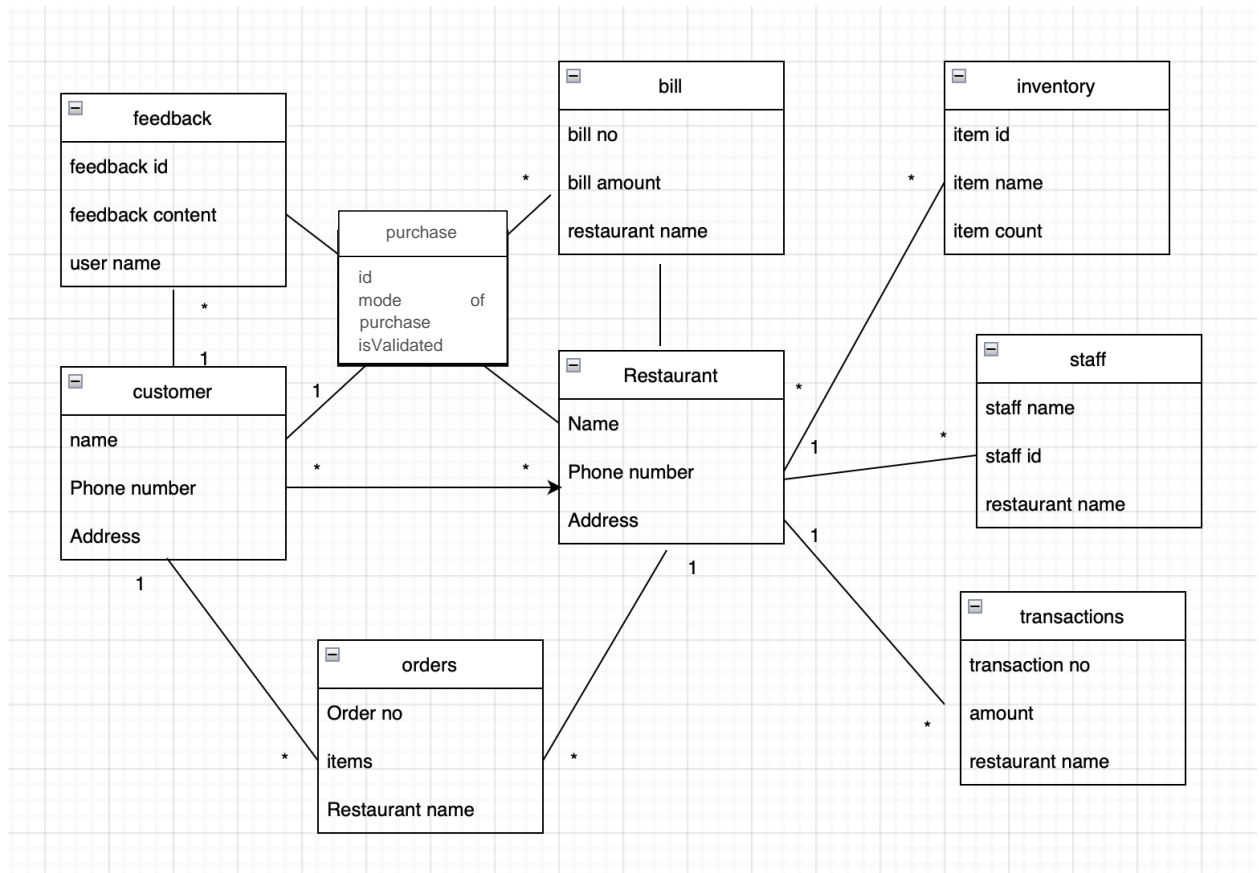
## Actor System Interaction Model. - Customer Feedback

	Actor: Customer	System
1	TUCBW leaving feedback for the restaurant	2 System navigates to a feedback page
3	Customers write feedback to the restaurant in the given space and send it	4 System stores the feedback in their database
5	TUCEW customer leave feedback	

In the above actor system interaction model, we can see how a customer can leave feedback regarding food or restaurant.

- User can leave feedback on restaurant or food in the feedback page available on the application.

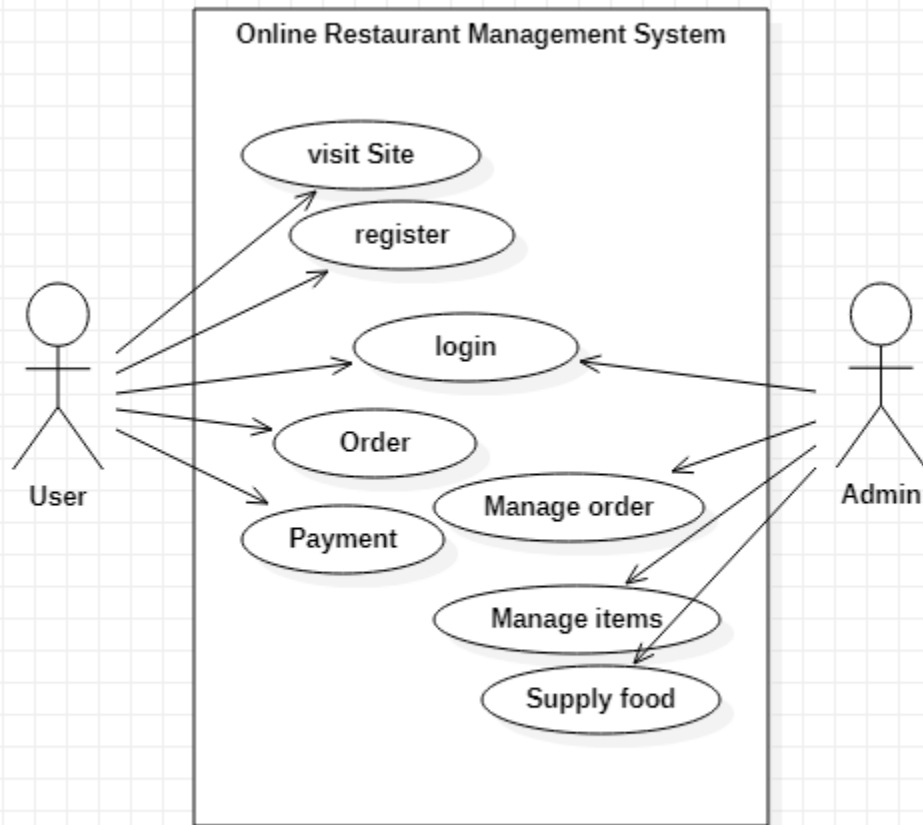
## Domain model



Here in this model, we can see the interactions and the flow of work in this application.

- User can interact with the restaurant to order food and also leave any feedbacks.
- User can monitor the orders on the application.
- Restaurants can check the transactions and bills.
- Restaurants can have the staff data.
- Inventory in the restaurant can be accessible.
- User can place several orders but sometimes orders due to unavailability will be purchased only purchased orders are billed.

# Use Cases



The High-level Use Cases are

- \* User should be able to login / Signup into the application
- \* User should be able to order and make payment for his order
- \* The Admin should be able to manage order and the items available
- \* The Admin should confirm whether the food is delivered or not.

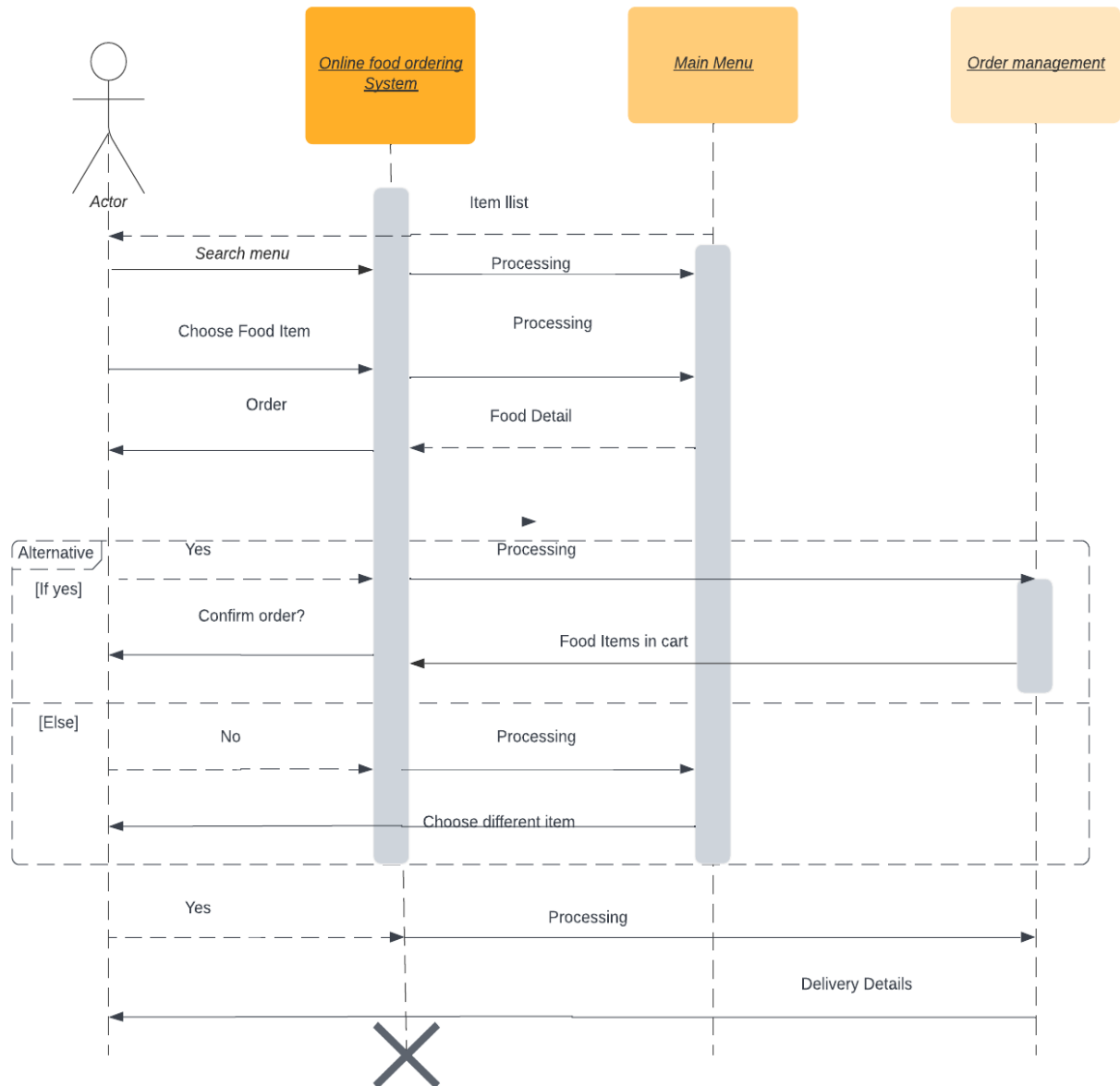
## Use cases for Restaurant/Admin:

- UC1 Receive orders
- UC2 Makes Payments
- UC3 Receive Feedbacks
- UC4 Manages Staff
- UC5 Manages Inventory
- UC6 Manage order status
- UC7 Generate Receipts

**Use cases for Customers:**

- UC8 Register customer
- UC9 Login Customer
- UC10 Logout Customer
- UC11 Reset Password
- UC12 Search Food
- UC13 Add into cart
- UC14 Make an order
- UC15 Tracking an order
- UC16 Notify User
- UC17 Provide a Feedback

# Sequence Diagram



# Requirement to Use Case Traceability Matrix

Requirements	Priority	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15	UC16	UC17
R1	5								x									
R2	5									x								
R3	4										x							
R4	3											x						
R5	4						x											
R6	5												x					
R7	2					x												
R8	3	x															x	
R9	4												x					
R10	2							x										
R11	1							x										
R12	5													x				
R13	4													x	x			
R14	4	x													x			
R15	2					x												
R16	2															x		
R17	1			x														x
R18	3		x			x												
R19	3			x			x											
R20	2				x													
	Score	7	3	4	2	7	7	3	5	5	4	3	9	9	8	2	3	1

# Increment Matrix

Usecase	Priority	Resources per week	Dependency	Assigned	Iteration 1	Iteration 2	Iteration 3
UC1	7	2	UC14, UC16	SM		2	
UC2	3	2	NONE	SS			2
UC3	4	1	UC17	RU		1	
UC4	2	4	NONE	MT,RU			4
UC5	7	3	UC2, UC14	SM			3
UC6	7	4	UC1, UC14	SS, MT			4
UC7	3	2	UC14	RU			2
UC8	5	3	NONE	MT	3		
UC9	5	3	UC8	SM	3		
UC10	4	2	UC8, UC9	SS	2		
UC11	3	1	UC8	RU	1		
UC12	9	5	UC8, UC9	MT, SS,SM	4	5	
UC13	9	4	UC12	SM, RU, SS	5	4	
UC14	8	4	UC13	SS, MT		4	
UC15	2	2	UC14	RU		2	
UC16	3	2	UC14	MT			2
UC17	1	1	UC6, UC14	SM			1
# of resources					18	23	16
# of hours					54	69	48
1 Resource per week=3 hours							
SM =>Shubash Muniyappa							
SS =>Shreethika Sivakumaran Sathyama							
RU =>Radha Yashwanth Uppuganti							
MT =>Bhargava Manikantan							



# Testing:

Software testing can take a number of different forms. Static testing involves reviews, walkthroughs, or inspections; dynamic testing involves actually running computer code against a predetermined set of test cases.

Static testing may be skipped, which is very frequently done in practice. When the application itself is utilized for the first time, dynamic testing occurs (which is generally considered the beginning of the testing stage).

Before the software is fully finished, dynamic testing could start to test specific code areas (modules or discrete functions). Use of stubs/drivers or execution from a debugger environment are two common methods for this. For instance, spreadsheet programs are by their very nature extensively interactively ("on the fly") tested, with the outcomes of each calculation or text manipulation displayed immediately.

When a tester gets access to internal data structures, algorithms, and the code that implements them, this is referred to as white box testing.

A software testing technique called "black-box testing" looks at an application's functionality without looking at its core components or workings. Practically every level of software testing, including unit, integration, system, and acceptance testing, may be conducted using this test methodology.

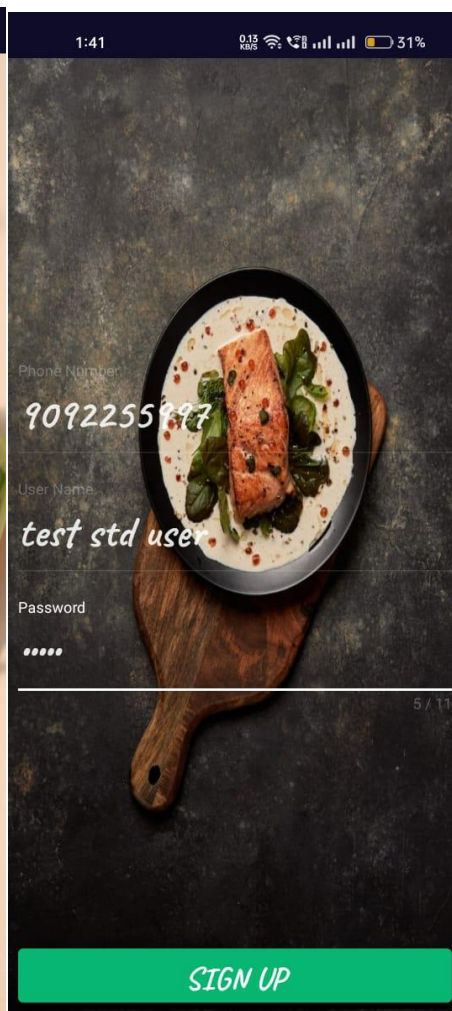
## Test Cases:

Test case ID	Scenario	Input data	Expected result	Actual result	Status
TC1	Run	Run the apk	It should run without error	Successfully running	Success
TC2	Register	Phone #, Name, Password	It should create a new user	Created the user	Success
TC3	Login	Phone #, Password	Display search bar	Logged in successfully	Success
TC4	Search	Food item	Display related food item	Displayed food item	Success
TC5	Select item	Click on food item	Display food item	Food is displayed	Success
TC6	Cart	Add item to cart	Items visible in cart	Selected items displayed in cart	Success
TC7	Order	Click order	Order placed	Items found in order module	Success
TC8	Change order status by admin	Open order at admin port change the status	Order status changed to delivered	Order changed to delivered only if key Yes is input. For int type no change	Fail
TC9	Status at user	NA	Status to be changed to delivered	Status changed to delivered	Success
TC10	Create admin	Phone #, Name, Password	Admin login to be created	User is created	Fail
TC11	logout	User Quits	Back to login	Back to login	Success

## Result :



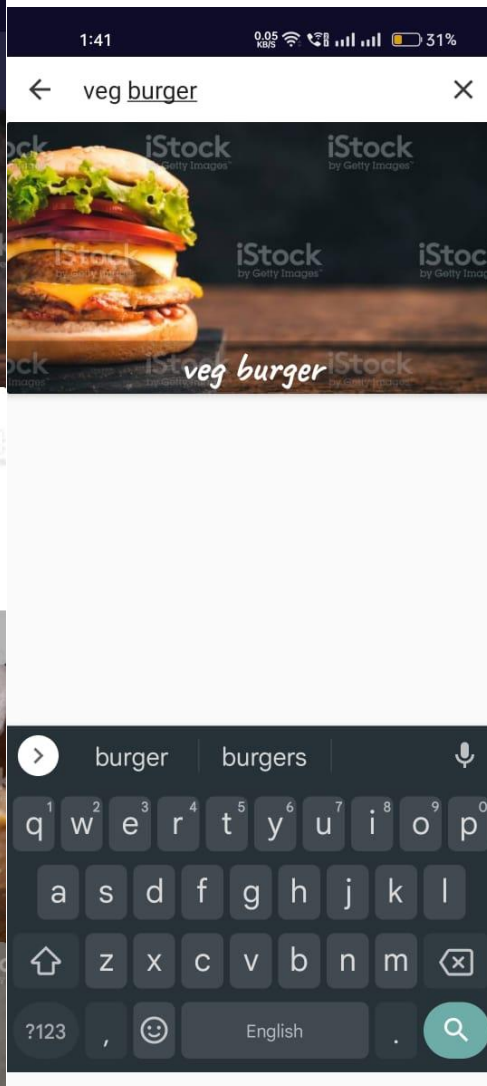
Default home page of the application



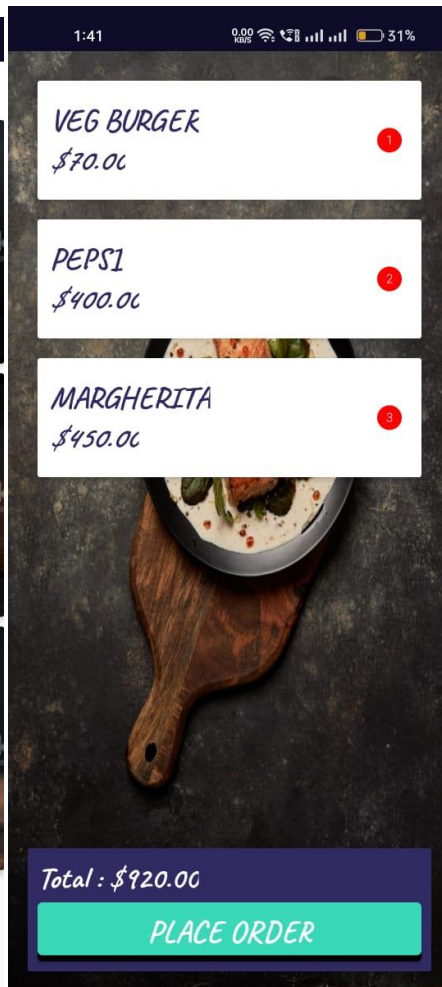
Sign-up Page for the Application



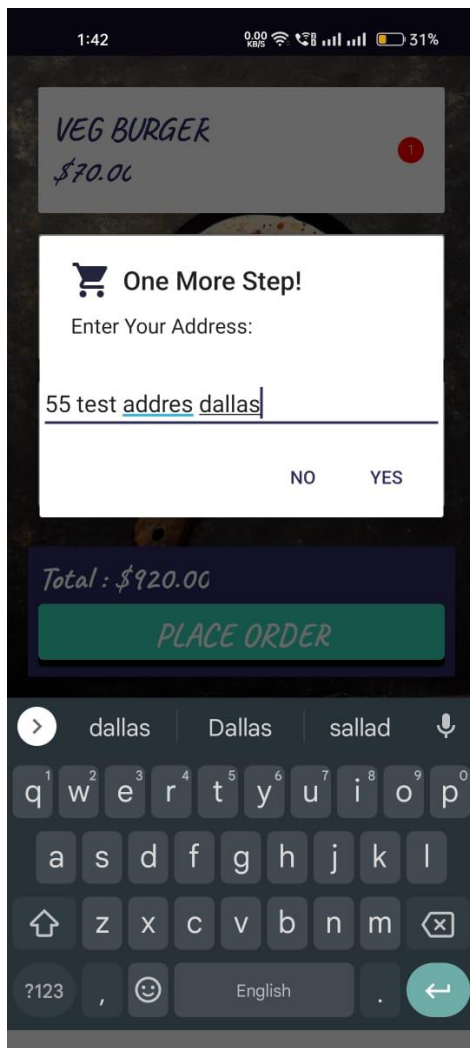
Home page of Standard User



Search Bar with Search Results



Detail View Of the Category Burgerz    Cart Containing Placed Order



Getting Delivery Address to Order

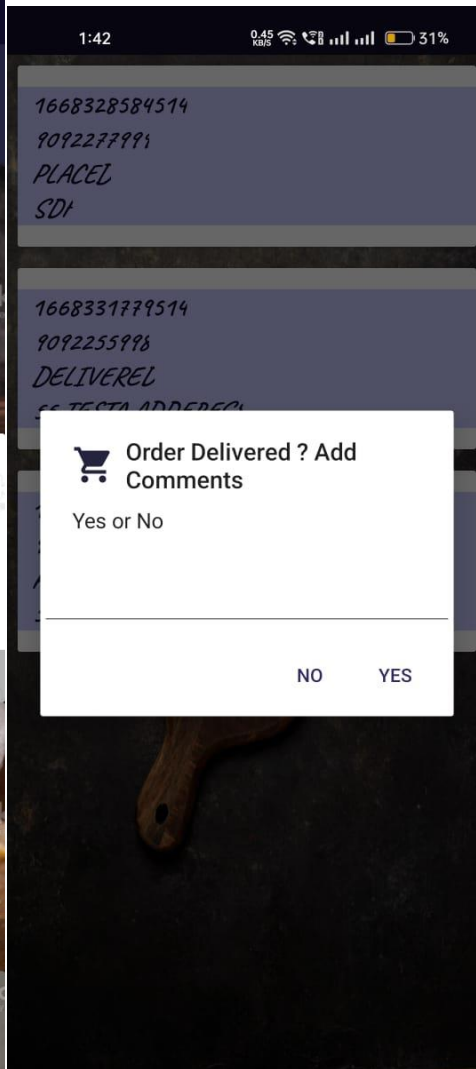


Placed Order List

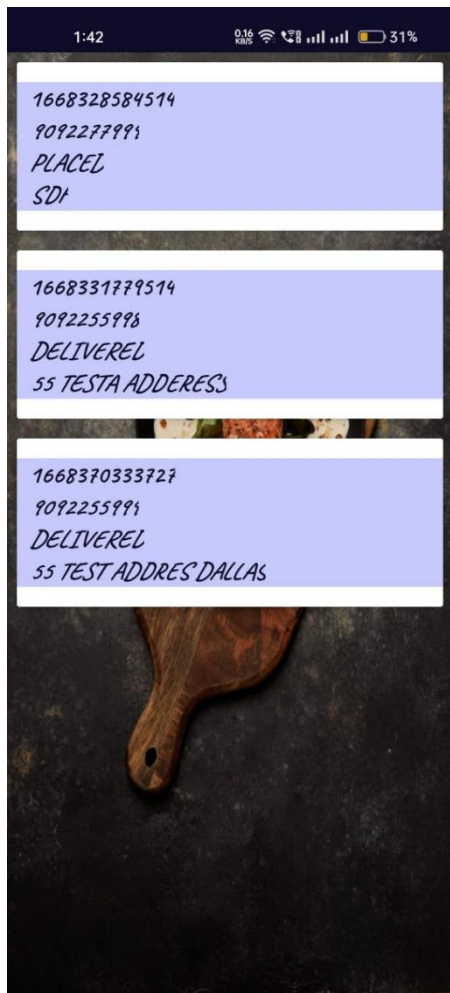




Admin User Home Page



Order Status Update Pop-up Only for Admin user



Order status as Delivered