Software Requirements Specification

for

Take Out

Prepared by Group 3

**128236 - Oyinbobola Shalom Owojori**

**138859 - Abdulrazak Omeiza Yakubu**

**138893 - Kamshinen Rotdung**

**149190 - Zainab Sada**

24.04.2017

# Introduction

This section gives a scope description and overview of everything included in this SRS document

## Purpose

## The purpose of this SRS is to describe the software functional and nonfunctional requirements for the Mobile Food Ordering System – Take Out. This document will also highlight the external interface requirements as well as its interactions with other external applications. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system.

## Document Conventions

This document conforms to the IEEE standard for Software Requirements Specification. Bold face and indentation is used on general topics and/or specific points of interest. The remainder of the document will be written using the standard font, New Times Roman.

## Intended Audience and Reading Suggestions

This document is intended to be read by the client. This is a technical document and the terms should be understood by the customer.

## Product Scope

The food ordering system - Take Out, project is a mobile android application that allows users to order food from local restaurants that have delivery services with ease, provided that they have an internet connection. It also allows restaurants receive order notifications and respond to them. It offers a more efficient customer to restaurant relationship.

One of the aims of the project is to design a mobile app that makes ordered food more accessible for people and to digitize the food ordering process. Eliminating the need to talk on the phone. Also have your favorite dishes known by the restaurants for improved customer service.

The application offers a **digital menu** that allows users(customers) to easily order and pick from optional ingredients thus making sure they get exactly what they want to eat without any extra unwanted ingredients. Avoiding the issues of Allergic reactions due to not knowing what your food is made of.

The purpose of the project is to make the food ordering process seamless, digital and save money while doing so. Eliminating the need to make a phone call when you want to order food and improving delivery response time and also more care for the customer’s food.

## References

1. Wiegers, Karl. Cafeteria Ordering System Vision and Scope Document,

[www.processimpact.com/projects/COS/COS\_vision\_and\_scope.doc](http://www.processimpact.com/projects/COS/COS_vision_and_scope.doc)

1. Patel, Mayurkumar, "Online Food Order System for Restaurants" (2015). Technical Library.Paper 219.

<http://scholarworks.gvsu.edu/cistechlib/219>

1. IEEE Software Engineering Standards Committee, “IEEE Std 830 - 1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

# Overall Description

## Product Perspective

Take Out is a system that replaces the current online food ordering system; Yemeksepeti. The context diagram in Figure 1 illustrates the external entities and system interfaces for the system. Although the mobile application is the main focus of this project, there is a server side which is responsible for the database service.



Figure Context Diagram

## Product Functions

Depending on the user class the food ordering system will have the following basic functions:

* Create an account.
* Manage their account.
* Log in to the system.
* Search restaurant or food
* Navigate the restaurant’s menu.
* Select an item from the menu.
* Add an item to their current order.
* Review their current order.
* Remove an item/remove all items from their current order.
* Provide payment details.
* Place an order.
* Receive confirmation in the form of an acceptance message from the restaurant.
* View order place
* Add/update/delete food category to/from the menu.
* Add /update/delete food item to/from the menu.
* Accept/Decline order.
* Update price for a given food item.
* Update additional information (description, photo, etc.) for a given food item.

## User Classes and Characteristics

The systems will be used on a mobile phone. The customers and restaurants are main users.

1. Customer: The users that place orders and pay.
2. Restaurant: The users that receive the orders and delivers the food.

## Operating Environment

OE-1: The food ordering system shall operate on any phone with an Android operating system.

## Design and Implementation Constraints

CO-1: The system’s design, code, and maintenance documentation shall conform to IEEE standards.

CO-2: The system shall use the current corporate standard Oracle database engine.

CO-3: All scripts shall be written in Java.

## User Documentation

The application will contain a Help menu, which is a collection of topics covering each of the applications menus and features which the user can always navigate to when they need help.

## Assumptions and Dependencies

It is assumed that the application always be used on mobile phones that have enough performance. If the phone does not have enough hardware resources available for the application, the application may not work as intended.

# External Interface Requirements

## User Interfaces

1. General Layout:

* From every menu, the user can access the built-in settings and back/previous button
  + These are physical buttons built into the phone.
* Every menu will have a descriptive title
* Menus will be readable, but will make efficient use of space so the user isn’t forced to navigate through too many menus

1. Welcome Screen:

* For first-time users only
* Prompts the user to enter account information to be stored on the server
* Notifies the user if their information is invalid

## Hardware Interfaces

The food ordering system is intended as a mobile application for the Android platform and hence is solely supported on Android-powered devices.

The Android platform provides abstractions for all network communication interfaces and thus the hardware as well.

## Software Interfaces

The Take Out app is to be developed under the Android operating systems using the Java JDK (Java Development Kit) and the Android SDK (software development kit) tools.

The application will interface with a Database Management System (DBMS) that stores the information necessary for the system to operate. The DBMS must be able to provide, on request and with low latency, data concerning the restaurant's menu, employees (and their passwords) and available dietary requirements.

## Communication Interface

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for both the mobile application

# System Features

## Customer

### Description and Priority

### The following are features of the system associated with the customer. They describe how the customer interacts with the system.

### Functional Requirements

REQ1: The system should enable Customers to register.

* Collect user information (Name, Email, Password)
* Check if information is valid
* Password not empty
* Password and Password confirm the same
* Email has not been used before and right Email format.
* If information is valid, save and add user to database
* Send confirmation message to user’s email.
* When user clicks confirmation link, authorize user access.

REQ2: The system should enable Customers to login.

* They shall enter their email and password.
* The information given shall be valid.
* Access shall be granted/denied.

REQ3: The system should enable Customers to logout.

* Log user out when user selects the logout button.

REQ4: The system should enable Customers to change a forgotten password

* Collect user email
* Send change password message to user’s email
* Collect user’s new password
* Save and edit user’s password in database.

REQ5: The system should enable customers to edit their profile.

* User shall be allowed to edit old information by entering new information
* This information shall replace the old one in the database

REQ6: The system should enable customers to view their profile.

* The system shall show users their information as saved in the database.

REQ7: The system should enable Customers to view Restaurants that deliver to their location.

* The system shall use the user’s address to display a page containing restaurants that deliver to the user’s address.
* The system shall display both restaurants that are open and closed for delivery, differentiating between closed and open with a color code.

REQ8: The system should enable Customers to view a restaurant.

* The system shall display the restaurant’s menu

REQ9: The system should enable Customers to search for menu item

* The Customer should be able to enter the name of the menu item to be searched for
* The system displays all menu items that fit the Customer’s search criteria.

REQ10: The system should enable Customers to view a menu item

* The system shall display the details of the selected menu item (Name, Picture, Description, Price)
* The system shall display the option to add the displayed menu item to cart

REQ11: The system should enable Customers to view cart

* The system shall display menu items the user has added to cart from various restaurants.
* The system shall display the selected quantity of each item.
* The system shall display individual and total prices of all displayed cart items

REQ12: The system should enable Customers to add menu items to cart

* The system shall allow customers to long press to add a menu item to cart.
* The system shall also allow customers to add menu item to cart by clicking a button when the customer views the menu item’s details.

REQ13: The system should enable Customers to remove menu items from cart

* The system shall display the option of deleting a menu item from the cart.

REQ14: The system should enable Customers to checkout / View order summary

* The system shall display all cart items (Name, Short Description, and Price) and allow the user to remove the item.
* The system shall display the total price of all the selected items
* The system shall give the user the option of selecting whether to pay with cash or card.
* The system shall allow the user to select delivery address from saved address list.
* The system shall allow the user to add a delivery address to the address list.
* The system shall allow the customer to reduce or increase quantity of a menu item.
* The system shall display a button that enables user to confirm and send the order to the restaurant.

REQ15: The system should enable Customers to add Addresses

* When editing profile and during checkout, the system shall give the user the option of adding an address to the users address list.

REQ16: The system should enable Customers to Send Order

* The customer should click the button on the checkout page that enables the user to send order.
* The system shall send the order to the respective restaurants the user has selected items from.

REQ17: The system should enable Customers to Receive order confirmation

* The system shall display an ‘order accepted’ message if the restaurant accepts the user’s order
* The system shall display an ‘order rejected message’ if the restaurant rejects the user’s order. The reason for rejection shall accompany the rejection message.
* If a time of X minutes(minutes saved on the user profile for wait time) runs out since the customer sent the order and the restaurant has not accepted or rejected the order, The system displays an order rejected message with reason – ‘time ran out’. Giving the user the option to resend the order.

REQ18: The system should enable Customers to search Restaurant names

* The Customer should be able to enter the name of the restaurant to be searched for
* The system displays all restaurants in the customer’s general location that fit the Customer’s search criteria.

REQ19: The system should enable Customers to set confirmation wait time

* The customer shall input the desired wait time before order is rejected by default due to time running out.
* The system shall allow restaurants to accept or reject the customer’s order in this period. After which a default reject message is sent to the customer with reason – ‘time ran out’. Giving the user the option to resend the order.

REQ20: The system should enable Customers to add restaurant and orders to favorites

REQ21: The system should enable Customers to view favorites

REQ22: The system should enable Customers to remove restaurant from favorites

REQ23: The system should enable Customers to view order history

## Restaurant

### Description and Priority

## The following are features of the system associated with the restaurant. They describe how the restaurant interacts with the system.

### Functional Requirements

REQ24: The system should enable Restaurants to register.

* The system collects restaurant information(Restaurant name, Email, Password)
* Check if information is valid
* Password not empty
* Password and Password confirm the same
* Email has not been used before and right Email format.
* If information is valid, save and add restaurant to database
* Send confirmation message to restaurant’s email.
* Send a special access code to the restaurant’s email.
* When restaurant clicks confirmation link, prompt for the access code.
* If access code is authorized, grant the restaurant access.

REQ25: The system should enable Restaurants to login.

* They shall enter their email, access code and password.
* The information given shall be valid.
* Access shall be granted/denied.

REQ26: The system should enable Restaurants to edit Restaurant info

* On first login, the system displays this page to fill restaurant info.
* The system shall allow the restaurant to fill in contact info (Phone number, Address, State, Region, Country)
* The system shall allow the restaurant to pick delivery regions.
* The system shall allow the restaurants to set delivery time
* The system shall allow the restaurants to upload logo

REQ27: The system should enable Restaurants to build menu

* The system shall allow the restaurant to input menu item (name, description, price, picture)
* The system shall allow the restaurant to add optional menu items (name, picture, price)

REQ27: The system should enable Restaurants to view menu

* The system shall display the saved restaurant menu as it will be displayed to the customers.

REQ28: The system should enable Restaurants to View Orders

* The system shall display all current orders becoming less opaque (disappearing) as the time for the order confirmation runs out.

REQ29: The system should enable restaurants to delete orders

* The system shall display accepted or rejected offers with a delete button
* The system shall delete orders every hour regardless of whether the restaurant deletes it or not

REQ30: The system should enable Restaurants to Select/Open Orders

* The system shall display a page with order details (Items, Optional Ingredients, Order address, Price, Customer telephone number, etc.)

REQ31: The system should enable Restaurants to Accept/Reject Orders

* The system shall display all current orders with an accept or reject button

# Other Nonfunctional Requirements

1. **Reliability**
   * The system should be available when requested for service by users: The system should work 24/7, it should always be up and running so that whenever the user wants to use it, it is available.
   * The system should have a very low failure rate: The failure rate should be kept as minimal as possible, preferably less than 0.01.
2. **Performance**
   * The system must have a good response time.
   * The load time for the user interface should take less than two seconds.
   * The log in information should be verified within five seconds.
   * Queries shall return results within five seconds.
   * The system should be able to achieve a lot in a specified amount of time.
   * The system should be able to withstand a heavy workload.
   * It should be able to respond to multiple numbers of people at the same time.
   * The system must run error free while operating with a huge set of data.
   * The system should be precise and accurate when dealing with data.
   * The system’s error rate should be minimal
3. **Security**
   * All external communications between the system’s data server and clients must  
     be encrypted
   * To ensure that the system is secure access to the various subsystems will be  
     protected by a user log in screen and requires a user name and password.
   * All system data must be backed up every 24 hours and the backup copies stored  
     in a secure location which is not in the same building as the system: This is  
     done to avoid loss of information in case of system crash. The system data  
     should be stored in storage device e.g. hard drive, CD, Flash drive or it could be  
     stored in files.
4. **Usability**
   * The system should include a questions page for user’s to ask questions or complaints about the application
   * The system should have a frequently answered questions page to tackle major frequent issues that user’s might have with using the application.
   * The system should have a well formed, easy to use graphical user interface
   * The system should be user friendly
   * The system must be easy to learn for both novices and users with experience from similar systems
   * The system must be efficient for the frequent user
   * The system must be easy to remember for the casual user
   * The user must understand what the system does
   * The user must feel satisfied with the system
5. **Safety**
   * The system should maintain a good backup: Maintaining backups ensures that the system’s database is secure, which means that in case of an emergency or accident the system can be easily restored.
6. **Supportability**
   * The system should be easy to maintain.
   * In order for the system to be easy to maintain it is done with an objected oriented language, which is easy to maintain.
   * Maintenance of the system should be cost efficient
   * Maintenance of the system should be less frequent
   * The system should easily adapt to changes made
   * The system should be able to detect the location of the user

# Use cases

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-1: | | | Login |
| Related  Requirements: | | | REQ 2, 25 |
| Initiating Actor: | | | Customer, Restaurant |
| Actor’s Goal: | | | To log into account |
| Participating  Actors: | | | Customer, Restaurant |
| Preconditions: | | | The system displays the login page for customer or restaurant to enter login details – (Email and Password for customer. Restaurant name, Access code and Password for restaurant). |
| Post conditions: | | | Customer: The system should redirect user to the display restaurants page.  Restaurant: On first login, the system should redirct user to the build restaurant page. For subsequent logins, the system should redirect user to the view orders page. |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer/Restaurant opens application and clicks customer/restaurant. | |
|  | 2. | System shows user a menu to enter login details. | |
|  | 3. | User enters the information and clicks on login button. | |
|  | 4. | System redirects user to default page after login successful. For customers, display restaurants. For Restaurants, view orders. | |
| Flow of Events for Extensions (Alternate Scenarios) : User enter invalid username or password | | | |
|  | 1. | System shows a warning message that the login details entered were wrong and asks the user to enter information again. | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-2: | | | Register |
| Related  Requirements: | | | REQ 1, 24 |
| Initiating Actor: | | | Customer, Restaurant |
| Actor’s Goal: | | | To register for an account |
| Participating  Actors: | | | Customer, Restaurant |
| Preconditions: | | | The system displays the register page for customer or restaurant to enter details. |
| Post conditions: | | | The system should redirect user to the registration confirmed page |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer/Restaurant opens login page and clicks register | |
|  | 2. | System shows user a menu to enter details. – (Name, Email, Password and Confirm Password). | |
|  | 3. | User enters the information and clicks on register button. | |
|  | 4. | System redirects user to the registration confirmed page. Telling the user to check their mail for further instructions. | |
|  | 5. | Customer: System sends message to user’s email with a link to confirm registration  Restaurant: System sends message to user’s email with a link to confirm registration and a special access code for security purposes | |
|  | 6. | User clicks confirm registration link | |
|  | 7. | System displays registration successful message. Allowing user to login. | |
| Flow of Events for Extensions (Alternate Scenarios) : User enter invalid username or password | | | |
|  | 1. | System shows a warning message that username or password entered was wrong and asked for entering information again. | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-3: | | | View restaurants |
| Related  Requirements: | | | REQ 7 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To view restaurants close to user’s location |
| Participating  Actors: | | | Customer |
| Preconditions: | | | Customer must be logged in |
| Post conditions: | | |  |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer/Restaurant logs in or clicks ‘restaurants’ link. | |
|  | 2. | System shows the user a page containing restaurants that can deliver to the user’s location. The page displays restaurant names, short descriptions, if the restaurants are open or closed. | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-4: | | | Select/Open Restaurant |
| Related  Requirements: | | | REQ 8 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To view a restaurant’s menu |
| Participating  Actors: | | | Customer |
| Preconditions: | | | The customer taps on a restaurant from the restaurants page. |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer/Restaurant opens restaurant page and selects a restaurant. | |
|  | 2. | System shows user a page containing the restaurant’s name and menu. | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-5: | | | Select menu item |
| Related  Requirements: | | | REQ 10 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To view details about an item on a restaurant’s menu. |
| Participating  Actors: | | | Customer |
| Preconditions: | | | The user clicks on a restaurant’s menu item. |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer selects a menu item | |
|  | 2. | System shows user a page with the details of the menu item. İncluding the option to add the selected item to cart. | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-6: | | | Add item to Cart |
| Related  Requirements: | | | REQ 12 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To add a menu item to cart |
| Participating  Actors: | | | Customer |
| Preconditions: | | | 1. The user selects the ‘add to cart’ option on the view menu item page. 2. The user long presses a menu item from the view restaurant menu page. |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer selects the ‘add to cart’ option on the view menu item page  Customer long presses a menu item from the view restaurant menu page. | |
|  | 2. | System displays an ‘added to cart’ message. | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-7: | | | Increase/Reduce item quantity in cart |
| Related  Requirements: | | | REQ 11 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To change quantity of items selected for order |
| Participating  Actors: | | | Customer |
| Preconditions: | | | The user opens the cart |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer selects the cart on the application | |
|  | 2. | System displays the cart containing all items selected to be ordered, giving the user the option to send order | |
|  | 3. | Customer selects either the + or – buttons to increase or reduce item quantity respectively | |
|  | 4. | System responds to user’s choice: increase quantity by 1 if user selects ‘+’, reduce quantity by 1 if user selects ‘–‘ | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-8: | | | View Cart |
| Related  Requirements: | | | REQ 11 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To view items selected for order |
| Participating  Actors: | | | Customer |
| Preconditions: | | | The user opens the cart link |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer selects the cart link on the application | |
|  | 2. | System displays the cart containing all items selected to be ordered, giving the user the option to send order | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-9: | | | Remove item from cart |
| Related  Requirements: | | | REQ 13 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To remove a menu item from cart |
| Participating  Actors: | | | Customer |
| Preconditions: | | | 1. The user opens the cart menu. 2. The user selects the ‘x’ beside the menu item wished to be deleted on the cart. |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer views cart. | |
|  | 2. | Customer selects the ‘x’ beside the menu item wished to be deleted on the cart. | |
|  | 2. | System deletes item from cart and displays a ‘removed’ message. | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-10: | | | Customer View Orders |
| Related  Requirements: | | | REQ 23 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To view previously sent orders |
| Participating  Actors: | | | Customer |
| Preconditions: | | | The user is logged in |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer selects view orders | |
|  | 2. | System displays page showing history of previously sent orders | |
|  | 3. | Customer selects a previously made order | |
|  | 4. | System shows user details of the order and option to reorder items | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-11: | | | View Favorites |
| Related  Requirements: | | | REQ 21 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To view previously saved favorite restaurants and orders |
| Participating  Actors: | | | Customer |
| Preconditions: | | | The user opens favorites |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer selects favorites on the application | |
|  | 2. | System displays the favorites containing the customers previously saved restaurants and orders. | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-12: | | | Add restaurant to favorites |
| Related  Requirements: | | | REQ 20 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To add a liked restaurant to favorites |
| Participating  Actors: | | | Customer |
| Preconditions: | | | The user is viewing the restaurant’s menu |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer selects add to favorites on the restaurant’s menu page | |
|  | 2. | System adds restaurant to favorite restaurants list and displays ‘added to favorites’ | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-13: | | | Add order to favorites |
| Related  Requirements: | | | REQ 20 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To add a specific order to favorites |
| Participating  Actors: | | | Customer |
| Preconditions: | | | 1. The user is on the place order page 2. The user is on the view orders page |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer selects add order to favorites on the place order page or when viewing a previous order on the view orders page | |
|  | 2. | System adds order to favorite orders list and displays ‘added to favorites’ | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-14: | | | Send order/Checkout  \*/8/89  3 |
| Related  Requirements: | | | REQ 16 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To send selected items to restaurant as order |
| Participating  Actors: | | | Customer, Restaurant |
| Preconditions: | | | Customer clicks the ‘send order’ button on the cart |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer views cart and selects the send order button on the cart. | |
|  | 2. | System displays page containing cart items, price and other options. Options include: Pick address, Add new address, pay with cash/card, change quantity, delete item, cancel order, set wait time. | |
|  | 3. | Customer adjusts order to preferences. And sets amount of time to wait for confirmation | |
|  |  | Customer selects ‘takeOut’ at the bottom of the page. | |
|  |  | System sends order to restaurant. | |
|  |  | Restaurant responds within specified wait time | |
|  |  | The system sends the customer a message with the restaurant’s response (accepted or rejected) | |
| Flow of Events for Extensions (Alternate Scenarios) : User selects cancel order | | | |
|  | 1. | System shows an alert asking the user if their sure they want to cancel the order. | |
|  | 2. | Customer selects yes | |
|  | 3. | System cancels order and returns user to main page | |
| Flow of Events for Extensions (Alternate Scenarios) : Restaurant does not accept/reject order within specified wait time | | | |
|  | 1. | System sends a message to customer ‘order rejected’ with reason – ‘time ran out’ and an option to resend order. | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-15: | | | Restaurant View Orders |
| Related  Requirements: | | | REQ 28 |
| Initiating Actor: | | | Restaurant user |
| Actor’s Goal: | | | To view orders sent from customers |
| Participating  Actors: | | | Restaurant user |
| Preconditions: | | | The user is logged in |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Restaurant selects view orders | |
|  | 2. | System displays page showing current orders sent from customers with options to accept or reject. With each individual order reducing in opacity according to time left as specified by customer | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-16: | | | Accept order |
| Related  Requirements: | | | REQ 31 |
| Initiating Actor: | | | Restaurant |
| Actor’s Goal: | | | To accept order |
| Participating  Actors: | | | Restaurant, Customer |
| Preconditions: | | | The user is logged in |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Restaurant selects accept order | |
|  | 2. | System changes order view options to view order and delete order | |
|  | 3. | System sends acceptance message to customer | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-17: | | | View order details |
| Related  Requirements: | | | REQ 30 |
| Initiating Actor: | | | Restaurant |
| Actor’s Goal: | | | To view an orders details |
| Participating  Actors: | | | Restaurant |
| Preconditions: | | | The user is logged in |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Restaurant selects an order | |
|  | 2. | System displays page showing details of order and options to accept or reject order | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-18: | | | Delete order |
| Related  Requirements: | | | REQ 29 |
| Initiating Actor: | | | Restaurant |
| Actor’s Goal: | | | To delete an order |
| Participating  Actors: | | | Restaurant |
| Preconditions: | | | The user is logged in |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Restaurant selects delete order | |
|  | 2. | System displays page with orders except the deleted order and displays message ‘order deleted’ | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-19: | | | Reject order |
| Related  Requirements: | | | REQ 31 |
| Initiating Actor: | | | Restaurant |
| Actor’s Goal: | | | To reject order |
| Participating  Actors: | | | Restaurant, Customer |
| Preconditions: | | | The user is logged in |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Restaurant user selects reject order | |
|  | 2. | System displays page showing reject reasons | |
|  | 3. | Restaurant user selects a reason and selects reject order | |
|  | 4. | System displays ‘order rejected’ and sends reject message with reason to customer | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-20: | | | Build Restaurant |
| Related  Requirements: | | | REQ 26 |
| Initiating Actor: | | | Restaurant |
| Actor’s Goal: | | | To fill details of restaurant on the application |
| Participating  Actors: | | | Restaurant |
| Preconditions: | | | 1. First time login for Restaurant |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Restaurant user logs in | |
|  | 2. | System displays message ‘Build your restaurant’ and prompts user to tap to continue to next step | |
|  | 3. | Restaurant user taps the page | |
|  | 4. | System displays page to input contact info (Phone number, Address Line 1, Address line 2, Address line 3, State, Region, Country) and prompts user to tap to continue to next step | |
|  | 5. | Restaurant user fills in info and taps to continue | |
|  | 6. | System displays page of selectable regions in the specified state and country of previously filled info asking the user to select regions food can be delivered to | |
|  | 7. | Restaurant user selects regions and taps to continue | |
|  | 8. | System displays page to set delivery time | |
|  | 9. | Restaurant user fills in delivery time and taps to continue | |
|  | 10. | System displays page to upload logo | |
|  | 11. | Restaurant user uploads logo and taps to continue | |
|  | 12. | System displays ‘Build menu’. Prompting the user to continue to the next step of building the restaurant’s digital menu that customers will see. | |
|  | 13. | Restaurant user taps screen to build menu | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-21: | | | Build Menu/Add menu item |
| Related  Requirements: | | | REQ 27 |
| Initiating Actor: | | | Restaurant |
| Actor’s Goal: | | | To build the restaurant’s menu on the application |
| Participating  Actors: | | | Restaurant |
| Preconditions: | | | Restaurant is logged in |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Restaurant user selects build menu/edit menu | |
|  | 2. | System displays page with options to add a menu item (Name, description, picture, price, add optional menu item, category). | |
|  | 3. | Restaurant user taps the + button to add the item to the menu. | |
|  | 4. | System displays page with options to either view menu or add another item to menu | |
|  | 5. | Restaurant user selects to add another item to menu | |
|  | 6. | Return to 2. | |
| Flow of Events for Extensions (Alternate Scenarios) : User selects view menu | | | |
|  | 6. | System displays built menu | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-22: | | | Edit Restaurant |
| Related  Requirements: | | | REQ 26 |
| Initiating Actor: | | | Restaurant |
| Actor’s Goal: | | | To edit details of restaurant on the application |
| Participating  Actors: | | | Restaurant |
| Preconditions: | | | Restaurant has to be logged in |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Restaurant user selects edit restaurant | |
|  | 2. | System displays message ‘Edit your restaurant’ and prompts user to tap to continue to next step | |
|  | 3. | Restaurant user taps the page | |
|  | 4. | System displays page with previously saved contact details and prompts user to tap to continue to next step | |
|  | 5. | Restaurant user edits info and taps to continue | |
|  | 6. | System displays page of selected delivery regions | |
|  | 7. | Restaurant user edits delivery regions and taps to continue | |
|  | 8. | System displays page with saved delivery time | |
|  | 9. | Restaurant user changes delivery time and taps to continue | |
|  | 10. | System displays ‘Details saved’ | |
|  | 11. | Restaurant user taps screen to return to view orders page | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case UC-23: | | | Logout |
| Related  Requirements: | | | REQ 3 |
| Initiating Actor: | | | Customer |
| Actor’s Goal: | | | To logout of account |
| Participating  Actors: | | | Customer |
| Preconditions: | | | The user is logged in |
| Post conditions: | | | None |
| Flow of Events for Main Success Scenario: | | | |
|  | 1. | Customer selects logout | |
|  | 2. | System displays prompt asking user ‘do you really want to log out?’ | |
|  | 3. | Customer selects yes | |
|  | 4. | System logs user out and redirects to login page | |
| Flow of Events for Extensions (Alternate Scenarios) : None | | | |

# Other Requirements

The database for Take Out will hold information about the customers, restaurants and thei order history. It will be using MySQL.

# Appendix A: Glossary

SRS Document – Software requirements specification document

IEEE - Institute of Electrical and Electronics Engineers

# Appendix B: Analysis Models