SRS DOCUMENT

Food Management System

ZOBIA SALMAN- 13122

Contents

1. Introduction

This document includes specifications and features of the software in detail. It helps understand the target audience and user classes accordingly and internal and external interface requirements.

1.1. Purpose

The purpose of this document is to mention requirements specification of “The Food court Management System”.

Food courts are a place to relax. During peak shopping seasons and sales, it gets very crowded the food stalls have long queues. Moreover, the manual system results in long hours of waiting and inconvenience. Workers have short break time and its difficult for them to leave their workplace during lunch hours

With the above stated problems, and the advancement in technology, we feel there is a need for an automatic system which makes it convenient for customers to dine inn and relax, pre-order and takeaway or get the food delivered to their shops.

The “Food court Management System” app allows users to scan QR code and place order online to avoid the hassles of long queues, hence save people time.

1.2. Document Convention

The main purpose of the app is to order food through QR scanning and it also provides additional options to pay online via credit card. There are multiple features of the app which are divided in 3 subsystems as mentioned in Scope and Vision document, however every feature has its own priority rate which makes it a complete successful application. However, some features have higher priority as

compared to others. For instance, “Payment online” has a lower priority than “Online food order”.

1.3. Intended Audience and Reading Suggestions

The SRS document is intended for all the stakeholders to confirm on all the requirements and features before initiating development. Stakeholders include the sponsors, Management team, Q/A staff Product Champion – (from each user group i.e. customers, shop keepers, office workers, Mall staff, Food brands).
The document states project scope with detail explanation of features and specifications. It will help for all the users to understand about the project and Q/A staff to prepare testing cases accordingly. The document also includes some limitations and constraints and the operating requirement in which the system would work.

1.4. Project Scope

The goal for developing this app is to create convenience for customers and save their time by utilizing the “pre-order” option.
Moreover, there are many other options which makes ordering and food selection feasible for the customers. Considering our target audience, budget is an important factor for middle class, students and workers while looking up for food options, hence the app provides “budget (low to high/high to low)” wise allocation of restaurants and deals making it convenient for people to order within their desired budget.
The project will be divided into three phases:

- Phase 1-Initial Phase: Basic Food Court QR App

- Phase 2: Online order placement and payment
- Phase 3: Recommendation and Data Analysis

Moreover, the business risks, opportunities and success measures have been mentioned in the Vision and Scope document provided. The scope of each release is also mentioned.

1.5. References

Refer to “Vision and Scope” document for detail understanding of scope of each subsequent release and the overall project

Refer to “Use cases” document which includes primary and secondary use cases with description and how each use case will interact with the system

Refer to “Business Rules” document to understand the business rules, limitations and constraints (if any) for this project.

2. Overall Description

The following section mentions overview of the system, its operating environment and dependencies.

2.1. Product Perspective

The “Food court Management System” is an entirely new application.

Few similar apps do exist abroad but fulfilling a different purpose. FoodPanda is a similar app but it is only for delivery with no dine inn option.

2.2. Product Features

Following are the major functional features of the app:

For Customers:

1. Scan QR Code to get list of food brands
2. Browse through different menus and deals along rates.
3. Customize the list according to the following criteria:

- All
- Food category
- Budget
- Deals/ Discounts
- Discount on Cards
- Ratings

4. Place Your Order for Dine inn
5. Place Your Order for Takeaway (Collect/Get it delivered)
6. Online Payment via credit card
7. Track Your Order
8. Helpline/Call for Complain
9. Report a Problem

For Food Brands:

1. Food brands will subscribe to create their profile
2. Food brands will get a list of malls to select in which mall they are present
3. Food brands can create/edit/delete Category Page
4. Food brands can create/edit/delete deals and discounts
5. Food brands can create/edit/delete branch or restaurant
6. Take Order

7. Payment Receipt

8. Food brand can communicate with customers through feedback/comments

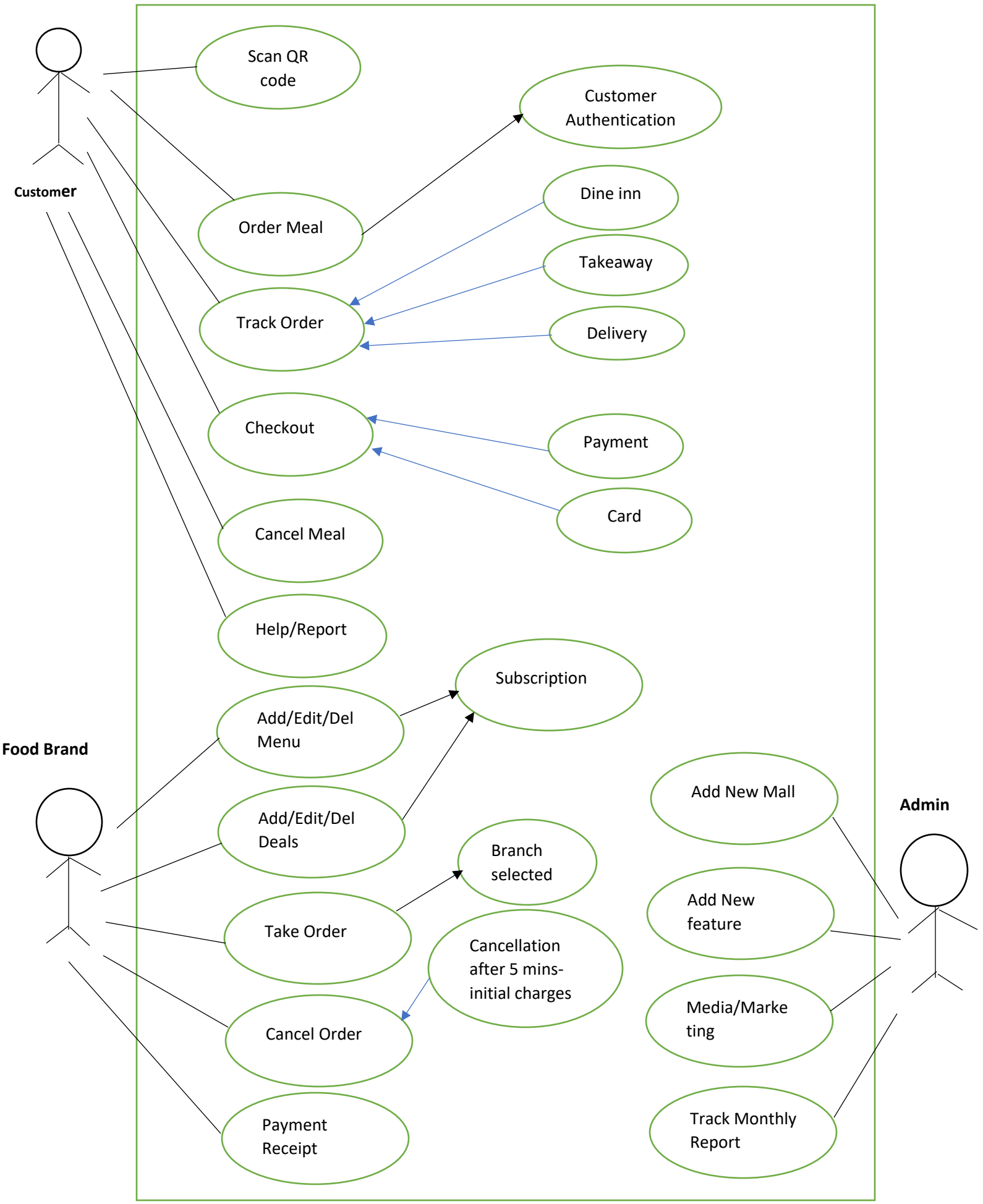
For Admin:

1. Add new feature

2. Manage accounts

3. Media/Marketing purpose

4. Track monthly report


Customer
Scan QR code
Customer Authentication
Order Meal
Dine inn
Takeaway
Track Order
Delivery
Checkout
Payment
Card
Cancel Meal
Help/Report
Food Brand
Subscription
Add/Edit/Del Menu
Add/Edit/Del Deals
Branch selected
Take Order
Cancellation after 5 mins- initial charges
Cancel Order
Payment Receipt
Admin
Add New Mall
Add New feature
Media/Marketing
Track Monthly Report

2.1. User classes and Characteristics

There are three major favored user classes:

- Customers (Dine inn, Takeaway, Delivery)
- Food brands
- Admin

The following are secondary users which somehow are interlinked with the system but not necessary use it:

- Banks
- Mall Management

2.2. Operating Environment

2.3. User classes and Characteristics

There are three major favored user classes:

- Customers (Dine inn, Takeaway, Delivery)
- Food brands
- Admin

The following are secondary users which somehow are interlinked with the system but not necessary use it:

- Banks
- Mall Management

2.4. Operating Environment

The system/application will be used in following environment:

• The users will be located close to each other in one region, same city. Later this can be extended to different cities if required.

• The users will access the system from 11am onwards to 11pm or 12pm, according to mall and food court timings.

• The users cannot tolerate service interruptions as this would annoy the user resulting in not using the application. Moreover, any service interruption would affect the entire process, placing orders etc. which would result in loss of business of food brands.

2.5. Design and Implementation Constraints

- Internet enabled computers are required. Moreover, for food brands a Web app is required.
- Customers will be required internet access to scan and order
- The mobile app and web app development can be done using a framework
- The system should follow all the business rules as stated in “Business rules” document
- The memory usage of the app will have to be constrained by the devices it is intended to run. Since most tablets, Android phones may have limited memory

2.6. User Documentation

Following user documentation components will be provided:

- User manuals for food brands
- Short tutorial clips for customers and brands
- User guide pamphlet for customers

2.7. Assumptions and Dependencies

Following are the assumptions:

AS-1 Internet-enabled computers will be available in the food court at each food stall

AS-2 Customer will be required internet access to order

AS-3 Riders will be available to deliver the food for nearby offices

The app will be dependent on certain external factors such as:

DE-1 The memory usage of the app will have to be constrained by the devices it is intended to run. Since most tablets, Android phones may have limited memory.

DE-2 The app would be dependent on rules and regulations by government authorities

DE-3 The system would require cloud for data storage

3. System Features

Following are the major features of the system. The detail description of each feature is given below:

3.0. System Feature - Customers

3.1. System Feature - Scan QR code and Login

3.1.1. Description and Priority

The feature will allow users to scan QR code and login to the app. Customers will be able to browse through the menu and check out different deals and discounts

3.1.2. Response Sequence

- The user will scan QR code
- User will login and is landed on home page

- The system displays various food brands on the home page according to user's preferences
- The system displays deals and discounts on the mid
- The user can browse through the menu of each food brands
- On clicking the menu, the system will display the price and ask the user to continue order or return to Home page

3.1.3. Functional Requirements

FE-1: Invalid login:

1. User enters wrong id and password
2. System provides 3 opportunities for correct Id password
    - System terminates the case

3.2. System Feature – Order Meal

3.2.1. Description and Priority

The feature will allow users to order meal through the app

3.2.2. Response Sequence

- The user will select a food brand and open its menu by clicking on its logo
- The system displays any deals of that particular brand on the top
- The user selects the meal he wants to order by clicking on it
- The system adds the meal to the cart and ask user to "continue order" or "select another meal".

- On continue, the system requires user details and payment method
- The system asks the user for “dine inn, take away or delivery” options
- The user selects the option as required. If the user selects “delivery”, the system shows the rider details and delivery time.
- After providing all the details, the user clicks on “Submit”
- The system displays a confirmation message that order has been placed
- The system returns to Home page

3.2.3. Functional Requirements

FE1: Cannot order (Restaurant is closed):

1. System terminates the order
2. User cancels the order

FE2: Rider not available

1. System sends a message for order delay due to non-availability of rider
2. Delivery not possible:

• System terminates the order

• User cancels the order

FE3: Meal is not available

1. System sends a message for order delay due to no availability of meal
2. Order is not possible:

• System terminates the order

• User cancels the order

3.3. System Feature – Payment

3.3.1. Description and Priority

The feature will allow user to select a payment method i.e. credit card or cash

3.3.2. Response Sequence

- Once the order is confirmed, the system asks user to select any payment method
- If the user selects cash, the order is confirmed and returns to Home page
- If the user selects online payment via credit card, the system shows different credit card options
- The user selects the required credit card
- The system displays discount on the credit card (if any)
- The user provides the credit card details
- The system sends and OTP confirmation
- The user enters the code and receives a confirmation message by the app and bank

3.3.3. Functional Requirements

FE1: Wrong OTP code

System gives customer three opportunities for correct OTP Code

1. System terminates the order

FE2: Wrong card No

System gives customer three opportunities for correct OTP Code

1. System generates an error message of wrong card No
2. System terminates the order

3.4. System Feature – Track Order

3.4.1. Description and Priority

The feature allows users to track their order

3.4.2. Response Sequence

- Once the order is confirmed and payment is done, the system shows a “Track your Order”
- The screen shows the order status and remaining time

3.5. System Feature – Rating / Feedback

3.5.1. Description and Priority

After order is delivered, the system asks user to rate the service and give feedback

3.5.2. Response Sequence

- Once the order is delivered, system displays rating screen
- User can rate from 1-5 (1 - lowest and 5 – highest) for the service and food quality
- User can also give feedback by sharing his comments
- User can select “Help/Complaint” option and system connect the user to customer service.

3.1. System Features – Food brand

3.6. System Feature – Create Profile

3.6.1. Description and Priority

After subscription, the food brands will be able to create their profile by providing relevant details

3.6.2. Response Sequence

- The food brands will login and create their profile
- The system will display the profile to public

3.6.3. Functional Requirements

FE1 Wrong CNIC

System gives user three opportunities for correct CNIC

1. System generates an error message of wrong card No
   • System terminates the order

3.7. System Feature – Create/Update/Delete Menu

3.7.1. Description and Priority

The food brands will be able to create a new menu, update the existing one or delete the existing menu

3.7.2. Response Sequence

- After the profile is created, the system shows an option to “create your menu” if it doesn’t exist.
- Once the menu is created, the system shows an “edit or delete” button at the top
- All the recent changes are saved

3.7.3. Functional Requirements

FE1: Cannot create menu on specified date

1. Food brand requests to create menu for the specified date

2. If Food brand requests to cancel menu for the specified date

• System terminates the use case

FE2: Cannot update menu on specified date

1. Food brand requests to update menu for the specified date

2. If Food brand requests to cancel menu for the specified date

• System terminates the use case

3.8. System Feature – Create/Update/Delete Deals

3.8.1. Description and Priority

The food brands will be able to create a new deal, update the existing one or delete the existing deal

3.8.2. Response Sequence

- After the menu is created, the system shows an option to "create your deal" if it doesn't exist.
- Once the deal is created, the system shows an "edit or delete" button at the top
- All the recent changes are saved

3.8.3. Functional Requirements

FE1: Cannot create deal on specified date

1. Food brand requests to create deal for the specified date

2. If Food brand requests to cancel deal for the specified date

• System terminates the use case

FE2: Cannot update deal on specified date

1. Food brand requests to update deal for the specified date

2. If Food brand requests to cancel deal for the specified date

- System terminates the use case

3.9. System Feature – Take Order

3.9.1. Description and Priority

The food brands will be able to take order from customers

3.9.2. Response Sequence

- As any customer order, the system displays the "order received" dialog with the description and quantity
- The food brand clicks on "Accept order" and system sends a confirmation message to the customer
- If due to any issue it cannot take the order, the brand clicks on "reject order" and a message is sent to the customer

3.9.3. Functional Requirements

FE1: Cannot take order (Restaurant is closed):

1. System terminates the order
2. Food brand cancels the order

FE2: Rider not available

1. System sends a message for order delay due to non-availability of rider
2. Delivery not possible:

- System terminates the order
- Food brand cancels the order

FE3: Meal is not available

1. System sends a message for order delay due to no availability of meal
2. Order is not possible:

- System terminates the order
- Food brand cancels the order

3.10. **Receive Feedback**

3.10.1. Description and Priority

This feature allows food brands to view the customers feedback and reply

3.10.2. Response Sequence

- Once the order is delivered, food brand receive customers feedback and ratings
- In case of any issues and complaints the food brand can respond to customer via comments/call/message

4. External Interface Requirements

4.1. User Interfaces

There would be two separate apps for food brands and customers

However, the app would maintain consistency and follow some standards:

- There shall be a fixed menu bar at the top with following buttons (All, Budget, Deals/discounts/Rating)
- There should be fixed drop down menu pointer at top left with following options (Profile, Help, Settings, Logout)
- On clicking the logo, the system shall return to Home Page
- There should be a “Contact us” and “Logout” buttons at bottom

The layout should be simple and easy to use

The layout should provide maximum user experience

In case of any errors, it should provide proper guidelines to resolve

4.2. Hardware and Software Interfaces

We would require following technology for development of the app:

Camera: for QR scanning

GPS receiver: indicates user location

Moreover, following technology stack can be used:

MySql database, Javascript and React for front end, PHP for backend, any framework such as Angular and server such as Apachee. A cache such as Memcached would be required to store data.

Cloud can be used for backup and retrieval of data

4.3. Communication Interfaces

The system will use following communication interfaces:

Emails

Social media

Text communication

Protocols would be required for secure communication and message encryption

5. Other non-functional Requirements

5.1. Performance Requirements

The app should provide greater performance with no delay. For food brands, who would be using web app on their desktop computers, the performance should be good and queries with minimum "join" statements are preferred for better and fast results. Too many tables in database can result in slower execution of queries, hence effecting the entire system

92% of the queries shall be completed in approximately 3.5-4 seconds. There should be no more than 0.5 – 0.8 second of delay in communication between customers and food brands.

5.2. Safety Requirements

RPO and RTO should be clearly defined to avoid loss of data that could affect the business. The system can not afford loss of data of its customers because it provides analysis on basis of it

5.3. Security Requirements

- Every user must change his initial password after first successful login

- If any user uses his credit card for payment, OTP is sent via text or call for confirmation
- The user shall receive a text message by bank on successful transaction

5.4. Software Quality Attributes

The mobile app should have a responsive layout and should be portable

The web app should be scalable and manages the data load accordingly

The mobile app should follow recognition rather than recall i.e. it is simple and easy to use and learn