Software Requirements Specification

for

Food Delivery Management

Version 1.0

Prepared by Anmol Muskan, Joshita Bolisetty, Shubhanshu Verma

Manipal Institute Of Technology, Manipal

11/05/2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1.Introduction

1.1 Purpose

Product Identification:
- Product Name: Food Delivery Management App
- Revision/Release Number: Version 1.0

Scope of the Product:
- The SRS covers the complete software requirements for the Food Delivery Management App, including the core functionalities and features.

1.2 Document Conventions

➢ Entire document should be justified.
➢ Convention for main title
• Font face: Times New Roman
• Font style: Bold
• Font size: 32
➢ Convention for Sub title
• Font face : Times New Roman
• Font Style : Bold
• Font size : 20
➢ Convention for body
• Font Face: Times New roman
• Font Size : 12

1.3 Intended Audience and Reading Suggestions

Some potential target audiences could include:

1. Restaurant Owners/Managers: Those who own or manage restaurants and want to streamline their food delivery operations.
2. Delivery Drivers: Individuals who work as delivery drivers and need a user-friendly app to manage their orders, routes, and earnings.
3. Customers: People who frequently order food for delivery and want a convenient app to track their orders, customize preferences, and provide feedback.
4. Food Delivery Aggregators: Companies or platforms that aggregate food delivery services from multiple restaurants and need an app to manage orders, logistics, and payments.

5. Small Business Owners: Entrepreneurs who run small-scale food delivery services and require an efficient app to manage their operations.

1.4 Product Scope

The food delivery management software being specified is a comprehensive application designed to streamline and enhance the operations of food delivery businesses. Its purpose is to provide a centralized platform for managing the end-to-end process of food delivery, from order placement to delivery tracking and payment processing. The software aims to improve efficiency, customer satisfaction, and overall business performance for food delivery operations.

Key Benefits:

1. Efficient Order Management: The software simplifies the process of receiving, organizing, and managing customer orders, allowing businesses to handle a large volume of orders more efficiently and accurately.
2. Seamless Communication: The software facilitates effective communication between the restaurant, delivery staff, and customers, ensuring smooth coordination and minimizing errors or delays.
3. Real-time Monitoring: The software provides real-time monitoring and reporting features that allow businesses to track delivery progress, monitor performance metrics, and identify areas for improvement.

Objectives and Goals:

The objectives and goals of the food delivery management software are aimed at improving the efficiency and effectiveness of food delivery operations. These include:

1. Streamlining Order Processing: The software aims to automate and streamline the process of receiving, organizing, and managing customer orders, reducing manual effort and potential errors.
2. Enhancing Delivery Efficiency: By optimizing delivery routes and providing real-time monitoring capabilities, the software aims to improve delivery time and increase overall operational efficiency.
3. Improving Customer Satisfaction: Through efficient order management and effective communication, the software aims to enhance the customer experience and satisfaction with the food delivery service.
4. Increasing Business Productivity: By automating manual tasks and providing actionable insights, the software aims to increase the productivity and effectiveness of the delivery staff, leading to improved business performance.

Although the software does not include order tracking, payment processing, or delivery status features as of now, it still provides significant value by simplifying order management, facilitating communication, optimizing delivery routing, and enabling real-time monitoring of operations.

1.5 References

SRS-Food Order System - Software Requirements Specification for Restaurant Food Ordering System - Studocu

(DOC) An Online Food Ordering System | Tanvir Ahmed - Academia.edu

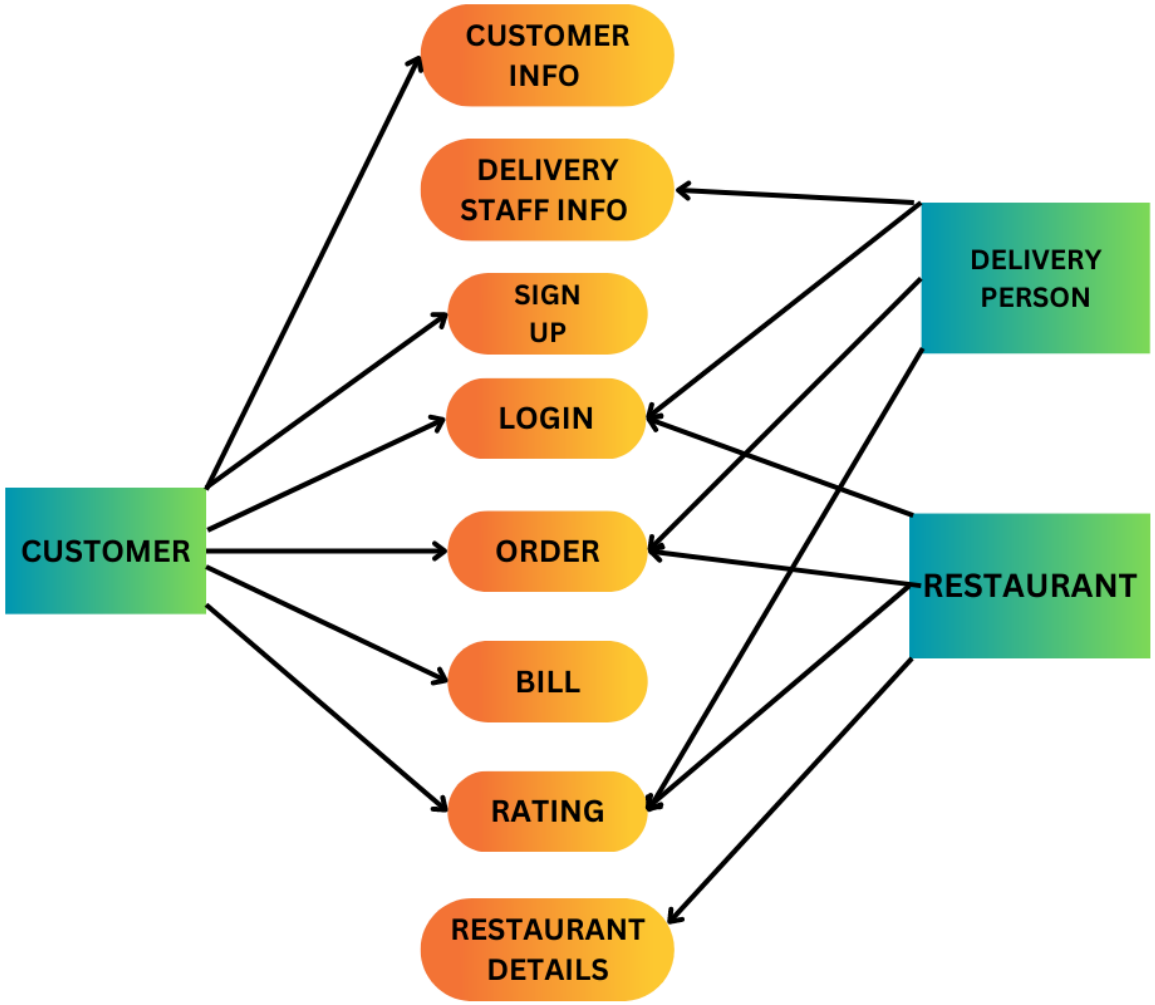
2. Overall Description

2.1 Product Perspective

The food delivery system is a self-contained product that is being developed to provide an efficient and user-friendly platform for customers, restaurant owners, and delivery personnel. The system will act as an intermediary between customers and restaurants, allowing customers to place orders and restaurants to view those orders. The delivery personnel will be responsible for delivering the food to the customers for the orders that were assigned to it.
The food delivery system is an application that will be accessible through a Windows Form application. It will be developed using modern technologies such as VC#. It will utilize a database management system to store and manage the database.
The system will be designed with scalability in mind to handle increasing user traffic and data storage requirements. The system will have redundancy mechanisms in place to ensure high availability and data integrity.

2.2 Product Functions

1. User registration: Allow users to create an account on the platform by providing basic personal information such as name, email address, and password.
2. User login: Provide a secure login system for users to access the platform by verifying their credentials.
3. Restaurant selection: Allow users to select restaurants that they want.
4. Menu browsing: Enable users to browse through the menus of various restaurants to view and select the items available and their prices.
5. Placing orders: Allow users to place orders for food items from the menus of selected restaurants.
6. Delivery person assignment: Allow auto assignment of delivery person to a specific order.
7. Rating and feedback: Allow users to rate and provide feedback on the restaurants, food items, and delivery persons.



2.3 User Classes and Characteristics

1. Customer
Characteristics: Individuals who use the food delivery service to order food from various restaurants. They may have different dietary restrictions and preferences.
Requirements: The system must allow customers to browse menus, select items, place orders, and rate restaurants and delivery personnel. The system should also provide a user-friendly interface.

2. Restaurant
Characteristics: Owners or managers of restaurants who use the food delivery service to receive and fulfill orders. Restaurants may vary in terms of size, location, type of cuisine, and technical expertise.

Requirements: The system must allow restaurants to view order history and customer ratings and feedback. The system should also provide a user-friendly interface that integrates with their existing restaurant management system.

3. Delivery Personnel
Characteristics: Individuals who deliver food orders from restaurants to customers. Delivery personnel may vary in terms of mode of transportation, and technical expertise.
Requirements: The system must allow delivery personnel to view delivery requests, navigate to the customer's location, and view the total payable amount. The system should also provide a user-friendly interface.

4. System Administrator
Characteristics: Individuals who manage and maintain the food delivery system.
Requirements: The system must allow system administrators to manage user accounts, monitor system performance, and troubleshoot technical issues. The system should also provide a secure login and authentication mechanism to prevent unauthorized access.

The most important user classes for this product are customers and restaurants, as they are the primary users who generate revenue and facilitate the delivery of food orders. Delivery personnel and system administrators are also important to ensure the smooth and efficient operation of the system.

2.4 Operating Environment

The software system of the food delivery project will operate in the following environment:

Hardware Platform:
The software system will run on devices with the following minimum hardware requirements:
- Processor: Intel Core i3 or higher
- RAM: 4GB or higher
- Storage: 256GB or higher
- Display: 1366 x 768 resolution or higher

Operating System:
The software system is designed to run on the following operating systems:
- Windows 10 or higher

Software Components:
The following software components and applications are required for the software system to operate smoothly:
- Database management system: Oracle 12c or higher

It is important that the software system operates seamlessly with the above hardware, operating system, and software components to ensure optimal performance and functionality.

2.5 Design and Implementation Constraints

The following are the design and implementation constraints for the food delivery system:

1. Technology and tools: The system will be developed using Microsoft Visual Studio 2022 and VC# programming language. The system database will be implemented using SQL Server 2019. Any other tools or technologies used in the development process must be approved by the project manager.
2. Security considerations: The system must comply with relevant security standards and regulations, including but not limited to user authentication, data encryption, and secure communication protocols.
3. Operating environment: The system must be compatible with Windows operating systems, specifically Windows 10 and above.
4. User interface: The system must follow a user-friendly interface design to ensure ease of use and accessibility for all users.
5. Performance requirements: The system must meet the performance requirements, such as response time, throughput, and availability, as specified in the SRS document. Any hardware limitations must also be taken into consideration during the development process.

2.6 Assumptions and Dependencies

Assumptions:

1. Availability of Database Connection: It is assumed that users of the food delivery management app are connected to the required database to access the application and its features.
2. User Device Compatibility: The app assumes that users will have compatible devices (such as smartphones or computers) capable of running the application.
3. Data Privacy and Security: It is assumed that appropriate measures will be taken to ensure the privacy and security of user data, including compliance with relevant data protection regulations.
4. Adequate Hardware and Infrastructure: The app assumes that the necessary hardware and infrastructure, including servers, databases, and network resources, will be available to support its operation and scalability.
5. Compliance with Local Regulations: The app assumes that it will comply with local laws and regulations related to food delivery services, including licensing requirements and health and safety standards.

Dependencies:

1. External APIs: If the app integrates with external systems or services, such as restaurant management systems or inventory management platforms, the project may depend on the availability and proper functioning of these APIs.
2. Operating System and Platform Requirements: The app may have dependencies on specific operating systems or platforms (e.g., Android, iOS, web browsers) and their compatibility with the chosen development tools and frameworks.

3. Database System: The food delivery management app depends on a reliable and scalable database system to store and manage essential data, such as user information, orders, and restaurant details.
4. Restaurant Partners: If the app collaborates with specific restaurants or food establishments, the project may depend on establishing partnerships and obtaining necessary agreements and data access from these partners.
5. User Device Compatibility: The app may have dependencies on the compatibility of user devices (smartphones, tablets, etc.) and operating systems with the software's requirements and functionalities.
6. Regulatory Compliance: The project may have dependencies on complying with local regulations and legal requirements related to food delivery services, including licensing, health and safety standards, and data protection regulations.
7. Development Tools and Frameworks: The project may depend on specific development tools, libraries, frameworks, or programming languages to build and deploy the food delivery management app.
8. Network Infrastructure: The app's performance and availability may depend on a stable and robust network infrastructure to ensure seamless communication between users, delivery personnel, and the backend systems.

   It's crucial to identify and manage these dependencies effectively throughout the project's lifecycle to ensure smooth development, integration, and operation of the food delivery management app. Regular communication and coordination with relevant stakeholders, service providers, and partners can help address any potential issues and mitigate risks associated with these dependencies.

3.External Interface Requirements

3.1 User Interfaces

The software product will have a user interface that allows users to interact with the system and perform various actions. The user interface will be designed using Windows Forms in Visual Studio 2022.

1. Customer

• The main screen will display a login form where users will enter their credentials to access the system.

• Once the user logs in, they will be directed to the main menu screen, which will provide options to navigate to different sections of the application.

• The main menu screen will have a clean and simple layout with large buttons for each section of the application, such as restaurant options and account information. Each button will have a descriptive label and an icon to help users quickly identify the function.

• In the order placement section, users will be able to select items from a menu, customize their order, and add it to their cart.

• The account information section will allow users to update their personal information such as their address.
2. Restaurant
• The restaurant section of the user interface will provide restaurants with access to view all the orders placed with them which includes detailed order information including the bill id,list of items ordered and total bill amount.
3. Delivery Personnel
• The user interface will also include a section for delivery personnel where they can view their assigned orders,customer rating and their personal details.

The user interface will follow standard Windows design guidelines and incorporate best practices for usability and accessibility. Error messages will be displayed in a clear and concise manner.

3.2 Software Interfaces

The food delivery system will use the following software components and interfaces:

1. Operating System: Windows 10 or higher
2. Development Environment: Visual Studio 2022
3. Database Management System: Oracle SQLPlus
4. .NET Framework: version 4.7 or higher
5. Oracle ODP.NET: version 19.3 or higher

The system will communicate with the Oracle SQLPlus database through the Oracle ODP.NET interface. The data items exchanged between the system and the database will include user login credentials, user details, restaurant and menu details, order details, and payment information. The system will also use .NET Framework libraries for various functionalities such as user interface, data handling, and security. The data sharing mechanism between the various software components will be implemented using the .NET Framework and ODP.NET.

4. System Features

4.1 System Feature : User Authentication and Authorization

4.1.1 Description and Priority:

This feature allows users to register for a new account, log in to their existing account, and authenticate themselves in order to access various features of the system. The priority of this feature is high since it is essential for ensuring the security and privacy of user data.

4.1.2 Stimulus/Response Sequences:

- When a new user visits the system, they will be prompted to create an account by providing their basic information, such as name, email address, and password. The system will store this information securely in the database.
- When an existing user wants to access the system, they will be prompted to enter their login credentials, such as login id and password. The system will authenticate the user's credentials by comparing them with the information stored in the database.
- If the login credentials are valid, the user will be granted access to the system and redirected to their corresponding dashboard.
- If the login credentials are invalid, the system will display an error message and prompt the user to enter their credentials again.

4.1.3 Functional Requirements:

- The system shall provide a user registration form that allows new users to create an account by providing their basic information, such as name, email address, and password.
- The system shall store user account information securely in the database.
- The system shall provide a login form that allows existing users to authenticate themselves by entering their email address and password.
- The system shall verify the user's login credentials by comparing them with the information stored in the database.
- The system shall grant access to the system and redirect the user to their corresponding dashboard upon successful authentication.
- The system shall display an error message and prompt the user to enter their credentials again upon unsuccessful authentication.

REQ-1: The customer shall be able to create an account with a unique username and password.
REQ-2: The customer shall be able to browse nearby restaurants and their menus.

System Feature 4.2: Order Placement

This feature allows customers to place an order for food delivery from a specific restaurant. It includes the ability to select menu items, specify quantities, enter delivery information, and select a payment method.

System Feature 4.3: Restaurant Management

This feature allows restaurant owners to view and manage orders placed from their restaurant.

System Feature 4.4: Delivery Management

This feature allows delivery personnel to view their delivery schedule.

System Feature 4.5: Customer Account Management

This feature allows customers to create and manage their account information ,like address,password

System Feature 4.6: Ratings and Reviews

This feature allows customers to rate and review restaurants and delivery personnel based on their experiences. It also allows restaurant owners and delivery personnel to view their own ratings over time.

5 Other Non Functional Dependencies

5.1 Performance Requirements

Response Time: The app should respond to user interactions, such as clicking a button or submitting a form, within a maximum of 2 seconds to provide a smooth and responsive user experience.

1. Order Processing Time: The time taken to process an order, from the moment it is placed to the point of confirmation, should not exceed 30 seconds. This ensures efficient order management and reduces waiting time for users.
2. System Availability: The app should be available and accessible to users at least 99% of the time, excluding scheduled maintenance periods. This ensures that users can access the app whenever they need to place orders or manage their accounts.

3. Scalability: The system should be able to handle a significant increase in user activity, such as during peak hours or promotional events, without significant degradation in performance. This includes the ability to handle a high volume of concurrent user requests without compromising response times.
4. Database Performance: Database queries and operations should be optimized to ensure fast and efficient retrieval and storage of data. For example, the database should be able to handle a large number of simultaneous read and write operations without noticeable delays.
5. Load Time: The app's initial load time should be optimized to provide a seamless user experience. It should load and display essential content within 3 seconds to minimize user frustration and improve engagement.
6. Data Synchronization: If the app operates in an offline mode and later synchronizes data with the server, the synchronization process should be efficient and completed within a reasonable timeframe to ensure data consistency and avoid conflicts.

These performance requirements aim to ensure that the food delivery management app delivers a fast, reliable, and responsive user experience, even during peak usage periods.

5.2 Safety Requirements

1. User Privacy and Data Protection: The app should comply with relevant data protection regulations and safeguard user privacy. This includes ensuring secure storage and transmission of personal information, implementing appropriate authentication and access controls, and preventing unauthorized access or data breaches.
2. Food Safety and Hygiene: If the app includes features related to food delivery, it should adhere to applicable food safety standards and regulations. This may involve ensuring that partner restaurants comply with food handling and preparation guidelines, maintaining proper temperature control during delivery, and providing information on allergens or dietary restrictions.
3. Secure Payments: The app should employ secure payment methods to protect users' financial information. This may involve utilizing encryption technologies, complying with Payment Card Industry Data Security Standard (PCI DSS) requirements, and integrating with trusted payment gateways.
4. Anti-Fraud Measures: The app should incorporate measures to prevent fraudulent activities, such as identity theft, payment fraud, or order manipulation. This may involve implementing fraud detection algorithms, monitoring suspicious activities, and providing mechanisms for users to report fraudulent incidents.
5. Delivery Personnel Safety: If the app involves delivery personnel, it should include safety measures to protect their well-being. This may include providing guidelines for safe driving practices, encouraging adherence to traffic rules and regulations, and promoting proper communication and behavior protocols during deliveries.
6. Compliance with Local Regulations: The app should comply with applicable laws, regulations, and policies related to food delivery services, data protection, privacy, and other safety-related aspects. This may include adhering to local health and safety standards, obtaining necessary licenses or permits, and complying with employment laws.

7. Accessibility: The app should strive to be accessible to users with disabilities, following accessibility guidelines and standards. This includes providing features such as screen reader compatibility, adjustable font sizes, and alternative text for images.
8. Safety Certifications: If there are specific safety certifications relevant to the food delivery management app, such as certifications related to payment security or data protection, the app should satisfy the requirements of those certifications.

These safety requirements are meant to ensure the well-being, privacy, and security of users and all involved stakeholders. It is crucial to refer to external policies, regulations, and industry standards that dictate safety issues affecting the app's design and use. Compliance with relevant safety regulations and certifications demonstrates the app's commitment to maintaining a safe and secure environment for its users and stakeholders.

5.3 Security Requirements

➢ System will use secured database.
➢ Normal users can just read information but they cannot edit or modify anything except their personal details.
➢ System will have different types of users and every user has access constraints.
➢ Proper user authentication should be provided
➢ There should be separate accounts for admin and members such that no member can access the database and only the admin has the rights to update the database.

5.4 Software Quality Attributes

The following quality attributes are important for the software product:

1. Usability: The system should be easy to use and navigate, with clear and concise instructions for all functions. It should have a low learning curve and allow users to complete tasks quickly and efficiently.
2. Reliability: The system should be reliable and operate without errors or unexpected behavior.
3. Security: The system should be designed with security in mind, with measures in place to prevent unauthorized access, protect sensitive information, and ensure data privacy.
4. Maintainability: The system should be designed in a modular and scalable manner, with clean code and documentation that makes it easy for developers to maintain and modify the system as needed.
5. Performance: The system should be able to process user requests quickly and efficiently, with low response times and minimal delays.
6. Scalability: The system should be able to handle a growing number of users and requests without performance degradation or system failure.

These attributes will be evaluated and measured through various testing methods such as functional testing, load testing, security testing, and user acceptance testing.

5.5 Business Rules

1. User Registration:
   - Only customers who have registered and created an account can place orders through the app.
   - Customers must provide valid and accurate personal information during the registration process.
   - Users may not have multiple active accounts unless explicitly allowed by the system.

2. Restaurant Partner Approval:
   - Restaurants wishing to be listed on the app must go through an approval process by the app administrator.
   - Approved restaurants can manage their menu, update prices, and accept or reject orders.

3. Order Placement and Modification:
   - Customers can place an order through the app by selecting available menu items from the listed restaurants.
   - Customers may modify or cancel their orders within a specified time frame before it is being prepared.

4. Delivery Assignment:
   - Delivery personnel are assigned orders based on their availability, distance, and workload.
   - The assignment algorithm considers factors such as delivery time, traffic conditions, and order priority.

5. Delivery Time Estimates:
   - The app provides estimated delivery times based on factors such as the restaurant's preparation time, distance, and current traffic conditions.
   - Delivery time estimates are adjusted in real-time to account for changes in traffic or order status.

6. Ratings and Reviews:
   - Customers can provide ratings and reviews for restaurants and delivery personnel based on their experience.
   - The system may have a mechanism to monitor and moderate ratings and reviews to maintain authenticity and prevent misuse.

7. Payment Processing:
   - Customers must make payment for their orders through approved payment methods before delivery.
   - The app may support multiple payment options, such as credit cards, mobile wallets, or cash on delivery, based on availability and user preferences.

8. Notifications and Alerts:
   - Users may receive notifications and alerts regarding order status updates, promotional offers, or important announcements.
   - The system may have rules to control the frequency and content of notifications to avoid overwhelming users.

These business rules help define the operational guidelines and user interactions within the food delivery management app. The functional requirements of the system should align with and enforce these rules to ensure consistent and compliant usage of the application.

6. Other Requirements

1. Database Requirements:
The system will require a database management system to store and retrieve data. The database system must be compatible with the chosen programming language (C#) and integrate seamlessly with the software application.

2. Internationalization Requirements:
The software application must be designed to support multiple languages and character sets. This will allow the application to be used by users from different countries and regions.

3. Legal Requirements:
The software application must comply with all relevant laws and regulations, including data protection laws and copyright laws. Any third-party software components used in the application must be licensed appropriately.

4. Performance Requirements:
The system should be designed to handle a large volume of users and transactions without experiencing performance issues. The response time for any user action should be less than two seconds, and the system should be able to handle a large number of concurrent users.

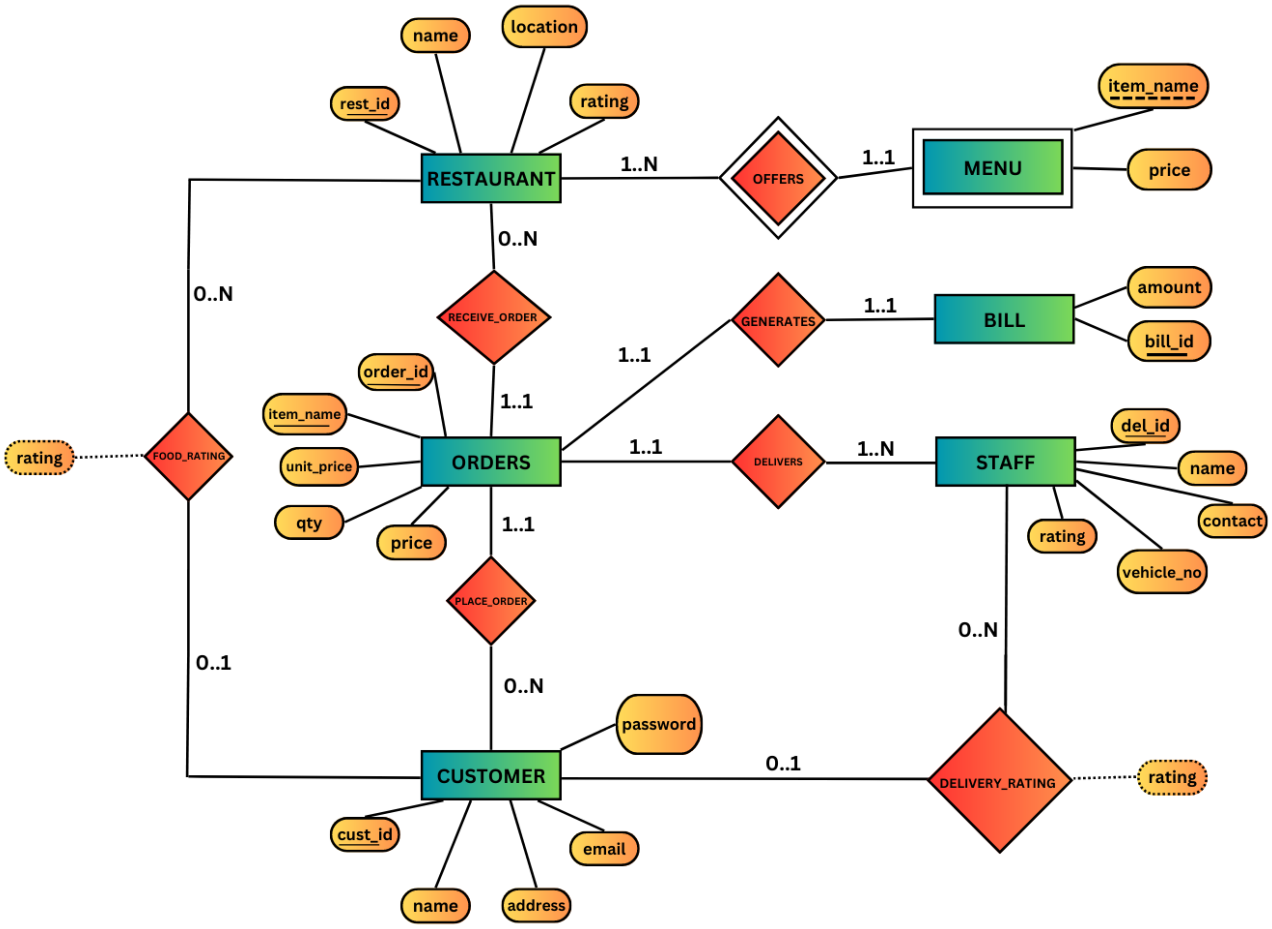
5. Reusability Objectives:
The system should be designed to be modular and reusable. This will allow the system to be easily modified and extended in the future as new requirements arise.

Appendix A: Glossary

- DBMS: Database Management System
- GUI: Graphical User Interface
- TBD: To be determined
- API: Application Programming Interface
- SMS: Short Message Service

Appendix B: Analysis Models

ER Diagram



Appendix C: To Be Determined List

1. User Documentation Components - user manuals, on-line help, and tutorials, that will be delivered along with the software.
2. Geolocation Services - for calculating distances, generating delivery routes, or providing location-based services, and lice tracking it may depend on external geolocation services or APIs.
3. SMS or Email Gateway - to send notifications or alerts to users or delivery personnel on integrating with an SMS or email gateway service to facilitate communication.
4. Payment Gateway Integration - for smooth and cashless online payments.

These TBD references should be tracked and resolved by the appropriate stakeholders to ensure the completion and accuracy of the SRS.