SOFTWARE REQUIREMENTS SPECIFICATION

For

Food delivery website and application

Done by
D. Teena
S. Vanmathi
S. Sangavi

Academic Year: 2022-2023

1. Introduction

1.1 Purpose

The main objective of this document is to illustrate the requirements of the project Food delivery system. The document gives the detailed description of the both functional and non-functional requirements proposed by the client. The purpose of this project is to provide a friendly environment to maintain the details of foods and customer members. The main purpose of this project is to maintain easy circulation system using computers and to provide different reports. This project describes the hardware and software interface requirements using ER diagrams and UML diagrams.

1.2 Document Conventions

- Entire document should be justified.
- Convention for Main title
  - Font face: Times New Roman
  - Font style: Bold
  - Font Size: 14
- Convention for Sub title
  - Font face: Times New Roman
  - Font style: Bold
  - Font Size: 12
- Convention for body
  - Font face: Times New Roman
  - Font Size: 12

1.3 Scope of Development Project

Food Delivery System is basically designing user-friendly for customers, delivery drivers into an internet-based application so that the customers can know the details of their accounts, availability of foods and placing.

The project is specifically designed for the use of a food delivery service and its users. The product will serve as a comprehensive user interface for managing food orders, delivery processes, and user interactions for both customers and delivery personnel. This Food Delivery Application can be adopted by any existing or new food delivery business to efficiently manage their food offerings, order processing, delivery tracking, and menu updates. It is particularly valuable for any restaurant or food delivery service where customization of the menu and features is essential. The application can be seamlessly implemented in various scenarios, and new features can be added as needed, ensuring reusability due to the flexibility of all the modules.

1.4 Definitions, Acronyms and Abbreviations

JAVA -> platform independence
SQL-> Structured query Language
ER-> Entity Relationship
UML -> Unified Modeling Language
IDE-> Integrated Development Environment
SRS-> Software Requirement Specification
SBN -> International Standard Book Number
IEEE ->Institute of Electrical and Electronics Engineers

1.5 References

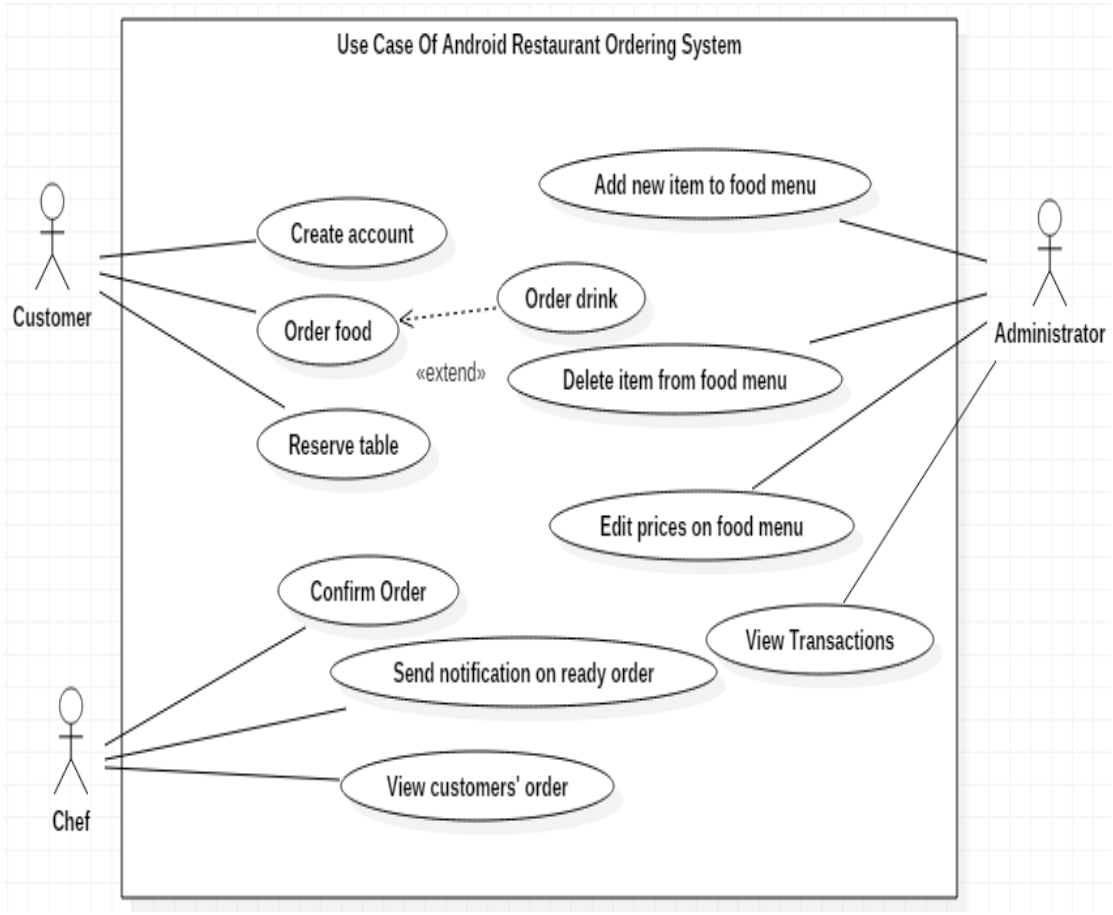
- Books
  - Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices (ACM Press) by Michael Jackson
  - Software Requirements (Microsoft) Second Edition by Karl E. Wiegers
  - Software Engineering: A Practitioner's Approach Fifth Edition by Roger S. Pressman

- Websites: **http://www.slideshare.net/**

2. Overall Descriptions

2.1 Product Perspective

Use Case Diagram of Food Delivery System



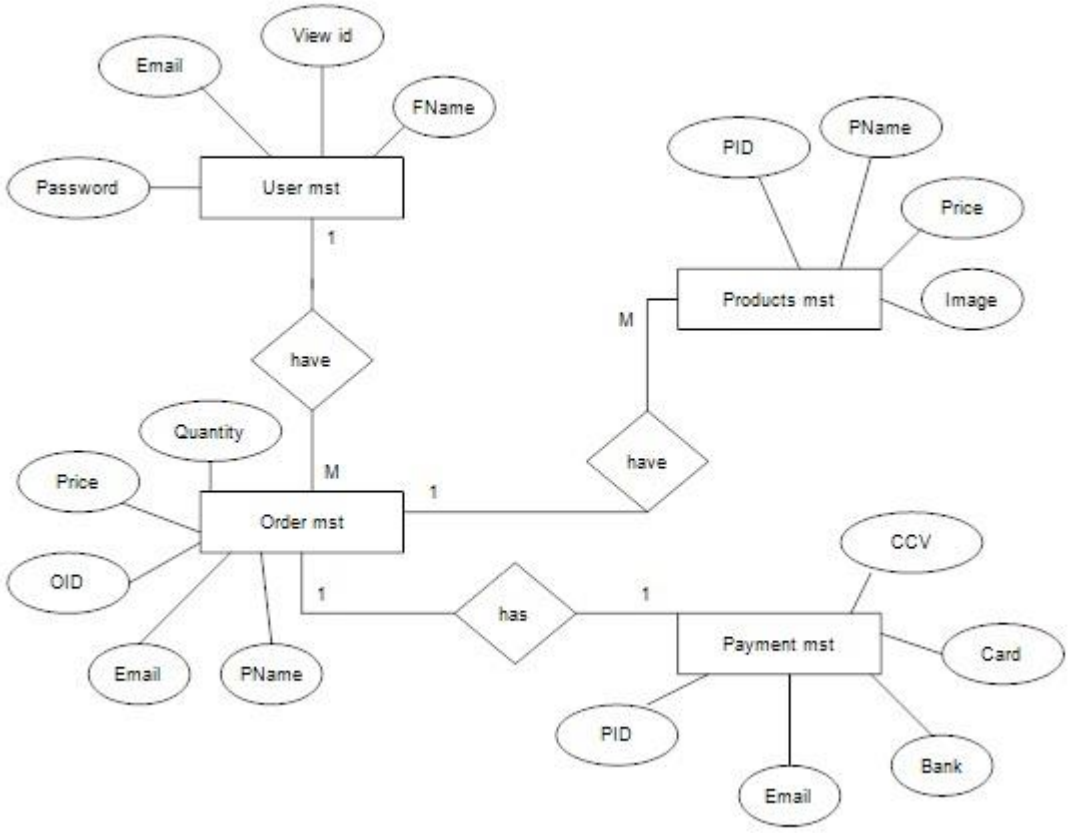
Customer: The primary actor in the system who interacts with it to place food orders, track deliveries, and manage their account.
Restaurant Staff: Actors representing employees at partner restaurants who receive and process food orders.

Delivery Driver: Actors representing the delivery personnel responsible for delivering orders to customers.

Administrator: An actor responsible for managing the system, including user accounts, menus, and order processing.

2.2Product Function

Entity Relationship Diagram of Library Management System



The Food delivery System provides online real-time information about the foods available in the restaurant and the user information. The main purpose of this project is to reduce the manual work. This software is capable of managing food deliveries, Issues, Returns, Calculating/Managing Fines, and Generating various Reports for Record-Keeping according to end user requirements. The chef will act as the administrator to control members and manage orders. The member's status of issue/return is maintained in the restaurant database. The member's details can be fetched by the restaurant from the database.

2.3User classes and classification:

The system provides different types of services.

2.4Types of users [Customer/Administrator].

The Administrator will be acting as the controller and he will have all the privileges of an administrator. The member can be either a customer or user of the restaurant who will be accessing the restaurant online.

The features that are available to the Administrator are:-

- A chef can issue a book to the customer.
- Can view the different categories of foods available in the Restaurant.
- Can view the List of foods available in each category.
- Can take the food returned from customer.
- Add foods and their information to the database
- Edit the information of existing foods
- Can check the report of the existing foods
- Can check the report of the issued foods
- Can access all the accounts of the customer

The features that are available to the customers are:-

- Can view the different categories of foods available in the Restaurant
- Can view the List of foods available in each category
- Can own an account in the restaurant.
- Can view the foods issued to him
- Can put a request for a new food
- Can view the history of foods issued to him previously
- Can search for a particular food.

2.5 Operating Environment

The product will be operating in windows environment. The Food Delivery System is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox. Also it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla Firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection. The hardware configuration include Hard Disk: 40 GB, Monitor: 15" Color monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

2.6Assumptions and Dependencies

The dependencies are:-

The coding should be error free

- The system should be user-friendly so that it is easy to use for the users
- The information of all users, foods and restaurant must be stored in a database that is accessible by the website
- The system should have more storage capacity and provide fast access to the database

- The system should provide search facility and support quick transactions
- The Restaurant System is running 24 hours a day
- Users may access from any computer that has Internet browsing capabilities and an Internet connection.
- Users must have their correct usernames and passwords to enter into their online accounts and do actions

The dependencies are:-

- The specific hardware and software due to which the product will be run
- The system should have the general report stored
- On the basis of listing requirements and specification the project will be developed and run
- The end users (admin) should have proper understanding of the product
- The information of all the users must be stored in a database that is accessible by the Restaurant System recorded to the database and the data entered should be correct

2.7Requirement

Software Configuration:-

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database.
Operating System: Windows NT, windows 98, Windows XP
Language: Java Runtime Environment, Net beans 7.0.1 (front end)
Database: MS SQL Server (back end)

Hardware Configuration:-

Processor: Pentium(R)Dual-core CPU
Hard Disk: 40GB
RAM: 256 MB or more

2.8Data Requirement

The inputs consist of the query to the database and the output consists of the solutions for the query. The output also includes the user receiving the details of their accounts. In this project the inputs will be the queries as fired by the users like create an account, selecting foods and putting into account. Now the output will be visible when the user requests the server to get details of their account in the form of time, date and which foods are currently in the account.

3 External Interface Requirement

User Interface (UI) Design

- The UI should be intuitive, user-friendly, and responsive across various devices (desktops, tablets, smartphones).
- Implement a visually appealing design with a clean layout, clear navigation, and consistent color schemes and branding.

User Registration and Authentication

- Users should be able to register, log in, and recover/reset their passwords.
- Provide authentication mechanisms (e.g., email verification, password encryption).

Restaurant Listings and Details

- Display a list of restaurants with key information (name, cuisine type, ratings, delivery time, distance).
- Allow users to click on a restaurant to view detailed information (menu, reviews, images).

Search and Filter Functionality

- Users should be able to search for restaurants based on location, cuisine, ratings, and price range.
- Implement advanced filtering options to refine search results.

Menu Presentation

- Each restaurant should have a clear menu presentation with item names, descriptions, prices, and images.
- Include options for customizations (e.g., toppings, quantity adjustments).

Ordering Process

- Users should be able to add items to a shopping cart, review their order, and proceed to checkout.
- Provide a seamless checkout process with options for payment

Real-Time Order Tracking

- Implement a real-time order tracking system that updates users on the status of their order (accepted, preparing, on the way, delivered).

User Profiles

- Users should have a profile page where they can view and edit personal information, track order history, and manage saved addresses.

Notifications and Alerts

- Send notifications and alerts to users for order confirmations, status updates, and promotions/offers.

Reviews and Ratings

- Allow users to rate and leave reviews for restaurants and individual dishes.
- Display an average rating for each item.

Customer Support

- Provide accessible customer support options (chat, email, phone) within the application.

Accessibility and Localization

- Ensure that the interface complies with accessibility standards to accommodate users with disabilities.
- Support multiple languages and currencies for a global user base.

Error Handling and Validation

- Implement proper error handling and validation messages to guide users in case of incorrect inputs or system errors.

Security

- Utilize secure authentication mechanisms and encryption techniques to protect user data and payment information.

Performance Optimization

- Optimize the application for fast loading times and smooth user experience.

Integration with Maps and Location Services

Use geolocation to allow users to set their delivery location and provide accurate delivery estimates.

Technology Stack

- Specify the use of Node.js for backend development and React.js for frontend development.

4 System Features

Users of the food delivery system should have confidence in the security of their accounts. This can be ensured by implementing the following measures:

User Authentication and Member Validation:

Users must authenticate themselves with unique login credentials.

Members should use their unique customer ID to access the system.

Proper Accountability:

User accounts should be designed to prevent one user from accessing another user's account .

Only the system administrator should have access to view and manage all customer accounts .

5 Other Non-functional Requirements

5.5Performance Requirement

The website and application should load within 3 seconds on 4G network connections.

The system should support a minimum of 1000 concurrent users without significant performance degradation.

The server response time for any request should be less than 200 milliseconds.

5.6 Security Requirement

User data, including personal information and payment details, should be encrypted using industry-standard encryption protocols.

The application should have protection against security threats such as SQL injection, cross-site scripting, and cross-site request forgery.

Access to sensitive information should be role-based and follow the principle of least privilege.

5.4 Requirement Attributes

All requirements should be tagged with priority levels (e.g., high, medium, low) and dependencies on other requirements.

Each requirement should have a unique identifier for easy reference and tracking.

5.5 Business Rules

Only registered users can place orders.

Vendors should be able to update their menu and prices during specific hours, and changes should be reflected immediately.

Customers should be able to rate and review their orders.

5.6 Usability and User Experience (UX)

The user interface should be intuitive and easy to navigate, ensuring a positive experience for users of all technical levels.

The website should provide clear instructions and feedback messages to guide users through the ordering process.

5.7 Cross-Browser Compatibility

- The application should be compatible with the latest versions of popular web browsers (e.g., Chrome, Firefox, Safari, Edge).

5.8 User Requirement

- Users should be able to create accounts using their email address or social media profiles.
- Users should have the ability to save multiple delivery addresses.
- Users should receive order confirmation with detailed information.

6 Other Requirements

6.1 Data and Category Requirement

There are different categories of users namely teaching staff, Librarian, Admin, students etc. Depending upon the category of user the access rights are decided.It means if the user is an administrator then he can be able to modify the data,delete, append etc. All other users except the Librarian only have the rights to retrieve the information about database. Similarly there will be different categories of books available. According to the categories of books their relevant data should be displayed. The categories and the data related to each category should be coded in the particular format.

6.2 Appendix

A: Admin, Abbreviation, Acronym, Assumptions; B: Books, Business rules; C: Class, Client, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; L: Library, Librarian; M: Member; N: Non-functional Requirement; O: Operating environment; P: Performance, Perspective, Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, System features; U: User, User class and characteristics, User requirement;

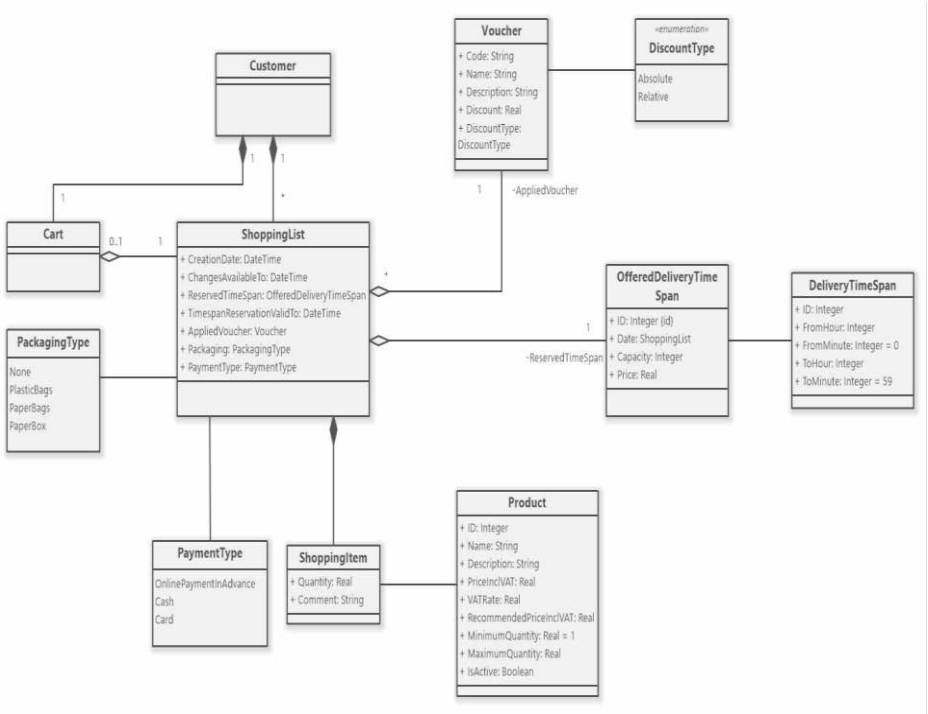
6.3 Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

- Administrator: A login id representing a user with user administration privileges to the software
- User: A general login id assigned to most users
- Client: Intended users for the software
- SQL: Structured Query Language; used to retrieve information from a database
- SQL Server: A server used to store data in an organized format
- Layer: Represents a section of the project
- User Interface Layer: The section of the assignment referring to what the user interacts with directly

- Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
- Data Storage Layer: The section of the assignment referring to where all data is recorded
- Use Case: A broad level diagram of the project showing a basic overview
- Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system's cases, their attributes, and the relationships between the classes
- Interface: Something used to communicate across different mediums
- Unique Key: Used to differentiate entries in a database

6.4Class Diagram



A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class.

The classes' structure and their relationships to each other frozen in time represent the static model. In this project there are certain main classes which are related to other classes required for their working. There are different kinds of relationships between the classes as shown in the diagram like normal association, aggregation, and generalization. The relationships are depicted using a role name and multiplicities. Here 'Administrator', 'Customers' and 'Foods' are the most important classes which are related to other classes.