Adhiparasakthi Engineering College

SOFTWARE REQUIREMENTS SPECIFICATION

For

Food Delivery Website and Application

Prepared by:-

Jayalakshmi
Nandhini
Divyadharshini
Priyadharshini

Academic Year: *2023-2024*

Department of Information Technology

1. Introduction

Purpose

The main objective of this document is to illustrate the requirements of the project Food Delivery Website and Application. The document gives the detailed description of the both functional and non-functional requirements proposed by the customer. The main purpose of a food delivery application is to facilitate the ordering and delivery of food from restaurants to customers' locations. These apps serve as a convenient platform for customers to browse menus, place orders, make payments, and have food delivered to their doorstep or a designated location.

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

Scope of Development Project

The Scope of the project (Android / web based panel application) are as follows: Food Ordering app can sale Food product, preferred brands, kitchen needs, essential restaurant supplies and more, through this online, one- stop Food store. It provides you with a convenient way to sale from your Food shopping app.

Online food ordering system is necessary for food businesses to manage all work load and to run process smoothly. So the main purpose of an online ordering system is to provide customers for a way to place an order at a restaurant over the internet.

Article Talk. Retail food delivery is a courier service in which a restaurant, store, or independent food-delivery company delivers food to a customer. An order is typically made either through a restaurant or grocer's website or mobile app, or through a food ordering company.

Definitions, Acronyms and Abbreviations

HTML -> Hyper Text Markup Language
CSS -> Platform Independent
JavaScript -> Platform Independent
UML -> Unified Modeling Language
SRS -> Software Requirement Specification

2.Overall Descriptions

Product Perspective

Use Case Diagram of Food Delivery Website and Application.

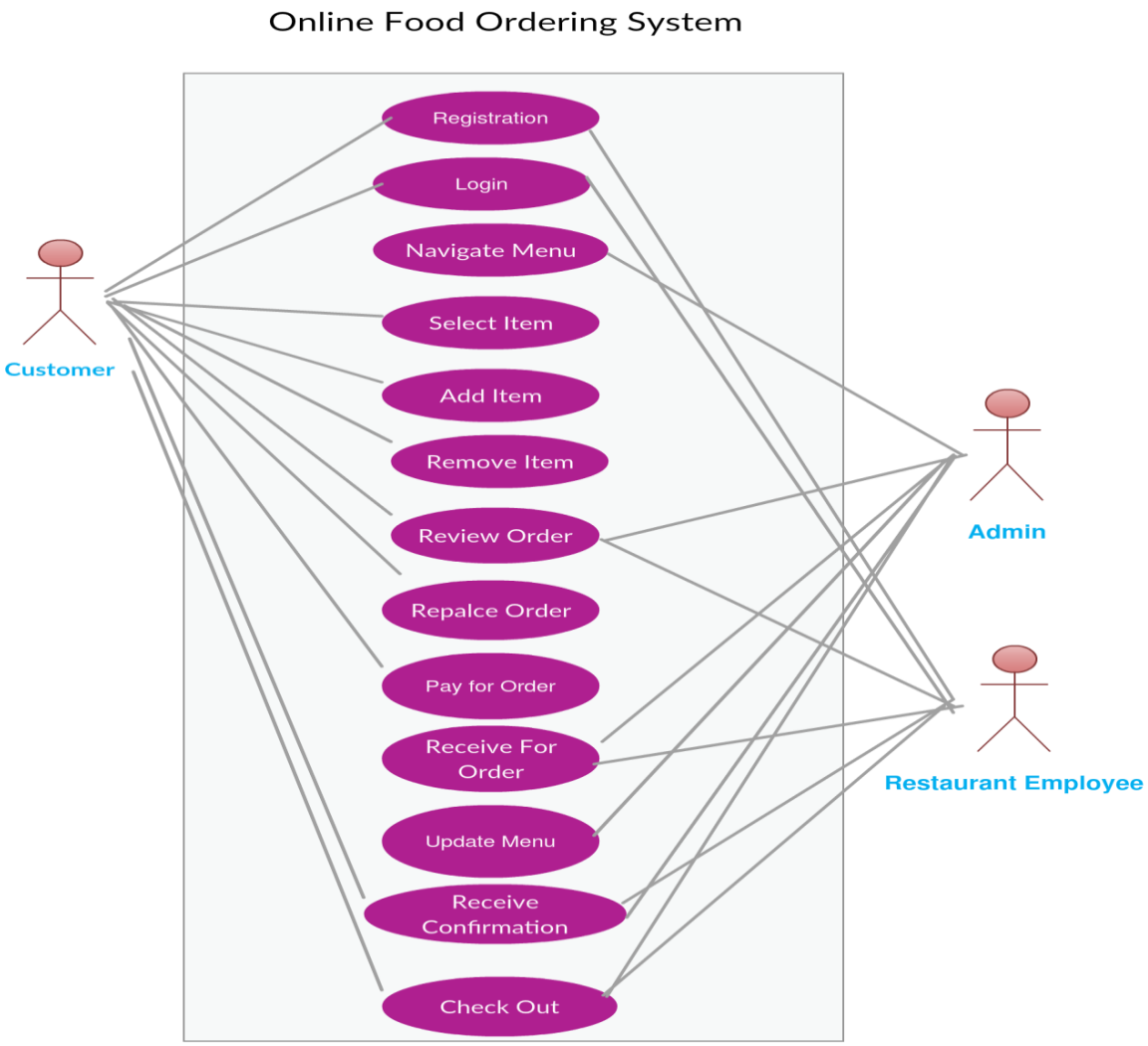
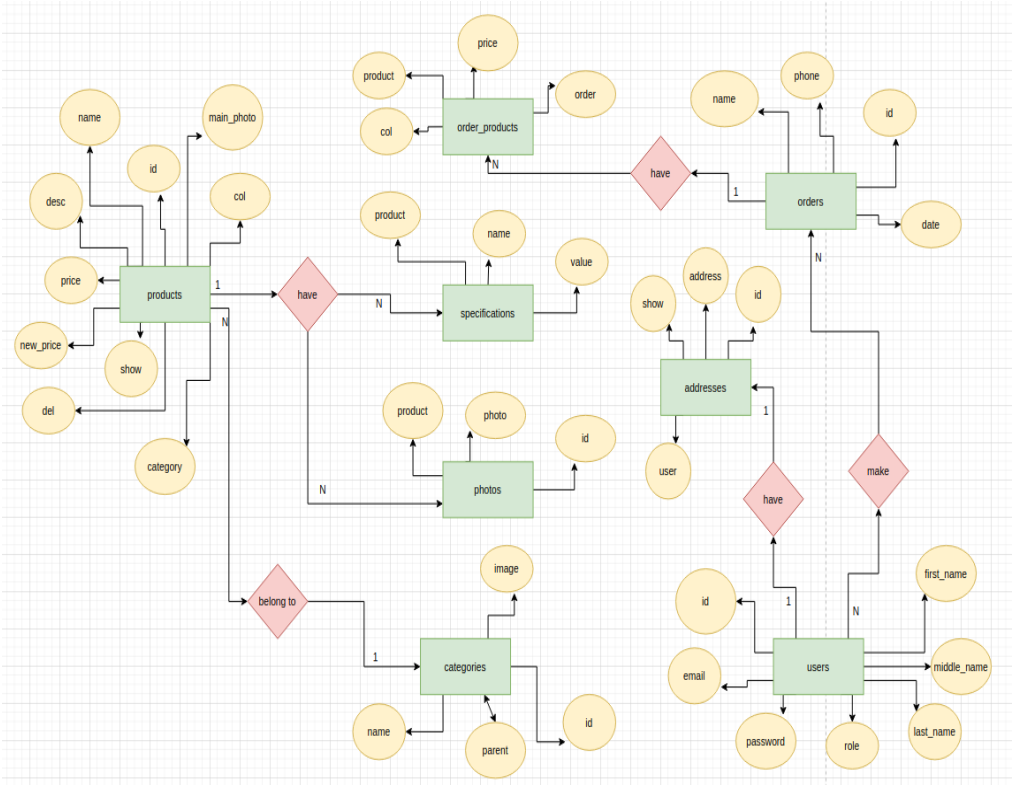


Figure : Use Case Diagram

A use case diagram is a visual representation in the Unified Modeling Language (UML) that depicts the interactions between various actors (external entities, such as users or systems) and a software system or application. It illustrates the different ways in which actors interact with the system to achieve specific goals or perform tasks. Use case diagrams are a fundamental tool in software engineering and system design for capturing and communicating the functional requirements and behavioral aspects of a system.

Product Function

Entity Relationship Diagram of Library Management System



An entity diagram, also known as an entity-relationship diagram (ERD), is a visual representation of the entities (objects, concepts, or things) within a system, their attributes (properties or characteristics), and the relationships between them. It is a widely used tool in database design and software engineering to model the structure and organization of data in a database system or application.

User Classes and Characteristics

The system provides different types of services based on the type of users [Customer/Manager]. The Restaurant Manager will be acting as the controller

and he will have all the privileges of an administrator. The member can be any people of the surrounding area who will be accessing the Restaurant online.

The features that are available to the Food Delivery Website and Application are:

- User Registration and Authentication
- User Profiles
- Restaurant Listings
- Menu and Ordering
- Online Payments
- Order Tracking
- Ratings and Reviews
- Notifications
- Delivery Options
- Customer Support
- Admin Panel
- Analytics and Reporting
- Security and Privacy
- Scalability and Performance
- Localization
- Integration

The features that are available to the Customer are:

- User Registration and Authentication
- Search and Browse Restaurants
- Menu Exploration
- Order Placement
- Checkout and Payment
- Order Tracking
- Order History and Reorder
- User Profile Management
- Rating and Reviews
- Customer Support
- Language and Accessibility
- Notifications and Alerts
- Social Sharing
- Recommendations and Personalization
- Promotions and Loyalty Programs
- Feedback and Surveys
- Order Cancellation and Refunds
- Privacy and Security
- Terms and Conditions
- Geolocation Services
- Multi-Platform Support

Operating Environment

The product will be operating in windows environment. The Food Deliver Website and Application is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer, Google

Chrome, and Mozilla Firefox. 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 basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

Assumptions and Dependencies

The assumptions are:-

- Users will continue to use smartphones for food ordering.
- There will be a constant influx of restaurants willing to partner with the platform.
- Users prefer platforms that are easy to navigate.
- Timely deliveries are essential for user satisfaction.
- Maintaining food safety standards is vital for the platform's reputation.
- The rating system is a valuable tool for user decision-making.
- Promotions will continue to be a strong incentive for orders.
- Maintaining robust payment security is a priority.
- Growth in the number of delivery zones is a strategic goal.
- Users may increasingly choose eco-conscious options if available.
- Data privacy regulations will continue to evolve, impacting the platform's operations.
- AI algorithms will be used to personalize recommendations and optimize delivery routes.
- Ongoing technological advancements will drive the need for constant innovation to stay competitive.

The dependencies are:-

- Depend on a diverse range of restaurants to offer menu choices to users.
- Implement secure user registration and login.
- Connect with payment gateways like PayPal, Stripe, or others for secure transactions.
- Create and update restaurant menus with categories, items, and prices.
- Real-time order tracking and status updates.
- Implement notifications for order updates, promotions, and delivery status.
- Allow customers to rate and review restaurants and delivery services.
- Provide robust search and filtering options for cuisine types, price range, and dietary.
- Allow users to sign in with social media accounts and share their food orders.

Requirement

Software Configuration:-

This software package is developed using HTML, CSS, JavaScript as front

end. Microsoft MongoDB as the back end to store the database.
Operating System: Windows
Database: MongoDB(back end)

Data 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 by the users like create an account, ordering food 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

GUI

1.User Interfaces:

Customer Interface: Specify the user interface elements for customers, including screens for browsing restaurants, placing orders, tracking deliveries, and managing user profiles.

Restaurant Interface: Describe the interface for restaurant partners to manage their menus, view and process orders, and adjust their availability.

Delivery Driver Interface: Outline the user interface for delivery drivers, including screens for accepting orders, navigating to delivery locations, and updating order statuses.

2. Web Interfaces (if applicable):

User Dashboard: Define the web-based dashboard for users to access their accounts, view order history, and interact with customer support.

Administrator Console: Specify the web-based interface for system administrators to monitor and manage the entire food delivery system, including user accounts, restaurant partnerships, and order processing.

3. Navigation and Usability:

Navigation Guidelines: Provide guidelines on how users can navigate through the GUI, including menus, buttons, and links.

Consistency: Ensure a consistent look and feel across different screens and modules of the application to enhance user familiarity.

4. Accessibility:

Accessibility Features: Detail the accessibility features implemented in the GUI to make the application usable by individuals with disabilities, adhering to standards like WCAG.

Keyboard Shortcuts: If applicable, specify keyboard shortcuts to assist users who may not use a mouse.

5. Multilingual and Multi-currency Support:

Language Options: Describe how the GUI supports multiple languages to cater to users in different regions.

Currency Conversion: Explain how the GUI handles multiple currencies and currency conversion for international users.

6. Third-Party Integrations:

Payment Gateways: Specify how the GUI interfaces with payment gateways for

secure and seamless online transactions.

Mapping and Location Services: Describe how location-based features, such as order tracking, are integrated into the GUI.

Communication Services: Define the integration with communication services like SMS and email for sending notifications and updates.

7. User Authentication:

Login and Registration: Provide details on how users authenticate themselves, including options such as email, social media, or phone number verification.

Forgot Password: Explain the process for users to reset their passwords.

8. Notifications and Alerts:

In-app Notifications: Specify how notifications are presented within the GUI (e.g., pop-ups, banners, alerts).

Email and SMS Notifications: Describe how users receive notifications through external channels.

9. Error Handling:

Error Messages: Define the format and content of error messages displayed to users.

User Support: Explain how users can seek assistance or report issues through the GUI.

10. API Integration:

If the GUI interacts with external APIs, provide information on the integration, including API endpoints, data exchange formats, and authentication mechanisms.

11. Compliance and Security:

Ensure that the GUI complies with relevant security standards and regulations, especially if it involves handling sensitive user data or financial transactions.

These external interface requirements for a GUI should be documented comprehensively to guide the design and development of the user interface, ensuring a user-friendly and reliable experience for the application's users while meeting any regulatory or industry-specific standards.

4.System Features

- An interface for restaurants to manage their menus, pricing, and availability.
- Order management and tracking.
- Tools for restaurants to update their inventory in real-time.
- Instant notifications of new orders via app, email, or SMS.
- Sales data, order history, and performance analytics.
- Insights to optimize menu and pricing.
- Ability to add images, descriptions, and modify menu items.
- Set minimum order amounts and delivery zones.

5.Other Non-functional Requirements

Performance:

Ensure fast response times and scalability to handle increased traffic.

Optimize database queries and minimize loading times.

Security:

Implement data encryption, user authentication, and authorization mechanisms.
Protect against common web vulnerabilities (e.g., SQL injection, cross-site scripting).

Usability:

Design an intuitive and user-friendly interface.Ensure a seamless user experience on both the website and mobile app.

Reliability:

Set up backup and disaster recovery plans to minimize service disruptions.
Ensure data integrity and reliability of payment processing.

Compliance:

Adhere to data protection and privacy regulations (e.g., GDPR).Comply with local food safety laws and regulations.

Technology Stack:

Specify the technologies, frameworks, and tools to be used for development.

Scalability:

Design the platform to handle increasing user and order volumes.

6.Other Requirements

User-Friendly Design:

- Intuitive and responsive user interface (UI) and user experience (UX) design.
- Support for various screen sizes and devices, including mobile phones and tablets.

Geolocation Services:

- Integration with GPS for accurate location tracking.
- Real-time location sharing for delivery tracking.

Payment Integration:

- Secure payment gateways to facilitate online payments.
- Multiple payment options, including credit/debit cards, digital wallets, and cash on delivery.

Menu Management:

- Easy-to-update menus for restaurants.
- Options for categorizing items, adding descriptions, and setting prices.
- Support for special deals, discounts, and promotions.

Order Management:

- Efficient order processing and confirmation.
- Real-time notifications for order status updates.
- Order history and order tracking.

Delivery Logistics:

- Integration with a reliable map and navigation service.
- Features for delivery route optimization.
- Delivery time estimation and tracking.

Glossary

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

- Administrator: A login id representing a user with user administration privileges to thesoftware
- 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 interactswith directly
- Application Logic Layer: The section of the assignment referring to the Web Server. Thisis 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

Class 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 ofrelationships 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 'Librarian', 'Member' and 'Books' are the most important classes which are

related to other classes.