**SOFTWARE**

**REQUIREMENTS SPECIFICATION**

**For**

# FOOD DELIVARY WEBSITE AND APPLICATION

## Prepared by:-

***MANIMEGAN T***

***DINESH KUMAR B***

***KAMALESH R***

***SIVAKUMAR M***

**Academic Year:** *2023-2024*

### 1. Introduction

#### 1.1 Purpose

The purpose of this document is to provide a comprehensive description of the requirements for the Food Order Application, including its features, functionality, and constraints. 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

#### We expect the food delivery app industry to grow significantly over the next five years. According to a recent report, the global food delivery market is projected to reach $365 billion by 2026, growing at a CAGR of 9.9% during the forecast period from 2020 to 2026.

#### 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

ISBN -> 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 EditionBy Karl E. Wiegers



Software Engineering: A Practitioner’s Approach Fifth Edition By Roger S. Pressman

* Websites
* [**http://www.slideshare.net/**](http://www.slideshare.net/)

**http://ebookily.net/doc/srs**

### 2. Overall Descriptions

#### 2.1 Product Perspective

#### Use Case Diagram of food delivery website and application;A diagram of a restaurant employee Description automatically generatedcation

#### FIG; Use Case Diagram of food delivery website and application.

#### Actors:

#### Customer/User

#### Restaurant Owner/Manager

#### Delivery Driver

#### Admin/Support

#### System Administrator/Developer

#### Marketing and Promotions Manager

#### Use Cases:

#### Search for Restaurants/Menu Items

#### Actor: Customer

#### Description: Customers can search for restaurants or specific menu items.

#### Place Order:

#### Actor: Customer

#### Description: Customers can place food orders from selected restaurants.

#### Manage Restaurant Information:

#### Actor: Restaurant Owner/Manager

#### Description: Restaurant owners/managers can update restaurant details and menu items.

#### Receive and Confirm Orders:

#### Actor: Restaurant Owner/Manager

#### Description: Restaurant owners/managers receive and confirm incoming orders.

#### Deliver Orders:

#### Actor: Delivery Driver

#### Description: Delivery drivers deliver orders to customers.

#### Manage User Accounts:

#### Actor: Admin/Support

#### Description: Admin/support staff can manage user accounts and access.

#### System Maintenance and Updates:

#### Actor: System Administrator/Developer

#### Description: System administrators/developers maintain and update the website/application.

#### Create and Manage Marketing Campaigns:

#### Actor: Marketing and Promotions Manager

#### Description: Marketing managers create and manage marketing campaigns and promotions.

#### Monitor and Enforce Policies:

#### Actor: Admin/Support

#### Description: Admin/support staff monitor and enforce platform policies.

#### View Reports and Analytics:

#### Actor: Analytics and Business Intelligence Team

#### Description: Analytics teams can view reports and analyze data.

#### Associations:

#### Customer interacts with "Search for Restaurants/Menu Items," "Place Order."

#### Restaurant Owner/Manager interacts with "Manage Restaurant Information," "Receive and Confirm Orders."

#### Delivery Driver interacts with "Deliver Orders."

#### Admin/Support interacts with "Manage User Accounts," "Monitor and Enforce Policies."

#### System Administrator/Developer interacts with "System Maintenance and Updates."

#### Marketing and Promotions Manager interacts with "Create and Manage Marketing Campaigns."

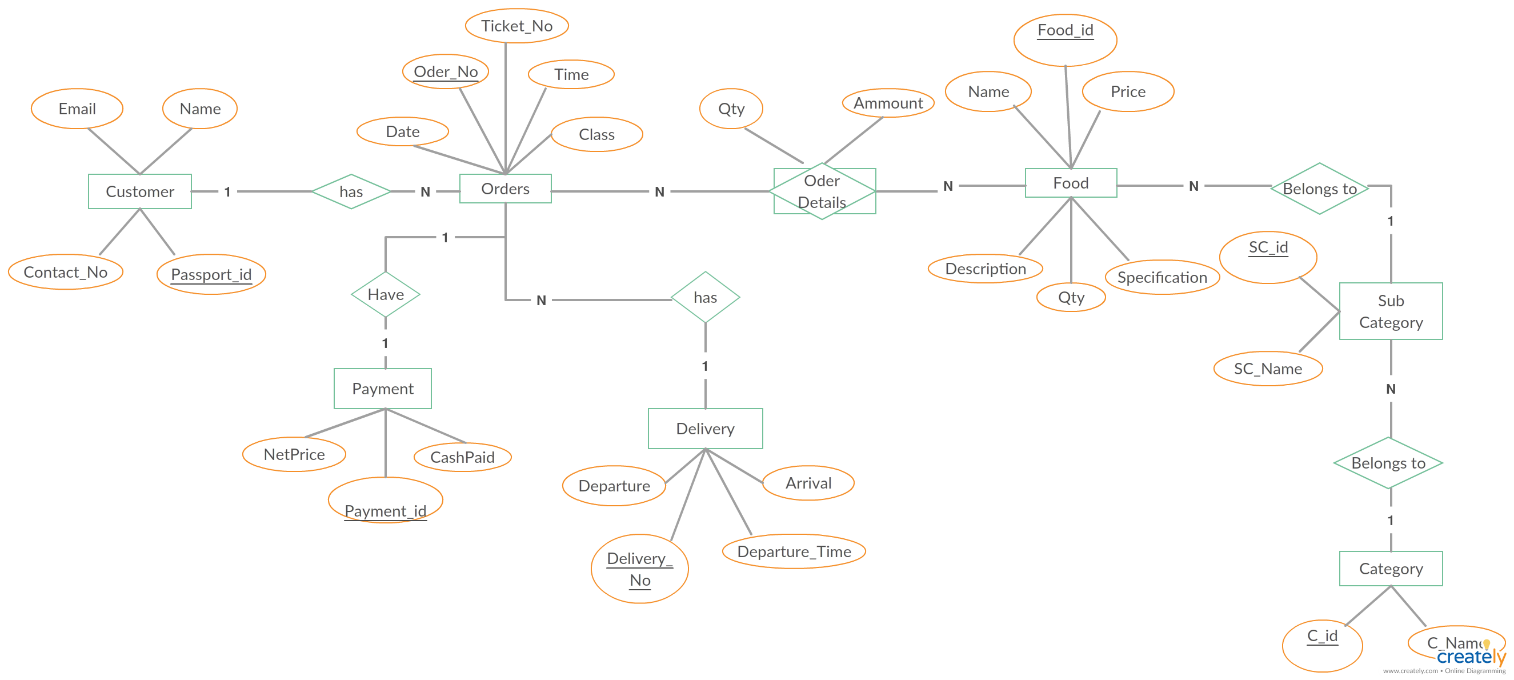
#### Analytics and Business Intelligence Team interacts with "View Reports and Analytics."

#### This diagram provides a high-level overview of the main interactions and functionalities within the food delivery website and application. It should be expanded and customized based on the specific features and requirements of your platform.

#### This is a broad level diagram of the project showing a basic overview. The users can be either staff or student.. This System will provide a search functionality to facilitate the search of resources. This search will be based on various categories viz. book name or the ISBN. Further the library staff personnel can add/update the resources and the resource users from the system.The users of the system can request issue/renew/return of books for which they would have to follow certain criteria.

#### 2.2 Product Function

Entity Relationship Diagram of food delivery website and application;



**Fig; Entity Relationship Diagram of food delivery website and application;**

The Online Library System provides online real time information about the books available in the Library and the user information. The main purpose of this project is to reduce the manual work. This software is capable of managing Book Issues, Returns, Calculating/Managing Fine, Generating various Reports for Record-Keeping according to end user requirements. The Librarian will act as the administrator to control members and manage books. The member’s status of issue/return is maintained in the library database. The member’s details can be fetched by the librarian from the database as and when required. The valid members are also allowed to view their account information.

#### 2.3 User Classes and Characteristics

#### Customer/User:

#### Characteristics:

#### Registered or guest users

#### Place food orders

#### Track order status

#### Save delivery addresses

#### View order history

#### Rate and review restaurants

#### Manage payment methods

#### Receive notifications

#### Restaurant Owner/Manager:

#### Characteristics:

#### Register and manage restaurant information

#### Add and update menu items

#### Set restaurant hours

#### Receive and manage incoming orders

#### Monitor customer reviews and ratings

#### Access sales and performance data

#### Manage restaurant profile

#### Delivery Driver:

#### Characteristics:

#### Receive order assignments

#### Navigate to customer locations

#### Confirm order pickup and delivery

#### Track earnings and payouts

#### Access order-specific instructions

#### Set availability status (online/offline)

#### View order history and statistics

#### Admin/Support:

#### Characteristics:

#### Manage user accounts

#### Verify restaurant registrations

#### Handle customer support inquiries

#### Monitor and enforce platform policies

#### Analyze data for business insights

#### Perform system maintenance and updates

#### Resolve disputes and issues

#### System Administrator/Developer:

#### Characteristics:

#### Maintain and update the website/application

#### Ensure system security and scalability

#### Implement new features and enhancements

#### Monitor system performance

#### Handle technical issues and bugs

#### Collaborate with other teams (e.g., database administrators, designers)

#### Marketing and Promotions Manager:

#### Characteristics:

#### Create and manage marketing campaigns

#### Offer discounts and promotions

#### Analyze customer behavior and preferences

#### Optimize user engagement and retention

#### Collaborate with restaurants for special promotions

#### Analytics and Business Intelligence Team:

#### Characteristics:

#### Analyze user data and behavior

#### Generate reports and insights

#### Identify trends and opportunities

#### Provide data-driven recommendations

#### Support decision-making and strategy.

#### Each of these user classes plays a critical role in the operation and success of a food delivery website and application. Their characteristics and functions may vary depending on the specific features and requirements of the platform. Additionally, some users may have overlapping roles or access to multiple functionalities, especially in larger organizations. 2.4 Operating Environment

The product will be operating in windows environment. The Library Management 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.

#### 2.5 Assumptions and Dependencies

The assumptions and dependencies are:-

• Document any assumptions made during the requirements gathering process.

• Identify external dependencies (e.g., third-party APIs, payment gateways).

#### 2.6 Requirement

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

#### 2.7 Data Requirement

List the types of data that the application will store and manipulate (e.g., user profiles, restaurant information, order history).

### 3. External Interface Requirement

#### 3.1 GUI

The software provides good graphical interface for the user and the administrator can operate on the system, performing the required task such as create, update, viewing the details of the book.

* It allows user to view quick reports like Book Issued/Returned in between particular time.
* It provides stock verification and search facility based on different criteria.
* The user interface must be customizable by the administrator
* All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined
* The design should be simple and all the different interfaces should follow a standard template
* The user interface should be able to interact with the user management module and a part of the interface must be dedicated to the login/logout module

Login Interface:-

In case the user is not yet registered, he can enter the details and register to create his account. Once his account is created he can ‘Login’ which asks the user to type his username and password. If the user entered either his username or password incorrectly then an error message appears.

Search:-

The member or librarian can enter the type of book he is looking for and the title he is interested in,then he can search for the required book by entering the book name.

Categories View:-

Categories view shows the categories of books available and provides ability to the librarian to add/edit or delete category from the list.

Librarian’s Control Panel:-

This control panel will allow librarian to add/remove users; add, edit, or remove a resource. And manage lending options.

### 4. System Features

The users of the system should be provided the surety that their account is secure. This is possible by providing:-

* User authentication and validation of members using their unique member ID
* Proper monitoring by the administrator which includes updating account status, showing a popup if the member attempts to issue number of books that exceed the limit provided by the library policy, assigning fine to members who skip the date of return
* Proper accountability which includes not allowing a member to see other member’s account. Only administrator will see and manage all member accounts

### 5. Other Non-functional Requirements

#### 5.1 Performance Requirement

The proposed system that we are going to develop will be used as the Chief performance system within the different campuses of the university which interacts with the university staff and students. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the university.

* The performance of the system should be fast and accurate
* Library Management System shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period. Thus it should have inbuilt error testing to identify invalid username/password
* The system should be able to handle large amount of data. Thus it should accommodate high number of books and users without any fault

#### 5.2 Safety Requirement

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

#### 5.3 Security Requirement

#### Regularly update security measures to protect user data and payment information.

#### 5.4 Requirement attributes

* There may be multiple admins creating the project, all of them will have the right to create changes to the system. But the members or other users cannot do changes
* The project should be open source
* The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
* The user be able to easily download and install the system

#### 5.5 Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data.This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

#### 5.6 User Requirement

The users of the system are members and Librarian of the university who act as administrator to maintain the system. The members are assumed to have basic knowledge of the computers and internet browsing. The administrators of the system should have more knowledge of the internals of the system and is able to rectify the small problems that may arise due to disk crashes, power failures and other catastrophes to maintain the system. The proper user interface, user manual, online help and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

The admin provides certain facilities to the users in the form of:-  Backup and Recovery

* Forgot Password
* Data migration i.e. whenever user registers for the first time then the data is stored in the server
* Data replication i.e. if the data is lost in one branch, it is still stored with the server
* Auto Recovery i.e. frequently auto saving the information
* Maintaining files i.e. File Organization
* The server must be maintained regularly and it has to be updated from time to time

### 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;

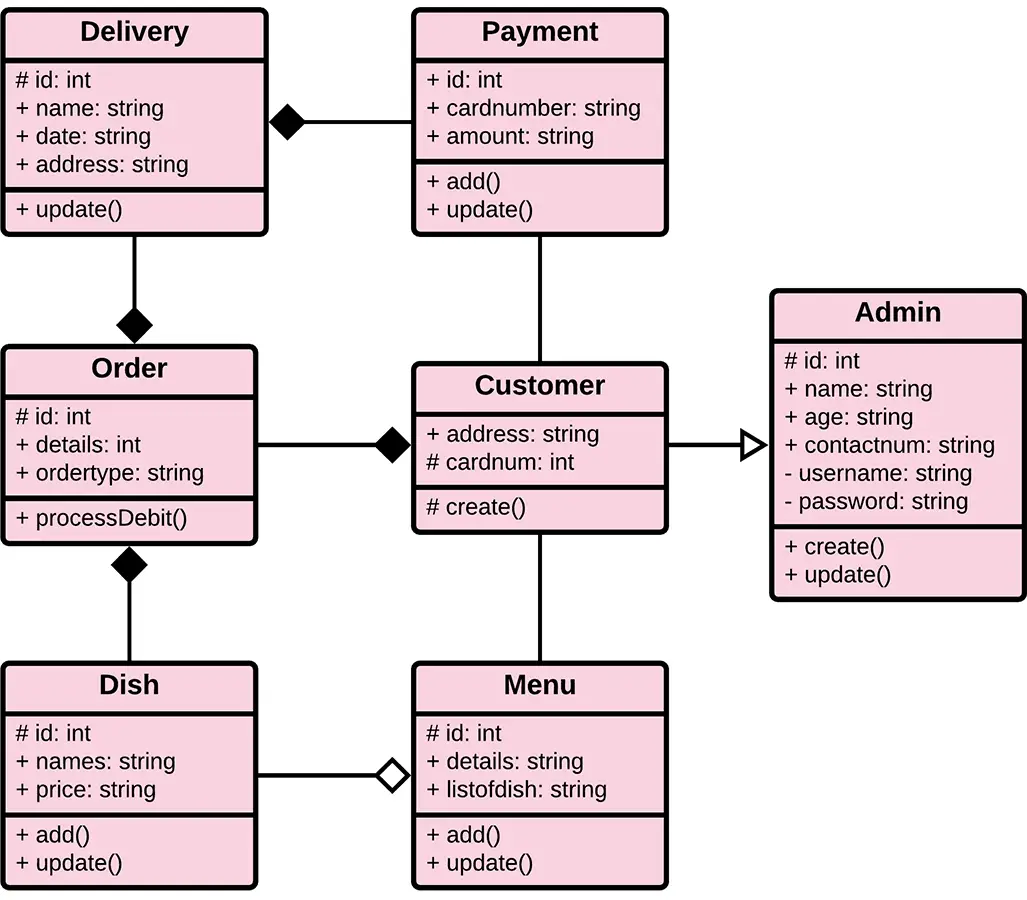
#### 6.3 Glossary

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

#### 6.4 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 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 ‘Librarian’, ‘Member’ and ‘Books’ are the most important classes which are related to other classes.



**FIG: Class Diagram of food delivery website and application**.

1. **Food Delivery :**

* Represents the overall system of the food delivery website and application.
* Contains attributes and methods related to managing users, restaurants, and drivers.

2. **User** :

* Represents users of the system, including customers.
* Attributes include UserID, Username, Password, Email, Phone, and Address.
* Methods include Register() for user registration, Login() for user authentication, PlaceOrder() for placing food orders, and RateRestaurant() for leaving reviews and ratings.

3. **Restaurant :**

* Represents restaurants that are registered on the platform.
* Attributes include RestaurantID, Name, Address, and a list of MenuItems.
* Methods include AddMenuItem() for adding menu items, EditMenuItem() for editing menu items, RemoveMenuItem() for removing menu items, and ProcessOrder() for managing incoming orders.

4. **MenuItem** :

* Represents individual food items available on a restaurant's menu.
* Attributes include ItemID, Name, Description, and Price.
* Has a method GetDetails() to retrieve information about the menu item.

5. **Order** :

* Represents a customer's order.
* Attributes include OrderID, references to the User and Restaurant, a list of menu items (Items), TotalPrice, and Status.
* Methods include AddItem() and RemoveItem() for managing items in the order, CalculateTotal() for calculating the total cost, and UpdateStatus() for tracking the order status.

6. **DeliveryDriver** :

* Represents the delivery drivers.
* Attributes include DriverID, Name, Vehicle, and Location.
* Methods include AcceptOrder() for accepting delivery orders and DeliverOrder() for marking orders as delivered.

7. **Payment** :

* Represents payment information associated with an order.
* Attributes include PaymentID, reference to the Order, Amount, and PaymentMethod.
* Has a method ProcessPayment() for handling payment processing.

8. **Review and Rating** :

* Represents customer reviews and ratings for restaurants.
* Attributes include ReviewID, references to the User and Restaurant being reviewed, Rating, and Comment.
* Includes a method SubmitReview() for submitting restaurant reviews.

9. **Admin** :

* Represents administrators of the system.
* Attributes include AdminID, Username, and Password.
* Methods include BanUser() for banning users, AddRestaurant() for adding new restaurants, and RemoveRestaurant() for removing restaurants.

10.**Notification** :

* Represents notifications that can be sent to users.
* Attributes include NotificationID, reference to the User, Message, and Timestamp.
* Contains a method SendNotification() to send notifications to users.

11. **Location** :

* Represents geographical locations.
* Attributes include LocationID, Latitude, Longitude, and Address.
* Includes methods GetCoordinates() and CalculateDistance() for handling location-related functionality.

These classes represent the core entities and functionalities of a food delivery system. They are interconnected through relationships like composition, aggregation, and associations, which are not explicitly represented in the textual diagram but can be defined based on the specific needs of your system. Class diagrams like this help visualize the structure and relationships of the components within your software system, making it easier to plan and develop the application.