SOFTWARE REQUIREMENTS SPECIFICATION

For

Application for Grocery Delivery System

Prepared by:-

BALAMURUGAN . T

Academic Year: 2020-2024

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for the development of a Grocery Delivery Application, hereafter referred to as "the application." This application will enable customers to order groceries online and have them delivered to their specified location.

1.2 Scope of Development Project

The scope of this document encompasses the functional and non-functional requirements of the application, including user interactions, system behavior, security, performance, and other relevant aspects.The project can be easily implemented under various situations. We can add new features as and when we require, making reusability possible as there is flexibility in all the modules.

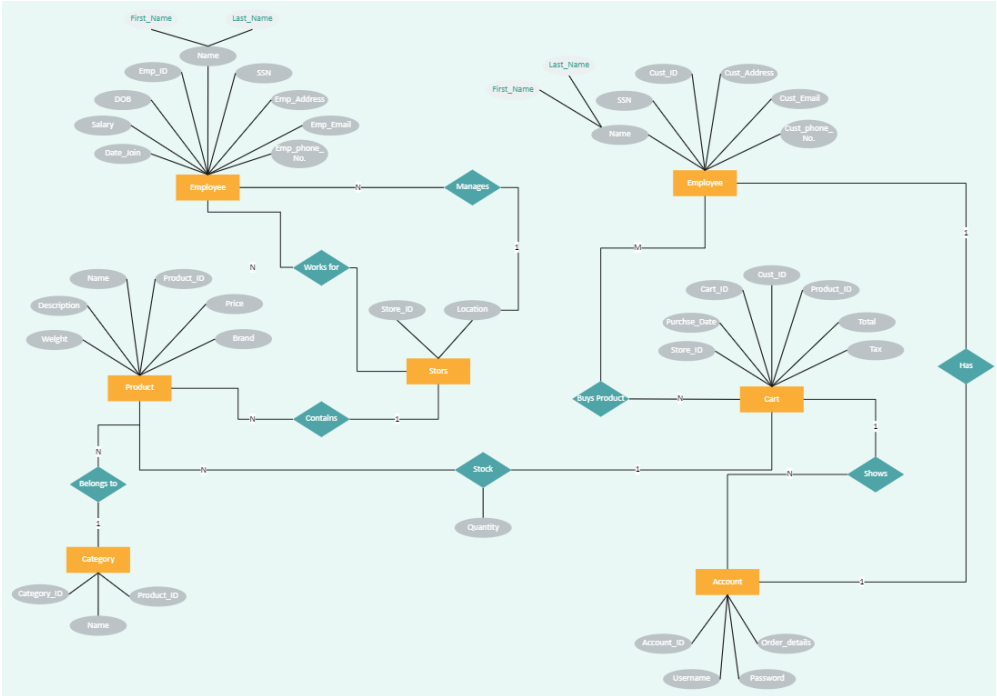
The language used for developing the project is Java as it is quite advantageous than other languages in terms of performance, tools available, cross platform compatibility, libraries, cost (freely available), and development process.

2. Functional Requirements

2.1 User Registration and Authentication

1. Users should be able to register for an account using their email or phone number.
2. The application must provide a secure authentication mechanism (e.g., password, two-factor authentication) for user login.

2.2 Product Function : Entity Relationship Diagram of grocery delivery application



1. Users can create and edit their profiles, including adding and modifying personal information and delivery addresses.
2. Users can view their order history.

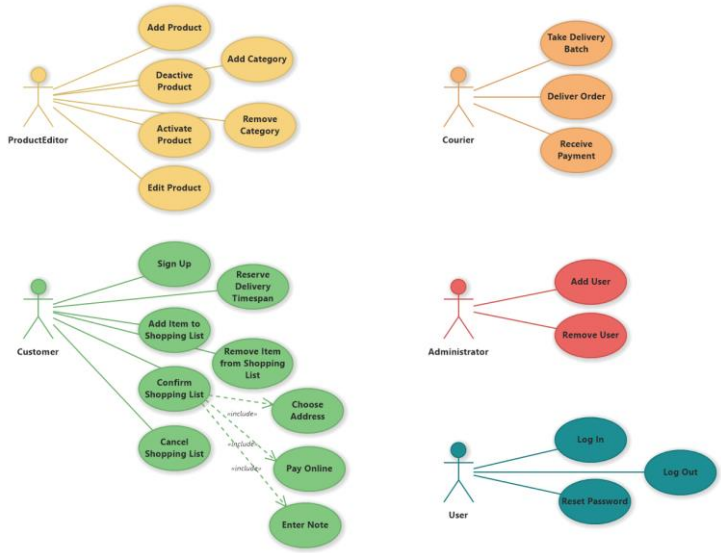
2.4 Product Catalog

1. The application should display a catalog of available grocery items, including product descriptions, prices, and images.
2. Users can search and filter products based on categories, brands, and keywords.

2.5 Cart Management

1. Users can add and remove items from their shopping cart.
2. Users can view the total cost of items in their cart.
3. Users can specify quantities for each item in their cart.

Product Perspective: Use Case Diagram of Library Management System



1. Users can place orders from their shopping cart.
2. Users can select a delivery date and time.
3. Users can specify special instructions for delivery.
4. Users receive an order confirmation email or notification.

2.6 Order Tracking

1. Users can track the status of their orders in real-time.
2. Delivery drivers can update the order status as they progress through the delivery process.

2.7 Payments

1. Users can securely enter payment information.
2. The application must support multiple payment methods (credit/debit

cards, digital wallets, etc.).

2.8 Notifications

1. Users receive notifications (email, SMS, in-app) for order updates, promotions, and special offers.
2. Users can opt-in or opt-out of specific notification types.

2.9 Admin Panel

1. Administrators can manage product listings, including adding, updating, and removing items.
2. Administrators can manage user accounts, including account suspension and removal.
3. Administrators can view and manage orders.

3.Non-Functional Requirements

3.1 Security

1. The application must use encryption for sensitive data, including user information and payment details.
2. Access to the admin panel should be restricted and protected with strong authentication.

3.2 Performance

1. The application should load product listings and user data quickly, even during peak usage.
2. The system should be scalable to handle a growing user base and increased order volume.

3.3 Availability

1. The application should have high availability, with minimal downtime for maintenance.
2. Scheduled maintenance should be communicated to users in advance.

3.4 Usability

1. The application must have an intuitive user interface that is easy to navigate.
2. It should be responsive and compatible with various devices and screen sizes.

3.5 Compliance

1. The application should comply with relevant data protection and privacy regulations (e.g., GDPR, CCPA).
2. Payment processing must comply with industry-standard security standards (e.g., PCI DSS).

4. Constraints

1. The application must be developed using [programming languages and frameworks].
2. The application should be compatible with [specific operating systems or

platforms].
3. The development should adhere to a predefined budget and timeline.

5. Assumptions and Dependencies

1. The application assumes that users have internet access and compatible devices.
2. The application depends on third-party payment gateways for processing payments.

6. Appendix

Include any diagrams, mockups, or additional documentation as needed to support the requirements outlined in this SRS.

This SRS serves as the foundation for the development of the Grocery Delivery Application. It should be reviewed and updated as necessary to ensure that the final product meets the specified requirements.

6.1Glossary : Glossary The following are the list of conventions and acronyms used in this document and the project as well:The following are the list of conventions and acronyms used in this document and the project as well:
Sure, here's a glossary of terms related to a grocery delivery application:

1. Grocery Delivery Application:The mobile or web-based platform that allows users to order groceries online for delivery to their doorstep.

2. User: An individual who uses the grocery delivery application to place orders and receive groceries.

3. Retailer/Store: The physical grocery store or supermarket that partners with the application to supply groceries to customers.

4. Product Catalog: A database or list of all the grocery items available for purchase through the application, including details like product name, price, and description.

5. Shopping Cart:A virtual basket or container where users add selected grocery items before proceeding to checkout and payment.

6. Checkout: The process of finalizing an order, selecting a delivery time, and making payment for the groceries in the shopping cart.

7. Payment Gateway: A secure online service that handles payment transactions, allowing users to pay for their groceries using credit/debit cards, digital wallets, or other payment methods.

8. Delivery Address:The address provided by the user where they want their groceries to be delivered.

9. Delivery Slot/Time: The designated time period during which the user expects

their groceries to be delivered to their specified address.

10. Order Confirmation: A notification or email sent to the user after successfully placing an order, confirming the details of the order.

11. Inventory Management: The system used by the retailer to track and manage the availability of products in their store and update the online catalog accordingly.

12. Push Notifications: Messages sent to users' mobile devices to provide updates on their order status, delivery time, or promotional offers.

13. Rating and Review: Users' feedback and ratings for products, delivery services, and the overall experience, which can help other customers make informed choices.

14. Promo Code/Coupon: A special code that users can apply to receive discounts or promotional offers on their grocery orders.

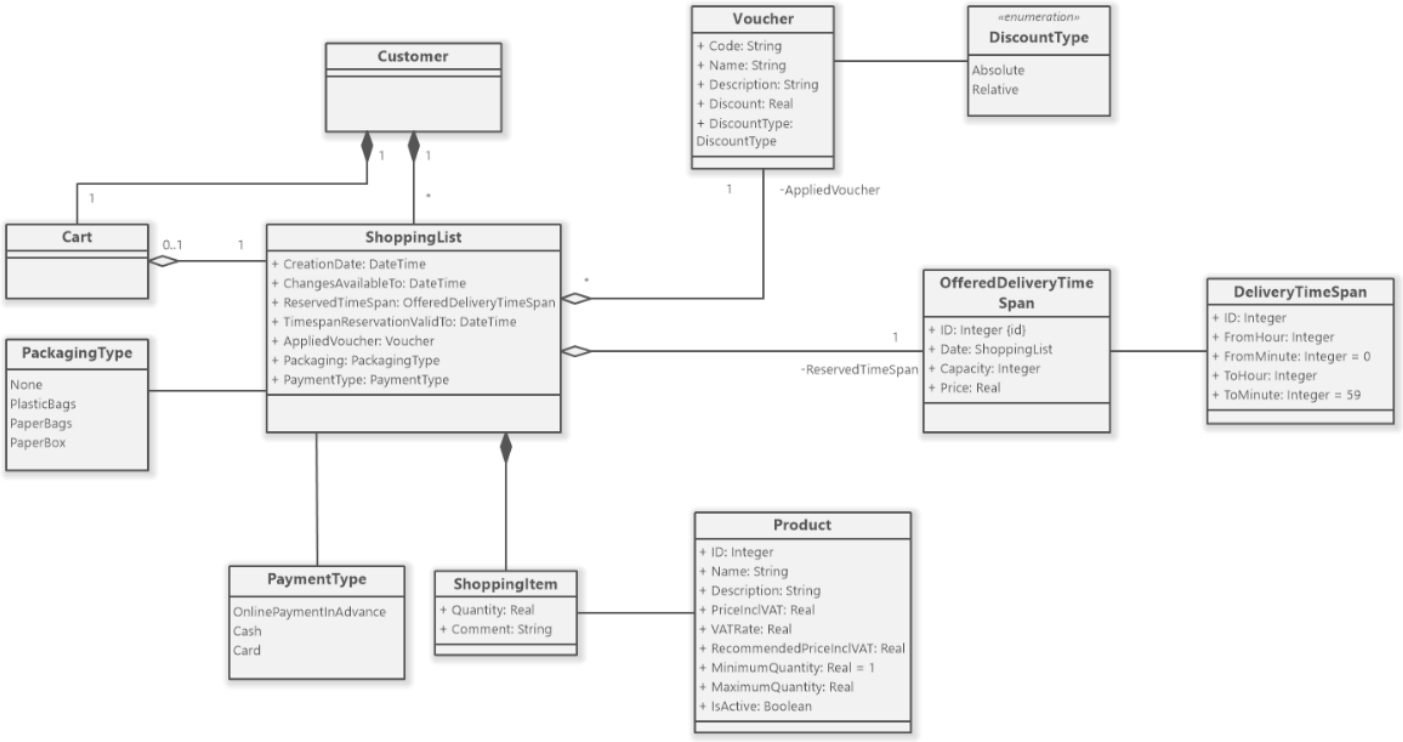
15. Driver/Delivery Person: The individual responsible for picking up the groceries from the store and delivering them to the customer's specified address.

16. Real-time Tracking: The feature that allows users to track the location of their delivery in real-time as it makes its way to their address.

17. Customer Support: The customer service team or chat support available to assist users with their inquiries, issues, or complaints related to the application.

6.2 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 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 classesUsers can register, log in, place orders, and track their orders.

- Delivery persons accept orders and update delivery status.
- Stores list available items, manage inventory, and process orders.
- Orders contain items, total amounts, and order status.
- Items have details like name, price, and quantity in stock.
- Payments are associated with orders and have payment details.
- Vehicles are assigned to delivery persons for order delivery.
- Notifications are sent to users.
- Users can submit reviews for stores.
- Admins manage stores, users, and view reports


Customer
Voucher
+ Code: String
+ Name: String
+ Description: String
+ Discount: Real
+ DiscountType: DiscountType
«enumeration»
DiscountType
Absolute
Relative
1
1
1
*
1
-AppliedVoucher
Cart
0..1
1
ShoppingList
+ CreationDate: DateTime
+ ChangesAvailableTo: DateTime
+ ReservedTimeSpan: OfferedDeliveryTimeSpan
+ TimespanReservationValidTo: DateTime
+ AppliedVoucher: Voucher
+ Packaging: PackagingType
+ PaymentType: PaymentType
*
1
-ReservedTimeSpan
OfferedDeliveryTime Span
+ ID: Integer {id}
+ Date: ShoppingList
+ Capacity: Integer
+ Price: Real
DeliveryTimeSpan
+ ID: Integer
+ FromHour: Integer
+ FromMinute: Integer = 0
+ ToHour: Integer
+ ToMinute: Integer = 59
PackagingType
None
PlasticBags
PaperBags
PaperBox
PaymentType
OnlinePaymentInAdvance
Cash
Card
ShoppingItem
+ Quantity: Real
+ Comment: String
Product
+ ID: Integer
+ Name: String
+ Description: String
+ PriceInclVAT: Real
+ VATRate: Real
+ RecommendedPriceInclVAT: Real
+ MinimumQuantity: Real = 1
+ MaximumQuantity: Real
+ IsActive: Boolean