

# **SOFTWARE REQUIREMENTS SPECIFICATION**

**For**

## **Application for Grocery Delivery**

**PREPARED BY:**

GEETHA M  
JAYANTHI N  
KARTHIGA A  
RAJALAKSHMI J

# **1. Introduction**

## **1.1 Purpose**

The purpose of this document is to present a detailed description of the Application for Grocery Delivery. It will explain the purpose and features of the system, the interfaces of the system and what the system will do. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. This project describes the hardware and software interface requirements using ER diagrams and UML diagrams.

The system is developed with a user-friendly and attractive GUI. It delivers a wide range of groceries available online. Users have to first login into the system to view the groceries and add them into their cart. They can then order it by making a secure online payment via CREDIT CARD. The system functionality of products and orders is stored on server side in a web service.

## **1.2 Document Conventions**

- Convention for Sub title
  - Font face: Times New
  - RomanFont style: Bold
  - Font Size: 16
- Convention for body
  - Font face: Times New
  - RomanFont Size: 14

## **1.3 Scope of Development Project**

The purpose of the Online Grocery Management System is to easy Grocery management and to Create a convenient and easy-to-use application for consumers/customers, trying to buy grocery Goods. The system is based on a relational database with its grocery management and delivery Functions. The system is also designed to allow an admin to manage and check the stocks of goods in the Grocery.

The software will facilitate communication between admins, clerks, buyers (retailers and wholesalers) and the IT professional via app or E-mail. Preformatted reply forms are used in Every transactions or Inquiry of customers through the system.

## **1.4 Intended Audience and Reading Suggestions**

This project is a prototype for the Online Grocery Management System and it is restricted within the college premise. This has been implemented under the guidance of college professors and professional IT individuals. This project is useful for the Grocery Staff, Admin and as well to be consumers and Customers.

## **1.5 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

IEEE -> Institute of Electrical and Electronics Engineers

## **1.6 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

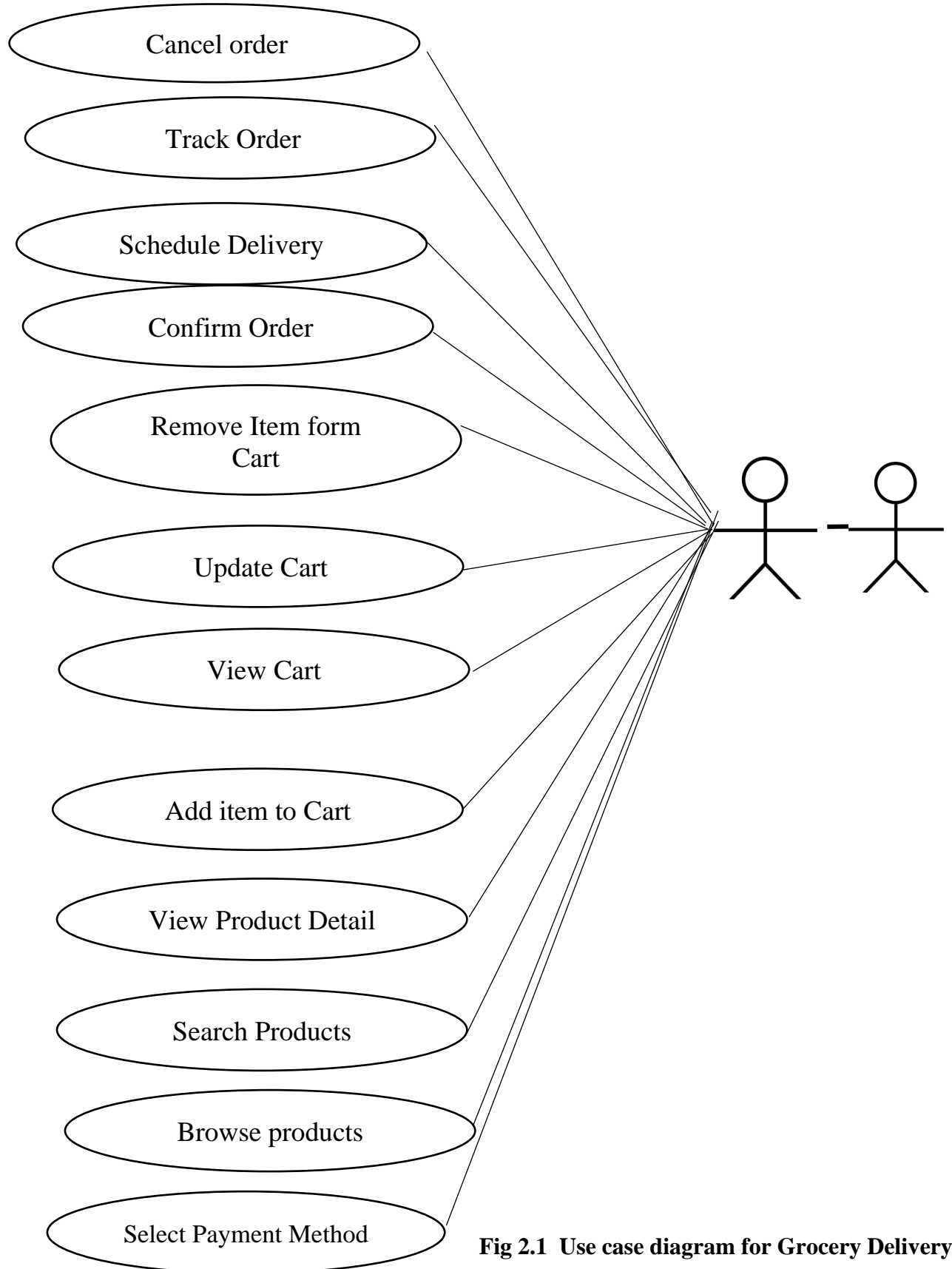
- <https://www.geeksforgeeks.org/how-to-write-a-good-srs>
- <https://techcrunch.com/2020/03/16/grocery-delivery-apps-see-record-downloads-amid-coronavirus-outbreak/>

## **2. Overall Descriptions**

### **2.1 Product Perspective**

A Use Case Diagram is a visual representation used in the field of software engineering and system analysis to depict the various interactions and functionalities of a system from the perspective of its users, known as actors. These diagrams are part of the Unified Modeling Language (UML) and help in understanding the system's requirements and how users or external entities interact with it.

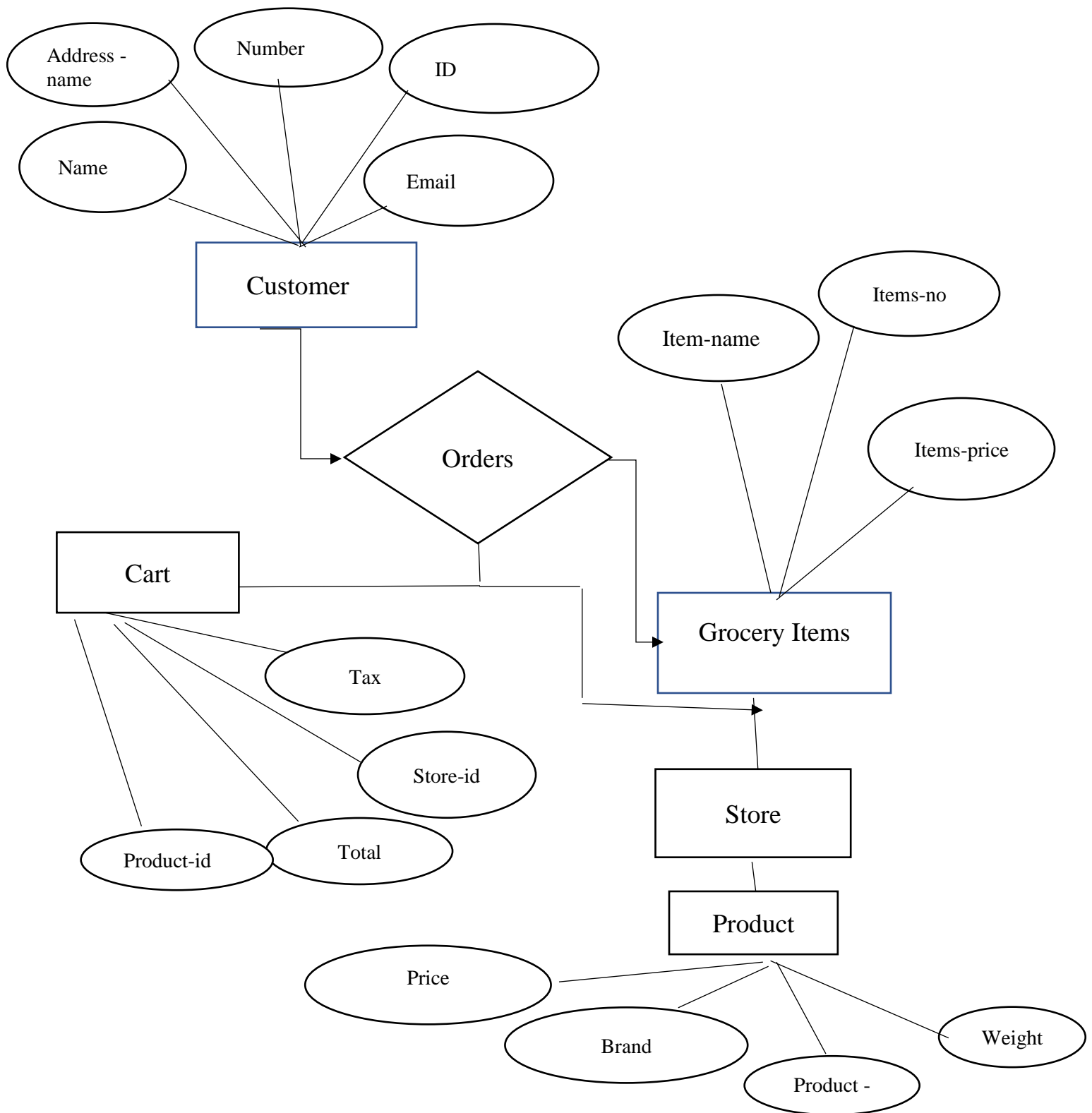
## Use Case Diagram Application for Grocery Delivery



**Fig 2.1 Use case diagram for Grocery Delivery**

## 2.2 Product Function

Entity Relationship Diagram of Application for Grocery Delivery



**Fig 2.2 Entity Diagram**

## 2.3 User Classes and Characteristics

For grocery delivery applications, there are different entities that have different characteristics, roles, and responsibilities to consider for a good environment setup.

### Customers:

Characteristics:

- Individuals or households.
- Place grocery orders.
- Preferences for specific products or vendors.

Roles:

- Browse, select, and order groceries.
- Monitor order status and receive updates.
- Provide feedback and ratings.

### Delivery Drivers:

Characteristics:

- Contracted drivers or couriers.
- Responsible for order deliveries.

Roles:

- Accept and fulfill delivery orders.
- Utilize GPS for route optimization.
- Confirm deliveries with digital records.

### Vendors:

Characteristics:

- Grocery stores or suppliers.
- Offer product listings and inventory.
- Manage order processing and fulfillment.

Roles:

- List and update products and pricing.
- Process incoming customer orders.
- Maintain inventory status.

### Administrators:

Characteristics:

- System administrators.
- Manage the application, users, and vendors.

Roles:

- Manage user accounts and roles.

## 2.4 Operating Environment

The Grocery Delivery Application will operate in a Windows environment, ensuring compatibility with various web browsers, including Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox. It will be optimized for Internet Explorer 6.0 and support most features on Mozilla Firefox and Opera 7.0 or higher versions. The sole requirement for using this online product is a stable internet connection. In terms of hardware configuration, the system will function with a minimum of a 40 GB hard disk, a 15" color monitor, and a 122-key keyboard. Basic input devices such as a keyboard and mouse, as well as output devices like monitors and printers, will be supported by the application. This inclusive approach to operating and hardware environments ensures accessibility and usability for a broad range of users.

## 2.5 Assumptions and Dependencies

### The Assumptions are:

- **Internet Connectivity:** The Grocery Delivery Application assumes that users have a reliable internet connection to access the platform and place orders.
- **User Device Compatibility:** Users are expected to have compatible devices, such as smartphones or computers, to access the application.
- **Delivery Infrastructure:** It assumes the availability of a delivery infrastructure, including drivers and vehicles, to fulfill customer orders.
- **Payment Systems:** Users are assumed to have access to payment methods supported by the application, such as credit/debit cards or digital wallets.

### The Dependencies are:

- **External APIs:** The application may depend on external APIs for services such as mapping and geolocation for delivery tracking.
- **Payment Gateways:** Integration with third-party payment gateways is a dependency for processing transactions securely.
- **Database Server:** The application relies on the availability and proper functioning of the Microsoft SQL Server database server for data storage and retrieval.
- **Cloud Services:** If the application uses cloud-based resources for scalability and performance, it depends on the reliability of these cloud services.

## **2.6 Requirement**

### **Software Configuration:-**

This software package is developed using Java as the front end which is supported by Sun Microsystem. Microsoft SQL Server as the back end to store the database.

Operating System: Windows NT, Windows 98, Windows XP and above.,

Language & IDE: 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**

The inputs consist of the query to the database and the output consists of the solutions for the query. The output includes user details for registration, product information such as catalog listings and inventory, order specifics like order IDs and delivery details, reviews and ratings data to facilitate user feedback, and payment records for secure transactions. Location data, both for users and drivers, plays a crucial role in tracking and ensuring timely deliveries. Administrative data logs system activities and analytics provide insights into sales and user behavior.

## **3. External Interface Requirement**

### **3.1 GUI**

The graphical user interface (GUI) design for the Grocery Delivery Application is of paramount importance, as it directly impacts user engagement and satisfaction. Our design approach prioritizes user-centered principles to create an intuitive and visually appealing interface. We consider the diverse user base, including customers, delivery drivers, vendors, and administrators, ensuring that the GUI caters to their specific needs and preferences. The design is responsive, adapting seamlessly to various devices and screen sizes, guaranteeing a consistent and enjoyable user experience on desktops, tablets, and smartphones.

Our GUI design focuses on efficient navigation and hierarchy, making it easy for users to browse products, manage their shopping carts, and track orders. We employ visually engaging elements, including high-quality images, icons, and a cohesive color scheme, to enhance the overall aesthetic and brand identity of the application.



## 4. System Features

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

- **User Registration and Shopping:** User-friendly registration and login, enabling customers to browse and purchase products.
- **Efficient Cart Management:** Streamlined shopping cart for easy item addition, removal, and real-time cost updates.
- **Secure Checkout and Payment:** Secure, multi-payment options, including digital wallets and card payments.
- **Delivery Tracking:** Real-time order tracking with driver information and delivery status updates.
- **Vendor and Inventory Control:** Vendor onboarding, inventory management, and low-stock alerts.
- **Analytics and Admin Control:** Data analytics for administrators, user reviews, and continuous system improvement

## 5. Non-functional Requirements

### 5.1 Performance Requirement

- **Response Time:** Ensure that the application responds quickly to user actions, with pages loading within a specified time frame.
- **Scalability:** The application should handle an increasing number of users and orders without significant performance degradation.

### 5.2 Reliability

- **Availability:** Aim for high availability, minimizing downtime for maintenance or unexpected issues.
- **Error Handling:** Provide informative error messages and gracefully handle errors to prevent service interruptions.

### 5.3 Security

- **Data Encryption:** Use encryption to protect sensitive data such as user information and payment details.
- **Authentication and Authorization:** Implement strong authentication and authorization to ensure that only authorized users can access specific features.

➤ **Secure APIs:** Ensure that APIs used for payment processing and other sensitive operations are secure and protected from attacks.

## 5.4 Data Management

- **Data Backup and Recovery:** Regularly back up user data and provide a robust data recovery mechanism in case of data loss
- **Data Privacy:** Comply with data privacy regulations and protect user data from unauthorized access.

## 5.5 Scalability and Load Handling

- **Load Testing:** Perform load testing to ensure the application can handle a high volume of concurrent users and orders.
- **Horizontal Scaling:** Be able to scale horizontally by adding more servers or resources during peak usage.

## 5.6 Usability

- **User Interface Design:** Create an intuitive and user-friendly interface that is easy to navigate.
- **Accessibility:** Ensure the application is accessible to users with disabilities, complying with accessibility standards.
- **Cross-Browser Compatibility:** Ensure the application works seamlessly on major web browsers.
- **Mobile Responsiveness:** Optimize the user experience for mobile devices through responsive design.
- **Logging:** Implement robust logging to capture system events, errors, and user activities for troubleshooting and auditing.
- **Monitoring:** Set up monitoring tools to track system health, performance, and security.
- **Regulatory Compliance:** Adhere to relevant legal and regulatory requirements, including food safety standards and e-commerce regulations.

## 6. Other Requirements

### 6.1 Data and Category Requirement

Data and category requirements are integral to the Grocery Delivery Application, governing how information is structured and utilized. This includes defining data types (user profiles, products, orders), schemas, and category hierarchies for product organization. User profiles' collection, roles, and data privacy measures are also specified. The application ensures real-time data synchronization, implements data security, and enforces compliance with data protection regulations. Additionally, it addresses data backup, reporting, and analytics needs. Finally, it supports localization and internationalization for a global user base, enriching the user experience.

### 6.2 Appendix

A: Admin, Abbreviation, Acronym, Assumptions;  
B: Business rules;  
C: Class, Client, Conventions;  
D: Data requirement, Dependencies;  
G: GUI, Grocery;  
K: Key;  
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 is 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 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

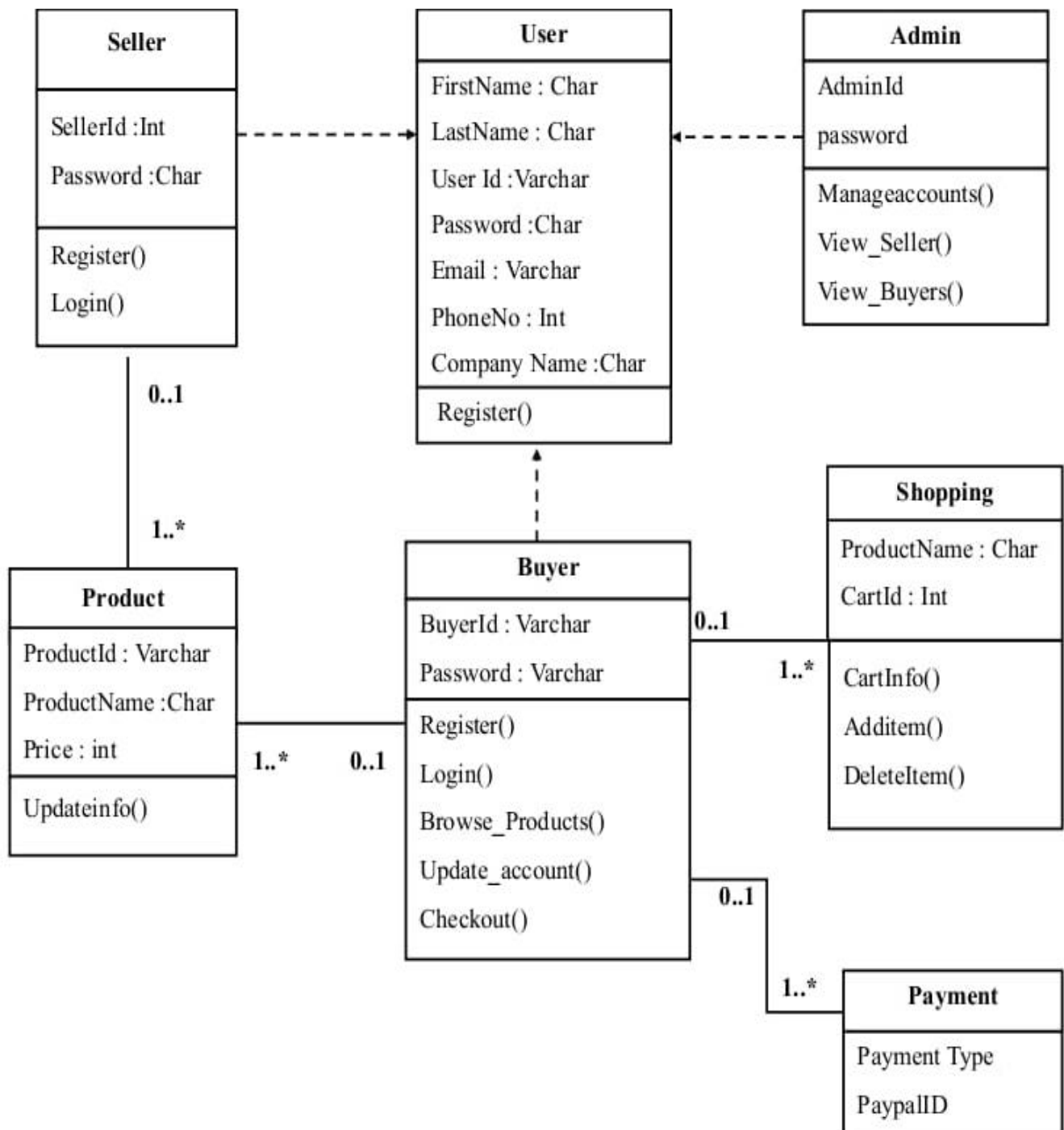
- **User Interface Layer:** The section of the assignment referring to what the user interacts with directly
- **Application Logic Layer:** The section of the assignment refers to the Web Server. This is where all computations are completed
- **Data Storage Layer:** The section of the assignment referring 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

This class diagram serves as a blueprint for the application's structure, illustrating how classes interact to deliver a seamless grocery shopping and delivery experience while accommodating user management, feedback, and administrative control.

The class diagram for the Grocery Delivery Application depicts the essential classes and their relationships within the system. The "User" class represents customers and administrators, allowing for user registration, login, and profile management. The "Product" class manages product details, enabling users to add items to their shopping carts. The "Shopping cart" class handles cart management, including item addition, removal, and cost calculation. The "Order" class tracks placed orders with details like order status and timestamps.

Vendors are managed by the "Vendor" class, which oversees product listings and inventory. The "Driver" class assigns and tracks delivery orders, ensuring efficient order fulfillment. "Admin" class controls system-wide activities, such as user management and data analysis. Reviews and notifications are managed by the "Review" and "Notification" classes, respectively, facilitating user feedback and communication. Finally, the "Payment" class handles transaction-related details, ensuring secure and efficient payment process



**Fig 6.4 class diagram**