

# *Software Requirements Specification*

*Author: KAVIN S*

*Version: 0.2*

*Date: 22,07, 2023*

## **TABLE OF CONTENT**

1	ABSTRACT
2	EXISTING SYSTEM
3	PROPOSED SYSTEM
4	MODULE DESCRIPTION
5	FUNCTIONAL REQUIREMENTS
6	USE CASE DIAGRAM
7	CLASS DIAGRAM
8	FLOW DIAGRAM
9	ER DIAGRAM

### **ABSTRACT:**

Online Food is an application that is used for buying the food that are needed for the customer.

### **EXISTING SYSTEM:**

The existing system does not have a fully functional Food application. It appears to be a work-in-progress code implementation of the application. The existing system consists of various Java classes that handle user registration, login, menu management, order placement, payment options, and order receipt generation. However, the implementation may not be complete, and some functionalities may still need to be developed.

### **PROPOSED SYSTEM:**

**Complete User Management:** The proposed system will have a robust user management system with secure user registration, login, and password handling. Users can create accounts, log in securely, and log out when they are done using the application.

**Improved Menu Management:** The menu management module will be expanded to allow admins to add, edit, or remove food items dynamically. The admin will have full control over managing the available food items on the menu.

**Order Placement and Management:** Customers can browse the menu, add multiple items to their order, and specify the quantity for each item. The system will calculate the total cost of the order and prompt the user for their delivery address. Order receipts will be generated and presented to the customer after successful order placement and payment.

**Multiple Payment Options:** Customers will have the option to choose between "Credit Card" and "Cash" as their payment methods. Credit card details, if provided, will be handled securely during payment processing, adhering to industry standards. The proposed system is a web-based chat application which allows applicants to communicate to the instructor in a direct and interactive way. The user can also post like shorts and images on the course as if of any regular updates on the course. In this way students don't get bored on the learning carrier and can interact live with the instructor.

## **MODULE DESCRIPTION:**

The application comprises the following major modules: Course module, Instructor module, Permission module, chat module and Student module

### **User Management Module**

- Allow users to create new accounts with unique usernames and strong passwords.
- Enable users to log in using their registered username and password.
- Validate user input for username and password during registration and login.
- Provide secure storage and handling of user passwords

### **Menu Management Module**

This module is responsible for managing the food items available in the menu.

Functionality:

- Maintain a list of food items with their names and prices.
- Display the menu to customers, showing the available food items in a user-friendly format.
- Allow admins to add, edit, or remove food items from the menu

### **Menu Management Module**

This module is responsible for managing the food items available in the menu.

Functionality:

- Maintain a list of food items with their names and prices.
- Display the menu to customers, showing the available food items in a user-friendly format.
- Allow admins to add, edit, or remove food items from the menu

### **Payment Management Module**

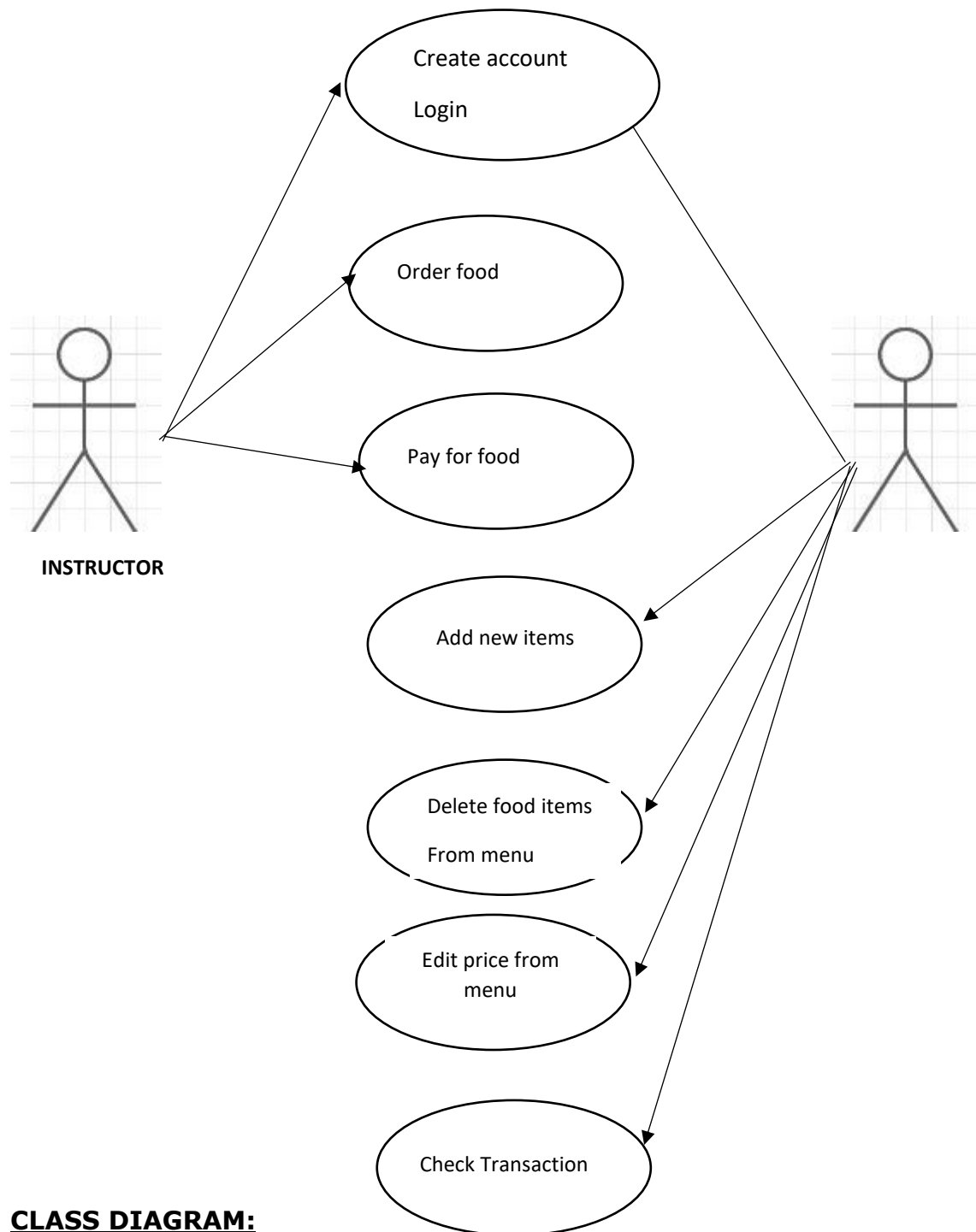
This module is responsible for handling payment options and processing payments.

Functionality:

- Provide customers with the option to choose between "Credit Card" and "Cash" as their payment methods.
- Validate credit card details if "Credit Card" is selected.
- Process payments securely and handle successful payment transactions.

- Ensure that credit card details are not stored in the system after payment processing

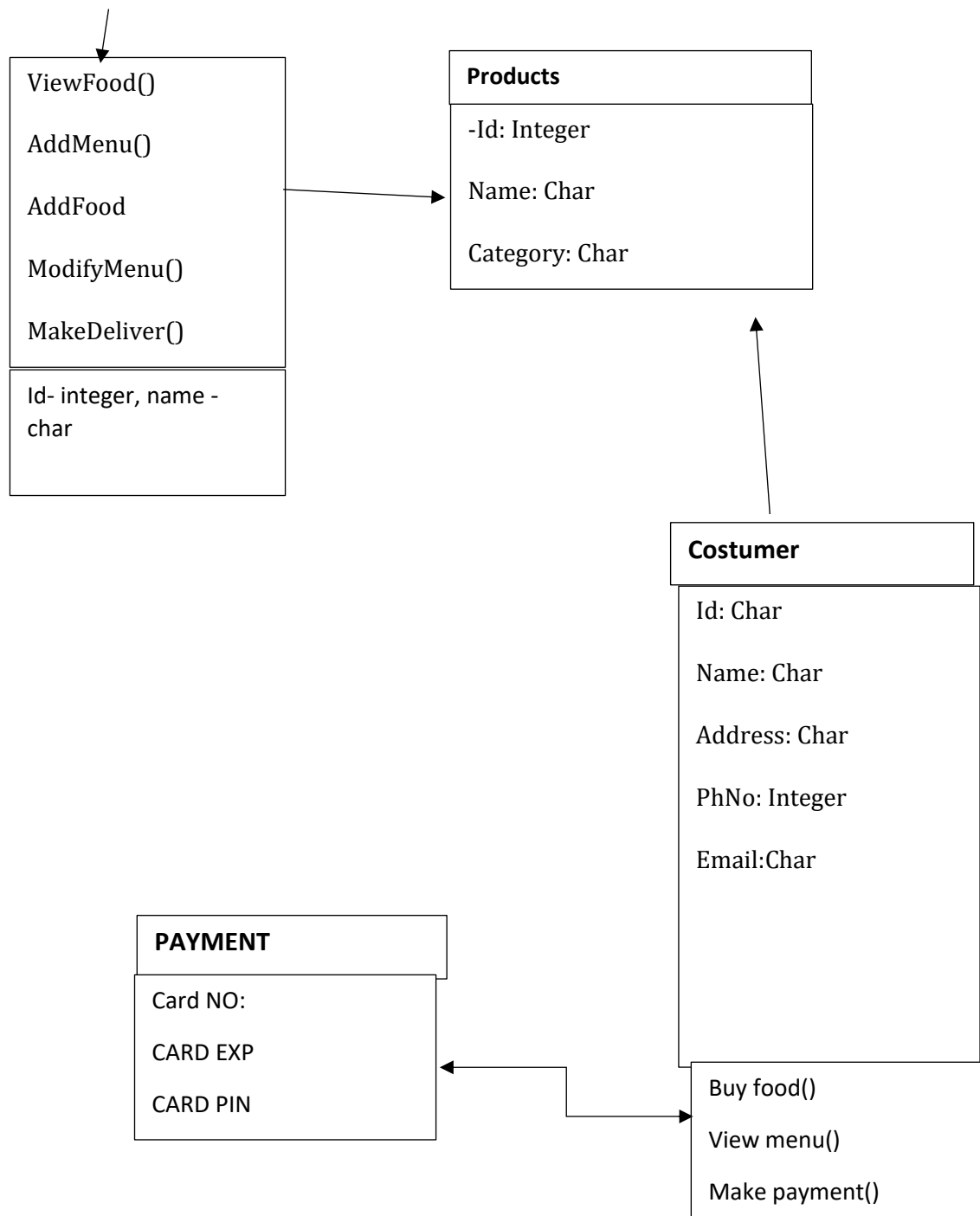
### USE CASE DIAGRAM:

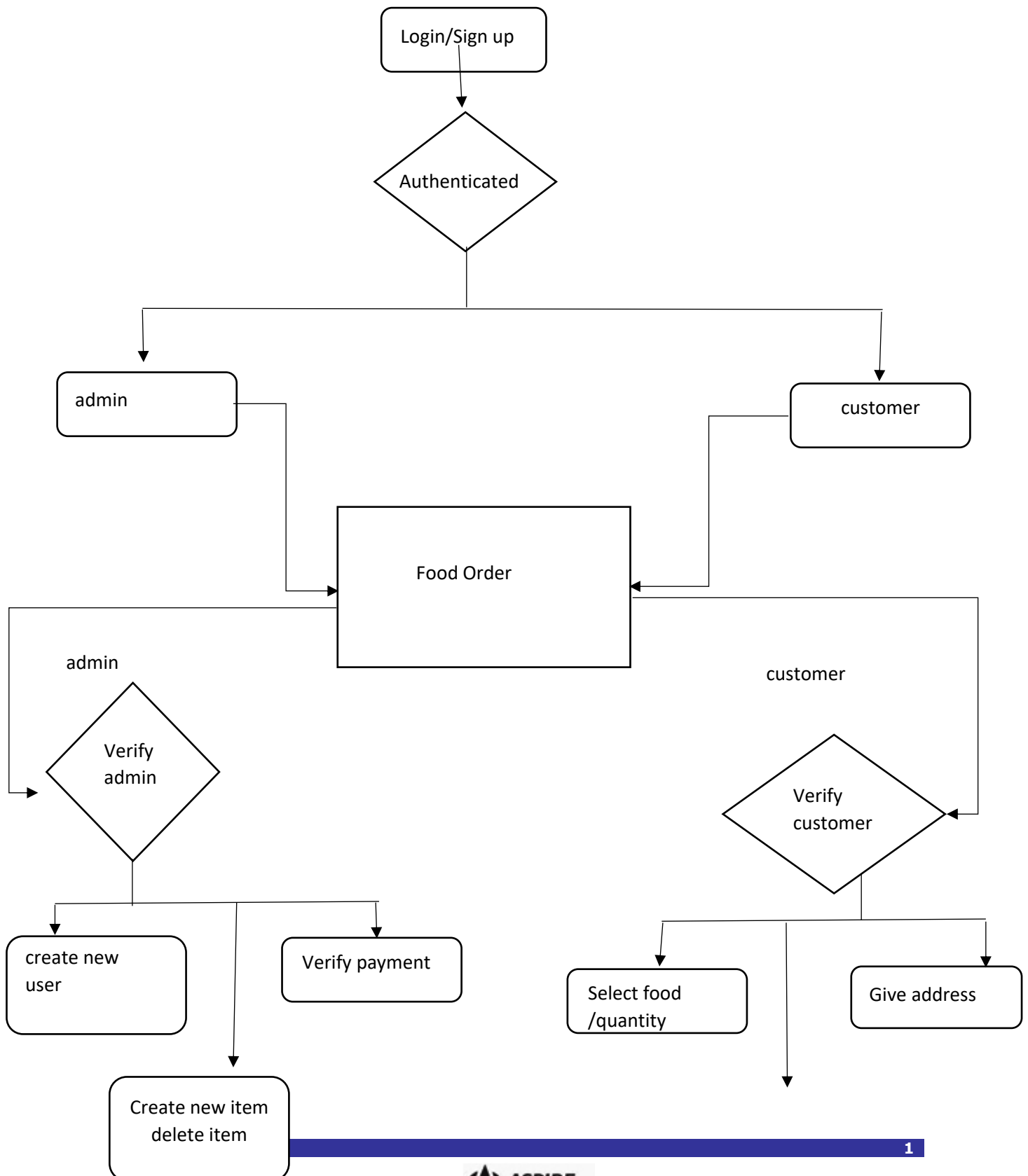


### CLASS DIAGRAM:

Admin

1



**FLOW DIAGRAM:**

Make Payment

**ER DIAGRAM:**