CHAPTER 3

ANALYSIS AND DESIGN

3.1 USE CASE DIAGRAM

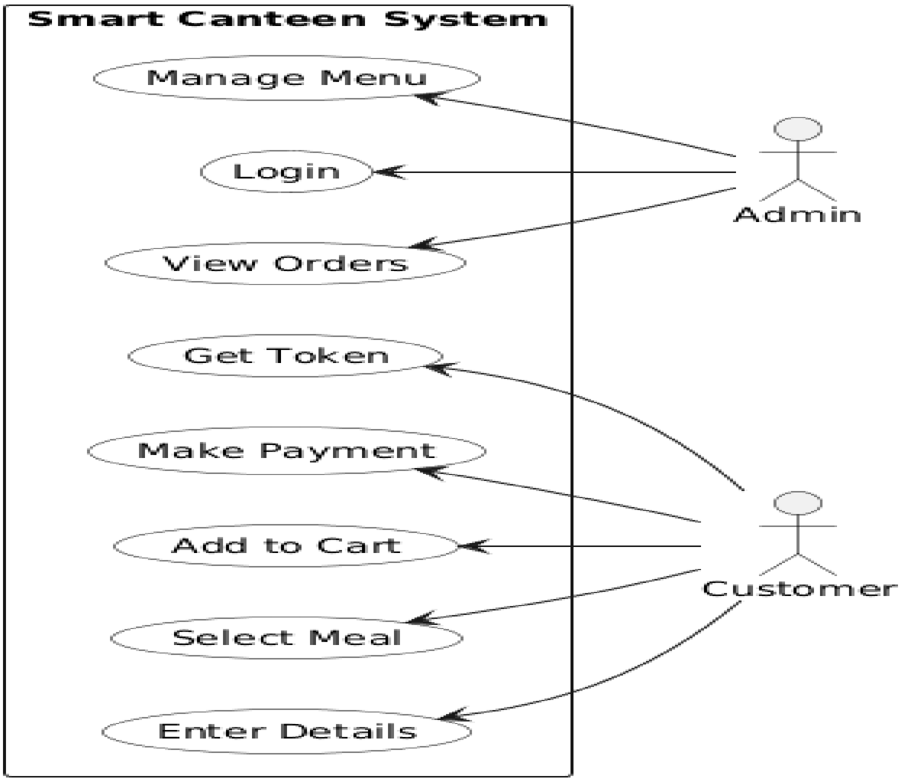

Figure 3.1 Use Case Diagram

3.1.1 USE CASE DESCRIPTION

The diagram represents a Use Case Diagram for a Smart Canteen System, showing interactions between the system and its users. There are two main actors: Admin and Customer. The Admin is responsible for managing the system, which includes logging in, managing the menu by adding or updating items, and viewing all customer orders. The Customer interacts with the system to place orders, starting with entering personal details and selecting meals from the menu. Once the meals are selected, they can be added to a cart and payment can be made through the system. After payment, the customer receives a token for order collection. The lines connecting actors to use cases illustrate these interactions clearly.

3.2 CLASS DIAGRAM

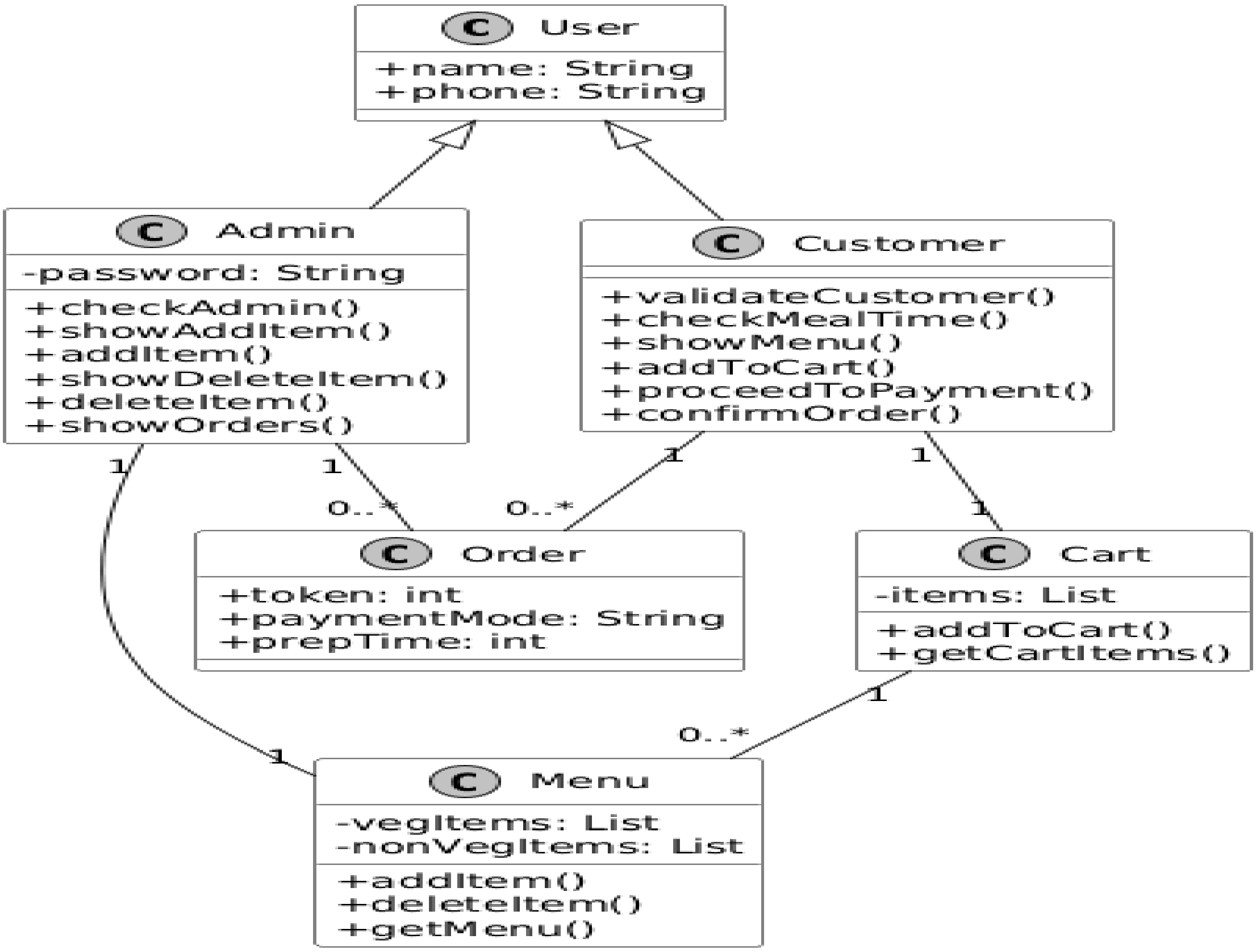

Figure 3.2 Class Diagram

3.2.1 CLASS DIAGRAM DESCRIPTION

The class diagram for the Smart Canteen Management System shows the main entities and their relationships. The User class is the parent, with subclasses Admin and Customer. Admin manages the menu and orders using methods like addItem(), deleteItem(), and showOrders(). Customer can select meals, add items to the Cart, and proceed to Order with methods like addToCart() and confirmOrder(). The Menu class stores veg and non-veg items, while Order keeps details like token, paymentMode, and prepTime. Cart maintains selected items for the customer. The diagram shows inheritance, associations, and multiplicities, clearly representing how users, menu, cart, and orders interact in the system.

3.3 ACTIVITY DIAGRAM

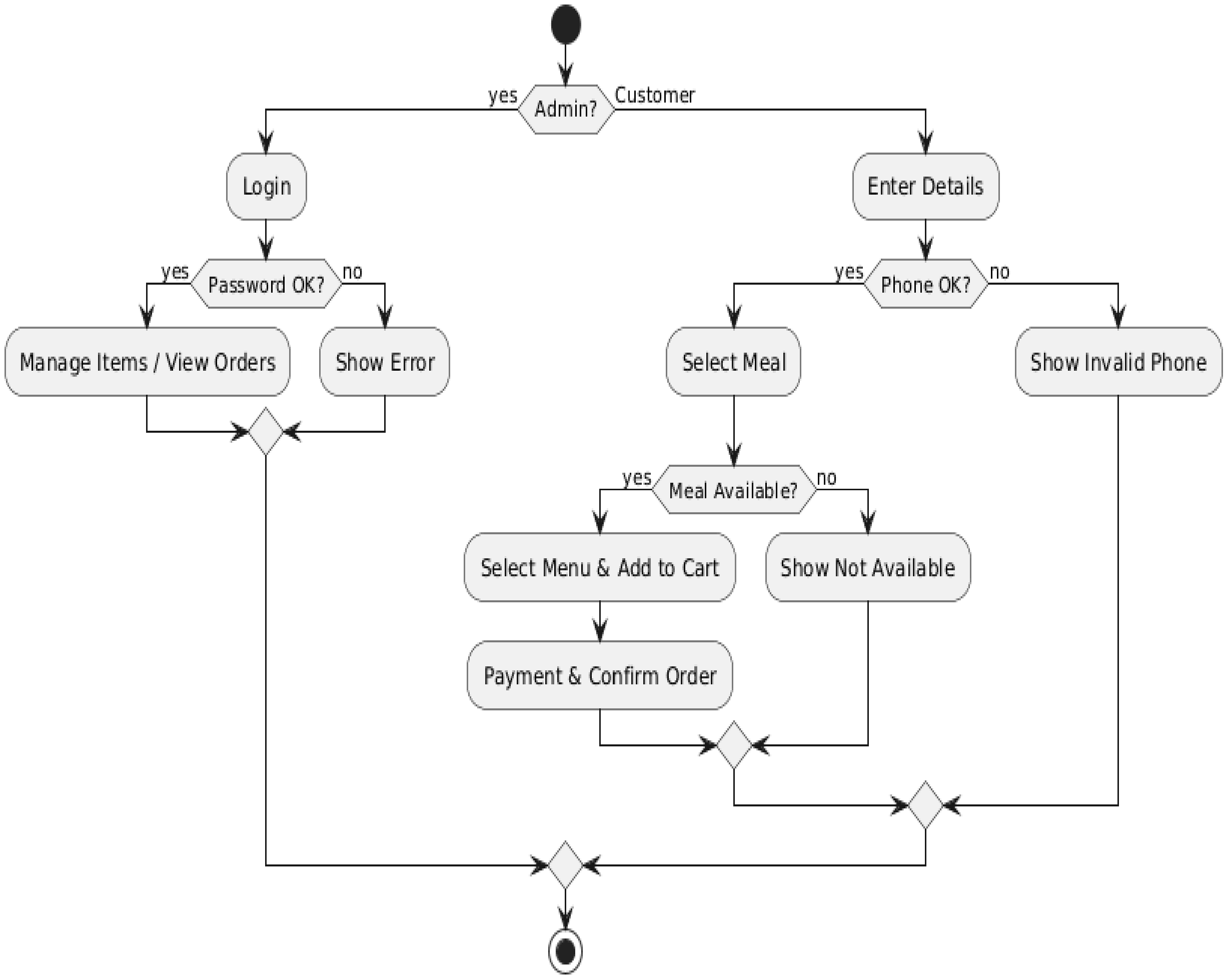

Figure 3.3 Activity Diagram

3.3.1 ACTIVITY DIAGRAM DESCRIPTION

The diagram represents the flow of a Smart Canteen Management System. It starts with identifying whether the user is an admin or a customer. If the user is an admin, they must log in and the system checks if the password is correct. Upon successful login, the admin can manage items or view orders; if the password is incorrect, an error message is displayed. For customers, they first enter their details and the system validates the phone number. If the phone number is invalid, an error message is shown. Once verified, the customer can select a meal, and the system checks its availability. If the meal is available, it is added to the cart, followed by payment and order confirmation. If the meal is not available, a “not available” message is displayed.

3.4 SEQUENCE DIAGRAM

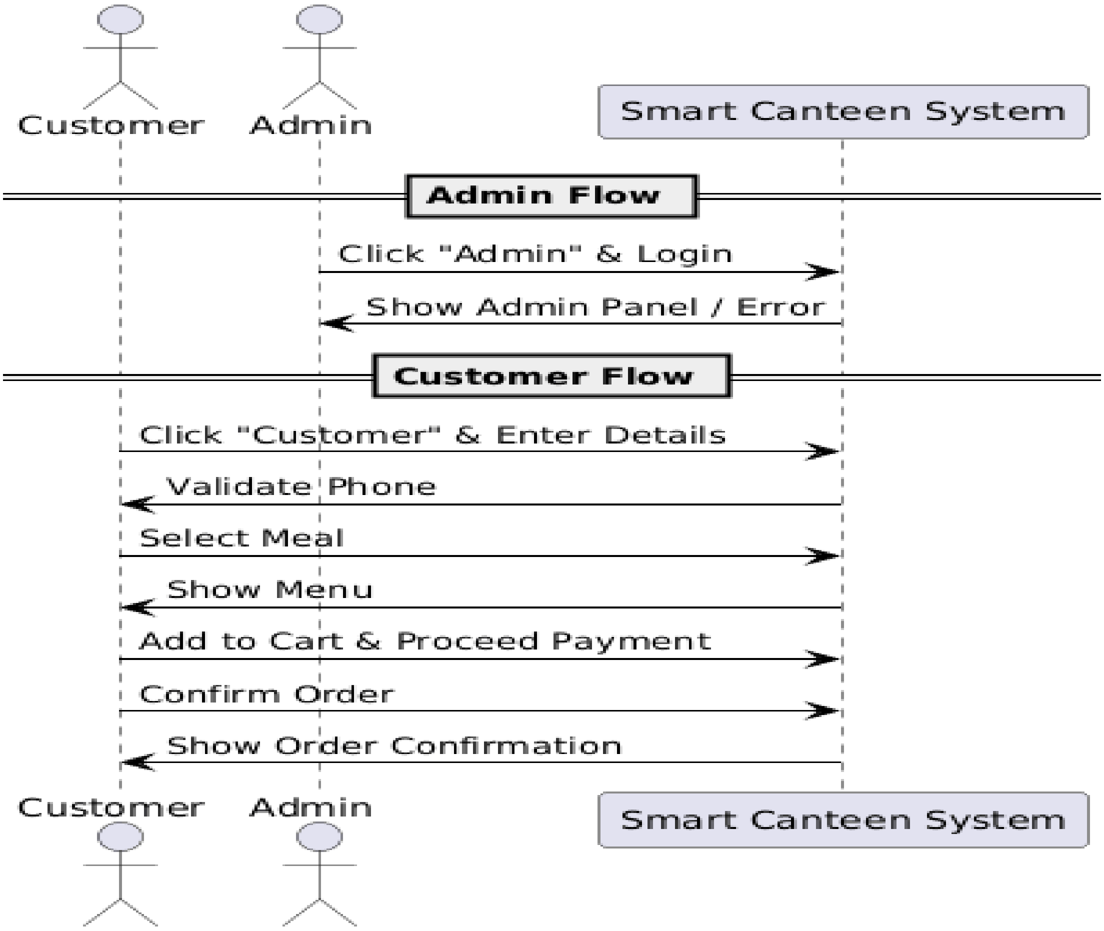

Figure 3.4 Sequence Diagram

3.4.1 SEQUENCE DIAGRAM DESCRIPTION

The sequence diagram shows how a customer interacts with the Smart Canteen System from login to order confirmation. The customer first logs into the system, which verifies the credentials and confirms successful login. The customer then views the menu, and the system retrieves and displays item details from the Menu component. When the customer places an order, the system checks item availability with the Menu and receives confirmation. After this, the system sends a payment request to the Payment Gateway, which processes the payment and returns a success message. Finally, the system confirms the order to the customer, completing the entire ordering and payment workflow.

3.5 STATE MACHINE DIAGRAM

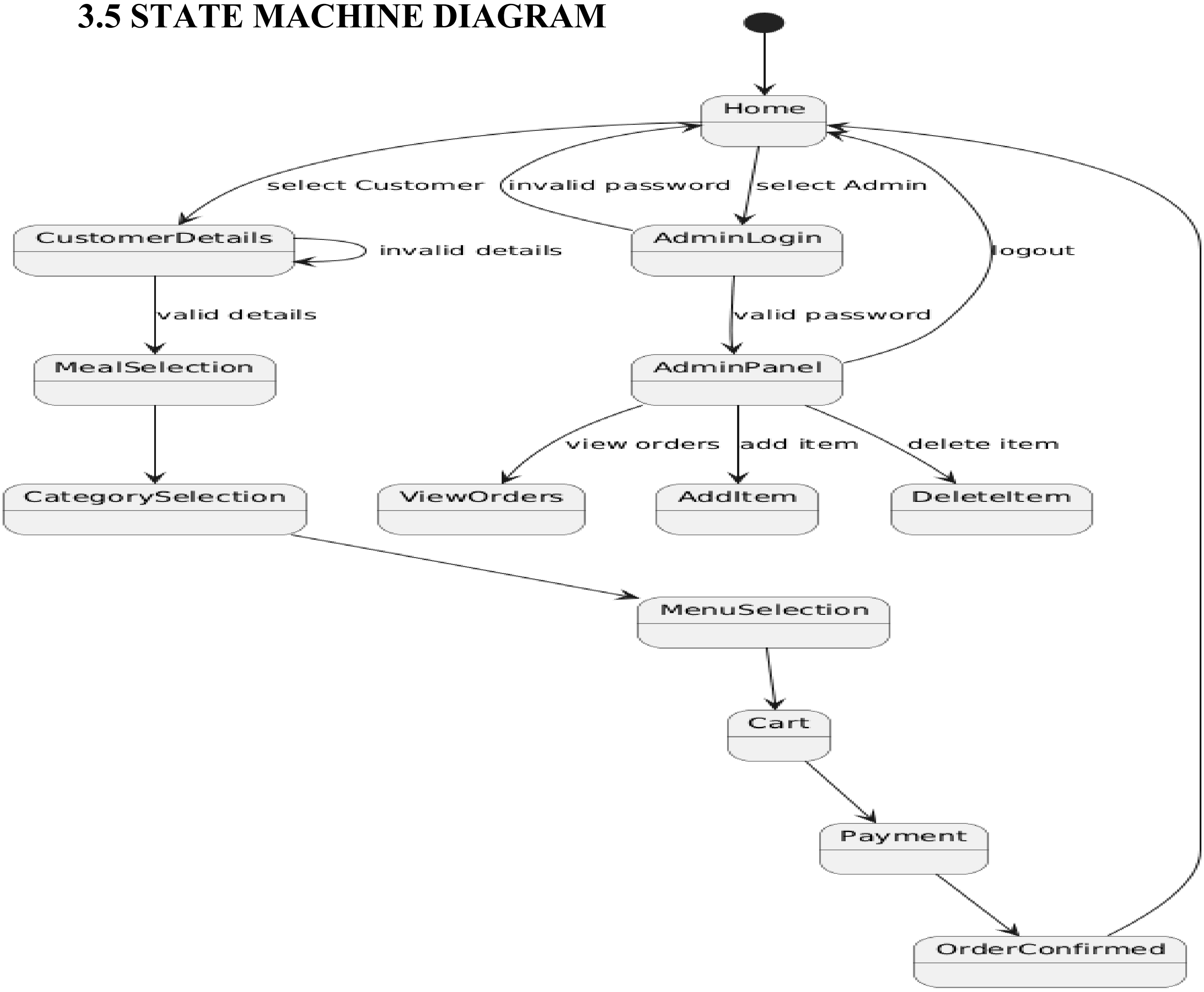

Figure 3.5 State Machine Diagram

3.5.1 STATE MACHINE DIAGRAM DESCRIPTION

The state machine diagram shows how the Smart Canteen system moves between different states based on user actions. The system begins at the Home state, where users choose either Admin or Customer roles. Admins log in to access the Admin Panel and manage items or view orders, then return to Home. Customers enter details, select meals, add items to the cart, and proceed to payment. After order confirmation, the system returns to the Home state, completing the workflow.

3.6 COMPONENT DIAGRAM

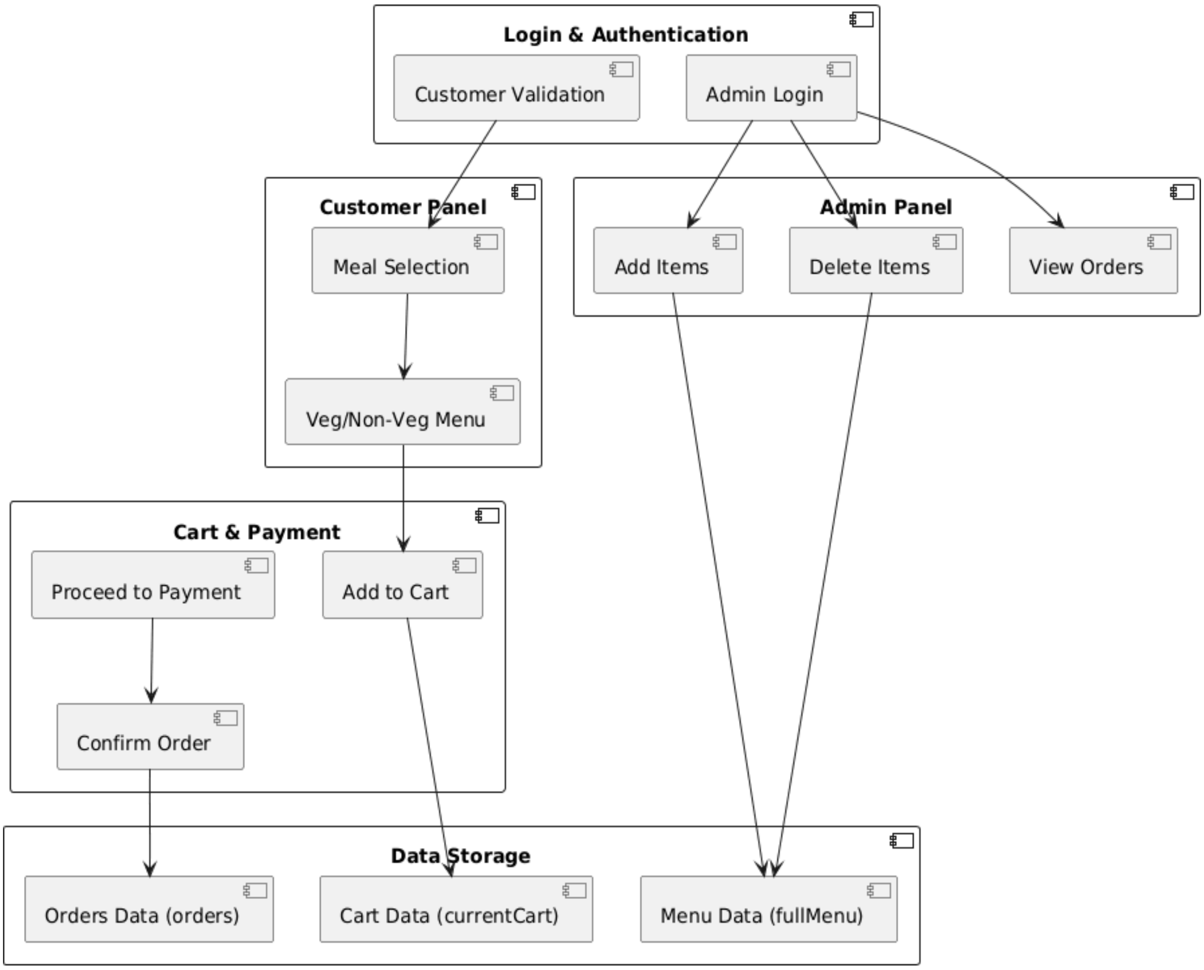

Figure 3.6 Component Diagram

3.6.1 COMPONENT DIAGRAM DESCRIPTION

The diagram shows the modular structure of the Smart Canteen System. It starts with the Login Module for admin authentication and customer verification. After login, admins access the Admin Panel to manage items and view orders. Customers move to the Customer Section to select meals and categories. Both admin and customer activities connect to the Meal Management module. This module handles cart operations and meal details. The process then moves to Order & Payment for completing purchases. Overall, the system separates roles and ensures smooth order completion.

3.7 DEPLOYMENT DIAGRAM

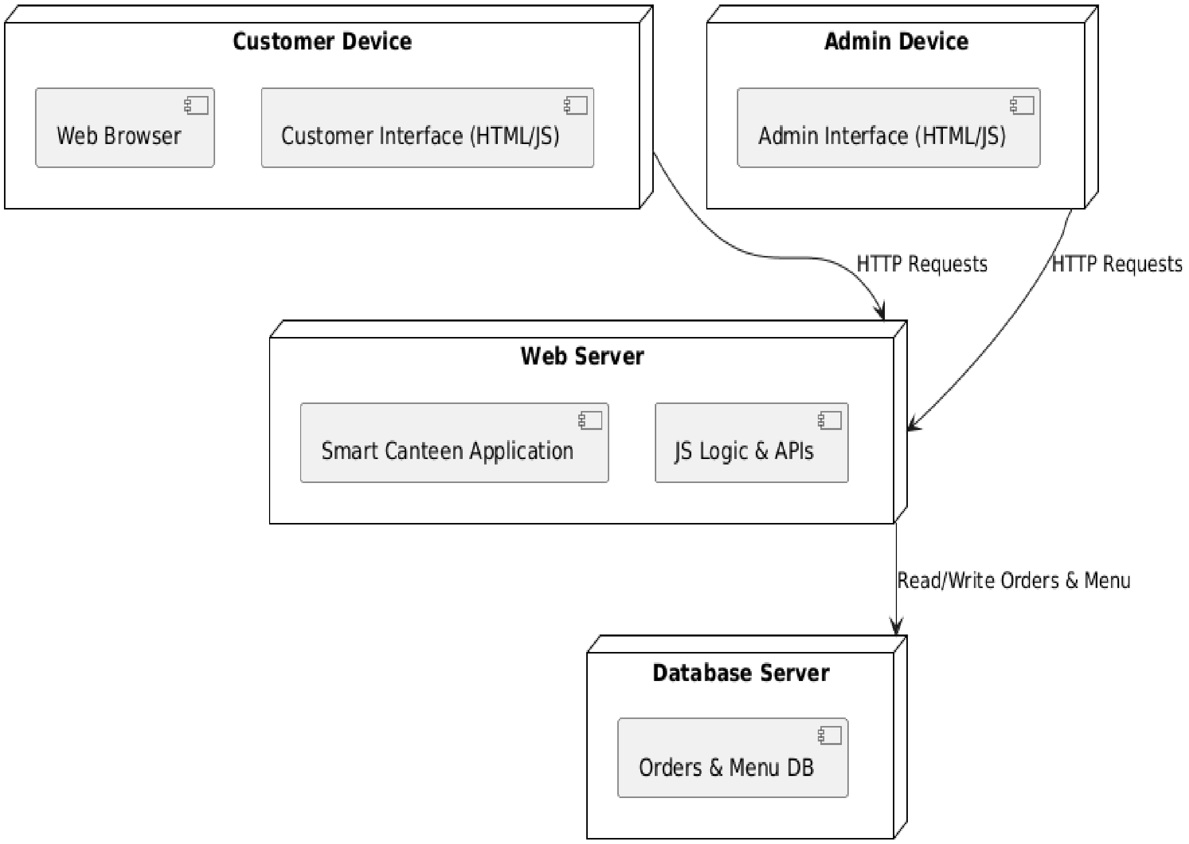

Figure 3.7 Deployment Diagram

3.7.1 DEPLOYMENT DIAGRAM DESCRIPTION

The deployment diagram shows how the Smart Canteen System is physically deployed across different devices and servers. The Customer Device runs the web or mobile app interface, which sends HTTP requests to the Canteen Server. This server hosts the Application Backend, responsible for processing user actions, managing orders, and interacting with the Database to store or retrieve data. When a payment is initiated, the backend communicates with the Payment Gateway Server, where the Payment Processor handles the transaction securely. Once the payment result is returned, the system updates the database and completes the order process.

3.8 PACKAGE DIAGRAM

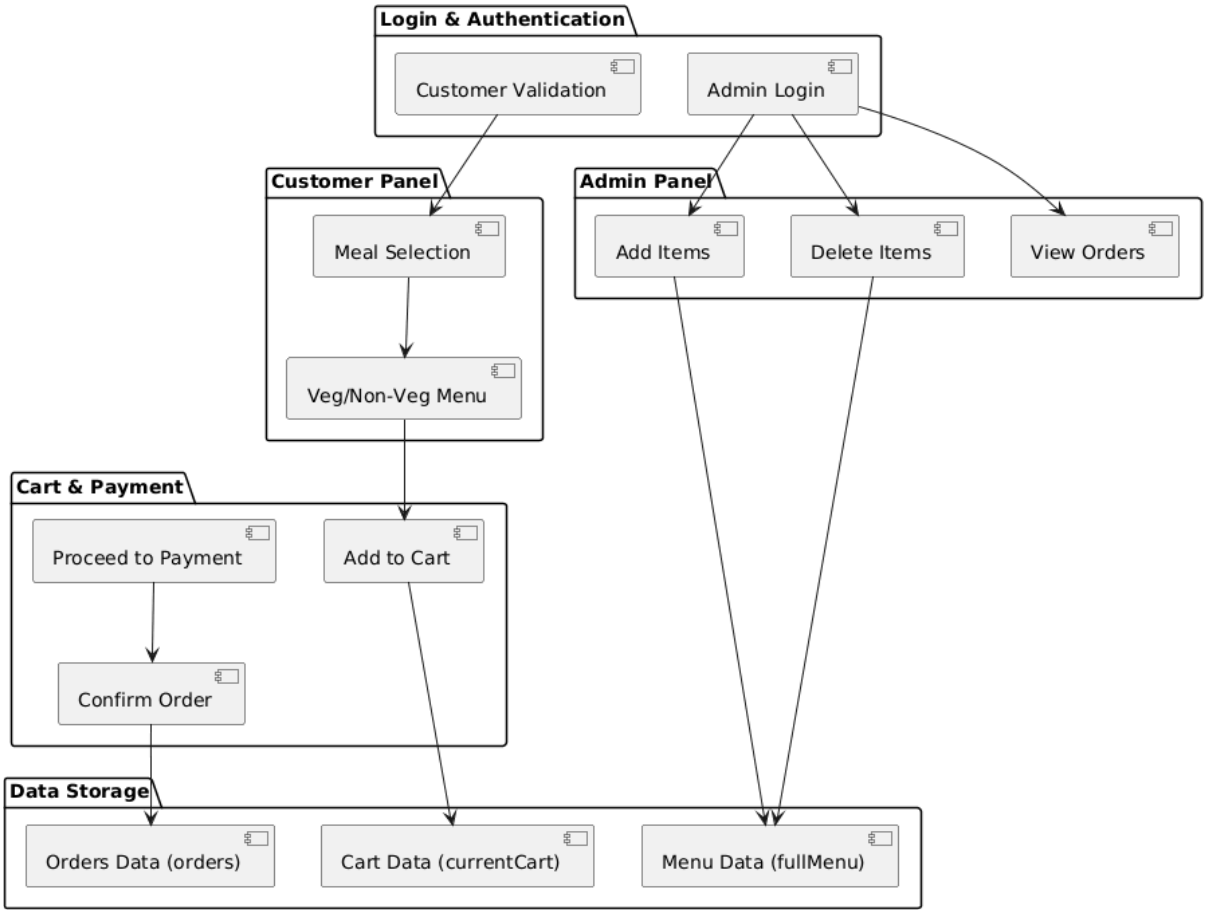

Figure 3.8 Package Diagram

3.8.1 PACKAGE DIAGRAM DESCRIPTION

The package diagram organizes the Smart Canteen Management system into five modules: Login & Authentication, Admin Panel, Customer Panel, Cart & Payment, and Data Storage. Admins log in to manage menu items and view orders, while customers validate their details to select meals and categories. The Cart & Payment module handles adding items, processing payment, and confirming orders. Data Storage keeps track of menu items, current carts, and orders. Arrows show dependencies, illustrating how user actions interact with data and different system modules, ensuring smooth workflow and modular structure.

3.9 DESIGN PATTERNS USED (GRASP, GOF)

The Smart Canteen Management system applies both GRASP and GoF design patterns to achieve a well-structured, scalable, and maintainable design. GRASP principles are clearly reflected through the Controller pattern, where JavaScript functions manage user interactions such as login, meal selection, cart updates, payment processing, and order confirmation. The Information Expert pattern is implemented by assigning data-related responsibilities to suitable structures like the menu list, cart, and order collections, which store and process relevant information efficiently. Low Coupling is achieved by separating admin operations, customer activities, and data management into independent modules, reducing interdependencies and simplifying maintenance. At the same time, High Cohesion ensures that each module focuses on a single, well-defined responsibility, improving clarity and reusability. From the GoF perspective, the Singleton pattern is conceptually applied to shared resources such as menu data and order records, ensuring consistency and avoiding duplication across the system. The Facade pattern is evident in the user interface, which offers a unified and simplified access point to complex internal operations like validation, order handling, and payment processing.