

**Aim:** To study the IEEE SRS standard and prepare SRS for the identified systems conceptualization.

## 1. Introduction

The Software Requirements Specification (SRS) outlines the functional and non-functional requirements for the development of a Food Ordering System. The system aims to provide a platform for users to browse restaurants, search for food items, place orders, and facilitate online payments. This document provides a comprehensive understanding of the system's features, constraints, and user interactions.

## 2. System Overview

The Food Ordering System is a web-based application accessible through desktop and mobile devices. It allows users to register, log in, browse restaurants, view menus, add items to the cart, place orders, and make online payments. Additionally, administrators can manage restaurants, menus, and user accounts through an admin portal.

## 3. Functional Requirements

### 3.1 User Module

**User Registration:** Users can register by providing essential details such as name, email, and password.

**User Login:** Registered users can log in using their credentials.

**User Profile Management:** Users can update their profile information, including contact details and address.

**Restaurant Search:** Users can search for restaurants based on location, cuisine, or restaurant name.

**Menu Browsing:** Users can view menus of available restaurants, including food items and prices.

**Item Selection:** Users can add food items to the cart for ordering.

**Order Placement:** Users can place orders for selected items.

**Order History:** Users can view their order history and track the status of current orders.

**Payment Gateway Integration:** Integration with a secure payment gateway to facilitate online payments.

### 3.2 Admin Module

Admin Login: Administrators can log in using their credentials to access the admin portal.

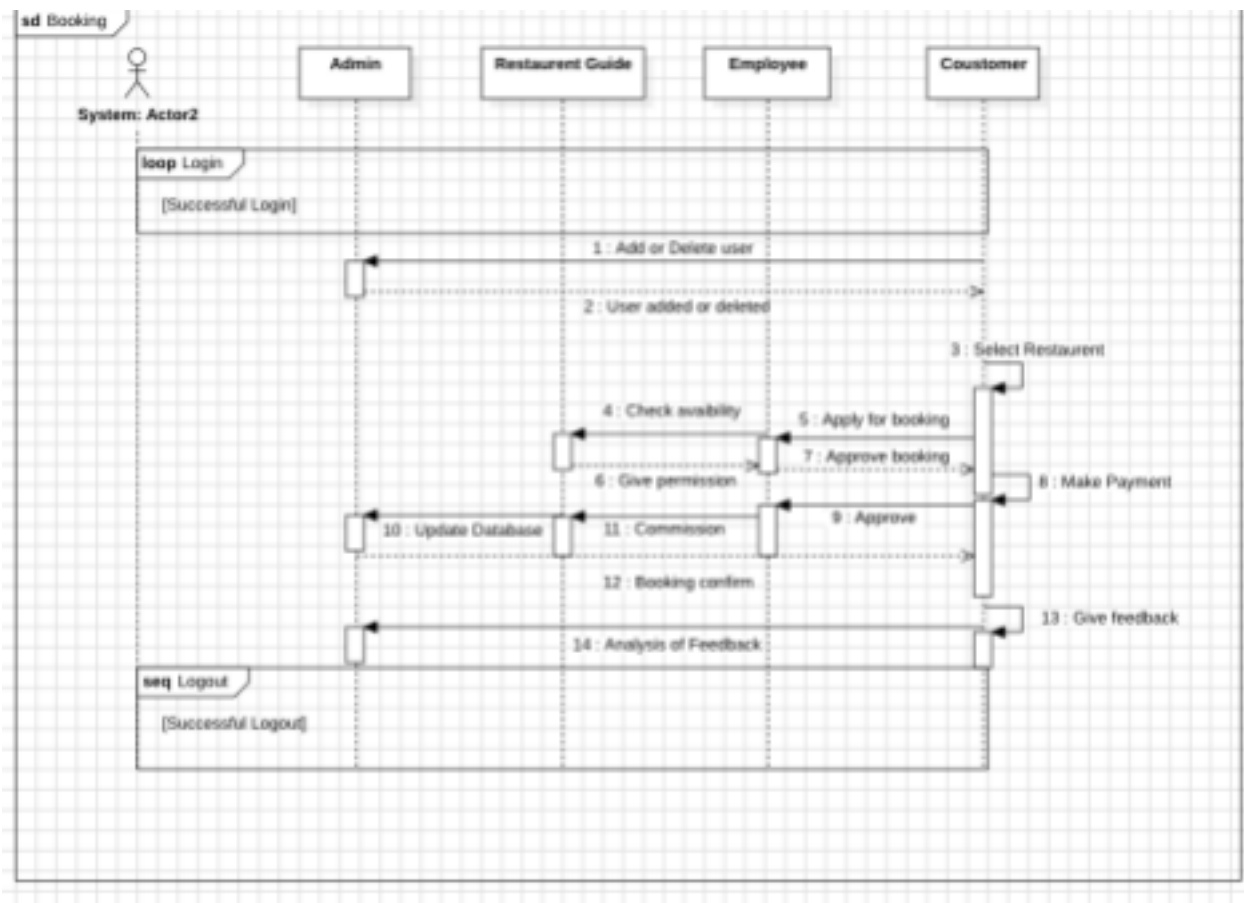
Restaurant Management: Administrators can add, edit, or remove restaurant listings.

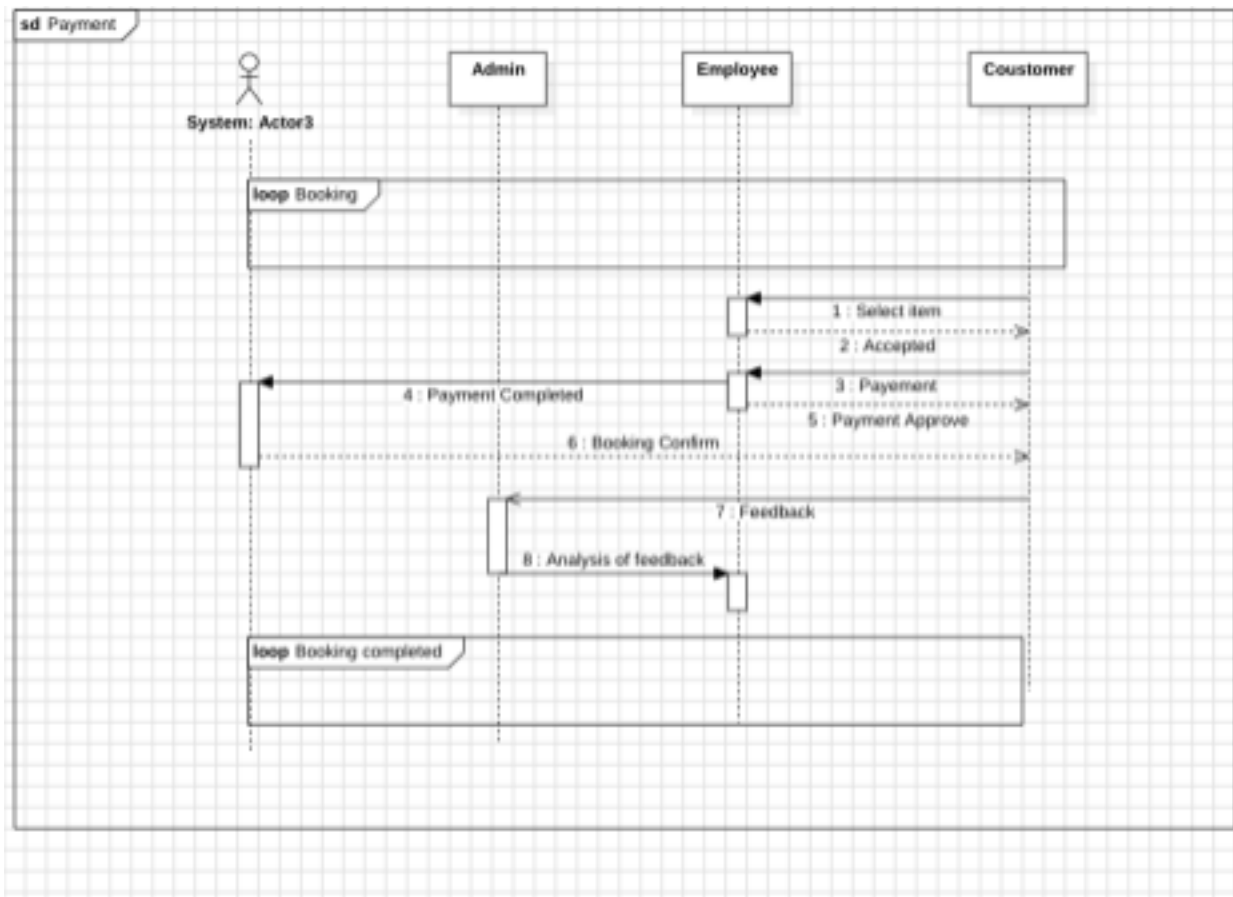
Menu Management: Administrators can manage menus, including adding, editing, or removing food items.

User Management: Administrators can view and manage user accounts, including registration details and order history.

Order Management: Administrators can view and manage orders, including order status and payment details.

We can use a sequence diagram for it:





#### 4. Non-Functional Requirements

**Performance:** The system must handle concurrent user requests efficiently to ensure responsiveness.

**Security:** User data, including personal information and payment details, must be securely stored and transmitted using encryption techniques.

**Scalability:** The system architecture should allow for easy scalability to accommodate a growing user base and increasing data volume.

**Usability:** The user interface should be intuitive and user-friendly, ensuring ease of navigation and efficient order placement.

**Reliability:** The system should be robust and reliable, minimizing downtime and ensuring uninterrupted service availability.

#### 5. Constraints

**Technology Stack:** The system will be developed using modern web development technologies such as HTML5, CSS3, JavaScript, and a backend framework like Django or Flask.

**Third-Party Integrations:** Integration with external services such as payment gateways and mapping APIs

is required for seamless functionality.

**Regulatory Compliance:** The system must comply with data protection regulations, ensuring the privacy and security of user data.

**Conclusion :** The Food Ordering System aims to provide a convenient and efficient platform for users to order food online from a variety of restaurants. By adhering to the outlined requirements and constraints, the system will deliver a seamless user experience while ensuring reliability, security, and scalability.