Online Food Ordering System

-Deepthi Sindhe S

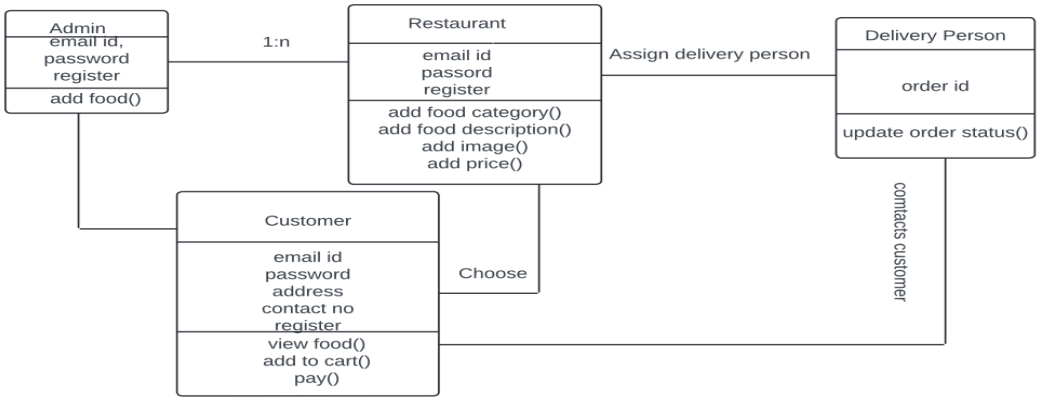
Abstract

This online food ordering system aims at providing a seamless experience to customers while ordering food from multiple restaurants. It mainly involves four different interfaces namely for the administrator, restaurant, customer and delivery person. In addition that the login and registration systems are present to ensure only authenticated users use the system.

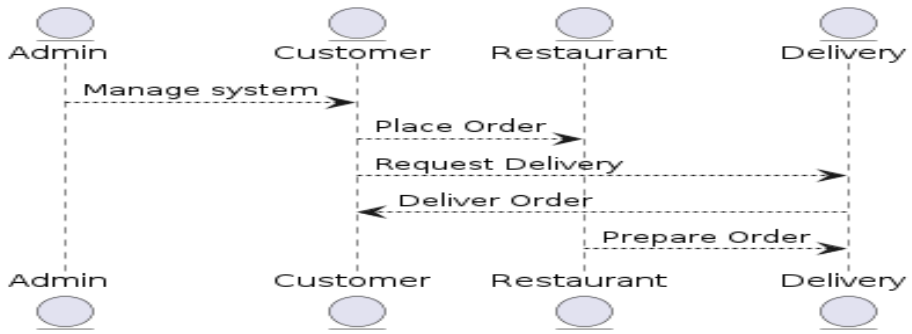
The administrator has access to add multiple food categories following which the restaurant can register itself add food items, update it and assign delivery person to a specific order. Furthermore, the customers can register to the system with the credentials such as email id, contact information and address. With that the customer can view the food items under different categories and add it to the cart. Following this the delivery person can view all the orders assigned to him or her and update the order status. Thus, this project delivers a system that runs smoothly and gives access to all type of users.

UML Diagrams

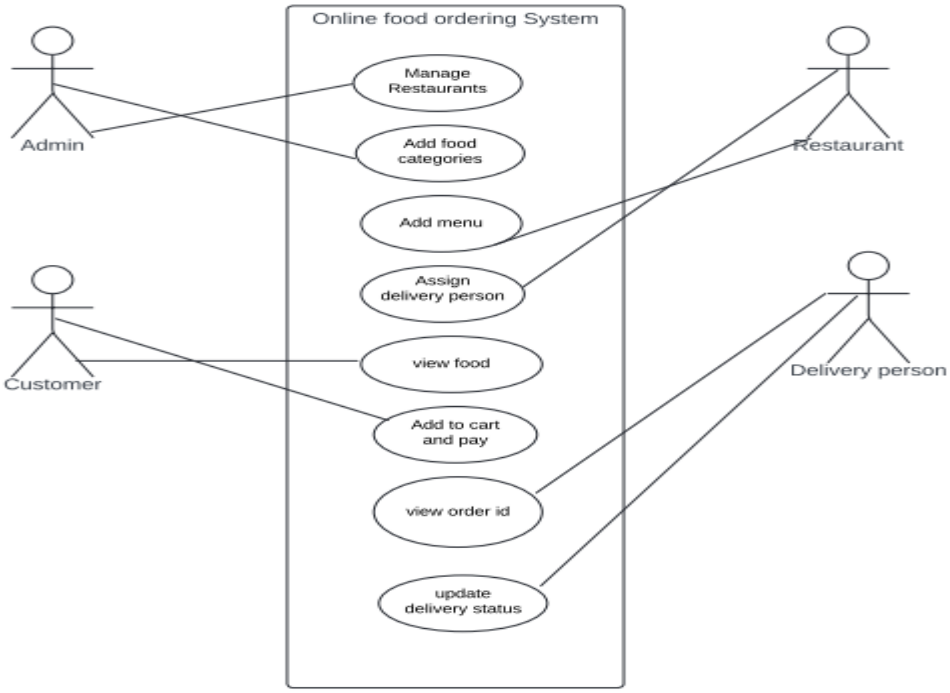
1. Class diagram



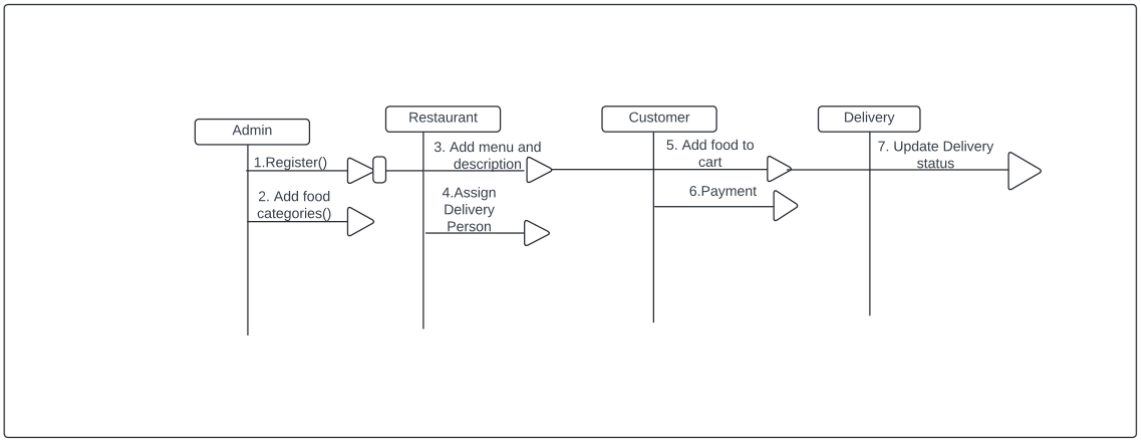
2. Data Flow Diagram



3.Use case Diagram



4. Sequence Diagram



Hardware Requirements:

1. Network Infrastructure: Ensure a stable internet connection for both development and production environments. For production, consider factors like bandwidth, load balancing, and security.
2. Storage: Depending on the scale of your application, you may need storage for the database, static files, and media uploads. Consider both the capacity and performance requirements for storage.
3. Backup System: Implement a backup system to prevent data loss in case of hardware failures or other unforeseen circumstances.

Software Requirements:

1. Python: Python must be installed on the system.
2. Web Framework: Select a web framework for building the backend of your application. Popular choices in Python include Django and Flask. Django is more feature-rich and comes with many built-in functionalities, which might be beneficial for a complex system like an online food ordering system.
3. Database: Choose a database management system (DBMS) to store data. Common choices for Python web applications include SQLite, PostgreSQL, MySQL, or MongoDB. The choice depends on factors like scalability, data structure, and deployment environment.
4. Frontend Technologies: For the frontend, you might use HTML, CSS, and JavaScript along with frameworks like React.js or Vue.js to create an interactive user interface.