ONLINE FOOD ORDERING SYSTEM

**ABSTRACT:**

The "Online Food Ordering System" is a project aimed at developing a comprehensive system to streamline the process of food ordering and delivery. This system facilitates interactions between customers, restaurants, and delivery person to ensure efficient and timely order processing and delivery.

**Key features of the "Online Food Ordering System" include:**

**User Authentication and Registration:** The registration and Login system has been added so that only authenticated users (Admin, Restaurant, Delivery, or Customer) can perform their functionalities.

**Customer Interface:** Customers can easily place orders through a user-friendly interface, providing information such as delivery address and payment details.

* + User registration.
  + Food search by name or category.
  + Cart management (adding, viewing).
  + Order history viewing.

**Restaurant Management:** Restaurants have access to an intuitive dashboard to manage menu items, view incoming orders, and assign delivery personnel.

* + Registration and management of delivery personnel.
  + Food item addition, viewing, and updating.
  + Order assignment to delivery personnel.

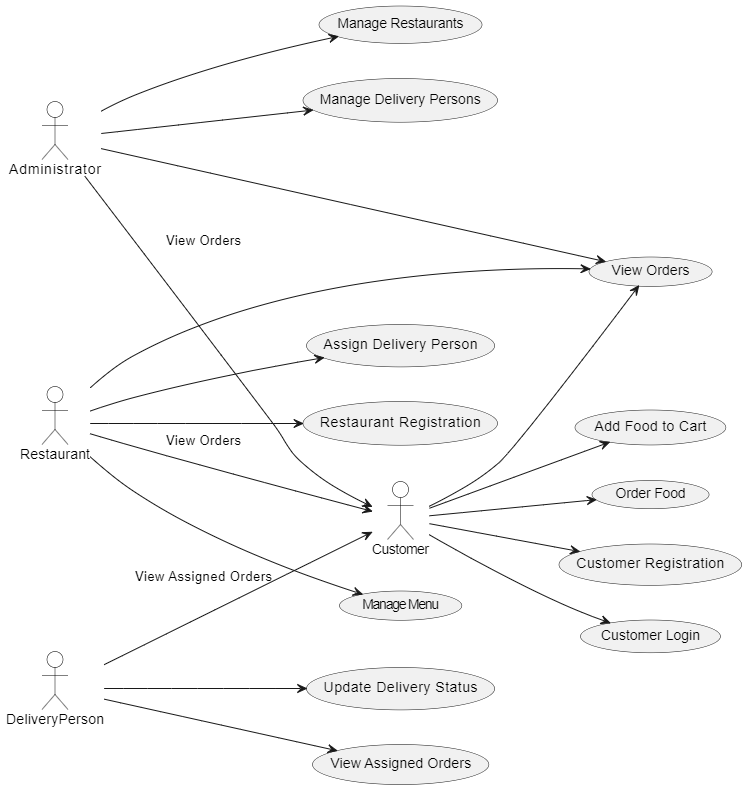
**Delivery Management:** Delivery personnel receive real-time notifications of assigned orders and can update delivery status directly through the system, ensuring seamless communication with customers and restaurants.

* + Assigned order viewing.
  + Real-time order delivery status updates.

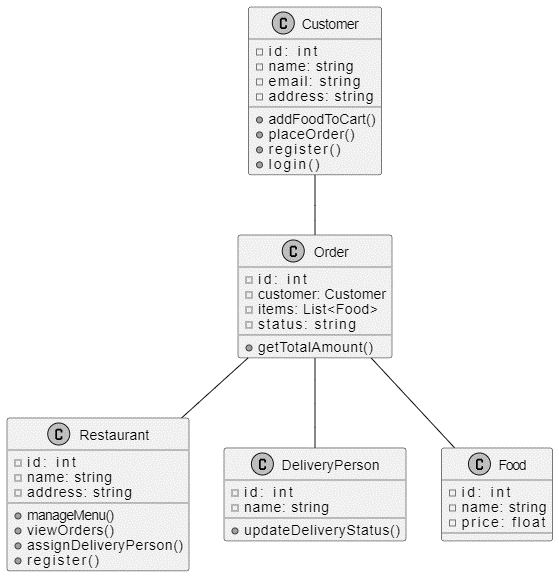
***In conclusion*** "The Order Management System" simplifies food ordering and delivery, benefiting customers, restaurants, and delivery person with its centralized approach.

**UML DIAGRAMS**

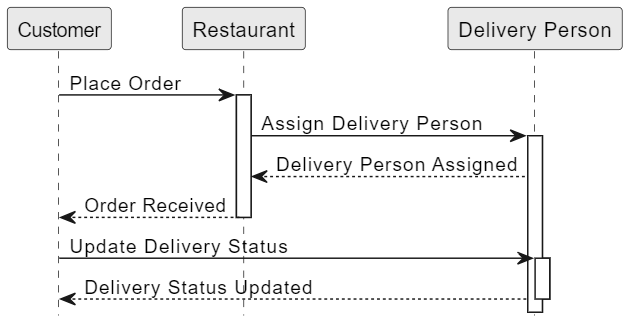
1.USE CASE DIAGRAM:



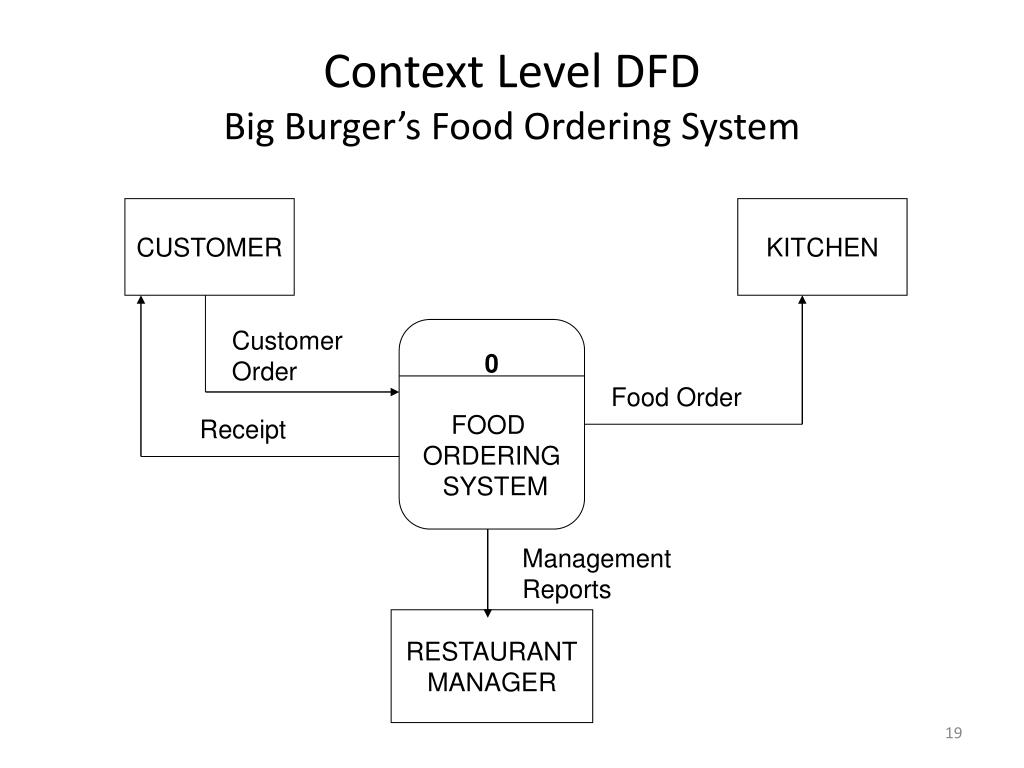
2.Class Diagram:



3.Sequence Diagram:



4.DFD Diagram:



**SOFTWARE REQUIREMENTS**

**Python:** Ensure that Python is installed on your system. It serves as the core programming language for the application.

**Web Framework:** Select a suitable web framework to build the backend of your application. Two popular choices are Django and Flask. Django is preferred for its extensive feature set and built-in functionalities, making it suitable for complex systems such as an online food ordering platform.

**Database Management System (DBMS):** Choose a DBMS to store and manage your application's data. Common options for Python web applications include SQLite, PostgreSQL, MySQL, or MongoDB. The selection should consider factors like scalability, data structure requirements, and deployment environment constraints.

**Frontend Technologies:** Utilize frontend technologies to create an interactive user interface. HTML, CSS, and JavaScript are fundamental for structuring and styling web pages. Additionally, frameworks like React.js or Vue.js can enhance frontend development, providing tools for building dynamic and responsive interfaces.

**HARDWARE REQUIREMENTS**

**Server Infrastructure:** Servers host and manage incoming requests, their specifics depend on traffic volume and application complexity, requiring load balancing and redundancy for high availability.

**Storage Solution:** Choose storage based on data type and size, like cloud services or dedicated servers, for images, user uploads, and logs.

**Networking Equipment:** Utilize routers, switches, and firewalls for seamless server communication and high-speed internet connectivity for efficient request handling.

**Backup and Recovery Systems:** Essential for data protection, implement regular backups and redundancy measures to minimize downtime during hardware failures.