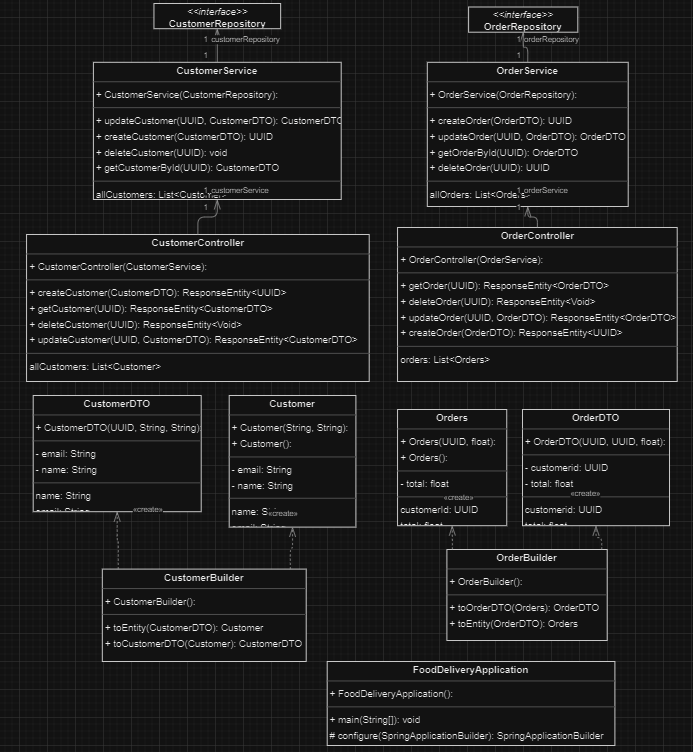
Food Delivery Application

**Description:** The project is designed to simulate a food delivery application, powered by Java Spring backend technology and React as frontend technology. At its core, the app offers comprehensive CRUD (Create, Read, Update, Delete) operations to manage various entities within the platform. The operations have been implemented for the Customer, Order and the User models.

**UML Class diagram:**

**Different Logins for Admin and User:**

In the food delivery application, different roles are implemented to segregate responsibilities and access levels within the system. The two primary roles are ADMIN and USER, each with distinct permissions and capabilities.

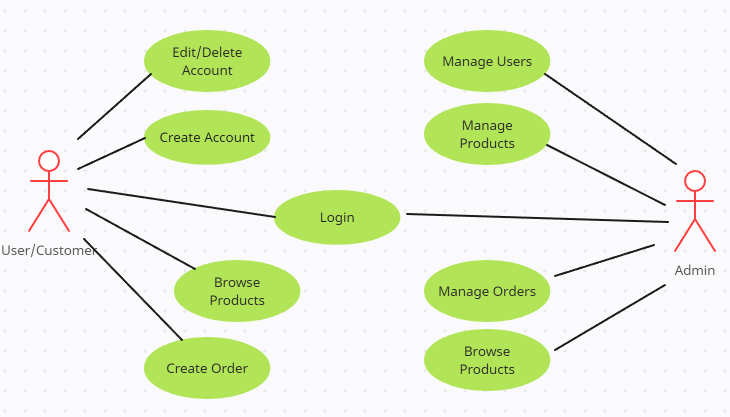
**Admin Role:**

* Admins have elevated privileges within the system compared to regular users.
* Admins can create, read, update, and delete customer information. This includes tasks such as adding new customers, updating their details, or removing inactive accounts.
* Admins have access to all orders placed on the platform. They can view order details, update order status, and manage any issues related to orders such as cancellations or refunds.
* Admins are responsible for maintaining the menu available on the platform. They can add new items, update existing ones, or remove items that are no longer available.
* Admins handle customer complaints, feedback, or any other issues that may arise during the operation of the application.

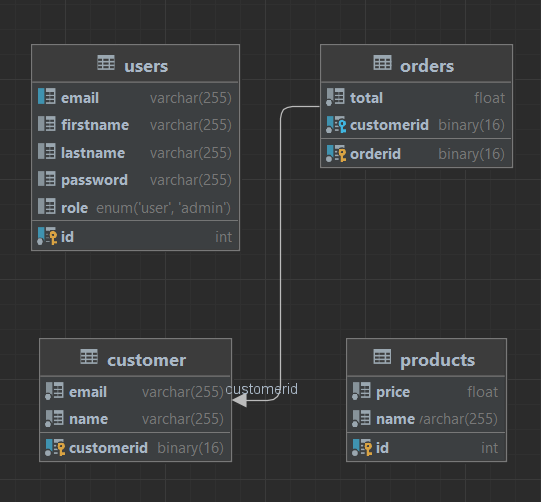
**User Role:**

* Users are the regular customers of the food delivery application.
* Their primary function is to browse the menu, place orders, and manage their personal information.
* Users can view the list of available food items, along with their descriptions and prices.
* Users can select items from the menu, specify quantities, and place orders for delivery.
* Users can access their order history to review past purchases and track the status of current orders.
* Users can update their profile information, such as contact details or delivery addresses.

By implementing separate login mechanisms for users and admins, the application maintains security and ensures that users only access functionalities appropriate for their roles. This approach enhances usability and streamlines operations for both regular users and administrative staff.

**Use Case diagram:**

**Database diagram:**



**Frontend:**The frontend of the application is built using React and React Router. The application is structured with several key components, each serving a distinct purpose to ensure a comprehensive and efficient user interface. The LoginPage and RegisterPage components handle user authentication, offering easy access to the platform and secure account creation. The HomePage acts as the central hub, guiding users to various features of the application. The ProductsPage and its nested components (ListProducts, CreateProduct, UpdateProduct, DeleteProduct) provide a robust interface for managing products, enabling users to view, add, edit, and delete product entries with ease. Additionally, the OrdersPage allows users to manage orders efficiently. Utilizing React Router's Routes and Route components, the application ensures smooth navigation and a responsive user interface, making the overall user experience intuitive and engaging.