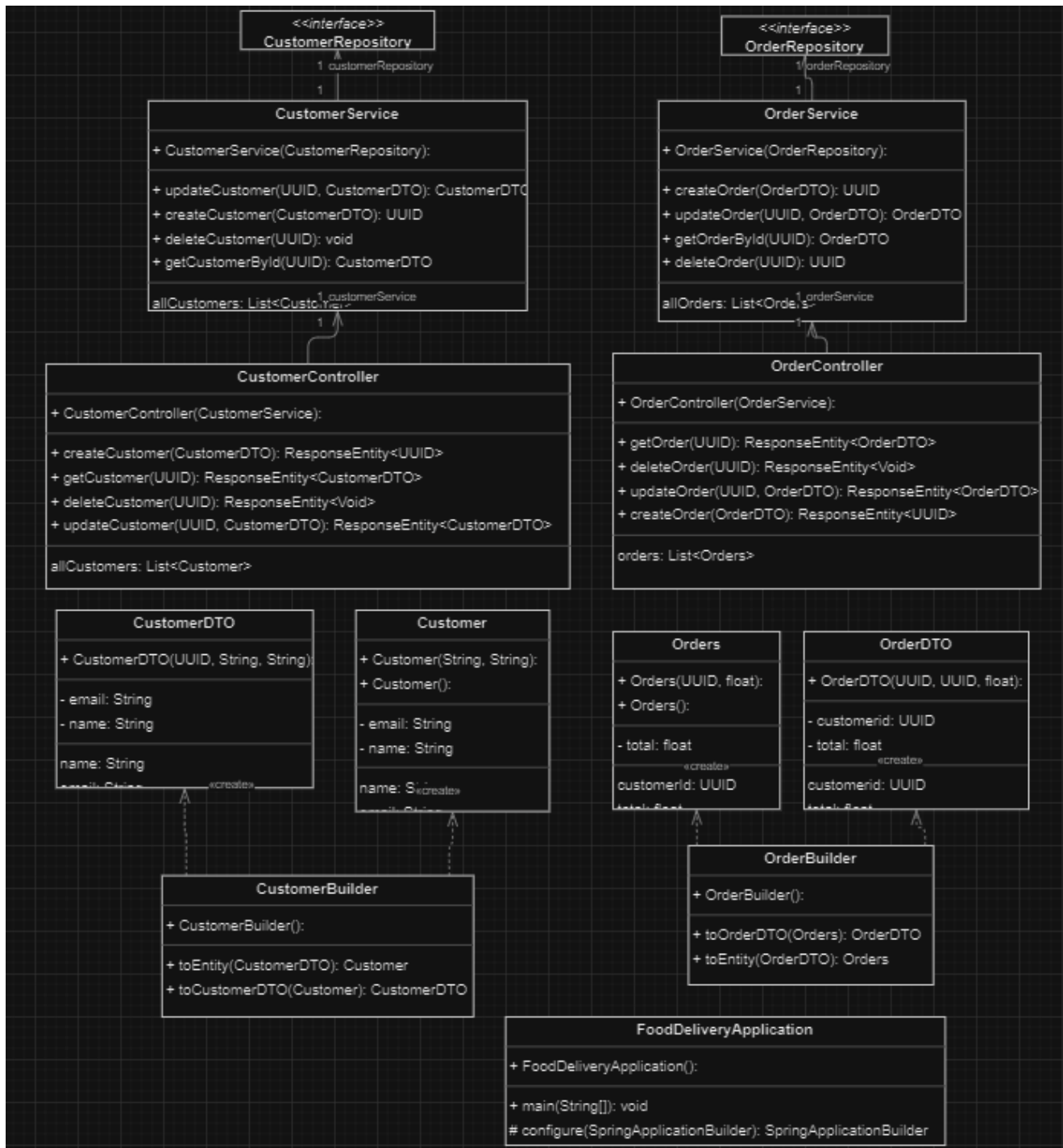


Food Delivery Application

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:

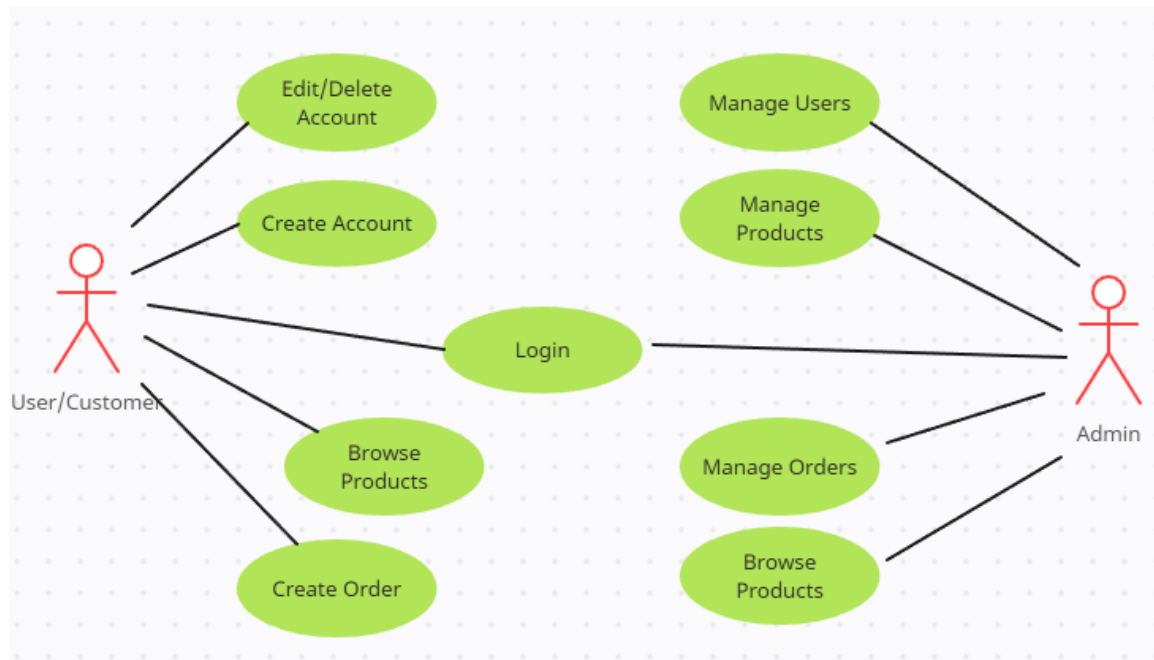
- Admins have elevated privileges within the system compared to regular users.
- Admins can create, read, update, and delete customer information. This includes tasks such as adding new customers, updating their details, or removing inactive accounts.
- Admins have access to all orders placed on the 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:

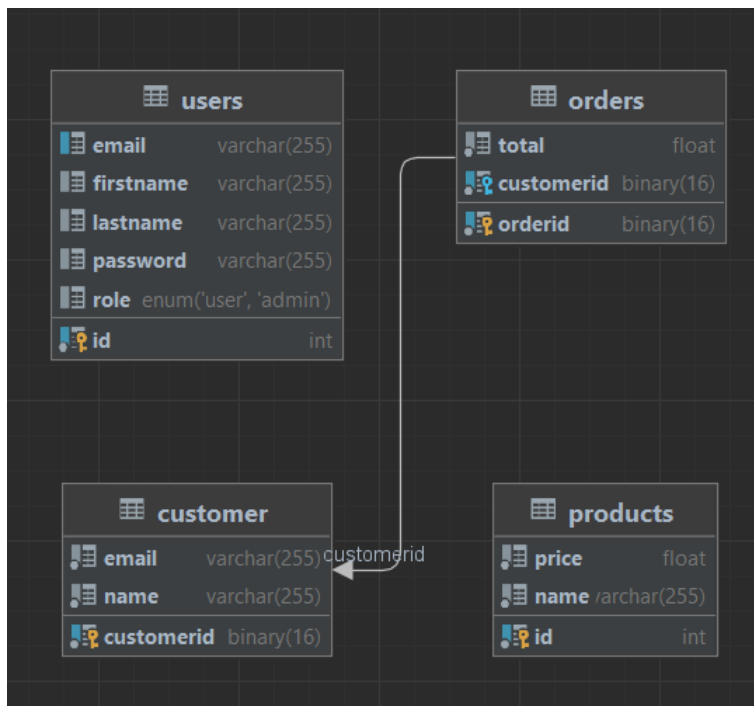
- Users are the regular customers of the food delivery application.
- Their primary function is to browse the menu, place orders, and manage their personal information.
- Users can view the list of available food items, along with their descriptions and prices.
- Users can select items from the menu, 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.

Backend Architecture:

The backend of the Food Delivery Application is developed using Java Spring, providing a robust and scalable foundation for managing the business logic and data operations. The application follows a RESTful architecture, allowing seamless communication between the frontend and backend components via HTTP requests. The backend contains multiple components:

- **Controllers:** they handle the HTTP requests and delegate them to the appropriate service layer for processing: “ProductController”, “UserController”, “OrderController”
- **Services:** they encapsulate the business logic of the application and interact with the data access layer: “ProductService”, “UserService”, “OrderService:
- **Repositories:** they provide an interface for performing CRUD operations on the database entities: “ProductRepository”, “UserRepository”, “OrderRepository”
- **Entities:** they represent the core data objects of the application and are mapped to the database tables: “Product”, “MyUser”, “Order”

Database Integration:

The application utilizes an underlying relational database management system (RDBMS) to persist data. It employs technologies like Hibernate ORM for object-relational mapping, ensuring seamless interaction between Java objects and database entities.

Frontend Architecture:

The frontend of the Food Delivery Application is built using React, a popular JavaScript library for building user interfaces. React Router is used for client-side routing, enabling navigation between different components of the application without page refreshes. The frontend components are:

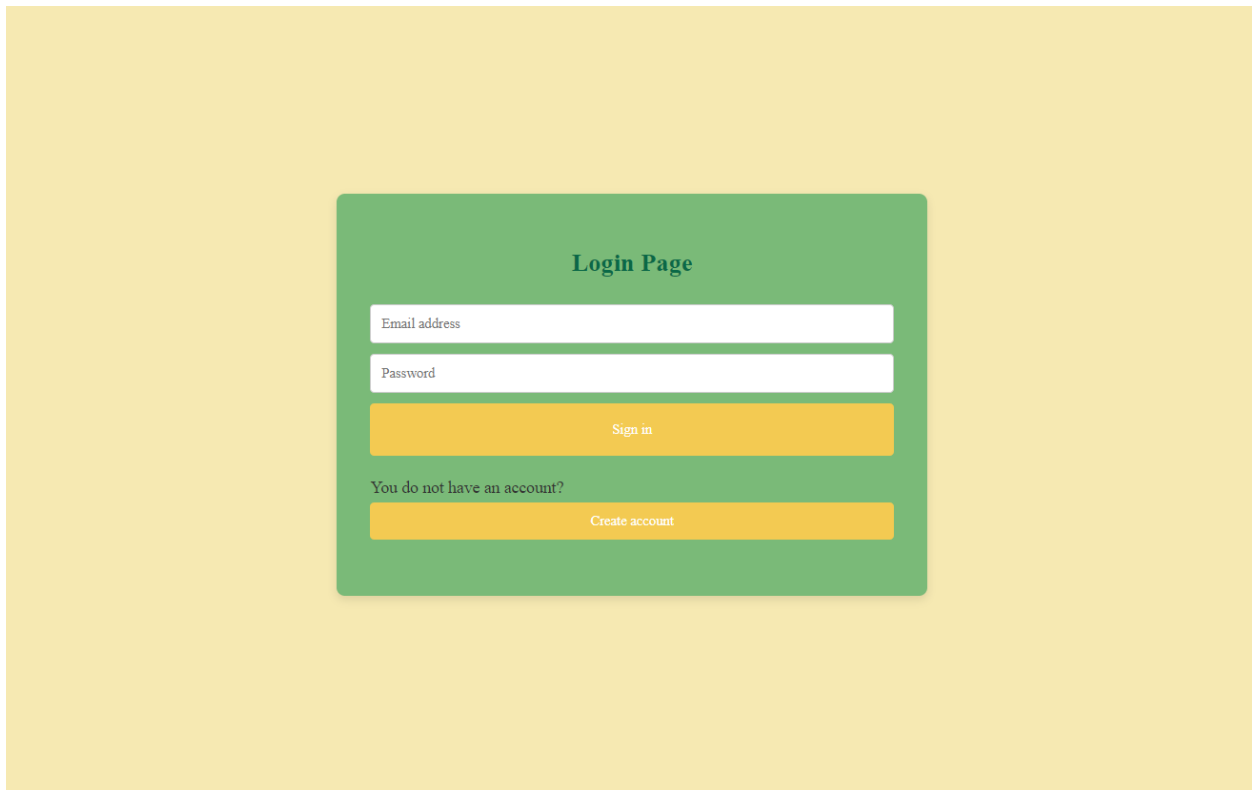
- **LoginPage:** Allows users to authenticate and access the application by providing their credentials.

- **RegisterPage:** Facilitates user registration by capturing necessary information such as username, email, and password.
- **HomePage:** Serves as the central hub of the application, providing navigation to various features and functionalities.
- **ProductsPage:** Enables users to see products and add them to an order.
- **OrdersPage:** Allows users to manage their orders, view their total and update products quantity in current order.

Security:

The application prioritizes security by implementing mechanisms such as password hashing, input validation, and HTTPS encryption to safeguard sensitive data and prevent unauthorized access.

Screenshots from application:



Register Page

Role:

Register

Already have an account? [Login](#)

Welcome to the Home Page!

Logout

Products

Orders

Product List

Search Product:

Refresh

Go Back

View Orders

Chips Lays - 13

Delete

Add to Cart

Update

Shaorma - 125

Delete

Add to Cart

Update

Paine - 214

Delete

Add to Cart

Welcome to the Orders Page!

[Go To
HomePage](#)

[Go Back](#)

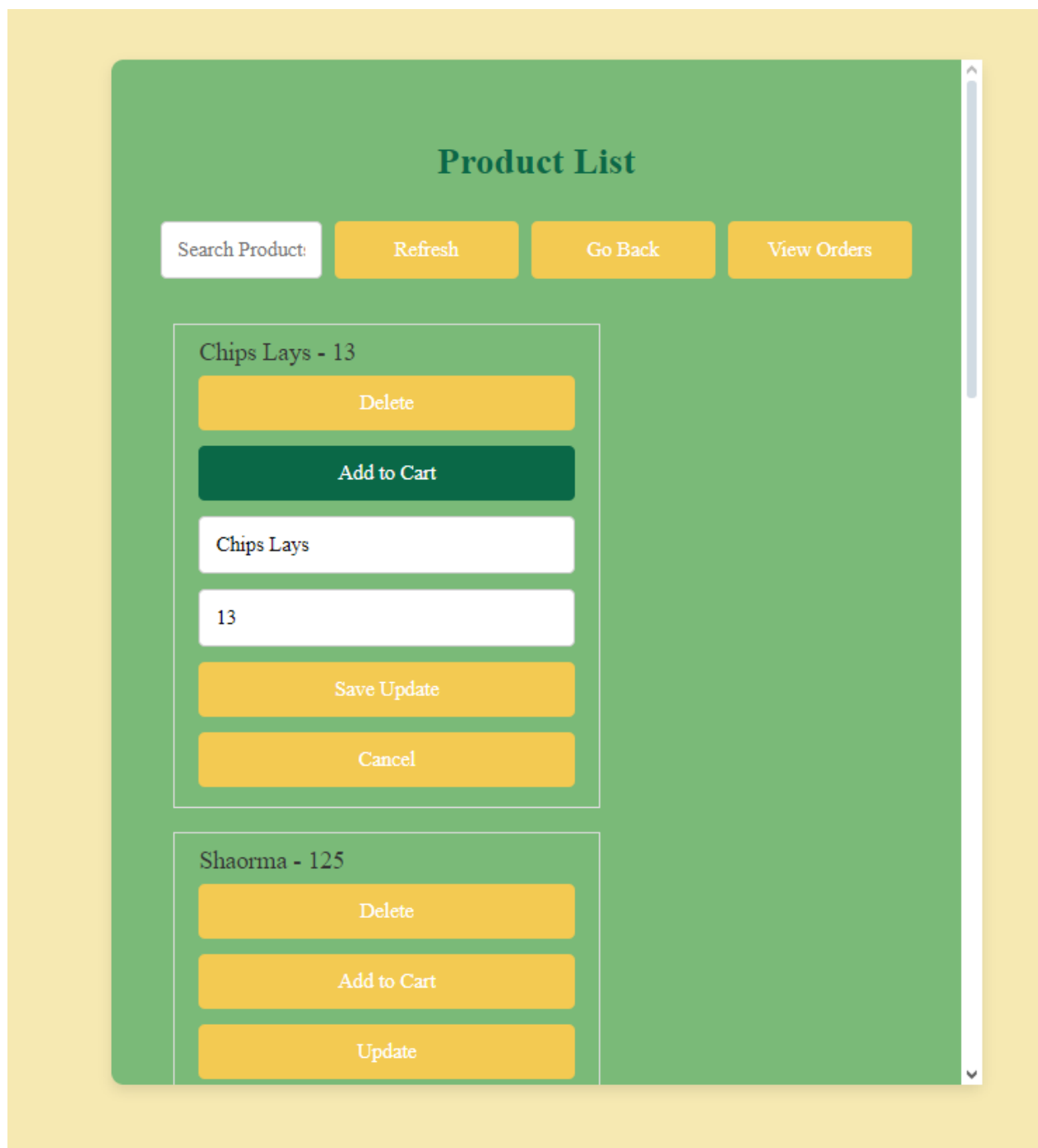
[Logout](#)

Current Order

Chips Lays	13 LEI	-	1	+
Shaorma	125 LEI	-	1	+
Paine	214 LEI	-	1	+

Total: 352 LEI

[Order Products](#)



Conclusion:

It was a very interesting and nice project to do, while I had some difficulties with different operations, over all it went well and the result is nice. The project could easily be upgraded and improved by adding categories at products, pictures, a prettier style, track the order and so on.