UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Food Delivery App

Name:Hulea Andrei-Florin
Group:30235

Table of Contents

Deliverable 1

Project Specification

Implement and design an online Java Spring application for food delivery where the users can make an order and wait for the courier to deliver it. There are 3 types of users: common users who can only order food and wait for it to be delivered, workers who deliver the food and administrators who manage the database.

Functional Requirements

User:

-Make a food order

-Place or cancel the food order

-View the current order or the order that has been placed

-View all restaurants and their menus

-Generating a bill when ordering

Admin:

-Create/Read/Update/Delete operations on all entities

-Generate .xml file for the restaurants

Use Case Model

Use Cases Identification

<u>Use-Case: Buying food</u>

Level: High

Primary Actor: User

Main success scenario: login -> make the order -> place the order

Extensions:login -> wrong password or username

login -> remaining stock value is insufficient

<u>Use-Case: Delivering food</u>

Level: High

Primary Actor: Worker

Main success scenario: login->deliver the food

Extensions:login -> wrong password

<u>Use-Case: Administrating the database</u>

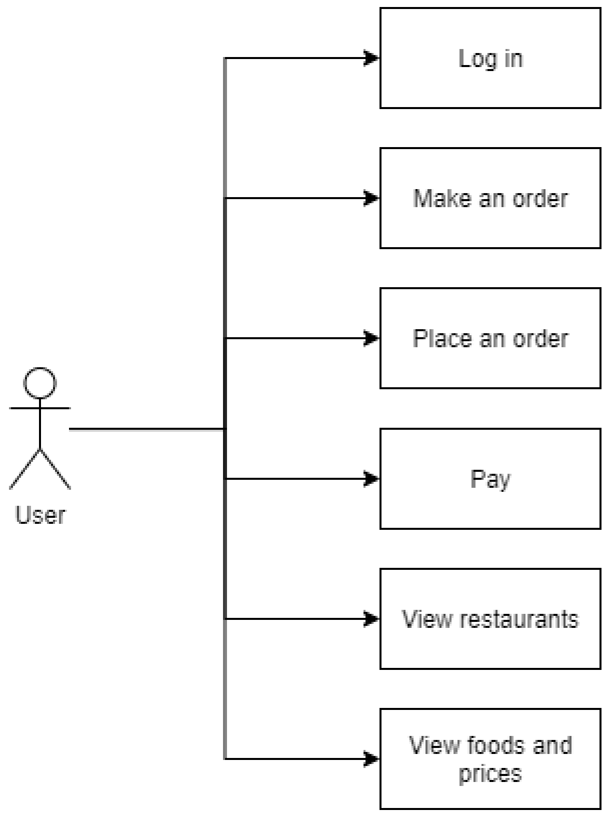
Level: High

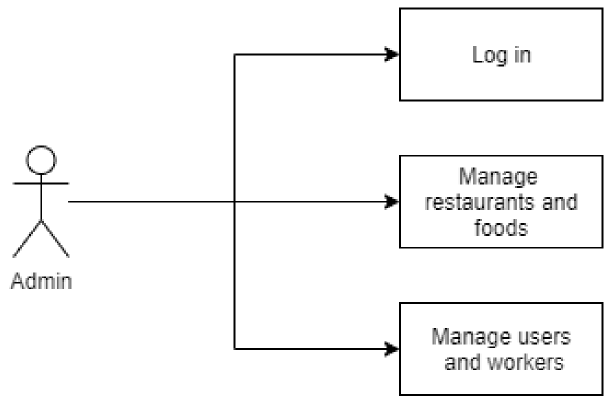
Primary Actor: *Admin*

Main success scenario: *login->modify the necessary values from the database*

Extensions:login -> wrong password

UML Use Case Diagrams





Supplementary Specification

Non-functional Requirements

-Portability:

The application will be usable from a variety of environments and compatible with their systems. It will run on computers, laptops, smartphones and mostly anything with an internet connection and a minimum of computational capability.

-Reliability:

Because of the way it’s built, the application can run without any failures for a long period of time. Also it can be accessed by anyone at any time, if the server is running.

-Security:

Security is a must in any application that works with money and contains personal information.

-Localization:

The application will find the restaurants from the database which are the closest to the user.

Design Constraints

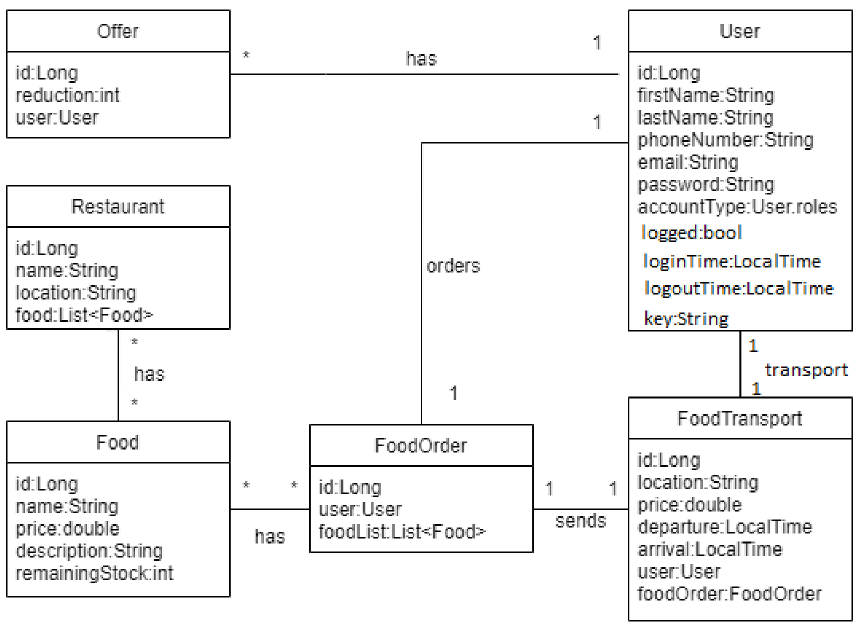
-For Object-Relational Mapping, Java Persistence API(JPA) is used. It is based on Hibernate and it lets us define which object should should be persisted. When an object is persisted its name becomes the name of the table and its fields become columns.
-To avoid the repetition of creating setters,getters and constructors the application uses lombok, which works on JPA entities.

-Spring Boot was also used to create a maven project, with all the needed dependencies. Spring is the most popular framework for Java development. It is lightweight and provides easy inversion of control and consequently.

-The application also uses Crud Repositories, which is at the base foundation of most dynamic websites and implements basic operations such as save,find,delete,count,etc.

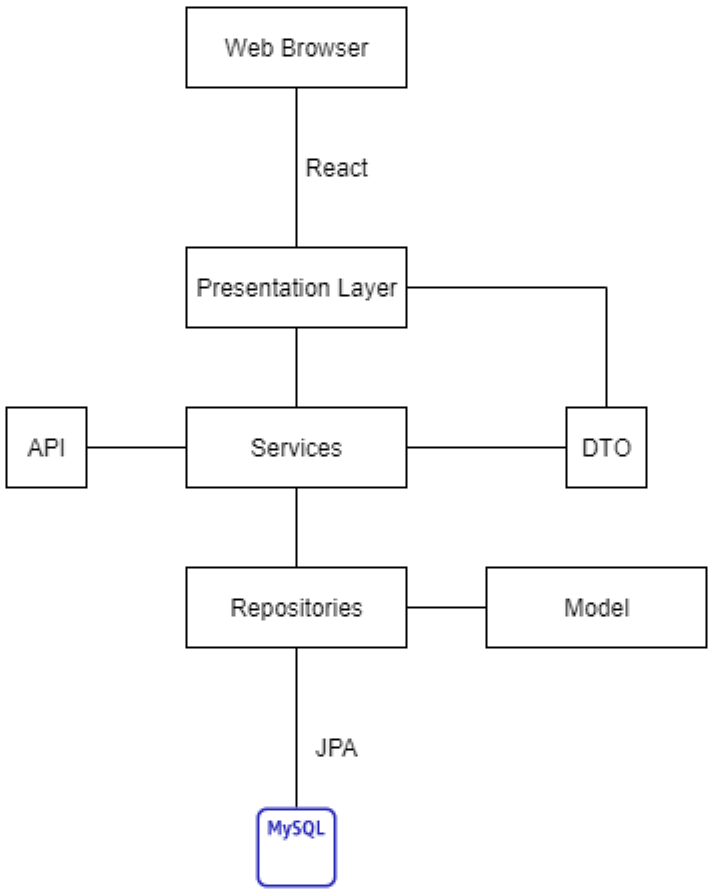
Deliverable 2

Domain Model

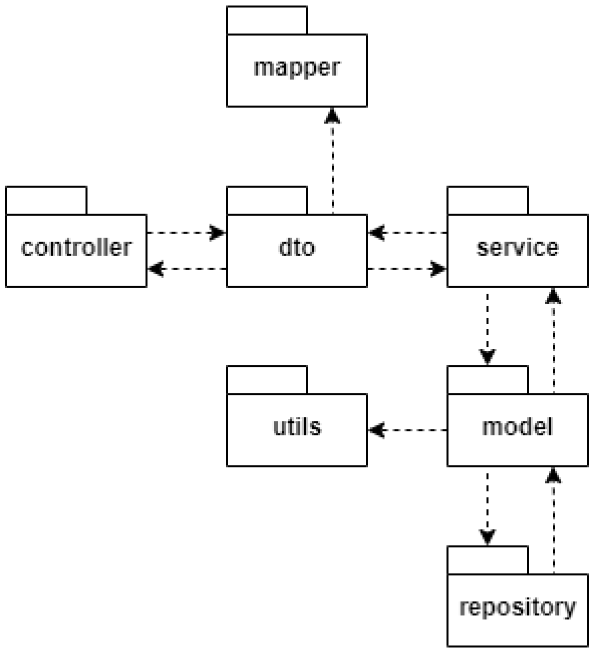


Architectural Design

Conceptual Architecture

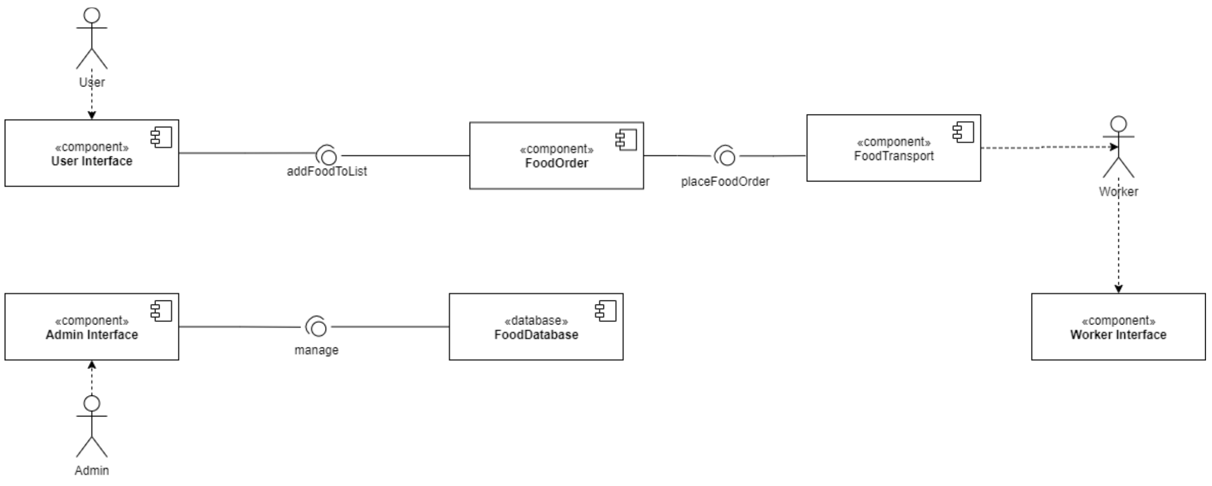


Package Design

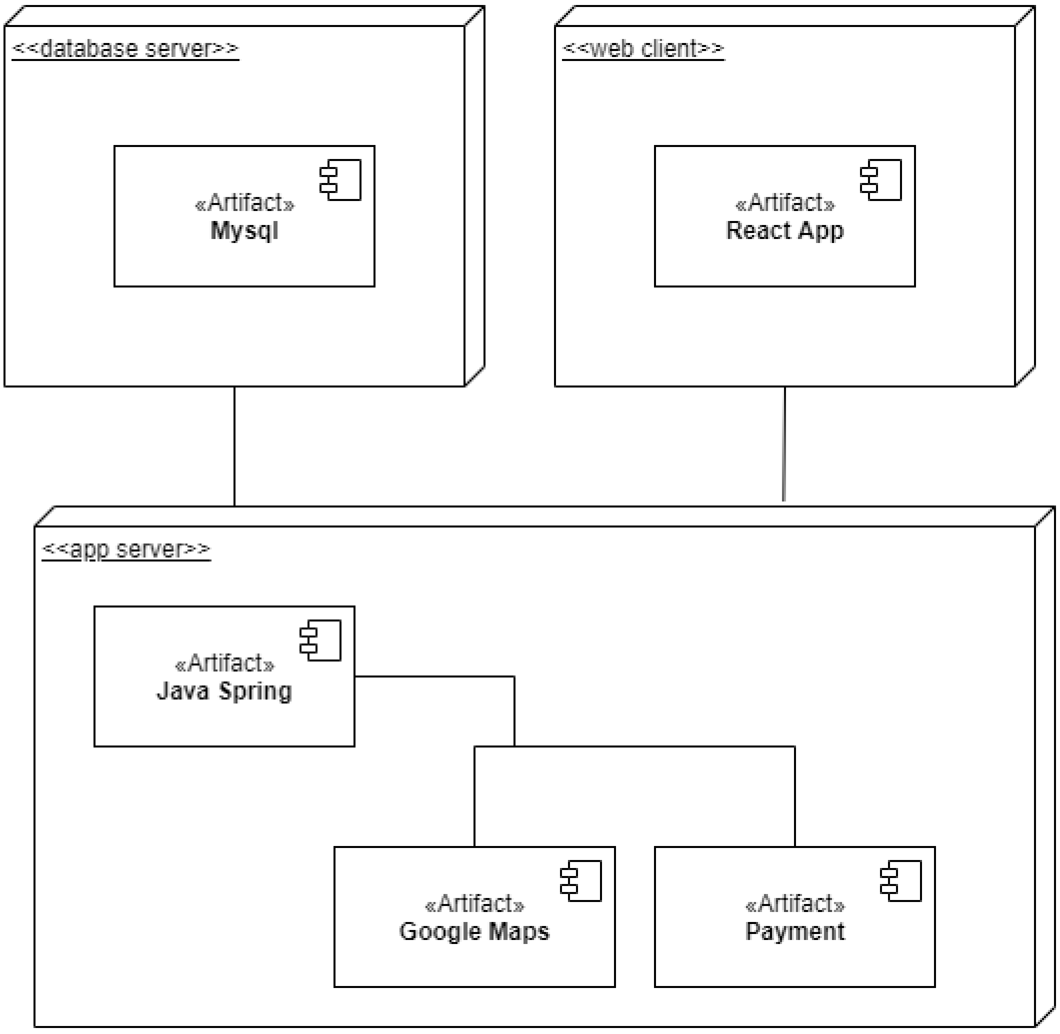


Component and Deployment Diagram

Component diagram



Deployment diagram

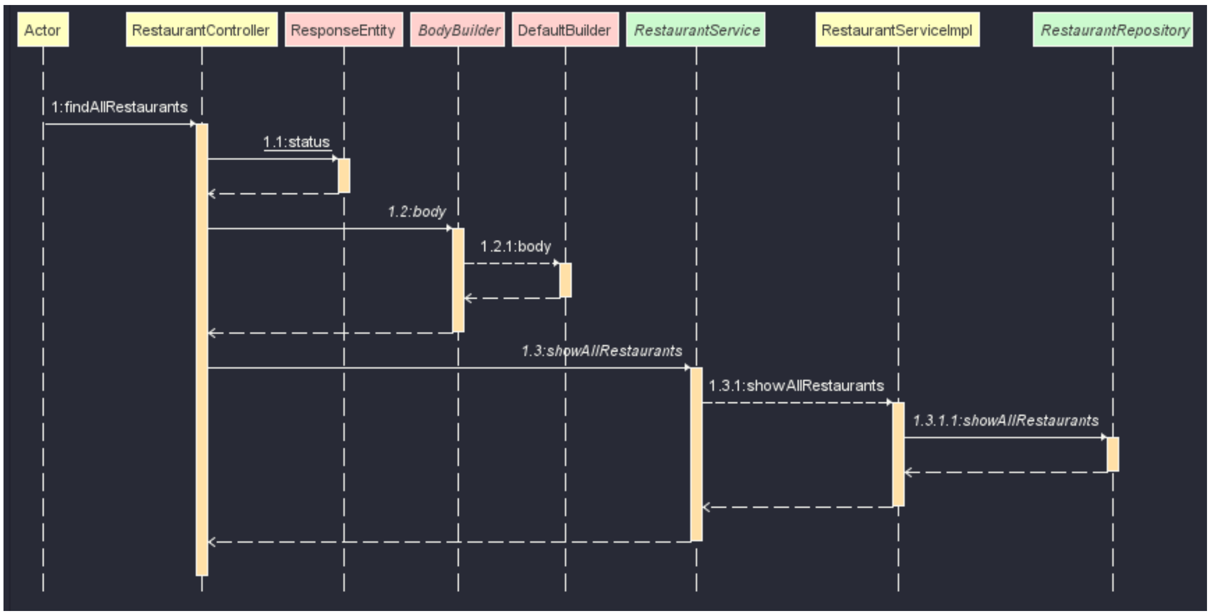


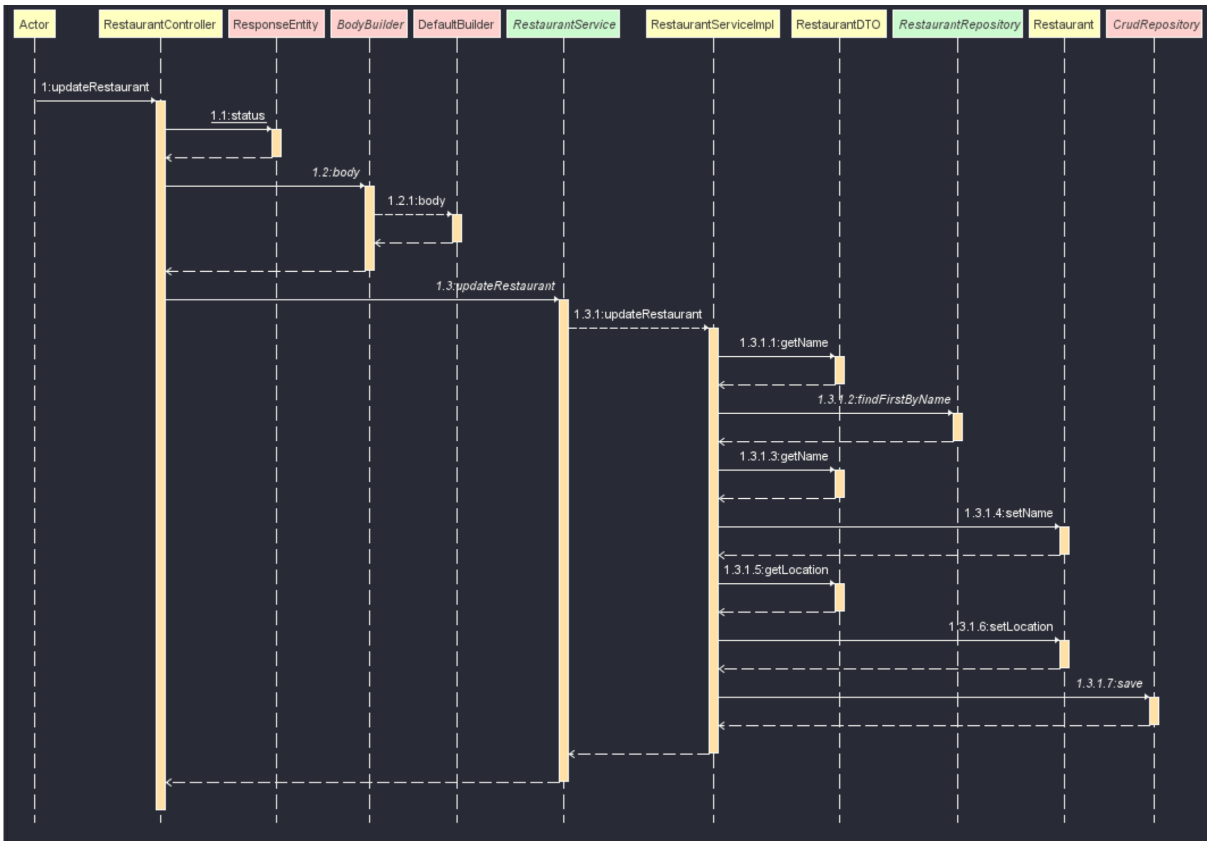
Deliverable 3

Design Model

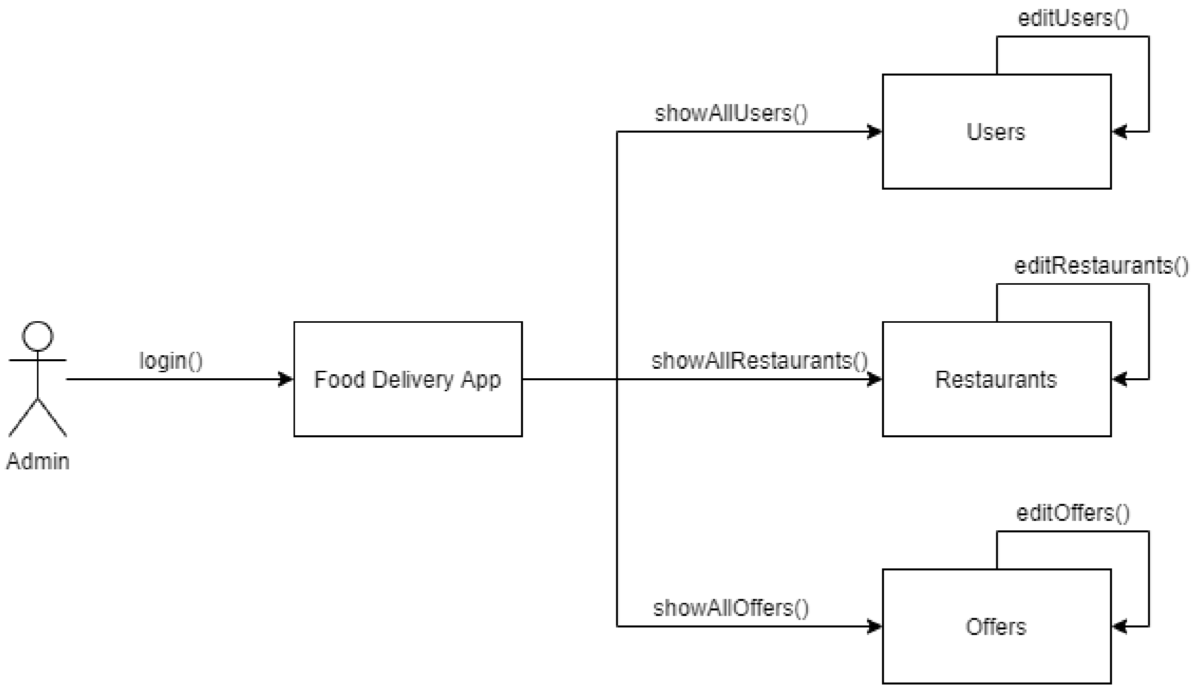
Dynamic Behavior

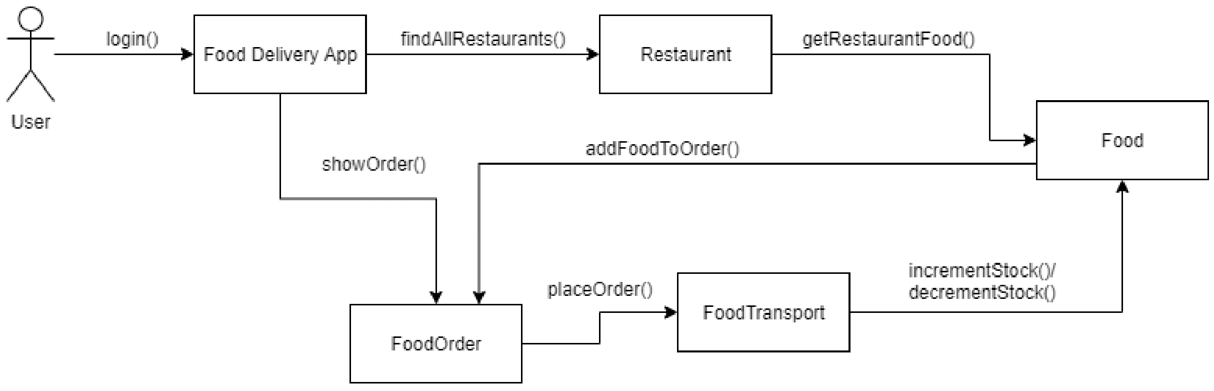
- Sequence Diagrams

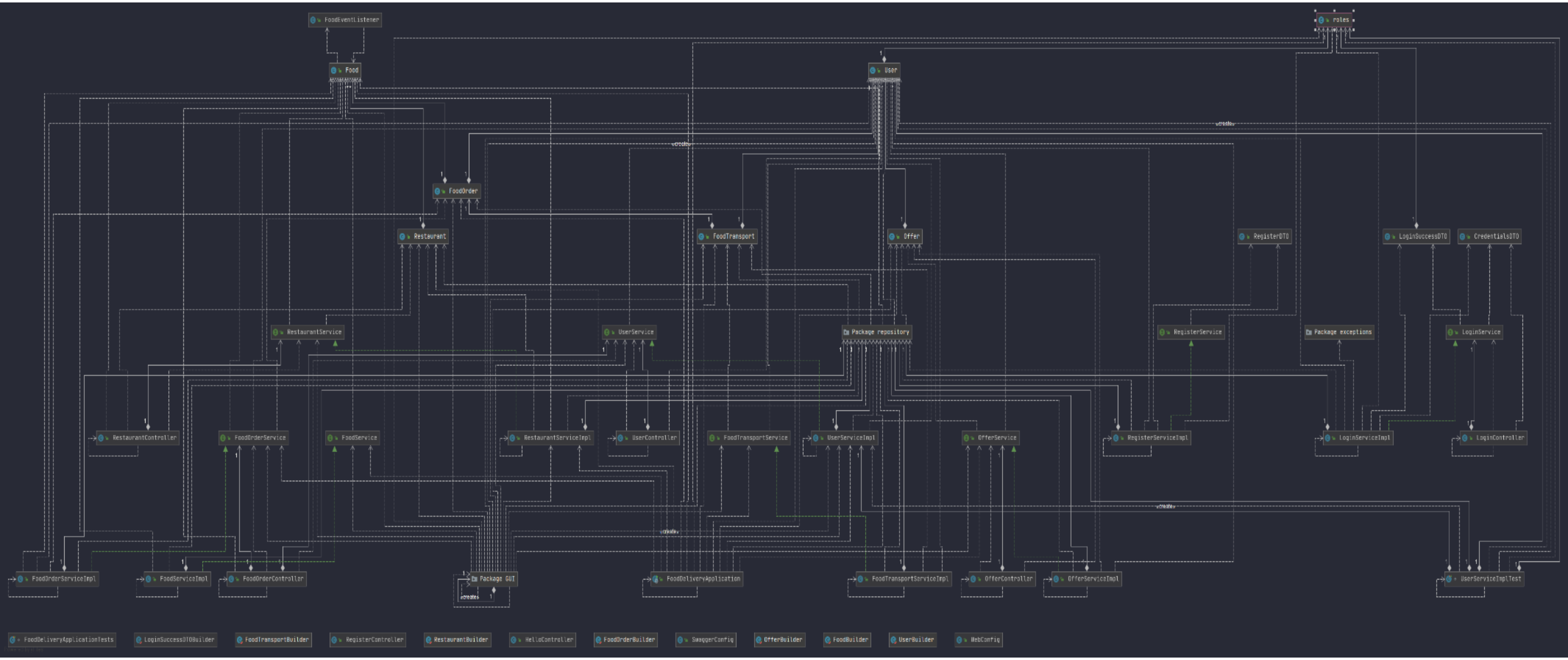




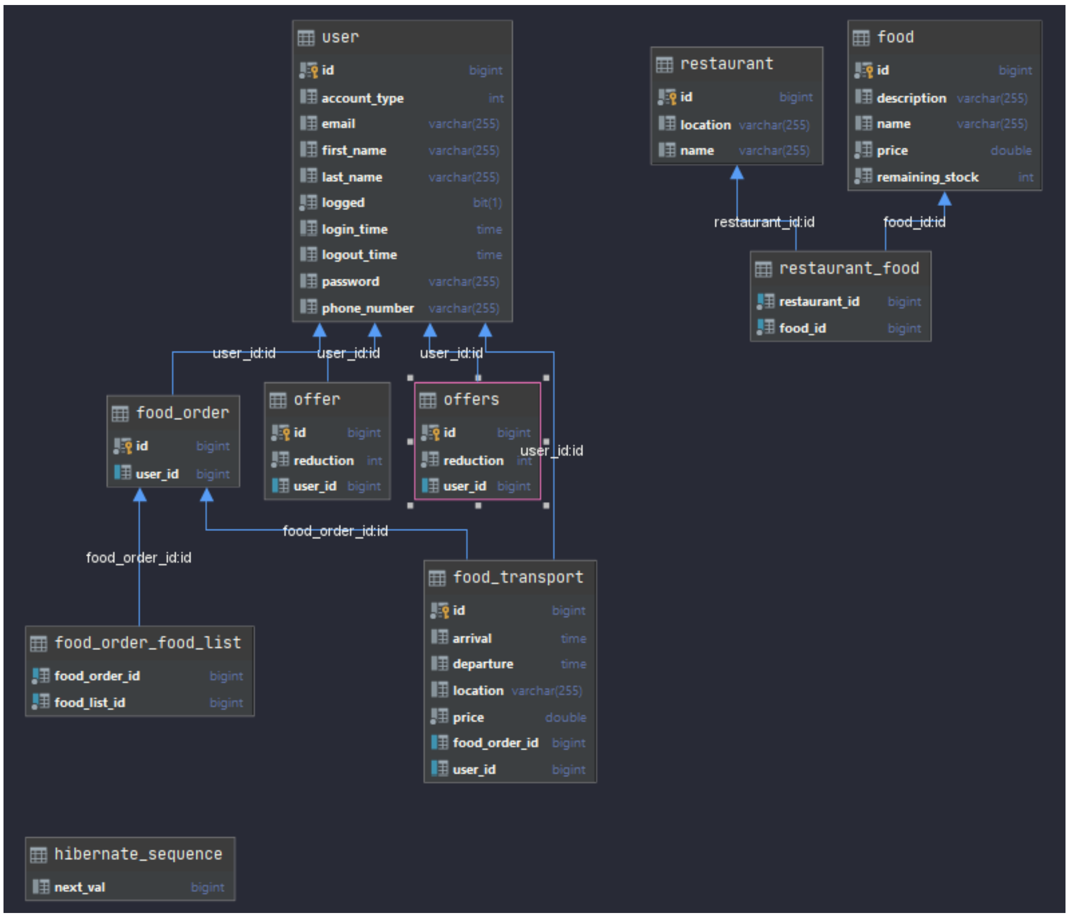
- Communication Diagrams






FoodEventListener
roles
Food
User
«create»
FoodOrder
Restaurant
FoodTransport
Offer
RegisterDTO
LoginSuccessDTO
CredentialsDTO
RestaurantService
UserService
Package repository
RegisterService
Package exceptions
LoginService
RestaurantController
FoodOrderService
FoodService
RestaurantServiceImpl
UserController
FoodTransportService
UserServiceImpl
OfferService
RegisterServiceImpl
LoginServiceImpl
LoginController
FoodOrderServiceImpl
FoodServiceImpl
FoodOrderController
Package GUI
FoodDeliveryApplication
FoodTransportServiceImpl
OfferController
OfferServiceImpl
UserServiceImplTest
FoodDeliveryApplicationTests
LoginSuccessDTOBuilder
FoodTransportBuilder
RegisterController
RestaurantBuilder
HelloController
FoodOrderBuilder
SwaggerConfig
OfferBuilder
FoodBuilder
UserBuilder
WebConfig

Data Model



System Testing

- Unit tests for CRUD operations of the model classes
- Postman request testing
- Functionality testing

Future Improvements

- Real time google maps tracking
- Credit card payment
- World-wide shipping
- Fidelity coupons

Conclusion

Spring is a powerful and versatile framework in the Java ecosystem that is also lightweight and allows building Java web applications with relative ease. It can be considered as a framework of frameworks because it provides various other useful frameworks. This project implements most of the basic principles of Spring and serves as a starting ground for larger and more complex projects. Also, this project uses React for the frontend, a JavaScript library for building user interfaces. Because React is component-based, composing complex UIs is a lot easier .