

Application Overview

You have to provide a running web application, which should do the following:

Hunger Net

Description:

Hunger Net is an online platform for food delivery.

- Registered clients have the possibility to order food from different restaurants and check their orders at any time. An order can be made only on active menus.
- Restaurant managers have the possibility to manage the menus and orders on their restaurant. A restaurant may have many menus but only one menu is active at a certain time (e.g. Breakfast Menu is active from 07:00 until 11:00, Lunch Menu is active from 11:00 until 17:00 etc).
- Admin users have the possibility to manage the users in the system as well as the restaurants.

Requirements:

As any user I want to have the login and logout functionality.

Hunger Net user roles:

- ADMIN
- RESTAURANT_MANAGER
- CLIENT

As an ADMIN user I want to be able to:

- Insert/update/delete/list users and restaurants into the system.
- RESTAURANT_MANAGER must be assigned to a restaurant when the user is being created.
- See full list of users (except other ADMIN users) and have the possibility to filter by role.
- See full list of restaurants.

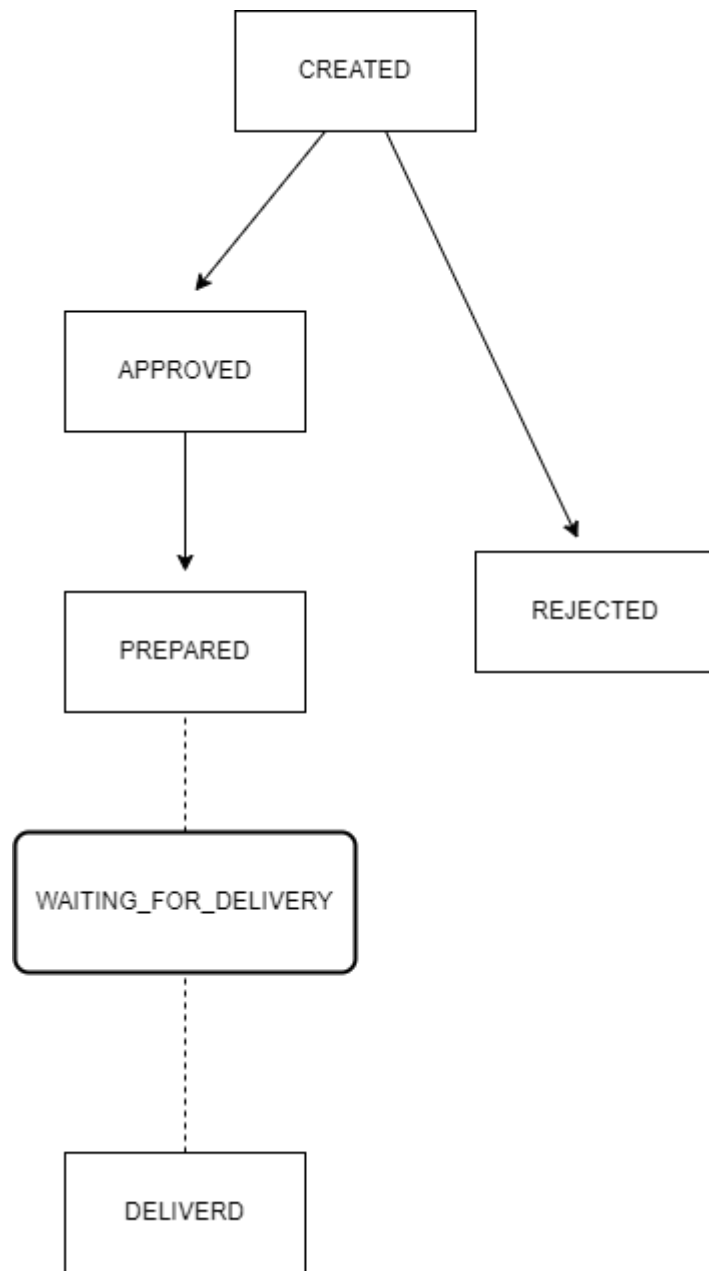
As a RESTAURANT_MANAGER user I want to be able to:

- Insert/update/delete/list menus of my restaurant.
- Insert/update/delete/list items in any menu.
- Approve/Reject orders and manage their status after approval.
- Check order details.
- List all orders and filter by status.

As a CLIENT user I want to be able to:

- See all restaurants (user can only see the active menu of the restaurant)
- Make an order
- Check order status
- See all of my orders (sort by date)

Order status workflow:



Technologies

The application should be well structured.

The following technologies/frameworks are mandatory:

- Spring for the service layer (Spring MVC and Spring Boot)
- Hibernate for the persistence layer
- MySQL (or any other RDBMS)
- Spring Security, implement JWT
- Maven for the dependencies between the modules besides the other proprietary libraries.
- JUnit
- Logging possibility via log4j.

Optional: Module for the Frontend having also the client for the Rest Web Services using Angular or React JS.

Bonus: Swagger

Note: Prepare the Postman requests before the presentation for the endpoints if UI is not implemented.