

Hungry Bellies

Teammates: Venkat Arigela, Pradyoth Srinivasan, Jay Patel

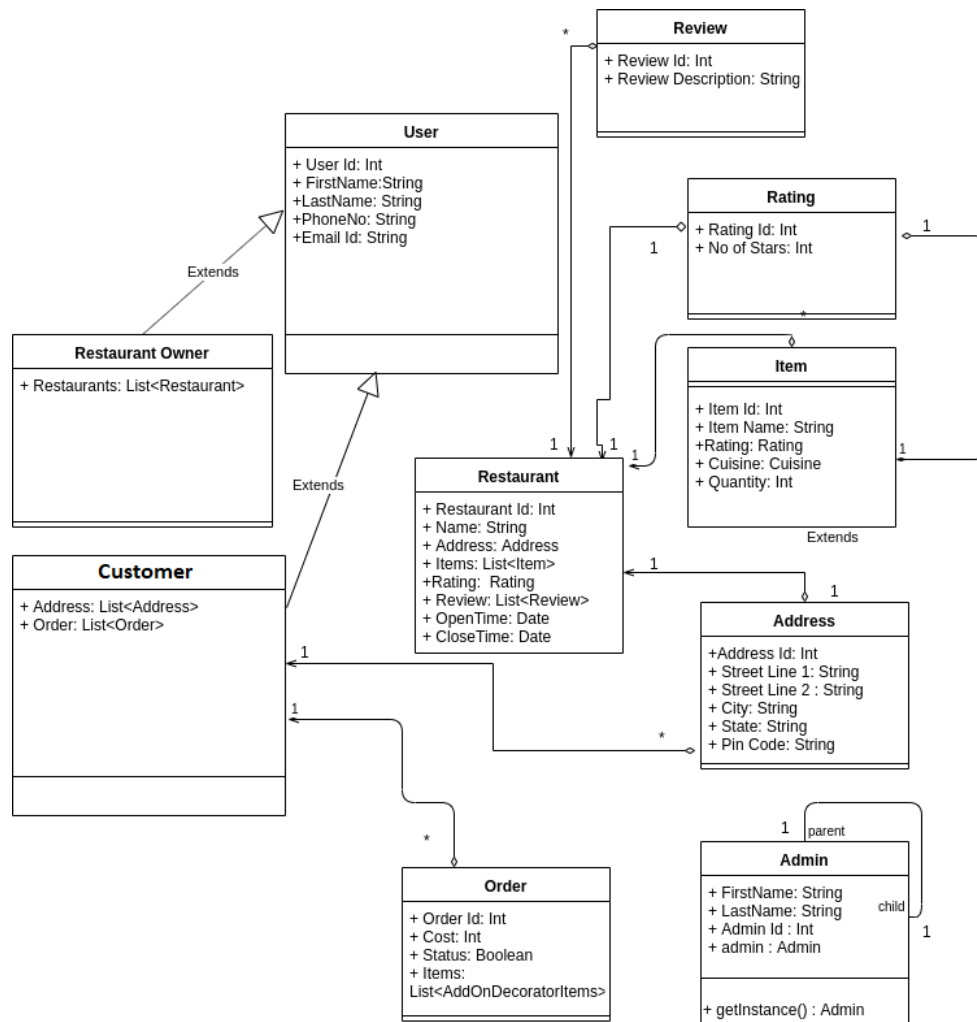
Final state:

We have made a single web application page for Food Delivery utility. We have implemented most of the features we planned to do in project 4. We have implemented the following use cases:

- 1) Sign up for user
- 2) Login and authentication
- 3) Search bar for search through Restaurant names
- 4) Place order (add to the cart and pay)
- 5) Retrieve the past Order history of a customer
- 6) Writing reviews for a particular restaurant and getting all reviews for the particular restaurant.

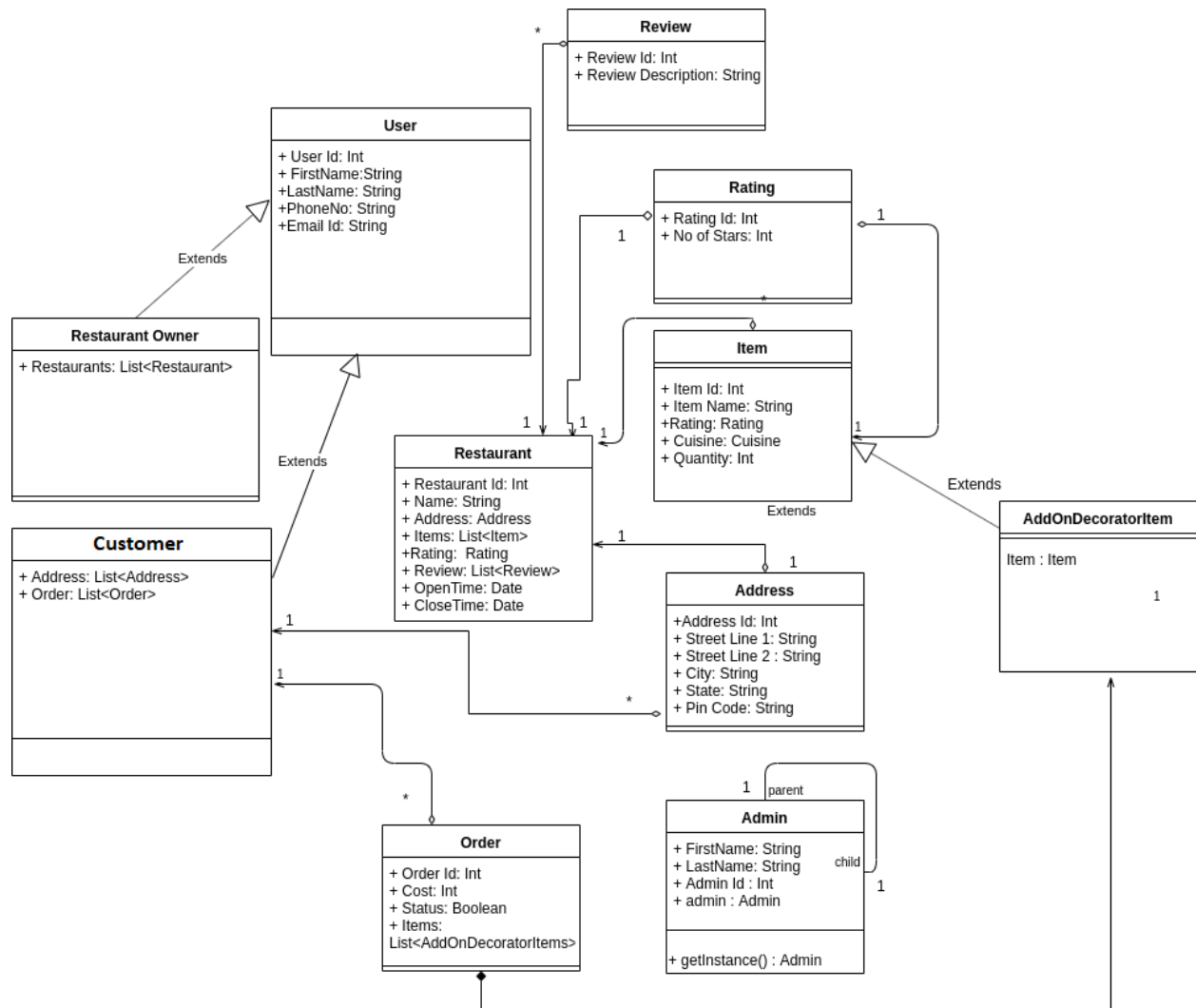
The features which we didn't implement are google map integration and docker image. These features were not implemented because of time constraint and lack of proper planning.

UML diagram of final system:



The patterns which we implemented are MVC, iterator, Facade and singleton.

Previous Class Diagram (Project 4):



Difference:

As from above 2 diagrams, we removed the implementation of decorator pattern for item and order from our class diagram. Apart from that, we have implemented all the classes which we gave in the class diagram for project 4. The features like Google live tracking or google map and docker image were not able to integrate in this system.

The third party's elements used:

We have used following tools and framework:

- 1) spring boot framework for the Backend
- 2) ReactJs for UI
- 3) Stripe.js framework for payment integration
- 4) Tomcat as web-server.
- 5) Swagger configuration.

OOAD process for entire project:

The major issues we faced while analyzing and designing the project are as follows:

- 1) The extensive prediction of the entities at the beginning of the project, and prediction of which external dependencies have to be used was tough.
- 2) After the design being made, trying to make it more modular and better in terms of the dependencies and maintainability was brainstorming.
- 3) Deciding the responsibilities between classes was also a issue.
- 4) The relationship establishment using JPA (Java Persistent API)