FLIGHT RIDER

Final Report:

Project Name: Flight Rider

Team Members: Sankaranarayanan, Mohan Dwarampudi.

Final State of the system:

We have implemented a flight booking website name FlightRider.com.

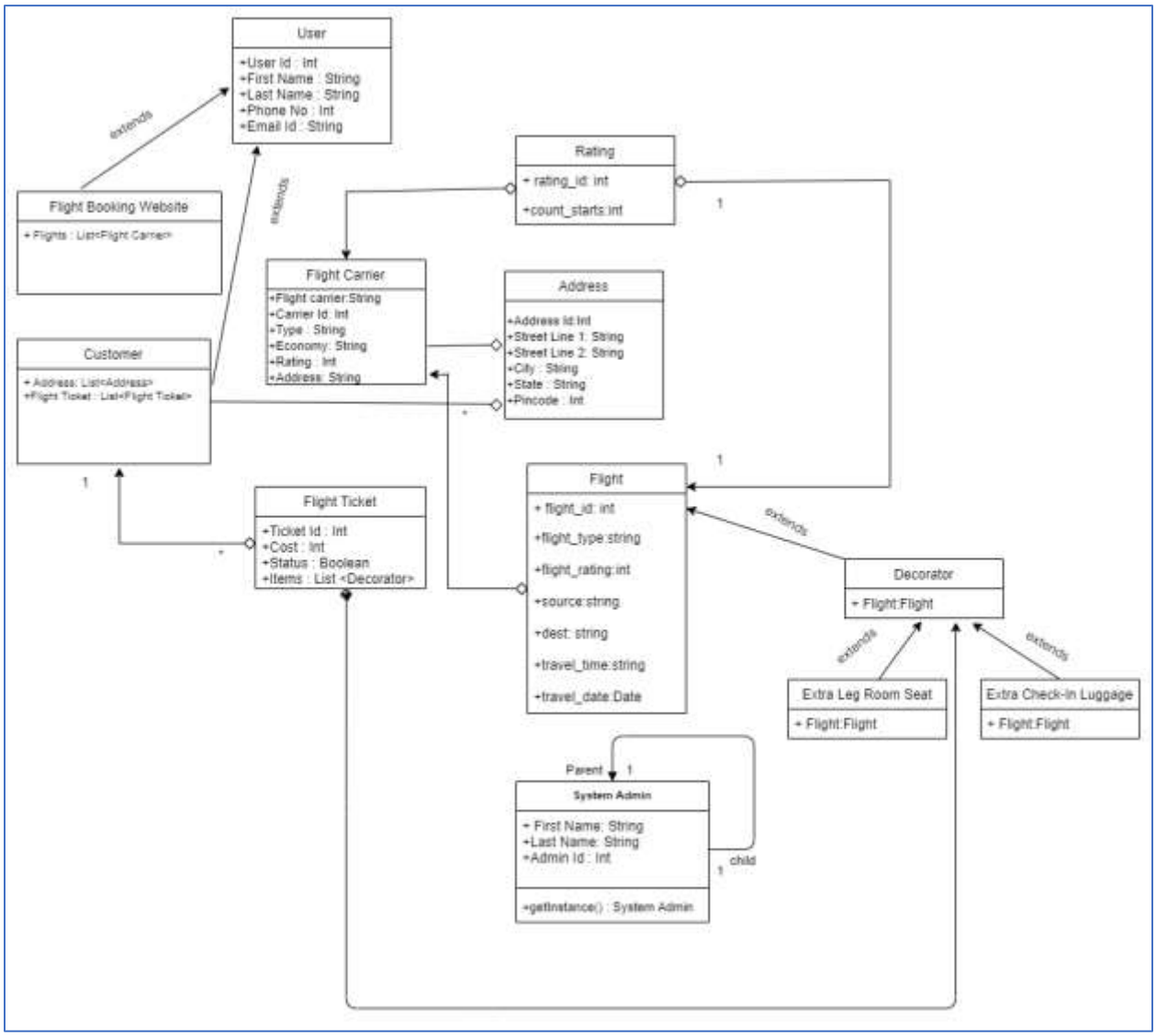
We have implemented the basic functionalities in the website.

1) We have implemented Register functionality for a new user to register his credentials.
2) Trip Booking functionality.
3) Flight check-in functionality.
4) Manage the flight booking functionality.
5) We have also implemented Ticket mailing functionality to a given customer using Node.js mailer.

Functionalities which we will be implementing in the future.

1) Display the all the flight schedules in the website.
2) Deploying the website using Docker or Heroku.

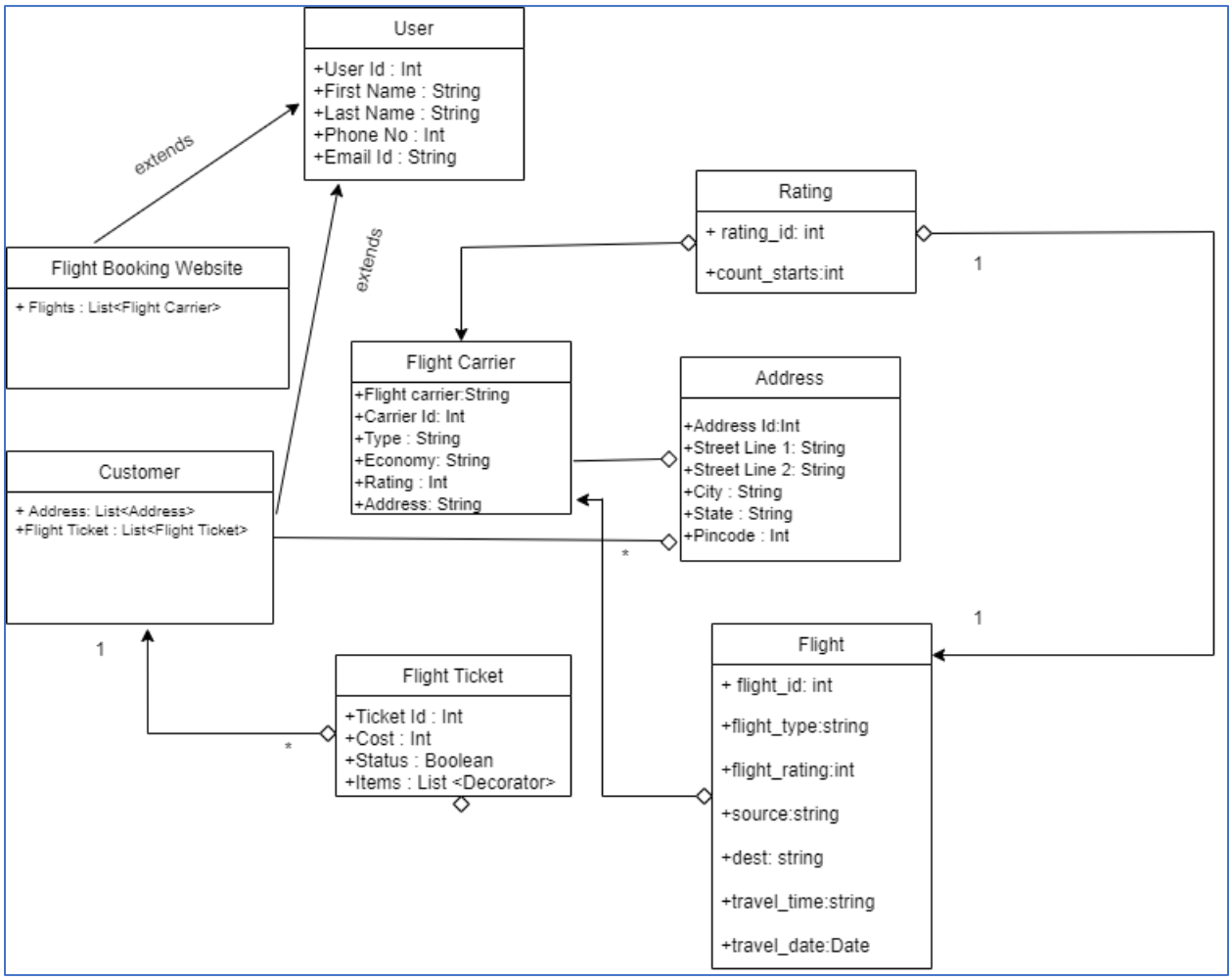
UML Diagram of final system:



The patterns that we have implemented are

a) Decorator pattern.
b) Observer Pattern.
c) Singleton Pattern.
d) MVC pattern.

Previous Class Diagram(Project 4):



Difference:

When compared to both the above two diagrams we have implemented decorator pattern by adding the type of cuisine required for a customer when booking a flight ticket. we also have implemented the singleton pattern for the registration page. we tried to deploy our website using Docker which we were not able to integrate with the system.

The Third Party's elements used:

We have used the following tech stack for our project:

a) React.js for front end.
b) Node.js for Rest API calls and backend.
c) Mongo DB for database.
d) Material UI for React Front End components.
e) ROBO 3T for MongoDB record visualization.
f) Mongoose ODM.

OOAD Process for the entire Project:

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

1) We had to restart the NodeJS server every time for the changes to be exhibited.
2) When using Mongoose ODM we had to make the schema in a singular form to the database being used in MongoDB, else it wont be able to load the data into the MongoDB. This small issue took 3-4 hours of debugging and online browsing.
3) As our team was new to using React.js and Node.js , we struggled for the project initiation and coming up with the deliverables to be delivered.
4) After the deliverables were fixed we struggled to make the code modular and better in terms of dependencies.

Overall, this project made us to learn a lot and enhance our knowledge. we are happy for choosing this course. Thank You.