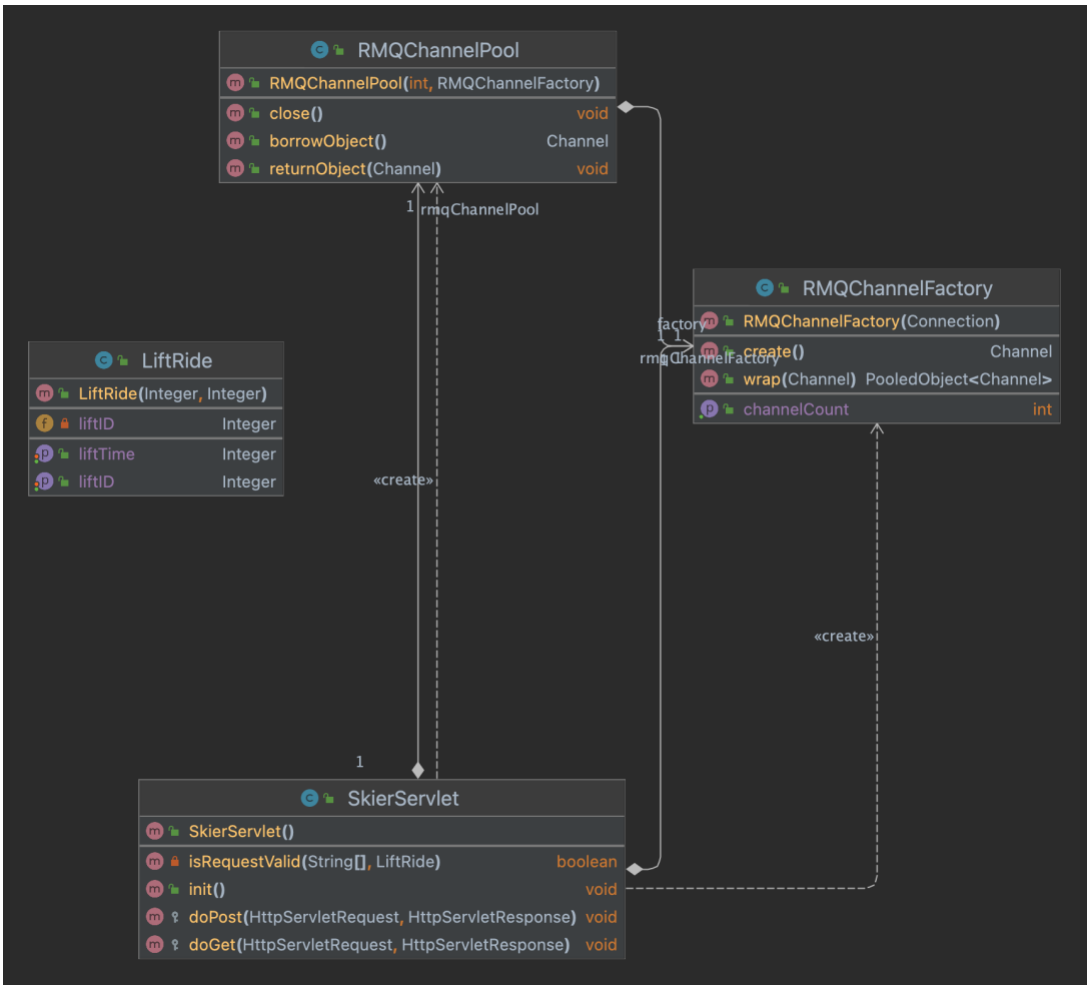
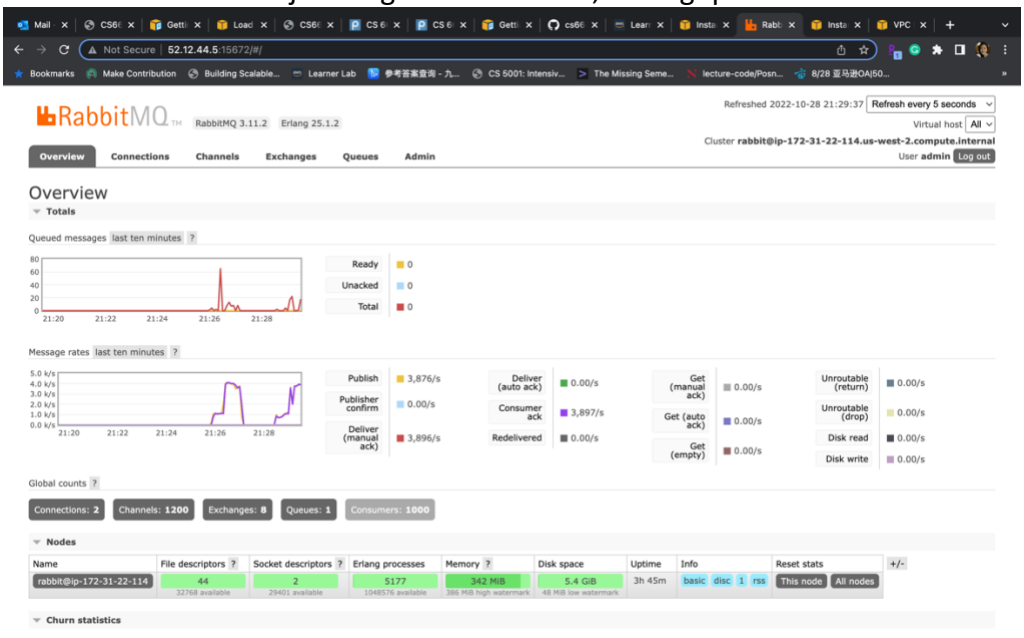
1. URL for repo:
   https://github.com/edellhou/CS6650/tree/main/Assignment2

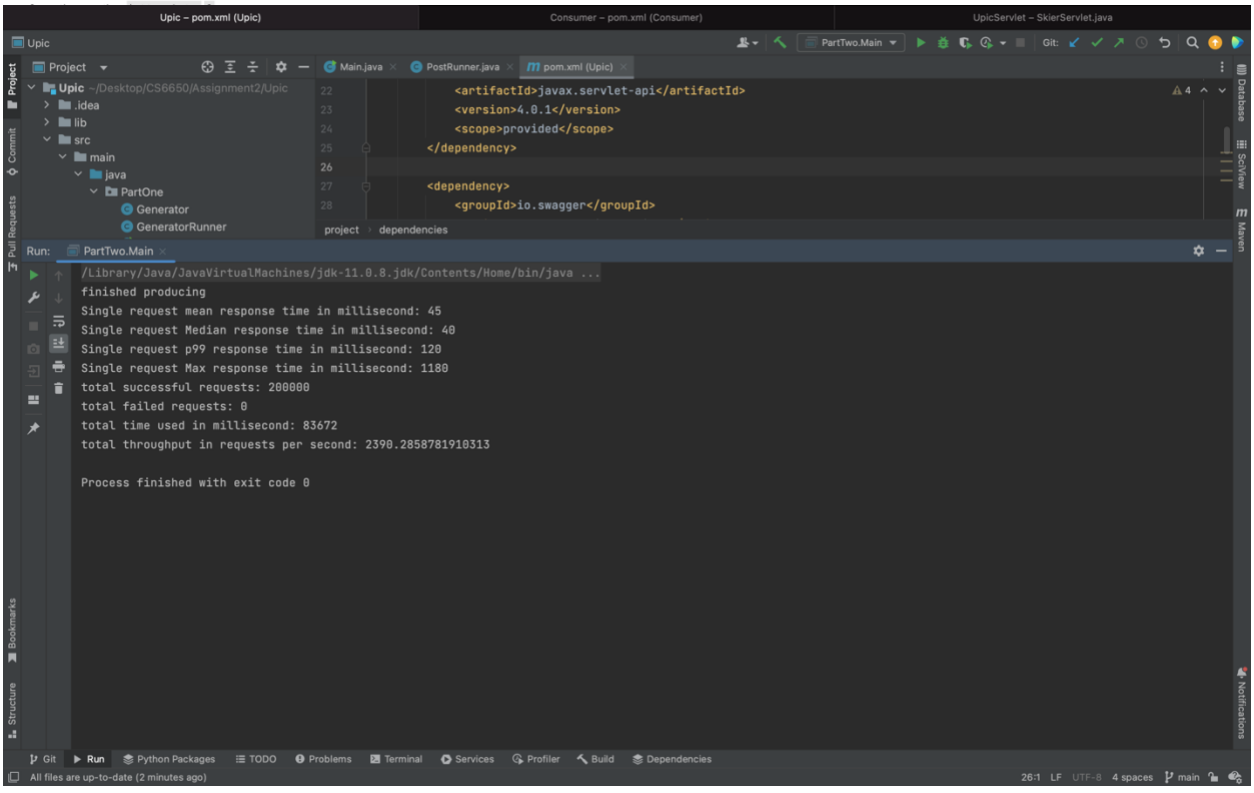
2.Server Design:
Here is the UML showing all the classes on the server side(publisher) and their relationship. The packages used are Gson and RBMQ.
LifeRide is a utility(helper) class that take the request body of liftRide and store it as LiftRide class.
The server sent messages to a message queue use RBMQ library, the exchange method used is fanout.
In the main class SkierServlet, connection is created in the init() method, to only create time consuming connection once. The channel is created in a channel pool, and channels can be recycled to send messages to queue. The channel and channel pool utility classes are RMQChannelFactor and RMQChannelPool.



Consumer side is pretty simple, it only has one class called Recv, and it poll the message from the queue, and stored info in a Hashmap(key is the skier id, and value is a string that contain all the lift ride information). The exchange method used is fanout, aligned with publisher.

3.Here is the stats for just single instance test, throughput is 2390.



4. Here is the stats for load balanced test, throughput increased to 2745