

Taxi Booking Framework

A taxi dispatching station having a pool of taxis serving customers.

The server provide RESTful web services for the pull-based interaction between the client and server for the stateless components .

Course : Object Oriented Design

Professor : Dr. Ashtiani

Presenters :

- Amirhosein Ghasemi
- Mobina Kashanian
- Mohamad Sajad Naghizade
- Mahnaz Haghani Nejad

Contents

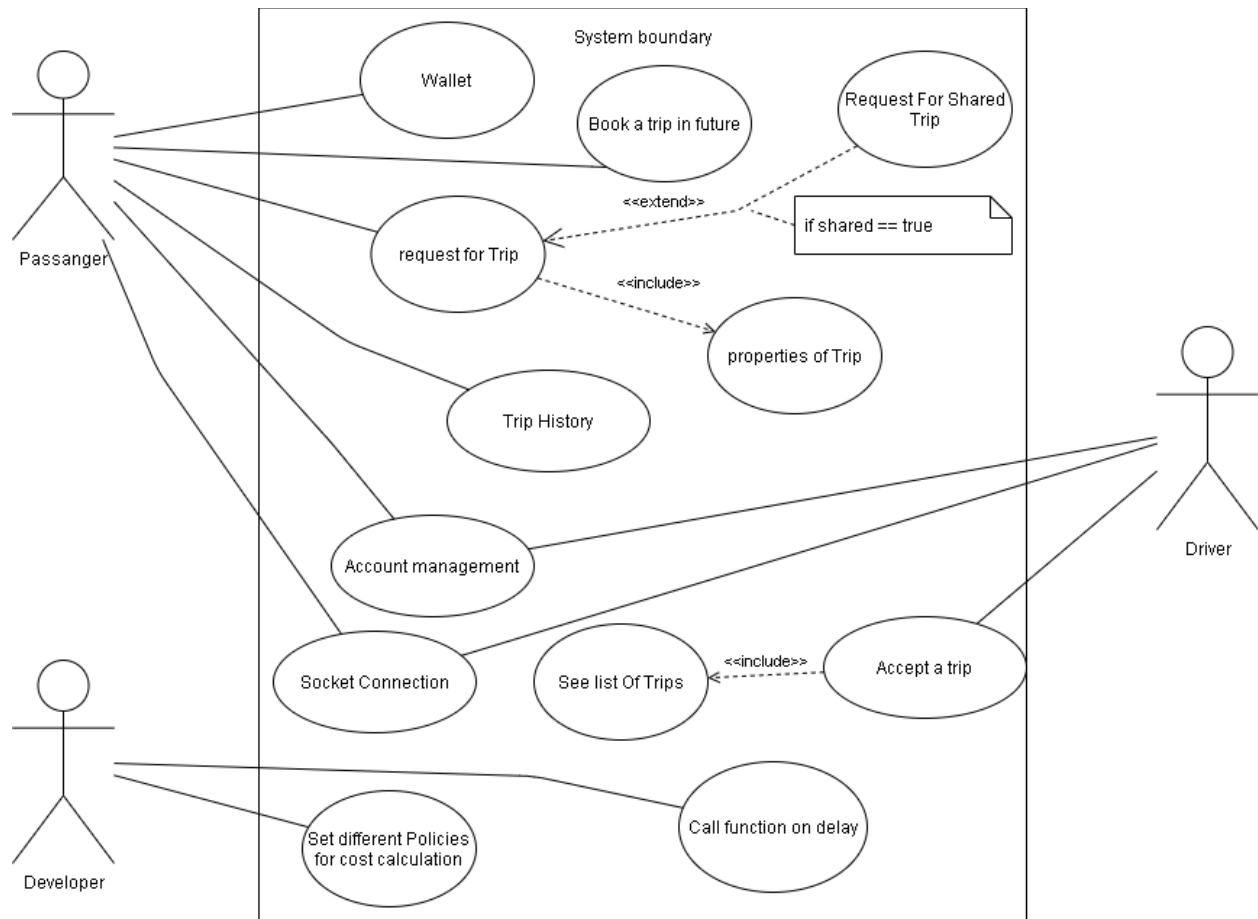
Taxi Booking Framework	1
Features	3
Use Case Diagram.....	4
Sequence Diagram.....	5
Implementation:	8
1. Class Diagram.....	9
1. API.....	10
2. Assumptions.....	10



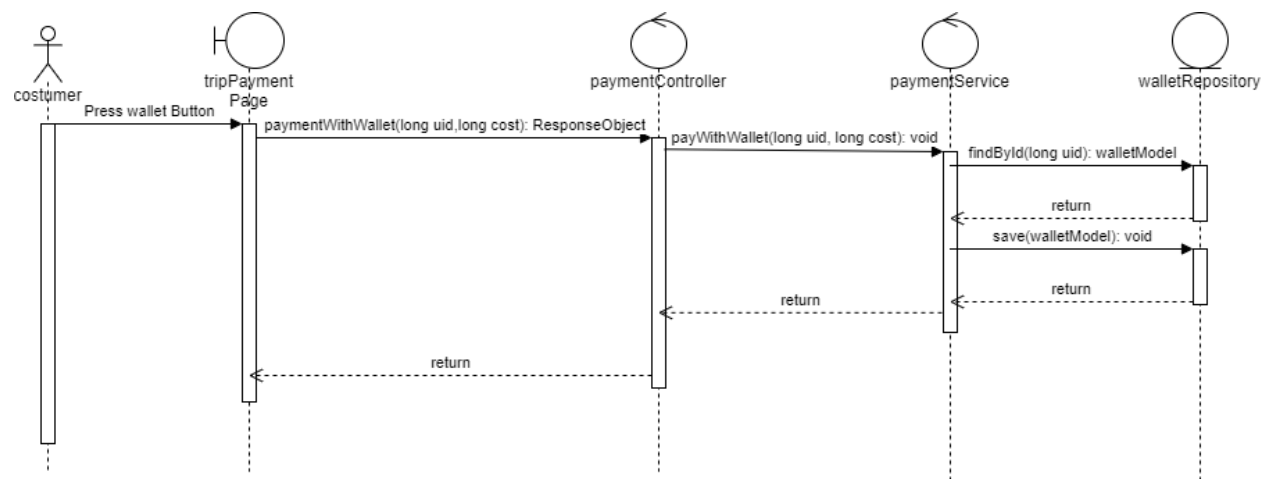
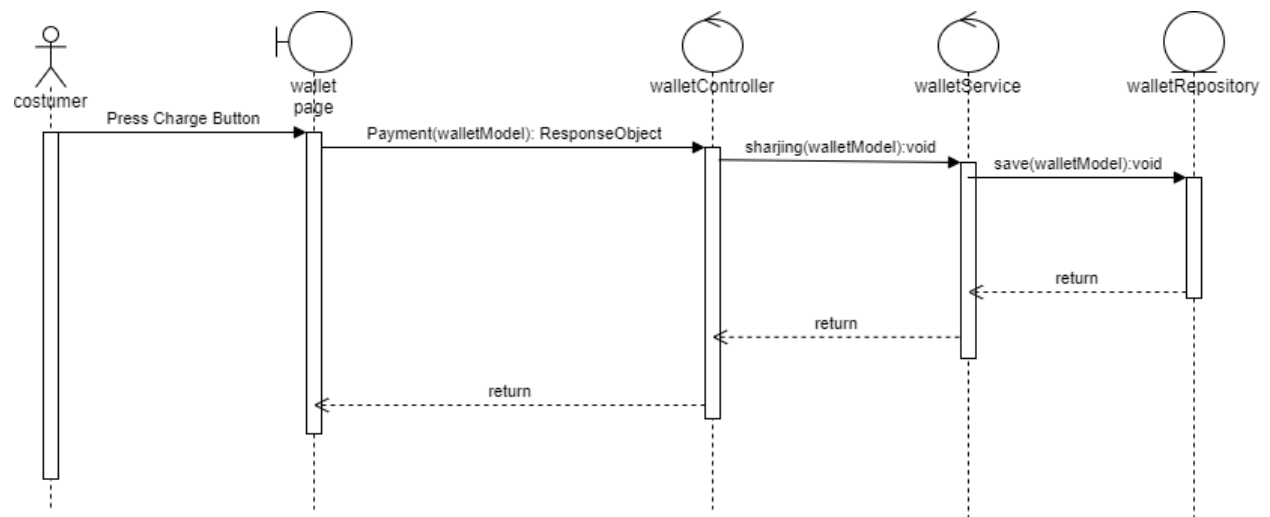
Features

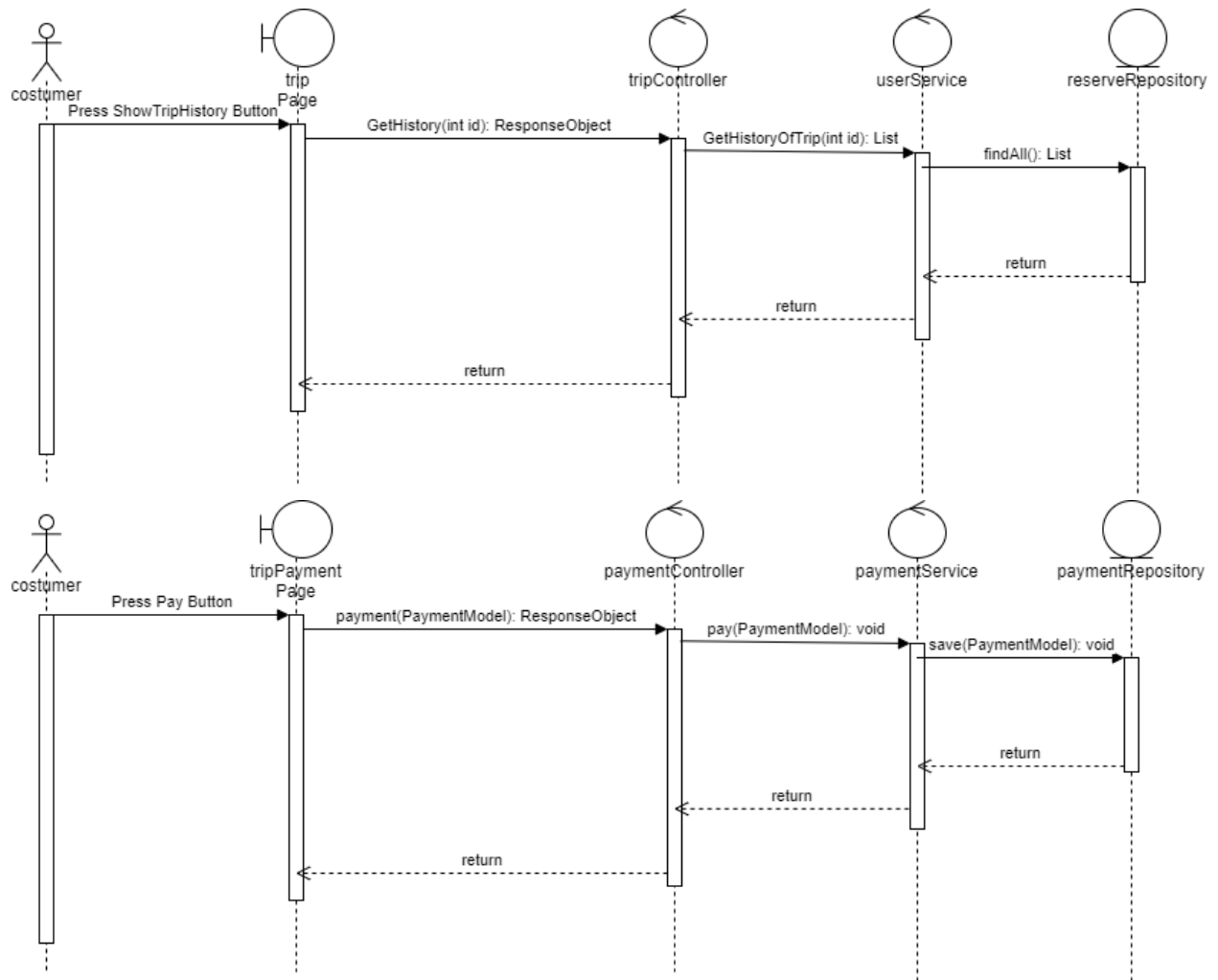
- Framework supports Trip Reservation.
- Framework supports Passengers and Drivers Management Panel.
(Includes account creation and other related APIs.)
- Framework supports Scheduled Reservation.
(book a trip in some date in future)
- Framework supports Different Function Call On Arriving Delaying By Passengers Or Drivers.
(can call functions on specified event)
- Framework supports Different Conditions for the Trip.
(a map structure representing Conditions is added to reserve API so that it would check for Conditions satisfactions before reserving)
- Framework supports Trip History.
- Framework supports Wallet for passengers and different payment methods.
- Framework can use different policies (specified by framework user) to calculate cost of trip.

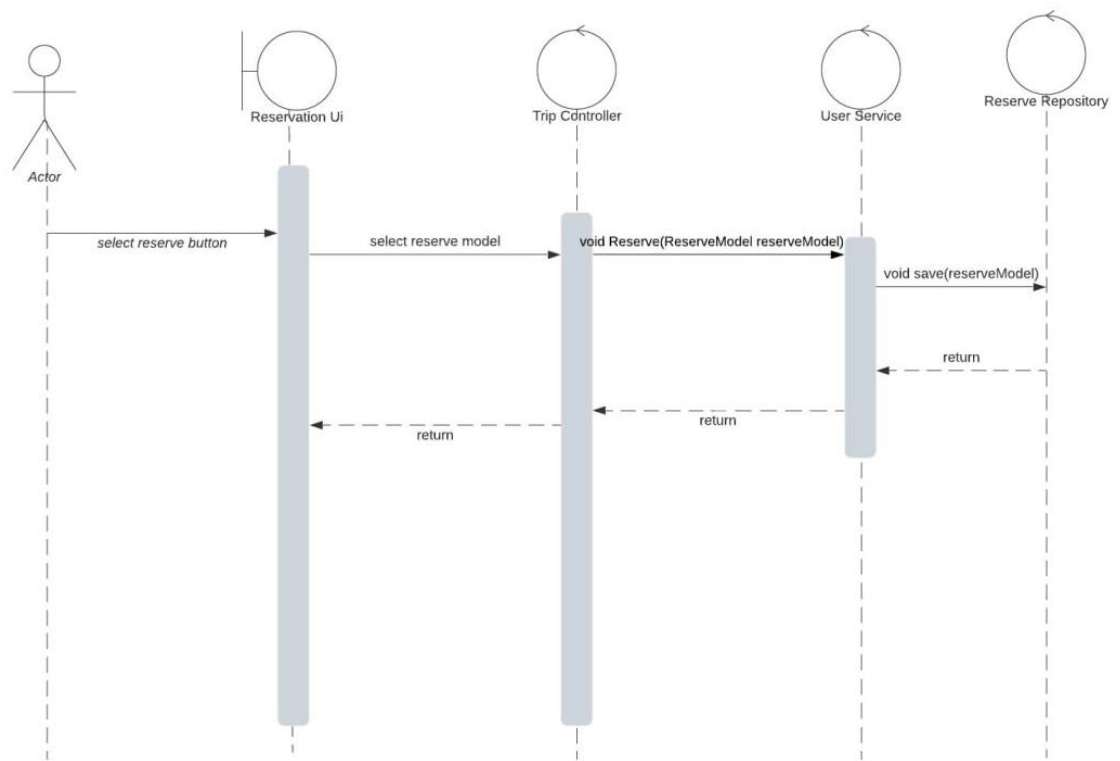
Use Case Diagram



Sequence Diagram







Implementation:

we used java spring boot restful service to implement these features.

Controller : which handles request from client.

Service : implements functionality.

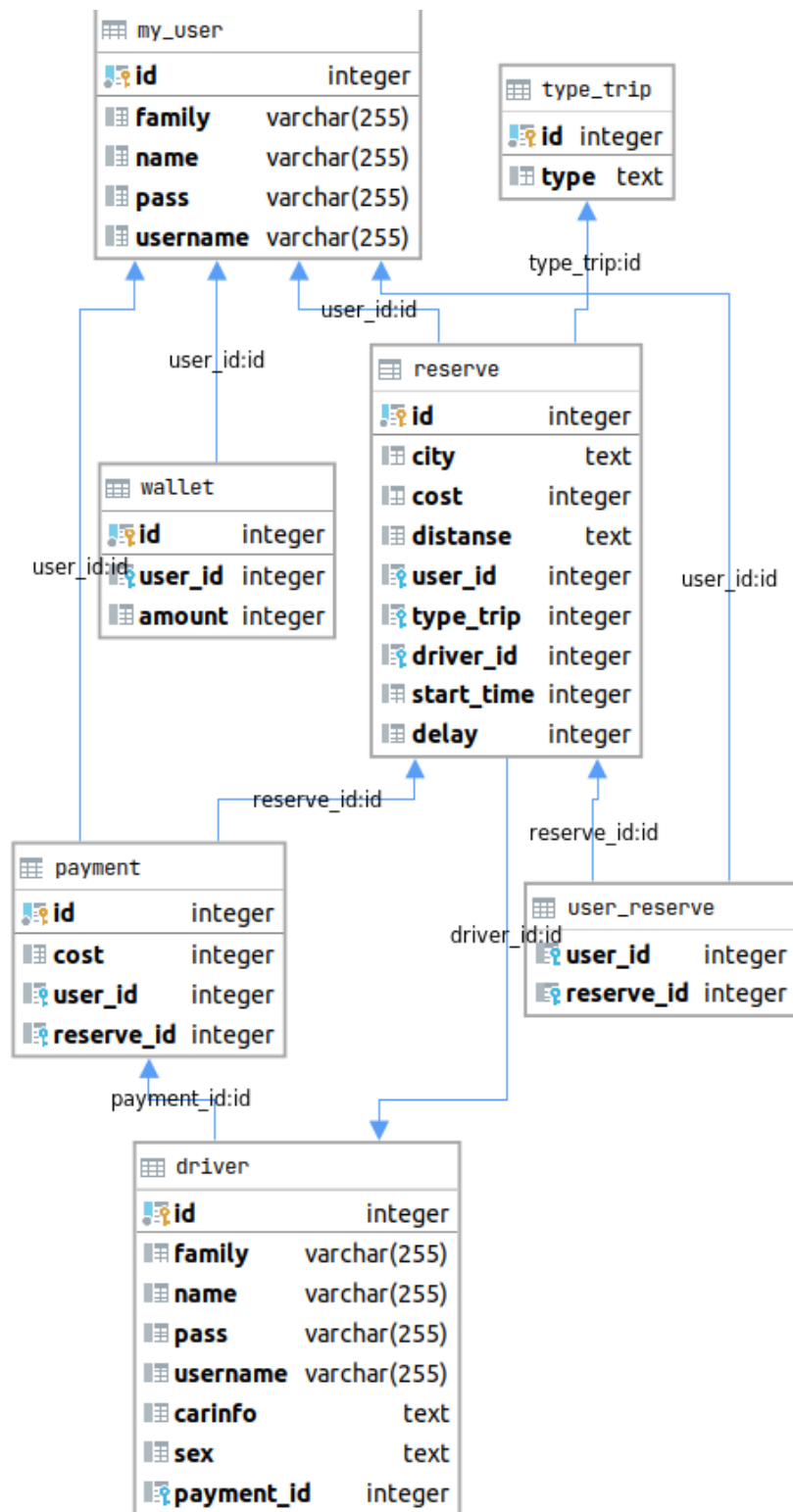
Repository : data access object between database and other components.

Model: Implements Entity.

DTO : fields of client endpoint request.

Config: configs that used in project like swagger.

1. Class Diagram



1. API

basic-error-controller	Basic Error Controller	>
create-driver-account-controller	Create Driver Account Controller	▼
POST	/driver/createAccount createAccount	🔒
create-user-account-controller	Create User Account Controller	▼
POST	/user/createAccount createAccount	🔒
payment-controller	Payment Controller	▼
POST	/payment payment	🔒
POST	/payment/wallet paymentWithWallet	🔒
trip-controller	Trip Controller	▼
GET	/user/GetHistory/{id} GetHistory	🔒
POST	/user/reserve Reserve	🔒
wallet-controller	Wallet Controller	▼
POST	/wallet/sharj payment	🔒

2. Assumptions

- For now it's not possible for Driver to see trip list and choose a trip as desired, (trip is automatically assign)
- We assumed that when a user use the wallet for payment , wallet 'money is bigger than trip cost.