



Accademic Year 2016/2017

# POWER ENJOY

---

DD Presentation  
Politecnico di Milano  
Software Engineering 2

Simone Bruzzechesse

Luca Franceschetti

Gian Giacomo Gatti

# INDEX

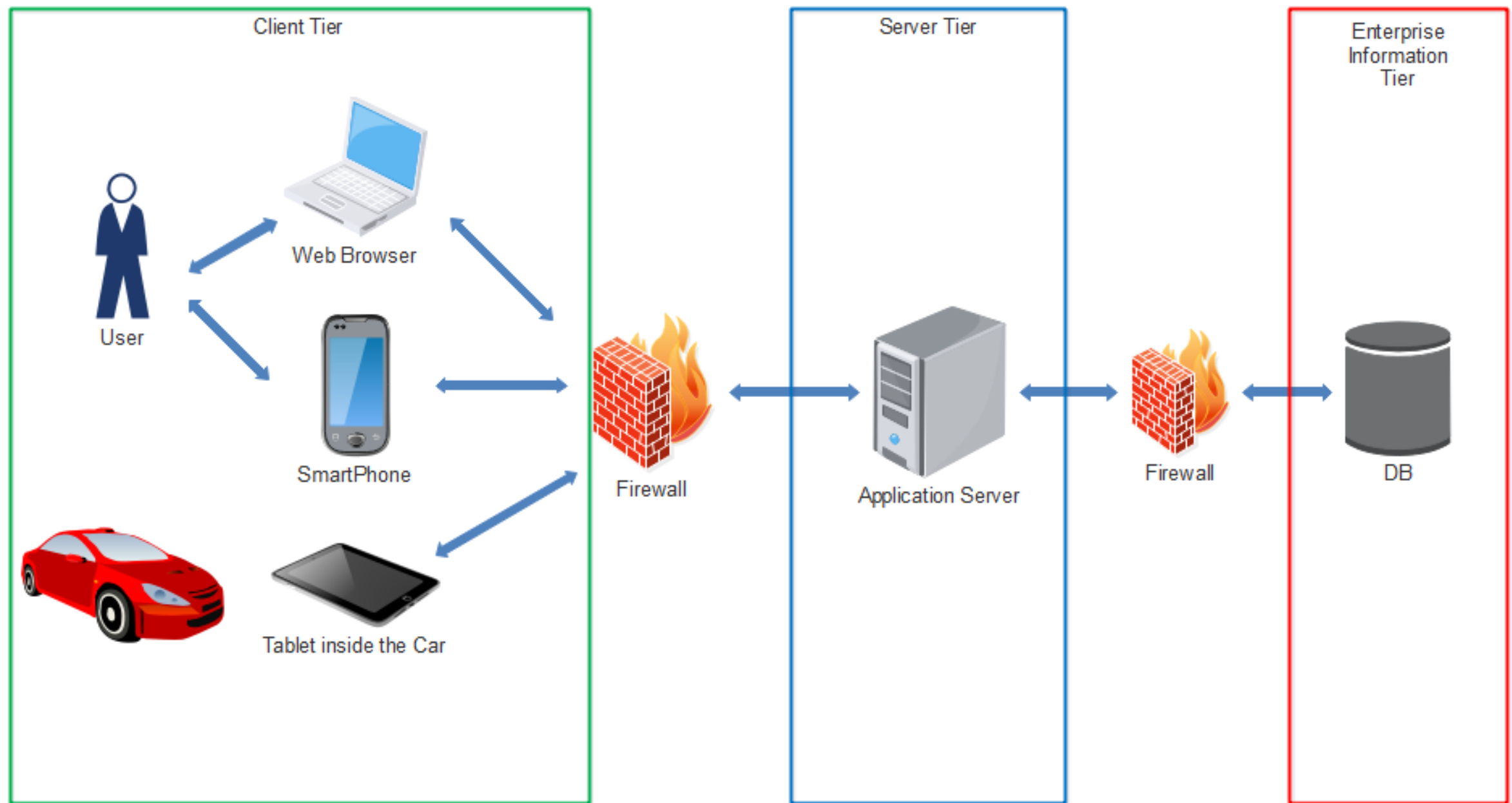
---

The purpose of Design Document is to identify:

- High level infrastructure
- Main components and their interfaces:
  - Component Diagram
  - Data Structure
  - Deployment View
- Runtime View
  - Sequence Diagram
  - Algorithm Design
- User Interface Design

# HIGH LEVEL INFRASTRUCTURE

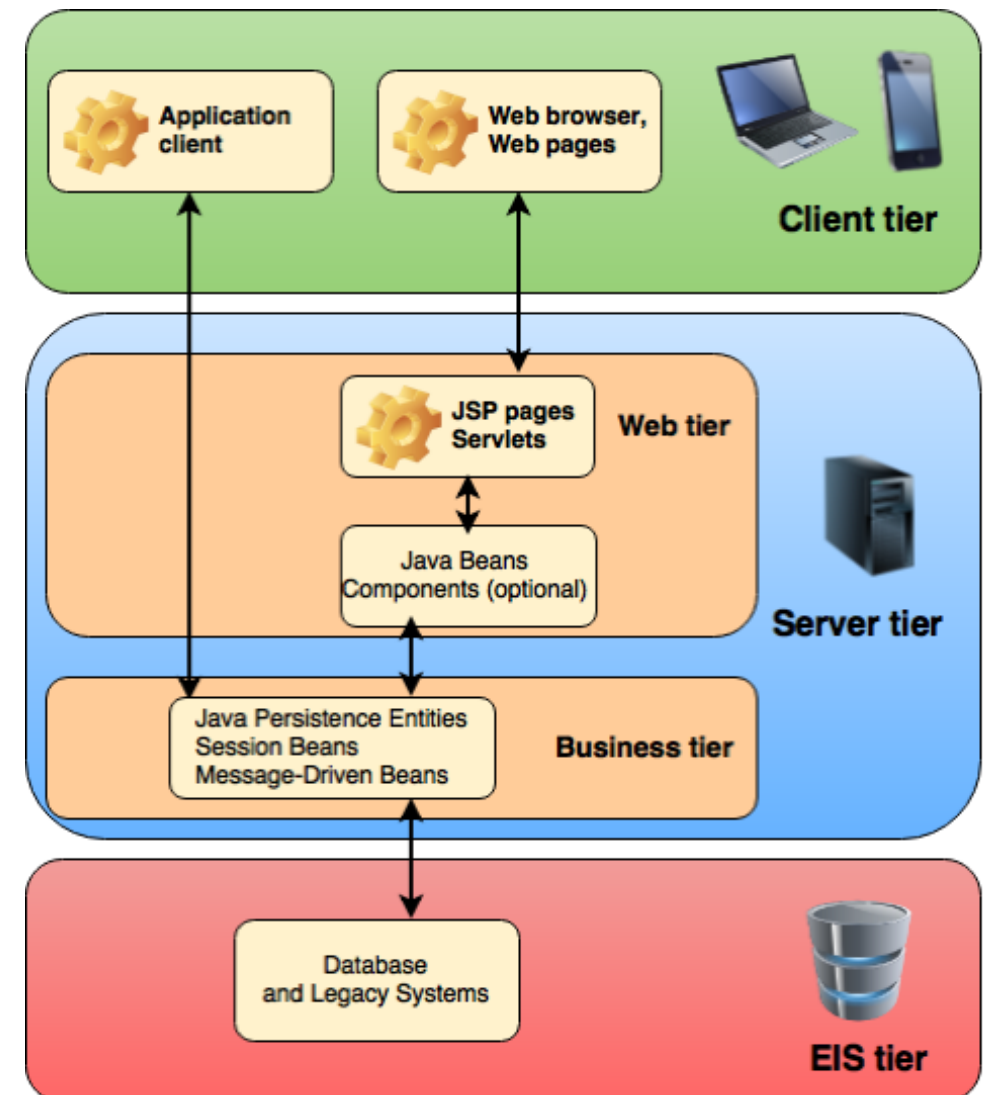
---



# J2EE INFRASTRUCTURE

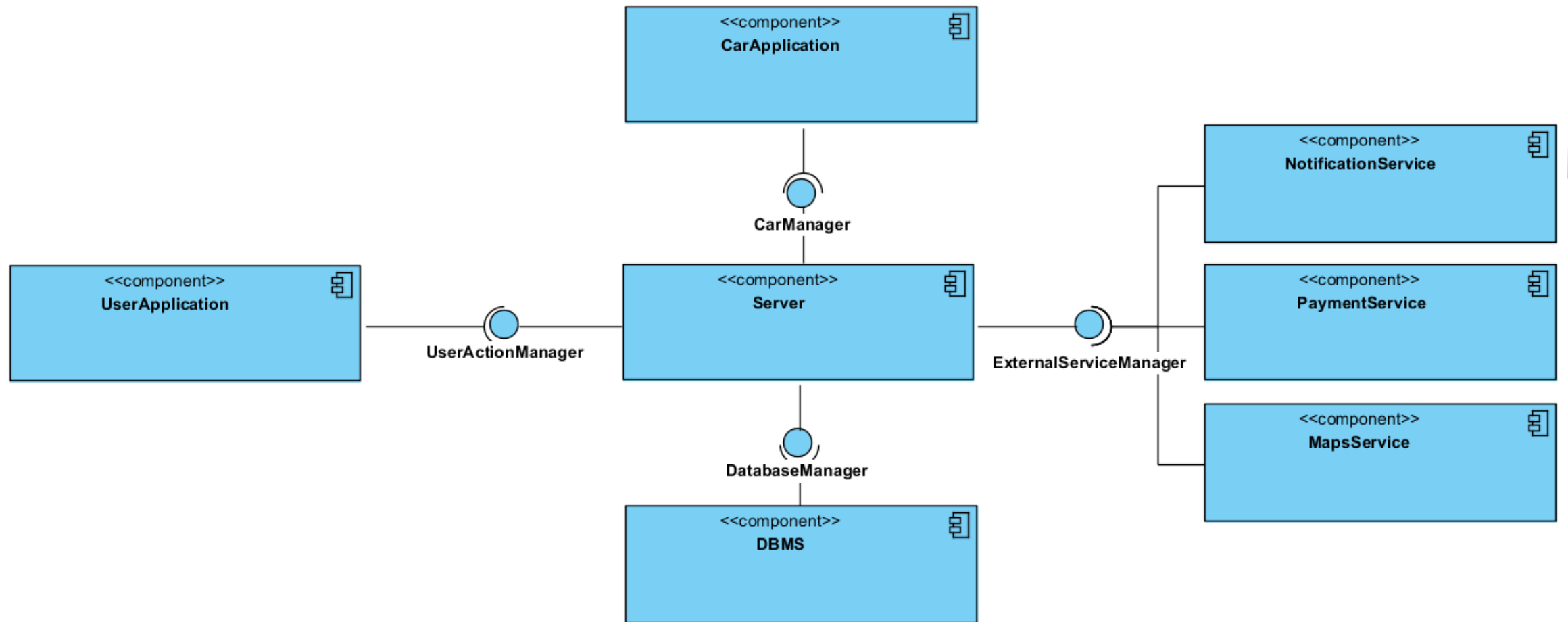
---

- **Client-tier:** run on the client machine, both on the web browser and mobile application and also on the machine's tablet
- **Server-tier:**
  - **Web-tier:** Servlets and JSP that are used to manage the interaction with the web application
  - **Business-tier:** these components manage the internal logic of the system
- **Enterprise Information System (EIS)-tie:** handle enterprise infrastructure systems, database systems and legacy systems

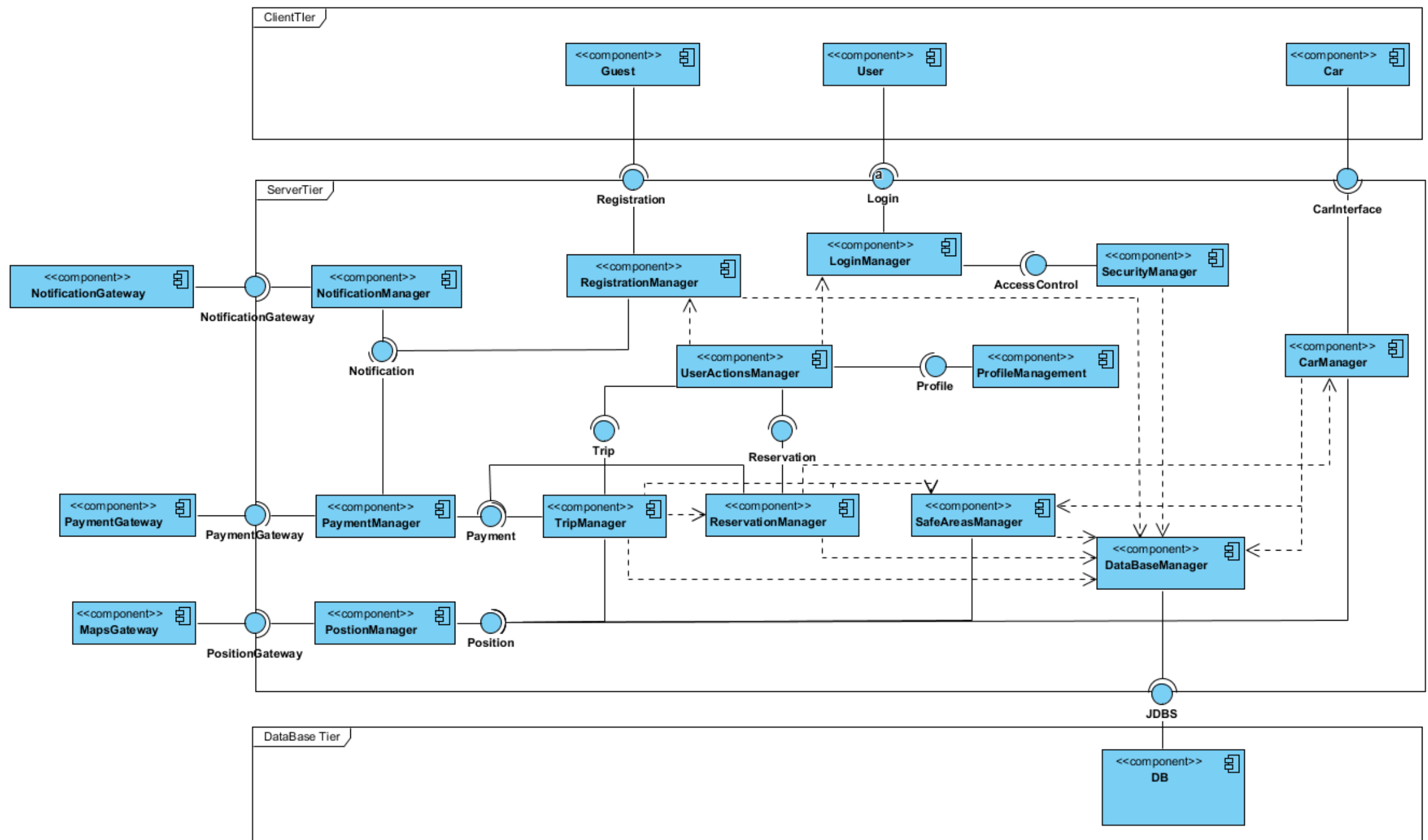


# MAIN COMPONENT DIAGRAM

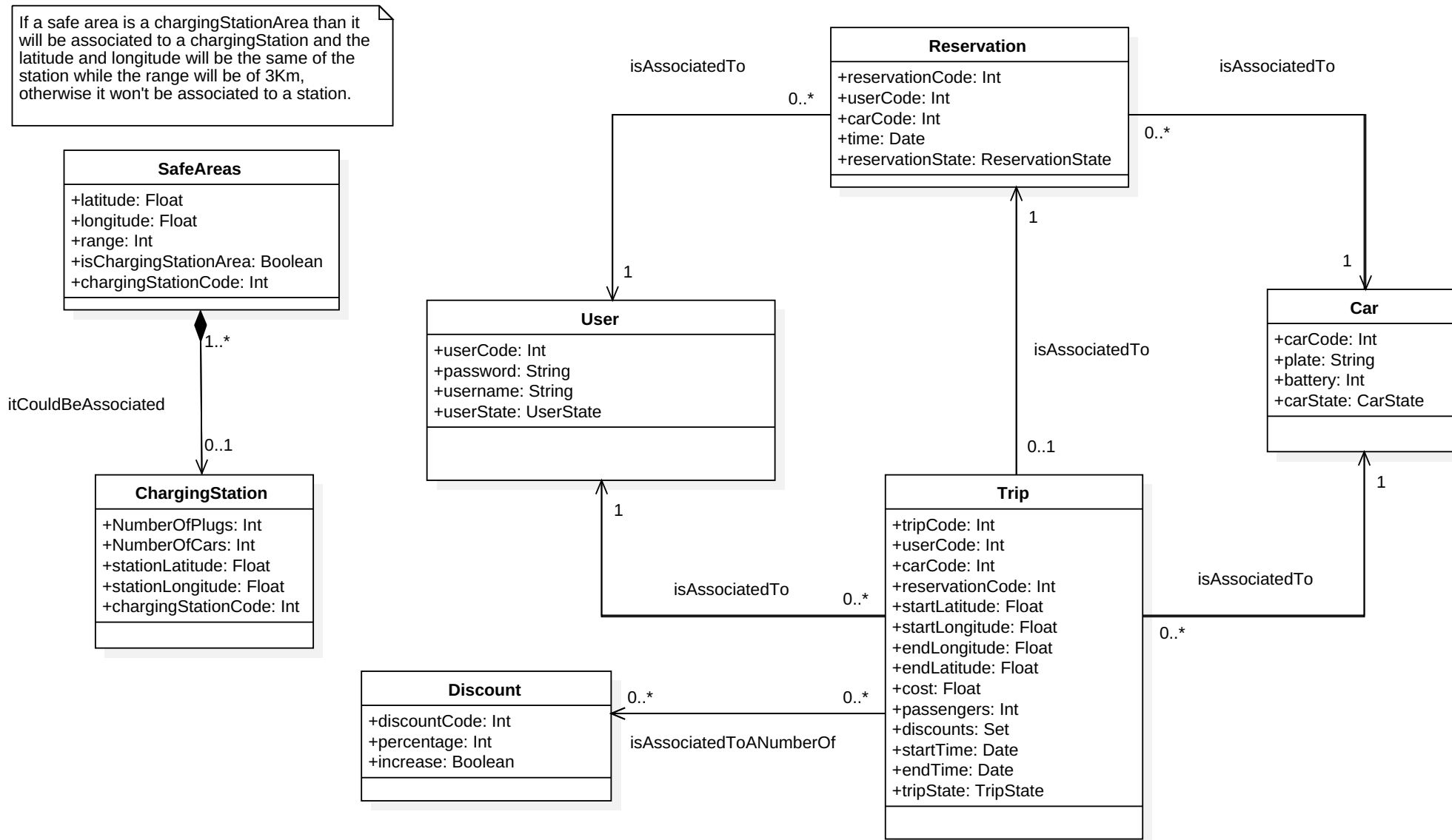
---



# COMPONENT DIAGRAM

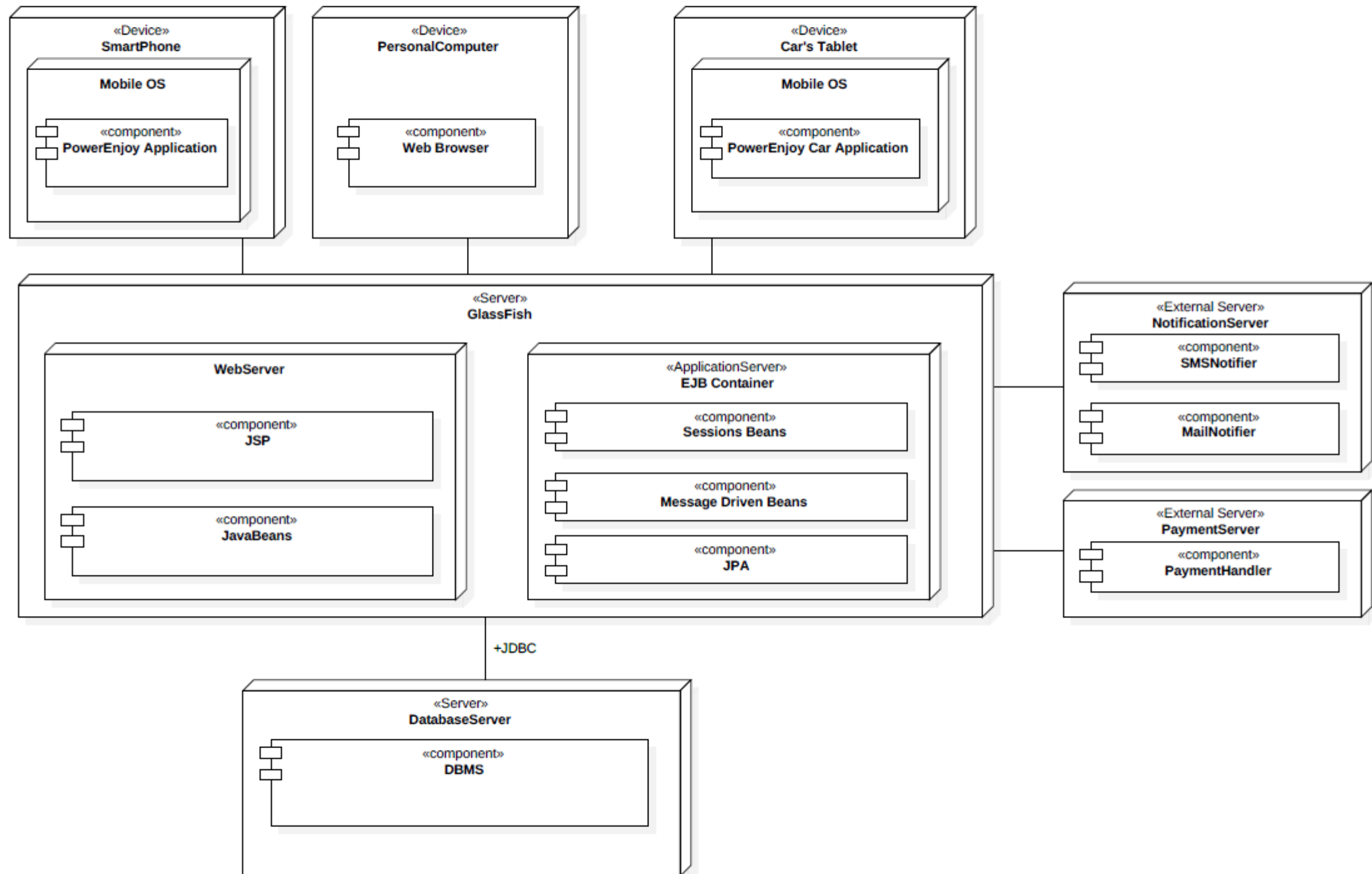


# DATA STRUCTURE



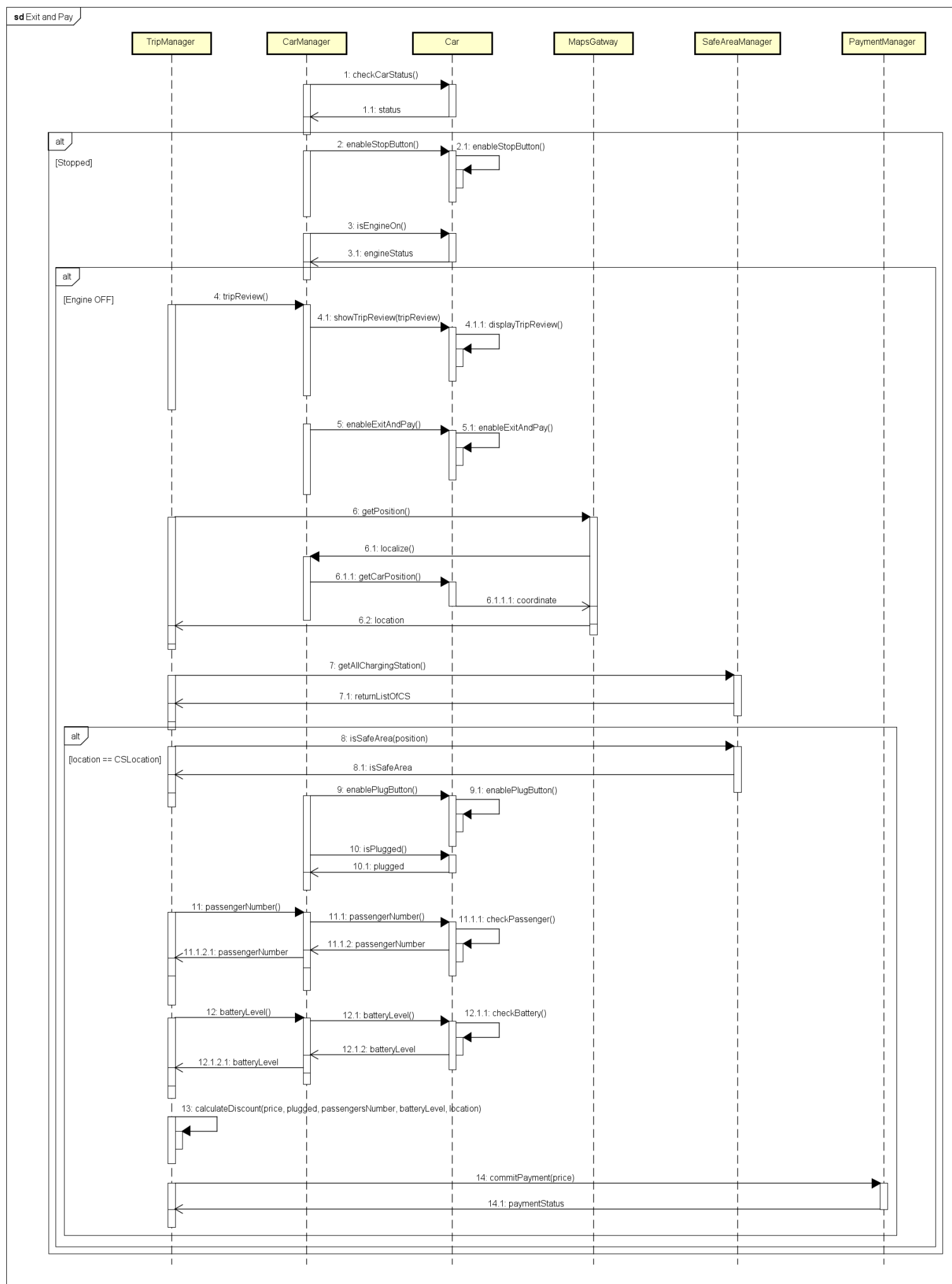
# DEPLOYMENT VIEW

---

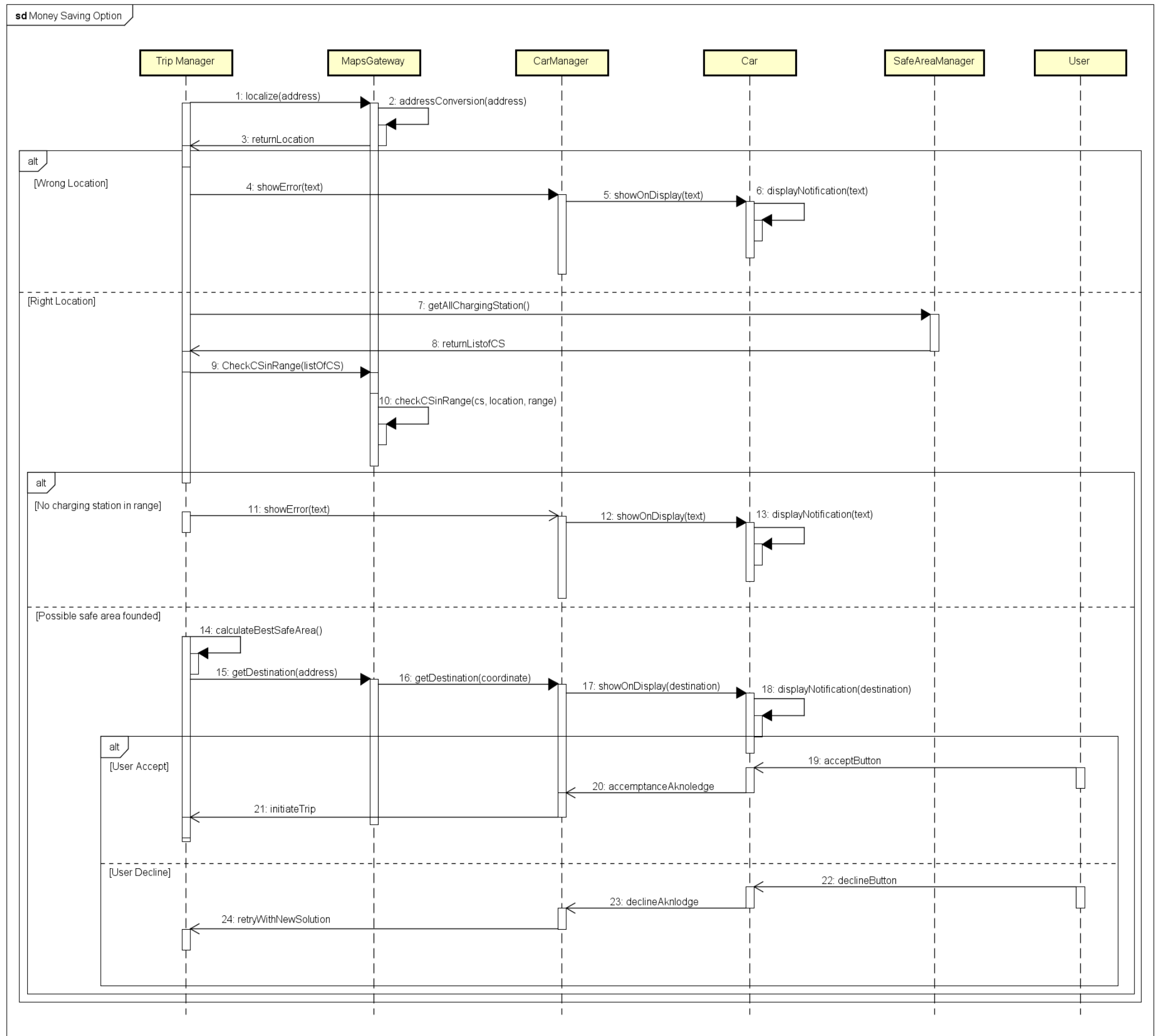




# SEQUENCE DIAGRAM



# SEQUENCE DIAGRAM



# ALGORITHM DESIGN

---

This first algorithm manages the “Money Saving” option in our system. We created a specific class for the location, to provide coherent and consistent information about car’s and charging station’s position. So, this function receives an address, the one that user has typed on the car’s display, and a range, also typed on display. CarManager plays an important role during this situation, because permits to interact with the display and consequentially with the user in the car. We created also a specific class for the charging station, which are provided by SafeAreaManager during the computation. MapsGateway is a third part service and can calculate if a specific charging station is or not in range, from the location desired by the user. In the end, before the user can see the result on display, the system checks which charging station has less car and it advices its position to the user.

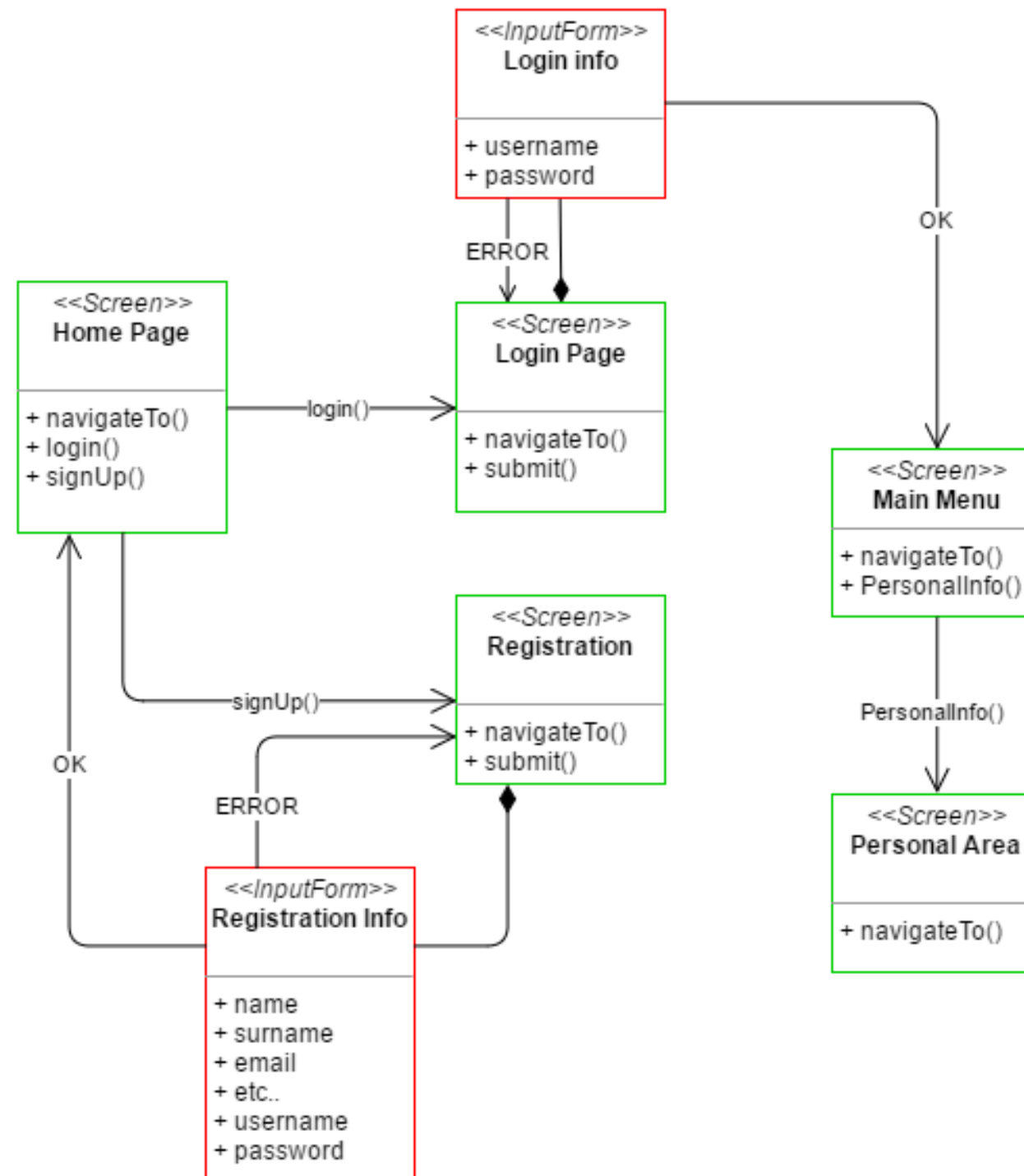
# ALGORITHM DESIGN

---

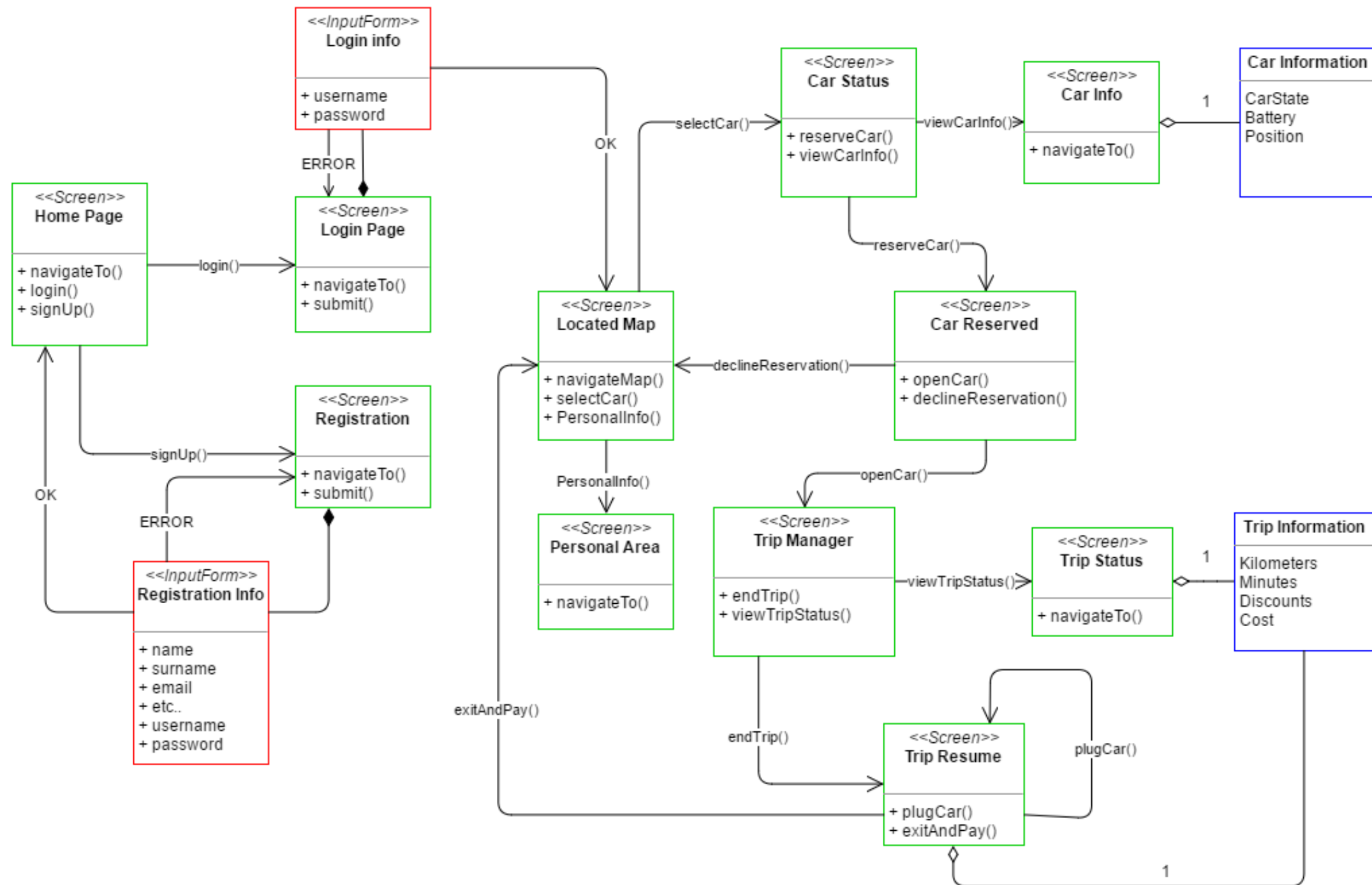
This second algorithm has the aim to explain the conclusion of the rent. This difficult situation has lot of possible alternatives so we should check more details in order to guarantee coherence between the trip and the payment. After variable's declaration, this algorithm begins to check the status of the car (precisely, if it stopped or not) and then if the user asks to stop its trip. As soon as the engine is stopped, the system stops charging the user. Then car's tablet shows the review and the details of the trip, calculated thanks to TripReview. The user can now touch the "Exit and Pay" button on the display, that is enabled coherently after some checks. The algorithm saves current location of the car thanks to CarManager that sends coordinates to MapsGateway, and then it processing received data in order to retrieve a location. The system downloads all the charging station, and this choice is justified by the fact we want to keep always updated the list of active charging station (with this method, system's administrator only should update the central database after, for example, a charging station's braking). If the user stops near a charging station, and then he is in a safe area, CarManager enables the possibility to plug the car as well as enables "plug" button. After TripManager calculates right discount of the trip (considering also passengers and battery level), PaymentManager tries to commit the payment, and the system must check if it successful. If it is unsuccessful, we provide the possibility to retry the payment manually and change some information about the payment's method through car's display. After three attempts, the user will be charged to discourage this event.

# USER INTERFACE DESIGN

---



# USER INTERFACE DESIGN



**THANKS FOR YOUR ATTENTION**

---