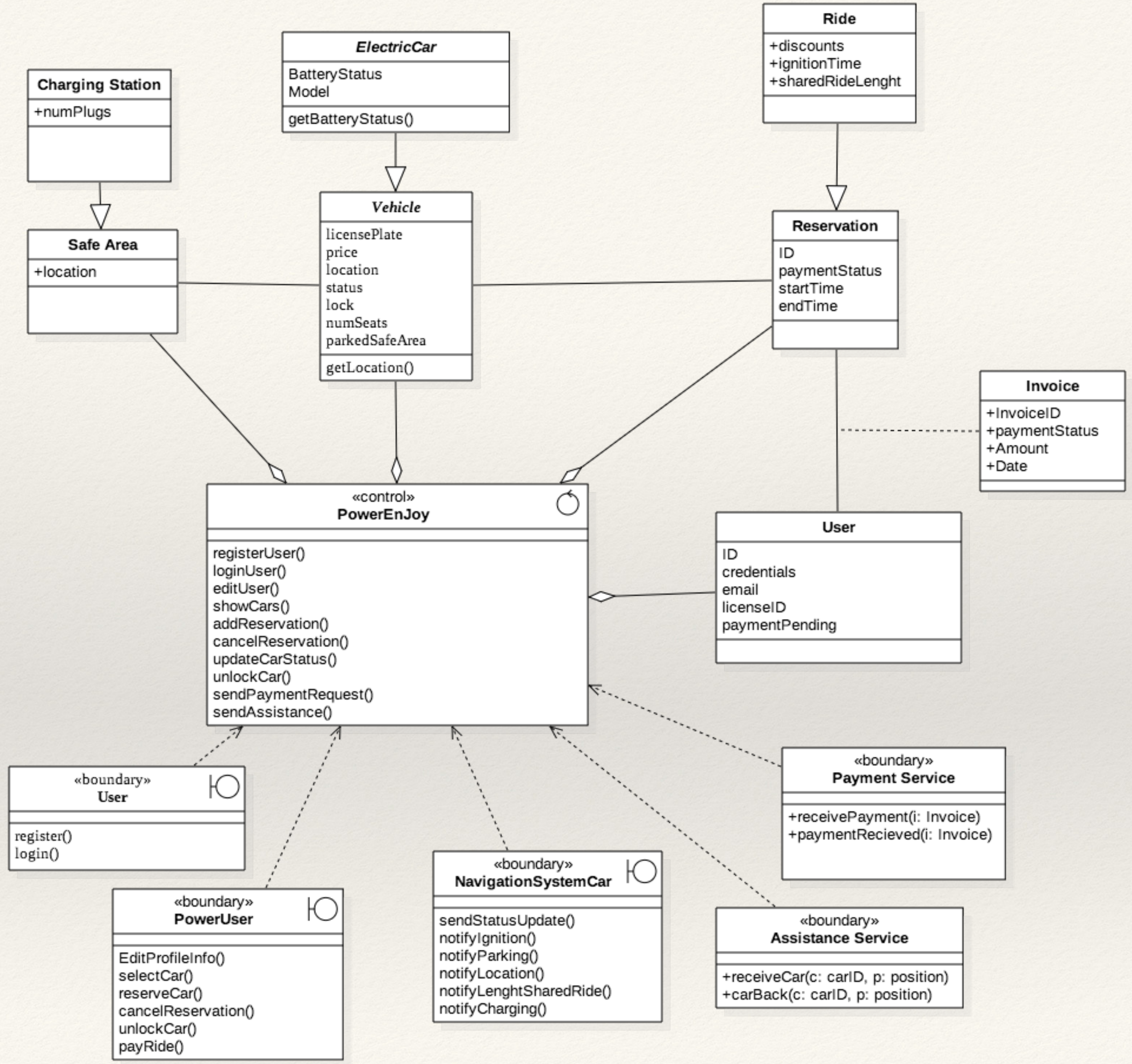
Software Engineering 2 Project

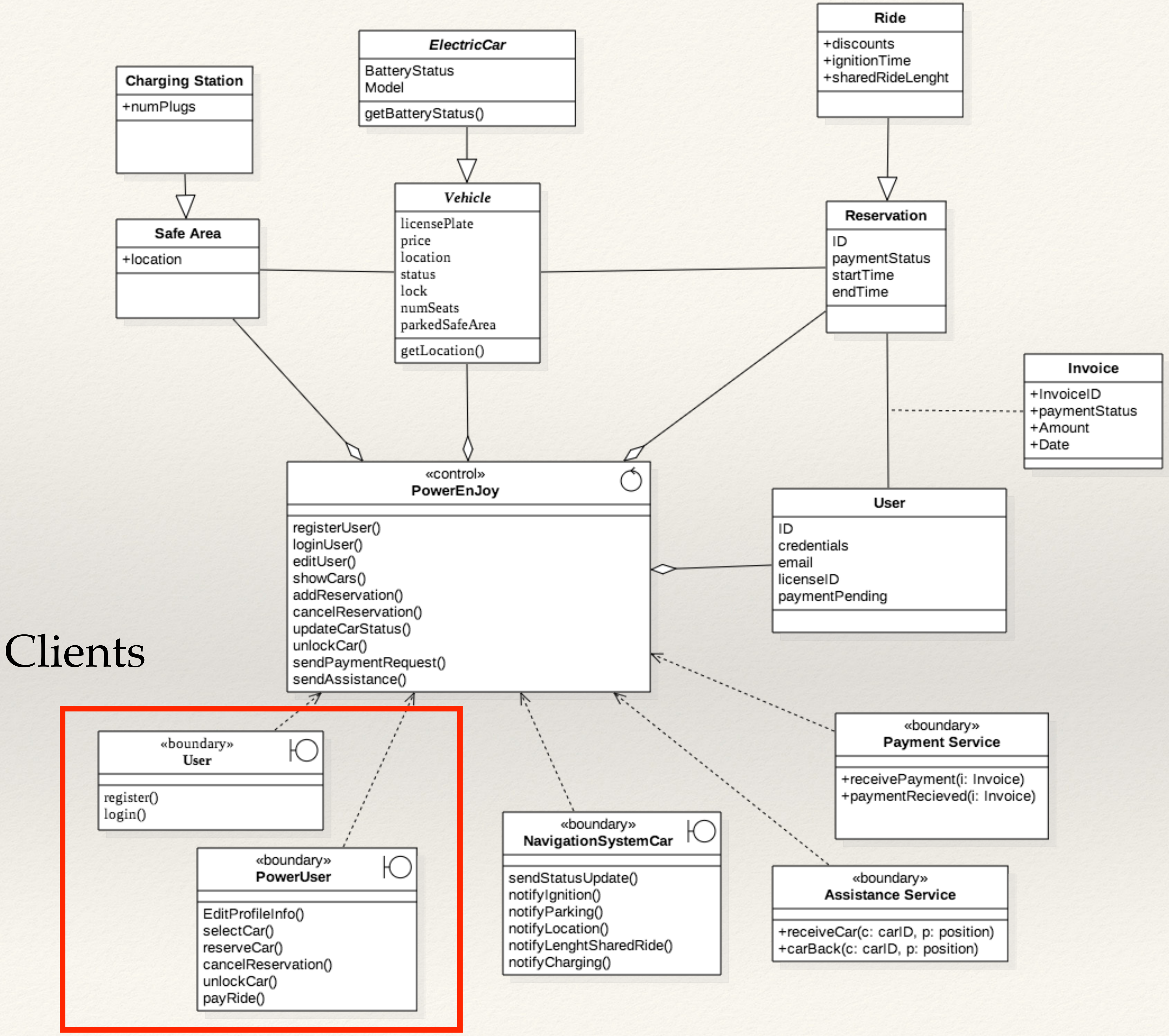
Design Document

Niccolo' Raspa
Matteo Marinelli

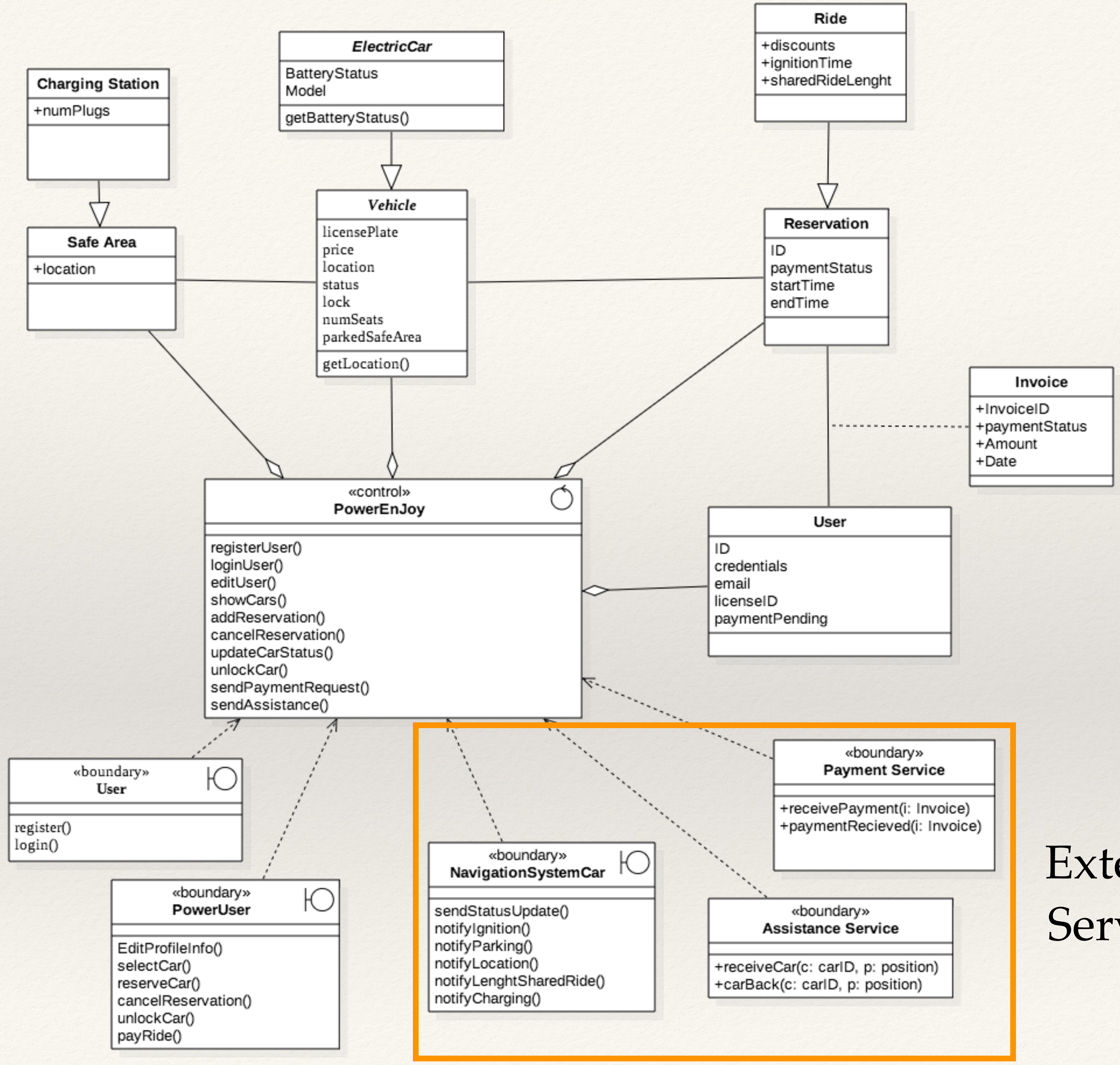
From Rasd To Design Document…


Charging Station
+numPlugs
ElectricCar
BatteryStatus
Model
getBatteryStatus()
Ride
+discounts
+ignitionTime
+sharedRideLenght
Safe Area
+location
Vehicle
licensePlate
price
location
status
lock
numSeats
parkedSafeArea
getLocation()
Reservation
ID
paymentStatus
startTime
endTime
Invoice
+InvoiceID
+paymentStatus
+Amount
+Date
«control»
PowerEnJoy
registerUser()
loginUser()
editUser()
showCars()
addReservation()
cancelReservation()
updateCarStatus()
unlockCar()
sendPaymentRequest()
sendAssistance()
User
ID
credentials
email
licenseID
paymentPending
«boundary»
User
register()
login()
«boundary»
PowerUser
EditProfileInfo()
selectCar()
reserveCar()
cancelReservation()
unlockCar()
payRide()
«boundary»
NavigationSystemCar
sendStatusUpdate()
notifyIgnition()
notifyParking()
notifyLocation()
notifyLenghtSharedRide()
notifyCharging()
«boundary»
Assistance Service
+receiveCar(c: carID, p: position)
+carBack(c: carID, p: position)
«boundary»
Payment Service
+receivePayment(i: Invoice)
+paymentRecieved(i: Invoice)

Clients

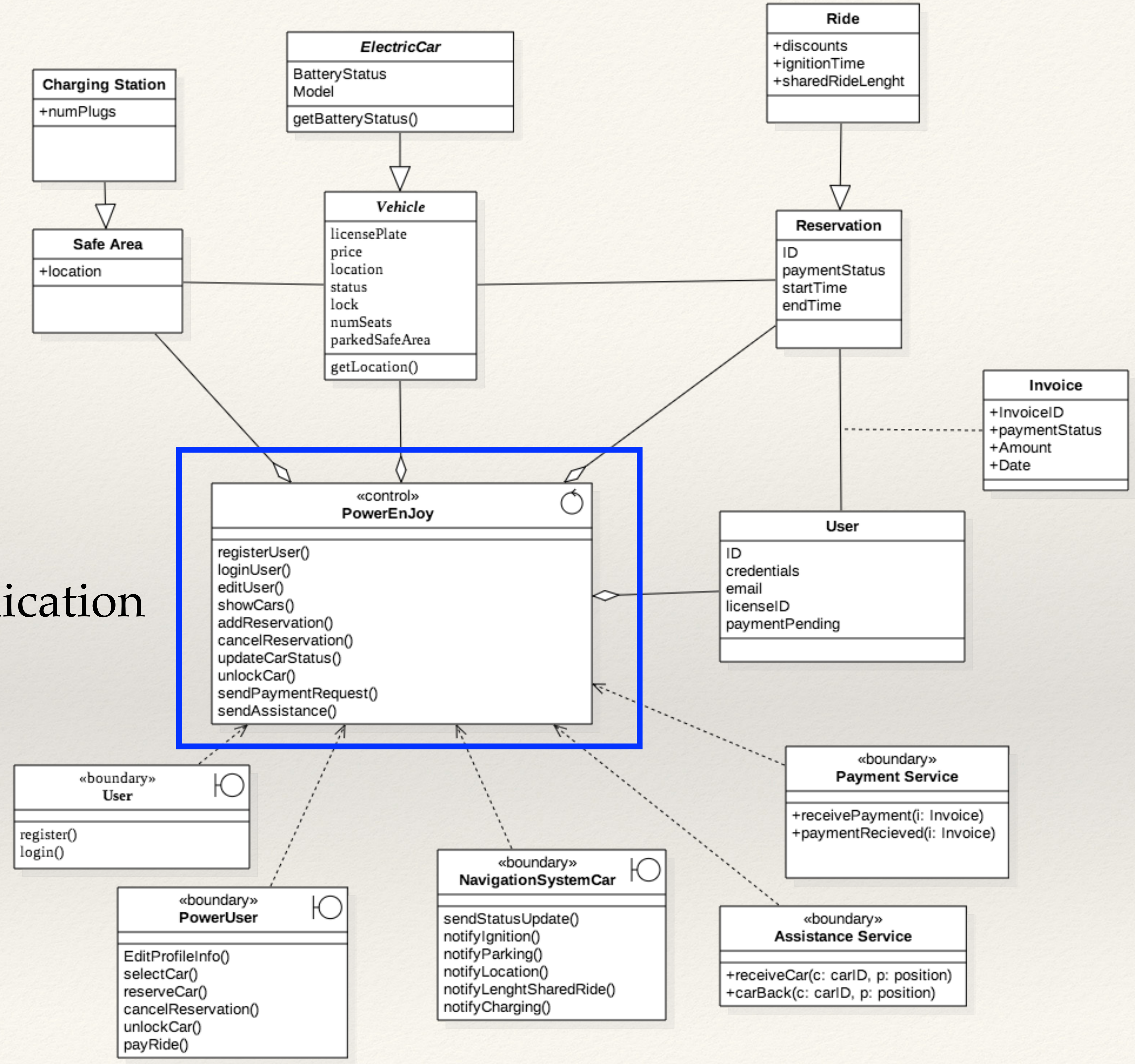


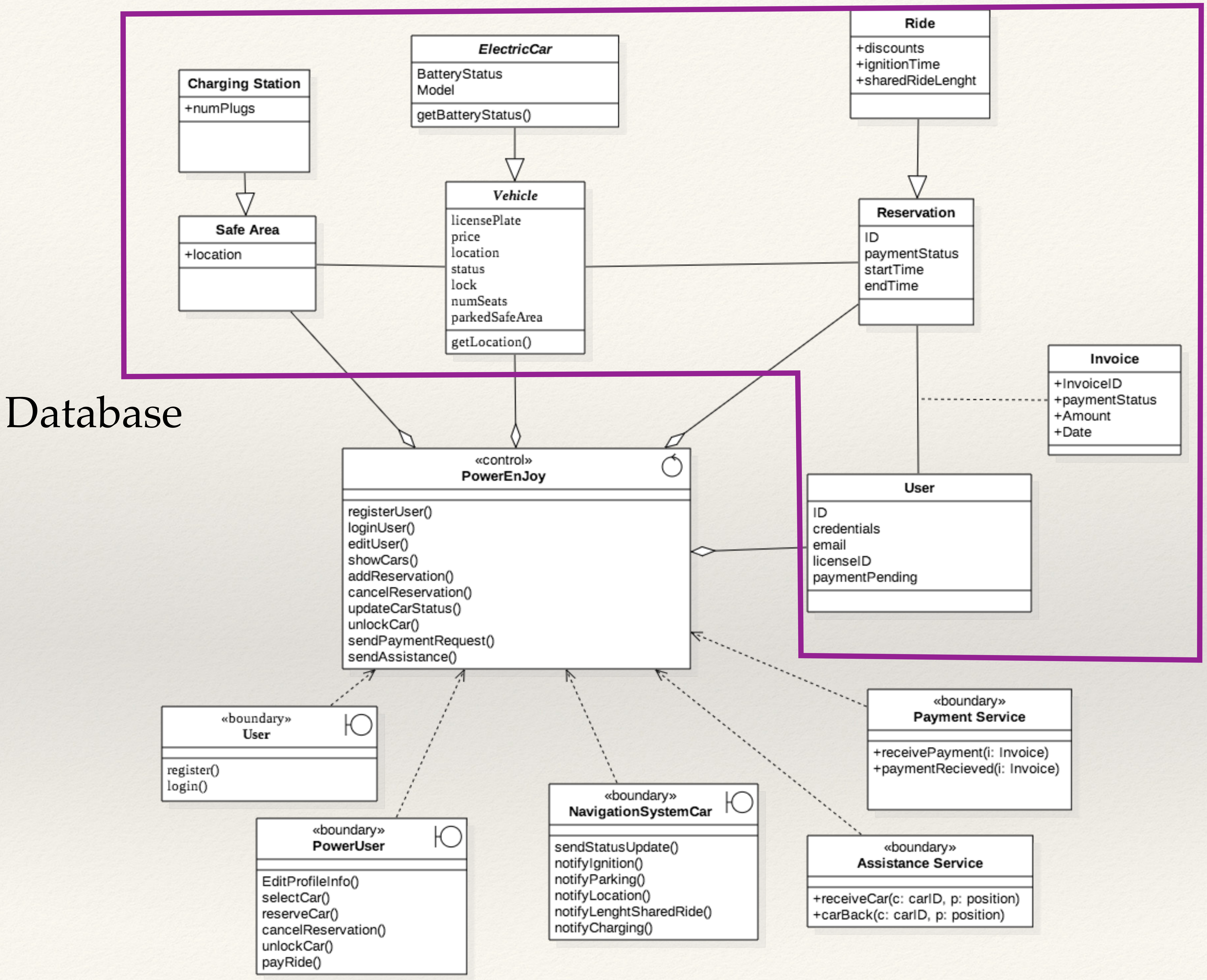
Charging Station
+numPlugs
ElectricCar
BatteryStatus
Model
getBatteryStatus()
Ride
+discounts
+ignitionTime
+sharedRideLenght
Safe Area
+location
Vehicle
licensePlate
price
location
status
lock
numSeats
parkedSafeArea
getLocation()
Reservation
ID
paymentStatus
startTime
endTime
Invoice
+InvoiceID
+paymentStatus
+Amount
+Date
«control»
PowerEnJoy
registerUser()
loginUser()
editUser()
showCars()
addReservation()
cancelReservation()
updateCarStatus()
unlockCar()
sendPaymentRequest()
sendAssistance()
User
ID
credentials
email
licenseID
paymentPending
«boundary»
User
register()
login()
«boundary»
PowerUser
EditProfileInfo()
selectCar()
reserveCar()
cancelReservation()
unlockCar()
payRide()
«boundary»
NavigationSystemCar
sendStatusUpdate()
notifyIgnition()
notifyParking()
notifyLocation()
notifyLenghtSharedRide()
notifyCharging()
«boundary»
Payment Service
+receivePayment(i: Invoice)
+paymentRecieved(i: Invoice)
«boundary»
Assistance Service
+receiveCar(c: carID, p: position)
+carBack(c: carID, p: position)



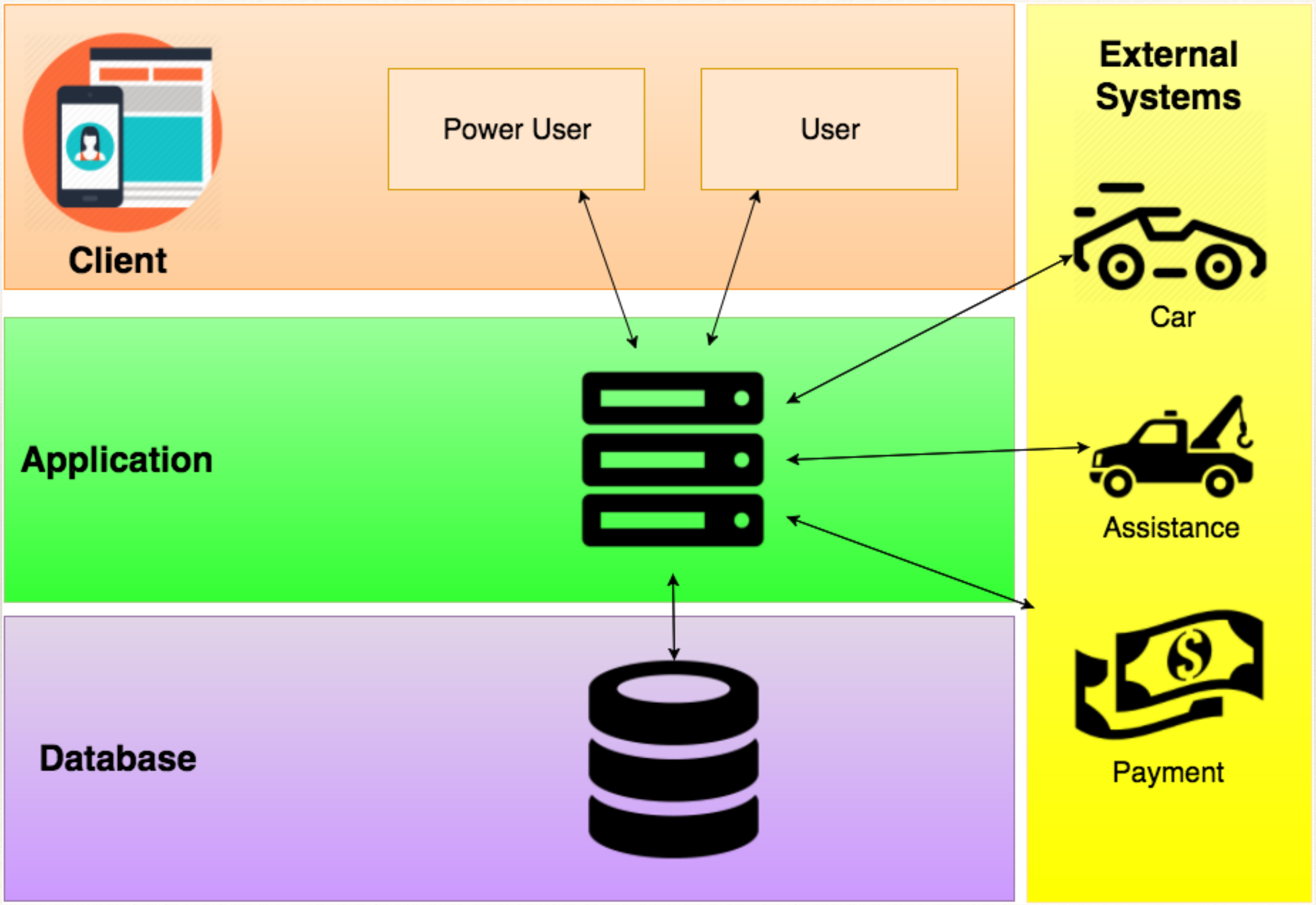
External Services

Application



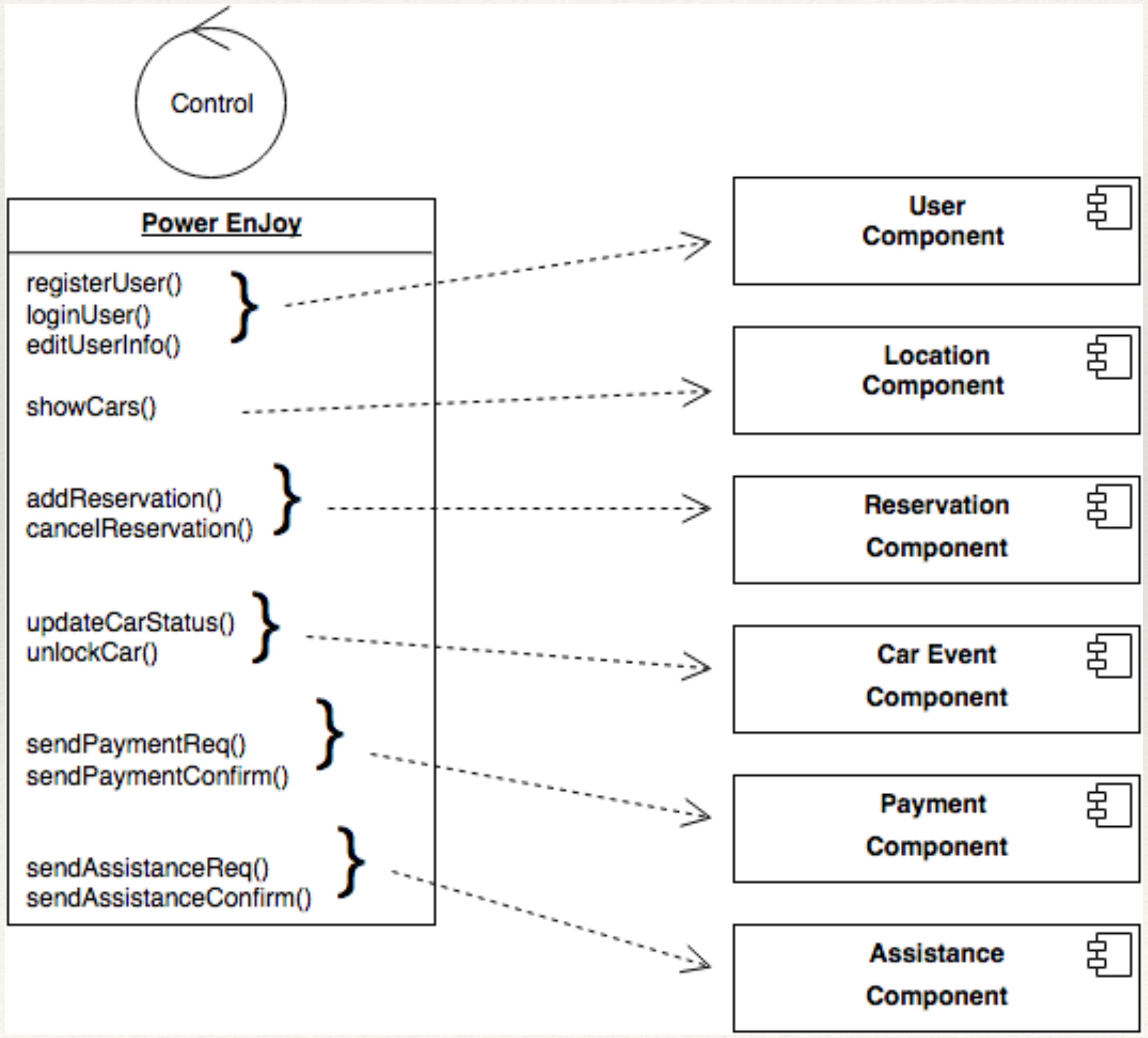


Charging Station
+numPlugs
ElectricCar
BatteryStatus
Model
getBatteryStatus()
Ride
+discounts
+ignitionTime
+sharedRideLenght
Safe Area
+location
Vehicle
licensePlate
price
location
status
lock
numSeats
parkedSafeArea
getLocation()
Reservation
ID
paymentStatus
startTime
endTime
Invoice
+InvoiceID
+paymentStatus
+Amount
+Date
Database
«control»
PowerEnJoy
registerUser()
loginUser()
editUser()
showCars()
addReservation()
cancelReservation()
updateCarStatus()
unlockCar()
sendPaymentRequest()
sendAssistance()
User
ID
credentials
email
licenseID
paymentPending
«boundary»
User
register()
login()
«boundary»
Payment Service
+receivePayment(i: Invoice)
+paymentRecieved(i: Invoice)
«boundary»
PowerUser
EditProfileInfo()
selectCar()
reserveCar()
cancelReservation()
unlockCar()
payRide()
«boundary»
NavigationSystemCar
sendStatusUpdate()
notifyIgnition()
notifyParking()
notifyLocation()
notifyLenghtSharedRide()
notifyCharging()
«boundary»
Assistance Service
+receiveCar(c: carID, p: position)
+carBack(c: carID, p: position)


Client
Power User
User
External Systems
Car
Application
Assistance
Payment
Database

Application Layer

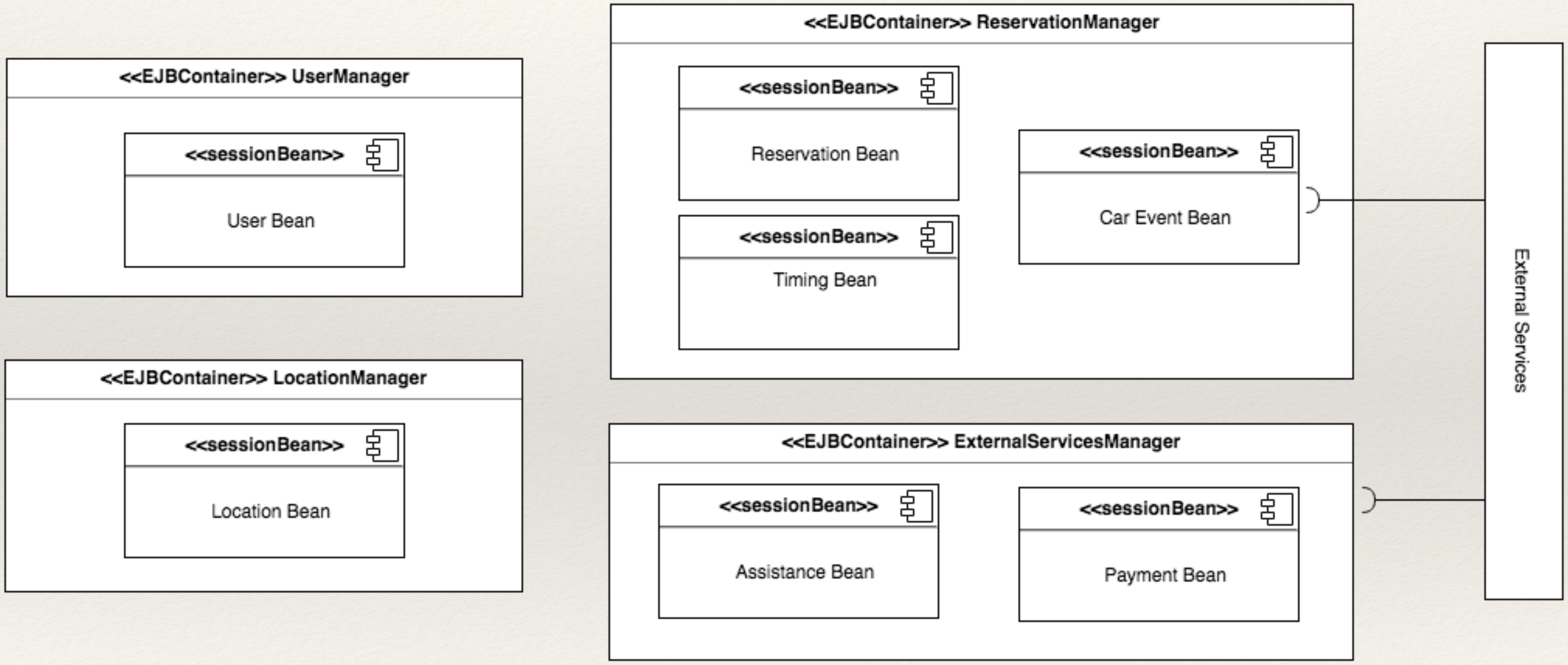
- This components implements the logic of the Power Enjoy Application, it's the core of our business



Application Layer - Implementation

- **Java Enterprise Edition 7 (JEE)**
  - Modular Components
  - Large Scale
  - Multi Tiered
  - Scalable
- **Enterprise Java Beans (EJB)**
  - Encapsulate Business Logic
- **GlassFish as Application Server**
  - Supports JEE7
  - Additional Features (Security, Load Balancing)

Application Layer - EJB



Client Layer

- **Considerations:**
    - Mobility In Mind
    - Mobile First
- **Expected Functionalities:**
    - Registration
    - Login
    - Edit Profile
    - See Recent Rides
    - Reservation/Ride
    - Make Payment

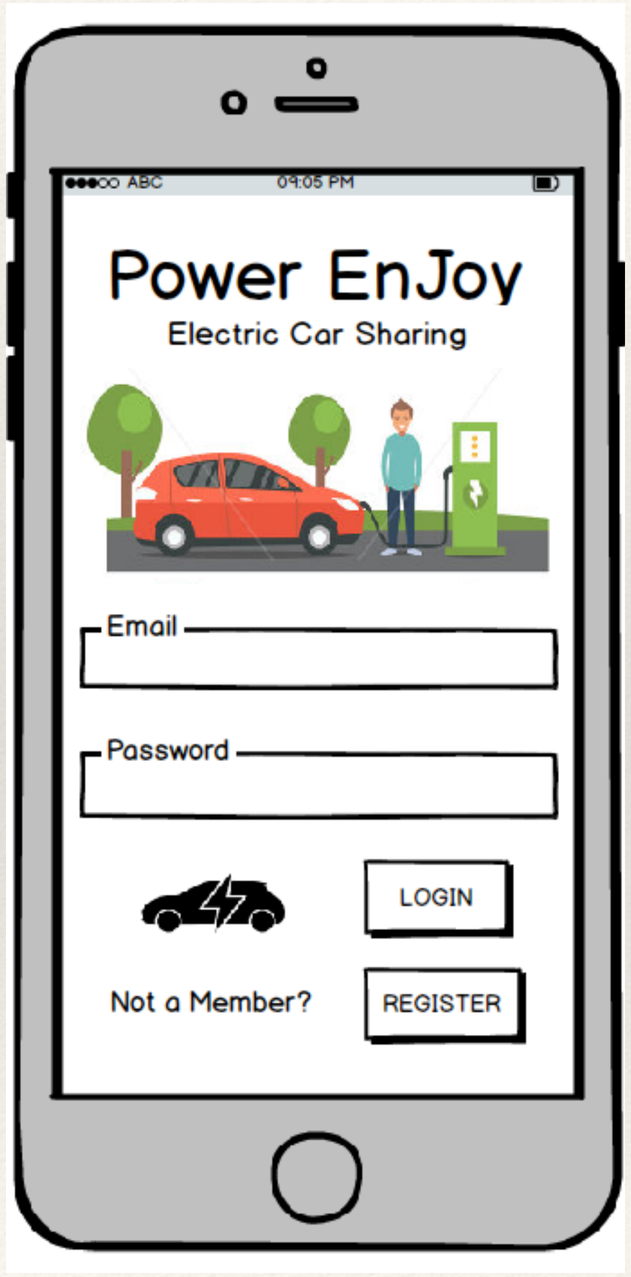
Client Layer

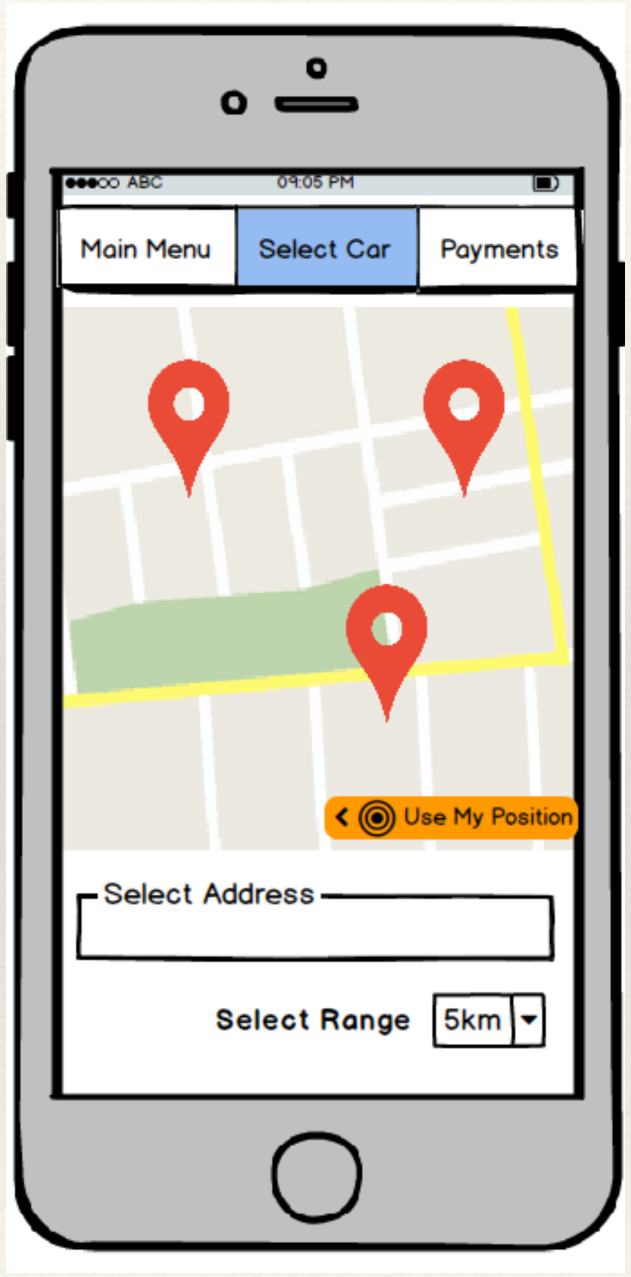
- **Considerations:**
  - Mobility In Mind
  - Mobile First
- **Expected Functionalities:**
  - Registration
  - Login
  - Edit Profile
  - See Recent Rides
  - Reservation/Ride
  - Make Payment

Profile Management

Car Sharing

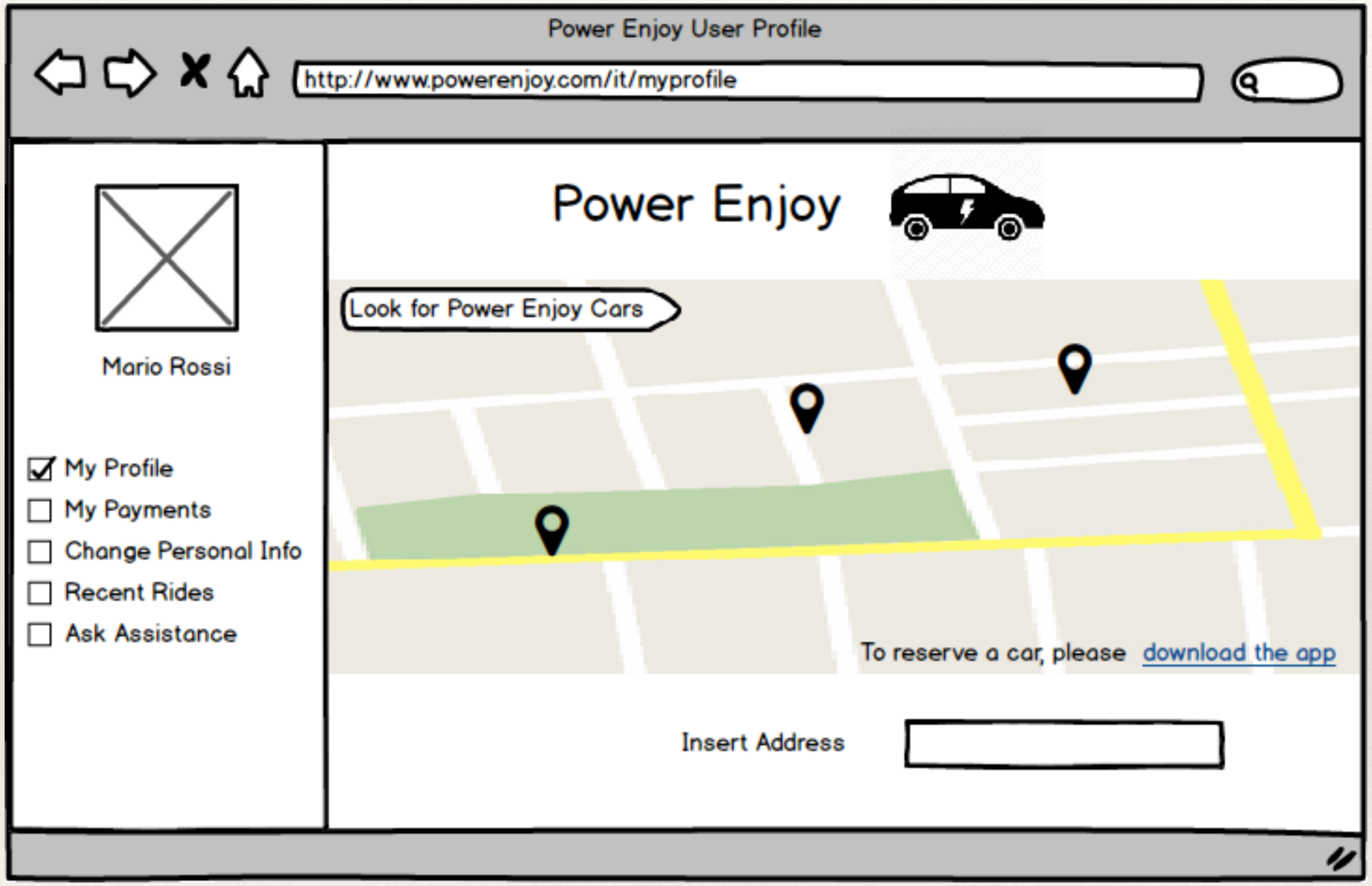
Client Layer - Mobile App



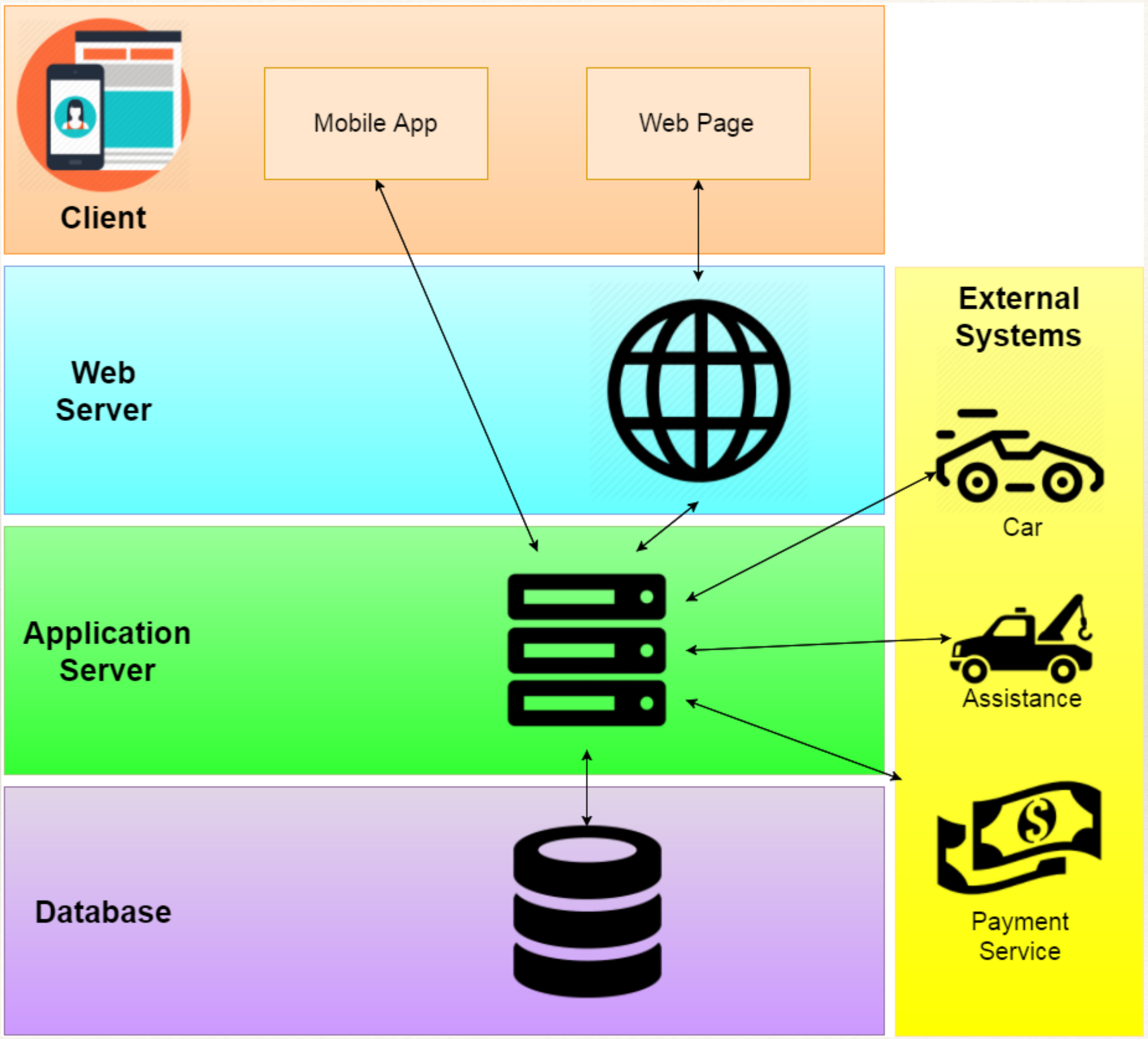


- Mobile App implements all expected functionalities
- Mobile First but not Mobile Only:
  - visibility
  - accessibility
  - scalability

Client Layer - Web Server



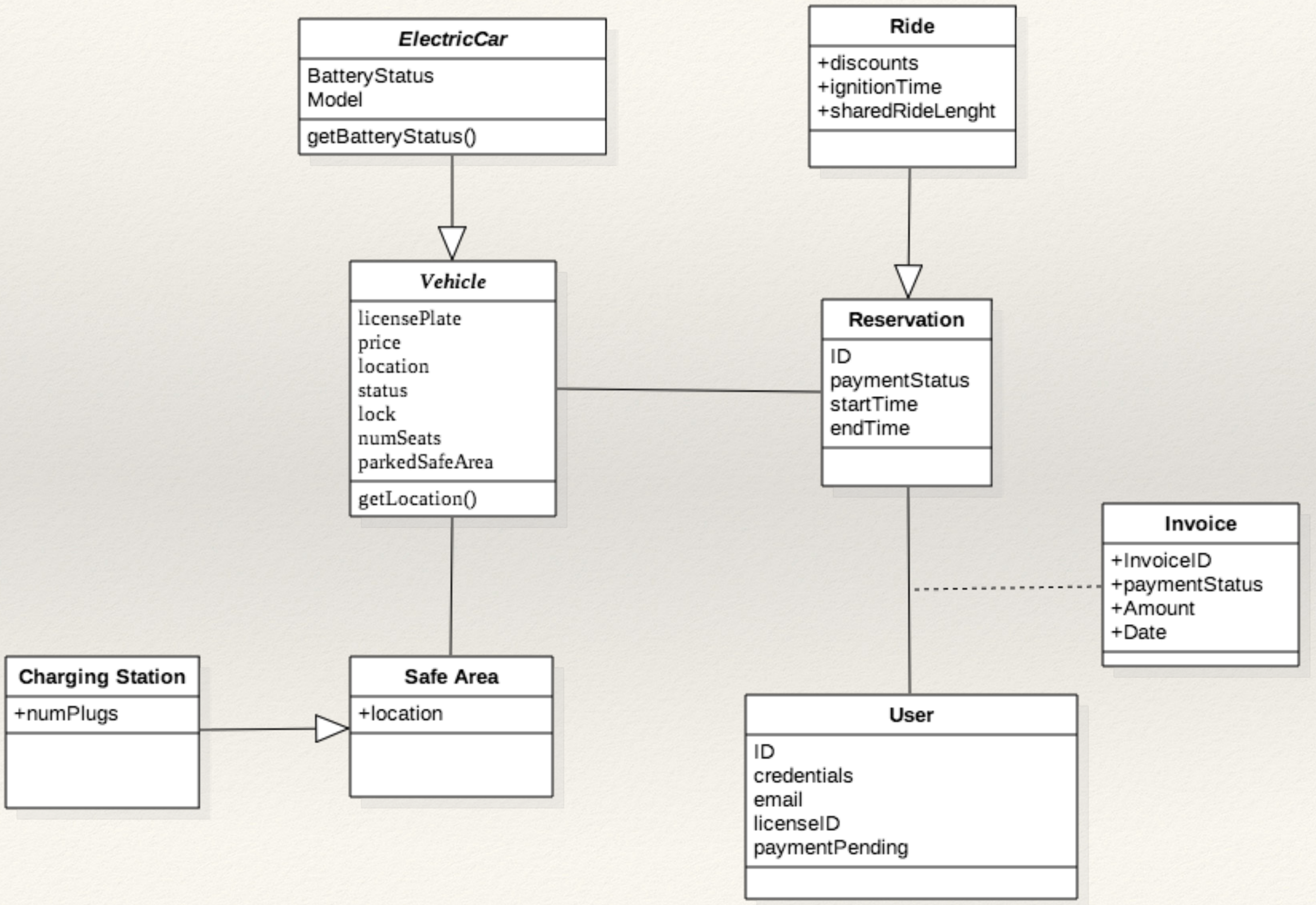
- Support to Mobile App
- New Tier?


Client
Mobile App
Web Page
Web Server
Application Server
Database
External Systems
Car
Assistance
Payment Service

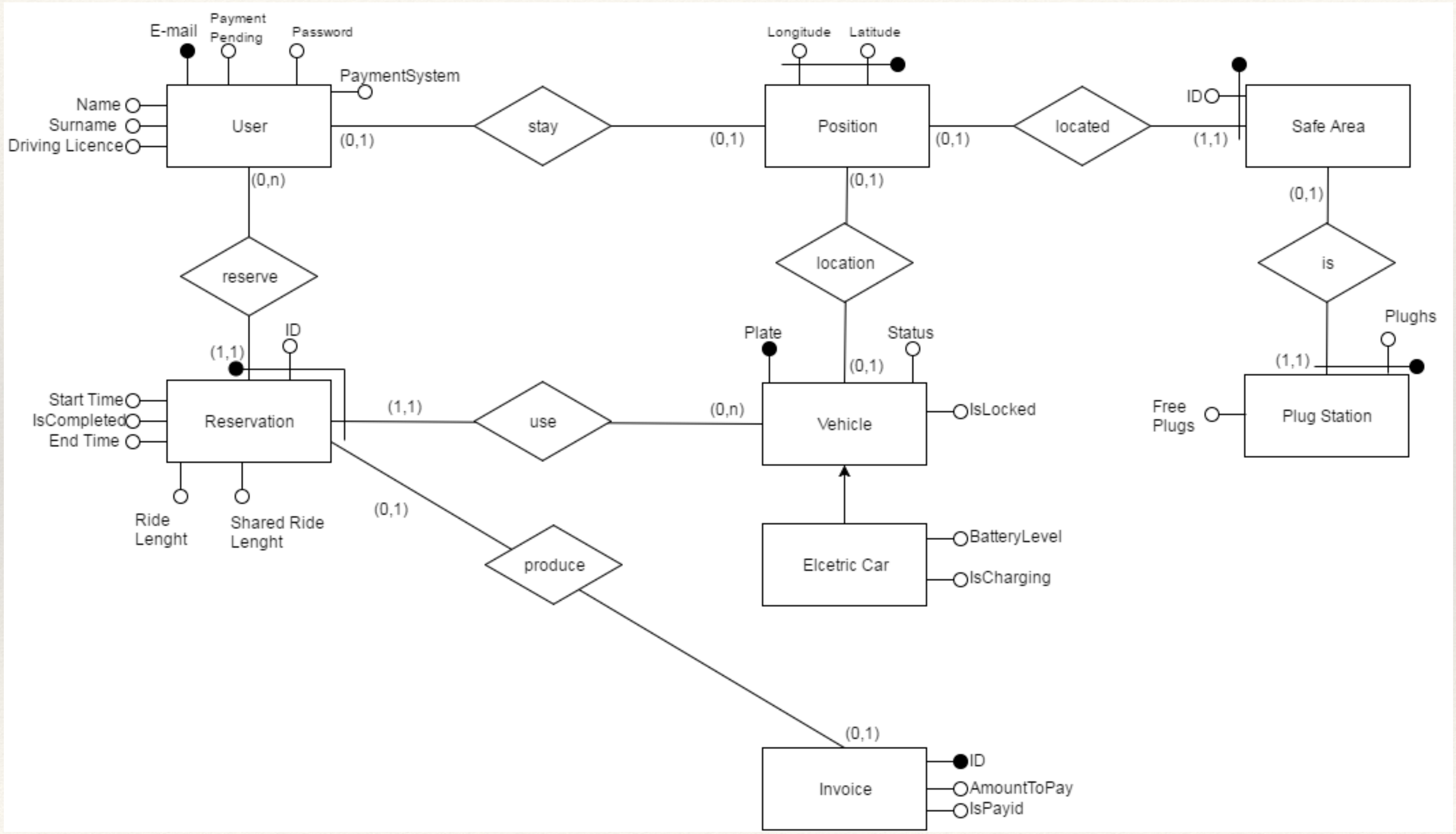
Client Layer - Implementation

- **Web Server:**
  - GlassFist with Java Server Pages
  - Communication to Application Server via RESTful APIs
- **Mobile App:**
  - Cordova
  - Cost Effective: free
  - Easy to modify: open source
  - Resource Effective: target multiple devices with one codebase

Database Layer



Database Layer

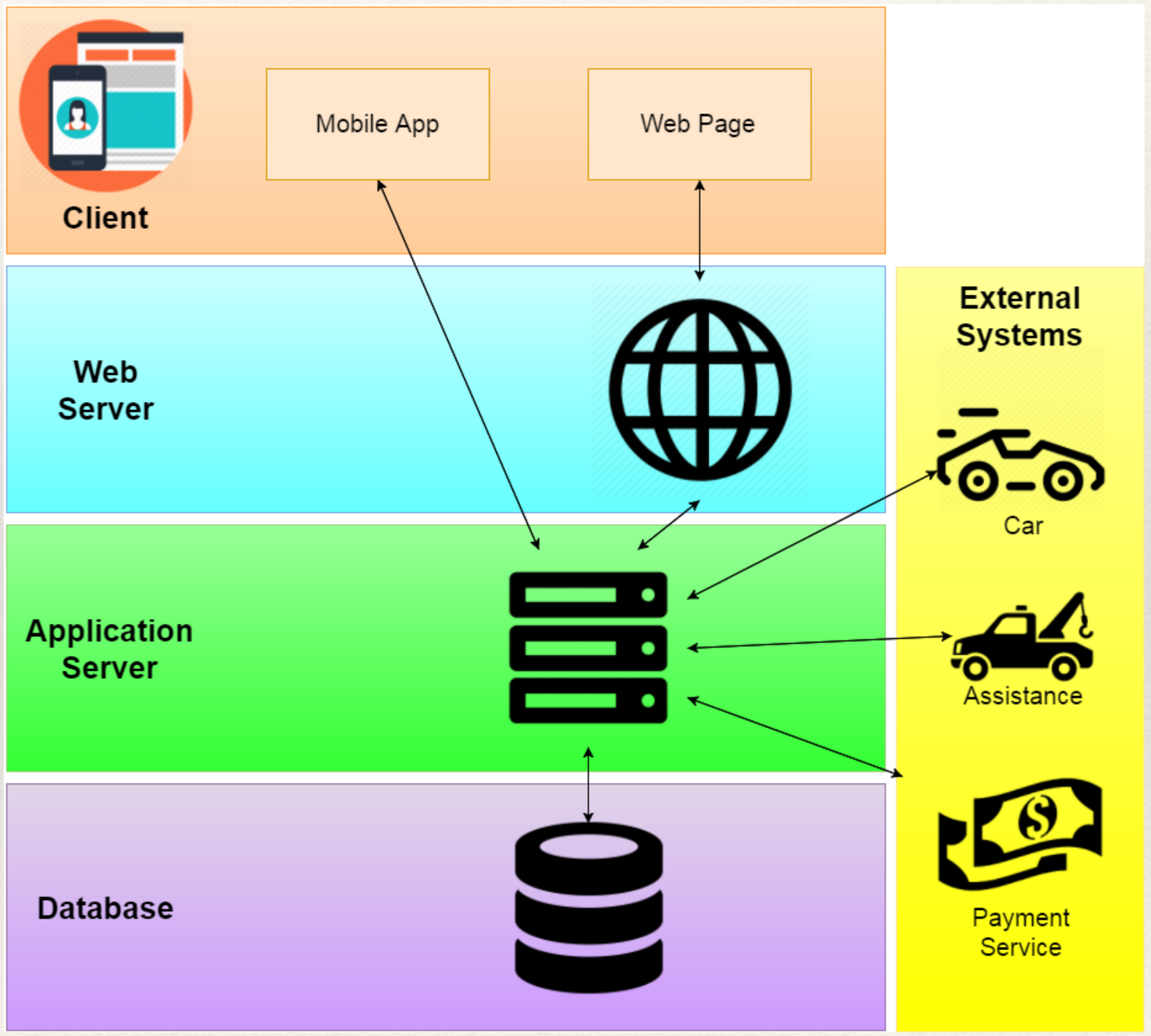


Database Layer

- **MySQL** as the relational database
    - Scalability
    - Flexibility
    - High Performance
    - High Availability
    - Easy to access from Application Server via JDBC

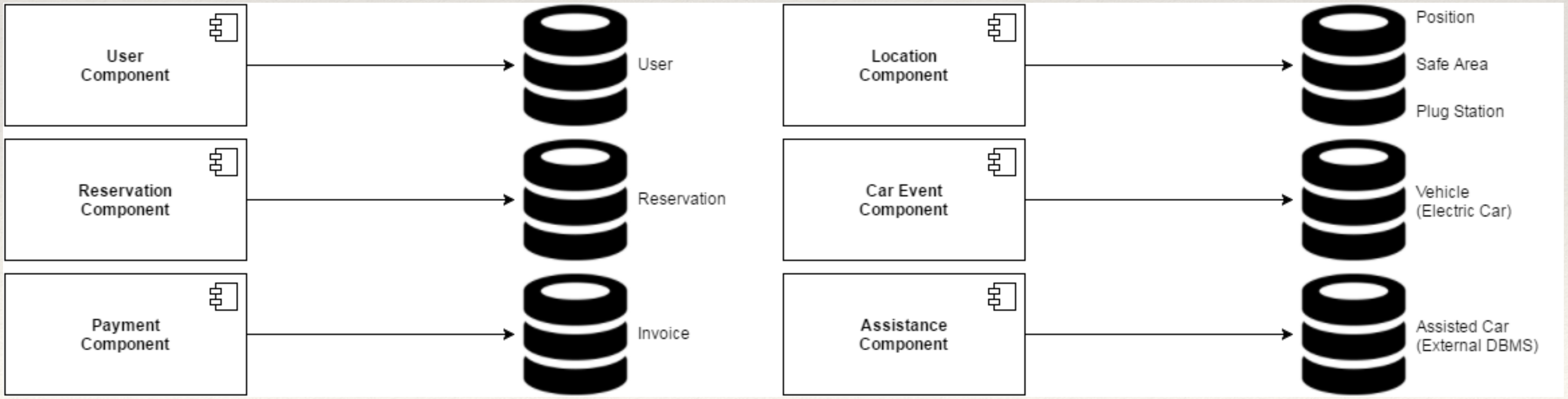
Some Problems…

- **Problems**
- Application Server is the bottleneck of our system
- The performance of this layer is strictly related to the overall performance of the system
- **Solutions**
- Multithreading?
- Sure, but we can do better…



Moving to a SOA approach

- Split the workload among different services
  - Simple and clear Interface to other components
  - Each component is responsible for some entities in the database



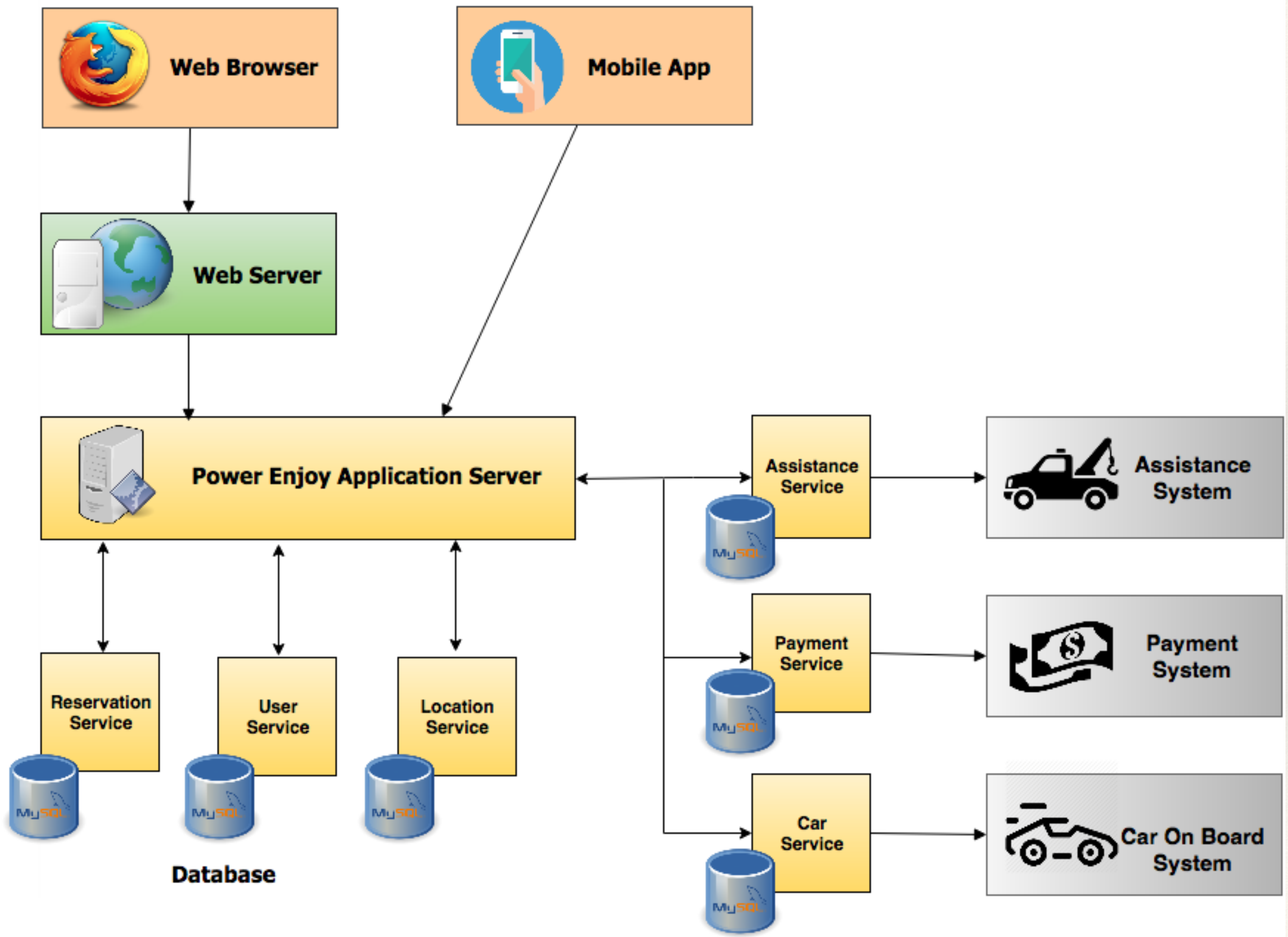
Benefits

- It’s a more **clean architecture**. Every component implements a service and provides an interface to all the other services.

- Changing/optimising each module **will not affect the whole system** as long as we maintain the same interface for each component.

- It’s very flexible, it’s will be easy in the future to **add new functionalities**.

- We can **divide the databases among different regions** (e.g. for the city of Milan we don’t need to keep track of the cars in Turin)


Web Browser
Mobile App
Web Server
Power Enjoy Application Server
Assistance Service
Assistance System
Payment Service
Payment System
Car Service
Car On Board System
Reservation Service
User Service
Location Service
MySQL
Database