PowerEnjoy Project

Design Document presentation

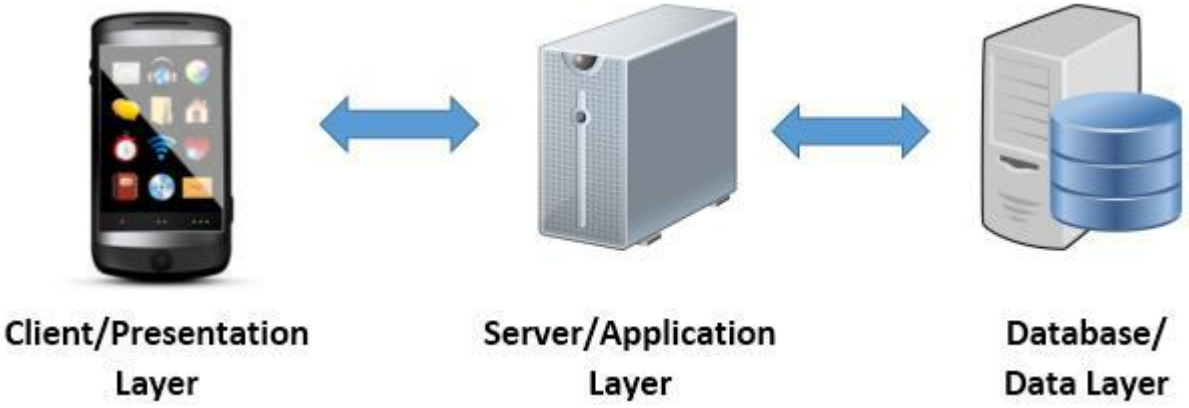
Vianney Payelle - Rémi Rigal - Noëlie Ramuzat

Content

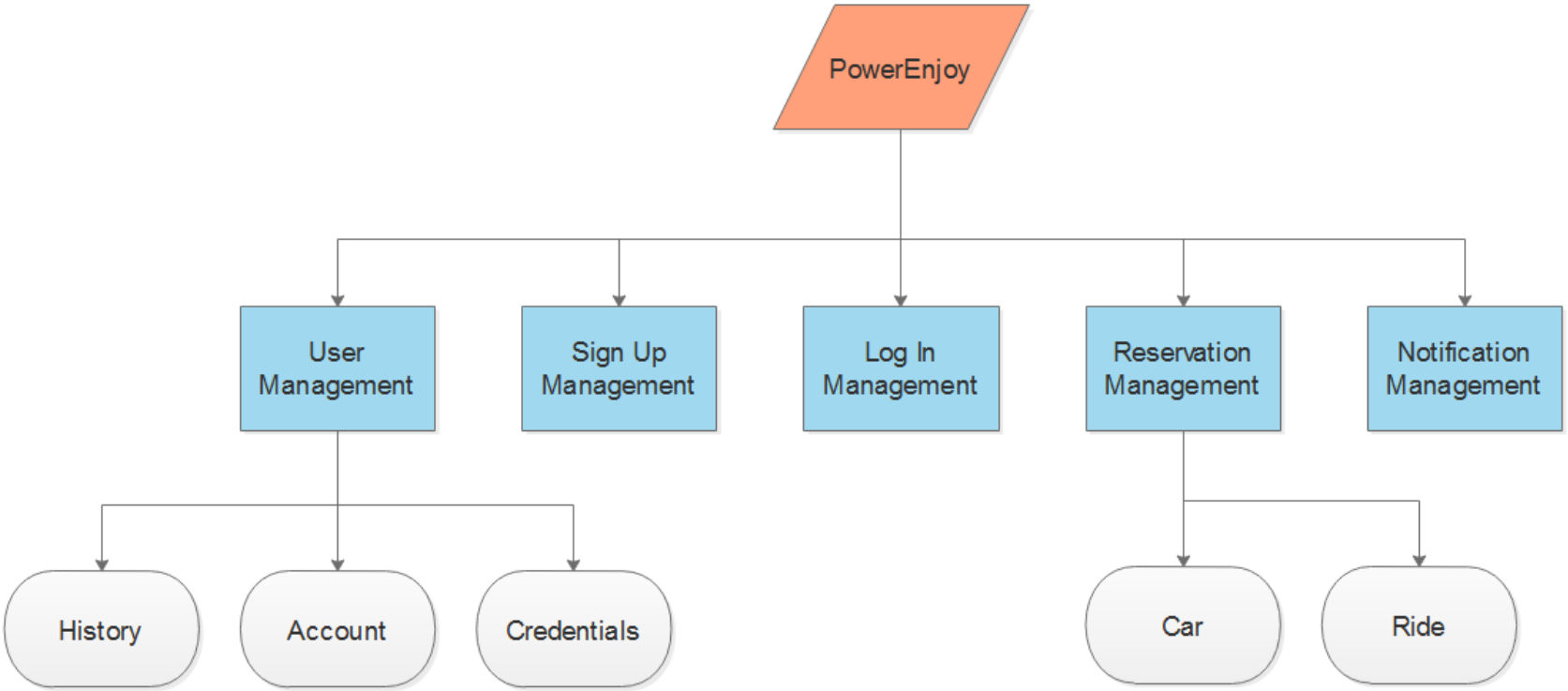
- Overview
- High Level Component
- Component View
- Deployment Diagram
- Sequence Diagrams Examples
- Algorithms Examples
- BCE Diagram
- Requirements Traceability

Overview

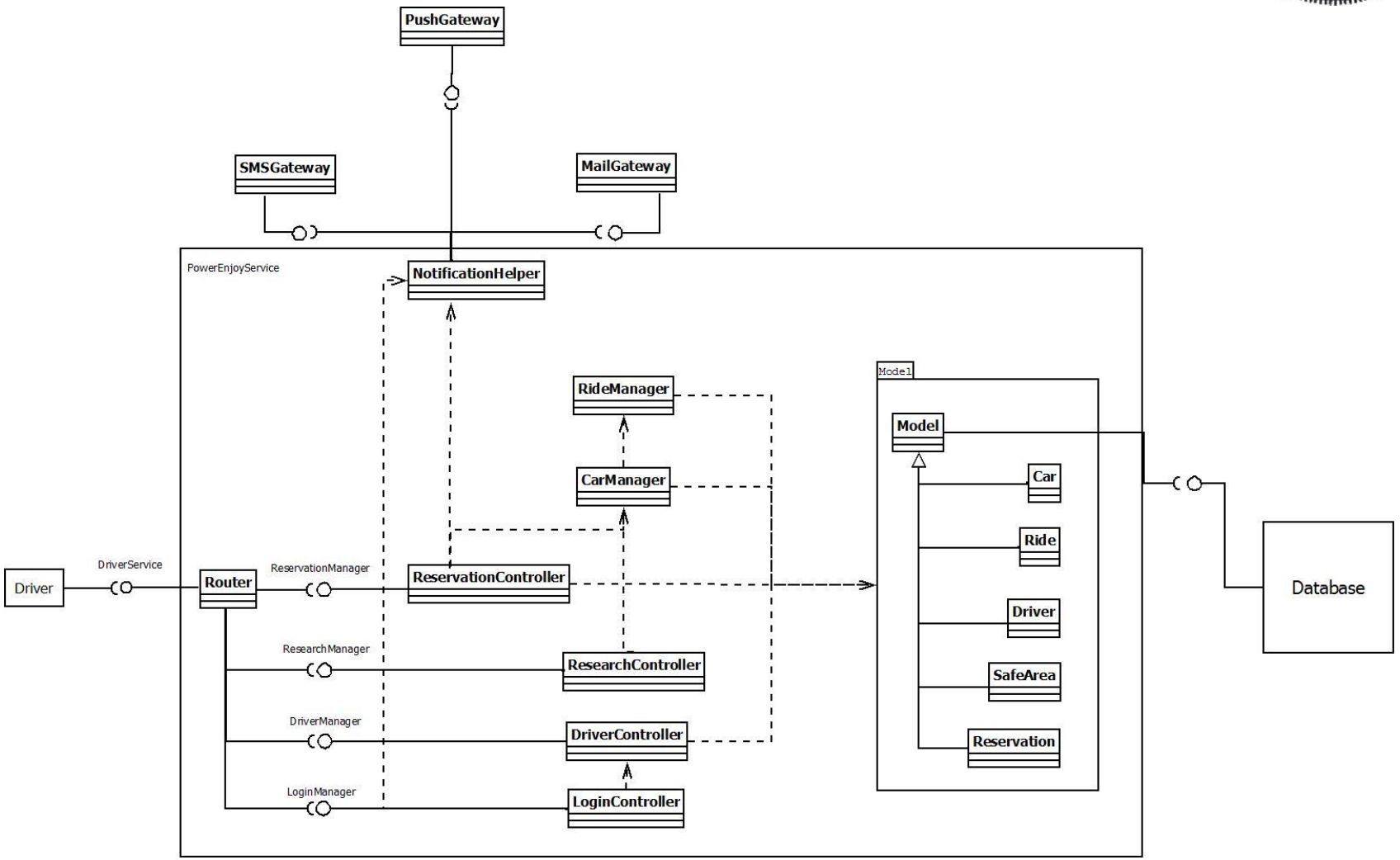
- Three tier architecture:
  - Client
  - Server
  - DBMS



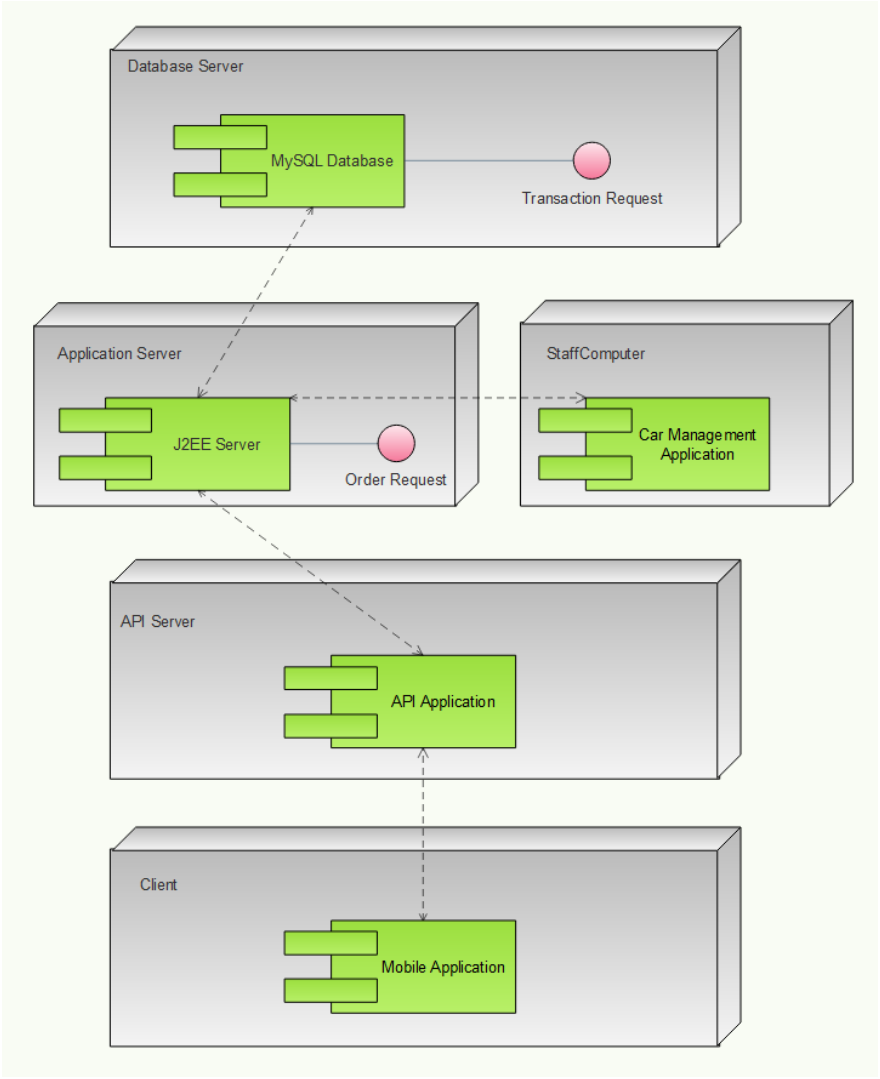
High level components



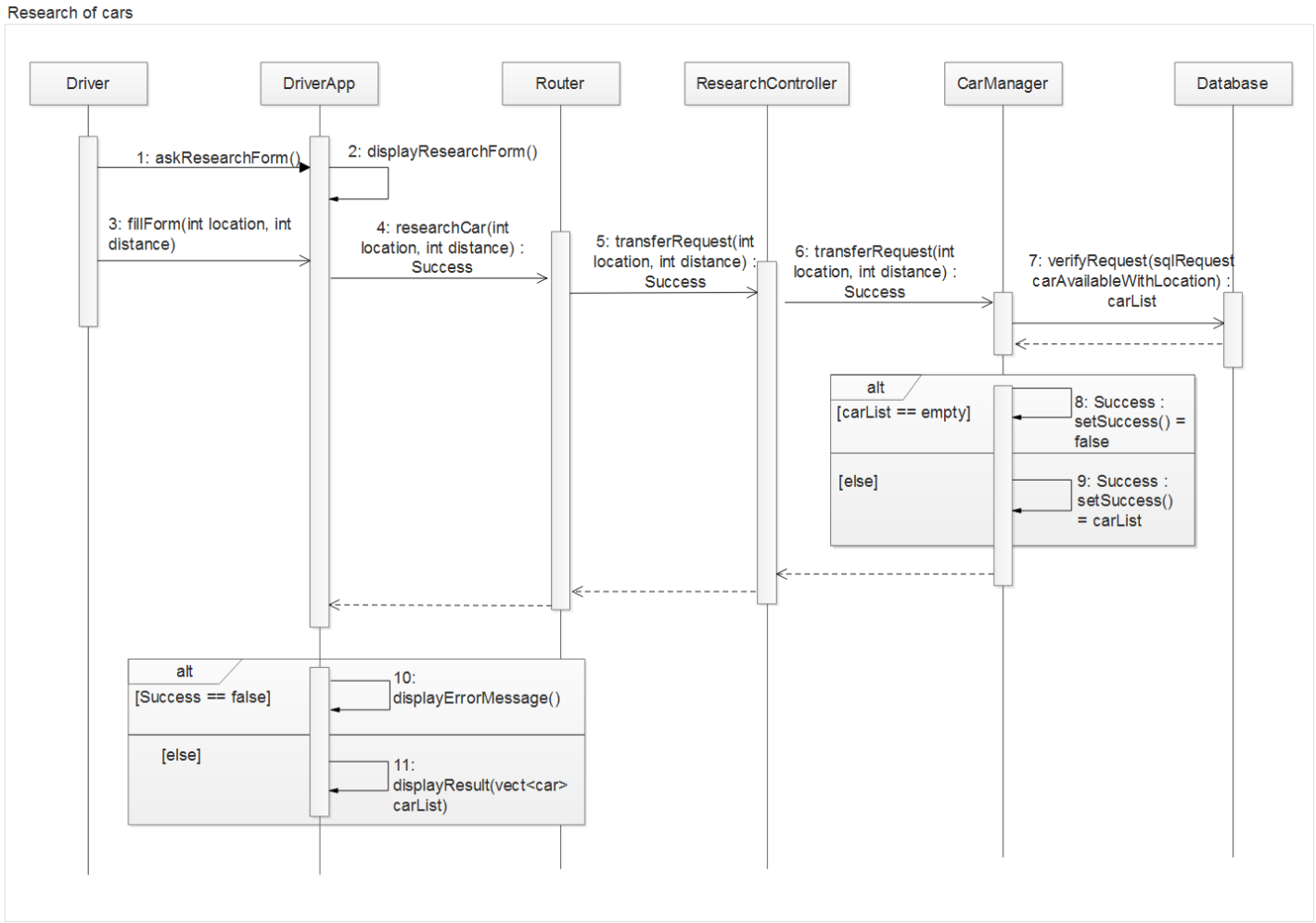
Component View



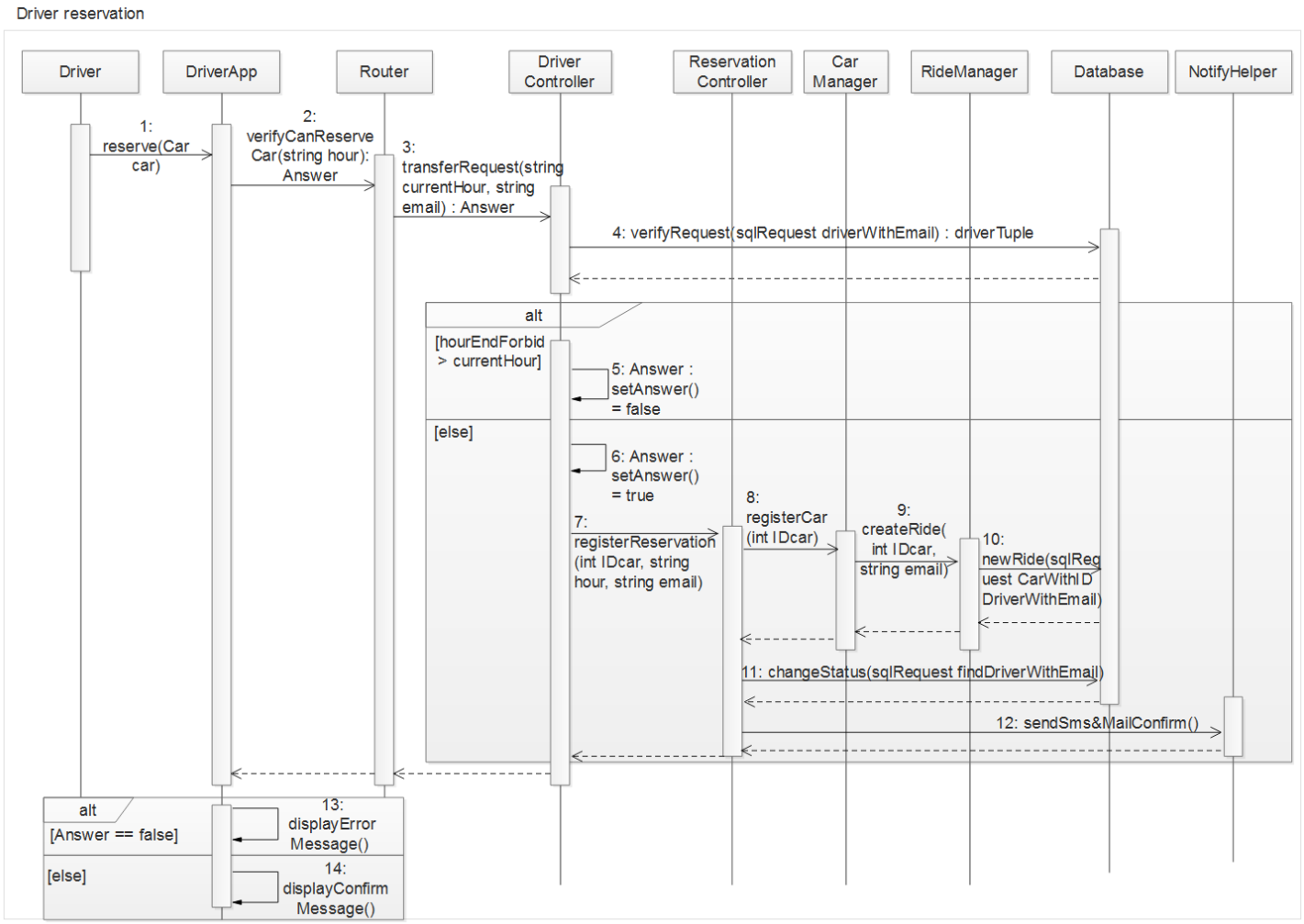
Deployment Diagram



Sequence Diagram (ex 1)



Sequence Diagram (ex 2)



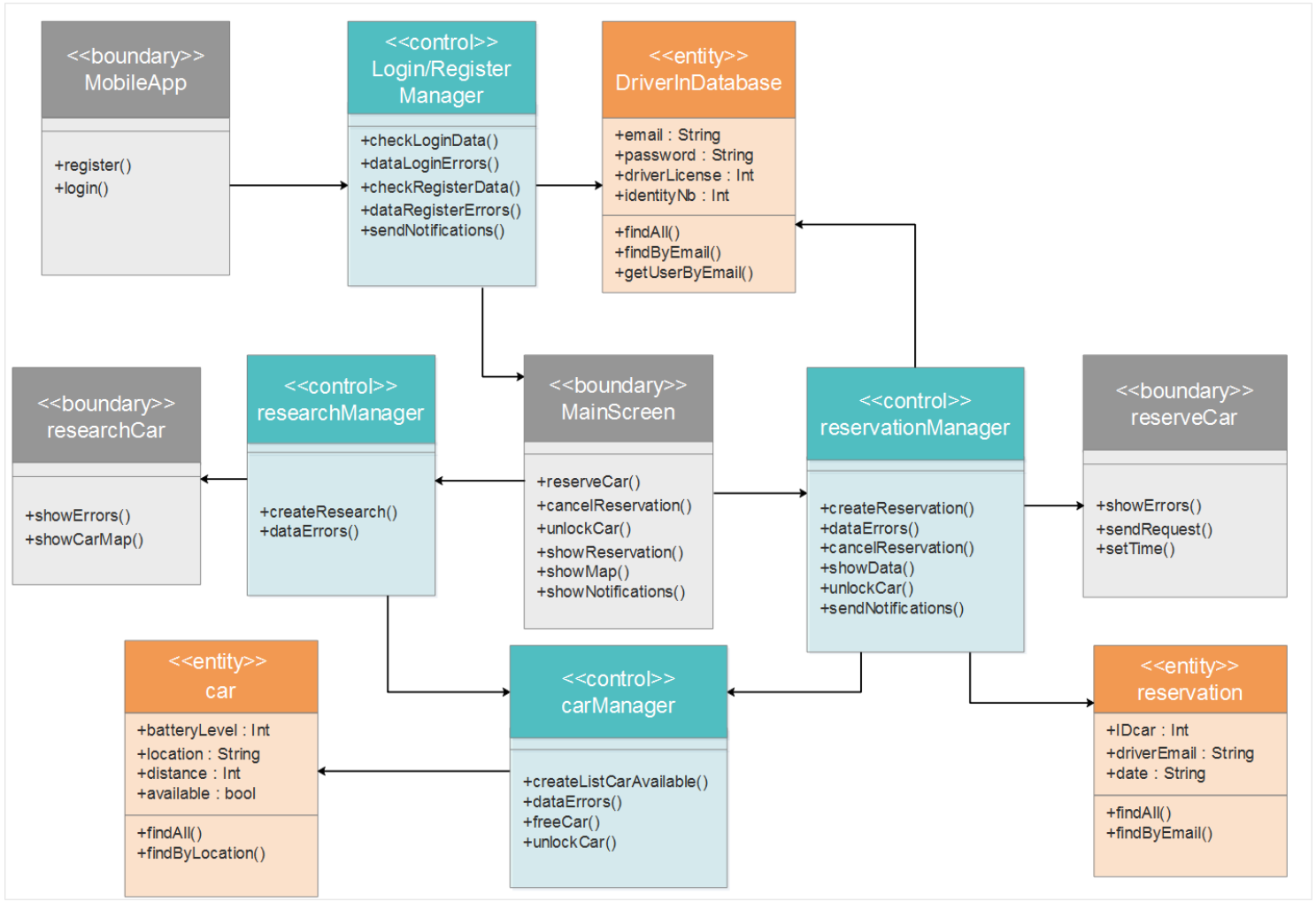
Car Reservation Algo.

```
Reservation(Driver, Car)
Pre-condition: Car->status == AVAILABLE, Driver->status == FREE
{
      As  transaction process
      (do everything single action or none,
      return True if they are done else False ):
      Process =
      {
            //Map of association between User and Car
            //It work in both direction and each association have a unique ID
            driver-Car-BidirectionalMap.add(Driver,Car)
            Driver->status = WAITTOPICKUP
            Car->status = UNAVAILABLE
      }
      if Process
            //Get the id of the association
            reservationId = driver-Car-BidirectionalMap.getId(Driver,Car)
            //Send a message on Driver's phone to confirm the reservation
            Driver.sendConfirmedReservation()
            //Add a one hour timer to the link between the driver and the car
            ReservationId.oneHourClock.start()
      else
            //Send a message on Driver's phone to signal an error
            Driver.sendErrorReservation()
}
```

Car Unlocking Algo.

```
Open(User)
Assumption: driver-Car-BidirectionalMap is global
{
	userPosition = User.getPosition();
	car = driver-Car-BidirectionalMap.getCar(User);
	carPosition = car.getPosition();
	if distance(userPosition,carPosition) < 1
	{
		//Send a signal to the car to open it
		//We suppose it won't fail
		car.sendOpenSignal();
		//wait return True if the Driver open the car before the timeout
		ack = wait(car.satus == OPENED, car.timeoutOpening)
		if ack == False
		{
			//Send a message to the user
			//asking him to redo the open procedure
			//and open the car before the timeout
			user.sendTooLateTryAgain()
		}
	}
	else
	{
		//Send a message: he is too far form the car
		user.sendToFarMessage();
	}
}
```

BCE Diagram


<<boundary>>
MobileApp
+register()
+login()
<<control>>
Login/Register
Manager
+checkLoginData()
+dataLoginErrors()
+checkRegisterData()
+dataRegisterErrors()
+sendNotifications()
<<entity>>
DriverInDatabase
+email : String
+password : String
+driverLicense : Int
+identityNb : Int
+findAll()
+findByEmail()
+getUserByEmail()
<<boundary>>
researchCar
+showErrors()
+showCarMap()
<<control>>
researchManager
+createResearch()
+dataErrors()
<<boundary>>
MainScreen
+reserveCar()
+cancelReservation()
+unlockCar()
+showReservation()
+showMap()
+showNotifications()
<<control>>
reservationManager
+createReservation()
+dataErrors()
+cancelReservation()
+showData()
+unlockCar()
+sendNotifications()
<<boundary>>
reserveCar
+showErrors()
+sendRequest()
+setTime()
<<entity>>
car
+batteryLevel : Int
+location : String
+distance : Int
+available : bool
+findAll()
+findByLocation()
<<control>>
carManager
+createListCarAvailable()
+dataErrors()
+freeCar()
+unlockCar()
<<entity>>
reservation
+IDcar : Int
+driverEmail : String
+date : String
+findAll()
+findByEmail()

Requirements Traceability

- G1: Allow driver to register the system by providing credential and payment details
- G2: Allow driver to log into the system with provided password
- G3: Driver can locate available cars within a certain distance around him or an address
- G4: Driver can reserve a car for up to 1 hour before pick it up
- G5: A reserved car but not picked up within one hour generate 1€ fee for the driver and forbidden him to reserve another one for few hours
- G6: Driver can cancel his reservation within the hour after he reserves it.
- G7: A driver close enough to a car reserved by him must be able to open it
- G8: The system starts charging once engine ignite
- G9: The set of safe area is predefined by the management system
- G10: The system stop charging once he leaves the car parked in a safe area
- G11: 10% discount on last ride if the car detects at least two passengers
- G12: The system apply 20% discount on the last ride if the car is left with more than 50% battery (over full battery)
- G13: If the driver parks the car in a power grid station, where the car can be charged, and takes care to plug the car. The system applies 30% discount on the last ride.
- G14: If the car is left with less than 20% battery or at more than 3km from the nearest power grid station, the system charges 30% more on the last ride.
- G15: The driver can enable the saving money option and so by giving his final destination to the system, this one is able to give him a station as destination and if the driver leaves the car and plugs it at this place he will get a special discount.

Thank you