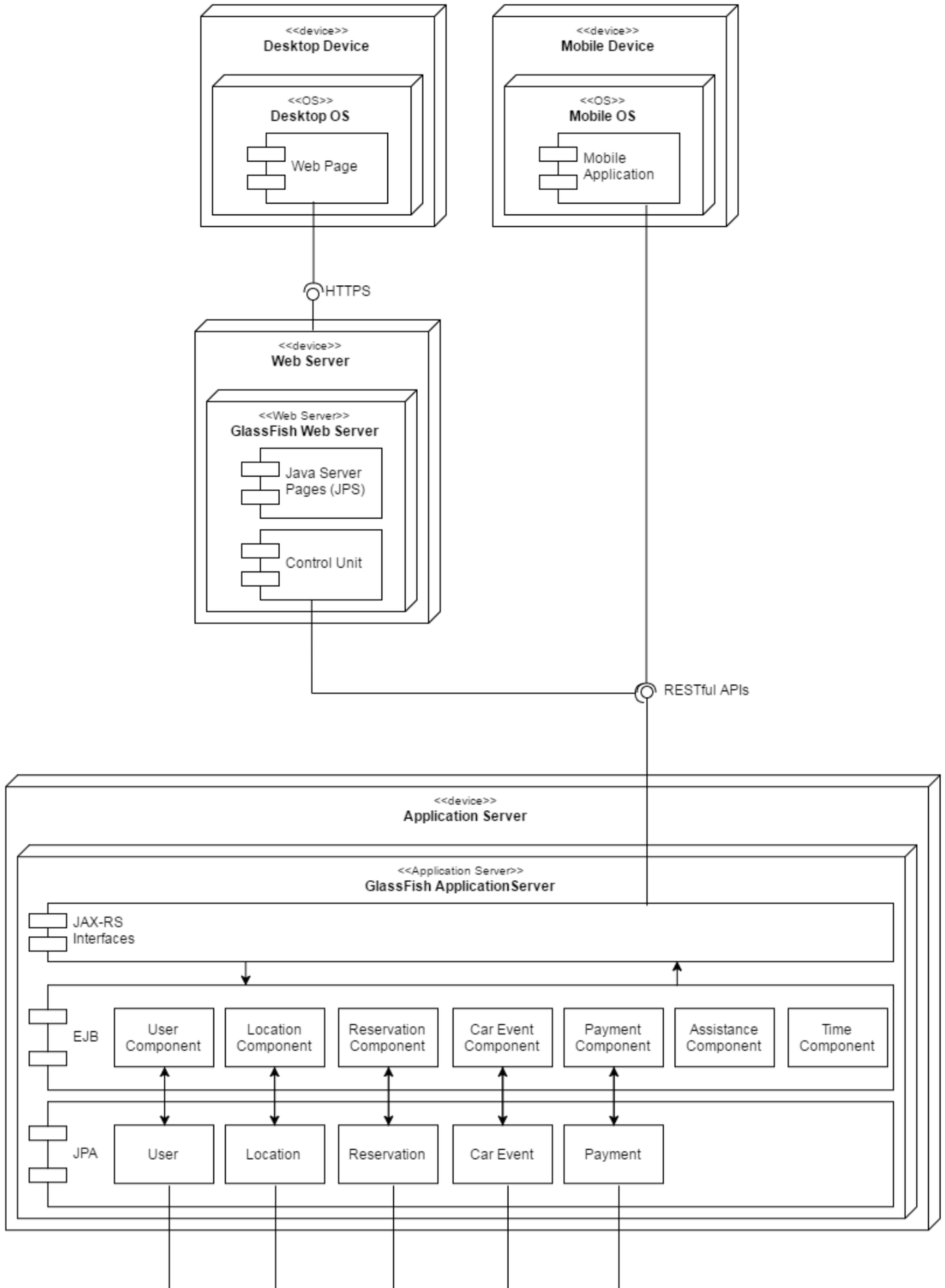
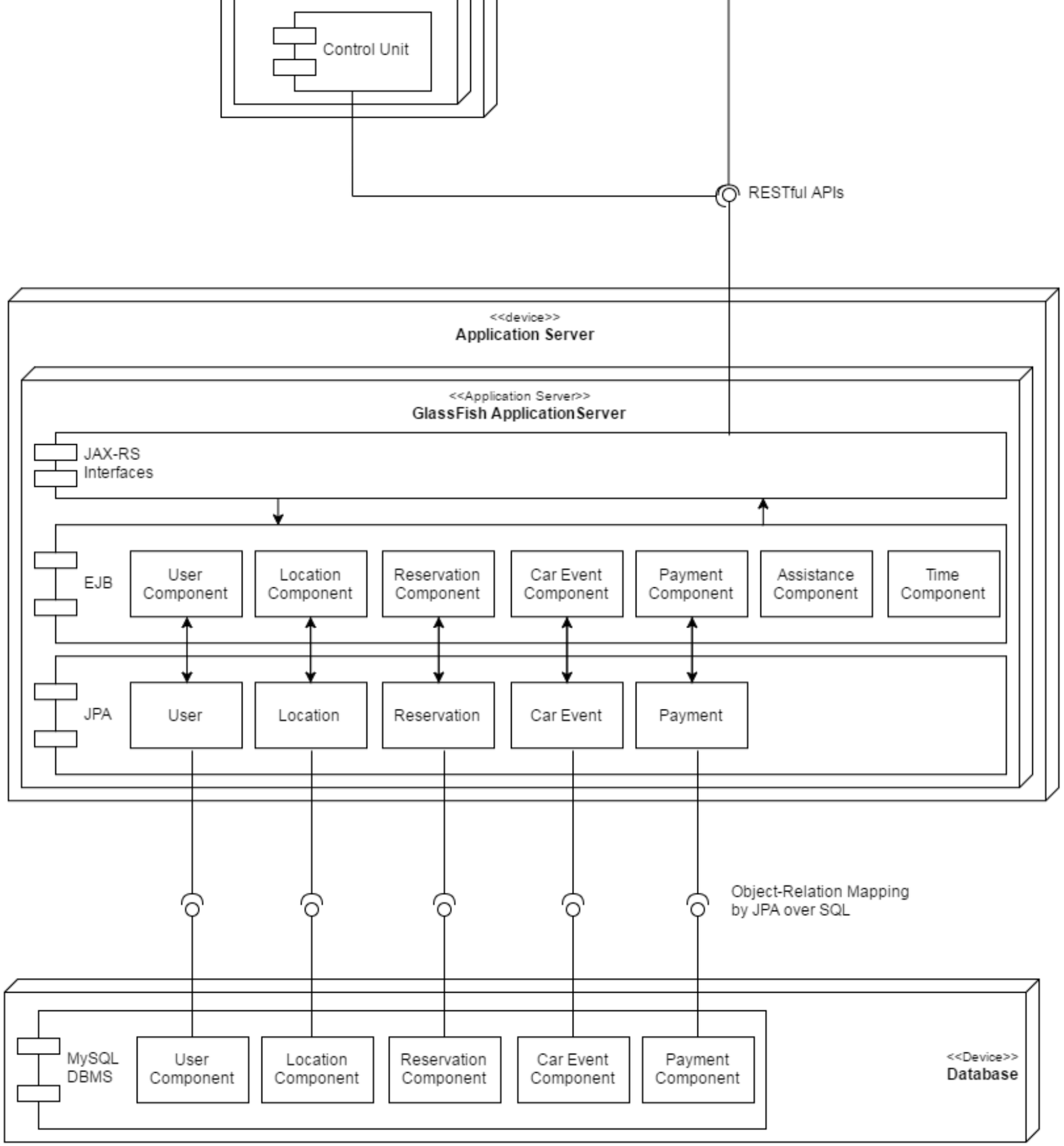
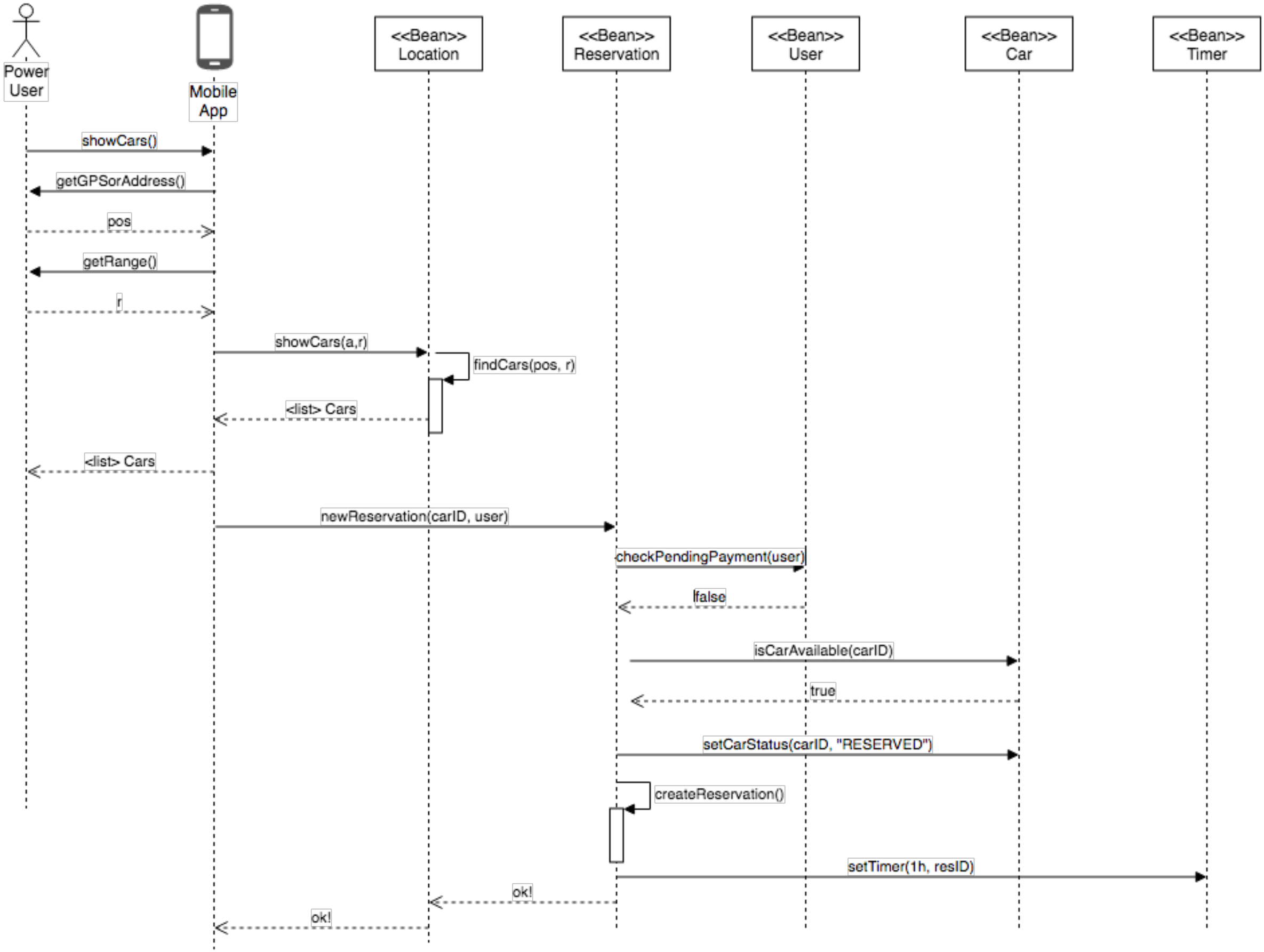
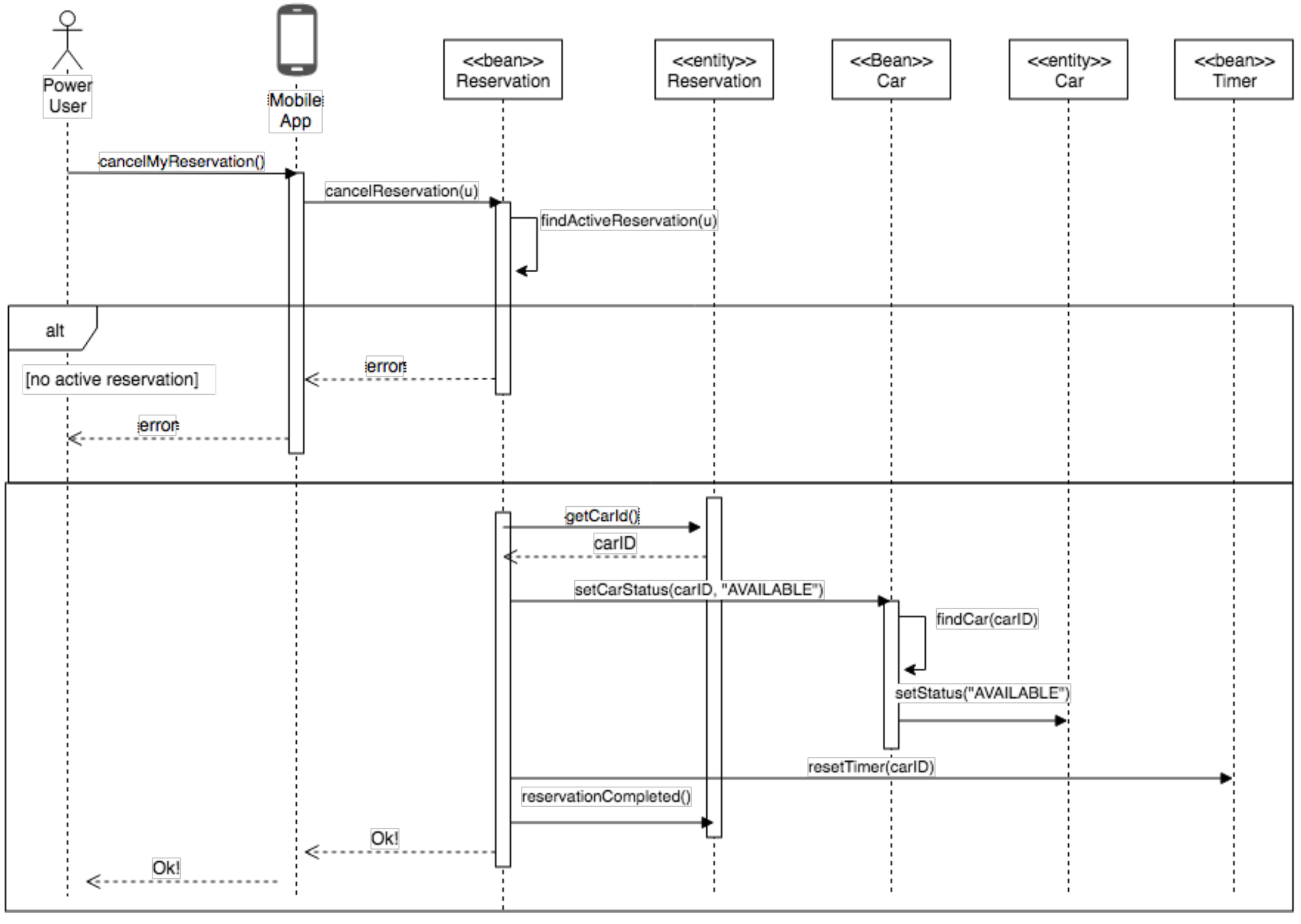

<<device>>
Desktop Device
<<OS>>
Desktop OS
Web Page
<<device>>
Mobile Device
<<OS>>
Mobile OS
Mobile Application
HTTPS
<<device>>
Web Server
<<Web Server>>
GlassFish Web Server
Java Server Pages (JPS)
Control Unit
RESTful APIs
<<device>>
Application Server
<<Application Server>>
GlassFish ApplicationServer
JAX-RS Interfaces
EJB
User Component
Location Component
Reservation Component
Car Event Component
Payment Component
Assistance Component
Time Component
JPA
User
Location
Reservation
Car Event
Payment

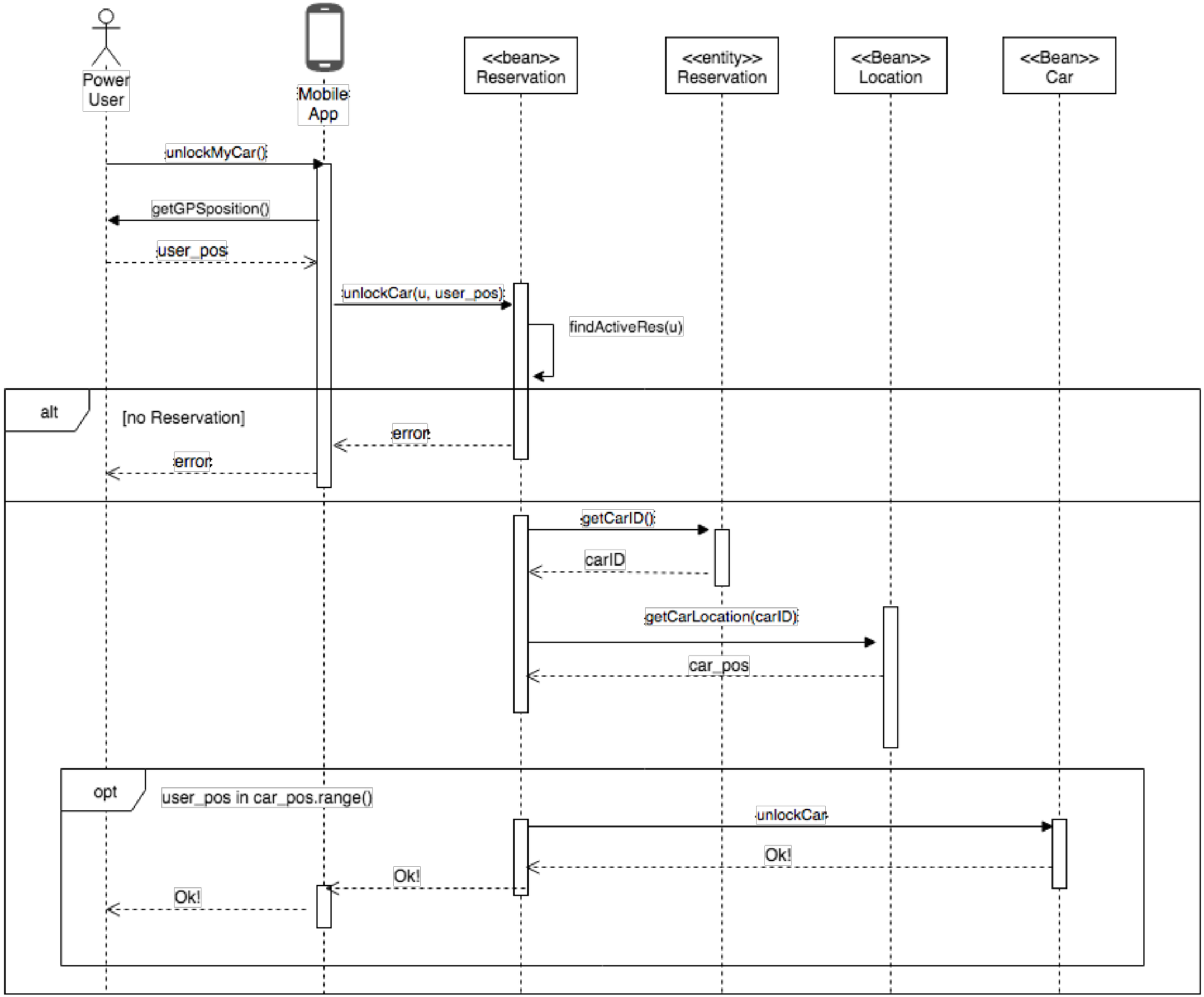

Control Unit
RESTful APIs
<<device>>
Application Server
<<Application Server>>
GlassFish ApplicationServer
JAX-RS
Interfaces
EJB
User Component
Location Component
Reservation Component
Car Event Component
Payment Component
Assistance Component
Time Component
JPA
User
Location
Reservation
Car Event
Payment
Object-Relation Mapping
by JPA over SQL
MySQL
DBMS
User Component
Location Component
Reservation Component
Car Event Component
Payment Component
<<Device>>
Database

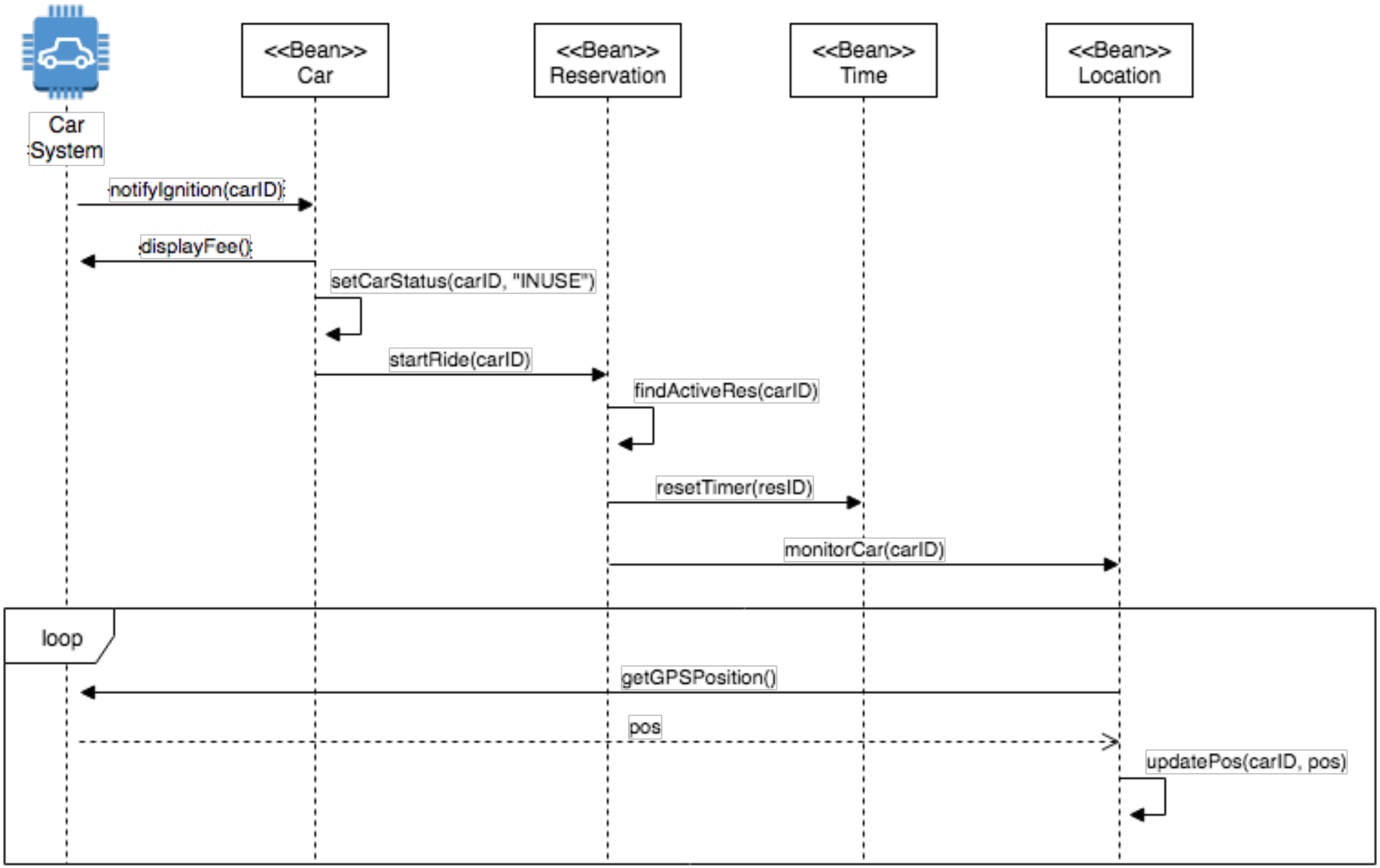

Power User
Mobile App
<<Bean>> Location
<<Bean>> Reservation
<<Bean>> User
<<Bean>> Car
<<Bean>> Timer
showCars()
getGPSorAddress()
pos
getRange()
r
showCars(a,r)
findCars(pos, r)
<list> Cars
<list> Cars
newReservation(carID, user)
checkPendingPayment(user)
false
isCarAvailable(carID)
true
setCarStatus(carID, "RESERVED")
createReservation()
setTimer(1h, resID)
ok!
ok!

Cancel Reservation

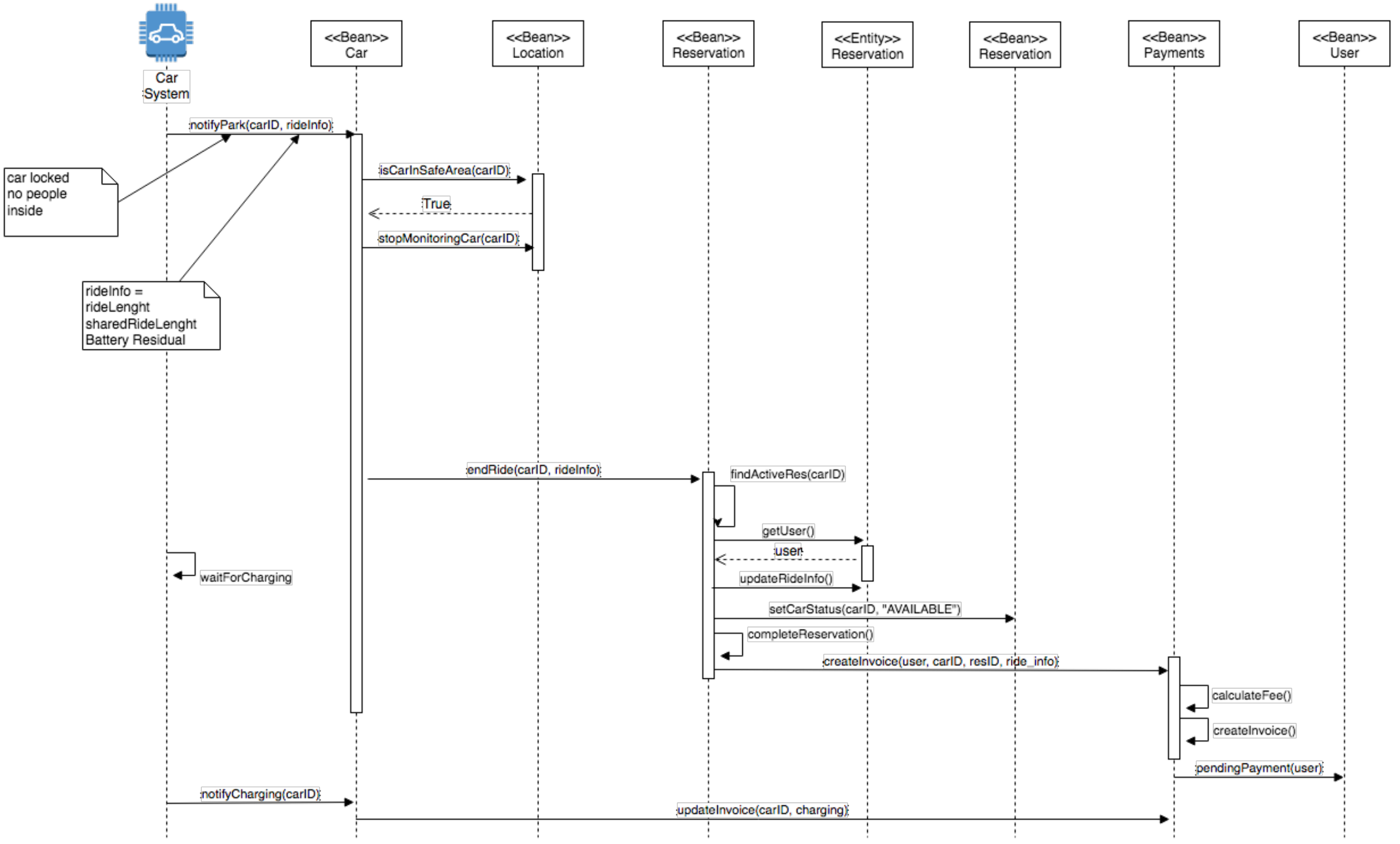


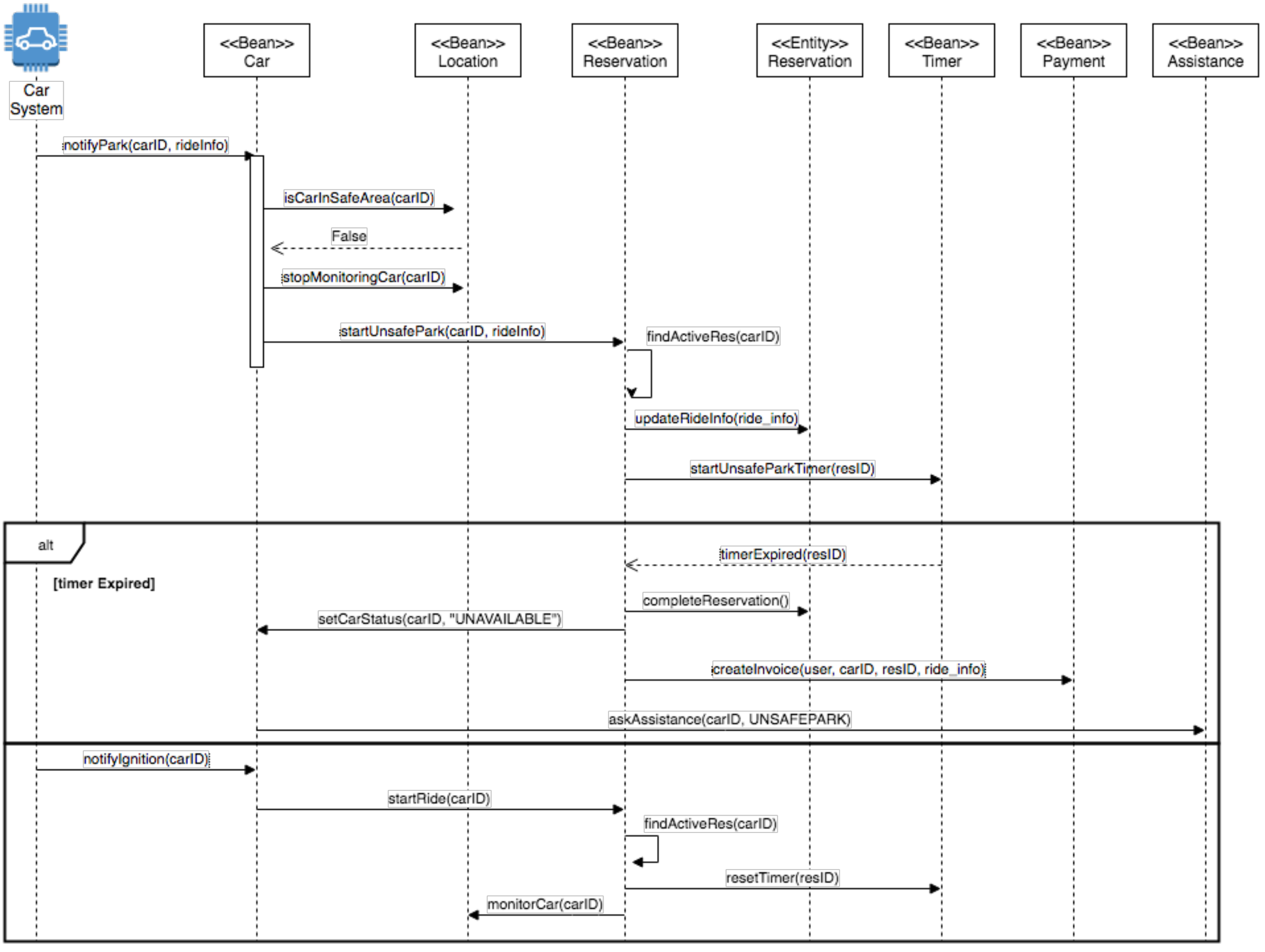
Car Unlock


Power User
:Mobile App
<<bean>> Reservation
<<entity>> Reservation
<<Bean>> Location
<<Bean>> Car
unlockMyCar()
getGPSposition()
user_pos
unlockCar(u, user_pos)
findActiveRes(u)
alt
[no Reservation]
error
error
getCarID()
carID
getCarLocation(carID)
car_pos
opt
user_pos in car_pos.range()
unlockCar
Ok!
Ok!
Ok!


Car System
<<Bean>> Car
<<Bean>> Reservation
<<Bean>> Time
<<Bean>> Location
notifyIgnition(carID)
displayFee()
setCarStatus(carID, "INUSE")
startRide(carID)
findActiveRes(carID)
resetTimer(resID)
monitorCar(carID)
loop
getGPSPosition()
pos
updatePos(carID, pos)

End Ride - Park in Safe Area




Car System
<<Bean>> Car
<<Bean>> Location
<<Bean>> Reservation
<<Entity>> Reservation
<<Bean>> Timer
<<Bean>> Payment
<<Bean>> Assistance
notifyPark(carID, rideInfo)
isCarInSafeArea(carID)
False
stopMonitoringCar(carID)
startUnsafePark(carID, rideInfo)
findActiveRes(carID)
updateRideInfo(ride_info)
startUnsafeParkTimer(resID)
alt
[timer Expired]
timerExpired(resID)
completeReservation()
setCarStatus(carID, "UNAVAILABLE")
createInvoice(user, carID, resID, ride_info)
askAssistance(carID, UNSAFEPARK)
notifyIgnition(carID)
startRide(carID)
findActiveRes(carID)
resetTimer(resID)
monitorCar(carID)