PROJECT REQUIREMENTS DOCUMENT

Team : Group 6

Stakeholders: Manish Mukund Kaujalgi, Vaishnavi Bhadresh

Project Description:

A Car request application support a variety of services involving request car services like renting, ride share, deliver services, car messenger services and buying/selling cars including unique cars.

We are implementing features such as Rent a Car and Mechanic Services

Features Provided are as Follows:

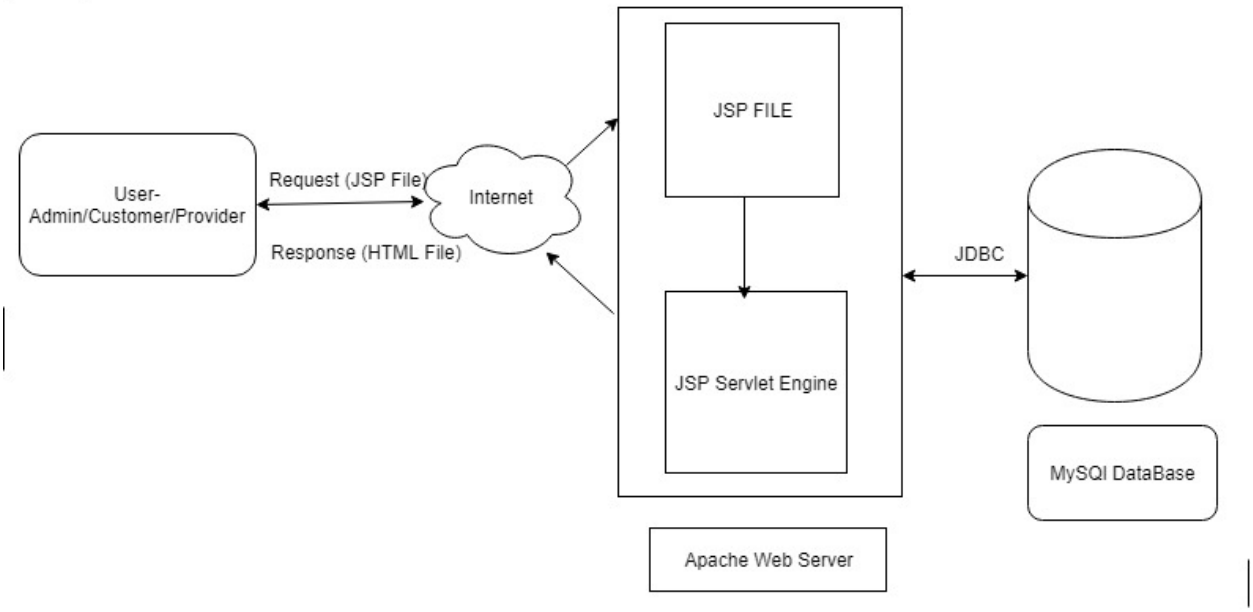
1. **User Management Module (Security):** This module deals with access control to the application
   - ✓ Type of users – Consumers, Providers and Admin
   - ✓ Login using User name and Password. Additionally, an email validation is added to provide security to the user by sending a OTP to the registered mail id.
2. **Customer Management Module:**
   - ✓ As a Customer: Providing service to Rent a Car through a module such as rent a car, book and pay
   - ✓ As a Provider: Providing service to add a car details, add a mechanic services
3. **Billing Module:**
   - ✓ Payment module for both customer and service provider

Technical Requirements

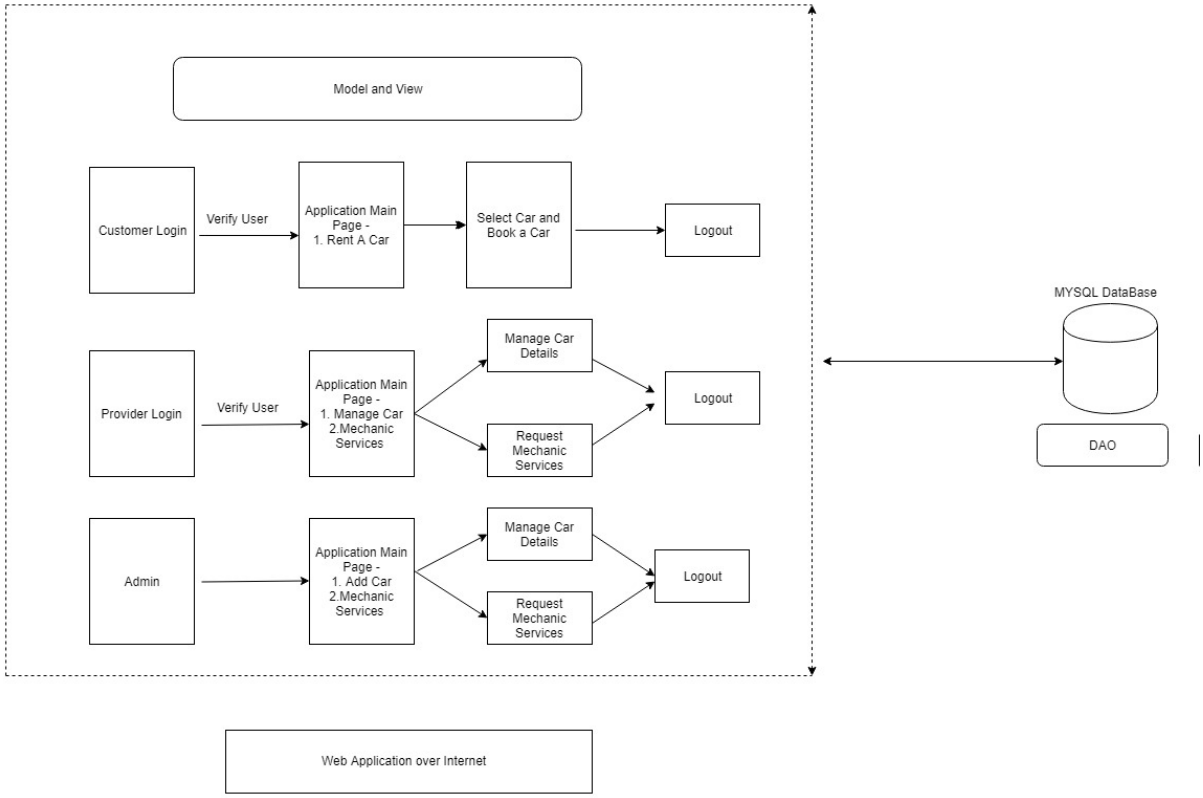
Req Id	Tools / Technologies	Priority
1	Eclipse	High
2	Tomcat Server	High
3	MySql Database	High
4	Java, JSP, Servlets	High
5	SQL Workbench	Medium

Architecture:

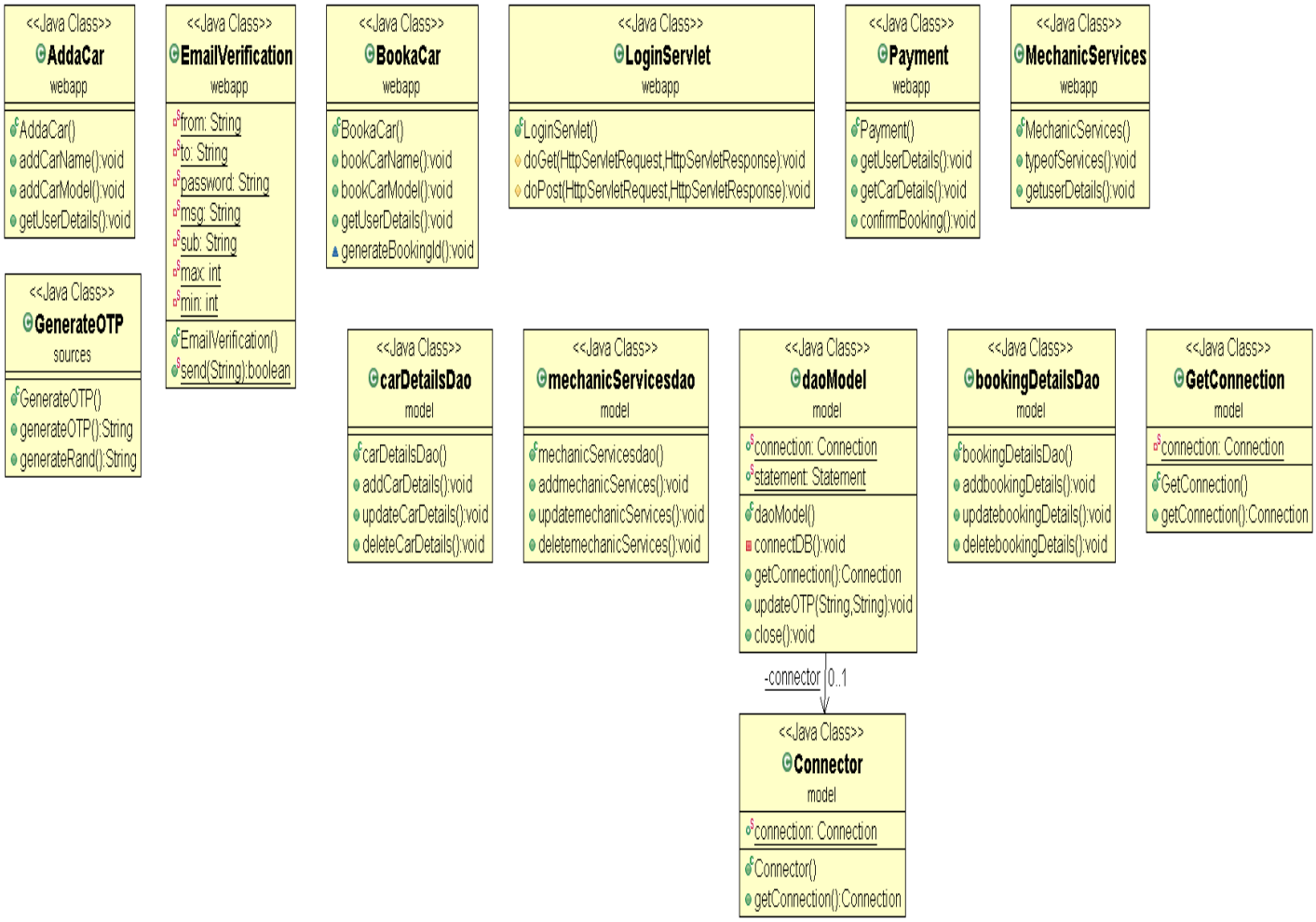
1.1 High level architecture



1.2 Low level architecture



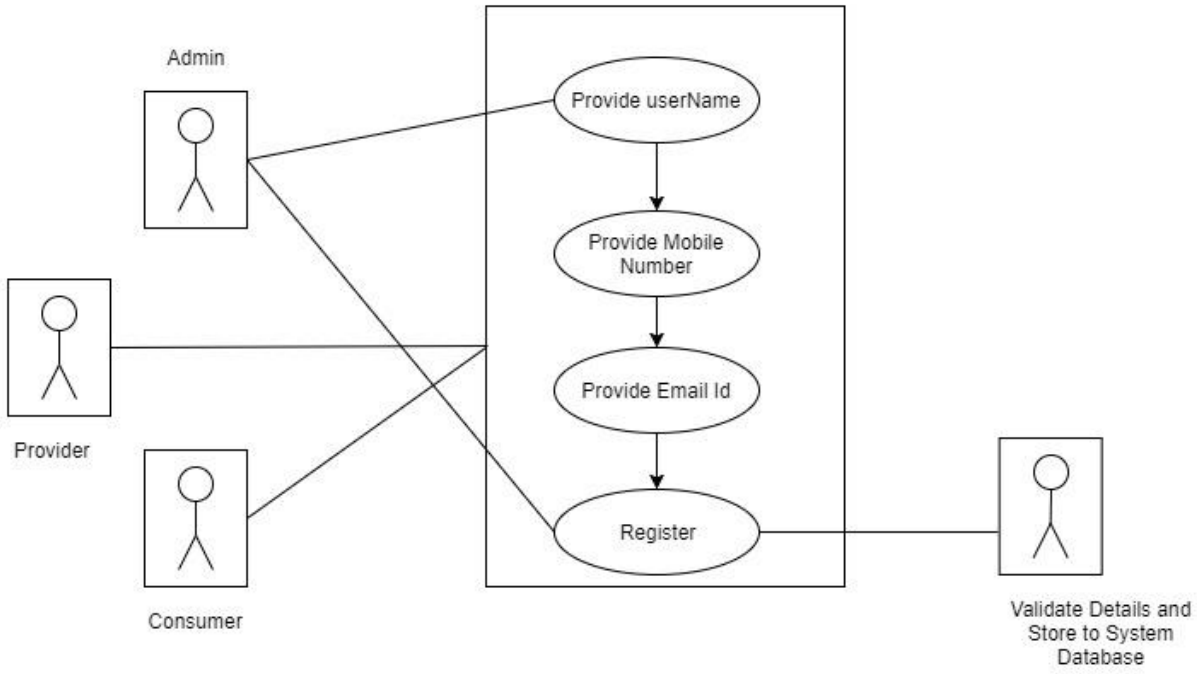
UML Diagram:


<<Java Class>>
AddaCar
webapp
AddaCar()
addCarName():void
addCarModel():void
getUserDetails():void
<<Java Class>>
EmailVerification
webapp
from: String
to: String
password: String
msg: String
sub: String
max: int
min: int
EmailVerification()
send(String):boolean
<<Java Class>>
BookaCar
webapp
BookaCar()
bookCarName():void
bookCarModel():void
getUserDetails():void
generateBookingId():void
<<Java Class>>
LoginServlet
webapp
LoginServlet()
doGet(HttpServletRequest,HttpServletResponse):void
doPost(HttpServletRequest,HttpServletResponse):void
<<Java Class>>
Payment
webapp
Payment()
getUserDetails():void
getCarDetails():void
confirmBooking():void
<<Java Class>>
MechanicServices
webapp
MechanicServices()
typeofServices():void
getuserDetails():void
<<Java Class>>
GenerateOTP
sources
GenerateOTP()
generateOTP():String
generateRand():String
<<Java Class>>
carDetailsDao
model
carDetailsDao()
addCarDetails():void
updateCarDetails():void
deleteCarDetails():void
<<Java Class>>
mechanicServicesdao
model
mechanicServicesdao()
addmechanicServices():void
updatemechanicServices():void
deletemechanicServices():void
<<Java Class>>
daoModel
model
connection: Connection
statement: Statement
daoModel()
connectDB():void
getConnection():Connection
updateOTP(String,String):void
close():void
-connector
0..1
<<Java Class>>
Connector
model
connection: Connection
Connector()
getConnection():Connection
<<Java Class>>
bookingDetailsDao
model
bookingDetailsDao()
addbookingDetails():void
updatebookingDetails():void
deletebookingDetails():void
<<Java Class>>
GetConnection
model
connection: Connection
GetConnection()
getConnection():Connection

Use case Diagrams:

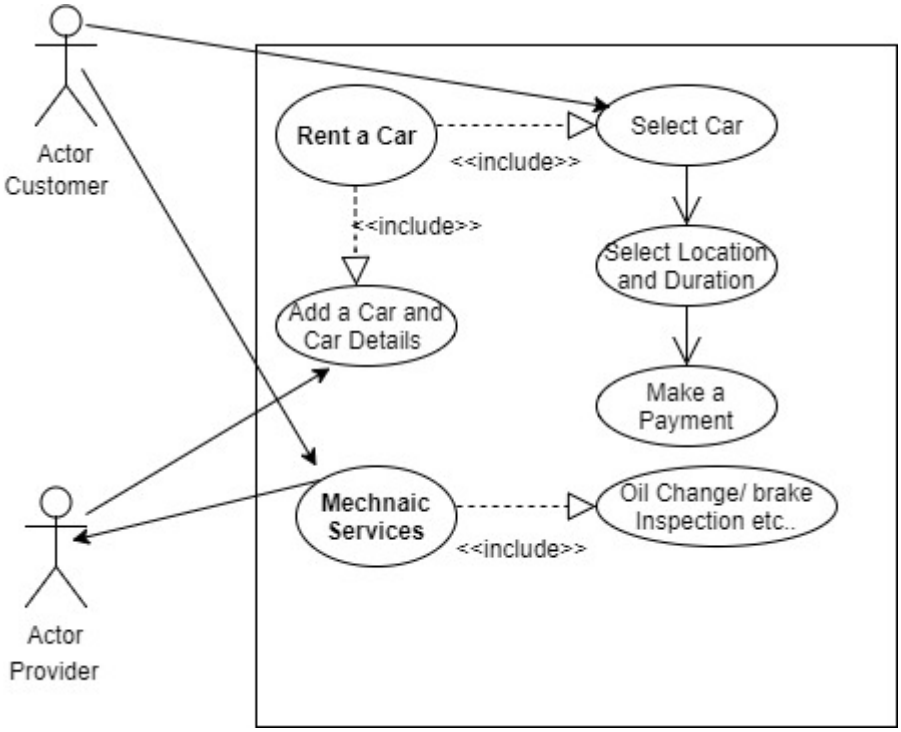
1.1 Login

—



—

1.2 Other functionalities:


Actor
Customer
Rent a Car
Select Car
<<include>>
<<include>>
Select Location and Duration
Add a Car and Car Details
Make a Payment
Mechnaic Services
Oil Change/ brake Inspection etc..
<<include>>
Actor
Provider

ER Diagram:

paymentDetails
cardNumber INT
cardExpiry DATE
cardName VARCHAR(45)
cardType VARCHAR(45)
issueBank VARCHAR(45)
Indexes
mechanicServices
oilchange INT
brake_repair VARCHAR(45)
gereral_service VARCHAR(45)
battery_service VARCHAR(45)
tyre_service VARCHAR(45)
electrical_service VARCHAR(45)
emission_test VARCHAR(45)
engine_service VARCHAR(45)
Indexes
verification
onetimepassword INT
email_id VARCHAR(45)
Indexes
car_details
model INT
name VARCHAR(45)
location VARCHAR(45)
seatingcapacity INT
colour VARCHAR(45)
driverName VARCHAR(45)
driverNumber INT
car_id VARCHAR(45)
Indexes
login
username INT
password VARCHAR(45)
email_id VARCHAR(45)
id VARCHAR(45)
userType VARCHAR(45)
Indexes
booking_details
location VARCHAR(45)
bookingTime VARCHAR(45)
carName INT
carModel VARCHAR(45)
bookingDate DATE
customerName VARCHAR(45)
startTime DATETIME
endTime DATETIME
duration INT
customerId VARCHAR(45)
booking_id VARCHAR(45)
Indexes