CARRENTAL PROJECT KICK OFF DOCUMENT

Project Name: CARRENTAL WEB APPLICATION PROJECT

Definition: Definition: The Car Rental application is a web application which is able to be rent a car directly by using a web browser in your computer, phone, tablet, or mobile device. There will be different types of user in the CarRental application. While unregistered users are able to see available cars' information, these users are able to register and login into application. The users who have Customer Role are able to make a reservation for a car and manage their own user information. In addition to these. administrator of the CarRental application can manage the users, cars, reservations and create some reports.
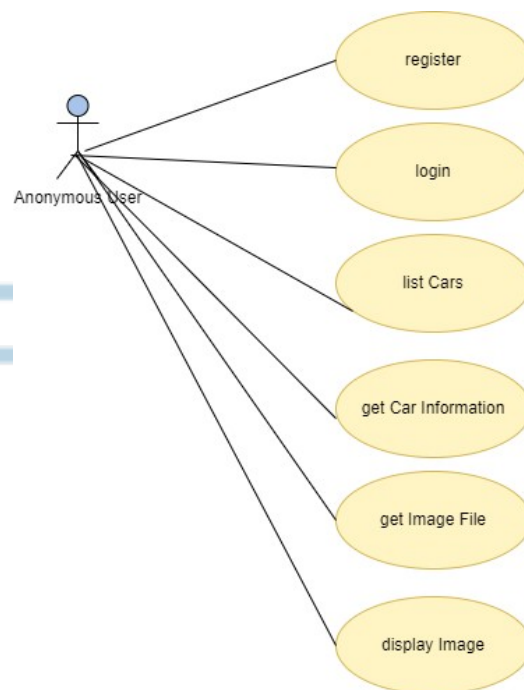
CARRENTAL WEB APPLICATION CUSTOMER REQUIREMENTS

1. CarRental Application must be a web application.

2. The application must have user friendly UI. (Non functional requirement)

3. The application must be secure. (Non functional requirement)

4. Language of the application must be English. (Non functional requirement)

5. The application must display response of user request nearby for 5 seconds. (Non functional requirement)

6. Anonymous users who will make a process should be able to register this application with their own information

7. There should be able two types of roles for authorized user: Customer, Admin

8. Registered user is able to have at least one role or all roles

9. Anonymous user should be able to register to the application

10. Registered user should be able to login to the application after registration

11. Authenticated user should be able to get its own user information

12. Authenticated user should be able to update its own user information

13. Authenticated user should be able to update its own password

14. User who has the admin role should be able to get all users' information

15. User who has the admin role should be able to get a user information

16. User who has the admin role should be able to update a user information

17. User who has the admin role should be able to delete a user

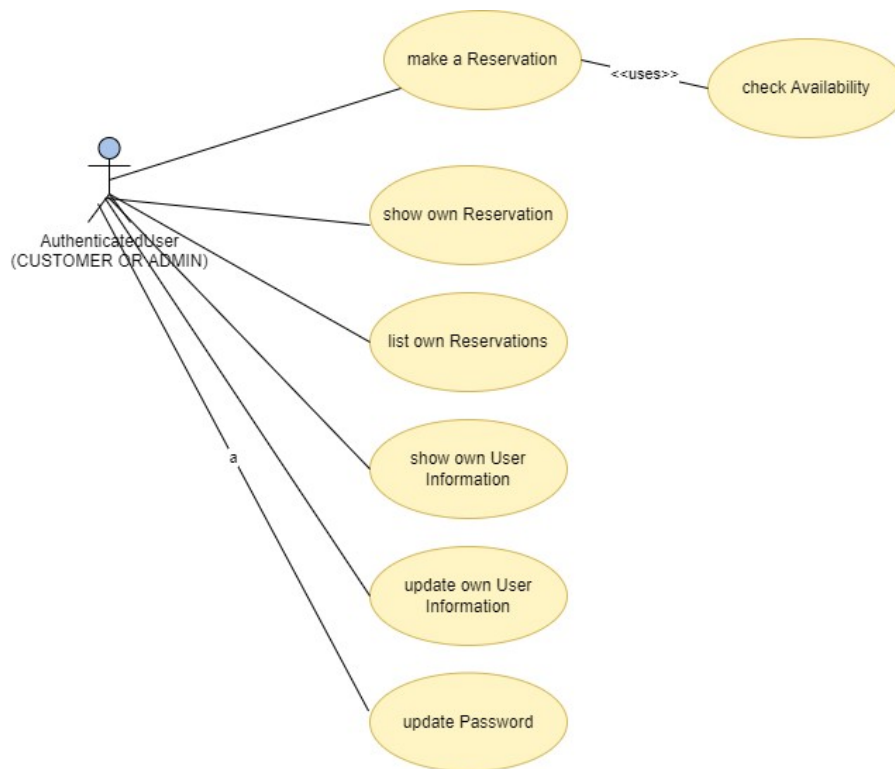18. Anonymous users should be able to get all cars' information

19. Anonymus users should be able to get a car's information

20. User who has the admin role should be able to add a car

21. User who has the admin role should be able to upload a car image

22. User who has the admin role should be able to get all cars' image

23. Anonymous user should be able to get a car's image

24. Anonymous user should be able to display a car's image

25. User who has the admin role should be able to update a car with an car image

26. User who has the admin role should be able to delete a car

27. User who has the admin role should be able to delete a car

28. Authenticated user should be able to make a reservation for a car

29. Authenticated user should be able to check availability of a car between selected datetime

30. User who has the admin role should be able to add a reservation for a car and a user

31. User who has the admin role should be able to get all reservations' information

32. User who has the admin role should be able to get a reservation information

33. User who has the admin role should be able to get a user's reservations' information

34. Authenticated user should be able to get its own reservation information

35. Authenticated user should be able to get its own reservations' information

36. User who has the admin role should be able to update a reservation

37. User who has the admin role should be able to delete a reservation

38. User who has the admin role should be able to get all users' information in an excel report

39. User who has the admin role should be able to get all cars' information in an excel report

40. User who has the admin role should be able to get all reservations' information in an excel report

What is Use Case Diagram: In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.
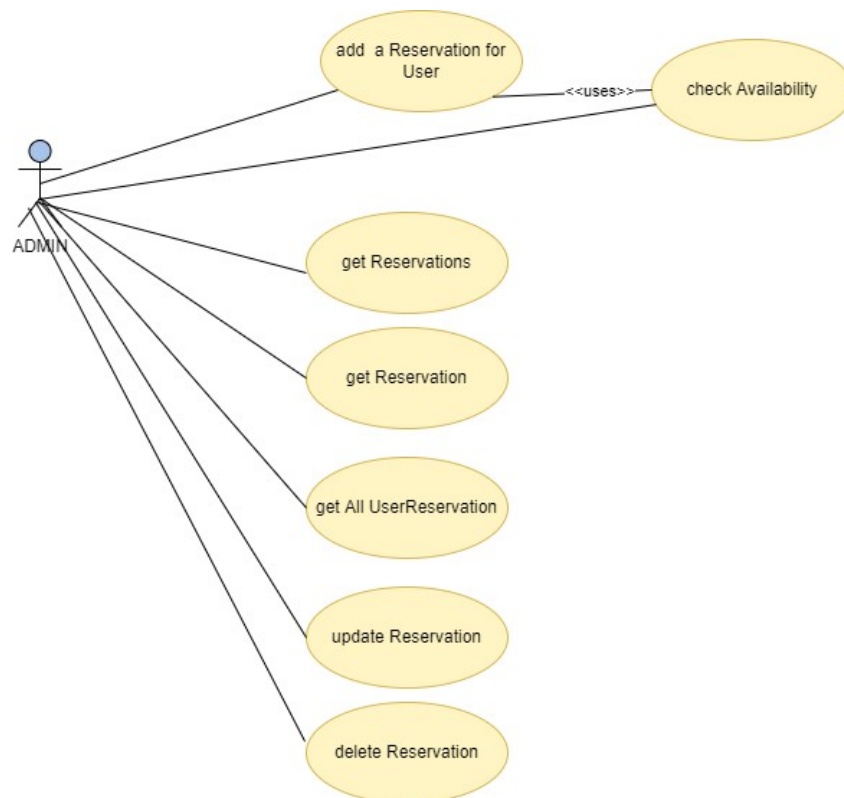
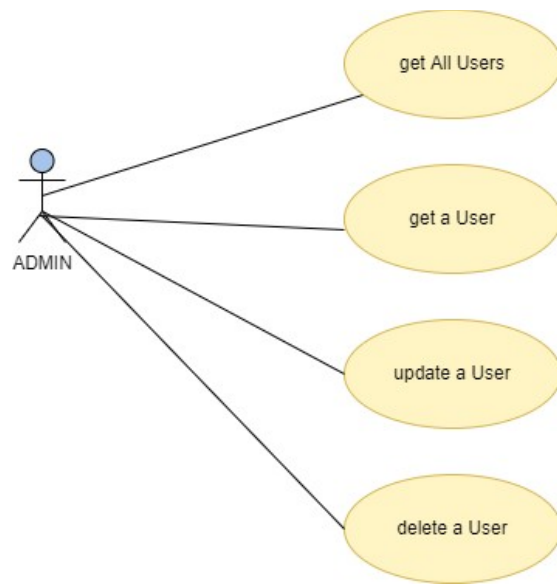You will see usecase diagrams about CarRental project below (Graphics 1, 2, 3, 4, 5, 6)
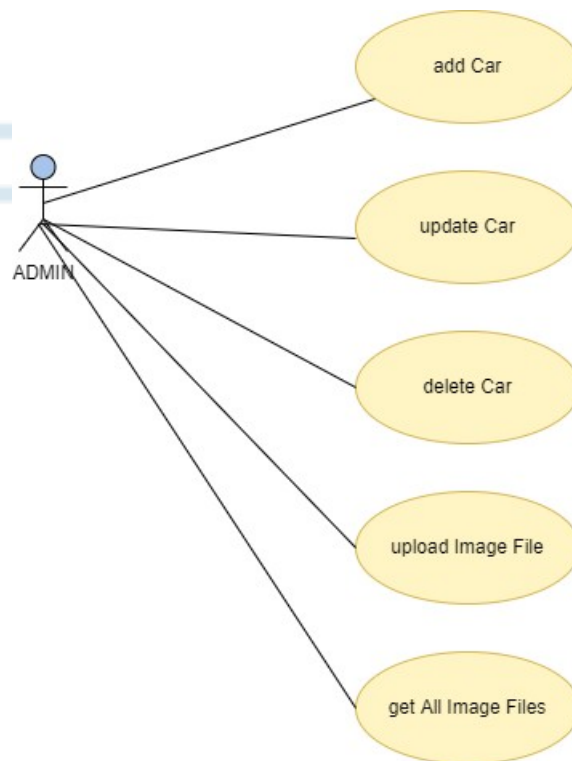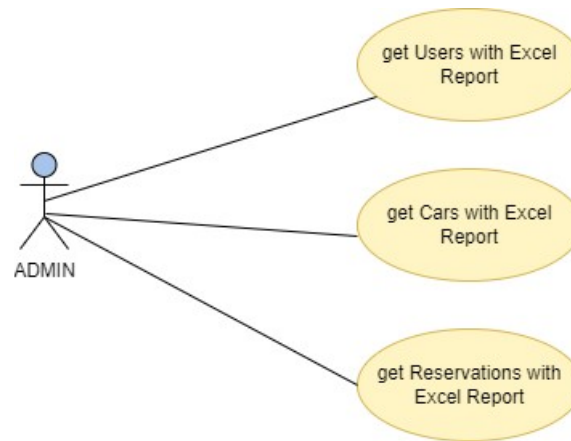

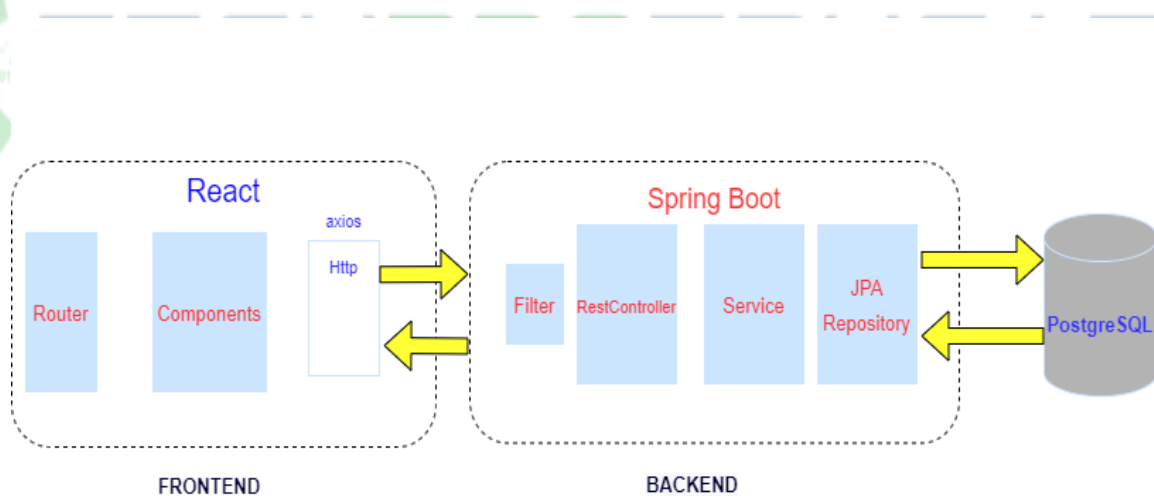*Graphic 1*

Graphic 2

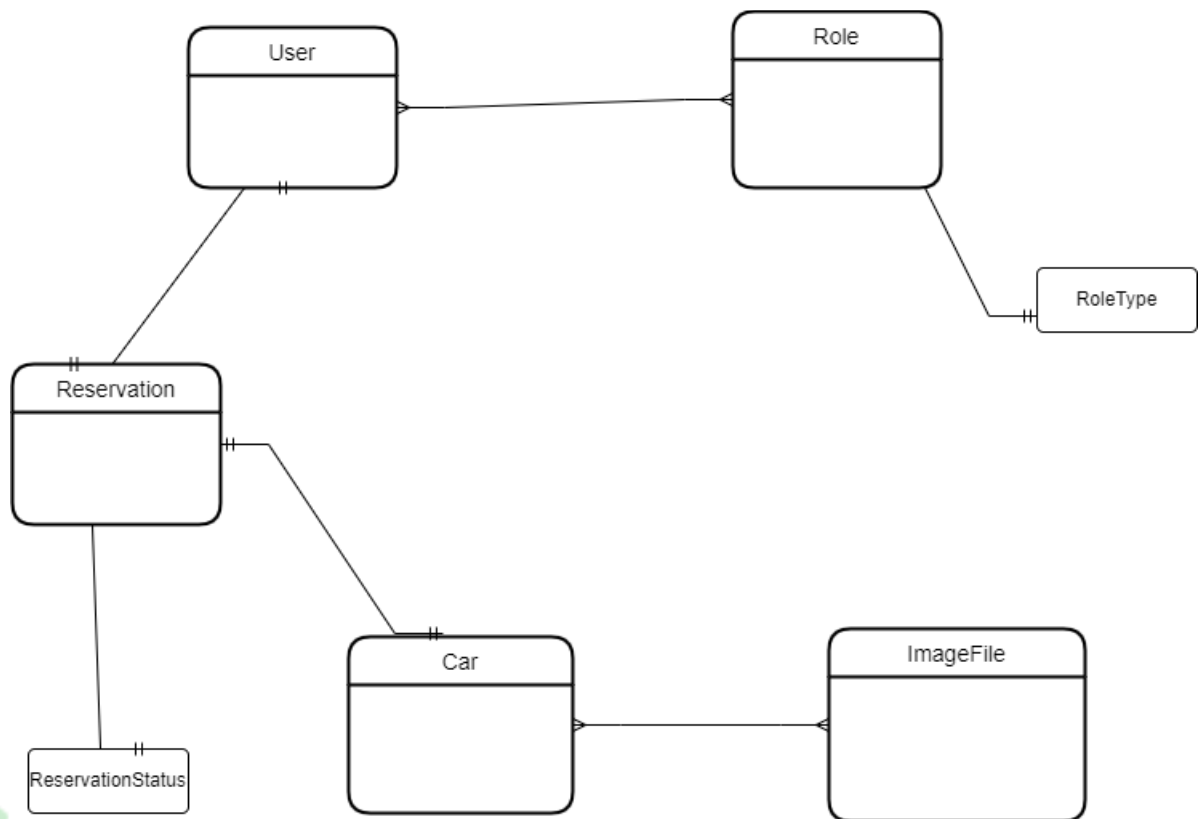*Graphic 3*

Graphic 4

Graphic 5

Graphic 6

Frontend: The part of a web site or web application that the user interacts with directly. In this project, React Library will be used for frontend.

Backend: It is the server site the wep application. It stores and arranges data. It is the part of the web app that you can't see and interact with. In this project, Sprint boot framework will be used as a backend tecknology. Moreover PostgreSQL will be used as relational database.

Graphic7 General system structure of the application

Graphic 8 Entities and Relations Between Entity in the Application.

Entities represent persistent data stored in a relational database automatically using container-managed persistence. The entities that will be created and used in the project is given above.

- One User can have many roles and One Role can belongs to Many User (Many To Many relation)

- One Reservation has a User (One To One relation)

- One Reservation has a Car (One To One relation)

- One Car can have many ImageFile and ImageFile can belongs to Many Car (Many To Many relation)

- Role has a RoleType (Enumeration)

- Reservation has a ReservationStatus (Enumeration)

DEVELOPMENT ENVIRONMENT, IDEs and TOOLS – INSTALLATIONS

1. FOR BACKEND

- Java 11 https://www.oracle.com/tr/java/technologies/javase/jdk11-archive-downloads.html
- Spring Tool Suite

  Go to https://spring.io/tools

  Download Spring Tools 4 for Eclipse depending on your OS.

  The file that you donwload is a jar file. Extract the file.

  You will see contents.zip and extract it as well. After extract contents.zip you will see sts-4.15.1.RELEASE

  Runnable File SpringToolSuite4.exe will be sts-4.15.1.RELEASE folder.
- Postman https://www.postman.com/
- PostgreSQL 12.11 https://www.enterprisedb.com/downloads/postgres-postgresql-downloads
  Guide for installation POSTGRESQL: https://www.postgresqltutorial.com/install-postgresql/

THE TOPICS THAT YOU SHOULD REVIEW BEFORE STARTING PROJECT

FOR BACKEND (SPRING BOOT PROJECT)
1. Core Java: OOPS, classes, enums, interfaces, exception handling, collections, stream (foreach, filter, map), lambda,optional key word and other fundamentals.
2. Logging (SLF4J, Logback)
3. Regular expressions
4. What is Spring Framework.
5. What is Spring Boot Framework
6. What is Spring Security Framework. (JWT Based Security)
7. What is inversion of control , dependency injection and Spring IOC Container
8. What is JPA, Hibernate and Spring Data JPA
9. What is entity class and how to create it
10. What are OneToOne, OneToMany, ManyToOne, ManyToMany relations on hibernate.
11. How to use @JoinTable, @JoinColum annotations.
12. JPQL (Java Persistence Query Language), Basic SQL knowledge
13. What is REST API
14. How , why to use @Bean, @Autowired, @RestController, @Service, @Repository annotations.
15. Jackson annotations (@JsonIgnore, @JsonFormat etc.)
16. Controller-Service-Repository layered structure in spring boot app
17. What is @Transactional annotation in org.springframework.transaction.annotation
18. What is the Data Transfer Object and how to use it.
19. Usage of @ResponseEntity, @ResponseBody, @Request Body, @RequestMapping, @PostMapping, @GetMapping, @DeleteMapping, @PutMapping. @Valid annotations.
20. HTTP Response Status Codes. (200, 201, 400, 404, etc.)
21. Why and How to use @PathVariable and @QueryParam annotations
22. Project Lombok

GOOD
LUCK
WITH THAT