

# Presentation Group 2

## Team Management



Owned by [Vu Nguyen](#) \*\*\*

Last updated: Jul 27, 2024 • 1 min read • See how many people viewed this page

Team member	Role
@Vu Nguyen	Product Owner & Scrum Master
@Ngoc Thinh Le	Developer
@Van Nhat Le	Developer
@AnhTuanHa	Developer

# Agenda

Introduction

Software development methodologies

Architecture

Backend

Frontend

Demo



# Software development methodologies

←

↺

🔒

https://vunguyen-miu.atlassian.net/jira/software/c/projects/CRS/pages

🔍

☆

🔴

⚙️

📄

🔖

📁

🔒

👤

⋮

🔗

PowerVaultMIUPersonalOrder details | eBayMIU\_Homework/src...imapedu.sharepoint...CARFAX Vehicle Hist...crisnguyenvn/crs-se...Find your trip - Find...

Atlassian uses cookies to improve your browsing experience, perform analytics and research, and conduct advertising. Accept all cookies to indicate that you agree to our use of cookies on your device. [Atlassian cookies and Tracking notice](#)

PreferencesOnly necessaryAccept all

Open in Confluence

✕

Problem Statement & Requirement

Owned by Vu Nguyen

Jul 26, 2024 • 1 min read • 📄 See how many people viewed this page

Problem Statement

Develop a Car Rental System to streamline the process of renting vehicles. The system should allow customers to view available cars, make reservations, and manage their rentals. Additionally, the system should enable the rental company to manage vehicle inventory, track reservations, and process payments efficiently.

Requirements

Functional Requirements

1. User Registration and Login:

◦ Users can create accounts and log in.

◦ User roles include customers and administrators.

2. Car Inventory Management:

◦ Administrators can add, update, and remove cars from the inventory.

◦ View car details such as make, model, year, and availability status.

3. Search and View Cars:

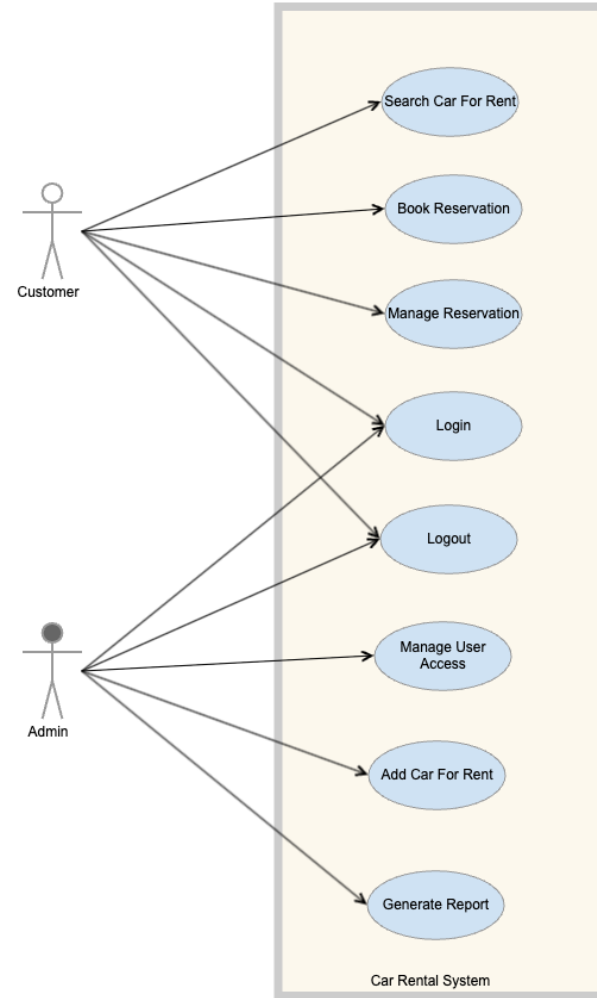
◦ Customers can search for available cars based on criteria like location, date, car type, etc.

◦ Display detailed car information including price, features, and rental terms.

4. Reservations:

Powered by Confluence







Backlog

A

VL

Version ▾ Epic ▾ Type ▾ Quick filters ▾

CRS - Sprint 3 10 Aug – 17 Aug (6 issues)

15 0 0 Complete sprint ...

VL

...

<input checked="" type="checkbox"/> CRS-33 Setup CI/CD		TECHNICAL FOUNDATI...	IN PROGRESS ▾	-	=	VL
CRS-29 Track Rental Periods	MVP - 0.1.0	RENTAL MANAGEMENT	TO DO ▾	3	=	
CRS-30 Track Return Dates	MVP - 0.1.0	RENTAL MANAGEMENT	TO DO ▾	3	=	
CRS-31 Sending Return Reminders	MVP - 0.1.0	RENTAL MANAGEMENT	TO DO ▾	3	=	
CRS-32 Managing Late Returns	MVP - 0.1.0	RENTAL MANAGEMENT	TO DO ▾	3	=	
CRS-15 User Dashboard Access	MVP - 0.1.0	AUTHORIZATION	TO DO ▾	3	=	

Backlog (0 issues)

0 0 0 Create sprint

Your backlog is empty.

+ Create issue



# User Story

Projects /

Car Rental System /

CRS-3 /

CRS-18

## View Car Inventory

+ Add

@ Apps

### Description

- As a rental system administrator,
- I want to view the entire car inventory
- So that I can manage and oversee the available cars.

### Acceptance Criteria:

- The car inventory page displays a list of all cars with their details (make, model, year, license plate number, rental price, and availability).
- The list includes search and filter options for car details.
- Users can view detailed information about each car by selecting it.

### Scenario:

- Given that I am logged in as an administrator and am on the car inventory management page,
- When I view the car inventory list,
- Then I should see all cars listed with their details,
- And I should be able to use search and filter options to narrow down the list,

Details

Assignee

NL

Ngoc Thinh Le

Assign to me

Components

None

Development

1 branch

4 commits11 days ago

2 pull requestsMERGED

Story Points

3

Original estimate

0m

Parent

+

CRS-3 Car Inventory Manage...

Sprint

None+1

Time tracking

No time logged

☒ Include subtasks



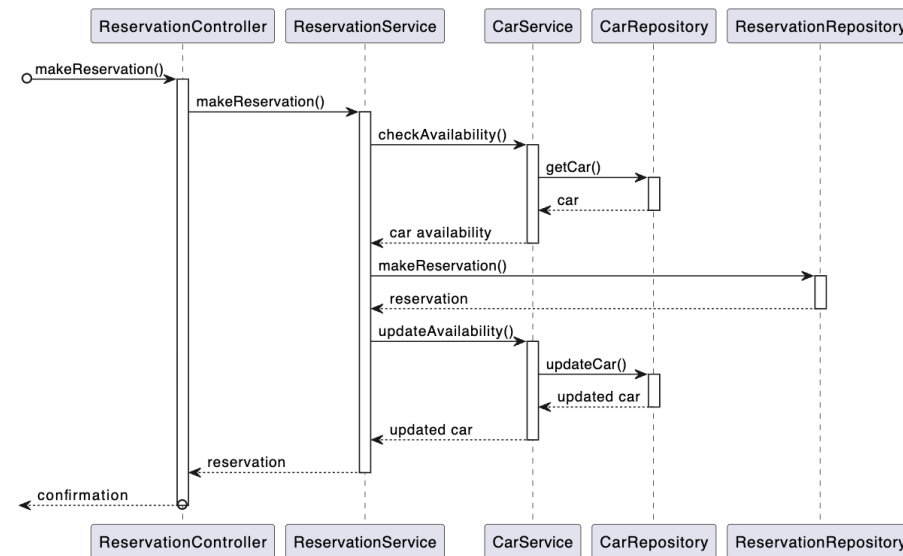
# Sequence Diagram

## Sequence Diagram - Make Rental Reservation

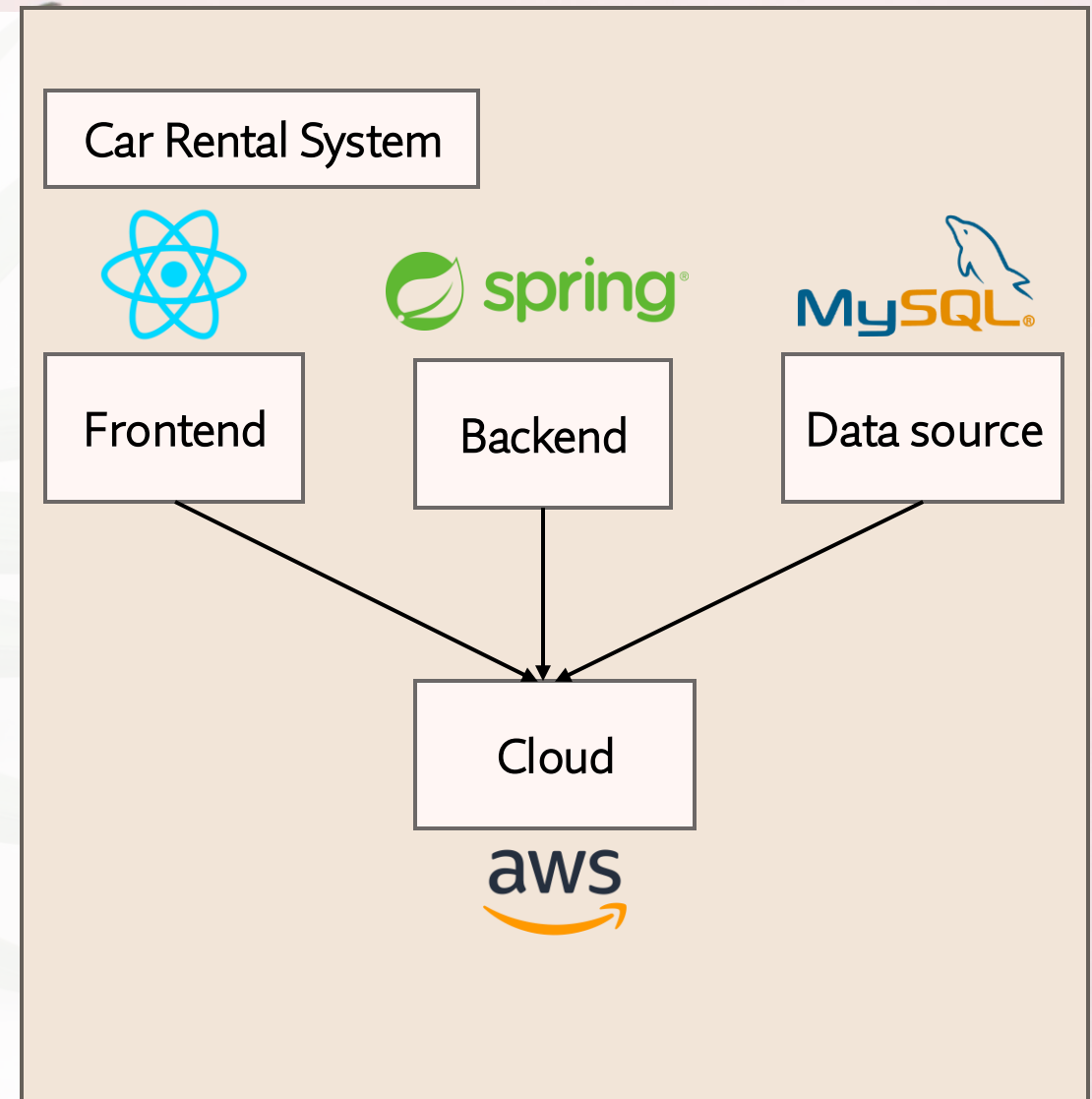


Owned by [Vu Nguyen](#) ...

Last updated: Jul 30, 2024 • 1 min read • [See how many people viewed this page](#)



# Architecture



# Backend (Restful API)

- Spring Boot
- Spring Data JPA
- Spring Security (JWT token)
- Springdoc OpenAPI
- MySQL (AWS RDS)
- Docker
- AWS EC2





car-rental-system

feature/Update\_api\_register\_user

CarRentalSystemApplication-dev

Project

src

main

java

miu.cs425

configurations

constants.enums

controllers

dtos

exceptions

mappers

models

repositories

services

utils

CarRentalSystemApplication

resources

test

target

.gitignore

car-rental-system.iml

docker-compose.yaml

Dockerfile

mvnw

mvnw.cmd

m pom.xml

README.md

External Libraries

Scratches and Consoles

CarRentalSystemApplication.java

m pom.xml (vehicle-rental-system)

```
2      <project xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://maven.apache.org/POM/4.0.0"
32      <dependencies>
33      <dependency>
34          <groupId>org.springframework.boot</groupId>
35          <artifactId>spring-boot-starter-data-jpa</artifactId>
36      </dependency>
37      <dependency>
38          <groupId>org.springframework.boot</groupId>
39          <artifactId>spring-boot-starter-security</artifactId>
40      </dependency>
41      <dependency>
42          <groupId>org.springframework.boot</groupId>
43          <artifactId>spring-boot-starter-web</artifactId>
44      </dependency>
45      <dependency>
46          <groupId>org.modelmapper</groupId>
47          <artifactId>modelmapper</artifactId>
48          <version>3.2.0</version>
49      </dependency>
50
51      <dependency>
52          <groupId>org.springframework.boot</groupId>
53          <artifactId>spring-boot-devtools</artifactId>
54          <optional>true</optional>
55      </dependency>
56      <dependency>
57          <groupId>org.springframework.boot</groupId>
58          <artifactId>spring-boot-docker-compose</artifactId>
59          <scope>runtime</scope>
60          <optional>true</optional>
61      </dependency>
```

project > build > plugins


car-rental-system > m pom.xml

139:19 CRLF UTF-8 4 spaces



←↻⚠ Not secure | 3.149.228.186:8080/swagger-ui/index.html

PowerVaultMIUPersonalOrder details | eBayMIU\_Homework/src...imapedu.sharepoint...CARFAX Vehicle Hist...crisnguyenvn/crs-se...Find your trip - Find...

Swagger  
Supported by SMARTBEAR

/v3/api-docs

Explore

OpenAPI definition

v0OAS3

/v3/api-docs

Servers

http://3.149.228.186:8080 - Generated server url

Authorize

vehicle-controller

GET/api/v1/vehiclesRetrieve all vehicles

POST/api/v1/vehiclesAdd new vehicle

POST/api/v1/vehicles/filterRetrieve all vehicles by filters, sort and paging

GET/api/v1/vehicles/{id}Get vehicle by Id

DELETE/api/v1/vehicles/{id>Delete vehicle by Id

PATCH/api/v1/vehicles/{id}Update vehicle by ID

user-controller

GET/api/v1/usersRetrieve all users

POST/api/v1/usersAdd new user

POST/api/v1/users/filterRetrieve all users by filters, sort and paging

[← Back to pull request #1](#)

## Setup CI/CD #49



Re-run all jobs

### Summary

#### Jobs

-  project-build
-  docker-build
-  deploy

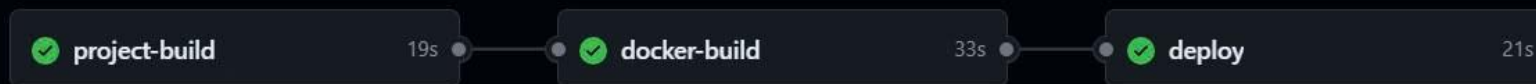
#### Run details

-  Usage
-  Workflow file

Triggered via pull request yesterday	Status	Total duration	Artifacts
 vle-miu synchronize #1 <a href="#">feature/cd-ci</a>	Success	1m 41s	1

### ci-cd.yml

on: pull\_request



### Annotations

5 warnings

#### project-build

The following actions uses node12 which is deprecated and will be forced to run on node16: actions/upload-artifact@v2. For more info: <https://github.blog/...>

[View more](#)

# Front-end Technologies

- React - TypeScript
- Material-UI
- Chart.js
- Axios
- Day.js
- GitHub Action - AWS S3 Bucket



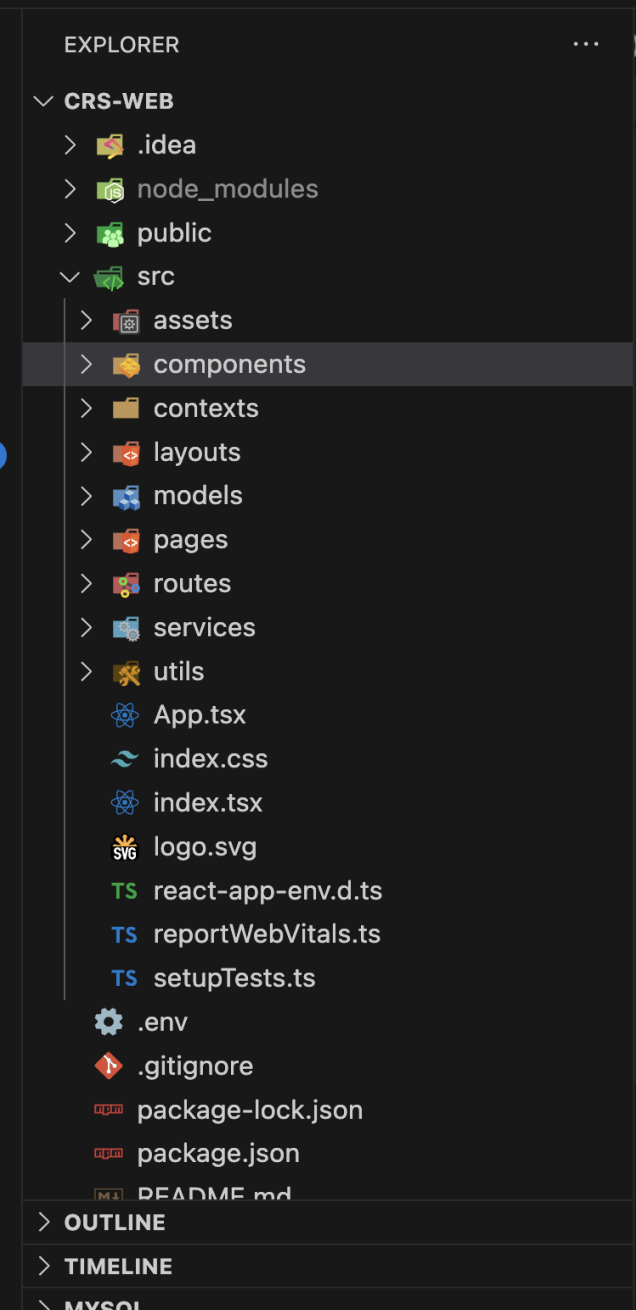
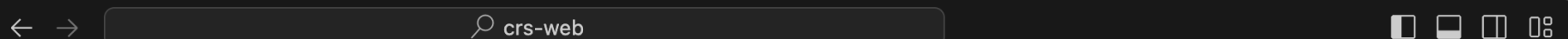


# Front-end Features

- Dashboard
- Reservation System
- User and Vehicle Management
- Error Handling and Notifications








TS reservationService.ts

TS *userService.ts* ✕

VehicleCardList.tsx

```
src > services > user > TS userService.ts >  UserService
```

```

2 import { User } from "../../models/User";
3 import apiInstance from "../../apiInstance";
4 import { FilterAndSortTable } from "../../models/FilterAndSortTable";
5 import { Page } from "../../models/Page";
6 import { CustomError } from "../../utils/customError";
7 import { StatusCode } from "../../models/enums/StatusCodeEnum";
8
9 export class UserService {
10     private USER_ENDPOINT: string = "/v1/users";
11
12     addUser = async (user: User): Promise<User> => {
13         const response: AxiosResponse = await apiInstance.post(
14             this.USER_ENDPOINT,
15             user
16         );
17         return response.data;
18     };
19
20     getAllUsers = async (): Promise<User[]> => {
21         const response: AxiosResponse = await apiInstance.get(
22             this.USER_ENDPOINT
23         );
24         return response.data;
25     };
26
27     getUsersByFilter = async (
28         initFilterAndSortTable: FilterAndSortTable
29     ) => {
30         const response: AxiosResponse = await apiInstance.get(
31             this.USER_ENDPOINT,
32             {
33                 params: initFilterAndSortTable.toParams()
34             }
35         );
36         return response.data;
37     };
38 }

```

## PROBLEMS

## OUTPUT

## DEBUG CONSOLE

## TERMINAL

## PORTS

zsh + ▢ 🗑️ ... ^ ✕

```
/dev/fd/13:18: command not found: compdef
(base) lengocthinh@MacBook-Pro-cua-Le crs-web %
```

```
(base) lengocthinh@MacBook-Pro-cua-Le crs-web %
```


0 Git Graph


Ln 9, Col 27

Spaces: 4

UTF-8

LF

LF { TypeScript





vle-miu / crs-web

Search Type to search



- <> Code
- Pull requests 1
- Actions**
- Projects
- Wiki
- Security
- Insights
- Settings

[← Back to pull request #1](#)

## ✓ Setup CI/CD #7

Re-run all jobs



### Summary

Jobs

- ✓ build (20.x)
- ✓ deploy

Run details

- Usage
- Workflow file

Triggered via pull request yesterday

vle-miu synchronize #1 [feature/ci-cd](#)

Status

Success

Total duration

2m 26s

Artifacts

1

### ci-cd.yml

on: pull\_request

Matrix: build

✓ 1 job completed

Show all jobs

✓ deploy

28s

### Annotations

5 warnings

build (20.x)

# Demo

[Car Rental Application  
\(com.s3-website.us-east-2.amazonaws.com\)](http://com.s3-website.us-east-2.amazonaws.com)



# Thank you

## Deployment information:

- Database: AWS RDS
- Spring API: Docker with EC2
- Reactjs: S3 static website

## CI/CD

- SpringBoot: <https://github.com/vle-miu/crs-service/pull/1>
- Reactjs: <https://github.com/vle-miu/crs-web/pull/1>

API document: [Swagger UI](#)

Front Site: <http://car-rental.com.s3-website.us-east-2.amazonaws.com/>

Account test: superadmin/1234567