

CS673 Software Engineering
Team Rocket - Project RCM: Rental Car Management
Software Design Document

Your project Logo
here if any

<u>Team Member</u>	<u>Role(s)</u>	<u>Signature</u>	<u>Date</u>
Alisa Belousova	Configuration Lead	<u>Alisa Belousova</u>	<u>9/9/2023</u>
Nick Cruz	Team Leader/ Requirements Lead	<u>Nickolas Cruz</u>	<u>9/9/2023</u>
Devon Duddley	QA Lead and Design and Implementation Lead	<u>Devon Dudley</u>	<u>9/9/2023</u>
Kris	Security Lead	<u>Cangqing (Kris) Wang</u>	<u>9/9/2023</u>
Chenghao Ye	Design and Implementation Lead	<u>Chenghao Ye</u>	<u>9/10/2023</u>

Revision history

<u>Version</u>	<u>Author</u>	<u>Date</u>	<u>Change</u>
1	Devon Dudley	9/24/2023	Filled out doc
2	Alisa Belousova	9/26/2023	Deleted mvc pattern cause we don't have it

[Introduction](#)

[Software Architecture](#)

[Class Diagram](#)

[UI Design \(if applicable\)](#)

[Database Design \(if applicable\)](#)

[Security Design](#)

[Business Logic and/or Key Algorithms](#)

[Design Patterns](#)

[Any Additional Topics you would like to include.](#)

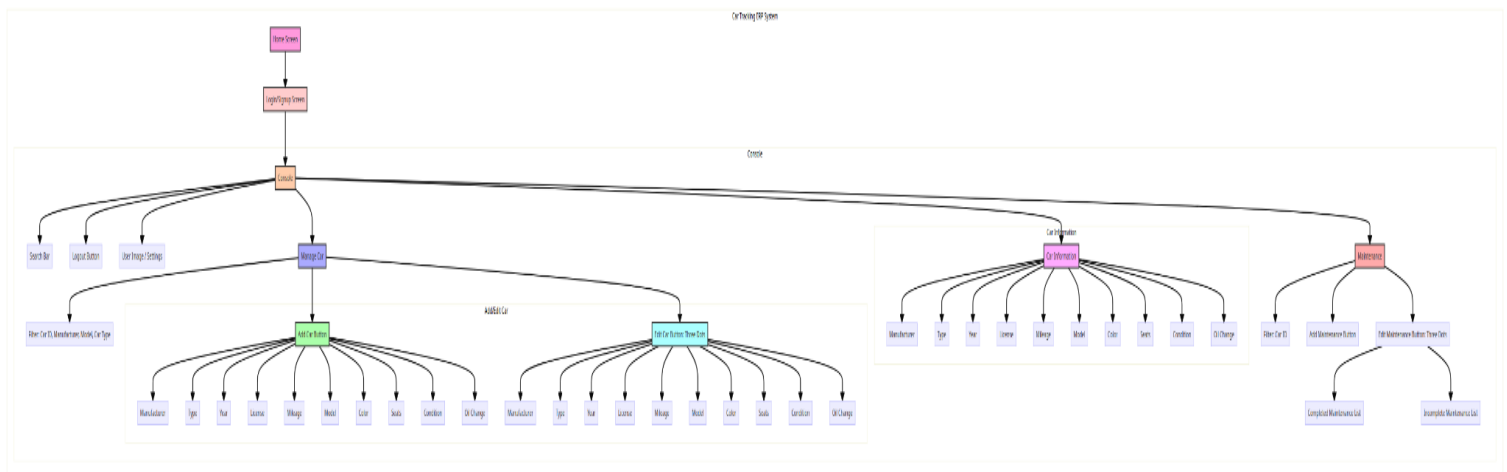
[References](#)

[Glossary](#)

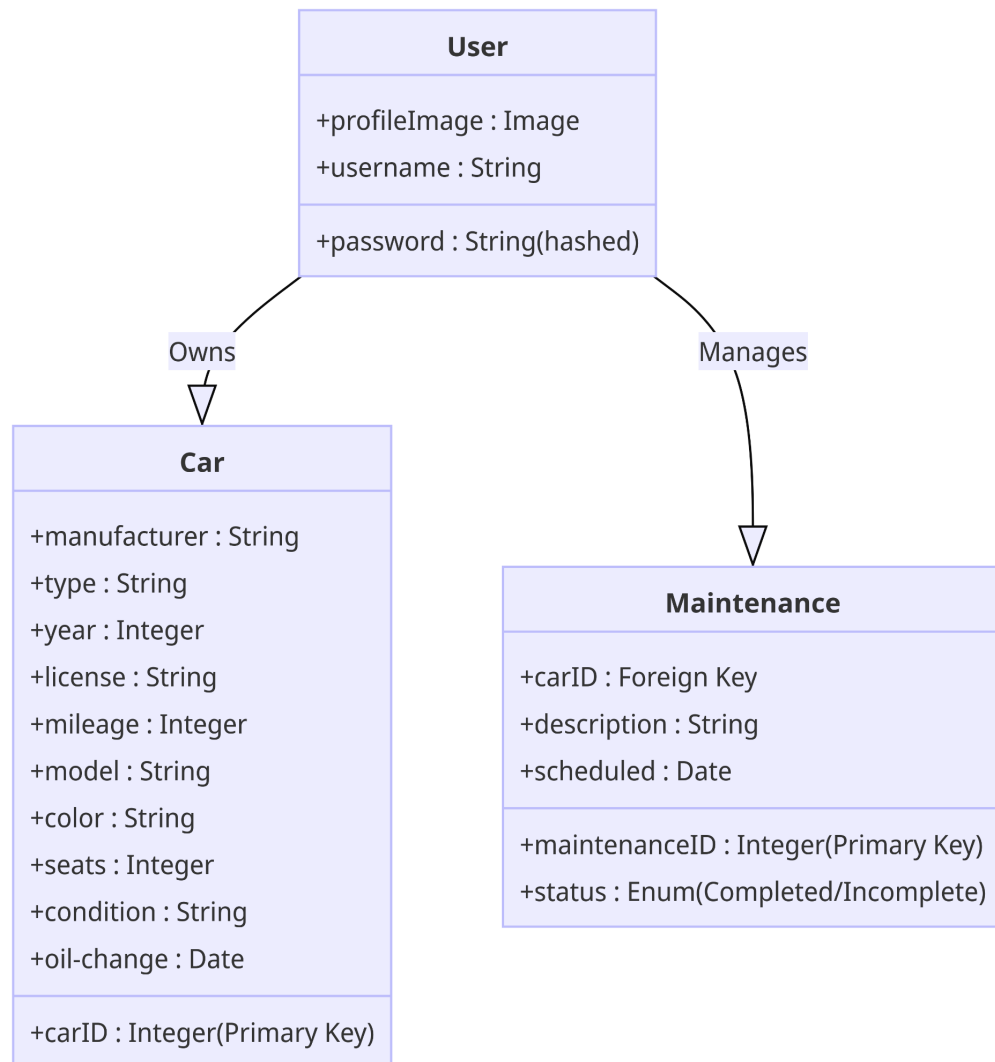
● Introduction

This document provides an in-depth design overview of our car tracking ERP system tailored for individuals renting out their cars. The design goals of our software system are to provide a user-friendly interface, facilitate easy car and maintenance management, ensure the security of user information, and optimize the retrieval and update operations for swift user interactions.

● Software Architecture ([View Diagram](#))



- Class Diagram ([View Diagram](#))

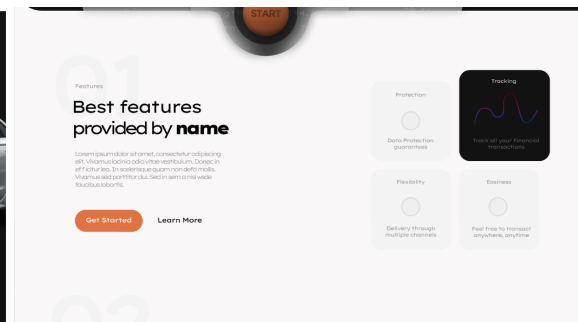


- UI Design (if applicable)

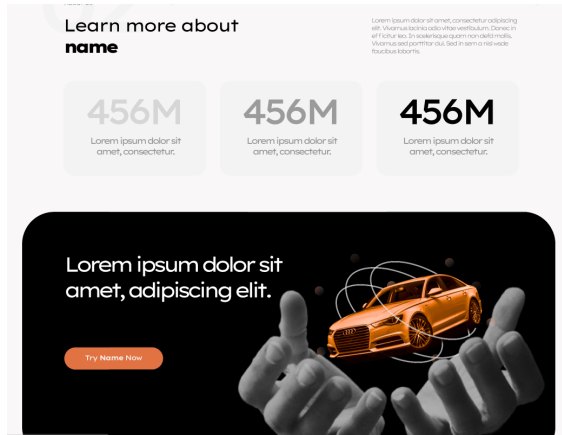
1 Home Screen



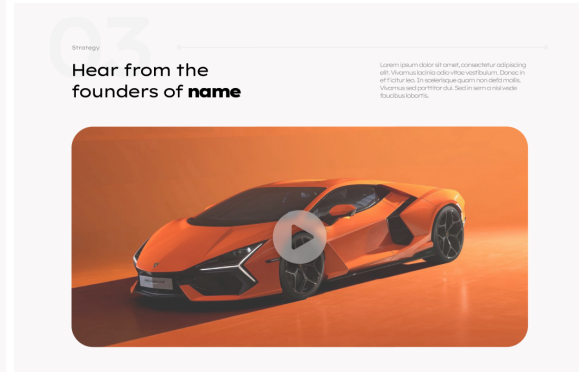
2



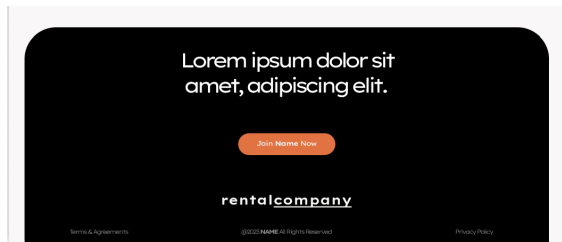
3



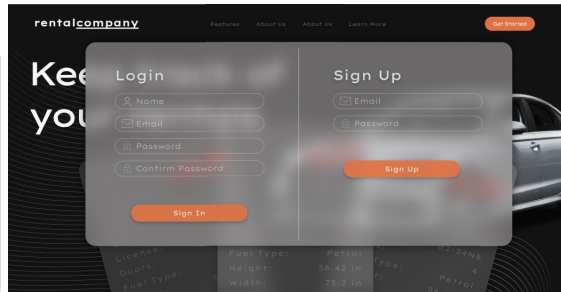
4



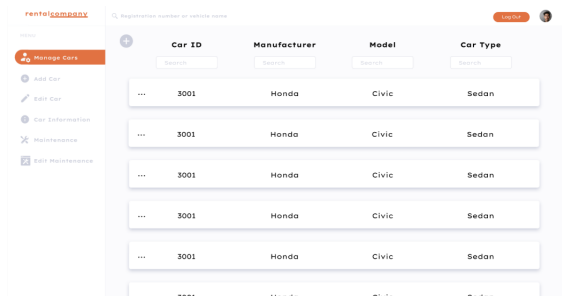
5



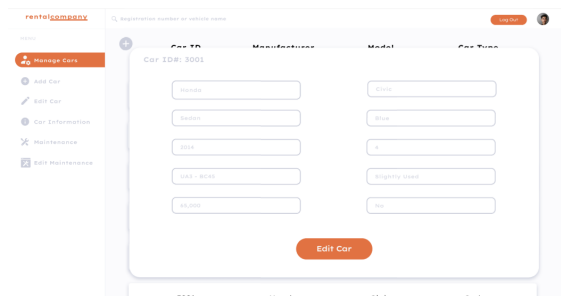
6 Login/Signin Screen



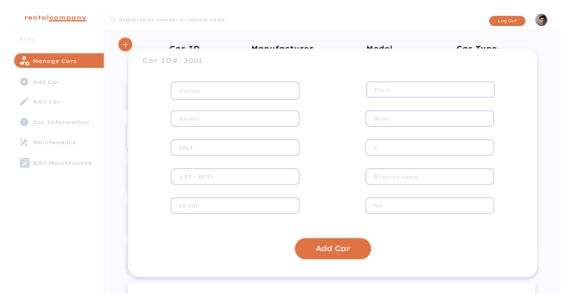
7 Conole



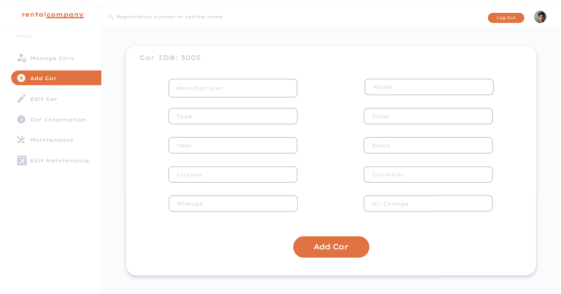
8



9



10



11

12

rentelcompany

Registration number or vehicle name

Car ID# 3005

Manufacturer: Model:

Type: Color:

Year: Seats:

Condition: Next Oil Change:

License:

Edit Car

13

rentelcompany

Registration number or vehicle name

Car ID	Maintenance Needed	Scheduled
3001	Tune Up	10/10/23
3001	Window Broken	10/15/23
3001	Oil Change	10/15/23
3001	Alignment	10/27/23
3001	Tune Up	11/10/23
3001	Tune Up	11/20/23

14

rentelcompany

Registration number or vehicle name

Completed	Date	Maintenance Needed	Scheduled
Tune Up:	10/10/23	<input type="button" value="Tune Up"/>	<input type="button" value="10/10/23"/>
Window Broken:	10/13/23	<input type="button" value="Tune Up"/>	<input type="button" value="10/10/23"/>
Oil Change:	10/13/23	<input type="button" value="Tune Up"/>	<input type="button" value="10/10/23"/>
Alignment:	10/23/23	<input type="button" value="Tune Up"/>	<input type="button" value="10/10/23"/>

● Database Design

ERP system, we've employed MongoDB as our database of choice, given its agility and scalability. This section delves into our database's specific design and schema considerations, detailing the structure of collections such as the User, Cars, and Maintenance collections and how they work together to manage and retrieve data effectively.

● Security Design

Password Hashing: A mechanism that negates storing user passwords in plain, readable text.

Session Management: An efficient system that ensures secure user sessions after login.

Data Validation: A rigorous protocol wherein all input fields undergo exhaustive validation checks to thwart SQL injections or other malevolent entries.

Data Encryption: As a proactive measure, data retained in MongoDB is encrypted, adding an extra security layer.

● Business Logic and/or Key Algorithms

Within our car ERP system, we've designed a set of algorithms and business logic to ensure efficient data retrieval, filtering, and user-car management.

Search Algorithm: Allows users to search through their cars swiftly. The algorithm takes user input and scans the Cars collection in MongoDB, returning matches based on car attributes.

Filter Algorithm: This algorithm helps filter cars based on specific attributes such as Car ID or Manufacturer, enabling more accessible access to entries.

Car Maintenance Scheduler: Schedules and tracks car maintenance.

Additional business logic, including user authentication, car addition, editing, and deletion, are also in place to ensure a seamless experience.

- **Design Patterns**

Singleton Pattern: Ensures only one instance of a class exists.

- **Any Additional Topics you would like to include.**

- **References**

- **Glossary**