



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**
Membre de
HONORIS UNITED UNIVERSITIES

HB TECH SERVICES

FLEET MANAGEMENT SYSTEM

TRAINEESHIP REPORT

3th Year IIR11 - 2022-2023

Prepared By
YAHYA ZINI
MOHAMED HAKKOU

Educational Supervisor
KADI ILHAM

Professional Supervisor
HICHAM BOUSHABA

TABLE OF CONTENTS

Acknowledgment	1
ABSTRACT	2
List of figures	3
General Introduction	4
Presentation of the Organization	5
1.1 HB TECH SERVICES	6
1.2 WHO IS HICHAM BOUSHABA	6
1.3 Achievements and contributions	7
Project Description	8
2.1 INTRODUCTION	9
2.2 KEY FEATURES	9
2.3 HOW IT WORKS	10
2.4 CONCLUSION	10
System Design	11
3.1 INTRODUCTION	12
3.2 USE CASE DIAGRAM	12
3.3 THE SEQUENCE DIAGRAMS	14
3.3.1 AUTHENTICATION SEQUENCE DIAGRAM	14
3.3.2 RESTORE PASSWORD SEQUENCE DIAGRAM	15
3.3.3 CREATE USER SEQUENCE DIAGRAM	16
3.3.4 EDIT USER SEQUENCE DIAGRAM	17
3.3.5 DELETE USER SEQUENCE DIAGRAM	17
3.3.6 CREATE CAR SEQUENCE DIAGRAM	18
3.3.7 EDIT CAR SEQUENCE DIAGRAM	19
3.3.8 DELETE CAR SEQUENCE DIAGRAM	19
3.3.9 CREATE LOCATION SEQUENCE DIAGRAM	20
3.3.10 UPDATE LOCATION SEQUENCE DIAGRAM	20
3.3.11 VISUAL TRACKING IN REAL-TIME SEQUENCE DIAGRAM	21
3.3.12 VISUAL TRACKING HISTORY SEQUENCE DIAGRAM	22
3.3.13 VIEWING WEBSITE SEQUENCE DIAGRAM	23
3.3.14 DISCONNECT SEQUENCE DIAGRAM	23
3.4 CLASS DIAGRAM	24
Implementation	25
4.1 INTRODUCTION	26
4.2 TOOLS AND FRAMEWORKS	26
4.3 ANDROID APP	27
4.2 WEB SITE	30
Conclusion.....	34
WEBOGRAPHIE	35

ACKNOWLEDGMENT

We take this opportunity to thank from the bottom of our hearts everyone who has contributed directly or indirectly to the realization of this work.

We would like to thank our project supervisors, Ms. Kadi and Mr. Boushaba for their support, advices and the quality time they give us.

Then, we would like to warmly thank all the people who helped us throughout our traineeship project, namely all the people working in Information Technology services for their moral support. Thanks to their kindness and their abilities, we benefited from all the necessary elements for the smooth running of our project.

I would also like to thank the professors and teachers of the Moroccan School of Engineering Sciences (EMSI) who helped us acquire the knowledge necessary to carry out this project.

ABSTRACT

In the world of creating websites and applications, web services play a vital role in simplifying and easing people's tasks. They enable developers and engineers to build platforms that streamline complex processes and enhance user experiences on the Internet.

This project encompasses three key components: an Android app, a Server application, and a Client application. The Server application exposes a REST API, this API allows consuming client applications to access a variety of functionalities as network-based services.

The Android app is designed to be installed on smartphones within a fleet of vehicles. It continuously sends real-time location data to a database via the REST API. The Client application, on the other hand, is used by managers to monitor the fleet's vehicles in real time, providing them with the option to track specific vehicles. Additionally, the API can be utilized for seamless integration with other systems.

This project caters to a diverse user base, ranging from small local businesses with tens or hundreds of vehicles to parents seeking to monitor their children's whereabouts in real time. Its primary objective is to enhance the control and tracking capabilities of vehicle managers and owners. Moreover, the exposed API can be leveraged by other developers to customize or extend the application. Notably, the project boasts easy installation, user-friendly operation, and accessibility for individuals with minimal computer skills.

LIST OF FIGURES

<u>Figure 1 : Use Case Diagram</u>	13
<u>Figure 2 : Authentication sequence diagram</u>	14
<u>Figure 3 : Restore password sequence diagram</u>	15
<u>Figure 4 : Create user sequence diagram</u>	16
<u>Figure 5 : Edit user sequence diagram</u>	17
<u>Figure 6 : Delete user sequence diagram</u>	17
<u>Figure 7 : Create car sequence diagram</u>	18
<u>Figure 8 : Edit car sequence diagram</u>	19
<u>Figure 9 : Delete car sequence diagram</u>	19
<u>Figure 10 : Create location sequence diagram</u>	20
<u>Figure 11 : Update Location sequence diagram</u>	20
<u>Figure 12 : Visual Tracking in Real-Time sequence diagram</u>	21
<u>Figure 13 : Visual Tracking History sequence diagram</u>	22
<u>Figure 14 : Viewing Website sequence diagram</u>	23
<u>Figure 15 : Disconnect sequence diagram</u>	23
<u>Figure 16 : Class Diagram</u>	24
<u>Figure 17 : Requesting user permission to access their location</u>	28
<u>Figure 18 : Android Main page</u>	29
<u>Figure 19 : Login page</u>	30
<u>Figure 20 : Restore password page</u>	31
<u>Figure 21 : Home page</u>	31
<u>Figure 22 : Manage cars page</u>	32
<u>Figure 23 : Manage users page</u>	32
<u>Figure 24 : Real time car location page</u>	33
<u>Figure 25 : Location history page</u>	33

GENERAL INTRODUCTION

Web services have become quintessential in web application development, which encourages developers and engineers to create platforms that manage people's complicated tasks and simplify those tasks in websites over the Internet.

This project consists of an Android app, a Server application and a Client application. The Server application exposes a REST API (Web Services developed using **R**epresentational **S**tate **T**ransfer (**REST**) protocol) using, which the consuming client applications can make use of various functionalities as services across the network.

The Android app would be installed in the smart phone present in each vehicle of the fleet, this app would send live location data to the database using the REST API. The manager uses the client application to track the vehicles in real time, the manager can also choose to track a particular vehicle. The API could also be used to integrate the services with other systems.

This project serves to a wide variety of users, from small local businesses owning tens or hundreds of vehicle to parents who would like to track the location of their children in real time. This project aims to help the managers/owners better control and track the vehicles. Also, the exposed API could be used by other developers to customize or extend this application. This project is easy to install, use and hence friendly for users with even minimal computer skills.

CHAPTER 1:

Presentation of the Organization

1.1 HB TECH SERVICES

Introducing HB TECH SERVICES, a thriving IT company solely owned and operated by a passionate mobile development expert. With a personal touch and unwavering dedication, Hicham Boushaba powerhouse specializes exclusively in mobile app development. From crafting innovative mobile applications to providing meticulous mobile consulting.

1.2 WHO IS HICHAM BOUSHABA

Hicham Boushaba is a Senior Mobile App Developer who started his journey after secondary education in Preparatory classes for engineering schools at Omar Ibn Al Khattab high school in Meknes from 2007 to 2009. In the same year, he succeeded in joining the National Institute of Posts and Telecommunications (INPT), where he earned his engineering degree in Information and Communications Technologies in 2012. He began his professional career as a Core Network Engineer at Huawei for one year, followed by a role as an NGN/IMS Support & Operations Engineer at INWI for four years. Hicham transitioned to the position of Senior Android Developer at Atos for five months, and he continued in the same role at IBM for eight months. He then joined HEETCH as an Android Developer for a year and, in November 2020, started working as an Android Engineer with Automattic, where he continues to work today.

1.3 Achievements and contributions

Hicham Boushaba has accumulated over a decade of experience in the IT field. During this time, he has gained valuable insights and expertise through his work with various companies, as well as his personal project contributions, which include:

1. Salaat First (Personal Project): Hicham developed the Salaat First Android app, a personal project that accurately calculates local prayer times using multiple conventions. The app also provides notifications for each prayer and allows users to select from different adhan sounds.
2. Tasbihat (Personal Project): Hicham independently worked on the Tasbihat Android app, another personal project. This app serves as a reminder for adhkaar (God's remembrance) and displays reminders in a non-intrusive manner, making it a convenient tool for spiritual practices.
3. Automattic's WooCommerce Android App: In addition to his personal projects, Hicham is actively involved in assisting Automattic in developing new features for the WooCommerce Android app. His contributions play a crucial role in enhancing the app's functionality and user experience.

CHAPTER 2:

Project Description

2.1 INTRODUCTION

The project is a comprehensive car fleet tracking system that offers a robust solution for businesses, rental companies, and concerned parents. It comprises a RESTful API, a SQL Server database, and two distinct applications for the frontend – an Android app for in-car use and a web app developed using React.js. The primary objective of this project is to provide real-time tracking and historical location data for vehicles.

2.2 KEY FEATURES

- Real-Time Tracking: The Android app, installed in each vehicle, continuously sends real-time location data to the REST API.
- Data Storage: The REST API manages the data flow between the Android app and the SQL Server database, ensuring secure and efficient data storage.
- Web Application: The web app, built with React.js, serves as the frontend interface for administrators or concerned parties to access vehicle location information.
- Historical Location Data: The system stores historical location data for each vehicle, allowing users to view the location history for specific days or periods.
- Business Applications: The system is tailored for rental stores and companies that need to monitor the whereabouts of their vehicle fleet in real time. It offers valuable insights into vehicle usage and can assist in optimizing operations.
- Parental Monitoring: Additionally, the system can be used by parents to track the real-time location of their children, enhancing safety and peace of mind.

2.3 HOW IT WORKS

- The Android app continuously collects GPS data and sends it to the REST API.
- The REST API processes the incoming location data and stores it in the SQL Server database.
- The web app's admin interface accesses the database through the API, providing a user-friendly dashboard to monitor vehicle locations in real time and access historical location data.

2.4 CONCLUSION

In summary, this car fleet tracker project provides a comprehensive solution for vehicle tracking, offering both real-time monitoring and historical location data access. It can benefit rental companies, businesses with vehicle fleets, and parents concerned about their children's whereabouts. The combination of the Android app, REST API, SQL Server database, and React.js web app creates a powerful tool for location-based tracking and analysis.

CHAPTER 3:

System Design

3.1 INTRODUCTION

Once the project requirements have been thoroughly understood, the subsequent phase in the development process is System Design. An effective system design lays the groundwork for a robust and reliable project. Typically, system designs are represented through various **UML** (Unified Modeling Language) diagrams, which serve as standardized and universally recognized tools. These diagrams adhere to specific rules and conventions, facilitating seamless communication among all project stakeholders, ensuring everyone is aligned when discussing abstract concepts related to the project's design and architecture.

3.2 USE CASE DIAGRAM

Use case diagrams serve as a visual representation of how various actors interact with a system, offering a means to capture and illustrate requirements effectively. These diagrams are designed to be straightforward and descriptive, making them easily comprehensible to both end users and domain experts. They play a crucial role in facilitating clear communication between end users and development teams. In the following use case diagram, we outline three key roles: the Visitor, the Admin, and the Android App.

- **Visitor:** As a Visitor, interactions with the system include viewing the website, logging into the application, and restoring a forgotten password.
- **Admin:** In addition to the Visitor's capabilities, the Admin enjoys extended privileges, allowing them to manage cars (create, delete, update, select), manage users (create, delete, update, select), perform visual tracking of cars in real-time, access location history for each car on a map, and disconnect from the system.
- **Android App:** The Android App contributes to the system by creating or updating car locations using registration plate information.

This use case diagram, illustrated in Figure 1, provides a comprehensive overview of the project's actors and the diverse use cases through which they interact with the system, helping to guide the development process and ensure clarity in project discussions.

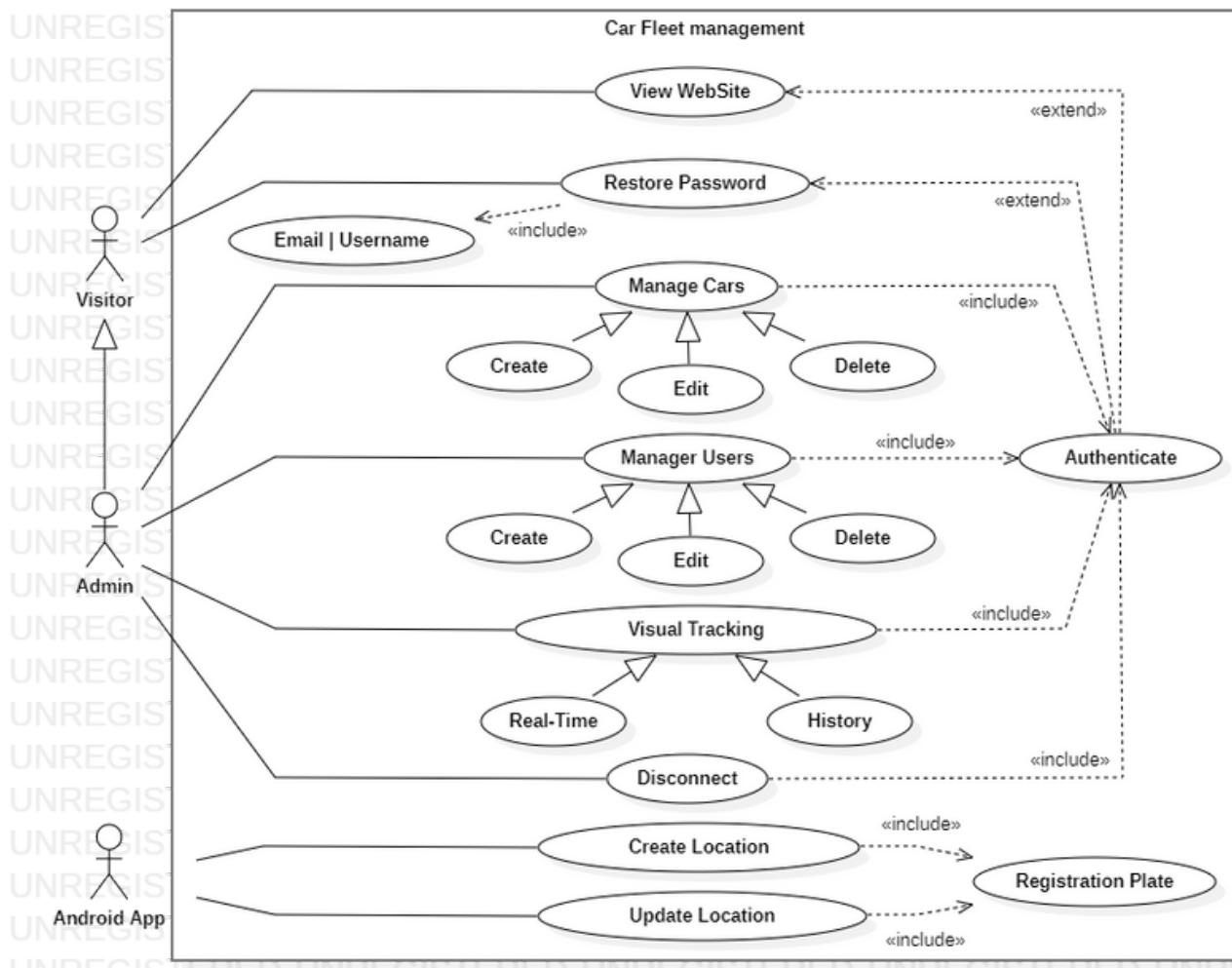


Figure 1: Use Case Diagram

3.3 THE SEQUENCE DIAGRAMS

A sequence diagram is a visual representation used in system design and software engineering to illustrate the chronological order of interactions and communications among different components, objects, or entities within a system. It consists of lifelines representing system participants, messages depicting communication between them, and activation bars showing the duration of their engagement in specific actions or message processing. Sequence diagrams are instrumental in designing, documenting, and communicating the dynamic behavior of a system, aiding developers in understanding interaction flows, facilitating stakeholder communication, and serving as a foundation for testing and debugging processes.

3.3.1 AUTHENTICATION SEQUENCE DIAGRAM

The diagram below shows an authentication sequence in action, portraying the step-by-step process of verifying a user's identity to grant access to WebSite.

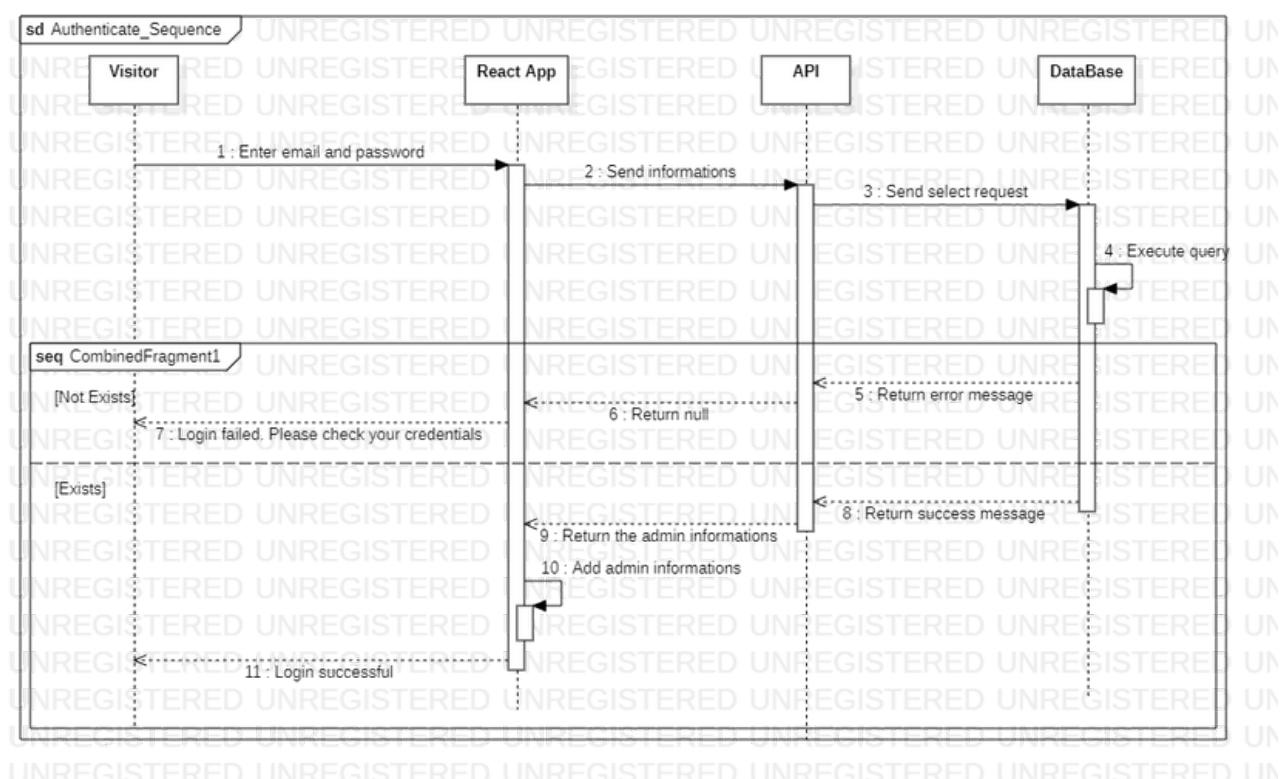


Figure 2: Authentication sequence diagram

3.3.2 RESTORE PASSWORD SEQUENCE DIAGRAM

The diagram below shows a "Restore Password" sequence in action, illustrating the step-by-step process of enabling a user to recover a forgotten password.

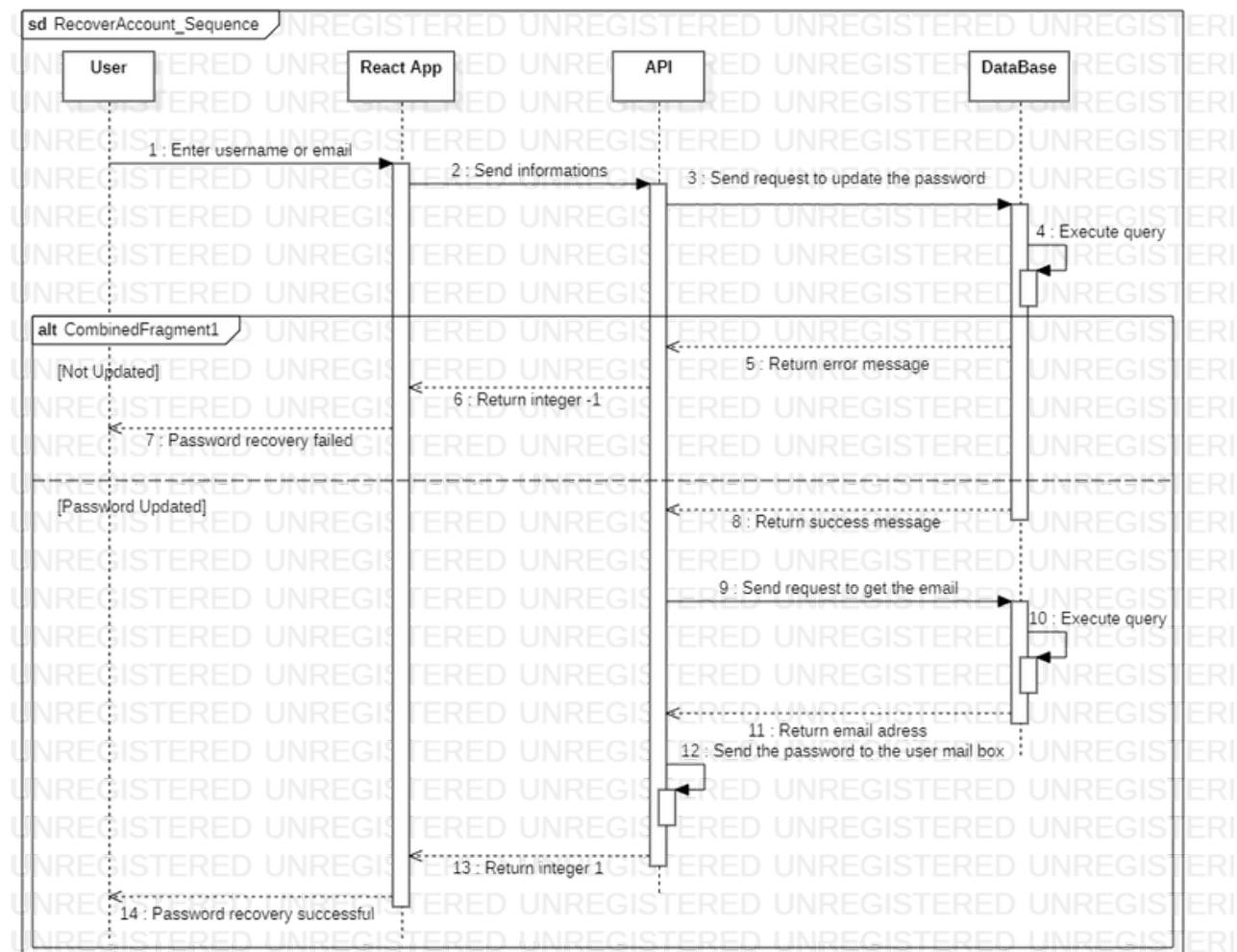


Figure 3: Restore password sequence diagram

3.3.3 CREATE USER SEQUENCE DIAGRAM

The diagram below shows a "Create User" sequence in action, providing a visual representation of the step-by-step process for adding a new user to the database.

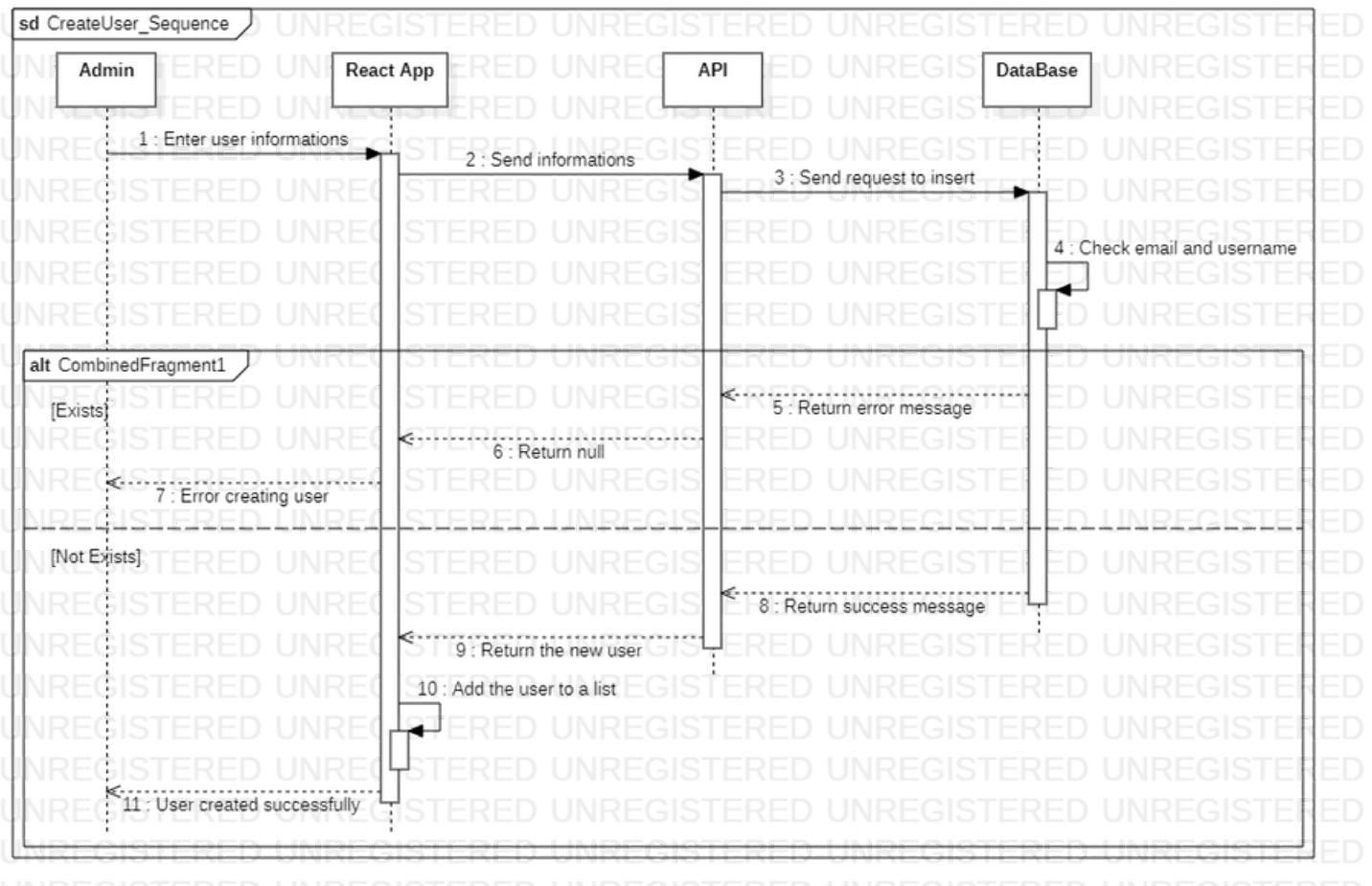


Figure 4: Create user sequence diagram

3.3.4 EDIT USER SEQUENCE DIAGRAM

The diagram below shows a "Edit User" sequence in action, providing a visual representation of the step-by-step process for updating a user in the database.

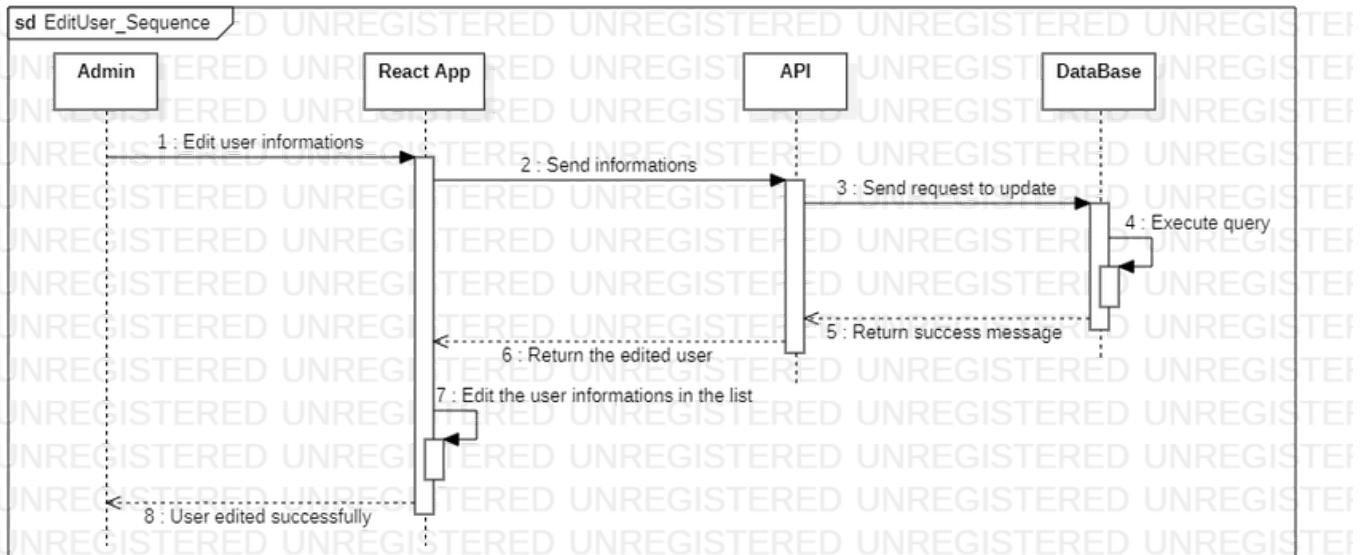


Figure 5: Edit user sequence diagram

3.3.5 DELETE USER SEQUENCE DIAGRAM

The diagram below shows a "Delete User" sequence in action, providing a visual representation of the step-by-step process for Deleting a user form the database.

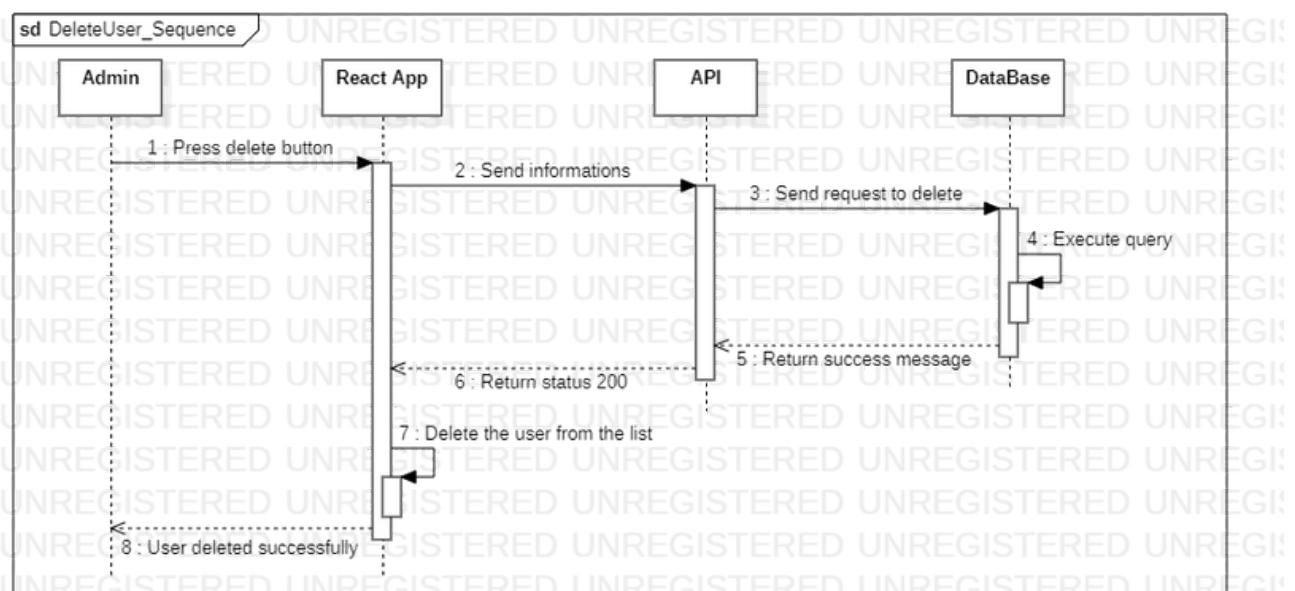


Figure 6: Delete user sequence diagram

3.3.6 CREATE CAR SEQUENCE DIAGRAM

The diagram below illustrates a "Create Car" sequence, providing a visual representation of the systematic process for adding a new car or vehicle to the database

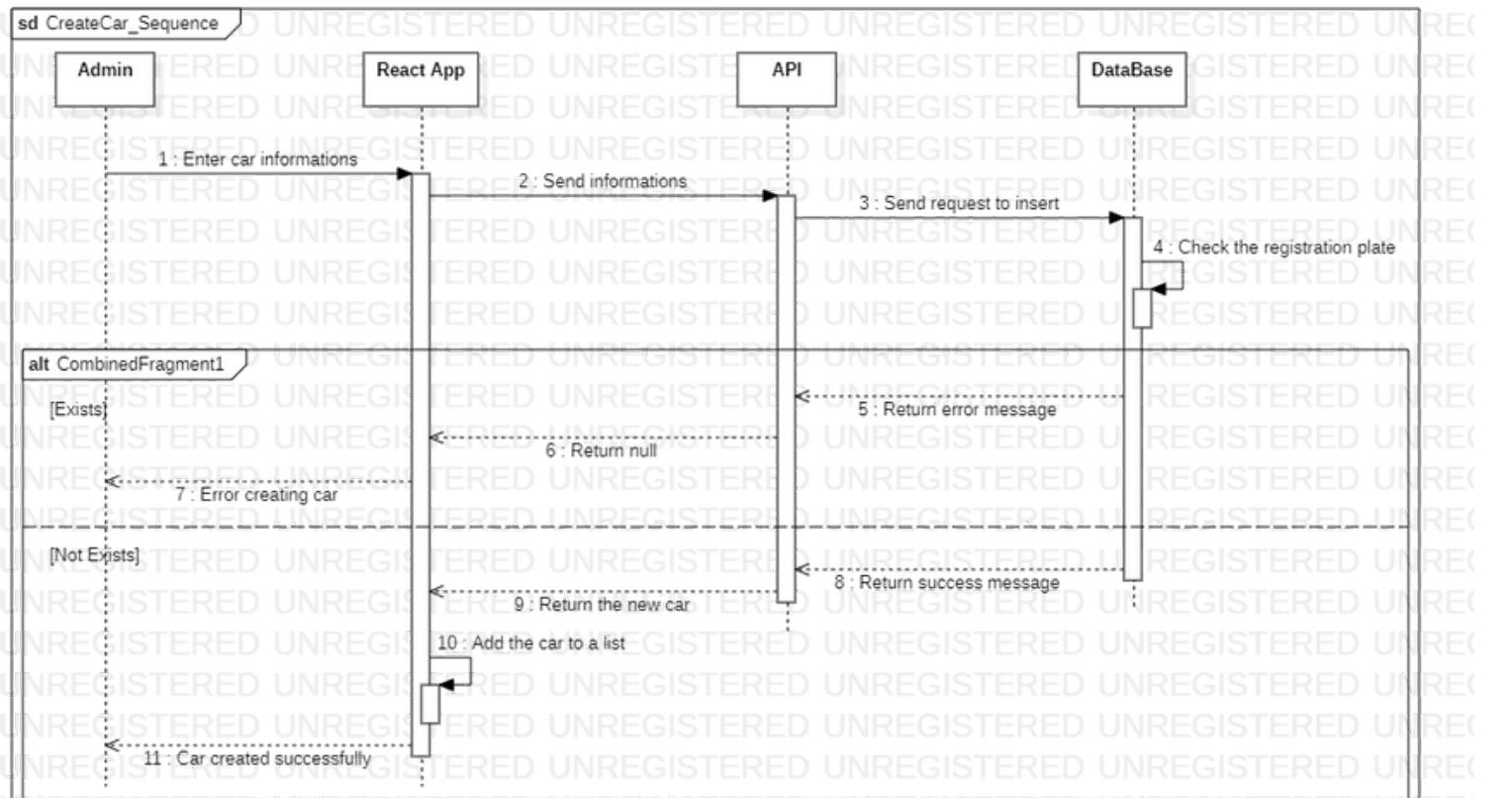


Figure 7: Create car sequence diagram

3.3.7 EDIT CAR SEQUENCE DIAGRAM

The diagram below illustrates a "Edit Car" sequence, providing a visual representation of the systematic process for Updating a car or vehicle in the database.

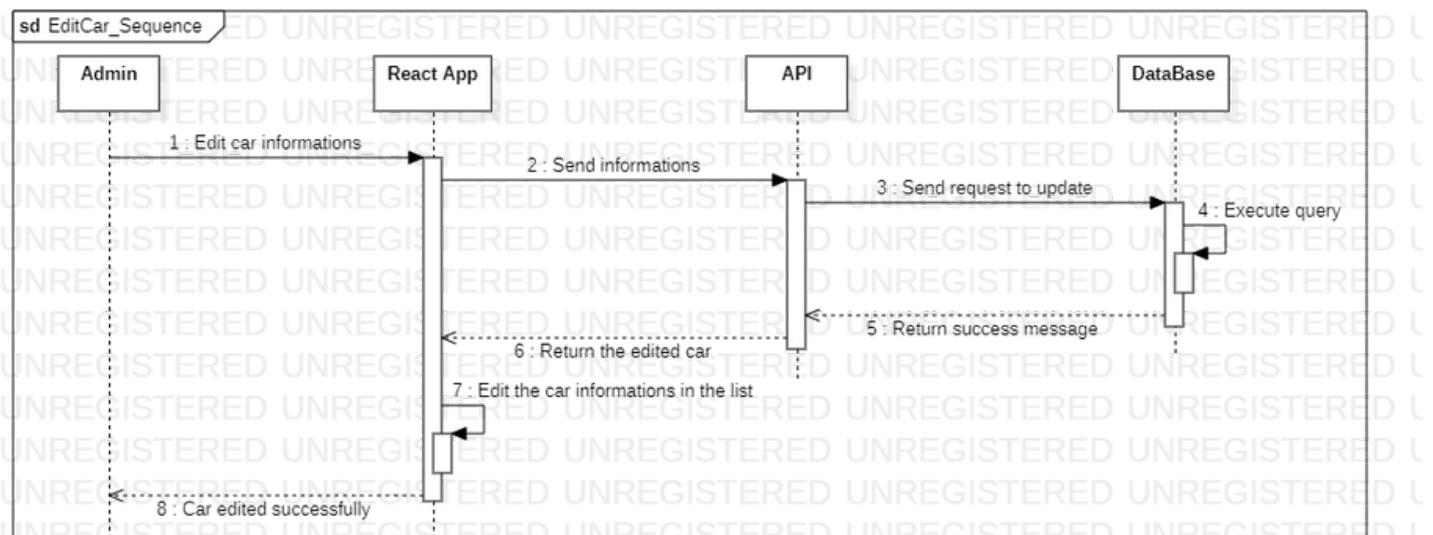


Figure 8: Edit car sequence diagram

3.3.8 DELETE CAR SEQUENCE DIAGRAM

The diagram below illustrates a "Delete Car" sequence, providing a visual representation of the systematic process for deleting a car or vehicle from the database.

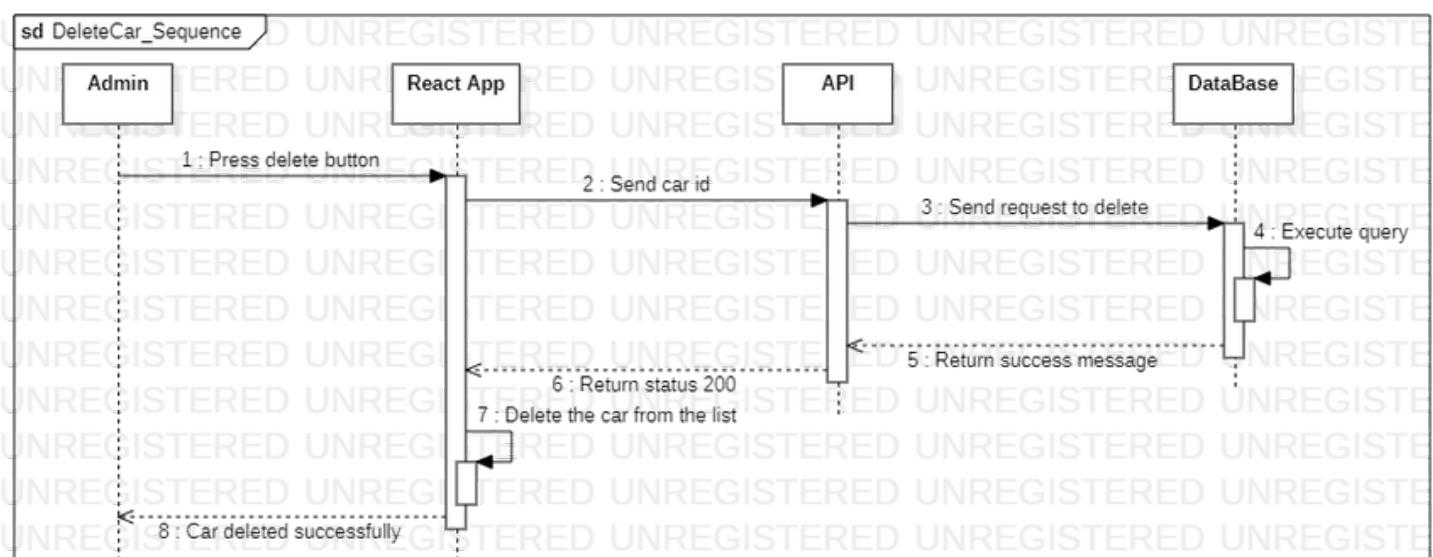


Figure 9: Delete car sequence diagram

3.3.9 CREATE LOCATION SEQUENCE DIAGRAM

The diagram below illustrates a "Create Location" sequence, providing a visual representation of the systematic process for adding a new Location to the database.

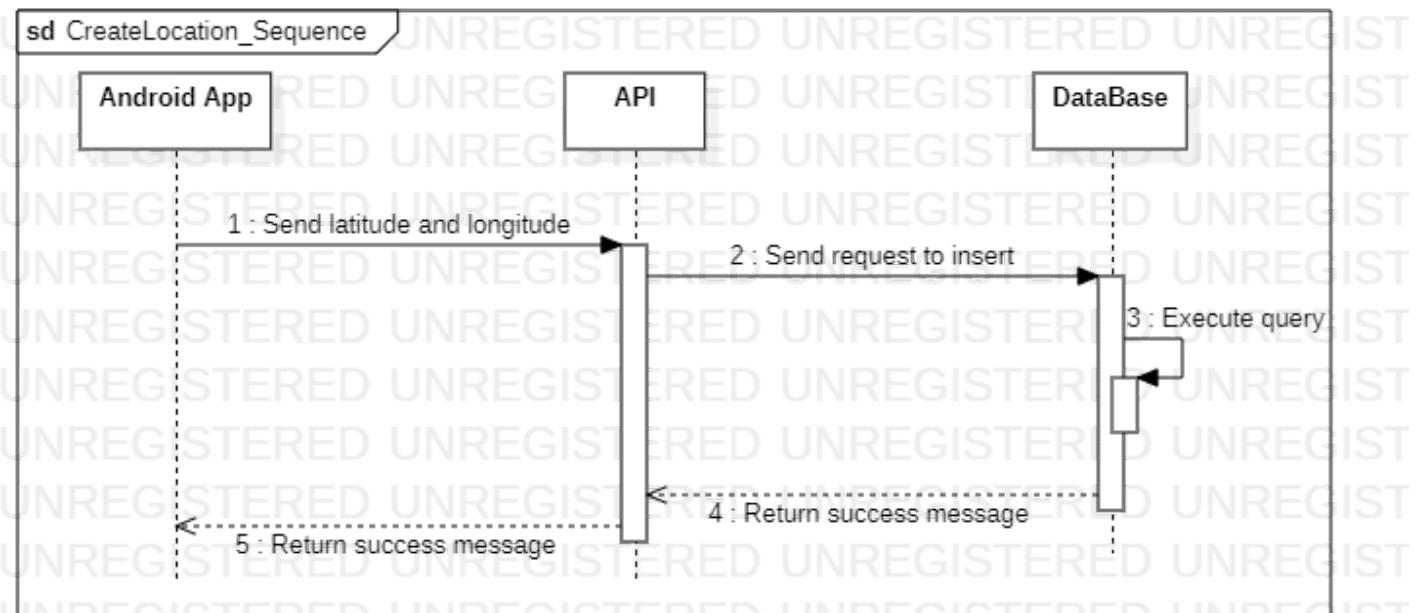


Figure 10: Create location sequence diagram

3.3.10 UPDATE LOCATION SEQUENCE DIAGRAM

The diagram below illustrates a "Update Location" sequence, providing a visual representation of the systematic process for Updating a Location in the database.

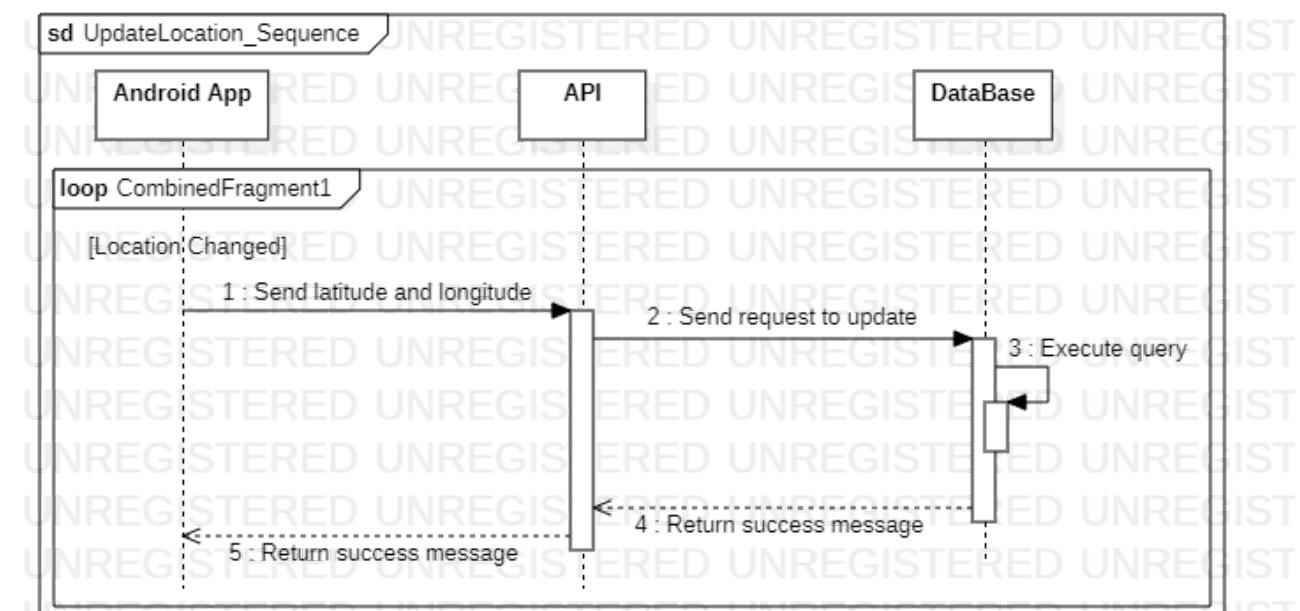


Figure 11: Update Location sequence diagram

3.3.11 VISUAL TRACKING IN REAL-TIME SEQUENCE DIAGRAM

The diagram below illustrates a "Visual Tracking in Real-Time" sequence, providing a visual representation of the systematic process for monitoring and tracking vehicles or objects in real-time.

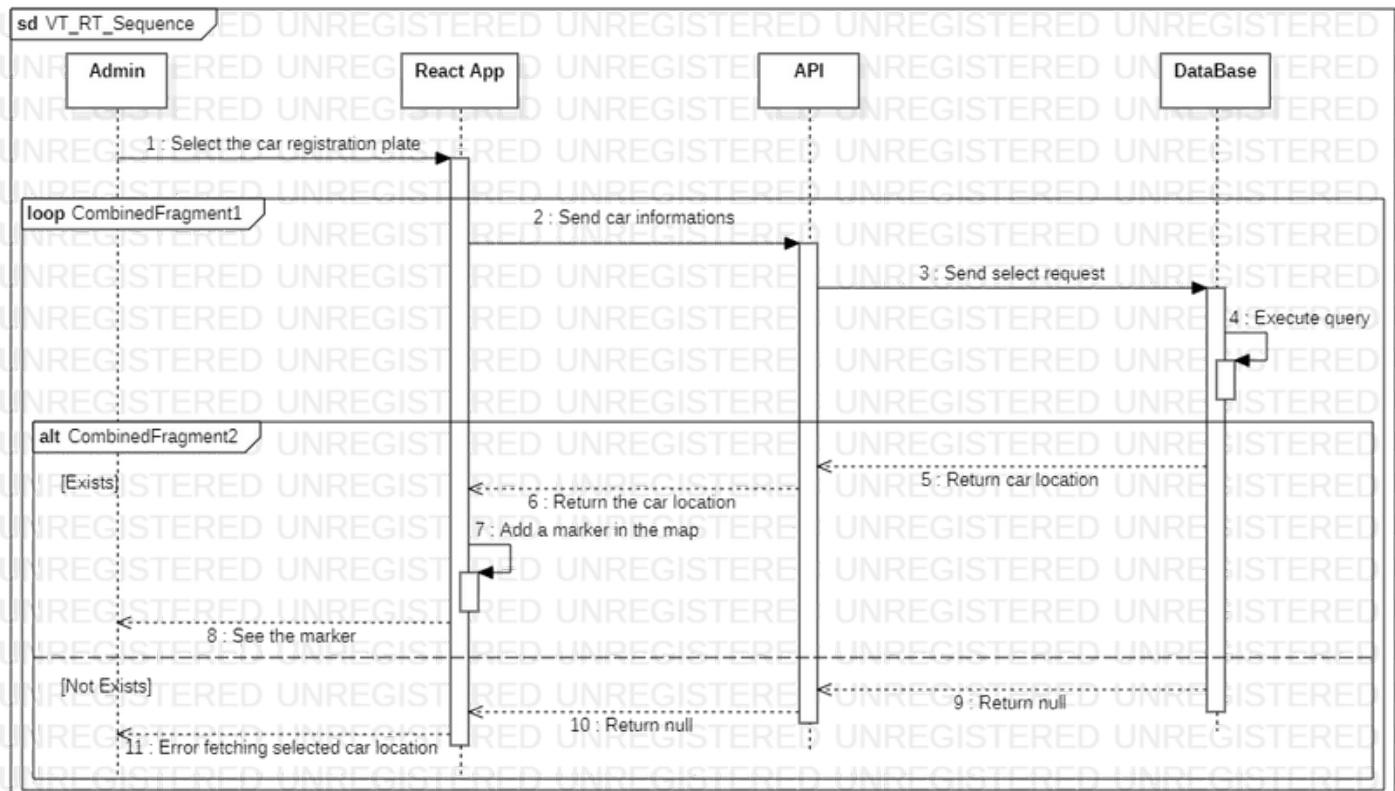


Figure 12: Visual Tracking in Real-Time sequence diagram

3.3.12 VISUAL TRACKING HISTORY SEQUENCE DIAGRAM

The diagram below illustrates a "Visual Tracking History" sequence, providing a visual representation of the systematic process for accessing and reviewing historical tracking data of vehicles or objects.

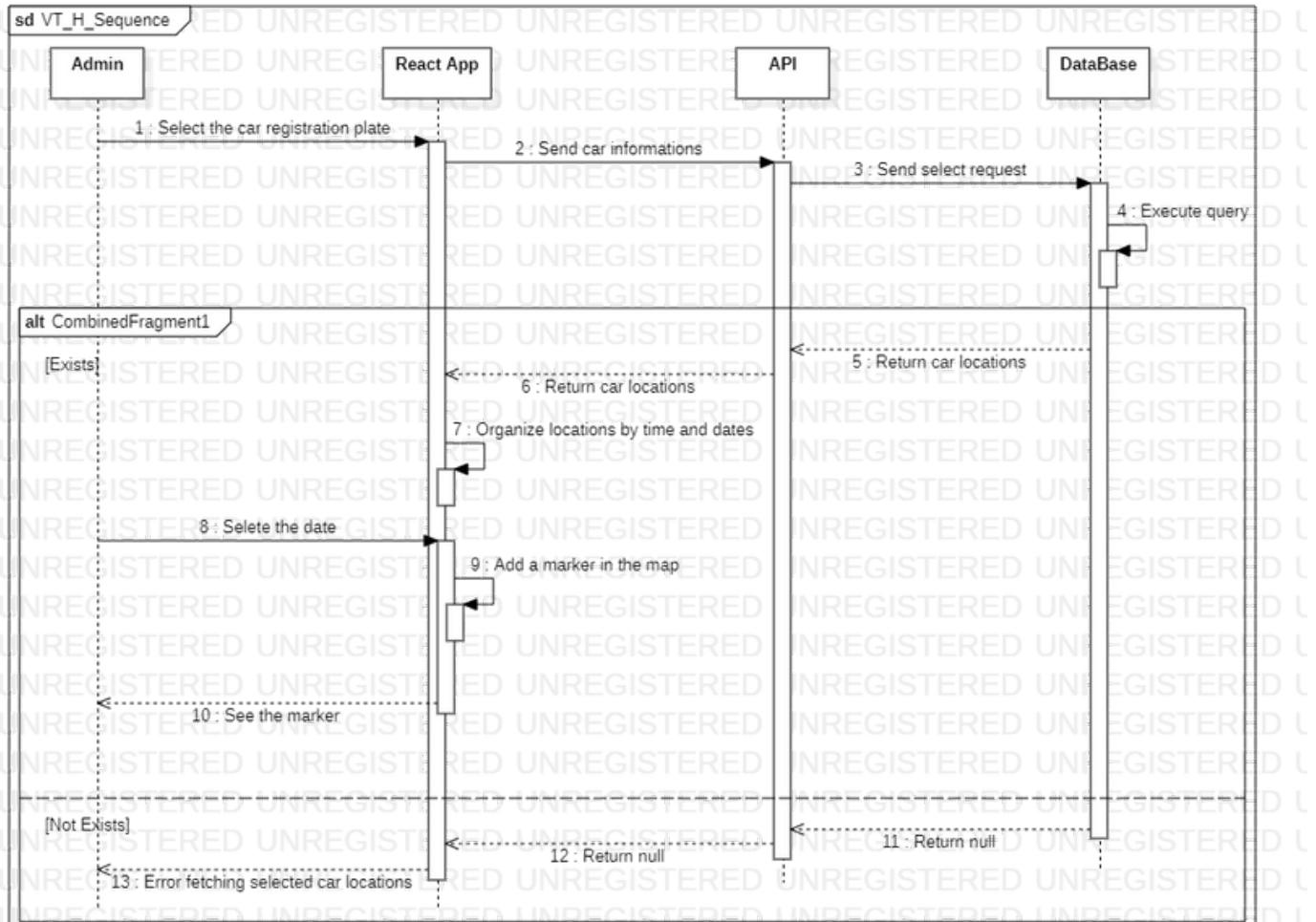


Figure 13: Visual Tracking History sequence diagram

3.3.13 VIEWING WEBSITE SEQUENCE DIAGRAM

The diagram below illustrates a "Viewing Website" sequence, providing a visual representation of the process involved when a user accesses and views the website

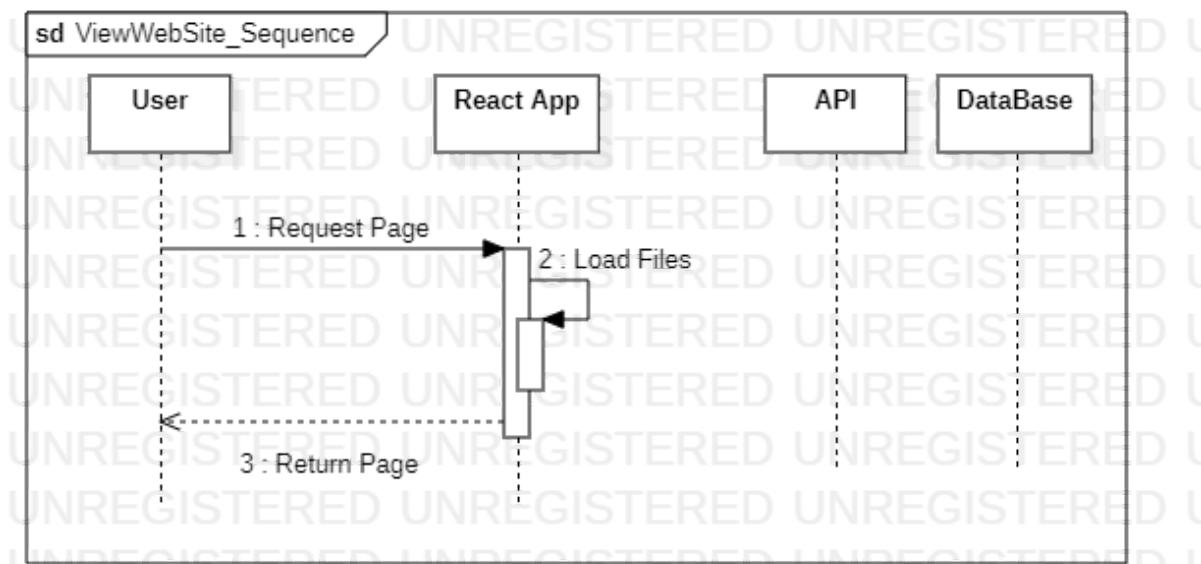


Figure 14: Viewing Website sequence diagram

3.3.14 DISCONNECT SEQUENCE DIAGRAM

The diagram below illustrates a "Disconnect" sequence, providing a visual representation of the step-by-step process for ending or terminating the connection.

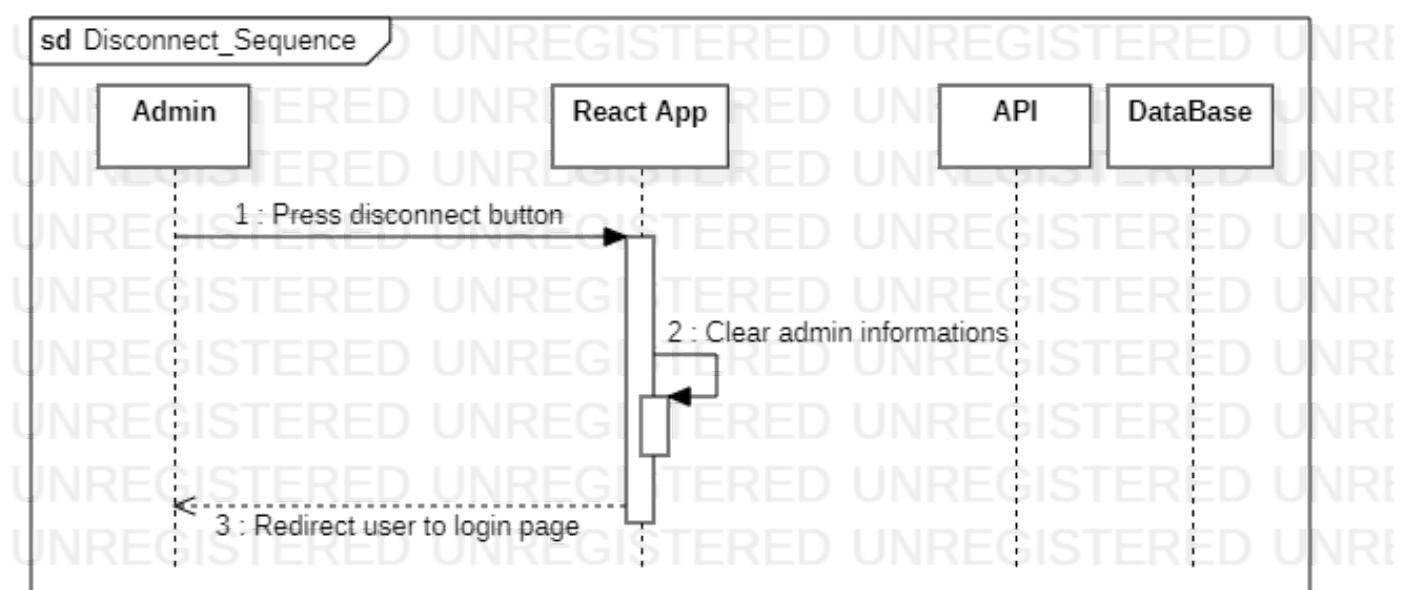


Figure 15: Disconnect sequence diagram

3.4 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

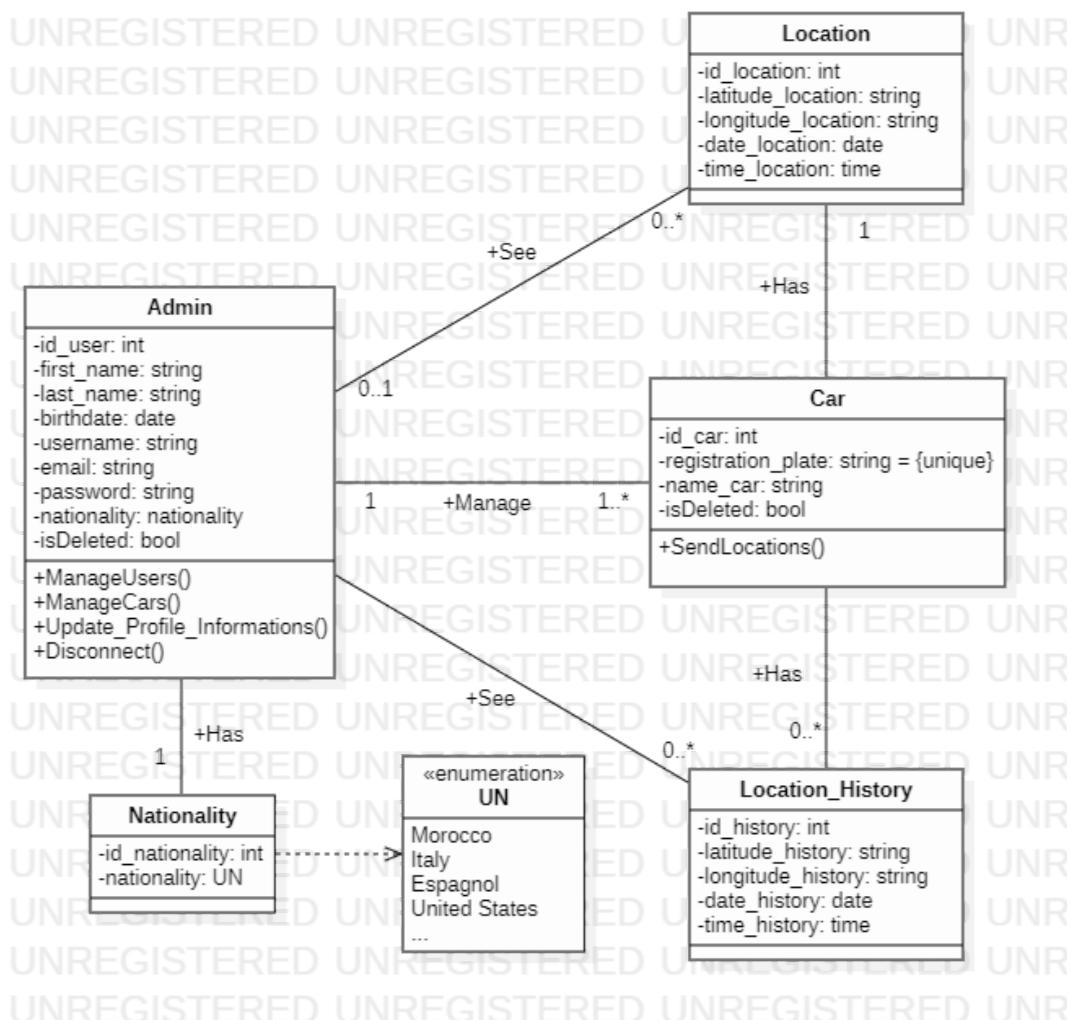


Figure 16: Class Diagram

CHAPTER 4:

Implementation

4.1 INTRODUCTION

The fleet management system comprises a web application with an accessible API for vehicle registration and tracking. Additionally, an Android app has been developed as part of the project. Furthermore, a web page has been created to display real-time vehicle locations on a map and update their positions continuously. This chapter provides an overview of our application, including key output screens for illustration.

4.2 TOOLS AND FRAMEWORKS

The following are the tools, frameworks and software's We have used for the development of this project:

Visual Studio Code: is a lightweight and highly extensible source code editor developed by Microsoft. It's designed for various programming languages and platforms.



Android Studio: Android Studio is an integrated development environment (IDE) specifically designed for Android app development. It provides tools, resources, and a code editor tailored to create, test, and deploy Android applications.



Eclipse: Eclipse is a widely used integrated development environment (IDE) primarily known for Java development but supports various programming languages through plugins.



Postman: Postman is a popular REST client tool used by developers for testing and interacting with web APIs.



SQL Server database: SQL Server is a relational database management system (RDBMS) developed by Microsoft. It's designed to store and manage structured data efficiently.



NO-IP Dynamic DNS: NO-IP Dynamic DNS (Domain Name System) is a service that allows users to associate a dynamic IP address with a domain name. It's commonly used by individuals and organizations that have internet connections with changing (dynamic) IP addresses.



Java: Java is a versatile and widely-used programming language known for its portability and platform independence. It's object-oriented, strongly typed, and features automatic memory management.

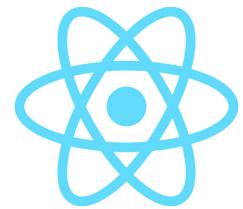


Spring boot:

Spring Boot is an open-source Java-based framework that simplifies the development of standalone, production-ready applications. It provides a streamlined way to build and deploy Java applications by offering a set of pre-configured templates and conventions.



React JS: React JS, often referred to as React, is an open-source JavaScript library for building user interfaces. It was developed and is maintained by Facebook and a community of individual developers and companies. React is widely used for creating interactive and dynamic web applications.



4.3 ANDROID APP

The Android app operates by first requesting user permission to access their location. Once granted, users can input their registered vehicle's license plate number, which is stored in our database. Subsequently, the app initiates location tracking, continuously sending updated location information whenever a change occurs.

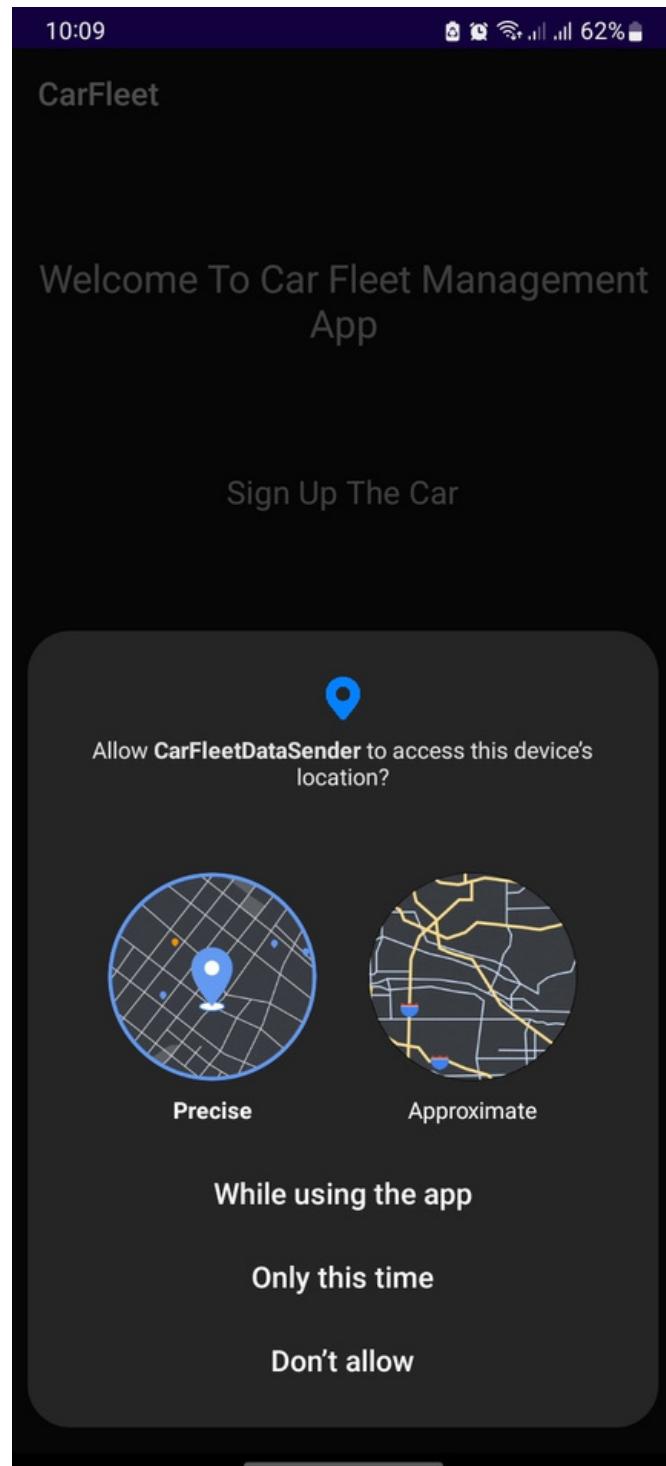


Figure 17: Requesting user permission to access their location

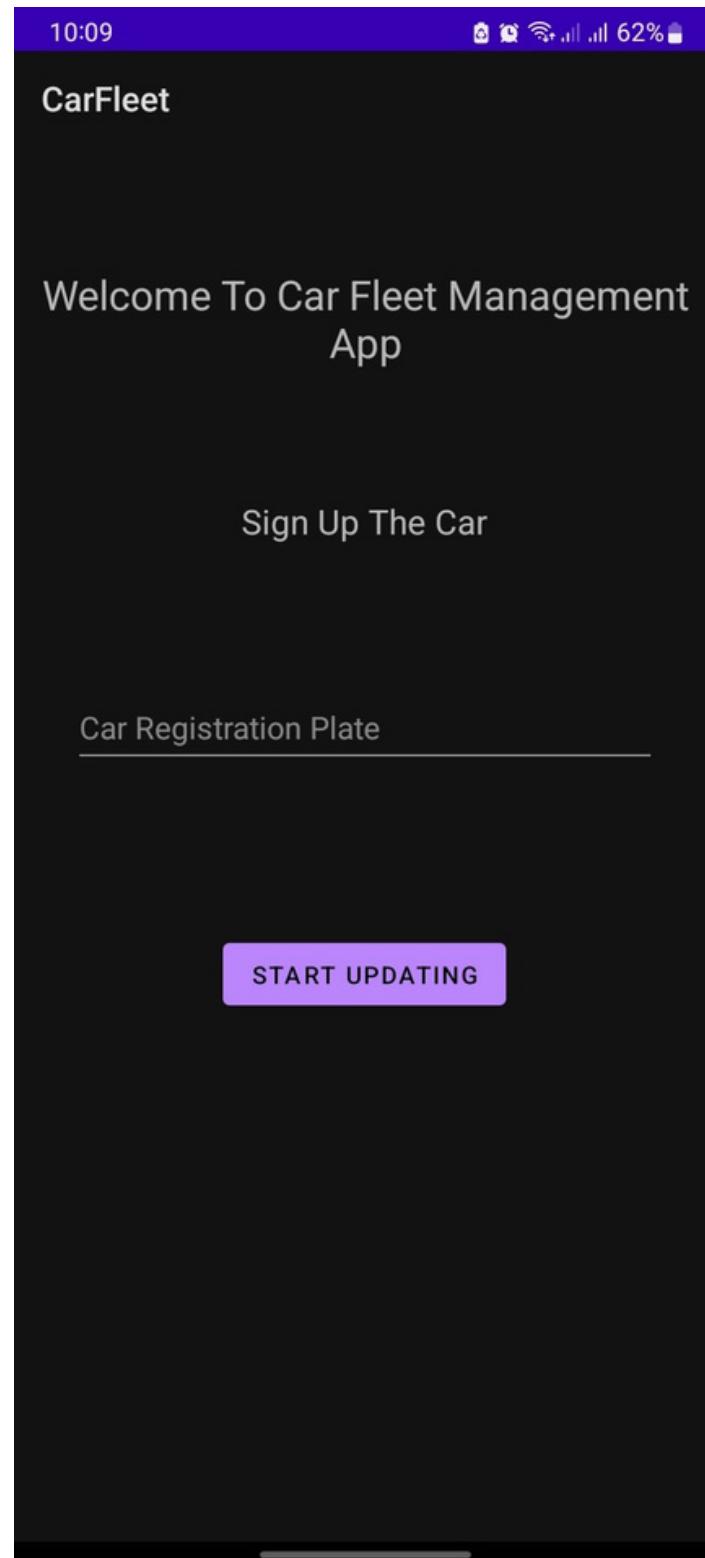


Figure 18: Android Main page

Upon entering the vehicle's registered license plate number and clicking on the "Start Updating" button, the Android app will commence transmitting the vehicle's location with high accuracy whenever any change in its position occurs.

4.2 WEB SITE

The website features comprehensive authentication functionalities, including user login and the ability to reset forgotten passwords. It also offers user and vehicle management capabilities, enabling administrators to oversee and control user accounts and vehicle data. Additionally, the website provides a real-time visual tracking feature for monitoring vehicle locations and offers access to historical location data for each vehicle.

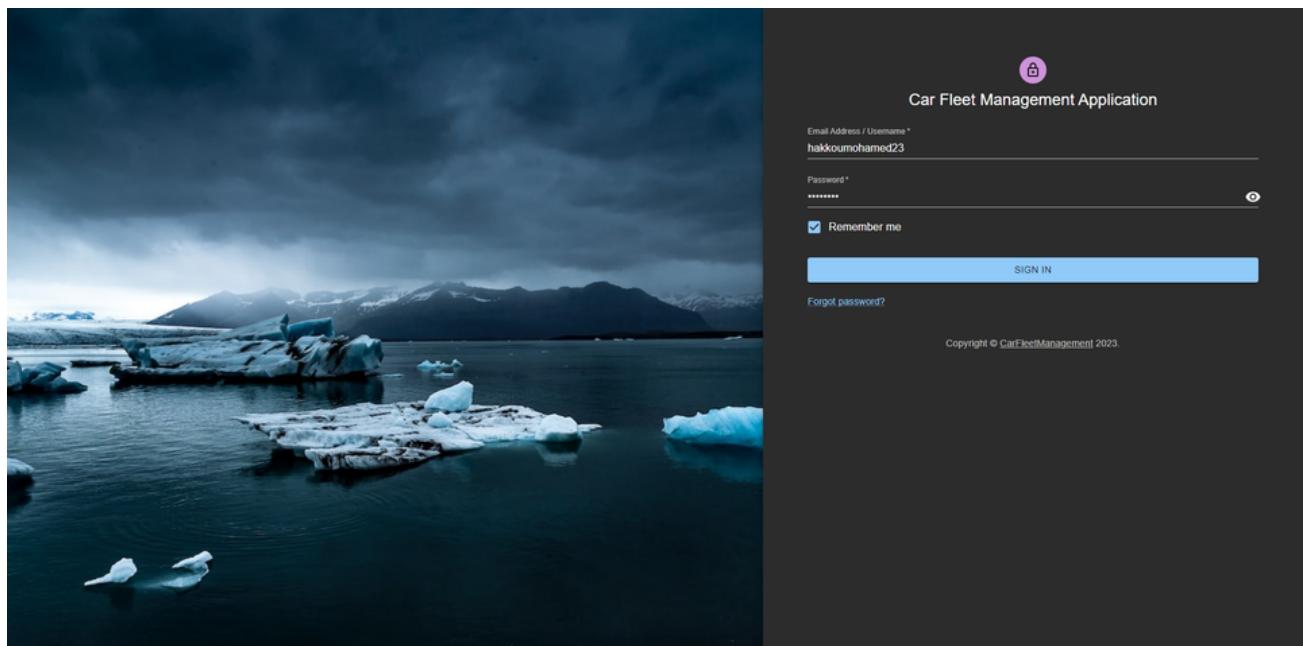


Figure 19: Login page

A typical login page consists of input fields for a username or email and password, along with a "Login" button. Users enter their credentials, and upon clicking "Login," the system verifies the information, granting access if valid or displaying an error message if not.

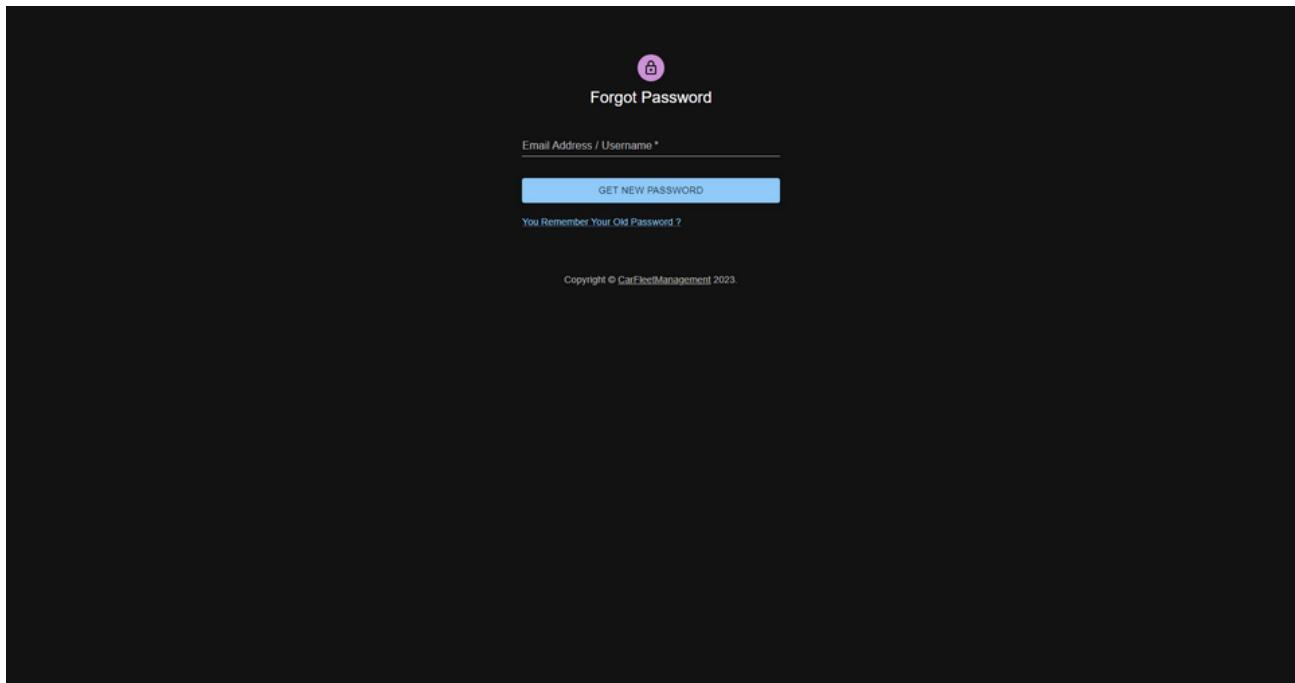


Figure 20: Restore password page

A password reset screen displays a form where users can enter their email address to receive the new password. It includes a "GET NEW PASSWORD" button for initiating the password recovery process.

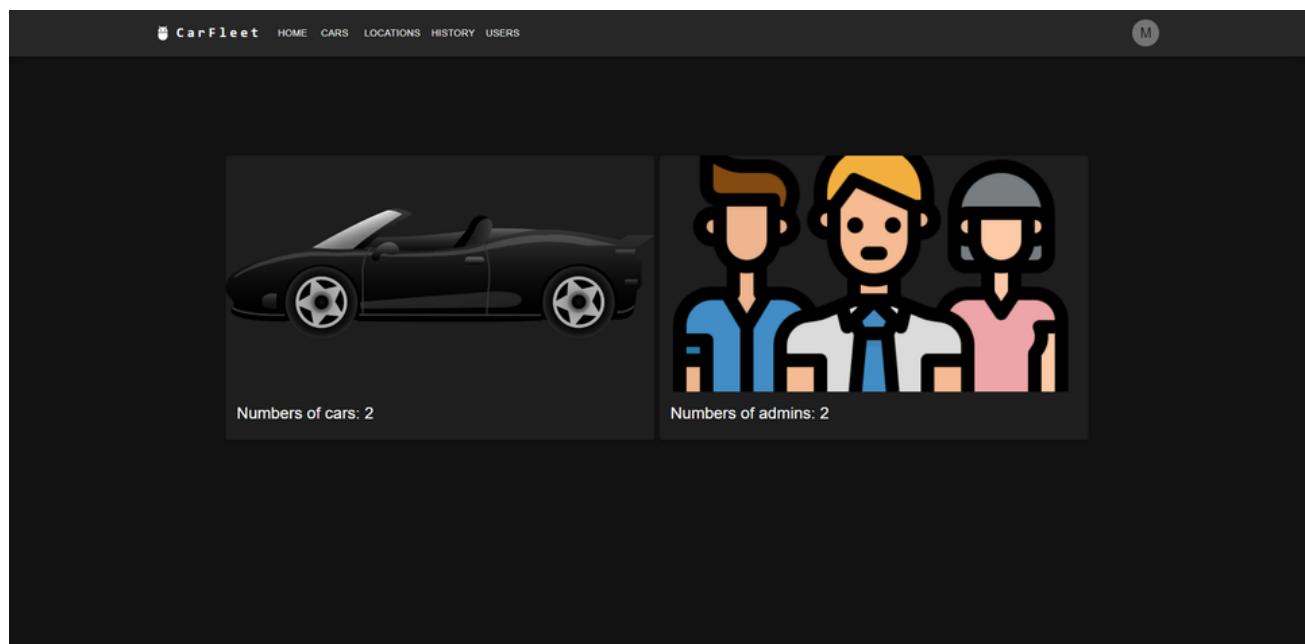


Figure 21: Home page

A homepage displays the website's main content and navigation, often featuring the logo and the menu at the top also cars and users numbers.

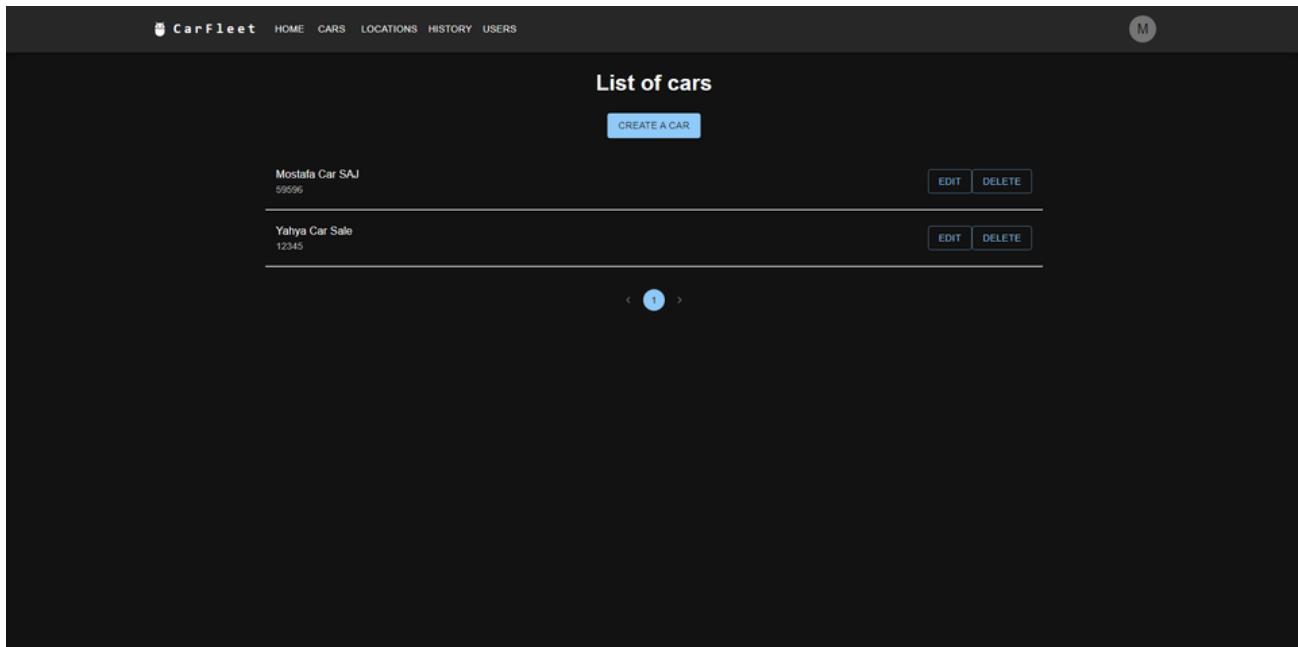


Figure 22: Manage cars page

A car list page displays a paginated list of vehicles, showing a limited number of cars per page providing functions on the buttons “Create A Car”, “Edit” ,”Delete”.

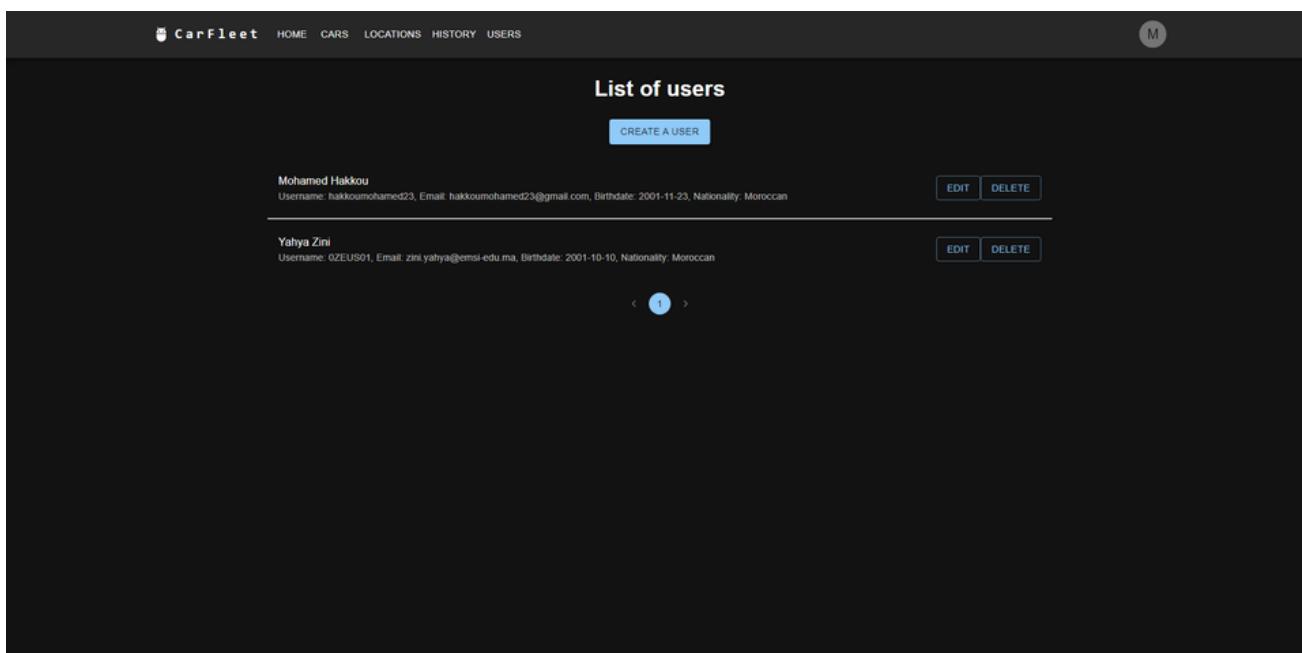


Figure 23: Manage users page

A user list page displays a list of user informations, including their names, usernames, emails, and other relevant information, allowing admins to browse and interact with the displayed user entries.

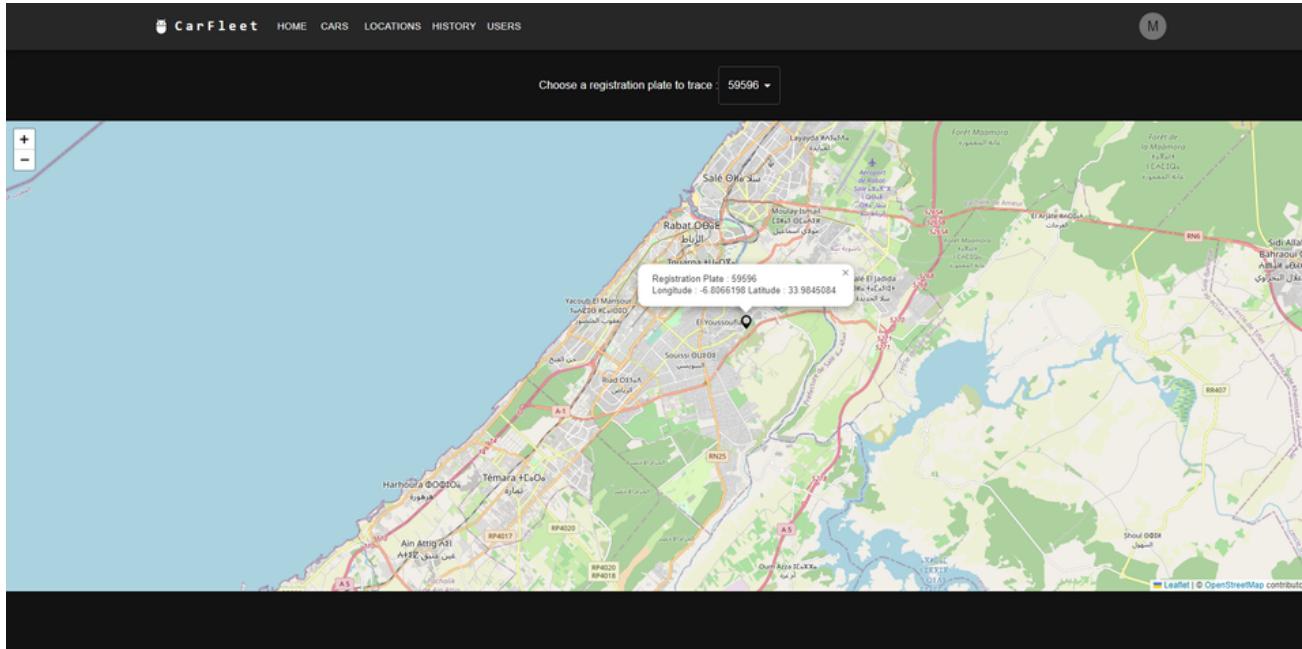


Figure 24: Real time car location page

A dynamic map displays real-time car locations, with markers updating as vehicles move, offering instant tracking and monitoring.

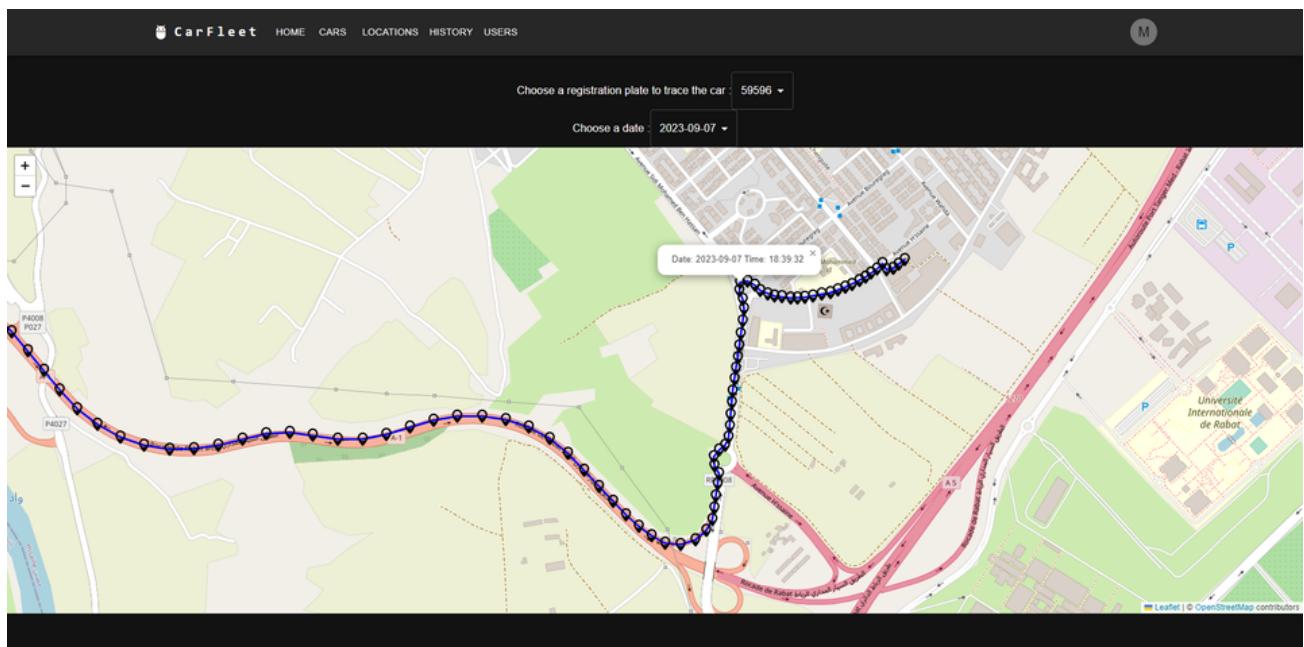


Figure 25: Location history page

A visual car location history page displays a timeline of past locations with timestamps, allowing users to track a vehicle's previous routes and stops.

CHAPTER 5:

Conclusion

In conclusion, our fleet management system consists of three essential components:

1. **Android App:** This application, after obtaining user location permission, allows users to input their registered vehicle's license plate. Once initiated, it continuously sends highly accurate location updates whenever a change occurs.
2. **Website:** The website complements the Android app with robust authentication features, including password reset functionality. It also provides administrators the tools to manage user accounts and vehicle data. Additionally, it offers real-time visual tracking of vehicle locations and access to historical location data.
3. **REST API:** Serving as the bridge between the front end and back end, the REST API facilitates seamless communication and data exchange between the Android app and the website, ensuring a smooth user experience and efficient data management.

Together, these components form a comprehensive fleet management system designed to meet the needs of tracking vehicles, managing users, and ensuring the security and functionality of the entire system.

WEBOGRAPHIE

- <https://linkedin.com/in/hichamboushaba/>
- <https://spring.io/projects/spring-boot>
- <https://react.dev/>
- <https://github.com/0ZEUS01/3EmsiStage>
- <https://www.eclipse.org/>
- <https://code.visualstudio.com/>
- <https://www.java.com/en/>
- <https://www.noip.com/?fpr=uxm8i>
- <https://play.google.com/store/apps/details?id=org.hicham.salaat&hl=en&gl=US>
- <https://play.google.com/store/apps/details?id=org.hicham.tasbihat&hl=en&gl=US>
- <https://woocommerce.com/mobile/>