

Software Engineering  
Software Requirements Specification  
(SRS) Document

CarMate

[9/22/23 ]

[Final Version ]

By: Nick Brandsma, Priyankaben Shah,  
Saniyah Khan

[UNCG Academic Integrity/ Honor Code: My words and actions will  
reflect Academic Integrity.

I will not cheat or lie or steal in academic matters.

I will promote integrity in the UNCG community.]

## Table of Contents

1. Introduction	5
1.1. Purpose	5
1.2. Document Conventions	5
1.3. Definitions, Acronyms, and Abbreviations	5
1.4. Intended Audience	6
1.5. Project Scope	7
1.6. Technology Challenges	8
1.7. References	8
2. General Description	
2.1. Product Perspective	9
2.2. Product Features	9
2.3. User Class and Characteristics	10
2.4. Operating Environment	10
2.5. Constraints	10
2.6. Assumptions and Dependencies	10
3. Functional Requirements	
3.1. Primary	10
3.2. Secondary	11
4. Technical Requirements	
4.1. Operating System and Compatibility	11
4.2. Interface Requirements	11
4.2.1. User Interfaces	11
4.2.2. Hardware Interfaces	11
4.2.3. Communications Interfaces	12
4.2.4. Software Interfaces	12

## 5. Non-Functional Requirements

5.1. Performance Requirements	12
5.2. Safety Requirements	12
5.3. Security Requirements	12
5.4. Software Quality Attributes	12
5.4.1. Availability	13
5.4.2. Correctness	13
5.4.3. Maintainability	13
5.4.4. Reusability	13
5.4.5. Portability	13
5.5. Process Requirements	13
5.5.1. Development Process Used	13
5.5.2. Time Constraints	13
5.5.3. Cost and Delivery Date	14
5.6. Other Requirements	14
5.7. Use-Case Model Diagram	14
5.8. Use-Case Model Descriptions	15
5.8.1. Actor: Sarah (Saniyah Khan)	15
5.8.2. Actor: Actor Name ( Priyankaben Shah)	16
5.8.3. Actor: Piedmont Dragway (Nick Brandsma)	17
5.9. Use-Case Model Scenarios	
5.9.1. Actor: Sarah (Saniyah Khan)	17
5.9.2. Actor: Piedmont Dragway (Nick Brandsma)	20
5.9.3. Actor: Actor Name (Priyankaben Shah)	21

## 6. Design Documents

6.1. Software Architecture	24
----------------------------	----

6.2. High-Level Database Schema	25
6.3. Software Design	
6.3.1. State Machine Diagram: Admin-Priyanka	26
6.3.2. State Machine Diagram: Saniyah-User	26
6.3.3. State Machine Diagram: Drag-Strip- Nich	27
6.4. UML Class Diagram	28
7. Scenario	
7.1. Brief Written Scenario with Screenshots	

## 1. Introduction

### 1.1. Purpose

The goal of the CarMate's application is to provide a safe and engaging platform for drag racing enthusiasts. It offers a variety of features including user registration, event hosting by drag strips, admin oversight, safety agreements, location-based services, and real-time communication. This application will help to keep street racers off the street and give many drag strips a greater chance at staying open.

### 1.2. Document Conventions

The purpose of this Software Requirements Document (SRD) is to describe the client-view and developer-view requirements. For the time period of this project, this will serve as a reference point in the many versions by providing details on the project's functions, data, scope, and expectations. In the document, the team will also provide users served by the system and limits to their profile base. Developer-oriented requirements outline software requirements and applications in detail for both the client and developer team to stay updated.

### 1.3. Definitions, Acronyms, and Abbreviations

Java	A programming language originally developed by James Gosling at Sun
Github	Cloud-based service for software development and version control to store and manage our code.
VS Code	Visual Studio Code. Source-code editor.
Bootstrap	Open-source CSS framework. This will help with front-end web development in using our application across different mobile devices.
.HTML	Hypertext Markup Language. This is the code that will be used to structure and design the web application and its content.
CSS	Cascading Style Sheets. A sheet language that is used for presenting documentation in HTML or XML.
API	Application Programming Interface. Where two or more computer programs can communicate with each other.
MongoDB	Easier for developers to store structured or unstructured data.

JavaScript	Programming language is going to be used for both client-side and server-side to make the webpage interactive.
------------	--

## 1.4. Intended Audience

### Stakeholders

1. Stakeholder
  - a. Client: Drag Strip Venues
  - b. Project Manager: Priyankaben Shah
  - c. Drag-Race Consultant: Nickolas Brandsma
  - d. Developers: Priyankaben Shah, Nickolas Brandsma, Saniyah Khan
  - e. Users: car enthusiasts using the application
2. The primary stakeholders that are the driving force behind this project. The stakeholders are drag-strip venues. Key sections of interest for the client include:
  - a. 1.1 - Purpose: Summarizes the project's goal and expected benefits for clients.
  - b. 1.2 - Document Conventions: Description of the main objectives of the Software Requirement Document (SRD) in the context of our application, CarMates.
  - c. 1.5 - Project Scope: Specifies the software goals with the overall business goals and the benefits of the project for the client and project manager to stay on track.
  - d. 1.6 - Technology Challenges: Technological constraints and performance expectations, for clients and developers to

### Developers

1. The development team is responsible for creating the software system. Sections relevant to developers include:
  - a. 2.3-User Class and Characteristics: Categorizing the users of the software is intended for

- b. 2.4-Operating Environment: The environment of the software that it is being designed to operate in
- c. 2.6-Assumptions and Dependencies- A list of assumptions regarding the software product and the environment along with any dependencies that can affect the project
- d. 3.1-Primary: All the requirements within the system or subsystem to determine the output that the software is expected to give in relation to the input
- e. 3.2-Secondary: Functions that are used to support the primary requirements
- f. 4.2-Interface Requirements
- g. 5.1- Performance Requirements: performance requirements for all the functional requirements

#### Project Managers

1. Project Managers oversee the entire project including planning, execution, and monitoring. Sections relevant to project managers include:
  - a. 2.1-Product Perspective: The origin of the product
  - b. 2.2-Product Features: An in-depth summary of the functions of the software that would perform and features included.
  - c. 2.5-Constraints: Limiting factors that pose a challenge to the development of the software
  - d. 4.1-Operating System and Compatibility: environments that will be needed to operate the system
  - e. 5.2-Safety Requirements: safeguards that need to be incorporated as a measure against the harm of the use of the software application may cause.
  - f. 5.3-Security Requirements: privacy and data protection regulations that are needed while designing the product

- g. 5.4-Software Quality Attributes: Additional qualities that need to be incorporated within the software
- h. 5.5-Process Requirements
- i. 5.6-Other Requirements

#### Users

1. System operators interact directly with the software system. They can view the system's features and functionalities from a user-centric view. Sections relevant to users include:
  - a.

### **1.5. Project Scope**

The goal of CarMate is to provide an easy-to-use interface for all users including, admins, drag strip accounts, and general users to interact and possibly attend events hosted by drag strips and monitored by admins. This software application aligns with car/drag race enthusiasts for their hobbies, and fostering a community, as well as allows drag strips to use their locations for enthusiasts that are not necessarily racing professionally.

The overall software goal instead of the business goal is to create a bridge for the drag racing community with each other while promoting safe and legal racing practices. This aligns with the overall client's business goals, as drag strip venues can promote events, gain sponsorship, and create awareness of their location to increase the visibility of location and participation; to align with monetization objectives. This application will also focus on allowing a general user-friendly experience with features such as live chat, preferred locations of users, and event notifications.

The successful development and implementation of CarMate will result in creating an application for the drag racing community to communicate with and increase participation in drag racing events whether it is as a racer or audience. For drag strip venues the app can make promotional events based on data received by the app's user preferences to ensure an increase in



revenue by reaching a targeted audience. Most importantly, CarMate will be able to promote a responsible racing culture to represent a positive image in the racing community.

### **1.6. Technology Challenges**

Some technology challenges throughout the project would be the live chat, as a possible language is javascript and react/redux for the backend and frontend. Though our team has some experience, it will be a learning experience. We have learned about API in class and implementation techniques. There are still other factors that we will look through when starting our implementation that we have yet to consider including finding an open API suitable for application.

### **1.7. References**

This section is subject to changes per the team's criteria.

API: <https://leafletjs.com/reference.html>

XAMPP: <https://www.apachefriends.org/>

Php: <https://www.php.net/>

Javascript program npm: <https://www.npmjs.com/>

W3schools bootstrap: <https://www.w3schools.com/bootstrap/>

Drag race directory: <http://speedwaysonline.com/tracks/category/dragstrips/>

## **2. General Description**

### **2.1. Product Perspective**

Illegal street racing and takeovers are a persistent problem that leads to accidents, injuries, and destruction of property. This application provides a safer alternative for members of the car community to race. It promotes legal drag racing at verified drag strips. It will allow users to easily connect with one another and facilitate racing at a drag strip as opposed to the street.

### **2.2. Product Features**

This application in its early phases will have admins to only allow valid users to prevent bot accounts, the admins will also be able to ban any user that does not meet guidelines. Another tier of accounts is drag strip venues for them to host and post-event notifications to either server. These accounts will also have to be approved by admins. Finally, there are the majority of accounts, the users who can interact in the app via creating a profile with the details of their vehicle. The team will also continue to discuss with clients on the posting feature and live chat but that will be updated in the early stages.

### **2.3. User Class and Characteristics**

Our application will be able to be used by anybody but for those who want to get the most out of it, the users will be ones who are a part of the local car community and are interested in drag racing their car. Other users who aren't interested in racing are still welcome and could use the app for notifications about events they would like to spectate. The users will not need to have any kind of special skill set to operate the app.

### **2.4. Operating Environment**

The application is designed to operate on the web across many different devices, with the database hosted on phpMyadmin

### **2.5. Constraints**

Limiting factors will be API and any additional features that could be added down the line.

### **2.6. Assumptions and Dependencies**

The software will be developed using VS Code and its go-live feature to stay updated on the look of the webpage. The API used at this stage will be a leaflet which is a maps API [<https://leafletjs.com/reference.html>] for a user to view nearby locations of venues. Some other features regarding the preferred location have yet to be finalized whether using live location via geo location or generally through an API with cities and states listed.

## **3. Functional Requirements**

### **3.1. Primary**

- FR0: The system will allow the user to pick which car they drive from a drop-down menu.
- FR1: The system will allow the user to enter the location they want to be associated with to find nearby dragstrips
- FR2: The system will allow the user to connect with other users to set up races at nearby dragstrips

### **3.2. Secondary**

[Some functions that are used to support the primary requirements.]

- Each user's information will be password protected when logging into their account
- Only authorized drag strip accounts will be able to edit the information for drag strips

## **4. Technical Requirements**

### **4.1. Operating System and Compatibility**

The application will be compatible with any operating system that can view and interact with traditional web pages.

### **4.2. Interface Requirements**

#### **4.2.1. User Interfaces**

[The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons, and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.]

Will be updated as the project progresses.

#### **4.2.2. Hardware Interfaces**

[All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.]

This application will run on any hardware device that has access to the internet, the ability to display webpages, and the ability to interact with web pages. This includes but is not limited to smartphones, tablets, desktop computers, and laptops.

### **4.2.3. Communications Interfaces**

[Determination of all the communication standards to be utilized by the software as a part of the project]

It must be able to connect to the internet as well as the local database on phpMyAdmin.

The communication protocol, HTTP, must be able to connect to the leaflet and show the maps API from the leaflet.js.

### **4.2.4. Software Interfaces**

[The interaction of the software to be developed with other software components such as frontend and the backend framework to be used, the database management system and libraries describing the need and the purpose behind each of them.]

We will use HTML, CSS, and JS to help build the frontend, as well as php for the backend database functionality. We will also use XAMPP to connect the front end to the backend.

## **5. Non-Functional Requirements**

[Constraints on the services or functions offered by the system (e.g., timing constraints, constraints on the development process, standards, etc.). Often apply to the system as a whole rather than individual features or services.]

### **5.1. Performance Requirements**

[The performance requirements need to be specified for all the functional requirements.]

- NFR0(R): The local copy of the vehicle violation database will consume less than 20 MB of memory
- NFR1(R): The system (including the local copy of the vehicle violation database) will consume less than 50MB of memory
- NFR2(R): The novice user will be able to create and print a ticket in less than 5 minutes.
- NFR3(R): The expert user will be able to create and print a ticket in less than 1 minute.

This will be updated as the project progresses.

## **5.2. Safety Requirements**

We will have a legal notice pop up for all users that will have them accept liability for the dangers that come with drag racing as well as stating that they will use this application for its intended purpose and not use it for street racing.

## **5.3. Security Requirements**

[Privacy and data protection regulations that need to be adhered to while designing the product.]

- NFR4(R): The system will only be usable by authorized users.

## **5.4. Software Quality Attributes**

[Detailing on the additional qualities that need to be incorporated within the software like maintainability, adaptability, flexibility, usability, reliability, portability, etc.]

### **5.4.1. Availability**

The application will be available to use 24/7.

### **5.4.2. Correctness**

[Details] - TBD

### **5.4.3. Maintainability**

The application will need the admins to maintain it by fixing bugs and adding new features and dragstrips that apply to be accepted.

### **5.4.4. Reusability**

This application can be used in different communities that share similar interests and altered to their needs as admins and users will be recyclable and accounts such as drag strip venues to be altered per the new project and client interests.

### **5.4.5. Portability**

This application will be able to be used on a website application.

## **5.5. Process Requirements**

API

XAMPP

phpMyAdmin

### **5.5.1. Development Process Used**

**[Software Process Model]**

**5.5.2. Time Constraints** – This section will be updated as the building starts.

### **5.5.3. Cost and Delivery Date**

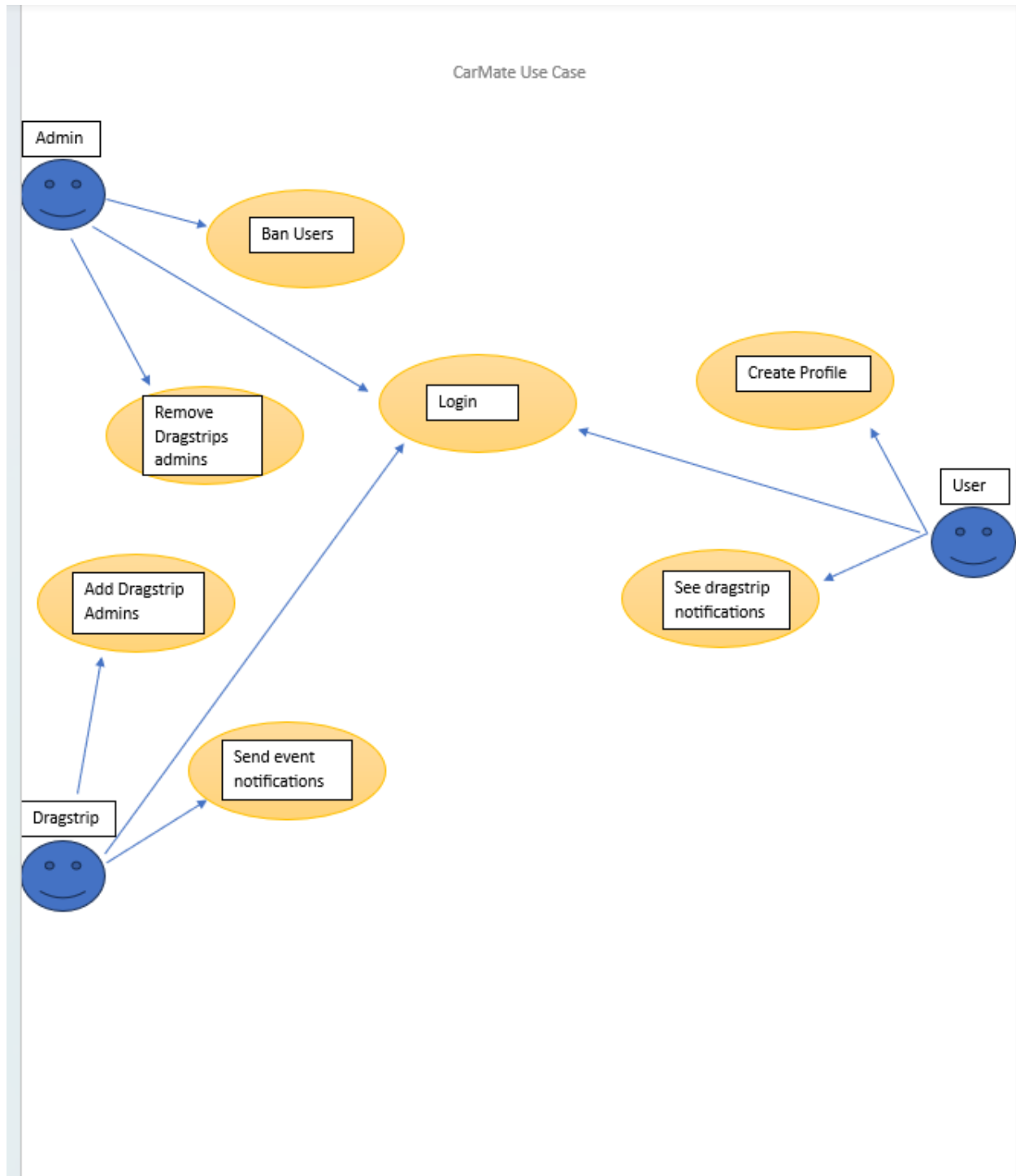
Delivery Date: December 5, 2023

Cost: A grade :)

### **5.6. Other Requirements**

This section will be updated as the building progresses.

### **5.7. Use-Case Model Diagram**



## 5.8. Use-Case Model Descriptions

### 5.8.1. Actor: Sarah (Saniyah Khan) = Admin

- **Use-Case Name:** Ban Users

The ability of authorized administrators of the web application to impose restrictions on specific users who violate any of the terms and conditions set by the web application. Examples of violating these terms can include, sending inappropriate messages, posting explicit content, etc. They will either not be able to use the web application for a duration of time or not be able to come back to the web application as a user.

**- Use-Case Name:** Add Dragstrip Permissions

The ability of authorized administrators of the web application to grant read and write permissions for the Dragstrip user to access and update information relating to the Dragstrip. Assign administrative permissions to specific users of the Dragstrip venue to manage the Dragstrip's event control, special events, event time, event security, etc.

**-User-Case Name:** Remove Dragstrips

The ability of authorized administrators of the web application to remove the dragstrip from the web application. The admin can remove the dragstrip. The user is met with a confirmation dialog displaying a message that the dragstrip is about to be removed from the application and data associated with it will not be accessible afterwards. The user can either select YES or NO.

**-User-Case Name:** Edit Dragstrip Information

The user with administrative rights will be able to modify or update information regarding the dragstrip. After they are done making the changes needed, they will need to confirm if they want to save the changes or not.

**-User-Case Name:** Login N



The administrator will enter a username and password to log in. They will need to have administrative-level credentials to successfully log in to the system.

**-User-Case Name:** Edit Profile

The administrator will need to have the correct credentials to edit the profiles of the users in the system.

### **5.8.2. Actor: Actor Name (Priyanka) - Users**

**- Use-Case Name:** Create Profile

Any user can create a profile with a nickname or username but also requires a birthdate to ensure age limitations of 18+ in this software. The user is then able to add their car to their profile to start connecting with others, this can include from just the model of the car to the specifications of their car.

**- Use-Case Name:** View dragstrip locations

Users can check for nearby drag strip venues for events or meetups to promote the venues and their events and allow for a safe space to essentially race legally.

**-User-Case Name:** Login

The administrator will enter a username and password to log in. They will need to have administrative-level credentials to successfully log in to the system.

**-User-Case Name:** Car Register

The user can register a car.

### **5.8.3. Actor: Piedmont Dragway (Nick Brandsma) - Drag strips**

**-User-Case Name:** Login

The administrator will enter a username and password to log in. They will need to have administrative-level credentials to successfully log in to the system.

**-User-Case Name:** Edit Profile/Drag Strip information

The administrator will need to have the correct credentials to edit the profiles of the users in the system. When the system sees that the user is a dragstrip account, they will be able to edit the information about the given dragstrip.

**- Use-Case Name:** Send event notifications

The administrator will be able to send out a mass notification to nearby users about upcoming events at their dragstrip.

**- Use-Case Name:** Add drag strip admins

The administrator will be able to look for and add other users without the dragstrip admin privileges to be linked with their account. That user account will then have administrator privileges that the Dragstrip account has.

## **5.9. Use-Case Model Scenarios**

### **5.9.1. Actor: Sarah (Saniyah Khan)-Admin**

**- Use-Case Name:** Removing Dragstrips

- Initial Assumption: The administrator can remove the dragstrip from the web application.
- Normal: The user with administrative privileges removes the dragstrip from the app.
- What Can Go Wrong: The user with administrator credentials accidentally removes a dragstrip that is still needed or in use. The user is met with a confirmation dialog displaying a message that the dragstrip is about to be removed from the application and data associated with it will not be accessible afterwards.
- Other Activities: The user can either click YES or NO on the dialogue box.
- System State on Completion: Depending on their action, the dragstrip is deleted or the dragstrip is not deleted.

**- Use-Case Name:** Add Dragstrip Permissions

- Initial Assumption: The process of granting and configuring specific access or privileges to users to edit Dragstrip-related actions.

- Normal: The user with administrator privileges can read, write, edit, delete, etc., dragstrip-related actions
- What Can Go Wrong: The user may not be of administrator level yet, which does not allow them to add dragstrip permissions
- Other Activities: The system will show a dialog box showing a warning or error message.
- System State on Completion: The user can contact the administrator if they need to correct the selection.

**- Use-Case Name: Ban Users**

- Initial Assumption: The ability of authorized administrators of the web application to impose restrictions on specific users who violate any of the terms and conditions set by the web application.
- Normal: The administrator can ban the user from the system.
- What Can Go Wrong: The selected user may not exist in the system.
- Other Activities: The system will provide an error message.
- System State on Completion: The administrator will need to correct their selection of the user they want to ban.

**- Use-Case Name: Edit Dragstrip Information**

- Initial Assumption: An authorized user will be able to modify and update details relating to dragstrip information.
- Normal: The user with administrative rights will be able to modify or update information regarding the dragstrip.
- What Can Go Wrong: The dragstrip that is selected may not exist in the system.
- Other Activities: The system will provide an error message stating that the dragstrip does not exist.

- System State on Completion: The user will need to correct their selection of the dragstrip information they tried to edit.

- **Use-Case Name:** Login

- Initial Assumption: The administrator has a registered account to log into the system. The account is saved in the database.
- Normal: The administrator will enter a username and password to log in.
- What Can Go Wrong: The administrator login credentials are not accepted as the username or password does not match the one saved in the database.
- Other Activities: The administrator can reset the password using the forgot password link
- System State on Completion: The administrator is logged in. They can view their dashboard.

- **Use-Case Name:** Edit Profile

- Initial Assumption: The administrator logs in and can edit user profiles on behalf of the users.
- Normal: The administrator can edit the profile information of the user such as editing their access level such as having administrator privileges for a dragstrip venue.
- What Can Go Wrong: The administrator is not able to find the user they searched for.
- Other Activities: The administrator will receive an error message from the system stating that the user cannot be found.
- System State on Completion: The administrator can search for the user again.

### 5.9.2. Drag Strip: Piedmont Dragway (Nick Brandsma)

- Use-Case Name: Log In

- Initial Assumption: The user will enter the correct information and will be allowed access to their account
- Normal: The user will gain access to their account after logging in

- What Can Go Wrong: The user enters the wrong information
- Other Activities: The user will be met with an error message
- System State on Completion: The user will be logged into their account
- Use-Case Name: Edit Dragstrip information
- Initial Assumption: The user will add their location and its information and it will then be discoverable by the normal users of the application
- Normal: The drag strip is added and then is discoverable by the other users
- What Can Go Wrong: The address/location of the dragstrip cannot be found or verified
- Other Activities: The dragstrip account is met with an error that the location cannot be found
- System State on Completion: A new dragstrip location is added and is now discoverable by the users
- Use-Case Name: Send event notifications
- Initial Assumption: The dragstrip account can send a mass notification to users in a certain area range notifying them of an event
- Normal: Users within the specified range of the drag strip get a notification of the event
- What Can Go Wrong: The server fails to be able to send out the notification
- Other Activities: The dragstrip account is met with an error that the notification was not sent out and should try again later
- System State on Completion: The notification is sent out notifying nearby users of the event
- Use-Case Name: Add dragstrip admins
- Initial Assumption: The user will choose another account they wish to be linked with the Dragstrip account to have access to the admin privileges
- Normal: The other account will now have admin privileges
- What Can Go Wrong: The account they are looking for is not able to be found
- Other Activities: The dragstrip account is met with an error that the account cannot be found

- **System State on Completion:** A new account is now added that can have the same privileges as the drag strip

### 5.9.3. Actor: Actor Name (Priyankaben Shah)

#### - Use-Case Name: Login

- **Initial Assumption:** The administrator has a registered account to log into the system. The account is saved in the database.
- **Normal:** The administrator will enter a username and password to log in.
- **What Can Go Wrong:** The administrator login credentials are not accepted as the username or password does not match the one saved in the database.
- **Other Activities:** The administrator can reset the password using the forgot password link
- **System State on Completion:** The administrator is logged in. They can view their dashboard.

#### - Use-Case Name: Edit Profile

- **Initial Assumption:** The administrator inins and can edit user profiles on behalf of the users.
- **Normal:** The administrator can edit the profile information of the user such as editing their access level such as having administrator privileges for a dragstrip venue.
- **What Can Go Wrong:** The administrator is not able to find the user they searched for.
- **Other Activities:** The administrator will receive an error message from the system stating that the user cannot be found.
- **System State on Completion:** The administrator can search for the user again.

#### - Use-Case Name: Create Profile

- **Initial Assumption:** Users can create a profile by signing up.

- Normal: The user creates a profile after verifying the email and creating a username, password, and birthdate.
- What Can Go Wrong: During this time when a new profile is created, data cannot be saved for the next login or does not check for a valid email address.
- Other Activities: The age limit needs to be verified as this app is for 18+ users.
- System State on Completion: Users will be able to create a profile and log back in successfully.

- **Use-Case Name:** View dragstrip locations - API = leaflet

- Initial Assumption: Dragstrip venues create their profile for users to find nearby locations.
- Normal: Dragstrip profiles also have their location tag on for users to find their Location.
- What Can Go Wrong: Location does not sync and prevents nearby users from locating them.
- Other Activities: Users can check their stats/information on location/
- System State on Completion: Dragstrip accounts can show their location to nearby users.

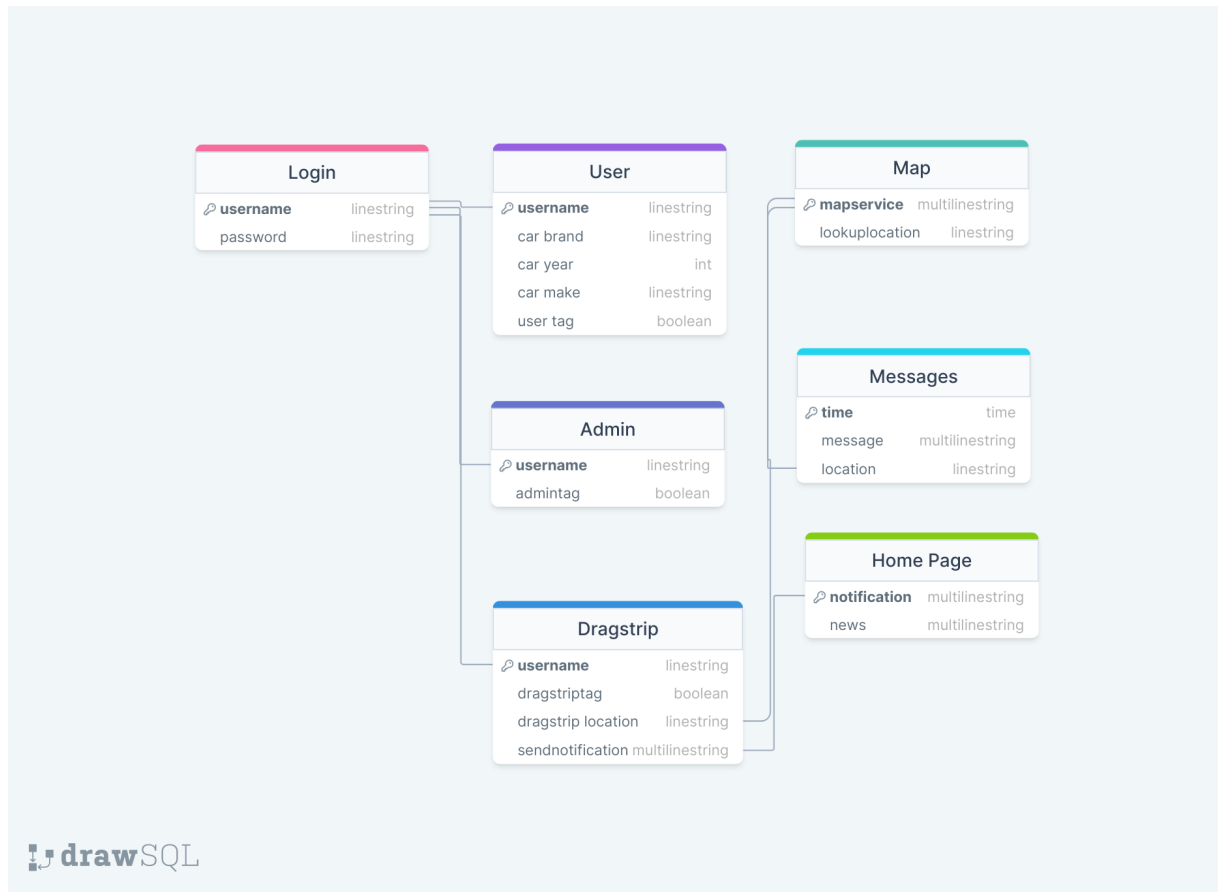
## 6. Design Documents

### 6.1. Software Architecture

<b>MODEL</b>	<b>User</b> <ul style="list-style-type: none"> <li>- Stores user registration detail</li> <li>- Profile data is stored</li> <li>- Message functionality</li> </ul>	<b>Admin</b> <ul style="list-style-type: none"> <li>- User credentials</li> <li>- Ban user actions stored</li> <li>- Edit user actions stored</li> <li>- Message functionality</li> </ul>	<b>Drag-strip</b> <ul style="list-style-type: none"> <li>- User credentials</li> <li>- Message functionality</li> <li>- Locations data stored</li> <li>- Notification actions stored</li> </ul>
<b>VIEW</b>	<b>User</b> <ul style="list-style-type: none"> <li>- Base case model of UI</li> <li>- User Dashboard</li> <li>- Profile/Message/Map</li> </ul>	<b>Admin(is_admin())</b> <ul style="list-style-type: none"> <li>- Admin Dashboard(Ban users)</li> <li>- Message</li> <li>- Map</li> </ul>	<b>Drag-Strip(is_DS())</b> <ul style="list-style-type: none"> <li>- Drag-strip dashboard</li> <li>- Notification to server</li> <li>- Map(edit location)</li> <li>- Message</li> </ul>
<b>CONTROLLER</b>	<ul style="list-style-type: none"> <li>- Manage user and drag-strip account authentication, registration, and profile setting(MyPhp)</li> <li>- Facilitate connection between two users in live chat, and maintain messaging functionality</li> <li>- Maintain database with login credentials via hashmap</li> <li>- Update real time notifications to server and stores data about notification</li> <li>- Maintain data for location of drag-strip venues (leaflet.js)</li> <li>- Records admin actions</li> <li>- API uses for user profile settings stored</li> <li>-</li> </ul>		

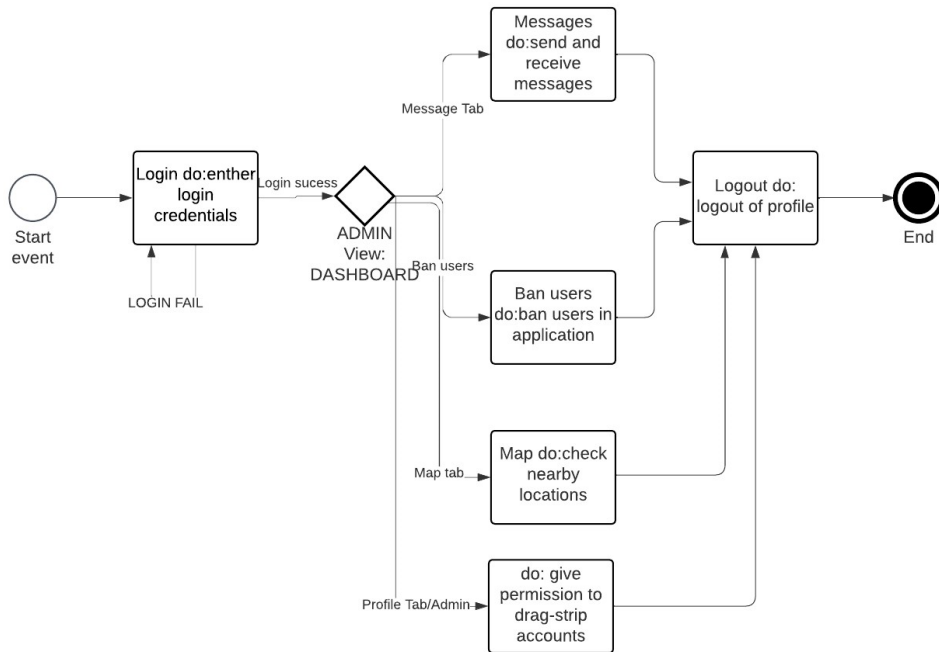
## 6.2. High-Level Database Schema



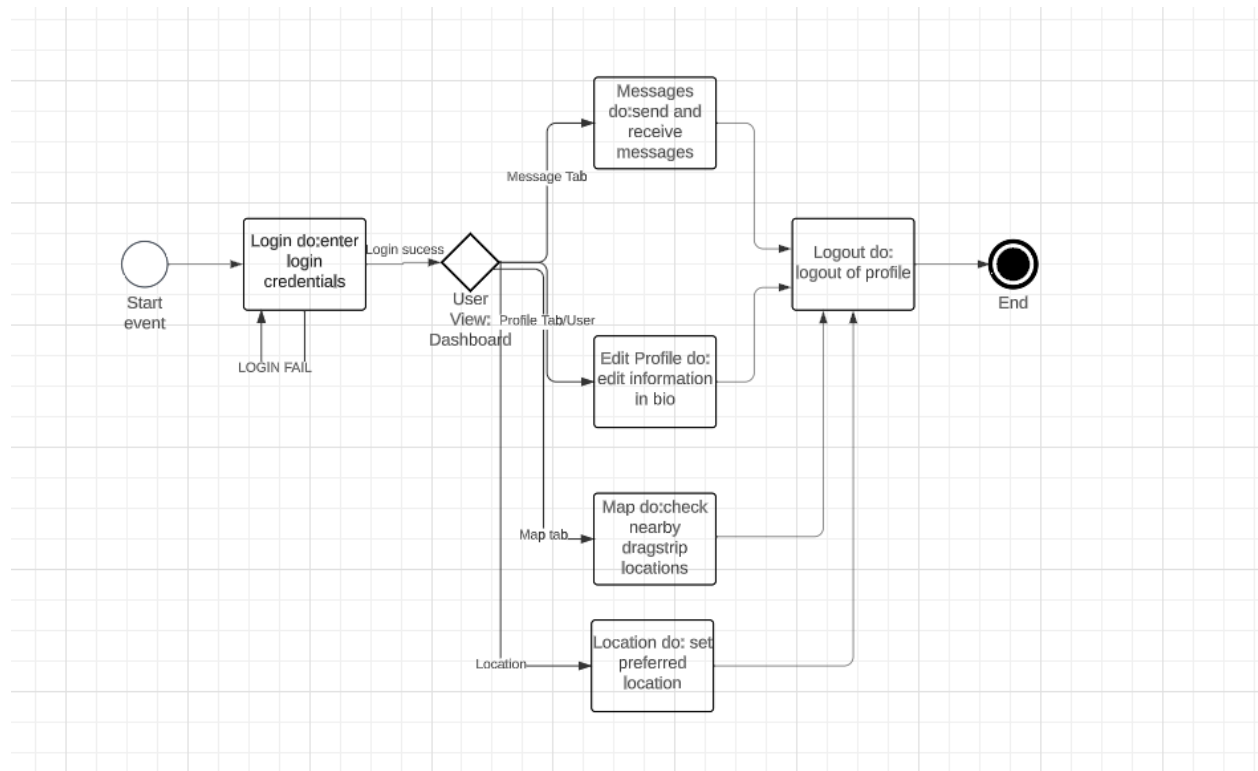


### 6.3. Software Design

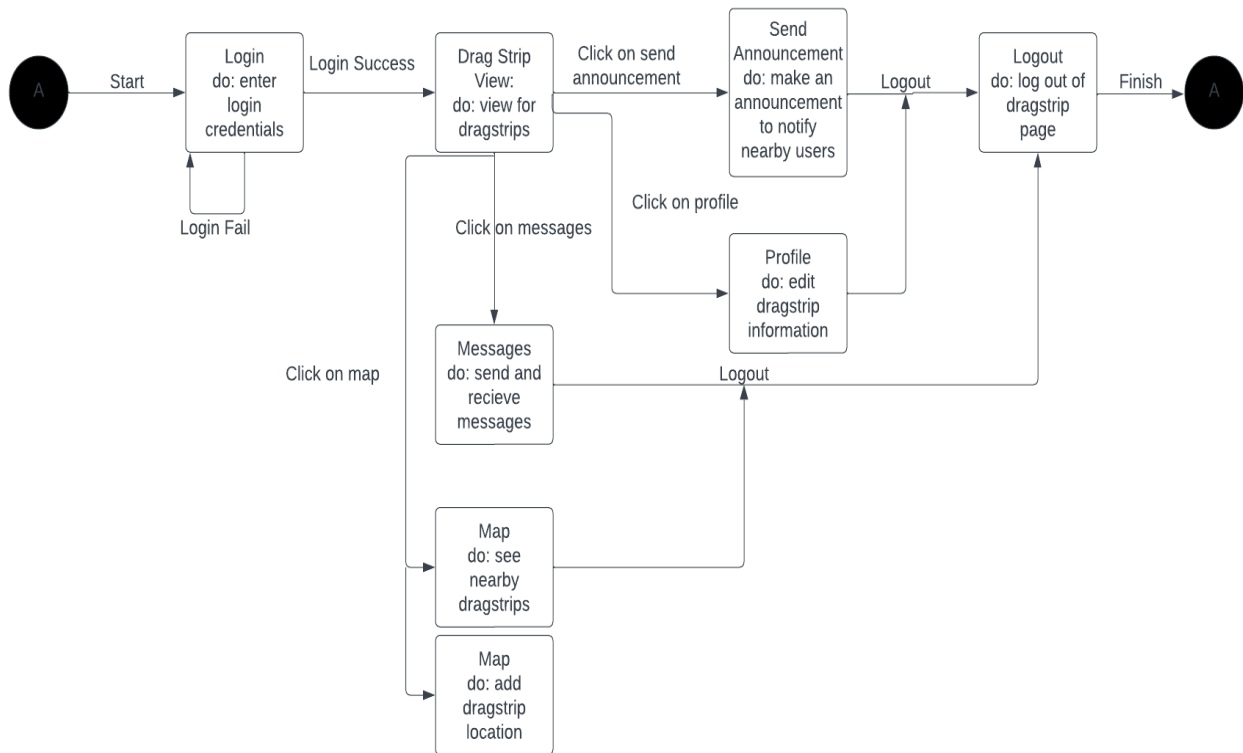
### 6.3.1. State Machine Diagram: Admin (Priyanka)



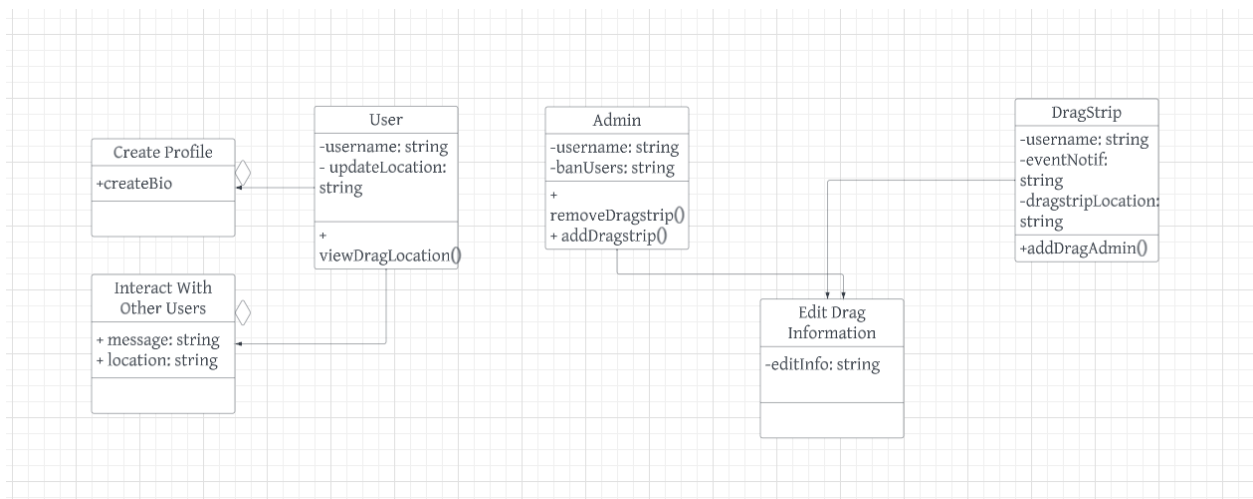
### 6.3.2. State Machine Diagram: User (Saniyah)



### 6.3.3. State Machine Diagram: Drag-strip (Nick)



### 6.4. UML Class Diagram



## 7. Scenario

Written Scenario

User: Write reviews use case:

The user creates a profile by registering.

The user then logs in and can enter the application.

Users can see dragstrip notifications.

Users can register a vehicle with a form.

Dragstrip: logs in for the first time and creates a profile

Dragstrip logs in.

Dragstrip can add admins from their venue.

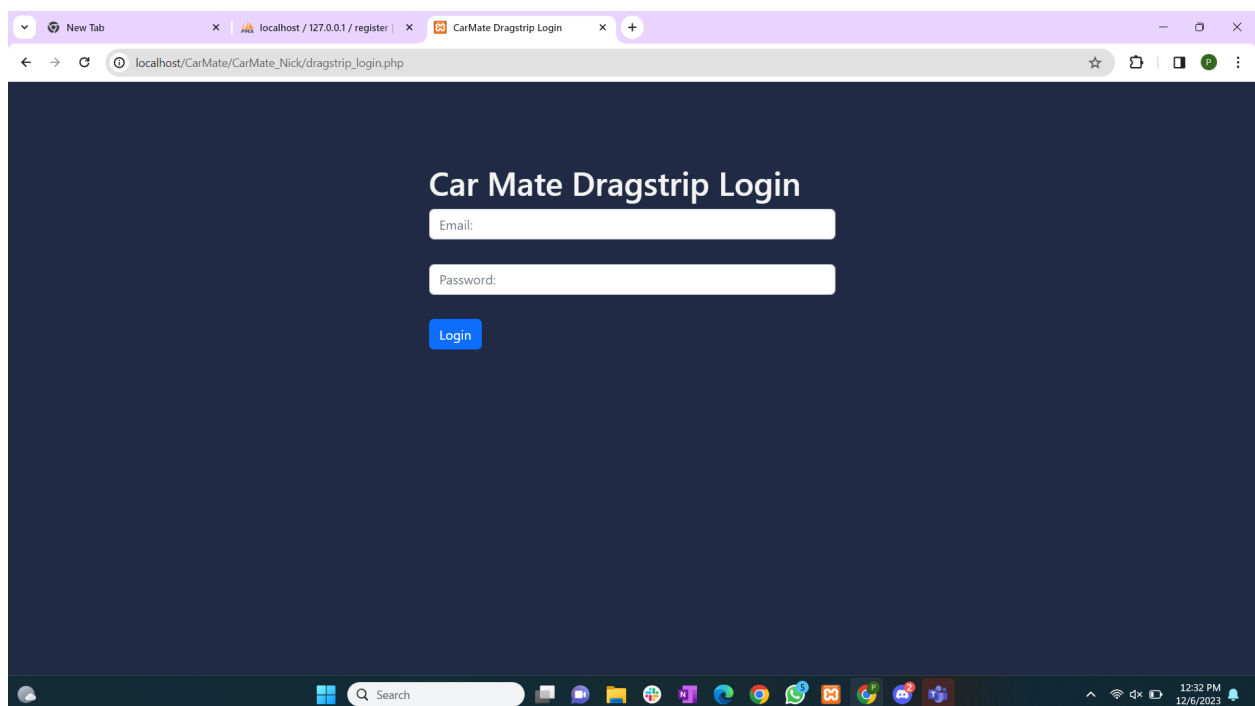
Dragstrip can add notifications for the server to see.

Admin: logs in

Admin enters their dashboard and can ban any account

Admin edits any profile type

If the user logs in they can not re-enter.



The screenshot shows the phpMyAdmin web interface in a browser. The address bar indicates the URL is `localhost/phpmyadmin/index.php?route=/database/structure&db=register`. The interface is for the 'register' database, which is located under the 'mysql' server. The left sidebar shows the database structure tree. The main panel displays the 'Structure' tab for the 'register' database. It lists 6 tables: `admindash`, `dragstrip_login`, `drag_admins`, `drag_events`, `form`, and `users`. Each table has a 'Browse' button and a 'Structure' button. The 'Structure' button is highlighted for the `users` table. Below the table list, there is a 'Create new table' section with a 'Table name' input field and a 'Number of columns' input field set to 4. The 'Create' button is visible. The bottom of the screen shows the Windows taskbar with the date and time 12:36 PM 12/6/2023.

Server: 127.0.0.1 » Database: register

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> <code>admindash</code>		3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> <code>dragstrip_login</code>		0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> <code>drag_admins</code>		3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> <code>drag_events</code>		2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> <code>form</code>		3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> <code>users</code>		5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
6 tables	Sum	16	InnoDB	utf8mb4_general_ci	96.0 KiB	0 B

☐ Check all With selected:

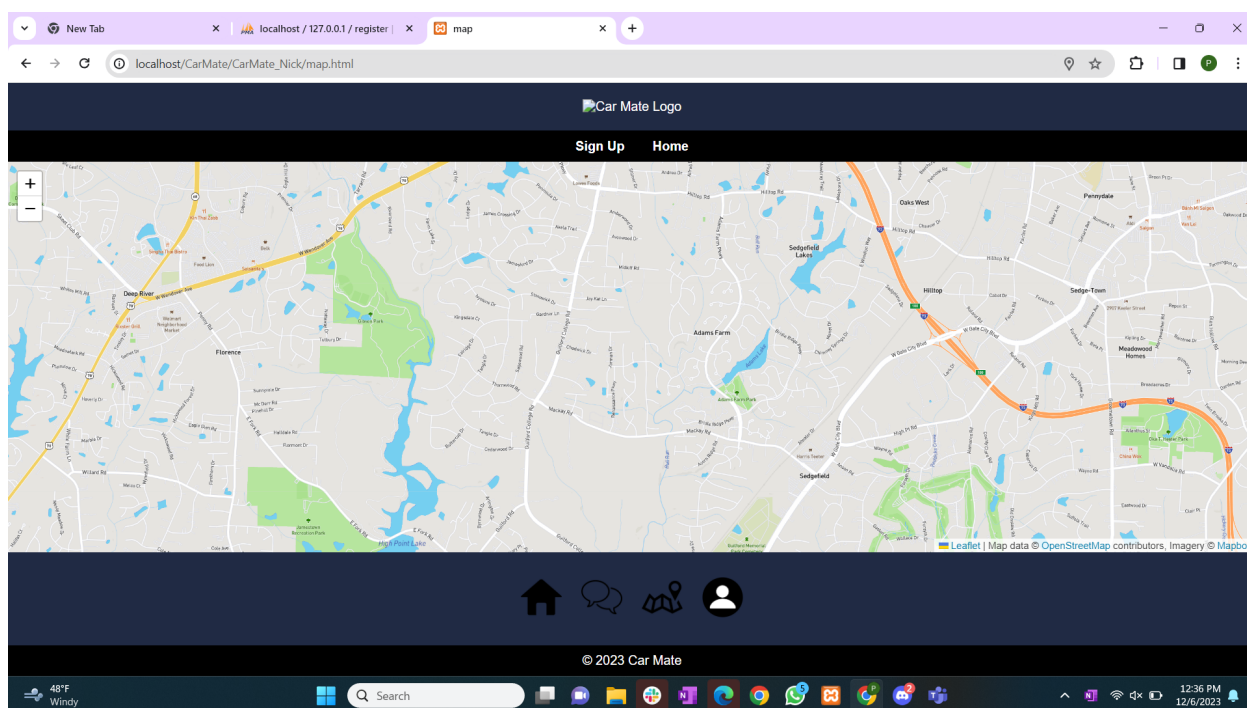
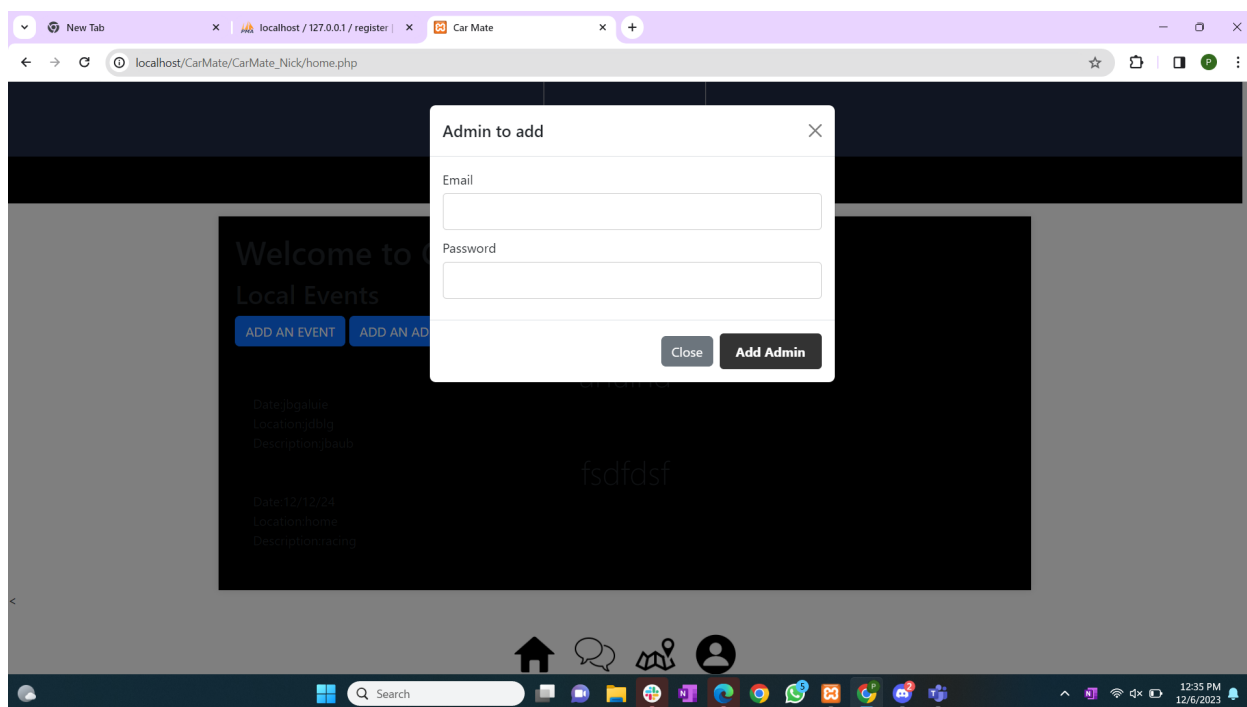
Print Data dictionary

Create new table

Table name:  Number of columns:

Console

48°F Windy 12:36 PM 12/6/2023



localhost / 127.0.0.1 / register | x Setting Form x +

localhost/CarMate/CarMate\_Nick/form.html

## Car Profile

Car Make:  
Acura

Car Type:  
Hybrids/EVs

Car Year:  
2023

Car Color:  
Red

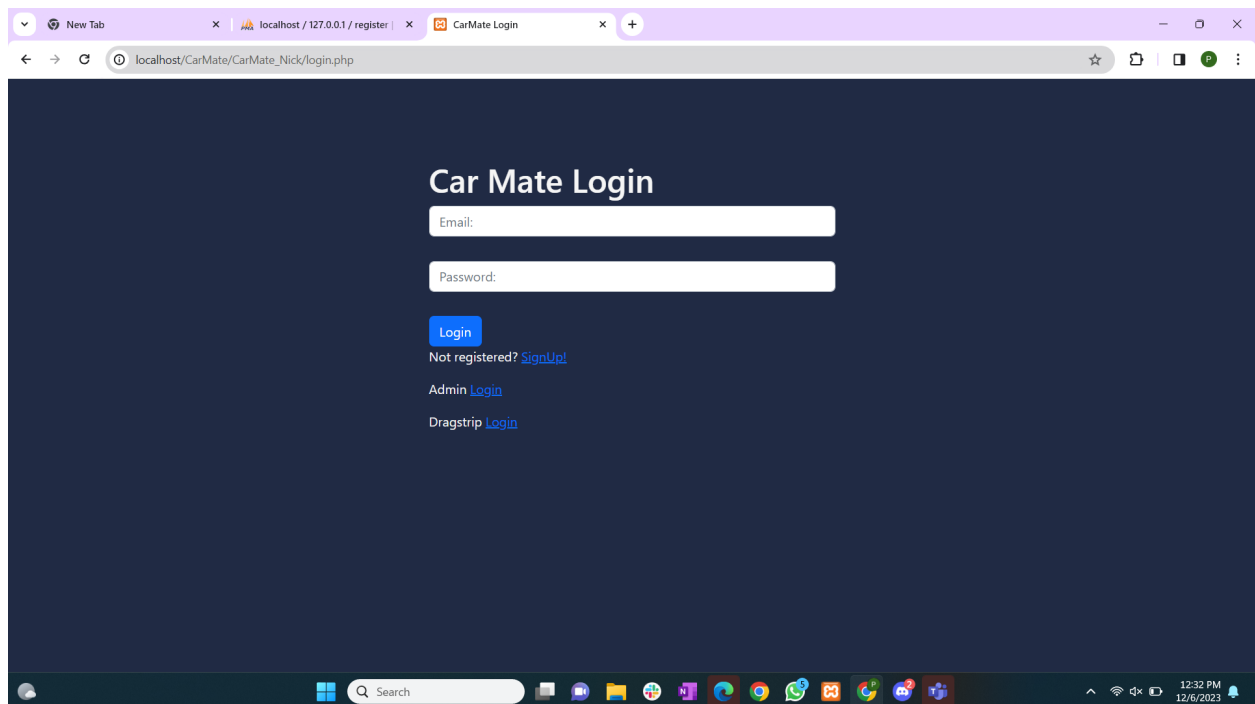
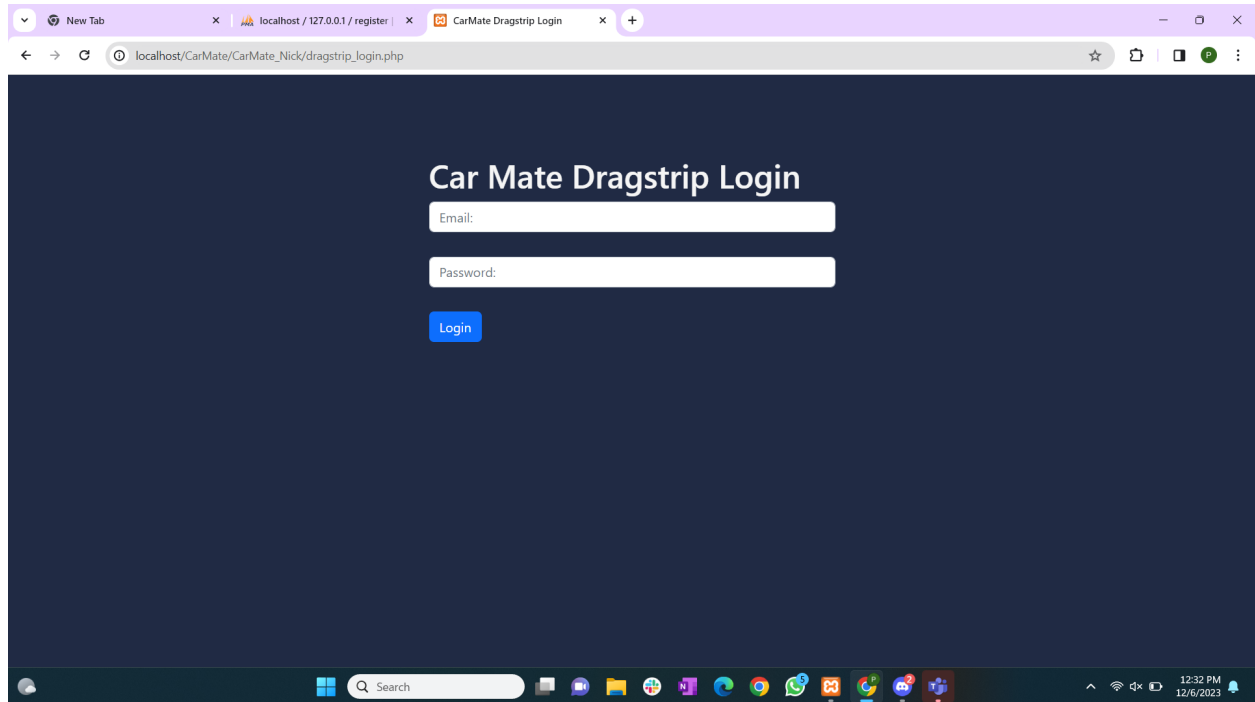
Car Modification:

Submit

Home Chat Map Profile

© 2023 Car Mate

48°F Windy Search 12:36 PM 12/6/2023





New Tab | localhost / 127.0.0.1 / register | CarMate SignUp

localhost/CarMate/CarMate\_Nick/register.php

## Sign Up for Car Mate

Name:

Email:

Password:

Repeat Password:

[Login](#)

12:32 PM 12/6/2023

localhost / CarMate / CarMate\_Nick / admindash.php

Name  
Enter your name

Email  
Enter your email

Password  
Enter your password

Submit

localhost / CarMate / CarMate\_Nick / display.php

Add user

ID	Name	Email	Password	Operations
2	obama	obame@example.com	1234test	<a href="#">Update</a> <a href="#">Delete</a> <a href="#">Unban</a>
3	fjbJ	fdini@exmple.com	test1234	<a href="#">Update</a> <a href="#">Delete</a> <a href="#">Unban</a>
5	name	name@email.com	test1234	<a href="#">Update</a> <a href="#">Delete</a> <a href="#">Ban</a>

