# # CAR RENTAL SYSTEM

The Car Rental System is a Python-based command-line application designed to manage car rental operations efficiently. It features a role-based access system with three user types: Administrators who manage the entire system, Staff who handle daily rental operations, and Customers who make booking requests. The system provides comprehensive functionality for managing cars, customers, and bookings, including features for adding and tracking vehicles, processing rental requests, managing customer information, and maintaining rental history. Built with SQLite database for data storage, it ensures reliable performance while maintaining data integrity through proper validation and error handling.

## READ ME

This is a simple Car Rental Management System developed in Python.

**## Features**

- User authentication (Admin/Staff)

- Car management (Add, Edit Rate, Delete, View)

- Customer management (Add, Edit, Delete, View)

- Booking Management System

  - Create new bookings

  - View pending bookings

  - Approve/Reject bookings

  - View booking history

- Rent car and log rentals

- View rental history

- Role-based access control (Admin/Staff/Customer permissions)

**## Setup**

1. Clone the repository.

2. Ensure you have Python installed.

3. Install required libraries:

   ```bash

   pip install termcolor

   pip install prettytable

   ```

*\*(Note: msvcrt is Windows-specific. For other OS, you might need a different library or approach for key press detection.)\**

4. Run the `database.py` script to set up the database tables.

     db has default 2 user:

     admin/admin

     staff/staff

**## Running the Application**

Run the `main.py` script:

```bash

python main.py

```

Follow the on-screen prompts to login as an admin or staff member and manage the car rental system.

**## Version Control**

This project is under version control. Commits reflect changes made during development, including bug fixes and feature implementations.

**\*\*Last Updated:\*\*** 2025-05-30

**## Progress**

**\*\*Update for 2025-05-30:\*\***

- Implemented Booking Management System

  - Added booking creation functionality

  - Implemented booking approval process

  - Added booking rejection with remarks

  - Created booking history view

  - Enhanced booking validation

- Improved Database Operations

  - Added proper transaction handling

  - Enhanced error handling for database operations

  - Improved data consistency checks

- UI/UX Improvements

  - Added emoji indicators for better user feedback

  - Enhanced error messages

  - Improved menu organization

**\*\*Update for 2025-05-28:\*\***

- Enhanced Data Management

  - Added timestamp tracking for all records

  - Updated\_at field now properly updates on all modifications

  - Improved data integrity with proper timestamp handling

- Improved Customer Management

  - Added validation to prevent deletion of customers on rent

  - Enhanced email validation to prevent duplicates

  - Added proper error handling for customer operations

- Code Improvements

  - Refactored database operations for better consistency

  - Improved error messages and user feedback

  - Enhanced validation logic across all operations

- Staff Page Improvements

  - Completed staff functionality

**\*\*Update for 2025-05-27:\*\***

- Implemented Customer Management

  - Add new customers

  - Edit customer details

  - Delete customers

  - View all customers

  - View customers on rent

- Implemented Renting a Car

  - License validation

  - Car availability check

  - Rental duration and cost calculation

  - Status updates for cars and customers

- Implemented role-based access control

  - Admin: Full access to all features

  - Staff: Limited access to certain features

**\*\*Update for 2024-05-26:\*\***

- Initial Project Setup

  - Created basic project structure

  - Set up database schema

  - Implemented user authentication system

  - Added basic car management features

  - Created initial documentation

## USER DOCUMENTATION

**# Car Rental System - User Documentation**

**## Table of Contents**

1. [Getting Started](#getting-started)

2. [User Roles](#user-roles)

3. [Admin Features](#admin-features)

4. [Staff Features](#staff-features)

5. [Customer Features](#customer-features)

6. [Common Operations](#common-operations)

7. [Troubleshooting](#troubleshooting)

**## Getting Started**

**### System Requirements**

- Python 3.x installed

- Required Python packages:

  - termcolor

  - prettytable

**### Installation**

1. Clone the repository

2. Install required packages:

   ```bash

   pip install termcolor

   pip install prettytable

   ```

3. Run the database setup:

   ```bash

   python database.py

   ```

4. Start the application:

   ```bash

   python main.py

   ```

**### Default Users**

- Admin: username: `admin`, password: `admin`

- Staff: username: `staff`, password: `staff`

**## User Roles**

**### Admin**

- Full system access

- Can manage all users, cars, customers, and bookings

- Can approve/reject bookings

- Can view all system reports

**### Staff**

- Limited system access

- Can view cars and customers

- Can process rentals

- Can view booking requests

**### Customer**

- Can view available cars

- Can make booking requests

- Can view their booking history

- Can view their rental status

**## Admin Features**

**### User Management**

1. View All Users

   - Access: Admin Dashboard → Manage Staff

   - Shows all system users with their details

2. Add User

   - Access: Admin Dashboard → Manage Staff

   - Required fields:

     - Username

     - Password

     - Role (admin/staff)

     - Name

     - Email

     - Phone

3. Edit User

   - Access: Admin Dashboard → Manage Staff

   - Can modify:

     - Password

     - Role (except admin)

     - Email

     - Phone

4. Delete User

   - Access: Admin Dashboard → Manage Staff

   - Cannot delete admin user

   - Requires email confirmation

**### Car Management**

1. View All Cars

   - Access: Admin Dashboard → Manage Cars

   - Shows all cars with their details

2. Add Car

   - Access: Admin Dashboard → Manage Cars

   - Required fields:

     - Make

     - Model

     - Year

     - Rate per day

     - Registration number

3. Edit Car

   - Access: Admin Dashboard → Manage Cars

   - Can modify rate per day

   - Cannot edit rented cars

4. Delete Car

   - Access: Admin Dashboard → Manage Cars

   - Cannot delete rented cars

**### Customer Management**

1. View All Customers

   - Access: Admin Dashboard → Manage Customer

   - Shows all customers with their details

2. Add Customer

   - Access: Admin Dashboard → Manage Customer

   - Required fields:

     - Name

     - Phone

     - Email

     - Address

     - License number

     - License expiry date

3. Edit Customer

   - Access: Admin Dashboard → Manage Customer

   - Can modify:

     - Phone

     - Email

     - Address

     - License expiry date

4. Delete Customer

   - Access: Admin Dashboard → Manage Customer

   - Cannot delete customers on rent

**### Booking Management**

1. View Pending Bookings

   - Access: Admin Dashboard → Manage Bookings

   - Shows all pending booking requests

2. Approve Booking

   - Access: Admin Dashboard → Manage Bookings

   - Steps:

     1. Select booking ID

     2. System validates:

        - Car availability

        - Customer license validity

        - Booking dates

     3. Confirm approval

3. Reject Booking

   - Access: Admin Dashboard → Manage Bookings

   - Steps:

     1. Select booking ID

     2. Enter rejection reason

     3. Confirm rejection

4. View All Bookings

   - Access: Admin Dashboard → Manage Bookings

   - Shows all bookings with their status

**## Staff Features**

**### Car Operations**

1. View Available Cars

   - Access: Staff Dashboard → Manage Cars

   - Shows all available cars

2. View Rented Cars

   - Access: Staff Dashboard → Manage Cars

   - Shows all currently rented cars

**### Customer Operations**

1. View All Customers

   - Access: Staff Dashboard → Manage Customer

   - Shows all customers

2. View Customers on Rent

   - Access: Staff Dashboard → Manage Customer

   - Shows customers with active rentals

**### Rental Operations**

1. Rent a Car

   - Access: Staff Dashboard → Rent a Car

   - Steps:

     1. Enter customer license number

     2. Select car

     3. Enter rental duration

     4. Confirm rental

2. Return a Car

   - Access: Staff Dashboard → Car Return

   - Steps:

     1. Enter customer license number

     2. Enter car plate number

     3. Add remarks

     4. Confirm return

**## Common Operations**

**### Viewing Rental History**

1. Access: Admin/Staff Dashboard → View Rent History

2. Shows all rental records with:

   - Customer details

   - Car details

   - Rental dates

   - Cost

   - Status

**### Making a Booking (Customer)**

1. Access: Customer Dashboard → Make Booking

2. Steps:

   1. View available cars

   2. Select car

   3. Enter start and end dates

   4. Confirm booking

3. Wait for admin approval

**## Troubleshooting**

**### Common Issues**

1. "Booking not found"

   - Check if booking ID is correct

   - Verify booking status is 'pending'

2. "Car not available"

   - Check car status

   - Verify booking dates

3. "Customer already on rent"

   - Check customer's current rental status

   - Process any pending returns

4. "License expired"

   - Check customer's license expiry date

   - Update license information if needed

**### Error Messages**

- ❌ Red messages indicate errors

- ✅ Green messages indicate success

- ⚠️ Yellow messages indicate warnings

**### Getting Help**

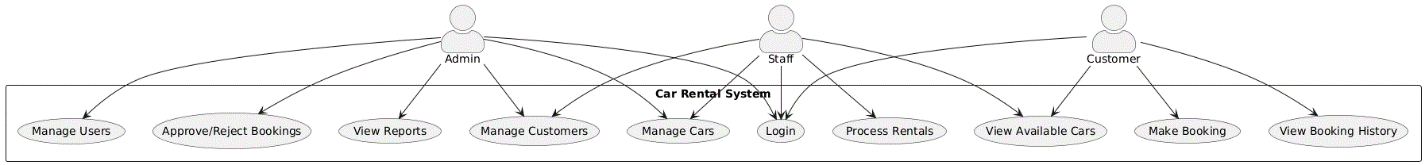
- Contact system administrator for technical issues

- Check the README.md for system updates

- Review this documentation for common procedures

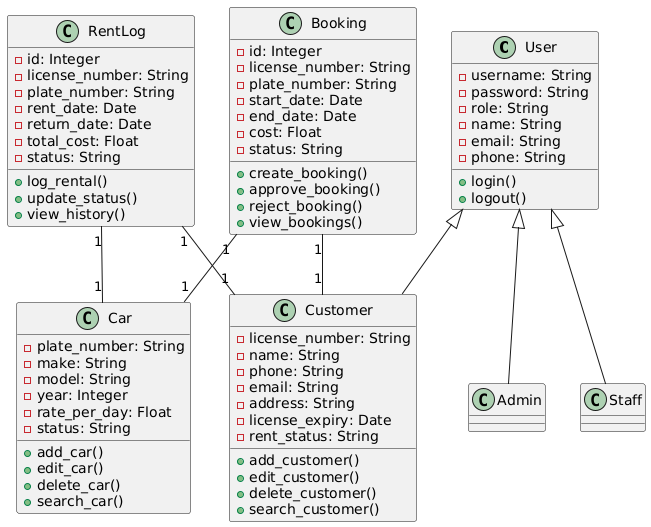
## DIAGRAMS

### Use Case Diagram



The Admin has the most extensive access, with capabilities to manage users, approve or reject bookings, view reports, manage customers, and manage cars. The Staff role focuses on operational tasks such as logging into the system, managing cars and customers, and processing rentals. Meanwhile, the Customer can interact with the system by logging in, viewing available cars, making bookings, and viewing their booking history. Each actor is linked to specific system functions, illustrating their respective permissions and responsibilities within the system. This diagram provides a clear overview of how different users interact with the system and helps guide the system’s functional design.

### Class Diagram



This defines six main classes: User, Customer, Car, Booking, RentLog, and two specialized subclasses — Admin and Staff.

The **User** class contains attributes like username, password, role, name, email, and phone, with methods for login() and logout(). Both **Admin** and **Staff** inherit from this class, suggesting they share common functionalities like authentication.

The **Customer** class includes personal and license details such as license\_number, name, phone, email, address, license\_expiry, and rent\_status, with methods to manage customer records.

The **Car** class defines each car by plate\_number, make, model, year, rate\_per\_day, and status. It includes methods to add, edit, delete, and search for cars.

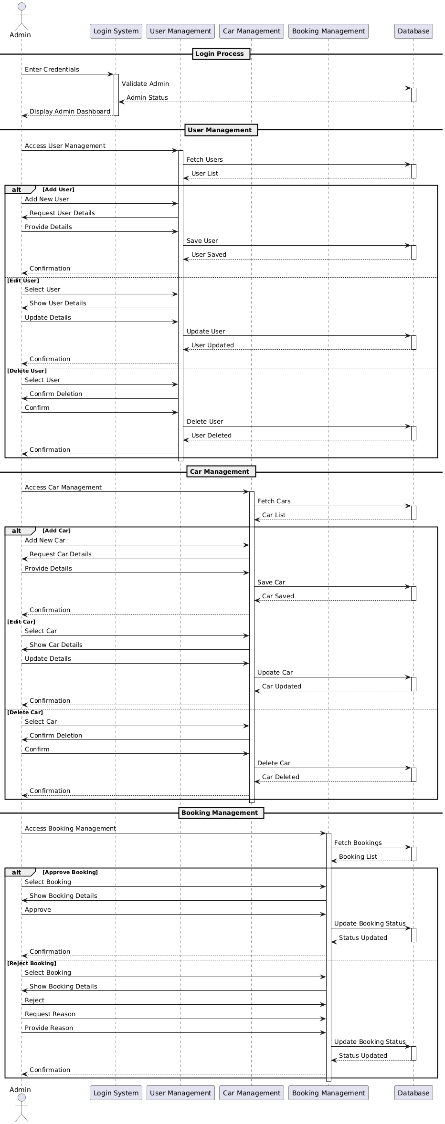
The **Booking** class handles booking records, identified by id, and includes license\_number, plate\_number, start\_date, end\_date, cost, and status. It supports methods for creating, approving, rejecting, and viewing bookings.

The **RentLog** class logs the rental history, tracking attributes such as id, license\_number, plate\_number, rent\_date, return\_date, total\_cost, and status, along with methods to log rentals, update statuses, and view history.

The diagram also shows **1-to-1 relationships** between the entities like Booking, Customer, Car, and RentLog, emphasizing how each booking and rental is associated with a specific customer and car. This diagram is essential for understanding the object-oriented design and data flow in the car rental application.

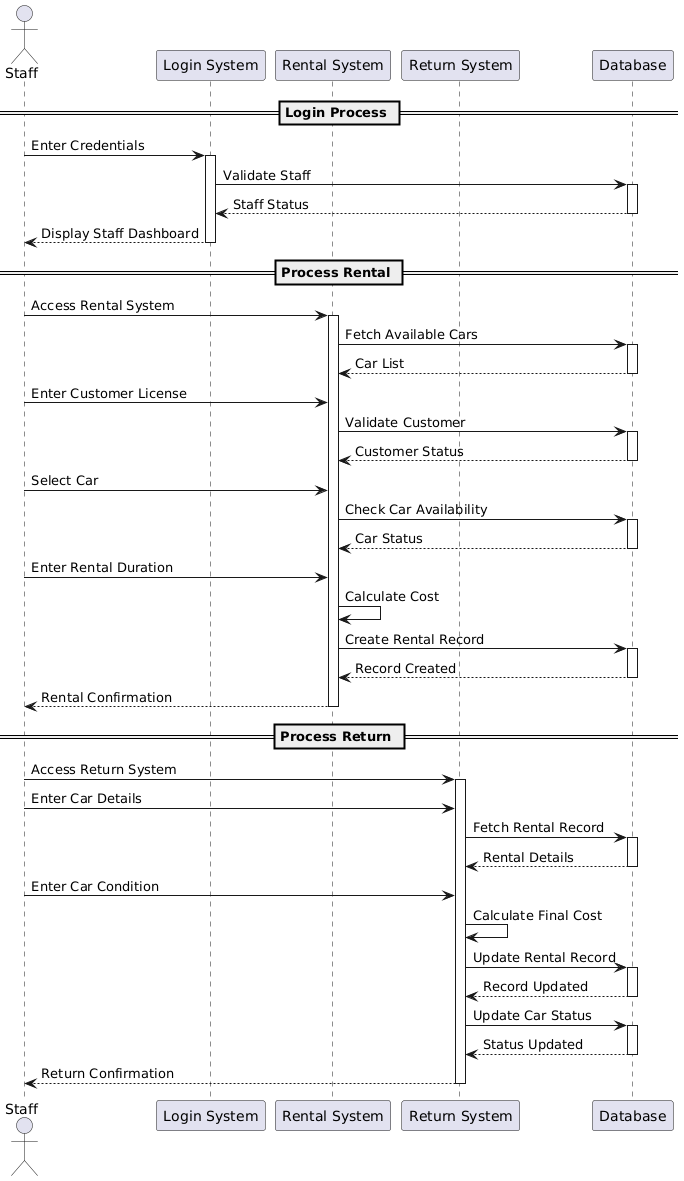
### Sequence Diagram

#### Admin Sequence Diagram



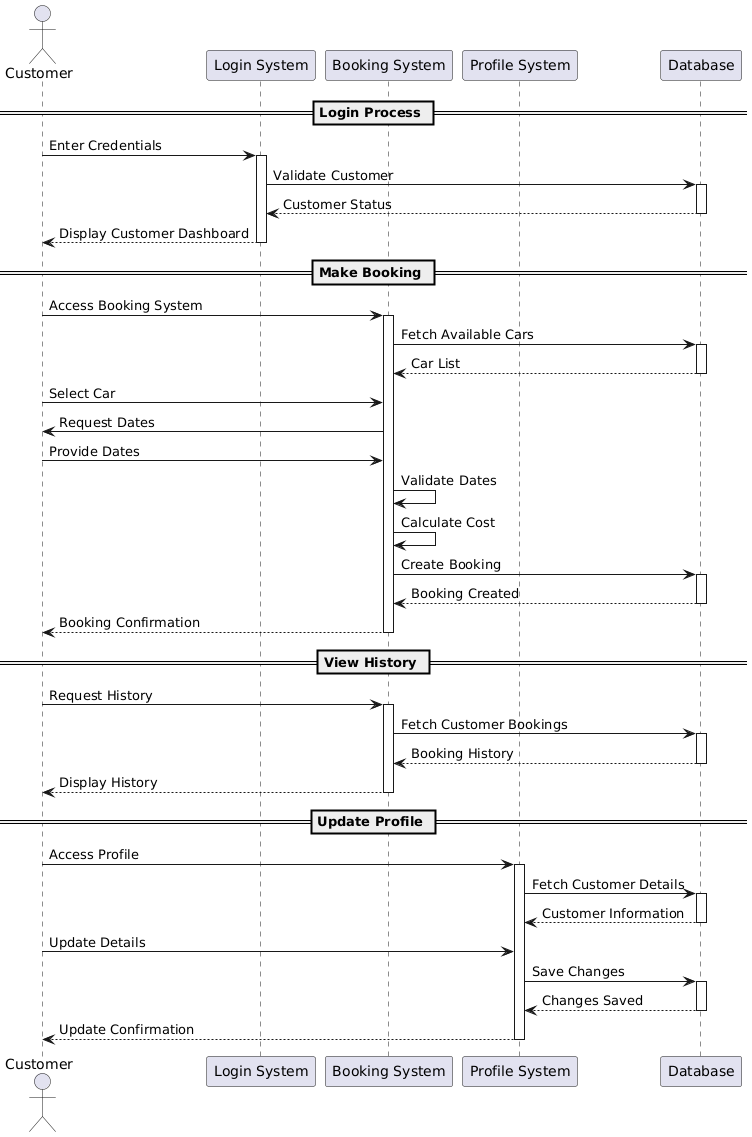
The sequence diagram for the Admin user captures the step-by-step interaction between the Admin and the system components during various administrative tasks. After the Admin logs in with valid credentials, the system authenticates the user and directs them to the Admin Dashboard. From there, the Admin can initiate user management, car management, or booking management operations. In user management, the Admin sends a request to add, edit, or delete a user. The system responds by validating the input, updating the database, and confirming the action. Similarly, in car management, the Admin requests to add or modify car details, and the system processes the input and updates the car records. For booking management, the Admin retrieves a list of pending bookings, selects a booking, and either approves or rejects it. Upon approval, the system updates the booking status and sends a confirmation to the customer; upon rejection, it prompts the Admin to enter a reason, updates the booking as rejected, and notifies the customer. The sequence ends when the Admin logs out, and the session is terminated.

#### Staff Sequence Diagram



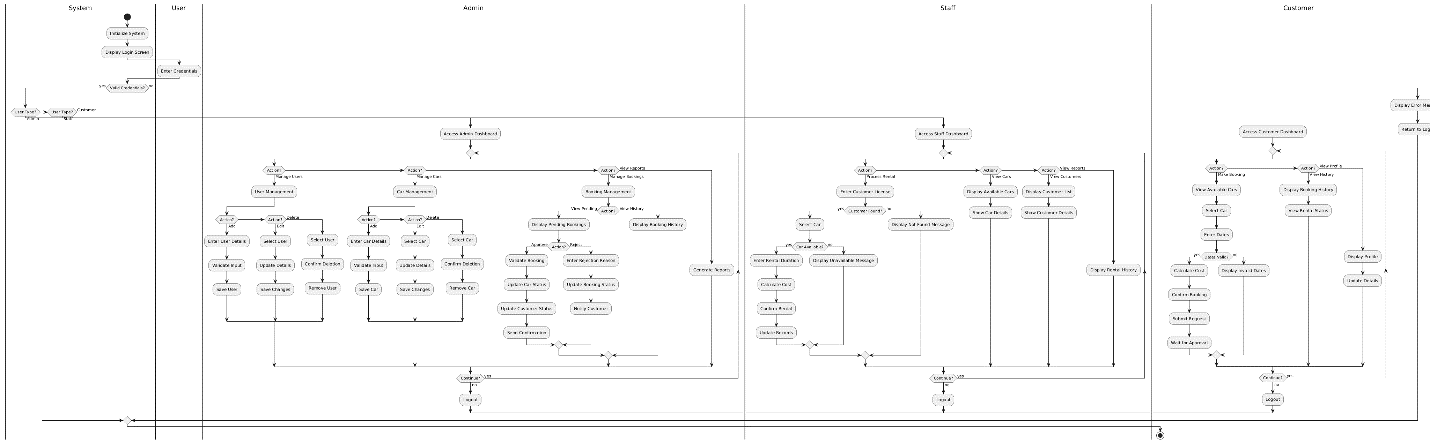
The sequence diagram for Staff shows how staff members interact with the system primarily to process rentals and manage basic car and customer data. After successful login and authentication, the staff member accesses the Staff Dashboard. To process a rental, the staff inputs the customer’s license number, prompting the system to search for the customer in the database. If the customer is found, the staff selects an available car and inputs the rental duration. The system checks car availability, calculates rental costs, and displays it. The staff then confirms the rental, and the system updates the rental records accordingly. The staff can also request to view car lists or customer profiles, to which the system responds with the relevant data. As with the other roles, the interaction ends when the staff member logs out.

#### Customer Sequence Diagram



The Customer sequence diagram illustrates the flow of interactions when a customer accesses and uses the car rental system. The sequence starts with the customer entering their login credentials. Once authenticated, the system grants access to the Customer Dashboard. The customer may initiate a booking by requesting to view available cars. The system responds with a list of cars, and the customer selects a car and inputs rental dates. The system then validates the dates, calculates the cost, and if all inputs are valid, it allows the customer to confirm the booking and submit the request. The system stores the booking as pending and waits for Admin approval. Additionally, the customer can view their booking history, rental status, or update their profile information. All actions involve a series of requests and responses between the customer and the system, ending with the customer choosing to log out.

#### Activity Diagram



This activity diagram illustrates the dynamic workflow of the Car Rental System, outlining the sequence of activities performed by different user roles—Admin, Staff, and Customer—starting from system initialization to the completion of various tasks. The system begins by initializing and displaying a login screen, where users enter their credentials. Once credentials are verified, the user is directed to the appropriate dashboard based on their role: Admin, Staff, or Customer.

For Admin users, the diagram shows access to an admin dashboard with several key functions. These include managing users (adding, editing, or deleting user records with input validation), managing cars (adding, editing, or removing cars), and handling bookings. In booking management, Admins can view pending requests, approve or reject bookings, update booking and customer statuses, and notify customers accordingly. Additionally, Admins have access to reports and booking histories to monitor system activity.

Staff users access their own dashboard, where they can process car rentals by entering a customer's license. If the customer is found in the system, the staff member selects an available car, enters rental duration, calculates the cost, confirms the rental, and updates the records. If the customer is not found or the car is unavailable, appropriate messages are displayed. Staff can also view cars, display their details, and access customer information and rental reports.

Customers interact with the system through a customer dashboard. They can make bookings by browsing available cars, selecting one, entering rental dates, and submitting a booking request. If the dates are valid, the system calculates the cost and awaits admin approval. Customers can also view their rental history and booking status, as well as access and update their profile information.

Each user role concludes their session by choosing to log out, which returns them to the login screen. In cases where invalid login credentials are entered, the system displays an error message. Overall, the activity diagram provides a comprehensive and structured overview of the system’s behavior and the flow of actions for each user role.