DBMSL (N2)- Mini-Project Report on

**CAR RENTAL MANAGEMENT SYSTEM**

Prepared by:

Soham Sathe Roll No. 31269

Amaan Shaikh Roll No. 31270

Sahil Shrotri Roll No. 31274

11th October, 2024.

**ABSTRACT**

The Car Rental Management System aims to streamline the process of renting cars by providing a user-friendly interface for both customers and administrators. The system allows customers to book and return cars, while administrators can manage the car inventory and oversee transactions. The project is implemented using Python and MySQL, ensuring a robust and scalable solution. This report details the system's design, functionalities, database structure, and testing processes.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Title | | Page No. |
| 1 | Introduction | | 1 |
| 2 | Scope | | 1 |
| 3 | System Requirements | | 1-2 |
|  | 3.1 | Hardware | 1 |
| 3.2 | Software | 1 |
| 3.3 | Requirement analysis of project (user-role wise functionalities) | 1-2 |
| 4 | System functionalities and features | | 2 |
| 5 | Database design - ER diagram | | 3 |
| 6 | Relational Database Design (Schema) | | 4 |
| 7 | Database Normalization | | 4 |
| 8 | Graphical User Interface | | 5 |
| 9 | Source code | | 6-9 |
| 10 | Test cases | | 10 |
| 11 | Conclusion | | 10 |

**INTRODUCTION**

The Car Rental Management System is developed to facilitate the operations of car rental services. The primary objectives are to provide a seamless booking experience for customers, ensure efficient management of car inventory by administrators, and maintain an accurate record of transactions. This system replaces manual processes, reduces errors, and enhances customer satisfaction through automation and real-time data processing.

**SCOPE**

The scope of this project includes:

* Developing an interface for customers and administrators.
* Implementing secure user authentication and role-based access control.
* Allowing customers to view available cars, make bookings, and return cars and see their past booking history.
* Enabling administrators to add, update, and delete car records.
* Managing rental transactions and generating receipts.
* Ensuring data integrity and security through database constraints and validation mechanisms.

**SYSTEM REQUIREMENTS**

**3.1 Hardware requirements**

* A computer with a minimum of 4GB RAM and 500GB hard disk.
* A reliable internet connection for accessing the web interface.

**3.2 Software requirements**

* Operating System: Windows 10 or later, macOS, or Linux.
* Python 3.7 or later.
* MySQL Server 8.0 or later.
* MySQL Connector for Python.
* Tkinter library for GUI development.
* Integrated Development Environment (IDE): Visual Studio Code.

**3.3 Requirement analysis of project (user-role wise functionalities)**

**Customers**:

* Register and log in to the system.
* View available cars.

1

* Book a car.
* Return a car.
* View their booking history and transactions.

**Administrators**:

* Log in to the admin panel.
* Add, update, and delete car records.
* View all bookings and transactions.
* Manage customer accounts and resolve issues.

**SYSTEM FUNCTIONALITIES AND FEATURES**

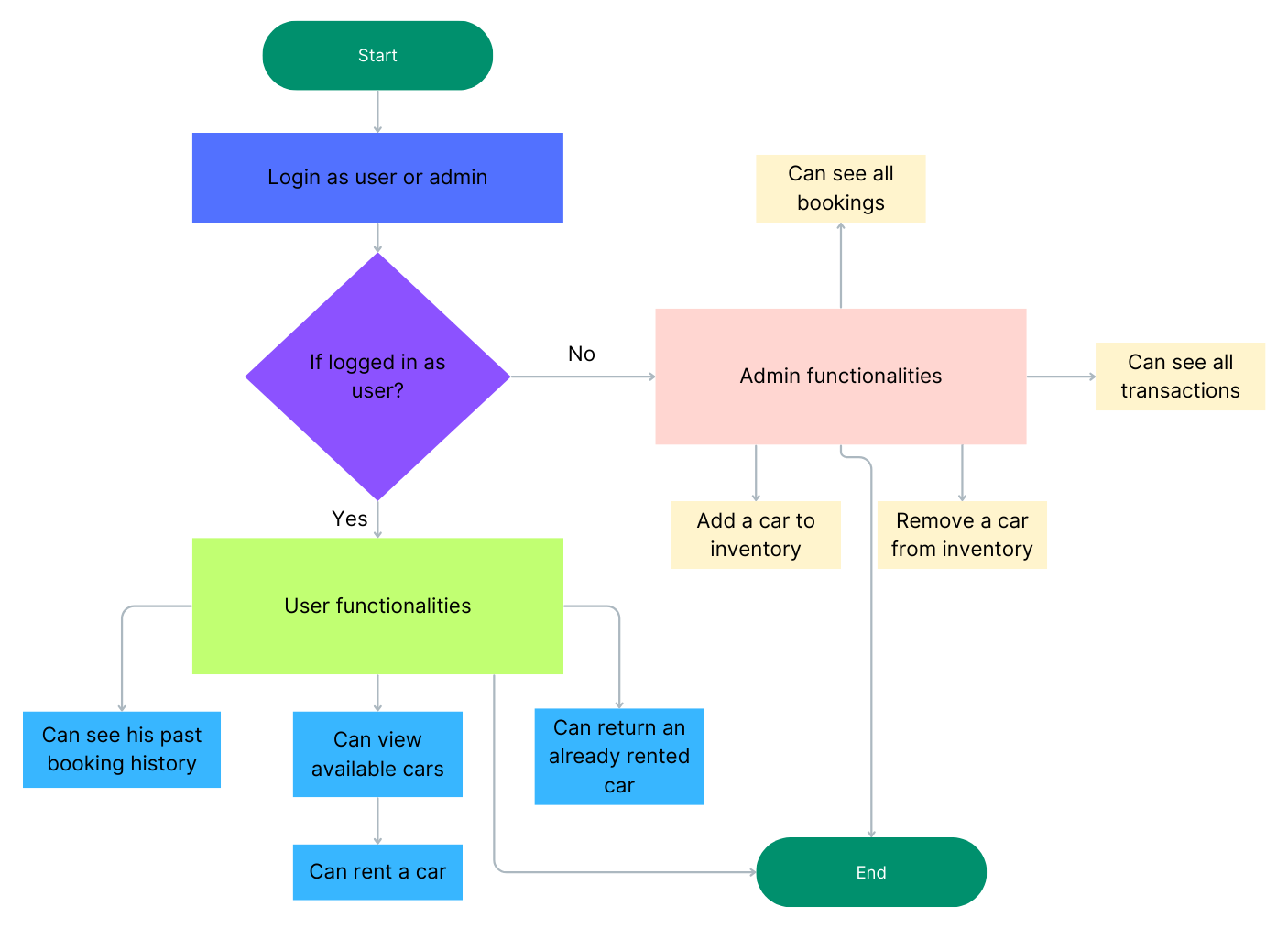


Fig 1. System functionalities

2

**DATABASE DESIGN (ER/ EER DIAGRAM)**

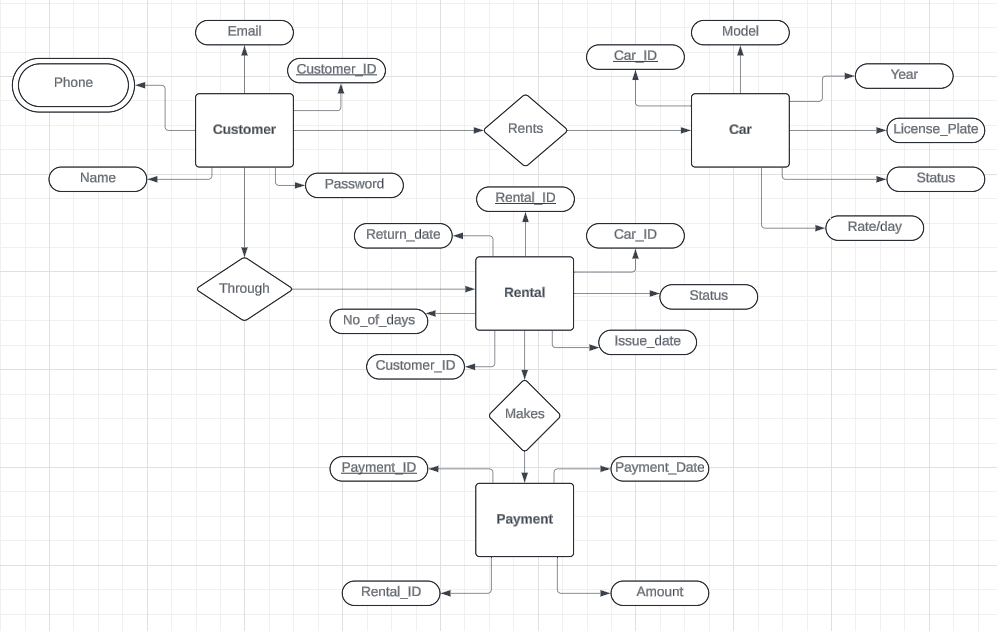


Fig 2. ER Diagram

3

**RELATIONAL DATABASE DESIGN (SCHEMA)**

The relational database schema outlines the structure of the database in terms of tables and their relationships. The primary tables in the system are:

* **Customer**: Customer\_ID (Primary Key), Name, Phone, Email, Password.
* **Car**: Car\_ID (Primary Key), Model, Year, License\_Plate, Status, Rate\_per\_day.
* **Rental**: Rental\_ID (Primary Key), Customer\_ID (Foreign Key), Car\_ID (Foreign Key), Issue\_date, Return\_date, No\_of\_days, Status.
* **Payment**: Payment\_ID (Primary Key), Rental\_ID (Foreign Key), Payment\_Date, Amount.

**DATABASE NORMALISATION**

The database is normalized to eliminate redundancy and ensure data integrity. The normalization process involves:

* **First Normal Form (1NF)**: Ensuring all tables have atomic columns and no repeating groups.
* **Second Normal Form (2NF)**: Removing partial dependencies; every non-key attribute is fully functionally dependent on the primary key.
* **Third Normal Form (3NF)**: Eliminating transitive dependencies; non-key attributes are not dependent on other non-key attributes.

4

**GRAPHICAL USER INTERFACE**

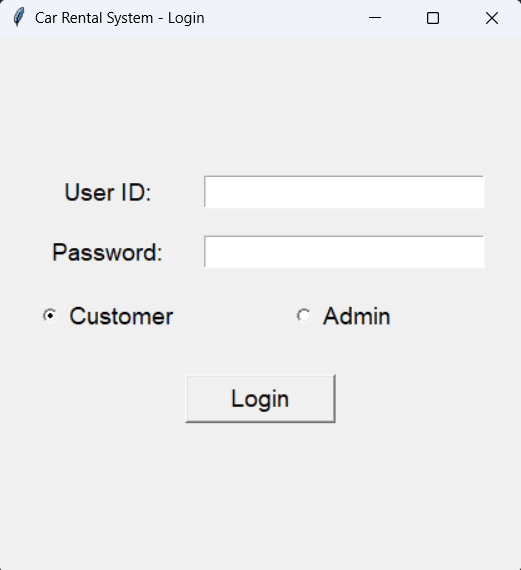
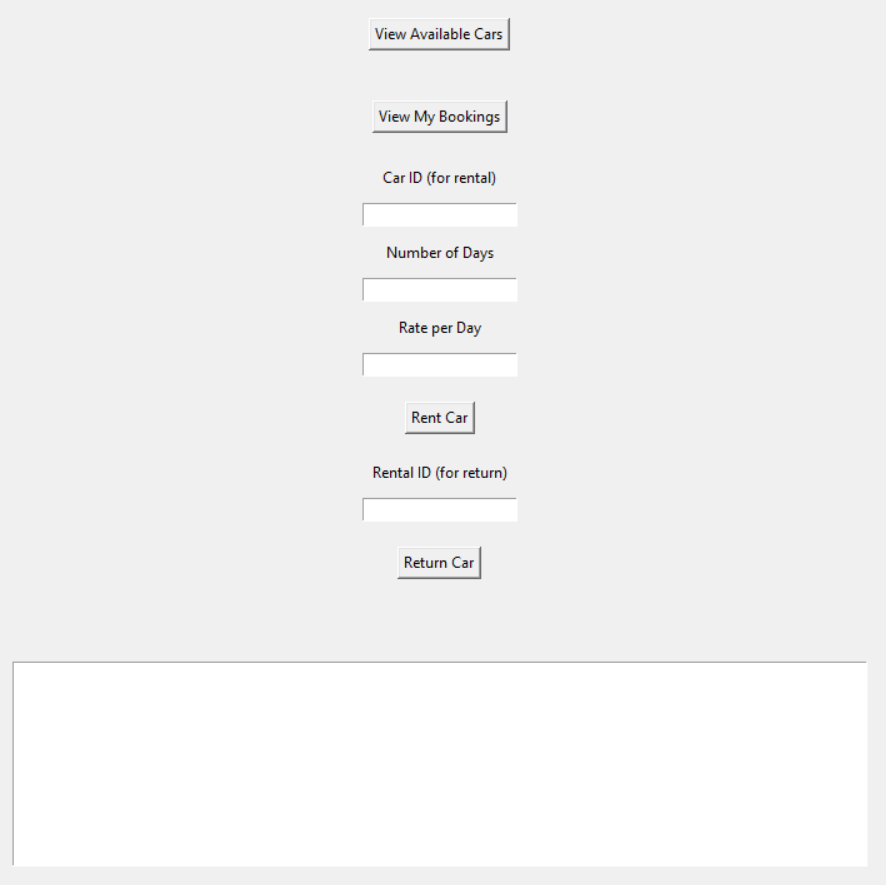


Fig 3. Login Page

Fig 4. Customer dashboard

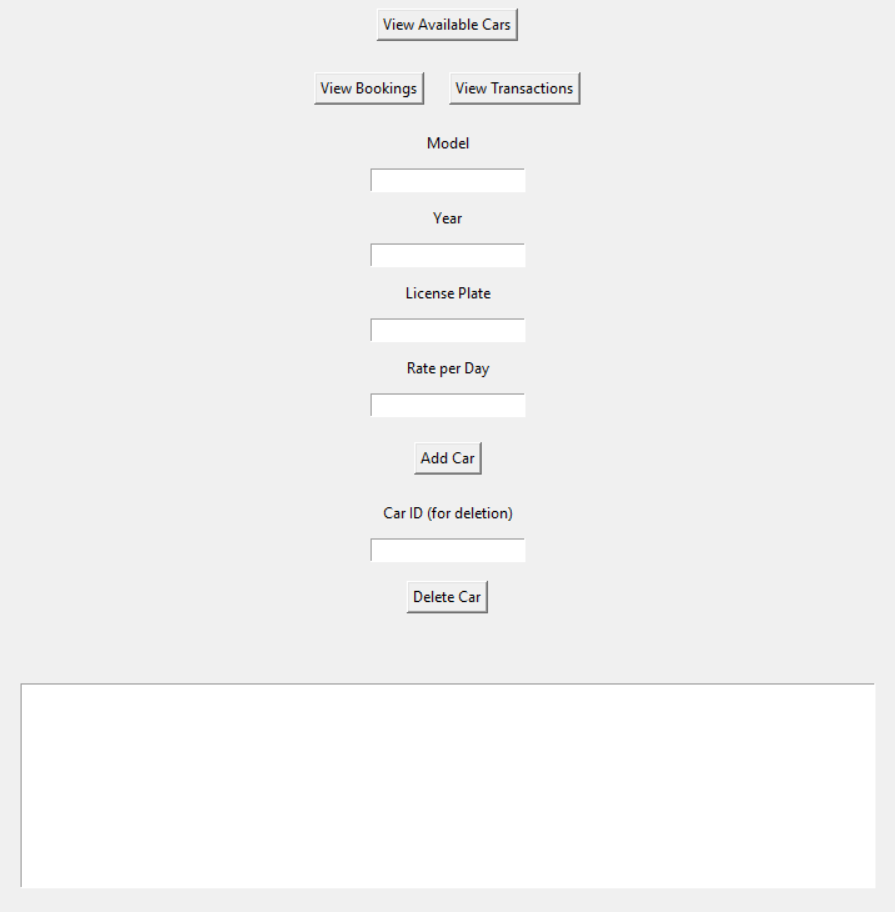
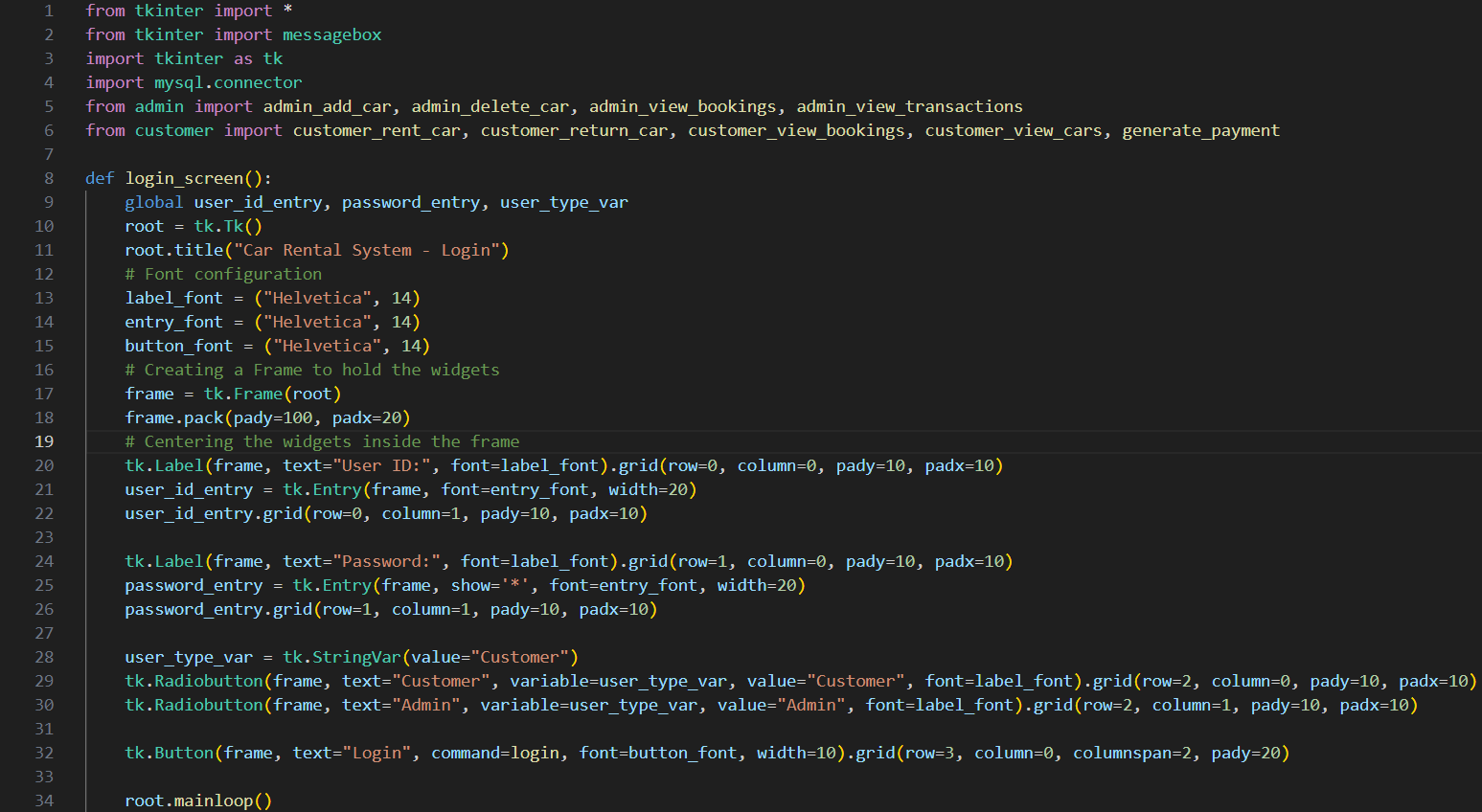
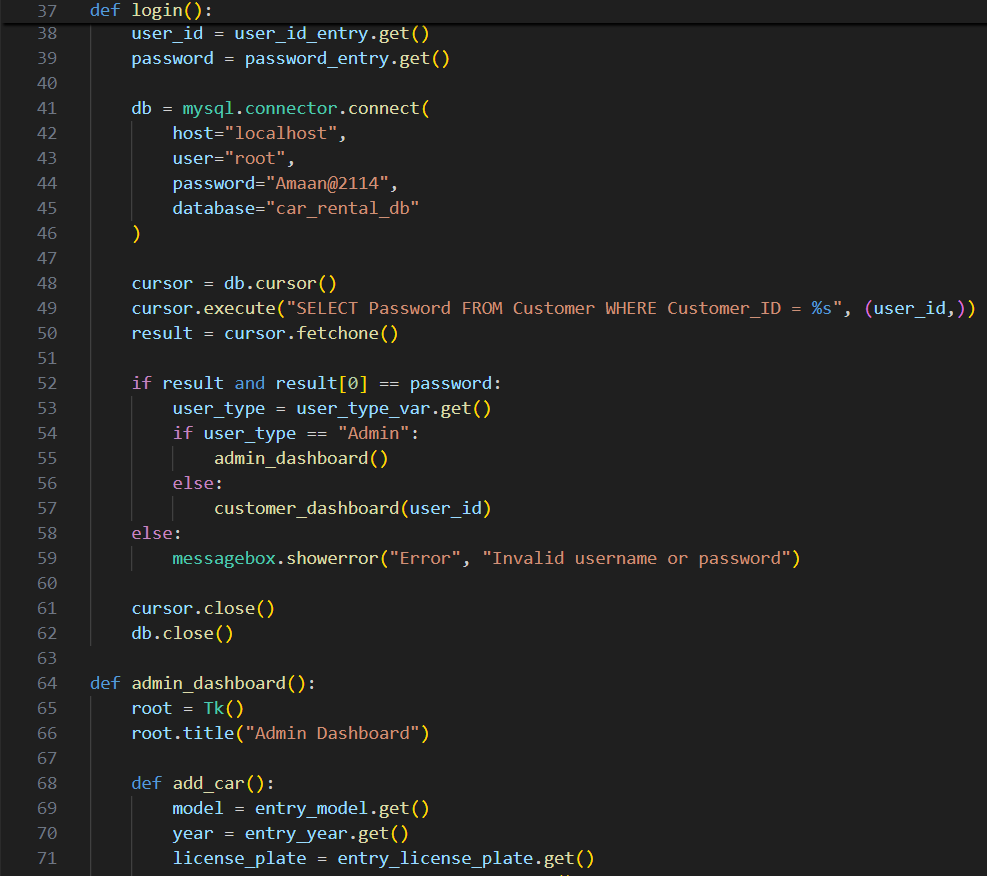


Fig 5. Admin dashboard

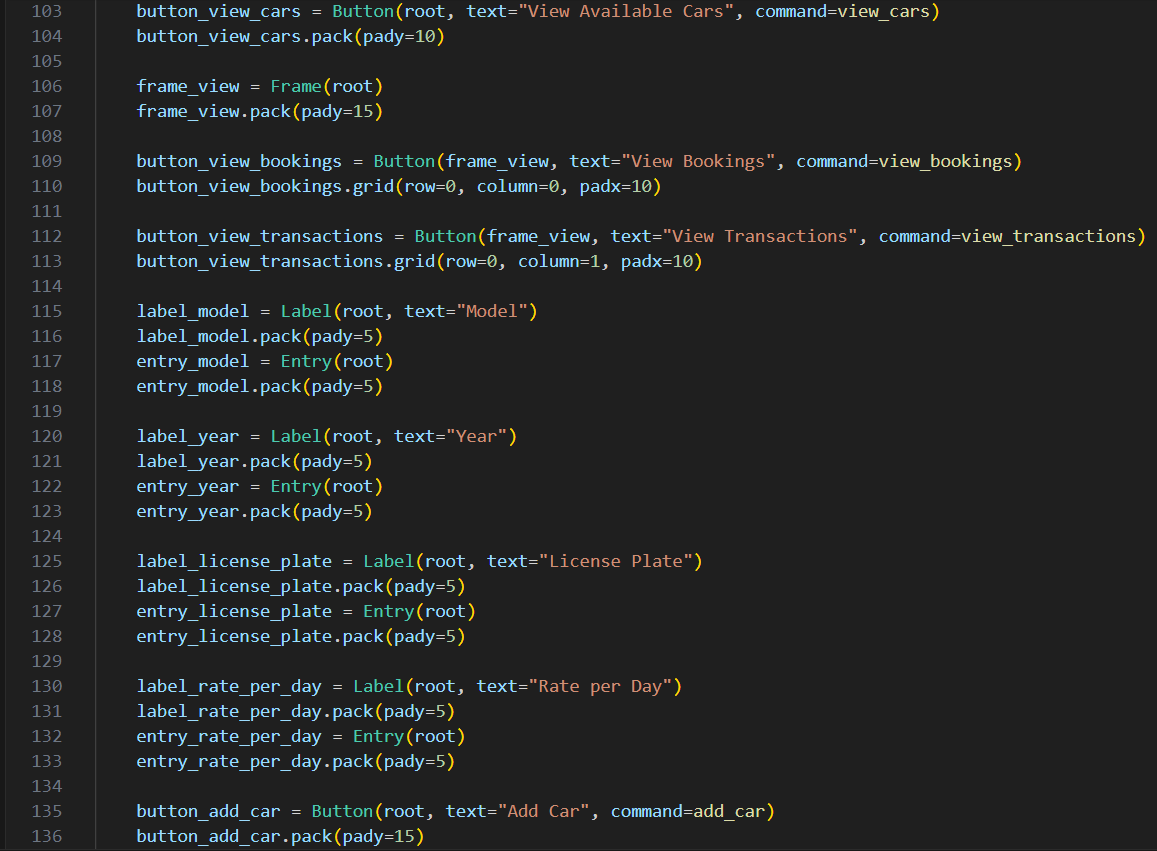
5

**SOURCE CODE**

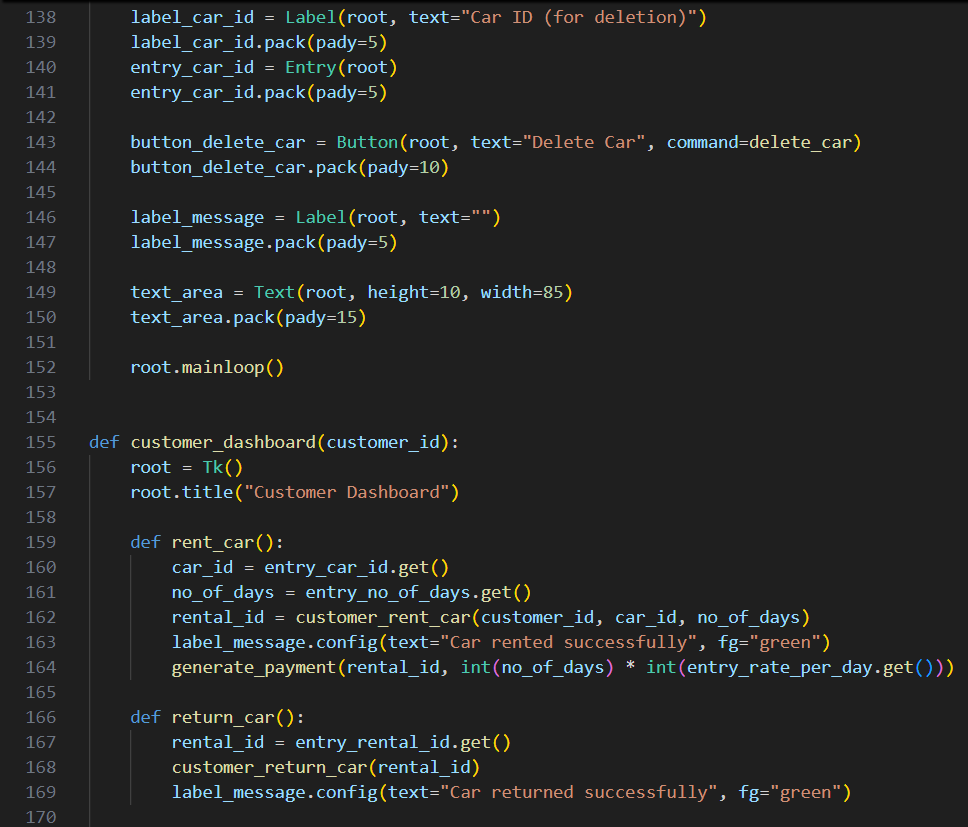


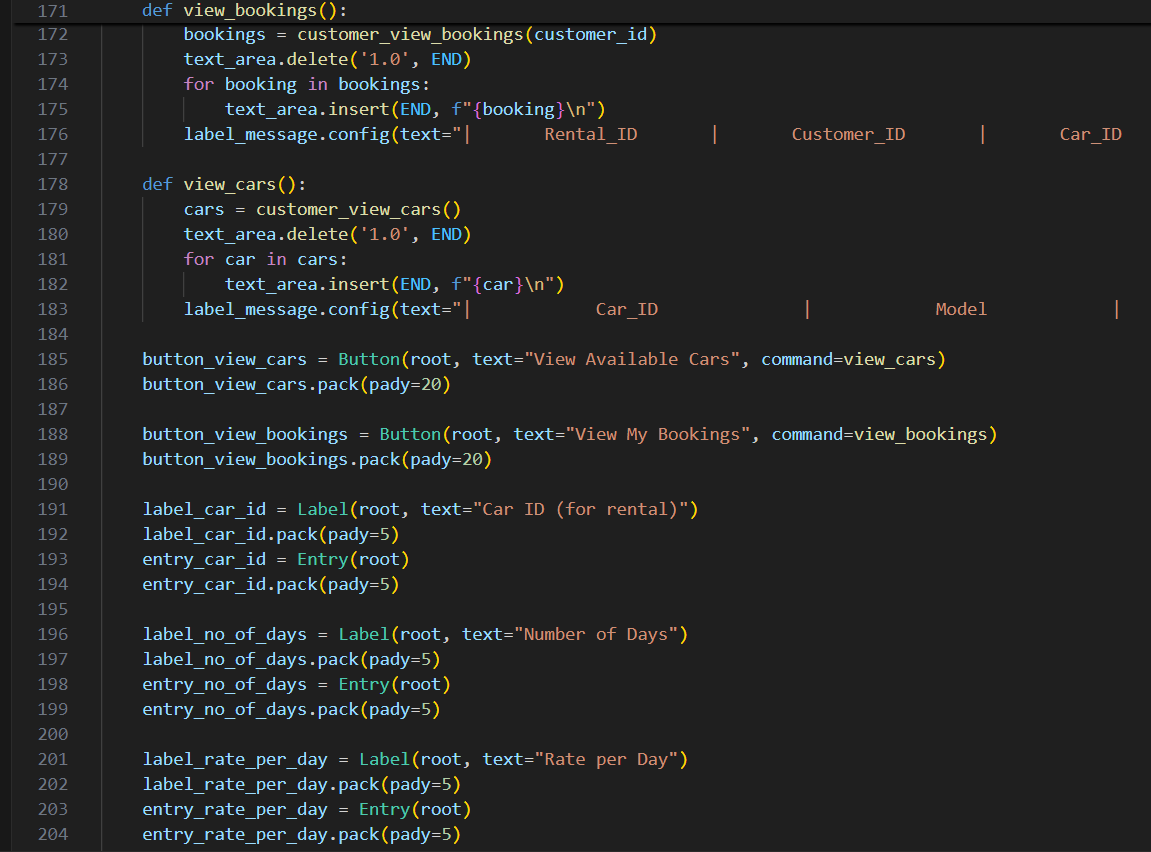


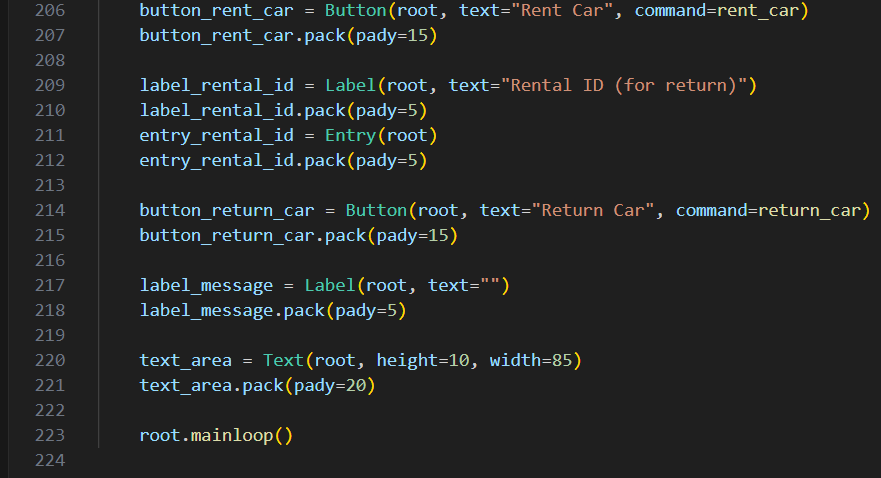
6



7







9

**TEST CASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **TEST CASE** | **TEST STEPS** | **EXPECTED RESULT** | **ACTUAL RESULT** | **STATUS** |
| T01 | Customer with customer Id 1 rents car with car Id 4 | 1) Logs in from 1  2) Rents car 4 by entering the car id, no. of days and rate/day | 1) Car 4 booked for customer 1 and is visible in current bookings  2) Car 4 not available for other customers  3) Rental for customer 1 with car 4 generated with the appropriate transactions being saved. | Same as expected result | Pass |
| T02 | Customer with customer Id 3 rents car with car Id 7 | 1) Logs in from 3  2) Rents car 7 by entering the car id, no. of days and rate/day | 1) Car 7 booked for customer 3  and is visible in current bookings  2) Car 7 also not available for other customers.  3) Rental for customer 3 with car 7 generated with the appropriate transactions being saved. | Same as expected result | Pass |
| T03 | Customer with customer Id 3 returns car with car Id 7 | 1) Logs in from 1  2) Returns car 4 by entering the rental id of that booking. | 1) Car 4 status ‘returned’ is shown in bookings of customer 1  2) Car 4 is now available for rental to other customers | Same as expected result | Pass |
| T04 | Admin adds car 9 | 1) Logs in from an admin account  2) Enters model, year, license plate, rate/day | 1) The car is added to the inventory and available to all customers | Same as expected result | Pass |
| T05 | Admin deletes car 6 | 1) Logs in from an admin account  2) Enters the car id of the car 6 to be deleted | 1) Car with car Id 6 is deleted from the inventory.  2) No customer views car with car Id 6 as available car for rental | Same as expected result | Pass |

**CONCLUSION**

The Car Rental Management System successfully automates the process of renting cars, enhancing operational efficiency and customer satisfaction. The system's design ensures scalability, security, and ease of use. Through rigorous testing and adherence to best practices in database design, the system provides a reliable and user-friendly solution for car rental services. Future enhancements could include adding features like online payment integration, mobile app support, and advanced reporting capabilities.

10