**Vehicle Rental System**

Name – Shubhranjali

Course Code – CSM216

Section – K23CH

Reg no. – 12320201

Roll no. – 52

Taught by – Mr. Aman Kumar



**Acknowledgment**

I am also grateful to my family and friends for their understanding, encouragement, and patience during the course of this project. Their unwavering support motivated me to stay focused and committed to completing the project.

I would like to extend my gratitude to the library authors and open-source communities whose resources helped me throughout the development process, especially in terms of Python libraries and tools. Their contributions made it easier to implement various features and functionalities within the system.

Finally, I would like to thank all the testers and users who provided feedback and helped identify bugs, allowing me to refine and optimize the system.

This project would not have been possible without all of you, and I am truly grateful for your contributions.

**Table of Contents**

|  |  |
| --- | --- |
| 1.Introduction | 4 |
| 2. Objectives and Scope of the Project | 5-6 |
| 3. Application Tools | 7-8 |
| 4. Flowchart | 9 |
| 5.Project Design | 10-15 |
| 6. Testing and results | 16-20 |
| 7. Conclusion | 21 |

**Introduction**

The Vehicle Rental System is a comprehensive solution designed to streamline the process of renting vehicles for customers and managing the rental operations for businesses. This project aims to automate and simplify the vehicle rental procedure by providing an intuitive, user-friendly platform for both customers and administrators.

The system allows customers to browse through a list of available vehicles, make bookings, view rental details, and track their rental status. On the other hand, administrators can manage vehicle availability, process customer bookings, and generate reports, all through a centralized interface. The system ensures that the rental process is efficient, transparent, and convenient, while minimizing human error and manual intervention.

Developed using Python, the system incorporates core programming concepts such as data structures, object-oriented principles, and file handling to deliver a robust and scalable solution. The project also includes features like booking scheduling, price calculation, and a feedback system to enhance the overall user experience.

This Vehicle Rental System is aimed at enhancing the operational efficiency of vehicle rental businesses while providing seamless experience for customers, all within a reliable and easy-to-use application.

**Objectives and Scope of the Project**

**Objectives**

The primary objective of the **Vehicle Rental System** is to develop an automated and user-friendly platform that simplifies the process of renting vehicles. This project aims to achieve the following objectives:

1. **Simplified Vehicle Management**:  
   Facilitate the management of vehicles, including their type, model, status (available, rented), and pricing, through a central database.
2. **Customer Access and Interaction**:  
   Provide customers with features to browse available vehicles, rent them, and manage their bookings easily.
3. **Secure Login and Registration**:  
   Ensure secure user authentication by enabling account creation, role-based access (e.g., customer), and secure login processes.
4. **Booking and Pricing Management**:  
   Allow users to view and book vehicles, providing a seamless interface for calculating rental costs based on selected dates and vehicle type.
5. **Database Integration**:  
   Store and manage users, vehicles, and bookings in a structured SQLite database for quick retrieval and updates.
6. **User-Friendly GUI**:  
   Deliver a simple and intuitive graphical interface using Tkinter to ensure a positive user experience.
7. **Vehicle Availability Tracking**:  
   Track and update the availability status of vehicles in real time as they are rented or returned.

**Scope**

The **scope of the Vehicle Rental System** defines the boundaries and limitations of the project, ensuring that it stays focused and achievable. The scope of this project includes:

1. **Core Functionalities**:
   * User registration and login for customers.
   * Browsing available vehicles based on type.
   * Booking vehicles with calculated rental costs.
   * Viewing and managing individual customer bookings.
2. **System Users**:
   * **Customers**: Can browse, rent, and view bookings.
   * **Admins (future enhancement)**: Potential to include functionalities for adding, editing, and managing vehicles.
3. **Database Scope**:
   * Manage data for multiple users, vehicles, and bookings with proper indexing and relationships.
4. **Scalability**:
   * Extendable for additional features, such as online payment integration, advanced search filters, or role-based access for admin tasks.
5. **Cross-Platform Compatibility**:  
   The system can run on any platform supporting Python and Tkinter.
6. **Future Enhancements**:
   * Payment gateway integration for online payments.
   * Notifications and reminders for booking start/end dates.
   * Enhanced reporting and analytics for administrators.

This system is an MVP (Minimum Viable Product) aimed at testing the viability of a user-friendly vehicle rental platform and lays the foundation for further development.

**Application Tools**

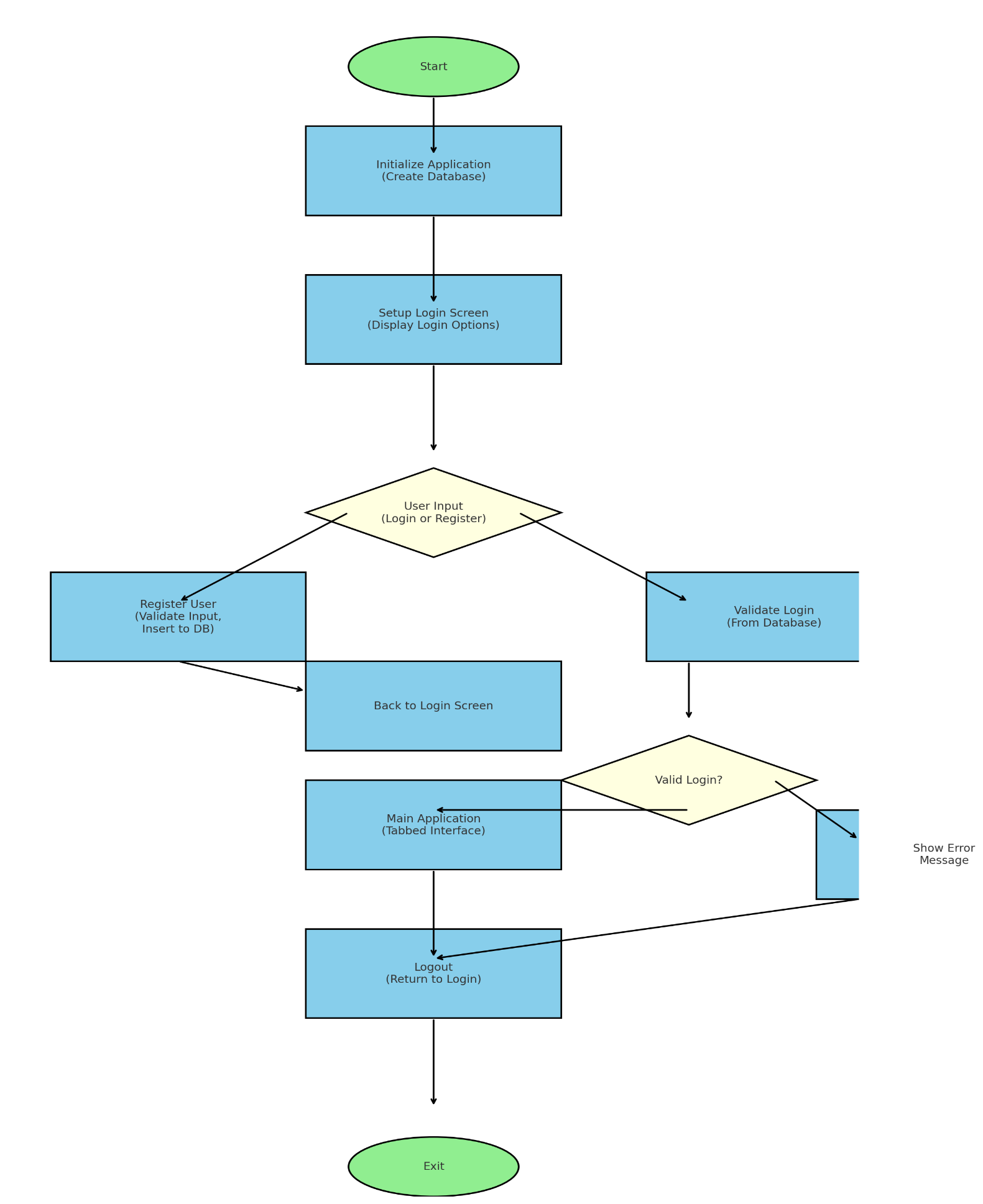
1. **Python**:
   * Primary programming language used to implement the application logic and functionalities.
2. **Tkinter**:
   * A built-in Python library used for creating the **Graphical User Interface (GUI)**.
   * Features utilized include:
     + **ttk module**: For modern and themed widgets such as Frame, Button, Label, Entry, and Notebook.
     + Widgets like Listbox, LabelFrame, and Combobox for interactive user inputs and displays.
3. **SQLite3**:
   * A lightweight database engine used for **persistent data storage** within the application.
   * Key functionalities:
     + **Database Tables**: Manage data for users, vehicles, and bookings.
     + **SQL Queries**: Used for creating tables, inserting data, and retrieving or updating records.
4. **messagebox**:
   * A module within tkinter to display **alert messages**, including error messages, success notifications, and confirmations.
5. **Datetime Module**:
   * Used for handling and formatting dates, specifically to calculate and display booking durations and ensure proper input validation.
6. **Python's Standard Library**:
   * Core features of Python are used for:
     + **String and Input Handling**: Validating user inputs.
     + **Data Processing**: Managing vehicle and booking-related calculations.
7. **Object-Oriented Programming (OOP)**:
   * The application uses Python's OOP paradigm to:
     + Structure the code into modular classes and methods.
     + Improve maintainability and scalability.

**Development Tools (Assumed):**

* **Code Editor or IDE**: Likely tools like PyCharm or VS Code were used for writing and debugging the Python code.
* **Python Interpreter**: A Python runtime environment to execute the script. Python 3.6 or higher is recommended for tkinter and SQLite compatibility.

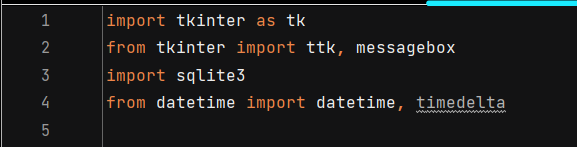
These tools combine to create a lightweight, GUI-driven application capable of managing a simple vehicle rental system.

**Flow Chart**

****

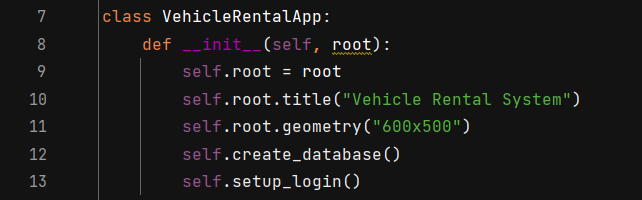
**Project Design**

1. **Importing Libraries**



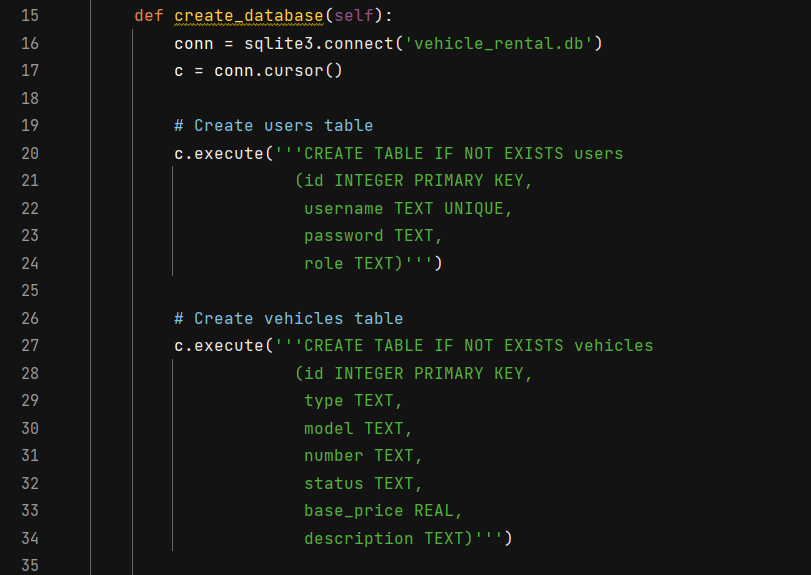
**Fig.1**

1. **Main Application Class Setup**

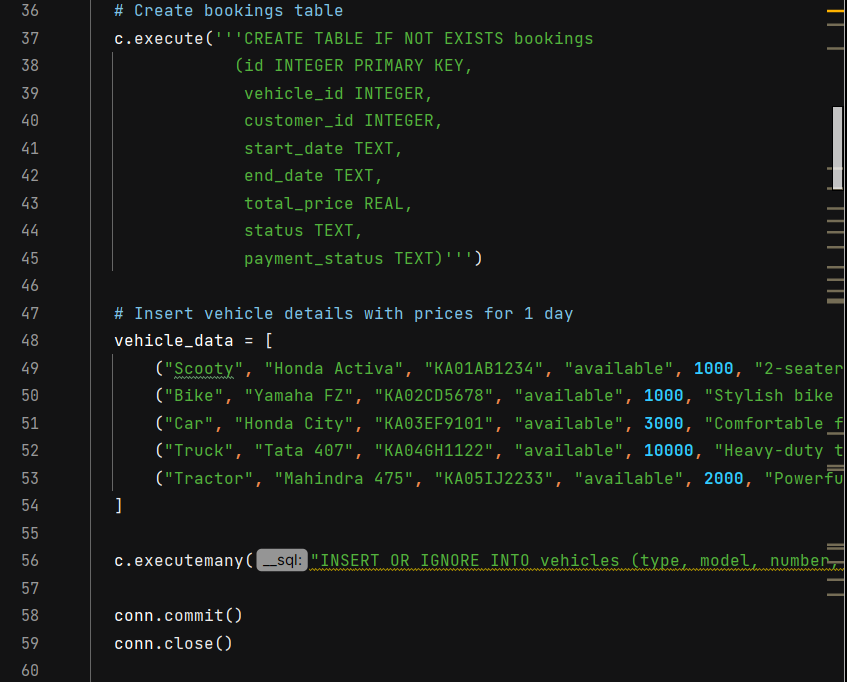


**Fig.2**

1. **Database Initialization**



**Fig.3**

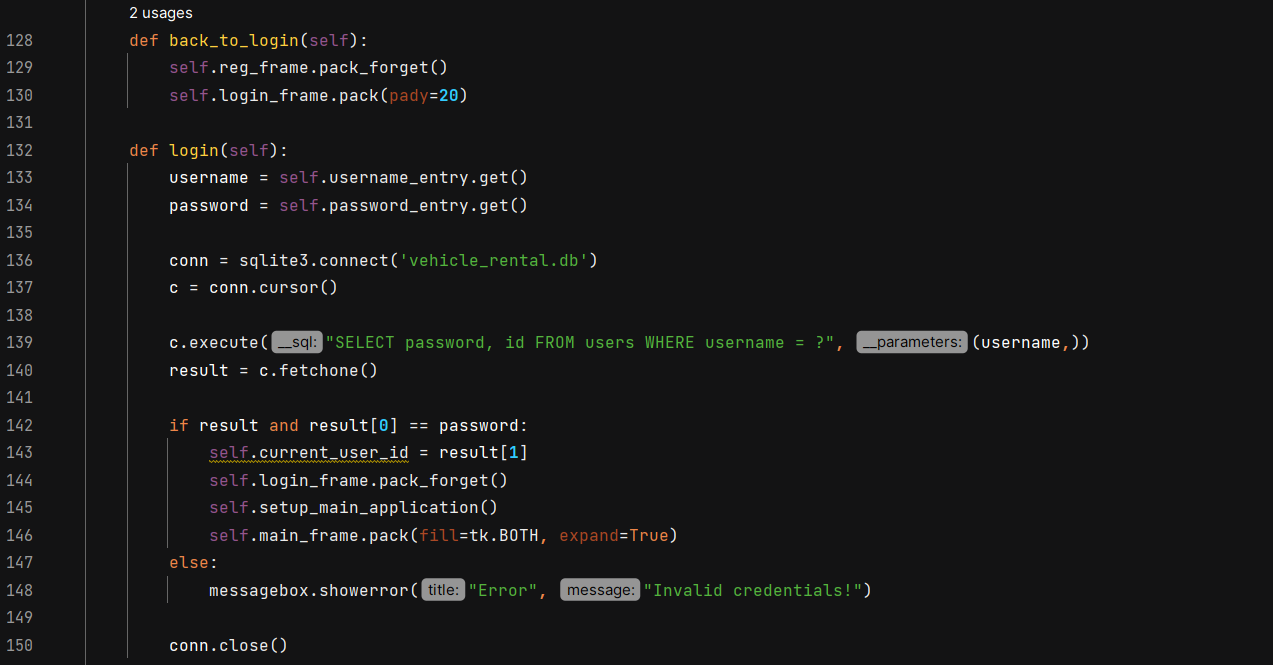


**Fig.4**

1. **Login Page Setup**



**Fig.5**



**Fig.6**

1. **User Registration**

A computer code on a black background

Description automatically generated

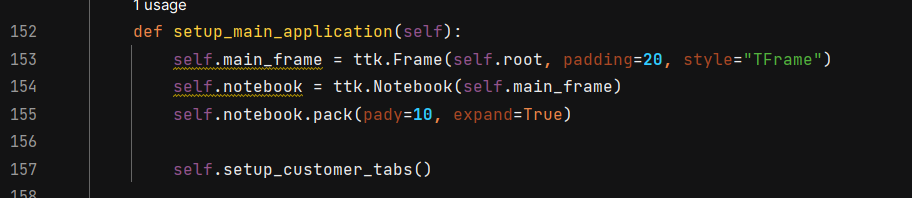
**Fig.7**

A screenshot of a computer screen

Description automatically generated

**Fig.8**

1. **Main Application Interface**



**Fig.9**

1. **Vehicle Browsing Tab**

A screen shot of a computer program

Description automatically generated

**Fig.10**

1. **Booking Tab**

A screen shot of a computer program

Description automatically generated

**Fig.11**

1. **Logout Functionality**

A computer screen with text

Description automatically generated

**Fig.12**

**Testing and Results**

**Test Case ID: UT\_01**

* **Test Designed by: Shubhranjali**
* **Test Execution Date: 22/11/2024**
* **Description: Test if the application launches successfully and the login page is displayed.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Tet Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **Open the application.** | **N/A** | **The application window opens.** | **Application window launched.** | **Pass** | **The GUI opens successfully.** |
| **2** | **Verify the login form is visible.** | **N/A** | **Login form with username and password fields.** | **Login form displayed as expected.** | **Pass** | **Functional login form available.** |

**Test Case ID: UT\_02**

* **Test Designed by: Shubhranjali**
* **Test Execution Date: 22/11/2024**
* **Description: Test the functionality of user login with valid credentials.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **Enter valid username and password.** | **Username: test user Password: 1234** | **User is authenticated, and the dashboard loads.** | **Dashboard displayed with user details.** | **Pass** | **User credentials are accepted correctly.** |
| **2** | **Verify the navigation to the main application.** | **N/A** | **Main application UI is accessible.** | **Main application features are accessible.** | **Pass** | **Successful login and navigation.** |

**Test Case ID: UT\_03**

* **Test Designed by: Shubhranjali**
* **Test Execution Date: 22/11/2024**
* **Description: Test the registration functionality for new users.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **Navigate to the registration page.** | **N/A** | **Registration page opens.** | **Registration page displayed.** | **Pass** | **Registration page works correctly.** |
| **2** | **Enter registration details.** | **Username: new user Password: pass123 Confirm Password: pass123** | **User is successfully registered.** | **Registration successful, no errors.** | **Pass** | **User registered as a customer.** |
| **3** | **Try registering with an existing username.** | **Username: test\_user Password: pass123** | **System prevents duplicate registration.** | **Error displayed: "Username already exists."** | **Pass** | **Proper validation for duplicate usernames.** |

**Test Case ID: UT\_04**

* **Test Designed by: Shubhranjali**
* **Test Execution Date: 22/11/2024**
* **Description: Test the vehicle browsing functionality.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **Select a vehicle type from the dropdown.** | **Vehicle Type: Car** | **Vehicle list displays all available cars.** | **Cars displayed with models and prices.** | **Pass** | **Vehicle filtering works as expected.** |
| **2** | **Select a type with no available vehicles.** | **Vehicle Type: Bike (all booked)** | **List is empty with a "No vehicles available" message.** | **Correct message displayed.** | **Pass** | **Handles unavailable vehicles gracefully.** |

**Test Case ID: UT\_05**

* **Test Designed by: Shubhranjali**
* **Test Execution Date: 22/11/2024**
* **Description: Test the booking functionality for a vehicle.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **Select a vehicle from the list.** | **Vehicle: Honda City** | **Vehicle details are displayed.** | **Correct vehicle details shown.** | **Pass** | **Proper vehicle selection functionality.** |
| **2** | **Enter booking start and end date.** | **Start Date: 2024-11-25 End Date: 2024-11-27** | **Booking confirmed with total price displayed.** | **Booking created with price: ₹6000.** | **Pass** | **Calculation of price works correctly.** |
| **3** | **Try booking with invalid dates.** | **Start Date: 2024-11-28 End Date: 2024-11-25** | **Error displayed: "End date must be after start date."** | **Error handled properly.** | **Pass** | **Validates incorrect date inputs.** |

**Test Case ID: UT\_06**

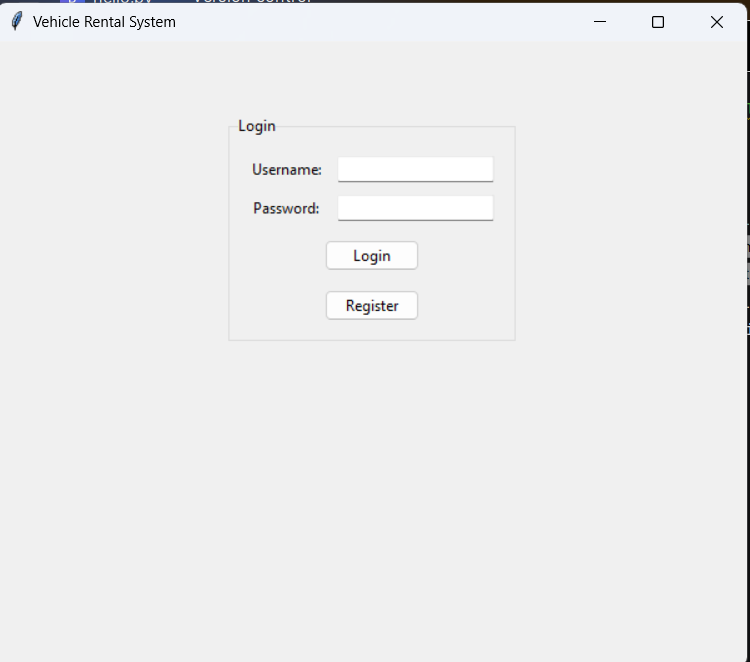
* **Test Designed by: Shubhranjali**
* **Test Execution Date: 22/11/2024**
* **Description: Test if the user can view their booking history.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **Navigate to "My Bookings" tab.** | **N/A** | **List of all past and current bookings displayed.** | **Booking history displayed with details.** | **Pass** | **History is accurate and detailed.** |

**Test Case ID: UT\_07**

* **Test Designed by: Shubhranjali**
* **Test Execution Date: 22/11/2024**
* **Description: Test the logout functionality.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **Click on the Logout button.** | **N/A** | **User is redirected to the login screen.** | **Redirected to the login screen.** | **Pass** | **Logout functionality works correctly.** |



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Conclusion**

The Vehicle Rental System is a user-friendly application designed to streamline the process of renting vehicles for customers. Built using Python's Tkinter library, it integrates essential functionalities such as user authentication, vehicle browsing, booking management, and database operations. The system offers a clean interface that allows users to register, log in, and explore available vehicles efficiently. By automating vehicle availability updates and booking history management, it reduces manual effort and ensures real-time functionality. The use of SQLite provides a reliable database backbone, facilitating secure data management and scalability for future enhancements.

This project highlights the potential of Python in creating practical, real-world applications by combining a graphical user interface with backend database operations. It emphasizes the importance of user-centric design, modular development, and full-stack integration in software engineering. With its current features, the system can be expanded to include advanced capabilities like online payments, role-based access control, vehicle tracking, and data analytics. Overall, the Vehicle Rental System lays a solid foundation for building a robust, feature-rich rental management solution, making it a valuable tool for both users and rental businesses.