

SHOWROOM MANAGEMENT SYSTEM

Documentation

Author: Sayab Gulfaraz Abbasi

Date: 20 February 2025

Version: 1.0

Table of Contents

1. Introduction
 2. Project Overview
 3. System Features
 4. Software Requirements
 5. System Workflow
 6. Implementation Details
 7. Database Structure
 8. User Guide
 9. Conclusion
-

1. Introduction

The **Showroom Management System** is a software application designed to manage a car showroom efficiently. It provides functionalities such as adding new cars, removing cars, booking cars, filtering cars by price, and viewing all available vehicles in the database. The system ensures smooth showroom operations and allows customers to check car availability.

2. Project Overview

Objective

The objective of this project is to create an efficient and user-friendly system that automates car showroom operations, reducing manual work and enhancing customer experience.

Scope

This system is designed for showroom managers and customers to register, book, and filter cars based on their requirements.

3. System Features

The **Showroom Management System** includes the following key features:

- ✓ **Car Registration** – Add new cars with details like make, model, price, color, type, and availability.
 - ✓ **Car Removal** – Remove cars from the database based on registration numbers.
 - ✓ **Car Booking** – Book available cars, updating their availability status.
 - ✓ **View Car Database** – Display all registered cars along with their details.
 - ✓ **Filter Car Prices** – Search for cars within a specified price range.
-

4. Software Requirements

- **Programming Language:** Python
 - **Database:** CSV or SQLite
 - **Libraries Used:** Pandas (for database management), OS (for file handling)
-

5. System Workflow

User Interaction Steps:

1. The user is greeted with a main menu containing six options.
 2. Based on the user's choice, they can register a car, remove a car, book a car, view the database, or filter by price.
 3. The system performs the requested action and updates the database accordingly.
 4. The user can continue using the system or exit.
-

6. Implementation Details

The system follows a structured approach with object-oriented programming (OOP). It includes a **Car class** that manages all car-related operations.

Main Functionalities:

- `add_car()` – Registers a new car.
 - `remove_car()` – Deletes a car by registration number.
 - `book_car()` – Updates car availability status.
 - `view_cars()` – Displays all available cars.
 - `filter_cars_by_price()` – Searches for cars within a given price range.
-

7. Database Structure

The system stores car details in a CSV file or a database with the following columns:

Column Name	Description
Number	Unique car registration ID
Make	Car manufacturer

Column Name	Description
Name	Car model name
Model	Car model year
Price	Car price in USD
Color	Car color
Car Type	Sedan, SUV, Hatchback, etc.
Availability	Available / Booked

User Guide

How to Use the System:

1. Run the Program

- Open the terminal and run the Python script.
- The system displays a main menu with available options.

2. Register a Car

- Select **Option 1** and enter car details.
- The system adds the car to the database and displays the updated list.

3. Remove a Car

- Select **Option 2** and enter the car's registration number.
- The system removes the car and updates the database.

4. Book a Car

- Select **Option 3** and enter the car's registration number.
- The system marks the car as **Booked**.

5. View Cars

- Select **Option 4** to see all available cars.

6. Filter by Price

- Select **Option 5** and enter the minimum and maximum price range.
- The system displays cars within that price range.

7. Exit the Program

- Select **Option 6** to quit the system.
-

License

This project, **Showroom Management System**, is licensed under the following terms:

- This software is provided **for educational purposes only**.
- You may **use, modify, and distribute** the code for learning, research, and personal projects.
- **Commercial use, resale, or redistribution** of this project without explicit permission is prohibited.
- The author is **not liable** for any misuse, data loss, or system errors arising from the use of this software.

By using this software, you agree to these terms.