Car Rental System

A command-line car rental system simulation program written in Python.

Project Description

This project is a command-line car rental system simulation program developed using Python. It aims to provide a simple and easy-to-use platform for simulating the car rental business process, including user registration and login, vehicle management, lease application and management, and more.

Key Features:

- **Multi-language Support:** Supports Chinese, English, and Japanese interfaces for users of different languages. If you want to add more supported languages, you can easily copy an exist language file and translate sentences in the file .
- **Menu-driven Interface:** Employs a clear command-line menu interaction method, making operations simple and intuitive.
- **Role-based Access Control:** Differentiates between administrator and regular user roles, providing different functional permissions.
- **Complete Functionality:** Implements core functionalities such as user management, vehicle management, lease application management, lease details query, and **customer car browsing and booking**.
- **Modular Design:** Clear code structure with modular design, easy to understand and extend.

Features

- **User Management:**
  - User registration and login
  - Create, delete, update, and query user information (Administrator Role)
- **Vehicle Management (Administrator Role):**
  - Create, delete, update, and query vehicle information
- **Lease Application Management (Administrator Role):**
  - Query lease applications
  - Audit lease applications
- **Lease Details Management (Administrator Role):**
  - Query lease details
- **Customer Features (Regular User Role):**
  - View available car list
  - Book a car, automatically compute the lease fee based on the lease period, daily rent and extra fee
  - View personal lease history

Project Structure

```
car_rental_system/
```

```
├── config.py          # Configuration file (default db configurations ...)
├── main.py            # Main program entry point
├── models/            # Data models module
│   ├── action.py      # Action class (menu option actions)
│   ├── menu.py        # Menu class
│   └── ...            # Other data models (e.g., User, Car, LeaseDetails, etc. - if any)
├── service/           # Business logic processing package (user service, vehicle service, lease service, etc.)
│   ├── car_admin.py
│   ├── user_admin.py
│   └── ...            # Other service modules (e.g. lease_admin.py, etc.)
├── languages/         # Language packages directory
│   ├── language_zh.py # Chinese language package
│   ├── language_en.py # English language package
│   └── language_jp.py # Japanese language package
├── README.md          # README file
└── ...                # Other files (e.g., requirements.txt, .gitignore, etc.)
```

Getting Started

Prerequisites

- **Python 3.x:** Ensure you have Python 3.x installed. You can download and install it from the official Python website.

Installation

1. **Clone the repository:**

```
git clone https://github.com/moumchen/car_rental_system.git
cd car_rental_system
```

2. **(Optional) Create and activate a virtual environment: It is recommended to use a virtual environment to isolate project dependencies.**

```
python -m venv venv  # Create a virtual environment (if venv is not installed, install it first: pip install virtualenv)
source venv/bin/activate  # Activate virtual environment (Linux/macOS)
venv\Scripts\activate  # Activate virtual environment (Windows)
```

3. **Install dependencies:**

```
pip install -r requirements.txt
```

4. **Run the program:**

```
python main.py
```

5. **Optional - Initialize database:**

```
python db_init.py
```

After starting, the program will prompt you to select a language (zh/en/jp) and then enter the main menu.

Usage

1. **Run main.py**: Start the car rental system.
2. **Select Language**: Choose your preferred language (Chinese, English, or Japanese) as prompted.
3. **Main Menu**: After entering the main menu, you can choose to register, log in, or exit the system. Administrator can add other administrators, and default administrator account is `admin` with password `123456`.
4. **Login**: After logging in, you will enter different menu interfaces based on your user role (administrator or regular user).
5. **Menu Operations**: Follow the menu prompts to select and perform corresponding functions.

Dependencies

To ensure the program runs correctly, the following dependencies are required:

- **tabulate:** For displaying data in a tabular format.
