Ouro, Technical Challange, Nidal Abu-Raed 29.3.2024

Instructions and documentation

This document describes the Dog Spa Reservation System, and how to deploy, use and test it with examples.

The reason why there are two Python scripts (mainCLI.py, mainREST.py), is for easier testing and demonstration purposes of the system features and functionalities.

mainCLI.py is easier for testing functionality of the system (like reservation files writing) with CLI way of interacting.

mainREST.py is convenient for testing the REST API functionality of the system (like HTTP POST request testing) and to execute Robot test of the system.

Here are the instructions for deployment, usage, and running the mainREST.py and corresponding Robot tests:

Deployment and Usage Instructions - mainREST.py

1. Install or Download the package:

Download the system artifacts from the email (.zip) and save it to you locarl machines, some directory

2. Install Dependencies:

Navigate to the directory where the package was extracted and install the required dependencies (requirements.txt works for both scripts, mainCLI.py and mainREST.py):

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

3. Run the API Server:

Start the API server by running the main.py script:

python mainREST.py

This will start the API server, which will be accessible at http://localhost:8080.

4. Interacting with the API:

Use any HTTP client (e.g., curl, Postman) to interact with the API. Here are some example requests:

- Making a reservation:

curl -X POST http://localhost:8080/make_reservation -H "Content-Type: application/json" -d '{"start_time": "2023-11-06T09:00:00"}'

- Canceling a reservation:

curl -X POST http://localhost:8080/cancel_reservation -H "Content-Type: application/json" -d '{"start_time": "2023-11-06T09:00:00"}'

- Viewing reservations:

curl http://localhost:8080/view_reservations

Running Robot Tests

1. Install Robot Framework:

If you haven't already, install Robot Framework using pip:

pip install robotframework

2. Run the Robot Tests:

Execute the Robot tests using the robot command:

robot -d results DogSpaReservationSystem/

This command will run all the tests located in the DogSpaReservationSystem/ directory and store the results in the results directory.

With these instructions, users can deploy the Dog Spa Reservation System (mainREST.py), interact with it through HTTP requests, and run the Robot tests with a single command.

Here are the instructions for deployment, usage, and running the mainCLI.py:

Deployment and Usage Instructions - mainCLI.py

1. Install or Download the package:

Download the system artifacts from the email (.zip) and save it to you local machines, some directory

2. Install Dependencies:

Navigate to the directory where the package was extracted and install the required dependencies (requirements.txt works for both scripts, mainCLI.py and mainREST.py):

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

3. Run the mainCLI.py:

Start the CLI based script by running the mainCLI.py script:

```
python mainCLI.py
```

This will start the script, and provide CLI based input queries for the user to interact with the application

4. Interacting with the functionalities:

There are the following functionalities in mainCLI.py script

- Make reservation (saves reservation to a JSON file)
- Cancel Reservation (deletes reservations from a JSON file)
- View Reservation (retrieves reservations from a JSON file)

With these instructions, users can deploy the Dog Spa Reservation System (mainCLI.py), and interact with it using CLI inputs, and test the some of its functionalities.

Documentation - Dog Spa Reservation System

The Dog Spa Reservation System is a command-line application written in Python. It allows users to make reservations for the dog spa.

Features:

1. Make Reservation: Users can make reservations for the dog spa.
2. Cancel Reservation: Users can cancel existing reservations.
3. View Reservations: Users can view the current reservations.
4. Save and Load Reservations: Reservations are saved to a JSON file, allowing the system to continue from the previous state after a restart.

Usage:

1. Initialization: When the application starts, it loads existing reservations from the reservations.json file. If the file does not exist or is empty, the system starts with an empty reservations list.

2. Making a Reservation: Users can make a reservation by providing the desired start time. The reservation duration is fixed at 50 minutes.

3. Canceling a Reservation: Users can cancel an existing reservation by specifying the start time of the reservation to be canceled.

4. Viewing Reservations: Users can view the current reservations, including the start and end times of each reservation.

5. Saving Reservations: After making any changes to the reservations (e.g., making or canceling a reservation), the updated reservations list is saved to the reservations.json file.

6. Exiting the System: Users can exit the system by selecting the appropriate option from the menu.

Technologies Used:

- Python: The application is written in Python programming language.
- JSON: Reservations are saved to and loaded from a JSON file for persistence.
- Flask: Application server to host HTTP requests.

Dependencies:

- datetime: Used for handling date and time operations.
- json: Used for reading and writing data in JSON format.
- flask: Used for handling HTTP requests.

Example Usage:

Welcome to the Dog Spa Reservation System!

1. Make a Reservation
2. Cancel a Reservation
3. View Reservations
4. Exit

Enter your choice: 1
Enter the desired start time (YYYY-MM-DD HH:MM:SS): 2023-11-08 10:00:00
Reservation successful!

1. Make a Reservation
2. Cancel a Reservation
3. View Reservations
4. Exit

Enter your choice: 3
Current Reservations:
1. Start Time: 2023-11-08 10:00:00, End Time: 2023-11-08 10:50:00

1. Make a Reservation
2. Cancel a Reservation
3. View Reservations
4. Exit

Enter your choice: 4
Exiting the Dog Spa Reservation System. Goodbye!

This documentation provides a comprehensive overview about using and understanding the Dog Spa Reservation System.
