Documentation for SmartPanda Restaurant Management System

1. Project Overview

1.1 Objective

The SmartPanda Restaurant Management System is a Python-based CLI - Command Line Interface program designed to provide an interactive and user-friendly restaurant system. The system allows you to easily place, view, update, and cancel orders. It has a voice assistant system to guide customers, though it's very basic. The primary goal is to simplify restaurant operations and practice real-life programs.

1.2 Features

- **Menu Navigation**: Users can easily browse the restaurant's menu, place orders, view their order history, update existing orders, and cancel orders.
- **User Authentication**: The system allows users to log in and register. It stores user data in a simple format (JSON) and manages login sessions to keep users secure.
- **Inventory Management**: Users can manage restaurant inventory by placing, updating, and canceling orders, ensuring accurate stock levels.
- **Order Management**: The system allows users to perform various operations such as placing, updating, and canceling orders.
- **Voice Interaction**: Users can interact with the system using voice commands to guide them through the menu.
- **Error Handling**: The system has a robust error handling system.

2. Design Decisions

2.1 File Structures

- **Common Utilities**: The `utilities/common.py` module contains utility functions for tasks like clearing the console, showing loading messages, and handling user input.
- **Dashboard**: The `app.py` module contains all dashboard logics and menus based on users logged in such as admin, manager, staff and customers.
- **User Authentication**: The `user_authentication/user_auth.py` module manages user login, registration, session, and authentication.
- **Inventory Management**: The `inventory_management/inventory.py` module manages product-related operations.
- **Frontend Management**: The `frontEnd/frontend.py` module handles the customer order related operations
- **Speech Recognition**: The `voice_ordering/panda_assitant.py` library is used to capture and process user input from the microphone. It converts speech to text, which is then processed by the system.

3. Key Modules

3.1 User Authentication

Manages user login, registration, and authentication to ensure secure access to the system.

3.2 Common Utilities

Contains utility functions for tasks such as clearing the console, displaying loading messages, and handling user input.

3.3 Main Dashboard System

Serves as the central hub for navigating through the system's functionalities, including order management and inventory.

3.4 Inventory Management

Handles the management of restaurant inventory, tracking stock levels, and ensuring accurate order fulfillment.

3.5 Frontend Management

Manages the user interface, providing an intuitive interaction with the restaurant's order data and system functionalities.

4. Testing Procedures

4.1 Input Testing

Unit tests were written to verify the core functionalities of the system, including:

- **Correct Input Value**: The system used `RegularExpression` to validate user input.
- **Command handling**: Testing that the system correctly handles commands and redirects the user to the appropriate pages.

4.2 Manual Testing by Sample Data

User, products, and orders all have been done with testing and providing sample data manually. We tested each and every function and inputs available in the project

5. Challenges Encountered

We have encountered huge number of issues specially input validation, code conflicting with others' code, etc. Another problem we encountered in implementing voice operation to work with the system, so we tried to leave it for our future improvement.

6. Additional Resources:

We have used a number of resources from online for solving problems we encountered during the work. Specially we used, Google, StackOverflow, Python.org, ChatGPT, and Github

7. Future Work

- **GUI Integration**: Adding a graphical user interface (GUI) for users who prefer visual interaction.
- **Enhanced Command Recognition**: Integrating more advanced natural language processing (NLP) techniques to improve the system's ability to understand complex commands.
- **Multi-Language Support**: Expanding the system to support multiple languages for a more diverse user base.

8. Conclusion

The Smart Panda Restaurant Management System is a small but practical real-life solution if we can improve the project more then it can be a great solution for restaurant system.