COCO GARDEN MANAGER – A CUSTOM RESTAURANT RESERVATION & ORDERING SYSTEM

Yuesong Huang
Team Number: 25
Member ID: 1
yhu116@ur.rochester.edu

Xingzhi Ye
Team Number: 25
Member ID: 2
xye8@u.rochester.edu

1 PROJECT DETAIL

Project Name: Coco Garden Manager

This project aims to develop a customized Restaurant Reservation and Ordering System specifically for Coco Garden, a Chinese restaurant. The system will address the limitations of the current outsourced solution by providing a flexible, user-friendly, and cost-effective database for managing reservations, menu items, customer orders, and customer interactions. The Coco Garden Manager will improve operational efficiency and enhance customer service, allowing the restaurant to offer customized services.

2 PROBLEM STATE

Coco Garden, a Chinese restaurant, currently uses an outsourced restaurant management system that is expensive and poorly suited to its needs. The current system creates inefficiencies in handling reservations, table assignments, menu management, and customer orders. Additionally, the high cost puts financial pressure on the business without delivering the expected performance.

This project, *Coco Garden Manager*, will provide a custom solution to manage core restaurant operations, including reservations, table assignments, menu items, customer orders, and membership systems. The new system will streamline processes, offer flexibility for customization, and improve the overall customer experience. Coco Garden can reduce inefficiencies, provide better customer service, and save on the high costs associated with outsourced systems by switching to this custom solution.

3 TARGET USERS[4]

The system will cater to two main types of users, each with specific roles and permissions:

- **Admin[1] (Owner)**:
  - **Privileges:**: Full access to all aspects of the system. Can manage other users, including managers and staff, view and modify reservations, orders, menu items, inventory, and customer information. The admin can also generate reports and analyze restaurant performance.
  - **Use Cases:**: Overseeing the overall operation of the restaurant, managing financial and operational data, setting up loyalty programs, and customizing the system as needed.
- **Merchant[2] (Manager/Staff)**:
  - **Privileges:**: Merchants, including managers and staff, have access to day-to-day operations such as managing reservations, orders, menu items, and inventory. Managers can also handle customer inquiries and support but have limited access to user management and financial data.

  - **Use Cases:**: Handling daily operations, overseeing staff, ensuring smooth restaurant functioning, and managing customer interactions.
- **Customer[3] and VIP[4]**):
  - **Privileges:**: Limited access to the system. Can make reservations, place orders, and view menu items. Customers can also manage their profiles and view loyalty points and membership balances.
  - **Use Cases:**: Making reservations, placing orders, checking membership balances, and participating in the loyalty program.

4 List of Relations

The database consists of several key tables, each designed to manage users, reservations, menu items, orders, and other essential operations:

- **Users**: Stores login information for both merchants and customers.
  Schema: `USER(user_id: integer, username: string, password: string, role: enum('Administrator', 'Merchant', 'Customer'))`
- **Reservations**: Manages customer reservations and tracks details like date, time, party size, and table number.
  Schema: `RESERVATIONS(reservation_id: integer, customer_id: integer, reservation_date: date, reservation_time: time, party_size: integer, table_number: integer, status: enum('Pending', 'Confirmed', 'Completed', 'Cancelled'))`
- **MenuItems**: Stores details of menu items, including category, price, and availability.
  Schema: `MENUIEMS(item_id: integer, item_name: string, description: text, category: enum('Appetizer', 'Main Course', 'Dessert', 'Drink', 'Special'), price: real, availability: boolean)`
- **Orders**: Logs customer orders, tracking the total amount and status.
  Schema: `ORDERS(order_id: integer, customer_id: integer, order_date: date, total_amount: real, status: enum('Placed', 'In Progress', 'Completed', 'Cancelled'))`
- **OrderDetails**: Stores details of individual items within an order.
  Schema: `ORDERDETAILS(order_detail_id: integer, order_id: integer, item_id: integer, quantity: integer, price: real, special_instructions: text)`
- **Customers**: Stores customer information, such as contact details, membership balances, and loyalty points.
  Schema: `CUSTOMER(customer_id: integer, customer_name: string, phone_number: string, email: string, membership_balance: real, loyalty_points: integer)`
- **LoyaltyProgram**: Defines the loyalty program, including how points are earned and redeemed.
  Schema: `LOYALTYPROGRAM(loyalty_id: integer, points_earned_per_dollar: real, points_to_dollar_ratio: real, redemption_threshold: real)`
- **Support**: Captures customer support inquiries.
  Schema: `SUPPORT(support_id: integer, name: string, email: string, phone_number: string, message: text)`

5 Web Interface

The web interface for Coco Garden Manager is simple and intuitive, designed to streamline restaurant operations. It includes features such as customer reservations, order management, and menu

viewing. The layout is designed to ensure both customers and staff can easily navigate and perform tasks efficiently.

- A live demonstration can be accessed at https://anakinhuang.github.io/.
- You can also see the end of our project proposal for our web interface designs.

6 DATA

To populate the database for Coco Garden Manager, I plan to use a combination of real-world data and synthetic data:

- **Menu and Inventory Data**: This will be collected directly from the owner of Coco Garden, including the actual menu items with descriptions, prices, and stock levels. This data is not sensitive and will help create a realistic dataset for the system.
- **Customer and Financial Data**: For more private information, such as customer details, bank statements, and raw material purchase prices, I will generate synthetic data using a program that creates realistic customer names, emails, membership balances, and loyalty points. Additionally, I will simulate reservations, orders, and inventory restocks to ensure the system is functional without using real private data.
- **Surveys for Volunteer Data**: I plan to distribute surveys to gather anonymized customer data from volunteers at other restaurants. This will help improve the accuracy of the customer-facing system.

By using a mix of real data from Coco Garden and synthetic data, the system will be fully populated while respecting privacy concerns.

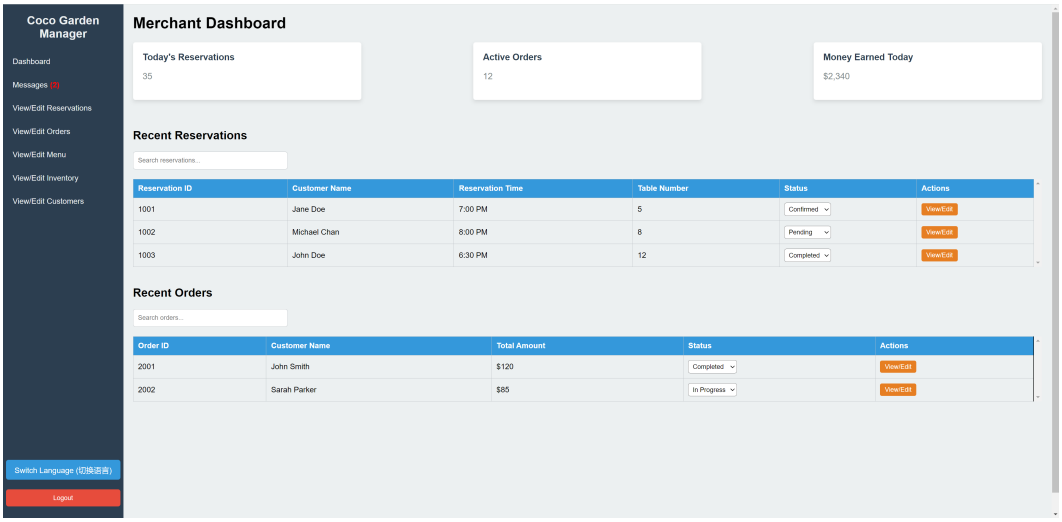

Figure 1:

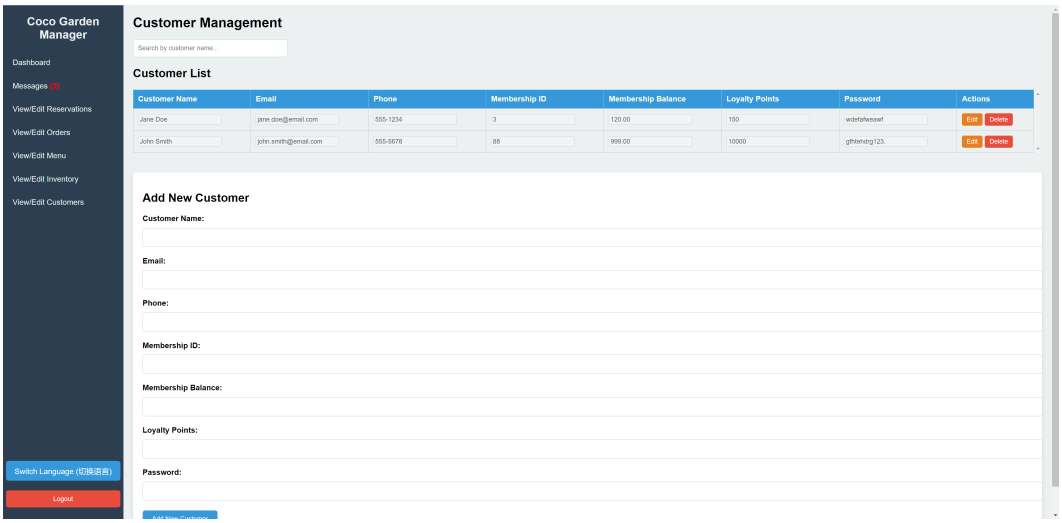

Figure 2:

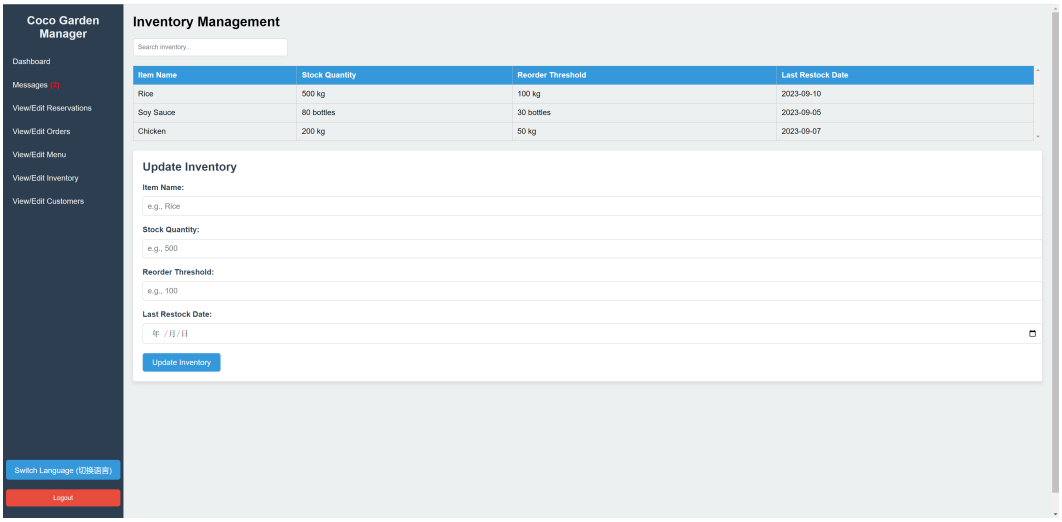

Figure 3:

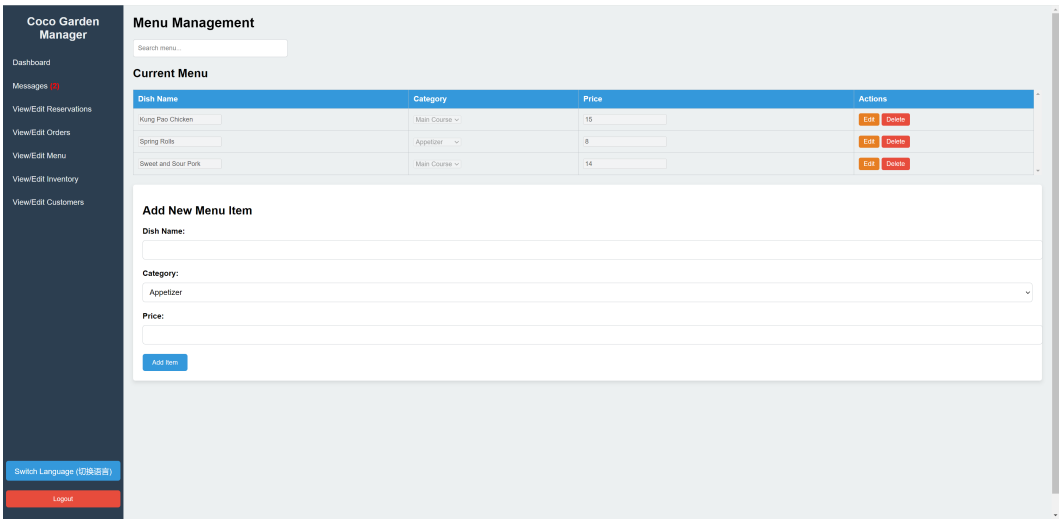

Figure 4:

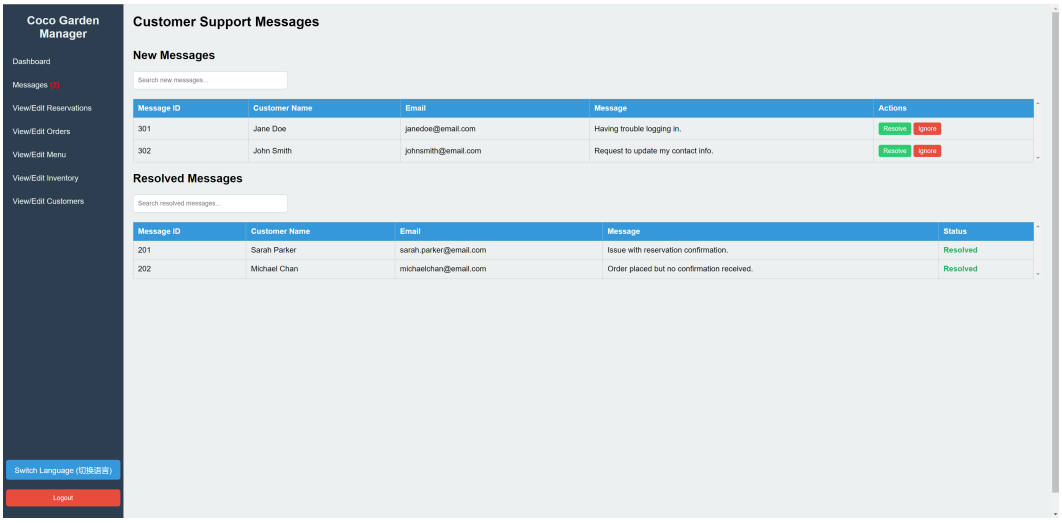

Figure 5:

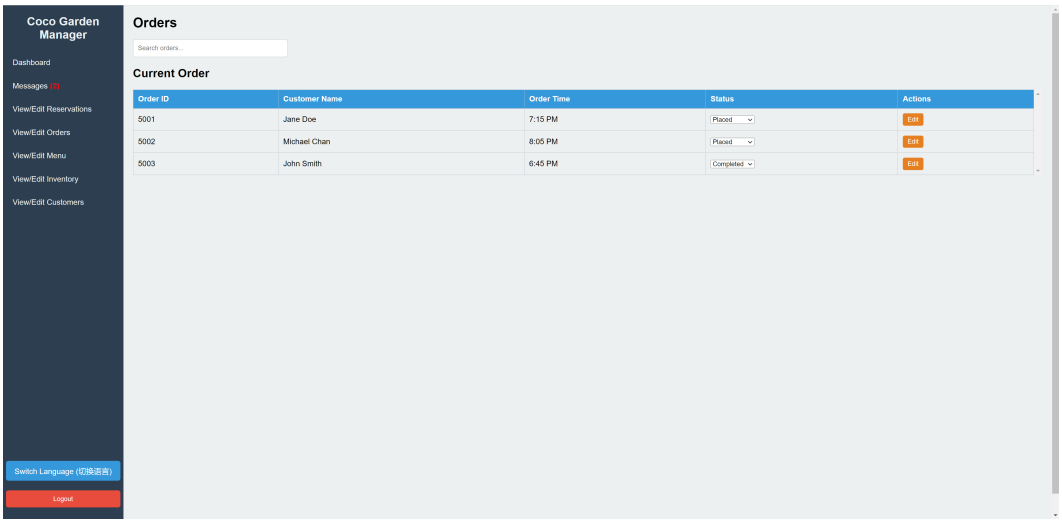

Figure 6:

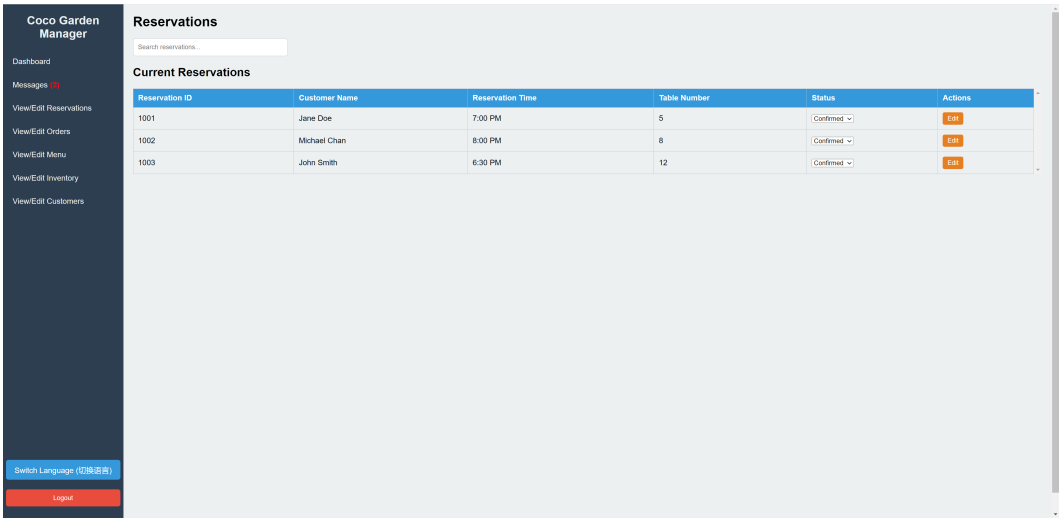

Figure 7: