Godwin

**Restaurant Management System (RMS)**

The **Restaurant Management System (RMS)** is designed to manage the operations of a restaurant. It will handle tasks such as customer orders, menu management, staff scheduling, table reservations, and billing. The system is meant to improve the efficiency of restaurant management and provide a better customer experience by streamlining these processes.

**Features and Functional Specifications**

1. **Login**:
   * The system will allow **three types of users** to log in:
     + **Admin**: Full control over the system. Can manage menus, staff, and generate reports.
     + **Waitstaff**: Can take customer orders, assign tables, and generate bills.
     + **Chef**: Can view orders and mark them as prepared.
   * **Login Requirements**:
     + **Username** and **Password** authentication.
     + User roles should be identified to provide appropriate access to features.
2. **Add Menu Item**:
   * Admin can add a new item to the menu.
   * Each menu item will include:
     + **Item ID**: Unique identifier for each menu item.
     + **Item Name**: Name of the dish.
     + **Description**: A brief description of the dish.
     + **Price**: The cost of the dish.
     + **Category**: (e.g., Appetizer, Main Course, Dessert).
   * **Validation**:
     + Ensure all fields are filled correctly.
     + Ensure **Item ID** is unique.
3. **Modify Menu Item**:
   * Admin can modify details of an existing menu item.
   * The menu item can be selected by **Item ID**, and then updated with new information.
4. **View Menu**:
   * Customers and staff can view the restaurant menu, which will display:
     + **Item Name**, **Description**, **Price**, and **Category**.
   * The system will support filtering by category (e.g., appetizers, desserts).
5. **Place Order**:
   * Waitstaff can take orders from customers, specifying:
     + **Table Number**: The table where the order is being placed.
     + **Menu Items**: A list of selected dishes.
     + **Quantity**: How many servings of each dish.
   * The system will record the order and send it to the kitchen for preparation.
6. **View Orders**:
   * Waitstaff can view the orders placed at any given time.
   * Orders will be displayed with:
     + **Table Number**, **Menu Items**, **Quantity**, and **Status** (Pending, Prepared, Served).
7. **Prepare Order**:
   * The chef will receive orders and mark them as **Prepared** once completed.
   * The chef can also view:
     + **Order ID**, **Table Number**, **Menu Items**, and **Status**.
8. **Generate Bill**:
   * Waitstaff can generate a bill for a table after the customer has finished their meal.
   * The bill will include:
     + **Menu Items**, **Quantity**, **Price**, and **Total Amount**.
     + Any discounts or promotions applied.
   * The system will calculate the total and allow for **payment** processing.
9. **Table Reservation**:
   * Customers can reserve tables in advance, specifying:
     + **Date and Time**, **Number of People**.
   * The system will check for availability and confirm the reservation.
   * Reserved tables will be displayed on the **reservation list** with status (e.g., Confirmed, Cancelled).
10. **View Reservations**:
    * Admin and waitstaff can view the table reservations for the day.
    * Display the **Customer Name**, **Table Number**, **Time**, and **Status** (Confirmed, Cancelled).
11. **Change Password**:
    * Users can change their login password.
    * The new password must meet security requirements (e.g., minimum length, mix of characters).
    * **Current Password** should be verified before updating.
12. **Logout**:
    * Ends the current session and returns the user to the login screen.

**Layout Design**

**1. Login Screen:**

**Purpose**: Authenticate users and provide access based on roles.

**Design**:

markdown

Copy code

-------------------------------------------------

| RESTAURANT MANAGEMENT SYSTEM |

-------------------------------------------------

| LOGIN |

| |

| Username: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| Password: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| |

| [Login] [Exit] |

-------------------------------------------------

**Functionality**:

* Username and password validation.
* Redirect to the main menu based on user role (Admin, Waitstaff, Chef).

**2. Main Menu (After Login):**

**Purpose**: Provide access to all system functionalities.

**Design**:

css

Copy code

-------------------------------------------------

| RESTAURANT MANAGEMENT SYSTEM |

-------------------------------------------------

| Welcome, [User Name]! |

-------------------------------------------------

| [1] Add Menu Item |

| [2] Modify Menu Item |

| [3] View Menu |

| [4] Place Order |

| [5] View Orders |

| [6] Prepare Order |

| [7] Generate Bill |

| [8] Table Reservation |

| [9] View Reservations |

| [10] Change Password |

| [11] Logout |

-------------------------------------------------

**3. Add Menu Item Form:**

**Purpose**: Collect details for a new menu item.

**Design**:

markdown

Copy code

-------------------------------------------------

| ADD NEW MENU ITEM |

-------------------------------------------------

| Item ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| Item Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| Description: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| Price: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| Category: [Appetizer/Main Course/Dessert] |

-------------------------------------------------

| [Save] [Cancel] |

-------------------------------------------------

**Functionality**:

* All fields should be validated (non-empty).
* **Item ID** must be unique.
* Saves data to a file (menu.dat).

**4. Modify Menu Item:**

**Purpose**: Edit an existing menu item.

**Design**:

markdown

Copy code

-------------------------------------------------

| MODIFY MENU ITEM |

-------------------------------------------------

| Enter Item ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

-------------------------------------------------

| [Search] |

-------------------------------------------------

* **After Search**:

markdown

Copy code

-------------------------------------------------

| Item Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| Description: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| Price: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| Category: [Appetizer/Main Course/Dessert] |

-------------------------------------------------

| [Save Changes] [Cancel] |

-------------------------------------------------

**5. View Orders:**

**Purpose**: Display current orders.

**Design**:

markdown

Copy code

-------------------------------------------------

| CURRENT ORDERS |

-------------------------------------------------

| Order ID | Table Number | Menu Items | Status |

-------------------------------------------------

| O001 | 1 | Pizza, Pasta | Pending |

| O002 | 2 | Salad, Soup | Prepared|

-------------------------------------------------

| [Back] |

-------------------------------------------------

**6. Generate Bill:**

**Purpose**: Generate and display the bill for a table.

**Design**:

markdown

Copy code

-------------------------------------------------

| GENERATE BILL |

-------------------------------------------------

| Table Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| Menu Items: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| Quantity: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| Total Amount: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| [Generate Bill] [Print Bill] |

-------------------------------------------------

**Functionality**:

* Calculate total based on the menu items, quantity, and price.
* Display any discounts or tax calculations.

**7. Table Reservation Form:**

**Purpose**: Reserve a table for a customer.

**Design**:

markdown

Copy code

-------------------------------------------------

| TABLE RESERVATION |

-------------------------------------------------

| Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

| Date & Time: [YYYY-MM-DD] [HH:MM] |

| Number of People: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

-------------------------------------------------

| [Reserve] [Cancel] |

-------------------------------------------------

**Flow Between Layouts**

1. **Login Screen → Main Menu**:
   * Upon successful login, users are directed to the main menu.
2. **Main Menu → Feature Forms**:
   * Selecting an option from the main menu takes the user to the corresponding feature (Add Menu Item, Place Order, etc.).
3. **Forms → Success/Failure Messages**:
   * Upon form submission (e.g., Add Menu Item, Table Reservation), display success or failure messages.
4. **Logout**:
   * Users can log out to return to the login screen.

**Development Plan**

**Days 1-2: Requirements Analysis and Database Design**

* Define all required features and workflows.
* Create file structures for storing menu, orders, and reservations (e.g., menu.dat, orders.dat).

**Days 3-5: Core Feature Implementation**

* Implement login functionality.
* Create features for adding, modifying, and viewing menu items.

**Days 6-7: Customer Orders and Reservations**

* Develop the ordering system and reservation management.
* Ensure table availability checks for reservations.

**Days 8-9: Billing and Reporting Features**

* Implement bill generation, including discounts and taxes.
* Add reservation status and reporting features.

**Day 10: Testing and Debugging**

* Perform unit tests for each module and fix bugs.
* Test edge cases like invalid input and missing fields.

**Resources**

**Programming Language: C Language**

* C offers efficient file handling for storing orders, menus, and reservations.

**Tools:**

* **IDE**: Code::Blocks or Dev-C++.
* **Compiler**: GCC or any compatible C compiler.

**Testing:**

* Manual testing of each module.
* Check for boundary conditions, e.g., invalid date formats or out-of-range quantities.

This **Restaurant Management System** aims to automate the core operations of a restaurant, improving efficiency and enhancing the customer experience. The use of C ensures low-level control over file handling and data storage, making the system lightweight and reliable.