UE22CS351A: Software Engineering

Project Name: Restaurant Management System

PRANEEL PATEL (PES1UG22CS437) &

PRIYANSHU CHAUDHARY (PES1UG22CS455)

H-Section, 5TH Semester

BTech-CSE

Restaurant Management System - Architectural Document

1. Introduction

1.1 Purpose

This document provides a high-level architectural overview of the Restaurant Management System (RMS). The system manages restaurant operations including menu management, table reservations, orders, and billing while ensuring a smooth workflow for both admins and customers.

1.2 Scope

The RMS facilitates restaurant management through features such as menu handling, order processing, inventory tracking, and payment management. This document is intended for system architects, developers, and restaurant administrators responsible for developing and maintaining the system.

1.3 Definitions, Acronyms, and Abbreviations

- **RMS**: Restaurant Management System

- **GUI**: Graphical User Interface
- **DBMS**: Database Management System
- **SQL**: Structured Query Language
- **POS**: Point of Sale

---

2. Architectural Representation

The RMS follows a **layered architecture** comprising three primary layers:

- **Presentation Layer:** Interfaces with users (both staff and customers).
- **Business Logic Layer:** Handles order processing, inventory updates, and billing.
- **Data Access Layer:** Manages interactions with the database.

The system implements the **Model-View-Controller (MVC)** design pattern, which separates user interface, data, and logic.

---

3. Architectural Goals and Constraints

3.1 Goals

- **Usability: The system should provide a user-friendly interface for both restaurant staff and customers.**
- **Efficiency: Ensure efficient order processing and fast inventory updates.**
- **Security: Use secure authentication and data encryption.**
- **Data Integrity: Real-time updates of order statuses and inventory levels.**

3.2 Constraints

- **Built using Python for logic, Tkinter for GUI, and MySQL for database management.**
- **All sensitive data, including customer payments, must comply with secure handling and transmission protocols.**

---

4. Use-Case View

4.1 Significant Use Cases

1. **User Authentication**
    - **Actors: Restaurant Admin**
    - **Description: Admins must log in using secure credentials to manage the restaurant's operations.**
    - **Architectural Impact: Secure credential storage and authentication system.**
2. **Menu Management**
    - **Actors: Admin**
    - **Description: Admins can add, update, or delete menu items and modify prices.**
    - **Architectural Impact: CRUD operations on the menu database.**
3. **Order Processing**
    - **Actors: Waiter, Customer**
    - **Description: Customers place orders, and the system processes them, updating inventory and generating bills.**
    - **Architectural Impact: Atomic transactions across orders, inventory, and billing.**
4. **Inventory Management**
    - **Actors: Admin**
    - **Description: Admins can manage ingredients and track inventory levels.**
    - **Architectural Impact: Real-time updates to inventory upon order placement.**
5. **Billing and Payment**
    - **Actors: Customer, Admin**
    - **Description: Customers can view and pay bills via the system.**

- **Architectural Impact: Integrates payment gateways and updates transaction records.**

---

5. Logical View

5.1 Package and Subsystem Layering

The system is divided into several packages:

- **Authentication Package: Manages admin and staff logins.**
- **Menu Package: Manages CRUD operations for menu items.**
- **Order Processing Package: Processes customer orders, updates inventory, and generates bills.**
- **Inventory Package: Manages ingredient availability for menu items.**
- **Billing Package: Handles payment processing and transaction logging.**

5.2 Process View

1. **User Interaction Process**
   - **Interfaces with the Presentation Layer, including customer order forms and admin dashboards.**
2. **Order and Inventory Process**
   - **Business logic layer handles order validation and inventory reduction.**
3. **Billing and Payment Process**
   - **Manages payment status updates and stores transaction records.**

5.3 Process Model to Design

- **Each process is mapped to the design layer it interacts with. For example, an order placed by the customer triggers the Presentation Layer, which communicates with the Business Logic Layer and updates the database.**

---

6. Deployment View

The system will be deployed in a restaurant environment with the following components:

- **Client-Side (Admin/Customer Interaction): Runs on tablets or desktop systems for both the customer (to place orders) and the staff (to manage orders).**
- **Server-Side (Database and Business Logic): Hosted on a local or cloud-based server, handling menu, orders, inventory, and payment processing.**

---

7. Performance

- **Response Time: All operations, such as order processing and inventory updates, should be completed within 1-2 seconds.**
- **Scalability: The system must handle increasing numbers of orders and menu items as the restaurant grows.**

---

8. Quality

8.1 Usability

The system should be intuitive and easy to use for staff and customers, with a well-organized GUI.

8.2 Security

- **Password Protection: Admin and employee logins must be secure with encrypted passwords.**
- **Payment Security: Sensitive payment information must be encrypted.**

8.3 Reliability

The system must ensure high availability and data integrity, particularly for transactions and inventory updates.
