Practical No. 01

Team Members:
Member 1 : Saylee Raut **Member 2 : Tanzila Taherin Sheikh**

1) Aim of the experiment

To design and implement a Restaurant Management System that automates the daily operations of a restaurant including order management, billing, and inventory control, to enhance efficiency and customer satisfaction.

---

2) Problem Statement

Managing a restaurant manually is time-consuming and prone to errors, especially in handling orders, billing, and inventory management. There is a need for a software system that can streamline these processes, reduce human errors, improve accuracy, and provide timely reports for better decision-making.

---

3) Objective

- To develop a user-friendly system that manages restaurant orders and billing efficiently.
- To automate inventory tracking and alert the staff when stock is low.
- To reduce waiting time for customers by speeding up order processing.
- To generate reports for sales, inventory, and customer feedback.
- To enhance overall restaurant management and improve customer satisfaction.

---

4) Introduction

A Restaurant Management System (RMS) is a software application designed to manage the operational aspects of a restaurant. This includes order taking, billing, kitchen management, and inventory control. RMS helps restaurant staff serve customers quickly and accurately while maintaining records of transactions and stock. By implementing such a system, restaurants can reduce manual errors, improve operational efficiency, and provide a better dining experience.

---

5) Theory

Characteristics of Requirements

- **Correctness:** Requirements must accurately represent the needs of users.
- **Completeness:** All possible scenarios and features must be covered.
- **Consistency:** Requirements should not contradict each other.
- **Unambiguous:** Clear and precise language should be used.
- **Verifiability:** Requirements must be testable.
- **Modifiability:** Easy to update or change as needed.
- **Traceability:** Ability to trace each requirement through the development process.

Categorization of Requirements

- **Functional Requirements:** What the system should do.
- **Non-functional Requirements:** System attributes like performance, usability, reliability.
- **Technical Requirements:** Hardware, software, and technology constraints.

Functional Requirements

- User registration and login.
- Table reservation management.
- Order placing and modification.
- Billing and invoice generation.
- Inventory management and stock updates.
- Report generation (daily sales, inventory status).
- User roles and access control (admin, staff, kitchen).

Non-functional Requirements

- Usability: Easy to use interface.
- Performance: Fast processing of orders and billing.
- Security: Secure login and data protection.
- Reliability: System uptime and data backup.
- Scalability: Ability to add more features or users.

- Maintainability: Easy to fix bugs and update the system.

Technical Requirements

- Programming language: Java/Python/C# (depending on your project).
- Database: MySQL/SQLite for storing data.
- Operating System: Windows/Linux/Mac.
- Development tools: IDE like Eclipse, Visual Studio, etc.

---

6) Software Requirement Specification (SRS)

Purpose

This document specifies the requirements for the Restaurant Management System to streamline restaurant operations and improve customer service.

Scope

The RMS will be used by restaurant staff to manage orders, billing, and inventory in real-time. It will support multiple user roles and generate reports to aid decision-making.

System Features

- **User Management:** Register, login, and role assignment.
- **Order Management:** Place, modify, and cancel orders.
- **Billing System:** Generate bills and process payments.
- **Inventory Management:** Track ingredient stock and send alerts.
- **Reports:** Generate sales and inventory reports.

External Interface Requirements

- User interface: Simple GUI or web interface.
- Hardware: Compatible with standard restaurant POS devices.

Constraints

- Must be compatible with the existing hardware.
- Must ensure data security and privacy.

Assumptions and Dependencies

- Reliable internet connectivity (if web-based).
- Staff will be trained to use the system.

---

7) References

- Software Engineering, Ian Sommerville, 10th Edition.
- Pressman, Roger S. Software Engineering: A Practitioner's Approach.
- Online resources on software requirement specification and restaurant management systems.
- Documentation and manuals for programming languages and database systems used.

---

8) Conclusion

The Restaurant Management System aims to automate and streamline the restaurant's day-to-day operations. By implementing this system, the restaurant will benefit from improved order accuracy, efficient billing, better inventory control, and enhanced customer satisfaction. This project demonstrates the application of software engineering principles in solving real-world problems effectively.
