

PENTLUX RESTAURANT MANAGEMENT SYSTEM DOCOMENTATION

COURSE CODE: OBJECT ORIENTED ANALYSIS DESIGN AND IMPLEMMENTATION

### GROUP NAME:GROUP 5

### PROJECT TOPIC: RESTAURANT MANAGEMENT SYSTEM

### LINK TO GITHUB REPOSATORY: <https://github.com/Sukuna52341/Pentlux-Restaurant-System>

GROUP LEADER: CHICK MARY CHELSEY ASEH

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| S/N | MEMBERS’NAME | REGISTRATION NUMBER | TEAM ROLE |
| 1 | CHICK MARY CHELSEY ASEH | ICTU20223089 | Scrum master |
| 2 | DEMMASSE SONITA CHAINA BRUNELLE | ictu20241131 | Team member |
| 3 | NGAH VALDES NKENG | ICTU20222940 | Team member |
| 4 | KOUAMOU STEPHANE LANDRY | ICTU20223392 | Team member |
| 5 | TEZI CARINE NKWENTEN | ICTU20223218 | Product owner |
| 6 | LONTUM ALAIN BRICE FOMONYUY | ICTU20223210 | Team member |

|  |
| --- |
|  |

**INTRODUCTION**

|  |
| --- |
|  |

Chapter One: Introduction

General Introduction

PentLux Restaurant Management System is a web application developed for PentLux restaurant. This application is designed to manage customers, orders, menus, and billing, providing an integrated solution to streamline restaurant operations.

Aim and Objectives

The primary aim of the PentLux Restaurant Management System is to create a platform that allows the restaurant to serve customers and manage their business online effectively. The objectives include:

• Enhancing communication between customers and the restaurant.

• Providing a secure and user-friendly digital platform for managing restaurant operations.

• Facilitating remote ordering to cater to the digital era.

Problem Statement

The project addresses the problem of inefficient communication and remote ordering for restaurants. By enabling customers to place orders from home, the system aims to increase customer convenience and expand the restaurant’s reach. Additionally, it helps the restaurant manage its operations more effectively and attract more customers through an online presence.

**Chapter Two: Literature Review**

Software Development Methodologies

We studied various software development methodologies, focusing on Agile and Scrum due to their relevance to our project.

Comparison between Different Software Development Methodologies

We compared several methodologies, including Waterfall, Agile, and Scrum. Waterfall follows a linear and sequential approach, which lacks flexibility. Agile, on the other hand, is iterative and incremental, allowing for adaptability and continuous improvement. Scrum, a subset of Agile, enhances teamwork and communication through defined roles and ceremonies, making it ideal for our project.

Reason for the Choice of Scrum Methodology

We chose the Scrum methodology for our project due to its ability to enhance team collaboration and communication. Scrum’s framework, with its sprints, daily stand-ups, and retrospective meetings, allows for regular feedback and continuous improvement, which are crucial for the success of our project.

General Review of Related Concepts

To develop an effective restaurant management system, we reviewed key concepts such as:

• Customer Relationship Management (CRM): Techniques for managing a restaurant’s interactions with current and potential customers.

• Order Management Systems: Methods to handle customer orders efficiently, from placement to fulfillment.

• Menu Management: Strategies for organizing and updating the restaurant’s menu items.

• Billing Systems: Secure and efficient methods for processing payments and generating bills.

Review of Related Literature

We reviewed similar projects to understand best practices and common challenges. Notable among them were:

• Cafe Management System: This project provided insights into handling orders and managing inventory in a cafe setting.

• Restaurant Web App: This study highlighted the importance of user-friendly interfaces and secure payment gateways in a restaurant management system.

Chapter Three: Methodology and Materials

Research Methodology

To gather requirements for the PentLux Restaurant Management System, we conducted interviews with potential users, including office personnel who frequently order lunch while at work, and with the restaurant staff. This helped us understand the needs and pain points of both customers and the restaurant management.

System Requirements

Functional Requirements:

• User Authentication: Secure login and registration for users.

• Order Placement: Functionality for users to place orders.

• Payment Processing: Integration of payment gateways for secure transactions.

• Billing and Reports: Generation and management of bills and reports.

• Menu Management: Categorization and management of the restaurant’s menu items.

Non-Functional Requirements:

• Security: Ensuring secure transactions and data protection.

• Performance: Efficient handling of multiple concurrent users and orders.

System Design

Architecture Description:

• Backend: Java Spring Boot was used for implementing backend functionalities and APIs to manage users, products, product categories, and billing.

• Database: MySQL was used for data storage and management.

• Frontend: HTML, CSS, and JavaScript were utilized to create a user-friendly interface.

(UML diagrams)

Application of Scrum

• Team Organization: Our team consisted of six members with defined roles:

• Scrum Master: Facilitated scrum ceremonies and ensured the team adhered to Scrum practices.

• Product Owner: Managed the product backlog and prioritized features.

• Developers: Focused on implementing the functionalities.

• Workflow Management: We conducted regular meetings both online and onsite to track progress and address issues.

• Conflict Resolution: Communication was a major challenge. Some members did not voice their difficulties, which hindered progress. We resolved this by encouraging open communication and requiring regular updates at the end of each sprint.

• Challenges Encountered and Solutions: The lack of communication and coordination was a significant challenge. We overcame this by implementing stricter communication protocols and ensuring all team members presented their progress regularly.

Scrum Artifacts

product backlog and sprint backlog

Test Case Document

Proposed Algorithms

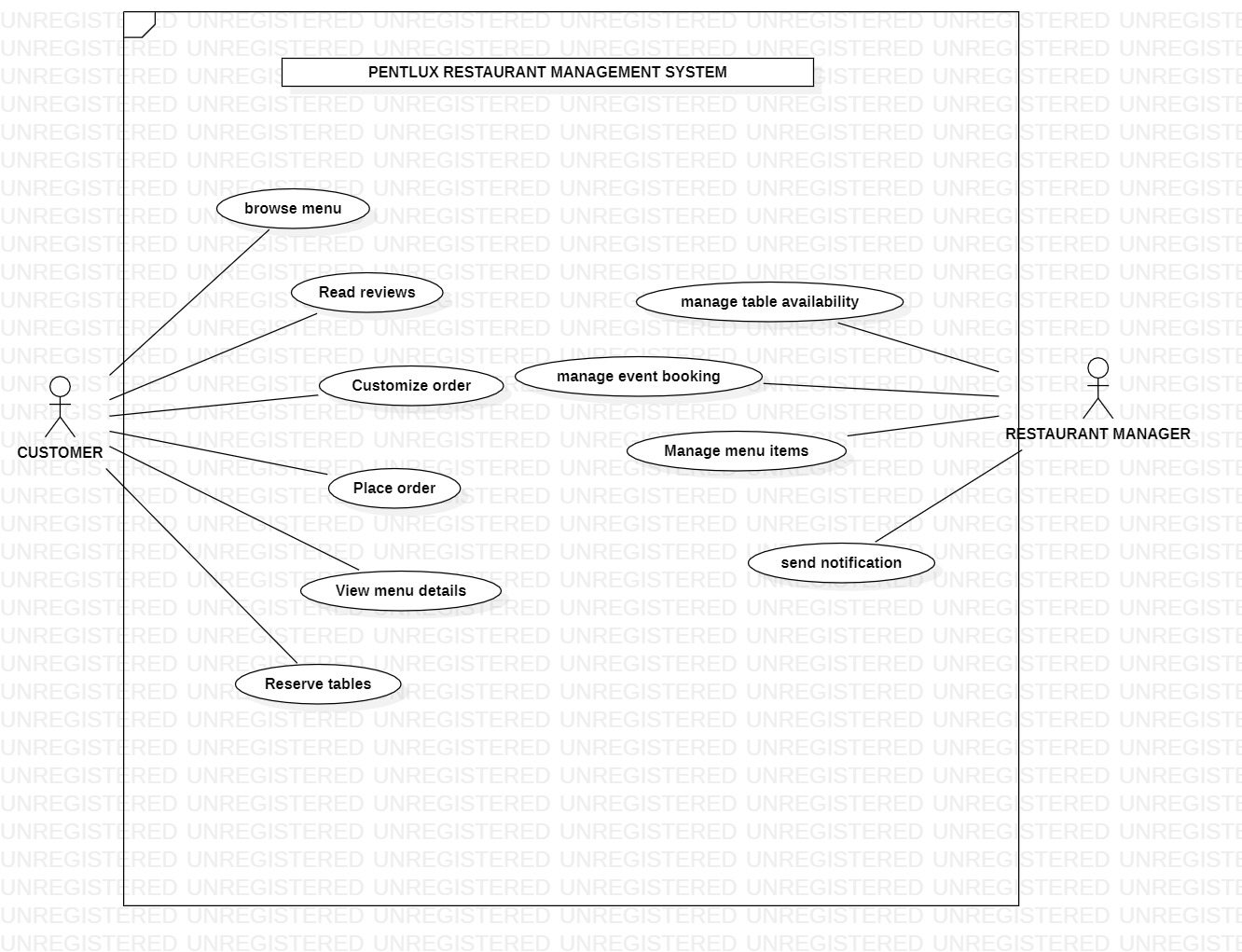
Materials and Technologies Used

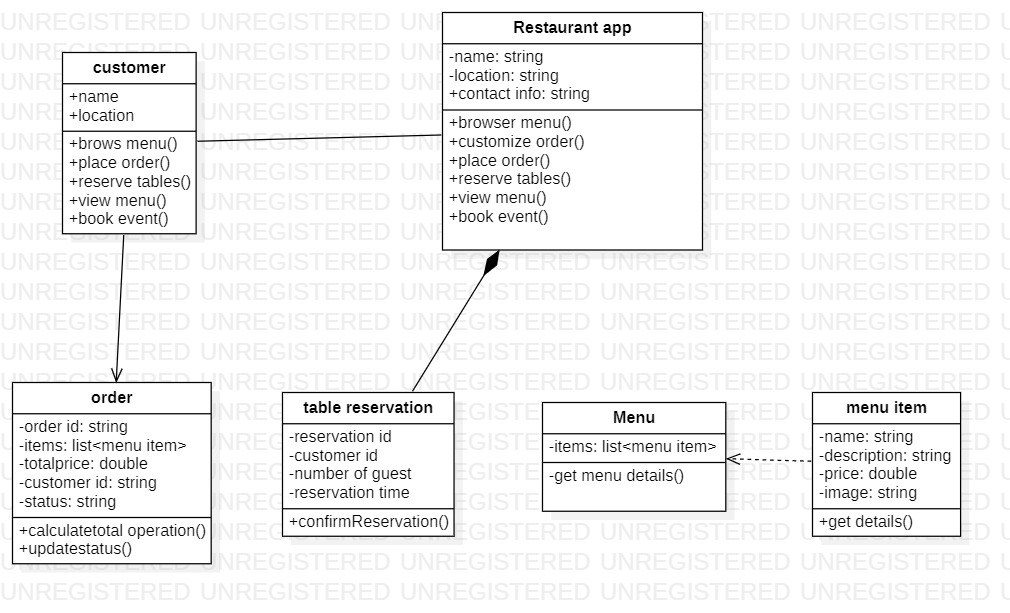
• Backend: Java Spring Boot

• Database: MySQL

• Frontend: HTML, CSS, JavaScript

CLASS DIAGRAM

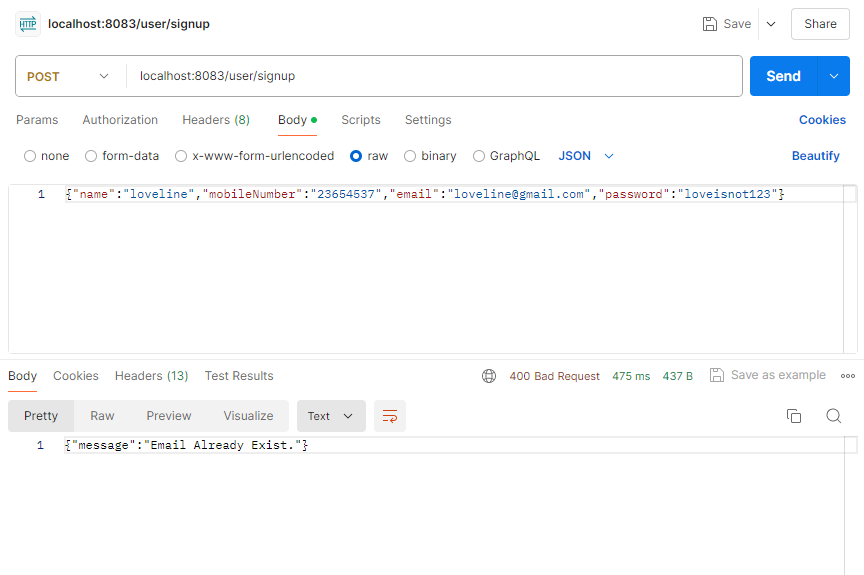


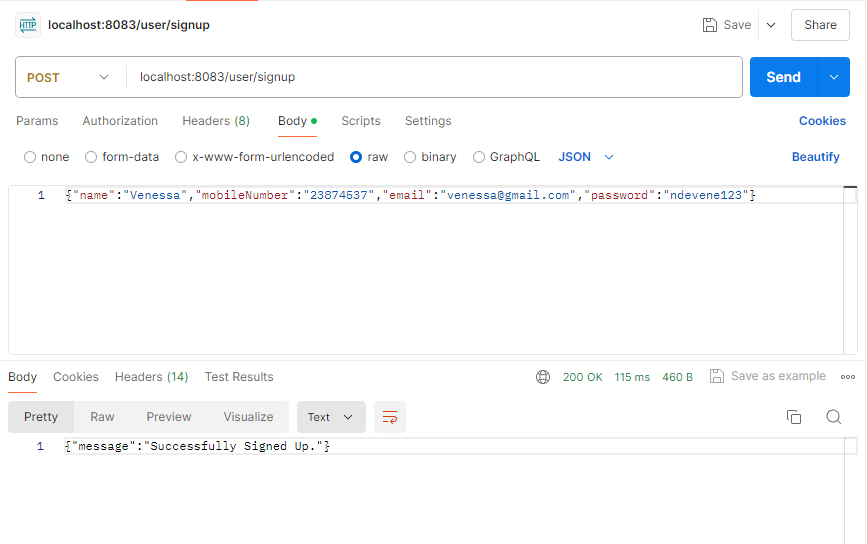


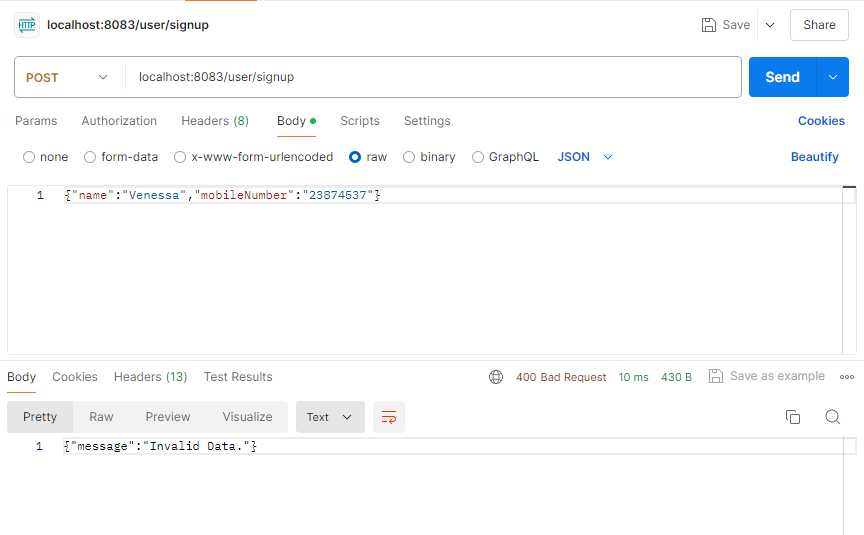
CLASS DIAGRAM

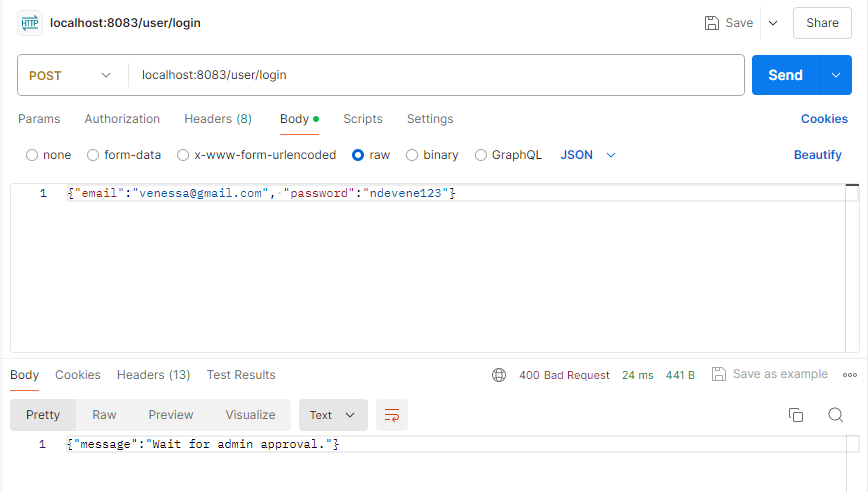
**API Requests and response**

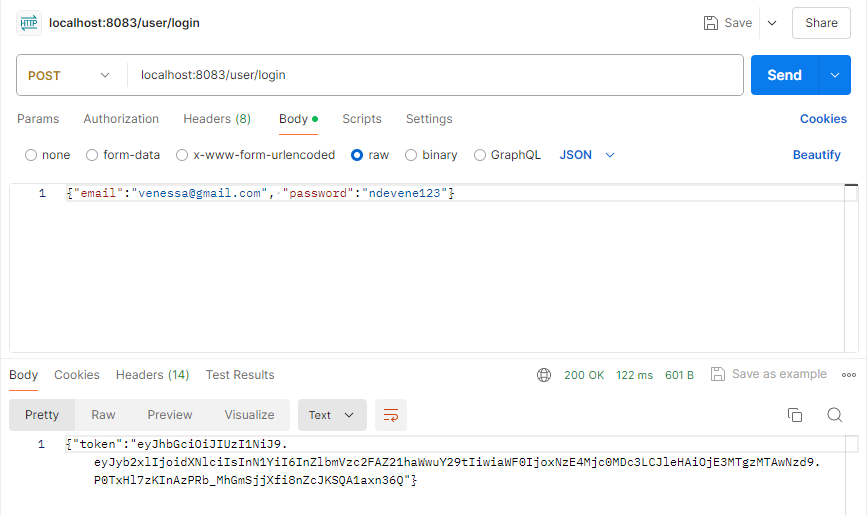
1. Sign up API





1. Login API







**CHAPTER 4: recommendation and conclusion**

Nevertheless there where some difficulties encountered which zas finding an up to date video which explains the usage of the software new versions and setting up the standalone component .

For our achievements we where successful in creating most of the pages needed in conclusion it was quite challenging work but we gave our best.