**Coursework 1**

Ivern Ng, Soh Qi Han, Sriharan s/o ganeshan

Kaplan Higher Education Academy, University of Portsmouth

PT UOP BScSE/DSA/CS 14 SETP: Software Engineering Theory and Practice

Mdm Marlar

Jul 8, 2024

# Problem Specification

## Requirements Elicitation Process

To gather requirements for our restaurant website project, we employed the following methods:

* + Brainstorming sessions with the team to identify essential features
  + Analysis of popular restaurant websites for common functionalities
  + Brief surveys with potential users (friends and family) to understand expectations

## User Requirements Overview

Based on our research, we identified the following key user requirements:

* View an up-to-date menu with prices and descriptions
* Place orders online for delivery or takeout
* Access restaurant information (hours, location, contact details)
* Simple and intuitive navigation
* Mobile-friendly interface

## System Requirements:

### Functional Requirements

* Display dynamic menu with items, prices, and descriptions
* Allow users to add or remove items from shopping cart
* Process online orders and send confirmations
* Provide restaurant information page
* Implement basic admin interface for menu management

## Functional Requirements

### Performance

Page load times under 3 seconds.

### Usability

Intuitive navigation, mobile-responsive design

### Security

Basic protection for user data

### Reliability

Consistent functionality across major browsers

# Design

## System Architecture

Our restaurant website follows a 3-tier architecture:

* Presentation Layer: HTML, CSS, JavaScript
* Application Layer: Node.js with Express.js
* Data Layer: MySQL database

## Figure 1

## 

# System Architecture Diagram

### Events and Promotions.

* Showcase event details and special promotions
* Provide a system for booking private events

### Customer Feedback.

* Implement a system for displaying customer ratings and reviews
* Include a contact form for customer inquiries

### Content Management.

* Allow easy updates to menu items, prices, and promotional content

### Non-Functional Requirements

### Performance.

* Ensure quick loading times across all devices
* Handle multiple concurrent users without significant slowdown

### Usability.

* Implement an intuitive and user-friendly design
* Ensure accessibility for users with disabilities

### Security.

* Implement secure data transmission for online payments
* Protect user data and comply with relevant data protection regulations

### Scalability.

* Design the system to handle increased traffic during peak times
* Allow for easy addition of new features in the future

### Reliability.

* Ensure high uptime and minimal system downtime
* Implement regular backups of critical data

# Design

## System Architecture

Our restaurant website will follow a 3-tier architecture.

### Presentation Layer

The user interface, implemented using HTML, CSS, and JavaScript.

### Application Layer

Server-side logic, handling requests and processing data implemented using Node.js with Express.js.

### Data Layer

Database for storing menu items, user information, and reservations implemented using MySQL database.

## Use Case Modeling

## Figure 2

A diagram of a customer

Description automatically generated

# Use Case Diagram

### Description of 5 Use Cases

* Browse Menu
* Place Order
* Make Reservation
* Leave Review
* Book Private Event

# Implementation

## Version Control

We have set up a GitHub repository for our project, allowing for collaborative development and version tracking.

## Figure 3

A screenshot of a computer

Description automatically generated

## Screenshot of Github Repository

## Implementation Tools and Technologies

We are using the following tools and technologies for our implementation:

* Frontend: HTML5, CSS3, JavaScript
* Backend: Node.js with Express.js
* Database: MySQL
* API Testing: Postman
* Integrated Development Environment (IDE): Visual Studio Code

## Testing

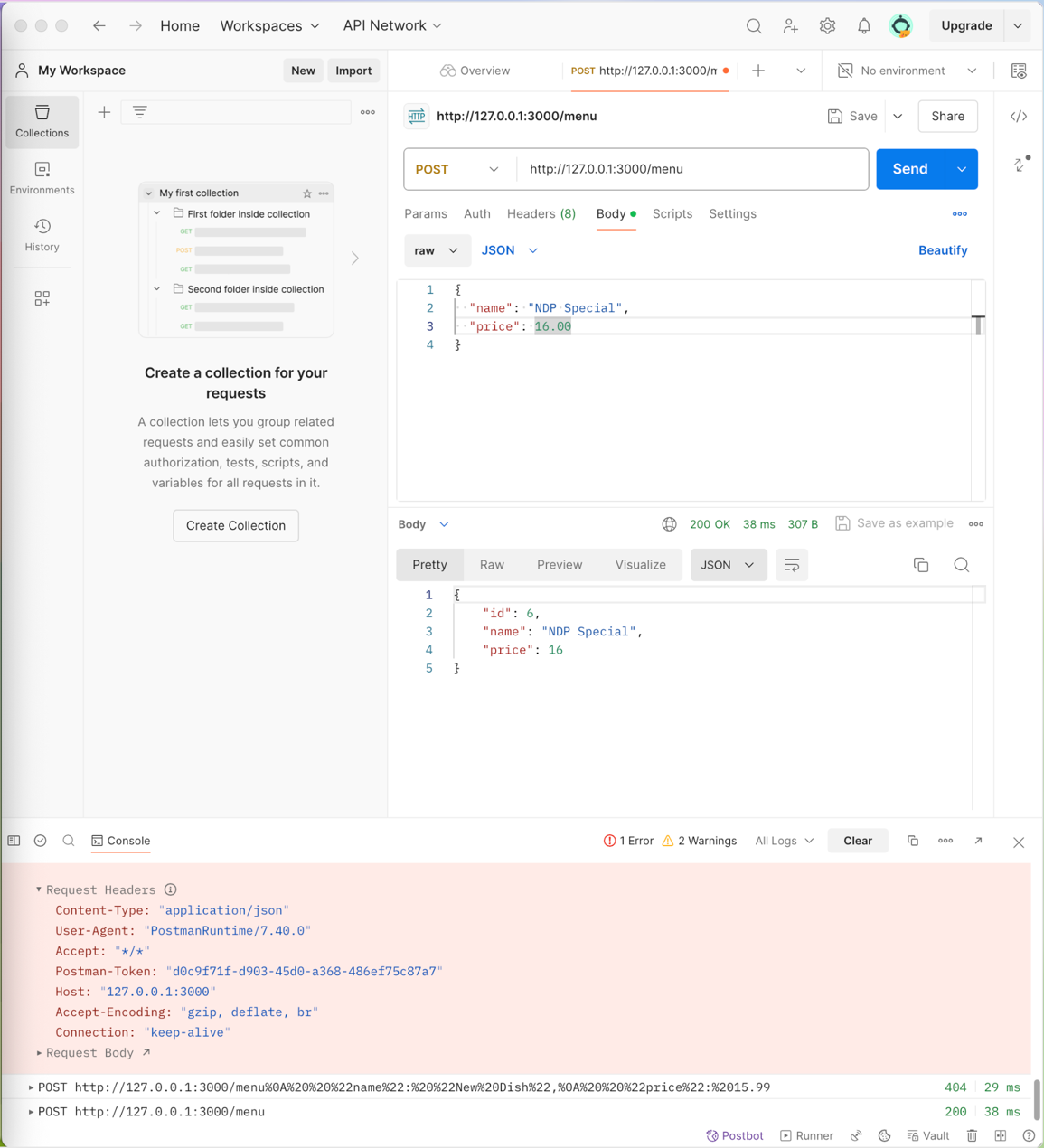
### Test Plan

Our testing strategy focused on ensuring the functionality of key features:

**API Testing with Postman.** The following tests were carried out.

* GET /api/menu endpoint for correct retrieval of menu items.
* POST /api/orders endpoint for successful order placement.

## Figure 4



# Screenshot of Postman Testing

**Manual Testing.** The following test were carried out.

* Cross-browser testing on Chrome, Firefox, and Safari
* Responsive design testing on various device sizes
* Functional testing of the ordering process

### Test Results

**API Endpoints:** All tested endpoints are responding correctly.

**Frontend functionality.** Basic features working as expected across tested browsers.

**Responsiveness.** Website adapts well to different screen sizes.

## Figure 5

A screenshot of a menu

Description automatically generated

# Screenshot of Webpage

# Critical Analysis

## Leadership

Our team adopted a collaborative leadership approach. Task are assigned based on individual strengths. We held regular Team meetings to discuss progress and challenges. Encouraged open communication and idea sharing.

## Progress Monitoring

* To track our progress effectively, we have set weekly goals and deadlines for feature implementation.
* Used Jira manage tasks and track completion.
* Conducted code reviews before merging new features.

## Conflict resolution

* While our team worked well together overall, we faced some challenges:
* Differences in coding styles were resolved by establishing team coding standards
* Conflicting ideas about features were addressed through team discussions and prioritization exercises
* Technical disagreements were resolved by researching best practices and sometimes seeking advice from our instructor.

# Conclusion

This report outlines our progress on the restaurant website project, covering the key areas of problem specification, design, implementation, testing, and team dynamics. While our implementation is still in progress, we have established a solid foundation and are continuously working to enhance and refine our website.