FeistaFlavours.com

(Restaurant Website)

Software Requirements Specification

10.04.2024

Shinkhal Sinha
12213952

Prepared for
Continuous Assessment 3
Spring 2024

Revision History

Date	Description	Author	Comments
10.04.2024	Version 1.0.0	Shinkhal Sinha	First phase of website

Table of Contents

REVISION HISTORY ..... **I**

CLIENT APPROVAL ..... **II**

1. INTRODUCTION **1**

1.1 PURPOSE 1
1.2 SCOPE 1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS 1
1.4 REFERENCES 1
1.5 OVERVIEWHTTPS://GITHUB.COM/SHINKHAL/RESTURANT_FRONTEND 1

2. GENERAL DESCRIPTION **2**

2.1 PRODUCT PERSPECTIVE 2
2.2 PRODUCT FUNCTIONS 2
2.3 USER CHARACTERISTICS 2
2.4 GENERAL CONSTRAINTS 2
2.5 ASSUMPTIONS AND DEPENDENCIES 2

3. SPECIFIC REQUIREMENTS **3**

3.1 EXTERNAL INTERFACE REQUIREMENTS 3
3.1.1 User Interfaces *3*
3.1.2 Hardware Interfaces *3*
3.1.3 Software Interfaces *3*
3.1.4 Communications Interfaces *3*
3.2 FUNCTIONAL REQUIREMENTS 3
3.2.1 <Functional Requirement or Feature #1> *3*
3.2.2 <Functional Requirement or Feature #2> *3*
3.3 NON-FUNCTIONAL REQUIREMENTS 4
3.3.1 Performance *4*
33.2 Reliability *4*
3.3.3 Availability *4*
3.3.4 Security *4*
3.3.5 Maintainability *4*
3.3.6 Portability *4*

4. ANALYSIS MODELS **5**

4.1 DATA FLOW DIAGRAMS (DFD) ..... **ERROR! BOOKMARK NOT DEFINED.**

5. GITHUB LINK…… **6**

1. Introduction

1.1 Purpose

The purpose of this SRS is to define the requirements for the development and implementation of the Fiesta Flavours restaurant website. This document is intended for software engineers, developers, designers, and stakeholders involved in the project.

1.2 Scope

1.Software Product: Fiesta Flavors - a restaurant website.

Link - https://feista-flavours.netlify.app/

2.Description:

- The website will allow users to interact with the restaurant's menu, create accounts, and reserve tables through intuitive and responsive frontend interfaces.
- It will feature user authentication, menu display, and reservation management.
- The scope includes implementing a secure login/signup process, menu presentation, and reservation system.

1.3 Definitions, Acronyms, and Abbreviations

- **MERN**: MongoDB, Express.js, React, Node.js
- **JWT**: JSON Web Token
- **API**: Application Programming Interface
- **CRUD**: Create, Read, Update, Delete

1.4 References

- [IEEE Guide to SRS] (https://ieeexplore.ieee.org/document/44566)
- [MongoDB Documentation] (https://docs.mongodb.com/)
- [React Documentation] (https://reactjs.org/docs/getting-started.html)

1.5 Overview

- This SRS document contains detailed requirements for the Fiesta Flavours restaurant website. It is organized into sections that describe the general description, specific requirements, interface requirements, non-functional requirements, design constraints, and analysis models.

2.General Description

2.1. Product Perspective

- Fiesta Flavours is a standalone web application developed using the MERN stack (MongoDB, Express.js, React, Node.js). It interacts with users through web interfacesand communicates with a MongoDB database for data storage.

2.2. Product Functions

The website will perform following functions :

- User authentication (signup, login)
- Menu display and management
- Table reservation system
- Present the restaurant menu with categorized items and detailed descriptions.

2.3. User Characteristics

- The target users of Fiesta Flavours frontend include restaurant customers who seek a visually appealing and user-friendly platform to explore menu options and make reservations

2.4. General Constraints

- Technology stack: uses Vite + React for frontend development
- Development environment: Supported on modern web browsers (Chrome, Firefox, Safari).

2.5. Assumptions and Dependencies

- Users have access to compatible web browsers capable of rendering modern web applications.
- The frontend interfaces will interact with backend APIs (Express.js) for data retrieval and processing.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interface

- **Signup/Login Pages**: User-friendly forms for account creation and authentication.
- **Menu Display:** Organized presentation of restaurant menu with descriptions.
- **Reservation Form:** Secure form for booking tables.

3.1.2. Software Interfaces

- **Backend API:** Express.js endpoints for user authentication, menu retrieval, and reservation handling.

3.2. Functional Requirements

3.2.1. User Authentication

- Requirement: Users must be able to create accounts securely.
  - Inputs: User details (name, email, password).
  - Processing: Passwords must be hashed using bcrypt for security.
  - Outputs: Successful account creation message.

3.2.2. Menu Display

- Requirement: Display restaurant menu items categorically.
  - Inputs: Menu data from MongoDB database.
  - Outputs: Menu presented on the frontend.

3.3. Non-Functional Requirements

3.3.1. Performance

- Requirement: Website should load within 3 seconds on standard internet connections.
    - Measurement: Page load times measured using web performance tools.

3.3.2. Readability

- Requirement: Maintain clean and well-organized code for improved readability and maintainability.
- Guidelines: Follow best practices in coding standards and naming conventions.

3.3.3. Availability

- Requirement: Ensure high availability of the website with minimal downtime.
- Implementation: Deploy on reliable hosting services and implement redundancy measures.

3.3.4. Security

- Requirement: User passwords must be securely hashed using bcrypt.
    - Verification: Password hashing verified during user authentication.

3.3.5. Portability

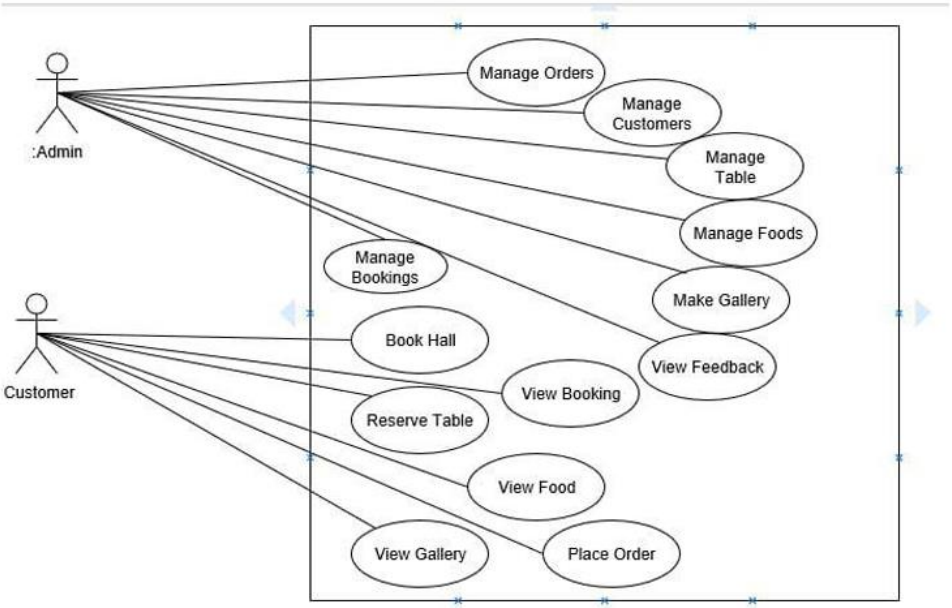
- Requirement: Ensure cross-browser and cross-device compatibility for seamless user experience.
- Testing: Conduct compatibility tests on major web browsers and mobile devices.

3.3.6. Maintainability

- Requirement: Develop with maintainability in mind, allowing for future updates and enhancements.
- Documentation: Maintain comprehensive documentation and version control for codebase.

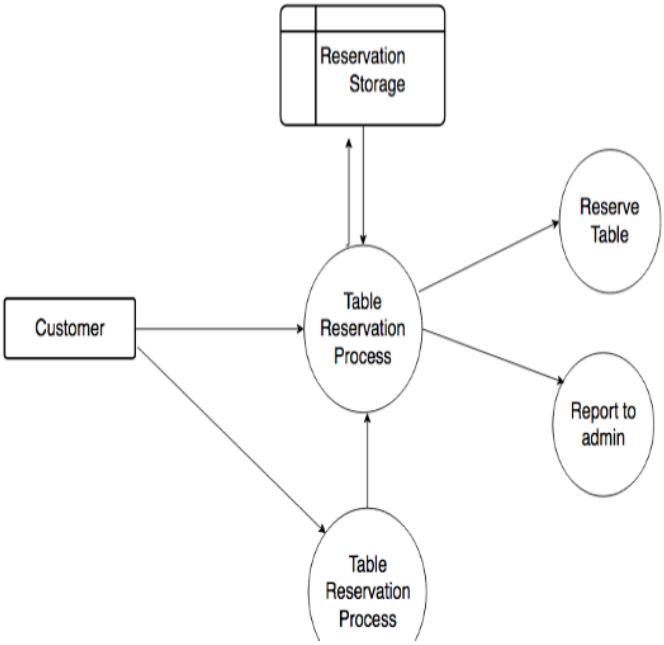
4. Analysis Models

4.1. Use Case Diagram



Restaurant Table and Online Booking

4.2.Data Flow Diagram



5. GitHub Link

https://github.com/Shinkhal/Resturant_Frontend