

SOFTWARE REQUIREMENTS SPECIFICATION

Complete Analysis of the MOS Burgers web application

Prepared by,

Theekshana Sankalpa

29-06-2024

Table of content

1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, Acronyms, and Abbreviations

2. Overall Description

- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints
- 2.6 Assumptions and Dependencies

3. Specific Requirements

3.1 External Interface Requirements

- 3.1.1 User Interfaces
- 3.1.3 Software Interfaces

3.2 Functional Requirements

- 3.2.1 Cashier Management
- 3.2.2 Customer Management
- 3.2.2 Order Management
- 3.2.2 Stock Management
- 3.2.2 Report Analyzation

3.3 Non-Functional Requirements

4. System Features

5. Other Nonfunctional Requirements

- 5.1 Performance Requirements
- 5.2 Safety Requirements
- 5.3 Security Requirements
- 5.4 Software Quality Attributes

6. Other Requirements

7. Use Case Diagram

1. Introduction

1.1 Purpose

This document outlines the software requirements for the MOS Burgers web application, which aims to digitize and streamline the operations of a medium-scale burger shop.

1.2 Scope

The web application will handle order processing, store management, bill issuing, and customer management for MOS Burgers.

1.3 Definitions, Acronyms, and Abbreviations

- SRS : Software Requirement Specifications
- UI : User Interface
- UX : User Experience

2. Overall Description

2.1 Product Perspective

The MOS Burgers web application will replace the current manual, paper-based processes for managing transactions and issuing bills.

2.2 Product Functions

- Cashier Management
- Customer Management
- Order Management
- Stock Management
- Report Analyzation

2.3 User Classes and Characteristics

- Shop Cashiers: Will use the system for day-to-day operations
- Shop Owner: Will use the system for management and report viewing

2.4 Operating Environment

- Web-based application accessible via desktop computers, tablets, and smartphones

2.5 Design and Implementation Constraints

- The application must be developed using HTML, CSS, and JavaScript
- The application must be hosted on GitHub Pages

2.6 Assumptions and Dependencies

- Users have basic computer literacy
- Stable internet connection is available at the shop

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- Responsive design for various devices
- Visually appealing layout

3.1.2 Hardware Interfaces

- Not applicable (web-based application)

3.1.3 Software Interfaces

- Web browsers (Chrome, Firefox, Safari, Edge)

3.1.4 Communications Interfaces

- HTTP/HTTPS for web communication

3.2 Functional Requirements

3.2.1 Cashier Management

- Ability to process orders
- Authentication and login
- Generating bills
- Process payments
- Update orders

3.2.2 Customer Management

- Add customers during order placement or separately
- Update, delete, and edit customer details
- View orders under each customer name

3.2.3 Order Management

- Select food items by name, code, or category
- Display item details (price, expiration warning, code, discount)
- Show selected items in cart with prices and quantities
- Apply discount to the whole order
- View, edit, delete, and update previous orders
- Generate receipt for completed orders (PDF format)

3.2.4 Stock Management

- View all food items categorically
- Update and delete food items
- Add new items to the system
- Notify users about expired food items
- Remove expired food items

3.2.5 Report Analyzation

- Generate monthly sales report
- View customers with highest order rates within the month
- Generate annual sales report
- Generate food items count report

3.3 Non-Functional Requirements

- Performance - The webpage must work in a fast manner to reduce the loading time
- Usability - The system should be easy to understand and use without needing much help
- Reliability - The website should work smoothly even when many people use it at the same time
- Security - Customer data should be protected
- Maintainability - Code should be well-organized and documented

4. System Features

(Detailed descriptions of each major feature listed in section 3.2)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system should handle 50-60 customers per day efficiently

5.2 Safety Requirements

- Not applicable

5.3 Security Requirements

- Secure storage of customer information

5.4 Software Quality Attributes

- Reliability, maintainability, usability

6. Other Requirements

- The system should be developed following best practices in web development
- Regular commits should be made to the GitHub repository during development

This SRS document provides a comprehensive overview of the requirements for the MOS Burgers web application. It can be used as a guide throughout the development process to ensure all specified features and functionalities are implemented correctly.
