# CEBU INSTITUTE OF TECHNOLOGY UNIVERSITY

COLLEGE OF COMPUTER STUDIES

# Software Requirements Specifications

*for*

HKOTISK

Developers

Lanz Roy L. Sumalpong Ben David C. Cadayona Mary Apple Ramos

# Change History

# Table of Contents

# 1. Introduction

## 1.1. Purpose

- *Describe the purpose of the SRS;*
- *Specify the intended audience for the SRS.*

## 1.2. Scope

- *Identify the software product(s) to be produced by name (e.g., Host DBMS, Report Generator, etc.);*
- *Explain what the software product(s) will, and, if necessary, will not do;*
- *Describe the application of the software being specified, including relevant benefits, objectives, and goals;*
- *Be consistent with similar statements in higher-level specifications (e.g., the system requirements specification), if they exist.*

## 1.3. Definitions, Acronyms and Abbreviations

- *provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS*

## 1.4. References

- *Provide a complete list of all documents referenced elsewhere in the SRS;*
- *Identify each document by title, report number (if applicable), date, and publishing organization;*
- *Specify the sources from which the references can be obtained.*

# 2. Overall Description

## *2.1. Product perspective*

*The Hkotisk API is a **modular order management system** designed to streamline food ordering processes for a campus canteen environment. While the system is primarily self-contained, it interfaces with multiple components to ensure efficient order processing and inventory management.*

***System Modules and Transactions:***

***Module 1: Order Processing***

- *Transaction 1.1: Order Submission*
- *Transaction 1.2: Order Prioritization*
- *Transaction 1.3: Real-time Order Status Tracking*
- *Transaction 1.4: Product Filtering*

***Module 2: Product Management***

- *Transaction 2.1: Stock Level Monitoring*
- *Transaction 2.2: Low Stock Alerting*
- *Transaction 2.3: Product Update*
- *Transaction 2.4: Product Filtering*
- *Transaction 2.5: Product Add*

***Module 3: Staff Management***

- *Transaction 3.1: User Authentication*

## *2.2. User characteristics*

### *User Types and Roles:*

***User:***

- *Can place orders*
- *View menu items*
- *Track order status*

***Canteen Staff:***

- *Process orders*
- *Manage inventory*
- *Update menu availability*

***System Administrators:***

- *Configure system settings*
- *Manage user roles*

## *2.4. Constraints*

***Regulatory and Operational Constraints:***

- *Compliance with campus food service regulations*
- *Real-time system responsiveness during peak hours*

***Hardware Limitations:***

- *Optimized for mobile and web interfaces*
- *Compatible with standard point-of-sale systems*
- *Low bandwidth network performance considerations*

***Security Considerations:***

- *Role-based access control*
- *Audit logging for all critical transactions*

## *2.5. Assumptions and dependencies*

***System Dependencies:***

- *Availability of stable internet connectivity*
- *Compatibility with modern web browsers*
- *Functioning campus network infrastructure*
- *Regular software updates and maintenance*

***Operational Assumptions:***

- *Consistent menu item availability*
- *Staff training on system usage*
- *Predictable peak hour patterns*
- *Student familiarity with digital ordering platforms*

***External Factors:***

- *Potential changes in campus dining policies*
- *Fluctuations in food supply chains*
- *Technology infrastructure upgrades*
- *Potential expansion of menu offerings*

# 3. Specific Requirements

## 3.1. External interface requirements

### 3.1.1. Hardware interfaces

*This should specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics (number of ports, instruction sets, etc.). It also covers such matters as what devices are to be supported, how they are to be supported, and protocols. For example, terminal support may specify full-screen support as opposed to line-by-line support.*

- *The system must support web browsers on desktops/laptops and mobile applications on Android devices.*
- *Optimized for standard mobile and web-based order management interfaces.*
-

### 3.1.2. Software interfaces

- ***Backend: Java Spring Boot.***
- ***Web Frontend: ReactJS.***
- ***Mobile Application: Android Kotlin.***
- ***Database: MySQL(DOCKER)***

### 3.1.3. Communications interfaces

- **WebSockets** for real-time updates.
- **RESTful API** endpoints for mobile and web applications.

## 3.2. Functional requirements

### Order Processing

***Web**:*

- ***Transaction 1.1: Order Prioritization***
  *Automatically categorizes and ranks incoming orders based on predefined criteria like customer tier, order value, and shipping urgency. Ensures critical orders are processed first for optimal customer satisfaction.*
- ***Transaction 1.2: Real-time Order Status Tracking***
  *Provides customers with live updates on their order's journey from processing to shipping. Integrates backend systems to display accurate, up-to-the-minute order progression.*

- ***Transaction 1.3: Product Filtering***
  *Enables customers to refine product searches using multiple parameters like price, category, brand, and availability. Enhances user experience by quickly presenting most relevant product selections.*

**Mobile:**

- ***Transaction 1.1: Order Submission***
  Allows customers to complete purchase transactions directly through mobile interface. Supports multiple payment methods and secure transaction processing.
- ***Transaction 1.2: Order Prioritization***
  Mirrors web functionality of categorizing orders based on strategic importance and customer preferences.
- ***Transaction 1.3: Real-time Order Status Tracking***
  Provides mobile-optimized order tracking with push notifications and detailed status updates.
- ***Transaction 1.4: Product Filtering***
  Mobile-specific product search and filtering mechanism tailored for smaller screen interfaces.

## *Product Management*

**Web:**

- ***Transaction 2.1: Stock Level Monitoring***
  *Real-time tracking of inventory across all product categories. Provides comprehensive overview of current stock quantities.*
- ***Transaction 2.2: Low Stock Alerting***
  *Automated system generating alerts when product inventory falls below predefined thresholds. Triggers restocking processes.*
- ***Transaction 2.3: Product Update***
  *Allows administrators to modify product details, pricing, descriptions, and images instantly.*
- ***Transaction 2.4: Product Filtering***
  *Advanced search and categorization tools for internal product management.*
- ***Transaction 2.5: Product Add***
  *Streamlined interface for adding new products to inventory management system.*

**Mobile:**

- ***Transaction 2.1: Cart Management***
  *Mobile-friendly shopping cart functionality with add/remove/modify capabilities.*
- ***Transaction 2.2: Product Filtering***

*Compact, touch-optimized product search and filtering interface.*

### *Staff Management*

***Web:***

- ***Transaction 3.1: User Authentication***
  Secure login system with role-based access control for staff members. Supports multi-factor authentication and permission levels.

### *Push Notification*

***Mobile:***

- ***Transaction 4.1: Order Status Updates***
  Automated mobile notifications informing customers about order progression stages.
- ***Transaction 4.2: Queue management alerts***
  Provides real-time notifications about system queues, wait times, and processing status.

. . .

## *3.4 Non-functional requirements*

### *Performance*

*Details*

### *Security*

*Details*

### *Reliability*

*Details*