Point of Sale System

Software Requirements Specification

Version 2.0

<Date>

<Your Name>

Lead Software Engineer

Prepared for

CIT/CSE 480 Senior Project

Instructor: Professor Patel

Fall 2015

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 09/17/2015 | Version 1.0 | Mel Chi | Getting Started |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | <Your Name> | Lead Software Eng. |  |
|  | A. David McKinnon | Instructor, CptS 322 |  |
|  |  |  |  |

**Table of Contents**

Revision History 2

Document Approval 2

1. Introduction 5

1.1. Purpose 5

1.1.1. Mel 5

1.2. Scope 5

1.3. Definitions, Acronyms, and Abbreviations 5

1.4. References 5

1.5. Overview 5

2. General Description 5

2.1. Product Perspective 6

2.2. Product Functions 6

2.3. User Characteristics 6

2.4. General Constraints 6

2.5. Assumptions and Dependencies 6

3. Specific Requirements 6

3.1. External Interface Requirements 6

3.1.1. User Interfaces 6

3.1.2. Hardware Interfaces 6

3.1.3. Software Interfaces 6

3.1.4. Communications Interfaces 7

3.2. Functional Requirements 7

3.2.1. Consumer Interface 7

3.2.2. Back of House Interface 7

3.2.3. Front of House Interface 7

3.2.4. Admin/ Interface 7

3.3. Use Cases 7

3.3.1. Consumer Interface 7

3.3.2. Back of House Interface 7

3.3.3. Front of House Interface 7

3.3.4. Admin/ Interface 7

3.4. Classes/Objects 7

3.4.1. MySQLUtility 7

3.5. Non-Functional Requirements 7

3.5.1. Performance 7

3.5.2. Reliability 7

3.5.3. Availability 7

3.5.4. Security 7

3.5.5. Maintainability 7

3.5.6. Portability 7

3.6. Inverse Requirements 7

3.7. Design Constraints 7

3.8. Logical Database Requirements 7

4. Analysis Models 8

4.1. Sequence Diagrams 8

4.2. DataFlow Diagrams (DFD) 8

4.3. State-Transition Diagrams (STD) 8

5. Change Management Process 8

A. Appendices 8

6. SAMPLE 8

6.1. SAMPLE 8

6.1.1. SAMPLE 8

6.2. SAMPLE 8

# Introduction

This Software Requirement Specification (SRS) is of the CSE 480 Point of Service (POS) Software. The POS Software is a complete interface for a restaurant and provides the restaurant owner to provide added services to its patrons.

## Purpose

The SRS is intended for developers to understand all features of the software, and is intended to be used to continuously update the system as needed.

### Mel

###### Mel

## Scope

The POS Software will include an interface for the hosts and waiters, one for the kitchen, one for the owner and manager, and one for a consumer.

The interface for host and waiters should be able to place orders and process payments, while the kitchen interface should allow the kitchen staff to see the orders and let the front of house know when any part of an order is ready. The interface for the owner is all encompassing and will provide the options to edit portions. Lastly the consumer interface will give the consumer information and allow the option to order online.

The back-end is a MySQL database system, while the front-end and back-end both utilize libraries and APIs available to the public.

## Definitions, Acronyms, and Abbreviations

*This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.*

## References

Java ArrayList Class Document, *Java Platform Standard Ed. 7*

<http://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html>

Java Collections Class Document, *Java Platform Standard Ed. 7*

<http://docs.oracle.com/javase/7/docs/api/java/util/Collections.html>

## Overview

The rest of this SRS contains more specifics as to what this product is, how it is organized and structured. This document is broken up into sections and subsections and these sections are listed in the table of contents.

# General Description

*This section of the SRS should describe the general factors that affect 'the product and its requirements.  It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.*

## Product Perspective

## Product Functions

## User Characteristics

## General Constraints

## Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

# Specific Requirements

This will be the largest and most important section of the SRS.  The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation, and testing.

Each requirement in this section should be:

·       Correct

·       Traceable (both forward and backward to prior/future artifacts)

·       Unambiguous

·       Verifiable (i.e., testable)

·       Prioritized (with respect to importance and/or stability)

·       Complete

·       Consistent

·       Uniquely identifiable (usually via numbering like 3.4.5.6)

Attention should be paid to the carefuly organize the requirements presented in this section so that they may easily accessed and understood.  Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

## External Interface Requirements

### User Interfaces

The Point of Sale system includes 4 interfaces for the consumer, admin/owner, kitchen staff, and the front of the house. Each one of these interfaces have different roles and limitations.

### Hardware Interfaces

The goal is to create an android app for each. However, we may consider the admin/owner to have a standalone program to help with administration details.

### Software Interfaces

The software will just need to be compatible with Android, but for the consumer, the goal is to allow any system that can connect to a web browser to access.

### Communications Interfaces

## Functional Requirements

### Consumer Interface

###### Ordering Food

###### Introduction

Using the consumer interface, the consumer has the option to order online. The ordering online also will include some features as notifying the person the food is ready.

###### Inputs

Consumer selects the menu item that he/she would like to buy and order. Connect with PayPal for cash handling.

###### Processing

Each order will be a new invoice number.

###### Outputs

Email with the order will be sent. It also goes to the user’s history.

###### Error Handling

If the order fails, it will try at least one more time to connect.

###### View History

###### Introduction

Using the consumer interface, the consumer can see all previous orders.

###### Inputs

User will just select the dates to view between, default will be 3 months.

###### Processing

The application will query the sql database between the 2 dates given.

###### Outputs

The application will show previous orders and the menu choices it picked.

Option to email as well.

###### Error Handling

Timeout if can not find and tell the user to see later.

### Back of House Interface

###### Get Orders

###### Introduction

The Back of House Interface will allow the kitchen staff to see the orders that have been placed. It is up to them to decide how to do it.

###### Inputs

There are no input as it will be the default screen for the kitchen staff. They will get it as orders come in.

###### Processing

###### Outputs

They will swipe up to clear.

###### Error Handling

###### Get Order History

###### Introduction

The Back of House Interface will allow the kitchen staff to see the orders that have been completed for the day

###### Inputs

There will be a button to press to see the orders

###### Processing

###### Outputs

They will see all of the orders for the day.

###### Error Handling

### Front of House Interface

###### Mel

### Admin/ Interface

###### Mel

## Use Cases

### Consumer Interface

###### Mel

### Back of House Interface

###### Mel

### Front of House Interface

###### Mel

### Admin/ Interface

###### Mel

## Classes/Objects

### MySQLUtilities

###### Attributes

###### Functions

## Non-Functional Requirements

### Performance

The Point of Service system will be available at all times during resturant operates. During the hours of 10 AM to 1 AM in the morning.

### Reliability

There will be no downtime during business hours.

### Availability

The whole system shall be available.

### Security

### Maintainability

Updates and Backups will happen during the non-service hours every night.

### Portability

## Inverse Requirements

## Design Constraints

## Logical Database Requirements

# Analysis Models

## Sequence Diagrams

## DataFlow Diagrams (DFD)

## State-Transition Diagrams (STD)

# Change Management Process

# Appendices

DO NOT TOUCH UNDER HERE

# SAMPLE

## SAMPLE

### SAMPLE

###### Mel

###### Mel

###### Mel

## SAMPLE