Software Requirements Specification

for

Restaurant Ordering System

**Version 1.0 approved**

**Prepared by Baca, Damien Paige**

**Duchene, London Isaiah**

**Hudson, Daniel James**

**Lavelle, Conor**

**Pearson, Tanner Lee**

**Torres, Alexander Nicholas**

**CS365 Group 7**

**16 February 2016**

**Table of Contents**

**Table of Contents**

**Revision History**

**1.** **Introduction**

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Product Scope

1.5 References

**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 User Documentation

2.7 Assumptions and Dependencies

**3.** **External Interface Requirements**

3.1 User Interfaces

3.2 Hardware Interfaces

3.3 Software Interfaces

3.4 Communications Interfaces

**4.** **System Features**

4.1 Age Check For Alcoholic Drinks

4.2 Pay Bill At Table

4.3 Privilege System

4.4 Special Requests

**5.** **Other Nonfunctional Requirements**

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

5.5 Business Rules

**6.** **Other Requirements**

**Appendix A: Glossary**

**Appendix B: Analysis Models**

**Appendix C: To Be Determined List**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Conor Lavelle | 2/15/16 | Wrote initial rough draft of section 5 | 0.1 |
| Tanner Pearson | 2/15/16 | Wrote first pass of section 3 review if needed | 0.1 |
| Damien Baca | 2/15/26 | Started System Feature #1 | 0.1 |
| London Duchene | 2/15/16 | Wrote section 2 | 0.1 |
| Daniel Hudson | 2/15/16 | Started Section 1 | 0.1 |
| Damien Baca | 2/16/16 | System Features Almost Complete | 0.2 |
| London Duchene | 2/16/16 | Finished section 2 | 0.2 |
| Alexander Torres | 2/16/16 | Added legal requirement. Added 2 analysis models | 0.2 |
| Daniel Hudson | 2/16/16 | Completed Section 1 | 0.2 |
| Daniel Hudson | 2/16/16 | Proofread and Standardized Document | 1.0 |

# Introduction

## Purpose

This specification covers the requirements for a Restaurant Ordering System. This system is used by waiters, chefs, and managers, each with a different level of access. The system must securely connect to a card swipe service and must allow editable menus.

## Document Conventions

In this document, the priority of any requirement is inherited from its high level requirements.

## Intended Audience and Reading Suggestions

This document is intended to be read by developers, project managers, marketing staff, testers, and documentation writers. The document starts with an overview of the product. This is followed by requirements for interfaces with external systems. The document ends with functional and nonfunctional requirements.

For any nontechnical readers, it is recommended to reference the appendix to understand the terminology used in this document. After this, all readers should read the overview. Developers should then read the functional requirements followed by the nonfunctional requirements. They should also skim the external interfaces for reference. Project managers and marketing staff should continue with the external interfaces with specific attention to the extra systems required for the product to function. Project managers should then read the functional requirements to gauge the progress of the developers. Testers should focus on the functional requirements, ensuring that all of them are thoroughly covered by tests. They should have a basic knowledge of the nonfunctional requirements, as they can be important for the tests they create. Documentation writers should take note of the appendix and overview. They should next focus on the functional requirements, which will be the focus of a bulk of the documentation they write. Documentation writers should also take note of the nonfunctional requirements, which are not required for most documentation, but can inform a curious user of the product.

## Product Scope

This project is a restaurant ordering system to be used by waiters, managers, and chefs. The goal of the project is to consolidate all aspects of ordering into one system. This simplifies the jobs of the employees of the restaurant. It also enforces alcohol purchase restrictions and allows the menu to be adjusted based on the current menu offerings. This system can simplify communication between waiters and chefs, reducing the number of clerical and communication errors.

# Overall Description

## Product Perspective

This system is meant to help waiters give cooks food orders and any other relevant information. It will be used in mid sized restaurants. This system is connected to the waiters taking the orders and inputting them into the software and customers who give the order and pay using the software. The software will be on an iPad and be using Google Pay.

## Product Functions

* The ability to input orders and send those orders to a different device.
* The ability to change the items that are on the menu.
* The ability to securely pay for orders.

## User Classes and Characteristics

The different users that will be using this device are waiters, cooks, managers, and customers.

The most important of them being the waiters because they will be constantly using the software. Waiters will be lower level employees that have no security authorization, a high school level education, and little to no technical expertise or experience. They will be using the software to input orders for the customers. Cooks will also be constantly be using the software, but primarily to read the order then cook it. They will be a higher level employee than the waiter but they will still have no security authorization. They have anywhere from a high school to a college level education, no technical expertise, but they have experience. Managers will sometimes be using the software, primarily for adding or removing items to the menu or to remove items from an order. They will have some technical expertise, a college or high school education, and a lot of experience. And finally the customer, who will be using the software to order and pay. The customer has no security level.

## Operating Environment

The software will be operating on an iPad, any iPad from the last few years. The software should be able to work next to any other application. The software can run on any iPad OS from the last few years also.The iPad will be held by the waiter or placed somewhere near the waiter.

## Design and Implementation Constraints

The only user who will have administrator privileges in the software is the store manager. The administrator privileges will include add/removing things from the menu and removing items from an order. The software will be set up to use Google Pay, which means only Google should have the customers money information. The software will be set up to support many languages.

## User Documentation

Within the software there will be a help button that you can press and receive general information. There will also be an online user manual that goes more in depth about each function and what is possible to do with the software.

## Assumptions and Dependencies

We assume that the waiters using the software have a general understanding on how to use an Apple product. We assume that the current iPad OS hasn’t changed significantly. The product is very dependent on the Apple iPad. The software will be using all in house code so there is no external code developing dependencies.

# External Interface Requirements

## User Interfaces

The UI for the touchpad system needs to be touchscreen friendly and simple to navigate. Each screen needs an undo / back button. Before any final action (sending orders to cook / payment) there must be a confirmation screen that reviews relevant data. The orientation of the screen that the ui is designed for is portrait. The style of the GUI should keep the colors of the restaurant but does not need to keep the style of the menu, for ease of use. All of the buttons need to be clearly marked and give feedback when pressed. They cannot be too small or close together to avoid mispresses.

## Hardware Interfaces

The primary part of the software would be an iOS app running on iPads but the app would connect to a central server to store data so any iPad could have the data for any table. This server would be hosted locally. There would also be a printer to print out the receipts for the chefs. The iPads would print directly and not go through the server for printing.

## Software Interfaces

The software is an iOS application that interfaces with the newest version of Apple’s iOS. The software would also interface with Google Pay to handle the customer transactions. The local server would be receiving messages from all the iPads about the different orders and would need to be able to keep them all stored in an orderly fashion.

## Communications Interfaces

The iOS application need to communicate back and forth with the local server. This is done using a simple TCP protocol. It also need to be able to send print request to a receipt printer. Neither of these need to have security because they are all local and don’t contain financial information. The data need to get to the server in a short amount of time but does not need to be as fast as possible.

# System Features

## Age Check For Alcoholic Drinks

4.1.1 Description and Priority

*Upon ordering an alcoholic drink, the system shall display a prompt for a waiter to approve that the customer ID has been checked.*

*Priority - High*

*Benefit - 9*

*Cost - 2*

*Risk - 2*

4.1.2 Stimulus/Response Sequences

1. *[Waiter] - The waiter adds the alcoholic item to the order.*
2. *[Waiter] - The waiter can continue to input the remainder of the order.*
3. *[Waiter] - Presses the button to submit the order*
4. *[ System ] - Prompts the waiter to validate the customer’s ID*
5. *[Customer] - Allows the waiter to validate their ID*
6. *[ System ] - Passed: Order is processed*
7. *[ System ] - Failed: Error page is displayed and the waiter has the option to resubmit their order.*

4.1.3 Functional Requirements

REQ-1: System shall prevent alcoholic order submission if ID validation is failed or incomplete.

REQ-2: System shall automatically display an error page in the result of a failed ID validation. Either removal of the item, or ID validation will be required to proceed.

REQ-3: System must allow easy configuration of menu items that require ID validation.

## Allows Configuration of Daily Specials

4.2.1 Description and Priority

The system must allow the addition of daily specials to the menu when needed. This can exist as both special prices on pre-existing menu items on certain days, or an entirely new menu item that is available on specific days.

*Priority - Medium*

*Benefit - 7*

*Cost - 5*

*Risk - 6*

4.2.2 Stimulus/Response Sequences

The system shall allow configuration of the following:

* Day(s) the special takes place on.
* The item’s title and description
* (Optional) Menu Item Picture
* Price reduction of the menu item if applicable

The special prices shall be managed automatically.

Any special menu item will only be processed if it is being served on the given day.

4.2.3 Functional Requirements

REQ-1: The system shall support both adjustment of existing prices, as well as the addition of new menu items based on the day.

REQ-2: The system shall be able to automatically alter the prices based on the day once configurated.

REQ-3: The system shall be able to support an unlimited amount of Daily Specials.

REQ-4: Only someone registered as a manager will be allowed to modify the daily specials.

## Privilege System

4.3.1 Description and Priority

The system shall allow different logins for managers and waiters. The waiters shall have limited access to some features of the system allowing the managers to have total control over the things that they, and they alone, must manage.

*Priority - High*

*Benefit - 10*

*Cost - 7*

*Risk - 7*

4.3.2 Stimulus/Response Sequences

1. [User] Before use, the user must identify themselves with a username and password. This will be configurable only by a manager.
2. [Manager] Managers will have access to all features of the system as well as the power to add and remove waiters.
3. [Manager] Managers will also be able to further limit or extend the privileges for each individual waiter for greater configuration.
4. [Waiter] Waiters will have access to the features that the manager deems necessary for them to effectively do their job.

4.3.3 Functional Requirements

REQ-1: The system shall have the capability to assign waiters a random employee ID to serve as a username.

REQ-2: The system privileges shall only ever be changed by a manager.

REQ-3: The system will allow for easy configuration of privileges for each individual employee.

REQ-4: Waiter passwords shall be set by the waiters themselves.

## Special Requests

4.3.1 Description and Priority

The system shall allow waiters to enter special requests issued by the customer for the chef.

*Priority - Low*

*Benefit - 5*

*Cost - 2*

*Risk - 2*

4.3.2 Stimulus/Response Sequences

1. The system shall prompt the waiter with a text box in which to enter any special requests that the customer had for their order.
2. The system shall accept an empty request box and interpret it as there being no special requests by the customer.

4.3.3 Functional Requirements

REQ-1: The system shall allow the waiter to provide special instructions to the chef, from the customer regarding their order.

REQ-2: An empty box shall be interpreted as the customer having no special instructions.

REQ-3: The special instructions shall be printed on the receipt so the customer can review it.

REQ-4: The system shall also have a box indicating whether the special instruction will result in a price adjustment.

REQ-5: Any special instruction that result in a price adjustment must be automatically added to the total of the order.

# Other Nonfunctional Requirements

## Performance Requirements

The system is meant to make the ordering process easier and more streamlined, therefore it needs to perform quickly and reliably. If any system actions take an exceedingly long time to execute or if the system is frequently crashing, it will slow down the ordering process. The System’s response to waiter input on the touch pad should be nearly instantaneous. The time needed to send information to the chef is more flexible, but still should be a short amount of time. However, because the operations the system needs to perform are relatively simple, these performance requirements should be easy to meet and should not significantly constrain the system’s design.

## Safety Requirements

In a restaurant ordering system, there aren’t too many safety concerns. The only possible harms and damages that could come from this system would be if it failed to properly send important customer and order information. For example, if a customer had a certain food allergy and requested an alteration to his order.

## Security Requirements

There are no security requirements for the information transmitted between the waiters and the chefs. As mentioned in 3.4, this communication will use a simple TCP protocol. Customers will be paying using Google Pay, which uses industry standard SSL to protect financial information.

## Software Quality Attributes

As mentioned in 5.1, reliability is essential to the system fulfilling its intended purpose. Furthermore, usability is a very important part of the system. Ideally, the system would be intuitive and easy for employees to use. Finally, the menu design feature needs to be relatively flexible in order to accommodate any sort of menu.

## Business Rules

Only the manager may remove an item from a customer’s bill. A manager might want to remove an item from the customer's bill in response to the food not being properly prepared, spilled, etc. In addition, customers must be 21+ years old in order to purchase alcohol.

# Other Requirements

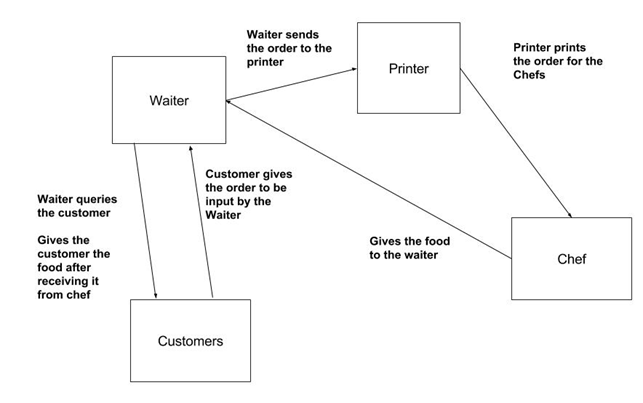
Legal Requirements: Allergy warnings. All menu items will have a special tag that will allow the manager to add Allergy warnings that will appear in red near the given food item. This allows for the waiter to quickly tell the customer what is potentially dangerous. This feature could also be used for flagging vegan options.

## Appendix A: Glossary

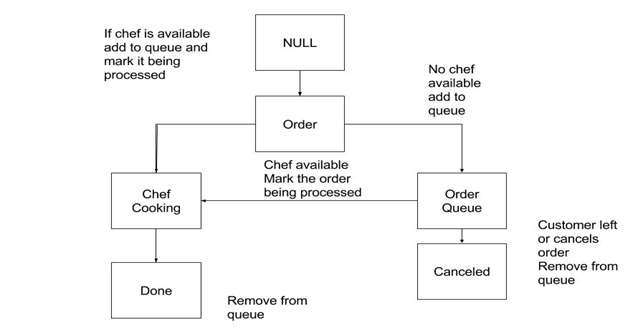
1. UI – User Interface
2. GUI – Graphic User Interface
3. App- short for application
4. TCP protocol – Transmission Control Protocol – reliable ordered and error checked delivery information stream
5. ID Identification

## Appendix B: Analysis Models

### Data Flow Diagram:



### State transition:



## Appendix C: To Be Determined List