****

**CentRes**

**Software Requirements Specification**

**Version #1**

**Team Number: 10**

**Project Manager: Katherine Landsman**

**Mentor: Susan Rizzo**

**Team Members: Carson Pribble, Chase Jamison, David Utshudiema, Mohamed Ibensilalen, Sergio Mainville**

**Revisions**

| **Version** | **Primary**  **Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
| V.1.0 | Carson, Chase, David, Mohamed, Sergio, Katherine | First draft of Software Requirements | 10/13/22 |
| V.1.1 | Carson, Chase, David, Mohamed, Sergio, Katherine | Finalized V1 of Software Requirements | 10/20/22 |

**Review History**

| **Reviewer** | **Version Reviewed** | **Date** |
| --- | --- | --- |
| Susan Rizzo | V.1.0 | 10/19/22 |
|  |  |  |

**Table of Contents (Version 1)**

1. Introduction
   1. Project Objectives
   2. Project Scope
   3. Project Overview
2. General Description
   1. Project Features / Functions
   2. User Stories
   3. Use Cases
3. Team Collaboration and Documentation Tools
4. Project Management Plan
5. Business Requirements
6. User Requirements
7. Functional Requirements
8. Non-Functional Requirements
9. Implementation (Performance) Requirements (Optional)
10. **Introduction**
    1. Project Objectives:

Allow for employees to easily interact with customers’ orders via an app and reduce miscommunication. Provide centralization of all server tables allowing servers to better manage and prioritize orders. App will help cut down on customer wait time by transmitting information directly to the kitchen.

* 1. Project Scope:
* App will not automatically rotate servers, this will be manually achieved by the hostess
* App will provide order capability
* App will provide order management including creating, deleting, adding, and removing items from an order.
* App will provide server management
* Managers will have upgraded admin privileges within the app to edit customer orders
* Menu items will be stored in a database
* App will keep track of volume of items sold each day for tracking purposes
* App will not have timesheet functionality. (Keeping track of hours worked)
* App will not keep track of employee’s weekly schedules.
* App will not have the functionality for communication between users (servers, managers, etc.)
  1. Project Overview:

Servers will have an easy-to-read display that keeps track of things such as menu items, prices, order time, and order status. They can also view their current tables and see the current bill/tab for each table, see the average order time, and display information associated with a table. Line cooks/Chefs will see the current orders placed and the time they were placed. They can notify the server when an order is ready. The application would display the menu and allow the manager the ability to add or edit items in an order. When viewing a table, the server/manager can view or distribute an itemized bill to the customer.

1. **General Description**
   1. Project Features / Functions
   2. User Stories
      1. As a server, I want to only see available tables so I can reserve it for the customer.
      2. As a server, I want to place an order in my device and send it directly to the kitchen.
      3. As a line cook, I want to receive notification for every new order and can mark it ready after preparation.
      4. As a server, I want to receive notification from the kitchen on every order that is ready so I can mark it completed after serving it to the right table.
   3. Use Case
      1. **Name:** Server

**Actor:** Server

**Basic Use Case Description:**

* Server reserves available table for the customer.
* Server sends order to the kitchen.
* Server gets notification of order completion and marks the order complete. after order is served to the customer.
  + 1. **Name:** Line Cook

**Actor:** Line Cook

**Basic Use Case Description:**

* Line cooks receive placed order with time stamps.
* Line cooks check off individual items within order.
* Line cooks mark the order ready after all items within the order are prepared.
  + 1. **Name**: Host

**Actor:** Host

Basic Use Case Description:

* Host can manage table availability.
* Host can assign tables to servers.
  + 1. **Name:** Manager

**Actor:** Manager

**Basic Use Case Description:**

* Manager can edit orders.
* Manager can alter bill.
* Manager can view overall wait times for all orders.
* Manager can view time stamps on orders.

1. **Team Collaboration and Documentation Tools**

* GitHub
  + Used as a VSC
* GitHub Projects
  + Tracking project tasks in a centralized location
* Microsoft Teams
  + Used for project communication

1. **Project Management Plan**

* Agile Principles
  + Allows for project transparency, and task optimization
* Scrum Methodology
  + Encourages higher productivity

1. **Business Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| BR1 | System must be available on both IOS and Android | M |
| BR2 | Effectively manage tables and orders | M |
| BR3 | Weekly efficiency reports are generated | M |
| BR4 | Keep long term data in a cloud or off-site server database with a data storage model allowing for frequent input of data but infrequent querying of that data (AWS) | S |
| BR5 | Rolling out special opportunities and events for customers in the rewards program | S |
| BR6 | Provide real-time business metrics to track sales, operations, and customer service performance. | M |

1. **User Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| UR1 | Host staff can manually mark tables as: a) Open for seating. b) Seated. c) Need bussing. | S |
| UR2 | Manager can edit orders, alter bill, overview of wait times, view time stamps in order | M |
| UR3 | Chef can view orders, time stamps for orders, and mark as ready | M |
| UR4 | Waiter can view orders, view tables, time stamp for orders, alter bill, view menu | M |
| UR5 | Waiter can add to order, print check, | M |
| UR6 | User can login with username and password | M |

1. **Functional Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| FR1 | Customer rewards program | C |
| FR2 | Display menu when requested | M |
| FR3 | Display orders, bill, wait times, and time stamps when requested | M |
| FR4 | Allow user to login when successfully entering in user and password | M |
| FR5 | Use 2FA for manager login to secure admin privileges (two factor authentication) | S |
| FR6 | Keep track of volume over time to create a predictive model for future use | S |
| FR7 | Require user log back in after tablet/device is idle for more than 60 seconds | S |

1. **Non-Functional Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| NFR1 | Database should update in a timely manner to keep communication between users accurate | M |
| NFR2 | Password must meet defined password criteria | M |
| NFR3 | Must not allow false input to a field | M |
| NFR4 | Users should only be able to access what they have permission to see | M |
| NFR5 | Can scale the restaurant table display volume and placement based on remodeling and rearrangement | S |
| NFR6 | Use of local database for efficiency and external/cloud-based database for long term data collection for aggregation | S |
| NFR7 | Use a third-party security/authentication system to avoid unwanted manipulation | C |
| NFR8 | Keep track of each table as an object with data attributes such as item orders per seat, bill cost, wait time | S |
| NFR9 | Implement customer rewards program with customer email | C |
| NFR10 | The system should support enough logged in users at any given time to satisfy the restaurants requirements for access frequency | M |

1. **Implementation (Performance) Requirements (Optional)**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| IR1 | The app should not go down due to an increase in users. | M |
| IR2 | The application should always load in less than 0.03 seconds. | C |