**[Project Title]**

**Software Requirements Specification**

**Version #1**

**Team: 5**

**Project Manager:** Shmeka Smith

**Mentor:** Cindy Luttrell

**Team Members:** Shmaya McLean, Ashia Pierce, Ashish Khanal, Benjamin Comfort

**Revisions**

| **Version** | **Primary**  **Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**Review History**

| **Reviewer** | **Version Reviewed** | **Date** |
| --- | --- | --- |
|  |  |  |
|  |  |  |

**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**

The main objective of the restaurant ordering system is to create a concise, reliable, and efficient system that will simplify the order process between the customers and the restaurant and reduce manual labor for the staff. This process would improve customer satisfaction and communication between both parties by decreasing response times by servers, guaranteeing the accuracy of orders made, and having an up-to-date menu. It could lead to the opportunity of increased profits by providing information on the number of supplies used, an easily accessible way to look at records of billing and orders placed, and having more knowledge on peak hours to accommodate appropriately.

General Objectives:

· Refining the accuracy of customer orders and improving communication between

customer and servers.

· Increasing customer satisfaction.

·Assists in keeping track of inventory/supply needs and use within the restaurant.

Specific Objectives:

· Decreasing the opportunity for inaccurate orders and miscommunications among

staff.

· Providing faster service to the customer.

· Reducing the queue between customers, increasing customer intake.

· Reduces the amount of restaurant paper and food wastage.

· Keeps records of orders placed and billing.

· Collects data for better time management, potentially reducing the cost of labor.

* 1. **Project Scope**

The restaurant management system software enables the restaurant to accept online orders from customers, and for servers to enter in-person orders. The point of sale (POS) is a restaurant software that takes orders. Online, the system will display a menu, so customers will know what to choose from and grants customers the ability to be descriptive with their orders. This allows customers to enter any allergies, food restrictions, personal preferences, or additional information

needed for the chefs. The system will then process the entered order, collect payment, and facilitate the delivery if needed.

In restaurant, the software will permit employees to take phone and in-person orders. This allows employees to personally take the customer’s order and to add any additional information needed, similar to the online orders. The system will then display an order to the chefs, so they can begin preparation. This will assist in reducing the number of miscommunications between the customers, servers, and chefs. Resulting in better customer service, reduced returns in meals sent back to the chefs, and less inventory waste. After the food order is completed, the chef will be able to mark an order as completed. After food preparation is complete, the system will color coordinate the order, so the employees will be able to differentiate the orders among customers. A color will be displayed by the system for easy recognition whether the order is dine-in, take-out, or delivery.

Additionally, the system will keep track of the food that is in and out of stock. If an item is out of stock, it will display so on the site for online orders, and when an employee enters a customer’s order. Management will be able to access the system, allowing manual addition or removal of the current inventory in the system if needed. Overall, we want the restaurant ordering system to be easy to use, whether it is for the customers, chefs, and employees.

* 1. **Project Overview**

This project is to develop software for the restaurant that can take orders and manage them. It manages the orders according to the numbers; each order successfully processed will be assigned a number. The software displays the detailed information of the order electronically for the employees and the customers about the orders items, price, time when the order was placed, whether it was take-out or dine-in orders, any comments from the customers, and their order number. It processes the order in person and online for the customer. In person, restaurant employees will use the system to place walk-in and phone-in orders, while customers can place their own orders using the online system. There will be a mobile app for both Android and iOS-based systems. The app will help to complete the ordering process successfully. The app will allow the customer to place an order by following the specific instructions; the customer can search for the menu items, choose the items, choose whether to take out the items or dine-in, select the payment methods, pay for the order. For in-person orders, it follows the same patterns as ordering online orders, whereas restaurant employees will assist the customer to place the orders.

1. **General Description**
   1. **Project Features / Functions**
   2. **User Stories**
      1. As a customer, I want to order my food online so I can pickup in store.
      2. As a customer, I want to order in person so I can dine in.
      3. As a customer, I want to cancel an order after the order has been placed.
      4. As a customer, I want to scan coupon codes with my phone's camera.
      5. As an employee, I want to input the customer’s order so the cooks can prepare the food.
      6. As an employee, I want to be able to take payment over the phone and in store so I can put the order in.
      7. As an employee, I want to be able to manually set the number of units of a food item we have left.
      8. As a cook, I want to have the order displayed on a screen, so I know what to cook for each order.
      9. As a cook, I want to mark an order as complete, so the employee can serve the order.
      10. As a manager, I want to add new food items to the system.
      11. As a manager, I want to set an alert that will trigger when a specific food item drops below a certain amount.
      12. As a manager, I want to be alerted when one or more food items is below a set amount.
   3. **Use Case**

|  |  |
| --- | --- |
| **Name of Use Case:** | Customer orders food in person for dine or take out |
| **Description:** | Customer orders food from the menu. |
| **Actors:** | Customer, employee, cook, system |
| **Preconditions:** | 1. Customer selects something from the menu 2. Customer pays for food 3. Food in customers order is available |
| **Postconditions:** | 1. Customer receives food 2. Food supply is adjusted |
| **Flow:** | 1. Customer orders food in person 2. Employee puts order in system 3. System checks to make sure items in the order are available 4. Employee takes money for the order and runs it through the system 5. Payment is validated by system if card is used 6. System sends order to cooks screen 7. Cook sees the order on a screen 8. Cook makes the order 9. Cook marks an order as complete when it’s put together 10. Employee gets the order and gives it to customer |
| **Alternative Flows:** | In step 3 of the normal flow, if the restaurant does not have part of the order available   1. Employee lets the customer know the item is out of stock and asks if the customer wants an alternative   In step 4 of the normal flow, if the customer pays with cash   1. Employee puts amount in system 2. System shows the change the customer needs 3. Employee gives the customer change |
| **Exceptions:** | Payment doesn’t go through   1. Request alternative payment until the order is paid. 2. Use case continues at step 6 |
| **Requirements:** | The following requirements must be met before execution of the use case   1. The order must be paid in full |

|  |  |
| --- | --- |
| **Name of Use Case:** | Customer orders food over the phone for pickup |
| **Description:** | Customer orders food from the over the phone for take out |
| **Actors:** | Customer, employee, cook, system |
| **Preconditions:** | 1. Customer selects something from the menu 2. Customer pays for food 3. Food in customers order is available |
| **Postconditions:** | 1. Customer receives food   1. Food supply is adjusted |
| **Flow:** | 1. Customer orders food over the phone   1. Employee puts order in system 2. System checks to make sure items in the order are available 3. Employee puts customers card number in the system 4. Payment is validated by system 5. System sends order to cooks screen 6. Cook sees the order on a screen 7. Cook makes the order 8. Cook marks an order as complete when it’s put together 9. Employee gets the order puts it on a shelf 10. Customer picks up order |
| **Alternative Flows:** | In step 3 of the normal flow, if the restaurant does not have part of the order available   1. Employee lets the customer know the item is out of stock and asks if the customer wants an alternative |
| **Exceptions:** | Payment doesn’t go through   1. Request alternative payment until the order is paid. 2. Use case continues at step 6 |
| **Requirements:** | The following requirements must be met before execution of the use case   1. The order must be paid in full |

|  |  |
| --- | --- |
| **Name of Use Case:** | Customer orders food online for pickup |
| **Description:** | Customer orders food and pays online. |
| **Actors:** | Customer, employee, cook, system |
| **Preconditions:** | 1. Customer selects something from an online menu 2. Customer pays for food 3. Food in customers order is available |
| **Postconditions:** | 1. Customer comes to pick up food 2. Food supply is adjusted |
| **Flow:** | 1. Customer goes to online ordering system 2. System asks if this is delivery or pickup 3. Customer chooses pickup 4. System displays available food 5. Customer orders food 6. Customer pays for food at online checkout 7. Payment confirmed 8. System sends order to cooks screen 9. Cook sees the order on a screen 10. Cook makes the order 11. Cook marks an order as complete once it’s put together 12. Employee puts the order on pickup shelf for customer 13. Customer picks up the order |
| **Alternative Flows:** | In step 6 of the normal flow, if the payment doesn’t go through.   1. Request an alternative payment until order goes through   In step 11 of the normal flow, if the order is never picked up.   1. Throw away the food 2. Customer calls to request a refund 3. Explain refund policy |
| **Exceptions:** | Payment doesn’t go through   1. Request alternative payment until the order is paid. 2. Use case continues at step 6 |
| **Requirements:** | The following requirements must be met before execution of the use case   1. The order must be paid in full before online order goes through. |

|  |  |
| --- | --- |
| **Name of Use Case:** | Customer orders food online for delivery |
| **Description:** | Customer orders food for delivery and pays online. |
| **Actors:** | Customer, employee, cook, system, delivery person |
| **Preconditions:** | 1. Customer selects something from an online menu 2. Customer pays for food 3. Food in customers order is available |
| **Postconditions:** | 1. Food is delivered 2. Food supply is adjusted |
| **Flow:** | 1. Customer goes to online ordering system  2. System asks if this is delivery or pickup  3. Customer chooses delivery  4. System displays available food  5. Customer orders food  6. Customer pays for food at online checkout  7. Payment confirmed  8. System sends order to cooks screen  9. Cook sees the order on a screen  10. Cook makes the order  11. Cook marks an order as complete for delivery once it’s put together  12. Delivery person gets the food  13. Delivery person delivers food to customer |
| **Alternative Flows:** | In step 6 of the normal flow, if the payment doesn’t go through.   1. Request an alternative payment until order goes through   In step 11 of the normal flow, if incorrect address is given or door is not answered.   1. Throw away the food 2. Alert customer the food was undeliverable 3. Explain refund policy |
| **Exceptions:** | Payment doesn’t go through   1. Request alternative payment until the order is paid. 2. Use case continues at step 6 |
| **Requirements:** | The following requirements must be met before execution of the use case   1. The order must be paid in full before online order goes through. |

|  |  |
| --- | --- |
| **Name of Use Case:** | Customer scans coupon |
| **Description:** | Customer uses phone to scan a coupon |
| **Actors:** | Customer, phone, system, employee |
| **Preconditions:** | 1. Customer selects food items 2. Food items are available |
| **Postconditions:** | 1. Customer pays for food items 2. Customer receives discount on their order 3. Customer receives food items 4. Food supply is adjusted |
| **Flow:** | 1. Customer selects food items 2. System checks to make sure food items are available 3. Customer proceeds to checkout 4. System prompts customer to scan coupon 5. Customer scans coupon 6. System adjusts the payment amount 7. Payment is validated 8. System sends order to cook screen 9. Cook sees the order on screen 10. Cook marks an order as once it is put together 11. System adjusts food supply 12. Employee brings food to customer |
| **Alternative Flows:** | If customer fails to scan their coupon   1. Customer prompted to try again |
| **Exceptions:** | If payment fails to validate   1. Request alternate payment until the order is paid |
| **Requirements:** | 1. Customer has coupon to scan |

|  |  |
| --- | --- |
| **Name of Use Case:** | Manager sets food alert |
| **Description:** | Manager uses application to set an alert that will trigger when the systems food supply goes below a certain level |
| **Actors:** | Manager, system |
| **Preconditions:** | 1. Food item is in the system |
| **Postconditions:** | 1. Food item will alert the manager when it drops below a certain amount |
| **Flow:** | 1. Manager logs into system 2. Manager selects a food item listed in the system 3. Manager marks it to alert them when number of units drops below a set amount 4. System prompts manager to input amount the alert will go off 5. Manager input a number of units 6. System records the number of units the alert goes off at 7. System alerts the manager that an alert will go off when the units of food drop below the select amount |
| **Alternative Flows:** | If an alert already exists for that food item   1. Manager selects a food item listed in the system 2. Manager inputs a new value at which the alert will go off 3. Continue at step 6 |
| **Exceptions:** | 1. Manager inputs number below 0 sets alert to 0 |
| **Requirements:** | Food item is already in the system |

|  |  |
| --- | --- |
| **Name of Use Case:** | Manager gets food count alert |
| **Description:** | One or more alerts have been triggered by the systems food counter falling below a preset level |
| **Actors:** | Employee, system, manager |
| **Preconditions:** | 1. An alert has been set 2. Food counter in the system is above the alert trigger |
| **Postconditions:** | 1. Food counter in the system is below the alert trigger amount 2. Manager is alerted to the food count |
| **Flow:** | 1. System sends order to cook 2. Cook prepares food 3. Cook marks down the order as complete in the system 4. System lowers the units of food items below preset threshold 5. System sends alert to manager on their work computer |
| **Alternative Flows:** | 1. An employee manually sets the food amount to below the threshold 2. System sends alert to manager on their work computer |
| **Exceptions:** | If the amount of food units was below threshold before the amount was lowered no alert is sent |
| **Requirements:** | 1. Alert is set to an amount 2. Food counter is lowered to or below preset amount |

|  |  |
| --- | --- |
| **Name of Use Case:** | Add a food item |
| **Description:** | Manager adds new food item to be tracked by the system |
| **Actors:** | Manager, system |
| **Preconditions:** | 1. Food item does not already exist in the system |
| **Postconditions:** | 1. Food item exists in the system 2. Food item has a number of units stored in the system |
| **Flow:** | 1. Manager logs in 2. Manager selects add food item 3. System prompts for name of food item 4. Manager inputs name of food item 5. System prompts for number of units of food item 6. Manager input number of units of food item 7. System saves the name and the number of units of food item |
| **Alternative Flows:** | If manager inputs the name of an already existing food item   1. System prompts manager for new name of the food item |
| **Exceptions:** | If manager inputs a number less than 0 for the number of units of food item   1. System sets the number of units of that food item to 0 |
| **Requirements:** |  |

|  |  |
| --- | --- |
| **Name of Use Case:** | Manually setting food number |
| **Description:** | Employee increases the number of units of food in the system, or decreases the number of units of food in the system without using orders |
| **Actors:** | Employee, system |
| **Preconditions:** | 1. Food item exists in the system |
| **Postconditions:** | 1. Food item will have a different or the same number of units marked in the system |
| **Flow:** | 1. Employee logs into system 2. Employee selects a food item 3. Employee selects option to change the number of units of that food item 4. System prompts user for number of units 5. Employee inputs the number of units 6. System saves the new number of units of food item |
| **Alternative Flows:** | If employee selects option to change the number of units of a food item   1. System prompts user for number of units 2. Employee cancels the change 3. System returns to the previous menu |
| **Exceptions:** | If employee sets the number of food items in the system to less than 0   1. System save the number of units of that food item to 0 |
| **Requirements:** | 1. Food item already exists in the system |

1. **Team Collaboration and Documentation Tools**
   1. Team Collaboration Tools
      1. Microsoft Teams and Basecamp
   2. Documentation Tools
      1. Basecamp
2. **Project Management Plan**
3. **Business Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| BR1 | The system must be available during operating hours 6 days a week, Monday through Saturday. | M |
| BR2 | Reject invalid credit cards. | M |
| BR3 | Track inventory items. | S |
| BR4 | Alert staff when inventory items fall below a preset threshold. | S |
| BR5 | Inform the cook when an order is placed and the content of that order. | S |
| BR6 | Take orders automatically over the phone. | C |
| BR7 | Take mobile orders. | S |
| BR8 | Make reports about the items ordered weekly. | C |

1. **User Requirements**

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement Description | MOSCOW |
| UR1 | The system accepts online orders, allows the employee to take the phone and in-person orders. | M |
| UR2 | The order screen displays all the available menu items and its price. | M |
| UR3 | The system accepts the additional comments for food allergies, food restrictions, personal preferences, and additional information. | M |
| UR4 | Display the detailed information of the order in the screen for review. | M |
| UR5 | Display available payment methods. | M |
| UR6 | Order must be paid in full to be successfully processed or the system will decline it. | M |
| UR7 | Send the order information to the printer, text or email the full receipts for customer. | S |
| UR8 | Successfully processed orders that are paid in full are sent to the kitchen. | M |
| UR9 | Display order information on the cook screen. | M |
| UR10 | Notify the employee and customer when the order gets ready. | S |
| UR11 | The system keeps detailed information of the inventory and able to manage it manually. | M |
| UR12 | The system able to track the inventory items and notify when it’s out of stock. | M |
| UR13 | Numerous users can use the system without any noticeable systems delays. | S |
| UR14 | System able to display the total revenue information for management team. | S |

1. **Functional Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| FR1 | Keeps a record of food inventory. |  |
| FR2 | Receives online orders and orders from employees. | M |
| FR3 | Calculates customers total. | M |
| FR4 | Calculates customers change. | S |
| FR5 | Print receipts for orders. | M |
| FR6 | Sends email confirmation for online orders. | M |
| FR7 | Displays online and in person orders to cook. | S |
| FR8 | Keeps a record of revenue. | S |
| FR9 | Processes credit card payments. | M |
| FR10 | Discounts food when coupon is scanned. | S |
| FR11 | Allows manual overrides with managers code. | M |
| FR12 | Color codes online vs in restaurant orders. | W |
| FR13 | Stores customers information when an online order is made. | S |
| FR14 | Alerts manager when inventory is low. | C |
| FR15 | Alerts employee when order is ready for customer. | M |
| FR16 | Alerts customer when online order is ready for pickup. | C |
| FR17 | Allows customers to download pdf of menu | C |
| FR18 | If user has forgotten password: A clickable link is emailed to the user to create a new password when a “confirmed” stored email has been provided | M |
| FR19 | When user enters credit card information or uses saved information - CSC/CVC requires entry | M |
| FR20 | Allows orders placed during business hours only. <<<< Sends alert to customer if order is placed during non-business hours. Showing the hours of business | M |

1. **Non-Functional Requirements**

**8.1 Online Account Security**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| NFR1.1 | User must have a unique username when registering for an account. | M |
| NFR1.2 | User must have a secure password.  Minimum requirements: 1 x Capital letter, 1 x Number, 1 x Symbol, Minimum 7 Characters. | M |
| NFR1.3 | Unique email associated with the account – One account per email address. | M |
| NFR1.4 | Confirmation email sent within 3 minutes of when a new account has been created. | S |
| NFR1.5 | Encryption of user passwords and all personal information that has been provided. | M |
| NFR1.6 | Should not save Card Security Code (CSC)/Card Verification Value (CVV) for the credit cards for security reasons. | M |
| NFR1.7 | System will logout user after 1 hour of inactivity. | S |
| **8.2 Systems and Security** |  |  |
| Requirement ID | Requirement Description | MOSCOW |
| NFR2.1 | Secure encrypted website (HTTPS and SSL Certification). | M |
| NFR2.2 | Local Server (or Cloud Based Hosting). | M |
| NFR2.3 | Secure and fast internet connection. | M |
| NFR2.4 | Scheduled maintenance for bugs, fixing issues, and database updates. | M |
| NFR2.5 | Only Administrators have full access to the data and information provided. | S |
| NFR2.6 | User will have the ability access the system from multiple operating systems (Windows, Apple, Android, etc.) and devices (Desktop, tablet/phone) | M |
| NFR2.7 | System should be able to easily restart in case of errors. Not taking longer than 5 minutes. | S |
| NFR2.8 | System can handle more than 75 users at a given point of time – includes in restaurant and online. | M |
| **8.3 Orders/Usability** |  |  |
| Requirement ID | Requirement Description | MOSCOW |
| NFR3.1 | Secure POS (Point-of-Sale) hardware and software system.  Enables: Payment process, inventory management, POS reports, receipts, tipping. | M |
| NFR3.2 | Menus should be easy to navigate for the user. | M |
| NFR3.3 | For Online: Confirmation screen sent before order is placed. | S |
| NFR3.4 | For Online: Email confirmations sent within 3 minutes after an order has been placed. | S |
| NFR3.5 | Orders should be made immediately available to managers and chefs to access. | M |

1. **Implementation (Performance) Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| IR1 | The application will allow no more than 5 users for testing. | M |
| IR2 | The application will return recommended food meals based on the customers previous ordering history. | M |
| IR3 | The application will pop up with an error message if the system is not responding |  |
| IR4 | If user enters in password wrong more than 3 times the system will have a wait period and the account will be on lock | M |