**Restaurant Automation**

*LAL: Logan Connolly, Abel Briggs, Lizkebe Manjuk*

1. **Introduction**

Our aim is to create a software system that will allow restaurants to better take customer orders, automate taking orders, handle payment automatically, and assist employees on what table each order should go to.

1. **Proposed System**

Our proposal is designed to be a user-facing kiosk that allows users to, for the most part, serve themselves by ordering food from the restaurant without the need of a cashier or other staff to assist them. The system will also display these orders to the chefs, allowing them to mark orders as completed or in progress.

* 1. *Functional Requirements*

IN-FN-01: The system shall allow the user to start their order by hitting the start order button.

IN-FN-02: The system shall take in orders (food and drink).

IN-FN-03: The system shall read in payment via card, Near Field Communication (NFC), and cash.

IN-FN-04: The system shall allow chefs to mark orders as complete.

OUT-FN-10: The system shall display all available food (the menu) to the customer.

OUT-FN-11: The system shall display each order to the chefs.

OUT-FN-12: The system shall display when the order is complete to the customer.

OUT-FN-13: The system shall display the total price of the order.

OUT-FN-14: The system shall print a receipt when the order is processed.

OUT-FN-15: The system shall display when the order is ready to be picked up.

OUT-FN-16: The system shall display a confirmation message to confirm the order when the customer attempts to check out.

PRO-FN-19: The system shall calculate sales tax.

PRO-FN-20: The system shall add sales tax to the bill.

PRO-FN-21: The system shall calculate the bill based on items ordered.

PRO-FN-22: The system shall keep track of how many orders go through in a day.

PRO-FN-23: The system shall keep track of analytics on how much each item is ordered.

PRO-FN-24: The system shall generate a report of analytics of order statistics.

PRO-FN-25: The system shall generate and maintain order numbers.

* 1. *Non-Functional Requirements*

IN-NF-05: The system must take in input through touch screen devices that are able to run Java Virtual Machine.

IN-NF-06: The system must be able to read both chip and swipe cards.

IN-NF-07: The system must be equipped with NFC chips to be able to read data from NFC payments.

IN-NF-08: The system must be equipped with a cash terminal.

IN-NF-09: The system must respond to user input within 500 ms.

OUT-NF-17: The system must be equipped with a printer to print receipts.

OUT-NF-18: The system must be network equipped to display the order to the chef.

PRO-NF-26: The system shall be implemented using the Java Virtual Machine (JVM).

PRO-NF-27: The system shall run on any OS that is able to run the JVM (Windows, Linux, etc.).

PRO-NF-28: The system must have an administrative control panel.

PRO-NF-29: The system shall not store credit/debit card information.

PRO-NF-30: The system shall display the orders to the chefs by communicating using the local area network (LAN).

* 1. *System Models*

Our system is very limited in scope so we have few use cases. We have use cases for each action that is initiated by the user and a diagram of what each actor can do to the system.

* + 1. Use Cases

|  |  |
| --- | --- |
| *Use Case Name* | Start Order |
| *Participating Actors* | Initiated by Customer |
| *Flow Of Events* | 1. Customer walks up to the kiosk and presses the start order button 2. System brings up the main order menu containing a list of food categories |
| *Entry condition* | Customer hits start order button |
| *Exit conditions* | Customer hits cancel button, customer starts order |
| *Quality requirements* | System will respond to user input within 500 milliseconds. |

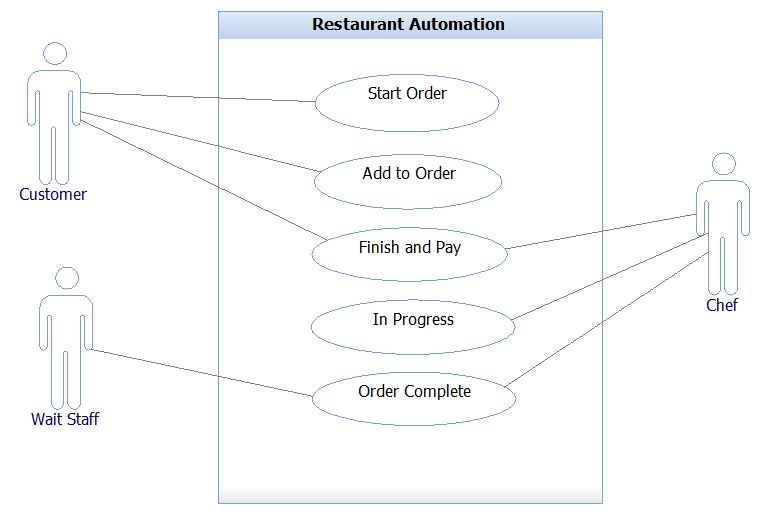
|  |  |
| --- | --- |
| *Use Case Name* | Add to Order |
| *Participating Actors* | Initiated by Customer |
| *Flow Of Events* | 1. Customer looks over menu options and selects an item they want, they hit add to order 2. The system retrieves the data of the selected item and adds it to an order summary on the right side of the screen. |
| *Entry condition* | Customer has already started an order |
| *Exit conditions* | Customer hits cancel order or hits finish order |
| *Quality requirements* | System responds to user input within 500 ms. |

|  |  |
| --- | --- |
| *Use Case Name* | Finish Order |
| *Participating Actors* | Initiated by Customer  Acknowledged by Chef |
| *Flow Of Events* | 1. When the customer feels the order is complete, they hit the Finish and Pay button. 2. System will show the total price and order summary of the order and give the options to pay. 3. Customer picks their payment option and pays the bill. 4. The system will confirm payment and display the order to the chef and indicates that the order has not been initiated yet. |
| *Entry condition* | Customer has finished adding items to their order. |
| *Exit conditions* | Customer pays the bill and chef marks order as in progress |
| *Quality requirements* | System responds to user input within 500 ms. |

|  |  |
| --- | --- |
| *Use Case Name* | In Progress |
| *Participating Actors* | Initiated by Chef |
| *Flow Of Events* | 1. Chef sees the order on the display and hits the in progress button 2. The system marks the order as in progress on the display 3. Chef makes the order. |
| *Entry condition* | Customer finishes order and the order is displayed to the chef |
| *Exit conditions* | Chef marks order as complete |
| *Quality requirements* | System responds to user input within 500 ms. |

|  |  |
| --- | --- |
| *Use Case Name* | Order Complete |
| *Participating Actors* | Initiated by Chef  Processed by Wait staff  Received by Customer |
| *Flow Of Events* | 1. Chef finishes making the order and hits the Order Complete button. 2. System prints a receipt for the order and deletes the order from the list of orders. 3. The wait staff takes the order and receipt to the customer. |
| *Entry condition* | Chef finishes making the order |
| *Exit conditions* | Order is completed |
| *Quality requirements* | System responds to user input within 500 ms. |

* + 1. Use Case Diagrams



1. **Glossary**

JVM – Java Virtual Machine

LAN – Local Area Network

NFC – Near Field Communication

OS – Operating System

1. **Reference**

We have not referenced any outside material,.