Final Project Documentation

Thursday 5PM — 7PM Group 3

Prepared by

Richard Barnett	***620150541***	***richard.barnett@mymona.uwi.edu***
Zaria Chen Shui	***620093688***	***zariachenshui@gmail.com***
Delano Brown	***620138606***	***delano.brown10@yahoo.com***
Danielle Johnson	***620139242***	***ddjohhnson1@gmail.com***
Kelandra Pringle	***620143953***	***kelandrapringle@gmail.com***

Course Instructor: **Dr. Ricardo Anderson**

Course: **COMP2140 – Introduction to Software Engineering**

Studio Facilitator: ***Mr. Andrew Litchmore***

Date: **December 1, 2022**

Software Requirements Specification

for

Krissy's Kiosk (Group 3)

Version 2.0

Prepared by

Richard Barnett	**620150541**	**richard.barnett@mymona.uwi.edu**
Danielle johnson	**620139242**	**ddjohhnson1@gmail.com**
Zaria Chen Shui	**620093688**	**zariachenshui@gmail.com**
Delano Brown	**620138606**	**delano.brown10@yahoo.com**
Kelandra Pringle	**620143953**	**kelandra.pringle@mymona.uwi.edu**

Course Instructor: **Dr. Ricardo Anderson**

Course: **COMP2140 – Introduction to Software Engineering**

Studio Facilitator: ***Mr. Andrew Litchmore***

Date: **November 5, 2022**

Table of Contents

Revisions

Version	*Primary Author(s)*	*Description of Version*	*Date Completed*
1.0	*Richard Barnett* *Zaria Chen Shui* *Danielle Johnson* *Kelandra Pringle* *Delano Brown*	*Completed version with all expected parts completed.*	*09/11/22*
2.0	*Richard Barnett* *Zaria Chen Shui* *Danielle Johnson* *Kelandra Pringle* *Delano Brown*	*Alterations made to functional requirements, their specifications, and the use case diagram.*	*01/12/22*

1 Overall Description

1.1. Product Context and Need

Krissy's Kiosk is a small business housed in one of the many retail shops located on the campus of the University of the West Indies, close to the faculty of Social Sciences. It has been operational for 15 years and provides students with small meals , snacks, phone chargers and other small useful electronics.

The current fixed location of the business limits customer traffic. Due to the size of the campus, students from other faculties are unable to support the business. There is currently no system for placing orders or receiving product information outside of the physical location. As it relates to inventory, the business currently has no formal system in place and instead relies on memory, or until a product is completely out of stock before a new order is placed. There is no means for product sale analysis and therefore the employees are unable to make educated decisions relating to the products offered. eg(the best performing product, the product that makes the most revenue etc)

1.2. Product Functionality

1.2.1 The proposed system should allow remote customers to view the available products and place orders.
1.2.2 The system should also be able to provide product information (new item, product no longer available etc) to remote customers through notifications.
1.2.3 The system should allow inventory to be tracked and provide alerts relating to the restock of products.
1.2.4 The product should be able to generate reports relating to product performance,

1.3. Stakeholders and Users Characteristics

Primary stakeholders include Krissy, the business owner and the individual whom developers will be corresponding with throughout the duration of the project. She will have the highest level of security privilege and will have access to all aspects of the software. The employees that will be utilising the system are also a part of the stakeholders and will only be granted access to functionalities directly related to processing and completing an order. Remote customers that will mainly be students of the university, will also be a part of the primary stakeholders and will be limited to functionalities relating to viewing products and order placement.
Secondary Stakeholders include the University of West Indies as the entity that host the location of the business. While they will have no formal interaction with the system, they affect the customer

base by providing the location for the business, as well as facilitating the internet that allows the orders to be received.

1.4. Operating Environment

The program will be accessed by several customers over several different machines. The program will mostly be used on mobile devices and must therefore be able to operate on any macOS and android operating systems.The software should also be supported by any 32-bit CPU operating on Windows 8 or later, or operating on the macOS. The device utilised by the company will need memory of at least 16GB and a recommended storage of 64GB to support inventory information, reports as well as product information.

1.5. Design and Implementation Constraints

Programming Language: Project implementation is limited to the use and functionalities of Python due to the assignment specifications as well as group competency.
Program maintenance: The maintenance of the software will be the responsibility of the company owner.

1.6. Assumptions and Dependencies

1.6.1 It is assumed that the company has at least one device that has the specifications required to run the software.
1.6.2 It is assumed that company will ensure software is adequately maintained
1.6.3 It is assumed that employees will be adequately trained to use software

2. Specific Requirements

2.1. External Interface Requirements

2.1.1. Hardware Interfaces

Our system will be a PC based system with a text based interface. This will use a keyboard to navigate through the different areas and pages in the system and also enter data by the user by using the alphanumeric data entry functionalities of the device.

2.1.2. Software Interfaces

We have decided to use the Windows operating system for the system as it is easy to use and easily accessible. For data storage, we will be using windows text files as it is easy to use and easily accessible.

2.1.3. Communications Interfaces

Sensitive data provided by the users will be encrypted in order to provide for a higher level of security. Their data will be encrypted using a key. This key will be well known within the software itself so that data can be easily encrypted and decrypted. All other communication will be handled by the operating system.

2.2. Functional Requirements

2.2.1 **Requirement ID**: 1 Add and Edit Product
Use Case: add and edit product
Rationale: The owner and employees need to be able to add new products and make changes to the details of existing products so that customers are able to see up-to-date details on each product.
User Requirement: The system shall allow an authorized user to add a new product to the product list of the system and edit an existing product.
System Requirements:
2.2.1.1 After confirming that the user is authorized to edit the product list, the system shall allow the user to add a name and price for the new product.
2.2.1.2 The system shall allow the user to add an optional product description.
2.2.1.3 For products that already exist in the system, the system shall provide a means for editing any of the product details.
2.2.1.4 The system shall provide a means for the user to ask if customers would like to be notified of the addition or change to the product list.
2.2.1.5 The system shall allow the user to view entered product data and confirm submission before it is added to the product list.

Acceptance Criteria:
1. To add a new product or edit an existing product, the user must be authorized to do so.
2. The data accepted by the system for each product detail must correspond to the appropriate type for that detail, i.e. alphanumeric characters for name and description and integers for price.
3. New product data must be saved and reflected in the system's product list.
4. Changes to existing products must override old data and save new data 100% of the time and be reflected on the system.

Relates to/Dependencies: Send Notification
Priority: High
Team Owner: Zaria Chen Shui

2.2.2 Requirement ID: 2 Send Notification
Use Case: send notification
Rationale: This requirement is necessary in order for users to be notified when there are changes in products and their information.
User Requirement: The system should allow specific users to be notified with information related to the change in a product and its information.
System Requirements:
2.2.2.1. The system should have an area where notifications are stored.
2.2.2.2. The system should allow authorized users to send out notifications when a new product is added to the system and when a product price has been changed.
2.2.2.3 The system should provide a means for an authorized user to send out custom notifications written by the user.
2.2.2.4 The system should provide a means for a user to indicate that they would like to be sent notifications about product list changes in future.
2.2.2.5 The system should provide a means for users to view notifications.

Acceptance Criteria:
1. Authorized users are able to send out notifications 100% of the time having requested to send out notifications.
2. Users are able to view notifications 100% of the time having requested to be shown notifications.

Relates to/Dependencies: Add and Edit Product
Priority: Medium
Team Owner: Richard Barnett and Zaria Chen Shui

2.2.3 Requirement ID: 3 Place Order
Use Case: place order

Rationale: This requirement is necessary in order for customers to place an order for the products offered.
User Requirement: The system shall allow the users to place an order for a product/products.
System Requirements:
2.2.3.1. The system shall display all products available and their prices
2.2.3.2. System shall allow users to select products and desired quantity
2.2.3.3. System should provide the total cost of all selections
2.2.3.4. System should display order confirmation with the order number.

Acceptance Criteria:
1. The user is allowed to place an order having selected at least one product and a numerical quantity 100% of the time.
2. The order placed must be reflected on the system.

Relates to/Dependencies: None
Priority: High
Team Owner: Danielle Johnson

2.2.4 Requirement ID: 4 View Product
Use Case: view product
Rationale: The users should have the ability to search for a particular product, arrange products in order based on their price and retrieve products based on their preferences.
User Requirement: The system shall allow the user to search for a product, retrieve the product. This will also allow the user to sort the products in ascending or descending order based on the product price.
System Requirements:
2.2.4.1 The system shall allow the user to search for a product(s) once available.
2.2.4.2 After searching for a product, the system shall allow the user to retrieve or select a product that user wish to purchase
2.2.4.3 System is able to view all the information related to the selected product
Acceptance Criteria: The user is able to view all products within the system 100% of the time would be how we can prove/check that the viewing of products is working perfectly.

Relates to/Dependencies: None
Priority: High
Team Owner: Kelandra Pringle

2.2.5 Requirement ID: 5 Cancel Order
Use Case: cancel order

Rationale: This requirement is necessary for the user to be able to cancel their order after they have placed one on the system and no longer wants it.
User Requirement: The user should have the ability to cancel any of their orders after they have placed them.
System Requirements:
2.2.5.1The system shall display all of the customers orders and their details including the order number.
2.2.5.2The system shall display a message saying that the user has no orders if they have no orders and request to cancel an order.
2.2.5.3The system shall accept a valid order number that corresponds to the user's orders which will be used to cancel the order.
2.2.5.4the system shall add back the quantity of the product that was ordered and then cancelled to the inventory after it has successfully been cancelled.
2.2.5.5The system shall display a confirmation message when an order has been cancelled successfully.
2.2.5.6The system shall remove the cancelled order from the orders database.
Acceptance Criteria:

1. All customers are able to cancel an order 100% of the time once they have placed an order and it has not been delivered as yet.
2. Once an order is cancelled it should be removed from the orders database 100% of the time.

Relates to/Dependencies: Place Order
Priority: High
Team Owner: Richard Barnett

2.2.6 Requirement ID: 6 Update Inventory
Use Case: Update inventory
Rationale: This requirement is necessary for the user to update the quantity of each product available to customers.

User Requirement: The user should have the ability to search for a specific and update it's available quantity.

System Requirements:
2.2.6.1The system shall keep track of the quantity of each product in inventory.
2.2.6.2 The system shall verify if the user is authorized to make changes to inventory.
2.2.6.3The system shall allow authorized users to search for a specific product and update its quantity.

Acceptance Criteria:

1. Changes to inventory must be reflected on the system 100% of the time.

2. Once product quantity is updated from zero (0), product should become available 100% of the time.

Relates to/Dependencies: Stock Alerts, Generate Inventory Report
Priority: Medium
Team Owner: Delano Brown

2.2.7 Requirement ID: 7 Stock Alerts
Use Case: Stock Alert
Rationale: This requirement is necessary to notify the user whenever stock is running low.

User Requirement: The system should allow specific users to be notified whenever stock levels are running low.

System Requirements:

2.2.7.1 The system shall verify if a user is authorized to view stock related alerts before allowing these alerts to be viewed by the user.

2.2.7.2 The system shall store a list of messages to be sent whenever stock numbers reach below 10 items of any product.

2.2.7.3 The system shall send an alert to the authorized user corresponding to the relevant changes in stock levels, once the quantity of a product in inventory falls below a given threshold.

2.2.7.4 The system shall provide a means to check if the quantity of a product in stock has fallen below the threshold after an order has been placed and after a product's quantity has been edited.

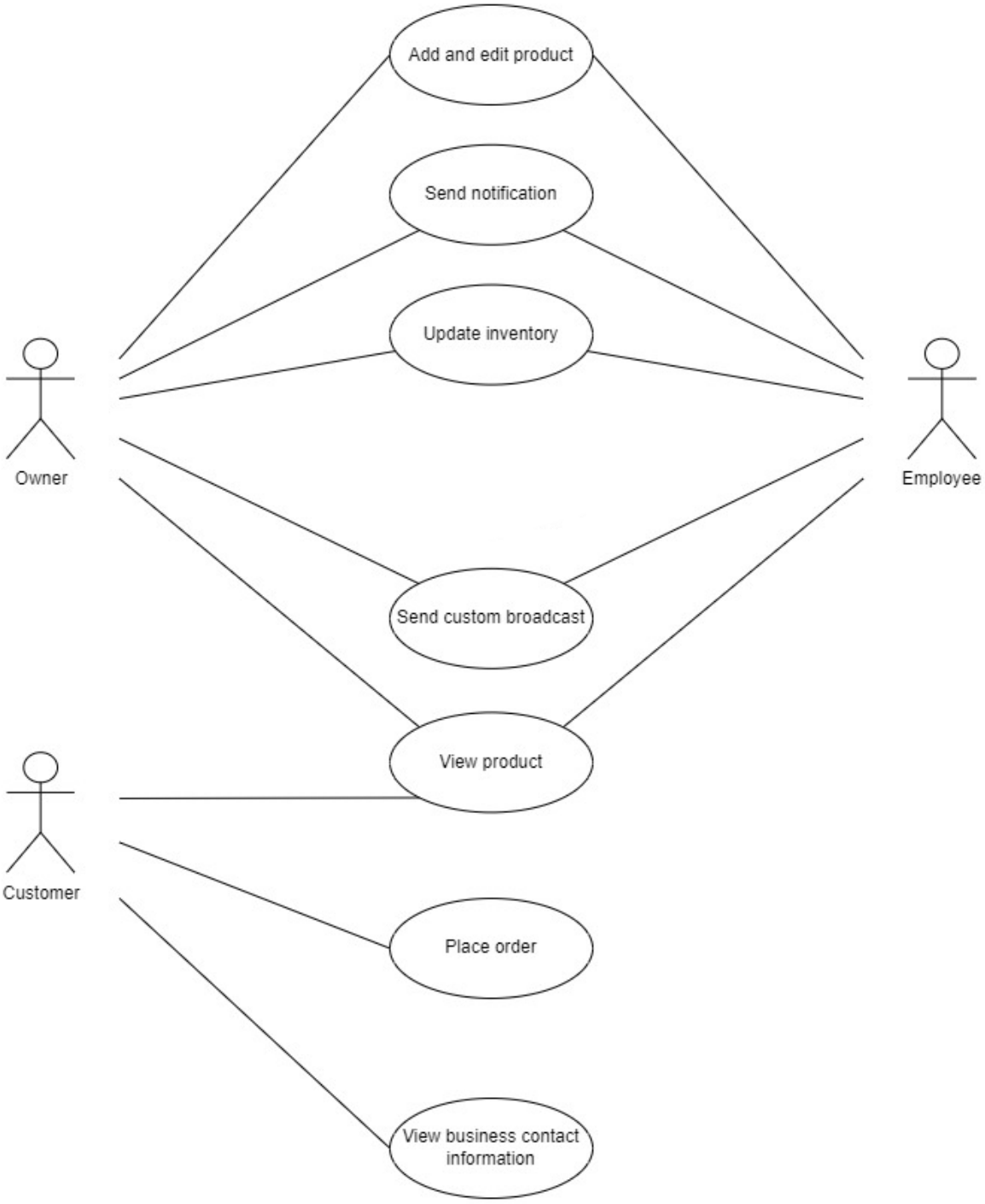
Acceptance Criteria:

1. Once a product's stock level falls below the desired threshold, alerts should be sent to authorized users 100% of the time.

Relates to/Dependencies: Update Inventory, Add or Edit Product, Send Notification
Priority: High
Team Owner: Delano Brown and Zaria Chen Shui

2.3. Behaviour Requirements

2.3.1. Use Case View



Name	1 Add and edit product
Description	New products can be added to the system by the owner or employee and these authorized users can edit the information of existing products.
Precondition	The user has been authorized to perform these actions
Basic Flow	1. The authorized user should indicate that they would like to add or edit a product. 2. The system should verify that the user is authorized to perform these actions. 3. When the user chooses to add an item, they should verify that this product does not already exist within the system. 4. If the product already exists within the system, the current product details should be displayed, and the user permitted to edit its details. 5. If the product does not exist within the system, the system should allow entry of a product name, price, and optional description.
Postcondition	The new or updated product information is shown to the user.

Name	2 Send notification
Description	An authorized user is able to select when a notification is to be sent out to customers regarding any changes in the product list.
Precondition	The user was confirmed to be an authorized user.
Basic Flow	1. The user should indicate that they would like to setup a notification regarding a new or edited product at the time of the product list adjustment. 2. Customers should indicate whether they are interested in receiving updates regarding prices and new products. 3. A list of customers interested in price and new product notifications should be stored within the system. 4. When an adjustment to the product list is confirmed by the authorized user, the system should send out a notification to each customer

	who has indicated an interest in receiving such notifications.
Postcondition	Relevant customers receive the notification after a trigger event has occurred (either a product price has been adjusted or a new product has been added and the authorized user who made this addition or adjustment has indicated that customers are to be notified of the change).

Name	4 View Product
Description	The user views the product list and product details without being allowed to edit the product list.
Precondition	None
Basic Flow	1. The user indicates that they would like to view the product list. 2. When viewing the product list, the user can sort through the list and select any product to view its unique details, at which point the product details are retrieved for viewing by the user.
Postcondition	The product list and any selected product's details are displayed for the user to view as desired.

Name	5 Cancel Order
Description	Customers are able to cancel any order that they have placed once they have not received it as yet.
Precondition	The user has to be a customer who has placed an order and has not received it as yet.
Basic Flow	1. The system prints the users orders if they have any. 2. If they have orders, the system then prompts the user to enter the order number of the order they would like to cancel. 3. The system then removes the order from the orders database and adds back whatever they ordered to the inventory. 4. The system then sends a confirmation message stating that the order has been

	successfully cancelled
Postcondition	The system has a new order list which no longer has the order that has been cancelled..

Name	6 Update inventory
Description	An authorized user can edit inventory records for any given product.
Precondition	The user is an authorized user.
Basic Flow	1. The user selects a product and indicates that they would like to update its stock information, at which point the system should retrieve and display that product's inventory level. 2. The user should be allowed to confirm their update prior to submission to the system. 3. The system should preserve a log of changes to inventory for report generation purposes.
Postcondition	The updated stock information should be visible for the customer to view.

3. Other Non-functional Requirements

3.1. Performance Requirements

3.1.1 Upon accessing the system, authorized users should be able to add and edit products within 10 keyboard word/phase entries. This is in order to increase efficiency so that the user does not have to spend a lot of time on the system.
3.1.2 When adding or editing products, authorised users should be able to send out a notification to all users with 1 keyboard entry. This is important as users need to be updated to different changes in prices, quantity and products as soon as possible.
3.1.3 Upon accessing the system, users should be able to place an order within 10 keyboard word/phase entries, which would increment by 2 for each type of product they would like to order. This is in order to increase the user friendliness of the system as users are able to place orders in a fast and efficient manner.
3.1.4 When an order is placed, authorised users should see the order details within 10 seconds of it being placed. This is necessary as it is important for the authorised users to see the orders as soon as possible so they can prepare the orders as soon as possible.
3.1.5 Users should be able to log into the system within 2 keyboard word/phrase entries. This is so that the user can get access to the system and its functionalities as soon as possible.
3.1.6Upon accessing the system, users should be able to view products within 5 keyboard word/phrase entries. This is in order to increase the user friendliness of the system as users are able to see all the products available quickly.
3.1.7 All screens should load within 5 seconds. This is in order to give the user access to what they want to view as quickly as possible which increases the user friendliness and efficiency of the system.

3.2. Safety and Security Requirements

3.2.1 When users enter their personal information such as emails into the system, it will not be accessible by any other regular user of the system. This is done by storing their information in a secure storage location that is only accessible by the authorised users and also encrypting the data.
3.2.2 Secured locations on the system are protected by a username and password that is unique to each user so that they will be the only ones with the information to log in and access their information.
3.2.3 When a password is being entered, it will be covered up so that if someone tries to look and see what they are typing, they will be unable to.
3.3.4 Orders sent by users will not be able to be seen by other customers as they will be saved in a secure storage location which is only accessible by authorised users.
3.3.5 Users will be provided with a unique order code which only they have access to, in which they will be required to present in order to collect their order. This prevents random people from taking your order and also ensures that you get the correct order.

3.3. Software Quality Attributes

3.3.1 The software shall be reusable as all the data related to the current business can be wiped and data for a similar business can be easily inserted and used for their business. This includes company name, information, products and their information and different usernames and passwords.
3.3.2 The software shall be adaptable as it has features that allow for authorised users to enter new productions, information and prices efficiently.
3.3.3 The software shall be usable as it shall provide a means for users to place orders and authorised users users will be able to see those orders and prepare them for the customers. The system will also be able to send out notifications to users and allow for inventory management.

Software Design Specification

for

Group 3: Krissy's Kiosk

Version 2.0

Prepared by

Richard Barnett	*620150541*	*richard.barnett@mymona.uwi.edu*
Zaria Chen Shui	*620093688*	*zariachenshui@gmail.com*
Delano Brown	*620138606*	*delano.brown10@yahoo.com*
Danielle Johnson	*620139242*	*ddjohhnson1@gmail.com*
Kelandra Pringle	*620143953*	*kelandrapringle@gmail.com*

Course Instructor:	***Dr. Anderson***
Course:	***COMP2140 – Introduction to Software Engineering***
Studio Facilitator:	**Mr. Litchmore**
Date:	***Nov. 19, 2022***

SDS Table of Contents

Revisions

Version	*Primary Author(s)*	*Description of Version*	*Date Completed*
1.0	*Richard Barnett* *Zaria Chen Shui* *Danielle Johnson* *Kelandra Pringle* *Delano Brown*	*Completed version with all expected parts completed.*	*19/11/22*
2.0	*Richard Barnett* *Zaria Chen Shui* *Danielle Johnson* *Kelandra Pringle* *Delano Brown*	*Alterations made to architectural diagram, component decomposition, and structure chart.*	*01/12/22*

1.0 Project Overview

Krissy's Kiosk is a small business based out of one of the many retail outlets on the University of the West Indies campus near the Faculty of Social Sciences. It has been in operation for 15 years and provides students with refreshments, snacks, cell phone chargers and other small and handy electronic gadgets.

The store's current fixed location limits customer traffic. Due to the large size of the campus, students from other faculties cannot support the operation. There is currently no system for ordering or obtaining product information outside of a physical location. As for inventory, the company currently does not have a formal system, instead relying on memory or being unable to place new orders until the product is completely sold out. With no means of product sales analysis, employees are unable to make informed decisions regarding the products offered. For example B. (best performing product, highest selling product, etc.)

The system will mainly be used for customers unable to visit the physical location. It should be able to display product information , produce reports when prompted and allow customers to place orders. It is also expected to be able to support on screen display to facilitate various functionalities.

2.0 Architectural Design

2.1 General Constraints

The system is intended to be run on devices that are commonly used by students. Regarding a desktop or laptop hardware environment, the system would need to be able to be run under the limitations of typical student laptop specifications, including an general expected maximum RAM availability of 8GB, and Intel i5 processor, and 250GB of storage. While these types of limitations convey the maximum expected general capabilities of user devices, when designing our solution, the device limitations must be considered in tandem with the performance and other non-functional requirements in order to implement an operable solution that has a balance between the environmental limitations and the minimum requirements expected of the system. For this reason, in order to design a solution that meets the requirements while conforming to the limitations, the requirements would be adhered to with minimal or nonexistent additional features.

The majority of students are expected to use Windows as their operating system of choice, which would be capable of running programs written in Python along with their associated databases written in SQL. Since the devices intended to be used to execute the system would be used by students and employees on campus, the system would also engage with the internet connectivity limitations provided by the University of the West Indies, Mona. It is expected that this connectivity will be sufficient for typical use of the system by students and employees.

Because students and employees are the intended users of the system, with each user type having distinct privileges within the system, the system would provide distinct views and available actions to each user based on their type and level of authorization. The access to varying levels of authorization views and processes would imply a limitation upon the system involving security. The level of security necessitated by a suitable system for the client imposes upon the system design itself the limitation that the design must factor in security of the data and processes contained within the system.

2.2 Alternatives Considered

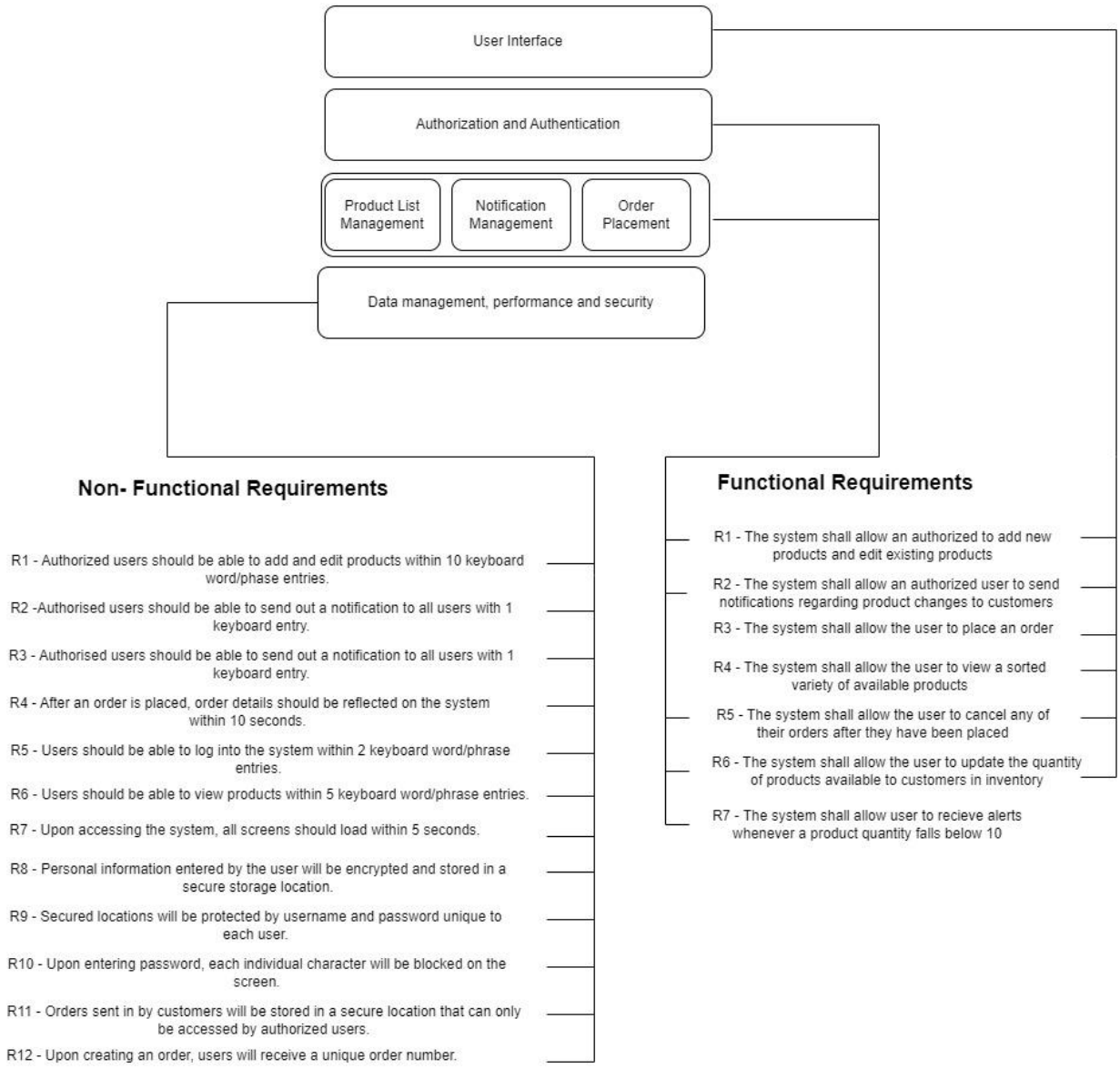
The system was designed to utilize the layered architecture pattern. Other architecture patterns were considered throughout the planning process; however, some patterns weren't as appropriate as the one chosen. These would include the MVC (Model-View-Control) architectural pattern and the Client-server architectural pattern.

The MVC (Model-View-Control) architectural pattern was considered since it is used when there are multiple ways to view or interact with data and because it entails three logical components that interact with each other such as the model, view, and control components. The model component would manage the system data and associated operations on that data such as inventory tracking, product prices and quantity available, the additions of new products added to the system and other relevant data including the quantity of products sold, that are needed to generate various reports. The view component would define and manage the presentation of data to various users such as an overview of products listed in inventory with their corresponding prices and quantity, the way how notifications are viewed by end users, and how generated reports are displayed to end users. The controller component of this architectural pattern would manage the interaction between the system and users, whether it may be navigating through a list actions or search inputs of a specific product.

An advantage of this pattern is that it allows data to change independently of its representation and it supports presentation of the same data in different ways so inventory would be updated automatically based on fulfilled orders and remaining quantities would be reflected in inventory and on reports generated. On the other hand, a disadvantage of this pattern could be the code being too complex due to interaction between data models being too simple. This architectural pattern wasn't as appropriate as it was decided that it would be more feasible to break down the system in a more layered approach where each layer would possess different functionalities and additional layers could be incorporated easily instead of just using three main components.

Another architectural pattern that was considered is the Client-server pattern that is typically used when data in a shared database must be accessed from a range of different locations. This pattern was considered since the functionality of the system is divided into services where each service or functionality of the system would be delivered from separate servers that clients would use. For each functionality of the system would be a specific server designated to carry out specific functionalities such as creating an order, updating inventory, generating reports and so on. An advantage of this pattern is that servers can be distributed across a network where specific services can be accessed by users of certain authorization whether it may be a customer trying to create an order or an employee trying to generate a report and general functionality doesn't need to be implemented by all servers. The main reason this pattern was not chosen to be used is because each service is a single point of failure so susceptible to service attacks or server failure and performance may be an issue due to the network issues as well as the system.

2.3 System Architecture Diagram



2.3.1 Architectural Description

User Interface- This is the means by which the user and the system will interact. In this component, the system shall allow authorized personnel to add new products and edit existing products as well as send notification to customers regarding product changes. Additionally, customers will be able to place an order, view a sorted variety of available products. The owner will be able to generate inventory reports as well update the quantity of inventory that is available.

Authorize and Authentication- All users of the system will be required to verify their authorization if they want to enter the employee side of the program. They will need to provide a username and password in order to gain access to all of its data and functionalities.

Product list, Notification, Order Placement and Reports Management - In this component, the functional requirements will be used where most of the fundamental functionalities are based.
The system shall allow authorized users to add new products to inventory list along with relevant details such as prices and quantity available and users should also be able edit a product in order to make a price adjustment.

The system shall give authorized users the option to send out notifications to customers regarding changes made to different products such as price increases or a product being unavailable.

The system shall give users the option to place orders from a list of products provided by the system.

The system shall provide stock reports to authorized users to see how different items are selling and the progress of the inventory over a given period of time.

The system shall allow users to receive notifications whenever a product quantity falls below 10 in inventory.

Data Management, performance and security- In this component, the non-functional requirements will be mainly used. The system should only allow authorized personnel to add and edit with 10 keyboard word/phrases entries. Authorized persons will also be able to send out notification to customers within 1 keyboard entry.

Upon accessing the system, all screens should load within 5 seconds. Users should be able to log into the system within 2 keyboard words/phrases entries. All personal information that is being entered by the user will be encrypted and stored in a secure storage location. These will be protected by the username and password, that is uniquely set by the users. Moreover, whenever the users are entering their password, each individual character will be blocked on the screen just to protect the user's passwords.

Additionally users should be able to view products within 5 keyboard word/phrases entries.Upon creating an order, users will receive a unique order number. After a customer placed an order, the order details should reflect in the system within 10 seconds. Orders that are sent in by customers will be stored in a secure location that can only be accessed by authorized users.

2.3.2 Architecture Justification

The non-functional requirements of the intended system informed our decision on which architectural pattern to select for our solution. In particular, we heavily weighted the non-functional requirements related to security, dependability and a particular aspect of maintainability and software evolution which will be discussed. Additionally, we considered heavily the potential development strategies to be employed by the development team, and this affected our choice of architecture further.

It has been established (within 2.1) that the intended solution prioritizes multi-level security. Each user is to have access to a set of processes and views based on their level of authorization. Therefore, as the user progresses from one set of views or processes to another, or attempts to do so, the system is to ensure that the user is authorized to access this element of the system. This speaks to a need for multi-level security, which is a strong advantage of designing a system using layered architecture.

Acting in the best interest of the client involves creating a solution that is easiest to be maintained and evolved as the needs of the client change. This speaks to value placement on the ease with which each component or element of the system can be altered or replaced with minimal to non-existent disruption to the proper functioning of the overall system. With layered architecture, as long as the interfaces between each layer remain the same, the layers can be altered or replaced entirely without disrupting the ability of requests to be transferred between according to the intentions of the developers. Once each layer is altered or replaced, it will process each request as desired. In this way, our client may incorporate transformations to features or new ways of handling a request at each process within the layer, while maintaining the overall structure of the system.

Our client is a kiosk that has been in operation for over 15 years. It is a successful small business with a successful business model in the eyes of the business owner and customers, with smooth operations and a good reputation. Any system that is incorporated into the normal flow of business must therefore add value with minimal risk to how the business operates. For this reason, the client heavily values the dependability of the system to be implemented. Utilizing an architectural pattern that boasts dependability as an advantage is therefore of importance for this system. Within layered architecture, the redundancies in place throughout the layers increases system dependability and the confidence that the system will operate as expected.

Last, we considered the factor of how the system will be developed. Layered architecture is suitable for use in a system when the system is intended to be developed by a distributed team, and when each division of the team has responsibility over a particular layer's functionality. This matches well with the expected development strategy to be utilized for the development of our system, where the responsibility of implementing the aspects of the system will be divided amongst group members, with the ultimate expectation that these developed components, once fit together, will work seamlessly. The interfaces between these components if developed in this way would need to be planned beforehand so as to allow seamless integration upon development.

The above factors played an important role in leading to the selection of a layered architecture pattern for our system. We determined that its advantages outweighed, for the purpose of our system, its disadvantages. One such disadvantage would be limitations to performance due to each request potentially needing to be processed by multiple layers, and the impracticability of expecting truly clean separation of layers in practice. We concluded that the layered architecture pattern's advantages pointed in the direction of choosing this pattern, particularly the layered pattern's increased dependability and modularity or comparative ease with which each layer can be replaced, along with the system's need for multi-level security and the development strategy to be employed by the development team.

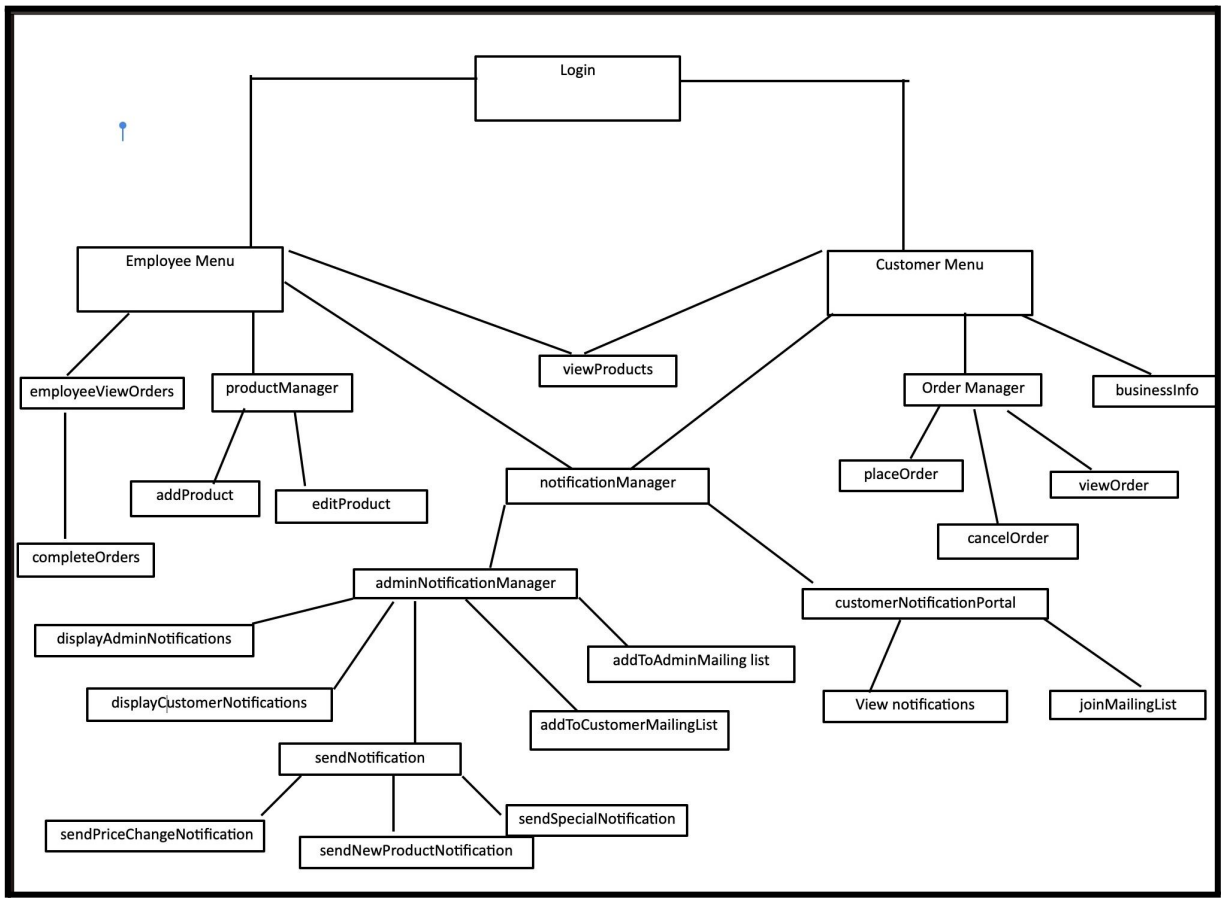
3.0 Architecture Decomposition

3.1 Component Decomposition – Modules

Requirement ID	Architecture Component	Class / <Module> Name	Description
Req 1 Req 2 Req 3 Req 4 Req 5 Req 6 Req 7	User Interface	employeeMenu	This will provide a text based layout of all the different options that an employee can gain access to.
		customerMenu	This will provide a text based layout of all the different options that a customer can gain access to.
		viewBusinessInformation	This will provide an area which outlines some of the information regarding the business.
		orderManager	This will provide a text based layout of all the different options that a customer has to place orders.
		notificationManager	This will provide a text based layout of all the different options for notifications.
Req 1.1 Req 2.2 Req 6.2 Req 6.3 Req 7.3 Safety and security	Authorization and Authentication	login	This will provide a user with options to sign into the system as a customer or an employee and their authorization to access the system as an employee is provided through a unique username and password.
Req 1.1 Req 1.2 Req 1.3 Req 1.5 Req 4 Req 6	Product list management	productManager	Provides a list of options pertains to products
		addProduct	Allows user to enter new product and its details
		editProduct	Allows user to select product and edit its details

		Inventory_Manager View_Products	Allows user to choose from list of functionalities pertaining to inventory This will provide a list of the products offered for customers to select the desired product.
Req 1.4 Req 2 Req 7	Notification management	notificationManager sendNotification joinMailingList viewNotifications	Provides a list of functions pertaining to notifications Allows user to create new notification to be sent This provides an area where users will be able to either enable join the mailing list for notifications. This provides an area where customers can view their notifications if they had enabled them.
Req 3 Req 5 Req 6	Order placements	placeOrder cancelOrder viewOrder	Provides an area for users to place their order Provides an area for users to cancel the orders they would have placed and not received as yet. Provides an area for users to view the order(s) that they made and had not received as yet.
Safety and security Performance Quality	Data management, performance and security		

3.2 Structural Design – Structure Chart



3.2.1 Design Notes

The diagram follows a top down approach as it starts at the login page and works its way down. Users can either sign in as an employee or a customer. If they sign in as an employee they are sent to a menu which gives the employee different options. The first option relates to customers orders, as they can view the orders and set them as complete after they have done them. Employees can also manage the products in their inventory by either adding a product or editing an existing product. They can also manage notifications by either sending out a notification or editing an existing one. They can also manage the inventory by viewing it, updating it or generating a detailed report on it. Finally they can view all their products in their inventory. This is a module that is also shared by customers when they gain access to it after signing in. After signing in, customers also have options for orders such as placing an order, canceling an order or editing an existing one that they would have placed but not received as yet. Customers can also view the business's information and finally they have options for notifications as they can view them or enable or disable the receipt of them.

Test Plan

Test No.	Test Name	Test Case	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.1	Login	Test with incorrect admin username and correct admin password	Username:drichards Password:password	Access Denied	Access Denied	Pass
1.2	Login	Test with incorrect admin username and incorrect admin password	Username: admin17 Password: KKSadmin #13	Access Denied	Access Denied	Pass
1.3	Login	Test with correct admin username and correct admin password	Username: admin1 Password: KKSadmin #1	Access Granted	Access Granted	Pass

2.1	addProduct	Test with incorrect data type for Product name and correct data type for all other fields	Product name: 23 Description : Sprite Bottle Soda Price: $150 Quantity: 12	Error message	Error Message	Pass
2.2	addProduct	Test with correct data type for Product name and correct data type for all other fields	Product name: Sprite Description : Sprite Bottle Soda Price: $150 Quantity: 12	Product successfully added to product list	Product was added successful ly	Pass
2.3	addProduct	Test with no submission for description and correct data type for all other fields	Product name: Sprite Description : Price: $150	Product successfully added to product list	Product successful ly added to product list	Pass

			Quantity: 12			
2.4	addProduct	Test with incorrect data type for Price and correct data type for all other fields	Product name: Sprite Description : Bottles Sprite Price: big Quantity: 12	Error message	Error Message	Pass
2.5	addProduct	Test with incorrect data type for the quantity and correct data type for all other fields	Product name: Sprite Description : Bottled Sprite Price: 150 Quantity: Twelve	Error Message	Error Message	Pass
3.1	businessInfo	This will provide an area which outlines some of the information regarding the business	username:zBoss123 password:password	The system will display the business information to users, which entails	The system displays the business	Pass

				the products offered, where the business is located and also how long the business has been in operation.	informatio n to user	
4.1	cancelOrder	Test with the incorrect order number	Username: kellyann345 Password: Twenty452 Order number: 35017	Error Message	Error Message	Pass
4.2	cancelOrder	Test with the correct order number	Username: kellyann345 Password: Twenty452 Order number: 35014	The system should automatically cancel the order and display a message that the order has been canceled.	The confirmati on message popped up to advise that the order was canceled.	Pass
5.1	createAcco unt	Test with the incorrect amount for the phone number along with	First name: Kelly Last name: Plesa	Error Message	Error Message	Pass

		the correct data for other fields.	Phone number: 8765432548 Username: 9995424 Password: @&$,:;			
5.2	createAccount	Testing with the incorrect data type for the user name	First name: Kelly Last name: Plesa Phone number: 8765432548 Username: 9995424 Password: @&$,:;	All the user to proceed to create their unique password	Allows the user to successfully create a password	Pass
5.3	createAccount	Testing with the incorrect password for confirmation	First name: Kelly Last name: Plesa Phone number: 8765432548	Error Message	Error Message	Pass

			Username: 9995424 Password: 274,:;			
5.4	createAccount	Testing with the correct information for all fields	First name: Kelly Last name: Plesa Phone number: 8765432548 Username: 9995424 Password: @&$,:;	The system will allow new users to create an account successfully	User was able to create an account successfully	Pass
6	completeOrder	Testing to what would happen if owner completes a order with the incorrect order number				
7	CustomerMenu	This will provide a text based layout of all different options that a customer can gain access to	username:zBoss123 password:password	The system will display the main menu to all customers, who will then select their choice from	The system displays the customer menu once the user is	Pass

				the main menu.	successfully log in	
8.1	editProduct	Test with the incorrect data type for price and correct data type for the other fields	Product:Soda Description: Pepsi Bottled Soda Price: One hundred Quantity: 10	Error Message	Error Message	Pass
8.2	editProduct	Test with the incorrect data type for quantity and correct data type for the other fields	Product: Coco bread Description: Fresh coco bread from Tastes Patty Price: $100 Quantity: Ten	Error Message	Error Message	Pass

8.3	editProduct	Test with the changing the product name with the wrong data type	Product: 47 Description :Fresh coco bread from Tastes Patty Price: $100 Quantity: 10	The system will change the product name	System successfully changed the product name	Pass
8.4	editProduct	Test without writing a description	Product: Soda Description : Price: $100 Quantity: 10	The system will change the description	System changed description successfully	Pass
8.5	editProduct	Test with the correct data type for price and quantity	Product: Soda Description :Coca Cola Bottled Soda Price: $200 Quantity: 30	The system will change the price and quantity for the product.	System successfully changed the quantity and price	Pass
9	employeeMenu	This will provide a text based of all the different		The system will display a menu for		

		options that an employee can gain access to		the employees where they should be able to view orders, products, update product details, update inventory and send notification.		
10	employeeViewOrder	Testing to see all the orders that users has made	Username: admin1 Password: KKSadmin #1	The system will display all customers' orders in sequential order.	The system displays all the customer's orders in sequential order	Pass
11	notification Manager	This provides a list of functions pertaining to notifications		The system will send out notification to customers to inform them on price change, new product and other notification from the owner.		

12.1	placeOrder	Testing with the incorrect data type for quantity.	Index: 11 Product: Earbuds Quantity: two Location: Law	An error message will display .	Error Message	Pass
12.2	placeOrder	Testing with the correct data type for quantity.	Index: 11 Product: Earbuds Quantity: 2 Location: Law	Order will be successfully placed	Order was placed successfully	Pass
12.3	placeOrder	Testing to try and select two index at the same time to make two purchases	Index: 11,13 Product: Earbuds Quantity: 2 Location: Law	Error Message	Error Message	Pass
12.4	placeOrder	Testing with the incorrect data type for index.	Index: Eleven Product: Earbuds Quantity: 2	Error Message	Error Message	Pass

			Location: Law			
13	orderManager	This will provide a text based layout of all different options that a customer has to place orders		The system will allow the user		
14.1	viewOrder	Test to see what would happen if a new user select view order even though they haven't made any purchase	Username:faith20 Password:1234567	The system will advised user that user that there's no current order	System displays error message and advised user that there's no current order	Pass
14.2	viewOrder	Test to see what would happen if an existing user select view order even though they haven't made any purchase	Username: kellyann345 Password: Twenty452	The system will advised user that user that there's no current order	System displays error message and advised user that there's no current order	Pass
14.3	viewOrder	Test to see what would happen if an existing user select view order even though they	Username: zBoss123 Password: password	They system will display all purchase that user has made along	The system displays all the orders made by the user	Pass

		have made a purchase		with their final total	along with the final total.	
15	viewProduct	Testing to see what will display once the user select view product	Username:faith20 Password: 1234567	The system will allow the customers to view the products that are available. The system will also display the product details, such as product name,description, price and availability.	The system displays all the available product, along with the product details	Pass
16	salesReport	The system generate a comprehensive analysis of the inventory over a selected period		The system will generate a report which the admin will be able to view. This will consist of the		

17	Update Inventory	The owner will be able to update the quantity of product that is available to customers, who will have the ability to see all products that are available and the quantity.		The system will update the inventory to reflect if there is any in stock or if the product is out of stock. The system will also show the updated quantity of the product that is currently in stock or out of stock.		
18	Stock Alerts	The owner will be able to update the product availability and customers will be notified if a product is out of stock or running low.		The system will also send out notification to customers to advise them on the updated inventory, whether they are running low on stock, out of stock or in stock.		

19.1	customerMailingList	Testing to see what would happen if a customer requests for notification more than once.	Username: zBoss123 Password: password	System will display an error message, and advise that they are already a part of the mailing list.	Error Message was displayed to users to advise them that they are already apart of the mailing list.	Pass
19.2	customerMailingList	Testing to see what would happen if a customer requested notification for the first time.	Username:faith20 Password: 1234567	System will advise user that they are successfully added	System displays message to user and advise that they are successfully added.	Pass
20.1	customerNotification	Testing to see what would happen if a existing user selected notification	Username: zBoss123 Password: password	The system will display all the notification to the user	System displays all notification to user	Pass
20.2	customerNotification	Testing to see what would happen if a new user selected notification	Username:faith20 Password: 1234567	The system will display all the notification to the user	System displays all notification to user	Pass

Code Link

The code for this project can be accessed on GitHub using this link:
https://github.com/zariacs/swen2140-final-project.git

Appendix

On October 28, 2022, an interview was conducted with Chrissy Toyloy, the owner of Krissy's Kiosk, on the requirements of the Software for her small business. Below were the interview questions that we asked Chrissy, to assist with creating the necessary requirements for the Software.

1. How often would you want to update inventory? Daily? Every 2 Weeks?
   Response: Daily
2. What kind of information would you want to know about your inventory? When something's running low? What's selling the best? The slowest?
   Response: Respondee answered yes to all conditions listed
3. Would you want customers to be able to contact you via text messages or emails?
   Response: Text Messages
4. Would you want to be able to send a broadcast to customers? Eg. We will be open on this public holiday? Closed early on this day? Reminder that our hours are such and such
   Response: yes

This questionnaire was used to determine the initial context and need for the project. Interview was done on September 16th with Krissy the owner of the shop.

Questionnaire

1. What is the name of your business?
   Response: Krissy's Kiosk

2. How long has it been open?
   0-2 years 3-5 years over 5 years
   Response: over 5 years

3. What would you say is the biggest inconvenience with your current way or operation?
   Response: customers are limited because of where on campus the shop is

4. What would you like this software to address?
   Response: I would like to find a way to expand the customers. I would also like to be able to make inventory easier to handle.

5. Do you want a login area for customers and employees?
   Yes No
   Response: Yes

6. Do you want users to be able to view all products and their details?

Yes No
Response: Yes

7. Do you want users to be able to choose if they would like to receive product related notifications?
   Yes No
   Response: Yes

8. Do you want users to be able to place orders using the system?
   Yes No
   Response: Yes

9. Do you want users to be able to pay through the system?
   Yes No
   Response: No