University of Calgary

Schulich School of Engineering

Department of Electrical and Computer Engineering

Course: ENSF 619.30

Software Requirements and Pro cess Management

Software Requirements Specification Document for:

Grocery Self-Checkout System

Team Members:

Toya Okeke, David Haehlen, Alexander Mantey, Craig Martis

Submitted in partial fulfilment of the requirements of

ENSF 619.30 – Fall 2018

Table of Contents

List of Figures

List of Tables

Glossary

TERM	DESCRIPTION
DOMAIN / SCOPE	The area considered within the study
USE CASE	A sub-process part of activity in a system
SEQUENCE DIAGRAM	A diagram that shows the sequences of actions in a use case
DOMAIN MODEL	A static UML model that shows the major domain concepts/objects that play an important role in a system
DOMAIN CLASS DIAGRAM	
GUI	Graphical User Interface
POS	Point of Sale – device for processing credit and debit payments

1 Introduction

1.1 Purpose

This document describes, in detail, a new grocery self-checkout system. The system allows customers to scan, enter, or search for items they wish to buy and then pay by cash or card once they have completed their order. Customers have the option of editing their purchase before making their payment but will need a manager to edit the order for them.

1.2 Overview of the Document

1.2.1 Document Conventions

In this document one can assume some entities inherit the behaviours of their parent entity in a hierarchical tree. For example, an admin inherits the behaviour of a manager and has its own additional features.

1.2.2 Readers and Audiences

This document provides an overview of the new grocery self-checkout system and the services it will provide. Therefore, this document is intended mainly for marketing staff, developers, management and any other users that wish to deploy the system. The functionality of the system will differ based on the user. For example, a customer will not be able to implement all the functions of this system that a manager or admin can. How each party uses this system will be discussed further. Furthermore, the operating environment the system requires will be explained. This includes hardware requirements, user and communication interfaces, system constraints and other software needs.

The system has many features to maintain information and handle various scenarios, such as editing an order before paying. These features will be listed and broken down for developers, employees and management to understand the system requirements and flow of data. The relationships between different classes and their information will also be provided. Non-functional requirements are listed to aid developers in their design decisions during the development phase.

1.3 Product Scope

The self-checkout system will give customers the ability to purchase items in-store by scanning the item or entering the item code. The system must also update the inventory when an item is purchased.

This system is not responsible for updating terms of service or any other contractual relations. Additionally, the system will not order more grocery items if the inventory is running low. The scope of the project does not include the front end; however, this document explains some of the detail of how the user will use the interface to interact with the system.

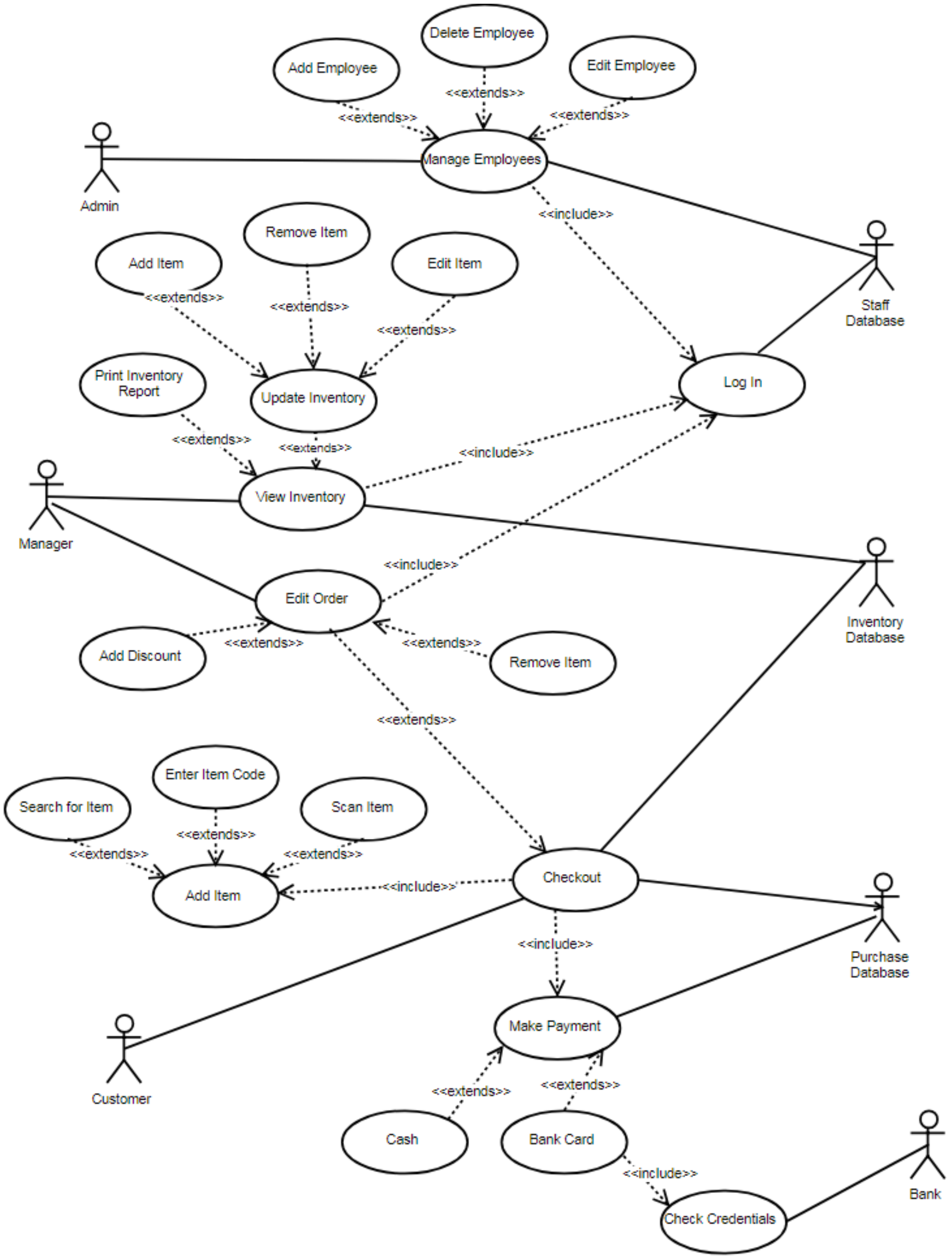


Figure 1 Use-Case Diagram of the Grocery Self-Checkout System

2 Overall Description of the System

2.1 Product Perspective

The grocery self-checkout system is a new system and not derived from an existing system. However, it is inspired by existing self-checkout systems and seeks to provide an alternative product. The system relates only to the software system and the environment it directly operates in. The companies that wish to implement this system are responsible for adding their initial stock, prices, along with ordering more items if their stock is running low.

2.2 Product Features

2.2.1 Major Features

The following features are essential to the system operating correctly:

- The system must allow managers to update the store inventory
- The system must allow administrators to manage employees in the store
- The system must retrieve the price of an item and add it to the customers order total
- The system must direct the customer to the payment method where they will either pay cash or insert their bank card and PIN number. The system will also print out the receipt of the transaction once the payment has been processed.

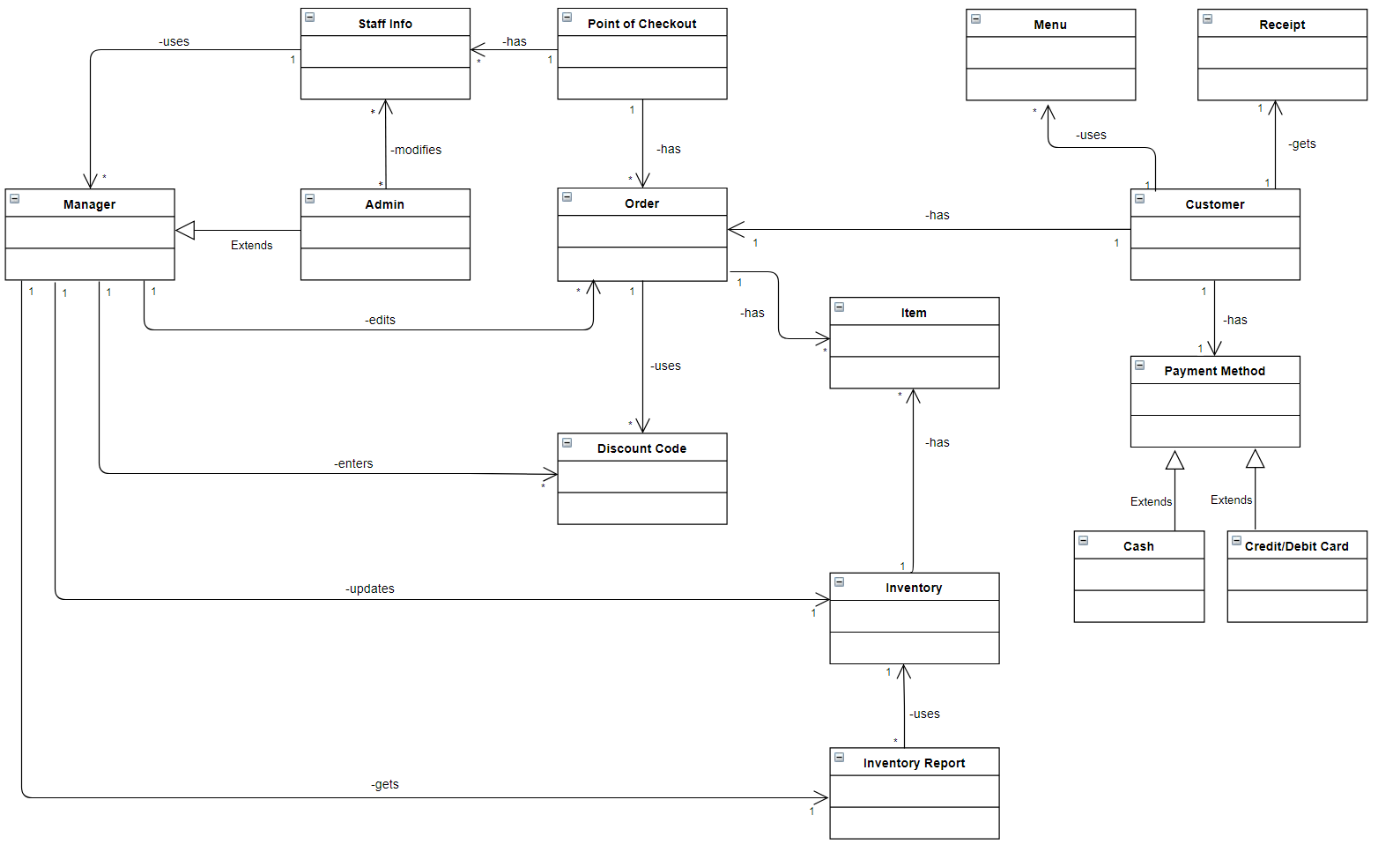


Figure 2 Domain Class Diagram of Grocery Self-Checkout System

2.3 User Classes and Characteristics

The main user classes that will use this product are the managers, the administrators, and the customers. What differentiates these classes are the functions that they use in the system. The managers view and update the inventory and edit customer orders. The administrators update staff information in the system database. The customers use the self-checkout system to add items to their order, pay for their order, and to receive a receipt after their purchase.

The most important user class to satisfy is the customer, however it is essential to the system functionality that the requirements of the manager is satisfied. The main system features can still function without satisfying the requirements of the administrator.

The system is designed to be very simple to operate for customers. Its user interface is simple and not intimidating for people that are not good with technology. The system is also easy to operate for managers and administrators. Managing databases does not need to be done with SQL operations, and a simple GUI is designed for their use cases.

2.4 Operating Environment

This is a private system that is not accessible online. Administrators and managers can only access the system from computers in the store. The databases will be hosted on-site to ensure that the system can continue to operate in the event of an internet outage, or a failure of a cloud database. The database will be backed up on a separate device, in case of a failure of the primary device. The database can also be updated by tills with manual checkouts.

2.5 System Constraints

The current system does not specify the number of items a customer can purchase in one order. In the event a customer forgets to take purchased item with them and does not return to collect it, the system will still update the inventory database. It will be up to the store manager to ensure the number of items in the store inventory is correct. Finally, the system has no way preventing a customer from leaving the store without scanning an item. Again, it will be up to store managers to ensure customers scan or enter all items they wish to purchase before entering the store.

2.6 User Documentation

An instruction manual on how to login and use the self-checkout system is available for mangers and system administrators. No documentation is available for customers. Should they have any question about how to use the system, the store employees must help using the information provided to them in the instruction manual.

2.7 Assumptions and Dependencies

Although a customer can purchase items, the company that implements this system will set a maximum item order limit for customers that wish to use the self-checkout system. This will prevent any long lineups for those who wish to purchase only a few items. However, customers will still have the freedom to purchase however many items they wish, so long as it is within the order item limit.

Given the system is a private platform, customers cannot make returns or request refunds from home. All returns and refund requests must be made in-store. The system also assumes there are only two acceptable methods of payment: cash or bank card. Therefore, other payments like cheque or store gift card cannot be done through the existing self-checkout system.

3 User Classes and Characteristics

Managers that will be using this system are responsible for editing a customer's order and managing items in a customer's order. Any changes to item quantity must be done by the employees. Before they can make changes to the inventory, they must authenticate themselves by submitting their login information and receiving a successful login confirmation.

System administrators can do everything a manager can. Additionally, they can add, remove or edit an employee's information.

4 System Use Cases

4.1 Use Case 1 (UC-1): Manage Employees

4.1.1 Description of Use Case

This use case gives system administrators the ability to make changes to the employees in the staff database. The sequence diagram in Figure 3 gives a visual representation of what happens in this use case. First, the administrator must log into the system. Once their credentials are verified by the system, it will display all the staff information to the administrator. From there, they can either add, remove or edit staff information and log out of the system once they are finished.

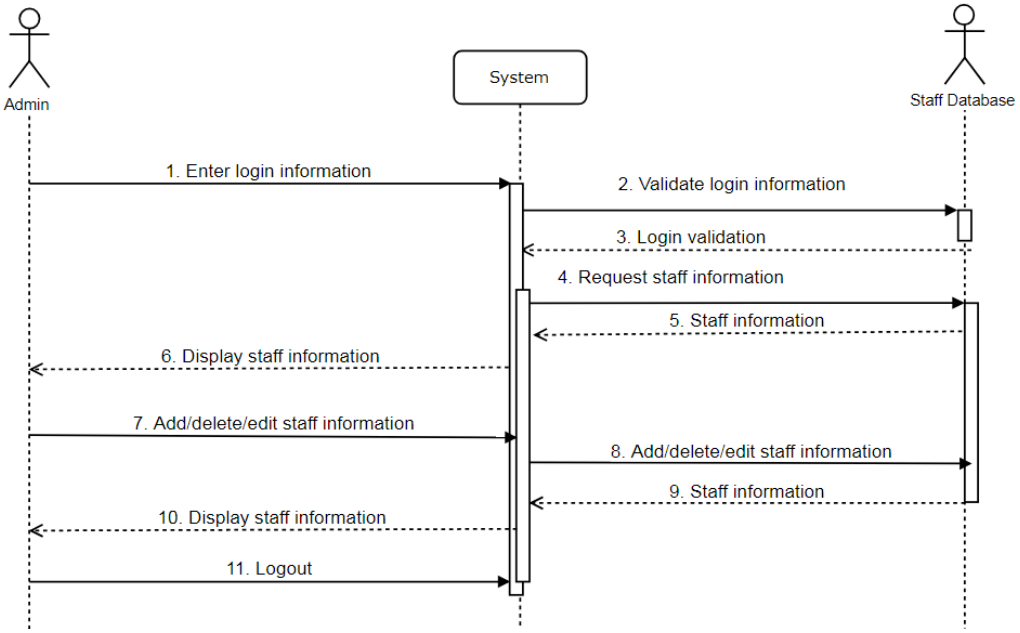

Admin
System
Staff Database
1. Enter login information
2. Validate login information
3. Login validation
4. Request staff information
5. Staff information
6. Display staff information
7. Add/delete/edit staff information
8. Add/delete/edit staff information
9. Staff information
10. Display staff information
11. Logout

Figure 3 UC-1: Manage Employees

4.1.2 Pre-Conditions and Post-Conditions

The system must be connected to the database prior to login or make that connection during login. For the administrator to manage employees, they must first log into the system. Once the system has confirmed their credentials, they can make changes to staff. Once they are done making changes, they must log out of the system.

4.1.3 Possible Risks

4.2 Use Case 2 (UC-2): View Inventory

4.2.1 Description of Use Case

This use case allows managers to view the store inventory. First, they must log into the system. Once the system verifies their credentials, it will display all information about the store inventory to the manager. The manager can now do two things: update the store inventory (version 1) or print an inventory report (version 2).

UC-2: View Inventory (Version 1)

If a manager wishes to update the store inventory when they login, they can choose to either add, remove or edit an item. The system will make these changes in the inventory database and display the updated information to the manager. The manager then logs out of the system once they are finished. The sequence diagram for this version of View Inventory can be seen in Figure 4.

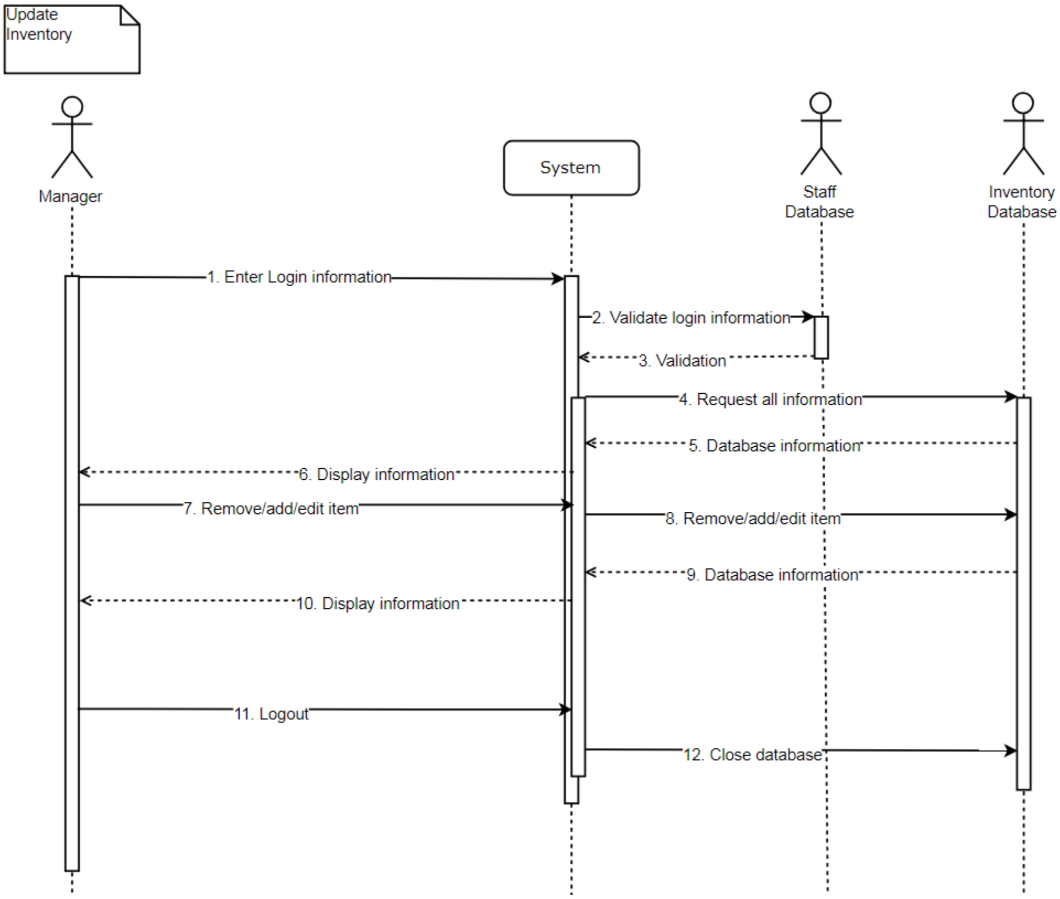


Figure 4 UC-2: View Inventory (Version 1)

UC-2: View Inventory (Version 2)

A manager can also print an inventory report after they have logged into the system. First, they request an inventory report from the system. When the system receives their request, it gets the analytics of the items in the store's inventory and prints the report. Once they are finished, the manager logs out of the system. The sequence diagram for version 2 of View Inventory is shown in Figure 5.

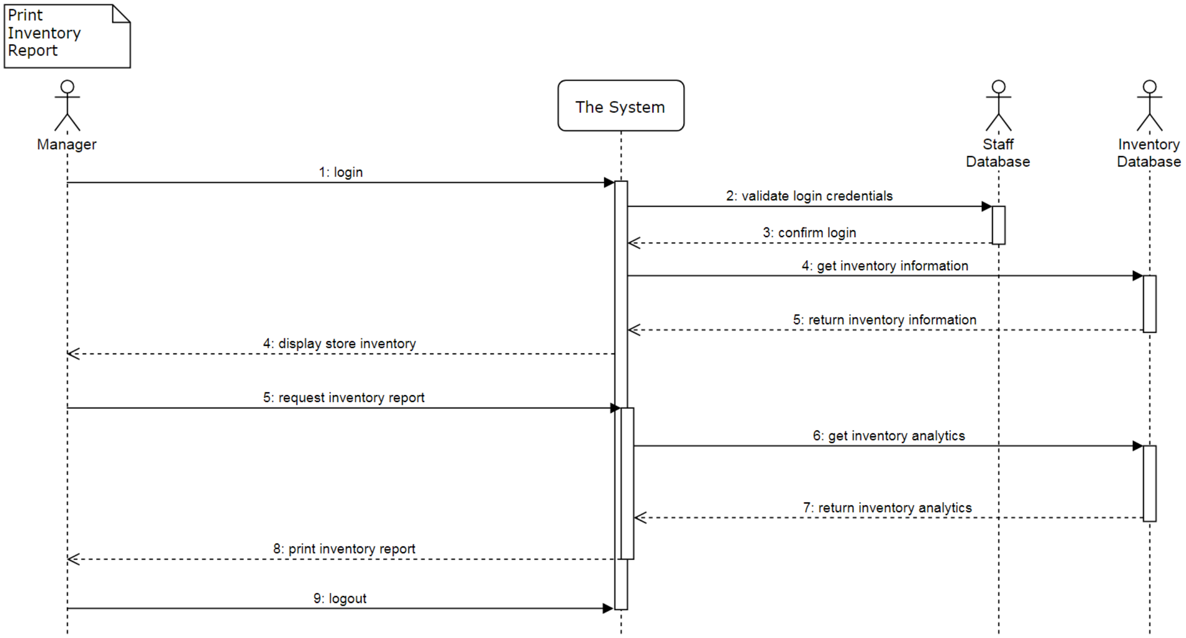

Print Inventory Report
Manager
The System
Staff Database
Inventory Database
1: login
2: validate login credentials
3: confirm login
4: get inventory information
5: return inventory information
4: display store inventory
5: request inventory report
6: get inventory analytics
7: return inventory analytics
8: print inventory report
9: logout

Figure 5 UC-2: View Inventory (Version 2)

4.2.2 Pre-Conditions and Post-Conditions

The manager must log into the system if they wish to view the store inventory. Once the system has confirmed their credentials, they can make changes to the inventory or print an inventory report. After they are finished, they must log out of the system.

4.3 Use Case 3 (UC-3): Edit Order

4.3.1 Description of Use Case

If a customer would like to make changes to their order, they must get a manager to make the changes for them. The manager first logs into the system. Once the system verifies the manager’s credentials, it will display the customer’s order. From here, the manager can either remove an item from the customer’s order (version 1) or apply a discount to the customer’s order (version 2).

UC-3: Edit Order (Version 1)

If the customer wants to remove an item from their order, the manager will remove that item from their order. The system will reduce the total purchase amount in the order and display the updated customer order. Afterwards, the manager must log out of the system. The sequence diagram for version 1 of Edit Order can be found in Figure 6.

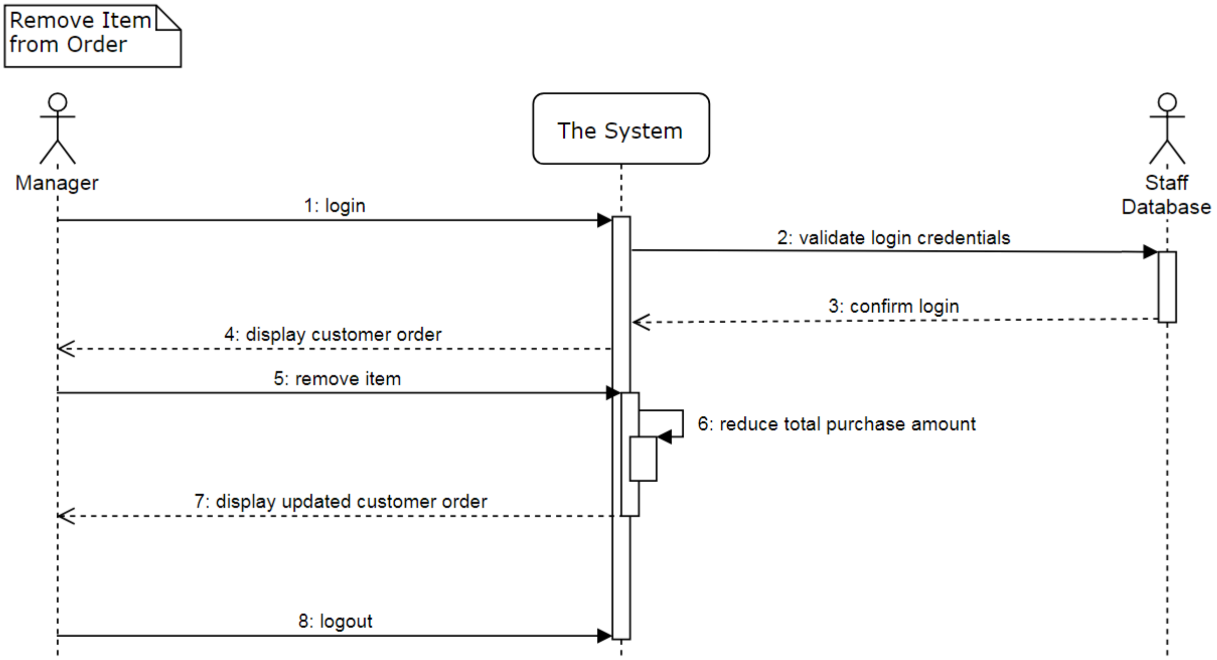

Remove Item from Order
Manager
The System
Staff Database
1: login
2: validate login credentials
3: confirm login
4: display customer order
5: remove item
6: reduce total purchase amount
7: display updated customer order
8: logout

Figure 6 UC-3: Edit Order (Version 1)

UC-3: Edit Order (Version 2)

If the customer wants to apply a discount to their order, the manager must enter or scan the discount code provided by the customer into the system. The system will validate the discount and apply the discount to the customer's order. Once the discount is applied, the system will display the customer's updated order. Finally, the manager must log out of the system. The sequence diagram for version 2 of Edit Order is shown in Figure 7.

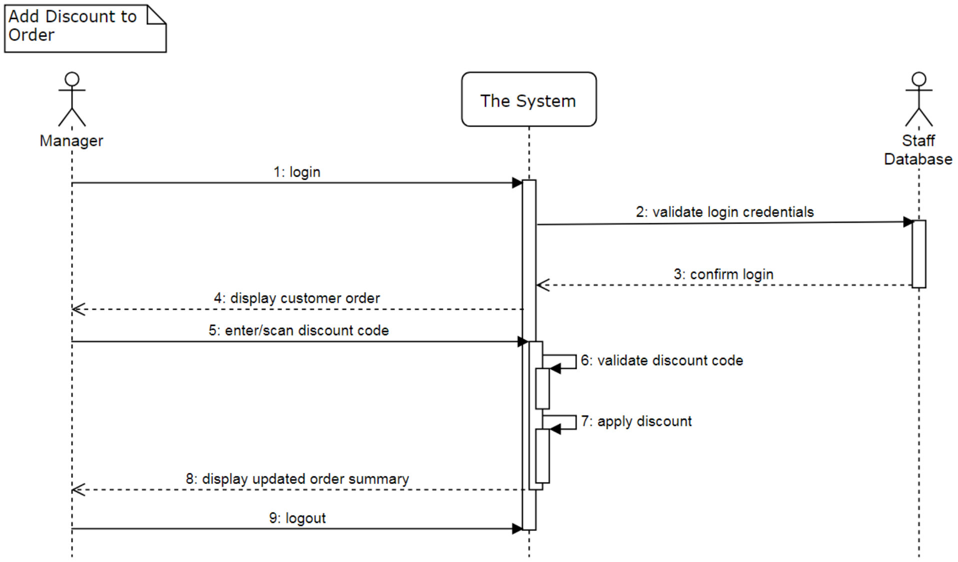

Add Discount to Order
Manager
The System
Staff Database
1: login
2: validate login credentials
3: confirm login
4: display customer order
5: enter/scan discount code
6: validate discount code
7: apply discount
8: display updated order summary
9: logout

Figure 7 UC-3: Edit Order (Version 2)

4.3.2 Pre-Conditions and Post-Conditions

If a customer wants to edit their order, a manager must log into the system to edit it for them. Once the manager has logged into the system and edited the customer's order, they must log out of the system.

4.3.3 Pre-Conditions and Post-Conditions

4.4 Use Case 4 (UC-4): Checkout

4.4.1 Description of Use Case

When a customer is ready to checkout their items, they will start a new order in the system. Here they will enter items for their order by either scanning the item, entering the item code or searching for the item in the system. The system will get the item information from the inventory database, add the item to their order and display the updated order to the customer. Once the customer is finished adding their items, they proceed to making their payment. From there the system will add the customer's purchase to the purchase database and print their receipt. After printing the receipt, the system will update the store inventory and close the order. The sequence diagram for this use case can be found in Figure 8.

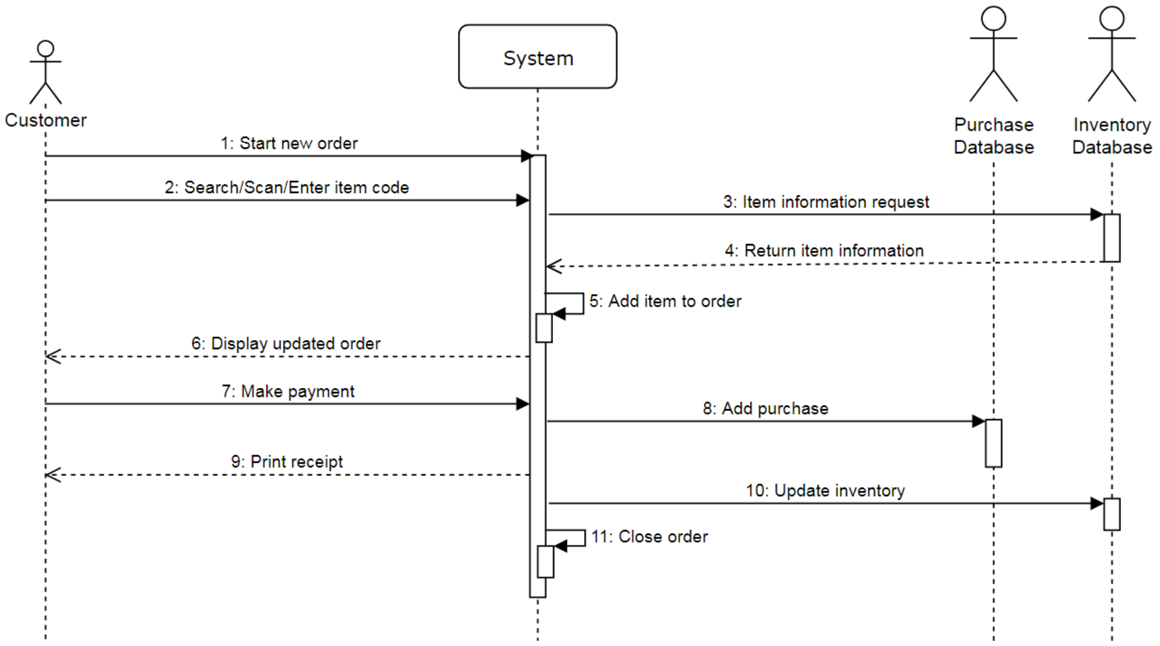


Figure 8 UC-4: Checkout

4.4.2 Pre-Conditions and Post-Conditions

If a customer wants to purchase items in the store using the self-checkout system, they must add their items to a temporary order in the system. Once they have finished adding items to their order, they must make a payment for their order. How the customer can make this payment will be described next

4.5 Use Case 5 (UC-5): Make Payment

4.5.1 Description of Use Case

When a customer wishes to checkout their items using the self-checkout system, they must complete their order by making a payment. First the customer must select their payment method. The system has two acceptable methods: cash (version1) or bank card (version 2).

UC-5: Make Payment (Version 1)

If the customer selects cash as their payment method, they must input their cash into the system. The system will give the customer their change and a receipt of their order, then add their purchase to the purchase database. The sequence diagram for this version of Make payment can be found in Figure 9.

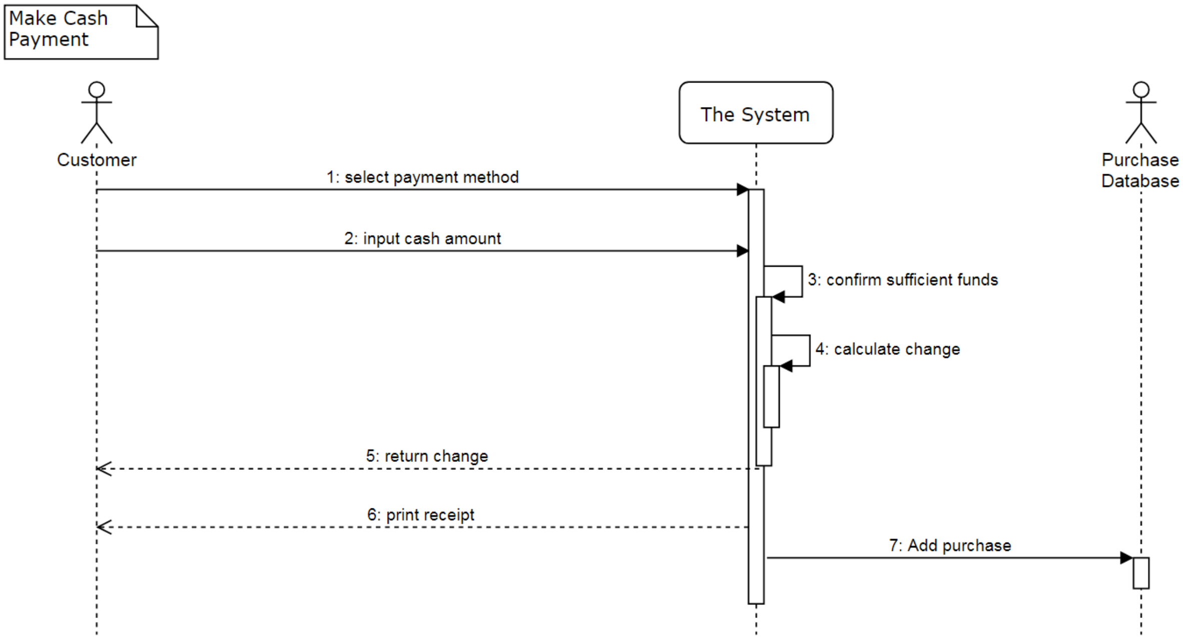


Figure 9 UC-5: Make Payment (Version 1)

UC-5: Make Payment (Version 2)

If the customer chooses to pay by bank card, they must first enter their card, and a PIN code. The system will then validate the card information with the bank and the bank will notify the system if the transaction is successful. After, the system will print a receipt of the order for the customer and add their purchase to the purchase database. The sequence diagram for version 2 of Make Payment is shown in Figure 10.

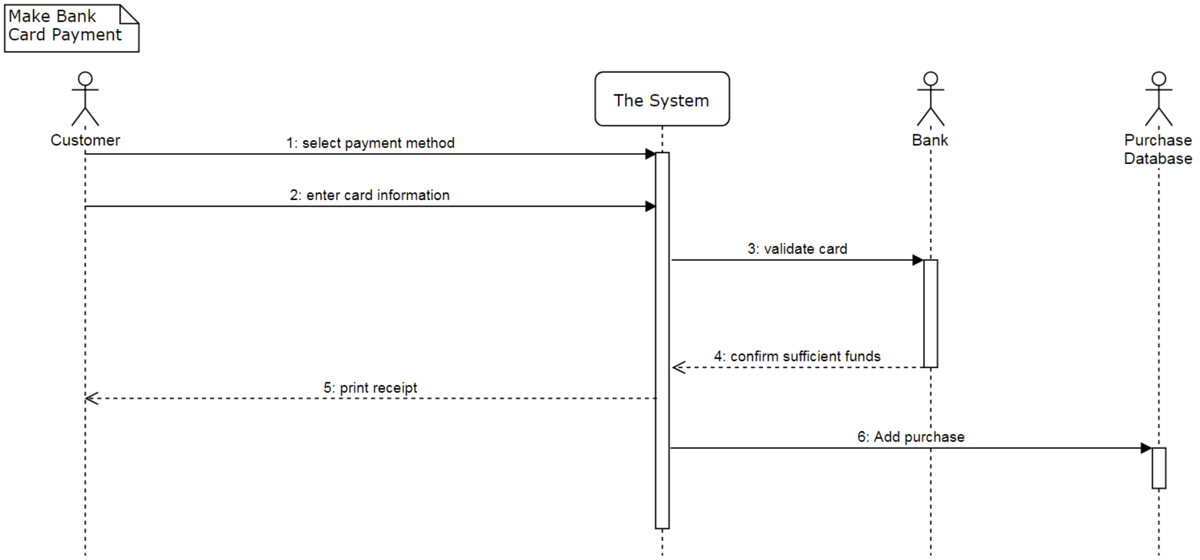


Figure 10 UC-5: Make Payment (Version 2)

4.5.2 Pre-Conditions and Post-Conditions

If the customer would like to pay for their order, they must select a payment method that the system can accept. Depending on the payment method, either the system or bank must confirm if the customer has sufficient funds before the order is complete.

5 Summary of Functional Requirements

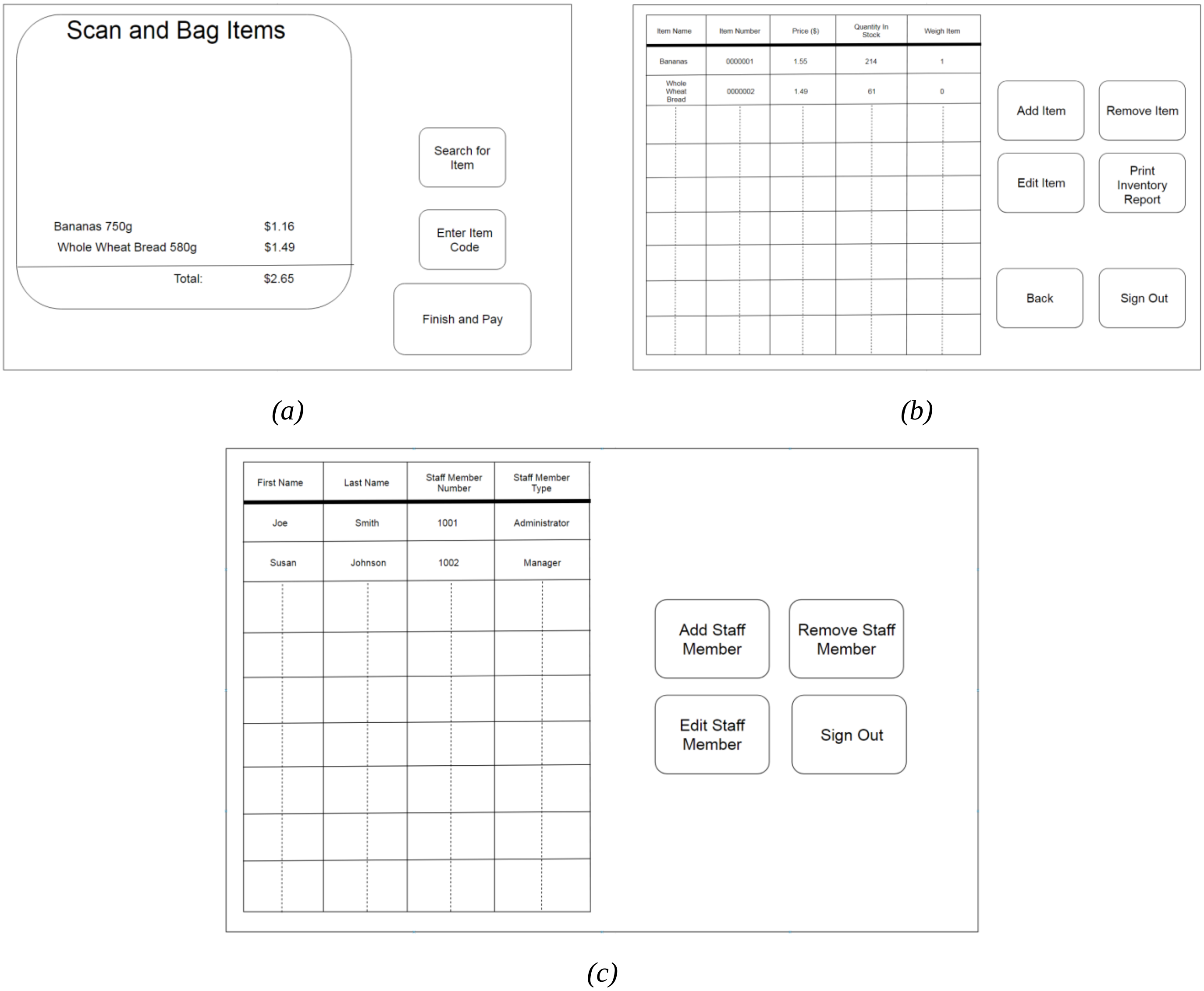
RE#	DESCRIPTION
RE1	Customer should be able to scan item to checkout cart
RE2	Customer should be able to search for items to add to cart
RE3	Customer should be able to enter item by product code
RE4	Customer should be able to pay with cash, debit and credit
RE5	Customer should receive a receipt after transaction completed
RE6	Inventory should be updated after a successful transaction is completed
RE7	Managers should be able to update the inventory, add and remove items
RE8	Admins should be able to update staff records, add and remove staff records
RE9	Managers should be able to view inventory and print reports
RE10	System should control access via login-ins
RE11	Managers should be able to edit orders and apply discount codes

6 External Interface Requirements

6.1 User Interfaces

User interfaces for the system are all screen based graphical interfaces. The customer interface is touch screen allowing interaction with the on-screen buttons and will display the items that have been scanned so far. The manager and admin GUI are run on typical desktops and are not intended to be touch screen based, interactions will be facilitated with the typical mouse and keyboard peripherals. Below are some possible prototype graphical interfaces for each class of user. For a customer to pay, there will be a typical POS device, or in the case of cash, slots for bills and coins. A printer will be located at each checkout terminal which will print the receipt for the user.

6.1.1 Prototype Graphic or Screenshot



Scan and Bag Items

Bananas 750g	$1.16
Whole Wheat Bread 580g	$1.49
Total:	$2.65

(a)

Item Name	Item Number	Price ($)	Quantity In Stock	Weigh Item
Bananas	0000001	1.55	214	1
Whole Wheat Bread	0000002	1.49	61	0

(b)

First Name	Last Name	Staff Member Number	Staff Member Type
Joe	Smith	1001	Administrator
Susan	Johnson	1002	Manager

(c)

Figure 11 The (a) customer, (b) manager and (c) administrator GUIs

6.2 Hardware Interfaces

The system can be accessed using a specialized self checkout machine, which features a touch screen display, a barcode scanner, a payment terminal for credit and debit cards, slots to insert cash, a scale to weigh items, and a printer to print receipts. These components allow customers to add items to their order and provide a payment method.

Managers and admins are able to login to the system on a computer to view and update inventory, and print inventory reports. Admins are able to view and update staff information from computers.

Whether the system is being accessed from the self checkout machine or from a computer, the system communicates to databases with the inventory information and staff information.

6.3 Software Interfaces

The computers in the store and the self checkout machines connect to the inventory and staff information databases. The computers will run Windows because they will be used for other business applications as well. The self checkout machine will run Linux because it is free and because of its customisability.

The system needs to be able to connect with a credit card or bank network through the internet to process the transaction and receive verification that the transaction was successful.

The machines and computers will receive inventory information from the database, which will include prices with datatypes of double, item names as strings, item numbers as string, a “Weigh Item” as a Boolean (where 1 means that the price is calculated based on the item’s weight), and product images as .png files. A manager can modify, add, or delete this inventory information, and the inventory will automatically be updated after a purchase has been made to indicate the change in stock of the items.

The staff information will consist of only their name, username, password, and staff type (manager/admin) all as strings, and this information is inputted to the system using one of the computers in the store.

6.4 Communications Interfaces

Managers and admins will login to the system using electronic forms from desktop computers in the store. The customers interact with the system using the self checkout machines and the menus that are displayed on the screen. Large buttons in the user interface allow the customers to navigate through the menus, and the transaction information is displayed on the screen.

The system cannot be accessed through a web browser, and it does not send emails.

7 Non-Functional Requirements/Quality Requirements

7.1 Ease of Use

A wide array of people all with different backgrounds, ages, and education will use this system. Therefore, an intuitive interface for the user is a must to reduce the need for employees and managers to help customers navigate the system and its uses. An easy to use system will also encourage customers to continue using the checkout system over traditional check outs, reducing the customer demand on company employees.

7.2 Performance Requirements

Although high speed performance is not critical, the system should operate relatively quickly. Customers are more likely to continue using the system if it is quicker than using a traditional checkout. Additionally, a faster system will reduce the time a customer needs to checkout their items allowing for a higher customer throughput and taking more load off the older systems.

7.3 Concurrent Use

Extending the performance requirements; several customers may be using the checkout system at the same time, there should be no performance degradation regardless of how many terminals are in use.

7.4 Data Integrity

Since prices and items can change often at a grocery store it is imperative that data is updated regularly and kept consistent across all databases to prevent issues such as incorrect billing, different prices for the same item, or items not found during checkout.

7.5 Security Requirements

Sensitive customer information is being handled by the system while processing payments. This information needs to be protected and the system should be secure to prevent information leaking to people who should not have it. Payment and customer information should not be passed around unencrypted.

8 Prioritization and Release Plan

8.1 Choice Prioritization Method

The highest priority level is Level 1and the lowest is Level 3. The rationale behind assignment of each function is given in the remarks in Table 1 below. Overall, functions that were directly responsible for the major functions of the system were assigned a Level 1. Functions that provided very useful functionality but, without which the system could still be made to operate, were assigned a Level 2. Functions that were assigned a Level 3 were nice-to-have features.

8.2 Prioritization Table of Requirement

Table 1 Prioritization and Release Plan Table

FEATURE ID	PRIORITY LEVEL	RELEASE DATE	REMARKS
RE1	2	March 11, 2019	Scanning an item is important but it is an extension of RE3, because the item number is entered when the barcode is scanned.
RE2	2	April 2,2019	Searching for items is an essential requirement for some items, but it is not essential for the system as a whole to function.
RE3	1	March 4, 2019	Adding items to an order using a product number is an essential component to the system. The other

			methods to add an item to an order are extensions of this method.
RE4	2	March 15, 2019	Having multiple functioning methods that the user can use to pay is an essential component to the system, but our priority is being able to add items to an order and interface with the inventory database.
RE5	3	March 15,2019	Printing receipts is a low level priority, but its release date is the same as RE4 because it is closely tied to it.
RE6	1	March 4, 2019	The core functionality of this system is the interaction between orders and the inventory database. Updating the database is key to the system working properly, and helps the business to know when to order more items.
RE7	1	March 4, 2019	Similar to RE6, this is part of the core functionality of the system.
RE8	3	Dec 31, 2019	This is a low level priority because the system can operate using only one manager account.
RE9	1	March 15, 2019	In order to see that the core functionality is working properly the system needs to be capable of displaying the inventory information to managers.
RE10	2	April 15,2019	This requirement is necessary for the system to be secure, but it is not necessary at the beginning stages of testing the system.
RE11	2	September 30, 2019	This is a somewhat important feature because the customers may incorrectly input items into their order, and managers must be able to assist them in correcting their order.