**Product Requirements**

|  |  |
| --- | --- |
| **Team** | s361-02a |

# Brief problem statement

The objective of this project is to create a new vending machine system that will allow for easier operation, management, and data analysis of the machines, products, etc. Three applications will be constructed to allow for better use from the standpoints of the customer, restocking employees, as well as the management and/or owners. The customer-side application will allow for easier access to purchasable goods from the vending machine itself, and provide a clear and accessible method of product sales. The restocking-side application will relieve some unnecessary work from restocking employees, by keeping track of inventory (how much of a product remains, expired/recalled products, etc.), and remove some of the extraneous busywork associated with restocking. The management-side application will allow for owners and managers to view sales data and analytics from each of the operational machines, and allow them to rearrange and determine what should be sold in each machine, and relay any changes to the restocking employee.

# Stakeholders

* Customers.
* Company owners specializing in vending goods.
* Product distributors (Frito-Lay, Herr's, etc.)

# Users profile

* Customers will be using the system, from within the Customer Application.
  + All customers should be assumed to have little to no knowledge of any computer system (Toddler level)
  + Customer should only be aware of basic vending machine operation (Press button, enter payment, recieve food)
* Restocking employees will be using the system, from within the Restocking Application.
  + Employees using the system should be assumed to have fair knowledge of everyday electronics (mobile devices, home computers, basic applications. Teenager level.)
  + Employee must be able to navigate any menus within the application, but not be forced to understand any extreme/unclear design choices.
* Managers/Owners will be using the system through the Management Application.
  + Users should have more specialized knowledge of the system, and be able to navigate through the management interface with a higher degree of understanding.

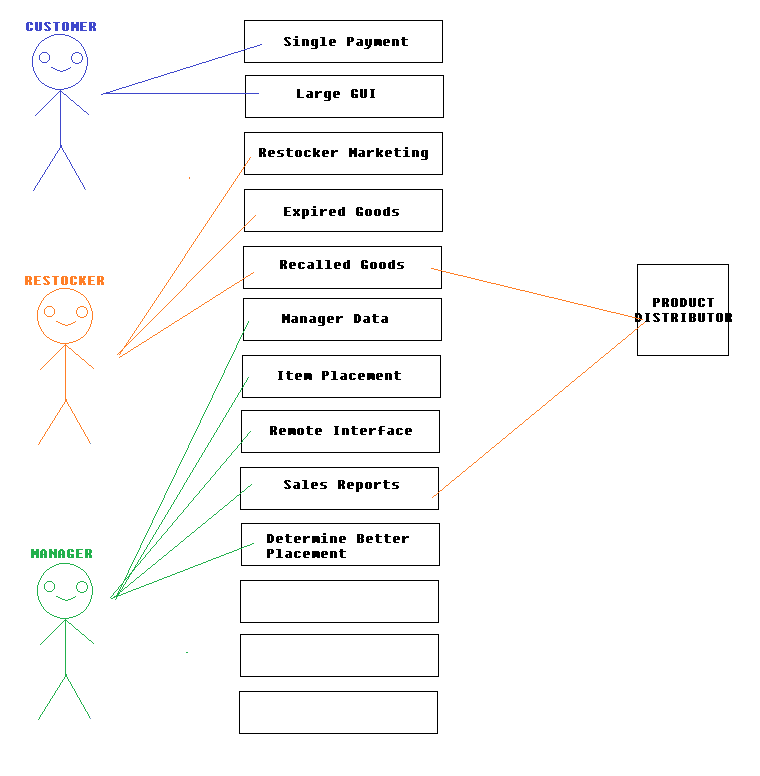
# System requirements

# Three applications, written in Java, must be constructed in order to provide services to the customers, managers, and restocking employees of the vending machine company. A terminal-based version must be constructed by Release 1. In addition, using JFrame, a GUI must be constructed by Release 2.

# Feature requirements (user stories)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **User Story Name** | **Description** | **Release** |
|  | CustomerPayment | As a customer, I want to make purchase several items, with one single transaction. | R1 |
|  | CustomerGUI | As a customer, I want to be able to use an easy-to-navigate GUI. | R2 |
|  | RestockerMarketing | As a restocker, I wish to only have to restock as the app tells me, no marketing decisions made by me on the spot. | R1 |
|  | RestockerExpired | As a restocker, I wish to know exactly what in the machine has expired or been recalled, in order to clearly indicate what must be removed. | R1 |
|  | RestockerInventory | As a restocker, I wish to know exactly how much of a product is left in the machine, and now much must be restocked, due to recalls, expiration, or simply being purchased. | R1 |
|  | ManagerData | As a manager, I want to be able to view real-time analytics from each machine, that shows the sales information from that location. | R1 |
|  | ManagerItemPlacement | As a manager, I want to be able to move items in the machine, to sell more of an item that might not be selling as well. | R1 |
|  | ManagerMothership | As a manager, I want to be able to do this all through a remote PC terminal at the office. | R1 |
|  | ManagerReport | As a manager, I want to view full sales reports of any and all sales at any machine. | R1 |
|  | ManagerAutomove | As a manager, in addition to manual placement, I wish to be able to have placements changed for me, depending on current item sales. | R2 |

**Use case diagram**



**Use case descriptions**

|  |  |
| --- | --- |
| **Use Case Number:** | UC-01 |
| **Use Case Name:** | CustomerPayment |
| **Overview:** | The customer wishes to be able to purchase multiple items in one, single checkout action. |
| **Actor(s):** | Customer |
| **Pre condition(s):** | Machine must be stocked with at least one purchasable, non-perished/non-recalled item. |
| **Scenario Flow:** | Main (success) Flow:   1. Customer approaches machine. 2. Customer selects one or more items he/she wishes to purchase. 3. Upon selecting all items, customer presses "Check Out" button. 4. Machine displays how much money is due. 5. Customer pays (assuming exact payment) 6. Machine dispenses goods. 7. Machine resets, ready for next customer. |
|  | Alternate Flows:   * Customer wishes to cancel entire purchase. * Customer wishes to cancel only parts of purchase. * Selected item is not available for purchase (Out of stock, expired, recalled) |
| **Post Condition:** | * All goods dispensed. * Machine records sales on internal sales tracker. * Machine returns to main purchasing GUI for next customer |

|  |  |
| --- | --- |
| **Use Case Number:** | UC-02 |
| **Use Case Name:** | RestockerInventory |
| **Overview:** | The restocking employee wishes to know exactly what must be removed and placed within the machine, and where in the machine to place it. Employee should not be forced to make any marketing decisions on the spot. |
| **Actor(s):** | Restocking Employee. |
| **Pre -condition(s):** | Machine should not be fully restocked. Machine should have at least one item missing, recalled, or expired, although it would be more appropriate to have multiple instances of missing items. |
| **Scenario Flow:** | Main (success) Flow:   1. Employee checks application, learns what must be removed from machine, and in what quantity. 2. Employee checks application, learns what must be added to the machine, and in what quantity. 3. Employee obtains new product from truck. 4. Employee opens machine, setting machine into RESTOCK mode. 5. Employee removes and replaces product, as outlined by application. 6. Employee closes machine. 7. Employee resets machine to CUSTOMER mode. |
|  | Alternate Flows:   * None to be determined at this time. |
| **Post Condition:** | * All goods in machine are fully stocked, non-expired, etc. * No expired or recalled goods are left remaining in the machine. * Machine returns to main purchasing GUI for next customer |

|  |  |
| --- | --- |
| **Use Case Number:** | UC-03 |
| **Use Case Name:** | ManagerData |
| **Overview:** | The Manager wishes to view real-time statistics and analytics of a given machine. |
| **Actor(s):** | Manager |
| **Pre -condition(s):** | Manager must be accessing the MANAGER application. Must be accessed remotely through a separate terminal. |
| **Scenario Flow:** | Main (success) Flow:   1. Manager accesses the machine using the MANAGER application. 2. Manager sets machine mode into MANAGORIAL mode. 3. Manager chooses, out of a menu, to display sales data of a certain machine. 4. Machine displays sales data, total stock, and and pending alerts (recalls/expirations) of all goods in machine. 5. Manager reviews data. 6. Manager resets machine into CUSTOMER mode. 7. Manager exits out of MANAGER application. |
|  | Alternate Flows:   * Before exiting application, manager wants to access another function of MANAGER application. * Manager wants to remotely turn off machine due to some error/restock request. |
| **Post Condition:** | * Manager now holds data related to the sales information of particular machine or all machines as a whole. * Machine is now in CUSTOMER mode. |