**Document 05 – Sprint 1 Report**

CS 4321 – Fall 2021

This document is contained in your GitHub repository in a folder named *docs*.

| | Group | 1 | | --- | --- | | Group Member Names |  |
| --- | --- | --- | --- | --- |
|  | | 1. | Greggory Buehring | | --- | --- | | 2. | Sean Vickers Jr. | | | 3. | Alex Acorn | | --- | --- | | 4. | Nathan Bailey | | 5. | David Truong | |

1. **Video Demo**

| **Deliverable**  Create a video (up to 20 minutes) for your demo and post the link here. The agenda for your demo:   1. (3-5 minutes) Discuss the design of your system using a class diagram.  * Explain at a high-level what each (or the most important) classes’ responsibilities are. Your goal is to give me a feel for your architecture and how the pieces fit together. * Don’t read off the list of methods! You can mention some key methods, or just describe what responsibilities each class has. If you need to go into more detail, it is fine, to show portions of the design and explain. * The class diagram should **not** show methods nor instance variables for any Gui classes (in StarUML, select the class, right-click, and choose: *Format, Suppress Operations* and *Suppress Attributes*. You can use the handles on the selected class to manually resize appropriately.) * If you need several versions of the class diagram, or need to break it into pieces that is fine. However, note how they join.  1. (5-15 minutes) Demo as many user stories as you want, starting from the top priority (lowest number) and work your way down. Then, for each:  * Display a user story (text, in word or whatever) and read it, Expand on it if necessary. * Illustrate how the software fulfills it.   You can produce several videos if needed, *i.e.* part 1, part 2, *etc.* |
| --- |

| Video Link | Part 1 : <https://youtu.be/zb6YhU2T2x8>  Part 2: <https://youtu.be/1VUPADGI0zQ> |
| --- | --- |

1. **Development Status**

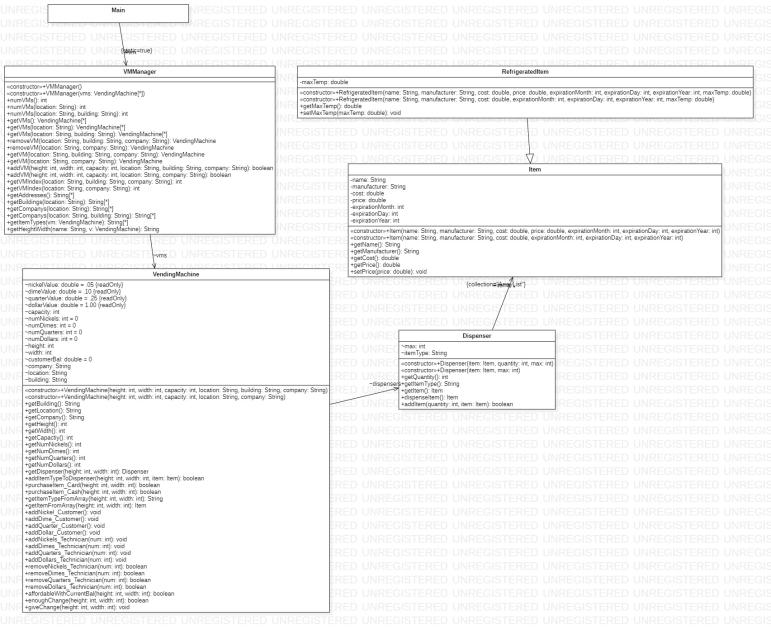
| **Deliverable**  Provide a list of user stories, ordered by their priority that were worked on during this Sprint. Include the user story number (“Num” in table below), status, title, and any comments if needed. For the status, use these symbols below. This is the *User Story Priority* table in Document 02, with Status and Comment columns added.   | **Symbol** | **Description** | | --- | --- | | ✔ | Complete (tested) | | ~ | In progress | | X | Not started | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |

Note: We switched to using Dr. Gibson’s User Stories that were given.

| **Priority** | **Num** | **Status** | **Title** | **Comment** |
| --- | --- | --- | --- | --- |
| 1 | 1 | ~ | View Product Names and Prices in VM | Currently you can see the names from VM. Prices to be added visually in Sprint 2. |
| 2 | 3 | ✔ | Purchase Product with Card | Works! |
| 3 | 5 | ✔ | Restock VM with Products | Works! |
| 4 | 8 | ✔ | Restock Cash in VM | Works! |
| 5 | 9 | ✔ | Add a Manufacturer | Works! |
| 6 | 10 | ✔ | Add a Non-Refrigerated Product | Works! |
| 7 | 12 | ✔ | Add a Location | Works! |
| 8 | 13 | ✔ | Add a Non-Refrigerated VM | Works! |
| 9 | 7 | ✔ | Remove Cash from VM | Works! |

1. **Class Diagram**

| **Deliverable**  Use Object Aid (HW-CSOA) to develop a neat, legible, properly sized UML class diagram(s) showing your current design that exactly reflects your code. Turn the page Landscape if needed. Provide several versions if desired. |
| --- |



1. **System Tests**

| **Deliverable**  Provide a number list of system tests that are performed manually. Can have multiple tests for a user story. Use the template below. Remove the prompts that are surrounded by brackets. An example follows. |
| --- |

| Test Num | 1 |
| --- | --- |
| US Num-Title | 12- Add a Location, 13- Add a non-refrigerated vm |
| Description | Added 1 Vending Machine along with a location |
| Flow of events | 1. Create A Vending Machine with address 1500 Main St, the building of Nevins, and the name Coke Machine 1 2. Use the customer to find the Vending Machine |
| Expected output/result | Expected- 1500 Main St -> Nevins -> Coke Machine 1  Result- 1500 Main St -> Nevins -> Coke Machine 1 |
| Comments | n/a |

| Test Num | 2 |
| --- | --- |
| US Num-Title | 12- Add a Location, 13- Add a non-refrigerated vm |
| Description | Added 2 Vending Machines with the same location |
| Flow of events | 1. Create A Vending Machine with address 1500 Main St, the building of Nevins, and the name Coke Machine 1 2. Create A Vending Machine with address 1500 Main St, the building of Nevins, and the name Coke Machine 2 3. Use the customer to find the Vending Machines |
| Expected output/result | Expected- 1500 Main St -> Nevins -> Coke Machine 1/Coke Machine 2  Result- 1500 Main St -> Nevins -> Coke Machine 1/Coke Machine 2 |
| Comments | n/a |

| Test Num | 3 |
| --- | --- |
| US Num-Title | 12- Add a Location, 13- Add a non-refrigerated vm |
| Description | Added 2 Vending Machines with the different location |
| Flow of events | 1. Create A Vending Machine with address 1500 Main St, the building of Nevins, and the name Coke Machine 1 2. Create A Vending Machine with address 2201 Baytree Rd, the building of H, and the name Coke Machine 2 3. Use the customer to find the Vending Machines |
| Expected output/result | Expected- 1500 Main St -> Nevins -> Coke Machine 1  2201 Baytree Rd -> H -> Coke Machine 2  Result- 1500 Main St -> Nevins -> Coke Machine 1  2201 Baytree Rd -> H -> Coke Machine 2 |
| Comments | n/a |

| Test Num | 4 |
| --- | --- |
| US Num-Title | 8 - Restock Cash In VM |
| Description | Restock Vending Machine’s Cash for change calculations of transactions |
| Flow of events | 1. Create A Vending Machine with address 1500 Main St, the building of Nevins, and the name Coke Machine 1.  2. Select the technician button from the main menu  3. Select Restock Currency  4. Select the address, building (if applicable), and name of the vending machine, then select Go To Vending Machine  5. Use the corresponding buttons to add the necessary change (One of each type in our case). |
| Result | Expected: Nickels: 1 Dimes: 1 Quarters: 1 Dollars: 1  Result: Nickels: 1 Dimes: 1 Quarters: 1 Dollars: 1 |
| Comments |  |

| Test Num | 5 |
| --- | --- |
| US Num-Title | 7 - Remove Cash In VM, 8 - Restock Cash In VM |
| Description | Remove Vending Machine’s Cash |
| Flow of events | 1. Create a vending machine with the address 1500 Main St, the building of Nevins, and the name Coke Machine 1.  2. Select the technician button from the main menu  3. Select Restock Currency  4. Select the address, building (if applicable), and name of the vending machine, then select Go To Vending Machine  5. Use the corresponding buttons to add the necessary change (One of each type in our case).  6. Use the corresponding buttons to remove the requested change (One nickel and one dime in our case). |
| Result | Expected: Nickels: 0 Dimes: 0 Quarters: 1 Dollars: 1  Result: Nickels: 0 Dimes: 0 Quarters: 1 Dollars: 1 |
| Comments | n/a |

| Test Num | 6 |
| --- | --- |
| US Num-Title | 8 - Restock VM with Products |
| Description | Restock Vending Machine’s Products that already are assigned to it |
| Flow of events | 1. Create A Vending Machine with address 1500 Main St, the building of Nevins, and the name Coke Machine 1 with dimensions of height 5, width 5, and capacity 5.  2. Create an item with name Coke and Manufacturer Coca-Cola  3. Select the technician button from the main menu  4. Select the Restock items button  5. Select the address, building (if applicable), and name of the vending machine, then select Go To Vending Machine  6. Select Coke and press the Choose Item button  7. Enter the amount to restock (5 in our case) and press the Restock button |
| Result | Expected- Success!  Result-Success! |
| Comments | n/a |

| Test Num | 7 |
| --- | --- |
| US Num-Title | 8 - Restock VM with Products |
| Description | Try to restock Vending Machine’s Products that already are assigned to it with too much product |
| Flow of events | 1. Create A Vending Machine with address 1500 Main St, the building of Nevins, and the name Coke Machine 1 with dimensions of height 5, width 5, and capacity 5.  2. Create an item with name Coke and Manufacturer Coca-Cola  3. Select the technician button from the main menu  4. Select the Restock items button  5. Select the address, building (if applicable), and name of the vending machine, then select Go To Vending Machine  6. Select Coke and press the Choose Item button  7. Enter the amount to restock (6 in our case) and press the Restock button |
| Result | Expected-Failed  Result-Failed |
| Comments | n/a |

| Test Num | 8 |
| --- | --- |
| US Num-Title | 3- Purchase Product With Card, 9. Add a Manufacturer |
| Description | Add 1 item and have the customer buy it with the card |
| Flow of events | 1. Create a vending machine 2. Create an item with name Coke and Manufacturer Coca-Cola 3. Have technician add Coke to vending machine 4. have customer access vending machine to see Coke 5. Purchase with card |
| Expected output/result | Expected- 1500 Main St -> Nevins -> Coke Machine 1 -> Coke -> purchase successful  Result- 1500 Main St -> Nevins -> Coke Machine 1 -> Coke -> purchase successful |
| Comments | n/a |

1. **Retrospective**
2. Read this short page about what a software retrospective is and why it is important:

<https://searchsoftwarequality.techtarget.com/definition/Agile-retrospective>

1. Meet as a group and discuss the following questions and provide a group written response below:
2. What worked well for us?

| **Answer**  Two of the key actions that our group had was regular meetings and communicating our ideas to each other. |
| --- |

1. What did not work well for us?

| **Answer**  Scheduling conflicts and other coordination issues made this sprint somewhat difficult.  The Gui’s program is confusing to look at. |
| --- |

1. What actions can we take to improve our process going forward?

| **Answer**  Actions that we could do to improve our process in the future are the following: more meetings throughout the week, having clear times certain objectives need to be done, and time management.  We can convert our GUI to FXML to make it more organized. |
| --- |