**Document 06 – Sprint 2 Report**

CS 4321 – Fall 2021

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Group | 6 | | Group Member Names |  |
|  | |  |  | | --- | --- | | 1. | Bello Abarchi | | 2. | Emmanuel Olofintuyi | | |  |  | | --- | --- | | 3. | Gavin Suttles | | 4. | Thuong Pham (Tim) | | 5. |  | |

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 | https://www.youtube.com/watch?v=kcStgVCifwA |
|  | This Video Link is also posted inside the main branch in GitHub |

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 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Priority** | **Num** | **Status** | **Title** | **Comment** |
| 1 | 12 | ✓ | Products information for customers | A customer can see the price, a picture of the product and the product name. |
| 2 | 19 | ✓ | Difference between refrigerated items and dry goods | Items are attributed with R or D, refrigerated or Dry. Dry and refrigerated are in separate machines. |
| 3 | 22 | ✓ | Restock vending machine properly so that customers can purchase items | The technician can restock the machine |
| 4 | 3 | ✓ | The machine receives customer change ONLY after purchasing | The customers change is separate |
| 5 | 14 | ✓ | Generate vending machine reports | Reports : Vending machine, product, location |
| 6 | 7 | ✓ | stock the machine with adequate change | The machine has base change, the technician can add more which also increases the machine’s total sum of change. |
| 7 | 10 | ✓ | Generate the problem report | In progress |
| 8 | 5 | ✓ | Keep track of inventory | Stores a quantity of items in a list, Technician can view them. |
| 9 | 8 | ✓ | Show accurate item details | Technician can view all the details of items, including expiration, quantity, price, and name. |
| 10 | 25 | ✓ | Receipt printing | A customer purchases, gets the product, and is told how much change they get back. |
| 11 | 16 | ✓ | Show Location/Company | Shows location and company. |
| 12 | 26 | ✓ | Items expire | Items have expiration dates that a technician can view |
| 13 | 27 | ✓ | Remove expired items | A technician can remove all the expired items from the machine |
| 14 | 20 | ~ | Only validated person can access the vending machine | As of right now, there are 3 views. No validation required, but a technician/customer/manager view all have separate functions and views. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

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. |

Note: Because the class diagram image is too big, we post it with the file name [ClassDiagram.jpg](https://github.com/vsu-se/team6_f21/blob/main/ClassDiagram1.jpg) on the main branch in GitHub.

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 | Ability to view vending machine contents |
| Description | The Users can view the items on the items list. Besides that, the buttons and input bar also present next to the lists on the vending machine that helps the users can easily access and purchase the items |
| Flow of events | 1. Run the system, there are 12 pictures of alter items present on the system 2. Customer has the own scene in the system to access the purchase 3. Input bar where the user can insert the cash to purchase the item 4. Radio buttons to choose the payment methods 5. Purchase button is the final button to buy the product after picking item, picking payment method, and inserting the cash 6. Clicking on every button on the scene without choosing the any item 7. After clicking of the expected items, there are some outputs show up to announce the desire item and prompting the payment method |
| Expected output/result | Running the application. When I click on the Customer tap, I want to see:  -All items image  -2 radio buttons for select method of payment. One for Cash option, and another for Credit card option  -input bar  - output bar  - purchase button  Clicking on every button, the output bar will have the greeting to the customer:  “Hi! How can I help you?”  Then after choosing one item, I expect to see the message as:  Name item: price  “Please select method of payment”  Please pick an item” |
| Comments | This test is checked and passed. All expected showed up after hitting the Customer tap and item |

|  |  |
| --- | --- |
| Test Num | 2 |
| US Num-Title | Prepare to buy product from the vending machine with a selected method |
| Description | Selecting the type of payment to pay for the product. For example, the user after choosing the item, they can choose the options to pay. It is by Cash or by Credit Card |
| Flow of events | 1. Selecting the item 2. The machine will prompt the payment method to purchase the item 3. Click on one of the payment methods |
| Expected output/result | If clicking on the Cash radio button, the vending machine prompts:  “Please insert the cash!”  After executing the purchase button, the vending machine will ask the user:  “Please insert the valid cash or smaller bill!”  If clicking on the Credit Card button, the vending machine prompts:  “none” (because the customer pays by their credit card, so there are no requesting cash input in this case) |
| Comments | This test is checked and passed. |

|  |  |
| --- | --- |
| Test Num | 3 |
| US Num-Title | Receive change from the machine |
| Description | The vending machine will give the change back to the customer if the customer deposit more than the item’s price. This perform only works on the Cash payment. |
| Flow of events | Choosing an item “Cheetos”.  Selecting the Cash payment  Insert the amount of number that is bigger than the item’s price $4.0  Clicking the purchase button |
| Expected output/result | In the output bar, the system will say:  “Thanks for shopping!”  “Your change is: $1.80”  The radio cash button also become false |
| Comments | ---The test is checked and passed  ---The team has not completed the requirements of A vending machine accepts cash (one dollar bills only), change (nickels, dimes, and quarters) in sprint 1. There the temporary presented as cash in this sprint is integer or double that type into the input bar |

|  |  |
| --- | --- |
| Test Num | 4 |
| US Num-Title | Accurate price and quantity of the product |
| Description | The manager can generate reports. The reports show the sold items and details of the vending machine. |
| Flow of events | purchasing first item: “Cheetos” by Cash  purchasing second item: “Lays” by Cash  purchasing third item: “Cliff Bar”  Clicking on the Manager tab and product report button |
| Expected output/result | In the output on the system will be all the list of items left and the products got sold:  : M$M $1.2  Dorito: $5.3  Takis: $3.4  Pringles: $4.5  Reese: $4.5  Tostitos: $1.75  KindEnergy: $3.5  Take Five: $1.75  Nutter butter: $1.80  The products get sold:  Cheetos: $2.2  Lays: $5.6  ClifBar: $2.5 |
| Comments | ---The test is checked and passed |

|  |  |
| --- | --- |
| Test Num | 5 |
| US Num-Title | Generate the vending machine report |
| Description | The manager can create the new vending machine and read the items, vending machine, location, and manufactory report |
| Flow of events | Create Vending machine by typing vM and Location’s name: “vending one”, “Nevins hall”  Clicking on the product report button to generate the specific vending machine  Clicking on the machine button to generate the sold items  Clicking on dry and refrigerator to generate the type of vending machine  Click on the Manufactories button to generate the manufactory button |
| Expected output/result | In the output on the system will be all the list of items left and the products got sold:  : M$M $1.2  Dorito: $5.3  Takis: $3.4  Pringles: $4.5  Reese: $4.5  Tostitos: $1.75  KindEnergy: $3.5  Take Five: $1.75  Nutter butter: $1.80  The products get sold:  Cheetos: $2.2  Lays: $5.6  ClifBar: $2.5 |
| Comments | ---The test is checked and passed |

|  |  |
| --- | --- |
| Test Num | 6 |
| US Num-Title | stock the machine with adequate change |
| Description | The machine has base change, the technician can add more which also increases the machine’s total sum of change. |
| Flow of events | Go to technichian tab  Select the change you want to restock in the listview “listChange”  Type the amount of change you want to restock in the text field “txtChangeAmount”  Press restock change  Updates how much change is in the money drawer |
| Expected output/result | In the output on the system will be the current amount of change in the system:  Nickels: 10  Dimes: 10  Quarters: 10  Dollars: 10  Total Sum: $14.0 |
| Comments | ---The test is checked and passed |

|  |  |
| --- | --- |
| Test Num | 7 |
| US Num-Title | Generate the problem report |
| Description | We do not have a specific problem report but a lot of our reports tells is whats wong woth the vending machine |
| Flow of events | Go to technician tab  Select the change you want to restock in the listview “listChange”  Type the amount of change you want to restock in the text field “txtChangeAmount”  Press restock change  Updates how much change is in the money drawer |
| Expected output/result | In the output on the system will be the current amount of change in the system:  Nickels: 10  Dimes: 10  Quarters: 10  Dollars: 10  Total Sum: $14.0 |
| Comments | ---The test is checked and passed |

|  |  |
| --- | --- |
| Test Num | 8 |
| US Num-Title | Keep track of inventory |
| Description | Stores a quantity of items in a list, Technician can view them. |
| Flow of events | Go to technician tab  You will see that all the items are initialized on the first listview with all their attributes including quanity. |
| Expected output/result | In the output will be the list of items in the current vending machine initialized in :  Cheetos| 2.2| 10| 12/8/2021| D  Lays| 5.6| 10| 12/8/2021| D  Doritos| 2.6| 10| 12/8/2021| D  M\_M| 1.2| 10| 12/8/2021| D  . . . . . . . . . . . |
| Comments | ---The test is checked and passed |

|  |  |
| --- | --- |
| Test Num | 9 |
| US Num-Title | Show accurate item details |
| Description | Technician can view all the details of items, including expiration, quantity, price, and name. |
| Flow of events | Go to technician tab  You will see that all the items are initialized on the first listview with all their attributes. |
| Expected output/result | In the output will be the list of items in the current vending machine initialized in :  Cheetos| 2.2| 10| 12/8/2021| D  Lays| 5.6| 10| 12/8/2021| D  Doritos| 2.6| 10| 12/8/2021| D  M\_M| 1.2| 10| 12/8/2021| D  . . . . . . . . . . . |
| Comments | ---The test is checked and passed |

|  |  |
| --- | --- |
| Test Num | 10 |
| US Num-Title | Receipt printing |
| Description | A customer purchases, gets the product, and is told how much change they get back. |
| Flow of events | Choosing an item “Cheetos”.  Selecting the Cash payment  Insert the amount of number that is bigger than the item’s price $4.0  Clicking the purchase button |
| Expected output/result | In the output bar, the system will say:  “Thanks for shopping!”  “Your change is: $1.80” |
| Comments | ---The test is checked and passed |

|  |  |
| --- | --- |
| Test Num | 11 |
| US Num-Title | Show Location/Company |
| Description | Shows location and company |
| Flow of events | Go to Manager tab  Click on generate report  Click on Location report to see the location of current vending machine  Or  Click on Manufacturer report to see the manufacturers that each item belongs to |
| Expected output/result | In the output bar, the system will say:  For location  ------Location Report-------  Valdosta State University  Or  For Manufacturer  ------Manufacturer report-------  PepsiCo  Cheetos  Lays  Doritos  Tostitos  Mars  Kind  M&Ms  Barcel  Takis  Hershey  Reese's  Take 5Clif Bar & Company  Clif Bar/nKellogg's  Pringles  Nabisco  Nutter Butter  Coca-Cola Company  coca\_cola  Diet coke  cherry coke  Sprite  Dasani  Fanta  Keurig Dr Pepper  CanadaDry  DrPepper  Hornell Brewing Co.  Arizona |
| Comments | ---The test is checked and passed |

|  |  |
| --- | --- |
| Test Num | 12 |
| US Num-Title | Items expire |
| Description | Items have expiration dates that a technician can view |
| Flow of events | Go to technichian tab  Press show expired button  Then the program goes through all the items and see if there expiration date is today or has passed  If it is expired then it will pop up in the text area right next to the button, with how many of the items are expired |
| Expected output/result | In the output bar, the system will show:  Cheetos 10  Lays 10  Doritos 10  M\_M 10  Tostitos 10  Takis 10  Pringles 10 |
| Comments | ---The test is checked and passed |

|  |  |
| --- | --- |
| Test Num | 13 |
| US Num-Title | Remove expired items |
| Description | A technician can remove all the expired items from the machine |
| Flow of events | Go to technichian tab  Press show expired button  Then the program goes through all the items and see if there expiration date is today or has passed  If it is expired then it will pop up in the text area right next to the button, with how many of the items are expired  Then press remove expired  The program sets the quanity of all expired items to 0 and updates the expiration date so it will not be expired  Press show expired again and the text area should be blank |
| Expected output/result | In the output bar, the system will show:    (text area should be blank) |
| Comments | ---The test is checked and passed |

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>

Ans: In an agile project, a Retrospective is a ritual held at the end of each iteration. The overall goal is for the team to review its last working cycle. Furthermore, this is a critical time to get feedback on what went well and what did not.

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

|  |  |
| --- | --- |
| |  | | --- | | * **Our ability to meet and work on the project was great** * **Asking questions to each other about coding approach and design** * **Learning new technology to make the GUI design easier** * **Overall great teamwork skills** * **Great coding skills and shared interest in developing the best software possible** * **Developed Pair programming skills during team meetings** * **Helping each other and improve one another’s skill** | |

1. What did not work well for us?

|  |
| --- |
| * **Software not working (javaFX not running on some computers, also github not working adequately on eclipse and in general)** * **TimeLog failing multiple times over the course of the project** * **Not testing before coding which is recommended by agile approach** * **Not knowing what which user story to implement first. Solved by rethinking ranking of user stories** |
|  |

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

|  |
| --- |
| * **Be more familiar with github.** * **Testing before coding.** * **Spend more time on design in order to make implementation easier.** * **Dynamic gui window(resizable)** * **Fix a few bugs in the program** |