# Computer Science Internal Assessment : Grade 10 Term 1

Shuren Gabriel Yu

# Criteria A : Planning

## Defining the Problem or Unanswered Question

In the ISF Academy, we value choice of food for both students and teachers. Throughout the past few years, the school lunch choices increased from just ABC lunch menus to having the Daily Carving, the Daily Special, the Salad bar, Hot Meals in a Bowl and more. However, school lunches often have a long waiting line, limiting the amount of social time among students during lunch break. While the school has increased lunch break time over the years, the wait time to collect school lunches have greatly decreased the time students are allowed to enjoy their break. By decreasing lunch break, students will have less time to do school work and less time to socialize. This decrease in lunch time limits productivity, development of social skills and in general well-being among students. Students may also not have enough time to eat lunch at school due to this issue. Not eating lunch would increase malnutrition and productivity at learning for students. While an increase of lunch choices would give students a higher sense of independence and choice, it does not greatly decrease how much time it takes for students to collect school lunches.

## Justification for the Proposed Product

Vending machines could help solve the problem of long lunch lines in ISF during lunch break. While preparation of food in current lunch selections takes around 15 seconds on average, the use of a vending machine would take only 5 seconds on average. By using vending machines, collecting lunch would decrease the time collecting school lunch by a factor of 3. Vending machines also take less money to maintain compared to staffed stations in the cafeteria, so it would also be beneficial from an economic standpoint for the school. My proposed product is to create hardware that will operate a vending machine. This hardware will be on an arduino board coded using the language Python. This hardware will include buttons and specific prices associated with the buttons. When the buttons are pressed, a monitor will display the price of the item. After this, the user can put currency through the coin slot, and when the price is reached, the user will receive the item selected. There will also be a refund button, and if the total money has not been reached yet or the user has gone over the price set, the money will be dispensed back to the user.

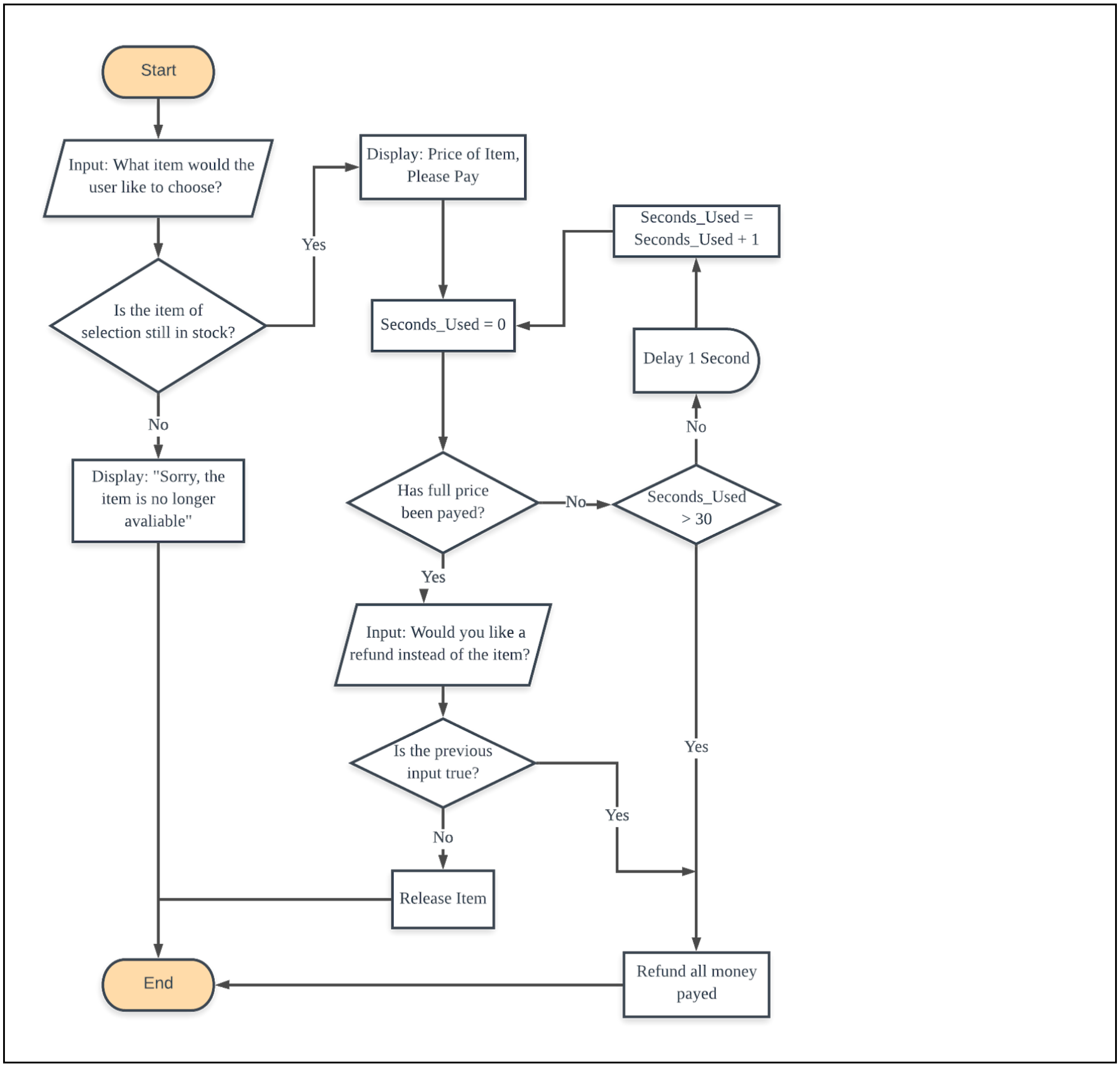
## Success Criteria

| Specification | Description |
|---|---|
| Function | - Program on python that would link to the arduino board and perform the vending machine specifications below.<br>- Software that controls hardware that dispenses a selected item that the user chooses when the price is paid.<br>- Refund system that dispenses the coins put in back to the user. |
| Ease of Use | - Buttons are easily interpreted for what they do<br>- Monitor information is easily read<br>- Refunded money is easily collect |
| Customer | - Must be able to deliver an item to the audience when they pay enough for it |
| Size | - Must be able to stand still and is accessible by a client when they are standing up |
| Components | - Buttons that will cause a monitor to display the price of the selected item<br>- Currency dispensing area, where the user can put currency and the currency dispensed is displayed on a monitor<br>- When the desired price is reached by the user, the user will receive the item that they selected on the vending machine<br>- Refund button that will dispense back money to the user |
| Aesthetics | - Must fit in with the schools architectural color theme depending on where it is put |
| Materials | - The program which will control the vending machine will be coded in Python, version 3.7<br>- Arduino board<br>- Buttons to click and choose items<br>- Display monitor which displays information<br>- Motor to dispense items from the vending machine<br>- Materials to construct a prototype vending machine, such as cardboard |

# Criteria B : Solution Overview

## Record of Tasks

### Flowchart


Start
Input: What item would the user like to choose?
Is the item of selection still in stock?
Yes
No
Display: "Sorry, the item is no longer avaliable"
Display: Price of Item, Please Pay
Seconds_Used = 0
Seconds_Used = Seconds_Used + 1
Delay 1 Second
No
Has full price been payed?
No
Seconds_Used > 30
Yes
Input: Would you like a refund instead of the item?
Is the previous input true?
Yes
Yes
No
Release Item
End
Refund all money payed

# Design Overview

## Database Design Example

https://drive.google.com/open?id=1p1xM6cKM9SxFRGe_Dd7GYzbNk7nBDtXr

| Number (unique integer) | Food_Eng (string size: 20) | Food_Chi (string size: 8, unicode) | Cost (float) | Stock (integer) |
|---|---|---|---|---|
| 1 | "Boiled Egg" | 煮雞蛋 | 8.5 | 4 |
| 2 | "Pork Chops" | 豬排 | 34.2 | 6 |
| 3 | "Beef Steak" | 牛排 | 54.0 | 1 |
| 4 | "Scrambled Egg Rice" | 炒蛋飯 | 20.0 | 0 |
| 5 | "Fried Chicken" | 炸雞 | 18.5 | 3 |
| 6 | "Fish Curry Rice" | 魚咖哩飯 | 32.0 | 7 |

## Function Headers + Documentation

https://drive.google.com/file/d/1Gy2_YOV4hccQCUJoLZkdELMAS-K9glTp/view?usp=sharing

## Use Cases

**Use Case 1**

**Use Case:** Select a food item

**Actor:** Vending machine customer

**Scope:** At school

**Brief:** The vending machine customer selects their food item of choice and the price is displayed on the vending machine screen/display. If there is no more stock to the item the user selected, the vending machine will display “Sorry, the item is no longer available”

Steps:

1. Vending machine customer clicks the button correlating to the food item of their choice

2. Vending machine finds the remaining stock of the item selected
3. If the vending machine is out of the item, it displays the message: “Sorry, the item is no longer available”
4. Otherwise the vending machine displays price correlating to the food item

Precondition

1. Vending machine has power and is charged
2. The vending machine is fully functional (nothing is broken)

Postcondition

1. No Success
    - The vending machine does not receive any information from the user pressing the button, and does not display the price of any food item or “Sorry, the item is no longer available”
2. Partial Success
    - The vending machine displays the price of the wrong food item
    - Or the vending machine displays the wrong message, e.g. if there is still stock and the vending machine displays “Sorry, the item is no longer available”
3. Full Success
    - If the item is still available, the vending machine displays the price of the item that the user selected
    - If there is no more stock, the vending machine displays: “Sorry, the item is no longer available”

**Use Case 2**

**Use Case:** Customer pays the desired amount of money for their item
**Actor:** Vending machine customer
**Scope:** At school
**Brief:** When the customer takes under 30 seconds to pay the full price, a message asking if the customer wants a refund is displayed

Steps:

1. Customer pays the desired amount of money for the item they selected within 30 seconds
2. Vending machine displays message asking if the customer wants a refund

Precondition

1. Vending machine has power and is charged
2. The vending machine is fully functional (nothing is broken)
3. There is full success in terms of use case 1, meaning that the item is selected and the correct price is displayed
4. The desired amount for the item is paid

Postcondition

1. No Success
    - Message displayed stays on the price message, with the customer getting nothing in compensation
    - No refund message displayed
2. Partial Success
    - The item is given to the customer, with no refund message displayed
    - The item is given to the customer while the refund message is also displayed
3. Full Success
    - A refund message is displayed and the item is not given to the customer

**Use Case 3**

**Use Case:** Customer takes to long, all money given is refunded
**Actor:** Vending machine customer
**Scope:** At school
**Brief:** When the customer takes over 30 seconds to pay the full price, all the money that the customer has paid so far is refunded in the refund slot

Steps:

1. Customer doesn't pay enough money for their selected item in 30 seconds
2. Vending machine refunds their money if they gave any

Precondition

1. Vending machine has power and is charged
2. The vending machine is fully functional (nothing is broken)
3. There is full success in terms of use case 1, meaning that the item is selected and the correct price is displayed
4. The desired amount for the item is not yet paid

Postcondition

1. No success
    - Customer doesn't receive any refund if they paid any money originally
2. Partial success
    - Customer who didn't pay full price gets the wrong amount refunded back
3. Full success
    - Customer gets all the money they originally paid refunded back 30 seconds after they choose their food item

**Use Case 4**

**Use Case:** Customer selects refund option after they pay for the item
**Actor:** Vending machine customer

**Scope:** At school
**Brief:** The customer selects the refund button, and all the money they gave is refunded

Steps:

1. Customer clicks the refund button option
2. Vending machine refunds all money that the customer put in the machine

Precondition

1. Use case 1 has full success
2. User has enough money and goes through use case 2 with full success
3. Vending machine has power and is charged
4. The vending machine is fully functional (nothing is broken)

Postcondition

1. No success
    - Customer does not get any money they put in the machine as a refund
    - Customer gets item instead of refund
2. Partial success
    - Customer gets a partial amount of money they put in the machine
    - Customer gets more than what they put into the machine as a refund
3. Full success
    - Customer gets the same amount of money refunded as they put into the machine originally

**Use Case 5**
**Use Case:** Customer does not select the refund option after they pay for the item
**Actor:** Vending machine customer
**Scope:** At school
**Brief:** The customer selects the non-refund button, and their desired item is dispensed

Steps:

1. Customer clicks the non-refund button option
2. Vending machine gives the customer their desired item (selected item)

Precondition

1. Use case 1 has full success
2. User has enough money and goes through use case 2 with full success
3. Vending machine has power and is charged
4. The vending machine is fully functional (nothing is broken)

Postcondition

1. No success

- The customer does not get the item or anything in compensation
- The customer gets a refund instead of their desired item

2. Partial success
    - Customer received wrong item from the machine (not their selected item)
3. Full success
    - Customer receives their desired item (selected item)

## Test Plan

### 1. Test Strategy

#### 1.1. Test Assumptions

**Key Assumptions**

- The software is already implemented within the hardware. This would allow both software and hardware errors to be accounted and therefore tested for.

**General**

- The hardware will have power to run its components during testing.
- Hardware will be built and assembled before testing begins.
- The software for the vending machine will be assembled/written before testing begins.
- The hardware has no physical issues, such as a collapsing body, or a broken button that does not work even with power.

#### 1.2. Test Principles

- Testing will be focused on achieving full success on all use cases described in the “Use Cases” part of the internal assessment.
- Testing can be concluded in no success, partial success and full success
- Testing processes will be well defined, yet flexible, with the ability to change as needed.
- Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
- Testing will be repeatable.
- Testing will have either/both qualitative and quantitative activity depending on the test being performed.
- Testing will be divided into phases, each with clearly defined objectives and goals.
- There will be an entrance and exit criteria for each test that will be performed.

#### 1.3. Scope and Levels of Testing

##### 1.3.1. Exploratory

- **Purpose:** The purpose of the level of exploratory test is to make sure critical defects are removed before the next level (functional test) can start.
- **Scope:** Software errors which prevent the software from starting, hardware errors which prevent the program from being run.
- **Tester:** Creator of the vending machine, who is also the writer of this internal assessment.
- **Method:** This exploratory testing on both software and hardware will be carried out without any test scripts and documentation.
- **Timing:** For software testing, when the software accounts for each of the use cases. For hardware testing, when the hardware is completely assembled together.

#### 1.3.2. Functional Test

- **Purpose:** Functional testing will be performed to check the functions of the software and the hardware in relation to the use cases. The functional testing is carried out by feeding the input .
- **Scope:** The software and hardware of the vending machine.
- **Tester:** Creator of the vending machine, who is also the writer of this internal assessment.
- **Method:** This test will be performed by following the "full success" criteria on each use cases. The result of each test will be written down on paper so that any errors or issues can be remembered and thus, fixed.
- **Timing:** After the exploratory test is completed

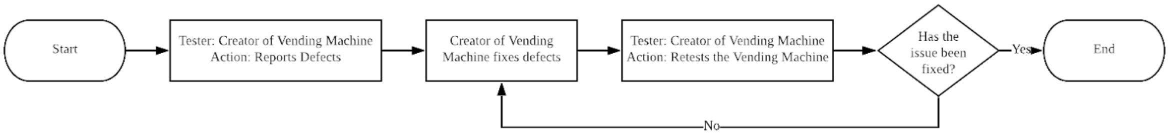
## 2. Execution Strategy

### 2.1. Entry and Exit Criteria

- The entry criteria refer to the desirable conditions in order to start test execution.
- The exit criteria are the desirable conditions in order to proceed with the testing process. This is all according to the full success criterion of the use cases.
- Entry and exit criteria are flexible benchmarks, and change depending on what is more suitable to achieve the success criteria of the vending machine.

| Use Case Number | Exit Criteria |
|---|---|
| 1 | - If the item is still available, the vending machine displays the price of the item that the user selected<br>- If there is no more stock, the vending machine displays: “Sorry, the item is no longer available” |
| 2 | - A refund message is displayed and the item is not given to the customer |
| 3 | - Customer gets all the money they originally paid refunded back 30 seconds after they choose their food item |
| 4 | - Customer gets the same amount of money refunded as they put into the machine originally |
| 5 | - Customer receives their desired item (selected item) |

### 2.2. Defect Tracking & Reporting


Start
Tester: Creator of Vending Machine
Action: Reports Defects
Creator of Vending
Machine fixes defects
Tester: Creator of Vending Machine
Action: Retests the Vending Machine
Has the
issue been
fixed?
Yes
End
No

## 3. Test Management Process

**Exploratory Testing:**

- Run software, if the software is prevented from running, fix the software error. Keep testing the software until it runs.
- Test hardware by running the software on the hardware, if the vending machine runs differently from what is needed for the vending machine to succeed, check for software issue. If it is a software issue, fix the software issue. If it is a hardware issue, fix the hardware by replacing the pieces which do not function properly.

**Functional Testing:**

- Run the program and test in relation to the success criterion of each use cases, starting with use case 1 and ending with use case 5 in an ascending order. If any errors arise, determine whether it is a software or hardware error. Fix the error. Then run it again in relation to the same use case being tested as before. If the vending machine works in full success of the use case, proceed to the next use case testing. If not, go back and fix the error again.

## 4. Approach in Testing

### 4.1. Who?

- The tests that will be performed on the vending machine will all be performed by the writer of this internal assessment, or the creator of the vending machine. This is because a vending machine does not require specific people to test it when following the use cases. Every person will experience the use cases the same, without any deviation. Therefore, having other people to test the vending machine will not be necessary. Testing it myself will be the most convenient option.

### 4.2. What is a Successful Test? Partially Successful? Failure?

- To determine the level of success for each of the tests that will be performed, they will all follow the success levels listed in each of the use cases. This is because the use cases are going to be what is tested.

### 4.3. How will the Overall Success of the Product be Determined?

- If all use cases go according to plan, and they all result in full success, the overall success of the product will be a full success.
- If only some of the use cases result in full success, the overall success of the product will be a partial success.
- If the test results do not match any of the two descriptions listed above, the overall success of the product will be a failure, or no success.

### 4.4. Scope of Testing

- The testing process of this internal assessment will test both in terms of hardware and software. If the product is physically incapable of performing in any of the use cases, it will be a hardware issue. Therefore, the testing in this internal assessment will only test in terms of the use cases of this internal assessment. This test will not test through an ad-hoc method, this is to ensure that all functions of the vending machine works perfectly before potential customer use. The tests will test white box, as any error detected could be found in the software and corrected for. This is because if I test black box, and it results in an error, there will be no way of finding where the error is in the program and therefore, not knowing where to fix it.

## 5. Time Frame of Testing

### 5.1. Amount of time needed for 1 round of testing

- If all hardware works perfectly, the amount of time needed for 1 round of testing will be 5 minutes. This ensures that the 30 seconds that are needed to pass for use case 2 and 3 will be included and that the two use cases can be tested. 5 minutes is enough, as there will be enough time for the “customers” action to be performed, enough time for the error to be observed, and enough time for the error to be noted down.

### 5.2. Rounds of Regression Testing that will be Performed

- There will be a total of n+1 rounds of regression testing that will be performed; with n being the number of use cases. This is because each use case will require all previous use cases to have full success in order to achieve full success; unless the use case is a part of a “split”, where the vending machine can potentially do two different things depending on what the customer does. Therefore, the maximum number of regression testing that is needed to be performed will be n+1.

# Criteria C : Development

## Developing the Product:

| |
|---|
| Process Journal Link:<br>https://docs.google.com/document/d/18Ei0_iSgaTZm6RPbpcE8HRa0fYdQjniHRo2gMjH8ww0/edit?usp=sharing |

# Criteria D : Functionality and Extensibility of Product

## Evidence of Functionality:

Video Link:
https://youtu.be/L2xFR_tL1xk

## Extensibility:

Vending Machine Extensibility Testing. (Shows how the items' stock number can be changed in both the vending machine interface and the csv file itself. The video also shows how it is possible to change the item's name and price through changing data within the csv file.)

https://youtu.be/76Gi95Dh9vQ

NOTE: If the program is to be run on one's computer, all three files will need to be installed without changing the name of the files.

Vending Machine Python Code:
https://drive.google.com/file/d/1L9TZL1un36dHCYIdDAInaKMzst7d9Pro/view?usp=sharing

Vending Machine csv Database File:
https://drive.google.com/file/d/1L_16VkKeFvXtNVZfKWYG9uQKFBtSpdGx/view?usp=sharing

Vending Machine Background Image:
https://drive.google.com/file/d/1s1KA03IoXpnVfxHfuqPzgIFkGoKnYY4I/view?usp=sharing

# Criteria E : Evaluation

## Evaluation of the Product:

I have selected 2 people to evaluate my product, both being MYP students in grade 10. They evaluated my product's success criteria as seen in the appendix.

When considering **function**, my evaluators have stated that the specifications were only partially met. This is mostly due to how the final product is not hardware based, so the specifications of "linking to the arduino board" and "refund system that dispenses physical coins back" is not possible. However, the product did allow for the dispensing of the user's selected item.

When considering **ease of use**, the final product is evaluated to be partially successful. The final product's buttons are easily interpreted, and the messages on the vending machine are easy to read. However, because the final product is only a digital simulation, it was not possible to refund physical money.

When considering **customers**, the product dispenses the selected item when the customer pays enough money but does not allow the customer to pay manually. Therefore, when considering customers, my product is only partially successful.

When considering **components**, the product is partially successful. This is because the product's buttons would display the selected item and the user will receive the item, however, due to the fact that the final product is not hardware based, a refund option and a currency dispensing area were not constructed.

When considering **aesthetics**, my evaluators stated that they did not find the product "fitting with the school's architectural color theme." This is because the product is only a simulation, and is not three dimensional, therefore it did not look realistic. My simulation's background is also evaluated to not match the color theme of my school.

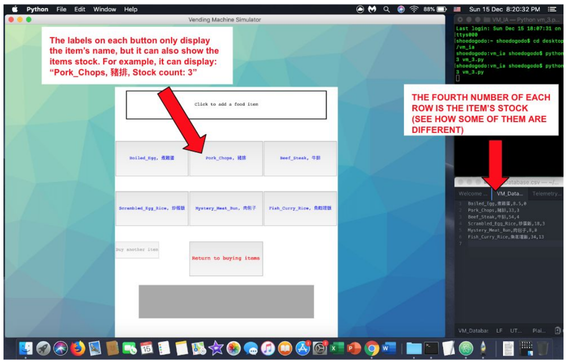
Finally, when considering **materials**, the product was evaluated to be partially successful. This is because while the product's software is coded in python 3.7, there is no hardware, so the final product does not match the original specifications of using hardware along with software.

# Recommendations for Further Development:

When considering recommendations, this paper will only consider future development for the software of the current product. This is because while a hardware based product is specified in the specifications, changing the current product to be hardware based would completely overhaul most of the current product, and a hardware based product might as well be a different product compared to the current software based simulator product.

### Recommendation 1: Display the Stock Count of Items while Adding Stock

As seen in the picture on the right, the current product does not display the stock count of the item. By showing how much stock is in each item, the person adding the stock can check how much of each item is in the vending machine without opening the csv file.



The labels on each button only display the item's name, but it can also show the items stock. For example, it can display: "Pork_Chops, 豬排, Stock count: 3"

THE FOURTH NUMBER OF EACH ROW IS THE ITEM'S STOCK (SEE HOW SOME OF THEM ARE DIFFERENT)

```
Last login: Sun Dec 15 18:07:31 on
ttys000
shoedogodo:~ shoedogodo$ cd desktop
/vm_ia
shoedogodo:vm_ia shoedogodo$ python
3 vm_3.py
shoedogodo:vm_ia shoedogodo$ python
3 vm_3.py
```

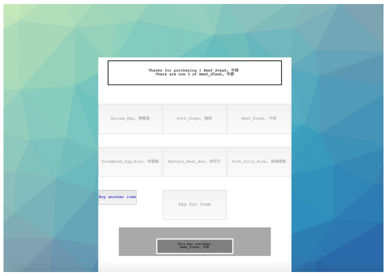
```
Boiled_Egg,煮雞蛋,8.5,0
Pork_Chops,豬排,33,3
Beef_Steak,牛排,54,4
Scrambled_Egg_Rice,炒蛋飯,18,3
Mystery_Meat_Bun,肉包子,8,0
Fish_Curry_Rice,魚咖喱飯,34,13
```

### Recommendation 2: Change Item Names and Item Prices Within the Vending Machine Interface

By allowing a user to change an item's name and price from within the vending machine simulator itself, the simulator will be more self-dependent. This means that the user no longer needs to go into the vending machine csv file to edit an item's name and the item's price. By implementing this recommendation, the vending machine simulator will be able to simulate a vending machine changing items and changing the item slot's data without needing to edit the internal software.

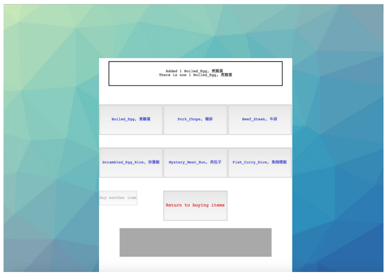
### Recommendation 3: Buying and Adding more than One Item at Once (From Within the Simulator)

As seen on the two images on the right, the current program can only add or buy one item at once. By allowing the user to buy or add more items at once, it would make the product more user friendly and more efficient of a vending machine.



```
Thanks for purchasing 1 Beef_Steak, 牛排
There are now 3 of Beef_Steak, 牛排
```

```
This box includes:
Beef_Steak, 牛排
```



```
Added 1 Boiled_Egg, 煮雞蛋
There is now 1 Boiled_Egg, 煮雞蛋
```

# Appendix / Appendices

**Feedback from Peers:**

| Specification | Description | Feedback: |
|---|---|---|
| Function | - Program on python that would link to the arduino board and perform the vending machine specifications below.<br>- Software that controls hardware that dispenses a selected item that the user chooses when the price is paid.<br>- Refund system that dispenses the coins put in back to the user. | **Aiden:** This specification was not met very well because the code did not support the arduino/hardware. However, the code does simulate the vending machine process pretty well.<br>**Austin:** Does not link to arduino. However, the code simulates a vending machine well. However, it doesn't refund so thats not good. |
| Ease of Use | - Buttons are easily interpreted for what they do<br>- Monitor information is easily read<br>- Refunded money is easily collect | **Aiden:** Partially, because the buttons are easily interpreted, and the monitor information can also be easily read. However, because it is only a simulation, therefore there are no coins to be actually collected.<br>**Austin**:Can't collect money cause there's no vending machine, and can't refund. But the buttons and monitor word. |
| Customer | - Must be able to deliver an item to the audience when they pay enough for it | **Aiden:** Partially, because it simulates that the customer pays enough already, but does not allow the customer to pay manually. |
| Size | - Must be able to stand still and is accessible by a client when they are standing up | **Aiden:** Does not meet the specification, because it is only a simulation with no |

| | | |
|---|---|---|
| | | actual hardware. |
| Components | - Buttons that will cause a monitor to display the price of the selected item<br>- Currency dispensing area, where the user can put currency and the currency dispensed is displayed on a monitor<br>- When the desired price is reached by the user, the user will receive the item that they selected on the vending machine<br>- Refund button that will dispense back money to the user | **Aiden:** Buttons causes the monitor to display price of item, but there is no currency and manual input from the user.  There is also no refund button, therefore this specification is partially completed.<br>**Austin**: Nothing was built, so you failed. |
| Aesthetics | - Must fit in with the schools architectural color theme depending on where it is put | **Aiden:** Does not fit with school's architectural color theme, because it is only a simulation, and I believe that the background does not match the color theme.<br>**Austin**: There is no color scheme because there is no vending machine. But the monitor is white and uses blue so at least the interface is ok. |
| Materials | - The program which will control the vending machine will be coded in Python, version 3.7<br>- Arduino board<br>- Buttons to click and choose items<br>- Display monitor which displays information<br>- Motor to dispense items from the vending machine<br>- Materials to construct a prototype vending machine, such as cardboard | **Aiden:** It was programmed on python, version 3.7.  It does not use arduino, but has buttons, display monitors.  However, because it is only a simulation it does not have a motor and does not use materials to construct the prototype vending machine.<br>**Austin:** You used python, but because you didn't build anything you don't use any materials so who knows. |

# Works Cited

Zach. “Vending Machines and its functions.” *The Green Book*, 13 May 2015, http://www.thegreenbook.com/vending-machines-and-its-functions.htm.

Knight, Matthew. “Uses of Vending Machines.” *bizfluent*, Leaf Group Ltd. 25 September 2017, https://bizfluent.com/13656955/uses-of-vending-machines.

Morton, Kari. “Vending Machines with a Different Purpose.” *BUSINESSLINK*, Business Link. 26 October 2016, https://businesslink.ca/blog/vending-machines-different-purpose/.