1. **Title: Vending machine System**

**2. X-Team Number: 202**

**3. Team Members:**

Ivan Hu [ilhu@wisc.edu](mailto:ilhu@wisc.edu)

ELYSE ASMUS [ecasmus@wisc.edu](mailto:ecasmus@wisc.edu)

Zuhua Cao [zcao65@wisc.edu](mailto:zcao65@wisc.edu)

HARMAN CHAWLA [hschawla@wisc.edu](mailto:hschawla@wisc.edu)

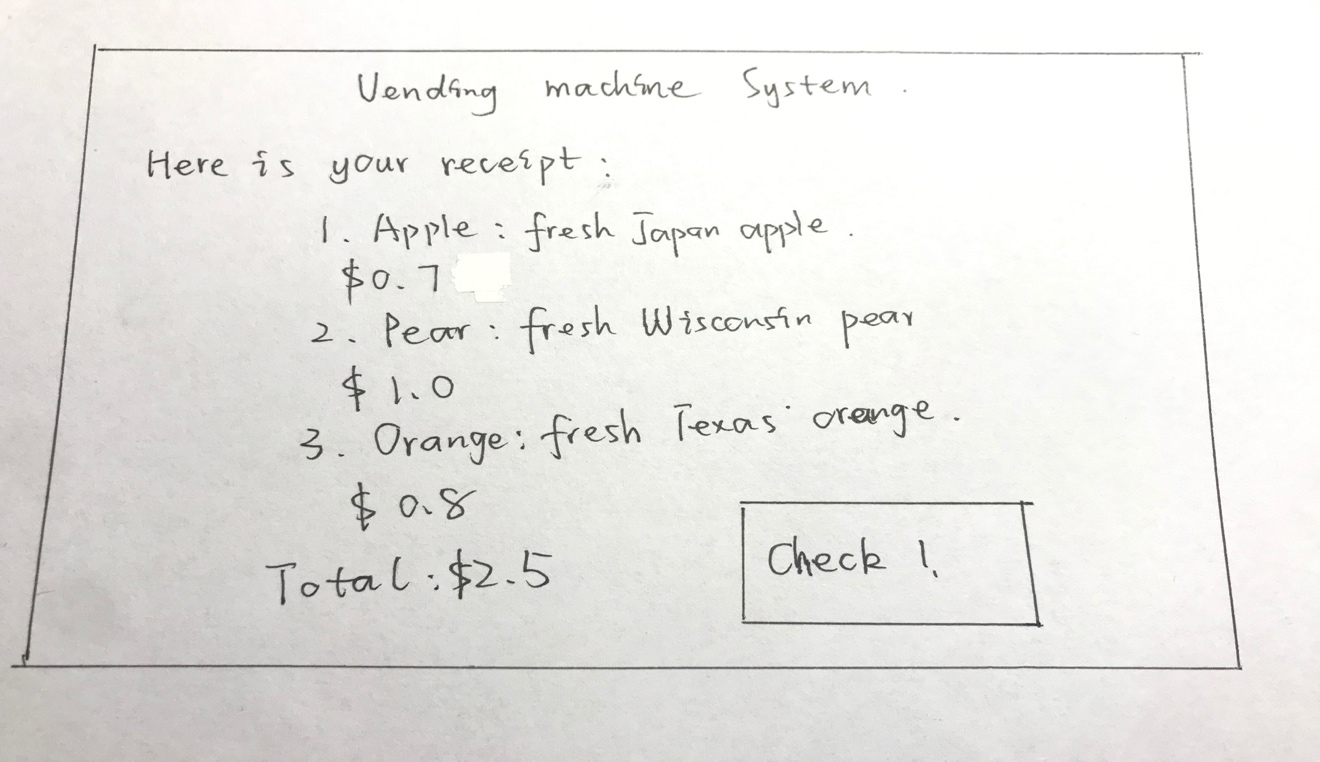
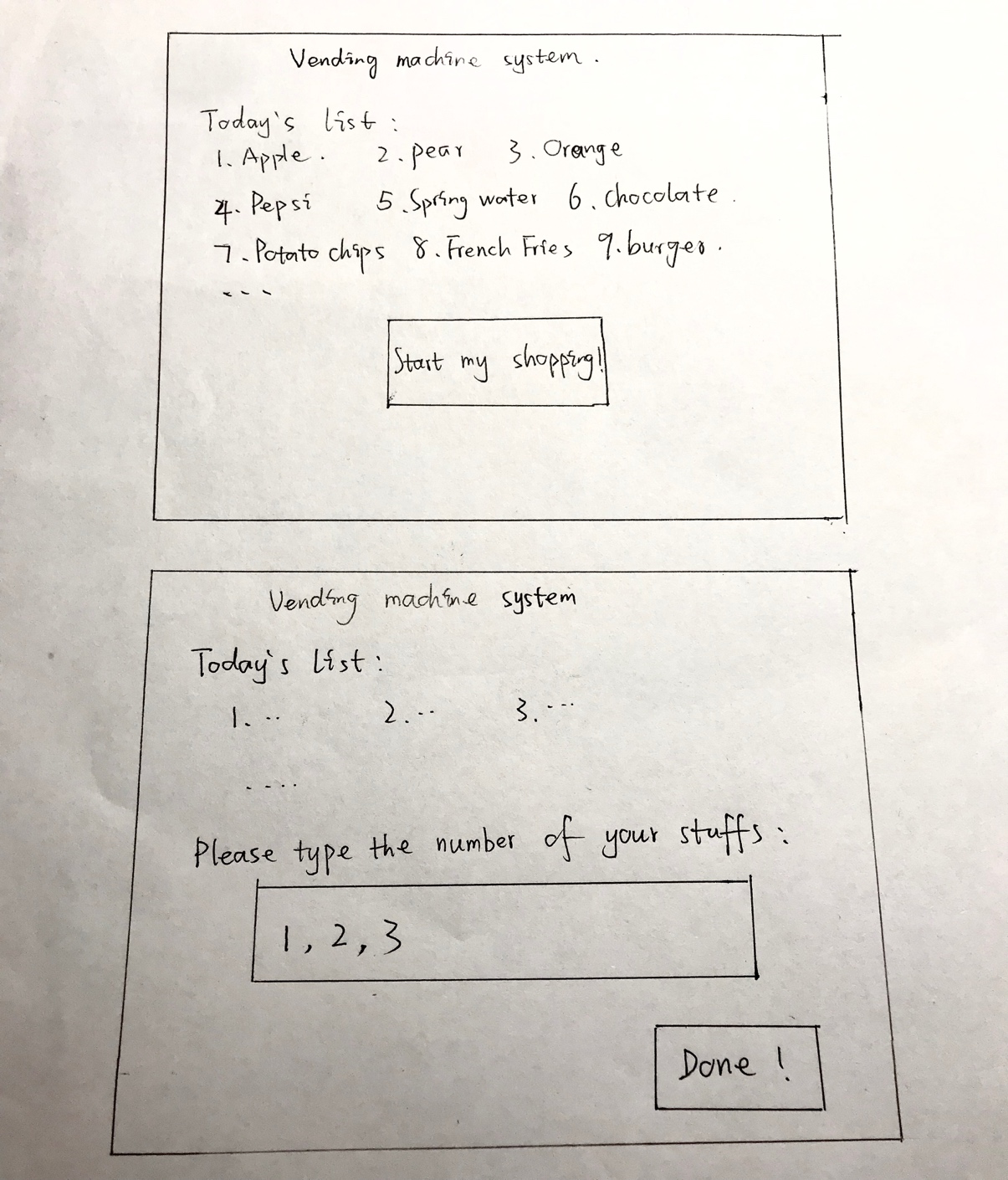
**4. Problem:**

As a vending machine system, we need to store the names and prices of many goods. Therefore, we must need an efficient data structure to store these data. And when a customer wants to buy some of them, we’ll quickly show them the descriptions and total prices of the stuffs. As for back-end, it needs to store all of the descriptions and prices of the goods, and it need be quick enough to search for the data we want. As for front-end, we need to provide the customer an interface to show the names of the goods, and they need to type the names of the aim goods. Then the interface will show them if the items exist and the description and total prices of the things.

**5. Primary stakeholder:**

The primary stakeholder will be the all of the customers in the shopping mall or convenience store.

**6. Graphical User interface**:

First, we need to click an exe file to start/call our user interface. At the first page, the front-end will show user a list of names of all goods. Then, the user will click “start my shopping” button. Then it pops an input box, which the user could type the number of the goods they want. After that, the user will click the “done” button. So, the front-end will call the back-end to solve the problem. After that, it will show the description of the customer’s receipt and the total price of the goods. Now the user just needs to click check, so they could complete the shopping. 

**7.Data Structure**:

A binary search tree (TreeMap) will be used to store the catalog of items. Every node will store a key, a description string and a value, the key will be the number of the node and the value will be the price of it. The user’s selection of items will be stored as an ArrayList.

*Prototype:* For a small catalog of items (at most ~20), an ArrayList is sufficient to store them.

**8. Input Data Format:**

The user types a line of numbers separated by commas, representing the items s/he wants to order.

For example: 5, 1, 3, 7, 8, 12, 13

9. **Output Example:**

The program will return several lines of strings:

For example:

Here is your receipt:

1. Apple, fresh Japan apple

$0.70

1. Pear, fresh Wisconsin pear

$1.00

1. Chocolate, Dove chocolate

$1.50

Total: $2.50

**10. Milestones:**

We could firstly complete data structure part, which is built by binary search tree. So, it will store all of the items’ prices and information. This is also the most important part of this project. Then, we could assign the input interface part to one of teammates, then assign the output interface to another team mate. The left two will work on the data structure part.