Lamess Kharfan
Student ID: 10150607

A4- Vending Machine Design

I've designed my vending machine as a layered architecture, with the Business Logic class at the top layer, the payment façade, communication façade, and the products façade at the middle layer, and the hardware façade at the bottom layer. The Business Logic class and the Hardware façade never directly interact with each other, instead they communicate by firing events, which the facades listen to, and the facades also fire events and communicate by method calls with the business logic class. The three façade classes also never communicate with each other directly, they communicate by firing events and by method calls to send the relevant data to and from one another through the business logic class.

Each façade centralizes payment, communication and products, and only contain information and methods that deal with what they are for:

The payment façade is designed to work by listening for events that are about payment being added to the machine, such as the CoinAccepted event from the Coin Slot piece of hardware of the vending machine. Once the payment is added, the payment façade will get the value of the payment inserted and convert it to cents, and add it to the total funds available. It also has methods for calculating the amount of change to dispense, storing payment, and a method for business logic to get the total funds added so far.

The communication façade is designed to work by listening for events about selections being made, such as the Press event from the selection buttons pieces of hardware of the vending machine, and can also communicate with a customer by displaying messages. When a selection is made and an event fires, the communication façade's event handler fires another event, notifying the business logic a selection has been made, and the business logic will make decisions about weather or not payment is sufficient, if change needs to be dispensed, and if the selected product should be dispensed. The selectionEventArgs class is for passing arguments about the selection to the business logic class so it can appropriately evaluate what product needs to be dispensed, know its cost, and its index in the machine. This facade also has a method for displaying a message about insufficient funds after a button press. More messages could be later added.

The products façade listens to the ProductRackEmpty event for all of the product racks, if one becomes empty, an event called ProductRackEmpty is fired, which is subscribed to by business logic, and business logic tells the communication façade to display a message stating that there are no more of a certain product available. This façade also dispenses the appropriate product to the delivery chute based on the selection the customer made using the selection buttons. Weather or not a product is dispensed is decided in the business logic class.

I've designed this so that each façade focuses on payment, communication, or products and nothing else. The business logic class does all of the communication between facades and deals with the logic of the vending machine, this way everything is central in regards to payment, communication and products, so when we want to add hardware for other methods of communication and payment, it is easy for the vending machine to be extensible and incorporate these things by adding the hardware to the hardware façade and listen for events for the new hardware in the facades, and the business logic will be able to handle communication between the facades already. The products façade is extensible because we are easily able to change what is dispensed by the machine by giving the type of product when we configure the vending machine.

For things like loading coins, loading product, extracting from the delivery chute, and unloading the vending machine, I have not included anything in the facades for these because they all deal with direct interaction with the hardware façade, so I believe the other facades do not need to be involved.