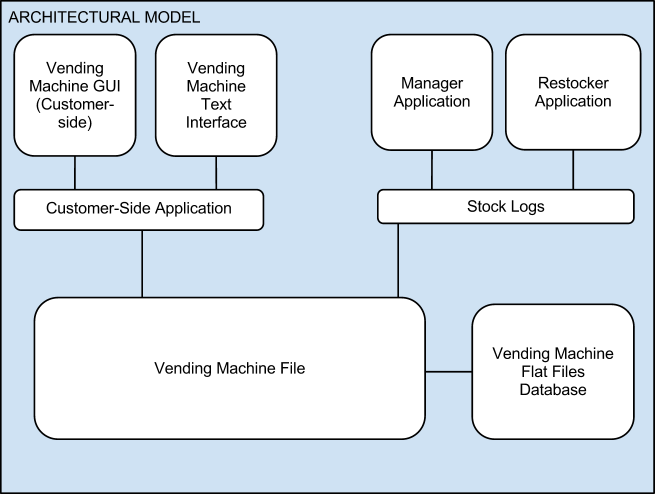
**Product Design**

|  |  |
| --- | --- |
| **Team** | Group S361-02A **Chalupas** |

# Architectural Model

This diagram represents the major subsystems of the product. The customer-side section includes the vending machine application as well as its user interfaces. When customers purchase items, the flat file for the vending machine is updated with current items and sales executed. Each vending machine file is part of a database that houses all the files; the manager has access to this. The manager application forwards stocking information to the restocker application. The restocker replies with what was restocked, and the vending machine file is updated with current items.



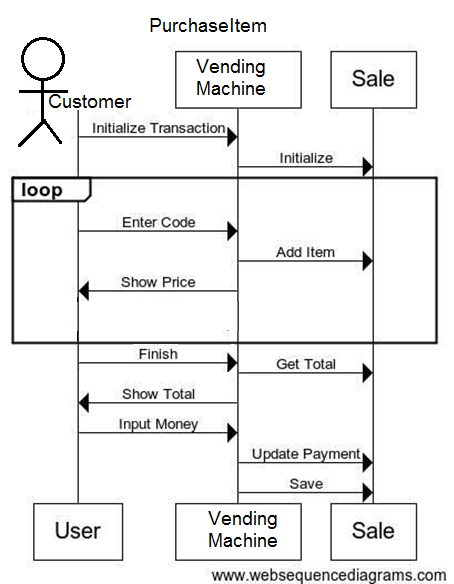
**Components and Functions**

|  |  |
| --- | --- |
| Vending Machine Application | Component State   * Maintains a collection of items in the vending machine.   Component behavior   * Execute sales -> update its collection of items * Simulates time |
| Restocker Application | Component State   * Current Stock of Current Vending Machine * List of Expired Items in Current Vendingg Machine * List of Recalled Items in Current Vending Machine * List of Purchased Items in Current Vending Machine   Component behavior   * Removes Recalled or Expired items in the current vending machine. * Rearranges the Vending Machine, if demanded by Manager * Refills Current Vending Machine with fresh items. * Reports what was actually stocked * Lockout customer and Manager app |
| Manager Application | Component state   * List of all vending machines currently in operation   Component behavior   * Reads Analytical data from Vending Machine * Reports analytical data * Rearranges item placement based on sales * Sends updated data back out to Restocker application. * Can request a Restocker to a location, if stocks are too low. * Updates a vending machine’s status remotely. * Can add and remove vending machines from the network. * Recalls items from all vending machines across the network |

# Class Diagram(s)

# 

# Sequence Diagram(s)



# Design Rationale

*Currently chosen concept is in bold.*

* Data management
  + Concept A: Store all data for every vending machine in a single file
    - Pros: Easy to manage a single file
    - Cons: If any of the applications needs to make a change to any vending machine they have to lock the entire. Corruption affects every vending machine’s data, and only one machine can alter it at once.
  + **Concept B: Store data for each vending machine in a separate file. Store vending machine metadata for all vending machines in a single file.**
    - Pros: Modifying one vending machine’s information does not require locking every vending machine’s data. Corruption of one vending machine’s data does not affect any other data. Locking can be managed in a separate layer (metadata) than the actual data. Can run multiple vending machines at once, as you would in the real world.
    - Cons: Lots of files to keep track of. More complicated coordination.
* VendingMachine & related object representations
  + Concept A: Vending Machine holds many, many objects for each individual item, using many containers as slots in a 2D array
    - Pros: All items hold their respective data, easier to determine item data. Can hold multiple object types in a single slot
    - Cons: Lots of files to keep track of. More complicated coordination.
  + **Concept B: Vending Machine holds custom slot objects, which hold a array list of expiration dates to represents individual items. Slot objects hold information related to product name, price, etc.**
    - Pros: Less files to manage, easier to keep track of objects
    - Cons: Cannot hold more than one type of object in a single slot

**Design Update Log**

* Updated UML to reflect the current changes made to the overall design
  + More functionality added to the Vending Machine, in order to handle proper item selection and dispersing.
  + Fixed names of object to reflect their in-app counterparts
  + Highlighted relationship between Restocker and Manager Apps (Request Handling and Responding)
* Updated Sequence Diagram to more accurately display the flow of the PurchaseItem use case.
* Updated the Comonent Description. Added missing functionality to the most updated version, including:
  + Component Behavior of Restocker -> Reports the stock of the vending machine
  + Component Behavior of Manager -> Recalls items throughout network, reports analytical data, updates current status of vending machine to the network
* Updated Design Rationale to further describe our base design