**Project Proposal Template**

**Student Name(s):** Michael Shi

**Proposed System Under Test (SUT):** Vending Machine

**Link to SUT Source Code:** https://hcitang.github.io/seng301/#!a3.md

**SUT Size:** ~1100 Lines of Code

# SUT Description

The system (written in C#) was given as an assignment for a software engineering course. The system simulates the logic and capabilities of a vending machine through the use of multiple classes and event handling.

# State of SUT

The system was given as an assignment in the Winter of 2017. It is no longer being actively developed. There is no formal specification and the source code has many different classes.

# Attributes

* **Accurate:** The vending machine accurately keeps track of the stock, funds, and delivers the correct product and amount of change
* **Feature-rich:** The vending machine has different features and components to accurately simulate a vending machine
* **Object-oriented:** The vending machine code uses aspects of object-oriented programming and event-driven programming
* **Organized:** The vending machine’s components is organized into meaningful classes/names
* **Testable:** The vending machine’s code is easily testable to see if the output/result is correct

# Components

* **VendingMachineLogic:** Calculates and updates the available funds, stock, and of the vending machine. Determines what happens when a selection button is pressed.
* **VendingMachineFactory:** Used to create vending machine objects, includes all the features (methods) and capabilities of each vending machine.
* **VendingMachineStoredContents:** Lists containing the amount of coins, funds, and pop in the vending machine.
* **Hardware Packages:** Packages that contain and define the hardware components in the vending machine. They are split into different classes (DeliveryChute, CoinRack, Display, SelectionButton).

# Capabilities:

# VendingMachineLogic is Accurate: The logic class accurately checks and updates the stock/funds after a selection.

# VendingMachineLogic is Accurate: The logic class accurately calculates the change delivered after a selection.

# VendingMachineLogic is Object-oriented: The logic class uses objects and events while performing calculations.

# VendingMachineLogic is Testable: The logic is within a class of its own, so it is testable and is easy to change. The results and output make testing straightforward.

# VendingMachineFactory is Object-oriented: The factory class makes use of objects.

# VendingMachineFactory is Feature-rich: The factory class contains features regarding the interaction with the vending machine.

# VendingMachineStoredContents is Organized: The class divides the contents that are stored and moved in the vending machine.

# VendingMachineStoredContents is Testable: The code is easily testable as there are Lists that keep track of all the stored contents.

# Hardware Packages are Feature-rich: The packages contain many classes that represent the hardware of a vending machine.

# Hardware Packages are Organized: The packages are organized in a manner that is easy to navigate with meaningful names.

# Hardware Packages are Object-oriented: The hardware classes respond to events.

# Capabilities Count:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Accurate | Feature-rich | Object-Oriented | Organized | Testable |
| *Logic* | 2 | 0 | 1 | 0 | 1 |
| *Factory* | 0 | 1 | 1 | 0 | 0 |
| *StoredContents* | 0 | 0 | 0 | 1 | 1 |
| *Hardware* | 0 | 1 | 1 | 1 | 0 |

# Basic Testing:

For the code-related attributes, I plan to write unit/integration tests to test and confirm each capability. This system has not been heavily tested and might require modifications to the codebase. It is expected that the basic testing will be a large component. In general, I will be performing different black box testing and white box testing techniques to ensure correctness throughout the system.

# Advanced V&V:

I want to use grammar-based techniques and plan to design a grammar that describes a subset of a particular vending machine. As the course progresses, I will look at the advanced V&V techniques taught in class and discuss with the professor in order to make an informed decision on how to approach applying these techniques to my system. Due to the expected amount of basic testing needed, the Advanced techniques will most likely consist of a small portion of the project.