**The following is part02 requirements with a short description below each to help QUB review**

***2.1 Add a (hidden) menu option to the main application that will enable a user (engineer) to access a (password protected) maintenance menu. This menu should provide the following options:***

*THIS MINTENANCE MENU CAN BE ACCESSED WITH:*

*SECRET CHOICE CASE: 99*

*Username: u*

*Password: p*

* ***view all data related to the vending machine, appropriately formatted***

This can be accessed by choice 1 on the test maintenance menu class. It calls the getSystemInfo() method from VendingMachine.class which was on the UML provided. It returns data related to the vending machine.

* **reset the machine to factory default (empty of money & items)**

This can be accessed by choice 7 on the test maintenance menu class. It calls the reset() method from VendingMachine.class which was on the UML provided.

* **switch status between VENDING\_MODE and SERVICE\_MODE**

This can be accessed by choice 8 on the test maintenance menu class. It calls the setStatus(status) method from VendingMachine.class which was on the UML provided. The user is asked to select option 1 or 2 which in turn changes the objects vmStatus.

* **a facility to add a new VendItem or to restock an existing item**

This can be accessed by choice 9 and 10 on the test maintenance menu class. They call the addNewItem(VendItem) method from VendingMachine.class which was on the UML provided and restock(itemId,restockAmmount) which is a method I added. They ask the user to input relevant data for parameters then carry out the method.

**2.2 Add instance data to the VendingMachine class to keep track of coin denominations it contains (in addition to their total value), reporting on change given for purchases in terms of coin combinations as well as total value. If change can’t be delivered (due to coin availability), this should be reported.**

The code for this additional feature can be found near the bottom of VendingMachine.class. I created additional instance data at the top keeping track of coins in system, they are affected every time a coin is inserted and system is exited (returnChange() is called). System will calculate coin change due to availability. If change can’t be fully given machine set to service mode and user let know.

**2.2 Add behaviour to the system to enable the state of a VendingMachine to be stored/retrieved to/from a (CSV) file. This should be restored when the application starts and stored when the application is exited.**

This feature is implemented using the new classes VendingMachineRead and VendingMachineWrite. It also uses the VendingMachineState.csv found in my submission.

\*please correct the csvInPath and csvOutPath at top of VendingMachineTestMenuCSV with the directory of VendingMachineState.csv