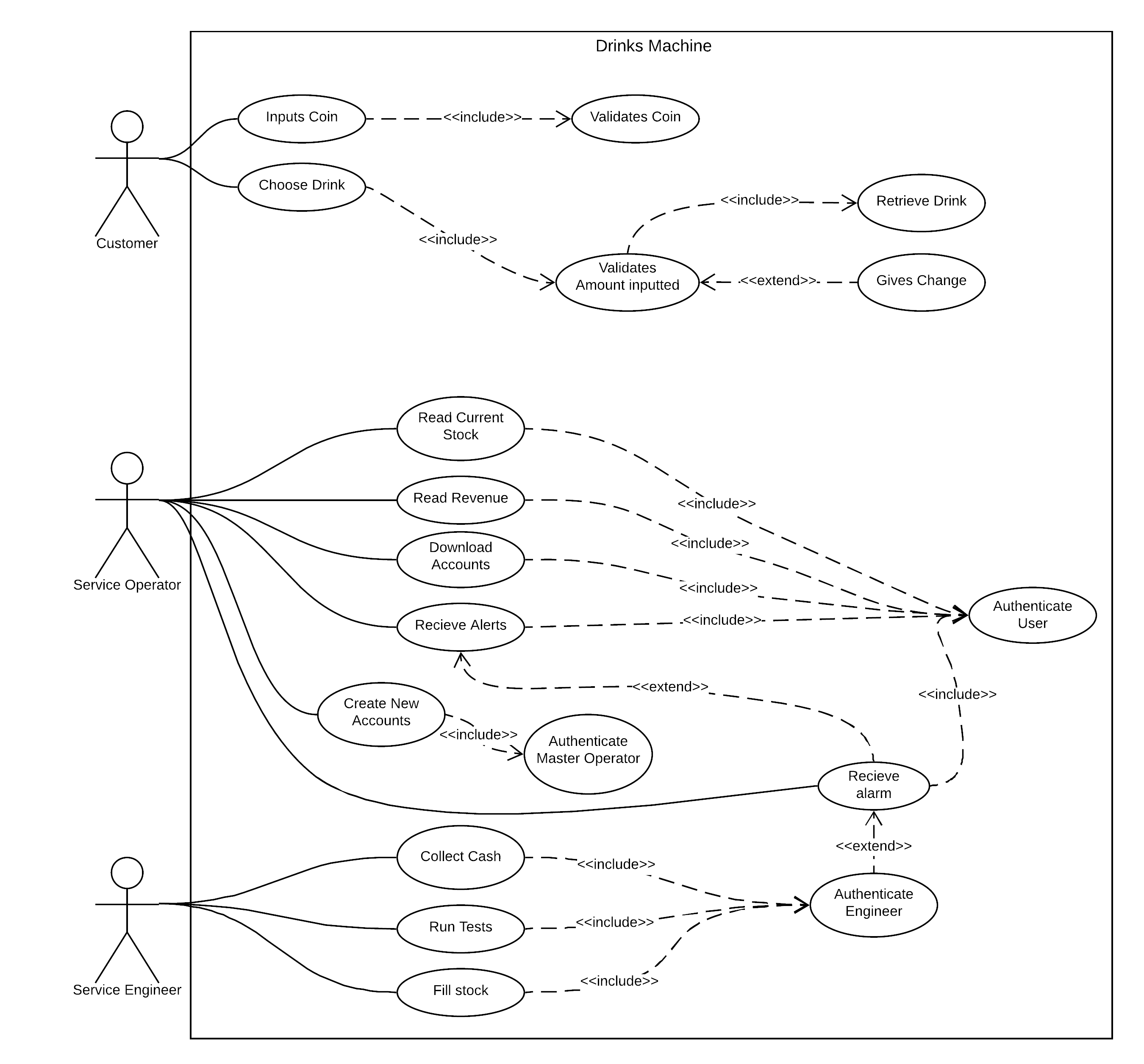
# Assignment 1.1 – COMP201

## Task 1

### Use-case diagram



### Use-case description

|  |  |
| --- | --- |
| **ID** | **UC1** |
| **Name** | Inputs coin |
| **Description** | The user sees the item he wants and puts in the coins to make the |
| **Pre-conditions** | The machine is on, fully functional and ready to be used. |
| **Event flow** | 1. If system is in a state where it doesn’t require the user to input coins, return the coins 2. Take coin from user and add it to balance |
| **Post-condition** | Coins inputted |
| **Includes** | UC2 (Validates Coin) |
| **Extensions** | - |
| **Triggers** | Balance incremented |

|  |  |
| --- | --- |
| **ID** | **UC2** |
| **Name** | Validates Coin |
| **Description** | To check if the following coins are allowed: 1p, 2p, 5p, 10p, 20p, 50p, £1, £2. If it’s anything that’s not those coins, return them to the customer. |
| **Pre-conditions** | Coins need to be inputted |
| **Event flow** | 1. If it’s not a valid coin then return it to the customer |
| **Post-condition** | Coins are now validated |
| **Includes** | UC1 (Inputs Coin) |
| **Extensions** | - |
| **Triggers** | Returns coin if not allowed |

|  |  |
| --- | --- |
| **ID** | **UC3** |
| **Name** | Choose Drink |
| **Description** | The user chooses the drink that they want |
| **Pre-conditions** | The user should have the correct balance |
| **Event flow** | 1. Checks sufficient balance 2. If balance is sufficient, the drink is chosen |
| **Post-condition** | Drink selected |
| **Includes** | UC4 (Validates Amount inputted) |
| **Extensions** | - |
| **Triggers** | Retrieving Drink, Change being given |

|  |  |
| --- | --- |
| **ID** | **UC4** |
| **Name** | Validates Amount inputted |
| **Description** | Validates the amount of coin inputted is correct comparing from the chosen drink. |
| **Pre-conditions** | The drink is chosen |
| **Event flow** | 1. The drink is chosen 2. If correct amount is not inputted, quit and request more coin input |
| **Post-condition** | Amount is validated |
| **Includes** | UC5 (Retrieve Drink) |
| **Extensions** | UC6 (Gives Change) |
| **Triggers** | Gives the user the drink or change if necessary |

|  |  |
| --- | --- |
| **ID** | **UC5** |
| **Name** | Retrieve Drink |
| **Description** | Give the user the drink that they selected |
| **Pre-conditions** | The coins and balance are validated |
| **Event flow** | 1. Drink is chosen 2. Coins and balance are validated 3. Given the drink |
| **Post-condition** | Drink is given to the user and finished service for the customer |
| **Includes** | UC4 (Validates Amount inputted) |
| **Extensions** | - |
| **Triggers** | The user balance is set to 0 |

|  |  |
| --- | --- |
| **ID** | **UC6** |
| **Name** | Gives Change |
| **Description** | The user is given change |
| **Pre-conditions** | The coins and balance are inputted and the user exceeded over the required amount |
| **Event flow** | 1. Drink is chosen 2. Coins and balance are validated. 3. Balance is over required limit 4. Change is given to the user |
| **Post-condition** | Now ready for next customer to come |
| **Includes** | - |
| **Extensions** | UC4 (Validates Amount inputted) |
| **Triggers** | The user balance is set to 0 |

|  |  |
| --- | --- |
| **ID** | **UC7** |
| **Name** | Read Current Stock |
| **Description** | This state reads the stock |
| **Pre-conditions** | The machine is working fine and that the service operator is authenticated |
| **Event flow** | 1. If the service operator is authenticated then authenticate him 2. If the service operator authentication fails then reject their access. 3. The current stock level would be read |
| **Post-condition** | The user is now definitely authenticated. |
| **Includes** | UC12 (Authenticate User) |
| **Extensions** | - |
| **Triggers** | - |

|  |  |
| --- | --- |
| **ID** | **UC8** |
| **Name** | The revenue is read |
| **Description** | The revenue is read to the service operator |
| **Pre-conditions** | The machine is working fine and that the service operator is authenticated |
| **Event flow** | 1. If the service operator is authenticated then authenticate him 2. If the service operator authentication fails then reject their access. 3. The Revenue made would be read |
| **Post-condition** | The user is now definitely authenticated. |
| **Includes** | UC12 (Authenticate User) |
| **Extensions** | - |
| **Triggers** | - |

|  |  |
| --- | --- |
| **ID** | **UC9** |
| **Name** | The accounts are downloaded |
| **Description** | The accounts are downloaded from the machine to see what sales happened at what time and how much money people spent etc. |
| **Pre-conditions** | The machine is working fine, no other actors are using the machine and that the service operator is authenticated |
| **Event flow** | 1. If the service operator is authenticated then authenticate him 2. If the service operator authentication fails then reject their access. 3. The accounts are now downloaded |
| **Post-condition** | The user is now definitely authenticated and the accounts have now been downloaded so there would be no need to download them again |
| **Includes** | UC12 (Authenticate User) |
| **Extensions** | - |
| **Triggers** | - |

|  |  |
| --- | --- |
| **ID** | **UC10** |
| **Name** | Receive alerts |
| **Description** | Alerts can be received from the machines such as low stock etc. |
| **Pre-conditions** | The machine is working fine, no other actors are using the machine and that the service operator is authenticated |
| **Event flow** | 1. If the service operator is authenticated then authenticate him 2. If the service operator authentication fails then reject their access. 3. The alerts have been received |
| **Post-condition** | The user is now definitely authenticated and the most recent alerts have been read and will not appear again. |
| **Includes** | UC12 (Authenticate User) |
| **Extensions** | UC11 (Receive alarm) |
| **Triggers** | - |

|  |  |
| --- | --- |
| **ID** | **UC11** |
| **Name** | Receive alarm |
| **Description** | Alarms will be rung for the server if someone has triggered the alarm. E.g. failure of authentication of the engineer. |
| **Pre-conditions** | The operator must be logged in to receive the alarm |
| **Event flow** | 1. If the service operator is authenticated then authenticate him 2. If the service operator authentication fails then reject their access. 3. The alarm rings |
| **Post-condition** | The alarm has gone off, the user is definitely logged in because they received the alarm. |
| **Includes** | UC12 (Authenticate User) |
| **Extensions** | UC10 (Receive alert)  UC19 (Authenticate Engineer) |
| **Triggers** | - |

|  |  |
| --- | --- |
| **ID** | **UC12** |
| **Name** | Authenticate user |
| **Description** | The user will have to sign in with a password. If the user is successful, they won’t have to login again for the rest of the session |
| **Pre-conditions** | The service operator machine will have to work |
| **Event flow** | 1. If the service operator is authenticated then authenticate him 2. If the service operator authentication fails then reject their access. |
| **Post-condition** | The user is now logged in and won’t have to login again for the rest of the session as they’re already authenticated |
| **Includes** | UC7 (Read Current Stock)  UC8 (Read Revenue)  UC9 (Download Accounts)  UC10 (Receive Alerts)  UC11 (Authenticate User) |
| **Extensions** | - |
| **Triggers** | - |

|  |  |
| --- | --- |
| **ID** | **UC13** |
| **Name** | Create New Accounts |
| **Description** | Can create new accounts for the service engineer or operator |
| **Pre-conditions** | The machine is fully functioning and if the master operator is logged in, they don’t have to login |
| **Event flow** | 1. If the master operator is authenticated then authenticate them 2. If authentication is successful then they’re logged in and won’t have to enter their password again for the session. 3. If the master operator authentication fails then reject their access. 4. They have the option to create new accounts |
| **Post-condition** | The user is now logged in and won’t have to login again for the rest of the session as they’re already authenticated. Also a new account would be created |
| **Includes** | UC14 (Authenticate Master Operator) |
| **Extensions** | - |
| **Triggers** | - |

|  |  |
| --- | --- |
| **ID** | **UC15** |
| **Name** | Authenticate Master Operator |
| **Description** | The user will have to enter a password to authenticate themselves. If they fail then they’re not authorised. |
| **Pre-conditions** | The machine is fully functional and the user is wanting to create a new account. |
| **Event flow** | 1. If the master operator is authenticated then ignore this state 2. If the master operator authentication fails then reject their access. 3. If authentication is successful then they’re logged in and won’t have to enter their password again for the session. |
| **Post-condition** | The user is now logged in and won’t have to login again for the rest of the session as they’re already authenticated. Also a new account would be created |
| **Includes** | UC13 (Create New Accounts) |
| **Extensions** | - |
| **Triggers** | - |

|  |  |
| --- | --- |
| **ID** | **UC16** |
| **Name** | Collect Cash |
| **Description** | Where the engineer collects the physical cash from the machine |
| **Pre-conditions** | No customers are using the machine and the service engineer has authenticated themselves |
| **Event flow** | 1. Authenticates 2. If authentication successful, you can take out the cash |
| **Post-condition** | No more cash in the machine and the machine is now unlocked |
| **Includes** | UC19 (Authenticate Engineer) |
| **Extensions** | - |
| **Triggers** | - |

|  |  |
| --- | --- |
| **ID** | **UC17** |
| **Name** | Run test |
| **Description** | The engineer can perform test on the system |
| **Pre-conditions** | No customers are using the machine and the service engineer has authenticated themselves |
| **Event flow** | 1. Authenticates unless machine has already been unlocked 2. If authentication successful, you can run the test |
| **Post-condition** | Machine has now been tested and the machine is now unlocked |
| **Includes** | UC19 (Authenticate Engineer) |
| **Extensions** | - |
| **Triggers** | - |

|  |  |
| --- | --- |
| **ID** | **UC18** |
| **Name** | Fill stock |
| **Description** | The stock of the machine will now be filled up. For example, the drinks machine would be filled up with coffee beans, cream, milk, sugar etc. |
| **Pre-conditions** | No customers are using the machine and the service engineer has authenticated themselves |
| **Event flow** | 1. Authenticates unless machine has already been unlocked 2. If authentication successful, you can run the test |
| **Post-condition** | The stock of the machine is full, and the user has authenticated |
| **Includes** | UC19 (Authenticate Engineer) |
| **Extensions** | - |
| **Triggers** | - |

|  |  |
| --- | --- |
| **ID** | **UC19** |
| **Name** | Authenticate Engineer |
| **Description** | The engineer will have to authenticate themselves to use the functions of the machine |
| **Pre-conditions** | No customer can use the machine |
| **Event flow** | 1. Authenticates unless machine has already been unlocked 2. If authentication successful, they won’t have to login again for the rest of the session. At the end of session, they will be unauthenticated, and the machine will lock up |
| **Post-condition** | Machine has now been tested and the machine is now unlocked |
| **Includes** | UC16 (Collect Cash)  UC17 (Run Tests)  UC18 (Fill Stock) |
| **Extensions** | UC11 (Receive Alarm) |
| **Triggers** | Failure of authentication will send an alarm to the service operator |

## Task 2

### Non-functional requirements

#### Security

The security is an important feature of the machine and access to any of the administrative functions is very important. The administrative features that people can access is very important.

#### Performance

The se

#### Capacity

The se

#### Interface

The se

#### Power

The se

#### Efficiency

The se