Blockchain Final Project

G Shalom
BT19GCS004

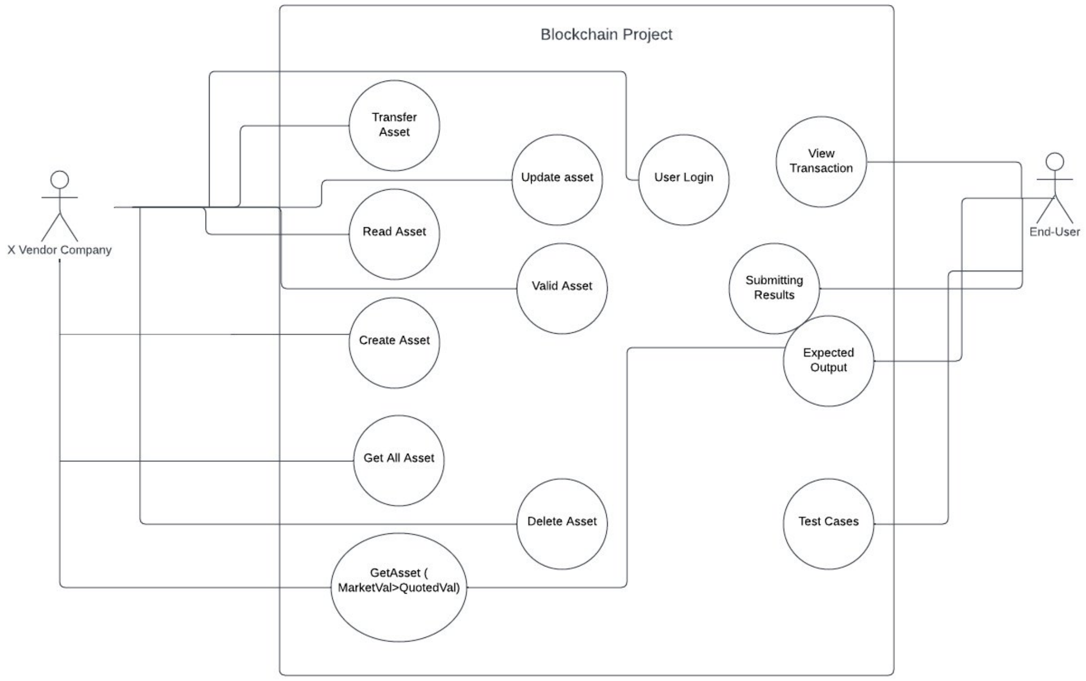
A brief project plan on how you approached the project execution (tasks, timelines etc)

I first started with installing the Hyperledger Fabric. As I was using a Windows machine, I installed the Hyperledger Fabric in WSL. I then proceeded to install the necessary pre-requisites, which were git, cURL, JQ and build-essential. After this step completed, I downloaded the fabric samples, docker images and binaries. Now moving on the first application, I brought the test-network down to ensure the environment is clean. Then I launched the Fabric test network using the network.sh script. After this, I deployed the chaincode by calling the ./network.sh script with the chaincode name and language set as javascript. Then, I moved to the application-javascript folder and ran ‘npm install’ command to install the key application dependencies defined in package.json. In the end, running the ‘node app.js’ command runs the application and displays the output.
2 attributes, named MarketValue and QuotedValue, were added and a new function GetAssetsWhereMarketValueGreaterOrEqualToQuotedValue.

The timeline is as follows
4th November to 6th November :- Installation and solving issues relating to it
7th November to 8th November :- Writing new attributes and function and testing code & solving issues which came up
8th November to 9th November :- Creation of report along with recording

A UML use-case diagram of the asset-transfer-basic smart contract

The UML use-case diagram is given below



A walk-through of the code you wrote for accomplishing steps 1a & 1b above

In the assetTransfer.js, located inside the chaincode-javascript folder, I added 2 new attributes in all the maps present in the assets array. After that, I added 2 new parameters in CreateAsset function and UpdateAsset function for the 2 new attributes. I also made sure that the 2 attributes are integers only. I also created a function called 'GetAssetsWhereMarketValueGreaterOrEqualToQuotedValue' where an if condition is set to allow only those assets whose market value is greater than or equal to quoted value.

Moving on to the app.js file located inside application-javascript folder, I called my newly created function there by passing the function name in evaluateTransaction and printed the results.