**Student Name: Mohammed Faiyaz Hussain**

**Student Number: ST2015599**

**Module code: CIS5027\_T1\_21**

**Module: Object Oriented System Design**

**Module Leader: Barry Bentley**

**UML Diagrams**

**A picture containing text, indoor

Description automatically generatedUse Class**

The diagram above shows how all the classes in the blackjack game interact with each other. The ‘Client/User’ and ‘Server’ are associated with each other, and they both connect to the ‘Blackjack’ game. The main class, which is called ‘Blackjack’ is inherited and aggregated with objects from other classes which is named the ‘Card’ class is inherited from two subcategories or Enums, which contain the ‘Suit’ and ‘Value’ which represents the structure and model of the card. The ‘Deck’ class is responsible for ‘Shuffling’, ‘Adding’ and ‘Removing’ a card, which then aggregates towards the ‘Main’ class thus, creating a deck for the ‘Player’. Finally, the ‘Player’ class is associated with the ‘Blackjack’ class by sharing the objects of information based on the player.

**Use Case**

Diagram, schematic

Description automatically generatedThe diagram shows how the ‘Client/User’ initiates the ‘Server’ in order to ‘Launch’ the ‘Blackjack Game’ they have to enter the ‘Number of Player(s)’ and their ‘Names’. The players go up against the ‘Dealer’. The players have an option to ‘Hold’ or ‘Deal’, which then determines the result. If the Dealer ‘Busts’, Player ‘Wins’. If Player ‘Busts’, Dealer ‘Wins’. But if it is a ‘Draw’ then they have an option to ‘Push’ which would go either way. Overall, each result leads to the game ending.

**Creating Blackjack**

Graphical user interface, text, application

Description automatically generated with medium confidence**Blackjack/main Class**

**Graphical user interface, text, application, email

Description automatically generatedDeck Class**

Text

Description automatically generated**Card Class**

**Graphical user interface, text, application

Description automatically generatedPlayer Class**

**Server/Client**

These classes were not able to be implemented due to issues with running the code.

**How to Run Blackjack**

**Running Blackjack**

**Required:** JAVA SE -1.8

A screenshot of a computer

Description automatically generated with medium confidence

Click the ‘Run’ Button to start the program.

**Text

Description automatically generatedAdding Players**

Introduces user to the game and asks them to enter the number of players and their names.

**Playing Blackjack**

**Text, letter

Description automatically generated**

Player goes against dealer, with having the option to deal or hold to resume the game and determine the result.

**GIT Repository**

<https://github.com/KingFu22/CIS5027-Object-and-Oriented>