Group3 Deliverable 2

Group Members

Elisha Nesci
Furkan Gonul
Omama Eiz Eddin
Wei-Hsiang Chang

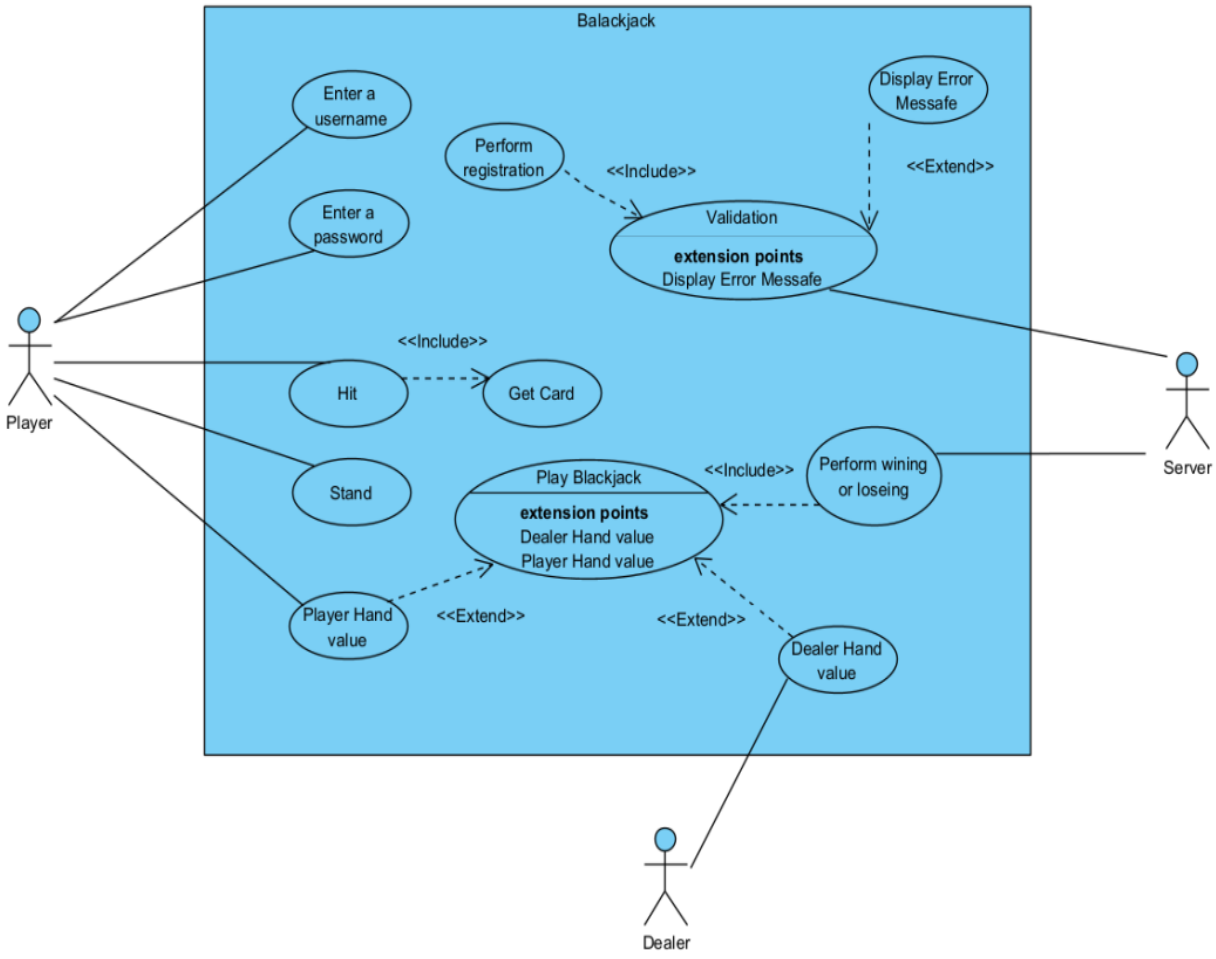
SYST17796 Fundamentals of Software Design and Development

Professor Paul Bonenfant

March.2021

Table of Contents

Use Case Diagram



Use Case Narratives

1. A Player wants to play the Blackjack card game
   1.1 Player creates an account by entering a username and a password
   1.2 Player login
   1.3 Connection success
      1.3.1 Player makes a bet
      1.3.2 Dealer deals 2 cards to the players and two to himself (1 card face up, the other face down)
      1.3.3 Player cards total is less than 21
         1.3.3.1 Player chooses to stand
            1.3.3.1.1 Dealer flips up his face down card
               1.3.3.1.1.1 Player hand is higher than dealer's
                  1.3.3.1.1.1.1 Player wins the bet
               1.3.3.1.1.2 Player hand is less than the dealers
                  1.3.3.1.1.2.1 Player loses his bet
         1.3.3.2 Player chooses to hit
            1.3.3.2.1 Dealer deals another card from the deck
            1.3.3.2.2 Dealers flips up his face down card
               1.3.3.2.2.1 Player hand is higher than dealer's
                  1.3.3.2.2.1.1 Player wins the bet
               1.3.3.2.2.2 Player hand is less than the dealers
                  1.3.3.2.2.2.1 Player loses his bet

      1.3.4 Player cards total is equal to 21
         1.3.4.1 Player wins the bet
      1.3.5 Player cards total above 21 (Bust)
         1.3.5.1 Player loses his bet
   1.4 Connection fail
      1.4.1 The username or the password you entered is not correct please try again
2. Player exits the program

Happy path: (wining path)

1 -> 1.1 -> 1.2 -> 1.3 -> 1.3.1 -> 1.3.2 -> 1.3.3 -> 1.3.3.1 -> 1.3.3.1.1 -> 1.3.3.1.1.1 -> 1.3.3.1.1.1.1 ->2

1 -> 1.1 -> 1.2 -> 1.3 -> 1.3.1 -> 1.3.2 -> 1.3.3 -> 1.3.3.2 -> 1.3.3.2.1 -> 1.3.3.2.2 -> 1.3.3.2.2.1 -> 1.3.3.2.2.1.1 ->2

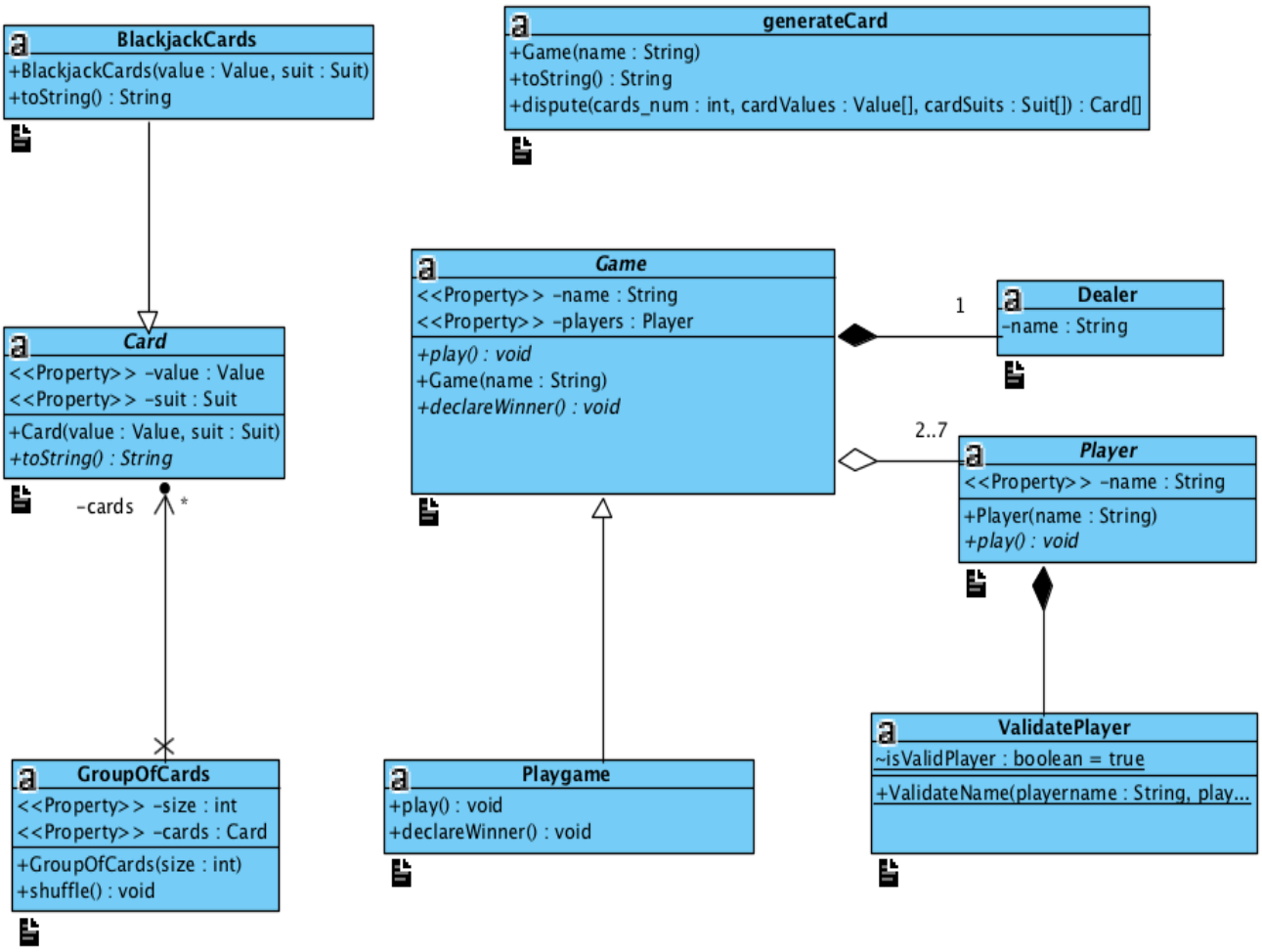
1 -> 1.1 -> 1.2 -> 1.3 -> 1.3.4 -> 1.3.4.1 ->2

Alternative path: (Connection fail)
1 -> 1.1 -> 1.2 -> 1.4

Cancel path: (Cancel after the connection)

1 -> 2
1 -> 1.1 -> 1.2 -> 1.3 -> 2

Class Diagram



1.Project Background and Description

Our project goal is to create a blackjack game. More details are as following:

1. The playing room can only be accommodated for 2 to 7 people at a time, over 7 people need to open a new room.
2. Every game needs to be a dealer in it, and all players compete with the dealer. If you are dealt 21 or get more values than the dealer, you win, otherwise you lose.
3. There is only one round in the game. in the round you can choose hit (add one more card) or stand. Finally, check the value with dealer, or maybe over 21then you lose. The one winning the dealer will be the winner. Otherwise, you lose the game. After playing this round, the game will terminate.

2. Design Considerations

There are some different associations between each classes from the class diagram, using various design principle to implement it, as following:

Inheritance

Blackjack is one kind of the card game, it is the subclass of the card. Also, accessing the same data type declared by superclass. Use **enum** to implement suit and values to be more **loose coupling** design.

Playgame extends Game class, override the method and **delegate** the jobs from Game, focus on playing the game and declaring the winner.

Composition

Each Game class must have a dealer, who is the composition of the game. Implementing Game as abstract class, it can be more **flexible** for the other subclass to focus on the concrete thing and **encapsulated** the method.

Player needs to validate first before entering the game, increasing the validator class to do specific thing to be more **cohesiveness**.

Aggregation

Each game accommodates 2 to 7 players, every player has a unique name with private String data type, and Game class has scope that contains Arraylist of player data type, it is **organized** construction.