BLACKJACK GAME

GROUP 3: Andre Rampono- Dawson Mercer – Quynh Pham

DECEMBER 9, 2021

CUSTOMER: ERIC STOKE

Table of Contents

1. Executive Summary

The goal of the project was to create a Blackjack game for a local client, Casino NL. This game is to have a web interface and its purpose will be to help advertise for the casino by increasing the number of people who are familiar with the Casino NL game by creating an interactive and engaging experience for online users. Our team's goal was to build the logic for the Blackjack game that is to be hosted on the client's website, CasinoNL.ca. This game will only be accessible to users who are 18 and above as per the Newfoundland & Labrador government's regulations surrounding gambling. The game was created using the programming language python. The Blackjack game we have created follows the basic rules of Blackjack as to not confuse any new users with too many rules, this game is to be used as a first impression into cards games in hopes to spark interest in future card games (online or in the casino). In the document, we have displayed and outlined the functionality of how the gameplay loop works between the dealer and player (Section 4). We have also displayed the decisions and activities a player encounters when playing the game (Section 5). Next, we have summarized the various classes that the software will have which includes Card, Hand, Deck, Player, and Dealer and how they interact with each other in the Class Diagram. Lastly, the Behavioural Models show the sequence in which the objects in the software interact with each other and how the player receives a card, and their score is updated (Section 6).

2. System Request

Project: Eric Stock Casino blackjack

Client: Eric Stock, President & CEO, NL Casino. ('Client').

Project Sponsor: The project sponsor is Eric Stock

Business Need

The Client wishes to have a prototype blackjack game created.

Business Value

The client is considering adding a web interface around the game and releasing it to help advertise their casino. (Note, you will only focus on the python blackjack component and not the web interface)

Special Issues or Constraints

The following issues need to be considered: system needs to be completed by early of December 2021

3. Requirements Definition

a. Functional Requirements

Gameplay:

- Two cards will be distributed to the players
- Players will place their bet with limitation
- Players can choose to get more cards (hit- enter H) or get no more cards (stand-enter S)
- The point will be calculated for each player and dealer
- Players can win or loose their bets depends on their scores and dealer's score.
- Players can have option to stop the game after each round or continue the game with new players if only one player stays in the game

Monetization:

- Players will enter their balance in the beginning of the game

b. Non-Functional requirements

Cultural & Political:

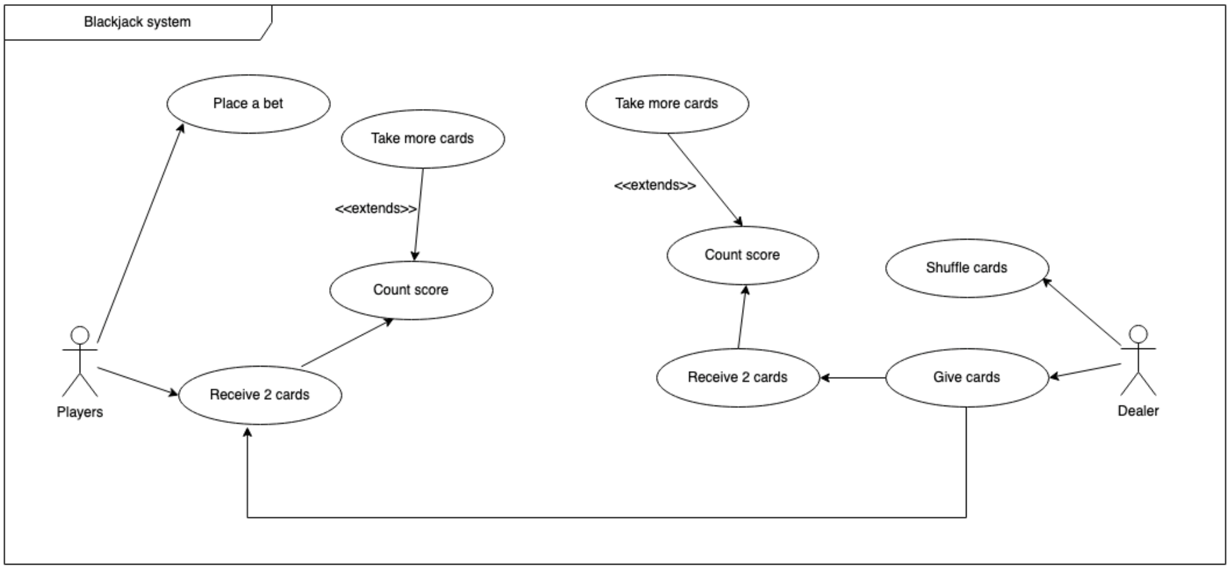
- The primary focus of the game is to have fun
- The game is allowed only for adult (18+)

Operational:

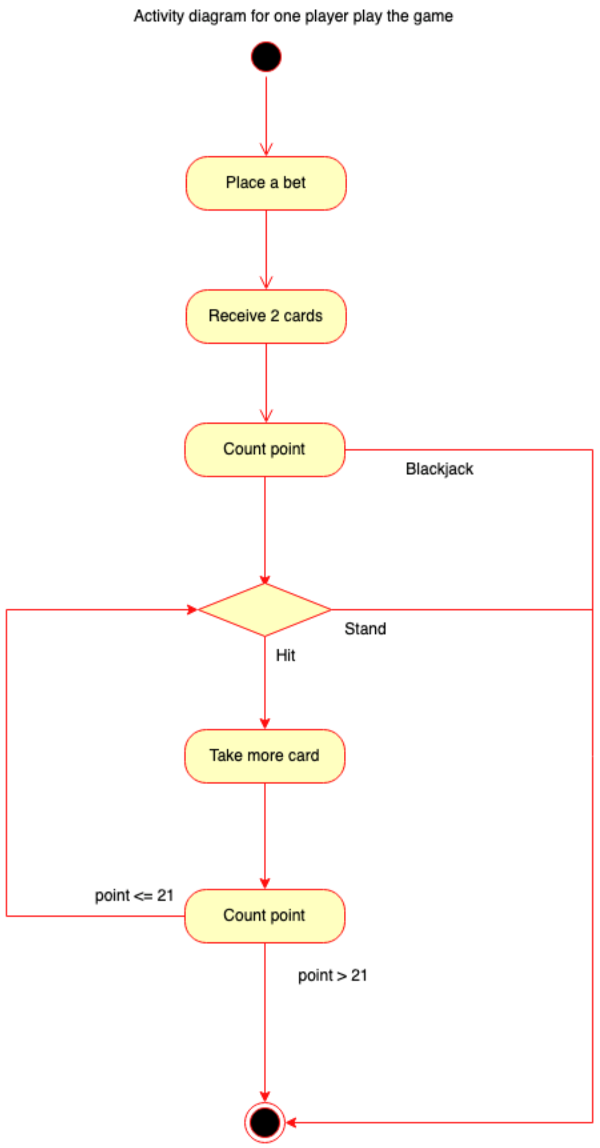
- The game will need an computer in order to access the game.

4. Functional Models

a. Use Case Diagram for the Entire System



b. Activity Diagram for a Major Use Case



5. Structural Models

a. CRC Cards for the Classes of the System

Front:
Class Name: Card ID: 2 Type: Concrete, Domain
Description: Represents cards in the blackjack game

Associated Use Cases:

Responsibilities: Collaborators: Deck

Back:
Attributes:
-Score

Relationships
Generalization (is a kind of): none
Aggregation (has parts): none
Other Associations: Deck

Front:
Class Name: Hand ID: 5 Type: Concrete, Domain
Description: Represents cards of player and dealer

Associated Use Cases: showCard, standCard, hitCard, isBlackJack, countScore, playGame, printResult

Responsibilities: Collaborators: Dealer, Player

Back:
Attributes:
- Cards

Relationships
Generalization (is a kind of): none
Aggregation (has parts): none
Other Associations: Dealer, Player

Front:
Class Name: Deck ID: 4 Type: Concrete, Domain
Description: Represents a deck containing 52 cards.

Associated Use Cases: giveCard, hitCard

Responsibilities: Collaborators: Card

Back:
Attributes:

- Cards

Relationships
Generalization (is a kind of): none
Aggregation (has parts): none
Other Associations: Card

Front:
Class Name: Player ID: 1 Type: Concrete, Domain
Description: Represents a player who plays blackjack

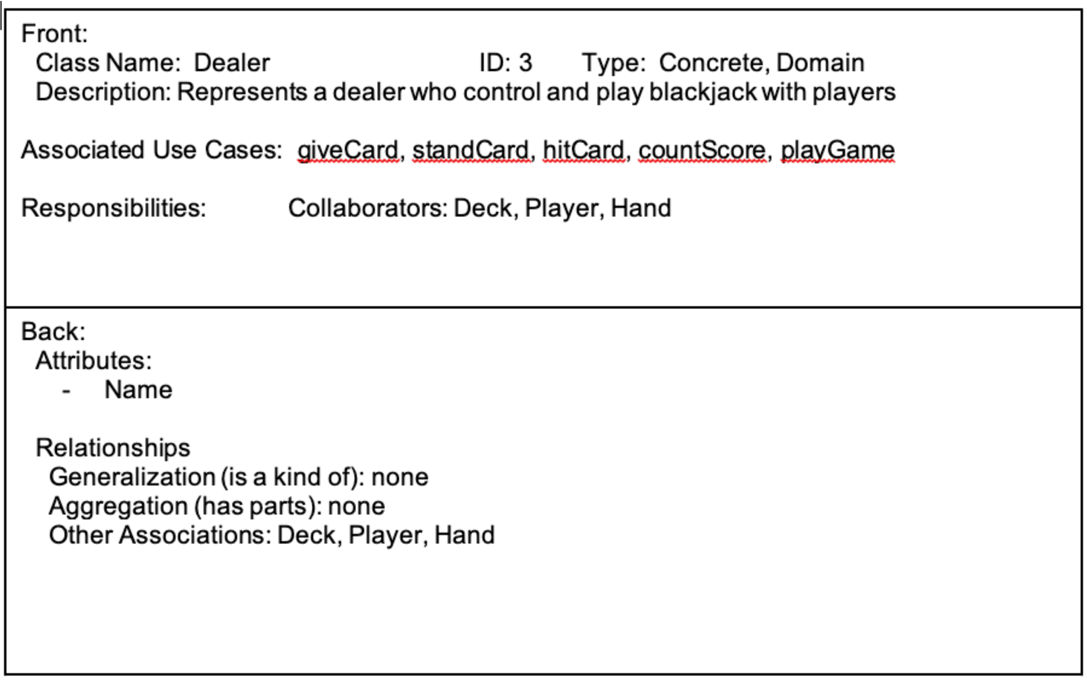
Associated Use Cases: placeBet, giveCard, standCard, hitCard, countScore, playGame, printResult

Responsibilities: Collaborators: Dealer, Hand

Back:
Attributes:

- Name
- TotalAmount
- BetAmount
- Age
- Hand

Relationships
Generalization (is a kind of): none
Aggregation (has parts): none
Other Associations: Dealer, Card



Front:

Class Name: Dealer ID: 3 Type: Concrete, Domain

Description: Represents a dealer who control and play blackjack with players

Associated Use Cases: giveCard, standCard, hitCard, countScore, playGame

Responsibilities: Collaborators: Deck, Player, Hand

Back:

Attributes:

- Name

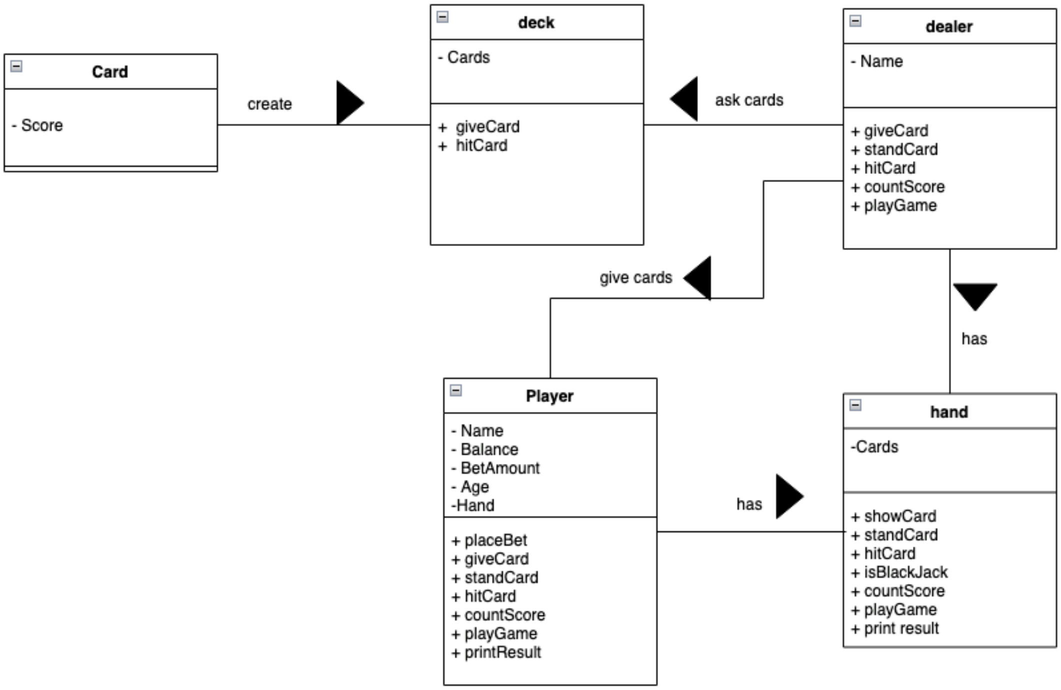
Relationships

Generalization (is a kind of): none

Aggregation (has parts): none

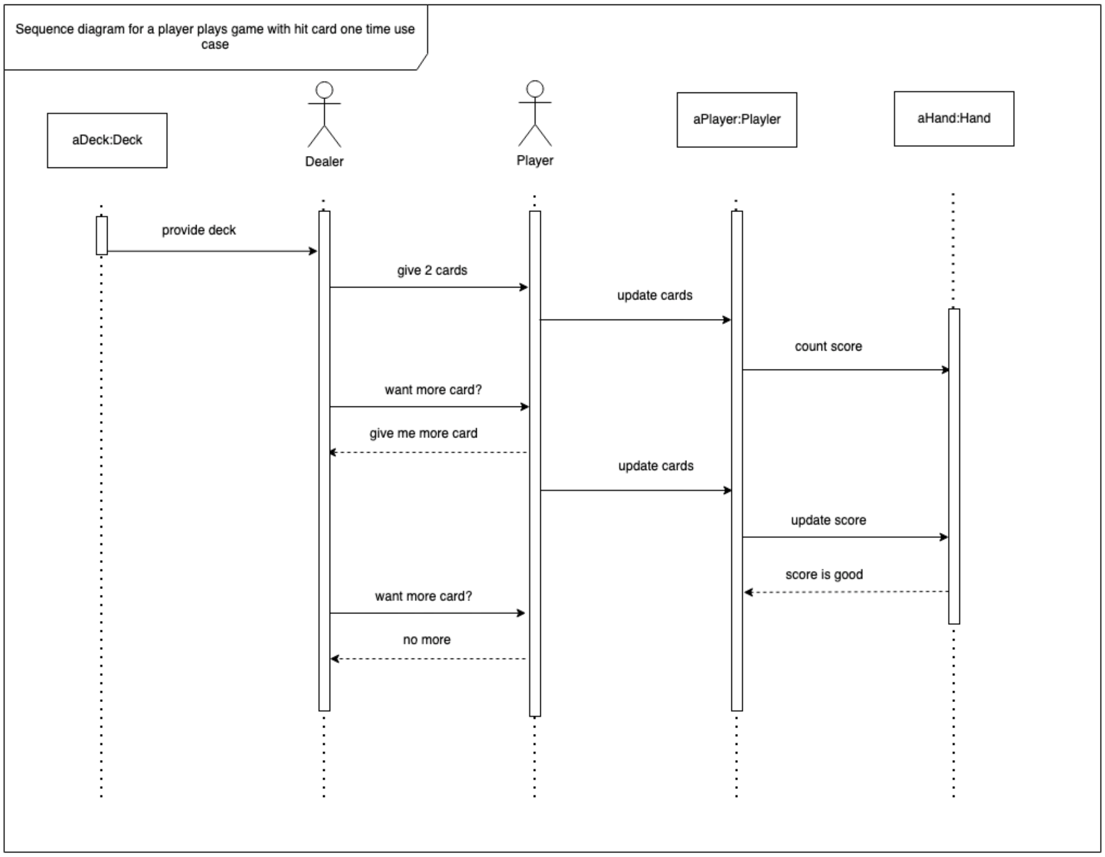
Other Associations: Deck, Player, Hand

b. Class Diagram for the System

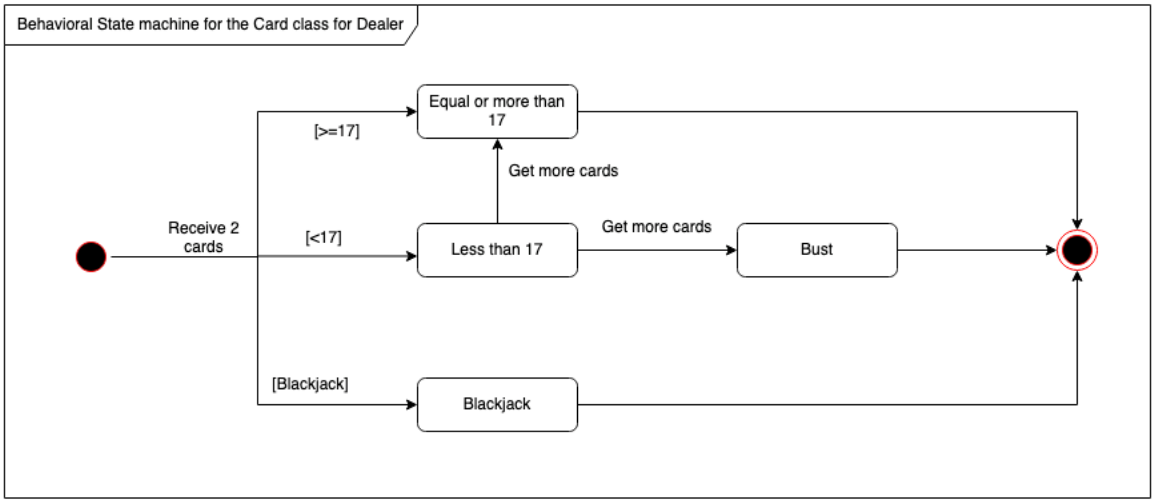


6. Behavioral Models

a. Sequence Diagram



b. Behavioral State Machine for One Class Which Undergoes State Changes


Behavioral State machine for the Card class for Dealer
Receive 2 cards
[>=17]
[<17]
[Blackjack]
Equal or more than 17
Get more cards
Less than 17
Get more cards
Bust
Blackjack