Designing an Online Blackjack Game

→ The objective of the game is to get closer to 21 than the dealer without exceeding the 21 points.

Value of Ace → 1 or 11

Jack, Queen & King → 10

2-10 → 2-10 Same value as Numbers written.

Expectations from interview

- How many players can play Blackjack?
- Can players play against each other?

Point Dynamics

- Upto how many point can the player or the dealer hit the card?
- What will happen if the dealer & the player both get the same points?

Card Limit

- Is there a limit on the number of cards the player take?

Requirements for the Online Blackjack Game

R1: The Blackjack game contains the shoe of cards which contains one or more deck of cards in it.

R2: The deck will consists of 52 cards in four suits, where each suit contains 13 cards. The ace 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen & King.

R3: Value calculated with each card.

Card	Face value
Ace	1 or 11
From 2 to 10	Equals the card number
Face cards (King, Queen, and Jack)	10

R4: There can be two types of users that can play the Blackjack game. i.e. the dealer & the player.

R5: The player places a bet at the start of the game

R6: The dealer will deal two cards to themselves & two to the player at the start of game

R7: The player will have both cards exposed, while the dealer has one card facing up & the other card down.

R8: The player can hit (Draw an addional card.) if their hand has less than 21 points.

R9: The dealer can hit if their hand is less than 17.

R10: If the player or the dealer hand is more than 21, they bust & lose the game.

R11: The player can decide to not get a further card by standing pat

R12: At the end, if the points of the player are more than that of the dealer but less than 21. they they win the game & get 100 percent profile.

R13: If a player gets 21 points by winning an ace & a face card of 10, then they are called a blackjack & get 150 percent profile

R14: If the player & the dealer have the same points at the end of the game, the player can take their bet money back or can replay the game.

R15: If the player left a game in the middle of the game, the dealer will win the game.

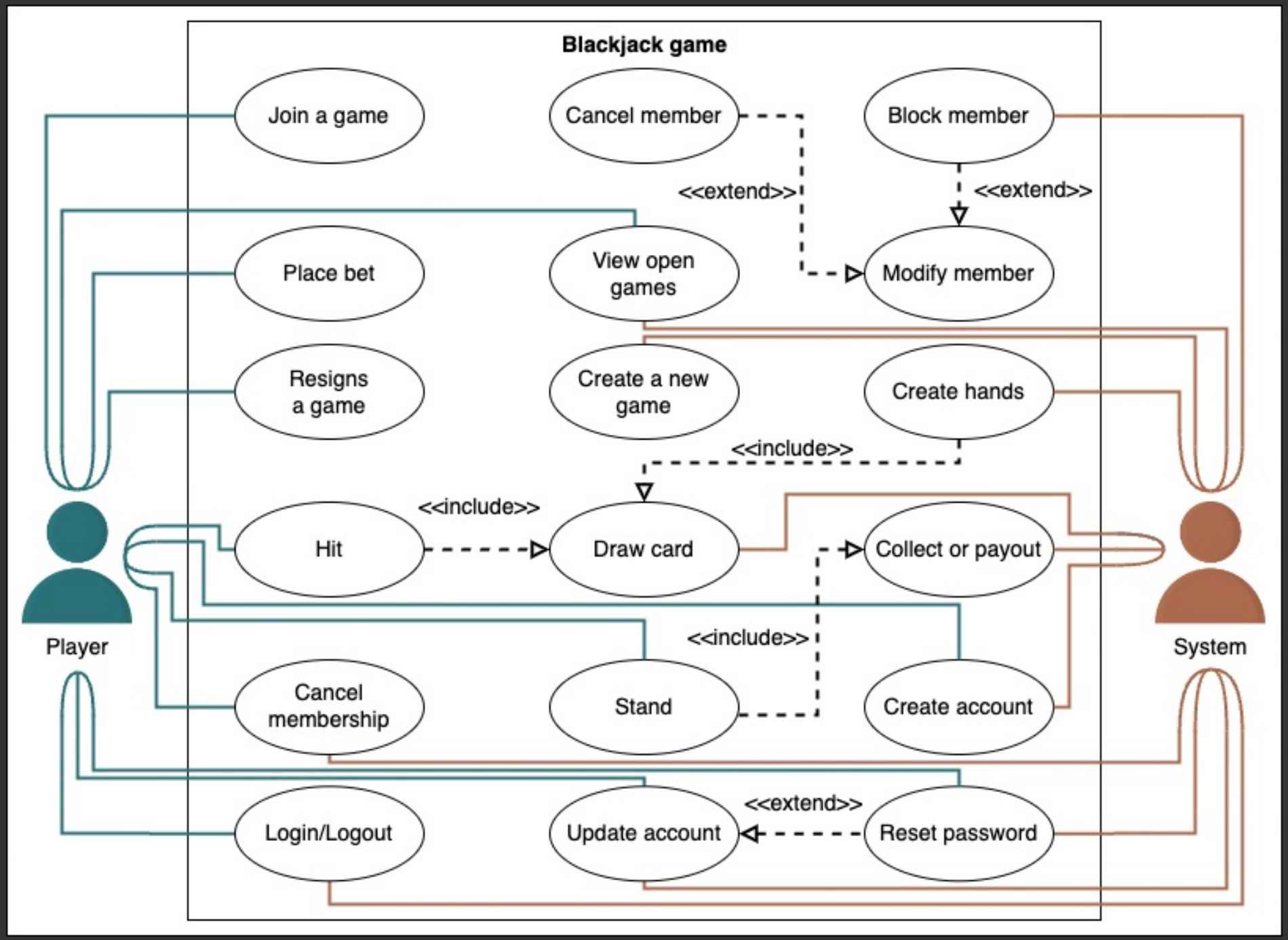
Actors

→ Primary Actors → Player

→ Secondary Actors → Dealer

Player	Dealer
Join a game	Create a new game
Place bet	View open games
View open games	Create hands
Resigns a game	Draw card
Hit	Collect or payout
Stand	Block member
Create account	Create account
Update account	Update account
Reset password	Reset password
Cancel membership	Cancel membership
Login/Logout	Login/Logout

Use Case Diagram → blackjack Game



Class Diagram for Online Blackjack Game

1. Cards

Card
- suit : Suit - faceValue : int
+ Card(Suit cardsuit, int cardFaceValue)

2. Deck

Deck
- cards : Card {list}
+ Deck( ) + getCard( ) : Card {list}

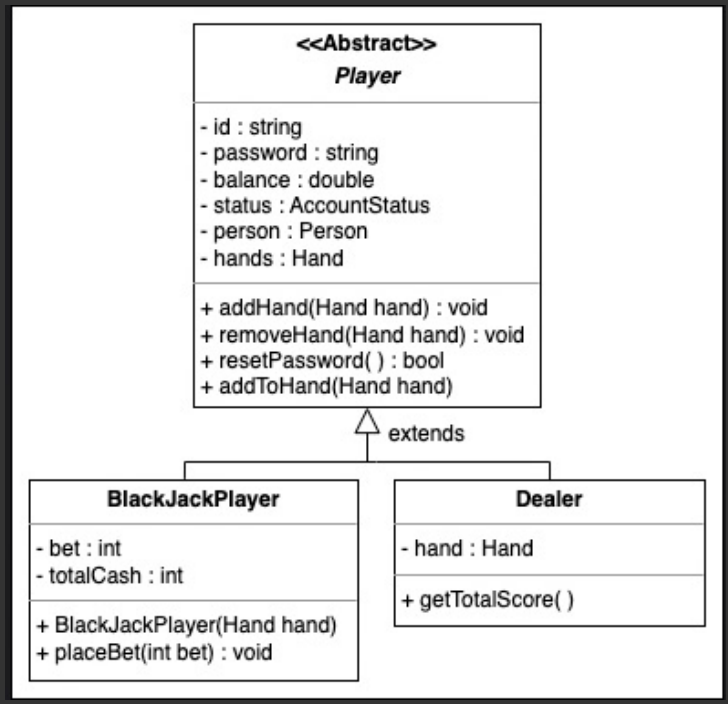
3. Shoe : Device to hold multiple Deck.

Shoe
- decks : Deck {list} - numberOfDecks : int
+ Shoe(int numberOfDecks, Deck decks) + createShoe( ) : void + shuffle( ) : void + dealCard( ) : Card

4. Hand

Hand
- cards : Card {list}
+ Hand(Card card1, Card card2) + getScore( ) : int + addCard(Card card)

5. Players



6. BlackJack Controller

BlackJackController

+ validateAction( ) : bool

7. Blackjack gameview

BlackJackGameView

+ playAction(string action, Hand hand) : void

8. Blackjack Game

BlackJackGame

- player : BlackJackPlayer
- dealer : Dealer
- shoe : Shoe
- maxNumberOfDecks : int

+ BlackJackGame(BlackJackPlayer player, BlackJackPlayer dealer)
+ playAction(string action, Hand hand) : void
+ hit(Hand hand) : void
+ stand(Hand hand) : void
+ start( ) : void

9. Enumerations

<<enumeration>>
Suit

Heart
Spade
Club
Diamond

<<enumeration>>
AccountStatus

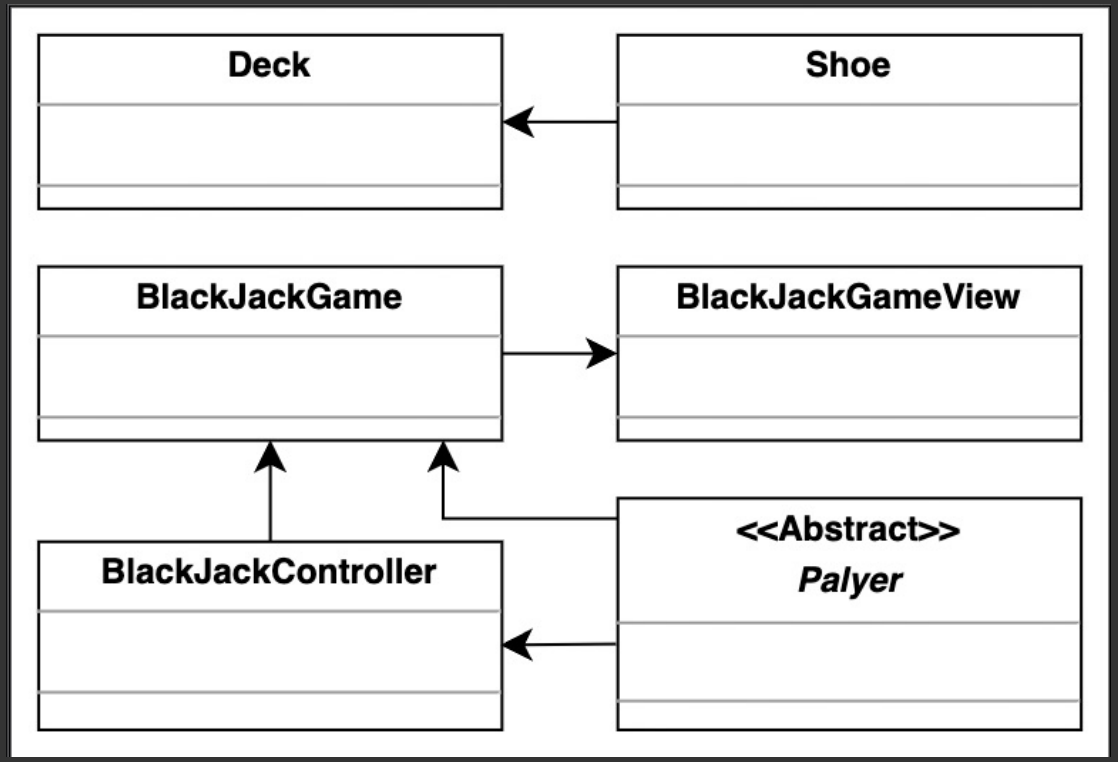
Active
Closed
Canceled
Blacklisted
None

10. Person

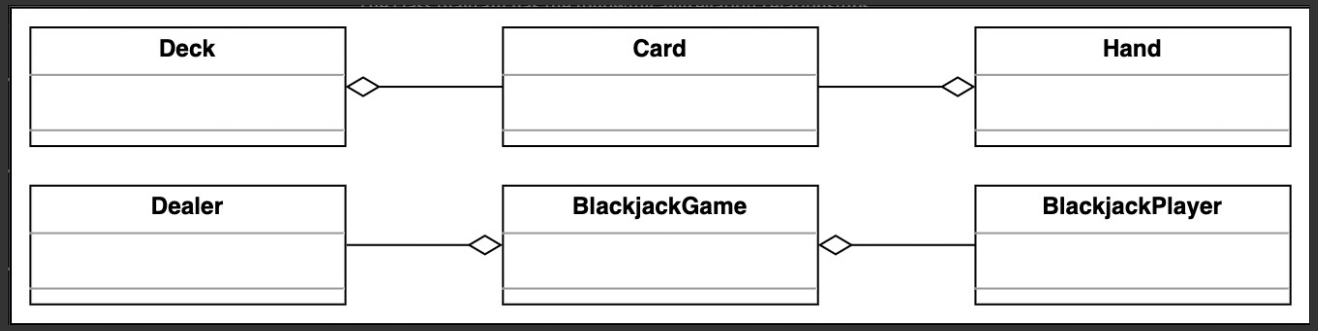
Person

- name : string
- streetAddress : string
- city : string
- state : string
- zipcode : int
- country : string

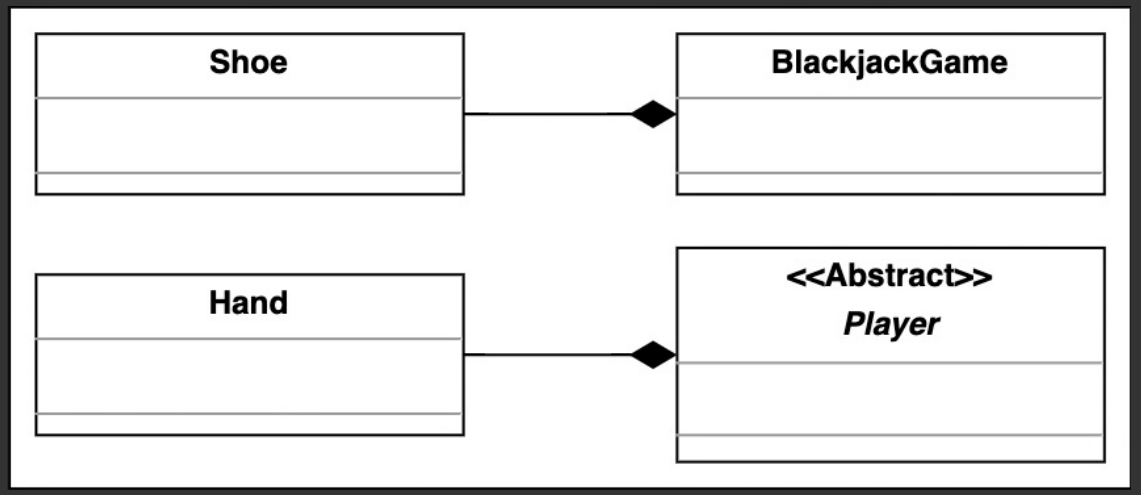
Associations



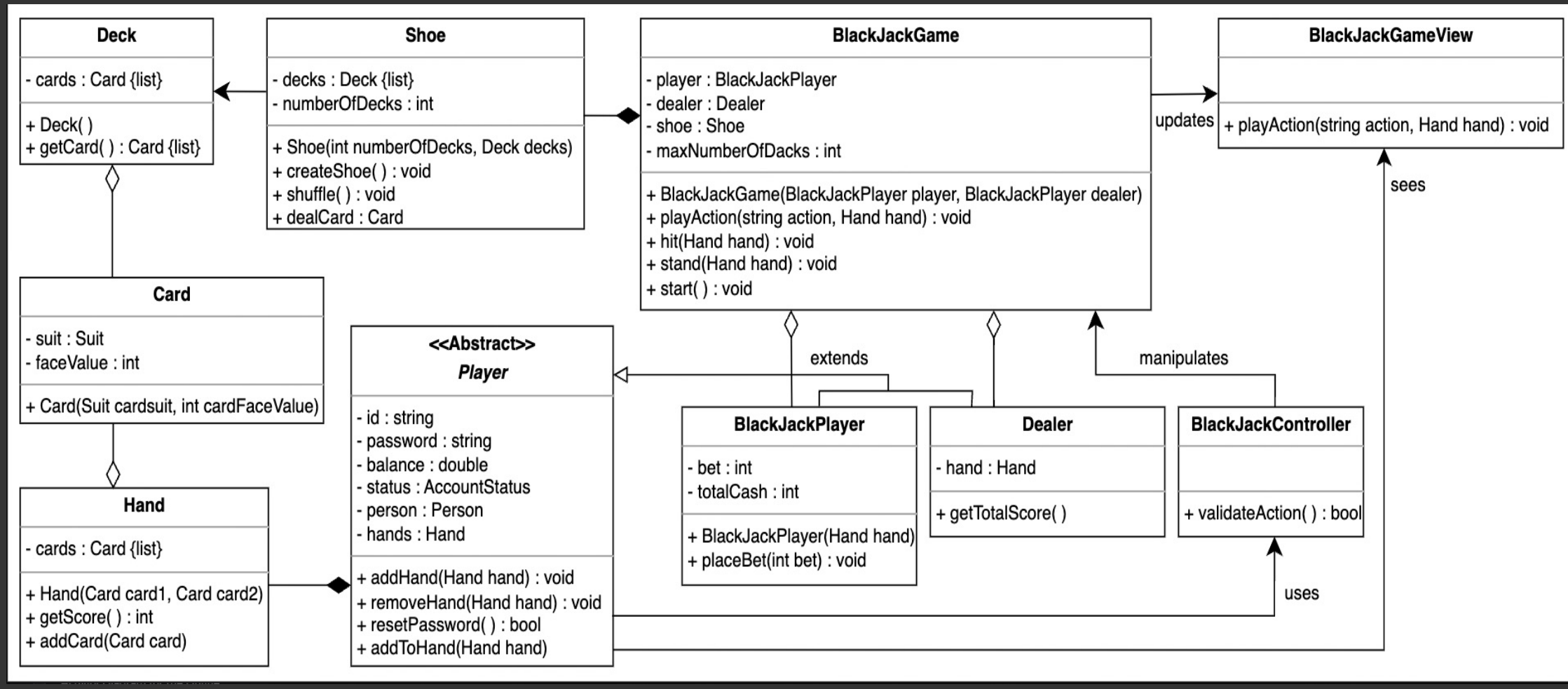
Aggregations



Composition

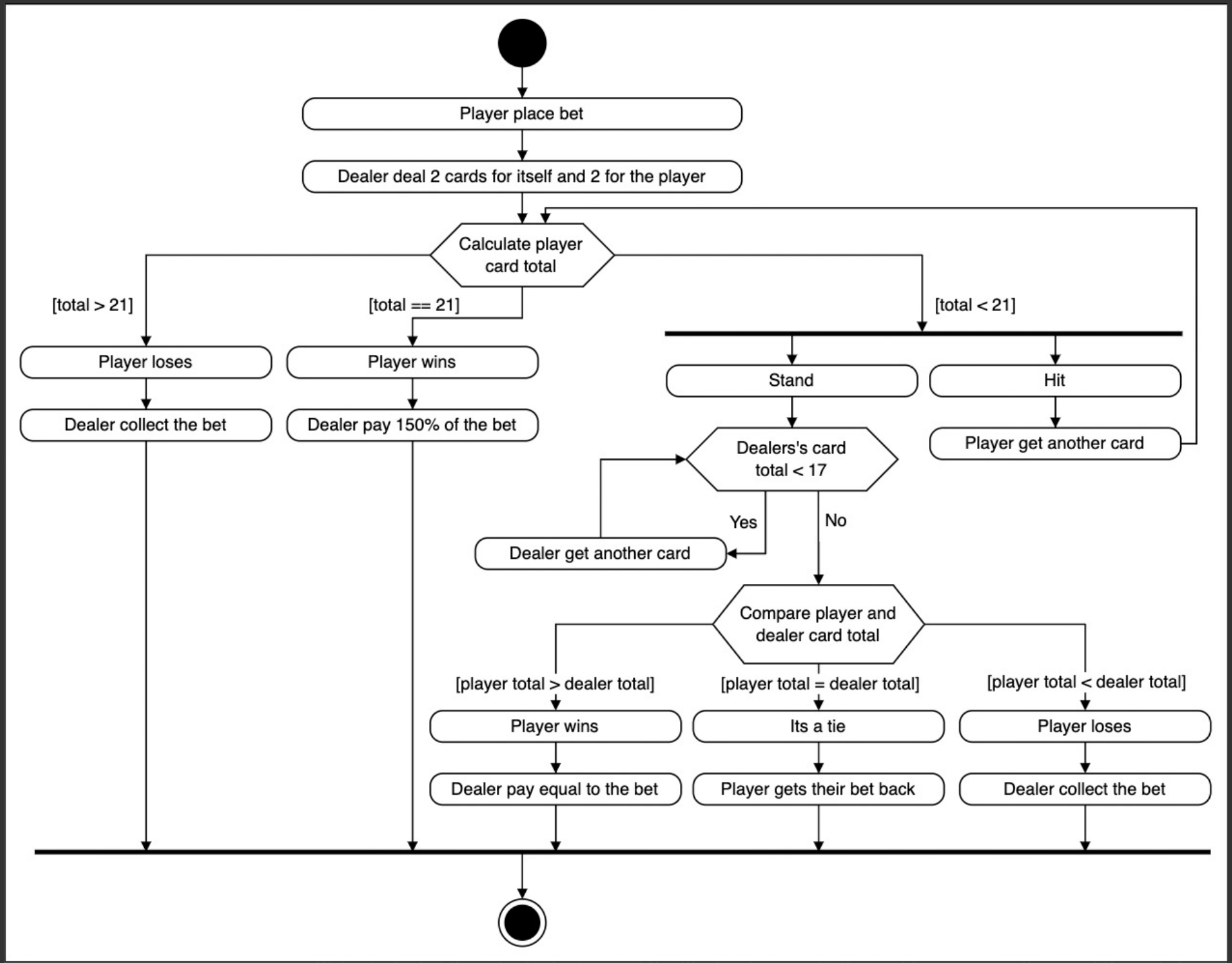


Class Diagram → Blackjack Diagram



→ Design Pattern → Strategy pattern can be applied.

Activity Diagram of BlackJack Game



Code

1. Enumeration & Data type

```
// Enumeration
enum Suit {
  HEART,
  SPADE,
  CLUB,
  DIAMOND
}

enum AccountStatus {
  ACTIVE,
  CLOSED,
  CANCELED,
  BLACKLISTED,
  NONE
}

// Custom Person data type class
public class Person {
  private String name;
  private String streetAddress;
  private String city;
  private String state;
  private int zipCode;
  private String country;
}
```

2. Card

```
public class Card {
  private Suit suit;
  private int faceValue;

  public Card(Suit suit, int faceValue);
}
```

3. Deck & Shoe

```
public class Deck {
  private List<Card> cards;

  public Deck();
  public List<Card> getCards();
}

public class Shoe {
  private List<Deck> decks;
  private int numberOfDecks;

  public Shoe(int numberOfDecks,
List<Deck> decks) {
    // 1. createShoe();
    // 2. shuffle();
  }
  public void createShoe();
  public void shuffle();
  public Card dealCard();
}
```

4. Hand

```
public class Hand {
  private List<Card> cards;

  public Hand(Card card1, Card card2);
  public int getScores();
  public void addCard(Card card);
}
```

5. Players

```
public abstract class Player {
  private String id;
  private String password;
  private double balance;
  private AccountStatus status;
  private Person person;
  private Hand hand;

  public void addHand(Hand hand);
  public void removeHand(Hand hand);
  public abstract boolean resetPassword();
  public void addToHand(Hand hand);
}

public class BlackjackPlayer extends Player {
  private int bet;
  private int totalCash;

  public BlackjackPlayer(Hand hand);
  public void placeBet(int bet);
  public boolean resetPassword(){
    // definition
  }
}

public class Dealer extends Player {
  private Hand hand;

  public int getTotalScore();
  public boolean resetPassword(){
    // definition
  }
}
```

6. Blackjack Controller & Game View

```
public class BlackjackController {
  public boolean validateAction();
}

public class BlackjackGameView {
  public void playAction(String action, Hand hand);
}
```

7. Blackjack Game

```
public class BlackjackGame {
  private Player player;
  private Dealer dealer;
  private Shoe shoe;
  private final int MAX_NUM_OF_DECKS = 4;

  public BlackjackGame(BlackjackPlayer player,
Dealer dealer);
  public void playAction(String action, Hand
hand);
  public void hit(Hand hand);
  public void stand();
  public void start();
}
```