**Use Case Specification Document**

**1. Use Case Overview**

**System**: Multiplayer Blackjack Game  
**Actors**: Player, Dealer, Game Server  
**Primary Goal**: Facilitate a multiplayer blackjack experience with betting, turn-based gameplay, and anti-cheating mechanisms.

**2. Use Case List**

1. **UC1: Player Login**
2. **UC2: Manage Account Funds**
3. **UC3: Join a Game**
4. **UC4: Place a Bet**
5. **UC5: Play a Turn (Hit/Stand/Double Down/Split)**
6. **UC6: Dealer Turn Execution**
7. **UC7: Determine Game Outcome**
8. **UC8: Update Leaderboard**
9. **UC9: Detect and Prevent Cheating**

**3. Detailed Use Cases**

**UC1: Player Login**

**Actors**: Player  
**Preconditions**: The player must have an existing account.  
**Basic Flow**:

1. The player enters login credentials.
2. The system validates credentials.
3. The player is authenticated and enters the main lobby.

**Alternative Flows**:

* **Invalid Credentials**: System prompts re-entry of login information.
* **Account Locked**: System denies access after multiple failed attempts.

**UC2: Manage Account Funds**

**Actors**: Player  
**Preconditions**: Player is logged in.  
**Basic Flow**:

1. The player selects deposit or withdrawal.
2. The player enters an amount.
3. The system processes the transaction.
4. The player’s balance updates.

**Alternative Flows**:

* **Insufficient Funds**: System notifies the player.
* **Transaction Error**: System notifies the player of failure.

**UC3: Join a Game**

**Actors**: Player, Game Server  
**Preconditions**: Player is logged in with sufficient funds.  
**Basic Flow**:

1. The player selects a game lobby.
2. The system assigns the player to a game.
3. The game starts when enough players join.

**Alternative Flows**:

* **Lobby Full**: Player is directed to another table.
* **No Available Dealers**: Player is notified and waits.

**UC4: Place a Bet**

**Actors**: Player, Game Server  
**Preconditions**: The player is in an active game.  
**Basic Flow**:

1. The player selects an amount to bet.
2. The system deducts the bet amount from their balance.
3. The system confirms the bet and deals initial cards.

**Alternative Flows**:

* **Insufficient Balance**: Player is notified and must adjust the bet.

**UC5: Play a Turn (Hit/Stand/Double Down/Split)**

**Actors**: Player  
**Preconditions**: The player has placed a bet.  
**Basic Flow**:

1. The player selects an action (Hit, Stand, Double Down, Split).
2. The system updates the player’s hand accordingly.
3. The turn moves to the next player.

**Alternative Flows**:

* **Bust**: Player loses and turn ends.
* **Split Not Possible**: System denies the action.

**UC6: Dealer Turn Execution**

**Actors**: Dealer  
**Preconditions**: All players have completed their turns.  
**Basic Flow**:

1. The dealer reveals their hand.
2. The dealer follows predefined Blackjack rules.
3. The dealer plays their turn automatically.

**UC7: Determine Game Outcome**

**Actors**: Game Server  
**Preconditions**: Dealer’s turn is complete.  
**Basic Flow**:

1. The system compares all hands.
2. The system determines winners and losers.
3. Payouts are distributed accordingly.

**UC8: Update Leaderboard**

**Actors**: Game Server  
**Preconditions**: A game round is complete.  
**Basic Flow**:

1. The system updates player rankings based on earnings.
2. The leaderboard displays the updated stats.

**UC9: Detect and Prevent Cheating**

**Actors**: Game Server  
**Preconditions**: Game is in progress.  
**Basic Flow**:

1. The system monitors player actions for patterns.
2. If cheating is detected, the system intervenes (e.g., reshuffling deck, removing player).

**4. Summary**

This document provides an in-depth examination of the primary use cases in the Multiplayer Blackjack Game. The system ensures a fair and engaging gameplay experience by incorporating key features such as account management, multiplayer interactions, betting mechanics, dealer execution, and automated game outcome determinations.

Additionally, measures for anti-cheating detection and leaderboard updates enhance the integrity and competitiveness of the game. The game server enforces rules and fair play through systematic tracking and intervention when necessary.

By implementing these well-defined use cases, the Multiplayer Blackjack Game will deliver a smooth and enjoyable user experience, ensuring both fairness and competitive engagement. The next steps involve validating these use cases against system requirements and refining user interactions through iterative testing.