**Test Plan for Multiplayer Blackjack Game**

#### **Classes to be Tested**

* **Card**: Represents a single card in the deck with properties like suit and value.
* **Deck**: A collection of cards that can be shuffled and dealt.
* **Shoe**: A collection of multiple decks.
* **Account**: Represents a player's/dealer account, storing personal data and balance.
* **Balance**: Manages a player's monetary balance.
* **Dealer**: The dealer in the game, who follows specific game rules like hitting/standing.
* **Player**: A player in the game, who takes actions like hitting, standing, and managing their hand.
* **Hand**: Represents a player's or dealer's hand of cards.
* **Lobby**: List of tables.
* **Table**: Manages the game table, including players, bets, and game state.
* **AccountManager**: Handles account creation, login, and balance management for players.

**Operations:**

**Account:**

TestNewDealer

testNewPlayer

testAccountCreationFailureWithEmptyUsername

**AccountManager**

testLoginSuccess

testLoginFailUsername

testLoginFailPassword

**BalanceTest:**

Valid Deposit

Deposits Negative Amount

Valid Withdraw

WithdrawNegativeAmount

WithdrawInsuficcientBalance

**Player:**

TestPlayerConstructor

* Id is set, array of hands is initialized with main hand and split hand, current bet is set to 0
* Includes getId, getHands, getCurrentBet

TestPlaceBet

* Bet is placed, account balance is updated
* Includes getCurrentBalance, getAccount

TestResetCurrentBet

* Bet is placed, current bet is reset, bet is now back to 0.0
* Includes resetCurrentBet, placeBet, gecurrentBet

**LobbyTable:**

TestNewTable

* Tests table constructor
* Includes getID, getNumPlayers, getMinBet

TestJoinTable

* Tests join method

TesIsOpen

**Lobby:**

TestLobbyConstructor

* Test lobby constructor
* Includes getNumTables, getAvailableDealers

TestTableList

* Test initial table list
* Includes getTableList

TestNewTableAdded

* Test updates to new table being added
* Includes getNumTables

TestNewDealerAdded

* Test updates to new dealer being added
* Includes newDealer, getAvailableDealers

**Dealer:**

TestDealerConstructor

* Test constructor
* Includes getId

TestDealNextCard

* Tests getNextCard method

**Card:**

TestCardConstructor

* Includes getSuit, getRank, getValues

**Hand:**

TestHandConstructor

* Includes getHandValue, getBet

TestHandHit

* Tests that your hand is updated when hit
* Card in hand == next card dealt in shoe

TestSetSplit

* Hand is split into two hands containing one card from the original hand
* Includes setSplit, getIsSplit, getHand, getTotalCards, getBet

TestDoubleDown

* Bet is doubled and final card is hit
* Includes getBet, getHand

#### **Instructions to Run the Test Suite**

1. **Set Up Environment**: Ensure that JUnit is included in your project
2. Shared Project (JUnit Test\_) ⇒ test ⇒ modelTest ⇒ Run SuiteClass.java
3. Server Project(JunitTest) ⇒ test ⇒ Run AccountManagerTest.java