## ****Implementation Manual****

### ****Overview****

This JavaFX application is a graphical representation of the Baccarat card game. The implementation involves creating the UI with buttons and images, handling game logic, and managing card values.

### ****Project Structure****

The project consists of the following classes:

1. **BaccaratApp**: Main JavaFX application class that controls the UI and the game flow.
2. **Deck**: Class that represents the deck of cards and handles card drawing.
3. **Card**: Class that represents individual cards with rank and suit.
4. **Player**: Class that represents a player (Player or Banker), their hand, and methods to calculate hand value.
5. **Images**: Card images stored in a folder named images.

### ****Dependencies****

* **Java 8 or above**: Since JavaFX is bundled with JDK 8, ensure that you are using an appropriate version of the JDK (Java 8 or later).
* **JavaFX Libraries**: Make sure the JavaFX SDK is set up properly in your project.
* **Images**: Card images are stored in the images folder. These should be named according to the format rank\_of\_suit.png, e.g., A\_of\_Spades.png, 2\_of\_Hearts.png.

### ****Detailed Breakdown****

#### 1. BaccaratApp **Class (UI and Game Flow)**

* The BaccaratApp class is the entry point for the game. It extends Application and sets up the UI with buttons for starting a round and quitting.
* The layout is managed using VBox for vertical alignment and HBox for horizontal alignment of the cards.

**Key Methods**:

* **start()**: Initializes the main stage, layout, and buttons.
* **startNewRound()**: Starts a new round by drawing two cards for both the Player and the Banker, and updates the UI to show the hands.
* **updateHandDisplay()**: Updates the displayed hands with images representing the drawn cards.

#### 2. Deck **Class (Deck Management)**

* The Deck class manages a deck of 52 cards. It stores the cards in a List<Card> and allows drawing of cards, shuffling the deck, etc.

**Key Methods**:

* **shuffleDeck()**: Shuffles the deck using Collections.shuffle().
* **drawCard()**: Draws the top card from the deck (removes and returns the first card).

#### 3. Card **Class (Card Representation)**

* The Card class represents a single card, containing a rank and a suit.

**Key Methods**:

* **getRank()**: Returns the rank of the card.
* **getSuit()**: Returns the suit of the card.

#### 4. Player **Class (Player and Banker)**

* The Player class represents the player (either the Player or Banker). It manages the cards in the player's hand and calculates the hand value according to Baccarat rules.

**Key Methods**:

* **addCard()**: Adds a card to the player's hand.
* **getHandValue()**: Calculates the hand value by adding the values of each card in the hand and applying the Baccarat mod 10 rule.
* **clearHand()**: Clears the player's hand for the next round.
* **getHand()**: Returns the player's hand (used to update the UI with the cards).

#### 5. **Images**

* The card images are stored in a folder named images located in the root directory of the project.
* Image filenames should follow the format: rank\_of\_suit.png (e.g., 2\_of\_Hearts.png, A\_of\_Spades.png).

### ****Steps to Run the Game****

1. **Download or Create Card Images**:
   * Ensure you have the card images for all 52 cards in the format rank\_of\_suit.png.
   * Place them in the images folder in the root directory of your project.
2. **Setup JavaFX in Your IDE**:
   * Make sure JavaFX is correctly configured in your project. You can download the JavaFX SDK from the official website if it’s not bundled with your JDK version.
   * Add the JavaFX libraries to your project’s classpath and ensure your IDE is configured to run JavaFX applications.
3. **Run the Application**:
   * Compile and run the BaccaratApp class in your IDE. A window should appear with the game's UI.
   * Press "Start New Round" to begin a round of Baccarat and see the results displayed.