UML Diagrams

System Context Diagram

Game Engine

Admin

Manage Players

Handle Game logic

Registered Player

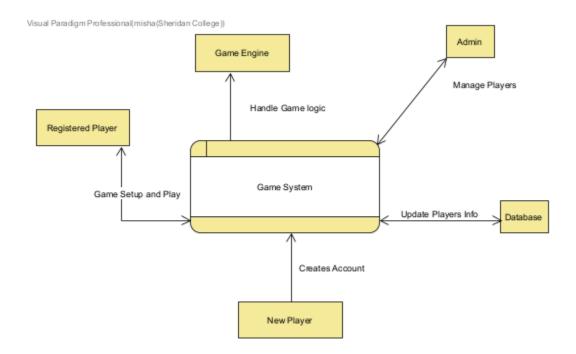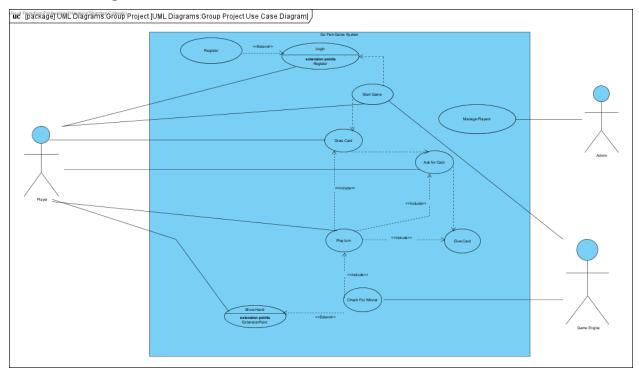Game System

Game Setup and Play

Update Players Info

Database

Creates Account

New Player

Use Case Diagram

Use Case Narrative

1. Player Registers a New Account

1.1. Player decides to cancel registration.

1.2. Registration success.

1.2.1. Player inputs registration details.

1.2.1.1. System validates details.

System creates a new user account.

1.2.1.2. System displays an error message if details are invalid.

1.2.2. Player cancels the registration process.

1.3. Registration failure.

System displays error.

2. Player Logs In

2.1. Player decides to cancel login.

2.2. Authentication success.

2.2.1. Player inputs username and password.

2.2.1.1. System verifies credentials.

System logs the player in.

2.2.1.2. System displays an error message if credentials are invalid.

2.2.2. Player cancels the login process.

2.3. Authentication failure.

System displays error.

3. Player Starts a New Game

3.1. Player decides to cancel starting the game.

3.2. Game start success.

3.2.1. Player selects the option to start a new game.

3.2.1.1. System initializes a new game session.

System shuffles the deck and deals cards.

3.2.1.2. System displays an error message if initialization fails.

3.2.2. Player cancels starting the game.

3.3. Game start failure.

System displays error.

4. Player Draws a Card

4.1. Player decides to cancel drawing a card.

4.2. Draw card success.

4.2.1. Player chooses to draw a card.

4.2.1.1. System checks if it's the player's turn.

System draws a card from the deck and adds it to the player's hand.

4.2.1.2. System displays an error message if it's not the player's turn.

4.2.2. Player cancels drawing the card.

4.3. Draw card failure.

System displays error.

5. Player Asks for a Card

5.1. Player decides to cancel asking for a card.

5.2. Ask for card success.

5.2.1. Player specifies the card to ask for.

5.2.1.1. System checks if it's the player's turn.

System prompts the targeted player to give the card if they have it.

5.2.1.2. System displays an error message if it's not the player's turn.

5.2.2. Player cancels asking for the card.

5.3. Ask for card failure.

System displays error.

6. Player Gives a Card

6.1. Player decides to cancel giving the card.

6.2. Give card success.

6.2.1. System prompts the player to give the requested card.

6.2.1.1. Player selects the card to give.

System transfers the card to the requesting player's hand.

6.2.1.2. System displays an error message if the player doesn't have the card.

6.2.2. Player cancels giving the card.

6.3. Give card failure.

System displays error.

7. Player Plays Their Turn

7.1. Player decides to cancel their turn.

7.2. Play turn success.

7.2.1. System notifies the player that it's their turn.

7.2.1.1. Player draws a card or asks for a card.

System processes the player's action and updates the game state.

7.2.1.2. System displays an error message if the action is invalid.

7.2.2. Player cancels their turn.

7.3. Play turn failure.

System displays error.

8. Player Shows Their Hand

8.1. Player decides to cancel showing their hand.

8.2. Show hand success.

8.2.1. Player selects the option to show their hand.

8.2.1.1. System displays the player's hand.

8.2.1.2. System displays an error message if the action is invalid.

8.2.2. Player cancels showing their hand.

8.3. Show hand failure.

System displays error.

9. Game Engine Checks for Winner

9.1. Player decides to cancel the check.

9.2. Check for winner success.

9.2.1. System checks the state of the game after each turn.

9.2.1.1. System evaluates if any player has met the winning condition.

System ends the game and declares the winner if found.

9.2.1.2. System displays an error message if no winner is found.

9.2.2. Player cancels the check for winner.

9.3. Check for winner failure.

System displays error.

10. Admin Manages Players

10.1. Admin decides to cancel managing players.

10.2. Manage players success.

10.2.1. Admin selects the option to manage players.

10.2.1.1. Admin adds, updates, or deletes player accounts.

System processes the changes and updates the player database.

10.2.1.2. System displays an error message if the action is invalid.

10.2.2. Admin cancels managing players.

10.3. Manage players failure.

System displays error.

Class Diagram

User

-username : String
-password : String

+login() : void
+logout() : void

Admin

+managePlayers() : void
+startGame() : void
+endGame() : void

New Player

+register() : void
+createAccount() : void

Registered Player

+askForCard() : void
+fishCard() : void
+showHand() : void

Game

-players : List<User>
-deck : Deck
-gameEngine : GameEngine

+startGame() : void
+playTurn() : void
+endGame() : void

Game Engine

+dealCards() : void
+shuffleDeck() : void
+checkWin() : void

Card

-rank : String
-suit : String

+getRank() : String
+getSuit() : String
+toString() : String

Deck

-cards : List<Card>

+shuffle() : void
+dealCard() : Card
+remainingCards() : int