**Team 7 (Vincent, Yvette). Lab 4. Planning write-up for a game of go fish**

We divided the work evenly. Vincent and Yvette worked together on the diagram during lab time, and we also got started with the algorithm. We assigned the tasks as follows:

- fix diagram as discussed: Vincent

- fix algorithm as discussed: Yvette

We then both finalized changes to the diagram and algorithm, and we agreed to split the code evenly. We both worked on the code and tested it, making several changes and updates as we tested the code. We decided to use insertion sort to keep player's hand sorted:

# Use insertion sort to keep the hand sorted after adding new cards

def sort\_hand(self):

rank\_order = {'2': 2, '3': 3, '4': 4, '5': 5, '6': 6, '7': 7, '8': 8, '9': 9, '10': 10, 'J': 11, 'Q': 12, 'K': 13, 'A': 14}

for i in range(1, len(self.hand)):

key\_card = self.hand[i]

key\_rank = rank\_order[key\_card.rank]

j = i - 1

while j >= 0 and rank\_order[self.hand[j].rank] > key\_rank:

self.hand[j + 1] = self.hand[j]

j -= 1

self.hand[j + 1] = key\_card

----------

And this is our high level algorithm: (note that in order to avoid a super long game, it was modified so that the game ends when either the deck runs out of cards or one player (who has books) runs out of cards).

**High-Level Algorithm:**

1. **Setup Game:**o Prompt the user to input the number of players (including computer).  
o Ask for names of human players.  
o Set up the computer as one of the players.  
o Deal 7 cards to each player if the number of players is 2–4. If there are 5–8 players, deal 5 cards each.

2. Game Flow:

o Shuffle the deck.

o In each turn:

§ A player (or computer) asks another player for a rank of card.  
§ If the other player has cards of that rank, they give them to the asking player.  
§ If the other player doesn’t have the requested cards, the asking player "Go Fish" (draws from the deck).

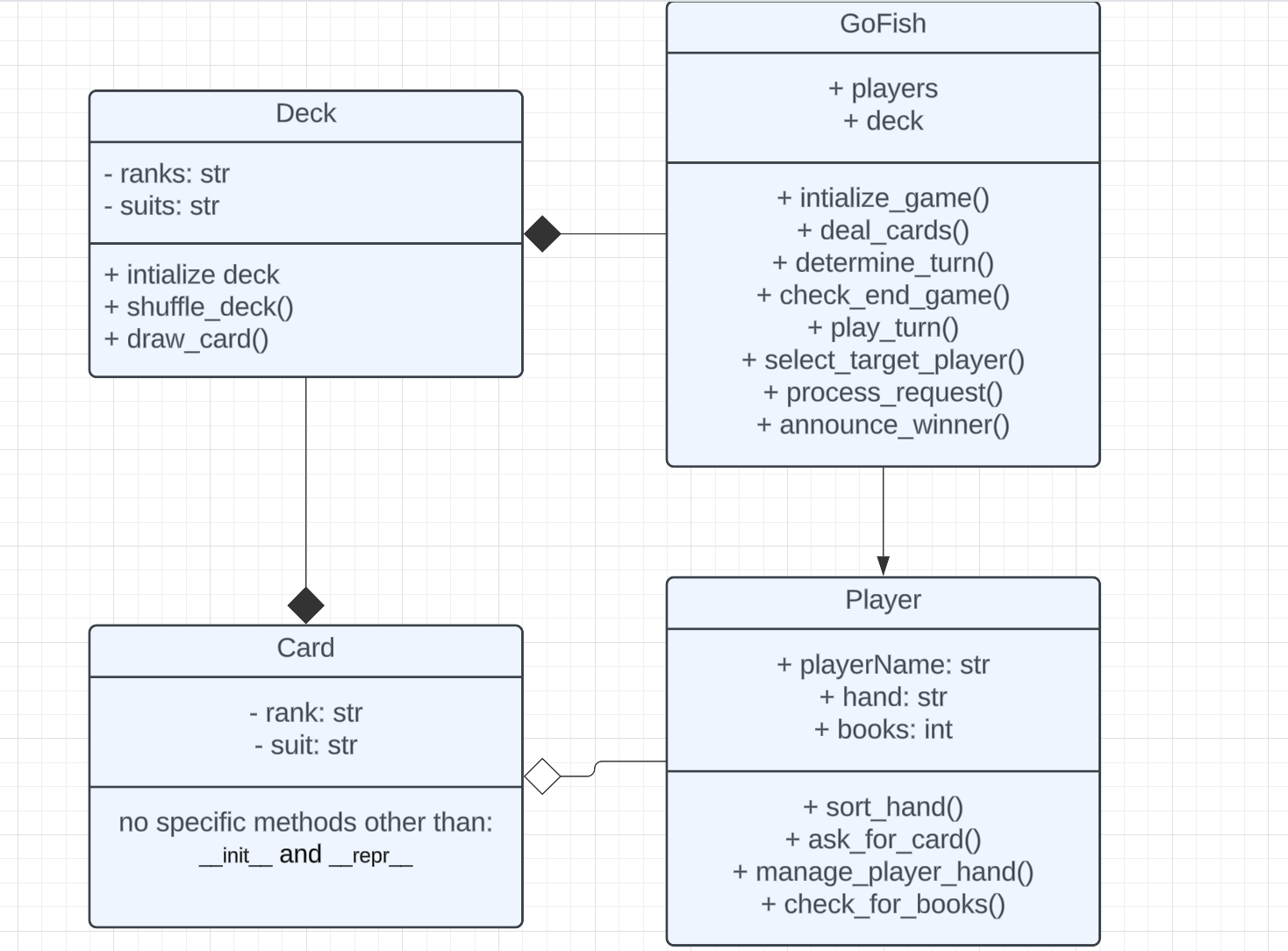
o Each player checks for sets of 4 cards of the same rank ("books").

o Player turns continue until the deck is empty or one player's hands are empty if that player has books.

o The game ends, and the player with the most books is the winner. There may be multiple winners.

3. **Classes:**o **Card:** Represents a single card with rank and suit.  
o **Deck:** Manages a shuffled deck of cards, including drawing cards and initializing deck.  
o **Player:** Manages a player's hand, including sorting using insertion sort, asking for cards, and checking for books.  
o **GoFish:** Manages the flow of the game, including turn-taking, checking game end, and displaying results.

**This is the diagram:**

****