Design Document:

Game Flow: The game flow is a simple loop that runs until the user decides to stop playing. Within each loop iteration, 13 hands are played, each with three rolls of five dice. After each roll, the user can choose which dice to keep and which to re-roll. After the third roll, the user selects a category to score the hand in, and the score is recorded on the scorecard. Once 13 hands have been played, the final score is displayed. The user can play up to 3 games.

Classes/Methods: The following classes will be implemented:

1. **YahtzeeGame** - This is the main class that runs the game. It contains the main method to play the game and a method to choose a category.
- play(): This method runs the game loop, prompting the user to play again after each game.
- chooseCategory(): This method prompts the user to select a category to score their hand in.

2. **ScoreCard** - This class represents the scorecard for the game.
- setScore(int category, int round, int score): This method sets the score for a given category and round.
- getScore(int category, int round): This method retrieves the score for a given category and round.

Data Structure for User Score: The ScoreCard class will use a two-dimensional array to store the user's scores. The array will have 13 rows (one for each category) and n columns (one for each round played).

Interaction between Game Flow and Scorecard: The YahtzeeGame class will interact with the ScoreCard class by calling its methods to set and retrieve scores for each hand played. The ScoreCard class will keep track of the user's scores and display them at the end of the game.

Relationships between Classes: The YahtzeeGame class will instantiate objects of the YahtzeeHand, YahtzeeScore, and ScoreCard classes as needed. The YahtzeeScore class will depend on the YahtzeeHand class to retrieve the current state of the dice, and the ScoreCard class will depend on the YahtzeeGame class to set and retrieve scores for each hand played.

For this I will be using a **console/ text implementation** rather than the graphical implementation.