**Deliverable 2 – Design Document**

**Contents:**

1. Overview
2. Use Case Narratives
   1. Use Case 1: Start Game
   2. Use Case 2: Play Round
   3. Use Case 3: Resolve War
   4. Use Case 4: Declare Winner
3. Design Considerations
   1. Encapsulation
   2. Delegation
   3. Cohesion
   4. Coupling
   5. Inheritance
   6. Aggregation
   7. Composition
   8. Flexibility/Maintainability

**Overview**

# Project Background and Description

This project is an extension of the **War** card game developed in Deliverable 1. The game is a two-player card game where each player draws a card from their deck, and the player with the higher-valued card wins both cards. If the drawn cards are of equal rank, a "War" is triggered, requiring each player to place additional cards to determine the winner. The game continues until one player collects all the cards or reaches a predefined condition (e.g., a maximum number of rounds).

## Scope of the Project

For Deliverable 2, the following enhancements are incorporated into the design:

* The game will terminate after **a maximum of 50 rounds** to prevent infinite play.
* A scoring system is implemented where each player’s score is based on the number

of rounds won.

* The implementation of a more robust **War** resolution system.
* Object-oriented design (OOD) principles are applied to ensure maintainability and scalability.

# Use Case Narratives

## Use Case 1: Start Game

**Actors:** Player 1, Player 2

**Description:** The game initializes with a shuffled deck, and players receive an equal number of cards.

**Main Flow:**

1. Players are prompted to enter their names.
2. The deck is shuffled and distributed.
3. The game begins with both players having an equal number of cards.

## Use Case 2: Play Round

**Actors:** Player 1, Player 2

**Description:** Each round, players draw and compare cards. **Main Flow:**

1. Both players draw their top card.
2. The cards are compared, and the player with the higher-value card wins.
3. The winning player collects both cards.
4. If a tie occurs, a "War" is triggered.

## Use Case 3: Resolve War

**Actors:** Player 1, Player 2

**Description:** When players draw cards of the same value, a war sequence begins. **Main Flow:**

1. Both players place three cards face down and one face up.
2. The new face-up cards are compared.
3. The winner collects all the played cards.
4. If another tie occurs, the War repeats.

## Use Case 4: Declare Winner

**Actors:** Game System

**Description:** The game determines the final winner. **Main Flow:**

1. The game ends when one player collects all cards or after 50 rounds.
2. The player with the highest number of collected cards wins.

# Design Considerations

## Encapsulation

* Each class has **private attributes** with **public getter and setter methods**,

restricting direct access.

* Example: The CardWar class encapsulates its suit and value attributes to maintain

data integrity.

## Delegation

* The WarGame class **delegates** deck management to the GroupOfCards class.
* The playRound() method in WarGame **delegates** card actions to PlayerWar

objects.

## Cohesion

* Each class is designed for **one specific purpose**.
* Example: PlayerWar manages player actions, while GroupOfCards manages the deck.

## Coupling

* The system uses **low coupling** by minimizing dependencies between classes.
* Example: WarGame interacts with PlayerWar and GroupOfCards but does not modify their internal workings.

## Inheritance

* The CardWar class extends Card, inheriting its behavior while adding game-

specific attributes.

* The PlayerWar class extends Player, implementing the play() method.

## Aggregation

* WarGame contains **two PlayerWar objects**, representing a **whole-part relationship**.
* The GroupOfCards class aggregates multiple CardWar objects.

## Composition

* The PlayerWar class **owns its hand of cards**, meaning if a player is removed,

their hand is also discarded.

* The WarGame class is responsible for the lifecycle of GroupOfCards and PlayerWar.

## Flexibility/Maintainability

• The design allows future expansion, such as:

 Adding a graphical user interface (GUI).

 Supporting additional card game variations.

 Extending the game logic for multiplayer mode.

**Narrative:-**

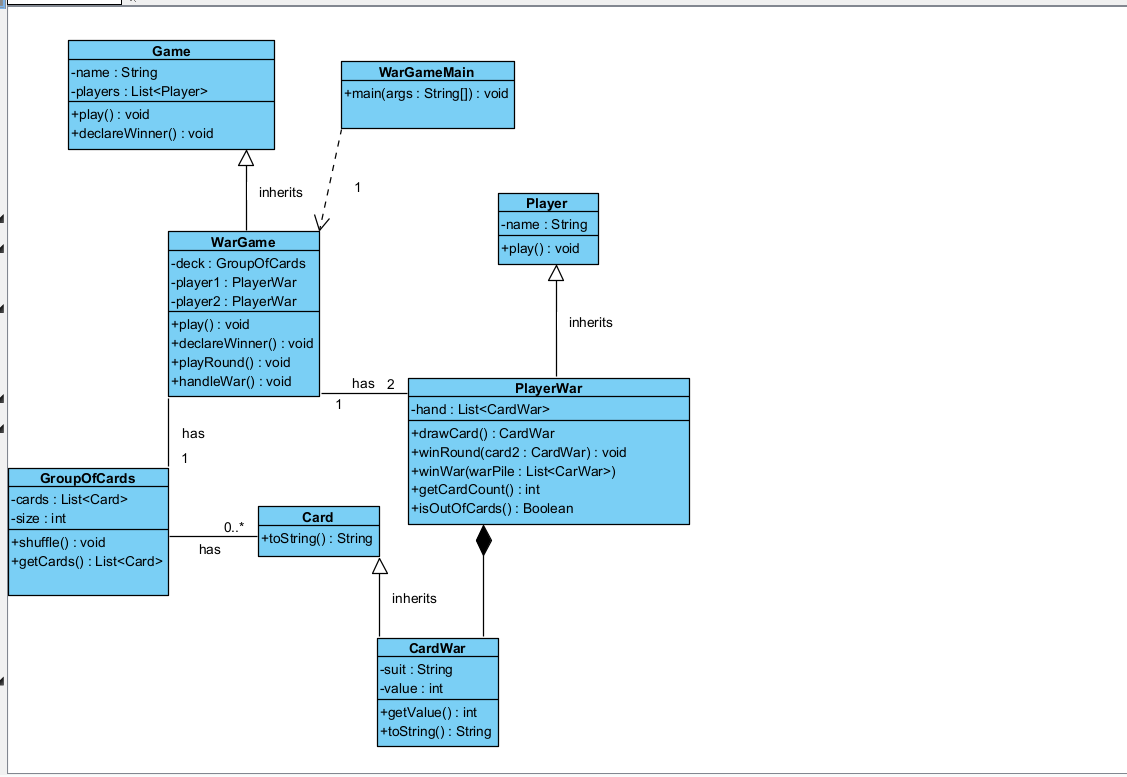
The War Game starts with both participants registering their names. The system then prepares by shuffling the deck and giving each player an equal card distribution. Once setup is complete, gameplay begins.

In each turn, players simultaneously reveal their topmost card. The person showing the higher value wins that round and collects both cards. When identical values appear, a "War" situation occurs. During War, each player places three cards face-down before revealing a fourth card. The higher fourth card determines who claims all eight cards in play. If these fourth cards also match, the War process continues until someone wins.

Play continues until either 50 rounds have elapsed or one player possesses all cards. At conclusion, the system calculates card totals and announces the participant with more cards as the victor.

This format balances luck and tactics, creating an engaging experience where fortunes can quickly change throughout the contest.

Diagrams:-



User case diagram:-   
