# Deliverable 1: Contract, Repository, UML Diagrams & Design Document

Bhavikkumar Pater, Darren Scarfo, Kevin To, Neil-Bryan Caoile

Faculty of Applied Science & Technology, Sheridan College

SYST17796 Fundamentals of Software Design and Development

Megha Patel

June 24, 2021

Contents

[Deliverable 1: Contract, Repository, UML Diagrams & Design Document 1](#_Toc75023228)

[Project Background & Description 3](#_Toc75023229)

[Project Scope 4](#_Toc75023230)

[High-Level Requirements 4](#_Toc75023231)

[Implementation Plan 5](#_Toc75023232)

[Design Considerations 5](#_Toc75023233)

[References 5](#_Toc75023234)

## Project Background & Description

The goal for the group known as the “Warriors” (also known as, the “Group”), is to successfully work together, with individual members leveraging their experiences, skill, and expertise, in order to develop a software application (the “Project”) in accordance with the requirements of the SYST 17796 course. More specifically, the end-goal of the Project is to develop a fully functional, playable, and entertaining version of the card game “War” using a console for game input and output.

The game of “War” is what is known as an “accumulating card game,” where the goal is for one player to “win” cards off the other player in a series of discrete turns or rounds. More specifically, according to Bicycle Cards (2021), War:

* Is played with a standard 52 deck of cards,
* Is played by only two players,
* Assumes that aces are ‘high’, meaning they have the highest value in the deck.

The game begins with the two players being dealt all 52 cards evenly and face down.

The game then occurs over a series of rounds in which each of the two players turn up a single card at the same time, with each player unaware of the value of the card that is being turned over. The players then compare cards, with the player who turned over the higher value card taking both cards, and putting both cards face-down on the bottom of their card deck. Players are not to reshuffle their decks throughout the game.

Notably, if the two cards that the two players turn over are of the same value, the game-state known as “War” begins. During “War,” each player turns up one card face down, and one card face up; the cards are then compared, with the player with the higher-value cards taking both piles of cards (i.e., all six cards). It should be noted that “War” can occur a second time (or even third or fourth time) if the cards are again of the same value, with the winning player then taking all cards being turned over on that round.

As described above, the goal of the game is to be the first player to collect all 52 cards in the deck via winning the individual rounds and the “War” rounds that comprise the game.

The starting code from which we will build the Project is provided by our instructor, known as ‘SYST 17796 Project Base Code’. The initial authors of the starting code are the following:

* Dancye
* Paul Bonenfant Jan 2020
* Megha Patel

The game will be developed using the Java Platform, Standard Edition 16.0, using the Apache NetBeans 12.3 IDE, and the Visual Paradigm Professional v16.3 computer-aided software engineering suite.

Coding conventions include the following:

* Standard Java naming conventions (i.e., camelCase, capitalized constants, classes beginning with uppercase, etc.)
* Brace style: end-of-line for opening braces, closing brace on own line,
* Indentation: one-tab character, being equal to 4 spaces in size,
* No lines longer than 80 characters,
* Wrapping lines as required, breaking before commas and operators, indenting two-tabs (the equivalent of 8 spaces) after a line break.,
* Commenting to include

## Project Scope

The development team members include:

* Bhavikkumar Patel – Team Lead
* Darren Scarfo
* Kevin To
* Neil-Bryan Caoile

As outlined in the Group Contract (previously submitted by the Team Lead), each of the four team members will agree on work assignments, without highly rigid roles; decisions will be made largely democratically whenever possible to ensure that the team is successful.

The Project will be developed using the IDE console for play, and will not include a graphical user input component at this time. The Project will be completed when the code is bug-free, games of “War” can be completed successfully, rules are consistently and correctly implemented, with all user-facing messages being completed.

## High-Level Requirements

The system must include the following:

* The ability for players to see a text-based version of the game board,
* The ability for players to tell the program that they are ready to “draw” the next card,
* The ability for players to know what cards were drawn, and which player “won” the round,
* The ability for players to enter their names in the game,
* The ability for the players to know who has currently won more cards,
* A notification that the “War” game state has begun,
* A notification that the “War” game state has ended,
* The ability for players to know how many turns have taken place since the start of the game,
* A final “win” or “loose” output, informing the players who has won, how the win-state was achieved, and in how many turns the game was played.

## Implementation Plan

The URL to our Git repository can be accessed at the following link:

https://github.com/Bhavik0997/Warriors

The Git repository will be used by all Group members, with each Group member checking in code at the end of each week or when required by other Group Members.

Text files are stored under a separate directory, code, UML diagrams have their own folders etc.

Include information on coding standards you intend to follow and tools you expect to use (VP, NetBeans, eclipse,

Junit...)

## Design Considerations

Talk about how the current code is structured as it relates to the following OO principles. Each principle should have 2 or 3 specific examples from the base code or your intended additional code (i.e. potential for improvement).

•Encapsulation

•Delegation

•Flexibility/Maintainability

# References

https://bicyclecards.com/how-to-play/war/