Sheridan College, Davis Campus

SYST17796 Fundamentals of Software Design and Development

Group 10: Project Deliverable-1

Jay Patel (991777023)

Jasleen Kaur (991774542)

Lakshit (991760767)

Team Contract

SYST 17796 TEAM PROJECT

Team Name: _____Group 10_____

Please negotiate, sign, scan and include as the first page in your Deliverable 1.

Please note that if cheating is discovered in a group assignment each member will be charged with a cheating offense regardless of their involvement in the offense. Each member will receive the appropriate sanction based on their individual academic integrity history.

Please ensure that you understand the importance of academic honesty. Each member of the group is responsible to ensure the academic integrity of all of the submitted work, not just their own part. Placing your name on a submission indicates that you take responsibility for its content.

For further information, read Academic Integrity Policy here :
https://caps.sheridancollege.ca/student-guide/academic-policies-and-procedures.aspx

Team Member Names (Please Print)	Signatures	Student ID
Project Leader: Jay Patel		991777023
Jasleen Kaur		991774542
Lakshit		991760767

By signing this contract, we acknowledge having read the Sheridan Academic Integrity Policy

Responsibilities of the Project Leader include:

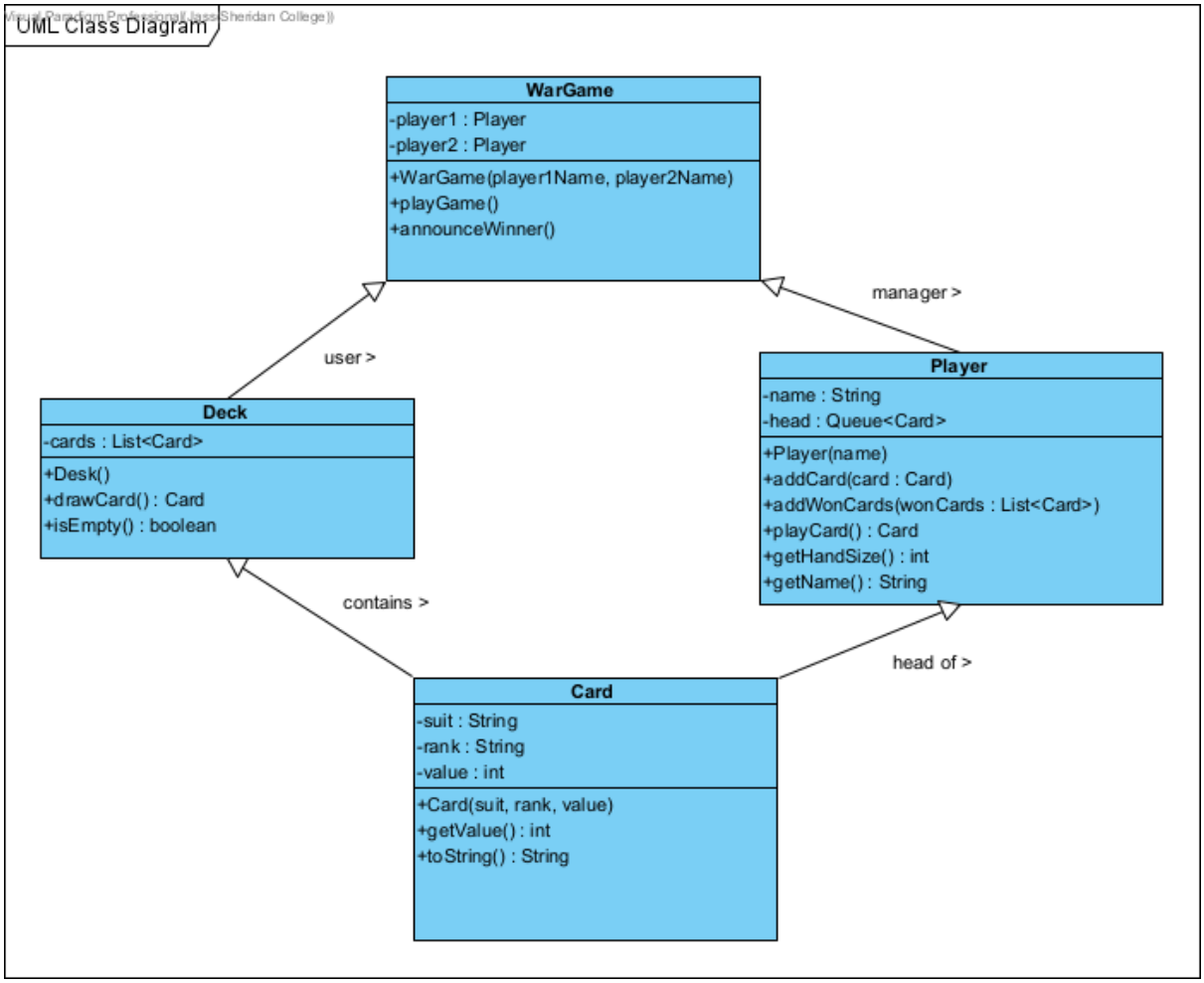
- Assigning tasks to other team members, including self, in a fair and equitable manner.
- Ensuring work is completed with accuracy, completeness and timeliness.
- Planning for task completion to ensure timelines are met.
- Notifying the professor of any issues in a timely manner so that corrective measures can be taken.
- Any other duties as deemed necessary for project completion.

What we will do if . . .

Scenario	**Accepted initials**	**We agree to do the following (Put an X corresponding to your choice in each box)**
Team member does not regularly attend team meetings and/or does not respond to communications in a timely manner.		Project leader emails the student citing the concerns and cc's the professor so they are aware of the situation at the very onset _X_ (**Mandatory**). a) __ In addition to above, the leader/team will (add your own content here):
Team member does not deliver component on time due to severe illness or extreme personal problem.		a) Team absorbs workload temporarily __ b) Team seeks advice from professor __ c) Team shifts target date if possible __ d) __ Other (specify):
Team member has difficulty delivering component on time due to lack of understanding or ability.		a) Team reassigns component __ b) Team helps member __ c) Team member must ask professor for help __ d) __ Other (specify):

Scenario	**Accepted initials**	**We agree to do the following (Put an X corresponding to your choice in each box)**
Team member does not deliver component on time due to lack of effort.		a) Team absorbs workload __ b) Team member(s) ask professor to request a Participation Form from <u>all</u> team members. This *may* result in individualized grades being awarded for a deliverable __ c) Both a. and b. above __ d) __ Other (specify):
Team cannot achieve consensus leaving one or more member(s) feeling that their voice(s) is/are not being heard in a decision which affects everyone.		a) Team agrees to abide by majority vote __ b) Team seeks advice from the professor __ c) __ Other (specify):
Team members do not share expectations for the quality of work on a particular deliverable.		a) Team members will draw on each other's strengths to help bring the quality of the deliverable to a minimal acceptable level __ b) Team votes on each submission's quality __ c) Team member(s) ask professor to request a Participation Form from all team members, which may result in individualized grades being awarded for a deliverable __ d) __ Other (specify):
Team member behaves in an unprofessional manner, e.g. being rude, uncooperative and/or making one or more		a) Team agrees to avoid use of all vocabulary inappropriate to a business/college setting __

Scenario	Accepted initials	We agree to do the following (Put an X corresponding to your choice in each box)
member(s) feel uncomfortable.		b) Team attempts to resolve the issue by airing the problem at a team meeting __ c) Team requests a meeting with the professor to discuss further __ d) __ Other (specify):
There is a dominant team member who insists on making all decisions on the team's behalf leaving some team members feeling like subordinates rather than equal members		a) Team will actively solicit consensus on all decisions which affect project direction by asking for each member's decision and vote __ b) Team will express subordination feelings and attempt to resolve issue __ c) Team seeks advice from the professor __ d) __ Other (specify):
Team has a member who refuses to participate in decision making but complains to others that s/he wasn't consulted		a) Team forces decision sharing by routinely voting on all issues __ b) Team routinely checks with each other about perceived roles __ c) Team discusses the matter at team meeting __


UML Class Diagram
WarGame
-player1 : Player
-player2 : Player
+WarGame(player1Name, player2Name)
+playGame()
+announceWinner()
manager >
user >
Player
-name : String
-head : Queue<Card>
+Player(name)
+addCard(card : Card)
+addWonCards(wonCards : List<Card>)
+playCard() : Card
+getHandSize() : int
+getName() : String
Deck
-cards : List<Card>
+Desk()
+drawCard() : Card
+isEmpty() : boolean
contains >
head of >
Card
-suit : String
-rank : String
-value : int
+Card(suit, rank, value)
+getValue() : int
+toString() : String

SYST 17796 DELIVERABLE 1

DESIGN DOCUMENT TEMPLATE

OVERVIEW

1. Project Background and Description

Players of the basic two-player War card game compare the value of their drawn cards. The round's winner with the greater value card takes both cards. Should both players draw cards with identical value, a "war" takes place whereby extra cards are chosen till a winner is decided. The game keeps on until one player gathers every card.

Base Code Description:

The provided base code impleemnts the fundamental blocks of the War game. It includes:

- Card class: Shows one individual playing card.
- Deck class: Manages a 52 card shuffled deck.
- Player class: shows a player together with their card queue.
- WarGame class: Manages game player, includ card comparison and war scnario.

The code designe in Java and work with object-oriented programming (OOP) principles.

2. Project Scope

Team Members & Roles

- [Jay Patel] - Development
- [Jasleen Kaur] – UML Class Diagram
- [Lakshit] - Documentation

Technical Scope

Designed to let two players engage in a turn-based game of War, the Java console-based card game is Following object-oriented programming ideas guarantees modularity and maintainability in the application. Each player chooses a card from their deck, and the game decides the winner of a round depending on card value via a command-line interface, therefore organizing player interactions. Should a tie, the system implements a "war" situation whereby more cards are selected and matched until a winner is decided. The game keeps on until one player has gathered all the cards, guaranteeing a whole gaming cycle. In this regard, the game ends when every participant has gotten a card. After that, the cards are compared to decide the round winner at the end of this process.

The architecture consists in a Card class to represent individual cards, a Deck class to handle card distribution, a Player class to manage every player's hand, and a WarGames class to control the general game flow.

The system is a console-based card game that:

- lets two gamers engage in War.
- Sort the drawn cards to find the round winner.
- manages "war" situations whereby players draw more cards.
- names the winning player by determining who of the two players has the highest card.

3. High-Level Requirements

The game must:

- Two players to play War.
- Handle card comprisons and determne round winers.
- Manage the "war" scenario when equal-value cards are drawn.
- Announce the winner.

4. Implementation Plan

Git Repository

- Repository URL: **https://github.com/pateljay7018/SYST17796-Group-10.git**
- Tools IDE: NetBeans
- Coding: Java

5. Design Considerations

Encapsulation

- The Card class has private characteristics such suit, rank, value with getters.
- The Deck class only permits controlled access using methods and maintains the private list of cards.

Delegation

- The WarGames class assigns actions to the Player and Deck classes.
- The Player class guarantees correct card retrieval and controls the hand of the player.

Flexibility & Maintainability

- The code is set to let simple changes—like including other card games—easily possible.
- Other card-based applications can make advantage of the Deck and Player classes.

6. UML Class Diagram