



SYST17796 Fundamentals of Software Design & Development

Project Deliverable #3

### INSTRUCTIONS

In Deliverable #2 you committed to a design for your game and you will have received feedback on the design. The first step for Deliverable #3 is to incorporate that feedback so you have a firm UML class diagram of the final game so that you can begin to code.

The same standards for groupwork, professional writing style and citations apply to all deliverables. If you have questions, you can refer to the project description or ask your instructor. Any groupwork conflicts will be dealt with using the contract from Deliverable #1.

To begin, you will update your UML Class Diagram, incorporating any identified improvements to your Deliverable #, including methods and attributes for each class.

Next, complete the code to satisfy the rules of the game (requirements) and the use cases you defined (scope). Ensure the final version is checked-in to your Git repository.

**PROJECT BACKGROUND AND DESCRIPTION**

The rules of the card game, war, was briefly described in deliverable one. In more depth, the game rounds are based on the number of cards each player has per round, the game is over when one player collects all the cards, and the other loses all the cards. In this case the player who have collected all the cards is the winner, the other player is loser, the game is over. This will result in a game over display that displays total score end of the game of the player who won the game, and the name of that player.

Following are the user stories with acceptance criteria as known as high level requirements, use case diagram that was made based on the high-level requirements and an updated class diagram:

**FIGURE 1: HIGH LEVEL REQUIREMENTS**

|  |  |
| --- | --- |
| Business Requirements | User Stories |
| Ability for each player to register with the game | As a player, I want to register with the game, so that I can participate in the game.  Acceptance Criteria: Player is successfully registered in the game before entering the game session and is issued a set of cards. |
| Ability for the game to communicate a win or loss | As a player, I want to know the outcome of the game, so that I know whether I won or lost a game.  Acceptance Criteria: When game over, total score and player winning condition must be displayed. |
| Ability for players to know their status (score) at all times | As a player, I want to know the score of the game at any time, so that I can know the status at all times.  Acceptance Criteria: Display each player score. |
| Ability for drawing the card from the set | As a player, I want to draw the card each round of the game session, so that I can play my turn in the current round.  Acceptance Criteria: Display one drawn card per player each round in normal game mode. |
| Ability for drawing four cards from the set, top one facing up | As a player, I want to draw four cards if the cards. match for the round, so that I can play my turn for war in current round of war.  Acceptance Criteria: Display four drawn card per player each round in war game mode. |
| Ability for entering a game session | As a player, I want to enter a game session so that I will know that I am in a game.  Acceptance Criteria: Players must be displayed with the number of their cards to confirm they have successfully entered the game session. |
| Ability for viewing opponent’s name while in the active game session | As a player, I want to know the name of my opponent, so that I can know whom I am playing against.  Acceptance Criteria: Display the opponent’s name in the game session |
| Ability for viewing active turn for player while in the active game session | As a player, I want to know who’s turn is it, so that I can know when it is my turn.  Acceptance Criteria: Display the active turn |
| Ability for viewing number of cards in the players pile or stack | As a player, I want to know how many cards I have left per round, so that I can keep track of my cards.  Acceptance Criteria: Display number of cards for the player. |
| Ability for player to know they are in active war | As a player, I want to know the type of game mode, so that I know whether I am in a war mode or a normal mode of the game mode, in the game session.  Acceptance Criteria: Display game mode in the game session. |
| Ability for player to know the round has ended | As a player, I want to know that the round has ended, so that I can keep track of each rounds.  Acceptance Criteria: Display number of rounds in the game session. |

**FIGURE 2: USE CASE DIAGRAM**

This user case diagram is based on the high-level requirements and the code that is currently under development, keeping in mind of the high-level requirements.

Diagram

Description automatically generated

**FIGURE 3: CLASS DIAGRAM**

The following class diagram is based on the business requirements (high-level requirements) and the code for the game, that is currently in development.

Diagram

Description automatically generated

**GIT REPOSITORY LINKS:**

Below are the git repositories for actual code for the game as well as JUnit tests.

**GIT REPO FOR CODE:**

<https://bitbucket.org/AdeelKhilji/cardgameproject_war/src/master/>

**GIT REPO FOR JUNIT TEST:**

<https://bitbucket.org/AdeelKhilji/cardgameproject_war/src/master/test/cardgame/war/controller/>

### SUBMISSION

Please submit one PDF document per group as well as the access credentials (URL pointing to your repo and make sure the Instructor has read access before submission) to your Git Repository. This means that the document should be professionally organized and have a uniform style throughout. It should look as though it came from one team, not separate students. Please note that instructors may choose to run your submission through TurnItIn or compare the submission with other students from other sections/semesters for academic integrity purposes. Please take the time to properly cite your sources.

You should also include a zip export of your project including the source code for your game labelled with your group name.

### RUBRIC

|  |  |  |  |
| --- | --- | --- | --- |
| ITEM | CRITERIA | POINTS | WEIGHT |
| Class Diagram | Updated to reflect feedback from Deliverable 2, methods and attributes included. Notationally correct and solid design as described in Deliverable #2 Design Document. | 10 | 10% |
| Source Code | Code is completed and conforms to the design specified by the class diagram. Code is functionally correct. Code follows standard coding conventions for comments, naming and indentation. | 10 | 30% |
| Source Code Design | The Design produced follows the principles of OOD studied in the course and provides flexibility, reusability and efficiency. | 10 | 30% |
| Game Playability | The application has enough functionality to be playable as a game. | 10 | 30% |

### DOCUMENT LAYOUT

One PDF with the following sections denoted using page numbers, headers and a table of contents:

1. Updated Class Diagram with methods and attributes.
2. Reference to the Git repository containing your source code.
3. Reference to the Git repository location (directory) of your JUnit tests.

AND One zip archive containing the source code for your game and your JUnit tests, labelled with your group name.