Team Contract

SYST 17796 TEAM PROJECT

Team Name: Team 7

Please negotiate, sign, scan and include as the first section 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 honesty 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 Honesty Policy on AccessSheridan or visit the faculty office and speak with the Program Support Specialist.

Team Member Names (Please Print)	Signatures	Student ID
Project Leader: Danish Chaudhry		991577142
Luqman Jehee		991576787
Hitish Mahay		991583225
Lam Bich		991578047

By signing this contract, we acknowledge having read the Sheridan Academic Honesty Policy as per the link below.

https://policy.sheridanc.on.ca/dotNet/documents/?docid=917&mode=view

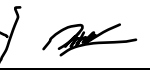
Responsibilities of the Project Leader include:

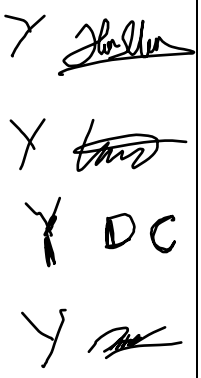
- Assigning tasks to other team members, including self, in a fair and equitable manner.
- Ensuring work is completed with accuracy, completeness and timeliness.
- Planning for task completion to ensure timelines are met
- Any other duties as deemed necessary for project completion

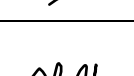
What we will do if . . .

Scenario	Accepted Y/N + initial	We agree to do the following
Team member does not deliver component on time due to severe illness or extreme personal problem	Y Y Y DC Y	a) Team absorbs workload temporarily ✓ b) Team seeks advice from professor __ c) Team shifts target date if possible __ d) Other:
Team member cannot deliver component on time due to lack of ability	Y Y Y DC Y	a) Team reassigns component ✓ b) Team helps member __ c) Team member must ask professor for reference material __ d) Other:
Team member does not deliver component on time due to lack of effort	Y Y Y DC Y	a) Team absorbs workload ✓ b) Team "fires" team member by not permitting his/her name on submission __

		c) Other:

Scenario	Accepted Y/N + initial	We agree to do the following
Team member does not attend team meeting	Y Y Y DC Y	a) Team proceeds without him/her and will assign work to the absent member ✓ b) Team doesn't proceed and records team member's absence __ c) Team proceeds for that meeting but "fires" member after __ occurrences __
A piece of production equipment fails such as a printer, disk drive, or laptop	Y Y Y DC Y	a) Backup copies will be made and kept in the college ✓ b) A locker or "share" directory will be used for joint access __ c) A photocopy and duplicate disk of all deliverables will be made __ d) Other:
An unforeseen constraint occurs after the deliverable has been allocated and scheduled (a surprise test or assignment)	Y Y Y DC Y 	a) Team meets and reschedules deliverable ✓ b) Team will cope with constraint __ c) Other:
Team cannot achieve consensus leaving one member feeling "railroaded",	Y Y	a) Team agrees to abide by majority vote ✓

"ignored", or "frustrated" with a decision which affects all parties		b) Team flips coin __ c) Other:
Team members do not share expectations for grade desired		a) Team will elect one person as "standards-bearer" who has the right to ask that work be redone ✓__ b) Team votes on each submission's quality __ c) Team will ask for individual marking and will identify sections by author __ d) Other:

Scenario	Accepted Y/N + initial	We agree to do the following
Team member behaves in an unprofessional manner by being rude or uncooperative	Y Y Y DC Y	a) Team attempts to resolve the issue by airing the problem at team meeting ✓__ b) Team requests meeting with professor to problem-solve __ c) Team ignores behaviour __ d) Team agrees to avoid use of all vocabulary inappropriate to the business setting __
Team member assumes or requests that his/her name be signed to a submission but has not participated in production of the deliverable	Y Y Y DC Y 	a) Team agrees that this is cheating and is unethical ✓__ b) Friends are friends and should help each other __ c) Team will submit with signature but will advise professor who will take action __
There is a dominant team member who is content to make all decisions on the team's behalf leaving some team members feeling like subordinates rather than equal members	Y Y Y DC Y	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) Other:
Team has a member who refuses to participate in decision making but complains to others that s/he wasn't consulted	Y Y Y DC Y	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 __

SYST 17796 DELIVERABLE 1

Table OF Contents

Syst 17796 Deliverable 1 1

Unified Modeling Language Diagram 3

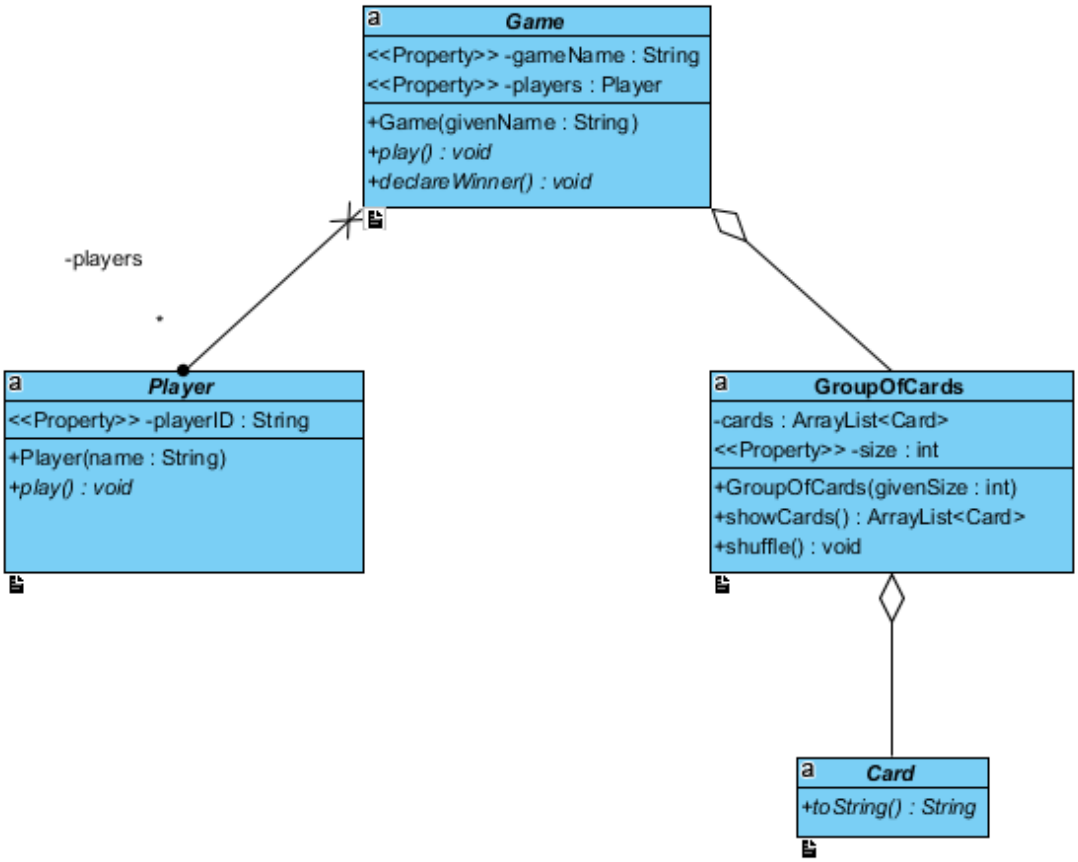
Case Diagrams 4

1. Player Case Diagram 4
2. Dealer Case Diagram 5

Design Document 6

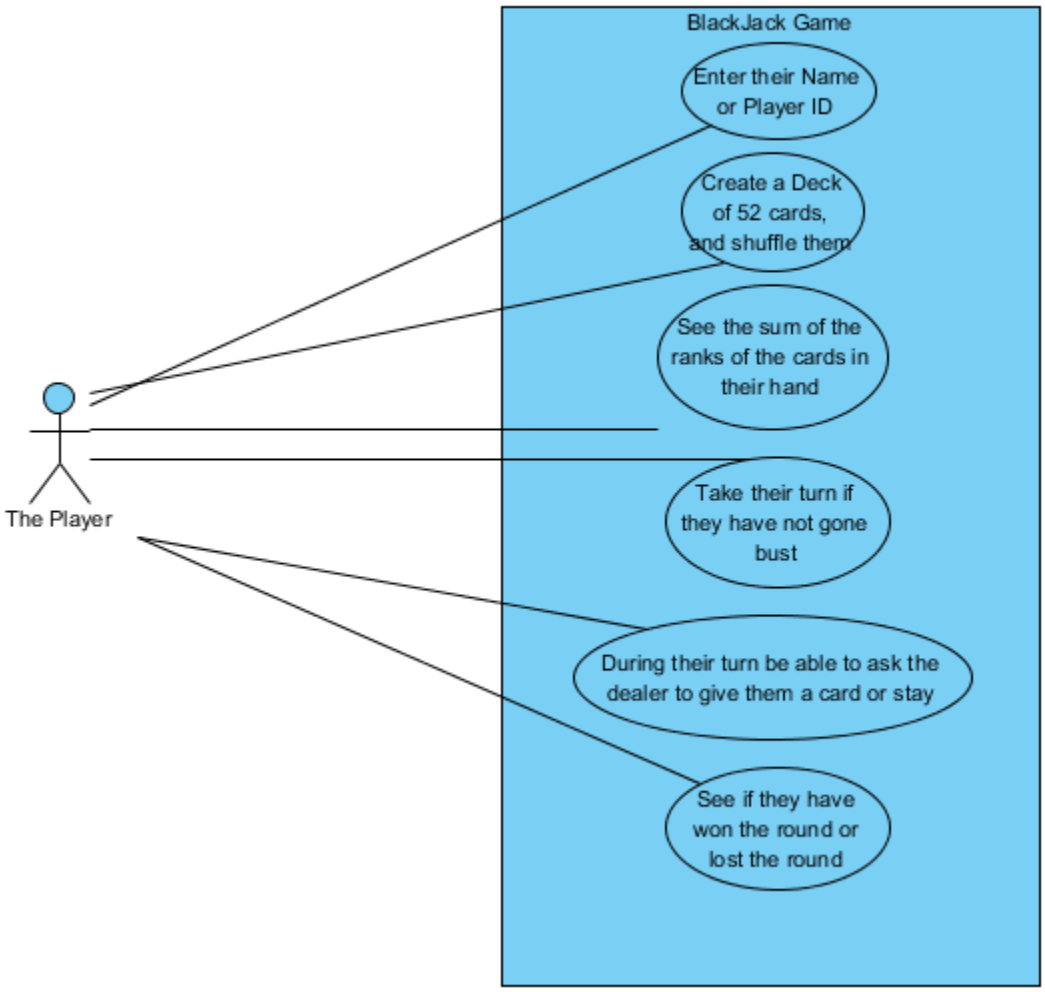
1. Project Background and Description 6
2. Project Scope 7
3. High-Level Requirements 8
4. Implementation Plan 8
5. Design Considerations 9

UNIFIED MODELING LANGUAGE DIAGRAM

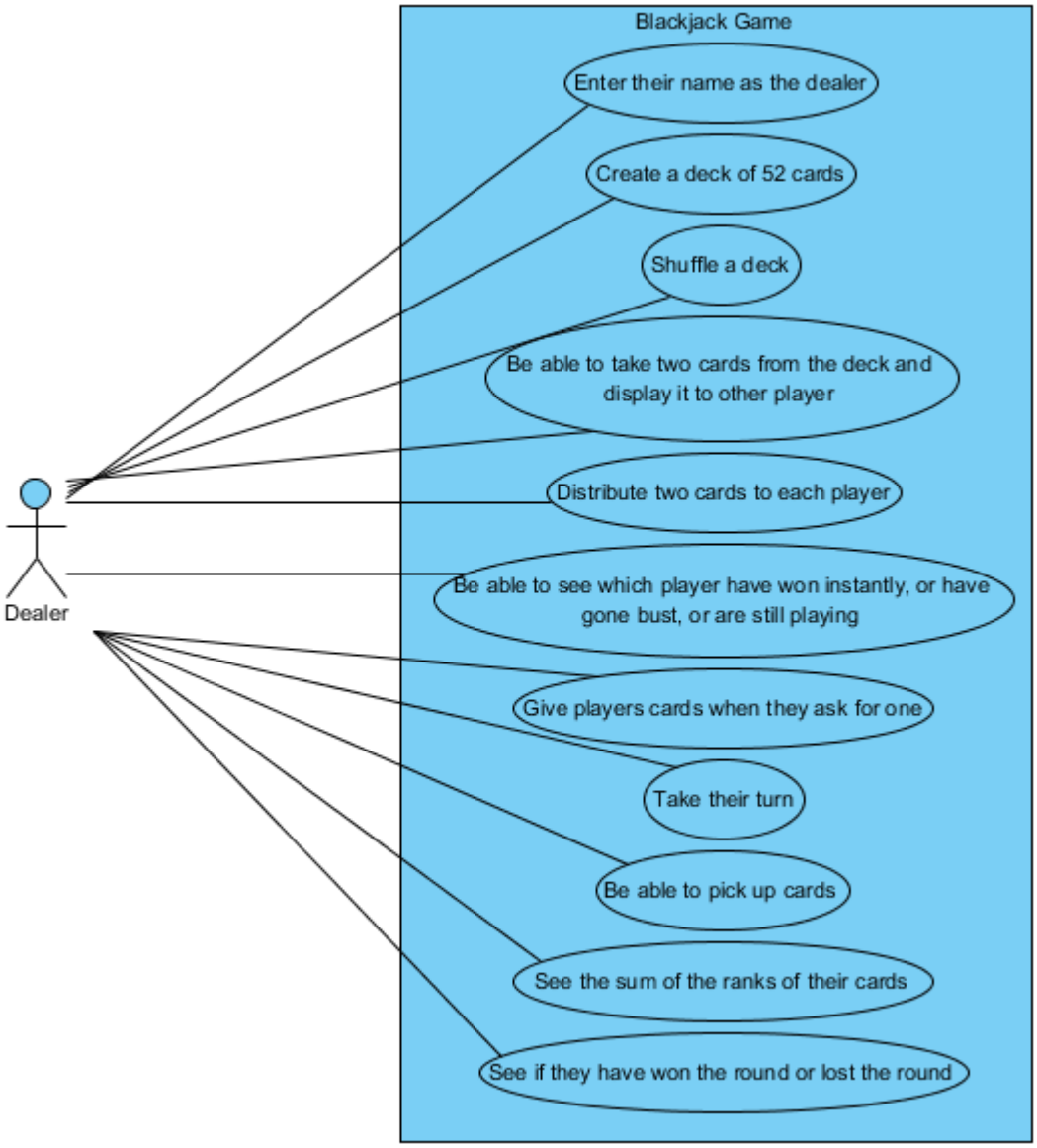


CASE DIAGRAMS

1. Player Case Diagram



2. Dealer Case Diagram



DESIGN DOCUMENT

1. Project Background and Description

Describe the project goals and final vision. Include a brief description of how to play the game you have chosen and a reference to the rules of the game you have chosen. Also describe the current starting base code. Use technical terms to describe the code including what language it is written in, any patterns you can see and any coding conventions used.

Response:

For our game, we have chosen Blackjack. We will create a computer program that allows five players to play a single round of Blackjack against a computer controlled dealer. The game will follow the rules used commonly in casinos.

The goal of a black jack is simple to get the sum of the rank of cards in a player's hand equal or as close to 21 as possible, without going over 21. The game is played against the dealer. If the dealer's hand sum is closer to or equal to 21, then the players lose. Our game will be played as such:

1. The dealer will pick two cards from the top of a shuffled deck of 52 cards.
2. The dealer will place the first card they picked face down on the board, and they will place the second card they picked face up on the board.
3. The dealer will then give each player two cards from the top of the deck placing them face up on the board.
4. Any player (other than the dealer) whose hand total is greater than 21 automatically loses the round, any player whose hand is equal to 21 automatically wins the round, and any player whose hand is less than 21 continues to play the round.
5. Now it’s the players' turn. They will take their turns in descending order. So, player 1 will go before player 2, and player 2 will go before player 3, and so on.
6. If the player's sum is equal to 16 or below 16, they must pick up cards until their sum is above 16. If their sum is above 16 and below 21 they have two options to get a card (hit me) or to end their turn and keep whatever total sum they have (stay).
7. During any player's turn if their sum goes above 21 (bust) they lose, or if their sum equals 21 they automatically win the round.
8. After all the players take their turn, it is the dealer's turn.
9. The dealer will reveal their face down card, and if their total is 21, all the remaining players lose the round, and if the dealer's sum is above 21, all the remaining players win the round.
10. If the dealer's sum is less than 21 and it is below 16 or equal to 16 they must pick up cards until their sum is above 16. After this if the dealer's sum goes over 21 the remaining players win the round or if their sum equals 21 the remaining players lose the round.
11. Once the dealers sum is above 16 and below 21, all the players compare their sums against the dealer's sum.
12. If the player's sum is above the dealer's sum they win the round and if their sum is below the dealer's sum they lose the round.

As shown above the game rules are simple and there is no ambiguity in the game flow. The game will be written in the programming language java. More requirements may be added as the project moves forward, this depends on the project members growing capabilities in programming using the java language.

The current starting base code has four classes. The Player class is an abstract class which represents card players, the Card class is an abstract class which represents the cards, the Game class is an abstract class which represents a card game, and the GroupOfCards represents the deck used for playing a card game.

2. Project Scope

Member Names	Responsibilities
Danish	• Will be the team lead in charge of setting the schedule and distributing workload to members • Will write the first iteration of a working game to prove such a game can be written in java, also to convert the game rules in to programming logic • Will aide in applying OOP rules to the future iterations, in order to improve the code for the next iteration
Hitish	• Will be in charge of designing UI and UX of the program • Will be in charge of handing in the deliverables in proper format and on due date • Will be in charge on maintaining and organizing the Github repository • Will create the various case diagrams needed for design
Lucky	• Will be the primary in applying OOP rules to the code • Will improve written code by rewriting it using better logic • Will aide in creating proper program documentation • Will maintain and create the UML diagrams
Lam	• Will test the program logic, and document the results of the tests • If faulty logic discovered will re-write code that will prevent that • In charge of documenting the program code and maintaining it throughout the project development

Currently the interface of the project will be simple and displayed using the console. The interface commands will be inputted through the keyboard; the Blackjack players will take their turn using the keyboard. Majority of the project time will be spent writing program code that would allow the player's to play the game using the Blackjack rules stated above. Once the program code is written the primary objective of the game will be complete, and the project could be concluded as finished.

Despite, the primary object being a playable Blackjack game, if this game's logic is coded and completed earlier than expected; then more time will be spent on creating a UI which would allows the user to play the game using the mouse rather than the keyboard, and shows the results using images rather than console text. This will be a secondary objective and optional, it will depend on the members desire to further develop the project.

3. High-Level Requirements

The new system must include the following:

- Create a deck of 52 playing that can then be shuffled at any given moment in game
- Create a program which will allow up to 5 players to play a single round of Blackjack against an automated dealer
- Ability to create five blackjack player, each with their own unique names or players IDs
- Any player including the dealer should be able to add cards them to their hands
- All players should be able to see the cards in their hands, and the sum of the rank of the cards in their hands
- Create a automated dealer that distribute the cards to the player hand
- Create players that can store their hand and the sum of their hand
- Create an interface that allows the players to view their cards, the dealer's cards, and the cards they are picking up during their turns
- Write programming logic that can calculate whether players have a sum of 21, or below 21, or above 21
- Based on cards in the player's hand the program should be able to determine whether the player won the round or lost the round
- Ability for the players to take their turn in a consecutive manner
- Ability for the player to be able to pick cards during their turn (hit me) or end their turn (stay)
- Ability for the automated dealer to take his turn and determine whether they have won or the remaining players have won

4. Implementation Plan

The Git Repository URL: https://github.com/Danny-Chaudhry/Deliverable_1.git

The Git repository will be used by all members to post their contribution of the project in a remote repository. This will allow all members to post their work and review other members' work. All members will post their work on a branch in the Git repository, and only after their work is approved by all members and confirmed by the team leader can they merge with the master branch.

The group will use the NetBeans IDE to complete the project. Although the use of NetBeans is not mandatory for our group, nonetheless it is preferred over alternatives IDEs because it is more convenient to use, and it provides the group with all the features necessary for the project.

Work done on UML diagrams or case diagrams will be kept within a separate folder called vpproject folder inside the main project folder. This folder will be pushed on to the Git remote repository, so that it is available to all members for editing or adding their own work.

Any extra files will be added to the Extra folder inside the main project folder. The structure and organization of the extra folder will depend on the members, and may change as the project develops.

All members will be required to check the repository at least every other day; this is to ensure that if any changes are made to the project they are aware of it. Lam and the team lead, Danish, will be required to check the repository status every day. This is because Lam's primary duties are testing and ensuring the code is properly documented, so he will daily view the code and ensure that code works as intended and the code is properly documented. Danish will view it every day because as the team leader he must be aware of how much work other members have completed and if they are meeting their set deadlines.

As stated earlier due to the complexity and vastness of this project, and the asynchronous schedules of the group members the Git repository will be heavily relied upon to organize the project so that deadlines are met on time.

5. Design Considerations

Currently, the code has some encapsulation within the classes. The data fields or attributes of each class are private. There are getter and setter methods written in each class to access those data fields. Certain blocks of code which represent specific functions in the Blackjack game such as the deck shuffle or create a deck of 52 cards, using 52 Card objects is encapsulated in a function. The exact algorithm of the function is not necessary for the programmer to know, in order to use the said function.

Currently, the only delegation being used in the code is when the collections class shuffle method is used on the array list cards. Although, at the present moment, the array list has not been initiated in the code, it has only been declared. In the future the scanner will be the delegate because the console will be used as the interface for reading the player input and writing the game results. The game class will also delegate its responsibilities onto the GroupOfCards class and the players, such responsibilities may include the player taking their turn, the GroupOfCards creating a deck, storing it, and shuffling it etc.

By creating a different class object for the player, the cards, deck, and the game itself, we are increasing the flexibility and the maintainability of the code. Through this, the changes made to one class do not adversely affect the functions and operations of another class. This also allows us to update each class without worrying about its effect on the whole game. By using different classes to create different objects we are maintaining high cohesion within the classes, and keeping our code loosely coupled among the classes. This will aid us in maintaining our program.