Team Contract

SYST 17796 TEAM PROJECT

Team Name: __BlackJackTeam10_____

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.

Team Member Names (Please Print)	Signatures	Student ID
Project Leader: Hoang Chuong Nguyen	HN	991651113
Sharif Kulmie	SK	991637947
Joshua Robinson	JR	991647546

For further information read Academic Honesty Policy on AccessSheridan.

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 . . .

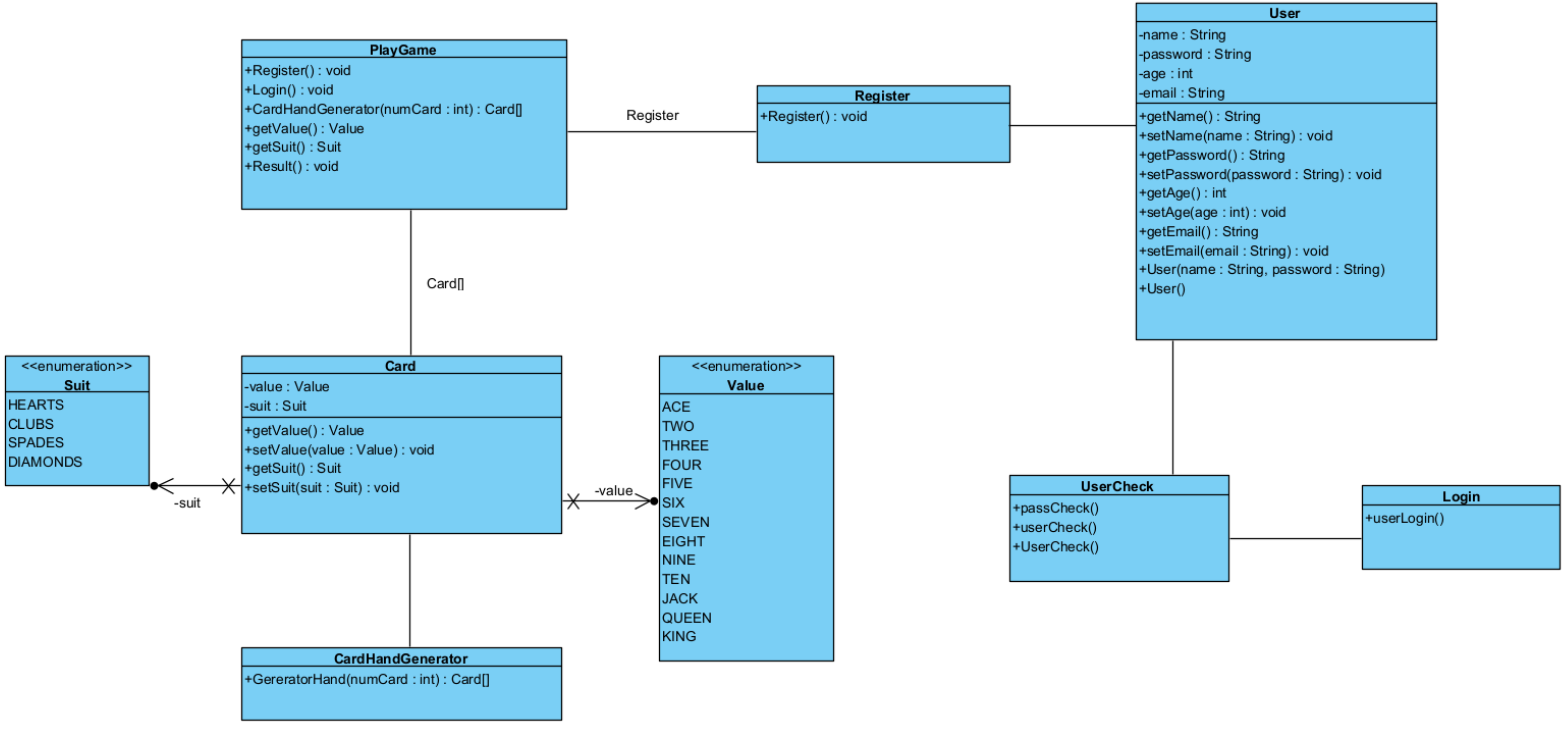
Scenario	**Accepted initials**	**We agree to do the following**
Team member does not deliver component on time due to severe illness or extreme personal problem		a) Team absorbs workload temporarily __ b) Team seeks advice from professor __ c) Team shifts target date if possible __ d) Other:
Team member cannot deliver component on time due to lack of ability		a) Team reassigns component __ b) Team helps member __ c) Team member must ask professor for reference material __ d) Other:

Scenario	Accepted initials	We agree to do the following
Team member does not deliver component on time due to lack of effort		a) Team absorbs workload __ b) Team "fires" team member by not permitting his/her name on submission __ c) Other:
Team member does not attend team meeting		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 __
An unforeseen constraint occurs after the deliverable has been allocated and scheduled (a surprise test or assignment)		a) Team meets and reschedules deliverable __ b) Team will cope with constraint __

Scenario	Accepted initials	We agree to do the following
		c) Other:
Team cannot achieve consensus leaving one member feeling "railroaded", "ignored", or "frustrated" with a decision which affects all parties		a) Team agrees to abide by majority vote __ 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:
Team member behaves in an unprofessional manner by being rude or uncooperative		a) Team attempts to resolve the issue by airing the problem at team meeting __ b) Team requests meeting with professor to problem-solve __

Scenario	Accepted initials	We agree to do the following
		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		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		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		a) Team forces decision sharing by routinely voting on all issues __

Scenario	Accepted initials	We agree to do the following
		b) Team routinely checks with each other about perceived roles __ c) Team discusses the matter at team meeting __


PlayGame
+Register() : void
+Login() : void
+CardHandGenerator(numCard : int) : Card[]
+getValue() : Value
+getSuit() : Suit
+Result() : void
Register
Register
+Register() : void
User
-name : String
-password : String
-age : int
-email : String
+getName() : String
+setName(name : String) : void
+getPassword() : String
+setPassword(password : String) : void
+getAge() : int
+setAge(age : int) : void
+getEmail() : String
+setEmail(email : String) : void
+User(name : String, password : String)
+User()
Card[]
<<enumeration>>
Suit
HEARTS
CLUBS
SPADES
DIAMONDS
-suit
Card
-value : Value
-suit : Suit
+getValue() : Value
+setValue(value : Value) : void
+getSuit() : Suit
+setSuit(suit : Suit) : void
-value
<<enumeration>>
Value
ACE
TWO
THREE
FOUR
FIVE
SIX
SEVEN
EIGHT
NINE
TEN
JACK
QUEEN
KING
CardHandGenerator
+GereratorHand(numCard : int) : Card[]
UserCheck
+passCheck()
+userCheck()
+UserCheck()
Login
+userLogin()

Hoang, Sharif, Joshua

SYST 17796 DELIVERABLE 1
DESIGN DOCUMENT TEMPLATE

GROUP 10

1. Project Background and Description

Project name: Blackjack

Gameplay rules:

Including 2 player (Host and user), for the first round, player and host will have 2 cards, First round, random 2 cards for both user/player and host. The Host will show the User 1 of his 2 cards.

There will be an option for the Player to continue drawing the next card or not. In the second round, if the host has a total of less than 15 points on the card, the host is required to draw a third card, and for the user, it is 16 points. In case the 3rd card is drawn, and the player's total score is more than 21 points, the player loses. At this point, there will be an option for the user to continue to draw cards or not if the total score is less than 21 points. If the user chooses not to draw any more cards, the host and the user will open the cards, if anyone has a total of less than 21, more than 16 and greater than the other person, they will win.

We have now designed and coded part of the rules and gameplay. We will add more registrations and bonus points for players.

2. High-Level Requirements

The game will include the following:

- Ability for each player to register with the game
- Ability for the game to communicate a win or loss
- Ability for players to know their status (score) at all times

3. Implementation Plan

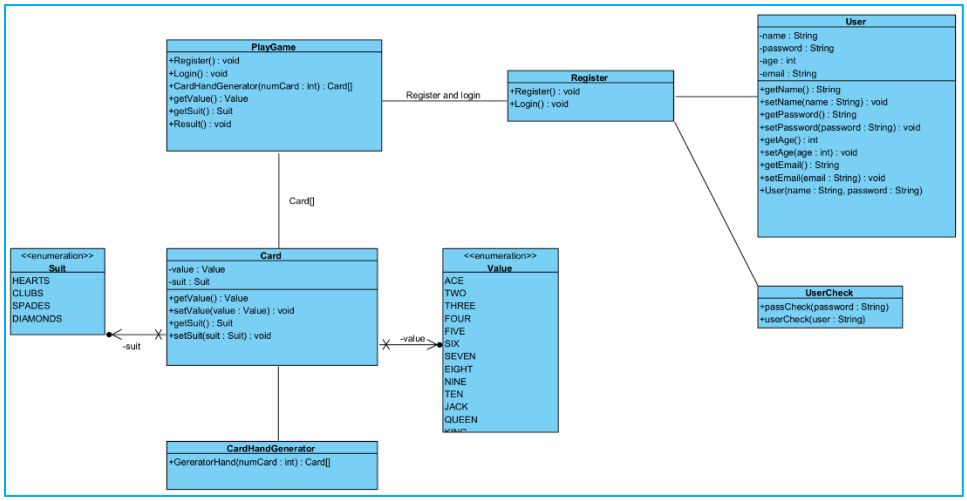
https://github.com/huy58501/BlackJackGroup10.git

We have updated our code every week since we started the project. Each student has committed at least once to the code and the code will continue to be updated as accordingly until finished, along with other extra interaction with the user. Our UML diagram contains all the classes, methods and variables used in the code, which was created using Visual Paradigm. We will be using NetBeans Apache 12 to complete our code. This report was produced using Microsoft Word. This report will be in a separate folder named ‘Documents’ in our repository and the UML diagram will be under the folder named ‘Diagram’. Our project uses Java 11 and will be completed using concepts learned from our previous semester as well as all that we have learned thus far in this course.

The code has been tested by all developers to ensure that everyone on the team is satisfied with the work that we have done and that the code operates properly with no syntax, compile, or runtime errors.

4. Design Considerations

We have used the *Single Responsibility Principle (SRP)* by writing our codes in High cohesion. This means that in our Project, every class has a single responsibility. This will make it easy for readability and finding maintainability.



For example, the Register class has one function: creating new users. Another example, Card class is just holding the card information, the Value which holds numbers of (1 – 13), and Suits which contain Card symbols (Heart, Diamond, Spade, and Club). Each Value number has a 4 Suit symbol making 52 cards.

We have also used the *Don't Repeat Yourself principle (DRY)*. So, there is no duplicated code in all classes. As we did in our project Card class delegated the task of creating cards to CardHandGenerator class. Furthermore, User class delegated the task of registering new users to the Registration class. And the Registration class delegated the behavior of checking the user validity and password to UserCheck class.