syst 17796 Deliverable 1

design document template

# Overview

## 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.

The game I chose to code was Go Fish, which is a card game that can be played with two players. The ideal project would allow for multiple users to play on the game, as well as users being able to see their own cards, and make a call on what card they want to see from their opponents. The starting base code I was given was very generalized, so I decided to remake the classes from abstract ones to concrete ones, without parent classes. The Player, and GroupOfCards class in the starting code used an ArrayList to map out their objects, but I felt more comfortable working with normal arrays of objects. There was also only one Game class in the starter code, and no main method, so in my project I divide the game into two separate classes, one of the game logic and rules, and another game that acts as a menu to initialize the game

I have included a set of rules in a txt file and took the rules from a youtube video:

<https://www.youtube.com/watch?v=-2z9xZYWqmo>

## Project Scope

Describe the names and roles of each team member. Describe the technical scope of the project by talking about the interface and how you will know when the project is complete.

The technical scope of the project isn’t too complex, all the logic needed to code the game can be done using the standard JDK library without the need of importing much except for the Random class library as well as the Scanner class. The interface from which the user’s will interact with the project will be text only, no GUI. When a user is able to input their name and successfully play one game, and have their score recorded, then the project will be complete.

## High-Level Requirements

The new system must include the following:

* Ability for 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
* Ability for players to know how many cards are left in the deck after each turn
* Players can continue to play game after win or loss and game will keep tally of total wins and losses

## Implementation Plan

Include your Git repository URL here and a brief description of the expected use (i.e. each developer checks in code at the end of each day/week). 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…)

The tools I expect to use are only NetBeans, for coding, and VP for creating the UML diagrams.

## 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).

* Delegation is the action of referring to a member of one object, in the context of another different object
  + Throughout the starter code there is some delegation of Class objects in other classes, with the usage of an ArrayList of Cards in the Player class, as well as an ArrayList of Players in the Game class.
  + Throughout my code there is much delegation, such as by Player object instances in the Game class, which in itself had Deck class objects, and used the Deck methods on their individual hands.
* Flexibility/Maintainability : High cohesion
  + All the code from the starter code had high cohesion, there was no overlap of methods acting on the same things in different classes.
  + In my own game code, there is some loose cohesion, especially in the Game class. Since it has methods which interact with objects from different classes to manipulate them, I was unsure of where to put some the methods in the class, such as with the player turn methods, since they were so long I was debating for a long time about placing something like them in the Player class instead, however I decided to keep the player methods in the Game class as that seems like the most related class to put them in.
* Maintainability
  + My Code isn’t the most maintainable as the Game class has to type out each additional Player that I would want to play with. I believe if I wanted to add additional players to the game, I would probably need to insert an array of players and add an argument to the playerTurn( ) method, of the Player whose turn it is. In addition I feel like I would need a lot of if conditions for each different player number situation.
  + If I were to further work on the code, I would probably need to change the regular array objects into ArrayList objects, to allow for more flexibility in the array sizes, especially for the deck objects in each Player.