Class Specification Document

Introduction: This document sets to expand on the classes in the class diagram for Group 2’s project for SWE 3313. Each class has its own purpose in the project and that will be specified in the sections below. Important things, like what the class is supposed to do and how it will perform that, are in this document. How each class interacts with each other through their relationships will also be expanded upon.

Classes:

* Card:
  + This class represents an individual card.
  + Each card is represented with a value like ace, queen, 10, 2, etc… and a suit like, Spade, Heart, Clubs, and Diamonds.
  + The class has to use a get method that will return the value and suit.
  + The constructor has to require both a suit and a value, so they can be set with those values when created in the board class.
* Board:
  + Creates array of cards with their own suit and value.
  + The board class will create and use objects Tableau, foundation, and card.
  + The board will represent the gameboard that has 13 Tableaus, 4 foundations, and 52 cards. That is how many objects will be created onto of the board object.
  + The Board class will have a relationship with the GameMaster class to allow the player to make moves and determine if the move is legal.
  + The Card Shuffler class will shuffle and distribute all cards on the board.
* Card Shuffler:
  + This might not need to be in a separate class, but instead could be built into the board class.
  + The Card Shuffler class will fill up each tableau object with 4 card objects on the board.
  + Card Shuffler will shuffle all cards randomly.
  + Moves kings to the back of each tableau deck they are in.
* BoardSlot:
  + The parent class to the tableau and foundation classes, with the basic function of being able to take in cards and store them on a stack.
* Tableau:
  + Has a max limit of 4 cards and is filled at the beginning of the game.
  + 13 of them will be created on the board.
* Foundation:
  + Can hold all cards of a specific suit.
  + Starts the game empty.
  + Game is won when all ace’s and kings are moved into foundation decks.
* GameMaster:
  + Handles player input.
  + Handles the legality of moves.
  + Determines if the player has won or lost the game.
  + To do this the GameMaster class must be able to check on the conditions on the board.
  + Handles GUI usability.