In the context of the Poker application described in assignment 1

Given the following strategy for AIP (which may force you to adjust your code from A1):

Strategy

- if AIP has a straight or better, it does not exchange any of its cards

- elseif AIP is one card away from a Royal Flush, a Straight Flush, *a Full House*, a Flush, or a Straight, it exchanges that card. Please note: one card away from a full house involves 2 cases:

    · you have 3 cards of the same rank and two other distinct ones, in which case you exchange the lowest-ranked of these two distinct cards

    · you have two pairs, in which case you exchange the card not part of these pairs

- elseif AIP has 3 cards of the same suit, it exchanges its two other cards

- elseif AIP has 3 cards in sequence (ie n, n+1, n+2), it exchanges the two other cards

- elseif AIP has 1 pair it exchanges its other 3 cards

- else AIP keeps its 2 highest-ranked cards and exchanges the 3 others

1) Develop a set of user stories that covers all requirements of A1 **relevant** to the user of your application.
2) Implement the testing of these user stories using Cucumber

Please note:

1) It is quite unlikely a correction grid will be issued for this assignment as we will not supply the list of user stories we envision.
2) You will be marked on:
    a. The completeness of your testplan wrt **customer-relevant** requirements (80%). That is, your user stories must take a black-box perspective of the system and focus on inputs and outputs.
    b. How elegantly you deal with equivalence partitioning (e.g., detecting a royal flush whatever the order of the cards are in a hand) (20%)