In this project, I learned a lot more about the overall syntax and data structures of Java. In my blackjack game, I implemented arraylists to store my hand as well as my deck of cards. Prior to learning about ArrayLists, the only option I had to store a set of data was through the use of an array. However, that presented some issues, mainly, that an array is fixed in size. So if I wanted to remove a card from the deck, then I would have to create another array containing all the cards except for the removed card. So then, I used an arraylist to create my deck because the arraylist can be increased or decreased to accommodate the number of objects it contains. Furthermore, in this project, I also implemented the Random Class. For my project, I needed to draw a random card from the deck, so then I had to use a random class to generate a random number to take that random card from the deck. Another concept that I used was a Scanner Class inside a method. Although I have used Scanner Classes before in my main to get user input, this was my first time actually using it inside another class’s method. In terms of time management, I definitely could have focused on that part better. As I had multiple commitments prior to this project, I ended up being a bit late on reaching the sprints. However, once I started to work on this project, I managed my workload, by splitting my work into different sections, based on the importance of the class/method. As for project management, I was working alone, so there was no need to communicate with others on the parts that I have done.

Some problems I faced included, my deck not taking exactly 4 10s, jacks, queens, and kings. My initial way of creating the cards with the value of 10 was to have a random number between 1 and 4, where the number was hashed to a particular 10 value card. However, the issue with that approach was that 4 cards of each was not guaranteed. How I fixed the issue was simply adding another attribute to my card, called name. Then all I had to do was loop through to add the cards to my deck. Another major issue I had was getting the total value of a hand. My first method was to add all the non ace cards first. Then I would add all the ace cards, and if the ace card value plus the current total was greater than 21, then I would lower the ace card to the lowest value possible(1). However, this method fails in some cases(etc A, A, 10). So then, my second method was to automatically treat ace cards as value 1 and add them to my total. Then, loop and add increments of 10 to the total until the difference between 21 and the total was less than 10, or the amount of ace cards in the hand was reached.

I believe overall, this project was very good for my understanding of OOP, as well as learning different data structures and classes, such as arraylists. I also learned more about managing my time overall, especially in regards to unexpected delays that I might have to look for, such as an unknown bug in my code.