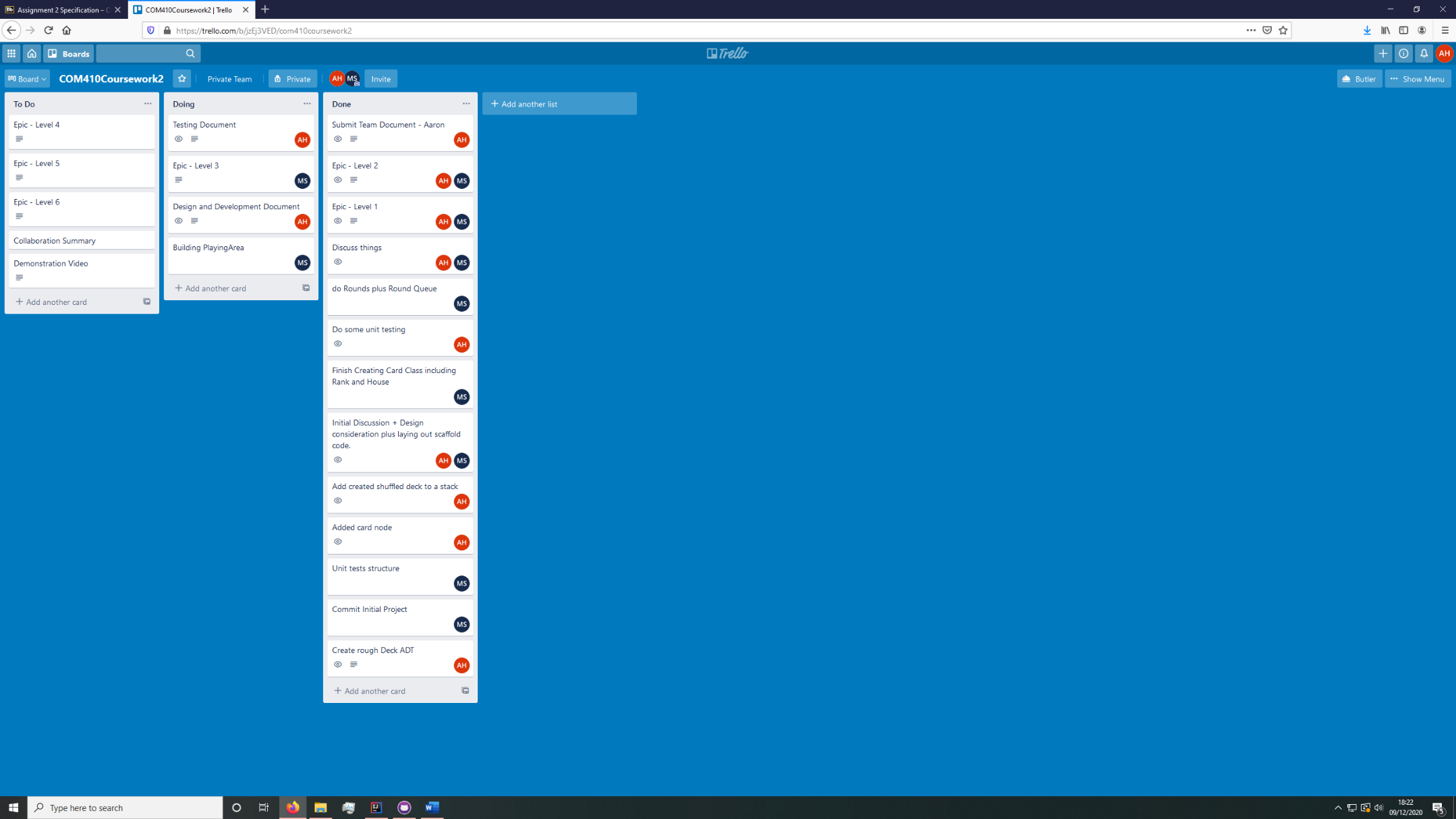
# Design and Development document

To start off the development project we met on zoom and talked around the specification of the coursework set to us and outlined the requirements needed to be developed in our java application. We did some research on the rules and how to play the elevens game in real life to gain a better understanding how the game is played in the real world. So now having some understanding of the rules and the requirements we went about doing some project management.

We created a Trello board, a to keep track of tasks and ensure the workload is shared out and to ensure we can keep an eye on the deadline and have the app developed on time and maintain organisation. We also created a github repository to store the code base to be accessible and push changed to in a central location, so both members of the team can work on the same code base and keep it up to date.

The requirement of the Elevens game was then discussed and broken down into a table below and set under a choice of titles to show their priority, required, desirable, nice to have and out of scope. This was to help prioritise which parts of the functionality should be developed first and in an order to be developed in.

|  |  |
| --- | --- |
| Required functionality by end user: | Required/ Desirable/ Nice to have/ Out of scope |
| Functionality for the player to have a new game present when they launch the elevens game. |  |
| The player should have a ready to play deck being dealt 9 cards from a shuffled deck. |  |
| The player should be able to choose pairs of cards to be removed from the game according to elevens rules, having two cards add up to 11. |  |
| When cards have been removed, the player should automatically have new cards dealt from the deck, to their playing hand, 9 cards before the next round begins. |  |
| The application is able to inform the player when the game has been won, when all cards have been removed from the game. |  |
| The application is able to inform the player when the game has been lost, when there are no more cards pairs which add to give 11, therefore the game cant progress. |  |
| The game is able to provide a hint to the player on the players request, giving a valid move or informing the player that no moves are possible. |  |
|  |  |

We then went to creating the project, scaffolding out some classes for the basic parts of functionality and thought of certain tasks that we need to get working on to fulfil the first couple of requirements. We went through each class and thought about which abstract data types we should use. We were going to use arrays for some of the classes, such as the deck class to hold the deck of cards. But we decided that I made most sense to use a stack ADT for the deck, since a deck of cards is lifted from top of the stack to the bottom in the Elevens card game in real life. It was decided that we would use a queue to add the round into that works along with the round class. We also used a bag class for the Card class to choose the cards. The deck class also works with the cardnode class based on the Node class from the lecture example, to get the position and data of the items in the stack. We then added those tasks to the trello board to keep track it. And got started on our initial tasks to get the app underway. We kept in contact through whatsapp to what work is needed and what we were working on and if we encountered any issues.