Design Document

The purpose of our group project was in ambition of creating a working and sufficient software which would be able to produce all the functionalities of the game Word Hunt in which we had created. This type of software is obviously in its early stages, and if proven valuable we would essentially translate this into a possible mobile application, or even try and modify it into a much larger scale game.

The entities that were inhibited in our design was a server client based threaded system, which had multiple threads being able to run per one server. This was all exemplified in java, where we had different socket openings, all while sending and receiving information via serializable data back and forth between server and client. Overall, this process was quite difficult as many documentation was needed to try and implement this four player multiplayer game. The game itself is based on a variable of word shuffling, where each client receives a jumbled up word, and is given the task of trying to correctly map out which standard english word maps out to that string of unordered letters.

Some low level methods that we had to execute was the usage of a hash map. We did lots of research on trying to find an efficient model in sufficiently holding our words in a proper yet quick data structure. A properly sized BST tree and graphs were highly lucrative, but we simply decided to follow up on the hashmap as it would quickly be able to map out words at almost a O(1) run time in certain scenarios. The hash map essentially helps our program efficiently work on the backend, as we randomize all the values and quickly pull out a word for the client to try and figure out. This approach would be very practical in the industry, as it

enables us to quickly and efficiently create data structures to provide the backbone of our game and code.

We believe that the game we created was very practical yet has the ability to become quite noteworthy in the future. It is very risk free to any suitors, as the game could be translated into other types of word games, all with the same underlying methods and algorithms. Our GUI on the other hand could be implemented on more sophisticated platforms, as right now it's simply buttons and grid panes. However, we would be willing to take the next step in trying to have the game be approachable in a possible graphical interface with possible graphics for the letters shown. Furthermore, right now we are simply running threads in java, however we believe in the future with Node JS, and implementing the MEAN model of Mongo/HTML/Angular/NodeJs we would be more efficiently be able to come up with a game that would be best suited as those working in the software industry right now.