**Team: 3-8**

We used a SCRUM type of team organization to track and organize our tasks for each sprint and we updated our task information to determine how much time has been spent on each task and who was in charge of each task. We used [trello.com](https://www.trello.com) in order to manage our SCRUM on a global level so that all members can keep track of the tasks no matter where we were. We chose the SCRUM leader based on who wanted to, and because no one wanted to, Braden Bird took the role of the leader and maintained the organization of our group.

Initially, we determined the tasks based on our comfortability with Java and we gave the easier tasks to those who were less skilled in programming and developing in Java. Other than judging the tasks based on skill, we also allowed for some developers to choose what they wanted to complete such that they could obtain knowledge in a specific area. We changed the way we divided up the work later due to increased comfortability with tasks and learning more about each person’s strengths and weaknesses. We also assisted each other on tasks when current assigned tasks for that sprint for said developer were already completed. Now that we know what particular strengths each person has and the amount of time to be expected on given tasks, we would reorganize the tasks to be in the most efficient manner such that we could complete the sprint early and potentially increase our efficiency of our program. We would also want to try a different way to manage our backlog because the website that we used did not provide much flexibility when it comes to recording the given data for time spent. This caused our graphs to look very sporadic when we adjusted values at SCRUM meetings.

The first major problem of which we encountered in the project would be getting the GUI and the backend program to be linked correctly and running so that the values changed on the GUI would represent the values that were changed in the backend. Another major problem of which we had encountered was implementing the client/server model. We experienced many issues with the client/server model because no one on our given project team had prior experience with network socketing and relaying messages between multiple clients through a server. We had to approach this issue using given information and a ton of debugging. This caused us to devote more time to the client/server model than the allotted time given which could have been used helping improve other parts of our program.

|  |  |
| --- | --- |
| Team Member: | Grade Ratio: |
| Matthew Stevens | 1.0 |
| Victor Carlon | 1.0 |
| Braden Bird | 1.0 |
| Huai Wu | 1.0 |