**C Sc 335 Analysis and Design Artifacts for Final Project**

*This must be added to your private Github repo in a directory named documents*

**1. Team Name:**  \_Cindiesel\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2. Team Members**: \_\_\_Sahil\_Dalal\_\_\_\_\_\_\_\_\_\_\_\_\_ \_Patrick Maley\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_Eric Wagner\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_Ana Huff\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. Candidate Objects or Class Hierarchies**

List the most important objects, or the name of an inheritance hierarchy, and the main responsibility.

|  |  |
| --- | --- |
| **Candidate Object** | **Single Responsibility in 1 or 2 sentences** |
| Player | Holds players username, password, dob, items they are holding, and most recent gameplay. |
| Items | This will be an abstract class with the methods that are common for all which will have subclasses (different tools) that extend it. |
| Rooms | This will be an abstract class with the methods that are common for all rooms which will have subclasses (different rooms) that extend it. |
| MOB | This will be an abstract class with the methods that are common for all mobs which will have subclasses (different mobs) that extend it. |
| Views | This will contain the multiple text views. The text view they type into, a text view showing the items they have, and the view showing their position on the map. |
| Server | Allows multiple user to play and interact with each other on the map. |
| Client | Allows multiple user to play and interact with each other on the map. |
| Moves (ENUM) | Has the movement of the players. |
| Account Holder | Holds all of the different players passwords and usernames so they can login |
| Map | Holds all the rooms |

**4. Class Diagram:** Your team UML Class Diagram must show at least all of your candidate objects from above. Show any relationships between them the classes such as inheritance or interface implementation. Draw general associations such as dependency or aggregation. Label some to help explain things. Add any multiplicity adornments that seem appropriate. Use notes to explain things if you feel it will help. Each UML class must show the class name. For full credit, each class must have an average of at least one attribute per class. There must be an average of at least 2.0 methods per class, which may be implicit (no need to repeat methods) if the class implements a Java interface with methods shown there.

*This class diagram may be written by hand and scanned or drawn with any*

*UML editor or drawing program*

**5. Sequence Diagram:** Your team UML Sequence Diagram should show the most important scenario you can think of. Your sequence diagram should show most of your objects from above and how they communicate with each other.

*This Sequence Diagram may be written by hand and scanned or drawn with any program or sequence diagram editor such as* [https://www.websequencediagrams.com/#](https://www.websequencediagrams.com/)