**C Sc 335 Analysis and Design Artifacts for the Final Project, Spring 2014**

*Due: 11:00pm Friday 11-April to the common repository in BitBucket as part of your Java project*

**1. Team Name:**  \_\_Rick, We Shrunk Ourselves\_

2. **Team Members**: \_\_\_\_Jenna Franco\_\_\_\_\_\_\_\_\_\_\_ \_\_\_Richard Bosse\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_Anh Tran\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_Devin Kempton\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. Project Chosen:** Circle your team's choice of final project

TRPG MUD Tower Defense Settlement Management

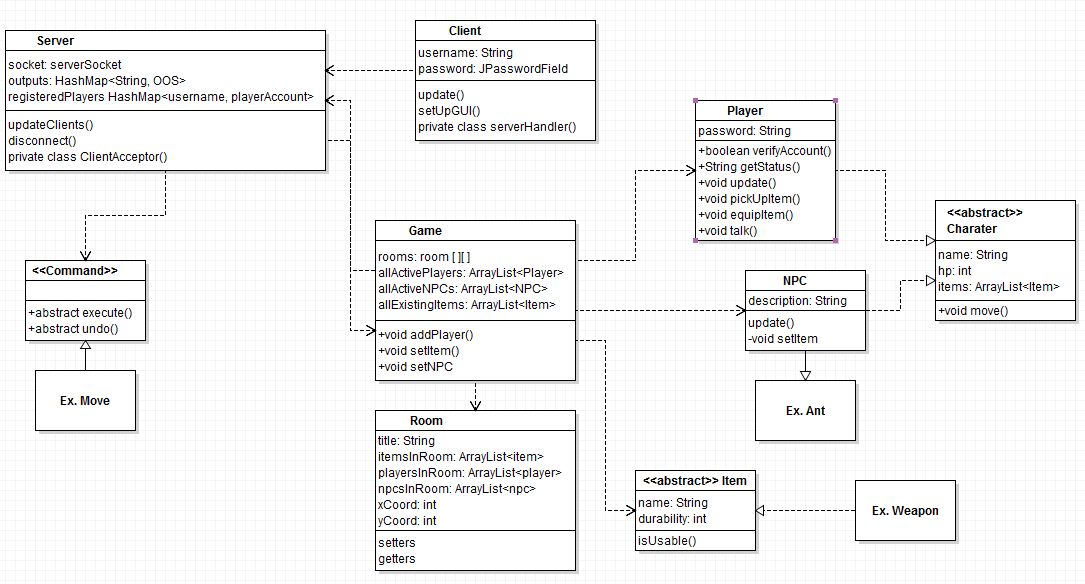
**4. Candidate Objects or Class Hierarchies**

List the seven most important objects, or the name of a hierarchy, and the main responsibility of each

|  |  |
| --- | --- |
| **Candidate Object** | **Single Responsibility in 1 or 2 sentences** |
| 1 Server | Creates server, accepts and managers client connections, start up the game, maintains game information and updates regularly |
| 2 Client | Opens connection on server, allows real time chat, allows player to view and interact with the game, is the GUI framework for the player (JFrame)  GUI:  Instead of panels, use composite tabs  Need at least two sections for text updates (social chat, all game text)  Possibly also section for list of active users, player stats (only yours), inventory list |
| 3 Player | Associate unique username/password combination, store specific stats, pick up items, equip items, talk to other players, move |
| 4 Command | Abstract class for commands in the game (ex. move, look, pick up item, attack, talk) |
| 5 RWSO Game | Creates the map (list of rooms), holds list of NPC, creates new game, populates where items are, allows player to use commands |
| 6 Character | Abstract class for characters in game (ex. player, monsters, helpful characters):  Stores name, hp, and items |
| 7 Item | Abstract class for different items (weapons, food, special items) |
| 8 NPC | Defines non player character |
| 9 Room | Defines a room in the game, will store information such as:  Exits, items contained, players inside, coordinates on 2D array |
|  |  |

*These Class and Sequence Diagrams may be written by hand, captured as a picture on a white board, or drawn with a UML editor such as Violet* [*http://sourceforge.net/projects/violet/files/violetumleditor/*](http://sourceforge.net/projects/violet/files/violetumleditor/)

**5. 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.



**6. 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.

