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

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

*with your project manager added as a collaborator*

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

**2. Team Members**: \_\_\_Kite Christianson\_\_\_ \_\_\_Jared Worthington\_\_\_

\_\_\_\_Chris Yingst\_\_\_\_ \_\_\_Steven Murrell\_\_\_

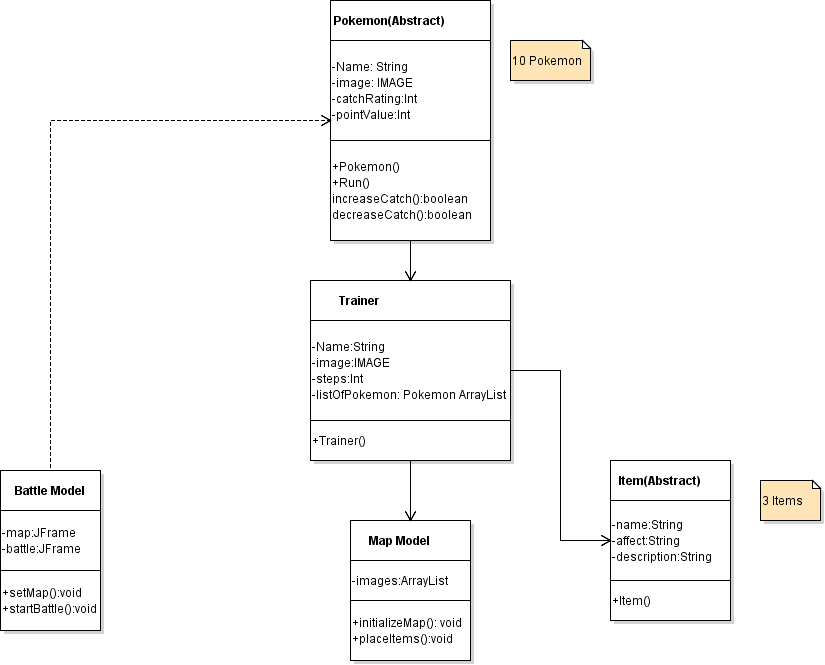
**3. Candidate Objects or Class Hierarchies**

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** |
| Pokemon (Abstract) | Represents all 10 Pokemon, with differing rarity, sprite, and catchability, but with similar actions such as running away, being catched, and animations. |
| Trainer | Represents the trainer responsible for maintaining things such as the number of steps, the number of pokeballs, pokemon caught so far, score, etc. |
| Map Model | A grid of spaces where a trainer can move. Spaces of the map model will contain the trainer, randomly placed pokemon and items. |
| Battle Model | Represents the actions and sequencing for battles between trainer and a pokemon. |
| Item (Abstract) | Represents three items that have differing effects and values, but with similar actions such as consuming. Trainer possesses an inventory represented by a list of these. |
| PokemonGUI | contains the Jframe and Jpanel used to show the game. Also implements keyboard listeners for movement and option selections |
| MapView | Draws images from the MapModel |
| Battle View | Draws Images from the map Model |
|  |  |
|  |  |

*These Class and Sequence Diagrams may be written by hand and scanned or drawn with a UML editor such as Violet* [*http://sourceforge.net/projects/violet/files/violetumleditor/*](http://sourceforge.net/projects/violet/files/violetumleditor/) *and / or the sequence diagram editor or* [*https://www.websequencediagrams.com/#*](https://www.websequencediagrams.com/)

**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.



**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.

