**C Sc 335 Analysis and Design Artifacts for Jukebox**

*Each team complete this form, put it in your project in a folder named* **doc** *and push to Github. This will be part of your Iteration 1 grade*

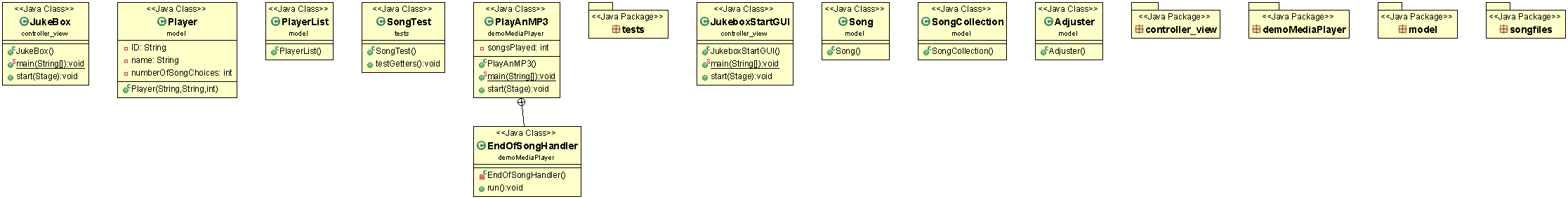
**1) Team Members**: 1. Andrew Lane 2. Suresh Krishna Devendran

**2) Candidate Objects**

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

|  |  |
| --- | --- |
| **Candidate Object** | **Responsibility in 1 or 2 sentences** |
| User | Have ID, Name, Count of Available song choices(3) |
| Juke Box | Plays song, is GUI |
| Song Collection | Database of Songs, verify choices exist in the collection, send message to Juke Box if exits, message to decide to play/ adjust if not |
| Decider/ Adjuster | Communicates with player, tells player to decrement song choice count. If identity verified; sends message to song collection |
| Song | Contains Artist, Tittle, audio file. Song Collection is comprised of songs |
| Player List | Has the list of valid accounts, and also holds the administrator |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**3) Class Diagram:** Write a UML Class Diagram that shows 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 about 1.5 methods per class.

**

**4) Sequence Diagram:** Write a UML Sequence Diagram should show the most important scenario you can think of. Your sequence diagram should show most of your candidate objects you listed above and how they communicate with each other.

