1. **Team Name:** Team Dank
2. **Team Leader for this deliverable:** Jonah Tollefson
3. **Team Members:**

Steven Hartnett,

Kyle Hoffhein,

Kyle Kwasniewski,

Jonah Tollefson

1. **Meetings:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Time-date** | **Attendees** | **Agenda** | **Action Items (who will do what)** |
| 3/28 – 8–10pm | Everyone | Work on the OO Design document | Jonah: Did the Functional Requirements  Kyle H: Did the Non-Functional Requirements  Steven: Did the schedule in Microsoft Project  Kyle K: Did the Class Diagram |
| 3/30 – 5-5:30pm | Everyone (Kyle K and Steven came a little late | Finish and correct the OO Design document and make the team report | Jonah: Wrote the team report  Kyle H: Helped edit the class diagram  Kyle K: Helped edit the class diagram  Steven: Finished the Microsoft Project Schedule |
| 4/6 – 5-5:30 | Everyone | Corrected our class diagram, update the report, and corrected the date in our plan | Jonah: Updated the report  Kyle H: Helped correct the class diagram  Kyle K: Helped correct the class diagram  Steven: Helped correct the class diagram and corrected the date in the MS Plan |

1. **Weekly Time Logs:**

|  |  |  |
| --- | --- | --- |
| **Person** | **Total Time in minutes** | **Tasks** |
| Jonah | 210 | Functional Requirements and Team Report |
| Kyle K | 190 | Created the Class Diagram |
| Kyle H | 220 | Non-Functional Requirements and helped with class diagram |
| Steven | 190 | Created the Microsoft Project schedule |
| **Total Time:** | 810 |  |

1. **Issues:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Issue Number** | **Discovery Date** | **Resolution Date ( Est. – Act. )** | **Responsible Person** | **Description ( Prob / Resolution )** |
| 01 | 3/28 | 3/29 |  | We found that the code given to us was not exactly like the playable flash game, specifically in our code the AI is not included / Talked to the Professor to see if we should port directly from the given code or to make it identical to the playable flash game. We were told that we can just port it from the given code. |

1. **Files and their locations:**

|  |  |  |
| --- | --- | --- |
| **Filename** | **Location** | **Contents** |
| CS3860\_TeamDank\_OODesign | changl/se3860\_cs5860/TeamDank\_/2\_ReengineeringProject/OODesign | Design document |
| CS3860\_TeamDank\_OODesign\_Report | changl/se3860\_cs5860/TeamDank\_/2\_ReengineeringProject/OODesign | Team report |
| CS3860\_TeamDank\_ReengineeringPlan | changl/se3860\_cs5860/TeamDank\_/2\_ReengineeringProject/OODesign | Schedule and plan |

1. **Plans for Coming Week:**

Next week we are planning on designing test cases.

1. **Comments:**

**Engineer 1:** *Jonah Tollefson*

For this deliverable I wrote all the Functional Requirements. I learned how to interpret requirements from one non object-oriented coding language and transfer them over into an object-oriented language.

**Engineer 2:**  *Kyle Kwasniewski*

For this deliverable I relearned how to create a class diagram in IBM Rhapsody and how helpful a structured analysis can be in creating a new OO design. It is especially helpful when you want to re-use some of the methods and functions in the original version before migration.

**Engineer 3:** Kyle Hoffhein

For this deliverable, I worked on defining the non-functional requirements for the system and supported in the construction of the UML Class Diagram and identifying Entity, Control, and Boundary classes. Through closer observation of the program, I learned how the source code differed from the code demonstrated. I was pretty comfortable with my understanding of UML Class Diagrams, but I did have some fun reviewing my knowledge with my partners, and working together to construct the diagram through compromise and understanding.

**Engineer 4:**  *Steven Hartnett*

I also did the majority of the plan for the maintenance plan and I found that planning for reengineering was much harder. For maintenance there was just a bunch of odd jobs that didn’t have much dependency on each other. On the other hand the reengineering plan required building an entire system from the ground up with lots of dependencies.