**Landen’s README**

Project Codename Olympia has proven to take much consideration and careful thought throughout the past several weeks. Starting out was a process in itself, because we had to create this system with primitive instructions and a vast imagination. Once the project started to be deciphered, the challenge did not cease. The process has required unmeasurable attentiveness and punctiliousness by every team member. Notwithstanding the aforementioned, the structural integrity of the system has thus far proven robust.

Communication between all group members has been superlative, and we have met twice a week to discuss future prospects of the system as well as current tweaks, concerns, and opinions that needed addressed at that particular point in time. If anything needed tackled outside of our meetings, conferring via text message and/or Github was utilized efficiently. When a quandary arose, and there have been a prodigious amount, we all collaborated and opined what would be the most advantageous route to take. Decisions in regards to differentiating between a subsystem, a class, and an attribute as well as which classes depended on, where associated with, inherited from other classes took much deliberation. The consequences that arose from these contemplations were a structurally sound foundation to lead us into the implementation phase with relative tranquility. From the first implementation of the Class Diagram, minimal alterations were made to achieve what we think is the quintessential paradigm for this system.

The workload was distributed as uniform as possible. The distribution occurred during the SCRUM meetings, via text, during our scheduled meeting times and, oftentimes, the workload was done in tandem by everyone in the group. The system has been ever-evolving with many successes. These aftereffects were only made possible with a tightly cohesive group behind the design of the system. To see who produced each portion of the system thus far, please consult the 3-ring binder in class as well as the Doxygen documentation for the construction of the classes. No qualms should be had pertaining to the completion of this system. Here at JALGroup, we value you as a client and we will be your number one choice for the Winter Olympics Automated Skating System (Project Codename Olympia).

**Andrew’s README**

My experience with the project has been interesting. Learning UML and defining classes proved to be a little difficult. The communication has been very good between all members. Me and Josh did the data dictionary. We all did the class specifications in a collaborative effort. I allocated athletes to each event and assigned judges to each event use case. I also did the IceDance, Pairskating, singleskating, scoring classes. I created the schema and database and I am currently starting the gui.

**Josh’s README**

Working on this project has been both challenging and very interesting to say the least. I’m very thankful to have a group leader that is so organized and I think because of that, our group works and communicates very well together. All of us in the group contributed to the class specifications during our time meeting together. I did the classes for the Speed Skating(500m,1500m,1000m) events, and did some of the used cases for the current standings, scores and display information. I also completed the data dictionary with Andrew. Currently, I’m working on entering the athlete information (first name, last name, gender, country) into the database.

**Final Report Walkthrough**

**Folder 1:** PCO - Contains original template (shell) of classes

**Folder 2:** ClassDiagramV6Uses.png - Class Diagram that describes which classes <<uses>> other classes. This includes the Database Subsystem.

**Folder 3:** UseCases2.docx - Descriptive list of all use cases with their optimistic and pragmatic flows.

UseCaseDiagramV4.png - Visual representation of use cases

**Folder 4:** ClassDiagramV5Functions.png - Class Diagram that has class functions (behavior) added to each class.

**Folder 5:** ClassDiagramV7Attributes.png - Class Diagram that has class attributes added to each class.

**Folder 6:** PCO - Updated C# program that includes all current member functions as well at all current attributes.

**Folder 7:** README.docx - Explanation of progress and distribution of work and directory of deliverables.

**Folder 8:** Doxygen - Contains auto-generated documentation for the Project Codename Olympia system in HTML format.

**Folder Progression\_NonDeliverable:** Contains visual progression of the diagrams, updated Class Specifications, Data Dictionary, and JALGroup documentation. These are also in the 3-ring binder for reference.