**Software Engineering 300: Iteration 1 Report**

Colin A, Jessie C, Bryan H, Brandon L, Shaina R.

March 3, 2019

**Product backlog: Refer to attached Excel Sheet**

### Summary of Meetings

**Sprint Planning Meeting**:

Conducted Online - All Members

Analyzed the project requirements and built class diagrams to analyze flow that the software would talk to accomplish requirements. Tasks were distributed based on the initial project proposal, and individual roles were assigned. Things such as environment were also mentioned, and the git had been made. Some of the requirements we thought would be required by the first iteration were also discussed, and software design and class diagrams were built.

*Deliverables:*

1. Sprint goal:

* Develop basic functionality for software that contain most requirements, create layouts and propose prototype designs for software interface.

2. Sprint backlog: **Refer to Excel Sheet**

3. Skeleton Class Diagram: **Appendix 1**

**Daily scrum meetings**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **What was done since last meeting** | **What will be done today** | **Any obstacles** |
| Feb 21 | **N/A** | Information exchanged, Initial Scrum meeting - online. All members attended. Initial get-together and planning of future project. Things such as roles and requirements were discussed | Planning of all the required classes proved to be difficult |
| Feb 22 | Class diagram partially built | All members reviewed, exchanged contents online - Refined the class diagram, added some more functional requirements to ensure all classes work. Created some skeleton classes to begin, work on the software. Began work on the software | Finding all the requirements and translating them into software functions. |
| Feb 25 | Skeleton of the software has been completed | Reviewed after class 1pm - Rough design of login and registration software completed. Design of the user interface under way, base code for the scholarship design beginning to be designed. | Checking how classes interact with each other proving to be somewhat challenging as we do not know if we have all the classes required |
| Feb 26 | Skeleton done, team members begun working on their individual classes | Refining of registration system, adding ability to store user data into database, Base model for the user interface completed, linking of systems to the user interface | Working around differences in code between members - fixing communication between classes |
| Feb 27 | Mostly finished fixing issues that came with merging classes | Linked the user interface to the login and registration system. | Setting outputs so that the user interface will be able to use existing methods |
| Feb 28 | Finished merging the login and registration systems with the user interface | Finished adjusting the login and registration user interface. Begun working on the inner panels for user interface. Scholarship standalone working | Swapping scenes and linking the scholarship system to the gui |
| Mar 1 | Attempting to list the contents of the scholarship database | Linking the user interface with the scholarship system. | Java swing methods to print are not working with our arrayList |

**Sprint review meeting**

Conducted in person after class - all members

Initial sprint for iteration 1 went well, we have managed to accomplish most user cases that we wanted to complete. Most of the code required for the current systems are finished, each system has the proper outputs to merge each class into the user interface. Time was taken to ensure each system works as anticipated, some rough testing done to the user interface to make sure no potential errors have occurred.

Some more work could be done to smoothen out the code, however class requirements that have been specified for iteration 1 have been mostly completed.

**Retrospective meeting**

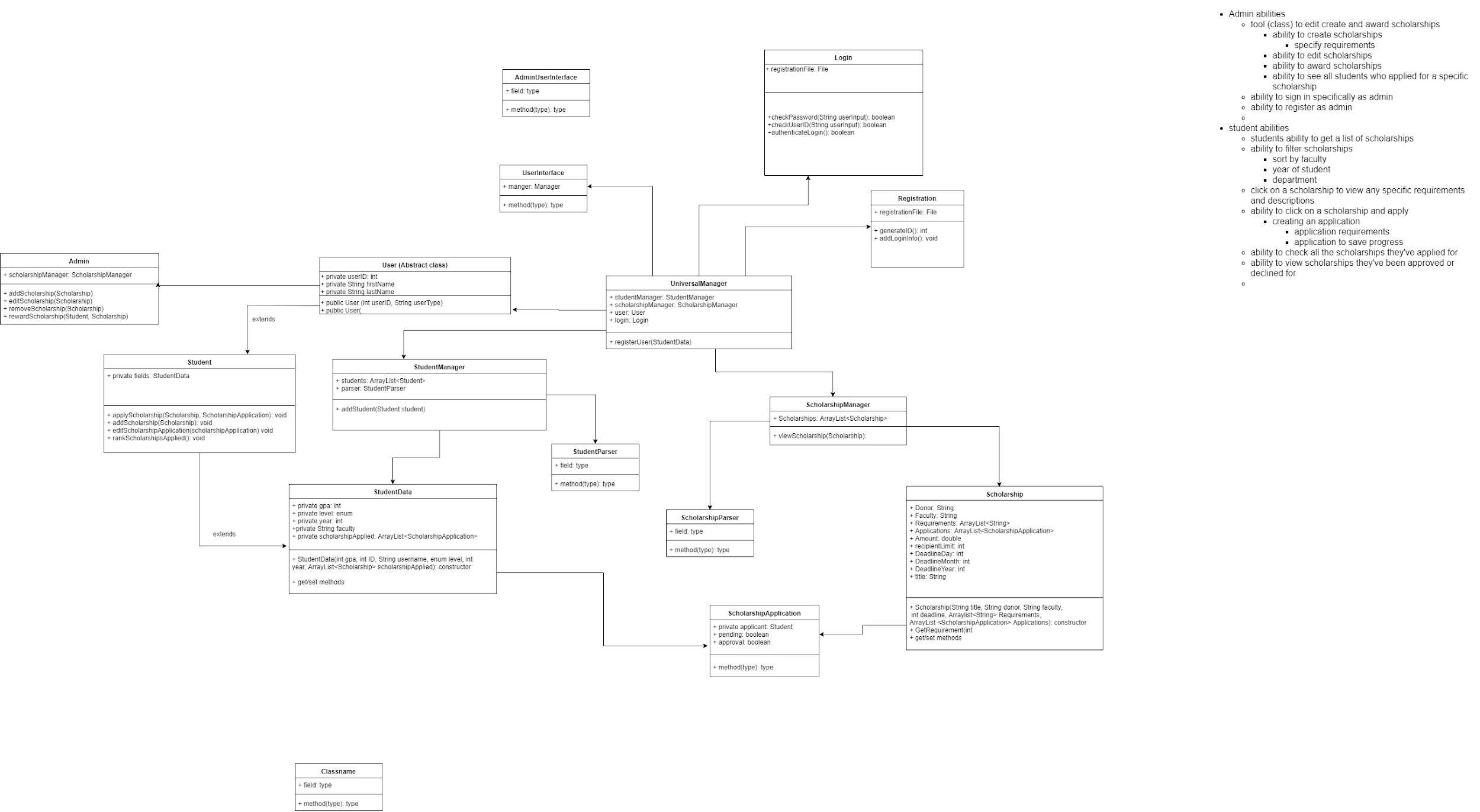
Conducted online - all members

Documentation could use some work, comments and perhaps short form javadoc would suffice until proper documentation can be completed. Perhaps better communication of how far each class has been done could be worked on.

Some improvements that could be done would be to define inputs and outputs required ahead of time, and to analyze each class so that no unsuspecting requirements may arise while working on different dependent classes.

Decision to work based on requirements helped to make system mostly work out of the box when combined together.

**Appendix 1: Skeleton Class Diagram**

****