Open Classes Project: Final Report

During the course of this spring semester, the Tech Ninjas team worked together to create the Open Classes application as part of the requirements for Software Engineering. This document will quickly outline the team’s experience working together on this project. First a brief description of the underlying problem will be provided. Then the process model and team structure chosen will be explained. A quick explanation of the system and interface design will be provided. Lastly, we will relate what we learned from this experience, what we would do if we had six more weeks to work on this project, and what we tell ourselves at the beginning of the semester if we could travel back in time.

The current system for looking up classes is slow and painstakingly tedious. It requires the user to drill down through a large amount of excess data. Plus, the current system resets all search parameters for each class search. Our customer wanted a system that will be easy to use and graphical in nature. They envision a system that accomplishes the same tasks as the current system by utilizing radio buttons, a pull-down menu, and other GUI characteristics. This application has been designed to make searching for courses in both computer science degree plans simple and uncomplicated. Instead of returning only a single course like the current system, our Open Classes application will display all available courses under a specified category within the undergraduate computer science degree plans. In addition to this convenience, our application will also the user to return to the selection screen without the loss of prior selections.

Most of the requirements specified in the requirements document remain unchanged, but adaptations have been made to a few of the requirements. Initially our team searched for a PDF to .txt converter, but an acceptable free option was not to be found. Instead, a program called PDF Tables was used to convert the PDF to a Word document, which was in turn converted into a .txt file. The other change to the requirements involved the database. The team chose to use Oracle 11g and Apache Tomcat in addition to MySQL Workbench.

The process model chosen by the team members was the Scrum process model. This process model was chosen due to its cyclical and iterative nature. The use of this model made it easy for team members to stay on track with project progress. Scrum is more adaptable to change than other process models, which the team found useful in that meetings were not traditional face to face meetings, but rather communication over communication applications such as WhatsApp. The team organization was democratic with every final decision made by consensus. Work was divided by task based on ability and reviews of an individual team member’s work was performed by all team members. Any member could request changes to the work of another team member.

The design of the Open Classes application involves the user interacting with software via drop-down menus, selector buttons, and radio buttons. The user must manually convert the class schedule PDF with software available on the project DVD. The system was programmed in Javascript and HTML on the front end, and Oracle 11g, Apache Tomcat, MySQL Workbench and Java were used on the back end. The application includes two types of databases: one which holds the degree plan lists of required courses and one which holds each semester’s class schedule. The application uses the degree plan database to match open classes from the schedule.

During testing, the team came across a few major issues with the application. The first issue discovered was that the program crashed when no selections were made. An error message was generated prompting the user to select a semester at the very least. Another issue found during testing was that when the Fall semester along with all subjects was selected, only a few subject categories were returned. This issue was also fixed during testing and the correct class categories are now returned for this search.

During the course of this semester, the Tech Ninjas have learned a great deal about the software development life cycle (SDLC). We learned of the importance of documentation, especially when collaborating in an environment where face to face meetings are not always possible. We learned that coding itself is truly only a small portion of the SDLC – system design and paperwork are far more time consuming! We also learned that there are always more test cases than you realize.

If we had six more weeks to work on this project, there are a few improvements that we would attempt to make to our application. First we would work to improve our documentation. Secondly, we would like to attempt to add another degree plan, such as mathematics. We would perform more extensive testing. Probably the most useful to the user, we would attempt to automate the installation process for the helper programs used to convert the PDF file to a .txt file. We would also attempt to find an easier way to keep the classes database updated as the registrar is constantly making changes to the classes file available on the website.

If we could travel back to the beginning of the semester, there are a few details we would tell ourselves. First, we would relate that any progress is better than no progress – work little by little to get work done. Steven would warn himself not to overwork, overstress, as well as warn himself not to get sick – if at all possible! We would also tell our newly formed team to make time for face to face meetings. We would suggest to ourselves that we should set aside an hour at a specified time each week so that there would always be a time we could meet face to face.

Overall, we believe the Tech Ninjas worked well together in order to accomplish the tasks set out before us. Each team member was self-motivated and worked hard to complete the tasks given to them. All in all, we believe this was a successful team effort and one of the best groups of which any of has ever been a part.