**Project Plan**

Christen Ford

EEC 521: Software Engineering

Introduction

Course registration is a ubiquitous software that can be found at many institutions of learning. As one of the primary pieces of software utilized by many campuses worldwide, course registration software has a well-defined set of features. Indeed, this software is so plentiful, that many institutions lease it from a software vendor rather than build their own. However, there are institutions that choose to build their own course registration software. Without proper planning, they may miss critical features or introduce flaws in the software that introduce abnormal functionality into the system.

Problem

Such is the case with Cleveland State Universities course registration system. This software resides in a package of software collectively referred to as “Campusnet”. Course registration is just one of the features offered by Campusnet. Major features of Campusnet also include: allowing prospective students to apply for admission and check their admission status; allowing current students to view their transcripts and degree audits; allowing students to sign up for campus alerts; and allowing current students to apply for and accept financial aid. Campusnet handles these features well enough. However, there are major flaws with the course registration system in Campusnet.

As a first-year doctoral student at Cleveland State, never having attended here before, I expected there to be issues with registering for courses. However, I also expected all my prior work in my undergraduate degree to correctly register with the Campusnet course registration system. This does not seem to be the case, as I am currently being repeatedly told that I do not meet the prerequisites for most courses I register for, even introductory courses. At first, I assumed that it was because I did not complete my undergraduate degree here. After talking with a few fellow students who did do so, it seems the issue is present at the undergraduate level too. This type of behavior from a piece of software generated by a university the size of Cleveland State is unacceptable. The whole point of this type of software is to ease the burden on administrative staff at the university, instead that burden was just shifted to advisory staff instead.

Solution

In the case of Cleveland State University’s course registration software, there is no need to replace the software, just to fix the existing software. However, given that it has existed in this state for at least four years, no one seems willing or wanting to do so. I would be willing to do so myself, however without access to Campusnet, I am unable to do so. In this regard, I propose the construction of a proof-of-concept showcasing working course registration software that includes all the courses offered at Cleveland state from the entire undergraduate and graduate course catalogs.

Target Customers

* Students – Not necessarily a customer per se as they would be using the software
* Advisors – Held in a similar regard as students, advisors would utilize this software to primarily view classes the student is enrolled in while additionally providing course overrides for the student during the registration process.
* Academic Institutions – Bona-fide customer of this software. If they already have a fully functional, working course registration system then there is no need for this software. However, if they are in a similar situation to Cleveland State, then this software will provide immense value to them.

Product Features

The product will implement the following features:

* Student course registration – Students can register for courses using the product.
* Student course unenrollment – Students can unenroll from courses using the product.
* Student schedule viewing – Students can view their current schedule using the product.
* Advisor course override – Advisors can provide course overrides for students using the product.
* Administrative course creation – Administrators can create new courses offered using the product.
* Administrative course removal – Administrators can remove courses offered using the product.
* Administrative course modification – Administrators can modify current courses offered using the product.
* Administrative course archival – Administrators can archive and unarchive courses.
* Administrative account creation – Administrators can create student and advisor accounts.
* View Courses – Students, advisors, and administrators can view courses offered.

Potential Users

Potential users include anyone who may use the product:

* Students – For course management
* Advisors – For advisory duties
* Administrators – For course administration

Anticipated Challenges

Given that course registration software is ubiquitous, there are a common set of challenges faced by the project.

* Integration with existing campus management software.
* Access to the current campus management software.
* Ensuring the software is up-to-date with the current course catalog.

Plan

This software will follow the waterfall methodology adjusted to the time table allotted for the project as follows:

|  |  |  |
| --- | --- | --- |
| Date | Goal | Objective |
| September 26th, 2018 | Project Plan, 3 Minute Presentation | The objective with this goal is to complete and submit a project plan as well as to give a three (3) minute presentation on the plan in class. |
| October 10th, 2018 | Software Requirements Specification (SRS) | The objective of this goal is to completely specify the SRS, a document detailing the requirements of the system. |
| October 17th, 2018 | Project Design, 10 Minute Presentation | The objective of this goal is to complete the project design document and give a ten (10) minute presentation of it in class. |
| October 31st, 2018 | Initial Version of Software, 10 Minute Demonstration | The objective of this goal is to implement an initial, working version of the product that showcases core functionality and give a ten (10) minute demonstration in class. |
| November 7th, 2018 | Test Plan | The objective of this goal is to complete the test plan for the product. The test plan will be used to verify and validate the product. |
| December 5th, 2018 | Final Software Product, 15 Minute Presentation and Demonstration | The objective of this goal will be to complete the software product itself. Implementation of all software features specified in the Project Plan and elaborated on in the SRS are to be complete and correct as determined by the Test Plan. |

In addition to the time table above, the project will keep a time log of all contributing members to the project.

Planned contributions to this project will take place largely on Friday through Sunday from 9:00 AM to ~5:00 PM as that is the best time slot available for me to dedicate full attention to the project. Additionally, there may be micro time slots allotted throughout each week as the project advances. Regardless, all time spent working on the project will be recorded in a time log.

Project Milestones

Note: The following milestones are for the implementation phase of the project, they are intended to drive development and have nothing to do with the design phase. Design phase milestones are near identical to the implementation milestones with the exclusion of system documentation generation.

Listed here are the project milestones to be completed in order:

1. Development of all data models, i.e. databases, classes, etc…
2. Development of backend software processes, i.e. course registration/maintenance, account creation, schedule viewing, etc…
3. Development of frontend graphical user interface.
4. System documentation generation, i.e. user manuals, etc…

Minor testing will occur during the process of completing each milestone. Major testing will occur after each milestone has been completed. I will only move onto the next milestone once all major testing passes.

Project Structure

The project will be maintained in a GitHub repository at the following link:

<https://github.com/gollum18/CourseRegistrationPOC>

Additionally, the project will contain the following folder structure:

|  |  |
| --- | --- |
| Folder | Description |
| CourseManagementPOC | The root folder of the project. Houses all folders described here as well as the Project Plan and Time Log. |
| CourseManagementPOC/Analysis | Contains work done prior to officially starting the project, like the project plan. |
| CourseManagementPOC/Design | Contains work done during the design phase such as design models and diagrams, as well as the System Design Specification |
| CourseManagementPOC/Implementation | Contains the source code for the project. |
| CourseManagementPOC/Testing | Contains files pertinent to testing the software such as the Test Plan, and all test cases. |
| CourseManagementPOC/Evolution | Contains files pertinent to evolving and maintaining the software such as user manuals and installation guides. |

This project will make best efforts to utilize a design-by-contract, layered architecture. Each layer will be responsible for only the tasks assigned to it and will be oblivious to the workings of other layers aside from those services it must utilize. The inner workings of the structure of the project will be solidified in the design phase of the project.