

Tom Byrne

Professor J. Dawkins

CS460 - Computer Science Capstone

February 5, 2013

Written Proposal

Goal

The goal of this project is to produce a web-based system that will allow staff members of the Non-Traditional Program(NTP) to search the WVUP course catalog for classes that fulfill specific degree requirements. This is currently done by hand by a member of the NTP staff. Each semester, whenever a student requires a specific type of class and asks the NTP office, a staff member must review the entire list of WVUP classes and look for classes that fulfill those specific requirements. They must then check that class's uniform syllabus or course description to verify that it does so. This tool will allow the staff to specify a set of criteria, for example: Humanities, 3 or more credit hours, Upper-Division, and the system will return a list of classes that fulfill these requirements, along with the number of available spots, the CRN number and other useful information (TBD). This tool will save dozens or hundreds of man-hours for the NTP staff. A secondary goal is to get this tool integrated into the NTP web site, so that students could use it themselves, without any NTP staff intervention.

Project specifications:

In order for this project to be successful, at a minimum, I will need to produce:

- A web based system that allows the NTP staff to query the WVUP catalog and schedule, and produce meaningful results.
- A tool for updating and maintaining the system's data - ie: what classes fulfill which requirements.
- A fully documented source base for WVUP, so that it can be maintained.

This project will require several parts.

1) Web Server

The de-facto standard of the NTP are web-based systems. This is a fairly standard trend throughout the education and business world. I will produce a customized web based tool that will be usable via standard browsers (IE, Firefox, Safari, Chrome). This will be backed by a web server that is standards-compliant and also capable of scaling to meet the needs of an enterprise or educational load.

2) Database

The tool will require custom data storage - each class at WVUP is unique, and assigned a specific CRN. Each class has many specific attributes, and some shared attributes. The data store will need to track the relations between these, and also be able to scale to a university sized load. In addition, the data definition layer should be abstracted so that the database can be replaced as needed in the future.

3) Standards-Based APIs

This tool will start as a standalone system, however, due to the desired expansion plans, all of the web-based interaction should be presented as a web-compliant API, so that it can be integrated

into other systems. This also keeps the possibility for other clients (such as an iOS or Android client) open.

4) Complete Deployment(Nice-To-Have)

The system will need to be completely packaged, including deployment checks. This will involve making sure that any required software exists, as well as instructions on how to install and configure dependent software, such as databases. This is to make integration by IS&T easier. This is a nice-to-have.

Dependencies

This project has two external dependencies. These are:

- **Integration with WVUP Records System**

The data for seeding the classes must come from WVUP. There is a canonical record that is maintained by the Records office and published to the web for student and staff consumption. A lower-level integration with this system via a publisher/subscriber or feed/polling mechanism is almost necessary for success.

The alternative to this is a much more fragile ‘web-scraper’ that would poll the existing published HTML via the WVUP web site and attempt to scrape this information and inject it into the DB. This alternative is error-prone, and I hope to avoid it.

- **Integration into WVUP Network**

This project requires deployment in a place that is accessible via the NTP staff. Given the requirement that it will need access to the Records system, the tool will need to be deployed

inside WVUP's secure network. This will require working with IS&T to follow any specific requirements that they have.

The alternative to this option is to run this on a single computer that is accessible only from the NTP staff's desktops, perhaps on an unused laptop or desktop machine. This is certainly less preferable, but is a possibility.

Outcome

I anticipate that there will be some difficulties in getting traction with IS&T and Records. I have had a discussion with my liaison in the NTP office and made her aware of them, and we will bring a proposal to both Records and IS&T as soon as possible in order to get things working as early as possible. In addition, I have planned out some less-desirable alternatives for both external dependencies that may be used in the event that we are unable to get traction and/or approval from one or both groups.

Success will be defined by deploying a working system for the NTP group, with or without the help of Records and IS&T, and having the NTP group use that system going forward.

Methodology

I plan to use several types of established methodologies:

1) User Observation

Observing a targeted end-user performing the task that this system will replace. This allows me to anticipate how the end users will attempt to use this system. This also allows me the

opportunity to discuss what parts of the current system the users like and don't like, and their thought processes behind the tasks

2) Interaction Design

Once a rough idea has been sketched out for the end-user, a prototype(either digital or paper) is brought to them, and they walk through the expected workflows and outcomes. This offers insight into the expected system and allows for rapid prototype refinements.

3) Prototyping

Once the main ideas have been worked out, I expect to have at least 2 sessions with early prototypes where the targeted user can see the system and attempt interactions.

4) Test-Driven Design

At a code and database level, I plan to utilize a test-driven model, where unit, regression and system-level tests will drive development of specific sections.

5) Load Testing (Nice-To-Have)

Since this system may be deployed for student use, I would like to do at least a preliminary set of load tests. Load tests will allow me to see if there are any bottlenecks in the deployment system, such as database connections or web sockets. This will be successful if we can reach an enterprise/education level of throughput, which I am defining as:

- Minimum complete requests as 1000/minute
- Maximum single request response time of 100ms, not including client->server latency

Chronology:

Week 1: ending February 8

- End-User Meeting
- Submit Project Concept Document
- Submit Project Specifications Document

Week 2: ending February 15

- Proposal to Records & IS&T
- Start technology evaluation for DB, Web & Application Server.
- Set up source control
- Set up bug tracking system.

Week 3: ending February 22

- Finish Technology evaluations
- Make technology decisions for DB, Web & Application Server.
- Write data definitions for DB
- Write API Definitions
- IS&T and Records communication

Week 4: ending March 1

- Write DAO Tests & Objects
- Write IO Tests & Code
- External Documentation outlines
- IS&T and Records communication

Week 5: ending March 8

- Integrate with Records & Data transforms.
- Rough UI
- First End-to-end tests
- UI Prototypes and discussion with end-users
- Schedule Presentation

Week 6: ending March 15

- End-User Meeting - Prototype 1.
- Evaluate Deployment platform, based on availability and IS&T restraints.

Week 7: ending March 22

- Alpha State
- Prototype 2
- UI Refinements

Week 8: ending March 29

- Feature Complete - Beta State

Week 9: ending April 5

- Deployment test 1
- User Meeting, Beta display.
- Debugging

Week 10: ending April 12

- Final refinements
- Deployment test 2
- Users should be able to use the tool by 4/12.

- Debugging

Week 11: ending April 19

- Debugging
- Final Deployment
- Presentation.

Week 12: ending April 26

- Deliver Final Documentation and Code

Priorities

Must Have:

- Web Server
- Database
- Interactive UI for Searching
- Interactive UI for maintaining data

Nice To Have:

- Integration into NTP website
- All-In-One Deployment & Dependency Checking tool
- Enterprise Scalability

Extra Time:

- Mobile App (Android/iOS)
- WVUP-wide accessibility via OLSIS

Cost Estimates

Hardware

The main cost incurred here will be the cost of a server to host this application. My anticipation is that for the NTP staff, any modern desktop or laptop will do the job. I have spoken with the NTP group and they have several extra computers that could be made to do this task.

If this is expanded to be hosted in the external web site, a larger server will be needed, and this cost could be \$1000-5000, depending on the estimated load. A baseline Dell PowerEdge R510 would cost in the \$2000 range, and one with maxed RAM, RAID5 and redundant power supplies prices out at roughly \$4000.

This is not expected for the completion of this project, but should this system be deployed for general use, this money will need to come from the college's budget, either from NTP or from IS&T. It is also possible that the IS&T and web groups may have servers or VM's available for these tasks.

Software

The intention is to use all free software.

- OS: Linux
- Language: Java
- DB: Postgres or MySQL
- Application/Web Server: Apache Tomcat or JBoss Netty.

This is an excellent solution, as they are widely used and known technologies for this problem space. This also keeps the cost down to a minimum.

Time & Personnel

I do not anticipate any costs for personnel at this time. The NTP liaison will interact with me as part of her regular job, and my time is effectively free.