**VirtualAdvisor**

**Author**: Nic O. Falcione

**Professor:** Chris Kreider

**Institution:** Christopher Newport University

**Course:** CPSC 498

**Abstract**

This project will aid the complex process of course advising for professors and students, specifically those at Christopher Newport University. Often, core advisors are expected to advise 30-60 students on their path to complete the minimum 120 credit hours, major and minor requirements and general requirements. The product to be completed in this assignment will be a website with a user-driven experience that recommends course paths based on the students completed requirements and preferences. An intelligent algorithm on the backend will automate this difficult process for professors that advise students.

Table of Contents

Introduction 3

Proposed Solution 3

Development Approach 3

High Level Plan 3

Project Schedule 4

Intermediate Milestones 5

Intermediate Milestone #1 5

Intermediate Milestone #2 5

Weekly Updates 5

Weekly Update #1 5

Weekly Update #2 5

Weekly Update #3 5

Weekly Update #4 6

Weekly Update #5 6

Weekly Update #6 6

Weekly Update #7 6

Weekly Update #8 6

Weekly Update #9 6

Project Deliverables 7

Conclusion 8

**Introduction**

As a computer science major and lover of technology, I have always had a goal to produce a working website that helps others accomplish a difficult task. This project combines my passion and experience in automation, backend optimization algorithms, database management, and system integration. If this project becomes successful, I’d love to deploy it for the CNU community and genericize its ability as much as possible. University advising is a difficult process that is completely case specific. The complexity of prerequisites, requirements, and the possibility of students having multiple majors and minors make it even more challenging. One professor cannot optimally advise over 50 students as the number of possible course paths amount to a number that approaches infinity.

**Proposed Solution**

**Development Approach**

I plan on developing my project and code with an agile development cycle while tracking my progress with a Kanban board. I plan to keep the agile format of implementing functionality in user-story format, and handling features and bugs as such. Also, I plan on performing three phases of testing: Unit Testing, System Integration Testing, and User Acceptance/Beta Testing. I feel as this is the best approach for the most productivity and to feel the most connected and responsive to the stakeholders and users of the product I plan to create.

**High-Level Plan**

The research phase will consist of identifying which languages and technologies I need to successfully complete my project. I need to identify what programming language will best work for the webpage U/I on the client-side, what database management service I will use, what type of web server I need, and how to setup the database to accurately represent the course catalog for CNU. Setting up the web server should not be too difficult; however, the system integration part will most likely take up the most time. I plan on prioritizing this over a developing a fancy webpage. The database will contain course relevant information such as prerequisites, subjective difficulty, and availability.

I plan on converting user-input to lower level variables to optimize on the backend, essentially putting together some recommendations with a constraint-based, flexible fitness function. I will use my experience with heuristic optimization algorithms for minimizing critical path of project’s scheduled tasks with predecessor/successor relationships to accomplish the project. I hope to have a survey that considers completed course requirements, major, minors, how a student wants to balance/frontload their college experience, and other customizations that provide a great advising result.

I will spend a great deal of time fine-tuning the algorithm with unit-test cases, adding small features as I go. System Integration testing will be the lengthiest process since there is a lot that can go wrong such as server problems, timeouts, U/I glitches, and database management problems. Lastly, I hope to have a user-acceptance testing phase where I start by asking my friends to test it out and provide me suggestions and report bugs. I hope to expand this to more of the CNU student body as I get closer to deployment.

**Project Schedule**

**Intermediate Milestones**

Milestone #1:

* Present to the class the skeleton website and demonstrate web server, U/I, and database connectivity

Milestone #2

* Demonstrate algorithm and functionality of the website to the class
* Allow suggestions for user-input and show results

**Weekly Updates**

Week of 2/4/20 (No Weekly Update): **Research/Design Phase**

Week of 2/11/20 (Weekly Update 1): **Setup Web Server**

* Create Amazon Web Server, connect to EC2 instance and RDS (2 hours)
* Test functionality of web server (1 hour)

Week of 2/18/20 (Weekly Update 2): **Create Database Schema/Populate**

* Create database schema (2 hours)
* Populate database with computer science courses (2 hours)

Week of 2/25/20 (Weekly Update 3): **Integrate Database & Web Server/Begin Web Development**

* Connect database to web server and test initial functionality (2-3 hours)
* Design Web U/I (2 hours)

Week of 3/3/20 (No Weekly Update): **Continue Web Development/Integrate all Parts**

* Begin to build U/I with JavaScript, PHP, and angular,js (2 hours)
* Ensure web server contains correct user-input and output path system 21 hour)
* Verify DNS/html functionality (1 hour)

Week of 3/10/20 (Weekly Update 4): **Start User Input/Optimization Algorithm**

* Create user-input surveys for information that the algorithm can use to recommend courses (1 hour)
* Connect to database to store information for user (1 hour)
* Design algorithm fitness function (1-2 hours)

Week of 3/17/20 (Weekly Update 5): **Continue Optimization Algorithm/Integrate into Web Server**

* Push backend code to web server and git (ongoing)
* Connect low-level variables in fitness functions with logic from user-input (2 hours)
* Setup queries from database that are relevant for user (2 hour)

Week of 3/24/20 (Weekly Update 6): **Unit Test Algorithm/ Fine tune Course Path Suggestions**

* Design test cases for algorithm (1 hour)
* Perform testing (2 hour)
* Fine algorithm based on testing (2 hours)

Week of 3/31/20 (Weekly Update 7): **System Integration Testing**

* Test database queries to output after user-input end-to-end (3 hours)
* Ensure web page does not break and U/I function correctly (2 hours)

Week of 4/7/20 (Weekly Update 8): **Fine Tune U/I of Webpage**

* Add to U/I (logo, pictures, etc.) (2 hours)
* Ensure course suggestions look presentable (2 hours)

Week of 4/14/20 (Weekly Update 9): **User Acceptance Testing/Finishing Touches**

* Open the website up to friends and colleagues for initial user-testing
* Analyze results and perform updates (3 hours)
* Spread the word about the website for as many users as possible to try to use the webpage

Week of 4/21/20 (No Weekly Update): **Genericize if Possible/Deploy!**

* Work as much as possible to genericize algorithm to work for more majors

**Project Deliverables**

**Documentation**

* Stakeholder Documentation: Results from the user’s that try the website
* Architecture Diagrams: Web page U/I decision graph, database schema design/population
* Database Entity Relationship Diagrams
* System Vision Document
* Kanban/Agile User Stories and Progress reports

**Technical Deliverables**

* Link to Git repo containing all code for the project
* Link to URL of the working website
* ZIP Archive containing the entire project source code and scripts
* Intructions/README

**Final Presentation Deliverables**

* Give class demonstration of webpage functionality
* Let them test for themselves if they haven’t already
* Presentation of project
* Project Tri-Fold Board

**Conclusion**

This goal of this project is to lighten the load off professors that are tasked with advising many students. I hope that the product will be of use to CNU staff and students in the future. Additionally, the project will take at least ten weeks of hard work, with 5-10 hours of work done each week at a minimum. My concerns come with the complexity of the course requirements that CNU majors have to offer and developing a good database schema and population to query from. That being said, I do believe that the finished product will help people build a course path that feels right based on their goals.