

Rhiannon Tully-Barr	Just Right Showers
V00801550	University Degree Planner
21 Jan 2016	2.0

Version	When	Who	What
1.0	06 Jan 2016	Rhiannon Tully-Barr	Initial Drafting
2.0	21 Jan 2016	Group 5	Detailed Request for Proposals

Table of Contents

- 1.0 Problem description
- 2.0 Project objectives
- 3.0 Current systems
- 4.0 Intended users and their interaction with the system
- 5.0 Known interaction with other systems within or outside the client organization
- 6.0 Known constraints to development
- 7.0 Project schedule
- 8.0 Project team
- 9.0 Glossary of terms

1.0 Problem description / expression of need

Designing a University course schedule that satisfies all relevant prerequisites as well as life restrictions can be a difficult task. Even when a schedule is determined, such as the one provided by the school administration, the slightest alteration (for example, failing a course) requires a re-working of the entire schedule.

This is an especially pressing problem for students in highly restrictive degree programs. For example, the engineering disciplines have a large number of required courses compared to electives, and many courses that are offered only once per year. This can easily cause a problem for students if there is a disruption in their schedule. Example disruptions include failing a course, taking a term off, reduced course load for part-time employment, and extended co-op work terms.

Here at Just Right Showers, we employ a number of junior developers who have not yet completed their undergraduate degrees in a part-time capacity. We are seeking a solution that will allow our employees to complete their undergraduate degrees as quickly as possible so that they can become full-time employees.

2.0 Project Objectives

Provide a software solution in the form of a website that accepts a list of courses that have been taken and courses that remain to be taken, and provides a valid schedule of courses for the remainder of the degree. The user should be able to log into the system so that this information will be saved and modified when courses are completed. The software should provide a handy tool for students to figure out how to graduate on time or on their own terms. It should help with the headaches that come along with trying to fit school and life requirements into a schedule.

3.0 Current System(s)

The University of Victoria website has some functionality to help students with degree planning. However, it is severely limited. The degree planning section (CAPP report) can only be accessed once a degree is declared, which generally occurs when students enter their second year. The report simply displays the courses taken and those yet to be taken, but contains no information about term availability or prerequisites and offers no planning or scheduling functionality.

The University of Victoria has a database of information concerning course prerequisites and term availability that would be necessary for the software to function well. The software should have up-to-date information about courses.

4.0 Intended users and their interaction with the system

Intended users are students of the University of Victoria, though the application should be designed in such a way that it is re-usable and could be integrated into other post-secondary school systems. Students would interact with the system to help them plan a University degree, in conjunction with discussing the plan with an academic advisor.

The software should begin by allowing the user to select a degree program. This list should be complete and updated using information from University of Victoria databases, including combined degree programs. The user should also be able to select "undeclared" if no degree program is applicable. If this option is selected, the user will be required to enter their list of courses taken, and a list of courses they desire to take, manually.

Following this choice, the user should be presented with a list of the required courses that they must take, divided by term, accompanied with check boxes. The user will check the boxes of courses that have already been completed. An entire term worth of courses can be selected at once, by checking the box associated with that year. If "undeclared" was chosen, the user will input the courses they want to take and those they have taken instead. There will be a drop down menu that is search-able and contains all of the courses offered at Uvic. When selected, the courses will be added to their respective lists (taken/not taken); the lists will be visible to the user.

Once the courses have been specified, the next page will allow the user to input additional restrictions and criteria. This page can be skipped by the user.

The software should accept the following criteria from the user:

- Maximum and/or minimum number of courses taken per term;
- Preferred year of graduation;
- Black-out terms: terms where no courses are taken due to co-op, travel or break;

The generated schedule should conform to the following restrictions on courses from the University:

- Courses only offered in specific terms of the year (spring, summer, fall);
- Courses that require students to be in a certain year;
- Courses that require students to have completed a Co-op term (399,499)
- Pre- and co-requisite restrictions;
- Maximum amount of credits permitted per term.

Once the restrictions are selected, the algorithm will run on this data and generate a valid schedule. The user will be able review and modify this schedule further.

It is expected that, for typical inputs to the algorithm, many valid schedules are possible. The algorithm should select the "best" option for a schedule. "best" will be defined as

- A schedule meeting all of the user's preferences and is optimal in some category, like degree length; or,
- If no possible schedule meets all of the user's preferences, a schedule that meets the largest number of criteria.

A schedule should be dynamically modifiable by the user. A click-and-drag interface is preferred. The user should be able to easily and intuitively select a class and move it into a different term. The suggested schedule should adapt when the user does this, for example, moving prerequisites into an earlier term so that the schedule remains valid. A valid schedule is one that meets all hard requirements (term offerings, prerequisite requirements, and amount of credits per term). If a change is made by the user that would produce an invalid schedule, the software should prevent this change and notify the user. It should provide the user with an option to manually override this restriction, as prerequisite requirements can be waived at the discretion of the department.

When modifying the schedule, the restrictions the user has given should be displayed on the screen and be modifiable. The user should be able to add requirements, remove them, and change their level of priority at this stage. The schedule should be re-generated any time a requirement is changed, in order to comply with the new requirements.

Finally, the user should be able to save a given schedule to their user account, revisit it and continue to modify it over the course of their degree.

5.0 Known interaction with other systems within or outside the client organization

The system will interact with the University of Victoria course database. The interface between the information provided by the university and the web application will require negotiation and co-operation.

If possible, some interaction with the existing website schedulecourses.com would be desirable, for importing a term's courses into the scheduling website. This would allow the user to plan their weekly schedule for a single term, and easily register for the courses they require. Ideally, this functionality would be available through a single click from our software. The feasibility of this feature will require further study.

6.0 Known constraints to development

The system would need to be compatible to some degree with existing University of Victoria systems. Ideally, the system would allow a student to import their finished schedule into the University's registration system on a term-by-term basis, simplifying the registration process.

The system will require enough memory to store information for all users in the form of their login information and the schedules they have created.

The level of co-operation and information sharing between the University and developers of the system could create constraints on development.

The project development time cannot exceed 4 months.

7.0 Project Schedule

21 Jan: Informal requirements definition; project website up and running

26 Jan: Customer feedback

16 Feb: Formal requirements specification

18 Feb: Customer feedback on requirements specification

1 Mar: Detailed requirements specification

3 Mar: Software service prototype demo; prototype can generate a valid schedule when provided with all course data

8 Mar: Customer feedback on prototype demo

15 Mar: Final requirements specification

22 Mar: User Manual written and provided as a tutorial on the website itself

24 Mar: Customer feedback on user manual and final requirements specification

29-31 Mar: Demo final project. Website generates valid schedules from course data and incorporates user's choices. Customer feedback on demo.

8.0 Project team

Puzzle(™)

Team members: Brendan Hell, Jonah Rankin, Ali, Nobari, Spencer Mandrusiak

9.0 Glossary of terms

Application: A standalone computer program.

Co-op: A temporary work placement completed as part of a University degree, for credit.

Demo: A demonstration to the customer of a partially or fully functional software product.

Dynamically modifiable: Users should be able to modify the course schedule, for example by moving one course to a different term, without having to re-create the whole schedule.

Requirements Specification: A document detailing the requirements that the delivered product should satisfy.

Prototype: An early version of a software application; A proof of concept that may meet some number of the customer's requirements, designed to demonstrate the progression of product development.

Software: Computer application designed to complete a task.

System: Encompasses the software part of a product as well as all of its interfaces with people, other systems, hardware, etc.

Executive Summary (1 page)

The objective of the proposed project is to create a web service allowing University students to plan out when to take each course to complete their degree. The service would allow for easy management of schedule restrictions, such as pre-requisites and timing constraints. The software will also account for user-defined constraints, such as off terms for travel opportunities or restrictions on the number of courses per term. The web service will be tightly integrated with the University of Victoria's systems in order to access all the course information necessary to create a schedule. The schedules created by the service will have a level of interactivity, allowing the user to move courses around to different terms, while providing a warning if such changes violate prerequisite requirements. The service would benefit the end user by facilitating planning, allowing the student to easily see which aspects of their plan are flexible and which are not, and will be a useful tool to plan a degree program in conjunction with help from an academic advisor. The web service will have the advantage of being able to adapt with the changing circumstances of the student, and the schedule itself can be modified when changes need to be made rather than needing to create a new schedule from scratch.

Project objectives:

- Provide University students with an easy-to-use web service that allows them to arrange a degree plan around their needs
- Accept restrictions provided by the user as well as those imposed by the University
- Help students figure out how to graduate as fast as possible while also accommodating their needs
- Allow students to figure out how to recover from unexpected changes like 8-month co-op terms, failed courses, or other disturbances to the plan

Requirements:

- Dynamically modifiable schedules, to allow the user maximum flexibility
- Tight integration with the University of Victoria's database of course information
- The ability to store user information