Rhiannon Tully-Barr	**<Company Name>**
V00801550	**University Degree Planner**
06 Jan 2015	**1.0**

Version	When	Who	What
1.0			Initial Drafting

Table of Contents

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.

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 software should be able to accept a number of criteria as restrictions, such as the maximum and minimum number of courses taken per term, preferred year of graduation, co-op or travel terms with no courses, and term restrictions on course offerings. The suggested schedule should be dynamically modifiable by the user, but should prevent the creation of a schedule that violates prerequisite orderings.

3.0 Current System(s)

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.

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

University of Victoria course database

6.0 Known constraints to development

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

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

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

Project team has not yet been selected.

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.

Project Proposal Summary (1 page)
Rhiannon - V00801550

The proposed project would create a web service allowing University students to easily plan a course schedule. The service would allow for easy management of pre-requisites and external requirements provided by the user. For example, lifestyle restrictions such as taking a maximum of four courses per term in order to work part-time, or taking a term off for travel opportunities. 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 provided that such actions do not violate prerequisite requirements. Such a 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.

My Résumé

Rhiannon - V00801550

Project management experience

Co-op work term with Ericsson – experienced working closely in a team of 10 developers, and some interaction with 170+ developers, on an extensive codebase.
Course work group projects – designed a robotic system for an engineering design course, have completed course work in groups of 4-5 in several courses at uvic.
Involved with Uvic Aero, an aeronautics club designing unmanned aerial vehicles. Experienced working in a small embedded software team to create software that interacts with components created by the other teams (mechanical, electrical, high-level software).

Writing experience

Course work – Technical writing course, including a detailed description of an engineering project.
Work term reports – During my co-op, I completed two work term reports concerning the work that I had done over 4 months. I also created documentation for the software tools I had been using, for the benefit of subsequent co-ops and regular employees.

Webmaster experience

Very little web development experience - some cursory html and css knowledge.

Software tool expert

Work term – A large part of my work on co-op involved managing a software tool set that maintained the integrity and consistency of our team's codebase. I am experienced with git and Jenkins, and comfortable in a linux environment and writing shell scripts.
Course programming experience – Through courses such as SENG 265, I learned about programming tools and environments including svn, gdb – debugging tools, text editors / integrated development environments, and the linux environment generally.

Programming skills

Programming languages – C, Java, python
Scripting languages - none

Design experience

Design experience and design tools - none
UML diagramming & object-oriented design skills – none
Some object-oriented programming experience through course work
Interface expert—user interface programming & presentation skills
Currently enrolled in SENG 310, which concerns user interface design.

Requirements engineering experience

Requirements elicitation – Some related experience on co-op, as I was modifying software tools according to the specifications of my team and manager.
Requirements analysis – Co-op experience: When asked to modify and improve software tools, I was required to determine whether a particular modification was possible or worthwhile.
Requirements verification, traceability – A part of my co-op work term report involved a comparison of the software tool performance before and after a significant overhaul of the system, to determine what improvements have been made.
Testing and reviewing skills – On co-op, ran detailed code profiling software on my team's software in order to validate the completeness and comprehensiveness of our test suite.