Assignment 2:
Component 2

Lindan Thillanayagam (213742176)

Saad Saeed (213968284)

Date: November 14, 2017

Course: EECS 3461

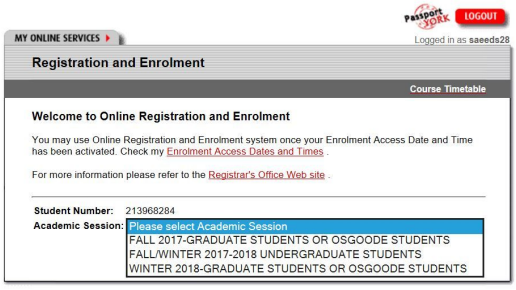
Professor: Melanie Baljko

User Needs

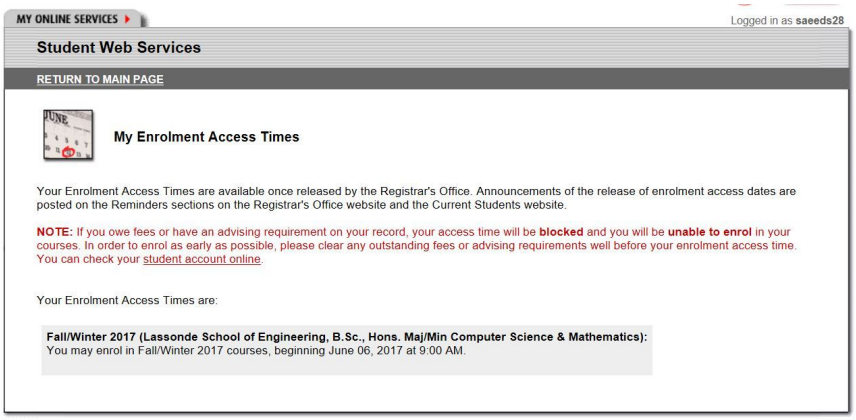
Observation-based requirement

The following requirements are solely based on observing users use the current Registration Enrolment Module (REM) at York University:

1) Users were brought to the start of the main REM page after logging in with their Passport York Credentials. A drop down menu was presented where the students picked the year they wanted to enrol in courses. **The program should be dynamic enough to realise which faculty the student belongs to and for which year the student is enrolling for.**



2) Currently, the way to access enrolment dates is a painstakingly long process. The user currently has to bypass three web pages before the enrolment date appears. Clearly this is a very unintuitive way to access something so obvious and useful. As a result, **all important data should be centralized to the main REM application to minimize user frustration.**



3) Some of the UI elements are very unintuitive and serve no purpose in the main application whatsoever. For example, the red triangles next to the course names should be clickable and should expand to bring up the course details. Instead they serve no purpose and a

Course Detail button is pressed to bring up the course information. The REM is littered with bad and unintuitive design choices. As a result, **the GUI for the REM shall be redesigned to make it more intuitive and pleasing for the users.**

4) It was observed that users would become frustrated and angry whenever they tried to enrol into a course that was full or reserved. Currently, the users have to find the catalog number for a course, enter it into the REM, and hope that they get enrolled. There is no way to know whether the course is full or reserved before the catalog number is inputted into the REM. People become frustrated when they realize that their efforts have failed and are greeted with a message saying “the course is full” or “the seats in this course are reserved”. There needs to be a way of letting people see the number of available seats before they commit to a course. As a result, **the proposed REM will be more dynamic and transparent in providing users with critical information before they commit to an action.**

Interview-based requirement

The following requirements were solely formulated after interviewing students at York University:

1. One of the things that people were complaining about was how hard it was to work with small text. **Therefore, the new REM will offer bigger text for those who cannot see well.**

2. Searching for courses in the course catalog takes time and people hate switching between windows. Everyone that was interviewed said that the course catalog was their least favourite part of the whole experience and wanted the catalog gone. Therefore, **the functionality of the course catalog shall be incorporated into the main REM application.** Users will be able to enter the course codes and relevant information will be displayed in the same application. An access to an external database will be required.

3. The students, who were interviewed, complained about the disconnect between the course enrollment module and their degree progress. Prior to 2016, students of Lassonde, in particular computer science majors, had no access to the degree progress report provided by the university. At this point they were only able to check their progress by comparing their transcripts with their already confusing academic calendars. They found this to be tedious and they were more prone to making errors. For instance, they may assume that they have completed a requirement even though they might have not. After they had access to the degree progress report provided by the university, they felt that there were some

improvements. However the degree requirements stated on the degree progress reports were sometimes outdated. For instance, they were to complete a course which did not exist during their time at the university. Also, numerous students have complained about the lack of valid information provided by the academic advisors with regards to their progress. Sometimes the information they give can deter students towards making a mistake. To avoid these problems, **the improved REM will incorporate their academic progress along with a list of possible courses they can take.**

User profiles

Future first-year university student

The average first year student is unaware of the procedure for choosing courses. They are very nervous in coming to a new school as they are “starting from scratch”. They are very inexperienced in choosing courses because they had a limited number of courses to choose from while they were in high school. Due to their inexperience, they are more prone to making errors while using the REM. They will have set of predefined 1st year courses that they are expected to take. For someone who is using the REM for the first time, it is essential for the course enrollment process to be a simple and enjoyable experience. They already have a lot on their plate because they are still familiarizing themselves with the inner workings of the university and the task of course enrollment should enhance their experience, not hinder it with the introduction of unnecessary nuances.

Future fourth-year university student

The average third-year student is familiar with the procedure for enrolling into courses. As they near entering their fourth year, they will have a better idea of the path they would like to follow, thus a set of courses they would like to take. They have already completed a certain amount of credits and are aware of how they can affect the paths they can pursue in the future. This student will review the academic calendar to view what courses they have to take next, and glance through the course catalog for course availability, only to make a decision thereafter.

Depending on the faculty, the academic calendar could be unorganized and vague, requiring users to place greater effort and time in order to make sense of the steps they have to take next. For the student to ensure that he/she is taking the right courses, he/she must consult advisors and the academic calendar frequently. Some students do not have time for such nuances as it conflicts with their schedule. Also, the

availability of an advisor may not be convenient for the student. This could place a burden on a student that either has little to no time to consult an advisor, or on a student willing to sacrifice time to see an advisor. For the former, there could be an increase in the possibility of making a poor decision. For the latter, the student allocates time for seeing an advisor, which could have been used towards doing something else. Changes made in their program may not be reflected in their program's academic calendar, causing uncertainty in the possible decisions to be taken.

PERSONA

Jeff is currently a third year computer science student at York University. He enjoys photography, working out, and cycling. Generally, he is known for having a laid-back attitude towards life, but when it comes to school, he is very serious. By his peers, he is described as a hardworking, organized, and dedicated individual. As a result, he has gotten good grades in university. Since grade 10, he has had an interest for computer science and worked hard throughout high school to get accepted into a university offering a good computer science program. At the beginning he was unsure about whether this interest was genuine, but after spending a year in the program, he decided that it was the program for him. From this, he is kept motivated after seeing where his hard work has lead him. His goal since his first year was to learn as much as he could from his program, and, as with any other student, graduate. With each passing year, Jeff has learned more about his weaknesses. For instance, he found that when something was presented to him in a disorganized manner, he would be more prone to making errors during evaluations. He has not had enough opportunities to work on this problem since it does not occur frequently.

Scenario 1:

Vivian Note is a recent high school student who is going to a well-respected university in a few months time. Like all first year students, he is nervous about starting his university career at a new school and is very eager about knowing the in's-and-out's of the place, finding where everything is located, and learning the procedure of enrolling into courses for the Fall semester. He has already taken a campus tour and has spoken to his academic advisors about what courses to take and when he can enrol. It turns out that his enrollment date is in a couple of days. Rather than investing in learning about how he can enrol into courses, he decides to wait until the enrolment date because he knows that the enrolment module will be intuitive and enjoyable to use. He bases this assumption on the fact that he goes to a world-renowned university.

This past week has been very busy for Vivian as he is overwhelmed with the amount of information that needs to be processed. Furthermore, his enrolment window has opened up. He logs into the computer with his credentials and a main window pops up with his personal information, along with information on his program, faculty and what semester he is trying to enrol in. Since he will be in first year, the window offers

him a list of first-year courses specifically based on his program of study. Since he has never taken any courses, upper-year courses will not be listed-nor will he be eligible to enrol in them-because he does not have the prerequisites for them. The GUI also offers a list of pre-approved faculties that he can take electives from. Furthermore, the program offers a textbox that allows Vivian to enter the course code. When he does enter the course code, the program fetches everything associated with the program (course name, instructor, time, whether the course has a lab or a tutorial, and the number of seats available in a particular section). If there is a seat available for a course, the program allows the user to press the Add Course button. Vivian presses the button next to his Introduction to Java programming course and a popup message appears that reads, “Success! The course has been added. At this point, Vivian is feeling great because he was able to enroll in all his courses without any issue. Furthermore, he loves the Registration Equipment Module (REM) because it is very simple to use and does not burden the user with a responsibility of knowing what courses he is eligible for beforehand. He also loves how everything is easier to see-Vivian has a hard time seeing the small text-and applauds the designers about taking accessibility requirements into consideration. Furthermore, he appreciates how everything is laid out on the screen; the visual cues are pretty great. Feeling amazing, Vivian’s time in university is off in the right direction.

Scenario 2:

Jeff’s third year is coming to an end, and he will start his fourth year in the following months. For the first two years of his program, managing his courses was easy as there was only a single course path to follow. In his third year, Jeff had more freedom to explore the many paths of his degree by taking courses in various fields, but as a result his course choices for fourth year have been narrowed down. It turns out that he wanted to focus on courses related to interactive systems. While he is aware of a few courses that must be completed in his fourth year, he is unsure of what else he can take. As the “degree progress report” page is under maintenance, the student is relieved to know that he can view his information on the REM. For instance, his program, faculty, the term he is to enrol for, and his progress. So when Jeff decides to add a course, he is presented with a list of possible courses. He recalls a course that he heard about, and finds it in the suggestion list. So with the click of a button and a confirmation, the course was added, and real time information was provided that the course will fulfill a degree requirement. He then recalls two years ago hearing about a course that must be taken, but he wonders why it did not appear in his “suggestions” list. He looks for the course code in the course catalog and attempts to add it, but upon input, real time information is provided which states that the student is missing a prerequisite, and as a result, the student is blocked from enrolling into the course. He then decides to enrol into the course, which was a prerequisite for the course he initially wanted to add, and finds it in the “suggestions” list. From there he learns that the course also consists of tutorial and lab periods,so he picks them based on his schedule. After this Jeff decided to remove the first course added by pressing the “remove” button next to the course name. From here, Jeff adds two other courses but decides to exchange

one of them with another course. So he presses the “course exchange” button to replace it with another course. After all this, Jeff feels prepared to take on the upcoming year.

Main Task:

The main task of the application domain is to help York University students manage their courses for each term. As typical of many enrollment modules, the basic functions include adding, removing, and exchanging courses. However the proposed product will simplify these functions by getting rid of the many intermediary steps. Within each course, there could exist multiple tutorial and/or lab periods which will also be included alongside the courses. Students will manage their courses depending on their program requirements, availability, and scheduling. A Hierarchical Task Analysis (HTA) is provided below:

0: In order to manage courses
1: Login to the system using credentials
2: View current program progress
3: Identify course of action
- 3.1: Add course
  - 3.1.1: Find course to add
    - 3.1.1.1: Find on course catalog and input course code or use suggestions
  - 3.1.2: Check if able to add course
    - 3.1.2.1: Check if prerequisites are satisfied
    - 3.1.2.2: Check if seats are available
  - 3.1.3: Select tutorial period or lab period if available
  - 3.1.4: Confirm addition of course
- 3.2: Remove course
  - 3.2.1: Select course to remove
  - 3.2.2: Confirm course to remove
- 3.3: Exchange course
  - 3.3.1: Select course to exchange
  - 3.3.2: Select course to exchange with
    - 3.3.2.1 : Find on course catalog and input course code or use suggestions
  - 3.3.3: Select tutorial period or lab period if available
  - 3.3.4: Confirm exchange of course

Use Case:

1) User logs into the system using university credentials.

2) A GUI appears with relevant student information (name, program faculty, enrollment date, list of required courses).

3) The user is given the option to enrol into, remove, and exchange a course.

3.1) The user tries to enrol into a course by entering the course code or by selecting from a list of suggested courses.

3.1.1) All the relevant information associated with the course appears (course name, instructor, lab/tutorial, time, etc) if the course code is valid

3.1.2) The user presses the “Add Course” button and a success prompt appears after the program checks for the seat availability and checks if the prerequisites are satisfied.

3.2) User removes the course by pressing the “remove” button next to the course name.

3.2.1) A success message appears when the task is successfully completed.

3.3) The user exchanges the courses by pressing the Exchange Course button.

3.3.1) Two text boxes appear, one asks the user to enter the course code for the course that would like to exchange and the other one asks the code for the one they would like to exchange it with, or they can select the course they would rather take from the suggestions list.

3.3.2) User enters the information and the program checks for seat availability and prerequisites before printing a success message, if everything goes right.

Alternative Courses

3.1.0) If the course is invalid, a failure message appears and the user is prompted to enter the course code again.

3.1.3) If the seats are unavailable in one section or if the student is missing prerequisites, a failure message appears which specifies the condition as to why the course addition failed. The user is then allowed to check the specific reasoning for the failure.

3.3.3) A failure message appears if the course codes are incorrect, if seats are unavailable, or if the user is missing prerequisites

Volere Requirements:

Requirement #: 1	Requirement Type: 3		Event/Use Case #: 1
Description:	The product shall allow the user to log in with their credentials		
Rationale:	Need to be able to log in to the REM to do anything		
Originator:	University Student		
Fit Criterion:	The login information is validated against database		
Customer Satisfaction:	5	Customer Dissatisfaction:	5
Priority:	VERY HIGH	Conflicts:	None
Supporting Materials:	See REM of York University		
History:	Raised by Saad Saeed on November 14th, 2017		

Requirement #: 2	Requirement Type: 9		Event/Use Case #: 3.1
Description:	Create course suggestions when adding/exchanging courses		
Rationale:	Reduces usage of external resources		
Originator:	University Student		
Fit Criterion:	The user does not input course code if there are suggestions.		
Customer Satisfaction:	4	Customer Dissatisfaction:	3
Priority:	MEDIUM	Conflicts:	NONE
Supporting Materials:	See REM for York University		
History:	Raised by Lindan Thillanayagam on Nov. 14, 2017		

Requirement #: 3	Requirement Type: 9		Event/Use Case #: 2
Description:	The product shall display the current progress of the student		
Rationale:	Student is up to date to his/her progress		
Originator:	University Student		
Fit Criterion:	Their progress can be monitored by the student		
Customer Satisfaction:	4	Customer Dissatisfaction:	3
Priority:	HIGH	Conflicts:	None
Supporting Materials:	See REM of York University		
History:	Raised by Saad Saeed on Nov. 14, 2017		

Requirement #: 4	Requirement Type: 9		Event/Use Case #: 3.1.2
Description:	The product shall allow the student to enrol in a course if prerequisites match and if seating is available.		
Rationale:	Can't really make an effective REM if the option to add courses is missing		
Originator:	University Student		
Fit Criterion:	Checking to see if a student fails to add a course by monitoring prerequisites and check seat availability from a database		
Customer Satisfaction:	5	Customer Dissatisfaction:	5
Priority:	VERY HIGH	Conflicts:	None
Supporting Materials:	See REM of York University		
History:	Raised by Tyler Noble on Nov. 23, 2017		

Requirement #: 5	Requirement Type: 9		Event/Use Case #: 3.2
Description:	The product shall allow the user to remove a course		
Rationale:	Student is up to date to his/her progress		
Originator:	University Student		
Fit Criterion:	Their progress can be monitored by the student		
Customer Satisfaction:	4	Customer Dissatisfaction:	3
Priority:	HIGH	Conflicts:	None
Supporting Materials:	See REM of York University		
History:	Raised by Vivian Octave on Nov. 14, 2017		

Requirement #: 6	Requirement Type: 10		Event/Use Case #: 3.3.1
Description:	The product shall allow the user to exchange a course in a well-designed GUI		
Rationale:	Students will be more pleased with a well-designed GUI		
Originator:	University Student		
Fit Criterion:	The amount of time it takes for the students to enter the courses		
Customer Satisfaction:	3	Customer Dissatisfaction:	2
Priority:	MEDIUM	Conflicts:	None
Supporting Materials:	See REM of York University		
History:	Raised by Benjamin Korobkin on Nov. 14, 2017		