**SYSC 4504**

**Project Report: Courses Selection Assistant**

**By:**

**Mohamed Bakroun**

**100779232**

**(Mohamed Ahmed)**

**100828374**

**Mohammed Al Fayez**

**100753511**

**Date: 28th November 2014**

**1)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | STUDENT NUMBER | FIRST AND LAST NAME | EMAIL |
| 1 | 100779232 | Mohamed Bakroun | mohamedbakroun@cmail.carleton.ca |
| 2 | 100753511 | Mohammed Al fayez | mohammedalfayez@cmail.carleton.ca |
| 3 | 100828374 | Mohamed Ahmed | mohamedahmed@cmail.carleton.ca |

**2) We used Software Engineering Program to test our application.**

**3) Mr. Abaza**

**4)**

**Mohamed Bakroun - HTML Client/Database:**

* Dynamically add course to DOM
* Take users program and year information
* Setup the database given by the Professor into a single database
* login page - show either **Login** or **Sign Up**

**Mohammed Al Fayez - Java Client**

* Create interface from course data using netbeans
* Login in using Java
* Lets the student into the completed courses into a given array
* Create View 1 of the project

**Mohamed Ahmed – Server**

* Create interface from course data
* create REST API
* handle routing for AJAX requests
* On pattern - return list of courses for this semester
* Off pattern - Select courses for this user's program that they can take in this semester
* Select courses in the program where the prerequisite is null or is one of the courses the user took

**5)**

**3 Folders: Client, Server, Java Client**

**course\_selection\_assistant/**

* **datafall.csv** - fall courses database file
* **datawinter.csv**  - winter courses database file
* **client/** - files required by the browser client
  + **css/** - css stlying
  + **img/** - images
  + **js/**  - JavaScript for front end logic
    - **login.js**  - logic for login forms
    - **main.js** - logic for course timetable builder
    - **process\_prereqs\_in\_client.js** - parse prerequisites into uniform data
    - **Utils.js**  - Utilities for dynamically modifying the HTML
  + **index.html -** timetable and course builder display
  + **login.html**  - log-in and sign-up forms
* **server**
  + **db.php**  - database connection
  + **insertion\_data.php** - MySQL table rows exported to PHP statement
  + **install.php** - installation script
  + **main2.php -** main server script - parses URL parameters
  + **routes.php -** functions to serve GET requests
  + **login.php -** handles and validates login form request
  + **signup.php -** handles and validates signup form requests
* **java**
  + **login (1).java** - java client
  + **process (1).php** - server process

**6)**

Instructions to deploy:

* install XAMPP
* put project in xampp/htdocs director
* go to browser
* run localhost/course-selection-assistant/server/install.php

Instructions to modify:

* to add courses available
  + modify insertion\_data.php
* to add engineering courses
  + add data to course\_program table, the course and the program it applies to

Complementary studies electives are are entries in the course\_program table, each elective is to be entered in a row as (<elective>, <program it applies to>)

To include all Engineering programs, simply provide the option in the HTML code for that program, add the necessary courses to the course\_program table

1. The prequistive data was parsed into a JSON object for each course. For exmaple:

{

course : "SYSC 4203",

prereq: "(SYSC 3600 or SYSC 3500 or SYSC 3610) and (ELEC 2507 or ELEC 3605 or SYSC 3203) and fourth-year status in Biomedical and Electrical Engineering or fourth-year status in Biomedical and Mechanical Engineering"

}

is parse into

"{

"courses": [

[

"SYSC 3600",

"SYSC 3500",

"SYSC 3610"

],

[

"ELEC 2507",

"ELEC 3605",

"SYSC 3203"

]

],

"yearStatus": 4

}

Therefore the student must take one of "SYSC 3600", "SYSC 3500", "SYSC 3610"

and one of "ELEC 2507", "ELEC 3605", "SYSC 3203"

as well as be in 4th year Engineering status

The function:

function canTakeCourse(course){

if(!courseMap[course]){

return true;

}

console.log(course);

//student has sufficient year status

var courseObj = courseMap[course];

if(courseObj.yearStatus !== undefined){

if(courseObj.yearStatus > getUserYearStatus()){

return false;

}

}

if( !isOfferedNextSemester(course) ){

return false;

}

if( !courseHasLectureSpace(course) ){

return false;

}

if( !userHasTakenPrereqs(course) ){

return false;

}

return true;

}

checks the the user has one prerequisite from each group of optional prerequisites, There is a hardcoded exception for course concurrency. If a student changes programs, they might have taken different courses which still fall in the group of prerequisites.

**8)**

The solutions calls a function getRegisterableCourses() which returns a list of courses to which the user is able to register. This is based on prerequisite courses, year status, students program, and course capacity.

That list is then displayed to the user. When a user clicks on a course, the program calls getAllSectionsOfCourse(course) which does as its name says and returns a list of all sections of that course, which are displayed with their capacity.

When the student hovers over a course its schedule is displayed on the timetable. When clicked, The course persists on the timetable and a register button appears, which allows the student to register for the course.

In the browser client, the solutions calls a function getRegisterableCourses() which returns a list of courses to which the user is able to register. This is based on prerequisite courses, year status, students program, and course capacity. Conflicts in the timetable are highlighted in red, allowing the user to avoid them and register for another section or another course entirely.