

# CPSC 471 FINAL REPORT

---

Timothy Mealey, Ben Roberts, Cory Jensen, Scott Saunders

April 10, 2017

## Abstract

The current process for creating a class schedule is to go through many periferal websites to acquire all the information needed to make a logical decision in semester schedule building. This process is distrupted with the introduction of the Dauwtrappen Website, which allows you to get all the required information, and create a schedule, in one internet browser tab.

## Introduction

### Problem

The existing tools for planning semester schedules are decentralized, making them difficult to use together. When building schedules with these tools, it is common to have more than four tabs open in one's internet browser, which is an unnecessarily complicated state for such a simple task. The goal with this project was to simplify the semester planning process by creating a single tool that encompassed the functionality of the UofC calendar, myUofC course search, degree navigator, and the UofC schedule builder.

### Implementation

A course search feature was built which displays course names, descriptions, prerequisites, and whether they are offered in any upcoming or current semesters. It is easily navigable using a searchable sidebar tree organized by faculty, department, and course number - selecting any of which loads that node in the main page. Selecting a node also updates the browser URL to a shareable link, without reloading the page, so that students can easily share their course planning with each other.

Similar to the courses page, the schedule page contains a sidebar tree from which one can choose courses. Differing from the courses page, this sidebar is organized by semester, using a dropdown menu, before dissecting into the different course branches. When one selects a course from the tree, its lectures, tutorials, and labs are added to a sidebar on the right. These sections can then be added to the calendar, located in the centre of the page, from this right side bar. After adding the desired sections to the calendar, the user can save their progress as a schedule.

A login system was started, from which users can be added to the database using an email and password. If a user exists in the database, they can log in using their unique credentials.

## Design

### Users

1. Course Browsing

A user interested in courses offered by the University of Calgary can navigate to the Dauwtrappen website. Once on the website, a series of SQL queries are executed which load the entire course list. These queries are executed in alphabetical order, by faculty, so that the user can begin interacting with the system before all of the data is loaded. Once some or all of the course data is loaded, the user can either browse through the courses using the course tree or perform a search for preferred courses by department, course numbers, and course names. Courses that fulfill their selection/search criteria display the corresponding course information such as their name, number, description, prerequisites, and when/whether they are offered.

A user would use the Dauwtrappen website, as opposed to the other course viewing websites, for the following three reasons: Firstly, the intuitive search tool. Secondly, the Dauwtrappen website has information about when courses are offered alongside the course descriptions. Thirdly, the overall integration of all aspects of the Dauwtrappen website, namely the schedule builder alongside the course viewer.

## 2. Schedule Building

A student, or perspective student, can navigate to the schedule building page of the Dauwtrappen website to get organized for the upcoming semester. Again, a series of SQL queries are executed, loading the course data in a cascading fashion. The user can browse through the list or use the same search tool that is present in the course selection. The user can then click on a course to add it to the selected courses sidebar, located on the right side of the screen. This sidebar will then display the various lectures, tutorials, and labs available for that course, as well as their corresponding times. The user can individually click on these sections to add them to their schedule. To remove a selected course, the user can click the X button on that course's section in the sidebar. This will remove it from the selected course sidebar and from the schedule itself. Once the user completes their schedule, they can save it to the database by clicking the save button. They can give it a name if they wish, but this is not required. They may also load schedules from the database by pressing the load button and selecting the schedule they wish to load.

Currently the user login system is not session based, so all schedules are accessible by all users of the system.

## 3. Login System

A user who wishes to login to the website navigates to the login page. Once there, if they already have an account, they can enter their email and password, and then press the login button. This executes an SQL query to confirm that the correct information has been entered. If they do not have an account, they can enter their desired email and password, and then press the create account button. This will create an entry in the database table for that user, enabling them to login to the system.

The website does not currently have sessions, so logging in does not carry over to the courses or schedule page.

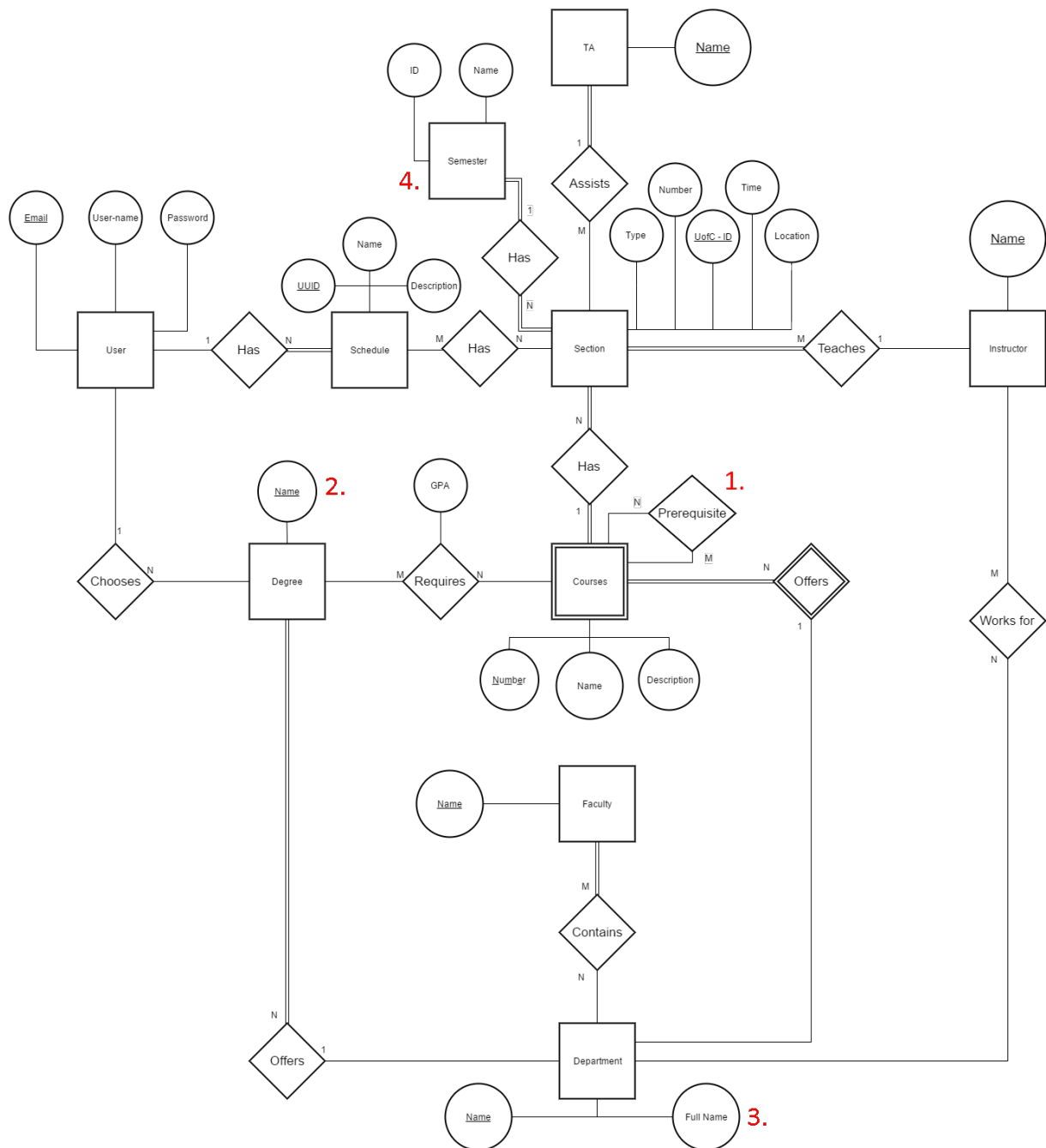


Figure 1: Entity relationship diagram.

## Entity Relationship Diagram

The red numbers on the diagram mark the changes from the initially presented diagram, and correspond to the list below.

1. A prerequisite relation from Course to Course was added, as certain courses are needed before enrollment in other courses is allowed.
2. Removed the “Course Requirements” attribute from Degree because it was redundant. It’s function is handled by the Requires relation from Degree to Course.
3. Added the “Full Name” attribute to Department because departments have a short code-name as well as a full English name. For example, “CPSC” and “Computer Science.”

4. A Semester entity type was added, and the Has relation from Semester to Section. This is needed to indicate when courses are offered.

## Implementation

### Relational Schema Diagram

Below you will find the relational schema diagram, which was created using the algorithm, described in class, that converts ERDs to RSDs. There is a tuple for each entity type and relationship from the ER diagram, with arrows pointing from foreign keys to the primary keys they reference.

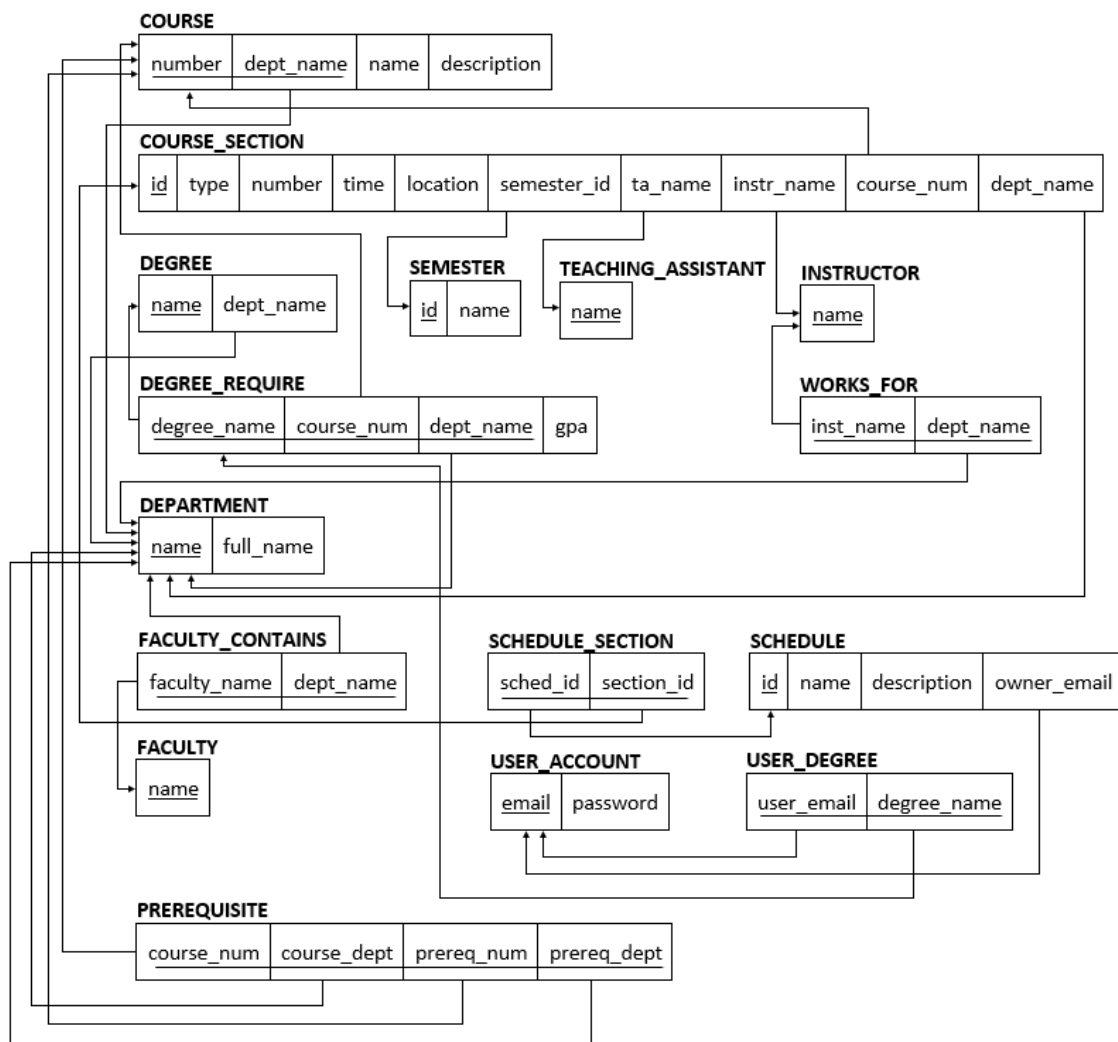


Figure 2: Relational schema diagram.

During the conversion from the ERD to the RSD, redundancies were found and required information was missing, leading to the changes that are described above the diagrams. There were also some constraints put on us by the DBMS that was used. For example, the “User” entity was renamed to “user\_account” because “user” was a reserved keyword.

## Database Management System

PostgreSQL is the DBMS used because it is a full-featured SQL implementation with good support and documentation. Here are the SQL queries that were written:

Checks if a user exists in the database with a given email and password:

```
1 SELECT email , password
2 FROM user_account
3 WHERE email=<EMAIL> AND password=<PASSWORD>
```

Checks if a user exists in the database with a given email:

```
1 SELECT email
2 FROM user_account
3 WHERE email=<EMAIL>
```

Adds a user account to the database with an email and password:

```
1 INSERT INTO user_account
2 VALUES (<EMAIL> , <PASSWORD>)
```

Get a list of all the semesters and the faculties they contain:

```
1 SELECT s.*, f.name AS fac_name
2 FROM faculty AS f ,
3     faculty_contains AS fc ,
4     course_section AS c ,
5     semester AS s
6 WHERE c.semester_id=s.id
7     AND c.dept_name=fc.dept_name
8     AND f.name=fc.faculty_name
9 GROUP BY s.id , f.name
```

Gets a list of all the semesters:

```
1 SELECT s.* FROM semester AS s
```

Gets a list of all the course sections from a faculty in a semester:

```
1 SELECT DISTINCT s.*, c.*,
2     fc.faculty_name AS fac_name ,
3     d.full_name AS dept_full_name
4 FROM course AS c ,
5     faculty_contains AS fc ,
6     course_section AS s ,
7     department AS d
8 WHERE s.semester_id=<SEMESTER_ID>
9     AND s.dept_name=fc.dept_name
10    AND c.number=s.course_num
11    AND c.dept_name=s.dept_name
12    AND fc.faculty_name=<FACULTY_NAME>
13    AND d.name=s.dept_name
```

Gets a list of all the courses and course sections from a faculty with their prerequisites:

```
1 SELECT DISTINCT c.*, s.*,
2   c.number AS course_num,
3   c.dept_name AS dept_name
4 FROM (SELECT DISTINCT c.*,
5   fc.faculty_name AS fac_name,
6   d.full_name AS dept_full_name
7 FROM faculty_contains AS fc,
8   department AS d,
9   (SELECT DISTINCT c.*,
10    p.prereq_num,
11    p.prereq_dept
12 FROM course AS c
13 LEFT JOIN prerequisite AS p
14 ON p.course_num=c.number
15 AND p.course_dept=c.dept_name
16 ) AS c
17 CROSS JOIN course_section AS s
18 WHERE d.name=c.dept_name
19 AND d.name=fc.dept_name
20 AND fc.faculty_name=<FACULTY_NAME>
21 AND c.dept_name=s.dept_name
22 ) AS c
23 LEFT JOIN course_section AS s
24 ON c.number=s.course_num
25 AND c.dept_name=s.dept_name
```

Gets the list of sections contained in a schedule:

```
1 SELECT s.*, c.*, cs.*
2 FROM schedule AS s,
3   course AS c,
4   course_section AS cs,
5   schedule_section AS ss
6 WHERE s.id=<SCHEDULE_ID>
7 AND ss.sched_id=s.id
8 AND cs.id=ss.section_id
9 AND c.number=cs.course_num
10 AND c.dept_name=cs.dept_name
```

Saves a schedule with an ID, name, description, and user email:

```
1 INSERT INTO schedule
2 VALUES (<SCHEDULE_ID>,<NAME>,<DESCRIPTION>,<USER_EMAIL>)
```

Adds a section to a schedule:

```
1 INSERT INTO schedule_section
2 VALUES (<SCHEDULE_ID>,<SECTION_ID>)
```

There are many more INSERT queries in the various scrapers and test-data scripts that populate the database with course information. These have been left out as they do not pertain to transactions within the site.

## User Interface

Every page is easy to navigate to, with there being only a small amount of pages to go to for users to create a schedule, gather information or login/create an account. The buttons at the top right of the screen are used to access each of these pages.

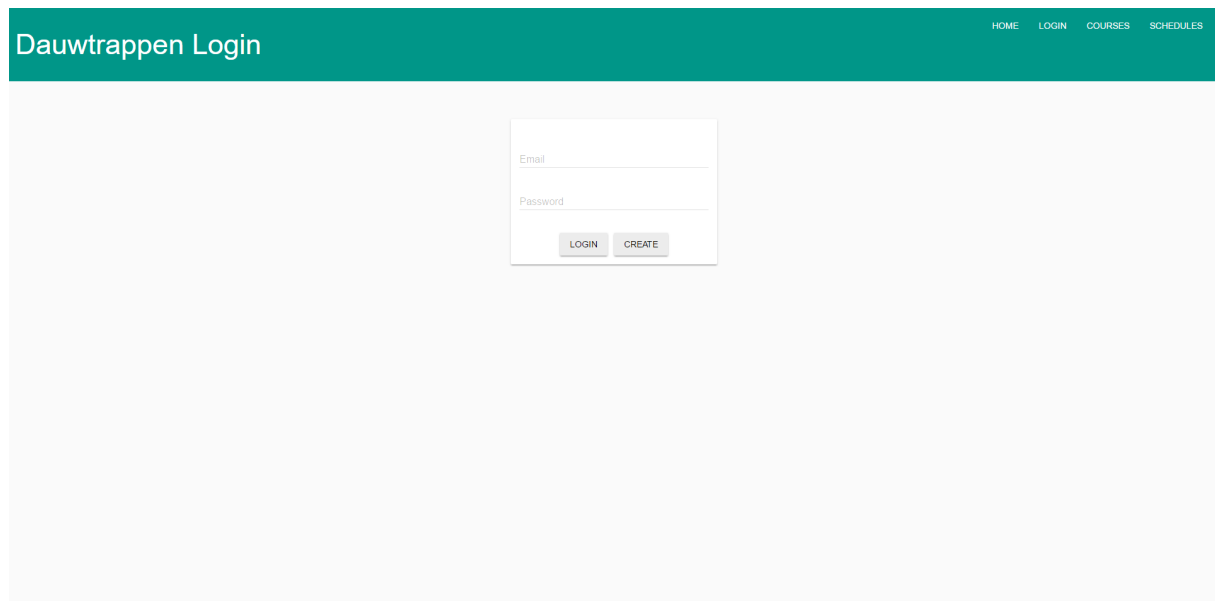


Figure 3: Login page.

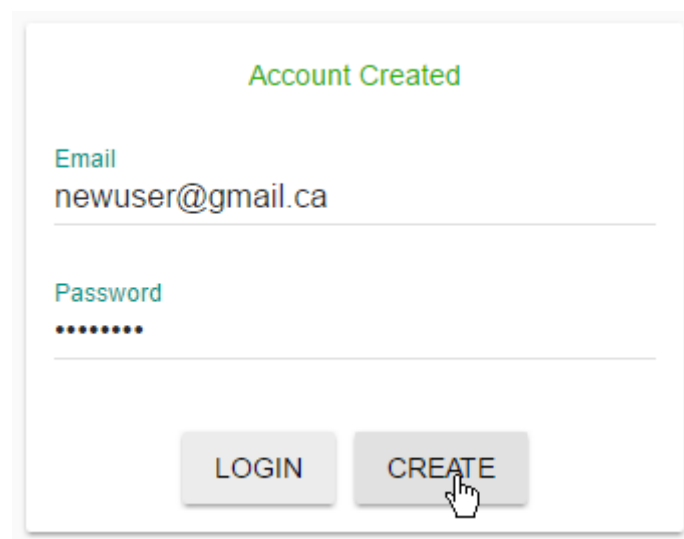
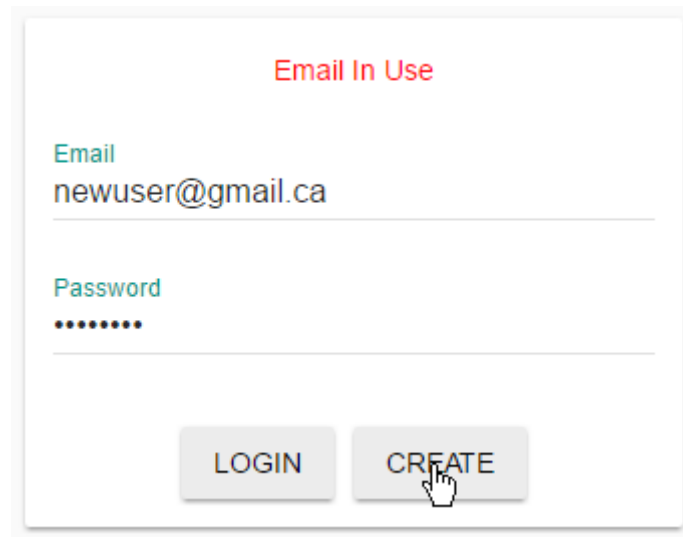
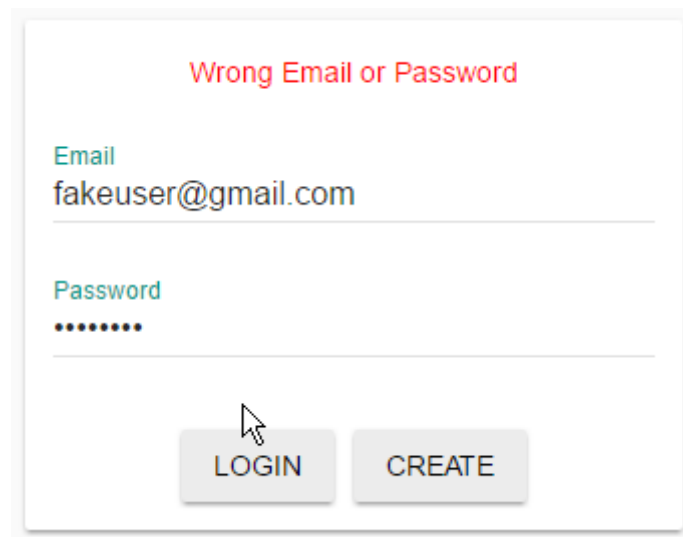


Figure 4: Clicked create with email not in the database.



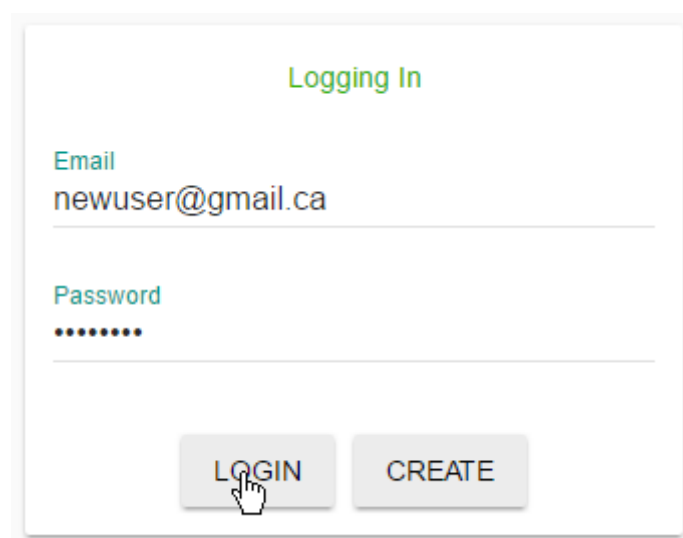
This screenshot shows a login form with a red error message at the top that reads "Email In Use". Below the message, the "Email" field contains the text "newuser@gmail.ca" and the "Password" field is filled with seven dots. At the bottom of the form, there are two buttons: "LOGIN" and "CREATE". A mouse cursor is positioned over the "CREATE" button.

Figure 5: Clicked create with email in the database.



This screenshot shows a login form with a red error message at the top that reads "Wrong Email or Password". Below the message, the "Email" field contains the text "fakeuser@gmail.com" and the "Password" field is filled with seven dots. At the bottom of the form, there are two buttons: "LOGIN" and "CREATE". A mouse cursor is positioned over the "LOGIN" button.

Figure 6: Clicked login with email/password pair not in the database.



This screenshot shows a login form with a green message at the top that reads "Logging In". Below the message, the "Email" field contains the text "newuser@gmail.ca" and the "Password" field is filled with seven dots. At the bottom of the form, there are two buttons: "LOGIN" and "CREATE". A mouse cursor is positioned over the "LOGIN" button.

Figure 7: Clicked login with email/password pair in the database.



Dauwtrappen Course Search
HOME
LOGIN
COURSES
SCHEDULES

Search...

A

Archaeology
Art
Art History
Arts
Accounting
Actuarial Science
Astrophysics
Anthropology
Arabic Language and Muslim Cultures
Architectural Studies
Applied Mathematics
300
400
500
600

B

Biochemistry
300
400
500
600
700

Business and Environment

Biology

200
300
400

Home > B

## B

### Biochemistry

#### 300 Level Courses

##### 341: Biochemistry of Life Processes

Emphasis is placed on describing the chemistry of biochemical molecules including proteins, carbohydrates, lipids, and nucleic acids, and how this relates to cell structure and life processes. Basic concepts of metabolism are introduced, focusing on the breakdown of carbohydrates for energy. The laboratory component reinforces learning of the lecture material, while teaching technical skills and the analysis and interpretation of experiments involving biochemical molecules.

**Prerequisites:** CHEM 351

**Offered in:** Fall 2017, Winter 2018

##### 393: Introduction to Biochemistry

The structure and function of carbohydrates, amino acids, proteins, lipids, coenzymes and enzymes will be presented, along with an introduction to metabolism and energy transduction. Laboratory: Overview of current biochemical techniques for studying proteins, enzymes and metabolic pathways.

**Prerequisites:** BIOL 311, CHEM 351

**Offered in:** Winter 2018

#### 400 Level Courses

##### 401: Biochemistry Lab Techniques I

Figure 8: Course search page. Displays course names, descriptions, prerequisites, and semesters offered.

Dauwtrappen Schedule Builder
HOME
LOGIN
COURSES
SCHEDULES

Fall 2016

C

CPSC
400
471: Data Base Management Systems

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
7:00 AM							
8:00 AM							
9:00 AM							
10:00 AM							
11:00 AM							
12:00 PM							
1:00 PM							
2:00 PM							
3:00 PM							
4:00 PM							
5:00 PM							
6:00 PM							
7:00 PM							
8:00 PM							

SAVE
LOAD

Select some courses to get started!

Figure 9: Schedule builder page. Demonstrating sidebar search feature.

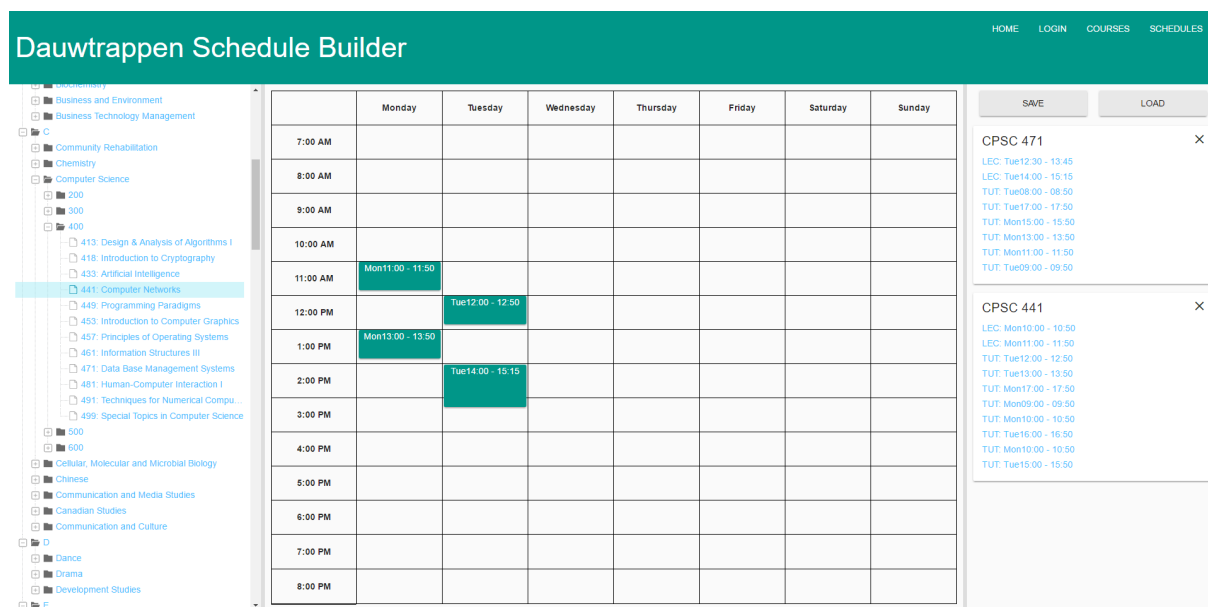


Figure 10: Building a schedule. Click courses in the left sidebar to display their sections in the right sidebar. Click the sections links to add them to the calendar view.

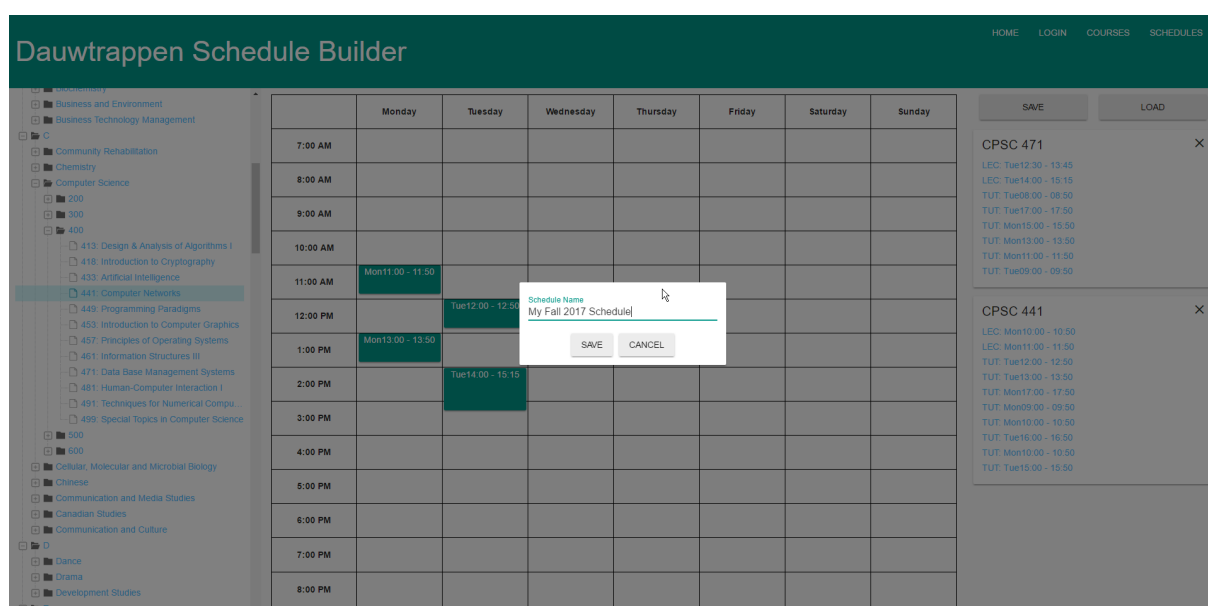


Figure 11: Clicked the save button. Enter a name then click save again to save the schedule to the database.