

# CoursePlanner

Project Details

Jaspal Sandhu  
4-16-2014

## Overview

I completed my final project for CMPT 213 – Object Oriented Design in Java at Simon Fraser University using Java, Swing, Eclipse, and SVN. The application allows the user to select a course offered by SFU and it displays all the past offerings of the course. A user can analyze this data and predict when the class may be offered in the future. The user interface consists of five panels: Course Filter panel, Course List panel, Course Offerings Panel, Statistics Panel, and the Details Panel.

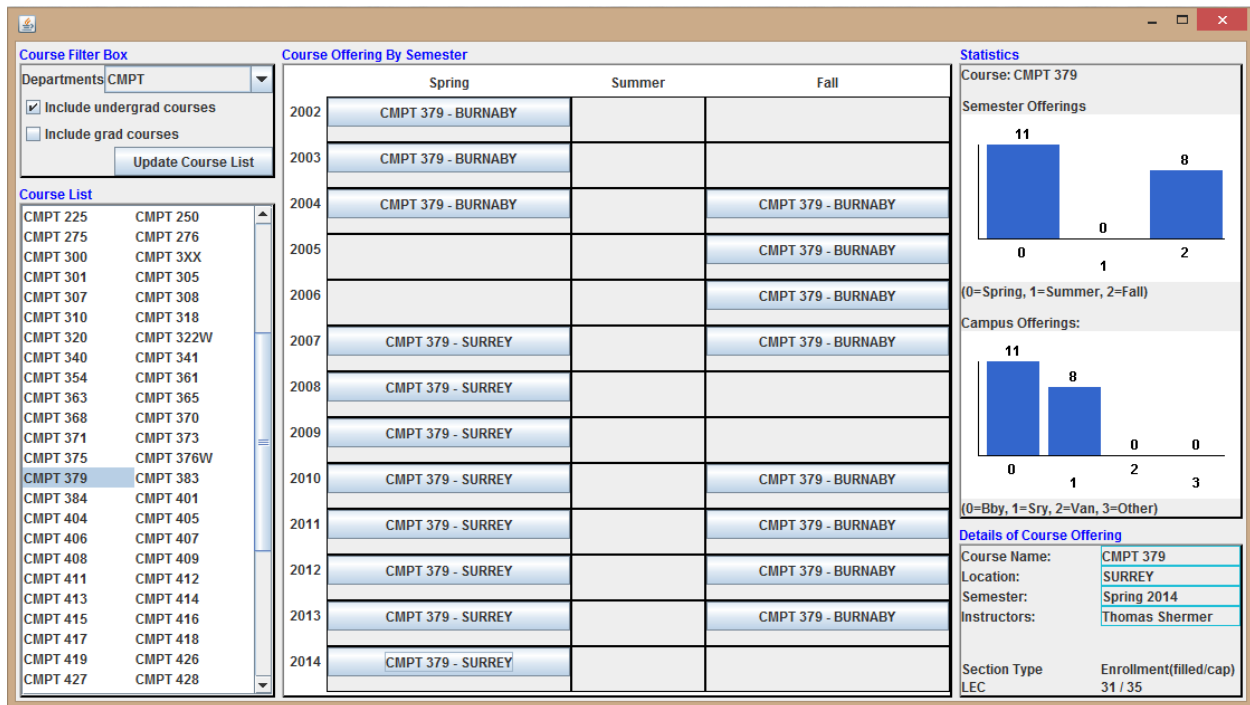


Figure 1: A course offering from Spring 2014 is selected for CMPT 379.

The Course Filter panel consists of a drop down menu for selecting a department such as CMPT. The user can select options to filter by undergrad courses or grad courses. A button sets the filter and updates the course list.

The Course List panel displays all the courses filtered by the user. The list is displayed in two columns and it supports a scrollbar if the UI cannot display all the courses. Each course is selectable and upon being selected it will show all of the course's offerings.

The Course Offering panel displays all the offerings of the selected course organized by semester, year, and campus location. Each offering is represented by a button and is displayed in a grid. When the UI is resized, other panels try to maintain their size and this panel will resize first.

The Statistics panel shows two histograms: a semester histogram and a campus histogram. The semester histogram displays how many times the selected course was offered for each semester. The campus histogram displays how many times the selected course was offered at each campus.

The Details panel displays the details of a selected course offering. So when the user selects an offering in the Course Offering panel, this panel displays additional details regarding that offering.

## Design

Development began with the design of the project by creating use cases, CRC cards, and a UML class diagram. The use cases outlined how a user would use the application. I created CRC cards to informally represent all the possible classes I might have and how they might interact with each other. The class diagram was the last step of my design process. It was a good way to formally structure my application and give me a good idea of how to begin my implementation. Included in the appendix is a representation of CRC cards and the class diagram I created.

## Implementation

Implementation was done using Java and Swing for the UI. I used Eclipse as my IDE and SVN to version control my code. Utilizing the Model-View-Controller (MVC) pattern I created two packages: one for the program model and one for the UI. The model contains all classes to maintain the state of the program and the UI classes use observers to listen to any state changes in the model and then redraw.

A CSV file, containing course information, is supplied in the following format:

```
SEMESTER, SUBJECT, CATALOG_NBR, LOCATION, ENRL_CAP, SSR_COMPONENT, ENRL_TOT, INSTRUCTORS
```

The CourseListParser begins by reading the file and parsing it to create the appropriate objects. All data is organized and structured into four objects: Subject, Course, CourseOffering, and Section. A subject contains many courses, a course contains many course offerings, and an offering contains one or more sections. Since there are many courses offered by Simon Fraser University this large amount of data has to be accessed efficiently so I used a hash table to store the courses. The CourseList object contains all the subjects, contains all the getters for the data, and it maintains a list of all observers to notify of changes for the UI. It also sets the filter, selected course, and selected course offering as selected by the user.

The user interface consists of a JFrame containing five JPanels. I created an abstract JPanel that sets a general look and feel for all the panels that will inherit from it. This prevents code duplication and allows me to easily change the look and feel of my application. All panels currently consist of a title and main content area surrounded by a border. My five main panels all inherit from my abstract JPanel. Each panel uses a Swing layout manager to place its components in an aesthetically pleasing manner and this also allows the user interface to be resized and still look good.

## After Thoughts

This project was a great learning step for me. Alongside me doing this project, the information I gathered from class lectures was extremely useful in efficiently implementing my solution. I learned about many useful patterns and I ended up utilizing the observer pattern, the strategy pattern, the composite pattern, and the decorator pattern. So far this has been the most I have worked with UI's and it was a valuable learning experience. I learned how to cleanly code UI's without letting the code turn into a mess (which is common with UI code). Through the course of the project it proved useful that I had followed the MVC pattern to separate the model and the UI. I used many different aspects of Java to create this application and my confidence in the language has grown.

# Appendix

## Screenshots

**Course Filter Box**

Departments: CMPT

☒ Include undergrad courses

☐ Include grad courses

Update Course List

**Course List**

Use a filter to select a course

**Course Offering By Semester**

Use a filter to select a course

**Statistics**

Course:

Semester Offerings

(0=Spring, 1=Summer, 2=Fall)

Campus Offerings:

(0=Bby, 1=Sry, 2=Van, 3=Other)

**Details of Course Offering**

Course Name:

Location:

Semester:

Instructors:

Figure 2: Program at start-up

**Course Filter Box**

Departments: STAT

☐ Include undergrad courses

☒ Include grad courses

Update Course List

**Course List**

STAT 602	STAT 645
STAT 650	STAT 675
STAT 685	STAT 801
STAT 802	STAT 804
STAT 805	STAT 806
STAT 811	STAT 812
STAT 830	STAT 850
STAT 851	STAT 852
STAT 853	STAT 855
STAT 870	STAT 880
STAT 881	STAT 882
STAT 883	STAT 889
STAT 890	STAT 891
STAT 894	STAT 895
STAT 898	STAT 899

Use a filter to select a course

**Course Offering By Semester**

Use a filter to select a course

**Statistics**

Course:

Semester Offerings

(0=Spring, 1=Summer, 2=Fall)

Campus Offerings:

(0=Bby, 1=Sry, 2=Van, 3=Other)

**Details of Course Offering**

Course Name:

Location:

Semester:

Instructors:

Figure 3: Filter set to grad courses only for the STAT department

## CRC Cards

### User Interface

#### CourseFilterPane

Display options for filtering courses Button filters course list	CourseFilter CourseList
---	----------------------------

#### CourseListPane

Display list of filtered courses Courses are selectable	CourseList
--	------------

#### CourseOfferingsPane

Display selected course by semester Course offerings are selectable	Course
--	--------

#### StatisticsPane

Display histogram of semester offerings Display histogram of campus offerings	Histogram Course
--	---------------------

#### CourseDetailsPane

Display selected course offering's details	Course
--	--------

## Model

### Histogram

Stores bars for a histogram Bars contain number of elements	
--	--

### CourseFilter

Filter course list	Course
--------------------	--------

### Course

Stores course details Add offerings to existing course Add sections to existing offerings	
---	--

### CourseList

Stores departments Manages observers	CourseFilter Department
---	----------------------------

### Department

Stores all courses for department	Course
-----------------------------------	--------

### CourseListParser

Reads in file Parses file into courses Sums up all details about a course	Course
---	--------

## UML Class Diagram

