CPSC 101
Sam Kelly
Scott Howes
Erik Searle
Rui Song

Double Double: Final Project Design Document

03/15/2016 V1.1

Table of Contents

Introduction:

The problem posed is to create a program that can be used to view and find course scheduling conflicts in a university course schedule.

Approach:

We approached this problem by determining what objects and classes we would need to read, organize, and display the information from the course scheduling file.

The first problem we faced was how to receive and organize the scheduling data. Our solution developed into what would eventually become our MasterSchedule class, which reads a chosen .csv file, creates the objects used to organize and store the data, and passes the necessary information to the GUI.

In order to organize the file data we defined Course objects, Block objects, Meeting objects, Professor objects, and Room objects. Each of the object types can be accessed directly from the MasterSchedule, and most of the objects are created within the MasterSchedule as well.

After organizing and storing the data, the natural next step is to build a GUI to display the information. Included in our GUI is a pop-up window that allows the user to specify a file path to a new .csv file, or they can use the preset default file path. When the file path has been specified, the program creates a MasterSchedule object for the file and the main window of the program displays, allowing the user to specify a view of selected courses on a timetable model.

Nouns:

Schedule Overlays:

Schedule overlays are overhead sheets with university major program schedules printed on them. They are used to find schedule conflicts by stacking different schedules over each other to find where there are overlaps, and then it is the administration's job to decided which overlaps must be fixed by changing different course times, and which ones can stay.

- Contain a visual representation of a formatted timetable with a schedule of specific courses (often organized by year and major for a particular program, or by professor or room number)

- Are able to overlap multiple overlays to compare course times and find time conflicts with different courses

Blocks:

A block is the term used to describe the time slot taken up by a scheduled course. They can be of many different lengths, though the most common are 1 hour, 1.5 hours, and 3 hours, and can occur at different times in the day/week with most occurring between 8:00-21:00 Monday through Friday (though some courses run earlier than 8:00, or may be scheduled on a Saturday).

- Most lengths are 1h, 1.5h, and 3h, however there are also some 2h and other unusual block length types

- Most of the scheduled blocks will be within 8:00-21:00, however some blocks may run earlier

- Most blocks will occur on weekdays (Monday through Friday), however there are some blocks that may occur on Saturdays

Courses:

A course is a scheduled and defined place and subject of learning where students attend and University professors teach. There are three “types” of courses: lectures, labs, and tutorials. These courses may be linked (where a lecture may require registration in a corresponding lab, tutorial, or both; or a lab may be its own course and possible linked to its own corresponding tutorial). Courses also come with different kinds of identifying and important information, like the course number, the subject, the professor’s surname, the starting time, the duration, the days it runs, what room it is held in, start dates and end dates, and whether or not the proposed time is forced or patterned.

- Has a course type (for example, lectures, labs, tutorials, practicums, etc)
- Have a specific course number
- Have a course name
- Have a professor (identified by surname)
- Have a starting time (either pattern or forced)
- Have a block duration (either pattern or forced)
- Have a day(s) (either pattern or forced)
- Have a room number
- Have a building ID
- Have start and end dates (either pattern or forced)

Draft Schedule:

The draft version of the course schedule is contained in an excel or .xls file, which can be converted into an easier to read .csv file type. The CSV file then holds all of the information from every course currently on the draft. The draft file itself has six kinds of lines: a date header, a title, division of forced time courses and patterned time courses, column headers, forced time course lines, and patterned time course lines.

- Starts as .xls and can be converted to .csv
- Has all currently scheduled courses and their corresponding information
- Has a date header
- Has a title
- Has a header dividing pattern and forced timed courses
- Has information column headers
- Has patterned time course lines
- Has forced time course lines

MasterSchedule

Purpose: To read from the .csv file of a schedule draft and organize the course data into a format that can be used by the main program.

Attributes:

- A group of Courses
- A group of Professors
- A group of Meetings
- A group of Rooms
- A group of Blocks

Behaviours:

- Read data from a .csv file
- Organize the file data into coherent objects

Collaborations:

- GUI

GUI

Purpose: To generate the Graphical User Interface display and run the main functions of the program

Attributes:

- Timetable display
- Course Selection panel
- Window Title
- Pop-up Selection window
- Window size

Behaviours:

- Displays course scheduling data as selected by the user

Collaboration:

- None

Course

Purpose: To store groups of course meetings and course information specific to a general course (i.e. the Course ID number)

Attributes:

- A Course has a name
- A group of Meetings associated with the course

Behaviors:

- Creates and assigns Block objects to Meeting objects

Collaborations:

- MasterSchedule

Meeting:

Purpose: To store data for one specific meeting of a course, on one particular day of the week, at one particular time

Attributes:
- An assigned Professor
- A start time
- A location
- A duration
- A course ID
- A component ID

Behaviors:
- Stores all of the identifying information for a specific meeting in one place

Collaborations:
- Course
- Professor
- Room
- MasterSchedule

Professor:

Purpose: To store professor information and identify what Meetings the professor is connected to

Attributes:
- A name

Behaviors:
- Assigns and stores references to Meetings connected to a particular Professor

Collaborations:
- MasterSchedule

Room:

Purpose: To store specific room information and connect a room to all of the Meetings it is involved with

Attributes:
- A room number

Behaviors:
- Identifies and stores the Meetings connected to a specific room

Collaborations:
- MasterSchedule

Block:

Purpose: To define 30 minute partitions of scheduled time, specifically used to format course time schedules in the GUI

Attributes:
- A start time
- A day of the week

Behaviors:
- Stores a start time and a day that can be later retrieved for use

Collaborations:
- Course
- MasterSchedule

Workload Distribution:

- Weekly Sunday meetings - All
- Maintenance of Design Plan - All

Class Construction:

- Erik: MasterSchedule
- Ray: Meeting, MeetingTime
- Sam: Course
- Scott: Professor, Room

Other:

- Erik and Scott: GUI and main
- Ray: File path pop-up window
- Sam: Logo design, User Manual

*Workload assignments denote who will be taking on the leadership role on each segment of the project, not who will be doing 100% of the work. All group members are dedicated to ensuring that work is shared as evenly as possible.