



Collège LaSalle
Montréal

Project : Student Registration System (SRS):

Requirement Specification:

We have been asked to develop an automated Student Registration System (SRS). This system will enable students to register online for courses each semester, as well as track a student's progress toward completion of his or her degree.

When a student first enrolls at the university, the student uses the SRS to set forth a plan of study as to which courses he or she plans on taking to satisfy a particular degree program, and chooses a faculty advisor.

The SRS will verify whether or not the proposed plan of study satisfies the requirements of the degree that the student is seeking. Once a plan of study has been established, then, during the registration period preceding each semester, the student is able to view the schedule of classes online, and choose whichever classes he or she wishes to attend, indicating the preferred section (day of week and time of day) if the class is offered by more than one professor.

The SRS will verify whether or not the student has satisfied the necessary prerequisites for each requested course by referring to the student's online transcript of courses completed and grades received (the student may review his or her transcript online at any time).

Assuming that (a) the prerequisites for the requested course(s) are satisfied, (b) the course(s) meets one of the student's plan of study requirements, and (c) there is room available in each of the class(es), the student is enrolled in the class(es).

If (a) and (b) are satisfied, but (c) is not, the student is placed on a first-come, first-served waiting list. If a class/section that the student was previously waitlisted for becomes available (either because some other student has dropped the class or because the seating capacity for the class has been increased), the student is automatically enrolled in the waitlisted class, and an email message to that effect is sent to the student. It is the student's responsibility to drop the class if it is no longer desired; otherwise, he or she will be billed for the course.

Students may drop a class up to the end of the first week of the semester in which the class is being taught.



PART1: Implementing the classes and Interfaces (50 points)

Implement the Student Registration System based on the class diagram given below. You should decide based on the requirement specification, the following:

1. Decide on suitable types for instance variables
2. Decide on which are concrete classes, interfaces and abstract classes.
3. Decide which methods need to be overridden in inheritance hierarchies and how
4. Implement the given classes with instance variables with their data structures, behaviors, and relationships with one another necessary to fulfill these requirements and mission

Your programs should be properly commented

The class definitions (Core classes):

Course: A semester-long series of lectures, assignments, exams, etc., that all relate to a particular subject area, and which are typically associated with a particular number of credit hours; a unit of study toward a degree. For example, Beginning Objects is a required course for the Master of Science degree in Information Systems Technology.

PlanOfStudy: A list of the courses that a student intends to take to fulfill the course requirements for a particular degree.

Professor: A member of the faculty who teaches sections or advises students.

Section: The offering of a particular course during a particular semester on a particular day of the week and at a particular time of day (for example, course Beginning Objects as taught in the Spring 2005 semester on Mondays from 1:00 to 3:00 p.m.).

Student: A person who is currently enrolled at the university and who is eligible to register for one or more sections.

Transcript: A record of all of the courses taken to date by a particular student at this university, including which semester each course was taken in, the grade received, and the credits granted for the course, as well as a reflection of an overall total number of credits earned and the student's grade point average (GPA).

For complete set of classes, their attributes and methods, refer the class diagram.

UML is located in LEA with project specification.



PART2: Data Access (30 points)

Write a class `SRSDataAccess.java` that takes care of loading some initial data from files into the application. This class methods should be called by main initialize course catalog, faculty list and schedule of classes.

Following files are provided in LEA along with this project specification document:

1. **SoC_SP2005.dat**

Schedule of classes data file for Spring 2005

Single course description per line

Format of one line: <courseNo> TAB <sectionNo> TAB <DayOfWeek> TAB
<timeOfDay> TAB <room> TAB <roomCapacity>

2. **Faculty.dat**

List of faculty

Single professor per line

Format of line: <name> TAB <ssn> TAB <title> TAB <department>

3. **TeachingAssignments.dat**

Teaching assignments of professors

Single section assignment per line

Format of line: <professor ssn> TAB <fullSectionNo>

4. **CourseCatalog.dat**

Course lists

Single course per line

Format of line: <courseNo> TAB <courseName> TAB <credits>

5. **Prerequisites.dat**

Prerequisites for courses

Single prerequisite per course per line

Format of line: <prerequisite> TAB <course>

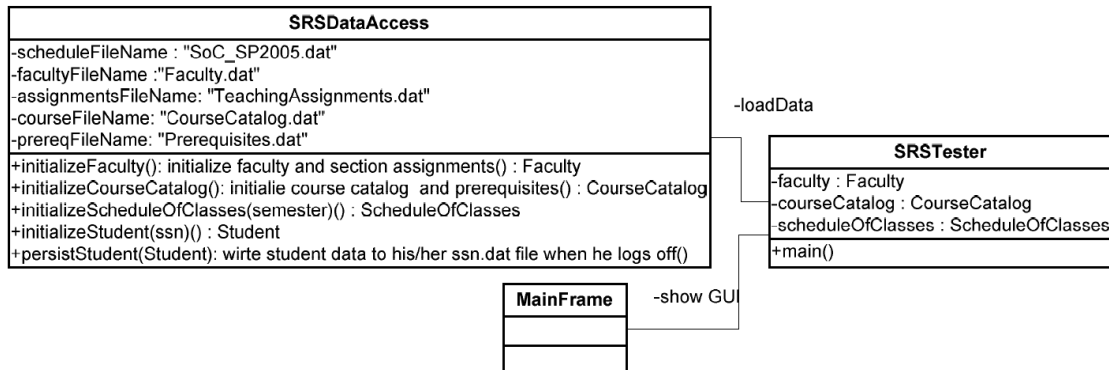
6. set of dat files each with the ssn of student and file name

111-11-1111.dat

222-22-2222.dat

333-33-3333.dat

The `initializeStudent` method of `SRSDataAccess` should use correct dat file based on input parameter ssn to the method to create a student object.



PART3: Event Driven Object Collaboration/ Console User Interface (20 points)

Your class objects should collaborate at runtime in an appropriate manner. Instantiating these classes to create the appropriate types and number of object instances Setting these objects in motion through external triggering events. To develop SRS application use console interface.

SRS console application may involve the collaboration of many different objects. Following are few different use cases that must be handled by the console you develop menu:

- log in
- log off: write updated/new data back to appropriate files.
- Allow student register for a Course
- Allow Student drop a Course
- Determine a Student's Course Load
- Choose a Faculty Advisor for a student
- Establish a Plan of Study for a student
- View the Schedule of Classes
- Allow Professor request a Student Roster for a Given Course
- Allow Professor request a Transcript for a Given Student
- Maintain Course Information (e.g., Allow change the course description, reflect a different instructor for the course, and so on)
- Determine a Student's Eligibility for Graduation
- Post Final Semester Grades for a Given Course



We may decompose any one of the use cases into steps, with each step representing a more detailed use case. For example, “Register for a Course” may be decomposed into these steps:

1. Verify that a student has met the prerequisites.
2. Check student’s plan of study to ensure that this course is required.
3. Check for availability of a seat in the course.
4. (Optionally) Place student on a wait list.

Each use case may have several scenarios involved. Use this one:

Scenario #1 for the “Register for a Course” Use Case

In this first scenario, a student by the name of Fred successfully registers for a course. The specific sequence of events is as follows:

1. Fred, a student, logs on to the SRS.
2. He views the schedule of classes for the current semester to determine which section(s) he wishes to register for.
3. Fred requests a seat in a particular section of a course entitled “Beginning Computer Concepts,” course number CMP101, section 1.
4. Fred’s plan of study is checked to ensure that the requested course is appropriate for his overall degree goals. (We assume that students are not permitted to take courses outside of their plans of study.)
5. His transcript is checked to ensure that he has satisfied all of the prerequisites for the requested course, if there are any.
6. Seating availability in the section is confirmed.
7. The section is added to Fred’s current course load.



Collège LaSalle
Montréal

From Fred's vantage point (sitting in front of a computer screen!), here's what he perceives to be occurring: after logging on to the SRS, he indicates that he wishes to register for CMP101, section 1 by choosing it from the available course list, and then

Student Registration System

SSN : 111-11-11111

Name : Fred Schnurd

Total Courses : 0

Registered for :

Option:

- 1- Choose schedule of classee
- 2- Add and save my schedule
- 3- Log off

----- Chose a schedule of classes -----

- 1- ART 101 – 1- M – 4:10 – 6:00 PM
- 2- CMP 101 – 1- M – 8:10 – 10:00 PM
- 3- CMP 101 -2 –W- 6:10 – 10:00 PM
- 4-

Chose one scheduling please: