Phong Nguyen
Loi Pham
Matthew Echert
Dylan Medina
CPSC 5011

# Homework 3: Team {{ team.name }}

On SU Online, **students** are able to <u>search</u> and <u>register</u> for **classes**. In order to add a **course**, students need to meet the **prerequisites** and the class may not be full. Students may also <u>drop</u> a class they decide not to take before the registration **deadline**. Students can only take a maximum of 18 **credits** if **undergrad** or 12 credits if **grad** student.

Students must be logged into **SU Online** before attempting to add or drop, but can search for classes either logged in or not. If the class is full, students must send an **email** to their **advisor** with their **name, SUID**, and the course they want to take; students will be put on a **waitlist** for the requested course. If **space** allows, students will be automatically enrolled in the order they are listed on the waitlist.

# Use Cases

## Student Use Cases

### <u>Search for a class</u>
1. The user enter search terms (keyword, department, times)
2. The system returns a list of matching classes along with their schedules, number of credits, prerequisites, and current availability.

### <u>Log In</u>
1. The user (student) enters a username and password
2. If the credentials are correct, the user is directed into the registration system.
3. If the credentials are incorrect, the user is provided options for password recovery.

### <u>Register for a class</u>
1. The user carries out **Log In**.
2. The user carries out **Search for a class** and selects a class from the search results.

3. If the user…
   a. ...meets all the prerequisites for the selected class, the class has availability, and would not put them over their maximum number of credits for the quarter, they are permitted to register immediately.
   b. ...meets all the prerequisites for the selected class but the class is currently full, they are offered the option to email the appropriate advisor to join the wait list and be enrolled on a space-available basis.
   c. ....does not meet all the prerequisites for the selected class, the incomplete prerequisites are displayed.

### Drop a class
1. The user carries out **Log In**.
2. The user is presented with a list of their currently registered classes.
3. The user selects one of their current classes.
4. If the registration deadline has not passed, the user has the option to drop the class.

## Administrator Use Cases

### Assign teacher to class
1. The user carries out **Log in.**
2. The user is presented with a list of faculty members.
3. The user is presented with a list of classes.
4. The user selects a faculty member for class.
5. If the faculty member has a scheduling conflict, the class will not be assigned.

### Manage Waitlists
1. The user carries out **Log in.**
2. The user is presented with a list of classes with waitlists.
3. The user moves students from waitlist to the class.
4. If the class is at capacity, the user can overload the class.

### Create Classes
1. The user carries out **Log in.**
2. The user is presented with preexisting course descriptions.
3. The user is presented with student enrollment and historical data.
4. The user creates classes based on the above information.

### Remove Classes
1. The user carries out **Log in.**
2. The user is presented with a list of classes.
3. The user can remove a class.

4. If the class has students, an email will be sent to alert them of the course cancellation.

Add/drop Students
1. The user carries out **Log in.**
2. The user is presented with a list of classes.
3. The user is presented with a list of students or can search by id.
4. The user can add or remove students enrolled in classes.
5. If the student is at maximum credits or minimum credits, the student will be alerted of the over or underload.

# Candidate List of Potential Classes

- Students: Student records are central to all registration operations. **Probably**
- Faculty: Faculty records are central to creating classes and how many classes can be offered. This can probably be handled by the class. **Reject**
- Classes: Along with student records, classes have essential attributes that need to be tracked for our purposes. **Probably**
- Administrators: The system needs to both be able to direct students to their advisors, and those advisors and other administrators need to have access to the system to manage classes and students, so there should be administrative functionality, too. **Probably**
- Users: superclass for Students and Administrators. **Probably**
- Courses: Same as Classes. **Reject**
- Registration: We will need to track each class a student registers for, along with relevant data about the registration--which quarter and which section at minimum. Maybe tests to see if the user is a student or administrator. **Probably**
- Schedule: A student's schedule for a quarter is the combination of all his/her Registrations for that quarter. But, somewhere we need to make sure that students don't register for classes in conflicting time slots, and maybe that makes sense here? **Probably**
- Location: A class depends on the availability of spaces in addition to students. **Probably**
- Wait List: Essentially, a slot on a wait list is a provisional registration, contingent upon space becoming available at a later date. This can probably be handled in Registration or Class. **Reject**

# Relationships

- Class and Students depend on Location.

- Registration depends on Student and Class
- Schedule depends on Registration, Student, and Class
- Advisors/Administrators collaborate with Student, Class, Location, and Schedule.

| Student | | User |
|---|---|---|
| • Has an SU ID<br>• Has a status (ie undergraduate, graduate) | | • Class<br>• Registration<br>• Schedule<br>• Location<br>• Advisor/Administrator |

| Administrator | | User |
|---|---|---|
| • Create/modify Classes and Locations<br>• Create/modify Registrations (including managing wait list) | | • Student<br>• Class<br>• Location<br>• Schedule |

| User | |
|---|---|
| • Has login credentials | • Student<br>• Administrator |

| Class | |
|---|---|
| • Has time/quarter/department<br>• Has a capacity | • Student<br>• Registration<br>• Schedule<br>• Advisor/Administrator |

## Registration

| | |
|---|---|
| • Manages course availability<br>• Checks if student can take a course<br>• Allows students to be added and removed. | • Class<br>• Administrator<br>• Student<br>• Schedule |

## Schedule

| | |
|---|---|
| • Maintains student schedule<br>• Checks if the schedule is full<br>• Checks if class times overlap | • Registration<br>• Student<br>• Administrator |

## Location

| | |
|---|---|
| • Manage classroom availability | • Class<br>• Administrator |