

## Registration System for Universities

By Team Moogle

Moayad Badawi #014612817

Ping Chih Wang #015137367

Jianqiang Du #017547307

Trung Nguyen #017618339

California State University Long Beach

CECS 343

## **Introduction:**

**Purpose:** A new university is in the process of opening up and ready to start its journey on educating egear students. They've asked us to create for them a program that will create them a registration system that will set up and modify courses, add professors, enroll students, grant administrative privileges, and set up and modify sub colleges along with their respective departments and majors.

**Goals:** This system will provide efficient registering for students. As well as accomplish detailed oriented task for those in administrative positions. All data will be stored and updated as it changes in database, while allowing users to see necessary information in order to register and accomplish tasks.

Request for Proposal  
CECS 343 Spring 2019

You are being asked to design and build a software system for a small university. The software must be a desktop application and written using either Eclipse with Java or Visual Studio with C++. I must be able to build and run your deliverables using a CECS lab load. I.e., the software that is installed in our computer labs. It must use a database and that database must be accessible from your executable without me having to use any software other than the lab load and what comes bundled in the image built by your IDE project. We will discuss alternatives for this in class.

Functional Requirements:

1. Allow for a University name of at most 60 characters. Must be configurable.
2. Allow for up to 10 colleges in the University.
3. College Name (60 characters)
4. College Dean (60 characters) Must be an Employee
5. Each College Contains up to 30 departments.
6. Department Name (60 characters)
7. Department Chair (60 characters) Must be an Employee
8. Each Department contains up to 50 Majors.
9. Major Name (60 characters)
10. Each Major contains up to 100 courses.
11. Course Name (60 characters)
12. Course Prerequisites – a list of courses from any college/department/major
13. Each course can have any number of sessions.
14. Each Course Session has a Semester (Fall/Spring), Building Name, Room Number in Building, Start Time (Hour / Minute), End Time (Hour/Minute) Professor, List of enrolled students
15. Each Building/Room has an occupancy limit. This is the enrollment limit for any course in that room.
16. Each Professor must be an Employee.
17. A Professor may teach any course/section of the university.
18. Each Employee has a First Name, Last Name, Middle Initial, Salary (Float)
19. Each Student has a First Name, Last Name, Middle Initial. They must be registered in order to take courses. For each semester they must have paid the tuition in order to take classes.
20. Each student may take up to 20 units per semester
21. Students must have completed all prerequisites of a course in order to enroll in a course.
22. Only university Administrators may maintain
  - a. University's Name
  - b. Employees
  - c. Colleges
  - d. Departments
  - e. Building Names, Room Numbers and Occupancies
  - f. Courses
  - g. Sessions for Courses
  - h. Assign Professors, Building Names/Room Numbers for a course/session.

- i. Assign grade to a student for a course/session.
  - j. Indicate a student has paid tuition for a given semester.
- 23. Students may only add course/sessions and drop course/sessions.
- 24. A Professor may not have any time conflicts
- 25. A Student may not have any time conflicts
- 26. A Building/Room may not have any time conflicts.
- 27. Need displays for: (Contents will be specified in class)
  - a. College/Department list
  - b. Department/Course list
  - c. Sessions
  - d. Student schedules

This is the basic system. I will be filling out more details and adding more functionality in class as part of your requirements analysis.

Use Case 1	Opening Admin Account
Actors	Super Admin
Pre-Conditions	User must be logged in as super admin New Admin's Information is present (At least First Name) Must Have Network Connection
Flow on Control	<ul style="list-style-type: none"> <li>Clicks add button</li> <li>Display input field for user</li> <li>Fill out field</li> <li>Click submit</li> <li>Validate input field</li> <li>Updates database showing new Admin with granted privilege</li> </ul>
Post-Conditions	The system now displays the new admin and grant them admin privilege. Also saves data.
Error-Condition	Input field is not validated <ul style="list-style-type: none"> <li>Empty field</li> <li>Character count exceeds max count (60)</li> </ul>
Non-Functional Requirements	Data should be processed and updated within 5 seconds.

Use Case 2	Remove Admin Account
Actors	Super Admin
Pre-Conditions	User must be logged in as super admin Must be on the admin management page Must Have Network Connection
Flow on Control	<ul style="list-style-type: none"> <li>Mark the admin that is to be removed</li> <li>Clicks admin remove button <ul style="list-style-type: none"> <li>System prompts for confirmation</li> </ul> </li> <li>User clicks confirm button <ul style="list-style-type: none"> <li>System shows admin is removed</li> </ul> </li> </ul>
Post-Conditions	Admin is removed from the database
Error-Condition	<ul style="list-style-type: none"> <li>Network connection is lost during the process.</li> </ul>

Use Case 3	Update Admin Information
Actors	Super Admin
Pre-Conditions	User must be logged in as super admin Must be on the admin management page Must Have Network Connection
Flow on Control	<ul style="list-style-type: none"> <li>User clicks on update button <ul style="list-style-type: none"> <li>System displays input fields</li> </ul> </li> <li>User modifies admin information</li> <li>User clicks submit button <ul style="list-style-type: none"> <li>System validates modifications</li> <li>System updates the database</li> </ul> </li> </ul>
Post-Conditions	Admin information in database is changed and updated
Error-Condition	Input field is not Validated <ul style="list-style-type: none"> <li>Empty field</li> <li>Character count exceeds max count (60) for admin name</li> </ul>
Non-Functional Requirements	Data should be processed and updated within 5 seconds.

Use Case 4	Add a college
Actors	admin
Pre-Conditions	Must have network connection.
Flow on Control	<ul style="list-style-type: none"> <li>User clicks add college button <ul style="list-style-type: none"> <li>System transfers user to input page</li> </ul> </li> <li>User input college name and click submit <ul style="list-style-type: none"> <li>System shows the updated data</li> </ul> </li> </ul>
Post-Conditions	Program saves and updates database
Error-Condition	<ul style="list-style-type: none"> <li>There are already 10 colleges in the university.</li> <li>College name exceed 60 characters.</li> </ul>
Non-Functional Requirements	System should process the instruction and update database within 5 seconds.

Use Case 5	Remove a college
Actors	admin
Pre-Conditions	Must be logged in as admin or higher Must be on college management page Must have network connection.
Flow on Control	<ul style="list-style-type: none"> <li>• User selects the college to be removed</li> <li>• User clicks submit button <ul style="list-style-type: none"> <li>• System prompts for confirmation</li> </ul> </li> <li>• User clicks confirm <ul style="list-style-type: none"> <li>• System shows that college is removed.</li> </ul> </li> </ul>
Post-Conditions	System removes college from the database.
Error-Condition	There are none college in the database.
Non-Functional Requirements	System should process the instruction and update database within 5 seconds.

Use Case 6	Update a college Information
Actors	admin
Pre-Conditions	Must have network connection.
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks update college button <ul style="list-style-type: none"> <li>• System transfers user to input page</li> </ul> </li> <li>• User modifies college name and clicks submit <ul style="list-style-type: none"> <li>• System shows the updated data</li> </ul> </li> </ul>
Post-Conditions	System saves and updates database
Error-Condition	College name exceed 60 characters.
Non-Functional Requirements	System should process the instruction and update database within 5 seconds.

Use Case 7	Changing School Information
Actors	Super Admin
Pre-Conditions	User is logged in
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks update button <ul style="list-style-type: none"> <li>• Show fields filled out with current data</li> </ul> </li> <li>• User changes the desired fields <ul style="list-style-type: none"> <li>• Program validates input fields</li> <li>• Program shows school information is updated</li> </ul> </li> </ul>
Post-Conditions	Program saves and updates database
Error-Condition	User has inputted into text fields more than 60 characters
Non-Functional Requirements	Data should be processed and saved in 5 seconds.



Use Case 8	Student enrolls session
Actors	<ul style="list-style-type: none"> <li>• Student</li> </ul>
Pre-Conditions	<ul style="list-style-type: none"> <li>• User status is student</li> <li>• One student_schedule page: <ul style="list-style-type: none"> <li>◦ Display student schedule which is linked to student database</li> <li>◦ One button (direct to search page)</li> </ul> </li> <li>• One select_term page: <ul style="list-style-type: none"> <li>◦ One name box</li> <li>◦ One select term table</li> <li>◦ One continue button</li> </ul> </li> <li>• One search_open_session page: <ul style="list-style-type: none"> <li>◦ One subject table</li> <li>◦ One course number box</li> <li>◦ One submit button</li> </ul> </li> <li>• One result_page: <ul style="list-style-type: none"> <li>◦ Open session table</li> <li>◦ Each session has one button for select</li> </ul> </li> </ul>
Flow on Control	<ul style="list-style-type: none"> <li>• Student click enroll button <ul style="list-style-type: none"> <li>◦ Direct to select_term page</li> <li>◦ Page shows name box and select term table</li> </ul> </li> <li>• Student input their Full Name</li> <li>• Student select term</li> <li>• Student click continue button <ul style="list-style-type: none"> <li>◦ System verifies that Full Name has paid tuition or not</li> <li>◦ System verifies that the term selected in is anytime in the future</li> <li>◦ Direct page to student schedule page</li> </ul> </li> <li>• Student click search button In the student schedule page <ul style="list-style-type: none"> <li>◦ Page direct to search criteria page that contain subject table, course number box</li> </ul> </li> <li>• Student pick subject</li> </ul>

	<ul style="list-style-type: none"> <li>● Student enter course number</li> <li>● Student click submit button <ul style="list-style-type: none"> <li>○ System check session database and use select method for showing less data.</li> <li>○ System verify those data and see if there is a data that has the same course number as user input</li> <li>○ Direct to result_page, if there is no match result, prompt nothing found,else show all datas.</li> </ul> </li> <li>● Student check all the data and see what session he want to add</li> <li>● Student click select button <ul style="list-style-type: none"> <li>○ System verify if the prerequisites class was taken, prompt failed if student hasn't take the prerequisites class, else prompt add successful</li> <li>○ System verify if the the session full or not, prompt failed if class is full.</li> <li>○ Student database update</li> <li>○ Student schedule update</li> <li>○ Session database update</li> </ul> </li> </ul>
Post-Conditions	<ul style="list-style-type: none"> <li>● Update student database after add new session</li> <li>● Student schedule will be updated as well</li> <li>● Update session information after student add class</li> </ul>
Error-Condition	<ul style="list-style-type: none"> <li>● Student has not paid tuition.</li> <li>● Student exceeds unit count.</li> <li>● Student did not provide enough information to search open session, like missing subject or course number</li> <li>● Student has not completed the prerequisites. For the class</li> </ul>
Non-Functional Requirements	<ul style="list-style-type: none"> <li>● Data is processed and save within 5 seconds</li> <li>● For all the pages in pre-condition, can add a button for return main page.</li> <li>● For student_schedule page, check if it was created, if yes, then just direct to that page, no need to create second student_schedule page</li> </ul>

Use Case 9	Student Drops Course
Actors	<ul style="list-style-type: none"> <li>• Student</li> </ul>
Pre-Conditions	<ul style="list-style-type: none"> <li>• User status is Student</li> <li>• One student_verified page <ul style="list-style-type: none"> <li>◦ Full name box</li> </ul> </li> <li>• One student_schedule page: <ul style="list-style-type: none"> <li>◦ Display student schedule which is linked to student database</li> <li>◦ Each session in the student schedule should have one drop button</li> </ul> </li> </ul>
Flow on Control	<ul style="list-style-type: none"> <li>• Student click drop class button <ul style="list-style-type: none"> <li>◦ System direct to student_verified page</li> </ul> </li> <li>• Student input full name <ul style="list-style-type: none"> <li>◦ System verified name matches in student database</li> <li>◦ If there is matches results, direct to student_schedule page, else prompt error message</li> </ul> </li> <li>• Student check session in student schedule and click the drop button on the session student want to drop <ul style="list-style-type: none"> <li>◦ System prompt drop successfully</li> <li>◦ Student database update</li> <li>◦ Student schedule update</li> <li>◦ Session database update</li> </ul> </li> </ul>
Post-Conditions	<ul style="list-style-type: none"> <li>• Students are dropped from the courses</li> </ul>
Error-Condition	<ul style="list-style-type: none"> <li>• System cannot find the matches name in student database</li> </ul>
Non-Functional Requirements	<ul style="list-style-type: none"> <li>• Data is processed and save within 5 seconds</li> <li>• For student_schedule page, check if it was created before, if yes, then just direct to that page, no need to create second student_schedule page</li> <li>• When student click drop button, maybe prompt warning to make sure user really want to drop the class</li> </ul>

Use Case 10	Add a new course to the system
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	User must be logged in as Admin status or higher Must Have Network Connection
Flow on Control	<ul style="list-style-type: none"> <li>• User fills out fields (course name, department name, which major) <ul style="list-style-type: none"> <li>• System adds newly inputted information</li> <li>• System displays new added course</li> </ul> </li> </ul>
Post-Conditions	Program saves and updates database with the new course under the selected department Course number generated.
Error-Condition	Entered course name is longer than 60 characters. Course conflicts with another existing course.
Non-Functional Requirements	Data should be processed and updated within 5 seconds.

Use Case 11	Remove a course from the system
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	The course must be in the system. Must have network connection.
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks remove course button <ul style="list-style-type: none"> <li>• System shows a list of all courses</li> </ul> </li> <li>• User selects which course to remove</li> <li>• User clicks submit <ul style="list-style-type: none"> <li>• System prompts for confirmation</li> </ul> </li> <li>• User clicks confirm <ul style="list-style-type: none"> <li>• A course is successfully removed from the system</li> </ul> </li> </ul>
Post-Conditions	System updates the database and confirms that the course is removed
Error-Condition	<ul style="list-style-type: none"> <li>• No error</li> </ul>
Non-Functional Requirements	System should process the instruction and update the database within 5 seconds.

Use Case 12	Update a Course Information
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	<p>Must be on course management page.</p> <p>Must have network connection.</p>
Flow on Control	<ul style="list-style-type: none"> <li>• User selects a course to modify its information <ul style="list-style-type: none"> <li>• System shows input fields</li> </ul> </li> <li>• User modifies course information <ul style="list-style-type: none"> <li>• System validates input fields that is modified</li> <li>• System shows data is updated</li> </ul> </li> </ul>
Post-Conditions	System saves and updates database
Error-Condition	Course name is taken.
Non-Functional Requirements	System should process the instruction and update database within 5 seconds.

Use Case 13	Register a student
Actors	Student, Admin
Pre-Conditions	<p>Must be logged in with admin status or higher</p> <p>Must have network connection.</p>
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks “Register Student” Button <ul style="list-style-type: none"> <li>• Display Input field asking for First Name, Middle Initial, and Last Name. (Only First Name is required to move through the process)</li> </ul> </li> <li>• User fills out given fields.</li> <li>• User clicks next. <ul style="list-style-type: none"> <li>• System shows that student is registered successfully.</li> </ul> </li> </ul>
Post-Conditions	System updates the database and confirms that student is added. Student ID is generated for the student.
Error-Condition	<ul style="list-style-type: none"> <li>• First name is not entered.</li> <li>• Input fields is longer than 6 characters</li> </ul>

Non-Functional Requirements	System should process the instruction and update the database within 5 seconds.
-----------------------------	---

Use Case 14	Clearing student for paying tuition
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	Have the specific students information. Verification of students payment. Connected to network.
Flow on Control	<ul style="list-style-type: none"> <li>• User inputs students ID</li> <li>• Retrieves student information</li> <li>• Displays options to enroll to class, drop, and change tuition status</li> <li>• Changes tuition status to true</li> <li>• Students information is updated and saved</li> <li>• Student should be able to register for classes</li> </ul>
Post-Conditions	Students status for tuition should be true and that specific student is now able to register for classes.
Error-Condition	Invalid ID
Non-Functional Requirements	Status is updated and saved within 5 seconds.

Use Case 15	Adding Department
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	<ul style="list-style-type: none"> <li>• User must be logged in as Admin status or higher</li> <li>• Must Have Network Connection</li> </ul>
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks Add Department</li> <li>• User fills out input field</li> <li>• User selects appropriate department.</li> </ul> <ul style="list-style-type: none"> <li>• Display input field for user.</li> <li>• System ask which college this department belongs too.</li> <li>• Information is saved and database is updated.</li> </ul>
Post-Conditions	Department is saved and updated under the college declared.
Error-Condition	Department name exceeds 60 characters. Department selects a college that doesn't exist.
Non-Functional Requirements	Status is updated and saved within 5 seconds.

Use Case 16	Removing Department
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	<ul style="list-style-type: none"> <li>• User must be logged in as Admin status or higher</li> <li>• Must Have Network Connection</li> </ul>
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks Remove Department <ul style="list-style-type: none"> <li>• Prompts user to select which college the department belongs in</li> </ul> </li> <li>• User selects from college <ul style="list-style-type: none"> <li>• Prompts user to select which department to remove.</li> </ul> </li> <li>• User selects department to remove. <ul style="list-style-type: none"> <li>• Selected department is removed.</li> </ul> </li> </ul>
Post-Conditions	Department is removed and database is updated. Colleges now show updated departments.
Error-Condition	Network connection is lost during the process.
Non-Functional Requirements	Status is updated and saved within 5 seconds.



Use Case 17	Update Department
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	<ul style="list-style-type: none"> <li>• User must be logged in as Admin status or higher</li> <li>• Must Have Network Connection</li> <li>• Have the new updated department information</li> </ul>
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks Modify Department <ul style="list-style-type: none"> <li>• Prompts user to select which college the department belongs in</li> </ul> </li> <li>• User selects from college <ul style="list-style-type: none"> <li>• Prompts user to select which department to Modify.</li> </ul> </li> <li>• User selects department to Modify. <ul style="list-style-type: none"> <li>• Prompts User to change name of the department.</li> </ul> </li> <li>• User fills out field. <ul style="list-style-type: none"> <li>• Program prompts user to select which college department belongs to.</li> </ul> </li> <li>• User selects which college the new Modified department belongs to. <ul style="list-style-type: none"> <li>• Information is saved and database is updated.</li> </ul> </li> </ul>
Post-Conditions	<p>Department name exceeds 60 characters.</p> <p>Department selects a college that doesn't exist.</p>
Error-Condition	<p>Department name exceeds 60 characters.</p> <p>Department selects a college that doesn't exist.</p>
Non-Functional Requirements	Status is updated and saved within 3 seconds.

Use Case 18	Adding employee to the system
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	<ul style="list-style-type: none"> <li>• User must be logged in as Admin status or higher</li> <li>• Must Have Network Connection</li> <li>• Have new employees information</li> </ul>
Flow on Control	<ul style="list-style-type: none"> <li>• Clicks add employee button <ul style="list-style-type: none"> <li>• Display input field for user</li> </ul> </li> <li>• Fill out field</li> <li>• Click submit <ul style="list-style-type: none"> <li>• Validate input field</li> <li>• Updates database showing new Employee with granted privilege</li> </ul> </li> </ul>
Post-Conditions	The system now displays the new employee and grant them employee privilege. Also saves data.
Error-Condition	Input field is not Validated <ul style="list-style-type: none"> <li>• Empty field</li> <li>• Character count exceeds max count (60)</li> </ul>
Non-Functional Requirements	Data should be processed and updated within 5 seconds.

Use Case 19	Remove employee from the system
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	User must be logged in as super admin Must be on the admin management page Must Have Network Connection Have employee's , that is to be removed, information
Flow on Control	<ul style="list-style-type: none"> <li>• Mark the employee that is to be removed</li> <li>• Clicks employee remove button               <ul style="list-style-type: none"> <li>• System prompts for confirmation</li> </ul> </li> <li>• User clicks confirm button               <ul style="list-style-type: none"> <li>• System shows employee is removed</li> </ul> </li> </ul>
Post-Conditions	Employee is removed from the database
Error-Condition	<ul style="list-style-type: none"> <li>• Network connection is lost during the process.</li> </ul>
Non-Functional Requirements	Status is updated and saved within 3 seconds.

Use Case 20	Update employee information
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	User must be logged in as admin Must be on the admin management page Must Have Network Connection
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks on update button               <ul style="list-style-type: none"> <li>• System displays input fields</li> </ul> </li> <li>• User modifies employee information</li> <li>• User clicks submit button               <ul style="list-style-type: none"> <li>• System validates modifications</li> <li>• System updates the database</li> </ul> </li> </ul>
Post-Conditions	Employee information is updated
Error-Condition	Input field is not Validated <ul style="list-style-type: none"> <li>• Empty field</li> <li>• Character count exceeds max count (60) for admin name</li> </ul>
Non-Functional Requirements	Data should be processed and updated within 5 seconds.

Use Case 21	Add a Major to the system
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	<ul style="list-style-type: none"> <li>• User must be logged in as Admin status or higher</li> <li>• Must Have Network Connection</li> </ul>
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks Add Major</li> <li>• Prompts user to select which Department the Major belongs in</li> <li>• User selects from Departments</li> <li>• Prompts user to enter name of the Major</li> <li>• User fills out field.</li> <li>• Information is saved and database is updated with new major.</li> </ul>
Post-Conditions	Information is saved and database is updated with new major.
Error-Condition	Major Name is greater than 60 characters.
Non-Functional Requirements	Status is updated and saved within 3 seconds.

Use Case 22	Remove a Major from the system
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	<ul style="list-style-type: none"> <li>• User must be logged in as Admin status or higher</li> <li>• Must Have Network Connection</li> </ul>
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks Remove Major <ul style="list-style-type: none"> <li>• Prompts user to select which Department the Major belongs in</li> </ul> </li> <li>• User selects from Departments <ul style="list-style-type: none"> <li>• Prompts to select which major to removes</li> </ul> </li> <li>• User selects which major to remove <ul style="list-style-type: none"> <li>• Information is saved and database is updated with removed major.</li> </ul> </li> </ul>
Post-Conditions	Information is saved and database is updated with removed major.
Error-Condition	Network connection failure.
Non-Functional Requirements	Status is updated and saved within 3 seconds.

Use Case 23	Update a Major in the system
Actors	Admin status or higher.
Pre-Conditions	<ul style="list-style-type: none"> <li>• User must be logged in as Admin status or higher</li> <li>• Must Have Network Connection</li> </ul>
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks Update Major</li> <li>• Prompts user to select which department the department belongs in</li> <li>• User selects from departments</li> <li>• Prompts user to select which Major to update.</li> <li>• User selects Major.</li> <li>• Prompts User to change name of the Major.</li> <li>• User fills out field.</li> <li>• Program prompts user to select which college department belongs to.</li> <li>• User selects which college the new Modified department belongs to.</li> <li>• Information is saved and database is updated.</li> </ul>
Post-Conditions	Information is saved and database is updated with removed major.
Error-Condition	Network connection failure.
Non-Functional Requirements	Modified major is updated and saved within 3 seconds.

Use Case 24	Adding a new class(instance of a course/a session) to the system
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	<ul style="list-style-type: none"> <li>• Required database for building room</li> <li>• Required database for course</li> <li>• Required database for professor</li> </ul>
Flow on Control	<ul style="list-style-type: none"> <li>• Admin clicks button ‘open new session’ <ul style="list-style-type: none"> <li>◦ Direct page to ‘open new session page’</li> <li>◦ Prompts user to select which course.</li> </ul> </li> <li>• Admin selects which course to open <ul style="list-style-type: none"> <li>◦ Page shows year box and semester box</li> </ul> </li> <li>• Admin select year and semester in each box</li> <li>• Admin click continue button <ul style="list-style-type: none"> <li>◦ Direct page to open’ course page’</li> <li>◦ Page shows input-field for course, professor and room location</li> </ul> </li> <li>• Admin input course_ID</li> <li>• Admin input professor_ID</li> <li>• Admin input Building room ID</li> <li>• Admin click submit button <ul style="list-style-type: none"> <li>◦ System verified course_ID in this database, prompt error when no match result find in database database</li> <li>◦ System verified professor_ID, prompt error when no match result find in employee database</li> <li>◦ System verified Building room ID, prompt error when no match result find in building database</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ System prompt add successfully</li> <li>○ Session database update</li> </ul>
Post-Conditions	<ul style="list-style-type: none"> <li>● A new session for class should be available for students to enroll in.</li> </ul>
Error-Condition	<p>Invalids for input fields:</p> <ul style="list-style-type: none"> <li>● (!@#\$\$%^&amp;*()) For Professor ID, Room Number, and Course Number</li> <li>● Year selected is in the past</li> </ul> <p>Cannot find match data in database for verify course_ID, professor_ID and building room ID</p>
Non-Functional Requirements	<ul style="list-style-type: none"> <li>● New course session is added and updated and saved within 3 seconds.</li> <li>● Whenever new session created, auto set the attribute of number of people as 0.</li> </ul>



Use Case 25	Removing a class(instance of a course/a session) from the system
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	Class must be in the system. Must have network connection Have course information
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks remove class button               <ul style="list-style-type: none"> <li>• System shows a list of all classes</li> </ul> </li> <li>• User selects which class to remove</li> <li>• User clicks submit               <ul style="list-style-type: none"> <li>• System prompts for confirmation</li> </ul> </li> <li>• User clicks confirm               <ul style="list-style-type: none"> <li>• A class is successfully removed from the system</li> </ul> </li> </ul>
Post-Conditions	Class is removed from the system and database is updated.
Error-Condition	Network connection is lost.
Non-Functional Requirements	Data should be processed within 5 seconds.

Use Case 26	Updating a class(instance of a course/a session) in the system
Actors	<ul style="list-style-type: none"> <li>• Super Admin/Admin</li> </ul>
Pre-Conditions	Must have network connection
Flow on Control	<ul style="list-style-type: none"> <li>• User clicks update class button <ul style="list-style-type: none"> <li>• System shows a list of all classes</li> </ul> </li> <li>• User selects which class to update <ul style="list-style-type: none"> <li>• System displays input fields</li> </ul> </li> <li>• User updates the class information by filling empty input fields or changing existing information</li> <li>• User clicks submit button <ul style="list-style-type: none"> <li>• System confirms update</li> </ul> </li> </ul>
Post-Conditions	Class information is updated in system.
Error-Condition	<ul style="list-style-type: none"> <li>• Class name exceeds 60 characters.</li> <li>• Class conflicts with other existing class(same time, year, semester)</li> <li>• Network connection is lost.</li> </ul>
Non-Functional Requirements	System should process data within 5 seconds.

Super Admin:

- Ultimately decides School names, Major provides, Who and Has power given to everyone below him

Admin:

- Has the ability to add students to a class
- Change class times
- Add classes
- Add required classes for majors

College: Up to 10

- Has a name
- Has a dean\*
- Has up to 50 departments
  - Department name
  - Has up to 50 majors

Employees:

- Professors\*
  - has a First Name, Last Name, Middle Initial
  - Teach a class
  - Add students to given class
    - Class
      - Course Name and Course Number
      - Must have a semester (Fall/Spring)
      - Building Name
      - Room Number
      - Start Time (Hour:Minute)/ End Time (Hour:Minute)
      - Has a size limit for students
      - Cannot be in conflict with another class
      - Must meet any pre-requirements if it were to have any
- Department Chair
  - Is a professor

#### Students:

- Has a student ID
- has a First Name, Last Name, Middle
- Must be enrolled in University
- Can enroll in class
- Drop class
- Swap class
- Can have a limit of 20 unit classes

#### Use Case:

- Super Admin creates a general information of school
- Admin naming university.
- Student is able to declare their major
  - And can change their major anytime they want
  - Can take any classes without regards their major

#### - Database:

- [User] (ID, name, status: {student || employee} ) : everyone
- [Student] (ID, isTuitionPaid<Boolean>, [Classes]<Array>) : just students
- [Employee] (ID, Title{ Professor, College-Dean, D-Chair},)
- [Session](sessionID, courseName, majorName, Time, Semester, Instructor, [Students]<Array> ) : All open classes
- [Course](courseName, [Classes]<Array>)
- [Major](majorName)