PCMJ Software

E-Grade

Requirements Specification

Courtney Locke

Fall'13

Table of Contents

1. Project Drivers

1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the requirements of the "E-Grade" registration software. This software is an improvement on Monmouth University's current grading system Web Advisor, and shall increase interactivity between the student users of the system and the system itself.

This document will list the hardware and software involved in creating the system, as well as the constraints of the project and the adjacent systems that it will interact with. This specification is intended to be used as guidelines for the programmers involved in this project, as well as a binding contract with Monmouth University for funding this software

1a User Business or Background of the Project Effort

Content

Many students have expressed their frustration with the current system. PCMJ has been chosen to develop and implement a new system for Monmouth University to use to allow its students to register for classes with as little frustration as possible. The University will be using this software for its own purposes. It does not plan to resell or market the software to any other organization.

Motivation

Students complain the current registration system Web Advisor is not as efficient as it should be. It requires too many screens to look up classes, input classes into two different worksheets, view their academic audits, and finally register for their classes. The current registration system in many ways prevents a percentage of students from registering for classes on time. Difficulties with the system force some students to physically go to their advisors and have them complete their schedules for them long after their registration window has passed.

Considerations

This situation is an irritation to the University's students. Because of this, their commitment to the University is affected and that creates a problem for the whole University and any accreditors that are affiliated. This is not yet a serious problem, however it could become one if the situation escalates. Students are paying good money to attend Monmouth and they have decided that it would be a good investment to create a system that better meets the requirements of its users.

Form

The community at Monmouth University comprised of both students and faculty acknowledge the frustration in bringing students into an enjoyable experience during their registration process by using the current Web Advisor system. The advantage of the E-Grade system will bring immediate information to students and allow faculty to better analyze the registration process all in the same screen vs. having to search through the system.

1b Goals of the Project

E-Grade aims to integrate all registration tools into one window to allow students to access all their information in one place. E-Grade also aims to let students have more independence in registering for classes by not requiring an advisor to unlock them to allow them to register. This will allow for users to have a smoother registration process.

This system's reason for creation is mainly to service the students of Monmouth University by creating a more user friendly system to register for classes. An advantage of this will be a decreased amount of time advisors spend correcting schedules and lower registration-related frustration levels across campus. Students will in return be more pleased with their schedules, which is good for the university and will therefore justify their investment. We anticipate completing this by doing the following:

1. Be able to flag classes of interest to the student prior to registration, and have them appear in list format to compare and contrast them during registration
2. Integrate registration, fagged classes, and the student's academic audit into one window to allow the student to view everything they need for registration in one window.
3. The previous stem also allows for the elimination of worksheets. Having two worksheets to enter classes into can be confusing and unnecessary. This will be changed by having the student flag classes first, and be able to view them simultaneously with the registration window.
4. The elimination of the advisor lock to allow you to initially register for classes. The majority of students are responsible when it comes to choosing their classes. For the students that need the extra help, they can still go to their advisor for help but it should not be required. After the semester starts, the advisor lock should be turned back on to prevent students from adding themselves into classes after the add-drop period.
5. The elimination of e-forms, by integrating adding and dropping classes directly into the registration window.

2 Stakeholders

2a&b The Client and Customer

1. Monmouth University

   This is the financial beneficiary of the project. This stakeholder will be used to gather requirements they would like to be implemented into the system. They have a great deal of involvement in the creation of this system as they are the individuals that are requesting it.

2c Other Stakeholders

1. Domain and Usability Experts

   Experts will provide input on how to design the system based on how other systems were designed. This will allow us to make a user friendly interface for individuals with a wide range of technical skills and needs.
2. Subject Matter Experts

   These experts will give input on popular conventions used for other internet registration systems. They will also provide information on the success rates and other aspects of other systems to help create requirements for E-Grade, ensuring that this system incorporate features that other registration systems may not have.
3. Requirements Analyst

   Individual in charge of creating and gathering the requirements from the client, users, and experts in the field. This individual also tests the requirements for feasibility and aids in creating the design document.
4. Testers

   These individuals will be responsible for testing all aspects of the system to make sure there are no holes in logic and to make sure that it can handle every possible situation an individual may encounter during the registration process.

2d-g Hands on Users of the Product

Key Users:

1. Student

   Tasks: Utilize the software to register for, view, add and drop classes

   Role: This will be the primary user of E-Grade. A student will be used for requirements gathering and testing purposes of the system. The project will also require feedback on the functionality of the system from this stakeholder. This user's feedback will be highly involved in gathering the requirements and the testing of this system.

   Involvement: High
   Subject Matter Experience: Novice
   Technological Literacy: Novice

2. Advisor

   Tasks: Request to add and remove students, lock and unlock students and approve transferring in and out of classes

   Role: Advisors will be a secondary user of E-Grade. Their feedback will also be used to test certain aspects of the system. The degree of influence this user has on the creation of the system will be moderate. The required level of technological literacy that is expected for this user to operate the system will be low to moderate.

   Involvement: High
   Subject Matter Experience: Master
   Technological Literacy: Journeyman

Secondary Users:

3. Professor

   Tasks: View the class list

   Role: Professors shall have the ability to use the system to view their class lists after the registration process has ended. They will also use the system to approve transfers into the class after the registration period has ended. The degree of influence this user has on the system will be low. The required level of technological literacy that is expected for this user to operate the system will be low to moderate.

   Involvement: Moderate
   Subject Matter Experience: Journeyman
   Technological Literacy: Journeyman

4. Registrar

   Tasks: Process registration requests, monitor class list (size), view a student's audit to make sure prerequisites are met

   Role: The registrar will be using the system to oversee all registration processes. The degree of influence this user has on the system will be moderate. The required level of technological literacy that is expected for this user to operate the system will be moderate.

   Involvement: Moderate
   Subject Matter Experience: Master
   Technological Literacy: Journeyman

2h Maintenance Users and Service Technicians

1. Maintenance Operator

   The maintenance operator is responsible for maintaining the system and making sure E-Grade is running as it should be. In the event that the system encounters a problem, the maintenance operator will be expected to determine the error and correct it in a timely fashion.

2. Programmers

   Programmers are responsible for building and implementing the system. They are also responsible for troubleshooting any complications that may arise during the creation of E-Grade.

3. Operational Support

   This stakeholder will be responsible for aiding users that run into complications or confusion with the system. This user will work with the users to find a solution, or find an individual who can find one in the event that it is beyond their skill level.

4. System Administrator

   This stakeholder is in charge of updating the databases that E-Grade will create of it's users. Their jobs will include adding the users to the database, updating users, creating user, and deleting users.

Project Constraints

3 Mandated Constraints

3a Solution Constraints

Description:	This product shall be integrated onto multiple platforms
Rationale:	Users currently use a variety of devices to access their registration system, and must be provided the option to continue use on these devices
Fit Criterion:	The product shall run on Windows, iOS, and Android devices

Description:	This product shall be compatible on different browsers
Rationale:	Users currently use different browsers to complete the registration process. The product must allow users access otherwise it could prevent students from registering for classes.
Fit Criterion:	The product shall run on Google Chrome, Firefox, Safari and Internet Explorer

Description:	This product must be developed within a $50,000 budget
Rationale:	While Monmouth University Students are currently unhappy with their registration system, it is not critical that students receive another one. Monmouth University has deemed it a good investment to implement a new registration system, but only at the right price.
Fit Criterion:	E-Grade will be developed within the University's budget, all requests to go over $50,000 must be discussed

Description:	This product shall communicate with the current Datatel system
Rationale:	The university currently works with Datatel to complete all class scheduling and audit information for students. E-Grade must be able to communicate with this system and retrieve audit information and send class information.
Fit Criterion:	E-Grade shall be able to communicate with this system and retrieve audit information and send class information.

Description:	This product must be completed prior to the fall 2014 registration period
Rationale:	Students are unhappy with the current registration system and the sooner that a new one is implemented, the smoother registration will be for the next semester
Fit Criterion:	This product shall be partitioned in a way that allows its completion by fall 2014 registration

3b Implementation Environment of the Current System

See Appendix B Figure 2

3c Anticipated Workplace Environment

The product will be used in a quiet school environment. This product will be accessible to all students and faculty from any computer on campus. The locations of campus computers include offices, library, computer labs and classrooms.

4 Naming Conventions and Terminology

4a Definitions of All Terms, Including Acronyms, Used by Stake Holders Involved in the Project

- E-Grade capable of communicating with other servers such as Datatel, that other institutions currently use for registration purposes.
- Registration - all functions surrounding the act of students signing up for or removing themselves from classes
- Audit - a complete collection of a students completed courses, grades, and graduation requirements needed to aid in registration
- Clients (companies or individuals) - ones the system is being developed for, and also where all requirements will be gathered from
- Prerequisite - something that you officially must have or do before you can have or do something else

5 Relevant Facts and Assumptions

5a Relevant Facts

- Individuals will be accessing system on multiple platforms, including desktop, mobile, and tabular devices
- Users accessing the registration system will require a proper username and password combination
- A student's Audit directly relates to the registration system

5b Assumptions

- Users are of average computer literacy
- Users accessing the registration system will be one of the following: student, faculty member, or employee
- Users accessing the main system will have limited access to site content.
- All courses in the system may or may not have a prerequisite, which is a condition that restricts certain users from registering for certain course.

Functional Requirements

6. The Scope of the Work

6a. The Current Situation

See Appendix B Figure 1

6b The Context of the Work

See Appendix B Figure 3

6c. Work Partitioning

Event Name	Input and Output	Summary of BUC
Registration Office adds new student IDs to the student's record.	Freshmen student IDs (in)	New student information is added and IDs are generated
Registration Office removes graduated student's information.	Student ID record changed (In)	Information of students who have graduated is removed from the database of students enrolled.
Registration Office adds a new class session	Number of class sessions available changed (in)	More sessions of a class become available for registration.
Bursar's Office places a block on student account.	Student's registration status changed (out)	The student is temporarily blocked from registering or making schedule changes
Student changes the amount of credits being taken	Enrollment status changed (in)	The student changes from part-time to full-time or vise versa.
Class instructor is changed.	New Instructor (In)	The department chair reassigns an instructor to teach a class section.
Period of Registration closes.	User not permitted to add/drop classes. (out)	Registration officially closes after the deadline to withdraw a class is reached.
student transfers credits from another institution.	course credits added to student's record (in)	Registration office manually adds the transferred credits

7. Business Data Model and Data Dictionary

7a Data Model

See Appendix B Figures 4-8

7b Data Dictionary

Name	Data Flows Generated by	Data Flows Used by	Description/Purpose	Content
Block Reason	University DB	Registration	Prevent students not in good standing from registering	[OK | NOTOK]
Building	University DB	Registration, Search Classes	Location on campus	Building Identifier
Building Room Number	University DB	Registration, Search Classes	Room in building on campus	3 digit number
Co-requisite	Registration, Search Classes	Registration, Search Classes	Dependencies that coexist with a course	Course
Course	Classes DB	Registration, Search Classes, Add Course, Drop Course	A unit of University curriculum	Course Identifier + Course Name + Course Type + Course Level + Course Description + Course Co-requisites + Course Prerequisites + Course Resources + Course Department
Course Section	Classes DB	Registration, Search Classes, Add Course, Drop Course	An instance of a University course	Course Type + Course Code + Course Status + Course Professor + ({Course Supplemental Instructors}) + Meeting Information + Course Credits + Course Term + Course Classroom Resources + Course Level
Course Code	Classes DB	Registration, Search Classes, Add	Progressively numeric in definition	Number between 0 and 700; refer to **Course**

		Course, Drop Course		**Codes** for list
Course Type	Classes DB	Registration, Search Classes, Add Course	Acronym list from A-W	[refer to **Course Types** for list]
Course Status	Classes DB	Registration, Search Classes, Add Course, Override Registration	Determination of Course maximum occupancy	[OPEN | CLOSED]
Course Status Cap/Avail	Classes DB	Registration, Search Classes, Add Course	Set of status fields regarding the occupancy availability of a course section	Course Status + Maximum Occupancy Allowance/Currently Enrolled Students
Course Faculty	University DB	Registration, Search Classes	Professor or other University Staff who teaches a course	University Faculty Identifier
Course Supplemental Instructors	Classes DB	Registration, Search Classes	Supplemental Instructors assigned to course section	S.I. First-name + (S.I. middle-name) + S.I. last-name)
Course Meeting Information	Classes DB	Registration, Search Classes	Set of locational values pertaining to a course section	Building + Building Room Number + Days Met + Time of Day Period
Course Credits	Classes DB	Add Class, Drop Class, Registration	Course credit number measured by numbers	{legal character}
Course Level	Classes DB	Add Class, Drop Class, Registration	A classification of student level	Course Level Identifier; refer to **Course Levels** for list
Course Resources	Classes DB	Search Classes	Required books for a course	Course books required Title + Course books required ISBN + Course books optional Title + Course books optional ISBN

Course Start/End Date	Classes DB	Registration	The beginning and end of a course section	Ranged from MM/DD/YY to MM/DD/YY
Course Department	Classes DB		Course Department Identifier	{legal character}
Course Restrictions	Classes DB	Registration, Add Class	Limitations on who can enroll in course	Course Restrictions Field
Course Term	Classes DB	Registration	Time of course	YY/SEASON
Department	Classes DB, University DB		A section of the University in which courses are taught under	Department Identifier + Department Acronym + Department Name + Department Description + Department Chair + (Assistant)
Drop Course			A course removal submission between student and University	
Override Registration			Authorized pass through of restrictions or limitations preventing student from self-enrolling in a course	
Program	University DB		Program of study	Program Identifier
Prerequisite	Classes DB	Add Class, Drop Class, Registration	The prior requirements needed to take a course	Course
Registration			An enrollment submission between student and University	
Registration Block Status	University DB	Add course, Search Classes	A set of attributes defining restrictions and/or limitations on a student's ability to	Student Identifier + Block Reason + (Added by University Faculty) + Log Time

			enroll	
Search Classes			A search for course sections which will take place in a University term	
Student	University DB	Registration, Search Classes, Add Course, Override Registration	Student registration attributes	Student Identifier + Year + Major + Minor + Credits Accumulated + Credits per term + Registration Block Status + Personal Advisor
University Faculty	University DB	Registration, Search Classes, Add Course, Override Registration	University Faculty attributes	University Identifier + Department + (Sections Taught)
Year		Registration, Search Classes, Add Course, Override Registration		YYYY
Legal character				[A-Z | a-z | 0-9 | ' | - | |]

8. Scope of the Product

8a Product Boundary

See Appendix B Figure 3

8b Product Use Case Table

PUC Number	Name	Actor(s)	Inputs	Outputs
1	Search for Class	System User	Courses DB	
2	User Flags Class	System User	Courses DB	Flagged Courses
3	Add Class to Registration	System User	Courses DB	Registration info.
4	Add Student to Wait List	System User	Courses DB	Registration info.
5	Remove class from Registration	System User	Registration info.	Courses DB
6	Submit Registration	System User	Registration info.	Courses DB
7	Verify Registration Eligibility	System	University DB	Registration info.
8	Drop Class	System User	Registration info.	Courses DB
9	Remove from Wait List	System User	Registration info.	Courses DB
10	Swap Class	System User	Registration info.	Courses DB
11	Faculty opens new class section	University Faculty	Courses DB	Courses DB
12	Faculty opens class section verification	University Faculty	University DB	Courses DB
13	Override Registration	University Faculty	University DB	Registration info.
14	Verify Override Registration Eligibility	University DB	University DB	Registration info.
15	Faculty remove student from class	University Faculty	Registration info.	Courses DB

8c Individual Product Use Cases

Use Case: 1	Search for Classes
Introduction:	The Faculty has the ability to search for classes whether it be for a student, or for personal use
Precondition:	1. Faculty must be under the "View Classes" tab
Flow of Events:	1. The use case begins when the user inputs the term. Faculty may also enter the subject, course level, course number, or course section, but it is not required. 2. User selects "Search" 3. System searches "Course Database" for matching criteria 4. System displays courses on screen 5. The use case ends
Alternate Path:	1. In number 1, if the user forgets to fill in the term the system displays dialogue box "You forgot to select a term, please enter and resubmit." 2. In number 1 if the user has searched for an invalid term the System displays dialogue box "The classes have not been submitted for this term yet, please select another." 3. In number 1 if the user submits an invalid course number the system displays dialogue box "Invalid course number, please re-enter." 4. In number 1, if the user submits an invalid course section, the system displays dialogue box "Invalid course section, please enter and re-submit."
Post Conditions:	1. User is now able to flag classes and store them for registration
Special Requirements:	1. User is not able to select more than one term at a time 2. User is only able to view semesters that fall within a year of the current date

GUI: See Appedix A Figure3

Entity Relationship: See Appendix B Figure 8

Use Case: 2	User Flags Course
Introduction:	The student is able to add classes of interest by selecting the class in the "View Classes" window
Precondition:	1. Student must have entered information to search for classes and is currently in the "View Classes" window
Flow of Events:	1. The use case begins when a student clicks on a class 2. The class is selected 3. The student clicks on the flag to the left of the course name 4. The system highlights the flag to the left of the course 5. The use case ends
Alternate Path:	1. In number 3, if the class has been previously selected, the system will un-select the flag and the class will no longer appear in the "Flagged Classes" pane 2. In number 3, if the student has Flagged the maximum number of classes the system shows a dialogue box stating "You have flagged your maximum of classes. Please deselect a class."
Post Conditions	1. The course is added to the side bar in the registration window to access during registration
Special Requirements	1. The maximum number of classes a student can flag at a time is 20

	2. System must be able to support 1000 students flagging classes concurrently

GUI: See Appedix A Figure2

Use Case: 3	Adds Class to Registration
Introduction:	Student is able to organize their potential classes when registering
Precondition:	1. Student needs to be enrolled in the University 2. Student needs to be in good standing with the University 3. Student must not have any blocks on their account 4. Student must have met the prerequisites of the course
Flow of Events:	1. The use case begins when a student clicks on a class in the “Flagged Classes” pane 2. The class is selected 3. Student clicks the “Add to Registration” button 4. The class is added to the registration 5. The use case ends
Alternate Path:	1. In number 3, if the class is full the system displays warning that class is full. A student may then choose if they wish to be added to the wait list. 2. Include {Add to Wait List} 3. In number 3, if the prerequisites not met this will appear listed in red alert font underneath class description 4. In number 3, if a prerequisite is in progress, ‘in progress’ will be listed underneath along with the expected date of completion 5. In number 2 if the student has not flagged any classes to add to the registration pane, the “Flagged Classes” pane will read “To add classes to registration, please select in the ‘View Classes’ tab.” 6. In number 3, if the class conflicts with another class “Conflicts with _____” and the class name will appear in red under the class name.
Post Conditions	1. System checks for University approval 2. Student charged for class 3. Student appears on the class roster
Special Requirements:	1. The maximum amount of time for adding a class may not exceed one hour. If it exceeds an hour, sessions stored to backend, a warning is displayed 2. System must be able to support 1000 students adding classes concurrently

GUI: See Appedix A Figure1

Use Case: 4	**Add Student to Wait List** Extends {Add Class to Registration}
Introduction:	This use case is inserted after step three in Add Class to Registration in case the class is already full.
Precondition:	1. Student must have met the prerequisites of the course 2. Class for which the student is trying to register must be full.

Flow of Events:	3. The use case begins when the user selects a class in the "Flagged Classes" pane 4. The system displays class options 5. The user selects the "Add to wait-list" option 6. The system adds the class to the waitlist. 7. the system displays the updated waitlist.
Alternate Path:	1. In Step 2 The user's session times out. a. System displays the log in window b. The user enters log in information. c. The system displays the waitlist.
Post Conditions	1. Instructor is notified of a student added to the wait-list. 2. Student is added to the wait-list. 3. System sends an email to the student confirming that they have been added to the wait list.
Special Requirements:	1. Student cannot add themselves to more than 6 classes 2. System should notify the student that they have been added to the wait list. within five minutes. 3. System may not add more than five students to a wait list.

GUI: See Appedix A Figure1

Entity Relationship: See Appendix B Figure 4

Use Case: 5	Remove Class from Registration
Introduction:	Student is able to remove classes from the registration pane
Precondition:	Student must have classes added to the registration pane
Flow of Events:	1. The use case begins when a class is selected in the registration pane 2. The User Selects the "Remove" button in the registration tool bar 3. The system removes the course from the registration pane 4. The use case ends
Alternate Path:	1. In number one, if there are no classes in the registration pane, they system will display the error "No classes in registration, please add from flagged classes."
Post Conditions	The student is no longer registering for the class
Special Requirements:	3. System must be able to support 1000 students removing classes concurrently

GUI: See Appedix A Figure1

Use Case: 6	Submit Registration
Introduction:	Student is able to submit full schedule registration to web advisor system
Precondition:	1. Student needs to be enrolled in the University 2. Student needs to be active in the user 3. Student is logged in 4. Student has the class list view loaded
Flow of Events:	1. The use case begins when a student has added all wanted courses to the

	Student's Class" window 2. Student clicks "Submit Registration" option in "Student's Class" option bar 3. Student verifies full Registration 4. The use case ends
Alternate Path:	1. Student must have under 18 credits in the "Student's Class" window, or else an e-form is required to complete registration 2. In number 3, the student does not accept full registration, nothing submits to web advisor system
Post Conditions	1. System enrolls student in selected courses
Special Requirements:	

GUI: See Appedix A Figure1

Use Case: 7	**Verify Registration Eligibility.** Extends {Submit Registration}
Introduction:	This use case is inserted after the steps for Submit Registration. The System verifies the requested registration .
Precondition:	1. System user has been directed from the Register use case
Flow of Events:	1. The use case begins after the student user confirms submission for registration 2. The system checks if the amount of credits for the class coincide with student record requirement 3. The system checks if the allowed occupancy of students after submission would be valid by querying both the fire codes for assigned room and university policy 4. The system checks if the student record for the student identifier attached to the submission request has any blocks or restrictions preventing unauthorized enrollment and the use case ends 5. The system responds with results from verifications, steps 2 through 4
Alternate Path:	1. In step 2, if the selected course brings students status outside of a defined range limitation, respond with an error stating credit number invalidation and end the use case 2. In step 3, if the room occupancy would be invalid after submission, respond with an error stating occupancy limitation and end the use case 1. In step 4, Student has financial aid requirements which would invalidate their applicability from receiving support when the condition of taking less than a certain amount of credits becomes true a. System displays warning student has financial aid obligations b. System displays student's specific required amount of credits c. System displays student's future standing after they will be removed 2. In Step 4, Student would be under full time amount of credits a. System displays warning that the student is currently enrolled as a full time student and after this drop would become relisted as part time 3. In Step 4, Student would be taking too few credits per related semester a. System displays warning that the student needs to maintain an amount of credits of which after this request would invalidate their status with the university

Post Conditions	1. The verification transaction gets logged by the system
Special Requirements:	1. Whenever the server does not respond within 15 seconds, the EGrade registration system considers the server to be down and cancels the use case

Use Case: 8	Drop Class
Introduction:	Student is able to drop a class in which he or she is registered for
Precondition:	1. Student needs to be enrolled in the University 5. Student needs to be active in the system 6. Student is logged in 7. Student must be registered for a class
Flow of Events:	1. The use case begins when a course is selected in "Student's Class" window in E-Grade 2. Student selects the "Remove" option in "Student's Class" option bar 3. Student verifies drop 4. The use case ends
Alternate Path:	4. In number 3, the student does not verify the drop, leaving the student's class schedule unchanged and course no dropped
Post Conditions	4. System checks database for registered course 5. The student is removed from the course
Special Requirements:	System must allow for 100 users to be removing students concurrently

GUI: See Appedix A Figure1

Entity Relationship: See Appendix B Figure 5

Use Case:9	Remove from Waitlist
Introduction:	The Student is able to be removed from a wait list if the student no longer wishes to be wait listed for a class.
Precondition:	1. The student must be wait listed in a class
Flow of Events:	1. The use case begins when the user selects the course in the "Wait Listed" pane 2. The user selects "Remove" from the registration toolbar 3. The System prompts the user to verify the removal 4. The System removes the student from the wait list 5. End of use case
Alternate Path:	1. In number 1, if the wait list for a class is no longer valid because it is past add-drop week the system displays a dialogue box stating "The wait list is no longer valid because it is after add-drop week."
Post Conditions:	1. System removes the student from the course 2. System sends an email to the student saying that they have been removed from the wait list
Special Requirements:	1. System must allow removal from more than one class by selecting more than one class 2. System must allow for 100 Faculty to be removing students simultaneously

GUI: See Appedix A Figure1

Use Case: 10	Swap Class
Introduction:	Student is able to switch a class section when already registered for a course
Precondition:	1. Student needs to be enrolled in the University 2. Student needs to be active in the user 3. Student is logged in 4. Student has the class list view loaded
Flow of Events:	1. The use case begins when a student clicks on desired class to switch under "Student's Class" window in E-Grade 2. Student clicks "Swap" option in "Student's Class" option bar 3. Student choses desired section number 4. Student verifies swap 5. The use case ends
Alternate Path:	1. In number 3, the student does not submit the swap, it leaves the student's class schedule unchanged and course is not swapped
Post Conditions	1. System checks database for registered course
Special Requirements:	

GUI: See Appedix A Figure1

Use Case: 11	Faculty opens new class section
Introduction:	Faculty is able to create a new instance of a course in the form of a class section
Precondition:	1. Faculty member needs to be logged into the system 2. Faculty must have authorization to create a class instance 3. Faculty must be in the allotted time for creating a class section
Flow of Events:	1. The use case begins when a faculty selects to open a class in the "Open class section" link 2. The faculty inputs program of study 3. The faculty inputs course codes 4. The faculty inputs amount of credits for the class 5. The faculty inputs Professor to teach class 6. The faculty inputs room number 7. The faculty inputs time of day 8. The faculty inputs allowed occupancy of students to take the class 9. The faculty inputs required resources 10. The faculty inputs course books required 11. The faculty inputs course books optional 12. The system presents input validation results 13. The faculty selects the "Confirm request" 14. Include {Open Class Section Verification} 15. The use case ends
Alternate Path:	1.
Post Conditions	1. The system stores the input transaction to the back end 2. The system logs the session 3. The system marks the class to be added as pending

	4. The system creates entry in course catalog 5. The system create entry in administrative class list 6. The system validates that the course catalog reflects the entry in the administrative class list 7. The system changes status of class added from pending to complete 8. The system emails the department head of the inputted program of study 9. The system emails the Professor assigned by the input 10. The system emails the faculty who has used this use case 11. The system emails the secretary of the department heads
Special Requirements:	1. The maximum amount of time for adding a class may not exceed 30 minutes a. If exceeds 25 minutes, a warning is displayed stating 5 minutes until request to open a class expires

Use Case: 12	**Faculty opens class section verification**
Introduction:	System verifies the request for faculty to open a class section
Precondition:	1. Open class section use case status is pending
Flow of Events:	1. The use case begins when faculty selects the “Confirm request” from the open class section use case 2. The system checks if23 the program of study is valid 3. The system checks if the course codes are valid with university policy 4. The system checks if the course codes are valid with other input parameters of amount of credits for the class, room number, time of day, allowed occupancy of students to take the class 5. The system checks if the amount of credits for the class are valid 6. The system checks if the Professor is appropriately assigned 7. The system checks if the Professor has scheduling conflictions 8. The system checks if the room number is valid 9. The system checks if the room number is preoccupied 10. The system checks if the time of day is valid 11. The system checks if the allowed occupancy of students to take the class is valid by querying both the fire codes for assigned room and university policy 12. The system checks if the student’s required resources are valid by querying the university book store 13. The system checks if the room’s required resources are available in inputted room number 14. The use case ends
Alternate Path:	1. In number 4, if the inputted course codes conflict an error is returned defining invalid fields 2. In number 7, if the assigned Professor to the open class section request has scheduling conflicts an error is returned showing their preoccupied conflict 3. In number 9, if the inputted room number is already assigned an error is returned showing class details 4. In number 12, if the inputted course codes conflict a warning is returned stating the need to correspond with the university book store 5. In number 13, if the desired room’s resources do not meet the input request an

	error is returned stating that the rooms resources status
Post Conditions	1. System returns verification for open class section input fields
Special Requirements:	1. The maximum amount of time to verify open class section request is 15 seconds

Use Case: 13	**Override Registration**
Introduction:	University Staff overrides block for student's request to add course section to their term enrollment
Precondition:	1. Register use case status is pending due to a restriction
Flow of Events:	1. The use case begins when university faculty selects the "Override Registration" from the faculty view for EGrade 2. System user chooses student identifier 3. System user chooses term 4. System user chooses course name and course section 5. The system responds information of the course section as is prior the use case 6. The system checks if the student identified by their identifier is eligible to enroll in the course and returns restrictions and/or blocks which would prevent them from doing so **Include {Verify Override Registration Eligibility}** 7. System user confirms override request by choosing university faculty digital signature 8. The system enrolls student in inputted course section from step 3 9. The system adds student to course section 10. The system updates student's record in the University Database
Alternate Path:	1. In step 3, if the input for student identifier is invalid, respond with an error 2. In step 4, if the input for semester term is invalid, respond with an error 3. In step 5, if the course name and section are not correctly associated, respond with an error 4. In step 8, if the system user denies the override request end the use case
Post Conditions	1. The system logs the transaction in the University Database 2. The system emails the course section associated department head 3. The system emails the Registrar with a receipt of the transaction
Special Requirements:	1. The maximum amount of time for the system user to complete the use case is 15 minutes

Entity Relationship: See Appendix B Figure 6

Use Case: 14	**Verify Override Registration Eligibility.** Extends {Add Student to Class}
Introduction:	System verifies the requested override registration
Precondition:	1. System user has been directed from the Override Registration use cases
Flow of Events:	1. The use case begins after the student or university faculty system user confirms submission for registration 2. The system checks if the amount of credits for the class coincide with student

	record requirement 3. The system checks if the allowed occupancy of students after submission would be valid by querying both the fire codes for assigned room and university policy 4. The system checks if the student record for the student identifier attached to the submission request has any blocks or restrictions preventing unauthorized enrollment and the use case ends 5. The system responds with results from verifications, steps 2 through 4
Alternate Path:	1. In step 2, if the selected course brings students status outside of a defined range limitation, respond with a warning of consequences 2. In step 3, if the room occupancy would be invalid after submission, respond with a warning of consequences 3. In step 4, Student has financial aid requirements which would invalidate their applicability from receiving support when the condition of taking less than a certain amount of credits becomes true a. System displays warning student has financial aid obligations b. System displays student's specific required amount of credits c. System displays student's future standing after they will be removed d. University Staff system user fills out note field with reason detailing any notes related to change e. University Staff system user signs off on change by their digital signature 4. In Step 4, Student would be under full time amount of credits a. System displays warning that the student is currently enrolled as a full time student and after this drop would become relisted as part time b. University Staff system user fills out note field with reason detailing any notes related to change c. University Staff signs off on change by their digital signature 5. In Step 4, Student would be taking too few credits per related semester a. System displays warning that the student needs to maintain an amount of credits of which after this request would invalidate their status with the university b. University Staff system user fills out note field with reason detailing any notes related to change 6. University Staff system user signs off on change by their digital signature
Post Conditions	1. The verification transaction gets logged by the system
Special Requirements:	1. Whenever the server does not respond within 15 seconds, the EGrade registration system considers the server to be down and cancels the use case

Use Case: 15	Faculty remove student from class
Introduction:	Faculty is able to remove a student from a scheduled course
Precondition:	1. Student needs to be enrolled in the University 2. Student needs to be active in the user 3. Faculty is logged in 4. Faculty has the class roster view loaded
Flow of Events:	1. Faculty has highlighted the student to remove from the class 2. Faculty selects to drop student

	3. Faculty fills out administrative fields 4. Faculty submits change to the class roster
Alternate Path:	1. Student has financial aid requirements which would invalidate their applicability from receiving support when the condition of taking less than a certain amount of credits becomes true a. System displays warning student has financial aid obligations b. System displays student's specific required amount of credits c. System displays student's future standing after they will be removed d. Faculty fills out note field with reason detailing any notes related to change e. Faculty signs off on change by their digital signature 2. Student would be under full time amount of credits a. System displays warning that the student is currently enrolled as a full time student and after this drop would become relisted as part time b. Faculty fills out note field with reason detailing any notes related to change c. Faculty signs off on change by their digital signature 3. Student would be taking too few credits per related semester a. System displays warning that the student needs to maintain an amount of credits of which after this request would invalidate their status with the university b. Faculty fills out note field with reason detailing any notes related to change c. Faculty signs off on change by their digital signature
Post Conditions	1. System logs transaction 2. Student balance gets credited for student's cost of taking the course 3. Student disappears from the class roster 4. System emails involved parties with at minimum the Faculty member and the student
Special Requirements:	1. The maximum amount of time for dropping a student from a class may not exceed 15 minutes a. If exceeds 15 minutes, a warning is displayed, and continuation must start fresh to proceed with any future attempts

Requirements

9 Nonfunctional Requirements

9a Look and Feel Requirements

Requirement #: 1 **Category: Nonfunctional** **Use Case #: 1**

Description: User shall be able to tell that the product belongs to Datatel's WebAdvisor
Rationale: The system shall look like it is used in a college setting. System has to share the appearance with all other WebAdvisor systems.
Fit Criterion: The WebAdvisor logo shall be visible on the registration page.
Dependencies:
Conflicts:
Supporting Materials:
History: Created 12/05/2013, by PCMJ

Requirement #: 2	Category: Nonfunctional	Use Case #: 13
Description: The product shall be recognizable to users of the university staff web site		
Rationale: A user must feel the overriding registration section of E-Grade is familiar		
Fit Criterion: The product shall be certified as complying with this year's university branding standards		
Dependencies: All requirements involving University Faculty (Override Registration, Faculty opens new class section)		
Conflicts:		
Supporting Materials: Appendix A Figure 1,2		
History: Created 11/28/13 by PCMJ		

Requirement #: 3	Category: Nonfunctional	Use Case #: 13
Description: The product shall appear simple to use		
Rationale: The product must be quicker to adapt to than the current product, so that users make fewer errors		
Fit Criterion: 65% of the intended audience will not need to consult a user manual to be able to use		

the product
Dependencies: All requirements with user interaction in the E-Grade product (Search for Classes, User Flags Course, Add Class to Registration, Add Student to Wait List, Remove Class from Registration, Submit Registration, Drop Class, Swap Class, Faculty opens new class section, Override Registration, Faculty remove student from class)
Conflicts:
Supporting Materials: Appendix A Figure 1,2
History: Created 11/28/13 by PCMJ

Requirement #: 4	Category: Nonfunctional	Use Case #: 13
Description: The product shall be intuitive		
Rationale: University Faculty users must find the new product easy and intuitive; otherwise they will not find it acceptable using it		
Fit Criterion: A University Faculty member shall be able to produce an override of registration within eight minutes of encountering the product for the first time without reference to any out-of-product help		
Dependencies: Override registration, override registration eligibility requirements		
Conflicts:		
Supporting Materials: Appendix A Figure 1,2		
History: Created 11/28/13 by PCMJ		

Requirement #: 5	Category: Nonfunctional	Use Case #: 13
Description: The product shall be in accessible by persons who have disabilities		
Rationale: The product must be able to accommodate for people with special needs		
Fit Criterion: The product shall be in compliance with the American with Disability Act		

Dependencies: All requirements with user interaction in the E-Grade product (Search for Classes, User Flags Course, Add Class to Registration, Add Student to Wait List, Remove Class from Registration, Submit Registration, Drop Class, Swap Class, Faculty opens new class section, Override Registration, Faculty remove student from class)
Conflicts:
Supporting Materials: Appendix A Figure 1,2
History: Created 11/28/13 by PCMJ

Requirement #: 6	Category: Nonfunctional	Use Case #: 2
Description: The product shall take the user no longer than a minute to discover how to flag and un-flag courses		
Rationale: A student must complete registration in a quick and manageable manner		
Fit Criterion: The course will be added to the registration pane relatively quickly		
Dependencies: The Courses DB to display the courses in the flag courses window		
Conflicts:		
Supporting Materials: Appendix A Figure 2		
History: Created 11/28/13 by J PCMJ		

Requirement #: 7	Category: Nonfunctional	Use Case #: 1
Description: User shall be able to figure out how to search for classes within five seconds of looking at the page.		
Rationale: The user shall be able to find the classes offered by section before flagging them. User shall have access to the classes being offered by term.		
Fit Criterion: User shall be able to		

Dependencies: **The user must have entered a course title.** **The user must have selected a term.** **The system has to have a clearly visible search option.**
Conflicts:
Supporting Materials:
History: Created 11/28/2013, by PCMJ

9b Performance Requirements

Requirement #: 8	**Category: Nonfunctional**	Use Case #: 13
Description: The product shall complete an override registration in under one minute		
Rationale: A university staff must complete overriding registration in a quick and manageable manner		
Fit Criterion: The system shall end the session if not completed with 15 minutes from the start time		
Dependencies: Override Registration, Verify Override Registration Eligibility)		
Conflicts:		
Supporting Materials:		
History: Created 11/28/13 by PCMJ		

Requirement #: 9	**Category: Nonfunctional**	Use Case #: 13
Description: The product must handle requests within an acceptable time		
Rationale: The server must be able to handle the traffic of a University allowing many students to register at once		
Fit Criterion: The product shall consider the server to be down and will end the session when a server does not respond within 15 seconds		
Dependencies:		

All requirements in the E-Grade product (Search for Classes, User Flags Course, Add Class to Registration, Add Student to Wait List, Remove Class from Registration, Submit Registration, Drop Class, Swap Class, Faculty opens new class section, Override Registration, Faculty remove student from class, Verify Registration Eligibility, Verify Override Registration Eligibility)
Conflicts:
Supporting Materials:
History: Created 11/28/13 by PCMJ

Requirement #: 10	Category: Nonfunctional	Use Case #: 6
Description: The product shall manage the workload amongst hosting resources		
Rationale: The product shall not overload any one resource with a majority of requests when using the product		
Fit Criterion: The product shall throttle multiple instances of registration eligibility on multiple cores on allocated server clusters		
Dependencies: All requirements		
Conflicts:		
Supporting Materials:		
History: Created 11/28/13 by PCMJ		

Requirement #: 11	Category: Nonfunctional	Use Case #: 13
Description: The product shall return the registration eligibility within an acceptable time		
Rationale: The product must not take too long verifying registration eligibility		
Fit Criterion: Every return of registration eligibility shall download in 7.5 seconds or less over a 115 KBps broadband connection, Every return of registration eligibility shall download in 4.0 seconds or less over the university intranet		
Dependencies: Override Registration, Add Class to Registration		

Conflicts:
Supporting Materials:
History: Created 11/28/13 by PCMJ

Requirement #: 12	Category: Nonfunctional	Use Case #: 6
Description: The product shall audit changes		
Rationale: The product shall ensure there is a way to back trace changes made		
Fit Criterion: The product shall retain a journal of all transactions for a business day		
Dependencies: Add Class to Registration, Add Student to Wait List, Remove Class from Registration, Submit Registration, Drop Class, Swap Class, Faculty opens new class section, Override Registration, Faculty remove student from class		
Conflicts:		
Supporting Materials:		
History: Created 11/28/13 by PCMJ		

Requirement #: 13	Category: Nonfunctional	Use Case #: 14
Description: The product shall be operational for University Faculty within work their hours		
Rationale: The product shall be able to handle all override registration and Faculty opens new class section requests		
Fit Criterion: In the first three months of operation, 98 percent of the time between 8 A.M. and 6 P.M.		
Dependencies: Override registration, Verify Override Registration, Faculty opens new class section		
Conflicts:		
Supporting Materials:		
History: Created 11/28/13 by PCMJ		

Requirement #: 14	Category: Nonfunctional	Use Case #: 14
Description: The product shall be operational for Students within university registration time periods		
Rationale: The product shall be able to handle all registration requests during the allowed time		
Fit Criterion: In the first three months of operation, 95 percent of the time between 12 A.M. and 12 A.M. (24 hours)		
Dependencies: Add Class to Registration, Add Student to Wait List, Remove Class from Registration, Submit Registration, Drop Class, Swap Class		
Conflicts:		
Supporting Materials:		
History: Created 11/28/13 by PCMJ		

Requirement #: 15	Category: Nonfunctional	Use Case #: 2
Description: The product shall take no longer than 10 seconds when registering a flag or un-flag in the system		
Rationale: A student must complete registration in a quick and manageable manner		
Fit Criterion: The selected courses shall appear in the registration window immediately after selection		
Dependencies: Requirements involving selecting courses to flag		
Conflicts:		
Supporting Materials:		
History: Created 11/28/13 by PCMJ		

Requirement #: 16	Category: Nonfunctional	Use Case #: 6
Description: The system shall time out if the user does not verify the registration in under 5 minutes		
Rationale: The system must not create an excessive amount of overhead while operating		
Fit Criterion: The system shall show a warning window that alerts the user that their session has		

timed out
Dependencies: The user is currently undergoing registration
Conflicts:
Supporting Materials:
History: Created 11/28/13 by PCMJ

10 Functional Requirements

Requirement #: 1	Category: Functional	**Use Case #: 1**
Description: User is able to find classes by searching for matching course criteria		
Rationale: A user must be able to search for specific courses without having to browse the list of all sections offered.		
Fit Criterion: The selected class is being offered during the term the user is looking for. The user must have selected a term that matches the present or future academic terms.		
Dependencies:		
Conflicts:		
Supporting Materials:		
History: Created 11/3/13 by PCMJ		

Requirement #: 2	Category: Functional	**Use Case #: 1**
Description: The user is able to tell the system to search for classes.		
Rationale: The user should have the option to look for classes by means other than browsing.		

Fit Criterion: User must have entered a term to search for user must have entered class information to search for in the selected term. User must have entered information matching, course title, course section, level, or course number.
Dependencies: Data entered in the search fields.
Conflicts:
Supporting Materials: (ER diagrams, charts, etc.)
History: Created 11/3/13 by PCMJ

Requirement #: 3	Category: Functional	**Use Case #: 1**
Description: The system searches the course database for matching classes.		
Rationale: The system should have access to the database of classes in order to provide accurate results.		
Fit Criterion: User must have entered course information to search for The system must have entered a term to search classes under.		
Dependencies:		
Conflicts:		
Supporting Materials: (ER diagrams, charts, etc.)		
History: Created 11/3/13 by PCMJ		

Requirement #: 4	Category: Functional	**Use Case #: 1**
Description: The system searches the course database for matching classes.		

Rationale: The system should have access to the database of classes in order to provide accurate results.
Fit Criterion: User must have entered course information to search for The system must have entered a term to search classes under.
Dependencies:
Conflicts:
Supporting Materials: (ER diagrams, charts, etc.)
History: Created 11/3/13 by PCMJ

Requirement #: 5	Category: Functional	**Use Case #: 1**
Description: The user is able to see the results the system displays		
Rationale: The user shall be able to see the classes offered on the term they searched for. and see the results produced by what they searched for.		
Fit Criterion: The user must have entered appropriate search criteria.		
Dependencies: Requirements dealing with “Search for Class”		
Conflicts:		
Supporting Materials: (ER diagrams, charts, etc.)		
History: Created 11/3/13 by PCMJ		

Requirement #: 1	**Category: Functional**	Use Case #: 2
Description: The System shall allow the user to flag courses they are interested in registering for.		
Rationale: To allow the user to view all courses they are interested in enrolling in in one place in the registration window		
Fit Criterion:		

The selected courses will be displayed in the "Flagged Courses" pane of the registration window
Dependencies: Requirement covering the requirement between the pane and University DB
Conflicts: Requirement prohibiting more than 20 flagged courses
Supporting Materials: Appendix A Figures 1 & 2, Appendix B Figure 3
History: Created 11/3/13 by PCMJ

Requirement #:	**Category: Functional**	Use Case #: 2
Description: The product shall cooperate with the University Database to reflect the correct information for those courses		
Rationale: To allow the user to create an accurate course schedule		
Fit Criterion: The course information in both locations will have no discrepancies		
Dependencies:		
Conflicts:		
Supporting Materials: Appendix B Figure 1		
History: Created 11/3/13 by PCMJ		

Requirement #:	**Category: Functional**	Use Case #: 2
Description: The system shall flag the course when the flag to the left of the course information is tapped once		
Rationale: The user must be able to add courses to the "Flagged Courses" pane		
Fit Criterion: The flag icon to the left of the course will be displayed in color		
Dependencies: "Un-flag Course" Requirement		
Conflicts: Requirement prohibiting more than 20 flagged courses		
Supporting Materials:		

Appendix A Figure 2
History: Created 11/3/13 by PCMJ

Requirement #:	**Category: Functional**	Use Case #: 2
Description: The System shall allow the user to flag multiple courses		
Rationale: The user must have the option to compare courses they are interested in completing		
Fit Criterion: All selected courses will appear in the "Flagged Courses" pane in the registration window		
Dependencies:		
Conflicts: Requirement prohibiting more than 20 flagged courses		
Supporting Materials: Appendix A Figure 2		
History: Created 11/3/13 by PCMJ		

Requirement #:	**Category: Functional**	Use Case #: 2
Description: The system shall un-flag the course when the flag to the left of the course information is tapped twice		
Rationale: The user must have the option to deselect a course in the event that they are no longer interested in it		
Fit Criterion: Once the course is selected, the product shall transfer the course to the "Flagged Courses" pane of the registration window.		
Dependencies: The course has been previously selected		
Conflicts:		
Supporting Materials: Appendix A Figure 2		
History: Created 11/3/13 by PCMJ		

Requirement #:	**Caegory: Functional**	Use Case #: 2

Description: The product shall not allow the user to flag more than 20 courses at the same time
Rationale: The System must not create too much overhead
Fit Criterion: The “Flagged Courses” pane will display 20 courses or less that have been selected by the user
Dependencies:
Conflicts:
Supporting Materials:
History: Created 11/3/13 by PCMJ

Requirement #: 1	Category: Functional	Use Case #: 3
Description: The system shall allow students to add a flagged class to the registration pane.		
Rationale: To allow the user to add a already existing class from the flagged class section to the registration pane.		
Fit Criterion: The selected courses will be displayed in the “Registration” pane of the registration window		
Dependencies: Requirement covering the flagging of classes.		
Conflicts: Requirement prohibiting more than 20 flagged courses.		
Supporting Materials: Appendix A Figures 1, Appendix B Figure 3		
History: Created 11/3/13 by PCMJ		

Requirement #: 2	Category: Functional	Use Case #: 3
Description: The system shall be able to automatically check if a prerequisite is completed.		
Rationale: The system will check if all prerequisite have been completed, before allowing a flagged class to be stored in the “flagged classes” pane.		
Fit Criterion: The course will have appeared in the “flagged classes” pane if prerequisites have been met.		

Dependencies: Flagging of a course requirement
Conflicts: Requirement prohibiting more than 20 flagged courses
Supporting Materials: Appendix A Figure 1&2
History: Created 11/3/13 by PCMJ

Requirement #: 3	Category: Functional	Use Case #: 3
Description: The product shall be able to deny registration if perquisite is not completed.		
Rationale: If prerequisites have not been met, an error message will appear, the class will not flag, and the class will not appear in the “flagged classes” pane.		
Fit Criterion: All selected courses will appear in the “Flagged Courses” pane in the registration window		
Dependencies: Flagging of a course requirement		
Conflicts: Requirement prohibiting more than 20 flagged courses		
Supporting Materials: Appendix A Figure 1&2		
History: Created 11/3/13 by PCMJ		

Requirement #: 1	Category: Functional	Use Case #: 4
Description: The product shall automatically waitlist a flagged class, if a class is at full occupancy.		
Rationale: To allow the user to have access to a course in which they are interested in, even if the course is full.		
Fit Criterion: The course at full occupancy will appear in the “waitlist” window when flagged.		
Dependencies:		
Conflicts:		
Supporting Materials: Appendix A Figure 1		

History: Created 11/3/13 by PCMJ

Requirement #:	**Category: Functional**	Use Case #: 5
Description: The product shall allow students to remove a class from the registration pane		
Rationale: The user must have the option to deselect a course in the event that they are no longer interested in registering for.		
Fit Criterion: Once the course is selected, the product shall transfer the course to the "Flagged Courses" pane of the registration window.		
Dependencies: The course has been previously flagged.		
Conflicts:		
Supporting Materials: Appendix A Figure 1&2		
History: Created 11/3/13 by PCMJ		

Requirement #:	**Category: Functional**	Use Case #: 5
Description: The product shall display an error message if the class has already been removed.		
Rationale: The System will not be able to process a course removal that is already removed		
Fit Criterion: An error message will appear		
Dependencies: Course has already been removed from "registration" pane		
Conflicts:		
Supporting Materials: Appendix A Figure 1		
History: Created 11/3/13 by PCMJ		

Requirement #:	**Category: Functional**	Use Case #: 6
Description: The System shall allow the user to submit their registration sheet		

Rationale: The System must send the finalized course choices of the user to the University to have on record
Fit Criterion: The user will receive an email stating that their registration sheet has been submitted
Dependencies:
Conflicts:
Supporting Materials: Appendix A Figure 1, Appendix B Figure 1
History: Created 11/3/13 by PCMJ

Requirement #:	Category: Functional	Use Case #: 6
Description: The system shall prompt the user to verify the registration sheet		
Rationale: The user must verify their registration prior to submitting to acknowledge that their class choices are correct		
Fit Criterion: The System will show a new window that says that their registration sheet has been submitted		
Dependencies: Requirements involving the registration process		
Conflicts:		
Supporting Materials:		
History: Created 11/3/13 by PCMJ		

Requirement #:	Category: Functional	Use Case #: 6
Description: The system shall send the registration information to the University Database		
Rationale: The attendance sheet for the course must be updated in the university database to keep record of class size		
Fit Criterion: The courses registered for shall reflect one less available seat when searching for courses		

Dependencies:
Conflicts:
Supporting Materials: Appendix B Figure 1
History: Created 11/3/13 by PCMJ

Requirement #:	Category: Functional	Use Case #: 6
Description: The system shall alert the user if their registration does not reflect an appropriate number of credits		
Rationale: -If the registered credits are below 12, the student will not be considered a full time student and must be notified of this at registration time -If the registered credits exceed 18, the additional credits are not covered by the student's tuition. They must first receive advisor approval and will be billed for the additional credits.		
Fit Criterion: The system will prompt the user to either enter more courses, or receive advisor approval		
Dependencies: The student has registered for a combination of courses that is either below 12 credits, or above 18 credits		
Conflicts:		
Supporting Materials:		
History: Created 11/3/13 by PCMJ		

Requirement #:	Category: Functional	Use Case #: 6
Description: The system shall alert the student it they have not met the course prerequisites		
Rationale: The student must not be able to register for the course if they have not been prepared for the course		
Fit Criterion: The system shall produce the error message: "You have not met the prerequisites to register for the course at this time."		

Dependencies:
Conflicts: The student has previously been waived by a faculty member to enroll in the course
Supporting Materials:
History: Created 11/3/13 by PCMJ

Requirement #:	Category: Functional	Use Case #: 6
Description: The system shall notify the user of their completed registration via email		
Rationale: The student must have record of the courses they are taking in the event that an error occurs		
Fit Criterion: The student shall receive an email to their student account from E-Grade stating that their registration request has been received		
Dependencies: The course registration requirement		
Conflicts:		
Supporting Materials:		
History: Created 11/3/13 by PCMJ		

Requirement #: 1	Category: Functional	**Use Case #: 7**
Description: **The system shall check whether the amount of credits coincide with the student record requirement**		
Rationale: **User should be able to ensure that a class has been removed from the waitlist.**		
Fit Criterion: **The record of the student will be matched against course requirements**		
Dependencies: **Override registration use case [7].**		

Conflicts:
Supporting Materials: **(ER diagrams, charts, etc.)**
History: Created 11/3/13 by PCMJ

Requirement #: 2	Category: Functional	**Use Case #: 7**
Description: **The system shall check occupancy by checking the building room fire codes for maximum occupancy**		
Rationale: **The system shall make sure the student to be enrolled will fit in with permissions of maximum occupancy set by the building occupancy code**		
Fit Criterion: **The system shall determine the maximum occupancy for a class by building code for the class room assigned to the course section's permissions**		
Dependencies: **Override registration use case [7].**		
Conflicts:		
Supporting Materials: **(ER diagrams, charts, etc.)**		
History: Created 11/3/13 by PCMJ		

Requirement #: 3	Category: Functional	**Use Case #: 7**
Description: **The system shall check occupancy by checking University policies for maximum occupancy within the scope of course section**		
Rationale: **The system shall make sure the student to be enrolled will fit in with permissions of maximum occupancy set by the university course section occupancy code(s)**		
Fit Criterion: **The system shall determine the maximum occupancy for a class by university policy for the class room assigned to the course section's permissions**		

Dependencies: **Override registration use case [7].**
Conflicts:
Supporting Materials: **(ER diagrams, charts, etc.)**
History: Created 11/3/13 by PCMJ

Requirement #: 4	Category: Functional	**Use Case #: 7**
Description: **The system shall end the session when registration blocks are found preventing unauthorized registration**		
Rationale: **The system shall prevent succession of registration when unauthorized attempts occur**		
Fit Criterion: **The system shall end the process of current registration when blocks in registration are found which would prevent completion of registration**		
Dependencies: **Override registration use case [7].**		
Conflicts:		
Supporting Materials: **()**		
History: Created 11/3/13 by PCMJ		

Requirement #: 5	Category: Functional	**Use Case #: 7**
Description: **The system shall respond with the results from class to student requirements**		
Rationale: **The user will be made aware of requirements from class to student requirements**		
Fit Criterion: **The system presents class to student requirements.**		
Dependencies: **Override registration use case [7].**		

Conflicts:
Supporting Materials: **()**
History: Created 11/3/13 by PCMJ

Requirement #: 6	Category: Functional	**Use Case #: 7**
Description: **The system shall respond with the results from maximum occupancy**		
Rationale: **The user will be made aware of requirements from maximum occupancy**		
Fit Criterion: **The system presents maximum occupancy requirements.**		
Dependencies: **Override registration use case [7].**		
Conflicts:		
Supporting Materials: **()**		
History: Created 11/3/13 by PCMJ		

Requirement #: 7	Category: Functional	**Use Case #: 7**
Description: **The system shall respond with the results from unauthorized enrollment**		
Rationale: **The user will be made aware of unauthorized enrollment**		
Fit Criterion: **The system presents unauthorized enrollment results .**		
Dependencies: **Override registration use case [7].**		
Conflicts:		
Supporting Materials: **()**		
History: Created 11/3/13 by PCMJ		

Requirement #: 8	Category: Functional	**Use Case #: 7**
Description: **The system shall respond with a credit invalidation message when credits mismatch defined range limitations**		
Rationale: **The system shall alert system user of student invalidation of credits when a potential risk from the current process arises**		
Fit Criterion: **The system alerts the system user of credit invalidation originating from University Database**		
Dependencies: **Override registration use case [7].**		
Conflicts:		
Supporting Materials: **()**		
History: Created 11/3/13 by PCMJ		

Requirement #: 9	Category: Functional	**Use Case #: 7**
Description: **The system shall respond with a occupancy invalidation message when maximum occupancy is above the limit**		
Rationale: **The system shall alert system user of student invalidation of occupancy when a potential risk from the current process arises**		
Fit Criterion: **The system alerts the system user of occupancy invalidation originating from University Database**		
Dependencies: **Override registration use case [7].**		
Conflicts:		
Supporting Materials: **()**		
History: Created 11/3/13 by PCMJ		

Requirement #: 10	Category: Functional	**Use Case #: 7**

Description: **The system shall respond with a financial aid warning message when student's applicability becomes invalidated**
Rationale: **The system shall alert system user of student invalidation of financial aid when a potential risk from the current process arises**
Fit Criterion: **The system checks financial aid requirements of student identified by identifier with potential post enrollment status if registration would be successful**
Dependencies: **Override registration use case [7].**
Conflicts:
Supporting Materials: **()**
History: Created 11/3/13 by PCMJ

Requirement #: 11	Category: Functional	**Use Case #: 7**
Description: **The system shall respond with required amount of credits for student when financial invalidation occurs**		
Rationale: **The system shall alert system user of student required amount of credits when a potential risk from the current process arises**		
Fit Criterion: **The system checks required amount of credits of the student identified by identifier with potential post enrollment status if registration would be successful**		
Dependencies: **Override registration use case [7].**		
Conflicts:		
Supporting Materials: **()**		
History: Created 11/3/13 by PCMJ		

Requirement #: 12	Category: Functional	**Use Case #: 7**

Description: **The system shall respond with future standing of financial aid status changes which will take place if submission succeeds**
Rationale: **The system shall alert system user of student required amount of credits when a potential risk from the current process arises**
Fit Criterion: **The system returns potential financial aid status of student, post-succession from registration process**
Dependencies: **Override registration use case [7].**
Conflicts:
Supporting Materials: **()**
History: Created 11/3/13 by PCMJ

Requirement #: 13	Category: Functional	**Use Case #: 7**
Description: **The system shall display a warning that the student if currently enrolled as full time when a full time student drops below full time status credit requirements**		
Rationale: **The system shall alert system user of current full time status which would become part time after registration**		
Fit Criterion: **The system displays student status as full time currently a part time post succession of registration process**		
Dependencies: **Override registration use case [7].**		
Conflicts:		
Supporting Materials: **()**		
History: Created 11/3/13 by PCMJ		

Requirement #: 14	Category: Functional	**Use Case #: 7**
Description: **The system shall display the change from full time status to part time status when a full time student drops below full time status credit requirements**		
Rationale: **The system shall display the change from full time to part time status**		
Fit Criterion: **The system displays student status as full time currently a part time post succession of registration process**		
Dependencies: **Override registration use case [7].**		
Conflicts:		
Supporting Materials: **()**		
History: Created 11/3/13 by PCMJ		

Requirement #: 15	Category: Functional	**Use Case #: 7**
Description: **The system shall display a warning that the student needs to maintain a set amount of credits in accordance with university policy when a student's credits would fall under required amount**		
Rationale: **The system shall display inform the student of credit amount requirements for their standing at the University**		
Fit Criterion: **The system checks the University Database for required amount of credits student identified by their identifier needs to maintain and displays those requirements to the system user**		
Dependencies: **Override registration use case [7].**		
Conflicts:		
Supporting Materials: **()**		
History: Created 11/3/13 by PCMJ		

Requirement #: 16	Category: Functional	**Use Case #: 7**

Description: **The system shall log the verify registration eligibility transaction**
Rationale: **The system needs to record checks for registration eligibility within the University Database on every successfully full run**
Fit Criterion: **The system stores the check for registration eligibility within the University Database**
Dependencies: **Override registration use case [7].**
Conflicts:
Supporting Materials: **()**
History: Created 11/3/13 by PCMJ

Requirement #:	**Category: Functional**	Use Case #: 8
Description: The system shall enable a user to drop a course they are currently registered for		
Rationale: In the event that a student no longer wishes to take a course after they are registered for it, they should be able to remove themselves from the course		
Fit Criterion: The student shall receive a confirmation email stating that they have dropped the course		
Dependencies:		
Conflicts: Add-Drop has passed		
Supporting Materials: Appendix A Figure 1		
History: Created 11/3/13 by PCMJ		

Requirement #:	**Category: Functional**	Use Case #: 8
Description: The system shall notify the University Database in the students removal from the course		

Rationale: The University DB should contain accurate information regarding the course roster
Fit Criterion: The student shall receive an email from E-Grade stating that they have successfully dropped the course
Dependencies: If the student will be above 18 or below 12 credits
Conflicts:
Supporting Materials:
History: Created 11/3/13 by PCMJ

Requirement #:	Category: Functional	Use Case #: 8
Description: The system shall prompt the user to verify the course drop		
Rationale: The user must verify that they wish to drop the selected course		
Fit Criterion: The system shall prompt the user to verify that they wish to drop the selected courses		
Dependencies: Requirements involving dropping courses		
Conflicts:		
Supporting Materials:		
History: Created 11/3/13 by PCMJ		

Requirement #:	Category: Functional	Use Case #: 8
Description: The system shall notify the user of a course drop by email		
Rationale: The user must have confirmation of a course drop for record purposes		
Fit Criterion: The student shall receive an email from E-Grade stating that they have successfully dropped the course		
Dependencies: Requirements involving dropping courses		
Conflicts:		

Supporting Materials:
History: Created 11/3/13 by PCMJ

Requirement #: 1 Category: Functional **Use Case #: 9**
Description: The user is able to see the waitlist.
Rationale: The user should be able to view the list and ensure that classes aren't listed twice.
Fit Criterion: There must be classes on the waitlist.
Dependencies: Requirements dealing with "add to waitlist."
Conflicts:
Supporting Materials: (ER diagrams, charts, etc.)
History: **Created 11/3/13 by PCMJ**

Requirement #: 2 Category: Functional **Use Case #: 9**
Description: User is allowed to edit the waitlist.
Rationale: The user should be allowed to edit the waitlist. they should be able to add, and remove students from waitlists
Fit Criterion: There have to be classes on the waitlist.
Dependencies:
Conflicts:
Supporting Materials: (ER diagrams, charts, etc.)

History: **Created 11/3/13 by PCMJ**

Requirement #: 3	Category: Functional	**Use Case #: 9**
Description: The user is able to see the updated waitlist.		
Rationale: User should be able to ensure that a class has been removed from the waitlist.		
Fit Criterion: There must be at least one class on the waitlist.		
Dependencies: Requirements dealing with "add to waitlist"		
Conflicts:		
Supporting Materials: (ER diagrams, charts, etc.)		
History: **Created 11/3/13 by PCMJ**		

Switch Class

Requirement #: 1	Category: Functional	**Use Case #: 10**
Description: The user has the option to view classes and keep them in the flagged courses even after a student has registered for classes.		
Rationale: The student should be able to switch from one class to another.		
Fit Criterion: Student must be registered for classes. Student has to have a class they wish to switch into, on their flagged classes.		
Dependencies: Requirements dealing with "Add class"		
Conflicts:		
Supporting Materials: (ER diagrams, charts, etc.)		

History: **Created 11/3/13 by PCMJ**

Requirement #: 2	Category: Functional	**Use Case #: 10**
Description: The user is able to indicate the class section that is to replace the current class.		
Rationale: The will know for sure which class they wish to go into, so they will want to be able to indicate the section they want to go into		
Fit Criterion: desired class section should not be full.		
Dependencies: Requirements dealing with “aAdd class” Requirements dealing with “add to waitlist.”		
Conflicts:		
Supporting Materials: (ER diagrams, charts, etc.)		
History: **Created 11/3/13 by PCMJ**		

Requirement #: 3	Category: Functional	**Use Case #: 10**
Description: User is able to select the “switch” option when the two classes to be switched are selected.		
Rationale:		
Fit Criterion: Both the current class, and class the student is to switch into, must be selected.		
Dependencies: Requirements dealing with “Add class”		
Conflicts:		
Supporting Materials: ()		
History: **Created 11/3/13 by PCMJ**		

Requirement #: 4	Category: Functional	**Use Case #: 10**
Description: Student is prompted to verify the switch		
Rationale: User is going to want to verify that the student is added to the new class		
Fit Criterion: A class must be swapped from the original schedule.		
Dependencies:		
Conflicts:		
Supporting Materials: ()		
History: **Created 11/3/13 by PCMJ**		

Requirement #: 2	Category: Functional	**Use Case #: 10**
Description: The user is able to indicate the class section that is to replace the current class.		
Rationale: The will know for sure which class they wish to go into, so they will want to be able to indicate the section they want to go into		
Fit Criterion: desired class section should not be full.		
Dependencies: Requirements dealing with “aAdd class” Requirements dealing with “add to waitlist.”		
Conflicts:		
Supporting Materials: (ER diagrams, charts, etc.)		
History: **Created 11/3/13 by PCMJ**		

Requirement #: 3	Category: Functional	**Use Case #: 10**

Description: User is able to select the "switch" option when the two classes to be switched are selected.
Rationale:
Fit Criterion: Both the current class, and class the student is to switch into, must be selected.
Dependencies: Requirements dealing with "Add class"
Conflicts:
Supporting Materials: ()
History: **Created 11/3/13 by PCMJ**

Requirement #: 4	Category: Functional	**Use Case #: 10**
Description: Student is prompted to verify the switch		
Rationale: User is going to want to verify that the student is added to the new class		
Fit Criterion: A class must be swapped from the original schedule.		
Dependencies:		
Conflicts:		
Supporting Materials: ()		
History: **Created 11/3/13 by PCMJ**		

Requirement #: FR-OR-1	Category: Functional	**Use Case #: 13**
Description: **The system shall allow the user to enter a student identifier**		

Rationale: **The system needs to authenticate the student user**
Fit Criterion: **The user using the system needs to provide a means to identify themselves**
Dependencies: **Override registration use case [13].**
Conflicts:
Supporting Materials:
History: Created 11/3/13 by PCMJ

Requirement #: FR-OR-2	Category: Functional	**Use Case #: 13**
Description: **The system shall allow the user to enter a term**		
Rationale: **The user needs to make changes in the correct time**		
Fit Criterion: **User must have entered a term to enroll**		
Dependencies: **Override registration use case [13].**		
Conflicts:		
Supporting Materials: **(ER diagrams, charts, etc.)**		
History: Created 11/3/13 by PCMJ		

Requirement #: FR-OR-3a	Category: Functional	**Use Case #: 13**

Description: **The system shall allow the user to enter course name**
Rationale: **The system should have access to the database of classes in order to provide accurate results.**
Fit Criterion: **User must have entered course information by name to search for** **The system must have presented a text field labeled course name.**
Dependencies: **Override registration use case [13].**
Conflicts:
Supporting Materials: **(ER diagrams, charts, etc.)**
History: Created 11/3/13 by PCMJ

Requirement #: FR-OR-3b	Category: Functional	**Use Case #: 13**
Description: **The system shall allow the user to enter course section**		
Rationale: **The system should have access to the database of classes in order to provide accurate results.**		
Fit Criterion: **User must have entered course information by section to search for** **The system must have presented a text field labeled course section.**		
Dependencies: **Override registration use case [13].**		
Conflicts:		
Supporting Materials: **(ER diagrams, charts, etc.)**		
History: Created 11/3/13 by PCMJ		

Requirement #: FR-OR-4 Category: Functional **Use Case #: 13**
Description: **The system shall respond with information pertaining to course section**
Rationale: **The system shall provide information for verification and informative purposes**
Fit Criterion: **The system shall provide information for verification and informative purposes**
Dependencies: **Override registration use case [13].**
Conflicts:
Supporting Materials: **(ER diagrams, charts, data dictionary.)**
History: Created 11/3/13 by PCMJ

Add student to waitlist.

Requirement #: FR-OR-5 Category: Functional **Use Case #: 13**
Description: **The system shall check if the student identified from their identifier is eligible to enroll in selected course**
Rationale: **The system needs to show preconditions for the course selected and the status of the student pertaining to those preconditions**
Fit Criterion: **User in override use case must be aware of student's standing and status prior to completion of an override**
Dependencies: **Override registration use case [13].**
Conflicts:
Supporting Materials: **(ER diagrams, charts, etc.)**
History: Created 11/3/13 by PCMJ

Requirement #: FR-OR-6 Category: Functional **Use Case #: 13**

Description: **The system shall check for registration blocks for selected student**
Rationale: **The system needs to verify student eligibility in the registration availability field**
Fit Criterion: **The system needs to check registration availability pertaining to student by identifier**
Dependencies: **Override registration use case [13].**
Conflicts:
Supporting Materials: **(ER diagrams, charts, etc.)**
History: Created 11/3/13 by PCMJ

Requirement #: FR-OR-7	Category: Functional	**Use Case #: 13**
Description: **The system shall allow system user to digitally sign the override**		
Rationale: **The system needs to log and verify University Faculty's exceptions to registration**		
Fit Criterion: **A text field for University Faculty is displayed to log and proceed with the override of registration**		
Dependencies: **Override registration use case [13].**		
Conflicts:		
Supporting Materials: **(ER diagrams, charts, etc.)**		
History: Created 11/3/13 by PCMJ		

Requirement #: FR-OR-8	Category: Functional	**Use Case #: 13**
Description: **The system shall enroll the student in selected course section**		

Rationale: **The system needs to place the student in enrollment for that course section**
Fit Criterion: **The system needs to update the enrollment for course section by appending student identified by their identifier to the internal list of enrollment**
Dependencies: **Override registration use case [13].**
Conflicts:
Supporting Materials: **(ER diagrams, charts, etc.)**
History: Created 11/3/13 by PCMJ

Remove from waitlist.

Requirement #: FR-OR-9	Category: Functional	**Use Case #: 13**
Description: **The system shall add the student to the course section roster**		
Rationale: **The system needs to place the student in enrollment visibility for that course section**		
Fit Criterion: **The system needs to update the external enrollment for course section so those currently enrolled and University Faculty may see a list of students taking the course section**		
Dependencies: **Override registration use case [13].**		
Conflicts:		
Supporting Materials: **(ER diagrams, charts, etc.)**		
History: Created 11/3/13 by PCMJ		

Requirement #: FR-OR-10	Category: Functional	**Use Case #: 13**
Description: **The system shall update the student’s record in the University Database**		

Rationale: **The system needs to make changes on the student account within the University Database pertaining to enrollment of course section from the override registration**
Fit Criterion: **The University Database's student record shall be updated reflecting changes of enrollment from override registration**
Dependencies: **Override registration use case [13].**
Conflicts:
Supporting Materials: **(ER diagrams, charts, etc.)**
History: Created 11/3/13 by PCMJ

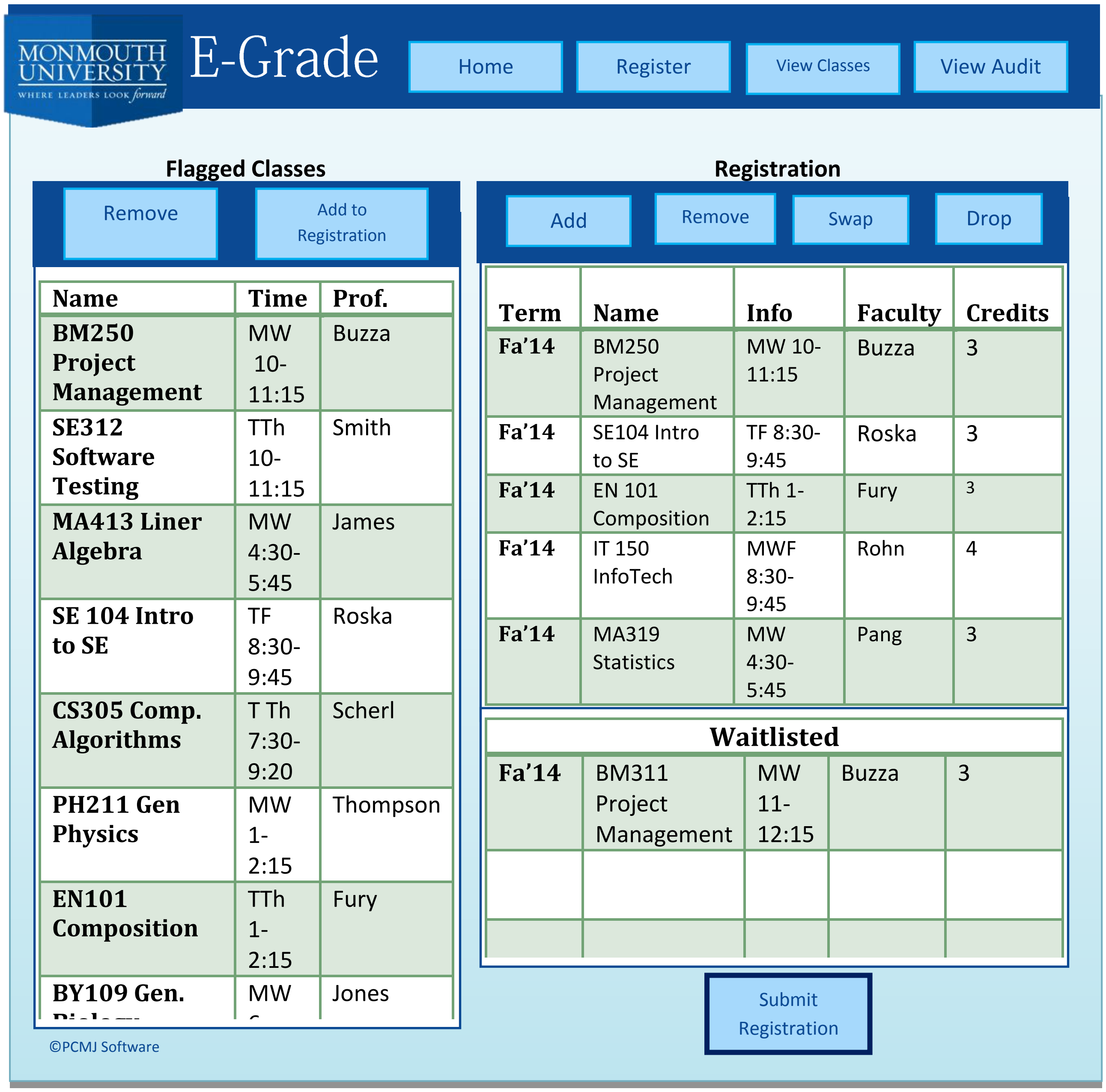
Requirement #:	Category: Functional	Use Case #: 15
Description: The system shall allow a faculty member remove a class from a student's registration		
Rationale: A faculty member must be able to remove a class from a student's registration if that student is taking a course in their major.		
Fit Criterion: The user will receive an email stating that their registration sheet has been submitted		
Dependencies: Registered courses		
Conflicts: Only faculty members in the courses department can remove same department courses, unless faculty member is students advisor.		
Supporting Materials: Appendix A Figure 1		
History: Created 11/3/13 by PCMJ		

Requirement #:	Category: Functional	Use Case #: 15
Description: The product shall notify student that a faculty member has altered their schedule.		
Rationale: The faculty member must submit the registration change, in order for the system to send out the automated email.		
Fit Criterion:		

The student user will receive an email from the registration system.
Dependencies: Faculty removal of a course.
Conflicts:
Supporting Materials:
History: Created 11/3/13 by PCMJ

Appendix A- Graphical User Interfaces

Figure 1:



Flagged Classes

Name	Time	Prof.
BM250 Project Management	MW 10-11:15	Buzza
SE312 Software Testing	TTh 10-11:15	Smith
MA413 Liner Algebra	MW 4:30-5:45	James
SE 104 Intro to SE	TF 8:30-9:45	Roska
CS305 Comp. Algorithms	T Th 7:30-9:20	Scherl
PH211 Gen Physics	MW 1-2:15	Thompson
EN101 Composition	TTh 1-2:15	Fury
BY109 Gen.	MW	Jones

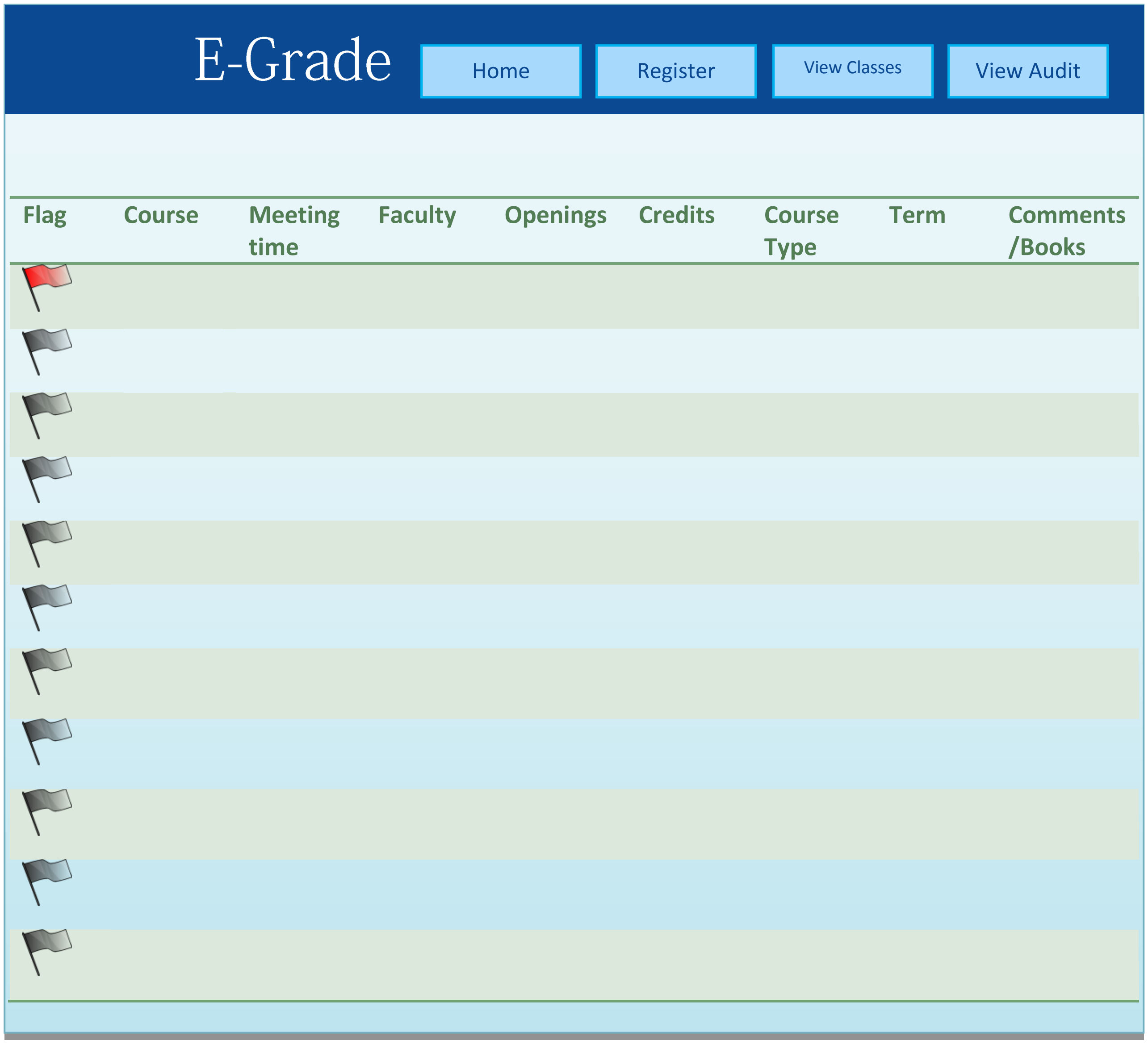
Registration

Term	Name	Info	Faculty	Credits
Fa'14	BM250 Project Management	MW 10-11:15	Buzza	3
Fa'14	SE104 Intro to SE	TF 8:30-9:45	Roska	3
Fa'14	EN 101 Composition	TTh 1-2:15	Fury	3
Fa'14	IT 150 InfoTech	MWF 8:30-9:45	Rohn	4
Fa'14	MA319 Statistics	MW 4:30-5:45	Pang	3

Waitlisted

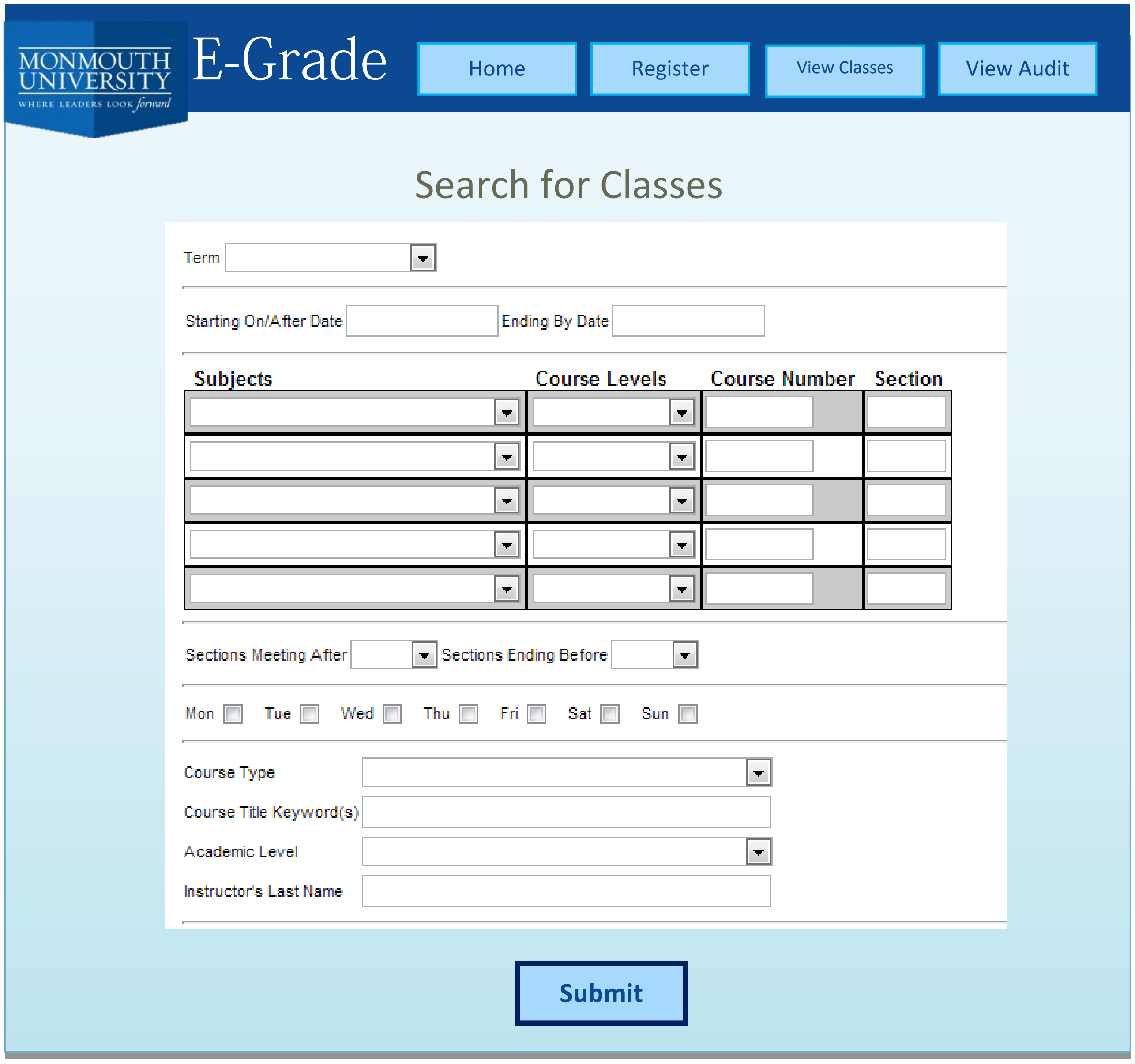
Fa'14	BM311 Project Management	MW 11-12:15	Buzza	3

Figure 2:



Flag	Course	Meeting time	Faculty	Openings	Credits	Course Type	Term	Comments /Books
🚩								
⚑								
⚑								
⚑								
⚑								
⚑								
⚑								
⚑								
⚑								
⚑								
⚑								

Figure 3:



Search for Classes

Term

Starting On/After Date
Ending By Date

Subjects	Course Levels	Course Number	Section

Sections Meeting After
Sections Ending Before

Mon Tue Wed Thu Fri Sat Sun

Course Type
Course Title Keyword(s)
Academic Level
Instructor's Last Name

Appendix B- Diagrams

Figure 1: Data Flow Diagram

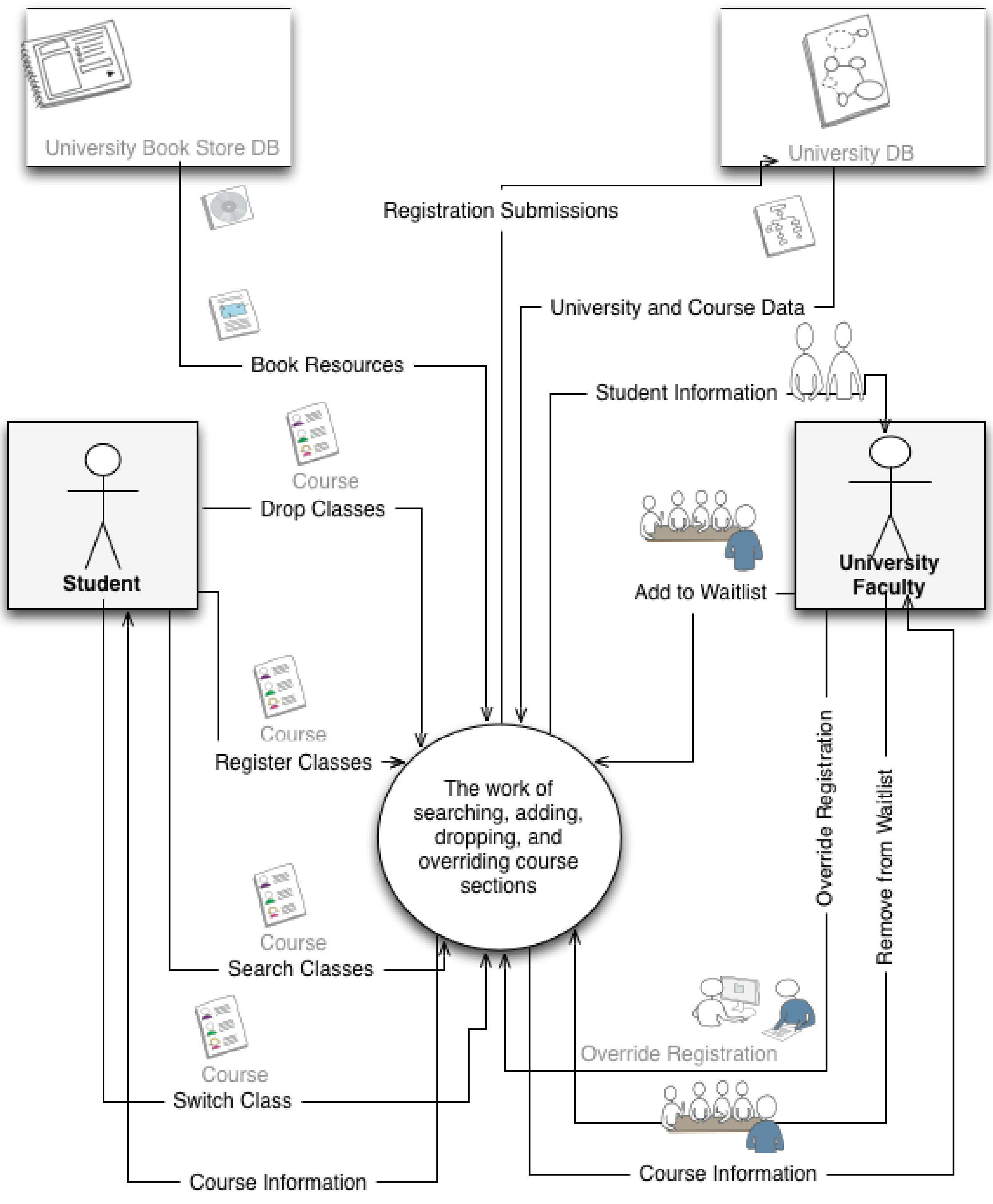


Figure 2: Implementation Diagram

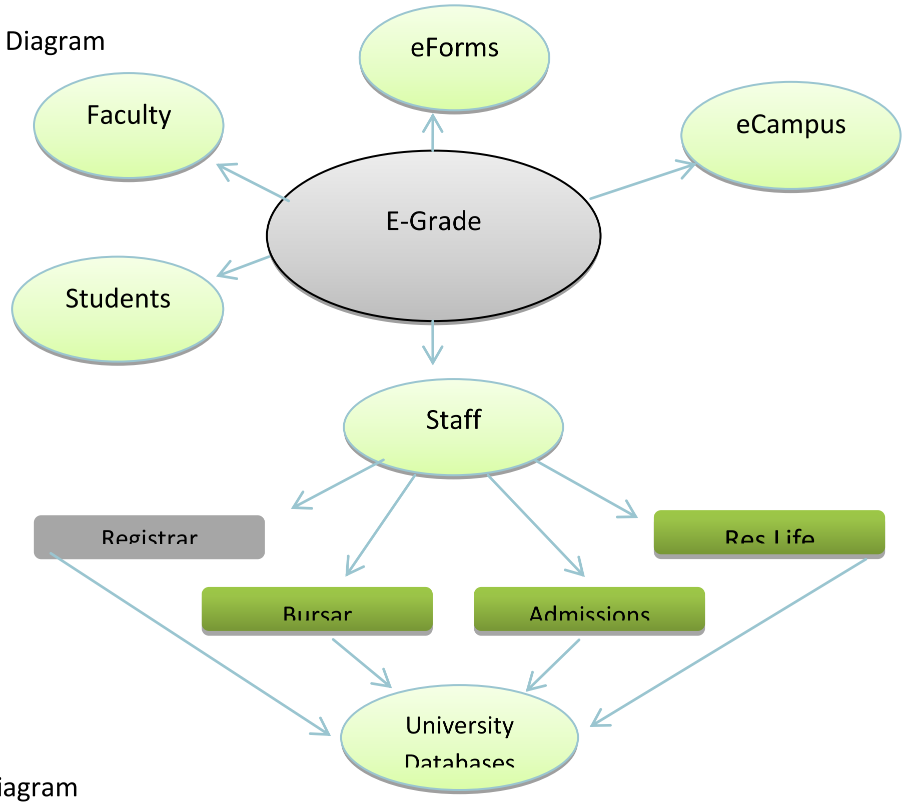


Figure 3: Product Scope Diagram

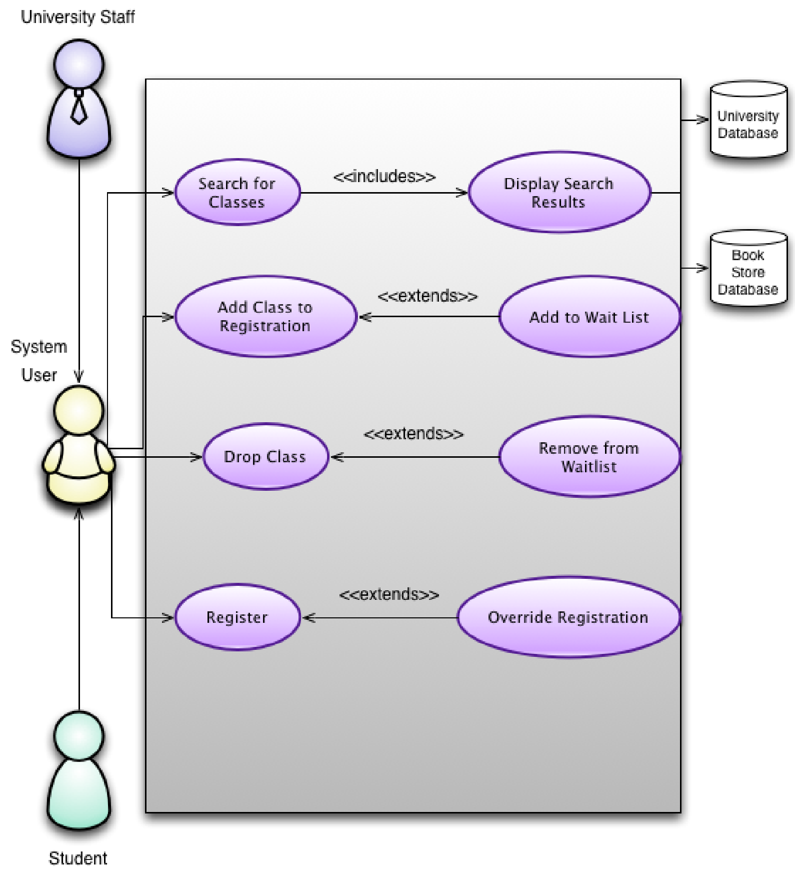


Figure 3: Use Case Diagram

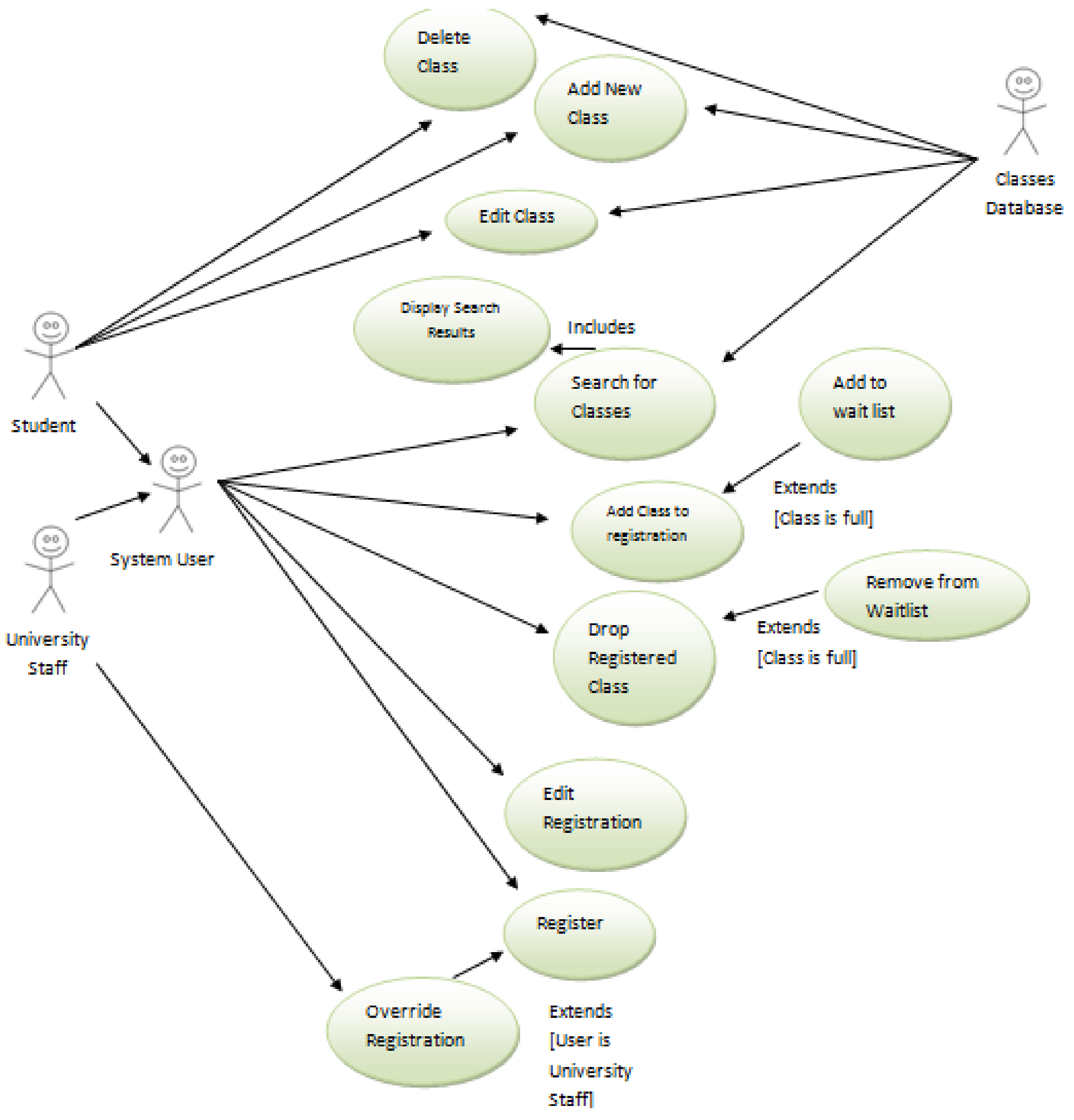


Figure 4: Add Student to Waitlist Entity Relationship Diagram

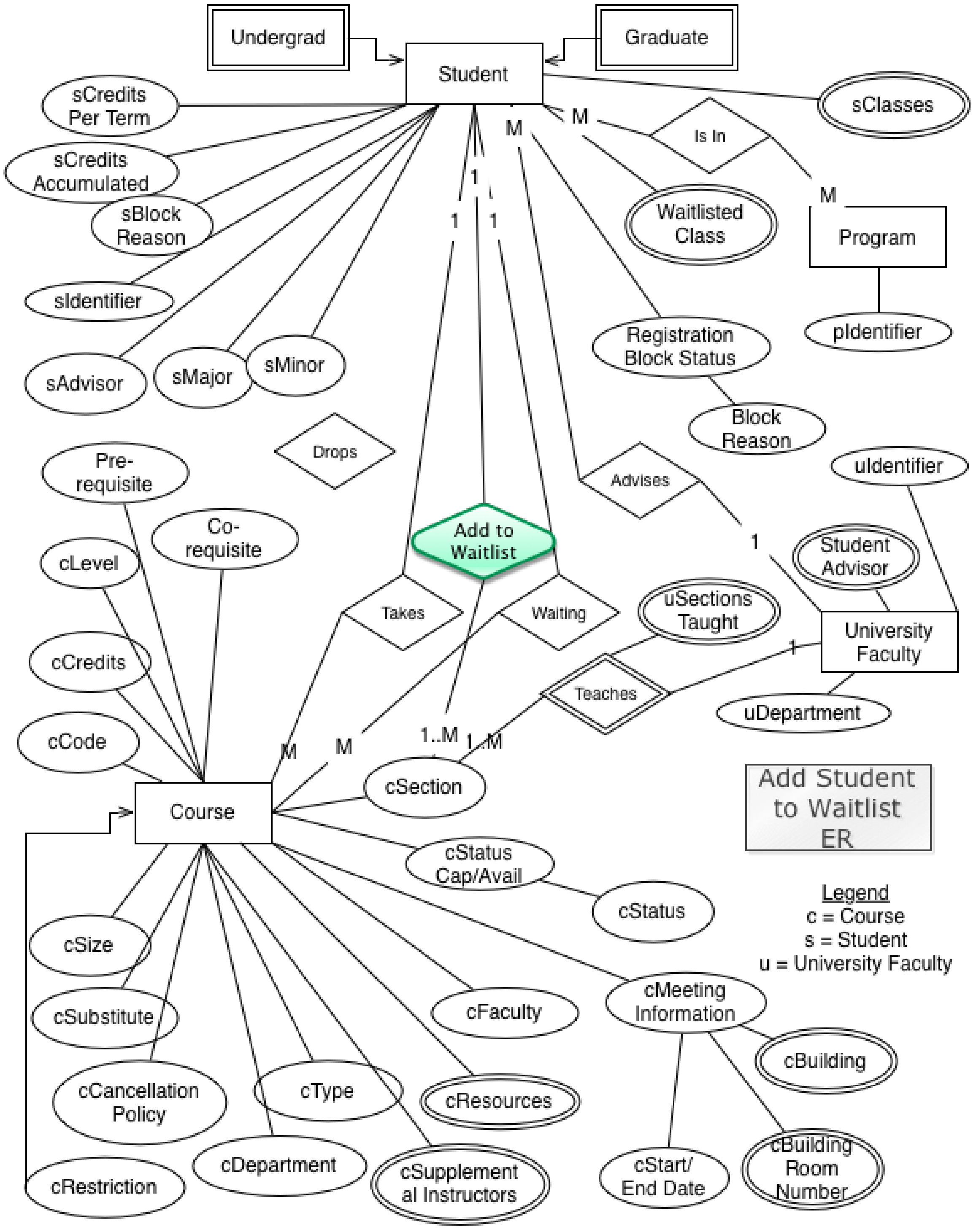


Figure 5:Drop Class Entity Relationship Diagram

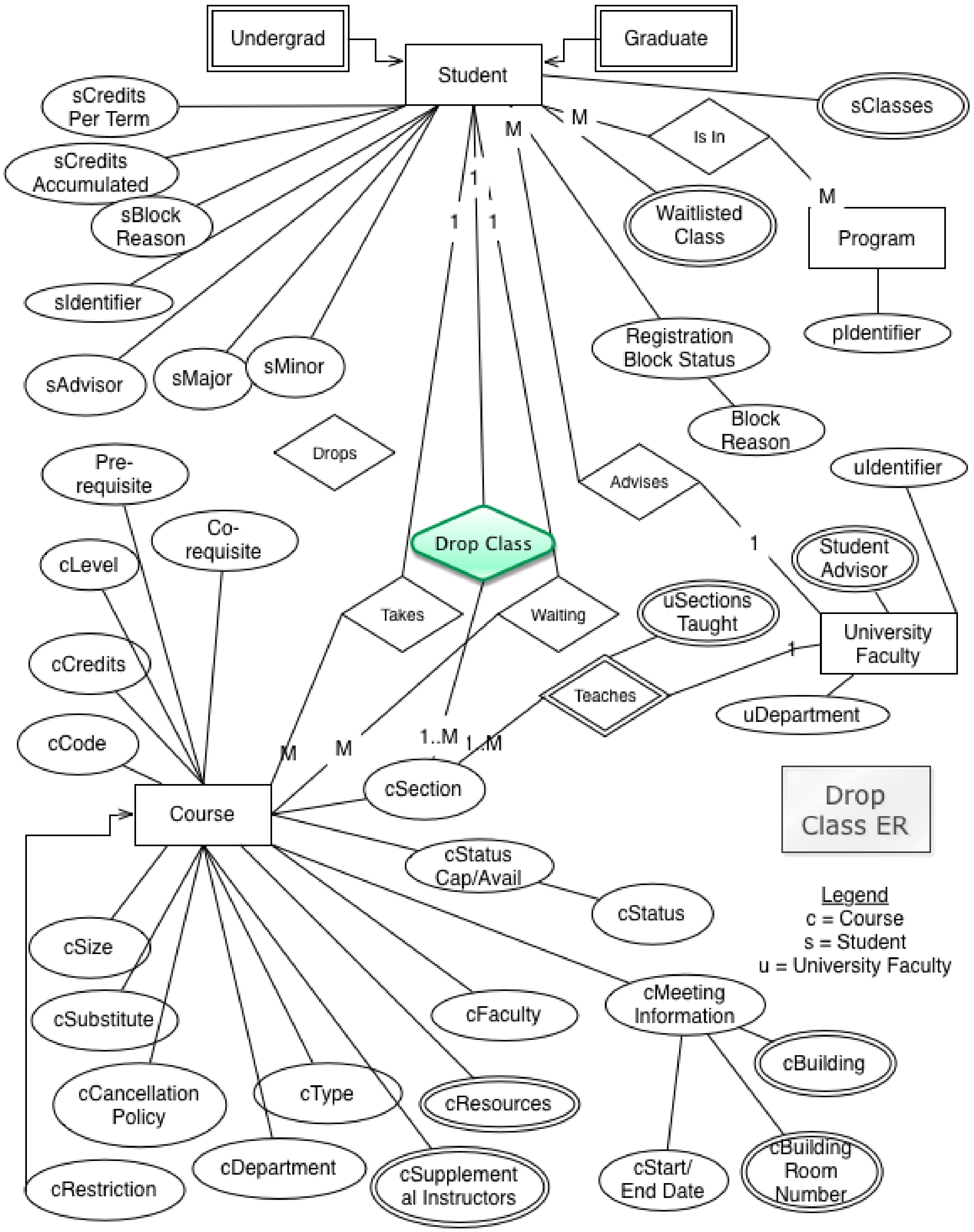


Figure 6: Override Registration Entity Relationship Diagram

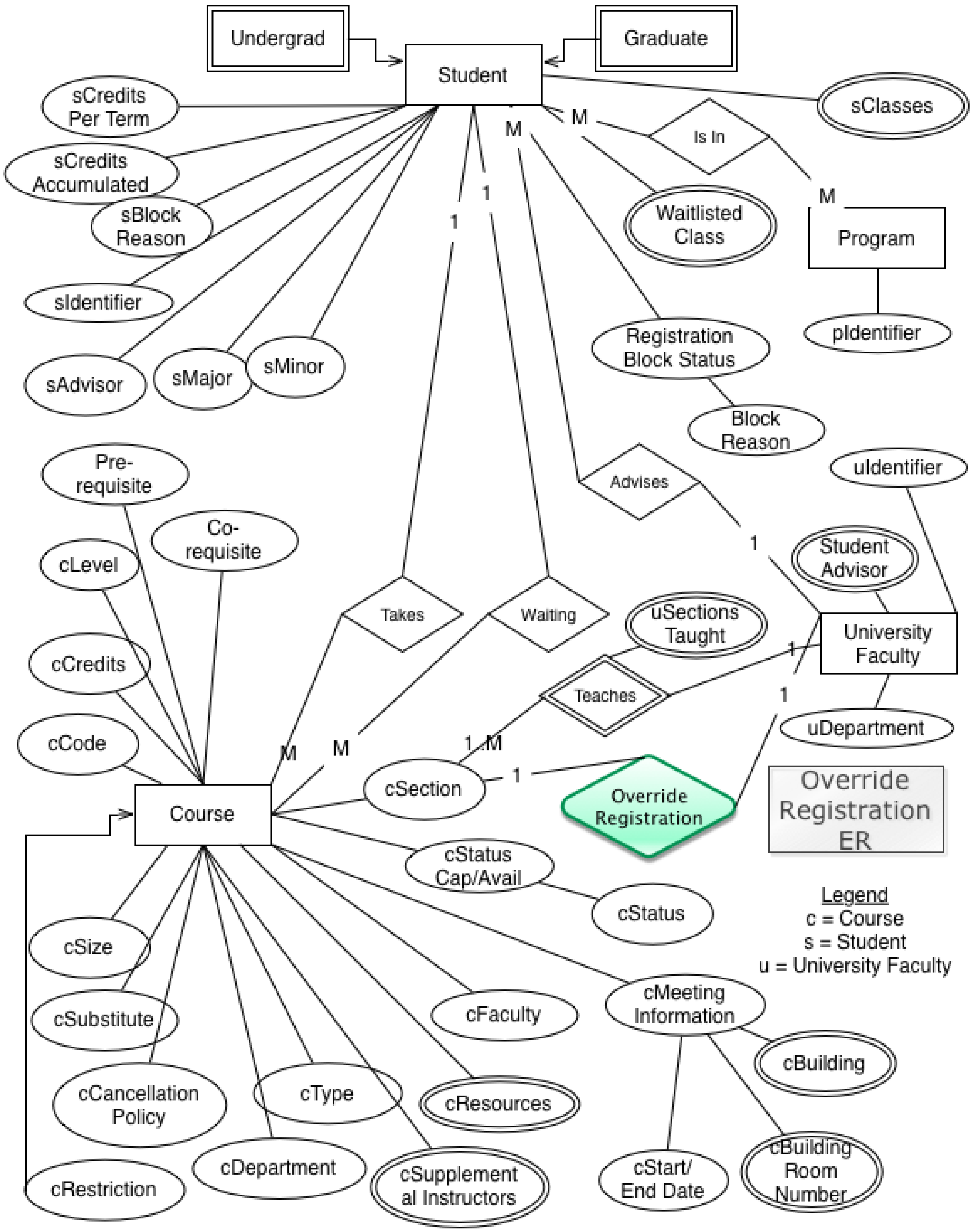


Figure 7:Add Class to Registration Entity Relationship Diagram

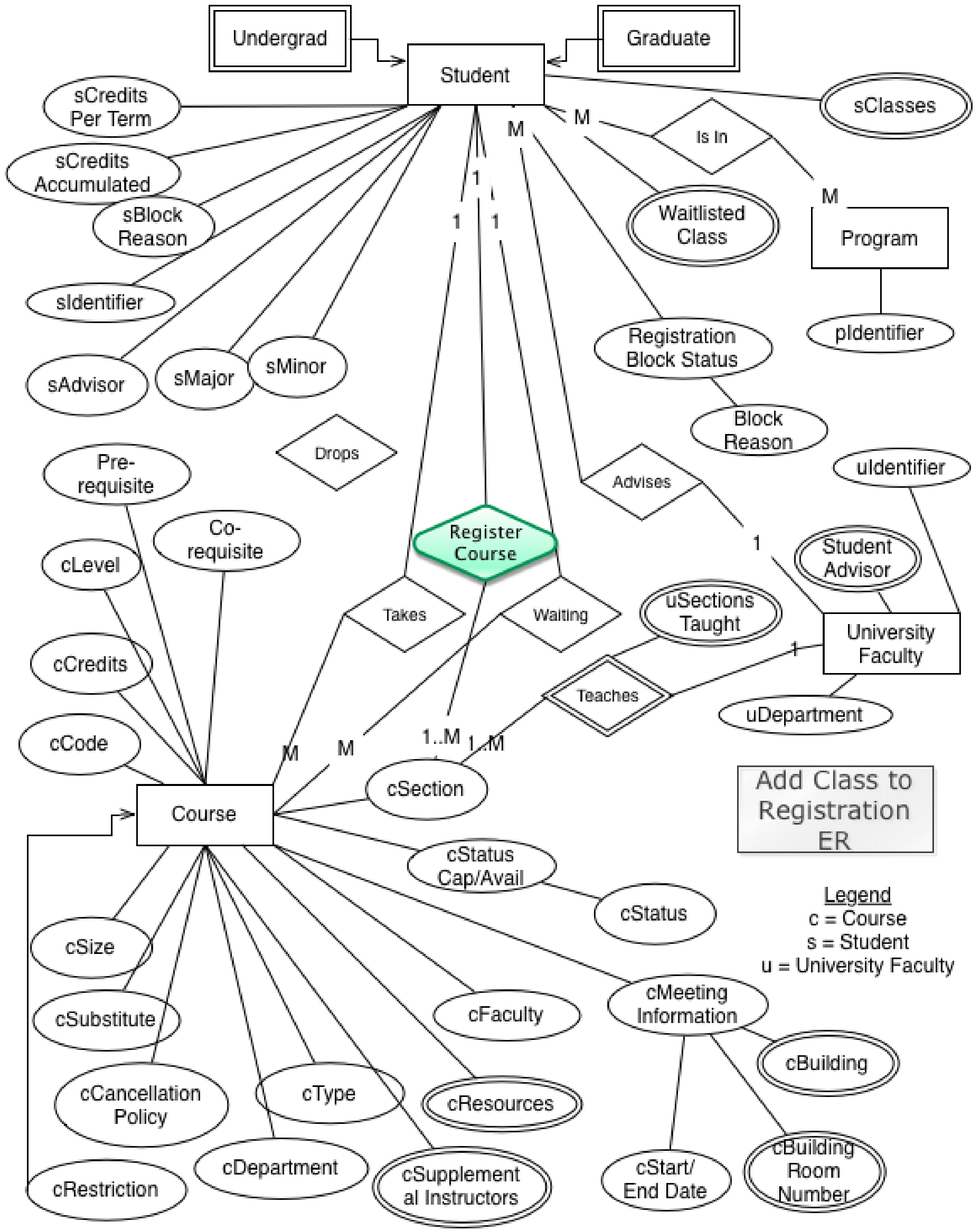
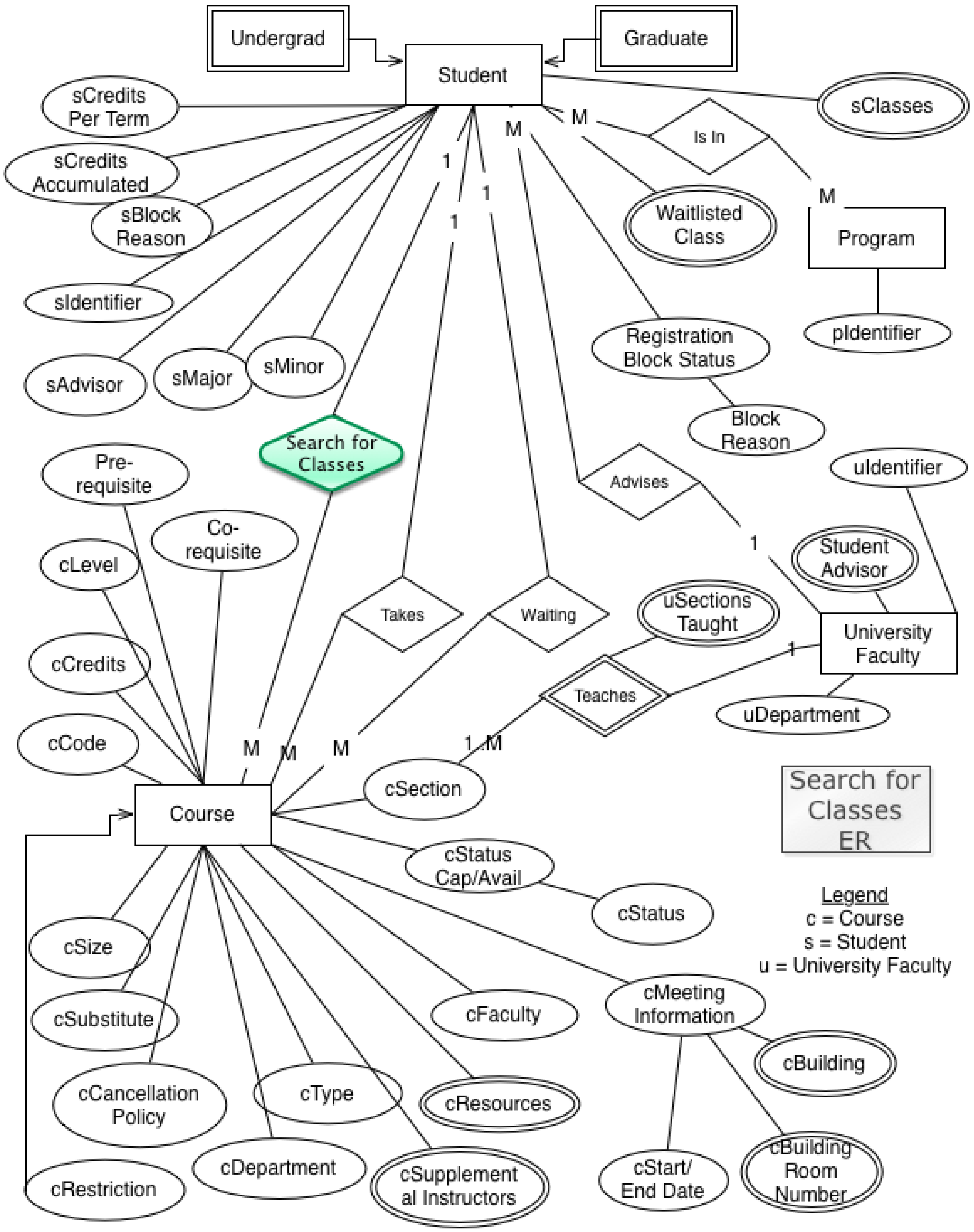


Figure 8: Search for Classes Entity Relationship Diagram



Appendix C- Registration Terminology

Course Codes

ARHIS	Art History
AT	Aesthetics
AUS	Study Abroad – Australia
BI.E	International Business electives
CC, CC1	Cross Cultural
CD	Cultural Diversity
CJHLS	Homeland Security
COxx	Communication majors
ENxx	English majors
EX1-EX5	Experiential Education
GLS	Global Sociology
GS	Gender Studies
GU	Global Understanding
HE.EL	Health Electives
HEPE	Health Physical Education Guided Electives
HO	Honors
HSxx	History majors
HS.SV	GE*2010 Historical Perspective
HY	Hybrid
IM	Interactive Media
ISP	GE*2010 Interdisciplinary Studies

ITA	Study Abroad – Italy
LA	Liberal Arts
LIT	Literature
LON	Study Abroad – London
LLC	Linked Learning Communities
MAT	Master of Arts in Teaching
MBA.xx	MBA majors
MC	Molecular Cell
ME	Marine & Environmental
MEBP	Marine and Environmental Biology & Policy Electives
NU.EL	Nursing Guided Elective
OL	Online
PO	Policy Studies
PSxx	Political Science majors
RD	Reasoned Oral Discourse
SI	Structural Inequality
SPA	Study Abroad – Spain
SS.SV	Social Science Survey
SUS	Sustainability
TL	Technological Literacy
TPS	Tech and Practice Skills
WT	Writing Intensive

Section Numbers – can indicate course restricted to a specific population. They are as follows:

EOF	Restricted to EOF
F	Restricted to freshmen
LON,RE,SPA	Study Abroad
AUS,ITA	Study Abroad
H	Restricted to Honors Students

Section Numbers - used in Fall and Spring also indicate schedule information:

A01,B01	“A” or “B” Pattern classes
HY	Hybrid Class
OL	Online section
01-49	Daytime Class
50+	Evening Class
60+	Weekend Class

- Section Numbers – For summer: The first character of the section number will designate the summer session (A,B,C,D, or E).

Course Levels
Developmental
First Year
Second Year
Third Year
Fourth Year
Graduate 500
Graduate 600
Doctorate

Academic Level	
GR	Graduate
UG	Undergrad
CE	Continuing Education
DOC	Doctorate

Course Types

LEC	Lecture
LAB	Lab

Glossary thoughts:

- E-Grade An enhanced upgrade from WebAdvisor, by Datatel / Ellucian compatible with other solutions by Ellucian, that other institutions currently use for registration purposes.
- Registration - all functions surrounding the act of students signing up for or removing themselves from classes
- Audit - a complete collection of a students completed courses, grades, and graduation requirements needed to aid in registration

- Clients (companies or individuals) - ones the system is being developed for, and also where all requirements will be gathered from
- Prerequisite - something that you officially must have or do before you can have or do something else
  -ORIGINAL END-
  -APPENDED POSSIBILITIES- Taken from WebAdvisor tutorials, terms relating to registration
- WEBregistration
- Worksheet 1
- Worksheet 2
- Academic Audit
- Curriculum Chart
- Sequence Chart
- WEBstudent
- Undergraduate
- Graduate
- ID
- Student
- Program
- Catalog
- Course Prerequisite Worksheet
- Undergraduate Curriculum Charts
- Undergraduate Sequence Charts
- Course Schedule Worksheet
- Term
- Semester
- Co-requisites
- Substitutions
- Building Codes
- Course Cancellation Policy
- Approval – displays for each student In-Person Registration Eligibility, WEBregistration Eligibility, priority registration date/start time, completed credits, advisor name and e-mail address, department and advisement status.
- Blocks
- Academic Probation
- Admission Status
- Action – a registration option for each of the WORKSHEET selections from the drop down. Three options are available: Register for the course, remove the course selection from the Worksheet Selection, add to waitlist – this option is available for if you wish to be waitlisted for closed course sections.
- RG – Register acronym

- RM – Remove from list acronym
- WL – Add to Waitlist