Web-Based Dynamic Registration System for
Aleutian Islands University

Systems Design and Implementations - CS5910 (Section 001)
Professor Naresh Gupta

Team Members:
Andrew Stephens (Team Leader) - asteph11@oldwestbury.edu - 516-348-4044
Gwen Alessi - galessi@oldwestbury.edu - 703-477-9338
Brian Mejia- bmejia2@oldwestbury.edu - 631-885-4068

www.owsysdb.com

Table of Contents

Proposed Meeting Time Slots

       The team leader proposed a project management methodology known as the **Agile Methodology**. Agile methodology includes concepts such as splitting large features into smaller, easier to comprehend Tasks. We have adopted **SCRUM** as our Agile Methodology due to its emphasis on a task and meeting scheduling routine called Sprints. **Our Sprints will be one-week periods**, instead of the preferred two week periods, due to time constraints. We are making full use of weekly **Sprint Reviews** and **Daily Standups**. **Daily Standups** are 15 minute meetings, occurring multiple times a week, that will be used to both discuss progress made by each team member from the previous day and to discuss plans that each team member plans to complete for the current day. Daily Standups will be helpful for keeping team members up-to-date with progress and roadblocks. After standups, we will open the forum to further questions and concerns. Our **Sprint Reviews** are full meetings that will occur once a week. During Sprint Reviews, all team members will Demo produced features and raise issues that came up during the Sprint. The **SCRUM Master** will then officially close the previous Sprint and officially start a new Sprint. **Tasks** will be created based on Client direction and they will be given order of priority during the current Sprint. Each team member will begin their next assigned Task based on the Sprint Backlog priority.

       Below you will find a table with our proposed meeting days and times:

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Start/ End	10AM - 11AM	10AM - 11AM	12PM - 3:30PM	10AM - 11AM	12PM - 3:30PM	10AM - 11AM	10AM - 11AM
SCRUM Meeting Type	Daily Standup	Daily Standup	Sprint Review	Daily Standup	Daily Standup / Meeting	Daily Standup	Daily Standup

Team Meeting Log

Date	Start Time	End Time	Meeting Notes / Completed Tasks	Team Members Attendance (S = Standup, F = Full) (C = Att. Class, O = Off. Hours)		
				Andrew Stephens	Gwen Alessi	Brian Mejia
pre-8/28/22	-	-	- Deployed AWS EC2 instance / researched dev-ops / and implemented Route 53 DNS registry, AWS Elastic IPs, EC2 Elastic (Application) Load Balancers - Deployed LAMP stack - Configured HTTPS certification - Established dev-ops connections and instructions for development programs - Explored Apache file structures - Created mock tables in database for relational testing and query testing - Created React components for testing - Updated manifest to contain proper information about the website. - Created Trello boards using SCRUM layout across all development fronts for increased productivity. - Formed AWS IAM credentials for team members for AWS project/billing access - Finished Task: Researched Agile development methodologies and adopted SCRUM for team management. - Finished Task: Discuss which university would be best for their Graduate/PHD programs. - Finished Task: Record Undergraduate Courses - Finished Task: Record Undergraduate Degrees	*F	*F	*F
8/28/22			- Updated React project packages.			
8/29/22			- Finished preliminary research on PHP security vulnerability handling with queries. - Finished task: Research on SRS requirements - Finished task: Preliminary research on MySQL stored procedures.			
8/30/22			- Finished preliminary research on MySQL stored procedures. - Finished Task: Standardize the Undergraduate Programs			
8/31/22			- Minor changes to EC2 instance and other AWS services			
9/1/22	11:00 PM	2:30 PM	- **Opened a team forum about concerns for the project.**	F	F	F
9/2/22	11:00 PM	1:30 PM	- Research and implementation of React Router with role authorization. - **Asked team leader questions about Server and other concerns regarding the project Stack**	F C	F C	F C

9/3/22			- Further implementation of role-based authorization successful.			
9/4/22			- Explored options for login authentication - Purchased Trello Premium			
9/5/22			- Updated Trello with tasks assigned for the next Sprint - Created 3 Calendars (SCRUM Meetings, Class Schedule, Team Conflicts) and shared them with the team - Tested Routing for authentication issues			
9/6/22	12:30	2:30 PM	- **Covered prior advancements and the basic project trajectory going forward during the team meeting.** - **Solidified team members (Andrew, Brian, Gwen) and team leader (Andrew)** - **Worked on rough outline of the SRS document** - **Copied over notes taken in class to the SRS document for easier access to information when writing document**	F C O	F C O	F C O
9/7/22	10:00 AM	11:00 AM	- **Reviewed User Constraints and Business Rules from class** - **Assigned team Goals due for Thursday's class** - **Used User Constraints rough draft to create a spreadsheet of information** - **Updated SRS Use Case rough draft**	S	S	S
9/8/22	1:00 PM	3:00 PM	- **Reviewed User Constraints and Forbidden use**	F	F	F
9/9/22	10:30 AM	11:00 AM	- **Updated Use Cases spreadsheet with Explicit Requirements, Explicit Forbidden, and Client-Silent Constraints** - **Initialized User Constraints rough draft for Tuesday's class**	S	S	S
9/10/22			- **Polished up the User Constraints draft due for Tuesday** - **Refined the Proposal document** - **Created and initialized a Proposal document**			
9/11/22	10:00 AM	11:00 AM	- **Discussed what the appropriate direction is for defining Use Cases** - **Created Visitor, Student, Faculty and Administrator Use Case Diagrams** - **Cleaned up all User Constraints**	S	S	S
9/12/22	10:00 AM	5:30 PM	- **Reviewed Use Case Diagrams in Standup** - **Edited Visitor, Student, Faculty, Administrator Use Case Diagrams** - **Created Researcher Use Case Diagram** - **Assisted other team members with finishing their Use Case Diagrams** - **Recorded Visitor, Student, Faculty, Administrator, and Researcher Use Case Statements to SRS**	S F	S F	S F
9/13/22	12:00 PM	3:30 PM	- **Cleaned up Use Case draft in preparation for class**	F O	F O	F O

9/14/22	9:00 AM	10:00 AM	**- Reviewed what necessary changes need to be made to Use Case Diagrams and Use Case Statements** **- Altered User Constraints, Use Case Diagrams, and Use Case Statements** **- Primed the Proposal Document for final edits for Thursday**	S	S	S
9/15/22	12:30 PM	3:00 PM	**- Completed Final copy of the Proposal Document**	F	F	F
9/16/22	10:00 AM	11:00 AM	- Worked on revisions provided by Prof. Gupta for the SRS - Updated Use Diagrams to remove unnecessary ones as stated by Prof. Gupta - Compiled a brief list of Entities for our Database - Worked on revisions provided by Prof. Gupta for the SRS - Updated Use Diagram to remove unnecessary ones as stated by Prof. Gupta - Added to the list of Entities for our Database - Worked on revisions provided by Prof. Gupta for the SRS - Updated Use Diagram to remove unnecessary ones as stated by Prof. Gupta - Added to the list of Entities for our Database	S	S	S
9/17/22			- Cleaned up Use Case Diagrams in preparation for Sitemaps prototyping - Prototyped a series of Sitemaps for Home and all Users - Revises Use Cases after Gupta critique			
9/18/22	10:00 AM	11:00 AM	- Reviewed Sitemaps with team - Discussed contacting Advisement about prerequisites - Began Design rough drafts in Adobe XD - Updated Trello with tasks assigned for the next Sprint - Worked on Database Summary and outline of entities and their relationships - Researched Stored Procedures/Functions/Routines - Researched Roles/Users	S	S	S
9/19/22	10:00 AM 2:30 PM	11:00 AM 4:30 PM	- Research done on Roles, Users and Grants in MySQL. - Added Use Cases to the Business Rules spreadsheet and created a format for breakdowns and real database Relational rules. - Research done on Roles, Users and Grants in MySQL	S	S	S
9/20/22	12:00 PM	3:30 PM	- Breaking down user constraints into basic statements that can be used in Database Business Rules - Translated Undergraduate prerequisites to the Undergraduate Courses sheet. - Filtering out Undergraduate Courses before	R F	R F	R F

			contacting Advisement for clarification on prerequisites - Breaking down user constraints into basic statements that can be used in Database Business Rules - Filtering out classes to contact for clarification on prerequisites - Recording Relational Business Rules (Breakdown) - Recording Undergrad Course Prerequisites and and sectioning off Prerequisites to Ask Department Advisors about			
9/21/22	9:00 AM	10:00 AM	- Completed conditional logic for prerequisites. Targeted Brian's completed Courses and simple, non-Advisement worthy Courses - Contact 3/3 Department Advisement and complete those prerequisites. - Completed contacts with Department heads and completed prerequisites - Attempted to contact Department Heads but no response.	S	S	S
9/22/22	12:00 PM	3:30 PM	- Discussed Home Page features and layout - Discussed how Stored Routines will be implemented - Discussed how the Prerequisites will function - Practiced Stored Procedures with Mock Data - Completed portion of Undergraduates Degree Prerequisites	R F	R F	R F
9/23/22	10:00 AM	11:00 AM	- Discussed the priorities due before Tuesday - Completed Undergraduates Degree Prerequisites - Completed Departments list - Completed Schools list - Completed Minor Degrees list - Completed Graduates Degrees list - Created outline for Graduate Degrees Requirements - Created outline for Graduate Courses - Assigned each member with a segment of Graduate Degree Requirements, Graduate Courses, and Minor Degree Requirements - Finished assigned Graduate Courses - Completed Undergraduates Degree Prerequisites	S	S	S
9/24/22			- Began work on formulating EERD Business Rules - Worked on Graduate Courses migration - Double checked Undergraduate degree courses for any inconsistencies or errors			
9/25/22	10:00 AM	11:00 AM	- Completed work on EERD Business Rules. - Created the Entity / Relational Entity list. - Began work on EERD in LucidChart. - Completed Graduate Courses migration	S	S	S
9/26/22	10:00 AM	11:00 AM	- Continued work on EERD. - Fixed errorsome EERD Business Rules. - Attempted a solution for prerequisite Entities and Relations -- albeit unsuccessfully.	S	S	S

			- Finished work on Graduate information migration - Began work on functional components for front end - Finished filling out Undergrad degree prerequisites			
9/27/22	12:00 PM	3:30 PM	- Cleaned up the EERD - Designed Prerequisites in the EERD - Discussed the general idea of Functional Components and the attributes within Functional Components -Continued work on functional components for Front End based on Use Cases - Discussed implementation of prerequisites in the database - Continued work on functional components (Student)	R F O	R F O	R F O
9/28/22	10:00 AM	11:00 AM	- Continued development of EERD revision 1. - Continued work on functional components for Front End based on Use Cases - Reviewed and studied EERD developed by team lead - Review of EERD	S	S	S
9/29/22	12:00 PM	3:30 PM	- Presented to Professor Gupta our EERD revision 1. - Continued discussion on prerequisites implementation - Discussed functional components with team during full meeting	R F O	R F O	R F O
9/30/22	10:00 AM	11:00 AM	- Began work on EERD Revision 2 based on revision 1 feedback from Professor Gupta. - Refined data for Degrees/Degree requirements for Undergraduates - Began work on Ph.D. Program data migration -Researched PhD degrees of Stony Brook, recorded findings in sheets to produce prerequisites	S	S	S
10/1/22	10:00 AM	11:00 AM	- Started production of atomic data for Course and Prerequisites based on the Old Westbury Course Catalog information and information provided by Stony Brook University. - Worked on Administration functional components - Finished work on Ph.D. Program data migration - Data Migration of PhD programs - Functional Components			
10/2/22			- Continued production of atomic data for Course and Prerequisites based on the Old Westbury Course Catalog information and information provided by Stony Brook University. - Started production of atomic data for Programs, Departments, Program Types, and Schools based on the Old Westbury Course Catalog information and information provided by Stony Brook University. - Continued development of EERD Revision 2.	S	S	S

			- Worked on Administration functional components -			
10/3/22	10:00 AM	11:00 AM	- Refined atomic data for Course, Program, Department, School, and Prerequisites based on the Old Westbury Course Catalog information and information provided by Stony Brook University. - Continued development of EERD Revision 2. - Finished work on Administration functional components for website - Began work on Faculty functional components for website	S	S	S
10/4/22	1:00 PM 7:00 PM	3:30 PM 8:00 PM	- Finished creating atomic data for Course, Program, Department, School, and Prerequisites. - Prepared a proper example of how Prerequisites will work in the database. - Continued development of EERD Revision 2, with suggestions based on feedback from Professor Gupta. - Confirmed a solution for Course Prerequisites. - Finished importing atomic data for Course, Program, Department, School, and Prerequisites. - Continued work on Faculty functional components for website - Tested course prerequisite tables in attempt to find any errors - Afternoon meeting with team - Reviewed prerequisite table proposal	S R F O	S R F O	S R
10/5/22	10:00 AM 3:30 PM	11:00 AM 6:30 PM	- Started EERD Revision 2 (w.attrs). - Created a query which successfully produces the appropriate basic output for checking a Student's Course Prerequisite completion. - Met with Andrew to discuss potential queries and implementation of Stored Procedures in database - Finished Faculty functional components and entities for website front-end	S	S	S
10/6/22	2:00 PM	3:30 PM	- Discussed and proposed solutions for Program Requirements. - Initiated discussion about logistical problems through complexity. Requested to Professor Gupta that we find a happy middle that supports complexity and rapid development. - Professor Gupta accepted our EERD Revision 2 (w.attrs) proposal, with minor modifications needed. - Worked on the modification for the degree requirements and passed the information on to Andrew. Awaiting any revisions	R F O	R F O	
10/7/22	12:30	1:30 PM	- Started EERD Revision 3 based on Professor Gupta's suggestions. Fixed attributes and relationships. - Revisions of the modifications made to the degree requirements approved and necessary changes made	S	S	S

			- Afternoon meeting - Functional components			
10/8/22			- Refined the Program requirements of all programs selected from source material. Tailored programs to meet criteria and modified credits to reflect inconsistencies from source. - Fixed "Missing Courses" from degree requirements, cross-referenced with every available course catalog to get the most accurate information possible.	S	S	S
10/9/22	12:00 PM	1:30 PM	- Continued updates for EERD Revision 3. Added missing attributes and trimmed Entities. - Began Program Requirement validation by cross-checking required courses. Missing courses from the Course Catalog were removed. Made adjustments to credit totals. - Worked on validating prerequisites and checking each one by hand to ensure the logic of the tables was sound -Progress on functional components	S	S	S
10/10/22	10:00 AM	11:30 AM	- Finalized updates for EERD Revision 3. - Began creation of Schema list based on EERD. - Finished Prerequisites validation. Missing courses from the Course Catalog were removed. Made adjustments to Prerequisite group logic. - Continued work on the validation of the prerequisites by hand and included the cross checking with MySQL tables to ensure returned values were the same - Finished functional components	S	S	S
10/11/22	12:00 PM	3:30 PM	- Formally explained the procedure for creating Course Requirement trees. Trees will display the possible routes for a Student to complete a Program. - Began work on Degree Requirement trees for the minors -Validated course prerequisites with stored procedure	R F O	R F O	R F O
10/12/22	10:00 AM	11:00 PM	- Completed data Program Requirements. Ready for testing. - Polished EERD Revision 3 with Program Requirement tables. - Completed Schema creation based on the EERD Revision 3. - Finished degree requirement trees for minors and finished cross referencing hand-done prerequisite data with MySQL query prerequisite data -Validated course prerequisites with stored procedure	S	S	S
10/13/22	12:00 PM	3:30 PM	- Began turning Program Requirements into relational data. - Validated degree requirement trees created for minors - Begin creating degree requirement trees for Undergraduates	R F O	R F O	R F O

10/14/22	10:00 PM	11:00 PM	- Completed turning Program Requirements into relational data. - Completed creating relational tables in Google Sheets to fill with Program Requirement Relational Tables. - Began work on Degree requirement trees for Graduate students - Continue creating degree requirement trees for Undergraduates	S	S	S
10/15/22			- Began validation of Minors Program Requirement trees - Began validation work on Undergraduate Program Requirement trees - Began creation of Users based on age and quantity - Continued work on Degree requirement trees for Graduate students			
10/16/22	12:00 PM	1:00 PM	- Continued validation of Minors Program Requirement trees - Continued validation work on Undergraduate Program Requirement trees - Continued creation of Users based on age and quantity -Validated degree requirement trees created for Graduate Students -Continue creating/revising degree requirement trees for Undergraduates	S		S
10/17/22	2:00 PM	3:00 PM	- Completed validation of Minors Program Requirement trees - Continued validation work on Undergraduate Program Requirement trees - Completed creation of Users based on age and quantity -Began work on Degree requirement trees for PHD students -Finish creating/revising degree requirement trees for Undergraduates	S	S	S
10/18/22	12:00 PM	3:30 PM	- Discussed with team the appropriate means of producing the required queries for specific Functional Components - Began creation of Student Users and Faculty Users based on age and quantity - Continued work on Degree requirement trees for PHD students - Began recording potential query descriptions -Begin recording query descriptions	R F O	R F O	R F O
10/19/22	10:00 AM 2:00 PM	11:30 AM 7:00 PM	- Completed creation of Student Users and Faculty Users based on age and quantity - Began to fix program trees that contained errors. -Validated all trees developed for Degree requirements -Continue recording query lists and descriptions	S F	S F	S F
10/20/22	12:00 PM	3:30 PM	- Determine the best course of action to create certain quantities of different Students in any	R F	R F	R F

			level based on available programs. - Continued to fix program trees that contained errors. - Finished recording list of query descriptions			
10/21/22	3:00 PM	4:00 PM	- Began analyzing programs and developing numerous sheets with guidelines for creating Student histories rapidly. - Finished fixing program trees that contained errors. - Began preliminary work on Student Histories as well as correcting errors made in data beforehand - Began creation of Student Histories starting with Undergrad Full Time	S	S	S
10/22/22			- Finished analyzing programs and developing numerous sheets with guidelines for creating Student histories rapidly. - Began creating sheets with specific Student History outlines based on program, time, and student level. - Research done on pagination and other useful MySQL features that will help in the management of data -Continuation of Student History Creations			
10/23/22	10:00 AM	11:00 AM	- Finished creating sheets with specific Student History outlines based on program, time, and student level. - Continued research and prepared for the beginning of Student History Data Generation -Continuation of Student History Creations	S	S	S
10/24/22	10:00 AM	11:00 AM	- Instructed the team how to go about using pre-created outline resources for Student History creation. Requested that they begin creating histories. - Began work on generating Student Histories for Undergrad Students Full-Time with 1 Major -Continuation of Student History Creations	S	S	S
10/25/22	12:00 PM	3:30 PM	- Oversaw the production of Student Histories - Began development of double program trees for 2 Major and 1 Major/1 Minor. - Continued work on generating Student Histories for Undergrad Students Full-Time with 1 Major -Continuation of Student History Creations	R F	R F	R F
10/26/22	10:00 AM	11:00 AM	- Oversaw the production of Student Histories - Continued development of double program trees for 2 Major and 1 Major/1 Minor. - Fixed the SRS to contain Part Time/ Full TIme credit limits based on SUNY standards. - Began work on generating Student Histories for Undergrad Students Full-Time with 1 Major and 1 Minor -Continuation of Student History Creations	S	S	S
10/27/22	12:00 PM	3:30 PM	- Oversaw the production of Student Histories - Finished development of double program trees	R F	R F	R F

Date			Description			
			for 2 Major and 1 Major/1 Minor. - Added Admin login information that coincides with the data of all Team members. - Brainstormed initial version of Time Windows. - Began creating sheets with specific History outlines based on double program, time, and student level. - Continued work on generating Student Histories for Undergrad Students Full-Time with 1 Major and 1 Minor -Continuation of Student History Creations			
10/28/22	10:00 AM 5:00 PM	11:00 AM 5:30 PM	- Oversaw the production of Student Histories - Created initial Time Windows. - Created Holds. - Fixed EERD to properly reflect the current database. - Finished creating sheets with specific History outlines based on double program, time, and student level. - Met with Professor Gupta over Zoom. - Fixed Time Windows based on professor's suggestions. - Fixed issues with Student Histories outlines. - Began work on generating Student Histories for Undergrad Students Full-Time with 2 Majors - Began work on generating Student Histories for Undergrad Students Part-time with 1 Major -Continuation of Student History Creations	S O	S O	S O
10/29/22			- Oversaw the production of Student Histories - Imported Personal Information and Logins into the database. - Obtained a good outline for Section Time Slots. - Imported Student, Undergrad Students, Grad Students, Undergraduate Full Time, Undergrad Part Time, Masters Graduates, PhD Graduates, Full Time Masters, Part Time Masters, Full Time PhD, Part Time PhD, Faculty, Faculty Part Time, Faculty Full Time, Administrators, Researchers. - Created query for website authentication containing role authorization. - Continued work on generating Student Histories for Undergrad Students Part-Time with 1 Major - Began work on generating Student Histories for Undergrad Students Part-Time with 1 Major and 1 Minor - Began work on generating Student Histories for Undergrad Students Part-Time with 2 Majors -Continuation of Student History Creations			
10/30/22	10:00 AM	11:00 AM	- Oversaw the production of Student Histories - Created Stored Procedure for displaying Personal Information. - Created backend code for requesting Personal Information. - Created frontend code for displaying Personal Information.	S	S	S

			- Began Work on Graduate Student Histories - Continuation of Student History Creations			
10/31/22	10:00 AM	11:00 AM	- Oversaw the production of Student Histories - Worked on frontend display of Personal Information. - Developed a grade bell curve for both pass/fail range and above-C range. - Began Work on PHD Student Histories - Continuation of Student History Creations	S	S	S
11/1/22	12:00 PM	3:30 PM	- Oversaw the production of Student Histories - Fixed problem with Github. - Created program for parsing created Student Histories for record entry. - Created program for splitting created Student History courses for record entry. - Created the initial version of the Student History Master list. - Created a sheet to display problems with Student count in each Course within a specific semester. - Finished initial generation of Student Histories - Went over any errors and data inconsistencies in the initial Student Histories - Finished initial creation of all Student Histories - Reviewed entries for errors	R F S	R F S	R F
11/2/22	10:00 AM	11:00 AM	- Oversaw the production of fixing Student Histories - Continued fullstack production. - Created user error feedback for invalid authentication input at the login screen. - Began work on validation on Student Histories - Validating student count within courses through semester - Changed/swapped courses that did not satisfy min 5 student count - Called for creation of new students when necessary	S	S	S
11/3/22	12:00 PM	3:30 PM	- Oversaw the production of fixing Student Histories - Fixed issue with displaying Personal Information properly. - Brainstormed with the Team the proper landing page for a user and the splitting up of Personal Information and Account information on different screens. - Continued work on validation and correction on Student Histories - Validating student count within courses through semester - Changed/swapped courses that did not satisfy min 5 student count - Called for creation of new students when necessary	R F	R F	R F
11/4/22	10:00 AM	4:00 PM	- Oversaw the production of fixing Student	S	S	S

			Histories - Full stack development of the Course search feature. - Continued work on validation and correction on Student Histories - Validating student count within courses through semester - Changed/swapped courses that did not satisfy min 5 student count - Called for creation of new students when necessary			
11/5/22			- Oversaw the production of fixing Student Histories - Completed full stack development of basic Course Search feature. - Continue to develop more advanced features for Course Search. - Completed the preliminary history creation based on created student history data. - Began full stack development of Profile Information page and components. - Continued work on validation and correction on Student Histories - Validating student count within courses through semester - Changed/swapped courses that did not satisfy min 5 student count - Called for creation of new students when necessary			
11/6/22	10:00 AM	11:00 AM	- Oversaw the production of fixing Student Histories - Installed MaterialUI for its pagination component. - Enhanced the Course Search by adding pagination and column ordering. - Began full stack development of the View Program Requirements component. - Maneuvering student histories so that every class met the lower bounds for a course section - Validating student count within courses through semester - Changed/swapped courses that did not satisfy min 5 student count - Called for creation of new students when necessary	S	S	S
11/7/22	10:00 AM	11:00 AM	- Oversaw the production of fixing Student Histories - Continued full stack development of the View Programs component. - Attempts made to refactor tables and Primary Keys for easier foreign key use. - Maneuvering student histories so that every class met the lower bounds for a course section - Correcting duplicates of classes in a student's history -Finished validating student count within courses through semesters	S	S	S

11/8/22	12:00 PM	3:30 PM	- Oversaw the production of fixing Student Histories - Began full stack development of the View Programs component. - Assigned team with task to produce periods for every course section between Spring 23 and Fall 22. - Assigned time slots to course sections that were created -Updating course section time slots so course section times do not overlap per student	R F	R F	R F
11/9/22	10:00 AM	11:00 AM	- Completed development of software to assist in producing scheduling of students to course sections based on a number of factors, including available period. - Continued full stack development of the View Programs component. - Continued full stack development of the View Program Requirements component. - Fixed time slots that course sections were assigned to in order to satisfy the constraints	S	S	S
11/10/22	12:00 PM	3:30 PM	- Completed development of software to assist in producing scheduling of students to course sections based on a number of factors, including available period. - Assigned team with tasks for fixing student's section registrations. - Small fixed to full stack development of the View Programs components. - Used fixed student's section registrations to ensure proper student count in each section. - Began writing the 'getAll' stored procedures for data transfer to the front end	R F	R F	R F
11/11/22	10:00 AM	11:00 PM	- Assigned team with Stored Procedures assignment for creating All queries. - Began official record creation of Course / Sections, Student / Histories, Student / Programs, Course Section / Periods, Periods / Time Slots, Periods, Timeslots, Semesters, etc. - Continued the creation of 'getAll' stored procedures - Finished Writing of "getAll" Stored Procedures in MySql for Data retrieval in the front end	S	S	S
11/12/22			- Assigned team with Stored Procedures assignment for creating ID-targeting queries. - Finished official record creation of Course / Sections, Student / Histories, Student / Programs, Course Section / Periods, Periods / Time Slots, Periods, Timeslots, Semesters, etc - Began transferring Faculty to Part and Full Time to fix counts of sections in a semester. - Began creating buildings and rooms for courses, faculty, and other room types. -Began and Finished the writing of 'getByID' stored procedures in the MySQL database			

11/13/22	10:00 AM	11:00 AM	- Full stack development: - Focus on changes to Course Catalog. - Began work on use case stored procedures as listed within the SRS and the User Requirements/Constraints - Update "getByID" Stored Procedures for secure retrieval - Export/Import Database data into a test instance for query writing - Begin organizing/assigning query development in MySql based on available data	S	S	S
11/14/22	10:00 AM	11:00 AM	- Full stack development: - Focus on changes to Course Catalog. - Continued the work on the generation of stored procedures within MySQL database - Begin planning/writing Stored Procedures from checklist in "AIU - Schemas/Queries" sheet	S	S	S
11/15/22	12:00 PM	3:30 PM	- Full stack development: - Focus on Major changes to Course Catalog, small additions to Login Form. - Continued the work on the generation of stored procedures within MySQL database -Continuation of writing Stored Procedures. -Specifics: "createNewUser" start writing nested procedures for "createStudent", Runs successfully in test instance - Implement DELETE/UPDATE CASCADE in Test Instance	R F	R F	R F
11/16/22	10:00 AM	11:00 AM	- Full stack development: - Began development of the Master Schedule. - Continuation of writing Stored Procedures. - Specifics: "createNewUser" nested Procedures, Provide "checks" for input parameters within Stored Procedure - Successfully test "checks"	S	S	S
11/17/22	12:00 PM	3:30 PM	- Full stack development: - Continued development of the Master Schedule. - Assisted team with Stored Procedures. - Continued the work on the generation of stored procedures within MySQL database -Continuation of writing Stored Procedures. - Specifics: "createNewUser" nested Procedures, creation of "createFaculty, createAdmin, createResearcher" procedures for additional UserType parameter in "createNewUser" - Test new nested procedures with checks/finished "createUser" - Creation of additional Stored Procedures: "checkIfStudentExists", checkIfFacultyExists", "updateCourseNameOrDescr"	R F	R F	R F
11/18/22	10:00 AM	4:00 PM	- Full stack development: - Nav changes, add user console and admin controls. - Created a more dynamic navigation system. - Refined the side nav bar. - Continued the work on the generation of	S	S	S

			stored procedures within MySQL database			
11/19/22	1:30 PM	5:00 PM	- Full stack development: - Nav changes, add user console and admin controls, styling to components. - Altered user console home admin to contain options. - Continued the work on the generation of stored procedures within MySQL database	S	S	S
11/20/22	10:00 AM	11:00 AM	- Full stack development: - Nav changes, add user console and admin controls, styling to components. - Altered user console home admin to contain options. - Continued the work on the generation of stored procedures within MySQL database	S	S	S
11/21/22	1:30 PM	5:00 PM	- Full stack development: - Nav changes, add user console and admin controls, styling to components. - Altered user console home admin to contain options. - Continued the work on the generation of stored procedures within MySQL database	S	S	S
11/22/22	12:00 PM	3:30 PM	- Full stack development: - Nav changes, add user console and admin controls, styling to components. - Altered user console home admin to contain options. - Worked on more stored procedures, including the creation of 'get' and 'set' calls - Continued the work on the generation of stored procedures within MySQL database	R F	R F	R F
11/23/22	10:00 AM	11:00 AM	- Full stack development: - Began development of Advisement and user console home pages. - Worked on the creation of stored procedures, including create and update procedures - Continued the work on the generation of stored procedures within MySQL database	S	S	S
11/24/22			- Full stack development: - Additions to Advisement and user console home pages. - Worked on the creation of stored procedures, including create and update procedures - Continued the work on the generation of stored procedures within MySQL database			
11/25/22	10:00 AM	11:00 AM	- Full stack development: - Restrictions to Advisement and dynamic components. - Worked on the creation of stored procedures, including create and update procedures - Continued the work on the generation of stored procedures within MySQL database	S	S	S
11/26/22			- Full stack development: - Refining Administrative Advisement and user console home pages.			

			- Worked on the creation of stored procedures, including create and update procedures - Continued the work on the generation of stored procedures within MySQL database			
11/27/22	10:00 AM	11:00 AM	- Full stack development: - Added Advisement, Personal Information, Semester Schedules, Researcher/Faculty/Student/Administrator navs, etc. - Revised all Stored Procedures made up until this point to ensure compatibility with the latest version of the database	S	S	S
11/28/22	10:00 PM	11:00 AM	- Full stack development: - Advisement, Semester Schedule, Master Schedule, other changes. - Began conversations about the documentation - Continued work with stored procedures, including nested stored procedures and 'checks' - Continued the work on the generation of stored procedures within MySQL database	S	S	S
11/29/22	12:00 PM	3:30 PM	- Full stack development: - Completed numerous View user constraints. - Added Room types into the database. - Generated new stored procedures based on new information presented by Prof. Gupta - Continued revision of all stored procedures made until this point - Out sick, but attended team meeting	R F	R F	R F
11/30/22	10:00 AM	11:00 AM	- Full stack development: - Completed Master Schedule and filtering of sections. - Began work on the compilation of all stored procedures into one complete file where they can be pulled from and used when necessary - Reworked some existing stored procedures to ensure compatibility with latest version of database	S	S	S
12/1/22	12:00 PM	3:30 PM	- Full stack development: - Completed section attendance for Faculty/Admin, and began other Course Section layouts. - Continued the work on the compilation of stored procedures - Finished the remaining stored procedures that needed to be made based on the information recently given to us by Prof. Gupta	R F	R F	R F
12/2/22	10:00 AM	11:00 AM	- Full stack development: - Fixed Advisement Console representing incorrect users. - A splash of color, bug fixes, Section Grades and Section Rosters. - Improved the side navigation bar further. - DBM: fixed missing rows in child Student tables. - Finished the compilation of stored procedures - Began assisting on the work being done on the	S	S	S

			Design Document			
12/3/22			- Full stack development: - Began development of attendance updating. - Added (self) Update Personal Information page and its respective database functionality. - Continued assisting the work done on the design document - Created a site map diagram that encompasses all web pages made on the front of as of this date -Create Design Document outline -Begin adding relevant information			
12/4/22	10:00 AM	11:00 AM	- Full stack development: - Improved (self) Update Personal Information page and its respective database functionality. - Continued assisting the work done on the design document - Developed a diagram based on the priority levels of different aspects of the project, based on the pyramid design -Continue working on design document	S	S	S
12/5/22	10:00 PM	11:00 AM	- Full stack development: - Added (self) Update Personal Information page and its respective database functionality. - Continued work on the design document, gathering screenshots of the website for use in other documentation.	S	S	S
12/6/22	12:00 PM	3:30 PM	- Full stack development: - Course Section add/drop (unchecked). Admin features expanded to control other users. - Continued work on design document, finished topics on high-level design and implementation	R F	R F	R F
12/7/22	10:00 AM	11:00 AM	- Full stack development: - Email through PHP; AWS SES verification. Password reset request ui. - Began work on debugging and fixing stored procedures for optimal usage -Continue working on design document	S	S	S
12/8/22	12:00 PM	3:30 PM	- Full stack development: - Create Program, Request password reset, AWS SES controls, etc - Continued work on fixing stored procedures - Altered create procedures in database - Begin Work on User Manual	R F	R F	R F
12/9/22	10:00 AM	11:00 AM	- Full stack development: - Delete Course Section now works and delivers a response. - Continued work on fixing stored procedures - Fixed stored procedures that create users and create programs - Continue Work on User Manual	S	S	S
12/10/22	10:00 AM	11:00 AM	- Full stack development: - Create Department finished, needs api call for available Faculty - Course Creation finished.	S	S	S

			- Enrollment is almost finished. - Began work on User Manual - Gathered screenshots and clips of website for use in User Manual - Continue Work on User Manual			
12/11/22	10:00 AM	11:00 AM	- Full stack development: - Pass resets, pass updates, create department/program/course/course-section, begin creating users. - Finished gathering of screenshots and began write up of User Manual - Continue Work on User Manual	S	S	S
12/12/22	10:00 PM	11:00 AM	- Full stack development: - Created Primary Administrator permissions and corrected priority-based access. - Advisement now shows all Advisors of a Student. - Completed basic display of prerequisite data in catalog's courses. - Bug fix for single/double enrollment. - User Creation completed; GUI not polished - Continue Work on User Manual	S	S	S
12/13/22	12:00 PM	3:00 PM	- Full stack development: - Registration is now blocked by unsatisfied prerequisites. Lacks user feedback. - Unofficial Transcript completed. - Tested registration to see if courses with unsatisfied requirements could be registered to - Continue Work on User Manual	S	S	S
12/14/22	10:00 AM	12:00 PM	- Full stack development: - Completed Degree Audit. Needs heading with completionist data. - Bug fixes, performance improvements, more color in Catalog and tables. - Created stored procedures for updating a course section - Updated application sitemap - Continued work on User Manual - Appended information to SRS and Design Document - Continue Work on User Manual - Update SRS - Test use cases in website	S	S	S
12/15/22	12:00 PM	4:00 PM	- Presented project to professor for the final time, received confirmation of grade	R F	R F	R F
12/16/22			- Finished updating documentation for final submission	S	S	S

Problem Statement

     The goal is to design and implement a web based dynamic registration system to be used at Aleutian Islands University. The system will be a multi-user system split into Students, Faculty, Administration and Research Staff. The system will be implemented in such a way that each of these users will have different levels of access to functions. Functions will provide access to data for user roles who have explicitly stated access to it. Since the system will be accessed by different types of users, there will be a variety of use cases.

Proposed Implementation

     The Website will be hosted through AWS (Amazon Web Services). We will use a free tier Ubuntu EC2 instance and **the Server** will also utilize many other AWS features for better functionality. Route 53 will be used for both domain registration and recording subdomains. Certificate Manager will be used to declare HTTPs certifications. Elastic IP will be used to preserve public access to the website address if the instance must be disconnected. Elastic Load Balancers, such as Application Load Balancers, will be used to redirect port 80 connections through port 443 for enforced HTTPS use. The server itself will make use of the LAMP stack (using Ubuntu, Apache, MySQL and PHP).

     The **Front-end** will use the React (ReactJs) framework. With React, we will use the language of Typescript (which is a strictly-typed derivative of Javascript).

     The **Back-end** will use API connections through MySQLi, which is a relational database driver that is used in PHP to provide an interface with the MySQL database. Security will be an important consideration, so the API will make use of Prepared Statements as well as Parameter and Result Bindings.

     This proposed implementation is, of course, subject to change and we will provide the necessary updates to the client if said changes occur.

Post Implementation Run Down

    The proposed web application was implemented in the way it was proposed, with a combination of the React framework and TypeScript for the front end, and API calls using MySQLi to the MySQL Database, PHP, and Stored Procedures in the back end. The system functions as specified in the below Use Cases, delivering a truly dynamic system for the client and users.

    The initial step towards implementation, and a system that aided us for the duration of this project, was the usage of Trello. Trello is a web-based, kanban-style, list-making application that we used for the organization and execution of tasks as a group. We began by separating the tasks at hand into one of six categories: Back-End, Database, Dev-Ops, Front-End, Queries and Documentation. This system aided in the breaking down of tasks for the individuals on the team, and deciding which tasks were done by those with a strong set of skills in that field. Team Lead Andrew Stephens was in charge of delegating the tasks to the other members of the team, Gwen Alessi and Brian Mejia.
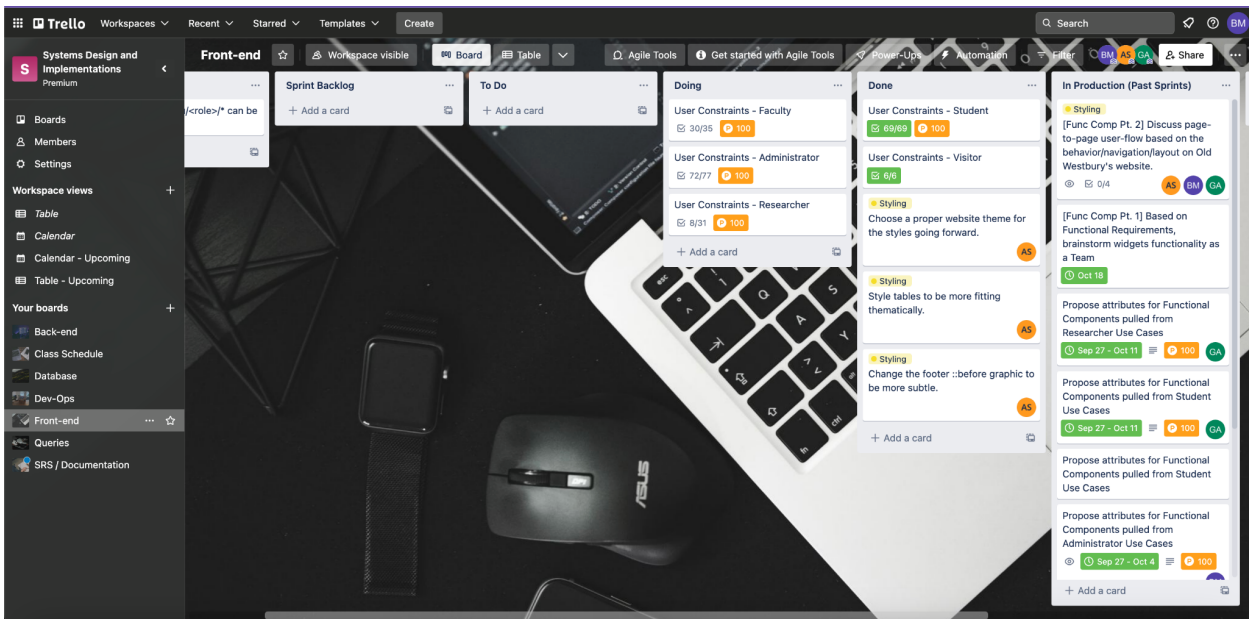
Image of the Trello Board used for the project

    After the splitting up of tasks, the following weeks were used to build the foundation of the web application, the database. The data that resides inside of it was generated primarily by hand, including student histories, course sections, advisor/advisee relationships, etc. Few was generated automatically, only personal information of the users such as street addresses, phone numbers and names. Course Catalog information was taken from

the SUNY Old Westbury website, with explicit permission from the client. MySQL was used as the Relational Database Manager

Once the data was generated, tested and finalized, the website was next in production. The website was hosted as stated above, using Amazon Web Services and the free-tier Ubuntu EC2 instance that they provide. We also used Elastic IP to ensure that access to the website was preserved even if the instance needed to be restarted for whatever reason. The server used the LAMP Stack as stated above for creation of the web application.

EC2 > Instances > i-07f0f0e2007da5c9f

Instance summary for i-07f0f0e2007da5c9f (SysDesign) Info
Updated less than a minute ago

C | Connect | Instance state ▼ | Actions ▼

Instance ID	Public IPv4 address	Private IPv4 addresses
i-07f0f0e2007da5c9f (SysDesign)	52.86.101.40 | open address	172.31.22.195
IPv6 address	Instance state	Public IPv4 DNS
–	⊘ Running	ec2-52-86-101-40.compute-1.amazonaws.com | open address
Hostname type	Private IP DNS name (IPv4 only)	
IP name: ip-172-31-22-195.ec2.internal	ip-172-31-22-195.ec2.internal	
Answer private resource DNS name	Instance type	Elastic IP addresses
IPv4 (A)	t2.micro	52.86.101.40 [Public IP]
Auto-assigned IP address	VPC ID	AWS Compute Optimizer finding
–	vpc-0b6ce069cc3b819fd	⊗ User: arn:aws:iam::655777653223:user/Brian is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * because no identity-based policy allows the compute-optimizer:GetEnrollmentStatus action Retry
IAM Role	Subnet ID	Auto Scaling Group name
–	subnet-03d9e7406f7c1bbc6	–

An image of the AWS instance hosting our website

Use case implementation followed, which led us through the ending of the allotted time for this project. Use Cases were handled by both React, and its ability to dynamically change components that have been generated depending on the data that is being returned, and MySQL stored procedures, which allows for the execution of SELECT, INSERT, UPDATE, and DELETE statements quickly and efficiently from the front end. Both required and forbidden functionalities were handled with this combination.

Were there more time, the group would have implemented another semester of data beyond the Spring 2023 semester, as the system is truly dynamic from top to bottom. Very little to no hard coding was done on either the front end or back end of the system to ensure that this web application could be used for any upcoming semester. Other desired additions would be the hosting of an email server on our website, for easier communication of events between different users. Overall, from this project we learned valuable time management skills, the process of implementing a website from the ground up and the focusing on individual strengths that contribute to the group's goals.

User Constraints

Visitor Constraints

Explicitly Required
View Home Pages

Explicitly Forbidden
-

Client Silent Required
View available Majors / Minors
View Major / Minor Requirements
Search the Course Catalog

Client Silent Forbidden
Cannot be successfully authenticated during login

Student Constraints

Explicitly Required
Be either Undergraduate or Graduate
Be either Full Time or Part Time
View Home Pages
View Login Page
View their own Personal Information
View Advisor
View Degree Audit
View Semester Schedule for Fall 2022
View Semester Schedule for Spring 2023
View Holds
View Semester Grades
View Unofficial Transcript
View available Majors
View available Minors
View Major Requirements
View Minor Requirements
Search Master Schedule for Fall 2022
Search Master Schedule for Spring 2023
Search the Course Catalog by Course name
Search the Course Catalog by Day
Search the Course Catalog by Department name
Search the Course Catalog by Faculty name
Search the Course Catalog by Period
Search the Course Catalog by Room number
Search a Course's Prerequisites
Add Major
Add Minor
Add Course Section
Change Major
Change Minor
Change Account Password
Drop Course Section
Update Personal information

Explicitly Forbidden
Cannot Add Course Section with a Hold
Cannot Re-add Course Section if Student already has a passing grade
Cannot Add Course Section if prerequisites are not met
Cannot Add Course Section if there is a time conflict with another course
Cannot Add Course Section for a particular course more than once within a particular semester
Cannot Add Course Section if course credits exceeds 16 credit limit for Full-Time
Cannot Add Course Section if course credits exceeds 11 credit limit for Part-Time
Cannot Add Course Section if no seats in section are available
Cannot Add Course Section for another student
Cannot Add a Course Section outside time window
Cannot Drop a Course Section outside time window
Cannot Declare more than 2 Majors and Minor declarations combined
Cannot Declare more than 1 Minor

Cannot Declare 1 Minor without a Major
Cannot Declare less than 1 Major
Cannot Declare more than 2 Majors

Client Silent Required
-

Client Silent Forbidden
Cannot Be both Full Time and Part Time
Cannot View Course Section roster
Cannot View unassigned Course Sections
Cannot View another Student's Personal Information
Cannot View another Student's Advisor
Cannot View another Student's Degree Audit
Cannot View another Student's Holds
Cannot View another Student's Semester Grades
Cannot View another Student's Semester Schedules that's outside of their own account
Cannot View another Student's Unofficial Transcript
Cannot View Student Semester Schedules that's before Fall 2022 Semester
Cannot View Student Semester Schedules that's after Spring 2023 Semester
Cannot Change another Student's Major
Cannot Change another Student's Minor
Cannot Change another User's Password
Cannot Change their new Password to the existing Password
Cannot Change their Password without fulfilling character and length requirements
Cannot Drop another Student from a Course Section
Cannot Drop from an unenrolled Course Section
Cannot Update Personal Information that is unsafe to change
Cannot Update Personal Information that is identifiable

Faculty Constraints

Explicitly Required
Be Assigned to Teach Course Sections
Be either Full Time or Part Time
Be Assigned to Advise Students
Belong to one or more Departments
View Home Pages
View Login Page
View their own Personal Information
View a Student's Advisor
View a Student's Degree Audit
View a Student's Holds
View a Student's Semester Grades
View a Student's Semester Schedules
View a Student's Unofficial Transcript
View their own Semester Schedules
View available Majors / Minors
View Major / Minor Requirements
View their own Course Section Roster
View their own Course Section Attendance
View their own Advisees
Search a Semester's Master Schedule
Search the Course Catalog
Change Account Password
Update Personal information
Update their Course Section Attendance
Assign Course Section with Grades

Explicitly Forbidden
Cannot Assign Course Section with Grades outside their own Course Sections
Cannot Assign Course Section with Grades outside time period
Cannot Update their own Course Section Attendance outside current day

Client Silent Required
View all Advisors
Attempt to Login

Client Silent Forbidden
Cannot View Personal Information that's above Faculty role
Cannot View Course Section roster outside of the current Semester
Cannot View unassigned Course Sections
Cannot Change another User's Password
Cannot Change their new Password to the existing Password
Cannot Change their Password without fulfilling character and length requirements
Cannot Update Personal Information that is unsafe to change
Cannot Update Personal Information that is identifiable

Administrator Constraints

Explicitly Required (A Secondary Administrator Can Only View):
Has Primary Level Security clearance
Be Full Time
View Login Page
View Home Pages
View anyone's Semester Schedules
View a Student's Advisor
View a Student's Degree Audit
View a Student's Holds
View a Student's Semester Grades
View a Student's Unofficial Transcript
View any Course Section's Course Section Roster
View any Course Section's Course Section Attendance
View available Majors
View available Minors
View Major Requirements
View Minor Requirements
View Advisees or any other Student
Search a Semester's Master Schedule
Search the Course Catalog
Add a Student to Course Section
Add a Student to Major
Add a Student to Minor
Assign a Student Grades
Change any Account Password
Change a Student Major
Change a Student Minor
Create a new Course Section
Create a new Major
Create a new Minor
Create a new User
Create the Master Schedule
Delete a Student from Course Section
Update the Master Schedule
Update Personal information

Explicitly Forbidden:
Cannot View any Semester Schedule if requested Semester is before Fall 2022
Cannot View any Semester Schedule if requested Semester is after Spring 2023
Cannot Add Student to Course Section if Student has a Hold
Cannot Re-Add Student to Course Section if Student already has a passing grade
Cannot Add Student to Course Section if prerequisites are not met
Cannot Add Student to Course Section if registered to another course with time conflict
Cannot Add Student to Course Section more than once within a particular semester
Cannot Add Student to Course Section if course credits exceeds 16 credit limit for Full-Time
Cannot Add Student to Course Section if course credits exceeds 11 credit limit for Part-Time
Cannot Add Student to Course Section if no seats in section are available
Cannot Add Student to Course Section for another Student
Cannot Add Student to Course Section outside time window

Cannot Declare a Student to have more than 2 Majors and Minor declarations combined
Cannot Declare a Student to have more than 1 Minor
Cannot Declare a Student to have 1 Minor without a Major
Cannot Declare a Student to have less than 1 Major
Cannot Declare a Student to have more than 2 Majors
Cannot Create a new Course Section Record if there is time/date conflict with another Section
Cannot Create a new Course Section Record if there is Faculty part-time/full-time status conflict
Cannot Create a new Course Section Record if there is conflict with a Faculty's department
Cannot Create a Master Schedule Record if a Faculty is not in related department
Cannot Create a Master Schedule Record if a Faculty has time conflicts
Cannot Create a Master Schedule Record if a Room has time conflicts
Cannot Create a Master Schedule Record if a Semester is ongoing or has passed
Cannot Assign Course Section with Grades outside Faculty's own Course Sections
Cannot Assign a Course Section with Grades for future Semesters after Fall 2022

Client Silent Required:
Be in one or more Departments
Attempt to Login
View all Advisors
Add Student to Advisor
Update a Course's catalog information

Client Silent Forbidden:
Cannot Be both Part Time and Full Time
Cannot Search a Semester's Master Schedule if requested Semester is before Fall 2022
Cannot Search a Semester's Master Schedule if requested Semester is after Spring 2022
Cannot Add Student to Advisor if Student has an Advisor
Cannot Add Student to Advisor if Advisor does not share a Department with Student
Cannot Change anyone's Password outside of their own account
Cannot Change anyone's Password to the existing Password
Cannot Change anyone's Password without fulfilling character and length requirements
Cannot Drop Student from Course Section if Student does not exist in the Section
Cannot Update anyone's Personal Information that is unsafe to change or that is identifiable

<u>Researcher Constraints</u>

<u>Explicitly Required:</u>
View Login Page
View their own Personal Information
View Collective Statistics
Search a Semester's Master Schedule
Search the Course Catalog
Change their own Account Password
Update their own Personal information

<u>Explicitly Forbidden:</u>
Cannot View anyone else's personally identifying information anywhere
Cannot View Course Section roster statistics outside of Fall 2022
Cannot View unassigned Faculty Course Sections

<u>Client Silent Required:</u>
Attempt to Login

<u>Client Silent Forbidden:</u>
Cannot Change their Password outside of their own account
Cannot Change their Password to the existing Password
Cannot Change their Password without fulfilling character and length requirements
Cannot Update their Personal Information that is unsafe to change or that is identifiable
Cannot Search a Semester's Master Schedule if requested Semester is neither current nor next
Semester

Use Cases

Use Case Visitor

Use Case:	View Home Pages
Actors:	Visitor
Initial Condition:	1. The visitor will land on the home page
Basic Flow:	1. The visitor will visit the home page 2. The visitor will inspect the homepages 3. The visitor will leave the website
Exit Condition:	1. The visitor leaves the website

Use Case:	View Login Page
Actors:	Visitor
Initial Condition:	1. The visitor will land on the home page
Basic Flow:	1. The visitor will arrive at the home page 2. The visitor will proceed to the login page 3. The visitor will leave the website
Exit Condition:	1. The visitor leaves the website

Use Case:	Attempt to Login
Actors:	Visitor
Initial Condition:	1. The visitor will land on the home page
Basic Flow:	1. The visitor will arrive at the home page 2. The visitor will proceed to the login page 3. The visitor will attempt to login 4. The visitor will exit the website
Exit Condition:	1. The visitor leaves the website

Use Case:	Search Course Catalog
Actors:	Visitor
Initial Condition:	1. The Visitor will land on the home page
Basic Flow:	1. The Visitor will navigate from the home page to the "Search Course Catalog" tab 2. The Visitor will search the catalog with the desired parameters
Exit Condition:	1. The visitor will leave the website

Use Case:	View Course Catalog
Actors:	Visitor
Initial Condition:	1. The Visitor will land on the home page
Basic Flow:	2. The Visitor will navigate from the home page to the "Academics" tab 3. The Visitor will select "Press for catalog" 4. The Visitor will inspect the desired information from the course catalog
Exit Condition:	5. The visitor will leave the website

Use Case:	View Master Schedule
Actors:	Visitor
Initial Condition:	1. The Visitor will land on the home page
Basic Flow:	2. The Visitor will navigate from the home page to the "Academics" tab 3. The Visitor will select "Press for schedule" 4. The Visitor will inspect the desired information from the Master Schedule
Exit Condition:	5. The visitor will leave the website

Use Case:	View About Page
Actors:	Visitor
Initial Condition:	1. The Visitor will land on the home page
Basic Flow:	2. The Visitor will navigate from the home page to the "About" tab 3. The Visitor will inspect the desired information from the About page
Exit Condition:	4. The Visitor exits the website

Use Case Student

Use Case:	View Advisor
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. From the Student Home Page, navigate to "View the Advisement Page" 2. View Advisor
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View Personal Account Information
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. When Student logs on, the landing page is the Profile 2. Navigate through "About" component 3. Inspect personal information
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View Personal Semester Schedule
Actors:	Student
Initial Condition:	1. User logs in with Student Credentials
Basic Flow:	1. From the Student home page, navigate to the "Student Console" page 2. Select the "Registration" page 3. Navigate to "Semester Schedule" 4. Inspect personal semester schedule
Exit Condition:	1. User logs out of account 2. Web browser closes

Use Case:	View Academics Page
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Academics Page"
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View Grades
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Academics Page" 2. Navigate to "Semester Grades" 3. View Grades
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View Holds
Actors:	Student
Initial Condition:	2. User logs in to website using Student Credentials
Basic Flow:	4. From the Student Home Page, Navigate to "Registration" 5. Navigate to "Holds" 6. View Holds
Exit Condition:	3. User logs out of account 4. Web browser is closed

Use Case:	View Degree Audit
Actors:	Student
Initial Condition:	1. User logs in to website with Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Academics" 2. Navigate to "Degree Audit" 3. View Degree Audit
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View Unofficial Transcript
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Academics" 2. Navigate to "Unofficial Transcript" 3. View Unofficial Transcript
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View Registration Page
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Registration" 2. View Registration Page
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View Registered Course Sections
Actors:	Student
Initial Condition:	1. User logs to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Registration" 2. Navigate to "Semester Schedule" 3. View Registered Course Sections
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View Master Schedule
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Registration" 2. Navigate to "Master Schedule" 3. View Master Schedule
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View Available Majors/Minors
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Course Catalog" 2. Navigate to "Programs" 3. View Available Majors/Minors
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View Majors/Minors Requirements
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Course Catalog" 2. Navigate to "Programs" 3. Select "View" on a Program 4. View Major/Minor Requirements
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View a Course's Prerequisites
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Course Catalog" 2. Navigate to "Courses" 3. Select "View" on a Course 4. View a Course's Prerequisites
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Search Department by School
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Course Catalog" 2. Navigate to "Departments" 3. Search Department by School
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Search Master Schedule by Period
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	2. From the Student Home Page, Navigate to "Registration" 3. Navigate to "Master Schedule" 4. Search Period
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Search Master Schedule by Day
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	2. From the Student Home Page, Navigate to "Registration" 3. Navigate to "Master Schedule" 4. Search Day
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Search Master Schedule by Professor
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Registration" 2. Navigate to "Master Schedule" 3. Search Professor
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Search Master Schedule by Course Name
Actors:	Student
Initial Condition:	1. User logs in to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Registration" 2. Navigate to "Master Schedule" 3. Search Course Name
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Update Personal Account Information
Actors:	Student
Initial Condition:	1. User logs in with Student Credentials
Basic Flow:	1. From the Student home page, the initial view is the "Profile" 2. Select "Edit" button 3. Make necessary updates to information 4. Save updates
Exit Condition:	1. Administrator makes updates to personal account information 2. User logs out of account 3. Web browser closes

Use Case:	Update Personal Account Password
Actors:	Student
Initial Condition:	1. User logs in with Student Credentials
Basic Flow:	1. From the Student home page, navigate to "Account" 2. Make necessary changes to password 3. Save updates
Exit Condition:	1. Administrator makes change to personal password 2. User logs out of account 3. Web browser closes

Use Case:	Unsuccessfully Change Password
Actors:	Student
Initial Condition:	1. The Student attempts to change a password to a new one that does not meet the requirements outlined for passwords
Basic Flow:	The Student tries to change a user's password The system rejects any change to this information and informs the Student that the password does not meet the necessary requirements
Exit Condition:	1. The Student leaves the website 2. The Student tries again with another combination of characters

Use Case:	Unsuccessfully Change Password
Actors:	Student
Initial Condition:	1. The Student attempts to change a password to the same password
Basic Flow:	1. The Student tries to change a user's password to the same password 2. The system rejects any change to this information and informs the Student that the password is the same as current password
Exit Condition:	1. The Student leaves the website 2. The Student tries again with another combination of characters

Use Case:	Add Registration for Course Sections
Actors:	Student
Initial Condition:	1. User logs to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Registration" 2. Navigate to "Course Registration" 3. Select appropriate Semester for registration 4. Select "Course Search" 5. View Master Schedule 6. Select "Add" on a Course 7. Register for Course
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Register for a Course Section
Actors:	Student
Initial Condition:	1. Cannot Register for same section more than once in a semester
Basic Flow:	1. Attempt to Register for Course Section 2. User is already registered for the selected Course Section this semester 3. Course Section is not added
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Register for a Course Section
Actors:	Student
Initial Condition:	1. Cannot Register same Course more than once in a Semester
Basic Flow:	1. Attempt to Register for Course 2. User is already registered for the selected Course this semester 3. Course is not added
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Register for a Course Section
Actors:	Student
Initial Condition:	1. Does not have the requirements/ prerequisites
Basic Flow:	1. Attempt to Register for Course 2. Student does not meet prerequisites/requirements for selected course 3. Course is not added
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Register for a Course Section
Actors:	Student
Initial Condition:	1. Course Has already been taken with a satisfying grade
Basic Flow:	1. Attempt to Register for Course 2. Student has previously completed course with a satisfying grade 3. Course is not added
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Register for a Course Section
Actors:	Student
Initial Condition:	1. Has Hold on Account
Basic Flow:	1. Attempt to Register for Course 2. Student has a hold on their Account 3. Course is not added
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Register for a Course Section
Actors:	Student
Initial Condition:	1. Course Time Conflict
Basic Flow:	1. Attempt to Register for Course 2. Student is already Registered for a course at the selected time 3. Course is not added
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Register for a Course Section
Actors:	Student
Initial Condition:	1. Cannot Register for Course Section if no seats available
Basic Flow:	1. Attempt to Register for Course 2. Selected course is already full, no seats available 3. Course is not added
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Register for a Course Section
Actors:	Student
Initial Condition:	1. Cannot Register for Course outside of time window
Basic Flow:	1. Attempt to Register for Course 2. Selected course is already full, no seats available 3. Course is not added
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Remove Registration of Course Section
Actors:	Student
Initial Condition:	1. User logs to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Registration" 2. Navigate to "Course Registration" 3. Select "Drop Section" on a Course Section 4. Select "Apply Changes" 5. Remove Course
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Add Major / Minors AND add Student to Advisor
Actors:	Student
Initial Condition:	1. User logs to website using Student Credentials
Basic Flow:	1. From the Student Home Page, Navigate to "Registration" 2. Navigate to "Program Enrollment" 3. Select one or more Programs 4. Add Major/Minor 5. Add Advisor to Student 6. Save Changes
Exit Condition:	3. User logs out of account 4. Web browser is closed

Use Case:	Unsuccessfully Remove a Major
Actors:	Student
Initial Condition:	1. Tries to remove more than 2 Majors
Basic Flow:	1. Attempts to Remove any combination of Major/Minor outside of the following allowables: (Can have up to 2 majors, cannot have zero majors, combination of 1 major with 1 minor) 2. Error message, system requires you to clarify qualifying Major/Minor
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Remove a Major
Actors:	Student
Initial Condition:	1. Tries to remove a Major while keeping 1 Minor
Basic Flow:	1. Attempts to Remove any combination of Major/Minor outside of the following allowables: (Can have up to 2 majors, cannot have zero majors, combination of 1 major with 1 minor) 2. Error message, system requires you to clarify qualifying Major/Minor
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Remove a Minor
Actors:	Student
Initial Condition:	1. Tries to remove a Minor
Basic Flow:	1. Does not have a Minor to remove 2. Error message, system requires you to clarify qualifying Major/Minor
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Add a Major
Actors:	Student
Initial Condition:	1. Tries to add more than 2 Majors
Basic Flow:	1. Attempts to add any combination of Major/Minor outside of the following allowables: (Cannot add more than 2 majors, cannot have zero majors, combination of 1 major with 1 minor) 2. Error message, system requires you to add qualifying Major/Minor
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Add a Major
Actors:	Student
Initial Condition:	1. Tries to add 2 Majors with 1 existing Minor
Basic Flow:	1. Attempts to add any combination of Major/Minor outside of the following allowables: (Cannot add more than 2 majors, cannot have zero majors, combination of 1 major with 1 minor) 2. Error message, system requires you to add qualifying Major/Minor
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Add a Minor
Actors:	Student
Initial Condition:	1. Tries to add Minor with 2 existing Majors
Basic Flow:	1. Attempts to add any combination of Major/Minor outside of the following allowables: (Cannot add more than 2 majors, cannot have zero majors, combination of 1 major with 1 minor) 2. Error message, system requires you to add qualifying Major/Minor
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case Faculty

Use Case:	View Faculty Page
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. The User lands at the Faculty Home page
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Advisement Page
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Faculty Console" 2. Navigate to "Advisement" 3. View the Advisement options
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Faculty's Advisees
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Faculty Consolee" 2. Navigate to "Advisement" 3. Select "View Advisees" 4. Select "Your Advisees" 5. View Faculty's Advisees
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View All Advisees
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Faculty Consolee" 2. Navigate to "Advisement" 3. Select "View Advisees" 4. Select "All Advisees" 5. View All Advisees
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View All Advisors
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Faculty Consolee" 2. Navigate to "Advisement" 3. Select "View Advisors" 4. View all Advisors
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Student Degree Audit
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Advisement" 2. Navigate to the "Your Advisees" or "All Advisees" 3. Select a Student to view 4. Navigate to "Degree Audit" on behalf of Student 5. View the Student Degree Audit
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Student Personal Information
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Advisement" 2. Navigate to the "Your Advisees" or "All Advisees" 3. Select a Student to view 4. Navigate to "Personal Information" 5. View the Student Personal Information
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Student Holds
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Advisement" 2. Navigate to the "Your Advisees" or "All Advisees" 3. Select a Student to view 4. Navigate to "Holds" 5. View the Student Holds
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Student Semester Grades
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Advisement" 2. Navigate to the "Your Advisees" or "All Advisees" 3. Select a Student to view 4. Navigate to "Semester Grades" 5. View the Student Semester Grades
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Student Semester Schedule
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Advisement" 2. Navigate to the "Your Advisees" or "All Advisees" 3. Select a Student to view 4. Navigate to "Semester Schedule" 5. View the Student Semester Schedule
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Student Unofficial Transcript
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Advisement" 2. Navigate to the "Your Advisees" or "All Advisees" 3. Select a Student to view 4. Navigate to "Unofficial Transcript" 5. View the Student Unofficial Transcript
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Faculty Course Sections
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to "Faculty Console" 2. Navigate to the "Semester Schedule" 3. View the Course Section options
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Course Section Roster
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to "Faculty Console" 2. Navigate to the "Semester Schedule" 3. Select "View" on a course section 4. View the Course Section Roster options
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Course Section Attendance
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to "Faculty Console" 2. Navigate to the "Semester Schedule" 3. Select "View" on a course section 4. View the Course Section Attendance options
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	Assign Students with Grades
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Faculty Console" 2. Navigate to the "Educator Console" 3. Navigate to the "Semester Schedule" 4. Select Course Section to view 5. Locate "Roster" component 6. Update the Student Grades
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Assign a Student with a Grade
Actors:	Faculty
Initial Condition:	1. Faculty attempts to change Student Grade outside of Fall 2022
Basic Flow:	1. From the Faculty Home page, navigate to the "Manage Course Sections" page 2. Navigate to the "Update Course Sections" page 3. Navigate to the "Assign Students with Grades" page 4. Update the Student Grades 5. Time window for adding or updating grades has expired
Exit Condition:	1. Send Request to Administrator for grade change 2. Administrator logs out of account 3. Web browser is closed

Use Case:	Assign Attendance
Actors:	Faculty
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Faculty home page, navigate to "Faculty Console" 2. Navigate to "All Users" or "Advisement" 3. Navigate to the "Semester Schedule" 4. Select Course Section to view 5. Locate "Attendance" component 6. Assign Attendance 7. Save changes
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	View Personal Account Page
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Account" 2. View the Personal Account options
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Update a Personal Account Password
Actors:	Faculty
Initial Condition:	1. Requested password change does not match password requirements
Basic Flow:	1. From the Faculty Home page, navigate to "Account" 2. Change Personal Account password 3. Fail to successfully submit new password
Exit Condition:	1. Reattempt password update 2. Administrator logs out of account 3. Web browser is closed

Use Case:	Unsuccessfully Update a Personal Account Password
Actors:	Faculty
Initial Condition:	1. Requested password matches the old password
Basic Flow:	1. From the Faculty Home page, navigate to "Account" 2. Change Personal Account password 3. Fail to successfully submit new password
Exit Condition:	1. Reattempt password update 2. Administrator logs out of account 3. Web browser is closed

Use Case:	Update Personal Account Password
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Account" 2. Change Personal Account password
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Personal Information
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Profile" 2. Navigate through "About" component 3. View the Personal Account information
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	Update Personal Information
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Profile" 2. Select "Edit" 3. Update the Personal Information
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Personal Semester Schedules
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to "Faculty Console" 2. Navigate to the "Semester Schedule" 3. View the Personal Semester options
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Master Schedule Page
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Faculty Console" 2. Navigate to the "Educator Console" 3. Navigate to "Master Schedule" 4. View the Master Schedule options
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Available Majors / Minors
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Course Catalog" 2. Navigate to the "Programs" 3. View the Available Majors/Minors
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Major / Minor Requirements
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Course Catalog" 2. Navigate to the "Programs" 3. View the Available Majors/Minors 4. Select "View" on a Program 5. View Major/Minor Requirements
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Courses
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Course Catalog" 2. Navigate to the "Courses" 3. View the View Courses
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Course Prerequisites
Actors:	Faculty
Initial Condition:	1. User logs in to website using Faculty Credentials
Basic Flow:	1. From the Faculty Home page, navigate to the "Course Catalog" 2. Navigate to the "Courses" 3. View the View Courses 4. Select "View" on a Course 5. View Prerequisites
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case Administrator

Use Case:	Create the Master Schedule
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Creator Sandbox" 2. Select "Create Course Section" 3. Fill in all necessary information to create Course Section 4. Save changes 5. Course Section is added to the Master Schedule
Exit Condition:	1. Administrator creates new master schedule 2. Administrator logs out of account 3. Web browser closes

Use Case:	Update the Master Schedule
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Creator Sandbox" 2. Select "Create Course Section" 3. Fill in all necessary information to create Course Section 4. Save changes 5. Course Section is added to the Master Schedule
Exit Condition:	1. Administrator makes updates to the master schedule 2. Administrator logs out of account 3. Web browser closes

Use Case:	View Personal Account Information
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Profile" 2. View personal information
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	Update Personal Account Information
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "Profile" 2. Select "Edit" button 3. Update Personal Information 4. Save updates
Exit Condition:	1. Administrator makes updates to personal account information 2. Administrator logs out of account 3. Web browser closes

Use Case:	Update Personal Account Password
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "Account" 2. Make necessary updates to password 3. Save updates
Exit Condition:	1. Administrator makes updates to personal password 2. Administrator logs out of account 3. Web browser closes

Use Case:	View Course Catalog
Actors:	Administrator
Initial Condition:	1. User logs in to website using Administrator Credentials
Basic Flow:	1. From the Administrator Home Page, navigate to "Course Catalog" 2. Inspect the Course Catalog
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	Update a course's catalog information
Actors:	Administrator
Initial Condition:	1. User logs in to website using Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "View Course Catalog" page 2. Select "Update Course Catalog" tab 3. Select Course whose information is going to be updated 4. Save changes to Course
Exit Condition:	1. Administrator updates course information 2. Administrator logs out of account 3. Web browser is closed

Use Case:	Unsuccessfully Update a Course Section
Actors:	Administrator
Initial Condition:	1. Administrator attempts to update a course section that has not been created
Basic Flow:	1. The Administrator will fill out information for the course they want to update 2. The website will detect that the information entered does not correspond with a course section listing 3. The Administrator will be notified that the course does not exist in the system
Exit Condition:	1. The Administrator leaves the website

Use Case:	Create a new course section
Actors:	Administrator
Initial Condition:	1. User logs in to website using Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "Admin Console" 2. Navigate to "Sandbox" 3. Select "Create Course Section" 4. Enter Relevant Information 5. Save changes made
Exit Condition:	1. Administrator creates new course section 2. Administrator logs out of account 3. Web browser is closed

Use Case:	Unsuccessfully Create a Course Section
Actors:	Administrator
Initial Condition:	1. Administrator attempts to create a course section that is already created
Basic Flow:	1. The Administrator will fill out information for this new course section they want to create 2. The website will detect that the information entered already corresponds with a course section listing 3. The Administrator will be notified that the course already exists in the system
Exit Condition:	1. The Administrator leaves the website

Use Case:	Delete a Course Section
Actors:	Administrator
Initial Condition:	1. User logs in to website using Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "Manage Master Schedule" 2. Select "Delete" on a Course Section 3. Save changes made
Exit Condition:	1. Administrator deletes a course section 2. Administrator logs out of account 3. Web browser is closed

Use Case:	Unsuccessfully Delete a Course Section
Actors:	Administrator
Initial Condition:	1. Administrator attempts to delete a course section has not been created
Basic Flow:	1. The Administrator will try and find a course to delete that has not been made 2. The system will not display said course because it has not been created
Exit Condition:	1. The Administrator leaves the website

Use Case:	Search the Course Catalog
Actors:	Administrator
Initial Condition:	1. User logs in to website using Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "Course Catalog" 2. From there, select "Search Course Catalog" tab 3. Search the catalog using the desired parameters of the course
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	Search the Course Catalog by Department
Actors:	Administrator
Initial Condition:	1. User logs in to website using Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Course Catalog" 2. Select "Departments" 3. Search the catalog by School
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Available Majors
Actors:	Administrator
Initial Condition:	1. User logs in to website using Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Course Catalog" 2. Select "Programs" 3. Inspect the listed Majors
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Available Minors
Actors:	Administrator
Initial Condition:	1. User logs in to website using Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Course Catalog" 2. Select "Programs" 3. Inspect the listed Minors
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Major Requirements
Actors:	Administrator
Initial Condition:	1. User logs in to website using Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "Course Catalog" 2. Navigate to Programs 3. Select a Program to view 4. View Major Requirements
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Minor Requirements
Actors:	Administrator
Initial Condition:	1. User logs in to website using Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "Course Catalog" 2. Navigate to Programs 3. Select a Program to view 4. View Minor Requirements
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	Search a Course's Prerequisites
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "View Course Catalog" page 2. Select the "Search Course Catalog" 3. From there, select the course whose prerequisites need to be known
Exit Condition:	1. Administrator logs out of account 2. Web Browser is closed

Use Case:	Create new Program
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Sandbox" 3. Navigate to "Create Program" 4. Fill in information need for new Program 5. Save changes made
Exit Condition:	1. Administrator creates new major 2. Administrator logs out of account 3. Web browser is closed

Use Case:	Unsuccessfully Create a Program
Actors:	Administrator
Initial Condition:	1. Administrator attempts to create a Major that already exists
Basic Flow:	1. The Administrator will fill out information for this new Program they want to create 2. The website will detect that the information entered already corresponds with a Program listing 3. The Administrator will be notified that the Program already exists in the system
Exit Condition:	1. The Administrator leaves the website

Use Case:	Update a Major AND Update Advisor
Actors:	Administrator
Initial Condition:	1. User logs in to website using Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "View Course Catalog" page 2. Select "Manage Majors and Minors" tab 3. Select Major whose information is going to be updated 4. Select new Advisor 5. Save changes to Major
Exit Condition:	1. Administrator updates Major information 2. Administrator logs out of account 3. Web browser is closed

Use Case:	Update a Minor AND Update Advisor
Actors:	Administrator
Initial Condition:	1. User logs in to website using Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "View Course Catalog" page 2. Select "Manage Majors and Minors" tab 3. Select Minor whose information is going to be updated 4. Select new Advisor 5. Save changes to Minor
Exit Condition:	1. Administrator updates Major/Minor information 2. Administrator logs out of account 3. Web browser is closed

Use Case:	Unsuccessfully Update a Major
Actors:	Administrator
Initial Condition:	1. Administrator attempts to update a Major that has not been created
Basic Flow:	1. The Administrator will fill out information for the Major they want to update 2. The website will detect that the information entered does not correspond with a Major listing 3. The Administrator will be notified that the Major does not exist in the system
Exit Condition:	1. The Administrator leaves the website

Use Case:	Unsuccessfully Update a Minor
Actors:	Administrator
Initial Condition:	1. Administrator attempts to update a Minor that has not been created
Basic Flow:	1. The Administrator will fill out information for the Minor they want to update 2. The website will detect that the information entered does not correspond with a Minor listing 3. The Administrator will be notified that the Minor does not exist in the system
Exit Condition:	1. The Administrator leaves the website

Use Case:	View the Master Schedule
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "Manage Master Schedule" 2. Inspect all necessary information from the master schedule
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	Unsuccessfully Assign a Faculty to a Course Section
Actors:	Administrator
Initial Condition:	1. The Administrator attempts to assign a faculty member to a course section not in their department
Basic Flow:	1. The Administrator will select a faculty member to their new course section that is not a member of the same department that course section falls under 2. The system will inform the Administrator that this is not possible due to the conflict above
Exit Condition:	1. The Administrator leaves the website

Use Case:	Unsuccessfully Assign a Faculty to a Course Section
Actors:	Administrator
Initial Condition:	1. The Administrator attempts to assign a faculty member to a course that takes place during another course the faculty member is already teaching
Basic Flow:	1. The Administrator will select a faculty member to their new course section that is already assigned to another Course Section in that same time slot 2. The system will inform the Administrator that this is not possible due to the conflict above
Exit Condition:	1. The Administrator leaves the website

Use Case:	Unsuccessfully Assign a Course Section to a Room
Actors:	Administrator
Initial Condition:	1. The Administrator attempts to assign a course section to a room that is already occupied
Basic Flow:	1. The Administrator will select a room for their new course that is not available at that time slot 2. The system will inform the Administrator that this is not possible due to the conflict above
Exit Condition:	1. The Administrator leaves the website

Use Case:	Search a Semester's Master Schedule
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "Manage Master Schedule" 2. Search the master schedule with the desired parameters
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	Unsuccessfully Update the Master Schedule
Actors:	Administrator
Initial Condition:	1. The Administrator tries to update the master schedule when it hasn't been made yet
Basic Flow:	1. The Administrator will attempt to update the master schedule 2. The Administrator will get a message saying there is no master schedule created yet
Exit Condition:	1. The Administrator leaves the website

Use Case:	Unsuccessfully Update the Master Schedule
Actors:	Administrator
Initial Condition:	1. The Administrator tries to update the master schedule after the time period has expired for updates
Basic Flow:	1. The Administrator will attempt to update the master schedule 2. The Administrator will get a message saying the time period to update master schedule has expired
Exit Condition:	1. The Administrator leaves the website

Use Case:	View Advisors
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisors" 4. View Advisors
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	View Advisees or All Students
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to "Faculty Console" 2. Navigate to "Advisement" 3. Select "View Advisees" 4. View Advisees
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	View Faculty Semester Schedule
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisors" 4. Select a Faculty to view 5. Navigate to Semester Schedule 6. View Faculty Semester Schedule
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	View Course Section Roster
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisors" 4. Select a Faculty to view 5. Navigate to Semester Schedule 6. View Faculty Semester Schedule 7. Select a Course Section 8. View the Course Section Roster
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	View Course Section Grades
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisors" 4. Select a Faculty to view 5. Navigate to Semester Schedule 6. View Faculty Semester Schedule 7. Select a Course Section 8. View the Course Section Grades
Exit Condition:	1. Administrator logs out of account 2. Web browser is closed

Use Case:	View Course Section Attendance
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisors" 4. Select a Faculty to view 5. Navigate to Semester Schedule 6. View Faculty Semester Schedule 7. Select Course Section 8. View Course Section Attendance
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	Assign Faculty to Course Section
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Sandbox" 2. Navigate to "Create Course Section" 3. Select Course to Add to Master Schedule 4. Assign Faculty to Course Section 5. Save changes
Exit Condition:	1. Administrator assigns faculty to a course section 2. Administrator logs out of account 3. Web browser closes

Use Case:	Unsuccessfully Assign a Course Section to a Faculty
Actors:	Administrator
Initial Condition:	1. The Administrator attempts to more than 1 Course Section to a faculty that is part-time
Basic Flow:	1. The Administrator will attempt to add a faculty member to a Course Section 2. The system will inform the Administrator that the faculty member is part-time, therefore can not be assigned to Course Section
Exit Condition:	1. The Administrator leaves the website

Use Case:	Unsuccessfully Assign a Course Section to a Faculty
Actors:	Administrator
Initial Condition:	1. The Administrator attempts to more than 3 Course Sections to a faculty that is part-time
Basic Flow:	1. The Administrator will attempt to add a faculty member to a Course Section 2. The system will inform the Administrator that the faculty member is at his Course Section limit, therefore they can not be assigned to Course Section
Exit Condition:	1. The Administrator leaves the website

Use Case:	Unsuccessfully Assign a Course Section to a Faculty
Actors:	Administrator
Initial Condition:	1. The Administrator attempts to more than 1 Course Section to a faculty that is part-time
Basic Flow:	1. The Administrator will attempt to add a faculty member to a Course Section 2. The system will inform the Administrator that the faculty member already has a Course Section at that time slot, therefore can not be assigned to Course Section
Exit Condition:	1. The Administrator leaves the website

Use Case:	Unsuccessfully Update a the Attendance of a Course Section
Actors:	Administrator
Initial Condition:	1. The Administrator attempts to update the attendance after the target day period has passed
Basic Flow:	1. The Administrator will attempt to update the attendance on a day that is not the current day 2. The system will inform the Administrator that the attendance can no longer be updated because the day has passed
Exit Condition:	1. The Administrator leaves the website

Use Case:	Assign Grades (Midterm/Final)
Actors:	Administrator
Initial Condition:	User logs in with Administrator Credentials
Basic Flow:	8. From the Administrator home page, navigate to "Admin Console" 9. Navigate to "All Users" or "Advisement" 10. Select a Faculty to view 11. Navigate to the "Semester Schedule" 12. Select Course Section to view 13. Locate "Roster" component 14. Assign Grades to Students 15. Save changes
Exit Condition:	3. Administrator assigns grade 4. Administrator logs out of account 5. Web browser closes

Use Case:	Assign Attendance
Actors:	Administrator
Initial Condition:	2. User logs in with Administrator Credentials
Basic Flow:	16. From the Administrator home page, navigate to "Admin Console" 17. Navigate to "All Users" or "Advisement" 18. Select a Faculty to view 19. Navigate to the "Semester Schedule" 20. Select Course Section to view 21. Locate "Attendance" component 22. Assign Attendance 23. Save changes
Exit Condition:	6. Administrator assigns grade 7. Administrator logs out of account 8. Web browser closes

Use Case:	Unsuccessfully Assign a Grade
Actors:	Administrator
Initial Condition:	1. The Administrator attempts to assign a grade after Fall 2022
Basic Flow:	1. The Administrator will attempt to assign a grade after the said time window has passed 2. The system will inform the Administrator that the grade can not be assigned due to the time window being expired
Exit Condition:	1. The Administrator leaves the website

Use Case:	View Faculty Account Information
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisors" 4. Select a Faculty to view 5. View Faulty Account Information
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	View Faculty Account Semester Schedule
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisors" 4. Select a Faculty to view 5. Navigate to "Semester Schedule" 6. View Faculty Semester Schedule
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	Update Faculty Account Information
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisors" 4. Select a Faculty to view 5. View Faculty Account Information 6. Update Faculty Account Information 7. Save updates made
Exit Condition:	Administrator makes updates to faculty account information Administrator logs out of account Web browser closes

Use Case:	Change Faculty Account Password
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisors" 4. Select a Faculty to view 5. Navigate to Faculty Account 6. Change Faculty Account Password 7. Save Faculty Password
Exit Condition:	1. Administrator makes change to account password 2. Administrator logs out of account 3. Web browser closes

Use Case:	View Student Advisor
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Locate an Advisee 5. Locate "Advisor ID" Column 6. View the Student's Advisor
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	View Student Holds
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Holds"
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	View Student Unofficial Transcript
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Unofficial Transcript"
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	View Student Semester Grades
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Semester Grades"
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	View Student Semester Schedule
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Semester Schedule"
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	View Student Degree Audit
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Degree Audit"
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	Create a new User
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Sandbox" 3. Navigate to "Create User" 4. Navigate to "Create Student" or "Create Faculty" or "Create Admin" or "Create Researcher" 5. Fill in appropriate parameters 6. "Create" User
Exit Condition:	1. Administrator creates a new student list record 2. Administrator logs out of account 3. Web browser closes

Use Case:	Add Student to Major AND add Advisor
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Enrollment" 6. Add Student to a Major 7. Assign Advisor to Student 8. Save Changes
Exit Condition:	1. Administrator adds a student to a major or minor 2. Administrator logs out of account 3. Web browser closes

Use Case:	Add Student to Minor
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Enrollment" 6. Add Student to a Minor 7. Save Changes
Exit Condition:	1. Administrator adds a student to a major or minor 2. Administrator logs out of account 3. Web browser closes

Use Case:	Change Student Major AND add Advisor
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Enrollment" 6. Add Student to a different Major 7. Assign Advisor to Student 8. Save Changes
Exit Condition:	1. Administrator changes a students major or minor 2. Administrator logs out of account 3. Web browser closes

Use Case:	Change Student Minor AND add Advisor
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Enrollment" 6. Add Student to a different Minor 7. Assign Advisor to Student 8. Save Changes
Exit Condition:	1. Administrator changes a students major or minor 2. Administrator logs out of account 3. Web browser closes

Use Case:	Unsuccessfully Add or Change to Student Major
Actors:	Administrator
Initial Condition:	1. Administrator tried to add or change a student's major when they don't qualify to be in it
Basic Flow:	1. Administrator attempts to add or change a student's major 2. The system notifies the administrator that the student does not meet the qualifications necessary to be in that major
Exit Condition:	1. The Administrator leaves the website

Use Case:	Unsuccessfully Add or Change to Student Minor
Actors:	Administrator
Initial Condition:	1. Administrator tried to add or change a student's minor when they don't qualify to be in it
Basic Flow:	1. Administrator attempts to add or change a student's minor 2. The system notifies the administrator that the student does not meet the qualifications necessary to be in that minor
Exit Condition:	1. The Administrator leaves the website

Use Case:	Add Student Registration to Course Section
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Registration" 6. Select appropriate Semester 7. Navigate to "Course Search" 8. Select Course to Add 9. Changes saved
Exit Condition:	1. Administrator adds student registration to course section 2. Administrator logs out of account 3. Web browser closes

Use Case:	Unsuccessfully Add a Student to a Course Section
Actors:	Administrator
Initial Condition:	1. The Administrator is attempting to assign a student to a course section that conflicts with another course's time slot
Basic Flow:	1. The Administrator begins to assign a student to a new course section 2. The system will let the Administrator know that another course is occupying that time slot, therefore this course can not be added
Exit Condition:	1. The Administrator leaves the website

Use Case:	Unsuccessfully Add a Student to a Course Section
Actors:	Administrator
Initial Condition:	1. The Administrator is attempting to assign a graduate school student to a course section that is meant for undergraduates
Basic Flow:	1. The Administrator begins to assign a student to a new course section 2. The system will let the Administrator know that this course is for undergraduate students only, and therefore the student can not take it
Exit Condition:	1. The Administrator leaves the website

Use Case:	Unsuccessfully Add a Student to a Course Section
Actors:	Administrator
Initial Condition:	1. The Administrator is attempting to assign an undergraduate school student to a course section that is meant for graduates
Basic Flow:	1. The Administrator begins to assign a student to a new course section 2. The system will let the Administrator know that this course is for graduate students only, and therefore the student can not take it
Exit Condition:	1. The Administrator leaves the website

Use Case:	Drop Student Registration from Course Section
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Registration" 6. Select appropriate semester 7. Drop Course Section 8. Changes saved
Exit Condition:	1. Administrator deletes student registration to course section 2. Administrator logs out of account 3. Web browser closes

Use Case:	Unsuccessfully Drop a Student from a Course Section
Actors:	Administrator
Initial Condition:	1. The Administrator is attempting to delete a student from a course section that the student is not registered for
Basic Flow:	1. The Administrator searches for the student in the course section registration list 2. The student does not appear because they are not registered for the Course Section
Exit Condition:	1. The Administrator leaves the website

Use Case:	View Student Account Information
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "Profile" 4. View Account Information
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	View Student Account Semester Schedule
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Semester Schedule" 6. View Semester Schedule
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	Update Student Account Information
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Profile" 6. Update Student Account Information 7. Save changes
Exit Condition:	1. Administrator makes updates to student account information 2. Administrator logs out of account 3. Web browser closes

Use Case:	Change Student Account Password
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Advisement" 3. Navigate to "View Advisees" 4. Select to View a Student 5. Navigate to "Account" 6. Change Account Password 7. Save changes
Exit Condition:	1. Administrator changes student account password 2. Administrator logs out of account 3. Web browser closes

Use Case:	View Researcher Account Information
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Manage Users" 3. Filter for "Researcher" 4. Select Researcher 5. View Researcher Account Information
Exit Condition:	1. Administrator logs out of account 2. Web browser closes

Use Case:	Update Researcher Account Information
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Manage Users" 3. Filter for "Researcher" 4. Select Researcher 5. View Researcher Account Information 6. Update the needed information 7. Save updates
Exit Condition:	1. Administrator updates researcher account information 2. Administrator logs out of account 3. Web browser closes

Use Case:	Change Researcher Account Password
Actors:	Administrator
Initial Condition:	1. User logs in with Administrator Credentials
Basic Flow:	1. From the Administrator home page, navigate to the "Admin Console" 2. Navigate to "Manage Users" 3. Filter for "Researcher" 4. Select Researcher 5. View Researcher Account 6. Change Password 7. Save Changes
Exit Condition:	1. Administrator changes researcher account password 2. Administrator logs out of account 3. Web browser closes

Use Case:	Unsuccessfully Change a User's password
Actors:	Administrator
Initial Condition:	1. The Administrator attempts to change a password to a new one that does not meet the requirements outlined for passwords
Basic Flow:	1. The Administrator tries to change a user's password 2. The system rejects any change to this information and informs the administrator that the password does not meet the necessary requirements
Exit Condition:	1. The Administrator leaves the website

Researcher Use Case

Use Case:	View Personal Account Information
Actors:	Researcher
Initial Condition:	1. User logs in to website using Researcher Credentials
Basic Flow:	1. From the Researcher home page, navigate to "Profile" 2. View Researcher Account Information
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Update Personal Account Information
Actors:	Researcher
Initial Condition:	1. User logs in with Researcher Credentials
Basic Flow:	1. From the Researcher home page, navigate to the "Profile" 2. Select to update information 3. Make necessary updates to information 4. Save updates
Exit Condition:	1. Administrator makes updates to personal account information 2. Administrator logs out of account 3. Web browser closes

Use Case:	Change Personal Account Password
Actors:	Researcher
Initial Condition:	2. User logs in with Researcher Credentials
Basic Flow:	1. From the Researcher home page, navigate to the "Account" 2. Make necessary changes to password 3. Save updates
Exit Condition:	1. Administrator makes change to personal password 2. Administrator logs out of account 3. Web browser closes

Use Case:	Unsuccessfully Change a Account Password
Actors:	Researcher
Initial Condition:	1. The Researcher attempts to change a password to a new one that does not meet the requirements outlined for passwords
Basic Flow:	1. The Researcher tries to change a user's password 2. The system rejects any change to this information and informs the Researcher that the password does not meet the necessary requirements
Exit Condition:	1. The Researcher leaves the website 2. The Researcher tries again with another combination of characters

Use Case:	Unsuccessfully Change a Account Password
Actors:	Researcher
Initial Condition:	1. The Researcher attempts to change a password to the same password
Basic Flow:	1. The Researcher tries to change a user's password to the same password 2. The system rejects any change to this information and informs the Researcher that the password is the same as current password
Exit Condition:	1. The Researcher leaves the website 2. The Researcher tries again with another combination of characters

Use Case:	View Master Schedule
Actors:	Researcher
Initial Condition:	1. User logs in to website using Researcher Credentials
Basic Flow:	1. From the Researcher Home Page, Navigate to "Researcher Console" 2. Navigate to "Master Schedule" 3. View Master Schedule
Exit Condition:	3. User logs out of account 4. Web browser is closed

Use Case:	View Available Majors/Minors
Actors:	Researcher
Initial Condition:	1. User logs in to website using Researcher Credentials
Basic Flow:	1. From the Researcher Home Page, Navigate to "Researcher Console" 2. Navigate to "Course Catalog" 3. Navigate to "Programs" 4. View Programs
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View Majors/Minors Requirements
Actors:	Researcher
Initial Condition:	1. User logs in to website using Researcher Credentials
Basic Flow:	1. From the Researcher Home Page, Navigate to "Researcher Console" 2. Navigate to "Course Catalog" 3. Navigate to "Programs" 4. View Programs 5. Select a Major/Minor 6. View Major/Minor Requirements
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View a Course's Prerequisites
Actors:	Researcher
Initial Condition:	1. User logs in to website using Researcher Credentials
Basic Flow:	1. From the Researcher Home Page, Navigate to "Researcher Console" 2. Navigate to "Course Catalog" 3. Navigate to "Courses" 4. Select a Course to view 5. View Prerequisites
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Search Department
Actors:	Researcher
Initial Condition:	1. User logs in to website using Researcher Credentials
Basic Flow:	1. From the Researcher Home Page, Navigate to "Researcher Console" 2. Navigate to "Course Catalog" 3. Navigate to "Departments" 4. Search Departments by School
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Search Master Schedule by Period
Actors:	Researcher
Initial Condition:	1. User logs in to website using Researcher Credentials
Basic Flow:	1. From the Researcher Home Page, Navigate to "Researcher Console" 2. Navigate to "Master Schedule" 3. View Master Schedule 4. Select one or more Periods from Pick List 5. View Course Sections available
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Search Master Schedule by Day
Actors:	Researcher
Initial Condition:	1. User logs in to website using Researcher Credentials
Basic Flow:	1. From the Researcher Home Page, Navigate to "Researcher Console" 2. Navigate to "Master Schedule" 3. View Master Schedule 4. Select one or more Days from Pick List 5. View Course Sections available
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Search Master Schedule by Professor
Actors:	Researcher
Initial Condition:	1. User logs in to website using Researcher Credentials
Basic Flow:	1. From the Researcher Home Page, Navigate to "Researcher Console" 2. Navigate to "Master Schedule" 3. View Master Schedule 4. Type Faculty in search 5. View Course Sections available taught by that Faculty
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	Search Master Schedule by Course Name
Actors:	Researcher
Initial Condition:	1. User logs in to website using Researcher Credentials
Basic Flow:	1. From the Researcher Home Page, Navigate to "Researcher Console" 2. Navigate to "Master Schedule" 3. View Master Schedule 4. Type Course Name in search 5. View Course Sections available of that Course Name
Exit Condition:	1. User logs out of account 2. Web browser is closed

Use Case:	View Statistics
Actors:	Researcher
Initial Condition:	2. User logs in to website using Researcher Credentials
Basic Flow:	1. From the Researcher Home Page, Navigate to "Statistics" 2. View ANONYMOUS/UNIDENTIFYING statistics
Exit Condition:	1. User logs out of account 2. Web browser is closed

Aleutian Islands University