Use Case Models and Description

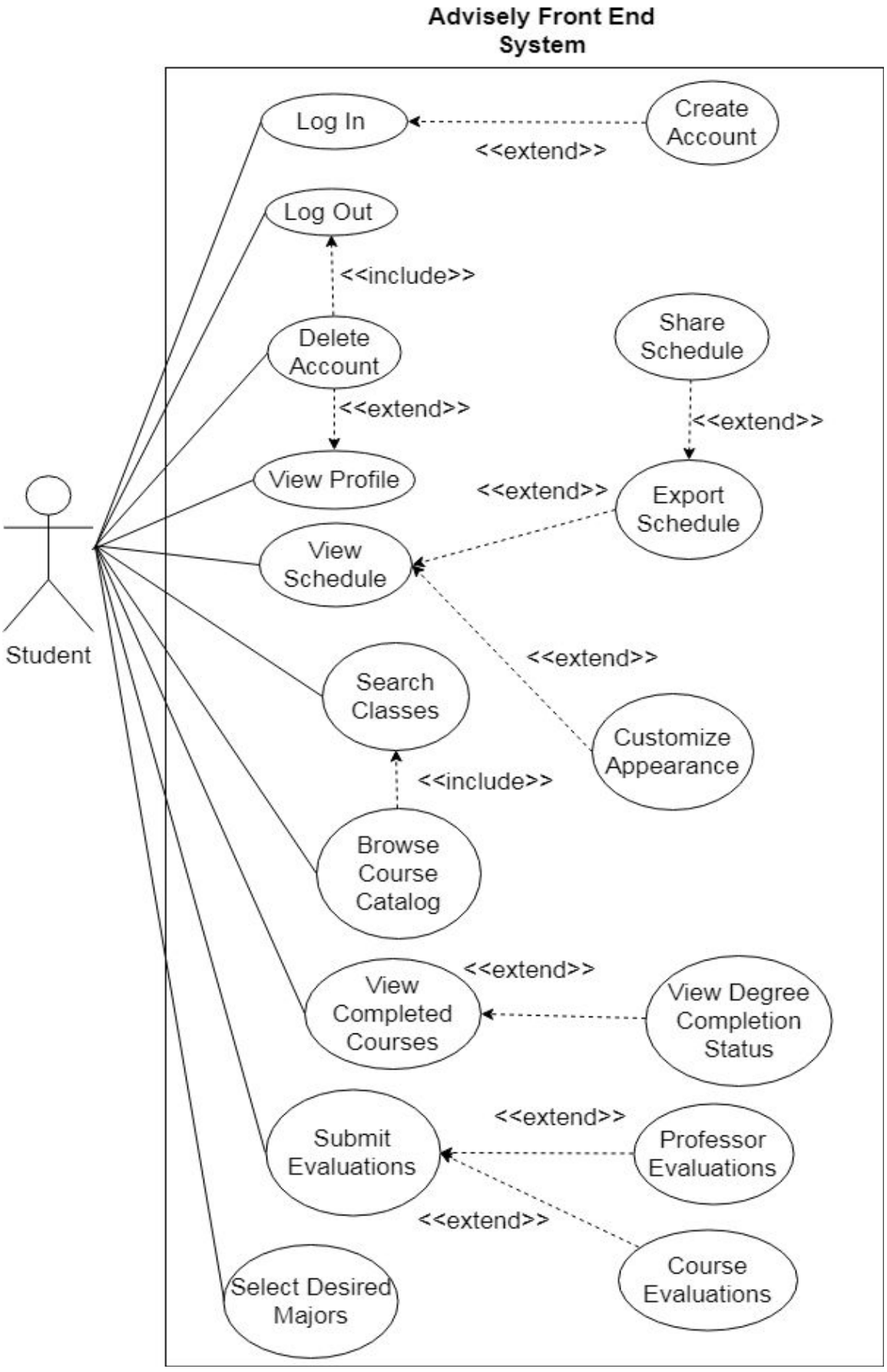
Advisely, An Education Support Application

Big Muscle Boys

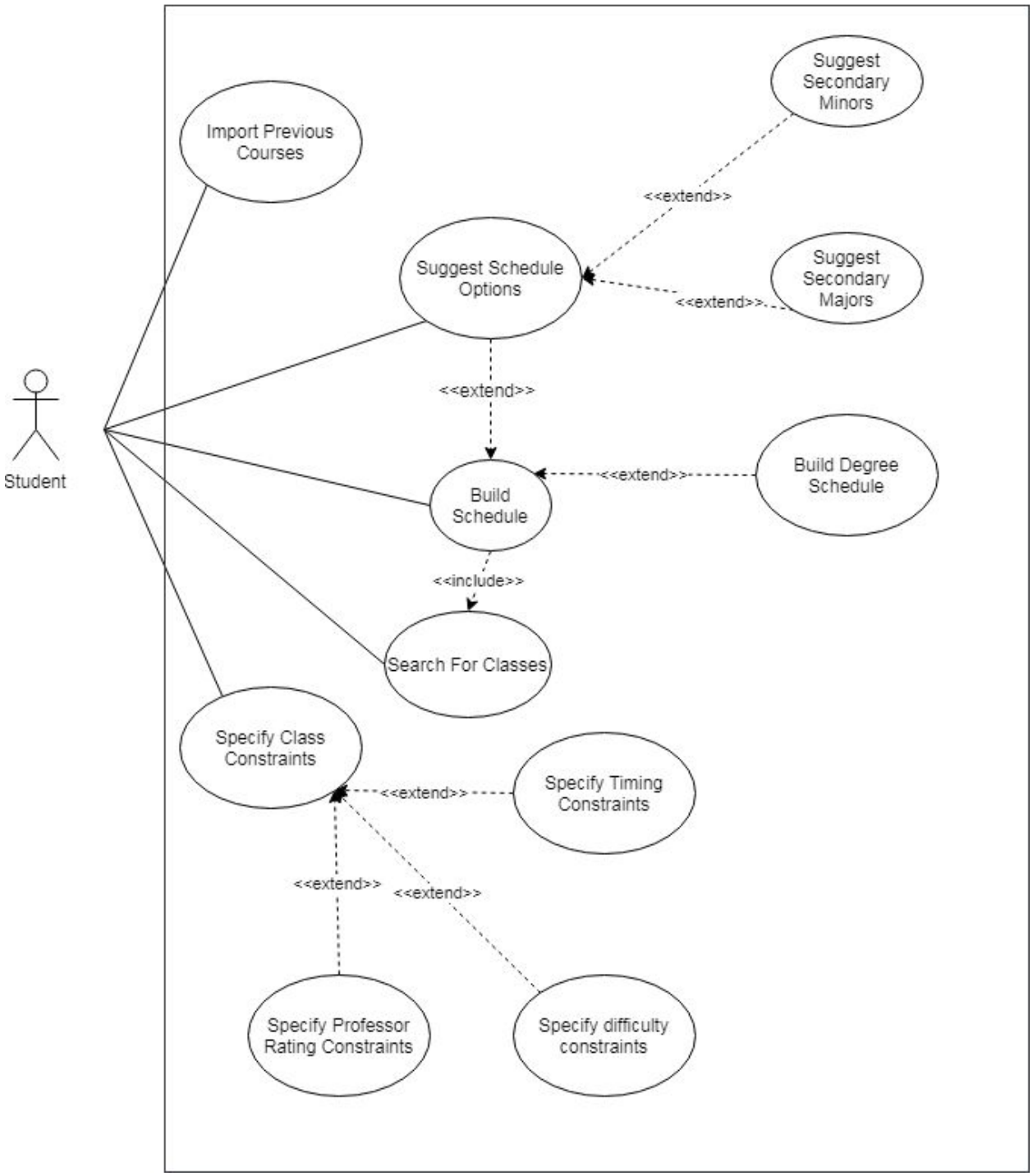
Sam Elliott, Jens Hansen, Connor Langlois, Eli Legere, Eric Schessler, Spencer Ward

COS 420 UMaine Spring 2019

Professor: Sepideh Ghanavati


Advisely Front End System
Log In
Create Account
<<extend>>
Log Out
<<include>>
Delete Account
<<extend>>
Share Schedule
<<extend>>
View Profile
<<extend>>
Export Schedule
View Schedule
Student
<<extend>>
Search Classes
Customize Appearance
<<include>>
Browse Course Catalog
<<extend>>
View Completed Courses
View Degree Completion Status
Submit Evaluations
<<extend>>
Professor Evaluations
<<extend>>
Course Evaluations
Select Desired Majors

Student Schedule Planning System



Use case: Login
Goals: Allow the user to access their account after creation
Summary: The user will enter their username and password to the system, which the system will authenticate. It will then either ask the user to retry if invalid, or move them on to the homepage if valid.
Steps:

Precondition: The student has created an account.	
Agent: Student	System: Advisely
1. The user enters the username & password they made during account creation 2. The user presses the “Login” button 5. The user presses the “OK” button on the popup	0. The system displays the login/create account page 2. The system authenticates the username and password entries If authenticated: 3. The system displays the home page If authentication failed: 4. The system displays a login failure pop-up 6. The system displays the login/create account page
Postcondition: The user is logged in.	

Use case: Create Account
Goals: Allow a new user to create an account
Summary: The user will enter a username, email address, and password in the system, which the system will check against existing accounts. If it finds a duplicate account, it will give them an error and ask them to retry. Otherwise, it will create their account and log them in.
Related use cases: Extension of Login
Steps:

Precondition: The student is a new user	
Agent: Student	System: Advisely
1. The user enters a username, email address, and password as their account credentials 2. The user clicks the “Create” button	0. The system displays the create account page 3. The system checks if the username or email address are attached to an existing account and a) if they are, return to step 1 with an error, or b) if they aren’t, create the account 4. The system logs the user in and displays the homepage
Postcondition: The account is created and the user logged in.	

Use case: Logout
Goals: Allow the user to log out of the system
Summary: The user will request to log out, the system will give them a confirmation pop up. If they confirm, the system will log them out.
Steps:

Precondition: The user is logged in and viewing the homepage.	
Agent: Student	System: Advisely
1. The user clicks the “Logout” button 3. The user can a) click “Yes”, or b) click “No	0: The system displays the homepage 2. The system displays a confirmation pop up asking the user to confirm that they want to log out 4. The system will a) close the popup, log the user out and display the login page b) close the popup
Postcondition: The user will be logged out if they select “Yes” in the popup.	

Use case: Delete Account
Goals: Allow the user to delete their account (and thereby delete associated information)
Summary: The user will select the “Delete Account” button, after which the system will ask them to confirm the action. Once confirmed the account and associated information will be removed from the database and the user will be logged out
Related use cases: Includes: Logout
Steps:

Precondition: The user is logged in and viewing their profile.	
Agent: Student	System: Advisely
1. The user presses the “DELETE ACCOUNT” button 3. The user presses the “I WISH TO PROCEED” button 5. The user presses the “OK” button	0. The system displays the view profile page 2. The system displays the confirm action page 4. The system displays a pop-up saying that the user account has been deleted 5. The system logs the user out 6. The system returns to the login/account creation page
Postcondition: The user is logged out and their account is deleted.	

Use case: View Schedule
Goals: Display the user's schedule
Summary: The user will click the "View Schedule" button on the homepage and the system will display the schedule page
Steps:

Precondition: The user is logged in and viewing the homepage.	
Agent: Student	System: Advisely
1. The user clicks the "View Schedule" button	0. The system displays the homepage 2. The system displays the schedule page
Postcondition: The user is viewing their schedule.	

Use case: Export Schedule
Goals: Allow the user to export their schedule to view outside of the site
Summary: The user will request to export their schedule, select a resolution, and export their schedule as an image.
Related use cases: Extension of: View Schedule
Steps:

Precondition: The user is logged in and viewing their schedule.	
Agent: Student	System: Advisely
1. The user clicks the "Export Schedule" button 3. The user selects a resolution from the dropdown 4. The user clicks the "Export"	0: The system displays the view schedule page 2. The system displays a popup asking them what resolution they would like 5. The system will export their schedule as an image with the requested resolution
Postcondition: The user's schedule is exported.	

Use case: Share Schedule
Goals: Allow the user to share their schedule on social media
Summary: The user will request to share their schedule, select their social media platform of choice, and select the people they want to share it with. The system will export the schedule as an image and share it.
Related use cases:
Extension of: Export Schedule
Steps:

Precondition: The user is logged in and viewing their schedule.	
Agent: Student	System: Advisely
1. The user presses the ”Share Schedule” button 3. The user selects the social media platform they wish to share through 5. The user selects the people they wish to share the schedule with	0. The system displays the the view schedule page 2. The system displays a social media selection pop-up 4. The system displays the selected platform-specific sharing interface 6. The system exports the file as a .png
Postcondition: The user’s schedule is shared.	

Use case: Customize Appearance
Goals: Allow the user to choose how their schedule is displayed
Summary: The user can choose how their schedule is displayed to them, layout, color etc. are all customizable and will be saved to the account after selection.
Related use cases: Extension of: View Schedule
Steps:

Precondition: The user is logged in and viewing their schedule.	
Agent: Student	System: Advisely
1. The user presses the “Customize Appearance” button 3. The user selects which changes they wish to apply to their schedule and press the “Okay” button	0. The system displays the users current schedule 2. The system prompts the user to choose how they want their schedule to appear 3. The system updates the schedules appearance and saves the users preference
Postcondition: The appearance of the user’s schedule is customized.	

Use case: Search Classes
Goals: Allow the user to search for classes based on criteria
Summary: The user will open the search page and enter their desired criteria, then the system will show them classes which match their criteria.
Steps:

Precondition: The user is logged in and viewing the homepage.	
Agent: Student	System: Advisely
1. The user presses the “Search Classes” button 3. The user fills out the search criteria and presses the “Search” button	0. The system displays the homepage 2. The system displays the search page and prompts the user to enter information on which classes they want to see 4. The system displays all classes which match the search criteria
Postcondition: The user is viewing class search results.	

Use case: Browse Course Catalog
Goals: Allow the user to browse the courses they can take
Summary: The user will open the catalog screen and scroll through the classes.
Related use cases: Includes: Search Classes
Steps:

Precondition: The user is logged in and viewing the homepage.	
Agent: Student	System: Advisely
1. The user clicks the “Course Catalog” button 3. The user scrolls through the catalog	0. The user is on the homepage 2. The system displays the course catalog screen
Postcondition: The user is browsing a list of classes.	

Use case: View Completed Courses
Goals: Allow the user to view the courses they have already completed
Summary: The user opens the progress screen, and the system shows them the courses they have completed so far.
Steps:

Precondition: The user is logged in and viewing the homepage.	
Agent: Student	System: Advisely
1. The user clicks the “Progress” button 3. The user can see the classes they have completed	0. The user is on the homepage 2. The system displays the progress screen
Postcondition: The user is viewing their completed courses.	

Use case: View Degree Completion Status
Goals: The goal of this is to show users how close to completion they are for any given degree they are working on, to instill a sense of accomplishment.
Summary: The user will navigate to their profile page, and select “degree progress”. They will then select which degree they’d whose progress they’d like to see, if there are several options. The system will then display the desired information.

Steps:

Precondition: The user is on their profile (and hence logged in)	
Agent: Student	System: Advisely
1. The user clicks the “Progress” button 3. The user can see the classes they have completed	0. The user is viewing their profile 2. The system displays the progress screen
Postcondition: The user is viewing their completed courses.	

Use case: Submit Evaluations
Goals: For users to be able to rate professors and courses based on their experiences
Summary: Users will navigate to an evaluation form. They will then designate which course or professor they wish to evaluate. Then, they will enter information about their perceived difficulty, enjoyment, and time spent for others to gauge from.
Steps:

Precondition: The user is on their schedule page (and hence logged in)	
Agent: Student	System: Advisely
1. The user clicks on one of the classes in their schedule 3. The user clicks the “review” button 5. The user fills out the form 6. The user clicks “Submit”	0. The user is on the schedule page 2. The system displays the page for that class 4. The system displays the evaluation form 7. The system submits the evaluation
Postcondition: The user’s evaluations are submitted.	

Use case: Professor Evaluations
Goals: To allow students to enter the experiences they had with any given professor.
Summary: The user will navigate to their profile. They will then select the evaluation feature along the sidebar. A form will be displayed to select the professor they wish to evaluate. Once their submission is complete, they will submit the form.
Related use cases: Extension of: Submit Evaluations
Steps:

Precondition: The user is logged in and viewing the homepage.	
Agent: Student	System: Advisely
1. The user navigates to their profile 2. The user selects “evaluate” 4. The user selects the person/class they wish to evaluate 5. The user submits the form	0: The system displays the homepage 3. The system loads the evaluation form 6. The system pushes this information to the database.
Postcondition: The user’s professor evaluations are submitted.	

Use case: Class Evaluations
Goals: To allow students to submit feedback about quality and difficulty of a course they've taken
Summary: The user will navigate to their profile. They will then select the valuation feature along the sidebar. A form will be displayed to select the class they wish to evaluate. Once their submission is complete, they will submit the form
Related use cases: Extension of: Submit Evaluations
Steps:

Precondition: The user is logged in and viewing the homepage.	
Agent: Student	System: Advisely
1. The user selects "EVALUATE" from the sidebar 3. The user presses the "CLASS" button 4. The user enters the evaluation information 5. The user press the "SUBMIT" button	0. The system displays the homepage 2. The system displays the evaluations page 3. The system displays the course evaluation form 6. The system pushes the information to the database 7. The system displays the evaluations page
Postcondition: The user's class evaluations are submitted.	

Use case: Select Desired Majors
Goals: For the student to enter the majors they wish to complete
Summary: The user will enter their account settings to enter the majors they wish to complete.
Steps:

Precondition: The user is logged in and viewing the homepage.	
Agent: Student	System: Advisely
1.The user selects his settings page 3. The user selects which alters which majors they wish to complete 4. The user submits this form	0. The system will display the homepage 2. The system loads the user settings page 5. The system stores this information
Postcondition: The user's desired majors are saved.	

Use case: View Profile
Goals: To show the user various metrics and pieces of information about their schedule and degree.
Summary: The user will elect to enter their profile. The system will then display the corresponding page.
Steps:

Precondition: The user is logged in and viewing the homepage.	
Agent: Student	System: Advisely
1. The user presses the “VIEW PROFILE” button	0. The system displays the homepage 2. The system them loads the profile
Postcondition: The user is viewing their profile.	

----------DIAGRAM 2----------

Use case: Import Previous Courses
Goals: This is to allow students to enter the courses they have already completed, to omit pointless suggestions from Advisely and create a more complete idea of their degree progress.
Summary: The user navigates to their profile. Once the user has arrived, they select the update courses button. When the form appears, the user enters which courses they have already completed.
Steps:

Precondition: The user is logged in and viewing the homepage.	
Agent: Student	System: Advisely
1. The user elects to enter their profile 3. The user selects the “update courses” button 5. The user enters their previously completed courses and submits the form	0. The user is on the homepage 2. The users profile is loaded 4. The system displays the “update courses” form 6. This information is stored for later reference
Postcondition: The user’s previous courses are imported.	

Use case: Suggest Schedule Options
Goals: The user is given a series of options for schedules that they can choose from.
Summary: The system offers some suggestions for schedules that will help the user achieve their goals. The user can select from these and create a schedule based off of them.
Related use cases: Extension of: Build Schedule
Steps:

Precondition: The user is logged in and viewing the Build Schedule page	
Agent: Student	System: Advisely
1. The user clicks the “Schedule Suggestions” button 3. The user selects one of the displayed schedules	0. The system displays the build schedule page 2. The system generates 3 sample schedules which fulfill the requirements for the users major and displays them to the user 4. The system updates the users schedule to reflect the selection
Postcondition: The user has their schedule populated with the chosen option	

Use case: Suggest Majors
Goals: Give the user suggestions for majors that would fit well into their schedules
Summary: The user opens the progress screen, and the system shows them suggestions for majors that would work well for them.
Related use cases:
Extension of: View Completed Courses
Steps:

Precondition: The user is logged in and viewing completed courses.	
Agent: Student	System: Advisely
1. The user clicks the “Progress” button 3. The user can see the system’s suggestions	0: The system displays the completed courses page 2. The system displays the progress screen
Postcondition: The user is viewing suggested majors.	

Use case: Suggest Minors
Goals: Give the user suggestions for minors that would fit well into their schedules
Summary: The user opens the progress screen, and the system shows them suggestions for major that would work well for them.
Related use cases: Extension of: View Completed Courses
Steps:

Precondition: The user is logged in and viewing completed courses.	
Agent: Student	System: Advisely
1. The user clicks the “Progress” button 3. The user can see the system’s suggestions	0: The system displays the suggested minors page 2. The system displays the progress screen
Postcondition: The user is viewing suggested minors.	

Use case: Build Schedule
Goals: To allow the user to construct a schedule based on the courses they still need to complete.
Summary: The user will navigate to their profile. From here they will select to build a schedule. Advisely will then suggest the preferred schedule path for their given degree. Students will then be able to import courses they desire to take, or ask for suggestions on what courses to take.
Related use cases: Includes: Search For Classes
Steps:

Precondition: The user is logged in and viewing the homepage.	
Agent: Student	System: Advisely
1. The student should navigate to their profile 3. The user selects the “build schedule option” 5. Users select the courses they wish to take and submit this information.	0. The system loads the homepage 2. The system loads the users profile. 4. The system loads the schedule builder application. 6. The system then stores the new schedule.
Postcondition: The user’s schedule is built.	

Use case: Build Degree Schedule
Goals: Allow the user to build their schedule based on their degree
Summary: While the user is building their schedule, they can choose to build their schedule around their degree classes.
Related use cases:

Extension of: Build Schedule

Steps:

Precondition: The user is logged in and is building their schedule.	
Agent: Student	System: Advisely
1. The user clicks the “Build by Degree” button	0. The user is on the schedule building page 2. The system displays degree requirements on the schedule building page 3. The system displays suggested classes for each of the requirements
4. The user clicks a) on a suggested class to view more details, or b) on a degree requirement to view the full list of classes	5. The system displays a) the page for the selected class and moves to step 6 b) the list of the classes that meet the selected requirement and moves to step 8
6. The user clicks “Add to Schedule”	7. The selected class is added to the user’s schedule and ends the use case
8. The user scrolls through the list of classes	
9. The user clicks on a class to learn more	10. The system displays the page for the selected class
11. The user clicks “Add to Schedule”	12. The system adds the selected class to the user’s schedule
Postcondition: The user’s degree schedule is built.	

Use case: Search For Classes
Goals: Allow the user to search for classes to add to their schedule
Summary: While the user is building their schedule, they have the option to search for a class. They enter their desired criteria and the system displays the matching classes. The user can click on a class to add it to their schedule.
Steps:

Precondition: The user is logged in and is building their schedule.	
Agent: Student	System: Advisely
1. The user clicks the “Search” button 3. The user selects their desired criteria 4. The user clicks “Search” 6. The user scrolls through the list of classes 7. The user clicks on a class they want to add 9. The user reads the class description 10. The user clicks the “Add to Schedule” button	0. The system is on the schedule building screen 2. The system displays the search screen 5. The system displays a list of classes that match the given criteria 8. The system displays the class page 11. The system adds the class to their schedule
Postcondition: The new class is added to their schedule	

Use case: Specify Class Constraints
Goals: Allow the user to specify constraints for classes so they are only suggested classes that meet these constraints
Summary: Users can specify aspects of classes by which they want to constrain the suggestions by. After specifying constraints the application will only suggest classes which fall within them.
Steps:

Precondition: The user is logged in and viewing the homepage.	
Agent: Student	System: Advisely
1. The user clicks the “Class Constraints” button 3. The user enters constraints 4. The user clicks the “Save” button	0. The system displays the homepage 2. The system displays the class constraints page and prompts the user to enter constraints 5. The system updates class constraints
Postcondition: The user’s class constraints are saved.	

Use case: Specify Timing Constraints
Goals: Allow users to filter their classes by what times they are available
Summary: Users can specify what time periods they want their classes to meet and will not be suggested classes which meet during times outside of those periods.
Related use cases: Extension of: Specify Class Constraints
Steps:

Precondition: The user is logged in and specifying their class constraints.	
Agent: Student	System: Advisely
1. The user enters what times they are willing to have classes in the time constraint field 2. The user clicks the “Save” button	0. The system displays the class constraints page 3. The system updates the users class constraints
Postcondition: The user’s timing constraints are saved.	

Use case: Specify Difficulty Constraints
Goals: Allow users to filter their classes by a difficulty range
Summary: Users can specify how difficult they would like their classes to be and will not be suggested classes which have difficulty ratings outside of that range.
Related use cases: Extension of: Specify Class Constraints
Steps:

Precondition: The user is logged in and specifying their class constraints.	
Agent: Student	System: Advisely
1. The user enters a range of difficulty ratings in the difficulty constraint field 2. The user clicks the “Save” button	0. The system displays the class constraints page 3. The system updates the users class constraints
Postcondition: The user’s difficulty constraints are saved.	

Use case: Specify Professor Rating Constraints
Goals: Allow users to filter their classes by a minimum professor rating
Summary: Users can specify a minimum rating for professors and will not be suggested classes which have professors with ratings bellow that constraint.
Related use cases:

Extension of: Specify Class Constraints

Steps:

Precondition: The user is logged in and specifying their class constraints.	
Agent: Student	System: Advisely
1. The user enters a minimum rating in the professor rating constraint field 2. The user clicks the “Save” button	0. The system displays the class constraints page 3. The system updates the users class constraints
Postcondition: The user’s professor rating constraints are saved.	