Use Case Models and Description

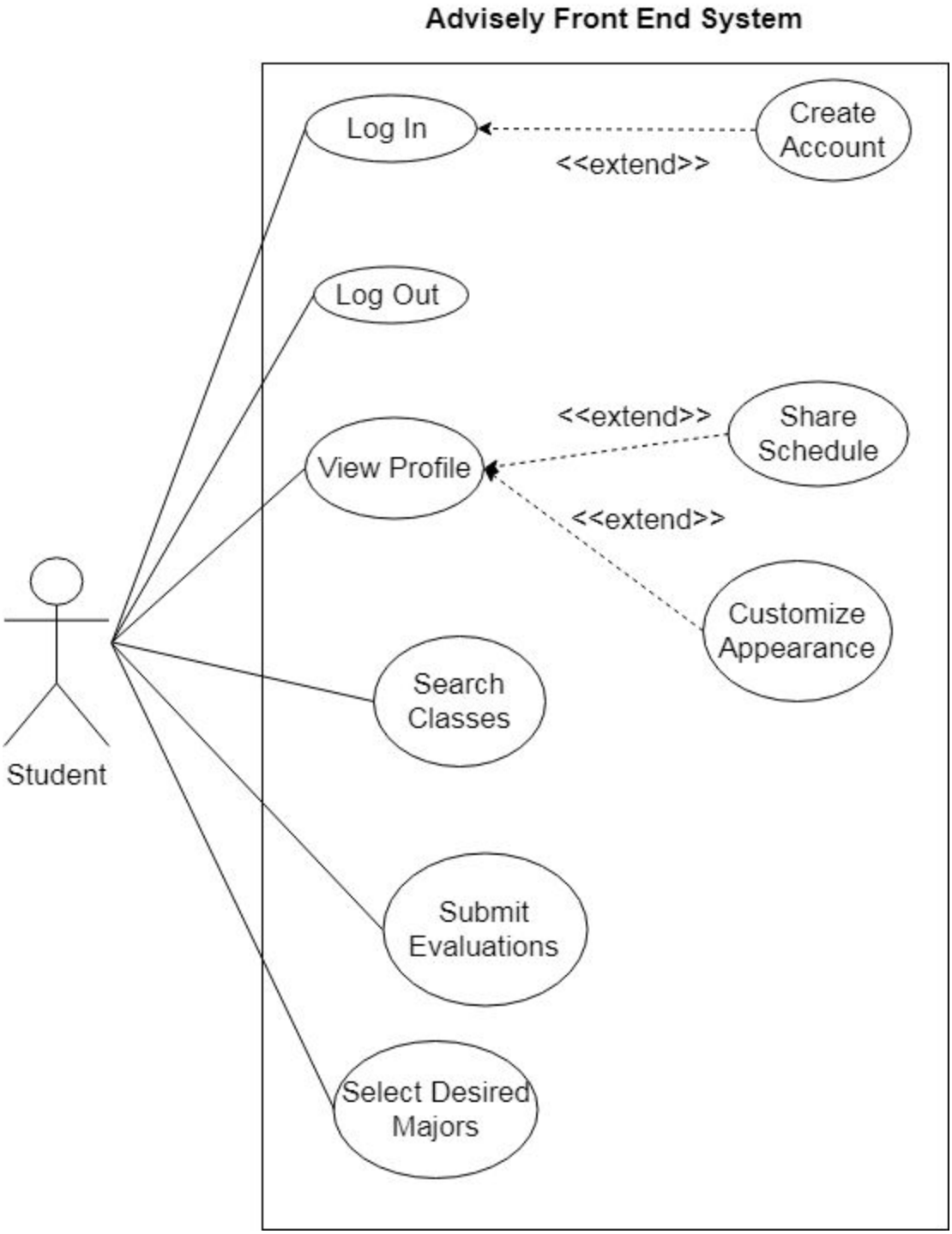
Advisely, An Education Support Application

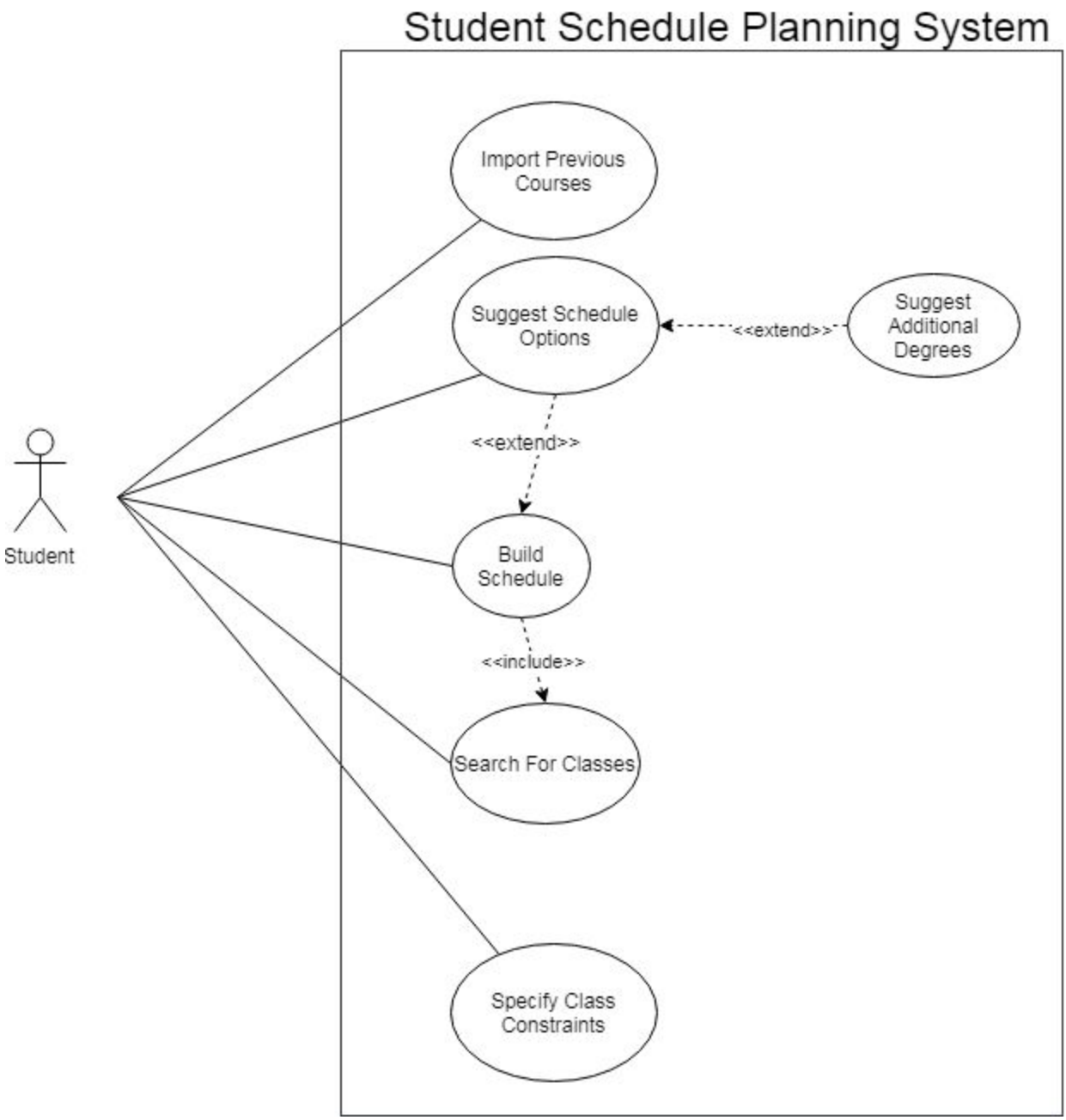
Big Muscle Boys

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
<<extend>>
Share Schedule
View Profile
<<extend>>
Customize Appearance
Search Classes
Submit Evaluations
Select Desired Majors
Student


Student Schedule Planning System
Import Previous Courses
Suggest Schedule Options
<<extend>>
Suggest Additional Degrees
<<extend>>
Build Schedule
<<include>>
Search For Classes
Specify Class Constraints
Student

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.
Related Use Cases: Extends Create Account
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	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 (directs to the create account use case)
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	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	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: View Profile
Goals: Allow the user to view information on a single display, including the schedule itself
Summary: The View profile feature will be the mainstay of what the user looks at post-schedule completion, which will be accessible from the homescreen. The screen will display everything important information the user has entered for their profile, and the completed schedule
Steps:

Precondition: The user is logged in and viewing the homepage	
Agent: Student	System: Advisely
1. The user clicks the “VIEW PROFILE” button	2. The system Displays the full profile view
Postcondition: The user will be viewing the full profile view	

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: View Profile
Steps:

Precondition: The user is logged in and viewing their schedule in their prrofile's page	
Agent: Student	System: Advisely
1. The user presses the "Share Schedule" button 3. The user selects the social media platform they wish to shdare through 5. The user selects the people they wish to share the schedule with	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 Profile
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	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: 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”	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: 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 presses the “MY DEGREES” button 3. The user selects which alters which majors they wish to complete 4. The user submits this form	2. The system displays the degree selection page 5. The system stores this information
Postcondition: The user’s desired majors are saved.	

----------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 presses the “UPDATE COURSES” button 5. The user enters their previously completed courses and submits the form	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; Extends Suggest Additional Degrees
Steps:

Precondition: The user is logged in and viewing the Build Schedule page	
Agent: Student	System: Advisely
1. The user presses the “Schedule Suggestions” button 3. The user selects one of the displayed schedules	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 Additional Degrees
Goals: Give the user suggestions for majors that would fit well into their schedules
Summary: The user opens the view profile page, and the system shows them suggestions for major and minors that would may be attainable based on classes they are in/have already taken
Related use cases:

Extension of: Suggest Schedule Options

Steps:

Precondition: The user is logged in and viewing completed courses.	
Agent: Student	System: Advisely
1. The user presses the “SHOW MY PROGRESS” button 3. The user presses “VIEW MY SUGGESTIONS”	2. The system displays the progress screen 4. The system displays suggested majors/minors with user-associated classes next to them in a pop-up
Postcondition: The user is viewing suggested majors.	

Use case: Build Schedule
Goals: Allow the user to build a schedule for their desired degree path
Summary: The app is able to build schedules based on degree requirements and, allows the user to add classes by searching and then manually adding them.
Related use cases:

Includes: Search for classes, Extends: Suggest Schedule Options

Steps:

Precondition: The user is logged in and is building their schedule.	
Agent: Student	System: Advisely
1. The user presses the “BUILD MY DEGREE” button 3. The user presses the “OK” button 4. The user selects the semester they wish to build for	2. The system displays degree requirements on the schedule building page 3. The system displays the semester selection pop-up 5. The system displays suggested classes for each of the requirements
6. The user selects a) a suggested class to view more details (move to step 8a) b) a degree requirement to view the full list of classes (move to step 8b)	8. The system displays a) the page for the selected class (move to step 9a) b) the list of the classes that meet the selected requirement (move to step 9b)
9. The user a) presses the “ADD” button (move to step 10) b) Selects a class from the list (move to 8a)	10. The selected class is added to the user’s schedule 11. The system displays the user’s current schedule
12. The user a) Presses the “ADD ANOTHER CLASS” button (move to 4) b) Presses the “FINISH” schedule button	13. The system displays the finished schedule
Postcondition: The user’s degree schedule is built.	

Use case: Search For Classes
Goals: Allow the user to search for classes they may wish 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.
Related Use Cases: Included by Build Schedule
Steps:

Precondition: The user is logged in and is building their schedule.	
Agent: Student	System: Advisely
1. The user presses the “SEARCH” button 3. The user selects their desired criteria 4. The user presses the “GO” button 6. The user scrolls through the list of classes	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
Postcondition: The new class is added to their schedule	

Use case: Specify Class Constraints
Goals: Allow the user to specify constraints for classes to refine their schedule’s creation
Summary: Users can specify aspects of classes by which they want to constrain the suggestions by. These constraints include input for when/who teaches the class, and how hard the class is. 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 for class time and/or desired professor and/or class difficulty 4. The user clicks the “SAVE” button	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.	