

“Chart My Course” Group Project

Iteration 1

Team Members:

- Harm Drenth (team leader)
- Warren Burrus
- Anshpreet Kaur
- Ricardo Boone
- Tacoma Velez
- Josh Wilson

Table of Contents:

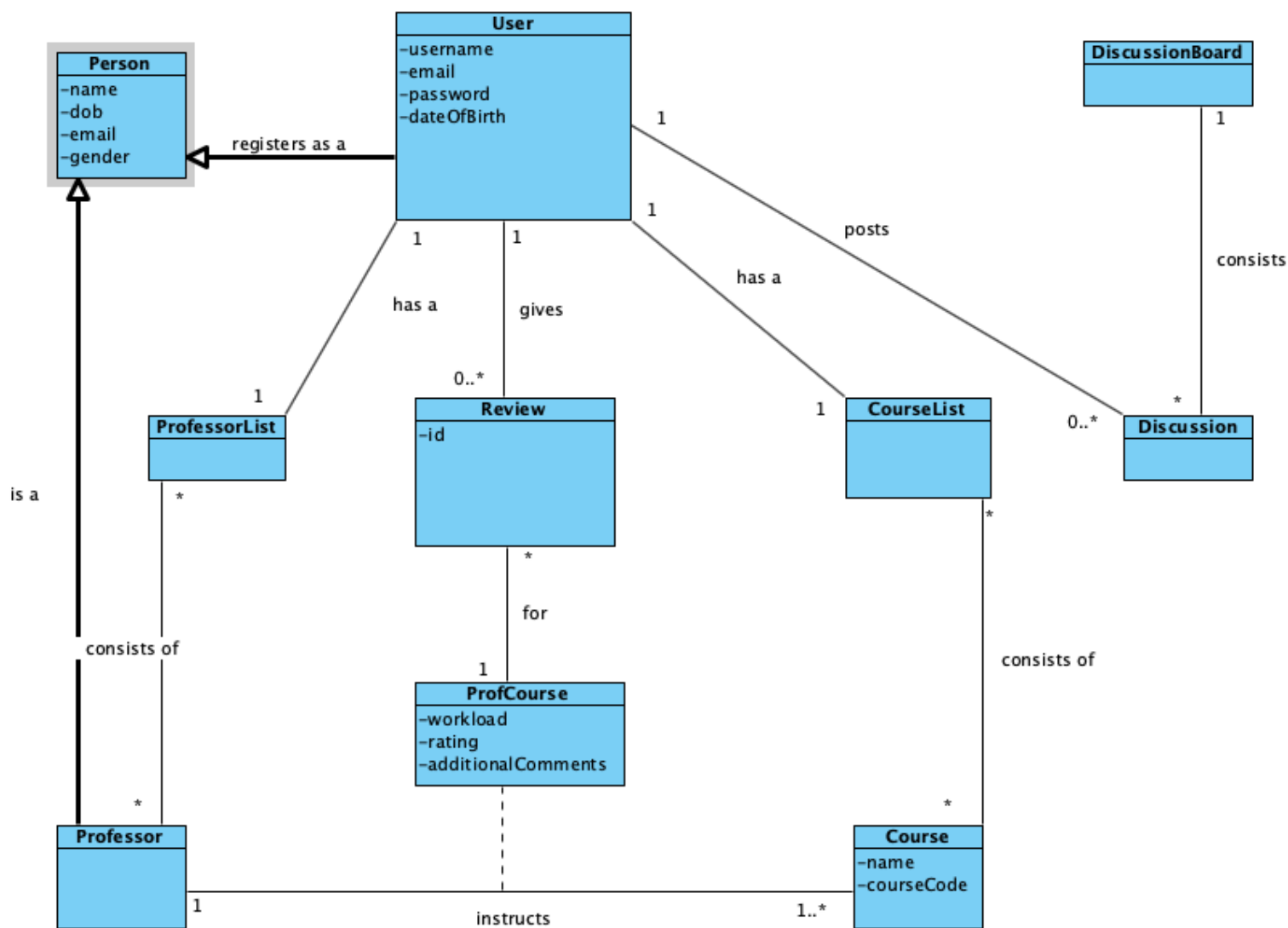
1. [Project Vision](#)
2. [Domain Model](#)
3. [Requirements](#)
 - a. [Functional](#)
 - b. [Non-functional](#)
4. [Use Cases](#)
5. [Traceability Matrix](#)
6. [Wireframes](#)
7. [Gantt Diagram](#)
8. [Jira](#)
9. [Website](#)
10. [Git Link](#)

1. Project Vision

Navigating college is hard. Navigating the choices of classes is even harder. Chart My Course seeks to simplify a Baylor Computer Science student's choices of courses to take, as well as professors that teach them.

We intend to create a utility similar to the now defunct BUBooks that aggregates reviews of not just the course offerings, but also professor/course pairings. This utility is intended to give potential students a realistic look at what sort of coursework or workload they can expect in a class.

2. Domain Model



3. Requirements

a. Functional

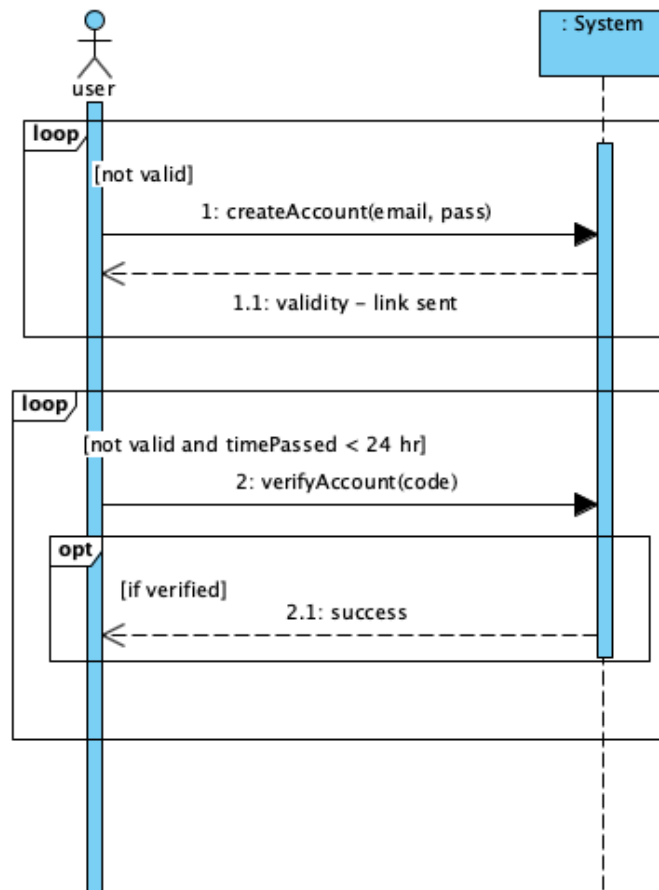
- The user can create a new account
- The user can log in and out of their account
- The user can find recommended courses and professors
- The user can add a review of a professor
- The user can search reviews of professors/courses
- The user can filter search results
- The user can request a review to be removed
- The user can have a list of saved professors
- The user can have a list of saved courses
- The system saves or deletes changes upon logout
- The user can post questions and answers in the discussion board
- The user can reply to discussions
- The user can upvote questions in the discussion board

b. Non-functional

- The system shall not share the personal information of the author of posts to other users, only information the operators of the system receive is the author's name
- All posts and user activity will be user friendly and comply to online community and safety guidelines
-
- The courses and professors in the system will be limited to Baylor's Computer Science department
- The system will adhere to all Baylor guidelines and policies
- The system will return search results within 30 seconds

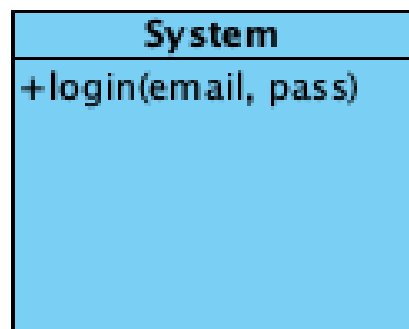
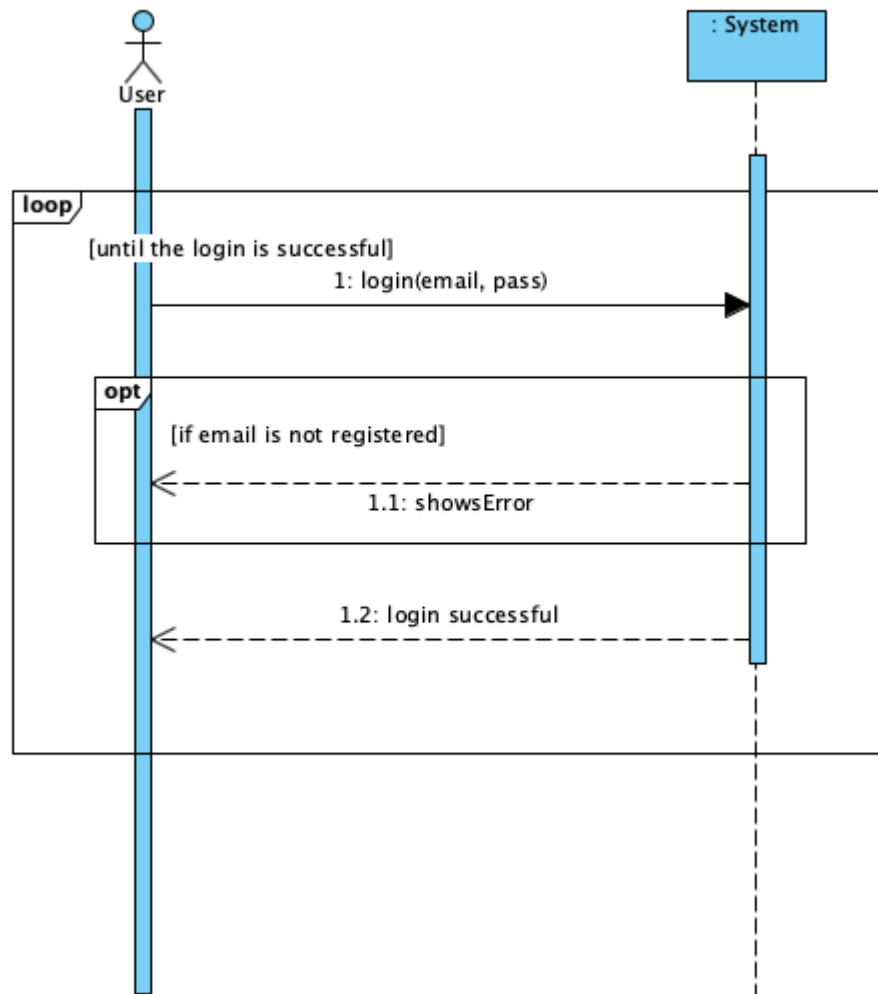
4. Use Cases

ID: UC Create Account
Scope: Course/Professor Planning
Level: User Goal
Actors: User - Student interacting with system Database Service - Site database with degree plan info and professor reviews info
Preconditions: The user is not logged in
Flow of events: <ol style="list-style-type: none"> 1. A new user wants to create an account 2. User clicks on Create Account 3. The system asks for Baylor email 4. The user enters their Baylor email 5. The system sends a verification email 6. The user clicks the link and is brought back to the screen 7. The system asks to enter the password and other information 8. The user enters the information and click submit
Extensions: <ol style="list-style-type: none"> 4a. The user does not enter Baylor Email <ol style="list-style-type: none"> 1. System will display "Please enter Baylor Email" 2. System will return to the start of step three 7a. The entered password does not meet the requirements <ol style="list-style-type: none"> 1. System will display "Please enter a new password"
Postconditions/Success: Account is created successfully



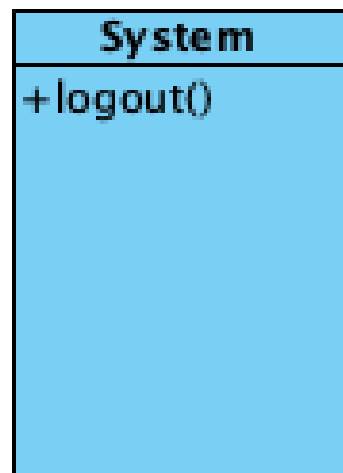
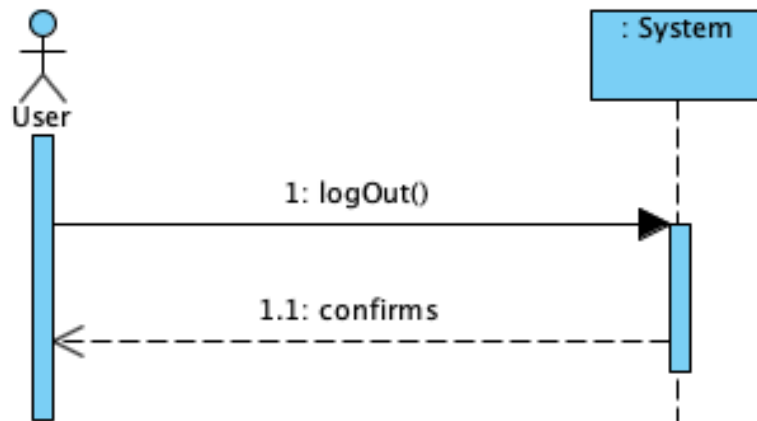
System
+createAccount(email, pass) +verifyAccount(code)

ID: UC Login
Scope: Course/Professor Planning
Level: User Goal
Actors: User - Student interacting with system Database Service - Site database with correct login information
Preconditions: The user has an account created
Flow of events: 1. The user clicks on login 2. The system asks for Baylor email and password 3. If the login credentials exist, the user is directed to the home page
Extensions: 2a. The user enters an email which does not exist 1. System will display "Please enter Baylor Email" 2. System will return to the start of step two 2b. The entered password does not match 1. System will display "Please re-enter" 2. If the user has had 3 tries, the system redirects to reset password
Postconditions/Success: User is logged in

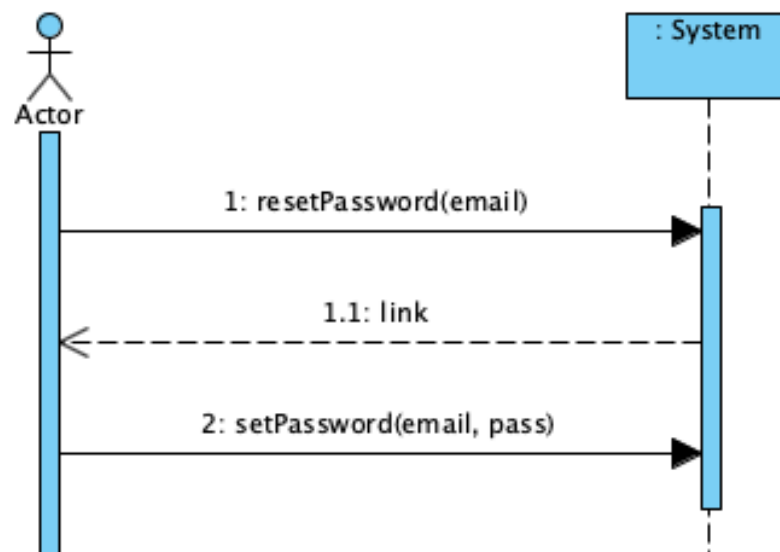


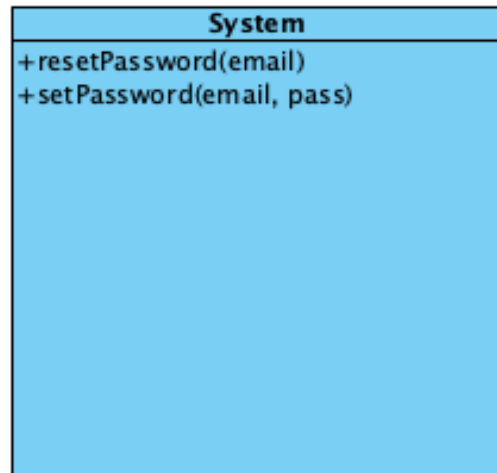
ID: UC Logout
Scope: Course/Professor Planning
Level: User Goal
Actors: User - Student interacting with system Database Service - Site database with correct login information
Preconditions: The user is logged in
Flow of events: 1. The user clicks on logout 2. The system confirms from the user about logging out and tells the user to save any changes 3. The user confirms 4. The system directs to the home page
Extensions: a* Anytime the system does not respond 1. User will restart the application
Postconditions/Success: User is logged out

ID: UC Login
Scope: Course/Professor Planning
Level: User Goal
Actors: User - Student interacting with system Database Service - Site database with correct login information
Preconditions: The user has an account created
Flow of events: 1. The user clicks on login 2. The system asks for Baylor email and password 3. If the login credentials exist, the user is directed to the home page
Extensions: 2a. The user enters an email which does not exist 1. System will display "Please enter Baylor Email" 2. System will return to the start of step two 2b. The entered password does not match 1. System will display "Please re-enter" 2. If the user has had 3 tries, the system redirects to reset password
Postconditions/Success: User is logged in

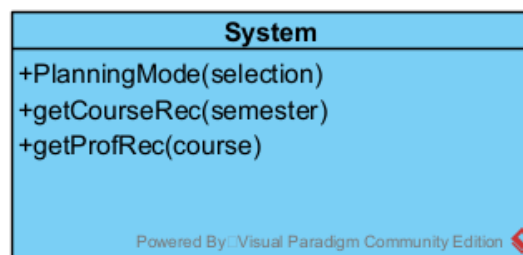
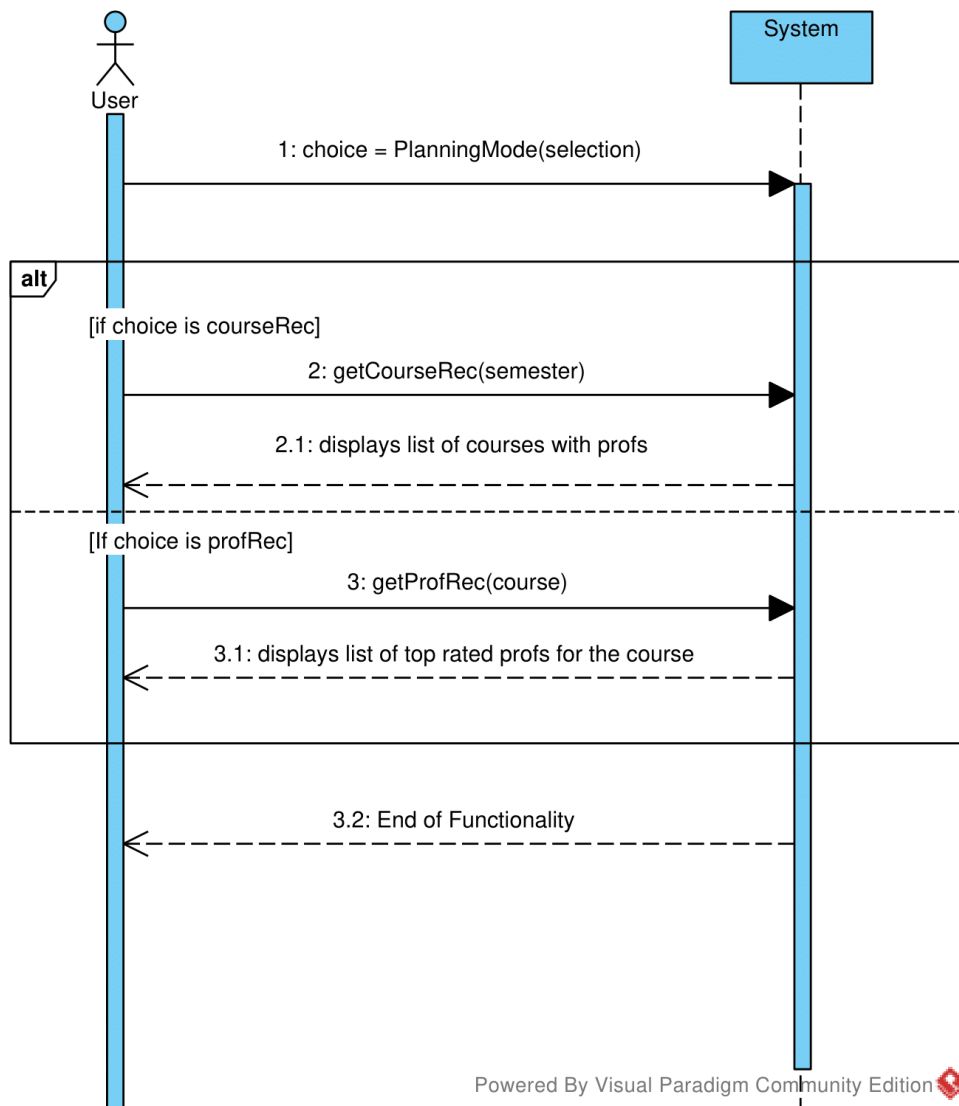


ID: UC Reset Password
Scope: Course/Professor Planning
Level: User Goal
Actors: User - Student interacting with system Database Service - Site database with correct login information
Preconditions: The user has an account created and is not logged in
Flow of events: <ol style="list-style-type: none"> 1. The user clicks on "Reset Password" 2. The system asks for Baylor email 3. The system link to their email 4. The user clicks the link and is brought back to reset password page 5. The system asks the user to enter the email and new password 6. The user enters and clicks on submit 7. The system updates the information in the database
Extensions: <ol style="list-style-type: none"> 2a. The user enters an email which does not exist <ol style="list-style-type: none"> 1. System will display "Please enter Baylor Email" 2. System will return to the start of step two 4a. The user does not respond to the link in 24 hours <ol style="list-style-type: none"> 1. The link expires. 5a. The password does not meet the requirements <ol style="list-style-type: none"> 1. The system asks the user to enter a new password
Postconditions/Success: The password is reset.

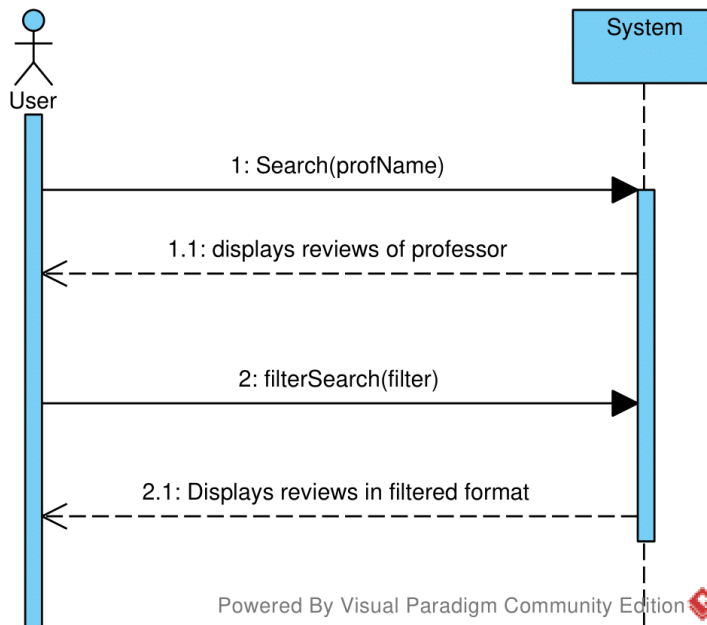


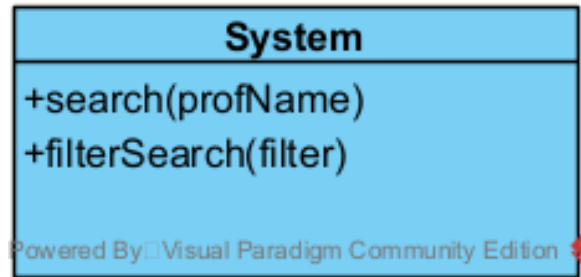


ID: UC Recommendations
Scope: Course/Professor Planning
Level: User Goal
Actors: User - Student interacting with system Database Service - Site database with degree plan info and professor reviews info
Preconditions: The user is on the planning page of the site
Flow of events: <ol style="list-style-type: none"> 1. User selects type of recommendation 2. System records the type of rec selected 3. If user choses course rec <ol style="list-style-type: none"> 1. User enters the semester they are in 2. Displays list of courses with professors 4. If user chooses prof rec <ol style="list-style-type: none"> 1. User enters the course they are in 2. Displays list of top rated profs for the selected course
Postconditions/Success: Site displays recommended courses along with the associated professors

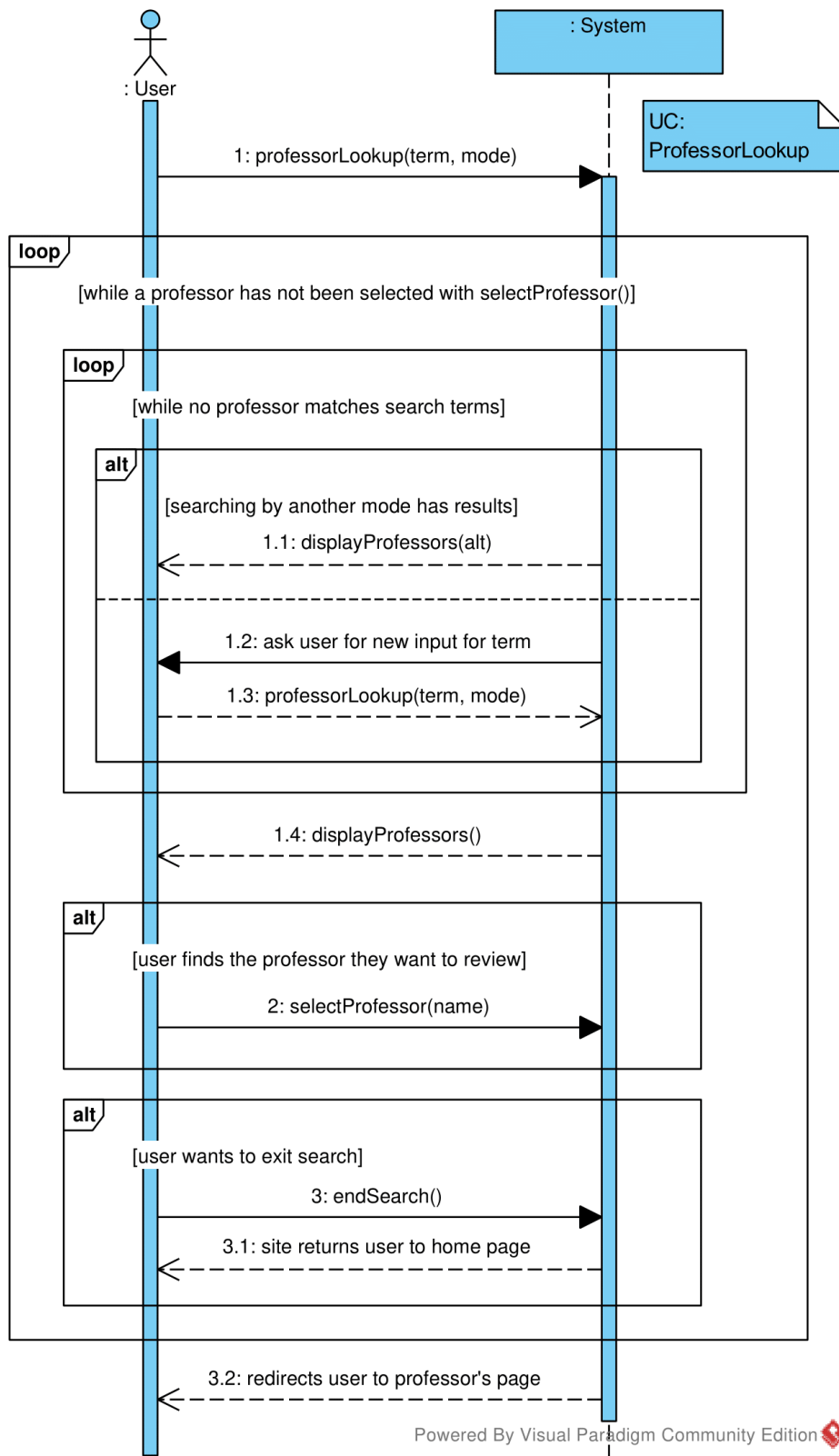


ID: UC Filter Results
Scope: Professor Reviews
Level: User Goal
Actors: User - Student interacting with system Database Service - Site database with reviews
Preconditions: The user is on the search professor page of the site
Flow of events: <ol style="list-style-type: none"> 1. User selects a professor's name from dropdown menu to search for reviews 2. Site displays the reviews of the given professor 3. If user wants to filter results the user can select a filter from the dropdown menu 4. The system filters the results, and the site displays filtered info
Postconditions/Success: Site displays the reviews with the given input from the user



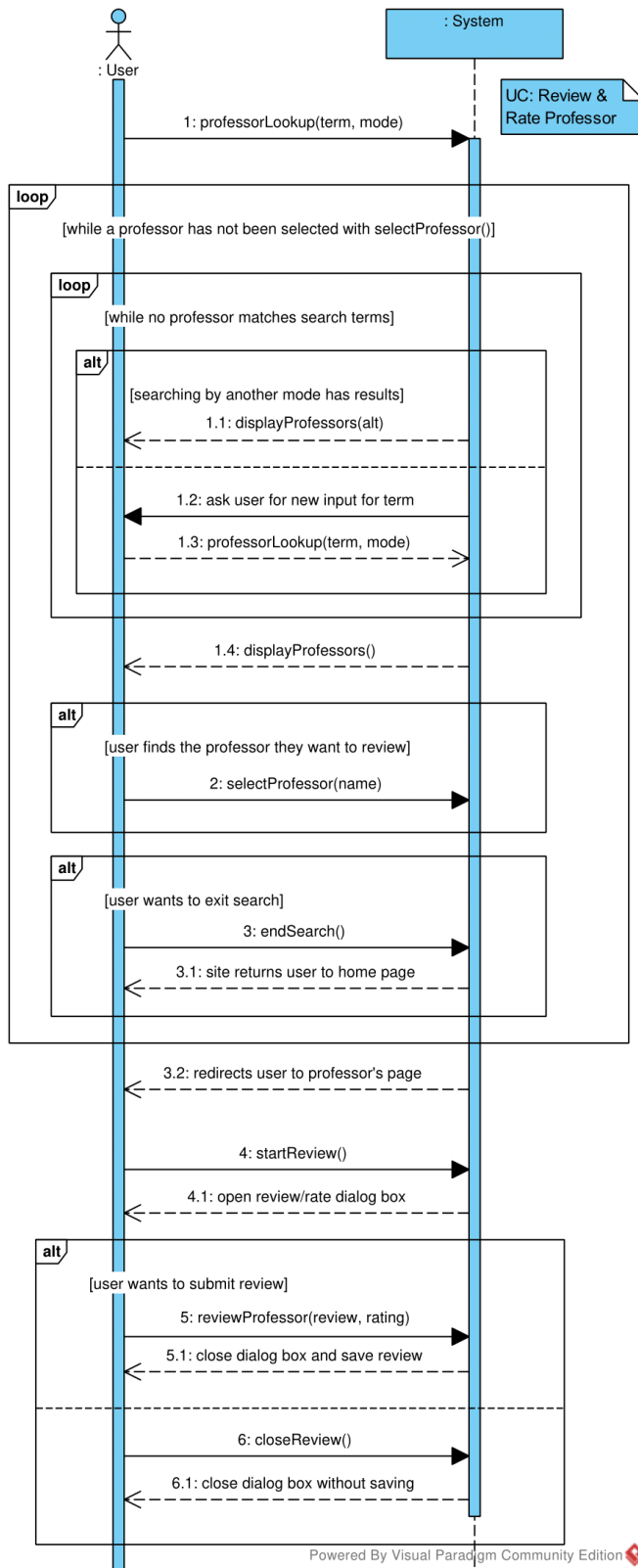


ID: UC Professor Lookup
Scope: Professor Lookup & Rating System
Level: User Goal
Actors: User - Student interacting with system Database Service - Site database with professor info
Preconditions: User must be logged into the website
Flow of events: <ol style="list-style-type: none"> 1. User opens the site main page 2. User enters search terms into the search box, and chooses a search mode (by professor name or by class) 3. Site redirects user to a list of all professors matching search terms 4. User selects the desired professor 5. Site redirects user to selected professor's page
Extensions: <ol style="list-style-type: none"> 3a. User exits search without choosing a professor <ol style="list-style-type: none"> 1. Site redirects user back to main page 3b. If no professors match search terms used <ol style="list-style-type: none"> 1. Site switches search mode and displays any matching results 2. If there are no results, site asks user for new search terms 3. Site continues this until a search provides valid results
Postconditions/Success: User is directed to professor's page by lookup system Site displays professor's reviews and ratings to user



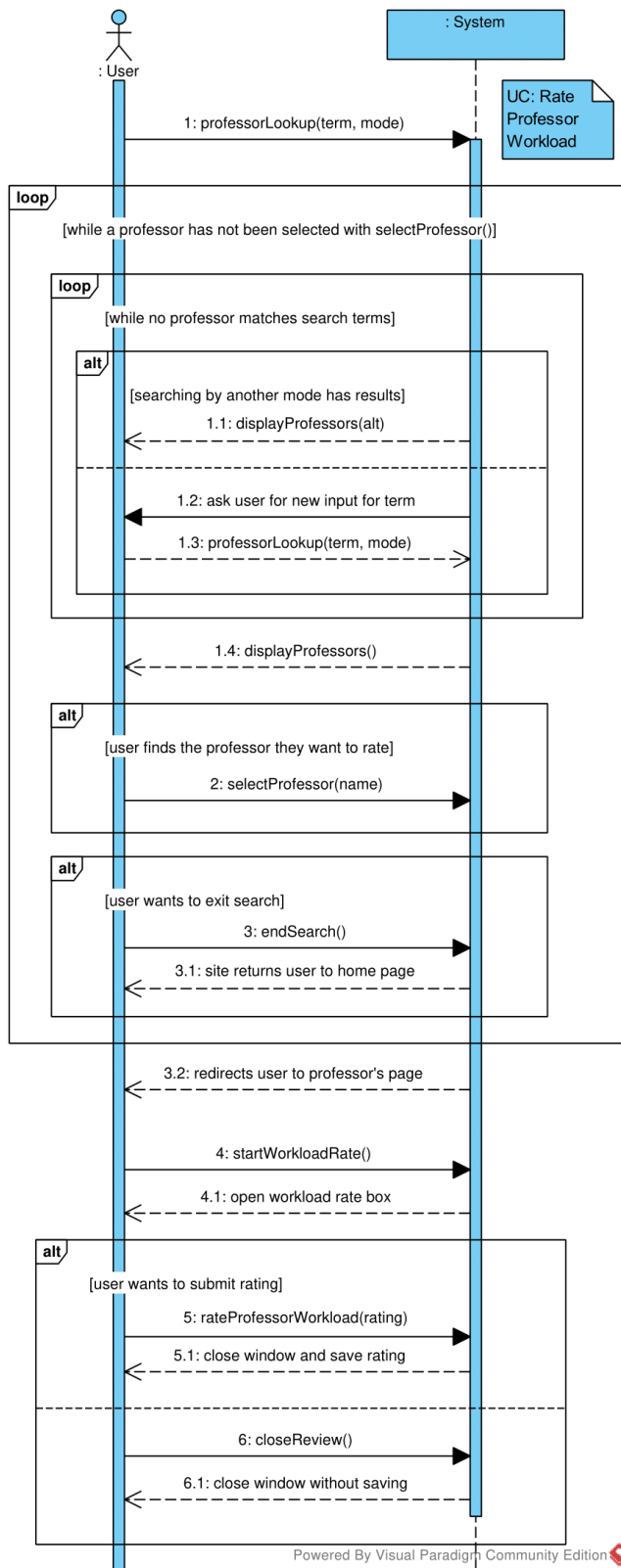
System
+professorLookup(term, mode) +displayProfessors() +selectProfessor(name) +endSearch() +startReview() +startWorkloadRate() +reviewProfessor(text, rating) +rateProfessorWorkload(rating) +closeReview()

ID: UC Review & Rate Professor
Scope: Professor Lookup & Rating System
Level: User Goal
Actors: User - Student interacting with system Database Service - Site database with professor info
Preconditions: User must be logged into the website
Flow of events: <ol style="list-style-type: none"> 1. User uses professor lookup system to navigate to desired professor's page 2. Site displays professor's page 3. User selects "Rate and Review Professor" 4. Site opens popup with text box and rating selection buttons 5. User types review into text box 6. User selects review rating with rating selection buttons 7. Site saves review and returns user to the professor's page
Extensions: <ol style="list-style-type: none"> 5a. User wants to exit review without posting <ol style="list-style-type: none"> 1. User clicks "X" button in review panel 2. Site returns user to professor's page and ends review/rating
Postconditions/Success: User's review and rating are stored in the site's database User's review and rating are now visible on the professor's page on the site



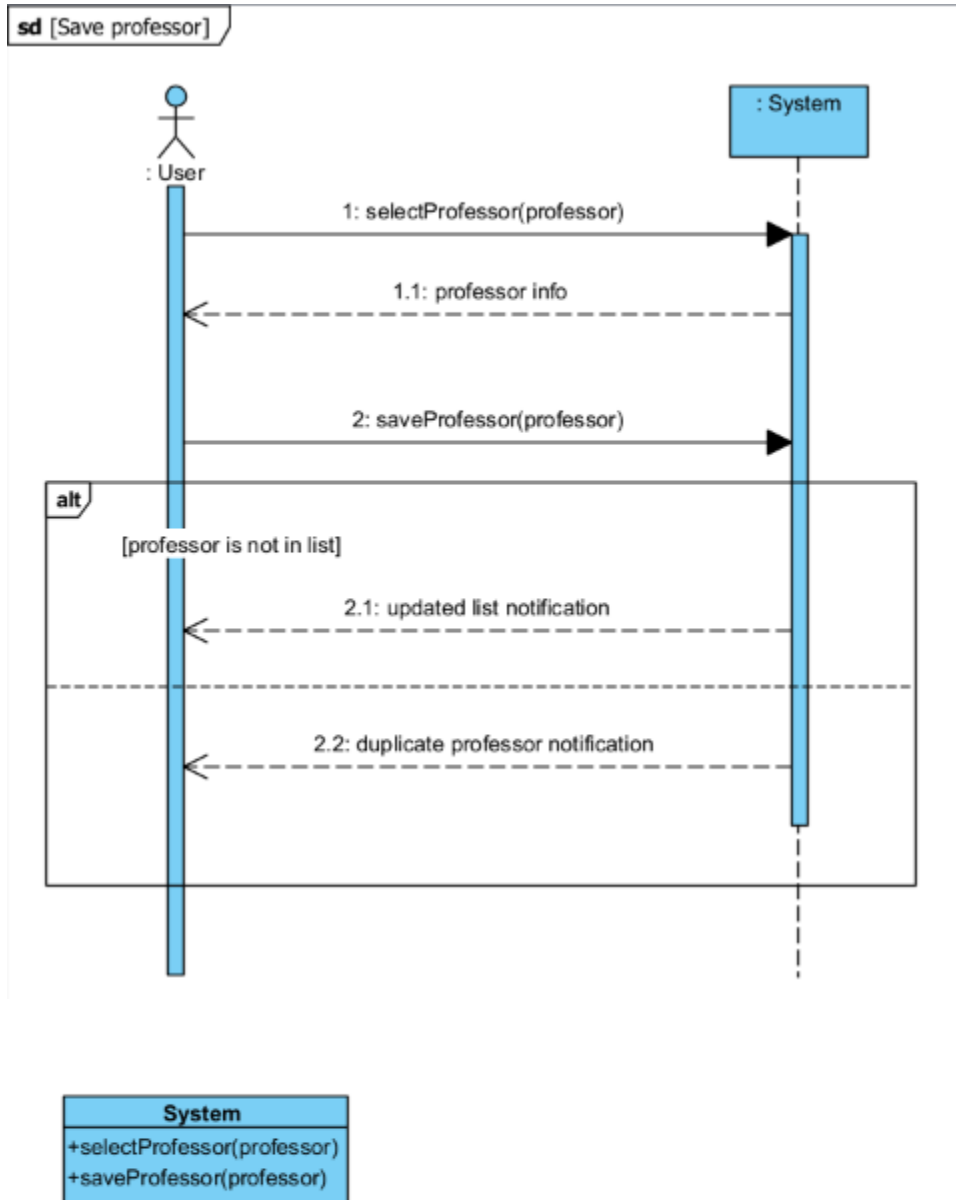
System
+professorLookup(term, mode) +displayProfessors() +selectProfessor(name) +endSearch() +startReview() +startWorkloadRate() +reviewProfessor(text, rating) +rateProfessorWorkload(rating) +closeReview()

ID: UC Rate Professor Workload
Scope: Professor Lookup & Rating System
Level: User Goal
Actors: User - Student interacting with system Database Service - Site database with professor info
Preconditions: User must be logged into the website
Flow of events: <ol style="list-style-type: none"> 1. User uses professor lookup system to navigate to desired professor's page 2. Site displays professor's page 3. User selects "Rate Professor Workload" 4. Site opens popup with rating selection buttons 5. User selects rating with rating selection buttons 6. Site saves rating and returns user to the professor's page
Extensions: <ol style="list-style-type: none"> 5a. User wants to exit rating without posting <ol style="list-style-type: none"> 1. User clicks "X" button in review panel 2. Site returns user to professor's page and ends rating
Postconditions/Success: User's workload rating is stored in the site's database User's workload rating is now visible on the professor's page on the site



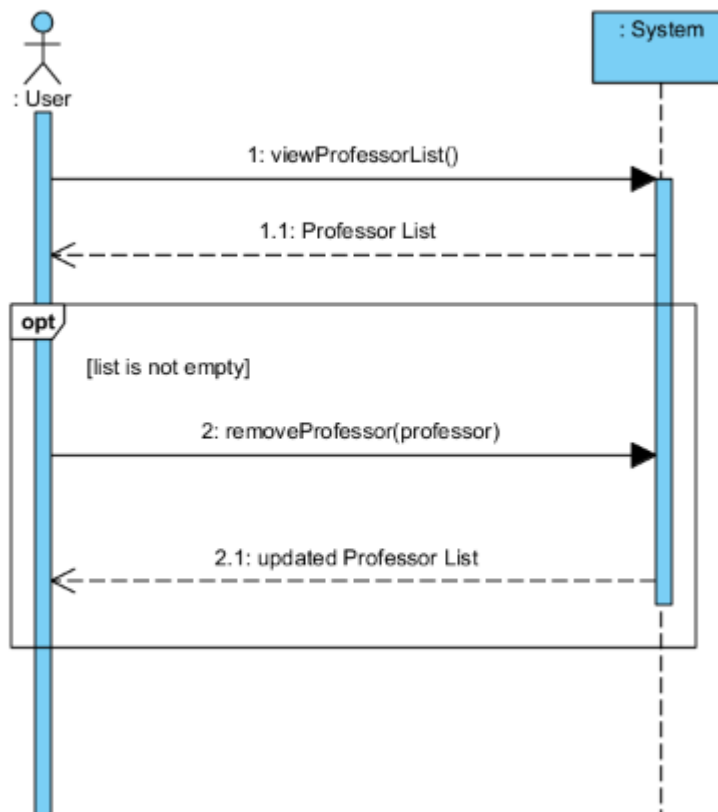
System
+professorLookup(term, mode) +displayProfessors() +selectProfessor(name) +endSearch() +startReview() +startWorkloadRate() +reviewProfessor(text, rating) +rateProfessorWorkload(rating) +closeReview()

ID: UC Save Professor
Scope: Professor Lookup and List of saved professors
Level: User Goal
Actors: User - Student interacting with system
Preconditions: User is logged into the website, and has successfully looked up a list of professors
Flow of events: <ol style="list-style-type: none"> 1. The user selects a professor from the search results 2. The system displays a page with that professor's info 3. The user selects the "Save Professor" icon on the professor's window 4. If the professor is not currently stored in the user's "Saved Professors" list, then <ol style="list-style-type: none"> a. The system displays a notification that the professor has been added to the end of the user's list 5. Else, <ol style="list-style-type: none"> a. The system displays a notification that the professor has already been saved to the list
Postconditions/Success: The user's "Saved Professors" list has one more professor at the end of the list, if it is not already in the list



ID: UC Remove Professor
Scope: List of saved professors and Professor Lookup
Level: User Goal
Actors: User - Student interacting with system
Preconditions: User must be logged into the website
Flow of events: <ol style="list-style-type: none"> 1. The user opens up their "Saved Professors" list 2. The system displays the user's list 3. If their list is not empty, then <ol style="list-style-type: none"> a. The user selects the "Remove Professor" icon next to a professor in their list b. The system removes that professor from the user's list and displays the updated list to the user
Postconditions/Success: The user's "Saved Professors" list has one less professor, or remains empty if originally empty

sd [Remove professor]

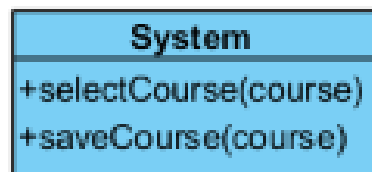
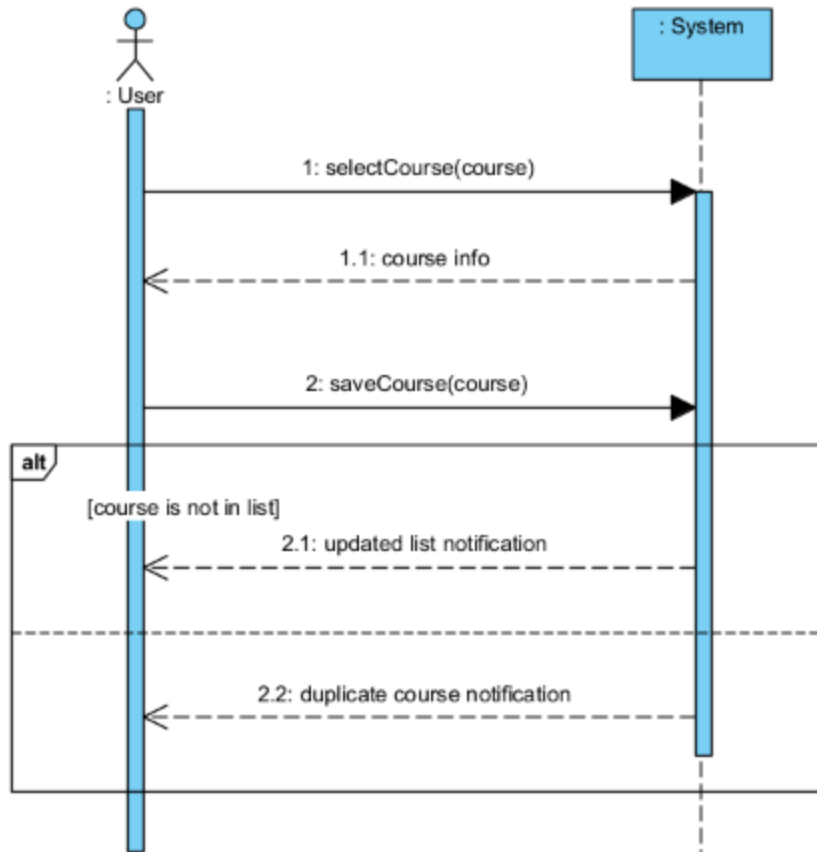


System
+viewProfessorList()
+removeProfessor(professor)

ID: UC Save Changes Exit Message
Scope: User Account Changes
Level: User Goal
Actors: User - Student interacting with system
Preconditions: User must be logged into the website
Flow of events: <ol style="list-style-type: none"> 1. User clicks the "Save Changes?" button 2. System checks if any changes made to "Saved Professors" list <ol style="list-style-type: none"> a. If yes, overwrites the previous "Saved Professors" list and saves for user 3. System checks if any changes made to "Saved Courses" list <ol style="list-style-type: none"> a. If yes, overwrites the previous "Saved Courses" list and saves for user
Postconditions/Success: User's changes to "Saved Professors" and "Saved Courses" lists are saved by the system System displays a "Save Successful" message

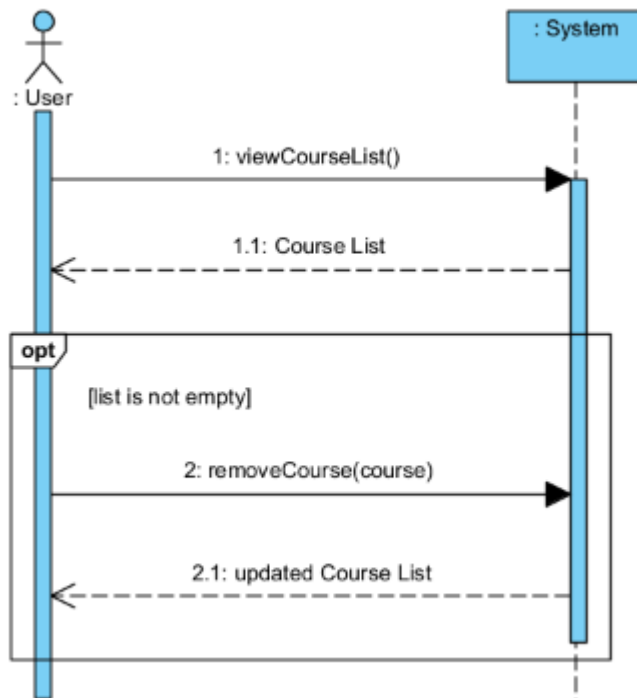
ID: UC Save Course
Scope: Course Lookup and List of saved courses
Level: User Goal
Actors: User - Student interacting with system
Preconditions: User is logged into the website, and has successfully looked up a list of courses
Flow of events: <ol style="list-style-type: none"> 1. The user selects a course from the search results 2. The system displays a page with that course's info 3. The user selects the "Save Course" icon on the course's window 4. If the course is not currently in the user's "Saved Courses" list, then <ol style="list-style-type: none"> a. The system displays a notification that the professor has been added to the end of the user's list 5. Else, <ol style="list-style-type: none"> a. The system displays a notification that the course has already been saved to the list
Postconditions/Success: The user's "Saved Courses" list has one more course at the end of the list, if it is not already in the list

sd [Save course]



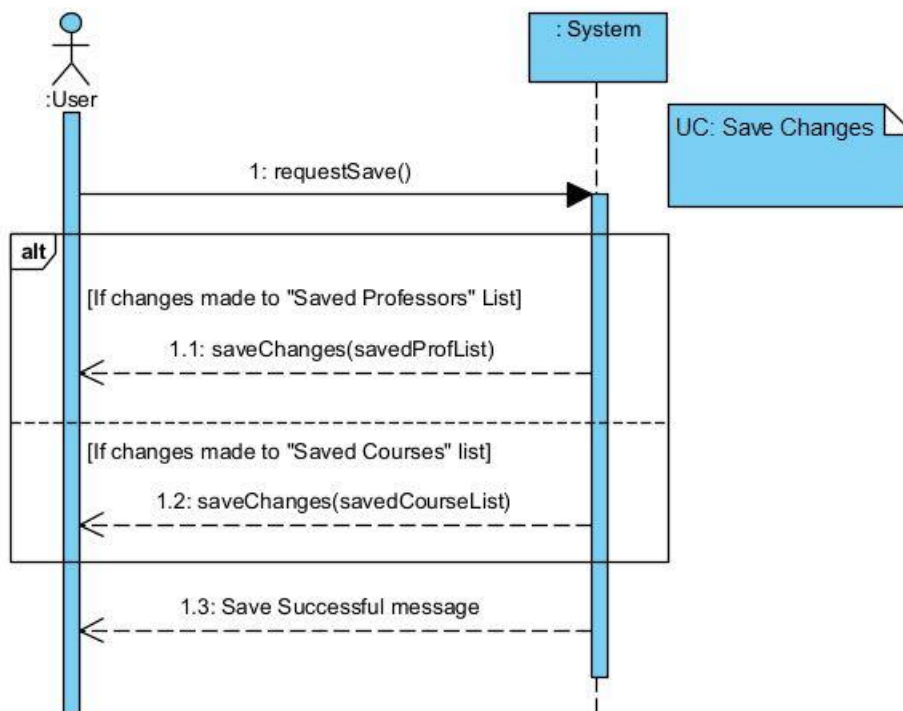
ID: UC Remove Course
Scope: List of saved courses and Course Lookup
Level: User Goal
Actors: User - Student interacting with system
Preconditions: User must be logged into the website
Flow of events: <ol style="list-style-type: none"> 1. The user opens up their "Saved Courses" list 2. The system displays the user's list 3. If their list is not empty, then <ol style="list-style-type: none"> a. The user selects the "Remove Course" icon next to a course in their list b. The system removes that course from the user's list and displays the updated list to the user
Postconditions/Success: The user's "Saved Courses" list has one less course, or remains empty if originally empty

sd [Remove course]



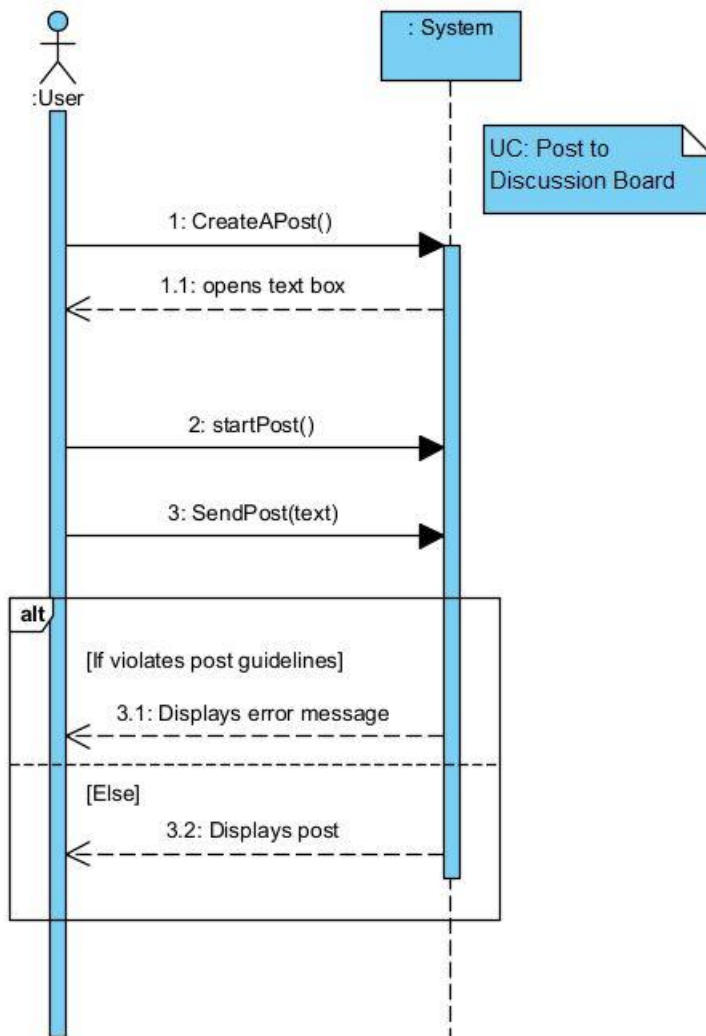
System
+viewCourseList()
+removeCourse(course)

ID: UC Save Changes Exit Message
Scope: User Account Changes
Level: User Goal
Actors: User - Student interacting with system
Preconditions: User must be logged into the website
Flow of events: <ol style="list-style-type: none"> 1. User clicks the "Save Changes?" button 2. System checks if any changes made to "Saved Professors" list <ol style="list-style-type: none"> a. If yes, overwrites the previous "Saved Professors" list and saves for user 3. System checks if any changes made to "Saved Courses" list <ol style="list-style-type: none"> a. If yes, overwrites the previous "Saved Courses" list and saves for user
Postconditions/Success: User's changes to "Saved Professors" and "Saved Courses" lists are saved by the system System displays a "Save Successful" message



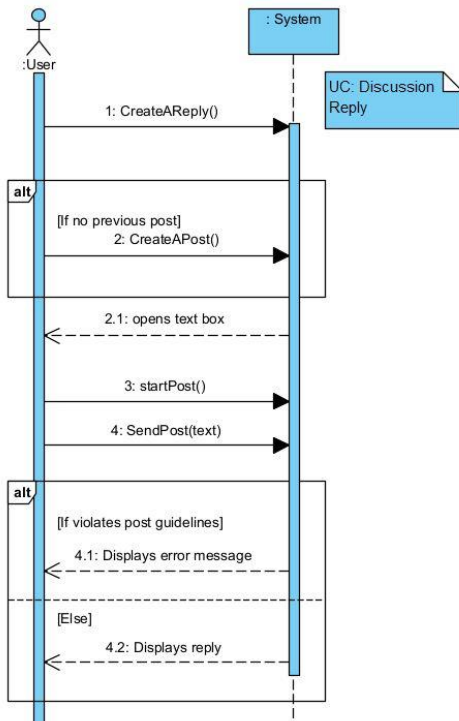
System
+requestSave()
+saveChanges(savedProfList)

ID: UC Post to a Discussion Board
Scope: Discussion Board
Level: User Goal
Actors: User - Student interacting with system
Preconditions: User must be logged into the website User is on the Discussion Board page
Flow of events: <ol style="list-style-type: none"> 1. User clicks the "Create a Post" button 2. System opens a text box 3. User inputs text in the text box 4. User clicks the "Send Post" button <ol style="list-style-type: none"> a. If a post does not follow post guidelines, system will display an error message 5. System displays the post
Postconditions/Success: System displays the post for all users to see



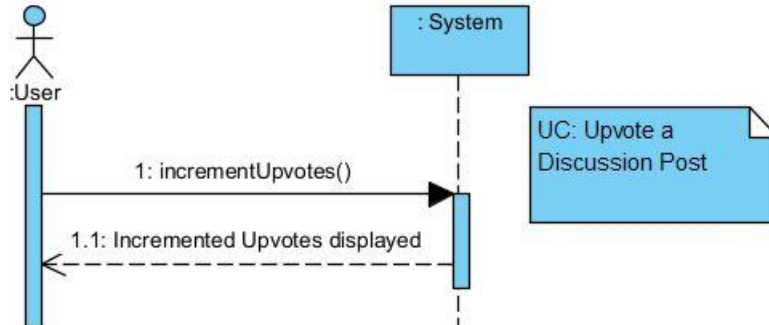
System
+CreateAPost() +startPost() +sendPost(text)

ID: UC Discussion Reply
Scope: Discussion Board
Level: User Goal
Actors: User - Student interacting with system
Preconditions: User must be logged into the website
Flow of events: <ol style="list-style-type: none"> 1. User clicks the "Reply" button below a post <ol style="list-style-type: none"> a. If a previous post does not exist, only button to click will be "Create New Post" 2. System opens a text box 3. User inputs text in the text box 4. User clicks the "Send Reply" button <ol style="list-style-type: none"> a. If a reply does not follow post guidelines, system will display an error message 5. System displays the reply below the original post
Postconditions/Success: System displays the reply under the original post for all users to see



System
+CreateAReply() +CreateAPost() +startPost() +SendPost(text)

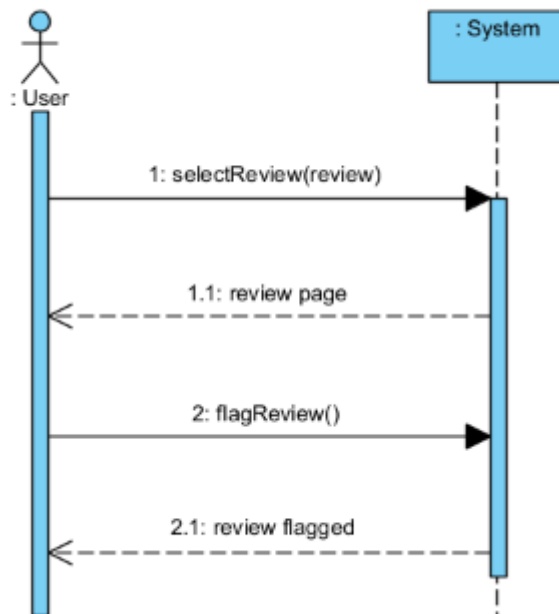
ID: UC Upvote a Discussion
Scope: Discussion Board
Level: User Goal
Actors: User - Student interacting with system
Preconditions: User must be logged into the website A post must exist
Flow of events: <ol style="list-style-type: none"> 1. User clicks the "Upvote" button under a discussion post 2. System increments the amount of Upvotes on the post
Postconditions/Success: System displays the amount of Upvotes under a discussion post



System
+incrementUpvotes()

ID: UC Flag Review
Scope: Professor reviews
Level: User Goal
Actors: User - Student interacting with system Database Service - Site database with reviews
Preconditions: The user has searched for professor reviews from the search professor page of the site
Flow of events: 1. The user selects a review from the professor review search results 2. The system displays a page with the selected review 3. The user selects the "Flag Review" icon on the review page 4. The system updates the flag icon to reflect that the review has been reported to admins
Postconditions/Success: The review is marked as flagged and is reported to the admins

sd [Flag review]



System
+selectReview(review)
+flagReview()

5. Traceability Matrix

Traceability Matrix	The user can create a new account	The user can log in an out of their account	The user can find recommended courses and professors	The user can add a review of a professor	The user can search reviews of professors/courses	The user can filter search results	The user can request a review to be removed	The user can have a list of saved professors	The user can have a list of saved courses	The system saves or deletes changes upon logout	The user can post questions and answers in the discussion board	The user can reply to discussions	The user can upvote questions in the discussion board
UC Create Account	x												
UC Log-in		x											
UC Log out		x											
UC Reset Password		x											
UC Recommendations			x										
UC Filter Results						x							
UC Professor Lookup					x		x						
UC Review Professor				x									
UC Rate Professor Workload				x									
UC Add Professor								x		x			
UC Remove Professor								x		x			
UC Add Course									x	x			
UC Remove Course									x	x			
UC Save Changes								x	x				
UC Post Discussion											x		
UC Reply Discussion												x	
UC Upvote Discussion													x
UC Flag Review							x						

6. Wireframes

Chart My Course

[Home](#)[Reviews](#)[Q&A](#)[Planning](#)[Log in](#)[Sign up](#)

My Courses

My Professors

Log In

Email:

Password:

[Forgot My Password](#)

[Sign up](#)

[Log in](#)

Sign Up

Name:

Username:

Email:

Password:

[Log in](#)

Create Account

Chart My Course

[Home](#)[Reviews](#)[Q&A](#)[Planning](#)[Log in](#)[Sign up](#)

Select Professor

Select Filter

Chart My Course

[Home](#)[Reviews](#)[Q&A](#)[Planning](#)[Log in](#)[Sign up](#)

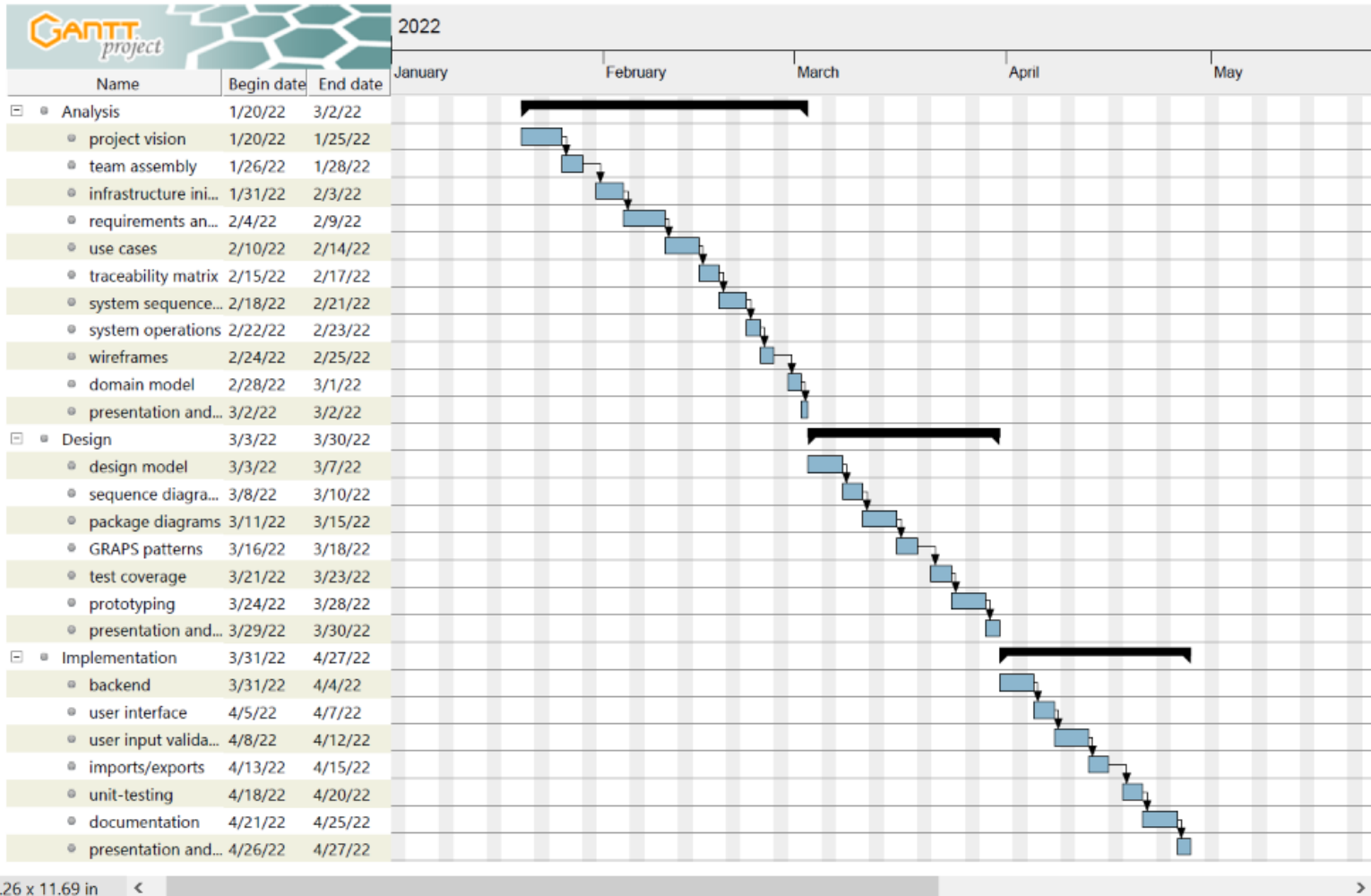
Search

Planning

Recommended Courses

Recommended Professors

7. Gantt Diagram



8. Jira

Our team uses Jira for task and time tracking – it is accessible at <https://chartmycourse.atlassian.net>.

9. Website

Our team maintains a website for content relevant to the project, found at <https://hkaase.github.io/ChartMyCourse/>

10. Git Link

Our Git repository can be found at <https://github.com/hkaase/ChartMyCourse>.