

# Brogrammers

Elaboration Specification

April 28th, 2021

Julie Dang, Alexander Overley, Rylan Jaquess, Walker Nicholson

## Table of Contents

1. System Requirements
2. Use Case Diagrams
3. Trace Matrix
4. Use Cases
5. Sequence Diagrams
6. Class Diagram
7. Database Design
8. Data Definitions
9. User Interface Navigation Diagram
10. Screen Layouts
11. Physical Architecture Design
12. Design Procedures for Security Concerns and Non-functional Requirements
13. Gantt Chart
14. Elaboration Phase Prototypes

# System Requirements

## Functional and Non-functional System Requirements:

A functional requirement is a statement of what the system should do-- it is a statement of system function. Below is a hierarchical list of functional requirements for the system:

1. The COBWeb system shall validate and enroll new users who are students, alumni, or members of the community.
2. The COBWeb system shall have a log-in and log-out for users to log-in and out of the system.
3. The COBWeb system shall pair students and mentors through a matching process.
4. The COBWeb system shall have a donation page to collect donations.
5. The COBWeb system shall have the ability for mentors to post jobs on the job board.
6. The COBWeb system shall have a discussion board for users to communicate.
7. The COBWeb system shall allow reports sent to system administrators anonymously.
8. The COBWeb system shall have announcements for the system administrator to create, edit, and delete.
9. The COBWeb system shall have an email blast for the system administrator to send out to users of the system.
10. The COBWeb system shall display an events page for users to see upcoming or past events held by the department.
11. The COBWeb system shall have a photo carousel and a photo board for users to upload, edit, and delete photos.
12. The COBWeb system shall have surveys for the system administrator to create, edit, and delete.
13. The COBWeb system shall have a backup.
14. The COBWeb system shall have a recovery for the backup.

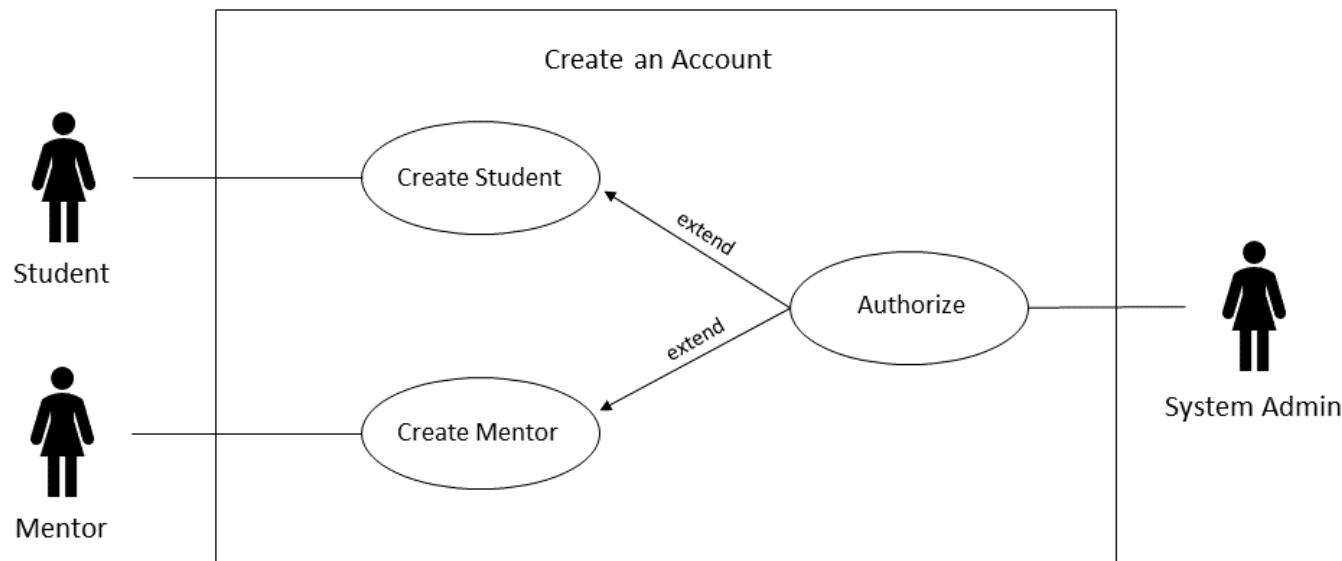
A non-functional requirement is a constraint placed on the system. Below is a hierarchical list of non-functional requirements for the system:

1. The COBWeb system shall be created on a content management system called WordPress.
2. The COBWeb system shall be secured by a firewall.
3. The COBWeb system shall use SQL Server for database management.
4. The COBWeb system shall utilize a payment system called Braintree in order to collect donations.
5. The COBWeb system shall be user-friendly.
6. The COBWeb system shall be accessible 24/7.
7. The COBWeb system shall be backed-up automatically.
8. The COBWeb system shall have system administrator functionalities.

# Use-Case Diagrams

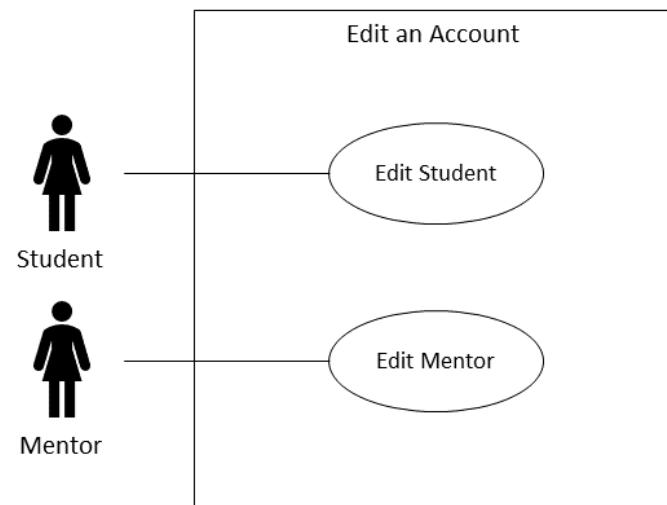
## Use Case IDs 01 and 04: Create Student and Create Mentor

This use-case diagram displays a student and a mentor creating an account for the system. The system will allow eligible students and mentors to create an account in order to gain access to the system. The system administrator will be the one to authorize user accounts.



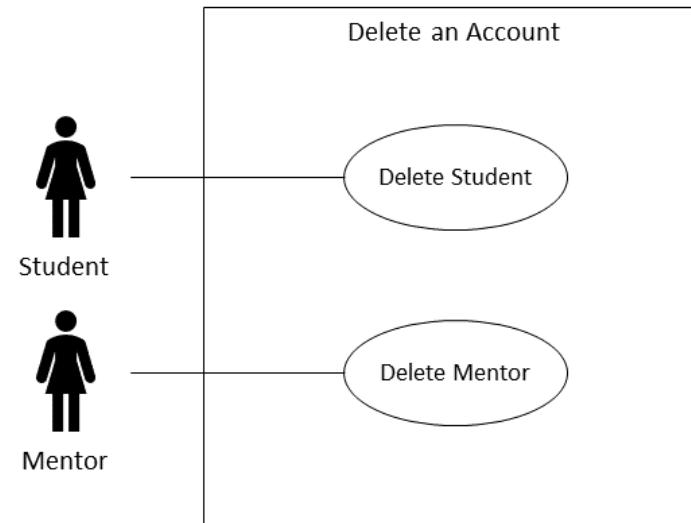
**Use Case IDs 02 and 05: Edit Student and Edit Mentor**

This use-case diagram displays a student and a mentor editing their accounts within the system. The system will allow students and mentors who own an account to edit their accounts to display information that they want to be displayed.



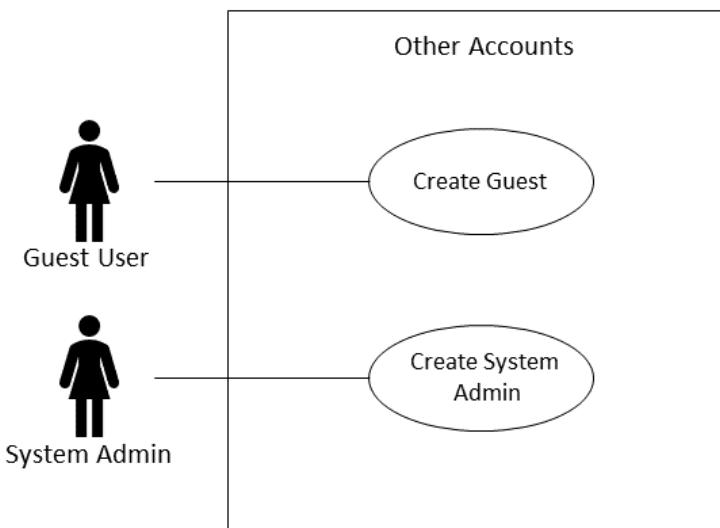
**Use Case IDs 03 and 06: Delete Student and Delete Mentor**

This use-case diagram displays a student and a mentor deleting their accounts in the system. The system will allow students and mentors who own an account to delete their accounts whenever necessary.



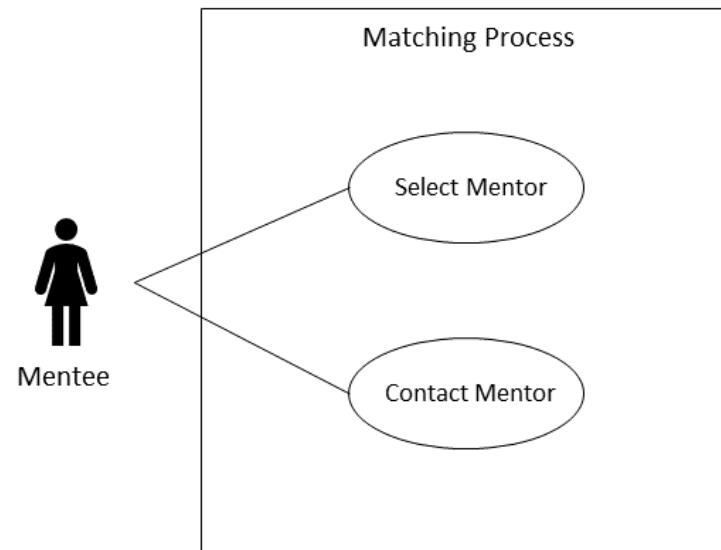
**Use Case IDs 07 and 08: Create Guest and Create System Administrator**

This use-case diagram displays a guest user and a system administrator creating an account for the system. The system will allow any guest users to create a guest account in order to make a donation. The system will also allow authorized personnel to create system administrator accounts.



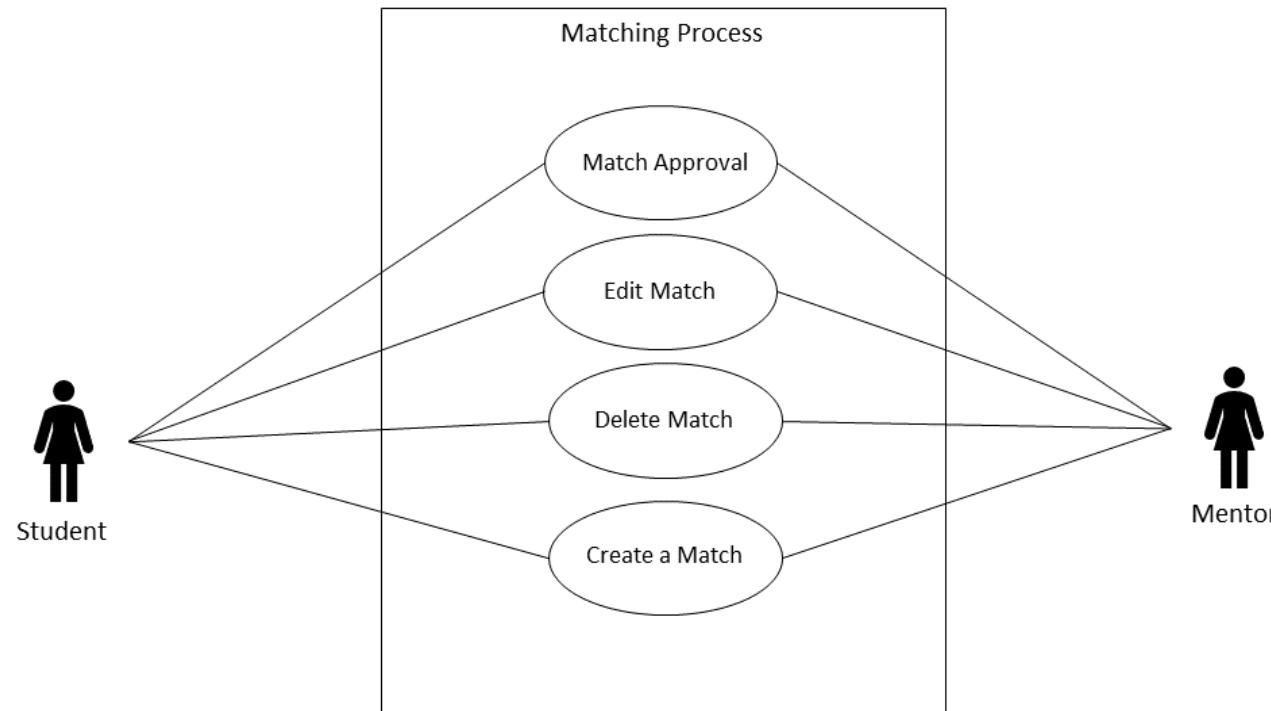
**Use Case IDs 09 and 10: Select Mentor and Contact Mentor**

The primary actor in this use-case diagram is the mentee (or student). The mentee is shown selecting a mentor and contacting their mentor. The system will allow mentees who go through the matching process to select a mentor and contact them as well.



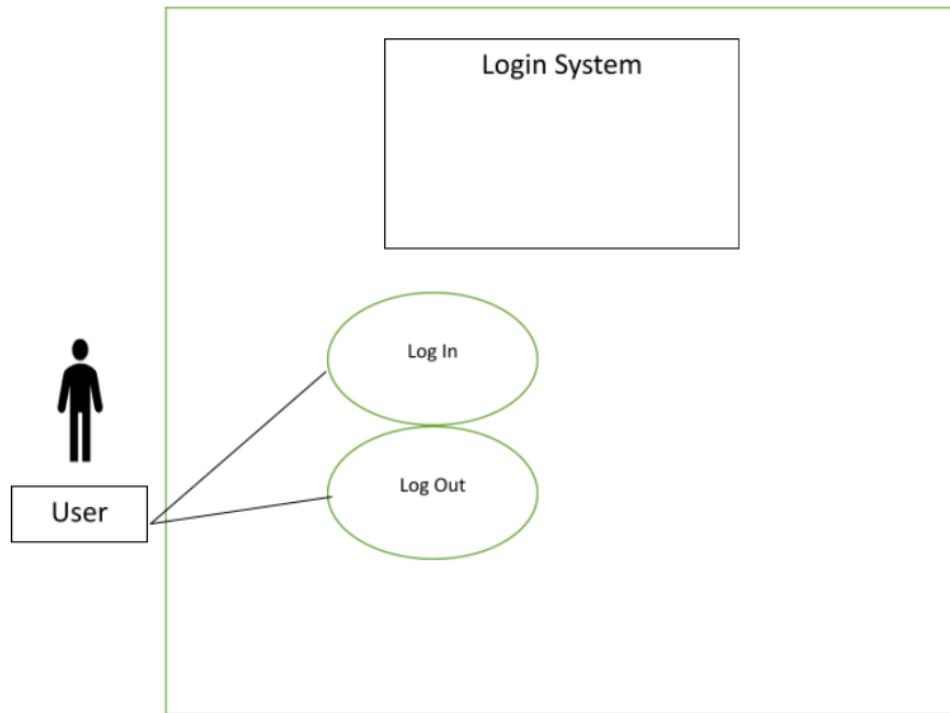
**Use Case IDs 11-14: Match Approval, Edit Match, Delete Match, and Create a Match**

The primary actors in these use cases are the mentee (student) and mentors. The system will allow mentees and mentors to find and approve their match, edit their match, and delete their match.



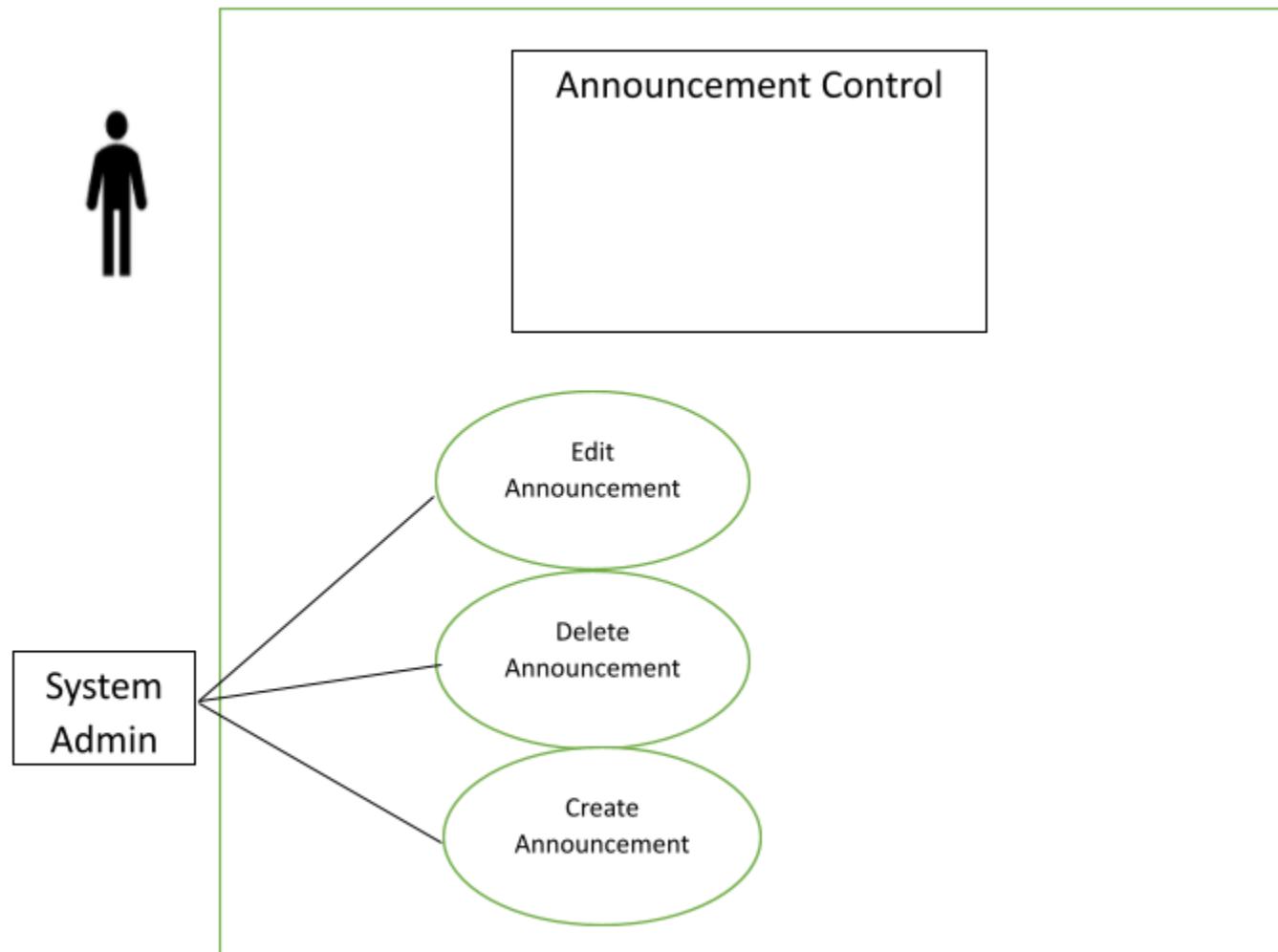
**Use Case IDs 15 and 16: Log In and Log Out**

This use case displays a user selecting log in and log out. This represents the user interacting with a button that will either verify their account credentials and log them in. Or, if the user is already logged in, the system will log them out.



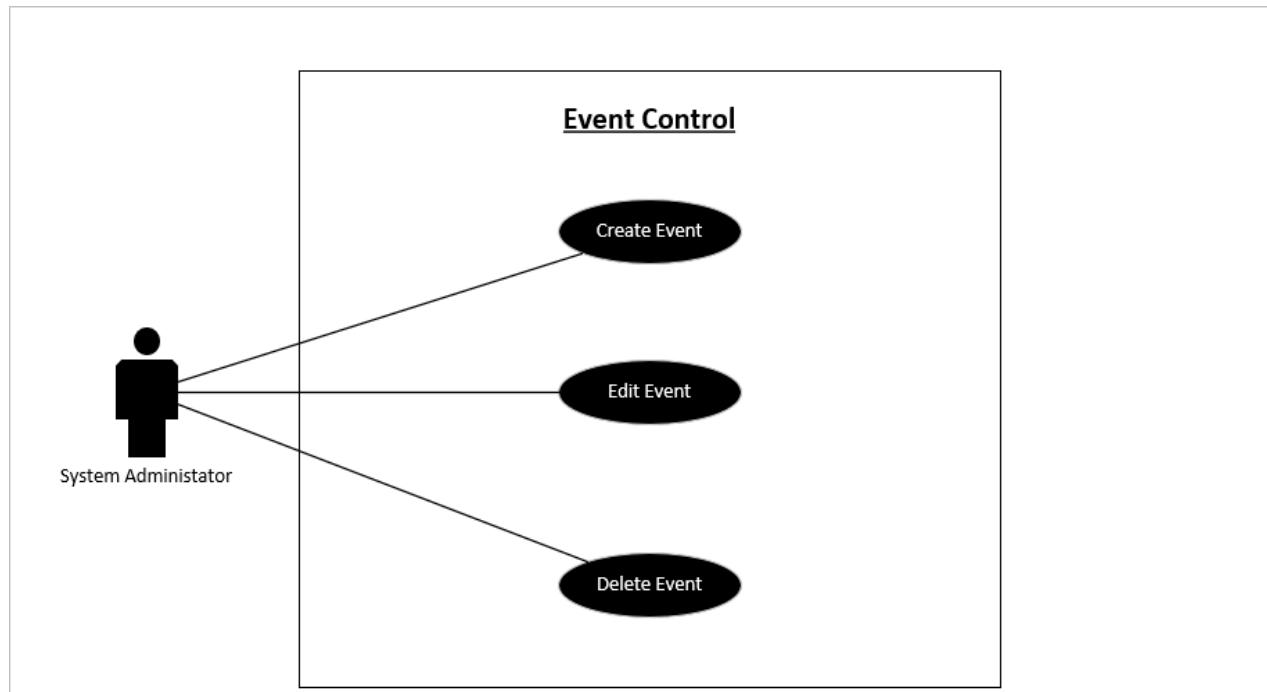
**Use Case IDs 17-19: Edit, Delete, and Create Announcements**

This use case displays a system administrator having the option to edit, delete, and create announcements. The system administrator needs to have the ability to post a global message and be able to remove it. They also need the ability to change it.



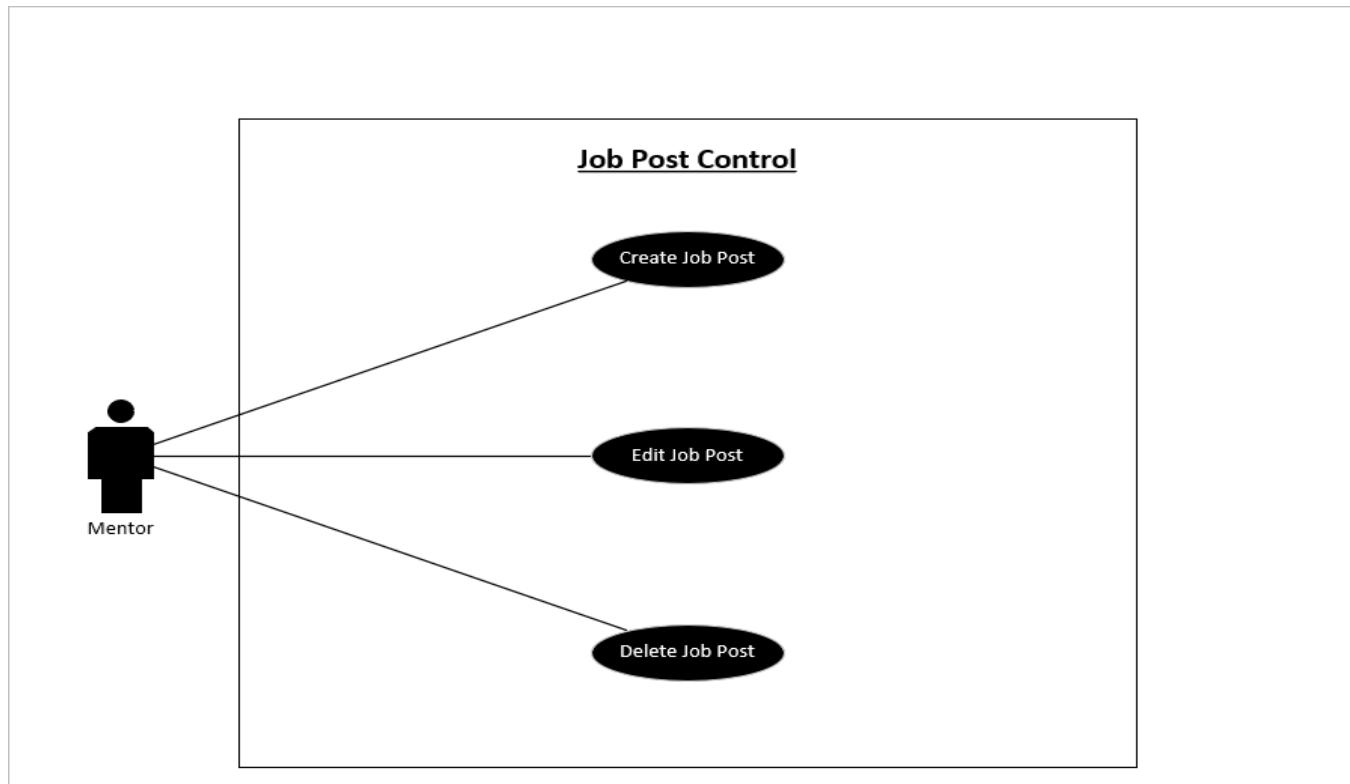
**Use Case IDs 20-22: Create, Edit, and Delete Event**

This use case displays a system administrator having the option to create, edit, and delete events. The system administrator needs to have the ability to post an event globally and be able to change it.



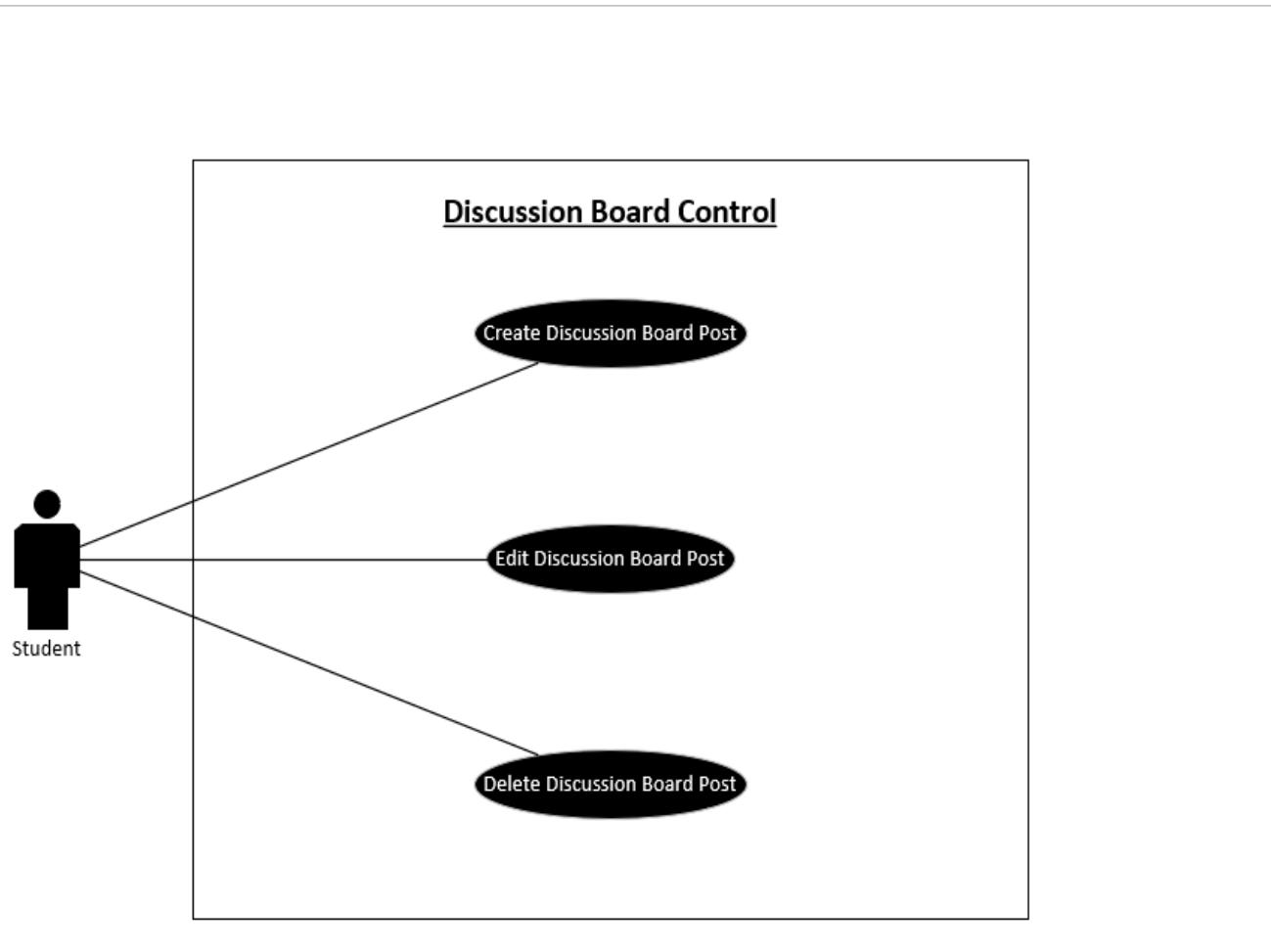
**Use Case ID 23-25: Create, Edit, Delete Job Post**

This use-case diagram shows how a mentor has the ability to create, edit, and delete a job posting to the job board.



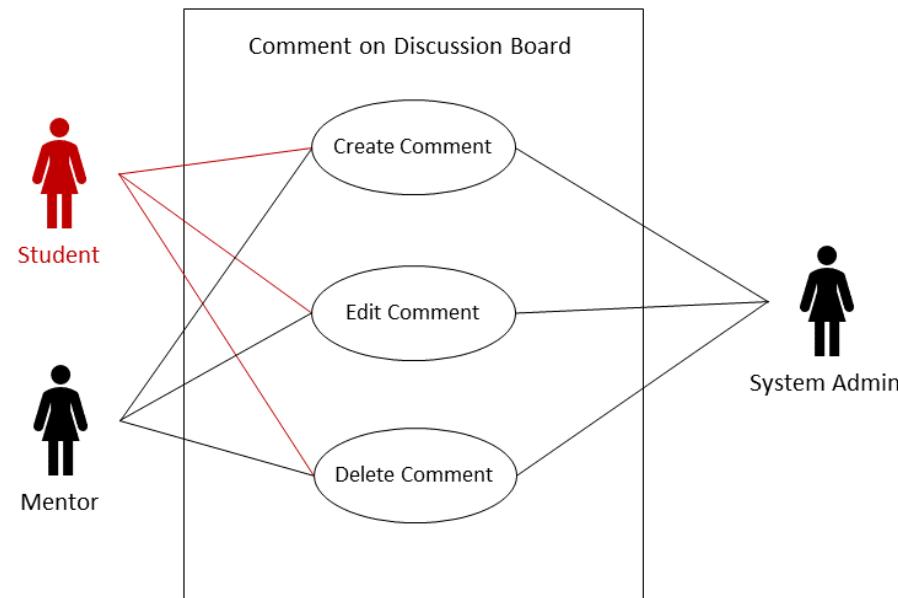
**Use Case ID 26-28: Create, Edit, and Delete Discussion**

This use-case diagram shows how a student will have the ability to create, edit, and delete their discussion board posts. This will be a convenient feature to have on the discussion board. If the student created a typo, they will be able to edit their post. If the student's question has already been answered, they can delete their post if they want.



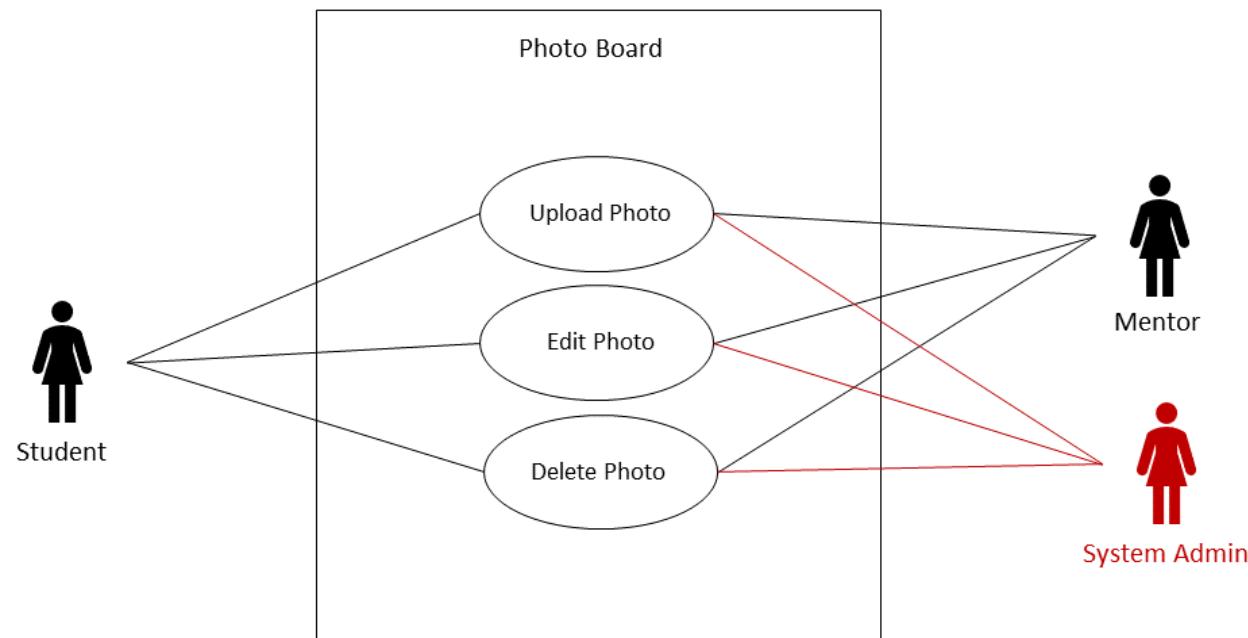
**Use Case IDs 29-31: Comment on Discussion, Edit Comment on Discussion, and Delete Comment on Discussion**

The primary actors in these use-cases are authorized users of the system: students, mentors, and system administrator. The system will allow these users to create comments on discussion board posts, edit the comment, or delete the comment.



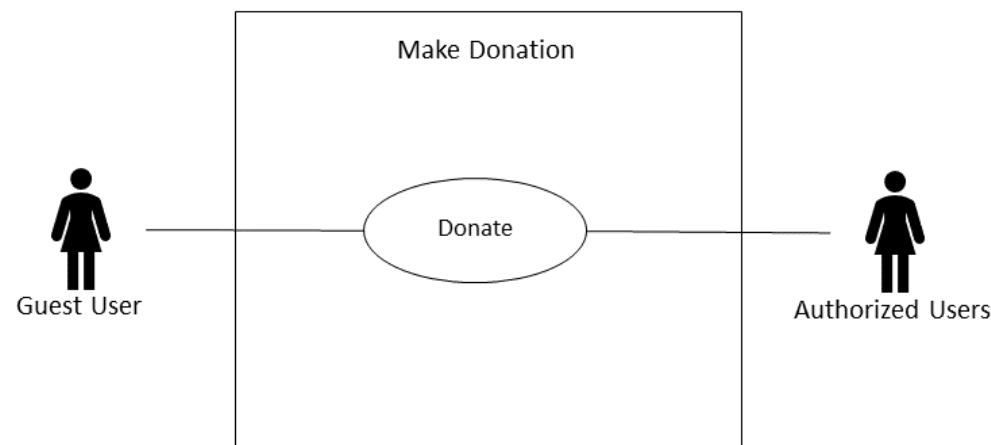
**Use Case IDs 32-34: Upload, Edit, and Delete Photos**

This use-case diagram shows all authorized users of the system (student, mentor, and system administrator) uploading, editing, and deleting photos on the photo board. The system will allow users to post, edit, and delete their photos on the photo board.



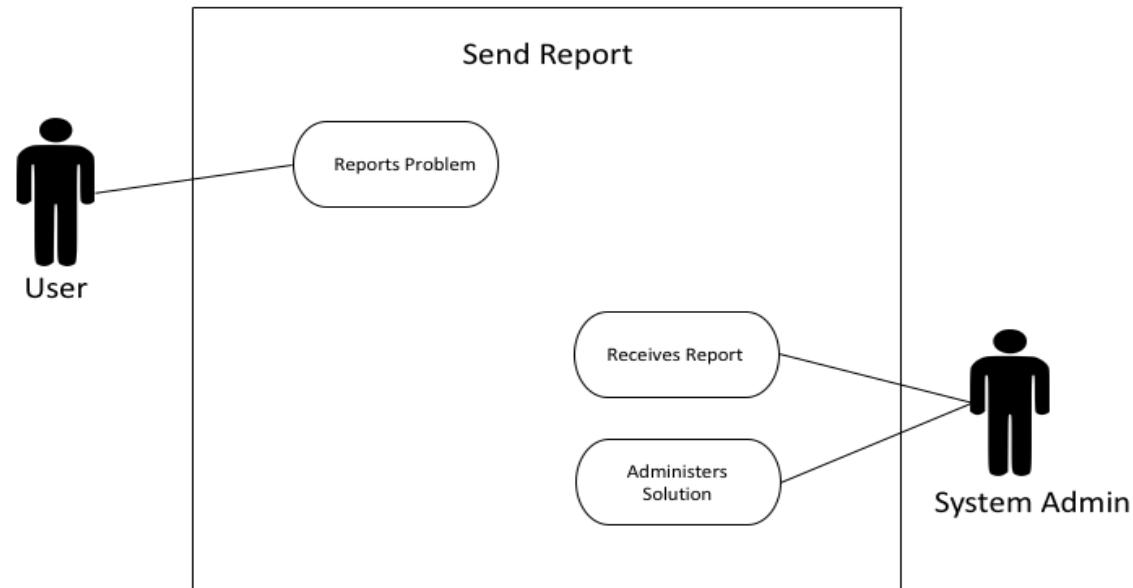
**Use Case ID 35: Donate**

This use-case diagram shows users of the system making a donation. The system will allow users to make donations.



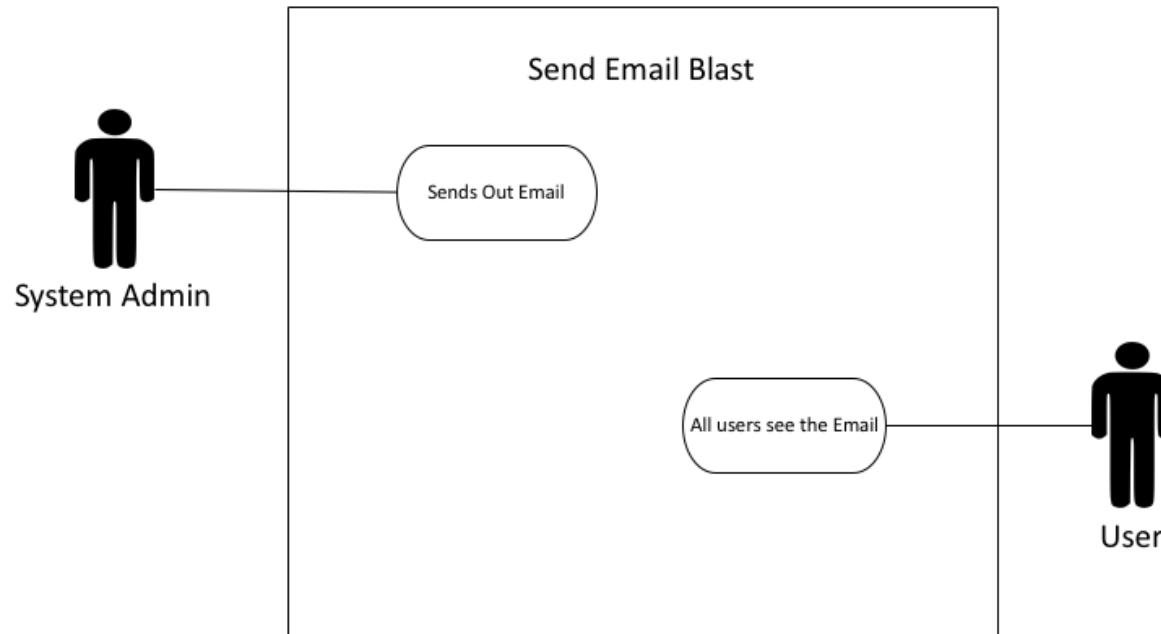
**Use Case ID 36: Send report**

This use-case diagram shows a user reporting a problem to the system administrator. The system will allow the system administrator to view the report and will then administer a solution to the problem.



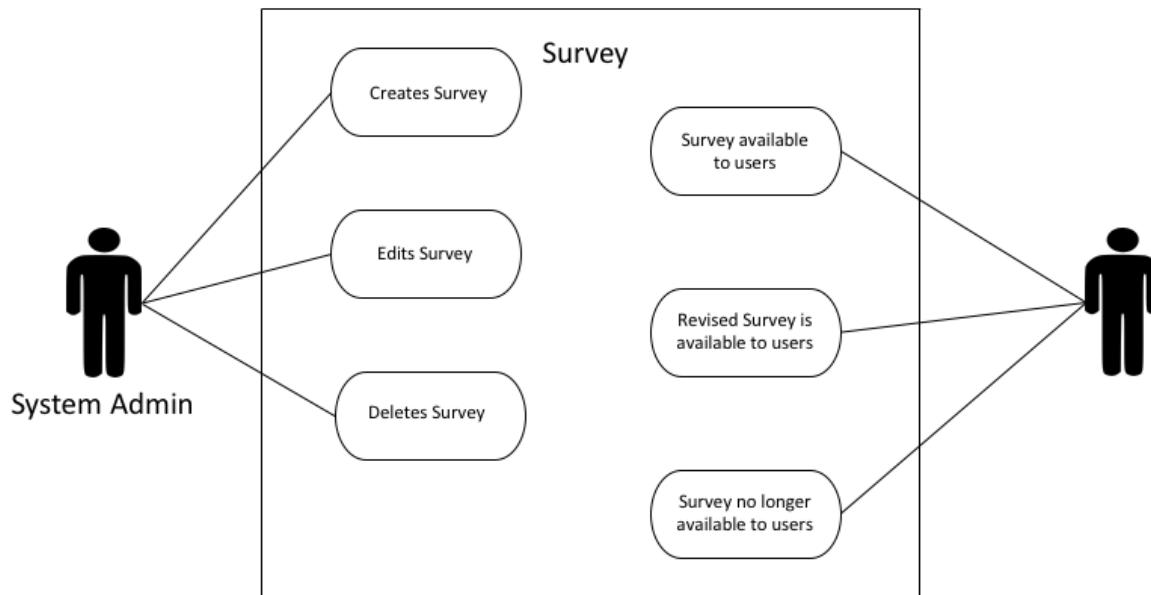
**Use Case ID 37: Send Email Blast**

This use-case diagram shows a system administrator sending out an email blast. The system allows the email to be sent to all users of the system.



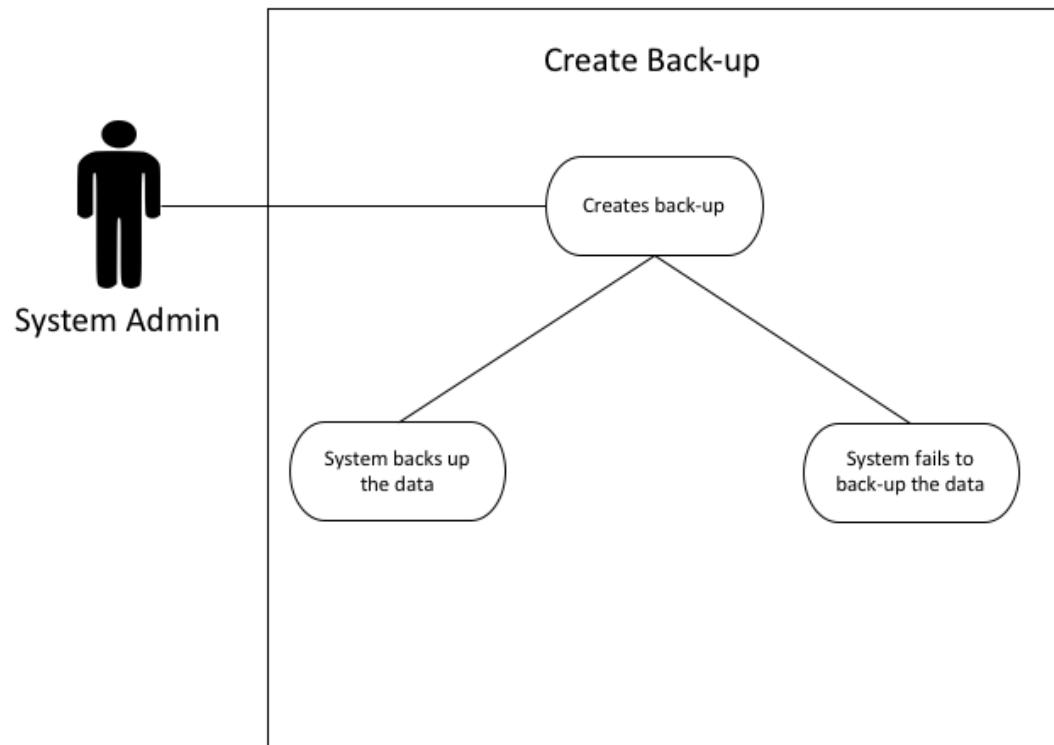
**Use Case IDs 38-40: Create Survey, Edit Survey, Delete Survey**

This use-case diagram shows the system administrator creating, editing and deleting a survey. It also shows the users having the ability to take the original survey, the revised survey, and no longer able to take a survey. The system allows the system administrator to create, edit and delete a survey that is seen by all users.



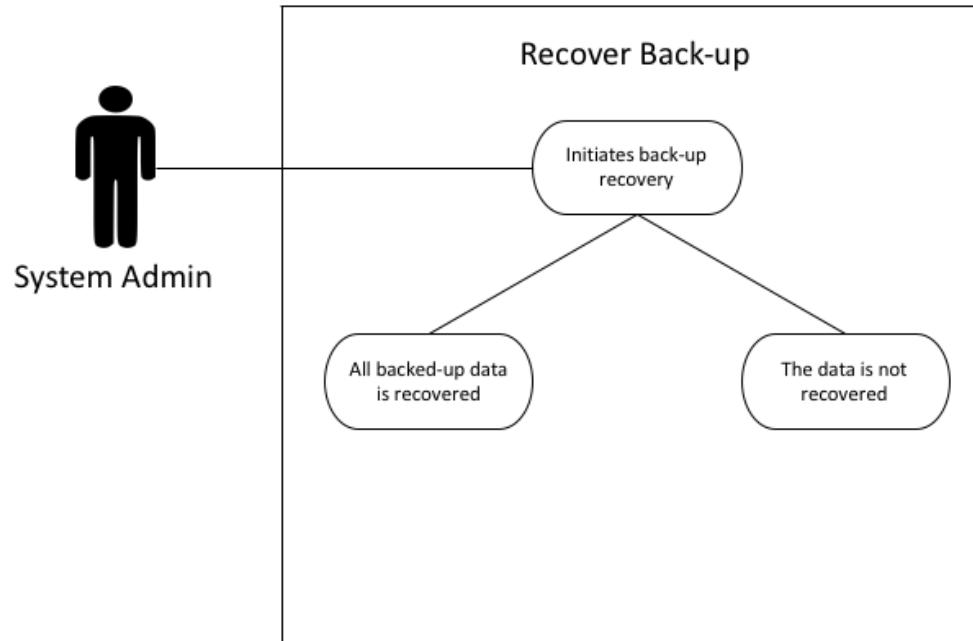
**Use Case ID 41: Create Back-up**

This use-case diagram shows a system administrator creating a data back-up. The system either backs up the data, or fails to back-up the data due to insufficient storage.



**Use Case ID 42: Recover Back-up**

This Use-Case diagram shows a system administrator initiating a back-up recovery. The system either recovers the data, or does not recover the backed-up data.



# Trace Matrix

This trace matrix is used to ensure that each functional requirement for the system is matched with at least one use case. The shaded green cells represent the intersection of a use case and a functional system requirement.

# Use-Cases & Sequence Diagrams

Use cases are a sequence of actions and activities performed by actors to make the system do what they want it to do. The following will lay out the use cases, a brief description of each activity, and the actors for each case:

1. **Create Student:** the system will need to allow **students** to sign up for an account.
2. **Edit Student:** **students** who sign up will be able to edit their account.
3. **Delete Student:** **students** will have the ability to delete their accounts.
4. **Create Mentor:** the system will need to allow **mentors** to sign up for an account.
5. **Edit Mentor:** **mentors** who sign up will be able to edit their account.
6. **Delete Mentor:** **mentors** will have the ability to delete their accounts.
7. **Create Guest:** the system will allow **users** without an authorized account to access the donation page and the about us page.
8. **Create System Administrator:** the system will allow **system administrators** to create an account.
9. **Select Mentor:** **students** will be able to select their mentors from a pool of mentors.
10. **Contact Mentor:** **students** will have access to their mentor's email address by clicking into their mentor's profile.
11. **Match Approval:** **students** will contact their **mentor** and wait for a match approval.
12. **Edit Match:** **student** and **mentor** will have the option to change their match if both parties agree to change.
13. **Delete Match:** **student** and **mentor** will have the option to delete their match if both parties agree to delete.
14. **Create Match:** the system will acknowledge when a match is made between a **student** and **mentor**.
15. **User Log-In:** the system will allow **users** to log-in.
16. **User Log-Out:** the system will allow **users** to log-out.
17. **Create Announcement:** the system will allow the **system administrator** to create announcements.
18. **Edit Announcement:** the system will allow the **system administrator** to edit their announcements.
19. **Delete Announcement:** the system will give the **system administrator** to delete their announcements.
20. **Create Events:** the system will allow the **system administrator** to create events that will be displayed on the events page.
21. **Edit Events:** the system will allow the **system administrator** to edit events on the events page.
22. **Delete Events:** the system will allow the **system administrator** to delete events on the events page.
23. **Create Job Post:** the system will allow **mentors** to post job postings to the job board.
24. **Edit Job Post:** the system will allow **mentors** to edit their job post on the job board.
25. **Delete Job Post:** the system will allow **mentors** to delete their job post on the job board.
26. **Create Discussion:** the system will allow **users** to post to the discussion board.

27. **Edit Discussion:** the system will allow **users** to edit their post on the discussion board.
28. **Delete Discussion:** the system will allow **users** to delete their post on the discussion board.
29. **Comment on Discussion:** the system will allow **users** to comment on the post from the discussion board.
30. **Edit Comment on Discussion:** the system will allow **users** to edit their comment(s) on the post from the discussion board.
31. **Delete Comment on Discussion:** the system will allow **users** to delete their comment(s) on the post from the discussion board.
32. **Upload Photos:** the system will allow **users** to upload photos to the photo board.
33. **Edit Photos:** the system will allow **users** to edit their photos posted on the photo board.
34. **Delete Photos:** the system will allow **users** to delete their photos posted on the photo board.
35. **Donate:** the system will allow **users** to donate to the program from the donations page.
36. **Send Report:** the system will allow **users** to anonymously report any problems to the **system administrator**.
37. **Send Email Blast:** the system will allow the **system administrator** to send email blasts to users.
38. **Create Survey:** the system will allow the **system administrator** to create surveys.
39. **Edit Survey:** the system will allow the **system administrator** to edit surveys and for intended **users** to fill out the survey.
40. **Delete Survey:** the system will allow the **system administrator** to delete the survey when they deem necessary or it will expire when the time runs out.
41. **Create Backup:** the system will need to allow the **system administrator** to create a backup of all the data or create a specified time frame to backup data automatically.
42. **Recover Backup:** the system will need to allow the **system administrator** to recover its most recent backup.

# Use Case Specification: Create Student

## 1. Use-Case: Create Student ID: 01

### Brief Description

This use-case will allow students to create an account for the system.

## 2. Flow of Events

### Basic Flow

- Student will click the create a student account button
- Student will create username
- Student will create password
- Student will enter first name
- Student will enter middle initial if applicable
- Student will enter last name
- Student will upload profile picture if they want
- Student will identify whether they are a CIS student at UofL
- Student will identify their major concentration
- Student will identify their gender
- Student will identify their ethnicity
- Student will enter their preferred email address
- Student will identify class year
- Student will enter birthdate

### Alternative Flow

- Account is not created
- Student is not a validated user

## 3. Pre-conditions

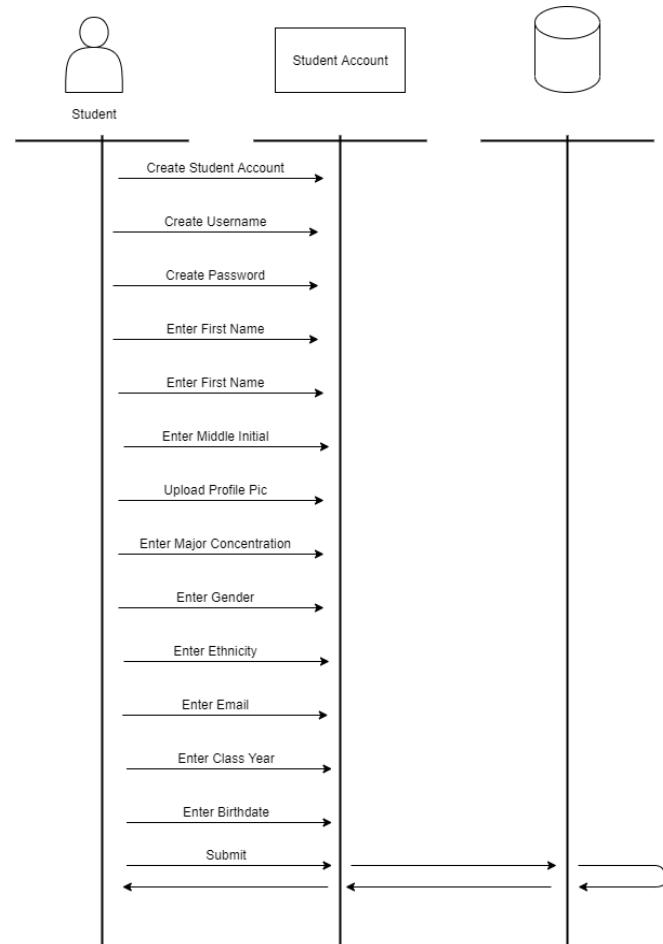
Student must be part of the Computer Information Systems department at the University of Louisville.

## 4. Post-conditions

Student must be approved by the system administrator to gain access to the system.  
Student account is created.

## 5. Extension Points

System administrator must accept the student in order to create an account.



# Use Case Specification: Edit Student

## 1. Use-Case: Edit Student ID: 02

### Brief Description

This use-case will allow students to edit their profiles in the system.

## 2. Flow of Events

### Basic Flow

- Student will click on edit profile
- Student may edit username
- Student may edit password
- Student may edit first name
- Student may edit middle initial
- Student may edit last name
- Student may edit profile picture
- Student may edit major concentration
- Student may edit their gender
- Student may edit their preferred email address
- Student may edit their class rank

### Alternative Flow

- Student will not edit their profile

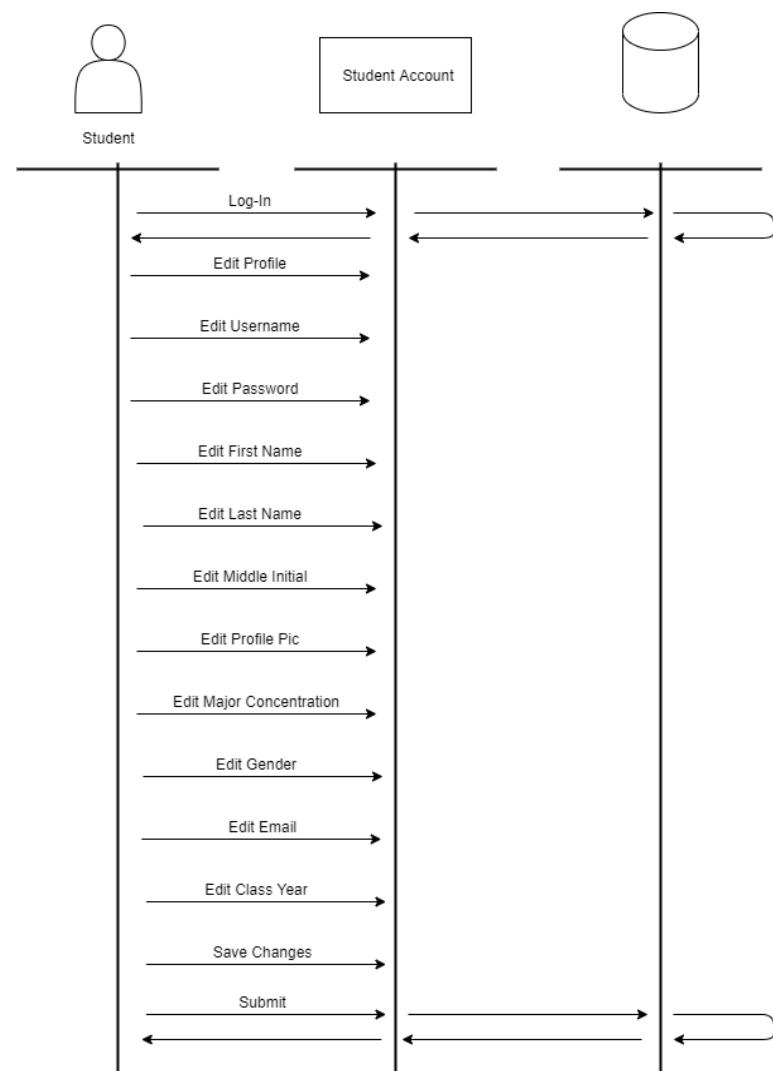
## 3. Pre-conditions

Student must be logged in to the system to edit their profile.

Student must have an account in order to edit their profile.

### Post-conditions

Student profile is edited.



# Use Case Specification: Create Mentor

## 1. Use-Case: Create Mentor ID: 04

### Brief Description

This use-case will allow mentors to create an account for the system.

## 2. Flow of Events

### Basic Flow

- Mentor will click create a mentor account button
- Mentor will create a username
- Mentor will create a password
- Mentor will enter first name
- Mentor will enter middle initial if applicable
- Mentor will enter last name
- Mentor will upload profile picture if they want
- Mentor will identify whether upperclassman
- Mentor will identify whether alumni
- Mentor will identify whether member of community
- Mentor will enter their preferred email address
- Mentor will identify their ethnicity
- Mentor will identify their gender
- Mentor will enter their birthdate

### Alternative Flow

- Account is not created
- Mentor is not a validated user

## 3. Pre-conditions

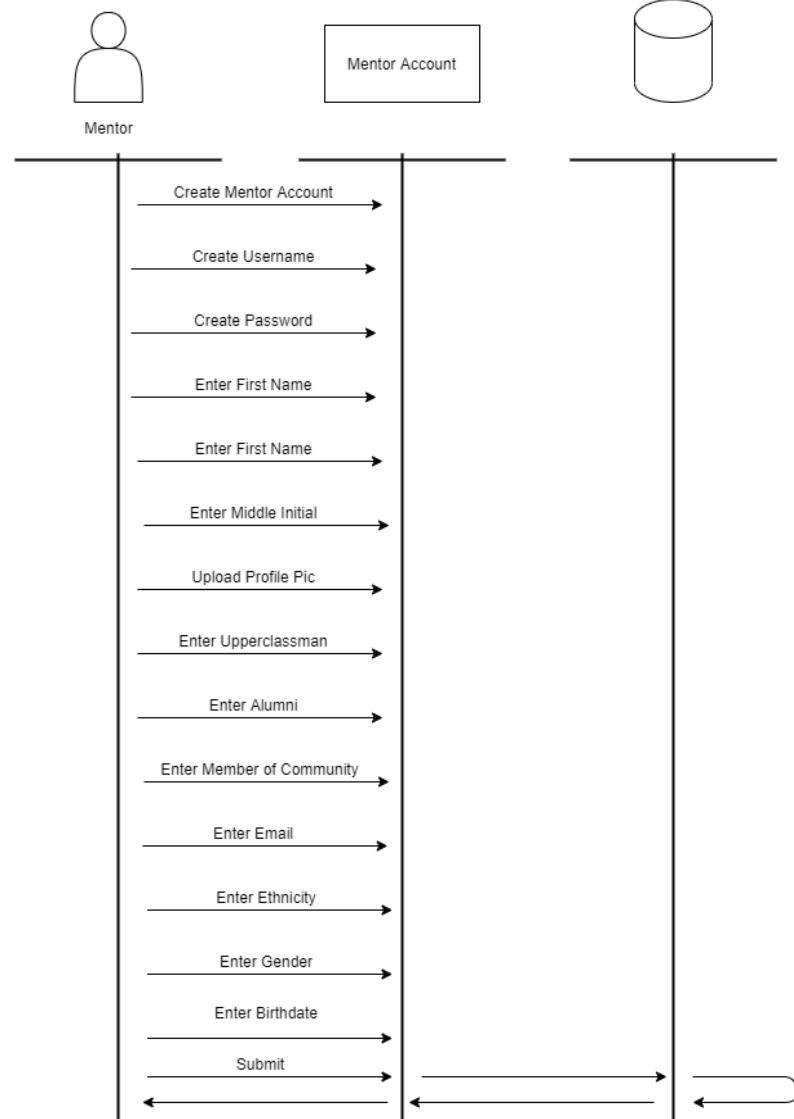
Mentor must be either an upperclassman or alum from the Computer Information Systems department at the University of Louisville.

Mentor may also be a member of the community.

## 4. Post-conditions

Mentor must be approved by the system administrator.

Mentor account is created.



## 5. Extension Points

System administrator must accept a mentor in order to create an account.

# Use Case Specification: Edit Mentor

### 1. Use-Case: Edit Mentor ID: 05

#### Brief Description

This use-case will allow mentors to edit their profiles in the system.

### 2. Flow of Events

#### Basic Flow

- Mentor will click on edit profile button
- Mentor may edit username
- Mentor may edit password
- Mentor may edit first name
- Mentor may edit middle initial
- Mentor may edit last name
- Mentor may edit profile picture
- Mentor may edit whether they are upperclassmen or alum
- Mentor may edit their preferred email address
- Mentor may edit their gender

#### Alternative Flow

- Mentor will not edit their profile

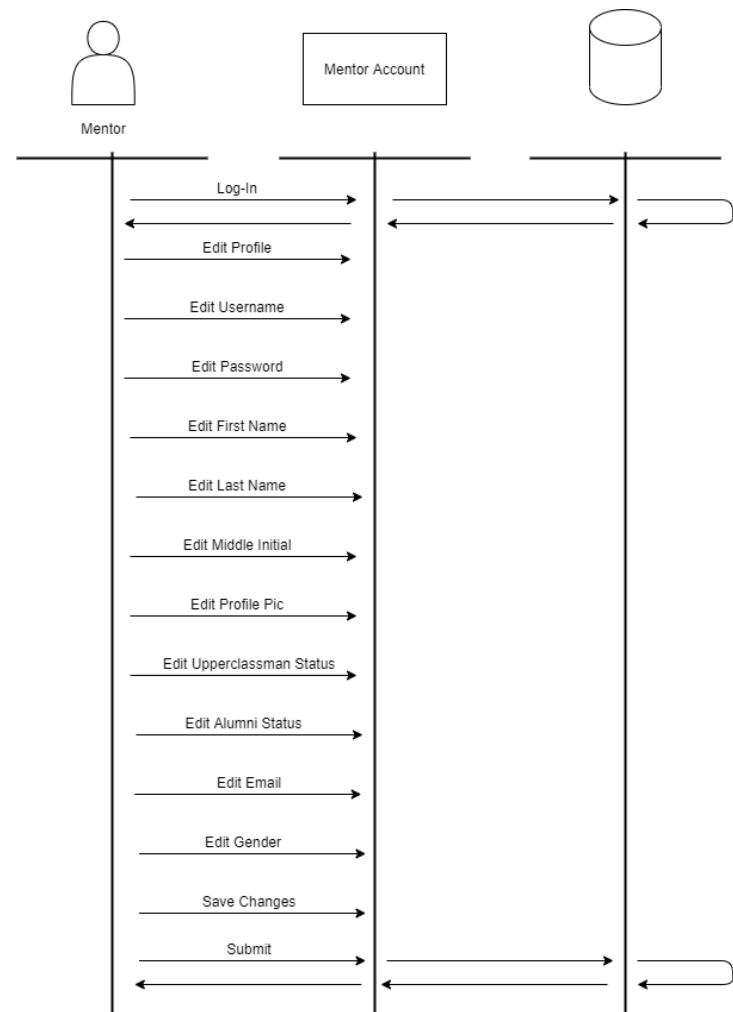
### 3. Pre-conditions

Mentor must have an account in order to edit their profile.

Mentor must be logged in to the system in order to edit their profile.

#### Post-conditions

Mentor profile is edited.



## Use Case Specification: Delete Student

### 1. Use-Case: Delete Student ID: 03

#### Brief Description

This use-case will allow students to delete their profile from the system.

### 2. Flow of Events

#### Basic Flow

- Student will click on edit profile
- Student will click on delete profile
- Student will click to confirm deletion of profile

#### Alternative Flow

- Student will not delete profile

### 3. Pre-conditions

Student must be logged in to the system.

Student profiles must be created in order to delete from the system.

#### Post-conditions

Student account is deleted.

## Use Case Specification: Delete Mentor

### 1. Use-Case: Delete Mentor ID: 06

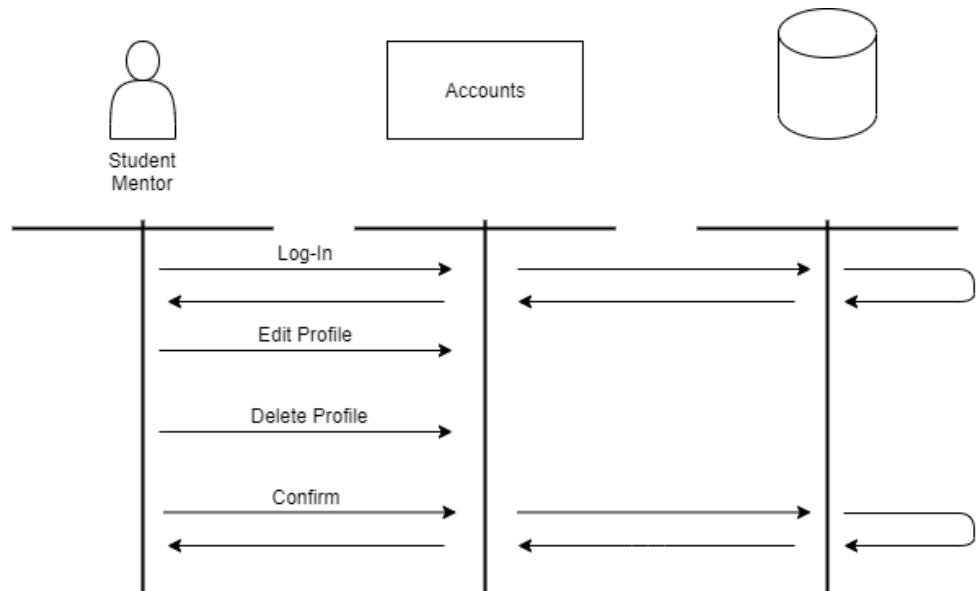
#### Brief Description

This use-case will allow mentors to delete their profile from the system.

### 2. Flow of Events

#### Basic Flow

- Mentor will click on edit profile button
- Mentor will click on delete profile button
- Mentor will click on confirm deletion of profile



**Alternative Flow**

- Student will not edit their profile

**3. Pre-conditions**

Mentor must be logged in to the system.

Mentor profile must be created in order to delete profile from the system.

**Post-conditions**

Mentor account is deleted.

**Use Case Specification: Create Guest****1. Use-Case: Select Mentor ID: 07****Brief Description**

This use-case will allow users who are not wanting to create an account to create a guest account.

**2. Flow of Events****Basic Flow**

- Guest will click on the create guest account on landing page
- Guest will fill out the necessary information on the sign-up form
- Guest will enter their email address
- Guest will enter their first name
- Guest will enter their last name
- Guest will click finish sign-up as guest

**Alternative Flow**

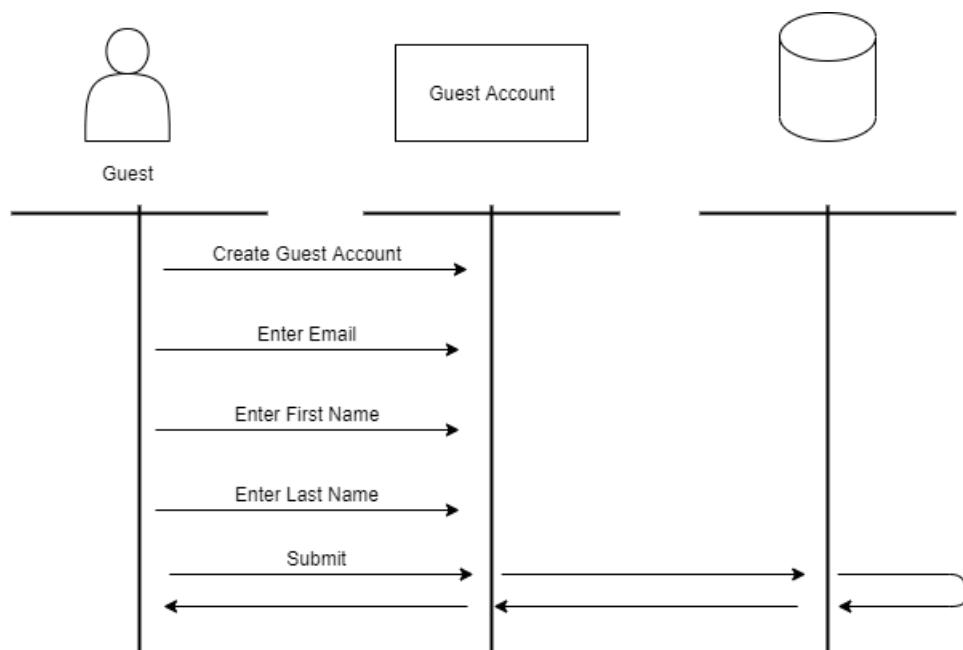
- User does not want to create guest account

**3. Pre-conditions**

Guest user must be wanting to make a donation or is curious about the webpage.

**4. Post-conditions**

Guest users will gain access to donations and about us pages.



## Use Case Specification: Create System Administrator

### 5. Use-Case: Select Mentor ID: 08

### 6. Brief Description

This use-case will allow the system administrators to create an account to access the system.

### 7. Flow of Events

#### Basic Flow

- System administrator will click on the create account on landing page
- System administrator will fill out the necessary information on the sign-up form
- System administrator will enter their email address
- System administrator will enter their first name
- System administrator will enter their last name
- System administrator will create a username
- System administrator will create a password
- System administrator will click finish sign-up

#### Alternative Flow

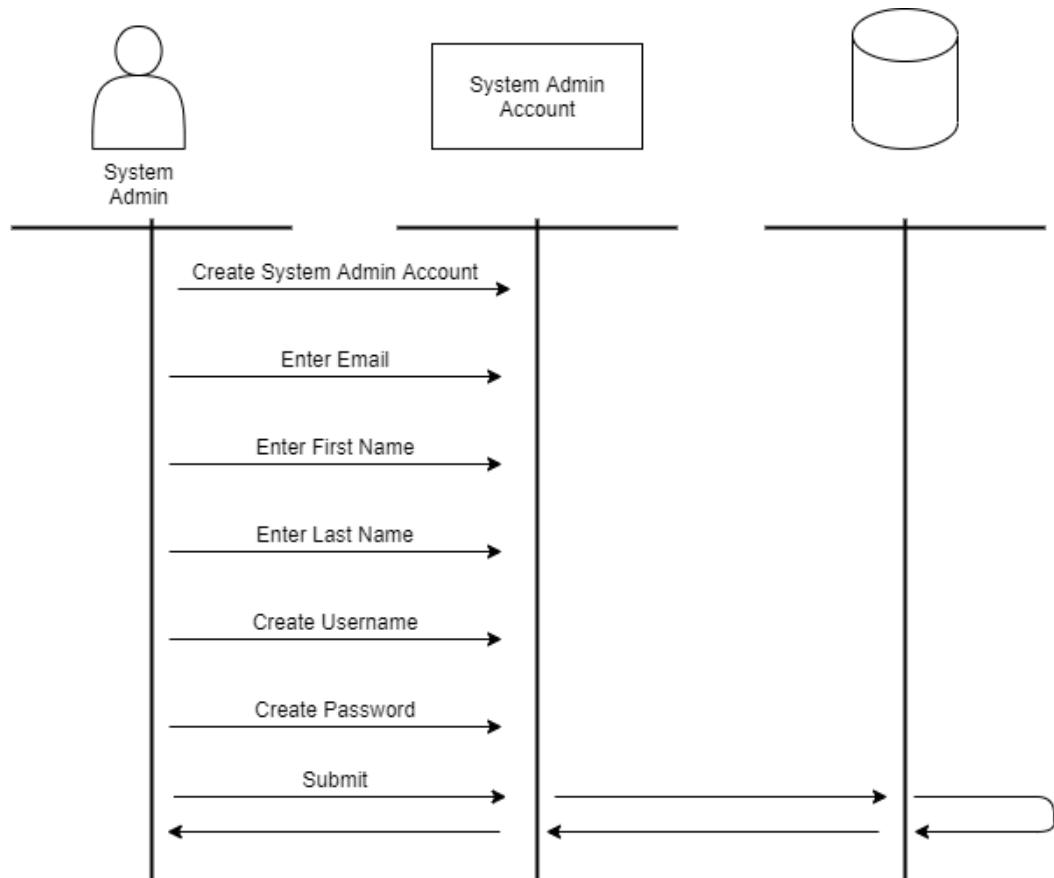
- User is not a system administrator

### 8. Pre-conditions

Users must be a system administrator in order to create this account.

### 9. Post-conditions

System administrator has an account created.



# Use Case Specification: Select Mentor

## 10. Use-Case: Select Mentor ID: 09

### Brief Description

This use-case will allow students to select their mentors.

## 11. Flow of Events

### Basic Flow

- Student will click into matching process
- Student will click on match as mentee
- Student will go through pool of available mentors
- Student will go through each mentor profiles
- Student will read through mentor questionnaires
- Student will choose their mentor

### Alternative Flow

- Student will not select a mentor

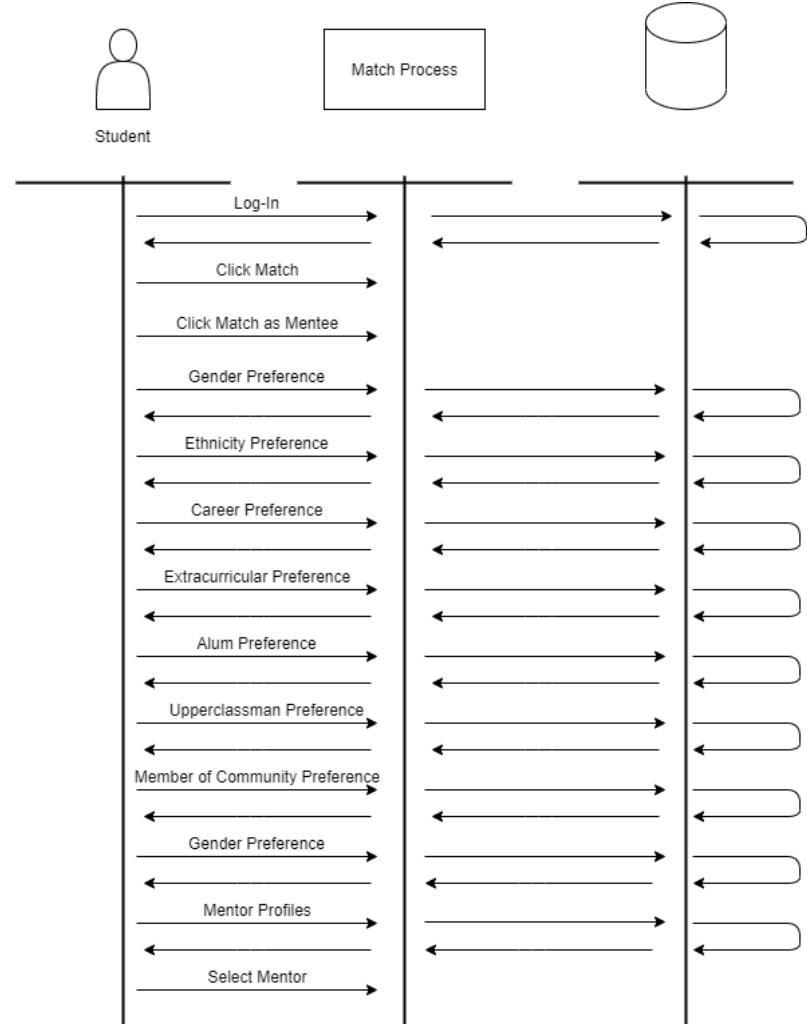
## 12. Pre-conditions

Student must be logged in to the system in order to find the pool of available mentors.

Student must be in the matching process.

## 13. Post-conditions

Student will contact their mentor via email.



## Use Case Specification: Contact Mentor

### 1. Use-Case: Contact Mentor ID: 10

#### Brief Description

This use-case will allow students to contact their mentors.

### 2. Flow of Events

#### Basic Flow

- Student will click into mentor profile
- Student will identify mentor email address
- Student will contact their mentor via email
- Student will notify system administrator of their mentor match

#### Alternative Flow

- Student will not contact their mentor

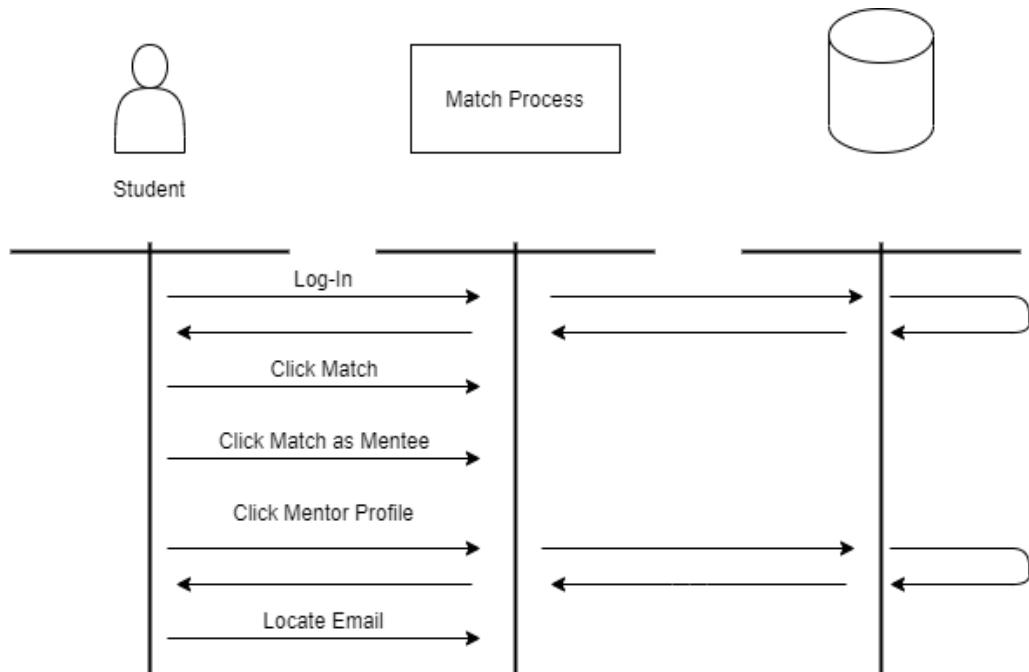
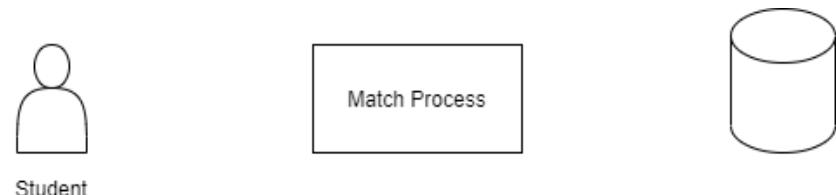
### 3. Pre-conditions

Student must be logged in to the system in order to find their mentor's contact information.

Student must go through the matching process.

### 4. Post-conditions

Student will contact their mentor.



# Use Case Specification: Match Approval

## 1. Use-Case: Match Approval ID: 11

### Brief Description

This use-case will allow students, who have already contacted their mentors and confirmed a match, to be approved for a match.

## 2. Flow of Events

### Basic Flow

- Student will have contacted their mentor about creating a match together
- Mentor will have confirmed the match
- The match is approved after both parties agree to enter the match

### Alternative Flow

- Mentor will deny the match

## 3. Pre-conditions

Student must have an account made.

Mentor must have an account made.

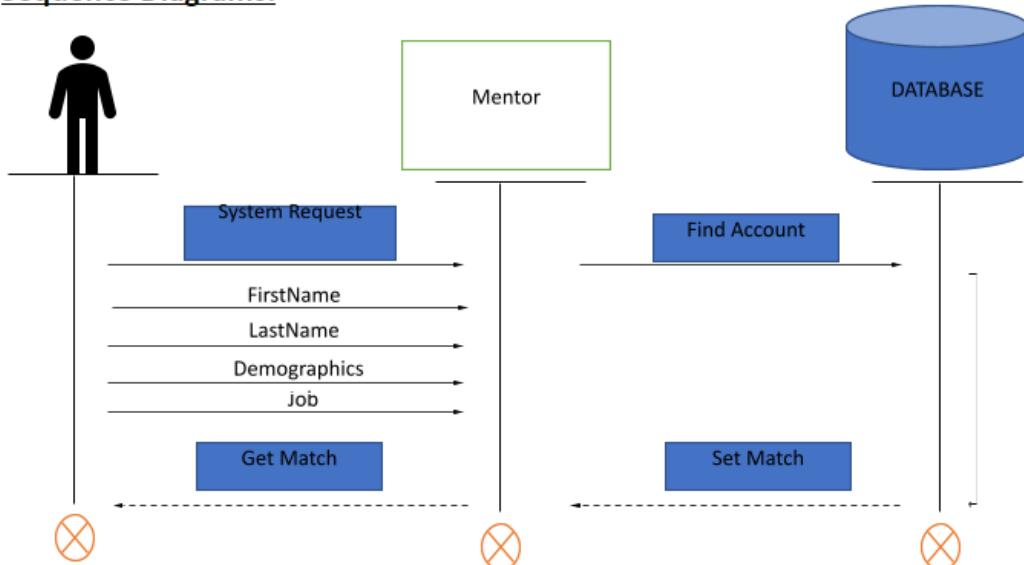
Student must have already contacted their mentor.

Student must have been successful in finding a mentor match.

## 4. Post-conditions

A match has been approved.

### Sequence Diagrams:



# Use Case Specification: Edit Match

## 1. Use-Case: Edit Match ID: 12

### Brief Description

This use-case will allow students to edit their match with their mentor if necessary.

## 2. Flow of Events

### Basic Flow

- Student contacts mentor about reassignment
- Student contacts system administrator about reassignment
- Student goes through matching process again
- Student finds a new mentor
- Student contacts new mentor
- Student must have new mentor confirm new match
- Student enters in a new match

### Alternative Flows

- Student does not edit the match

## 3. Pre-conditions

Student must be matched to a mentor already in order to be edited.

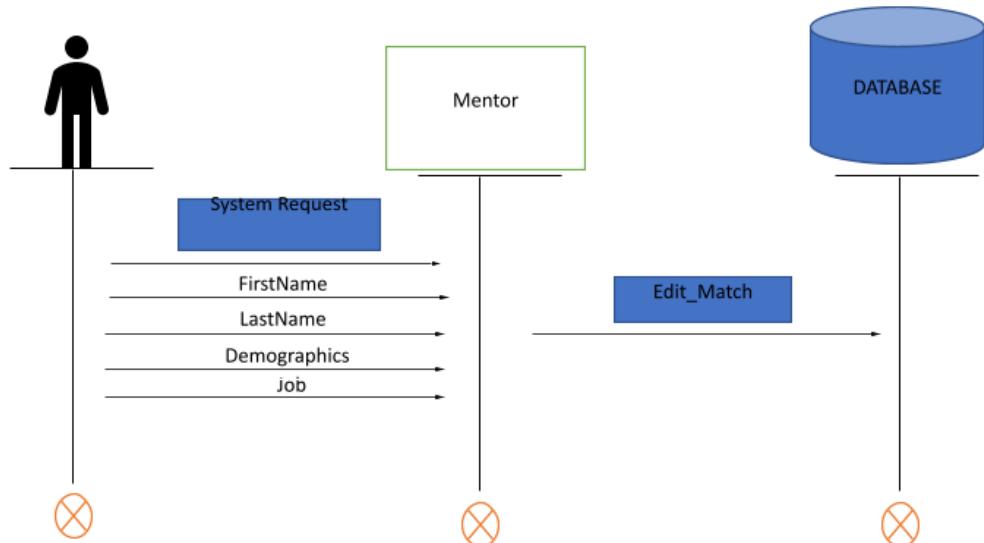
Student is unhappy with the current match.

A request has been made to edit the match.

## 4. Post-conditions

The match is edited.

Student has a new mentor.



# Use Case Specification: Delete Match

## 1. Use-Case: Delete Match ID: 13

### Brief Description

This use-case will allow the system to delete a match when both parties involved agree to delete their match.

## 2. Flow of Events

### Basic Flow

- System will receive a request to delete a match
- System will review request
- System will review relevant party profiles
- System will review any available information provided by parties involved
- System will delete the match

### Alternative Flows

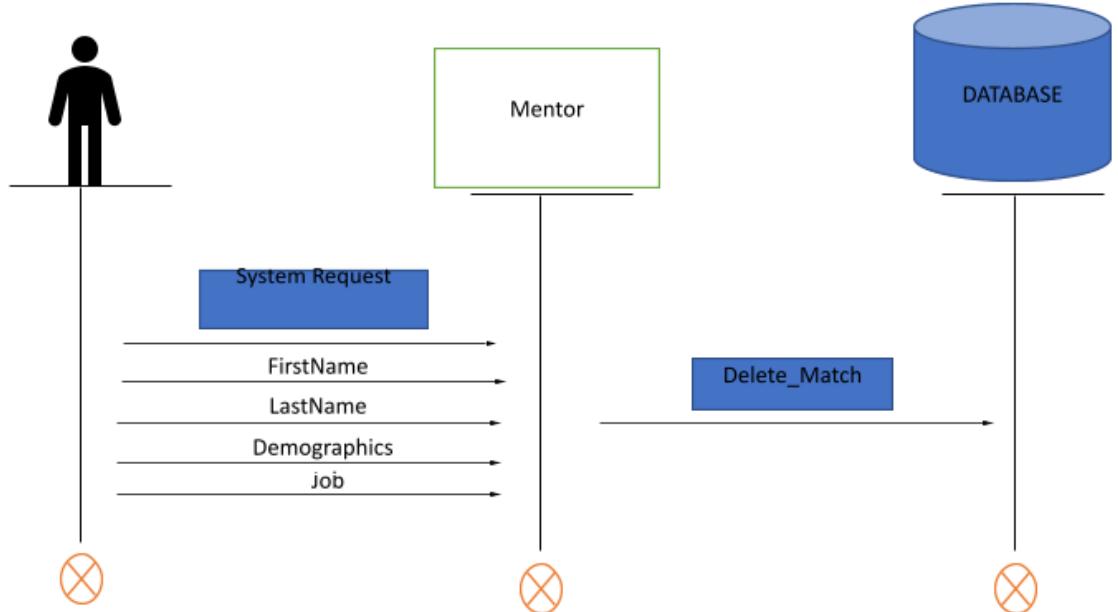
- The system does not delete the match.

## 3. Pre-conditions

Student must be matched to a mentor already in order to be deleted.  
Match deletion has to be requested by one of two users.

## 4. Post-conditions

The match will be deleted.  
The match will not be deleted.



# Use Case Specification: Create a Match

## 1. Use-Case: Create a Match ID: 14

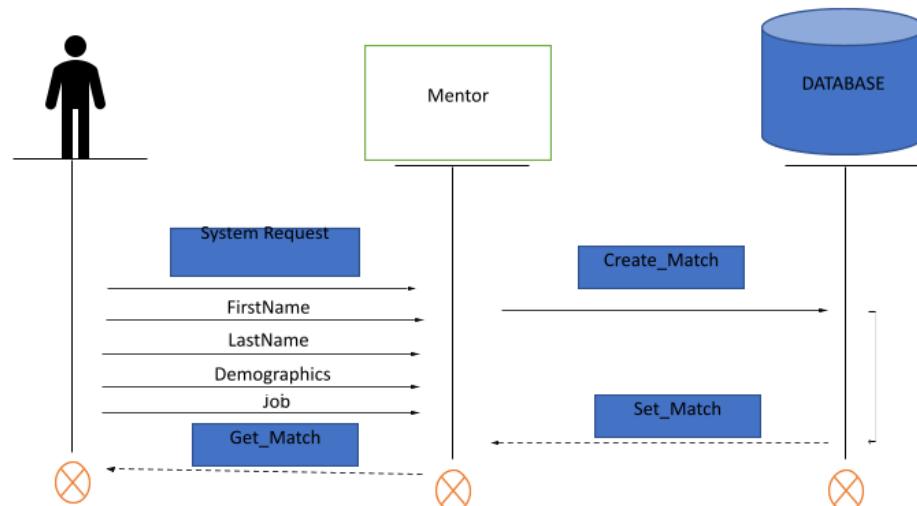
### Brief Description

This use-case will allow an administrator to create a match if a student is unable to find a match.

## 2. Flow of Events

### Basic Flow

- An administrator will be alerted if match not found
- An administrator will review student profile
- An administrator will review mentor profile
- An administrator will assess each profile
- An administrator will reach out to involved parties
- An administrator will make recommendations to candidate mentee
- Candidate mentee will reach out to recommended mentor
- Candidate mentee will communicate with system administrator
- Recommended mentor will communicate with system administrator
- System administrator will create a match



## 3. Special Requirements

Candidate mentee was unable to be matched to a mentor

## 4. Pre-conditions

- Student must have their profile ready.
- Student had to make an attempt to find a match.
- Student was unable to find a match.
- Student must request the system administrator to create a match.

## 5. Post-conditions

Student will be matched with a mentor.

# Use Case Specification: User Log In

## 1. Use-Case Name: Log In: ID 15

### Brief Description

- The system will allow approved **users** to log-in.

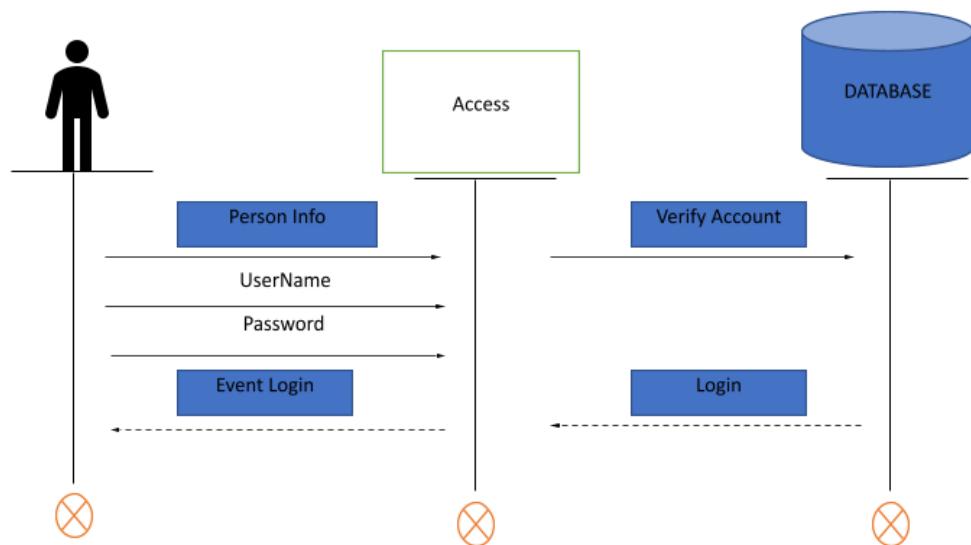
## 2. Flow of Events

### Basic Flow

- User clicks log-in button
- System processes request
- System verifies user has account
- User enters System on their account

### < User Log-In Denied >

- User clicks log-in button
- System processes request
- System verifies User does not have an account
- System redirects User back to log in



## 3. Special Requirements

### User has account

- User must have an account created and verified by the system administrator

## 4. Pre-conditions

- User has account
- User enters login credentials
- User clicks log-in

## 5. Post-conditions

- User log-in to system on their account

# Use Case Specification: User Log Out

## 1. Use-Case Name: User Log Out: ID 16

### Brief Description

- The system will allow approved users to log-out

## 2. Flow of Events

### Basic Flow

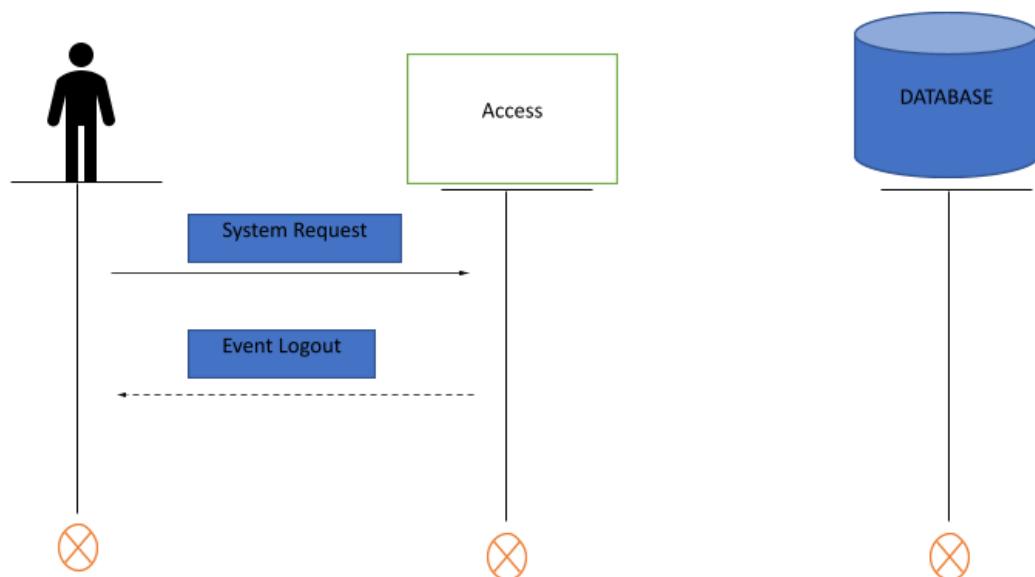
- Users selects log out button
- System process request
- System completes request
- User redirected to home page out of account

## 3. Pre-conditions

- User must be logged in

## 4. Post-conditions

- User logged out of account



# Use Case Specification: Create Announcement

## 57. Use-Case Name: Create Announcement: ID 17

### Brief Description

- The system will allow the **system administrator** to create announcements.

## 58. Flow of Events

### Basic Flow

- The system administrator creates a message
- The system administrator will send out the message using the system function
- The system will process request
- The system completes the request
- The system will display the message

## 59. Special Requirements

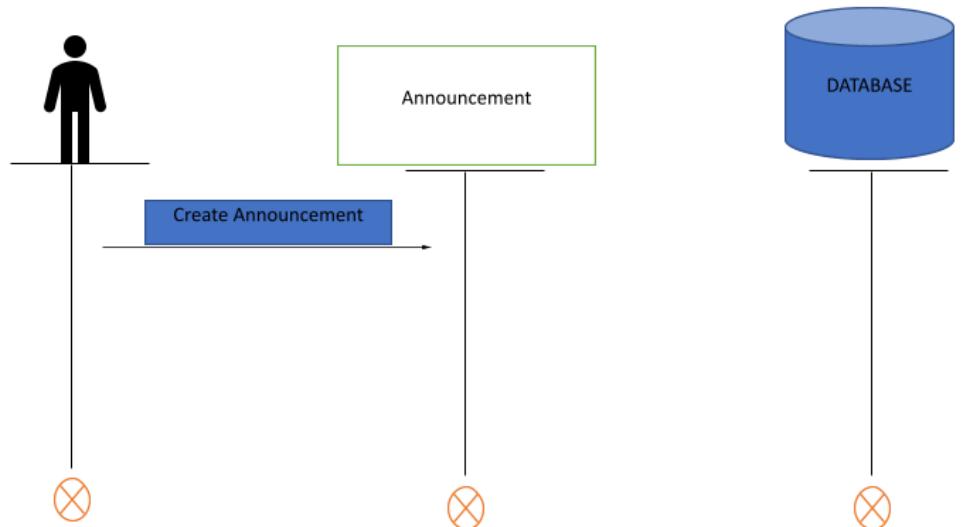
- A need for the system administrator to create a message

## 60. Pre-conditions

- A function to create and send messages
- The system administrator need of sending a message

## 61. Post-conditions

- Message will be displayed in the system to all users



# Use Case Specification: Edit Announcement

## 62. Use-Case Name: Edit Announcement: ID 18

### Brief Description

- The system will allow the **system administrator** to edit their announcements.

## 63. Flow of Events

### Basic Flow

- The system administrator presses a button to edit message
- The system administrator will write a new message
- The system administrator will send out the new message using the system function
- The system will process request
- The system completes the request
- The system will display the new message

## 64. Special Requirements

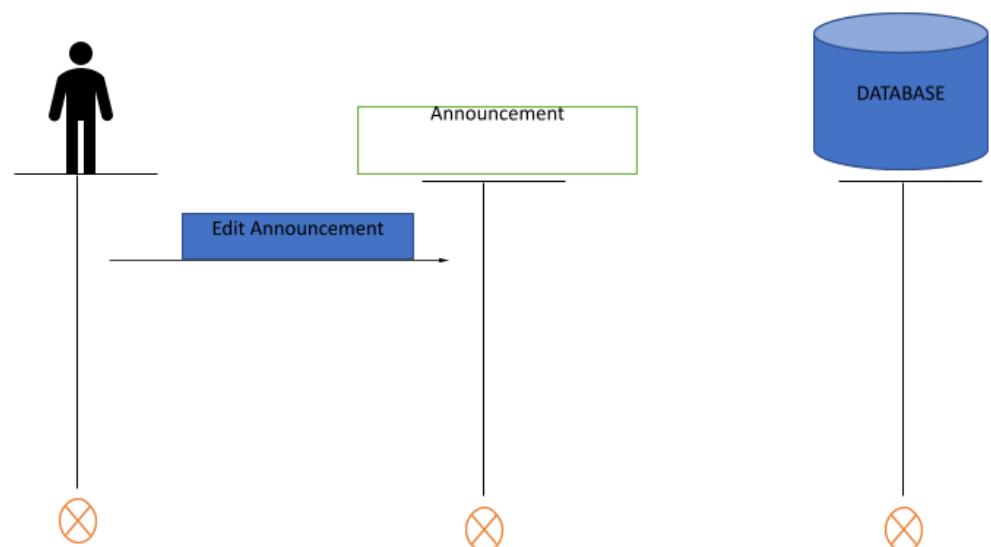
- The system administrator will need to have reason to edit a message

## 65. Pre-conditions

- A message must already exist

## 66. Post-conditions

- A new message will appear in place of old message



# Use Case Specification: Delete Announcement

## 67. Use-Case Name: Delete Announcement: ID 19

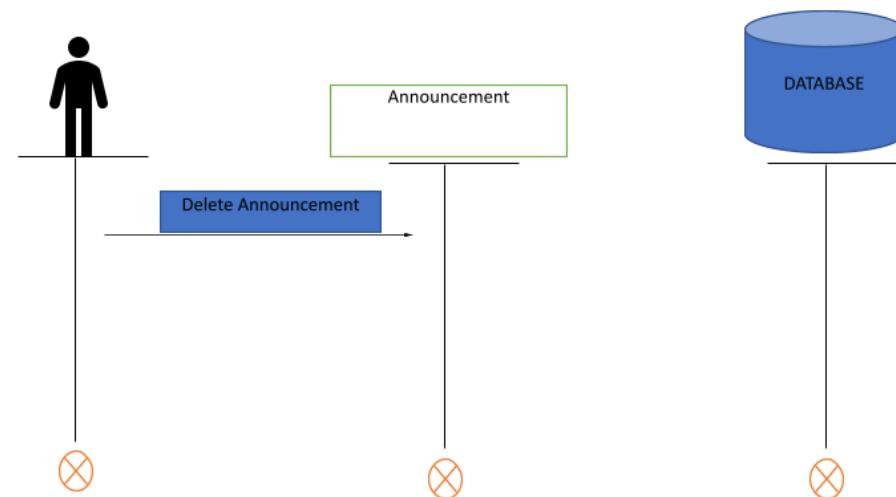
### Brief Description

- The system will give the **system administrator** ability to delete their announcements.

## 68. Flow of Events

### Basic Flow

- The system administrator presses a button to delete a message
- The system will process request
- The system completes the request
- The system will remove the message from display



### Alternative Flows

#### Confirm Delete

- The system administrator presses a button to delete a message
- The system will prompt the system administrator to confirm the action
- The system will process request
- The system completes the request
- The system will remove the message from display

#### Exit Delete

- The system will process request
- The system completes the request
- The system will not remove the message from display

## 69. Special Requirements

- The system administrator has reason to remove a message

**70. Pre-conditions**

- A message must already exist

**71. Post-conditions**

- A message will be deleted
- A message will not be deleted

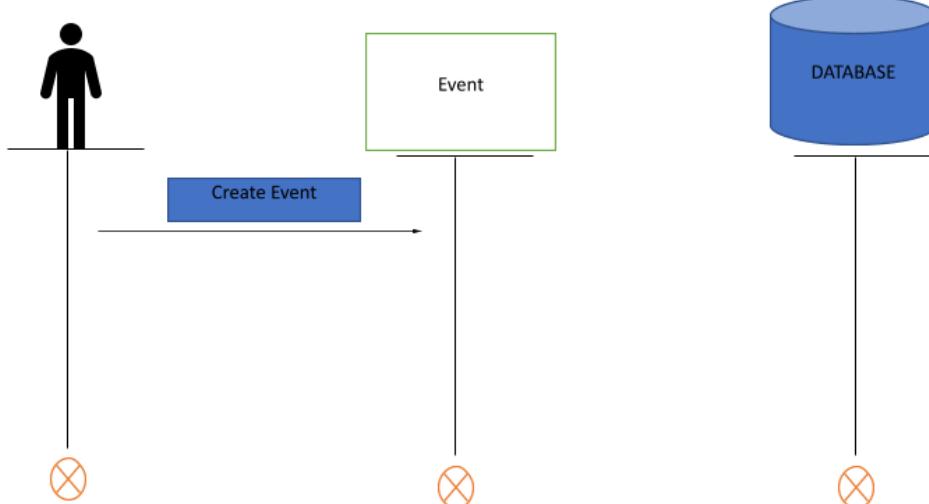
## Use Case Specification: Create Events

**72. Use-Case Name: Create Events: ID 20****Brief Description**

- The system will allow the **system administrator** to create events that will be displayed on the events page.

**73. Flow of Events****Basic Flow**

- The system administrator receives request for event
- The system administrator will review the request for event
- The system administrator will use system function to create event
- The system processes the request
- The system completes the request
- The system will display the event

**Alternative Flows****System Administrator Denies Event**

- The system administrator receives request for event
- The system administrator will review the request for event

- The system administrator will not create event

**74. Pre-conditions**

- An event must exist
- The system administrator must be notified of event

**75. Post-conditions**

- An event is created
- An event is not created

# Use Case Specification: Edit Event

## 76. Use-Case Name: Edit Event: ID 21

### Brief Description

- The system will allow the **system administrator** to edit events on the events page.

## 77. Flow of Events

### Basic Flow

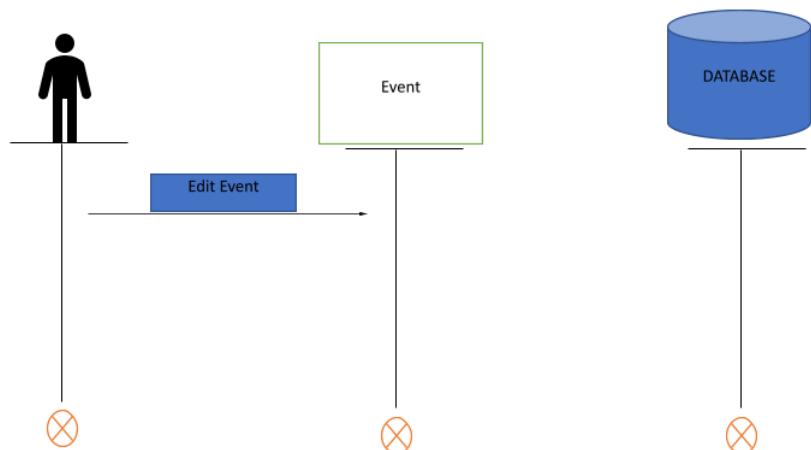
- The system administrator presses a button to edit event
- The system administrator will modify the current event
- The system administrator will send out the modified event using the system function
- The system will process request
- The system completes the request
- The system will display the modified event

## 78. Pre-conditions

- An event must exist

## 79. Post-conditions

- An event will be modified



# Use Case Specification: Delete Events

## 1. Use-Case Name: Delete Events: ID 22

### 1.1 Brief Description

The system administrator will be able to delete events that were displayed on the events page if needed.

## 2. Flow of Events

### 2.1 Basic Flow

- The system administrator presses a button to edit an event
- The system administrator presses a button to delete an event.
- The system will process the request
- The system completes the request
- The system will remove the event from display.

### 2.2 Alternative Flows

#### 2.2.1 Confirm Delete

- The system administrator presses a button to edit an event
- The system administrator presses a button to delete an event
- The system will prompt the system administrator to confirm the action
- The system administrator presses a button to confirm the action
- The system will process the request
- The system completes the request
- The system will remove the event from display

#### 2.2.1.1 Cancel Delete

- The system administrator presses a button to edit an event
- The system administrator presses a button to delete an event
- The system will prompt the system administrator to confirm the action
- The system administrator presses a button to cancel the action



- The system will process the request
- The system completes the request
- The system will not remove the event from display

**3. Special Requirements**

- The system administrator has reason to remove an event

**4. Pre-conditions**

- System administrator must be logged in
- An event must already exist

**5. Post-conditions**

- An event will be deleted
- An event will not be delete

# Use Case Specification: Create Job Post

## 6. Use-Case Name: Create Job Post: ID 23

### 6.1 Brief Description

The system will allow mentors to post job opportunities to the job board.

## 7. Flow of Events

### 7.1 Basic Flow

- Mentor will press a button to add a job post
- Mentor will enter job title
- Mentor will enter job description
- Mentor will enter job requirements
- The system will process the request
- The system completes the request
- The system will create job post

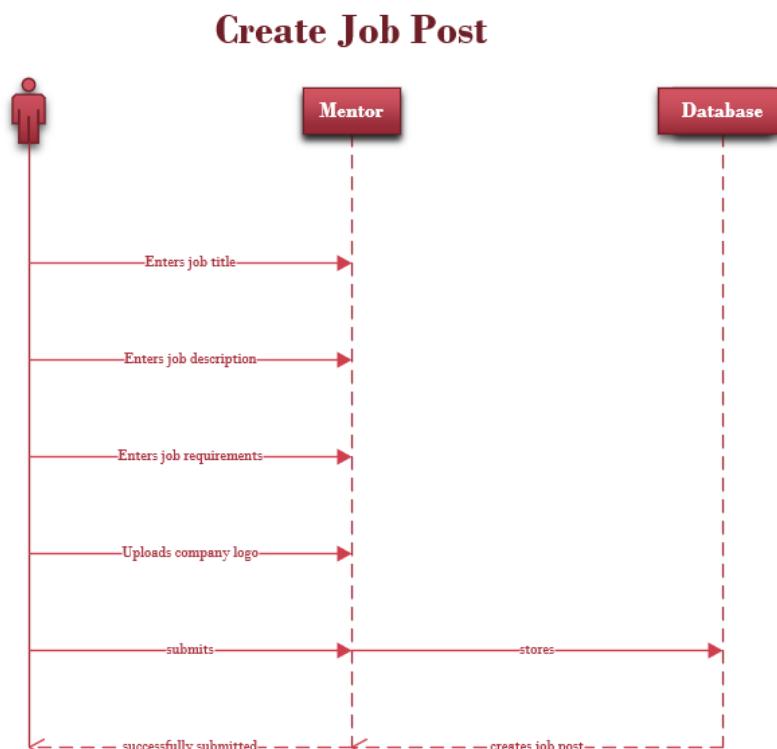
### 7.2 Alternative Flows

#### 7.2.1 Confirm Create Job Post

- Mentor will press a button to add a job post
- Mentor will enter job title
- Mentor will enter job description
- Mentor will enter job requirements
- The system will prompt mentor to confirm that they want to create a job posting
- The mentor will click confirm
- The system will process the request
- The system completes the request
- The system will create job post

#### 7.2.1.1 Cancel Create Job Post

- Mentor will press a button to add a job post
- Mentor will enter job title
- Mentor will enter job description
- Mentor will enter job requirements
- The system will prompt mentor to confirm that they want to create a job posting



- The mentor will click cancel
- The system will process the request
- The system completes the request
- The system will not create job post

**8. Pre-conditions**

- Mentor must be logged in to the system
- Mentor must be an alum from the University of Louisville

**9. Post-conditions**

- A job posting will be created
- A job posting will not be created

# Use Case Specification: Edit Job Post

## 10. Use-Case Name: Edit Job Post: ID 24

### 10.1 Brief Description

The system will allow mentors to edit their job post on the job board.

## 11. Flow of Events

### 11.1 Basic Flow

- Mentor will press a button to edit job post
- Mentor may edit job title
- Mentor may edit job description
- Mentor may edit job requirements
- The system will process the request
- The system completes the request
- The system will save changes to job board

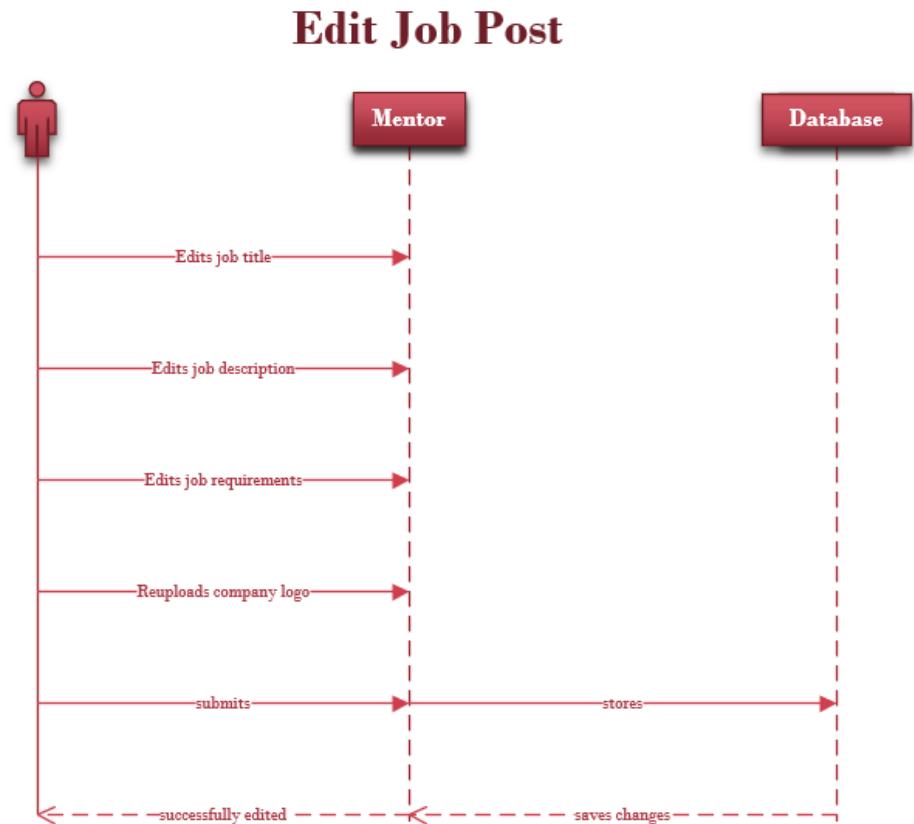
### 11.2 Alternative Flows

#### 11.2.1 Confirm Edit Job Post

- Mentor will press a button to edit job post
- Mentor may edit job title
- Mentor may edit job description
- Mentor may edit job requirements
- The system will prompt mentor to confirm that they want to save changes made to job posting
- The mentor will click confirm
- The system will process the request
- The system completes the request
- The system will save changes to job board

#### 11.2.1.1 Cancel Edit Job Post

- Mentor will press a button to edit job post
- Mentor may edit job title
- Mentor may edit job description



- Mentor may edit job requirements
- The system will prompt mentor to confirm that they want to save changes made to job posting
- The mentor will click cancel
- The system will process the request
- The system completes the request
- The system will not save changes to job board

**12. Pre-conditions**

- Mentor must be logged in to the system
- Mentor must be an alum from the University of Louisville
- Job post must already exist

**13. Post-conditions**

- Changes made will be saved to job posting
- Changes made will not be saved to job posting

## Use Case Specification: Delete Job Post

### 14. Use-Case Name: Delete Job Post: ID 25

#### 14.1 Brief Description

The system will allow mentors to delete their job post on the job board.

### 15. Flow of Events

#### 15.1 Basic Flow

- Mentor will press a button to edit job post
- Mentor will click delete job post
- The system will process the request
- The system completes the request
- The system will delete job post

#### 15.2 Alternative Flows

##### 15.2.1 Confirm Delete Job Post

- Mentor will press a button to edit job post
- Mentor will click delete job post
- The system will prompt mentor to confirm that they want to delete job post
- Mentor will click confirm
- The system will process the request
- The system completes the request
- The system will delete job post

##### 15.2.1.1 Cancel Delete Job Post

- Mentor will press a button to edit job post
- Mentor will click delete job post
- The system will prompt mentor to confirm that they want to delete job post
- Mentor will click cancel
- The system will process the request
- The system completes the request
- The system will not delete job post



**16. Pre-conditions**

- Mentor must be logged in to the system
- Mentor must be an alum from the University of Louisville
- Job post must already exist

**17. Post-conditions**

- Job post will be deleted
- Job post will not be deleted

## Use Case Specification: Create Discussion

### 18. Use-Case Name: Create Discussion: ID 26

#### 18.1 Brief Description

The system will allow students to post to the discussion board

#### 19. Flow of Events

##### 19.1 Basic Flow

- Student will press a button to create discussion post
- Student will enter discussion title
- Student will enter text in text box
- The system will process the request
- The system completes the request
- The system will create discussion post

##### 19.2 Alternative Flows

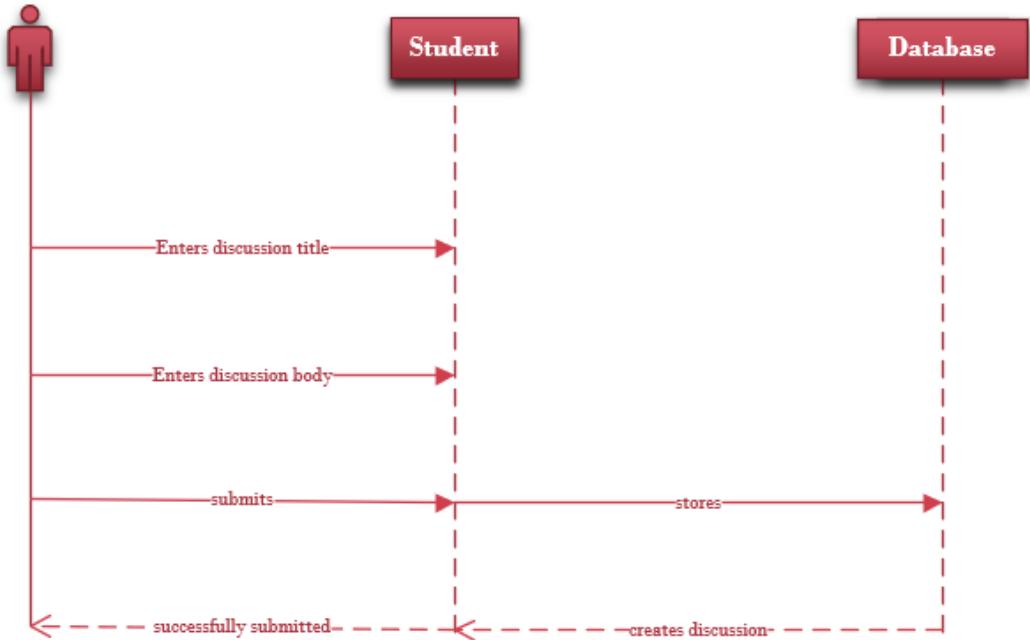
###### 19.2.1 Confirm Create Discussion

- Student will press a button to create discussion post
- Student will enter discussion title
- Student will enter text in text box
- The system will prompt student to confirm that they want to create discussion post
- Student will click confirm
- The system will process the request
- The system completes the request
- The system will create discussion post

###### 19.2.1.1 Cancel Create Discussion

- Student will press a button to create discussion post
- Student will enter discussion title
- Student will enter text in text box
- The system will prompt student to confirm that they want to create discussion post

## Create Discussion



- Student will click cancel
- The system will process the request
- The system completes the request
- The system will not create discussion post

19.2.1.2 *Save Entry as Draft*

- Student will press a button to create discussion post
- Student will enter discussion title
- Student will enter text in text box
- The system will prompt student to confirm that they want to create discussion post
- Student will click save as draft
- The system will prompt student to confirm if they want to save entry as draft
- Student will click confirm
- The system will process the request
- The system completes the request
- The system will save discussion entry as draft

**20. Pre-conditions**

- Student must be logged into the system

**21. Post-conditions**

- Discussion post will be created
- Discussion post will not be created
- Discussion post will be saved as a draft

## Use Case Specification: Edit Discussion

### 22. Use-Case Name: Edit Discussion: ID 27

#### 22.1 Brief Description

The system will allow students to edit their post on the discussion board

### 23. Flow of Events

#### 23.1 Basic Flow

- Student will press a button to edit discussion post
- Student may edit discussion title
- Student may edit text in text box
- The system will process the request
- The system completes the request
- The system will edit discussion post

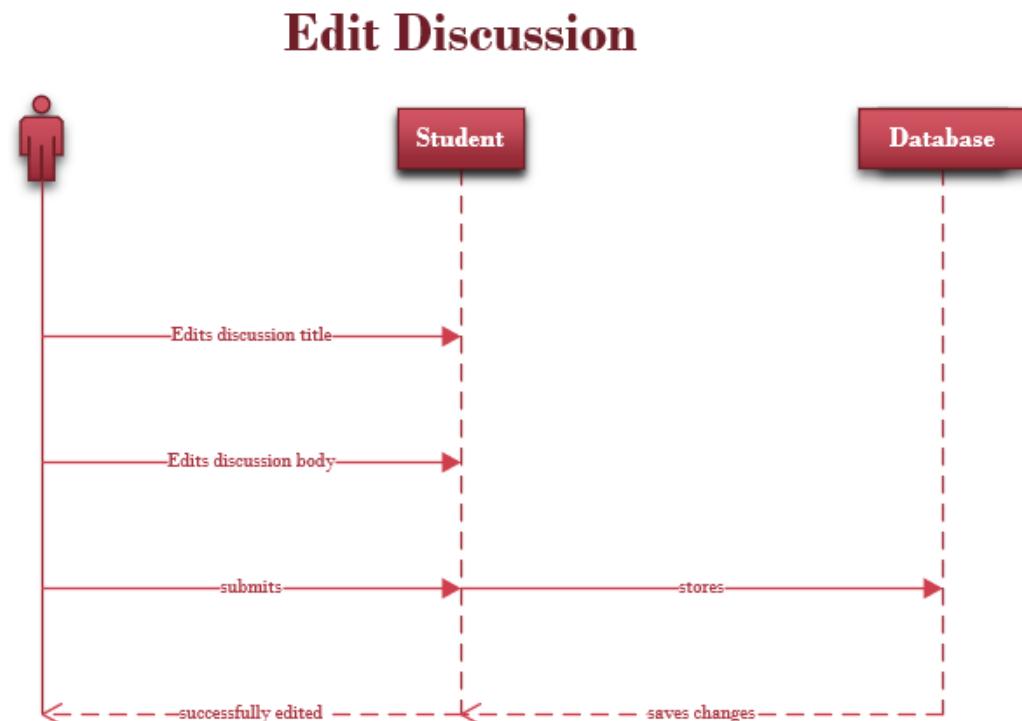
#### 23.2 Alternative Flows

##### 23.2.1 Confirm Edit Discussion

- Student will press a button to edit discussion post
- Student may edit discussion title
- Student may edit text in text box
- The system will prompt student to confirm that they want to edit discussion post
- Student will click confirm
- The system will process the request
- The system completes the request
- The system will edit discussion post

##### 23.2.1.1 Cancel Edit Discussion

- Student will press a button to edit discussion post
- Student may edit discussion title
- Student may edit text in text box
- The system will prompt student to confirm that they want to edit discussion post



- Student will click cancel
- The system will process the request
- The system completes the request
- The system will not edit discussion post

23.2.1.2 *Save Entry as Draft*

- Student will press a button to edit discussion post
- Student may edit discussion title
- Student may edit text in text box
- The system will prompt student to confirm that they want to edit discussion post
- Student will click save as draft
- The system will prompt student to confirm if they want to save the changes made as a draft
- Student will click confirm
- The system will process the request
- The system completes the request

**24. Pre-conditions**

- Student must be logged into the system
- Discussion post must exist

**25. Post-conditions**

- Discussion post will be edited
- Discussion post will not be edited
- Changes made to discussion post will be saved as a draft

# Use Case Specification: Delete Discussion

## 26. Use-Case Name: Delete Discussion: ID 28

### 26.1 Brief Description

The system will allow students to delete their post on the discussion board

## 27. Flow of Events

### 27.1 Basic Flow

- Student will press a button to delete discussion post
- The system will process the request
- The system completes the request
- The system will delete discussion post

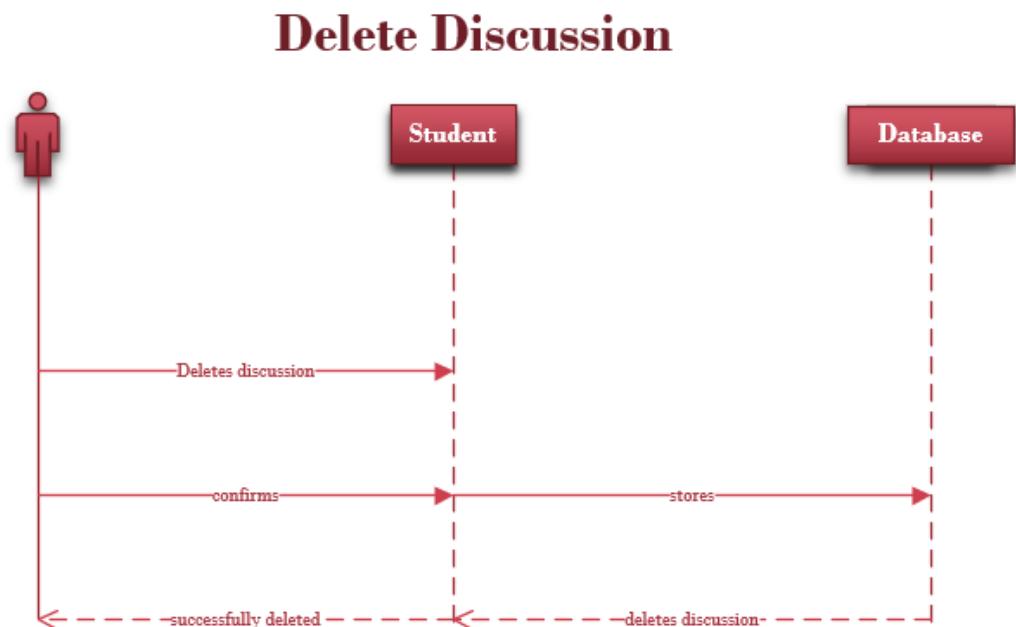
### 27.2 Alternative Flows

#### 27.2.1 Confirm Delete Discussion

- Student will press a button to delete discussion post
- The system will prompt student to confirm that they want to delete discussion post
- Student will click confirm
- The system will process the request
- The system completes the request
- The system will delete discussion post

#### 27.2.1.1 Cancel Delete Discussion

- Student will press a button to delete discussion post
- The system will prompt student to confirm that they want to delete discussion post
- Student will click cancel
- The system will process the request
- The system completes the request
- The system will not delete discussion post



## 28. Pre-conditions

- Student must be logged into the system
  - Discussion post must exist

## 29. Post-conditions

- Discussion post will be deleted
  - Discussion post will not be deleted

## **Use Case Specification: Comment on Discussion**

**Use-Case: Contact Mentor ID: 29**

## Brief Description

This use-case will allow authorized users to comment on discussion board posts.

## Flow of Events

## Basic Flow

- User will log in to the system
  - User will click on the discussion board tab
  - User will read through discussion board posts
  - User will find a post to comment

## Alternative Flow

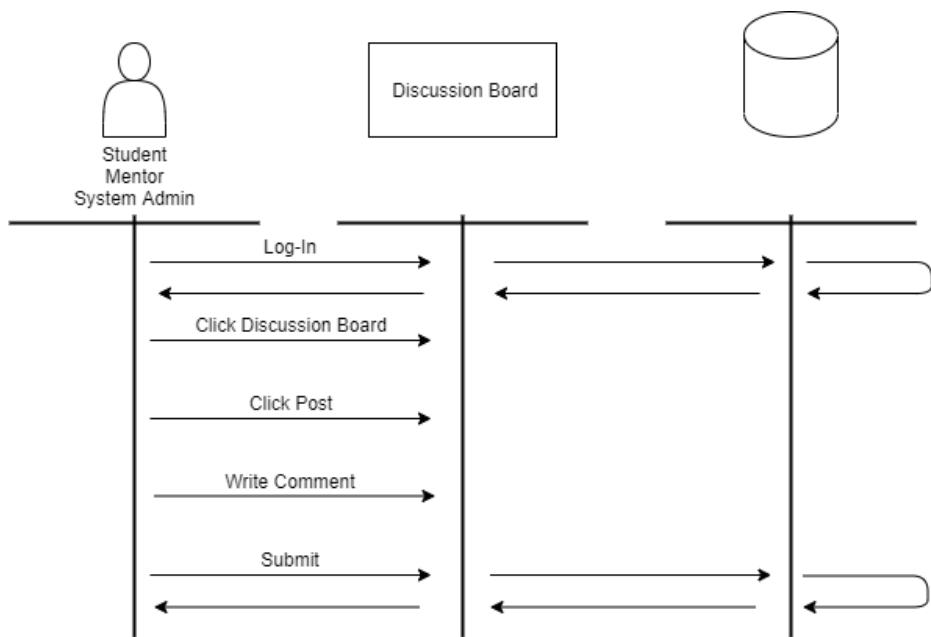
- User does not post a comment on a discussion board post

## Pre-conditions

- Users must be logged in to the system in order to access the discussion board.
  - Users must feel the need to comment on a discussion board post.

## Post-conditions

- A new comment is made on the discussion board post.



# Use Case Specification: Edit Comment on Discussion

## Use-Case: Contact Mentor ID: 30

### Brief Description

This use-case will allow authorized users to edit their comment(s) on discussion board posts.

### Flow of Events

#### Basic Flow

- User will log in to the system
- User will go to discussion board
- User will locate their comment on discussion board post
- User will click on pencil icon on their comment
- User will edit their comment on the post
- User will confirm edit
- User will submit edited comment

#### Alternative Flow

- User does not edit their comment

### Pre-conditions

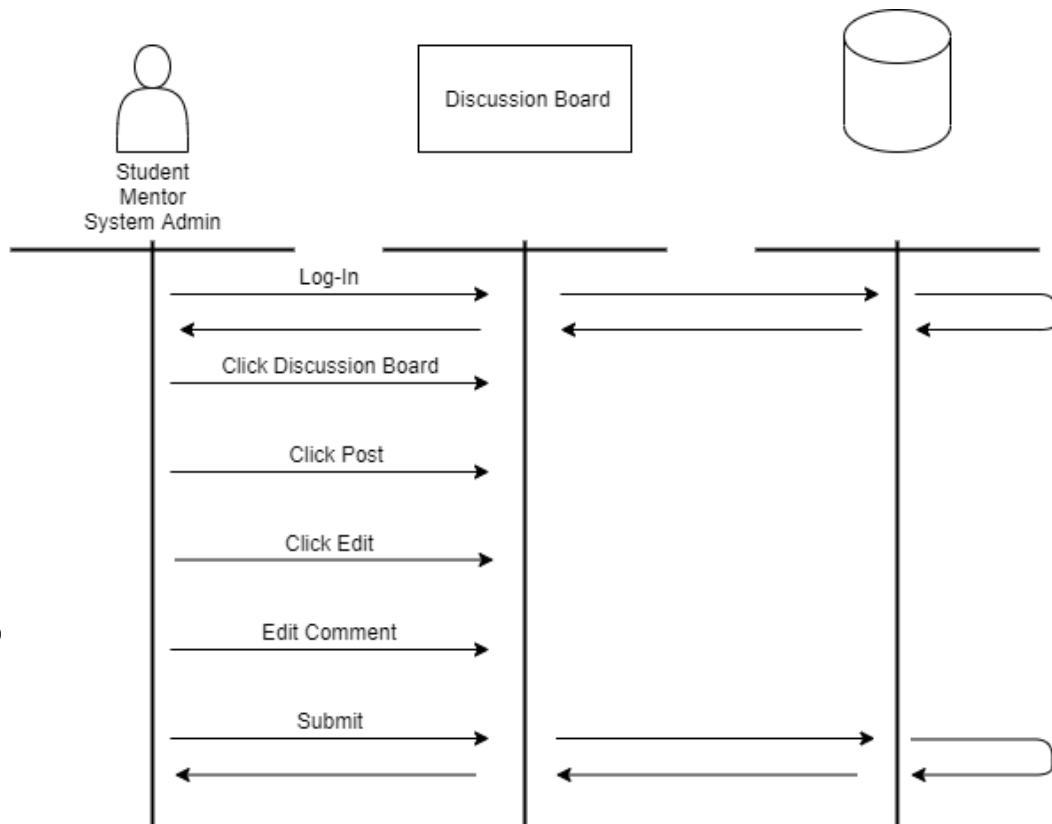
User must be logged in to the system in order to edit their comment on the discussion board.

User must have already posted a comment.

User must feel the need to edit their comment.

### Post-conditions

A user's comment has been edited.



# Use Case Specification: Delete Comment on Discussion

## Use-Case: Contact Mentor ID: 31

### Brief Description

This use-case will allow authorized users to delete their comment(s) on discussion board posts.

### Flow of Events

#### Basic Flow

- User will log in to the system
- User will go to discussion board
- User will locate their comment on discussion board post
- User will click on trash bin icon
- User will confirm deletion of comment

#### Alternative Flow

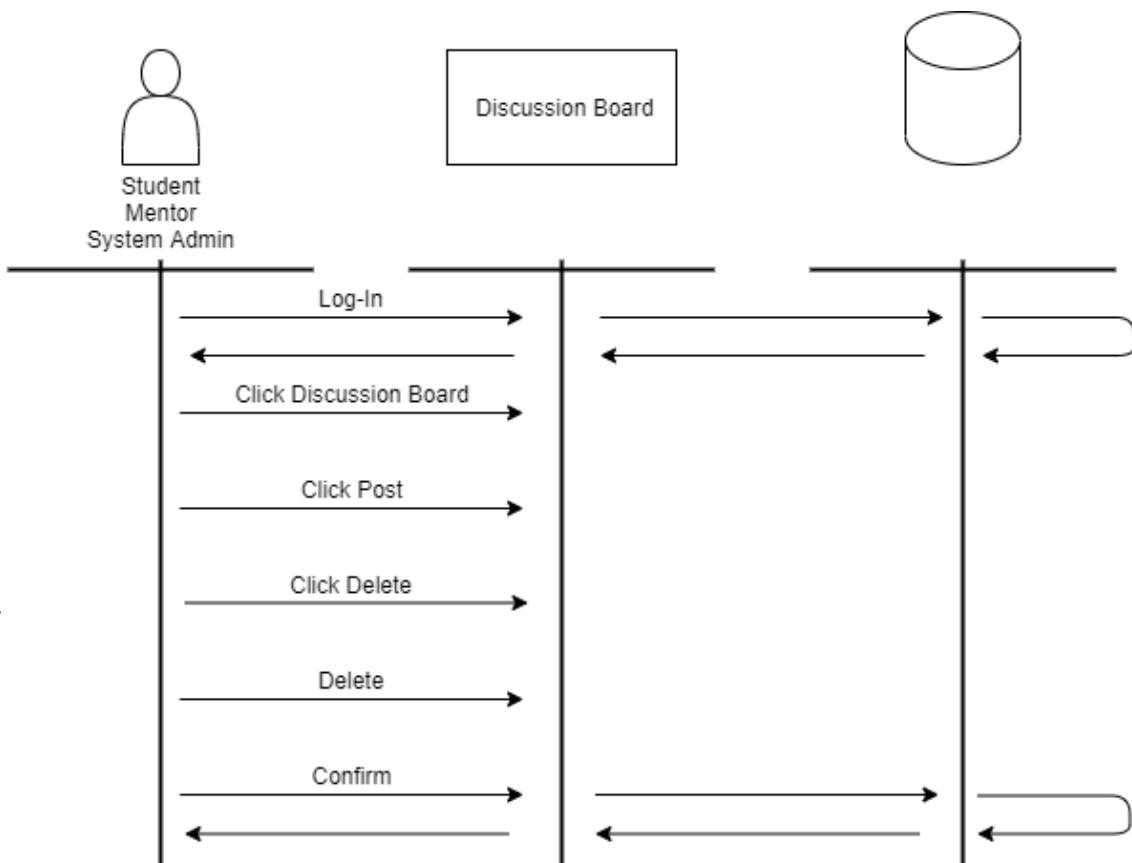
- User does not delete their comment

### Pre-conditions

- Users must be logged in to the system in order to delete their comment on the discussion board.
- Users must have already posted a comment.
- Users must feel the need to delete their comment.

### Post-conditions

- A user's comment is deleted.



# **Use Case Specification: Upload Photo(s)**

## **Use-Case Name: Upload Photo: I.D. 32**

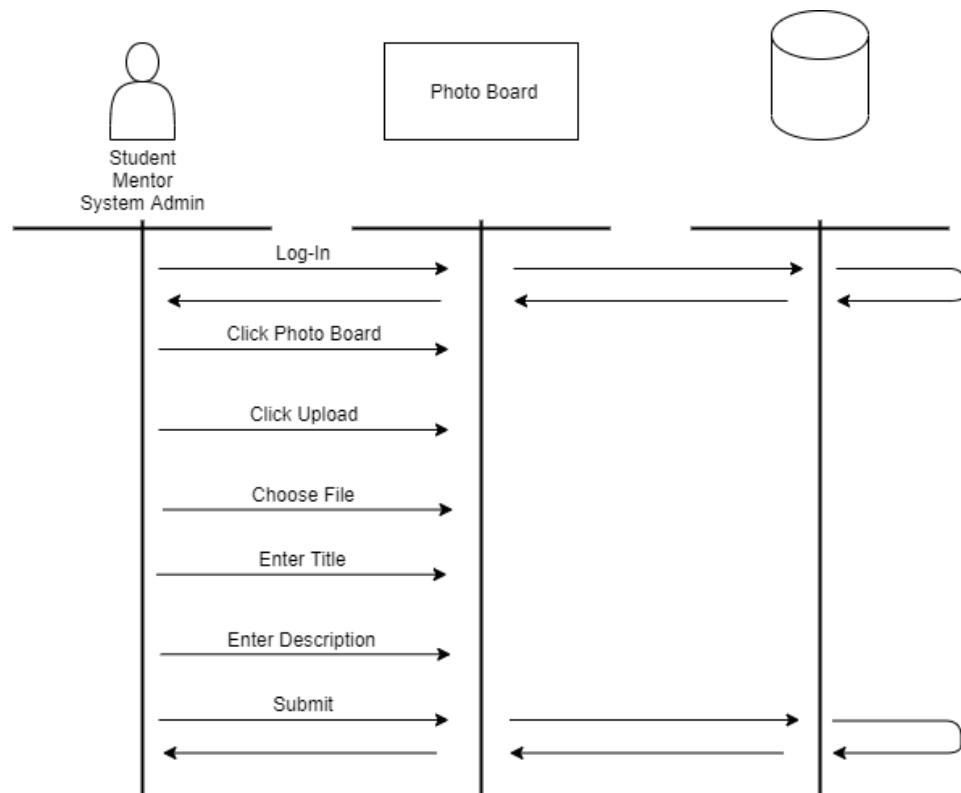
## Brief Description

The system will allow **users** to post photos on the homepage and the photo board.

## Flow of Events

## Basic Flow

- The user presses a button to upload a photo(s)
  - The user selects the photo(s)
  - The user will enter a photo title
  - The user will enter a photo description
  - The user selects submit
  - The system processes the request
  - The system completes the request
  - The photo(s) will be displayed on the homepage
  - The photo(s) will be displayed on the photo board



### Alternative Flow

### *Upload Fails insufficient storage*

- The user presses a button to upload photo(s)
  - The user selects the photo(s)
  - The user selects submit
  - The system processes the request
  - The system denies the request
  - The photo(s) is not displayed on the homepage
  - The photo(s) is not displayed on the photo board

### **Special Requirements**

- User must have a photo(s) to upload

## Pre-Conditions

- A function to select from a pool of photos
  - A function to upload photos to the site

## Post-Condition

- The photo(s) will be uploaded to the webpage

# **Use Case Specification: Edit Photo(s)**

## **Use-Case Name: Edit Photo(s): I.D. 33**

## **Brief Description**

The system will allow **users** to edit photos posted on the homepage and the photo board.

## Flow of Events

## Basic Flow

- The user will press edit photo button
  - The user will modify the current photo
  - The system will process the request
  - The system will complete the request
  - The modified photo(s) will be displayed on the homepage

## **Special Requirements**

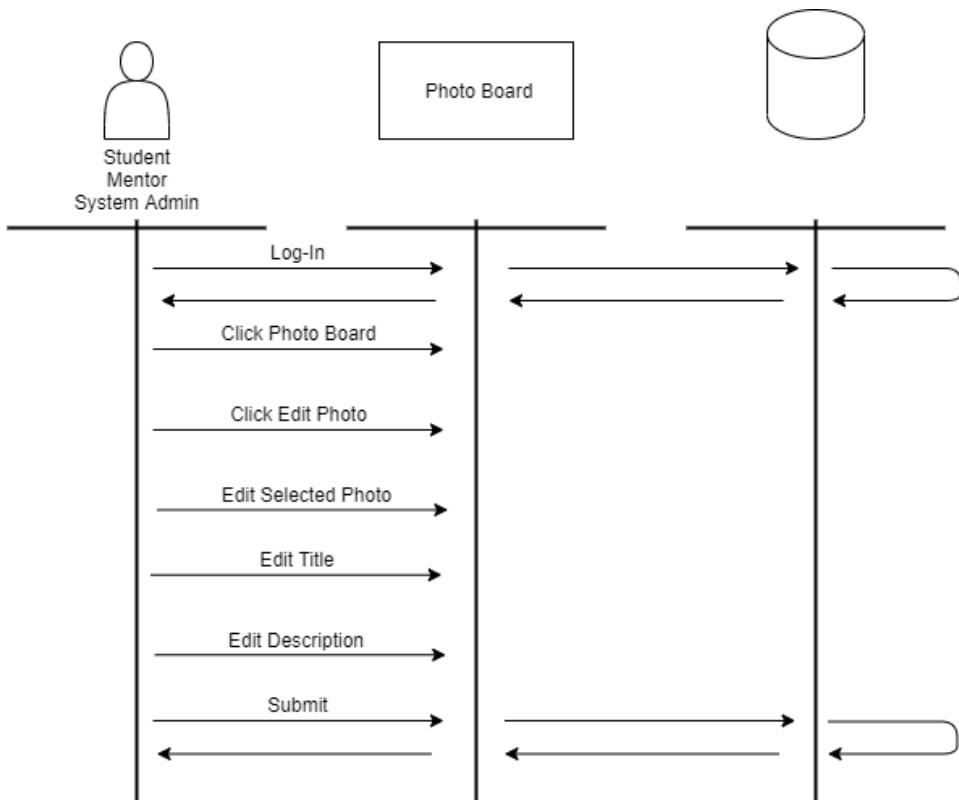
- The user needs a reason to edit the photo(s)

## Pre-Conditions

- Photo(s) must already be uploaded

## Post-Conditions

- The modified photo(s) will replace the original



# Use Case Specification: Delete Photo(s)

Use-Case Name: Delete Photo(s): I.D. 34

## Brief Description

The system will allow **users** to delete photos posted on the homepage

## Flow of Events

### Basic Flow

- The user presses a button to delete photo(s)
- The system will process the request
- The system will complete the request
- The photo(s) will be removed from the homepage

## Special Requirements

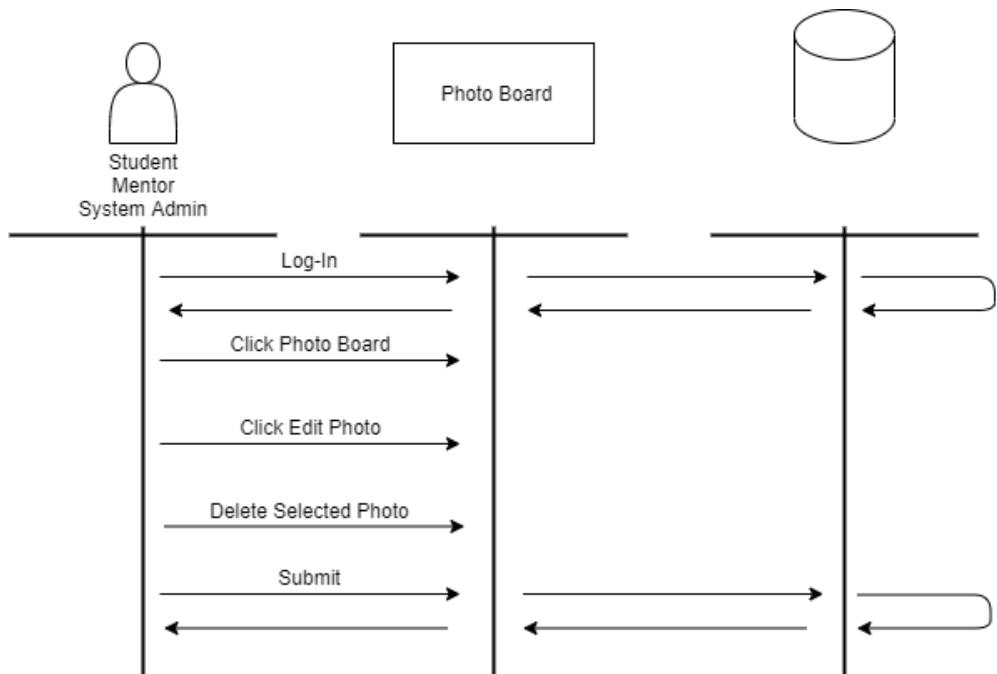
- The user has a need to delete the photo(s)

## Pre-Conditions

- A photo(s) must already exist

## Post-Conditions

- The photo(s) will no longer be displayed



# Use Case Specification: Donate

Use-Case Name: Donate: I.D. 35

## Brief Description

The system will allow **users** and **visitors** to donate to the program from the donations page.

## Flow of Events

### Basic Flow

- The user presses a button to donate
- The user enters the dollar amount they'd like to donate
- The user enters their payment information
- The system processes the request
- The system completes the request
- The donation is received

### Special requirements

- The user must want to donate

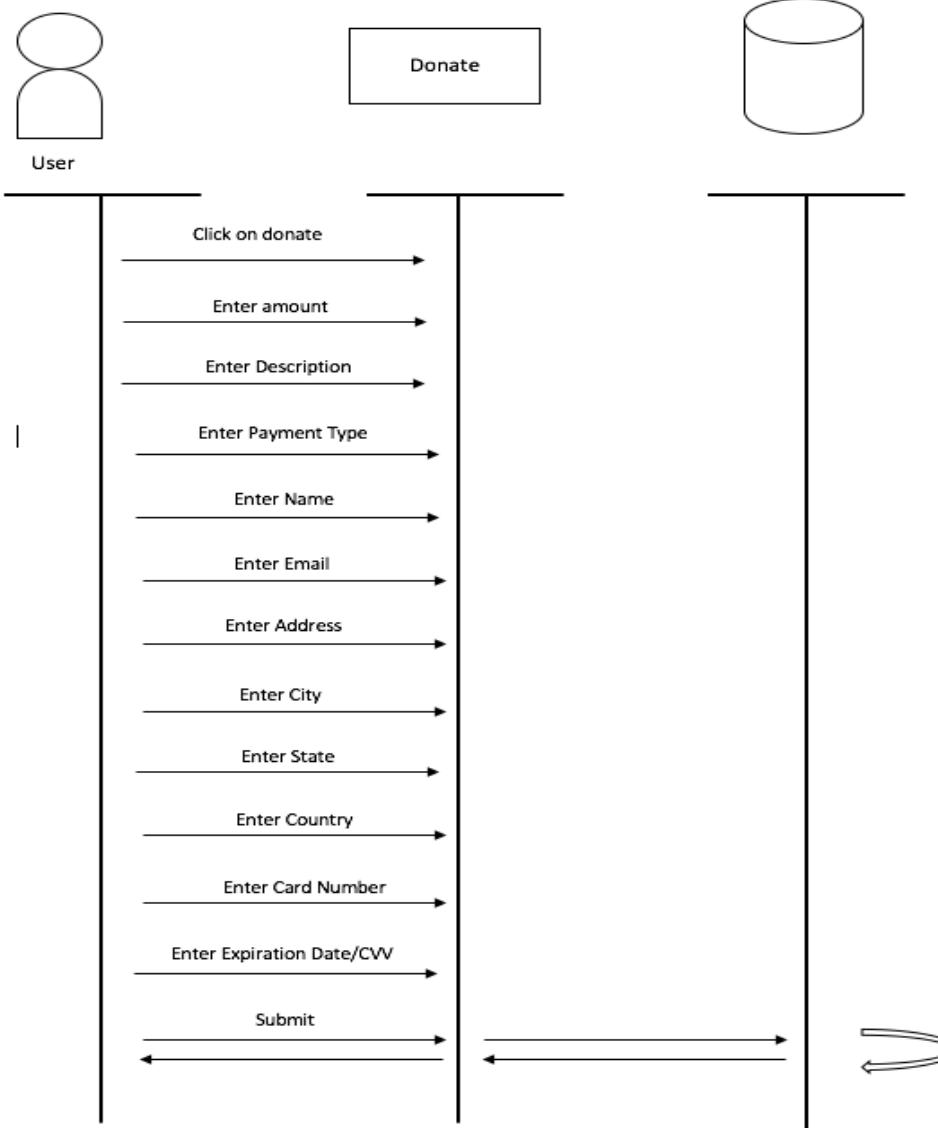
### Pre-Conditions

- A function to accept donations must exist
- A function to accept the users payment information must exist

### Post-Conditions

- The money will be received

Sequence Diagram for Use-Case ID 33: Donate



# Use Case Specification: Send Report

## 1. Use-Case Name: Send Report: I.D. 36

### 1.1 Brief Description

The system will allow **users** to anonymously report any problems to the **system administrator**.

### 2. Flow of Events

#### 2.1 Basic Flow

- The user presses a button to send a report
- The user types the problem in a text box
- The user presses send
- The system processes the request
- The system completes the request
- The system administrator receives the report
- The system administrator will administer a solution to the problem

### 3. Special Requirements

#### 3.1 A problem must be encountered

- The user must have a problem to report to the administrator

### 4. Pre-conditions

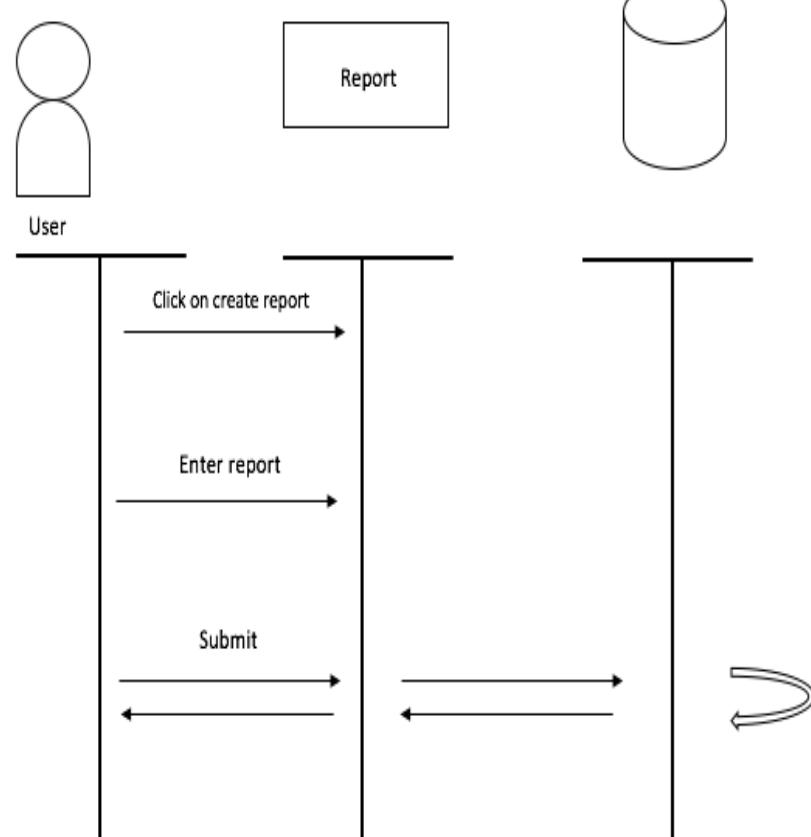
#### 4.1 User has account

- User must have an account if they wish to report a problem
- The system administrator must be able to receive the report

### 5. Post-conditions

- The system administrator fixed the problem

Sequence Diagram for Use-Case ID 34: Send Report



# Use Case Specification: Send Email Blast

## 1. Use-Case Name: Send Email Blast: I.D. 37

### 1.1 Brief Description

The system will allow the **system administrator** to send email blasts to users

### 2. Flow of Events

#### 2.1 Basic Flow

- The system administrator creates an email
- The system administrator sends out the email to all users
- The system will process the request
- The system will complete the request
- The email will be displayed to all users

#### 8.0 Special Requirements

- A need for the system administrator to create an email

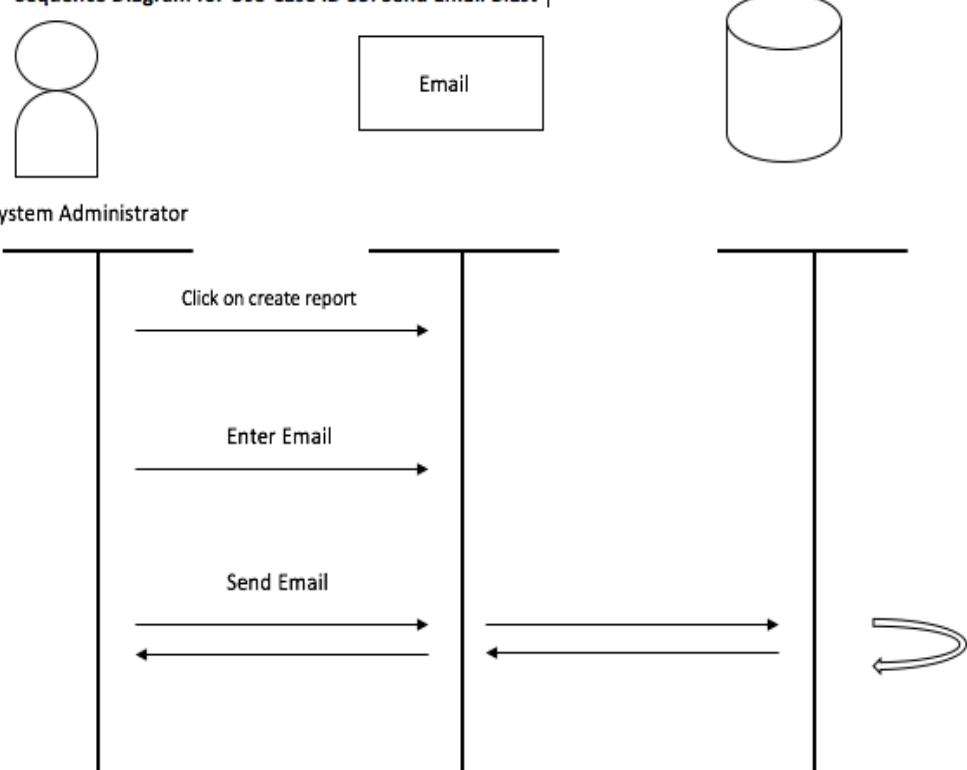
#### 9.0. Pre-Conditions

- A function to create emails
- A function to send out email to all users

#### 10. Post-Conditions

- Email displayed to all users in the system

Sequence Diagram for Use-Case ID 35: Send Email Blast |



## Use Case Specification: Create Survey

### 11. Use Case Name: Create Survey: I.D. 38

#### 11.1 Brief Description

The system will allow the system administrator to create surveys.

#### 12. Flow of Events

##### 12.1 Basic Flow

- The system administrator will create a survey
- The system administrator sends out the survey to all users
- The system processes the request
- The system completes the request
- The survey is available to all users

##### 13. Special Requirements

- A need for the system administrator to create a survey

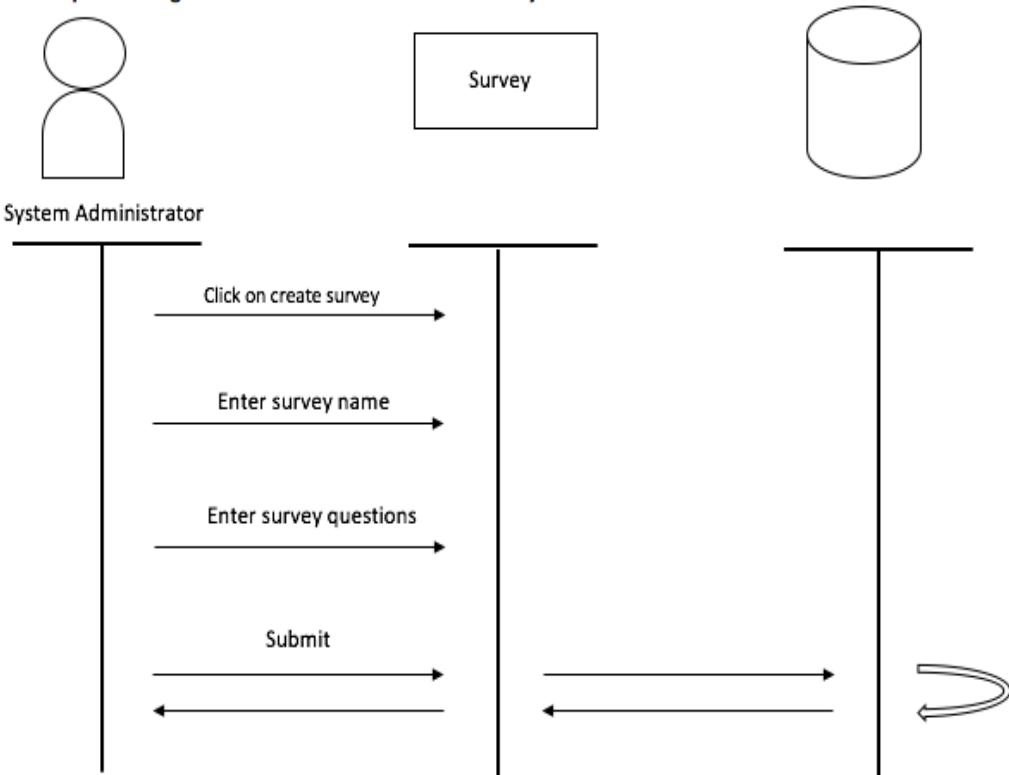
##### 14. Pre-Conditions

- A function to create surveys
- A function to send out the survey to all users

##### 15. Post-Condition

- Survey available to all users in the system

Sequence Diagram for Use-Case ID 36: Create Survey



## Use Case Specification: Edit Survey

### 16. Use Case Name: Edit Survey: I.D. 39

#### 16.1 Brief Description

The system will allow the system administrator to edit surveys and for intended users to fill out the survey.

#### 17. Flow of events

##### 17.1 Basic Flow

- The System administrator will press edit survey button
- The System administrator will modify the current survey
- The System administrator will send out the new version of the survey
- The System will process the request
- The system completes the request
- The system displays the new version of the survey

##### 18. Special Requirements

- System administrator needs a reason to edit the survey

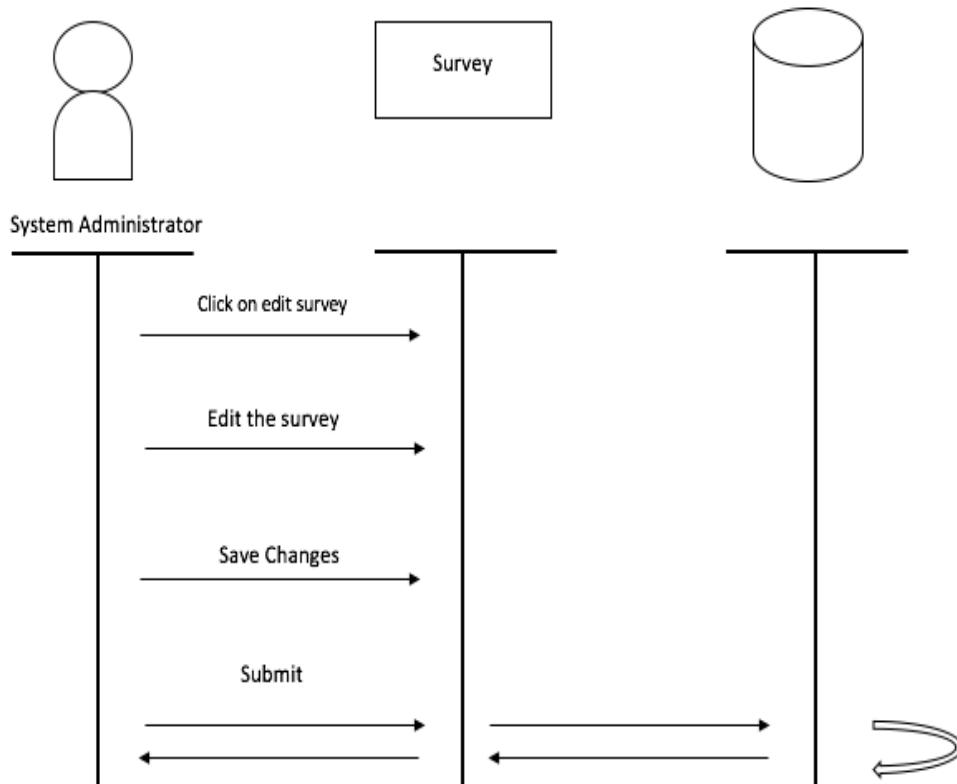
##### 19. Pre-Conditions

- A survey must already exist

##### 20. Post-Conditions

- A new survey will replace the old survey

Sequence Diagram for Use-Case ID 37: Edit Survey



# Use Case Specification: Delete Survey

## 21. Use Case Name: Delete Survey: I.D. 40

### 21.1. Brief description

The system will allow the **system administrator** to delete the survey when they deem necessary or it will expire when the time runs out.

### 22. Flow of Events

#### 22.1 Basic Flow

- The system administrator presses a button to delete survey
- The system processes the request
- The system completes the request
- The system removes the survey

#### 22. Special requirements

- The system administrator has a need to delete the survey

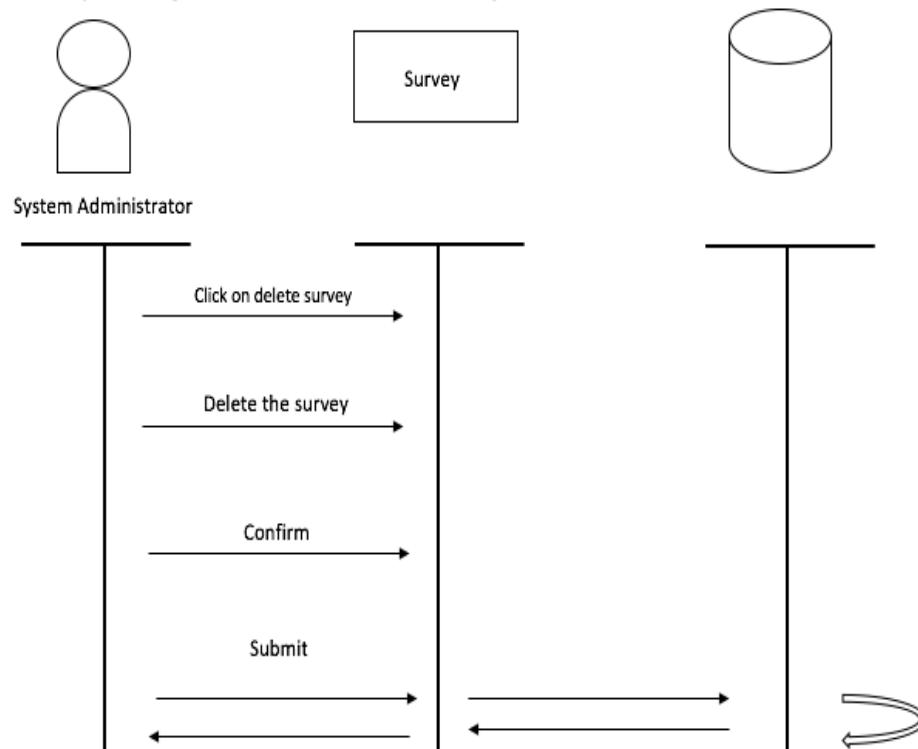
#### 23. Pre-Conditions

- A survey must already exist

#### 24. Post-Conditions

- The survey will no longer exist

Sequence Diagram for Use-Case ID 38: Delete Survey



# Use Case Specification: Create Backup

## 24. Use Case Name: Create Backup: I.D. 41

### 24.1 Brief Description

The system will need to allow the **system administrator** to create a backup of all the data or create a specified time frame to backup data automatically.

### 25. Flow of Events

#### 25.1 Basic Flow

- The system administrator will press a button to back-up the data
- The system will process the request
- The system completes the request
- The system backs up the data

#### 25.2 Alternative Flows

##### 25.2.1 Automatic back-up

- The system administrator presses a button to automatically back-up the data
- The system administrator will press a button selected a time frame for the data to backup
- The system will process the request
- The system completes the request
- The system backs up the data during the allotted time

##### 25.2.1.1. Insufficient storage

- The system administrator presses a button to back-up the data
- The system processes the request
- The system denies the request
- An error message is displayed
- The data is not backed up

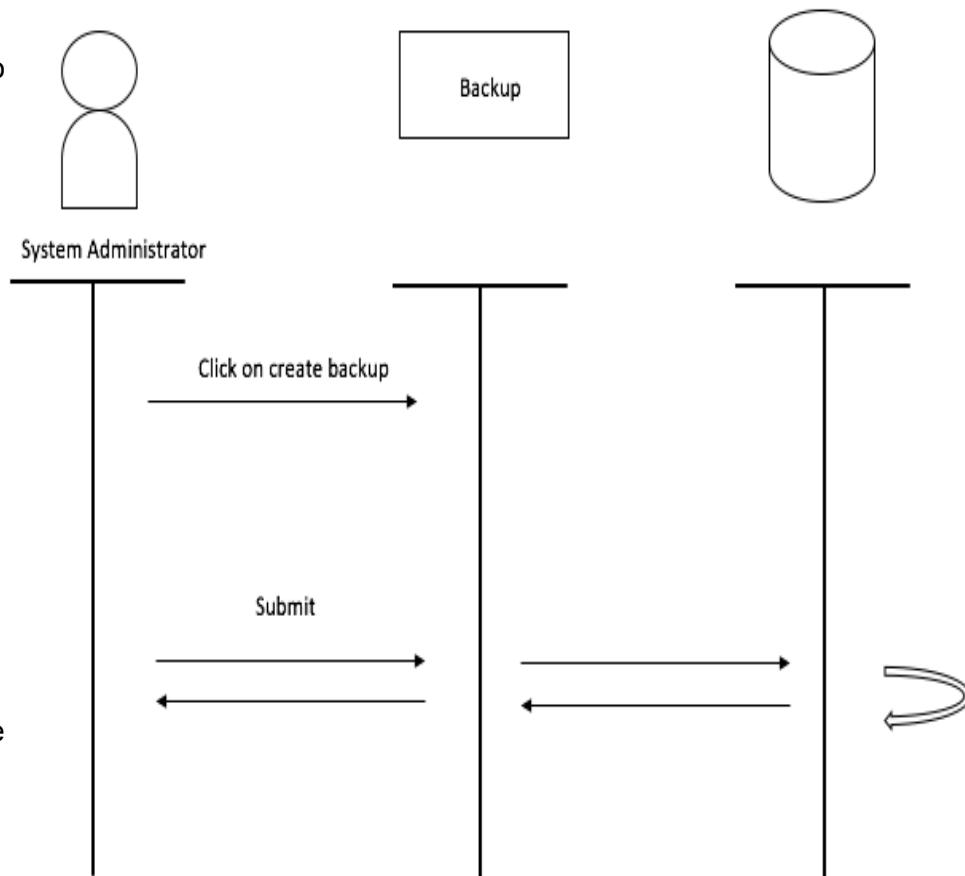
### 26. Pre-Conditions

- The system must have data to back-up

### 27. Post-Conditions

- The data is backed-up
- The data is not saved

Sequence Diagram for Use-Case ID 39: Create Backup



# Use Case Specification: Recover Back-up

## 28. Use-Case Name: Recover Back-up: I.D. 42

### 28.1 Brief Description

The system will need to allow the **system administrator** to recover its most recent back-up.

### 29. Flow of Events

#### 29.1 Basic Flow

- The system administrator presses a button to recover back-up
- The system will process the request
- The system will complete the request
- All of the data is recovered

#### 29.2 Alternative Flows

##### 29.2.1. *Back-up fails*

- The system administrator presses a button to recover back-up
- The system processes the request
- The system denies the request
- The data is not recovered

### 30. Special Requirements

- The system administrator must have a need to recover the backed-up data

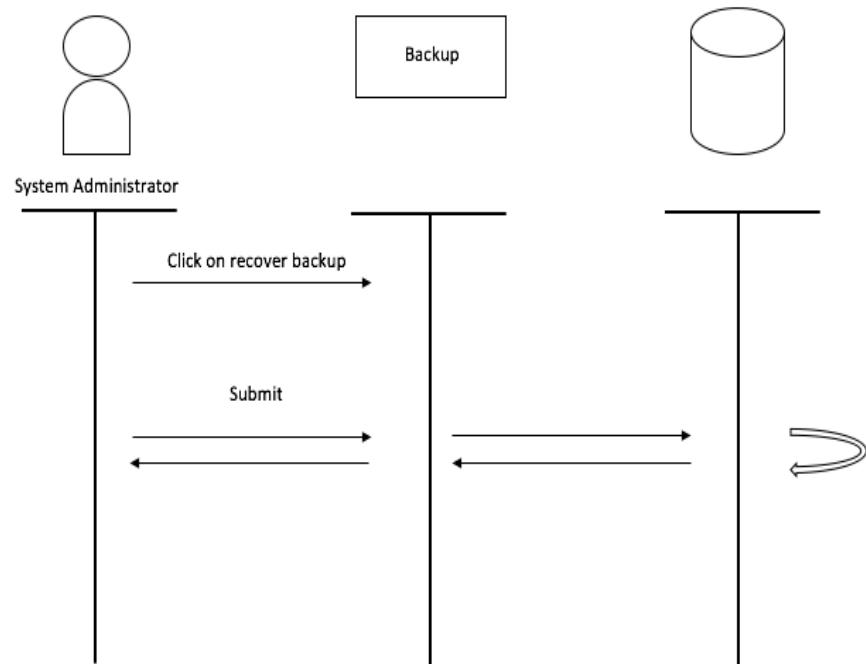
### 31. Pre-Conditions

- There must be data that was recently backed-up

### 32. Post-Condition

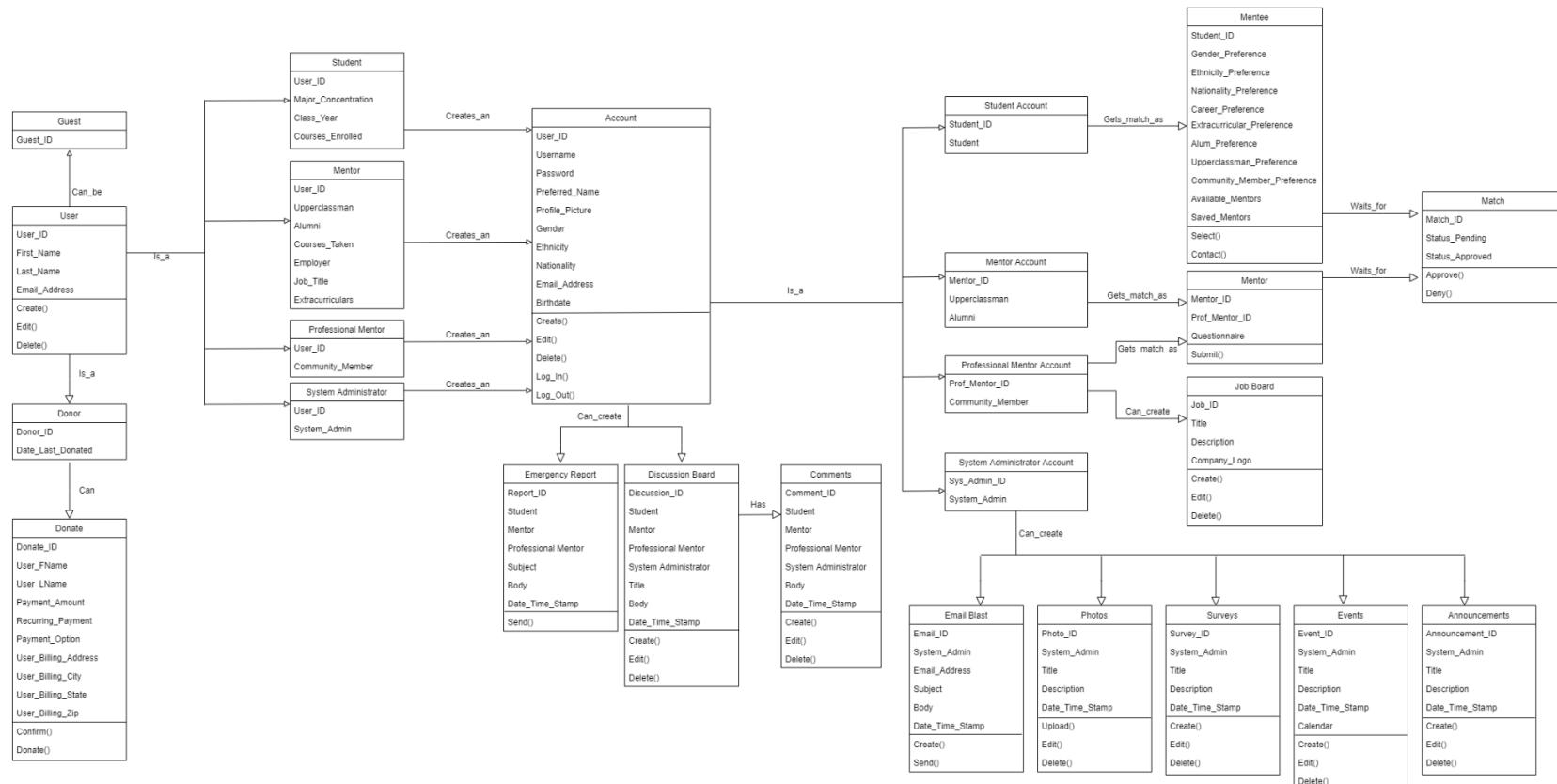
- The data from the recent back-up is recovered
- The data from the recent back-up is lost

Sequence Diagram for Use-Case ID 40: Recover Backup



# Class Diagram

Pictured below is the class diagram for the COB Web system. A class diagram is a static structure diagram that describes the structure of a system by showing the system's classes, attributes, methods, and relationships among objects. We can read this class diagram from left to right, following the arrows, it will show the relationships among the objects. At the top of each table reads the class name, followed by the attributes, and finally the methods.



## Verb-Noun Analysis

Looking at the diagram from the left, we start off with a class named User. A **User** can be a **Guest** or is either a **Student, Mentor, Professional Mentor, or a System Administrator**. A **User**, any user that either has an account or does not have an account, will be able to *make* donations hence the relationship pointing down towards the **Donate** class.

A **Student, Mentor, Professional Mentor, and System Administrator** can *create* an **Account** for the system. Each of these users will be able to Create, Edit, and Delete their accounts. They will also be able to Log-In and Log-Out of the system once their account is created. These users will also have the ability to *create* **Emergency Reports** that will be sent directly to the system administrator. These users can also create, edit, and delete **Discussion Board** posts and **Comments** on Discussion Board posts. Lastly, these users can create, edit, and delete **Photos**. The Photos will contain a title, description, and date time stamp.

A **Student Account** matches as a **Mentee**. In the matching process, the mentee will choose their preferences for their mentors. The attributes in the **Mentee** class are all the preferences that the mentee will identify. After the **Mentee** has selected and contacted their mentor, they will *wait* for the **Match** to be approved or denied by the mentor they have chosen.

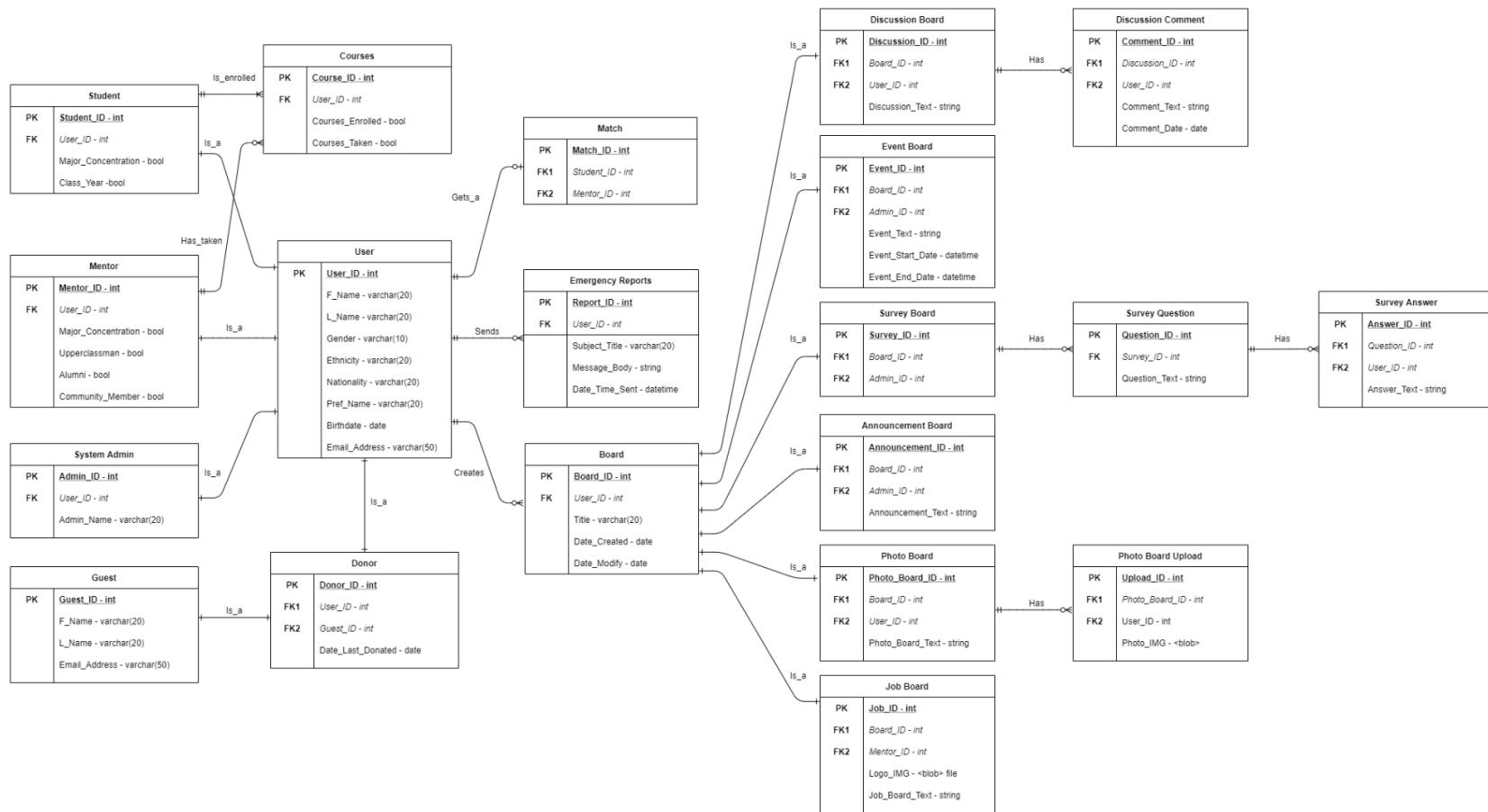
A **Mentor Account** and a **Professional Account** matches as a **Mentor**. A **Mentor Account** can be either an Upperclassman or an Alumni. A **Professional Mentor Account** is an authorized member of the community. In the matching process, the mentor will be asked a series of questions in a questionnaire that will be given to mentees to read. After mentors are done with their questionnaires, they will submit the answers and *wait* for a mentee to contact them so they can approve or deny the **Match**.

A **Professional Mentor Account** can *create* **Job Board** postings. A **Job Board** posting will have a title, description, and contain the company's logo if applicable. The professional mentor will be able to create, edit, and delete job board postings.

A **System Administrator Account** can *create* **Email Blasts, Surveys, Events, and Announcements**. An **Email Blast** will contain a subject, body, recipient email addresses, and the date time stamp. The system administrator will be able to send email blasts. The system administrator will be able to upload, edit, and delete all photos. The **Surveys** will contain a title, description, and date time stamp. The system administrator will be able to create, edit, and delete all surveys. The **Events** will contain a title, description, date time stamp, and a calendar displaying all events. The system administrator will be able to create, edit, and delete all events. The **Announcements** will contain a title, description, and date time stamp. The system administrator will be able to create, edit, and delete all announcements.

# Database Design - Entity Relationship Diagram

The following is the ERD (Entity Relationship Diagram) for the system's database. An ERD lays out the major entities within the system and the inter-relationships among these entities. For the purpose of privacy, we did not include a Donations entity table therefore, all financial transactions that will be handled in this system will go through a payment gateway called Braintree.



## Data Definitions

In this portion of the document, we will be defining the data types that will be used and stored in the database. The data types that will be used in the database are: <BLOB> (large binary objects), INT (numbers that you can do math with), BOOL (true or false values), DATE (holds the date), DATETIME (holds the date and the time), STRING (a text string) and VARCHAR (variable character strings that store the number assigned). The following is a breakdown of those data types including the table names:

USER TABLE

Attribute	Data Type	Storage	Key Type	Description
Nationality	varchar	20 characters		Holds information on user nationality
Gender	varchar	20 characters		Holds information on user gender
Ethnicity	varchar	20 characters		Holds information on user ethnicity
Fname	varchar	20 characters		Holds information on user first name
Lname	varchar	20 characters		Holds information on user last name
PrefName	varchar	20 characters		Holds information on user preferred name
Email	varchar	50 characters		Holds information on user email address
Bday	DATE	3 bytes		Holds information on user birthday
User_ID	int	4 bytes	PRIMARY	Unique identifier for each user entity

*Entity table that stores basic information for all users*

STUDENT TABLE

Attribute	Data Type	Storage	Key Type	Description
Major	bool	1 byte		States student's CIS Major Concentration
Year	bool	1 byte		States student's year (freshman, sophomore, junior, senior)
Courses_Enrolled	bool	1 byte		Lists student's CIS courses enrolled
Student_ID	int	4 bytes	PRIMARY	Unique identifier for student
User_ID	int	4 bytes	FOREIGN	Unique identifier for user, relates one student entity to one user entity from user table where ID matches

*Entity subtype of user table that has a one to one relationship with user table*

**MENTOR TABLE**

Attribute	Data Type	Storage	Key Type	Description
Major	bool	1 byte		States Mentor's Major
Upperclassman	bool	1 byte		Signifies whether or not mentor is upperclassman
Alumni	bool	1 byte		Signifies whether or not mentor is UofL alumni
Courses_Taken	bool	1 byte		List mentor's relevant computer courses enrolled/taken
Mentor_ID	int	4 bytes	PRIMARY	Unique identifier for mentor
User_ID	int	4 bytes	FOREIGN	Unique identifier for user, relates one mentor entity to one user entity from user table where ID match

*Entity subtype of user table that has a one to one relationship with user table*

**ADMIN TABLE**

Attribute	Data Type	Storage	Key Type	Description
Admin_Name	varchar	20 characters		Special admin name identifier
Admin_ID	int	4 bytes	PRIMARY	Unique identifier for mentor
User_ID	int	4 bytes	FOREIGN	Unique identifier for user, relates one admin entity to one user entity from user table where ID match

*Entity subtype of user table that has a one to one relationship with user table*

**COURSES TABLE**

Attribute	Data Type	Storage	Key Type	Description
Courses_Enrolled	bool	1 byte		List student's enrolled courses
Courses_Taken	bool	1 byte		List mentor's courses taken
Course_ID	int	4 bytes	PRIMARY	Unique identifier for courses
User_ID	int	4 bytes	FOREIGN	Unique identifier for user, relates one course entity to one user entity from user table where ID match

*Entity table that has a one to many relationship with student and mentor*

**DONOR TABLE**

Attribute	Data Type	Storage	Key Type	Description
Donor_ID	int	4 bytes	PRIMARY	Unique identifier for donor
Guest_ID	int	4 bytes	FOREIGN	Unique identifier for guests that relates to donor
User_ID	int	4 bytes	FOREIGN	Unique identifier for user that relates donor
Date_Of	date	3 bytes		Date of recorded donation
Date_Amount	int	4 bytes		Amount donated at date of entry

*Entity table that has a one to many relationship with user table*

**GUEST TABLE**

Attribute	Data Type	Storage	Key Type	Description
F_Name	varchar	20 characters		Holds information on guest first name
L_Name	varchar	20 characters		Holds information on guest last name
Email_Address	varchar	50 characters		Holds information on guest email address
Guest_ID	int	4 bytes	PRIMARY	Unique identifier for guest

*Entity table that has a one to one relationship with donor table*

**BOARD TABLE**

Attribute	Data Type	Storage	Key Type	Description
Title	varchar	20 characters		Holds information on title of board post
Date_Created	date	3 bytes		Holds information on date post created
Date_Modify	date	3 bytes		Holds information on date post modified
Board_ID	int	4 bytes	PRIMARY	Unique identifier for boards
User_ID	int	4 bytes	FOREIGN	Unique identifier for user, relates one board entity to one user entity from user table where ID match

*Entity table that has a one to many relationship with user table*

**DISCUSSION BOARD TABLE**

Attribute	Data Type	Storage	Key Type	Description
Discussion_Text	text	2GB		Stores discussion board text
Discussion_ID	int	4 bytes	PRIMARY	Unique identifier of discussion board
User_ID	int	4 bytes	FOREIGN	Identifies which user created the post

*Entity table that has a one to one relationship with board table*

**DISCUSSION COMMENT TABLE**

Attribute	Data Type	Storage	Key Type	Description
Comment_Text	text	2GB		Stores discussion board comments as a text string
Comment_Date	datetime	4 bytes		Stores comments date and time posted
Comment_ID	int	4 bytes	PRIMARY	Uniquely identifies each comment
Discussion_ID	int	4 bytes	FOREIGN	Identifies which discussion post comment is linked to
User_ID	int	4 bytes	FOREIGN	Identifies which user commented

*Entity table that has a one to many relationship with discussion board table*

**EVENT BOARD TABLE**

Attribute	Data Type	Storage	Key Type	Description
Event_Text	text	2GB		Stores the description of an event
Event_Start_Date	date	3 bytes		Stores the start date of an event
Event_End_Date	date	3 bytes		Stores the end date of an event
Event_ID	int	4 bytes	PRIMARY	Unique identifier for events posted on event board
Board_ID	int	4 bytes	FOREIGN	Identifies the board linked to event board
Admin_ID	int	4 bytes	FOREIGN	Identifies the admin that created the event

*Entity table that has a one to one relationship with board table*

**SURVEY BOARD TABLE**

Attribute	Data Type	Storage	Key Type	Description
Survey_ID	int	4 bytes	PRIMARY	Unique identifier of survey board
Board_ID	int	4 bytes	FOREIGN	Identifies the board that is linked to survey board
Admin_ID	int	4 bytes	FOREIGN	Identifies which admin created the survey

*Entity table that has a one to one relationship with board table*

**SURVEY QUESTION TABLE**

Attribute	Data Type	Storage	Key Type	Description
Questions_Text	text	1GB of text data		Stores the survey questions in a text string
Question_ID	int	4 bytes	PRIMARY	Uniquely identifies each question in the survey
Survey_ID	int	4 bytes	FOREIGN	Identifies which survey questions belongs to which survey

*Entity table that has a one to many relationship with survey board table*

**SURVEY ANSWER TABLE**

Attributes	Data Type	Storage	Key Type	Description
Answer_Text	text	1GB of text data		Stores the survey answer in a text string
Answer_ID	int	4 bytes	PRIMARY	Uniquely identifies every survey answer
Question_ID	int	4 bytes	FOREIGN	Identifies which answers belongs to which questions
User_ID	int	4 bytes	FOREIGN	Identifies which user answered

*Entity table that has a one to many relationship with survey question table*

**ANNOUNCEMENT BOARD TABLE**

<b>Attribute</b>	<b>Data Type</b>	<b>Storage</b>	<b>Key Type</b>	<b>Description</b>
Announcement_Text	text	2GB		Stores the announcement in a text string
Announcement_ID	int	4 bytes	PRIMARY	Uniquely identifies each announcement created
Board_ID	int	4 bytes	FOREIGN	Identifies the board that is linked to announcement board
Admin_ID	int	4 bytes	FOREIGN	Identifies which admin user created the announcement

*Entity table that has a one to one relationship with board table*

**PHOTO BOARD TABLE**

<b>Attribute</b>	<b>Data Type</b>	<b>Storage</b>	<b>Key Type</b>	<b>Description</b>
Photo_Board_Text	text	1GB of text data		Stores the photo board description as a text string
Photo_Board_ID	int	4 bytes	PRIMARY	Uniquely identifies each photo board post
Board_ID	int	4 bytes	FOREIGN	Identifies the board that is linked with photo board
User_ID	int	4 bytes	FOREIGN	Identifies which user created the post

*Entity table that has a one to one relationship with board table*

**PHOTO BOARD UPLOAD TABLE**

<b>Attribute</b>	<b>Data Type</b>	<b>Storage</b>	<b>Key Type</b>	<b>Description</b>
Photo_IMG	<blob>, .jpeg, .png	2GB		Stores the photo uploaded
Upload_ID	int	4 bytes	PRIMARY	Uniquely identifies each photo uploaded
Photo_Board_ID	int	4 bytes	FOREIGN	Identifies the photo board linked to the uploaded photo
User_ID	int	4 bytes	FOREIGN	Identifies which user uploaded the photo

*Entity that has a one to many relationship with photo board table*

**JOB BOARD TABLE**

<b>Attribute</b>	<b>Data Type</b>	<b>Storage</b>	<b>Key Type</b>	<b>Description</b>
Job_Board_Text	text	1GB		Stores the job description as a text string
Logo_IMG	<blob>, .jpeg, .png	1GB		Stores the company logo
Job_ID	int	4 bytes	PRIMARY	Uniquely identifies each job board post
Board_ID	int	4 bytes	FOREIGN	Identifies the board that is linked to the job board
Mentor_ID	int	4 bytes	FOREIGN	Identifies the mentor that created the job post

*Entity table that stores a job board entry and has many to one relationship with user*

**EMERGENCY REPORT TABLE**

<b>Attribute</b>	<b>Data Type</b>	<b>Storage</b>	<b>Key Type</b>	<b>Description</b>
Subject_Title	varchar	20 characters		Title of emergency report entry
Message_Body	text	2GB of text data		Binary Large object, stores event board entry
Date_Time_Sent	datetime	3 bytes		Date and time of report creation
Report_ID	int	4 bytes	PRIMARY	Uniquely identifies each report
User_ID	int	4 bytes	FOREIGN	Identifies which user is linked to the report

*Entity table that has a one to many relationship with user table*

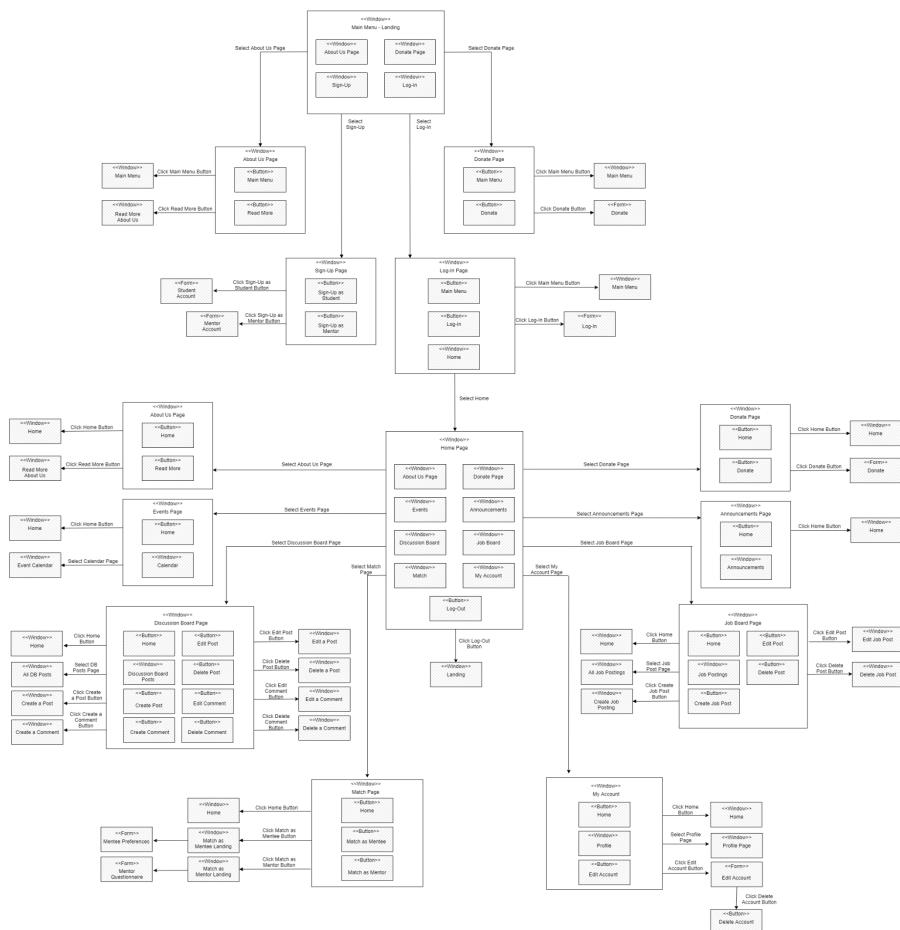
**MATCH TABLE**

<b>Attribute</b>	<b>Data Type</b>	<b>Storage</b>	<b>Key Type</b>	<b>Description</b>
Match_ID	int	4 bytes	PRIMARY	Unique identifier for match
Student_ID	int	4 bytes	FOREIGN	Unique identifier for student that relates match
Mentor_ID	int	4 bytes	FOREIGN	Unique identifier for mentor that relates match

*Entity table that has a one to one relationship with user table AND mentor table*

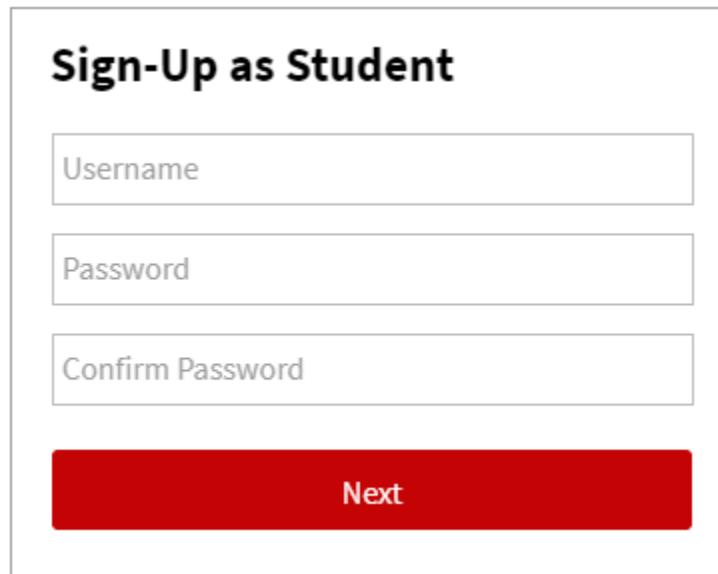
# User Interface Navigation Diagram - Window Navigation Diagram

A window navigation diagram defines the basic structure of the interface, it is a top-down model that shows the screens that the user will see if they were to navigate through the webpage. These diagrams show all the screens, forms, and reports in the system, how they are connected, and how the user moves from one to another. This leads us to the next portion of our presentation, the user-interface prototypes.



# Screen Layouts

This is the screen layout for students signing up for an account. This form asks students to create a username, password, and to confirm password. This relates to the use-case Create Student Account.



The image shows a wireframe-style representation of a sign-up form titled "Sign-Up as Student". The form consists of three input fields: "Username", "Password", and "Confirm Password", each enclosed in a light gray rectangular box. Below these fields is a large red rectangular button with the word "Next" in white capital letters. The entire form is contained within a larger gray rectangular frame.

**Sign-Up as Student**

Username

Password

Confirm Password

Next

This is the second page that the students will see when they sign-up for an account. This form will ask students for their first name, last name, email address, ethnicity, nationality, gender, and birthdate. This relates to the use-case Create Student Account.

## Sign-Up as Student

First Name

Last Name

Preferred Name

Email Address

Ethnicity

Nationality

Gender

Date of Birth

**Next**

This is the third page that students will see when they sign-up for an account. This form will ask students their class year, their major concentration, courses they are enrolled in currently, and if they would like to upload a profile picture. This relates to the use-case Create Student Account.

**Sign-Up as Student**

What is your class year?\*

Freshman  
 Sophomore  
 Junior  
 Senior

What courses are you enrolled in?\*

CIS205    CIS310  
 CIS305    CIS320  
 CIS199    CIS350  
 CIS150    CIS411  
 Web Development    CIS200    CIS410  
 Information Security  
 Business Process Management  
 Data Analytics

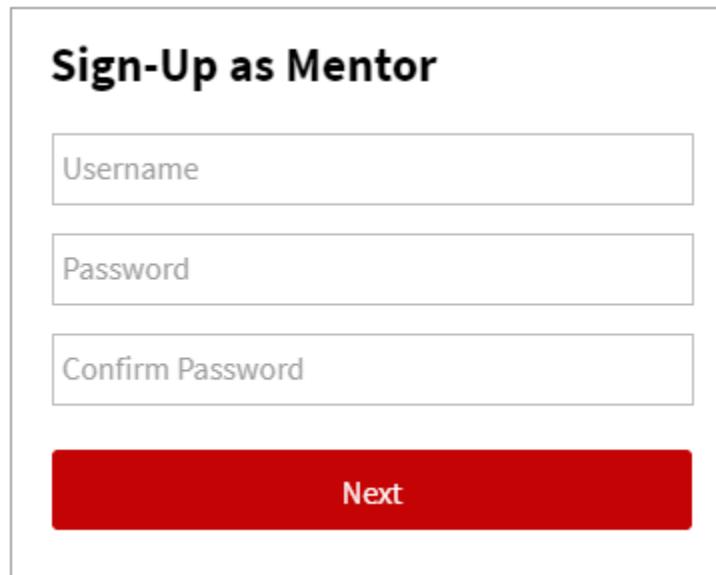
Other courses not listed:

Upload Profile Picture:

Drag and drop files here to upload  
( or )  
**Select files to upload**

**Finish Sign-Up**

This is the first page that mentors will see when they sign-up for an account. This form asks mentors to create a username, password, and to confirm their password. This relates to the use-case Create Mentor Account.



The image shows a user interface for 'Sign-Up as Mentor'. It features a title 'Sign-Up as Mentor' at the top. Below the title are three input fields: 'Username', 'Password', and 'Confirm Password'. A large red button labeled 'Next' is positioned below the input fields.

**Sign-Up as Mentor**

Username

Password

Confirm Password

Next

This is the second page that mentors will see when they sign-up for an account. This form asks mentors for their first name, last name, email address, ethnicity, nationality, gender, and birthdate. This relates to the use-case Create Mentor Account.

## Sign-Up as Mentor

First Name

Last Name

Preferred Name

Email Address

Ethnicity

Nationality

Gender

Date of Birth

**Next**

This is the third page that mentors will see when they sign-up for an account. This form asks mentors to identify whether they are an alum, upperclassman, or a member of the community. This form also asks if mentors have taken any of the courses listed, their major concentration if they have one, extracurriculars, employer, and job title. It also asks if they would like to upload a profile picture. This relates to the use-case Create Mentor Account.

Sign-Up as Mentor

Are you a UofL Alum?\*

Yes  
 No

Are you a CIS upperclassman?\*

Yes  
 No

Are you a community member?\*

Yes  
 No

What courses have you taken?

CIS205     CIS310  
 CIS305     CIS320  
 CIS199     CIS350  
 CIS150     CIS411  
 CIS200     CIS410

Other courses not listed:

What was your concentration?

Web Development     Business Process Management  
 Information Security     Data Analytics

Extracurriculars:

Employer:   
Job Title:

Upload Profile Picture:

Drag and drop files here to upload  
(or)

This is the form that users with an authorized account will see when they log-in to the system. They will be asked to enter their email address and password. This relates to the use-case Log-In.

## Welcome! Please Log-In

Email

Password

Show Password

Login

Forgot your password?

This is the form that users without an authorized account will see when they log-in to the system. The system will have the option for users to log-in as a guest. This is reserved for users who are not students, mentors, or professional mentors. This will give guest users access to the donation page and the about us page. They will be asked to enter their first and last name.

### Log-In as Guest

First Name

Last Name

Email Address

**Log-In as Guest**

When students who choose to be a mentee go through the matching process, this is the form that they will see after they click into match as a mentee. They will be asked to fill out several preferences in order to better filter the pool of available mentors. This relates to the use-case Select Mentor.

**What are you looking for in a mentor?**

Do you prefer to be paired with an alum?  Yes  No Gender Preference:

Do you prefer to be paired with an upperclassman?  Yes  No Ethnicity Preference:

Do you prefer to be paired with a member of the community?  Yes  No Nationality Preference:

Concentration Preference:  Business Process Management  Web Development  Information Security  Data Analytics Extracurricular Preference:   
Job Title Preference:

**Next**

When mentors go through the matching process, this is the form they will see after they click into match as mentor. This is the questionnaire that has been prepared for mentors to answer in order for mentees to get to know them better.

**Questionnaire for Mentors**

What are your hobbies?

What is one fun fact about yourself?

Where is your hometown?

What is your preferred pronoun?

How many languages do you speak?

Favorite movie?

Favorite show?

Favorite book?

Favorite color?

Favorite music genre?

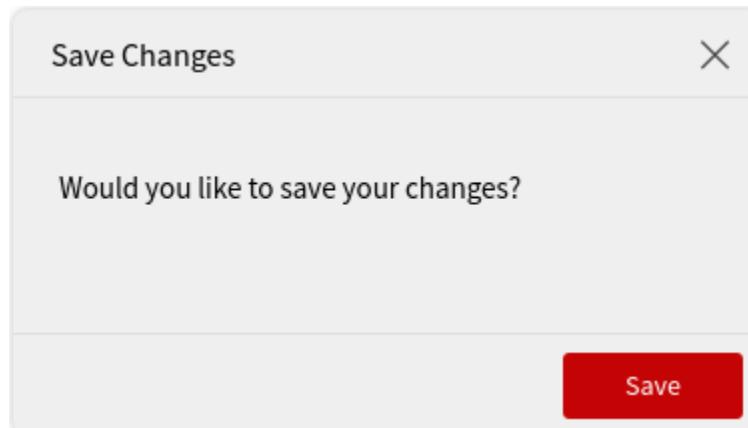
Favorite food?

**Next**

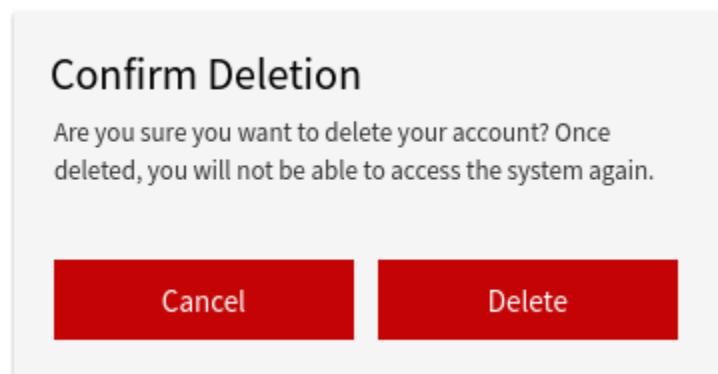
This is the last question that mentors will see on their questionnaire. It is an open-ended question that is meant to give mentees a better understanding of who the mentors are as a person.

# Questionnaire for Mentors

This is the pop-up window that shows whenever a user wants to edit their account information and save changes made to their profiles. They can also click the X in the top right corner if they are still editing their profiles. This relates to the use-cases Edit Student Account, Edit Mentor Account, Edit Discussion Board Post, and Edit Comment.



This is the pop-up window that shows whenever a user wants to delete their account. The pop-up window asks if they would like to confirm deletion of their account. They can click on the cancel button or the delete button. This relates to the use-case Delete Student Account and Delete Mentor Account.



This is the form that shows whenever a user wants to create a comment on a discussion board post. They will be asked to enter their name and the body of the comment. This relates to the use-case Create Comment on Discussion.

## Comments (2)



**John Doe**  
March 12, 2017 at 1:28pm

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui.



**Jane Doe**  
March 12, 2017 at 1:28pm

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui.

## Create a Comment

Name

Comment

[Delete Comment](#)

[Create Comment](#)

This is the form that shows whenever a user wants to edit their comment. The user will be asked to save their changes or if they want to delete their comment, they will have the option to do so on this page. This relates to the use-case Edit Comment on Discussion and Delete Comment on Discussion.

### Comments (2)



**John Doe**  
March 12, 2017 at 1:28pm

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui.



**Jane Doe**  
March 12, 2017 at 1:28pm

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui.

### Edit a Comment

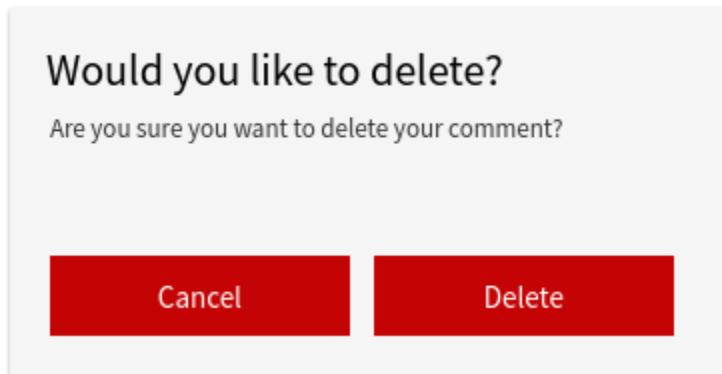
Name

Comment  

Placeholder text for comment area

[Delete Comment](#) [Save Changes](#)

This is the pop-up window if the user clicks on the delete comment button. They will be asked if they are sure they want to delete their comment. This relates to the use-case Delete Comment on Discussion.



This is the form that users will see when they want to create a discussion board post. They will be asked to enter their name, a title and a body then create when done. This relates to the use-case Create Discussion board post.

# Create a Discussion Board Post

Name  
Jane Doe

Title  
Tips for someone looking for an internship

Body:  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui.

This is the form that users will see when they want to edit their discussion board post. They can also delete their post through this page as well. This relates to the use-case Edit Discussion board post and Delete Discussion board post.

## Edit Discussion Board Post

Name

Title

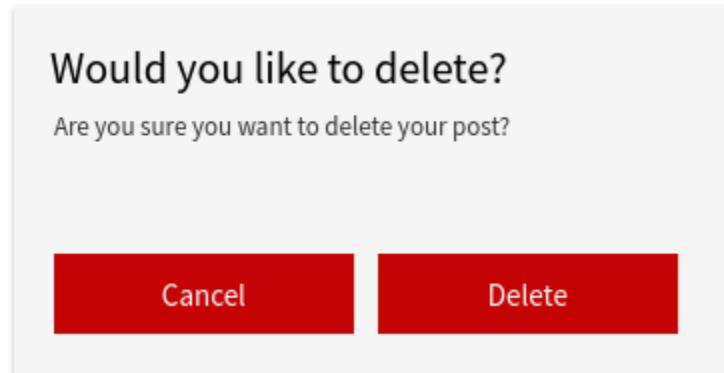
Body:

Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui.

Delete Post

Submit Changes

This is the pop-up window users will see when they want to delete their discussion board post. They will be asked to confirm if they want to delete their post. This relates to the use-case Delete Discussion board post.



This is the form that allows users to report any issues that may be experiencing anonymously directly to the system administrator. The user will be asked to enter a title and the body of the message. When they are done they can click the submit button. There is also an option to cancel if the user decides not to send the report. This relates to the use-case Send Report.

**Report**

Having issues? Report them anonymously directly to the system administrator here:

Subject:

<Subject Title>

Body:

<Body of Message>

CancelSubmit

This is the form that users will see if they want to make a donation. They will be asked to fill out payment amount, description of the payment, and the payment type. They will also be asked to enter their first name, email, billing address, and payment method. Once the user is done, they can submit their payment. This relates to the use-case Donate.

## Braintree Advanced Payment Terminal

### Payment Details

\*Amount

\*Description

\*Payment Type  One Time  Recurring

### Billing Address

\*Address

\*City

\*State/Zip  Zip

\*Country

### Your Information

\*Name

\*Email

### Payment Method

\*Name on Card

\*Card Number

\*Expiration/CVV     CVV

Submit Payment

This is the form that users will see when they choose to upload a photo from the photo board. This form asks users to enter a title for the photo, a description, and to choose a file to upload. Once they are done, they will click the upload photo button to publish their photo board post.

The screenshot shows a user interface for uploading a photo. At the top, the title "Upload Photo" is displayed in a large red font. Below it, there are two input fields: "Enter Photo Title" and "Enter Photo Description", both in black text. Underneath these fields is a dashed rectangular area containing a red cloud icon with an upward arrow, indicating where files can be uploaded. Below the icon, the text "Drag and drop files here to upload" is displayed, followed by "(or)". A dark grey button labeled "Select files to upload" is located at the bottom of this area. In the bottom right corner of the entire form, there is a large red button labeled "Upload Photo" in white text.

This is the form that professional mentors will see when they want to create a job post to the job board. They will be asked to enter a job title, the job description, and job requirements. They will also have the option to upload a company logo. This relates to the use-case Create Job Post.

## Create Job Post

Enter Job Title

Enter Job Description

Enter Job Requirements

Upload Company Logo

Upload File

Create Job Post

This is the form that professional mentors will see when they choose to edit their job posting on the job board. They will have the option to edit any information they entered when they created the job post. They will also have the option to delete the job post entirely. This relates to the use-case Edit Job Post and Delete Job Post.

## Edit Job Post

KFC Hiring IT Interns

KFC is now hiring IT interns for the fall semester. Please click [here](#) for more information about the job.

- Current CIS Student
- 3.0 GPA or better
- Ability to work 20+ hours per week

Upload Company Logo



Upload File

Delete Job Post

Save Changes

This is the page that the system administrator will see when they log-in to the system. They will have access to creating, editing, and deleting events, surveys, photos, and announcements. They will also have access to the reports that are sent anonymously by students and or mentors. This relates to the use-cases Create, Edit, and Delete Announcements, Events, Photos and the Send Report use-case.

# Welcome to the CoB Web!

We are so happy to have you here.

< >

## Announcements + /

Hope everyone is having a wonderful spring break! - Dr. B

## Events + /

Networking Event...10/02  
Business Career Fair...09/05  
COB Networking Event...04/12

## Surveys + /

No available survey at the moment

## Reports + /

You have one unread report...

This is the form that the system administrator will see when they want to create an announcement. They will be asked to enter the announcement title and announcement description/body. Once they have completed, they will click the create announcement button. This relates to the use-case Create Announcement.

## Create an Announcement

Enter Announcement Title

Enter Announcement Description

**Create Announcement**

This is the form that the system administrator will see when they want to edit and or delete their announcement. They will be asked to enter any modifications made and once finished they will click the save changes button and it will publish their changes. They may also delete their announcement altogether by clicking the delete announcement button. This relates to the use-case Edit and Delete Announcement.

## Edit Announcement

Enter Announcement Title

Enter Announcement Description

Delete Announcement

Save Changes

This is the form that the system administrator will see when they want to create an event for the events page. They will be asked to enter the event title and the event description. They will also select the event date on the calendar. Once they have finished, they will click on the create event button. This relates to the use-case Create Events.

## Create an Event

Enter Event Title

Enter Event Description

Select Event Date:

Select Event Date:						
		June 2021				
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Create Event

This is the form that the system administrator will see when they want to edit or delete an event. They will be asked to enter any modifications made then click the save changes button. They also have the option to delete the event entirely by clicking on the delete event button. This relates to the use-case Edit and Delete Events.

## Edit Event

Enter Event Title

Enter Event Description

Delete Event

Select Event Date:

Select Event Date:						
		June 2021				
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Save Changes

This is the form that the system administrator will see when they want to create a survey. They will be asked to enter their survey questions. Once they are finished, they will click on the create a survey button. This relates to the use-case Create Survey.

## Create Survey

Question

Question

Question

Question

Question

Question

Question

Question

[Save Changes](#)

< >

Create a Survey

This is the form that the system administrator will see if they want to edit the survey. They will click on the pencil icon in order to make modifications. After they have finished, they will click on the save changes button to update their survey. This relates to the use-case Edit Survey.

## Create Survey

Question  

Question  

Question  

Question  

Question  

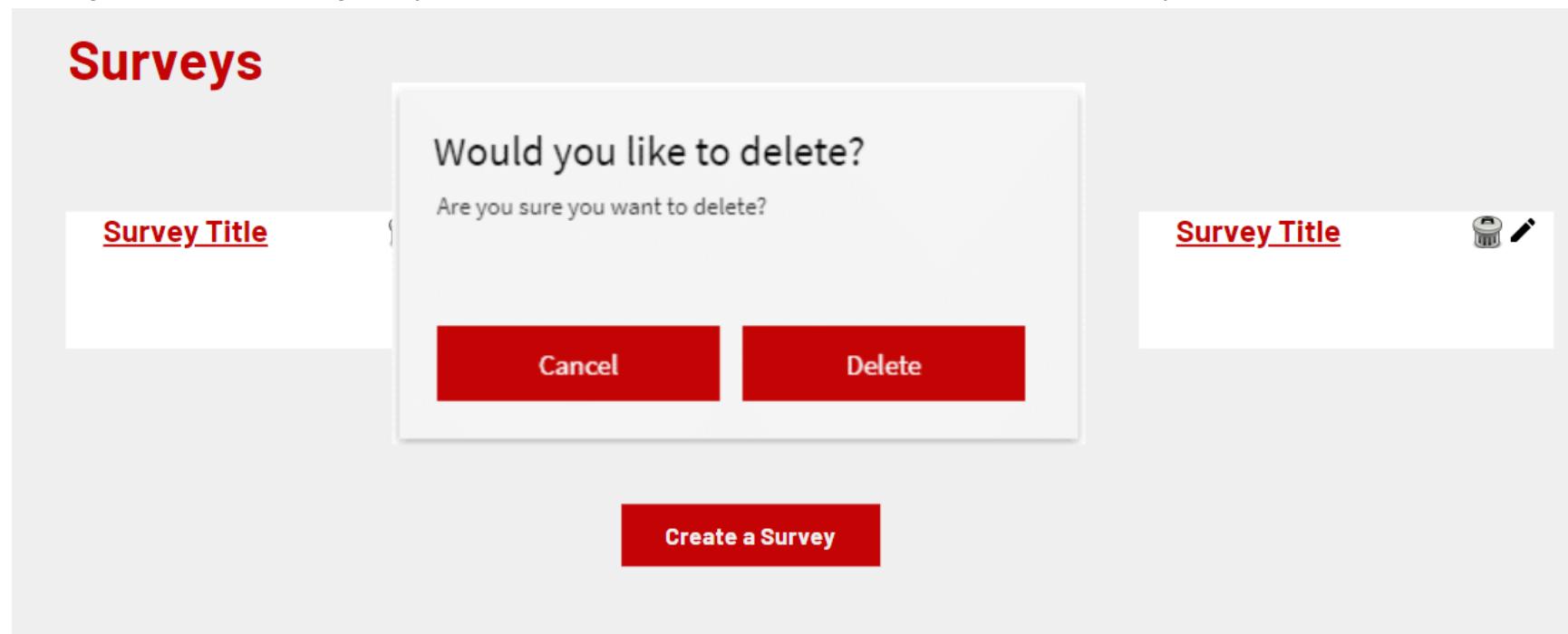
Question  

Question  

Question  

[Save Changes](#)

This is the page that the system administrator will see when they want to delete their survey. They will click on the trash can icon and a dialog box will show asking if they would like to delete. This relates to the use-case Delete Survey.



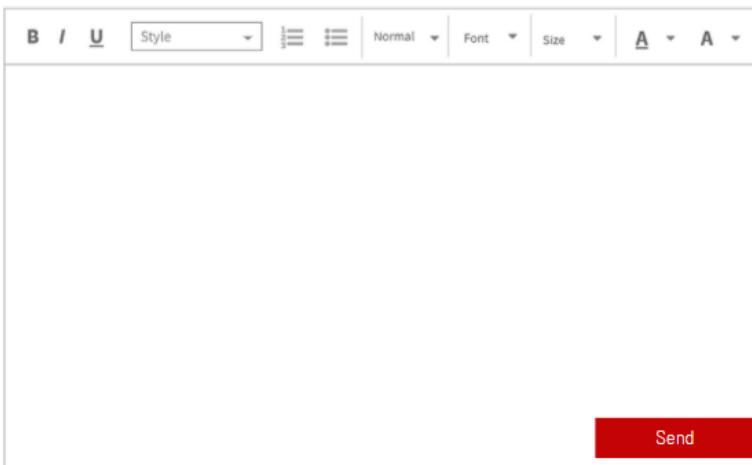
This is the form that the system administrator will see when they want to create an email blast. They will be asked to enter their email address, the recipients, subject, and body of the email. Once they are done, they will click the send button and the email will be sent to everyone in the system. This relates to the use-case Send Email Blast.

## **Compose Email Blast**

From: \_\_\_\_\_

To: \_\_\_\_\_

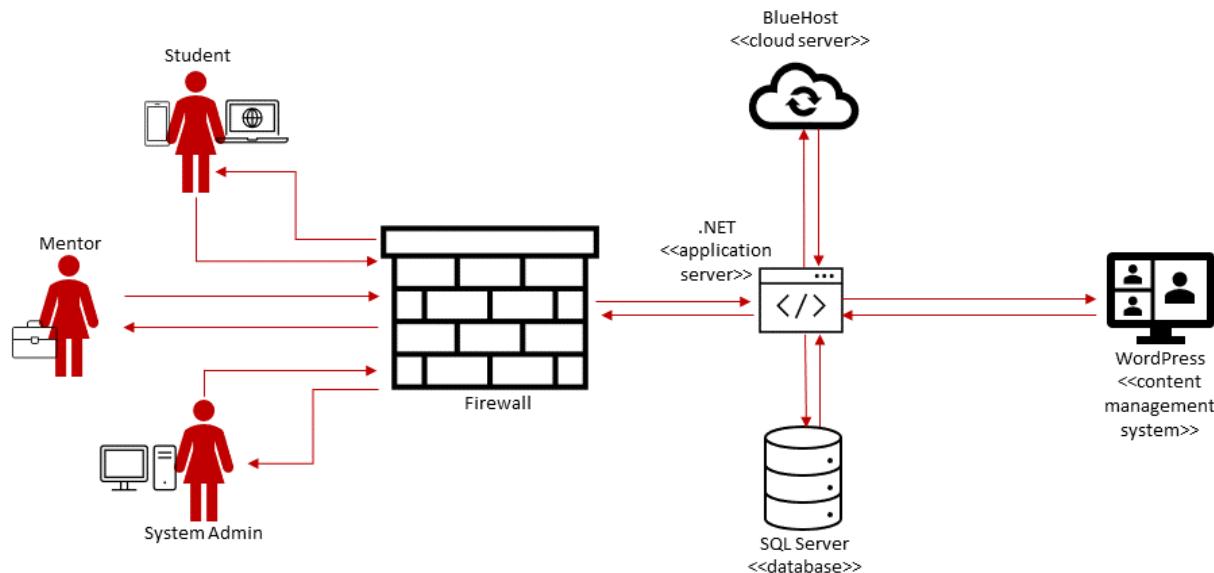
Subject: \_\_\_\_\_



The image shows a user interface for composing an email. At the top, there is a toolbar with icons for bold (B), italic (I), underline (U), style dropdown, alignment dropdown, font dropdown, size dropdown, and font color dropdown. Below the toolbar is a large text area for the email body. In the bottom right corner of the text area, there is a red "Send" button.

# Physical Architecture Design

The physical architecture design shows the process of accessing the system. There are three different types of users that will have access to the system: students, mentors, and system administrators. The system we intend to build will require five main components: a content management system, a database server, a cloud storage system, an application server, and a firewall. We will be using WordPress as our content management system. SQL Server will be our database server. BlueHost will be used as our cloud storage system. Our application server will be the .NET Framework. Lastly, we will use the firewalls provided by the cloud server for our firewall. Pictured below is the system's physical architecture design. The arrows are going to and from each component because information will be flowing both ways.



## Design Procedures: Security Concerns & Non-functional Requirements

1. The UofL CIS Mentorship system shall be created on a content management system called WordPress.
2. The UofL CIS Mentorship system shall be secured by a firewall.
3. The UofL CIS Mentorship system shall use SQL Server for database management.
4. The UofL CIS Mentorship system shall utilize a payment system called Braintree in order to collect donations.
5. The UofL CIS Mentorship system shall be user-friendly.
6. The UofL CIS Mentorship system shall be accessible 24/7.
7. The UofL CIS Mentorship system shall be backed-up automatically.
8. The UofL CIS Mentorship system shall have system administrator functionalities.

To ensure security for the non functional requirements, the system value, access control, encryption, and virus control requirements need to be used effectively. The system value is high as it helps provide a connection platform for students, professors, and community members. Additionally, it connects to a database that stores information on all these users. It is critical that it is secured. For that reason, users will have access to their data and be able to edit it. Certain user data will be visible to other users, but not editable. Only administrators can edit anyone's data. The data will be secured with encryption and proper program design to ensure that only the registered individual can access the data. The users will not have direct access to the database, instead having a class control what data passes between. The administrators will have access control lists to enforce the access requirements.

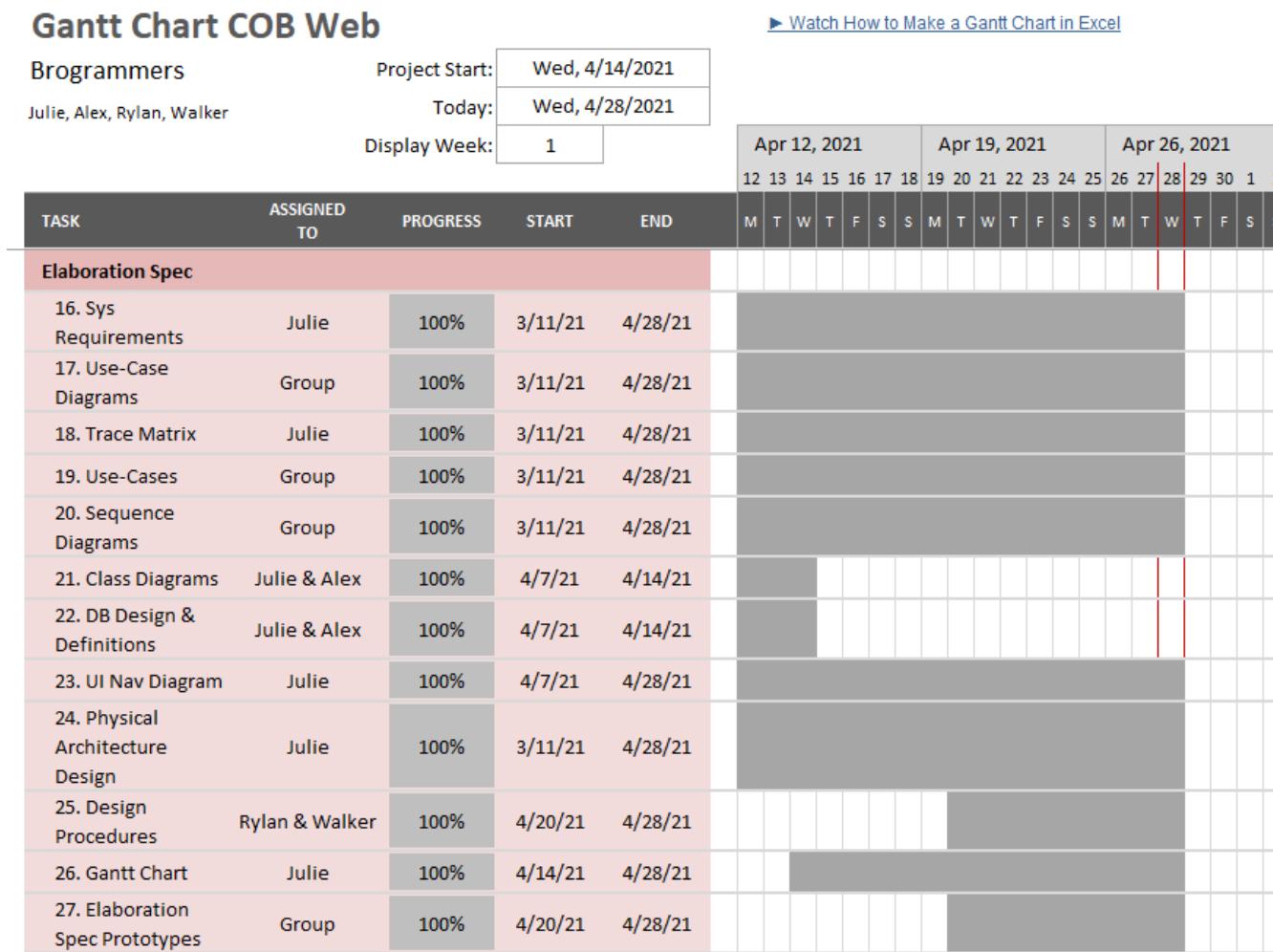
As this system is large, a public key encryption method will be used. A public key that is available will be used to encrypt data going between the site and user. A private key will then decrypt the data. This encryption will also be enforced on any calls between the site and the database. A separate public key encryption method is used with the Braintree plugin to process donation payments. To further ensure access control and encryption all accounts will have usernames and passwords. Passwords must be strong consisting of at least 12 characters with one number, one letter, and one special character. Passwords and usernames must be stored in encryptions to insure their security.

To meet virus control requirements, antivirus software will be implemented and run regularly. It will be kept up to date and monitored by administrators. Firewalls will also be implemented to ensure further protection. Furthermore, the software, website, plugins, and whole system will be updated regularly and swept for issues or breaches.

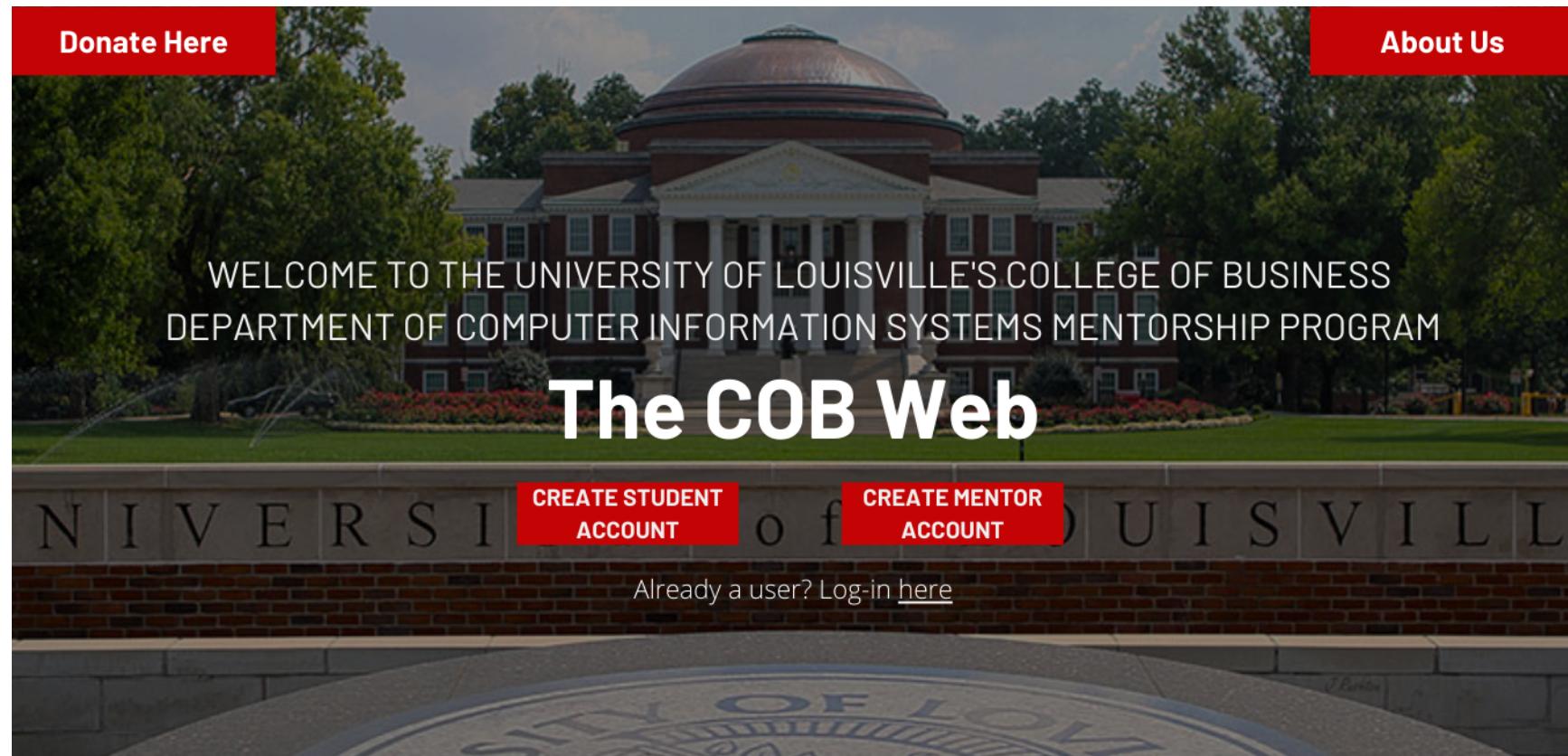
Lastly, the system will undergo regular backups of data and review information processed to ensure compromised or redundant data is removed. Transactions will also be checked and stored as well. Maintaining regular check ups by system administrators is key to keeping the system secured.

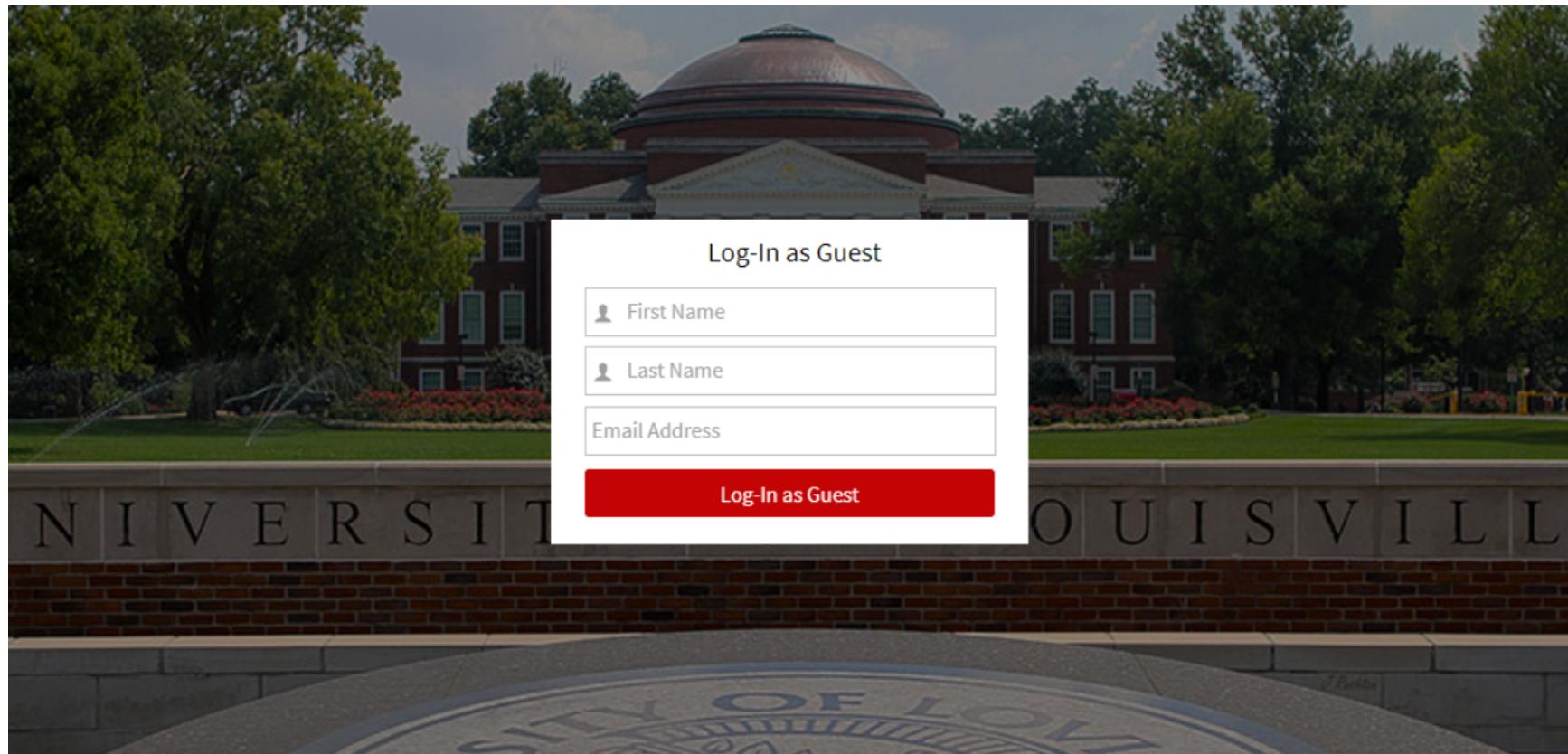
## Gantt Chart

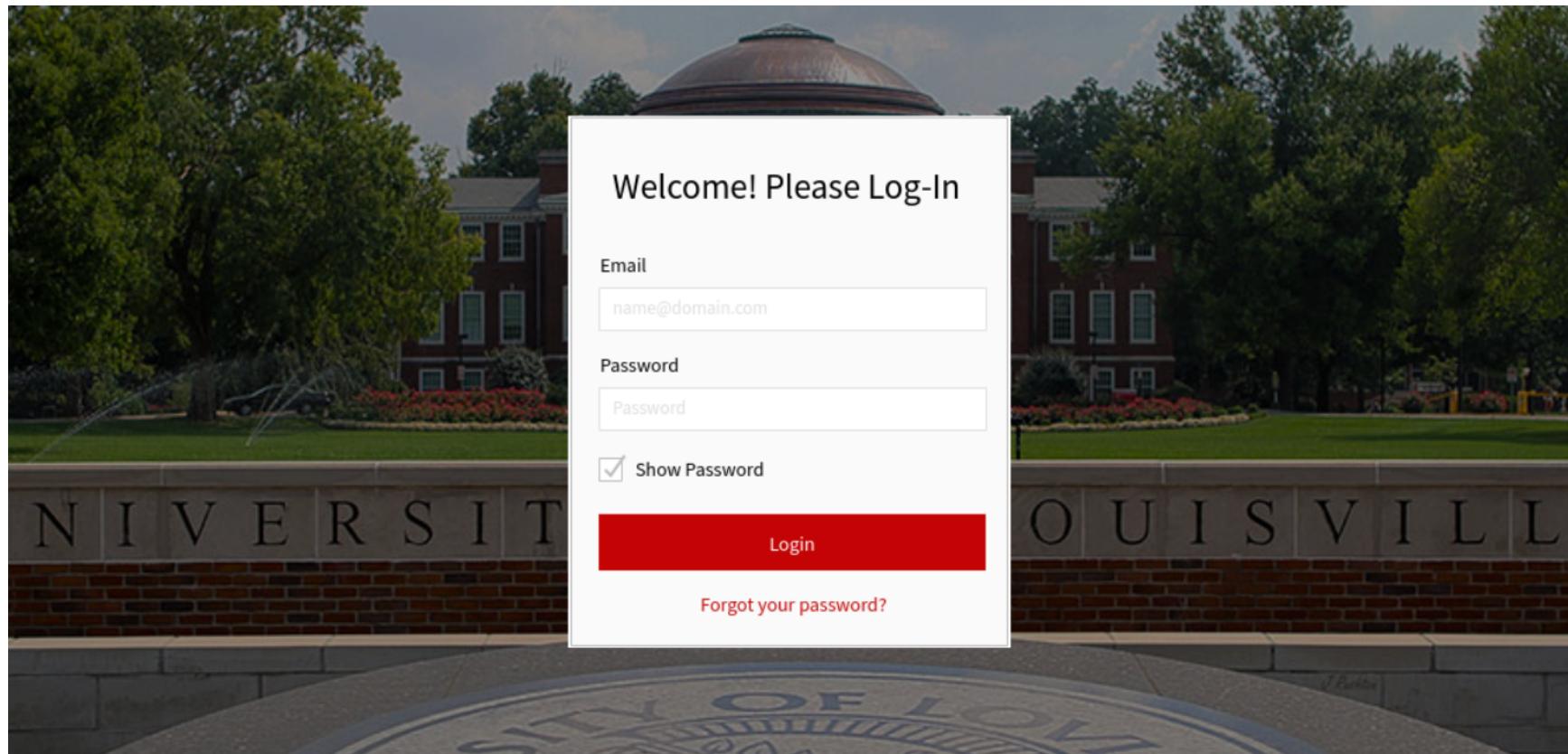
A Gantt Chart is used to visually express all of the work that is left to do in the project and how long each task will take. The Gantt Chart provides us with an ID for each task, the task names, the start and end date for each task, who the task is assigned to and it identifies which task is dependent on another. This chart is a tool used to keep the team on track and provide us with a timeline. These assignments balance the workload because each team member is responsible for an equal portion of the use-cases and the diagrams that are needed for the system.

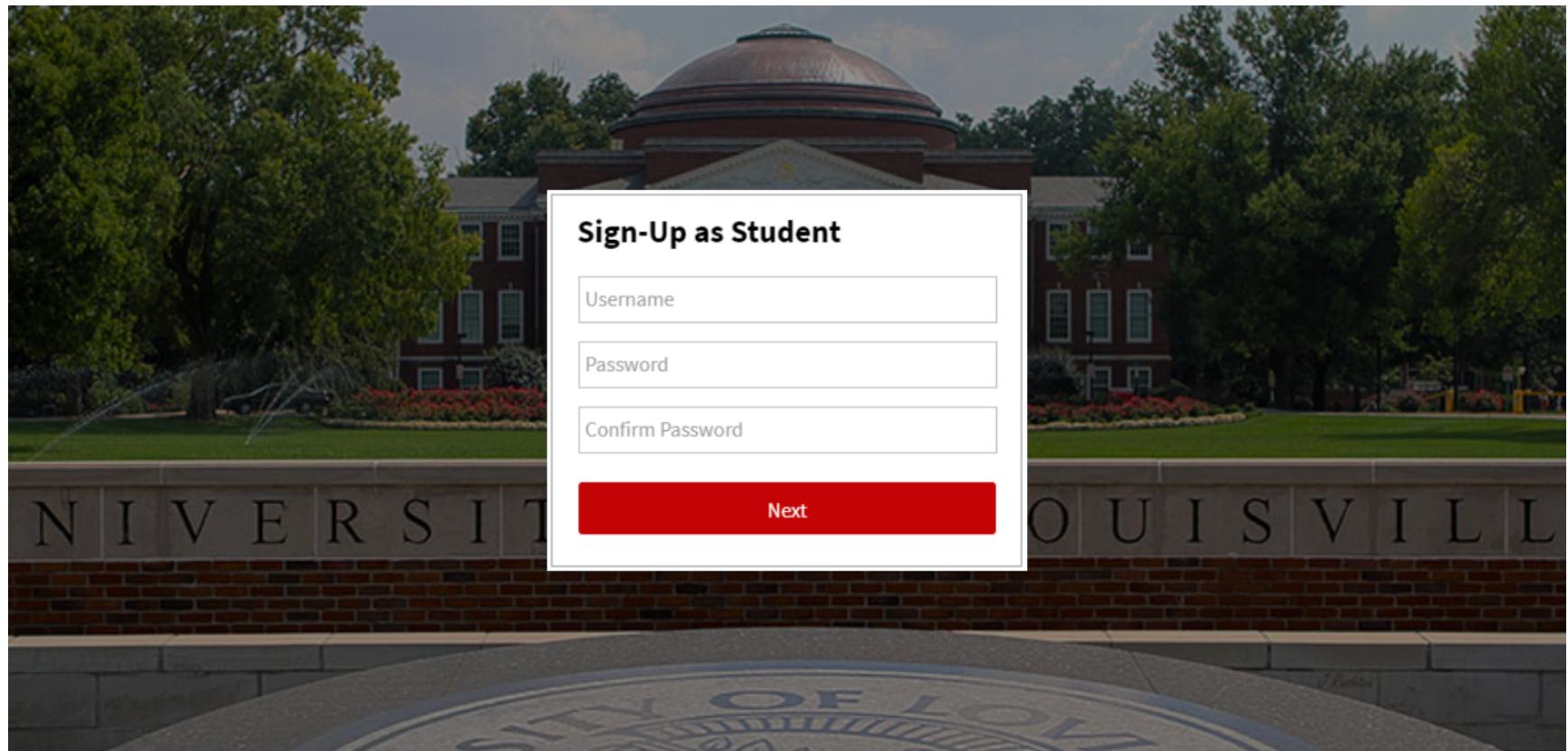


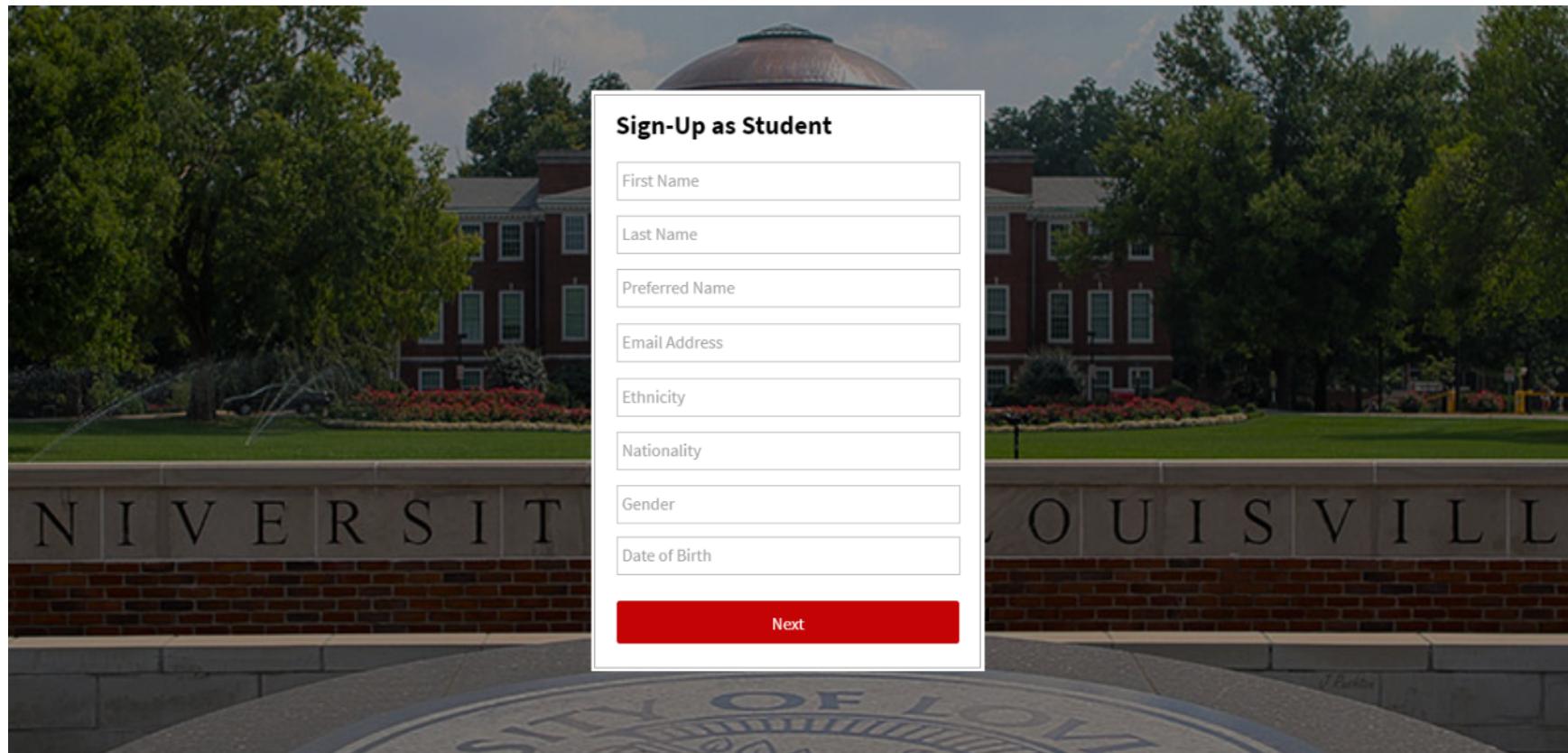
## Elaboration Phase Prototypes

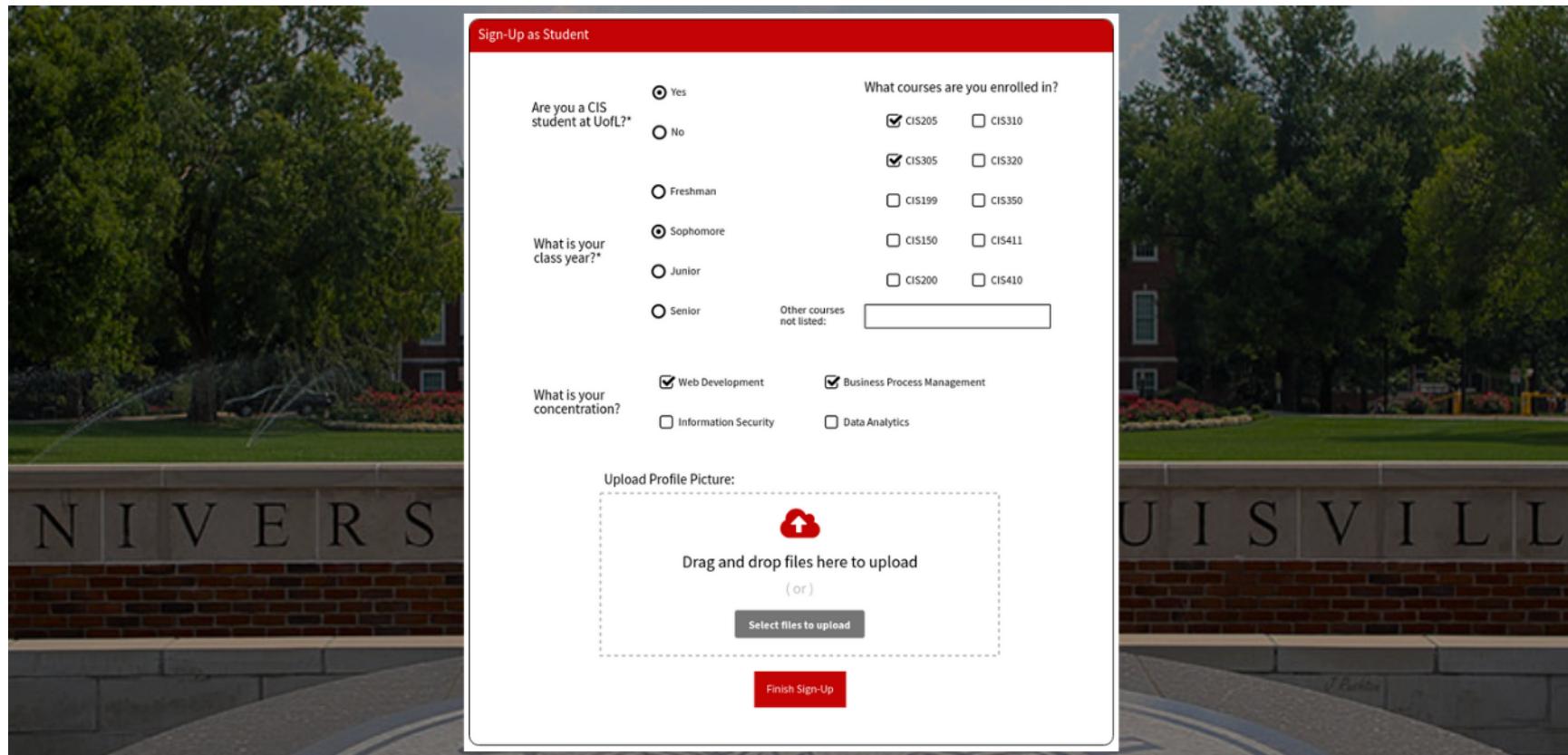












**Sign-Up as Student**

Are you a CIS student at UofL?\*

Yes  
 No

What is your class year?\*

Freshman  
 Sophomore  
 Junior  
 Senior

What courses are you enrolled in?

CIS205  CIS310  
 CIS305  CIS320  
 CIS199  CIS350  
 CIS150  CIS411  
 CIS200  CIS410

Other courses not listed:

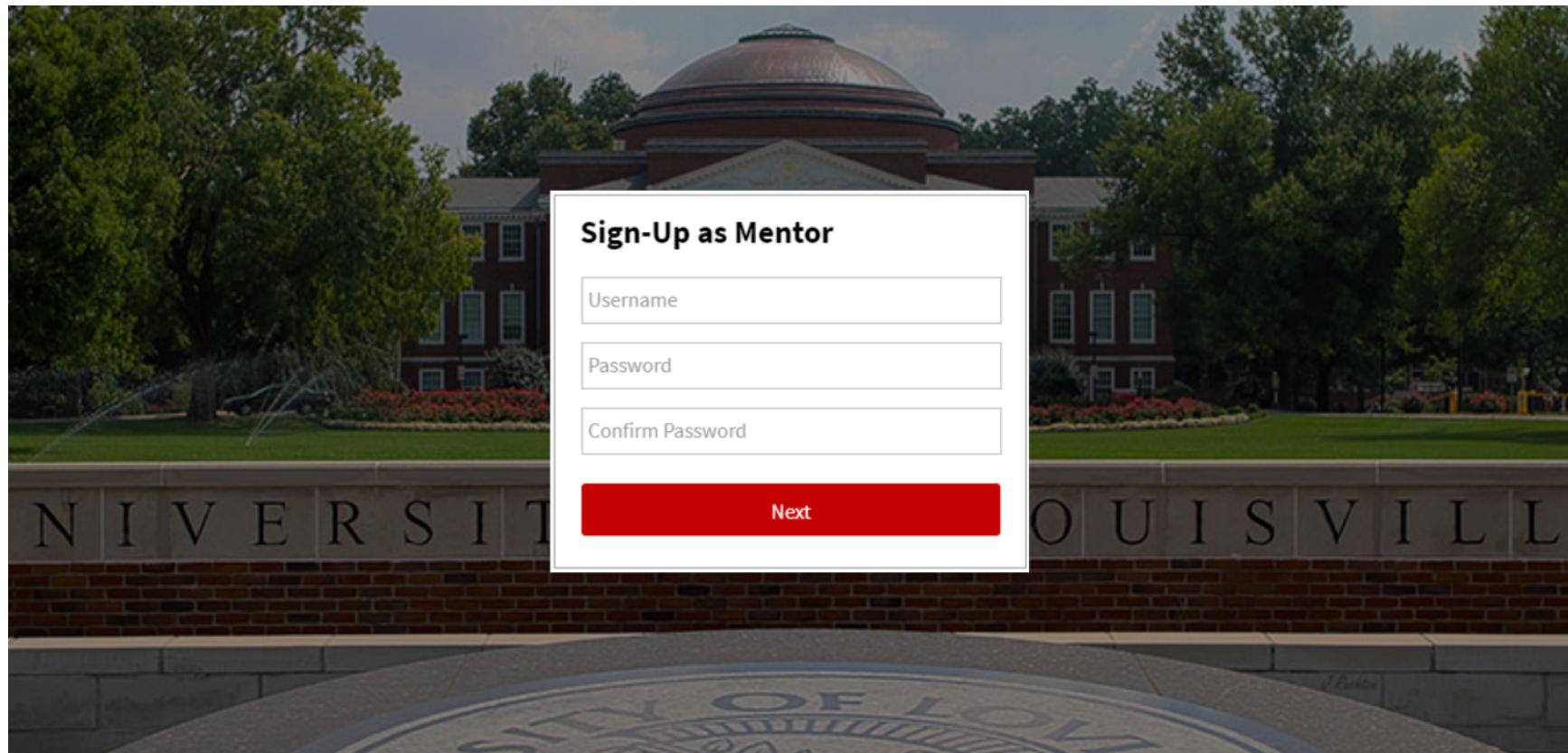
What is your concentration?

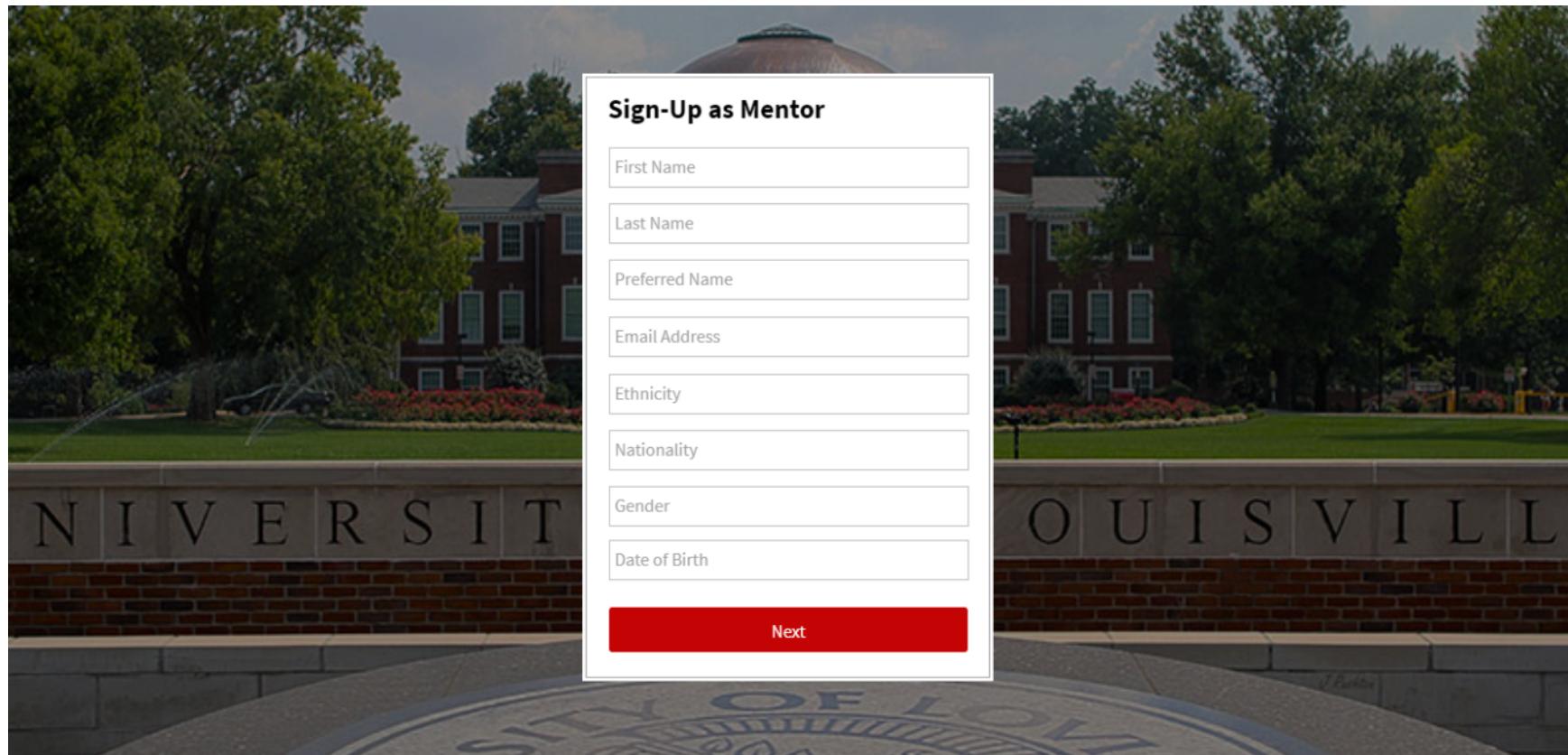
Web Development  Business Process Management  
 Information Security  Data Analytics

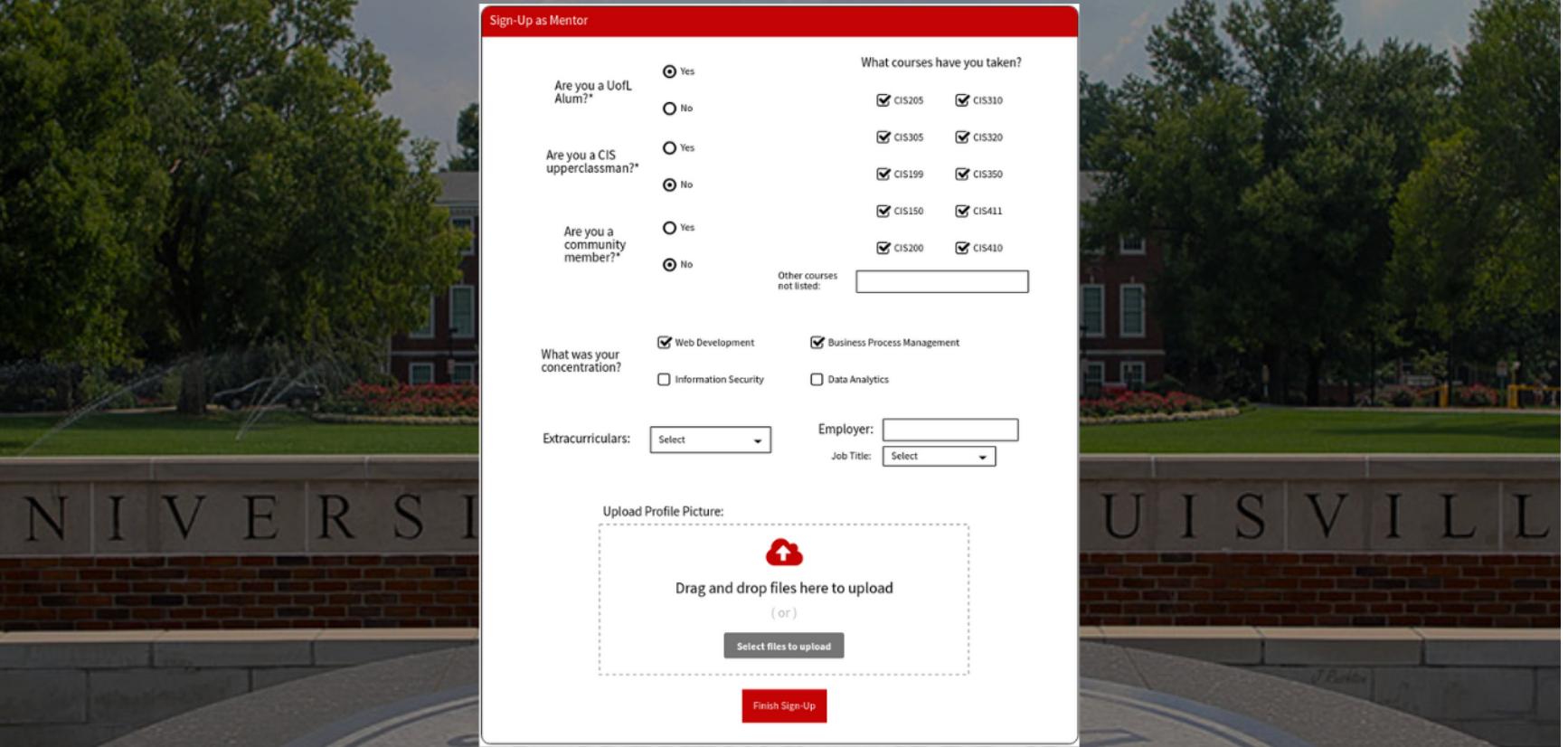
Upload Profile Picture:

Drag and drop files here to upload  
(or)  
Select files to upload

Finish Sign-Up







**Sign-Up as Mentor**

Are you a UofL Alum?\*

Yes  
 No

What courses have you taken?

CIS205  CIS310  
 CIS305  CIS320  
 CIS199  CIS350  
 CIS150  CIS411  
 CIS200  CIS410

Other courses not listed:

Are you a CIS upperclassman?\*

Yes  
 No

Are you a community member?\*

Yes  
 No

What was your concentration?

Web Development  Business Process Management  
 Information Security  Data Analytics

Extracurriculars:

Employer:   
Job Title:

Upload Profile Picture:

Drag and drop files here to upload  
(or)

Home

About Us

Donate

Jobs

Match

Events

Discuss

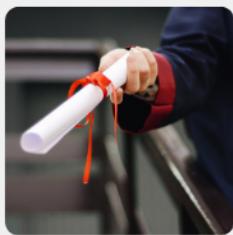
Photos

 My Account

Log Out

# Welcome to the COB Web!

We are so happy to have you here.



<

>

## Announcements

Hope everyone is having a wonderful spring break! - Dr. B

## Events

Networking Event...10/02

Business Career Fair...09/05

COB Networking Event...04/12

## Surveys

No available survey at the moment

## Report a Problem

Having issues? Report them anonymously [here](#)

Home

About Us

Donate

Jobs

Match

Events

Discuss

Photos

My Account

Log Out

## Photo Board



User Uploaded  
Photo Title   
Photo Description  
Date Created

User Uploaded  
Photo Title   
Photo Description  
Date Created

User Uploaded  
Photo Title   
Photo Description  
Date Created



Upload Photo

[Home](#)[About Us](#)[Donate](#)[Jobs](#)[Match](#)[Events](#)[Discuss](#)[Photos](#)[!\[\]\(cf5376891898fe7b4aaf81c6a8c6b974\_img.jpg\) My Account](#)[Log Out](#)

## Upload Photo



Drag and drop files here to upload

( or )

[Select files to upload](#)

[Upload Photo](#)

[Home](#)[About Us](#)[Donate](#)[Jobs](#)[Match](#)[Events](#)[Discuss](#)[Photos](#)[!\[\]\(102fd69583a64cdd4b16a60c6f63c873\_img.jpg\) My Account](#)[Log Out](#)

## Edit Photo

[Delete Photo](#)[Save Changes](#)

[Home](#)[About Us](#)[Discussion Board](#)[Job Board](#)[Donate](#)[Match](#)[Events](#)

My Account

[Log Out](#)

# Discussion Board

Let's discuss!

[CREATE A DISCUSSION](#)**Discussion Title**<Discussion Body Here>  
<Discussion Comments Here>[Add a Comment...](#)**Discussion Title**<Discussion Body Here>  
<Discussion Comments Here>[Add a Comment...](#)

Home

About Us

Donate

Jobs

Match

Events

Discuss

Photos

 My Account

Log Out

## Create a Discussion Board Post

Name

Jane Doe

Title

Tips for someone looking for an internship

Body:

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui.

 Create Post



## Comments (2)



John Doe

March 12, 2017 at 1:28pm

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui.



Jane Doe

March 12, 2017 at 1:28pm

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui.



## Create a Comment

Name

Jane Doe

Comment

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui.

[Delete Comment](#)[Create Comment](#)

Home

About Us

Donate

Jobs

Match

Events

Discuss

Photos

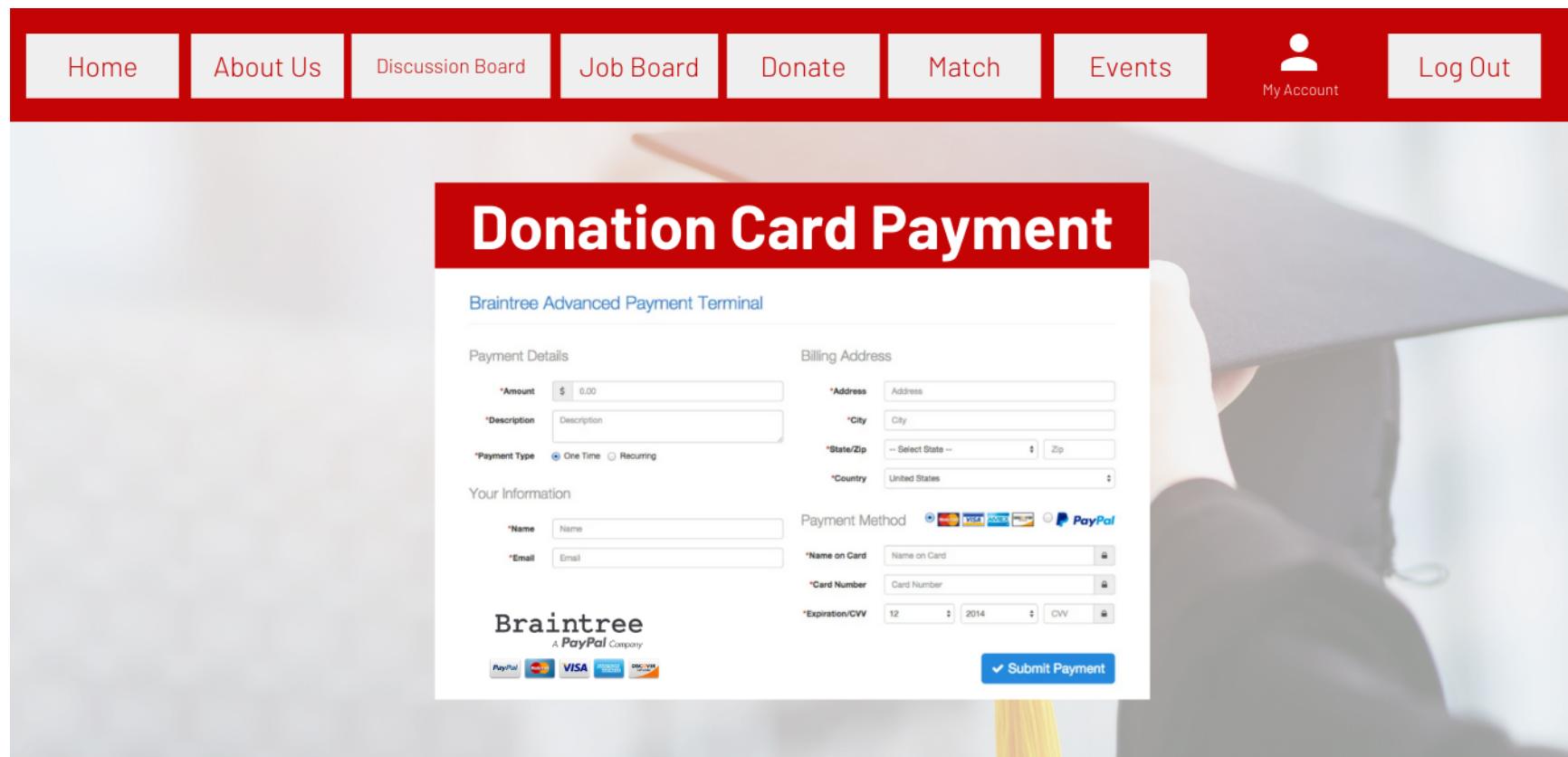
My Account

Log Out



## Donations

The University of Louisville College of Business strives to maintain the best and most up to date technology for our Computer Information System students. With the help of your kind donations, we will ensure that those needs are met. Thank you for your donation! [CLICK HERE TO DONATE](#)



The image shows a donation card payment interface. At the top, there is a red navigation bar with white text and icons. The menu items include Home, About Us, Discussion Board, Job Board, Donate, Match, Events, My Account, and Log Out. Below the navigation bar, the main content area has a large red header with the text "Donation Card Payment". Underneath the header, the Braintree Advanced Payment Terminal is displayed. The form is divided into several sections: "Payment Details" (Amount: \$ 0.00, Description: [empty], Payment Type: One Time selected), "Billing Address" (Address, City, State/Zip, Country: United States), "Your Information" (Name, Email), and "Payment Method" (radio buttons for Credit Card, Debit Card, Bank Transfer, and PayPal, with PayPal selected). The Braintree logo is visible at the bottom left, and a "Submit Payment" button is at the bottom right. The entire form is set against a background of a person's hand holding a pen over a document.

[Home](#)[About Us](#)[Discussion Board](#)[Job Board](#)[Donate](#)[Match](#)[Events](#)

My Account

[Log Out](#)

## Upcoming Events

### Event Title 1

<Event Description Here>  
<Event Date Here>

### Event Title 2

<Event Description Here>  
<Event Date Here>

### Event Title 3

<Event Description Here>  
<Event Date Here>

### Event Title 4

<Event Description Here>  
<Event Date Here>

### Event Title 5

<Event Description Here>  
<Event Date Here>

### Event Title 6

<Event Description Here>  
<Event Date Here>

Event Calendar:

June 2021						
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					



## Create an Event

Enter Event Title

Enter Event Description

Select Event Date:

June 2021						
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Create Event

Home    About Us    Donate    Jobs    Match    Events    Discuss    Photos     My Account    Log Out

## Job Board

<    >



**KFC Hiring IT Co-Op!**

KFC is now hiring IT co-ops for the fall semester. Please click [here](#) for more information about the job.

# Humana

**Humana is Hiring for Spring!**

Humana IT is hiring for the spring semester. Please click [here](#) to learn more about the roles of the positions available.



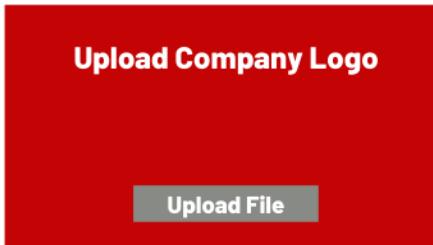
**PPL companies**

**LGE Hiring Summer Interns**

LGE is looking to hire full-time summer interns for the year 2021. Please click [here](#) for additional information.

[Home](#)[About Us](#)[Donate](#)[Jobs](#)[Match](#)[Events](#)[Discuss](#)[Photos](#) My Account[Log Out](#)

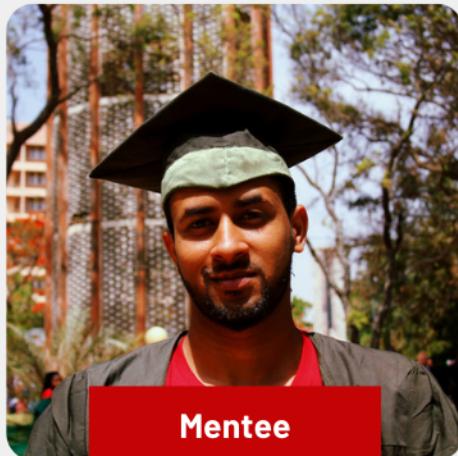
## Create Job Post

**Upload Company Logo**

[Home](#)[About Us](#)[Donate](#)[Jobs](#)[Match](#)[Events](#)[Discuss](#)[Photos](#)[!\[\]\(e550904263924d545f13e279fdd11ba4\_img.jpg\) My Account](#)[Log Out](#)

## Mentor Matching Process

Welcome to the COB Web! Please select whether you are a mentee or a mentor below:



Mentee



Mentor

Home    About Us    Discussion Board    Job Board    Donate    Match    Events     My Account    Log Out

**What are you looking for in a mentor?**

Do you prefer to be paired with an alum?  Yes  No    Gender Preference:

Do you prefer to be paired with an upperclassman?  Yes  No    Ethnicity Preference:

Do you prefer to be paired with a member of the community?  Yes  No    Nationality Preference:

Concentration Preference:  Business Process Management  Web Development  Information Security  Data Analytics    Extracurricular Preference:     Job Title Preference:

**Next**

[Home](#) [About Us](#) [Donate](#) [Jobs](#) [Match](#) [Events](#) [Discuss](#) [Photos](#) [!\[\]\(6022ba615527227c4dd1f56a68fc8cbb\_img.jpg\) My Account](#) [Log Out](#)

**Available Mentors**

Click [here](#) to view saved mentor profiles



**John Doe**  
UofL Alumnus Class of 2016  
Works at KFC Yum! as Project Manager  
[Click here to view profile](#)





**Jonathan Do**  
Member of Louisville Community  
Works at LG&E as Business Analyst  
[Click here to view profile](#)





**Jane Doe**  
Senior at UofL  
Works at Humana as Software Developer  
[Click here to view profile](#)





**Janet Dough**  
UofL Alumna Class of 2019  
Works at UPS as Cyber Intelligence Analyst  
[Click here to view profile](#)



[Next](#)

Home    About Us    Donate    Jobs    Match    Events    Discuss    Photos     My Account    Log Out

Your Saved Mentors

	<p><b>John Doe</b> UofL Alumnus Class of 2016 Works at KFC Yum! as Project Manager Click <a href="#">here</a> to view profile</p>	
	<p><b>Jane Doe</b> Senior at UofL Works at Humana as Software Developer Click <a href="#">here</a> to view profile</p>	
	<p><b>Jacob Dancer</b> Member of Louisville Community Works at UPS as Business Analyst Click <a href="#">here</a> to view profile</p>	

**Next**



## Jacob Dancer's Profile



Jacob Dancer

Member of Louisville Community  
Works at UPS as Business Analyst  
jacob.dancer@gmail.com 

I identify as a Male

Birthdate: April 20th

To learn more click [here](#)

The screenshot shows a web application interface with a red header bar containing navigation links: Home, About Us, Discussion Board, Job Board, Donate, Match, Events, My Account, and Log Out. Below the header is a large white form area with a red border. The title of the form is "Questionnaire for Mentors". The form contains ten questions, each with a label and a corresponding input field or dropdown menu:

- What are your hobbies? [Input Field]
- What is one fun fact about yourself? [Input Field]
- Where is your hometown? [Select] (dropdown menu)
- What is your preferred pronoun? [Select] (dropdown menu)
- How many languages do you speak? [Select] (dropdown menu)
- Favorite movie? [Input Field]
- Favorite show? [Input Field]
- Favorite book? [Input Field]
- Favorite color? [Select] (dropdown menu)
- Favorite music genre? [Select] (dropdown menu)
- Favorite food? [Input Field]

At the bottom of the form is a red "Next" button.

Home

About Us

Discussion Board

Job Board

Donate

Match

Events



My Account

Log Out

### Questionnaire for Mentors

#### Why do you want to become a mentor?

Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat.

Submit

Home

About Us

Donate

Jobs

Match

Events

Discuss

Photos

My Account

Log Out

## My Account



Jamie Dougherty

Sophomore at UofL  
Web Development Track  
jamie.dougherty@louisville.edu  
I identify as a female  
Looking for a co-op for fall term  
Looking for a mentor

**Edit Profile**

Home    About Us    Donate    Jobs    Match    Events    Discuss    Photos     My Account    Log Out

### Account - Edit Profile



Jamie Dougherty  
Sophomore at UofL    [Save Changes](#)

Click to edit photo

Change username:

Change password:

Change first name:

Change middle initial:

Change last name:

Change major concentration:

Change gender:

Change preferred email address:

Change class year:

Change bio:

[Delete Profile](#)    [Submit](#)

Home    About Us    Donate    Jobs    Match    Events    Discuss    Photos     My Account    Log Out

## Account - Edit Profile



Jonathan Do  
Member of Louisville Community  
Works at LG&E as Business Analyst

Save Changes

Change username:

Change password:

Change first name:

Change middle initial:

Change last name:

Change mentor status:

Change gender:

Change preferred email address:

Change bio:

Delete Profile

Submit

Home

About Us

Donate

Jobs

Match

Events

Discuss

Photos

 My Account

Log Out



## About Us

The COB Web was established in 2021. Our mission is to help CIS students connect with mentors in hopes of building lasting relationships. Our goal is for the mentors to guide students to graduation and beyond graduation.

[READ MORE](#)

Home

About Us

Donate

Jobs

Match

Events

Discuss

Photos

My Account

Log Out

## Create Backup



Backup Data



Recover Backup