**Elaboration Specification Deliverables**

**Team Cowboy Bebop**

Nora Alshimary

Ethan Grimes

Kristian Seigle

Patrick Mcintyre

Aaron Henry

**System Requirements**

System Requirements include all the activities a system will need to do and all the components the system will need to include to accomplish its goals.

**Functional Requirements**

Functional Requirements are those requirements that detail what the system needs to do.

1. The system will create student accounts.
2. The system will edit student accounts.
3. The system will delete student accounts.
4. The system will create administrator accounts.
5. The system will edit administrator accounts.
6. The system will delete administrator accounts.
7. The system will create mentor accounts.
8. The system will edit mentor accounts.
9. The system will delete mentor accounts.
10. The system will allow users to log-in.
11. The system will allow users to log-out.
12. The system will have a landing page.
13. The system will create system announcements.
14. The system will edit system announcements.
15. The system will delete system announcements.
16. The system will create surveys for Alum users.
17. The system will edit surveys for Alum users.
18. The system will create database for Alum users survey.
19. The system will allow staff to send newsletter to mailing list using the datasheet.
20. The system will allow the admin to approve accounts.
21. The system will allow the admin to edit student accounts.
22. The system will allow the admin to delete student accounts.
23. The system will allow admin to lock accounts.
24. The system will allow mentees to search for mentors using a filtering system.
25. The system will provide mentor contact information to mentee.
26. The system will create a message from the user to the Admin.
27. The system will create a discussion board thread or post.
28. The system will edit a discussion board thread or post.
29. The system will delete a discussion board thread post.
30. The system will allow users to view discussion board.
31. The system will create an event on the event calendar.
32. The system will edit an event on the event calendar.
33. The system will delete an event on the event calendar.
34. The system will provide a log in page for created users.
35. The system will provide a Frequent Asked Questions page for users.
36. The system will create a link with mentor contact information for mentees.
37. The system will take donations.
38. The system will take recurring donations.
39. The system will create a job posting.
40. The system will edit a job posting.
41. The system will delete a job posting.
42. The system will filter job postings.
43. The system will create a backup.
44. The system will have a recovery.

**Nonfunctional Requirements**

Nonfunctional Requirements detail the constraints and specifications of the system.

1. The system will be designed and implemented with Drupal and will be coded with HTML.
2. The system will be secure, and we will be using firewalls provided by Microsoft Windows.
3. The system will include a SQL server for database management.
4. The system will have a cloud storage capability, using OneDrive for a backup and recovery.
5. The system will have an online payment system through Stripe.
6. The system will have a datasheet that includes the names of those on the newsletter mailing list taken from the server to track the recipients easier.
7. The systems will be available 24 hours a day, every day of the year.
8. The system will be available for desktop and mobile.

**Use Case Diagrams**

Use Case Diagrams are used to describe the use cases from above in terms of action items that the system should do when someone, the actor, interacts with it. In the diagrams below, there is an actor on the left with lines pointing to actions the system will do while the user interacts with it. The lines represent who does what action. The use cases are set in a boundary, represented by the box.

Graphical user interface, application

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

The diagram above illustrates the relationship between the administrator actor and the system backup and restore use cases. It encompasses the create backup and system restore use cases.

Diagram

Description automatically generated

The diagram above illustrates the relationship between the system administrators, the donation feature, and the donors. It encompasses the create, modify, and delete donation use cases.

Diagram

Description automatically generated

The diagram above illustrates the relationship between the students, system administrators, the job listing feature, and the donors. It encompasses the create, modify, and delete job listing use cases.

Diagram

Description automatically generated



The trace matrix ensures the completion of the functional system requirements and use cases by matching them up and making sure each functionals system requirement matches at least one use case and vice versa. Each green block represents the intersection of a use case and a system requirement.

**Trace Matrix**

Chart

Description automatically generated

**Use Cases**

Use cases are an analysis model that presents the expected behaviors of a system by describing the interactions between external actors and the system itself. They are used by developers to get an understanding of what specific flow of steps a requirement goes through, so they can code and develop it to the specifications of the customer.

# Create Student Account

## Brief Description

The system will allow student users to create an account to use the features offered through the website.

## Flow of Events

### Basic Flow

1. The student will select the log-in link on the landing page and be redirected to the log-in page.
2. The student will then select the create student account link on the log-in page and will then be redirected to the create account page.
3. On this page, the student will provide their personal information and create a username and password.
4. The student will then select the button stating create account and the account will be created and registered on the system.
5. A profile page will then be created for the user that shows all their information.
6. The user will be returned to the landing page.
7. End of Use Case.

### Alternative Flow

*The user leaves field(s) blank.*

The system will fail to create an account and indicate which fields must be filled out on the form.

*The user inputs invalid Student ID.*

The system will fail to create an account and indicate that a valid Student ID must be used.

## Pre-Conditions

*The user must have valid Student ID.*

*The user must have valid Student Email.*

*The user must be a current UofL student.*

*The user must have internet access.*

## Post-Conditions

*The user will view the create an account form.*

*The user will input information into the create an account form.*

# Create Alumni Account

## Brief Description

The system will allow Alumni users to create an account to use the features offered through the website.

## Flow of Events

### Basic Flow

1. The Alumni will select the log-in link on the landing page and be redirected to the log-in page.
2. The Alumni will then select the create alumni account link on the log-in page and will then be redirected to the create account page.
3. On this page, the Alumni will provide their personal information and create their username and password.
4. The alumni will then select the button stating create account and the account will be created and registered on the system.
5. A profile page will then be created for the user that shows all their information.
6. The user will be returned to the landing page.
7. End of Use Case.

### Alternative Flow

*The user leaves field(s) blank.*

The system will fail to create an account and indicate which fields must be filled out on the form.

## Pre-Conditions

*The user must have valid Email.*

*The user must be a former UofL student.*

## Post-Conditions

*The user will view the create an account form.*

*The user will input information into the create an account form.*

*The user must have internet access.*

# Create Administrator Account

## Brief Description

The system will allow administrator users to create an account to use the features offered through the website.

## Flow of Events

### Basic Flow

1. The administrator will select the log-in link on the landing page and be redirected to the log-in page.
2. The administrator will then select the create administrator account link on the log-in page and will then be redirected to the administrator create account page where they will input their employee credentials.
3. On this page, the administrator will provide their personal information including username and password.
4. The administrator will then select the button stating create account and the account will be created and registered on the system.
5. A profile page will then be created for the user that shows all their information.
6. The user will be returned to the landing page.
7. End of Use Case.

### Alternative Flow

*The user leaves field(s) blank.*

The system will fail to create an account and indicate which fields must be filled out on the form.

*The user inputs invalid employee credentials.*

The system will fail to create an account and indicate that valid employee credentials must be used.

## Pre-Conditions

*The user must have valid employee credentials.*

*The user must have valid UofL Email.*

*The user must be a currently employed at UofL.*

*The user must have internet access.*

## Post-Conditions

*The user will view the create an account form.*

*The user will input information into the create an account form.*

# Edit Student Account

## Brief Description

The system will allow student users to edit an account to update their information.

## Flow of Events

### Basic Flow

1. The student will select the profile link on the landing page and be redirected to their profile page.
2. The student will then select edit profile button.
3. The student will update their information by editing the fields.
4. The student will then select the button stating save all changes.
5. The student will then return to the profile page.
6. End of Use Case.

### Alternative Flow

*The user leaves field(s) blank.*

The system will fail to update the account and indicate which fields must be filled out on the form.

## Pre-Conditions

*The user must have an account.*

## Post-Conditions

*The user will view the profile page.*

*The user will input information into the profile page.*

# Edit Alumni Account

## Brief Description

The system will allow alumni users to edit an account to update their information.

## Flow of Events

### Basic Flow

1. The alumni will select the profile link on the landing page and be redirected to their profile page.
2. The alumni will then select edit profile button.
3. The alumni will update their information by editing the fields.
4. The alumni will then select the button stating save all changes.
5. The alumni will then return to the profile page.
6. End of Use Case.

### Alternative Flow

*The user leaves field(s) blank.*

The system will fail to update the account and indicate which fields must be filled out on the form.

## Pre-Conditions

*The user must have an account.*

## Post-Conditions

*The user will view the profile page.*

*The user will input information into the profile page.*

# Edit Administrator Account

## Brief Description

The system will allow administrator users to edit an account to update their information.

## Flow of Events

### Basic Flow

1. The administrator will select the profile link on the landing page and be redirected to their profile page.
2. The administrator will then select edit profile button.
3. The administrator will update their information by editing the fields.
4. The administrator will then select the button stating save all changes.
5. The administrator will then return to the profile page.
6. End of Use Case.

### Alternative Flow

*The user leaves field(s) blank.*

The system will fail to update the account and indicate which fields must be filled out on the form.

## Pre-Conditions

*The user must have an account.*

## Post-Conditions

*The user will view the profile page.*

*The user will input information into the profile page.*

# Delete Student Account

## Brief Description

The system will allow student users to delete their account.

## Flow of Events

### Basic Flow

1. The student will select the profile link on the landing page and be redirected to their profile page.
2. The student will then select delete student account button.
3. A pop-up will ask if the student is sure that they want the account deleted.
4. The student will select yes or no depending on their choice.
5. The account will be deleted.
6. The student will then return to the landing page.
7. End of Use Case.

### Alternative Flow

## Pre-Conditions

*The user must have an account.*

## Post-Conditions

*The user account will be deleted.*

*The user’s credentials will no longer be accepted.*

*The user will no longer receive messages from the system.*

# Delete Alumni Account

## Brief Description

The system will allow alumni users to delete their account.

## Flow of Events

### Basic Flow

1. The alumni will select the profile link on the landing page and be redirected to their profile page.
2. The alumni will then select delete student account button.
3. A pop-up will ask if the alumni is sure that they want the account deleted.
4. The alumni will select yes or no depending on their choice.
5. The account will be deleted.
6. The alumni will then return to the landing page.
7. End of Use Case.

### Alternative Flow

## Pre-Conditions

*The user must have an account.*

## Post-Conditions

*The user account will be deleted.*

*The user’s credentials will no longer be accepted.*

*The user will no longer receive messages from the system.*

# Delete Administrator Account

## Brief Description

The system will allow administrator users to delete their account.

## Flow of Events

### Basic Flow

1. The administrator will select the profile link on the landing page and be redirected to their profile page.
2. The administrator will then select delete student account button.
3. A pop-up will ask if the administrator is sure that they want the account deleted.
4. The administrator will select yes or no depending on their choice.
5. The account will be deleted.
6. The administrator will then return to the landing page.
7. End of Use Case.

### Alternative Flow

## Pre-Conditions

*The user must have an account.*

## Post-Conditions

*The user account will be deleted.*

*The user will no longer receive messages from the system.*

*The user’s credentials will no longer be accepted.*

Use Case Specification: Create A Discussion Board Post

# Create A Discussion Board Post

## Brief Description

This use case describes how a user will create a discussion board post by using the website. The user will navigate to the discussion area of the website and click the create discussion button to start the process. At that time, they will be prompted to enter a title and message for the new discussion board.

# Flow of Events

## Basic Flow

### The user submits a title.

### The user submits a post message.

### The discussion board title is checked to make sure its not blank.

### The post message is checked to make sure its not blank.

### The post is stored in the database.

## Alternative Flows

### Users leaves the post title blank

### The system will return an error message to the user asking for a corrected input.

### Users leaves the post message blank

### The system will return an error message to the user asking for a corrected input.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Active internet connection

## User must be logged in

# Post-conditions

There are no post-conditions associated with this use case.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: Edit A Discussion Board Post

# Edit A Discussion Board Post

## Brief Description

This use case describes how a user will edit a discussion board post by using the website. The user will navigate to the discussion area of the website and click on a discussion then click on the edit button to start the process. At that time, they will be prompted to edit the title and message for the post.

# Flow of Events

## Basic Flow

### The system fetches the original post.

### They system displays the original post in an editable form.

### The user submits an updated post title.

### The user submits an updated post message.

### The discussion board title is checked to make sure its not blank.

### The post message is checked to make sure its not blank.

### The post is updated in the database.

## Alternative Flows

### Users leaves the post title blank

### The system will return an error message to the user asking for a corrected input.

### Users leaves the post message blank

### The system will return an error message to the user asking for a corrected input.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Active internet connection

## User must be logged in

## User must be the author of the original post or an admin

# Post-conditions

There are no post-conditions associated with this use case.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: Delete A Discussion Board Post

# Delete A Discussion Board Post

## Brief Description

This use case describes how a user will delete a discussion board post by using the website. The user will navigate to the discussion area of the website and click on the post they want to remove and click on delete post button to start the process. At that time, they will be prompted to confirm their actions.

# Flow of Events

## Basic Flow

### The user starts the deletion process by click on the delete button.

### The system prompts the user to confirm their actions.

### The user confirms their actions.

### The post is dropped from the database.

## Alternative Flows

### There are no alternative flows associated with this use case.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Active internet connection

## User must be logged in

## User must be the author of the original post or an admin

# Post-conditions

There are no post-conditions associated with this use case.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: View Discussion Board

# View Discussion Board

## Brief Description

This use case describes how a user can view a discussion board on the website. The user will navigate to the discussion area of the website and be shown a list of discussion board to view. They can then click on a specific discussion board to view it.

# Flow of Events

## Basic Flow

### The system fetches a list of all discussion boards from the database.

### The users is shown the list of discussion board titles.

### The user clicks on a title to view the discussion board.

### The system fetches the discussion board from the database.

### The system displays the discussion board to the user.

## Alternative Flows

### There are no alternative flows associated with this use case.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Active internet connection

## User must be logged in

# Post-conditions

There are no post-conditions associated with this use case.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: Create An Event On The Calendar

# Create An Event On The Calendar

## Brief Description

This use case describes how a user will create an event on the calendar. The user will navigate to the calendar portion of the website and click the add event button to start the process. At that time, they will be prompted to enter a title, description, start date, start time, end date, and end time for the new event.

# Flow of Events

## Basic Flow

### The user clicks on the create event button.

### The system displays an event creation form.

### The user submits an event title.

### The user submits an event description.

### The user submits an event start date.

### The user submits an event start time.

### The user submits an event end date.

### The user submits an event end time.

### The event title is checked to make sure its not blank.

### The event description is checked to make sure its not blank.

### The event start date is checked to make sure its not blank.

### The event start time is checked to make sure its not blank.

### The event end date is checked to make sure its not blank.

### The event end time is checked to make sure its not blank.

### The event is stored in the database.

## Alternative Flows

### Users leaves the event title blank

### The system will return an error message to the user asking for a corrected input.

### Users leaves the event description blank

### The system will return an error message to the user asking for a corrected input.

### Users leaves the event start date blank or invalid.

### The system will return an error message to the user asking for a corrected input.

### Users leaves the event start time blank or invalid.

### The system will return an error message to the user asking for a corrected input.

### Users leaves the event end date blank or invalid.

### The system will return an error message to the user asking for a corrected input.

### Users leaves the event end time blank or invalid.

### The system will return an error message to the user asking for a corrected input.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Active internet connection

## User must be logged in

## User must be the author of the original event or an admin

# Post-conditions

There are no post-conditions associated with this use case.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: Edit An Event On The Calendar

# Edit An Event On The Calendar

## Brief Description

This use case describes how a user will edit an event on the calendar. The user will navigate to the calendar portion of the website and click on an event. At that time, an edit button will appear. If the user clicks on the edit button, they will be prompted to enter an updated title, description, start date, start time, end date, and end time for the new event.

# Flow of Events

## Basic Flow

### The user clicks on the update event button.

### The system displays an event creation form.

### The user submits an event title.

### The user submits an event description.

### The user submits an event start date.

### The user submits an event start time.

### The user submits an event end date.

### The user submits an event end time.

### The event title is checked to make sure its not blank.

### The event description is checked to make sure its not blank.

### The event start date is checked to make sure its not blank.

### The event start time is checked to make sure its not blank.

### The event end date is checked to make sure its not blank.

### The event end time is checked to make sure its not blank.

### The event is updated in the database.

## Alternative Flows

### Users leaves the event title blank

### The system will return an error message to the user asking for a corrected input.

### Users leaves the event description blank

### The system will return an error message to the user asking for a corrected input.

### Users leaves the event start date blank or invalid.

### The system will return an error message to the user asking for a corrected input.

### Users leaves the event start time blank or invalid.

### The system will return an error message to the user asking for a corrected input.

### Users leaves the event end date blank or invalid.

### The system will return an error message to the user asking for a corrected input.

### Users leaves the event end time blank or invalid.

### The system will return an error message to the user asking for a corrected input.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Active internet connection

## User must be logged in

# Post-conditions

There are no post-conditions associated with this use case.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: Delete An Event On The Calendar

# Delete An Event On The Calendar

## Brief Description

This use case describes how a user will delete an event on the calendar. The user will navigate to the calendar portion of the website and click on an event. At that time, a delete button will appear. If the user clicks on the delete button, they will be prompted to confirm their actions.

# Flow of Events

## Basic Flow

### The user starts the deletion process by click on the delete button.

### The system prompts the user to confirm their actions.

### The user confirms their actions.

### The event is dropped from the database.

## Alternative Flows

### There are no alternative flows associated with this use case.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Active internet connection

## User must be logged in

## User must be the author of the original event or an admin

# Post-conditions

There are no post-conditions associated with this use case.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: Search Frequent Asked Questions

# Search Frequent Asked Questions

## Brief Description

This use case describes how a user will search a FAQ database on the website. The user will navigate to the FAQ page the website and enter a search term and click the search button. At that time, a list of search results will appear that the user can click on to read the related article.

# Flow of Events

## Basic Flow

### The users submit a keyword.

### The submission is checked to make sure its not blank.

### The system fetches all FAQ articles that have that keyword in them.

### The system displays the results to the user.

### The user clicks on an article to open it.

### The system fetches the article.

### The system displays the article.

## Alternative Flows

### Users leaves the keyword blank

### The system will return an error message to the user asking for a corrected input.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Active internet connection

# Post-conditions

There are no post-conditions associated with this use case.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: User Login

# User Login

## Brief Description

This use case describes how a user will log into the website. They will navigate to or be redirected to the login page to start the process. At that time, they will be prompted to enter an email address and password for authentication.

# Flow of Events

## Basic Flow

### The system will prompt the user to provide their credentials.

### The user submits their email address.

### The user submits their password.

### The email address is checked to make sure its not blank.

### The password is checked to make sure its not blank.

### The system checks if the email address exists in the database.

### The system checks if the password is correct.

### The system creates the user’s session.

## Alternative Flows

### Users leaves the email address blank

### The system will return an error message to the user asking for a corrected input.

### Users leaves the password blank

### The system will return an error message to the user asking for a corrected input.

### Users does not exist in the database

### The system will return an error message to the user asking for a corrected input.

### Users enters an invalid password

### The system will return an error message to the user asking for a corrected input.

### The system will record the failed login attempt.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Active internet connection

# Post-conditions

There are no post-conditions associated with this use case.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: Backup Mentoring System

# Backup Mentoring System

## Brief Description

This use case describes how an admin will backup the website and database. The admin will navigate to Drupal administration area of the website and click on the backup and migrate module and then click on backup. At that time, they will be prompted to select what they want to backup and where to back it up to.

# Flow of Events

## Basic Flow

### The admin submits which database to be backed up.

### The admin submits which files are to be backed up.

### The admin submits which backup provider to use.

### The system makes sure a database has been selected.

### The system makes sure files have been selected.

### The system makes sure the backup provider is working.

### The system uploads the data to the backup provider.

## Alternative Flows

### The admin dose not select a database

### The system will return an error message to the admin asking for a corrected input.

### The admin dose not select any files

### The system will return an error message to the admin asking for a corrected input.

### The admin dose not select a backup provider

### The system will return an error message to the admin asking for a corrected input.

### The backup provider is offline

### The system will return an error message to the admin.

### The backup fails to upload

### The system will email the admin.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Active internet connection

## Must be an admin

# Post-conditions

There are no post-conditions associated with this use case.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: Restore Mentoring System

# Restore Mentoring System

## Brief Description

This use case describes how an admin will restore the website and database. The admin will navigate to Drupal administration area of the website and click on the backup and migrate module and then click on restore. At that time, they will be prompted to select what they want to restore and where to restore it to.

# Flow of Events

## Basic Flow

### The admin submits which backup provider to use.

### The system makes sure the backup provider is working.

### The system retrieves a list of backups.

### The admin submits which database to be restored.

### The admin submits which files are to be restored.

### The system makes sure a database has been selected.

### The system makes sure files have been selected.

### The system downloads the data from the backup provider.

## Alternative Flows

### The admin dose not select a database

### The system will return an error message to the admin asking for a corrected input.

### The admin dose not select any files

### The system will return an error message to the admin asking for a corrected input.

### The admin dose not select a backup provider

### The system will return an error message to the admin asking for a corrected input.

### The backup provider is offline

### The system will return an error message to the admin.

### The backup fails to download

### The system will email the admin.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Active internet connection

## Must be an admin

# Post-conditions

There are no post-conditions associated with this use case.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: Log-in to account

# Use-Case Name: Log-in to account

## Brief Description

The user will access the system and enter the personal credentials they have created into the specific log-in form. This is the only way a user can gain access to the functions and viewing of the website.

# Flow of Events

## Basic Flow

1. The user accesses the website.
2. The user enters their personal credentials into the log-in form.
3. The system authenticates the user’s information.
4. The user gains access to the system and its functions.
5. The use case ends.

## Alternative Flows

### The user enters incorrect information into the log-in form

### The user is already currently logged into a session.

# Pre-conditions

1. The user already has an existing account.
2. The user is not currently logged in.

# Post-conditions

1. The user is authenticated gains access to the functions of the website.

Use Case Specification: Log-out of account

# Use-Case Name: Log-out of account

## Brief Description

The user will navigate to the log-out function of the website and end the current session. This will remove the user’s access to the functions of the system.

# Flow of Events

## Basic Flow

1. The user has access to the website.
2. The user is finished accessing the system and its functions.
3. The user clicks the log-out button.
4. The user’s current session is terminated.
5. The user is redirected to the default log-in screen.
6. The use case ends.

## Alternative Flows

### The user is currently logged into the system.

### The user wishes to end their session.

# Pre-conditions

1. The user already has an existing account.
2. The user is not currently logged in.

# Post-conditions

1. The user is no longer logged into the system.

Use Case Specification: View Landing Page

# Use-Case Name: View Landing Page

## Brief Description

The user will access the system and have access to viewing the landing page of the website. The user will be brought to the landing page after logging in or can navigate to the landing page through any other page in the system.

# Flow of Events

## Basic Flow

1. The user accesses the system.
2. The user logs into their account after being prompted to do so.
3. The system authenticates the users information.
4. The user gains access to the system and is directed to the landing page.
5. The user views the landing page and all the links to the rest of the system functions.
6. The use case ends.

## Alternative Flows

### The user is already logged in and is using a different function on the website

The user navigates to the “Landing Page” link on the webpage

The user is redirected to the landing page from their current webpage on the website.

# Pre-conditions

1. The user is logged into their account.

# Post-conditions

1. The user is viewing the landing page.

Use Case Specification: Create System Announcement

# Use-Case Name: Create System Announcement

## Brief Description

The admin will gain access to the system and navigate to the admin-only functions in the system. The admin will choose to access the “Announcements” button. The admin will choose to create a new announcement to be viewed through email, on the landing page, or on the dedicated announcement page. The announcements will give system-wide updates on the website and other relevant information.

# Flow of Events

## Basic Flow

1. The admin accesses their account in the system.
2. The admin navigates to the admin-only functions of the system.
3. The admin clicks on the “Announcements” tab in the list of admin functions.
4. The admin views the possible functions in the specific tab.
5. The admin clicks on “create new announcement” button.
6. The admin chooses where the announcement will be sent.
7. The admin creates their announcement using the blank field provided.
8. The admin clicks the “submit” button when they are finished creating their content.
9. The admin successfully submits their announcement post and is prompted with a message describing the post as successful.
10. The use case ends.

## Alternative Flows

### The admin receives a message describing the post as unsuccessful and is prompted to check their connection.

### The admin does not have the necessary security permissions to access the announcements function.

### The admin session has timed out and requires a successful log-in.

# Pre-conditions

1. The admin has necessary security functions.
2. The admin is currently logged into their account.

# Post-conditions

1. The admin has successfully submitted an announcement.

Use Case Specification: Edit System Announcement

# Use-Case Name: Edit System Announcement

## Brief Description

The admin will gain access to the system and navigate to the admin-only functions in the system. The admin will choose to access the “Announcements” button. The admin will choose to edit an existing announcement to be viewed through email, on the landing page, or on the dedicated announcement page. The announcements will give system-wide updates and other relevant information.

# Flow of Events

## Basic Flow

1. The admin accesses their account in the system.
2. The admin navigates to the admin-only functions of the system.
3. The admin clicks on the “Announcements” tab in the list of admin functions.
4. The admin views the possible functions in the specific tab.
5. The admin clicks the “edit” button under a currently existing announcement.
6. The admin chooses who will be able to view the updated announcement.
7. The admin edits their content of the announcement.
8. The admin clicks the “submit” button when they are finished creating their content.
9. The admin successfully submits their announcement post and is prompted with a message describing the post as successfully edited.
10. The use case ends.

## Alternative Flows

### The admin receives a message describing the post as unsuccessful and is prompted to check their connection.

### The admin does not have the necessary security permissions to edit the announcements function.

### The admin session has timed out and requires a successful log-in.

# Pre-conditions

1. The admin has necessary security functions.
2. The admin is currently logged into their account.

# Post-conditions

1. The admin has successfully edited an announcement.

Use Case Specification: Delete System Announcement

# Use-Case Name: Delete System Announcement

## Brief Description

The admin will gain access to the system and navigate to the admin-only functions in the system. The admin will choose to access the “Announcements” button. The admin will choose to delete an existing announcement from viewing from specific locations such as the landing page or the dedicated Announcements page.

# Flow of Events

## Basic Flow

1. The admin accesses their account in the system.
2. The admin navigates to the admin-only functions of the system.
3. The admin clicks on the “Announcements” tab in the list of admin functions.
4. The admin views the possible functions in the specific tab.
5. The admin clicks on “delete” button under an existing announcement that is visible.
6. The admin chooses from where the announcement will be deleted.
7. The admin clicks the “Delete” button when they are finished choosing their preferences.
8. The admin successfully deletes their announcement post and is prompted with a message describing the deletion as successful.
9. The use case ends.

## Alternative Flows

### The admin receives a message describing the post as unsuccessful and is prompted to check their connection.

### The admin does not have the necessary security permissions to access the announcements function.

### The admin session has timed out and requires a successful log-in.

# Pre-conditions

1. The admin has necessary security functions.
2. The admin is currently logged into their account.

# Post-conditions

1. The admin has successfully removed the announcement.

Use Case Specification: Create Alumni Survey

# Use-Case Name: Create Alumni Survey

## Brief Description

The admin will gain access to the system and navigate to the admin-only functions of the website. The admin chooses the “Surveys” button and is given a list of functions pertaining to the surveys given to the users of the website. The admin will choose the section of the surveys related only to the alumni and will create a new survey for the alumni to answer giving us a source of information that is related only to the alumni.

# Flow of Events

## Basic Flow

1. The admin accesses their account in the system.
2. The admin navigates to the admin-only functions of the system.
3. The admin clicks on the “Surveys” tab in the list of admin functions.
4. The admin views the possible functions in the specific tab.
5. The admin clicks on “create new survey” button.
6. The admin chooses who the survey will be available to.
7. The admin chooses what type of responses will be available to be answered.
8. The admin will create questions for the alumni to answer within the system.
9. The admin will click the “submit survey” button when completed.
10. The admin will receive a message saying the survey has been successfully submitted.
11. The use case ends.

## Alternative Flows

### The admin receives a message describing the post as unsuccessful and is prompted to check their connection.

### The admin does not have the necessary security permissions to access the announcements function.

### The admin session has timed out and requires a successful log-in.

# Pre-conditions

1. The admin has necessary security functions.
2. The admin is currently logged into their account.

# Post-conditions

1. The admin has successfully submitted a survey for alumni.

Use Case Specification: Edit Alumni Survey

# Use-Case Name: Edit Alumni Survey

## Brief Description

The admin will gain access to the system and navigate to the admin-only functions of the website. The admin chooses the “Surveys” button and is given a list of functions pertaining to the surveys given to the users of the website. The admin will choose the section of the surveys related only to the alumni and will edit an existing survey for the alumni to answer giving us a source of information that is related only to the alumni.

# Flow of Events

## Basic Flow

1. The admin accesses their account in the system.
2. The admin navigates to the admin-only functions of the system.
3. The admin clicks on the “Surveys” tab in the list of admin functions.
4. The admin views the possible functions in the specific tab.
5. The admin clicks on “edit existing survey” button.
6. The admin chooses who the survey will be available to and changes if necessary.
7. The admin chooses what type of responses will be available to be answered.
8. The admin will edit questions for the alumni to answer within the system.
9. The admin will click the “submit survey” button when completed.
10. The admin will receive a message saying the survey has been successfully edited.
11. The use case ends.

## Alternative Flows

### The admin receives a message describing the post as unsuccessful and is prompted to check their connection.

### The admin does not have the necessary security permissions to access the announcements function.

### The admin session has timed out and requires a successful log-in.

# Pre-conditions

1. The admin has necessary security functions.
2. The admin is currently logged into their account.

# Post-conditions

1. The admin has successfully submitted a survey for alumni.

Use Case Specification: Create Alumni Survey Database

# Use-Case Name: Create Alumni Survey Database

## Brief Description

The admin will gain access of the system and will utilize the admin-only functions of the website. From the list of admin-only functions, the admin will choose the “surveys” button and will navigate to the database function within the surveys tab. The admin will create a database for a specific survey to retain all the information provided to use as data.

# Flow of Events

## Basic Flow

1. The admin accesses their account in the system.
2. The admin navigates to the admin-only functions of the system.
3. The admin clicks on the “Surveys” tab in the list of admin functions.
4. The admin views the possible functions in the specific tab.
5. The admin clicks on “database” button and is redirected to another page dedicated to the related survey databases.
6. The admin clicks “create new database” button.
7. The admin links a specific currently available survey or one that is going to be created to the new database.
8. The admin will choose where the data will be sent and stored.
9. The admin will click the “save database” button when completed.
10. The admin will receive a message saying the database has been successfully created.
11. The use case ends.

## Alternative Flows

### The admin receives a message describing the database as unsuccessful and is prompted to check their connection.

### The admin does not have the necessary security permissions to access the database function.

### The admin session has timed out and requires a successful log-in.

# Pre-conditions

1. The admin has necessary security functions.
2. The admin is currently logged into their account.

# Post-conditions

1. The admin has successfully submitted a survey for alumni.

Use Case Specification: Approve Account

# Approve Account

## Brief Description

Admin will have access to a page where they are able to approve account requests from all users, mentors and mentees.

# Flow of Events

## Basic Flow

1. Admin chooses “Processes” button
2. Admin chooses “Account Requests” button
3. Admin selects applicable account request
4. Admin reviews the account request
5. The admin chooses “approved” or “denied” button for the account request

## Alternative Flows

### < First Alternative Flow >

1. Log-In information incorrect

# Pre-conditions

1. Admin must already be a registered user and registered admin
2. There must be account requests submitted for approval

## < Pre-condition One >

# Post-conditions

* Users account is now approved and ready to be interacted with on the platform

Use Case Specification: Edit Account Admin

# Edit Account Admin

## Brief Description

Admin will have access to a page where they are able to edit user accounts. Both users and admins are able to complete this step.

# Flow of Events

## Basic Flow

* Admin chooses “Account Search” button
* Admin has option to input search by Name
* Admin has option to filter user by graduation year
* Admin has option to filter user by academic standing
* Admin has option to filter user by hometown
* Admin has option to filter user by current area of residence
* Admin has option to filter user by fraternity/sorority affiliations
* Admin has option to filter user by current field of employment
* Admin has option to filter user by CIS major concentration
* Admin has option to filter user by Gender
* Admin has option to filter user by Ethnicity
* Admin has option to search interests using keywords
* Admin selects “search” button
* Admin selects desired Account from search output
* Admin selects “edit account”
* Admin has option to edit Name
* Admin has option to edit graduation year
* Admin has option to edit current area of residence
* Admin has option to edit fraternity/sorority affiliations
* Admin has option to edit current field of employment
* Admin has option to edit CIS major concentration
* Admin has option to edit Gender
* Admin has option to edit Ethnicity
* Admin chooses save button

## Alternative Flows

### < First Alternative Flow >

* Log-In information incorrect

# Pre-conditions

* Admin must already be a registered user and registered admin
* Admin must already be logged in
* There must be an account to be edited

## < Pre-condition One >

# Post-conditions

* Changes are now saved to the user’s account

Use Case Specification: Delete Account Admin

# Delete Account Admin

## Brief Description

Admin will have access to a page where they are able to edit and delete a users accounts. Both users and admins are able to complete this step.

# Flow of Events

## Basic Flow

* Admin chooses “Processes”
* Admin chooses “Accounts”
* Admin has option to input search by Name
* Admin has option to filter user by graduation year
* Admin has option to filter user by academic standing
* Admin has option to filter user by hometown
* Admin has option to filter user by current area of residence
* Admin has option to filter user by fraternity/sorority affiliations
* Admin has option to filter user by current field of employment
* Admin has option to filter user by CIS major concentration
* Admin has option to filter user by Gender
* Admin has option to filter user by Ethnicity
* Admin has option to search interests using keywords
* Admin selects “search” button
* Admin selects desired Account from search output
* Admin chooses “Delete Account” button
* Admin chooses “Yes” when prompted “Are you sure?”
* Admin chooses save button

## Alternative Flows

### < First Alternative Flow >

* Admin enters incorrect values in the filter service

# Pre-conditions

* Admin must already be a registered user and registered admin
* Admin must already be logged in
* There must be an already approved account to be deleted

## < Pre-condition One >

# Post-conditions

* Users account is now deleted from the filtering outputs and online network

Use Case Specification: Lock Account Temporarily

# Lock Account Temporarily

## Brief Description

Admin will have access to a page where they are able to manually lock a user’s account. The admin will be able to choose how many days the account is locked.

# Flow of Events

## Basic Flow

* Admin chooses “Processes” button
* Admin chooses “Accounts” button
* Admin has option to input search by Name
* Admin has option to filter user by graduation year
* Admin has option to filter user by academic standing
* Admin has option to filter user by hometown
* Admin has option to filter user by current area of residence
* Admin has option to filter user by fraternity/sorority affiliations
* Admin has option to filter user by current field of employment
* Admin has option to filter user by CIS major concentration
* Admin has option to filter user by Gender
* Admin has option to filter user by Ethnicity
* Admin has option to search interests using keywords
* Admin selects “search” button
* Admin selects desired Account from search output
* Admin selects “Lock Account” button
* Admin selects “Yes” when prompted “Are you sure?”
* The website presents account lock options including “Timed Lock-Out”
* Admin inputs number of days account is to be locked
* Admin selects “ok”
* Admin chooses save button

## Alternative Flows

### < First Alternative Flow >

* Admin enters incorrect values in the filter service
* Admin enters unaccepted values into times lock-out form

# Pre-conditions

* Admin must already be a registered user and registered admin
* Admin must already be logged in
* There must be an already approved account to be locked

## < Pre-condition One >

# Post-conditions

* Users account is now temporarily locked, and they are unable to log in
* When users with locked accounts attempt to log in, log-in page will produce “Account Currently Locked” error
* After account locked time period is up, user is automatically able to log back into account

Use Case Specification: Lock Account Indefinitely

# Lock Account Indefinitely

## Brief Description

Admin will have access to a page where they are able to manually lock a user’s account. The admin locks the account until situation would require otherwise.

# Flow of Events

## Basic Flow

* Admin clicks the “Processes” in top ribbon of homepage
* Admin clicks “Accounts” in left hand ribbon of page
* Admin has option to input search by Name
* Admin has option to filter user by graduation year
* Admin has option to filter user by academic standing
* Admin has option to filter user by hometown
* Admin has option to filter user by current area of residence
* Admin has option to filter user by fraternity/sorority affiliations
* Admin has option to filter user by current field of employment
* Admin has option to filter user by CIS major concentration
* Admin has option to filter user by Gender
* Admin has option to filter user by Ethnicity
* Admin has option to search interests using keywords
* Admin selects “search” button
* Admin selects desired Account from search output
* Admin selects “Lock Account” button
* Admin selects “Yes” when prompted “Are you sure?”
* The website presents account lock options including “Indefinite Lock-Out”
* Admin selects “ok”
* Admin chooses save button

## Alternative Flows

### < First Alternative Flow >

* Admin enters incorrect values in the filter service
* Admin enters unaccepted values into times lock-out form

# Pre-conditions

* Admin must already be a registered user and registered admin
* Admin must already be logged in
* There must be an already approved account to be locked

## < Pre-condition One >

# Post-conditions

* Users account is now temporarily locked, and they are unable to log in
* When users with locked accounts attempt to log in, log-in page will produce “Account Currently Locked” error

Use Case Specification: Search Mentor Accounts

# Search Mentor Accounts

## Brief Description

Mentees will have access to a page that allows them to search through the registered mentors on the platform through both, specific search fields and filtering systems.

# Flow of Events

## Basic Flow

* User selects “Account Search” button
* User selects “Mentor Search” radio button
* User has option to search specifically by Name
* User has option to filter mentors by graduation year
* User has option to filter mentors by current area of residence
* User has option to filter mentors by fraternity/sorority affiliations
* User has option to filter mentors by current field of employment
* User has option to filter mentors by CIS major concentration
* User has option to filter mentors by Gender
* User has option to filter mentors by Ethnicity
* User has option to search interests using keywords
* User selects “search” button

## Alternative Flows

### < First Alternative Flow >

* User enters unaccepted values into search criteria

# Pre-conditions

* User must already be a registered user
* User must already be logged in

## < Pre-condition One >

# Post-conditions

* User is provided a filtered list with the mentors that match their search criteria

Use Case Specification: Contact Mentor Request

# Contact Mentor Request

## Brief Description

After finding a mentor of interest, mentees will be able to request contact information from the mentors.

# Flow of Events

## Basic Flow

* Mentee selects “Account Search” button
* Mentee selects “Mentor Search” radio button
* Mentee has option to search specifically by Name
* Mentee has option to filter mentors by graduation year
* Mentee has option to filter mentors by current area of residence
* Mentee has option to filter mentors by fraternity/sorority affiliations
* Mentee has option to filter mentors by current field of employment
* Mentee has option to filter mentors by CIS major concentration
* Mentee has option to filter mentors by Gender
* Mentee has option to filter mentors by Ethnicity
* Mentee has option to search interests using keywords
* Mentee selects “search” button
* Mentee selects desired Account from search output
* Mentee selects “Request Contact Information” button
* Mentor is sent notification of contact info request
* Mentor selects notification
* Mentor has options of “approve” or “deny” buttons for the contact request

## Alternative Flows

### < First Alternative Flow >

* User enters unaccepted values into search criteria
* Mentor denies the contact request

# Pre-conditions

* User must already be a registered user
* User must already be logged in
* User must have searched and found the profile for the mentor they are interested in contacting
* Mentor must have contact information input into the system and updated regularly

## < Pre-condition One >

# Post-conditions

* After Mentor approves the contact request, mentee is sent a notification saying

Use Case Specification: Contact Admin

# Contact Admin

## Brief Description

All users can send system Admin(s) questions regarding both the platform and general inquiries about the University of Louisville CIS department.

# Flow of Events

## Basic Flow

* User selects “Contact Admin” button
* User is brought to the “Contact Admin” page with form
* User inputs name into contact form
* User inputs message subject into contact form
* User inputs main body into contact form
* User chooses “Submit” button

## Alternative Flows

### < First Alternative Flow >

* Form puts out error after “submit” clicked if all three fields not filled out

# Pre-conditions

* User must already be a registered user
* User must already be logged in

## < Pre-condition One >

# Post-conditions

* User can view admins response to their message in the notifications area of page

Use Case Specification: Send Newsletter

# Send Newsletter

## Brief Description

Admin can upload a newsletter that will then be emailed out to all users that are currently signed up to receive newsletters with the platform.

# Flow of Events

## Basic Flow

* Admin selects “Processes” button
* Admin selects “Newsletter” button
* Admin inputs email subject in the Newsletter form
* Admin uploads Newsletter pdf into the file receiver of form
* Admin inputs Send Date of Newsletter
* Admin inputs Send Time of Newsletter
* Admin selects “Send Newsletter”
* Admin selects “Yes” when prompted “Are you sure?”

## Alternative Flows

### < First Alternative Flow >

* Form puts out error after “Send Newsletter” is clicked if any form field is left empty

# Pre-conditions

* Admin must already be a registered user and registered admin
* Admin must already be logged in
* There must be an already created Newsletter to be uploaded and emailed

## < Pre-condition One >

# Post-conditions

* Admin can view past newsletter sends through the same “Newsletter” tab under “Processes”

Use Case Specification: Create System Backup

# Create Backup

## Brief Description

The system must be able to generate a copy of its files at the user’s request for redundancy. A limited number of users may have access to this page which allows him/her to save a copy of the system’s software found on the host server.

# Flow of Events

## Basic Flow

* User selects system partition
* User enters save destination
* User clicks submit
* User clicks to confirm settings
* Use case ends

## Alternative Flows

### Download Failure

The user will restart the basic flow with the previously entered settings pre-filled.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## User Has Internet Access

The user’s device must have a stable internet connection.

## User Is on System Backup Page

The user must have navigated through the system and reached the backup form.

## User Account Must Have Full-System Access

The user needs to be logged into an account with permission to view backup & recovery system functions.

# Post-conditions

## Backup Is Created and Stored

The system creates an image of the selected partitions and the user successfully downloads it to their device.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: Create New Donation

# Create Donation

## Brief Description

The system requires revenue to be feasible. CIS alumni and UofL affiliates will access this donation page to choose the payment amount, frequency of payments, and payment method. This should then be recorded and submitted to Stripe for processing.

# Flow of Events

## Basic Flow

* User enters gift amount
* User enters first name
* User enters last name
* User enters email address
* User enters street address
* User enters ZIP code
* User enters city
* User enters state
* User enters country
* User enters phone number.
* User enters credit card number
* User enters expiration month
* User enters expiration year
* User enters CVV security pin number.
* User clicks submit
* Use case ends

## Alternative Flows

### Failure to Enter All Required Information

A missing-required-information error message is displayed.

* User enters required information
* User clicks submit
* Use case ends

#### Incorrect Formatting

An error message describing the formatting error is displayed.

* User enters correct required information
* User clicks submit
* Use case ends

#### Credit Card is Expired

An error message describing the error is displayed.

* User enters credit card number
* User enters expiration month
* User enters expiration year
* User enters CVV security pin number.
* User clicks submit
* Use case ends

### Billing Address is Different Than Contact Address

* User unchecks box
* User enters street address
* User enters ZIP code
* User enters city
* User enters state
* User enters country

### Recurring Donation

* User checks box
* User selects frequency

### Payment is Declined

If at any point a payment, single or recurring payment is declined, the user must receive a notification explaining this.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Internet Access

The user’s device must have a stable internet connection.

## Navigated System

The user must have navigated through the system and reached the donation form.

## Logged In

The user must have previously completed the account login use case.

# Post-conditions

## Revenue Collected

The system submits to Stripe at specified frequency. Payment is received.

## Donation Information is Saved

The system saves all information connected to a user’s donation. This record must be maintained with a high level of security and restricted access.

# Extension Points

## Modify Donation

The system must allow this information to be modified by a user with necessary access at any point after its creation.

### Delete Donation

The system must allow donations to be cancelled by a user with necessary access at any point after its creation.

Use Case Specification: Making a Job Listing

# Create Job

## Brief Description

Employers want to reach CIS students as their needs arise

# Flow of Events

## Basic Flow

* User enters company name
* User enters phone number
* User selects industry
* User enters job title
* User enters job location
* User selects policy on remote work
* User enters quantity of positions
* User enters hire deadline date
* User enters job description
* User enters phone number
* User enters email address
* Use case ends.

## Alternative Flows

### Failure to Enter All Required Information

A missing-required-information error message is displayed.

* User enters required information
* User clicks submit
* Use case ends

#### Incorrect Formatting

An error message describing the formatting error is displayed.

* User enters correct required information
* User clicks submit
* Use case ends

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Internet Access

The user’s device must have a stable internet connection.

## Navigated System

The user must have navigated through the system and reached the create job listing form.

## Logged In

The user must have previously completed the account login use case and be on this create new job listing page.

# Post-conditions

## Job Listing Created

The system places the created job into a page of others for the students to browse and filter.

## Job Information is Saved

The system saves all information connected to a user’s listing. This information must be maintained with a high level of security and restricted access.

# Extension Points

## Modify Job

The system must allow this information to be modified by a user with necessary access at any point after its creation.

### Delete Job

The system must allow job listings to be cancelled by a user with access at any point after its creation.

Use Case Specification: Cancel Donation

# Delete Donation

## Brief Description

Users must be able to cancel future donations set to be collected. This use case begins on the same page as the modify donation form which displays past and future donations created by the user.

# Flow of Events

## Basic Flow

* User clicks hyperlink
* User confirms
* Use case ends.

## Alternative Flows

### Cancel Delete

* User closes confirmation
* Use case ends.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Internet Access

The user’s device must have a stable internet connection.

## Navigated System

The user must have navigated through the system and reached the donation form.

## Logged In

The user must have previously completed the account login use case.

## Donation on Record

The user must have previously completed the create donation use case. The user must have a future donation set to be collected in the future to modify it.

# Post-conditions

## Donation Cancelled

The system updates donation. Future donation is cancelled, and payment isn’t collected.

## Donation Information is Saved

The system saves all information connected to a user’s donation. This includes the donation before and after this use case has been completed. This record must be maintained with a high level of security and restricted access.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: Remove Job Listing

# Delete Donation

## Brief Description

Users must be able to cancel active job listings when they fill up or pass the deadline. This use case begins on the same page as the edit job listing form which displays inactive and active job listings.

# Flow of Events

## Basic Flow

* User clicks hyperlink
* User confirms
* Use case ends.

## Alternative Flows

### Cancel Delete

* User closes confirmation
* Use case ends.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Internet Access

The user’s device must have a stable internet connection.

## Navigated System

The user must have navigated through the system and reached the edit job listing form.

## Logged In

The user must have previously completed the account login use case and be on this job listing page.

## Listing on Record

The user must have previously completed the create job use case. The user must have an active job listing to remove it.

# Post-conditions

## Listing Cancelled

The system updates the listing. Future applicants won’t see listing.

## Listing Information is Saved

The system saves all information connected to a user’s job listing. This includes the listing before and after this use case has been completed. This record must be maintained with a high level of security and restricted access.

# Extension Points

There are no extension points associated with this use case.

Use Case Specification: Modify Donation

# Modify Donation

## Brief Description

Users who have previously created may want to make changes such as making payments more frequent. They will be shown a preview page of their past and future contributions. Future contributions will have a hyperlink leading to an alteration of the create donation form with the user’s information filled in.

# Flow of Events

## Basic Flow

* User selects Donation
* User clicks hyperlink
* User enters new information
* User clicks submit
* Use case ends.

## Alternative Flows

### Failure to Enter All Required Information

A missing-required-information error message is displayed.

* User enters required information
* User clicks submit
* Use case ends

#### Incorrect Formatting

An error message describing the formatting error is displayed.

* User enters correct required information
* User clicks submit
* Use case ends

#### Credit Card is Expired

An error message describing the error is displayed.

* User enters credit card number
* User enters expiration month
* User enters expiration year
* User enters CVV security pin number.
* User clicks submit
* Use case ends

### Billing Address is Different Than Contact Address

* User unchecks box
* User enters street address
* User enters ZIP code
* User enters city
* User enters state
* User enters country

### Recurring Donation

* User checks box
* User selects frequency

### User Cancel

User Has decided they no longer want to make changes to a donation

* User clicks hyperlink
* Use case ends

# Special Requirements

## Sensitive Information

Only the list four digits of any previous entered credit card number may be shown to the user.

# Pre-conditions

## Internet Access

The user’s device must have a stable internet connection.

## Navigated System

The user must have navigated through the system and reached the donation form.

## Logged In

The user must have previously completed the account login use case.

## Donation on Record

The user must have previously completed the create donation use case. The user must have a future donation set to be collected in the future to modify it. User may view previous donations.

# Post-conditions

## Changes Saved

The system updates the changed fields. Donations use new information.

## Donation Information is Saved

The system saves all information connected to a user’s donation. This includes the donation before and after this use case has been completed. This record must be maintained with a high level of security and restricted access.

# Extension Points

## Delete Donation

The system must allow donations to be cancelled by a user with necessary access at any point after its creation.

Use Case Specification: Edit Job Listing

# Modify Donation

## Brief Description

Users who have previously created a job listing may want to make changes such as moving the hire deadline. They will be shown a preview page of their job posts. Active opportunities will have a hyperlink leading to an altered create job listing form with the user’s information filled in.

# Flow of Events

## Basic Flow

* User selects job listing
* User clicks hyperlink
* User enters new information
* User clicks submit
* Use case ends.

## Alternative Flows

### Failure to Enter All Required Information

A missing-required-information error message is displayed.

* User enters required information
* User clicks submit
* Use case ends

#### Incorrect Formatting

An error message describing the formatting error is displayed.

* User enters correct required information
* User clicks submit
* Use case ends

### User Cancel

User Has decided they no longer want to make changes to the listing

* User clicks hyperlink
* Use case ends

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## Internet Access

The user’s device must have a stable internet connection.

## Navigated System

The user must have navigated through the system and reached the edit job listing form.

## Logged In

The user must have previously completed the account login use case and be on this create new job listing page.

## Listing on Record

The user must have previously completed the create job use case. The user must have an active job listing to modify it. User may view previously removed listings.

# Post-conditions

## Changes Saved

The system updates the changed fields. Visible listings use new information.

## Job Information is Saved

The system saves all information connected to a user’s listing. This information must be maintained with a high level of security and restricted access.

# Extension Points

## Delete Job Listing

The system must allow jobs to be removed by a user with necessary access at any point after its creation.

Use Case Specification: Create System Backup

# Create Backup

## Brief Description

The system must be able to generate a copy of its files at the user’s request for redundancy. A limited number of users may have access to this page which allows him/her to save a copy of the system’s software found on the host server.

# Flow of Events

## Basic Flow

* User selects system partition
* User enters save destination
* User clicks submit
* User clicks to confirm settings
* Use case ends

## Alternative Flows

### Download Failure

The user will restart the basic flow with the previously entered settings pre-filled.

# Special Requirements

There are no special requirements associated with this use case.

# Pre-conditions

## User Has Internet Access

The user’s device must have a stable internet connection.

## User Is on System Backup Page

The user must have navigated through the system and reached the backup form.

## User Account Must Have Full-System Access

The user needs to be logged into an account with permission to view backup & recovery system functions.

# Post-conditions

## Backup Is Created and Stored

The system creates an image of the selected partitions and the user successfully downloads it to their device.

# Extension Points

There are no extension points associated with this use case.

**Sequence Diagrams**

Sequence Diagrams are used to describe how, and in what order, certain processes happen. They are used by software developers to understand requirements for a new system or to document an existing process.

**Class:**Acceptance

**Use Case:**Approve Account

Diagram

Description automatically generated

**Class:**Account Admin

**Use Case:**Edit Account Admin (first), Delete Account Admin (second)

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

**Class:**Access

**Use Case:**Lock Account

Diagram, schematic

Description automatically generated

**Class:**User Account

**Use Case:**Search Mentor Account

Diagram

Description automatically generated

**Class:**Mentor Contact

**Use Case:**Contact Mentor Request

Diagram, schematic

Description automatically generated

**Class:**Admin Contact

**Use Case:**Contact Admin

Diagram

Description automatically generated

**Class:**Newsletter

**Use Case:**Send Newsletter

Diagram

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated with medium confidence

A picture containing chart

Description automatically generated

Chart

Description automatically generated with low confidence

A picture containing chart

Description automatically generated

A picture containing chart

Description automatically generated

A picture containing chart

Description automatically generated

A picture containing chart

Description automatically generated

Create Backup

A picture containing timeline

Description automatically generated

System Restore

Diagram, timeline

Description automatically generated with medium confidence

Create Donation

A picture containing graphical user interface

Description automatically generated

Modify Donation

Diagram

Description automatically generated with low confidence

Delete Donation

Box and whisker chart

Description automatically generated with low confidence

Create Job Listing

A picture containing timeline

Description automatically generated

Modify Job

Diagram

Description automatically generated with low confidence

Delete Job

Box and whisker chart

Description automatically generated with medium confidence

# Create A Discussion Board Post:



# Edit A Discussion Board Post:



# Delete A Discussion Board Post:



# View Discussion Board:



# Create An Event On The Calendar:



# Edit An Event On The Calendar:



# Delete An Event On The Calendar:



# Search Frequent Asked Questions:



# User Login:



**Class Diagram**

The class diagram presented shows the associations and relationships the classes involved with our system. Each class essentially represents a major webpage that will fulfill the completion of an important operation. Each of these classes denotes a necessary execution of a function of the system. One can see that the associations are all tied back to the Account superclass. The Account superclass is the parent class and the other classes such as login, messages, and the different specific kinds of accounts all have certain attributes derived from Account. The diagram was also derived using a prototype analysis. Essentially, every class was created from the prototype webpages. Each attribute being represented by a box/field, and each operation being represented by the button/function that would cause the system to do something. For example, the fields of the login page would represent the attributes and the and the actual button you click to login would be the operation as that data is sent back and forth from the class to the database back to the user in order to authenticate the credentials.

**Word, timeline

Description automatically generated**

**Database Design (Entity Relationship Diagram)**

An Entity Relationship (ER) Diagram is a database design tool and type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. In our project these entities are aspects of the website. ER diagrams are used in database design to model and design relational databases. By having a well-developed ERD it makes the job of the data architect easier when it comes time for them to develop, and later maintain the database.

**Diagram, schematic

Description automatically generated**

**User Interface Navigation Diagram (Windows Navigation Diagram)**

User interface diagrams, also called Windows Navigation Diagrams, allow you to define and model the basic structure of the interface of the website. Interfaces are any screens, forms, or reports shown on the website that the user would interact with. The diagram shows how these interfaces are connected and the click path that would be taken to get to any certain function of the website. This layout is taken from the use cases and their main flows that depicted the path a user would take to complete a task on the website.

Diagram

Description automatically generated

**Physical Architecture Design**

The physical architecture design shows the process of accessing the website. The user can log into the website through either, laptop, phone or tablet. The firewall provided by the university keeps the website to stay safe from attacks and ensures nothing will happen to the data stored there. The OneDrive cloud is used to store a backup of the data in case anything should happen to our main database storage, our SQL server. The .NET application server and Drupal content management system lets us design the website in a visually pleasing way for our users, encouraging them to continue visiting the site.

Diagram

Description automatically generatedDiagram, box and whisker chart

Description automatically generated

**Design Procedures for Security Concerns and Non-Functional Requirements**

The procedures we need to take for the non-functional requirements are:

* Connect to Amazon Web Services firewall
* Buy Microsoft OneDrive for Cloud Storage
* Implement a SQL server database
* Comply with Linux standards
* Create website using Drupal Content Management System
* Integrate Drupal’s content management system and Stripe Electronic Payment System

Connection to the AWS firewall contributes to user security by protecting user data being transmitted between the user device and server host. There, data will be stored in a SQL server and function as the website’s database. Backups of this data will be stored externally on Microsoft OneDrive. Compliance with Linux standards is an effort towards futureproofing the system as it enables broad host compatibility. Additionally, this will allow modular integration for external services and plugins.

Many critical system features rely on Drupal 9, an open-source content management software that allows for customizable plugins. The mentoring website will not be handling any of the payment information for donations. All payment information and donation amounts will be protected by Stripe, which is the payment management system. Stripe will be responsible for safeguarding the users’ payment information.

During design, building the SQL server is a precondition to building the mentoring website since this will involve creating SQL objects. Next, we will need to integrate the SQL server with the Microsoft OneDrive cloud storage so development versions may be stored. We will then begin designing the service and its features, then connect the AWS firewall to that website. Finally, we will integrate both Drupal and Stripe to the SQL server.

**Gantt Chart**

A Gantt chart is a horizontal bar chart that visualizes the tasks of the project, who was responsible for them, and the duration of the task. Below is the Gantt chart for iteration 4, iteration 5, and the upcoming elaboration spec. The elaboration spec brings together all previous parts of the project, which is why it includes the same material from the iteration 4 and 5 diagrams.

Iteration 4Table

Description automatically generated

Iteration 5

Table

Description automatically generated

Elaboration Spec

A picture containing timeline

Description automatically generated

**Prototypes**

**Approve Accounts**

Here the admin will be able to approve or deny user account requests.

Graphical user interface

Description automatically generated

**Edit Account Admin**

Here the admin will be able to select to edit any account registered to The CobWeb.

Graphical user interface, website

Description automatically generated

**Edit Account Admin**

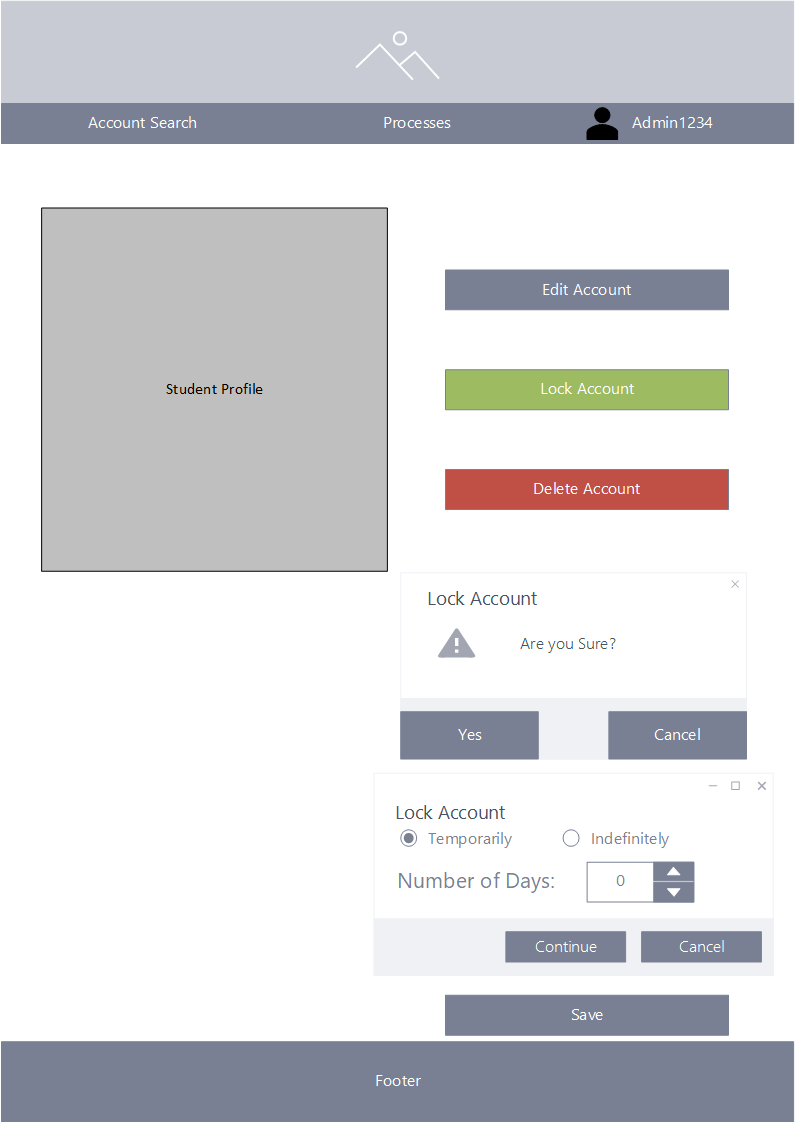
Here the admin will be able to select to Delete any account registered to The CobWeb. They will verify their decision and the account will be deleted from both the website and database.

Graphical user interface, application, website

Description automatically generated

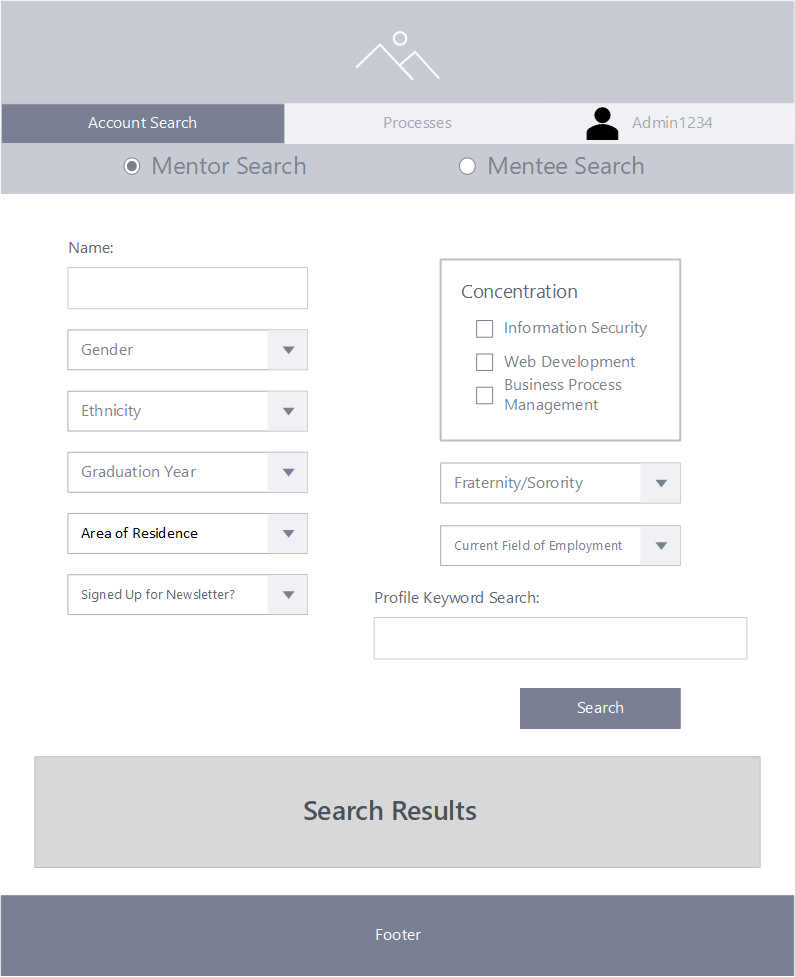
**Lock Account**

Here the admin will be able to lock an account. These account locks can either be temporarily or permanently/until the admin goes in and manually unlocks the account. This will be based on the lockout duration.



**Mentor Account Search**

Here all users will be able to filter through accounts on The CobWeb to look for mentors or mentees.



**Contact Mentor Request**

Through the contact mentor request mentees will be able to request the contact information of any mentor on the platform by clicking the “Request Contact Information” button (first image). After the mentor approves the request (second image), they will then have access to the mentors contact info and begin conversation.

Graphical user interface, website

Description automatically generated

Graphical user interface, application

Description automatically generated

**Contact Admin**

Through this form all users will be able to send a message to the systems admin voicing any questions, concerns, or need for support regarding the platform and the University of Louisville’s CIS department as well.

Graphical user interface

Description automatically generated

**Newsletter**

On this page the systems admin will be able to upload, schedule, and send off newsletter to those on The CobWeb’s mailing list. They may also review old sends.

Graphical user interface, application

Description automatically generated















System Backup and Restore

Graphical user interface, application, Teams

Description automatically generated

Create Job Listing

Graphical user interface, application, email

Description automatically generated

Create Donation

Graphical user interface, application, website

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application, table

Description automatically generated

Graphical user interface

Description automatically generated

Log-Out Account

Graphical user interface, application

Description automatically generated

View Landing Page

Graphical user interface, application, website

Description automatically generatedCreate System Announcement

Graphical user interface, text, application

Description automatically generated

Edit System Announcement

Graphical user interface, application

Description automatically generated

Delete System Announcement

Graphical user interface, application

Description automatically generated

Create Alumni Survey

Graphical user interface, application

Description automatically generated

Edit Alumni Survey

Graphical user interface, application

Description automatically generated

Delete Alumni Survey

Graphical user interface, application

Description automatically generated

Create Alumni Survey Database

Graphical user interface, application

Description automatically generated