

## **Alumni Management System Functionality**

The primary goal of the project is to build an alumni management system that will serve as the administrative backbone for the Information Systems and Security Department. It will be a website where authorized users can log in to manage all the information the department stores about their alumni.

The system will provide users with the core abilities to:

1. View alumni information.
2. Enter new alumni into the database.
3. Update existing alumni information.
4. Delete data when necessary.

To ensure data integrity, the system will include validation to help limit typing mistakes.

### **System Workflow for a User**

A user will start by going to a login page to enter their credentials. This login process confirms their identity and sets their permissions for what they are allowed to do on the site, such as viewing, inserting, or updating data. The system is designed to keep the user logged in as they navigate across pages.

After logging in, the user is directed to a central "Alumni Page," which is the main hub for managing alumni information. From here, the user can search for a specific alumnus or navigate to pages to manage their own details. The system is designed to capture a complete profile for each student, including their address information, degree history, employment path, donation record, and specialized skillsets. For example, a user can enter multiple addresses (work and home) and mark one as primary for contact. If an alumnus moves, the old address is kept in the system as "inactive" to maintain a historical record. This same historical tracking applies to employment, where new job records are created to track an alumnus's career growth over time.

*Below is a pseudocode example covering the login process of a user:*

Start Login Process:

1. Show the login screen with fields for Username and Password.
  2. Wait for the user to enter their Username and Password and click "Log In".
  3. If Username is empty OR Password is empty:  
Then show message: "Please enter both your username and password."  
Stop and wait for input.
  4. If Username OR Password contains invalid characters (like symbols not allowed):  
Then show message: "Please check your input for typos or errors."  
Stop and wait for input.
  5. Search the system for a user with the entered Username.
  6. If no user is found:  
Then show message: "This username does not exist."  
Stop the process.
  7. If Username is found, check the Password.
  8. If Password does not match:  
Then show message: "Incorrect password. Please try again."  
Stop the process.
  9. If Password matches:  
Then:
    - Create a variable called `currentUser`.
    - Write the user's information into `currentUser` (e.g., name, ID, and role).
    - Mark the user as 'Logged In'.
    - Check the user's Role:
      - If Admin:  
Then allow full access to view, add, edit, and delete alumni records.
      - If Staff:  
Then allow access to view and edit records but not delete records.
      - If Viewer:  
Then allow access to view records only.
    - Save this `currentUser` information so the system can use it on every page.
  10. Take the user to the main dashboard of the system.
- End Login Process.

*Below is a pseudocode example covering how the user stays logged in while navigating through the site:*

Each time the user opens a new page:

1. If `currentUser` is empty (no one is logged in):  
Then send the user back to the login page.
2. If the user has been inactive for more than 15 minutes:  
Then clear `currentUser` and log them out.  
Show message: "You have been logged out due to inactivity."
3. If `currentUser` exists:  
Then check their Role:
  - If Admin Then Show all tools.

- If Staff Then Show edit and view tools.
  - If Viewer Then Show view-only tools.
- Keep allowing access based on their role.  
End Page Access Check.

*Below is a draft of the Alumni homepage:*

