SOFTWARE REQUIREMENTS SPEFICATIONS

FOR

ACADEMIC COUNSELLING ASSISTANT

Prepared By:

Md. Masudur Rahman (1631189042)

Ishrat Jahan Ananya (1631636042)

Saraf Sumaita Hasan (1631258042)

Shadab Hafiz Choudhury (1631335042)

Submitted To:

Dr. Nabeel Mohammed (NbM)

Assistant Professor

(CSE327.1)

Table of Contents **Page Number**

Introduction

1. Purpose

   The purpose of this document is to present a detailed description of the Academic Counselling Assistant. It will explain the goals of the project, the features implemented, the navigation of the user interface and finally the limitations and future-proofing aspects of this project.

2. Document Conventions

   This document has various sections plainly divided and each subsection numbered. In the cases where fully-fledged subsections were unwarranted, italics have been used to divide individual aspects or points.

3. Intended Audience and Reading Suggestions

   This document is intended for both project managers, who will find the list of functions and requirements useful for ensuring efficient management of the development process. Similarly, the developers will also find this document somewhat useful, as it illustrates some requirements they need to meet when building this product's functions.

   This document does not illustrate any specific points of failure and uses some technical terms, so it may be less useful to testers and business managers.

4. Project Scope

   The objective of Academic Counselling Assistant is to create a platform that will aid students who wish to study abroad by giving them the ability to explore their options and browse Universities and Subjects. In addition, the project aims to specifically to guide the ones who have little or no idea about the whole process, thus being a one stop destination for all educational related queries. This means students can not only get an idea of where they want to study, but also the process by which they can apply for admission, visas and related matters. Some of the individual features of this platform are already present in various other platforms, but with this project, we aim to unify all the disparate features under one platform.

   The ultimate objective of the Academic Counselling Assistant is to provide a straightforward and painless method for students to ensure they pick the best place possible for higher studies.

5. References

   This document makes no explicit references to any other document or source of information, and stands alone.

Overall Description

1. Product Perspective

   The Academic Counselling Assistant is designed to be an all-inclusive platform for prospective international students. As such, the majority of its features are aimed at providing the necessary information and guidance needed by students to make the decision of where to travel and how to do so. The primary users of this product will be students, but there will also be access for administrators, who manage the system 'behind the scenes'.

   While the front-end will be a series of web pages, the back-end of the product will consist of a Database system and a Web Server to host and run our program. The 'administrators' will be able to access the database through a web interface to add, remove or change the contents of the database. The database itself will contain the user account details and the details of institutions.

2. Product Functions

   Initially, the information regarding universities and institutions will be stored in a database. The information itself will be gathered from other websites by web-crawling and can be updated at regular intervals.

   Afterwards, the user can use the website by first making an account and entering their details. These details are also stored in a database. The user can then use the various features of the website detailed later in this document.

   A machine learning algorithm is utilized, using both the user details and the details of the universities stored in the database, to generate a list of suggestions for each individual user.

3. User Classes and Characteristics

   There are two main user classes present in this product. The General User, who sees the front-end of the system and is essentially anyone who uses the product, and the Administrator, who also sees the back-end databases and manages the information stored there.

   - General Users
     - Making an account
     - Editing user and account details
     - Using the features detailed below under System Features.
   - Administrators
     - Viewing user information
     - Managing and updating user information
     - Viewing tables present in the database
     - Managing and updating database tables (including details of institutions)

4. Operating Environment

   As it is a web app, this product can be used on all desktop and mobile operating systems that support the use of a web browser. This product should function properly in all major web browsers, on both desktop and mobile, including but not limited to Internet Explorer (9 onwards for best compatibility), Microsoft Edge, Mozilla Firefox, Google Chrome, Opera and Apple Safari.

5. Design and Implementation Constraints

   As this product is being implemented as a web app, the basic functions of the website will experience little design or implementation constraining so long as all the External Interface Requirements, listed below, are made use of properly.

   The main implementation constraints will come from a lack of time and limited data sets to train any advanced machine learning algorithm, and in developing a web-crawler that will function even through changes in page layout.

6. User Documentation

   The documentation for this system consists of the two documents produced during planning and development, the SRS and SDS, to be used by the owners of this product, and a short user manual that will be provided for the end user in the final product.

7. Assumptions and Dependencies

   This product relies on information gathered from other sources. While it maintains a local copy of the info currently being displayed, there is no way to tell if the original information gathered from a third-party website is correct or not.

   Additionally, the website is being hosted on a local machine and may be shifted to a third-party web provider. If this is done so, the uptime of the website (the time during which it is accessible) may depend on the third-party.

External Interface Requirements

1. User Interfaces

   The user interface will consist of several separate webpages, each of which will carry out a different task. The general layout of pages will be created from a suitable, preexisting website template.

   Index Page – This page will contain an introduction to the website, as well as instructions on how to proceed. It will also contain links to the other pages, allowing the user to navigate around the website.
   Registration and Login Page – Allows the user to register a new account in order to use the features of the website. If they user has an account, they will then log in with their credentials.
   Profile Page – Allows the user to manage their details and enter or update personal, academic and other information.
   Browse Universities Page – This page will contain a list of universities along with their information, extracted from the database. This page will also contain a section for comparing universities
   Suggestions Page – This page will contain the list of suggestions generated for the user, as well as a career map in order to help with the suggestions
   Guidance Page – This page will contain the step-by-step process of guiding a user through university applications, along with the checklist.
   Admin Panel – This page is only accessible to the administrator and will contain representations of the databases and the options to add, delete or edit the database contents.

2. Hardware Interfaces

   This project makes no use of any specific hardware interfaces, being contained entirely within the bounds of a web app. A hardware interface, specifically a Web Server, may come into play if the owners of this product decide to host the website on a private server as opposed to letting a third party manage that.

3. Software Interfaces

   The web pages will contain PHP. In order to run this PHP code, an Apache webserver will be used, hosted locally through Xampp. Additionally, Laravel will be used as a framework for the PHP. The databases will be hosted in the MySQL Database Manager. The majority of the web interface will be written in Html, Css and JavaScript, and as thus, will not require any specialized software to run other than an up-to-date web browser.

   Additionally, an external web-crawler software will be used to gather data, in order to minimize the cost and time spent in writing an individual web-crawling algorithm from scratch.

4. Communication Interfaces

   At the current scale of this project, it will be hosted on a local machine and thusly, will not require an internet connection to function. However, it will require an internet connection to run the web-crawler and gather updated data regarding universities from the internet. If this product is deployed on a wider scale with a dedicated web host, then it will be necessary to have a server-side https authentication and for the user to have an internet connection to use this product.

System features

1. User Account Information Management
   a. Priority – High
   b. Description - Users can make an account and input information including, but not limited to, personal information (name, contact details, etc), academic information (current level of study, CGPA, research papers, academic awards, work experience, etc) and preferences (preferred subjects, preferred country, financial condition, etc). The user can also use a pre-existing account from a third-party account credentials provider, Google.
   c. Functional Requirements:
      i. R1, Login Screen – There must be a panel or window or page with fields for users to enter an identifier (username or email) and a verifying password.
      ii. R2, Registration Screen – There must be a panel or window with fields for users to enter their desired identifier, password, and other details, in order to create an account with our system
      iii. R3, Account Info Management Screen – There must be a panel, window or page with fields for users to, once they have logged in, modify or add any of the details mentioned above in section b.

2. Searching for Institutions
    a. Priority – High
    b. Description - Users can use a wide range of filters, including, but not limited to, country, subjects, ranking and popularity, and expense, to search through a global database of universities and institutions for higher education
    c. Functional Requirements:
        i. R1, Table to Display List – The university listings should be displayed to the user in the form of a table
        ii. R2, Manipulating List – The user should have the ability to filer and sort the list in several ways described in b.

3. Institution Suggestions
    a. Priority – Medium
    b. Description – Users are provided suggestions on what university would be best for them. These suggestions are generated by algorithms that use variables from both the user information and university information.
    c. Functional Requirements –
        i. R1, Displaying Suggestions – A panel that displays a list of recommended universities with some details.
        ii. R2, More Info – Additional Panel displaying the reasons for recommendation.

4. Comparing Institutions
    a. Priority – Medium
    b. Description – Users can select multiple institutions and directly compare them in terms of statistics
    c. Functional Requirements –
        i. R1, Selecting – Select institutions for comparison from the main list in 2.
        ii. R2, Comparing – List the selected institutions' details side by side in a separate window or panel.

5. Financial Calculator
    a. Priority – Low
    b. Description – Users can calculate the total cost of the degree and the institution they're planning to study at, in order to plan out their finances in advance.
    c. Functional Requirements –
        i. R1, Display Costs – List the different costs involved and the annual sum total
        ii. R2, Adjust Budgets – Let users adjust the budget for some costs and ensure the sum total updates

6. Career Map
    a. Priority – Low
    b. Description – Users can choose an appropriate university to see a proper career map.
    c. Functional Requirements –
        i. R1, Display Career Map – A separate window or panel in which a career map in the form of a flowchart, along with other information or statistics, is displayed for a specific subject or degree
7. Application Guidance

a. Priority – High
b. Description – Users can peruse an interactive guide that will clearly outline the steps they need to take to apply, and they will also have access to an individualized checklist for application requirements.
c. Functional Requirements –
    i. R1, Display Steps – A separate window or panel with multiple pages that displays each step of the entire process
    ii. R2, Remembering Steps – Instead of starting from the beginning every time, if a user closes the Application Guidance, it will remember what step the user was last on and jump there directly.
    iii. R3, Updating Steps through User Information – The steps will use the user's personal data they input. For example, if the user specifies they have taken an English Proficiency Test, the specific step for that will be skipped over.

Non-Functional Requirements

1. Performance Requirements

   The website should be efficient on memory and loading times. Every page should load in less than one second with a modern internet connection of 1 Mbps.

2. Safety Requirements

   The website, once deployed on a web server rather than a local host, should get an *HTTPS* connection certificate, in order to ensure when a user enters the website they know their data is secure and the connection is encrypted.

3. Security Requirements

   As this website will contain personal data regarding its users, security must be maintained. Users should be cautioned from putting sensitive data anywhere, and their passwords will be stored in hashed form in order to prevent brute-force access to their accounts. Users and Administrators should also be discouraged from sharing their account credentials, to prevent data breaches.

4. Software Quality Assurances

   The main requirements here is that the system is straightforward and easy to use, since it is a replacement of other pre-existing solutions to the same problem where those solutions are convoluted and a hassle to use properly.

5. Business Rules

   There are no specific business rules to be maintained in regards to this product.

Other Requirements

1. Federated User Credentials

   In addition to making their own accounts whose credentials are stored in the project's database, users will have the option of using their Google accounts to log in. As Google is one of the largest and widely used platforms on the internet, many of our users may already have their own Google accounts. This would reduce the hassle for a new user of the website and improve ease of access.

   Again, security must be maintained on both the developers' and users' side, in order to ensure the users do not suffer any security breaches regarding their Google account.

Appendices

1. Appendix A: Glossary

   Administrator – A person, preferably an employee of the owner of this product, who will manage the back-end of the system

   Database – A group of tables organized in a manner as to be related to each other and so maintain ease of updating, usage and maintenance.

   (Web) Server – A computer that will contain the website and assorted applications and enable anyone to access it at any time the computer is active.

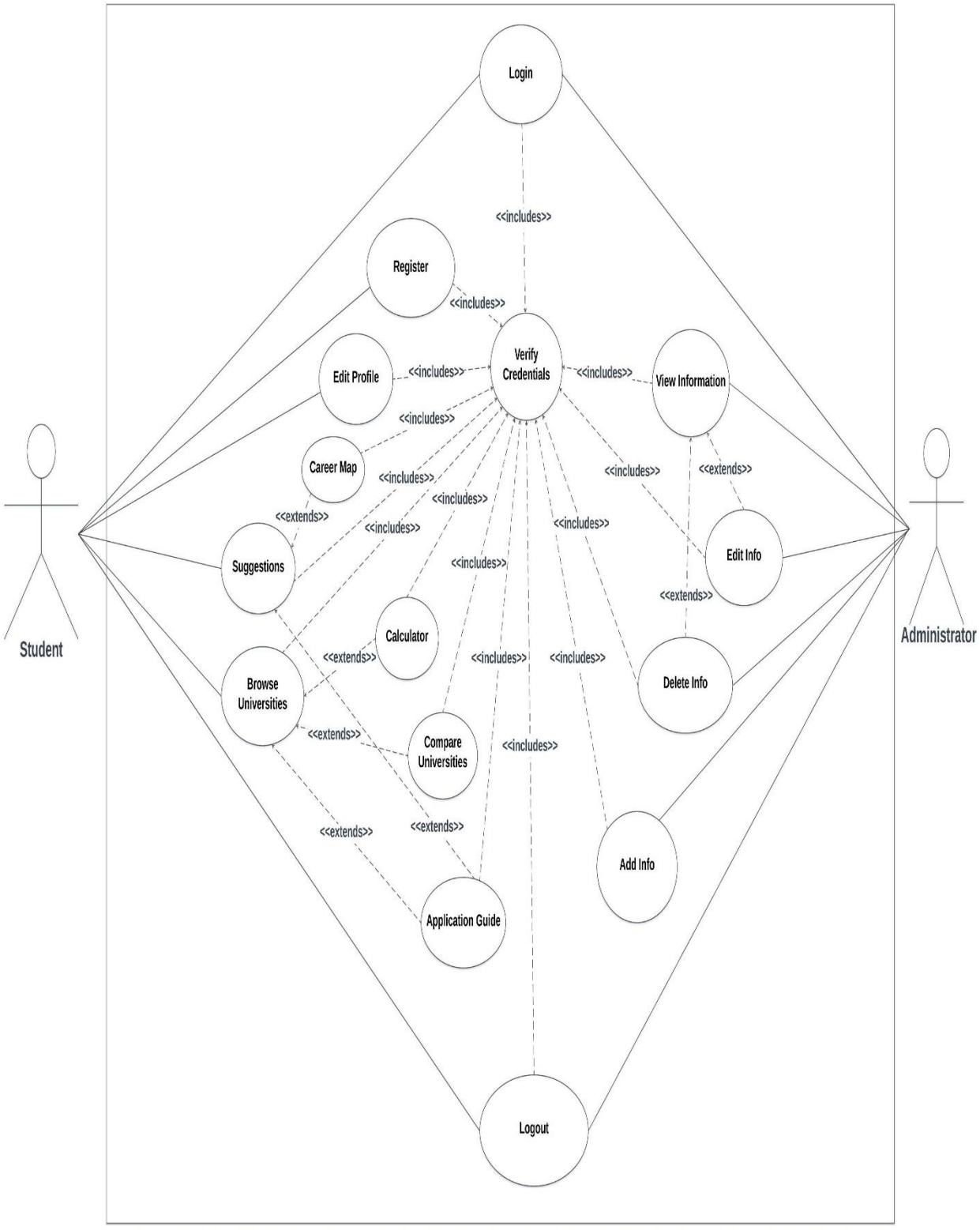
   PHP – Hypertext Preprocessor. A programming language used to connect a website's front-end to back-end databases.

   Web Crawler – A program that searches other websites, gathers data and saves it.

   Website – A series of pages that may be viewed in a web browser software, which are available on almost all PCs and modern mobile phones. A short list of common web browsers is mentioned in the Operating Environments section.

2. Appendix B: Analysis Models

Use case Diagram:



Expanded Use Cases:

Use Case	Register
Actors	Student
Purpose	To create an account
Overview	The user makes a username, password and enters other details to create an account.
Type	Primary
Cross references	Use cases: Verify Credentials

Typical course of events:

Actor Action	System Response
1. This use case is initiated when a user wants to create a new account.	2. The system provides the user with necessary fields to fill in.
3. The user enters the required information.	4. Account is created.

Alternative courses:

Section 3: If the entered username is already taken, user is notified.

Section 3: If the user fails to enter the new password twice without a mistake, user is notified.

Section 4: If any of the compulsory fields are left empty, prompt is shown.

Section 4: If third party authentication fails, error message is shown.

Use Case	Login
Actors	Student, Administrator
Purpose	To log in to the account.
Overview	The user enters his/her username and password and logs in.
Type	Secondary
Cross references	Use cases: Verify Credentials

Typical course of events:

Actor Action	System Response
1. This use case is initiated when a user wants to log in to their account.	2. The system asks for the user's username and password.
3. The user enters their username and password.	4. The system logs the user into their account.

Alternative courses:

Section 4: If the entered username or the password is incorrect, the system shows an error message and asks for the username and the password again.

Use Case	Suggestions
Actors	Student
Purpose	To view the university suggestions page.
Overview	The user clicks to view the university suggestions generated by the system based on the information given by the user.
Type	Primary
Cross references	Use cases: Verify Credentials

Typical course of events:

Actor Action	**System Response**
1. This use case is initiated when a user clicks suggestions.	2. The system opens the suggestions page for the user.

Alternative courses:

Section 2: If none of the universities meet the user requirements, then the user is notified that there are no results to be shown.

Use Case	Browse Universities
Actors	Student
Purpose	To search for the suitable institutions.
Overview	The user searches for universities by choosing various filters. The relevant universities are shown.
Type	Primary
Cross references	Use cases: Verify Credentials

Typical course of events:

Actor Action	System Response
1. This use case is initiated when a user clicks browse universities.	2. The system gives the user a number of filters to search by.
3. The user selects a filter.	4. The system shows the suitable universities to the user.

Alternative courses:

Section 4: If none of the universities under the chosen criteria is suitable for the user, then the user is notified that there are no results to be shown.

Use Case	Calculator
Actors	Student
Purpose	To calculate total expense.
Overview	The user inputs the necessary figures into the appropriate fields and the amount is calculated and shown.
Type	Primary
Cross references	Use cases: Verify Credentials Use cases: Browse Universities Use cases: Suggestions

Typical course of events:

Actor Action	System Response
1. This use case is initiated when a user clicks calculator.	2. The system opens the calculator and waits for user input.
3. The user enters the relevant figures.	4. The result is calculated and shown to the user.

Alternative courses:

Section 4: If any of the fields are left empty, an error message is shown.

Section 4: If any of the inputs are of the invalid type, for example if by mistake the user gives an alphabetical input instead of a numerical input, an error message is shown.

Use Case	Career Map
Actors	Student
Purpose	To view the career map for each university.
Overview	The user selects a university and views the corresponding career map, which shows the popular degrees pursued in this institution and their career prospects.
Type	Primary
Cross references	Use cases: Verify Credentials Use cases: Browse Universities Use cases: Suggestions

Typical course of events:

Actor Action	System Response
1. This use case is initiated when a user selects a particular university and clicks career map.	2. The system opens the career map for the user to see.

Alternative courses:

Section 2: If no university is selected, career map is not shown. The user is asked to select a university.

Use Case	Compare Universities
Actors	Student
Purpose	To compare universities with one another.
Overview	The user selects a number of universities and compares them in terms of statistics and other factors.
Type	Primary
Cross references	Use cases: Verify Credentials Use cases: Browse Universities Use cases: Suggestions

Typical course of events:

Actor Action	System Response
1. This use case is initiated when a user clicks compare universities.	2. The system will ask the user to select more than one university from a list.
3. The user selects multiple universities.	4. The system gives the user a side by side display of the statistical data of the universities for the user to compare.

Alternative courses:

Section 4: If the user selects only one university or none of the universities, an error message is shown.

Use Case	Edit profile
Actors	Student, Administrator
Purpose	To make changes to the profile information.
Overview	The user adds new information or modifies existing information in the profile and then saves the changes.
Type	Primary
Cross references	Use cases: Verify Credentials

Typical course of events:

Actor Action	System Response
1. This use case is initiated when a user clicks edit profile.	2. The system opens the existing profile information page, in an editable format.
3. The user makes necessary changes and saves it.	4. The system saves the changes.

Alternative courses:

Section 4: If any of the compulsory fields are left empty, a prompt is shown.

Section 4: If user is changing the password, and fails to meet the preconditions set for the password when he/she enters the new password, an error message is shown.

Use Case	Application Guide
Actors	Student
Purpose	To get guidance on applying to universities.
Overview	The user views the steps required to be fulfilled in order to apply for a university. The user can also view a checklist which gets updated with time, as the user makes progress and updates his/her profile.
Type	Primary
Cross references	Use cases: Browse Universities Use cases: Suggestion Use cases: Verify Credentials

Typical course of events:

Actor Action	System Response
Use Case: Suggestion or Browser Universities included	
2. This use case is initiated when a user clicks application guide.	3. System shows the appropriate checklists and all the other guidelines.

Alternative courses:

Section 2: If the user didn't choose a university first or not being suggested a university then it will show an error message.

Use Case	Logout
Actors	Student, Administrator
Purpose	To log out of the account.
Overview	The user clicks logout and gets out of the site.
Type	Primary
Cross references	Use cases: Verify Credentials

Typical course of events:

Actor Action	System Response
1. This use case is initiated when a user clicks logout.	2. The system logs the user out of their account.

Alternative courses:

Section 2: If the user is not logged in to start with, the system will show a prompt.

Use Case	View Information
Actors	Administrator
Purpose	To view information stored regarding the users and the universities.
Overview	The admin selects a category; users or universities and views the tables, that stores information regarding them in the database.
Type	Primary
Cross references	Use cases: Verify Credentials

Typical course of events:

Actor Action	System Response
1. This use case is initiated when a user clicks View Information.	2. The system gives the user two categories to choose from; Users or Universities.
3. The user selects one of the two categories.	4. The tables present in the database under that category are shown to the user.

Alternative courses:

Section 4: If no tables are present in the database under the chosen category, user is notified.

Use Case	Edit Info
Actors	Administrator
Purpose	To make changes to the information contained in the table.
Overview	The admin will firstly trace a particular user or university and then edit the information stored regarding them.
Type	Primary
Cross references	Use cases: Verify Credentials

Typical course of events:

Actor Action	System Response
1. This use case is initiated when a user clicks the Edit Info.	2. The system will provide a user with a search bar.
3. The user will search for a particular user or university.	4. The system shows the information stored regarding that user or university.
5. The user makes changes.	6. The system saves the changes.

Alternative courses:

Section 4: If the user searches for a user or university that is not present in the database, the user is notified.

Use Case	Add Info
Actors	Administrator
Purpose	To add new information into the database.
Overview	Admin can add information of a user
Type	Primary
Cross references	Use cases: Verify Credentials

Typical course of events:

Actor Action	System Response
	1. Register included.

Use Case	Delete Info
Actors	Administrator
Purpose	To remove a user or a university from the database.
Overview	The admin will firstly trace a particular user or university and then delete that entity.
Type	Primary
Cross references	Use cases: Verify Credentials

Typical course of events:

Actor Action	System Response
1. This use case is initiated when a user clicks the Delete Info.	2. The system will provide a user with a search bar.
3. The user will search for a particular user or university.	4. The system shows the information stored regarding that user or university.
5. The user clicks delete.	6. The system saves the changes.

Alternative courses:

Section 4: If the user searches for a user or university that is not present in the database, the user is notified.

Use Case	Verify Credentials
Actors	System
Purpose	To authenticate a user.
Overview	The user is authenticated by checking their username and password.
Type	Secondary
Cross references	Use cases: Login

Typical course of events:

Actor Action	System Response
1. This use case is initiated whenever a user clicks on any sort of function.	2. The system checks the validity of the user by checking the username and the password.

Alternative courses:

Section 2: If the validation is not successful, the requested function is not executed.