



Software Testing Plan

April 4, 2022

Team CareerNet

Team Sponsors:

Dr. Andy Wang,
José R. Díaz Aquino

Team Mentor:

Han Peng

Team Members:

Carter Taylor,
McKenzie Clark,
Carmen Montalvo,
Ran Li

Table of Contents

1.0 Introduction	1
2.0 Unit Testing	3
2.1 Front End Functionality	3
2.2 Back End Functionality	9
2.3 Database Verification	10
3.0 Integration Testing	11
3.1 Front End Result Testing	11
3.2 Database Result Testing	12
4.0 Usability Testing	14
4.1 Web Page Design	14
4.2 Element Quality	14
4.3 Page Modularity	15
5.0 Conclusion	16

1.0 Introduction

CareerNet is a web application that scrapes student information from popular career-building websites and surveys and organizes the relevant data to be easily accessible by college deans so that they may understand the career growth of their students. Tracking a student's career milestones is valuable to any university because it can give insight into not only the student but the development of the degree plan they are in. This allows a college to get a better grasp on what degree plans are succeeding and how they can improve upon them. This product applies to all colleges interested in creating a student success-focused university.

Software testing is a process of evaluation that is done to ensure the program is doing what it is supposed to do, find any error within the program, and improve the performance of the application. In order to test our web application, we plan to put our program through unit, integration, and usability testing. For our unit testing, we will focus on each module within the front-end and back-end. For the front-end, we are looking at the flow and outcomes of the login system, CSV module, searchable student milestones, and administrator settings. As for the back-end, we focus on testing its functionality using the Postman software as well as verifying the information being put into our program is the same that is being stored in the database. Our integration testing will focus on looking at the results of our major modules which are our front and back end. Lastly, for the usability testing, we will look at how our web application functions with different browsers, user interaction, and its design and interaction within itself.

Overall, for CareerNet, we are performing several different tests that are all important in ensuring that our application is doing everything it was designed to do, but our main focus will be our unit testing. This step seems to be more intensive since it is not only the components themselves we are testing but the way in which they interact with each other, as well as being the first topic of discussion in this document.

2.0 Unit Testing

Unit testing is a process that focuses on testing an application down to its simplest state. This is to isolate every logical piece of code to ensure that the code works as it was intended, but also that there are no vulnerabilities that can be taken advantage of in the application. This section will go over three main parts of this project, the front end, the back end, and the database. Unit testing these areas of the application allows for a broad and thorough overview of testing its code.

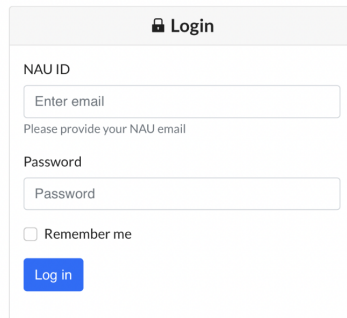
2.1 Front End Functionality

This application's front-end system handles all user interaction, as well as communicates the user input to the back end. Testing this part of the system ensures that the user is seeing what they are supposed to be seeing. Front-end functionality testing will be separated per page. Included in each test is a description of what the page will look like, the main flow of the functionality of the page, as well as the expected outcome.

2.1.1 Login System

2.1.1.1 Description

The purpose of this function is to allow the user to log in with their previously given credentials, giving the user access to certain permissions and information within the system.



The screenshot shows a login form titled "Login" with a lock icon. It contains two input fields: "NAU ID" with a placeholder "Enter email" and "Password" with a placeholder "Password". Below the password field is a checkbox labeled "Remember me". At the bottom is a blue "Log in" button. A note below the email field says "Please provide your NAU email".

Figure 1: Screenshot of the application login page

2.1.1.2 Flow of Functionality

- a. Go to the application login page.
- b. Enter user email address and password.
- c. There is a 'Remember Me' box option, to save credentials.
- d. Select the login button.
- e. The user is directed to the Student Search page.

2.1.1.3 Expected Outcome

The user is successfully logged into their designated profile, with their allotted user permissions and data.

2.1.2 CSV Upload Module

2.1.2.1 Description

The purpose of the function is to allow the user to upload two types of CSVs to the application. Each CSV should be formatted according to the user manual. If an uploaded CSV does not match the correct format, a popup will be displayed notifying the user of the error. One CSV is for uploading new student data, the other allows for the user to upload student milestone data.

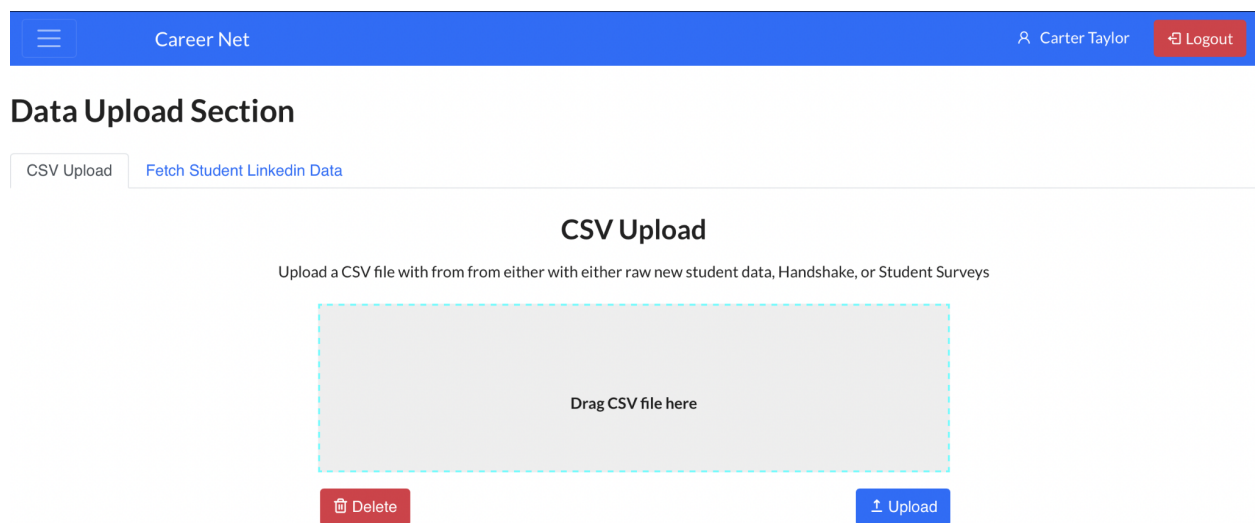


Figure 2: Screenshot of CSV Upload application page

2.1.2.2 Flow of Functionality

- a. Drag CSV to the box that states 'Drag CSV file here'.
- b. The system will automatically give a preview of the uploaded CSV to the user.
 - i. If the CSV is successfully processed with no errors, a notification will be displayed to the user stating that the

CSV's format has been recognized and to select the 'Upload' button.

- ii. If the CSV does not match either of the two required formats, a notification will be displayed to the user stating that no valid CSV has been provided and to check the user manual for correct formatting.

c. Select the 'Upload" button.

d. The system will process CSV.

- i. If a certain student's data can not be uploaded, a notification will be displayed to the user stating what students were not able to upload.

- ii. If the upload process was successful, the user will receive a notification stating that it was a successful upload.

2.1.2.3 Expected Outcome

The user's CSV has been successfully uploaded to the database.

2.1.3 Student Milestone Search

2.1.3.1 Description

The purpose of this page is to allow the user to search for specific student information that is stored in the database. The user can choose to select a student to view their currently stored milestones.

Figure 3: Screenshot of the Student Milestone Search Page

2.1.3.2 Flow of Functionality

- a. Type a name into the search bar. Users have the option to sort by major or by the date uploaded.
- b. Select chosen student
- c. Page then leads to the selected student's milestone timeline. Data is accurate and up to date from collected CSV and LinkedIn information.

2.1.3.3 Expected Outcome

Users can find and view any individual student's information that has been uploaded to the application.

2.1.4 Administrator Settings

2.1.4.1 Description

The purpose of this function is a page that is only able to be viewed by users with administrative privileges. This page allows the administrator to add or edit user profiles, as well as control user access to student information.

The screenshot shows the 'Search Student' interface. At the top, there is a blue navigation bar with a hamburger menu icon, the text 'Career Net', a search icon, the user name 'Carter Taylor', and a 'Logout' button. Below the navigation bar, the main content area is titled 'Search Student'. It features a large search input field with the placeholder text 'Enter Student Name'. Below the search field, there is a prompt: 'Search a User, and hit Fetch, to grab the students information'. To the left of this prompt is a 'Major Filter' dropdown menu, and to the right is a 'Date Range' dropdown menu. Below these elements is a blue 'Fetch Student Data' button. At the bottom of the form, there is a note: 'Please click the fetch button in order to fetch the most up to date student information'.

Add User

User Information

First Name

Last Name

Email address

Permissions

User Type

Upload New Data Check for user to have ability to upload new student data

Choose Major Access

Computer Science

Mechanical Engineering

Applied Computer Science

Electrical Engineering

Figure 4: Screenshot of Administrator Setting's Add User Page

2.1.4.2 Flow of Functionality

Flow of Adding a User Profile

- a. Select the 'Add User' tab underneath Settings.
- b. Input relevant user information, such as name and email address.

- c. Under the permissions title, choose permissions for what the new user will have access to.
- d. Select the add user button.
- e. New User is added to the system and a notification is sent to the user.

Flow of Editing a User Profile

- a. Select the 'Edit User' tab.
- b. Search for existing users in the search bar.
- c. The same format as adding a new user appears. Users can change any users with the 'Normal Faculty' user type.
- d. When submitting changes, the user receives edit user confirmation notification.
- e. Once confirmed, the user will receive another notification that states that the changes have been made.

2.1.4.3 Expected Outcome

Users can add and edit a new user profile to the system.

2.2 Back End Functionality

This application's back-end system handles all logic handling, as well as communicates the user input to the database. Testing this part of the system ensures that the application is responding to what the user is inputting. Back-end functionality testing will be conducted by a testing software called Postman.

2.2.1 Postman Testing

For this section of functionality testing, the application will be tested using the software called Postman. Postman is an API tester that logs and tests HTTP requests to ensure that the back-end system is doing what it is supposed to. Postman testing will be implemented on the entire application, with the main focus on CSV Uploading and Administrative Settings. These two sections are being focused on because of the large amount of data being handled or changed based on user input.

2.3 Database Verification

This application's database system stores all user and student data, as well as communicates the data back to the user in the front end. Testing this part of the system ensures that the data being inputted is the same information that is being stored. Database verification testing will also be separated per data table type, to ensure that the data is being stored properly and efficiently.

2.3.1 Database Verification Comparison

For this section of functionality testing, we will rely on comparing data sets between what the input is, what is being stored, and how the output is being retrieved. We will do this in small quantities to ensure that we can track the data accurately. These comparison tests will be implemented in different parts of the application, primarily in areas where data is being entered or changed for the first time. The sections that we will be focusing on are CSV student upload and admin settings.

3.0 Integration Testing

Integration testing is the process of logically testing individual major modules of an application to ensure that they work well separately, and then combining all modules and testing how well they work together. This type of testing's main goal is to understand how interfaces work between modules. This is important for this application because it allows us to understand if there are any parts of the application that are being neglected between modules.

Testing for this application can be separated into two major modules, Front End Result Testing and Database Result Testing. These modules were chosen because they are major components of the application but also because they can be separated cleanly for testing purposes.

3.1 Front End Result Testing

With front-end result testing we will focus on the front-end pages and how they interact with the back-end and database components. Because integration testing focuses on how individual modules interact with each other, for this example we will focus on the Front End CSV Upload Module. The CSV upload page takes in a user uploaded CSV and passes it through the back end to be processed and stored in the database. For testing, we will focus on individual functions that are used in the back end and pass dummy data through them. The dummy data is used to track how the function works on an individual level to make sure the logic is still intact. Below in Figure 5, we see a schema of the communication between the front end, back end and database.

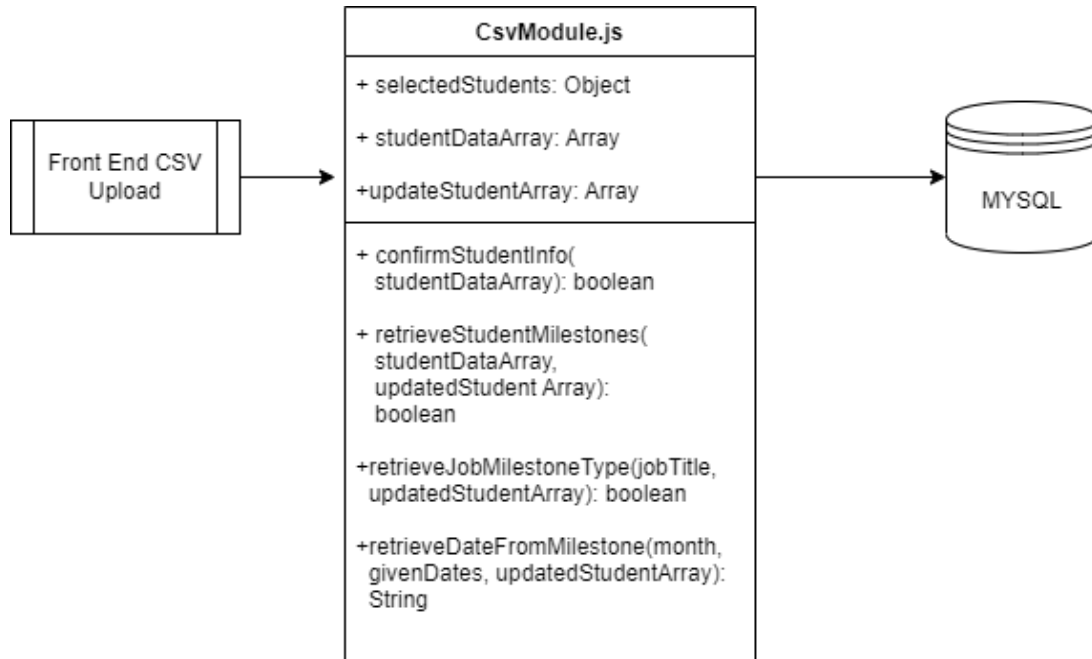


Figure 5: CSV Upload Communication Schema

3.2 Database Result Testing

With the database result testing, we will focus on the communication between the back end of the application and the application’s database. Continuing with the previous example of focusing on the CSV upload module, we are going to be analyzing the variables that are being stored in the database with the information that has been uploaded in the front end and passed to the database. The dummy data that we had previously entered for front-end testing is now the same data that we will be handling and sorting in the database. For testing purposes, we will follow how the data is being stored and sorted to understand how both modules work as a whole. Figure 6, is an example of how the tables are connected in the database system.

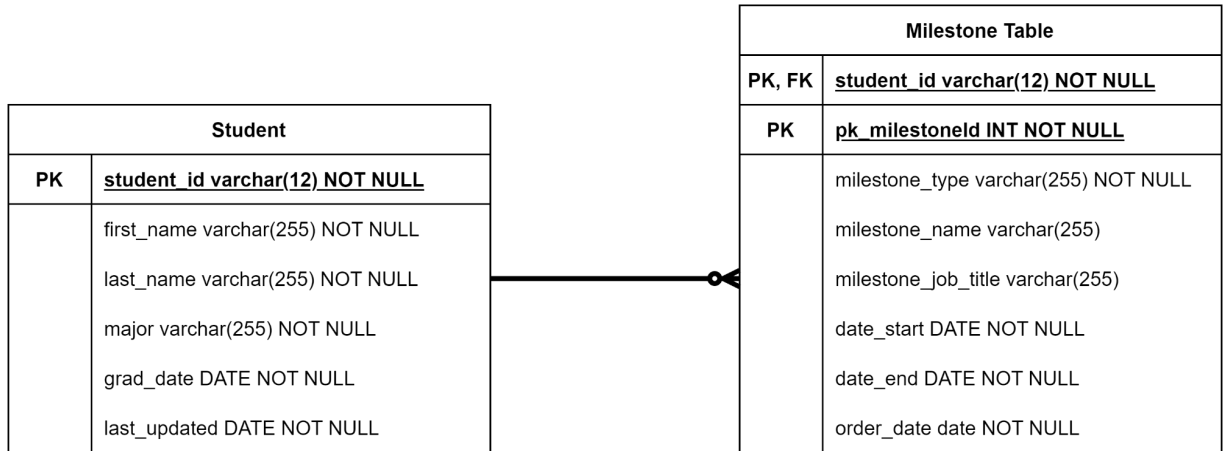


Figure 6: Database Student Milestone Tables

4.0 Usability Testing

The focus of usability testing is how well a user can use the product. This type of testing relies heavily on outside users to test the product to ensure that the page design, quality, and modularity fit to users as intended by the developers. To test for these qualities we will be using both quantitative and qualitative data, relying on outside user input from volunteers and our clients. Usability testing focuses primarily on the user interface of the application.

4.1 Web Page Design

This section will focus on the overall layout of the user interface and how users can navigate through it intuitively. User interaction on this page will determine what needs to be changed to better the user experience. Excellent web page design can be determined by a number of things including, layout, color scheme, element descriptions, and a variety of other things. Web page design will be primarily tested by volunteers to collect qualitative data. Qualities of user usability and experience will be taken into consideration in this process.

4.2 Element Quality

Element quality is determined by how well the components that build up the front end hold up to user interaction. This section of testing is a mix of user interface testing through volunteers as well as testing by our team. Element quality focuses on not only the intended functionality of the page elements but also the unintended functionality. This could mean if the user uploads a document that is not a CSV, or if an element is repeatedly clicked on. These outcomes

need to be logged and accounted for in order to ensure that the application will not break.

4.3 Page Modularity

This section will focus on how this application reacts to different browser interfaces, as well as different window sizes. Currently, most browsers run on Google Chrome's operating system. Even if this is the case there are some issues that could arise if a user decides to use a browser that turns out to be incompatible with our application. Page modularity testing ensures that the application will be able to maintain its intended formatting no matter what system it is run on. Testing for this section will be completed by our team on multiple browsers. The browsers that will be tested include Google Chrome, Microsoft Edge, Opera, and Firefox. By making sure that our application can safely and efficiently run on these browsers, we can eliminate any usability issues for our users down the line.

5.0 Conclusion

The CareerNet web application has been carefully designed to ensure the most efficient resource users and the fastest user experience. The front-end system handles the interaction with the user very well. The Login System allows users to access information after they have logged in. The CSV Upload module allows users to submit two different types of CSVs according to the user manual, one CSV for uploading new student data and the other allowing users to upload student milestone data. Administrator Settings only allow access to users with administrative privileges, which allows administrators to manage access to user information to a great extent. The backend system handles all the logic nicely, scraping the data the user needs from the LinkedIn website and passing the user input to the database. The database system will store all user and student data, linking the data with the front-end and back-end systems so that users can then retrieve the data they need. By using these key technologies, we will be able to deliver different student data for different types of users. Ordinary faculty users will know the data of the students they need to meet their research needs. At the same time, the administrator can also approve a common faculty user to access the student data of multiple majors and can compare the data. Administrators can control all accessible permissions, as well as access to all data. Overall, the design of the CareerNet application ensures that our user will get a solid solution to the current problem: the lack of any simple visualization of student career data. The technologies we chose will help our clients observe data on students' careers in a reliable and simple way.