|  |
| --- |
| Internship & Fieldwork Application |

|  |
| --- |
|  |

|  |  |
| --- | --- |
| Seraphim Dmitrieff  Project Type: Project  Class Key: s19-09 |  |

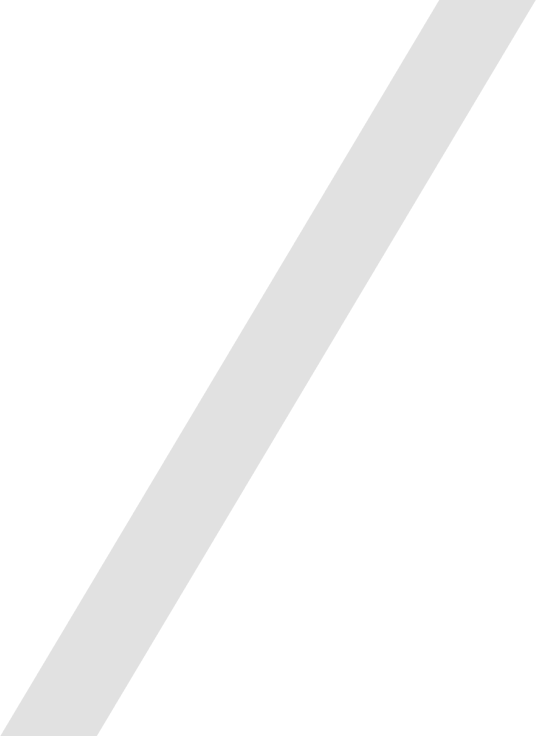


TABLE OF cONTENTS

1. Problem Description

*1.1 Business Context and Goals ……………….……….…………….……. page 03*

*1.2 Technical Requirements .……………………………….……….………. page 03*

*1.3 Your Responsibilities …….……………………………….……….………. page 04*

2. Technologies

*2.1 Related Technologies ….…………………….……………………..…….. page 05*

*2.2 Newly Learned Skills/Technologies ….……………..………………. page 06*

3. Design

*3.1 System Architecture ……………………………………….………………. page 07*

*3.2 Components ..…….…………………………………………..………………. page 12*

4. Software/System Description *………………….……………..……..…………. page 13*

5. Test Results/Observations

*5.1 Experiment/Observation # 1 (Performance) ....………………. page 15*

*5.2 Experiment/Observation # 2 (Quality) ..….……..………………. page 17*

6. Professional and Career Benefits *……..……..…..…….……..……………. page 18*

7. Conclusions *…………….……..…………………………...…………..………………. page 19*

8. Appendices ……………….……………………………………………..………………. *page 20*

|  |
| --- |
|  |

## 1.1 Business Context & Goals:

The Career Resource Center currently is responsible for managing student fieldwork and internship applications for academic credit that currently are only offered in a paper format. The form is responsible for providing an academic context for real-world experience and is utilized to offer college credit for work done in a professional environment. The form currently must be completed or approved by the student, their faculty instructor, the department chair as well as the dean of the school, following this process the paperwork must be compiled and submitted to Records and Registration to allow the student to be approved to receive academic credit for their fieldwork experience.

In the past the Career Resource Center has attempted to adopt web-based software to help manage the various users as well as form data, but found the annual subscription cost of over $1,500 as well as its lack of customization to meet their specific needs has resulted in them requesting our services with designing this new system. The Career Resource Center would like to include the Employer the student will be working for during their Fieldwork experience to be included in the application process so they may be able to view the prospective academic learning objectives as well as offer feedback to help better focus these objectives.

## 1.2 Technical Requirements:

To implement our design we will need to host our code on a server, for development we will be using the Computer Science Departments Wyvern Apache server. The benefit in this is there is no cost for web hosting as we have access as students and we are able to access the system remotely using PuTTy using SSH or WinSCP using FTP to be able to update and manage files regardless of location.

Beyond the server, we need to be able to manage user access using a login system that will query our database as well as manage user data between pages using PHP sessions. HTTP requests do not pass information unless we utilize a GET or POST request, which we will not always be able to use, as we want to keep our front-end form data separate from our backend MySQL query logic. We want this distinction for two purposes, to better organize our code as well as mask our database queries to increase the security of our system.

To make sure MySQL queries are submitted correctly and the database integrity is maintained we implement some basic form controls to ensure that we are not duplicating data, using Javascript DOM manipulation to dynamically update form elements based upon user interaction. An example of this includes select dropdown boxes used for our sequencing code that is used to set and display the order of the webpages in our workflow. In addition to regular Javascript we used AJAX, which is an asynchronous implementation of Javascript to dynamically build pages, loading in files that query the database separately and display its contents by setting a destination for the file code. This allowed us to simplify the design as we were able to build webpages that may be utilized by different user types that need to be able to display different content based upon the user.

With this web-based approach to our system as a user they would only require a device with internet access and a browser capable of running Javascript as this is managed through the browser, all our PHP logic is handled by the server operating separately from the frontend. We used the currently available Bootstrap CSS to enhance the responsiveness of the site to make it mobile user friendly to allow for increased access and ease of use if a desktop computer is currently unavailable.

## 1.3 Your Responsibilities:

As a group we were tasks with using the existing system previously developed by a project group last semester and finish the fieldwork/internship application based upon the specifications provided by our client from the Career Resources Center, Beth King. Beth currently coordinates the application process and is requesting the electronic form to be able to better manage the process as well as increase the accessibility to the users, Students, Faculty and Employers.

My contribution to this project was to design and implement the backend system that takes the front end form data and insert it into the database or update the existing information if it already exists. Once the query is completed the user is then redirected to the next webpage in the application sequence until they complete their portion and it is assigned to the next user. In addition to the design and coding I was tasked with coordinating meetings with our client to check in and get feedback as we progressed through the development process. This allowed us to be able to produce a better finished product, more aligned with our client’s needs.

## 2.1 Related Technologies:

We are designed our system to simulate a typically application that a user would submit, similar to an application filed to apply for school or a job posting online. Our goal was to simplify data entry for the user to allow for quicker processing as well as prevent system issues when interacting with the database. To implement our design we used:

* *The Wyvern Apache Server* – This will be responsible for hosting our website during development. The server comes equipped with MySQL as well as PHP support. The runtime environment is Linux based, which I have past experience with in other projects courses. This server is easily accessible remotely using PuTTy, to allow for command line interactions with the file directories as well as access the MySQL server and manage those features. WinSCP allowed for more user friendly file transfer to and from the server to allow for quick changes to be made, allowing for increased testing of our coding logic.
* *PHP* – Our webpages, the components that comprise them as well as the logic that manages data and access are all based in PHP files stored and hosted on our Server. PHP will also allow us to manage data between webpages by allowing us to securely store user information in Sessions, which will also provide more ease of accessibility if the user were to leave the page and want to return. As a preprocessor we are able to manipulate the page displays using logical statements to allow for better code utility and avoid redundancies in coding separate pages for each user type.
* *MySQL –* This is software we are using to implement our database design, it is already available on our server and partners well with PHP to provide ease with connectivity as well as allowing us to manage and display user information by make queries to our database. This is the essential piece with our website utility, acting as a storage system it allows us to house many different data components and its quick responsiveness makes it a good choice to use for web-based functionality,
* *HTML –* Not only is PHP usual with adding MySQL functionality it also allows us to implement HTML forms within its code to allow for us to pass data to a separate submission page that can be called upon as a POST method. HTML allows the form elements to display in the browser and is what users will be interacting with in our system primarily.
* *Javascript –* Just as we are able to include HTML within our PHP files, we are also able to include Javascript that will help primarily improve the look and feel of our front-end. Javascript is run through the web browser and can interact with our HTML code using DOM functions. This allows us to better manage data input as well as ease resources on our server by running the code client side. In addition to regular Javascript we also utilized AJAX in our code, which stands for Asynchronous Javascript and XML, that allowed for queries and coding that may take long than the page load to be able to be utilized in our front end. The asynchronicity of this code allows for the remaining page code to execute while the backend file accesses the database and builds the forms based on the data available from the database query.

## 2.2 Newly Learned Skills/Technologies:

I have previously used MySQL in an academic capacity to study SQL coding, but this project demands a much more involved implementation as it is intended to be used in a production capacity by our client, the Career Resource Center. The database will be required to manage information for all 7 different types of users (student, instructor, employer, department secretary, department chair, the dean of the school, and the Career Resource Center) as well as all the associated form data. To help reduce overhead with querying information we have to implement a series of smaller tables that help store our data, but keep the information of each table relevant to the task it is meant to be used for. Towards the end of our implementation we were tasked with creating a more intelligent system by adding sequence control into the database using a series of tables to determine which page the user is redirected to following a successful query.

Aside from designing a more complex relational database, PHP was a completely new language that I began using for this project implementation. I have done previous web development using NodeJS, and VueJS to create an API to pass data from the front to backend systems, PHP simplifies that system by being able to be used for both systems and allows us to be able to use different languages in the same page as well as added the ability to query the database and pass the results into our HTML and Javascript codes. In its capacity it acts as a preprocessor, which means that its code is implemented prior to the page load. The benefit to this system is that we are able to display different information on the same page using Boolean logic to set parameters for the desired code we wish to output.

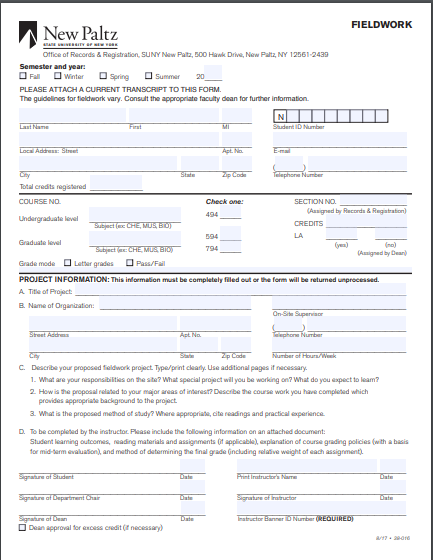
I have used Javascript before for web design so implementing it for front end form manipulation was not what I would consider as newly learned. By adding AJAX request code into our system I was able to build the frontend forms as more like components to allow for better code reuse and avoiding needing my query logic to be handled on the frontend. Using Javascript became useful to enhance the features of our website allowing for a better user experience, while also aiding in helping control the form data that would be eventually submitted into our database.

Outside of the new coding languages and skills I acquired through this project, I was also exposed to working for a client instead of in an educational capacity. The challenge with this was working with individuals who have set expectations, but do not necessarily have the knowledge of coding and implementation. Despite not being able to provide technical feedback, we were able to demonstrate our project and get feedback from a different perspective and were able to make adjustments based on that feedback.

## 3.1 System Architecture:

To first discuss the architecture we must first analyze the actual flow of information as knowing this process is integral in designing and implementing our project. We started by looking at the existing paper form that is currently being utilized and analyzed its contents and discussed with our client where the current limitations reside. The form is only one sheet that contains regions to only enter in basic information as well as show approval signatures for the various faculty members associated with it. Three main limitations were observed, the responsible parties for certain form fields were intermingled with other elements that were completed by another user, despite being noted on the form it can lead to confusion and would needed to be edited resulting in a slowdown in its processing. The other is the lack of space for required information from both the student and the instructor, with the student needing to answer questions meant to describe the academic potential of their requested project, as well as the grading criteria and syllabus to be included by the instructor. Lastly, the form has to be passed physically from one user to the next, which if we factor in the additional pages of documentation that would need to be included as just described, it becomes subject to information being lost and potentially delaying the approval process.

Figure 1: Paper Fieldwork Application:



With these limitations in mind we then worked with the existing information present from the previous group as well as through meetings with our client and instructor we determined the desired flow of information. Below is the requested flow of information our client was requesting for the electronic form system we were tasked with building:

1. The Student will request a new form application from the department secretary
2. The Department secretary will enter student’s id, with the id we can determine if the student already existing in our system, if they do not the secretary will be tasked with adding the basic user information to set up their account. This information is added to the database to allow the student to access their profile.
   1. Once a student account is existing the secretary will authorize a new fieldwork application to be assigned to the student. The faculty instructor as well as the associated academic department are added to the application here.
3. The student completes their profile information, if they are a first time user as well as completes all relevant sections of their fieldwork application and add the employer information. The application is updated in the database and assigned to the faculty instructor for further review.
   1. The Employer is sent an email to verify access into our system, they will be allowed to create an account, but will be unable to view any additional information until the Employer is verified by the faculty instructor.
4. The Faculty instructor adds relevant course evaluation material as well as learning outcomes/objectives. Once they approve the application the database is updated and the application is assigned to the employer for further review.
5. The Employer reviews the learning objectives and adds comments if applicable, they are presented with an approve or deny option, any deny submission will require comments to be entered. Once approved the database is updated and the application is assigned to the Department Chair for Review.
6. The Department Chair will review the application in its entirety as opposed to exclusively the learning objectives that are viewable to the Employer. If approved the application is updated in the database and assigned to the dean, if denied the application is referred back to the faculty instructor.
7. The Dean will approve/deny the application in a similar fashion to the Department Chair’s role in Step 6, however, they will be presented with additional options for Excess Academic Credits as well as a Liberal Arts Options as these currently exist on the Fieldwork application. If denied the application will be referred back to the faculty instructor who will then have to update and send back to the Department Chair once completed.
8. Following all faculty approvals the CRC is able to review all the application contents, export all the data into a file and submit the request to Records and Registration.

To organize this information we created a flow chart to outline the path of our data flow as well as better illustrate the approval and denial flow of the application we designed in our system (See Appendix A: Fieldwork Application Flow).

Once we had an established structure we designed the forms to be displayed on the front end, breaking down this information by user and with each user we separated the information by the sections its located in the paper application. An example of this is for the student we had four forms, one for the course information (The top field on the paper application in Figure 1), the Project Information (The text fields located on the paper formed labelled as Project Information, only including the elements that directly pertain to the student), the Description of the Proposed Fieldwork (Shown in Part C of the application on Figure 1) and Lastly entering the employer information, that will later be used to enter these individuals into our application system.

With each form page we created an associated database table to house the information linking each table using a form id the database auto-assigns upon the creation of the application. The structure of our database is shown in Appendix B: Database Table Design. Once the data is updated in the table, we then redirect the user to the next form page that needs to be completed, repeating this process until all required elements are completed and the system then assigns it to the next user. We built the system using a generic linear flow hardcoded into the system as illustrated in Figure 2.

Figure 2: Linear Data Flow:

5. Assign Application to Next User

4. Next Front end form, repeat process 2-4 until, all forms completed for given user

1. Start of Application

3. Back-end PHP file that updates the associated database table. Then redirects to the next front end page

2. Front-end form to be completed by the user

We were successful in designing and implementing the linear design flow of the application pages to follow the flow of data we previously described. We had the opportunity to demonstrate the code for members of Records & Registration, the Digital Communications Department and the Career Resource center and received the feedback that we needed to allow for more customization of the application flow, as some departments do not handle the application in the generic flow we outline previously as well as show in Appendix A: Fieldwork Application Flow. To accommodate this request, we had to redesign the system from the linear process shown in Figure 2 and include a backend file to act as a controller to produce an intelligent workflow design (Figure 3).

Figure 3: Intelligent Workflow Design



The controller is tasked with identifying the current step in the process, to do this we use a keyword tag with each frontend page in the workflow which matches an associated column name in our database table that orders pages by value. The system the increments the returned value and builds a second database query that determines the next page by order returning it as the same keyword tag we have associated with each frontend webpage. The returned value is then compared against an array stored on the server that uses the returned value as a key and returns the actual link to then redirect the user. To implement this we needed to adjust the existing code by removing the hardcoded links as described and shown in the linear design in Figure 2 and instead have each backend page redirect to the controller file passing it the keyword tag as a parameter, which then performs the actions previously described and redirects the user to the correct frontend page.

Lastly our system is structured to have admin’s manage the system, with our system this designation will be given to our client who can then create and assign new admin users. The admins will be allowed to create new users of all types as well as create and edit academic department information designating the users that will act as the Department Chair and Dean. The admin is able to edit the workflow order, specifying which type of user is assigned which elements of the application. The admin is also given the ability modify the editing privileges of each user type based on the resulting academic department the application is assigned to. The admin is also able to force approvals for a specified application through in case there is an issue with the workflow or a user is unavailable.

## 3.2 Components:

We implemented our system architecture as a group of components to help in development by reducing redundant coding in the system. First, we have several HTML components that was included with every frontend page, it provides the head and body tags as well as including our CSS on every page to allow for uniformity in our code as well as with the user interface. These files are named ‘skeleton.head’ and ‘skeleton.foot’ and should be included in any frontend form as it also loads our navbar and debug coding when in debug mode is set to on. Every frontend page consists of an HTML form that redirects to a backend PHP page that connects to the database and updates the associated table. With the frontend PHP pages we include coding to check user permissions to see if they are able to edit existing form information as well as checking if the user has the ability to generate emails. The permissions are designated by our system admin and when updated will add/remove edit options on the associated user based on the admin set permissions.

Inside our backend PHP pages we have several components that are integral to the functionality of the system. Essential to each PHP file we needed to start/resume the web session. The session allows us to retain data between webpages without the need for making the request using a GET or POST method in our HTML. Storing data in a session allows us to manipulate the Boolean logic in our PHP code to allow for dynamic user displays based upon the session data. The system initiates the session following a user login, but each PHP needs to resume the existing session otherwise the data is inaccessible. Once the session is set, we call the code to connect to our database named ‘db\_conn2’ which contains the access credentials to log into the database as well as sets a connection. We also include PHP file that houses common PHP functions we reuse throughout the workflow including setting banner messaging as well as email sending functionality. This file is named ‘util.php’. Once these files are included with each page we can then create our database query and update the associated tables. Once the query is completed the backend page redirects to our sequence controller code named ‘sequence-controller’ which then determines the next path the user will be redirected to.

The database is one of our system’s most integral components, housing all data entered by all system users as well as containing the logic values to allow our controller program to work, as well as storing the edit permissions the admin can assign to each user. The database is what gives our system its functionality, without it present we would only be able to pass data between webpages in a limited fashion and once the system is exited all the stored data would be lost.

## 4. Software/System Description:

This system is accessible by 8 different user types described below:

1. Secretary – able to search the system to see if a student already exists in our database, if not they are able to create a new student user. Following this search process they are able to assign the student a new application, assigning the faculty advisor (designated as instructor in our system), and the resulting academic department.
2. Student – This user initiates the application process, completing all admin specified pages. Our default flow has them enter the course and project information sections of the current paper application (Figure 1), as well as enter their employer information which generates a profile for an employer but not a user login.
3. Instructor – Reviews student entries, including employer information. If they confirm the employer information portion this will create a user entry into our database for the employer which will send them access credentials. The instructor in the default flow also adds learning objectives to the application and uploads the course grading documentation (outlined in part D of Project Information in Figure 1).
4. Employer – Reviews Project Information from the student and learning objectives from the instructor, with the ability to approve or deny the application.
5. Chair – Reviews all application entries and is given the ability to edit the application by default which can be updated by the admin.
6. Dean – Acts similarly to the chair in practice but has an additional form for Liberal Arts and Excess Credit designations.
7. Admin – manages users, can update permissions, manage academic departments, updating the associated chair and dean user. They can manage permissions for editing forms and email generation specifying these settings by academic department instead of by user type as this allows for increased customization of our system. They are also able to specify the application sequence to adjust the order if need be or even remove specific user types from the process as specific academic departments have different processes for handling Fieldwork/Internships.
8. RecReg – This user reviews the final form and confirms the completion of the application process.

When attempting to access our system, currently with the link:

<https://cs.newpaltz.edu/p/f18-02/s19-v1/v4>

If a session is not currently set the user will be redirected to a login screen. Upon logging in the system identifies by checking the database if the user is accessing the system for the first time and if so will redirect them through registration that is comprised of a password reset and profile creation. The first time user will reset the system generated password, following this they will be asked to enter personal information. This information will be used as their profile, which we utilize elsewhere in our system to display contact information as well a using their names to assist with data entry. If the user is not a first time user, or just completed the registration process previously described, they are directed to a home page which if the user is a student will show all their applications but only makes ones specifically assigned to them accessible as a link that will allow them to edit that particular application. If the user is of any other designation they are only able to view applications currently assigned to them in the process.

The initial redirect from the home page passes the fieldwork form id as a get request which then sets the associated session variables. This was hardcoded into our webpages when we were working with our linear system (described in Figure 2), however with the Intelligent workflow design we need to include it in all pages that are not the home page in order to allow each page to be specified as the first page in the workflow sequence for a given user. We do this by checking if the session variables are present using the PHP method isset() checking for the ‘fw\_id’ key in particular and if its not present it sets the session using the GET data from the redirect of the homepage.

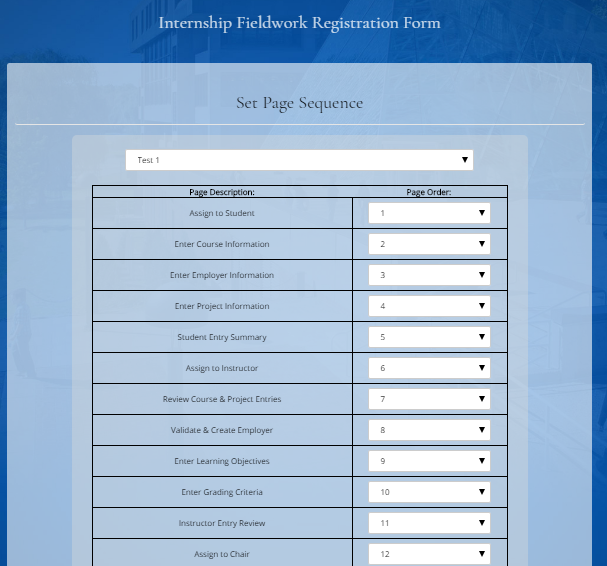
The user then completes each form in their assigned sequence until the sequence controller detects that the next redirect is back to the home page. This indicated the application sequence designated to the user has been completed. The page then redirects to a submit page that checks the database for the next user assigned in the sequence and assigns the application to them and redirects the user back to their homepage, which loads the applications as previously described.

Users of the type secretary, admin are excluded from the flow shown above, instead they have their own designated landing pages, the secretary is directed to a search form where they can carry out the search and application creation process described above. The admin is directed to a gui where they are able to manage the system settings and perform the tasks previously described. The admin and secretary redirects are unable to be modified as we are able to do so with the application flow.

## 5.1 Experiment/Observation # 1 (Performance):

Because of its importance with controlling how the system will work I based the performance of the system on the sequence controller we built. To test the controller we developed two admin panels to manage the sequence. The first sets the user sequence and the second controller the order of pages (See Figure 4: Admin Set Page Sequence).

Figure 4: Admin Set Page Sequence



This tool filters by the existing academic departments in the system and allows the user to update the order of pages visited and excludes the user specific pages based on the selected user sequence. With this page we used select drop downs that default to the order of pages initial sequence. Using Javascript we were able to prevent overlapping page sequence numbers which will result in our sequence controller erroring because page values will override one another and cause errors in the system.

With this portion tested we removed the employer type from our sequence and updated the page sequence following that. Currently our design can allow for the user sequence to be updated without needing to update the page sequence as when the next assignment triggers it will be assigned to the correct user and when that user accesses the application the sequence controller will skip over the unvisited pages as it will resume using the correct page key value (the name of each front end user page, used in the sequence controller). Once we updated the sequence, we created a new test application and proceeded through the workflow, completing the student and instructor portions and once complete the application assigned to the correct user, which is the department chair.

Following checking the user removal from the user sequence I adjusted the page sequence for the same user type and tested the transition from front end page to front end page. I was able to reverse the sequence of pages for the student, leaving only the final review page in its correct order I was able to verify the information was passed successfully through the application, verifying the sequence controller is effectively responsive.

## 5.2 Experiment/Observation # 2 (Quality):

For quality observations I chose to focus on certain bugs I’ve noticed in my testing that I was unable to fully address during this semester:

1. We disabled most of our security coding for the system excluding the admin tools, which essentially operated by limiting access to pages based upon the user’s type designation. The primary reason for this is that this aspect will be the focus of the project pertaining to this in the upcoming semester.
2. We were able to fully implement form control, excluding special characters in the text areas that would reduce the potential for SQL injection into our database. Also, we allow for form inputs to be inserted as blank values, which can result in errors with submitting the application. The workflow should still work, however the user experience will be diminished as the missing data can delay the application process.
3. Following submitting the application to the next or previous user based upon the accept/reject options respectively the user can select the back button in the browser and therefore will be allowed to change their submission despite not technically having access to that application anymore.

These specific issues if not addressed can cause issues with data entry and can have a negative impact on the database. In terms of remaining features we were unable to fully implement include:

1. Output all Fieldwork Application data into a pdf format, we did have the ability to write the information into a file that could be downloaded from our server, we felt that this form would be better handled by someone more skilled in graphic design to allow for the stylings to fit the guidelines and allow for dynamic data entry.
2. We were unable to add certain features into our Admin toolbar based on time restrictions. These include user impersonation to allow for applications to be forced to the next user as well as the ability to update email text generation for our system.

## 6. Professional and Career Benefits:

Despite the academic nature of the projects course, this specific experience has been invaluable in providing me with project management experience. I did not have the opportunity to purse and internship during my academic studies and was seeking to branch out from the more structured projects course I have previously taken. We were given some creative freedoms with the design of the system following the constraints given to us by our client and the guidance of our instructor, Professor Pham. We were able to select our coding languages and supplied the resources we needed to build this system.

Having this course be more open ended required us to be more diligent with our own schedules, designating time to work on the project as well as arranging the meeting with our client ourselves, instead of having the tasks just assigned to us. Working with Danny we shared tasks between implementing the backend and frontend systems as well as testing and debugging the code allowing us to experience every portion of the development process instead of only being exposed to specific aspects of it. The open ended aspect also meant we needed to be able to research more how to code in specific language and integrate different languages together as we built our system. I had never used PHP before this project, but will continue to use it afterwards I really liked the flexibility it had with integrating different languages and systems together seamlessly.

We also had the opportunity to be able have our coding reviewed by members of the Communications Department, Records & Registration, and the Career Resource Center, where we were able to have our System, then a work in progress, scrutinized by different users and utilize that feedback to make adjustments to our system. The reason we implemented the Intelligent Workflow was a direct result of this meeting.

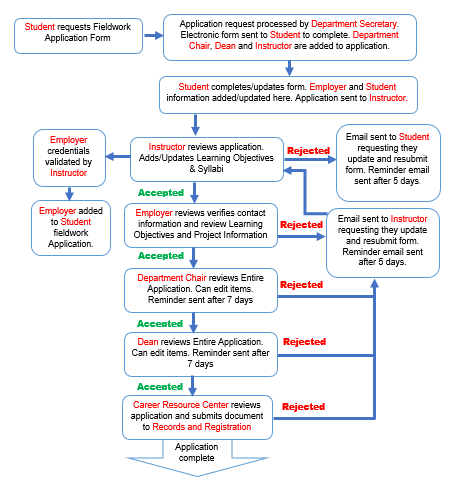
Implementing the sequence controller was a challenge in an of itself. It required us to taking our existing system and break the links that were hardcoded into them, redirecting to a controller file which then determines the next link in the process. The design of this system was challenging in that it required to determine how to dynamically load and update this information to allow for the customization and expansion of the finished product we are submitting to our client.

## 7. Conclusions:

This System when looking we first encountered it looking at the design seemed simple enough to implement, seemingly to be comprised of a series of forms that we chain together for the end result, but the resulting product was way more complex than I ever imagined. Making the system more responsive and intelligent was my biggest challenge to date, especially since I needed to dismantle the existing system we built and expand the existing database design even further

In making the more intelligent workflow we provided the opportunity to expand and customize the workflow even further than its current design to allow for additional customization of the system we built. I am proud to have had the opportunity to work on this project and hope that it will be fully implementable by the end of next semester when the next group to pick up this mantle work on the security of the system, which will need to include form validation as we were unable to fully implement this check and is vital to making sure the application process is maintained.

## 9. Appendix A – Fieldwork Application Flow:



## 9. Appendix B – Database Design:

## 9. Appendix C – User Guide:

The Career Resource Center is recognized by the system as an Administrative User (or admin).

After logging in, the Career Resource Center may:

* Search for a user by their e-mail address.
* Create a new user (of any type).
* Look up a Department.
* Create a new Department.

Some admin tools we plan on implementing later are:

* Modifying e-mail text.
* Modifying application sequence (by department).
* Editing any user’s profile or login information (or sending them an e-mail prompting them to reset their password).

Figure 2.0.1 shows a screenshot of their home page:



Figure 2.0.1: A screenshot of the tools available at the Career Resource Center’s disposal.

## Creating a New User as the Career Resource Center

Creating a new user is fairly straightforward as the Career Resource Center.

Once you’re logged in, click the “Create” button under User Tools (shown with a Notepad emoji 📝). Once you’ve done that, you’ll be prompted to enter some information about the user, including their e-mail address, their Banner ID, their initial password and a confirmation of that password, and that user’s User Type. This user type can be any recognized type of user in the system including: Student, Secretary, Employer, Instructor, Department Chair, Dean, Administrator, or Records and Registration. Figure 2.1.1 shows a screenshot of the dialog shown.

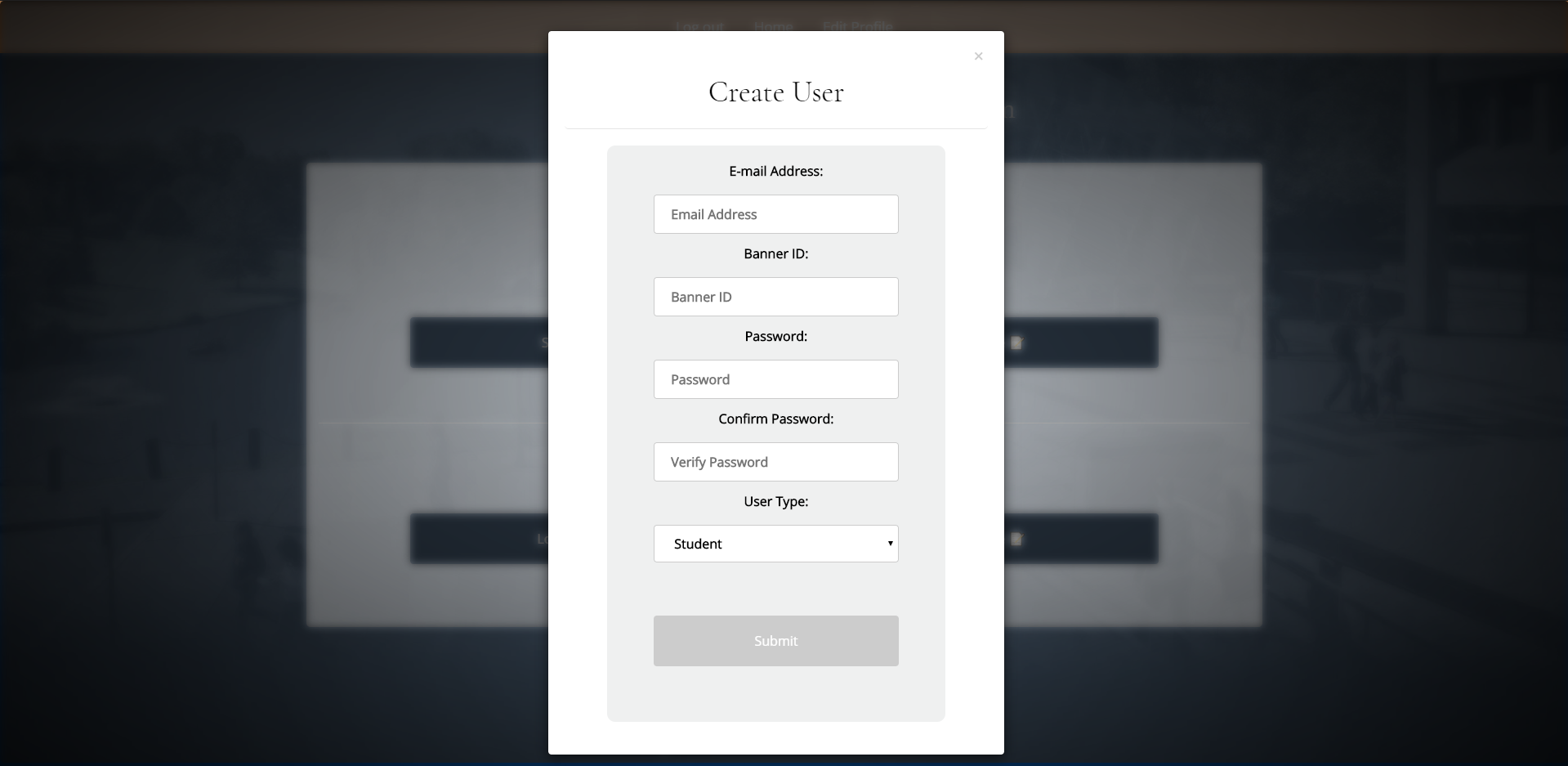


Figure 2.1.1: A screenshot showing the modal dialog to create a new user.

If you know a user’s email, you may look it up, and then enter it into the dialog box presented when you click on “Look Up 🔎”. When you do, you’ll be presented with the user’s information as another modal dialog, as shown in Figure 2.2.1.

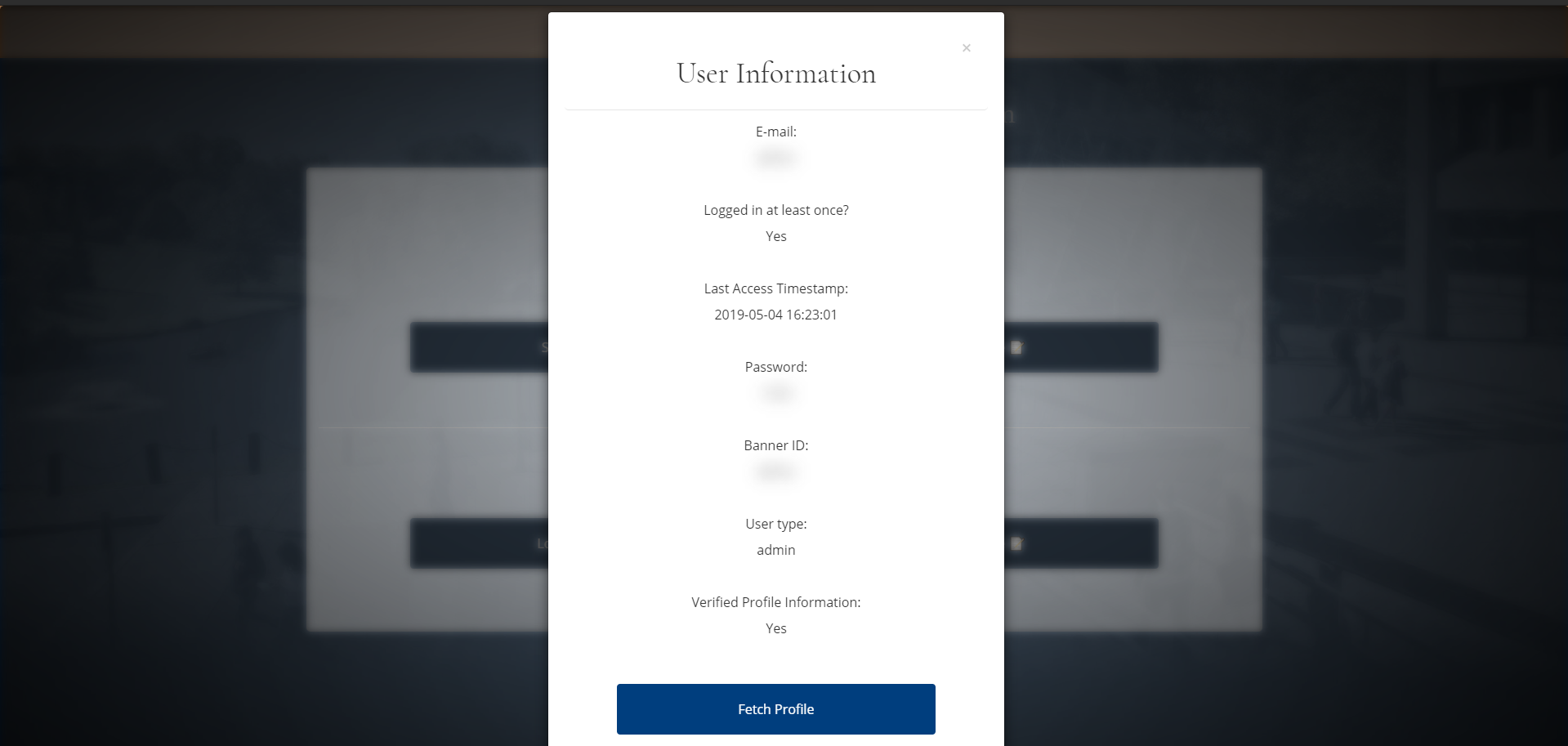


Figure 2.2.1: User Information presented as a modal dialog box. Actual information is shown blurred in the screenshot.

Upon clicking “Fetch Profile,” you’ll find more information on that person, including their name and other details, depending on their user type.

To delete a user, look up the user first. When you see the dialog shown in Figure 2.2.1, scroll down to the bottom of the modal dialog. You will see a button that looks similar to the button shown in Figure 2.3.1.

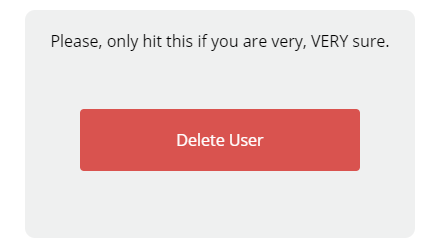


Figure 2.3.1: The “Delete User” button.

## “Edit Permissions” of a Department

As the text above the button shows, please only hit the button if you are very, VERY sure, as you won’t be able to recover any of that user’s information afterwards, including their user profile and any applications they might have filed.

## Creation of a Department

To create a department, log in, and, under “Department Tools,” click “Create 📝". This will bring up a modal dialog box asking you to specify the name of the department, a department code, and there should be drop-down boxes allowing you to select a chair and dean of the department by their e-mail addresses.

Note: The dean and chair of the newly created department should already have profiles before you start this process. You will need to start by creating their profiles before you can complete any tasks. To create profiles for them, see “2.1 - Creating a User as the Career Resource Center.” Once the department is created, the next time the new chair and dean log in, they’ll be prompted to update their profile information. When they update their information, they should include that newly created department.

At creation time, you may also specify permissions for staff to edit certain portions of the application’s information (See Section 2.5). You may also specify preferences for the department as for which events certain staff will receive e-mails (See Section 2.6). Note that information entered here can be edited later on after looking up the department.

Edit Permissions specify which items certain staff members may edit of an application. These may be modified after the department has been created, or specified with the department’s creation.

A member of the Career Resource Center may edit whether or not these statements hold true for certain departments:

* Instructor may modify course information.
* Instructor may modify project information.
* Instructor may modify employer information.
* Employer may modify project information.
* Employer may edit instructor submitted learning objectives.
* Department chair may edit course information.
* Department chair may edit project information.
* Department chair may edit employer information.
* Department chair may edit learning objectives.
* Dean can modify course information.
* Dean can modify project information.
* Dean can modify learning objectives.

Much like editing permissions, e-mail preferences are set at department creation time and can be modified later.

E-mail preferences can be set by type of user (student, instructor, dean, chair, employer) and can be set to notify if they get rejection e-mails, general updates, and reminder e-mails.

Students get reminder e-mails after 5 days of inactivity, and each other type of user gets reminders after 7 days of inactivity. Reminder e-mails will be sent every day until they log back into the system.

## Future Admin Tool Plans

At the time of writing this user guide, we’ve yet to implement a front end admin control for sequence control modification. This should be set at department creation time and can be modifiable later.

We also plan on including an administrative tool to edit the contents of e-mails to allow the Career Resource Center full control of wording.

The Career Resource Center should also be able to see all applications in the system, and delete/modify applications at will for any department.

The Career Resource Center should also be able to act as any user.

# New Users

A new user of any type will be prompted to do two things upon logging in:

1. Submit a new password into the system.
2. Update their User Profile with some personal information. For students, this will include their address, and for faculty members, this will include their office hours and departments. For employers, this will include the name of their business and some information on that.

Once they’ve logged in, any user may see applications assigned to them (besides Department Secretaries, the Career Resource Center, and Records and Registration). Students will see applications *they’ve created* instead, along with a progress bar and where it’s now assigned.

Your information may be edited at any time.

If you forget your password, click the “Forgot Password?” link on the login page. You’ll be prompted for a Banner ID number, and you’ll receive a newly randomly generated password via e-mail that you reset once you log in.

Employers: Since you won’t have real Banner ID’s, you will need to use an E-id number.. You will receive an email with your E-id number when your account is generated (Example: E10). This ID is only used in the case of resetting your password, so please keep it in a safe place.

# Secretaries

Secretaries can create new student accounts and create new application workflows.

When you log in, you’ll be prompted to edit a student’s Banner ID number. Enter their Banner ID number *regardless of whether or not that student exists in the system.* If you’re prompted to enter their e-mail, continue to Section 4.1. Otherwise, skip to Section 4.2.

## Creation of a New Student Account

If you’ve entered in the student’s Banner ID number and you get a red message with “No User Found” and you are prompted to enter their e-mail, that means they do not yet have an account.

Enter the student’s e-mail in the box presented to you. This will make a new student user account with their Banner ID and e-mail address tied to with it, with a randomly generated password. *This password will be sent to their e-mail, and once you enter their e-mail, you’ll get a message prompting you to have them check their e-mail.* This is for that password.

That e-mail should look something like Figure 4.1.1. Note that it looks like it’s coming from the Career Resource Center - this is for security reasons.

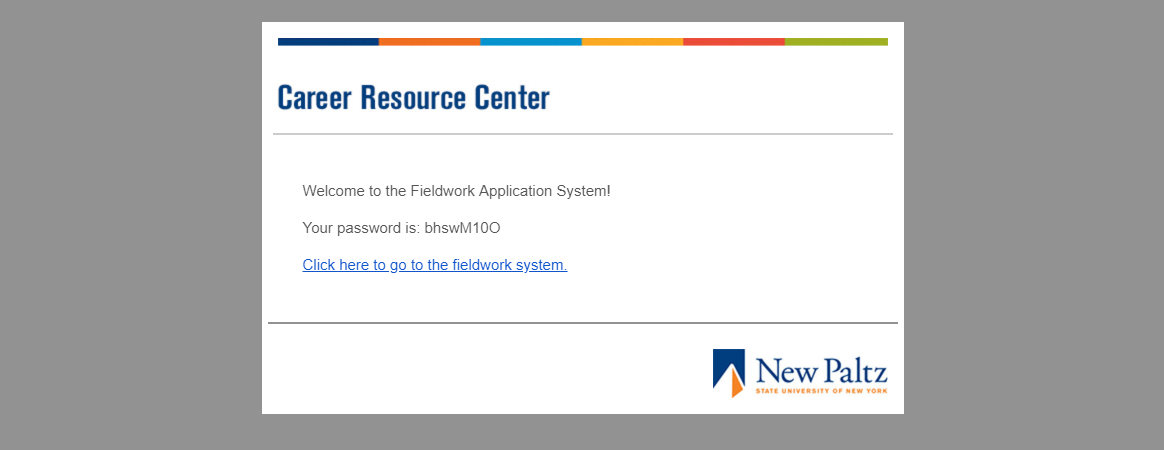


Figure 4.1.1: Students will get an e-mail with their password.

Once they get this e-mail, have that student log in and complete their profile. After that, enter in their Banner ID number again, and continue on assigning them an application with the instructions from Section 4.2.

## Creation of a New Application

If you get a green message saying “Found User!” and a page asking you to select a department, the student exists and you can assign them a new application.

First, select the department the student is going through for this piece of Fieldwork. For example, if the student is working through the Computer Science department, select “Computer Science.” If the department doesn’t yet exist, contact the Career Resource Center.

Next, select their professor within this department.

If there aren’t any professors in this selected department, the page may say “We Have a Problem.” This is a worrisome message that translates to, there’s a department without any professors in it. If this happens, contact the Career Resource Center, or a professor in the department with a profile.

Figure 4.2.1 shows an example of selecting a department and a professor within that department.



Figure 4.2.1: Selection of a department and a professor to assign the application to.

Once that is done, click the Submit button, and the application will be created. You’ll then be redirected back to the screen where the student enters in their Student ID.

# Students

To start the process of adding internship field work as course credit, please talk to your Department Secretary. Once you’re able to log in and you have a piece of field work on your profile assigned to you, log in and submit the information presented to you. This information can include, but is not limited to:

* Course Information
* Employer Information
* Project Information

Once you fill out what you need to fill out, you’ll have a chance to review your information that was submitted.

From there, you’ll be able to view progress on your application as a progress bar with a small description under it, and, if your department allows it, you’ll get updates when your application gets pushed onto the next person in your department.

Depending on your department and how it’s set up, you may also get e-mails if your application gets rejected at any point, and you may get reminder e-mails if it’s been 5 days since you had an application assigned to you for any reason.

If it gets rejected, the person that rejected it can leave a comment viewable next to the progress bar. Take the advice with the comment, and try again!

# Instructors

Once you get an application assigned to you, you must review it before it can proceed to the next user. If you do nothing for 7 days, depending on how your department has it set up, you’ll get a reminder e-mail reminding you to get on the system every day until you fill it out.

Depending on how your department has it set up, you’ll get up to five pages.

* Review of Application information.

When you review application information, you’ll get a chance to review Course Information and Project information. If your department allows it, you may edit this information if it’s a quick edit (say, a typo).

Otherwise, you may accept or reject the application at this point. If you reject it, you must provide an explanation.

* Review of Employer Information.

When you review employer information, you can either accept it, make a quick edit if your department allows that, or reject it (again, you’d have to provide an explanation).

* Enter Learning Objectives.

You’ll have to enter the learning objectives the student had to accomplish while working with the department. As you type these, you’ll have an opportunity to select learning objectives from the database if they exist in there already.

* Upload a Syllabus.

You can upload a syllabus that other staff may refer to when reviewing the student’s application. *In the future, we’d like instructors to be able to pick through syllabi that they or anyone else in the department has submitted before, and they’re named in the system after the department code.*

* Final Review.

You’ll get a chance to review your information that was submitted before sending it to the next person in your department. You may make edits on this page.

# Department Chairs and Deans

Department Chairs and Deans may get applications assigned to them, depending on their department. If they do, they will get a page reviewing the information that was submitted to them.

Department Chairs and Deans may:

* Edit the information, if the department gives them the permission to.
* Accept the information and submit the information to the next staff member.
* Reject the information, provide a reason, and send it back to the previous staff member.

If your account is inactive for seven days, depending on your e-mail preferences, you’ll get an e-mail reminding you to get on your profile, depending on how your department is set up.

You’ll additionally get e-mails for general updates and rejections, depending on your department’s e-mail preferences. To change these, speak with the Career Resource Center.

# Records and Registration

The page for the Records and Registration office has yet to be implemented. However, when the page does get implemented, they may view completed applications at any point in time.