

## 1. Problem & Solution

The Senior Project class is designed to test the prospective graduates' readiness to work in a real-world software development environment. During the first week, the students must find a project of interest and form a team.

The current system that is used to handle these initial stages of the Senior Project class is outdated and lacks automation. Version one of the website attempts to automate this process, but has some issues associated with it; thus, it must be improved before it can be deployed next semester (see "Current System/Previous Version").

More specifically, the problems that I seek to solve are:

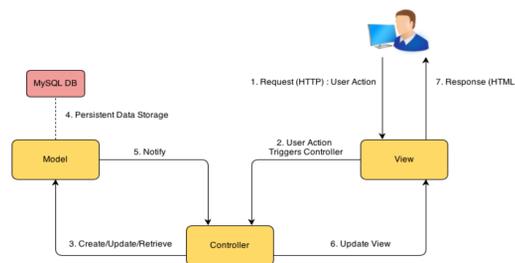
- The lack of admin functionalities for the Head Professor of the Senior Project course (see "Requirements" for details).
- The website's functionality is limited to solely meet the requirements imposed during the initial stages of the class. After the first couple of weeks of the semester, the website does not add any value to the Senior Project course.
- No support for uploading profile pictures from local storage.

### My Solution:

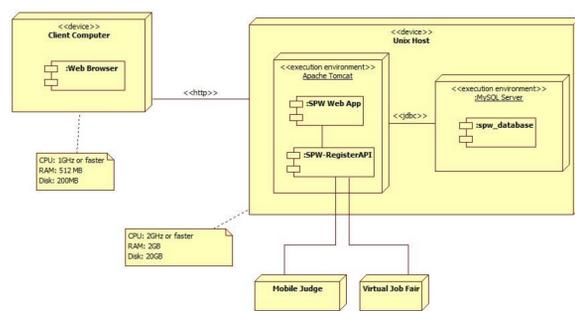
- Design an easy-to-use admin dashboard display for the Head Professor. This dashboard will make the privileged actions stipulated in the "Requirements" section accessible.
- Integrate a repository that will enable students to share files with their project team members and mentors throughout the duration of the semester. This will extend the usage of website beyond the initial stages of the course.
- Enable picture upload from local storage to edit user profiles.

## 4. System Design

The system was developed using the MVC architectural pattern.



### Deployment Diagram for SPW version 2



## 2. Current System / Previous Version

Currently, students enrolled in the Senior Project course at FIU SCIS use a Google Drive document to formalize their participation within a team project. Last semester, a group of Senior Project students identified the drawbacks of using the shared document to carry out this procedure. As a result, they developed version one of the Senior Project Website (SPWv.1) to automate this process. SPWv.1 implements some functionality that aims to resolve the problem, but there are some issues associated with the design that inhibit the system from actually being operational.

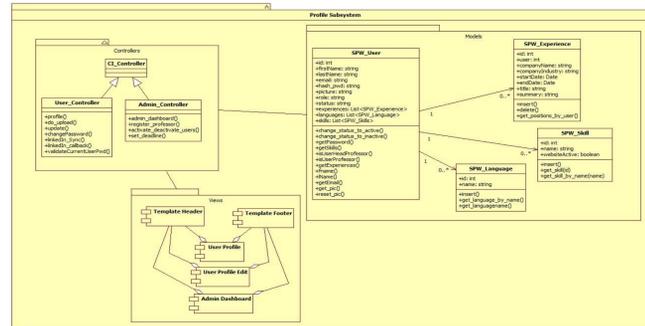
### Issues associated with SPWv.1

- All types of users have the ability to execute the same actions within website; there are no differences between actors.
- Users have the ability to change their role in the context of the system.
- The privileged functionality that was originally specified for the "Head Professor" and "Professor" roles are missing.
- System allows the registration/login of any user without verifying if they are Senior Project students or registered professors.

## 5. Object Design

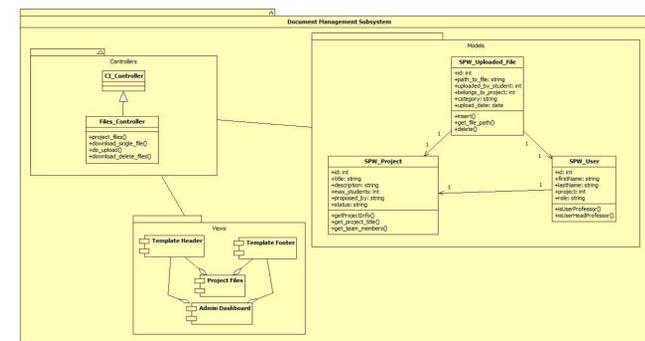
### Class Diagram for the Profile Subsystem:

A user profile is a visual representation of the attributes/relationships specified in the model package. Furthermore, if a user is a Head Professor then he/she has the ability to execute certain privileged functions that are handled by the admin controller. These privileged actions can be invoked from the admin dashboard view.



### Class Diagram for the Document Management Subsystem:

Every uploaded file is uploaded by a single user and belongs to a single project. Furthermore, if the user is a student then he/she is a member of a single project. The file controller is responsible for processing the upload/download/delete operations. The project files view will render the appropriate view of the files depending on the user's role. If user is a student, only the files for that particular student's project will be accessible. Otherwise, if the user is any type of professor, then all uploaded files will be displayed.



## 3. Requirements

- ✓ User profile edit must support profile picture upload from local storage.
- ✓ Incorporate a file sharing mechanism for project team members (Document management system for projects).
  - Need to support upload/download/delete operations.
- ✓ Admin dashboard panel for Head Professor must enable him/her to:
  - ✓ Set the join/leave/propose project time period.
  - ✓ Create professor accounts.
  - ✓ Activate/deactivate users.
  - ✓ Manage uploaded files on the server.

## 6. Implementation

Continued use of the same technologies used to develop SPWv.1:



### Implementation Details:

- ✓ Admin dashboard panel for the Head Professor:
  - ✓ Enables the setting of an internal deadline that will be used in the context of the system to determine the accessibility of certain functionalities (i.e. when students are allowed to join/leave/propose projects, when the file sharing functionality for the projects is accessible). Uses JQuery DatePicker.
  - ✓ Enables the creation of professor accounts. Email notifications are sent when accounts are created.
  - ✓ Enables the activation/deactivation of professor users. Inactive users are not able to login into the system.
  - ✓ Enables the management of all files that have been uploaded by students. The interface for managing the uploaded content on the server is integrated with download/deletion support.
- ✓ Document Management System for Projects:
  - ✓ The files repository is accessible by all SPW registered users. Those users can upload a file and associate it with a specific project. Every SPW registered user can download a shared file. The Head Professor has the ability to delete any file that has been uploaded to the server. All other users can only delete files that they have uploaded.
  - ✓ No restriction upon file types or file sizes when uploading.
  - ✓ Single file download at a time is supported.
  - ✓ Deletion of multiple files at a time is supported.
- ✓ User profiles have been modified to make a clear distinction between user roles.
  - ✓ Supports profile picture upload from local storage. Supported file types are: PNG | JPEG | JPG | GIF.

## 8. Verification

- Tested the environment with preset values.
- Executed the system use cases from the user interface (manual system testing).
- Unit Testing using CodeIgniter's Unit Testing Class.

The screenshot shows the results of unit tests for the SPW\_Uploaded\_File\_Model class. It lists test names, test datatypes, expected datatypes, results, file names, line numbers, and notes for each test case.

Fig 1: Set of CodeIgniter Unit Test Results for the SPW\_Uploaded\_File\_Model

## 7. Screenshots

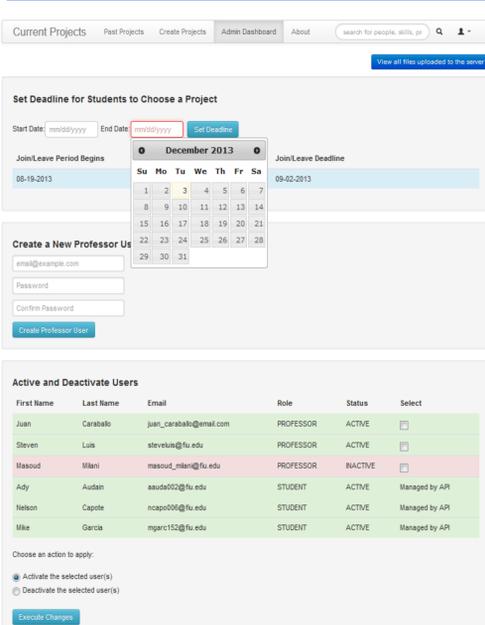


Fig 1: The admin dashboard display for the Head Professor.

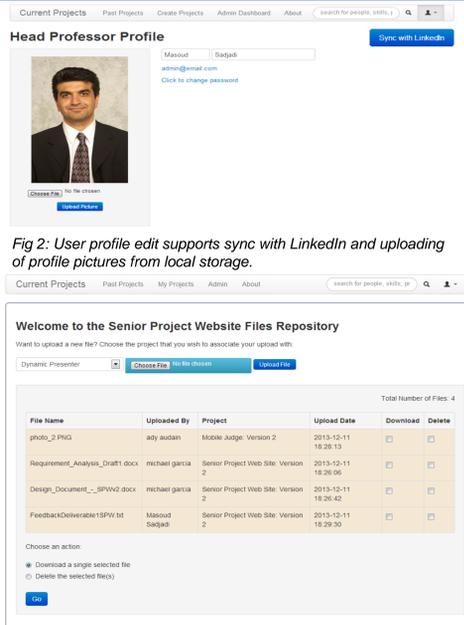


Fig 2: User profile edit supports sync with LinkedIn and uploading of profile pictures from local storage.

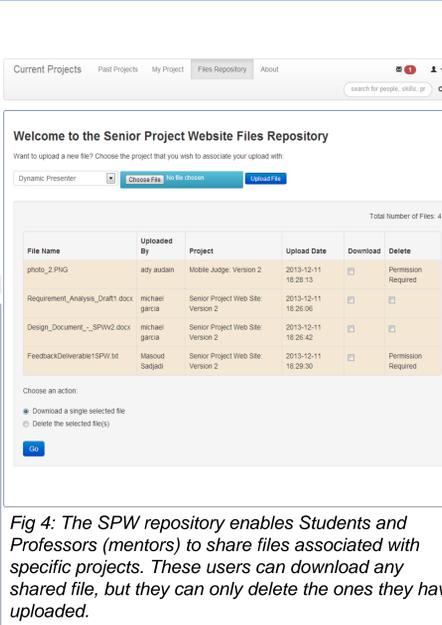


Fig 3: The Head Professor has the ability to download/delete all the files in the repository if he/she desires.

## 9. Summary

- ✓ Accomplished the objective to build upon/polish the existing version of the Senior Project Website.
- ✓ The new capabilities provided for admin make the site and the class itself more manageable for the Head Professor.
- ✓ Extended the usability of the system beyond the initial stages of the class by providing a central file sharing repository for SPW registered users, which can be used throughout the entire semester.
- ✓ Equipped the site with added conveniences to improve user experience.
- ✓ Solidified the meaning of different roles within the context of the system.
- ✓ User authentication through the SPW API improves the security of the site.
- ✓ Maintained the intuitive UI design established in version one of the website.
- ✓ Eliminated relations that jeopardized the sustainability of the website in a live production environment. (i.e. users could previously change their roles at will, users had multiple login options through popular social networks.)

## Acknowledgements:

I am thankful for the help that I received from my team: Nelson Capote and Antonio Vazquez.

I also want to thank Camilo Sanchez, Yaneli Fernandez, and Keiser Moya (the team that developed version one of this project). Their efforts on version one gave rise to a very strong foundation for the system.