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 find the team that will work with them on the most important project of their university career.

Students from last semester’s senior project class designed a website to automate this process. While this existing system is step in the right direction towards resolving the issue at hand, its implementation is not perfect and it is flawed from several key perspectives.

Our role this semester was to continue working on version 2 of the system to improve some of the existing functionalities and to add new functionality, with the goal of having a consistent system ready to be deployed for Spring 2014 semester.

My role on this project is to provide an API capable of providing information about students enrolled on the class that will help to control access to the system as well as to have updated information constantly.

The API was implemented in java using Representational State Transfer (REST) software architecture. It has a servlet class that runs a daily task to update the SPW database with the latest info.

Notifications System which includes: Join / Leave Project, Send Project for Approval, Accept/Reject project proposals now has emails notifications using PHP and CodeIgniter framework.

The API 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.

**Major issues associated with SPWv1**
- All types of users have the ability to execute the same actions within website, there is 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 is missing
- System allows to register/login any user without verifying information about current students enrolled on the class, from the API. Method getAll will allow Mobile Judge team to get all the students and the project they are working on.

The API returns a JSON object representing the PantherUserInfo class and it requires a Token to be consumed. Method SPWRegister : This is the portal for external applications to consume the REST service. Method getUserInfo is to validate if a student is part of the class. Method refresh will allow the Head Professor to get the most updated information about current students enrolled on the class.

<table>
<thead>
<tr>
<th><strong>Requirements</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Authenticate using FIU Panther Mail and Senior Project authentication system.</td>
</tr>
<tr>
<td>• Allow guest access.</td>
</tr>
<tr>
<td>• Create an API to gather information about FIU senior project students.</td>
</tr>
<tr>
<td>• Information includes: Full Name, Email, Panther ID and the Title of their project.</td>
</tr>
<tr>
<td>• Admin dashboard for Head Professor.</td>
</tr>
<tr>
<td>• Set join/leave deadline, create professor accounts, activate/deactivate users, update users from API, manage uploaded documents.</td>
</tr>
<tr>
<td>• Head professor to accept/reject projects</td>
</tr>
<tr>
<td>• Documents management system for projects.</td>
</tr>
<tr>
<td>• Create / Edit projects</td>
</tr>
<tr>
<td>• Browse user profiles and projects</td>
</tr>
<tr>
<td>• Internal messaging system integrated with email notification.</td>
</tr>
</tbody>
</table>

**System Design**

- The system was developed using the MVC architectural pattern.
- **Deployment Diagram**

**Object Design**

- **API Class Diagram**

**Implementation**

- **State Machine for Register/Login subsystem**

**Verification**

- Test Cases created for the API rest service using JUnit framework. No failures were reported.

**Screenshots**

- **Email Notifications**

- **API call for all students on class**

- **API call to validate a student**

**Summary**

- Website implemented using open source platform which allows the faculty staff to host and maintain the system with the resources and training they already have.
- Architectured using Model-View-Controller design pattern which enforces separation of concerns and therefore a clear, organized and granular code.
- Through the notification system every user keeps track of the latest changes in their team.
- The use of the API makes this site consistent on information and allows only access to those who need it.
- In addition it provides services to Mobile Judge and Virtual Job Fair senior projects.