



# FIU School of Computing & Information Sciences

## Senior Project, 2014, Spring

### Senior Project Website

**Student:** Christopher Kerrutt, Florida International University  
**Mentor:** Masoud Sadjadi, Florida International University  
**Instructor:** Masoud Sadjadi, Florida International University



## 1. Problem

The Senior Project course was created to test a potential graduates ability to work in a real software development environment. Potential graduates' success in this course will enable them to provide a significant software project that may help them land a job in the real-world.

The Senior Project Website coincides with this course for students to join a desired project. Now, the Head Professor would like to be able to recommend one project best suited for each student to join without having to inspect every projects' set of skills and every students' set of skills. These skills are provided by the Senior Project Website.

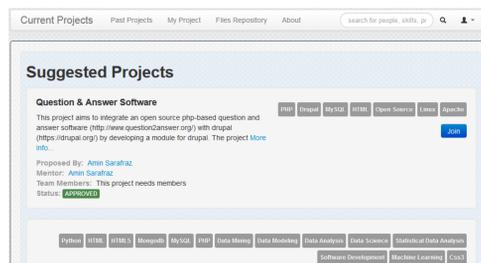
After inspection, the Head Professor would have to manually match or arrange students with projects as he or she sees fit. Doing so would be very time consuming.

## 2. Current System

The system supports project recommendations to students, but this would not help the Head Professor make a recommendation for one best project to a student, as the Head Professor does not have access to these project recommendations to students.

The current feature does not:

- o involve the Head Professor in the project selection process.
- o provide a single best project to recommend to a student.



## 3. Requirements

- Provide a means for the Head Professor to run a matching algorithm that produces all possible matches of students to projects. Many students can be in one project.
- Provide a means for the Head Professor to allow only certain projects to be considered in the set of all matches, by means of priority ranking.
- Provide a means for the Student to rank his or her favorite projects.
- Provide a score to each arrangement of projects and students that take into account the ranks above. This will help in choosing the best possible match of students to projects.

## 4. System Design

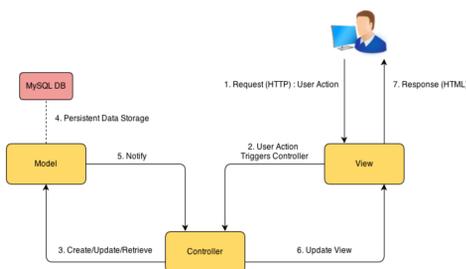


Figure 1. The system was developed using the three-tier and MVC architectural patterns.

## 5. Object Design

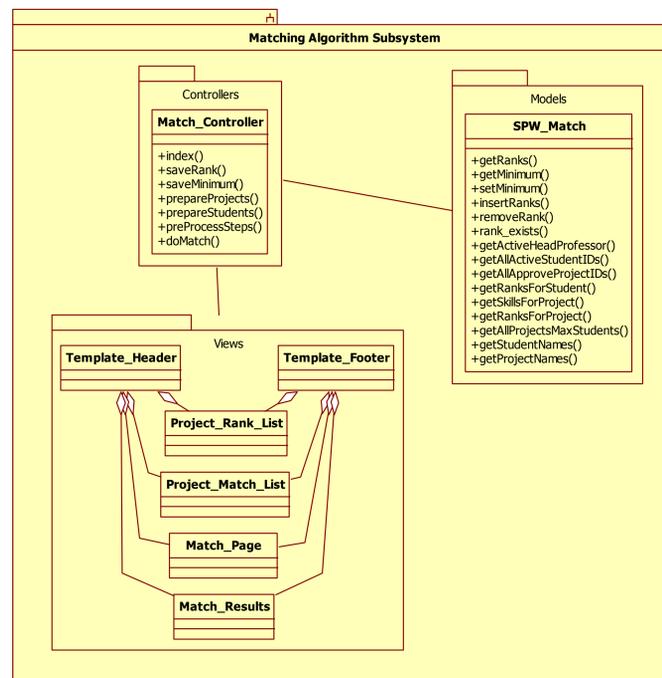


Figure 2. Deployment Diagram for the SPW v.3

## 6. Implementation

Continued use of the same technologies used to develop SPW v.1 and SPW v.2.

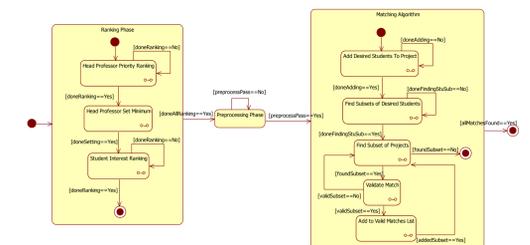
Details:

Enable the Head Professor to:

- ✓ Priority rank the approved projects.
- ✓ Set a minimum number of projects for students to rank.
- ✓ Run a matching algorithm when desired.

Enable students to:

- ✓ Interest rank a minimum number of projects using the minimum provided by the Head Professor.

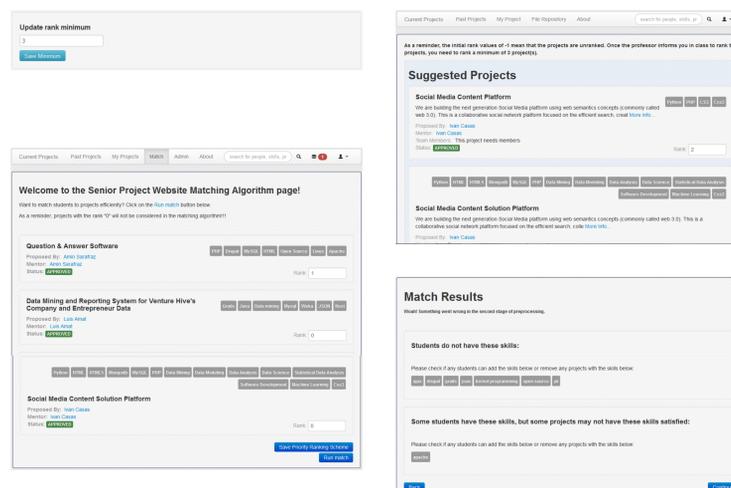


## 7. Verification

- Unit testing was performed using the CodeIgniter Unit Testing Class.
- Integration testing was performed using the big-bang approach. This method was chosen because the subsystems that needed to be integrated are very independent and would not affect each other in any way.
- Tested the system using hard preset data to make sure the results matched the expected.



## 8. Screenshots



## 9. Summary

- The current system did not provide the Head Professor an easy method of finding a single best project to recommend to each student in the Senior Project course.
- The impractical approach of manually going through each projects' set of skills and each students' set of skills to do find a single best recommendation for each student would be time consuming.
- The solution to this problem was to extend the usability of the system by creating an automated and interactive method to match students to projects.
- In the process of creating the solution it was most feasible to maintain the intuitive UI design established in version one of the website.

## Acknowledgement

The material presented in this poster is based upon the work supported by Masoud Sadjadi . I am thankful for the help that I received from my group members, Cynthia Tope and William Marquez. I would also like to thank the previous team that developed SPW v.2. Their hard work created a solid structure for great extensibility .