**Requirements Document**

CIS 4911 – Senior Project U01

Virtual Job Fair 3.0

**Members**

Ana L. Hernandez

Manuel Bouza

Enio Pena Navarro

Tomas Acosta

**Instructor**

Masoud Sadjadi

**Mentor**

Masoud Sadjadi

**Date**

July 25th 2014

**COPYRIGHT**

Copyright © Florida International University 2014 All Rights Reserved. No part of the Virtual Job Fair Project or documentation may be reproduced or modified without the express consent of Florida International University. Redistribution or commercial use is strictly prohibited.

**ABSTRACT**

Chapter 1 gives basic information about the Virtual Job Fair V3, including the problem definition, background on the problem, definition of important terms, and an overview of the document. Chapter 2, the Feasibility Study chapter, provides a description of the current system used worldwide and introduces the purpose of our system, and states the list of high-level user requirements. Finally, this chapter includes an analysis of alternative solutions to the problem. Chapter 3, Project Plan, introduces project management concepts of the project, such as personnel organization, hardware and software resources used for the project, and a list of tasks, milestones, and deliverables.

Chapter 4 introduces the functional requirements of the system, with a description of the usability, reliability, performance and supportability of each use case. Also, it includes descriptions on the use case, static and dynamic models of the system. Chapter 5 is a glossary of domain-specific terms used in the document. Chapter 6, the appendix, contains miscellaneous charts and information, such as an AgileZen table with the division of the project task, a feasibility matrix, static and dynamic diagrams, a cost matrix, and a diary of meetings. Finally, Chapter 7 contains references to external documents that have been used for reference.

Contents

[1. Introduction 10](#_Toc374330621)

[1.1 Problem Definition 10](#_Toc374330622)

[1.2 Scope of system. 10](#_Toc374330623)

[1.3 Terminology - Definitions, acronyms, and abbreviations. 11](#_Toc374330624)

[1.4 Overview of document 11](#_Toc374330625)

[2. Current System (limitations and problems) 12](#_Toc374330626)

[3. Project Plan 13](#_Toc374330627)

[3.1 Project organization 13](#_Toc374330628)

[3.2 Work breakdown 15](#_Toc374330629)

[3.3 Cost Estimate 17](#_Toc374330630)

[4. Proposed System Requirements 18](#_Toc374330631)

[4.1 Functional Requirements 18](#_Toc374330632)

[4.2 Analysis of System Requirements 32](#_Toc374330633)

[4.2.1 Use case model 32](#_Toc374330634)

[4.2.2 Static model 32](#_Toc374330635)

[4.2.3 Dynamic model 34](#_Toc374330636)

[5. Glossary 34](#_Toc374330637)

[6. Appendix 35](#_Toc374330638)

[6.1 Appendix A - Complete use cases 35](#_Toc374330639)

[6.2 Appendix B Use case diagram using UML 60](#_Toc374330640)

[6.3 Appendix D - Dynamic UML diagrams 66](#_Toc374330642)

[Select Upload Image 77](#_Toc374330643)

[New System’s State Diagrams 77](#_Toc374330645)

[Collaborative Editor 77](#_Toc374330646)

[SMS 78](#_Toc374330648)

[ScreenShare 78](#_Toc374330650)

[Whiteboard 79](#_Toc374330652)

[6.5 Appendix E - User Interface designs. 80](#_Toc374330654)

[Log In Using FIU SCIS Credentials 84](#_Toc374330655)

[Whiteboard 84](#_Toc374330657)

[Image Sharing 85](#_Toc374330658)

[6.6 Appendix F - Diary of meeting and tasks. 86](#_Toc374330660)

[7. References 93](#_Toc374330661)

# 1. Introduction

The introductory chapter gives some background information about the Virtual Job Fair v3.0 system. Firstly, the chapter states the problem definition, and scope of the system. Next, the design methodology used is identified. This methodology includes the software process models and the types of models used. Moreover, definitions, acronyms, and abbreviations of terms that will be used in this deliverable are introduced and explained. Finally, it contains an overview of the whole project, which explains the information contained on each chapter.

* 1. **Problem Definition**

Employers looking for talent are always interested in filling out positions with the best possible candidates. In order to accomplish this task, the most effective method to date is to tap local talent, whether it is at universities or job fairs. Given the increasing globalization trend, and the fact that not all employers have the financial or the time capabilities to be registering on every website to seek for potential candidates in different locations. In addition, the current solution provided by universities and job sites is less than ideal.

Our solution to this problem, we will provide FIU Computer Science students with a user-friendly interface where students will be able to get job matches from different job search sites, and not just from the employers who register in the VJF site. This way the students do not need to go looking in different sites, he or she can find everything on our site. Also students will receive notification of jobs matching his/her saved search criteria, so that way the students will know as soon as an opportunity is available. Therefor providing the employer and potential employee a more easy and friendly way to find their match.

* 1. **Scope of system**

One of the main purposes of the Virtual Job Fair v3.0 is to expand the scope of the previous versions. While past versions provided students with the ability to search for jobs, it was restricted to employers who were register with our system. In addition, the search engine was very basic search, and did not provide enough features. Moreover, the notification system was lacking customization, and it was restricted. The students did not have the option to merge/link more accounts in order to have a centralized login no matter which login third party they want to use.

Virtual Job Fair v3.0 expanded on existing functions of the system, however, the system does not currently have the interview functionality. VJF is currently not allowing Employers who register with our system to post jobs. Employers cannot receive customized notifications about students with low matching rate. Students cannot search for jobs based on zip code, or distance. Students cannot register/login with any other third party account other than Google, FIU, FIU Senior Project, and LinkedIn. Moreover, administrator cannot delete any users from the database.

## 1.3 Terminology - Definitions, acronyms, and abbreviations.

**Definitions**

- **Student**: an individual who is currently enrolled in the School of Computing & Information Sciences FIU

- **Job:** an activity done in exchange for payment

- **Full-time:** requiring 40 hours or more hours per week

- **Part-time:** requiring less than 40 hours per week

- **Paid internship:** an internship for which a student will receive compensation

- **Unpaid internship:** an internship for which the student will not receive compensation

- **Benefits:** non-salaried compensation for employees, such as insurance, tuition reimbursement, and retirement benefits

- **Work authorization:** current legal work status of a student. Categories include U.S. Permanent Resident and U.S. Citizen

- **Grade point average:** a number out of 4.0 which gives a representation of a student’s grades in his/her classes throughout his/her college career

**Acronyms**

- **VJF**: Virtual Job Fair

- **FIU:** Florida International University

- **GPA:** Grade point average

- **SCIS:** School of Computing & Information Sciences

**Abbreviations**

As of right now, there are no abbreviations for this project.

## 1.4 Overview of document

The Requirements Document covers several aspects of the Virtual Job Fair project. In Chapter 1, general information such as problem statement, background information on this specific project and definitions is found. Chapter 2 introduces the current system, including its limitations and problems. Moreover, in Chapter 3, project organization is detailed, with specific roles for each of the members assigned. Moreover, work breakdown and cost estimates are given.

Chapter 4 details the functional requirements in terms of use cases and presents the reader with a use case model, static models and dynamic models that represent the system. Chapter 5 contains definitions of domain-specific terms used in the document. Chapter 6 contains the Appendix, in which miscellaneous information, such as static and dynamic models, user interface designs and a diary of meetings. Finally, Chapter 7 contains works used as references.

# 2. Current System (limitations and problems)

The current system was modified in the fall 2013 semester. As it is, it allows students to search for jobs, and get emails with jobs based on their skills. The current system also allows employers to post jobs for students to browse and apply. Moreover, it offers a messaging system, through which employers and students can communicate privately.

The current system also has standard functionality, allowing users to login and logout, to register in the system, to edit their profiles, to change their passwords and to retrieve forgotten passwords.

**Limitations and Constraints**

As implemented, the current system has the following limitations/constraints:

- **Aimed towards FIU Computer Science students:** the scope of the system is geared towards students who are both FIU students and Computer Science majors. However any Computer Science student can greatly benefit from this site.

**- Limited FIU Seniors:** even though the current system does not restrict any user from signing in, an easy way to register should be given to FIU Seniors, given that this project was born from an FIU SCIS Senior course and it is has been strongly suggested by the faculty to include this feature in the system.

- **Limited students registration:** even though the current system does contain a form that allow student to register, the current system lack a way for the students to register using third party accounts like (Google, LinkedIn, FIU Mail, Senior Project web).

- **Limited students’ login:** the current system lacks a way for the students’ user to login using a third party account like (Google, LinkedIn, FIU Mail, Senior Project web).

- **Limited students linking capabilities:** the system lack the ability to provide student a way to link all their third party accounts together into one account. Also the system lack a wizard interface where the student can choose between link conflicts. Example, the student should be able to choose between the name that he or she has on file, and the name that is coming from the third party accounts.

- **Student limitation:** the system lack a way to guide the student when he try to register with an e-mail that is already in the system. In addition; the student should be told that the e-mail is already in the system, and provide the student with some option of what to do next.

- **Inability for student to merge accounts:** the current system lacks a feature that allow student to merge two accounts into one.

**- Students search for job is restricted:** the current system only provides search by skills or search by company and job type.

**- Student job email notification based on skills:** student will only get notified of jobs matching their skills.

- **Student are not allow to save job search queries:** after students query the system for jobs, they are able to save query to future search.

- **Job posting limited to registered employers:** job posting are only available from registered employers only.

- **Limited student notification:** the current system does not allow student to enable/disable their notification. The student’s user is not able to say if he or she is looking for jobs. Since a student’s user may already be employ and want job notification to stop show in his or her e-mail.

- **Limited student job notification:** the current system do not provide the student’s user with job’s notifications base on their skillset.

**- Limited student notification based on saved custom job queries:** the current system does not allow student’s user to save their job search and receive notification base on it.

- **Limited admin setting notification:** the current system does not allow the admin to toggle between enable/disable notifications globally.

**- Limited employer notification:** the current system does not allow employers to receive notification of students matching job posting description.

- **Not API implement:** the current system lacks an API that allow external sources (CareerPath) to push job posting into the VJF system.

- **Limited admin API management:** the current system lacks to allow administrators to control all aspects of the implemented VJF API through an intuitive interface.

- **Limited admin interface:** the current system lacks an administrative interface to bulk import jobs from the FIU SCIS CareerPath system, using a provided CareerPath API endpoint.

- **Limited interview functionality:** the system lacks for video interviews and live chat, limiting the interaction between students and employers to a very basic level.

**- Limited Document Collaboration functionality:** the current system has a very limited collaboration ability. It lacks the ability to have users of the system collaborate on documents in real-time which is a crucial part of an interview process.

- **Inability to share images between students and employers:** the current system lacks an image-sharing feature which allows students and employers to exchange pictures which can enhance the interview experience.

- **No drawing feature:** the current system does not have any type of drawing feature that may allow students and employers to brainstorm, exchange drawings or jot down ideas while interviewing.

- **Reminder system:** the current system lacks a way of reminding students or employers of upcoming interviews.

**- Users are not allowed to delete their accounts:** once registered, students and/or employees are not allowed to remove themselves from the system’s database.

- **Employers are not able to contact students through other means other than by email:** The current implementation makes it very hard to keep both students and employers connected outside of it.

- **Students and employers are not reminded of important deadlines:** The current system does not have any functionality that alerts users of new events.

# 3. Project Plan

The project plan chapter introduces VJF 3.0 from a project management perspective. Firstly, the project organization is described, with the roles for each member listed. After that, milestones, tasks, and deliverables will be listed. Finally, a cost estimate for whole project is presented in terms of a feasibility matrix.

## 3.1 Project organization

For this project, each of the four members will be in charge of adding at least two pieces of functionality to the system.

Ana L. Hernandez will be responsible for creating a Google like advanced search which will allow student to search for jobs within our database (CareerPath jobs), and/or using external sources (Indeed & CareerBuilder). Also, she will expand the navigation bar search to search by keywords. In addition, she will be responsible for the integration for the Indeed.com and CareerBuilder.com API for the expansion of the job search for both the advanced search and the navigation bar search. Moreover, she will implement a save query feature which will allow students to save any query they just searched for, and then modify its preference to receive email notification with the saved queries.

Manuel Bouza will be responsible for the login and registration pages. He will find what it is wrong with the third party accounts links, and he will find a solution that will prevent this from happening again. In addition he will provide the student’s user with a wizard that will guide the student through the registration page when the student try to register with an email that is already in the database. Also Manuel will be responsible for finding a way so that the student’s user can link all their third party accounts into one account. Manuel will change the profile page so it lets the student’s user know to which third party account the student’s user is link, and to which the student’s user is not link. Finally Manuel will be responsible for providing in the toolbar of the student’s user a button to merge two accounts. Furthermore; Manuel has to provide the student’s user with a wizard that will allow student to choose which information the user want to keep when a merge conflict happens, or when linking accounts.

Enio Pena Navarro will be responsible for adding this next features. First, Enio will designing an API that allows external sources (SCIS CareerPath) to push job postings into the VJF system. Second, he will provide the administrators a way to control all aspects of the implemented VJF API through an intuitive interface, and provide an administrative interface to bulk import jobs from the FIU SCIS CareerPath system, using a provided CareerPath API endpoint. Third, Enio will allow the administrators to maintain the system’s job skills database, and provide the administrators with an interface to manage job posting and users. In general, improve the administrator’s interface.

Tomas Acosta will be responsible of adding two main features to the system. First, Tomas will provide a user Interface settings in user’s profiles that enables the users to activate/deactivate notifications. Second, Tomas is responsible of Job Matching Notification Engine: This engine is in charge of sending students and employers notifications of job listings and students matches respectively.

Below is a table which represents the roles of each of the members:

|  |  |  |
| --- | --- | --- |
| **Team Member** | **Primary Task** | **General Task** |
| Ana L. Hernandez | Developer  Document Editor | Additional shared tasks |
| Tomas Acosta | Developer  Document Editor | Additional shared tasks |
| Manuel Bouza | Developer  Document Editor | Additional shared tasks |
| Enio Pena Navarro | Developer  Document Editor | Additional shared tasks |

## 3.2 Work breakdown

Below is a list of all different tasks, milestones, and deliverables for the project:

|  |  |  |
| --- | --- | --- |
| Week | Date | Task, Milestones, Deliverables |
| 2 | 5/20/14 | Feasibility Study & Project Plan, &  Requirement Document, & Design Document  Design Milestone |
| 3 | 5/27/14 | Implementation: Unit Testing  Testing Milestone |
| 4 | 6/3/14 | Debugging & Code Refinement |
| 5 | 6/10/14 | Debugging & Code Refinement |
| 6 | 6/17/14 | Debugging & Code Refinement |
| 7 | 6/24/14 | Implementation: System Testing  Testing Milestone |
| 8 | 7/1/14 | Debugging & Code Refinement |
| 9 | 7/8/14 | Debugging & Code Refinement |
| 10 | 7/15/14 | Debugging & Code Refinement |
| 11 | 7/22/14 | Debugging & Code Refinement |
| 11 | 7/25/14 | Final Presentation, & Final Deliverable, &  Project Deployment  Documentation Milestone |

## 3.3 Cost Estimate

The following feasibility matrix represents an estimate of the items and labor required for the project. These estimated costs are accurate as of Monday, September 23rd, 2013.

|  |  |
| --- | --- |
| **Item** | **Item Cost** |
| Yii Framework | $0.00 |
| Twitter Bootstrap | $0.00 |
| Easy RTC | $0.00 |
| Hardware | $1200.00 |
| Development | $0.00 |
| Testing | $0.00 |
| Contingencies | $240.00 (~20% of total cost) |
|  | **Total:**$1200.00 |

# 4. Proposed System Requirements

The proposed system is Virtual Job Fair v3.0, which would expand upon the already-implemented Virtual Job Fair v2.0. The main idea behind Virtual Job Fair v3.0 is to add new dynamic functionality to the job search, login, notification, and administrator interface. This chapter will introduce the functional and non-functional requirements of the system and the requirements analysis phase of the system.

## 4.1 Functional Requirements

**Current System’s Functional Requirements**

The system shall…

**Allow students and employers to register**

* **Usability**: The register form is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 2 seconds.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow students and employers to view respective profiles**

* **Usability**: Data displayed in profiles is easy to follow. Students are only able to see their own profile and the employer ones. Employers can see all student profiles.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow students and employers to edit their basic profile information**

* **Usability**: The edit form is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow students and employers to take part in a video interview**

* **Usability**: Starting a video interview is simple and understandable.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 3 seconds when connecting.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow students to upload a resume and video resume**

* **Usability**: The upload form is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1-5 seconds, depending on the file size.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow students and employers to upload an image for their profile**

* **Usability**: The upload form is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow students to associate skills to their profile**

* **Usability**: The ability to add skills to a profile is simple and understandable. It can be done by using LinkedIn connect or adding them manually.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within one 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow students to add and delete education information**

* **Usability**: The corresponding form is easy to complete and follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow students to add and delete experience information**

* **Usability**: The corresponding form is easy to complete and follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow students to apply to open job postings and provide a cover letter**

* **Usability**: Students are presented with a user-friendly interface that is easy to complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow students to reply to an employer’s message**

* **Usability**: Students are presented with a clear and simple interface to send messages.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow students to search for jobs based on skills**

* **Usability**: The search form is easy to follow and complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow employers to post jobs**

* **Usability**: The post job form is easy to understand and complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow employers to close a job posting**

* **Usability**: The closing of a post is easy to complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow employers to associate skills to a job posting**

* **Usability**: the addition of skills to a post is simple to complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second when adding each skill.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow employers to search for students based on skills**

* **Usability**: The search form is simple to submit.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow employers to view student profiles**

* **Usability**: The view of a student profile is easy to understand.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within one 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow employers to send messages to students**

* **Usability**: Employers are presented with a clear and simple interface to send messages.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow employers to give students a “virtual handshake” to show interest in the student**

* **Usability**: The virtual handshake form is easy complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow an administrator to disable an account (security)**

* **Usability**: Disabling a user is simple to complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow an administrator to close a job posting**

* **Usability**: Closing a job post is simple to complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow an administrator to validate an employer registration (security)**

* **Usability**: The validation of an employer is done by one click and is simple to complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Require a username and password to log into the system (security)**

* **Usability**: This is required for a user to log in. Form is simple and easy to follow.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Encrypt the user password before storing into the database (security)**

* **Usability**: Storing user password in a secure way without user intervention.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Require login before viewing user profiles (security)**

* **Usability**: Security measure for system. Interface is simple to complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow users to delete a document.**

* **Usability:** The document deletion interface should be intuitive, easy to understand and navigate.
* **Reliability:** The system should perform correctly 99% of the time.
* **Performance:** The system should respond in less than 1 second.
* **Supportability:** The system should be supported on Google Chrome version 29+ the system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow users to import a document.**

* **Usability:** The import document interface should be intuitive, easy to understand and navigate.
* **Reliability:** The system should perform correctly 99% of the time.
* **Performance:** The system should respond in less than 3 seconds.
* **Supportability:** The system should be supported on Google Chrome version 29+ the system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow users to open a document.**

* **Usability:** The document open interface should be intuitive, easy to understand and navigate.
* **Reliability:** The system should perform correctly 99% of the time.
* **Performance:** The system should respond in less than 3 seconds.
* **Supportability:** The system should be supported on Google Chrome version 29+ the system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow employers to send SMS to students**

* **Usability**:  Sending an SMS should not take more than 15 seconds for a novice user.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**:  The SMS should be sent in under 2 seconds after user presses send
* **Supportability**:  Sending SMS page should be supported by IE, Firefox, Chrome and Safari.

**Allow users to receive automatic email and SMS reminders**

* **Usability**:  Not applicable
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: Users should receive a reminder 30 minutes before scheduled interview
* **Supportability**:  Not applicable

**Allow users to confirm their phone numbers**

* **Usability**:  Confirming a phone number should take less than 30 seconds for inexperienced users
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: Sending authentication code and validating it should take less than 2 seconds respectively.
* **Supportability**:  Confirm phone number page should be supported by IE, Firefox, Chrome and Safari

**Allow students to build their profile (basic information, skills, education, and experience) from LinkedIn.**

* **Usability**: The connection with LinkedIn should be easy to follow. Users will enter their LinkedIn credentials and get appropriate data that the user allowed.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow FIU Computer Science Seniors to login using their FIU SCIS credentials**

* **Usability:** The system should provide an easy and integrated login process for FIU SCIS Seniors using the school UNIX account.
* **Reliability:** The system should perform correctly 99% of the time.
* **Performance:** The system should respond in less than 1 second.
* **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow FIU student to login using their FIU credentials**

* **Usability:** The system should provide an easy and integrated login process for FIU students using the school UNIX account.
* **Reliability:** The system should perform correctly 99% of the time.
* **Performance:** The system should respond in less than 1 second.
* **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow student to login using their Google credentials**

* **Usability:** The system should provide an easy and integrated login process for Google accounts.
* **Reliability:** The system should perform correctly 99% of the time.
* **Performance:** The system should respond in less than 1 second.
* **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow student to login using their LinkedIn credentials**

* **Usability:** The system should provide an easy and integrated login process for LinkedIn accounts.
* **Reliability:** The system should perform correctly 99% of the time.
* **Performance:** The system should respond in less than 1 second.
* **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**New System’s Requirements**

**Allow students to search for jobs based on skills, company name, job type, and position on navigation bar**

* **Usability**: The search form is easy to follow and complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 2 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow students to search for jobs based on skills, company name, job type, position, and location on the advanced search**

* **Usability**: The search form is easy to follow and complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 3 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow students to search for jobs based on skills, company name, job type, position, and location on the advanced search**

* **Usability**: The search form is easy to follow and complete.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 3 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow job search results from FIU CareerPath**

* **Usability**: The search button should be press.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 3 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow job search results from Indeed.com**

* **Usability**: The search button should be press.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 3 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow job search results from CareerBuilder.com**

* **Usability**: The search button should be press.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 3 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow students to receive email notification with expanded job search results**

* **Usability**: The profile preference are easy to setup.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should send email within 1, 7 or 30 days.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Prompt student to enter name for query to be save**

* **Usability**: The student should be able to enter name with ease.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow students to save queries to their profiles**

* **Usability**: The system should provide a save button.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow students to active / deactivate saved queries**

* **Usability**: The system should provide a checkbox.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow students to delete saved queries**

* **Usability**: The system should provide a delete icon next to query.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow students to link their third party accounts into one**

* **Usability:** The system should provide an easy and integrated linking process for third parties accounts.
* **Reliability:** The system should perform correctly 99% of the time.
* **Performance:** The system should respond in less than 1 second.
* **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow students to choose between their current information and the information coming in from their third party accounts when linking accounts.**

* **Usability:** The system should provide an easy and integrated interface for linking accounts.
* **Reliability:** The system should perform correctly 99% of the time.
* **Performance:** The system should respond in less than 1 second.
* **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Guide the students to recover his or her password; when trying to register with an e-mail that is already in the system.**

* **Usability:** The system should provide an easy and integrated interface for recovering password.
* **Reliability:** The system should perform correctly 99% of the time.
* **Performance:** The system should respond in less than 1 second.
* **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allows external sources (SCIS CareerPath) to push job postings into the VJF system.**

* **Usability**: The form should be easy to use, Administrators can specify from what day to what day, and submit.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow administrators to control all aspects of the implemented VJF API through an intuitive interface.**

* **Usability**: The interface should be easy to use.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Provide an administrative interface to bulk import jobs from the FIU SCIS CareerPath system, using a provided CareerPath API end-point.**

* **Usability**: The interface should be easy to use.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow administrators to maintain the system’s job skills database.**

* **Usability**: The form should be easy to use, Administrators can specify from what day to what day, and submit.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow an interface for administrators to manage users.**

* **Usability**: The interface should be easy to use.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow an interface for administrators to manage job postings.**

* **Usability**: The interface should be easy to use.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow the administrator to manage the notification system.**

* **Usability**: The interface should be easy to use.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow the administrator to manage the notification system.**

* **Usability**: The interface should be easy to use.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow students to enable/disable if he/she is looking for jobs**

* **Usability**: The button to enable/disable should be easy to use.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow admin users to enable/disable notifications globally**

* **Usability**:  The button to enable/disable should be easy to use.
* **Reliability**:  The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow students to receive jobs notification based on saved custom job search queries to third party job websites**

* **Usability**:  The emails notifications should be easy to use.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow students to receive jobs notification based on their skillset**

* **Usability**:  The emails notifications should be easy to use.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

**Allow employers to receive notification of students matching job posting description**

* **Usability**:  The emails notifications should be easy to use.
* **Reliability**: The system should perform correctly 99% of the time.
* **Performance**: The system should respond within 1 second.
* **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes

## 4.2 Analysis of System Requirements

Analysis models – contains the complete functional specification and is mainly for the designers and programmers. This section describes the diagrams in the Appendices B - D and validates the models against the use cases.

### 4.2.1 Use case model

The use case diagram describes the overall view and functionality that we will add to the system. The proposed functionality deals with the two type of users displayed in the diagram. Students, and Administrators are the targets of these functional requirements.

### 4.2.2 Static model

REFER TO APPENDIX C

The Static Model in appendix C provides developers with a specification of all the classes that should be implemented for the system. It does not differentiate between the different subsystems; it simply contains all of the necessary classes that were identified during the requirements analysis, i.e. the collection of all the classes from the different subsystems. Conventional UML notation was used to make it easier to differentiate between boundary, controller, and model classes, as per-required for the MVC architectural pattern.

## 4.2.3 Dynamic model

In our sequence diagrams, actors can be either students, employers, or any. In all of our use cases, the actors communicate with an object from the view component. Also, some sequence diagrams require communication with our database.

# 5. Glossary

|  |  |
| --- | --- |
| Term | Meaning |
| Class Diagram | A pictorial representation of all the classes in the system |
| Functional Requirement | A function supported by the system, where a function is a set of inputs, the behavior, and outputs. |
| Non-Functional Requirement | A requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. |
| Object Diagram | A pictorial representation of an instance of a class with example of how the data of the class will be populated |
| Sequence Diagram: | A pictorial representation of how processes operate with one another and the user during the course of a specific piece of functionality. |
| Use Case | List of steps defining the interaction between the user and the system to achieve a goal |

* + **Salary:** a periodic payment made to an employee in exchange for services provided. Salaries are provided in yearly terms.
  + **Résumé:** a document which describes a student’s qualifications, skills and education.
  + **Cover Letter:** a document which is used by students to introduce themselves to the companies that they are applying to. It usually goes together with a resume.

# 6. Appendix

## 6.1 Appendix A - Complete use cases

|  |  |
| --- | --- |
| Use Case ID | VJF-001 Registration |
| Description | Registration process for a student type |
| Actor | Student |
| Pre-conditions | * User has navigated to Virtual Job Fair |
| Steps | 1. User clicks on register link 2. User selects registration 3. User selects student or employer 4. User fills in required details 5. User clicks submit |
| Post-conditions | 1. User is redirected to page where he/she is asked to check email for verification link 2. Verification email is sent |
| Exceptions | 1. User fails to fill out one of the details in the form 2. User fails to provide an FIU email address |

|  |  |
| --- | --- |
| Use Case ID | VJF-002 View Profile |
| Description | Viewing profiles for student and employers |
| Actor | All Actors |
| Pre-conditions | 1. User is logged in |
| Steps | 1. User clicks on username hyperlink 2. User is redirected to the user profile |
| Post-conditions | 1. User is on profile page |
| Exceptions | 1. If a student is viewing another student’s profile, certain information is withheld |

|  |  |
| --- | --- |
| Use Case ID | VJF-003 Edit Basic Info |
| Description | Allow a user to edit their profile |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in |
| Steps | 1. User clicks on “My Profile” 2. User clicks on “Edit Basic Info” 3. User changes necessary data 4. User clicks “Save” |
| Post-conditions | 1. User is redirected to “My Profile” page |
| Exceptions | 1. User inputs invalid/empty data |

|  |  |
| --- | --- |
| Use Case ID | VJF-004 Verify Email |
| Description | Allow a user to verify his account |
| Actor | Student, Employer, Faculty |
| Pre-conditions | 1. User has registered for an account and email has been sent |
| Steps | 1. User navigates to verification email sent by system 2. User clicks on verification link 3. User is redirected to Virtual Job Fair page to verify Email |
| Post-conditions | 1. Access is granted to user with correct verification link |
| Exceptions | 1. Verification link does not match system’s expectation |

|  |  |
| --- | --- |
| Use Case ID | VJF-005 Login |
| Description | Allow a user to login to his account |
| Actor | All user types |
| Pre-conditions | 1. User has navigated to Virtual Job Fair website |
| Steps | 1. User enters username and password 2. User clicks “Login” 3. User is redirected to his home page |
| Post-conditions | 1. User is on homepage |
| Exceptions | 1. User entered incorrect username/password combination |

|  |  |
| --- | --- |
| Use Case ID | VJF-006 Logout |
| Description | Allow a user to logout from his account |
| Actor | All user types |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks Logout 2. User is redirected to main page |
| Post-conditions | 1. User is on main page |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-007 Home Page |
| Description | Allow a user to visit Home Page |
| Actor | All user types |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks Home 2. User is redirected to his home page |
| Post-conditions | user is his home page |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-008 Upload Picture |
| Description | Allow user to change his profile picture |
| Actor | all user types |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks My Profile 2. User is redirected to his profile page 3. User clicks on the edit picture button 4. User pick his picture and then click save. |
| Post-conditions | user post his picture |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-009 Upload Resume |
| Description | Allow user to Upload his resume |
| Actor | Student |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks My Profile 2. User is redirected to his profile page 3. User clicks on the edit resume button 4. User pick his resume file and then click save. |
| Post-conditions | user post his resume |
| Exceptions | User file invalid/empty data |

|  |  |
| --- | --- |
| Use Case ID | VJF-0011 Add Education |
| Description | Allow user to add education to his profile |
| Actor | Student |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks My Profile 2. User is redirected to his profile page 3. User clicks on the add education 4. User add his education info then click save |
| Post-conditions | user add education to his profile |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0012 Delete Education |
| Description | allow user to delete education from his profile |
| Actor | Student |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks My Profile 2. User is redirected to his profile page 3. User clicks on the delete education button |
| Post-conditions | user delete the education that has been chosen to be deleted |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0013 Add Experience |
| Description | user can add experience to his profile |
| Actor | Student |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks My Profile 2. User is redirected to his profile page 3. User clicks on the add experience 4. User add his experience info then click save |
| Post-conditions | user add experience to his profile |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0014 Delete Experience |
| Description | user can delete experience from his profile |
| Actor | Student |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks My Profile 2. User is redirected to his profile page 3. User clicks on the delete experience button |
| Post-conditions | user delete the experience that has been chosen to be deleted |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0015 Change password |
| Description | user can change his password |
| Actor | All Actors |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks My Profile 2. User is redirected to his profile page 3. User clicks on the change password button 4. User is redirect to a new page for changing his password 5. User is typing his old, new password and retype the new password 6. User click submit and redirect to the login page |
| Post-conditions | user change his password |
| Exceptions | - Old Password was incorrect.  - Passwords do not match |

|  |  |
| --- | --- |
| Use Case ID | VJF-0016 Add Skill |
| Description | user can add skill to his profile |
| Actor | Student |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks My Profile 2. User is redirected to his profile page 3. User type a new skill 4. User clicks add skill |
| Post-conditions | user add a new skill |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0017 Delete skill |
| Description | user can delete skill from his profile |
| Actor | Student |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks My Profile 2. User is redirected to his profile page 3. User is clicking on the skill delete button |
| Post-conditions | Skill is deleted |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0018 Change skills Order |
| Description | user can change the order of each skill |
| Actor | Student |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks My Profile 2. User is redirected to his profile page 3. User drags the skill to any position 4. User clicks save skills |
| Post-conditions | User change the order of the skill |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0019 Integrate LinkedIn |
| Description | Get user information from LinkedIn |
| Actor | Student |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks My Profile 2. User is redirected to his profile page 3. User clicks on LinkedIn link 4. User is redirected to page where prompted for username and password for LinkedIn 5. User enters username and password for LinkedIn and clicks continue. 6. User is redirected to My Profile with complete information from LinkedIn |
| Post-conditions | User profile is built |
| Exceptions | User cancels the action |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | VJF-0020 Start Video Interview |
| Description | | Start Video Interview |
| Actor | | Employer, Student |
| Pre-conditions | | 1. Actors are logged in 2. A video interview has been previously scheduled 3. A notification for the video interview was sent to both parties and is displaying in the homepage 4. Actors are in homepage |
| Steps | | 1. Actor clicks on link to video interview in the notifications window in homepage    1. Notification for employer: ([You schedule interview with Diego on 2013-03-13 at 3:00pm](http://srprog-spr13-01.aul.fiu.edu:8080/demos/videointerview.html?view=Diego&notificationRead=842&usertype=2) )    2. Notification for Student: ([Company IBM wants to have a video interview with you](http://srprog-spr13-01.aul.fiu.edu:8080/demos/videointerview.html?view=IBM) [2013-03-13 at 3:00pm](http://srprog-spr13-01.aul.fiu.edu:8080/demos/videointerview.html?view=Diego&notificationRead=842&usertype=2) .[Good Luck!](http://srprog-spr13-01.aul.fiu.edu:8080/demos/videointerview.html?view=IBM) ) 2. Actor is redirected to the video interview page where he/she sees to the left the video connection tools and the video window, and to the right:    1. For the employer:       1. the profile of the student participating in the interview.    2. For the student:       1. the profile of the employer conducting the interview. 3. Once the employer is ready to start the interview he/she clicks in the connect button which allows the student participating in the interview to connect right after 4. Once both parties are connect the video interview is started 5. Once the interview is finish, the employer and student clicks the finish button to be disconnected |
| Post-conditions | | Both parties participated in a video interview |
| Exceptions | Actor ends the interview before it is finished | |

|  |  |
| --- | --- |
| Use Case ID | VJF-0021 Accept Interview |
| Description | Student user accepts a video interview |
| Actor | Student |
| Pre-conditions | Actor is logged in and is at the home page |
| Steps | 1. Actor clicks on the video interview notification section. 2. Actor is notified that a video interview has been schedule for him 3. Actor is asked to confirm the video interview 4. Actor clicks the accept button |
| Post-conditions | The video interview is confirmed and a notification is sent back to the user who originated the interview (employer user) |
| Exceptions | Actor does not accept the interview and closes the notification |

|  |  |
| --- | --- |
| Use Case ID | VJF-0022 View Student Profile |
| Description | Allow Employer to view student profile |
| Actor | Employer |
| Pre-conditions | Employer is logged in and is at the home page |
| Steps | 1. Employer is typing student name on the search input. 2. Employer is clicking on the student that he want to view |
| Post-conditions | The employer is on the student profile view |
| Exceptions | The employer is typing a wrong student name that doesn't exist |

|  |  |
| --- | --- |
| Use Case ID | VJF-0023 Send Message |
| Description | Send a message to a user |
| Actor | A user |
| Pre-conditions | -User is in the compose message page |
| Steps | 1-User populates the ‘To’ field 2-User populates the ‘Subject’ field 3-User types in the message in the text area  4-User clicks ‘Send’ |
| Post-conditions | The system sends the message. The message appears in the inbox of the target user |
| Exceptions | -Inexistent username selected as the receiver of the message  -Wrong username format typed in the ‘To’ field |

|  |  |
| --- | --- |
| Use Case ID | VJF-0024 Reply to Message |
| Description | Reply to a message from some user |
| Actor | A user |
| Pre-conditions | -User has selected a message to be seen |
| Steps | 1-User clicks on the Reply button that appears when reading a message |
| Post-conditions | 2- The System redirects the user to the compose a message page, and the original message the user had selected appears in the text area in the format:  On <Date> <User> wrote:  <message> |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0025 Get Inbox |
| Description | User requests to see all the received messages |
| Actor | A User |
| Pre-conditions | -User is logged in |
| Steps | 1- User navigates to the messages page |
| Post-conditions | User is shown with a list of all the received messages in the format: <Sender> <Subject> |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0026 Get Sent Messages |
| Description | User requests to see all the messages he/she has sent |
| Actor | A User |
| Pre-conditions | -User is logged in |
| Steps | 1- User selects the ‘Sent” messages from the messages page |
| Post-conditions | User is shown with a list of all the sent messages in the format: <Receiver> <Subject> |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0027 Get Trashed Messages |
| Description | User requests to see all the trashed messages |
| Actor | A User |
| Pre-conditions | -User is logged in |
| Steps | 1- User selects the ‘Trash” messages from the messages page |
| Post-conditions | User is shown with a list of all the trashed messages in the format: <Sender/Receiver> <Subject> |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0028 Delete Messages |
| Description | User checks all the messages he/she wants to send to the trash |
| Actor | A User |
| Pre-conditions | -User is logged in |
| Steps | 1-User checks the messages to be sent to the trash  2-User clicks on the trash icon |
| Post-conditions | The selected messages are sent to the trash |
| Exceptions | User does not select any messages before clicking on the trash icon. The System invokes an alert message |

|  |  |
| --- | --- |
| Use Case ID | VJF-0029 Post Job |
| Description | Employer posts a job for students to apply to |
| Actor | Employer |
| Pre-conditions | 1. Employer is logged in 2. Employer is on Home Page |
| Steps | 1. Employer clicks on Post Job Menu Item 2. Employer fills in job details (type, description, compensation, expire date) 3. Employer adds skills to posting if necessary 4. Employer clicks post job |
| Post-conditions | 1. Employer is taken to student match page to view students whose skillset is aligned with the job skillset |
| Exceptions | 1. Employer fills in job details incorrectly, is given an error |

|  |  |
| --- | --- |
| Use Case ID | VJF-0030 Virtual Handshake |
| Description | Employer gives student a virtual handshake to show interest |
| Actor | Employer, Student |
| Pre-conditions | 1. Employer has posted a job |
| Steps | 1. After employer posts job, he is taken to a student match page 2. Employer can review the list of students who matches with the job he posted 3. Employer clicks on “virtual handshake” for any student |
| Post-conditions | 1. Employer remains on student match page 2. Student receives a notification that the employer has shown interest in him for the position |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0031 Edit Company Info |
| Description | Employer changes his company Information |
| Actor | Employer |
| Pre-conditions | 1. Employer is logged in 2. Employer is viewing his profile |
| Steps | 1. Employer clicks on edit image in company info section 2. Employer changes text in any of the text boxes 3. Employer clicks on checkmark |
| Post-conditions | 1. New company info is saved 2. Employer is redirected back to profile page |
| Exceptions | Employer fills in invalid values for the fields |
| Use Case ID | VJF-0032 Search Jobs |
| Description | Student searches for jobs by skill |
| Actor | Student |
| Pre-conditions | 1. Student is logged in 2. Student is on home page |
| Steps | 1. Student clicks on job search text box 2. Student begins typing a skill 3. Student either selects skill from auto complete or fully types out skill 4. Student clicks submit |
| Post-conditions | 1. Student is redirected to search result page with relevant jobs, and option to search more jobs |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0033 Search Students |
| Description | Student searches for students by skill |
| Actor | Employer |
| Pre-conditions | 1. Employer is loggedin 2. Employer is on home page |
| Steps | 1. Employer clicks on job search text box 2. Employer begins typing a skill 3. Employer either selects skill from auto complete or fully types out skill 4. Employer clicks submit |
| Post-conditions | Employer is redirected to search result page with relevant students, and option to search more students |
| Exceptions | None |
| Use Case ID | VJF-0034 Close Job |
| Description | Close a job from further applications |
| Actor | Employer |
| Pre-conditions | 1. Employer is logged in |
| Steps | 1. Employer views his own profile 2. Employer selects a job from one of his own postings 3. Employer clicks on “Close Job” |
| Post-conditions | 1. User is redirected back to the job page 2. The job is closed |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0035 Administrator Close Job |
| Description | Close a job from further applications |
| Actor | Admin |
| Pre-conditions | 1. Admin is logged in |
| Steps | 1. Admin goes to home page 2. Admin enters text included in a job title 3. Admin is redirected to search results with a list of relevant jobs 4. Admin clicks on “delete” for a job of his choosing |
| Post-conditions | 1. The respective job is closed 2. Admin is redirected to search page |
| Exceptions | Search may not return any results |

|  |  |
| --- | --- |
| Use Case ID | VJF-0036 Disable User |
| Description | Disable a user from the website |
| Actor | Admin |
| Pre-conditions | 1. Admin is logged in 2. Admin is on home page |
| Steps | 1. Admin enters a search for a username 2. Admin is taken to results page with list of users 3. Admin can disable users by clicking on “delete” |
| Post-conditions | 1. User is disabled 2. Admin is taken back to search page for more users |
| Exceptions | Search may not return any results |

|  |  |
| --- | --- |
| Use Case ID | VJF-0037 Apply to Job |
| Description | Student applies to an employer’s job posting |
| Actor | Student |
| Pre-conditions | Student is viewing a job |
| Steps | 1. Student clicks on apply 2. System displays a popup box 3. Student fills in a cover letter 4. Student clicks submit |
| Post-conditions | 1. System notifies employer of new application 2. User is redirected back to job page |
| Exceptions | Student has already applied for the job |

|  |  |
| --- | --- |
| Use Case ID | VJF-0038 Read notification |
| Description | User read notification from his/home page |
| Actor | All user types |
| Pre-conditions | 1. User is logged in 2. User is on home page |
| Steps | 1. User is clicking on the notification section that he or she will like to read from. 2. User gets list of notifications |
| Post-conditions | User read his notifications |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0039 Schedule interview |
| Description | Employer is schedule interview with the student |
| Actor | Employer |
| Pre-conditions | 1. Employer is logged in 2. Employer is on student profile view. |
| Steps | 1. Employer is clicking on the button video interview. 2. Employer is choosing the date and time. 3. Employer clicks submit |
| Post-conditions | 1. System is notifies the employer for a new schedule interview that he posted 2. System is notifies the student for a new schedule interview that he has been invited for. |
| Exceptions | employer type wrong input for time and date |

|  |  |
| --- | --- |
| Use Case ID | VJF-0040 Validate an Employer Register |
| Description | Admin validate a new employer that register |
| Actor | Admin |
| Pre-conditions | 1. Admin is logged in. 2. Admin is on his home page. |
| Steps | Admin is clicking on the notification like that validate the new employer. |
| Post-conditions | The new employer got validate. |
| Exceptions | noon |

|  |  |
| --- | --- |
| Use Case ID | VJF-041 Upload Video Resume |
| Description | Allow user to Upload his resume |
| Actor | Student |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | 1. User clicks My Profile 2. User is redirected to his profile page 3. User clicks on the edit video resume button 4. User pick his resume file and then click save. |
| Post-conditions | user post his video resume |
| Exceptions | User file invalid/empty data |

|  |  |
| --- | --- |
| Use Case ID | VJF-042 Share Screen |
| Description | Allow a user to hare his screen |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in. 2. User is in the homepage 3. Interview has been scheduled. 4. Notification for the interview is displayed for both involved parties 5. Other user involved in interview is not sharing screen |
| Steps | 1. User clicks on scheduled interview notification 2. User is redirected to the interview page. 3. User clicks on share screen |
| Post-conditions | User is able to share his screen and database is update with required information. |
| Exceptions | User tries to share screen while the other party is sharing. |

|  |  |
| --- | --- |
| Use Case ID | VJF-043 View Screen Share |
| Description | Allow a user to see a shared screen |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in. 2. User is in the homepage 3. Interview has been scheduled. 4. Notification for the interview is displayed for both involved parties 5. Other user involved in interview is sharing a screen |
| Steps | 1. User clicks on scheduled interview notification  2. User is redirected to the interview page.  3. User clicks on view screen share |
| Post-conditions | User is able to view screen shared by other party |
| Exceptions | The other user is not sharing a screen |

|  |  |
| --- | --- |
| Use Case ID | VJF-044 End Screen Sharing |
| Description | Allow a user to end current screen sharing |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in. 2. User is in the interview portal 3. User is sharing screen |
| Steps | 1. User clicks on end screen sharing |
| Post-conditions | 1. User is not sharing screen  2. System information is updated to allow other party to share screen  3. Other party involved will not continue to see live feed from user screen |

|  |  |
| --- | --- |
| Use Case ID | **VJF-045 Send SMS to student** |
| Description | Allows Employer to send a text message to student |
| Actor | Employer |
| Pre-conditions | 1. User is logged in. 2. User is in the homepage 3. Student has a phone number associated with his account 4. Student has allowed employers to contact him through SMS |
| Steps | 1. User clicks on send SMS  2. User is redirected to SMS page  3. User enters user name of student to contact  4. User enters message  5. User presses send |
| Post-conditions | Selected Student receives text message on his phone. |
| Exceptions | The student has not allowed contact by sms  Student has not entered a phone number |

|  |  |
| --- | --- |
| Use Case ID | **VJF-046 Send interview reminder** |
| Description | An interview reminder is sent to involved parties |
| Actor | Time |
| Pre-conditions | 1. An employer has set up an interview appointment 2. Interview starts in the next 30 minutes. |
| Steps | 1. Database is continuously checked for interviews starting in the next 30 minutes  2. After finding jobs that meet this criteria an email message is sent to involved parties about the event.  3. If any of the accounts is set up to receive SMS then a text message will be sent as well. |
| Post-conditions | Parties involved in the interview will receive an email reminder and SMS according to set up permissions |

|  |  |
| --- | --- |
| Use Case ID | **VJF-047 Confirm phone number** |
| Description | Allows user to confirm and validate a phone number |
| Actor | Employee, Student |
| Pre-conditions | 1. User entered a phone number in the system  2. User is logged in.  3. User is in the homepage |
| Steps | 1. User clicks on validate phone number  2. User is redirected to validate number page  3. User clicks on validate phone  4. An SMS message is sent to user’ phone  5. User enters received code  6. User presses validate button |
| Post-conditions | The system is updated to reflect phone validation |
| Exceptions | User enters a wrong authentication code |

|  |  |
| --- | --- |
| Use Case ID | **VJF-048 Create new document** |
| Description | Allow user to create a new document |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in 2. User is on the interview portal |
| Steps | 1. User clicks on Collaborative Editor 2. User clicks on create new document button 3. The New document editing session is started |
| Post-conditions | 1. User is on the interview portal 2. A new document is displayed |
| Exceptions | 1. Connection Error |

|  |  |
| --- | --- |
| Use Case ID | **VJF-049 Share active document** |
| Description | Allow users to invite another user to a shared document |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in 2. User is on the interview portal 3. User has at least one document to share |
| Steps | 1. User clicks on Collaborative Editor 2. User clicks on share document button 3. Invitation is sent to other user 4. Other user receives notification 5. Other user joins the shared document session |
| Post-conditions | 1. User is on the interview portal 2. User is on the shared document session |
| Exceptions | 1. Connection Error 2. Other user not available |

|  |  |
| --- | --- |
| Use Case ID | **VJF-050 Delete shared document** |
| Description | Allow user to delete a document |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in 2. User is on the interview portal 3. User has at least one document to delete |
| Steps | 1. User clicks on Collaborative Editor 2. User clicks on the manage documents button 3. User selects a document from documents list 4. User clicks on the delete document button 5. User is presented with a confirmation dialog 6. User confirms deletion of file 7. Document is deleted |
| Post-conditions | 1. User is on the interview portal |
| Exceptions | 1. Connection Error 2. User did not select a document to delete |

|  |  |
| --- | --- |
| Use Case ID | **VJF-051 Import document** |
| Description | Allow user to import a document |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in 2. User is on the interview portal 3. User has at least one document to import |
| Steps | 1. User clicks on Collaborative Editor 2. User clicks on import document button 3. User chooses file to import and drags it over the import document area 4. The document is imported into the system |
| Post-conditions | 1. User is on the interview portal |
| Exceptions | 1. Connection Error 2. File is not a valid document |

|  |  |
| --- | --- |
| Use Case ID | **VJF-052 Rename document** |
| Description | Allow user to rename a document |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in 2. User is on the interview portal 3. User has at least one document to rename |
| Steps | 1. User clicks on Collaborative Editor 2. User clicks on the manage documents button 3. User clicks on document to rename 4. User clicks on the rename document button 5. Rename dialog appears 6. User chooses new document name 7. The document name is changed |
| Post-conditions | 1. User is on the interview portal |
| Exceptions | 1. Connection Error |

|  |  |
| --- | --- |
| Use Case ID | **VJF-053 Save document** |
| Description | Allow user to save a document |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in 2. User is on the interview portal 3. User has at least one active document to save |
| Steps | 1. User clicks on save document link 2. The document is saved 3. User is notified |
| Post-conditions | 1. User is on the interview portal |
| Exceptions | 1. Connection Error 2. There are no active documents to save |

|  |  |
| --- | --- |
| Use Case ID | **VJF-054 Open document** |
| Description | Allow user to open a document |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in 2. User is on the interview portal 3. User has at least one active document to open |
| Steps | 1. User clicks on Collaborative Editor 2. User clicks on manage documents 3. User clicks on document to open 4. User clicks on the open document button 5. Document is loaded into the Editor |
| Post-conditions | 1. User is on the interview portal |
| Exceptions | 1. Connection Error 2. There are no active documents to open |

|  |  |
| --- | --- |
| Use Case ID | **VJF-055 Export document** |
| Description | Allow user to export a document |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in 2. User is on the interview portal 3. User has at least one active document to export |
| Steps | 1. User clicks on Collaborative Editor 2. User clicks on the manage documents button 3. User clicks on document to export 4. User clicks on export document 5. User receives the document as a file |
| Post-conditions | 1. User is on the interview portal |
| Exceptions | 1. Connection Error 2. There are no active documents to export |

|  |  |
| --- | --- |
| Use Case ID | **VJF-056 Log in using FIU SCIS Credentials** |
| Description | Allow certain users, FIU SCIS, to login to the system using their FIU SCIS Credentials / UNIX account, Provided by the Senior Project API |
| Actor | Student |
| Pre-conditions | 1. User is at the Login page |
| Steps | 1. User enters his / her username 2. User enters his / her password 3. User is logged in |
| Post-conditions | 1. User is on the interview portal |
| Exceptions | 1. User fails to fill the login form 2. Connection Error |

|  |  |
| --- | --- |
| Use Case ID | **VJF-057 Maintain document access boundaries** |
| Description | Users accessing the system using their personal accounts will have no access to shared documents in the interview session |
| Actor | System |
| Pre-conditions | 3. User is logged in using their personal account |
| Steps | 1. User clicks on create new document link 2. User is informed of the restriction 3. User is required to click on the create temporary document editing session |
| Post-conditions | 1. User is on the interview portal |
| Exceptions | 1. Connection Error |

|  |  |
| --- | --- |
| Use Case ID | **VJF058- Upload Image** |
| Description | Upload a previously-selected image to the server during a live interview |
| Actor | Student or professor |
| Pre-conditions | - A interview session is active  - A whiteboard session has been started  - An image has been selected for submission, according to the previous use case |
| Steps | 1) The user shall click on the “Submit Drawing” button on top of the whiteboard  2) The system shall upload the image  3) The system shall let the user know whether the image was uploaded successfully by displaying a message below the “Choose File” button |
| Post-conditions | The file was successfully upload to the server and can be viewed by either user |
| Exceptions | 1) In step 2, if the file was not one of the following types:  “.gif”, “.jpeg”, “.jpg”, “.pjpeg”, “x-png” or “png”, then the system will display an error message below the “Choose File” button asking the user to upload a different file |

|  |  |
| --- | --- |
| Use Case ID | **VJF059- View Uploaded Image** |
| Description | View a previously-uploaded image specific to an interview session |
| Actor | Student or professor |
| Pre-conditions | - A interview session is active  - A whiteboard session has been started  - An image has been successfully submitted by either user and is ready to view |
| Steps | 1) The user shall click on the “View Drawing” button, located below the “Choose File” button  2) The system shall display the uploaded image below the whiteboard, ready for viewing |
| Post-conditions | The file was successfully upload to the server and can be viewed by either user |
| Exceptions | 1) In step 2, if the image could not be retrieved for any reason, the system will display an error message below the “Choose File” button |

|  |  |
| --- | --- |
| Use Case ID | **VJF060-Show or Restore Whiteboard** |
| Description | Places the whiteboard view in the shared interview screen |
| Actor | Student or professor |
| Pre-conditions | - A interview session is active  - The whiteboard is not showing in the shared interview screen. It is either blank or occupied by a different feature |
| Steps | 1) The user shall click in the blue “Whiteboard” button below the video chat |
| Post-conditions | The whiteboard is now showing in the shared interview screen for the user who clicked on the “Whiteboard” button |
| Exceptions | There are no exceptions for this use case |

|  |  |
| --- | --- |
| Use Case ID | **VJF061- Select Upload Image** |
| Description | Select an image to upload to the server during a live interview to share it with the other user |
| Actor | Student or professor |
| Pre-conditions | 1) A interview session is active  2) A whiteboard session has been started |
| Steps | 1) User shall click on “Choose File” button on top of the whiteboard  2) User shall browse for the image in his local drive and click on the “Open” button to upload it after clicking on the right file  3) The system shall display the file name in the text box next to the “Choose File” button |
| Post-conditions | The file is ready for upload to the system |
| Exceptions | There are no exceptions for this use case |

|  |  |
| --- | --- |
| Use Case ID | **VJF-062 Draw With Pencil** |
| **Description** | Allows actor to draw on the whiteboard using the pencil functionality |
| **Actor** | Employer or student |
| **Pre-conditions** | 1) A interview session is active  2) A whiteboard session has been started |
| **Steps** | 1) The user shall click (either left or right click) anywhere within the shared interview screen where he/she wants to start drawing  2) The user shall drag the mouse, without lifting the click in step 1, to draw  3) Use case ends when the user lifts the mouse and the drawing is put in the whiteboard |
| **Post-conditions** | There will be a drawing visible in the whiteboard |
| **Exceptions** | 1) In Step 2, if the user drags the mouse outside of the shared interview screen, the drawing will be cut off and the use case will end |

|  |  |
| --- | --- |
| Use Case ID | **VJF-063 Change Drawing Tool Pencil** |
| **Description** | Allows an user to change the color of drawing tool |
| **Actor** | Employer or student |
| **Pre-conditions** | 1) A interview session is active  2) A whiteboard session has been started |
| **Steps** | 1) The user shall click on the “Color” button on the left hand side of the whiteboard in the shared interview screen  2) The user shall select the color of the pencil from any of the following colors: red, green, blue, black, yellow, brown or purple, clicking on the button corresponding to the color he/she would like to use |
| **Post-conditions** | There will be a drawing on the whiteboard available for saving and viewing |
| **Exceptions** | 1. In step 2, user clicks outside of the color palette, and the color is not changed |

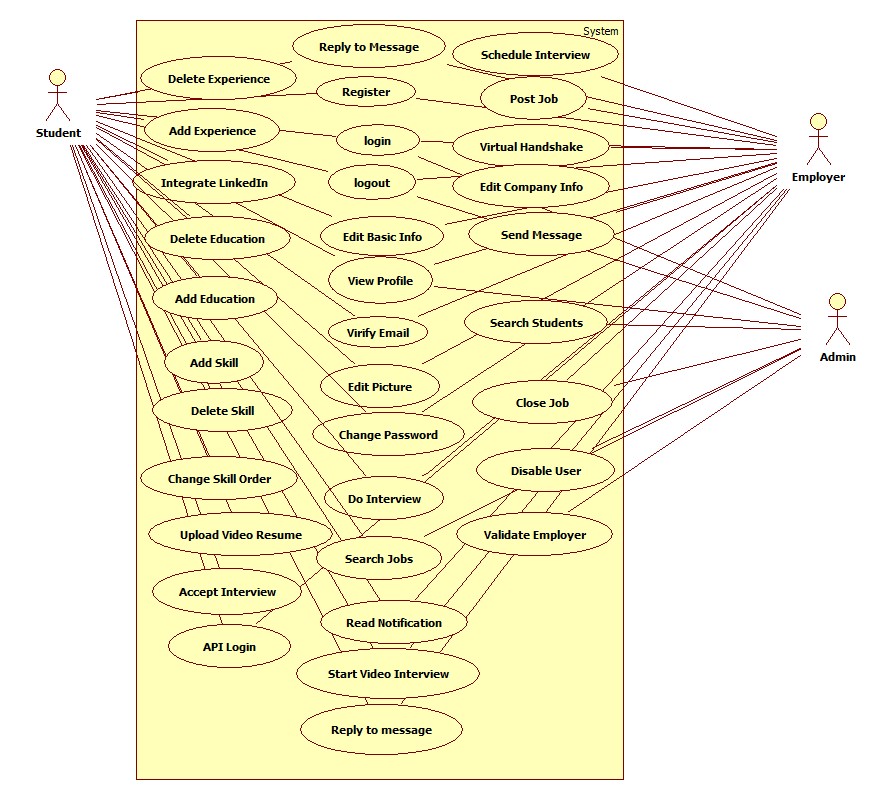
|  |  |
| --- | --- |
| **Use Case ID** | **VJF-064 Type Text Into Whiteboard** |
| **Description** | Allows a user to type text into the whiteboard |
| **Actor** | Employer or student |
| **Pre-conditions** | 1) A interview session is active  2) A whiteboard session has been started |
| **Steps** | 1. Use case begins when, on the whiteboard screen, the actor clicks on the “Pencil” option on the left-hand side 2. From the sub-menu that pops up, the user shall click on the “Text” option 3. The user shall click anywhere on the whiteboard screen where he/she wants the text to show 4. The system shall display a window with a text box, and the user shall type the text that will appear 5. Use case ends when the user presses “OK” after entering the text |
| **Post-conditions** | 1. The text the user typed will appear on the screen where the user initially clicked before typing the text |
| **Exceptions** | 1. In step 2, if the user clicks on the whiteboard when the sub-menu is showing, the current whiteboard feature selected will be used  2. In step 4, if the user clicks “Cancel” instead of “OK”, the window will disappear, and no text will be shown |

|  |  |
| --- | --- |
| **Use Case ID** | **VJF-065 Clear contents of whiteboard** |
| **Description** | Allows users to clear the current contents of the whiteboard |
| **Actor** | Employer or student |
| **Pre-conditions** | 1) A interview session is active  2) A whiteboard session has been started |
| **Steps** | 1. Use case begins when, on the whiteboard screen, the actor clicks on the “Menu” option on the left-hand side  2. From the sub-menu that pops up, the user shall click on “Clear”  3. The system shall display a pop-up message confirming if the user wants to continue   1. Use case ends when user clicks “OK” on the pop-up |
| **Post-conditions** | 1. The whiteboard screen is cleared |
| **Exceptions** | 1. In step 4, if the user clicks on “Cancel”, the whiteboard contents will not be cleared, and the whiteboard screen will be shown again |

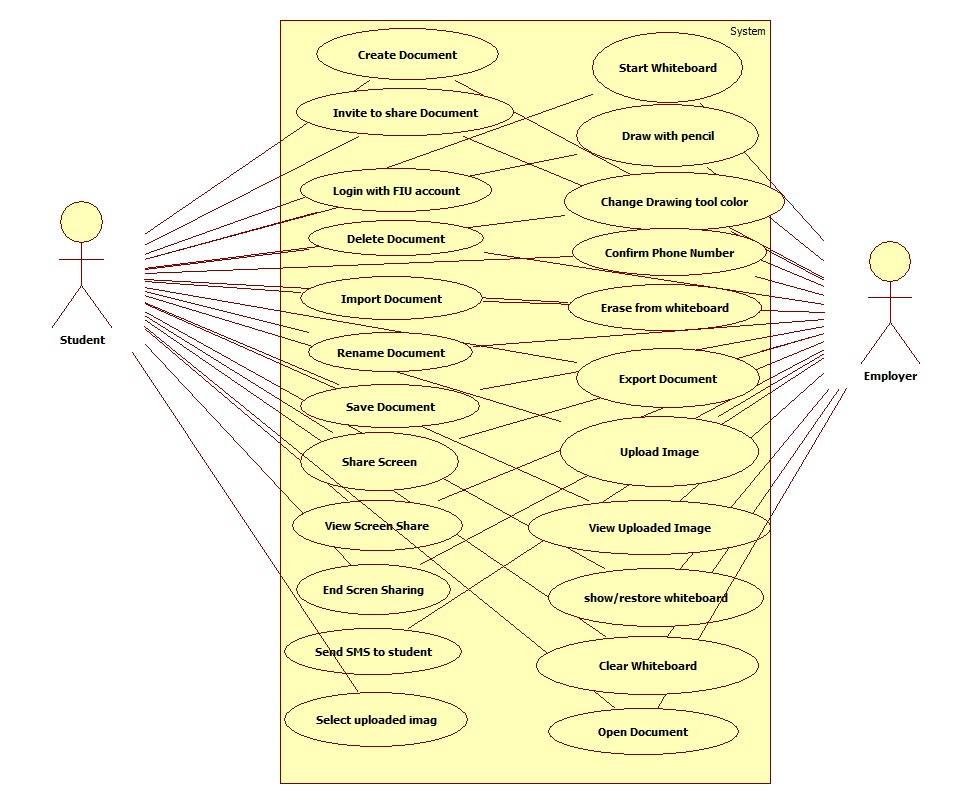
|  |  |
| --- | --- |
| **Use Case ID** | **VJF-066 Erase From Whiteboard** |
| **Description** | Allows users to clear to partially erase drawings |
| **Actor** | Employer or student |
| **Pre-conditions** | 1) A interview session is active  2) A whiteboard session has been started  3) The user has drawn on the whiteboard already |
| **Steps** | 1. Use case begins when, on the whiteboard screen, the actor clicks on the “Pencil” option on the left-hand side  2. From the sub-menu that pops up, the user shall click on “Eraser”  3. The user shall click in the whiteboard location where he/she would like to start erasing   1. The user shall drag the cursor to where he/she wants to erase from the whiteboard 2. Use case ends when the user releases the click |
| **Post-conditions** | 1. The drawing on the whiteboard screen is partially erased |
| **Exceptions** | 2. In step 4, if the user drags the cursor beyond the boundaries of the share screen interview, the whiteboard contents will stop being erased and the use case will end |

## 6.2 Appendix B

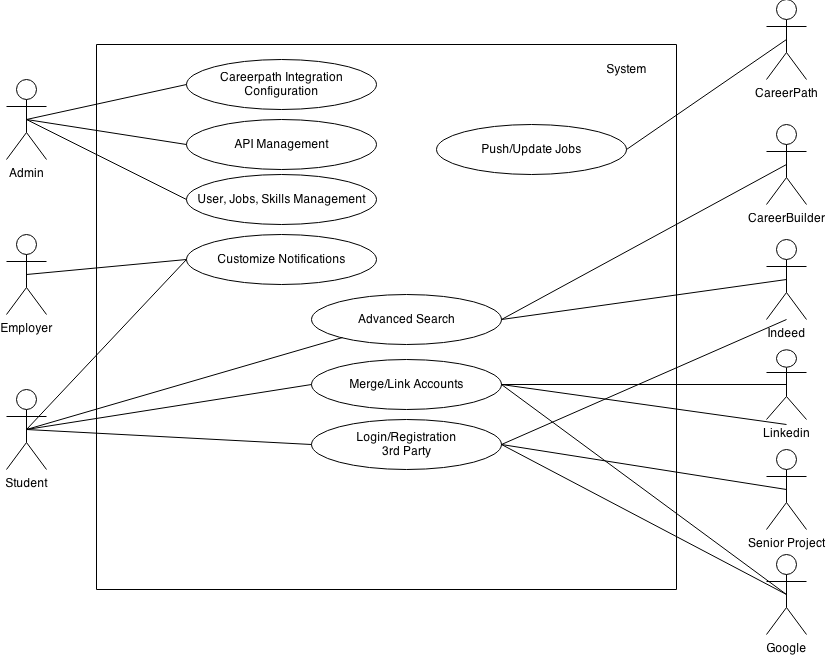
**Spring 2013 System’s Use Case Diagram**



**Current System’s Use Case Diagram**

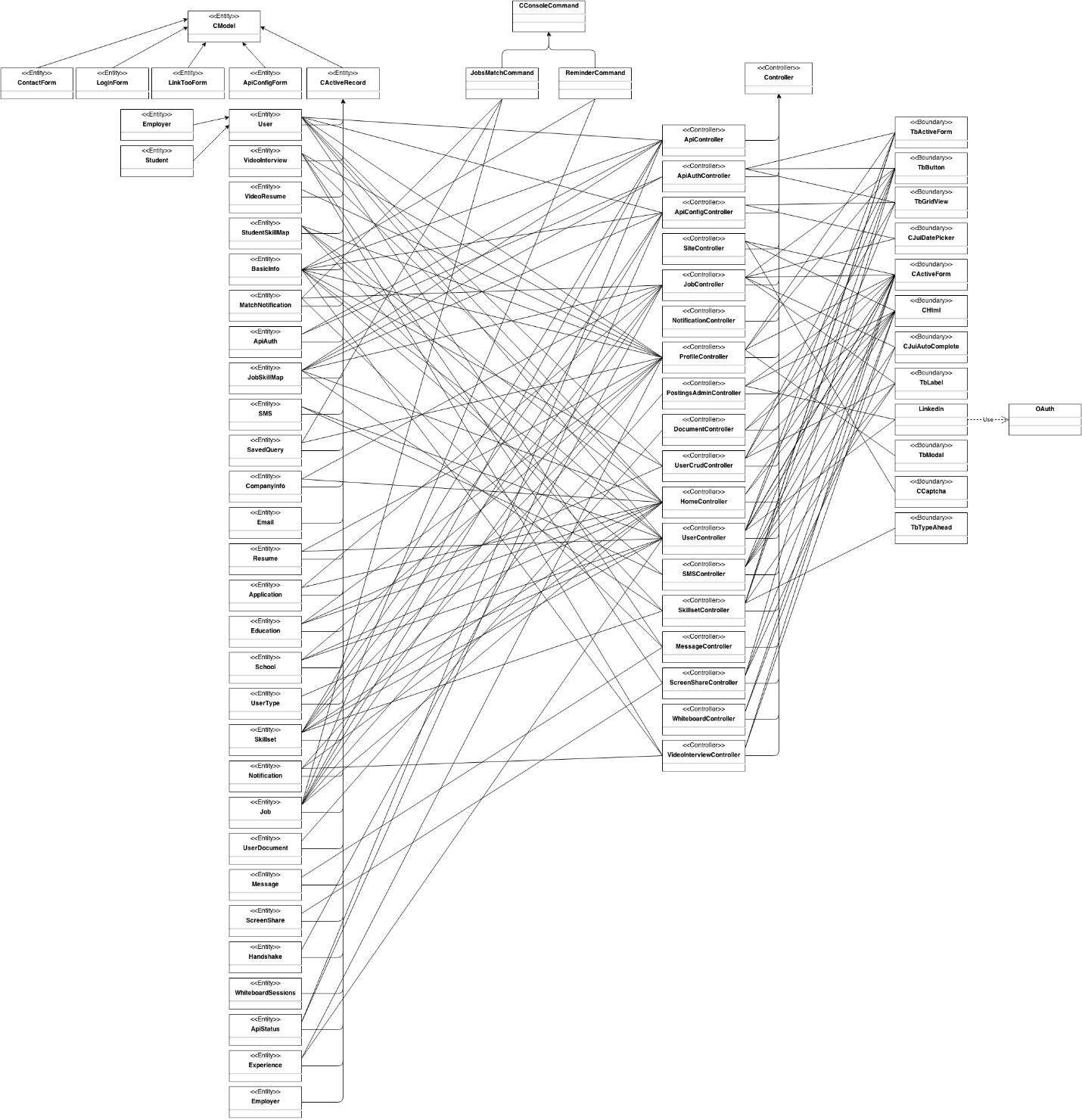


**New System’s Use Case Diagram**

****

**Figure 6.2.1 Use Case Diagram**

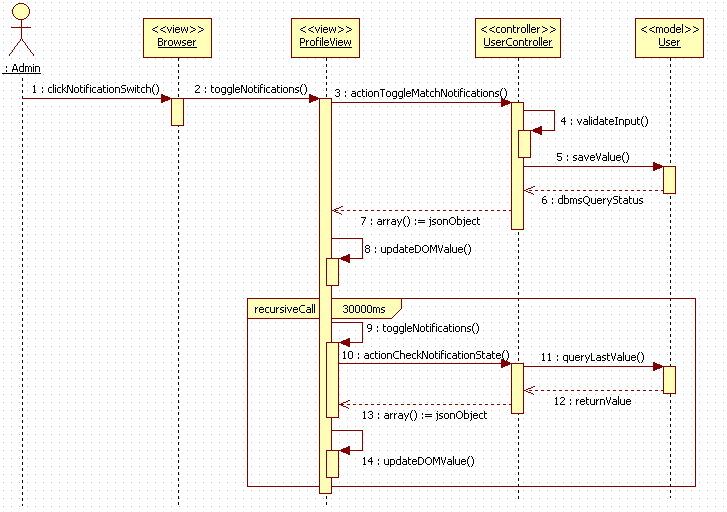
**Minimal Class Diagram**

****

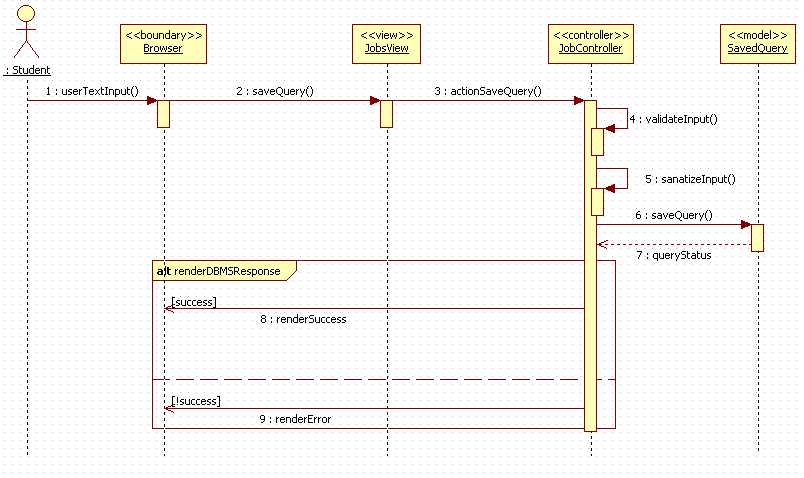
## 6.4 Appendix D - Dynamic UML diagrams

**New System’s Sequence Diagrams**

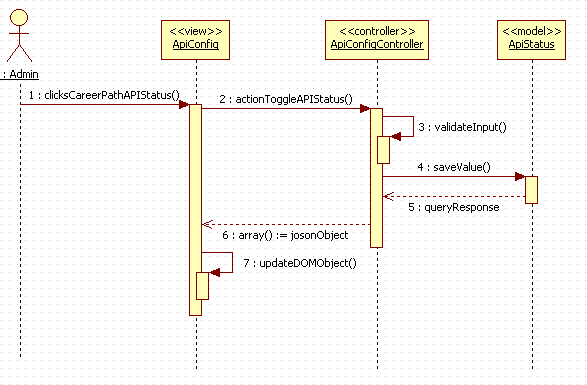
**Automated Notification - Admin Enable Notification**

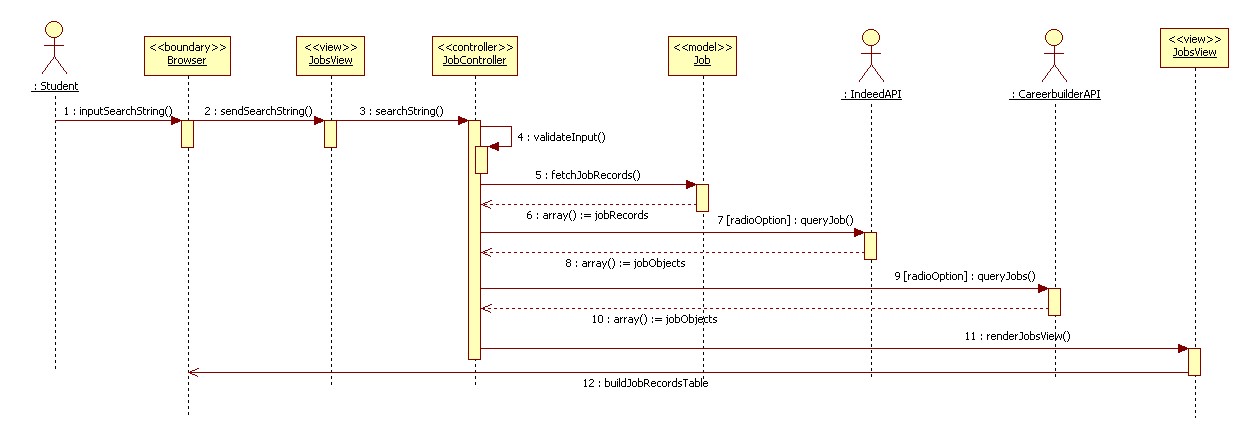
****

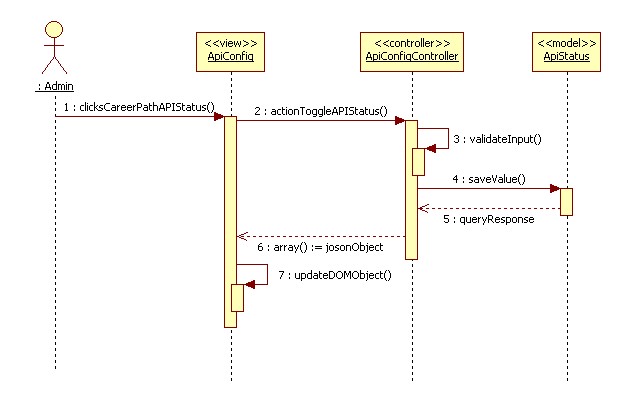
**Search Subsystem - Save Query**

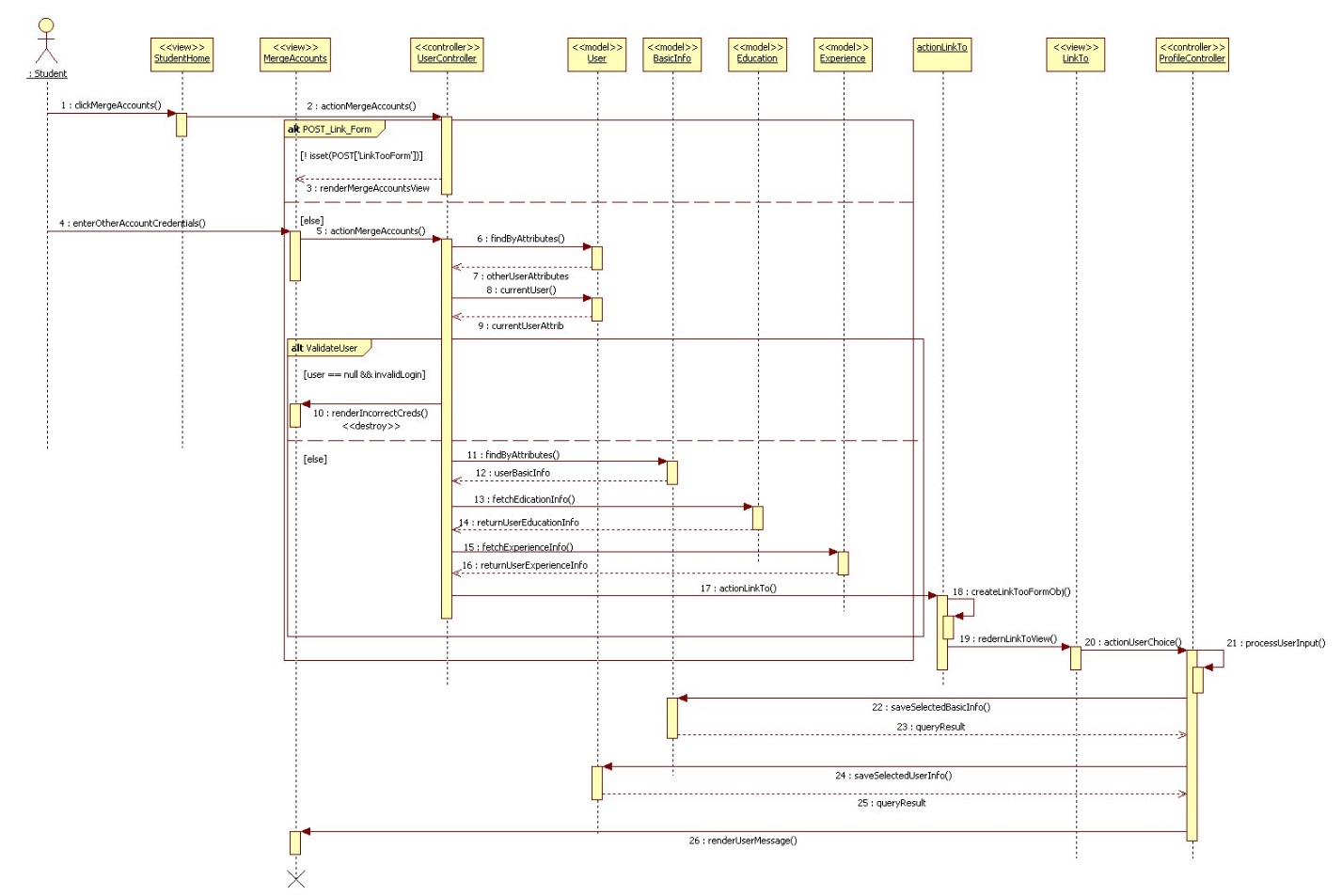
****

**API subsystem - Import Jobs**

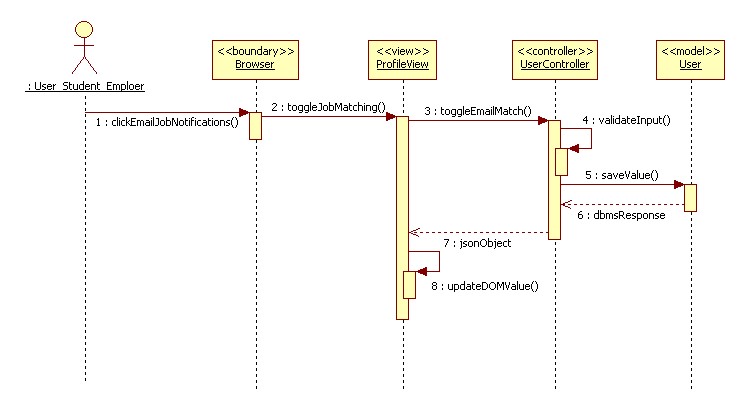
****

**** **Search Subsystem – Advance search**

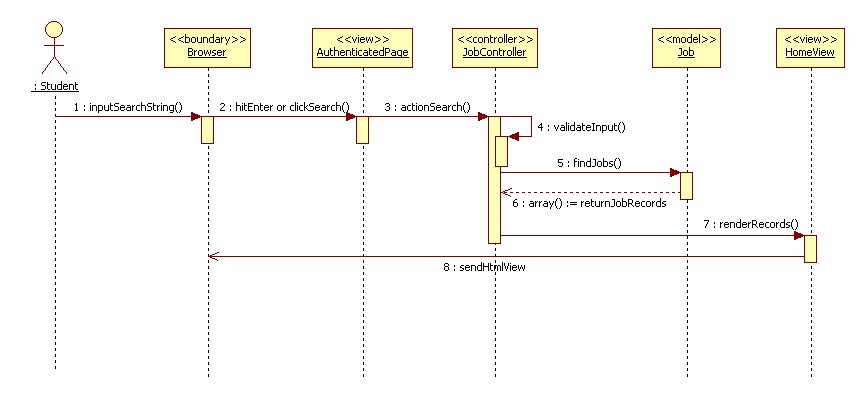
**API Subsystem – Disable API Querying**

**** **Merge Account – Account Merge Action**

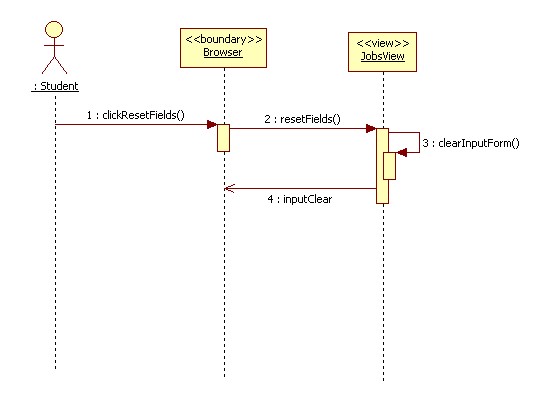
**Automated Notification – Set Notifications On/Off**

****

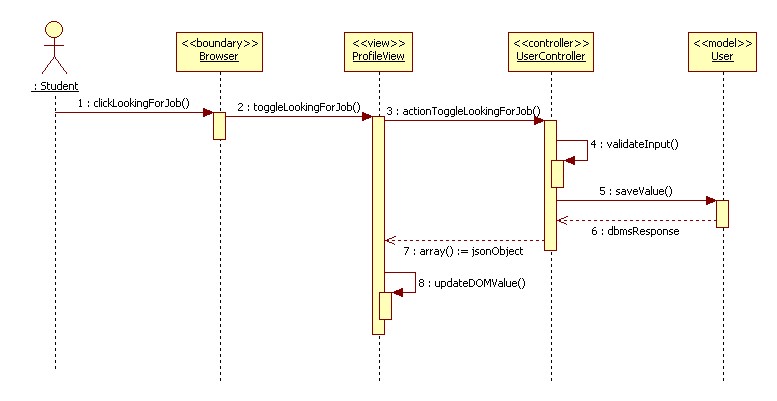
**Search Subsystem – Navigation bar Search**

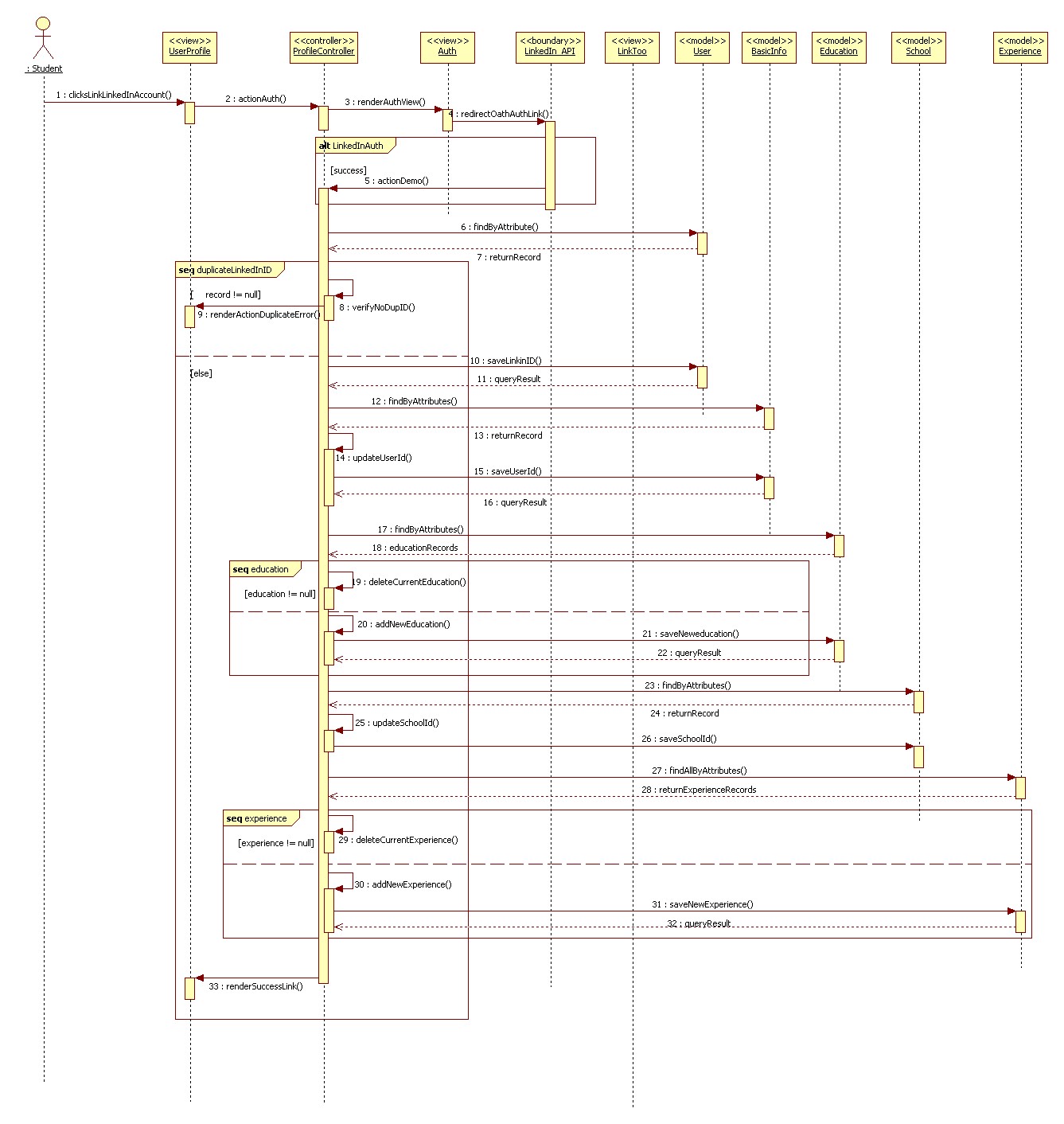
****

**Search Subsystem – Reset Advance Search Inputs**

****

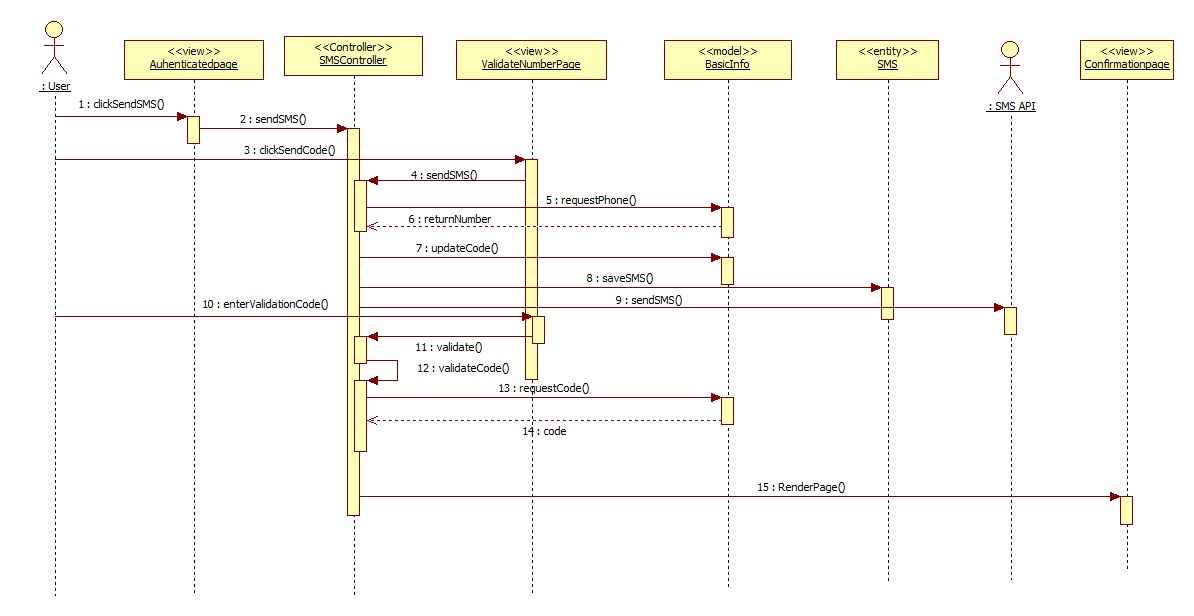
**Search Subsystem –Set Job Search Status**



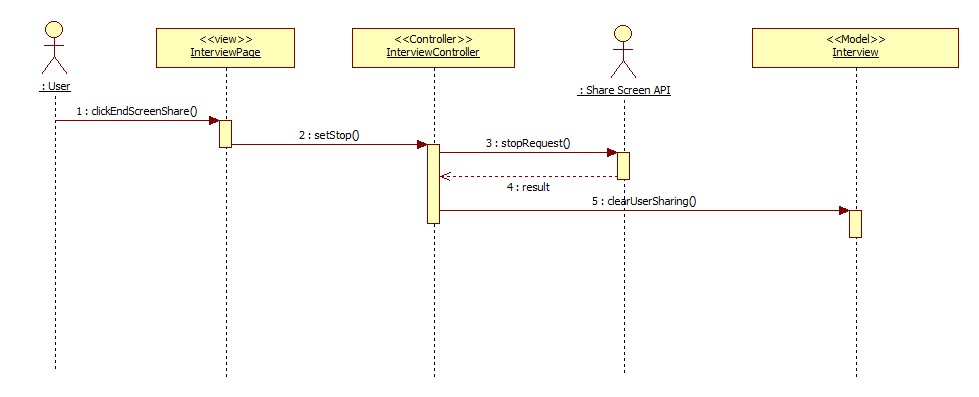
**Link Account Subsystem - Linking Account**

**Current System’s Sequence Diagrams**

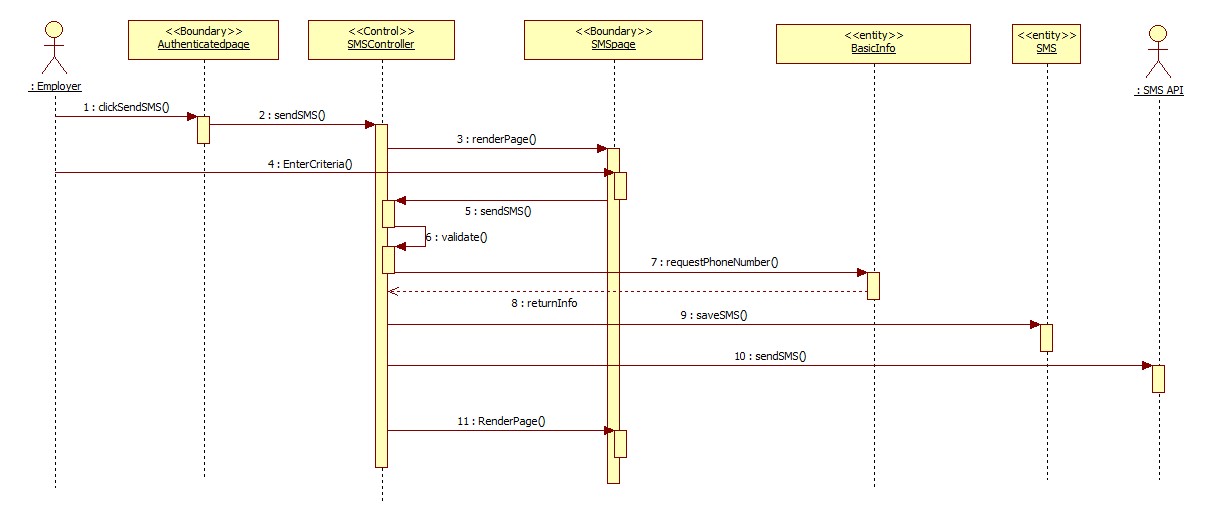
Confirm Phone Number



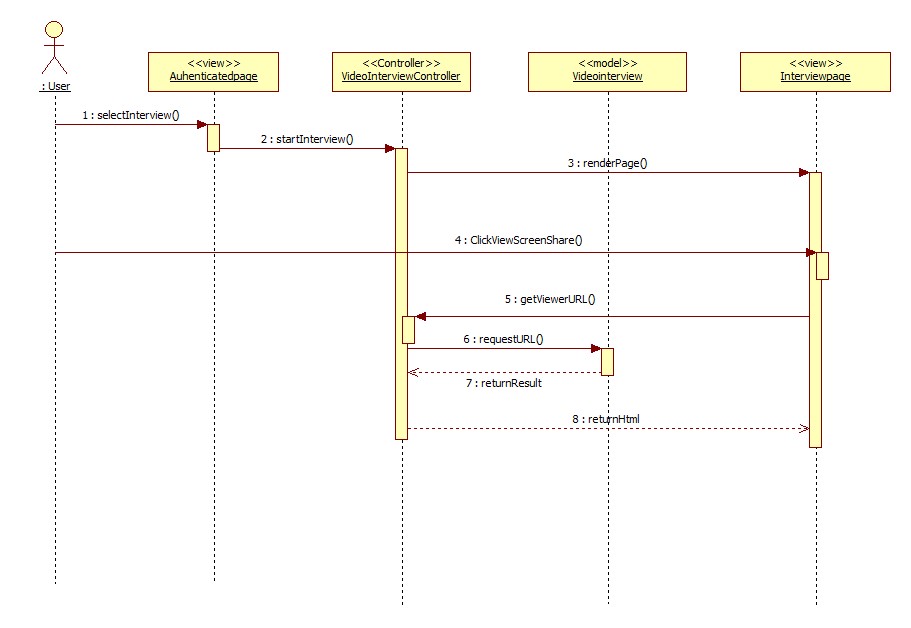
End Screen Share



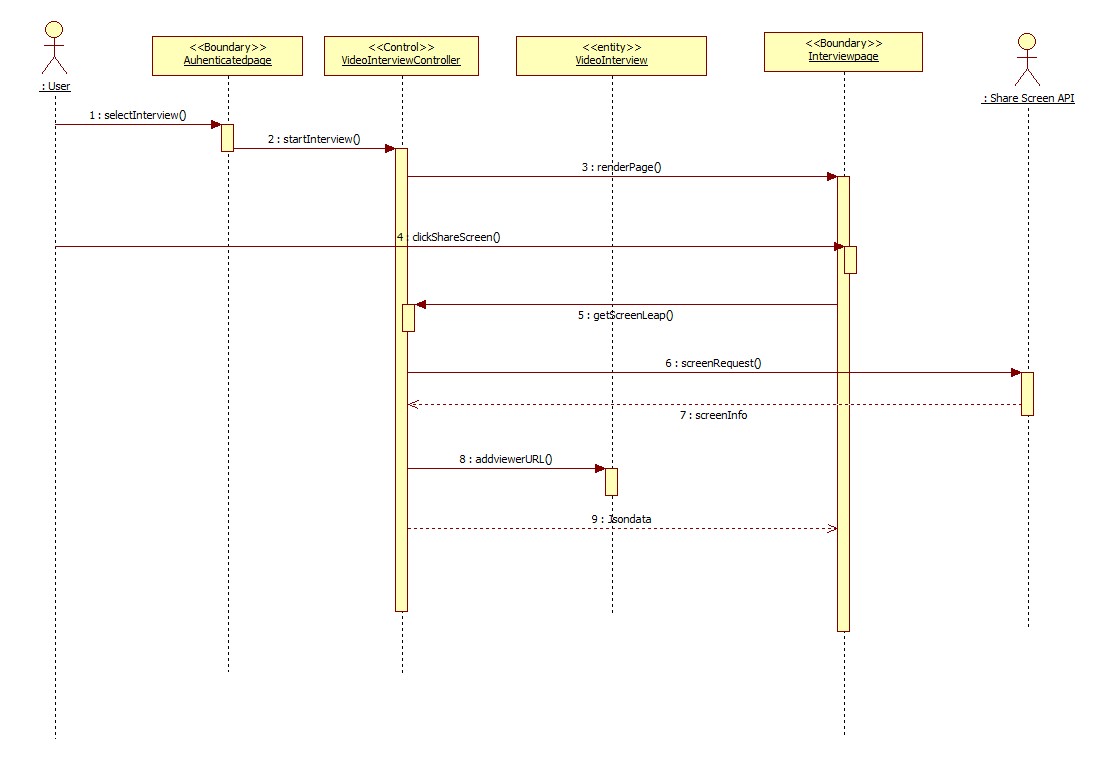
Send SMS



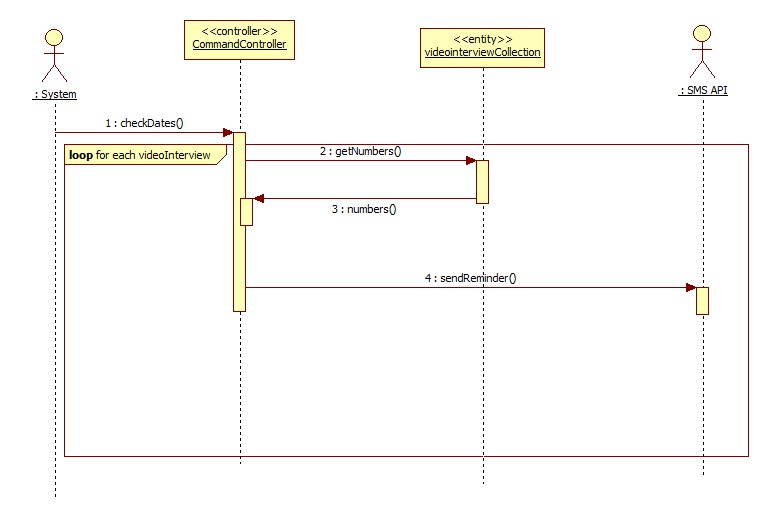
View Screen Share



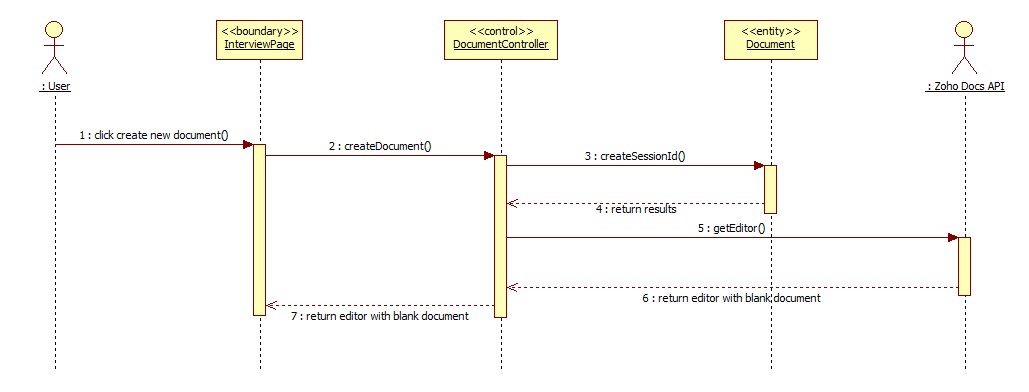
Share Screen



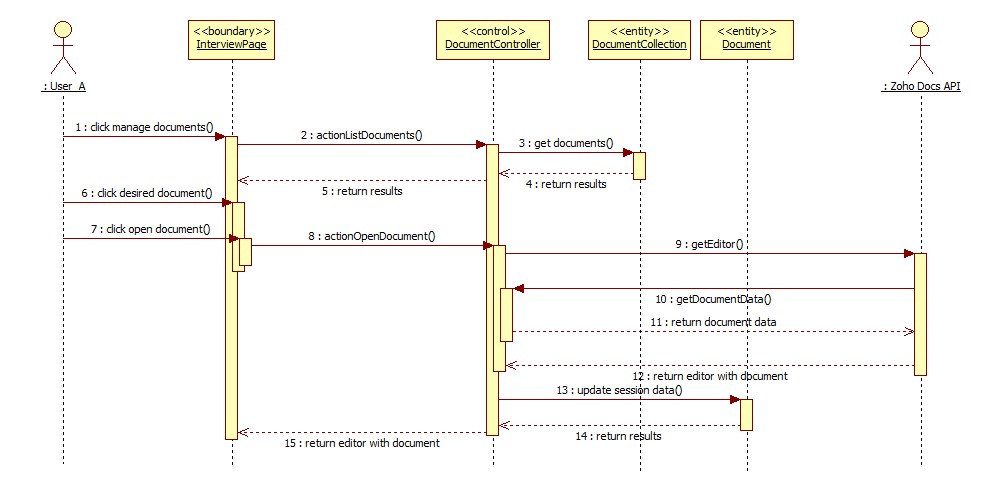
Send Reminder



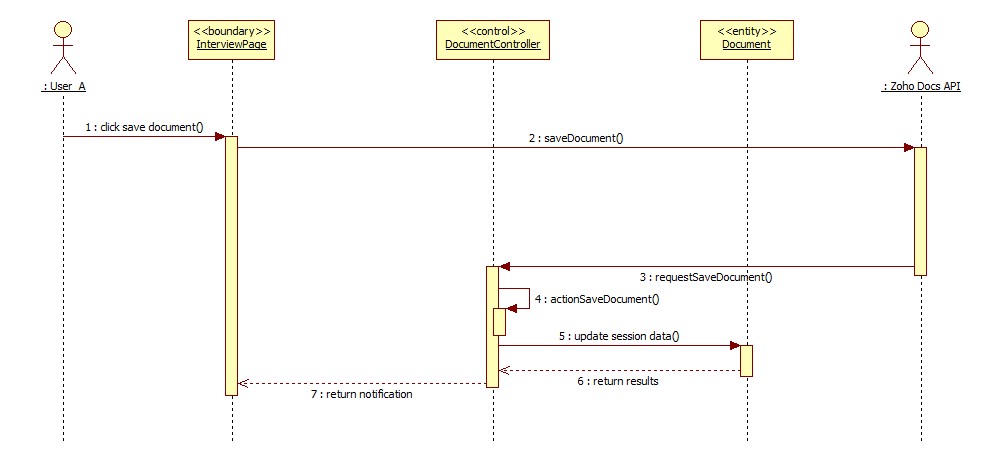
Create New Document



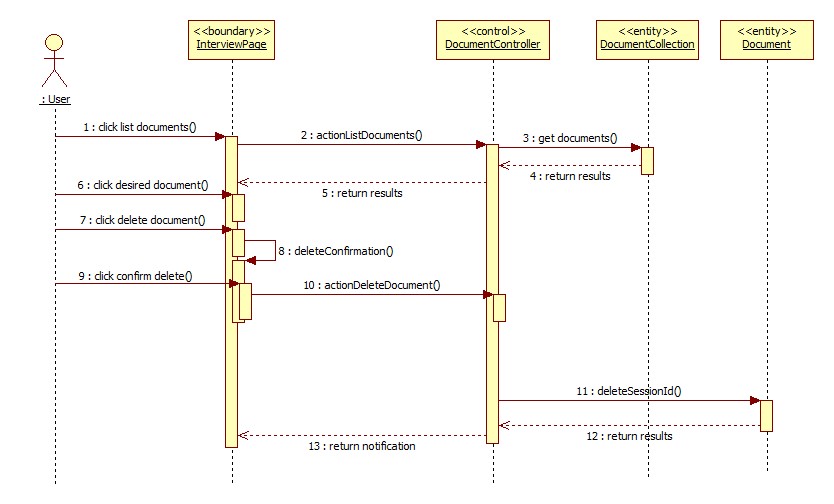
Open Document



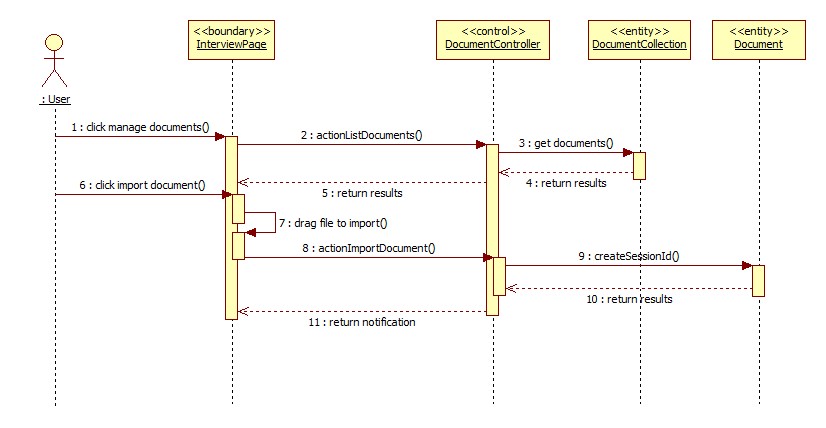
Save Document



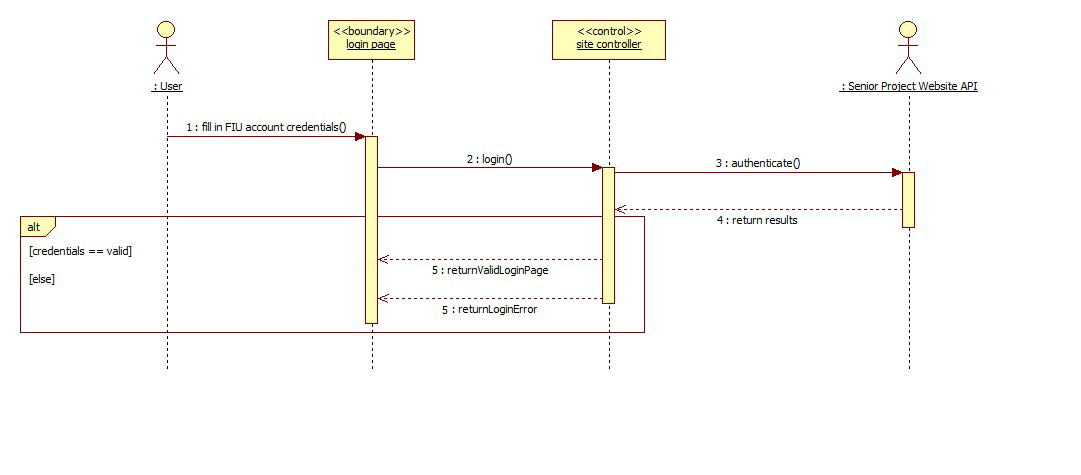
Delete Document



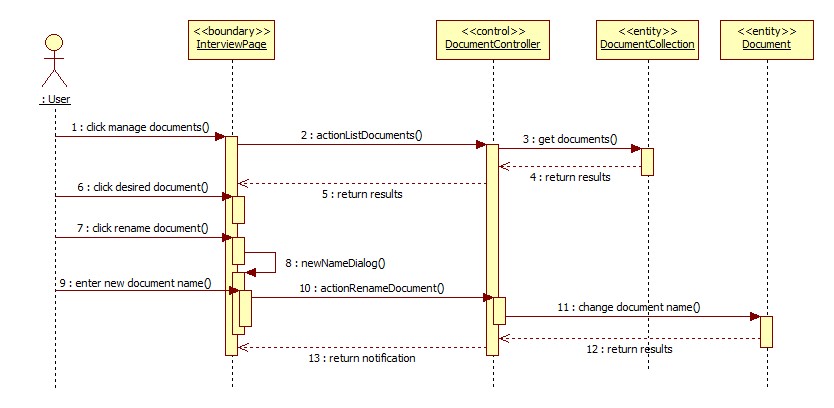
Import Document



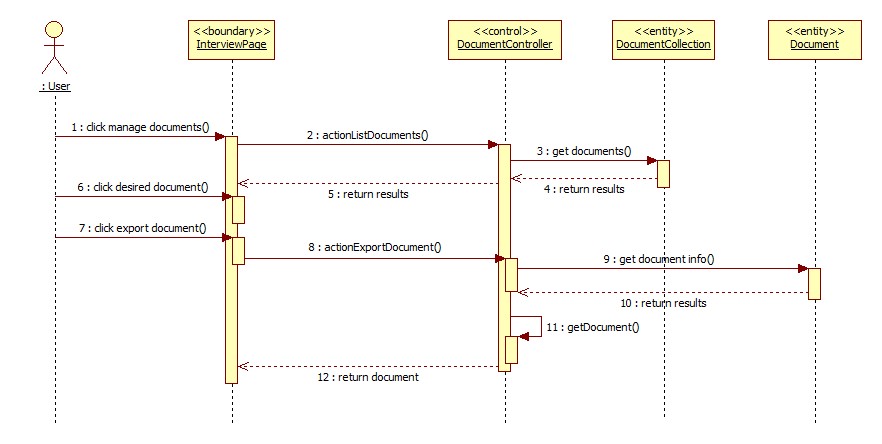
Login with FIU credentials



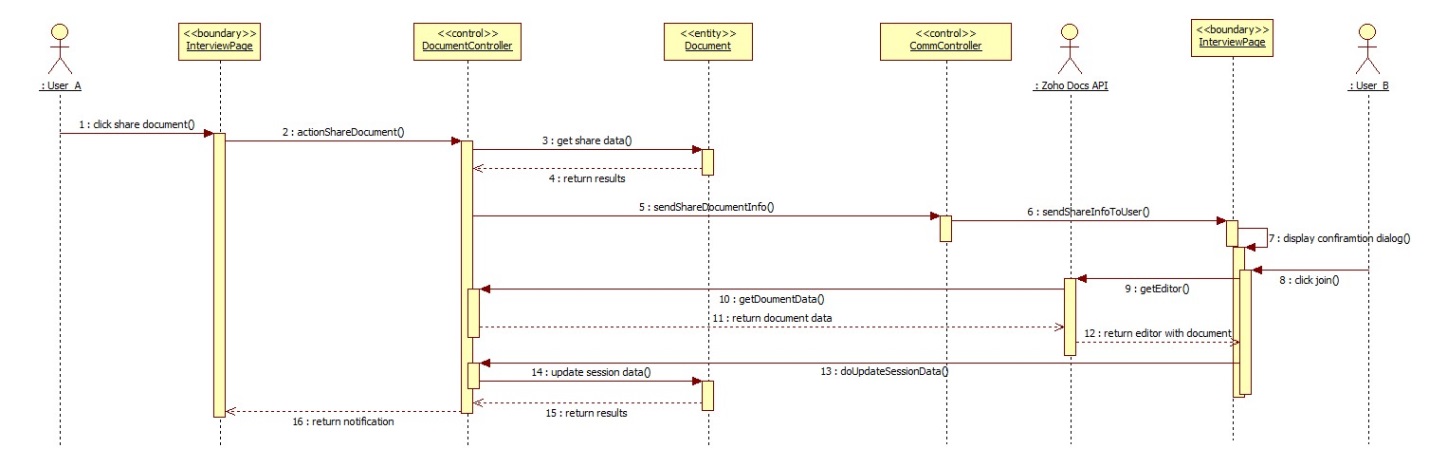
Rename Document



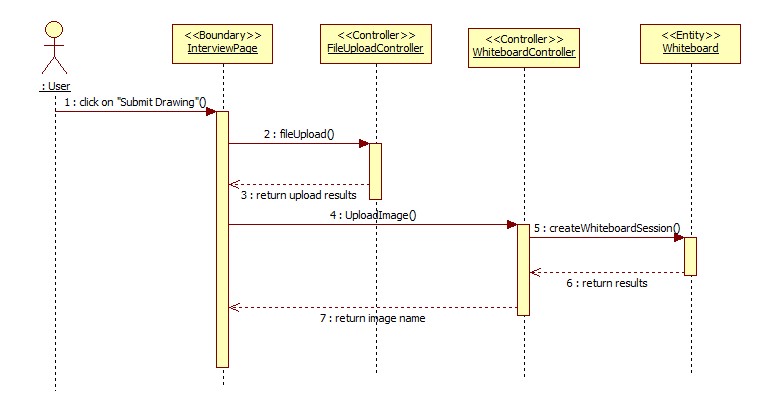
Export Document



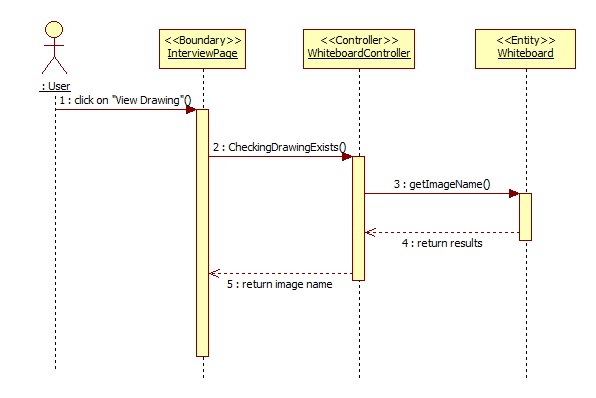
Share Document



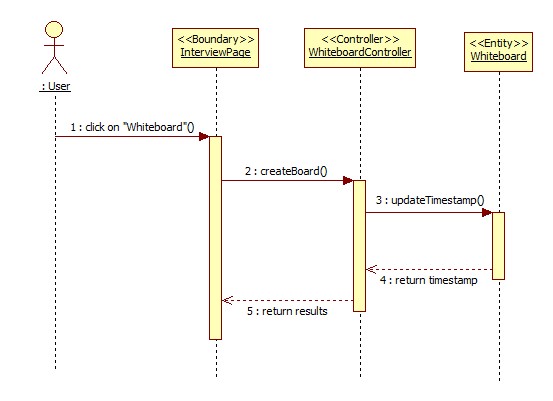
Upload Image



View Uploaded Image



Show or Restore Whiteboard

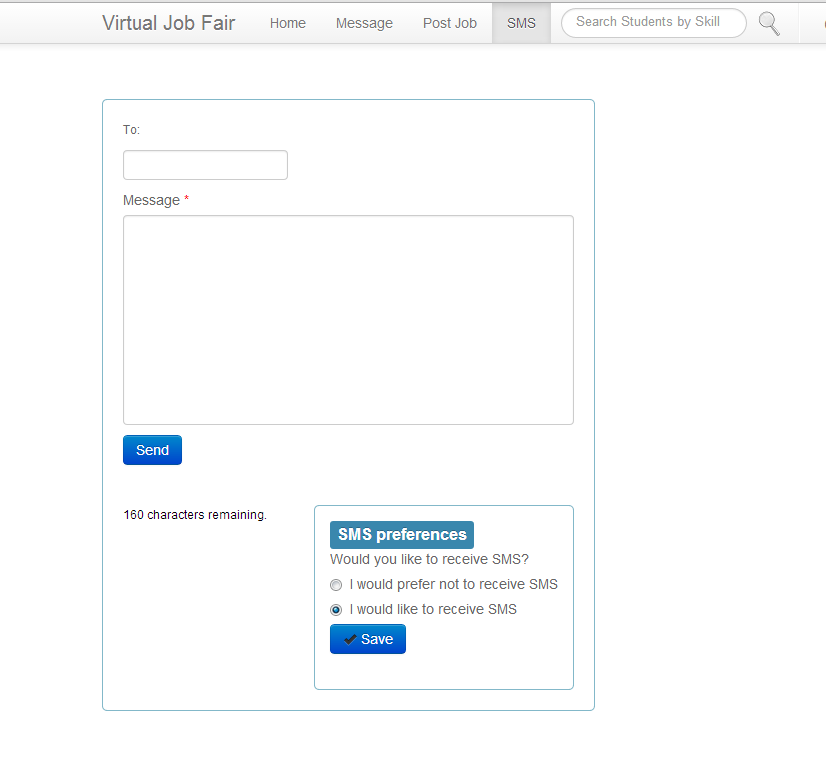


## Select Upload Image

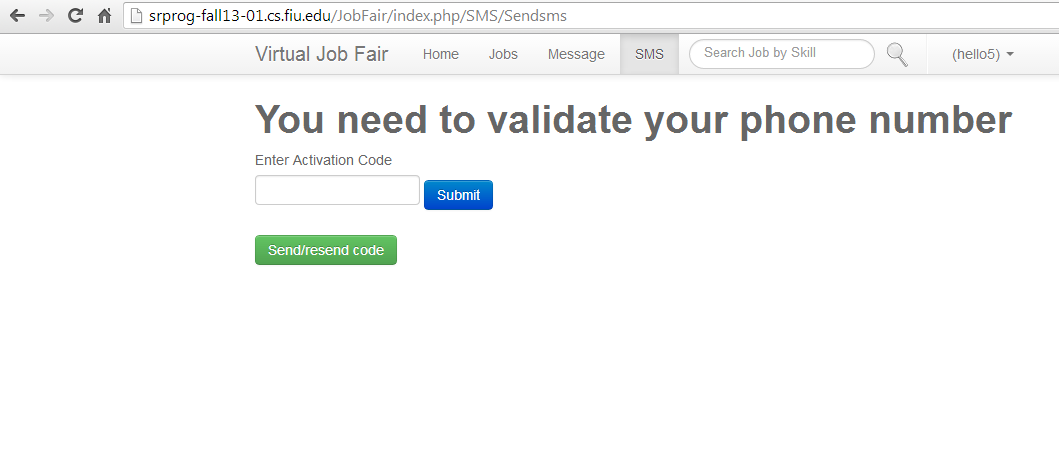
## 

## 6.5 Appendix E - User Interface designs.

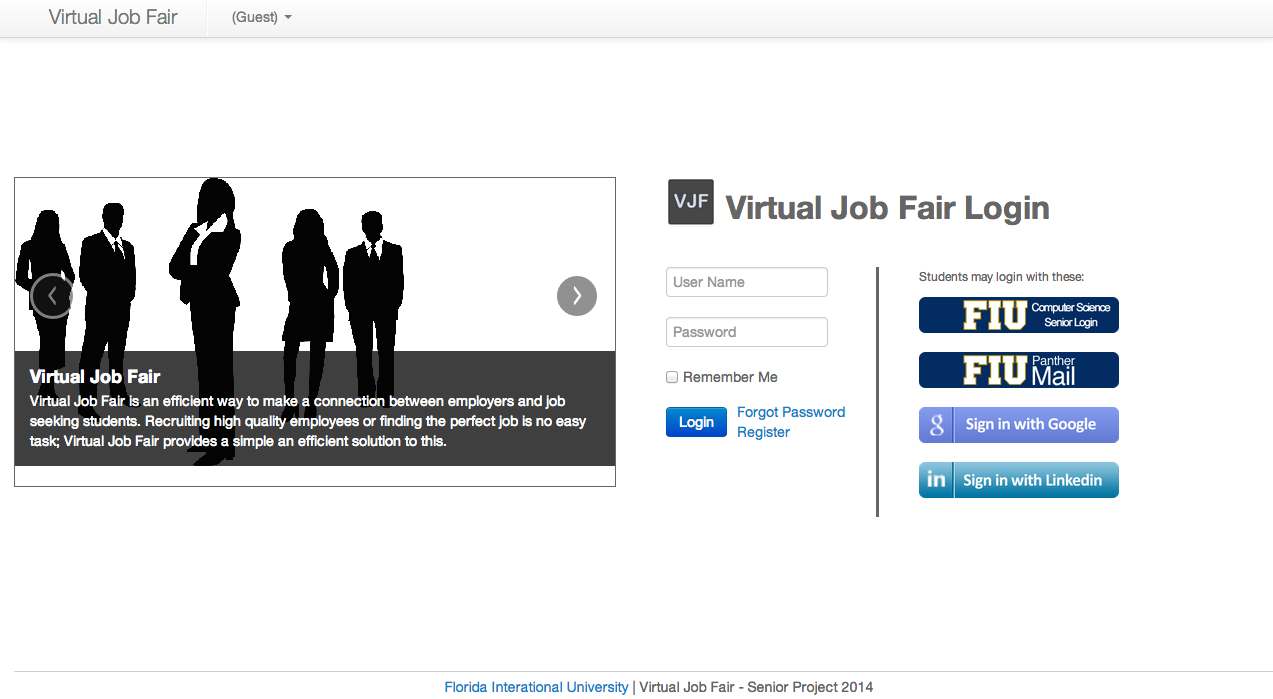
**Send SMS**

****

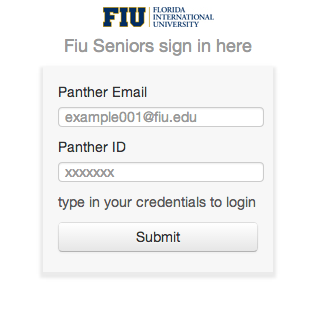
**Validate Phone Number**

****

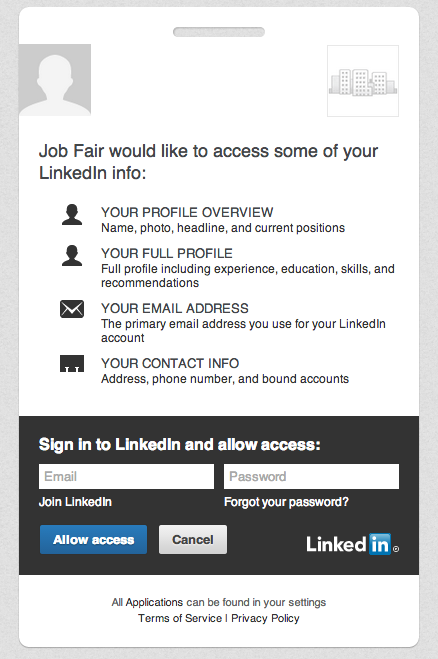
**Log In Page**



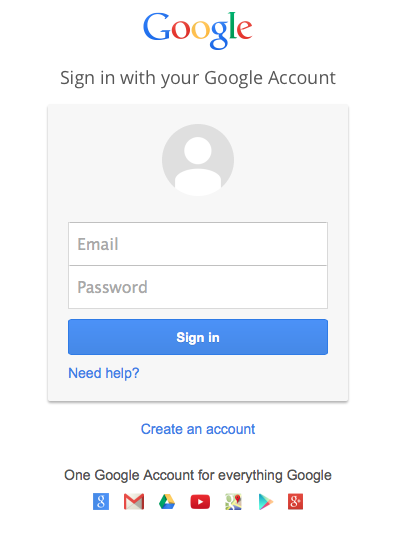
**Log In Using FIU SCIS Credentials**

****

**Log In Using LinkedIn Credentials**

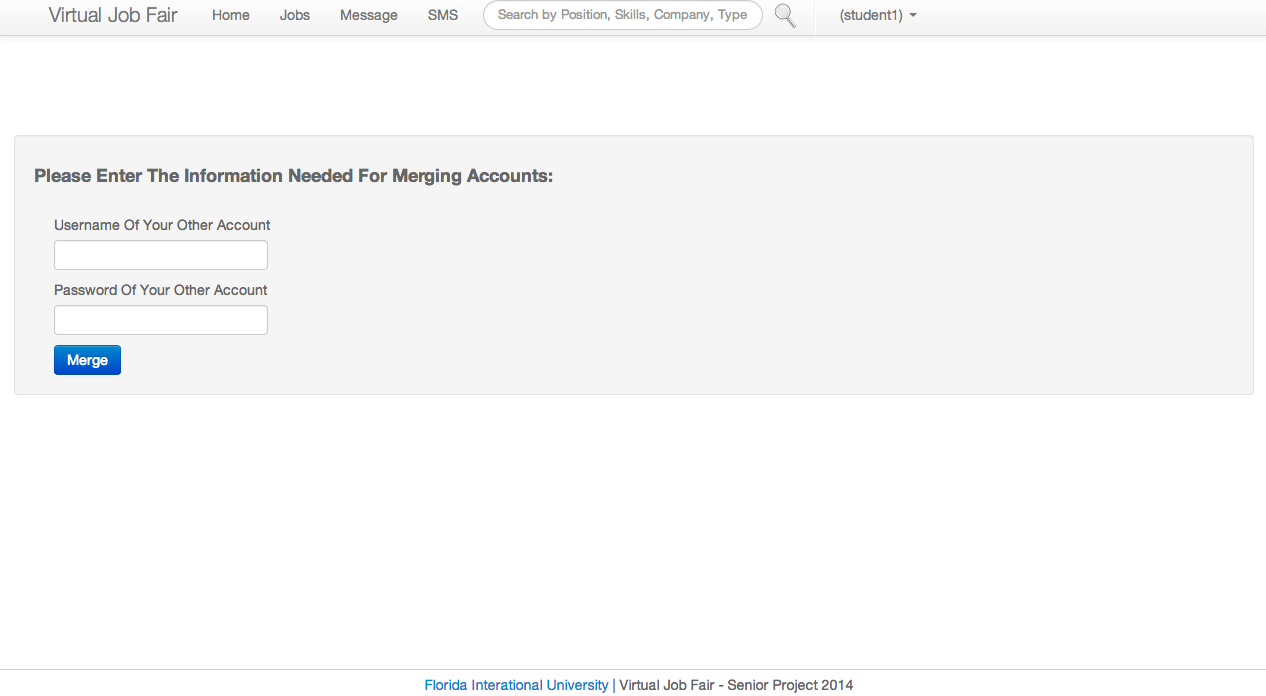


**Log In Using FIU and Google Credentials**

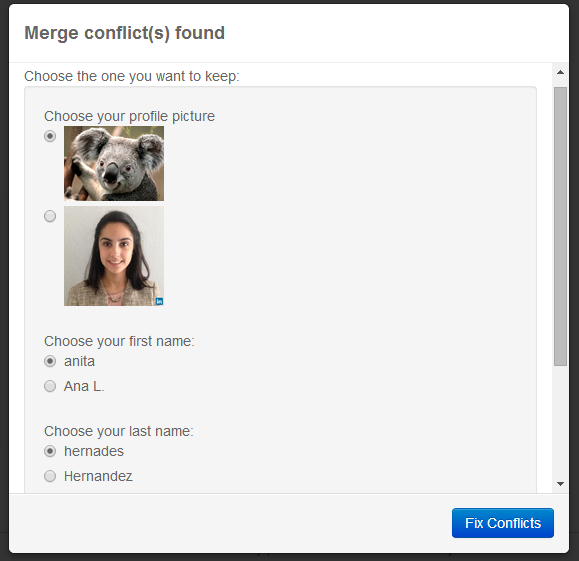


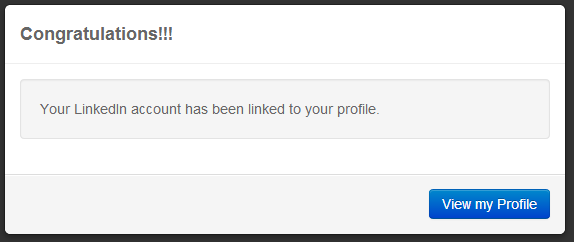


**Merge Accounts from Profile Drop Down Menu**

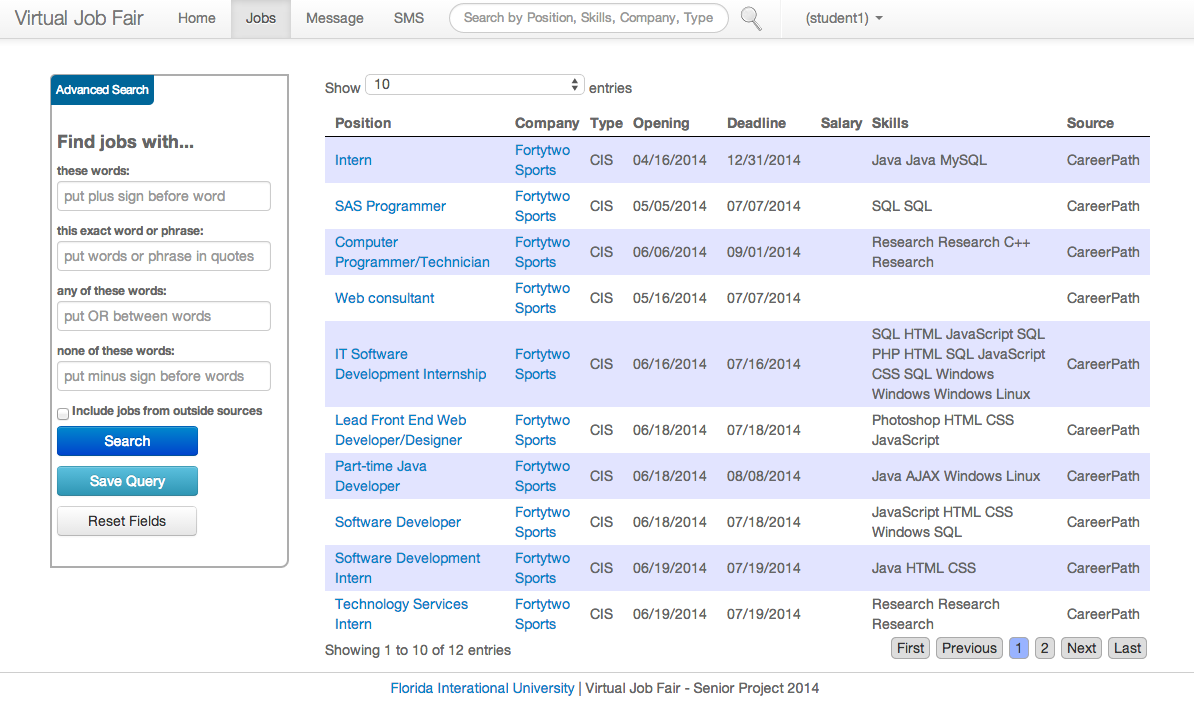


**Merge Conflicts when Linking / Merging Accounts**

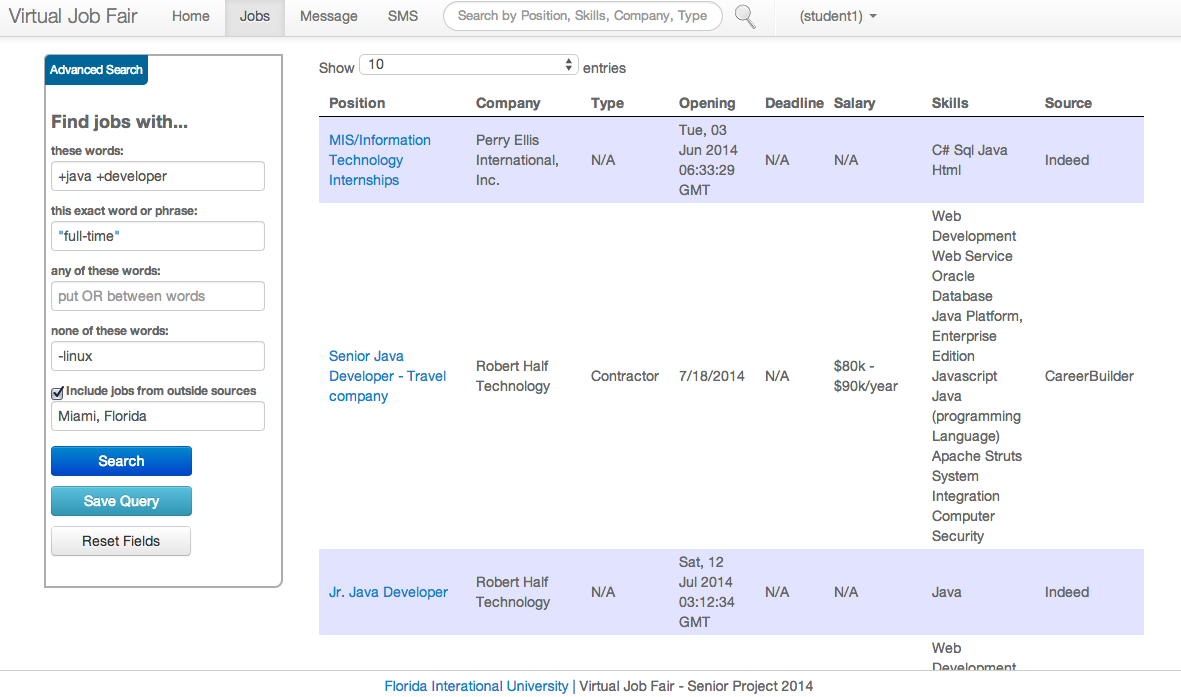




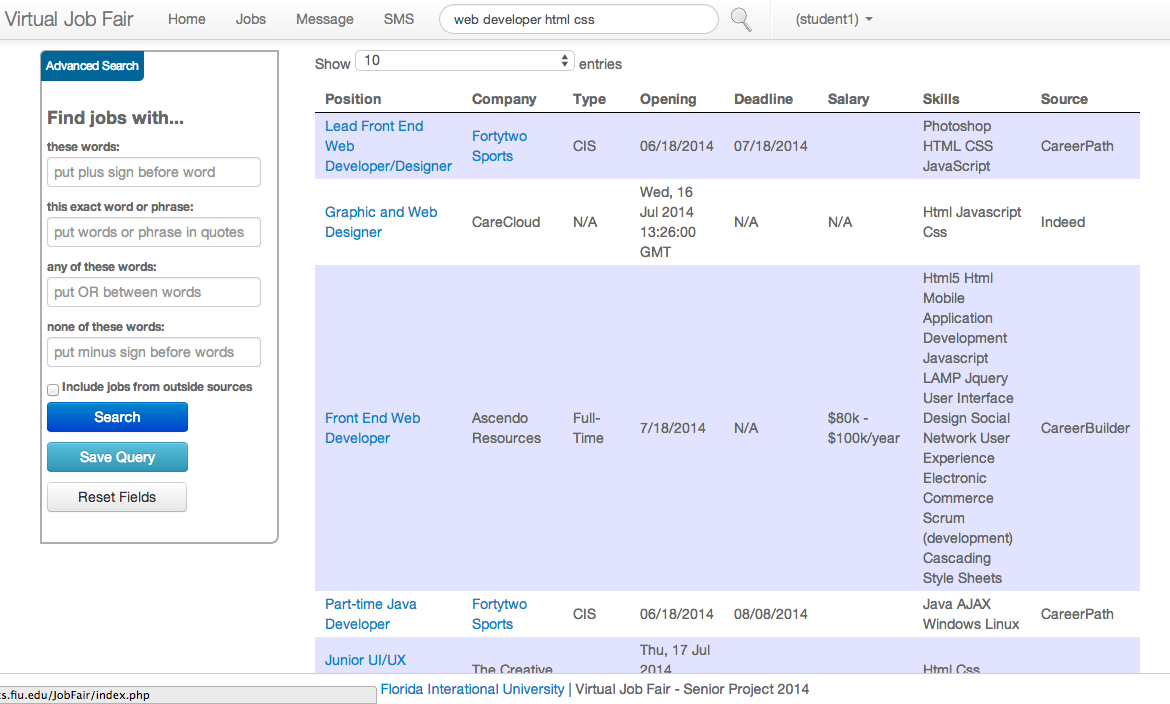
**Jobs Page**



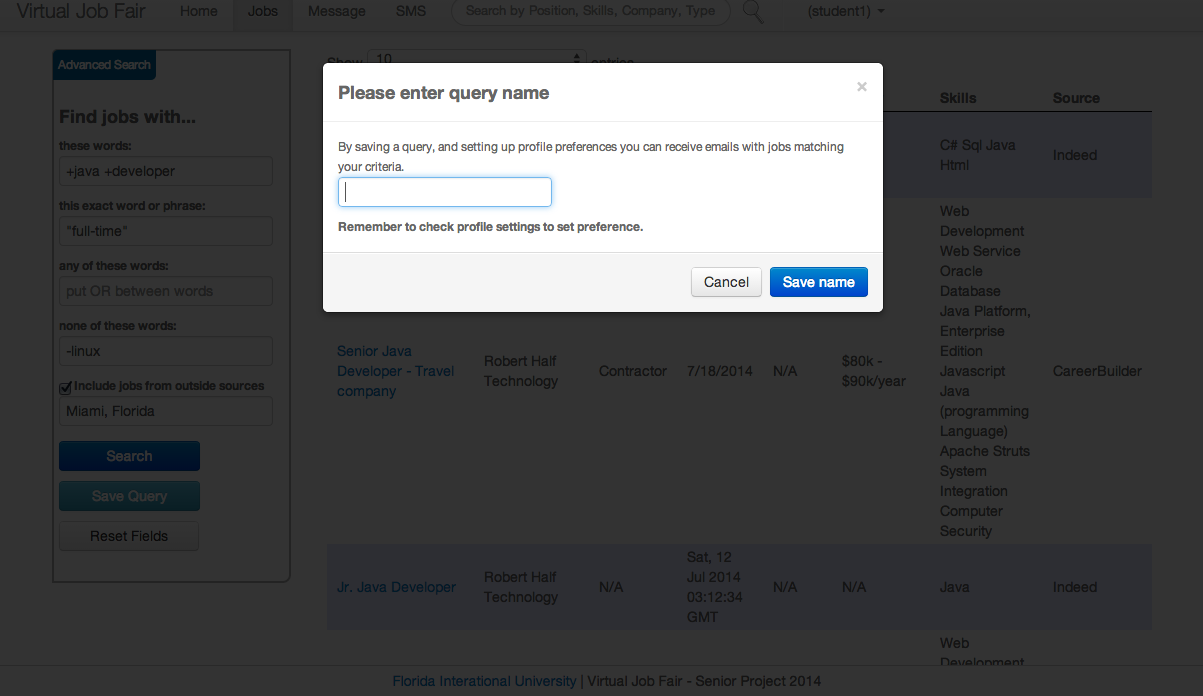
**Jobs Advanced Search**



**Job Search from Navigation Bar**

****

**Save Query from Advanced Search**



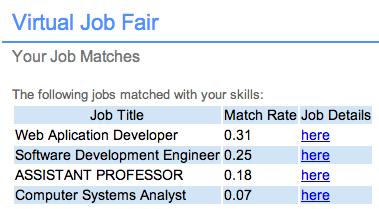
**Save Query Name Validation**



**Profile Settings for Query Preference**



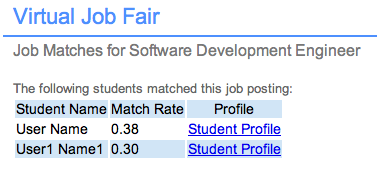
**Student Email Notification based on Skills**

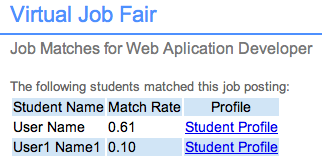


**Student Email Notification based on Saved Queries**

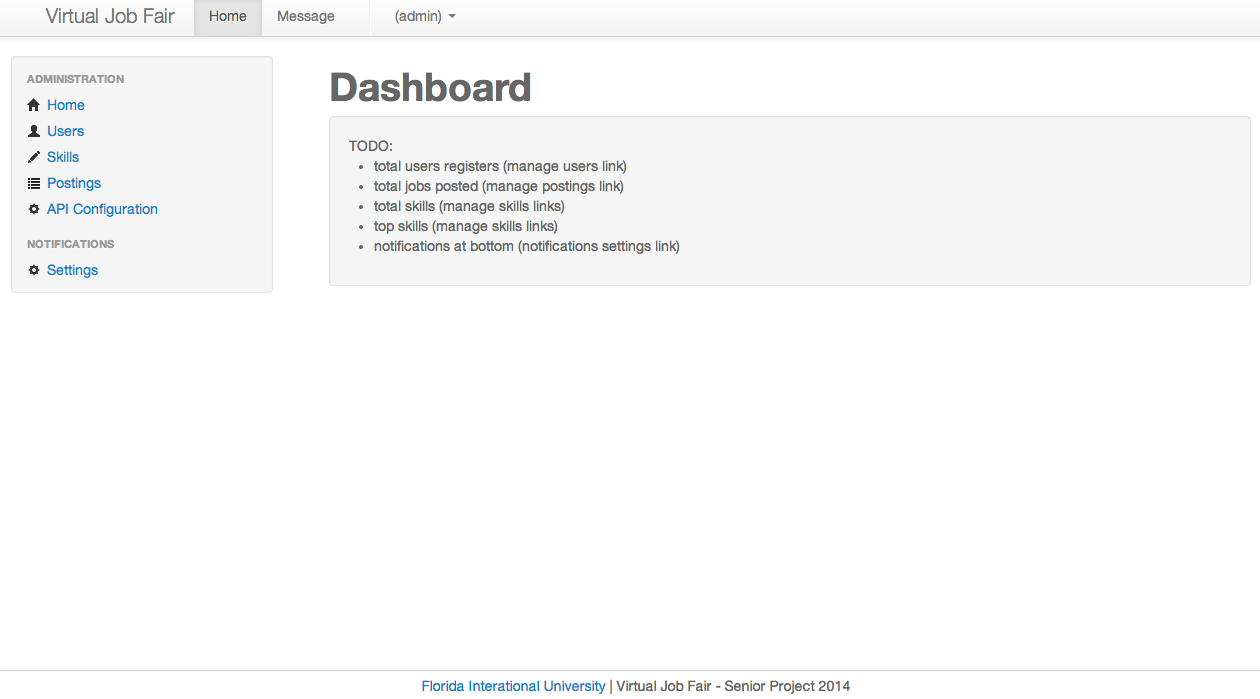


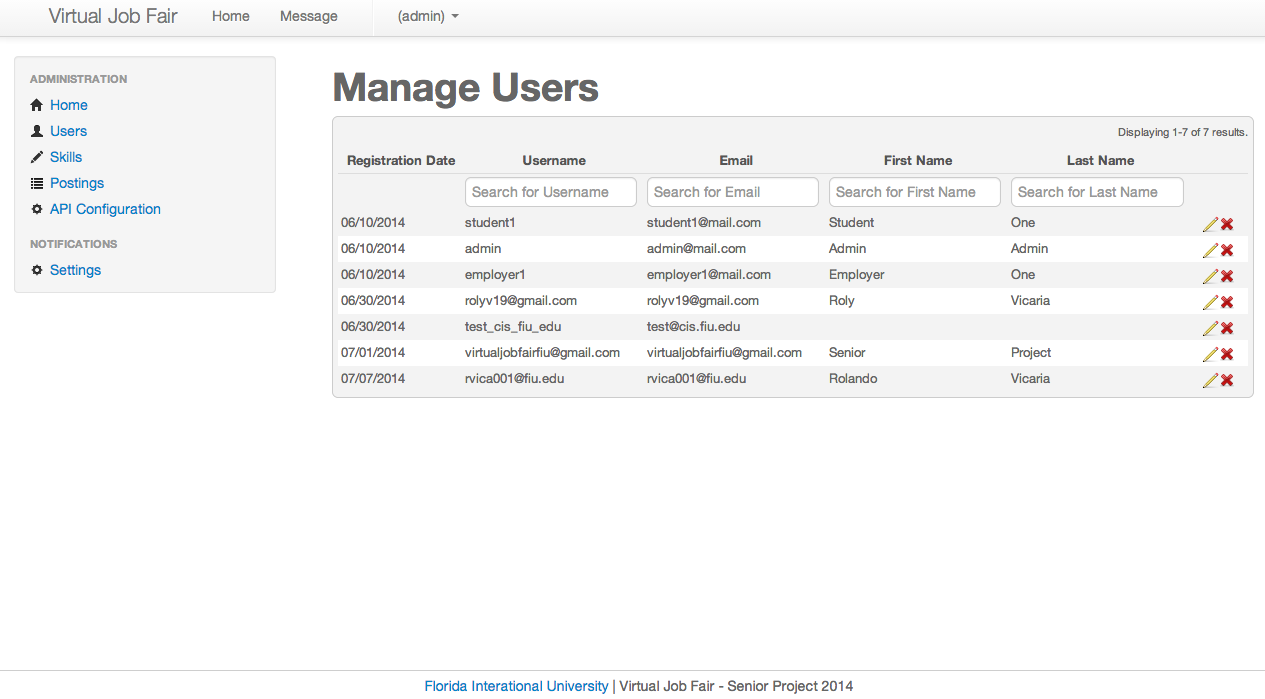
**Employer Email Notification based on Skills Matching per Job Posting**



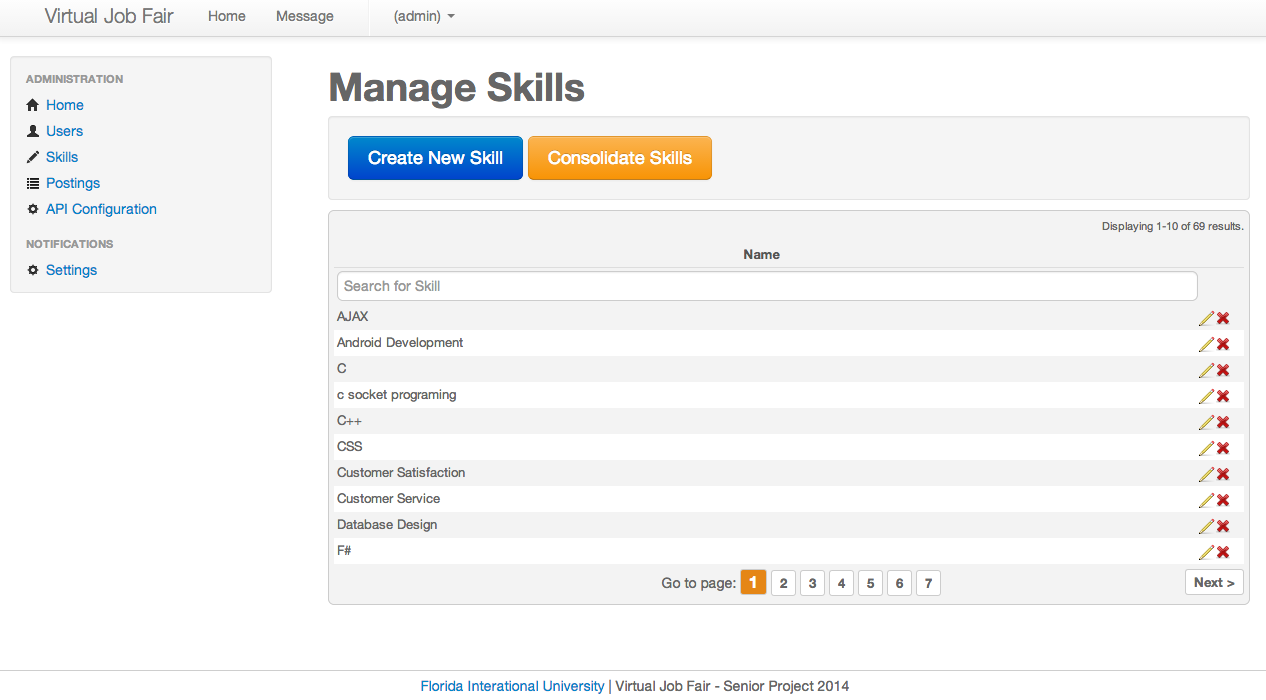


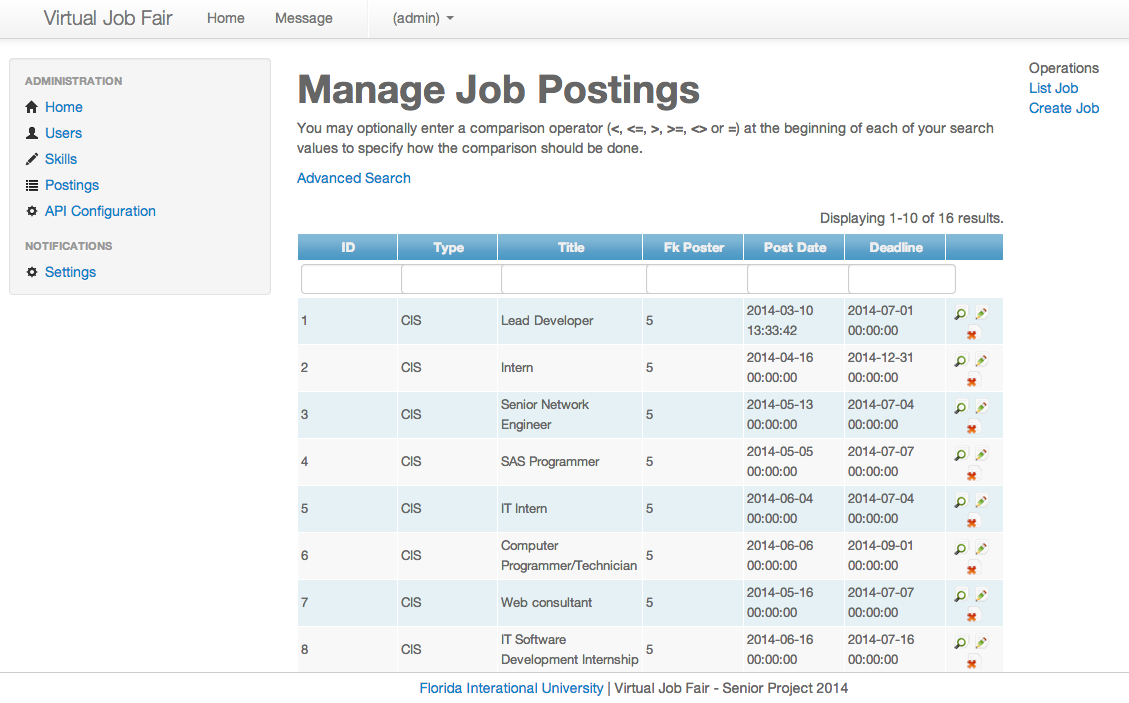
**Admin Dashboard**

****

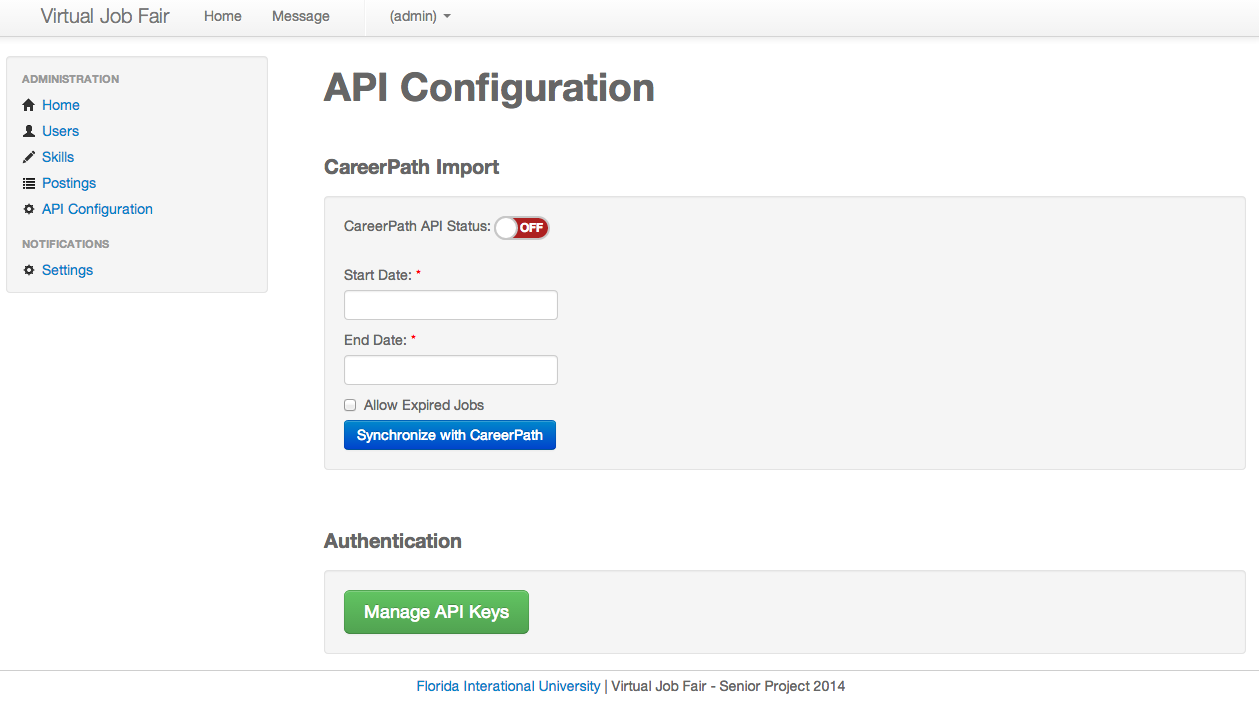
**Admin Manage Users**

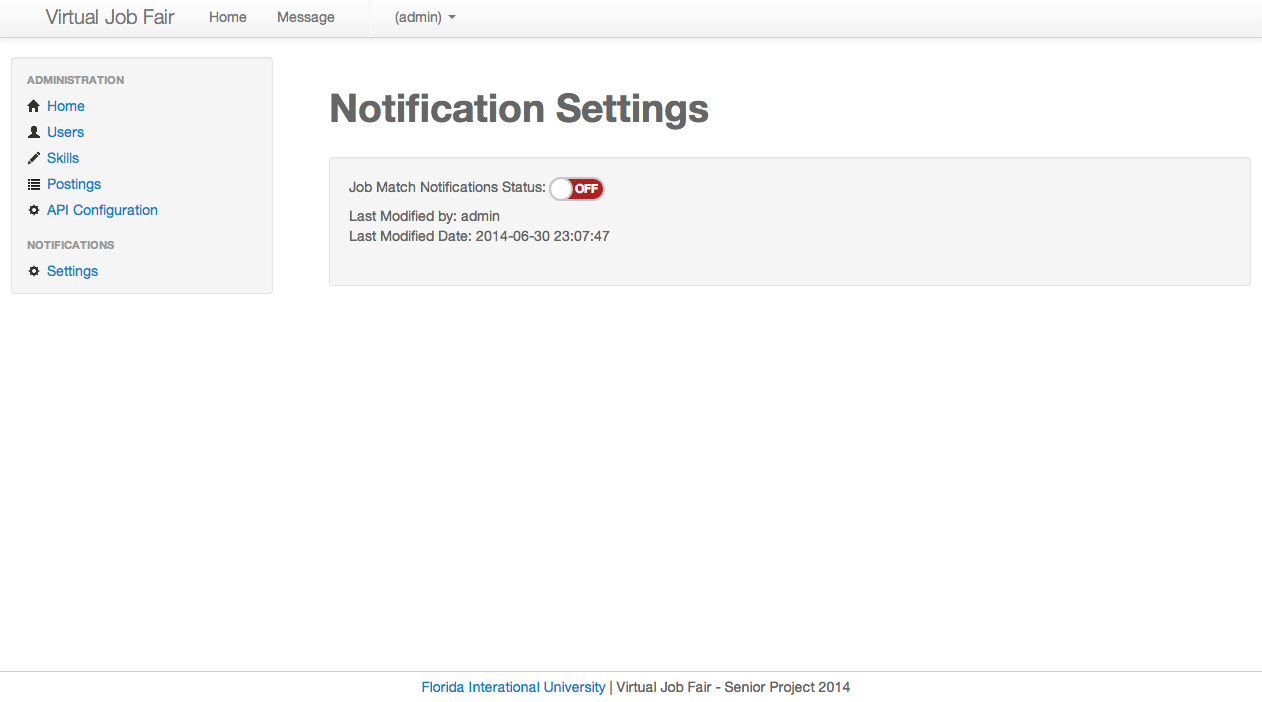
**Admin Manage Skills**



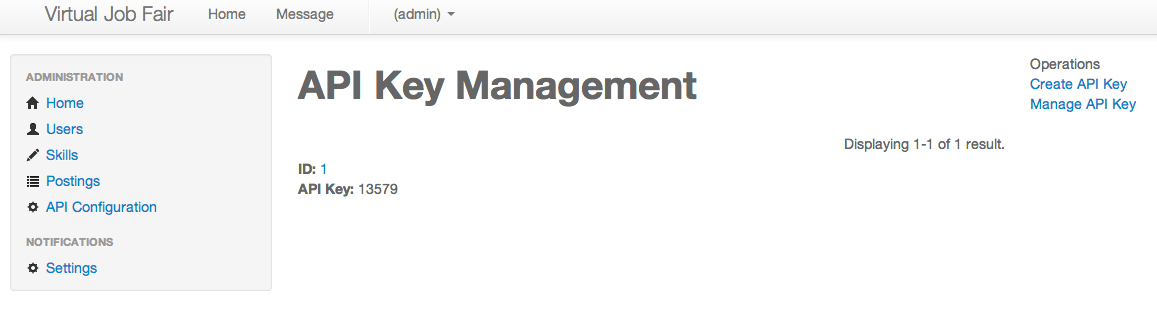
**Admin Manage Job Postings**

**Student API Configuration**

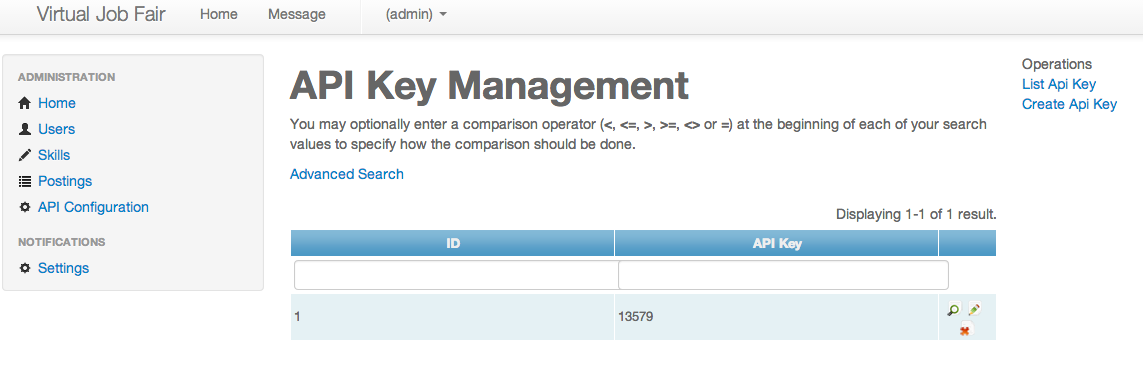


**Admin Notification Settings**

**Admin API Key Management**



**Admin API Key Management**



## 6.6 Appendix F - Diary of meeting and tasks.

During the duration of the project, our team did daily conference call stand ups with project manager, Rolando Vicaria to ensure adequate progress and discuss any problems we had encounter, as well as to receive feedback from our project manager.

**Meeting 1:**

**Date:** May 13, 2014

**Start Time:** 6:00PM

**End Time:** 8:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Read documentation, and get familiar with the code.

**Meeting 2:**

**Date:** May 15, 2014

**Start Time:** 7:00PM

**End Time:** 9:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Set up of the local environment.

**Meeting 3:**

**Date:** May 18, 2014

**Start Time:** 2:00PM

**End Time:** 4:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Re-Install IDEs for local environment. Setup BitBucket for code control.

**Assigned Tasks:**

* **Manuel B. :** Install PHPStorm, configure HTTP server, install local server WAMP, setup MySQL Workbench, setup git.
* **Ana H. :** Install PHPStorm, configure Apache, setup mysql command line interface, setup git.
* **Enio P. :**Configure IntelliJ, configure HTTP server, configure local mysql server, setup git.
* **Tomas A. :** Configure HTTP server, configure local mysql server, setup git.

**Meeting 4:**

**Date:** May 20, 2014

**Start Time:** 7:00PM

**End Time:** 9:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Move bitbucket repo to Gitlab, and set up local branches.

**Meeting 5:**

**Date:** May 22, 2014

**Start Time:** 7:00PM

**End Time:** 9:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta, Rolando Vicaria.

**Agenda:** Create tentative schedule and requirements.

**Assigned Tasks:**

* **Manuel B. :** Fix broken links in application.
* **Ana H. :** Remove search result page and research generalize search by keyword.
* **Enio P. :**Research existing SCIS job posting system.
* **Tomas A. :** Research existing notifications.

**Meeting 6:**

**Date:** June 3, 2014

**Start Time:** 7:00PM

**End Time:** 9:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta

**Agenda:** Check local branch integration with development branch and admin to production mode.

**Meeting 7:**

**Date:** June 11, 2014

**Start Time:** 6:00PM

**End Time:** 8:30PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta

**Agenda:** Automate deployment of MySQL DB schema. Evaluate using Yii migrations if not design another method.

**Meeting 8:**

**Date:** June 17, 2014

**Start Time:** 6:00PM

**End Time:** 9:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta

**Agenda:** PerformIntegrationtesting, and document problems.

**Meeting 9:**

**Date:** June 20, 2014

**Start Time:** 6:00PM

**End Time:** 8:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Rolando Vicaria.

**Agenda:** Arrange AgileZen work schedule based on priorities discussed during class presentation.

**Meeting 10:**

**Date:** June 22, 2014

**Start Time:** 2:00PM

**End Time:** 4:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena.

**Agenda:** Discuss requirement progress, problems, and tentative schedule for the remaining requirements.

**Meeting 11:**

**Date:** June 27, 2014

**Start Time:** 6:00PM

**End Time:** 9:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Discuss development repo problems, and agreed on approaches to avoid further problems with the development branch.

**Meeting 12:**

**Date:** July 3, 2014

**Start Time:** 5:00PM

**End Time:** 8:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** PerformIntegrationtesting, and document problems. Discuss development progress, and updated AgileZen.

**Meeting 13:**

**Date:** July 10, 2014

**Start Time:** 6:00PM

**End Time:** 8:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Discuss local branches integration issues, work on current requirements, and do debugging.

**Meeting 14:**

**Date:** July 13, 2014

**Start Time:** 3:00PM

**End Time:** 8:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta, Rolando Vicaria.

**Agenda:** Work on finalization of requirements, integration testing, and poster presentation.

**Meeting 15:**

**Date:** July 17, 2014

**Start Time:** 6:00PM

**End Time:** 8:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Discuss final requirements progress, and uploaded documents to Google Drive to be able to share the work more efficiently.

**Meeting 16:**

**Date:** July 18, 2014

**Start Time:** 6:00PM

**End Time:** 8:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Work on final documentation, and final PowerPoint presentation.

**Meeting 17:**

**Date:** July 19, 2014

**Start Time:** 12:00PM

**End Time:** 8:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Work on final documentation, and final PowerPoint presentation.

**Meeting 18:**

**Date:** July 20, 2014

**Start Time:** 12:00PM

**End Time:** 10:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Work on final documentation, and final PowerPoint presentation.

**Meeting 19:**

**Date:** July 21, 2014

**Start Time:** 4:00PM

**End Time:** 11:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Work on final documentation, finalization of requirements.

**Meeting 19:**

**Date:** July 22, 2014

**Start Time:** 10:00AM

**End Time:** 2:30PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Work on final documentation, finalization of requirements.

**Meeting 20:**

**Date:** July 23, 2014

**Start Time:** 10:00AM

**End Time:** 11:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Work on final documentation, finalization of requirements, fixing PowerPoint presentation, poster.

**Meeting 21:**

**Date:** July 24, 2014

**Start Time:** 10:00AM

**End Time:** 11:00PM

**In Attendance:** Manuel Bouza, Ana Hernandez, Enio Pena, Tomas Acosta.

**Agenda:** Work on final documentation, finalization of requirements, fixing PowerPoint presentation, poster. Rehearsal of final presentation.

## 7. References

1. Feasibility Study & Project Plan Document v1.0. Fall 2013.
2. Feasibility Study & Project Plan Document v2.0. Spring 2013.
3. Requirement Document v1.0. Fall 2013.
4. Requirement Document v2.0. Spring 2013.
5. Design Document v1.0. Fall 2013.
6. Design Document v2.0. Spring 2013.
7. Final Deliverable v1.0. Fall 2013.
8. Final Deliverable v2.0. Spring 2013.
9. "Bootstrap." *2.2.2 Documentation*. N.p., n.d. Web. 24 July 2014. <http://bootstrapdocs.com/v2.2.2/docs/>.
10. "Documentation." *Yii PHP Framework: Best for Web 2.0 Development*. N.p., n.d. Web. 24 July 2014. <http://www.yiiframework.com/doc/>
11. Chacon, Scott. *Pro Git*. Berkeley, CA: Apress, 2009. Online. <http://git-scm.com/book/en/Getting-Started>