**REQUIREMENT DOCUMENT**

CIS 4911 – Senior Project

**Virtual Job Fair 5.0**

Members:

**Rogelio Alonso**

Instructor:

**Masoud Sadjadi**

Mentor:

**Juan Caraballo**

FLORIDA INTERNATIONAL UNIVERSITY

**ABSTRACT**

Chapter 1 gives basic information about the Virtual Job Fair V4, 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 a Gantt chart 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 references.

Table of Contents

1.Introduction 3

1.1. Problem definition 3

1.2. Scope of the system 3

1.3-Definitions, Acronyms, and Abbreviations 3

1.4-Overview of document 4

2.Current System (limitations and problems) 5

2.1-Description of Current System (Limitations and Constraints) 5

3. Project Plan 8

3.1-Project Organization 8

3.1.1-Project Personnel Organization 9

3.2-Identification of Tasks, Milestones and Deliverables (work breakdown) 9

3.3-Cost Estimate 10

4. Proposed System Requirements 11

4.2 Analysis of System Requirements 31

4.2.1 Use case model 31

4.2.2 Static model 31

4.2.3 Dynamic model 31

5. Glossary 32

6. Appendix 32

6.1-Appendix A – Complete Use Cases 32

6.2-Appendix B – Use Cases Diagram using UML 74

6.3-Appendix C - Static UML diagram 79

6.4-Appendix D – Dynamic UML diagrams 81

6.5-Appendix E– User Interface designs. 87

6.6 Appendix F- Diary of meeting and tasks. 98

# Introduction

The introductory chapter gives some background information about the Virtual Job Fair v5.0 system. This chapter will talk about the problem that companies are facing as well some background on this problem. It will discuss the previously designed Virtual Job Fair website and its purpose.  Moreover this chapter will define and explain the definitions, acronyms and abbreviations of terms that will be used later in this deliverable. Finally this chapter contains an overview of the all of the project, which explains the information contained on each chapter.

## 1.1. Problem definition

Employers looking for talent are always interested in filling out positions with the best possible people. 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 logistical capabilities to seek for potential candidates in different locations, 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. We will also provide a way for employers to search for specific skills on the students, as well as save queries of their own which they can be notified according to whichever time they want. Therefore providing the employer and potential employee a more easy and friendly way to find their match. Moreover, we will grant access to any visitor as an Employer or a Student Guest with limited functionality.

## 1.2. Scope of the system

One of the main purposes of the Virtual Job Fair v5.0 is to expand the scope of the previous versions. While in past versions of the job fair system, the employers needed to be registered to post jobs or tap talent, and the students needed to have an account registered in order to benefit from the job postings search. Likewise, the PDF and video resume was lacking an update feature to constantly keep these files up to date. The system had number of stability issues and the advance search result was unable to exclude unwanted keywords.

## 1.3-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

-          **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 résumé

**Acronyms**

-      **VJF**: Virtual Job Fair

-          **FIU:** Florida International University

-          **GPA:** Grade point average

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

## 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)

## 2.1-Description of Current System (Limitations and Constraints)

The current system was modified in the fall 2014. 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. Furthermore, it offers a messaging system, through which employers and students can communicate privately.  It also allows users to save their search queries as well as lets them get a notification with the frequency defined by the student. Additionally, employers who have account on the system can save queries and search for students based on skill sets. Finally, the administrator actor is provided with a dashboard displaying useful statistics, and can create user accounts from the administrator menu.

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.

**- Employer limitation:** the systems lacks a way to connect a newly registered employer account waiting for account revision and activation. The registration flow for the employer user ends due to the need of an admin monitoring new accounts and granting access to it.

**- Guests limitation:** the current system does not allow visitors, either for employers or students, therefore and account must be created at all cases.

- **Inability to students to update resumes:** the current system only provides a way to upload a PDF/Video resume; but, does not allow to replace them and update them.

**- Students search for job is defective:** The current system boolean search results include unwanted keywords, and is unable to combine different types of search criteria.

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

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

-**Creation of account for administrator:** the current system does not allow creating of new account from the administrators account.

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

**- 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.

**- The system does not count with a regression testing script:** the current system does not provide a way to test if existent functionalities were affected by new ones.

# 3. Project Plan

The project plan chapter introduces VJF from a project management perspective. Firstly, the project organization is described, with the roles for each member listed. Then, hardware and software requirements for the development of the project will be listed. After that, milestones, tasks, and deliverables will be listed.

## 3.1-Project Organization

For this particular project each of the two members will be adding some major functionality to the Virtual Job Fair that will show off our skills as programmer.

I, Rogelio Alonso, will be responsible for gathering the stories from the product owner, refine those stories into tasks and plan 5 sprints of 10 days each. Each sprint after the stories are refined, I would make the design and implement, test and integrate back to the development server. Below is brake down of the work done in each sprint:

* Sprint 1:
  + Add two new actors to the system. (Guest Employer and Guest Student)
* Sprint 2:
  + Test all the student, employer actor user stories while providing an automated regression testing script.
* Sprint 3:
  + Allow student user to update/replace his video resume, PDF resume and profile picture. Increased resume video size limit to 80 MB. Provide more system stability to the following sections of the project:
    - Importing Linked In profile picture.
    - Change password functionality validation.
    - Profile page visualization.
    - Student registration functionality validation.
* Sprint 4:
  + Conduct a Feasibility Study to include and integrate a search engine with the system. Provide more system stability to the following sections of the project:
    - Editing the student/employer basic info section.
    - Deleting student/employer users from the admin section.
    - Messaging system.
    - Search jobs.
    - Visibility of key GUI buttons.
* Sprint 5:
* Install, configure and Index data for Solr search engine. Provide more system stability to the following sections of the project:
  + - Indeed API.

### 3.1.1-Project Personnel Organization

|  |  |  |
| --- | --- | --- |
| Team Member | Primary Task | General Task |
| Rogelio Alonso | SCRUM MASTER / Developer / Tester | Sprint Planning / Design / Documentation / Implementation / Testing |

## 

## 3.2-Identification of Tasks, Milestones and Deliverables (work breakdown)

|  |  |  |
| --- | --- | --- |
| Week 1 | 1/12/15 | Introduction: Course, Projects, Students, Groups |
| Week 2 | 1/19/15 | Accept Project, Assign Roles, User Stories, Requirement Analysis and Product Backlog |
| Week 3 | 1/26/15 | Feasibility Study, Project Plan, System Design and Object Design |
| Week 4 | 2/2/15 | Project Introduction, Sprint 1 Planning, Implementation and Testing |
| Week 5 | 2/9/15 | Project Introduction, Sprint 1 Integration and Testing. Sprint 1 Review |
| Week 6 | 2/16/15 | Project Report, Sprint 2 Planning, Implementation and Testing |
| Week 7 | 2/23/15 | Project Report, Sprint 2 Integration and Testing. Sprint 2 Review |
| Week 8 | 3/2/15 | Project Report, Sprint 3 Planning, Implementation and Testing |
| Week 9 | 3/9/15 | Spring Break |
| Week 10 | 3/16/15 | Project Report, Sprint 3 Integration and Testing. Sprint 3 Review |
| Week 11 | 3/23/15 | Project Report, Sprint 4 Planning, Implementation and Testing |
| Week 12 | 3/30/15 | Project Report, Sprint 4 Integration and Testing. Sprint 4 Review |
| Week 13 | 4/6/15 | Project Report, Sprint 5 Planning, Implementation and Testing |
| Week 14 | 4/13/15 | Project Report, Sprint 5 Integration and Testing. Sprint 5 Review |
| Week 15 | 4/20/15 | Finalize Documents, Poster |
| Week 16 | 4/27/15 | Videos, Deliverable and Presentation |

## 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, April 20th, 2015.

|  |  |
| --- | --- |
| **Item** | **Item Cost** |
| Yii Framework | $0.00 |
| Twitter Bootstrap | $0.00 |
| Easy RTC | $0.00 |
| Solr Search Engine | $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 v5.0, which would expand upon the already-implemented Virtual Job Fair v4.0. The main idea behind Virtual Job Fair v5.0 is to improve stability behind the search, messaging, registration, profile update and administration functionalities. Concurrently, we will introduce two new actors to the system and present a search engine for future developers to use while querying the system. 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**

Ø  **Usability**: The register form is simple and easy to follow.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 2 seconds.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **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.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø   **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: The edit form is simple and easy to follow.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: Starting a video interview is simple and understandable.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 3 seconds when connecting.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: The upload form is simple and easy to follow.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1-5 seconds, depending on the file size.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: The upload form is simple and easy to follow.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **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.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within one 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: The corresponding form is easy to complete and follow.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: The corresponding form is easy to complete and follow.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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**

Ø  **Usability**: Students are presented with a user-friendly interface that is easy to complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: Students are presented with a clear and simple interface to send messages.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: The search form is easy to follow and complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow employers to post jobs**

Ø  **Usability**: The post job form is easy to understand and complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow employers to close a job posting**

Ø  **Usability**: The closing of a post is easy to complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: the addition of skills to a post is simple to complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second when adding each skill.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: The search form is simple to submit.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow employers to view student profiles**

Ø  **Usability**: The view of a student profile is easy to understand.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within one 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow employers to send messages to students**

Ø  **Usability**: Employers are presented with a clear and simple interface to send messages.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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**

Ø  **Usability**: The virtual handshake form is easy complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: Disabling a user is simple to complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: Closing a job post is simple to complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: The validation of an employer is done by one click and is simple to complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: This is required for a user to log in. Form is simple and easy to follow.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: Storing user password in a secure way without user intervention.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

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

Ø  **Usability**: Security measure for system. Interface is simple to complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easy to maintain and make appropriate changes.

**Allow users to delete a document.**

Ø  **Usability:** The document deletion interface should be intuitive, easy to understand and navigate.

Ø  **Reliability:** The system should perform correctly 99% of the time.

Ø  **Performance:** The system should respond in less than 1 second.

Ø  **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.**

Ø  **Usability:** The import document interface should be intuitive, easy to understand and navigate.

Ø  **Reliability:** The system should perform correctly 99% of the time.

Ø  **Performance:** The system should respond in less than 3 seconds.

Ø  **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.**

Ø  **Usability:** The document open interface should be intuitive, easy to understand and navigate.

Ø  **Reliability:** The system should perform correctly 99% of the time.

Ø  **Performance:** The system should respond in less than 3 seconds.

Ø  **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**

Ø  **Usability**:  Sending an SMS should not take more than 15 seconds for a novice user.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**:  The SMS should be sent in under 2 seconds after user presses send

Ø  **Supportability**:  Sending SMS page should be supported by IE, Firefox, Chrome and Safari.

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

Ø  **Usability**:  Not applicable

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: Users should receive a reminder 30 minutes before scheduled interview

Ø  **Supportability**:  Not applicable

**Allow users to confirm their phone numbers**

Ø  **Usability**:  Confirming a phone number should take less than 30 seconds for inexperienced users

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: Sending authentication code and validating it should take less than 2 seconds respectively.

Ø  **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.**

Ø  **Usability**: The connection with LinkedIn should be easy to follow. Users will enter their LinkedIn credentials and get appropriate data that the user allowed.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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**

Ø  **Usability:** The system should provide an easy and integrated login process for FIU SCIS Seniors using the school UNIX account.

Ø  **Reliability:** The system should perform correctly 99% of the time.

Ø  **Performance:** The system should respond in less than 1 second.

Ø  **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**

Ø  **Usability:** The system should provide an easy and integrated login process for FIU students using the school UNIX account.

Ø  **Reliability:** The system should perform correctly 99% of the time.

Ø  **Performance:** The system should respond in less than 1 second.

Ø  **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**

Ø  **Usability:** The system should provide an easy and integrated login process for Google accounts.

Ø  **Reliability:** The system should perform correctly 99% of the time.

Ø  **Performance:** The system should respond in less than 1 second.

Ø  **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**

Ø  **Usability:** The system should provide an easy and integrated login process for LinkedIn accounts.

Ø  **Reliability:** The system should perform correctly 99% of the time.

Ø  **Performance:** The system should respond in less than 1 second.

Ø  **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, and position on navigation bar**

Ø  **Usability**: The search form is easy to follow and complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 2 second.

Ø  **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**

Ø  **Usability**: The search form is easy to follow and complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 3 second.

Ø  **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**

Ø  **Usability**: The search form is easy to follow and complete.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 3 second.

Ø  **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**

Ø  **Usability**: The search button should be press.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 3 second.

Ø  **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**

Ø  **Usability**: The search button should be press.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 3 second.

Ø  **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**

Ø  **Usability**: The search button should be press.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 3 second.

Ø  **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**

Ø  **Usability**: The profile preference are easy to setup.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should send email within 1, 7 or 30 days.

Ø  **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**

Ø  **Usability**: The student should be able to enter name with ease.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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**

Ø  **Usability**: The system should provide a save button.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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**

Ø  **Usability**: The system should provide a checkbox.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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**

Ø  **Usability**: The system should provide a delete icon next to query.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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**

Ø  **Usability:** The system should provide an easy and integrated linking process for third parties accounts.

Ø  **Reliability:** The system should perform correctly 99% of the time.

Ø  **Performance:** The system should respond in less than 1 second.

Ø  **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.**

Ø  **Usability:** The system should provide an easy and integrated interface for linking accounts.

Ø  **Reliability:** The system should perform correctly 99% of the time.

Ø  **Performance:** The system should respond in less than 1 second.

Ø  **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.**

Ø  **Usability:** The system should provide an easy and integrated interface for recovering password.

Ø  **Reliability:** The system should perform correctly 99% of the time.

Ø  **Performance:** The system should respond in less than 1 second.

Ø  **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.**

Ø  **Usability**: The form should be easy to use, Administrators can specify from what day to what day, and submit.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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.**

Ø  **Usability**: The interface should be easy to use.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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.**

Ø  **Usability**: The interface should be easy to use.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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.**

Ø  **Usability**: The form should be easy to use, Administrators can specify from what day to what day, and submit.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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.**

Ø  **Usability**: The interface should be easy to use.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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.**

Ø  **Usability**: The interface should be easy to use.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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.**

Ø  **Usability**: The interface should be easy to use.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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.**

Ø  **Usability**: The interface should be easy to use.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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**

Ø  **Usability**: The button to enable/disable should be easy to use.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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**

Ø  **Usability**:  The button to enable/disable should be easy to use.

Ø  **Reliability**:  The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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**

Ø  **Usability**:  The emails notifications should be easy to use.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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**

Ø  **Usability**:  The emails notifications should be easy to use.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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**

Ø  **Usability**:  The emails notifications should be easy to use.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **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 administrator to create student account**

Ø  **Usability**: The form should be easy to use, Administrator will only have to fill the most basic information about the student.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow administrator to create employer account**

Ø  **Usability**: The form should be easy to use, Administrator will only have to fill the most basic information about the employer.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow administrator to view website and user statistics.**

Ø  **Usability**: The interface should be easy to see and should display all the important information.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow student to search saved queries from advance search.**

Ø  **Usability**: The form should be easy to use, student will only have to select which query he wants to run.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 3 second.

Ø  **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow student to search multiple saved queries at the same time from advance search.**

Ø  **Usability**: The form should be easy to use student will only have to select which queries he wants to run.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow student to run search on advance search form and saved query at the same time.**

Ø  **Usability**: The form should be easy to use student will only have to select which queries he wants to run and type information into the advanced search form.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow student to receive notifications on saved queries dynamically.**

Ø  **Usability**: The student should have a box to type the interval of days he wants to be notified in between.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow employers to search for a job based on skill, location, zip code, school, major, graduation date, experience and position on the advance search.**

Ø  **Usability**: The advanced search form should be easy to understand and provide enough explanation to the user of how it works.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow employers to save queries.**

Ø  **Usability**: The employer should be able to easily save queries.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow employers to search saved queries form advanced search.**

Ø  **Usability**: The form should be easy to use employer will only have to select which query he wants to run.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow employers to search multiple saved queries at the same time form advanced search.**

Ø  **Usability**: The form should be easy to use employer will only have to select which query he wants to run.

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow employer to run search on advance search form and saved query at the same time.**

Ø  **Usability**: The form should be easy to use employer will only have to select which queries he wants to run and fill the information in the advanced search form

Ø  **Reliability**: The system should perform correctly 99% of the time.

Ø  **Performance**: The system should respond within 1 second.

Ø  **Supportability**: The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow employer to receive notification on saved queries dynamically.**

Ø **Usability:** The employer should have a box to type the interval of days he wants to be notified in between.  
  
Ø  **Reliability:** The system should perform correctly 99% of the time.  
  
Ø  **Performance:** The system should respond within 1 second.  
  
Ø  **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow system to retrieve job postings from stackoverflow.com**

Ø **Usability:** The student will see a button to import jobs from the outside.  
  
Ø  **Reliability:** The system should perform correctly 99% of the time.  
  
Ø  **Performance:** The system should respond within 1 second.  
  
Ø  **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification change.

**Allow system to retrieve job postings from monster.com**

Ø **Usability:** The student will see a button to import jobs from the outside.  
  
Ø  **Reliability:** The system should perform correctly 99% of the time.  
  
Ø  **Performance:** The system should respond within 1 second.  
  
Ø  **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow student guest user to access the system without login in.**

Ø **Usability:** The student user will see a button to login into the system as an student guest.  
  
Ø **Reliability:** The system should perform correctly 99% of the time.  
  
Ø  **Performance:** The system should respond within 1 second.  
  
Ø  **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow student guest user to search for job postings without login in.**

Ø **Usability:** The student user will see the Job page to search for job postings.  
  
Ø **Reliability:** The system should perform correctly 99% of the time.  
  
Ø  **Performance:** The system should respond within 1 second.  
  
Ø  **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow employer guest user to access the system without login in.**

Ø **Usability:** The employer user will see a button to login into the system as an employer guest.  
  
Ø **Reliability:** The system should perform correctly 99% of the time.  
  
Ø  **Performance:** The system should respond within 1 second.  
  
Ø  **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow employer guest user to post a job without login in.**

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

**Allow employer guest user to search for student based on skills without login in.**

Ø **Usability:** The employer user will see a search bar to seek students based on skills.  
  
Ø **Reliability:** The system should perform correctly 99% of the time.  
  
Ø  **Performance:** The system should respond within 1 second.  
  
Ø  **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow student user to replace his current PDF resume.**

Ø **Usability:** The employer user will see an upload button to upload an updated PDF resume to his/her profile.  
  
Ø **Reliability:** The system should perform correctly 99% of the time.  
  
Ø  **Performance:** The system should respond within 3 second.  
  
Ø  **Supportability:** The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**Allow student user to replace his current video resume.**

Ø **Usability:** The employer user will see an upload button to upload an updated video resume to his/her profile.  
  
Ø **Reliability:** The system should perform correctly 99% of the time.  
  
Ø  **Performance:** The system should respond within 60 second. (Depending on user connection bandwidth)  
  
Ø  **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

REFER TO APPENDIX D

In our sequence diagrams, actors can be 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 | Student, Employer |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | User clicks My Profile          User is redirected to his profile page          User clicks on the edit picture button          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 | User is on Virtual Job Fair                   User is logged in |
| Steps | User clicks My Profile                   User is redirected to his profile page                   User clicks on the edit resume button                   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 | User is on Virtual Job Fair                   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: 3. For the employer:    1. the profile of the student participating in the interview. 4. For the student:    1. the profile of the employer conducting the interview. 5. 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 6. Once both parties are connect the video interview is started 7. 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, Guest 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 | Student, Employer |
| 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 | Student, Employer |
| 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 | Student, Employer |
| 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 | Student, Employer |
| 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 | Student, Employer |
| 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 | Student, Employer |
| 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, Guest Employer |
| Pre-conditions | 1. Actor is logged in 2. Actor 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, guest Employer |
| Pre-conditions | 1. Actor is logged in 2. Actor 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 | Employer, 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 |

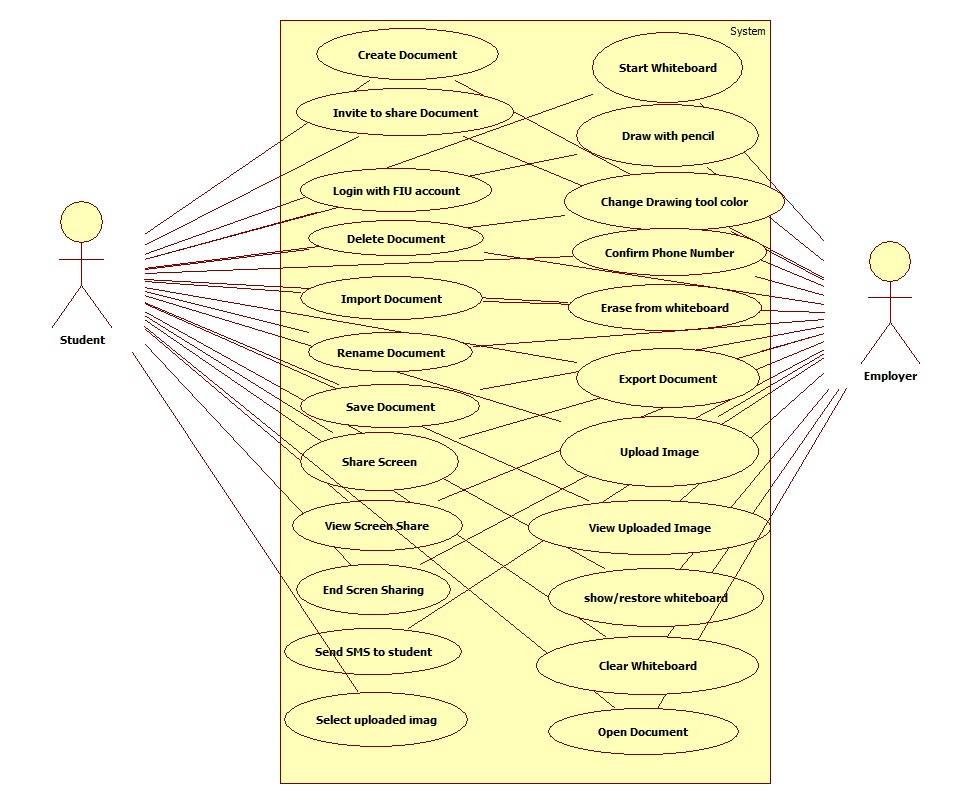
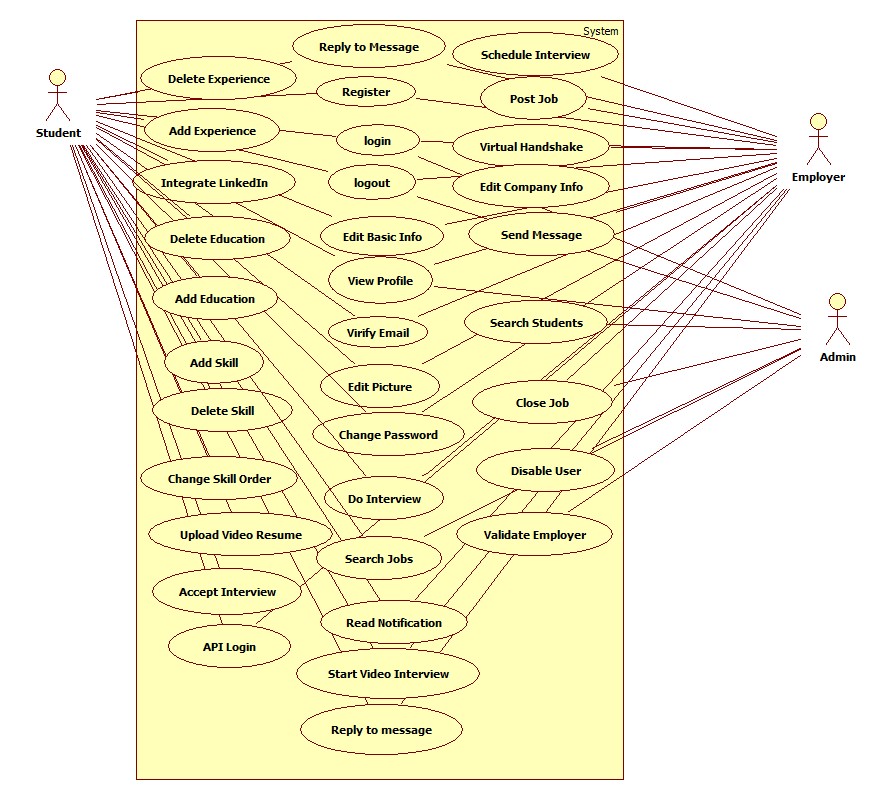
|  |  |
| --- | --- |
| Use Case ID | **VJF-062 Draw With Pencil** |
| **Description** | Allows actor to draw on the whiteboard using the pencil functionality |
| **Actor** | Employer or student |
| **Pre-conditions** | 1) A interview session is active  2) A whiteboard session has been started |
| **Steps** | 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 |

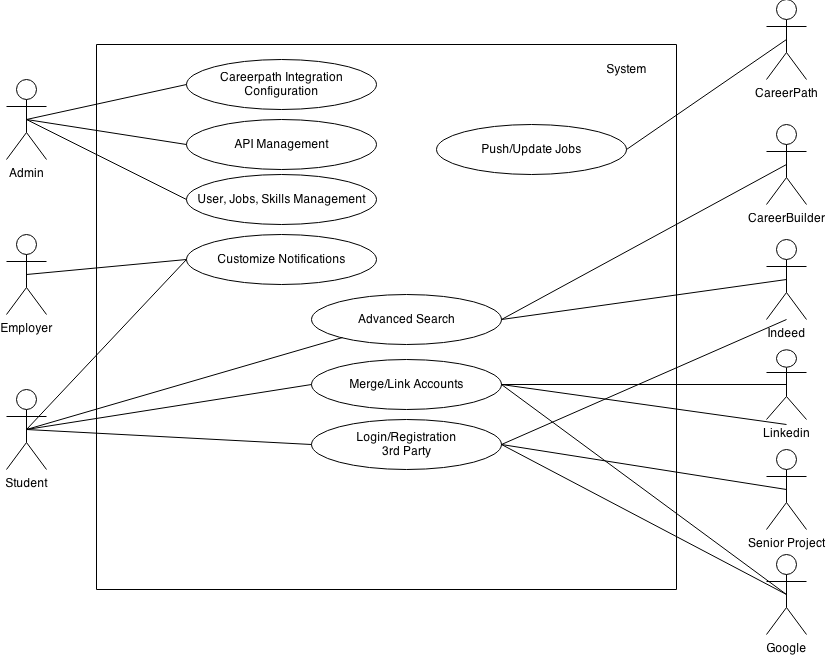
|  |  |
| --- | --- |
| **Use Case ID** | **VJF-064 Type Text Into Whiteboard** |
| **Description** | Allows a user to type text into the whiteboard |
| **Actor** | Employer or student |
| **Pre-conditions** | 1) A interview session is active  2) A whiteboard session has been started |
| **Steps** | 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 |
| Use Case ID | **VJF-067 Advanced Employer Search** |
| Description | Employer searches for students by skill, location, school, major, graduation date, experience, and previous positions. |
| Actor | Employer, Guest Employer |
| Pre-conditions | 1. Actor is logged in 2. Actor is on advance student search tab |
| Steps | 1. Employer search for in skill: “+java +programming” 2. Employer enters school: “Florida International University.” 3. Employer leaves all other fields empty. 4. Employer may select zip code button. 5. Employer clicks search |
| Post-conditions | Employer gets a view to the right of the search result page with relevant students. |
| Exceptions | None |
| Use Case ID | **VJF-068 Save Queries** |
| Description | Employer will be able to save query depending on skill, location, zip code, school, major, graduation date, experience, and previous positions. |
| Actor | Employer |
| Pre-conditions | 1.   Employer is logged in   1. Employer is on advance employer advance search. |
| Steps | 1. Employer clicks on the save query button. 2. Employer may type one or multiple skills. 3. Employer may type one location. 4. Employer may type one zip code. 5. Employer may type one school. 6. Employer may type one major. 7. Employer may type one graduation date. 8. Employer may type experience. 9. Employer may type one position. 10. Employer clicks save query 11. Window of query name comes out 12. Employer must type query name 13. Employer clicks save. |
| Post-conditions | Employer gets redirected to advance search page and a confirmation or error message gets displayed. |
| Exceptions | None |
| Use Case ID | **VJF-069 Dynamic Query Notification** |
| Description | Employer can select which query he wants to be notified about and every how much time he wants to be notified about. |
| Actor | Employer |
| Pre-conditions | 1. Employer is logged in 2. Employer is on profile page. |
| Steps | 1. Employer clicks on the student notification button (ON). 2. Employer selects one or more queries he wants to be notified about. 3. Employer types the number of days that will make the interval that he will be notified about. 4. Employer clicks the save button. |
| Post-conditions | Employer gets a view of the profile view with the updated information. |
| Exceptions | None |
| Use Case ID | **VJF-070 Search Query** |
| Description | Employer can select which query he wants to search on the database. |
| Actor | Employer |
| Pre-conditions | 1. Actor is logged in 2. Actor is on student advanced search. |
| Steps | 1. Employer clicks on the query he wants to run. 2. Employer clicks on the search button. |
| Post-conditions | Employer get a view with the results of the search. |
| Exceptions | None |
| Use Case ID | **VJF-071 Admin Add Student** |
| Description | The administrator is able to add one user to the database. |
| Actor | Admin |
| Pre-conditions | 1. Administrator is logged in.  2. Administrator is on Home Page |
| Steps | 1. Administrator clicks on the Add Student Tab in the Administrator Dashboard 2. Administrator must add username 3. Administrator must add password and retype password fields 4. Administrator must add student first and last time 5. Administrator clicks create button. |
| Post-conditions | Administrator will get a confirmation model displayed at the right of the dashboard. |
| Exceptions | None |
| Use Case ID | **VJF-072 Admin Add Employer** |
| Description | The administrator is able to add one user to the database. |
| Actor | Admin |
| Pre-conditions | 1. Administrator is logged in.  2. Administrator is on Home Page |
| Steps | 1. Administrator clicks on the Add Employer Tab in the Administrator Dashboard 2. Administrator must add username 3. Administrator must add password and retype password fields 4. Administrator must add employer first and last time 5. Administrator must add hide email from student option. 6. Administrator must write in about me. 7. Administrator must add company name 8. Administrator must add Location of company 9. Administrator must add city of company. 10. Administrator must add the state the company resides in. 11. Administrator must add zip code of the company. 12. Administrator must add company description. 13. Administrator clicks create button. |
| Post-conditions | Administrator will get a confirmation model displayed at the right of the dashboard. |
| Exceptions | None |
| Use Case ID | **VJF-073 Admin Statistics** |
| Description | The administrator is able to view information about the website and database. |
| Actor | Administrator |
| Pre-conditions | 1. Administrator is logged in.  2. Administrator is on Home Page |
| Steps | 1. Administrator clicks on the website statistics tab in the Administrator Dashboard. |
| Post-conditions | Administrator will get a statistics model displayed at the right of the dashboard. |
| Exceptions | None |
| Use Case ID | **VJF-074 Student Search** |
| Description | The employer is able to search for a student on one skill. |
| Actor | Employer, Guest Employer |
| Pre-conditions | 1. Actor is logged in. |
| Steps | 1. Employer types one skill the search box. 2. Employer can select a skill from auto complete or fully type out the skill 3. 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-075 Job Search** |
| Description | The student is able to search for a job on one skill. |
| Actor | Student, Guest Student |
| Pre-conditions | 1. Actor is logged in. |
| Steps | 1. Student types one skill the search box. 2. Student can select a skill from auto complete or fully type out the skill 3. Student clicks submit. |
| Post-conditions | Student is redirected to search result page with relevant jobs, and option to search more jobs. |
| Exceptions | None |
| **Use Case ID** | **VJF-076 View PDF Resume** |
| Description | Allows student actor to view his/her PDF Resume. |
| User Story #392 | As a Student I would like to see my PDF resume from my profile page and be able to upload a more updated one. This will benefit me as I will be able to reassure that I have uploaded a file with my resume, and it will allow me to have the most recent copy by replacing the old one. |
| Actor | Student |
| Pre-conditions | 1. Actor is logged in. 2. Actor is in his/her profile page. 3. Actor has a PDF resume uploaded in the system. |
| Steps | 1. Use case begins when the actor clicks on the view resume button. 2. System displays actor resume. |
| Post-conditions | Actor see his/her resume file |
| Alternative Steps | N/A |
| Exceptions | No resume on profile |
| **Use Case ID** | **VJF-077 Replace PDF Resume** |
| Description | Allows student actor to replace his/her PDF resume. |
| User Story #392 | As a Student I would like to see my PDF resume from my profile page and be able to upload a more updated one. This will benefit me as I will be able to reassure that I have uploaded a file with my resume, and it will allow me to have the most recent copy by replacing the old one. |
| Actor | Student |
| Pre-conditions | 1. Actor is logged in. 2. Actor is in his/her profile page. 3. Actor has a PDF resume to upload to the system. |
| Steps | 1. Use case begins when the actor clicks on the update resume button. 2. System displays a windows to select the file to be uploaded. 3. Actor selects his/her resume file and click open. 4. System stores the pdf resume and updates its path on the database. 5. Systems displays the upload was successful. |
| Post-conditions | Actor update his/her resume file. |
| Alternative Steps | 1. From step 4 the file cannot be uploaded or the path inserted in the database. 2. System displays the upload had an error. |
| Exceptions | 1. Invalid file format 2. Invalid file extension 3. No file selected |

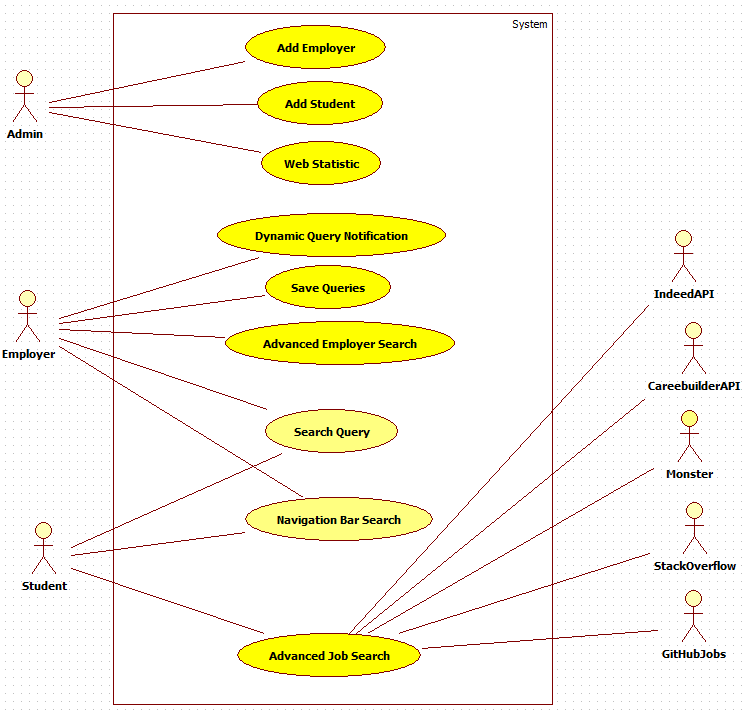
|  |  |
| --- | --- |
| **Use Case ID** | **VJF-078 Replace Video Resume** |
| **Description** | Allows student actor to replace his/her video resume. |
| **User Story #392** | As a Student I would like to see my video resume from my profile page and be able to upload a more updated one. This will benefit me as I will be able to reassure that I have uploaded a file with my resume, and it will allow me to have the most recent copy by replacing the old one. |
| **Actor** | Student |
| **Pre-conditions** | 1. Actor is logged in. 2. Actor is in his/her profile page. 3. Actor has a video resume to upload to the system. |
| **Steps** | 1. Use case begins when the actor clicks on the update resume button. 2. System displays a windows to select the file to be uploaded. 3. Actor selects his/her resume file and click open. 4. System stores the pdf resume and updates its path on the database. 5. Systems displays the upload was successful. |
| **Post-conditions** | Actor update his/her video resume file. |
| **Alternative Steps** | 1. From step 4 the file cannot be uploaded or the path inserted in the database. 2. System displays the upload had an error. |
| **Exceptions** | 1. Invalid file format 2. Invalid file extension 3. No file selected |
| **Use Case ID** | **VJF-079View Video Resume** |
| **Description** | Allows student actor to view his/her video Resume. |
| **User Story #392** | As a Student I would like to see my video resume from my profile page and be able to upload a more updated one. This will benefit me as I will be able to reassure that I have uploaded a file with my resume, and it will allow me to have the most recent copy by replacing the old one. |
| **Actor** | Student |
| **Pre-conditions** | 1. Actor is logged in. 2. Actor is in his/her profile page. 3. Actor has a PDF resume uploaded in the system. |
| **Steps** | 1. Use case begins when the actor clicks on the view resume button. 2. System displays actor video resume. |
| **Post-conditions** | Actor see his/her resume video |
| **Alternative Steps** | N/A |
| **Exceptions** | No resume on profile |

## 6.2-Appendix B – Use Cases Diagram using UML

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

**Summer 2014 System’s Use Case Diagram**

**New System’s Use Case Diagram**



**Spring 2015 System’s Use Case Diagram**

## 

## 

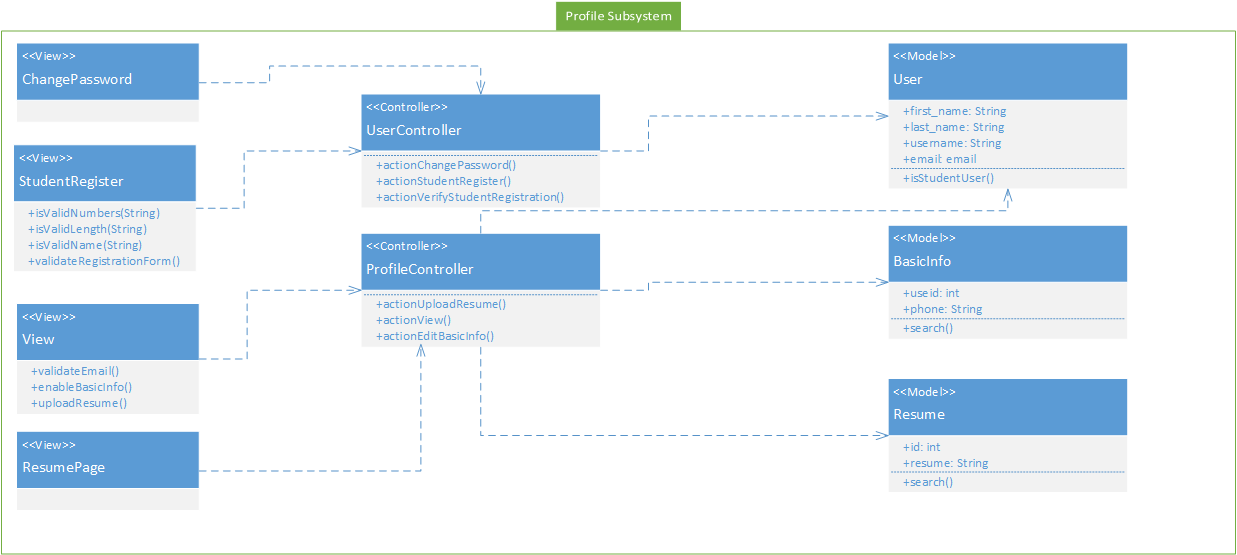
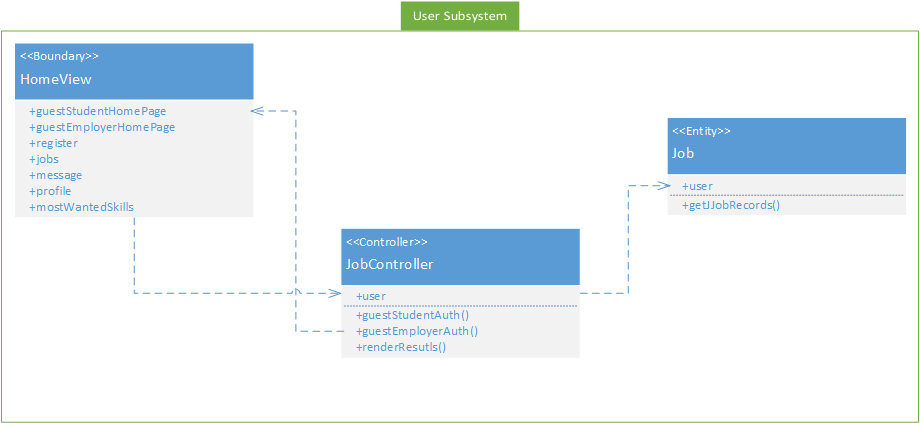
## 6.3-Appendix C - Static UML diagram

## ::AdvStudentSearch.PNG

## :::Downloads:object2-2.PNG:::Downloads:object3.PNG

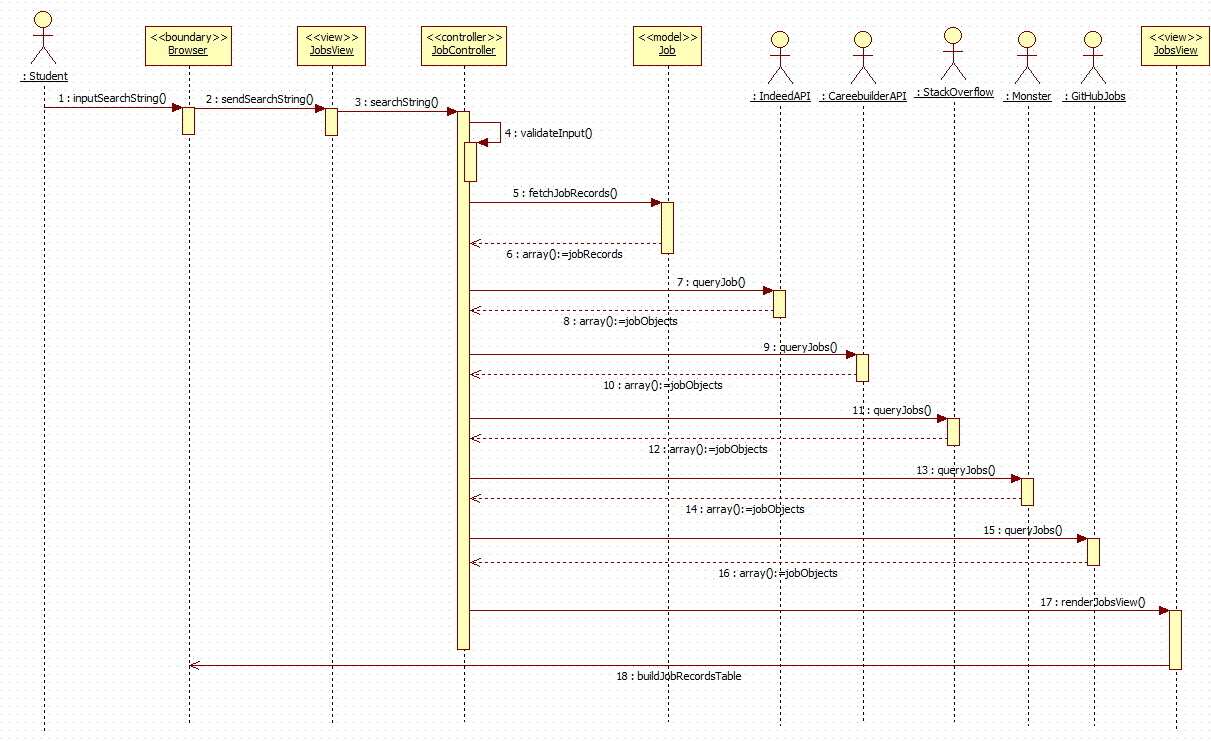
## :::Downloads:object4.PNG

**Spring 2015 User Subsystem Class Diagram**

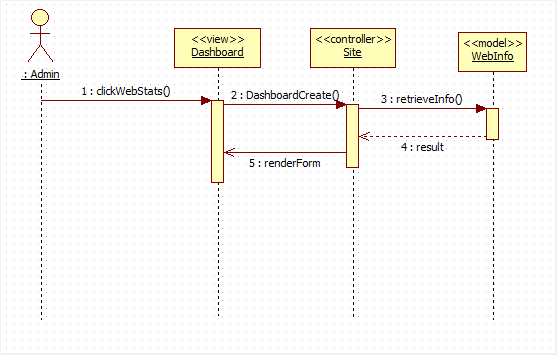


## 6.4-Appendix D – Dynamic UML diagrams

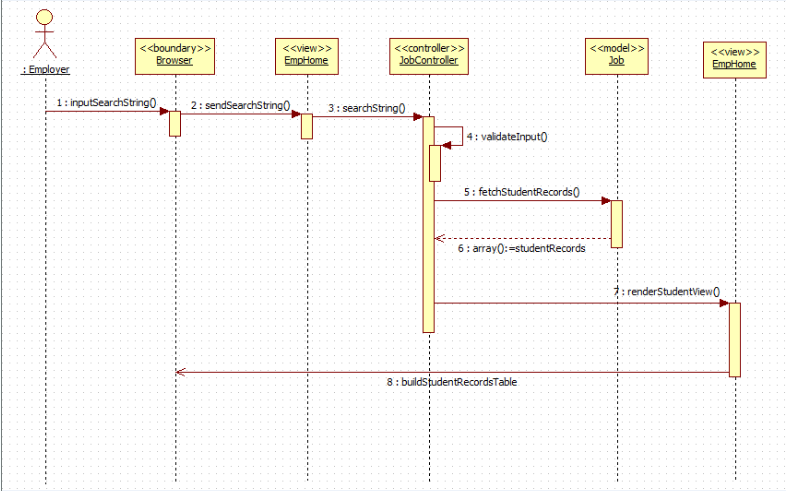
**Advance Job Search**



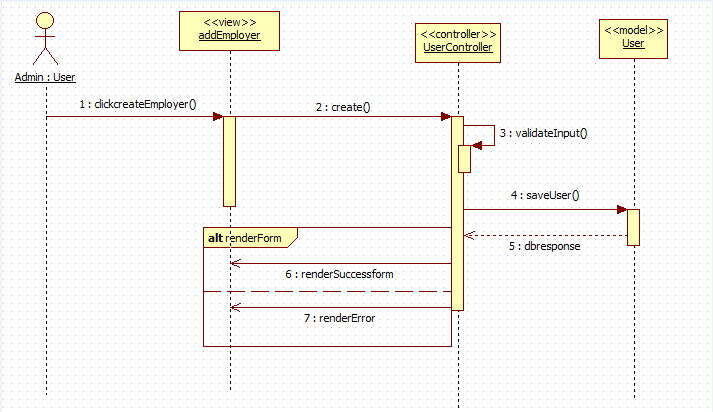
**Admin Statistics**



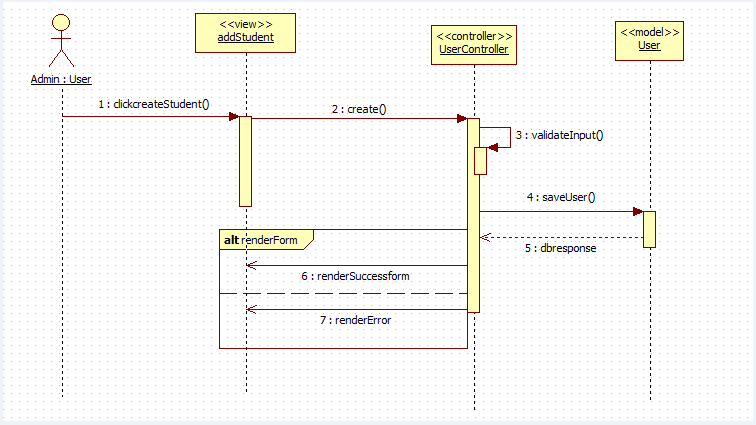
**Advanced Employer Search**



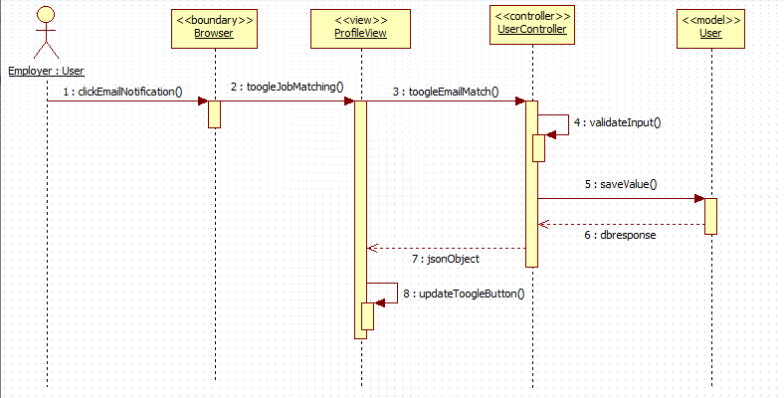
**Add Employer**



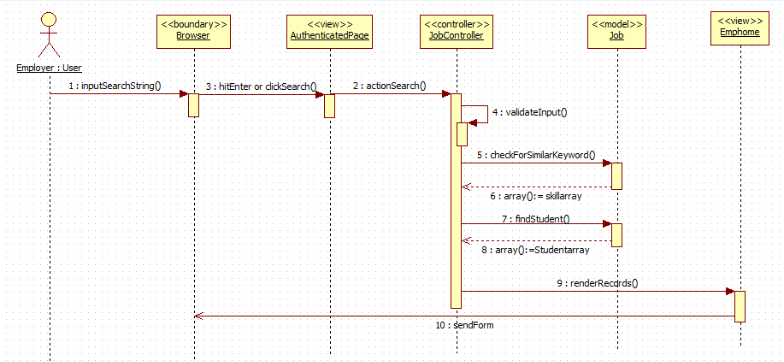
**Add Student**



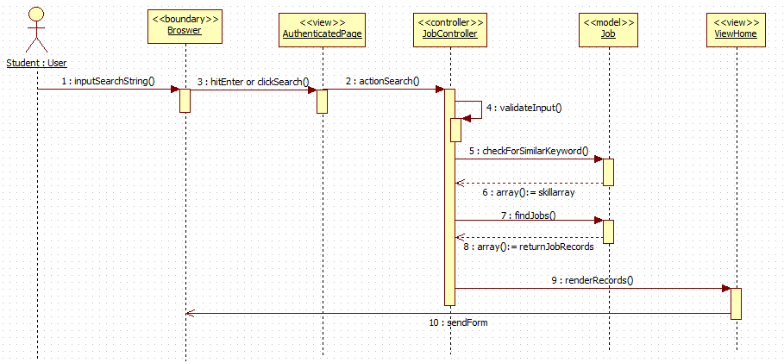
**Employer Notification**



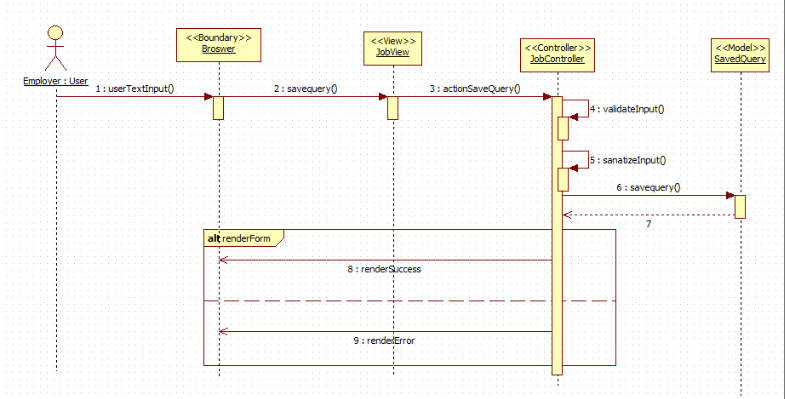
**Employer Navigation Search Bar**



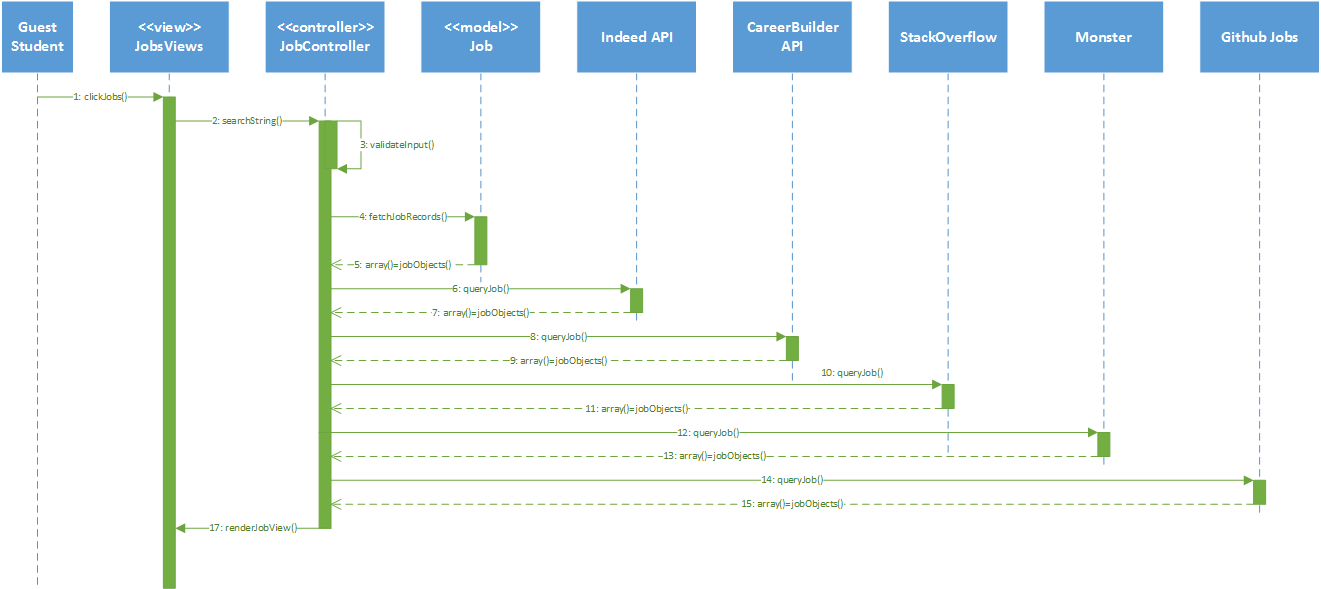
**Student Navigation Bar Search**



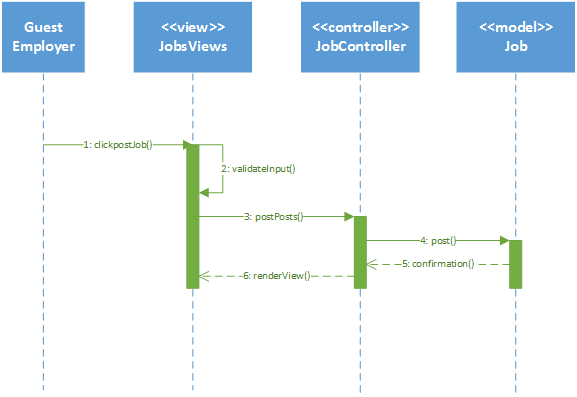
**Save Query**



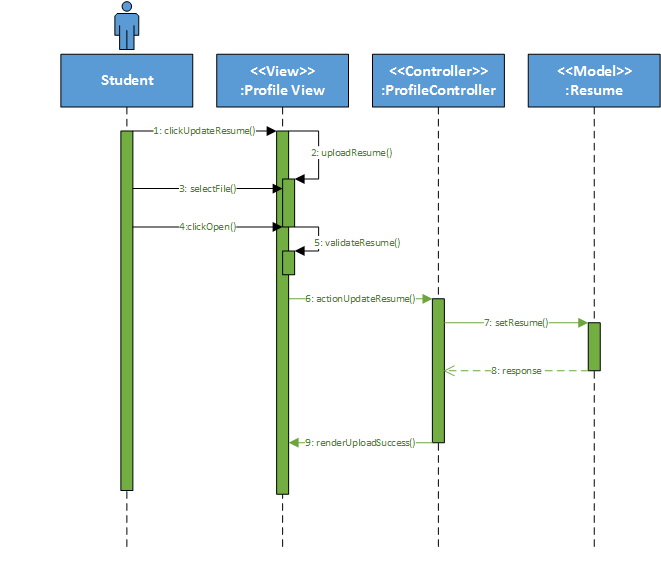
## Spring 2015 Guest Student User Search Job Sequence Diagram



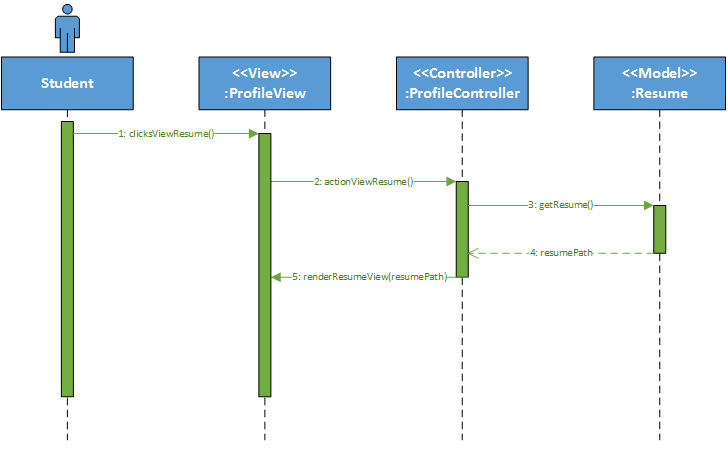
## Spring 2015 Guest Employer User Job Post Sequence Diagram



## Student Profile Upload Resume Sequence Diagram

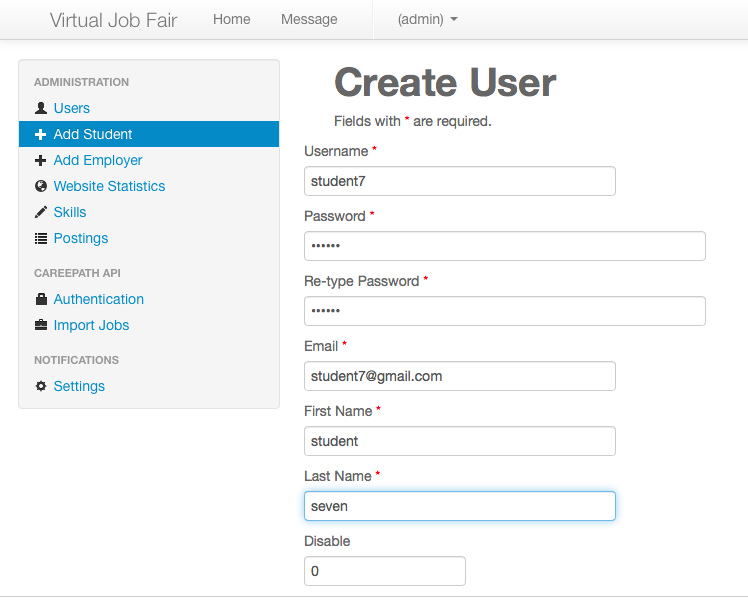


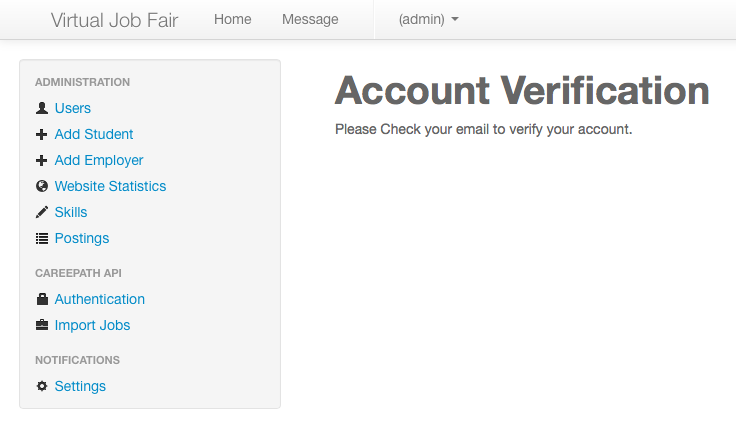
## Student Profile Views Resume Sequence Diagram

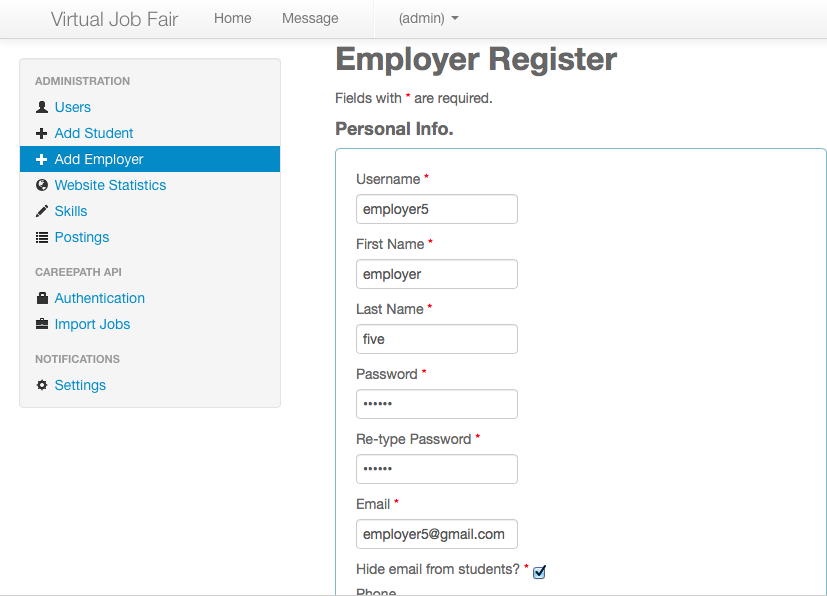


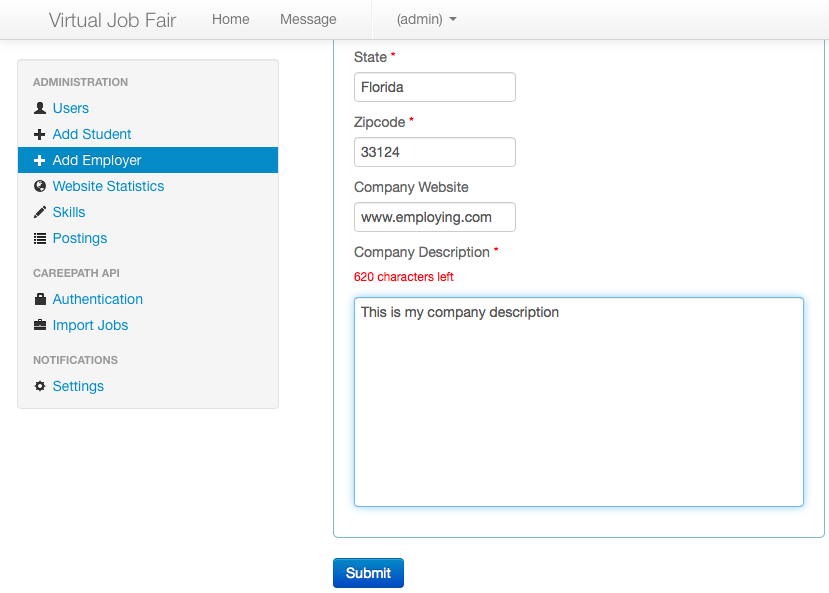
## 

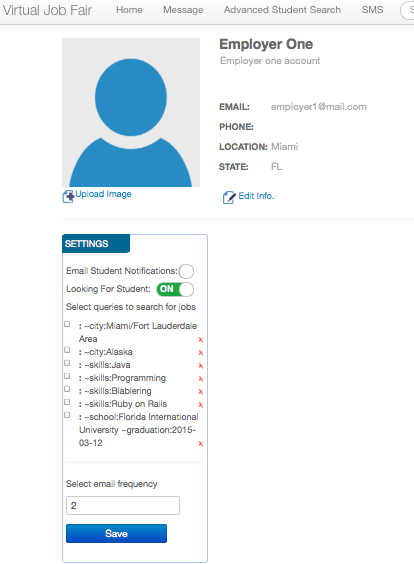
## 6.5-Appendix E– User Interface designs.

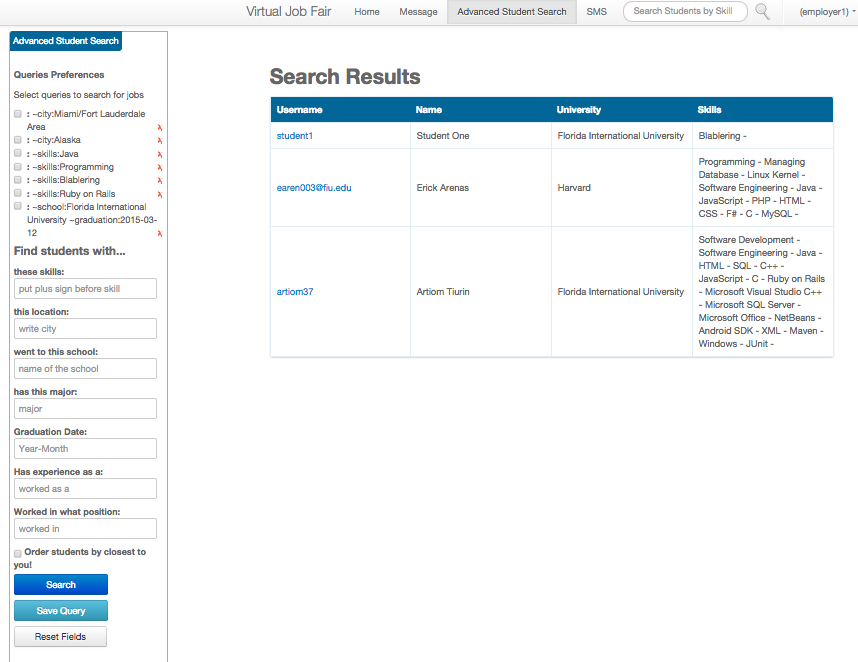


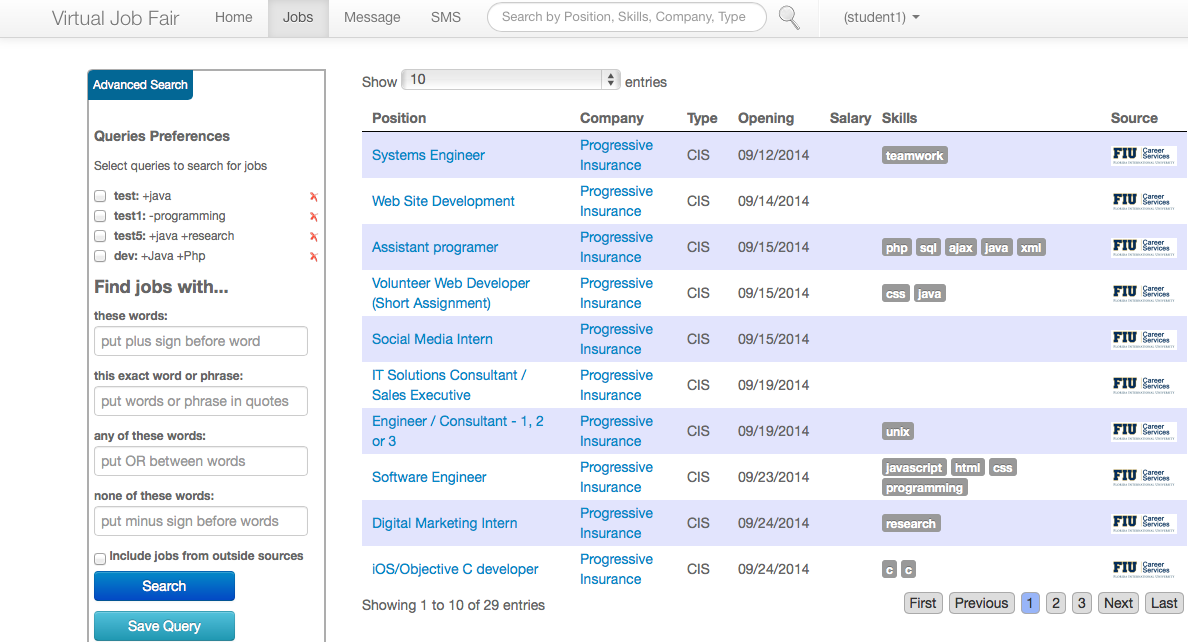












## 

## 

## 

## 

## 

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

Meeting 1:

Date: 1/23/2015

Start Time: 1:00 PM

End Time: 1:55 PM

On this day I met with Instructor Masoud Sadjadi for 50 minutes from 1:00 to 1:50 and gave me guidelines about GitHub. Allowed me access the documentation and discussed the goal of the initial sprint. We tried to deploy the latest version from development to production unsuccessfully. We agreed on a new meeting on 1/26/2015 @ 9:00 AM.

Meeting 2:

Date: 1/26/2015

Start Time: 9:30 AM

End Time: 10:30 AM

On this day I met with Instructor Masoud Sadjadi and we exported Dev Database into his environment. We set up the development branch in GitHub, his local repository and tagged existing master as v4.0. We created a copy of the main.php file located under JobFair/protected/config/ and we ignore the main.php as it contains the username and password for dev and prod environments. We found a bug while deploying the project in the webserver root folder such that if the name of the folder is not set to “JobFair” the project will not function as this is hardcoded into SiteController.php under JobFair/protected/controllers/. As a result we created two new user stories as follows:

1. Making base directory configurable:
   1. As the admin of virtual job fair, I would like to be able to configure the base url. As it is now, the base url of /JobFair/ is hardcoded in many places in the code. I would like to be able to set the base directory in /protected/controllers/SiteController.php and by changing that, I should be able to install the virtual job fair under any directory that I like.
2. Making the base url configurable:
   1. As the admin of virtual job fair, I would like to be able to install this project on different places (e.g., vjf.cis.fiu.edu, vjf-dev.cis.fiu.edu, or localhost). There should not be anywhere in the code referring directly to the actual base url; instead they should refer to a variable defined in /protected/controllers/SiteController.php that specifies the base url.

Meeting 3:

Date: 1/30/2015

Start Time: 5:15 PM

End Time: 6:30 PM

Met with instructor Sadjadi and mentor Juan Caraballo and went over the user stories collected. We decided that the most important stories are the ones that improves the student and employer user directly.

Meeting 4:

Date: 2/2/2015

Start Time: 9:40 PM

End Time: 10:00 PM

Met with product owner Juan Caraballo to discuss new stories ideas (One click application process for students, administrator reporting service and on the fly resume creation). He instructed me of the policies regarding applying jobs fetched from the different API’s used by VJF. He encouraged me to follow the one click application process for jobs that are directly posted on the VJF system.

Meeting 5:

Date: 2/9/2015

Start Time: 10:00 PM

End Time: 10:20 PM

Met with product owner Juan Caraballo regarding the user stories chosen for the project. In particular he is interested in the Guest Employer and the Guest Student User. We agreed to have another meeting on Thursday, February 12, 2015 at 4:00 PM.

Meeting 6:

Date: 2/13/2015

Start Time: 5:00 PM

End Time: 5:30 PM

Met with product owner Masoud Sadjadi to present the users stories developed in Sprint 1. We agreed for Sprint 2 we should test all functionalities for the system and document the ones not performing as per the Product Owners expectations.

Meeting 7:

Date: 2/23/2015

Start Time: 6:00 PM

End Time: 6:30 PM

Met with product owner Masoud Sadjadi to check on the progress made in sprint 2.

Meeting 8:

Date: 2/27/2015

Start Time: 6:30 PM

End Time: 7:00 PM

Met with product owner Masoud Sadjadi to present the users stories developed in Sprint 2. We agreed for Sprint 3 to focus on several bug fixing to provide more stability to the system. Also, he accepted to extend the upload resume functionality to include update PDF resume, video resume and profile picture.

Meeting 9:

Date: 3/23/2015

Start Time: 9:00 AM

End Time: 10:00 AM

Met with product owner Masoud Sadjadi to present the users stories developed in Sprint 3. We agreed for Sprint 4 to focus on more bug fixing to provide more stability to the system. Also, to do a feasibility study about search engines. All with the vision to incorporate one to the VJF with a unique search bar similar to Google.

Meeting 10:

Date: 3/30/2015

Start Time: 4:00 PM

End Time: 5:00 PM

Met with product owner Masoud Sadjadi to check on the progress made in sprint 4.

Meeting 11:

Date: 4/3/2015

Start Time: 6:00 PM

End Time: 6:30 PM

Met with product owner Masoud Sadjadi to present the users stories developed in Sprint 4. We agreed for Sprint 5 to focus on fixing Indeed API and search result coming from Indeed. Also, to incorporate Solr search engine into VJF.

Meeting 12:

Date: 4/17/2015

Start Time: 7:00 PM

End Time: 7:30 PM

Met with product owner Masoud Sadjadi to present the users stories developed in Sprint 5. He advised the approach needed to make successful videos for the final presentation.

**7. References**

1. Feasibility Study & Project Plan Document v2.0. Spring 2013.
2. Feasibility Study & Project Plan Document v3.0. Summer 2014.
3. Feasibility Study & Project Plan Document v4.0. Fall 2014.
4. Requirement Document v2.0. Spring 2013.
5. Requirement Document v3.0. Summer 2014.
6. Requirement Document v4.0. Fall 2014.
7. Design Document v2.0. Spring 2013.
8. Design Document v3.0. Summer 2014.
9. Design Document v4.0. Fall 2014.
10. Final Deliverable v2.0. Spring 2013.
11. Final Deliverable v3.0. Summer 2014.
12. Final Deliverable v4.0. Fall 2014.
13. "Bootstrap." *2.2.2 Documentation*. N.p., n.d. Web. 24 July 2014. <http://bootstrapdocs.com/v2.2.2/docs/>.
14. "Documentation." *Yii PHP Framework: Best for Web 2.0 Development*. N.p., n.d. Web. 24 July 2014. <http://www.yiiframework.com/doc/>
15. Chacon, Scott. *Pro Git*. Berkeley, CA: Apress, 2009. Online. <http://git-scm.com/book/en/Getting-Started>