Feasibility Study and Project Plan

CIS 4911 – Senior Project U01

Virtual Job Fair 2.0

**Members:**

Luis Irizarry

Jorge Fernandez

Luis Benjumea

**Instructor:**

Masoud Sadjadi

**Mentor:**

Juan Caraballo

**Date:**

10/7/2013

Copyright © Florida International University 2013. All Rights Reserved. No part of the Virtual Job Fair Project or documentation may be reproduced or modified without the express consent of Florida International University. Redistribution or commercial use is strictly prohibited.

**ABSTRACT**

The Design Document gives an insight into the system structure of the Virtual Job Fair 2.0 project. By describing system architecture, subsystem decomposition, design methodology, hardware and software mapping, persistent data management and security/privacy gives the reader a better idea of the process that was used to design Virtual Job Fair 2.0. Chapter 1 introduces basic information about the project, such as problem definition, definitions, acronyms and overview of document. Chapter 2 introduces the structure of the system by describing its system design and subsystem decomposition.

Chapter 3 delves into to the detail design, including static models and dynamics models plus code specification with class interfaces. Then, a glossary is introduced with domain-specific terms. Then, finally, the appendixes contain miscellaneous information, such as use case diagrams, use cases being implemented, document class interfaces and diary of meeting and tasks.

Contents

[1.      Introduction 5](#_Toc374337944)

[1.1.   Problem definition 5](#_Toc374337945)

[1.2.   Design methodology used 6](#_Toc374337946)

[1.3.   Definitions, acronyms, and abbreviations. 7](#_Toc374337947)

[1.4.   Overview of document 8](#_Toc374337948)

[2.      System Design (i.e., overall system design) 9](#_Toc374337949)

[2.1.   Overview 9](#_Toc374337950)

[2.2.   Subsystem Decomposition 10](#_Toc374337951)

[2.3.   Hardware and Software Mapping 14](#_Toc374337952)

[2.4.   Persistent Data Management– 15](#_Toc374337953)

[2.5.   Security/Privacy 35](#_Toc374337954)

[3.      Detailed Design 37](#_Toc374337955)

[3.1.   Overview 37](#_Toc374337956)

[3.2.   Static model 38](#_Toc374337957)

[3.3.   Dynamic model 42](#_Toc374337958)

[5.      Appendix 51](#_Toc374337959)

[5.2.   Appendix B 53](#_Toc374337960)

[5.3.   Appendix C 95](#_Toc374337961)

[5.4.   Appendix D - Diary of meeting and tasks. 96](#_Toc374337962)

[6. References 105](#_Toc374337963)

# 1.      Introduction

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

## 1.1.   Problem definition

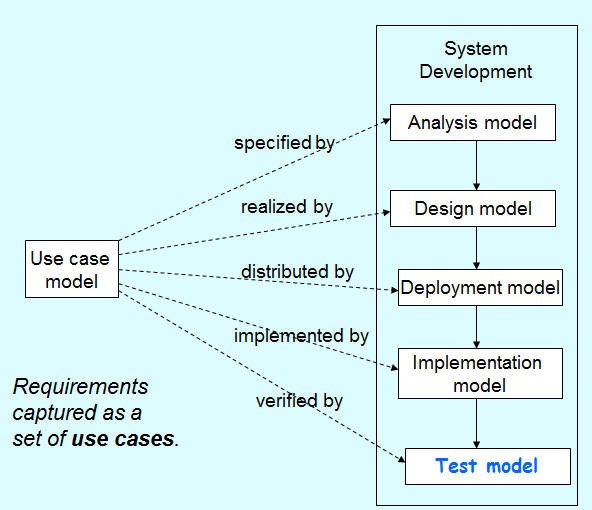
Employers looking for talent are always interested in filling out positions with the best possible people. In order to accomplish this task, the most effective method to date is to tap local talent, whether it is at universities or job fairs. Given the increasing globalization trend, and the fact that 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 intends to provide a more transparent interview process that allows employer and potential employee to interact as closely as possible and provide a better interview experience.

## 1.2.   Design methodology used

We will take advantage of the Unified Development Process in order to develop the new system and add the necessary functionality.  Since USDP iterative and use case driven in nature; we will focus our attention to adhering as much as possible to the functional requirements and gathered uses cases. It is very important to follow the rule imposed by this methodology in order to deliver a quality product that conforms and exceeds all standards.

We have found the iterative process to be very simple since it allows us to make small but concise progress towards the completion of the project. It is also important to understand the importance of the Unified Modeling Language (UML) and the significance of the models used to represent the design. We rely heavily in diagrams in order to visualize complicated concepts and devise solutions that are both efficient as well as robust.



## 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

**Acronyms**

-       **VJF**: Virtual Job Fair

-       **FIU:** Florida International University

-       **GPA:** Grade point average

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

-       **USDP:** Unified Software Development Document

**Abbreviations**

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

## 1.4.   Overview of document

In chapter 1, the main problem is introduced, along with the design methodology used for the project, definitions, acronyms and abbreviations. In chapter 2, the system is introduced in terms of system architecture, with its subsystem decomposition, hardware and software mapping, persistent data management and security/privacy aspects explained. In chapter 3, the behavior of each subsystem is described, and the static models and dynamic models used are explained.

In chapter 4, a glossary with domain-specific terms is introduced. In the appendix, miscellaneous material, such as use case diagrams, use cases being implemented and documented class interfaces can be found. Finally, a diary of meetings and references can be found at the end of the document.

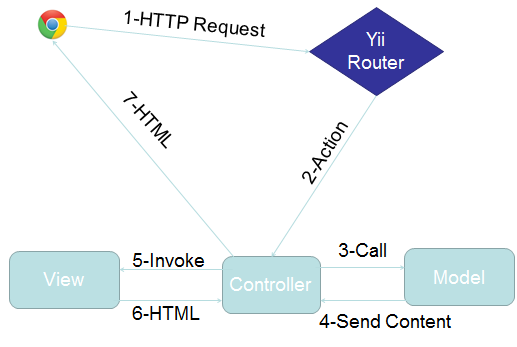
# 2.      System Design (i.e., overall system design)

This chapter gives an insight into the architectural pattern(s) used to build the system. Virtual Job Fair was subdivided into subsystems, each one with a specific functionality that adds richness to the interview process. In this chapter, an overview of the system design is introduced. Then, the decomposition of the system into subsystems is explained. Moreover, hardware and software mapping and persistent data management aspects of the project are discussed. Finally, the security and privacy issues of the system are introduced.

2.1.   Overview

The previous team implemented Virtual Job Fair using the Model-View-Controller architecture. Our team is adding functionality to the system, meaning that the Model-View-Controller architecture will be maintained, with models, views and controllers added to account for the new functionality. Below is their description of the architecture of the system:

 “The Architectural pattern used to built the system was the popular Model-View-Controller. This is the architecture implemented by the Yii Framework which was used by the team of developers. The following diagram depicts the architecture:”



2.2.   Subsystem Decomposition

The previous group had a total of 5 subsystems, while we will be adding an extra 4. Below are the subsystems that the previous group used.

**Video Interview Subsystem**

The Video Interview Subsystem allows users to have a virtual interview. This is possible thanks to Web-RTC technology. Web-RTC is an open source project that allows web browsers to communicate directly with each other with the aid of Java Scrip API calls and HTML5. This new technology makes the communication between internet users easier than traditional methods. Thanks to Web-RTC one can share video feed with other users without the need for media servers or plug-ins.

The synchronization of video interviews is handled in a table in the database. This is very important because we need to make sure only users scheduled to have an interview can be in the interview page. This is done by using a session key which is unique for each video interview scheduled. Users that arrive at the interview page would do so via a link which has many parameters such the session key. Once both users arrive at the interview page, the application will check for the session key parameter and match them, and only those users that have matching session keys would be able to connect to each other.

The uses cases related to this subsystem are:

·         VJF-0020 Start Video Interview

·         VJF-0021 Accept Interview

·         VJF-0030 Schedule Video Interview

**Messaging Subsystem**

The messaging subsystem is a very important one because it allows users to keep in contact and communicate with each other right on the system. It is very similar to a traditional inbox, only that it is internal to the system, similar to LinkedIn’s messaging. The messaging subsystem uses database tables to store and retrieve messages sent between users. It allow employers to message students and in doing so open up a line of communication with them; i.e., a student is able to message an employer only after the employer has initiated the communication with that particular student. Storing and retrieving messages efficiently is imperative to a successful messaging system. The messaging subsystem uses AJAX to rapidly access and store data; allowing users to interact faster with the system without having to wait for server calls.

The use cases related to this subsystem are:

·         VJF-0022 Reply to Message

·         VJF-0023 Send Message

**Notification Subsystem**

The notification subsystem is a very important part of the system. This is what allows users to stay up to date with the latest system interactions. The notification subsystem alerts users of any interaction by other users that might implicate them, such as a video interview been schedule for a user, or a new job post that matches a user’s skills.

The notification subsystem relies on the database structure to be able to efficiently keep the users inform. Due to the relationships between tables in the database the notification subsystem can easily detect what notification belongs to what user. Also it is important to sort each notification by category and level of importance. The notification subsystem achieves this by storing different types of notifications in the database and mapping them to their respective categories.

The uses cases related to this subsystem are:

·         VJF-0021 Accept Interview

·         VJF-0025 Post Job

·         VJF-001 Registration

·         VJF-0033 Apply to Job

·         VJF-0034 Read notification

**Profile creation subsystem**

Being able to create a good profile fast and efficiently is very important. The profile creation subsystem takes care of this by allowing students to import profile information from third party websites such as LinkedIn. This ensures integrity of the data in students’ profiles, and makes it very easy for students to create their profiles.

This is possible by using API calls to LinkedIn and retrieving the data from LinkedIn users. As it is to expect, the user must grant permission to do this by providing his/her login credentials which are handled by the LinkedIn API.

The use cases related to this subsystem are:

·         VJF-001 Registration

·         VJF-0019 Integrate LinkedIn

·         VJF-003 Edit Basic Info

·         VJF-004 Verify Email

·         VJF-008 Edit Picture

·         VJF-009 Upload Resume

·         VJF-0011 Add Education

·         VJF-0012 Delete Education

·         VJF-0013 Add Experience

·         VJF-0014 Delete Experience

**Student job match subsystem**

Making the right connection is what this web application is all about. Therefore, an efficient algorithm to match students to job openings is very important. The student job match subsystem takes care of matching students with the required skills to job post, making the job of the recruiters easier, as it shrinks the search to only the most qualified individuals for the job.

The student job match subsystem relies on the relationships between the data in the job table. By matching job skills to students skills listed on their profile the algorithm can effectively narrow down the search to only those individuals who possess those skills.

The uses cases related to this subsystem are:

·         VJF-0016 Add Skill

·         VJF-0017 Delete skill

·         VJF-0018 Change skills Order

·         VJF-0026 Virtual Handshake

Below, are the 4 additional subsystems that we will be using to decompose the system further.

**Collaborative Editor Subsystem**

The collaborative text editor subsystem allows users to edit a document in realtime with one another. The Zoho Docs API will provide the Features, Methods, Calls, and Functionality needed to implement the Collaborative Text Editor Subsystem. Zoho Docs is implemented as a web based application that requires no additional plugins or addons to be installed by the end users. The Zoho Docs API provides a clean, refreshing and intuitive user interface which facilitates usage of the system by end users. Zoho Documents is built upon AJAX technology, this provides the users with the fast, real-time, user experience required of the Subsystem.

The use cases related to this system are:

VJF-048 Create a new Document

VJF-049 Share active Document

VJF-050 Delete shared Document

VJF-051 Import Document

VJF-052 Rename Document

VJF-053 Save Document

VJF-054 Open Document

VJF-055 Export Document

**Screen Share subsystem**

Additionally the video screen Share subsystem allows users to share their screens with one another. This functionality is facilitated by the Screen Leap API. Screen Leap is a service that provides screen sharing functionality without the need for any of the users to install any specific software all that is required to function is the latest version of Java for the broadcaster.  Screen Leap is able to function perfectly with the system by providing different API calls that allow for easy sharing of information. By eliminating the need for the user to download extra software we are able to facilitate and simplify the usage of the Virtual Job Fair website.

VJF-042 Share Screen

VJF-043 View Screen Share

VJF-044 End Screen Sharing

**SMS message subsystem**

One of the most important features of the website is the ability to send messages to students as well as allow the users to receiver reminders about upcoming interviews directly in their SMS enabled phones. The SMS messaging subsystem is designed to allow this kind of functionality which can keep the users connected and active with the system.

All of this is possible by using the Twilio cloud communications service. Twilio is a company that provides services such as SMS sending and receiving, speech and text recognition, conference calling etc.  The most compelling feature is that an extensive and well documented API is provided to developers, this will be necessary in order to create a system that allows for sending text messages to users, as well as validating their identities.

VJF-045 Send SMS to student

VJF-046 Send interview reminder

VJF-047 Confirm phone number

**Whiteboard subsystem**

The whiteboard subsystem will allow users to draw on a whiteboard during a live interview. The whiteboard includes drawing, writing, erasing, clearing and picture-saving functionality, which makes it an ideal tool to share images during a live interview, to jot down notes or simply to brainstorm. The use cases related to the whiteboard are:

VJF-060 Show or Restore Whiteboard

VJF-062 Draw With Pencil

VJF-063 Change Drawing Tool Pencil

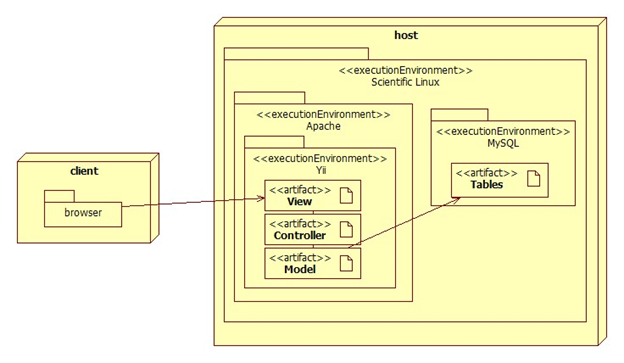
VJF-064 Type Text Into Whiteboard

VJF-065 Clear contents of whiteboard

VJF-066 Erase From Whiteboard

## 2.3.   Hardware and Software Mapping

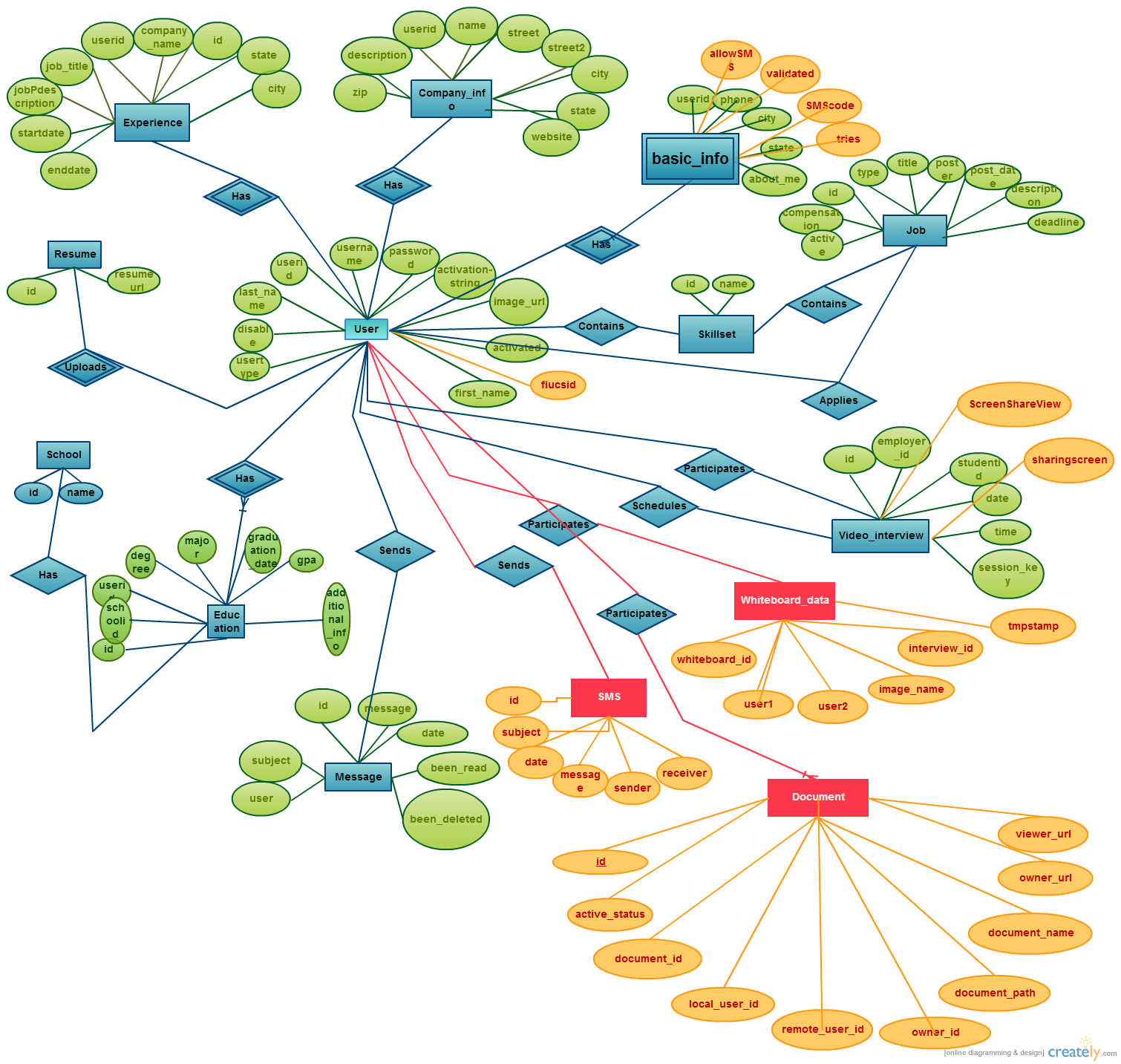
The previous team developed a deployment diagram which indicates the mapping of hardware and software. For our Virtual Job Fair 2.0, we are still using the Model-View-Controller architectural pattern. Moreover, our team requested a virtual machine running on Scientific Linux with an Apache server and the Yii framework on the FIU SCIS network, identical to the set up used by the previous team. For that reason, the same deployment diagram as before will be used. Therefore, the description below was that done by the previous team, and remains valid for Virtual Job Fair 2.0.



The deployment diagram shown above represents the hardware and software mapping in the Virtual Job Fair system.  The main components of the system are the Apache and MySQL environment hosted on a Linux operating system.  The Yii framework environment is using apache to execute, and contains our various artifacts used in development (Model, View, and Controller).  The models are mapped to tables in the MySQL environment set up on the same machine.  The browser on the client’s machine communicates with the server using HTTP.

2.4.   Persistent Data Management–

The previous group completed an ER diagram. Since we are adding new tables and columns into existing tables and database, respectively, below is the updated version of the ER diagram of the original group, which includes our new data, which is identified by a yellow background and dark red text.



Below is the data dictionary for the old group, to which we added several definitions, starting with “Collaborative Editor”:

**application**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **jobid** | int(11) | NO | PRI |  |  |
| **userid** | int(11) | NO | PRI |  |  |
| **application\_date** | varchar(45) | NO |  |  |  |
| **coverletter** | text | YES |  |  |  |

**basic\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **userid** | int(11) | NO | PRI |  |  |
| **phone** | varchar(15) | YES |  |  |  |
| **city** | varchar(45) | YES |  |  |  |
| **state** | varchar(45) | YES |  |  |  |
| **about\_me** | text | YES |  |  |  |
| **hide\_phone** | int(11) | YES |  |  |  |
| **allowSMS** | int(11) | 0 |  |  |  |
| **validated** | int(11) | 0 |  |  |  |
| **smsCode** | int(11) | null |  |  |  |
| **tries** | int(11) | 0 |  |  |  |

**company\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **FK\_userid** | int(11) | NO | PRI |  | auto\_increment |
| **name** | varchar(45) | YES |  |  |  |
| **street** | varchar(45) | YES |  |  |  |
| **street2** | varchar(45) | YES |  |  |  |
| **city** | varchar(45) | YES |  |  |  |
| **state** | varchar(45) | YES |  |  |  |
| **zipcode** | varchar(45) | YES |  |  |  |
| **website** | varchar(45) | YES |  |  |  |
| **description** | text | YES |  |  |  |

**education**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **degree** | varchar(45) | NO |  |  |  |
| **major** | varchar(45) | NO |  |  |  |
| **graduation\_date** | date | NO |  |  |  |
| **FK\_school\_id** | int(11) | YES | MUL |  |  |
| **FK\_user\_id** | int(11) | YES | MUL |  |  |
| **gpa** | float | YES |  |  |  |
| **additional\_info** | text | YES |  |  |  |

**experience**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **FK\_userid** | int(11) | YES | MUL |  |  |
| **company\_name** | varchar(45) | YES |  |  |  |
| **job\_title** | varchar(45) | YES |  |  |  |
| **job\_description** | text | YES |  |  |  |
| **startdate** | datetime | YES |  |  |  |
| **enddate** | datetime | YES |  |  |  |
| **city** | varchar(45) | YES |  |  |  |
| **state** | varchar(45) | YES |  |  |  |

**handshake**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **jobid** | int(11) | YES | MUL |  |  |
| **employerid** | int(11) | NO | MUL |  |  |
| **studentid** | int(11) | NO | MUL |  |  |

**job**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **type** | varchar(45) | NO |  |  |  |
| **title** | varchar(45) | NO |  |  |  |
| **FK\_poster** | int(11) | NO | MUL |  |  |
| **post\_date** | datetime | NO |  |  |  |
| **deadline** | datetime | YES |  |  |  |
| **description** | longtext | NO |  |  |  |
| **compensation** | varchar(45) | YES |  |  |  |
| **other\_requirements** | text | YES |  |  |  |
| **email\_notification** | int(11) | YES |  |  |  |
| **active** | int(11) | YES |  | 1 |  |
| **matches\_found** | int(11) | YES |  |  |  |

**job\_skill\_map**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **jobid** | int(11) | NO | MUL |  |  |
| **skillid** | int(11) | NO | MUL |  |  |
| **level** | varchar(45) | YES |  |  |  |
| **ordering** | int(11) | YES |  |  |  |

**message**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **FK\_receiver** | varchar(45) | NO | MUL |  |  |
| **FK\_sender** | varchar(45) | NO | MUL |  |  |
| **message** | text | YES |  |  |  |
| **date** | datetime | YES |  |  |  |
| **been\_read** | int(11) | YES |  | 0 |  |
| **been\_deleted** | int(11) | NO |  | 0 |  |
| **subject** | varchar(255) | YES |  |  |  |
| **userImage** | varchar(255) | YES |  |  |  |

**notification**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **sender\_id** | int(11) | NO | MUL |  |  |
| **receiver\_id** | int(11) | NO |  |  |  |
| **datetime** | time | NO |  |  |  |
| **been\_read** | int(11) | NO |  | 0 |  |
| **message** | varchar(5000) | YES |  |  |  |
| **link** | varchar(150) | YES |  |  |  |
| **importancy** | int(11) | NO |  | 0 |  |

**resume**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  |  |
| **resume** | varchar(255) | YES |  |  |  |

**school**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **name** | varchar(100) | NO |  |  |  |
| **email\_string** | varchar(45) | YES |  |  |  |

**skillset**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **name** | varchar(45) | NO | UNI |  |  |

**student\_skill\_map**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **userid** | int(11) | YES | MUL |  |  |
| **skillid** | int(11) | YES | MUL |  |  |
| **level** | varchar(45) | YES |  |  |  |
| **ordering** | int(11) | YES |  |  |  |

**user**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **username** | varchar(45) | NO | UNI |  |  |
| **password** | varchar(255) | YES |  |  |  |
| **FK\_usertype** | int(11) | NO | MUL |  |  |
| **email** | varchar(45) | NO | UNI |  |  |
| **registration\_date** | datetime | NO |  |  |  |
| **activation\_string** | varchar(45) | NO |  |  |  |
| **activated** | int(11) | YES |  |  |  |
| **image\_url** | varchar(255) | YES |  |  |  |
| **first\_name** | varchar(45) | NO |  |  |  |
| **last\_name** | varchar(45) | NO |  |  |  |
| **disable** | int(11) | YES |  |  |  |
| **has\_viewed\_profile** | int(11) | YES |  |  |  |
| **linkedinid** | varchar(45) | YES |  |  |  |
| **googleid** | varchar(45) | YES |  |  |  |
| **hide\_email** | int(11) | YES |  |  |  |
| **fiucsid** | varchar(45) | YES |  |  |  |

**usertype**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **type** | varchar(45) | NO |  |  |  |

**video\_interview**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **FK\_employer** | int(11) | NO | MUL |  |  |
| **FK\_student** | int(11) | NO | MUL |  |  |
| **date** | date | NO |  |  |  |
| **time** | time | NO |  |  |  |
| **session\_key** | varchar(45) | NO |  |  |  |
| **notification\_id** | varchar(45) | NO |  |  |  |
| **screenShareview** | varchar(90) | NO |  |  |  |
| **sharingscreen** | int(11) | NO |  |  |  |

**video\_resume**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  |  |
| **video\_path** | varchar(100) | YES | UNI |  |  |

**New System’s Persistent Data**

**Collaborative Editor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **active\_status** | tinyint(1) | NO |  |  |  |
| **document\_id** | varchar(256) | NO |  |  |  |
| **local\_user\_id** | int(11) | NO |  |  |  |
| **remote\_user\_id** | int(11) | NO |  |  |  |
| **owner\_id** | int(11) | NO |  |  |  |
| **document\_path** | varchar(255) | NO | UNI |  |  |
| **document\_name** | varchar(255) | NO | UNI |  |  |
| **owner\_url** | varchar(255) | NO |  |  |  |
| **viewer\_url** | varchar(255) | NO |  |  |  |

**basic\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **userid** | int(11) | NO | PRI |  |  |
| **phone** | varchar(15) | YES |  |  |  |
| **city** | varchar(45) | YES |  |  |  |
| **state** | varchar(45) | YES |  |  |  |
| **about\_me** | text | YES |  |  |  |
| **hide\_phone** | int(11) | YES |  |  |  |
| **allowSMS** | int(11) | YES |  |  |  |
| **SMScode** | int(11) | YES |  |  |  |
| **validated** | int(11) | YES |  |  |  |
| **Tries** | Int(11) | NO |  |  |  |

**video\_interview**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **FK\_employer** | int(11) | NO | MUL |  |  |
| **FK\_student** | int(11) | NO | MUL |  |  |
| **date** | date | NO |  |  |  |
| **time** | time | NO |  |  |  |
| **session\_key** | varchar(45) | NO |  |  |  |
| **notification\_id** | varchar(45) | NO |  |  |  |
| **ScreenShareView** | varchar(45) | YES |  |  |  |
| **Sharing** | int(11) | NO |  |  |  |

**SMS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **id** | int(11) | NO | PRI |  | auto\_increment |
| **receiver** | int(11) | NO |  |  |  |
| **sender** | int(11) | NO |  |  |  |
| **message** | varchar(45) | NO |  |  |  |
| **date** | date | NO |  |  |  |
| **subject** | varchar(45) | YES |  |  |  |

**whiteboard\_sessions**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **user1** | varchar(20) | NO |  |  |  |
| **user2** | varchar(20) | NO |  |  |  |
| **Whiteboard\_id** | varchar(70) | NO |  |  |  |
| **Interview\_id** | Int | NO | PRI |  |  |
| **Tmpstamp** | TIMESTAMP | NO |  |  |  |

2.5.   Security/Privacy

The previous team defined security and privacy features for log in, registration, and access to the Yii framework. Moreover,  security and privacy features were added for each of the functionality that was implemented. Both of these are described below:

**Security Features**

·         User password will be hashed in the database.

Upon registration into the system, passwords entered will be hashed right away and will not be saved anywhere on the system.  Upon login, the password entered again will be hashed and the hashed data will be used to query the database.

**·         Administrator will be able to disable users and delete jobs**

An administrative console will be provided to a person to allow basic duties that may be needed in the future.  Due to abuse of the system, it may be necessary to delete jobs or disable users.

· **Yii access control rules**

The Yii framework provides access control with respect to any controller being used.  This access control will reject a subset of users (not logged, students, employers, etc…) from performing certain actions.  For example, users that are not logged in will not have access to profile pages.

**·         Cross-site Scripting Prevention**

The Yii framework takes measures against common web exploitations such as cross-site scripting or MySQL injection.  Using Yii, we can be rest assured that such things should not occur.

**·         Secure registration process**

The registration process is not as simple as most sites, especially for employers.  Administrators will have to verify employers after they register to ensure they are actual employers to ensure the integrity of the system.  Only then will they be able to post jobs and interact with students.

**Privacy**

Students and Employers are distinct user types and therefore have distinct permissions.  It may be necessary to allow employers to do actions that students cannot.  For example, students should not be able to post a job or schedule an interview, which clearly employers should be able to.  Likewise, students will only be able to view an employer’s profile and will not be able to view other students profiles, since it may contain information which should not be shared, such as phone number or email.

Below are these security and privacy features for the new functionality of the system:

**Whiteboard**

For the whiteboard functionality, the URL of the interview page will be hashed using the SHA1 function in PHP. The purpose of this is to make the whiteboard interaction more secure, preventing an intruder from typing in a URL which would give him/her access to the whiteboard session between a student and an employer.

**Collaborative Editor**

Maintain Document Access Boundaries:

Users collaborating on documents will have no access to documents other than the ones that have been shared in a particular session.

Maintain User Limit on Collaboration Sessions:

Users collaborating on documents will only be able to share/collaborate on a document with one other person at a time ( max 2 users collaborating on document )

# 3.      Detailed Design

The detailed design chapter introduces the system in terms of subsystems and the relationships among them. Initially, the system is decomposed into subsystem, with each subsystem described in terms of behavior and structure. Then, the static model is introduced in terms of subsystems with descriptions for each. After that, the dynamic model is presented in terms of state machine diagrams with the main control object for each subsystem. Finally, class interfaces and constraints for the main control object in each subsystem are presented.

3.1.   Overview

**Whiteboard Subsystem**

The whiteboard subsystem handles the interaction of students and employers through the Virtual Whiteboard. Adding to the already implemented video interview functionality Virtual Job Fair, a whiteboard seems like the next logical step. This whiteboard will allow students to make drawings while interviewing, which is ideal for different majors. The whiteboard will function real-time, meaning that there will be a single whiteboard which will be shared by both employers and students.

**Collaborative Editor Subsystem**

The Collaborative Editor Subsystem handles Interactions between users collaborating on a Document. It is meant to be intuitive, responsive and feature rich. The collaborative editor relies on Scientific Linux, ext3 filesystem, PHP and the Yii Framework to provide a robust Document Server and Document Repository Features. It relies on HTML, CSS, Js, and AJAX to provide a light and responsive editor that can be used by the users to collaborate on a document.

**Screen Share subsytem**

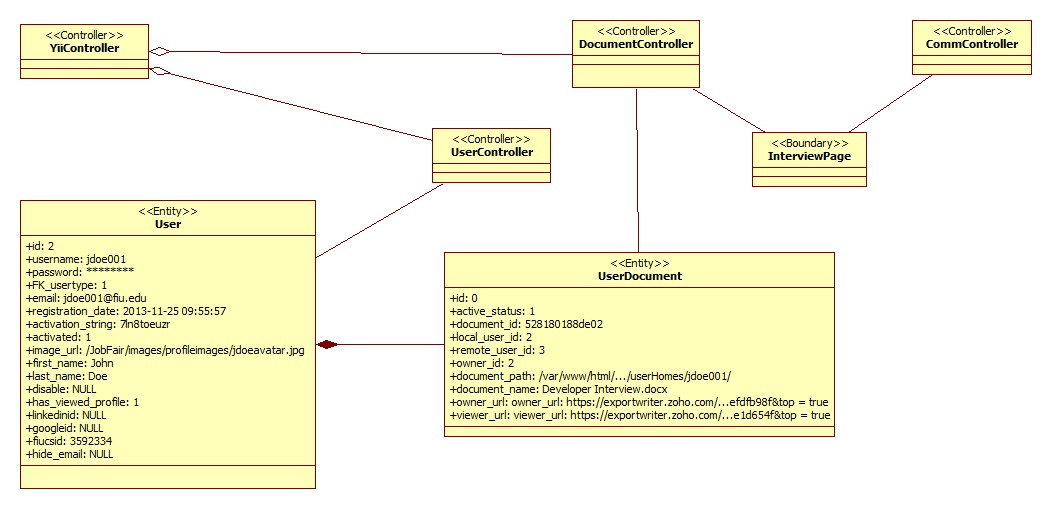
The ScreenShare subsystem handles the display and broadcasting of the user’s screens.  When users request a screen Share session, the system send a request to the Screen Leap Api which returns the state of the request as well as other usable information such as a viewer URL. This URL is stored in order to allow the other user to connect and see what is being displayed.  The screenLeap service makes it possible to provide this feature without having any additional services running in the system.  The ScreenShare .

**Sms subsystem**

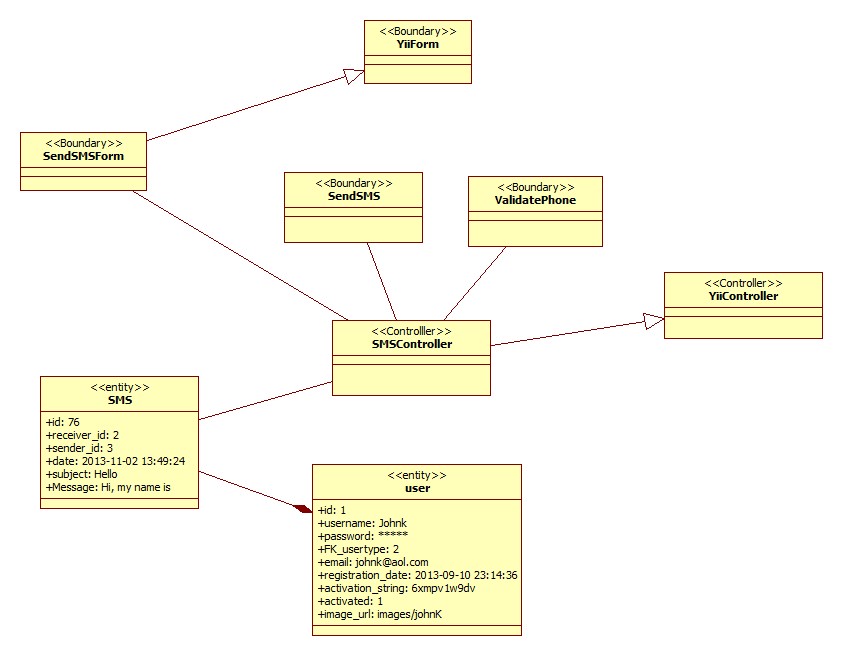
The sms subsystem allows students and employers alike to receive text message notifications about their upcoming interviews. Additionally, employers will be able to send direct text messages  to students they would like to interview with in a short notice.  The Twilio Service provides a very detailed and rich API that allows for these features to be implemented in a way that allows for reliability and security and most importantly ensuring the privacy of our users.

3.2.   Static model

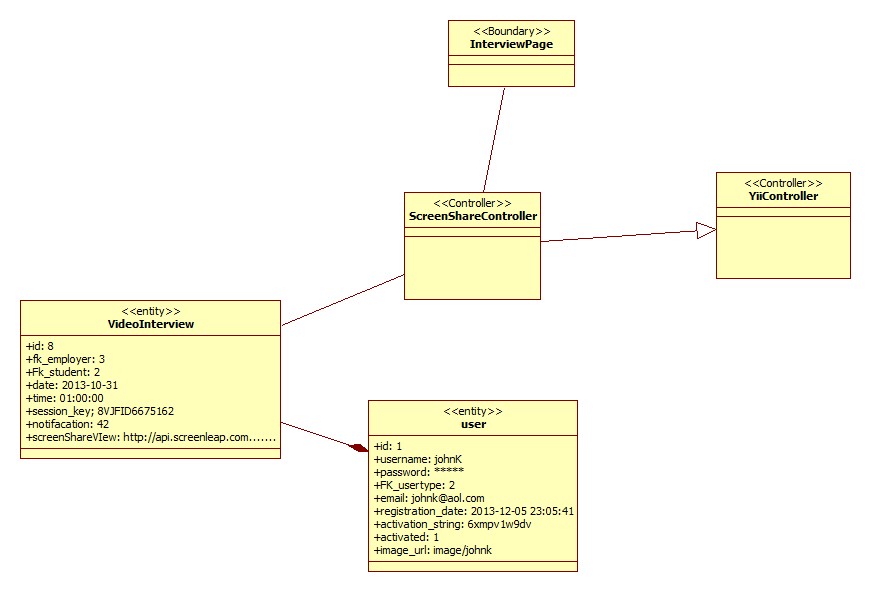
**Collaborative Editor Sub System**



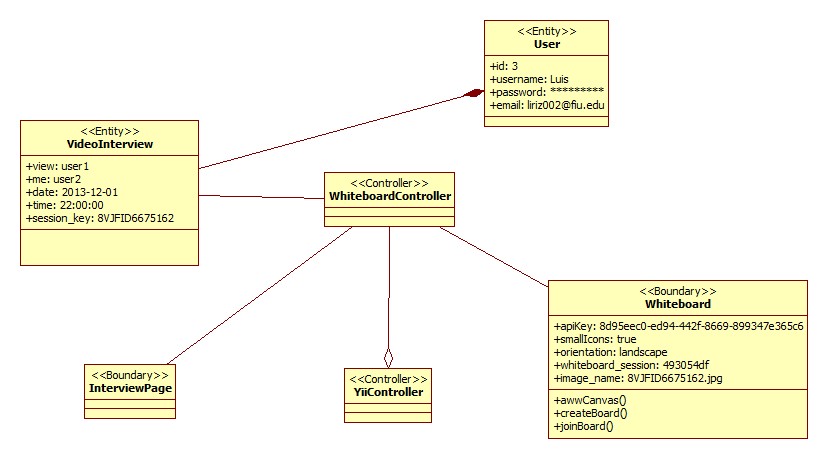
**SMS subsystem**



**Screen Share subsystem**

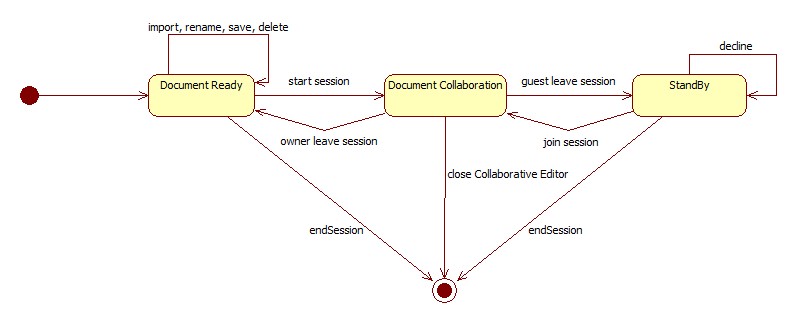


**Whiteboard subsystem**

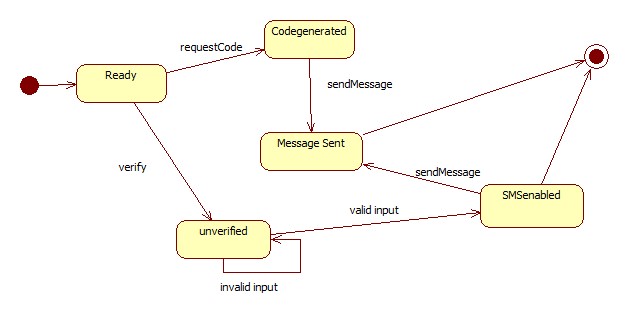


3.3.   Dynamic model

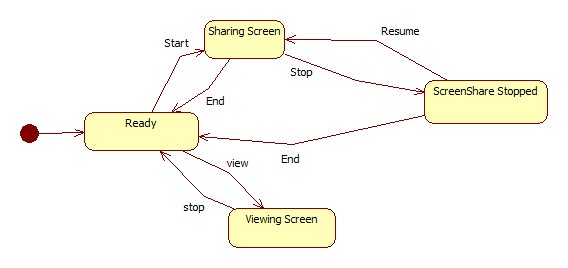
**Collaborative Editor State machine**



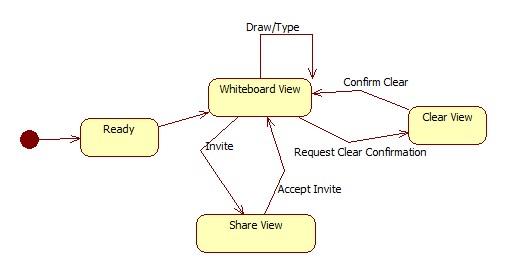
**Sms State machine**



**ScreenShare State machine**



**Whiteboard**



**3.4.**Code Specification

**Whiteboard Subsystem**

Method Signatures

**public function actionCheckWhiteboardExists()**

@precondition: an image has been uploaded server

@postcondition: view receives answer regarding existence of whiteboard

**public function actionStoreWhiteboardSession()**

@precondition: a valid image has been uploaded to the server

@postcondition: the "image\_name" field of the database is updated to reflect the new image

**public function actionCheckDrawingExists()**

@precondition: the user has clickd "View Drawing"

@postcondition: user receives message of whether there is an image for viewing

**public function actionUploadImage()**

@precondition: the usr has clicked on "Submit Drawing"

@postcondition: an image will have been uploaded to the server

**Collaborative Editor Subsystem**

Class:DocumentController

**Method Signatures:**

@dbg

@precondition: user has a message to debug

@postcondition: user has debugged a message

**private function dbg($msg)**

@create new editor

@precondition: interview has been scheduled

@precondition: user is at interview portal

@postcondition: user has a new editor

**private function createNewEditor($uN, $rN, $dI, $mD, $zDI)**

@open existing editor

@precondition: interview has been scheduled

@precondition: user is at interview portal

@postcondition: user has a opened an existing document

**private function openExistingEditor($uN, $rN, $dI, $mD, $zDI)**

@prepare home folder

@precondition: a new document will be created

@precondition: user is at interview portal

@postcondition: user has a folder to save documents

**private function prepareHomeFolder($username, $documentId)**

@save db

@precondition: user has data to save

@precondition: user is at interview portal

@postcondition: data is saved

**private function saveDb($recordArray)**

@mark documents innactive

@precondition: user is at interview portal

@precondition: user has selected a new document

@postcondition: user has a new document

**private function markDocumentsInactive(**

@create document

@precondition: interview has been scheduled

@precondition: user is at interview portal

@postcondition: user has a new document

**public function actionCreateDocument($rU)**

@import document

@precondition: interview has been scheduled

@precondition: user is at interview portal

@precondition: user has at least one document to import

@postcondition: user has a document imported

**public function actionImportDocument()**

@rename document

@precondition: interview has been scheduled

@precondition: user is at interview portal

@precondition: user has at least one document to rename

@postcondition: user has at a document with a different name

**public function actionRenameDocument($document, $newName)**

@save document

@precondition: interview has been scheduled

@precondition: user is at interview portal

@precondition: user has at least one document to save

@postcondition: user has saved a document

**public function actionSaveDocument()**

@delete document

@precondition: interview has been scheduled

@precondition: user is at interview portal

@precondition: user has at least one document to delete

@postcondition: user no longer has the document

**public function actionDeleteDocument($document)**

@share document

@precondition: interview has been scheduled

@precondition: user is at interview portal

@precondition: user has at least one document to share

@postcondition: user shared their document

**public function actionShareDocument($documentUrl, $documentName, $remoteUserId)**

@list document

@precondition: interview has been scheduled

@precondition: user is at interview portal

@precondition: user wants to manage documents

@postcondition: user can view his/her documents

**public function actionListDocument()**

@export document

@precondition: interview has been scheduled

@precondition: user is at interview portal

@precondition: user has at least a document to export

@postcondition: user has downloaded their document

**public function actionExportDocument()**

@open document

@precondition: interview has been scheduled

@precondition: user is at interview portal

@precondition: user wants to open an existing document

@postcondition: user has his document open

**public function actionOpenDocument($rU, $dI)**

**SMS Subsystem**

Class:SMScontroller

**Method Signatures:**

@send reminder to user and employers via SMS

@precondition: interview has been scheduled

@postcondition: users receive an SMS reminder

**public function actionsendReminder()**

@Employer sends SMS to student

@precondition: user is logged in, user has requested to send an SMS

@postcondition: user receives information about message action

**public function actionsendsms()**

@request phone number validation

@precondition: user has entered a phone number, user clicks on validate

@postcondition: Code is sent to users phone

**public function actionSendCode()**

@verify phone number code

@precondition: user enters validation code

@postcondition: phone is activated or validated accordingly

**public function actionValidation()**

@verify sms contents

@precondition: user has validated phone

@postcondition: SMS contents are valid to be sent

**public function actionverify()**

@specifies access rules

**public function accessRules**

**public function actionGetAutoComplete()**

@return auto-complete values to user

@postcondition: SMS usernames are returned to user

**ScreenShare Subsystem:**

Class:ScreenSharecontroller

**Method Signatures:**

@share screen

@precondition: interview has been scheduled

@postcondition: users shares his screen

**public function actionGetScreenLeap($interview)**

@get the url belonging to the interview

@precondition: user is sharing a screen

@postcondition: user receives screenshare url

**public function actionGetviewerUrl()**

@stop the screen Share

@precondition: user is sharing

@postcondition: Screen share is stopped and sharing field is cleared

**public function actionSetstop()**

**4.**Glossary

**-          Salary:** a periodic payment made to an employee in exchange for services provided. Salaries are provided in yearly terms.

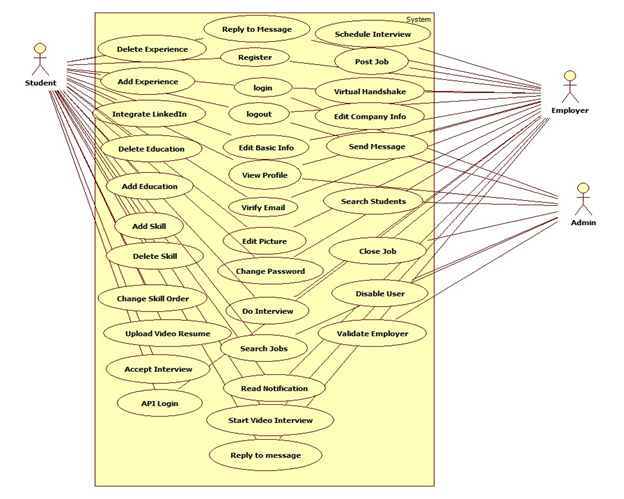
**-          Résumé:** a document which describes a student’s qualifications, skills and education

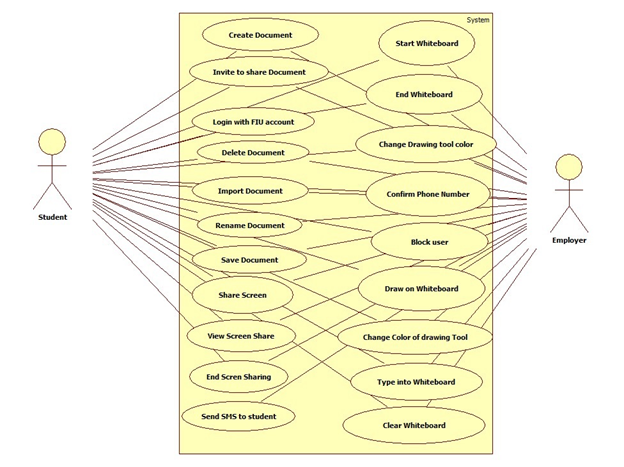
**-         Cover Letter:** a document which is used by students to introduce themselves to the companies that they are applying to. It usually goes together with a résumé

# 5.      Appendix

**5.1.   Appendix A - Use case diagram for use cases being implemented.**

**Current System’s Use Case Diagram**



**New System’s Use Case Diagram**

5.2.   Appendix B

The first 41 use cases were implemented by the previous group, while the remaining ones were added for Virtual Job Fair 2.0.

|  |  |
| --- | --- |
| 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 expectectation |

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

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

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

|  |  |
| --- | --- |
| Use Case ID | VJF-008 Upload Picture |
| Description | Allow user to change his profile picture |
| Actor | all user types |
| Pre-conditions | 1. User is on Virtual Job Fair 2. User is logged in |
| Steps | User clicks My Profile                 User is redirected to his profile page                 User clicks on the edit picture buttom                 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 |
| Pre-conditions | Employer is logged in and is at the home page |
| Steps | 1. Employer is typing student name on the search input. 2. Employer is clicking on the student that he want to view |
| Post-conditions | The employer is on the student profile view |
| Exceptions | The employer is typing a wrong student name that doesn't exist |
| Use Case ID | VJF-0023 Send Message |
| Description | Send a message to a user |
| Actor | A user |
| Pre-conditions | -User is in the compose message page |
| Steps | 1-User populates the ‘To’ field  2-User populates the ‘Subject’ field  3-User types in the message in the text area  4-User clicks ‘Send’ |
| Post-conditions | The system sends the message. The message appears in the inbox of the target user |
| Exceptions | -Inexistent username selected as the receiver of the message  -Wrong username format typed in the ‘To’ field |

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

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

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

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

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

|  |  |
| --- | --- |
| Use Case ID | VJF-0029 Post Job |
| Description | Employer posts a job for students to apply to |
| Actor | Employer |
| Pre-conditions | 1. Employer is logged in 2. Employer is on Home Page |
| Steps | 1. Employer clicks on Post Job Menu Item 2. Employer fills in job details (type, description, compensation, expire date) 3. Employer adds skills to posting if necessary 4. Employer clicks post job |
| Post-conditions | 1. Employer is taken to student match page to view students whose skillset is aligned with the job skillset |
| Exceptions | 1. Employer fills in job details incorrectly, is given an error |
| Use Case ID | VJF-0030 Virtual Handshake |
| Description | Employer gives student a virtual handshake to show interest |
| Actor | Employer, Student |
| Pre-conditions | 1. Employer has posted a job |
| Steps | 1. After employer posts job, he is taken to a student match page 2. Employer can review the list of students who matches with the job he posted 3. Employer clicks on “virtual handshake” for any student |
| Post-conditions | 1. Employer remains on student match page 2. Student receives a notification that the employer has shown interest in him for the position |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0031 Edit Company Info |
| Description | Employer changes his company Information |
| Actor | Employer |
| Pre-conditions | 1. Employer is logged in 2. Employer is viewing his profile |
| Steps | 1. Employer clicks on edit image in company info section 2. Employer changes text in any of the text boxes 3. Employer clicks on checkmark |
| Post-conditions | 1. New company info is saved 2. Employer is redirected back to profile page |
| Exceptions | Employer fills in invalid values for the fields |
| Use Case ID | VJF-0032 Search Jobs |
| Description | Student searches for jobs by skill |
| Actor | Student |
| Pre-conditions | 1. Student is logged in 2. Student is on home page |
| Steps | 1. Student clicks on job search text box 2. Student begins typing a skill 3. Student either selects skill from auto complete or fully types out skill 4. Student clicks submit |
| Post-conditions | 1. Student is redirected to search result page with relevant jobs, and option to search more jobs |
| Exceptions | None |
| Use Case ID | VJF-0033 Search Students |
| Description | Student searches for students by skill |
| Actor | Employer |
| Pre-conditions | 1. Employer is loggedin 2. Employer is on home page |
| Steps | 1. Employer clicks on job search text box 2. Employer begins typing a skill 3. Employer either selects skill from auto complete or fully types out skill 4. Employer clicks submit |
| Post-conditions | Employer is redirected to search result page with relevant students, and option to search more students |
| Exceptions | None |
| Use Case ID | VJF-0034 Close Job |
| Description | Close a job from further applications |
| Actor | Employer |
| Pre-conditions | 1. Employer is logged in |
| Steps | 1. Employer views his own profile 2. Employer selects a job from one of his own postings 3. Employer clicks on “Close Job” |
| Post-conditions | 1. User is redirected back to the job page 2. The job is closed |
| Exceptions | None |

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

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

|  |  |
| --- | --- |
| Use Case ID | VJF-0037 Apply to Job |
| Description | Student applies to an employers 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 share 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.   1. User is in the interview portal   3.      User is sharing screen |
| Steps | 1.      User clicks on end screen sharing |
| Post-conditions | 1.   User is not sharing screen  2.   System information is updated to allow other party to share screen  3.   Other party involved will not continue to see live feed from user screen |
| Use Case ID | VJF-045 Send SMS to student |
| Description | Allows Employer to send a text message to student |
| Actor | Employer |
| Pre-conditions | 1. User is logged in. 2. User is in the homepage 3. Student has a phone number associated with his account 4. Student has allowed employers to contact him through SMS |
| Steps | 1.      User clicks on send SMS  2.      User is redirected to SMS page  3.   User enters user name of student to contact  4.   User enters message  5.   User presses send |
| Post-conditions | Selected Student receives text message on his phone. |
| Exceptions | The student has not allowed contact by sms  Student has not entered a phone number |

|  |  |
| --- | --- |
| Use Case ID | VJF-046 Send interview reminder |
| Description | An interview reminder is sent to involved parties |
| Actor | Time |
| Pre-conditions | 1. An employer has set up an interview appointment 2. Interview starts in the next 30 minutes. |
| Steps | 1.      Database is continuously checked for interviews starting in the next 30 minutes  2.      After finding jobs that meet this criteria an email message is sent to involved parties about the event.  3.      If any of the accounts is set up to receive SMS then a text message will be sent as well. |
| Post-conditions | Parties involved in the interview will receive an email reminder and SMS according to set up permissions |
| Use Case ID | VJF-047 Confirm phone number |
| Description | Allows user to confirm and validate a phone number |
| Actor | Employee, Student |
| Pre-conditions | 1.      User entered a phone number in the system  2.      User is logged in.  3.      User is in the homepage  4. User phone has not been validated |
| Steps | 1.      User clicks on SMS page  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 | VJF-057  Start whiteboard functionality |
| Description | Allows user to start the whiteboard functionality |
| Actor | Employer or student |
| Pre-conditions | Actor is logged in and is in the interview portal page |
| Steps | 1. Actor clicks on “Start Whiteboard” 2. Use case ends when whiteboard becomes visible to the actor |
| Post-conditions | The user will have either an embedded whiteboard or a whiteboard on a new window |
| Exceptions | 1.   There is a connection error connecting with the whiteboard |

|  |  |
| --- | --- |
| Use Case ID | VJF-058 Draw on whiteboard |
| Description | Allows actor to draw on the whiteboard |
| Actor | Employer or student |
| Pre-conditions | Actor is logged in and currently has the whiteboard functionality open |
| Steps | 1. On the whiteboard screen, the actor clicks on the drawing tool 2. On the blank space in the whiteboard screen, actor clicks and holds the left click of the mouse 3. Actor drags the mouse around the whiteboard screen 4. Use case ends when actor releases the left click |
| Post-conditions | There will be a drawing on the whiteboard |
| Exceptions | None |
| Use Case ID | VJF-059 Change color of drawing tool |
| Description | Allows actor to change color of drawing tool |
| Actor | Employer or student |
| Pre-conditions | Actor is logged in and currently has the whiteboard functionality open |
| Steps | 1. Use case begins when, on the whiteboard screen, the actor clicks on the color palette tool 2. From the options that pop up, the actor selects the color he or she wants to use 3. Use case ends when the new color shows up \_\_\_\_\_\_ |
| Post-conditions | There will be a drawing on the whiteboard |
| Exceptions | 1.   User clicks outside of the color palette, and the color is not changed |

|  |  |
| --- | --- |
| Use Case ID | VJF-060 Type text into whiteboard |
| Description | Allows to type text into the whiteboard |
| Actor | Employer or student |
| Pre-conditions | Actor is logged in, is interviewing and  has the whiteboard functionality open |
| Steps | 1. Use case begins when, on the whiteboard screen, the actor clicks on the “Pencil” option on the left 2. From the sub-menu that pops up, the user shall click on “Text” 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 textbox, and the user shall type the text that will appear 5. The user shall press “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 not text will be shown |
| Use Case ID | VJF-061 Clear contents of whiteboad |
| Description | Allows actor to clear the current contents of the whiteboard |
| Actor | Employer or student |
| Pre-conditions | Actor is logged in, is interviewing and  has the whiteboard functionality open |
| Steps | 1.      Use case begins when, on the whiteboard screen, the actor clicks on the “Menu” option on the left  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-062    Block an user |
| Description | Allows a student to block an employer (and vice versa) |
| Actor | Employer or student |
| Pre-conditions | Actor is logged in and is viewing the profile of the user he/she wants to block |
| Steps | 1. Use case begins when the actor clicks on “Block User” 2. The system shall present a pop-up asking the actor whether he/she really wants to block the other user 3. The actor shall click “Yes” to confirm the blocking 4. Use case ends when system gives confirmation of blocking with a pop-up message |
| Post-conditions | 1. The blocking user will not be found when searched by the user who was blocked  2. Actor will be sent to profile of the actor who was blocked |
| Exceptions | In step 3, if the actor clicks on “No”, then no blocking will be done, and the actor will remain on the same website |

|  |  |
| --- | --- |
| Use Case ID | VJF-063    Unblock an user |
| Description | Allows a student to unblock an employer that was previously blocked (and vice versa) |
| Actor | Employer or student |
| Pre-conditions | Actor is logged in and is viewing the profile of the user he/she wants to unblock |
| Steps | 1.      Use case begins when the actor clicks on “Unblock User”  2.      The system shall present a pop-up asking the actor whether he/she really wants to unblock the other user   1. The actor shall click “Yes” to confirm the blocking 2. Use case ends when system gives confirmation of unblocking with a pop-up message |
| Post-conditions | 1. The blocking user will be able to be found when searched by the user who was blocked  2. Actor will be sent to the profile of the user who has unblocked |
| Exceptions | In step 1, if the user that is being unblocked had not been blocked before, then the actor will not see the “Unblock user” option |

5.3.   Appendix C

**Whiteboard**

public function startWhiteboard( );

public function startSharingImage();

public function clearWhiteboard();

public function endWhiteboard();

**Collaborative Editor Subsystem**

private function dbg($msg)

private function createNewEditor($uN, $rN, $dI, $mD, $zDI)

private function openExistingEditor($uN, $rN, $dI, $mD, $zDI)

private function prepareHomeFolder($username, $documentId)

private function saveDb($recordArray)

private function markDocumentsInactive(

public function actionCreateDocument($rU)

public function actionImportDocument()

public function actionRenameDocument($document, $newName)

public function actionSaveDocument()

public function actionDeleteDocument($document)

public function actionShareDocument($documentUrl, $documentName, $remoteUserId)

public function actionListDocument()

public function actionExportDocument()

public function actionOpenDocument($rU, $dI)

## 

## 5.4.   Appendix D - Diary of meeting and tasks.

**Project:** Virtual Job Fair

**Date:** Friday, September 6th, 2013

**Start Time:** 7:00PM

**End Time:** 10:00PM

**In Attendance:** Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:** None

**Agenda:** Start creating an overview of the project and call our mentor, Mr. Caraballo, to discuss specific issues about the project

Assigned Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Set up Yii framework to review the spring 2013 project | Create a list with information and to-do items based on Mr. Caraballo’s questions; review the documentation of the old project | Work on the definitions, acronyms, and abbreviations section of the Deliverable | Called Mr. Caraballo and asked project-specific questions |

**Project:** Virtual Job Fair

**Date:** Saturday, September 7th, 2013

**Start Time:** 9:00AM

**End Time:** 2:00PM

**In Attendance:** Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:** None

**Agenda:** Based on brainstorming and information gathered from Mr. Caraballo, clearly define the purpose of the new system when compared to the current system. Also, keep working on the first deliverable, which is due on Monday, September 9th, 2013.

Assigned Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Watch tutorials on the Yii framework; statically analyze the old group’s project to better understand that group’s implementation | Put the agreed-upon problem definition into words and complete the high-level requirements | Work on the diary of meetings and research free tools to create GANTT charts | Keep working on Deliverable #1 as a group. Also, define specific roles for each member of the group |

**Project:** Virtual Job Fair

**Date:** Sunday, September 8th, 2013

**Start Time:** 10:30AM

**End Time:** 7:00PM

**In Attendance:** Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:** None

**Agenda:** Finish setting up the framework in which to run the old group’s project, complete the Feasibility Study and Plan, and do the PowerPoint presentation for tomorrow

Assigned Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Check the old project’s code thoroughly and change absolute paths to match that of our framework; work on the cost matrix; create GANTT chart with project schedule information | Determine a rating scale and assign values to alternatives for the feasibility matrix; list the tasks, milestones, and deliverables to be used as reference for the GANTT chart | Introduce each chapter briefly; format the document for it to have a consistent look and feel; detail hardware and software requirements | Finish the first deliverable and brainstorm for the presentations that are due on Monday, September 10th, 2013 |

**Project:** Virtual Job Fair

**Date:** Sunday, September 10th, 2013

**Start Time:** 8:00PM

**End Time:** 10:00PM

**In Attendance:** Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:** None

**Agenda:** Set up the virtual machine that was provided by SCIS and install everything all the software that will be needed

Assigned Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Research different software packages and APIs for integration into the project | Set up the LAMP environment that will be used for development and testing in the virtual machine | Create a document that contains the proposed functionality of the system to discuss with our mentor and our instructor | Set up the virtual machine as soon as possible in order to start developing and adding features |

**Project:** Virtual Job Fair

**Date:** Wednesday, September 11th, 2013

**Start Time:** 7:00PM

**End Time:** 11:00PM

**In Attendance:** Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:** None

**Agenda:**

Assigned Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Review the existing code in order to get the video interview system working | Replace old paths from existing code to point our current virtual machine for interview process | Review the existing code in order to get the video interview system working | Set up the video interview system from the old’s project so it works on our virtual machine |

**Project:** Virtual Job Fair

**Date:** Saturday, September 14th, 2013

**Start Time:** 2:00PM

**End Time:** 4:30PM

**In Attendance:** Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:** None

**Agenda:**

Assigned Tasks:

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Plan GANTT chart for the whole project based on new schedule | Lead the selection of features of the system that will be implemented | Modify current system section of Deliverable #1 to reflect the project done in spring 2013 | Advance on Deliverable #1 |

**Project:** Virtual Job Fair

**Date:** Sunday, September 15th, 2013

**Start Time:** 2:00PM

**End Time:** 6:00PM

**In Attendance:** Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:** None

**Agenda:**

Assigned Tasks:

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Redo individual GANTT charts for the presentation, according to a new schedule | Formalize the features of the new system | Edit the features of the new system and add them to the document, each with a description | Finish Deliverable #1 and individual presentations for resubmission |

**Project:** Virtual Job Fair

**Date:** Tuesday, September 17th, 2013

**Start Time:** 7:30PM

**End Time:** 10:00PM

**In Attendance:** Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:** None

**Agenda:**

Assigned Tasks:

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Perform research on EasyRTC’s API and start integrating it into project | Perform research on how to be able to allow users to have a collaborative text editor available | Email 8 companies regarding the use of their whiteboard applications and correspondent API; do more research on how to implement the whiteboard functionality | Finish Deliverable #1 and individual presentations for resubmission |

**Project:** Virtual Job Fair

**Date:** Friday, September 20th, 2013

**Start Time:** 6:30PM

**End Time:** 10:30PM

**In Attendance:** Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:** None

**Agenda:**

Assigned Tasks:

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Develop screen-sharing use cases | Development collaborative text editor use cases and do research on feasibility of using Google API for the text editor | Develop whiteboard use cases based on the requirements analysis | Work on use cases; start working on sequence diagrams for these use cases |

**Project:** Virtual Job Fair

**Date:** Sunday, September 22nd, 2013

**Start Time:** 12:00PM

**End Time:** 9:00PM

**In Attendance:** Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:** None

**Agenda:**

Assigned Tasks:

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Develop class diagram by modifying last class diagram done by the old project’s members | Work on the use cases assuming the Google API will be used for the collaborative text editor | Work on functional/non-functional requirements for the system | Finish sequence diagrams for all of the use cases |

**Project:** Virtual Job Fair

**Date:** Monday, September 23rd, 2013

**Start Time:** 12:00PM

**End Time:** 9:00PM

**In Attendance:** Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:** None

**Agenda:**

Assigned Tasks:

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Write summaries of use case and static diagrams; develop the use case diagram in UML | Organize functional requirements appropriately and check overall flow of document before turning in | Edit document extensively and for revision after merging everyone’s parts | Finish Deliverable #2 to turn it in; complete presentations in order to possibly present on Monday, September 23rd, 2013 |

**Project:**Virtual Job Fair

**Date:**Thursday, October 3rd, 2013

**Start Time:** 7:00PM

**End Time:**9:00PM

**In Attendance:**Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:**None

**Agenda:**

Assigned Tasks:

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Verified persistent data store relationships, system integration, outlined general feature functionality | Worked on use case diagrams and state machine diagrams, revised ER diagram for correctness | Edit document extensively and for revision after merging everyone’s parts | Worked on diagrams and designed general functionality of features implemented |

**Project:**Virtual Job Fair

**Date:**Friday, October 6th, 2013

**Start Time:** 7:30PM

**End Time:**10:00PM

**In Attendance:**Jorge Fernandez, Luis Benjumea, Luis Irizarry

**Late:**None

**Agenda:**

Assigned Tasks:

|  |  |  |  |
| --- | --- | --- | --- |
| **Jorge F.** | **Luis B.** | **Luis I.** | **Group** |
| Refined persistent data store relationships, specific subsystem description | Refined diagrams, refined persistent data storage tables worked on specific subsystem description | Edit document extensively and for revision after merging everyone’s parts specific subsystem description | Work on description of subsystems |

# 6. References

The References chapter has references to external documents that have been used in this document.

Bhushan Agarwal, Bharat. *Software Engineering*. 2nd ed. New Delhi: Firewal Media, 2009. Print.