Requirements Document

CIS 4911 – Senior Project U01

Virtual Job Fair 2.0

**Members:**

Luis Irizarry

Jorge Fernandez

Luis Benjumea

**Instructor:**

Masoud Sadjadi

**Mentor:**

Juan Caraballo

**Date:**

Monday. September 23rd, 2013

**COPYRIGHT**

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**

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

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

Contents

[1. Introduction 10](#_Toc374330621)

[1.1 Problem Definition 10](#_Toc374330622)

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

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

[1.4 Overview of document 11](#_Toc374330625)

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

[3. Project Plan 13](#_Toc374330627)

[3.1 Project organization 13](#_Toc374330628)

[3.2 Work breakdown 15](#_Toc374330629)

[3.3 Cost Estimate 17](#_Toc374330630)

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

[4.1 Functional Requirements 18](#_Toc374330632)

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

[4.2.1 Use case model 32](#_Toc374330634)

[4.2.2 Static model 32](#_Toc374330635)

[4.2.3 Dynamic model 34](#_Toc374330636)

[5. Glossary 34](#_Toc374330637)

[6. Appendix 35](#_Toc374330638)

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

[6.2 Appendix B 60](#_Toc374330640)

[6.3 Appendix C - Static UML diagram 62](#_Toc374330641)

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

[Select Upload Image 77](#_Toc374330643)

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

[Collaborative Editor 77](#_Toc374330646)

[SMS 78](#_Toc374330648)

[ScreenShare 78](#_Toc374330650)

[Whiteboard 79](#_Toc374330652)

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

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

[Whiteboard 84](#_Toc374330657)

[Image Sharing 85](#_Toc374330658)

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

[7. References 93](#_Toc374330661)

# 1. Introduction

The introductory chapter gives some background information about the Virtual Job Fair 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. Also, background information was provided on the previous project. 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 Scope of system.

One of the main purposes of the Virtual Job Fair is to expand the scope of the original. While the original project was geared towards Computer Science majors at FIU, our project expands on that by focusing on the global population. In our project, basically a student in any given university in the world can be interviewed by any employer located anywhere. This helps solve the global problem of employers being unable to interview students who might be potential employees who are located remotely.

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

**Definitions**

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

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

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

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

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

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

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

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

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

**Acronyms**

- **VJF**: Virtual Job Fair

- **FIU:** Florida International University

- **GPA:** Grade point average

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

**Abbreviations**

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

## 1.4 Overview of document

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

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

# 2. Current System (limitations and problems)

The current system was implemented in the spring 2013 semester. As it is, it allows students and employers to engage in interviews through online video chats. It also allows employers to post jobs for students to browse and apply. Moreover, it offers a messaging system, through which employers and students can communicate privately. Also, the current system allows users to log in using their LinkedIn, Google or FIU account.

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

**Limitations and Constraints**

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

**- Aimed towards FIU Computer Science students**: the scope of the system is geared towards students who are both FIU students and Computer Science majors. This leaves out two major groups of people:

a) Students who study at FIU but whose major is not Computer Science

b) College students who do not study at FIU, but at other universities instead

Therefore, it is geared toward a very narrow group of people, even when potentially any college student could benefit from this service

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

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

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

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

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

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

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

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

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

# 3. Project Plan

The project plan chapter introduces VJF 2.0 from a project management perspective. Firstly, the project organization is described, with the roles for each member listed. After that, milestones, tasks, and deliverables will be listed. Finally, a cost estimate for whole project is presented in terms of a feasibility matrix.

## 3.1 Project organization

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

Luis Benjumea will be responsible for integrating a collaborative text editor that will be available during the interview. Also, he will be working with the API integration of the Senior Project website.

Jorge Fernandez will be responsible for screen-sharing functionality to be used during interview, allowing students and employers to share screens during interview. Also, he will be handling the integration of an SMS/email notification system to remind students and employers of upcoming interviews.

Luis Irizarry will be responsible for integrating a virtual whiteboard into the interview process. Also, he will be working on an image-sharing feature which allows users to dynamically share images during a live interview.

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

|  |  |  |
| --- | --- | --- |
| **Team Member** | **Primary Task** | **General Task** |
| Jorge Fernandez | Front End Developer  Tester  Timekeeper | Additional shared tasks |
| Luis Irizarry | Team Leader  Developer  Database Manager | Additional shared tasks |
| Luis Benjumea | Back End Developer  Configuration Manager  Document Editor | Additional shared tasks |

## 3.2 Work breakdown

For a visual representation of all tasks, milestones and deliverables, please refer to the GANTT chart at the end of this section. Also, a diary with descriptions of daily meetings can be found in Appendix B.

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

**Tasks**

Requirement Elicitation

Requirement Analysis

Framework Setup

Deliverable 1 (Feasibility Study and Project Plan)

Database Design

Database Implementation

System Design

Deliverable 2 (Requirements Document)

Object Design

Deliverable 3 (Design Document)

System Implementation

Test Case Design

Test Case Implementation

Integration and System Testing

Deliverable 4 (Final Deliverable)

Documentation

Project Deployment

**Milestones**

Analysis Milestone

Design Milestone

Testing Milestone

Documentation Milestone

**Deliverables**

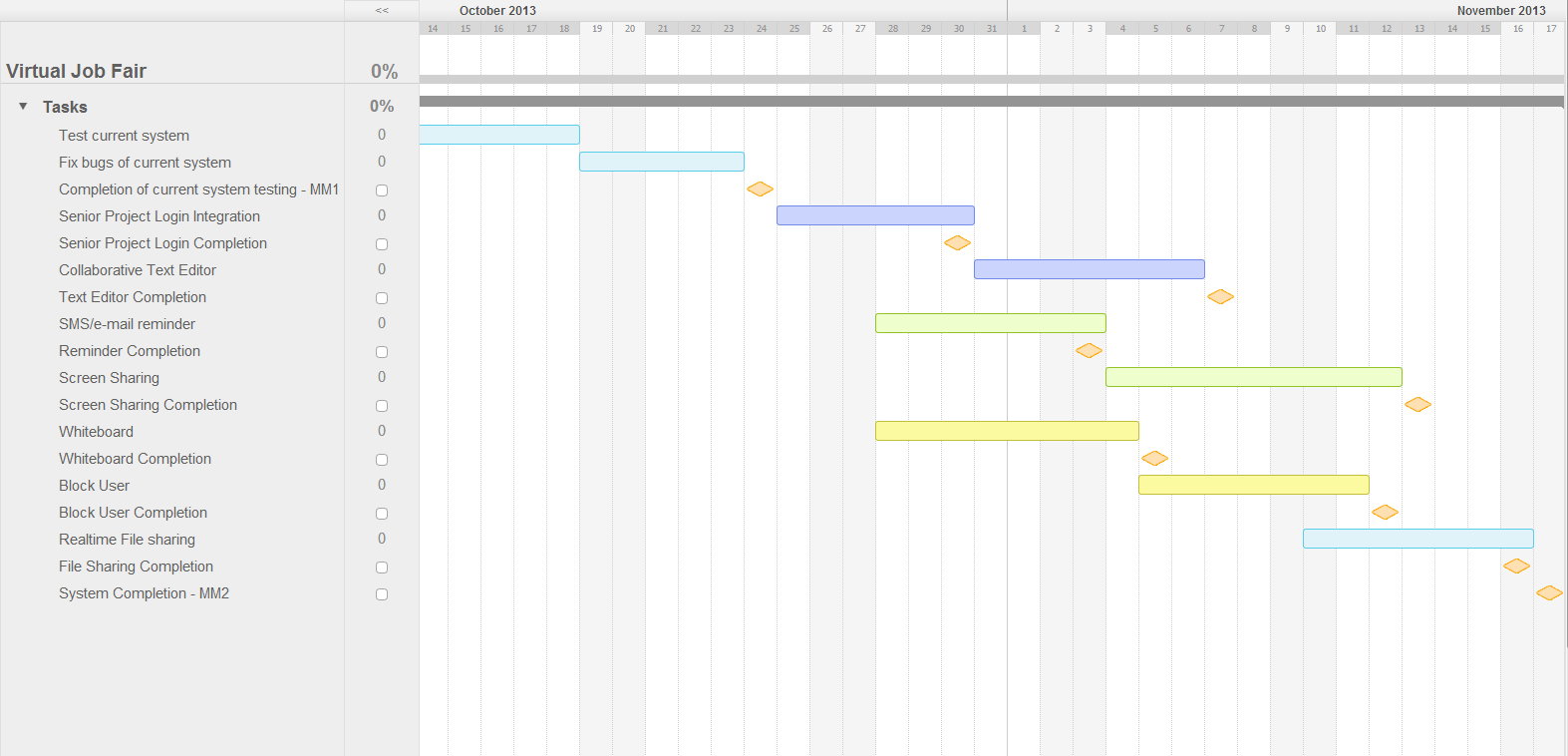
- **Deliverable #1 (Feasibility Study & Plan):** September 9th, 2013

**- Deliverable #2 (Requirements Document):** September 23rd, 2013

**- Deliverable #3 (Design Document):** October 7th, 2013

**- Deliverable #4 (Final Deliverable):** December 9th, 2013

The following GANTT chart contains the project schedule for the semester:



## 3.3 Cost Estimate

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

|  |  |
| --- | --- |
| **Item** | **Item Cost** |
| Yii Framework | $0.00 |
| Twitter Bootstrap | $0.00 |
| Easy RTC | $0.00 |
| Shared Whiteboard | $120.00 |
| Hardware | $1200.00 |
| Development | $0.00 |
| Testing | $0.00 |
| Contingencies | $260.00 (~20% of total cost) |
|  | **Total:**$1570.00 |

# 4. Proposed System Requirements

This chapter introduces the functional requirements for the project in terms of use cases, describing the usability, reliability, performance and supportability of each. Also, an analysis is given for each of the models presented: use case model, static models and the dynamic models. Together, these models give an overview of the behavior and structure of the system.

## 4.1 Functional Requirements

**Current System’s Functional Requirements**

The system shall…

· **Allow students and employers to register**

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

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

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

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

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

- **Usability**: Data displayed in profiles is easy to follow. Students are only able to see their own profile and the employer ones. Employers can see all student profiles.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- **Usability**: The ability to add skills to a profile is simple and understandable. It can be done by using LinkedIn connect or adding them manually.

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

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

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

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

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

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

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

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

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

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

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

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

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

· **Allow students to integrate with their LinkedIn account to provide education and experience information (security)**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

· **Allow employers to post jobs**

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

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

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

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

· **Allow employers to close a job posting**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

· **Allow employers to view student profiles**

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

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

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

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

· **Allow employers to send messages to students**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**New System’s Functional Requirements**

**· Allow users to create a new shared document.**

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

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

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

- **Supportability:** The system should be supported on Google Chrome version 29+ The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**· Allow users to open a document.**

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

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

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

- **Supportability:** The system should be supported on Google Chrome version 29+ The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**· Allow users to share a document.**

- **Usability:** The invitation should be transparent to the users upon document creation. The invitation interface on new documents should be easy to use.

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

- **Performance:** The system should respond in less than 1 seconds.

- **Supportability:** The system should be supported on Google Chrome version 29+ The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**· Allow users to delete a document.**

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

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

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

- **Supportability:** The system should be supported on Google Chrome version 29+ The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**· Allow users to import a document.**

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

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

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

- **Supportability:** The system should be supported on Google Chrome version 29+ The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**· Allow users to export a document.**

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

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

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

- **Supportability:** The system should be supported on Google Chrome version 29+ The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**· Allow users to rename document.**

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

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

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

- **Supportability:** The system should be supported on Google Chrome version 29+ The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**· Allow users to save a shared document.**

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

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

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

- **Supportability:** The system should be supported on Google Chrome version 29+ The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

**· Maintain access boundaries between non-collaborating temporary accounts (security).**

- **Usability:** Temporary accounts not related to a collaborating session shall have no access to that session’s documents.

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

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

- **Supportability:** The system should be supported on Google Chrome version 29+ The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

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

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

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

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

- **Supportability:** The system should be supported on Google Chrome version 29+ The system should be easily maintainable, allowing for improvements, corrections and adaptation in environment, requirement and/or specification changes.

· **Allow users to share their screens**

- **Usability**: User should be able to share their screens solely with mouse clicks

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

- **Performance**: The system should start streaming content in under 3 seconds

- **Supportability**: Screen sharing page should be supported by Google Chrome versions 29 and up.

· **Allow users to view shared screens**

- **Usability**: User should be able to watch a shared screen with single mouse click

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

- **Performance**: The system should start displaying the screen in under 2 seconds

- **Supportability**: Screen sharing page should be supported by Google Chrome versions 29 and up.

· **Allow users to stop sharing their screens**

- **Usability**: User should be able to stop sharing their screens with a single mouse click

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

- **Performance**: The system should stop the stream in under 1 second.

- **Supportability**: Screen sharing page should be supported by Google Chrome versions 29 and up.

· **Allow employers to send SMS to students**

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

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

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

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

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

- **Usability**: Not applicable

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

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

- **Supportability**: Not applicable

· **Allow users to confirm their phone numbers**

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

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

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

- **Supportability**: Confirm phone number page should be supported by IE, Firefox, Chrome and Safari

**· Allow students and employers to start using the whiteboard functionality**

**- Usability:** Starting the whiteboard is a matter of clicking a button while interviewing, making it a fairly easy process to do

**- Reliability:** The whiteboard tool should be available 90% of the time, since not every major/employer will use it

**- Performance:** The whiteboard should be started within 1 second

**- Supportability:** The system be supported by Chrome version 29 or above

**· Allow students and employers to upload an image to share during an interview**

**- Usability:** if an image has been selected for uploading, actually uploading it to the system is as simple as click the “Submit Drawing” button, making it simple to use

**- Reliability:** Uploading images should work 99% of the time, since it is a critical function of the image-sharing subsystem

**- Performance:** Depending on its size, the maximum time it should take to upload an image should be 10 seconds

**- Supportability:** The system be supported by Chrome version 29 or above

· **Allow students to view images uploaded by the other party in an interview**

**- Usability:** viewing an image which has been uploaded already relies simply on clicking a “View Drawing” button, so it is fairly simple to use

**- Reliability:** as uploading an image, viewing an image should be working 99% of the time, since it is a critical function of the image-sharing subsystem

**- Performance:** depending on the size of the image, the maximum time it should take to view an uploaded image should be 10 seconds

**- Supportability:** The system be supported by Chrome version 29 or above

**· Allow students and employers show or restore a whiteboard session**

**- Usability:** starting a whiteboard session involves only clicking the “Whiteboard” button below the video chat section of the interview page, making it a simple process

**- Reliability:** the whiteboard should be available 99% of the time, since it is a main feature of the system

**- Performance:** displaying the whiteboard should take 1 second, since a default whiteboard is not preloaded with any information

**- Supportability:** The system be supported by Chrome version 29 or above

**· Allow students and employers to select an image to upload to the server for sharing purposes**

**- Usability:** selecting an image requires basic experience with a browsing dialog, which should be fairly common knowledge, making it a simple functionality

**- Reliability:** the ability to select an image is critical to the image-sharing feature, so it should be available 99% of the time

**- Performance:** since selecting an image does not depend on image size, this functionality should take less than 1 second

**- Supportability:** The system be supported by Chrome version 29 or above

**· Allow students and employers to draw using the whiteboard**

**- Usability:** drawing on the whiteboard simply requires to drag the mouse through the whiteboard’s area, which makes it a simple process

**- Reliability:** without being able to draw in the whiteboard, the whiteboard serves no purpose, so drawing in the whiteboard should be available 99.99% of the time

**- Performance:** since the whiteboard is self-contained and does not need any API calls, the drawing should take less than half a second

**- Supportability:** The system be supported by Chrome version 29 or above

**Allow students and employers to change the color of the drawing pencil tool**

**- Usability:** changing the color is a matter of clicking on the new color desired, so it is fairly simple to do

**- Reliability:** changing the color of the drawing tool is not of the utmost importance, so it is expected to work 90% of the time

**- Performance:** since the whiteboard is self-contained and does not need any API calls, changing the color should take less than half a second

**- Supportability:** The system be supported by Chrome version 29 or above

**Allow students and employers to type text into the whiteboard**

**- Usability:** typing text into the whiteboard requires clicking a button and typing text, so it is fairly simple to use

**- Reliability:** since typing text is not the main whiteboard feature, it is expected to work 90% of the time

**- Performance:** the text should appear in the whiteboard in less than three seconds

**- Supportability:** The system be supported by Chrome version 29 or above

**· Allow students and employers to clear the drawings of the whiteboard**

**- Usability:** Clearing the contents is a matter of clicking a single button, making it a fairly easy process to conduct

**- Reliability:** It should be available 99% of the time that the whiteboard is being used, given that new drawings might be required

**- Performance:** Whiteboard screen should be cleared out within 1 second

**- Supportability:** The system be supported by Chrome version 29 or above

**Allow students and employers to partially erase drawings from the whiteboard**

**- Usability:** Partially erasing from the drawing is a matter of clicking a single button, drawing the cursor and releasing the click, making it a fairly easy process to conduct

**- Reliability:** It should be available 99% of the time that the whiteboard is being used, given that a user might make a mistake while drawing or that new drawings might be required

**- Performance:** selected sections should be erased within 1 second

**- Supportability:** The system be supported by Chrome version 29 or above

## 4.2 Analysis of System Requirements

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

### 4.2.1 Use case model

The use case diagram describes the overall view and functionality that we will add to the system. The proposed functionality deals with the two type of users displayed in the diagram. Both student and employers are the targets of these functional requirements since the majority of use cases dealing with the administrator user type was developed by the previous team working on Virtual Job Fair V1.

### 4.2.2 Static model

**Object Diagrams Analysis**

**Whiteboard**

An object from the Whiteboard class is instantiated when a user first requests to use the whiteboard. The instantiated Whiteboard object has 4 properties. autoJoin is a feature that deals with automatically joining a whiteboard. “toolbar” determines whether the toolbar will be shown on the whiteboard. “smallIcons” decides whether small icons will be forced on the whiteboard. Finally, “orientation” determines the view of the whiteboard (acceptable values are “portrait”, “landscape” and “auto”.

**Document**

An object which represents the Document class gets instantiated upon initiation of the Collaborative Text Editor feature, more specifically, upon the request of any of the features that involve document manipulation. The Document Object describes 10 properties:

- id: unique identifier.

- active\_status: represents if the current document is in an active state.

- document\_id: represents a unique document, it's an internal unique identifier that represents a file.

- local\_user\_id: contains the unique user id of the user who creates a local document.

- remote\_user\_id: contains the unique user id of the user who collaborates with the document remotely.

- owner\_id: which represents which of the users collaborating in the document owns it.

- document\_path: the document path on the filesystem.

- document\_name: represents the document identifier from the point of the user, not the system.

- owner\_url: the url for the Zoho API that accesses and displays a specific document.

-viewer\_url: the url for the Zoho API that permits another user to join a specific document.

**SMS**

An object from the SMS class is instantiated when a SMS message will be sent to a particular user. The object has 6 properties which are useful for any interaction involving a text message. The id property specifies the unique identifier of this text message, the receiver and sender properties specify the identification numbers of the parties involved, the message field contains the content of the SMS, then there is the date field which stores the current date and finally subject which holds the topic of the message.

The other two objects, BasicInfo and VideoInterview, were already explained by the old group.

**Class Diagram Analysis**

The Class Diagram in Appendix C depicts the additional classes that will be added to the original system. It portrays the proposed changes to the scheme, where we build on the implementation of the previous team. The new proposed classes are colored green in order to emphasize the difference between the new and old implementations. In addition, the Diagram follows UML notation in order to facilitate the developers as well as adhere to industry standards.

## 4.2.3 Dynamic model

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

# 5. Glossary

- **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

# 6. Appendix

## 6.1 Appendix A - Complete use cases

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

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

|  |  |
| --- | --- |
| Use Case ID | **VJF-044 End Screen Sharing** |
| Description | Allow a user to end current screen sharing |
| Actor | Student, Employer |
| Pre-conditions | 1. User is logged in.   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 |
| Steps | 1. User clicks on validate phone number  2. User is redirected to validate number page  3. User clicks on validate phone  4. An SMS message is sent to user’ phone  5. User enters received code  6. User presses validate button |
| Post-conditions | The system is updated to reflect phone validation |
| Exceptions | User enters a wrong authentication code |

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

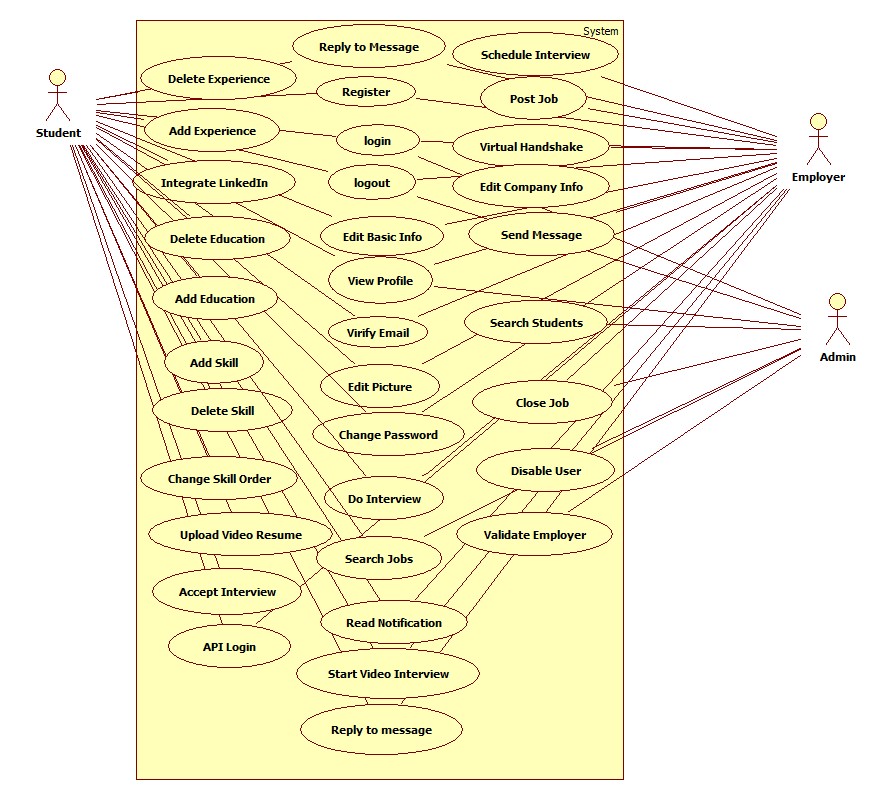
|  |  |
| --- | --- |
| **Use Case ID** | **VJF-064 Type Text Into Whiteboard** |
| **Description** | Allows a user to type text into the whiteboard |
| **Actor** | Employer or student |
| **Pre-conditions** | 1) A interview session is active  2) A whiteboard session has been started |
| **Steps** |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

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

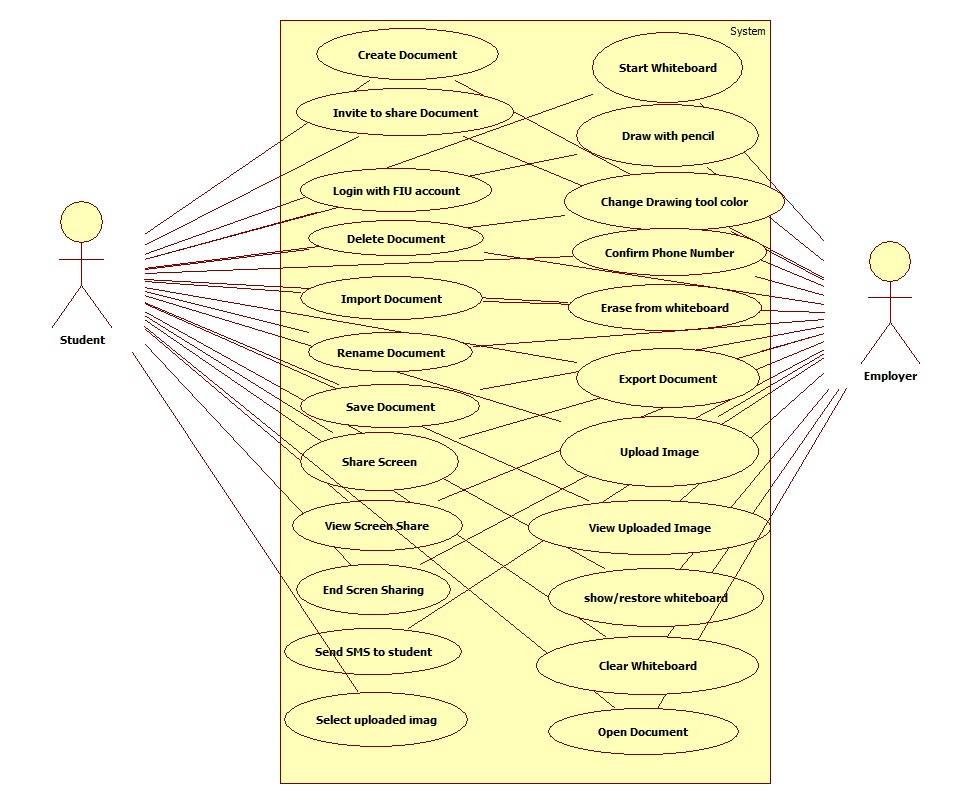
|  |  |
| --- | --- |
| **Use Case ID** | **VJF-066 Erase From Whiteboard** |
| **Description** | Allows users to clear to partially erase drawings |
| **Actor** | Employer or student |
| **Pre-conditions** | 1) A interview session is active  2) A whiteboard session has been started  3) The user has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

## 6.2 Appendix B

**Current System’s Use Case Diagram**



**New System’s Use Case Diagram**

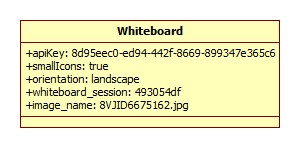


## 6.3 Appendix C - Static UML diagram

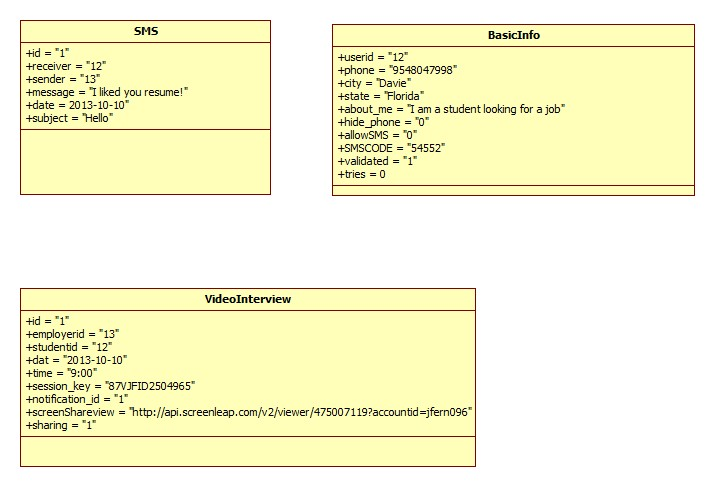
**Object Diagrams**

These object diagrams represent the new features that we implemented

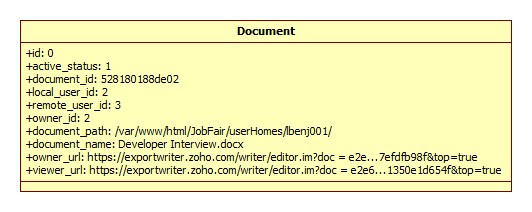
**Whiteboard**



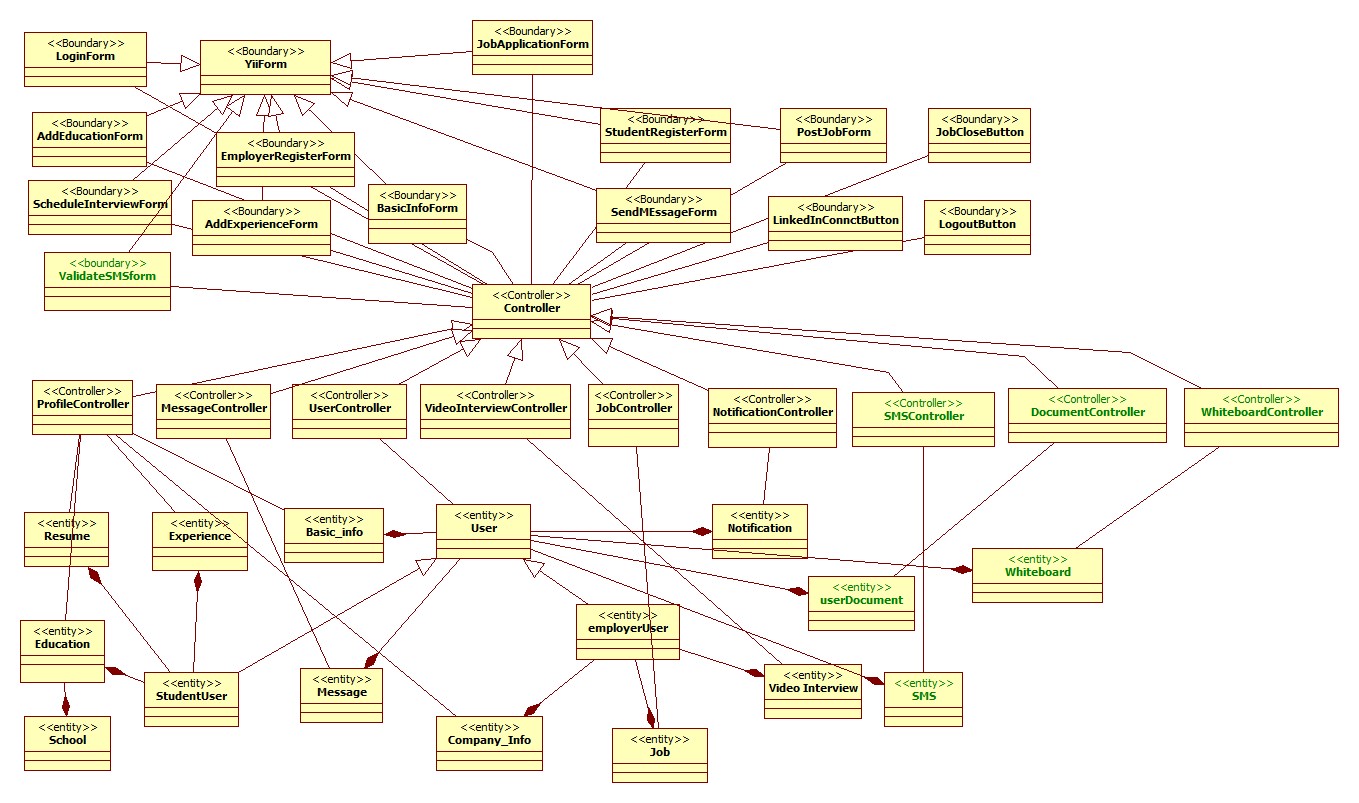
**SMS, BasicInfo and VideoInterview**

****

**Document**



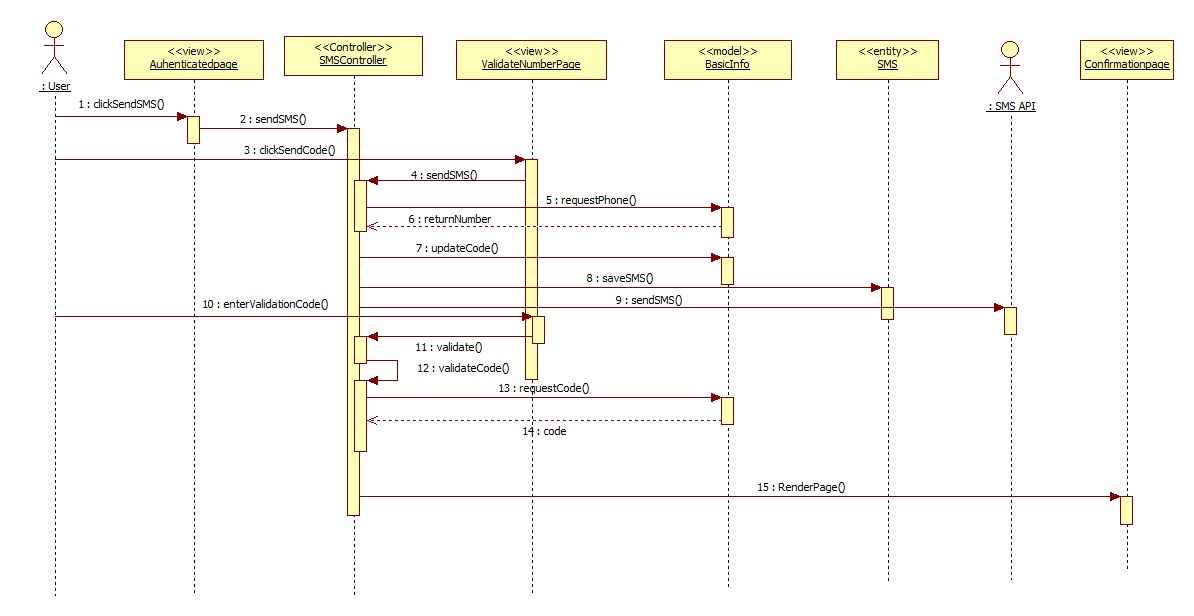
**Class Diagram**



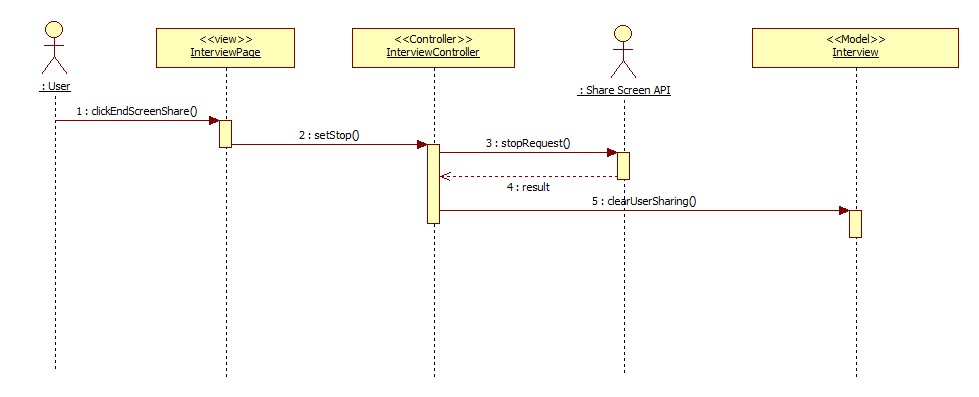
## 6.4 Appendix D - Dynamic UML diagrams

**New System’s Sequence Diagrams**

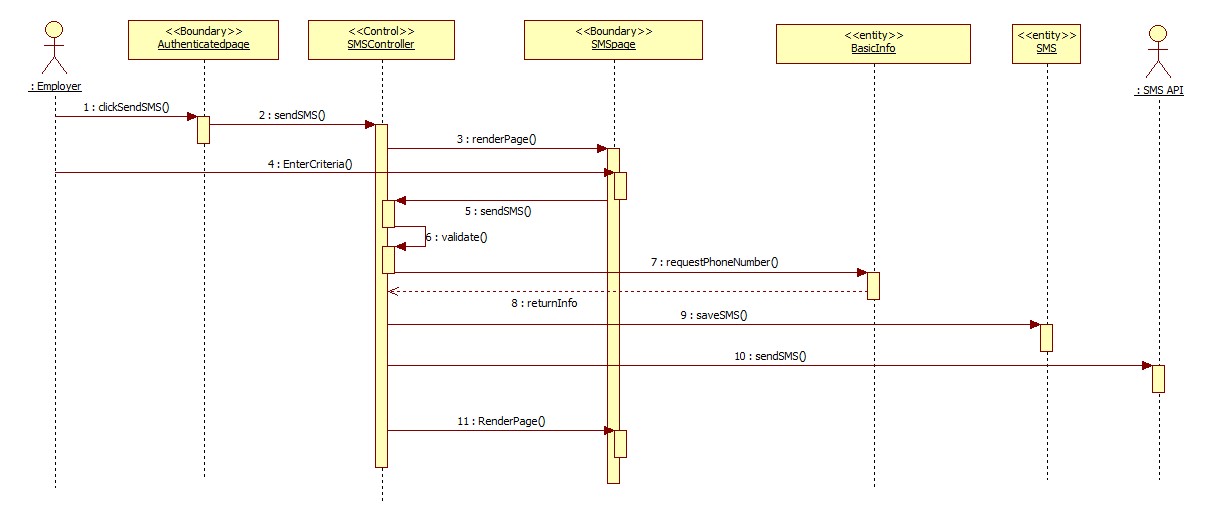
Confirm Phone Number



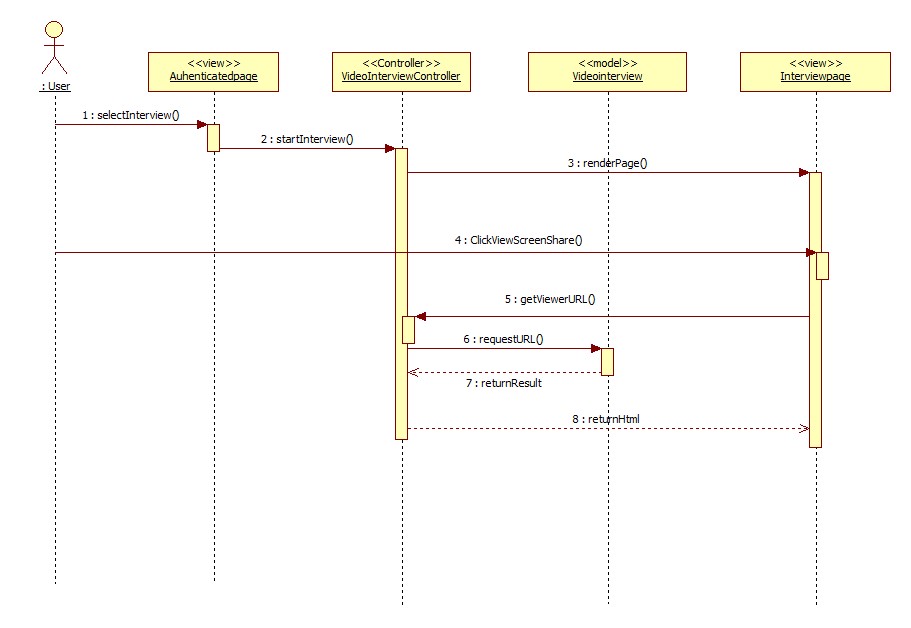
End Screen Share



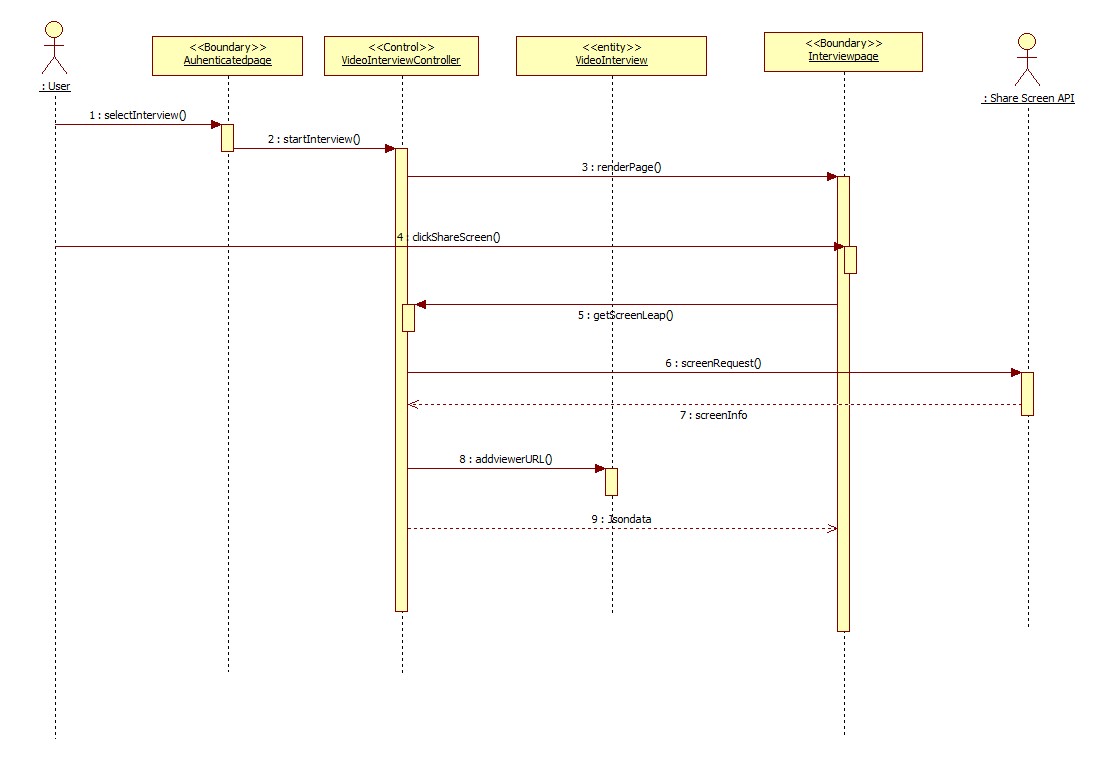
Send SMS



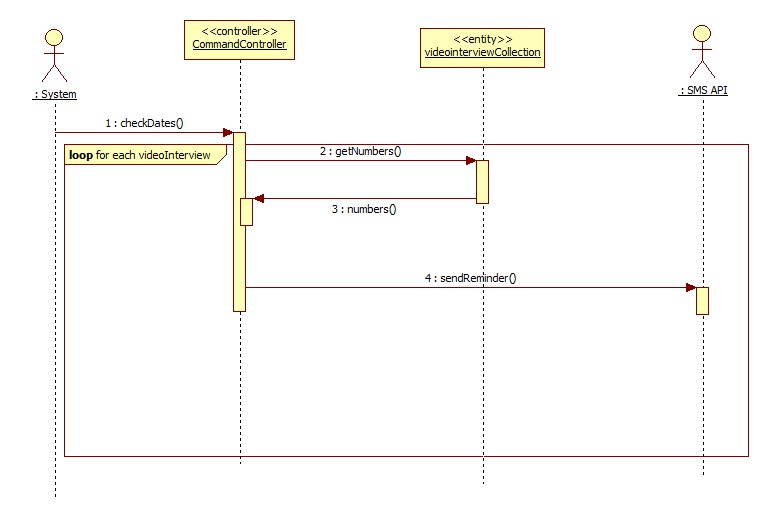
View Screen Share



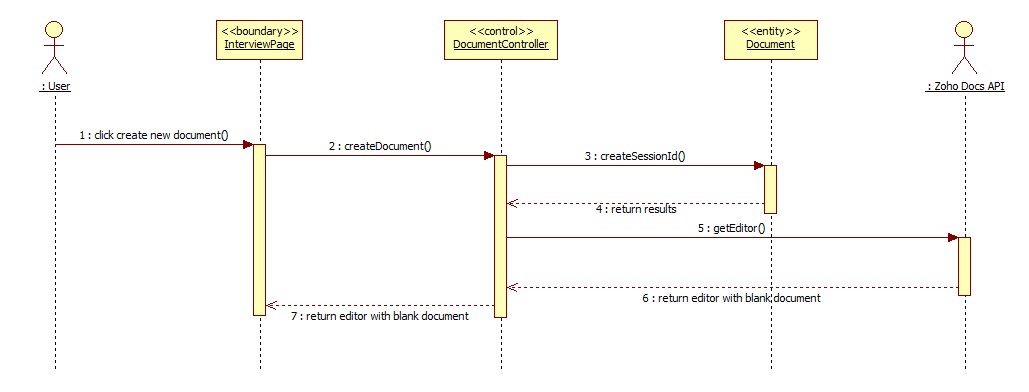
Share Screen



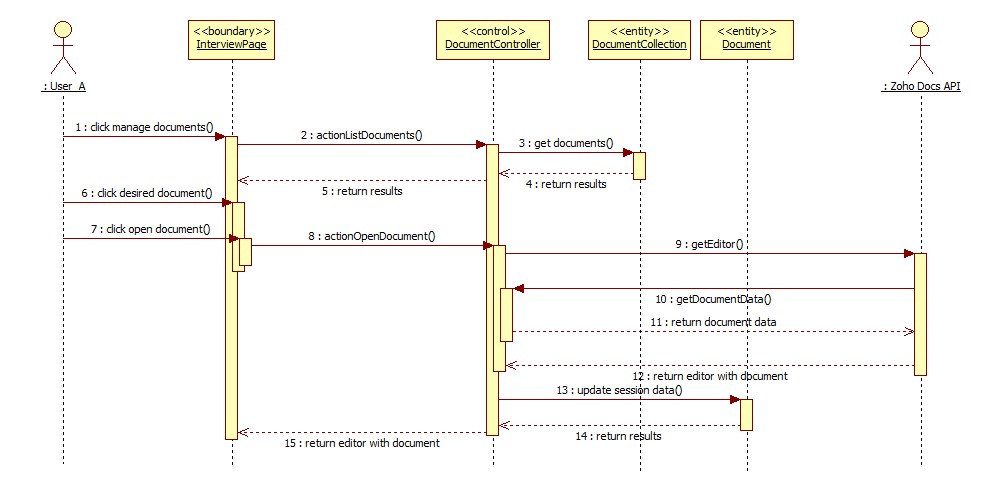
Send Reminder



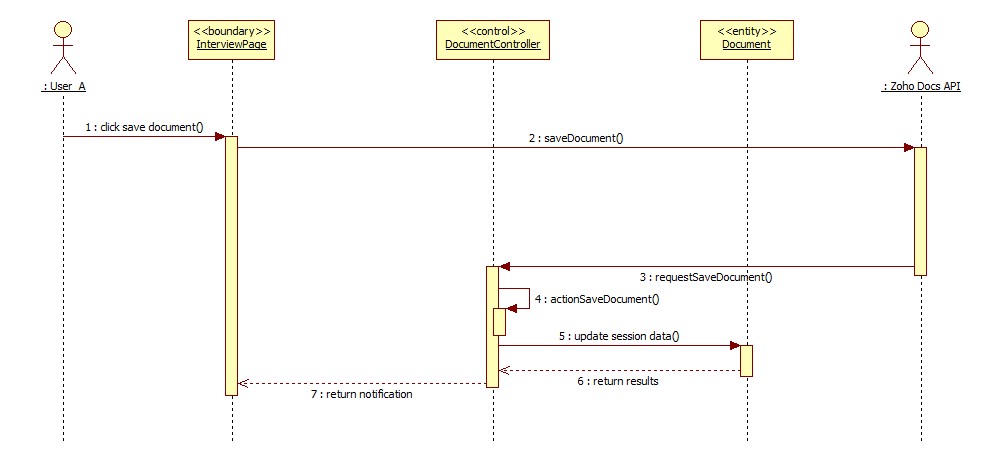
Create New Document



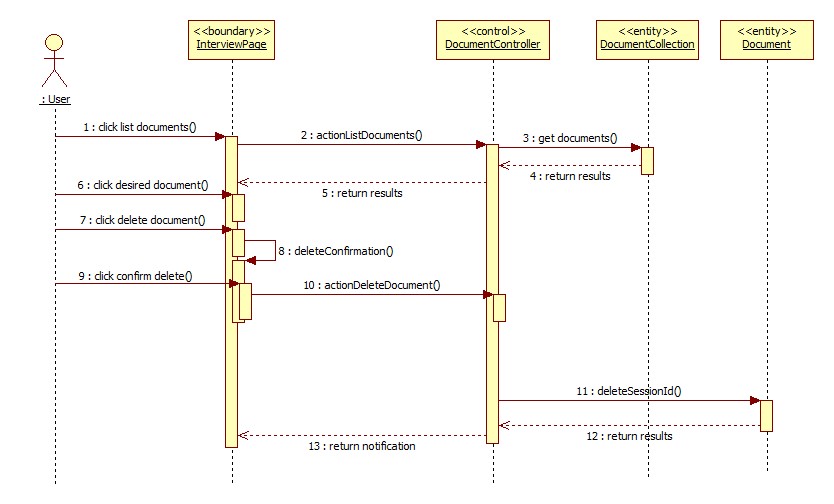
Open Document



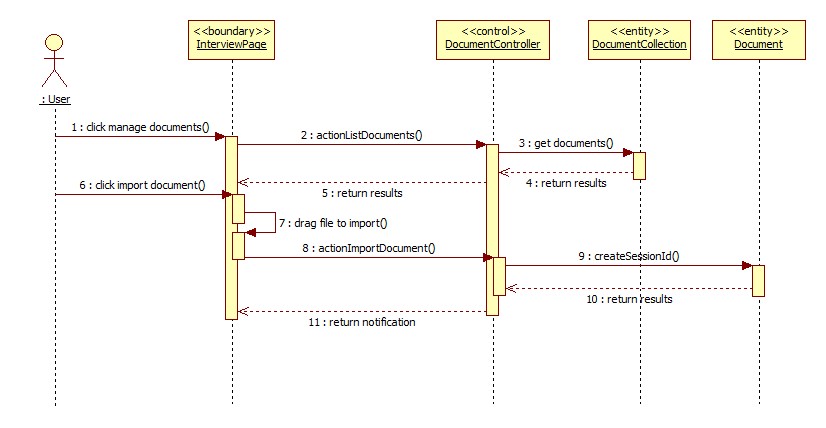
Save Document



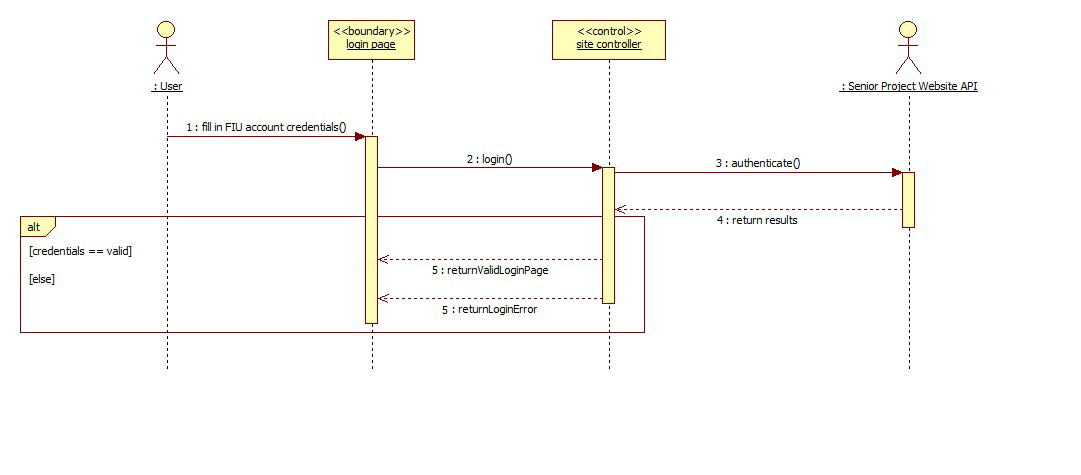
Delete Document



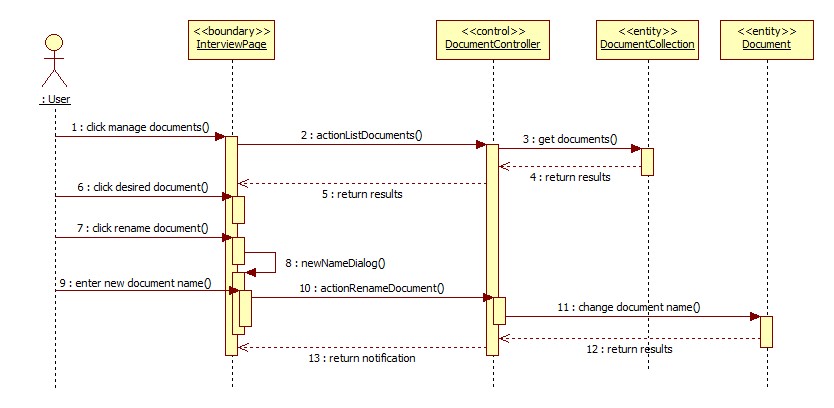
Import Document



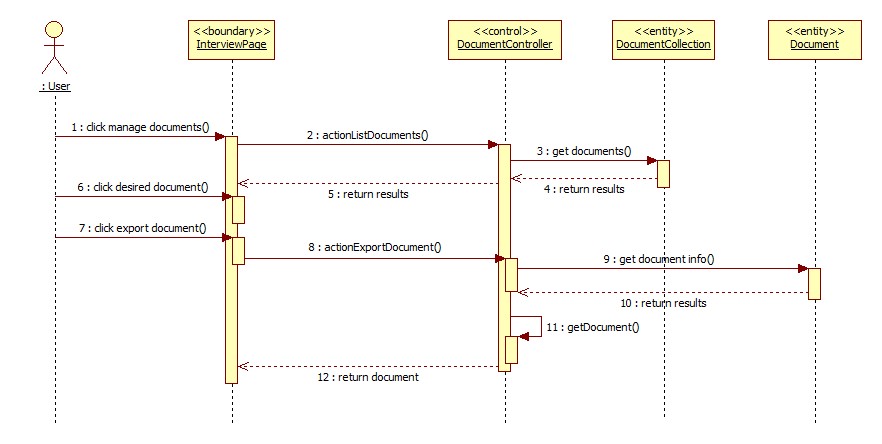
Login with FIU credentials



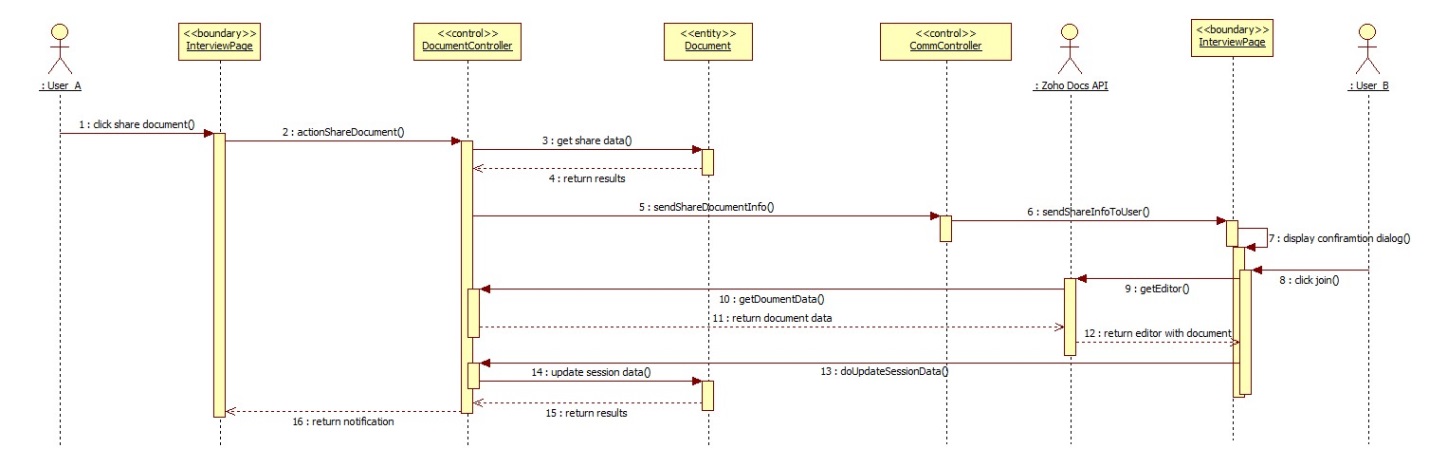
Rename Document



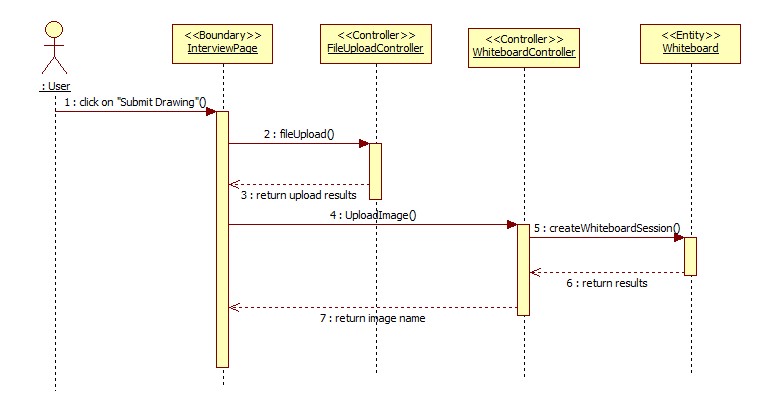
Export Document



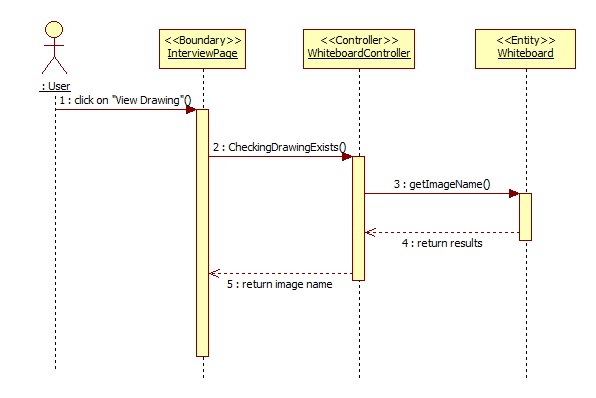
Share Document



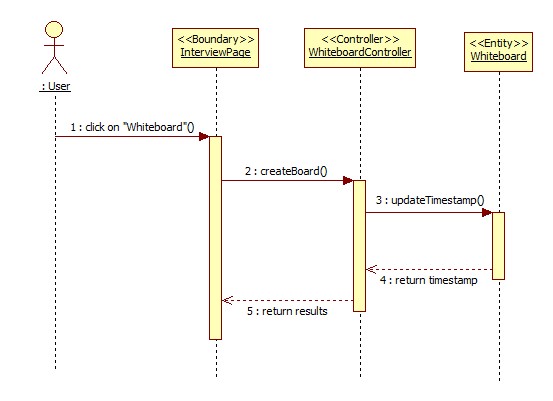
Upload Image



View Uploaded Image



Show or Restore Whiteboard



## Select Upload Image

## 

## New System’s State Diagrams

## Collaborative Editor

## F:\Luis B Final Revision UML\State Diagram.jpg

## SMS

## F:\Jorge F Final Revision UML\SMS state.jpg

## ScreenShare

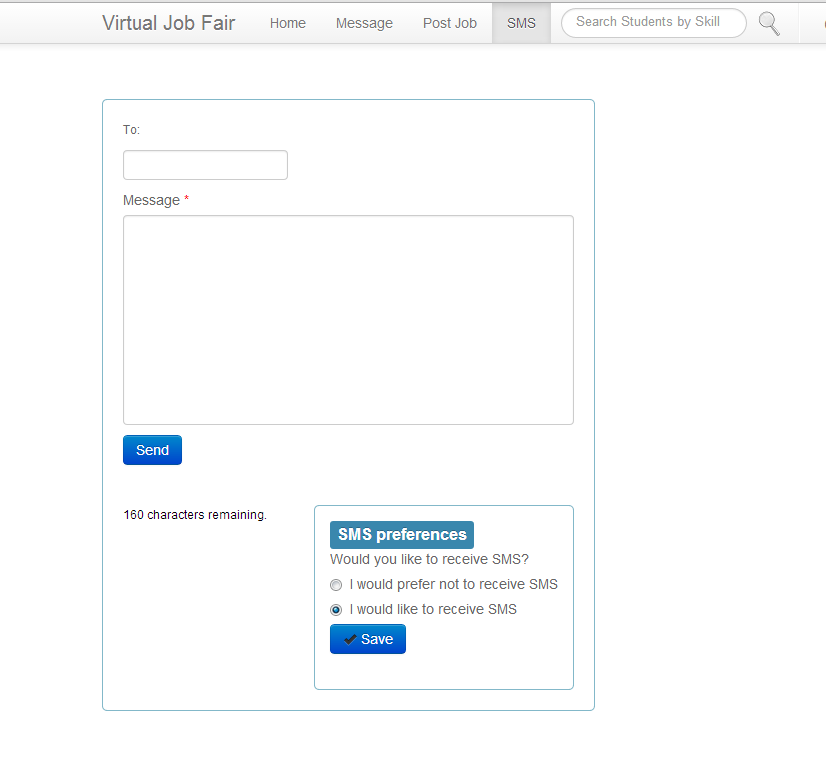
## F:\Jorge F Final Revision UML\ScreenShare state machine.jpg

## Whiteboard

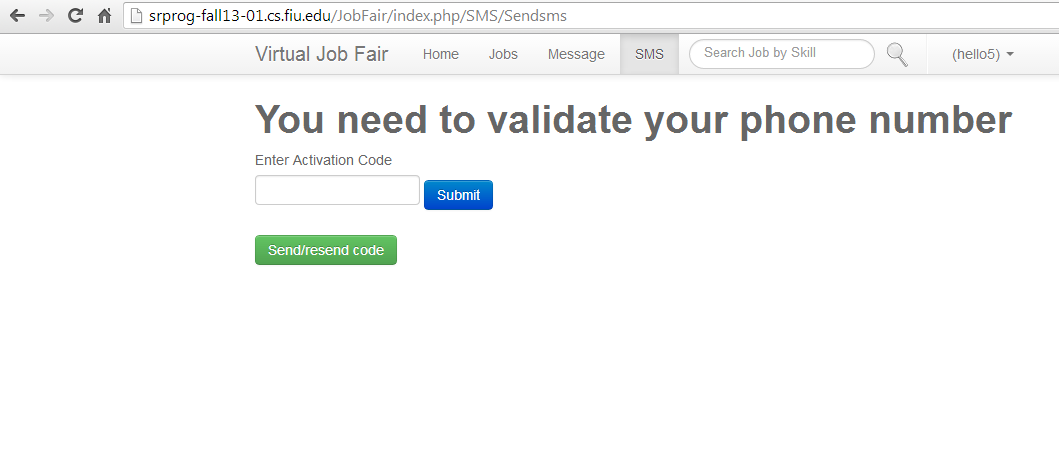
## https://lh6.googleusercontent.com/llelbiahyJkfBLElZoCEYyDwLFdh0MEenZv1DMvLGGUx36tH4Wuvf-BPOgrE-YF3HbbSHHCvyUo7dKi0AKQyJEGUMfruNNYInQ8BNyzGRWJk3WcTFIKZzR3AU_kWbbCbov8

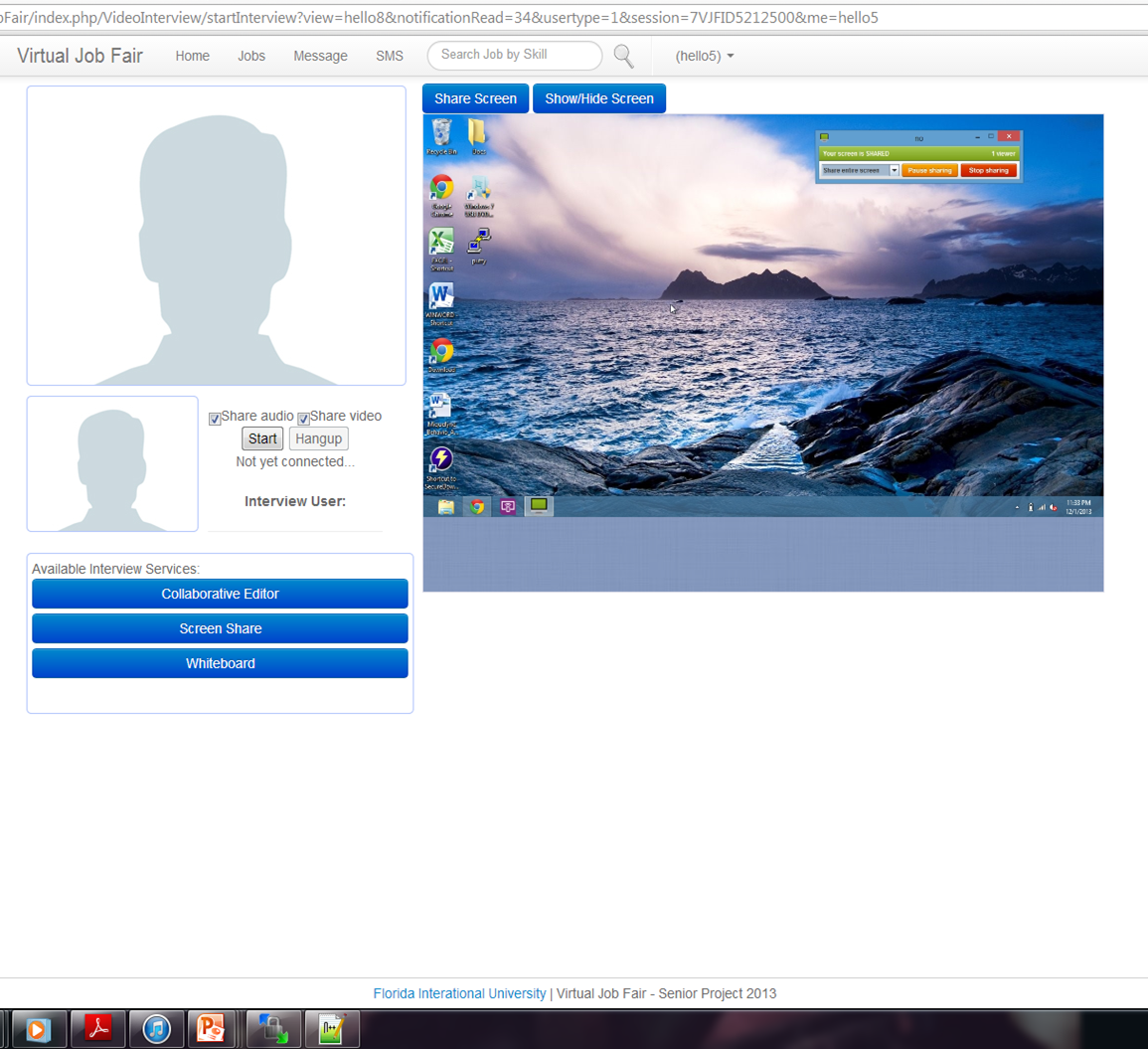
## 6.5 Appendix E - User Interface designs.

**Send SMS**

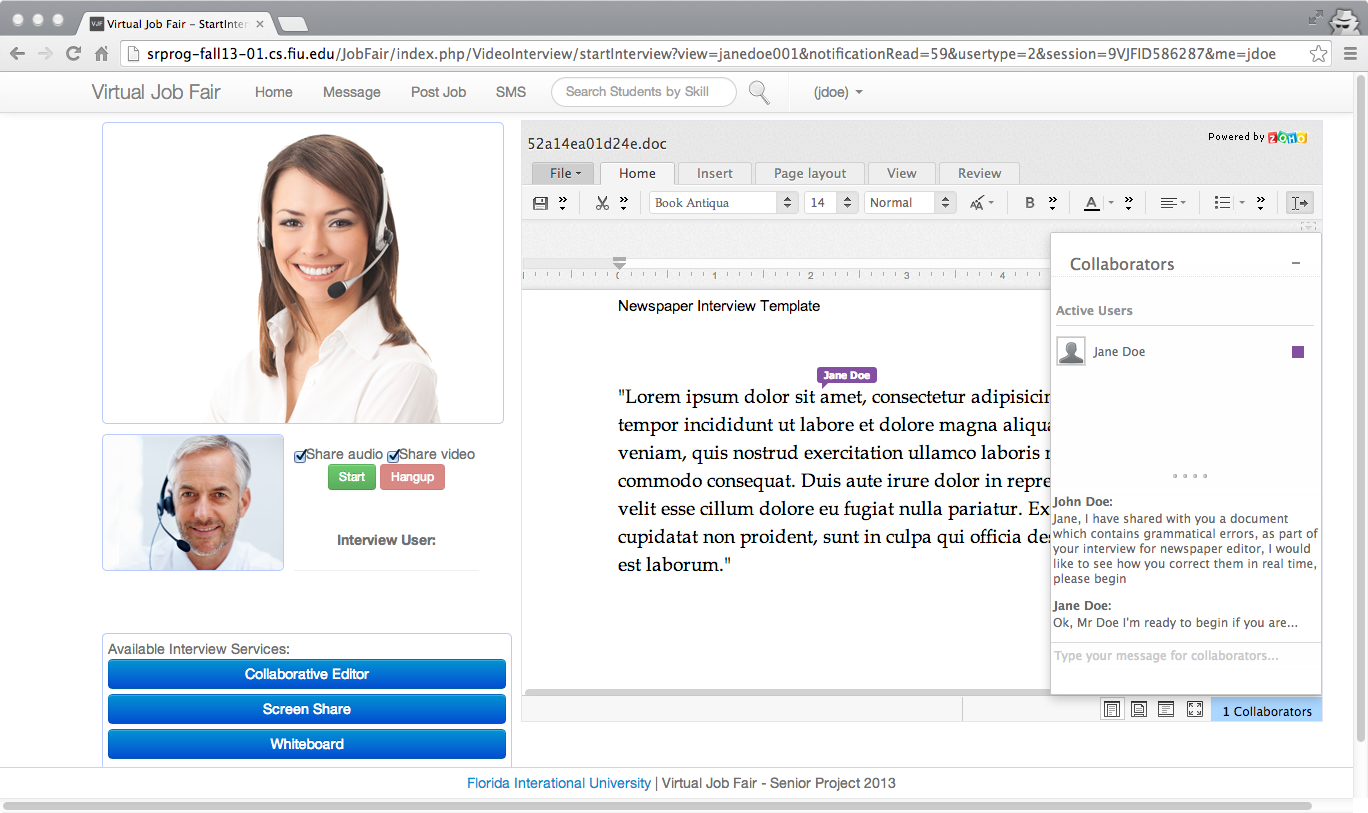
****

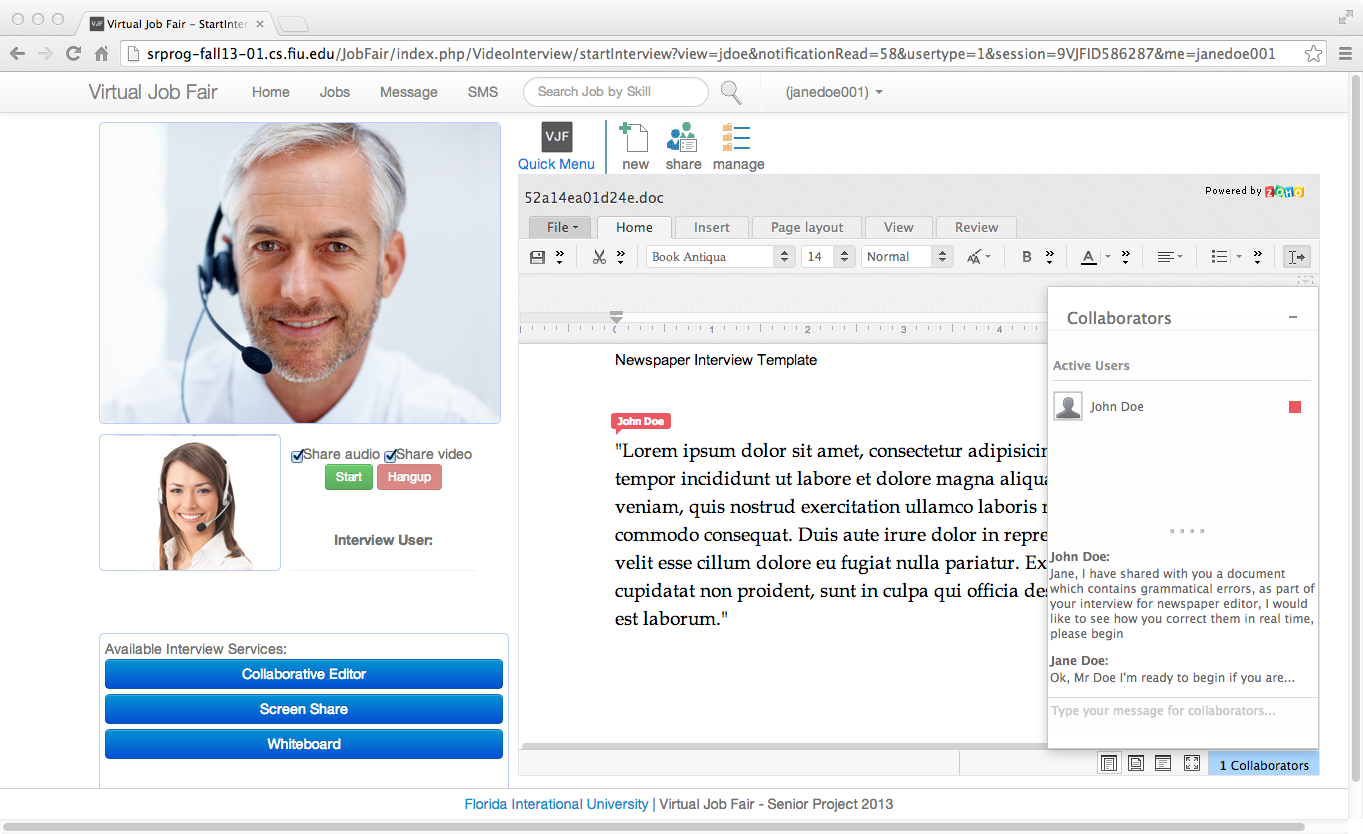
**Validate Phone Number**

****

**Screen Share**

**Collaborative Editor**

****



## Log In Using FIU SCIS Credentials

## F:\Luis B Final Revision UML\login screenshot.jpeg

## Whiteboard\\buffalo.cs.fiu.edu\homes\Desktop\CIS 4911\Deliverable #2 User Interface\Screenshot 1.png

## Image Sharing

## \\buffalo.cs.fiu.edu\homes\Desktop\CIS 4911\Deliverable #2 User Interface\Screenshot 2.png

## 6.6 Appendix F - 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 |

# 7. 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.