Requirements Document

CIS 4911 – Senior Project U01

Virtual Job Fair

**Members:**

Justin Korah

Andres Gonzalez

Tomer Doar

Diego Perez

Enmanuel Corvo

**Mentors:**

Masoud Sadjadi

Juan Caraballo

Copyright Page

*The contents of the Virtual Job Fair system and all associated documents belong to Florida International University and are only to be used for the purposes decided by FIU faculty. Redistribution or commercial use is prohibited without permission by the appropriate faculty in charge of the project and its contents.*

Abstract

*Virtual Job Fair seeks to aid the process of recruitment for SCIS students at Florida International University. The project attempts to connect students and companies in a more personal way than your typical job posting and application site. More particularly, the system attempts to serve as a tool for companies, especially for those who are unable to make campus visits to evaluate the talent at the university.*

*This document in particular contains the software requirements for the Virtual Job Fair application. It goes into the process how gathered the requirements. Furthermore, the requirements analyzed and specified to aid in the design process for the system. UML model is used extensively in the analysis process.*

Contents

[1. Introduction 5](#_Toc354467210)

[1.1 Problem Definition 5](#_Toc354467211)

[1.2 Scope of system 5](#_Toc354467212)

[1.3 Terminology 6](#_Toc354467213)

[1.4 Overview of document 6](#_Toc354467214)

[2. Current System 6](#_Toc354467215)

[3. Project Plan 9](#_Toc354467216)

[3.1 Project organization 10](#_Toc354467217)

[3.2 Work breakdown 10](#_Toc354467218)

[3.3 Cost Estimate 11](#_Toc354467219)

[4. Proposed System Requirements 11](#_Toc354467220)

[4.1 Functional and Non-functional Requirements 11](#_Toc354467221)

[4.2 Analysis of System Requirements 16](#_Toc354467222)

[4.2.1 Use case model 16](#_Toc354467223)

[4.2.2 Static model 16](#_Toc354467224)

[4.2.3 Dynamic model 16](#_Toc354467225)

[5. Glossary 17](#_Toc354467226)

[6. Appendix 18](#_Toc354467227)

[6.1 Appendix A - Complete use cases 18](#_Toc354467228)

[6.2 Appendix B - Use case diagram using UML 40](#_Toc354467229)

[6.3 Appendix C - Static UML diagram 41](#_Toc354467230)

[6.4 Appendix D - Dynamic UML diagrams 44](#_Toc354467231)

[6.5 Appendix E - User Interface designs. 67](#_Toc354467232)

[7. References 71](#_Toc354467233)

# 1. Introduction

The introduction gives a brief overview of the proposed system. It allows the reader to quickly get an idea of why the system is being built, and a high level description of what the system is expected to do. The introduction will also provide context to the rest of the context with a terminology section, which can be referenced to from other sections. Lastly, it gives a summary of what to expect throughout the rest of the document.

## 1.1 Problem Definition

Recruiters everywhere are always looking for talent at the college level to fill in internship and entry level roles. Currently the most effective way for employers to recruit at schools is to visit them and set up presentations or attend career fairs at the school. Since not all employers have the resources to actively visit schools searching for talent, universities have provided career sites that enable employers to post jobs along with contact info, so that students can apply. This solution, however, is not sufficient and is not all that much different than online job boards. It is not nearly as effective as a campus visit where companies can connect with students on a face to face basis, which is what one of the aspects the Virtual Job Fair will try to emulate.

## 1.2 Scope of system

The system will allow for employer and student registration. Employers will be able to register and give information about themselves as well as the company, which will be shown in a public profile available for student viewing. Likewise, students will create profiles which contain education information, experience information, and information about his or her skillset. These student profiles are viewable by employers and employers will have the functionality to search for students with certain criteria.

Additionally, employers are able to post job and internship openings which can be viewed and searched for by students. Students can then apply to these jobs and provide a cover letter for each application. Employers can view who applied to their postings and view their profile, where they could also download their resume. Furthermore, employers are able to schedule video interviews with students, which will take place on the system.

The system contains other functionality to make recruiting easier on the employers and the students. These features include, but are not limited to: a messaging system, a notification system, video resumes, and job to student skill matching

## 1.3 Terminology

|  |  |
| --- | --- |
| Term | Meaning |
| API | Application Programmer Interface |
| VJF | Virtual Job Fair |
| UML | Unified Modeling Language |
| SCIS | School of Computer and Information Sciences |

## 1.4 Overview of document

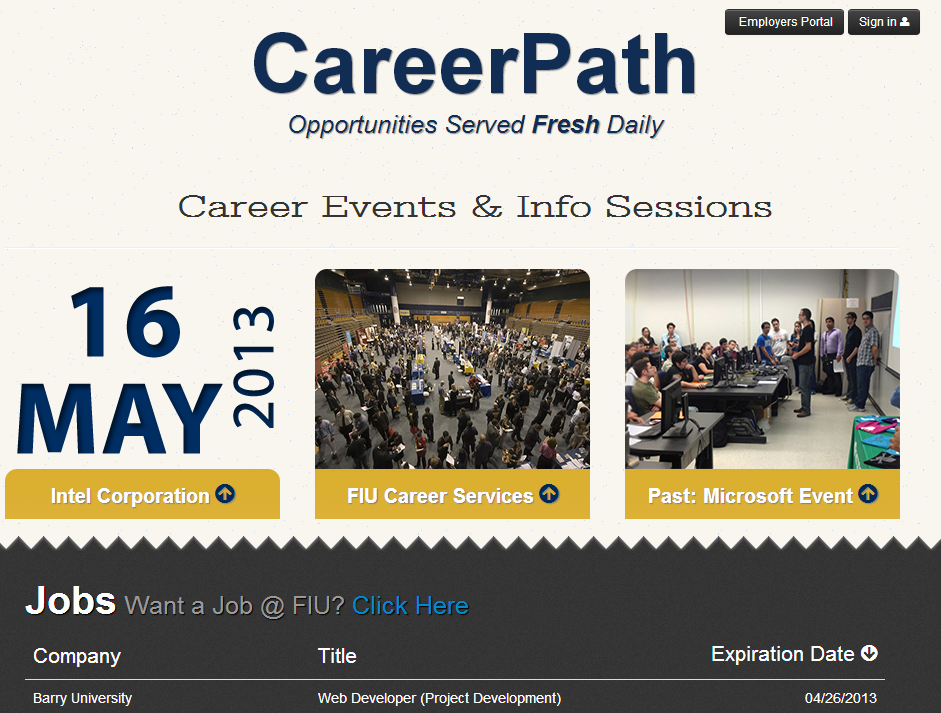
This is an overview of the content of this document, briefly describing what is covered in each section.

Section 2 covers the current systems available as a solution to the problem we are trying to solve. It goes into details to explain why the current systems do not efficiently solve the problem and what they lack. Section 3 covers the project plan; this will include the project organization and the work breakdown between team members and their individual contribution. Section 4 covers the system requirements, which include functional requirements and analysis of the system requirements. This section will include diagrams covering the implementation requirements of the system, such as Use Case Model, Static Models, and Dynamic Models. Section 5 is a glossary where definition to specific terms used can be found. Section 6 consists of appendixes which contain mostly diagrams. Finally, section 7 contains the references used in this document.

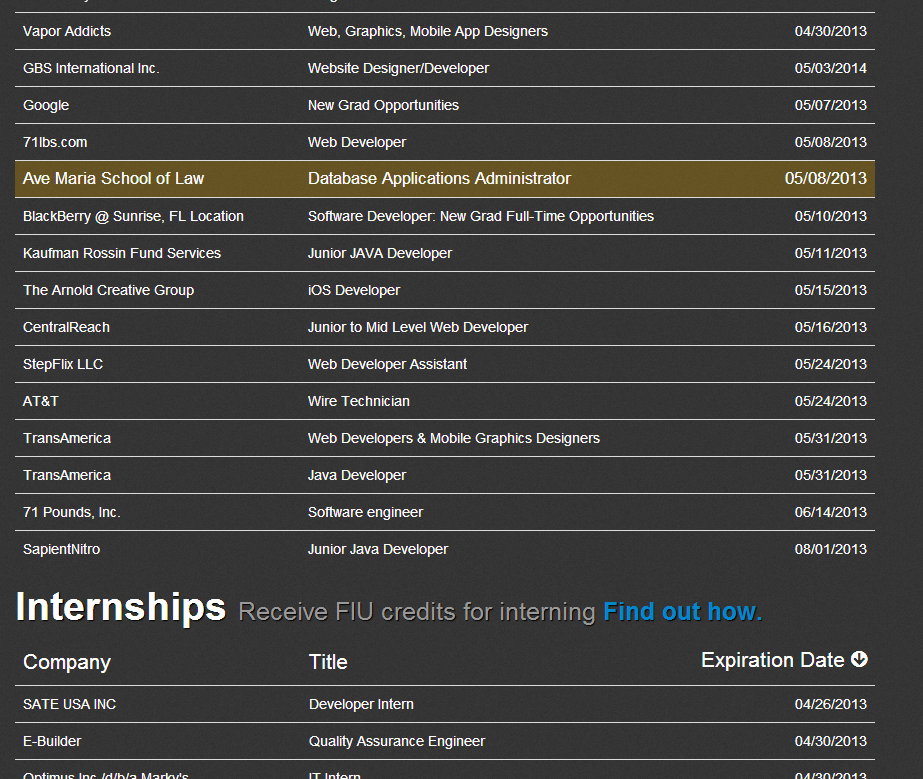
# 2. Current System

The SCIS department currently has a web page set up with listings of internships and jobs. This current setup is known as CareerPath and is located at [www.cis.fiu.edu/careerpath](http://www.cis.fiu.edu/careerpath). In its current implementation, the system allows companies to post jobs and internships. These postings include information about the company, a description for the open position, the expected duties, and the qualifications needed to perform in the position. It also provides an email address to contact as well as a website to refer to.

The system is fairly limited because of the fact that it does not contain enough resources for students and employers. Currently, the employers can post jobs and only wait for qualified people to apply. Modern job sites allow employers to actively seek out potential talent, which is one of the things the proposed system strives to accomplish.



**Home page for CareerPath**



**Job list in CareerPath**



**Job description in CareerPath**

# 3. Project Plan

The project plan illustrates how effectively the project was carried out during this deliverable. It gives a quick rundown of where the bulk of the time was being spent in the form of a Gantt chart. Additionally, it assigns responsibility of the work to each respective individual working on the project. The project plan (mostly the work breakdown) is important to refer to when analyzing the project and the project schedule as a whole.

## 3.1 Project organization

The completion of this project would be possible due to the cooperation of all team members who will take on different roles throughout the implementation phase.

Enmanuel Corvo’s contribution to the project will be the implementation of the video interview functionality, which allows students and employers to virtually connect and have an interview.

Justin Korah’s contribution to the project will be the implementation of a matching algorithm which will help narrow down the search for employers. With the student job match, employers and students will benefit because it will automatically match appropriate candidates for a job post. This will enable the employers to dedicate more time to the suitable candidates instead of having to search students one by one.

Tomer Doar’s contribution to the project will be the implementation of a notification system which would keep everyone in the loop. Whenever an employer interacts with a student the notification system will be responsible for keeping track of this. Anything from scheduling a video interview to posting a new job will be filtered by the notification system, which will be responsible for alerting the users of recent interactions.

Diego Perez’s contribution to the project will be to ensure that the system is easy to access. This includes allowing users to register using third party websites such as LinkedIn and Google to mention some. In addition, Diego would work with LinkedIn API to allow students to import their profile information from LinkedIn into their Virtual Job Fair account. Diego will also work on allowing students to upload a video resume into their profile, an alternative to traditional paper resume.

Andres Gonzalez’s contribution to the project will be the implementation of an internal messaging system. This will allow employers to easily communicate with the students they choose and keep in touch with them. Only an employer would be able to initiate that line of communication.

## 3.2 Work breakdown

Appendix A contains a project schedule. The work for this deliverable, which focuses on the requirements, can be broken down into the following work breakdown structure:

## 3.3 Cost Estimate

Given the open sourced nature of the technologies that will be used, there will be no costs associated with building our project other than the time factor.

# 4. Proposed System Requirements

The following section provides information about the functional and non-functional requirements of the proposed system. These requirements were gathered and analyzed to come up with the non-functional demands with regards to: usability, reliability, performance, and supportability. This section only contains high-level information about the functional requirements. More information about the requirements can be found in Appendix A in the form of uses cases. The section also contains analysis of the requirements to guide the designers and programmers with their design efforts. This document can therefore be consulted during the design process.

## 4.1 Functional and Non-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.

## 4.2 Analysis of System Requirements

This section gives a brief overview of each of the requirement artifacts presented in Appendix B-D. It goes into a bit more detail regarding the requirements of the system, presenting the team of developers with the specific tasks, artifacts, and interactions that will need to be implemented to complete the system.

### 4.2.1 Use case model

REFER TO APPENDIX B

The Use Case diagram in appendix B gives a high-level overview of all the functional requirements of the system. It identifies the three types of users that our system interacts with: students, employers, and administrator. For each type of user, the model identifies the tasks that the user should be able to carry out upon completion of the system.

The tasks on the left-hand side encompass everything a student should be able to accomplish. Most of these tasks have to do with the creation and management of a student profile, while the last task listed represents the ability of students to apply for jobs.

The tasks on the right-hand side represent what the employers and administrator should be able to do upon completion of the system, while the tasks shown in the center represent functionality common to all types of users.

### 4.2.2 Static model

REFER TO APPENDIX C

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

### 4.2.3 Dynamic model

REFER TO APPENDIX D

The Dynamic model in appendix D contains the sequence diagrams extracted from analyzing the specific requirements of each of the Use Cases presented in Appendix A. It gives the team of developers a more specific view of the interaction between the user and the system for each specific piece of functionality from the Use Case Model.

By referring to this diagram, developers can see the main artifacts involved in the flow of information for each use case, and how they are associated with each other: boundary objects as the intermediaries between user requests and controllers, controllers accessing model objects and passing along the returned data to the boundary objects, etc.

Conventional UML notation was used to produce the sequence diagrams.

# 5. Glossary

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

# 6. Appendix

## 6.1 Appendix A - Complete use cases

Instructions: Copy blank table below last filled table.  For  Preconditions, Steps, and Postconditions, click on the empty row and press(Ctrl+Shift+7) to start numbered lists

|  |  |
| --- | --- |
| 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 | 1. User clicks My Profile 2. User is redirected to his profile page 3. User clicks on the edit picture buttom 4. User pick his picture and then click save. |
| Post-conditions | user post his picture |
| Exceptions | None |

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

|  |  |
| --- | --- |
| Use Case ID | VJF-0031 Edit Company Info |
| Description | Employer changes his company Information |
| Actor | Employer |
| Pre-conditions | 1. Employer is logged in 2. Employer is viewing his profile |
| Steps | 1. Employer clicks on edit image in company info section 2. Employer changes text in any of the text boxes 3. Employer clicks on checkmark |
| Post-conditions | 1. New company info is saved 2. Employer is redirected back to profile page |
| Exceptions | Employer fills in invalid values for the fields |

|  |  |
| --- | --- |
| Use Case ID | VJF-0032 Search Jobs |
| Description | Student searches for jobs by skill |
| Actor | Student |
| Pre-conditions | 1. Student is logged in 2. Student is on home page |
| Steps | 1. Student clicks on job search text box 2. Student begins typing a skill 3. Student either selects skill from auto complete or fully types out skill 4. Student clicks submit |
| Post-conditions | 1. Student is redirected to search result page with relevant jobs, and option to search more jobs |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0033 Search Students |
| Description | Student searches for students by skill |
| Actor | Employer |
| Pre-conditions | 1. Employer is loggedin 2. Employer is on home page |
| Steps | 1. Employer clicks on job search text box 2. Employer begins typing a skill 3. Employer either selects skill from auto complete or fully types out skill 4. Employer clicks submit |
| Post-conditions | Employer is redirected to search result page with relevant students, and option to search more students |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | VJF-0034 Close Job |
| Description | Close a job from further applications |
| Actor | Employer |
| Pre-conditions | 1. Employer is logged in |
| Steps | 1. Employer views his own profile 2. Employer selects a job from one of his own postings 3. Employer clicks on “Close Job” |
| Post-conditions | 1. User is redirected back to the job page 2. The job is closed |
| Exceptions | None |

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

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

|  |  |
| --- | --- |
| Use Case ID | VJF-0037 Apply to Job |
| Description | Student applies to an 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 |

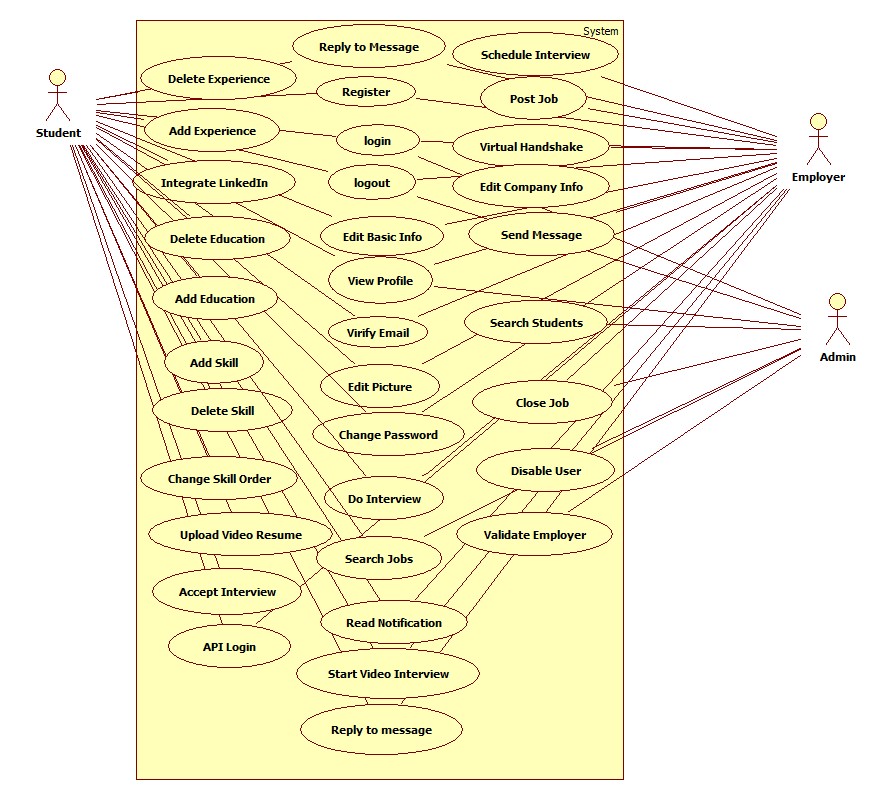
|  |  |
| --- | --- |
| Use Case ID | VJF-0038 Read notification |
| Description | User read notification from his/home page |
| Actor | All user types |
| Pre-conditions | 1. User is logged in 2. User is on home page |
| Steps | 1. User is clicking on the notification section that he or she will like 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 |

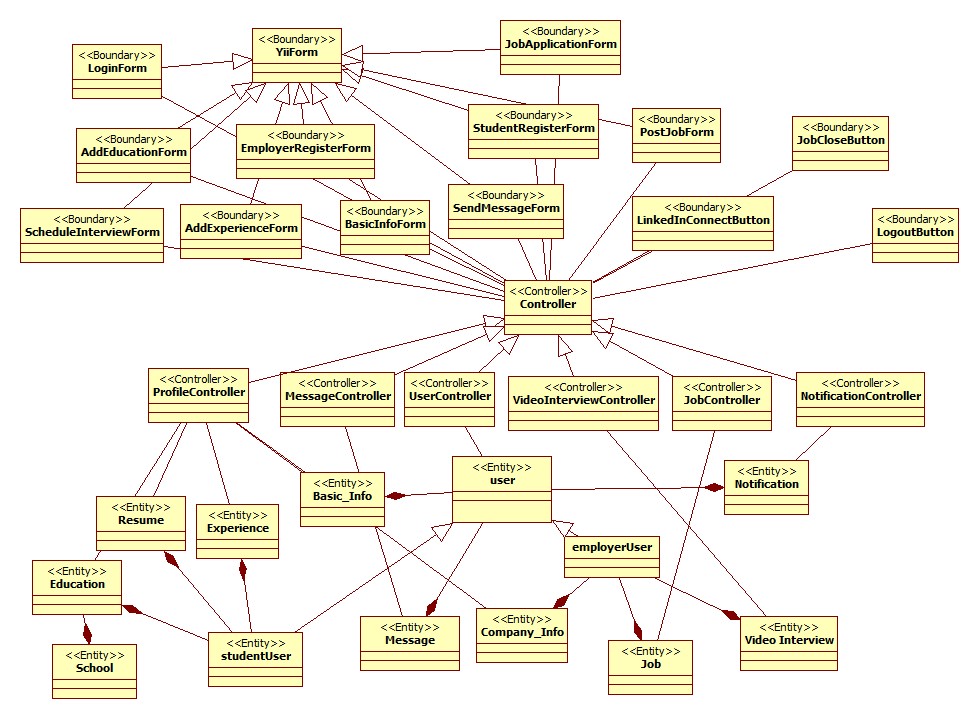
|  |  |
| --- | --- |
| Use Case ID | VJF-0040 Validate an Employer Register |
| Description | Admin validate a new employer that register |
| Actor | Admin |
| Pre-conditions | 1. Admin is logged in. 2. Admin is on his home page. |
| Steps | Admin is clicking on the notification like that validate the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

## 6.2 Appendix B - Use case diagram using UML

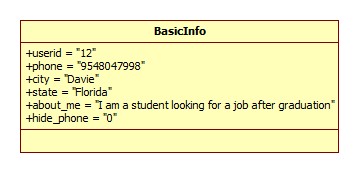


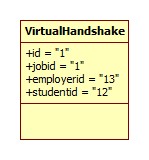
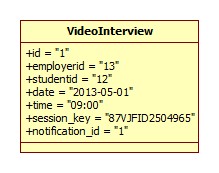
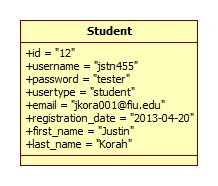
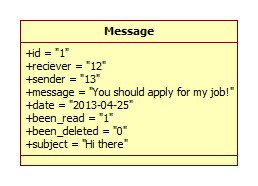
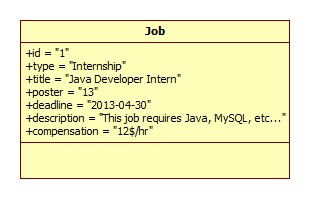
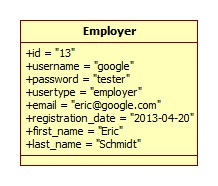
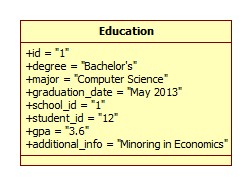
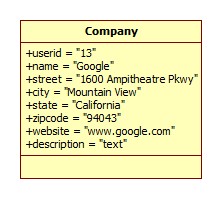
## 6.3 Appendix C - Static UML diagram



Class diagram of the VJF System

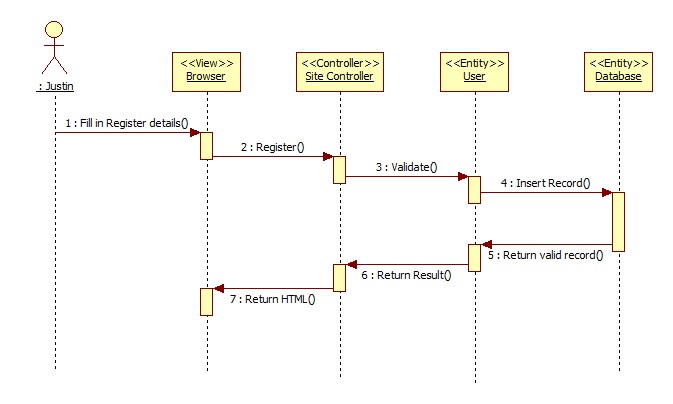
Below are object diagrams for entities found in the Virtual Job Fair system



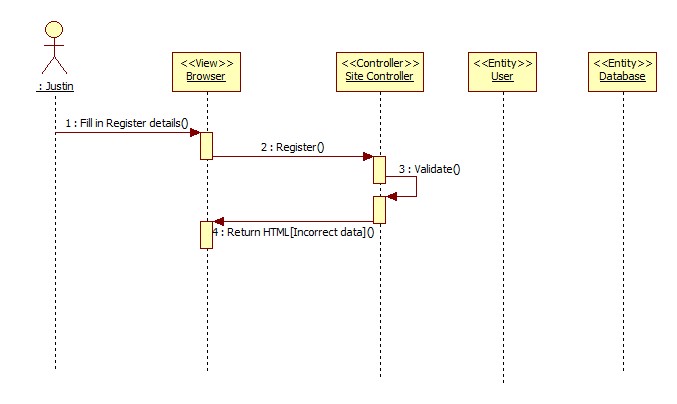


## 6.4 Appendix D - Dynamic UML diagrams

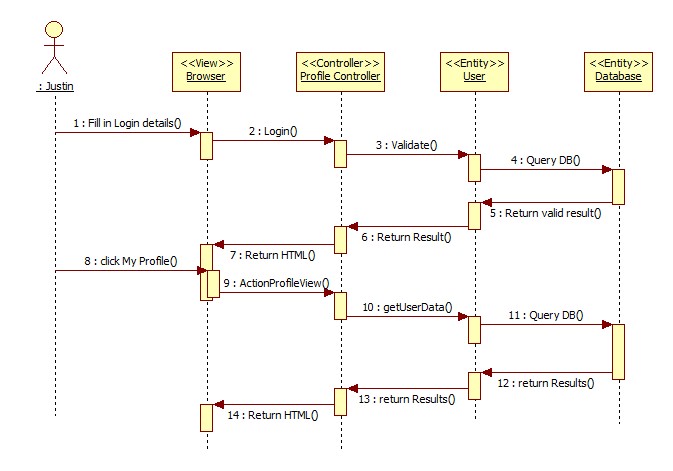
VJB-001 Registration (Sunny Day)

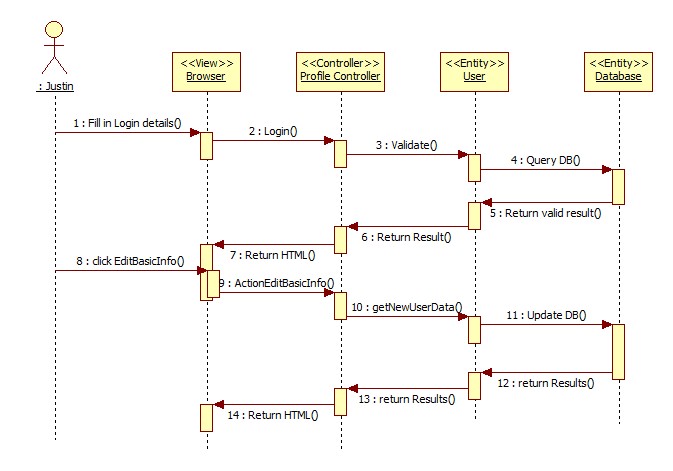


VJB-001 Registration (Rainy Day)

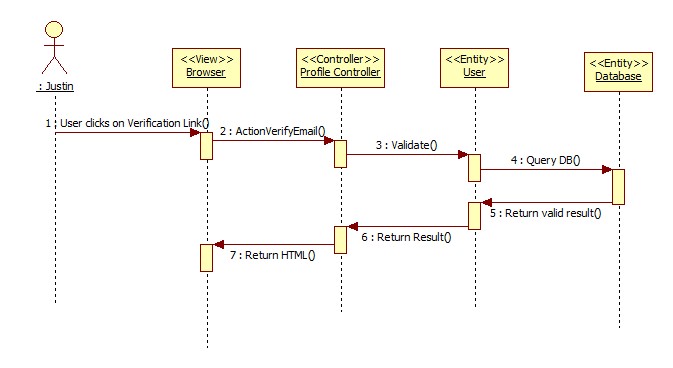


VJB-002 View Profile (Sunny Day)

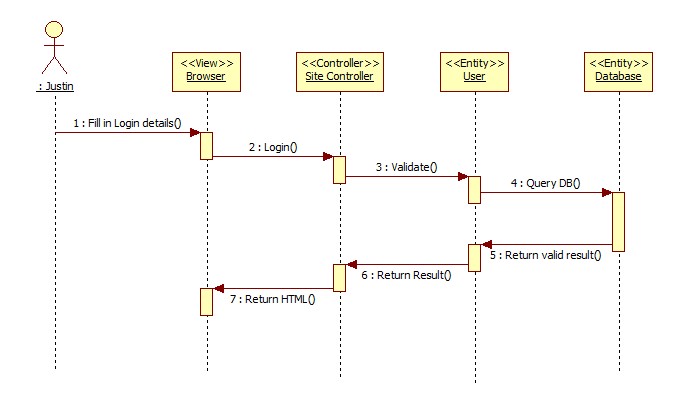


VJB-003 Edit Basic Information (Sunny Day)

VJB-004 Verify Email (Sunny Day)



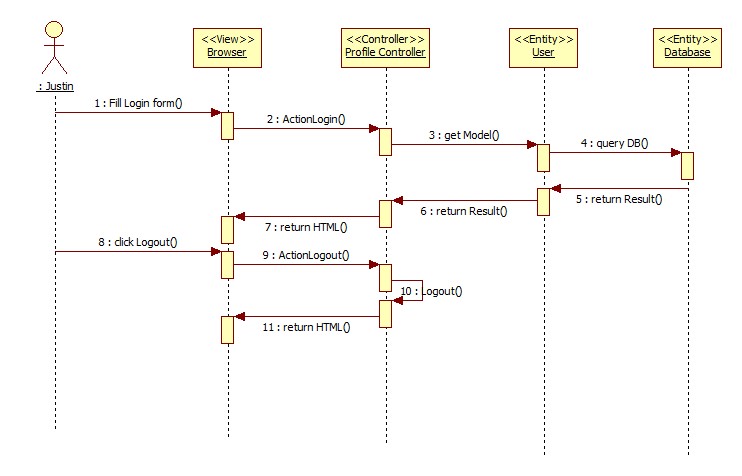
VJB-005 Login (Sunny Day)



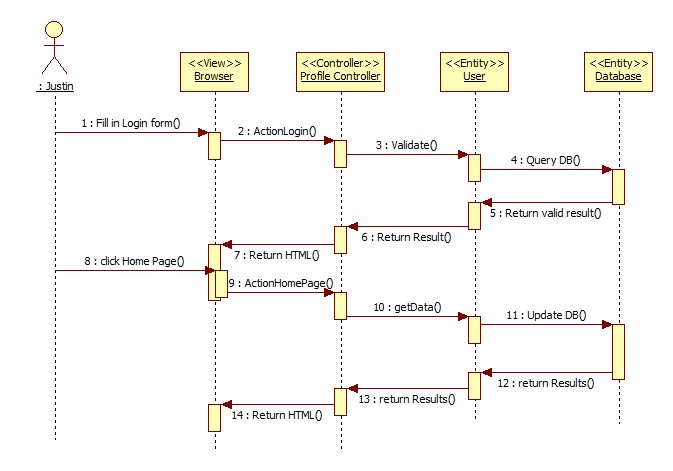
VJB-005 Login (Rainy Day)



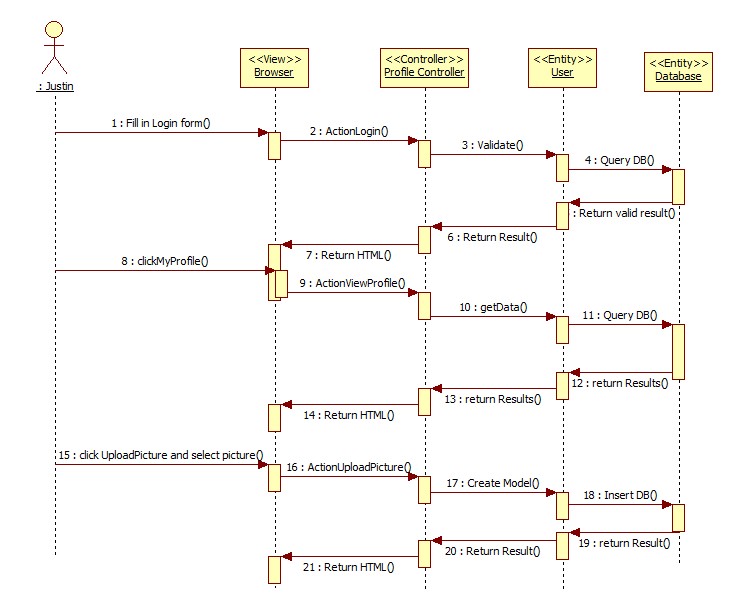
VJB-006 Logout (Sunny Day)



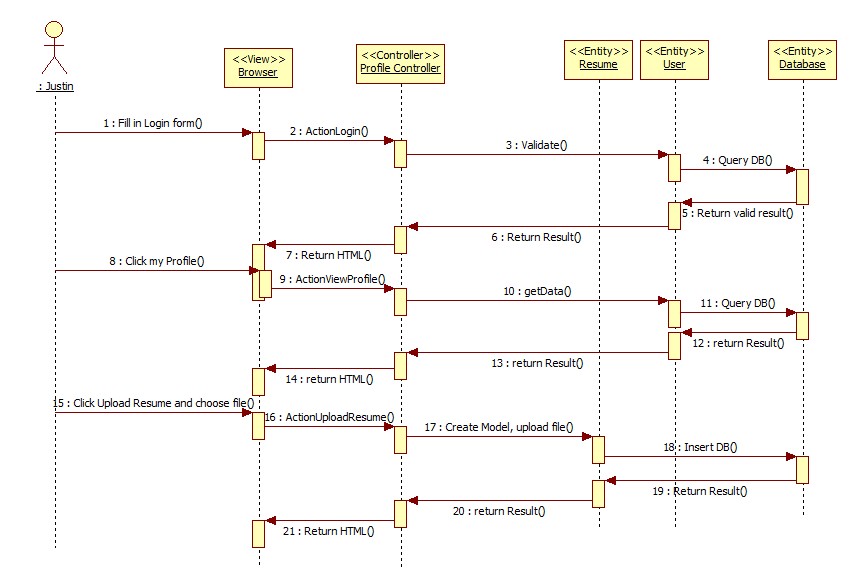
VJB-007 Home Page (Sunny Day)

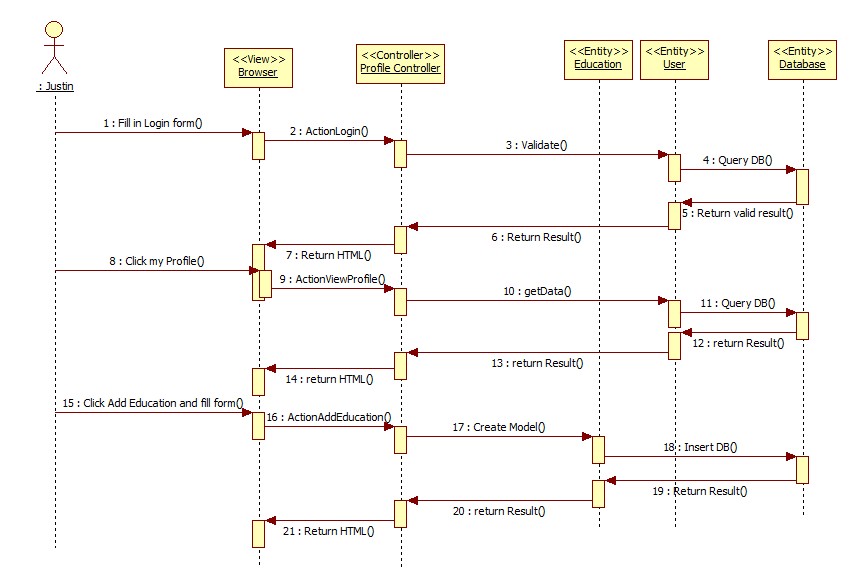


VJB-008 Upload Picture (Sunny Day)

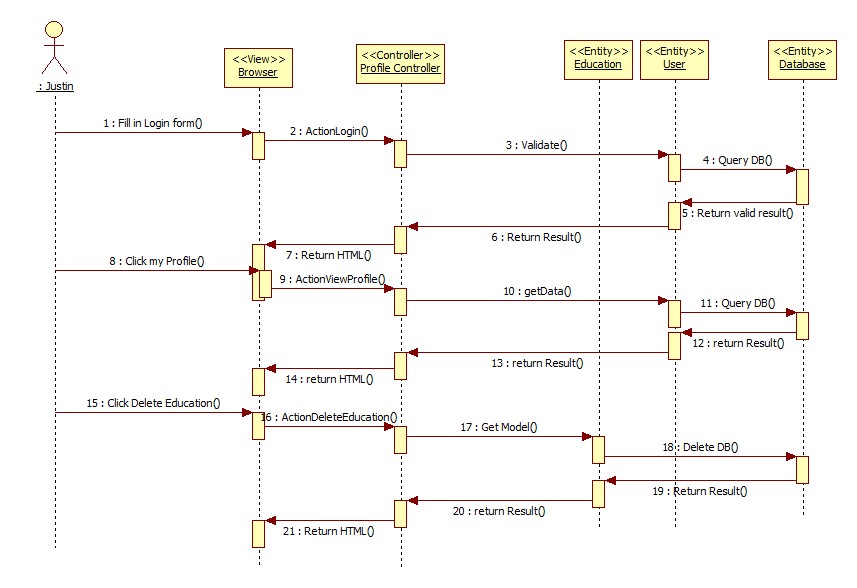


VJB-009 Upload Resume (Sunny Day)

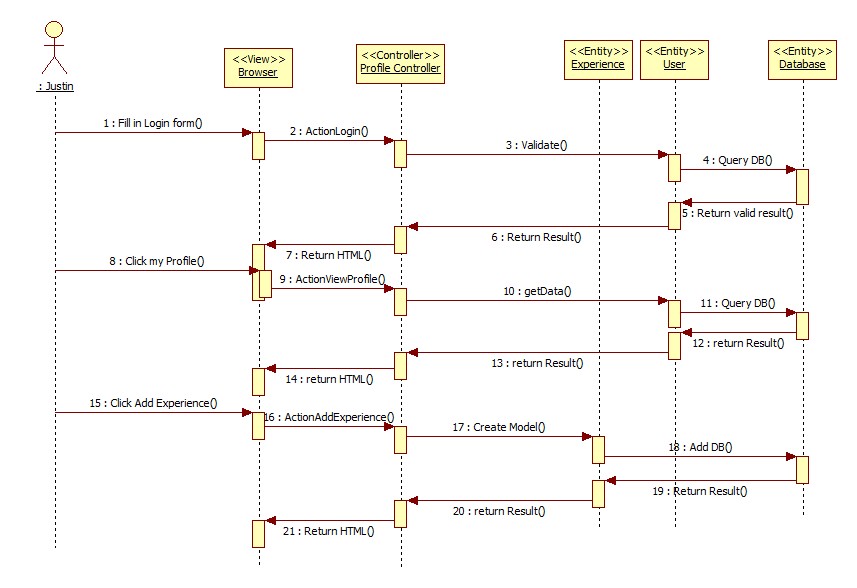


VJB-011 Add Education (Sunny Day)

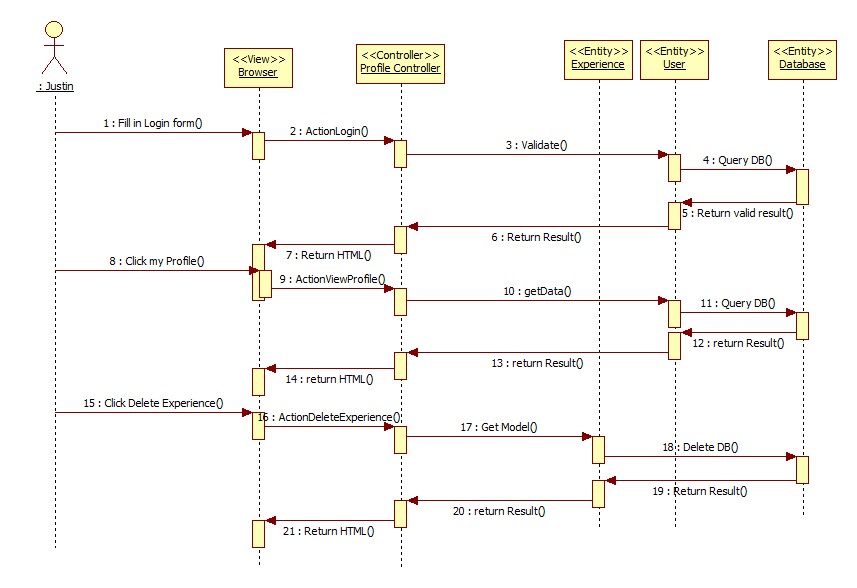
VJB-012 Delete Education (Sunny Day)



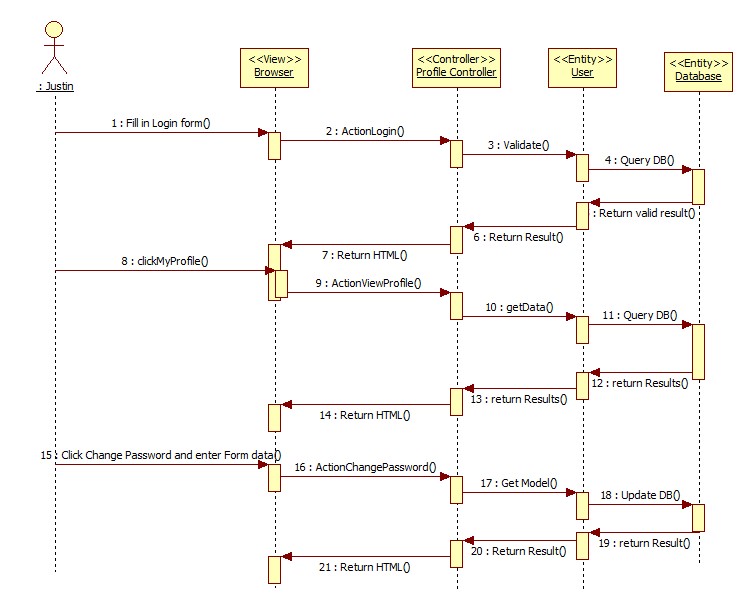
VJB-013 Add Experience (Sunny Day)



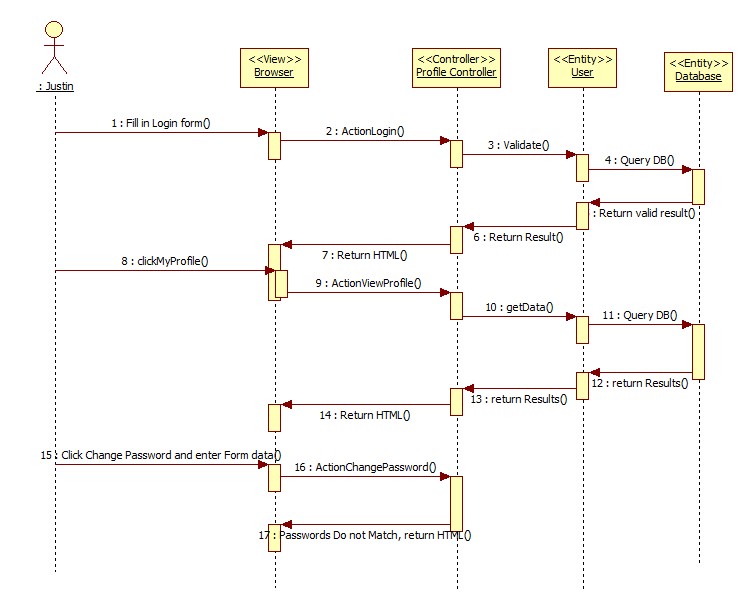
VJB-014 Delete Experience (Sunny Day)



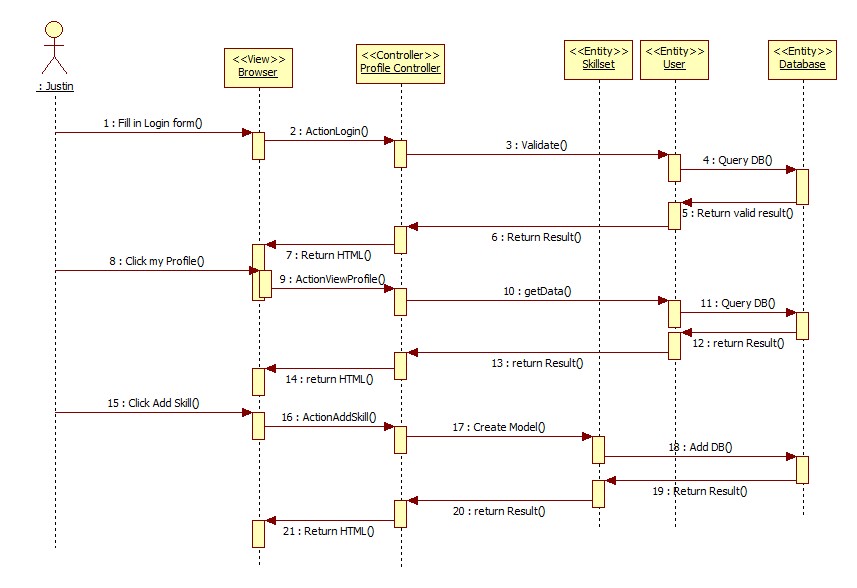
VJB-015 Change Password (Sunny Day)



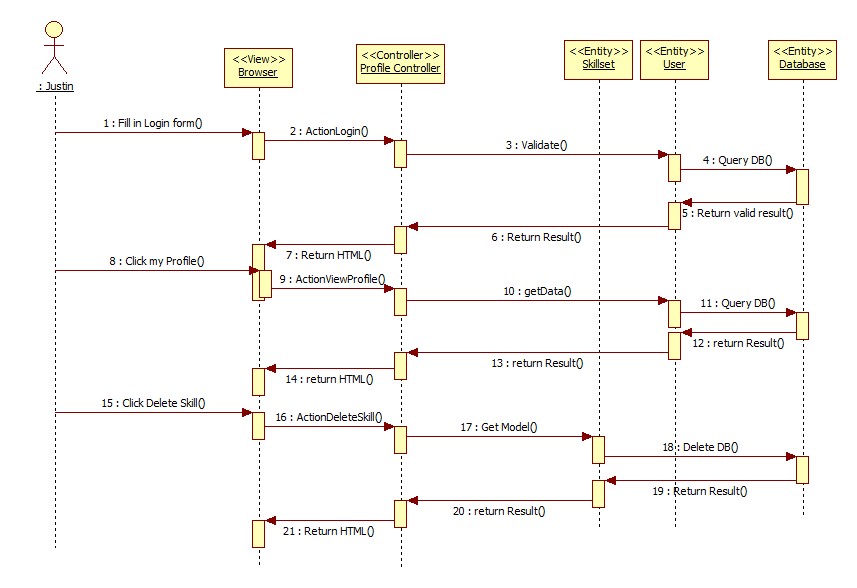
VJB-015 Change Password (Rainy Day)



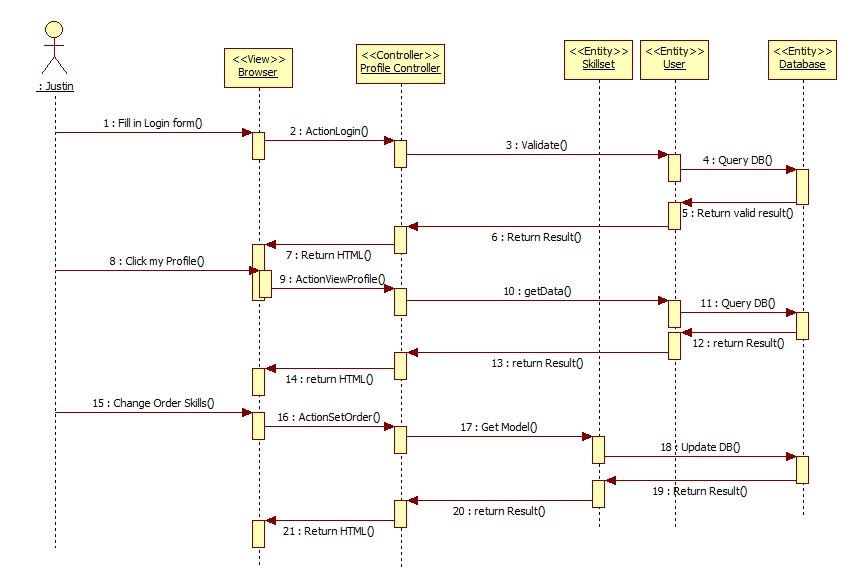
VJB-016 Add Skill (Sunny Day)



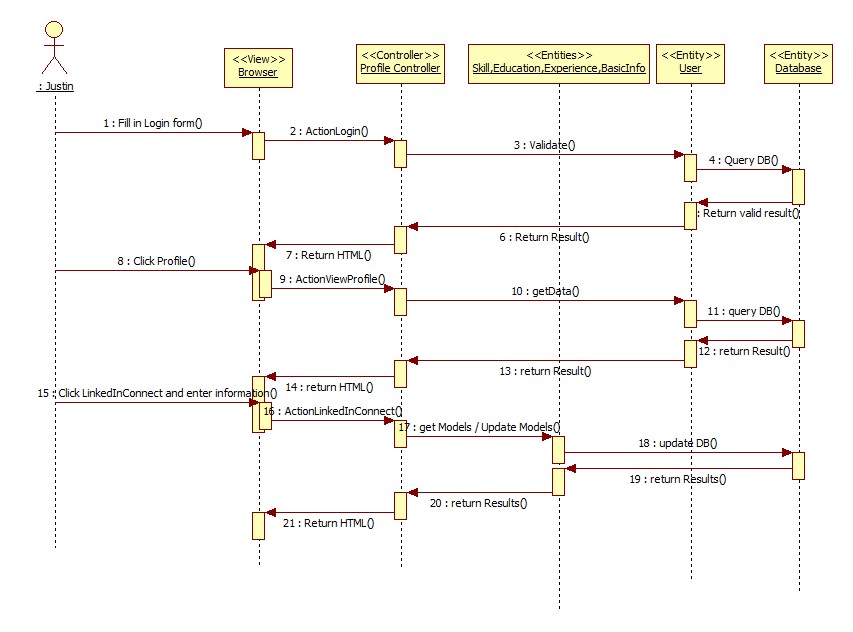
VJB-017 Delete Skill (Sunny Day)



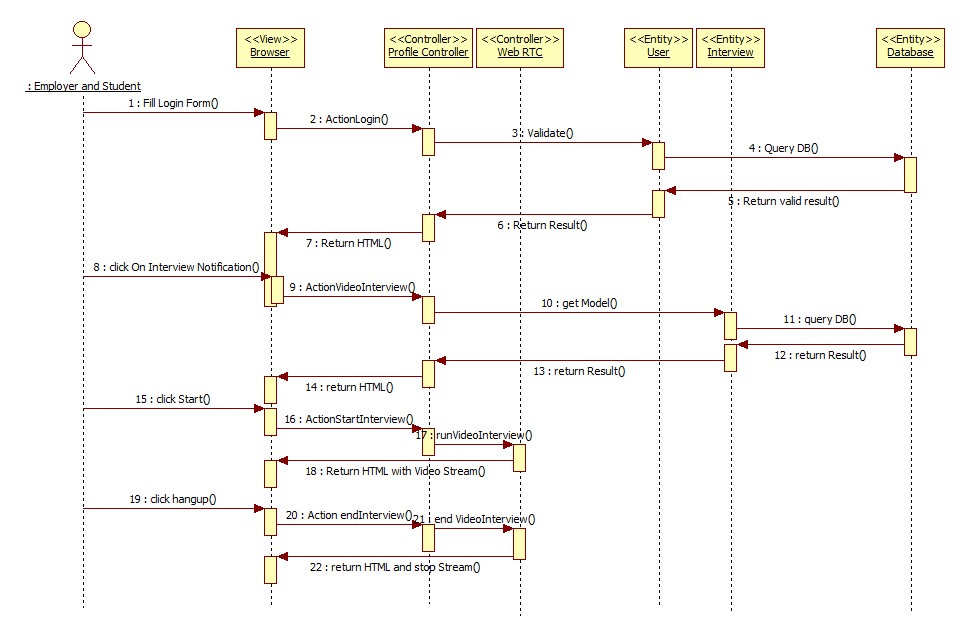
VJB-018 Change Skill Order (Sunny Day)



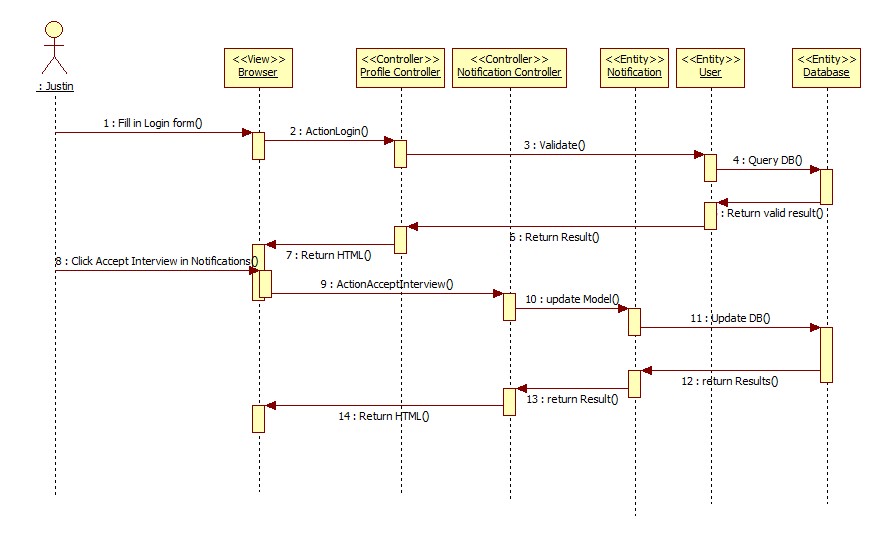
VJB-019 Integrate LinkedIn (Sunny Day)



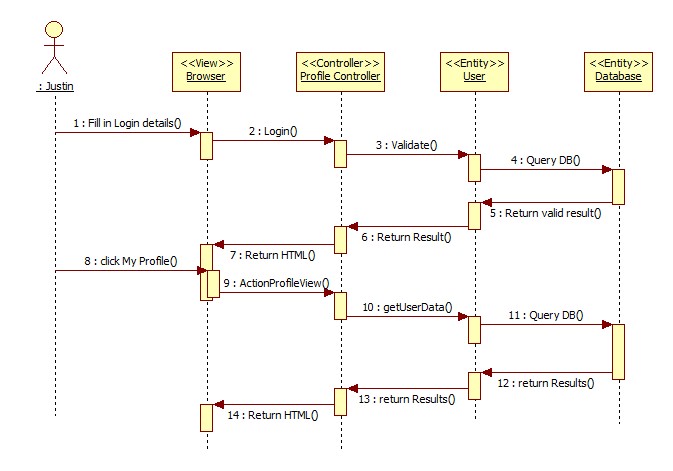
VJB-020 Video Interview (Sunny Day)



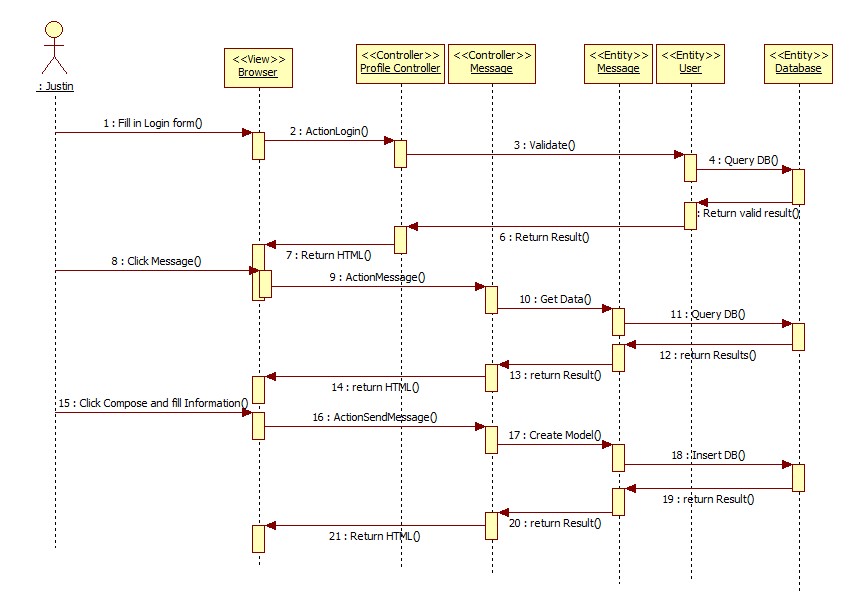
VJB-021 Accept Interview (Sunny Day)



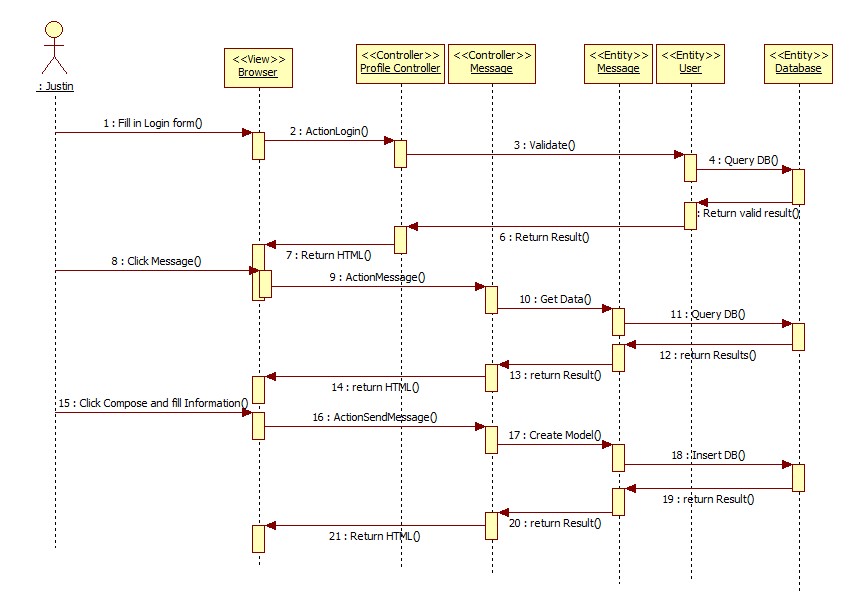
VJB-022 View Student Profile (Sunny Day)



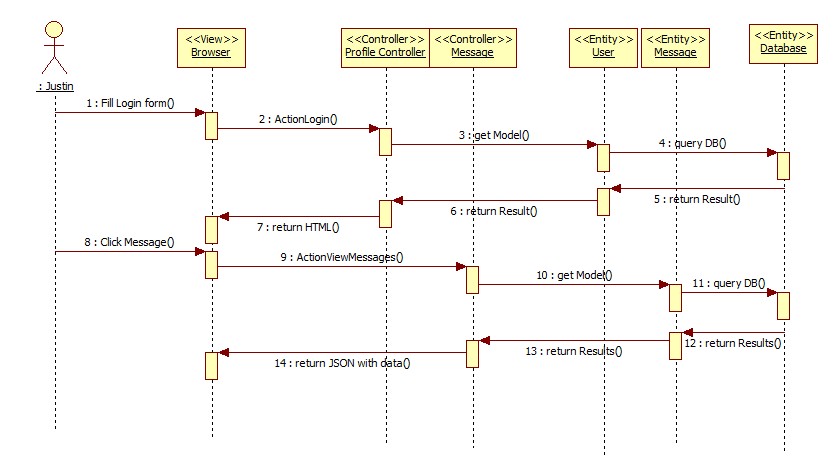
VJB-023 Send Message (Sunny Day)



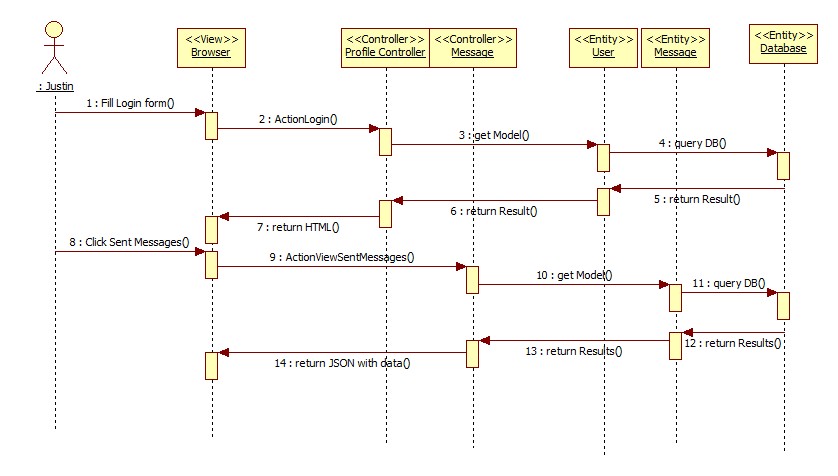
VJB-024 Reply To Message (Sunny Day)



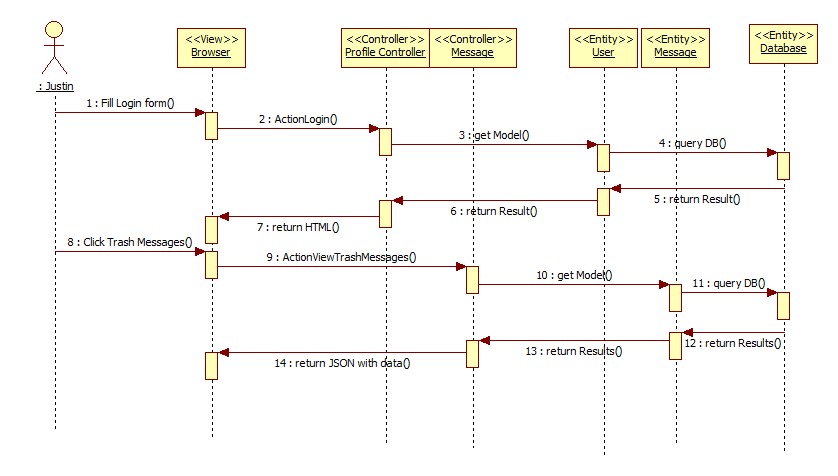
VJB-025 Get Inbox (Sunny Day)



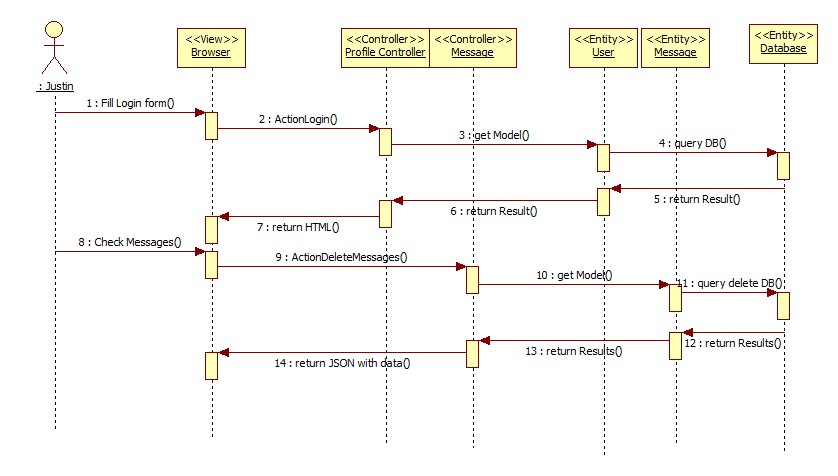
VJB-026 Get Sent Messages (Sunny Day)



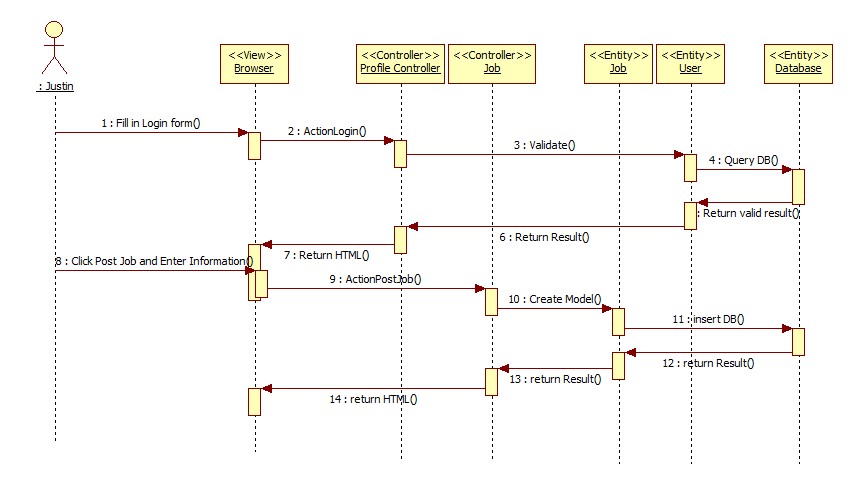
VJB-027 Get Trash Messages (Sunny Day)



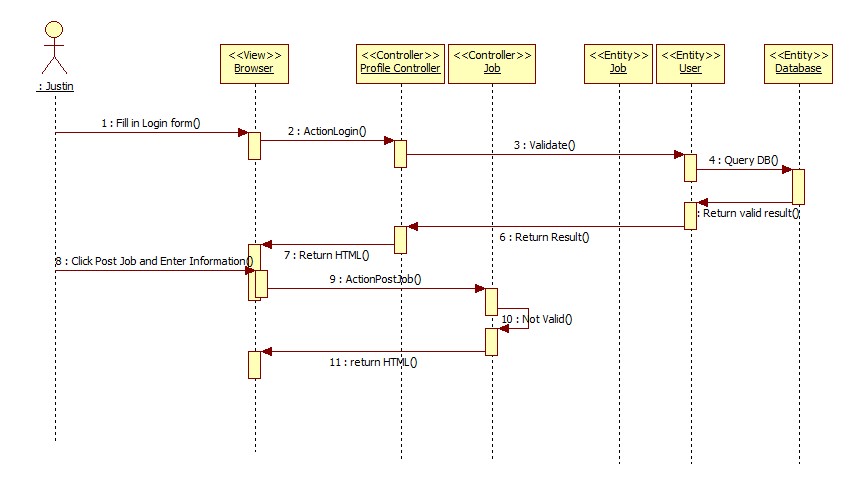
VJB-028 Delete Messages (Sunny Day)



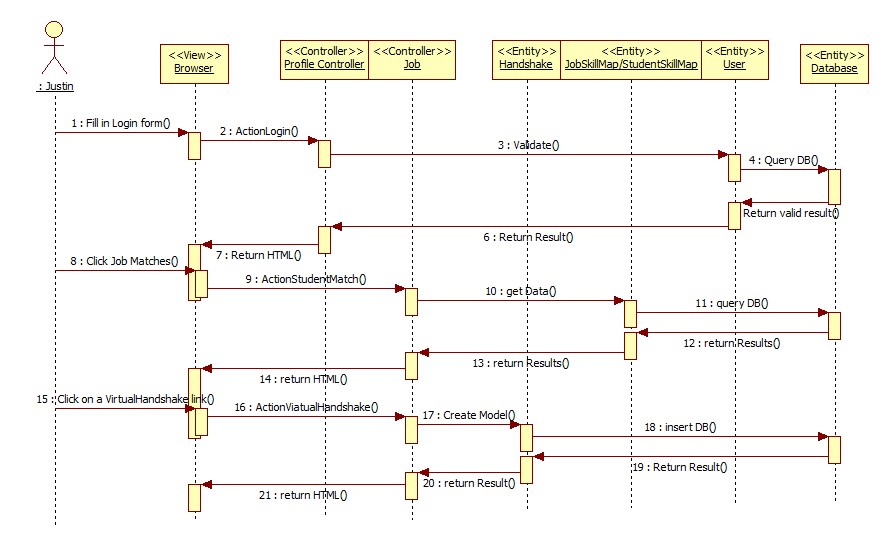
VJB-029 Post Job (Sunny Day)



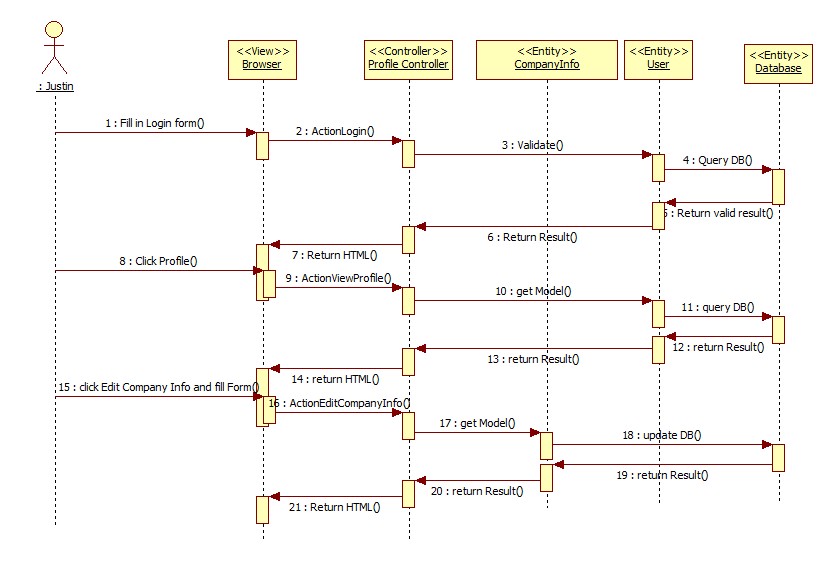
VJB-0029 Post Job (Rainy Day)



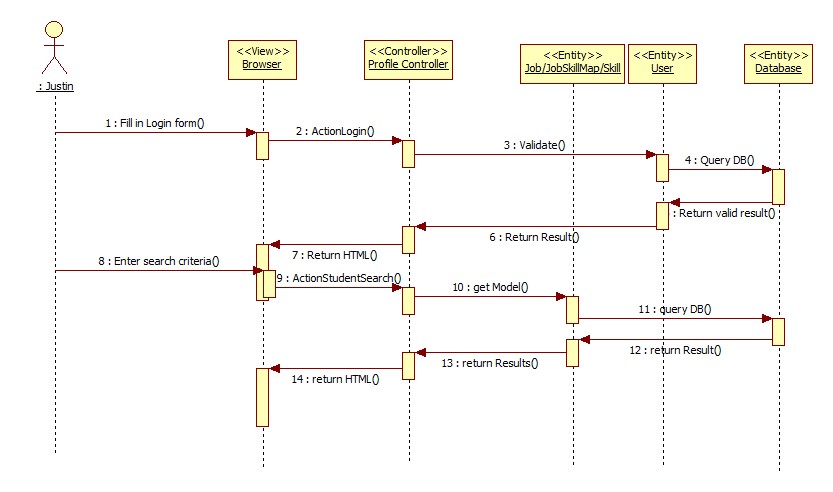
VJB-030 Virtual Handshake (Sunny Day)



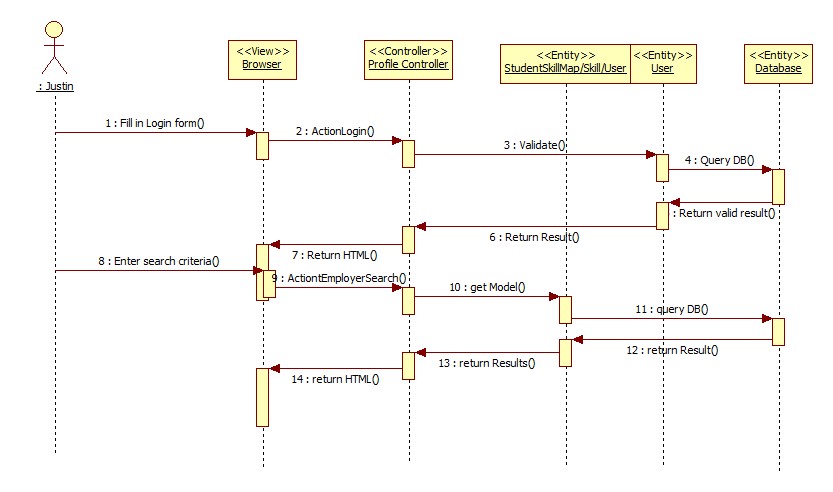
VJB-031 Edit Company Info (Sunny Day)

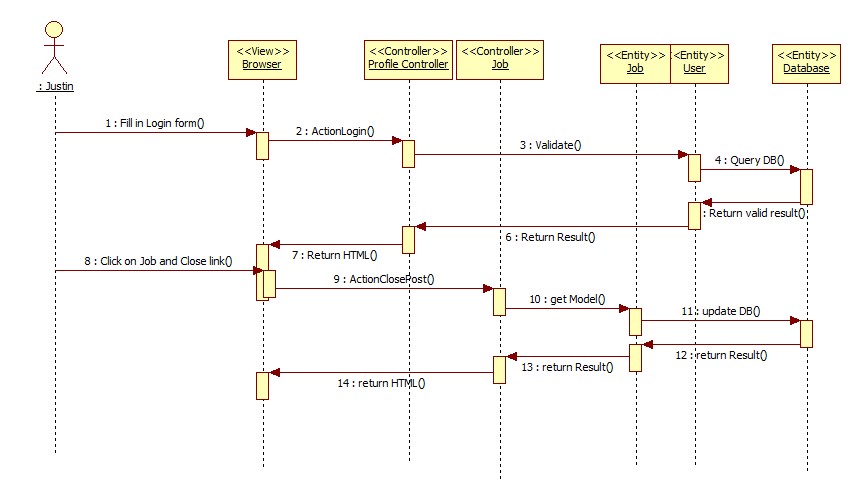


VJB-032 Search Jobs (Sunny Day)

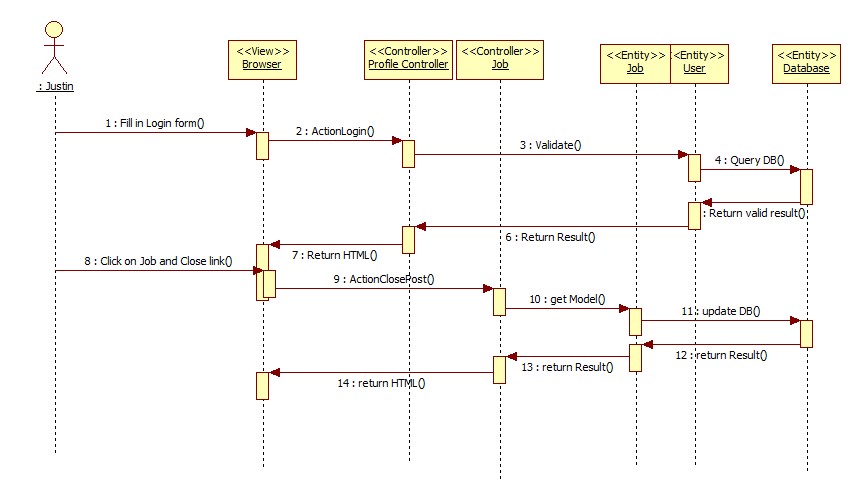


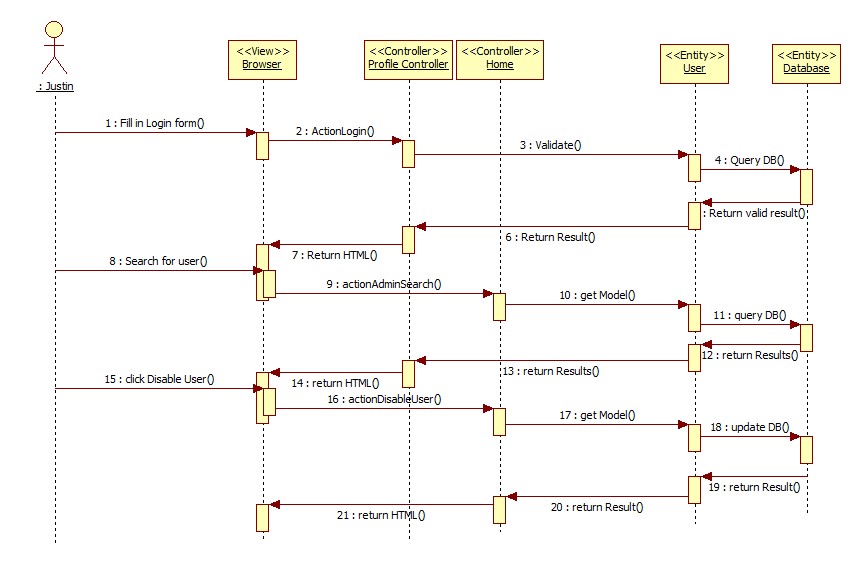
VJB-033 Search for Students (Sunny Day)

VJB-034 Close Job (Sunny Day)

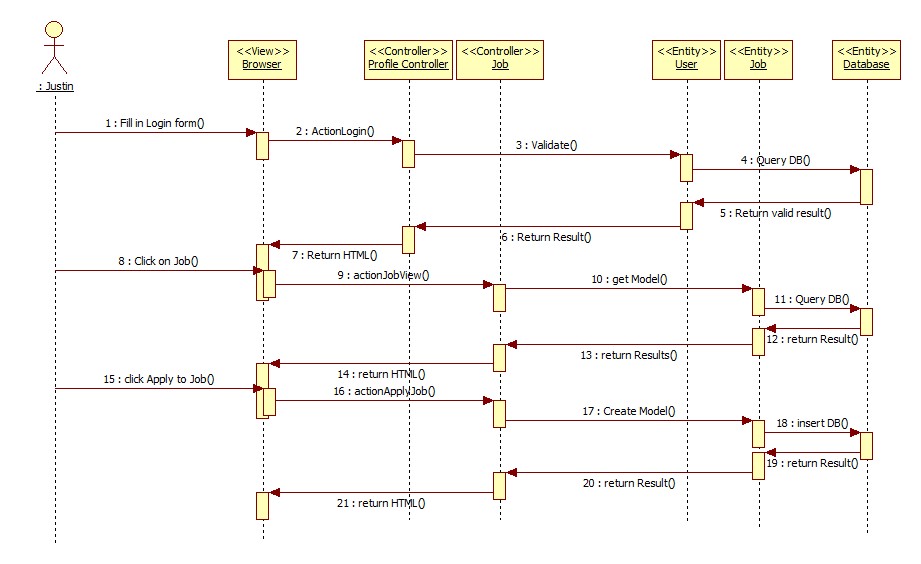


VJB-035 Admin Close Job (Sunny Day)

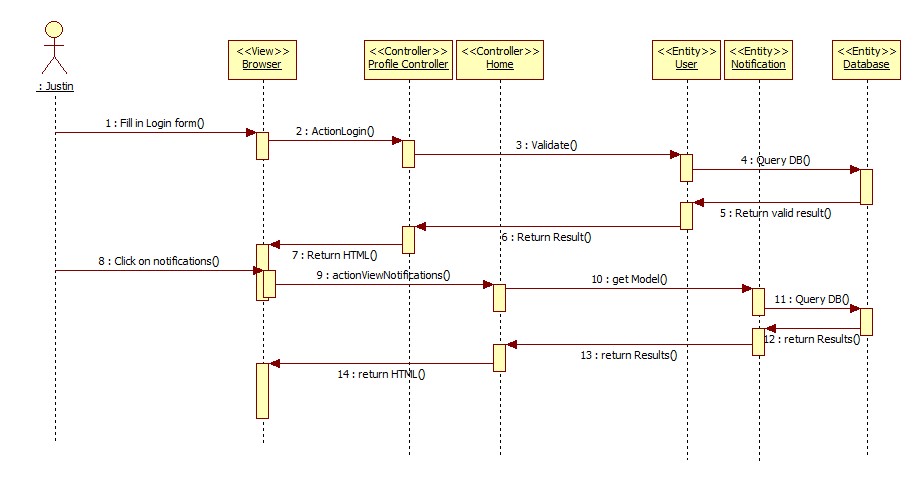
VJB-036 Disable User (Sunny Day)



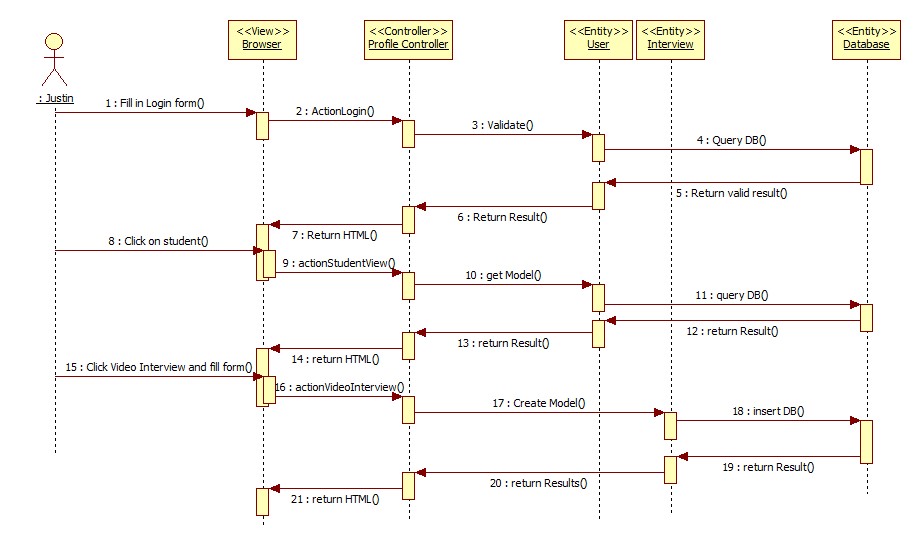
VJB-037 Apply to Job (Sunny Day)

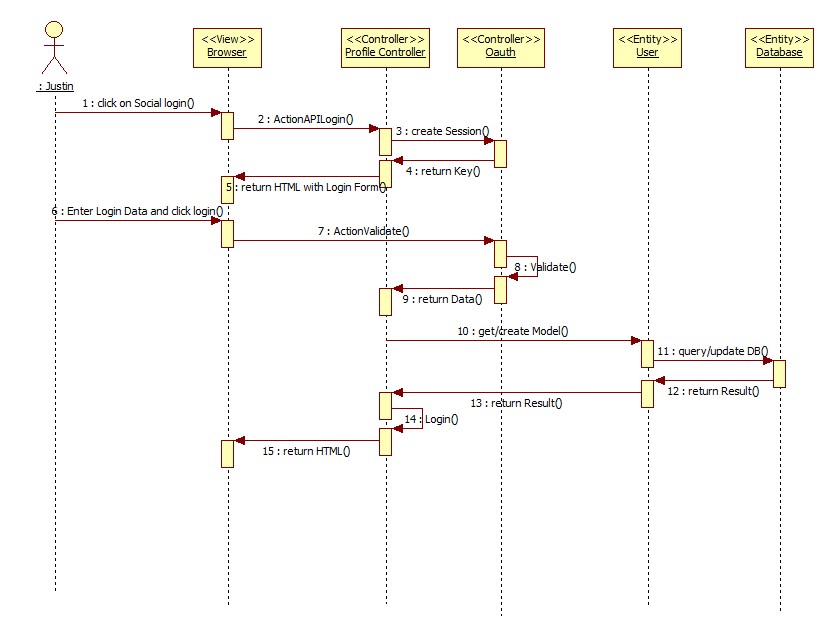


VJB-038 Read Notification (Sunny Day)

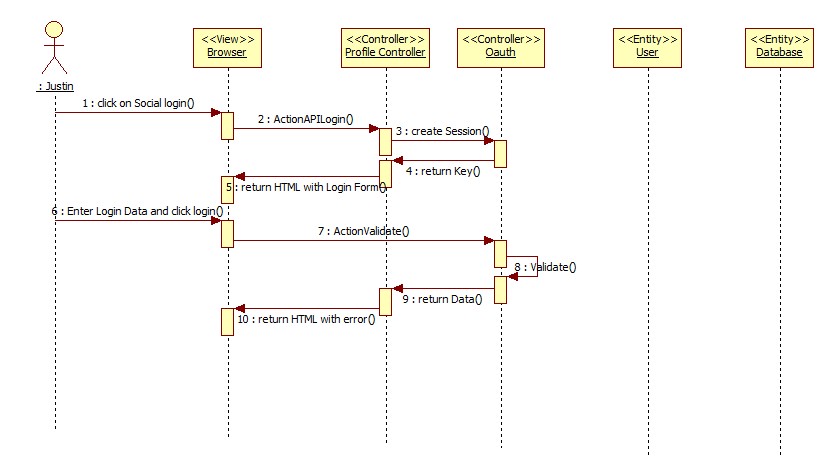


VJB-039 Schedule Interview (Sunny Day)

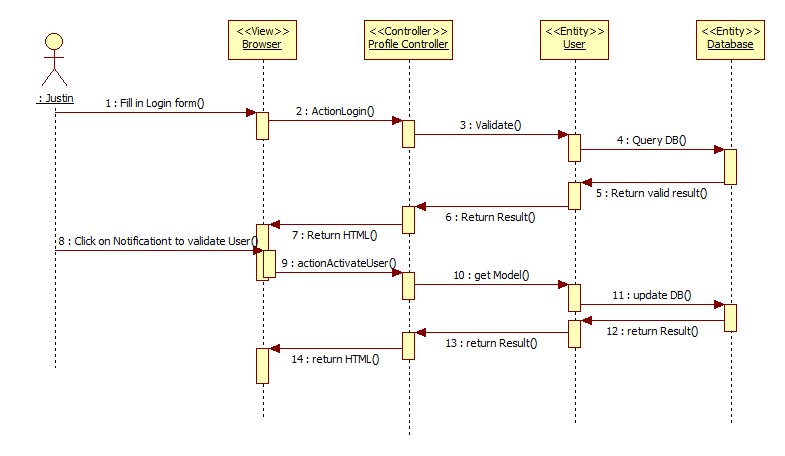
VJB-040 Login/Register With Social API (Sunny Day)



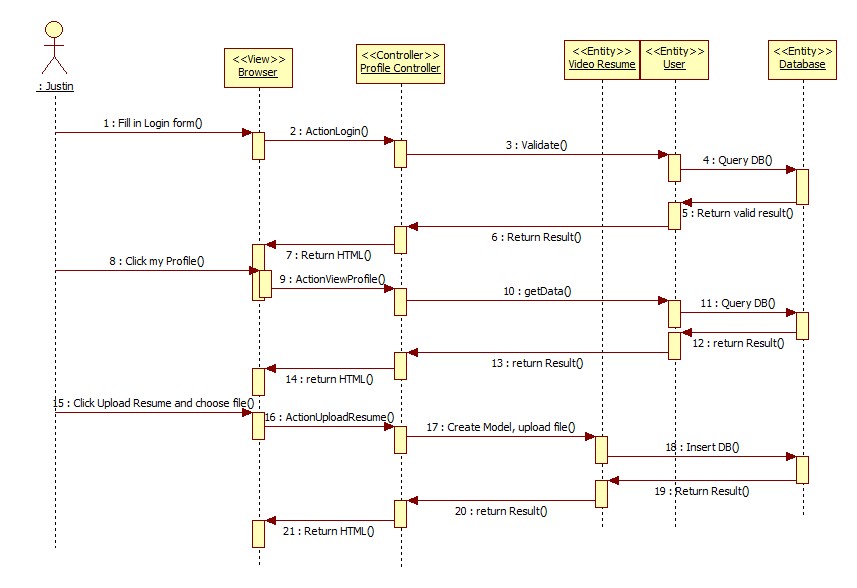
VJB-040 Login/Register With Social API (Rainy Day)



VJB-041 Validate Employer (Sunny Day)



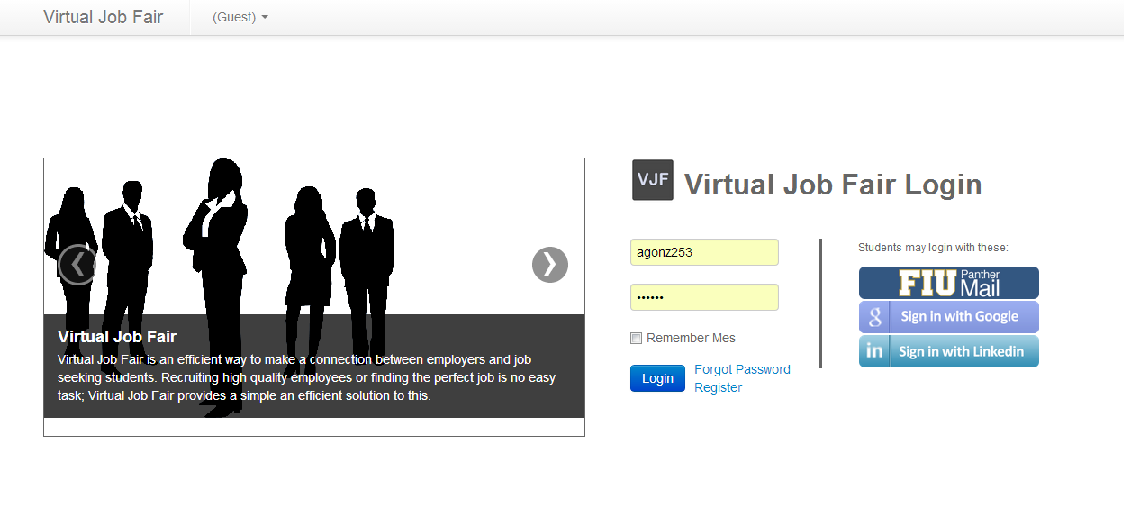
VJB-042 Upload Video Resume (Sunny Day)



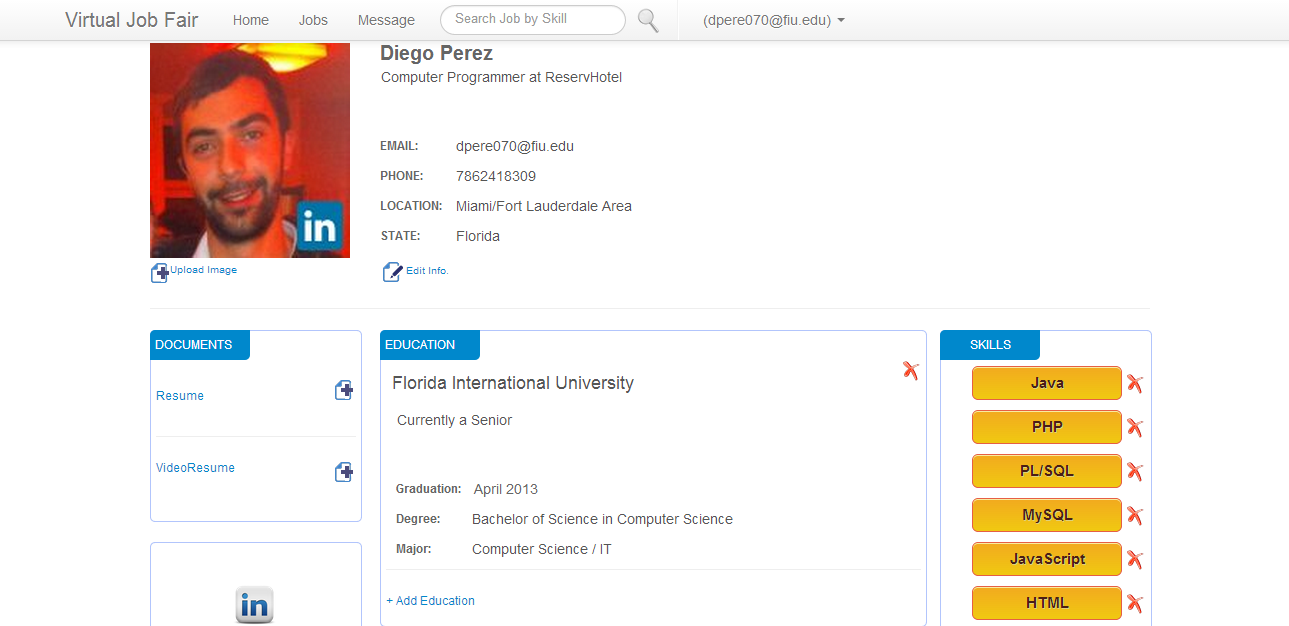
## 6.5 Appendix E - User Interface designs.

To come up with the UI design of the system, a number of paper prototypes were produced in the brainstorming stage. Following that, the following UI design was decided upon:

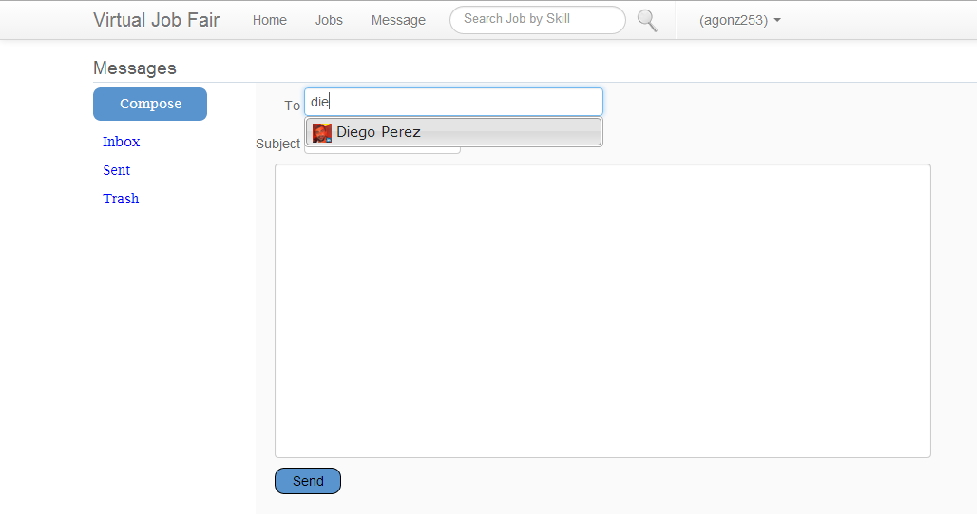
Login Page:



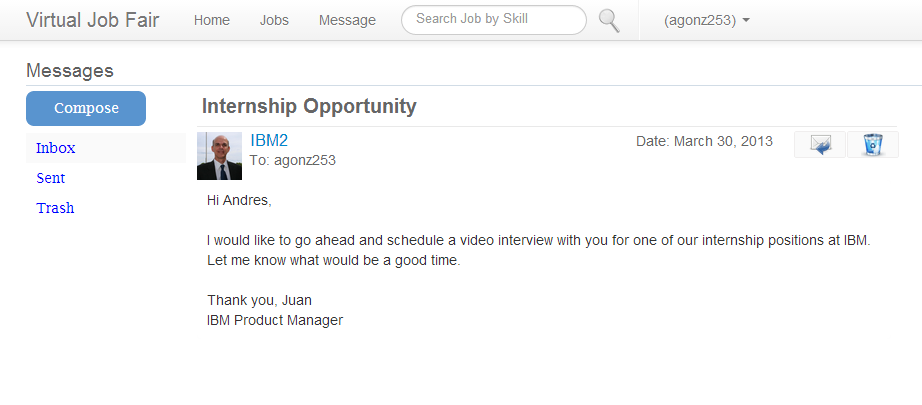
Profile Page:



Compose Message Page:



Message Page:



Jobs Page:



Home Page:

6.6 Appendix F - Diary of meeting and tasks.

|  |  |  |
| --- | --- | --- |
| Date | Activities | Participants |
| 01/29/2013 | Worked on the database design | Every member of the team |
| 02/02/2013 | Worked on Deliverable 2 Project Plan. Decided on work breakdown | Every member of the team |
| 02/05/2013 | Finished the database design | Every member of the team |
| 02/09/2013 | Identified all the Use Cases. | Every member of the team |
| 02/12/2013 | Started working on the system design, identifying the main classes required | Every member of the team |
| 02/16/2013 | Finished System Design. Worked on the Dynamic Model of the System (sequence diagrams) | Every member of the team |
| 02/19/2013 | Finished 2nd Deliverable | Every member of the team |
| 02/23/2013 | Worked on the implementation of general system functionality | Every member of the team |

# 7. References

None.