***Enhanced Job and Candidate Application***

***Design Specification***

C:\Users\Simul\Desktop\Winter 2015\CSC 4996\MVC App\User Login\User Login\Content\Images\computech logo.png 

Version: 1.0

03/10/2015

**Prepared by:**

Simul Kadakia

Wesley Trescott

Gagandeep Singh

Contents

[1. Introduction 3](#_Toc412996026)

[1.1 Purpose 3](#_Toc412996027)

[1.2 Scope 3](#_Toc412996028)

[1.3 Definitions, Acronyms and Abbreviations 3](#_Toc412996029)

[1.4 References 3](#_Toc412996030)

[2. Assumptions / Constraints / Standards 4](#_Toc412996031)

[2.1 Design Constraints: 4](#_Toc412996032)

# 1. Introduction

This section gives a scope description and overview of everything included in this document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

## Purpose

The purpose of this Software Design Specification (SRS) document is to provide a detailed description of the design framework of the ‘Enhanced Job and Candidate Application’ system. It will also provide specific information about the input and desired output, software and hardware architecture, database, security and sequence diagrams and test cases persisting to the application.

## Scope

Enhanced Job and Search Candidate is an application developed by 3 students at Wayne State University for Computech Corporation. The goal of the application is to provide a web application service to job seekers to search for available jobs at Computech Corporation and apply to those that they are interested in. The application will provide filtering options to reduce the jobs displayed based on certain criteria. Job seekers will be able to store their profile information and resume path (resume will be stored in a folder) in database and resume which can be used when they are applying for a job using a web interface. The application will also provide functionalities to admin to deactivate and delete users abusing the system.

## 1.3 Definitions, Acronyms and Abbreviations

* EJCA - Enhanced Job and Candidate Application
* User – Job seeker who uses the application
* Admin – Admin/administrator who manages the users
* Admin portal - Part of the web application that provides special facilities to Admin
* Front End - The part of the application the user interacts with
* Back End – The part of application that manages data and is managed by developers.
* UI – User Interface which is the front end of the application
* Server –Machine that will host the web application as well as database.

## 1.4 References

* Microsoft ASP.NET MVC - <http://www.asp.net/mvc>
* Microsoft SQL Server - <https://msdn.microsoft.com/en-us/sqlserver/aa336270.aspx>
* Razor - <http://www.asp.net/web-pages/overview/getting-started/introducing-razor-syntax-(c)>
* JQuery - <http://jquery.com/>
* Bootstrap – <http://getbootstrap.com>

# 2. Assumptions / Constraints / Standards

## 2.1 Design Constraints:

This is a web application and the context of rendering the user interface based on the type of device used. End users will use the application from a modern internet browser such as Safari, Chrome, or Internet Explorer.

## 2.2 Assumptions and Dependencies

Assumptions to properly use the application include:

* Internet connection with enough bandwidth (about 1 to 2 mbps) to fully render all web application pages
* A modern internet browser with an up to date JavaScript engine and support for HTTP cookies to remember returning user logins

## 2.3 Components:

* **Microsoft ASP.NET MVC framework:** EJCA will be developed using MVC which is a software [architectural pattern](http://en.wikipedia.org/wiki/Architectural_pattern) for implementing [user interfaces](http://en.wikipedia.org/wiki/User_interface). It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user
* **Microsoft SQL Server:** EJCA will use MS SQL server database which will store users, jobs and admin information.
* **Razor:** Razor is a view engine which will embed server-based code on the web page.
* **JQuery:** EJCA will be using JavaScript library for more user and mobile friendly EJCA.
* **Bootstrap:** EJCA needs to be compatible on mobile browsers along with traditional desktop and bootstrap framework will enable to use the same layout on multiple platforms.

# 3. Architecture Design

## 3.1 Logical View

## 3.2 Hardware Architecture

## 3.3 Software Architecture

## 3.3 Security Architecture

## 3.4 Communication Architecture

## 3.5 Performance

# 4. System Design

## 4.1 Use-cases

## 4.1.1 Search:

**Actors:** Job seeker

**Description:** User can search for various jobs available. When the user opens the website, the home page will show featured jobs with an option to conduct a job search. Users can also conduct a search for the same jobs once logged into their profiles.

**Trigger:** User clicks on **Search Jobs** button.

**Pre-conditions:** None

**Post-conditions:** None

**Normal Flow:**



**Alternative flow:** N/A

**Exception:** N/A

**Assumptions:** None

## 4.1.2 Register:

**Actors:** Job seeker

**Description:** Users can create a new account by providing personal information such as email id, password and first and last name. This information will be stored in database and email id and password will be used to log in user.

**Trigger:** User clicks on **Register** button under Users dropdown menu on the header.

**Pre-conditions:** None

**Post-conditions:** None

**Normal Flow:**



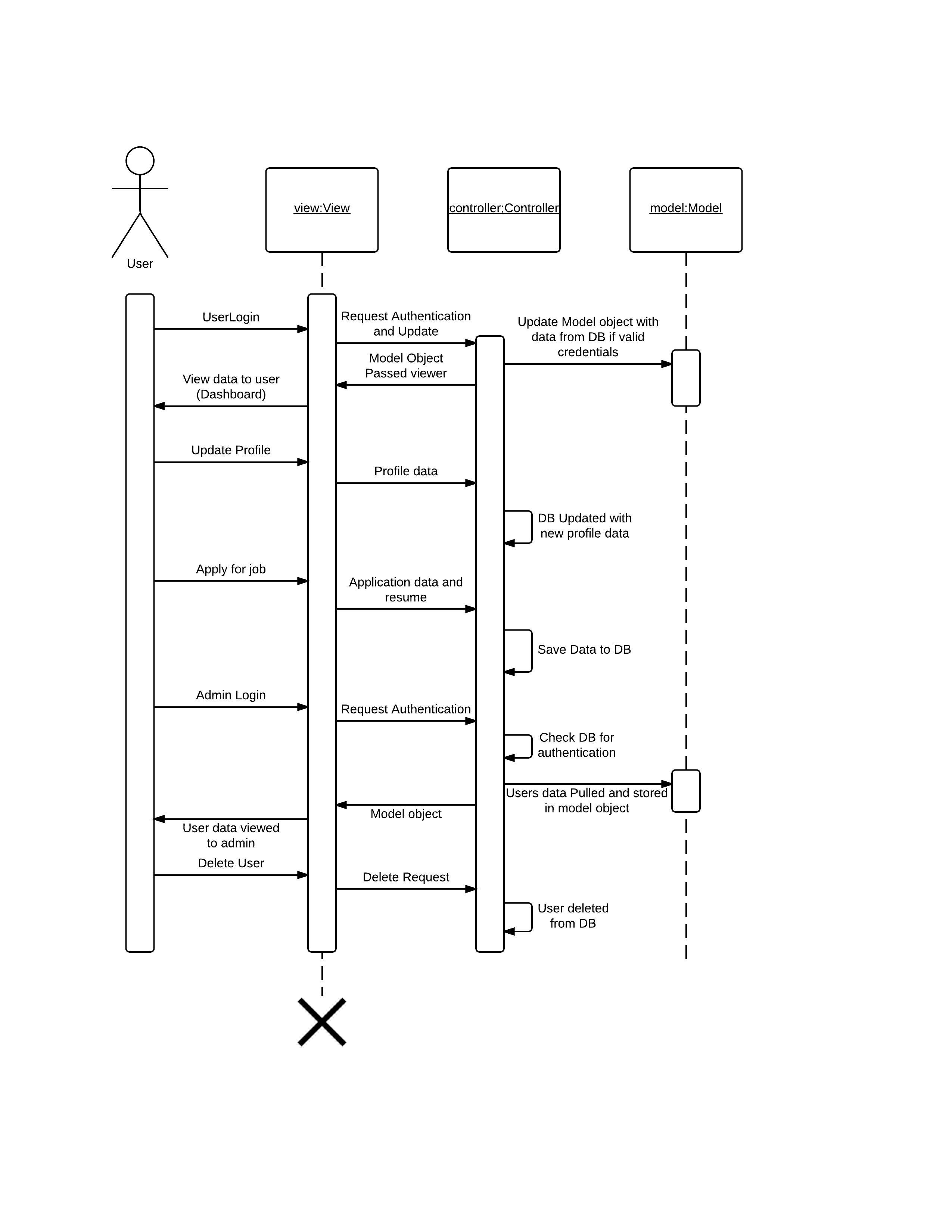
**Alternative flow:** N/A

**Exception:**



**Assumptions:** User has not previously registered with the same email address.

## 4.2 Sequence Diagram



## 4.3 Data-flow diagram

## 4.3 Database Design



## 4.4 Application Program Interface

## 4.5 User Interface Design

# 5. Product Design Specification Approval