

# Software Requirements Specification

---

For The

## ***Enhanced Job and Candidate Application***

**Prepared by:**

Simul Kadakia  
Wesley Trescott  
Gagandeep Singh

## Table of Contents

SOFTWARE REQUIREMENTS SPECIFICATION.....	1
<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1 PURPOSE .....	1
1.2 SCOPE.....	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS .....	2
1.4 REFERENCES.....	2
<b>2. GENERAL DESCRIPTION.....</b>	<b>3</b>
2.1 PRODUCT PERSPECTIVE .....	3
2.2 PRODUCT FUNCTIONS .....	3
2.3 USER CHARACTERISTICS.....	3
2.4 GENERAL CONSTRAINTS .....	3
2.5 ASSUMPTIONS AND DEPENDENCIES .....	3
<b>3. SPECIFIC REQUIREMENTS .....</b>	<b>4</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	4
3.1.1 <i>User Interfaces</i> .....	4
3.1.2 <i>Hardware Interfaces</i> .....	5
3.1.3 <i>Software Interfaces</i> .....	5
3.1.4 <i>Communications Interfaces</i> .....	5
3.2 FUNCTIONAL REQUIREMENTS .....	6
3.2.1: <i>User class - Users</i> .....	6
3.2.2: <i>User class - Admin</i> .....	7
3.3 NON-FUNCTIONAL REQUIREMENTS .....	8
3.3.1 <i>Performance</i> .....	8
3.3.2 <i>Reliability</i> .....	8
3.3.3 <i>Availability</i> .....	8
3.3.4 <i>Security</i> .....	8
3.3.5 <i>Maintainability</i> .....	9
3.3.6 <i>Portability</i> .....	9
3.4 DESIGN CONSTRAINTS .....	9
3.5 LOGICAL DATABASE REQUIREMENTS .....	10
3.6 OTHER REQUIREMENTS .....	10
<b>4. ANALYSIS MODELS .....</b>	<b>10</b>
4.1 DATA FLOW DIAGRAMS (DFD).....	10

## **1. Introduction**

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

### **1.1 Purpose**

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the functionalities of the 'Enhanced Job and Candidate Application' system. Additionally, it will serve as a tool to communicate the design plans of Enhanced Job and Candidate Application with Computech Corporation so that both parties may fully understand the specifications, requirements, and design of the project. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

### **1.2 Scope**

The 'Enhanced Job and Candidate Application' is a web application targeting initially Wayne State University students, later a general audience who can search for jobs based on their skill sets and apply to those they are interested in. The web application is not specific to a particular device and needs to be compatible with smartphones, tablets, and desktop environments.

Users can log into a secure portal where they can access their saved data such as email address, contact information, and experience. They can filter job searches based on specific criteria, such as location and industry, which will narrow the search results of the available jobs. Users can apply to the jobs that they are interested in, with the option of uploading a resume. To apply, users can either manually enter their data or populate the fields from saved data previously entered in their user profile. Based on users' saved information, the application will provide the user with an automatic job matching functionality that displays to the user a list of jobs potentially fitting his or her credentials.

Additionally, the application will feature an admin mode that will interface with the Computech's Pinpoint system to delete or ban users, lock accounts, and search for users based on name, email, or location.

### 1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
EJCA	Enhanced Job and Candidate Application
User	Job seekers who use the application
Admin/Administrator	System administrator who is given specific permission for managing and controlling the system
Admin portal	Part of the web application that provides special facilities to Admin/Administrator
Stakeholder	Any person who interacts with the application except developers
API (Application Programming Interface)	A generalized interface used to allow EJCA to interact with external software/applications (Pinpoint)
Front end	The part of the application the user interacts with
Back end	The part of application that handles data and is maintained by developers
UI (User Interface)	Front end of EJCA
DESC	Description
DEP	Dependency
MUST	The minimum level required to avoid failure
PLAN	The level at which an endeavor is successful
WISH	A desirable level of achievement that may not be attainable through available means
DEFINED	The official definition of a term

### 1.4 References

“Pinpoint.” *Computech Corporation Products*. Computech Corporation, n.d. Web. 9 Feb 2015.

Pliastas, Vygandas. “Creating a custom user login form with .NET C# MVC 4 Razor.” *Code Project: For those who code*. Code Project, 23 Oct 2012. Web. 9 Feb 2015.

*Additional helpful sources not directly referenced in this document:*

- 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-\(c\)](http://www.asp.net/web-pages/overview/getting-started/introducing-razor-syntax-(c))
- JQuery - <http://jquery.com/>

## 2. General Description

### 2.1 Product Perspective

The project is built from scratch, and thus there is no initial foundation to build our system upon. The system will consist of a backend and a frontend. The backend will handle the storage of job listings and user information data, as well as the querying and delivery of this data back to the frontend to be displayed. The frontend will allow a user to interact directly with a webpage which will allow a user to specify a particular filter and then send a query to the backend to retrieve data.

### 2.2 Product Functions

- User can login
- User can browse open job listings
- User can search/filter based on certain criteria
- User can apply to jobs
- User can check what jobs he or she has applied to
- Admin can login
- Admin can search for users by his or her first and last name
- Admin can lock/ban/delete users

### 2.3 User Characteristics

We expect two different types of users for this project: Users and Admin.

Users are considered the most important interactors because they are the ones who search for jobs and apply to the ones that they are interested in. They are also going to be the majority of site views because the web app is mainly focused on delivering content to them. Most potential users will simply browse job listings until they find job listing(s) they are interested in, at which point they can apply.

Admins are the second most important users of our site. They can search for users based on their first and last names. They can also lock, ban and remove users who abuse the application and apply to multiple jobs in a short period of time without any relevant experience/skills.

### 2.4 General Constraints

This is a web application and the details about the operating environment are not important. End users will use the application from a modern internet browser such as Safari, Chrome, or Internet Explorer.

### 2.5 Assumptions and Dependencies

Assumptions to properly use the application include:

- Internet connection with enough bandwidth to fully render various displays
- A modern browser with up to date technologies, such as a modern JavaScript engine

### 3. Specific Requirements

#### 3.1 External Interface Requirements

##### 3.1.1 User Interfaces

Users should be able to see the job search page directly when the application is opened. Here the user chooses the type of search he or she wants to conduct by filtering with combo boxes and checkboxes. There will also be a Login link if the user wants to log into his or her account and a Register link if the user wants to create a new account. Once logged in, the user will see a dashboard screen displaying jobs matching his or her experience or credentials for which he/she can apply. The user will also have a search functionality to search for additional jobs.

Every page will include a standard header with the aforementioned links. Some preliminary user interface pages are shown below:

The login page features a dark gray header bar at the top with the links 'Home', 'Register', and 'Sign In' in white. Below the header, the word 'Login' is centered. The form includes a 'Username' label above a text input field, a 'Password' label above another text input field, a checkbox labeled 'Remember me on this computer', and a 'Log In' button.

A preliminary user login with the gray header bar

The dashboard has a blue header bar with 'Home', 'Logout', and 'Settings' (with a gear icon). Below the header, it says 'Welcome, John Doe!'. The main content area is divided into two sections. The left section displays a 3x3 grid of 'Job Info' buttons in alternating blue and orange colors. The right section, titled 'Search for Jobs', contains two dropdown menus labeled 'Column 1' and a large red rectangular placeholder for search results.

A preliminary user dashboard with recommended jobs at center and a job search option to the right.

### **3.1.2 Hardware Interfaces**

To function properly, the EJCA will require two types of hardware interfaces: a webserver and the user's machine. The webserver will be the physical machine hosting the site, including the server software it runs; the webserver will receive HTTP requests and provide HTTP responses, ensuring the constant availability of the application. This piece of hardware is owned by Computech. Additionally, each user of the application will access it using his or her own machine, whether that be in a desktop environment, or through a laptop, smartphone, or tablet.

### **3.1.3 Software Interfaces**

EJCA will run on the MVC architecture, a design pattern offered as part of the Microsoft ASP.NET framework. Since this architecture provides a simple three-tier system for displaying pages and managing data, it is ideal for the needs of EJCA. Being a web application, EJCA will also require a web service, which will run on a web server. Both these software interfaces are vital to the public availability of the application so that it will be accessible to its users.

### **3.1.4 Communications Interfaces**

As discussed above, the web application utilizes the MVC architecture, so communication between the model, view, and controller components of the system is important since they depend on each other. Though communication between each of these components is handled via the .NET framework, accessibility of the application itself to users is handled by the HTTP protocol. The messages passed will be GET and POST requests from the user or admin, as well as calls to insert, delete, or select data from our database.

## **3.2 Functional Requirements**

### **3.2.1: User class - Users**

#### **3.2.1.1 Search:**

**DESC:** User can search for various jobs available. When the user opens the website, the page will show available jobs. User can search among these jobs.

**REQ:** None.

#### **3.2.1.2 Register:**

**DESC:** User 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.

**REQ:** None.

#### **3.2.1.3 Login:**

**DESC:** Given that a user has registered, the user can then login by entering email id and password entered during registration process.

**REQ:** FR 3.2.1.2

#### **3.2.1.4 Retrieve password:**

**DESC:** Given that a user has registered, the user should then be able to retrieve password through email. User can request to change password which will send further instruction to his or her email.

**REQ:** FR 3.2.1.2

#### **3.2.1.5 Apply:**

**DESC:** Given that a user is logged in, the first page he or she can see is the home page. User can search/filter jobs and apply for jobs.

**REQ:** FR 3.2.1.3

#### **3.2.1.6 Filter by skills, location, and experience:**

**DESC:** User can filter jobs based on skills, location and experience required for the job. Jobs can be filtered based on single criteria or multiple criteria.

**REQ:** None

#### **3.2.1.7 Profile Information:**

**DESC:** Given that a user is logged in, the user can enter their information such as skills and experience in his or her profile page. The user can store his or her information which can be used to autofill fields when applying for a job.



**REQ:** FR 3.2.1.2

**3.2.1.8 Change Information:**

**DESC:** Given that a user is logged in, the user can change information stored in his or her profile page. User should be able to change any fields stored in his or her profile page.

**REQ:** FR 3.2.1.3

**3.2.1.9 Upload Resume:**

**DESC:** Given that a user is logged in, the user can upload a resume and attach it to his or her profile. User can only attach one resume file which will be stored in his or her profile information in the database.

**REQ:** FR 3.2.1.3

**3.2.2: User class - Admin**

**3.2.2.1 Login:**

**DESC:** Admin can login by entering email id and password stored in database. Admin information is stored in database and there is no registration page for admin.

**REQ:** None

**3.2.2.2 Search:**

**DESC:** Admin can search for all registered users by their first and/or last names. Admin can search for users which will display basic user information such as name, skills and so on.

**REQ:** FR 3.2.2.1

**3.2.2.3 Deactivate users:**

**DESC:** Admin can deactivate users which will disable them to login into their account. Users who abuse the system can be deactivated by the admin which will lock them to log in to their account.

**REQ:** FR 3.2.2.1

**3.2.2.4 Delete users:**

**DESC:** Admin can delete users which will delete all of their information from the database. Users who abuse the system can be deleted by the admin. All information related to user such as login and profile information will be deleted from the system.

**REQ:** FR 3.2.2.1

### **3.3 Non-Functional Requirements**

The following requirements specify, for the system level, the mandatory performance, reliability, availability, security, maintainability, and portability of the software product.

#### **3.3.1 Performance**

This requirement specifies the mandatory performance metrics of the system during user interaction with the software. This includes response time to user logins, registrations, job searches, and admin logins and activities. Included in this metric are page transition times, as for example from the home screen to the job search screen. Since database queries and requests in the application deal with relatively small amounts of information, with either basic user information or a number of job listings being retrieved from or entered into the database, response times should be quick. In particular, we aim for overall loading quickness to be at a maximum of five seconds 100% of the time, with a target of less than two seconds 98% of the time, which will be verified during five hours of testing in our phase 3 system tests, as per our development plan.

#### **3.3.2 Reliability**

This requirement specifies the extent to which the software application reliably produces the correct job search results based on valid search criteria, as well as suggests jobs to users based on their skills and experience. Also, this metric encompasses report messages to application users regarding lost internet connections or incorrect registration or search inputs. Each of these requirements will be thoroughly tested and evaluated during the phase 2 testing mentioned in the development plan. In particular, we plan to create at least 25 user accounts with both valid and invalid credentials (invalid credentials being, for example, a duplicate email address or password). These user accounts will contain various experience data, which will be used to assess job recommendation functionality. We will also perform at least 50 job searches using these accounts on a limited jobs database to assess correctness of search algorithms. In these tests, we will require correct error reporting 100% of the time, correct job search results 98% of the time, and correct job recommendations 98% of the time.

#### **3.3.3 Availability**

This requirement specifies the readiness of the system for use whenever it is run. This entails the accessibility of the application from an internet browser not considering network failures. Thus, this requirement assumes that the user has an internet connection so that the application can connect to the database. The availability requirement will be tested in the phase 3 system tests mentioned in the development plan. In testing, we plan for five hours of active use of the application, during which period our target system availability is 100% of the time, with a 98% of the time availability considered passing.

#### **3.3.4 Security**

This requirement specifies that the admin mode and all user passwords will be encrypted using the SHA1 encryption algorithm before being stored in the database. SHA1 was developed by the NSA and is an example of a cryptographic hash function, considered nearly impossible to decode, thus ensuring the

security of user password information from the system administrator. This requirement also encompasses the validation of user logins, so that incorrect user credentials will not result in access to the application. Furthermore, the functionality of the admin mode ensures that users abusing the application will be removed. Each of these security requirements will be expected to function correctly 100% of the time in our tests.

### **3.3.5 Maintainability**

This requirement specifies the degree of extensibility and testability of the EJCA. Firstly, extensibility implies both the easy addition of new features to the application and also the ease of portability to new, previously unsupported devices. This requirement is satisfied by our use of popular technology (ASP.NET), which is a framework widely used in web application development across multiple platforms, therefore attesting to its adaptability to new environments. Also, our program will be written using the MVC design pattern, which allows for flexible addition of new features through new controllers and models. Furthermore, our application supports easy testability by taking advantage of the test suites generated by ASP.NET for all of the functionality of the EJCA.

### **3.3.6 Portability**

This requirement specifies that the EJCA be portable to multiple devices. Given the up-front project requirement of compatibility with both mobile and desktop environments, the application will be designed and coded so that all its functionality is compatible with web browsers in PC, Mac, Android, iPhone, and iPad. The effect of this multi-platform design intent is that the application will be easily portable to each supported environment. Complete system testing of all functionality of the application will be carried out on each supported device during phase 3 testing.

## **3.4 Design Constraints**

The largest design restraints in our team's view are the amount of RAM the web application consumes during normal use as well as the application's rendering in the default internet browsers of multiple devices. Consuming too much memory will slow down the user's device and provide a less than optimal user experience. Therefore, we have set the constraint that the EJCA will require at most four gigabytes of RAM during normal use. Also, due to differences in browser software across devices, and differences in screen size and resolution, our user interface is constrained in size and complexity to support portability to touchscreen devices and devices with small screens.

A further design constraint is the difficulty of implementing a social media login, as originally anticipated. Due to user data privacy constraints, logging into the application using the user's social media account remains a challenge, to be left for possible implementation at a later stage in the application development. This constraint necessitates the user's manual entry of data that could otherwise be automatically populated from the records of a social networking site.

### 3.5 Logical Database Requirements

A Microsoft SQL Server database will be used in the backend to store information entered by the user, display jobs to users, and display registered users to admin. The front-end will be on the user/admin machine.

### 3.6 Other Requirements

#### 3.6.1 Mobile Platform Compatibility

The web application is compatible with mobile phones and tablets along with PCs, primarily iOS and Android based devices. The user view is configured to the device's screen resolution and can be accessed from any supported web browser installed on the device. The user has the ability to use all functionality in the app on these mobile devices. However, the user is unable to upload resumes from iOS based devices.

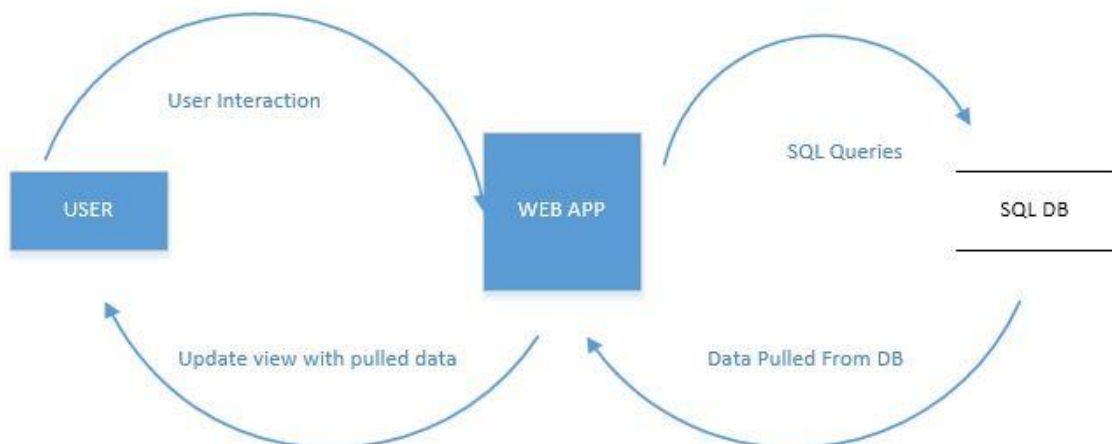
## 4. Analysis Models

### 4.1 Data Flow Diagrams (DFD)

The following diagrams model the processes of our application, depicting both how data flows between the user and the application, and how data flows within the application.

#### 4.1.1 Level1 DFD

The Level1 DFD shows the flow of data among the user, web application, and the SQL database.



#### 4.1.2 Level2 DFD

Level2 DFD provides in-depth view of the data flow among the various components of the web application such as the UI, Application Core i.e. Model and Controller components, Viewer, and the SQL database.

