



**CV Scanner**

# SOFTWARE REQUIREMENTS SPECIFICATIONS

**Version:** 4.0.0

**Date:** September 28<sup>th</sup>, 2025

**Status:** Final

**Document Type:** Software Requirement Specification

**Project:** CV Scanner - Automated CV Analysis System

## **Team Members**

Marcelo Parsotam

Unaisah Hassim

Talhah Karodia

Abdullah Pochee

Ronan Smart

## Table of Contents

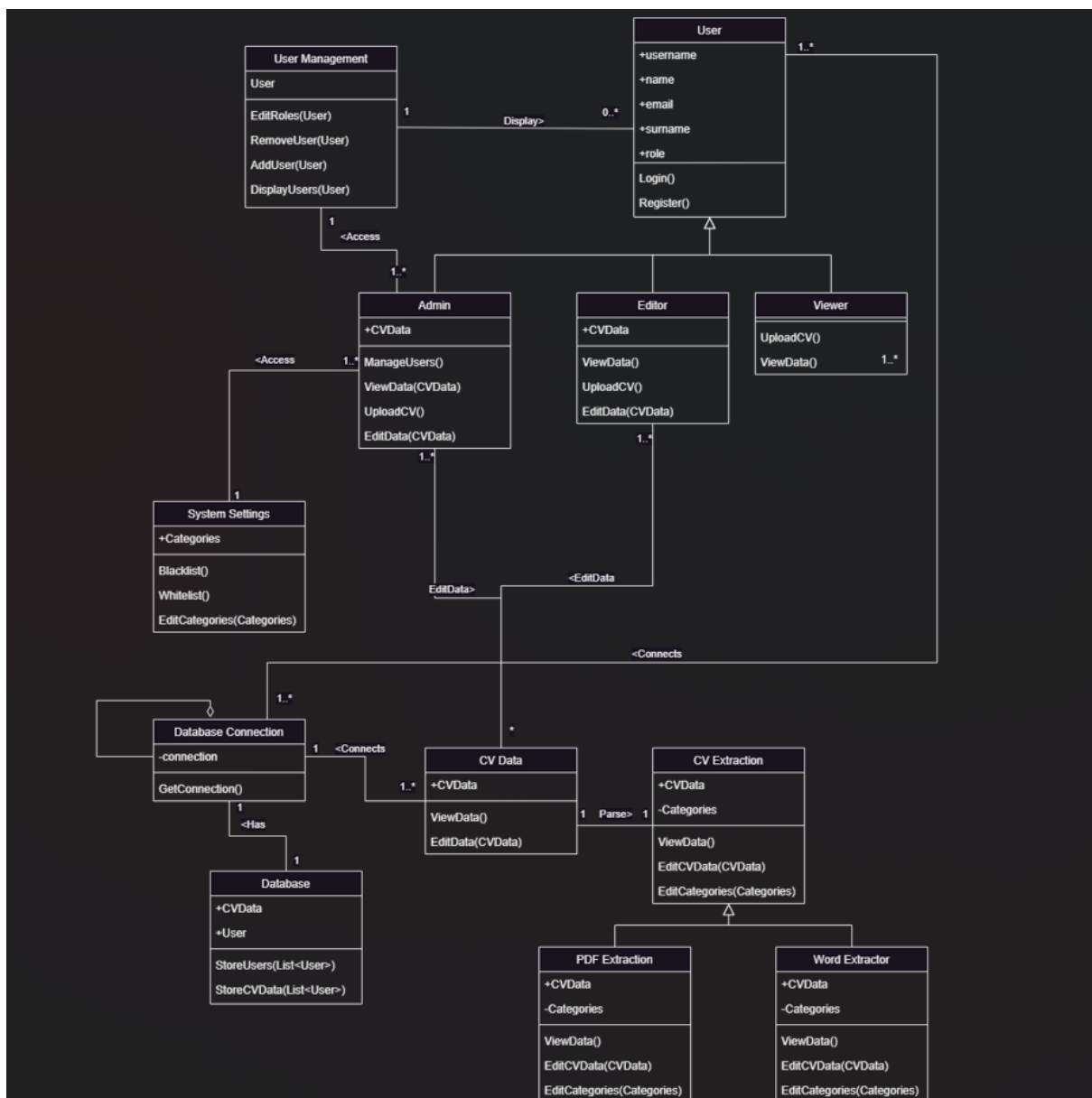
---

1. Introduction .....	2
2 Domain Model .....	2
3 User Stories/ User Characteristics .....	3
3.1. Viewer .....	3
Characteristics: .....	3
Permissions: .....	3
User Story: .....	3
3.2. Editor .....	3
Characteristics: .....	3
Permissions: .....	3
User Story: .....	3
3.3. Admin.....	3
Characteristics: .....	3
Permissions: .....	4
User Story: .....	4
4. Use Cases.....	5
UC1: Upload CV.....	5
UC2: View Candidates.....	5
UC3: Edit Candidate Data.....	6
UC4: Manage Users .....	6
UC5: Configure System.....	7
UC6: Delete Records .....	7
UC7: Assign Roles .....	8
UC8: Extract CV Data.....	8
UC9: Get Profile Summary.....	8
5. Use Case Diagrams.....	9
6. Functional Requirements .....	10
7. Optional Functional Requirements .....	12
8. Non-Functional Requirements .....	13
9. Service Contracts.....	13

## 1. Introduction

Entelect receives a high volume of CVs from job applicants. Manually reviewing these is time-consuming and error prone. The CV Scanner project aims to automate CV parsing and analysis using NLP and machine learning to extract relevant technologies, summarize candidate profiles, and recommend suitable project environments. This will streamline recruitment and enhance hiring accuracy.

## 2 Domain Model



## 3 User Stories/ User Characteristics

---

### 3.1. Viewer

Characteristics:

- Typically, a hiring manager or junior recruiter.
- Cannot modify or upload data.

Permissions:

- View candidate summaries and profiles.

User Story:

- As a viewer, I want to browse candidate summaries without making any changes, so I can assess candidates securely and pass feedback to editors or admins.
- As a viewer, I want to View a specific candidate's work history and skills without making any changes.

### 3.2. Editor

Characteristics:

- Recruiter or HR personnel responsible for curating CV data.
- Has permissions to modify data but not to manage users or system settings.

Permissions:

- View and edit candidate data.
- Upload CVs.
- Cannot manage users, configure the system, or delete records.

User Story:

- As an editor, I want to upload CVs and edit candidate profiles, so that I can keep candidate data accurate and up to date for recruiters.

### 3.3. Admin

Characteristics:

- Technical or HR system administrator.

## CV Scanner – Software Requirements Specifications

- Full control of the platform.

### Permissions:

- View and edit candidate data.
- Upload CVs.
- Manage user roles and permissions.
- Configure system settings.
- Delete candidate records when needed.

### User Story:

- As an admin, I want full control over candidate data so that I can maintain data integrity and ensure secure management of CVs.
- As an admin, I want to configure system settings so that all users operate within defined system policies and workflows.
- As an admin, I want to manage and remove users when necessary, so that only authorized personnel have access to the system.

### 4. Use Cases

---

#### UC1: Upload CV

- Actors: Editor, Admin
- Preconditions: User must be logged in with appropriate role; file must be in valid format (PDF/Word).
- Basic Flow:
  1. User navigates to the upload screen.
  2. User selects a CV file.
  3. System validates the file type and size.
  4. System triggers CV data extraction (UC8).
- Alternative Flows:
  - 3a. Invalid file → System shows error and aborts upload.
- Postconditions: Candidate data is stored, and extraction process is triggered.

#### UC2: View Candidates

- Actors: Viewer, Editor, Admin
- Preconditions: User must be logged in and have view permission.
- Basic Flow:
  1. User accesses the candidate list.
  2. System fetches stored candidates.
  3. System displays summarized candidate cards or detailed views.
- Postconditions: Candidate data is displayed.

### UC3: Edit Candidate Data

- Actors: Editor, Admin
- Preconditions: Users must have edit access.
- Basic Flow:
  1. User opens a candidate profile.
  2. User clicks "Edit" and makes changes.
  3. System validates inputs.
  4. Changes are saved to the database.
- Alternative Flows:
  - 3a. Validation fails → show error, user retries.
- Postconditions: Candidate data is updated and saved.

### UC4: Manage Users

- Actors: Admin
- Preconditions: Admin must be logged in.
- Basic Flow:
  1. Admin opens the User Management panel.
  2. Admin creates/edits/deletes user accounts.
  3. System confirms and saves changes.
- Postconditions: User accounts are updated.

### UC5: Configure System

- Actors: Admin
- Preconditions: Admin logged in.
- Basic Flow:
  1. Admin accesses configuration module.
  2. Admin modifies filters, models, or preferences.
  3. System validates and saves changes.
- Alternative Flows:
  - 3a. Invalid config → error message, no changes saved.
- Postconditions: System settings updated.

### UC6: Delete Records

- Actors: Admin
- Preconditions: Admin logged in, record exists.
- Basic Flow:
  1. Admin selects a candidate record.
  2. Admin confirms deletion.
  3. System deletes data.
  4. System confirms deletion.
- Alternative Flows:
  - 2a. Admin cancels → no action taken.
- Postconditions: Record is removed permanently.



### UC7: Assign Roles

- Actors: Admin
- Preconditions: Admin logged in, user account exists.
- Basic Flow:
  1. Admin accesses a user's profile.
  2. Admin selects a new role.
  3. System updates access levels.
  4. Users receive updated permissions.
- Postconditions: Roles updated.

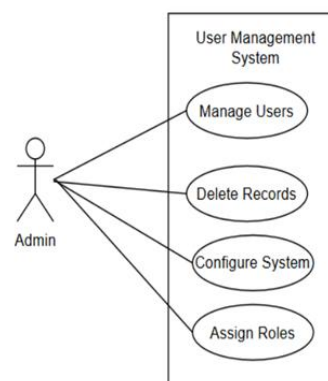
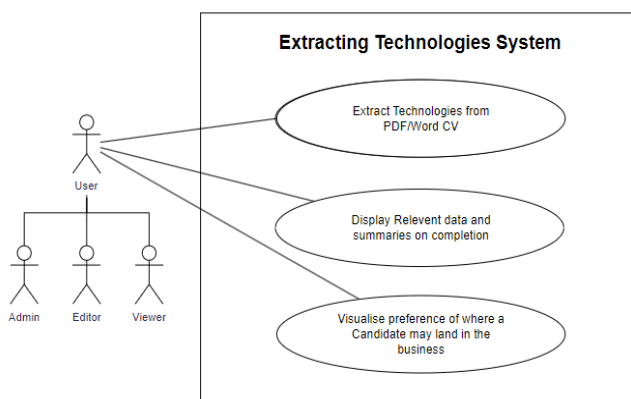
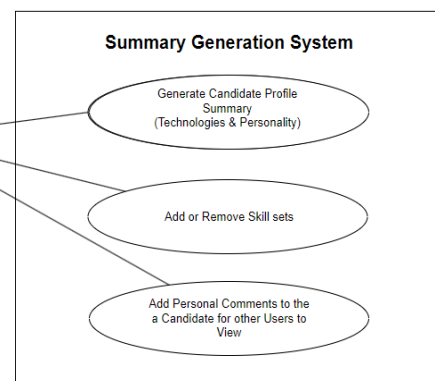
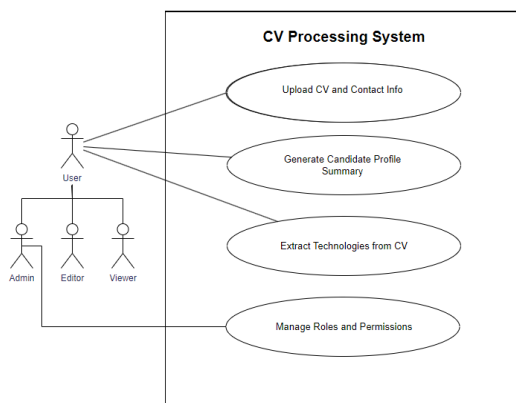
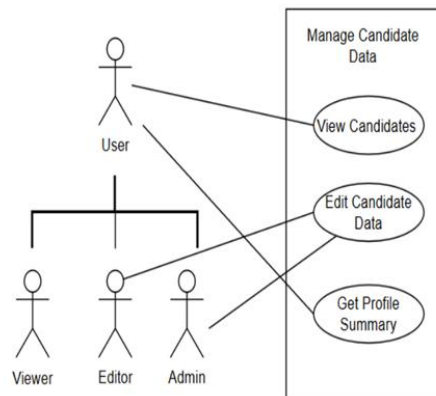
### UC8: Extract CV Data

- Actors: System (triggered by Viewer, Editor, Admin)
- Preconditions: Valid CV has been uploaded.
- Basic Flow:
  1. System reads uploaded CV.
  2. System runs an NLP/extraction module.
  3. Extracted data is structured and saved.
  4. Extraction status is updated for users.
- Postconditions: Parsed data stored in the system.

### UC9: Get Profile Summary

- Actors: Viewer, Editor, Admin
- Preconditions: Candidate data exists.
- Basic Flow:
  1. User selects a candidate.
  2. System processes profile data.
  3. System returns summary view (tech stack, traits, etc.).
- Postconditions: Summary returned.

## 5. Use Case Diagrams



## 6. Functional Requirements

---

- FR1: The system must process PDF and Word document CVs and extract relevant technologies.
  - FR1.1: Users must be able to search for a .doc, .docx or .pdf file and add it to the system.
  - FR1.2: Users must be able to convert data and display a summary of the data.
- FR2: The system must generate a summary of the candidate's profile, including technology preferences and personality traits.
  - FR2.1: Users must be able to see extracted data in an organised manner.
  - FR2.2: Users must have access to specific pages of the extracted data that they are looking for (e.g. Skills).
- FR3: The system must indicate the type of project the candidate would be most effective on (technical, collaborative, business engagement, high autonomy, etc.).
  - FR3.1: User must be able to see a how accurately the system can see how effective the user is in those project sections
- FR4: The system must save and link the extracted data and CV reference to a database that can be viewed, edited, or optimized via an admin portal.
  - FR4.1: Users must be able to review a candidate's CV
  - FR4.2: Users must be able to edit or remove candidate data when needed if they are the allowed role for it.

## CV Scanner – Software Requirements Specifications

- FR5: The system must allow admin users to specify skills to ignore and/or include specifically.
  - FR5.1: Admins must be able to select skill types based on field types and qualifications.
  - FR5.2: Admins must also have a candidate search action to find users with these specific fields and project type.
- FR6: The system must allow users to upload CVs and provide some information and contact details.
  - FR6.1: Users must be able to give a rough description and append it to the outcome of a processed CV.
  - FR6.2: Users must be able to add their credentials to show who had added the information to the CV.
- FR7: The system must allow admin users to edit existing user profiles.
  - FR7.1: Admins must be able to update user details such as name, email, and role.
  - FR7.2: The system must ensure that only valid and existing users can be edited.
- FR8: The system must allow admin users to deactivate (soft delete) users.
  - FR8.1: Admins must be able to deactivate a user by email, making them inactive without removing their data from the database.
  - FR8.2: The system must prevent deactivated users from accessing the system.

- FR9: The system must allow admin users to view all users.
  - FR9.1: Admins must be able to retrieve a list of all users, including their roles and status (active/inactive).
  - FR9.2: The system must support pagination or filtering for large user lists.
- FR10: The system must allow admin users to filter users by role.
  - FR10.1: Admins must be able to retrieve users based on their assigned role (e.g., Admin, User, Recruiter).
  - FR10.2: The system must display filtered results in an organized manner.
- FR11: The system must allow admin users to add new users to the system.
  - FR11.1: Admins must be able to specify username, email, name, role, and password when adding a user.
  - FR11.2: The system must validate that usernames and emails are unique before adding a user.

## 7. Optional Functional Requirements

---

- OFR12: The system could make use of available language models or come up with its own models to improve the accuracy of technology extraction.
  - OFR 12.1: The System can have a fuzzy search for the AI to track technologies and Candidate Data like names for more accuracy.
- OFR13: The system could include a mobile app for recruiters and other users to access the data on-the-go.
  - OFR 13.1: The system can be used in mobile use with Easier UI and scalable for phone usage.

## 8. Non-Functional Requirements

---

- NFR1: The system must be hostable on native IIS instances and Azure App Services.
- NFR2: The database must be Azure SQL DB.
- NFR3: The system must ensure secure document and user data handling.
- NFR4: The system should respond to user interactions within 2 seconds (performance).
- NFR5: The system must support at least 100 concurrent users (scalability).

## 9. Architectural Requirements Specification

---

Find the Technology Requirements and quality Requirements on our GitHub Page.

Or follow the link below:

<https://drive.google.com/file/d/1S5L43vqvFdNAobK-nL1oQAtp6S5Aj8Ql/view?usp=sharing>

## 10. Service Contracts

---

Find the Service Contracts in the Architectural requirements and design documentation on our GitHub Page.

Or follow this link below:

[https://drive.google.com/file/d/1VhFjvYFYU73nHI929ZE86LR9yMAGkc\\_y/view?usp=sharing](https://drive.google.com/file/d/1VhFjvYFYU73nHI929ZE86LR9yMAGkc_y/view?usp=sharing)

## 11. Technical Installation Manual

---

Find the Technical Installation Manual in our GitHub Page.

Or follow the link below:

[https://drive.google.com/file/d/1GQ\\_1FtPZSQDWn0eyabKHLnrVWuyNj2ny/view?usp=sharing](https://drive.google.com/file/d/1GQ_1FtPZSQDWn0eyabKHLnrVWuyNj2ny/view?usp=sharing)

## 12. User Manual

---

Find the User Manual in our GitHub Page.

Or follow the link below:

[https://drive.google.com/file/d/1Drewau\\_SXilGcay2Elxx9il7a3aqc-bnF/view?usp=sharing](https://drive.google.com/file/d/1Drewau_SXilGcay2Elxx9il7a3aqc-bnF/view?usp=sharing)

## 13. Coding Standard

---

Find the Coding Standard in our GitHub Page.

Or follow the link below:

<https://drive.google.com/file/d/1K0bpz-3rYhGe7H2q1EAscawLXG1Lyj6L/view?usp=sharing>