

# Plan-based (Waterfall): Requirement & Specification Document

## Part 1: Introduction

**Introduction:** The AI Resume Analyzer will critique, improve, and provide career recommendations based on a given Resume. This Software will initially have the User upload their Resume, after which it will be interpreted, and job positions that follow the skillsets will be listed. Selecting one of these job positions will then allow for further analysis of the Resume to flag errors/inconsistencies and provide solutions to make the resume more suitable for the desired job position.

**Goal/Purpose:** To analyze and improve resumes by identifying the strengths, weaknesses, and job matches to help users improve their resumes, specifically for job opportunities.

**Constraints:** Potential situations that could cause a hindrance to development.

- Time constraint of 4 weeks to research, develop, and deploy the Software
- Closed-source APIs/AIs will possibly require a purchase to utilize
- API Implementations will potentially cause varying results based on usage time

## Part 2: Requirements Glossary

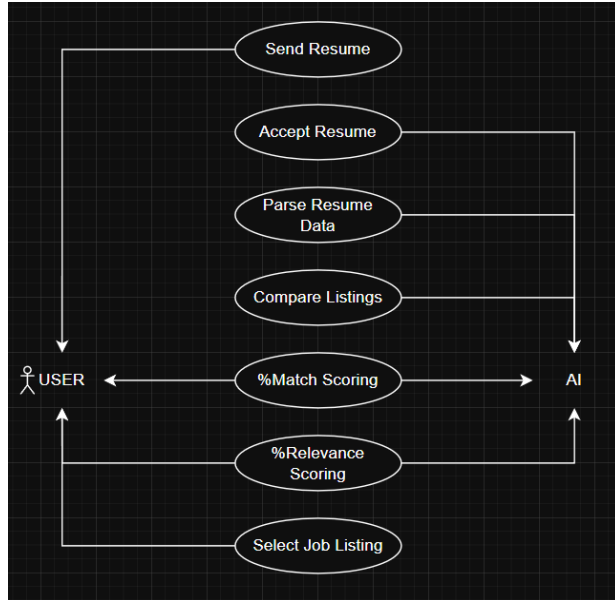
**API (Application Programming Interface):** A set of defined rules, protocols, and tools that allows different software applications to communicate and interact with each other.

**Iterations:** The repetition of a block of code until a specific condition is met or a set number of times is completed.

**Artificial Intelligence (AI):** The development of computer systems that can perform tasks that typically require human intelligence, such as reasoning, problem-solving, learning, and decision-making

## Part 3: User Requirements

## Plan-based (Waterfall): Requirement & Specification Document



- An operating system that is compatible with our application (whether it's built using an executable through Python, dmg for Mac, etc.)
- A valid application (docx, pdf, etc.)
- Access to the internet (for calling the API and job boards)

### Part 4: System Requirements

#### Functional System Requirements:

- Converting and validating the integrity of uploaded files in PDF or DOCX formats and parsing to JSON, to call the api with
- Fully resolved errors, type mismatch handling, and edge case error handling
- The system should be accurate to a certain percentage when analyzing the resumes.
- 

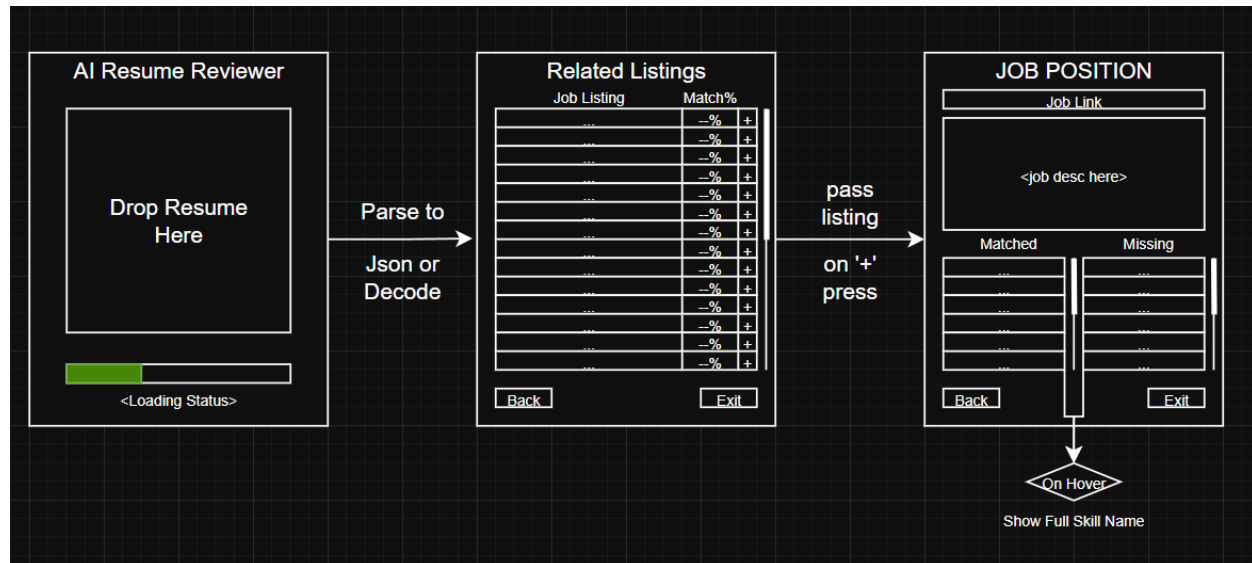
**Use case templates descriptions:** describe each possible use case

Action	Actor	System response
Upload resume(correct format)	User	Receive resume and parse information.
Upload resume(wrong format)	User	Catch wrong resume formatting and warn the user
Analyze resume	AI	Receive parsed information

## Plan-based (Waterfall): Requirement & Specification Document

		from AI
--	--	---------

### Part 5: Prototyping



### Part 6: Non-fundamental Requirements

Nonfunctional Requirements: Specific properties of the system.

#### 5 Major Nonfunctional Requirements:

- **Accuracy:** must be at least 90% accurate.
  - **Why?** Inaccurate results could lead to problems when applying for the job that was matched.
- **Compatibility:** must be open to any format of file.
  - **Why?** Users should be able to easily upload their resume without needing a specific format.
- **Reliability:** The program should not have little to no errors.
  - **Why?** User should expect the program to run without unexpected errors
- **Usability:** the UI should be easy for the user to understand and navigate without little issue
  - **Why?** For first time users, they should be able to navigate the program intuitively, otherwise the UI design is too complicated.
- **Privacy:** the user should expect any data given to be kept confidential
  - **Why?** Information from users should not be seen anywhere in any sort of shape or form.

#### **Technical requirements:**

- (Coding Language) **Python**

## Plan-based (Waterfall): Requirement & Specification Document

- (Formatting) **Tags**
- (API) **<https://cloud.google.com/talent-solution/job-search/v3/docs/basics>**

### Project Risks:

- Personal data leak
- Inaccurate API response

## Project Management:

**I = In-class Learning**

**P** = Prototyping

**W** = Documentation Drafts

## F = Developing Functionals

**N = Developing Non-Functionals**

**T** = Testing & Polishing

**D** = Deploy

[illegible]