**A purple square with a pen and a notebook

AI-generated content may be incorrect.**

**Software Requirements Specification**

Prepared By:

|  |  |  |  |
| --- | --- | --- | --- |
| **Member** | **ID#** | **Program** | **Signature** |
| Coral S. Schmidt Montilla | #148830 | Computer Science |  |
| Jaime J. Cuebas Ruiz | #121703 | Computer  Engineering |  |
| Christopher Ortiz | #130607 | Computer Science |  |
| Christian Brito | #9734 | Computer Science |  |
| Taishali N. Jimenez Quinones | #148017 | Computer Engineering |  |

**REVISION HISTORY**

The following table represents the work done by each member and the version of the SRS where the revision was made.

| **Name** | **Dates** | **Version** | **Task** |
| --- | --- | --- | --- |
| Coral S. Schmidt Montilla | 12/1/2024 | 1.0 | Restructure the document, and make it look presentable. |
| Coral S. Schmidt Montilla | 12/6/2024 | 1.1 | Insert our logo, change the name of application and update tables |
| Jaime J. Cuebas Ruiz | 12/10/2024 | 1.2 | Added User Authentication Interface and logical database for User Authentication |
| Coral S. Schmidt Montilla | 12/17/2024 | 1.3 | Filling out and fixing information in the Overall Description section and Introduction section. |
| Coral S. Schmidt Montilla | 1/4/2025 | 1.4 | Filling out and fixing information in the Overall Description section.  Fixed the reference section.  Fixed index for the tables.  Re-formatted document. |
| Taishali N. Jimenez Quinones | 01/05/2025 | 1.5 | Added the Memory Constraints and the Operations. |
| Jaime J. Cuebas Ruiz | 01/05/2025 | 1.6 | Added Cloud Storage Interface and logical database for Cloud Storage. Fixed table numbers |
| Jaime J. Cuebas Ruiz | 01/05/2025 | 1.7 | Added logical database for Job API |
| Jaime J. Cuebas Ruiz | 01/06/2025 | 1.8 | Added logical database for Template Management |
| Taishali N. Jimenez Quinones | 01/07/2025 | 1.9 | Added the Site Adaptations Requirements, Product Functions, User Characteristics, Constraints, and Assumptions and Dependencies. |
| Coral S. Schmidt Montilla | 1/11/2025 | 2.0 | Finished my part of filling out and fixing information in the Overall Description section. |
| Coral S. Schmidt Montilla | 1/16/2025 | 2.1 | Modified the purpose of this document. |
| Christian Brito | 1/22/2025 | 2.2 | Edited figure 2, 3, 4 & 5 |
| Christopher Ortiz | 1/28/2025 | 2.3 | Added Database Requirements and Mayors. |
| Coral S. Schmidt Montilla | 1/28/2025 | 2.4 | Added some finishing touches and edits. |
| Coral S. Schmidt Montilla | 4/22/2025 | 2.5 | Updated the database pictures and reviewed the material so it’s consistent with the application. |
| Coral S. Schmidt Montilla | 5/4/2025 | 2.6 | Updated logo |

**Table of Contents**

[1. Introduction 1](#_Toc196229565)

[1.1 Purpose 1](#_Toc196229566)

[1.2 Scope 1](#_Toc196229567)

[1.3 Definitions, Acronyms, and Abbreviations 2](#_Toc196229568)

[1.3.1 Acronyms 2](#_Toc196229569)

[1.3.2 Abbreviations 2](#_Toc196229570)

[1.4 Reference 3](#_Toc196229571)

[1.5 Overview 3](#_Toc196229572)

[2. Overall Description 4](#_Toc196229573)

[2.1 Product Perspective 4](#_Toc196229574)

[2.1.1 System Interfaces 6](#_Toc196229575)

[2.1.2 User Interfaces 6](#_Toc196229576)

[2.1.3 Hardware Interfaces 7](#_Toc196229577)

[2.1.4 Software Interfaces 7](#_Toc196229578)

[2.1.5 Communication Interfaces 7](#_Toc196229579)

[2.1.6 Memory Constraints 7](#_Toc196229580)

[2.1.7 Operations 8](#_Toc196229581)

[2.1.8 Site Adaptations Requirements 8](#_Toc196229582)

[2.2 Product Functions 8](#_Toc196229583)

[2.3 User Characteristics 9](#_Toc196229584)

[2.4 Constraints 9](#_Toc196229585)

[2.5 Assumptions and Dependencies 10](#_Toc196229586)

[3. Specific Requirements 11](#_Toc196229587)

[3.1 External Interface 11](#_Toc196229588)

[3.1.1 User Interface Requirements 12](#_Toc196229589)

[3.1.2 Hardware Interface Requirements 14](#_Toc196229590)

[3.1.3 Software Interface Requirements 17](#_Toc196229591)

[3.1.4 Communication Interface Requirements 21](#_Toc196229592)

[3.2 Functions 24](#_Toc196229593)

[3.2.1 Production Functions. 24](#_Toc196229594)

[3.3 Performance Requirements 25](#_Toc196229595)

[3.4 Logical Database Requirements 25](#_Toc196229596)

[3.5 Types of Information used by various functions 26](#_Toc196229597)

[3.6 Frequency of Use 28](#_Toc196229598)

[3.7 Design Constraints 28](#_Toc196229599)

[3.8 Software System Attributes 29](#_Toc196229600)

[3.8.1 Reliability 29](#_Toc196229601)

[3.8.2 Availability 29](#_Toc196229602)

[3.8.3 Security 29](#_Toc196229603)

[3.8.4 Maintainability 29](#_Toc196229604)

[3.8.5 Portability 30](#_Toc196229605)

**List of Tables**

[Table 1: Definitions 2](#_Toc196229606)

[Table 2: Acronyms 2](#_Toc196229607)

[Table 3: Abbreviations 2](#_Toc196229608)

[Table 4: Memory Constraints 7](#_Toc196229609)

[Table 5: Minimum Requirements to use the application 9](#_Toc196229610)

[Table 6: Assumptions and Dependencies 10](#_Toc196229611)

[Table 7: External Interface Table 11](#_Toc196229612)

[Table 8: Home Page 12](#_Toc196229613)

[Table 9: Registration Page 13](#_Toc196229614)

[Table 10: News 13](#_Toc196229615)

[Table 11: Specification 15](#_Toc196229616)

[Table 12: Specification 16](#_Toc196229617)

[Table 13: Specification 17](#_Toc196229618)

[Table 14: Specification 18](#_Toc196229619)

[Table 15: Specification 19](#_Toc196229620)

[Table 16: Specification 20](#_Toc196229621)

[Table 17: Specification 21](#_Toc196229622)

[Table 18: Specification 22](#_Toc196229623)

[Table 19: Specification 23](#_Toc196229624)

[Table 20: Specification 24](#_Toc196229625)

[Table 21: Functions 24](#_Toc196229626)

[Table 22: Entity and Types of Attributes 26](#_Toc196229627)

[Table 23: Entity and Types of Attributes 26](#_Toc196229628)

[Table 24: User Authentication Entity and Types of Attributes 26](#_Toc196229629)

[Table 25: Cloud Storage Entity and Types of Attributes 27](#_Toc196229630)

[Table 26: Job API Entity and Types of Attributes 27](#_Toc196229631)

[Table 27: Template Management Entity and Types of Attributes 28](#_Toc196229632)

**List of Figures**

[Figure 1: Feature Comparison between different similar apps 5](#_Toc196229633)

[Figure 2: System Interfaces 6](#_Toc196229634)

[Figure 3: User Navigation through UI Screens. 6](#_Toc196229635)

[Figure 5: Database Requirements 25](#_Toc196229636)

# 1. Introduction

## 1.1 Purpose

This document specifies the requirements to develop a resume building application. The goal is to provide an easy platform for users to build professional resumes. The application will guide the users through a series of questions, suggest sections relevant to the job that they are applying for, and offer formatting options.

This document is intended for the following audience:

* Job seekers
* Career changers
* Students
* Interns
* Anybody re-entering the job market
* Non- tech savvy users
* Busy users

## 1.2 Scope

The Resume Builder project aims to deliver a web-based application that allows users to create professional resumes with ease. The application will guide users through a series of steps to input personal, educational, and professional information, select from a range of resume templates, and generate a downloadable PDF document. This tool is targeted at job seekers, students, and professionals who require assistance in building their resumes. The project will be developed using Agile methodologies, with iterative feedback from stakeholders, ensuring that the product evolves based on user needs. Key features include template selection, resume customization, and integration with external APIs for job-specific recommendations.

## 1.3 Definitions, Acronyms, and Abbreviations

The following table illustrates the series of definitions, acronyms and abbreviations that are necessary to be aware of for proper understanding of this document:

|  |  |
| --- | --- |
| **Term** | **Definitions** |
| App | Application |
| UI | User Interface |
| Resume | Formal document that lists professional and educational track record and achievements |

Table 1: Definitions

### 1.3.1 Acronyms

The following table illustrates the series of acronyms that are necessary to be aware of for proper understanding of this document:

|  |  |  |
| --- | --- | --- |
| Acronym | Term | Definition |
| UI | User Interface | Visual part of the app that users interact with to input info |
| API | Application programming Interface | Set of functions that allow the app to communicate with external services |
| PDF | Portable Document Format | File format that allows for easy sharing and printing of documents |
| DB | Database | System that stores and manages user data and app info |

Table 2: Acronyms

### 1.3.2 Abbreviations

The following table illustrates the series of abbreviations that are necessary to be aware of for proper understanding of this document:

| **Abbreviation** | **Term** |
| --- | --- |
| Res. | Resume |
| T&C | Tips and Comments |
| App | Application |
| Rec | Recommendations |

Table 3: Abbreviations

## 1.4 Reference

This section represents the reference used for this document. The following references are cited using the IEEE format:

[1] C. Eilers, “Best Resume Templates in 2025 (10+ Picks to Use Now),” *zety*, Jan. 13, 2018. Available: <https://zety.com/blog/best-resume-templates>. [Last Accessed: Jan. 04, 2025]

This article was used to explore different resume templates and formats that are popular in today’s job market. It helped us select the templates included in the app to give users professional-looking options.

[2] K. Purcell, “How to Create an ATS-Friendly Resume in 2025,” *Jobscan*, Dec. 13, 2024. Available: <https://www.jobscan.co/blog/20-ats-friendly-resume-templates/>. [Last Accessed: Jan. 04, 2025]

This resource explains how to create resumes that can get through Applicant Tracking Systems (ATS). It influenced our decision to include tips and suggestions in the app that help users make their resumes more compatible with ATS software.

[3] “Resume Writing 101: Tips for Creating a Resume (With Examples),” *Indeed Career Guide*, 2025. Available: https://www.indeed.com/career-advice/resumes-cover-letters/resume-writing-101. [Last Accessed: Jan. 04, 2025]

This guide provided a solid foundation on how to write resumes effectively. We used it to structure the app’s step-by-step process for helping users build their resumes from scratch.

## 1.5 Overview

This Software Requirements Specification (SRS) outlines the functionality and behavior of the resume building app. It details the app’s purpose and specific requirements for the development.

# 2. Overall Description

In this section of the SRS document, we will examine the key factors influencing the system's requirements. This overview aims to enhance the user's understanding of the proposed system's architecture, operational context, and the various functions it encompasses, including software, hardware, and communication components.

## 2.1 Product Perspective

The application is designed to streamline the resume creation process through a comprehensive step-by-step guidance system. It provides customized suggestions and insights tailored to the specific job type the user is targeting, enabling the development of a polished resume with minimal effort. Unlike other applications that charge fees upon completion, this service is completely free and easily accessible, empowering users to create their resumes without facing any financial barriers.

Figure 1: Feature Comparison between different similar apps

### 2.1.1 System Interfaces

The application will interface with a range of external services to enhance the user experience. It will interact with:

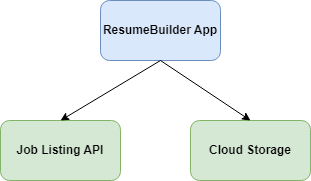
1. Job Listing API: This service retrieves job categories and relevant skills to aid in customizing resume sections.
2. Cloud Storage: This feature securely stores user resumes and other data, allowing users to access their information effortlessly across multiple devices.

Figure 2: System Interfaces

### 2.1.2 User Interfaces

The user interface is designed to be both user-friendly and intuitive. Users will navigate through the following primary screens:

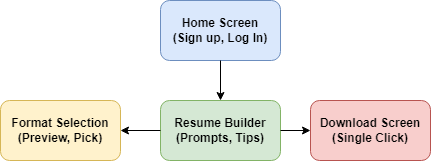
1. Home Screen: Offers options to initiate the resume-building process, create an account, or log in.
2. Resume Builder Screen: Users will enter information guided by prompts and helpful tips, which will assist them through each section. Each section will provide relevant advice tailored to the specific job type selected.
3. Format Selection Screen: Users can preview and select from a variety of resume styles.
4. Download Screen: After completing their resume, users can download it with a single click.

Figure 3: User Navigation through UI Screens.

### 2.1.3 Hardware Interfaces

Since the application is web-based, it is compatible with any device that supports a modern web browser. This encompasses:

1. Mobile Devices: Users can access the application on smartphones and tablets.
2. Desktop/Laptop Computers: The application is also fully functional on standard desktop and laptop computers.

### 2.1.4 Software Interfaces

The application will be developed using a responsive framework, such as React or Angular, to ensure seamless functionality across a variety of devices. It will rely on a back-end database, such as Firebase or MySQL, for the temporary storage of session data, as well as cloud storage for users who wish to save their resumes.

### 2.1.5 Communication Interfaces

The application is designed to enhance communication over the internet, enabling seamless interaction with external APIs and cloud storage services. Data exchanges will be secured through HTTPS, ensuring that user information remains protected throughout transmission. Additionally, when users choose to download their resume, the application will generate a file that can be saved locally.

### 2.1.6 Memory Constraints

This application is designed as a web-based application, which minimizes the need for local memory. Primary memory usage will involve temporary data storage during the resume-building process, while cloud storage will securely save progress for registered users.

| **Memory Constraints of the Application** | | |
| --- | --- | --- |
| Memory Type | Should be (At Least) | Must be (At most) |
| Temporary Storage | 5-10 MB | 50 MB |
| Cloud Storage (per user) | 10 MB | 1 GB |

Table 4: Memory Constraints

### 2.1.7 Operations

The application is designed with mechanisms that ensure smooth operation, however in the chance that there is an anomaly like internet disconnection or unexpected errors, there will be key operational behaviors that include:

* The app will temporarily store user data locally until the connection is restored
* The app will provide a clear message to inform used about incomplete fields or issues during the download process.

### 2.1.8 Site Adaptations Requirements

The application is designed to adapt seamlessly to various screen sizes and orientations. Users can access the app via any modern web browser without additional configuration.

## 2.2 Product Functions

The app provides an easy-to-use platform with several core functionalities to help users build their resumes:

 1. **User Data Input**: The app will collect essential details such as educational background and the job they are creating the resume for.

2. **Section Suggestions**: The app will then recommend relevant sections (e.g., Skills, Work Experience, Education, etc.) based on the selected job.

 3. **Tips and Descriptions**: Offers detailed tips and explanations for each section so that the user in order to enter the appropriate information.

 4. **Format Selection**: Provides users with a range of professional resume templates to choose from to give their document a polished look which will be tailored for the selected job and industry.

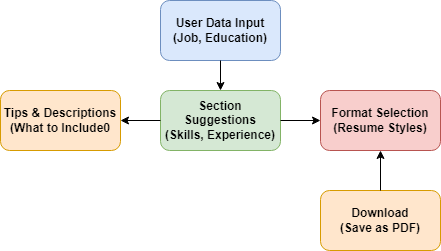
 5. **Download**: Allows users to preview and download their resume in PDF format with one click.

Figure 4: Product Function

## 2.3 User Characteristics

The app caters to a range of users, particularly individuals with minimal resume-building experience. The main clientele of users will include:

* Recent Graduates: Individuals creating their first professional resumes.
* Career Changers: Individuals needing to tailor their resume for a new industry.
* Professionals: Individuals who need help to update their resume.

It is designed to help a range of users, from recent graduates to professionals looking to change careers. The app’s intuitive interface ensures accessibility to users of varying technical skills.

## 2.4 Constraints

The application will have the following constraints:

* Browser Compatibility: The application must function seamlessly with the latest versions of popular web browsers, such as Chrome, Firefox, Safari, and Edge.
* Offline Functionality: Users should be able to continue building their resume in an offline state after initial data entry, enhancing flexibility.

| **Computers** | |
| --- | --- |
| **Operating System (OS)** | **Overall Specs** |
| * Windows, macOS, Linux * iPhone, Android | * Modern web browser (Safari, Chrome, Firefox, Edge) 4GB RAM, 500MB Storage |

Table 5: Minimum Requirements to use the application

## 2.5 Assumptions and Dependencies

The application’s functionality is dependent on assumptions and dependencies. The application will assume that the user will have an active connection to the internet as soon as they go into the application even though certain functions will be available offline once their data is entered. The application will depend on external services, like cloud storage and job-listing APIs, to enhance user experience by suggesting relevant resume sections and saving progress.

|  |  |
| --- | --- |
| **Assumptions** | **Dependencies** |
| Users will have access to internet | The application requires cloud storage and job-related APIs to offer suggestions and save data. |
| External services will remain reliable | The application functionality depends on third party APIs to enhance user experience with job specific data |
| Users have access to modern browsers. | The application relies on browser features that support responsive design and data transmission |
| Cloud services are secure and available consistently. | The integrity and security of the data relies on cloud providers. |

Table 6: Assumptions and Dependencies

# 3. Specific Requirements

In this section of the SRS document, we will be describing the specific requirements this system will have or need to function in an optimal matter with different interfaces. We will also describe any attributes or constraints that the system will be utilizing.

## 3.1 External Interface

The following table will illustrate various external interfaces of the program.

| **External Interface Table** | | | |  |
| --- | --- | --- | --- | --- |
| **Name** | User Input | Job API | Cloud storage access | Download functionality |
| **Description** | Collects data from users such as job type and education level | Collects job specific data for relevant resume sections | Saves and collects user’s resume data to and from the cloud | Allows users to download their completed resume in various formats |
| **I/O** | Input | Input/Output | Input/Output | Output |
| **Valid Range, Accuracy and Tolerance** | N/A | Accurate and updated job information | Reliable storage/encrypted | High quality PDF or other format output |
| **Units of Measure** | Text | JSON, Text | Data packets | PDF, DOCX |
| **Timing** | Immediate | < 1 second response | Consistent access | Immediate |
| **Relationship with other inputs** | Provides data for Section and Suggestions | Uses data from user input | Connects with download feature | Dependent on all completed input |

Table 7: External Interface Table

### 3.1.1 User Interface Requirements

Describe the user interfaces that are to be implemented. Below are the tables of the interface specifications subdivided by each interface.

| **Home Page** | |
| --- | --- |
| **Username** | Field for users to enter their username to log in |
| **Password** | Field for users to enter their credentials to access their account |
| **Login** | Button that users click to log in after entering their credentials |
| **Register** | Option for new users to create an account to save and access resumes |
| **Forgot your Password?** | Link that allows users to reset their password if they have forgotten it |
| **News** | Section for updates and information on new features |
| **Guide** | Provides a tutorial on how to use the app |
| **Information** | General information about the app and its purpose |
| **Contact Information** | Details on how to contact support if help is needed |

Table 8: Home Page

| **Registration Page** | |
| --- | --- |
| **Go Back** | Returns users to the previous screen without saving changes |
| **First & Last Name** | Collects user’s full name for display on their resume |
| **Date of Birth** | Optional field to include the user’s age if relevant to the resume |
| **Gender** | Optional demographic information for statistical purposes |
| **Select Town** | Allows users to specify their location for location-based job recommendations |
| **Select Condition** | Includes options to mention any conditions or preferences |
| **Weight & Height** | N/A for-resume purposes but may be used if health related info is necessary for certain job applications |
| **Email** | User’s email contact information on the resume and for app related notifications |
| **Username** | Chosen name for logging into the app (should be unique) |
| **Password & Confirm Password** | Users set a password to secure their account with confirmation to reduce mistakes |

Table 9: Registration Page

| **User Page** | |
| --- | --- |
| **Menu** | Provides access to different sections of the user page so they can navigate |
| **Update Item** | Allows users to update or edit their personal information and resume details as needed. It enables customization and keeps user data current |
| **Guide** | Offers tips and tutorials on resume building, with guidance on what to include for different job types. Helpful for first time users |
| **News** | Displays announcements or updates related to the app such as market insights, resume tips, and new features |

Table 10: News

### 3.1.2 Hardware Interface Requirements

This section defines any hardware interfaces that are to be supported by the Hardware, including logical structure, physical addresses, expected behavior, etc.

| **Specification** | | | | |
| --- | --- | --- | --- | --- |
| **Name of item** | User Device | Job API | Cloud Storage | Download Function |
| **Description of purpose** | Used by users to enter data and view the output | Job specific data suggestions for resume | Save and retrieve resume data | Download completed resume |
| **Source of input or destination of output** | Input/output | Input | Input/Output | Output |
| **Valid range, accuracy, and/or tolerance** | N/A | Accurate and current job info | Secure and reliable storage | High quality output (DOCX or PDF) |
| **Units of measure** | N/A | Text, JSON | Data Packets | PDF, DOCX |
| **Timing** | Immediate | <1 second response | Consistent access | Immediate |
| **Relationships to other inputs/outputs** | Interfaces with web application | Works with resume section suggestions | Stores user data for access across devices | Triggered after all input is completed |
| **Screen formats/organization** | Adaptive to screen size | JSON data retrieval | Cloud access screen | Standard download interface |
| **Windows format/organization** | Adaptive to screen size | Web based layout | Cloud storage interface | File compatible viewer |
| **Data Formats** | JSON, Text | JSON | Encrypted Data | PDF, DOCX |
| **Command formats** | Standard UI commands | HTTP requests | HTTP requests | Download command |
| **End messages** | “Thank you for using the app” | “Job data retrieved successfully” | “Data saved successfully” | “Resume downloaded successfully” |

Table 11: Specification

| **Specification** | | | |
| --- | --- | --- | --- |
| **Name of item** | UI screens | Tips and Descriptions | Format Selection |
| **Description of purpose** | Guides users through input, section selection, formatting, and download | Provides users with info on what to include in each resume section | Allows users to choose a resume style |
| **Source of input or destination of output** | Input/Output | Output | Input |
| **Valid range, accuracy, and/or tolerance** | N/A | Clear and concise text | Options are selectable, displayed accurately |
| **Units of measure** | N/A | Text | Visual Styles |
| **Timing** | Immediate | Immediate | Immediate |
| **Relationships to other inputs/outputs** | Linked to user device and download function | Displayed within resume builder screen | Connects with resume builder and download function |
| **Screen formats/organization** | Responsive layout | Inline with data fields | Interactive visual selection |
| **Windows format/organization** | Cross platform compatible | Formatted tips selection | Compatible with PDF/DOCX formats |
| **Data Formats** | JSON, text | Text | Style format |
| **Command formats** | UI based navigation | Text display | Style selection |
| **End messages** | “Your resume is ready to download” | “Tips displayed successfully” | “Style applied successfully” |

Table 12: Specification

### 3.1.3 Software Interface Requirements

This section defines any software interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, etc.

| **Specification** | |
| --- | --- |
| **Name of item** | Job API |
| **Description of purpose** | Provides data on different job types, which the app uses to suggest relevant sections and tips for the resume |
| **Source of input or destination of output** | Input: Job type data from external databases  Output: Relevant resume sections and job specific tips |
| **Valid range, accuracy, and/or tolerance** | API should return accurate job data and suggestions within a valid range for each industry |
| **Units of measure** | JSON, text |
| **Timing** | Must respond within 1 second |
| **Relationships to other inputs/outputs** | Works closely with the Resume Builder using job type input to give suggestions and tips |
| **Screen formats/organization** | Integrated within the Resume Builder screen, displaying suggestions based on the job selected |
| **Windows format/organization** | Responsive web layout for desktop and mobile views |
| **Data Formats** | Encrypted data for security |
| **Command formats** | HTTP POST for saving and HTTP GET for retrieving data |
| **End messages** | “Data saved successfully” |

Table 13: Specification

|  |  |
| --- | --- |
| **Specification** | |
| **Name of item** | Tips and suggestions interface |
| **Description of purpose** | Provides users with tips on what to include in each section of their resume, helping guide them through the resume-building process |
| **Source of input or destination of output** | Input: Pulls data from job API and user’s selected job type  Output: Displays suggestions based on job type and section selection. |
| **Valid range, accuracy, and/or tolerance** | Must display accurate and relevant tips based on selected job type and section with no delay |
| **Units of measure** | Text |
| **Timing** | ≤ 1 second (Immediate) |
| **Relationships to other inputs/outputs** | Interacts with Resume Builder screen, using the job type input to display relevant tips for each resume section |
| **Screen formats/organization** | Text boxes or tools beside each section, expanding when clicked or hovered over |
| **Windows format/organization** | Designed to be compatible across desktop and mobile screen |
| **Data Formats** | JSON, Text |
| **Command formats** | Tool tips or pop-up message triggered by hover/click |
| **End messages** | “Tips displayed successfully” |

Table 14: Specification

|  |  |
| --- | --- |
| **Specification** | |
| **Name of item** | User Authentication Interface |
| **Description of purpose** | Provides secure login, registration, and session management. |
| **Source of input or destination of output** | Input: User info  Output: authentication token, login success or error message. |
| **Valid range, accuracy, and / or tolerance** | Email: must be in a valid email format.  Password: at least 8 characters, with letter, numbers and symbols.  Authentication token: |
| **Units of measure** | Session validity |
| **Timing** | Login response time < 2 seconds. Session expiration based on inactivity (15 min) |
| **Relationships to other inputs / outputs** | Inputs are linked to secure backend validation. Outputs are linked to protected routes and user session states. (active or inactive) |
| **Screen formats / organization** | User input fields with login and register buttons. Error messages displayed if not successful. |
| **Windows format / organization** | Organized with responsive layouts. |
| **Data Formats** | Email: string  Password: encrypted string (hashed).  Token: JSON Web Token. |
| **Command formats** | HTTP POST request to authentication endpoints. |
| **End messages** | Success: “Login successful” or “Account created successfully”  Error: “Invalid email”, “Email already exists” or “Password must have...” |

Table 15: Specification

|  |  |
| --- | --- |
| **Specification** | |
| **Name of item** | Cloud Storage Interface |
| **Description of purpose** | Enables users to store resumes securely in the cloud, ensuring accessibility and data backup. |
| **Source of input or destination of output** | Input: Resume File, Template file and Metadata (file name, description etc...)  Output: Saved file, file link, resume preview, download module. |
| **Valid range, accuracy, and/or tolerance** | File types: .pdf and .docx.  File size: max 10MB (example) |
| **Units of measure** | File size in MB. |
| **Timing** | Real-time upload/download within 2-5 seconds per MB (depends on network condition) |
| **Relationships to other inputs/outputs** | Input: User’s resume data  Output: downloadable link for the resume. |
| **Screen formats/organization** | File upload area, File list display, storage usage indicator, folders for organization, search and filter and responsive design. |
| **Windows format / organization** | Organized by resume type and creation date.  Resumes displayed as a list with metadata (name, size, type, last modified) and actions (delete, download) |
| **Data Formats** | JSON for metadata storage, binary for file storage. |
| **Command formats** | Upload: POST request with file payload.  Retrieve: GET request with file ID. |
| **End messages** | Success: “File uploaded successfully”  Error: “Upload failed, please try again.” |

Table 16: Specification

### 3.1.4 Communication Interface Requirements

Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, etc.

| **Specifications** | |
| --- | --- |
| **Name of item** | User Input Interface |
| **Description of purpose** | Allows user to enter their personal information, education background, and job preferences directly into the app |
| **Source of input or destination of output** | Input from the user then it is processed by the app |
| **Valid range, accuracy, and/or tolerance** | Input should be expected format with validation to ensure accuracy |
| **Units of measure** | Text, Numeric |
| **Timing** | Real time processing as user types or submits information |
| **Relationships to other inputs/outputs** | Connects directly to the app, feeding data into other components |
| **Screen formats/organization** | Form fields organized by categories like personal information, education, and job preferences |
| **Windows format/organization** | Responsive web form accessible across different devices |
| **Data Formats** | JSON, Text |
| **Command formats** | Form submission commands (next, save) |
| **End messages** | “Your information has been saved” |

Table 17: Specification

| **Specifications** | |
| --- | --- |
| **Name of item** | Section Suggestions Interface |
| **Description of purpose** | Uses the job type and other user inputs to suggest relevant sections for the resume (Skills, Experience, Education) |
| **Source of input or destination of output** | Input: User’s job type and education level  Output: Suggested resume sections |
| **Valid range, accuracy, and/or tolerance** | Accurate section suggestions based on job type |
| **Units of measure** | Text |
| **Timing** | Immediate |
| **Relationships to other inputs/outputs** | Pulls data from user input and job API, integrates with app and display suggestions |
| **Screen formats/organization** | Shows suggestions and the user progresses |
| **Windows format/organization** | Web layout that adapts to different screen sizes |
| **Data Formats** | JSON |
| **Command formats** | Selection commands (“Add sections”) |
| **End messages** | “Section added to your resume” |

Table 18: Specification

|  |  |
| --- | --- |
| **Specification** | |
| **Name of item** | Template management interface |
| **Description of purpose** | Allows users to choose from various professional resume styles to apply to their document |
| **Source of input or destination of output** | Input: User selected style  Output: Preview and application of the format selected |
| **Valid range, accuracy, and/or tolerance** | Style previews reflect the final output format |
| **Units of measure** | Visual styles |
| **Timing** | Immediate |
| **Relationships to other inputs/outputs** | Interacts with app to apply the style and download for file formatting |
| **Screen formats/organization** | Style thumbnail for preview and selection |
| **Windows format/organization** | Fully responsive. Allows users to preview and select styled on desktop and mobile |
| **Data Formats** | CSS templates |
| **Command formats** | Selection commands |
| **End messages** | “Style applied to your resume” |

Table 19: Specification

| **Specification** | |
| --- | --- |
| **Name of item** | Download Functionality interface |
| **Description of purpose** | Allows users to download their completed resume in PDF or DOCX |
| **Source of input or destination of output** | Output: Saves the resume to the user’s device in the chosen format |
| **Valid range, accuracy, and/or tolerance** | File output should be high quality reflecting the selected style and format |
| **Units of measure** | PDF, DOCX |
| **Timing** | 5 seconds or less |
| **Relationships to other inputs/outputs** | Uses data from the app and format selection to create downloadable file |
| **Screen formats/organization** | Download button on final screen |
| **Windows format/organization** | Opens standard file save dialog on user’s device |
| **Data Formats** | PDF, DOCX |
| **Command formats** | Download button |
| **End messages** | “Your resume has been downloaded successfully” |

Table 20: Specification

## 3.2 Functions

This section will provide the details of the diagrams made in section 3.1.4

### 3.2.1 Production Functions.

It will specify in each function the primary actors and secondary actors and what the function is made by doing so.

|  |  |  |
| --- | --- | --- |
| Function Name | Primary Actor | Secondary Actor |
| User Data Input | User | Web application |
| Section Suggestions | Web Application | User |
| Tips and Descriptions | Web application | User |
| Format Selection | User | Web application |
| Download Resume | User | Cloud Storage |

Table 21: Functions

Function Details:

 1. User Data Input: Users enter personal data like education and job preferences, which the app uses to personalize resume sections.

 2. Section Suggestions: Based on the job type entered, the app suggests relevant resume sections like Work Experience, Skills, and Education.

 3. Tips and Descriptions: The app provides users with hints and suggestions for each section, making it easier to know what to include.

 4. Format Selection: Users can preview and select a resume style to give their document a professional look.

 5. Download Resume: Once complete, users can download their resume as a PDF or DOCX file, stored in the cloud for future access

## 3.3 Performance Requirements

The app should be responsive and efficient providing a smooth experience for users:

* 1. Each function (data input, section suggestions, etc.) should respond within 2 seconds.
  2. The resume download feature should allow users to save their resume in under 5 seconds after selection.
  3. The app should handle at least 100 simultaneous users without a drop in performance.

## 3.4 Logical Database Requirements

A computer screen shot of a computer

AI-generated content may be incorrect.The app’s database needs to store user information securely and allow quick access. It should include tables for user profiles, resume sections, format styles, and download logs. This data is essential for personalizing the user experience and enabling resume retrieval.

Figure 5: Database Requirements

## 3.5 Types of Information used by various functions

| Table | |
| --- | --- |
| **Attribute** | **Type** |
| User ID | Integer |
| Username | String |
| Email | String |
| Education Level | String |
| Job type | String |
| Resume Format | String |
| Download Status | Bool |

Table 22: Entity and Types of Attributes

|  |  |
| --- | --- |
| Table | |
| **Attribute** | **Type** |
| Resume ID | Integer |
| User ID | Integer |
| Job Type | String |
| Education Level | String |
| Selected Format | String |
| Created At | Date time |
| Last updated | Date time |

Table 23: Entity and Types of Attributes

|  |  |
| --- | --- |
| Table | |
| **Attribute** | **Type** |
| Email | String |
| Password | String |
| Token | String |

Table 24: User Authentication Entity and Types of Attributes

|  |  |
| --- | --- |
| Table | |
| **Attribute** | **Type** |
| File ID | String |
| User ID | String |
| File Name | String |
| File Type | String |
| File Size | Integer |
| File Path | String |
| Upload Date | Date time |
| Last Modified Date | Date time |
| Folder Name | String |
| Tags | String |
| Storage Status | String |

Table 25: Cloud Storage Entity and Types of Attributes

|  |  |
| --- | --- |
| Table | |
| **Attribute** | **Type** |
| Job ID | String |
| Title | String |
| Company | String |
| Location | String |
| Salary Range | String |
| Description | Text |
| Job URL | String |
| Source | String |
| Posted Date | Date time |
| Application Status | String |
| User ID | String |

Table 26: Job API Entity and Types of Attributes

|  |  |
| --- | --- |
| Table | |
| **Attribute** | **Type** |
| Template ID | String |
| Template Name | String |
| Description | Text |
| Preview URL | String |
| Style File URL | String |
| Category | String |
| Is Active | Boolean |
| Last Synced | Date time |
| User Template ID | String |
| User ID | String |
| Customizations | JSON |
| Last Used Date | Date time |

Table 27: Template Management Entity and Types of Attributes

## 3.6 Frequency of Use

This app is likely going to be used a lot by people who need a quick and easy way to make or update their resumes. It’s designed for anyone from students just starting out to professionals looking for a career change. Users might use it a bunch of times over a few days while getting their resume just right or pop back in now and then to make updates.

## 3.7 Design Constraints

The design of the resume-building app has a few key limitations to keep in mind:

1. Platform Compatibility: The app needs to work smoothly on both desktops and mobile devices, specifically on popular browsers like Chrome, Firefox, Safari, and Edge.
2. Data Security: Since people are entering personal info, we must make sure all data is encrypted and stored safely, following the latest data protection rules.
3. User-Friendly Interface: The app should be super easy to navigate, especially for people who might not have a lot of experience building resumes.
4. Performance Efficiency: The app should be fast and responsive. No one likes to wait around, so it needs to load quickly and handle user inputs without any lag.

## 3.8 Software System Attributes

This section specifies the attributes of the software that can serve as requirements.

### 3.8.1 Reliability

The app should be reliable, meaning it should rarely crash or experience downtime. It needs to handle lots of users at once and make sure their data is saved as they go, so they don’t lose progress if something goes wrong. To help with this, the app will have built-in error-checking to keep everything accurate.

### 3.8.2 Availability

This app should be available basically all the time, with any downtime kept to a minimum for updates and maintenance. Because users might be working on their resumes at different times, the app should be accessible 24/7. Using cloud storage will also help users access their saved resumes from any device. 

### 3.8.3 Security

We’ve got to make sure user data is protected, so the app will use strong encryption both when data is sent over the internet and when it’s stored. Secure login and regular security updates will keep the app safe from potential threats. User data will be stored in the cloud, following privacy laws to make sure everything stays secure.

### 3.8.4 Maintainability

The app should be easy to update. By keeping the code organized and modular, future updates or fixes won’t require a complete overhaul. This means we can add new features or change things up based on user feedback without breaking everything. The app will also be flexible to handle updates to any external APIs we use.

### 3.8.5 Portability

The app needs to work on pretty much any device that can run a web browser, whether that’s on Windows, Linux, macOS, iOS, or Android. It should automatically adjust to different screen sizes, so users can have a good experience whether they’re on a computer, tablet, or phone. This version presents the details in a way that’s clear and straightforward, making it easier to understand without all the formal language.