

Polytechnic University of Puerto Rico

Electrical & Computer Engineering and Computer Science Department

Winter 2021

**CECS 4204-80 Software Engineering**

**Prof. Carlos M. De La Torre Ugarte**

**Software Requirements Specification**

Latest Revision:

[datexxx]

**Version 1.0**

For:

[xxx]

Prepared By:

|  |  |  |  |
| --- | --- | --- | --- |
| **Member** | **ID#** | **Program** | **Signature** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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** |
| --- | --- | --- | --- |
| [xxx] | [xxx] | [xxx] | [xxx] |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1. Introduction 1](#_Toc120563885)

[1.1 Purpose 1](#_Toc120563886)

[1.2 Scope 1](#_Toc120563887)

[1.3 Definitions, Acronyms, and Abbreviations 1](#_Toc120563888)

[1.3.1 Acronyms 2](#_Toc120563889)

[1.3.2 Abbreviations 2](#_Toc120563890)

[1.4 Reference 2](#_Toc120563891)

[1.5 Overview 2](#_Toc120563892)

[2. Overall Description 3](#_Toc120563893)

[2.1 Product Perspective 3](#_Toc120563894)

[2.1.1 System Interfaces 3](#_Toc120563895)

[2.1.2 User Interfaces 3](#_Toc120563896)

[2.1.3 Hardware Interfaces 3](#_Toc120563897)

[2.1.4 Software Interfaces 3](#_Toc120563898)

[2.1.5 Communication Interfaces 3](#_Toc120563899)

[2.1.6 Memory Constraints 4](#_Toc120563900)

[2.1.7 Operations 4](#_Toc120563901)

[2.1.8 Site Adaptations Requirements 4](#_Toc120563902)

[2.2 Product Functions 4](#_Toc120563903)

[2.3 User Characteristics 4](#_Toc120563904)

[2.4 Constraints 4](#_Toc120563905)

[2.5 Assumptions and Dependencies 5](#_Toc120563906)

[3. Specific Requirements 5](#_Toc120563907)

[3.1 External Interface 5](#_Toc120563908)

[3.1.1 User Interface Requirements 6](#_Toc120563909)

[3.1.2 Hardware Interface Requirements 7](#_Toc120563910)

[3.1.3 Software Interface Requirements 9](#_Toc120563911)

[3.1.4 Communication Interface Requirements 10](#_Toc120563912)

[3.2 Functions 13](#_Toc120563913)

[3.3 Performance Requirements 14](#_Toc120563914)

[3.4 Logical Database Requirements 14](#_Toc120563915)

[3.5 Design Constraints 15](#_Toc120563916)

[3.6 Software System Attributes 15](#_Toc120563917)

[3.6.1 Reliability 15](#_Toc120563918)

[3.6.2 Availability 15](#_Toc120563919)

[3.6.3 Security 15](#_Toc120563920)

[3.6.4 Maintainability 15](#_Toc120563921)

[[xxx] 15](#_Toc120563922)

[3.6.5 Portability 15](#_Toc120563923)

**List of Tables**

[Table 1: Definitions 1](#_Toc120563924)

[Table 2: Acronyms 2](#_Toc120563925)

[Table 3: Abbreviations 2](#_Toc120563926)

[Table 4: Feature Comparison between different similar apps 3](#_Toc120563927)

[Table 5: Memory Constraints 4](#_Toc120563928)

[Table 6: Minimum Requirements to use the application 5](#_Toc120563929)

[Table 7: Assumptions and Dependencies 5](#_Toc120563930)

[Table 8: External Interface Table 6](#_Toc120563931)

[Table 9: [xxx] Home Page 6](#_Toc120563932)

[Table 10: [xxx] Registration Page 7](#_Toc120563933)

[Table 12: [xxx] News 7](#_Toc120563934)

[Table 17: [xxx] Specification 8](#_Toc120563935)

[Table 18: [xxx] Specification 9](#_Toc120563936)

[Table 19: [xxx] Specification 10](#_Toc120563937)

[Table 20: [xxx] Specification 10](#_Toc120563938)

[Table 21: [xxx] Specification 11](#_Toc120563939)

[Table 22: [xxx] Specification 12](#_Toc120563940)

[Table 23: [xxx] Specification 12](#_Toc120563941)

[Table 24: [xxx] Specification 13](#_Toc120563942)

[Table 25: Functions 13](#_Toc120563943)

[Table 26: Users Entity and Types of Attributes 14](#_Toc120563944)

[Table 27: [xxx] Entity and Type of Attributes 14](#_Toc120563945)

**List of Figures**

[Figure 1: System Interfaces 3](#_Toc120480448)

[Figure 2: 3](#_Toc120480449)

[[xxx] 4](#_Toc120480450)

[Figure 18: Product Function 4](#_Toc120480451)

[[xxx] 14](#_Toc120480452)

[Figure 19: Database Tables 14](#_Toc120480453)

# 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 building app will help the users complete a personalized resume by:

* Collecting user information
* Offering resume formats and enabling downloads
* Suggesting relevant sections with tips as to what to include

## 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 list professional and educational track record and achievements |

### 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] “Best Resume Templates,” Zety Resume Builder.

[https://zety.com/blog/best-resume-templates](https://urldefense.com/v3/__https:/zety.com/blog/best-resume-templates__;!!Nkc5UzxO!oQx3gV9wL_P1k9XvMyM-nc5M3S0mjtRmTbxM6bqz_hkYAz8SdoPTyg0_j8YrvyHjdkDdLQPCYe6U1DZCwO2KGDE$)

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] “20 ATS-Friendly Resume Templates,” Jobscan.

[https://www.jobscan.co/blog/20-ats-friendly-resume-templates/](https://urldefense.com/v3/__https:/www.jobscan.co/blog/20-ats-friendly-resume-templates/__;!!Nkc5UzxO!rz0ubdQj04N2aQ2WH2C88MZDABHJU0SuIBjgBi4wKQud7UFvOXNDZeCqAu0A6psqRX0ezbTZ0xw-LSxOz1HpqVQ$)

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: How to Write a Resume,” Indeed Career Guide. <https://www.indeed.com/career-advice/resumes-cover-letters/resume-writing-101>

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, the general factors that affect the requirements of the system will be given. It is intended for the user’s audience to give them a better understanding of what is going to be built in the proposed system, how the proposed system will work in context of the software, hardware, and communication, as well as the number of functions that compose it.

## 2.1 Product Perspective

The app is intended to reduce the complexity of resume creation by guiding the users step by step. It gives suggestions and tips based on the type of job the user is applying for, helping them create a professional resume with less effort.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| User Input | Users provide their education and job type |
| Section Suggestions | The app recommends which resume sections to include |
| Download | Users choose a style and download their resume |

Table 4: Feature Comparison between different similar apps

### 2.1.1 System Interfaces

The app will connect with multiple external services to enhance the user experience. It will communicate with:

1. **Job Listing API:** Retrieves job types and relevant skills to help customize resume sections.
2. **Cloud Storage**: Saves user resumes and other data, allowing users to access their information across devices.

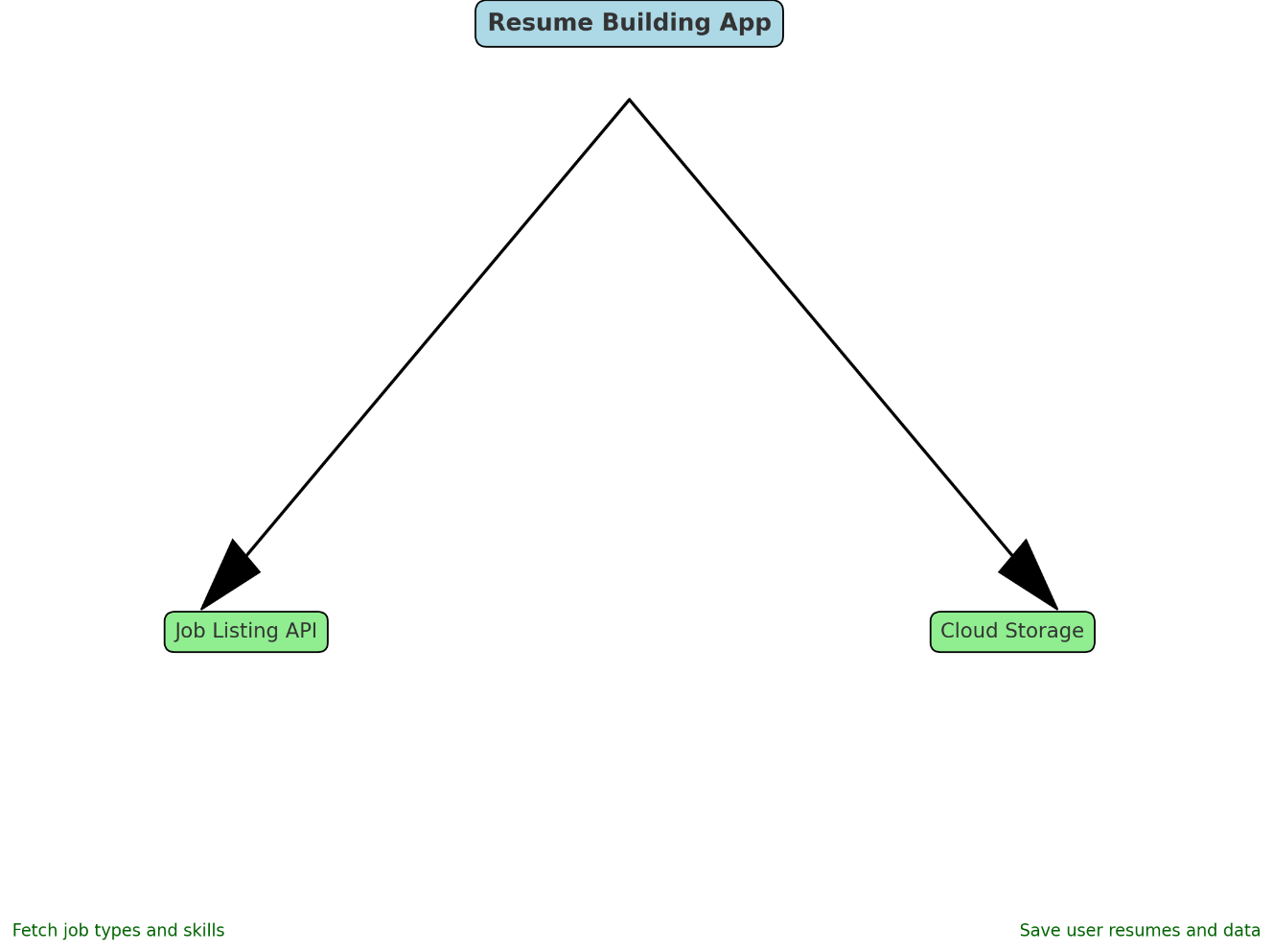


Figure 1: System Interfaces

### User Interfaces

The user interface is designed to be simple and intuitive. Users will navigate through the following main screens:

1. **Home Screen**: Options to start building a resume, sign up, or log in.
2. **Resume Builder Screen**: Users will enter information with prompts and tips displayed to guide them through each section. Each section will show relevant tips based on the job type selected.
3. **Format Selection Screen**: Users can preview and pick from different resume styles.
4. **Download Screen**: After completing their resume, users can download it with one click.

A diagram of a user navigation

Description automatically generated

Figure 2: User Navigation through UI Screens.

### Hardware Interfaces

Since the app is web-based, it will run on any device that supports a modern web browser. This includes:

1. **Mobile Devices**: Users can access the app on smartphones or tablets.
2. **Desktop/Laptop Computers**: The app will also work on standard desktop and laptop computers

### 2.1.4 Software Interfaces

The app will be developed using a responsive framework, such as React or Angular, to ensure it works seamlessly across different devices. It will rely on a back-end database (such as Firebase or MySQL) for temporary storage of session data and cloud storage for users who want to save their resumes.

### 2.1.5 Communication Interfaces

The app will communicate over the internet to interact with external APIs and cloud storage services. Data exchanges will be secured with HTTPS to ensure that user data is protected during transmission. Additionally, when users choose to download their resume, the app will generate a file that users can save locally.

### 2.1.6 Memory Constraints

The app will not require significant memory as it is a web-based application. The main memory use will be for temporary data storage while the resume is being built. For users who log in and save their resumes, a small amount of storage will be needed on the cloud to retain resume data.

| **Memory Constraints of the Application** | | |
| --- | --- | --- |
| Memory Type | Should be (At Least) | Must be (At most) |
| Temporary Storage | 10 MB | 100 MB |
| Cloud Storage (per user) | 10 MB | 1 GB |

Table 5: Memory Constraints

### 2.1.7 Operations

In case of any anomaly, the app will operate as follows:

In the event of any errors (loss of internet connection, etc.), the app will temporarily store user data locally until the connection is restored, ensuring users don’t lose progress. The app will also provide a message to the user if any required fields are missing or if there is a problem with the download process.

### 2.1.8 Site Adaptations Requirements

The app is designed to be adaptive, so it will automatically adjust to various screen sizes and orientations. Users can access the app through any modern web browser without additional setup. No site-specific customization is necessary for the app to work.

## 2.2 Product Functions

The app includes several main functions to help users easily build their resumes:

 1. **User Data Input**: Users enter their educational background and the type of job they are applying for.

2. **Section Suggestions**: Based on job type, the app suggests which resume sections to include (Skills, Work Experience, Education).

 3. **Tips and Descriptions**: Each section has tips on what kind of information is expected, helping users understand what to include.

 4. **Format Selection**: Users can pick from various professionally designed resume formats to give their document a polished look.

 5. **Download**: Once complete, users can download the resume with one click, saving it as a PDF or other file format.

A diagram of a job application

Description automatically generated with medium confidence

Figure 18: Product Function

## 2.3 User Characteristics

The app targets users with minimal experience in resume building. It is designed to help a range of users, from recent graduates to professionals looking to change careers. The app’s simplicity makes it accessible to people of all tech skill levels.

## 2.4 Constraints

1. Browser Compatibility: The app must work with the latest versions of popular web browsers, such as Chrome, Firefox, Safari, and Edge.
2. Offline Functionality: Users should be able to complete their resume in an offline state after initial data entry, allowing flexibility in usage.

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

Table 6: Minimum Requirements to use the application

## 2.5 Assumptions and Dependencies

The app assumes that users will start with an internet connection to access all the features, even though certain functions will be available offline once their data is entered. Additionally, the app depends 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 | App needs access to the cloud storage and job-related APIs to offer suggestions and save data |
| External services will be consistently available | App functionality depends on third party APIs to enhance user experience with job specific data |
| Users will have access to modern web browsers | App relies on browser features that support responsive design and data transmission |
| Reliable and secure cloud storage | User data is stored in the cloud to allow access across devices. |

Table 7: 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 its 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 8: 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 sign 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 the 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 9: 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 particular 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 10: 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 12: 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 17: 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 18: 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  Ouput: 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 19: 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** | Output: Displays suggestions based on job type and section selection.  Input: Pulls data from job API and user’s selected job type |
| **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 20: 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 21: 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 22: Specification

|  |  |
| --- | --- |
| **Specification** | |
| **Name of item** | Format selection 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 23: 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 24: Specification

### 3.2 Functions

This section will provide the details of the diagrams made in section 2.2 Production Functions. It will specify in each function the primary actors and secondary actors and what is the function 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 25: 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

 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 19: Database Tables

* + **Types of Information used by various functions**

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

Table 26: Users Entity and Types of Attributes

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

Table *27: Entity and Type of Attributes*

* + **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.

# 3.5 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

# 3.6 Software System Attributes

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

### 3.6.1 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.

### 3.6.2 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. 

### 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.

### 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.

### 3.6.5 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. Let me know if you need more adjustments!