

SE 3XA3: Software Requirements
Specification
The Resume Shotgun

Team 5, Proper Mars Tribe
Gavin Jameson, jamesong
Jeremy Langner, langnerj
Sam Gorman, gormans

February 10, 2022

Contents

1	Project Drivers	1
1.1	The Purpose of the Project	1
1.2	The Stakeholders	1
1.2.1	The Client	1
1.2.2	The Customers	1
1.2.3	Other Stakeholders	2
1.3	Mandated Constraints	2
1.4	Naming Conventions and Terminology	2
1.5	Relevant Facts and Assumptions	3
2	Functional Requirements	4
2.1	The Scope of the Work and the Product	4
2.1.1	The Context of the Work	4
2.1.2	Work Partitioning	5
2.1.3	Individual Product Use Cases	7
2.2	Functional Requirements	7
3	Non-functional Requirements	9
3.1	Look and Feel Requirements	9
3.2	Usability and Humanity Requirements	10
3.3	Performance Requirements	10
3.4	Operational and Environmental Requirements	10
3.5	Maintainability and Support Requirements	11
3.6	Security Requirements	11
3.7	Cultural Requirements	11
3.8	Legal Requirements	11
3.9	Health and Safety Requirements	12
4	Project Issues	12
4.1	Open Issues	12
4.2	Off-the-Shelf Solutions	12
4.3	New Problems	12
4.4	Tasks	12
4.5	Migration to the New Product	13
4.6	Risks	13
4.7	Costs	13

4.8	User Documentation and Training	13
4.9	Waiting Room	13
4.10	Ideas for Solutions	14
5	Appendix	15
5.1	Symbolic Parameters	15

List of Tables

1	Work Partitioning Business Events	6
2	Revision History	16

List of Figures

1	Work Context Diagram	4
2	Use Case Diagram	7

This document describes the requirements for The Resume Shotgun. The template for the Software Requirements Specification (SRS) is a subset of the Volere template (?).

1 Project Drivers

1.1 The Purpose of the Project

The modern job application process typically follows the same steps; applicants find potential employer's postings on a job board, determine if they think it's a good job, decide if they are a qualified candidate, and finally apply. This process is extremely tedious and mundane for most since it requires a substantial amount of reading through similar postings and similar processes for each. The purpose of this project is to create a functioning application that allows the user to efficiently apply for multiple jobs on a variety of job board websites. The application utilizes user input from a resume to effectively match the user's experience/characteristics with potential job application descriptions. Once potential posting(s) are found, the user may confirm to submit an application for such position(s). The application will also allow for the user to upload auxiliary documents, such as cover letters and resumes. Multiple versions will be able to be uploaded in an attempt to tailor submissions for each job based on extracted key phrases.

1.2 The Stakeholders

1.2.1 The Client

The clients for this project, Dr Bokhari and the TA's for 3XA3, will be actively monitoring the project during its development and assuring that it follows their specifications.

1.2.2 The Customers

The customers for this project consist of people searching for a job, specifically students looking to enter the workforce. The product suits the needs of these customers by casting a wide net and applying to all relevant positions. For the target customer base, the main issue with job searching is getting

an interview for any relevant job, rather than the quality of the jobs applied for. This is the issue that the Resume Shotgun aims to solve.

1.2.3 Other Stakeholders

Using a template to apply to a job will inevitably make it less personal and less polished than it could be if one was to take the time to apply to jobs individually. Therefore, companies offering positions hold stakes in this project as the quality of the applications may go down due to using a more generalized template.

1.3 Mandated Constraints

- **Description:** The product will run on modern machines with operating systems capable of executing Python code and installing required libraries.

Rationale: Python is a popular language with lots of documentation and support for any installing or execution issues the user may encounter.

Fit Criterion: Python supports all current and planned features, ensuring the product will be functional to the user.

- **Description:** The product will run on any of the commonly used browsers (Chrome, Firefox, Safari).

Rationale: There are many widely used browsers with their own benefits/consequences. Allowing support for a variety of browsers lets the user choose their desired browser based on their own preferences and needs.

Fit Criterion: The product should require desired browser to be declared by the user to ensure proper functionality.

1.4 Naming Conventions and Terminology

Some relevant naming conventions that will be used in this SRS and overall project are as follows:

- **OS:** An operating system which is the integral software that binds the hardware to user capabilities.

- Windows: The operating system Microsoft Windows.
- GUI: Graphical User Interface, which is what the user will be using to interact with the program.
- PDF: Portable Document Format, a versatile format for text documents. Most basic text documents can be converted to this form.
- Python: The programming language that the program will be built in.
- Unittest: A module used to test python projects.
- Browser: The internet browser which the tool will use to access job boards. Common examples include Chrome and Firefox.

In addition to this, common terminology will be used within the application's documentation:

- "Our Team" will refer to Gavin Jameson, Jeremy Langner and Sam Gorman, the group working on the application.
- "Script" will refer to the portion of the product constructed in Python.

1.5 Relevant Facts and Assumptions

Relevant Facts

- This project will be written entirely in Python.

Assumptions

- The user must be qualified enough to be accepted to the jobs they are searching for; the success of this project for any individual depends on the credentials of the user. In other words, this project merely increases the amount of chances to get an offer, not the chance of an offer itself ($100 \times 0\% = 0\%$).
- The user must have a modern machine with an operating system capable of supporting modern web browsers (Chrome, Safari, Firefox). The operating system must also be able to support Python 3 and all relevant libraries.

- The user has a familiarity with running command line Python scripts or is competent enough to troubleshoot any issues that they may encounter.
- The user must have a resume (and any other relevant files) in PDF format on their local machine to upload.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

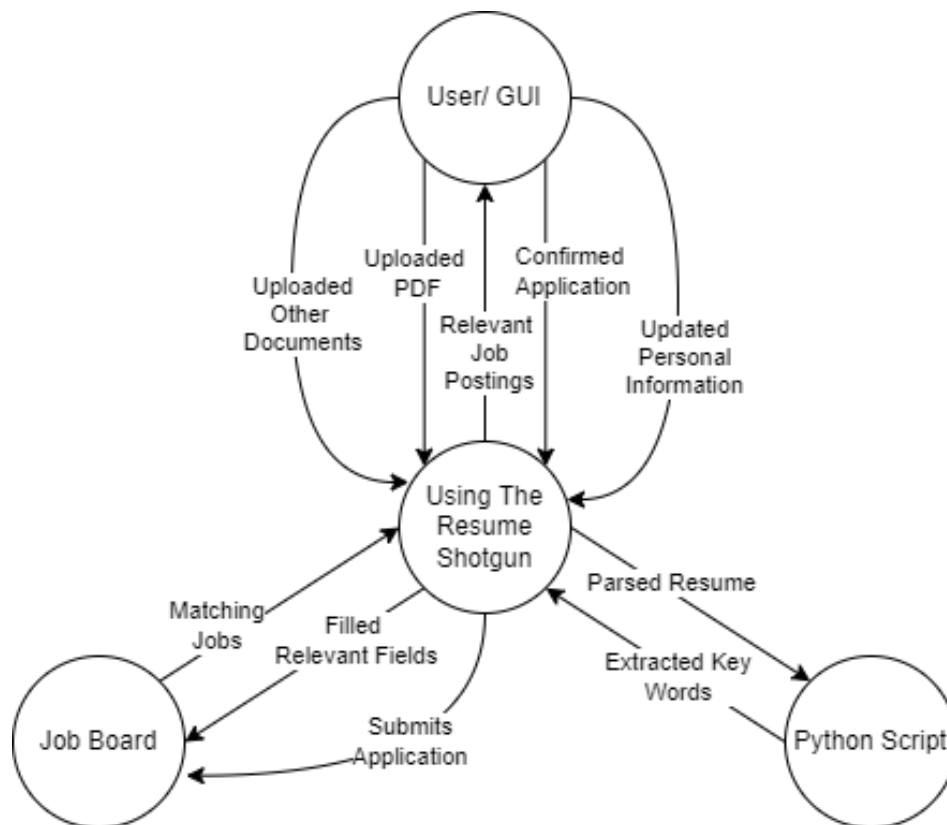


Figure 1: Work Context Diagram

2.1.2 Work Partitioning

#	Event Name	Input/Output	Summary
1	User uploads resume	PDF Document (in) List of Keywords (out)	The user uploads their Resume in the form of a PDF and the system uses this information to extract and construct a list of key words/phrases.
2	User uploads auxiliary document	PDF Document (in)	The user uploads their cover letter, transcript, or similar document to be stored.
3	User edits information	Information text (in)	The user inputs information to fill or overwrite personal information used to complete applications.
4	User edits search terms	Information text (in)	The user inputs information to fill or overwrite criteria used to decide whether a job is relevant to consider for user applications, and what websites to look in.
5	User starts search for job links	PDF Document (in) List of jobs' information (out)	The requested job board(s) are searched for postings with matching key words. A list is returned to the user containing the links and basic information about the job.

#	Event Name	Input/Output	Summary
6	User accepts recommended job posting	Signal to apply on job board (out)	Job posting gets accepted by the user and the user information is automatically sent to the job board.
7	User rejects recommended job posting	Signal to disregard job posting (out)	Information gets removed from posting and the position is <i>not</i> sent an application.

Table 1: Work Partitioning Business Events

2.1.3 Individual Product Use Cases

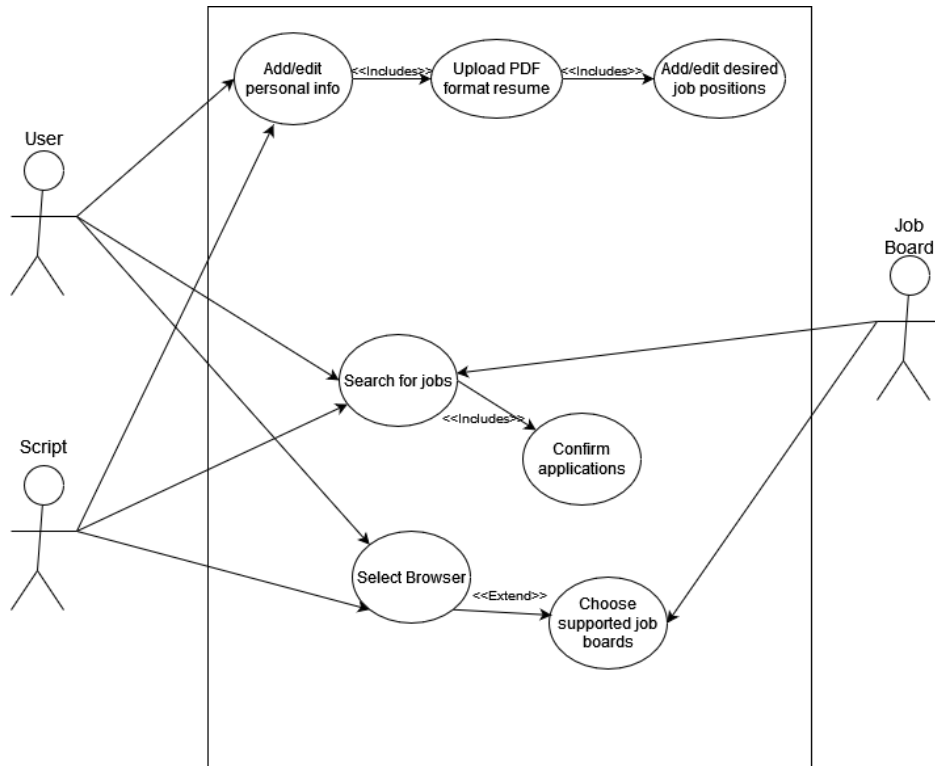


Figure 2: Use Case Diagram

2.2 Functional Requirements

FR1 The product shall allow the user to input information about themselves.

Rationale: Job applications require information about applicants.

Fit Criterion: The product shall contain a list internally of every information field needed for the supported websites, and allow the user to fill in the blanks of the self-profile.

FR2 The product shall be able to interact with PDF files designated by the user.

Rationale: The product needs to be able to extract keywords from the resume. Additionally, it needs to be able to upload PDF files to

application websites, and PDF files are usually, if not always, accepted.

Fit Criterion: When the user inputs their personal information in text form, the product shall also allow the user to also select any relevant files that may be needed for a job application. Additionally, it shall store this PDF file so it can be recalled later to be uploaded to websites.

FR3 The product shall store user information, including after it has been stopped.

Rationale: If the user is applying to multiple sites or across multiple days, inputting all the information each time is working against the goal of the product.

Fit Criterion: The product shall store a profile on the user from the information they have inputted. It shall allow it to be both read and written to so the product can use it to fetch information for applying, and the user can update their profile.

FR4 The product shall allow the user to select what websites to search for job openings on.

Rationale: The user should be able to have control over their search; the user should be able to only search on websites they like.

Fit Criterion: When the user inputs their information, the product shall also allow the user to select as many websites from the list of supported websites as they like.

FR5 The product shall allow the user to specify search terms for identifying jobs of interest.

Rationale: In order for the provided information to be effective in landing the user a job, the jobs need to be related to the user.

Fit Criterion: When the user inputs their information, the product shall also allow the user to set up a "profile" for the job openings they are trying to find.

FR6 The product shall allow the user to trigger a process that uses previously stored information to apply for jobs.

Rationale: The purpose of the product; convenient and fast job applications.

Fit Criterion: After user information has been provided, have a trigger that starts the job scraping process.

FR7 The product shall maintain a list of all jobs it has found, as well as applied to.

Rationale: If the product is used across multiple sites or multiple times on the same site in the short term, it should try and prevent multiple applications to the same job.

Fit Criterion: The product shall maintain a list of jobs it has found and have them marked separately for if they were found but not used, or found and applied to. When doing subsequent job searches, the product shall reference this list to see if a job has previously been applied to. The product shall also raise warnings for the user to investigate if it finds jobs with similar descriptions and companies on different websites, as although the link would be different, the job might be the same.

FR8 The product shall allow the user to preview job openings before automatically applying.

Rationale: The user should have control over their search. The product could therefore also be used only to scrape for jobs as opposed to applying immediately. Additionally, the user should be able to avoid applying to jobs they do not like that they did not anticipate finding.

Fit Criterion: Between the points of scraping for job openings and submitting the applications, the product shall present the links and basic job information to the user so they can review what jobs they are going to apply for. They could then cancel automatic applications for jobs they may be interested in spending more time applying to, or want to avoid applications for altogether.

3 Non-functional Requirements

3.1 Look and Feel Requirements

LF1 The system shall be easily readable with ASCII text through the command line that sufficiently displays all information.

Rationale: Accurate information is necessary for proper job applications, being able to read the output of the system is necessary for user success.

Fit Criterion: Variety of testers will be used to test out the visibility/look of the final product and its interfacing.

3.2 Usability and Humanity Requirements

UH1 The product will only require identifying data and a resume to ensure a no discriminatory application process. The system will not inquire about sex, race, religion, age, and or genetic conditions.

Rationale: Sensitive user information as listed above is morally wrong and illegal to ask for in a job application.

Fit Criterion: No user input will be related to any potential discriminatory information.

3.3 Performance Requirements

PR1 The product shall be efficient in searching and matching up to 100 jobs in an acceptable amount of time of 5 minutes.

Rationale: Having minimal response time for job applications is important since this product is specifically designed to be efficient and reduce the manual labour required for extended amounts of applications.

Fit Criterion: Measuring the time between the user's job search to match of a single job will be tested thoroughly and incremented to larger amount of jobs.

3.4 Operational and Environmental Requirements

OE1 The product should be executable on accessible modern operating systems such as Windows 10/11, MacOS and Linux.

Rationale: These are the most common operating systems currently being used on modern machines like laptops and desktops.

Fit Criterion: The product will be tested on all listed OS using personal and University provided resources.

OE2 The product should be supported on the most accessible browsers; Chrome, Safari and Firefox.

Rationale: These browsers are supported by one if not more OS as described in [OE1](#) and used frequently.

Fit Criterion: The product will be tested on such browsers using personal and University provided resources.

OE3 The product should be built and run with Python 3.

Rationale: Python 3 has great developer and user support as well as

plenty of free libraries to use.

Fit Criterion: The team will follow a rigorous development cycle using Python 3.

3.5 Maintainability and Support Requirements

MS1 All code will be publicly available and documented thoroughly.

Rationale: Thorough documentation allows developers to efficiently edit and maintain the well explained code.

Fit Criterion: All deadlines and revisions will be made public following the team's commitment to effective documentation.

3.6 Security Requirements

SR1 The product shall respect the user's privacy of personal information.

Rationale: The user should be aware of what is being done with their information; manual applications have this security feature so it should not be overlooked if automated.

Fit Criterion: For every website being used, ask the user if they have read the privacy and release of information policies before releasing information to the job boards for applications. Provide links to said policies so they can be easily accessed if the user wants to read them.

3.7 Cultural Requirements

See Usability and Humanity Requirement **UH1**.

3.8 Legal Requirements

LR1 Product will require thorough consent to necessary terms and conditions of applying on various job boards and personal information sharing.

Rationale: The user needs to be aware of job boards' intentions and their application process in using their website.

Fit Criterion: The team will thoroughly create accounts for the supported job boards and read all terms and conditions that will be necessary to share with the user.

3.9 Health and Safety Requirements

N/A

4 Project Issues

4.1 Open Issues

There are no known open issues.

4.2 Off-the-Shelf Solutions

This product is built off of an existing job application script ([found here](#)), so there are similar, previously-existing products. However, the user has less control in the original product and it is not friendly for those unfamiliar working at lower than surface level with software.

There are also job application bots on the market (such as [these ones](#)), but these appear to be subscription based, sometimes costing upwards of \$100 per year - this is not ideal for our audience, who likely want to avoid spending money to land a job.

4.3 New Problems

In upgrading the original script, we now have the challenge of developing an intuitive way to interface with all of the configuration options. This also needs to be in a way that improves over the existing user interaction, as in it's original form it is not very accessible to those without some knowledge of code.

A very challenging part, obtaining all the information from a website to then use in a program, was done in the original script. However, this was done only for one site - other sites may still prove difficult despite having a working base to go off of. Furthermore, this does not account for differences between browsers.

4.4 Tasks

All tasks to complete for this project are outlined in the [Gantt Chart](#) developed in the planning stage.

4.5 Migration to the New Product

The original code is commented fairly well, but does not include proper test cases. For that reason, the original script will likely have parts rewritten for continuity or following conventions, but will largely stay and serve as the base for the new product. The original components will have tests created for them along with the new components being written.

4.6 Risks

The product adds no new risks otherwise specified within the terms and services of specific job board websites since it is a merely an automated tool for applying to positions more efficiently.

4.7 Costs

There are no costs associated with using/developing this product apart from computer and OS requirements. All software, libraries, job boards and browsers are free to use. However, some job boards allow for a premium subscriptions to be purchased. This product will not support any features from these paid subscriptions to ensure the costs are kept to a minimum and the product is available to as many users as possible. It is to note however that certain OS like Windows 10/11 and MacOS require a purchase of a specific device or the OS itself for use.

4.8 User Documentation and Training

The team will continuously use Doxygen to create well documented code that can be easily interpreted by those with a limited technical background. All revisions of the SRS will be made public. The product will include some direction to the user to guide them through use.

4.9 Waiting Room

N/A

4.10 Ideas for Solutions

- Similar to the original script, to avoid privacy concerns, the user will need to "prepare" the browser pages before activating the script. This means opening all of the job boards they want to search on, and logging in where applicable. This way, the product does not need to store login information.
- It is still unclear as to the most efficient way of dealing with PDF documents, but there have been a few options identified. Certain Python libraries allow files to be uploaded to a script (for example [Tkinter](#)), or the script could simply store the path to the PDF.
- For storing user information, making use of a YAML file could be helpful, as it allows reading/writing both inside and outside the program. Using such a file may necessitate use of a library such as [PyYAML](#).
- To take on the portability issues (including access to Python and OS), our Python project could be compiled into an EXE file (using resources listed [here](#)).
- Dealing with PDF files can be done with the [PyPDF2](#) library, including which allows extracting document information and splitting documents by page.
- To make the product more intuitive to use, potentially expanding the user geographic, a GUI could be implemented (Tkinter, mentioned above, also works as a Python GUI). This would also allow easier viewing of all the information and files you have uploaded to be used.

5 Appendix

5.1 Symbolic Parameters

N/A

Table 2: **Revision History**

Date	Version	Notes
Feb 8	0.1	Sections and subsections 1, 2.2
Feb 10	0.2	Fill of all remaining sections