**\*Partially Adapted From: IEEE STANDARD 830: Software Requirements Specification (SRS):**

1. The Software Requirements Specification (SRS) document template given below are guidelines to the contents of your SRS. The SRS should include at least

2. For the project, your team may have good reasons for wanting to deviate from this proposed template. In that case, intimate the deviations to your Instructor In-charge before writing your document. these sections.

3. If a section is not applicable in your case, do not delete it; instead, give the topic heading and write "Not applicable".

4. You will note that there is some overlap in the content between different documents (i.e. the Software

Requirements Specification Document and the Software Design Specification Document.) This redundancy allows each document to stand on its own.

**=====================================\*\*\*\*\*\*\*\*\*====================================**

# **Software Requirement Specification Report**

By:

Gaurav Bhardwaj – 18103034

Girish Kumar – 18103035

Hardik Punj – 18103039

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Topics** | **Page No.** | **Sign.** |
| 1. | Document Review History | 1 |  |
| 2. | **Introduction to SRS** | 2 |  |
|  | Purpose and scope  Document Convention  Definitions  Intended Audience  Reference | 2 |  |
| 3. | **Overall Description** | 3 |  |
|  | Product Prespective  Functions  Operating Environment  User Documentation  General Constraints | 3 |  |
| 4. | **Specific Requirements** | 4 |  |
|  | External Interface Requirements  Functional Requirements  Design and Implementation Constraints  Other Requirements | 4-7 |  |

# **Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Version History** | **What’s New?** | **Recent submit** |
| 1. | Version alpha(α) | Basic documentation | 15/11/20 |
| 2. |  |  |  |
| 3. |  |  |  |
| 4. |  |  |  |
| 5. |  |  |  |
| 6. |  |  |  |
| 7. |  |  |  |

# **1. The Software Requirements Specification Outline**

## **1. Introduction**

### **1.1 Purpose and Scope**

Our Plan is to provide you with a software to help your profile look the best of a kind, With help of an interactive Form filling experience. This software have functionalities like easy forms, various templates, one can create their own templates, once information is fed, data will stored in disk, give suggestion for best resume style sorted according to preferences of the jobs, companies (in which you want to apply), support for different languages for people around the globe, it is desktop system so network is not required, and since data will be stored one time so no need to put it again and again, of-course data updating will be there if needed.

### **1.2 Document Conventions**

--None--

### **1.3 Definitions, Acronyms, and Abbreviations**

None. List Definitions, Acronyms, and Abbreviations used in the SRS here.

### **1.4 Intended Audience and Reading Suggestions (Overview)**

The intended Audience include all types of people like client, developers, testers, etc.

For client: This software is created for client to help them create resume document. The client is suggested to read the manual of the software before actually using the software to get familiarize with the user interface and working.

For developers: The future developers are suggested to read the manual for understanding the working, the developer should have through knowledge of Python Language as well as of its user interface modules for updating the software in future. Since software is free and open source so experienced developer can modify the source code for their own working.

For Testers: The testers should be having through knowledge of Python and GUI.

For marketing staff: They should read the manual of the software.

### **1.5 References**

## <https://python-docx.readthedocs.io/en/latest/>

* <https://py-googletrans.readthedocs.io/en/latest/>
* <https://pyinstaller.readthedocs.io/en/stable/>

## **2. Overall Description**

### **2.1 Product Perspective**

This project of making a resume builder is to provide a software to help the user to create wonderful portfolios to help him/her get through the interview section when applying for a job in a company. One can use web-based builder software to get the job done but the issue in this approach:

* May be internet connections are weak in certain region.
* The data used in template are stored in web site’s server database from where it may get leaked
* May be the template structures are fixed i.e. data can be updated only not the arrangement of the sections cannot be changed.

In context of on disk software, till now Microsoft’s Word tool is only providing this functionality. The only problem in this software is that only templates are provided and the user has to put all data by themselves, and any time you want to create new document, data is to be updated every time, plus only few templates are provided and are free, if other template is required which are downloadable are sometimes requires payment.

### **2.2 Product Functions**

The software will come with UI to get your information and store it for next step, i.e to create a CV document, user can give information in some languages If they want to, save the build doc into pdf, docx files and able to print them in instant.

### **2.3 Operating Environment**

We are creating a Desktop application so, we have planned Windows platform to be the first environment for this software. Every member is having Windows Operating System, so it will be easy to transfer data to other members. Later on, we will try to provide this software for other operating systems.

### **2.4 User Documentation**

The user guide will be provided by the Feasibility-Study-Report.

### **2.5 General Constraints, Assumptions and Dependencies**

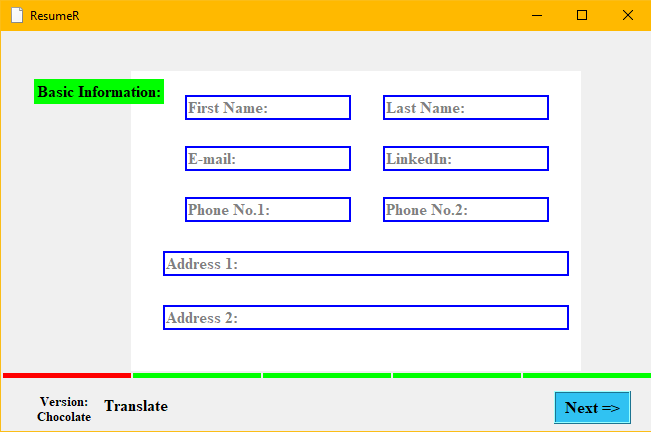
For changing language feature we use Google translate APIs i.e. **googletrans** we are restricted by the language support which this module has. we are assuming that module will work efficiently with this constraint. Similarly, for the other modules as well.

# **3. Specific Requirements**

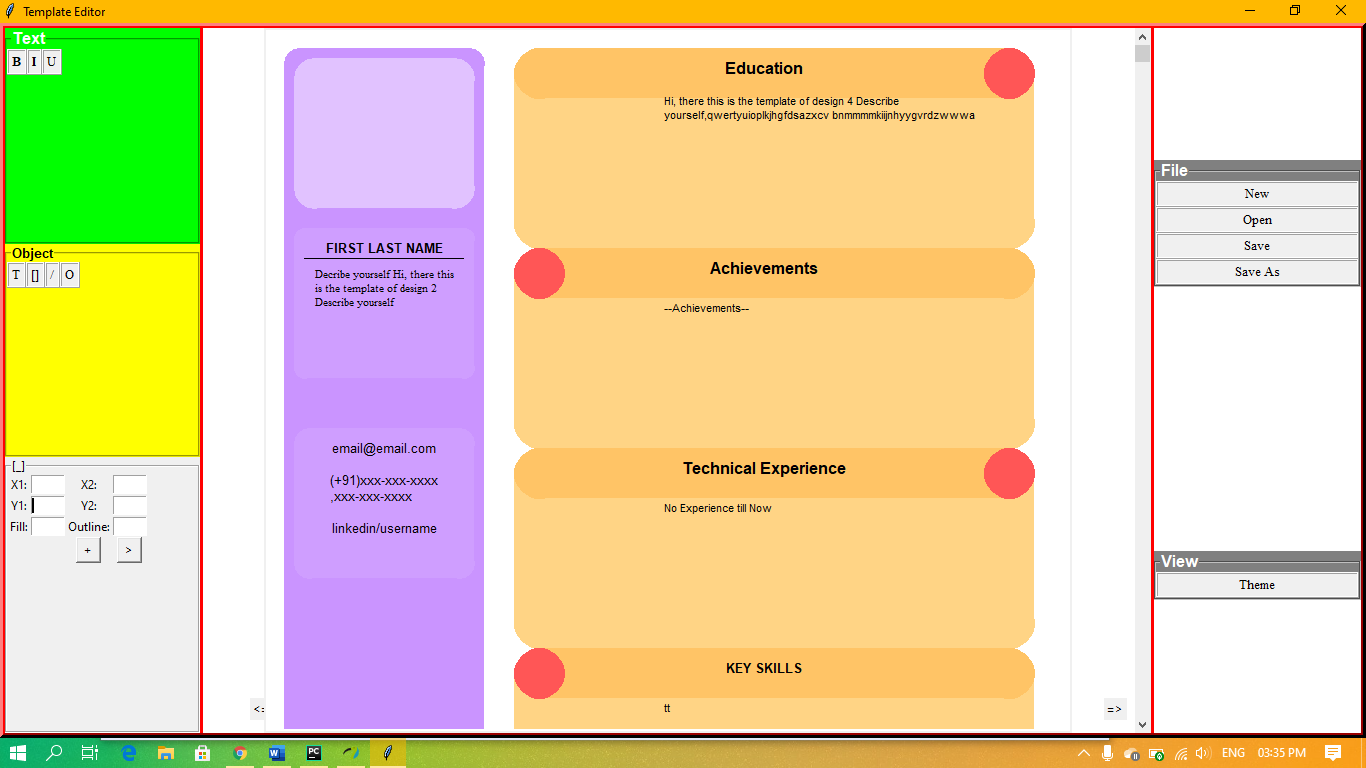
### **3.1 External Interface Requirements**

#### **3.1.1 User Interfaces**

The first half of system is data collecting input boxes. The input in the boxes varies for every input box in the application like, name field => single-line input, phone no. => 10-digit number, address fields => text boxes having multi-line inputs, etc. The data is taken on instance when button is clicked. These input boxes are placed inside frames as shown in picture below.



The second half of the system is Editing UI, The window is divided into 3 sections: Panel 1, Panel 2, Panel 3. Panel 1 consists of options/objects/tools that help you create the templates on plane canvas. Panel 2 contains the main workspace consisting the canvas (1200 x 800)px with slider on Left end. Panel 3 contains other main options for non-functionality requirements. The Current UI is shown below:



#### **3.1.2 Hardware Interfaces**

The user is required to have a PC for using this software, in addition, the user can also use a printer for printing purposes.

#### **3.1.3 Software Interfaces**

We are building this product using Python language and it’s modules(libraries) like tkinter, python-docx, mail-merge, etc. In detail-

* Python Language 3.9
* Tkinter 8.4
* Python-docx 0.8.10
* Sqlite3 Database Interface 2.0
* Pyinstaller 4.0
* googletrans 0.3.11

We are using Python in Windows 10 Operating System. The standalone product is built with tkinter module in Pycharm Editor. **Sqlite3** module provides the interface to connect to database for storing and retrieving your entered data.

**Docx** in **python-docx** module provide interface for creating or updating Ms Word .docx file.

#### **3.1.4 Communications Interfaces**

--None—

**3.2 Functional Requirements**

#### **3.2.1 Form like UI**

The form ui consists of various input taking Entry boxes taking data in different type of formats. Some of that are required fill field like first name, contact number, etc..

Purpose: To take correct information form the user to store in the database and to show on CV.

Inputs: Most of the fields take string input, some fields like phone number, scores, etc. required to be filled with numeric inputs only.

Output: Information is stored in a database table.

#### **3.2.2 Supports different languages**

This features will help to convert the English filled data to convert to specified language data.

Purpose: There may be possibility that a company takes individuals whose documents are written in specific language or a case where the user want to convert the language to be changed to other language.

Input: Select the Source-Destination language, by default Source language is English.

Output: Specified language

#### **3.2.3 Template builder**

This feature plays a main role with building a document Template from scratch, or outputting the whole document. It comes with various options like creating text objects, creating shapes, etc.

Purpose: Extending above part.

Input: for a object, its attributes as text/numeric/colour fields. To create the objects command given as button click.

Output: Showing various styles on the canvas.

#### **3.2.4 Document Converter**

To output the created document as **.pdf/.docx** files by using modules/library files of FPDF/python-docx.

Purpose: to directly help you to get various types of files of different types.

Input: file name and Button Click

Output: Saves the file as the given file name.

### **3.3 Design and Implementation Constraints**

Using Python + Tkinter limits the developer to create static shaped applications i.e. it has same appearance when it is full screened or the original size. The developer is not able to assign icons/ pictures in developing the application. The syntax of sqlite3 database should be proper.

### **3.4 Other Requirements**

--NONE--