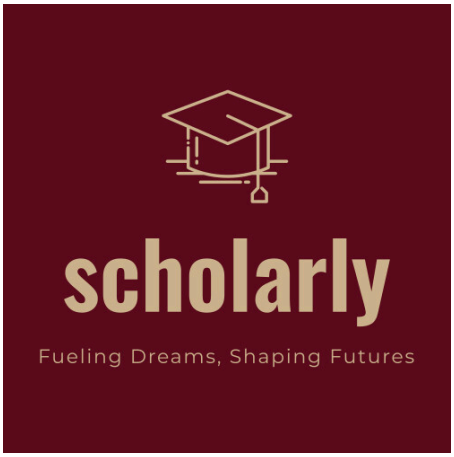
# Software Requirements Specification

## Project Title: Scholarly




## Client/Organization:
## MSU Computer Science Department

## Team Name:
## S.E.E.S

## Authors:
## Angel Badillo and Khethulwazi Kunene

## Version: 1.0

## Date: March 18, 2024

# Table of Contents

# 1. Introduction

This document is the software requirements specification for the software project named “Scholarly”, being developed by Angel Badillo and Khethulwazi Kunene. Scholarly is to be developed for the Midwestern State University Department of Computer Science for the purpose of aiding in the processing of scholarships and academic awards. This section provides insight on the purpose of this document, the scope of the software products being produced, provides definitions for terms and acronyms used in this document, provides references to other information referenced in this document, and provides an overview for the succeeding portions of this document.

## 1.1 Purpose

The purpose of this document is to serve as a software requirements specification. This document will outline the functional and non-functional requirements, the constraints, and the intended audience of this software requirements specification is Midwestern State University Department of Computer Science, the scholarship committee for the department, and the chair of the committee, as well as being involved and affiliated with scholarship application management within the department.

## 1.2 Scope

The software products to be produced are Scholarly. The scope of Scholarly, a scholarship application and academic award management system, is to streamline the process of approving scholarship applications and selecting candidates for academic awards for the Midwestern State University Department of Computer Science. Scholarly aims to provide a centralized platform for the evaluation, and approval process of scholarship applications and academic awards. The intended user is the chair of the Midwestern State University Department of Computer Science Scholarship Committee. The user can utilize and run Scholarly on devices with a Windows 11 operating system or above, with an Internet connection. Scholarly will allow the user to import student scholarship application data from a CSV file, apply a filter or a query to get the list of students qualified for a selected scholarship, and provide a means of selecting and automatically generating award letters. In addition, Scholarly will also allow the user to perform a filter or a query on imported student data, facilitate the selection of nominees for the Outstanding Student awards, and create a poll or a form for the purpose of the scholarship committee voting for the nominees.

## 1.3 Definitions, Acronyms, and Abbreviations

Below is a list of all abbreviations and acronyms that are used in this document. In addition, below are also some definitions to aid in the understanding of terms mentioned in this document.

- CEFR Level C1: A level of English proficiency where an individual can fluently communicate in English and make efficient use of it in a professional or academic setting. It is also referred to as the “advanced” level [4].
- CEFR:  Common European Framework of Reference for Languages, a rubric or guideline for assessing and describing levels of proficiency in a language [4].
- CSV: Comma Separated Values, a plain text format used to store tabular data.
- GB: Gigabytes, refers to the memory or storage capacity of a system.
- GHz: Gigahertz, refers to clock speed or frequency.
- GUI: Graphical User Interface.
- HTTPS: Hypertext Transfer Protocol Secure, used in RESTful APIs and in web applications.
- LTS: Long Term Support.
- MSU: Midwestern State University.
- Scholarly: The software to be developed in accordance with this SRS.
- SQL: Structured Query Language, a language for relational databases.
- SQLite: A library that implements a small, fast, self-contained, and reliable SQL database engine. The SQLite format is designed to be stable, backwards compatible, and cross platform. It is promised that SQLite’s LTS will last until the year 2050. SQLite’s source code is entirely public-domain and allows for free use for any purpose [2].
- SRS: Software Requirements Specification.
- UI: User Interface.

## 1.4 References

This document was written following the IEEE 830-1998 standard for recommended practices for writing an SRS [1]. Below are citations to other documents, sources, and information that were referenced in this document.

[1] "IEEE Recommended Practice for Software Requirements Specifications," in *IEEE Std 830-1998*, vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286.
https://www.sqlite.org/index.html

[2] “About SQLite,” SQLite, https://www.sqlite.org/about.html.

[3] “Windows 11 Requirements,” Microsoft Learn, https://learn.microsoft.com/en-us/windows/whats-new/windows-11-requirements.

[4] “English C1 Level,” English C1 Level - CEFR Definition, https://www.efset.org/cefr/c1/.

## 1.5 Overview

The remainder of the content in this document is organized in the following manner: Section 2 details the context of the product, the features and functionality of the product, the characteristics of the user, the constraints on the system, and the assumptions being made and the dependencies. Section 3 goes over the specific requirements, such as the external interface requirements, the functional requirements, the performance requirements, design constraints, quality requirements, and others as needed.

# 2. Overall Description

As previously mentioned, Scholarly is a software product that is meant to aid the processing of scholarships and academic awards. The software will have functionality that corresponds to that, such as a means of filtering through student applicant data and generating personalized emails for recipients of scholarships. This section goes into more detail about the context of the product, the functionality of the product, the characteristics of the user, the constraints placed on the project, and assumptions that are made as well as dependencies.

## 2.1 Context of Product

Scholarly is an application that will be designed to process student data, generate letters, and create a poll for voting for candidates for the Outstanding Student Awards. There are similar products to Scholarly, such as scholarship application management systems, like SurveyMonkey Apply, SmarterSelect, and CampusLabs. However, Scholarly differs in the sense that the primary user is the Chair of the MSU Computer Science Scholarship Committee, and does not include students or applicants as users. Scholarly mainly just makes it easier for the user to select candidates, generate a letter for the candidate, as well as polling for candidates for the Outstanding Student Awards. Scholarly is a product that is completely self-contained and is not a part of some other larger system.

## 2.2 Product Function

The function of the product defines what our product can do, which has been derived from an assessment of the requirements. Below is a list of major functions Scholarly will perform:

- The application shall provide a way to filter through student data based on the criteria for a scholarship.
- The application shall provide a means of generating letters for the purpose of emailing them to students who qualify for a scholarship.
- The application shall allow the user to filter through student data to create a Google Form that serves as a poll for the Outstanding Student Awards where the other members of the MSU Computer Science Scholarship Committee can vote on the candidates for each classification and category of the Outstanding Student Awards.

## 2.3 User Characteristics

The characteristics of the end user should provide insight on how they may intend to use the product/system consequently, identification and examination of our end user is crucial. Firstly, we identified our primary client and end user will be the Chair of the MSU Computer Science Scholarship Committee. The characteristics of this user are as follows:

- CEFR Level C2 English proficiency or better
- Chair of the Scholarship Committee
- Computer literate
- Experience with CSV files
- Experience with Google Forms
- Experience with writing letters and sending emails
- Has access to student records and information
- Knowledgeable on criteria for scholarships and academic awards
- No disabilities impairing the use of the application

## 2.4 Constraints

- Scholarly must be designed for devices with the Windows 11 operating system or above.
- The Scholarly source code must be coded in the Python programming language, version 3.12.2 or above.
- Scholarly must not transmit, release, or distribute sensitive student data in an unauthorized fashion.

- When transmitting student data online, Scholarly must do so using secure, encrypted protocols, such as HTTPS.
- Scholarly must be designed to be a GUI application.

## 2.5 Assumptions and Dependencies

- It is assumed that the operating system Windows 11 will be available on the device running the software product.
- It is assumed that the device running the software product will have a web browser.
- It is assumed that the device running the software product will have the capability to access the Internet.
- It is assumed that the network that the device is connected to will have no web filter, or a web filter that allows access to the Google Forms site, the site where Scholarly's manual will be available online, and the website with the about information for Scholarly.

# 3. Specific Requirements

This section focuses on sufficiently detailing requirements that enables the developers of the system to properly satisfy the requirements, and for testers of the system to properly ensure that the system satisfies the requirements. Requirements are organized in the following categories: external interface requirements, functional requirements, performance requirements, design constraints, quality requirements and other requirements.

## 3.1 External Interface Requirements

This subsection focuses on external interface requirements as well as the input and outputs of the software system.

### 3.1.1 User Interfaces

- The UI for Scholarly must be a GUI.
- Scholarly must have a full screen mode.
- Scholarly must have a windowed mode.
- Scholarly must have a menu strip.
    - Scholarly must have a submenu named "File" that menu items pertaining to opening of a file, saving to a file, and closing a file.
    - Scholarly must have a submenu named "Help" that provides a means for the user to access the user manual.

- Scholarly must have a table to display the student data imported from a file.
- Scholarly must have a UI component that enables the user to select an option for filtering student data based on the type of scholarship or academic award.
- Scholarly must have a UI component that enables the user to sort the student data in descending order of cumulative GPA.
- Scholarly must display descriptive messages in the event of an error.

### 3.1.2 Hardware Interfaces

Scholarly must run on hardware that meets all the requirements for running the operating system Windows 11 [3] or later. Below are the main hardware requirements for Windows 11. For more information, refer to [3].

- 1 GHz 64-bit compatible processor with 2 or more cores.
- Memory of 4 GB or more.
- Storage of 64 GB available disk space or more.
- Internet connection
- 720p display or above.
- Graphics card or integrated graphics compatible with DirectX Version 12 or above.

### 3.1.3 Software Interfaces

- Scholarly must run on devices with the Windows 11 operating system or later. Scholarly will be designed to run specifically on devices with the Windows 11 operating system. Below are details about the Windows 11 operating system:
    - Name: Microsoft Windows 11 Operating System
        - Version: 23H2 or above.
        - Source: https://www.microsoft.com/software-download/windows11
- Scholarly must be able to open Microsoft Edge, version 122.0.2365.80 or later, or the device's default browser, if possible. If it is not possible to access the device's default browser, then Microsoft Edge will be the browser that is opened. Scholarly will open the browser for the purpose of displaying the created Google Form for the purpose of voting for candidates for the Outstanding Student Awards. Below are details about Microsoft Edge:
    - Name: Microsoft Edge
        - Version: 122.0.2365.80
        - Source: https://www.microsoft.com/en-us/edge/download
- Scholarly must be written in the Python programming language, version 3.12 or later. Below are details about Python:
    - Name: Python
        - Version: 3.12.2
        - Source: https://www.python.org/downloads/release/python-3122/

- Scholarly must be designed to be a GUI application, therefore the use of the PyQt6, a library that provides Python bindings to the Qt v6 libraries, which are cross-platform C++ libraries for modern desktop development, will be employed. Below are details about PyQt6:
    - Name: PyQt6
        - Version: 6.6.1
        - Source: https://pypi.org/project/PyQt6/#installation
- Scholarly must be able to create and modify Google Forms, thus several Python packages, developed by Google, are required. The packages are used for connecting to Google APIs and authorization purposes. Below are the required packages:
    - Name: google-api-python-client
        - Version: 2.122.0
        - Source: https://pypi.org/project/google-api-python-client/#installation
    - Name: google-auth-httplib2
        - Version: 0.2.0
        - Source: https://pypi.org/project/google-auth-httplib2/#installing
    - Name: google-auth-oauthlib
        - Version: 1.2.0
        - Source: https://pypi.org/project/google-auth-oauthlib/#installing

### 3.1.4 Communications Interfaces

The communication protocol that will be used in this project is HTTPS. Since the usage of Google APIs for the purpose of creating Google Forms will occur, communication between the Google APIs will conform to HTTPS.

## 3.2 Functional Requirements

### 3.2.1 Functional Requirement 1

**Priority: High**
Scholarly must facilitate the selection of qualified students for a scholarship by the MSU Computer Science Scholarship Committee.

### 3.2.2 Functional Requirement 2

**Priority: High**
Scholarly must have a feature to open and read a CSV file with student data through a file dialog.

### 3.2.3 Functional Requirement 3

**Priority: Low**
Scholarly must validate student data that is read in from the CSV file.

### 3.2.4 Functional Requirement 4

**Priority: Low**
Scholarly must show an error message in the event of incorrectly formatted data from the CSV file.

### 3.2.5 Functional Requirement 5

**Priority: Low**
Scholarly must display only CSV files when displaying the file dialog.

### 3.2.6 Functional Requirement 6

**Priority: High**
Scholarly must have a feature for sorting student data based on the eligibility criteria for scholarships.

### 3.2.7 Functional Requirement 7

**Priority: High**
Scholarly must have a feature for filtering student data based on the eligibility criteria for scholarships.

### 3.2.8 Functional Requirement 8

**Priority: Medium**
Scholarly must allow the user to manage eligibility criteria for a scholarship.

#### 3.2.8.1 Functional Requirement 8.1

**Priority: Medium**
Scholarly must allow the user to add new eligibility criteria for a scholarship to the list of existing criteria.

#### 3.2.8.2 Functional Requirement 8.2

**Priority: Medium**
Scholarly must allow the user to modify existing eligibility criteria for a scholarship.

#### 3.2.8.3 Functional Requirement 8.3

**Priority: Medium**

Scholarly must allow the user to remove existing eligibility criteria for a scholarship.

### 3.2.9 Functional Requirement 9

**Priority: High**
Scholarly must facilitate the voting for candidates for the Outstanding Student awards by the MSU Computer Science Committee and the Chair.

### 3.2.10 Functional Requirement 10

**Priority: High**
Scholarly must allow the user to create a Google Form with the top candidates for each category of the Outstanding Student Awards.

### 3.2.11 Functional Requirement 11

**Priority: Low**
Scholarly must allow the user to save the results of a query for students fulfilling the criteria for a given scholarship to a CSV file.

### 3.2.12 Functional Requirement 12

**Priority: High**
Scholarly must generate scholarship acceptance letters for selected students that fulfill the criteria for a scholarship.

## 3.3 Performance Requirements

- Scholarly must support one or more main windows of the application, as the device memory and storage allows.
- Scholarly must be able to read in 10,000 student records from a CSV file in less than 20 seconds.
- Scholarly must be able to generate 10,000 scholarship acceptance letters in less than 60 seconds.
- Scholarly must be able to complete a standard sort or filtering of students records of 10,000 students within 20 seconds.

## 3.4 Design Constraints

Below are the design constraints for the system, as previously stated in the 2.4 constraints section:

- Scholarly must be designed for devices with the Windows 11 operating system or above. Meaning, Scholarly must run on devices meeting the minimum requirements for Windows 11.
- The Scholarly source code must be coded in the Python programming language, version 3.12.2 or above.
- Scholarly must be designed to be a GUI application.

## 3.6 Other Requirements

- Scholarly must utilize some form of relational database for the purpose of querying student data.
    - The relational database must be serverless, and self-contained, similarly to SQLite 3.

# Appendix A: Example Input Data Format

Below is an example format of how the student data may be represented as a table in the relational database. Similarly, the CSV file will contain the same columns as the table in the relational database. However, the exact format of the student data is subject to change.

| name | student_ID | cum_gpa | major | classification | earned_credits | enrolled | email | gender | in_state |
|---|---|---|---|---|---|---|---|---|---|
| Assante, Roy | M41848895 | 2.571 | Computer Science | Senior | 96 | Yes | rassante5@de.vu | Male | No |
| Libri, Alick | M51043098 | 1.669 | English | Senior | 101 | No | alibri6@yelp.com | Male | No |
| Jedrzaszkiewicz, Arron | M66365006 | 3.657 | Biology | Freshman | 12 | No | ajedrzaszkiewicz7@hud.gov | Male | Yes |
| Bowne, Cesya | M59977508 | 0.608 | Psychology | Sophomore | 78 | Yes | cbowne8@drupal.org | Female | Yes |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Cattow, Bellina | M61231910 | 0.85 | Chemistry | Junior | 87 | No | bcattow9@naver.com | Female | Yes |
| Cuskery, Shurlocke | M87652171 | 1.23 | Psychology | Sophomore | 46 | Yes | scuskerya@ocn.ne.jp | Male | Yes |
| Maltster, Annice | M49119560 | 2.399 | Computer Science | Graduate | 33 | Yes | amaltsterb@123-reg.co.uk | Female | Yes |