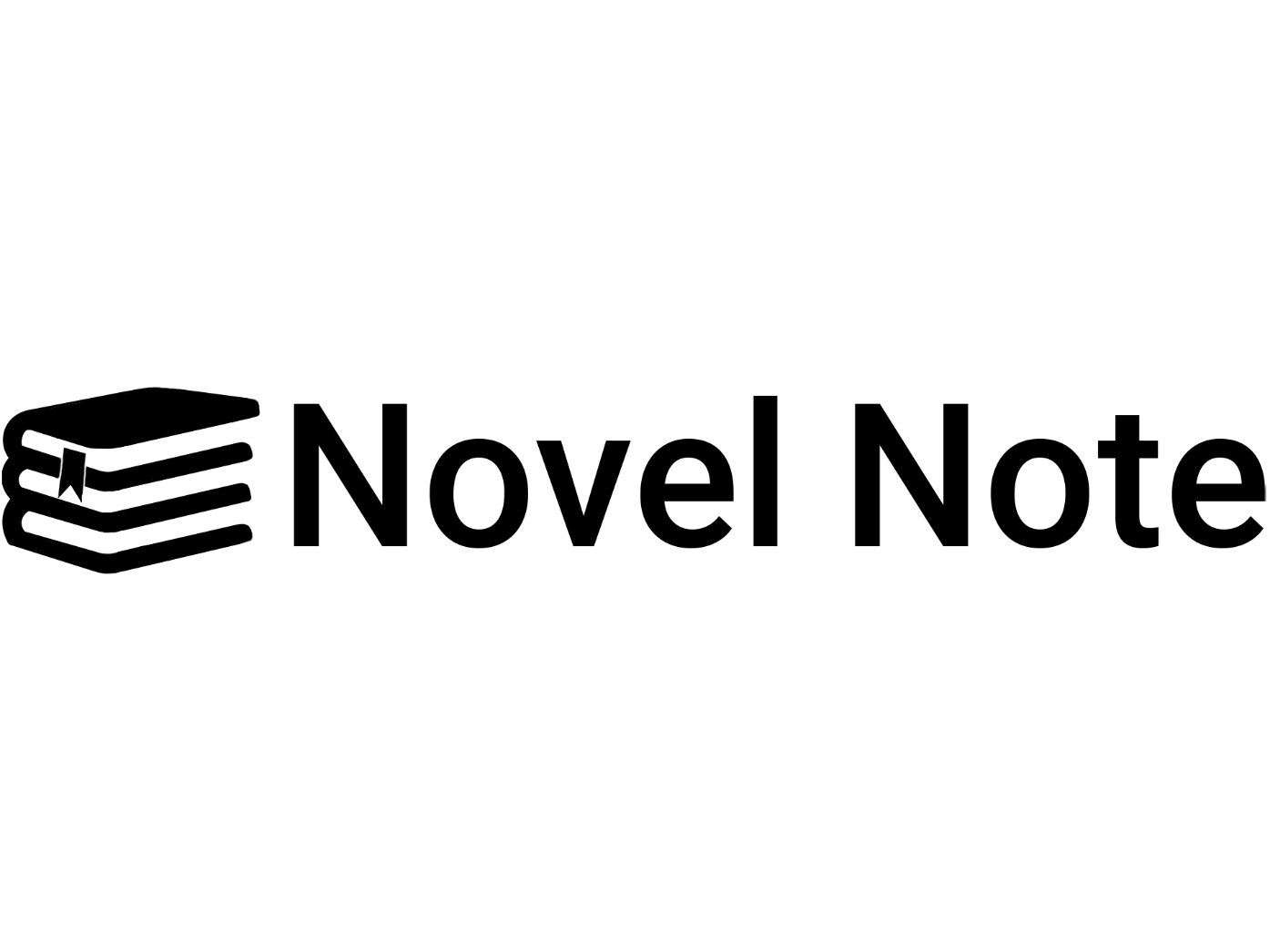
­­­­­­­­

12SDD Major Project

­­­­2023

William Gaston

Contents

[Defining and Planning 5](#_Toc140958565)

[**Problem Statement** 6](#_Toc140958566)

[Problem Profile: 6](#_Toc140958567)

[Client Requirements: 6](#_Toc140958568)

[Overview of Aims/Objectives, Specifications, Limitations: 7](#_Toc140958569)

[**Overview of Hardware and Software** 8](#_Toc140958570)

[Hardware: 8](#_Toc140958571)

[Software: 8](#_Toc140958572)

[**Discussion of Development Approach** 9](#_Toc140958573)

[Justification of Structured Approach: 9](#_Toc140958574)

[Comparing and Contrasting other Development Approaches: 9](#_Toc140958575)

[Designing 11](#_Toc140958576)

[**Gantt Charts** 12](#_Toc140958577)

[**System Documentation Overview and Justification** 13](#_Toc140958578)

[Purpose/Importance of Documentation: 13](#_Toc140958579)

[Examples of how Design Tools may Assist Development: 13](#_Toc140958580)

[**Design Tools** 15](#_Toc140958581)

[IPO Chart: 15](#_Toc140958582)

[Data Dictionary: 18](#_Toc140958583)

[Context Diagram: 21](#_Toc140958584)

[Structure Chart: 22](#_Toc140958585)

[System Flowchart: 25](#_Toc140958586)

[UI Design: 26](#_Toc140958587)

[Storyboard: 31](#_Toc140958588)

[Social and Ethical Issues 32](#_Toc140958589)

[**Social and Ethical Issues Overview** 33](#_Toc140958590)

[Relevant Social and Ethical Issues: 33](#_Toc140958591)

[Intellectual Property: 33](#_Toc140958592)

[Privacy: 33](#_Toc140958593)

[Quality: 33](#_Toc140958594)

[Inclusivity: 34](#_Toc140958595)

[Ergonomics: 34](#_Toc140958596)

[**Justification of Social and Ethical Issues in Relation to Major Project** 35](#_Toc140958597)

[Implementing the Solution 37](#_Toc140958598)

[Algorithms 37](#_Toc140958599)

[**Login:** 37](#_Toc140958600)

[**Search Functions:** 40](#_Toc140958601)

[**Select Frame:** 42](#_Toc140958602)

[**Save Entry:** 44](#_Toc140958603)

[**Save Quotes Entry:** 46](#_Toc140958604)

[**Remove Entry:** 48](#_Toc140958605)

[**Library Treeview Insert:** 50](#_Toc140958606)

[**Quotes Treeview Insert:** 52](#_Toc140958607)

[**TopNovel Treeview Insert:** 54](#_Toc140958608)

[**TopCharacter Treeview Insert:** 56](#_Toc140958609)

[**Recent Entry Treeview Insert:** 58](#_Toc140958610)

[**Open Toplevel Window (no condition):** 60](#_Toc140958611)

[**Open Toplevel Window (selection condition):** 61](#_Toc140958612)

[**Choose Chart:** 63](#_Toc140958613)

[**Filter:** 65](#_Toc140958614)

[**Close/Cancel Window:** 68](#_Toc140958615)

[**Logout Window:** 69](#_Toc140958616)

[User Interface Development 70](#_Toc140958617)

[**Login Window:** 70](#_Toc140958618)

[**Home Screen:** 72](#_Toc140958619)

[**Quotes Screen:** 75](#_Toc140958620)

[**Library Screen:** 80](#_Toc140958621)

[**Stats Screen:** 86](#_Toc140958622)

[**Search Screen:** 88](#_Toc140958623)

[**Profile Popup:** 89](#_Toc140958624)

[Source Code 90](#_Toc140958625)

[User Manual 90](#_Toc140958626)

[Testing and Maintaining 91](#_Toc140958627)

[Testing Overview 91](#_Toc140958628)

[Test Data Table 92](#_Toc140958629)

[Maintenance Overview 103](#_Toc140958630)

[Learning Journal 104](#_Toc140958631)

# **­**

# **Defining and Planning**

## **Problem Statement**

Project Name: Novel Note

Client Name: William Gaston

Age: 17

### Problem Profile:

William Gaston is a high school student with a primary hobby of reading all sorts of literature, particularly web novels. The client has expressed a desire for an all-purpose, manual, novel and literature ‘tracking’ software in order to properly remember all the literature they have read/are reading/want to read as it is not uncommon for the client to lose track of the novels they've read, what they thought about them, and what they want to read next. This is especially true as a lot of the web novels William reads are translated from other languages, and thus do not have the English-speaking online presence to be made available on similar, existing applications such as Goodreads and StoryGraph. Furthermore, the client has expressed his desire for a platform which can be used to store quotes; where they come from and who it has been said by. This is due to the lack of suitable solutions on the market which combine quote and novel recording capabilities. The client has requested that the software be a desktop application.

## Client Requirements:

The client has engaged in discussions to outline the basic functionality of the application. The application must facilitate the multiple facets of novel and quote tracking, allowing readers to track their reading progress, give ratings, and organise their library of novels and quotes. This application should accurately record information from novels and quotes including title, chapter counts, ratings, genre information/tags and completion status, and the quote, character saying the quote and the novel source. This would allow the client to create a personalised “library of novels and quotes”. Additionally, the client has emphasised that the ability to manually input novels be available, as they commonly read niche titles with minimal, online presence.

The software should utilise a graphical-user-interface (GUI) with a clean, minimalistic, and intuitive design, that allows for quick and easy navigation. Furthermore, the client would like a ‘tab’ system for the GUI which allows them to seamlessly switch between different parts of the application, primarily a ‘Home’ tab, ‘Quote’ tab, ‘Library’ tab, and ‘Stats’ tab. This would eliminate the need for excessive pop-up and back buttons, simplifying the interface.

* The ‘Home’ tab should consist of recent entries and a simple tiering system for novels and characters.
* The ‘Library’ tab should consist of a table showcasing the library, filtering options, a search bar, adding and editing buttons for the novels.
* The ‘Quotes’ tab should function similar to the ‘Library’ but in place of novels it consists of quotes.
* The ‘Stats’ tab should consist of relevant charts and graphs highlighting key data from the library, for example, listing the total chapter read.

Further discussions will be conducted to refine the client’s desired features and enhance the application’s functionality and design to further reflect the wants of the client.

### Overview of Aims/Objectives, Specifications, Limitations:

The primary aim of the ‘Novel Note’ application is to create a user-friendly, ergonomic application that allows the users to keep up with and record information about novels and quotes. The objectives of the application are as follows:

* Create an interface for users to record novels they’ve read, are reading, or plan to read.
* Allow users to provide ratings for novels.
* Allow users to list various quotes, including character and novel source.
* Create and display relevant statistics from novel data.

Specifications:

* Ergonomic, Minimalistic and Intuitive Interface
* Simple Navigation
* Multiple ‘Tabs’
* Manual Novel Recording

Limitations:

* Does not have cloud storage options.
* No novel pictures.
* Limited security measures
* Limited premade novel libraries (database)

## **Overview of Hardware and Software**

### Hardware:

* Microsoft Surface Laptop 4 (Processor: 11th Gen Intel(R) Core(TM) i7-1185G7, RAM: 16gb)

-Will be used for the duration of the project as the primary computer system.

### Software:

* Design Tools

-LucidChart: Used to create system diagrams/charts

-Dia Designer: Used to create system diagrams/charts

- Microsoft Word: Used to create primary portfolio document

- Microsoft Excel: Used to create Gantt charts and csv files

- Microsoft OneNote: Used to synthesise portfolio and access course guidelines

- Adobe Illustrator: Used to create logo and UI designs

- Adobe Photoshop: Used to create logo and UI designs

* Coding Environment

-Visual Studio Code 2022 Used as an IDE for coding

(The coding language used throughout the duration of the project will be Python)

* Miscellaneous

-Firefox (including stack overflow, git hub) Used as dedicated search engine

## **Discussion of Development Approach**

### Justification of Structured Approach:

Structured Approach: “a process-oriented approach, aiming to break a large complex project into a series of smaller, more manageable modules” – Muirfield High School

The Structured Approach is an effective software development approach as it offers a systematic and organised approach to project creation, management, and development. This is particularly effective in the creation of this project as it offers a plan to abide by and follow, allowing development to follow the strict time schedule of an HSC assessment period. Due to the demanding nature of the HSC course (within software design, and development, as well as other subjects) this will prove vital in ensuring the project is completed for the due date, as it outlines what must be done when. Additionally, the rigorous documentation required is similar to that of the structured approach due to continued abstraction throughout the creation of the project allowing each step to be worked on and focused individually. Furthermore, the structured approach dictates that the development process must follow a sequential order and must be completed ‘step-by-step’ which is a similar requirement to that of the major project.

Below are further justifications to the use of the structured approach:

* Improved Clarity and Organisation

-The structured approach breaks large problems into smaller problems allowing for easier organisation and understanding of the tasks/processes to be completed.

* Better Time Management

-The structured approach offers realistic deadlines which can be tracked to monitor progress.

* Greater Efficiency

-A clear guideline of what must be completed reduces time wastage.

* Improved Communication and Collaboration

-The structured approach offers a solid framework of the software, allowing developers to refine the project with the assistance and input of the client.

### Comparing and Contrasting other Development Approaches:

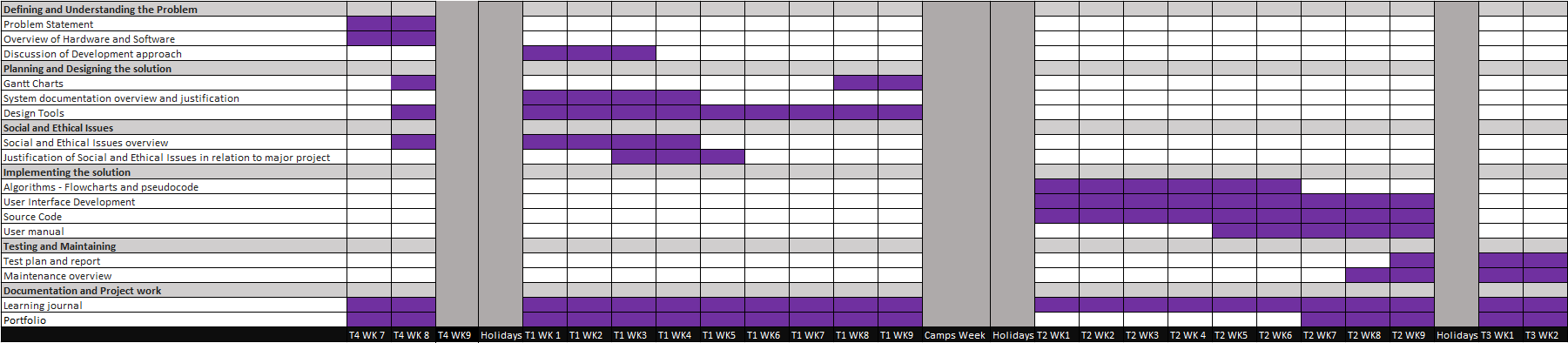
Agile Approach: “an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches” – Atlassian.

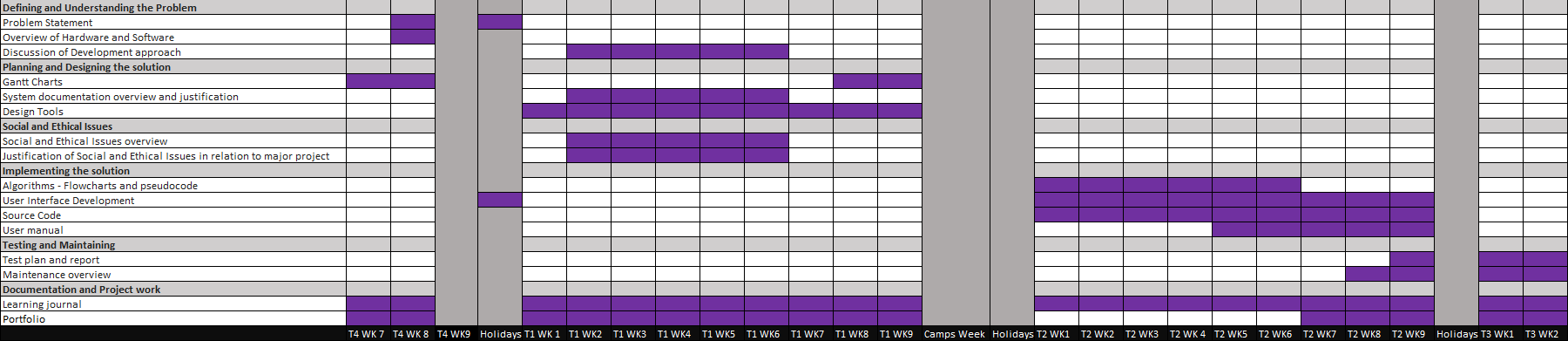
The agile development approach is similar to the structured approach in that it is a method of developing a software project. The agile approach is characterised by its iterative and flexible nature, with an emphasis on collaboration, continued refinement, and rapid delivery of new updates, as opposed to the rigid, sequential and heavily order. The agile approach derives its value from enhanced user feedback, adaptable software and processes, and a fast turnaround on new content. Furthermore, this approach is advised for systems which may undergo changes to system requirements and circumstance, aiming to best suit the customer’s need.

In the scope of the HSC major project, the agile development approach is not the most beneficial. The major project is split up into two sections, system planning, and system implementation. The system planning stage is used to gain a clear understanding of the framework of the project, creating a pre-determined plan with which to code, and as such is benefitted by the strict sequential order of the structured approach. This stage, once completed, will allow the developer to have a clear understanding of the user requirements and problem which must be solved. As such, there is little need for iterative updates and changes resulting from a change in user requirements and needs, thus eliminating the need for the flexibility of the agile development approach. Additionally, the system implementation stage is where the development of the application itself will take place. This stage builds on from the previous, utilising a solid framework (as utilised in the structured approach) of tasks which must be completed. The nature of the agile approach means that it is best used when the framework of the software is everchanging and is not the best approach for such a rigid development cycle. That being said, elements of the agile approach such as high collaboration between the user and developer and the utilisation of sprints for different sections of development are useful in both stages of the project to ensure the project is both on time and functional to the user’s specifications. These elements will likely be combined with the structured approach to ensure the software is developed to the highest standard.

# **Designing**

## **Gantt Charts**

V1:

V2:

V3:

\*not assessed\*

## **System Documentation Overview and Justification**

### Purpose/Importance of Documentation:

System documentation are all the documents, both internal and external, which provide an overview of the system's technical, functional, and operational information. Documentation is primarily used as a reference for developers, testers, and other members of the project management/development team to better understand the system’s design, development, testing and maintenance. A key component of documentation is its ability to simplify complex systems, offering a vessel for comprehensive understanding. Without documentation it can be difficult to understand the complexities of a system and is of vital importance in the development, maintenance and testing phase to accurately gauge the desired functionality of a system and ensure a high quality product.

Below are some more specific objectives/purposes of system documentation:

* **Communication:**

-documentation offers a clear, concise overview of a system, therefore it is often used as a method of communication between the developer and user, as well as between the user’s requirements and development.

* **Maintenance:**

-documentation is useful in maintaining a system as it provides up to date information on expected performance of the system, which is necessary in changing any aspects of the system to reduce unintended effects.

* **Consistency:**

-in providing standardised procedures and guideline, developers are able to work more efficiently and effectively, while also reducing unneeded inconsistencies by sticking to the precise user requirements.

* **Troubleshooting/Testing:**

**-**a core part of the testing process is ensuring that the software works as intended. Documentation gives the software a basis with which to test against, allowing for potential test scenarios/errors to be identified.

* **Training:**

-documentation provides a reference for learning by outlining key components of the system and breaking them down into smaller processes which are easy to understand without prior knowledge.

### Examples of how Design Tools may Assist Development:

* **Code Comments:**

Code comments are one of the primary methods of internal documentation, that is, documentation within the code itself. Code comments enhance the readability of the code, offering insight to the developer’s vision, and allowing easier maintenance since inconsistencies can be more easily examined.

* **Data Flow Diagram:**

A data flow diagram provides a ‘visual representation of the information flow through a process or system’. This allows for a succinct depiction of the requirements for the sequences and processes of the system, primarily the movement of data between these components. The graphical format allows for detail while also expressing complex information.

* **Context Diagram:**

A context diagram ‘outlines how external entities interact with an internal software system’. This diagram shows the entire scope of the system under a singular entity, showing the relationships between each of the individual processes within this scope. This allows for abstraction of the user’s requirements and ‘map(ping) out’ the system, and the various other entities which it interacts with (including users and external databases).

## **Design Tools**

### IPO Chart:

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| username, password  -Login Button click | -Check if all entry fields have inputs | -If False, display error message “You must input values for all fields” |
|  | -If all fields have inputs is True,  -Check if username/password match to existing in user\_list | -If False, display error message “Username or Password Incorrect” |
|  | -If all fields have inputs is True,  And username/password match to existing in user\_list  Load Home Window | -Display Login confirmation message  “Login Successful”  -Display Home Window |
| username, password  -Sign Up button click | -Check if all entry fields have inputs | If False, display error message “You must input values for all fields” |
|  | -If all fields have inputs is True,  -Check if username matches to existing in user\_list | -If True, display error message “Username already in use. Please choose another” |
|  | -If all fields have inputs is True,  And username does not match to existing in user\_list  -Write username and password to user database  -Write username and password to user\_list  -Load Home Window | -Display Sign Up confirmation message “Registration Successful”  -Display Home Window |
| -SIGN UP text click | -Load Sign Up window  -Withdraw Login window | -Display Sign Up window |
| -LOGIN text click | -Load Login window  -Withdraw Sign Up window | -Display Login window |
| -Add Entry Button Click | -Load Add Entry popup | -Display Add Entry popup |
| -Edit Entry Button Click | -Load Edit Entry popup/  Top character popup/  Top novel popup | -Display selected popup |
| novel, character, quote (Quote entry fields) |  |  |
| -Save Entry Button Click (quote form) | -Check if all entry fields have inputs | -If False, display error message “You must input values for all fields” |
|  | -If all fields have inputs is True,  -Write entry to quote database,  -Write entry to quote\_list | -Display updated quote\_list in quote treeview |
| novel\_title, chapter\_count, novel\_type, genre, library\_status, rating (Library entry fields) |  |  |
| -Save Entry Button Click (library form) | -Check if all entry fields have inputs | -If False, display error message “You must input values for all fields” |
|  | -If all fields have inputs is True,  -Write entry to library database,  -Write entry to library\_list | -Display updated library\_list in library treeview |
| top\_novel (Top Novel entry fields) |  |  |
| -Save Entry Button Click (top\_novel form) | -Check if all entry fields have inputs | -If False, display error message “You must input values for all fields” |
|  | -If all fields have inputs is True,  -Write entry to top database,  -Write entry to top\_novel\_list | -Display updated top\_novel\_list in top novel treeview |
| top\_character (Top Character entry fields) |  |  |
| -Save Entry Button Click (top\_character form) | -Check if all entry fields have inputs | -If False, display error message “You must input values for all fields” |
|  | -If all fields have inputs is True,  -Write entry to top database,  -Write entry to top\_character\_list | -Display updated top\_character\_list in Top Novel treeview |
| -Cancel Entry Button Click | -Withdraw Add Entry Button or Edit Entry Button popup | -Display updated Add Entry Button or Edit Entry Button popup |
| -Delete Entry Button Click (library form) | -Check if novel\_title is selected | -If False, display error message “You must select a novel” |
|  | -If novel\_title is selected is True  -Delete record for the novel\_title from the library database  -Delete record from library\_list | -Display updated library\_list in library treeview |
| -Delete Entry Button Click (quote form) | -Check if quote is selected | -If False, display error message “You must select a quote” |
|  | -If quote is selected is True  -Delete record for the quote from the quote database  -Delete record from quote\_list | -Display updated quote\_list in quote treeview |
| Select Screen  -click NovelNote logo, library, quote, stats  -type in search bar | -Remove current frame  -Input selected frame | -Display selected frame |
| -Keystroke in search bar | -Detect keystroke  -Filter list (quote, library, premade) by keystroke | -Display updated list in treeview |
| -Help Button Click | -Check the selected screen | -Display error message relevant to screen (home, library, quote, stats). |
| filter\_novel/  filter\_quote/  filter\_character |  |  |
| -Click Quote Filter button | -Refine quote\_list by filter option | -Display updated quote\_list in quote treeview |
| filter\_novel\_title/ filter\_chapter\_count/  filter\_novel\_type/  filter\_genre/  filter\_library\_status/  filter\_rating |  |  |
| -Click Library Filter button | -Refine library\_list by filter option | -Display updated library\_list in library treeview |
| -Click advanced\_chart Filter button | -Refine advanced\_chart by filter option | -Display updated advanced\_cahrt in on stats screen |
| -Click profile button | -detect user from user\_list | Display message “User: username” |

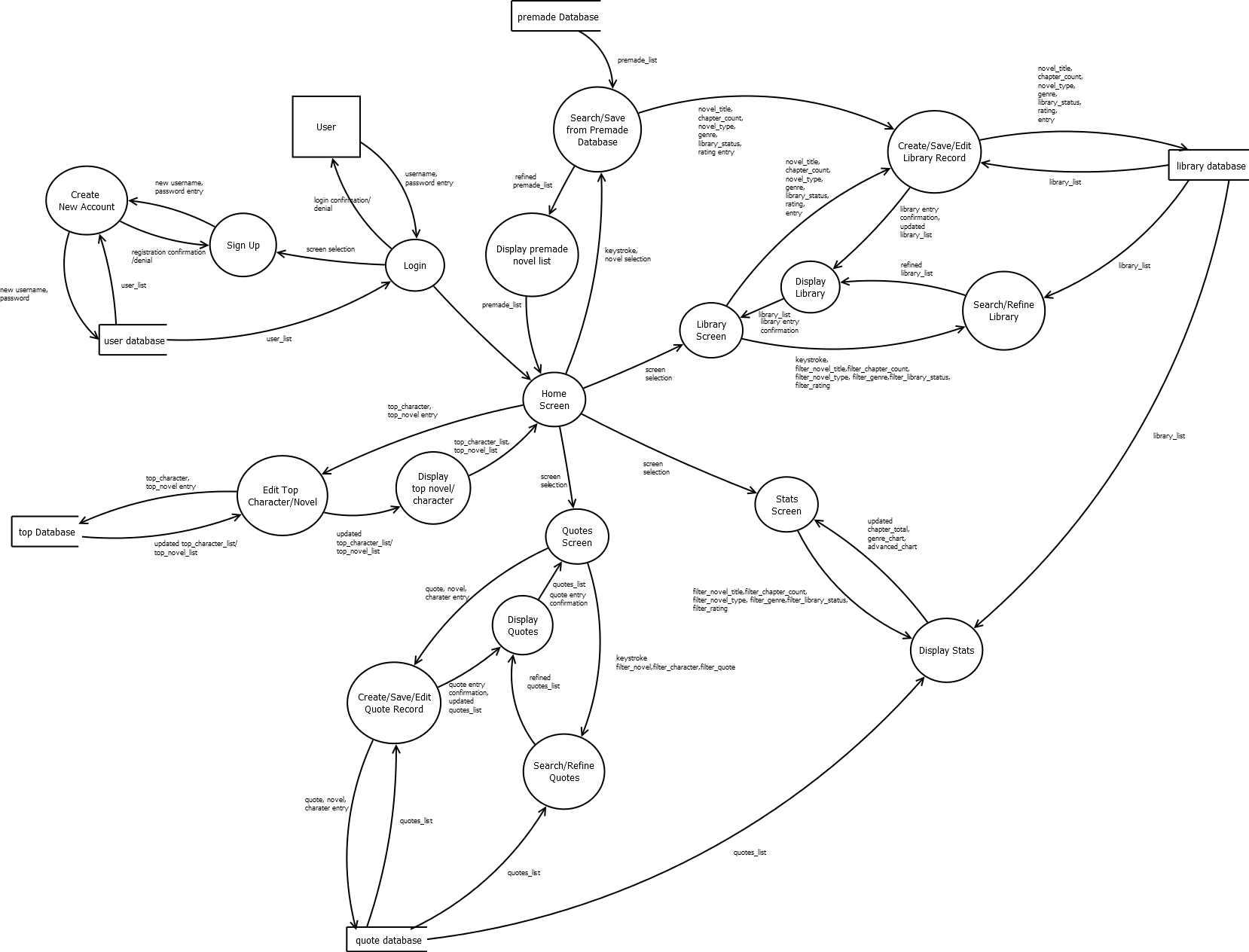
### Data Dictionary:­

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Item | Data Type | Format | Description | Example | Validation |
| username | String |  | Username of the client | ‘SectUncleBai’ |  |
| password | String |  | Password for the client’s account | ‘qwert!123’ |  |
| user\_list | List | username  password | List of user records |  |  |
| novel | String |  | title of the novel in quote entry | ‘A Will Eternal’ |  |
| character | String |  | Name of the character | ‘Bai Xiaochun’ |  |
| quote | String |  | Quotation being recorded/said by a character | ‘reduce you to ashes’ |  |
| quotes\_list | List | novel,  character, quote | List of quotes records stored in quotes csv |  |  |
| novel\_title | String |  | Title of the novel | ‘I Shall Seal the Heavens’ |  |
| chapter\_count | Integer |  | Number of chapters read | ‘1257’ |  |
| novel\_type | String |  | Type of novel (form) | ‘Chinese Novel’ | Restricted Choice:  Manhwa, Manhua, Manga, Chinese/Japanese/  Korean/English Novel |
| genre | String |  | Genre of novel | ‘Xianxia’ |  |
| library\_status | String |  | Status of novel’s reading completion | ‘Completed’ | Restricted Choice:  Currently Reading, Plan to Read, Completed, On Hold, Dropped |
| rating | Integer | NN | Numerical rating of the novel from 1 to 10 | ‘10’ | rating >= 0  rating <= 10 |
| library\_list | List | novel\_title,  chapter\_count,  novel\_type,  genre,  library\_status,  rating | List of library records stored in library csv |  |  |
| chapter\_total | Integer |  | Sum of total chapter\_count for all entries in library\_list | ‘35000’ | Chapter\_total>=0 |
| genre\_chart | Array |  | Sum of genre as a percentage of genre types |  |  |
| advanced\_chart | Array |  | library\_list refined for filter criteria |  |  |
| keystroke | String |  | Characters entered into the ‘Search’, ‘Library’ and ‘Quotes’ search bars | ‘Manhua’ |  |
| top\_character | String |  | Name of Top character | ‘Meng Hao’ |  |
| top\_novel | String |  | Title of Top novel | ‘Beseech the Devil’ |  |
| top\_character\_list | List | top\_character,  top\_character,  top\_character | List of top\_character stored in top csv |  |  |
| top\_novel\_list | List | top\_novel,  top\_novel,  top\_novel | List of top\_novel stored in top csv |  |  |
| premade\_list | List | novel\_title,  novel\_type,  genre | List of novel records stored in premade csv |  |  |
| filter\_novel | String |  | Filter option for novel | ‘Ascending’ |  |
| filter\_character | String |  | Filter option for character | ‘Ascending’ |  |
| filter\_quote | String |  | Filter option for quote | ‘Ascending’ |  |
| filter\_novel\_title | String |  | Filter option for the novel\_title | ‘Ascending’ |  |
| filter\_chapter\_count | Integer |  | Filter option for the chapter\_count | ‘Ascending’ |  |
| filter\_novel\_type | String |  | Filter option for the novel\_type | ‘Ascending’ |  |
| filter\_genre | String |  | Filter option for the genre | ‘Ascending’ |  |
| filter\_library\_status | String |  | Filter option for library\_status | ‘Ascending’ |  |
| filter\_rating | Integer |  | Filter option for rating | ‘Ascending’ |  |
| all\_fields\_full | Boolean |  | Describes whether all the entry fields are populated. | ‘True’ | True or False |

### Context Diagram:

Diagram

Description automatically generated

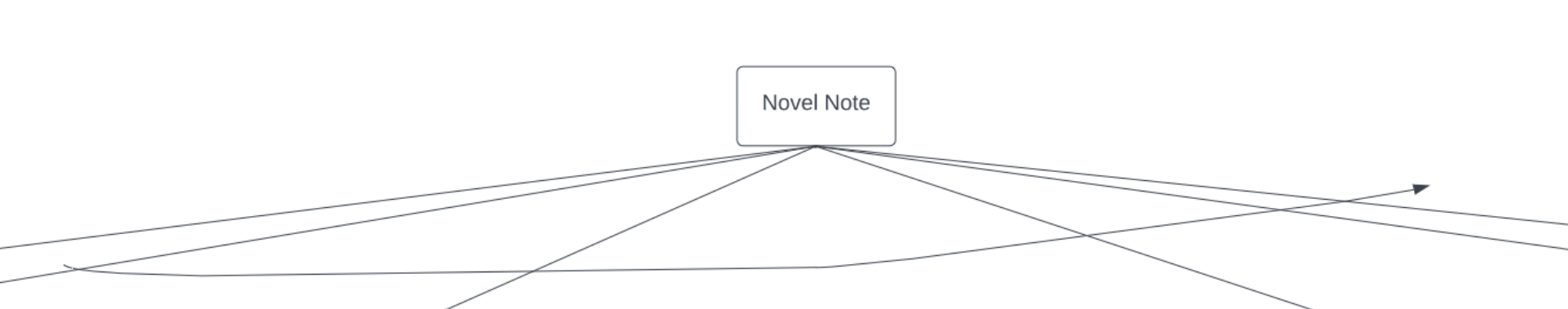
Data Flow Diagram:

### Structure Chart:



Due to the size of this chart, it has been broken down further below so that it is more readable. This enhanced version is identical to the above and is formatted from left to right.

Structure Chart Top:



Diagram

Description automatically generatedStructure Chart 1/4:

Diagram

Description automatically generatedStructure Chart 2/4:

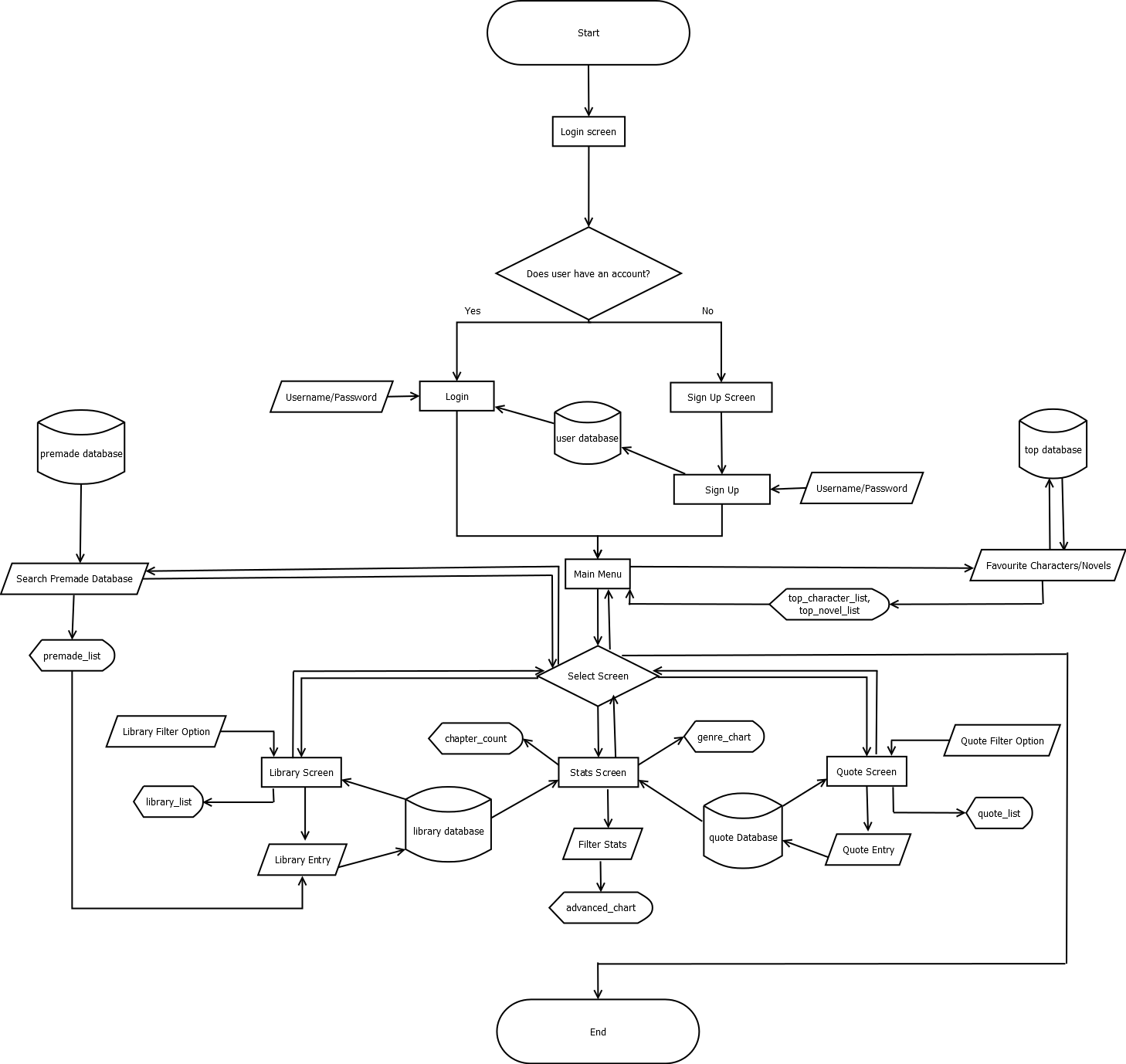
Diagram

Description automatically generatedStructure Chart 3/4:

Diagram

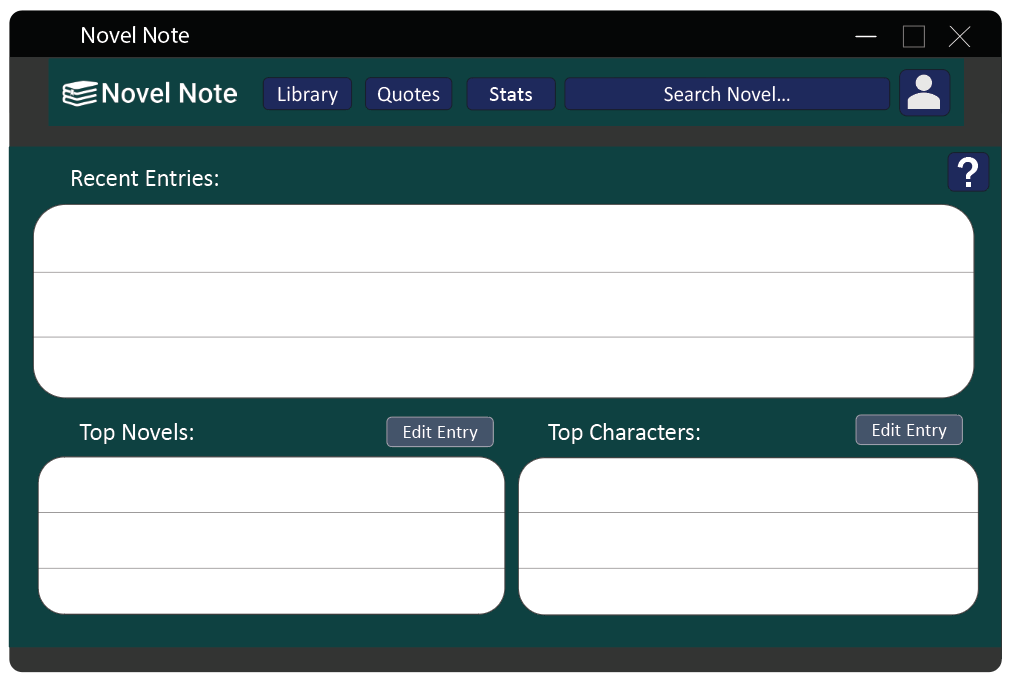
Description automatically generatedStructure Chart 4/4:

### System Flowchart:



### UI Design:

Home Screen:



Quotes Screen:

Timeline

Description automatically generated

Library Screen:

A picture containing timeline

Description automatically generated

Stats Screen:

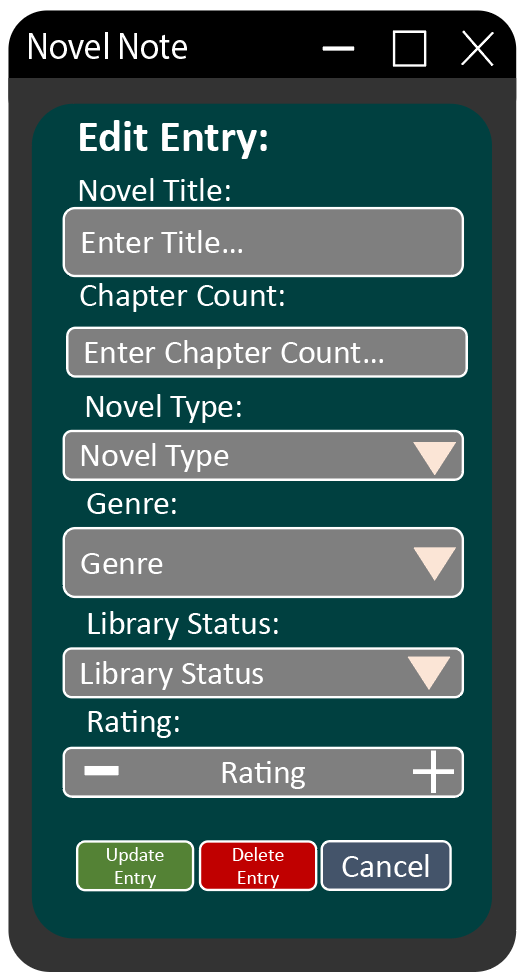
Graphical user interface, application

Description automatically generated

Search Screen:

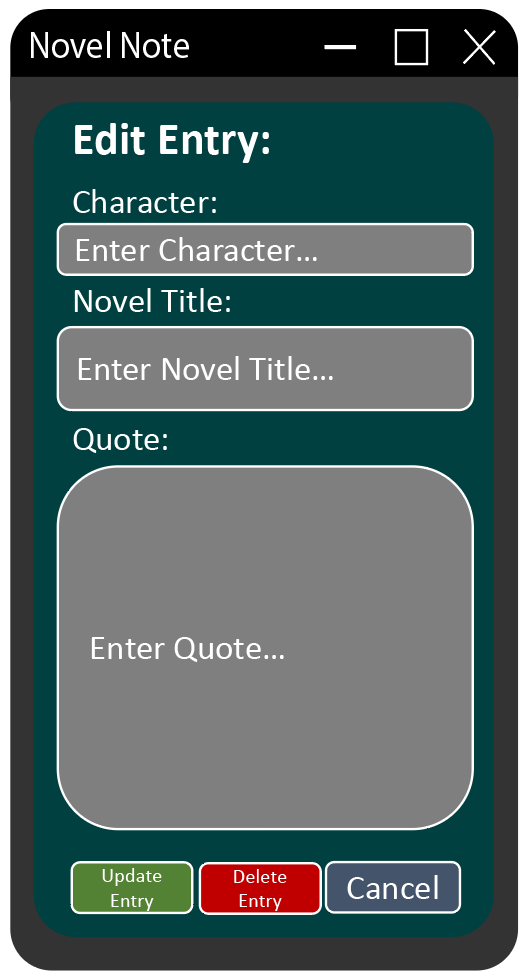
Table

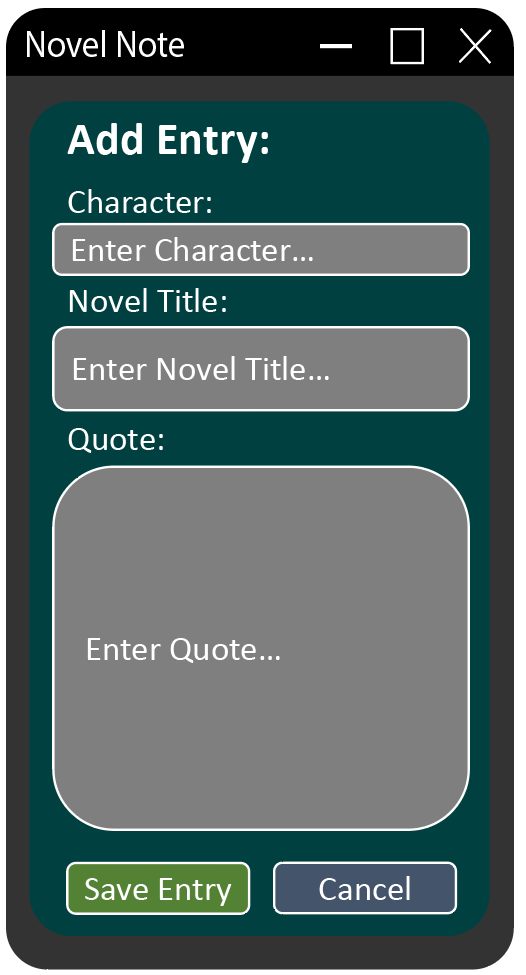
Description automatically generated

Add Library Entry Popup: Edit Library Entry Popup:

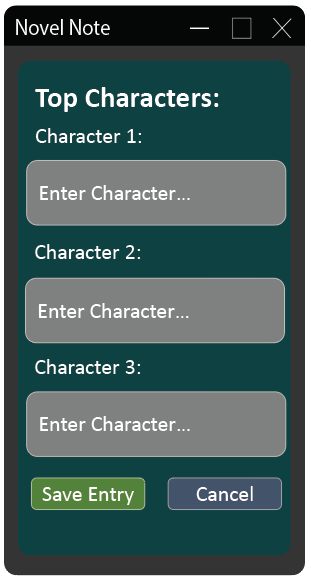
A screenshot of a phone

Description automatically generated with medium confidence

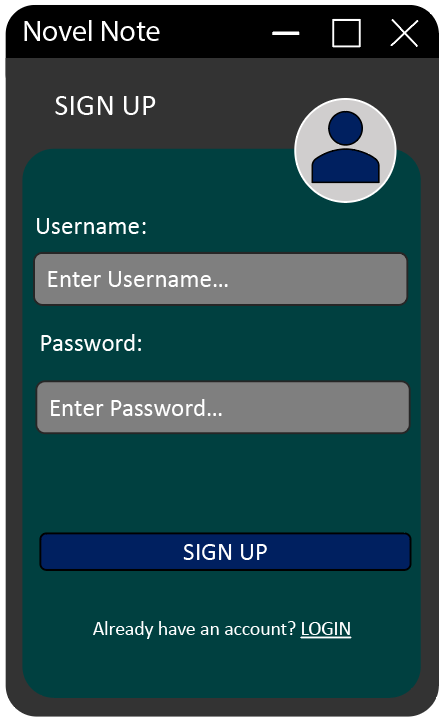
Add Quote Entry Popup: Edit Quote Entry Popup:

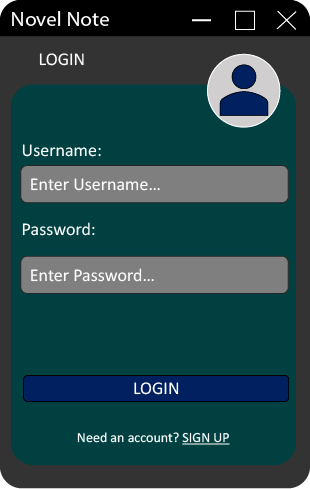


Top Novel Popup: Top Character Popup:

A screenshot of a phone

Description automatically generated with medium confidence

Login Screen: Sign Up Screen:



### Storyboard:

Diagram

Description automatically generated



# **Social and Ethical Issues**

## **Social and Ethical Issues Overview**

### Relevant Social and Ethical Issues:

Software developers have an obligation to users of their software products, that their product is safe to use, able to be used as intended, and has a satisfactory user experience. A software developer who does not adhere and compensate for social and ethical issues, including IP, Quality, Inclusivity, Privacy and Ergonomics, has failed to follow the software developers code of conduct, and as such has failed to deliver a suitable solution to their user.

As a novel logging app, Novel Note is an application built around data input, storage, and output. As such, the application is subject to the scrutiny of various social and ethical issues. Below are a few of the most prolific:

### Intellectual Property:

-Intellectual Property refers to the ‘fruits of mental labour’ and is a legal concept which safeguards the creation and use of software in an attempt to protect the developer from copyright concerns.

-In the context of a software development project, intellectual property is a prominent concern, to ensure that the developer does not illegally utilise components which are protected under law and to ensure that created code and systems are not illegally used by external parties. Since Novel Note is being created for a real-world client for commercial purposes, breaches in intellectual property are illegal.

### Privacy:

-Privacy is ensuring the protection of various data types utilised and input by the user to ensure that it is not stolen or illegally used.

-The primary concern for Novel Note is privacy. The application utilizes various types of personal and sensitive data to function; thus, it is imperative that this data is protected and safeguarded against outside threat. The application has a total of five (5) separate databases: library, quotes, user, top and premade, each of which handle important information about the system, particularly the User database which records the username and passwords of all users of the system. This information may be misused or used with malicious intentions if privacy conventions/regulations are not followed.

### Quality:

-Software quality is ‘a field of study and practice that describes the desirable attributes of software products. Quality gives insight into the effectiveness of a computer system in performing correctly, as intended by the user requirements.

-A software solution of high quality is desirable as it best suits the needs of the user, thus the quality of Novel Note is of vital importance. Quality consists of components including reliability (does the software work all the time?), compatibility (does the software work with other systems?), and security (is privacy guaranteed?). Novel Note must be of high quality for it to be effective in its purpose and to satisfy the client.

### Inclusivity:

-Inclusivity is a concept where, ‘no individuals are excluded or marginalised from use of the software including those having physical or intellectual disabilities or belonging to other minority groups.

-In the context of Novel Note, inclusivity is a lesser priority due to its nature as a personalised system. Nevertheless, inclusivity must also be ensured so that the accessibility of the software has continuity in the event that multiple users of differing backgrounds utilised the product, to ensure they have a quality user experience.

### Ergonomics:

-Ergonomics is the interaction between humans and their environment. Ergonomics is of vital importance for all software applications as they offer a connection between a computer and a user, thus the ergonomics of the system is imperative in creating a sustained and effective experience.

-In Novel Note ergonomics is a vital concern as the client expressed a desire for a minimalistic and intuitive design, thus it is an imperative that a suitable user interface is designed to minimise any ergonomic issues related to a poor user experience and keep useability of the software as an utmost priority.

## **Justification of Social and Ethical Issues in Relation to Major Project**

In order to create a suitable software solution which adheres to the guidelines of the HSC major project rubric and meets the client’s requirements, it is imperative that social and ethical issues are met with relevant, and effective methods to minimise negative effects on the project. In particular, it is important that Intellectual Property, Quality, Ergonomics, Inclusivity and Privacy are identified and properly controlled. The primary method of ensuring these issues are controlled will be by following of the principles of the IEEE Computer Society “Code of Ethics”, in particular the following:

1. PUBLIC – Software engineers shall act consistently with the public interest.

2. CLIENT AND EMPLOYER – Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.

3. PRODUCT – Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

4. JUDGMENT – Software engineers shall maintain integrity and independence in their professional judgment.

5. MANAGEMENT – Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

6. PROFESSION – Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

*Code of ethics: IEEE Computer Society Code of Ethics | IEEE Computer Society. Available at:* [*https://www.computer.org/education/code-of-ethics*](https://www.computer.org/education/code-of-ethics)*.*

Some more specific methods include:

Intellectual Property:

-Ensuring that all work (including documentation, portfolio, designs, and code) are my own intellectual property, and that anything created by a third party will be appropriately recognised and obtained legally and ethically. E.g., Citations for code within code comments.

Privacy:

-Ensuring that data collected from the user is secure and free from outside influence which may use it for malicious purposes. This includes strategies such as the use of secure data storage platforms and user profiles protected by passwords.

Quality:

-Ensuring that any limitations to the scope are properly communicated to the client to ensure that the user requirements are able to be correctly followed, and any resulting adjustments are appropriately accommodated. Additionally, adequate testing, documentation and security measures will be implemented at an acceptable cost and schedule to suit the needs of the client and develop a satisfactory system.

Inclusivity:

-The system will be appropriately modified to increase the inclusivity of the system wherever possible, including suitable, sparingly used colour combinations for the client’s colour deficiency, and a help button on each major page to assist in the literacy of the application. Furthermore, general accessibility improvements including unrestricted username input for various language types and conventions and no cost for the economically insecure.

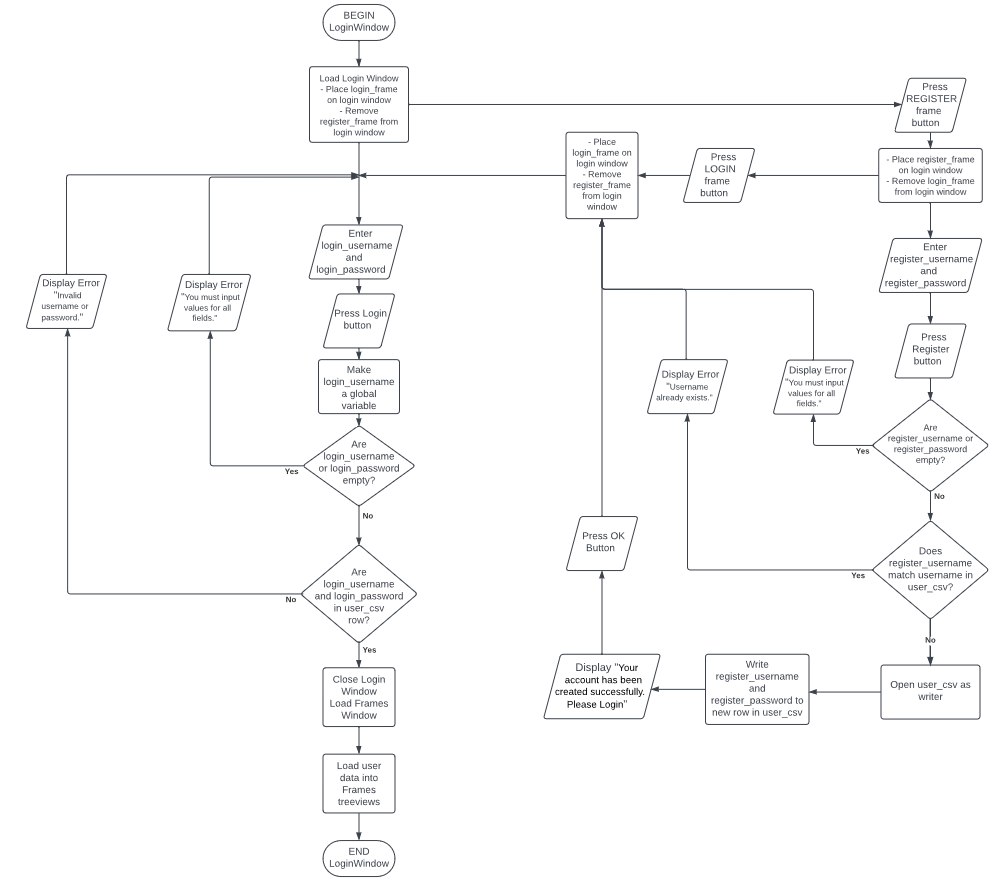
Ergonomics:

-The solution will be developed with a central emphasis on the user experience to ensure that the design meets the needs of the end user and is intuitive and adaptable in use. This includes the implementation ­­of consistent colour schemes, the appropriate use of white space, sustained design paradigms, appropriate font and the continual communication and testing of the interface with the client. This will ensure that the user interface is useable and satisfactory.

# Implementing the Solution

## Algorithms

### **Login:**

**Flowchart:**

**Pseudocode:**

BEGIN Login

Get users\_csv

Load Login window

Place login\_frame

Forget register\_frame

Enter login\_username

Enter login\_password

Press Login Button

Make login\_username a global variable

IF login\_username or login\_password = “” THEN

Display “You must input values for all fields”

ELIF login\_username and login\_password not in users\_csv row THEN

Display “Invalid username or password”

ELSE

Close Login window

Load Frames window

InsertIntoFramesTreeviews(user\_data)

Press Register Screen Button

Place register\_frame

Forget login\_frame

Enter register\_username

Enter register\_password

Press Register Button

IF register\_username or register\_password = “” THEN

Display “You must input values for all fields”

ELIF register\_username matches username in users\_csv THEN

Display “Username already exists”

ELSE

With csv opened as Writer

Writer.writerow(username, password)

Display “Your account has been created successfully. Please Login”

Press Ok Button

Place login\_frame

Forget register\_frame

Press Login Screen Button

Place login\_frame

Forget register\_frame

END Login

### **Search Functions:**

**Flowchart:**

**A diagram of a flowchart

Description automatically generated**

**Pseudocode:**

BEGIN Search

Get username, csv, treeview

Enter search\_text

reader = ReadCSVIntoIterable(csv)

matching\_rows = “”

IF Search\_Text = not “”

matching\_rows = rows in reader which include search\_text in column >= 2, and column(username) = username

ELSE

matching\_rows = rows in reader where column(username) = username

DeleteContents(treeview)

InsertIntoTreeview(matching\_rows)

END Search

### **Select Frame:**

**Flowchart:**

**Pseudocode:**

BEGIN Select Frame

Enter frame\_selection

frames\_window = main application window

Home = home\_frame

Library = library\_frame

Quotes = quote\_frame

Stats = stats\_frame

Search = search\_frame

Frame\_Name = NameOfFrame(frame\_selection)

IF Frame\_Name = Home THEN

Place home\_frame on frames\_window

ELSE

Remove home\_frame from frames\_window

IF Frame\_Name = Library THEN

Place library\_frame on frames\_window

ELSE

Remove library\_frame from frames\_window

IF Frame\_Name = Quotes THEN

Place quotes\_frame on frames\_window

ELSE

Remove quotes\_frame from frames\_window

IF Frame\_Name = Stats THEN

Place stats\_frame on frames\_window

ELSE

Remove stats\_frame from frames\_window

IF Frame\_Name = Search THEN

Place search\_frame on frames\_window

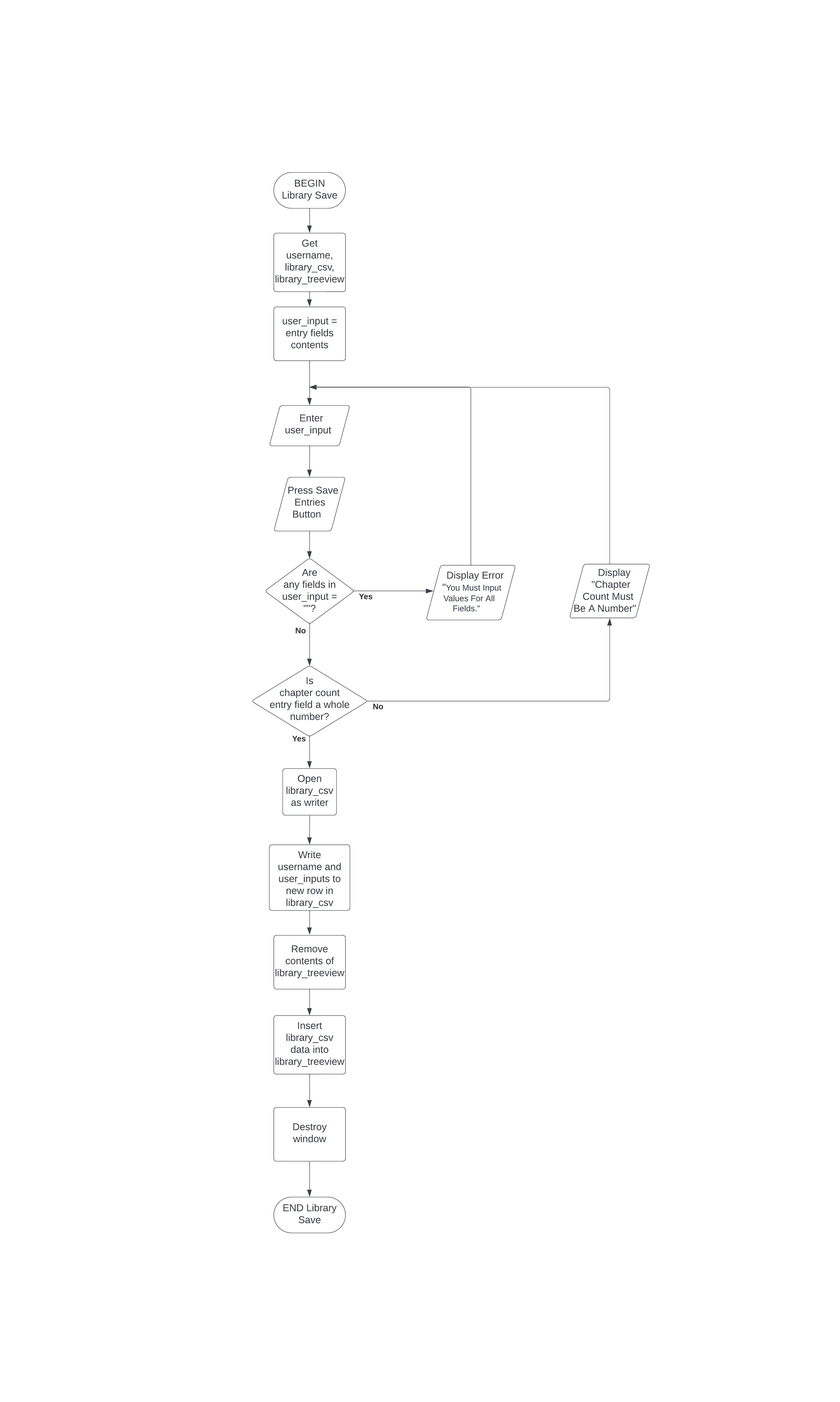
ELSE

Remove search\_frame from frames\_window

END Select Frame

### **Save Entry:**

**Flowchart:**



**Pseudocode:**

BEGIN Library Save

Get username, library\_csv, library\_treeview

user\_input = entry field contents

Enter user\_input

IF any field in user\_input = “” THEN

Display error “You Must Input Values For All Fields.”

ELIF chaptercount\_entry field is not whole number THEN

Display “Chapter Count Must Be A Number”

ELSE

With library\_csv opened as Writer

Writer.writerow(username, user\_input)

DeleteContents(library\_treeview)

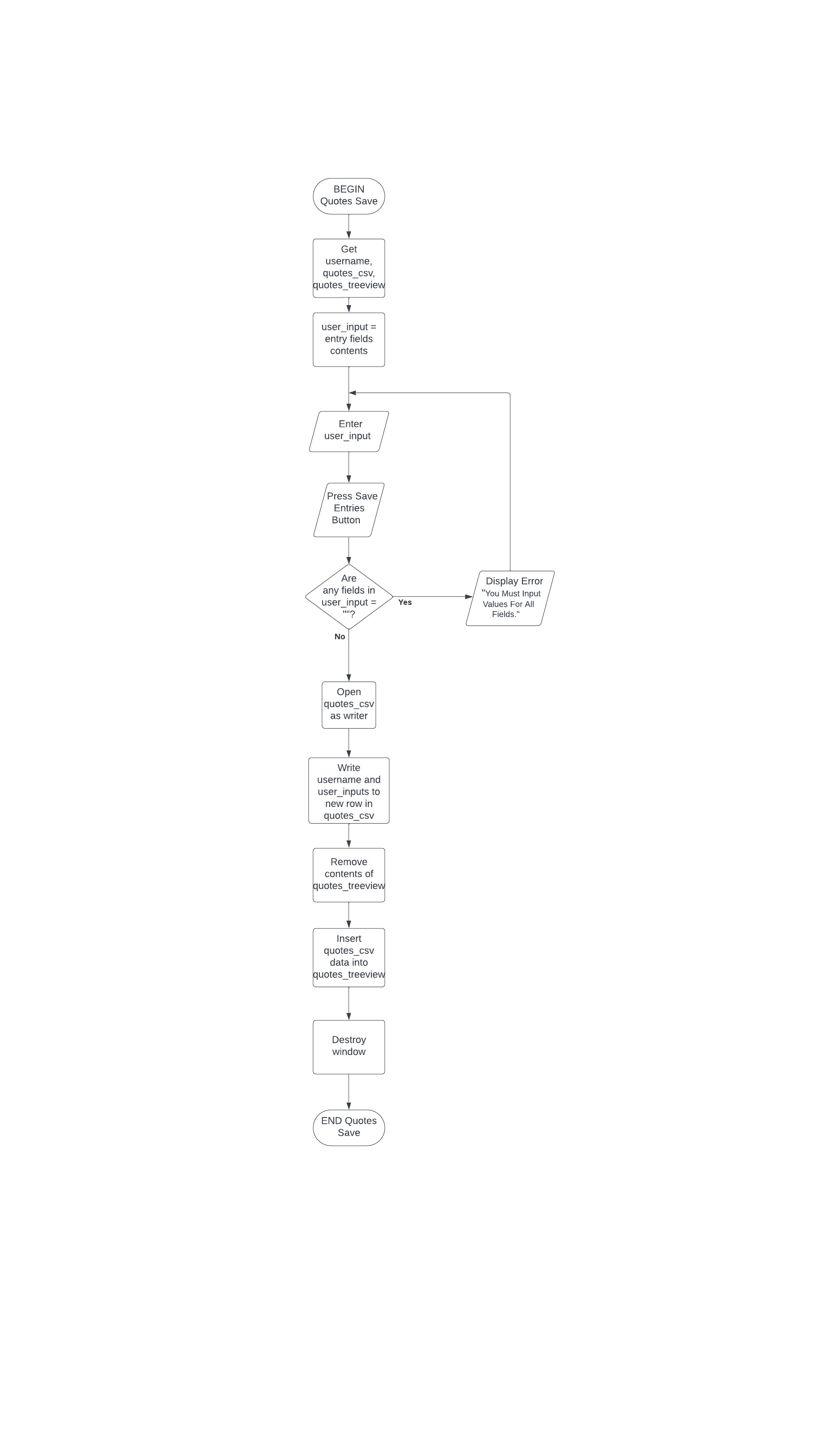
InsertIntoTreeview(library\_csv)

Destroy window

END Library Save

### **Save Quotes Entry:**

**Flowchart:**



**Pseudocode:**

BEGIN Quotes Save

Get username, quotes\_csv, quotes\_treeview

user\_input = entry field contents

Enter user\_input

IF any field in user\_input = “” THEN

Display error “You Must Input Values For All Fields.”

ELSE

With quotes\_csv opened as Writer

Writer.writerow(username, user\_input)

DeleteContents(quotes\_treeview)

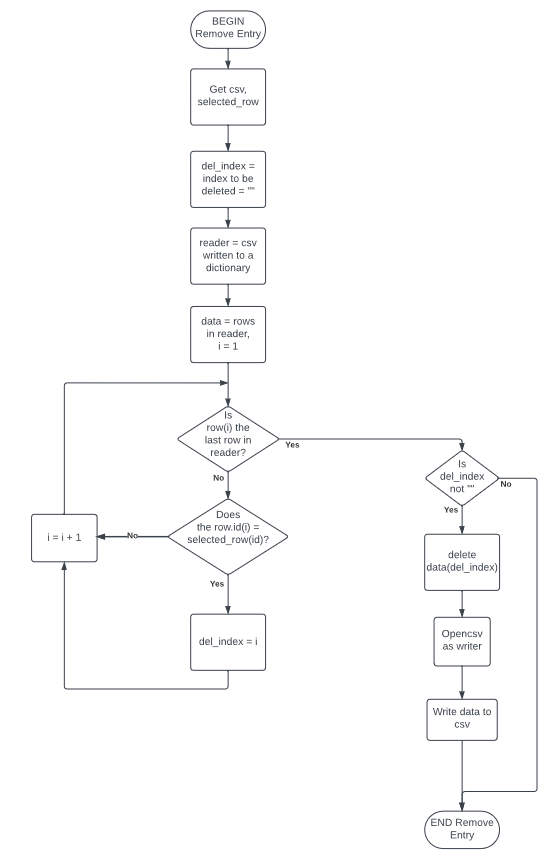
InsertIntoTreeview(quotes\_csv)

Destroy window

END Quotes Save

### **Remove Entry:**

**Flowchart:**



**Pseudocode:**

BEGIN Remove Entry

Get csv, selected\_entry

del\_index = index to be deleted = “”

reader = ReadCSVIntoDictionary (csv)

data = rows in reader

i = 1

WHILE row(i) is not last row in reader

IF row(id) = selected\_entry(id) THEN

del\_index = i

END IF

i = i + 1

END WHLE

IF del\_index is not “”

Delete row(del\_index) from data

With csv opened as Writer

Writer.writerow(data)

END IF

END Remove Entry

### **Library Treeview Insert:**

**Flowchart:**

**A diagram of a flowchart

Description automatically generated**

**Pseudocode:**

BEGIN Library Treeview Insert

Get username, library\_csv, library\_treeview

df = ReadCSVIntoDataFrame (library\_csv)

i = 1

WHILE row(i) in df is not last row

IF username = df.username(i) THEN

Insert row(i) into library\_treeview

END IF

i = i + 1

ENDWHILE

END Library Treeview Insert

### **Quotes Treeview Insert:**

**Flowchart:**

**A diagram of a flowchart

Description automatically generated**

**Pseudocode:**

BEGIN Quote Treeview Insert

Get username, quote\_csv, quote\_treeview

df = ReadCSVIntoDataFrame (quote\_csv)

i = 1

WHILE row(i) in df is not last row

IF username = df.username(i) THEN

Insert row(i) into quote\_treeview

END IF

i = i + 1

ENDWHILE

END Quote Treeview Insert

A white background with black dots

Description automatically generated

### **TopNovel Treeview Insert:**

**Flowchart:**

**Pseudocode:**

BEGIN TopNovel Treeview Insert

Get username, topnovel\_csv, topnovel\_treeview

df = ReadCSVIntoDataFrame (topnovel\_csv)

RemoveContents(topnovel\_treeview)

novel\_number = “”

novel\_value = “”

i = 1

j = 1

WHILE row(i) in df is not last row

IF username = df.username(i) THEN

IF j < 11 THEN

novel\_number = “novel \_” + j

novel \_value = df.novel\_number

IF novel \_value is not “” THEN

value = CreateTuple(novel\_value)

Insert value into topnovel\_treeview

END IF

j = j + 1

END IF

END IF

i = i +1

ENDWHILE

END TopNovel Treeview Insert

### **TopCharacter Treeview Insert:**

A close-up of a diagram

Description automatically generated**Flowchart:**

**Pseudocode:**

BEGIN TopCharacter Treeview Insert

Get username, topcharacter\_csv, topcharacter\_treeview

df = ReadCSVIntoDataFrame (topcharacter\_csv)

RemoveContents(topcharacter\_treeview)

character\_number = “”

character\_value = “”

i = 1

j = 1

WHILE row(i) in df is not last row

IF username = df.username(i) THEN

IF j < 11 THEN

character\_number = “novel \_” + j

character \_value = df.character\_number

IF character\_value is not “” THEN

value = CreateTuple(character\_value)

Insert value into topcharacter\_treeview

END IF

j = j + 1

END IF

END IF

i = i +1

ENDWHILE

END TopCharacter Treeview Insert

### **Recent Entry Treeview Insert:**

**Flowchart:**

**A diagram of a flowchart

Description automatically generated**

**Pseudocode:**

BEGIN Recent Entry Treeview Insert

Get username, recent\_csv, recent\_treeview

df = ReadCSVIntoDataFrame (recent\_csv)

RemoveContents(recent\_treeview)

filtered\_dataframe = “”

i=1

WHILE row(i) in df is not last row

IF username = df.username(i) THEN

Add row(i) to filtered\_dataframe

END IF

i = i + 1

ENDWHILE

filtered\_dataframe = ReverseDataFrameOrder(filtered\_dataframe)

filtered\_dataframe = RemoveRowsOutOfTop5(filtered\_dataframe)

i = 0

WHILE i < 6

Insert filtered\_dataframe row(i) into recent\_treeview

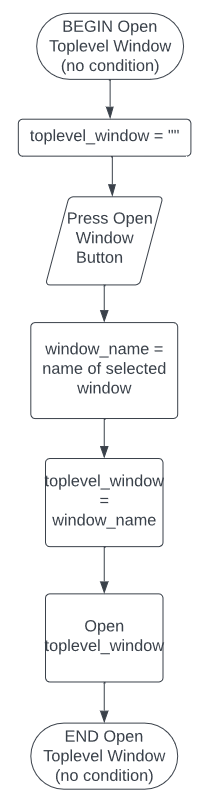
i = i + 1

ENDWHILE

END Recent Entry Treeview Insert

### **Open Toplevel Window (no condition):**

**Flowchart:**

****

**Pseudocode:**

BEGIN Open Toplevel Window (no condition)

Select window

toplevel\_window = “”

window\_name = NameOfWindow(selected window)

Press Open Window button

toplevel\_window = window\_name

Open toplevel\_window

END Open Toplevel Window (no condition)

### **Open Toplevel Window (selection condition):**

**Flowchart:**

**A diagram of a flowchart

Description automatically generated**

**Pseudocode:**

BEGIN Open Toplevel Window (selection condition)

Get treeview

toplevel\_window = “”

Select window, row

window\_name = NameOfWindow(selected window)

selection = TreeviewContents(selected row)

Press Open Window button

IF selection = “” THEN

Display “You Must Select An Entry”

ELSE

toplevel\_window = window\_name

Open toplevel\_window

Populate toplevel\_window with selection

END Open Toplevel Window (selection condition)

### **Choose Chart:**

**Flowchart:**

**A diagram of a flowchart

Description automatically generated**

**Pseudocode:**

BEGIN Choose Chart

Get piechart, typechart, statuschart

Place piechart on piecanvas

Place typechart on typecanvas

Place statuschart on statuscanvas

Enter chart\_selection

IF chart\_selection = piechart THEN

Place piecanvas on window

Remove typecanvas from window

Remove statuscanvas from window

ELIF chart\_selection = typechart THEN

Place typecanvas on window

Remove piecanvas from window

Remove statuscanvas from window

ELSE chart\_selection = statuschart

Place statuscanvas on window

Remove piecanvas from window

Remove typecanvas from window

END Choose Chart

### A diagram of a flowchart Description automatically generated**Filter:**

**Flowchart:**

****

**Pseudocode:**

BEGIN Filter

Get username, library\_treeview, library\_csv

Enter genre\_selection, type\_selection, status\_selection

Enter rating\_selection, chapter\_selection

genre\_selection = genre\_combobox content

type\_selection = type\_combobox content

status\_selection = status\_combobox content

rating\_selectiom = rating\_combobox content

chapter\_selection = chapter\_combobox content

IF rating\_selection = “Ascending” or “Descending” THEN

Set chapter\_combobox to “Any”

RemoveContents(library\_treeview)

reader = ReadCSVIntoIterable(library\_csv)

sort\_order = rating\_selection

rows = rows in reader where reader.genre(genre\_selections), reader.type(type\_selections), reader.status(status\_selections)

sorted\_rows = Sort rows by sort\_order

i = 0

WHILE rows(i) is not last row

IF username = row.username(i) THEN

Insert row(i) into treeview

END IF

i = i + 1

END WHILE

ELIF chapter\_selection = “Ascending” or “Descending” THEN

Set rating\_combobox to “Any”

RemoveContents(library\_treeview)

reader = ReadCSVIntoIterable(library\_csv)

sort\_order = chapter\_selection

rows = rows in reader where reader.genre(genre\_selections), reader.type(type\_selections), reader.status(status\_selections)

sorted\_rows = Sort.rows(sort\_order)

i = 0

WHILE rows(i) is not last row

IF username = row.username(i) THEN

Insert row(i) into treeview

END IF

i = i + 1

ENDWHILE

ELSE

RemoveContents(library\_treeview)

reader = ReadCSVIntoIterable(library\_csv)

rows = rows in reader where reader.genre(genre\_selections), reader.type(type\_selections), reader.status(status\_selections)

i = 0

WHILE rows(i) is not last row

IF username = row.username(i) THEN

Insert row(i) into treeview

END IF

i = i + 1

ENDWHILE

END Filter

### **Close/Cancel Window:**

**Flowchart:**

**A diagram of a button

Description automatically generated**

**Pseudocode:**

BEGIN Close/Cancel

Get toplevel\_window\_name

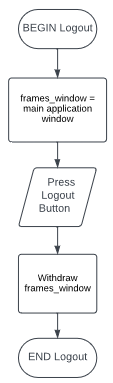
Press Cancel/Close Button

Close toplevel whose name = toplevel\_window\_name

END Close/Cancel

### **Logout Window:**

**Flowchart:**

****

**Pseudocode:**

BEGIN Logout

frames\_window = main application window

Press Logout Button

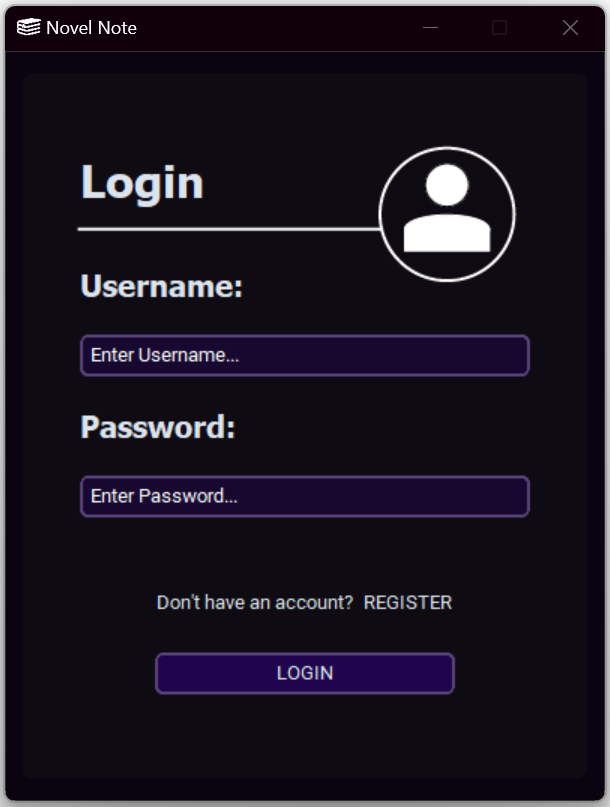
Withdraw frames\_window

END Logout

## User Interface Development

### **Login Window:**

Login Frame:



Register Frame:

A screen shot of a login form

Description automatically generated

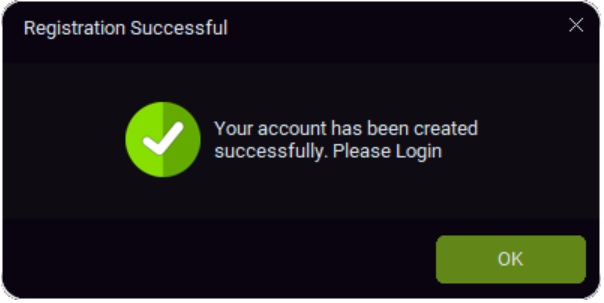
Login Window Help/Error Messages:

A screen shot of a computer screen

Description automatically generated

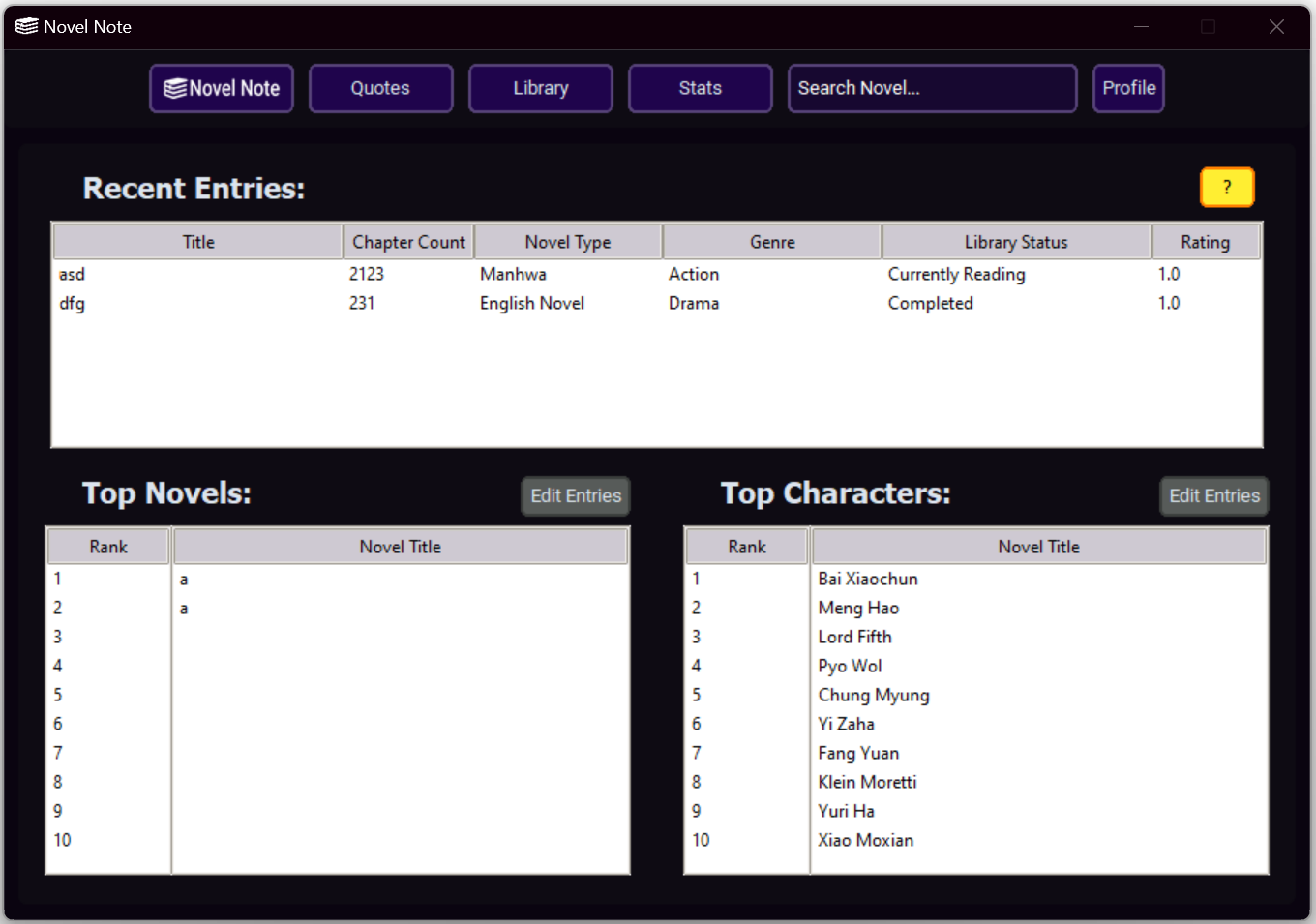
A screen shot of a computer screen

Description automatically generated



### **Home Screen:**

Home Frame:

  
Home Frame Help Message:

A screenshot of a computer

Description automatically generated

Edit Top Novel Entries Window:

A screenshot of a computer

Description automatically generated

Edit Top Novel Entries Window Help/Error Messages:

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

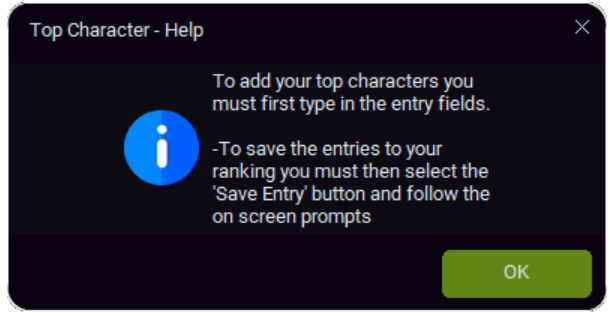
Description automatically generated

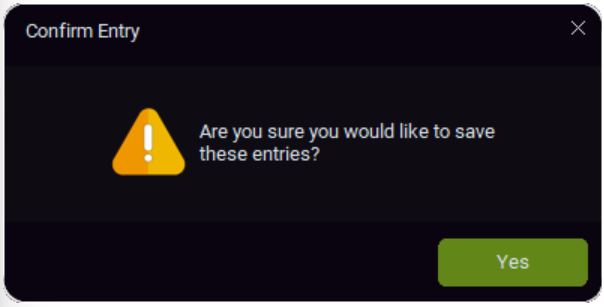
Edit Top Character Entries Window:

A screenshot of a computer

Description automatically generated

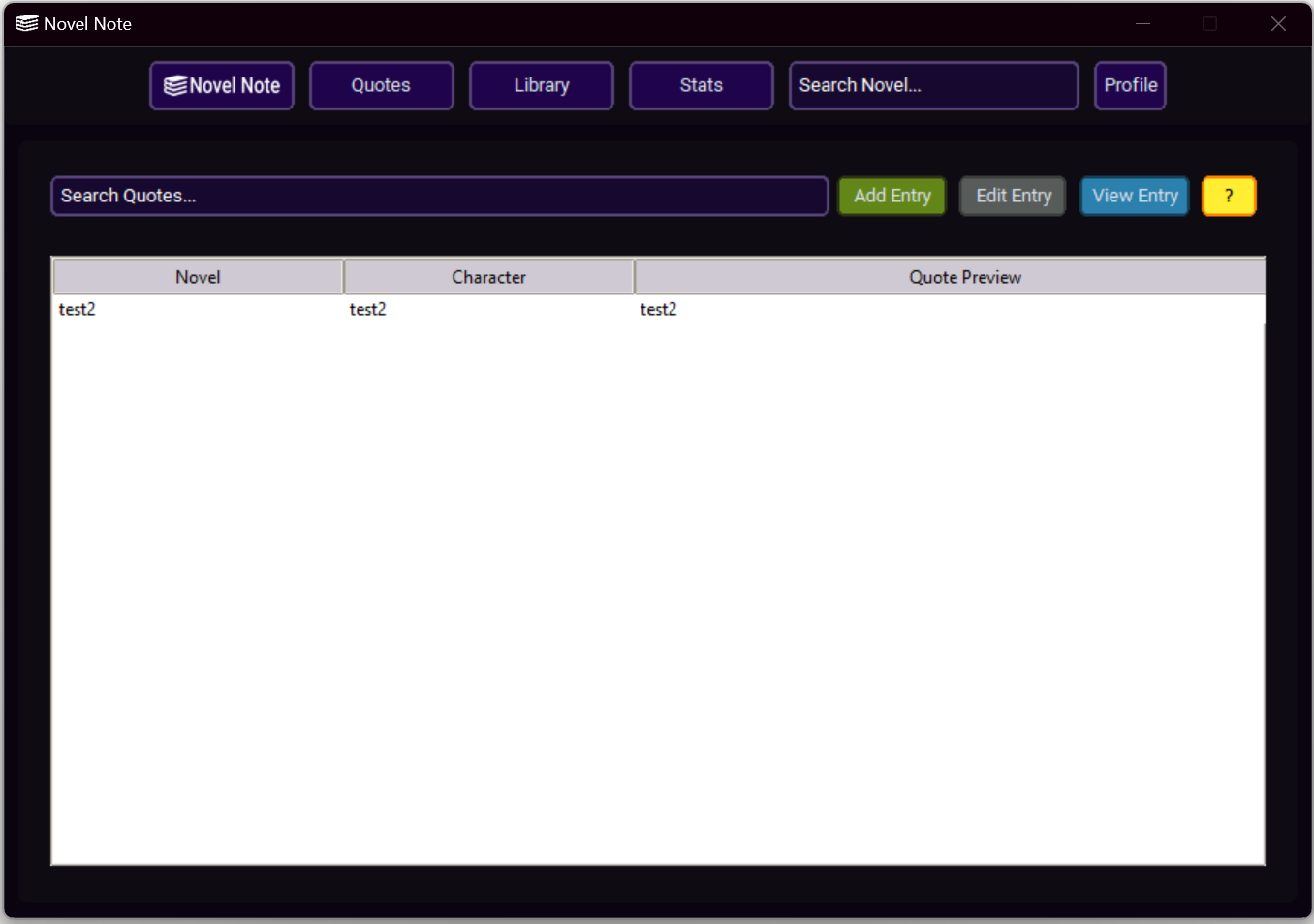
Edit Top Novel Entries Window Help/Error Messages:



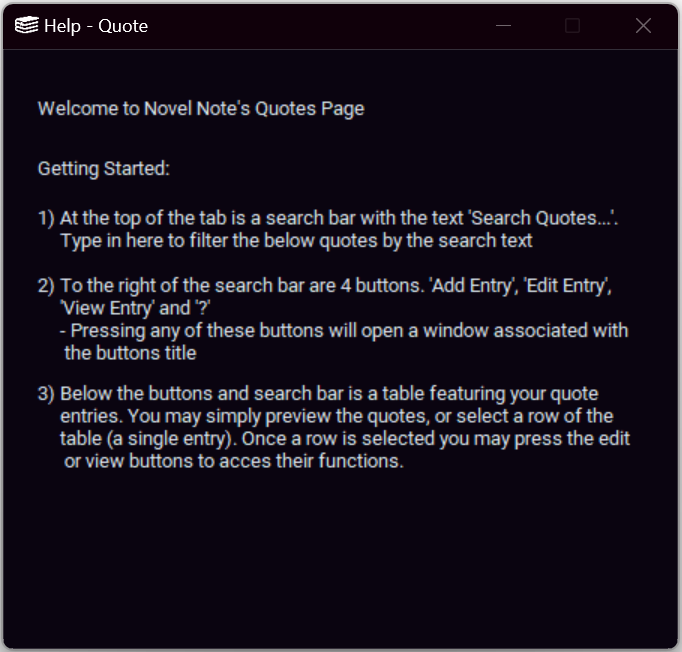


### **Quotes Screen:**

Quotes Frame:



Home Frame Help/Error Messages:



A screen shot of a warning sign

Description automatically generated

Add Quote Entry Window:

A screenshot of a computer

Description automatically generated

Add Quote Entry Help/Error Messages:

A screenshot of a chat

Description automatically generated

A yellow triangle with white text

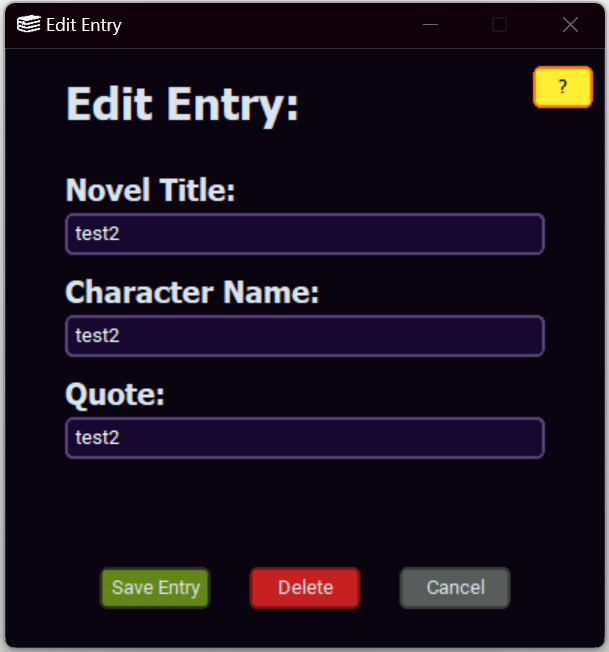
Description automatically generated

Quotes Frame Entry Selection:

A screenshot of a computer

Description automatically generated

Edit Quote Entry Window:



Edit Quote Entry Help/Error Messages:

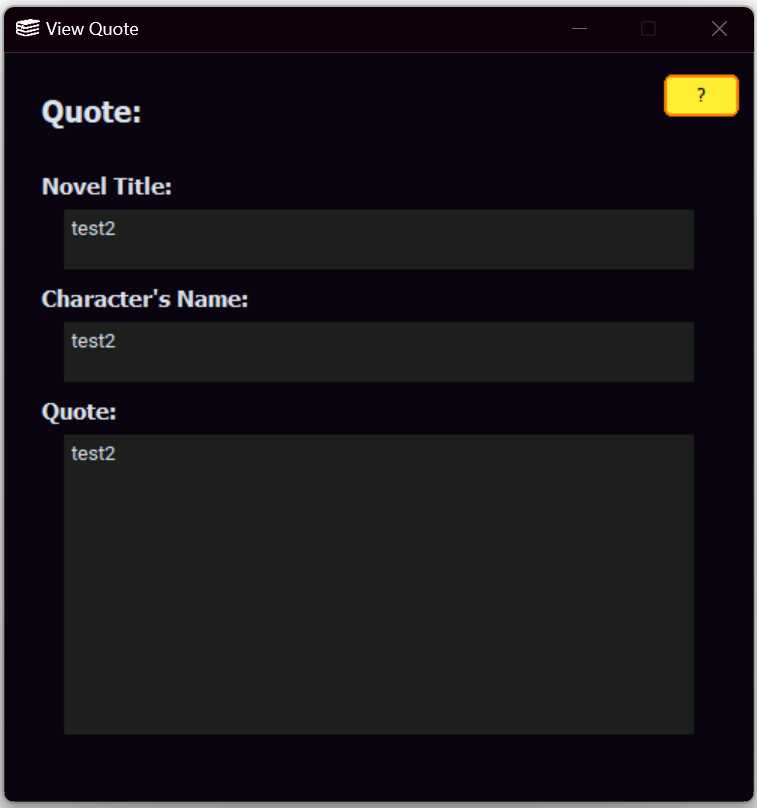
A screenshot of a computer

Description automatically generated

A yellow triangle with white text

Description automatically generated

View Quote Entry Window:



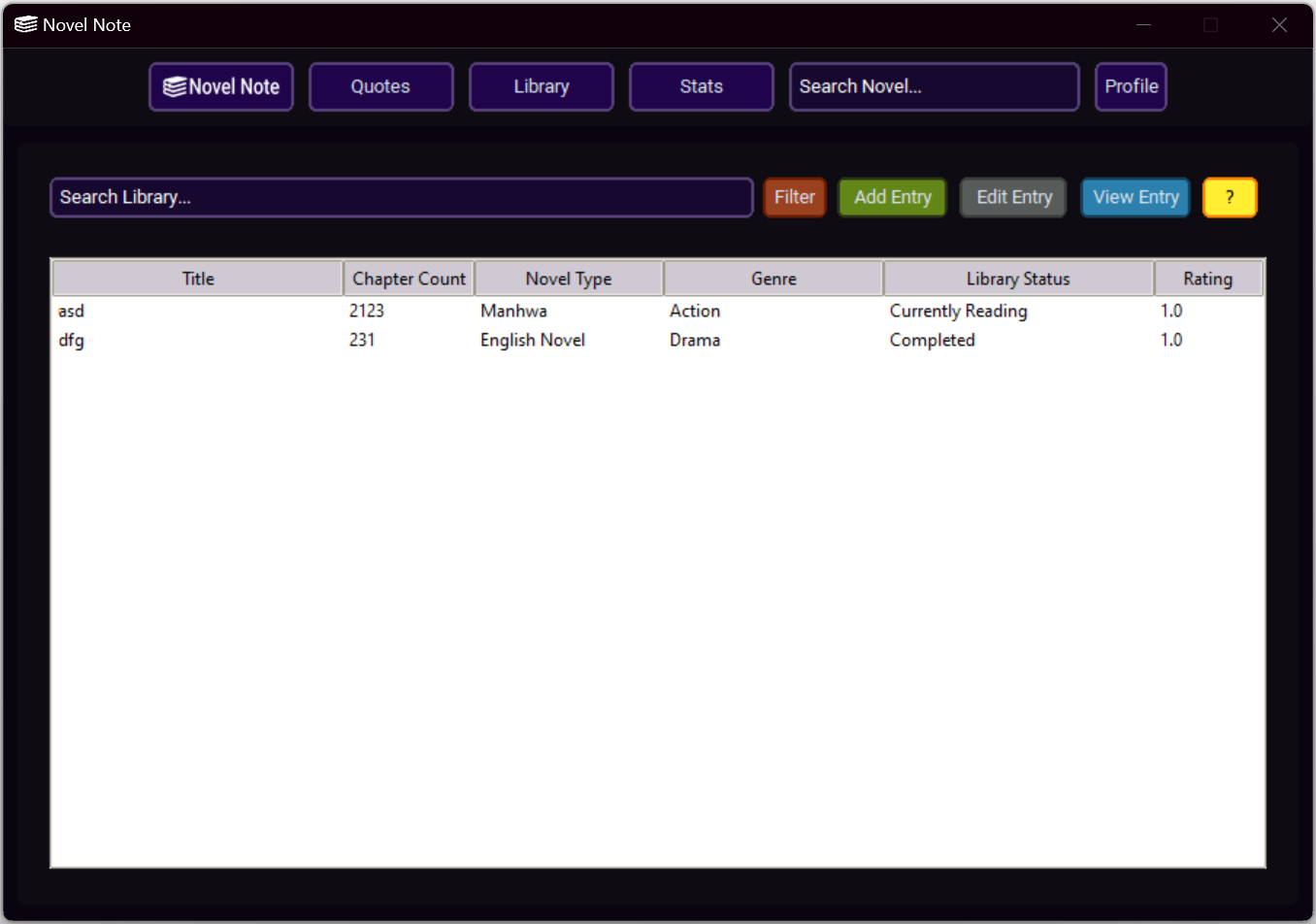
View Quote Entry Help Message:

A screenshot of a computer screen

Description automatically generated

### **Library Screen:**

Library Frame:



Library Frame Help/Error Messages:

A screenshot of a computer

Description automatically generated

A screen shot of a warning sign

Description automatically generated

Filter Library Window:

A screenshot of a computer

Description automatically generated

Filter Library Help Message:

A screenshot of a computer

Description automatically generated

Add Library Entry Window:

A screenshot of a book

Description automatically generated

Add Library Entry Help/Error Messages

A screenshot of a computer

Description automatically generated

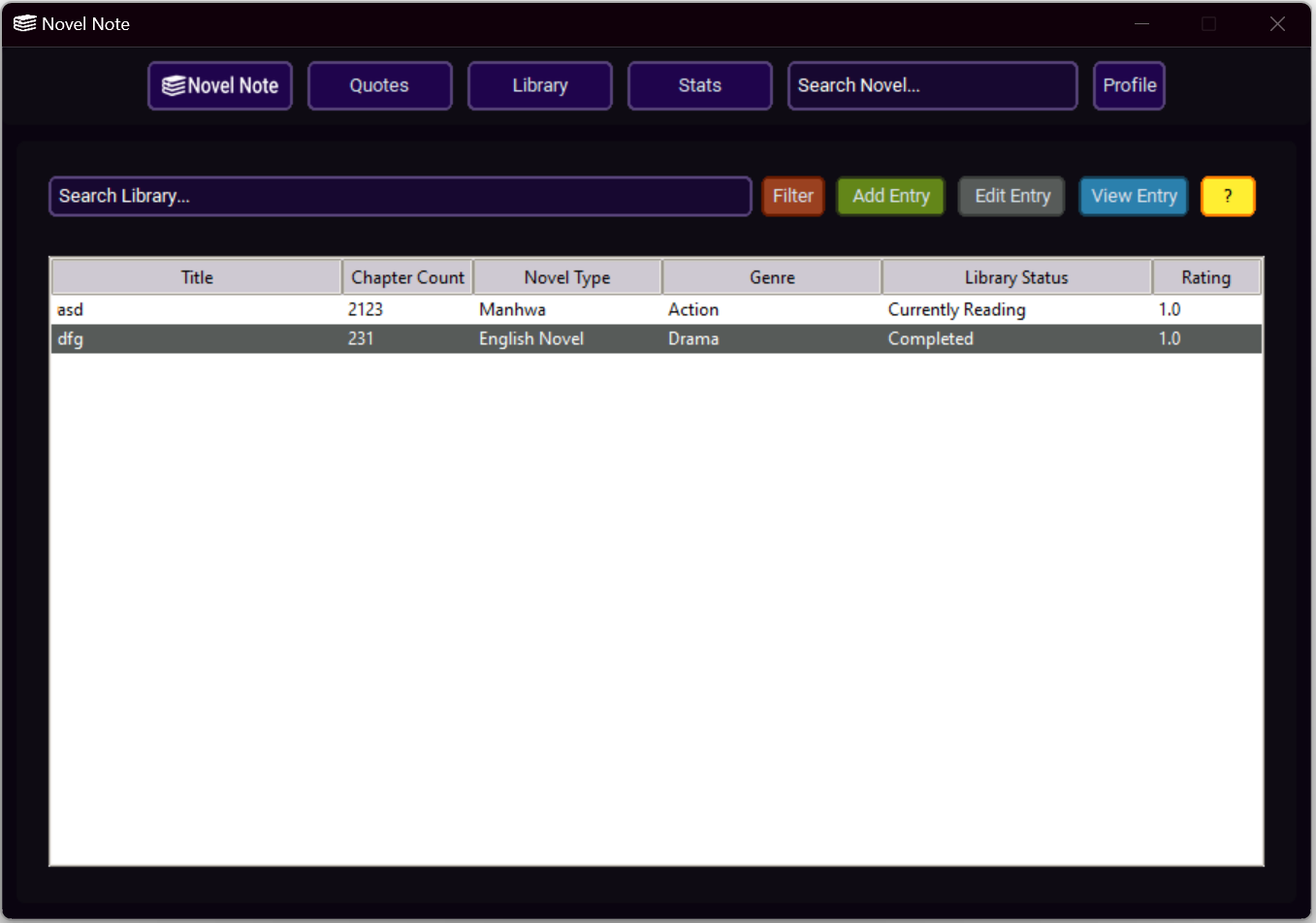
A yellow triangle with white text

Description automatically generated

A screen shot of a computer

Description automatically generated

Library Frame Entry Selection:

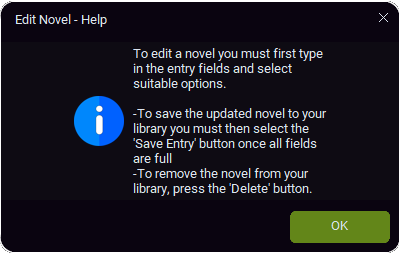


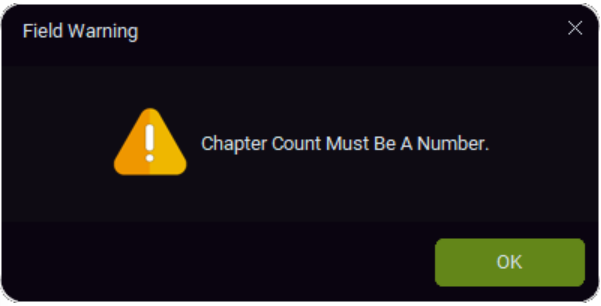
Edit Library Entry Window:

A screenshot of a book

Description automatically generated

Edit Library Entry Help/Error Messages:





View Library Entry Window:

A screenshot of a computer

Description automatically generated

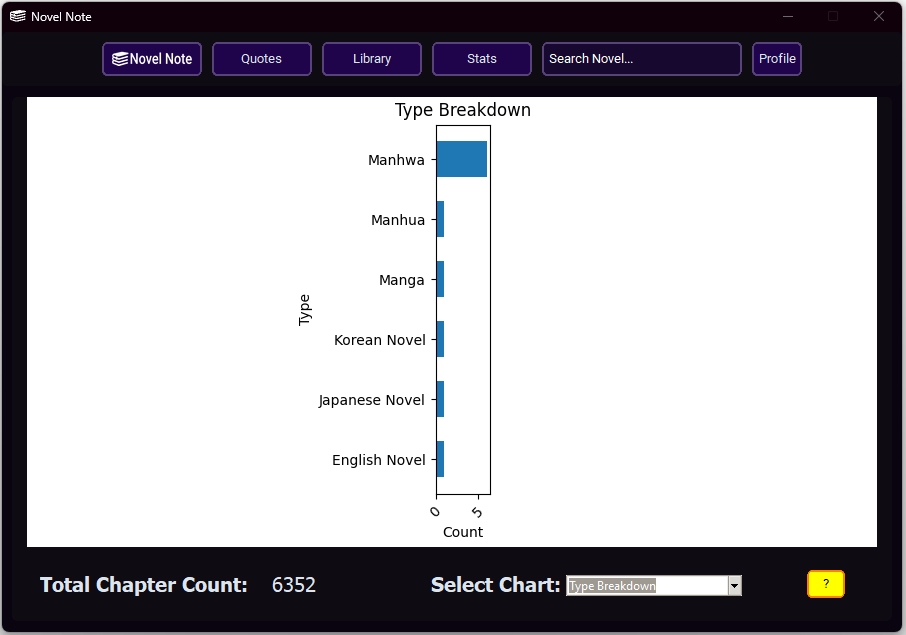
View Library Entry Help Message:

A screenshot of a computer screen

Description automatically generated

### **Stats Screen:**

Type Breakdown Chart Selection:



Status Breakdown Chart Selection:

A screenshot of a computer

Description automatically generated

Genre Breakdown Chart Selection:

A screenshot of a computer

Description automatically generated

Stats Frame Help Window:

A screenshot of a computer screen

Description automatically generated

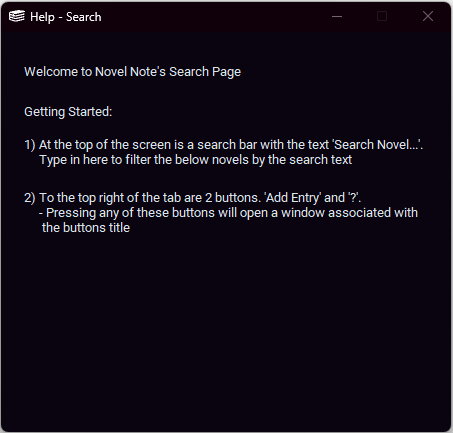
### **Search Screen:**

Search Frame:

A screenshot of a computer

Description automatically generated

Search Frame Help/Error Messages:



A screen shot of a warning sign

Description automatically generated

### **Profile Popup:**

A screenshot of a computer

Description automatically generated

## Source Code

## User Manual

# **Testing and Maintaining**

## Testing Overview

Throughout the development of the Novel Note application, various testing was conducted to ensure the functionality and quality of individual modules and the product as a whole.

In the testing process the following types and levels of testing were conducted:

**Module Level Testing:**

Individual modules were isolated from the main application and test data was run through them to ensure the functionality of the involved functions. For example, the Add Library Entry Window was called individually, and mock data was input and saved. Then the csv was checked to see if the entry was correctly saved. This process was altered and conducted for all relevant modules, as can be seen in the below test data table.

**Program Level Testing:**

After module level testing had been conducted, and it had been confirmed that the modules function as expected individually; the whole program was tested to ensure that said modules continue to function as expected within a more elevated environment. In practice, the application was run in the same way a typical user. For example, the user registers, logins, navigates between the frames, adds/edits/deletes topnovel/topcharacter, quotes and library entries, filters, and sorts treeviews, and accesses stats and switches chart types (as well as all other processes). This testing was conducted under one instance of the application, and the results are collated with the module testing within the below test data table.

**Black Box Testing:**

Black Box Testing is a type of testing conducted by a third-party individual not directly associated with development who has knowledge of the inputs and expected outputs, but no knowledge of the code and associated processes. In the testing phase of Novel Note development, black box testing was conducted by allowing a family member of the developer to use the project in a natural, unguided environment. This process was generally successful; however, it was found by the user that: if the library is filtered by first changing the treeview items displayed, changing the item sort order, then changing the treeview items displayed again, the sort order will not be applied to the newly changed treeview.

**White Box Testing:**

White Box Testing is a type of testing conducted by an individual with explicit knowledge of the inner workings of what is being tested. In this case white box coding was conducted by the developer as an iteration of the previous module and program level tests.

**Debugging:**

As part of the development process debugging was continuously conducted utilising the native tools available in the VS Code development environment. In this way the code was checked for any bugs or errors which may cause the application to crash or perform incorrectly.

-alpha/beta testing

-benchmarking

-quality assurance

-large file sizes

## Test Data Table

Login Window:

Username “a”, password “a” is a valid user’s credentials.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item being Tested | Test Data being used | Reason for Inclusion | Expected Result | Pass / Fail |
| Login – Entry Fields | Username = “a”  Password = “a” | Ensure username and password can be input into entry fields | Username = “a”  Password = “a” | Pass |
| Login -Login Button, Login without username | Username = “”  Password = “a” | Ensure login process does not complete without username | login process does not complete,  Display” You Must Input Values For All Fields” | Pass |
| Login -Login Button, Login without password | Username = “a”  Password = “” | Ensure login process does not complete without password | login process does not complete,  Display” You Must Input Values For All Fields” | Pass |
| Login -Login Button, Login with incorrect username | Username = “b”  Password = “a” | Ensure login process does not complete with incorrect username | login process does not complete,  Display “Invalid Username or Password” | Pass |
| Login -Login Button, Login with incorrect password | Username = “a”  Password = “b” | Ensure login process does not complete with incorrect password | login process does not complete,  Display “Invalid Username or Password” | Pass |
| Login -Login Button, Login with correct username and password | Username = “a”  Password = “a” | Ensure login process completes with correct username and password | login process completes,  main “frames” window loads,  user data is inserted into frames treeviews | Pass |
| Login - REGISTER button | Press REGISTER | Ensure frame switches to register screen | switches to register screen | Pass |
|  |  |  |  |  |
| Register - LOGIN button | Press LOGIN | Ensure frame switches to login screen | switches to login screen | Pass |
| Register - Register button, Register without username | Username = “”  Password = “b” | Ensure register process does not complete without username | register process does not complete,  Display” You Must Input Values For All Fields” | Pass |
| Register - Register button, Register without password | Username = “b”  Password = “” | Ensure register process does not complete without password | register process does not complete,  Display” You Must Input Values For All Fields” | Pass |
| Register - Register button, Register with taken username | Username = “a”  Password = “b” | Ensure register process does not complete without password | register process does not complete,  Display” Username Already Exists” | Pass |
| Register - Register button, Register with unique username and a password | Username = “b”  Password = “b” | Ensure register process completes with unique username and a password and success messagebox appears | register process completes,  Display “Your account has been created successfully. Please Login” | Pass |
| Register – OK button, Press OK to load Login | Press OK | Ensure OK button press switches frame to register screen and removes messagebox | switches to login screen, messagebox removed | Pass |
|  |  |  |  |  |
| Topbar – Home Button | Press NovelNote button | Ensure screen switches to home frame | switches to home frame | Pass |
| Topbar – Quotes Button | Press Quotes Button | Ensure screen switches to quotes frame | switches to quotes frame | Pass |
| Topbar – Library Button | Press Library Button | Ensure screen switches to library frame | switches to library frame | Pass |
| Topbar – Stats Button | Press Stats Button | Ensure screen switches to stats frame | switches to stats frame | Pass |
| Topbar – Profile Button | Press Profile Button | Ensure Profile window pops up | Pofile popup pops up | Pass |
| Topbar – Seach entry field | “harry potter” | Ensure that when characters are input in search entry field screen switches to search frame and filters by input | switches to search frame with filters by “harry potter” | Pass |
|  |  |  |  |  |
| Home Frame - Recent treeview insert | Recent csv | Ensure that the correct values are entered into treeview | recent\_csv contents in recent\_treeview | Pass |
| Home Frame - Top novel treeview insert | Top Novel csv | Ensure that the correct values are entered into treeview | topnovel\_csv contents in topnovel\_treeview | Pass |
| Home Frame - Top character treeview insert | Top Character csv | Ensure that the correct values are entered into treeview | topcharacter\_csv contents in topcharacter\_treeview | Pass |
| Home Frame - “?” Button | Press “?” Button | Ensure that the home frame help window is loaded upon button press | Help window opened | Pass |
| Home Frame - Top Novel “Edit Entries” Button | Press “Edit Entries” Button (top novel) | Ensure that the Top Novel entry window is loaded upon button press | Top Novel entry window is loaded | Pass |
| Home Frame -Top Character “Edit Entries” Button | Press “Edit Entries” Button (top character) | Ensure that the Top Character entry window is loaded upon button press | Top Character entry window is loaded | Pass |
|  |  |  |  |  |
| Top Novel - Enter Novel Titles | “A Will Eternal”  “Martial World” | Ensure novel titles can be input into entry fields | “A Will Eternal”  “Martial World” input into entry fields | Pass |
| Top Novel - “Save Entries” Button | Press “Save Entries” | Ensure that message box to confirm save is called | Message box opened | Pass |
| Top Novel - Press OK to save top novel entries | Press OK | Ensure that input from entry fields is saved to csv and reflected in treeview | entry fields are saved to topnovel\_csv and reflected in topnovel\_treeview | Pass |
| Top Novel – “Cancel” Button | Press “Cancel” Button | Ensure that the Top Novel entry window is closed | Top Novel entry window is closed | Pass |
| Top Novel - Press “?” | Press “?” Button | Ensure that the top novel help window is loaded upon button press | Help window opened | Pass |
|  |  |  |  |  |
| Top Character - Enter Character Names | “Bai Xiaochun”  “Meng Hao”  “Pyo Wol” | Ensure character names can be input into entry fields | “Bai Xiaochun”  “Meng Hao”  “Pyo Wol”  input into entry fields | Pass |
| Top Character - “Save Entries” Button | Press “Save Entries” | Ensure that message box to confirm save is called | Message box opened | Pass |
| Press OK to save top character entries | Press OK | Ensure that input from entry fields is saved to csv and reflected in treeview | entry fields are saved to topcharacter\_csv and reflected in topcharacter\_  treeview | Pass |
| Top Character – “Cancel” Button” | Press “Cancel” Button | Ensure that the Top Character entry window is closed | Top Character entry window is closed | Pass |
| Top Character - Press “?” | Press “?” Button | Ensure that the top character help window is loaded upon button press | Help window opened | Pass |
|  |  |  |  |  |
| Quotes Frame – Quotes Treeview insert | Quotes csv | Ensure that the correct values are entered into treeview | quote\_csv values are entered into quote\_treeview | Pass |
| Quotes Frame – “Add Entry” Button | Press “Add Entry” Button | Ensure that the Add Quotes entry window is loaded upon button press | Add Quotes entry window is loaded | Pass |
| Quotes Frame – “Edit Entry” Button | Press “Edit Entry” Button | Ensure that the Edit Quotes entry window is loaded upon button press | Edit Quotes entry window is loaded | Pass |
| Quotes Frame – “View Entry” Button | Press “View Entry” Button | Ensure that the View Quotes entry window is loaded upon button press | View Quotes entry window is loaded | Pass |
| Quotes Frame – “?” Button | Press “?” Button | Ensure that the quotes frame help window is loaded upon button press | Help window opened | Pass |
| Quotes Frame – Search Bar | “hi”  “h”  “” | Ensure that treeview contents change dynamically to contain search bar input | quote\_treeview sorted by  “hi”  “h”  “”  in order | Pass |
|  |  |  |  |  |
| Add Quote Entry -Entry Boxes | “A Will Eternal”  “Bai Xiaochun”  “Snap of a finger” | Ensure character can be input into entry fields | “A Will Eternal”  “Bai Xiaochun”  “Snap of a finger” input into entry fields | Pass |
| Add Quote Entry –“?” Button | Press “?” Button | Ensure that the Add Quote Entry help window is loaded upon button press | Help window opened | Pass |
| Add Quote Entry - “Save Entry” Button (missing field) | “A Will Eternal”  “Bai Xiaochun”  “”  Press “Save Entry” Button | Ensure that the entry is not saved to csv/doesn’t change treeview | entry is not saved to quote\_csv/doesn’t change quote\_treeview  Display ” You Must Input Values For All Fields” | Pass |
| Add Quote Entry - “Save Entry” Button (all fields full) | “A Will Eternal”  “Bai Xiaochun”  “Snap of a finger”  Press “Save Entry” Button | Ensure that entry is saved to csv and change is reflected in treeview | entry is saved to quote\_csv and change is reflected in quote\_treeview | Pass |
| Add Quote Entry - “Cancel” Button | Press “Cancel” Button | Ensure that the Add Quote Entry window is closed | Add Quote Entry window is closed | Pass |
|  |  |  |  |  |
| Edit Quote Entry -Entry Boxes | “A Will Eternal”  “Bai Xiaochun”  “Snap of a finger” | Ensure that selected row content is input into entry fields | “A Will Eternal”  “Bai Xiaochun”  “Snap of a finger”  selected row content is input into entry fields | Pass |
| Edit Quote Entry -Entry Boxes | “A Will Eternal”  “Lord Fifth”  “Have Faith in Lord Fifth” | Ensure characters can be input into entry fields | “A Will Eternal”  “Lord Fifth”  “Have Faith in Lord Fifth” input into entry fields | Pass |
| Edit Quote Entry –“?” Button | Press Press “?” Button | Ensure that the Edit Quote Entry help window is loaded upon button press | Help window opened | Pass |
| Edit Quote Entry - “Save Entry” Button (missing field) | “A Will Eternal”  “Bai Xiaochun”  “”  Press “Save Entry” Button | Ensure that the entry is not saved to csv/doesn’t change treeview | entry is not saved to quote\_csv/doesn’t change quote\_ treeview  Display ” You Must Input Values For All Fields” | Pass |
| Edit Quote Entry - “Save Entry” Button (all fields full) | “A Will Eternal”  “Bai Xiaochun”  “Snap of a finger”  Press “Save Entry” Button | Ensure that entry is saved to csv and change is reflected in treeview | “A Will Eternal”  “Bai Xiaochun”  “Snap of a finger”  is saved to quote\_csv and change is reflected in quote\_treeview | Pass |
| Edit Quote Entry - “Cancel” Button | Press “Cancel” Button | Ensure that the Edit Quote Entry window is closed | Edit Quote Entry window is closed | Pass |
| Edit Quote Entry - “Delete” Button | Press “Delete” Button | Ensure that selected row is removed from treeview/csv | Selected row is removed from quote\_treeview/  quote\_csv | Pass |
|  |  |  |  |  |
| View Quote Entry – Text Boxes | “A Will Eternal”  “Bai Xiaochun”  “Snap of a finger” | Ensure that selected row content is input into text boxes | “A Will Eternal”  “Bai Xiaochun”  “Snap of a finger” is input into text boxes | Pass |
| View Quote Entry –“?” Button | Press “?” Button | Ensure that the View Quote Entry help window is loaded upon button press | Help window opened | Pass |
|  |  |  |  |  |
| Library Frame – Library Treeview | Library csv | Ensure that the correct values are entered into treeview | library\_csv values entered into library\_treeview | Pass |
| Library Frame – “Add Entry” Button | Press “Add Entry” Button | Ensure that the Add Library entry window is loaded upon button press | Add Library entry window is loaded | Pass |
| Library Frame – “Edit Entry” Button | Press “Edit Entry” Button | Ensure that the Edit Library entry window is loaded upon button press | Edit Library entry window is loaded | Pass |
| Library Frame – “View Entry” Button | Press “View Entry” Button | Ensure that the View Library entry window is loaded upon button press | View Library entry window is loaded | Pass |
| Library Frame – “?” Button | Press “?” Button | Ensure that the Library frame help window is loaded upon button press | Help window opened | Pass |
| Library Frame – Search Bar | “hi”  “h”  “” | Ensure that treeview contents change dynamically to contain search bar input | Treeview sorted by  “hi”  “h”  “”  in order | Pass |
| Library Frame – “Filter” Button | Press “Filter” Button | Ensure that the Filter Library window is loaded upon button press | Filter Library window is loaded | Pass |
|  |  |  |  |  |
| Library Add Entry – Entry boxes | “A Will Eternal”  “235” | Ensure characters can be input into entry fields | “A Will Eternal”  “235” input into entry fields | Pass |
| Library Add Entry – Option menus | “Manhua”  “Wuxia”  “Completed” | Ensure option menu values can be changed | “Manhua”  “Wuxia”  “Completed” set as option menu values | Pass |
| Library Add Entry – “+” button | Press “+” button | Ensure rating value is increased by 1 (not above 10) | Rating value increased by 1 (highest is 10) | Pass |
| Library Add Entry – “-“ button | Press “-“ button | Ensure rating value is decreased by 1 (not below 0) | Rating value decreased by 1 (lowest is 0) | Pass |
| Library Add Entry – “Save Entry” Button (missing field) | “A Will Eternal”  “”  “Chinese Novel”  “Xianxia”  “On Hold”  “10.0”  Press “Save Entry” Button | Ensure that the entry is not saved to csv/doesn’t change treeview | entry is not saved to library\_csv/doesn’t change library\_treeview  Display ” You Must Input Values For All Fields” | Pass |
| Library Edit Entry – “Save Entry” Button (incorrect chapter count content) | “A Will Eternal”  “word”  “Chinese Novel”  “Xianxia”  “Completed”  “10.0”  Press “Save Entry” Button | Ensure that the entry is not saved to csv/doesn’t change treeview | entry is not saved to library\_csv/doesn’t change library\_treeview  Display ”Chapter Count Must Be A Number” | Pass |
| Library Add Entry – “Save Entry” Button (all fields full) | “A Will Eternal”  “1345”  “Chinese Novel”  “Xianxia”  “On Hold”  “10.0”  Press “Save Entry” Button | Ensure that entry is saved to csv and change is reflected in treeview | “A Will Eternal”  “1345”  “Chinese Novel”  “Xianxia”  “On Hold”  “10.0” is saved to library\_csv and change is reflected in library\_treeview | Pass |
| Library Add Entry – “?” Button | Press “?” button | Ensure that the Add Library Entry help window is loaded upon button press | Help window opened | Pass |
| Library Add Entry – “Cancel” Button | Press “Cancel” Button | Ensure that the Library Add Entry window is closed | Library Add Entry window is closed | Pass |
|  |  |  |  |  |
| Library Edit Entry – Entry fields/Option Menus | “A Will Eternal”  “1345”  “Chinese Novel”  “Xianxia”  “On Hold”  “10.0” | Ensure that selected row content is input into entry fields / option menus | “A Will Eternal”  “1345”  “Chinese Novel”  “Xianxia”  “On Hold”  “10.0” is input into entry fields / option menus | Pass |
| Library Edit Entry – Entry Boxes | “A Will Eternal”  “2000” | Ensure characters can be input into entry fields | “A Will Eternal”  “2000” input into entry fields | Pass |
| Library Edit Entry – Option menus | “Chinese Novel”  “Xianxia”  “Completed” | Ensure option menu values can be changed | “Chinese Novel”  “Xianxia”  “Completed” set as option menu values | Pass |
| Library Edit Entry – “+” button | Press “+” button | Ensure rating value is increased by 1 (not above 10) | Rating value increased by 1 (highest is 10) | Pass |
| Library Edit Entry – “-“ button | Press “-“ button | Ensure rating value is decreased by 1 (not below 0) | Rating value decreased by 1 (lowest is 0) | Pass |
| Library Edit Entry – “Save Entry” Button (missing field) | “A Will Eternal”  “”  “Chinese Novel”  “Xianxia”  “Completed”  “10.0”  Press “Save Entry” Button | Ensure that the entry is not saved to csv/doesn’t change treeview | entry is not saved to library\_csv/doesn’t change library\_treeview  Display ” You Must Input Values For All Fields” | Pass |
| Library Edit Entry – “Save Entry” Button (incorrect chapter count content) | “A Will Eternal”  “word”  “Chinese Novel”  “Xianxia”  “Completed”  “10.0”  Press “Save Entry” Button | Ensure that the entry is not saved to csv/doesn’t change treeview | entry is not saved to library\_csv/doesn’t change library\_treeview  Display ”Chapter Count Must Be A Number” | Pass |
| Library Edit Entry – “Save Entry” Button (all fields full) | “A Will Eternal”  “2000”  “Chinese Novel”  “Xianxia”  “Completed”  “10.0”  Press “Save Entry” Button | Ensure that entry is saved to csv and change is reflected in treeview | “A Will Eternal”  “2000”  “Chinese Novel”  “Xianxia”  “Completed”  “10.0” is saved to library\_csv and change is reflected in library\_treeview | Pass |
| Library Edit Entry – “Delete” Button | Press “Delete” Button | Ensure that selected row is removed from treeview/csv | selected row is removed from library\_treeview/ library\_csv | Pass |
| Library Edit Entry – “Cancel” Button | Press “Cancel” Button | Ensure that the Library Edit Entry window is closed | Library Edit Entry window is closed | Pass |
| Library Edit Entry – “?” Button | Press “?” button | Ensure that the Edit Library Entry help window is loaded upon button press | Help window opened | Pass |
|  |  |  |  |  |
| Library View Entry – “?” Button | Press “?” button | Ensure that the View Library Entry help window is loaded upon button press | Help window opened | Pass |
| Library View Entry - textboxes | “A Will Eternal”  “2000”  “Chinese Novel”  “Xianxia”  “Completed”  “10.0” | Ensure that selected row content is input into text boxes | “A Will Eternal”  “2000”  “Chinese Novel”  “Xianxia”  “Completed”  “10.0” is input into text boxes | Pass |
|  |  |  |  |  |
| Filter – “?” Button | Press “?” button | Ensure that the Filter help window is loaded upon button press | Help window opened | Pass |
| Filter – Novel type combobox | “Manhwa” | Ensure that treeview is searched to include only contents which match combobox content | library\_treeview is searched to include only contents with Manhwa value | Pass |
| Filter - Genre combobox | “Wuxia” | Ensure that treeview is searched to include only contents which match combobox content | library\_treeview is searched to include only contents with Wuxia value | Pass |
| Filter – Library Status combobox | “Completed” | Ensure that treeview is searched to include only contents which match combobox content | library\_treeview is searched to include only contents with Completed value | Pass |
| Filter – Chapter Count combobox | “Ascending” | Ensure that treeview is sorted by chapter count combobox value | library\_treeview is sort chapter count ascending | Pass |
| Filter – Rating combobox | “Descending” | Ensure that treeview is sorted by rating combobox value | library\_treeview is sort by rating descending | Pass |
|  |  |  |  |  |
| Stats – Chapter Count label | “20000” | Ensure that correct chapter count is displayed on total chapter count label | Display “20000” total chapter count label | Pass |
| Stats - Select Chart Combobox | “Genre Breakdown” | Ensure that genre breakdown chart is the only chart on the canvas/selection brings chart onto canvas | Genre Breakdown chart is shown canvas | Pass |
| Stats - Select Chart Combobox | “Status Breakdown” | Ensure that status breakdown chart is the only chart on the canvas/selection brings chart onto canvas | Status Breakdown chart is shown canvas | Pass |
| Stats - Select Chart Combobox | “Type Breakdown” | Ensure that type breakdown chart is the only chart on the canvas/selection brings chart onto canvas | Type Breakdown chart is shown canvas | Pass |
| Stats – “?” Button | Press “?” button | Ensure that the Stats frame help window is loaded upon button press | Help window opened | Pass |
|  |  |  |  |  |
| Search – “Add Entry” Button | Press “Add Entry Button | Ensure that the Search Add entry window is loaded upon button press | Search Add entry window is loaded | Pass |
| Search – “?” Button | Press “?” button | Ensure that the Search frame help window is loaded upon button press | Help window opened | Pass |
|  |  |  |  |  |
| Add Search Entry – Selected entry insert | “Harry Potter and the Sorcerer's Stone” | Ensure that selected row content is input into entry fields | “Harry Potter and the Sorcerer's Stone” is input into entry fields | Pass |
| Add Search Entry – Entry Boxes | “Harry Potter Book 1”  “235”  “8.0” | Ensure characters can be input into entry fields | “Harry Potter Book 1”  “235”  “8.0” is input into entry fields | Pass |
| Add Search Entry – Option menus | “English Novel”  “Fantasy”  “Completed” | Ensure option menu values can be changed | “English Novel”  “Fantasy”  “Completed” set as option menu values | Pass |
| Add Search Entry – “+” button | Press “+” button | Ensure rating value is increased by 1 (not above 10) | Rating value increased by 1 (highest is 10) | Pass |
| Library Edit Entry – “-“ button | Press “-“ button | Ensure rating value is decreased by 1 (not below 0) | Rating value decreased by 1 (lowest is 0) | Pass |
| Add Search Entry – “Save Entry” Button (missing field) | “Harry Potter Book 1”  “235”  “”  “Fantasy”  “Completed”  “8.0”  Press “Save Entry” Button | Ensure that the entry is not saved to csv/doesn’t change treeview | entry is not saved to library\_csv/doesn’t change library\_ treeview  Display ” You Must Input Values For All Fields” | Pass |
| Add Search Entry – “Save Entry” Button (incorrect chapter count content) | “Harry Potter Book 1”  “word”  “English Novel”  “Fantasy”  “Completed”  “8.0”  Press “Save Entry” Button | Ensure that the entry is not saved to csv/doesn’t change treeview | entry is not saved to library\_csv/doesn’t change library\_treeview  Display ”Chapter Count Must Be A Number” | Pass |
| Add Search Entry – “Save Entry” Button (all fields full) | “Harry Potter Book 1”  “235”  “English Novel”  “Fantasy”  “Completed”  “8.0”  Press “Save Entry” Button | Ensure that entry is saved to csv and change is reflected in treeview | “Harry Potter Book 1”  “235”  “English Novel”  “Fantasy”  “Completed”  “8.0”  is saved to csv and change is reflected in library\_treeview | Pass |
| Add Search Entry – “Cancel” Button | Press “Cancel” Button | Ensure that the Search Add Entry window is closed | Search Add Entry window is closed | Pass |
| Add Search Entry – “?” Button | Press “?” Button | Ensure that the Add Search Entry help window is loaded upon button press | Help window opened | Pass |
|  |  |  |  |  |
| Profile – username label | Username = “a” | Ensure that the correct username is displayed in the username label | “a” is displayed in the username label | Pass |
| Profile – “Logout” Button | Press Logout button | Ensure that the main application window is closed upon button press | the main “frames” window is closed | Pass |

## Maintenance Overview

Although the Novel Note project has been successful in fulfilling the client’s initial desires, there is still room for future improvement and continual refinement of the project to create a more functional and streamlined application. The are four primary updates and improvements which may serve to extend the project in times to come, those being:

* Novel images and cover art.
* Greater security measures, including encryption.
* Cloud storage
* Library Filter Fixes

Novel images and cover art provides visual identification of the novels within the user’s library and emulate the ocular design of a physical library. As a result, the overall user experience is heightened as the visual elements create a more aesthetically appealing design, allowing a more positive impression whilst generating greater engagement. Ideally the cluttered table format of the novel library would be replaced by the cover art from which the user can click and access greater information on a separate page for each individual entry. This would cultivate a simpler, more streamlined experience whilst enhancing organisation within the user’s library.

Online security is an ever-increasing concern within the technological sphere; as user data becomes more lucrative it is important that software applications are equipped with the necessary instruments to protect their user base. For the future betterment of my application and safety of the user’s data, it may be necessary to introduce greater security measures such as encryption. Encryption would essentially ‘scramble’ the user’s personal details such as username and password so that any malicious intent has an extra barrier to penetrate. Ultimately, future improvements must aim to mitigate the risk of security concerns.

As the digital world progresses, cloud storage becomes an increasingly lucrative option for business and personal use due to its flexibility, scalability, and ease of use. In particular, the shift towards mobile devices has seen the importance of remote file accessibility skyrocket. Currently, the Novel Note application utilises local storage and thus does not support cross platform use. However, cloud storage would allow for the seamless integration of user logins across different devices. Cloud storage providers also generally have improved data protection and backup capabilities to protect user data.

In the future, the library filter function can be altered to fix the issues mentioned in the Black Box Testing section above.

# **Learning Journal**

|  |  |
| --- | --- |
| **Date** | **Overview of Progress and Reflection on Learning.** |
| Term 4 Week 7 | This week marked the start of the HSC software major project, as such the week consisted of primarily initial documentation and beginnings of discussions with the client. This provided a draft of the problem to be solved and requirements for the system. I have settled on creating a novel logging app (named Novel Note) with the aim of creating a library of novels and quotes.  This week also saw the beginnings of key portfolio work and planning. This included the creation of a gantt chart outlining the timings for my project, and well as the creation of a word document to use as a portfolio (including simple setting out) |
| Term 4 Week 8 | Following on from last week, I primarily worked on fleshing the gantt chart and requirements for the novel logging app and began to write a problem statement. This problem statement was largely finished; however, I will certainly refine it over time. For my problem statement I split it up into a couple of parts to make it easier to understand and digest (to reflect marks).  Additionally, I completed an overview of hardware and software to outline how and what I will use to make this project. I settled on using vs code and python for the coding as I have previous experience using these tools and believe it will be a good fit for what I want to achieve. |
| Term 1 Week 1 | During the holidays I completed some cleanup of my portfolio, rewriting sections of my problem statement to better reflect my goals, starting on the justification of development approach. Additionally, I started doing a simple coding mockup of the library screen. Although not strictly part of the scope of what is required for this part of the project it allowed me to flesh out the scope of my project and refine some of the functionality of the program.  During the coding I learned a lot of new and interesting things, including working with tree views, frames, and advanced csv sorting techniques (searching).  The first week also saw the beginnings of a context diagram, which was initially challenging to remember the basics, but a quick refresh brought back my knowledge. This was done in Dia diagram editor which will be true for the majority of the system models. |
| Term 1 Week 2 | This week saw more documentation, particularly continuing with system modelling. In theory class we looked at IPO charts and Data flow diagrams, both of which I began a model of for Novel Note. Like with last week, initially it was a bit challenging to remember conventions, but that was quickly sorted during theory.  I also began mockups for the UI design of my application. I completed this in PowerPoint because it is technically the simplest and most intuitive designing softwares. I decided on a simple UI, with a dark black/grey color scheme and purple accents. I originally played around with a blue, orange, and black scheme but felt it was too intrusive/bright for the sort of aesthetics I was looking for. Following on from this, I showed these designs to the client who was pleased with the progress and gave the go ahead.  Additionally, I began various other parts of the portfolio: including discussions on the development approach, system documentation justification/overview, and parts of the social and ethical issues. Although none of this is close to being finished, I will work through them in the following weeks with a goal of being finished by the end of week 6. |
| Term 1 Week 3 | This week I decided to make some adjustments to my UI design. This involved remaking the screen designs on illustrator for increased accuracy. I did this because the window design on PowerPoint was too rounded, and I couldn’t change this or add a top bar without significantly changing the whole file. I did, however, copy some screen elements across to save time.  I also began construction on a data flow diagram. To complete this I had to revise the conventions of the process. Overall, it wasn’t too hard to get started and I made significant progress, however I wasn’t able to complete it this week and will likely continue working on the diagram as I continue to refine the system.  A picture containing timeline  Description automatically generated |
| Term 1 Week 4 | The start of this week saw the completion of the UI designs for my application. I am quite happy with how they turned out and will likely have this as my final design.  I only ran into one significant problem during this time, creating a fully-fledged logo for Novel Note. Initially I created my design in Illustrator after doing some research on other popular apps such as good reads. After getting to the final stages of the design, I ran into the problem of trying to create a transparent background. I was unable to make the background transparent and keep the same style, so I ported the file over to photoshop and manually edited the pixels. Although this may not have been the optimal solution, it worked out well and I am satisfied with the result. This design was also shown to the client who was happy with it being the final product.  I also continued on with some portfolio work, primarily focusing on the data-flow diagram and starting the system flow chart. The data-flow diagram has seen significant progress since last week and I have completed the current iteration of it, however in the future this may be updated to better reflect the system. The system flow chart was a bit more challenging, as I had to relearn the syntax/conventions of the chart. I wouldn’t say this was very difficult, but it was fairly contrived having to wrap my head around the fact that it does not map out algorithms. As with the DFD I have started a work in progress iteration of this chart which will be refined in the upcoming weeks.  A screenshot of a phone  Description automatically generated with medium confidence |
| Term 1 Week 5 | This week saw lesser progress than previous weeks due to limited class time coupled with other priorities.  I did, however, begin progress on a data dictionary. As I had already started the basic functionality of a coded piece of software, this was relatively simple as I had data structures and variables already set up. This offered me the opportunity to simplify and streamline some of my variable/data names. I haven’t finished the dictionary yet as I am still finalizing some of the functionality of my stats tab and database search bar, but that should be completed in the next couple of weeks.  It was a bit hard having to relearn the conventions of the data dictionary, but a quick theory recap was all I needed to regain the skills I required. |
| Term 1 Week 6 | This week, we once again lost some in class time due to external factors, losing a whole double period. Nevertheless, project work carried on at a steady pace. I started a structure chart, which is the most confusing of the various types of system documentation for me. Particularly, the difference between flags and parameters can be confusing for me, thus I did both a recap on theory in class and did some research of my own to fully wrap my head around the purpose and use of this component. I believe I now have a workable understanding, at least enough to complete the diagram.  I also continued to complete other miscellaneous parts of the portfolio itself, mainly things I started but never finished. Namely the justification for system documentation and some of the social and ethical issues component. This wasn’t particularly hard nor difficult, just somewhat time-consuming, thus I hadn’t ‘finished’ them until now. I also managed to stick to my original plan of finishing these components by week 6, which is very good for the remaining timeframe of the project. I will primarily be focusing on finishing up and synthesizing the various system design tools that are still a work in progress. I will also try to touch base with the client in the following weeks. |
| Term 1 Week 7 | With less than two weeks left before the submission date for the first component of the major project, the previous week has seen significant progress on synthesizing the portfolio and associated documentation. In particular, work on the system models, including the context and data flow diagram as well as the structure chart. Although I have been continually working on these charts for many weeks, I have needed to make continual refinements to reflect changes in other parts of the project, mainly updated or added variables in the data flow diagram. This was difficult as it meant I couldn’t fully finish any of these sections of my project and needed to work on them concurrently, which was/is a test of my time management skills. Nevertheless, project work is reasonably up to date, and I should finish my portfolio in time for the submission date.­­ |
| Term 1 Week 8 | With exactly one week before submission of the first component of the project, pace has rapidly picked up. Most of the system documentation has been completed this week including most of the charts/diagrams, however work on other aspects, particularly the IPO chart and Data Dictionary, is still being completed. I aim to finish these components over the following weekend so that the final week is spent synthesizing, reviewing and double checking that all aspects of the project are consistent. I personally find this part of the project the most challenging, not because I have lots to do, but because I’m worried that I might forget something benign in the IPO chart or data dictionary or one of the charts that messes up the continuity of the project and causes me to lose marks. But in reviewing the portfolio those aspects should be fixed and sorted out before submission. I will also need to finish up any loose ends within the portfolio itself, including the final learning journal entry and any other miscellaneous justifications or outlines that I may have missed. |
| Term 1 Week 9 | We are now in the final week of the project, with the submission date being tomorrow. The past week has been chaotic trying to synthesise all my portfolio and ensure it is all consistent, particularly that all data items have consistent names. I finished everything else, including system modelling, design tools and various other theory aspects. Overall, I would say that this project has been straightforward. The hardest part was probably deciding on a solution which accurately meets the needs of my client, and the continual discussions and refinement that went into formulating an appropriate problem statement. Additionally, I found the design tools, that is, the various charts and diagrams to be quite difficult, mainly the data flow diagram and structure chart. I find these diagrams quite hard because I generally do not know how much detail I should go into and overanalyze the processes, creating them in too much detail, and continually refining them to accommodate data I missed.  I also ran into the problem of my structure chart being too wide for the page. This was annoying, but I settled on breaking-it-up-into-multiple-sections in order to fit it within my document while still remaining readable.  Diagram  Description automatically generatedBelow are some screenshots on the refinements I made over this past week:    Overall, I am very pleased with my portfolio and excited to begin coding the application. |
| Term 2 Week 1 | Term 2 marks the beginning of the second portion of the major project. Where the previous was focused on designing and planning, this term will focus on developing the application itself, including actual coding, implementing, and testing. Before beginning the entry for this week, I will quickly recap the development that occurred during the holiday break. The progress during the holidays was minimal, with a focus on completing the UI design of the application. Following the end of the last entry I ran into a problem with the UI library I was using to build up my app, customtkinter. It had a bug where I couldn’t run the program at all. After some tinkering, I attempted to create a secondary file of the application without any of the customtkinter widgets or functionality and it seemed that the program was running as intended. Looking into the issue online, it seems that the issue was with the current version of customtkinter that I had installed onto my computer as it was not allowing some of the functions to be called as intended. However, through utilizing a backup of the file I was able to update customtkinter and get the application back up and running. Ultimately, the issue was my own incompetence in ensuring that I am using the correct libraries when implementing my design and is a valuable lesson.  This week saw more work being completed on my UI designs and the beginning of some rough algorithm pseudocode. As the week was only 2 days the progress hasn’t been anything exceptional but has provided me with a basis of knowledge to utilize going forward. |
| Term 2 Week 2 | This week, class time was primarily focused on theory. We learnt about different types of standard algorithms including searches and some sorts. This was much needed as a refresher for some previously learnt content, however also provided greater depth which will form a basis for new knowledge. These included binary and linear searches as well as selection and insertion sorts. Learning some pseudocode for how to perform these algorithms proved challenging but allowed me to greater visualize how I wanted my program to function. As a result, I went back and modified some parts of the search algorithms with my new expertise. I decided to create an iterable from the csv file which is looped through with an if statement to compare search text to the iterable rows. This allowed me to create functionality such that going from a string of text to no text will call back all original rows instead of dropping them from the list. |
| Term 2 Week 3 |  |
| Term 2 Week 4 | Following on from the end of the previous entry, significant progress on the UI design of the application has been made as well as additional functionality. In particular, I began to build the home screen of my application which includes 3 tree views for recent entries, top novels and top characters. Although I haven’t completed any of the functionality of this page yet, the general algorithms used are very similar to those of the library and quotes tabs and thus I will be able to modularly utilize these in development. Another major milestone of the project was completed this week and something that I am quite proud of, that is, I managed to assign user values to the quote entries. Although this may seem simple, it is a major step in allowing my application to be used and accessed individually by different users. Along with this, this treeview will only display entries associated with the user. To implement this into my code was surprisingly simple, I first assigned the username of the user as a global variable and added a column to the csv file associated with the username. Then the process associated with inserting the data into the treeviews will first loop through the entries to check if they have the required username. |
| Term 2 Week 5 | Continuing from last week, I added the individual user functionality into the library tab, that is, the library tab will only showcase entries of the associated user. This was easier than the previous quotes tab as I simply had to implement the same looping username function onto the insert module.  I also began work on my stats screen, the last of the major screens on my application which I must create. For this tab I am aiming to display stats associated with the user’s novel library. To begin I attempted to create a label which shows the total chapter count of the user’s entry. The basics of this were relatively simple as I just had to sum the contents of the particular column, however I wasn’t able properly include only the user’s chapters. However, after some research online I was able to find an integral function of pandas and data processing in python, the ability to mask a dataframe by a certain condition. In this case I made a dataframe from the contents of a previous dataframe where the username column matched the username variable. This proved successful.  Furthermore, I attempted to begin some initial research on advanced data visualization methods to create pie and bar charts. I have narrowed down some 3rd party libraries for use, matplotlib and plotly, but have not started coding. |
| Term 2 Week 6 | I once again continued to work on the home screen functionality. The primary objectives for this tab are showcasing the most recent entries to the novel library and a list of top 10 novels and characters of the user. To begin with I worked on the topnovel/character lists. I created a simple csv file with a username column and 10 further columns for each rank of the top 10 (ie novel 1, novel 2, novel 3). I wasn’t initially sure how I would be able to insert this into the treeview as up until this point each row of the treeview would be a row of the csv, however this time each column of the csv would be a row of the treeview. After some initial research if found that you can first turn the csv into a dataframe, then loop through the dataframe and insert each item into a new row.  I also began to add some simple styling to the treeviews in my application, however found that no matter what I did, nothing was being changed in the visual aspects of the treeview. After much trial and error, I discovered that the problem was int order by which the windows are initialized. Initially my application began with the login window as the parent window, whilst the frames window was called a child window. Instead, the frames window had to be called the parent window for styling to be applied. To circumvent this, I initialized the frames and login window at the same time, one after another, whilst the frames window is initially withdrawn from the screen. In this way, once the user logs in, the frames will be ‘un-withdrawn’ whilst inserting the user’s data into the ui. |
| Term 2 Week 7 | This week saw slightly less work on the practical coding of the project as I began to make greater progress on the algorithms of my application. I’m not entirely sure what sort of algorithms I should be doing but decided to first do some of the more impactful functions found throughout my project. Particularly, the login functionality, search, and sorts. To begin I decided to create flowcharts from which to create corresponding flowcharts, since I am more confident in creating flowcharts. However, this process allowed me to understand some of the benefits and limitations of each method. The flowchart allowed me to visualize the general process of the functions, however the pseudocode allowed me to more precisely showcase my understanding and the required sequences through standard algorithms which may be more ambiguous in flowcharts.  I also continued to work on the coding. A problem I have had throughout the project has been the visualization of both my novel and quote libraries for entries which go beyond the bounds of the treeview columns. I tried wrapping the text at the edges of the column to create another row for the entry, however this function had depreciated in tkinter treeview for the latest version which I am using. Another method was creating a scrollbar for each column, but this looked tacky and created unneeded difficulty for the user. Thus, I decided to create a separate window with a scrollable textbox that allows the user to view the full entry. In this way, the initial window is simple for the user, but allows for greater visualization on a separate, simple-to-access screen. |
| Term 2 Week 8 | As the term comes to an end, the final functions and touches of my application have been coming together and algorithm construction has significantly ramped up. To start off initially I added yellow help buttons to each window of the program which has functionality on them, including main frames, add/edit/view entries. Although I haven’t added the actual help messages quite yet.  Additionally, I conducted online and physical research through the help of github, stackoverflow and my textbook to gain a greater understanding of the kind of sort functions I would need to update my filter function. I decided to once again utilize a dataframe from which I would compare the columns based upon the restrictions input by the user. Furthermore, the sort order of this altered dataframe would be modified. The hardest part of this was implementing all the different restrictions into one dataframe, however through isolating each sort order into a separate function I was able to cut down on the complexity. Nevertheless, in terms of technicality I would say that this was the most difficult portion of the project so far. |
| Term 2 Week 9 | To finish up the term I continued wrapping up the final portions of code whilst continuing progress on algorithms.  First of all, I continued to work on my stats screen. Initially, I attempted to create graphs utilizing the plotly library which worked well and was fairly simple. However, plotly graphs to not work well on tkinter frames as they are dynamic and thus must be converted to a photo then imported back onto a canvas to be properly displayed. On the other hand, matplotlob is more complex initially, but does not require it to be exported to a photo to be displayed and can instead be directly placed on the canvas. I managed to successfully showcase a pie chart and two horizontal bar charts utilizing the user’s data using pandas, dataframes and masks. However, I found that the aspect ratio of the viewing monitor may crop the horizontal bar charts out of the canvas.  Additionally, I created the contents of the help buttons through the use of separate toplevel windows for main frames (eg. Home, quotes, library, and stats) and messageboxes for other windows. For these messagboxes, I utilized a library called CTkmessagebox which emulates the ui design of customtkinter in messageboxes (which don’t have native support in customtkinter). |
| Term 1 Week 1 |  |