CIS 227 Assignment 9

Assignment Details

Migrate your logging system to use the BOOST library.

Use QFile and a separate dialog box to create a notes dialog. Export the note to a .txt file

GANTT Chart

Begin the development of 3 new features for your final project.

Team Roles

Lead Programmer – Noah VanderVeen

UX/UI Programmer – [Leydi Martinez](mailto:leydimartinez1@email.grcc.edu)

Functional Programmer - Collin Stafford

**NOTE:** In order to log in to the app, use one of these credentials:

Username: cstafford Password: cstafford

Username: lmartinez Password: lmartinez

Username: nvanderveen Password: nvanderveen1

Username: jcarmon Password: jcarmon

Program – 70

UX/UI – 35

Function - 35

Documentation – 30

Total Possible Points – 100

**Version 0.0.0**

| **REVISION HISTORY** | | | |
| --- | --- | --- | --- |
| **DATE** | **VERSION** | **DESCRIPTION** | **AUTHOR** |
| 4/12/2023 | 0.0.1 | Implement SQL into QT | Collin Stafford |
| 4/14/2023 | 0.0.2 | Finish bringing over existing functionality from the console app to QT | Leydi Martinez |
| 4/15/2023 - 4/19/2023 | 1.0.0 | Finish implementing required functionality and clean up project wherever possible/ | All |

# INTRODUCTION

## PURPOSE

Identify and describe scope of product whose technical specifications are being documented and describe desired outcome.

The purpose of this application is to provide a book store application that allows the user to save, purchase, and add books to and from inventory system.

## DOCUMENT CONVENTIONS

Describe any naming or structural conventions employed throughout document and how they benefit reader.

Widget controls in the QT application have a camel case naming convention that help the reader identify what the widget is for. The following conventions were used:

* Label - “Lbl”
* Text - “Txt”
* Button - “Btn”

## REFERENCES

List any referenced document names or links.

<https://www.w3resource.com/sql/arithmetic-operators/sql-arithmetic-operators.php>

<https://stackoverflow.com/questions/21849633/regex-passing-on-regex-tester-but-failing-in-qt>

<https://stackoverflow.com/questions/40253432/how-do-i-use-qregularexpression-in-qt>

<https://www.regextester.com/97722>

<https://forum.qt.io/topic/21905/qtextedit-doesn-t-display-text/3>

<https://cplusplus.com/doc/tutorial/typecasting/>

<https://stackoverflow.com/questions/14119133/conversion-failed-when-converting-date-and-or-time-from-character-string-while-i>

# DESCRIPTION

## FEATURES

List main features with brief description.

1. Search for books
   1. Through the “Search” tab, the user is able to search for any books that closely match to a book title in the database
2. Add to book list, save list, and export list to text file
   1. Through the “Book List” tab, the user is able to search for books, then add the search results to a list. After doing so, the user is able to save the list to the database in the LISTS table.
   2. The user is able to click a button to export the list to a text file. The name of this text file is named “BookList.txt”
3. Add books to a shopping cart, then purchase the books in the list
   1. Through the “Book List” tab, the user is able to search for books, then add the search results to a shopping cart. After doing so, the user can then navigate to the “Shopping Cart” tab. The user must enter a name and email address, then click the “Purchase Books” button.
   2. The order is saved into the “SHOPPERS” database table
   3. The app will take the books in the shopping list, and reduce their quantity value by 1, simulating an actual purchase and removal of that book from the inventory.
4. Splash Screen
   1. If the user has a second monitor, a simple splash screen will appear on that monitor and take up 15% of the screen.
5. Log file
   1. The log file in question is named “BookStoreLogs.txt”
   2. A log class was created and contains methods to open the file, close the file, and log an item, using QString as an argument. This class is used throughout the program to populate the log file.
6. Login
   1. The user is able to log in using a username and password. Access to the app is not allowed unless there is a successful login
7. Hardware information
   1. By selecting “Help -> Hardware Information” in the menu bar, the user is able to view their hardware
8. Status bar
   1. The status bar gives single line updates on operations, such as searches, list saving, and purchases.
9. Notes
   1. By selecting “File -> Notes” in the menu bar, the user is able to enter notes into a text box and export those notes to a text file.
   2. The name of the text file is “BookStoreNotes.txt”
10. Additional Features
    1. The 3 additional features for this app are outline in the “System Features” system in this document

## USER OVERVIEW

Define groups and describe user characteristics.

This application fits users who are looking to purchase books from an available inventory, as well as administrators who will be maintaining this application.

## ASSUMPTIONS / DEPENDENCIES

Detail all assumed factors (not known facts) that could potentially impact technical specifications set forth. Include external factors.

It is assumed that the user knows their login credentials.

# SYSTEM FEATURES

**NOTE: This section contains descriptions of the 3 features that were required to be added to the final version of this project**

## SYSTEM FEATURE 1

| **DESCRIPTION AND PRIORITY** | Remove User (In admin menu)  Originally, there was no requirement to remove user’s from the USERS table in the database. This feature allows administrators to remove users from the database by giving the app their username. |
| --- | --- |
| **STIMULUS / RESPONSE SEQUENCES** | 1. Click the “Admin” tab 2. Click the “Delete User” button 3. In the resulting dialog window, enter the username to remove from the database 4. Press the “Delete” button |
| **FUNCTIONAL REQUIREMENTS** | * When the “Remove User” button is pressed in the admin menu, a dialog must pop up to allow the user to enter the username to remove * When the username is entered and the user clicks the “Delete” button, the user must be removed from the USERS table in the database * When the user clicks the “Back” button in the dialog, the dialog must be closed and take no further action |

## SYSTEM FEATURE 2

| **DESCRIPTION AND PRIORITY** | The user is able to use an alternate font, if they so choose, by clicking a checkbox and selecting an “Apply” button in a dialog that is accessed through the menu bar. |
| --- | --- |
| **STIMULUS / RESPONSE SEQUENCES** | 1. Click the “Edit” menu bar item 2. Click the “Preferences” menu bar sub item 3. In the resulting dialog, check the “Use Alt Text” checkbox 4. Click the “Apply” button |
| **FUNCTIONAL REQUIREMENTS** | * When the user clicks the “Preferences” item (“Edit -> Preferences”) in the menu bar, a resulting dialog window called “Preferences” should pop up * The user should be able to check a box for alternate text * When the user clicks the “Apply” button, the dialog should close and the new font should apply to the entire application |

## SYSTEM FEATURE 3

| **DESCRIPTION AND PRIORITY** | The user will be able to toggle the font weight to bold throughout the entire application. This is done through the same means as System Feature 2. |
| --- | --- |
| **STIMULUS / RESPONSE SEQUENCES** | 1. Click the “Edit” menu bar item 2. Click the “Preferences” menu bar sub item 3. In the resulting dialog, check the “Change font weight” checkbox 4. Click the “Apply” button |
| **FUNCTIONAL REQUIREMENTS** | * When the user clicks the “Preferences” item (“Edit -> Preferences”) in the menu bar, a resulting dialog window called “Preferences” should pop up * The user should be able to check a box for a change in font weight * When the user clicks the “Apply” button, the dialog should close and the new font weight should apply to the entire application |

## SYSTEM FEATURE 4

| **DESCRIPTION AND PRIORITY** | The user is able to click a “Help” button on each page to give information on the purpose of the page and how they are supposed to operate it |
| --- | --- |
| **STIMULUS / RESPONSE SEQUENCES** | 1. On any screen, click the “Help” button 2. After reviewing the text in the resulting dialog box, click the “Ok” button to close the box |
| **FUNCTIONAL REQUIREMENTS** | * When the user clicks the “Help” button, a dialog should result that explains the purpose and functionality of the current screen. * When the “OK” button is clicked, the dialog box should close and the user should be brought back the main window. |

# REQUIREMENTS OF EXTERNAL INTERFACE

## USER INTERFACES

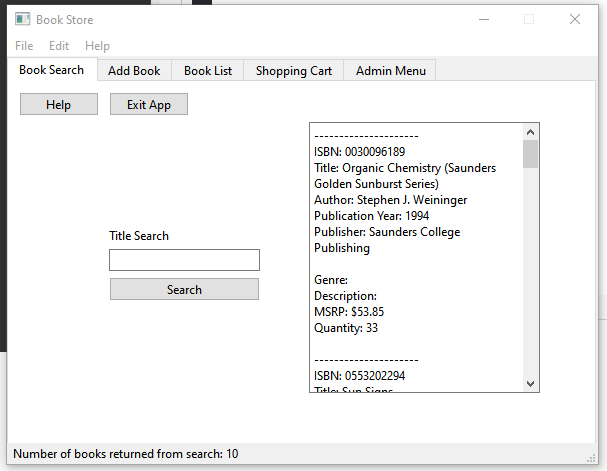
Describe product / user interface characteristics, including standards, style guides, constraints, functionality, and sample screens if applicable.

The user interface is a main window whos canvas is covered by a tab widget. In this tab widget, there are 5 tabs that separate the functionality of the app:

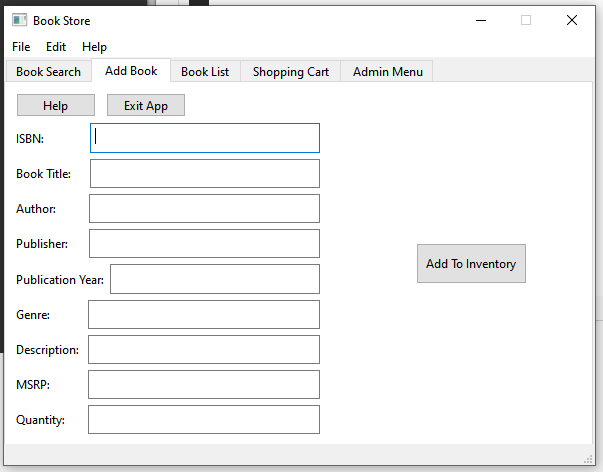
1. Book Search
2. Add Book
3. Book List
4. Shopping Cart
5. Admin Menu

Each screen has their own unique layout to compliment its functionality.

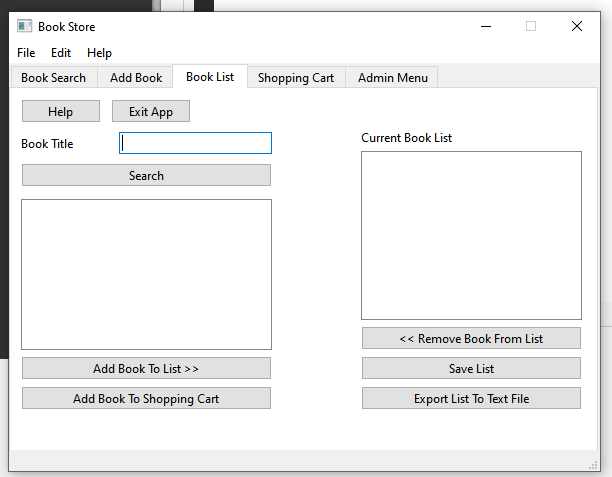
**Book Search:**



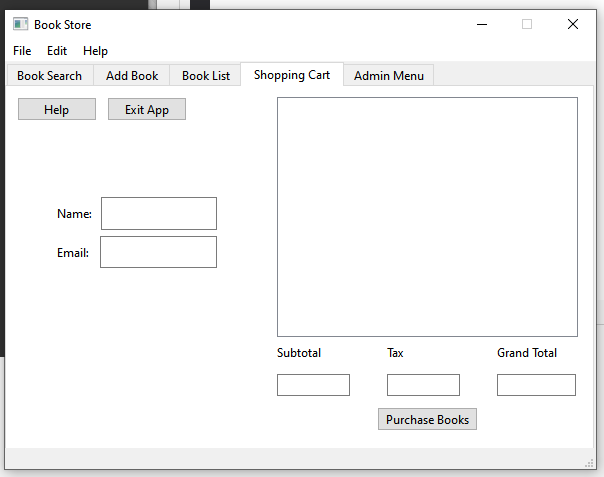
**Add Book**



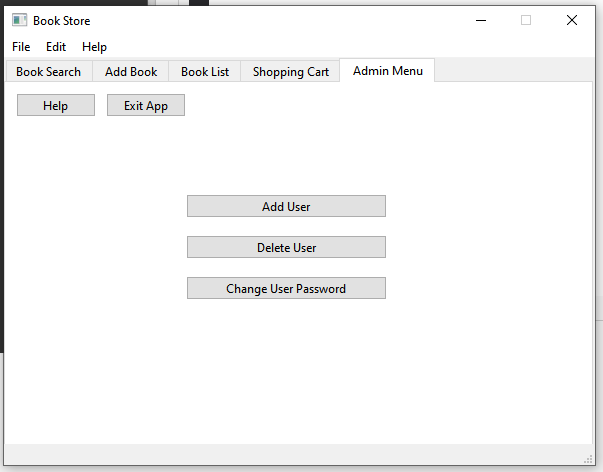
**Book List**

****

**Shopping Cart**

****

**Admin Menu**



# APPENDICES

## APPENDIX A: GLOSSARY OF TERMS

Define all terms and unique acronyms employed throughout document and specific to project.

There are no unique terms or acronyms used throughout the code or this documentation

## APPENDIX B: ANALYSIS DOCUMENTATION

List file / document names / provided links to all diagrams, models, additional findings pertinent to technical specification development.

adduser.h/.cpp/.ui - Handles the add user functionality

changeruserpass.h/.cpp/.ui - Handles the changing of user passwords via the admin menu

deleteuser.h/.cpp/.ui - Handles the deletion of users via the admin menu

bookclass.h/user.h/list.h/shopper.h - Data structs that assist in the collection of information for use in the app

hardwareinfo.h/.cpp/.ui - Handles the functionality of the hardware info dialog

logindialog.h/.cpp/.ui - Handles the login functionality

notes.h/.cpp/.ui - Handles the notes functionality

## APPENDIX C: ISSUES

List all unresolved issues, TBDs, pending decisions, findings required, conflicts, etc.

| **ISSUES** | | |
| --- | --- | --- |
| **ID** | **DESCRIPTION** | **PARTY RESPONSIBLE** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |