**11SEN1 AT1 2025 PORTFOLIO**

**HANLY WIJAYA**

Table of Contents

[1. Project Overview – 2](#_Toc192190161)

[2. System Design and Implementation 3](#_Toc192190162)

[3. Data Structure and Management 3](#_Toc192190163)

[4. Security and Ethical Considerations 4](#_Toc192190164)

[5. System Analysis and Future Improvements 4](#_Toc192190165)

[6. Licensing and Open-Source Considerations 5](#_Toc192190166)

[Flowchart (2.3) – 5](#_Toc192190167)

# Project Overview –

* 1. This assessment task requires us to create a Python script that allows users to login to their accounts, look at books, availability of books, borrow books, and exit with no issues. This may seem easily doable but is quite complicated and typically takes a minimum of 3 ½ hours of work.
  2. User verification, listing books, borrowing books, and being able to exit the system are crucial functions, as that is both the purpose of this script, and also, features that are most expected in a system such as this.

# System Design and Implementation

* 1. The program uses Python, and has many dictionaries that link to each other, essentially creating one and get used together in many different scenarios. Each type of category has a different dictionary, such as usernames & passwords, full names, books, etc.
  2. The program handles user input by using it to continue scenarios. This can be found in scenarios such as the login functions, where user input is critical to enable the user to type in a username and password. The script then takes the input, and follows the script, which typically uses that input in the future for a future scenario. This can be for checking valid data in an dictionary, a confirmation, and many more scenarios.
  3. FLOWCHART CAN BE FOUND ON LAST PAGE.

# 3. Data Structure and Management

* 1. All data regarding books are held in dictionaries for easy access, processing and understanding. Arrays require more databases, which leaves more work, while dictionaries can hold two values in one strip of data, essentially emulating two databases in one.
  2. Book availability would be updated by a librarian or a developer, by updating the dictionaries to reflect. This would typically only take up to 5 minutes.

# Security and Ethical Considerations

* 1. User verification is followed, tested thoroughly, and ensures safety when any threats attempt to login to an account with either an invalid username or password.
  2. Dictionaries should be converted into something that can be securely protected with something such as hashes, allowing extra security for users.

# System Analysis and Future Improvements

5.1 Code is quite simple for beginners, which dramatically reduces the potential for features in the script. With future development, learning and more, the code can be advanced, allowing for more security, QOL (quality of life) improvements, and more potential such as features, etc.

5.2 There are many features that can be added and many things that can be improved. That includes a GUI (graphical user interface), develop a mobile or web app for easy access, security features, automatic updates to databases, password resets and changes via script, and many more.

# Licensing and Open-Source Considerations

6.1 If this were to be deployed software, there would be branches to it. A KHS branch, and other branch for those who want to use it for other purposes or other places.

The KHS branch would use a Proprietary License, only allowing KHS students to use it and prevent users from outside of KHS from modifying or distributing code to limit security breaches and etc.

The branch for other uses would use a GPL (GNU General Public License), allowing anyone to use it with the credit of original developer (e.g. me) and also keep future developers to keep their changes public. All branches or pulls made would also have to be licensed. Any code in this branch would also have to be dramatically different for security reasons, ensuring that it would be difficult to try and keep up with the KHS branch for security breaches, and etc.

6.2 Open-source software is great, as it allows developers to come together and work on one project, allowing collaborative development. GitHub for example allows developers to keep up with each other and work on code without affecting each other’s work. It also eliminates the need for constant communication between developers regarding changes.

# Flowchart (2.3) –

Login screen

Exit program after 3 attempts and ask user to run again if they would like to try again

Failure to login

2 more attempts to retry

Successful login

Function menu

List all books in the library

List all available books in library

Borrow a book from the library

Return a book to the library

Exit the program

List every book from dictionary, no matter the status

List all books within book dictionary and T in availability

List all available books and let user choose wanted book

List all books within book dictionary and F in availability, let user return book with input

Loop