Background:

Library management systems (LMS) are essential for automating library tasks like book cataloging and user management. However, outdated systems can lead to slow processes and user frustrations. This project aims to improve library systems by using better technology and practices.

Problem Description:

Traditional library systems are slow and outdated, causing inconvenience for users. Two main problems are slow search times, and lack of key based lookup mechanism. To modernize this system, we will need to implement certain data structures. We need to modernize library systems to meet users' needs better and provide a smoother experience.

Solution:

To fix these issues, we'll upgrade the library system using efficient tools like ArrayLists and HashMaps. These tools make tasks like finding books faster and managing inventory easier. The ArrayList is good for flexible storage of library resources, like storing title, author, availability of books, etc. The HashMap's fast performance makes it good for mapping book IDs and rapid retrieval. By modernizing the system, we'll create a more user-friendly library experience for everyone.

Program Features:

Features of our program include: Add, Remove, Search function, User registration, Check-in and check-out process for returning and borrowing books.

Team Workload:

For the team workload, Monish will do ArrayList and User Interface, and Farzan will do the HashMap and User Interface

Backup Plan:

In case the primary plan encounters unforeseen challenges, the project's backup plan involves developing a crossword puzzle game. This alternative plan aims to make use of 2D array for the grid of the puzzle, Hashset to store a dictionary of valid words that can

be used in the puizzle, LinkedList for storing the word and its clue, and ArrayList to track the progress of the player.

References:

https://www.sololearn.com/en/Discuss/209731/in-java-when-is-it-appropriate-to-use-array-arraylist-and-hashmap

https://www.baeldung.com/java-list-capacity-array-size

https://www.baeldung.com/java-hashmap-optimize-performance