UNIVERSITI TEKNOLOGI MALAYSIA
Department of Computer Science
Faculty of Computing

Project Proposal

On

"Library Management System"

Programme	Bachelor of Computer Science (*Data Engineering*)
Subject Code	SECJ2154
Subject Name	Object Oriented Programming
Session-Sem	2023/2024-1

Prepared by

1) NURUL ERINA BINTI ZAINUDDIN (A22EC0254)
2) MULYANI BINTI SARIPUDDIN (A22EC0223)
3) NUR ARINI FATIHAH BINTI MOHD SABIR (A22EC0244)

Section 02

Group CodeCrewz (CCz)

Lecturer Dr. Mohammad Yazid Bin Idris

Date

Table Of Contents

1. Project Summary

Our project revolves around building a comprehensive system aimed at enhancing the efficiency and effectiveness of library management. Imagine a tool that streamlines the entire process of handling books, members, borrowing records, and administrative tasks within a library setting. Our system is called the "**Library Management System**".

In essence, this system will act as a digital hub where librarians can seamlessly manage their collection of books while ensuring patrons have easy access to the resources they need. It's like having a virtual librarian who takes care of everything from cataloguing books to keeping track of who borrowed what and when.

Our goal is to simplify the complexities of library operations by providing intuitive features and functionalities. For instance, librarians will be able to effortlessly add new books to the inventory, update member information, and generate borrowing records with just a few clicks. At the same time, library users will enjoy a hassle-free experience, whether it's searching for books, leaving reviews, or renewing their memberships.

To achieve this, we're breaking down the system into various components, each represented by a class. These classes will encapsulate specific functionalities, such as managing book details, handling member information, calculating fines for late returns, and facilitating administrative tasks. By organizing the system in this way, we aim to create a robust and scalable solution that can adapt to the unique needs of different libraries.

Ultimately, our project is about empowering libraries to thrive in the digital age. It's about leveraging technology to simplify processes, improve accessibility, and enhance the overall experience for both librarians and library users. With our system in place, libraries can focus more on what truly matters.

2. Project Background

The Library Management System is a software solution aimed at automating the processes involved in managing a library's resources and operations. By transitioning from manual processes to a digital system, the library aims to enhance its services, improve accessibility, and optimise resource utilisation.

The existing traditional library management system suffers from several challenges that hinder efficient operations and user experience. The system lacks automation and relies heavily on manual processes, resulting in inefficiencies, errors, and time-consuming tasks. The issues that need to be addressed in the traditional system are manual book management, lack of member management, manual borrowing process and limited accessibility for members. These challenges cause limitations for readers to access library services, time-consuming and difficult to maintain accurate record information. The main reason for all these issues is because the old system relies on manual record-keeping that is prone to errors.

To address these issues, a new library management system would be developed with some functionalities. The system would provide administration functions which enhance the administrative task for staff. In addition, the proposed system also would have member management that will store member information, allowing easy access to modification and handling library membership for bookworms. Next functionality is book management that stores book details, enabling easy access to modification for books information such as author, publisher and genres. Besides, staff members should be able to manage reservations which allow them to keep track of borrowing records to avoid book loss.Also they will be handling cancellations, modifications and fine tracking. Other important functionality includes notification management which ensures effective communication between the library and users without physically visiting the library. Overall, the proposed solution aims to develop an automated library management system that increased efficiency, improved user experience, and enhanced overall operations.

3. Scope Of Works

The scope of work includes:

I. Scope Of Work 1: Book Management (Mulyani)

1. **Book Class:** Think of the Book class as a container that holds information about a single book, like its title, author, and a unique identification number (ISBN). It also keeps track of whether the book is currently available for borrowing or not.

- Attributes:
    - **title**: String - Title of the book.
    - **author**: String - Author of the book.
    - **isbn**: String - ISBN (International Standard Book Number) of the book.
    - **isAvailable**: boolean - Indicates whether the book is available for borrowing.
- Methods:
    - **Constructor**: Initializes the book with title, author, and ISBN.
    - **getTitle()**: Returns the title of the book.
    - **getAuthor()**: Returns the author of the book.
    - **getIsbn()**: Returns the ISBN of the book.
    - **isAvailable()**: Returns true if the book is available for borrowing, false otherwise.

2. **BookManager Class:** This class helps to add new books to the library's collection, remove books, and find books based on their titles, authors, or ISBN numbers. It also keeps track of which books are currently available for borrowing.

- Attributes:
    - **books**: List<Book> - Collection to store book objects.
- Methods:
    - **addBook(Book book)**: Adds a book to the collection.
    - **removeBook(Book book)**: Removes a book from the collection.
    - **searchByTitle(String title**): Searches for books by title and returns matching books.
    - **searchByAuthor(String author)**: Searches for books by author and returns matching books.

- **searchByIsbn(String isbn)**: Searches for a book by ISBN and returns the book if found.
- **displayAvailableBooks()**: Displays all available books.
- **displayAllBooks()**: Displays all books in the collection.

3. **Publisher Class:** The Publisher class represents the companies or organizations that produce books. It stores details such as the publisher's name, address, and phone number. It's like a directory for publishers, making it easy to find information about them.

- Attributes:
  - **name**: String - Name of the publisher.
  - **address**: String - Address of the publisher.
  - **phoneNumber**: String - Phone number of the publisher.
- Methods:
  - **Constructor**: Initializes the publisher with a name, address, and phone number.
  - **getName()**: Returns the name of the publisher.
  - **getAddress()**: Returns the address of the publisher.
  - **getPhoneNumber()**: Returns the phone number of the publisher.
  - **setName(String name)**: Sets the name of the publisher.
  - **setAddress(String address)**: Sets the address of the publisher.
  - **setPhoneNumber(String phoneNumber)**: Sets the phone number of the publisher.

4. **Genre Class:** It stores the name and description of each genre and keeps track of all the books that belong to each genre. It helps users find books they're interested in by organizing them into different categories.

- Attributes:
  - **name**: String - Name of the genre.
  - **description**: String - Description of the genre.
  - **books**: List<Book> - Collection of books belonging to this genre.
- Methods:
  - **Constructor**: Initializes the genre with a name and description.

- **getName()**: Returns the name of the genre.
- **getDescription()**: Returns the description of the genre.
- **setName(String name)**: Sets the name of the genre.
- **setDescription(String description)**: Sets the description of the genre.
- **addBook(Book book)**: Adds a book to the collection of books belonging to this genre.
- **removeBook(Book book**): Removes a book from the collection of books belonging to this genre.
- **getBooks()**: Returns the collection of books belonging to this genre.

II. Scope Of Work 2: User Management (Erina)

1. **Member Class**

   Attributes: Name, Email, Address, Phone Number

   Methods: Getters and setters for attributes

2. **Admin Class**

   Attributes: Username, Password

   Methods: Login, Add book to inventory, Remove book from inventory, Generate report

3. **LibraryCard Class**

   Attributes: Card ID, Member object, Expiry Date

   Methods: Generate card ID, Renew membership

4. **Review Class**

   Attributes: Review ID, Book object, Member object, Rating, Comment

   Methods: Add review, View reviews for a book

III. Scope Of Work 3: Library Operations (Arini)

1. **LibraryInventory Class :** This class plays a crucial role in maintaining accurate records of the library's collection and availability of books.

- Attributes:
    - **bookList**: List<Book> - List of Book objects.
    - **bookQty**: Map<String, Integer>- Available quantity for each book.
- Methods:
    - **addBook(Book book)**: void - Add book to inventory
    - **updateBook(String isbn, int quantity)**: void - Update book quantity
    - **searchBook(String isbn)**: Search book by ISBN

2. **BorrowingRecord Class :** This class ensures accurate tracking of borrowing activities and helps maintain the library's borrowing records effectively.

- Attributes:
    - **recordID**: String - A unique identifier for each borrowing record.
    - **book**: Book - The Book object associated with the borrowing record.
    - **member**: Member - The Member object associated with the borrowing record.
    - **borrowDate**: Date - The date when the book was borrowed.
    - **returnDate**: Date - The expected return date for the borrowed book.
- Methods:
    - **generateRecordID()**: String - Generates a unique identifier for a new borrowing record.
    - **calculateFine()**: double - Calculates the fine or penalty for a late book return.
    - **updateReturnDate(returnDate: Date)**: void - Updates the return date for the borrowing record.
    - **searchBook(String isbn)**: Search book by ISBN

3. **ReservationQueue Class :** This class will ensure a fair and organised reservation process, allowing members to reserve books that are currently unavailable and receive them once they become available.

- Attributes:
    - **bookQueue**: Queue<Book> - A queue that holds the reserved books.
    - **member**: Member - The member associated with the reservation.
- Methods:
    - **addBookToQueue(book: Book)**: void - Adds a book to the reservation queue.
    - **removeBookFromQueue(book: Book)**: void - Removes a book from the reservation queue.
    - **getNextBookInQueue()**: Book - Retrieves the next book in the reservation queue.

4. **Notification Class :** This class will facilitate important communication, such as notifying members about reserved books, overdue items and upcoming events.

- Attributes:
    - **notificationID**: String - A unique identifier for each notification.
    - **recipient**: Member - The member who receives the notification.
    - **message**: String - The content of the notification.
    - **timestamp**: Date - The timestamp when the notification is sent.
- Methods:
    - **sendNotification(recipient: Member, message: String)**: void - Sends a notification to a member.
    - **viewNotificationsForMember(member: Member)**: List<Notification> - Retrieves the notifications for a specific member.