VARENDRA UNIVERSITY

বরেন্দ্র বিশ্ববিদ্যালয়

Dept. of CSE
Varendra University

Department of Computer Science & Engineering

Project Proposal

Project Title: Library Management System

Course Title: Software Development Lab I

Course Code: CSE-136

Batch No: 31st

Semester: 3rd Semester

Section: E

Team: 1

Submitted by	Submitted to
Name: Md Borhan Uddin Ashik ID: 223311161	A.S.M. Delwar Hossain Lecturer Department Of CSE Varendra University
Name: Abu Raihan Md Sohag ID: 223311162	
Name: Afia Akhter ID: 223311163	*Signature*

Date of Submission: 13/11/2023

Motivation:

After seeing Varendra University library system, I feel that it needs to be more student friendly and improved. Only the librarian could know whether there are any required books in the library, the books of any publication, how many books are in stock through internal apps. Which seems like a lot of trouble and time consuming for students like us. We thought how to improve this system and bring that service to the student level.

Objectives:

The primary objectives of this project are to:

1. Develop a student-friendly and efficient LMS: The system will provide a seamless interface for librarians and patrons, enabling them to manage library resources and access information with ease.
2. Enhance library operations: Streamline book cataloging, circulation management, allowing librarians to focus on providing exceptional service to patrons.
3. Every student can easily find his desired book without going to the library or asking the librarian.
4. Student can also see here his total number of books issued, total number of publications and number of categories of books.
5. This system has an account creation and login system to identify the identity of the issuer.
6. Librarians can view all issues and easily view stock out books.
7. Librarians can easily add new books to the library. And which book can be deleted from the library database.

Overview:

- **Description of the Project:**

  The Library Management System is a software application developed in C++. It is designed to overcome the manual record-keeping process in libraries. The system utilizes efficient data structures for faster retrieval and storage of data. The problem it solves is the inefficiency, inaccuracy, and lack of scalability of manual library management, replacing it with an automated, scalable, and user-friendly system.

  - The scope of a Library Management System in C++ includes but is not limited to:
  - Keeping track of the return dates and checking for overdue books.
  - Providing a user-friendly interface for librarians to manage the system.

- Develop a comprehensive cataloging system for efficient organization and retrieval of library resources.
- Enable users to search for books based on various criteria, including book name, isbn etc.
- Create and manage user profiles, storing relevant information such as borrowing history and personal details.

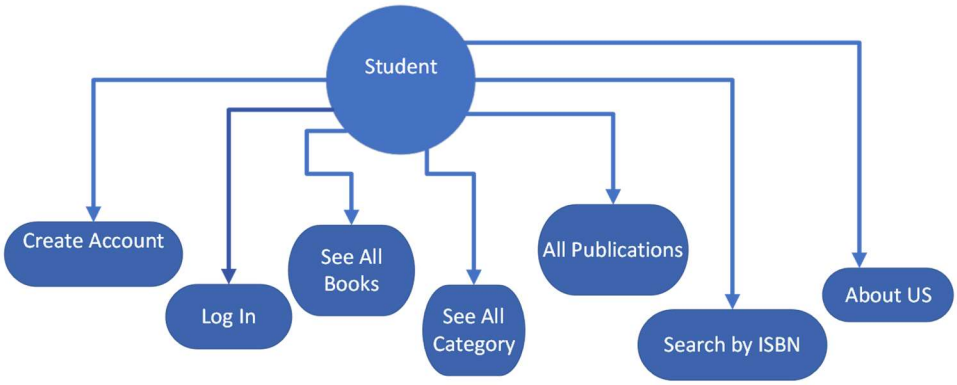
♦ **Materials and Methods:**

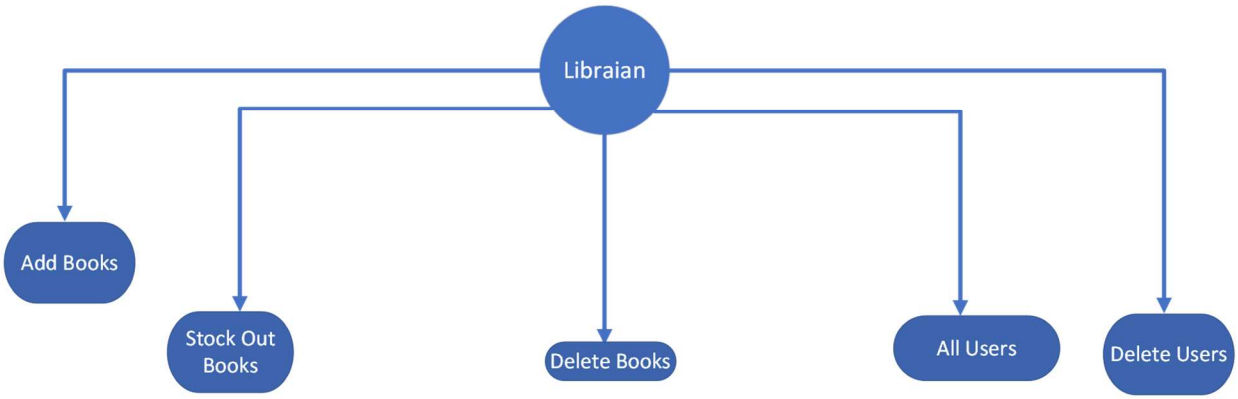
The project is developed using the C++ programming language, leveraging its Object-Oriented Programming (OOP) capabilities. The system is designed around various classes representing entities such as 'LMS', 'Users', and 'Books'. These classes encapsulate the data and provide methods for interacting with that data, adhering to the principles of encapsulation, inheritance, and polymorphism.

The Standard Template Library (STL) is used for implementing efficient data structures, and file handling methods in C++ are used for persistent storage of data. Each class is responsible for managing its own data, ensuring a high degree of cohesion and low coupling, which are key principles of OOP.

As for the Data Flow Diagram (DFD), it would typically include entities like 'Librarian', 'System', and 'Database', and show how data flows between them during different processes like 'Add/Update/Delete Book', 'Issue Book', 'Return Book', etc. Unfortunately, I can't draw a DFD here, but there are many online tools available that you can use to create one.

Use case Diagram:





Features:

1. **User-Friendly Interface:** The system has a simple and intuitive interface, making it easy for librarians and users to interact with the system.
2. **Efficient Data Structures:** The system uses efficient data structures provided by the Standard Template Library (STL) in C++, ensuring fast retrieval and storage of data.
3. **Object-Oriented Design:** The system is designed using Object-Oriented Programming (OOP) principles. This makes the code more manageable, scalable, and reusable.
4. **Persistent Data Storage:** The system uses file handling methods in C++ for storing and retrieving data. This ensures that data is not lost when the system is closed.
5. **Comprehensive Record Keeping:** The system keeps track of various details such as books issued, due dates, late fees, etc. This helps in maintaining a clear record of all transactions.
6. **Search Functionality:** The system provides a search functionality for book and member records. This helps in quickly finding the required information.
7. **Data Consistency and Integrity:** The system ensures data consistency and integrity. Any changes in the data are accurately reflected across the system.

Software Requirements/Tools:

To successfully complete the Library Management System project in C++, you would need the following software tools:

1. **C++ Compiler:** A C++ compiler is required to compile and run your C++ code. Examples include GCC, Clang, and MSVC.
2. **Integrated Development Environment (IDE):** An IDE can help you write, compile, and debug your code more efficiently. Examples include Code Blocks and Visual Studio.
3. **Version Control System:** A version control system like Git can help you manage different versions of your project, especially if you are working in a team.
4. **UML Tool:** A tool for creating UML diagrams (like Data Flow Diagrams) can be useful for planning and visualizing your project. Examples include Draw.io, Lucid chart, and Microsoft Visio.
5. **Text Editor:** A text editor can be useful for writing and editing your project documentation. Examples include Visual Studio Code.

Timeline:

The project is expected to take 2 months to complete. The following is a tentative timeline:

Phase 1: Project Planning and Design (1 months)

- Requirements gathering
- System design and architecture
- User interface prototyping

Phase 2: Development (1 months)

- Core functionality development
- Unit testing and integration testing
- User acceptance testing

Future Enhancement:

We acknowledge that our current project has many limitations. With these, we will try to improve the project by adding more features in the future. Among the features that I would like to add to this project are:

1. We will bring it to web version in future.

2. Various roles will be added like admin, librarian, general student etc.

3. A student can easily view the issued book and get a reminder to submit the issued book by email.

4. From university student data to authenticate users

Account creation will be allowed after cross checking the user data.

5. For easy account creation, Student Info scans the QR code on the ID card if implemented.

6. To get the ISBN, the book's QR code scan or bar code scan will have the option to add it easily.

7. Librarians can easily view stock out books.

8. Once the date of return of the issued books is over, the list of those students can be easily viewed.

9. Librarians can view monthly, yearly reports on their dashboard.

10. There will be an admin who can see all library users, books, librarian reports on the dashboard.

Conclusion:

The Library Management System project in C++ is a comprehensive solution for managing library operations. It leverages the power of Object-Oriented Programming (OOP) and efficient data structures provided by the Standard Template Library (STL) in C++. The system is designed to be user-friendly and efficient, ensuring a smooth experience for both librarians and users.

The project stands out due to its comprehensive features, including efficient record keeping, search functionality, automated reminders, and ensuring data consistency and integrity. The use of appropriate software tools like a C++ compiler, an Integrated Development Environment (IDE), a version control system, a UML tool, and a text editor will aid in the successful completion of the project.