**University:** Jomo Kenyatta University Of Agriculture And Technology

**Course:** Bachelor Of Science Computer Science

**Title:** A proposal for implementing a library management system

**Group members**

* Eliud Marierie SCT211-0094/2022
* Rotich Kibet Asbel SCT211-0087/2022
* Jacob kasunzu SCT211-0089/2022
* Collins sifa Kasena SCT211- 0066/2022
* Kelvin mwenda SCT211-0062/2022
* Moses muiruri SCT211-0002/2022

# Functional Requirements:

### Books:

* Add Book
* Retrieve Books
* Update Book
* Delete Book

### Borrower:

* Add New Borrower
* Retrieve Borrower
* Update Borrower
* Delete Borrower

### Transaction:

* Borrow Books
* Return Books
* Overdue with Payments

### Category:

* Add New Category
* Retrieve Categories
* Update Category
* Delete Category

### Manage Users:

* Add New User
* Retrieve Users
* Update User
* Delete User

### Report:

* Inventory of Books
* List of Books

### User Logs:

* Login and Logout

# Non-Functional Requirements:

### Performance:

- The system should respond to user actions within 2 seconds, even under heavy load.

### Reliability:

* + The system should be available 99.9% of the time, with scheduled downtimes for maintenance communicated in advance.

### Security:

* + User data should be encrypted both in transit and at rest.

### Usability:

* + The user interface should be intuitive, requiring no more than 30 minutes of training for new users.

### Scalability:

* + The system should support up to 10,000 concurrent users without performance degradation.

### Compatibility:

* + The system should be compatible with the latest versions of Chrome, Firefox, Safari, and Edge browsers.

### Maintainability:

* + The system should be easy to maintain, with updates and patches applied within 48 hours of release.

### Regulatory Compliance:

* The system should comply with GDPR regulations, ensuring the protection of user privacy and data rights.

## HARDWARE REQUIREMENTS:

* Processor: Intel Core i5 or equivalent
* RAM: 8GB or higher
* Storage: 256GB SSD or higher
* Display: 1920x1080 resolution or higher
* Internet Connection: Broadband for downloading development tools and updates
* Server (if applicable): Minimum requirements for hosting the application, depending on expected usage and traffic.

## SOFTWARE REQUIREMENTS

### Development Language:

VB.NET for backend development.

### Frontend:

For a web-based application: ASP.NET with HTML, CSS, and JavaScript. For a Windows desktop application: Windows Forms or WPF.

### Database:

SQL Server or MySQL for database management, easily connectable with VB.NET.

### Server (for web application):

IIS (Internet Information Services) for hosting the ASP.NET application.

### Development Tools:

Microsoft Visual Studio for developing and debugging both VB.NET and ASP.NET applications.

### Testing Tools:

For VB.NET applications: Microsoft Test Manager or NUnit.

## SCHEDULE/PROJECT PLAN:

### Week 1: Conceptualization and Design

* Define detailed specifications and functionalities with a focus on VB.NET capabilities.
* Design the user interface, using Windows Forms or WPF, for web application use ASP.NET.

### Week 2-3: Backend Development with VB.NET

* Develop the user authentication, profile management, and database interactions using VB.NET.
* Implement server-side logic for book management, borrower management, and transaction processing.

### Week 4-5: Frontend Development and Integration

* For a web application, develop the frontend using ASP.NET with HTML, CSS, and JavaScript.
* For a desktop application, build the user interface with Windows Forms or WPF.
* Integrate the frontend with the backend services.

### Week 6: Testing and Refinement

* Conduct thorough testing of the application, focusing on functionality, usability, and database integration.
* Refine based on test results and user feedback.

### Week 7: Deployment and Finalization

* + Deploy the web application to IIS or finalize the desktop application for distribution.
  + Final bug fixes, optimization, and preparation for project presentation.

## FEASIBILITY STUDY

### Technical Feasibility:

* + Assessment: The technical feasibility analysis indicates that the development and implementation of the library management system are feasible. The chosen development language (VB.NET) and database management system (SQL Server or MySQL) are compatible with the existing infrastructure and systems

.

* + Conclusion: The technical requirements and capabilities necessary for the project are readily available, and there are no significant technical challenges or limitations that would hinder the feasibility of the project.

### Economic Feasibility:

* + Assessment: The economic feasibility analysis reveals that the initial investment required for developing the library management system, including hardware, software, and development costs, is within budget constraints. Additionally, the potential return on investment (ROI) and cost savings resulting from improved library management efficiency are significant.
  + Conclusion: Considering the potential benefits and cost-effectiveness of the project, it is economically feasible to proceed with the development and implementation of the library management system.

### Operational Feasibility:

* + Assessment: The operational feasibility assessment indicates that the library management system will have a positive impact on day-to-day operations, workflows, and staff responsibilities. Library staff have demonstrated readiness and willingness to adapt to and utilize the new system effectively, and there are no significant organizational or cultural barriers that would impede the successful implementation of the system.
  + Conclusion: The operational aspects of the project are feasible, and the library management system is likely to be well- received by users and staff.

### Risk Assessment:

* + Assessment: The risk assessment identifies several potential risks and challenges associated with the development, implementation, and adoption of the library management system, including technical issues, user resistance, and budget constraints.
  + Conclusion: While there are risks involved, the likelihood and potential impact of each risk have been evaluated, and mitigation strategies and contingency plans have been developed to address them. With proper risk management measures in place, the project remain feasible.

### Feasibility Report:

* + Based on the comprehensive feasibility assessment conducted, the project team recommends proceeding with the development and implementation of the library management system. The technical, economic, and operational aspects of the project are feasible, and the potential benefits outweigh the risks and challenges. A detailed feasibility report has been compiled and presented to stakeholders, outlining the strengths, weaknesses, opportunities, and threats associated with the project and providing recommendations for moving forward.

## DATABASE DESIGN

### Book

* + BookID (Primary Key): A unique identifier for each book.
  + Title: The title of the book.
  + Author: The author of the book.
  + ISBN: International Standard Book Number, a unique identifier for books.
  + CategoryID (Foreign Key): Links to the Category entity to categorize the book.
  + PublicationYear: The year the book was published.
  + Status: Current status of the book (Available, Borrowed, Reserved).

### Category

* + CategoryID (Primary Key): A unique identifier for each category.
  + CategoryName: The name of the category (e.g., Fiction, Science, History).

### Borrower

* + BorrowerID (Primary Key): A unique identifier for each borrower.
  + Name: The name of the borrower.
  + Address: The address of the borrower.
  + ContactNumber: The contact number of the borrower.
  + Email: The email address of the borrower.

### Transaction

* + TransactionID (Primary Key): A unique identifier for each transaction.
  + BorrowerID (Foreign Key): Links to the Borrower entity to identify the borrower in the transaction.
  + BookID (Foreign Key): Links to the Book entity to identify the book involved in the transaction.
* IssueDate: The date when the book was issued.
* ReturnDate: The date when the book was returned.
* DueDate: The date when the book is due to be returned.Status: The status of the transaction (Issued, Returned).

### User

* + UserID (Primary Key): A unique identifier for each user.
  + UserName: The username of the user.
  + Password: The password for the user account.
  + Role: The role of the user in the system (Admin, Staff, User).

### Book Issue Log

* + Id (Primary Key): A unique identifier for each book issue log entry.
  + BookIssueId (Foreign Key): Links to the Book Issue entity to identify the associated book issue.
  + StudentId (Foreign Key): Links to the Student entity to identify the student involved in the issue.
  + IssueBy (Foreign Key): Links to the User entity to identify the staff member who issued the book.
  + IssuedAt: The timestamp when the book was issued.
  + ReturnDate The timestamp when the book was returned.
  + TimeStamp: The timestamp of the log entry.

### Students

* + StudentId (Primary Key): A unique identifier for each student.
  + FirstName: The first name of the student.
  + LastName: The last name of the student.
  + Approved: Approval status of the student (0 or 1).
  + Rejected: Rejection status of the student (0 or 1).
  + Category: Category of the student.
  + Year: The academic year of the student.
  + BooksIssued: The number of books issued to the student.
  + EmailId: The email address of the student.

### Staff

* + StaffID
  + Name
  + Position

### Book record

* + BookId
  + TransactionId
  + DueDate
  + ReturnDate

### Role

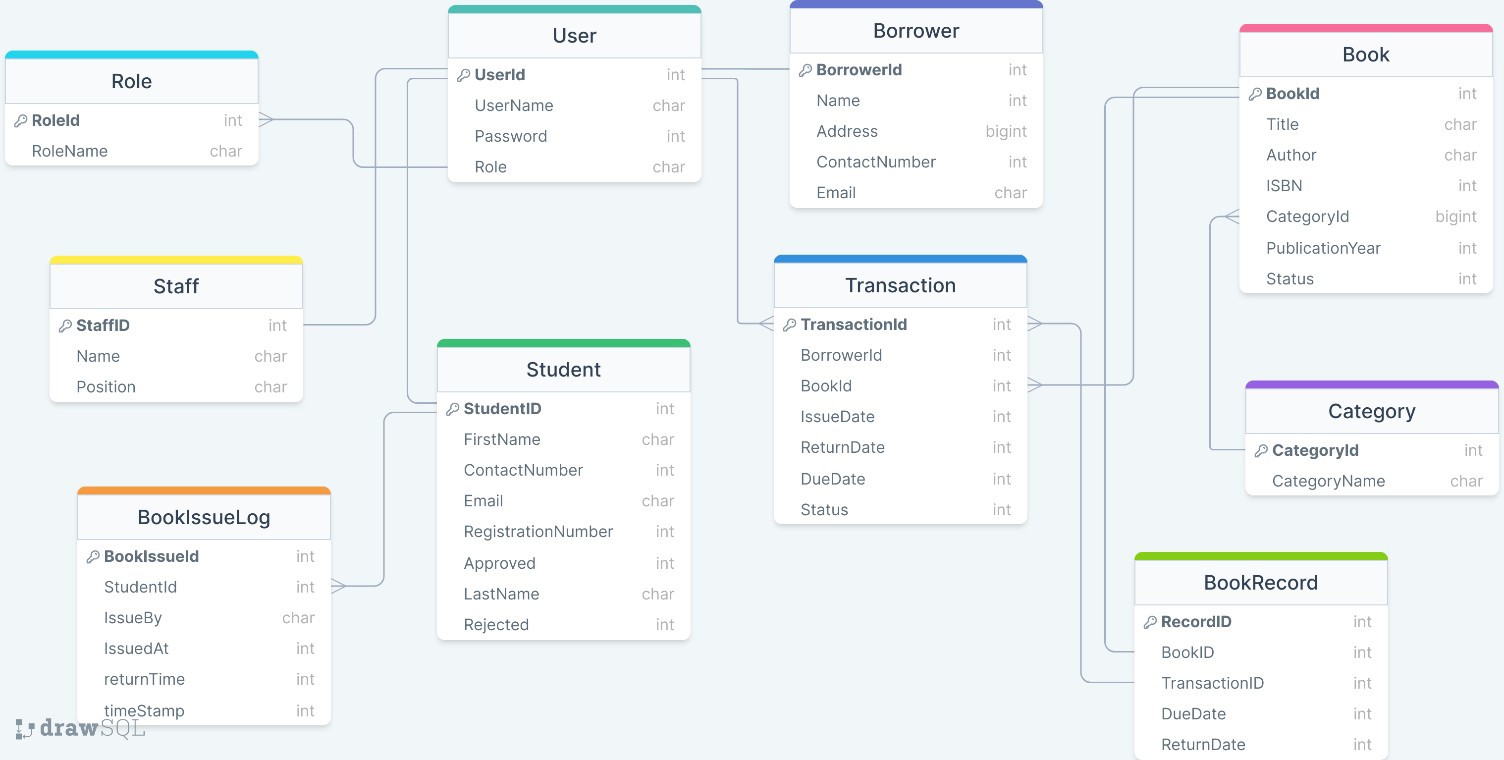
* + RoleId
  + RoleName

**Relationships:**

 A Book belongs to a Category. Each book is associated with one category, while each category can have multiple books. A Borrower is involved in multiple Transactions. Each transaction records the borrowing activity of a borrower.

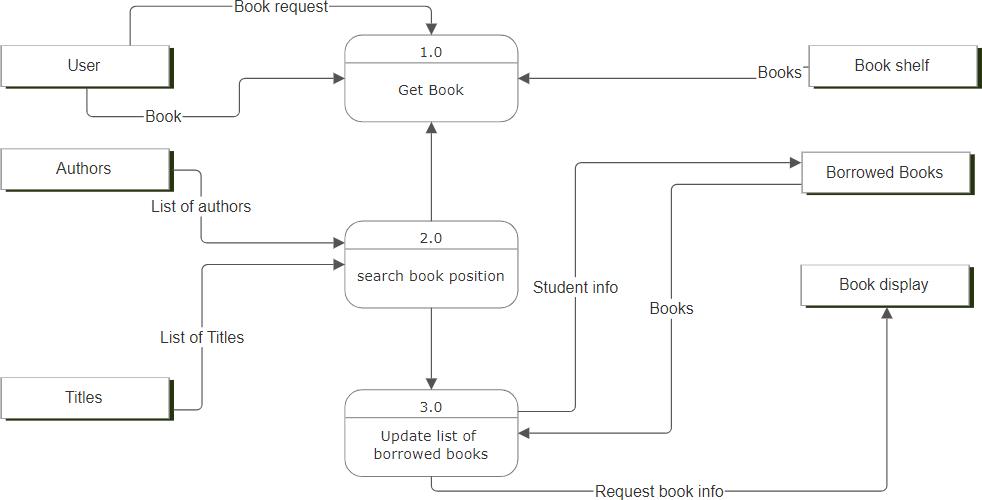
 Each Transaction includes one Book. A book can be part of multiple transactions over time.

This database design effectively organizes and manages the data necessary for a library management system, ensuring efficient retrieval and maintenance of information about books, borrowers, transactions, and users.



## DATAFLOW

Here is a data flow diagram (DFD) for a library management system. This diagram visually represents the flow of data between external entities, processes, and data stores within the system. It includes external entities like library users (borrowers), library staff, and external library systems, and processes such as book issue, book return, book search, user registration, user authentication, inventory management, and reporting. The diagram also illustrates the data stores involved, such as user data, book data, transaction data, and category data. The flow of data is indicated by arrows, showing how information moves through the system.



## FLOW CHART

Here is the flowchart for the library management system project. This diagram visually represents the sequence of steps involved in the project, from project initiation to maintenance and support, using standard flowchart symbols. It outlines each critical phase of the project, providing a structured roadmap for its development and implementation.

