

## Introduction:

- Project Overview: The purpose of this database is to store a collection of different books, magazines, movies able to be rented out by different users.
- **Scope:**
  - The purpose of a Library Management System is to make a useful, scalable, and educational tool that works like a real library. As part of the project, we will be able to work directly with all aspects of database development, from conceptual modeling to logical design to physical implementation. We will also add ways to make sure that borrowing rules are followed and useful reports will be made to help with operational decisions, collection growth, and member participation. At the end of the day, this project will be both a useful way to learn and show how well-designed database systems can help solve difficult information management issues.
- **Glossary:**
  - ISBN: International Standard Book Number; is a unique, 13-digit numerical identifier for a specific book edition and format, used by libraries, bookstores, and online retailers to catalog, track, and sell books globally.
  - SQL: Structured Query Language
  - Functional Requirement: specify what a system must do, detailing the specific actions, functions, and features it needs to perform to meet user and business needs
  - Non-Functional Requirement: specify what a system must do, detailing the specific actions, functions, and features it needs to perform to meet user and business needs

## Stakeholders:

- Library Staff (primary users)
- Library Members (clients)
- Database Administrators (team/developers)

## Requirements:

- *Functional Requirements:*
  - Adding books, adding users, seeing what books are available, renting a book, changing user statuses, removing books,
  - The software needs to have a way that administration and maintenance can access the code (user administration), be able to insert data (data entry), call on stored data (retrieval), have the ability to change stored data (updates), remove stored data (deletions), and generate reports to the user/stakeholder's specification (report generation).
- *Data Entities:*
  - Book
    - Title

- Author/creator
  - ISBN
  - Publication year
  - Genre
  - Availability status/ amount available
  - CopyID
  - Number of check outs
- Magazine
  - Title
  - Issue number
  - Publication date
  - Availability status
  - Publisher
  - Category
  - CopyID
  - Check outs
- Movies
  - Title
  - Release date
  - Genre
  - Rating
  - Director
  - Number of check outs
- Client
  - ID
  - First name
  - Last name
  - Phone number
  - Email address
  - Home address
  - Account status
  - Late fee amount
  - Which items has checked out
- Constraints
  - One person can only borrow 5 books at a time
  - One person can only borrow one book for 3 weeks at a time
    - Can renew the book once
  - After the book is overdue late fees will start occurring at 25 cents a day
  - One cannot check out a book if they haven't paid their late fees
- *Non-functional Requirements (Optional):*
  - Send notification by email if there is a book about to be late
  - Reserve a book

- Security / only admins can see client information
- Performance, for fast query results
- Reliability, backups of data

**Hardware and Software Requirements:**

- *Database Engine*: MySQL if local, MariaDB (or SQL) if from the school servers.
- *Hardware*: Laptop if local, EECS cycle servers if remote.
- *Software*: Github for version control, server to store data (if applicable), python to provide a user interface, connect to database to enforce rules, and to automate reports.

**Appendices:** //what we can/want to add to project

Github Repository Management:

[EECS 447 Github Repo](#)