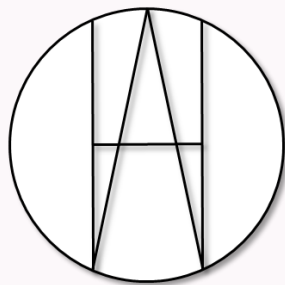


# Summit Logistics



## Library Database

Ithaca Local Library  
832 Park Street  
(666)-452-7892

# Table Of Contents

## Who Are We

Mission and Goals	3
Core Values	3
Team Roles	3

## Client

Background	4
Current System	4
Problem with Current System	4

## Objectives

Purpose	5
Users and Requirements	5
Scope	6
User Requirements	6
Use Case Scenarios	6

## Feasibility

User Expectations	7
Budget	7
Timeline	7
Project Feasibility	7
Business Rules, ERD, and Schema	8-9

# Who Are We

## Mission and Goals

Our mission at Summit Logistics is to bring the future to small businesses. For those who are struggling with paper or very minimal systems, we will upgrade them to more efficient and effective systems. We pride ourselves on putting the customer first and ensuring we give our clients all they need and more.

## Core Values

Customer Satisfaction - We strive for all customers to be satisfied with their solutions. We will stop at nothing to make our clients happy.

Efficiency - Everything must work smoothly; both our team and our databases.

Dependability - We want our clients to trust and rely on us to get things done.

## Team Roles

Alex Burdick - Software developer: Alex has an education in computer science, and specializes in integrated systems and automation.

Owen Schnell - Designer and Programmer: In charge of basic design of the database structure (i.e., what information goes into the database, what are the table headers, etc). Also helping with programming to create the database itself.

Hailey Fiske - Programmer and graphic designer: In charge of creating a clean and easy to understand user interface.

# Client

## Background

Ithaca Local Library is a public library located in Ithaca, NY. The library is housed in a small, one story building. They collect and lend out books for free to the local community. Their staff is made up of four people; a librarian, a stocker, a book procurer, and a maintenance person.

## Current System

The library's information system for tracking inventory and customers is almost entirely paper-based. They use a card catalog to look up and find inventory, and checkout cards with stamps (name of library, book title and author, and a list of who took it out with date issued) for history. Since the OCLC has stopped making library cards, the library can no longer get any new ones. They have computers on site that are used for web browsing. The librarian keeps an Excel spreadsheet with customer information which needs to be referenced when contacting customers. However the list of books checked out is kept in a paper spreadsheet, and the two must be cross referenced when they notify customers of overdue returns. The list of the library's collection is also maintained on paper, with books that they no longer have crossed out.



## Problems with Current System

The problems with the paper system are time, efficiency, and human error. The list of total inventory has redundant information, which is not checked in any way and is incorrect in places. A single customer's information is sometimes recorded in multiple entries in the spreadsheet. The list of books that have been lent out is not sorted in any way, and it is easy to overlook books that are overdue but have been renewed. Checkout cards get lost or incorrectly filled out. Additionally, the library does not keep a history of any of the checkout information. When cards are full they are discarded and

replaced. The same is true for the book that contains checkout information.

Because most of the library's information is not on a computer, it is not accessible through the internet. This means that customers have to go to the library to see what books are in the collection and available. This is inconvenient and likely discourages people from going to the library.

## Objectives

### Purpose

Our aim is to make the inventory management more efficient, accurate, and accessible. Moving the information from paper to computers will allow them to enforce data integrity, automate tracking and overdue notices, and make specific information available outside of the library's walls. The ability to check the library's book availability should increase library usage, which has been on the decline in recent years.

### Users and Requirements

- Stocker: check books in/out, view book locations within the library, see the availability of books
- Procurer: add/remove books from the collection, check books in/out, view book locations within the library, see the availability of books
- Librarian: view book locations within the library, add/remove books from the collection, see the availability of books, check books in/out, view book locations within the library, see the availability of books, send overdue notifications
- Customers: view book locations within the library, see the availability of books, check books out

## Scope

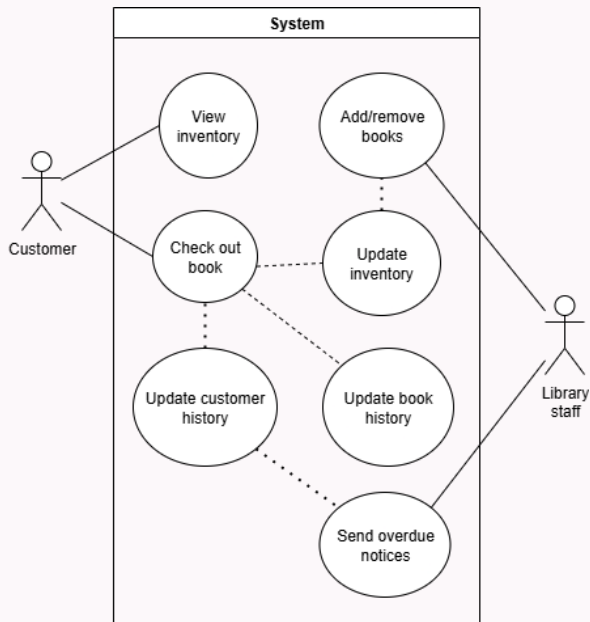
- Keep detailed information about individual books, publishers, genres, authors, and awards
- Tracking books (in or out of the library) for finding who has books and availability
- Store book locations (where they belong in the library) to know where they are placed within the library
- Checkout history of individual books
- Customer transaction history
- Allow for books to be added/removed from the collection

## User Requirements

- Security - customer information must be private and only accessible to certain library staff.
- Interface - library staff and customers will need to be able to interact with the library inventory information, both on premises and over the internet.
- Data protection and recovery - the client will need to be able to backup data digitally and on a paper record. They will also need to print situation specific information.

## Use Case Scenarios

- Stocker looks up the section and aisle to find the location to replace a returned book
- Librarian finds information about the customer who checked out a book to send an overdue reminder
- Procurer purchases a new book for the library and adds it to the collection
- Customer views books that are in the collection, and available at the library



# Feasibility

## User Expectations

Users of the Summit Logistics Database System (SLLDS) will be able to reduce paper waste and save time otherwise spent searching through physical catalogs across the library. The system will also allow users to locate specific books within the library using the database more quickly and efficiently than physical catalogs and other book sorting systems.

## Budget

One-time fee for design, installation, and a yearly charge for maintenance.

## Timeline

4 months to design and implement, and then monthly maintenance after.

## Project Feasibility

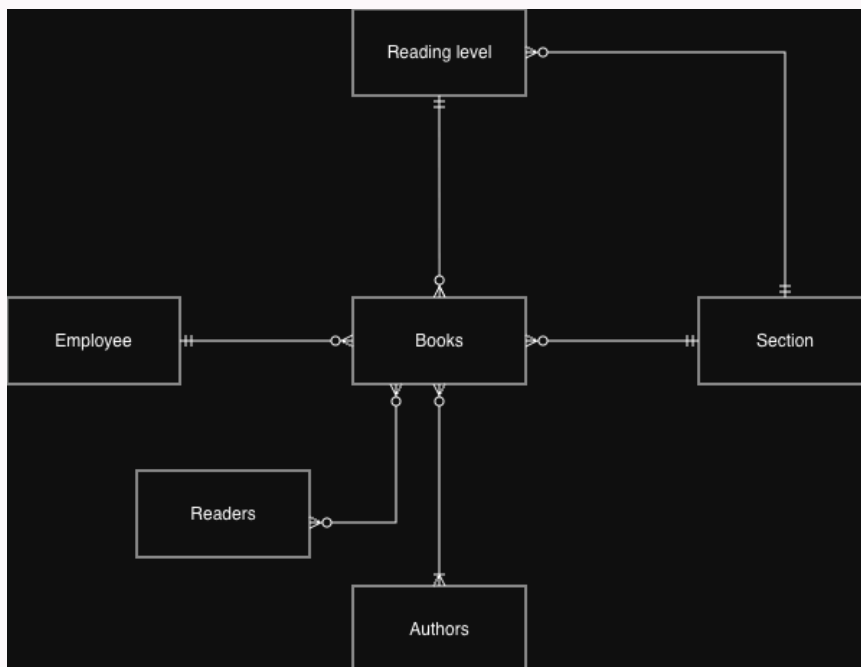
The Summit Logistics Library Database System provided to the Ithaca Public Library will help increase book checkout efficiency as well as decrease late

books and stolen books while also being deployable to other local libraries around the Finger Lakes region. While the current paper-based system is functional, it lacks control and efficiency. Switching to the Summit Logistics Library Database System will increase book checkout times as well as locating books. Overall, improving the efficiency and quality of life of going to the library.

## Business Rules

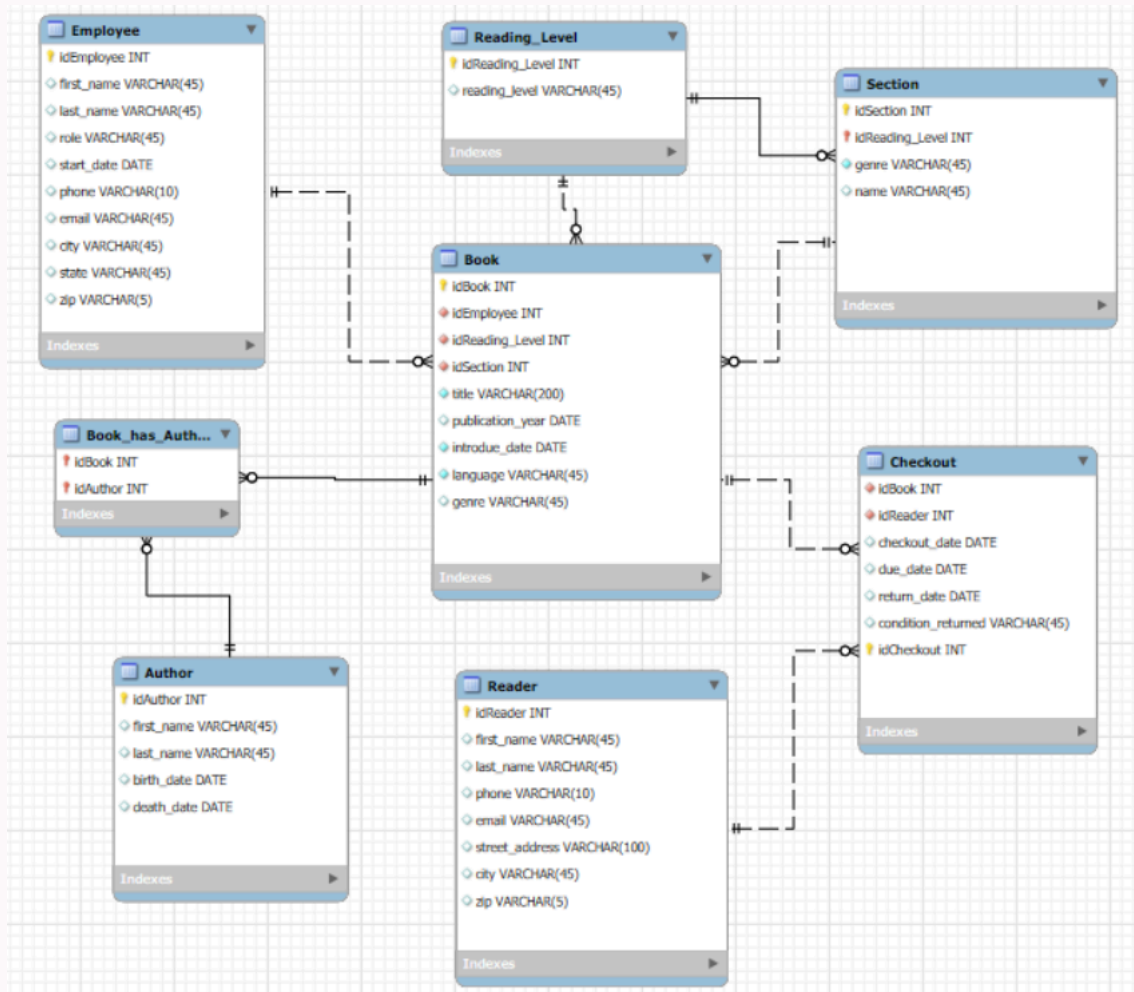
- An Employee can remove or add many Books. A Book is added or removed by one Employee.
- A reader can check out many books. A book can be checked out by many people.
- A book must have at least one author. An author can have many books.
- A book belongs in one section. A section can have many books.
- A Book must have one Reading Level. A Reading Level can be assigned to many Books.
- A Section must have one Reading Level. A Reading Level can be for many Sections.

## ERD





## Schema



## Textual Tables

**Book**(idBook(PK), idEmployee(FK), idReading\_Level(FK), idSection(FK), title, publication\_year, introduce\_date, language, genre)

**Employee**(idEmployee(PK), first\_name, last\_name, role, start\_date, phone, email, city, state, zip)

**Reader**(idReader(PK), first\_name, last\_name, phone, email, street\_address, city, zip)

**Author**(idAuthor(PK), first\_name, last\_name, birth\_date, death\_date)

**Checkout**(idCheckout(PK), idBook(FK), idReader(FK), checkout\_date, due\_date, return\_date, condition\_returned)

**Section**(idSection(PK), idReading\_Level(FK), genre, name)

**Reading\_Level**(idReading\_Level(PK), reading\_level)

**Book has Author**(idBook(PK,FK), idAuthor(PK,FK))

## Functional Dependencies

idBook -> idEmployee, idReading\_Level, idSection, title, publication\_year, introduce\_date, language, genre

idEmployee -> first\_name, last\_name, role, start\_date, phone, email, city, state, zip

idReader -> first\_name, last\_name, phone, email, street\_address, city, zip

idAuthor -> first\_name, last\_name, birth\_date, death\_date

idCheckout -> idBook, idReader, checkout\_date, due\_date, return\_date, condition\_returned

idSection -> idReading\_Level, genre, name

idReading\_Level -> reading\_level

## User Manuel

How to use this code! Source the following .sql files in this order:

1. Create database -> library\_management\_creation.sql
2. Insert data -> library\_management\_inserts.sql
3. Add procedures file -> library\_management\_procedures.sql
4. To see all checked out books, run -> get\_checked\_out.py.
5. To add a book to the database, run -> add\_book.py
6. Done!

## Group Contributions

Alex: wrote sql statements, wrote python scripts, created database schema, and wrote sections of the pitch and business rules

Hailey: designed pitch document + logo, wrote mission + goals, core values, and background section, designed tables and business rules with group, wrote textual table listings and functional dependencies, used chatgpt to create some insert statements, wrote some of user manuel, tested code

Owen: created erd, wrote notes, inquired about design choices, and wrote sections of pitch and certain rules