**Final Project Proposal Prep**

**Project Participants:**

Adrian (Ed) S. Locquiao

**Title:**

**P**ersonal **L**ibrary **U**tility **M**anagement **S**ystem (PLUMS)

**Executive Summary**

The library comes in many shapes, forms, sizes, and carries many kinds of reading materials called books. The public library and school library are great examples. The internet, technology, culture, and individual preferences dictate public library usage, but the Covid-19 quarantine paved the way for the Personal Library Utility Management System (PLUMS).

The final project discusses the key features of the PLUMS system that is first created utilizing key concepts from Java, MySQL, and Spring Boot. Initially, the PLUMS system allows a family member to manage readers, to create book title entries, and to search and view books. Future updates include the ability to create individual libraries, update and delete book entries, and add additional book fields.

**Initial Features**

My List of Features/Endpoints for a PLUMS API

**Entities and Methods**

All MySQL identifiers listed in the following sections are listed in parentheses.

**Entity:** Book | **Table:** Books (books)

**Properties:**

* + Primary key [INT(4)]: Book Number (booknumber\_pk) Not Null
* ISBN [VARCHAR(30)]: (isbn) Not Null
* Genre [enum]: (genre)
  + {'ADVENTURE', 'COMPUTER FORENSICS', 'CYBERSECURITY', 'EDUCATION', 'FICTION', 'JAVA', 'RECIPES', 'SCIFI', 'SPIRITUAL', 'THRILLER'}
* Title [VARCHAR(26)]: (title)
* Author/Authors[VARCHAR(128)]: (book\_authors)
* Notes [TEXT(52)]: (notes)

**Methods**

* Create a book entry
* Read (locate) an entry and view
* Update book status
* Delete book
  + method that deletes a book by its book number

**Entity:** Reader | **Table:** Reader (readers)

**Properties:**

* Primary key [INT(4)]: Reader ID number (readerid\_pk) Not Null
* Foreign key [INT(4)]: Book Number (booknumber\_pk) Not Null
* Reader Id Identity [VARCHAR(40)]: (readerid\_id)
* First name [VARCHAR(26)]: (first\_name)
* Last name [VARCHAR(42)]: (last\_name)
* Phone number [VARCHAR(12)]: (phone\_number)
* Email [VARCHAR(32)]: (email)

**Methods**

* Create reader entry
  + validate that there is at least one book to assign
* Read reader entry
  + Get a reader entry by ID Number or Book Number

**Entity:** Library (collection of books) | **Table:** Libraries (libraries)

**Properties:**

* Primary key [INT(4)]: Library ID number (libraryid\_pk) Not Null
  + validate no null book entry (or entries)
* Foreign key [INT(4)]: Book Number (booknumber\_pk)

**Methods**

* Create a library entry
  + validate that book and reader are not null

**Entity:** Library\_Book (Many to Many)

**Table:** Libraries\_Books (libraries\_books)

* Primary key [INT(4)]: Library Book number (librarybookid\_pk)
* Foreign Key #1 [INT(4)]: Library ID number (libraryid\_pk)
* Foreign Key #2 [INT(4): Book Number (booknumber\_pk)

**Entity:** PLUMS Main\_Database

**Table:** PLUMS Main Database (plums\_main\_db)

* Primary key [INT(4)]:PLUMS ID number (plumsid\_pk)
* Foreign key [INT(4)]: Reader ID number (readerid\_pk)
* Foreign Key [INT(4)]: Library ID number (libraryid\_pk)
* Foreign Key [INT(4)]: Book Number (booknumber\_pk)

**Stretch Goals**

The features below are planned for creation only after initial project completion. Additionally, these features also may require additional research.

**Reader**

* Update a reader entry
* Delete a reader entry

**Library**

* Read a library list
* Update a library list
* Delete a library list

**Testing**

* Red-Green-Refactoring
  + on methods reader and library methods

**Additional Notes**

**Relationships**

* One-To-Many
  + Reader to book
* Many-To-Many
  + Libraries to books
    - Library ID number
    - Book number