Gabriel Quiñones-Sanchez
Joseph Doctor
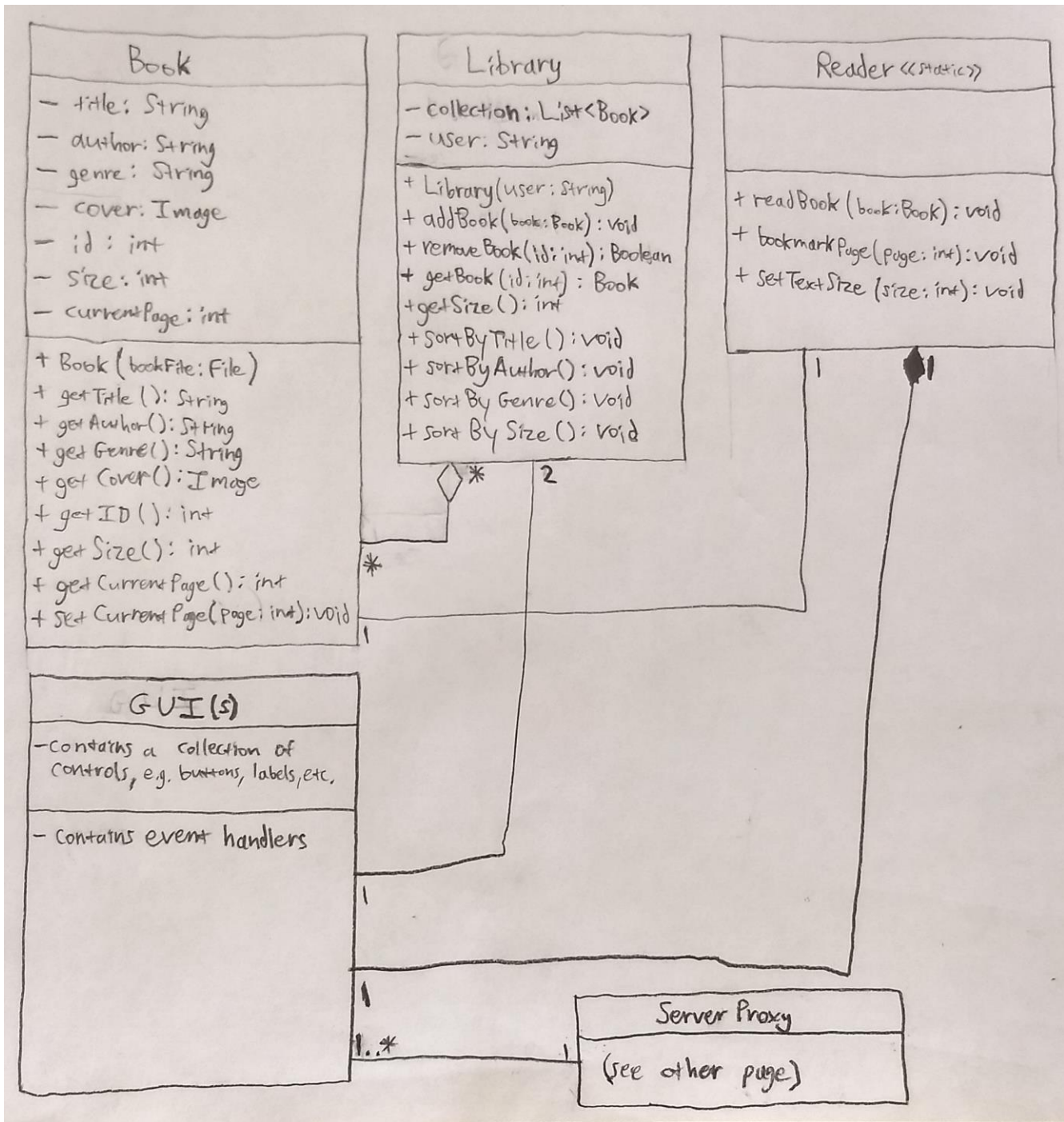ACS-560

Gutenberg Library: Design Document

System Modules:
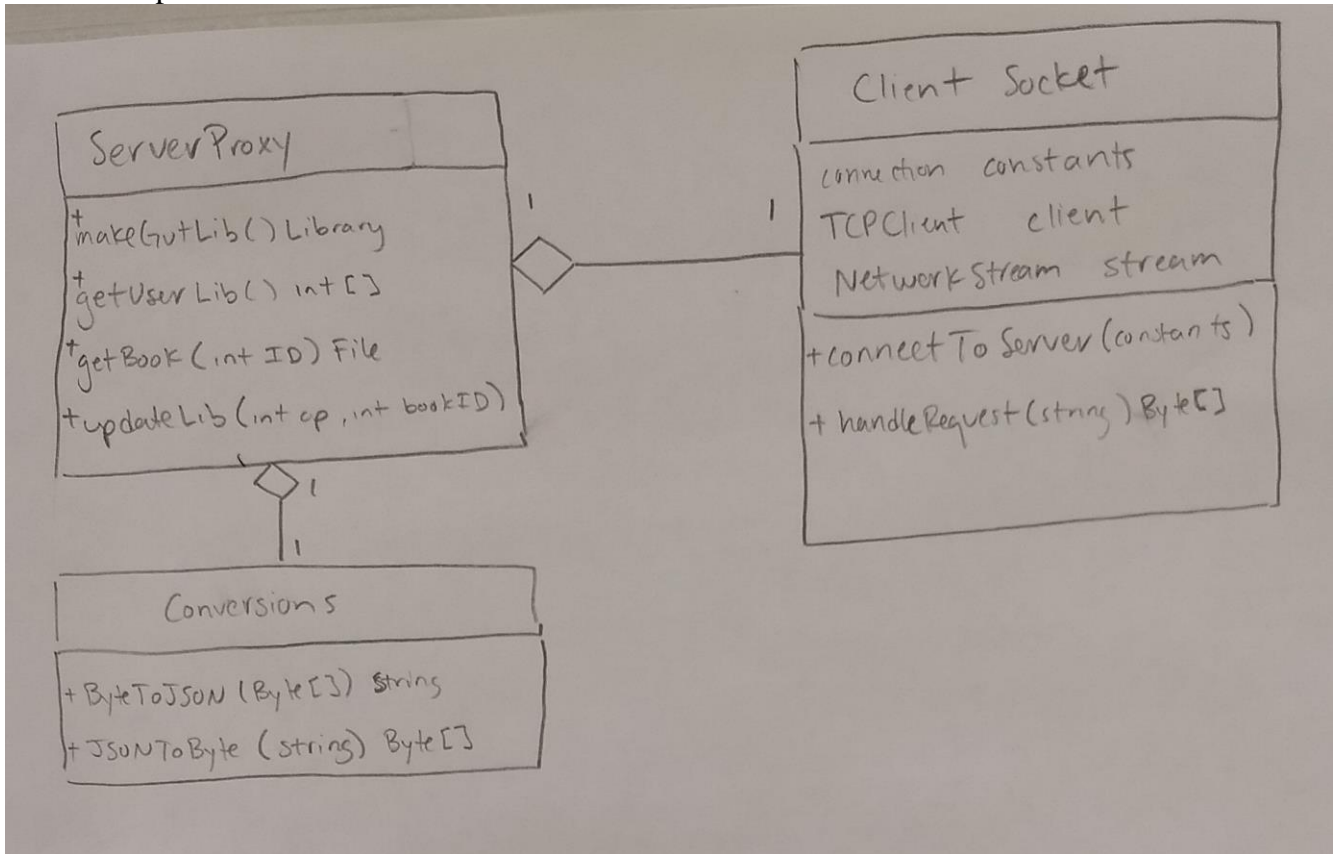The major system modules and their communications are as follows:

1) **User Interface:** communicates directly and only with the Server Proxy module. It uses the Server Proxy module to interact with the server. Sends requests, made by user or system, to Server Proxy and waits for response if required.

2) **Server Proxy:** abstracts server operations for the User Interface. Given a request from the User Interface, the Server Proxy will determine what kind of query needs to be made to the server. It will then charge the Client Socket module with the task of sending the query and also receiving any data which may be returned back to it, the Server Proxy.

3) **Client Socket:** makes connection to Server Socket and does all client-side data conversions needed in order to transfer data between Client and Server. The Client Socket will get queries from the Server Proxy to send over to the Server Socket, and also will, depending on the Server Proxy query, receive data from the server. All received data is also converted into its appropriate type once transfered through the sockets in the Client Socket module to be used by the Server Proxy.

4) **Server Socket:** listens for connections made by the Client Socket module and does all server-side data conversions in order to transfer data between Client and Server. The Server Socket will get queries from the Client Socket, and is simply responsible for reading them, doing the proper data conversion, and then passing them to the Business Logic module. If the query requires data to be sent back to the Client Socket the Server Socket also handles sending this data for the Business Logic module.

5) **Business Logic:** responsible for determining what kind of query the user has sent and what protocol needs to be followed. Gets client requests via the Server Socket module, and then it use the Database Driver module to communicate with the Database and perform CRUD operations as needed.

6) **Database Driver:** responsible for connecting Business Logic module and the MySQL database used by the system. Takes queries from the Business Logic implements them and then returns data to Business Logic if required.

7) **Database:** MySQL database which contains all system data. Communicates only with the Database Driver module.

Class Documents & Relationships:

Client side:

Book

- title: String
- author: String
- genre: String
- cover: Image
- id: int
- size: int
- currentPage: int

+ Book (bookFile: File)
+ getTitle (): String
+ getAuthor(): String
+ getGenre(): String
+ getCover(): Image
+ getID (): int
+ getSize(): int
+ getCurrentPage (): int
+ setCurrentPage(page: int): void

Library

- collection: List<Book>
- user: String

+ Library(user: String)
+ addBook(book: Book): void
+ removeBook(id: int): Boolean
+ getBook (id: int): Book
+ getSize(): int
+ sortByTitle (): void
+ sortByAuthor(): void
+ sortByGenre(): void
+ sortBySize (): void

Reader «static»

+ readBook (book: Book): void
+ bookmarkPage(page: int): void
+ setTextSize (size: int): void

GUI(s)

- Contains a collection of controls, e.g. buttons, labels, etc.

- Contains event handlers

Server Proxy

(see other page)

Server side pt. 1:

ServerProxy

+ makeGutLib() Library
+ getUser Lib() int[]
+ getBook (int ID) File
+ updateLib (int op, int bookID)

Client Socket

connection constants
TCPClient   client
NetworkStream   stream

+ connectToServer (constants)
+ handleRequest (string) Byte[]

Conversions

+ ByteToJSON (Byte[]) String
+ JSONToByte (string) Byte[]

Server side pt. 2:

Server Socket

connection constants

+ Main() void
+ StartServer() void
# Listen for Connections (l net.Listener) void
# Handle Connection (conn net.Conn) void
# SendData ([] byte) void

Conversions

ByteToJSON ([] byte) string
JSONToByte (string) [] byte

Business Logic

+ Execute Query (string)[] byte

DBDriver

DB connection constants

+ GetGutenLib() string
+ GetUser Lib() string
Get Book (ID int) string
+ Validate User (name string, password string) boolean
+ AddToUser Lib (ID int)
+ Del FromUser Lib (ID int)

Sequence Diagrams:

Reading a book:

Read a book:

User — Sign-in — Library — Reader

- Enter username + password → Validate
- [Success]
- Retrieve library (collection of books)
- Select book from library
- Read book
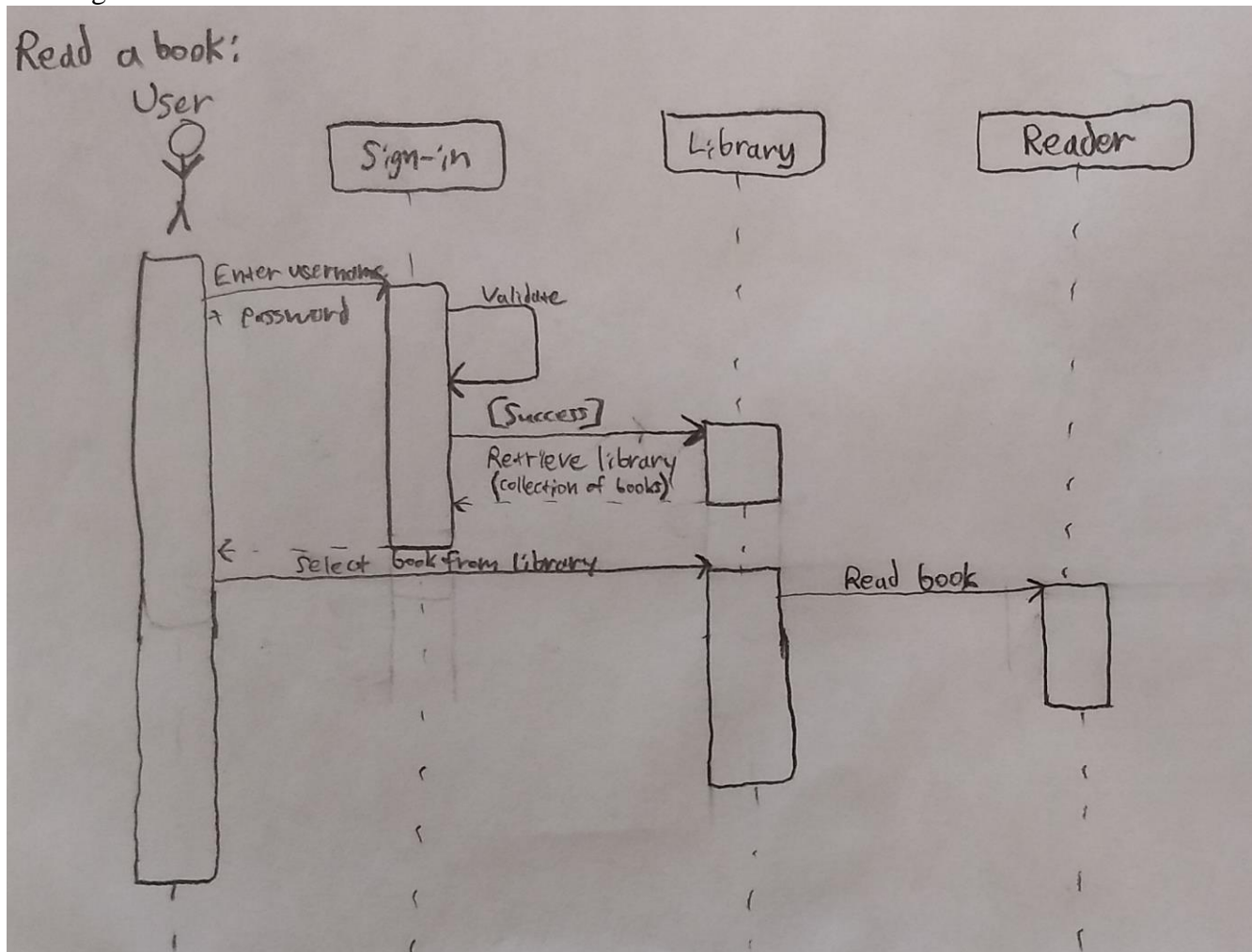
Finding a new book:

Find a new book:

User | Sign-in | User Library | Gutenberg Library

Enter username + password → Validate

[Success]

Retrieve library

Browse Gutenberg library → Sort

download book
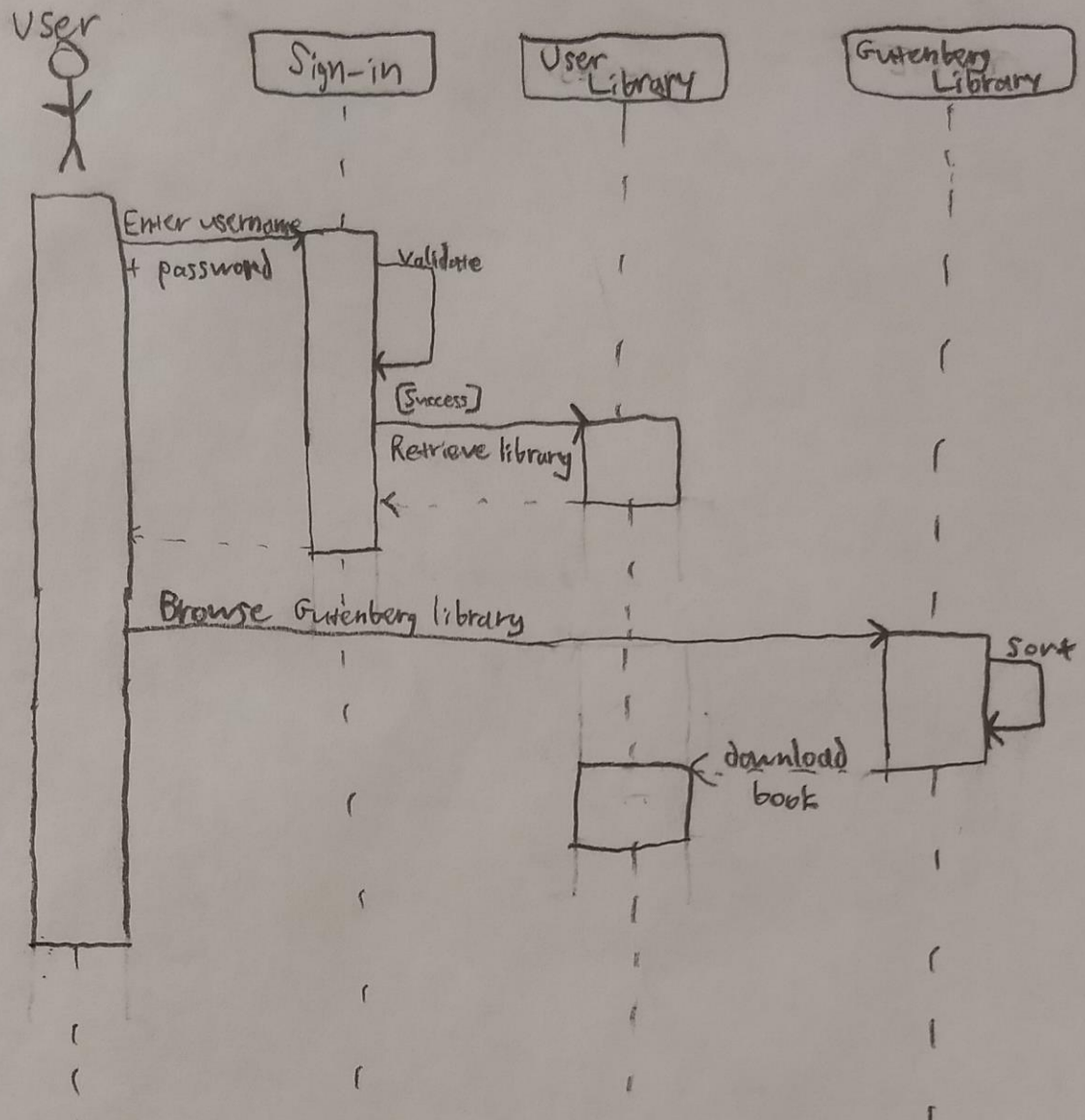
Server side sequence diagram: