**Automated digital and**

**distributed system for**

**managing the activity of a**

**library**

Written by:  
Lupu Ionuț  
Barbălată George Mihai

Barbu Andy Valentin

Date:25/3/2021

Table of contents:

1. **Introduction 1** 
   1. Purpose 2
   2. Scope 2
   3. Overview 2
   4. Reference Material 3
   5. Definitions and Acronyms 3
2. **System Overview 3**
3. **System Architecture 4**

3.1 Class diagram 4

3.2 Use case diagram for book actions 5

3.3 High level web app design 6

3.4 Use case diagrams for member activity 7

3.5 Database relational diagram 9

**1. INTRODUCTION**

**1.1 Purpose**

The purpose of this project is to help the day to day activities of a library by giving keeping track

of the book stock, keeping track of the books that have been borrowed, and by allowing readers

to see what books are currently available at what locations and reserving them for later

borrowing.

**1.2 Scope**

The SDD shows how the software system will be structured to satisfy the requirements identified in the software requirements specification document. It is a translation of requirements into a description of the software structure, software components, interfaces, and data necessary for the implementation phase. In essence, the SDD becomes a detailed blueprint for the implementation activity. In a complete SDD, each requirement must be traceable to one or more design entities.

**1.3 Overview**

This document consists of eight clauses. The first clause defines the scope of the recommended practice and the second clause references the general description of the functionality, context and design. The third clause provides decomposition of the subsystems in the architectural design and design rationale. The fourth clause places the SDD into the data transformation phases. Within the fifth clause we take a closer look at what each component does in a more systematic way and the sixth clause describes the functionality of the system from the user’s perspective. The eight clause provides appendices, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.

**1.4 Reference Material**

IEEE Software Engineering Standards Committee, ‘IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications’

**1.5 Definitions and Acronyms**

SDD – Software Design Document

SRS – Software Requirements Specification

HIghLevelUsers – User of the application that is not a basic reader. Library staff and System administrators belong in this category.

**2. SYSTEM OVERVIEW**

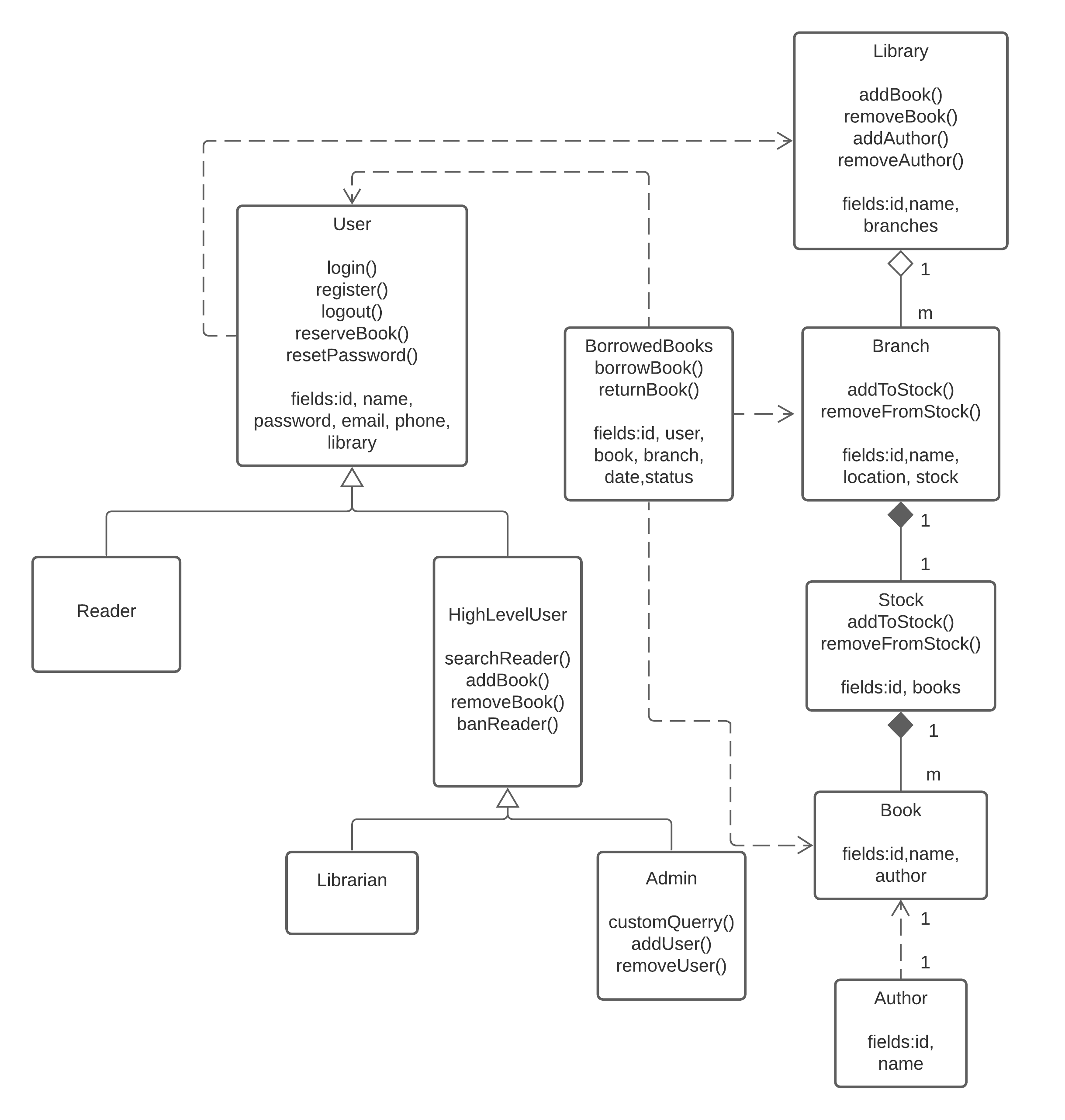
The system will consist of a web application that will manage the activity of a library, while also providing an online interface for it.

The project will have a database for storing information about clients, books and other information that needs to be stored.

The main scope of this product is to minimize the manual work done by the library workers by automating the inventory of the library and automating the borrowing process. The readers are also able to search for the desired books and to receive the respecting books without being necessary to be physically in a library.

**3. System Architecture**

3.1 Class diagram

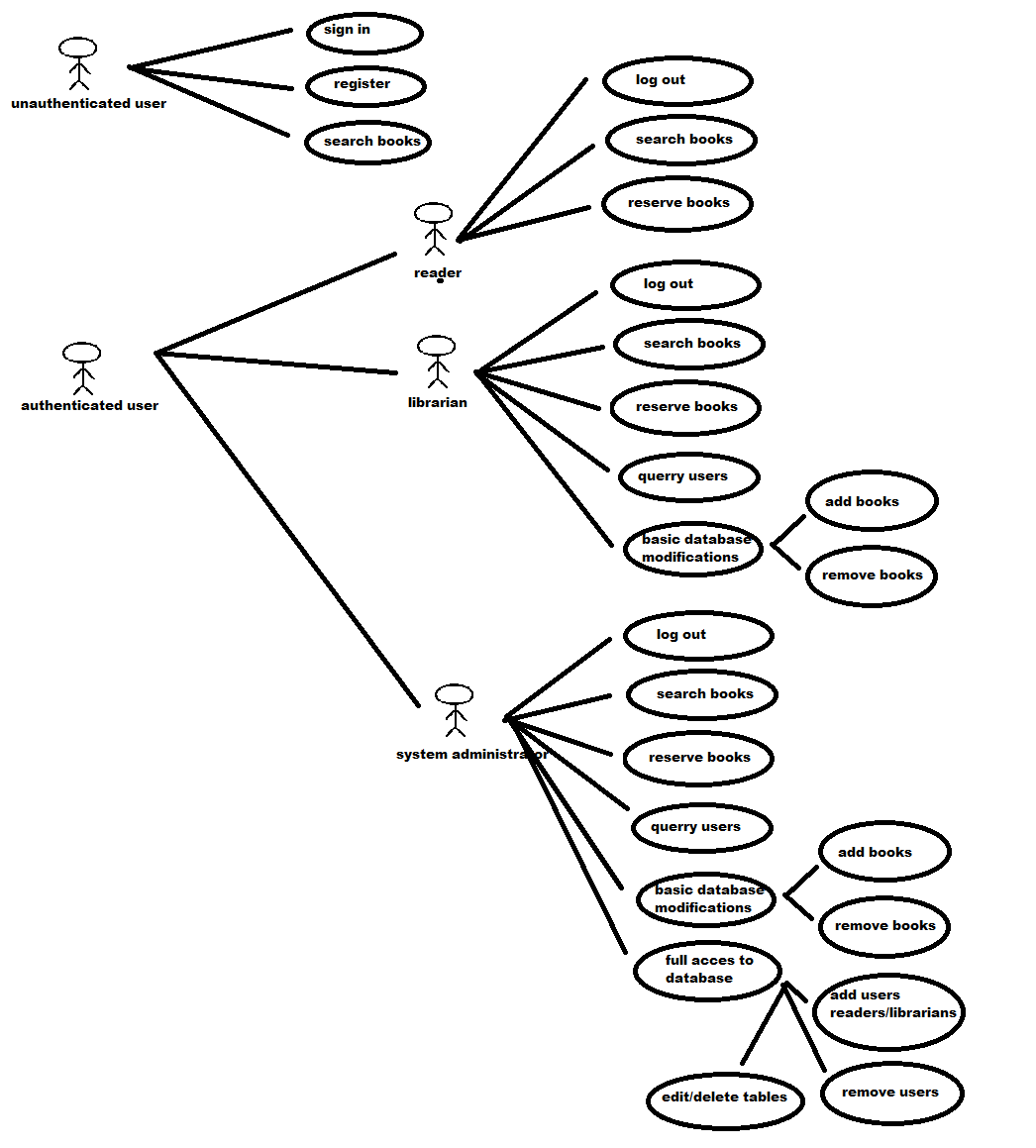
****

For this application, there are 3 types of users: Reader, Librarian and System administrator. The reader has only the basics functions at his disposition, while Librarians and System administrators, from here on being referred as HighLevelUsers, have more functions.

The hierarchy is done in such a way that each class inherits the lower classes functions, while also gaining new functions (the System administrators can also add books).

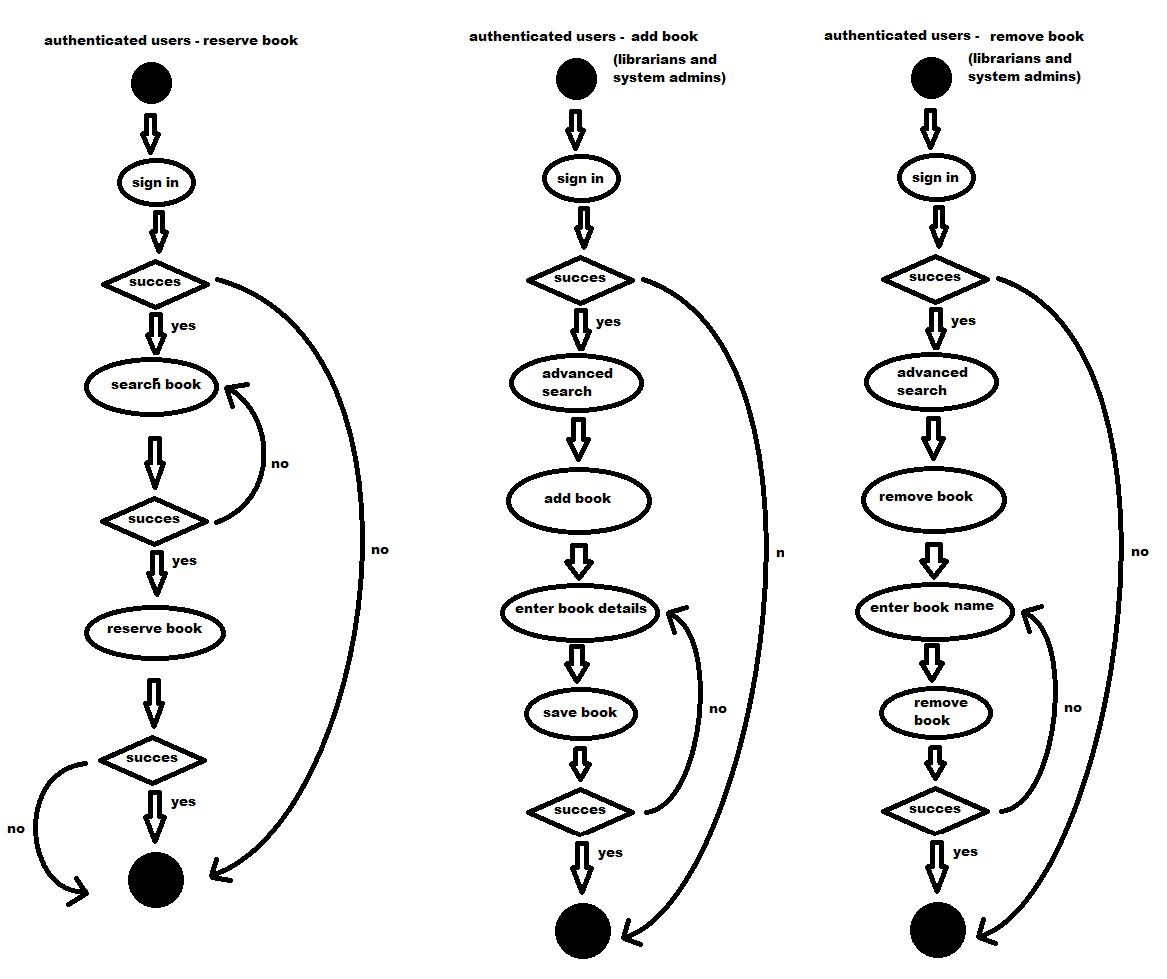
The library has an array of branches, which in turn, each branch has a stock containing the books and the quantities. A book has an author.   
The user also has a dependency on the library. This way, the user is known on which library it belongs to, an also the HighLevelUsers have a direct access to the library.

3.4. Use case diagrams for member activity  
In this diagram, the full functionalities are presented for each type of user, even unauthenticated users.

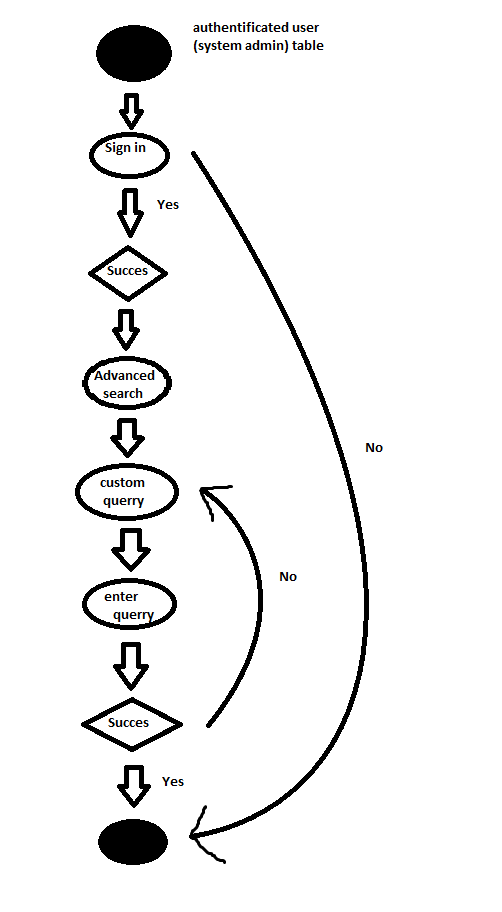


3.2 Use case diagram for book actions

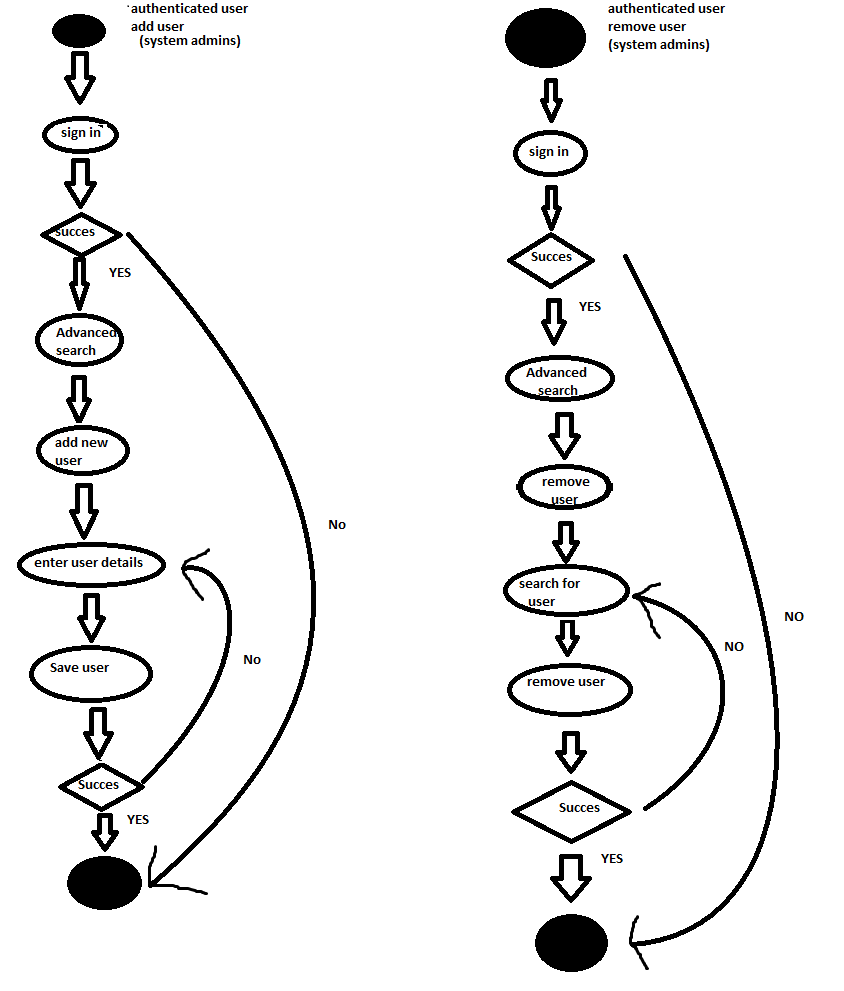
To reserve, add or remove books, the steps presented below must be followed. It can be observed that only HIghLevelUsers are able to add and remove books.

****

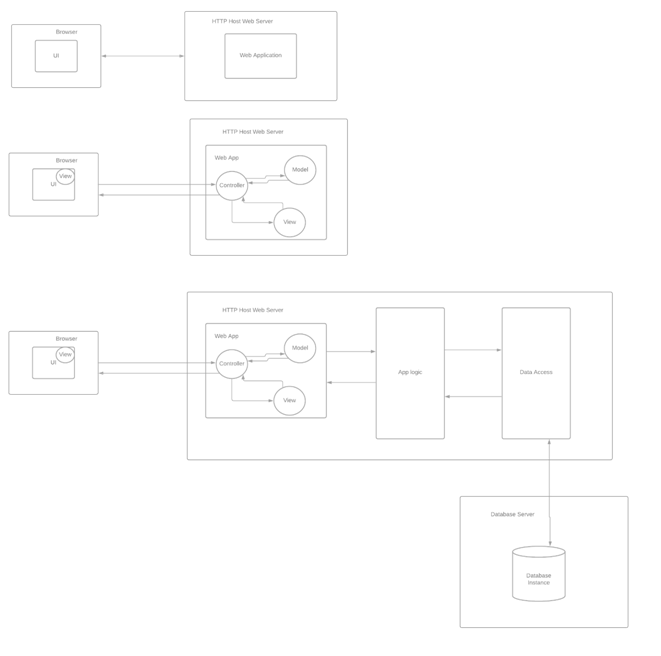
The Custom query function is the most powerful tool in this application, being available only to system administrator. “Hidden” behind the “advanced search” tab, the system administrator can write any query in the designated text box and the system will execute it. It has the potential to break the database, so it must be used carefully. It requires some SQL skills, as no predetermined advanced queries are implemented.



The add and remove users function can be done only by system administrators with a preset query function. The steps that must be followed are presented below:

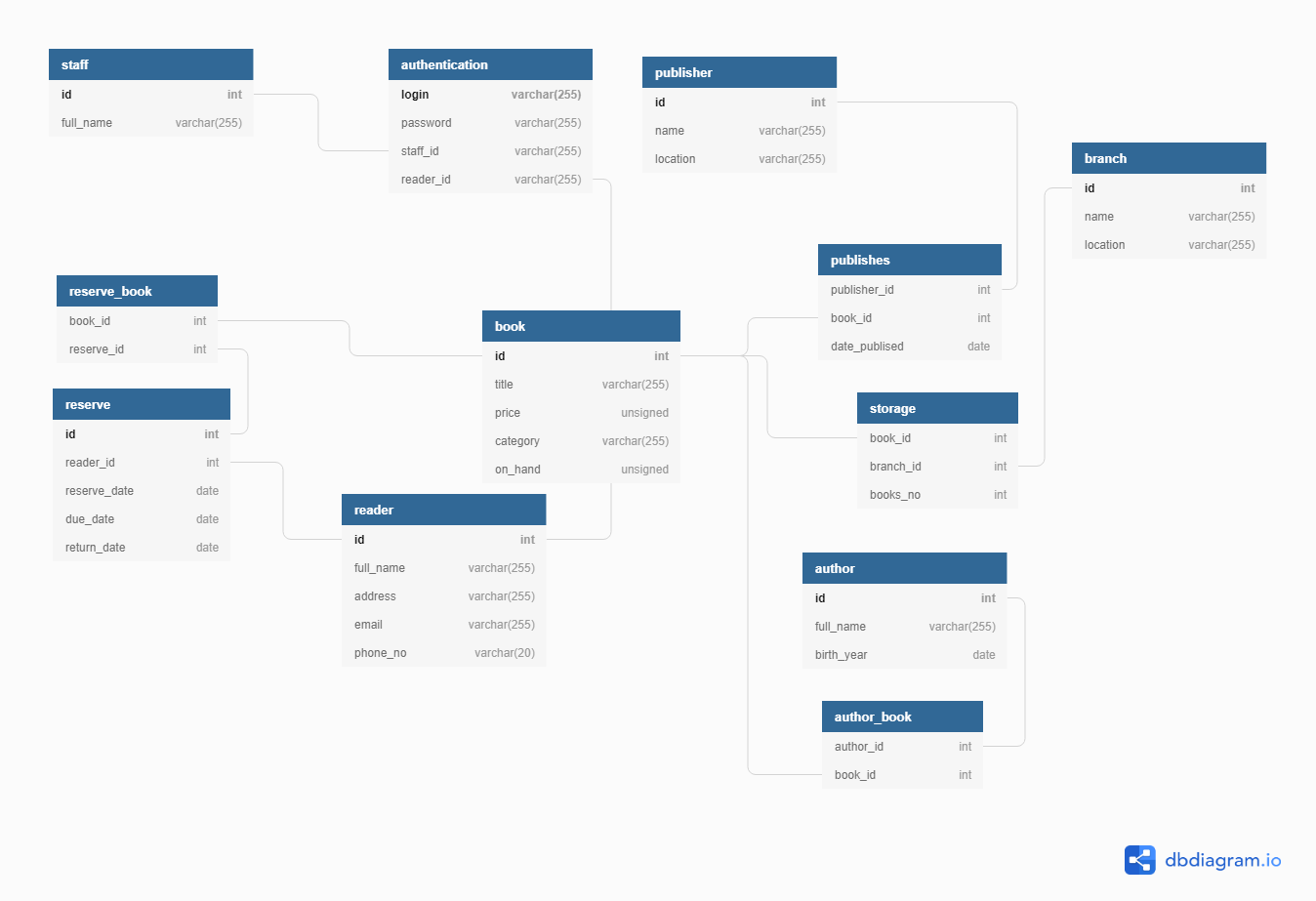


3.3 High level web app design

As the application is done with .Net core MVC, the basic Model View Controller schematic is closely followed.

3.5 Database relational diagram

The database that the application runs is presented below. As the software that was used to create this diagram limits the elements that can be used during a free trial, only the primary keys, foreign keys and the relationships has been added. Other information, such as name, email, phone number etc… are not presented below, but will be included in the application.



Presented below is the high level design. It can be seen that the Presentation and the Application Logic are interconnected, while the Data Access is connected to the data base. As the authentication status will affect the Presentation layer, these two will be closely linked. For the authentication to take place, it must access the database.

