<Project Name>
Software Design Description

1. Revision History

Version	*Date*	*Author*	*Change Description*
3	10.04.2017	M.İkbal KILAVUZ	Adjustments to the database
3.1	17.04.2017	Enes Karabulut	Minor changes in the interface
3.14	24.04.2017	Simge Acımış	Test and fix

2. INTRODUCTION

2.1 Purpose and Scope

The purpose of this software is to present the library book loan system. Users will be able to benefit from books without paying reliably. The first aim of the software is to serve the maximum number of users in a regular and sequential manner. The service is also trying to provide reliable, user friendly and fast software.

2.2 Document Overview

The next chapter of the document has described architectural design of the Library Loan System. The high level components and their interactions, suitable architectural patterns, physical arrangement of components and design decisions applied to the whole system. Includes design patterns, sequence diagrams, class diagrams, database design and user interface design with screen shots of the interfaces.

2.3 System Overview

If the user is not registered to the book rental system, he / she will first register from the registration page. If the user is registered, he logs in to the system with the user name and password from the login page. After logging in to the system, there is a page where the user is facing the menus. This page also has functions such as book search, profile lookup, author search, checkout. The user searches the book he wants to rent from the system. If the book is available in the system, ie not rented by other users. A contract is signed indicating that the return date is shown and that the user agrees to accept the future if the user does not return the book. If the user wants to return the book earlier, he can change the date of the return. After accepting the contract, the book is allocated to the user.

2.4 Definitions, Acronyms, and Abbreviations

All non-standard terms, acronyms and abbreviations that are unique to this document should be included in this section. References to other appendices or reference documents may be included. Index information should be in bold.

- IEEE – Institute of Electrical and Electronics Engineers
- SRS – System Requirements Specification

Also see IEEE Std 1002-1987, IEEE Standard Taxonomy for Software Engineering Standards.

Term/Acronym	***Definition***
Algorithm Design	Specific method to create a mathematical process in solving problems
Architectural Design	Establishing the overall structure of software system
Compatible	Capable of orderly efficient integration and operation with other elements in a System with no modification
Database	A collection of stored related data
ER diagram	Entity Relationship Diagram; Data model for describing a data base in an abstract way
Sequence Diagram	An interaction diagram that shows how process interact with one another and in what order
SDD	Software Design Document
SDS	System Design Specification

2.5 References

[Provide references for any pertinent document including reference number, version number, release date, and version name.]

2.6 Design Constraints and Decisions

The limited number of database fields means that the number of users can not be reached in high numbers. Since the Android operating system bringing out a new version every year, there are lots of Android versions. Because of our system need for this operating system, version differences make our process difficult. We have set our Android version to 4.0 as the lowest. The reason why we choose this version is that it is the most used version in Turkey.

Decisions:

Object oriented software development methods

- **Improved software maintainability.**
- **Faster development**
- **Lower cost development**
- **Improved software development productivity**
- **Higher quality software**

MVC Architectural Pattern

- **It should interact with other machines or users effectively.**
- **For more efficient interaction system should have flexible interfaces.**
- **MVC can be taken as for a popular and easy to handle web application development style that has the feature of separating the presentation, Business & intermediate logics.**
- **Ease to coding and provide well defined interfaces within each logic.**

Three-Tier Client Server Architecture

- **As more users access the system a three-tier solution is more scalable than the other solutions because you can add as many middle tiers as needed to ensure good performance.**
- **Security is also the best in the three-tier architecture because the middle layer protects the database tier.**

3. Design Details

Login:

1. Get Username and Password

2. If user name is equal to the entered Username & the password is equal to the entered Password

3. Then login successful

4. Else login failed

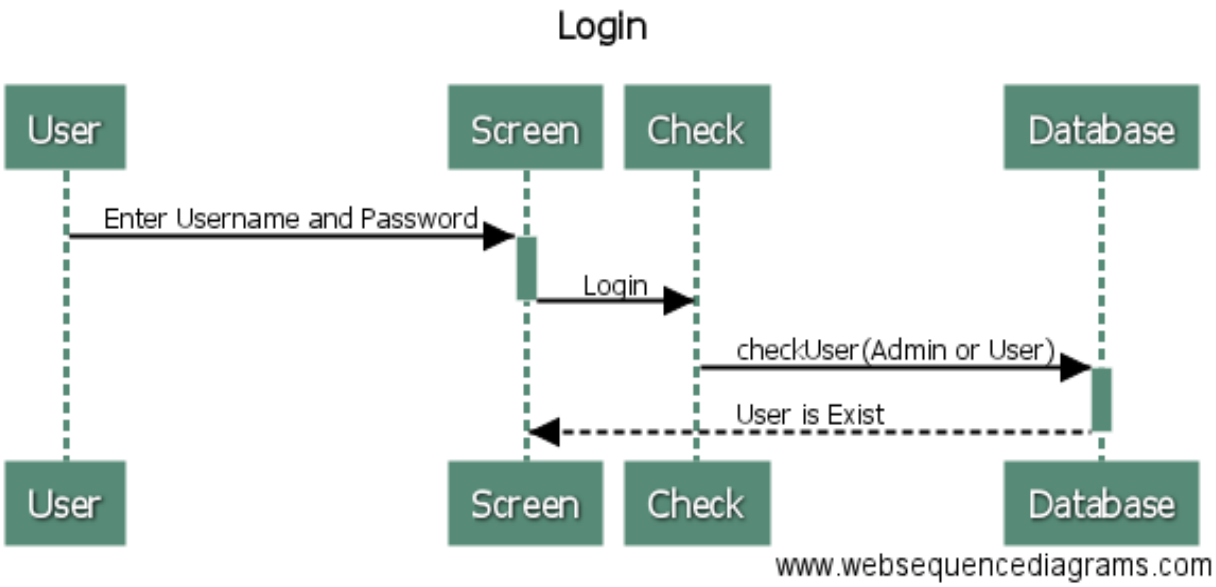
5. End If.

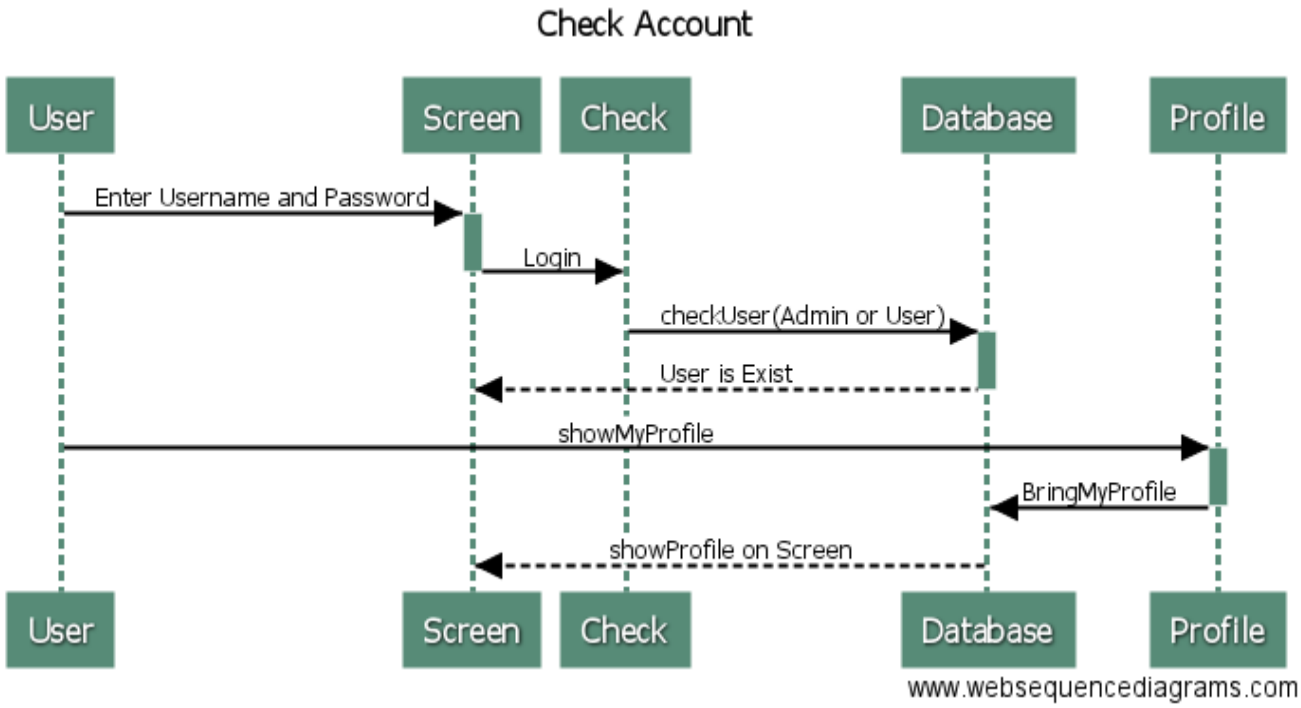
3.1 Software Components

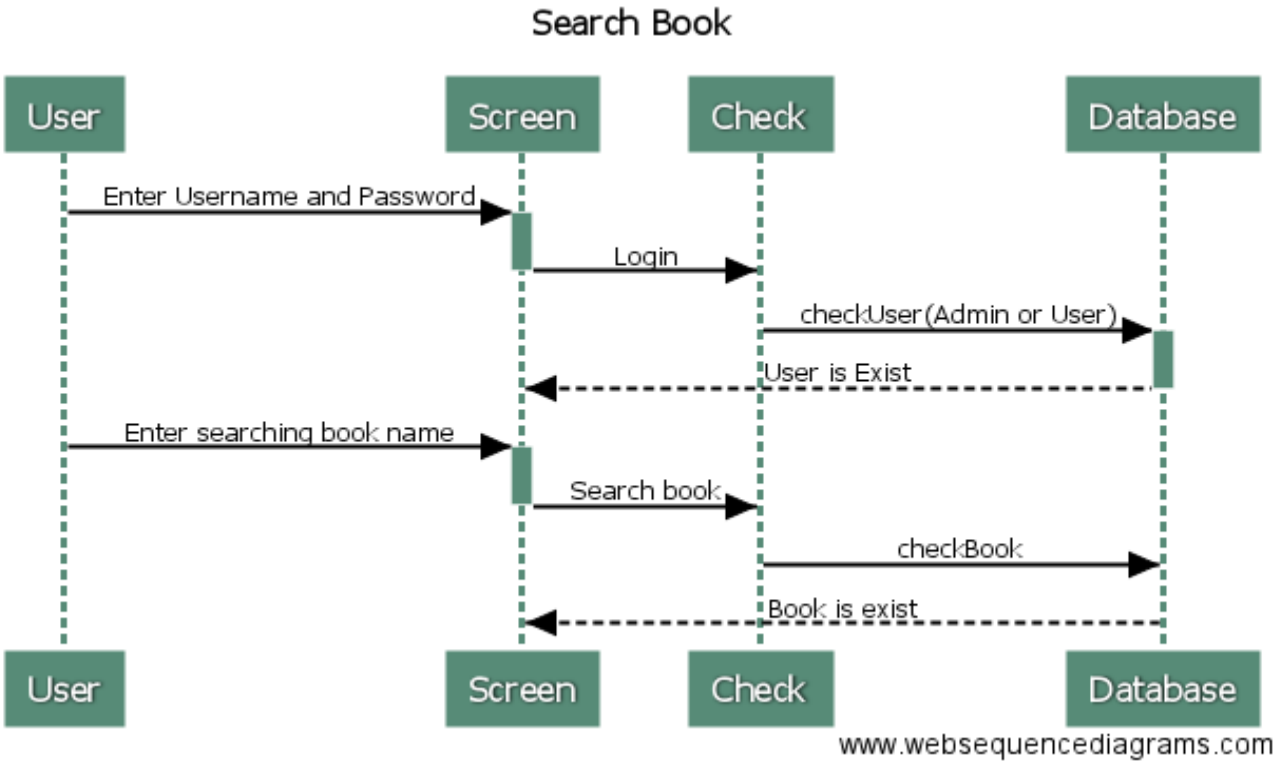
[Draw the class diagram for the system. Each class should have its list of attributes and methods. Note that details on the class diagram should match with the function names shown on the sequence diagrams (as discussed in class).]

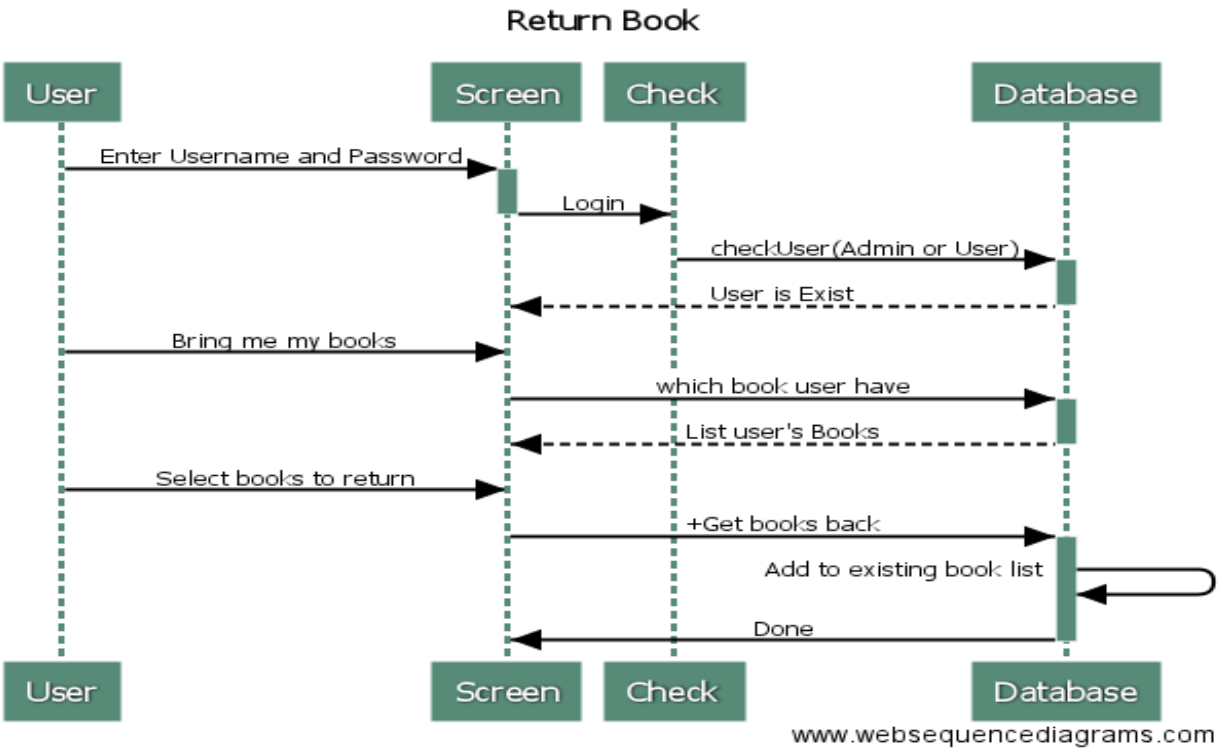
3.2 Software Behavior

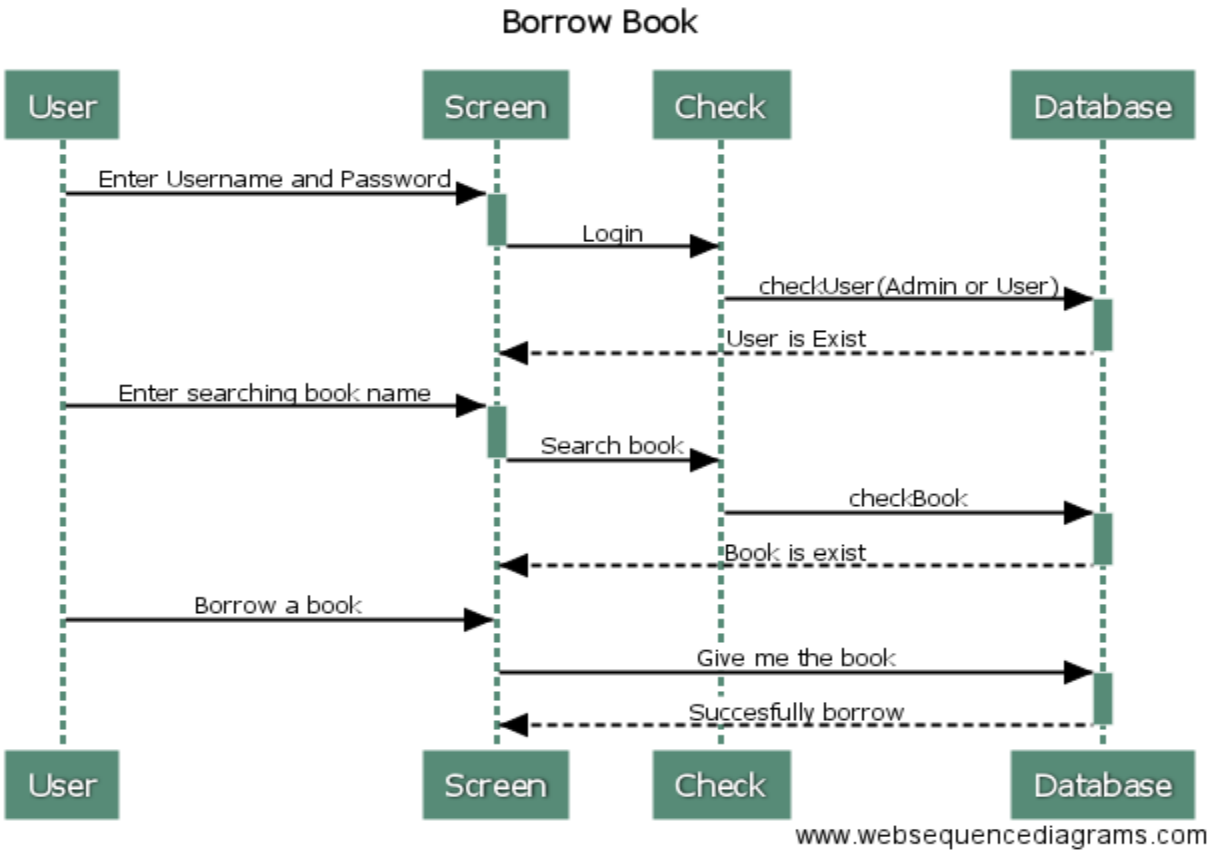
Sequence Diagrams:

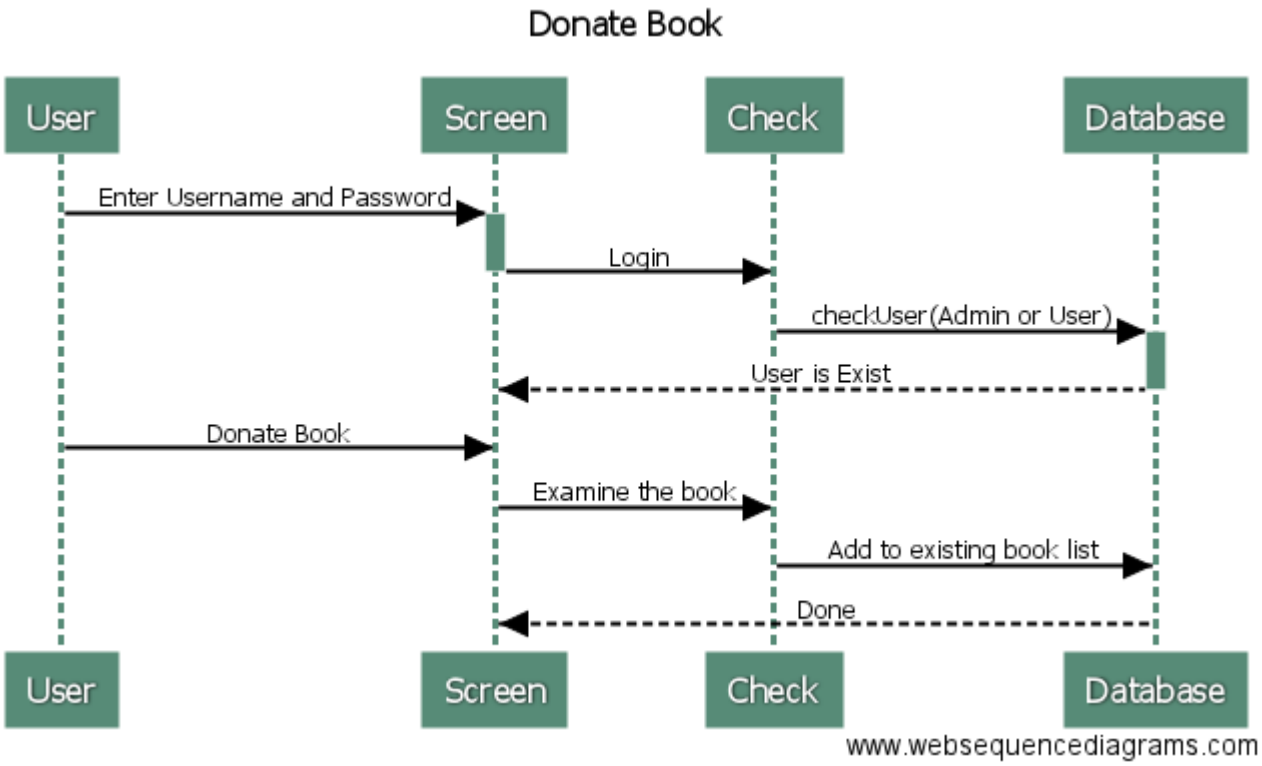


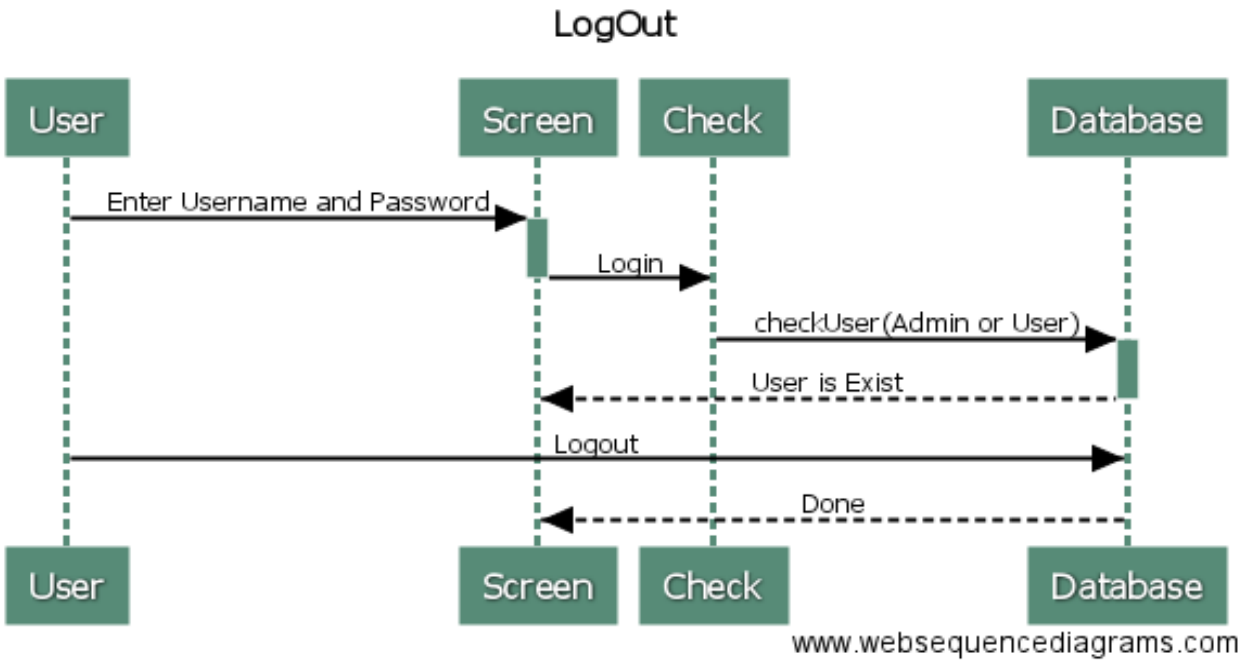

Check Account
User
Screen
Check
Database
Profile
Enter Username and Password
Login
checkUser(Admin or User)
User is Exist
showMyProfile
BringMyProfile
showProfile on Screen
User
Screen
Check
Database
Profile
www.websequencediagrams.com


Search Book
User
Screen
Check
Database
Enter Username and Password
Login
checkUser(Admin or User)
User is Exist
Enter searching book name
Search book
checkBook
Book is exist
User
Screen
Check
Database
www.websequencediagrams.com

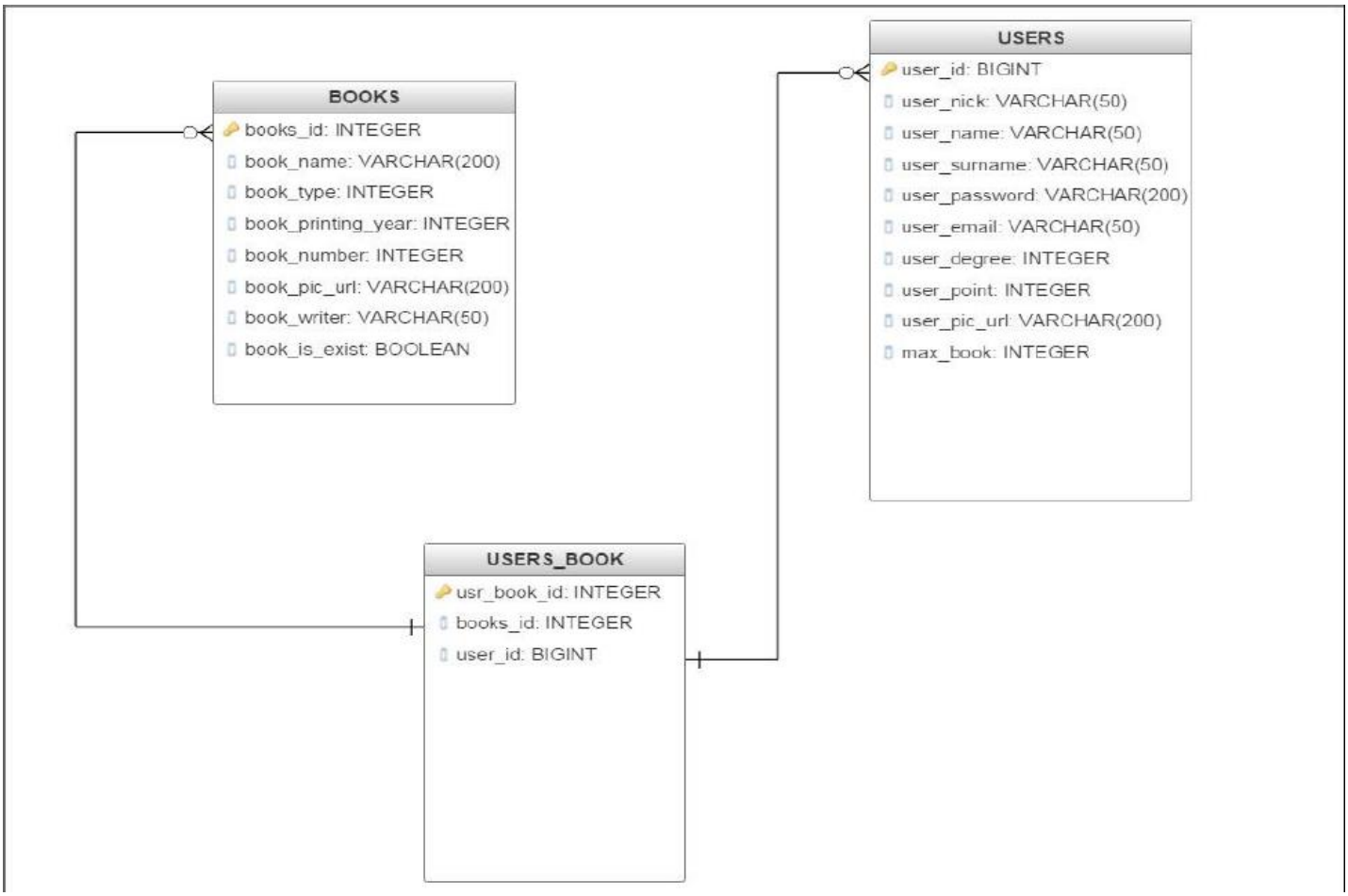

Return Book
User
Screen
Check
Database
Enter Username and Password
Login
checkUser(Admin or User)
User is Exist
Bring me my books
which book user have
List user's Books
Select books to return
+Get books back
Add to existing book list
Done
User
Screen
Check
Database
www.websequencediagrams.com


Borrow Book
User
Screen
Check
Database
Enter Username and Password
Login
checkUser(Admin or User)
User is Exist
Enter searching book name
Search book
checkBook
Book is exist
Borrow a book
Give me the book
Succesfully borrow
User
Screen
Check
Database
www.websequencediagrams.com


Donate Book
User
Screen
Check
Database
Enter Username and Password
Login
checkUser(Admin or User)
User is Exist
Donate Book
Examine the book
Add to existing book list
Done
User
Screen
Check
Database
www.websequencediagrams.com


LogOut
User
Screen
Check
Database
Enter Username and Password
Login
checkUser(Admin or User)
User is Exist
Logout
Done
User
Screen
Check
Database
www.websequencediagrams.com

3.3 Data Model (E-R Diagram)



3.4 User Interface Design

[Describe interfaces with the various system users. Even if this information could already be described in previous related documents (i.e. SRS or analysis documents), in this section the description of this interfaces should be more specifically done (Visibility of system status, User control and freedom, Error prevention, Aesthetic and minimalist design etc.)]

4. Requirements Traceability

[Establish traceability relationship between requirements and software design to show that all requirements are met. Demonstrate relationship between classes and requirements using traceability matrix.]

5. Annexes

[If some parts of the document need to be explained in more detail, create a separate document and annotate it in this section.]