HACETTEPE UNIVERSITY DEPARTMENT OF COMPUTER ENGINEERING
BBM487: SOFTWARE ENGINEERING LABORATORY

SOFTWARE DESIGN DOCUMENT

GROUP - 6

CEMAL ÜNAL   **21328538**

İREM KOCABAŞ   **21328188**

METİN ARSLANTÜRK   **21426625**

<Library Book Loan System>
Software Design Description

1. Revision History

Version	Date	Author	Change Description
1.0	11.03.2017	İrem Kocabaş	Logout added
2.0	21.03.2017	Metin Arslantürk	Architecture changed
3.0	08.04.2017	Cemal Ünal	UI improved
4.0	21.04.2017	İrem Kocabaş	Book searching area changed

2. INTRODUCTION

2.1 Purpose and Scope

This document includes design,user interfaces' detailed descriptions in our project which has completed abstract levels and design.Also it shows; how we made it corresponding to requirements and how many requirements we ignored.
Also it has entity relationship diagram.Also it describes the system more clearly with sequence diagrams.

2.2 Document Overview

In this document ;
- 2nd section there is general descriptions and acronyms.
- In 3rd section there are descriptions of design decisions.
- In 4.1; properties of system and class diagram.
- In 4.2;Components of software and sequence diagram
- In 4.3; Architectural approach of our system.
- In 4.4; User interfaces and briefly descriptions.
- In 5 ; How much requirement we made it real and how much requirement we ignored which we determined at the start of project.
- In 6; There are annexes corresponding the necessaries.

2.3 System Overview

Our system has Turkish user interface so with the librarians who knows Turkish, it has many usable features which a library system may needs.

Our system installed on MVC architecture and it has book adding,updating,removing (CRUD) operations.
It has extra error checking system and we can say it has maintenance feature so we can update and improve it easily.

2.4 Definitions, Acronyms, and Abbreviations

Term/Acronym	Definition
LBLS	Library Book Loan System
Pr.Mn.	Project Managers
Bk.Sts	Book Status

2.5 References

Reference No	Version	Version Date	Version Information
1	1.0	11.03.2017	Vision and Plan of Project
2	2.0	21.03.2017	Requirement Document of Project
3	4.0	21.04.2017	Architecture and Test Cases of Project

Also report templates which given by TAs and course notes which given by lecturers.

3. Design Constraints and Decisions

There was not much constraint from outside, which influenced our design, because the given project was freed to a large extent by our favor. In addition, we decided our own design boundaries so that we could produce our own project.
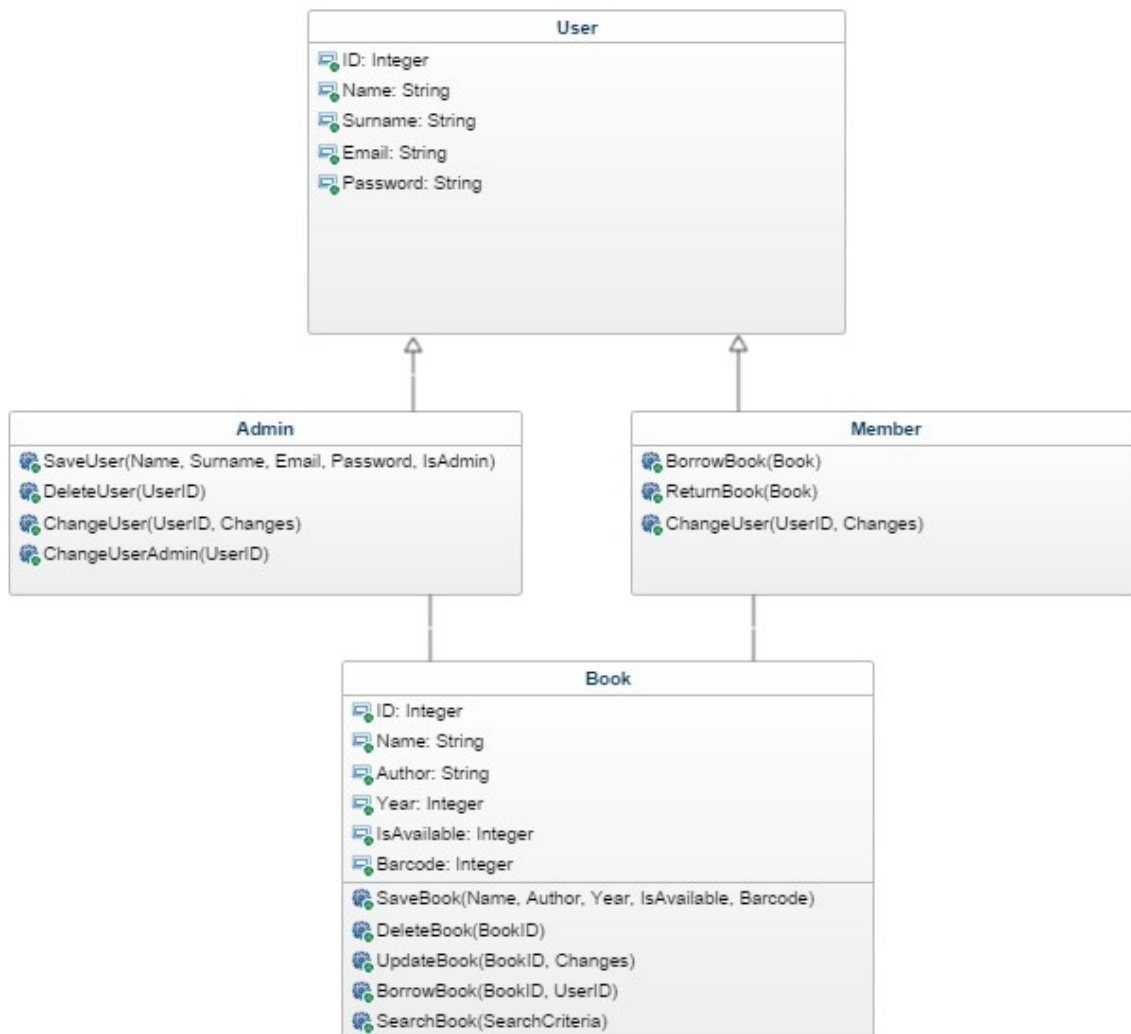
The decisions are as follows:

- Our application is web-based. There should be internet connection.
- We should use MVC architecture while developing our project.
- Each button in the system interfaces must be responsible for performing just one operation.
- System should be flexible.
- The system should give an error message in case of incorrect entry of the password or unable to find the record in the entered information.
- All user interfaces in the system should be understandable, easy to use.
- The system must have the necessary infrastructure to prevent data loss.

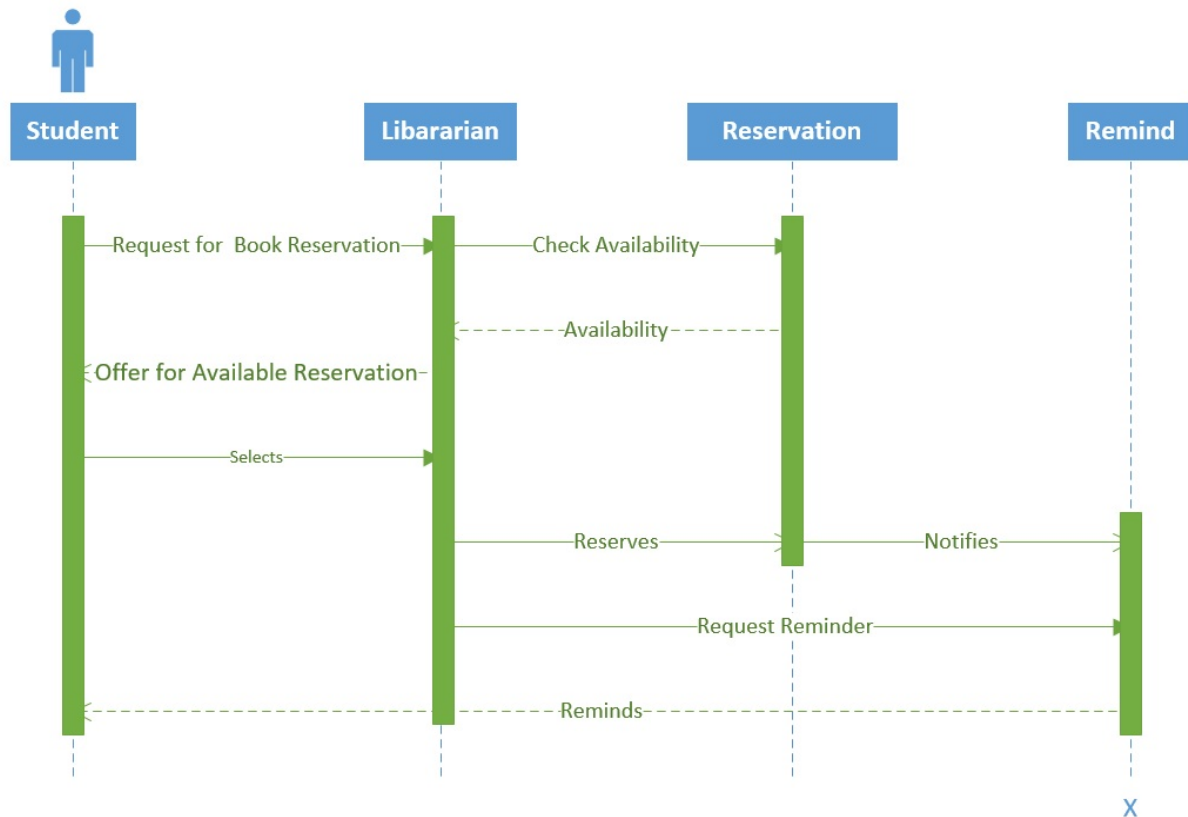## 4. Design Details

4.1 Software Components

Since language of our project language is PHP, we did not work object-oriented, so we do not have certain classes. But we showed general structure of our application in the diagram below.

User
- ID: Integer
- Name: String
- Surname: String
- Email: String
- Password: String

Admin
- SaveUser(Name, Surname, Email, Password, IsAdmin)
- DeleteUser(UserID)
- ChangeUser(UserID, Changes)
- ChangeUserAdmin(UserID)

Member
- BorrowBook(Book)
- ReturnBook(Book)
- ChangeUser(UserID, Changes)

Book
- ID: Integer
- Name: String
- Author: String
- Year: Integer
- IsAvailable: Integer
- Barcode: Integer
- SaveBook(Name, Author, Year, IsAvailable, Barcode)
- DeleteBook(BookID)
- UpdateBook(BookID, Changes)
- BorrowBook(BookID, UserID)
- SearchBook(SearchCriteria)

4.2   Software Behavior

Add name to book waiting list use case:

Login use case:
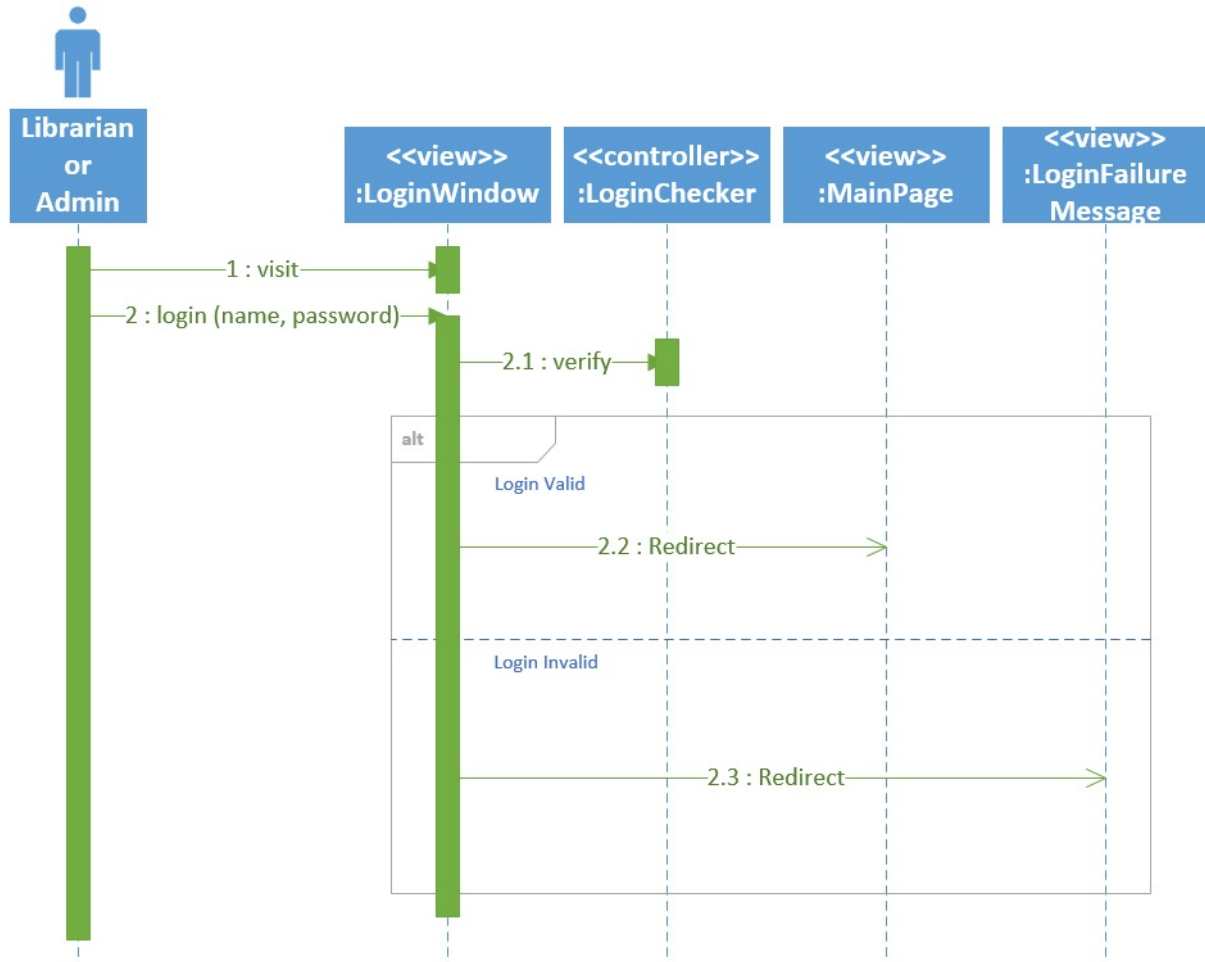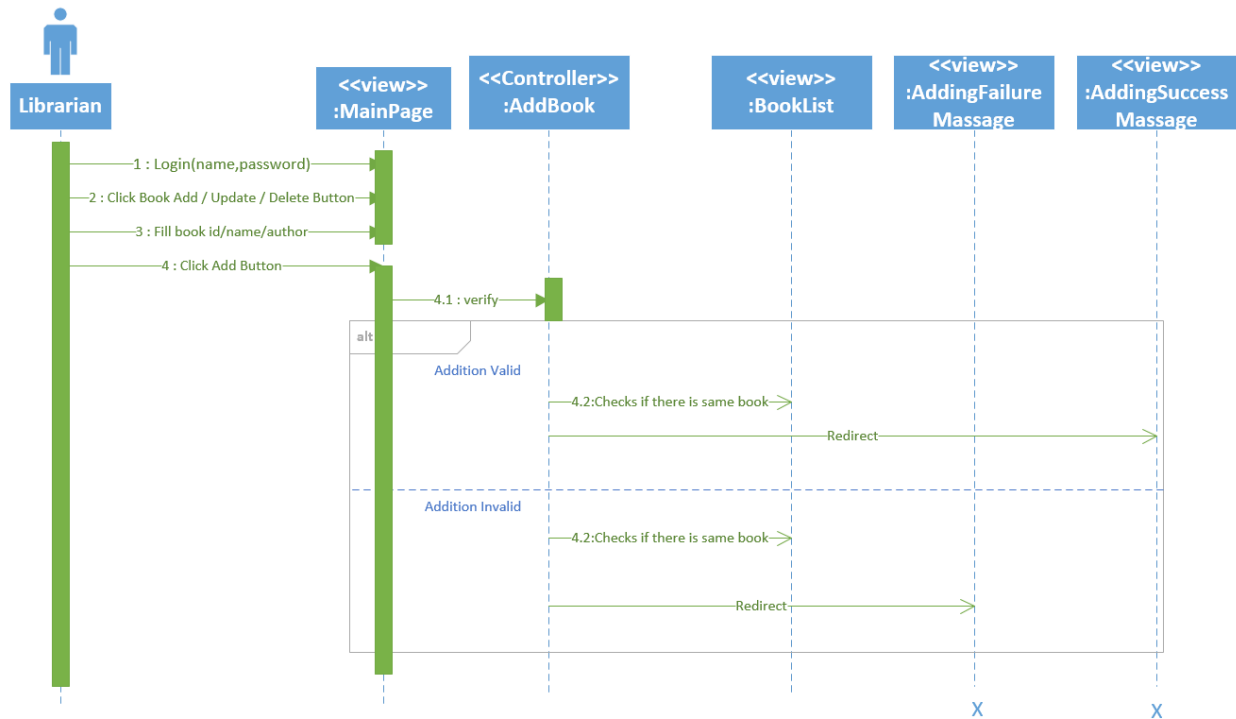
Add book use case:

Delete book use case:
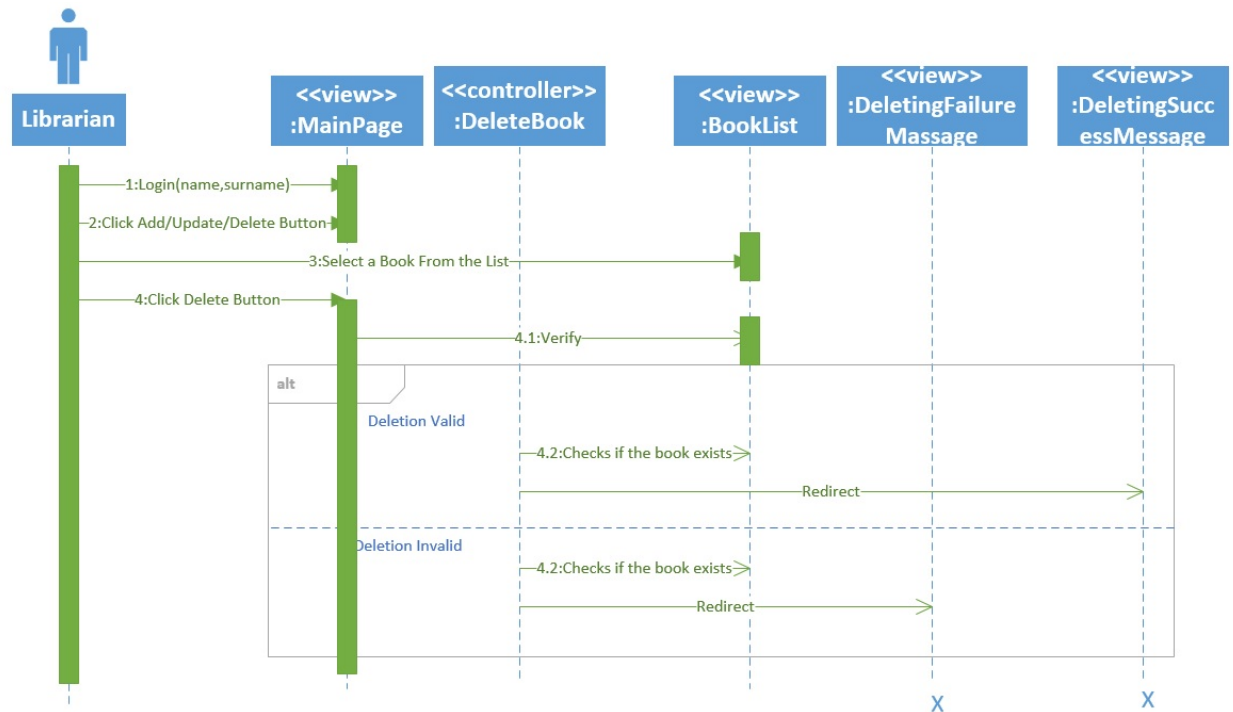
Update book use case:
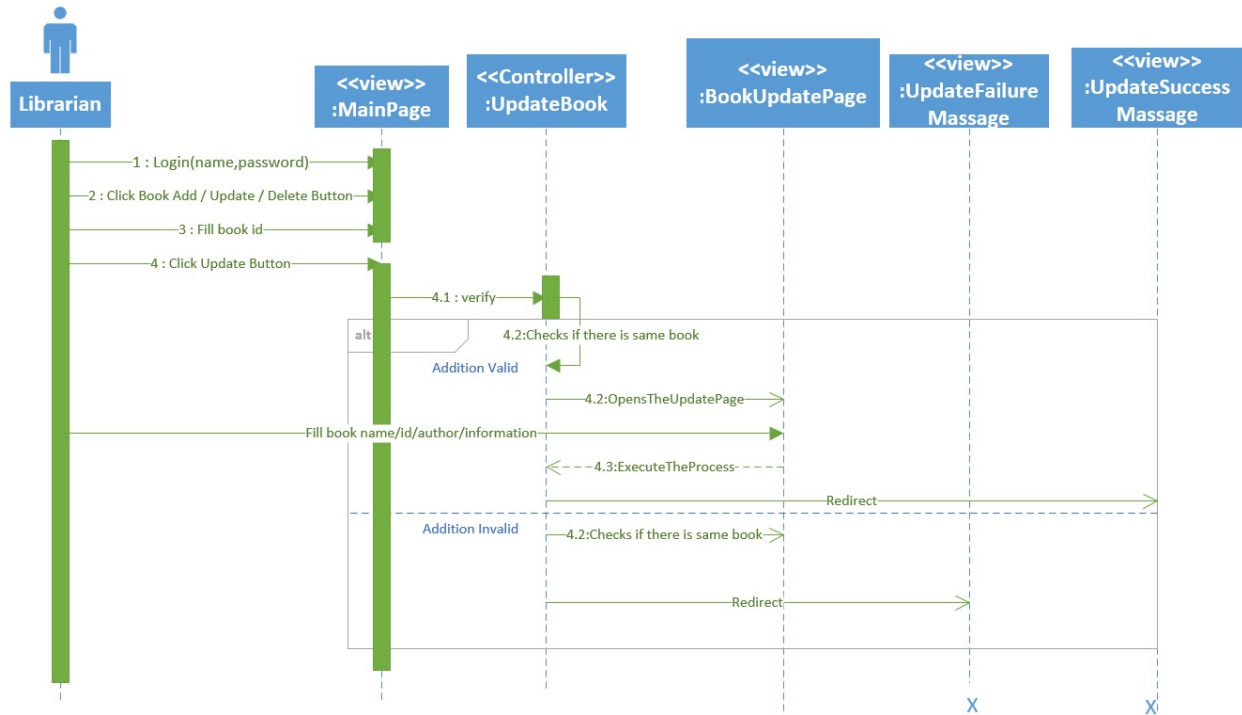
Add member use case:
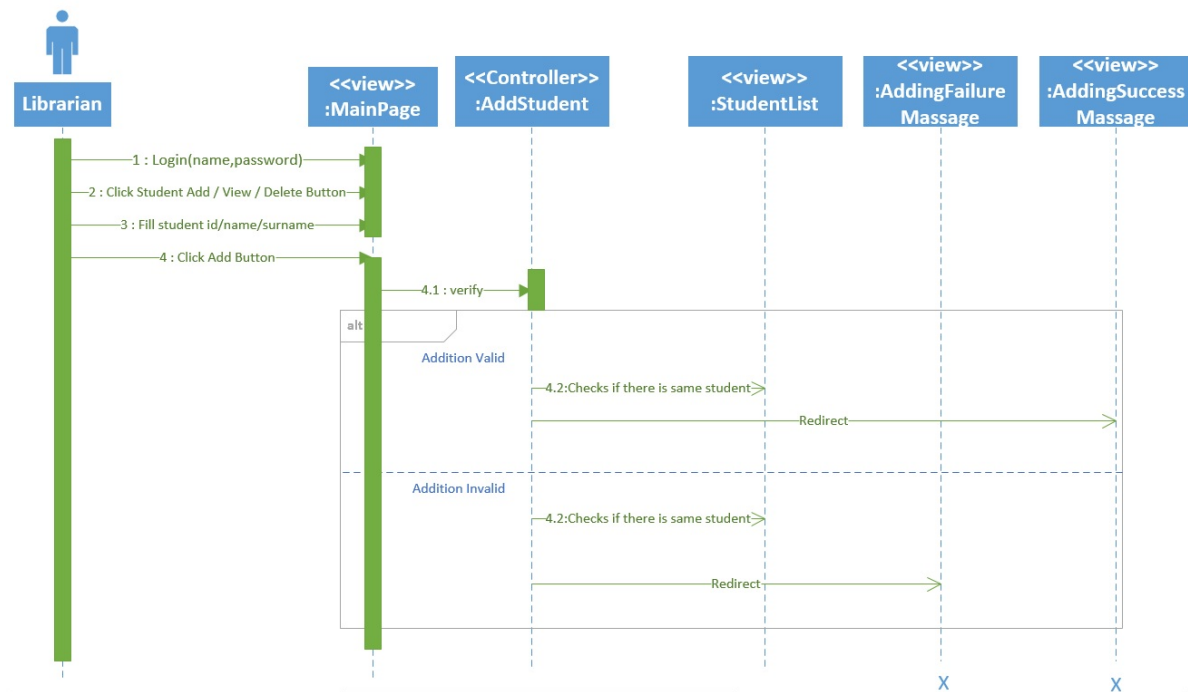
Delete member use case:

Update Member:
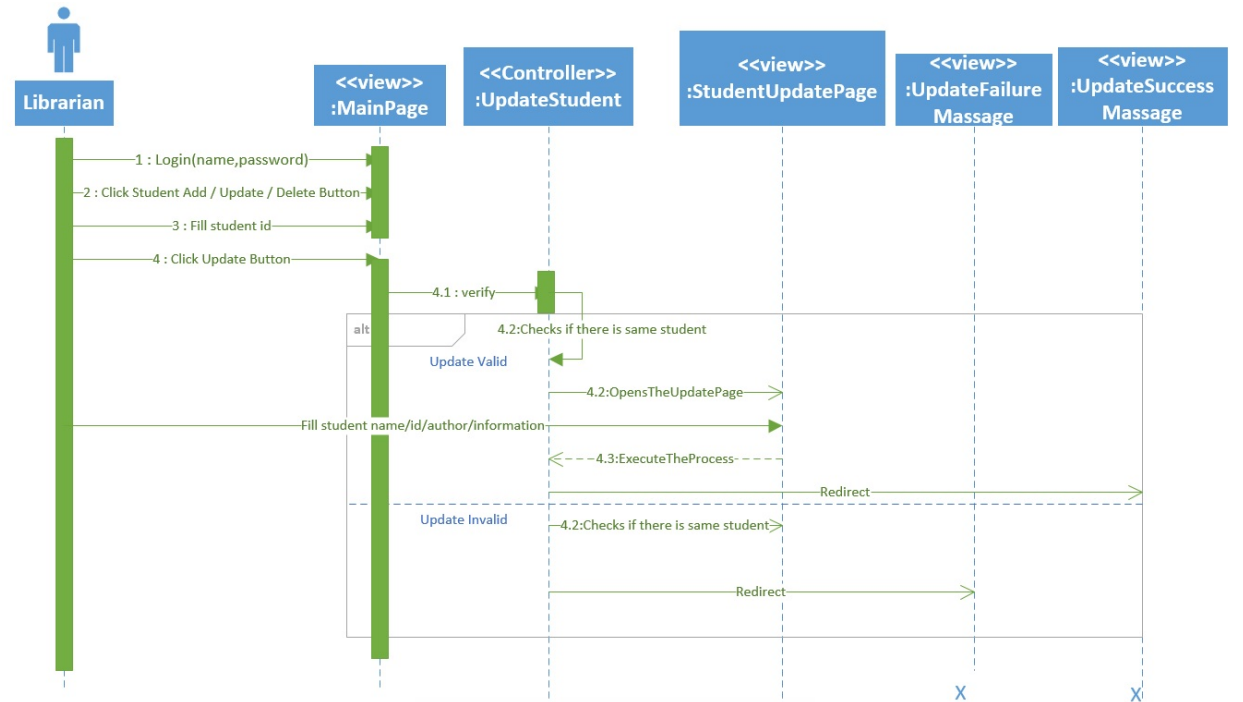
View member use case:

View member use case diagram:

Participants: Librarian, <<view>> :MainPage, <<Controller>> :ViewStudent, <<view>> :StudentPage, <<view>> :ViewFailureMassage

1 : Login(name,password)
2 : Click Student Add / View / Delete Button
3 : Fill student id
4 : Click View Button
4.1 : verify
alt
4.2:Checks if there is same student
View Valid
4.3:ViewTheStudent
View Invalid
4.2:Checks if there is same student
Redirect

Self return use case:

Self return use case diagram:
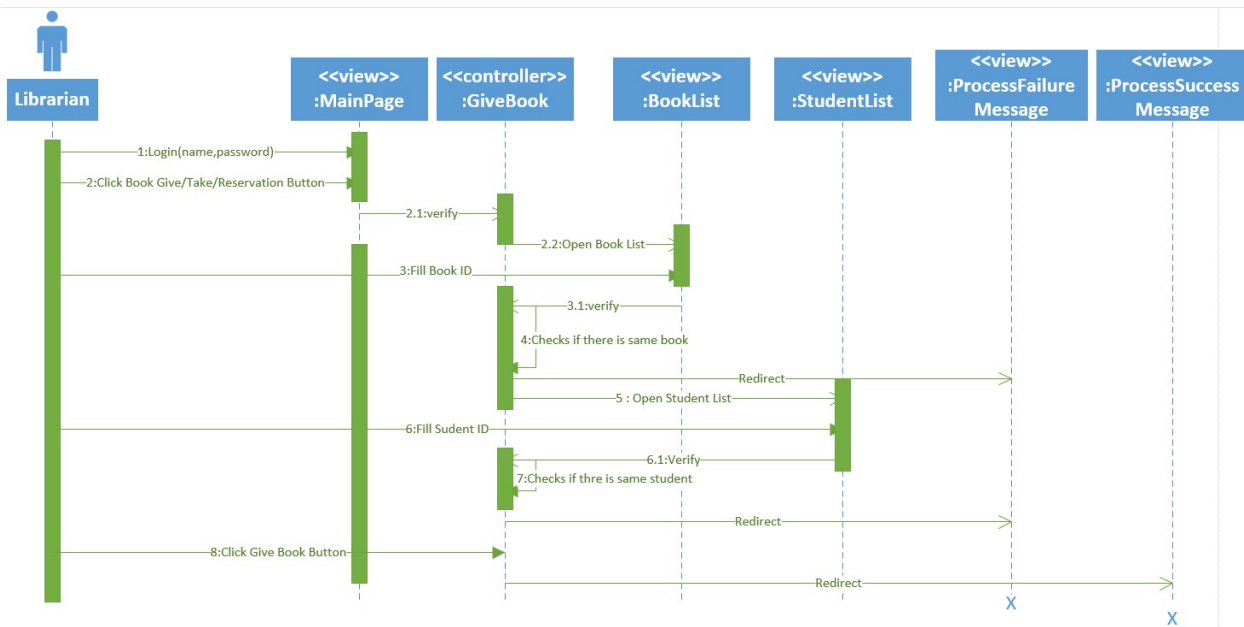
Participants: Librarian, <<view>> :MainPage, <<controller>> :GiveBook, <<view>> :BookList, <<view>> :StudentList, <<view>> :ProcessFailure Message, <<view>> :ProcessSuccess Message

1:Login(name,password)
2:Click Book Give/Take/Reservation Button
2.1:verify
2.2:Open Book List
3:Fill Book ID
3.1:verify
4:Checks if there is same book
Redirect
5 : Open Student List
6:Fill Sudent ID
6.1:Verify
7:Checks if thre is same student
Redirect
8:Click Give Book Button
Redirect

Borrow book use case:
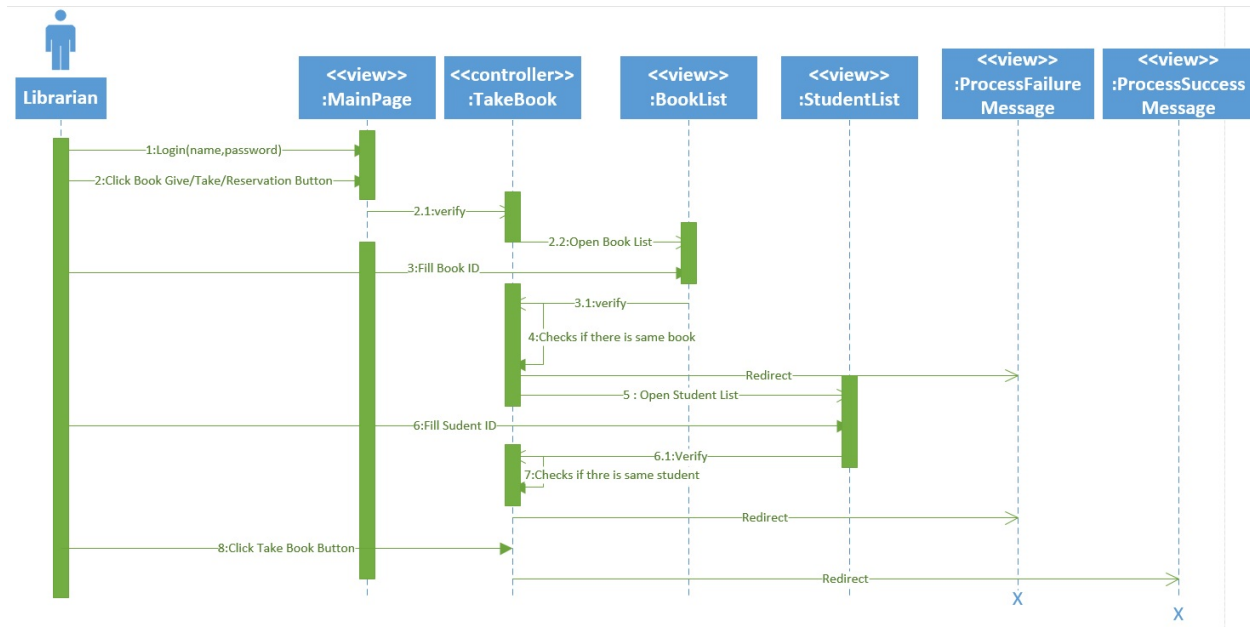
4.3   Data Model (E-R Diagram)
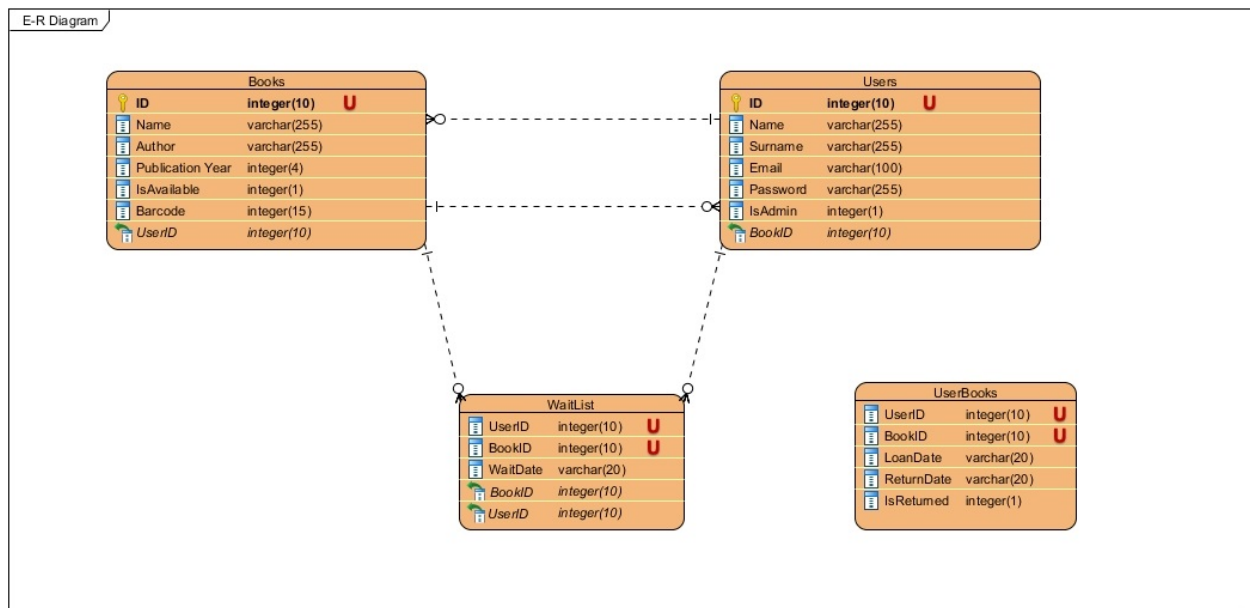
4.4   User Interface Design

When the design were being made, the thing that we thought first was the color spectrum. Since when we think about the books the main color that comes to mind is yellowed page color, we used this color range almost for all of the design.

After determining the color the main concern was how to make it interesting and useful at the same time. Buttons are easy and understandable but can't be used when choosing and element from a list because it is not useful. Eventually we decided that we will use buttons only when they are necessary. With that thought the borrowing operation will be done in the book pages which will be explained in this document after.

There is a link to home page in every page which is an effective solution to access everywhere.

<u>User Interface Design of our Library Book Loan System is provided in appendix which name is GUI.pdf</u>

5.   Requirements Traceability

	Login	Logout	Add Name to Book Waiting List	View My Books	Manipulate Books	Manipulate Members
login.php	X					
logout.php		X				
addWaitList.php			X			
mybooks.php				X		
addbook.php					X	
updatebook.php					X	
deletebook.php					X	
adduser.php						X
deleteuser.php						X
updateuser.php						X

6.   Annexes

User intefaces are provided in the GUI.pdf file.

We used PHP Language in our Software and we applied PHP's coding standard rules in our Software. You can find the Coding Standard in the Coding Standard.pdf file.