## **Term Project (individual) report**

**201636417 심우석**

1. Introduction
   1. Purpose of the system

I thought I needed a library management program to keep a lot of books in the library and make them available to users efficiently.

This program should be readily available to both library users and administrators. For example, a user should be able to check the location of a book, and whether the book to be borrowed is available for loan.

Administrators will need to check the location of the book to return the returned book to its original location, also check overdue books for use by other users. Also, if we include the function to read or study books in the reading room, both users and administrators can be more convenient libraries

1.2 Scope of the system

We aim to create a library management program that is available to both small and large libraries. In a small library, the program is used as it is, and in a large library, the interoperability of the program is enhanced through database linkage with other libraries.

For example, you tried to borrow a book called "ABC" from the library closest to your house, but there may be no books in the library in front of your house. At that time, you can find a library close to the book or check whether the book is in a specific library

1.3 Objectives and success criteria of the project

The first goal is to create programs that can be used in small libraries. And the second goal is to connect databases to enable interoperability between multiple libraries.

And the code of the program should be designed without error and there should be no safety problem when the program is executed.

1. Current system (Related works)

Currently using these programs...

1. KOLASYS-NET (Korean Library Automation System) : it is used in public libraries such as the National Library of Korea and the library under the Yeongdeungpo Cultural Foundation.

2. **ILUS : it is a system developed by Futurenuri Co., Ltd. And is being used in all municipal libraries and small libraries in Gangseo-gu / Gangnum-gu, Seoul.**

3.  **“책꽂이” : it is used in small apartments. When purchasing many new books, you can receive the MARC file together and download the bibliography to simplify the office work**

4. **DLS (Digital Library System): ﻿**A system developed by the government and provided to elementary, middle and high schools for free.

All these programs are very well structured. However, they have some problems such as not supporting MARC function, there is no serial module, or operating cost is high etc. Therefore, it would be a much better program to supplement these problems

1. Proposed system

3.1 Overview

People who use the program are divided into users and administrators. And the use section is divided into library and reading room.

Through the program, the user checks if the book is available for loan now, Where the book is located, and you can search how many books are currently available for loan. You can also borrow or return books through a program or administrators.

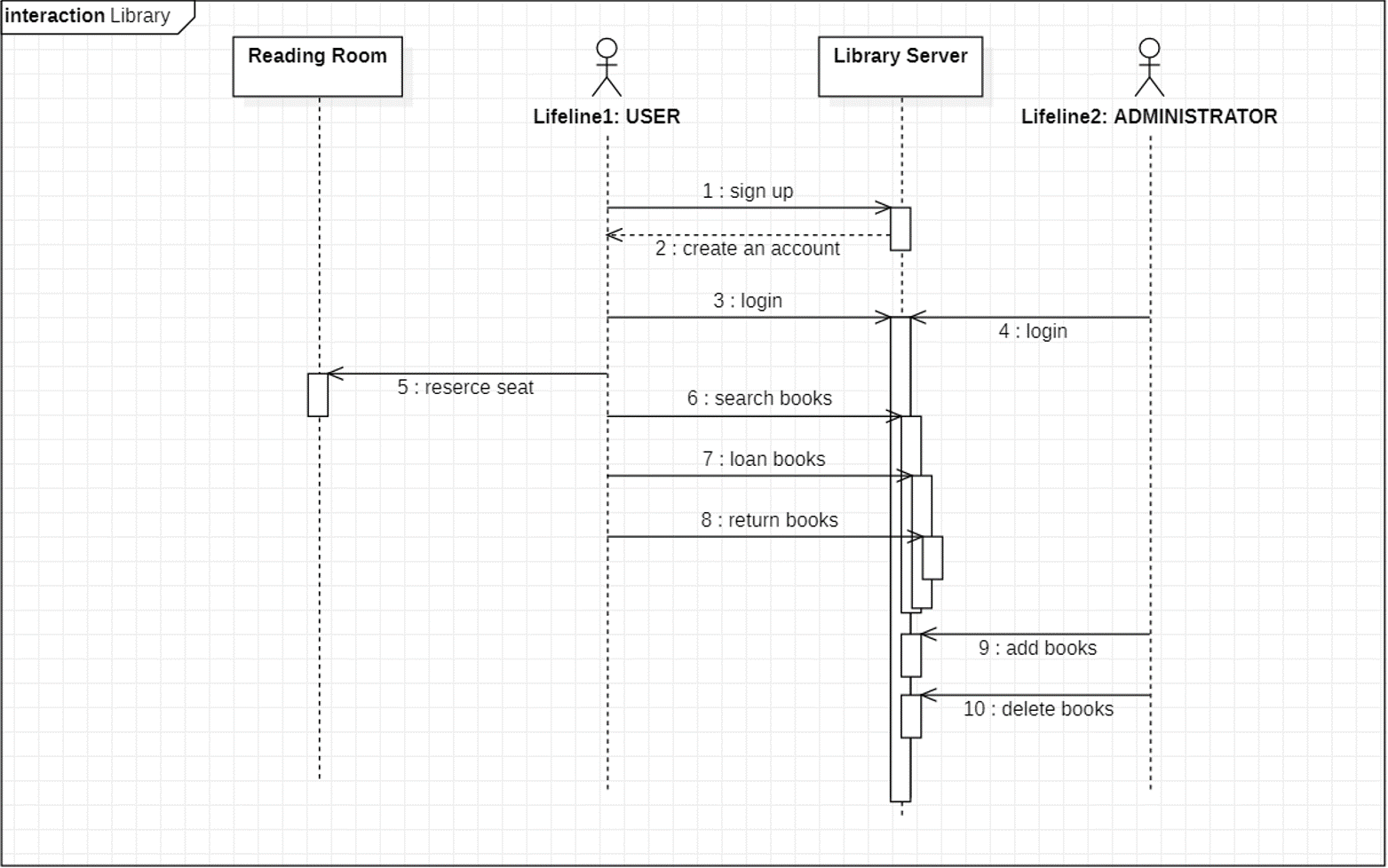
The administrator will enter the book's unique number into the program and check how many books are currently overdue.

In addition, the user can use a program such as which seat is empty in the reading room, reservation of a seat, and the like.

3.2 Functional requirements (“shall lists”)

Actor is divided into user and administrator. Also, system is divided into reading room, library server.

The user can connect to the system to reserve a seat in the reading room, and in the library server, user can sign up for membership, search for books, lend and return.

Administrator can add / remove books after logging into the program. In addition, administrator can keep track of the current state of library’s operations by managing the current state of library’s books.

* 1. Nonfunctional requirements
     1. Usability (사용성)

We will implement the program using Java. It will be implemented in the form of an application, but if it can be expanded, it will also work in the form of a web.

Stakeholders who are not developers do not have background knowledge about the program, so we will use the GUI to supplement the interface.

In addition, the database will be configured and linked separately from the program. This connection will be implemented with Client – server architecture. If we do this, it will be easier to manage the storage of the program

* + 1. Reliability (신뢰성)

Programs used by users and administrators will be separated from each other. In order to access the programs used by administrators, a password must be entered.

In addition, the user’s personal information, etc. must enter the password of the final administrator, not the general administrator. This program uses a layered architecture to put sensitive information such as user’s personal information at the bottom of the program.

Because of this, security will be better. In addition, safety will be secured through a lot of testing.

* + 1. Performance

In the process of implementation, the function of the system is very important. But we will not miss both consistency and completeness.

And if possible, we will support both web and app to increase user convenience. Also, when user access the app, user will be able to easily view PDF files for periodicals.

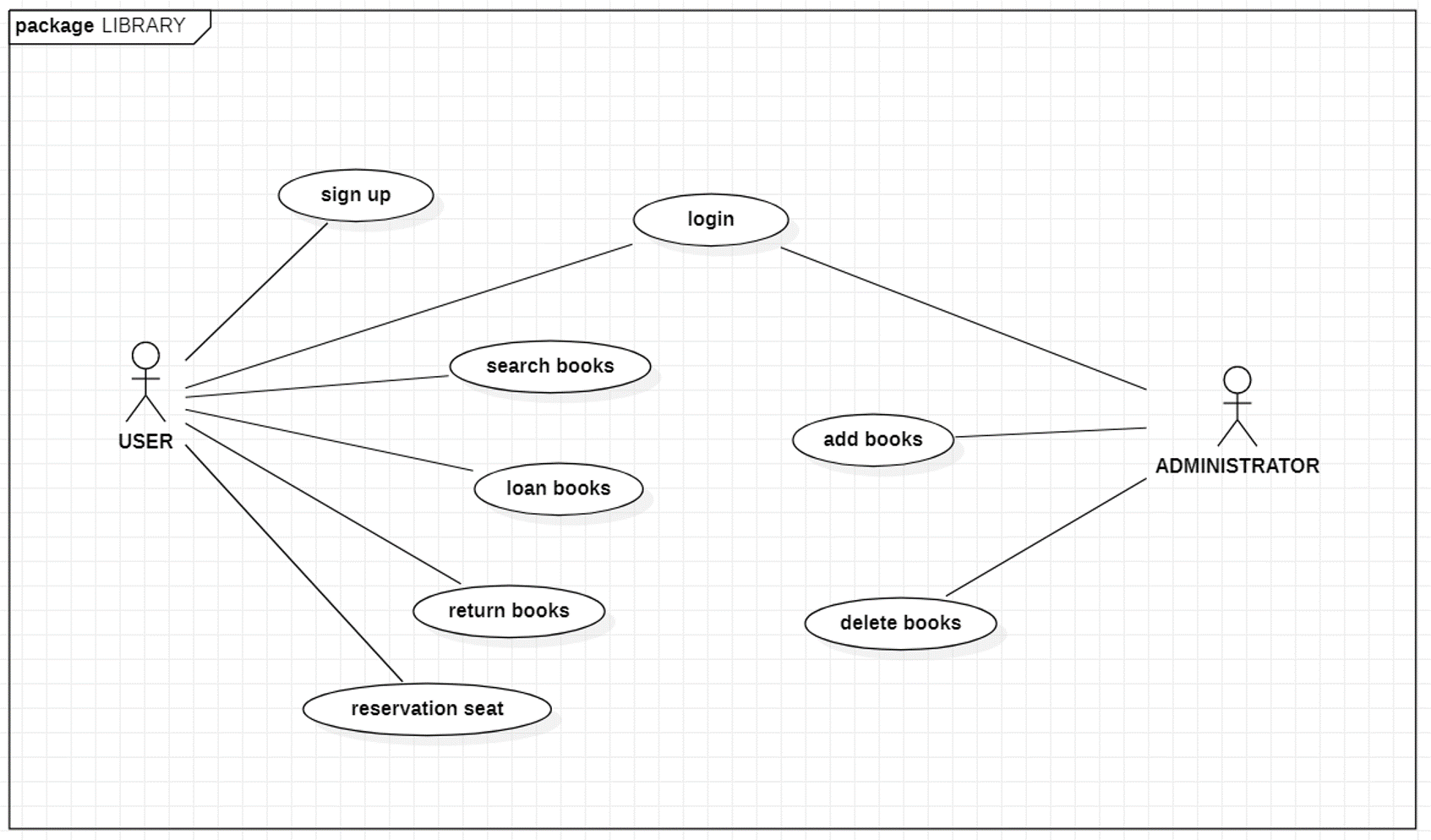
It will show performance as a more advanced program from the general library operation program

* 1. System models
     1. Scenarios

First, we need to define the system boundaries. For Example, there is a book loan / return system that should be in a library management program. Conversely, parts such as a database of books or personal information of users will be linked to the program.

In addition, the program will be implemented while the program is envisioned using Agile development.

* + 1. Use case model

When doing system modeling, our team will actively utilize UML. Through this, it will be possible to fill in the lack of features that are not currently implemented. Like this.

* + 1. Analysis object model

The structural model will be used to show the components and structure of the program. We will also show how the program functions using Behavioral models.

And we will compare it to the code of the program currently being used in libraries. By doing this, we will be able to complement the functional parts of the program we are going to implement and find out what to add. In addition, non – functional parts such as security may be additionally supplemented.

1. Test and evaluation
   1. Test case generation

When implementing code in Java, we will steadily fix errors using the debugger. Also, through the unit test, the smallest unit will be tested and analyzed one by one to reliably correct errors.

Next, we will fix the overall error in the program using a test case generator. Also, when implementing the interface thorough the GUI, errors in the interface will be corrected using the GUI capture and replay program.

It will also block problems caused by configuration management.

* 1. Code coverage

To comply with the license, the current library operating program will use a source that can be modified and distributed.

So, we will use program sources such as “Open biblio”, “dongwoon books”. These programs are BSD.

We plan to modify the necessary parts and modify the interface using the source of these programs.

1. Glossary
2. Database : a database is an organized collection of data, generally stored and accessed electronically from a computer system.
3. MARC : (machine readable cataloging). This is a computerized method of recording the information needed in a cataloging record
4. Java : java is a general propose programming language that is class – based, object – oriented, and designed to have as few implementation dependencies as possible.
5. GUI : (Graphical user interface) is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation.
6. Client – server architecture : distributed system model which shows how data and processing is distributed across a range of components
7. Layered architecture : Organizes the system into a set of layers (or abstract machines) each of which provide a set of services
8. Agile development : program specification, design and implementation are inter – leaved. And minimal documentation (focus on working code)
9. UML : (Unified Modeling Language) is a general – purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.
10. Debugger : debugging is a methodical process of finding and reducing the number of bugs, or defects, in a computer program
11. Unit test : unit testing is a method by which individual units of source code are tested to determine if they are fit for use
12. Test case generator : test case generators synthesize full test cases.
13. Configuration management : configuration management is a method is used to manage change
14. BSD : (Berkley standard distribution) is a non – reciprocal license, which means you are not obliged to re – publish any changes or modifications made to open source code.