CalPolyPomona

Development and Management of the Project: UWON Library System

Team Members

	Student 1	Student 2	Student 3	Student 4
Name				

Table of Content

Table of Figures

PROJECT PLAN

CONFIGURATION PLAN

REQUIREMENTS SPECIFICATION

DESIGN

PAPER-BASED PROTOTYPE

Table of Tables

PROJECT PLAN

CONFIGURATION PLAN

REQUIREMENTS SPECIFICATION

TESTING PLAN

Abstract

Library Management System targeted for converting the UWON library system into a new system to ensure more effective access to state-of-the-art books while reducing operational cost. The new UWON library system provides better control over transactions like integrating all departments, requesting books, Bibliographical searches, tracking books' conditions, monitoring loan status, and generating intelligent reports. This project's main motto is to reduce unnecessary duplicate acquisition by several departments and to improve the efficiency of the book search.

The goal of this project is to develop a system for the computerization of the library system which integrates all departments' sub-systems. The common transactions of the UWON library system include the maintenance of mess user registration(staff, student, professor), book and book copy management, book reservation, loan management with limited loan period, and a limited number of simultaneously or successively loans made by users. In addition, Library Management System autonomic provides return deadline warnings and fine issues per day through e-mail. Users can perform bibliographic research in terminals in the library with a self-guided tool and intelligent reports will be generated to guide book acquisition or inform books not returned on time. All bibliographical queries and results shall be accessible to students from any department, and the response to queries will be fast and efficient.

Keywords: UWON, library system, self-guided, intelligent, reservation

Project Plan

1. Introduction

1.1 System-As-Is

The UWON system consists of multiple unconnected department library subsystems. Each department subsystem is responsible for its own library based on department-specific procedures for book acquisition, user registration, loan management, and bibliographical search. Those services are essentially manual in most UWON libraries and rely on card indexes maintained by librarians according to keyword-based classification schemes which are specific to each subsystem (departments). Several departments manage their library system by using rudimentary file-based software. So many issues may happen such as duplicate acquisition, lack of tractability, tedious bibliographical search, incomplete or ineffective search.

1.2 System-to-Be

The New UWON library system integrates all department library subsystems based on a software-based solution and is responsible for all departments according to exactly the same procedures for book reservation, user registration, loan management, and bibliographical search. The new library system should provide a graphical user interface for users and use MySQL databases to maintain users' information, book and book copy reservations' information etc. A self-guided tool should be provided for users conducting bibliographic research in the graphical user interface in the library and an intelligent report should be generated by librarians for guide book acquisition or trace book status. The new library system should reduce unnecessary duplicate acquisition by several departments and improve the efficiency of the book search.

1.3 Project Deliverables

Table 1: Project Deliverables Table

<table>
<tr><th colspan="5">Project Deliverable Table</th></tr>
<tr><td colspan="5"></td></tr>
<tr><td><b>Project Name:</b></td><td colspan="2">UWON Library System</td><td><b>Project ID:</b></td><td>Version 1.5</td></tr>
<tr><td><b>Project Manager:</b></td><td colspan="2">Student 1</td><td><b>Status:</b></td><td>Complete</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td><b>Sr No</b></td><td><b>Deliverable Name</b></td><td><b>Description</b></td><td><b>Owner</b></td><td><b>Status</b></td></tr>
</table>

1	Prepare project plan	Design process Model, method and techniques. Provide guidelines, procedures and standards. Resources distributions and effort schedule. Demonstrate the procedures to be followed	Student1	Complete
2	Prepare configuration Plan	Prepare software configuration management, identify configuration items. Determine environment and infrastructure, and provide versioning and other documents etc.	Student2	Complete
3	Requirements Specification	Clear High-Level Goals, Use Case diagram and specifications. Provide Functional requirement and non-functional requirement. Traceability Matrix is also included.	Student3	Complete
4	Design	ER diagram, Logical Model, Class Diagram, and other diagrams etc.	Student4	Complete
5	Prepare testing Plan	Prepare unit tests, integration tests, and system.Acceptance tests, and provide Quality Assurance plan	Student2	Complete
6	Design prototype	Design paper-based prototype and provide technique description of the system	Student3	Complete
7	Discussion	Analysis of the results obtained, and lessons learned	Student4	Complete

2. Process Model

2.1 Activities to be Undertaken

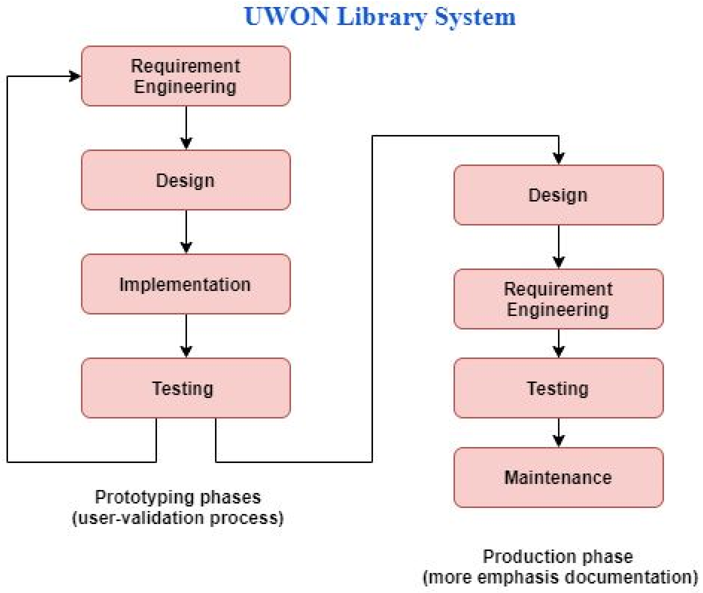


Figure 1. Evolutionary Prototype Model

The UWON Library System will use Evolutionary Prototype as we see from Figure1.1, since we have two prototypes, the development of one more prototypes may help me to better capture those requirements. These prototypes are tested intensively before a real production line is set up. We will fix errors or bugs during the user-validation process (2 prototype) and then move to the production phase. That's the reason why we use Evolutionary Prototype Model

2.2 Milestone to Measure Progress

Table 2: Milestone to Measure Progress

Activity	Duration	Constraints	Milestone
Requirements	5		
Design	10	Requirement finished	
Test plan	5	Design finished	M1
Coding	10	Design finished	M2
Test	10	Coding finished, Test plan finished	M3

After requirements are finished, we will conduct project design and project design is also a constraint of the Testing plan so we set the Test plan as milestone M1. Project Design is also a constraint of Coding, and we will conduct Coding when Project Design finishes. We set the Coding as milestone M2. Coding and Test plan are constraints of Test and we set Test as milestone M3. All milestones make sure the project is on a good track and helps to keep it on time and on budget.

3. Organization of the Project

3.1 Information, services, resources, and facilities to be provided

All book and book copy resources should be provided by the old library system database which is from each department sub-system database when the UWON library system starts. The University of Wonderland will provide all users' information and facilities information anytime is needed. Most library services will be inherited from the old system and also add new features such as self-guide tools, intelligent reports and etc.

3.2 Specification of the roles

Table 3: Specification of Roles

Name	Role	Responsibility
Student1	Project manager	Design process Model, method and techniques. Provide guidelines, procedures and standards. Resources distributions and effort schedule. Demonstrate the procedures to be followed
Student1	Designer	Design ER-diagram, Logical Model, Class Diagram, and other

		diagrams, etc. Design paper-based prototype and provide technique description of the system
Student2	Analyst	Clear High-Level Goals, Use Case diagram, and specifications. Provide Functional requirements and non-functional requirements. Traceability Matrix is also included
Student3	Programmer	Coding and debugging. Designing and testing computer structures. Troubleshooting system errors. Writing computer instructions. Managing database systems.
Student3	Tester	Prepare unit tests, integration tests, and system. Acceptance tests. Reviewing software requirements and preparing **test** scenarios. Executing tests on software usability. Analyzing **test** results on database impacts, errors or bugs, and usability.

4. Methods and techniques

Teams will design ER-Diagram, Use Case Diagram, Traceability Matrix by using **draw.io** during the requirements engineering and also will provide UML, Logical Diagram, and Physical Diagram by using **draw.io** and **sql develope**r in order to make the project more clear and organized. We will conduct our coding and testing by using **Java** language in **Visual Studio IDE**, and all data or information will be stored in the **MySQL** database. For UI design, teams will use **SWING GUI**, in addition, automating testing will be also provided by using Selenium, Cucumber, Junit. Since we will use Evolutionary Prototypes, we will compose our documentation in the **JIRA** software during the production phase.

5. Standards, guidelines, procedures

One of the teams constantly reviews requirement specifications, and makes sure to meet all the requirements. One of the teams will periodically inspect project progress and project quality before moving to subsequent phases correctly, after that teams will write verification processes. Since **Initial Level** is being currently implemented(CMMI), teams currently implement Requirements managements, Project planning, Project monitor and control and also working on Process and product quality assurance , Measurement and analysis and Configuration management. After teams finished those process, project will move to **Repeatable Level**

6. Work packages

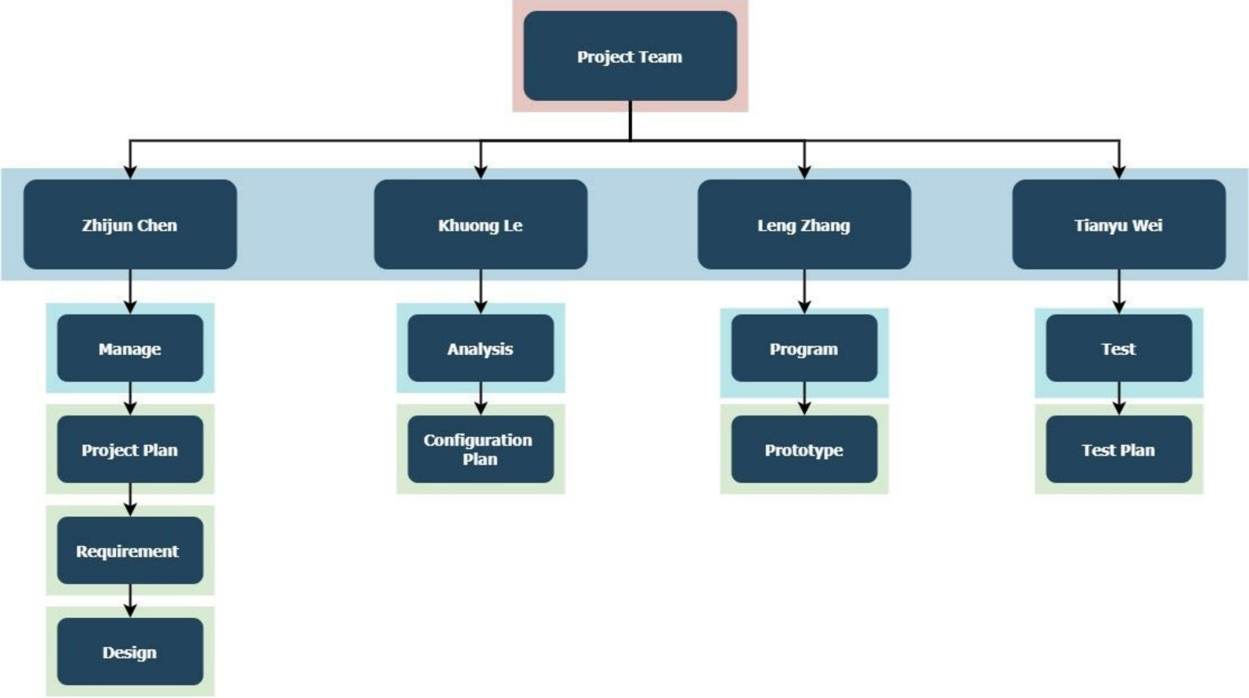


Figure 2. Work Packages

7. Resources

7.1 Hardware

The UWON Library System uses Javascript for frontend, Java for backend, and MySQL for database. Any OS that supports Javascript, Java and MySQL should be able to support UWONLibrary System. For instance, Windows, Linux, and Mac OSX. Suggested minimal production environments: Quad Core (64 bit CPU and OS), 4 GB RAM, and 100 GB HDD. In addition to the computer and display, the program needs a normal computer keyboard and mouse for controlling it.

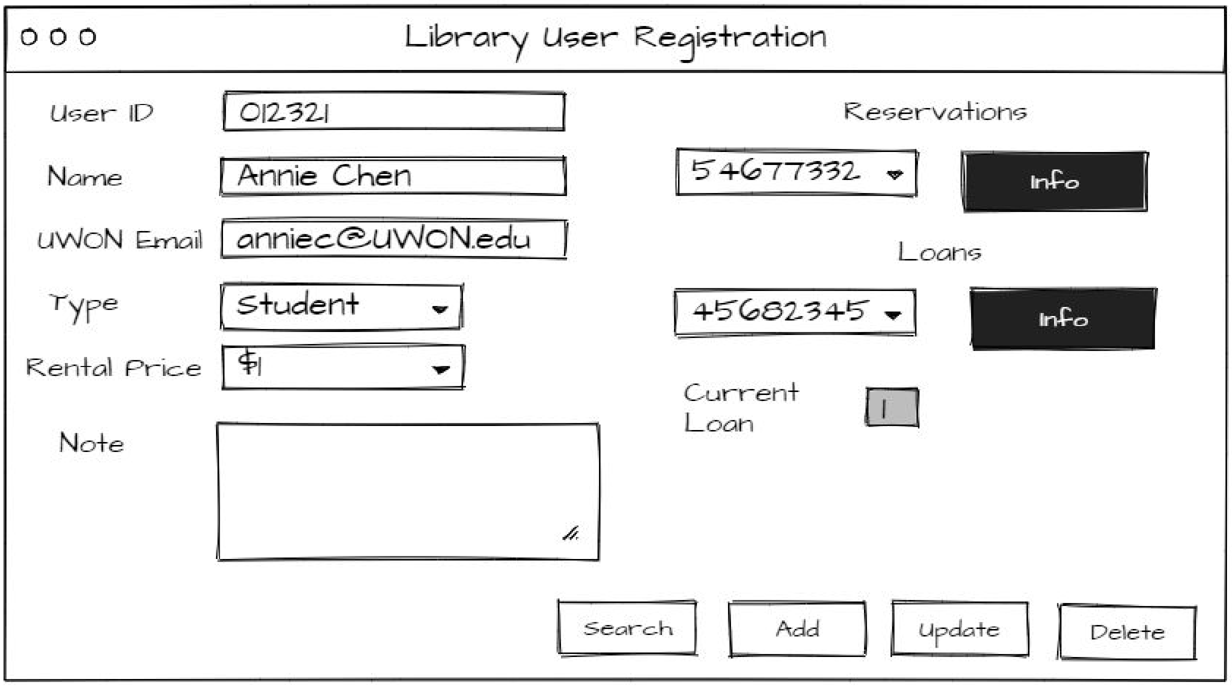
7.2 Personnel

One project manager, One designer, One analyst, One programmer, One tester

8. Effort and schedule

8.1 Effort

SCREEN_01



Library User Registration

User ID: 012321

Name: Annie Chen

UWON Email: anniec@uWON.edu

Type: Student

Rental Price: $1

Note:

Reservations: 54677332

Loans: 45682345

Current Loan: 1

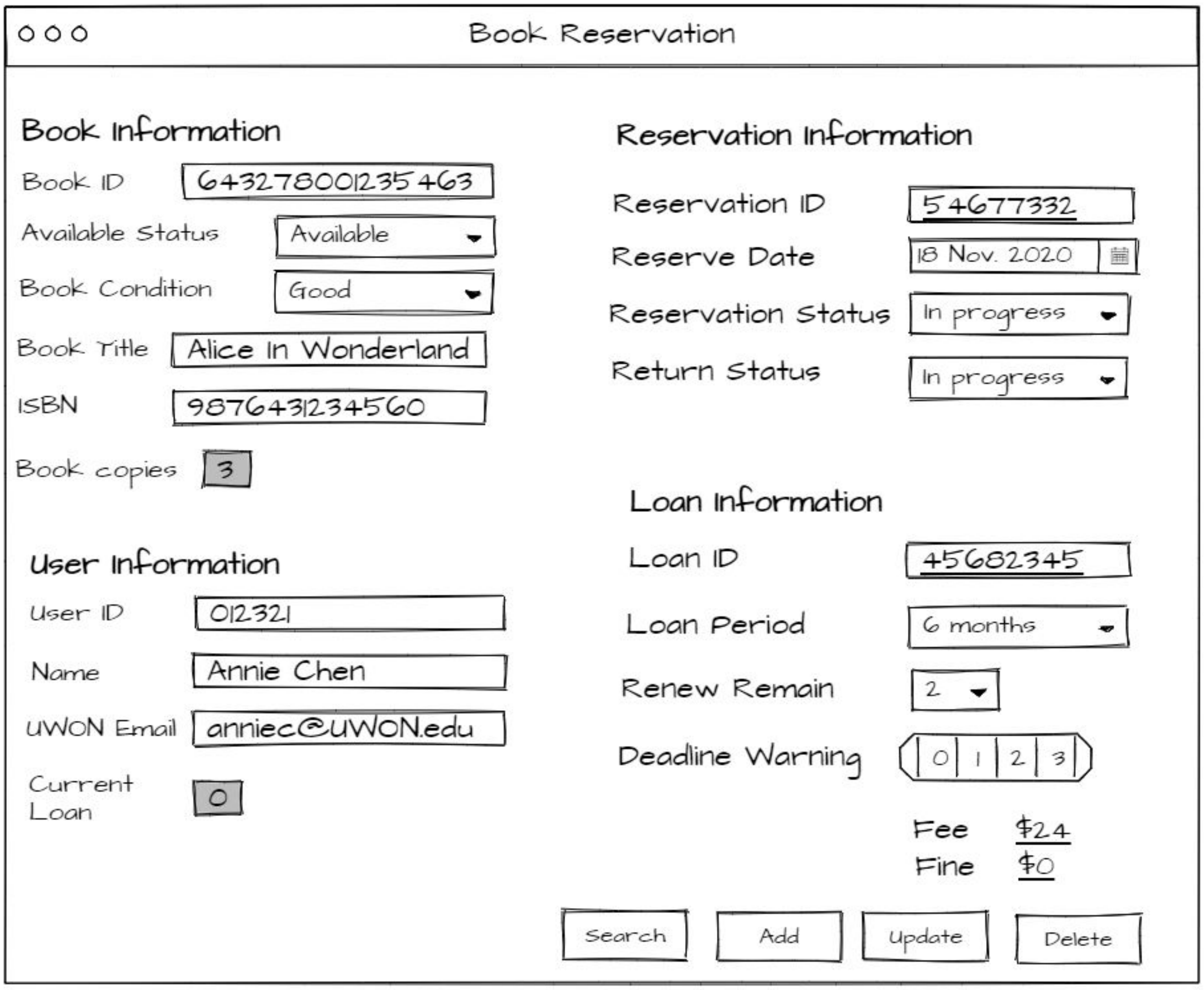
Figure 3. SCREEN_01: Library User Registration

Table 4. SCREEN_01 Function Points

Elementary Process	Type	DET	RET/FTR	Complex	FP
Files					
User	ILF	5	2	Low	7
Reservation	EIF	4	1	Low	5
Loan	EIF	6	1	Low	5
Functions					
Buttons					
Search	EO	9	3	Average	5
Add	EI	9	1	Low	3
Update	EI	9	1	Low	3
Delete	EI	9	1	Low	3

Reservation Info	EO	3	1	Low	4
Loan Info	EO	3	1	Low	4
Dropdown List					
Dropdown Type	EQ	3	1	Low	3
Dropdown Price	EQ	3	1	Low	3
Dropdown Reservation	EQ	3	1	Low	3
Dropdown Loan	EQ	3	1	Low	3
Calculated Box					
Current Loan	EO	1	1	Low	4
Total					65

SCREEN_02



Book Reservation

Book Information

Book ID 643278001235463

Available Status Available

Book Condition Good

Book Title Alice In Wonderland

ISBN 9876431234560

Book copies 3

User Information

User ID 012321

Name Annie Chen

UWON Email anniec@UWON.edu

Current Loan 0

Reservation Information

Reservation ID 54677332

Reserve Date 18 Nov. 2020

Reservation Status In progress

Return Status In progress

Loan Information

Loan ID 45682345

Loan Period 6 months

Renew Remain 2

Deadline Warning 0 | 1 | 2 | 3

Fee $24

Fine $0

Figure 4. SCREEN_02: Book Reservation

Table 5. SCREEN_02 Function Points

Elementary Process	Type	DET	RET/FTR	Complex	FP
Files					
Book Info	ELF	6	2	Low	5
User Info	EIF	4	2	Low	5
Reservation Info	ILF	4	1	Low	7
Loan Info	EIF	6	2	Low	5
Functions					
Buttons					

Search	EO	20	4	High	7
Add	EI	20	4	High	6
Update	EI	20	4	High	6
Delete	EI	20	4	High	6
Dropdown List					
Dropdown Available Status	EQ	3	1	Low	3
Dropdown Book Condition	EQ	3	1	Low	3
Dropdown Reserve Date	EQ	3	1	Low	3
Dropdown Reservation Status	EQ	3	1	Low	3
Dropdown Return Status	EQ	3	1	Low	3
Dropdown Loan Period	EQ	3	1	Low	3
Dropdown Renew Remain	EQ	3	1	Low	3
Calculated Box					
Book Copies	EO	1	1	Low	4
Current Loan	EO	1	1	Low	4
Deadline Warning	EO	1	1	Low	4
Fee	EO	1	1	Low	4
Fine	EO	1	1	Low	4
Total					98

SCREEN_03

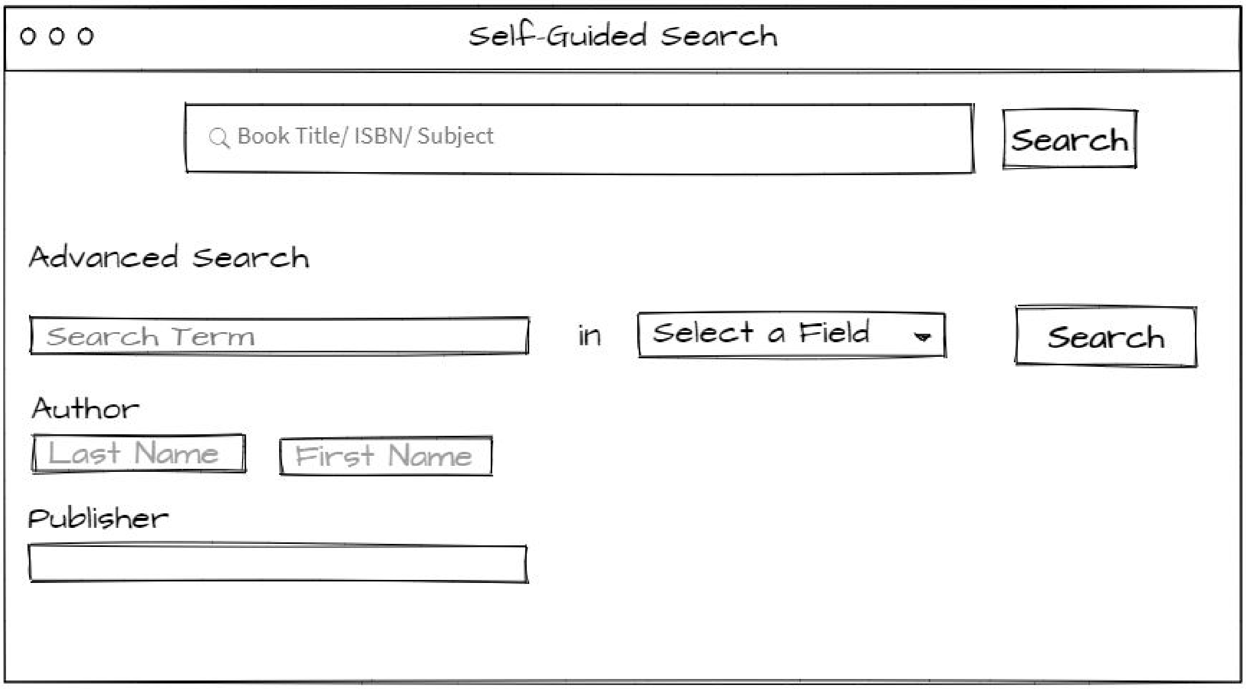
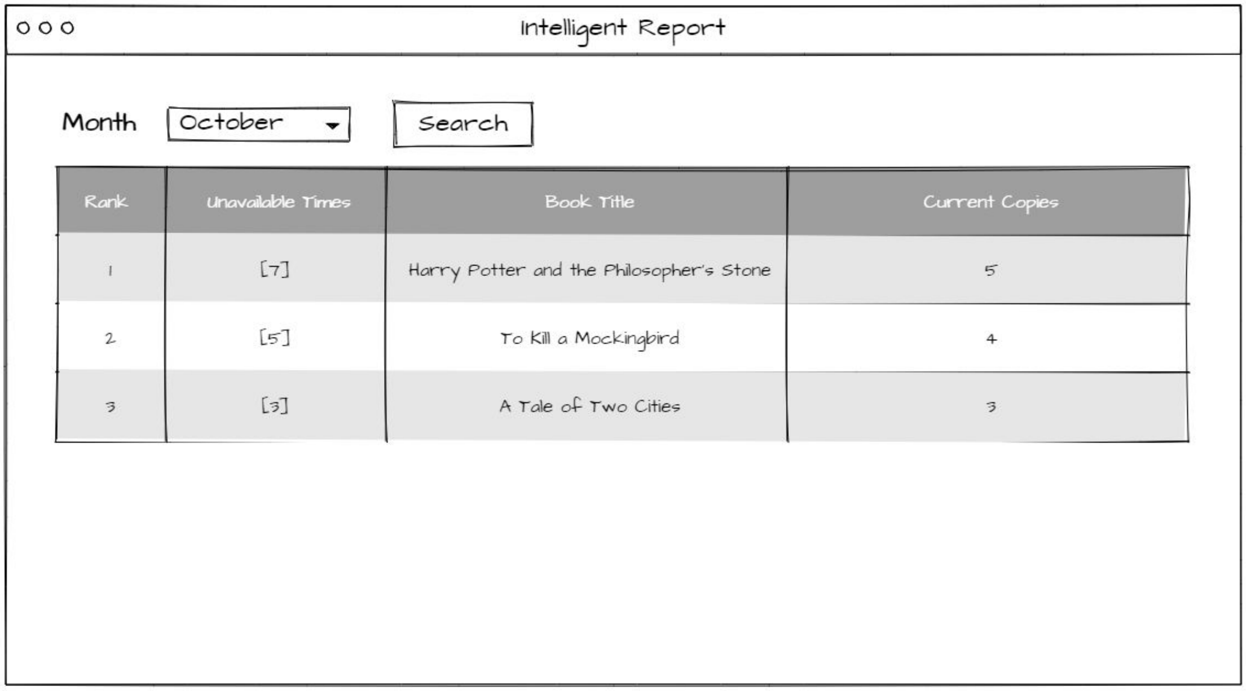


Figure 5. SCREEN_03: Self-guided Search

Table 6. SCREEN_03 Function Points

Elementary Process	Type	DET	RET/FTR	Complex	FP
Files					
Book Info	EIF	6	2	Low	5
Functions					
Buttons					
General Search	EQ	1	2	Low	3
Advanced Search	EQ	5	2	Low	3
Dropdown List					
Dropdown Field	EQ	3	1	Low	3
				Total	14

SCREEN_04



Rank	Unavailable Times	Book Title	Current Copies
1	[7]	Harry Potter and the Philosopher's Stone	5
2	[5]	To Kill a Mockingbird	4
3	[3]	A Tale of Two Cities	3

Figure 6. SCREEN_04: Intelligent Report

Table 7. SCREEN_04 Function Points

Elementary Process	Type	DET	RET/FTR	Complex	FP
Files					
Report	EIF	3	2	Low	5
Functions					
Buttons					
Search	EQ	3	2	Low	3
Dropdown List					
Dropdown Month	EQ	3	1	Low	3
Display					
Book List	EO	4	2	Low	4
				Total	15

The total effort (measure in terms of number of function points) estimate for the project:

65+98+14+15=192

8.2 Schedule

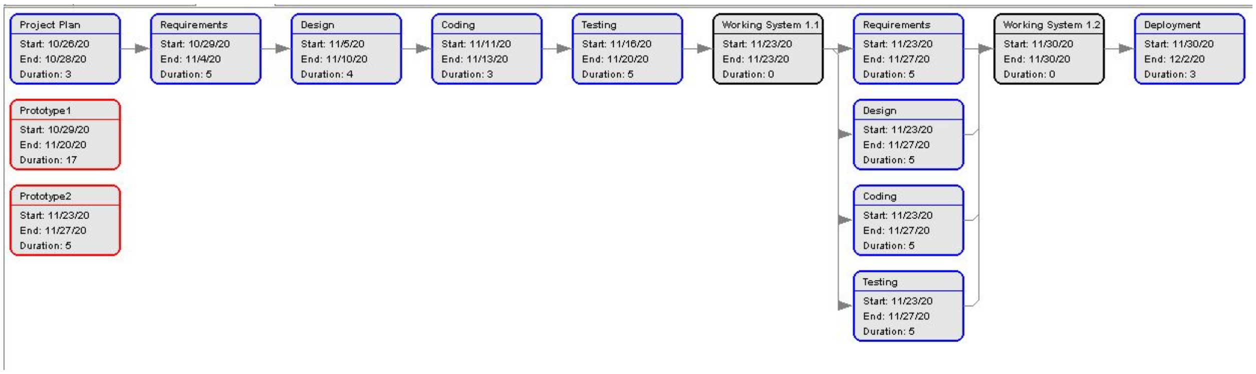
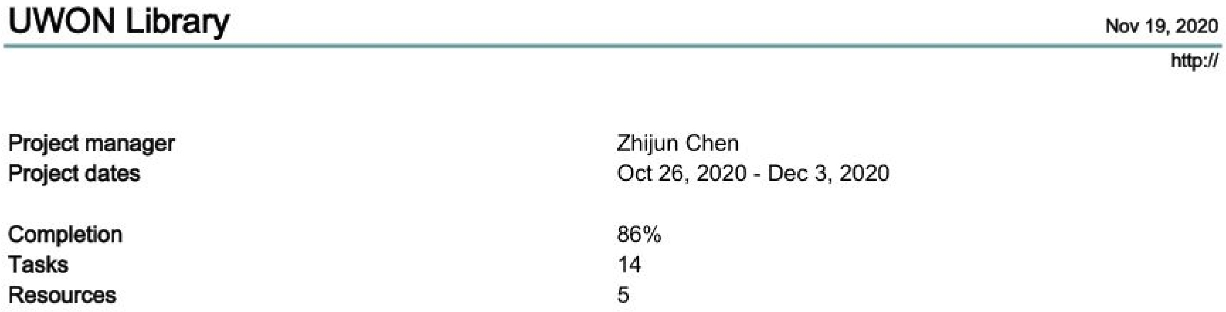


Figure 7. PERT Chart



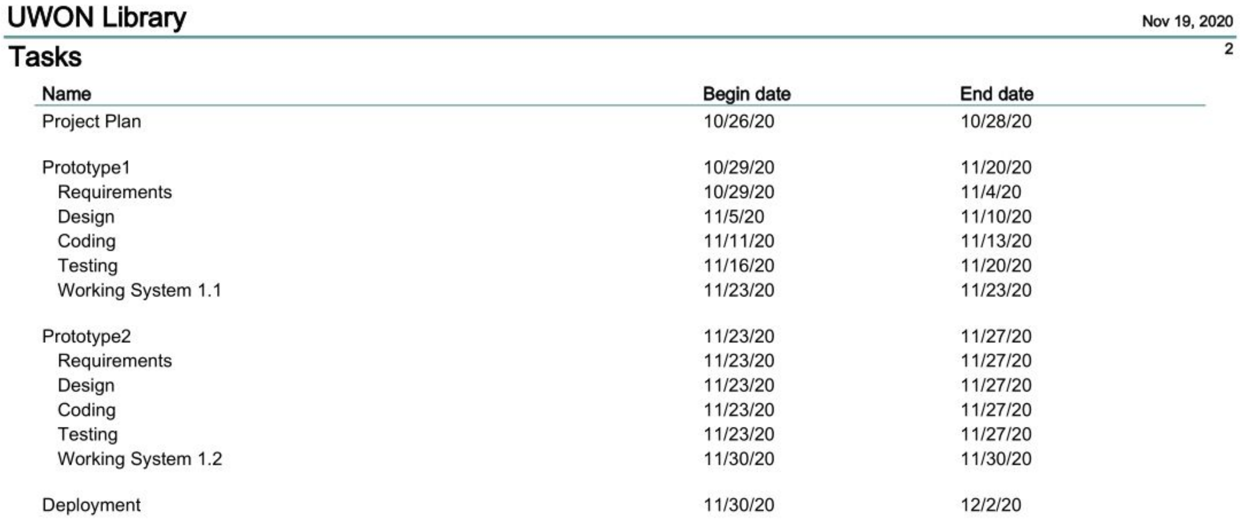
UWON Library

Nov 19, 2020

http://

Project manager	Zhijun Chen
Project dates	Oct 26, 2020 - Dec 3, 2020
Completion	86%
Tasks	14
Resources	5

Figure 8. Project Information



UWON Library

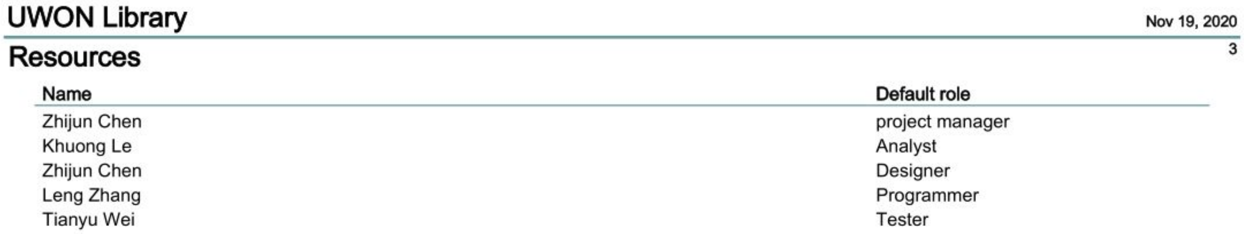
Nov 19, 2020

Tasks

2

Name	Begin date	End date
Project Plan	10/26/20	10/28/20
Prototype1	10/29/20	11/20/20
Requirements	10/29/20	11/4/20
Design	11/5/20	11/10/20
Coding	11/11/20	11/13/20
Testing	11/16/20	11/20/20
Working System 1.1	11/23/20	11/23/20
Prototype2	11/23/20	11/27/20
Requirements	11/23/20	11/27/20
Design	11/23/20	11/27/20
Coding	11/23/20	11/27/20
Testing	11/23/20	11/27/20
Working System 1.2	11/30/20	11/30/20
Deployment	11/30/20	12/2/20

Figure 9. Tasks Schedule



UWON Library

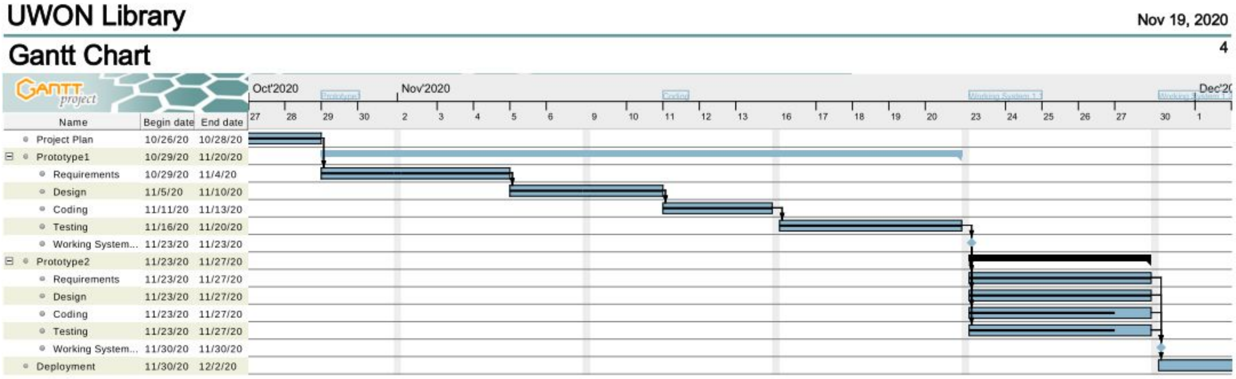
Nov 19, 2020

Resources

3

Name	Default role
Zhijun Chen	project manager
Khuong Le	Analyst
Zhijun Chen	Designer
Leng Zhang	Programmer
Tianyu Wei	Tester

Figure 10. Resources Information



UWON Library

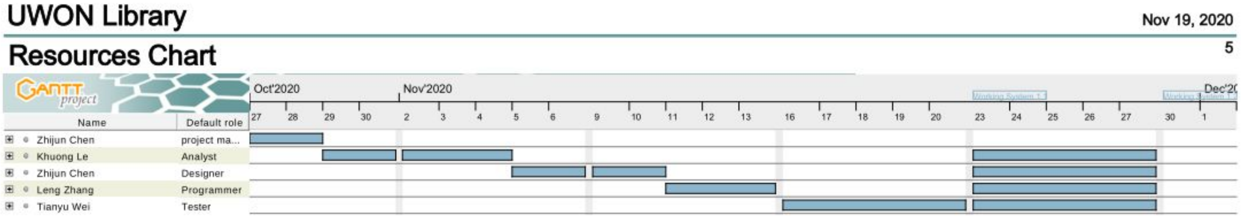
Nov 19, 2020

Gantt Chart

4

Name	Begin date	End date
Project Plan	10/26/20	10/28/20
Prototype1	10/29/20	11/20/20
Requirements	10/29/20	11/4/20
Design	11/5/20	11/10/20
Coding	11/11/20	11/13/20
Testing	11/16/20	11/20/20
Working System...	11/23/20	11/23/20
Prototype2	11/23/20	11/27/20
Requirements	11/23/20	11/27/20
Design	11/23/20	11/27/20
Coding	11/23/20	11/27/20
Testing	11/23/20	11/27/20
Working System...	11/30/20	11/30/20
Deployment	11/30/20	12/2/20

Figure 11. Gantt Chart



UWON Library

Nov 19, 2020

Resources Chart

5

Name	Default role
Zhijun Chen	project ma...
Khuong Le	Analyst
Zhijun Chen	Designer
Leng Zhang	Programmer
Tianyu Wei	Tester

Figure 12. Resources Chart

9. Delivery

9.1 Login procedure:

- In order to use UWON library system, user must be a member of UWON and have a valid library account.

9.2 Reservation procedure:

For library user:

- If user wants to take the book out of the library, s/he must go to the help desk and ask a librarian to make a reservation for that book. A user can only reserve/borrow books if they are available on the shelves.
- While making the reservation, user must provide a valid library account and agree to pay for the loan for that reservation.
- A loan will be U$1, U$2, U$3 per week for staff, student, professor respectively.
- User cannot have more than 3 loans at the same time. In order words, a user may not borrow more than three books at the same time.
- The maximum loan period of a loan is 6 months.
- User can renew a loan up to 2 times. While approaching to deadline, UWON library system will

send out deadline warning through UWON email within 3 days.

- User fails to pay for the loan will be charged fine U$1 per day.
- User will receive a message about the daily fines and fine being charged after the due date.
- User can check their reservation and loan information in the library system.

For librarian:

- A book can be borrowed by a user only if at least one copy of the book is available.
- After the reservation, the available status of that book with a unique book id will be marked unavailable and total copies of that book with the same ISBN will decrease 1.
- Librarian should check the condition of the book (good/damaged) and record it before lending it to the user.
- After the due date of a reservation, if the user returns the book and pays the fee, librarian can change the reservation status to “finish” and return status to “return”. If the user doesn’t return the book on time, change the reservation status to “past due” and return status to “no return”.

9.3 Search procedure:

- User can search the book title/ISBN/subject in the search bar for general search.
- User can also use advanced search with a specific term in a selected field, author’s name, and publisher.

9.4 Intelligent Report procedure:

- Librarians can get an intelligent report about the popular book in the months.
- If the system shows unavailability of a book 3 times in a month, this information should be included in an intelligent report for future acquisition of extra copies.

Configuration Plan

1. Introduction

1.1 Conventions:

- Each developer works on own branch.
- Pull request has to be created for merging new features or bug fixing to the dev branch and request at least 2 reviewers to approve merging.
- Updates from the dev branch have to push to the beta server for testing and can be pushed to the production server after passing all tests in the beta server.

1.2 Terms:

REQ	- Requirements
BLD	- Builds
MN	- Manuals
DEV	- Development
PROD	- Production
DEVS	- Development Server
BETAS	- Beta Server
PRODS	- Production Server

1.3 Abbreviations Identify Configuration Manager (s):

ZC - Student1

TW - Student2

KL - Student3

LZ - Student4

2. Software Configuration Management

List of Process Activities

- Have a group meeting 3 times a week (Monday, Wednesday, Friday) to discuss what we are going to do and update on the current process.
- Identify High-Level Goals
- Identify Primary and Secondary Actors
- Identify Use Case Diagrams and Specifications
- Identify Requirements
- Identify Configuration Items
- Identify Responsibility for each Configuration Item
- Create Repository
- Commit

- Generate Versions and Baselines

3. Identify Configuration Items

Configuration items that will be produced throughout the project life cycle

- Project Plan
- Configuration Plan
- Requirements Specification
- Design Specification
- Paper-based Prototype
- Test Plan
- Quality Plan

4. Identify Responsible for each Configuration Item

Table 1: Configuration Item Responsibility

Configuration Item	Responsible Person
Project Plan	TW
Configuration Plan	KL, LZ
Requirements Specification	ZC
Design Specification	ZC, KL,TW
Paper-based Prototype	ZC
Test Plan	LZ
Quality Plan	ZC, TW, KL, LZ

5. Tools

Version control tools that will be used.

- GitHub: Control versions of the apps
- Jira: Running sprints through this software. Managing planned tasks, in process tasks and testing tasks and finished tasks.
- IDE: Eclipse

6. Environment and Infrastructure

- There are going to be 3 different servers: dev, beta, production: dev server is for developing, beta server is for internal testing, production server is for the releasing
- All the manual docs are stored on the dev server.

7. Configuration Policy

Nomenclature for configuration items such as <project> - <type-artifact> - <name>, where <project> is the name of the project, <type-artifact> is the configuration item, <name> is the name of the document

7.1 Project Plan:

UWONLS-ProjectPlan-ProcessModel

UWONLS-ProjectPlan-Organization

UWONLS-ProjectPlan-MethodsAndTechniques

UWONLS-ProjectPlan-Standard

UWONLS-ProjectPlan-WorkPackages

UWONLS-ProjectPlan-Resources

UWONLS-ProjectPlan-Delivery

7.2 Configuration Plan

UWONLS-ConfigurationPlan-Conventions

UWONLS-ConfigurationPlan-Terms

UWONLS-ConfigurationPlan-SoftwareConfigurationManagement

UWONLS-ConfigurationPlan-ConfigurationItems

UWONLS-ConfigurationPlan-Tool

UWONLS-ConfigurationPlan-EnvironmentAndInfrastructure

UWONLS-ConfigurationPlan-ConfigurationAndChangeControl

UWONLS-ConfigurationPlan-SystemBuilding

UWONLS-ConfigurationPlan-ReleaseManagement

UWONLS-ConfigurationPlan-ContingencyPlan

7.3 Requirements Specification

UWONLS-RequirementsSpecification-High-Level Goals

UWONLS-RequirementsSpecification-PrimaryAndScondaryActors

UWONLS-RequirementsSpecification-UseCaseDiagram

UWONLS-RequirementsSpecification-UseCaseSpecification

UWONLS-RequirementsSpecification-Assumptions

UWONLS-RequirementsSpecification-DomainProperties

UWONLS-RequirementsSpecification-FunctionalRequirements

UWONLS-RequirementsSpecification-NonFunctionalRequirements

UWONLS-RequirementsSpecification-TraceabilityMatrix

7.4 Design Specification

UWONLS-DesignSpecification-ERDiagram

UWONLS-DesignSpecification-LogicalModel

UWONLS-DesignSpecification-SequenceDiagram
UWONLS-DesignSpecification-StateMachineDiagram
UWONLS-DesignSpecification-SystemArchitectureViewAndStylePattern

7.5 Paper-based Prototype

UWONLS-PaperBasedPrototype-TechnicalDescription

7.6 Test Plan

UWONLS-TestPlan-Scenarios
UWONLS-TestPlan-UnitTests
UWONLS-TestPlan-IntegrationTests
UWONLS-TestPlan-SystemAcceptanceTests

8. Document Version

Definition of a version numbering scheme for configuration items and baselines
UWONLS.major.minor[.fix][-configuration]

UWONLS	- University of Wonderland Library System
.major	- Main Version with 5 or more new features
.minor	- Minor Version with less than 5 new features or updating features
[.fix]	- bug fixing
[-configuration]	- configuration label

9. Evolution of the Version Number

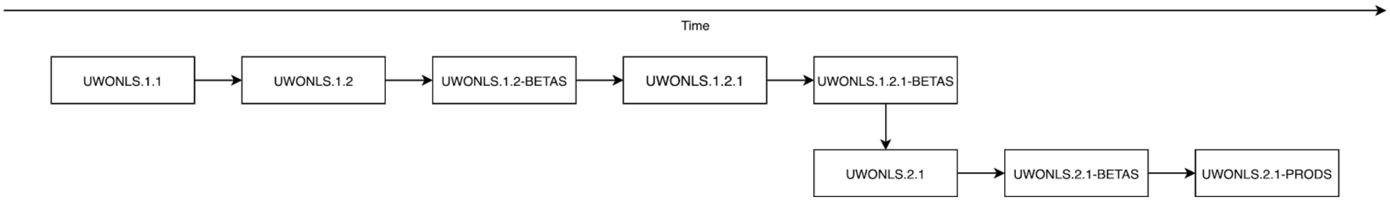


Figure 1. Version Evolution

UWONLS.1.1
Implemented User login by university email and password
UWONLS.1.2
Implemented update password feature
UWONLS.1.2-BETAS
Push version UWONLS.1.2 to beta server and test.
Bug_01: After the user uses the update password feature, the user is not able to login with the new password.

UWONLS.1.2.1

Fixed Bug_01

UWONLS.1.2.1-BETAS

Push version UWONLS.1.2.1 to beta server and test.

UWONLS.2.1

Implemented the self-guided tool which the logined user is able to search books by name, subject, ISBN, publisher, or author.

UWONLS.2.1-BETAS

Push version UWONLS.2.1 to beta server and test.

UWONLS.2.1-PRODS

Push version UWONLS.2.1-BETAS to production server.

10. Configuration and Change Control

We are going to use Trello to update the progress of the project, ZC is the one who manages it.. There are five different columns to keep track of work.

First column is “Investigate”, the owner will create tasks in the “Problems” then our team will decide if we will reject, defer or approve it. If we feel that it’s not feasible, then we reject it. If we think it’s doable but need more confirmation then we request the owner to give me information. If all the information is given and it’s doable, we will set up a plan to finish that task.

Second column is “Schedule”, planning up all subtasks for needed architecture, system, codes, and time for the main task given by the owner.

Third column is “Implementation”, showing what tasks KL and LZ are currently working on. It will show the progress of each task and when will that one finish.

Fourth column is “Testing” to keep track of what tasks have been done by KL, LZ and those tasks are ready to be tested by TW.

Fifth column is “Done” to keep track of all the done tasks, and report to the owner if we finish all the big main task.

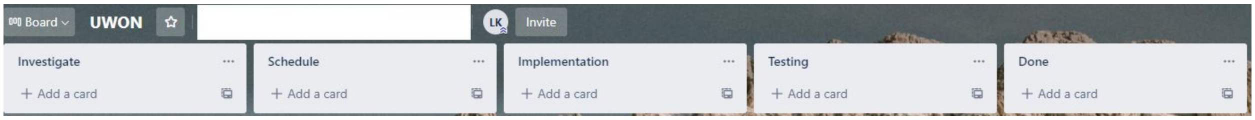


Figure 2. Configuration and Change Control

11. System Building

1. Create repository
2. Configure Java development environment
3. Configure MySQL environment and manage data to match with UWON’s database
4. Follow use cases and implement those functionalities.
5. Follow test cases to validate and verify requirements
6. Compile the project and then write documentations on how to use it.

12. Release Management

Strategy to system delivery, client system version management

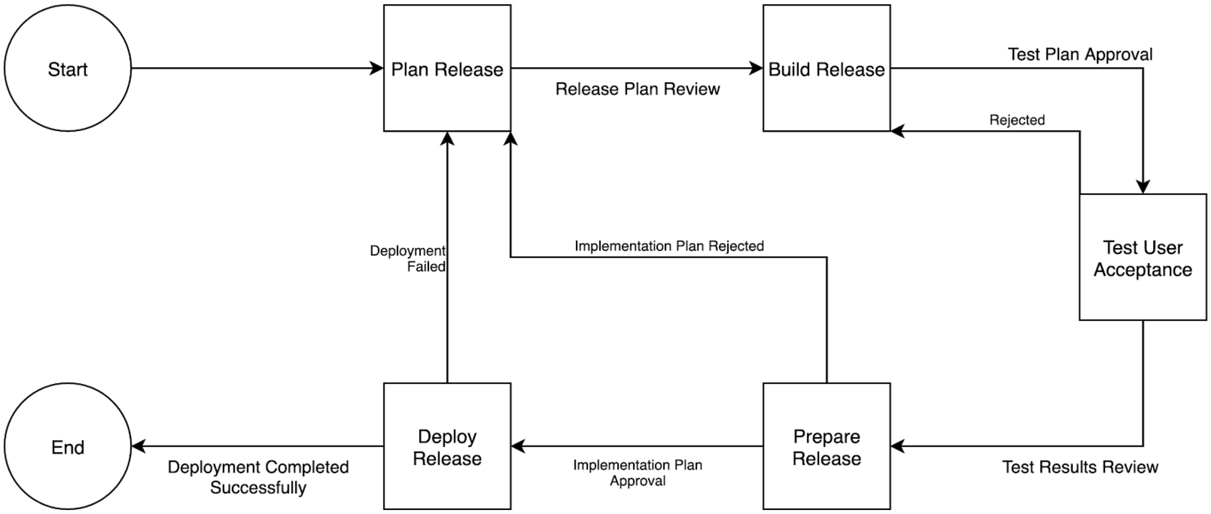


Figure 3. Release Management

13. Contingency plan

Procedures to be executed in case of data loss or compromise

- Separate the development database and the production database
- Backup the production database every two weeks
- Before updating new version to production server, backup the production database
- Try to make clear and nice description of comments and pull requests, so developers can easy to find the cause of data loss or compromise
- Comments all changes and push to own branch daily
- Make logs for every database updates and save logs separate from the database
- When data loss or compromise, developers can try to rollback to the latest and working backup and update the backup by logs.

Requirements Specification

1. High-Level Goals (Unique Identifiers)

G1: Deliver a sufficient way to use the library for university staff and students.

G2: Provide efficient methods to manage library activity.

2. Primary and Secondary Actors

Primary Actors: user (student, staff, professor)

Secondary Actors: library system, library staff

3. Use Case diagram (s)

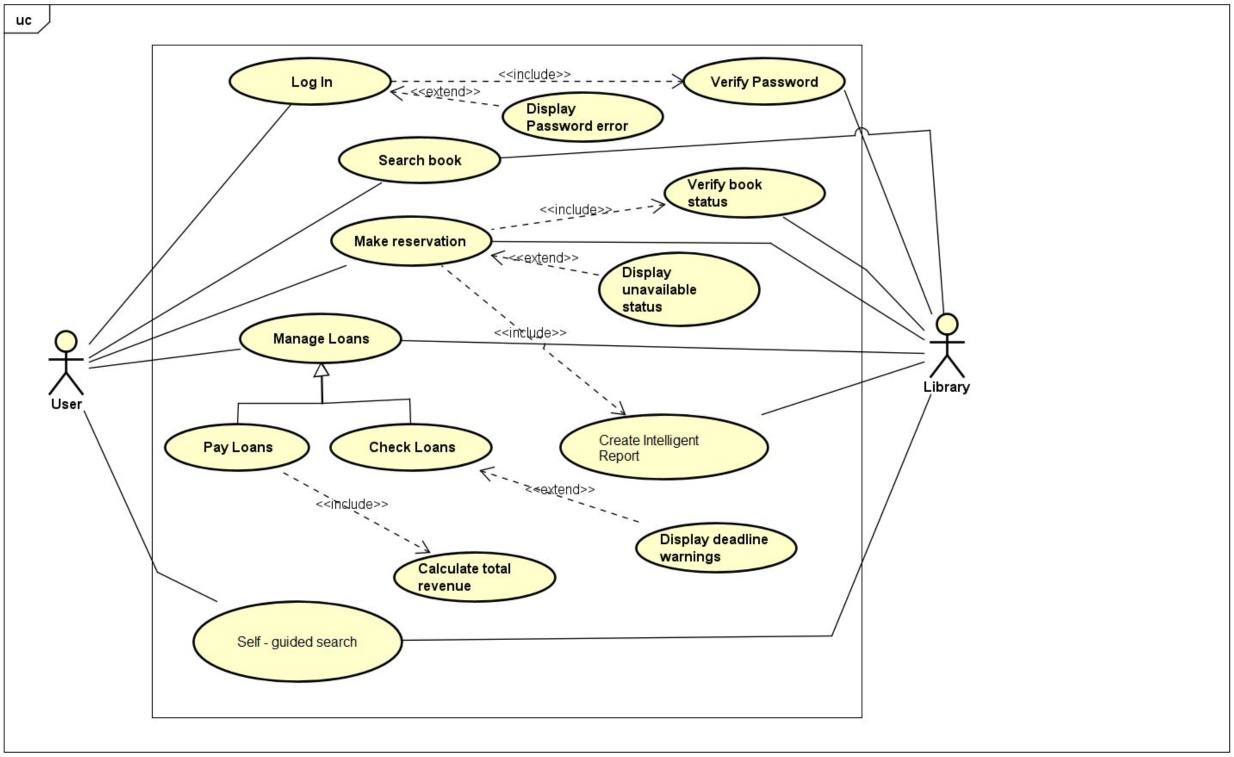


Figure 1. User Case Diagram

4. Use Case Specification (s)

Table 1. Use Case: Log In

Use Case: Log In
Primary Actor: User (student, professor, staff)
Secondary Actor: None

Precondition: 1. Actor has university email and clicked on button LOG IN
Postcondition: 1. Actor successfully logged into the system or an error will be displayed if the input password is wrong.
Main Success Scenario: 1. The app's "Log In" screen loads up 2. User types in their username / email and password 3. Actor selects on LOGIN button in the bank app 4. App starts to verify account and password 5. After successfully verified, actor allow to navigate into the home screen of the app
Alternate Course:
Exception Course: 4. App shows an error message when the username and password doesn't match with the database. Resumes at step 1

Table 2. Use Case: Verify Password

Use Case: Verify Password
Primary Actor: Staff
Secondary Actor: Library
Precondition: Credentials are submitted by the staff
Postcondition: Authentication is completed
Main Success Scenario: 1. Staff clicks the button LOGIN 2. Library system encrypts the informed credentials and compared them to the ones stored in the DB

3. A Boolean message with True is produced if the credentials match
Alternate Course: 3a – A Boolean message with False is produced if the credentials do not match
Exception Course:

Table 3. Use Case: Display Login Error

Use Case: Display Login Error
Primary Actor: Staff
Secondary Actor: None
Precondition: Credentials were verified
Postcondition: A login error message is displayed
Main Success Scenario: 1. A Boolean message with False is produced by UC Verify Password 2. A login error message is generated 3. A login error message is displayed
Alternate Course:
Exception Course:

Table 4. Use Case: Search Book

Use Case: Search Book
Primary Actor: User
Secondary Actor: Library
Precondition: Actor logged into the system
Postcondition: Redirect to the page that returns the search result.
Main Success Scenario:

1. Actor clicks on the search box 2. Actor types the title of the book or article 3. Actor presses on the search button or “ENTER” 4. Library system redirect to the search result page
Alternate Course: 3a – A Boolean message with False is produced if the book or article is not available
Exception Course:

Table 5. Use Case: Make Reservation

Use Case: Make Reservation
Primary Actor: User
Secondary Actor: Library
Precondition: Actor is redirected to the search result pages.
Postcondition: Actor successfully reserves one or many books
Main Success Scenario: 1. Actor chooses the book they want 2. Actor clicks on button MAKE RESERVATION 3. Include UC verify book status 4. A book status success message is displayed 5. Actor successfully reserves one or many books
Alternate Course:
Exception Course: 5a – Extend UC Display Login Error when book is not available. Resumes at step 1.

Table 6. Use Case: Verify Book Status

Use Case: Verify Book Status
Primary Actor: User
Secondary Actor: Library system

Precondition: Book reservation has been made by user
Postcondition: Book status verification is completed
Main Success Scenario: 1 - Staff clicks the button "MAKE RESERVATION" 2 – Application encrypts the informed credentials and compared them to the ones stored in the DB 3 – A Boolean message with True is produced if the credentials match
Alternate Course: 3a – A Boolean message with False is the book is not available
Exception Course:

Table 7. Use Case: Display Unavailable Status

Use Case: Display Unavailable Status
Primary Actor: User
Secondary Actor: None
Precondition:
Postcondition:
Main Success Scenario: 1. A Boolean message with False is produced by Library system when book is not available 2. An unavailable error message is generated 3. An unavailable error message is displayed
Alternate Course:
Exception Course:

Table 8. Use Case: Create Intelligent Report

Use Case: Create Intelligent Report

Primary Actor: User
Secondary Actor: Library
Precondition: Actor has made some previous reservations before.
Postcondition: Display the reservation report
Main Success Scenario: 1. Go to actor's reservations. 2. Actor click on button CREATE REPORT 3. Library system starts to gather data 4. Library displays the report
Alternate Course:
Exception Course:

Table 9. Use Case: Pay Loans

Use Case: Pay Loans
Primary Actor: User
Secondary Actor: Library
Precondition: Actor logged into the system and has made some previous reservations before
Postcondition: User finish paying their loans
Main Success Scenario: 1. Actor clicks on MANAGE LOANS in the menu 2. Library system redirected to Manage Loans screen 3. Actor clicks on PAY LOANS 4. Include UC Calculate total revenue 5. Actor confirms that they successfully pay for loans
Alternate Course: 3a. Actor is informed of the total revenue that they need to pay. Resume at step 4.
Exception Course:

Table 10. Use Case: Check Loans

Use Case: Check Loans
Primary Actor: User
Secondary Actor: Library
Precondition: Actor logged into the system and has made some previous reservations before
Postcondition: Actor knows about their current reservations' deadline
Main Success Scenario: 1. Actor clicks on CHECK LOANS in the menu 2. Library system redirected to Check Loans screen 3. Actor clicks on CHECK LOANS 4. Include UC Display deadline warnings
Alternate Course:
Exception Course:

Table 11. Use Case: Display Deadline Warnings

Use Case: Display Deadline Warnings
Primary Actor: User
Secondary Actor: Library
Precondition: Actor clicks on CHECK LOANS in MANAGE LOANS section
Postcondition: UI display to show all the deadline for reservations
Main Success Scenario: 1. Library system starts to collect data inside the DB 2. Waiting screen is initiated while waiting 3. Display the deadlines to actor
Alternate Course:
Exception Course:

Table 12. Use Case: Calculate Total Revenue

Use Case: Calculate Total Revenue
Primary Actor: User
Secondary Actor: Library
Precondition: Actor is navigated to PAY LOANS screen
Postcondition: Actor knows about their loan that new need to pay
Main Success Scenario: 1. Library system starts to collect data inside the DB 2. Library system apply formula to calculate the loan base on each user and deadline 3. Display the amount of loan user has to pay
Alternate Course:
Exception Course:

Table 13. Use Case: Self-guided Search

Use Case: Self-guided Search
Primary Actor: User
Secondary Actor: Library
Precondition: Actor logged into the system and press on BIBLIOGRAPHIC RESEARCH
Postcondition: The systems show the research to actor
Main Success Scenario: 1. Actor inputs values on the bibliographic research 2. Library system gathered information and searched on DB 3. Library system returns the data to actor
Alternate Course:
Exception Course:

5. Assumptions

5.1 Credentials

- User should be a member of the University of Wonderland (UWON).
- User should have an active university email to register to the UWON library system.

5.2 Security

- No user can take out the book without reservation.
- No user can make a reservation without a loan.

6. Domain Properties

- A book with an unique Book ID can not be borrowed by different users at the same time.
- A book can not be available and borrowed at the same time.
- A reservation can not exist with a loan.

7. Functional Requirements

People Registration:

REQ_1 - User Registration

Rationale: A USER is identified by User ID, and has the attributes Name, Type, Price, and Email.

REQ_2 - Library Staff Registration

Rationale: A LIBRARY STAFF is identified by Staff ID and has an attributes Name.

Library Registration:

REQ_3 - Book Registration

Rationale: A BOOK is identified by Book ID, and has the attributes Book Title, ISBN, Condition, and the calculated attributes Number of Copies, and Available Status.

REQ_4 - Reservation Registration

Rationale: A RESERVATION is identified by Reservation ID, and has the attributes Reserve Date, Reservation Status, and Return Status.

Fund Registration:

REQ_5 - Loan Registration

Rationale: A LOAN is identified by Loan ID, and has an attribute Loan Period, Number of Renew and the calculated attributes Fee, Fine, and Deadline Warming.

People Action Relationship:

REQ_UL_1 - Relationship between User and Loan

Rationale: Each USER can make at most 3 LOANS, and each LOAN must be made by only one USER.

REQ_UR_1 - Relationship between User and Reservation

Rationale: Each USER can have many RESERVATIONS, and each RESERVATION must belong to only one USER.

REQ_UB_1 - Relationship between User and Book

Each USER can use the self-guided tool to search many BOOKS, and each BOOK can be searched by the self-guided tool for many USERS. Each search by the self-guided tool has the attributes name, subject, ISBN, publisher, and author which inform searching properties of the BOOK.

REQ_LSR_1 - Relationship between Library Staff and Reservation

Rationale: Each LIBRARY STAFF can manage many RESERVATIONS, and each RESERVATION can be managed by many LIBRARY STAFF.

REQ_LSR_2 - Relationship between Library Staff and Reservation

Rationale: Each LIBRARY STAFF can get intelligent reports from many RESERVATIONS, each RESERVATION can generate an intelligent report for many LIBRARY STAFF. Each intelligent report has a calculated attribute Number of Extra Copies Needed.

REQ_LSL_1 - Relationship between Library Staff and Loan

Rationale: Each LIBRARY STAFF can manage many LOANS, and each LOAN can be managed by many LIBRARY STAFF.

Library Management Relationship:

REQ_BR_1 - Relationship between Book and Reservation

Rationale: Each BOOK can be reserved by many RESERVATIONS, and each RESERVATION must reserve a single BOOK.

REQ_RL_1 - Relationship between Reservation and Loan

Rationale: Each RESERVATION must belong to only one LOAN, and each LOAN must have only one RESERVATION.

8. Non-Functional Requirements

8.1 Product requirements:

NON_REQ_PR_1 - Usability Requirement

Rationale:

- The system shall allow the users to access the system from Library Computer User interface. The system uses a Swing GUI client as an interface. Since all users familiar with the usage of UI, no more training is required
- A user should be a member of the University of Wonderland(UWON) and has an active university email which can be used to register a library account.
- The system is user friendly and online help makes using the system easy and also the product will support multiple languages such as Madarin, Japanese, Korean and English.
- The system shall generate bibliographical results when users conduct the self-guided search.
- The system shall generate an Intelligent report when librarians send a request to the system.
- The users borrow the books or copies at most 3 times simultaneously, which means users can make loans at most 3 times. Users shall return books or copies on time otherwise the library system will issue a fine per day after the deadline.
- The system shall return a precise reservation status when users check the reservation status.
- The librarians will make a reservation and manage the loans for each users correctly.

NON_REQ_PR_2 - Availability Requirement

Rationale:

- The library system is available for all the departments and users and is used 24 hrs and 365 days a year. The system should be operational 24 hrs a day and 7 days a week.
- The information(Book and Copy, Reservation, Loan) is refreshed at regular intervals depending on whether some updates have occurred or not.
- The format for bibliographical queries and answers shall be accessible to students from any department, responses to these queries shall take less than 3 seconds
-

NON_REQ_PR_3 - Reliability Requirement

Rationale:

- The library system should accurately provide real time bibliographical search, reservation, book, and loan's information taking into consideration various concurrency issues. The system has to be 100% reliable and shall provide 100% access reliability.
- Mean Time to Repair - Even if the library system falls, the system will be recovered back up in an hour or less.
- Restarting the library system is acceptable when a failure event occurs
- Failure rate less than or equals 2%
- The system should send automatically a warning deadline email to users about book

reservation’s due date, and loan deadline warning.

NON_REQ_PR_4 - Performance Requirement
Rationale:

- The system shall respond to the user in less than 3 seconds from the time all kinds of requests such as bibliographical search, book reservation, intelligent report generation and etc are submitted. The responses to view results shall take no more than 4 seconds to appear on the UI page. The search result shall be complete and effective.
- The library system shall support 20000 concurrent users and should be able to handle a large amount of data. Thus it shall accommodate a high number of books and users without fault.
- The library system shall handle expected and unexpected errors in ways that prevent loss in information and downtime period.
- The library system has strong traceability to previous borrowers when books are found to be damaged.

NON_REQ_PR_5 - Supportability(including maintainability and portability
Rationale:

- Any changes (new user registration, database changes, reservation change, book database) must be verified once per day at least by library staff. The system should provide automatic notification email to users about book’s overdue, and loan deadline warning.
- The library system shall be ported in 64-bit linux operating system and Android operating system.
- The library system shall be maintained by library staff and shall be updated once a month

8.2 Organizational requirements:

NON_REQ_QR_1 - Implementation Requirements
Rationale:

Deploying and commissioning the library system shall take one month or less, and the implementation involves library system installation, maintenance, library staff and user training. Ability to deal with any department library system branch, users, user categories, library staff, book categories, books, loans and other data. Easy to use search and tracking policies, strong user management, intuitive navigation, and extensive administrative permission for staff accounts.

NON_REQ_QR_2 - Software Constraints
Rationale:

- The quality of the database is maintained in such a way that it can be very user friendly to all the users of the database. The library databases shall be MySQL.
- To develop the web server of the library system we will use Java and Java Swing GUI to design UI pages. Other related software associated with the library system shall be written by using Java, to comply with University of Wonderland(UWON) library’s

policy.

NON_REQ_QR_3 - Hardware Constraints
Rationale:

The system requires a database in order to store persistent data. The database should have backup capabilities.

8.3 External requirements:

NON_REQ_ER_2 - Security Requirements
Rationale:

- The databases may get crashed at any certain time due to virus or operating system failures. Therefore the database shall be backed up by library staff and it shall be recovered within fifteen minutes when the failure happened.
- There are different categories of users namely library staff, members(student, faulty, professor), administrator etc. Depending on the category of user the access rights are different. If the user is a staff or administrator then he is able to modify the data, delete, reserve books and manage loans for members. All the other users other than library staff and administrators only have the rights to retrieve the information about the database.
- All system data must be backed up every 24 hours and the backup copies stored in a secure location which is not in the same building as the system.

NON_REQ_ER_3 - Legal Requirements
Rationale:

User information shall be protected. The development process to be used must be explicitly defined and must be conformant with ISO 9000 standards.

9. Traceability Matrix

Table 14: Traceability Matrix

Requirement	Design	Implementation	Test Case
REQ_1	User	SCREEN_01	TS_01, TS_02, UT_01, UT_02, UT_03, UT_04, UT_05, UT_06, UT_07, UT_08, UT_09, IT_01, IT_02, IT_03, IT_04, IT_05, ST_01, ST_02, ST_03, ST_04
REQ_2	LibraryStaff	SCREEN_01	TS_01, UT_01, UT_02, UT_03, UT_04, UT_05, UT_06, IT_01, IT_02, IT_03, ST_01, ST_02

REQ_3	Book	SCREEN_02	UT_07, UT_08, UT_09, IT_04, IT_05, ST_03, ST_04
REQ_4	Reservation	SCREEN_02	
REQ_5	Loan	SCREEN_02	
REQ_UB_1	SelfGuidedSearch	SCREEN_03	UT_07, UT_08, UT_09, IT_04, IT_05, ST_03, ST_04
REQ_LSR_1	IntelligentReport	SCREEN_04	

Design

1. Entity Relationship Diagram (conceptual model)

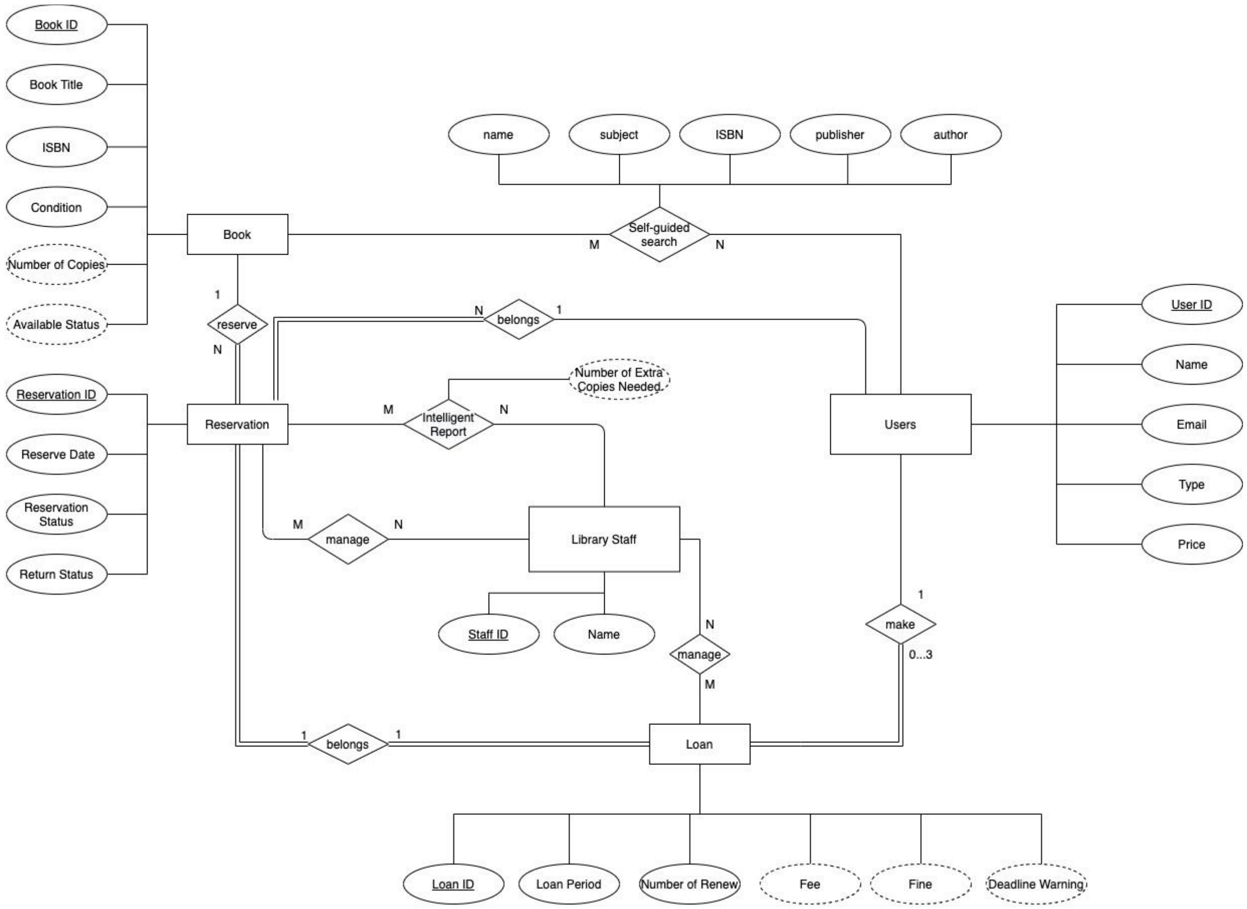


Figure 1. ER Diagram (conceptual model)

2. Logical Model

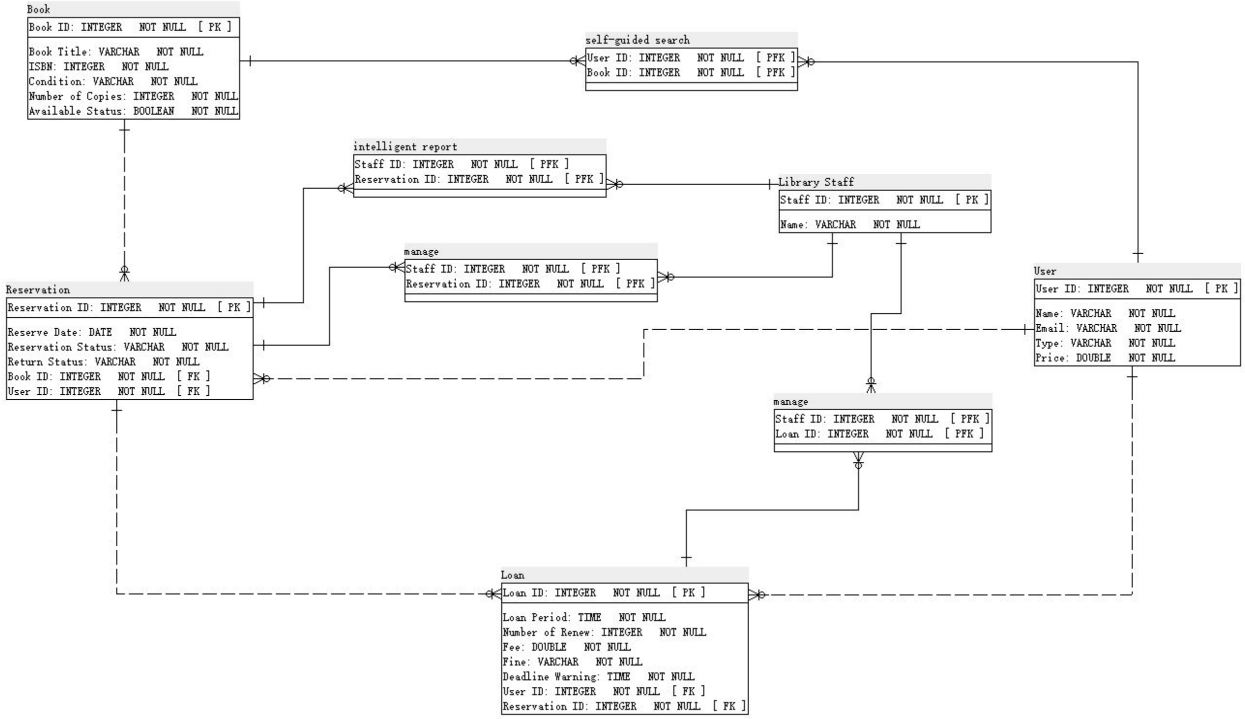

Book
Book ID: INTEGER NOT NULL [PK]
Book Title: VARCHAR NOT NULL
ISBN: INTEGER NOT NULL
Condition: VARCHAR NOT NULL
Number of Copies: INTEGER NOT NULL
Available Status: BOOLEAN NOT NULL
self-guided search
User ID: INTEGER NOT NULL [PFK]
Book ID: INTEGER NOT NULL [PFK]
intelligent report
Staff ID: INTEGER NOT NULL [PFK]
Reservation ID: INTEGER NOT NULL [PFK]
Library Staff
Staff ID: INTEGER NOT NULL [PK]
Name: VARCHAR NOT NULL
manage
Staff ID: INTEGER NOT NULL [PFK]
Reservation ID: INTEGER NOT NULL [PFK]
User
User ID: INTEGER NOT NULL [PK]
Name: VARCHAR NOT NULL
Email: VARCHAR NOT NULL
Type: VARCHAR NOT NULL
Price: DOUBLE NOT NULL
Reservation
Reservation ID: INTEGER NOT NULL [PK]
Reserve Date: DATE NOT NULL
Reservation Status: VARCHAR NOT NULL
Return Status: VARCHAR NOT NULL
Book ID: INTEGER NOT NULL [FK]
User ID: INTEGER NOT NULL [FK]
manage
Staff ID: INTEGER NOT NULL [PFK]
Loan ID: INTEGER NOT NULL [PFK]
Loan
Loan ID: INTEGER NOT NULL [PK]
Loan Period: TIME NOT NULL
Number of Renew: INTEGER NOT NULL
Fee: DOUBLE NOT NULL
Fine: VARCHAR NOT NULL
Deadline Warning: TIME NOT NULL
User ID: INTEGER NOT NULL [FK]
Reservation ID: INTEGER NOT NULL [FK]

Figure 2. Logical Model

3. Class Diagram

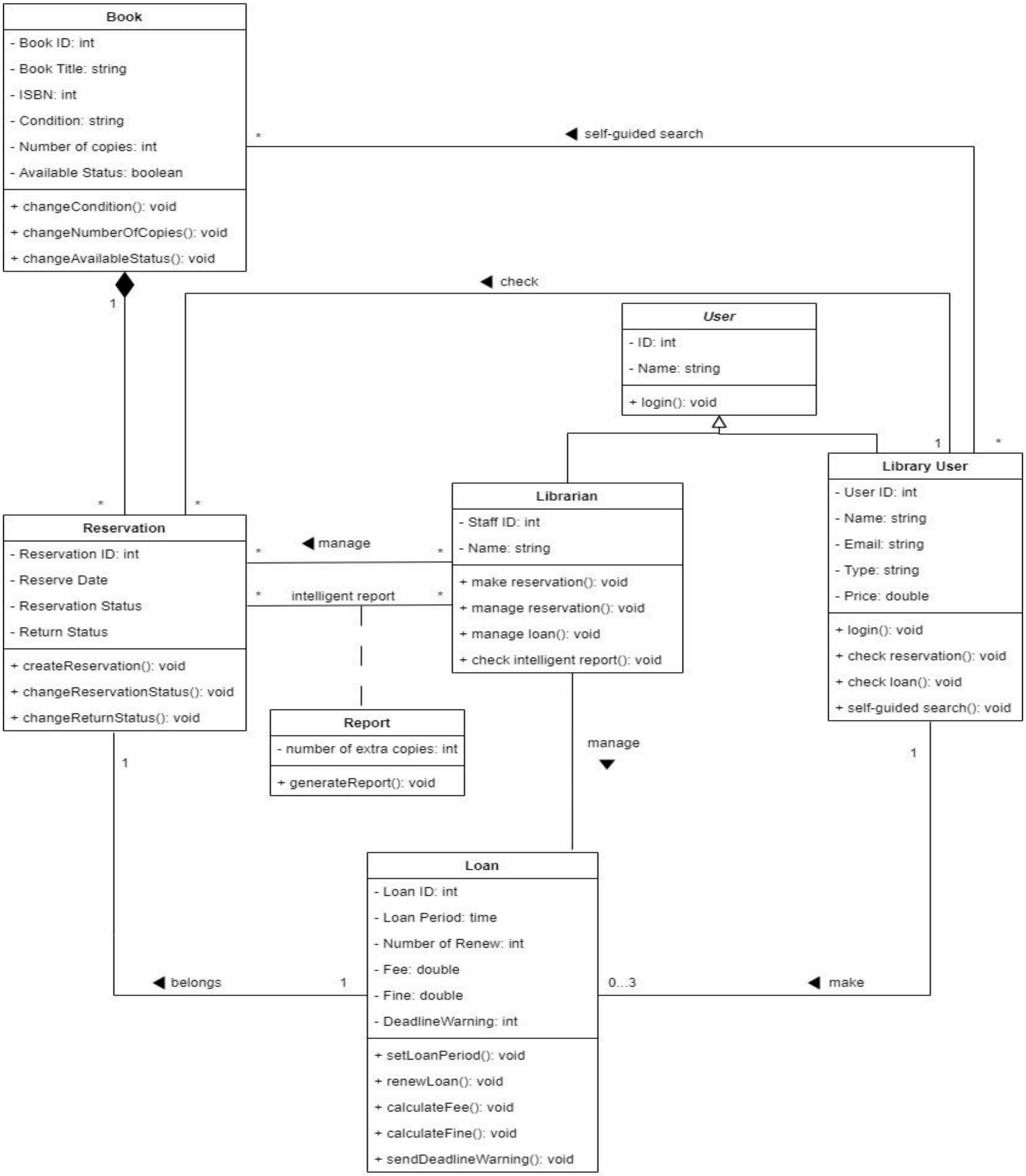


Figure 3. Class Diagram

4. Sequence Diagram

4.1 Make a Book Reservation

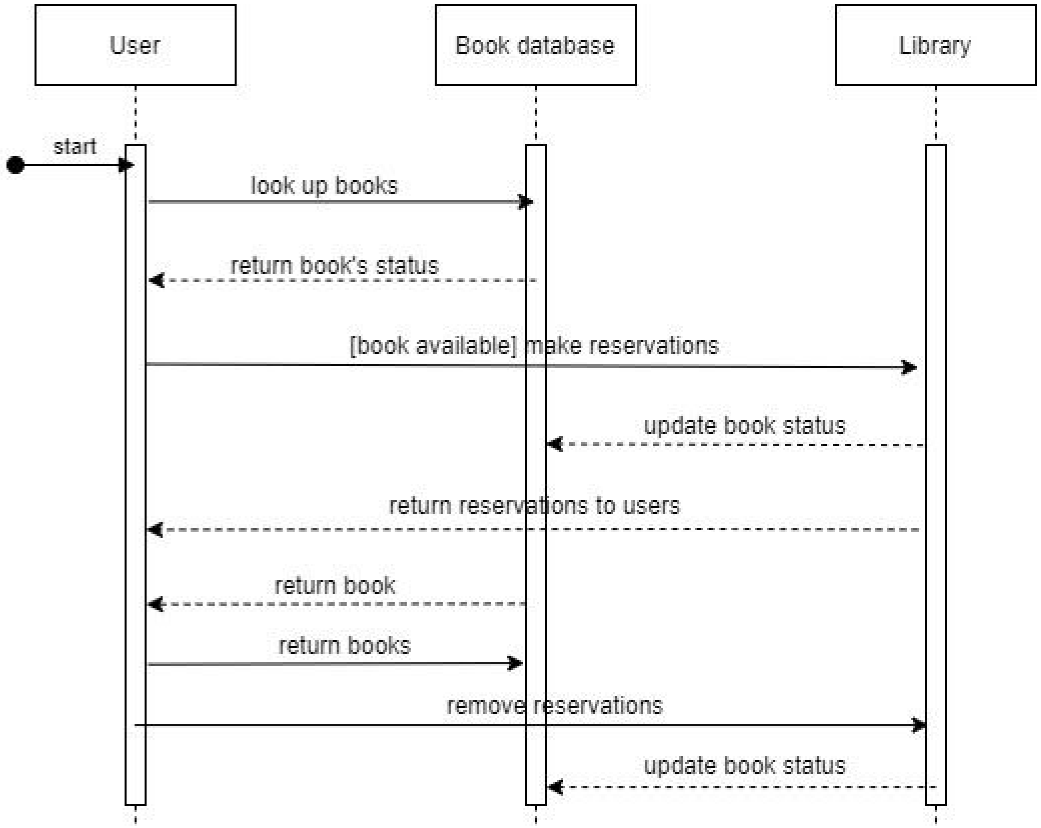


Figure 4. Sequence Diagram (Make a Book Reservation)

4.2 Manage Loan

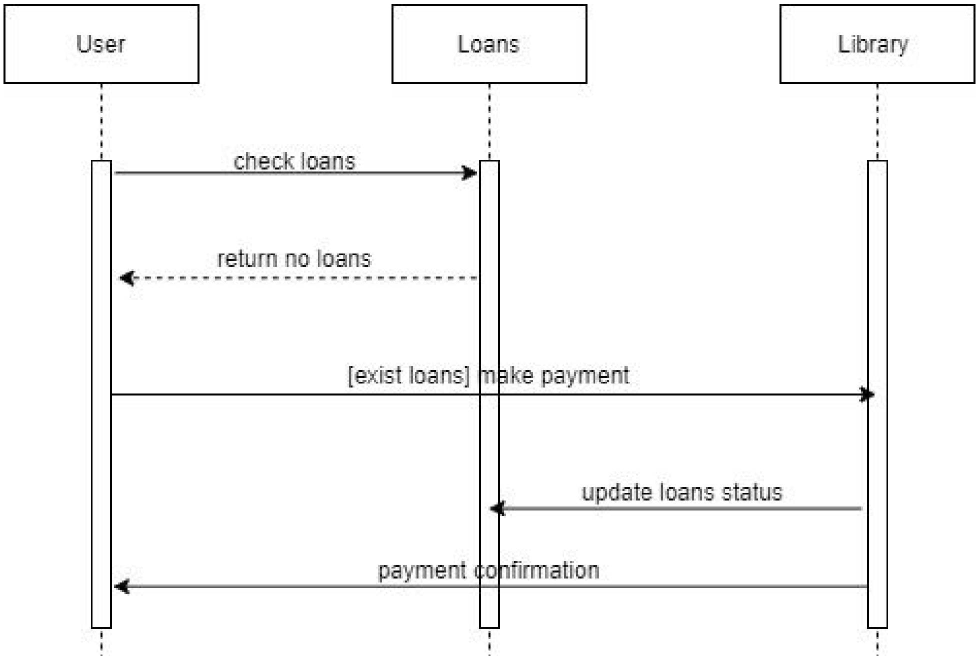


Figure 5. Sequence Diagram (Manage Loan)

5. State Machine Diagram

5.1 Make a Book Reservation

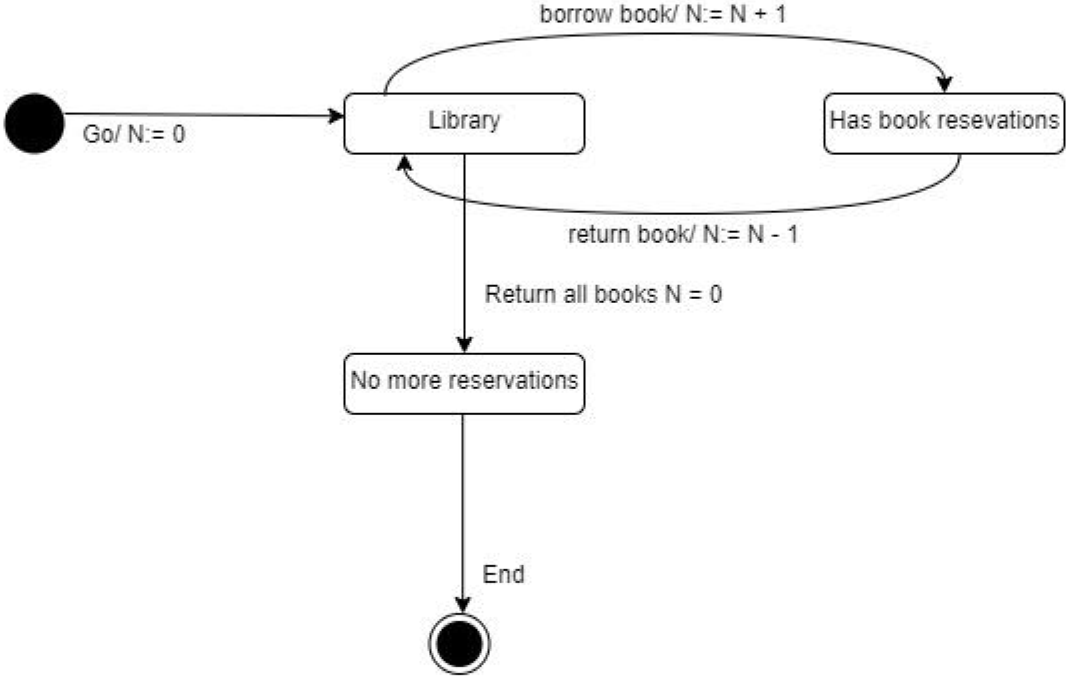


Figure 6. State Machine Diagram (Make a Book Reservation)

5.2 Manage Loan

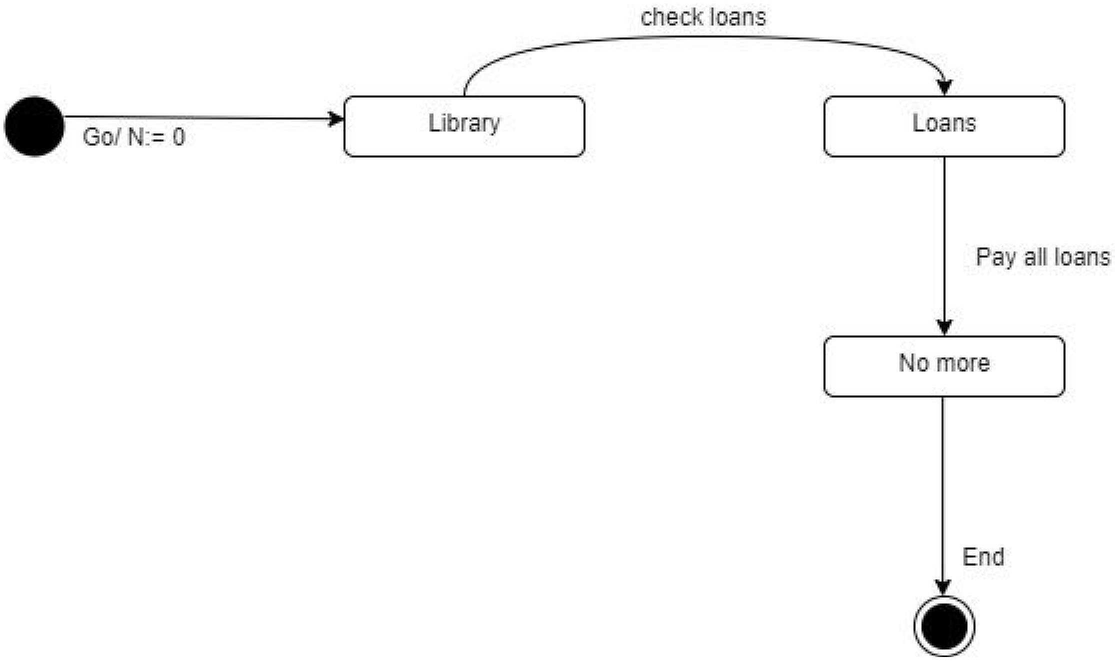


Figure 7. State Machine Diagram (Manage Loan)

6. System Architecture View and Style/Pattern

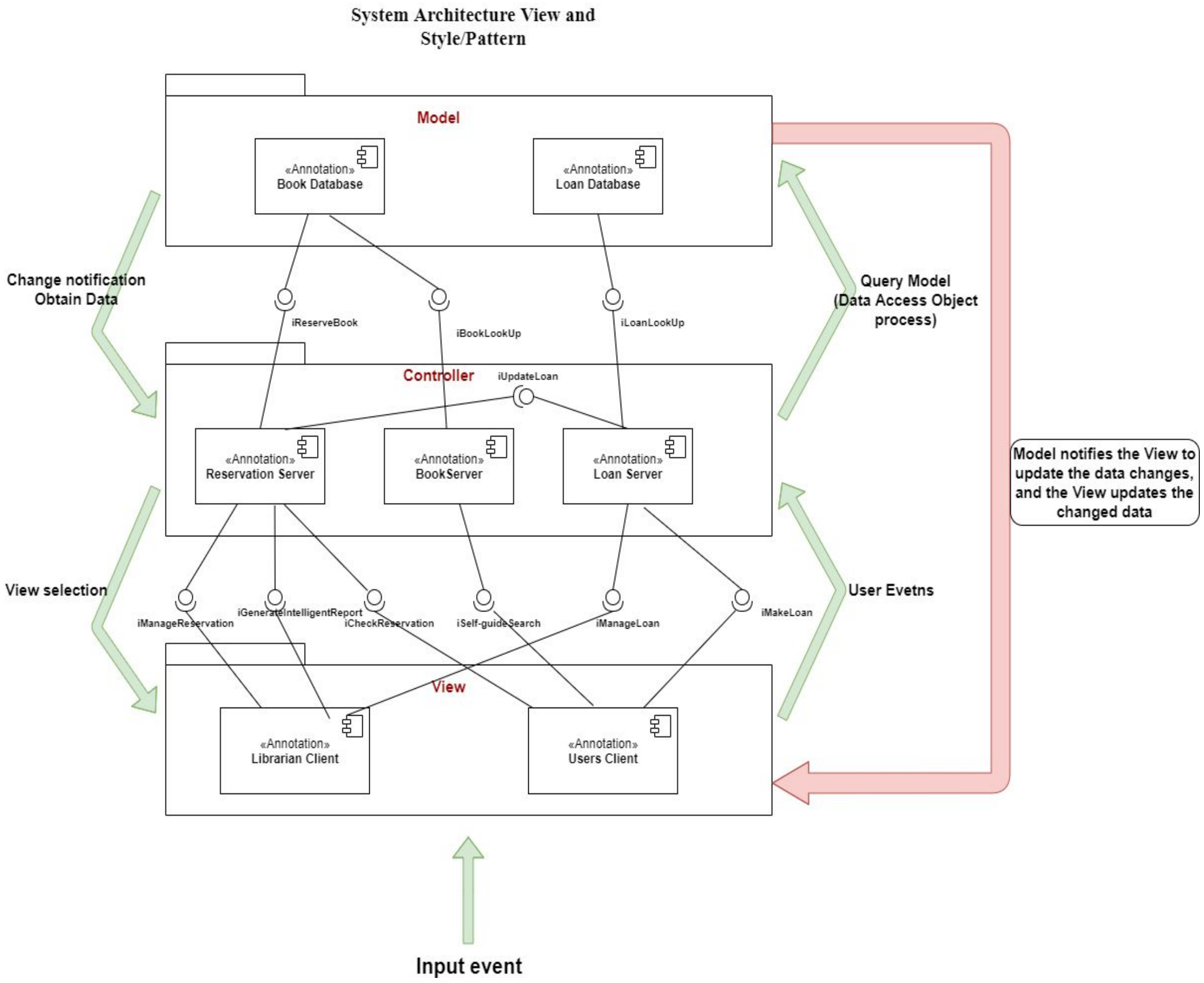

System Architecture View and Style/Pattern
Model
«Annotation» Book Database
«Annotation» Loan Database
Change notification Obtain Data
Query Model (Data Access Object process)
iReserveBook
iBookLookUp
iLoanLookUp
Controller
iUpdateLoan
«Annotation» Reservation Server
«Annotation» BookServer
«Annotation» Loan Server
Model notifies the View to update the data changes, and the View updates the changed data
View selection
User Evetns
iManageReservation
iGenerateIntelligentReport
iCheckReservation
iSelf-guideSearch
iManageLoan
iMakeLoan
View
«Annotation» Librarian Client
«Annotation» Users Client
Input event

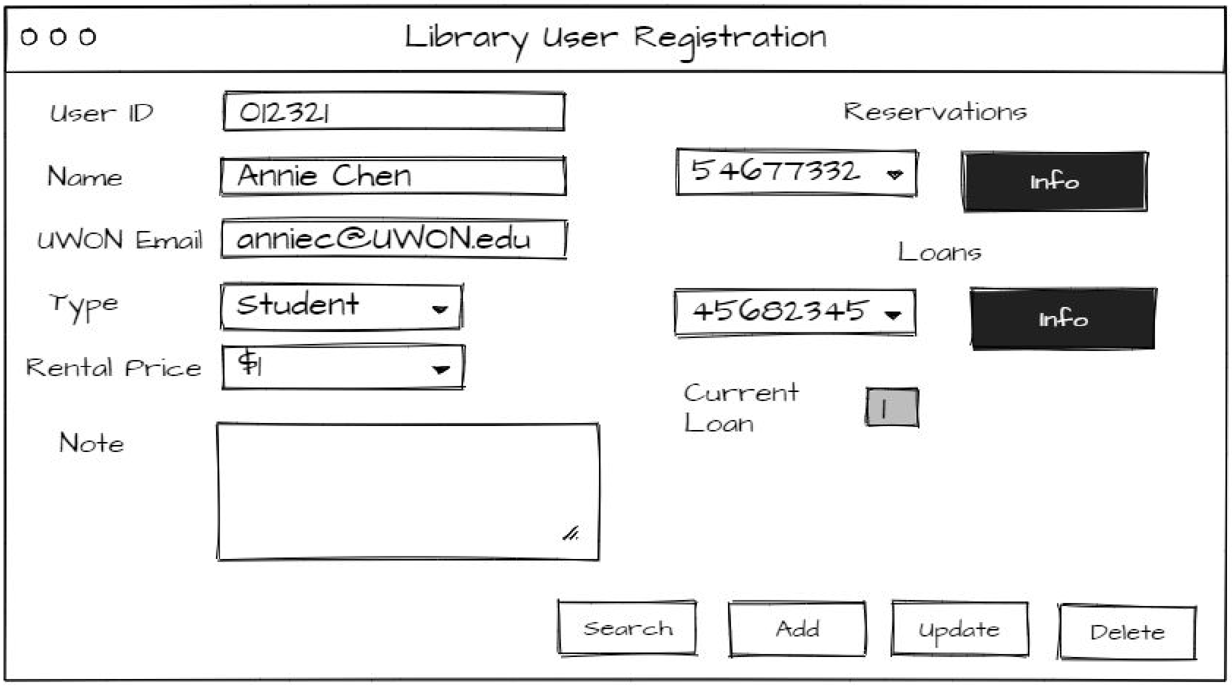
Figure 8. MVC architecture Pattern

We will use MVC System Architecture Pattern to design this project, as we know, **the Model layer** is business data, **View layer** is user interface and **Controller** is used to handle dispatcher events. When the View layer accepts a user interface request(input event) such as self-guided search, makes a loan etc, it will transfer requests to Controller, and Controller will operate Model layer and manipulate the database such as **Add**, **Delete**, **Update**, **Search** functions. Finally, the Model layer will notify the View layer to update the data changes, and View updates changed data.

Paper-based Prototype

1. Technical Description of the System

SCREEN_01



Library User Registration

User ID: 012321

Name: Annie Chen

UWON Email: anniec@UWON.edu

Type: Student

Rental Price: $1

Note:

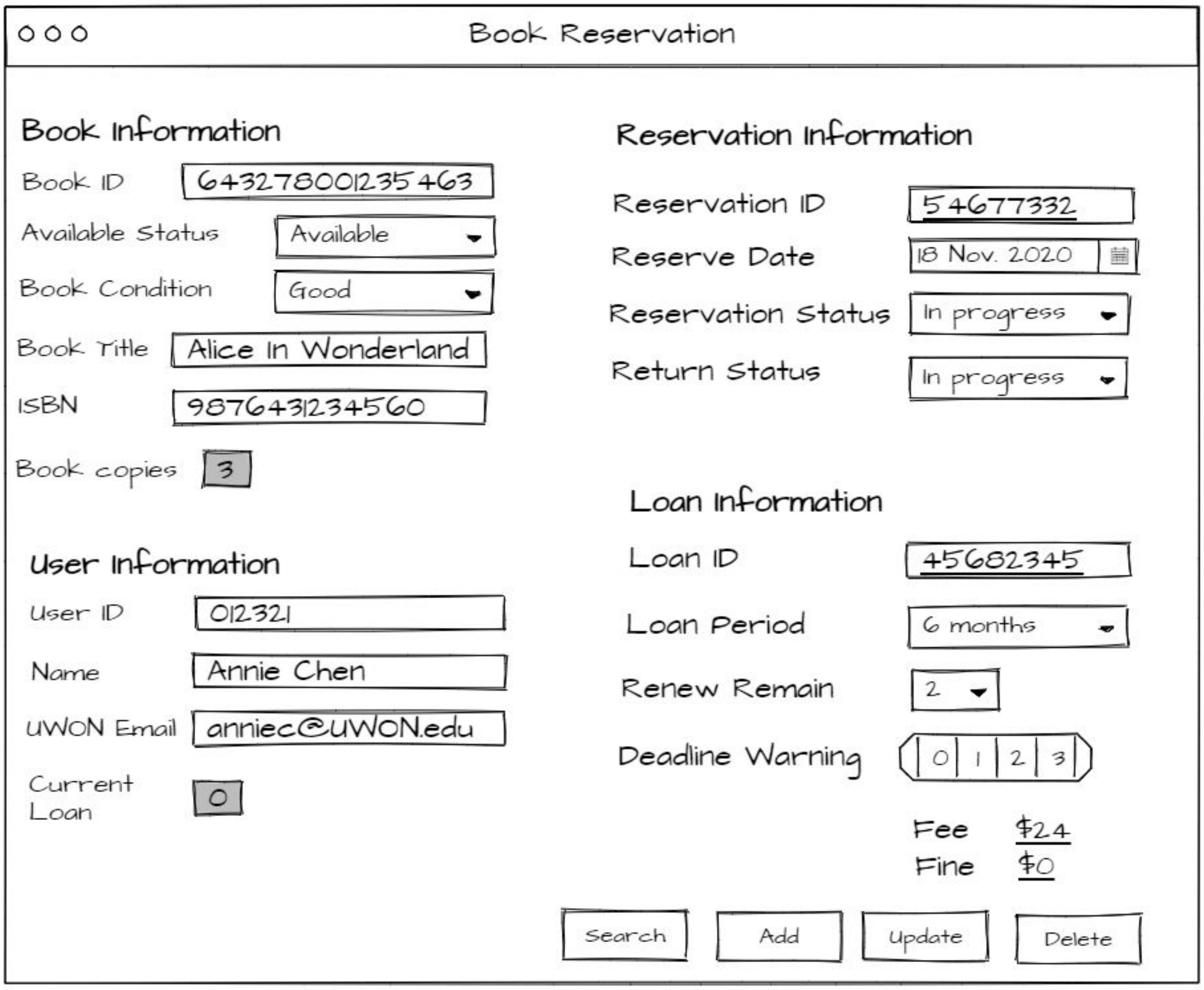
Reservations: 54677332

Loans: 45682345

Current Loan: 1

Figure 1. SCREEN_01: Library User Registration

SCREEN_02



Book Reservation

Book Information

Book ID: 64327800I235463

Available Status: Available

Book Condition: Good

Book Title: Alice In Wonderland

ISBN: 9876431234560

Book copies: 3

User Information

User ID: 012321

Name: Annie Chen

UWON Email: anniec@UWON.edu

Current Loan: 0

Reservation Information

Reservation ID: 54677332

Reserve Date: 18 Nov. 2020

Reservation Status: In progress

Return Status: In progress

Loan Information

Loan ID: 45682345

Loan Period: 6 months

Renew Remain: 2

Deadline Warning: 0 | 1 | 2 | 3

Fee $24

Fine $0

Figure 2. SCREEN_02: Book Reservation

SCREEN_03

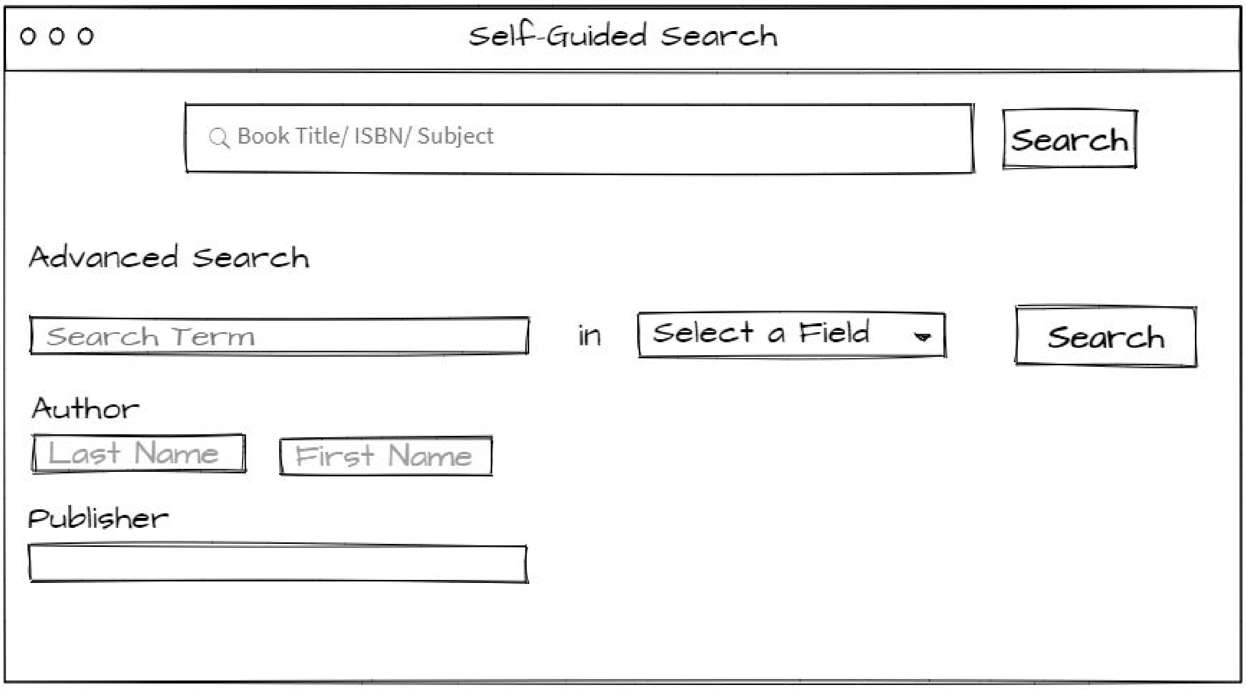
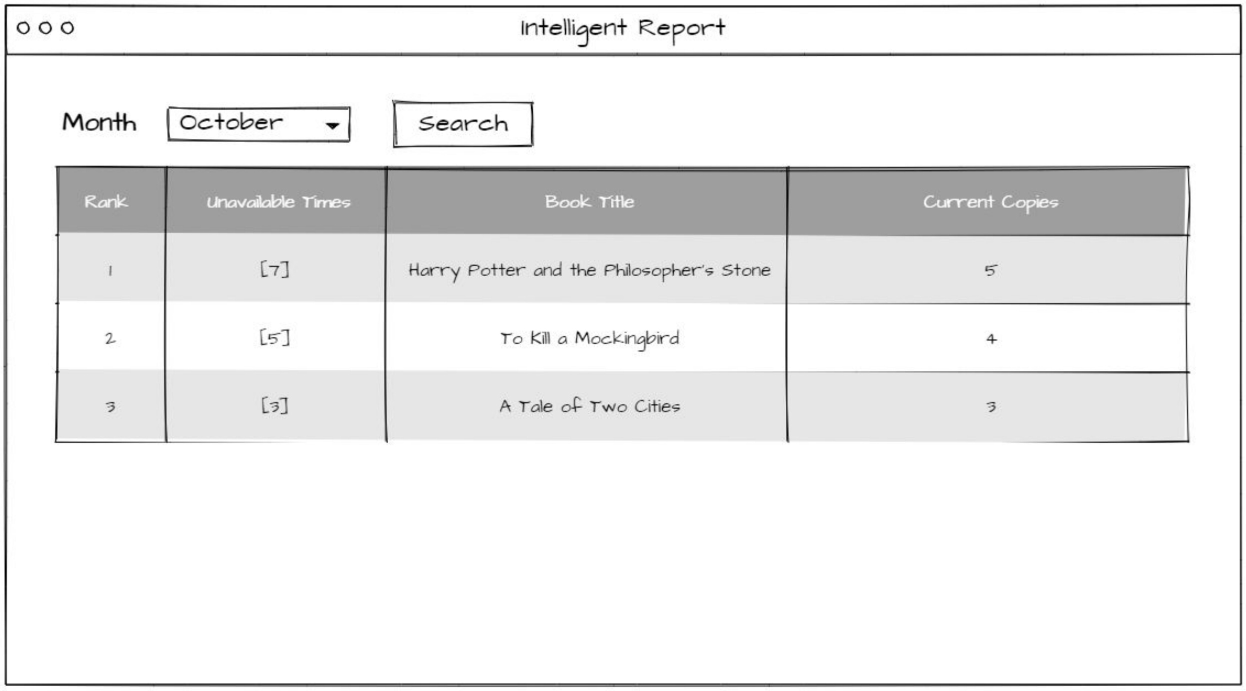


Figure 3. SCREEN_03: Self-guided Search

SCREEN_04



Month October

Rank	Unavailable Times	Book Title	Current Copies
1	[7]	Harry Potter and the Philosopher's Stone	5
2	[5]	To Kill a Mockingbird	4
3	[3]	A Tale of Two Cities	3

Figure 4. SCREEN_04: Intelligent Report

Testing Plan

1. Test Scenarios

Table 1: Test Scenarios

Test Scenarios		
Scenario ID	Title	Description
TS_01	Log In	Users are able to login to the system by user's university email and password on the login screen, and then the application redirects to the next screen based on the type of the user.
TS_02	Search Books	Library users are able to search books by name, subject, ISBN, publisher, or author of the book on the self-guided search screen.

2. Unit Tests

Table 2: UT_01

Test Scenario ID	TS_01
Test Scenario	Log In
Test Case ID	UT_01
Test Case	Login email format is invalid
Pre-conditions	1. The "Log In" screen loaded up. 2. Email field and password is empty. 3. The "Login" button is disabled.
Instructions (Steps)	1. Enter a random string which does not match email format on the email field
Expected Outputs	1. The email field turns red, and an error message "email is invalid" displays under the email field. 2. The "Login" button is disabled.
Type of Implementation	MANUAL
Phase	Implementation
Dependencies	
Requirement Tested	REQ_1, REQ_2

Table 3: UT_02

Test Scenario ID	TS_01
Test Scenario	Log In
Test Case ID	UT_02
Test Case	Login password length is too short.
Pre-conditions	1. The “Log In” screen loaded up. 2. Email field and password are empty. 3. The “Login” button is disabled.
Instructions (Steps)	1. Enter a random string which the length is less than 8 on the password field
Expected Outputs	1. The password field turns red, and an error message “password is too short” displays under the password field. 2. The “Login” button is disabled.
Type of Implementation	MANUAL
Phase	Implementation
Dependencies	
Requirement Tested	REQ_1, REQ_2

Table 4: UT_03

Test Scenario ID	TS_01
Test Scenario	Log In
Test Case ID	UT_03
Test Case	Login password length is too long.
Pre-conditions	1. The “Log In” screen loaded up. 2. Email field and password are empty. 3. The “Login” button is disabled.
Instructions (Steps)	1. Enter a random string which the length is more than 13 on the password field
Expected Outputs	1. The password field turns red, and an error message “password is too long” displays under the password field. 2. The “Login” button is disabled.
Type of Implementation	MANUAL

Phase	Implementation
Dependencies	
Requirement Tested	REQ_1, REQ_2

Table 5: UT_04

Test Scenario ID	TS_01
Test Scenario	Log In
Test Case ID	UT_04
Test Case	Login email and password are valid
Pre-conditions	1. The “Log In” screen loaded up. 2. Email field and password are empty. 3. The “Login” button is disabled.
Instructions (Steps)	1. Enter an university email on the email field 2. Enter a password on the password field
Expected Outputs	1. The “Login” button is enabled.
Type of Implementation	MANUAL
Phase	Implementation
Dependencies	
Requirement Tested	REQ_1, REQ_2

Table 6: UT_05

Test Scenario ID	TS_01
Test Scenario	Log In
Test Case ID	UT_05
Test Case	When the login success message is displayed, clicking the “OK” button can trigger the *redirect()* function.
Pre-conditions	1. The login success message is displayed with a “OK” button. 2. Mock the *redirect()* function which is triggered by clicking the “OK” button to print “redirecting...” on the console
Instructions (Steps)	1. Click the “OK” button
Expected Outputs	1. The message “redirecting…” is printed out on the console

Type of Implementation	MANUAL
Phase	Implementation
Dependencies	IT_02
Requirement Tested	REQ_1, REQ_2

Table 7: UT_06

Test Scenario ID	TS_01
Test Scenario	Log In
Test Case ID	UT_06
Test Case	When the login error message is displayed, clicking the “OK” button can close the error message.
Pre-conditions	1. The login error message is displayed with a “OK” button.
Instructions (Steps)	1. Click the “OK” button
Expected Outputs	1. The login error message disappears.
Type of Implementation	MANUAL
Phase	Implementation
Dependencies	IT_03
Requirement Tested	REQ_1, REQ_2

Table 8: UT_07

Test Scenario ID	TS_02
Test Scenario	Search Books
Test Case ID	UT_07
Test Case	A dropdown list of subject fields shows up after the “Select a Field” selection field is clicked.
Pre-conditions	1. Library user is logged in. 2. The self-guided search screen loaded up.
Instructions (Steps)	1. Click on the “Select a Field” selection field.
Expected Outputs	1. A dropdown list shows up and displays a list of subject fields such as “Computer Science”, “Biology”, “Physics”, and etc.

Type of Implementation	MANUAL
Phase	Implementation
Dependencies	ST_01
Requirement Tested	REQ_1, REQ_3, REQ_UB_1

Table 9: UT_08

Test Scenario ID	TS_02
Test Scenario	Search Books
Test Case ID	UT_08
Test Case	The "Select a Field" selection field displays the selection after the subject field is selected.
Pre-conditions	1. Library user is logged in. 2. The self-guided search screen loaded up.
Instructions (Steps)	1. Click on the "Select a Field" selection field. 2. Select "Computer Science" on the dropdown list.
Expected Outputs	1. The dropdown list disappears. 2. "Computer Science" shows on the "Select a Field" selection field under Advanced Search section
Type of Implementation	MANUAL
Phase	Implementation
Dependencies	UT_07, ST_01
Requirement Tested	REQ_1, REQ_3, REQ_UB_1

Table 10: UT_09

Test Scenario ID	TS_02
Test Scenario	Search Books
Test Case ID	UT_09
Test Case	All text fields are able to take inputs from user input and show on text fields.
Pre-conditions	1. Library user is logged in. 2. The self-guided search screen loaded up.

Instructions (Steps)	1. Enter “a” on the search bar 2. Enter “b” on the “Search Term” field 3. Enter “c” on the “Last Name” field under Author section 4. Enter “d” on the “First Name” field under Author section 5. Enter “e” on the text field under Publisher section
Expected Outputs	1. “a” shows on the search bar 2. “b” shows on the “Search Term” field 3. “c” shows on the “Last Name” field under Author section 4. “d” shows on the “First Name” field under Author section 5. “e” shows on the text field under Publisher section
Type of Implementation	MANUAL
Phase	Implementation
Dependencies	ST_01
Requirement Tested	REQ_1, REQ_3, REQ_UB_1

3. Integration Tests

Table 11: IT_01

Test Scenario ID	TS_01
Test Scenario	Log In
Test Case ID	IT_01
Test Case	The *login()* function can be triggered by the “Login” button.
Pre-conditions	1. Mocking the *login()* function takes email and password as arguments and return an array [email, password] 2. The “Log In” screen loaded up.
Instructions (Steps)	1. Enter an valid university email on the email field 2. Enter a valid password on the password field 3. Click the “login” button
Expected Outputs	The *login()* function is fired and returns an array as *result* which *result[0]* is equal to the value in the email field and *result[1]* is equal to the value in the password field.
Type of Implementation	MANUAL
Phase	Testing and Integration
Dependencies	UT_04
Requirement Tested	REQ_1, REQ_2

Table 12: IT_02

Test Scenario ID	TS_01
Test Scenario	Log In
Test Case ID	IT_02
Test Case	Login success message displays.
Pre-conditions	1. Mock the login api responds a mock login credential 2. The “Log In” screen loaded up.
Instructions (Steps)	1. Enter an university email on the email field 2. Enter a password on the password field 3. Click the “login” button
Expected Outputs	1. The application received the login credential from login api response and stored the login credential. 2. A success message is displayed with a “OK” button.
Type of Implementation	MANUAL
Phase	Testing and Integration
Dependencies	UT_04, IT_01
Requirement Tested	REQ_1, REQ_2

Table 13: IT_03

Test Scenario ID	TS_01
Test Scenario	Log In
Test Case ID	IT_03
Test Case	Login error message displays.
Pre-conditions	1. Mock the login api responds an error 2. The “Log In” screen loaded up.
Instructions (Steps)	1. Enter an university email on the email field 2. Enter a password on the password field 3. Click the “login” button
Expected Outputs	1. The application received the error response. 2. An error message is displayed with a “OK” button.
Type of Implementation	MANUAL

Phase	Testing and Integration
Dependencies	UT_04, IT_01
Requirement Tested	REQ_1, REQ_2

Table 14: IT_04

Test Scenario ID	TS_02
Test Scenario	Search Books
Test Case ID	IT_04
Test Case	The general search button which is behind the search bar can trigger the *generalSearch()* function with the value in the search bar.
Pre-conditions	1. Library user is logged in. 2. The self-guided search screen loaded up. 3. Mock *generalSearch()* function to take *searchStr* as argument and print the *searchStr* to console.
Instructions (Steps)	1. Enter “a” on the search bar 2. Enter “b” on the “Search Term” field 3. Selection “Computer Science” on the “Select a Field” selection field under Advanced Search section 4. Enter “c” on the “Last Name” field under Author section 5. Enter “d” on the “First Name” field under Author section 6. Enter “e” on the text field under Publisher section 7. Click the “Search” button after the search bar.
Expected Outputs	1. A message “a” is printed out on the console.
Type of Implementation	MANUAL
Phase	Testing and Integration
Dependencies	UT_08, UT_09, ST_01
Requirement Tested	REQ_1, REQ_3, REQ_UB_1

Table 15: IT_05

Test Scenario ID	TS_02
Test Scenario	Search Books
Test Case ID	IT_05
Test Case	The advanced search button which is under the Advanced Search section can trigger *advancedSearch()* function with values of all text fields and the

	selection field on the self-guided search screen.
Pre-conditions	1. Library user is logged in. 2. The self-guided search screen loaded up. 3. Mock *advancedSearch()* function to take *searchStr*, *subject*, *subjectField*, *lastName*, *firstName*, and *publisher* as arguments and print an array [searchStr, subject, subjectField, lastName, firstName, publisher] to console.
Instructions (Steps)	1. Enter “a” on search bar 2. Enter “b” on the “Search Term” field 3. Selection “Computer Science” on the “Select a Field” selection field under Advanced Search section 4. Enter “c” on the “Last Name” field under Author section 5. Enter “d” on the “First Name” field under Author section 6. Enter “e” on the text field under Publisher section 7. Click the “Search” button under the Advanced Search section.
Expected Outputs	1. A message [“a”, “b”, “Computer Science”, “c”, “d”, “e”] is printed out on the console.
Type of Implementation	MANUAL
Phase	Testing and Integration
Dependencies	UT_08, UT_09, ST_01
Requirement Tested	REQ_1, REQ_3, REQ_UB_1

4. System/Acceptance Tests

Table 16: ST_01

Test Scenario ID	TS_01
Test Scenario	Log In
Test Case ID	ST_01
Test Case	Successfully login by student
Pre-conditions	1. The “Log In” screen loaded up.
Instructions (Steps)	1. Enter an university email of a student on the email field 2. Enter a correct password on the password field 3. Click the “login” button 4. Click the “OK” button after the login success message is displayed
Expected Outputs	1. The application redirects to the self-guided search screen.

	2. The self-guided search screen loads up.
Type of Implementation	MANUAL
Phase	After Integration Test
Dependencies	UT_04, IT_01，IT_02
Requirement Tested	REQ_1, REQ_2

Table 17: ST_02

Test Scenario ID	TS_01
Test Scenario	Log In
Test Case ID	ST_02
Test Case	Successfully login by librarian
Pre-conditions	1. The “Log In” screen loaded up.
Instructions (Steps)	1. Enter an university email of a librarian on the email field 2. Enter a correct password on the password field 3. Click the “login” button 4. Click the “OK” button after the login success message is displayed
Expected Outputs	1. The application redirects to the librarian dashboard screen. 2. The librarian dashboard screen loads up.
Type of Implementation	MANUAL
Phase	After Integration Test
Dependencies	UT_04, IT_01, IT_02
Requirement Tested	REQ_1, REQ_2

Table 18: ST_03

Test Scenario ID	TS_02
Test Scenario	Search Books
Test Case ID	ST_03
Test Case	General Search
Pre-conditions	1. Library user is logged in. 2. The self-guided search screen loaded up.

Instructions (Steps)	1. Enter “Software Engineering” on the search bar 2. Click the “Search” button after the search bar.
Expected Outputs	1. The application redirects to the screen which displays the list of found books. 2. All titles of books on the screen include keywords “Software Engineering”.
Type of Implementation	MANUAL
Phase	After Integration Test
Dependencies	IT_04, ST_01
Requirement Tested	REQ_1, REQ_3, REQ_UB_1

Table 19: ST_04

Test Scenario ID	TS_02
Test Scenario	Search Books
Test Case ID	ST_04
Test Case	Advanced Search
Pre-conditions	1. Library user is logged in. 2. The self-guided search screen loaded up.
Instructions (Steps)	1. Enter “Software Engineering” on the search bar 2. Enter “Wiley” on the text field under the Publisher section. 3. Click the “Search” button under the Advanced Search section.
Expected Outputs	1. The application redirects to the screen which displays the list of found books. 2. All books on the screen have titles which include keywords “Software Engineering” and publishers which are “Wiley”.
Type of Implementation	MANUAL
Phase	After Integration Test
Dependencies	IT_05, ST_01
Requirement Tested	REQ_1, REQ_3, REQ_UB_1

Discussion

The UWON Library System project helps us review and understand the knowledge covered in the class such as software life cycle, configuration management, team organization, software testing, and design patterns. We understand how to apply those principles of software engineering to the design, development, maintenance, testing, and evaluation of computer software when we work on this project and fully integrate software engineering principles into the real situation. The UWON Library System project consists of 6 parts as Project plan, Configuration Plan, Requirements Specification, Design, Paper-based Prototype, and Testing Plan, and some of them require us to overcome certain difficulties to accomplish. Let me show you the challenges that we have encountered in this project and how we solved them finally.

In the Project Plan, the difficulty is how to determine our process model. As we know, the Evolution Prototype is used to obtain the requirements of some aspect of the system. The development of one or more prototypes may help to better understand the requirements of this project and then we can get some feedback from users when prototypes are tested intensively so that we can modify and perfect our requirements and coding and then move to the production phase. That's the reason why we select Evolution Prototype as our process model. In addition, our teams also have to do research for the work package to find out what it means and its format, what's more, we also need to search what is perfect hardware requirements to support the library system.

In the Identify configuration items, we thought we needed to select some tools which can be used for the project at the beginning, finally, we figured out that configuration items are those documents we generated to control the project. This is one of the tricky parts we have encountered.

ER-Diagram is a necessary and important part of our project and many other diagrams and Assumptions and functional requirements should be based on it. We were working on ER-Diagram so hard together for 4 hours and we used Google docs and Zoom to get everybody on the same page and to design it. Some teams live in different States and we are in different time zones but we still find some time to set up weekly meetings together no matter how difficult it is. The only goal for our teams is to finish our UWON Library System project on time and do our best to perfect it. We have accumulated lots of valuable experience in the UWON Library project, which will provide references for our future work.

Reference

1. JGraph Ltd(2005). Draw.io Diagram (V 7.1.9)[Diagram Modeling software] Published in 2005. Retrieved from http://app.diagrams.net/
2. Kenji Hiranabe(1999). Astah Professional JUDE(V 1.4)[UML Modeling software] Published in 1999. Retrieved from http://astah.net/
3. Best of BI (2009) SQL Power Architect (V 1.0.7)[Data modeling Software] Published in 2009. Retrieved from http://www.bestofbi.com/
4. Produle System(2008). MockFlow(Basic)[UI mockup Software]Published in 2008. Retrieved from https://mockflow.com/#login
5. Textbook: Vliet H. V., “Software Engineering: Principles and Practice”, 3th Edition, Wiley.