HÀ NỘI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software requirement Specification

Library Management System

Subject: ITSS Software Development

Group 6

Chử Quốc Hưng

Nguyễn Quốc Huy

*Hà Nội, December 7th 2017*

Table of contents

[1.1 Objective 2](#__RefHeading___Toc2447_3437929756)

[1.2 Scope 2](#__RefHeading___Toc2449_3437929756)

[1.3 Glossary 2](#__RefHeading___Toc2451_3437929756)

[1.4 References 3](#__RefHeading___Toc2453_3437929756)

[2 Overall Description 4](#__RefHeading___Toc2455_3437929756)

[2.1 Actors 4](#__RefHeading___Toc2457_3437929756)

[2.2 General use case diagram 4](#__RefHeading___Toc2459_3437929756)

[2.3 Use case decomposition Diagram 6](#__RefHeading___Toc2461_3437929756)

[2.3.1 Use case “Authenticate” decomposition 6](#__RefHeading___Toc2463_3437929756)

[2.3.2 Use case “Manage accounts” decomposition 7](#__RefHeading___Toc2465_3437929756)

[2.3.3 Use case “Manage Book Information” 7](#__RefHeading___Toc2467_3437929756)

[2.3.4 Use case “Manage Copy Information” 8](#__RefHeading___Toc2469_3437929756)

[2.4 Business processes 9](#__RefHeading___Toc2471_3437929756)

[2.4.1 Register a new account 9](#__RefHeading___Toc2473_3437929756)

[2.4.2 Add a new book 11](#__RefHeading___Toc2475_3437929756)

[2.4.3 Add a new copy 13](#__RefHeading___Toc2477_3437929756)

[2.4.4 Register to borrow book 14](#__RefHeading___Toc2479_3437929756)

[3 Use case specification 16](#__RefHeading___Toc2481_3437929756)

[3.1 Use case UC001 “Register Account” specification 16](#__RefHeading___Toc2483_3437929756)

[3.2 Use case UC002 “Add new book” specification 17](#__RefHeading___Toc2485_3437929756)

[3.3 Use case UC003 “Add new copy” specification 18](#__RefHeading___Toc2487_3437929756)

[3.4 Use case UC004 “Register to borrow book” specification 19](#__RefHeading___Toc2489_3437929756)

[4. Supplementary specification 20](#__RefHeading___Toc2491_3437929756)

[4.1 Functionality 20](#__RefHeading___Toc2493_3437929756)

[4.2 Usability 20](#__RefHeading___Toc2495_3437929756)

[4.3 Reliability 20](#__RefHeading___Toc2497_3437929756)

[4.4 Performance 20](#__RefHeading___Toc4316_1507369371)

## Objective

This document presents the detailed description for Library management system, user types and their rights at run time. This document also describes the objectives and features of the system, interfaces and constraints of the system in response to external action.

This document is for stakeholder and related software developers.

## Scope

In reality, any software needs to have the function of managing users or group of users, and dynamically granting permissions for usage of certain functions in the system.

The software’s goal includes creating users right managements, role and function available to users at runtime. Users can register accounts then use it to log in for access to system functions. Logging in is done via either using registered account in the system or Facebook account. Any users can update their personal info in the systems.

The administrators can disable an account, or request the user to change their password at a certain time or periodically. Administrator is able to grant certain role to a user. A user may possess many roles in the software. Each user’s role is configured to be used with some functions. Each function may be ultilised by many roles of users. If a new function is added, Administrator needs to provide information for this function in order to manage it using the software. After a successful log in, depends on the user’s role, the software will automatically create a menu according to the granted action of such user. Each time a function on the menu is selected, the corresponding interface will be presented.

## Glossary

The glossary contains the working definitions for the key concepts in the Library System.

**User**: a person that have an account in the system.

**Facebook:** a social network

**Gmail:** email service provided by Google

**HUST:** Hanoi University of Science and Technology

**Study period:** expected time to study in school. Example: 2014 - 2019

**Borrowing cards:** card use to borrow books

**Librarian:** a person that receives requests from user and takes care of the library everyday

**Copy:** a version of a book, a book can have many same copies

**Overdue:** the state of a borrowed book that is not returned after predetermined day

**Compensation:** money must be paid by borrower for any damage, loss or overdue

## References

# Overall Description

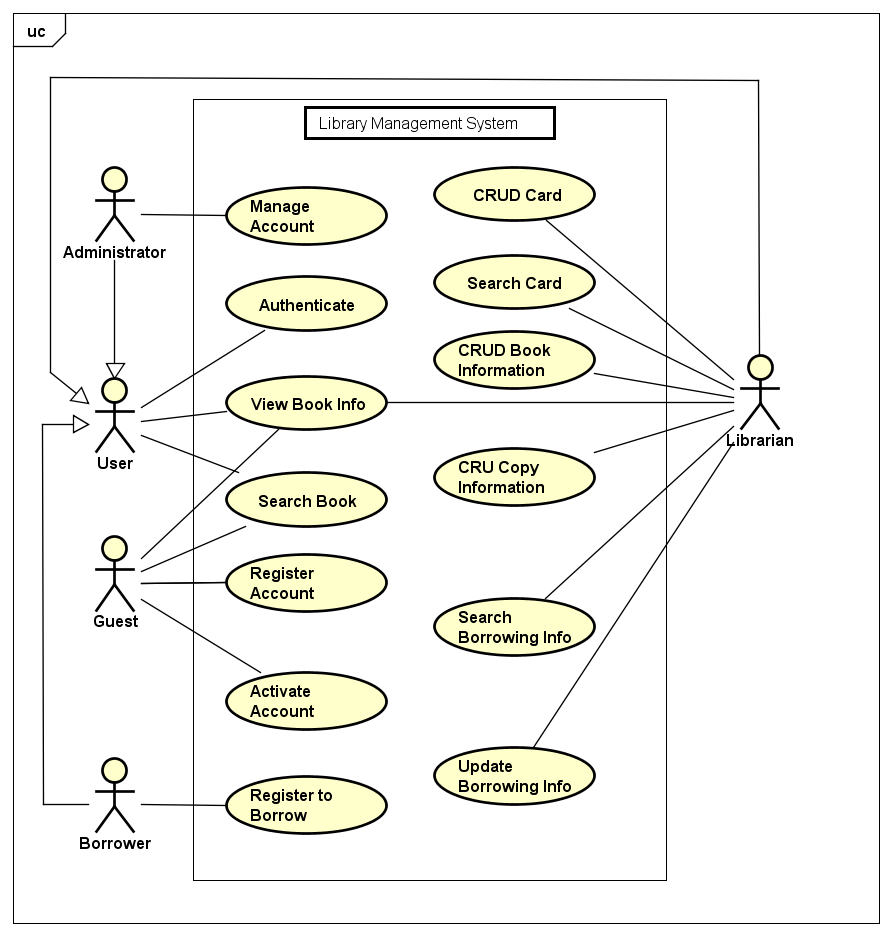
## Actors

There are 3 main actors in the system: Borrower, Librarian and Administrator which can be considered child actors of Users. User is the role of a person after successfully logging into the system. Borrower is the role of customer that can search and borrow books. Librarian is the role of librarian that manage books, copies and other activities of the library. Administrator is the most privileged role that have all the responsibilities of a librarian and can also manage users.

## General use case diagram

Before log in, a person can creates new account, log in, request password change if forgotten, and reconfigure the password after receiving instruction from the sent email. After receiving a successful log in, the system will present a menu according to each kind of users. When a guess request a password change, the system creates token and send the instruction for it using email.

After log in, user has the abilities to see and update personal information, and change password.



Any users can authenticate.

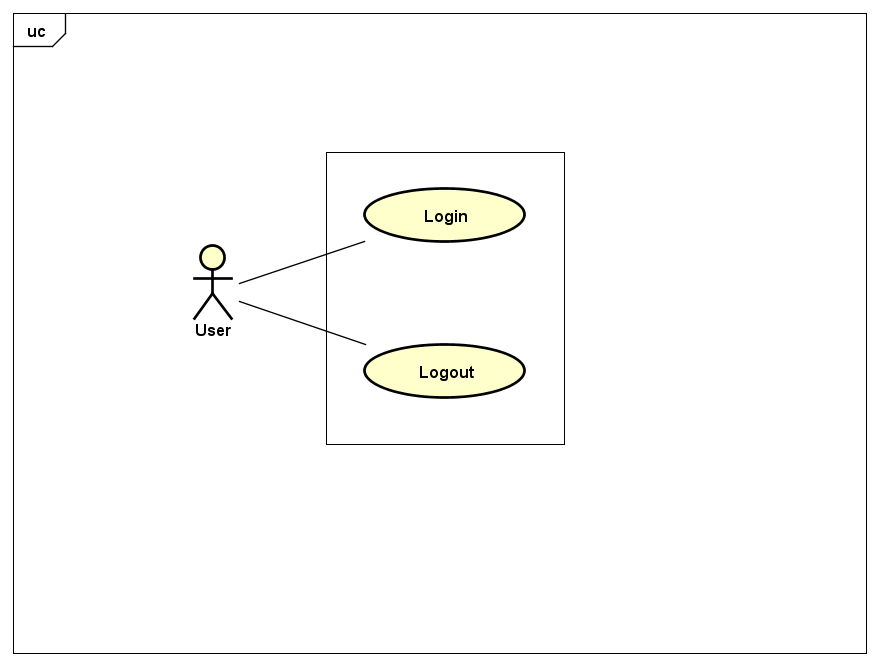
Administrator has the rights to manage accounts (other users).

Librarian has the abilities to mange book and copy information.

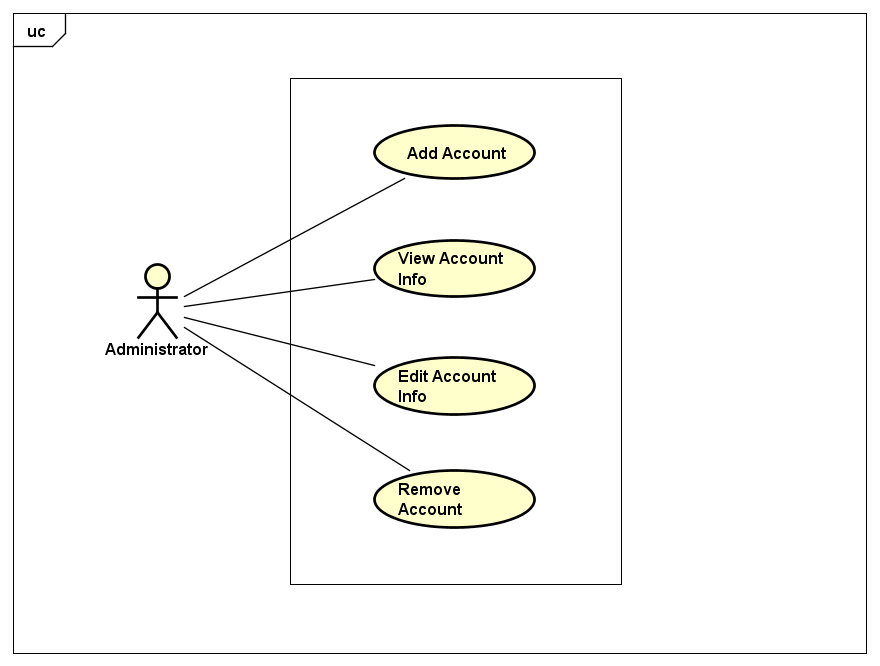
These are compound use cases whom details are given in the Use case decomposition diagram in the next section.

## Use case decomposition Diagram

### Use case “Authenticate” decomposition



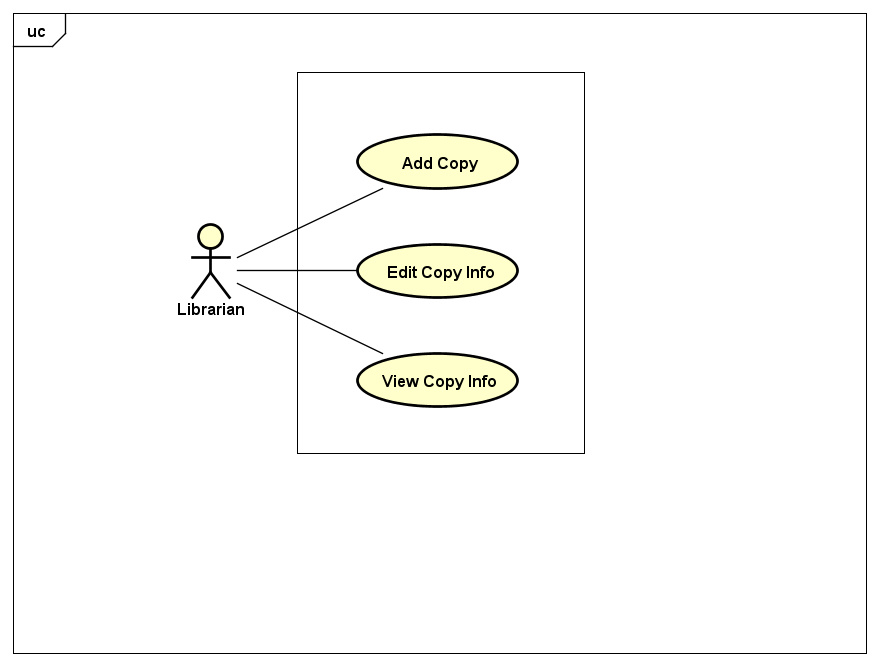
### Use case “Manage accounts” decomposition



### Use case “Manage Book Information”

### 

### Use case “Manage Copy Information”



## Business processes

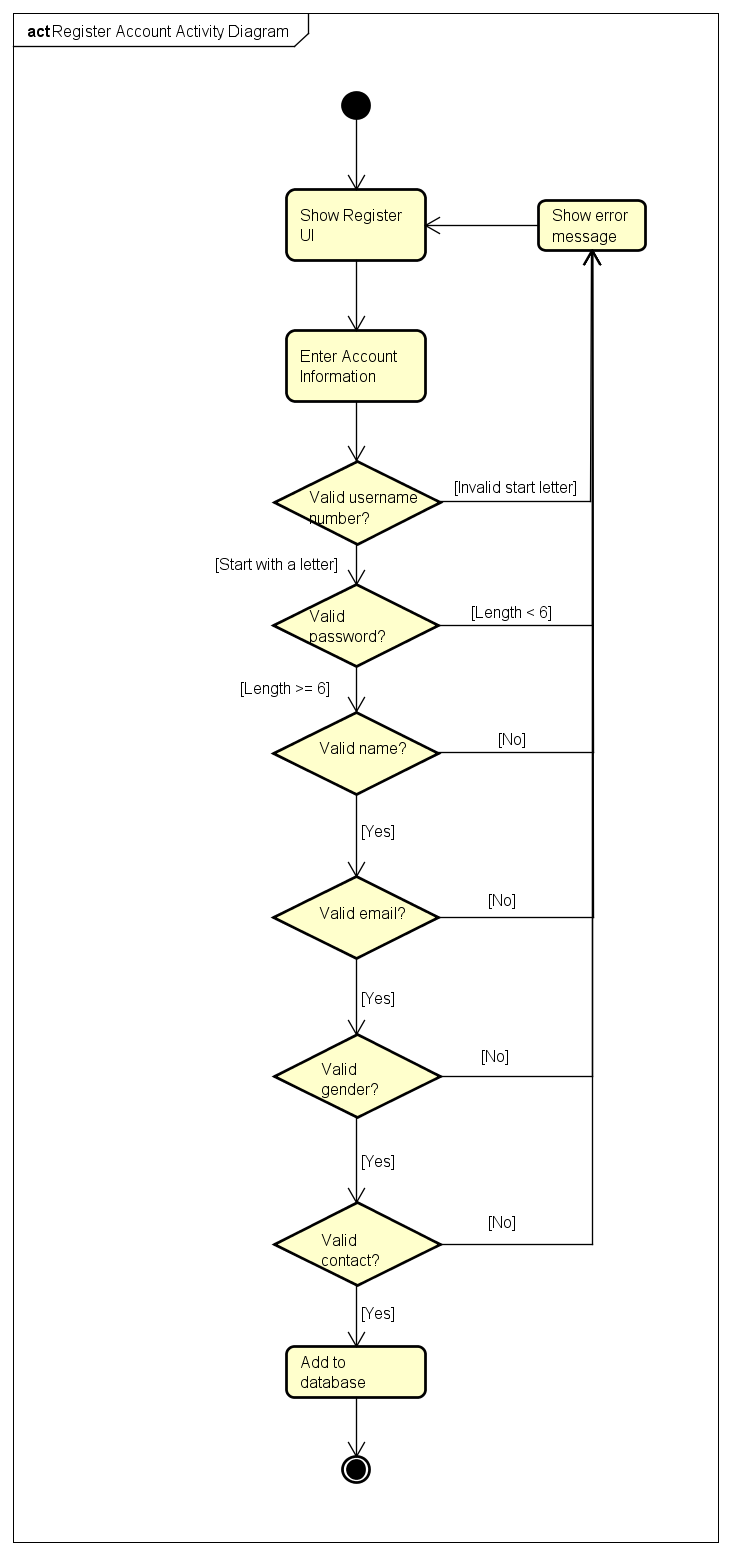
In this subsystem, there are 4 processes should be covered: register a new account, add a new book, add a new copy.

The details of these processes are modeled in the sub-section of each process.

### Register a new account

Guest can register his/her own account and log in to use the software afterward.

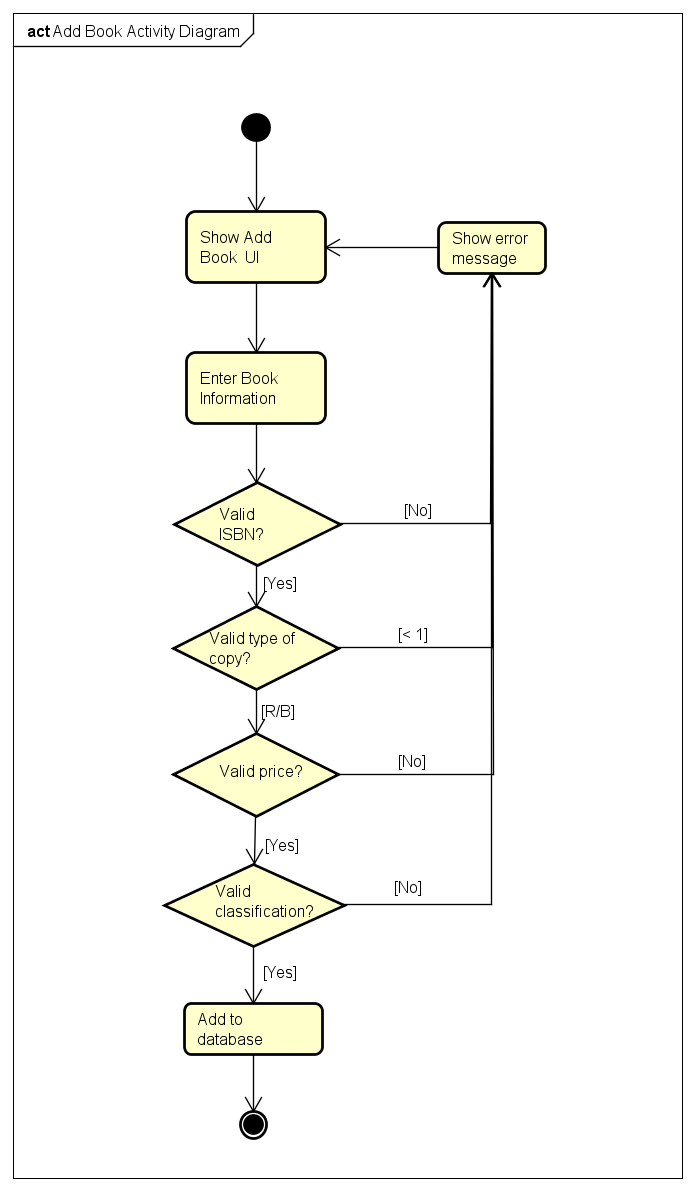
A GUI is displayed for user to enter information to register account. Each input option is then validated. If there are any input errors, corresponding message should be displayed. After all conditions are met new user account is added in to the system



### Add a new book

Librarian can add a new book into the library system.

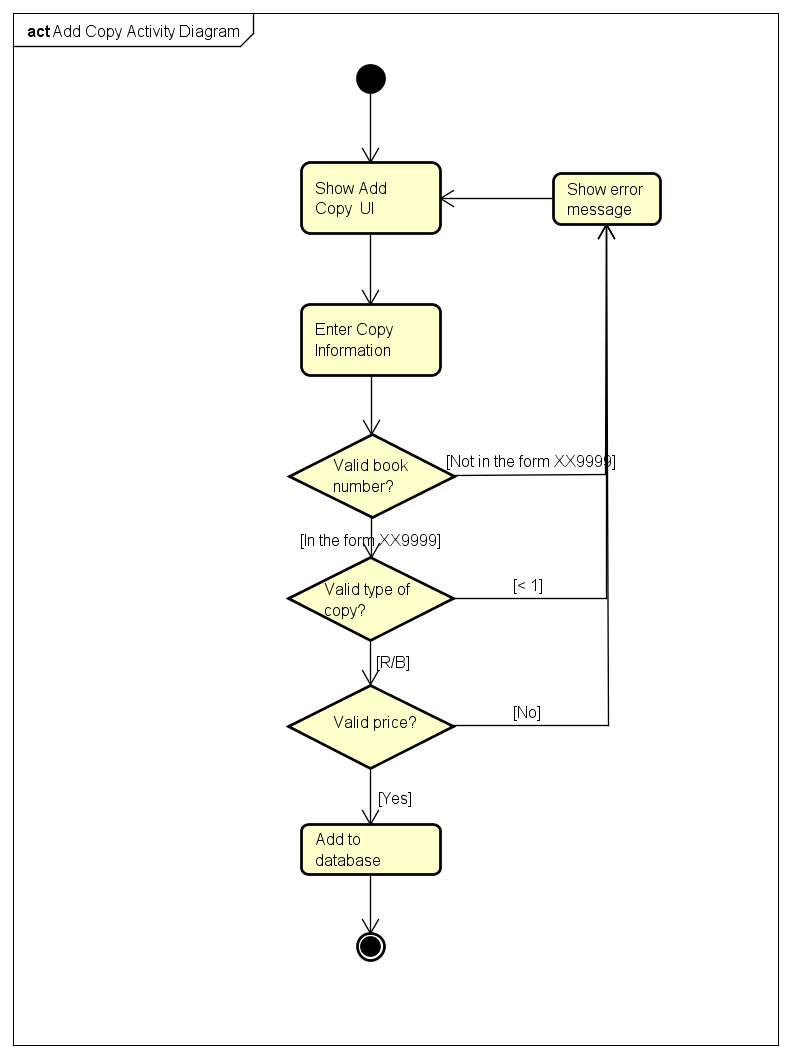
A GUI is displayed for librarian to enter book information. Each input option is then validated. If there are any input errors, corresponding message should be displayed. After all conditions are met new book is added in to the system



### Add a new copy

Librarian can add a new copy into the library system.

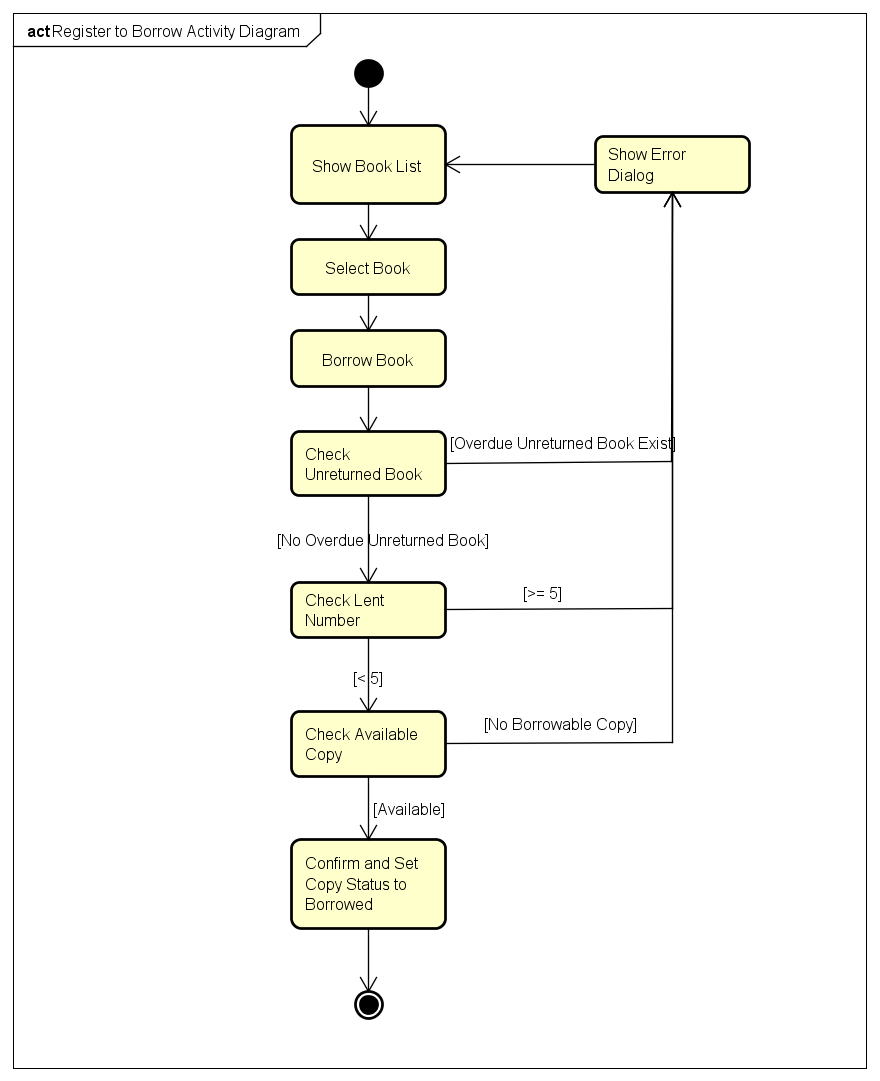
A GUI is displayed for librarian to enter copy information. Each input option is then validated. If there are any input errors, corresponding message should be displayed. After all conditions are met new copy is added in to the system



### Register to borrow book

A borrower can registered to borrow a book.

A GUI is displayed for user to choose a book to borrow. There are several conditions need to be checked. If there are any conditional errors, corresponding message should be displayed. After all conditions are met new borrow information is added in to the system



# Use case specification

Details of the use cases given in the 2nd section are specified below.

## Use case UC001 “Register Account” specification

|  |  |  |  |
| --- | --- | --- | --- |
| Use case code | UC001 | Use case name | Register Account |
| Actor | Guest (User) | | |
| Pre-condition | None | | |
| Main flow of events (Success) | |  |  |  | | --- | --- | --- | | No | Actor | Action | |  | Guest | Select Register function | |  | System | Display Register Account interface | |  | Guest | Input account information(detailed descriptions are given below \*) | |  | Guest | request Register new accout | |  | System | check whether guest had input the required field | |  | System | check whether the input in the correct form | |  | System | Check if the user already existed in the system | |  | System | Add new user to database and show log in GUI | | | |
| Alternative flow of events | |  |  |  | | --- | --- | --- | | No | Actor | Action | | 6a. | System | output error to guest: need to input into the missing required field if hadn’t done so | | 7a. | System | output error: corresponding error for each input | | 8b. | System | Output error: existed username | | | |
| Post-condition | None | | |

\* Input data of personal information include these data fields:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** |
|  | Username |  | Yes | Start with a letter | huy123 |
|  | Password |  | Yes | Length >= 6 | huy12345 |
|  | Full name |  | Yes |  | Nguyen Huy |
|  | Email |  | Yes | Email form | huy123@gmail.com |
|  | Gender |  | Yes | M or F | M |
|  | Contact |  | Yes |  | Ha Noi |
|  | Student Id |  | No |  | 20141234 |
|  | Study period |  | No |  | 2014 - 2019 |

## Use case UC002 “Add new book” specification

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case code** |  | **Use case name** | Add new book |
| **Actor** | Librarian | | |
| **Pre-condition** | Logged in as Librarian | | |
| **Main flow (Success)** | |  |  |  | | --- | --- | --- | | **#** | **Doer** | **Action** | |  | Librarian | Choose Add new book feature | |  | System | Show interface for librarian to add new book. Auto generate Book Number | |  | Librarian | Enter book information (detail below \*) | |  | Librarian | Send add new book request | |  | System | Check if librarian entered Title | |  | System | Check if librarian entered Publisher | |  | System | Check if librarian entered Authors | |  | System | Check if librarian entered ISBN | | 9. | System | Check if librarian entered Copy Information (detail below \*\*) | | 10. | System | Confirm and notify librarian | | | |
| **Alternative flow** | |  |  |  | | --- | --- | --- | | **#** | **Doer** | **Action** | | 5,6,7,8,9a. | System | Notify librarian and require them to enter missing info Title, Publisher, Authors, ISBN, Copy Information | | | |
| **Post condition** | No | | |

\*Book info:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field** | **Description** | **Required?** | **Condition** | **Example** |
|  | Title |  | Yes |  | The Godfather |
|  | Publisher |  | Yes |  | Pearson |
| 3. | Authors |  | Yes |  | Nate Silver |
| 4. | ISBN |  | Yes |  |  |
| 5. | Copy Information \*\* |  | Yes |  |  |

\*\*Copy info:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field** | **Description** | **Required?** | **Condition** | **Example** |
| 1 | Type of Copy |  | Yes | Reference or borrowable | reference |
| 2. | Price |  | Yes |  | 20000 |

## 3.3 Use case UC003 “Add new copy” specification

|  |  |  |  |
| --- | --- | --- | --- |
| Use case Code | UC003 | Use case Name | Add new copy |
| Actor | Librarian | | |
| Pre-condition | Logged in as Librarian | | |
| Main flow of events (Success) | | No | Actor | Action | | --- | --- | --- | |  | Librarian | choose Add new copy function | |  | System | display Add new copy GUI | |  | Librarian | input copy information (detailed description are given below \*) | |  | Librarian | request add copy | |  | System | check whether the librarian had input all the required fields | |  | System | Check if inputs are correct | |  | System | save the new copy | | | |
| Alternative flow of events | |  |  |  | | --- | --- | --- | | No | Actor | Action | | 6a. | System | output error to guest: need to input into the missing required field | | 7a. | System | output error: corresponding to each conditions of input | | | |
| Post-conditions | None | | |

\*Input data of personal information includes these data fields:

| **No** | **Data Fields** | **Description** | **Mandatory** | **Valid condition** | **Example** |
| --- | --- | --- | --- | --- | --- |
|  | Book Number |  | Yes | In the form XX9999 | SC0001 |
|  | Type |  | Yes | BORROWABLE  or REFERENCE |  |
|  | Price |  | Yes | number | 50000 |

## 3.4 Use case UC004 “Register to borrow book” specification

|  |  |  |  |
| --- | --- | --- | --- |
| Use case Code | UC004 | Use case Name | Search for user |
| Actor | Borrower (Users) | | |
| Pre-condition | Logged in as Borrower | | |
| Main flow of event (Success) | |  |  |  | | --- | --- | --- | | No | Actor | Action | |  | System | Show book list | |  | Borrower | Choose to borrow a book | |  | System | Check overdue unreturned book | |  | System | Check already borrowed number if = 5 | |  | System | Check available copy of the book | |  | System | Confirm and set copy status to LENT | | | |
| Alternative flow of events | |  |  |  | | --- | --- | --- | | No | Actor | Action | | 4a. | System | output: overdue unreturned book | | 5a. | System | output: already borrowed 5 books | | 6a. | System | Output: no copy available | | | |
| Post-condition | None | | |

# 4. Supplementary specification

## 4.1 Functionality

* Among the flow of events of use case, in all the steps that interacts with the database system, if there are errors in the connection or operation processes, there need to be a corresponding error notifications so that the actor knows that the error is related to the database system rather than the user.
* Use cases by Administrator, Borrower and Librarian must require the User to log in with the corresponding role

## 4.2 Usability

All the functionalities of the system need to be easy to operate. The system need to notify users when they did the wrong operation. The cause of the error and how to fix it should be display in the notification.

## 4.3 Reliability

The system must be available 100% for users because of the importance of data; the damages can happen by incorrect or incomplete data.

The system shall be operational 24 hours a day and 7 days a week.

If the system fails, the system will be recovered back up within an hour or less.

## 4.4 Performance

* **Response time**: The software shall respond to the user in not less than two seconds from the time of the request submittal. The software shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 3 seconds to appear on the screen.
* **Scalability**: the course registration system shall be capable of supporting at least 1,000 users when implemented into a suitable production environment.