Software Engineering Project(CS223)

Software **R**equirement **S**pecification(**SRS**)

Version: 1.0

12-02-2018

BOOK MANAGEMENT SYSTEM

THE LEARNING HUB | FACILITATING EDUCATION AT FINGERTIPS

Authors:

1. **Anurag Shah (B16CS034)**
2. **Chinmay Garg (B16CS041)**

Under guidance of:

Dr. Chiranjoy Chattopadhyay

Revision History

Date	Version	Description	People
19 Feb 2018	1.0	First draft	Project Group: 10
5 Mar 2018	2.0	Second draft	Project Group: 10

Table Of Contents:

1. Introduction 3
    1 Objective 3
    2 Scope 3
    3 Constraints 3
    4 Assumptions 4
    5 Dependencies 4
    6 Definitions, Acronyms and Abbreviations 4
    7 References 5
2. Overall Description 5
    1 Product Functions 5
        1 Functional Requirements 6
            1 Use Case Diagram 9
            2 Class Diagram 10
            3 Sequence Diagram 11
        2 Non-Functional Requirements 15
    2 User Characteristics 15
3. Specific Requirements 17
    1 Use case description 17
    2 Reliability 20
    3 Supportability 20
    4 Design Constraints 21
    5 Online User Documentation and Help System Requirements 21

1.0 Introduction

1.1 Objective

The objective of this documentation is to provide the specific information on what are the services provided by the project “Book Management System”, under what constraints it will work, and how it will interact with the users.

This SRS report is primarily for the the end-users but the developers can also refer this document for better insight of the software.

This project describes the workflow of the software using different UML diagrams.

1.2 Scope

This software will be used by all the members to access the rich resource collection of knowledge the Institute provides. This software is meant to increase the efficiency of managing the books by the staff.

The project is designed for the use of librarians, students and faculties of the institute.

The language used for developing the project is C++.

1.3 Constraints

1. Every member has a valid institute ID with which they should register in the system.
2. A member cannot issue more than one copy of same resource.

1.4 Assumptions

1. Every member registers through a valid college id.
2. All members have the software installed.
3. Enough store space is available to store the data of the members and resources.
4. The coding should be error free.
5. The system should be user-friendly so that it is easy to use for the users.
6. The Library System must be running 24 hours a day.

1.5 Dependencies

1. Any update regarding the book from the library is to be recorded to the system immediately and the data entered should be correct.
2. The end users (at least librarians) should have proper understanding about the working of the product.
3. The information of all the users and the resources must be stored.
4. The specific hardware and software due to which the product will be run.

1.6 Definitions, Acronyms and Abbreviations

Member	Anyone who is registered in the software
Librarian/ Staff	The library staff of the institute
User	Faculty or student registered in the software
Resource	Books/ Journals

1.7 References

- Books:
  - Roger S. Pressman, Software Engineering: A Practitioner's Approach 5th edition.
  - Ian Sommerville, Software Engineering 10th edition.
  - UML @ Classroom_ An Introduction to Object-Oriented Modeling-Springer International Publication
- Idea:
  - http://172.16.100.155:8080/newgenlibctxt/Home
- Online resources:
  - www.slideshare.net

2.0 Overall Description

2.1 Product Functions

Functions present in the software:

- Register members
- Issue resources
- Return resources
- Calculate Fines
- Add new resource
- Renew resources
- Search resources
- View issue history

2.1.1 Functional Requirements

R.1:Register/log in

- Description : First the user will have to register/sign up. There are two different type of users.
    - The librarian/staff : He/she have to provide details about the name of library ,phone number, address, email id.
    - User : The user have to provide details about his/her name, institute id, address, phone number, email id, password etc. All this will be entered into the system by the librarian during the registration.

R.1.1: Sign up

- Input: Detail about the user as mentioned above. This will be done by the librarian.
- Output: success report.
- Processing: Institute id will be checked for validity and the membership will be granted.

R.1.2 : Login

- Input: Enter the institute id and password.
- Output : Member will be able to use the features of software according to the access provided to them.

R.2 : Resource management.

R.2.1 : See your history.

- Description : List of books issued currently and in past will be displayed along with data of issue and return. Also, information regarding fine would be displayed.

R.2.2 : Search

- Input : Enter the title/ author's name/ resource ID/ topic of the resource to be searched.
- Output : List of all the resources related to the keyword.

R.2.3 : Issue

- Input : enter the ID of the resource which the user wants. This would be done by the librarian.
- Output : confirmation for resource issue or the error message.
- Processing : If the selected resource is available then book will be issued else the error will be displayed. Also if the user has reached his limit of currently issued resources, then the error will be displayed.

R.2.4 : Renew

- State : Resource is issued and is about to reach the date of return. This can be done by the user himself/ herself.
- Input : enter the ID of the resource to be renewed.
- Output : confirmation message.
- Processing : Check if the user has the authority to renew a resource, i.e., if he/ she must be faculty or a PhD student to renew a resource.

R.2.5 : Return

- Input : Return the resource to the library. Librarian will mark the resource returned in the system.
- Output : The issued list will be updated for both the resource and user.

R.2.6 Fine

- Input : check for the fines.

- Output : Description of the fines on different resources issued by the user.
- Processing : The fine will be calculated. If the resource crossed the date of return and the user did not renewed if then fine will be applied. In this case, return date subtracted by issue date will be used to calculate number of extra days the resource was with the user.

R.3 Manage resources

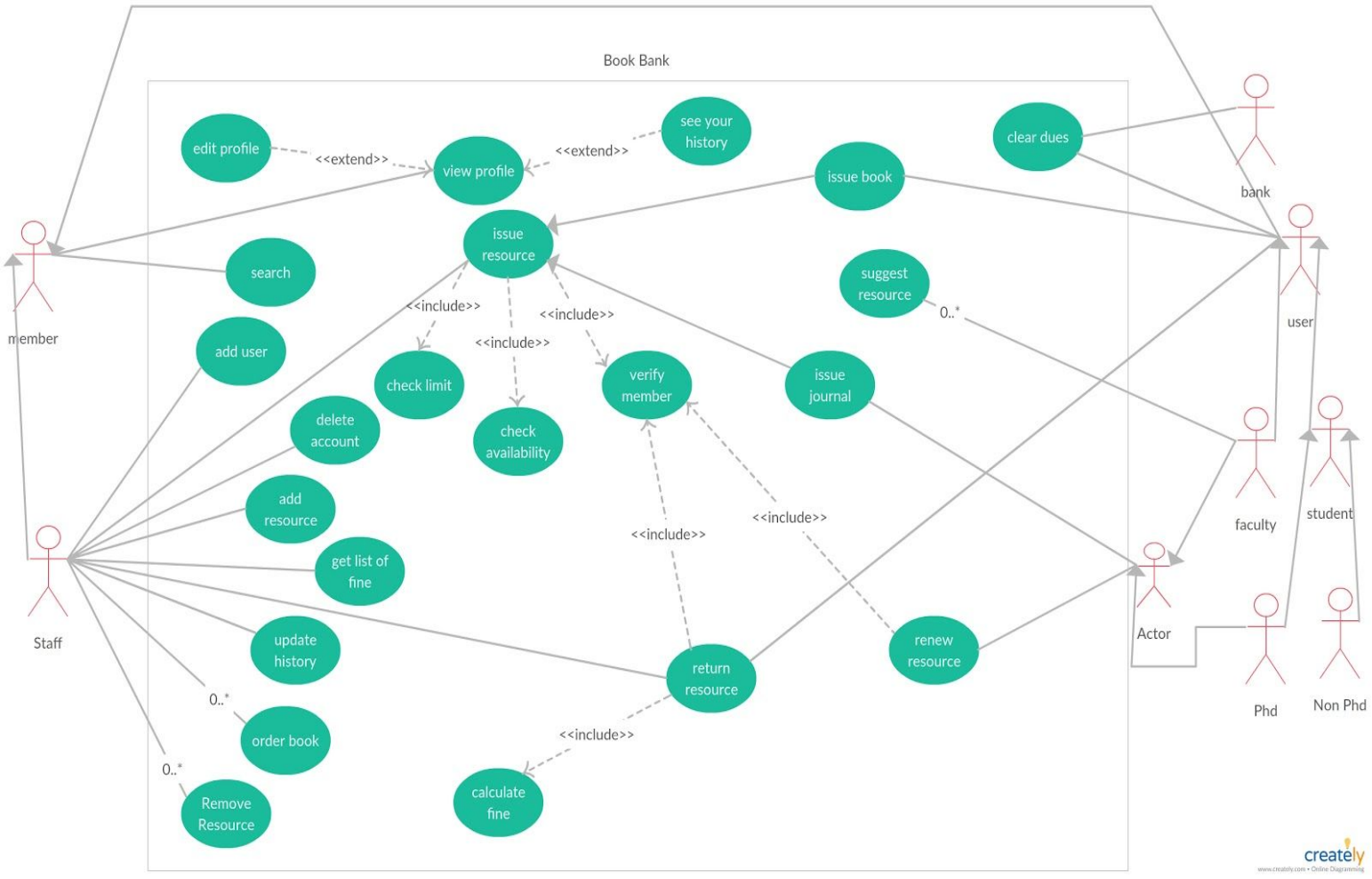
R.3.1 Add resource

- Input : Enter the details of the books such as title, author's name, date of purchase, resource id, quantity, etc.
- Output : confirmation of addition.

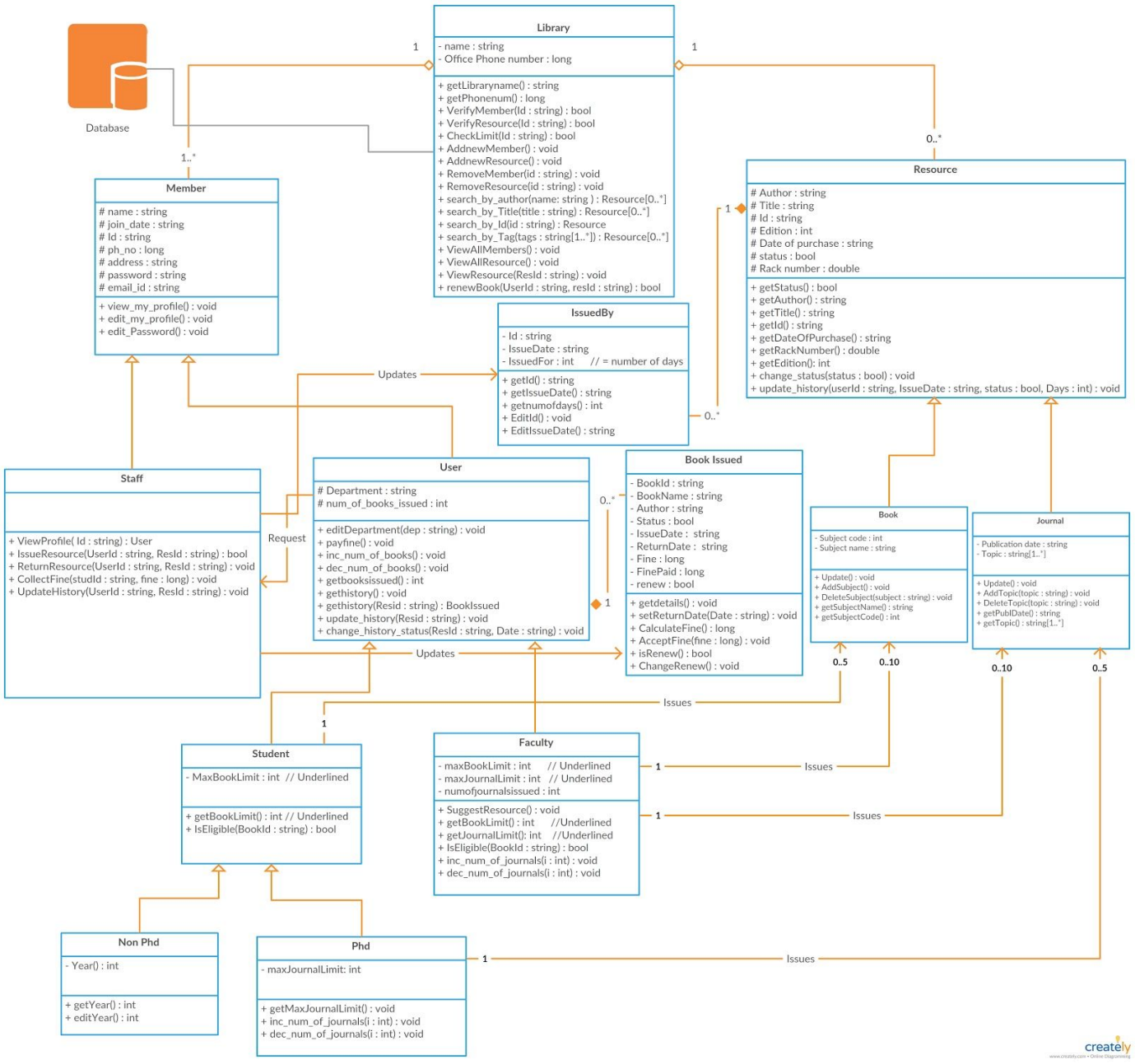
R.3.1 Remove books

- Input : Enter the id of the resource and the quantity of that resource to be removed.
- Output : Update the list of the resources available.

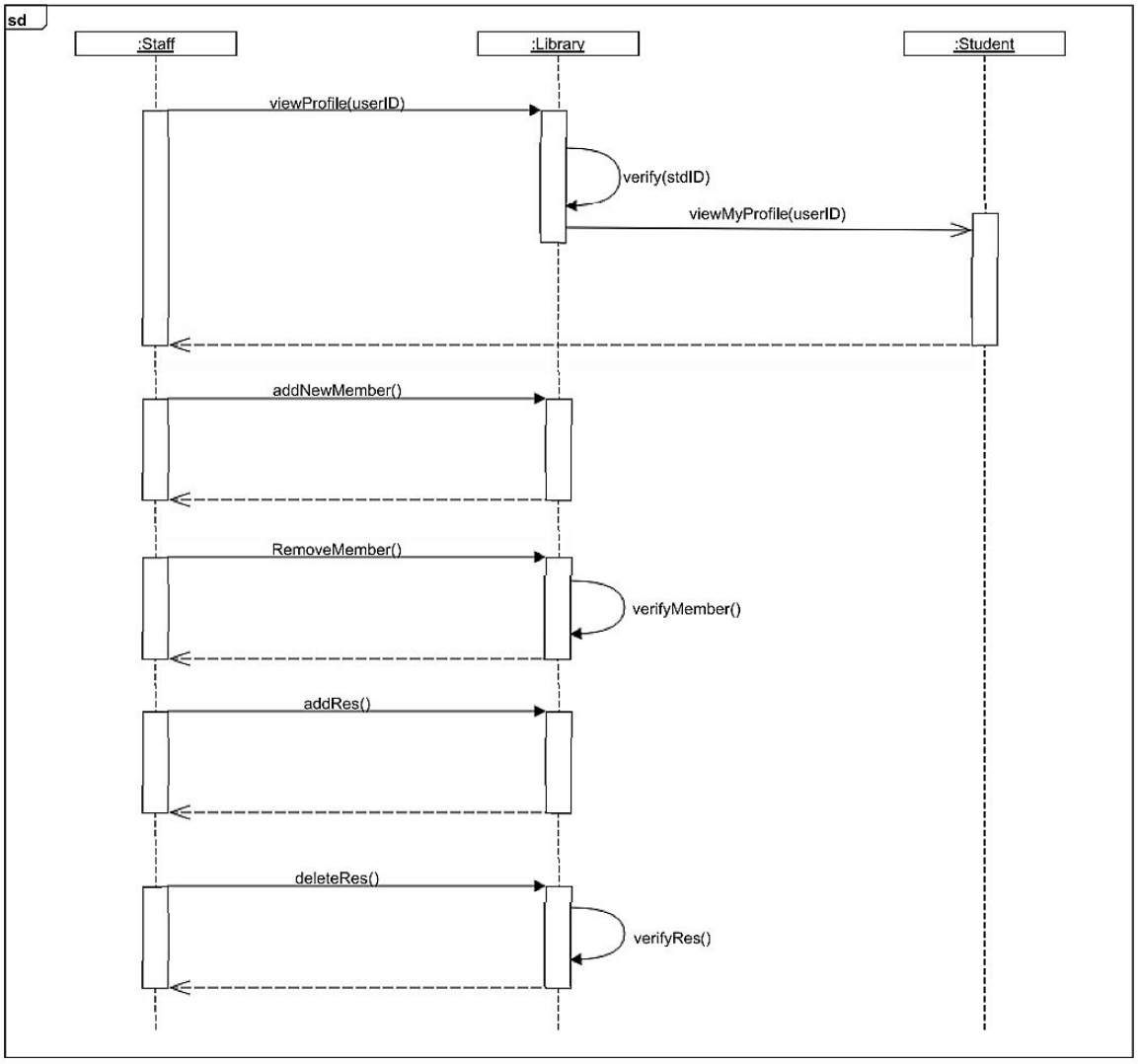
2.1.1.1 Use Case Diagram

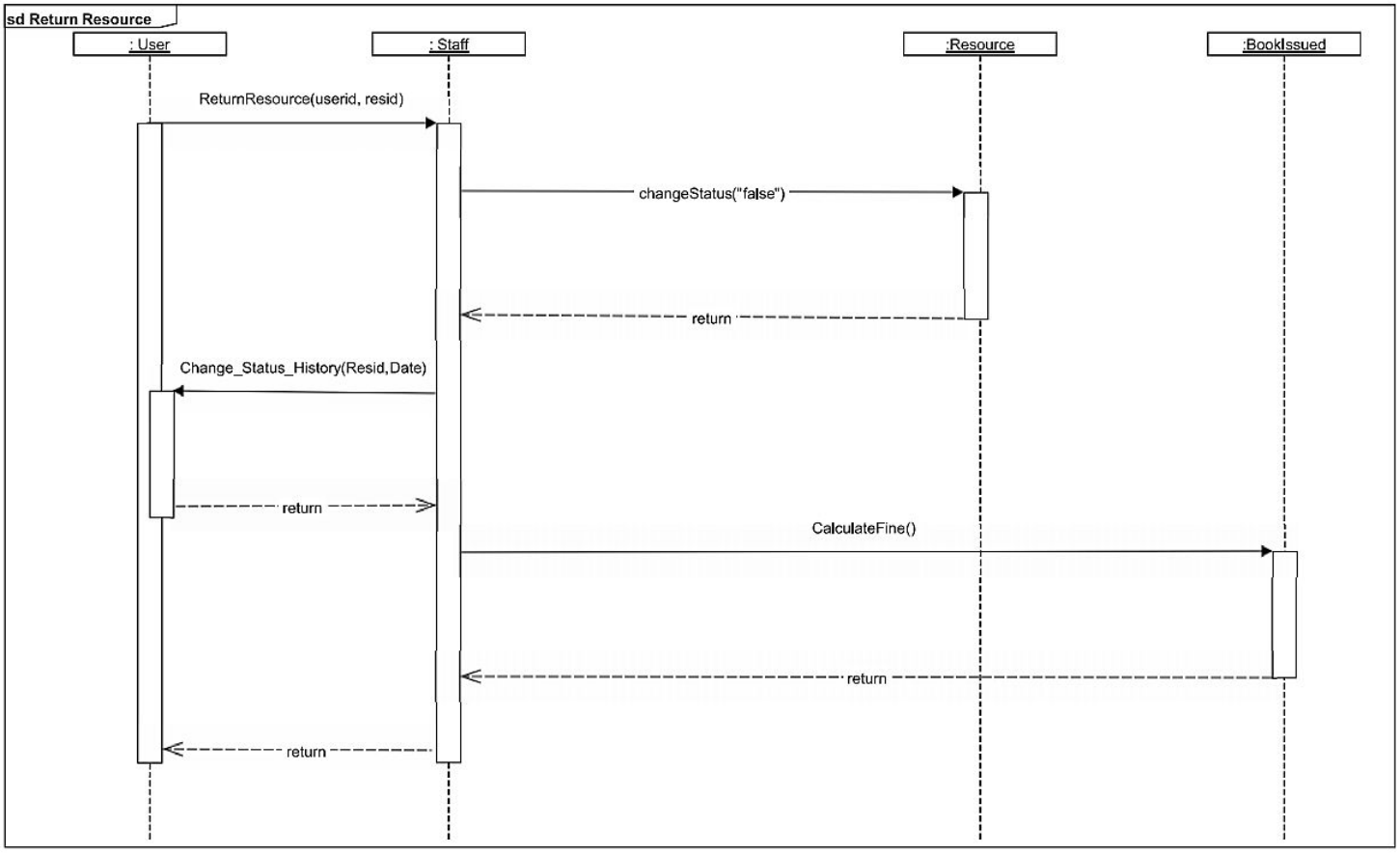


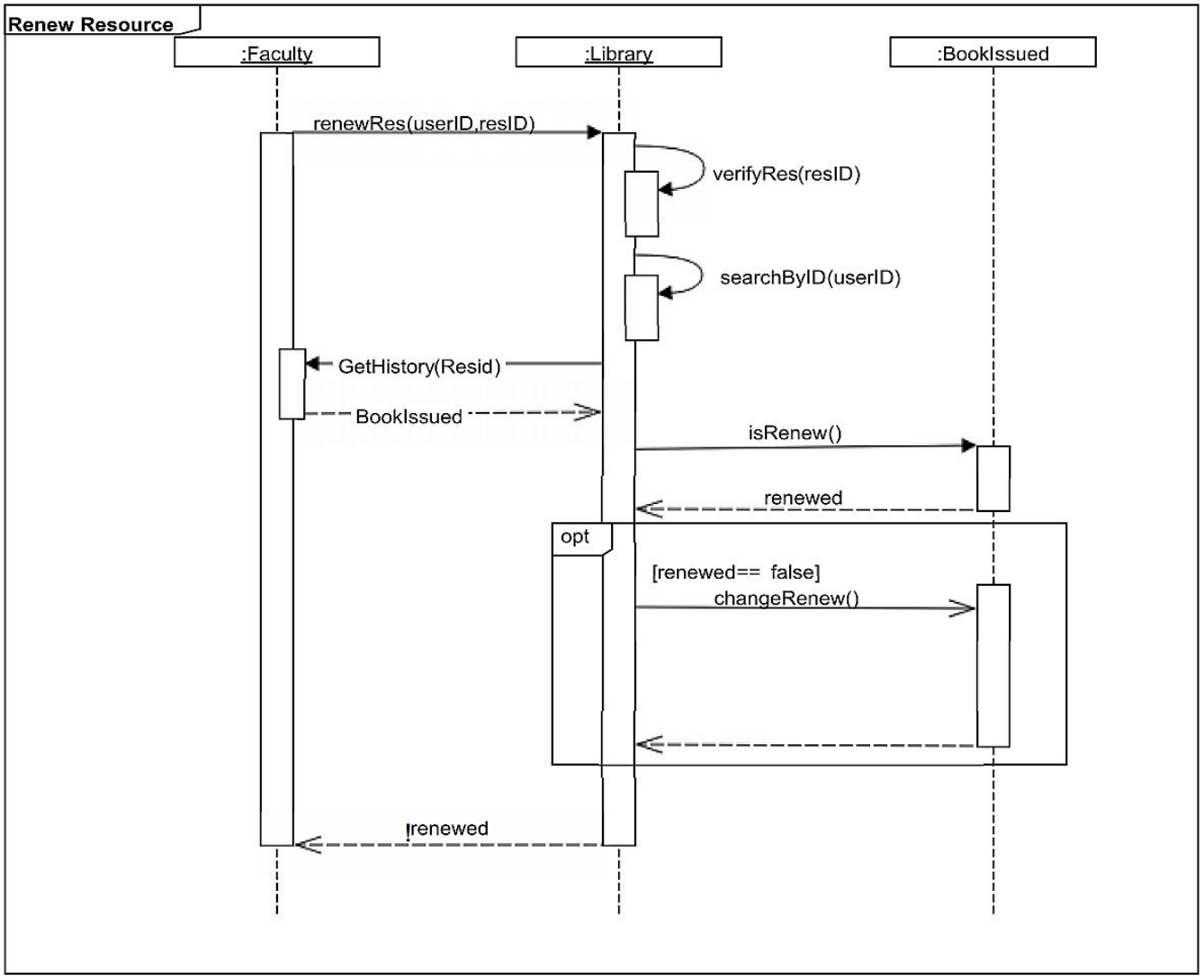
2.1.1.2 Class Diagram

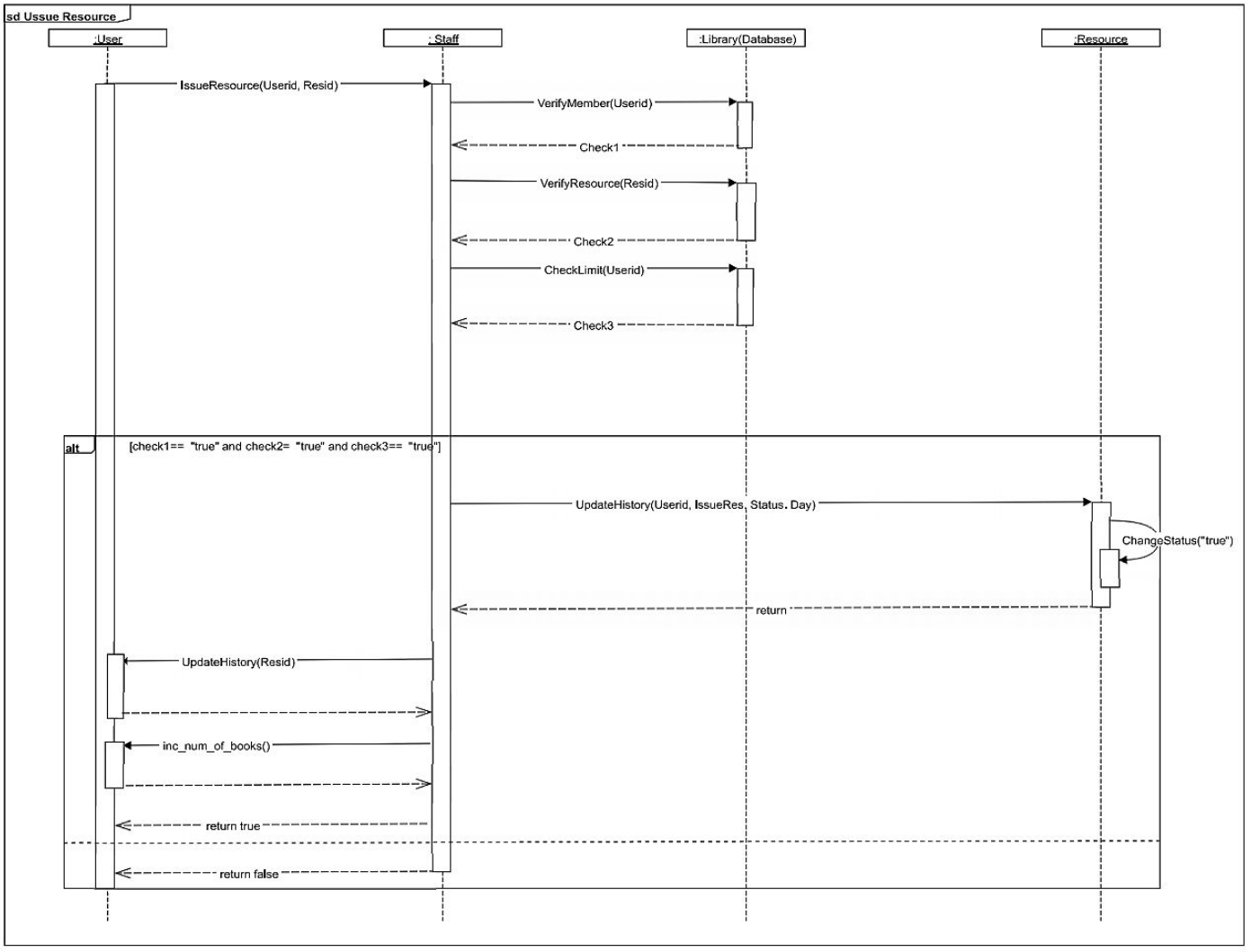


2.1.1.3 Sequence Diagram




sd Return Resource
: User
: Staff
:Resource
:BookIssued
ReturnResource(userid, resid)
changeStatus("false")
return
Change_Status_History(Resid,Date)
return
CalculateFine()
return
return


Renew Resource
:Faculty
:Library
:BookIssued
renewRes(userID,resID)
verifyRes(resID)
searchByID(userID)
GetHistory(Resid)
BookIssued
isRenew()
renewed
opt
[renewed== false]
changeRenew()
renewed


sd Ussue Resource
:User
: Staff
:Library(Database)
:Resource
IssueResource(Userid, Resid)
VerifyMember(Userid)
Check1
VerifyResource(Resid)
Check2
CheckLimit(Userid)
Check3
alt
[check1== "true" and check2= "true" and check3== "true"]
UpdateHistory(Userid, IssueRes, Status. Day)
ChangeStatus("true")
return
UpdateHistory(Resid)
inc_num_of_books()
return true
return false

2.1.2 Non-Functional Requirements

2.1.2.1 Hardware Configuration

runs on any pc.

2.1.2.2 Software Configuration

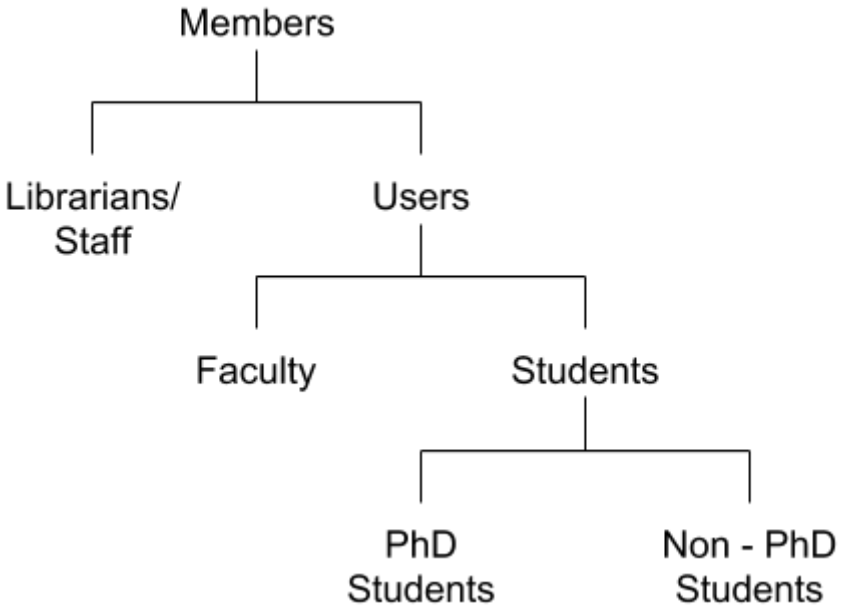
C++ compiler must be installed.

2.1.2.3 Requirement Attributes

- Only librarians can act as administrators. Users must not interfere.

2.2 User Characteristics

End users of the software are:



Functionalities available with:

Every Member

- view profile
- edit profile
- search book
- change password

Librarian

- add/ remove resources
- add/ remove members
- issue/ return resources for users
- collect fine
- update history
- view all members and resources

Every User

- checkout books
- pay fines

Faculty

- checkout journals
- suggest resources to buy
- renew resources

PhD Student

- checkout journals
- renew resources

3.0 Specific Requirements

3.1 Use Case Description

Use-Case Name:	add user
Primary Actor:	librarian, user
Stakeholders and interest:	user: wants to register into the system librarian: wants the user to get registered into the into the system
Pre-condition :	user has submitted his registration form
Post-condition :	record for a user has been added
Main success scenario :	1. librarian logs in his account. 2. selects the option add user. 3. enters the information of user 4. user gets registered

Use Case Name	issue resource
Primary Actor	librarian, user
Stakeholders and interest	user: wants to issue a resource librarian: wants the user to get the resource issued
Pre-condition	user knows his and the resource’s ID
Post-condition	resource is issued to the user

Main success scenario	1. librarian logs in his account. 2. selects the option issue resource. 3. enters the student's and the book's ID 4. resource gets issued

Use Case Name	return resource
Primary Actor	librarian, user
Stakeholders and interest	user: wants to return the resource librarian: wants the user to return the resource
Pre-condition	user tells the librarian his and the book's email ID
Post-condition	book has been returned
Main success scenario	1. librarian logs in his account. 2. selects the option return resource. 3. enters the student's and the resource ID 4. resource gets returned

Use Case Name	renew resource
Primary Actor	faculty, phd
Stakeholders and interest	faculty/ phd: wants to renew the resource
Pre-condition	faculty/ phd student knows the

	resource ID
Post-condition	book has been renewed
Main success scenario	1. faculty/ phd student logs in his account. 2. selects the option renew resource. 3. enters the resource ID 4. resource gets renewed

Use Case Name	order book
Primary Actor	librarian
Stakeholders and interest	librarian: wants to order a resource
Pre-condition	librarian knows the title, author's name and publisher's name of the resource
Post-condition	order for the resource has been placed with the vendor
Main success scenario	1. librarian logs in his account. 2. selects the option order book. 3. enters the information about the resource. 4. confirms the order.

3.2 Reliability

The system must be completely reliable and fool-proof due to the importance of data stored in it. Huge damages may be caused by incomplete or incorrect data. The system must run 24/7 for all days of the year as it is a C++ program.

Maintenance

Any update regarding any resource from the library is to be recorded, updated with correct values, and any fine on a member should be calculated correctly.

Maximum bug rate

There will be a maximum of 1 bug/KLOC.

Security Considerations

- Users can only read data and can't edit or modify anything except their personal information.
- The software has different types of users and every user has access constraints. Which have already been described.
- Proper member authentication has been provided.
- Only librarians can order resources.
- No one can issue any resource without a librarian.

3.3 Supportability

Naming Convention

All code will be written as specified by the Hungarian Naming Convention.

Coding Standards

Coding conventions we follow are:

- Comment conventions
- IEEE conventions
- Indent-style conventions
- Line-length conventions
- Naming conventions

3.4 Design Constraints

Software Language

Software is based on C++ programming language and thus, requires a C++ compiler.

3.5 Online User Documentation and Help System Requirements

The program is self instructive and hence does not requires any special help support.