

Mobile Library



By

Iqra faqraz

Aroona Ashkenaz

Waleed Bin Haider

**Submitted to the Faculty of Computing Software Engineering
National University of Sciences and Technology, Islamabad
In partial fulfillment for the requirement of B.E Degree in
Software engineering**

June 2015

ABSTRACT

We are working on an android application named “Mobile Library”. It aims to target the college/university students and faculty (MCS in particular) by providing them with an easy access to the MCS Library via a dedicated android application. This application will incorporate most of the options existing on the MCS Library website along with some added features to best facilitate the students regarding academic material. One of the major positives will be the minimized repeated physical labor to and from the library and overall promote the use of Libraries for attainment of authentic academic material. This application will have one module

Since the website of MCS, a part of PC Module relevant to this application is already in function. Our prime focus is on the providing the user with an interface to best utilize the library functions via their phones. This application will eliminate the time wastage in coping E-books from library and issuing / returning hard copies and library hour’s restriction.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

Certified that work contained in the thesis - Android Application for Mobile library carried out by Iqra faqraz, Aroona Ashkenaz and Waleed Bin Haider under supervision of Dr. Naima Altaf for partial fulfilment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

Dr. Naima Altaf

CSE-DEPARTMENT MCS

DATED: _____

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and unstinting cooperation,
a work of this magnitude would not have been possible

ACKNOWLEDGEMENTS

To begin with, there is no greater guide than **ALLAH (SWT)** Himself and we feel blessed that He gave us enough strength to complete this project well in time.

In addition to this, we all would deeply and genuinely like to thank **Ma'am Dr. Naima** for her persistent guidance and continuous support. Ma'am you are an exceptional supervisor and without you we could not have come this far.

Thank you very much.

Contents

Introduction

1.1 Introduction.....	1
1.2 Statement Of Problem.....	1
1.3 Statement of Goals & Scope of the Project.....	2

Literature Review

2.1 Introduction.....	3
2.2 Mobilna Biblioteka	3
2.3 Kritika Koha	3
2.4 OneClickdigital App	4

Software requirement Specifications

3.1 Introduction.....	5
3.2 Purpose.....	5
3.3 Main Scope	6
3.5 User Classes.....	6
3.6 User Interfaces	7
3.7 Functional requirements.....	11
3.8 Non-Functional Requirements	18

Software design specifications

4.1 Introduction.....	22
4.2 Architecture Diagram.....	22
4.3 Use-Case Diagrams.....	24
4.4 Activity diagrams.....	32

4.5Sequence diagrams.....	50
4.6Class diagram.....	59

Implementation

_5.1Introduction.....	60
_5.2User Interface.....	60
_5.3Backend Functionalities:.....	60
_5.4Database Configuration:	60
5.5 Integration of Database with Android App:.....	61

Testing and analysis

6.1Introduction	62
6.2Verification and validation	62
6.3Functional testing.....	62
6.4Functional testing plans	63
6.5Unit testing	72
6.6Integration testing	73
6.7System testing	74
6.8Summary.....	74

Future work

7.1Introduction.....	75
----------------------	----

Appendix

Glossary	98
----------------	----

List of Tables

Test cases

Table 1:Feature ID	64
Table 2:Registration	64
Table 3:Login.....	65
Table 4:Catalogue	67
Table 5:Reserve	67
Table 6:Renew	68
Table 7:Bookmark	68
Table 8:Tag	69
Table 9:Course group.....	69
Table 10:Account.....	70
Table 11:functionality on book.....	71
Table 12:Help	72

Table of figures

Figure 1:Mainscreen and registration.....	8
Figure 2:Login and Main menu	8
Figure 3: Search.....	9
Figure 4: book options and account	9
Figure 5 : help and about us	10
Figure 6:System overview	22
Figure 7:System Architecture	23
Figure 8:Login	24
Figure 9:Signup.....	25
Figure 10:Download	25
Figure 11:Order.....	26
Figure 12:Preview.....	26
Figure 13:Reserve	27
Figure 14:Bookmark	27
Figure 15:Sharing	28
Figure 16:Tagging.....	29
Figure 17:Course group	29
Figure 18:Mini library.....	30
Figure 19:Notifications	30
Figure 20:My account.....	31
Figure 21:Customize.....	31
Figure 22:Help	32
Figure 23 : Login and signup-Activity diagram.....	33
Figure 24 : Guest-Activity diagram	33
Figure 25 : Search-Activity diagram.....	34
Figure 26 : Download:Activity diagram	35
Figure 27 : Order results-Activity diagram.....	36
Figure 28: preview –Activity diagram.....	36
Figure 29 : Reserve-Activity diagram.....	37
Figure 30 : Tag-Activity diagram	37
Figure 32 : Share-Activity diagram	38
Figure 33 : Course Group-Activity diagram.....	38
Figure 34 : Reissue-Activity diagram	39
Figure 35 : Search-Activity diagram.....	40
Figure 36 : Download:Activity diagram	41
Figure 37 : Order results-Activity diagram.....	42
Figure 38: preview –Activity diagram.....	42

Figure 39 : Reserve-Activity diagram.....	43
Figure 40 : Tag-Activity diagram	43
Figure 45 : Mini library-Activity diagram	46
Figure 46 : Notification manager-Activity diagram	47
Figure 48: Customization –Activity diagram.....	49
Figure 49 : Help-Activity diagram.....	49
Figure 50 : Login and signup-Sequence diagram	50
Figure 51 : Guest-Sequence diagram	51
Figure 52 : Search-Sequence diagram	51
Figure 53 : Downloading-Sequence diagram.....	52
Figure 54 : Preview-Sequence diagram	53
Figure 55 : Reserve-Sequence diagram	53
Figure 56 : Reissue-Sequence diagrams	54
Figure 57 : Bookmarks-Sequence diagram.....	54
Figure 58 : Tag and share-sequence diagram.....	55
Figure 59 : course group-Sequence diagram.....	55
Figure 60 : Mini library-Sequence diagrams	56
Figure 61 : Notification manager-Sequence diagram	57
Figure 62 : Account-Sequence diagram.....	57
Figure 63 : Customization-Sequence diagram	58
Figure 64 : Help-sequence diagram	58
Figure 65 : Class diagram	59
Figure 66: implementation overview	61

Chapter 1

Introduction

1.1 Introduction

“Mobile Library” aims to target the MCS students in particular by providing them with an easy access to the MCS Library via a dedicated android application. This application will incorporate most of the options existing on the MCS Library website along with some added features to best facilitate the students regarding academic material. One of the major positives will be the minimized repeated physical labor to and from the library and overall promote the use of Libraries for attainment of authentic academic material. This application will have one module:

Cell-Phone Module

Since the website of MCS, a part of PC Module relevant to this application is already in function; our prime focus is on the providing the user with an interface to best utilize the library functions via their phones.

Server

The server side that will contain the database.

1.2 Statement Of Problem

As the time has progressed, the use of libraries has rather diminished. Mainly because of the fact that with growing technology people have somewhat become accustomed to ease in their methods. Initially, the library role was declining because of the physical labor required for the attainment of required material and furthermore the paperwork required for it. Then, after the introduction of mobile phones, it was noticed that people could not be bothered to find a PC in order to carry out their library use endeavor on the computer

platform. Thus there is a need for an android application that readily provides access to the Library to students at runtime using their mobile phones.

1.3 Statement of Goals & Scope of the Project

Mobile Library aims to target the MCS students in particular by providing them with an easy access to the MCS Library via a dedicated android application. One of the major positives will be the minimized repeated physical labor to and from the library and overall promote the use of Libraries for attainment of authentic academic material. Our prime focus is on the providing the user with an interface to best utilize the library functions via their phones.

Main Scope

User can do the registration

User can login/logout to/from the library account

User can manage his account settings

User can manage his/her course group

User can view records

User can search available books via various attributes

Reserve/Renew books

Bookmark desired books

Share book details

Tag

View book details

Complaints & queries submission

Extended Scope

Download/Preview

Mini Library folder

Customize app

Chapter 2

Literature Review

2.1 Introduction

It is the imperative for the modern world to keep the need for Libraries alive by catering for the factors that will ease the process. The primary step would be to make the reach to the Libraries easier and promote, rather, resurrect the use of Libraries in general. Mobile Library aims at achieving exactly that, but a few other of applications have also been produced keeping a similar approach in mind.

2.2 Mobilna Biblioteka

It is an application which embodies the same concept as the Mobile Library but it's additional features prove that this application is not exclusive to the books and the overall Library aspect, rather it is more general and includes books, movies, music disks and other items. The general idea though similar, is not exactly the same and rather far off in terms of what we hope to achieve with Mobile Library.

Main features includes manage books ,manage music, manage movies.

2.3 Kritika Koha

Kritika Koha is an android client for Koha. Koha is the first open-source Integrated Library System (ILS) which is most near to our project of Mobile Library. Main features are make account, search forms and Order Search

2.4 OneClickdigital App

An iOS based application that allows you to download and transfer audio books from OneClickdigital after making an account on OneClickdigital. Main features include are make account,bookmark audio ,Books Download ,Audio Books

Chapter 3

Requirement Specification

3.1 Introduction

“Mobile Library” aims to target the MCS students in particular by providing them with an easy access to the MCS Library via a dedicated android application. This application will incorporate most of the options existing on the MCS Library website along with some added features to best facilitate the students regarding academic material. One of the major positives will be the minimized repeated physical labor to and from the library and overall promote the use of Libraries for attainment of authentic academic material. This application will have one module:

Cell-Phone Module:

Since the website of MCS, a part of PC Module relevant to this application is already in function our prime focus is on the providing the user with an interface to best utilize the library functions via their phones.

Server

The server side that will contain the database

3.2 Purpose

The Software Requirement Specification document covers the overall description of the project including product perspective, features, functional and nonfunctional requirements, system and safety requirements, characteristics and constraints of the application.

3.3 Main Scope

User can do the registration

User can login/logout to/from the library account

User can manage his account settings

User can manage his/her course group

User can view records

User can search available books via various attributes

Reserve/Renew books

Bookmark desired books

Share book details

Tag

View book details

Complaints & queries submission

Extended Scope:

Download/Preview

Mini Library folder

Customize app

3.4 User Classes

System User

The system user will be the person in use of the application to accomplish the relevant tasks. The user can be students and/or faculty who know how to operate android smart phones.

Operating Environment

To run this software user will require the most common commodity these days, a smart phone with an android operating system.

Design and Implementation Constraints

This system requires a Wi-Fi connection, which is critical to its operation. Since it's an online Mobile Library, majority of the work, namely login & searching etc. will require a connection with the database server .

3.5 User Interfaces

Hardware Interfaces

The application will run on android based smart phone

Database Server

Software Interfaces

The application communicates directly with a MySQL database using the Java Database Connectivity API.

Wi-Fi connection

Android OS

Communication Interfaces

Web services will be written for android application

The web based application will access the web browser and communicate over HTTP protocol.

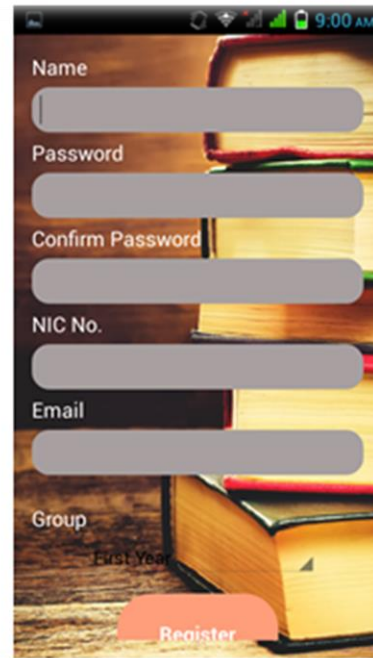


Figure 1: Mainscreen and registration

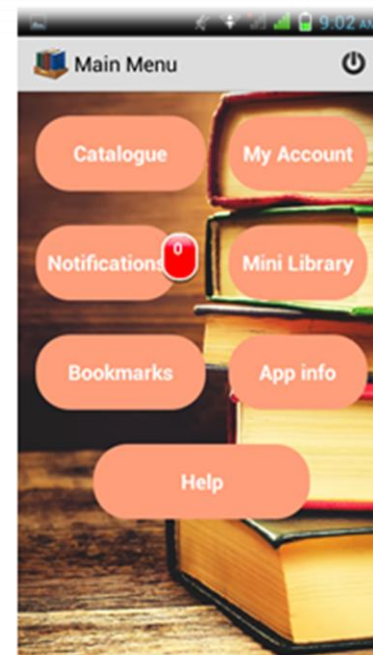


Figure 2: Login and Main menu



Figure 3: Search



Figure 4: book options and account



Figure 5 : help and about us

3.6 Functional requirements

Feature	Login/Logout
Description	This feature will allow the user to login his library account using his own id and password provided by the Library Management. when user login to his account.
Priority	9 (highest)
Normal Flow	<p>User enters his user name and passwords The app verifies and displays the main screen .</p> <p>User clicks on logout button The app logouts.</p>
Alternate Flow	If id or password mismatch occurs the system should prompt the user and ask to enter the id or password again
Functional Requirements	<p>The system should be able verify whether the id and password is correct.</p> <p>User's account will open only when the id and password has been verified.</p>

Table 1:Login

Feature	Search
Description	This feature allows user to look for a book in the library catalogue by entering a book's name or a related keyword in the search bar. The user can also search by Advance Search option
Priority	8

Normal Flow	<p>User clicks on search button after he enters book's name or keyword in search bar.</p> <p>Book name is searched in library's catalogue that matches the entered name or keyword.</p> <p>User clicks on Advance search button</p> <p>It displays some more options like search by ISBN, publisher name etc.</p>
Alternate Flow	<p>If the requested book is not found user is informed by a message on screen.</p>
Functional Requirements	<p>The user should be able to enter a book's name or keyword to search.</p> <p>The application should be able to search entered name or keyword in library's catalogue.</p> <p>The user should be able to search a book by its attributes (title, author, subject, series title, format, location, barcode, call number, publisher, ISBN).</p> <p>The application should be able to search entered attributes in library's catalogue.</p> <p>The application should be able to display searched results .</p>

Table 2:Search

Feature	Download and Preview Books (extended scope)
Description	This feature allows the user to preview and download the books.
Priority	8

<p>Normal Flow</p>	<p>The user enters Keyword and clicks Search Button.</p> <p>The app displays a list of Related Books</p> <p>The user selects any book</p> <p>The app displays the details of books</p> <p>The user clicks on preview icon.</p> <p>The app opens the book to preview</p> <p>The user clicks on download.</p> <p>The app display a Dialogue Box for confirmation</p> <p>The user selects Yes.</p> <p>The app starts downloading the Book</p>
<p>Alternate Flow</p>	<p>If the user selects NO, the downloading shouldn't start</p> <p>If the user selects Cancel button, the app should cancel the downloading and should be able to transfer the system to its previous state</p> <p>If the user press back button, the previous screen should be displayed</p>
<p>Functional Requirements</p>	<p>The database must be able to maintain its state and always be in updated form .</p> <p>The app should display an error message if downloading doesn't properly starts</p> <p>The app must generate notifications when necessary</p>

Table 3:Download

<p>Feature</p>	<p>Reservation</p>
<p>Description</p>	<p>This feature allows the user to reserve a Hard copy of book online. The user can renew the book .</p>
<p>Priority</p>	<p>8</p>

Normal Flow	<p>The user searches a book and clicks on reserve icon .</p> <p>The app displays a confirmation box</p> <p>The user clicks yes</p> <p>The app reserves a book and generate notification</p> <p>The user clicks on Renew button.</p> <p>The app displays a confirmation box</p> <p>The user clicks yes</p> <p>The app displays a notification after reissuing the book</p>
Alternate Flow	<p>If the user clicks NO , the app does not reserves the book and display previous screen.</p> <p>If the user clicks NO, the app does not renew the book .</p> <p>If the user press back or home button, the app will display respective screen</p>
Functional Requirements	<p>The database must be updated all the time</p> <p>The app must check for availability of books</p> <p>The app must generate notifications when necessary</p>

Table 4:Reservation

Feature	View book details
Description	This feature allows the user to view book specifications like (title, author, subject, barcode, call number, publisher, ISBN).
Priority	7
Normal Flow	<p>User clicks on view book details button.</p> <p>System gives a list of all the attributes of book that the</p>

	user requested
Alternate Flow	If details are not available, user should be informed by message on the screen.
Functional Requirements	<p>The user should have the view details button available to view book details.</p> <p>The system should be able to retrieve all book details data from the database.</p> <p>The system should display all the details to the user.</p>

Table 5Book details

Feature	Adding books to Mini Library Folder(Extended Scope)
Description	This feature will allow user to view downloaded books
Priority	7
Normal Flow	<p>User clicks the icon in Mini library.</p> <p>System displays all the downloaded books</p>
Alternate Flow	If the user clicks on delete button, the book should be deleted from the folder
Functional Requirements	The system should provide the user with downloading option with searched items.
Feature	Other option
Description	These are some feature that allows the user to share book information, Adding books to their favorites and tagging books to their class fellows and other users.
Priority	7

Normal Flow	<p>The user searches a book and clicks on tag icon .</p> <p>The app displays a box to enter name of person</p> <p>The user enters the name</p> <p>The app tag the person and generates a message</p> <p>After selecting the book,the user clicks on share icon.</p> <p>The app displays a dialogue box to confirm</p> <p>The user clicks yes</p> <p>The app shares the book info to his/her course group</p> <p>After selecting the book, the user clicks on bookmark</p> <p>The app displays a dialogue box to confirm</p> <p>The user clicks yes</p> <p>The app adds the book to favorites list</p>
Alternate Flow	<p>If the user clicks NO , the sharing of book will be canceled.</p> <p>If the user clicks NO, the book info will not be added to Favourites list.</p> <p>If the user press back or home button, the app will display respective screen</p>
Functional Requirements	<p>The app should be able to retrieve data from database</p> <p>The app should allow the user to search in database</p>

Table 6:Mini Folder

Feature	Help
Description	This feature allows the user to compose an email or make a phone call to library staff
Priority	5
Normal Flow	<p>The user clicks on Help</p> <p>The system displays him a list containing email id and phone number of library administration</p>

	<p>The user clicks on Make Call</p> <p>The system makes a call to the relevant staff</p>
Alternate Flow	If the user clicks cancel , it should cancel the call.
Functional Requirements	The app should be able to generate a call

Table 7:Help

Feature	Account settings
Description	This feature allows the user manage his/her account settings
Priority	7
Normal Flow	<p>User clicks on My Account icon</p> <p>The system display him with some options like My Personal info and Change password.</p> <p>The user selects my personal info</p> <p>The system display him his details</p> <p>The user Clicks on Edit option</p> <p>The system allows him to edit any field</p> <p>The user clicks on save button</p> <p>The system saves all the data back to database</p> <p>The user clicks on Change password button</p> <p>The system allows him to change password</p> <p>The user enters new password twice and presses OK button</p> <p>The system matches and notifies the user</p>
Alternate Flow	<p>After viewing details ,if user presses Home button then system will display Home screen</p> <p>If user clicks on Don't save then changes will not be save in Database</p> <p>If password doesn't matches ,the system will ask to re-enter</p>

Functional Requirements	<p>The system should have a complete detail of User</p> <p>The system should verify whether entered passwords are same or not</p> <p>The system should be able to retrieve and save Data to and from Database</p>
--------------------------------	---

Table 8:Account settings

3.7 Non-Functional Requirements

Performance Requirements

System Response

The application should respond to the user’s request for an action within approximately 5-10 seconds or more provided the internet connection is established. Scrolling should take at most 1 sec. If in any case the app has been unresponsive for a considerable period of time it should offer the user an option to quit the app.

Safety Requirements

When preparing to install ,the application should display a dialog to the user that indicates the permissions requested and asks whether to continue the installation so that incorrect or malicious use of resources by the application or threat from other applications is avoided that can adversely impact the user experience, the network, or data on the device.

Security Requirements

Authorization id and password is provided by the library management to the user. The user has to enter id and password to gain access to his account details. Following are the parameters of the security features:

If a user enters the id or password wrong for three times then system does not allow user to input code for next 10 seconds.

Software Quality Attributes

Some of the quality attributes identified includes:

Adaptability

The application should be adaptable to devices running Android version 2.3.3 (gingerbread) to Android version 4.2 (Jelly bean). Updating the Android OS version will not have any impact on the working of the application.

Availability

The application will be available all the time, provided mobile is in working state and the internet connection is available.

Backup

Backup should be enabled for the application. If a user wipes the data on his device or upgrades to a new Android-powered device, application should restore the user's previous data when the application is reinstalled.

Capacity

The application should efficiently utilize the android storage options to store application data in databases, files, or preferences, in internal or removable storage. In case of limited capacity it should prompt the user.

Compatibility

The application is compatible with android. Minimum version of android should be gingerbread.

Efficiency

The application should ensure reduced memory allocation so that the user does not encounter slow interface. Memory leaks should be avoided so that the application does not crash by going out of memory.

Flexibility

The architecture of the application will be flexible enough so if addition or change can be done if needed.

Interoperability

The user should be able to use application from any Web services enabled platform. The application should be interoperable with SQL database through web services.

Maintainability

The application should be made such that whenever there is a change in requirement or bug found, it is easy to find what needs to be changed, easy to make changes and easy to check that the changes have not introduced any bugs. Maintainable software should be easy to be understood by someone new - or someone that hasn't seen it in a while - with a minimum of effort.

Portability

In API, portability can be defined as “compatibility of application with platform (Android’s version) upgraded or downgraded versions. In Android’s platform when an up gradation is done, application requires some changes for compatibility with new version. Extensive testing will be required for verification & validation to address compatibility issues. Supported base android version will be Android 2.3.3 with support up to Android 4.2.

Reliability

Application should have extensive error handling mechanism. Any task if not executed properly or changing state within approx.10-15 seconds should generate an error. A generic error message should be used instead of detailed error messages.

Robustness

Application should regularly save to disk any information necessary to restart. If a failure occurs, the application should pick up from where it was before, rather than completely starting over. If the connection to the internet is broken prior to a request being either confirmed or cancelled the application should prompt the user that the connection has been interrupted.

Stress

The application should be capable of handling extremely high or stressful load. Application should have a “safe mode” to which it can return if something truly unexpected occurs. If all else fails, it should log the user out and close the browser window.

Usability

Users should be able to simply, quickly, and intuitively use the application. The user interface of the application will be designed user friendly and self-explanatory. Required minimum training of 5-10 minutes will be enough for the understanding of the system. Color scheme, layout and order of elements should be constant.

Chapter 4

Software Design Specifications

4.1 Introduction

Scope of this document focus on various aspects of Mobile Library project which includes architectural, procedural design and design assumptions, dependencies and constraints. A wide range of design considerations would be explained, from design restrictions to design goals and guidelines construct the scope of this document. Diagrams are used in a formal and intelligible fashion in order not to make any disambiguation in readers' minds. Description of the external libraries and tools upon which the Mobile Library project is going to be established, will be covered throughout the following sections of this design report.

4.2 Architecture Diagram

Overview

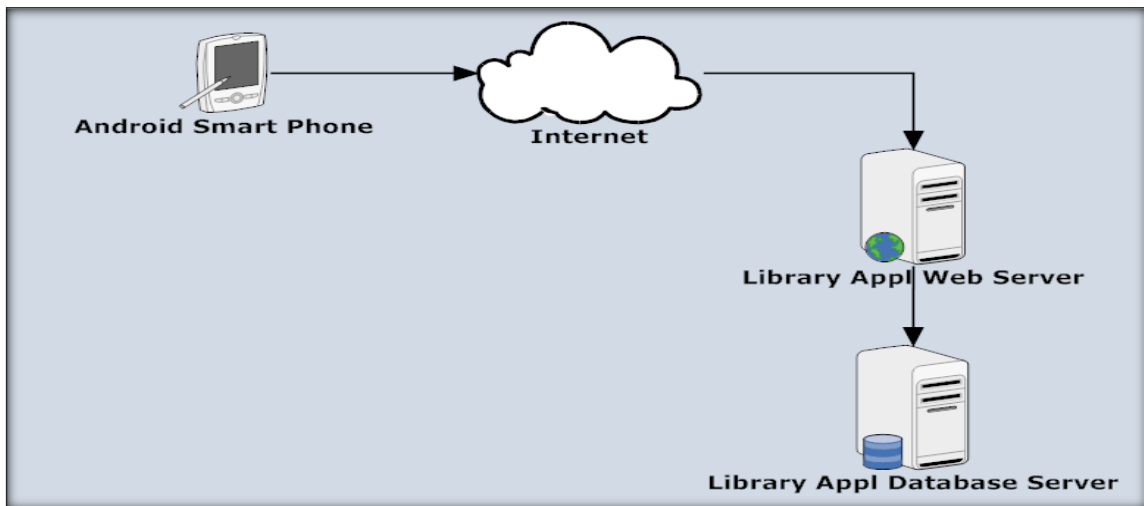


Figure 6: System overview

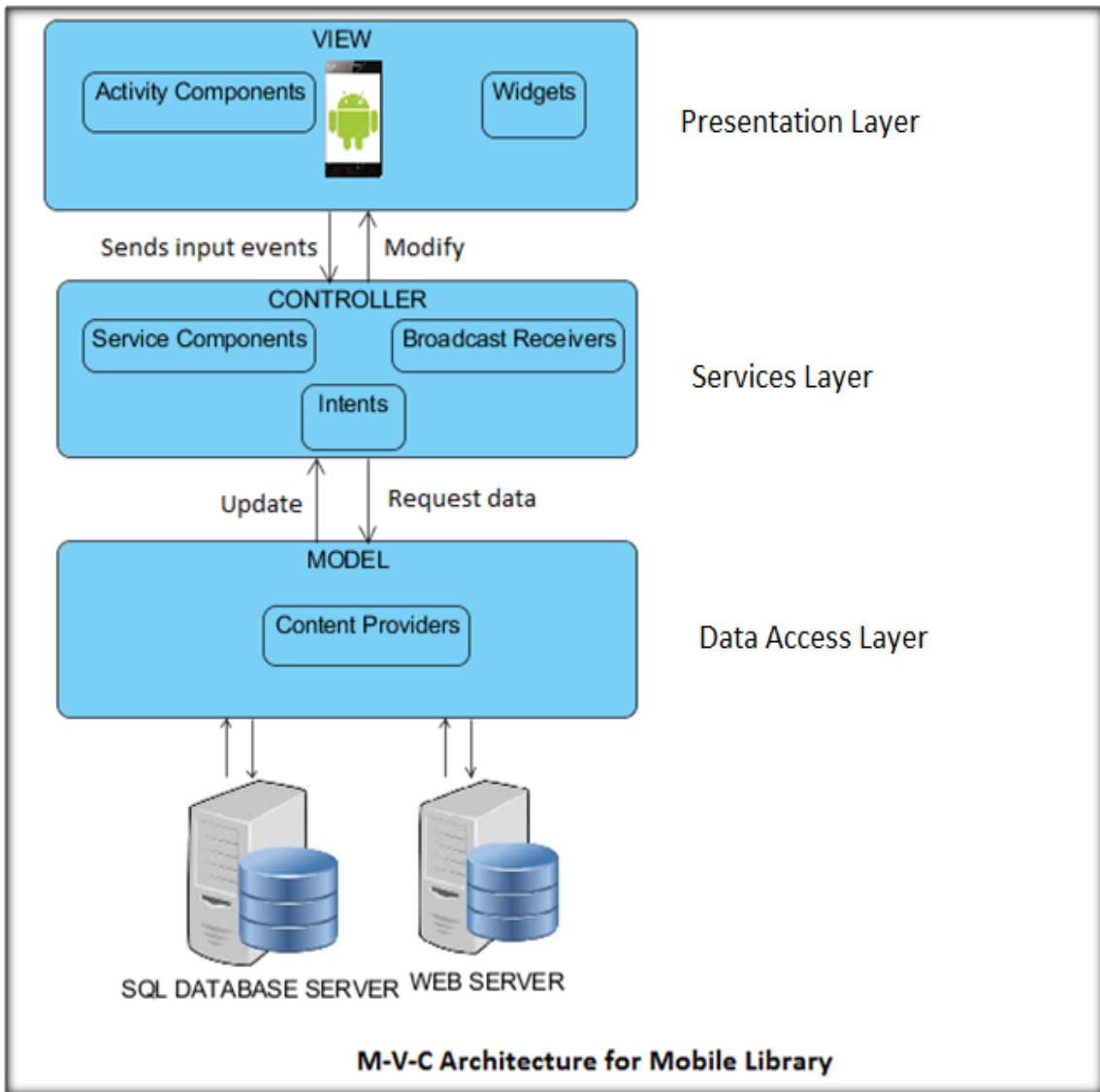


Figure 7: System Architecture

The components of android application (Activity, Service, Content Provider, Broadcast Receiver, Widgets, and Intents) that will be used in mobile library have been divided in M-V-C architecture model. The layered architecture supports the M-V-C model.

Model

Model is where the application’s data objects are stored. The model doesn’t know anything about views and controllers. It responds to the Controller’s requests for data and web services.

View

View is what's presented to the users and how users interact with the app. It fetches queries from the user and forwards it to Controller.

Controller

The controller is the decision maker and the glue between the model and view. It processes the queries.

SQL Data Base Server: Library database for all student and books info.

Web Server: Server for handling requests for web services.

4.3 Use-Case Diagrams

Guest Login:

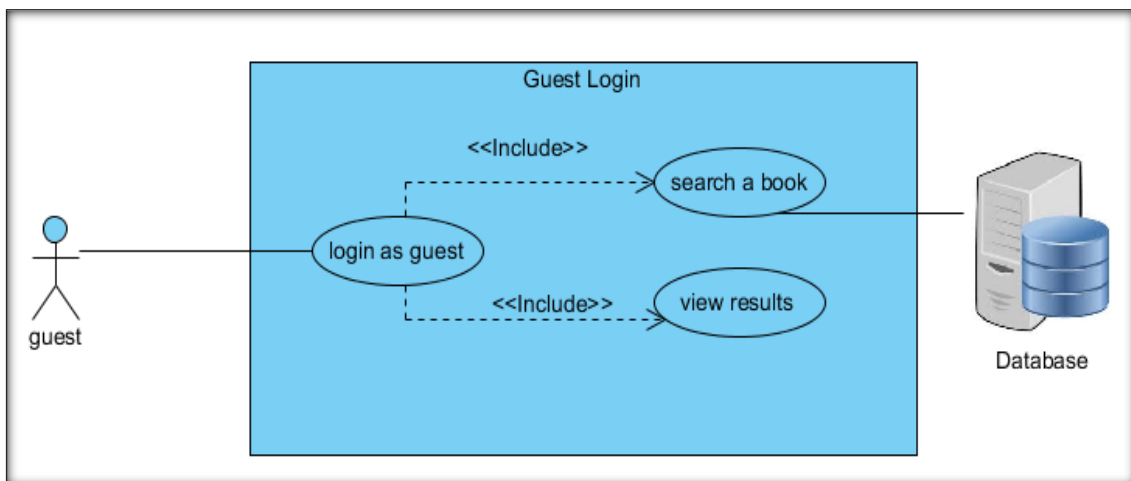


Figure 8:Login

Access Account:

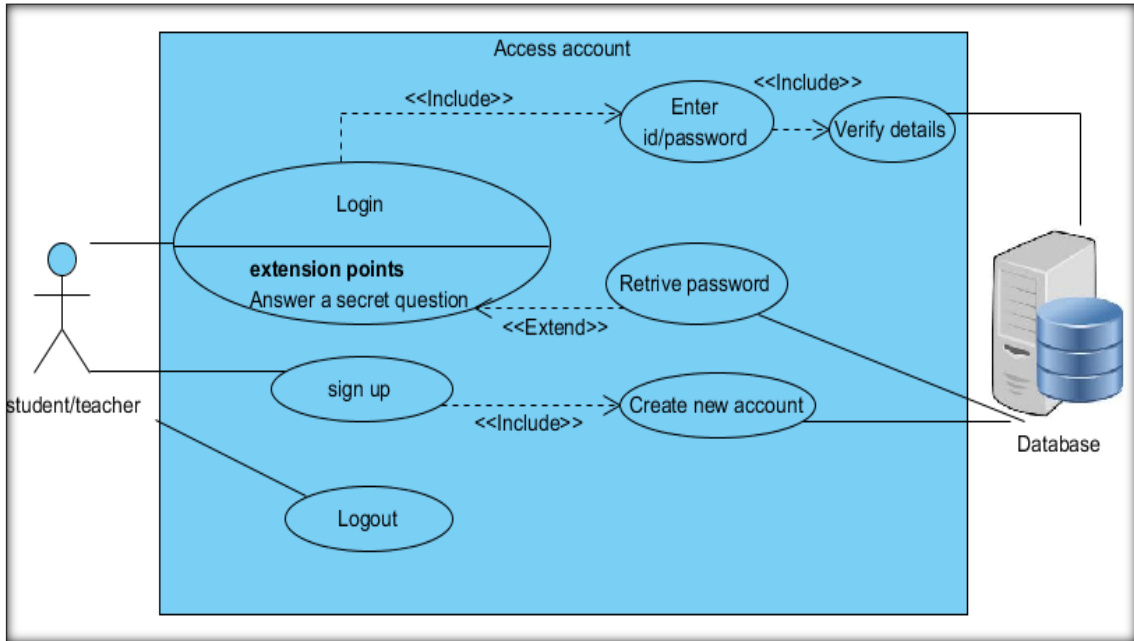


Figure 9:Signup

Downloading:

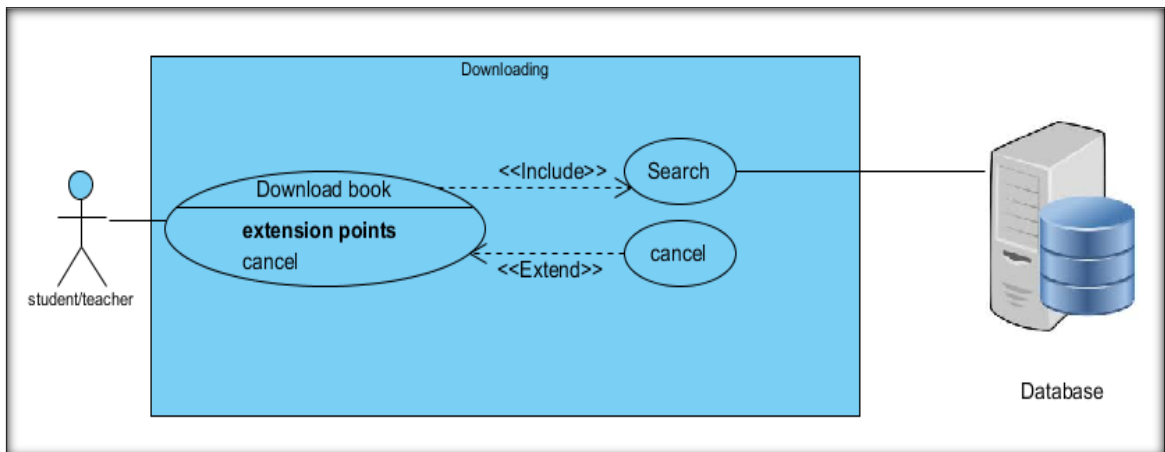


Figure 10:Download

Ordering:

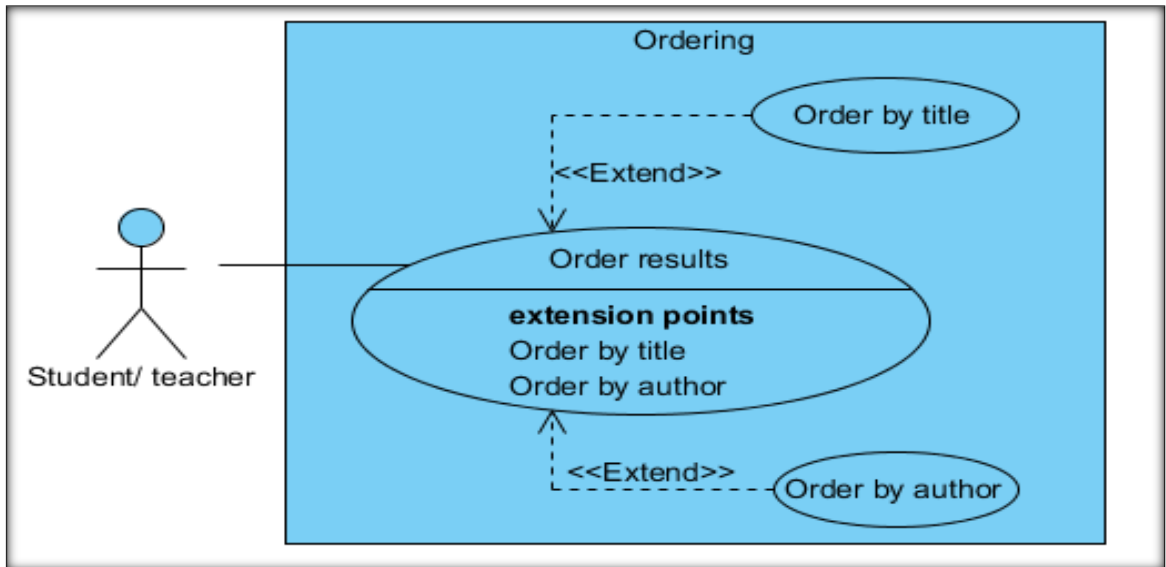


Figure 11:Order

Preview:

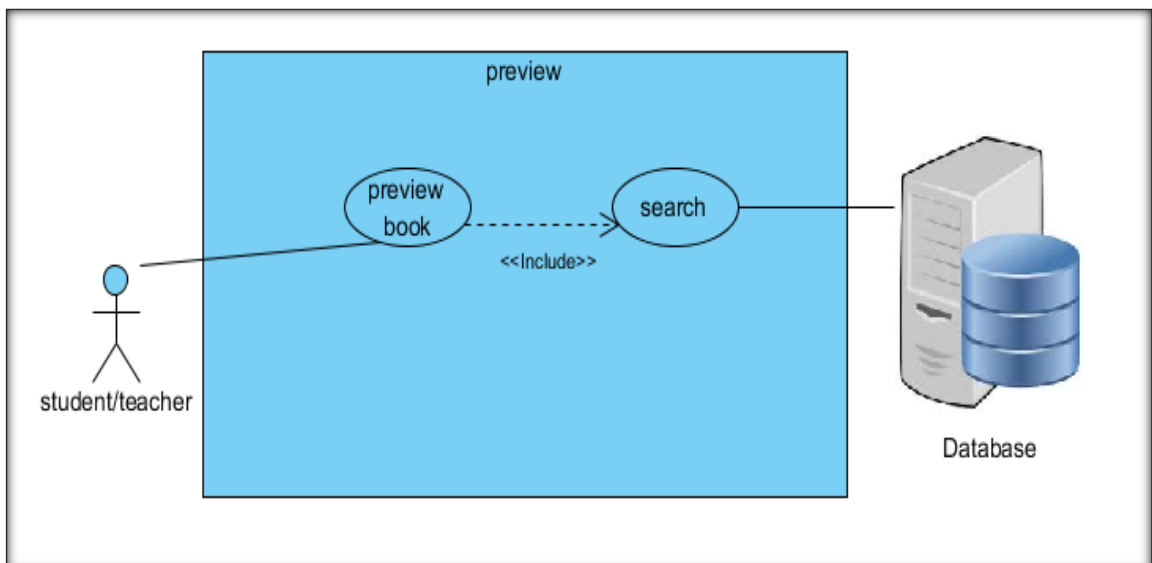


Figure 12:Preview

Reservation:

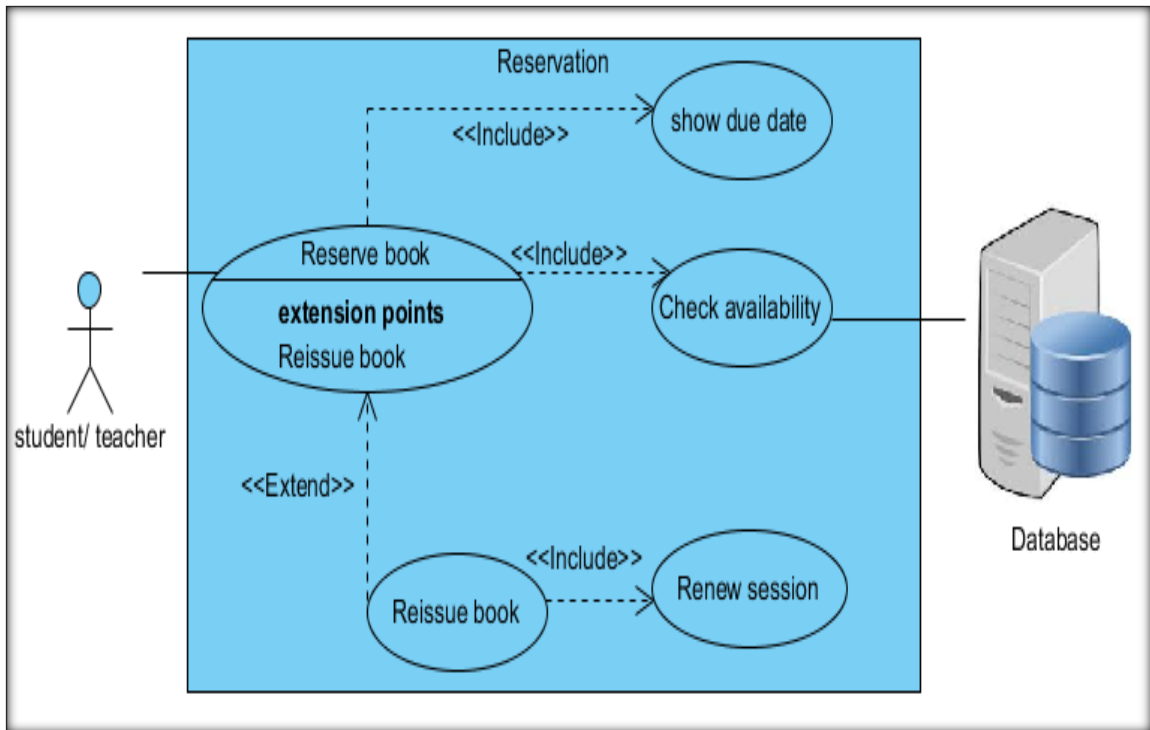


Figure 13:Reserve

Bookmark:

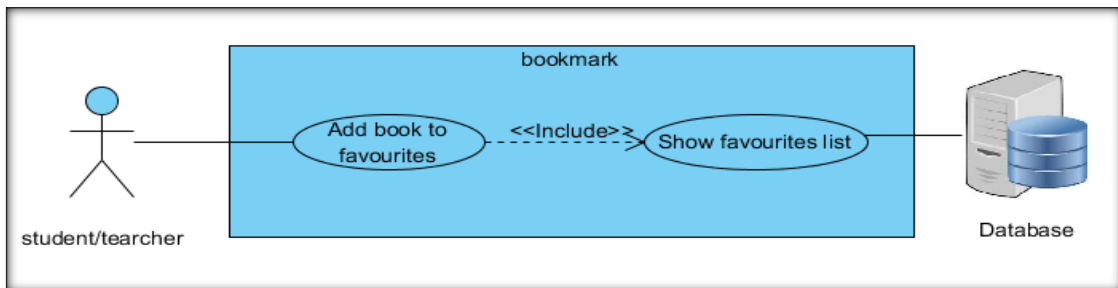


Figure 14:Bookmark

Sharing:

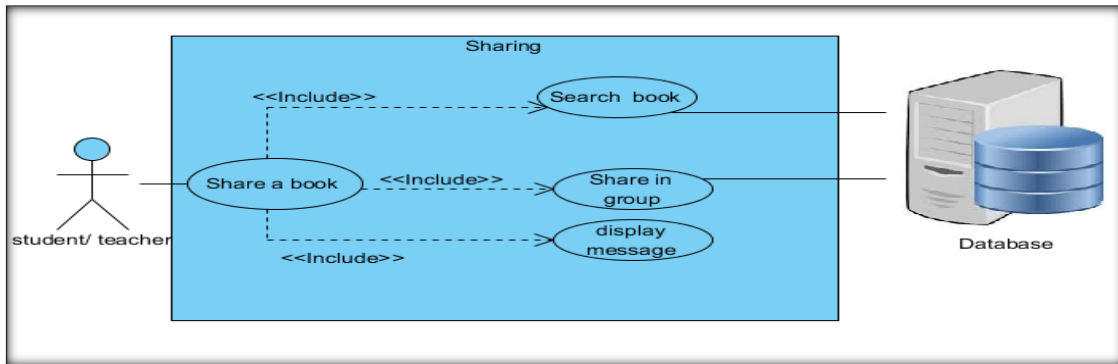


Figure 15:Sharing

Tagging:

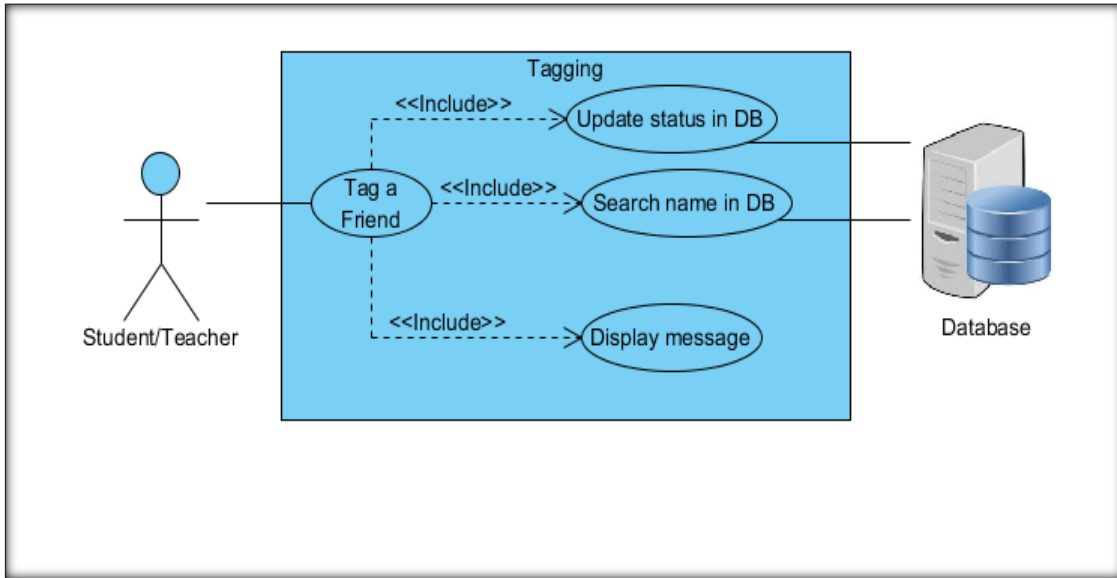


Figure 16: Tagging

Course Group:

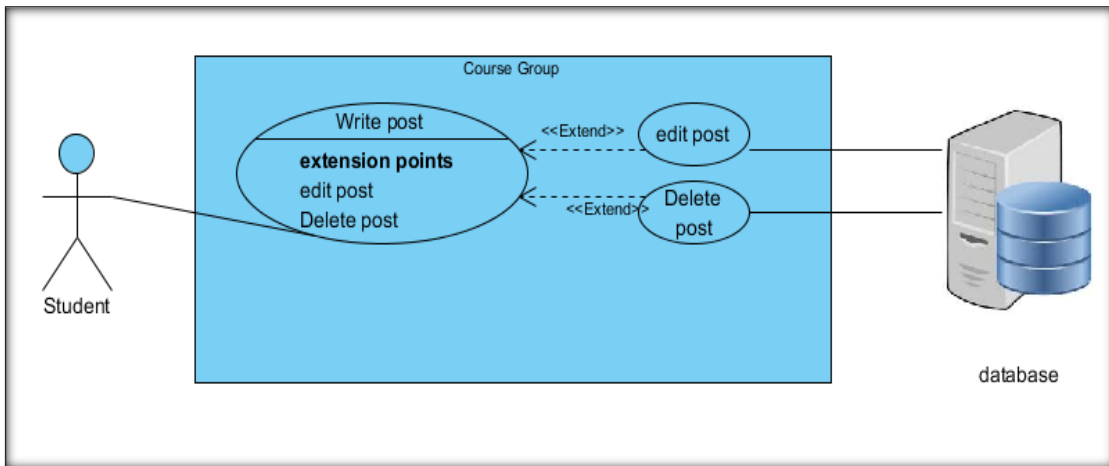


Figure 17: Course group

Mini Folder:

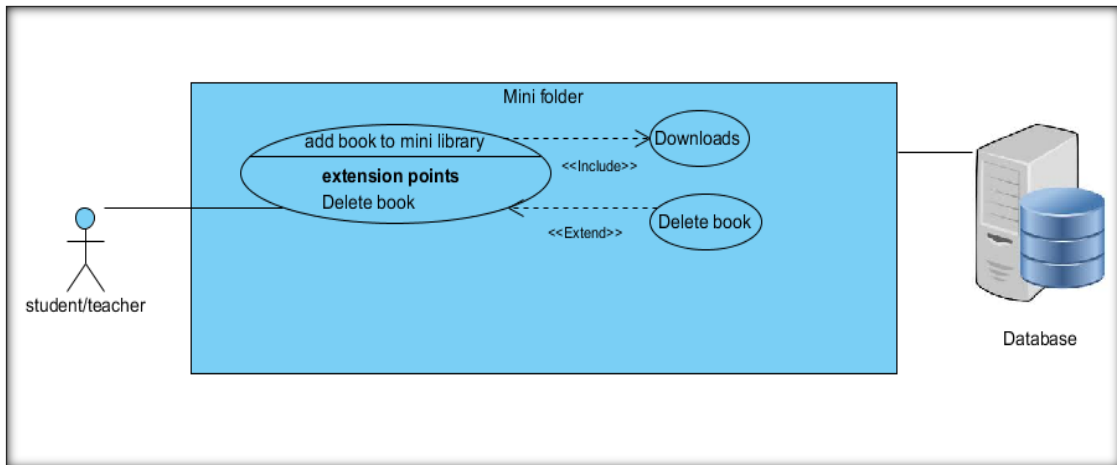


Figure 18:Mini library

Notifications:

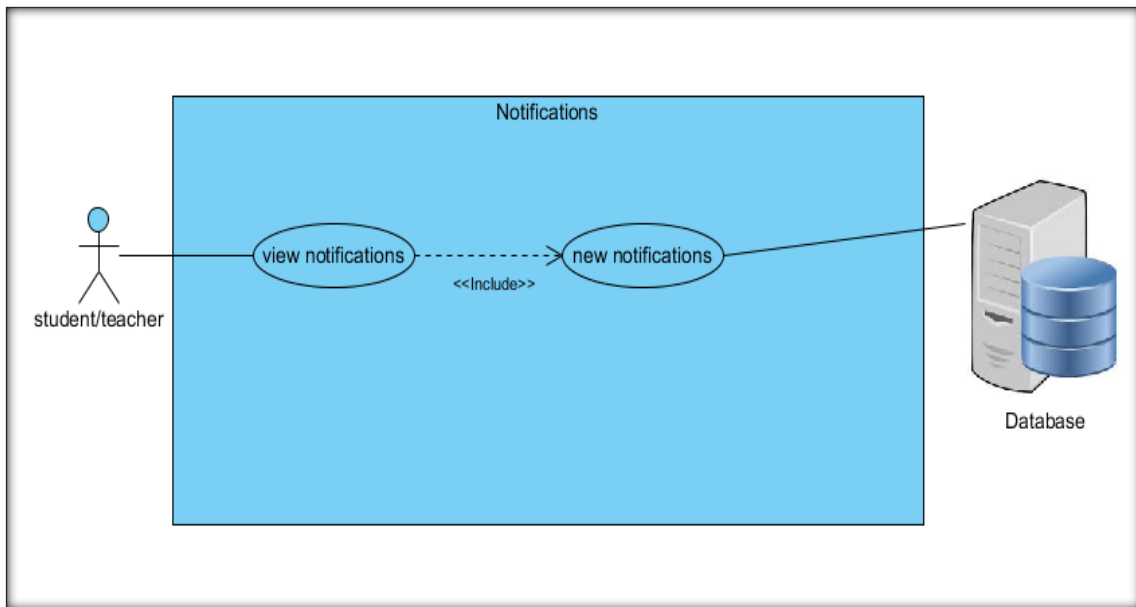


Figure 19:Notifications

My Account:

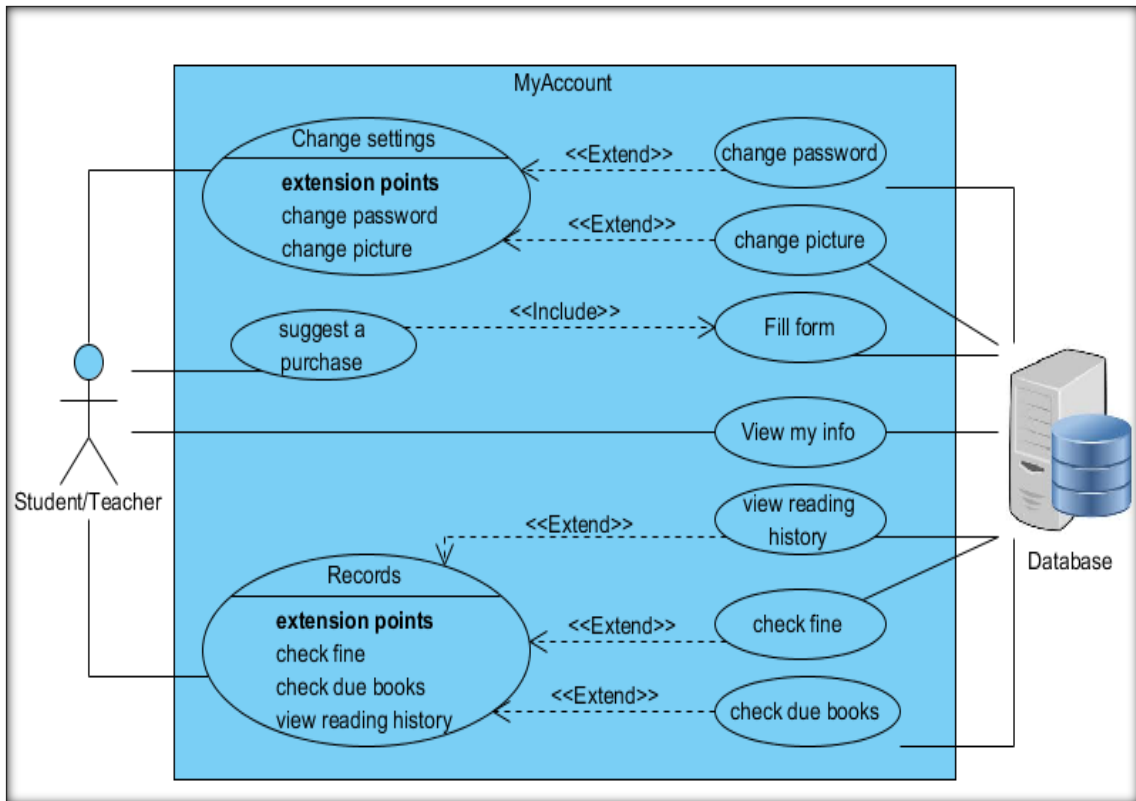


Figure 20:My account

Customization:

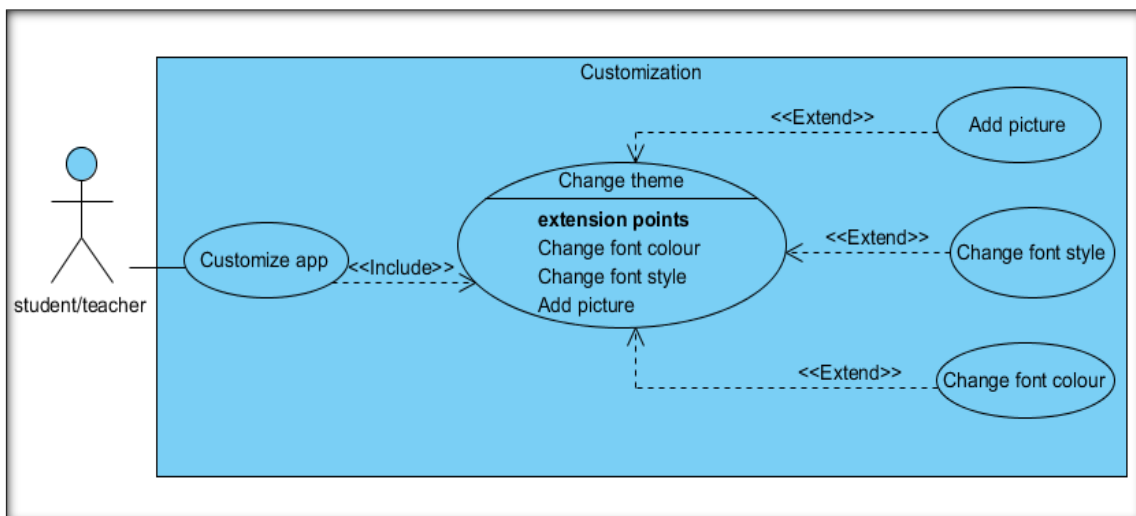


Figure 21:Customize

Help Info:

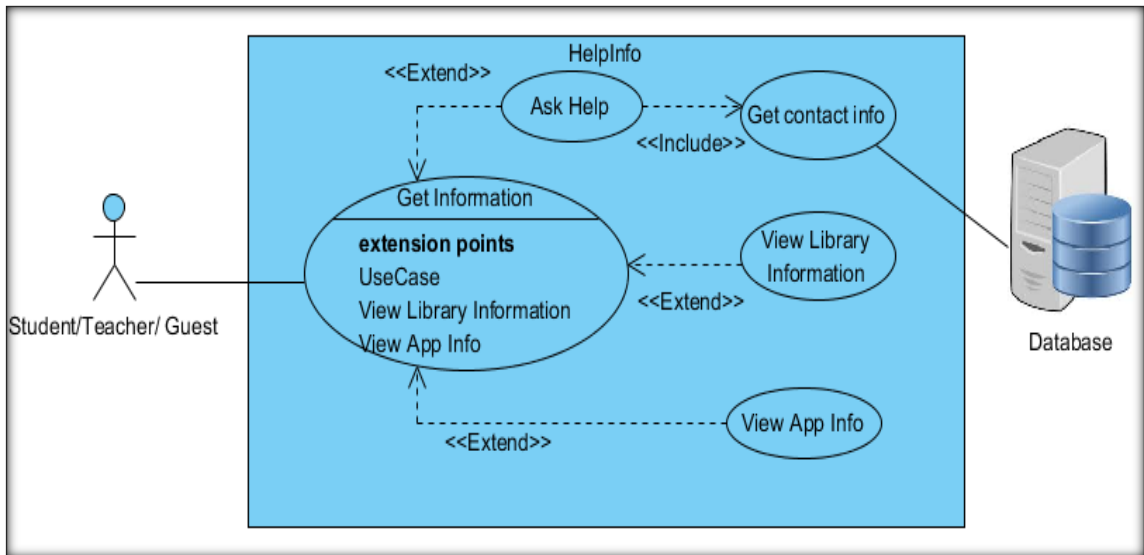


Figure 22:Help

4.4 Activity diagrams

Signup/Login:

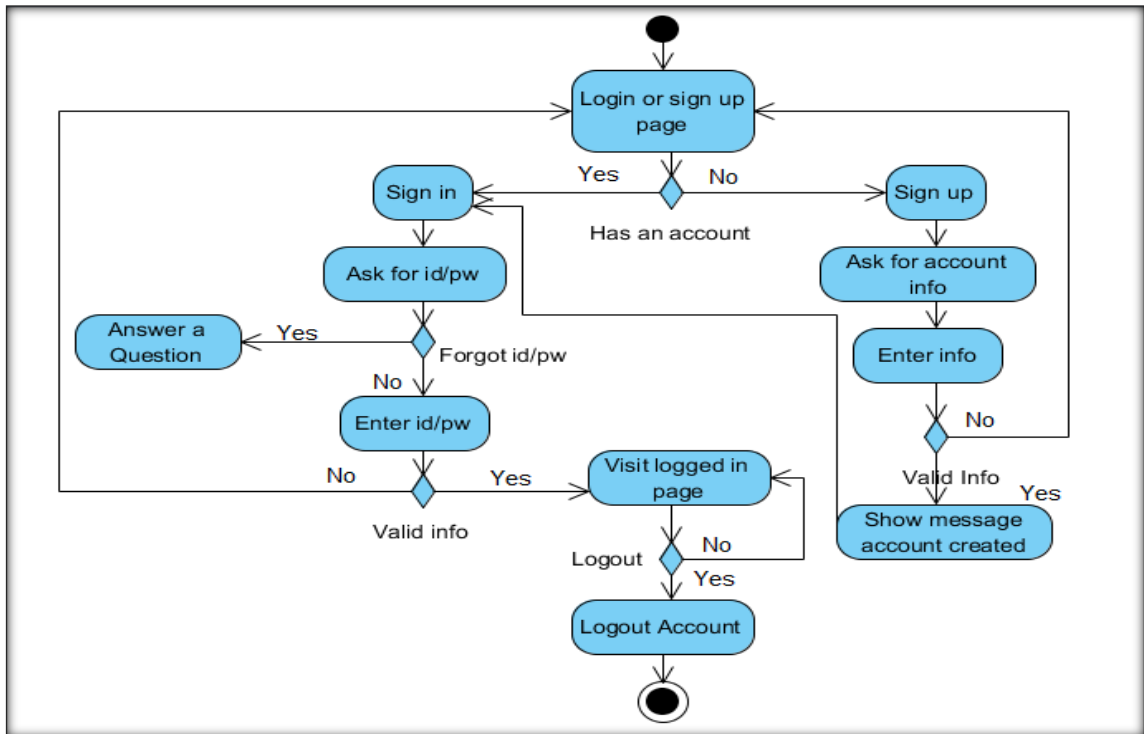


Figure 23 : Login and signup-Activity diagram

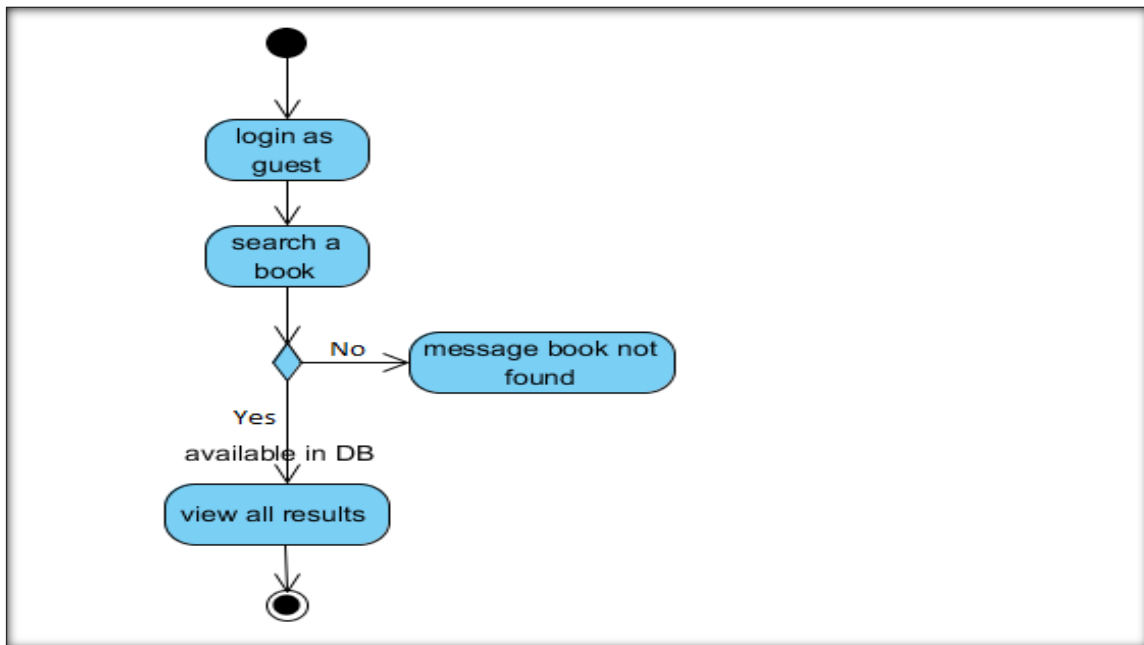


Figure 24 : Guest-Activity diagram

Search:

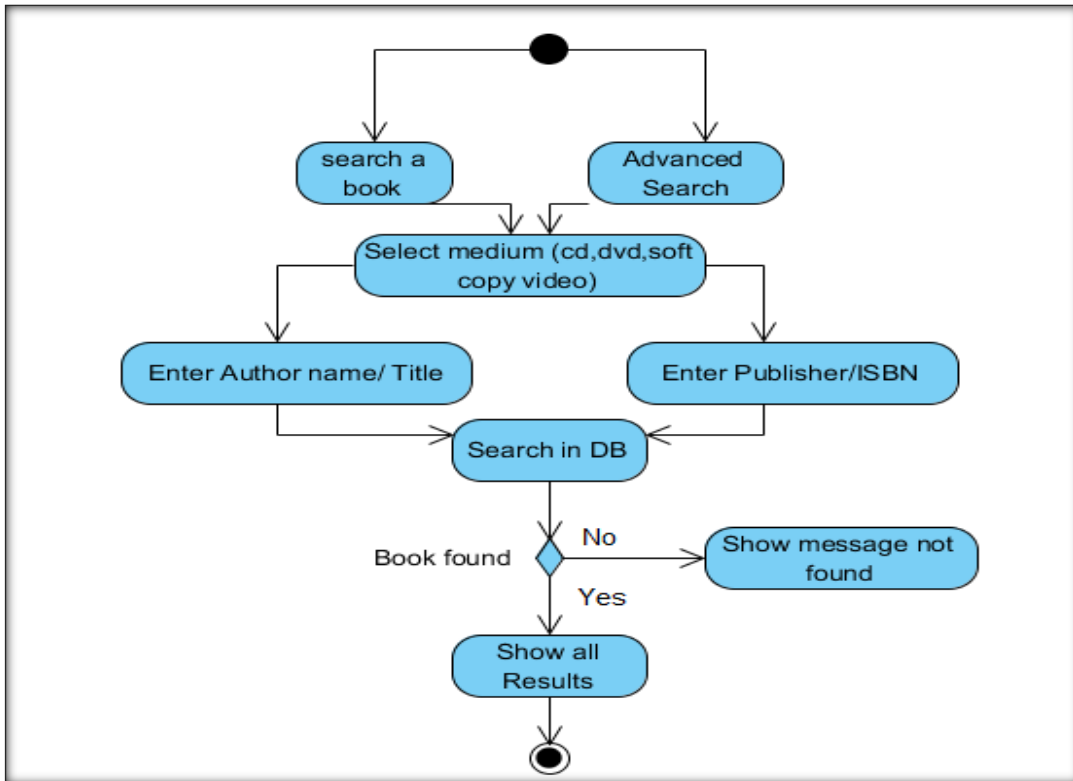


Figure 25 : Search-Activity diagram

Download and order results:

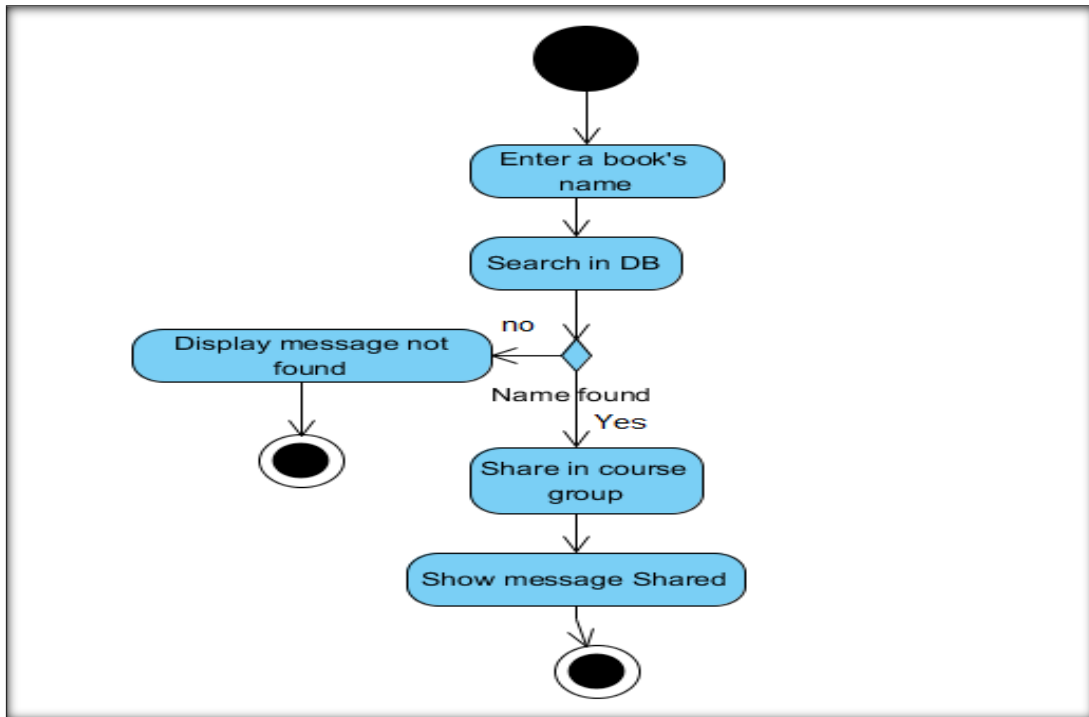


Figure 26 : Download:Activity diagram

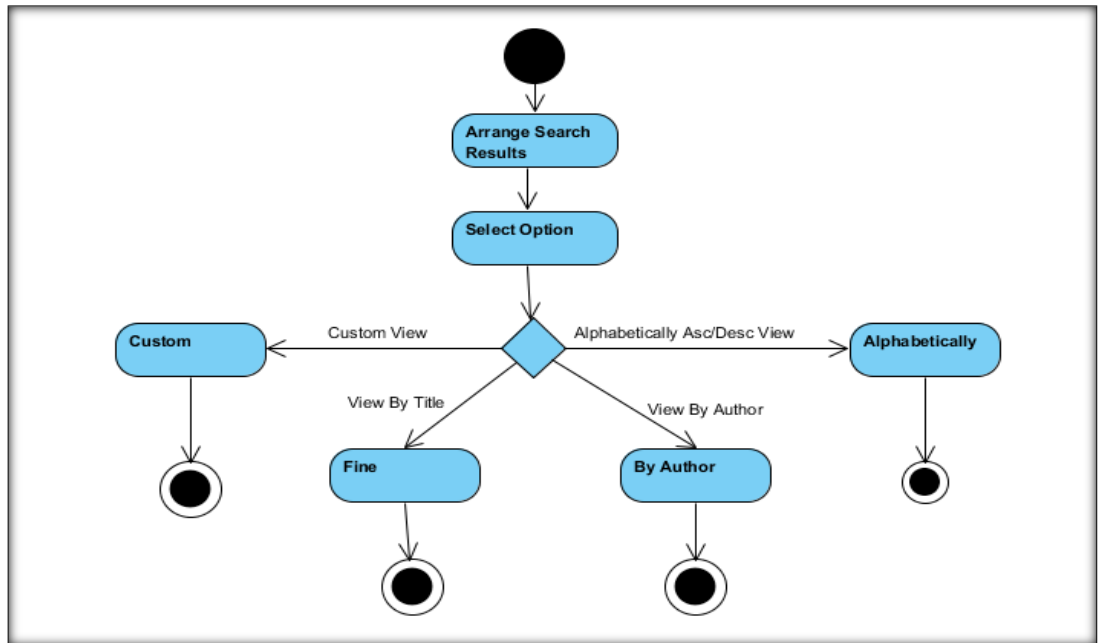


Figure 27 : Order results-Activity diagram

Preview and Reserve:

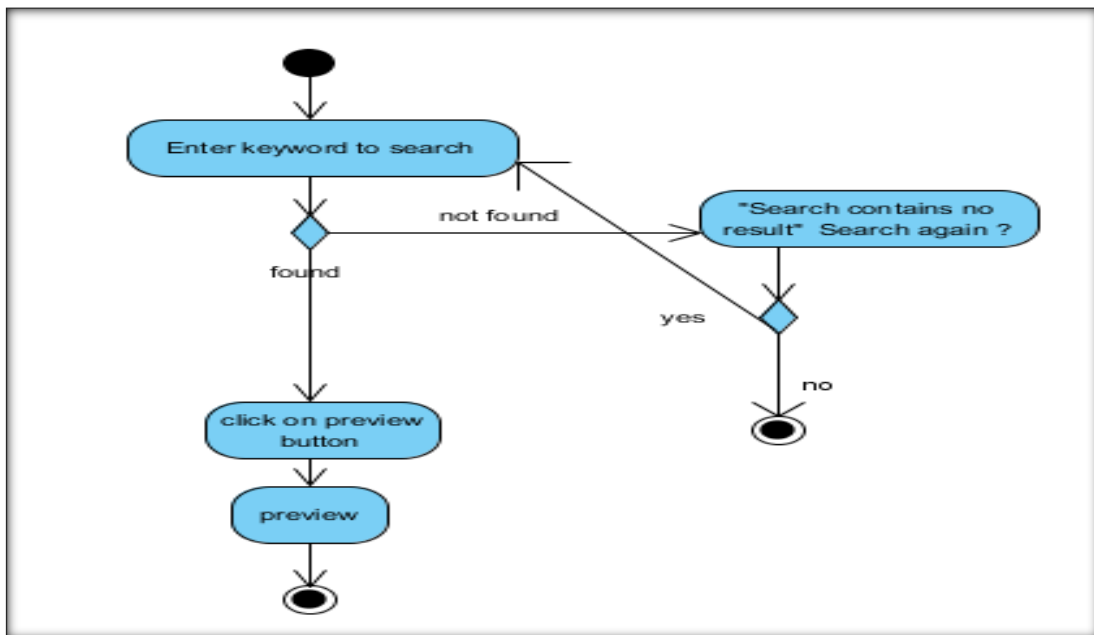


Figure 28: preview –Activity diagram

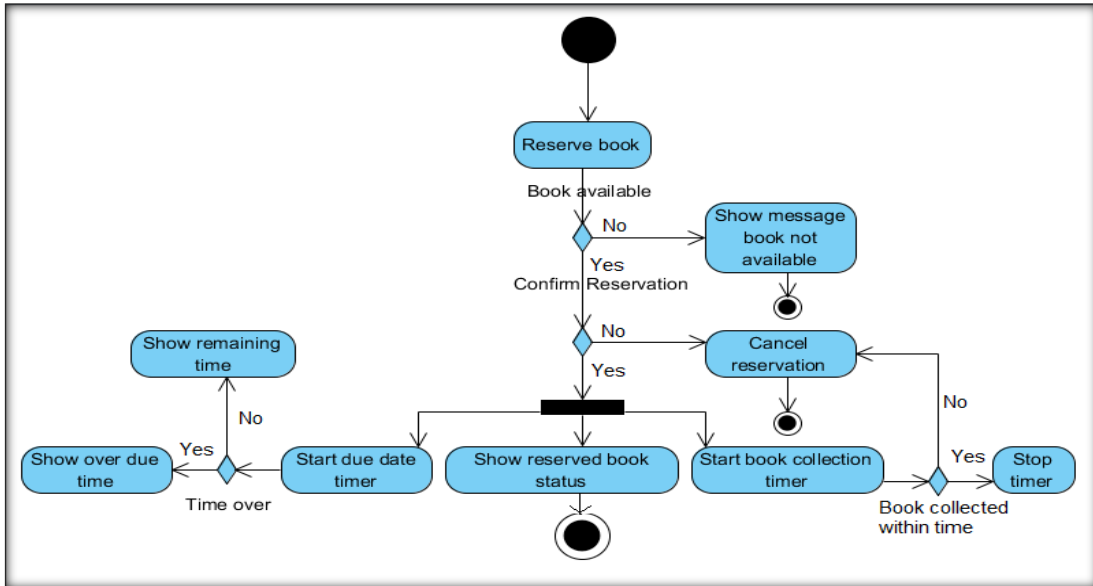


Figure 29 : Reserve-Activity diagram

Tag

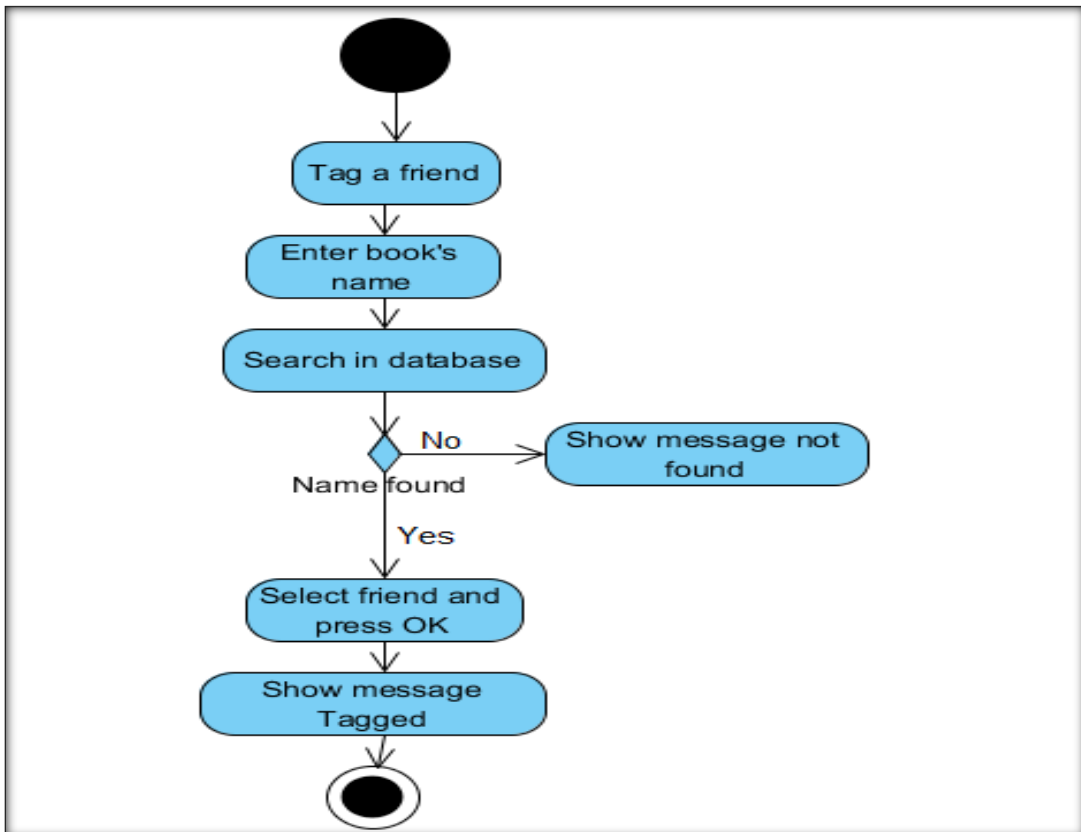


Figure 30 : Tag-Activity diagram

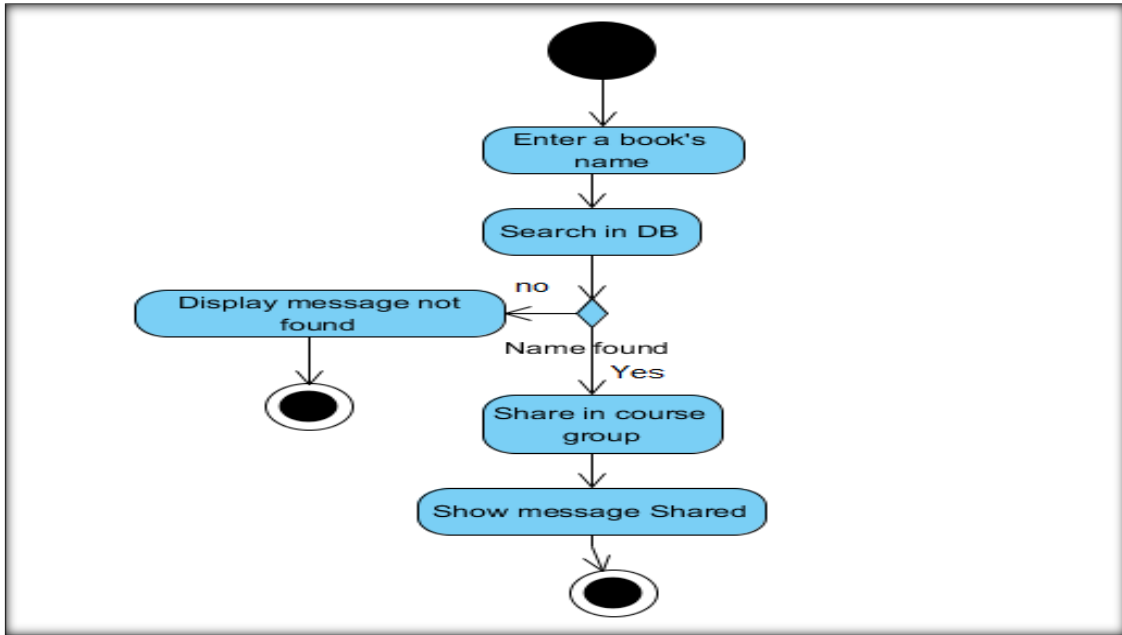


Figure 31

Figure 32 : Share-Activity diagram

Group study

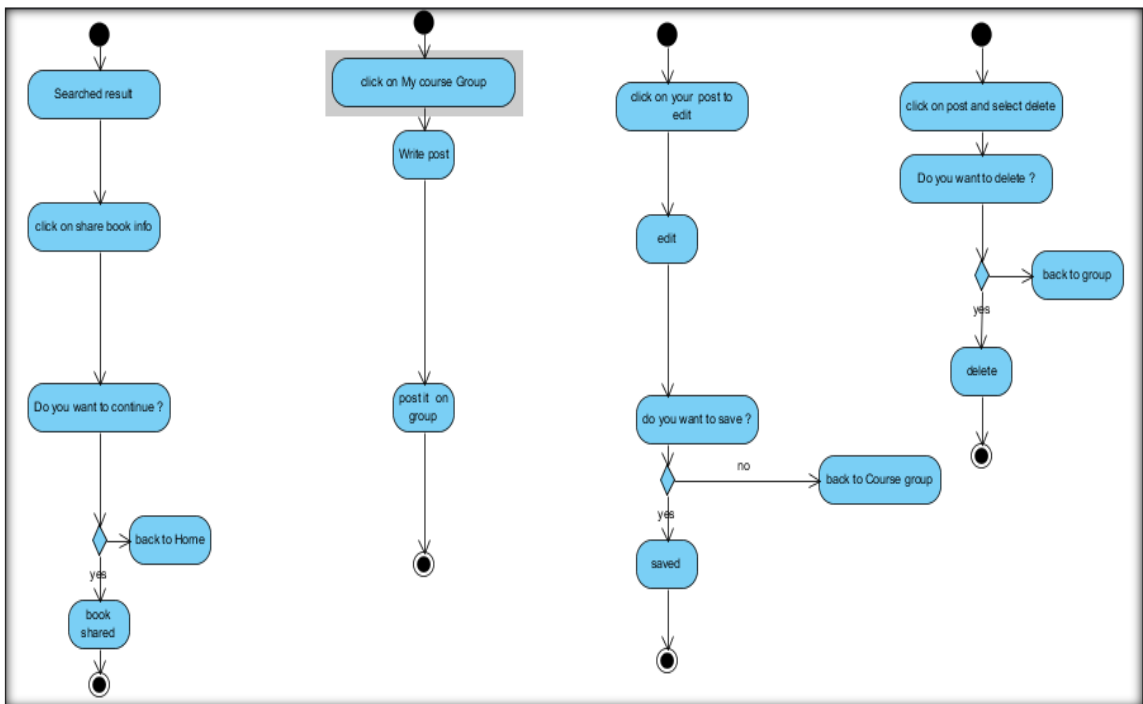


Figure 33 : Course Group-Activity diagram

Reissue

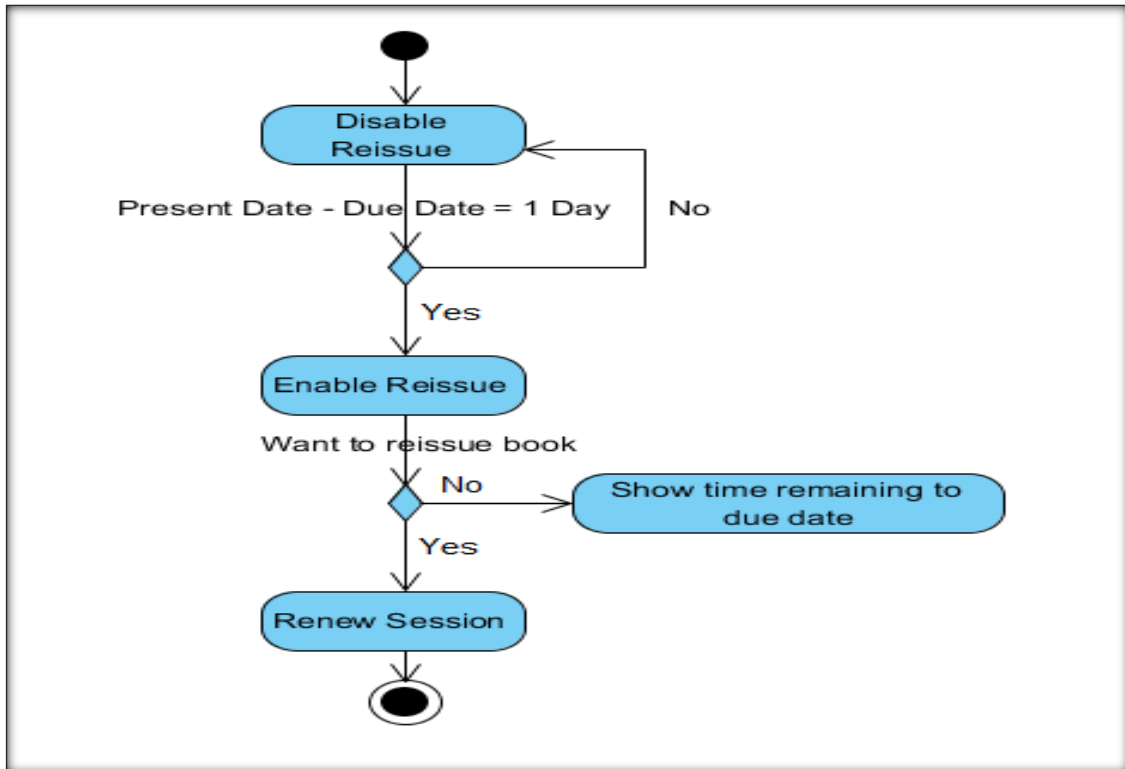


Figure 34 : Reissue-Activity diagram

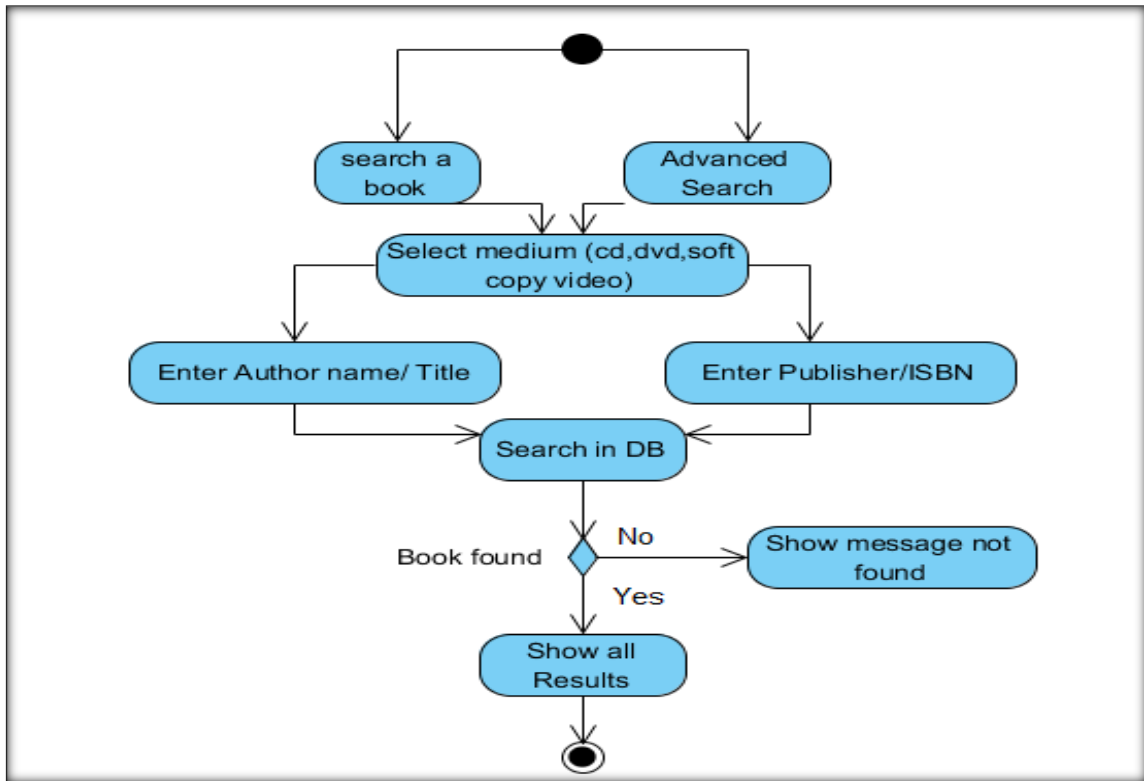


Figure 35 : Search-Activity diagram

Download and order results

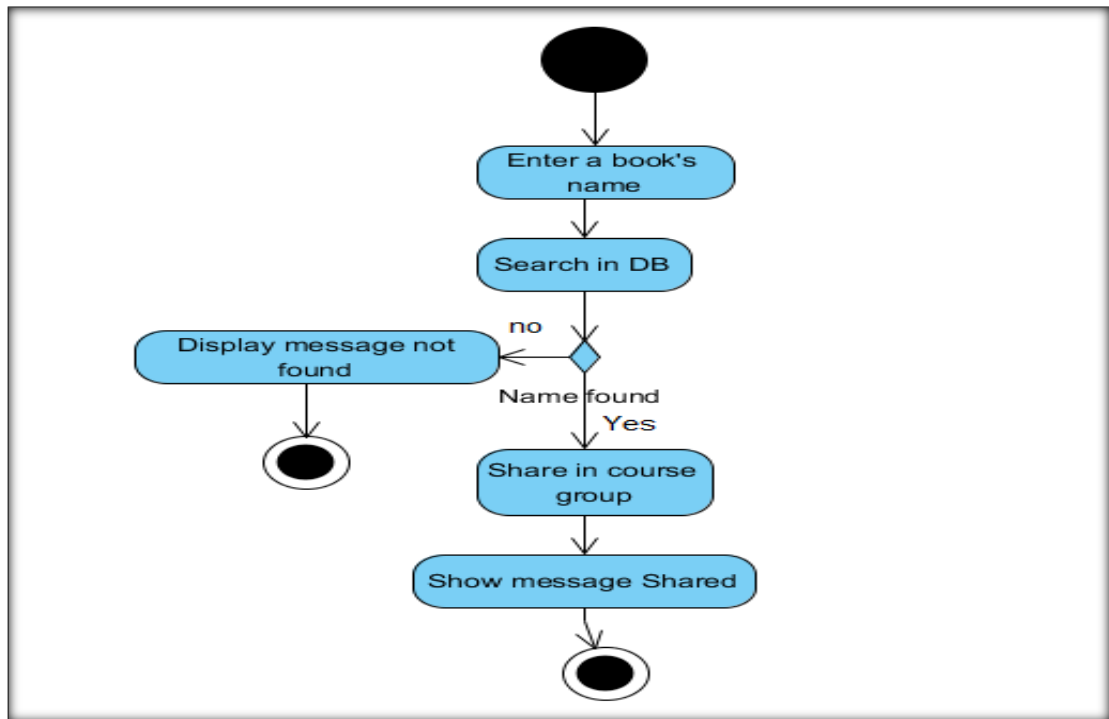


Figure 36 : Download:Activity diagram

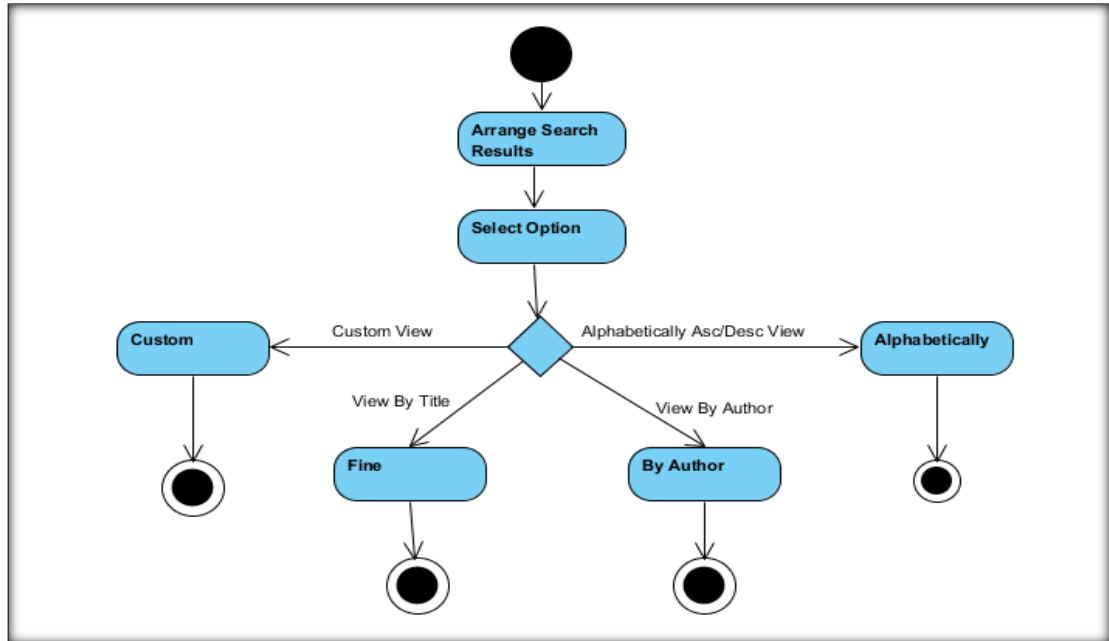


Figure 37 : Order results-Activity diagram

Preview and Reserve

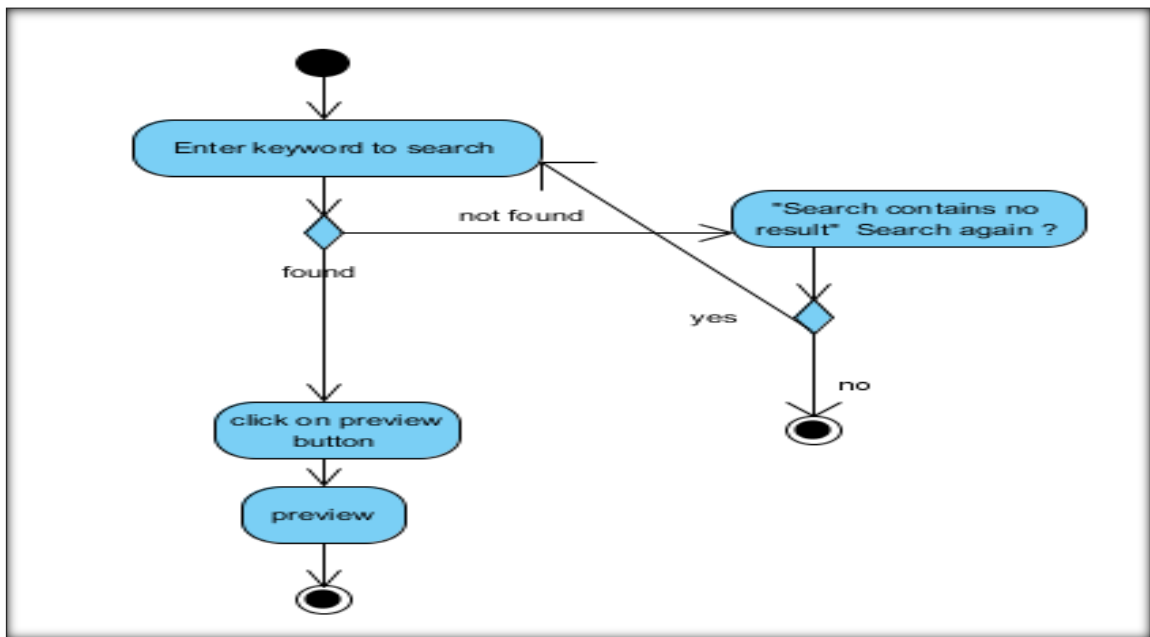


Figure 38: preview –Activity diagram

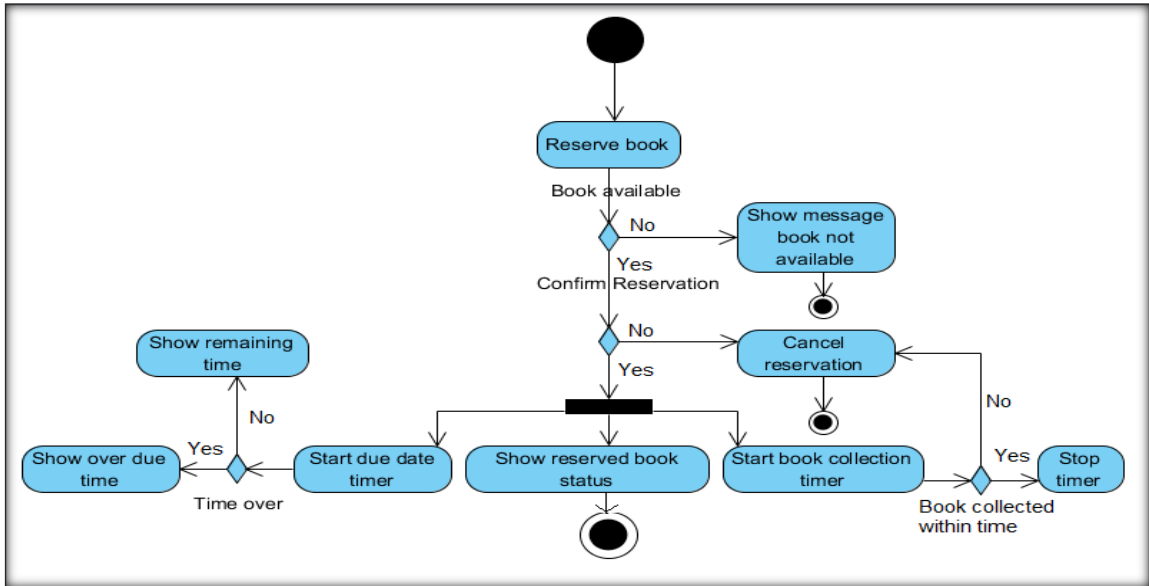


Figure 39 : Reserve-Activity diagram

- Tag

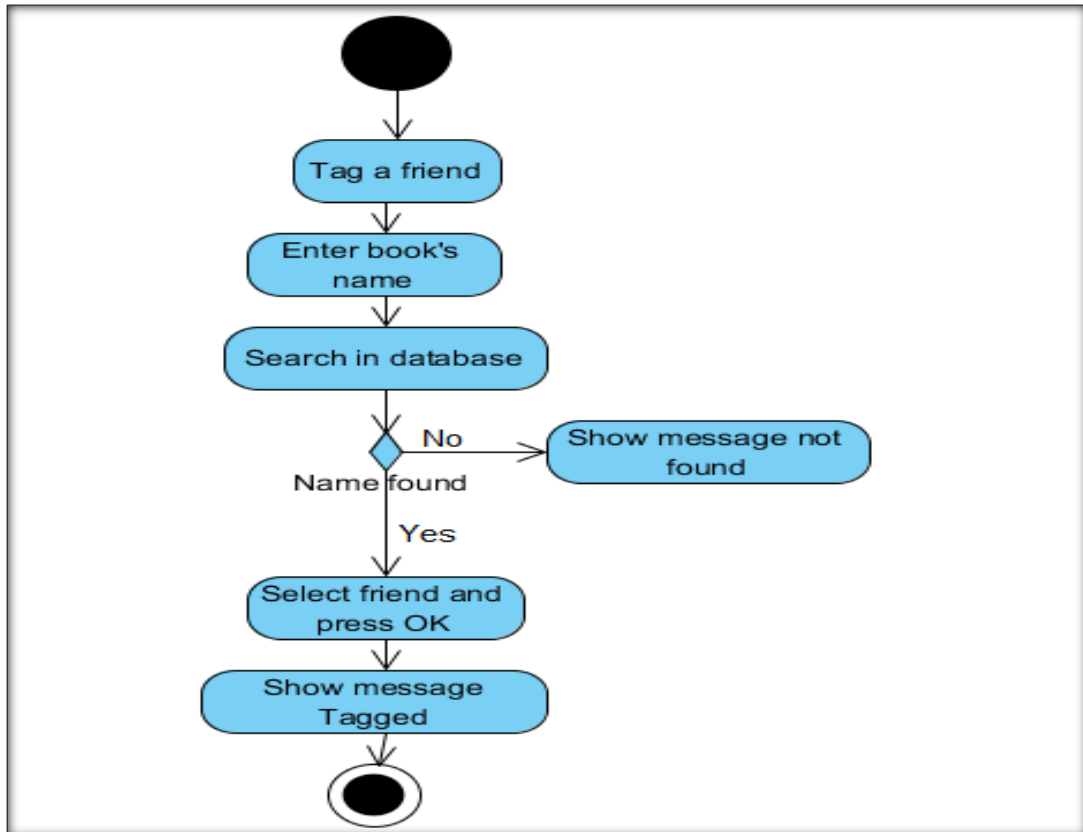


Figure 40 : Tag-Activity diagram

Group study

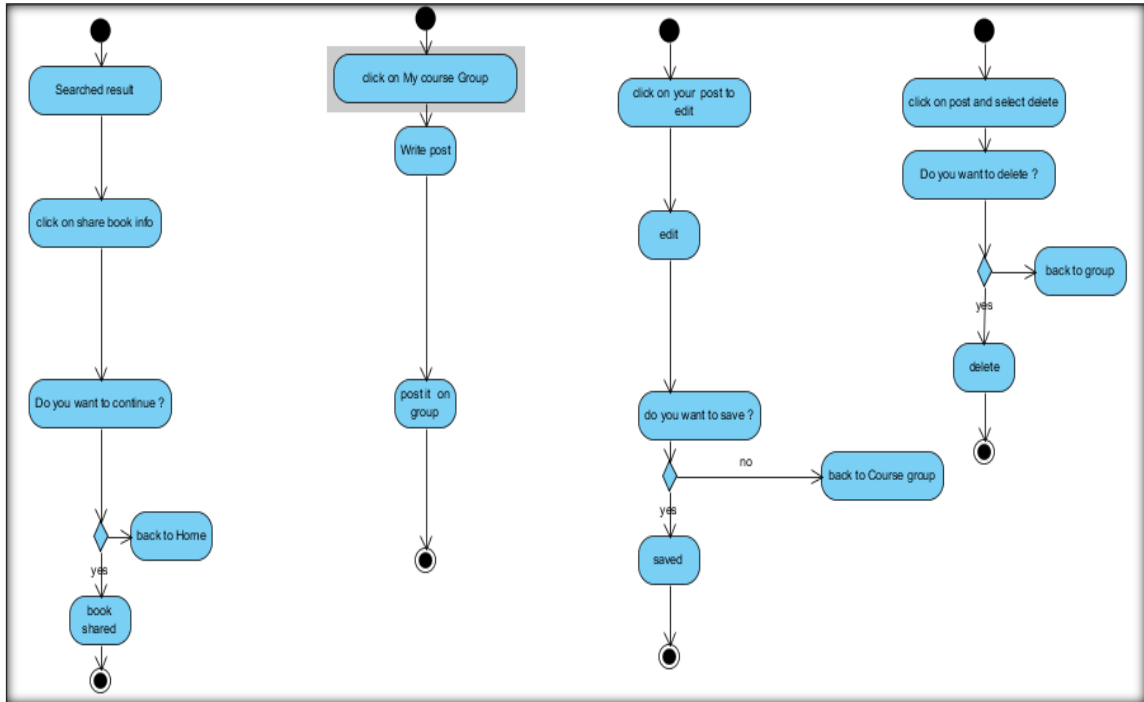


Figure 41:Course group

Reissue

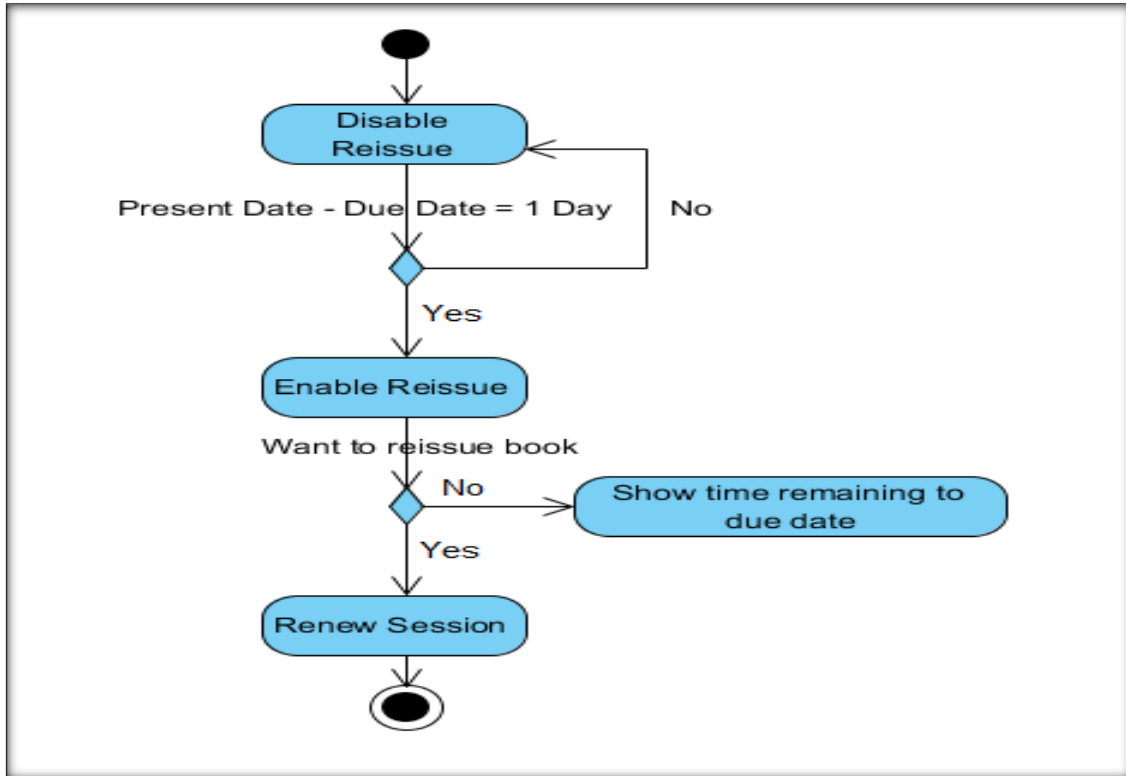


Figure 42

Mini library

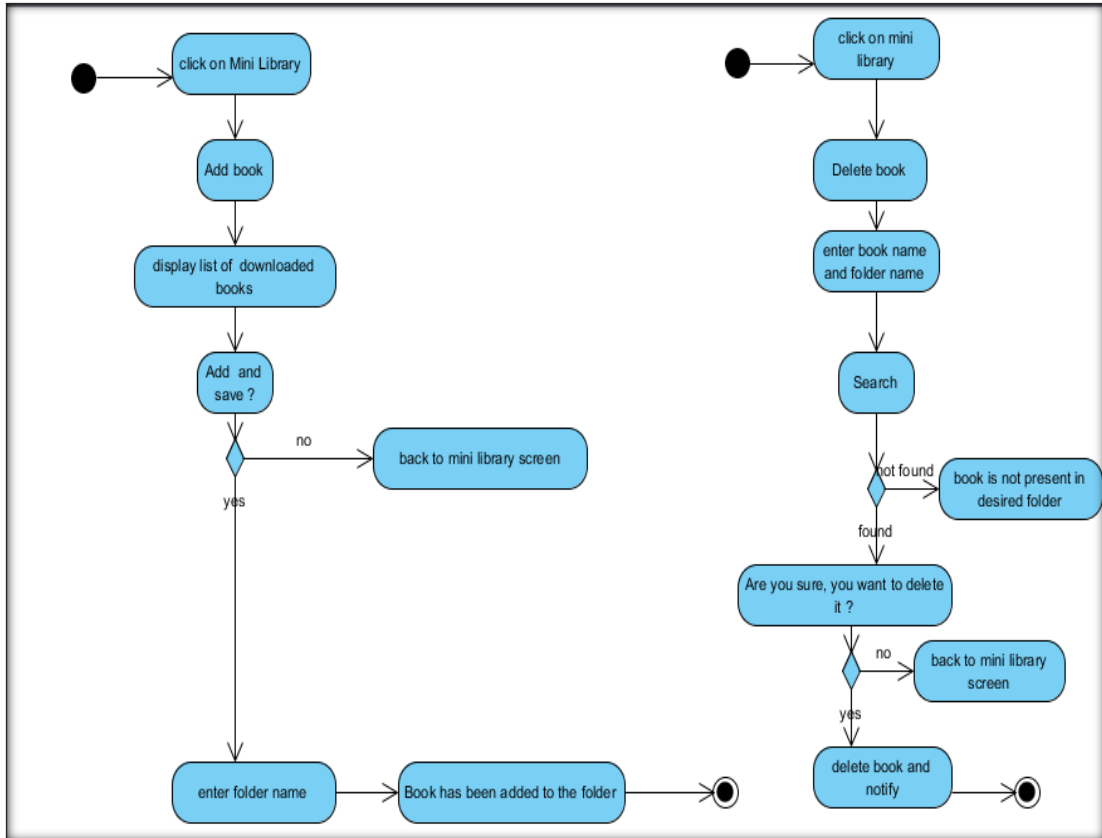


Figure 43 : Mini library-Activity diagram

Notification-manager

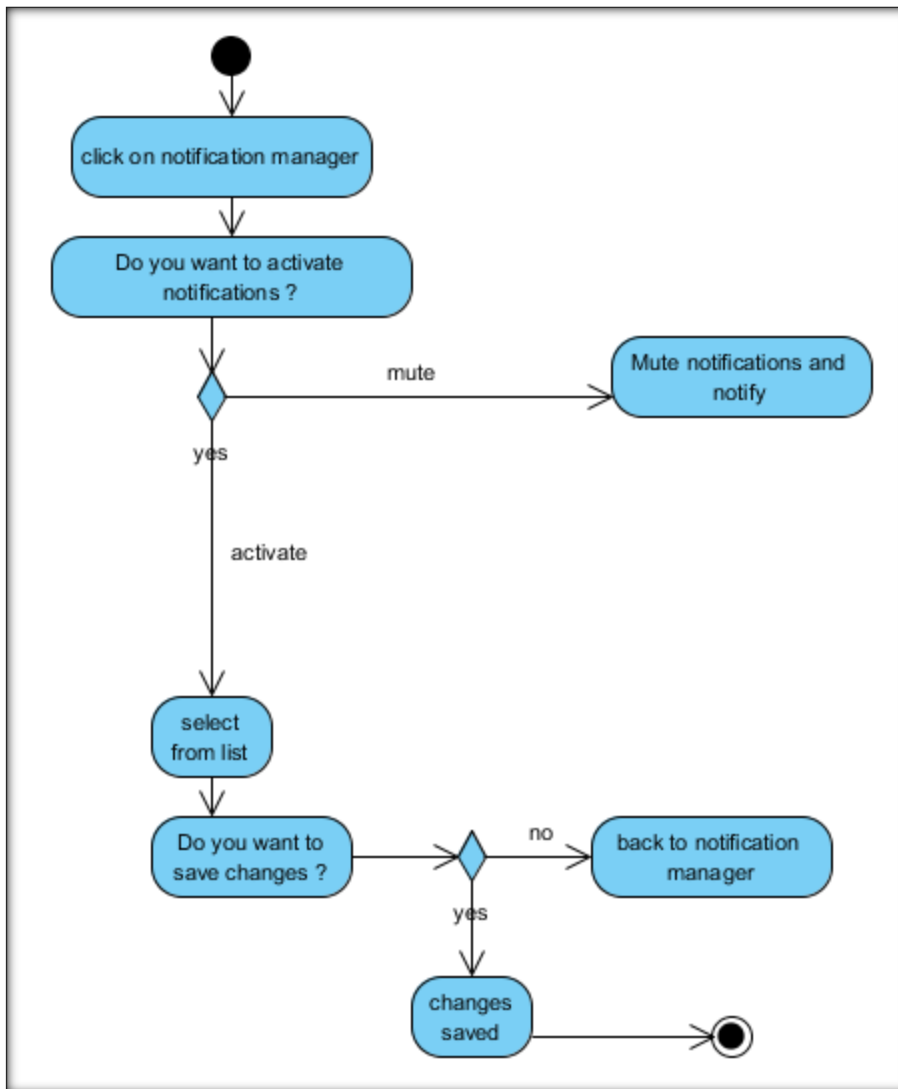
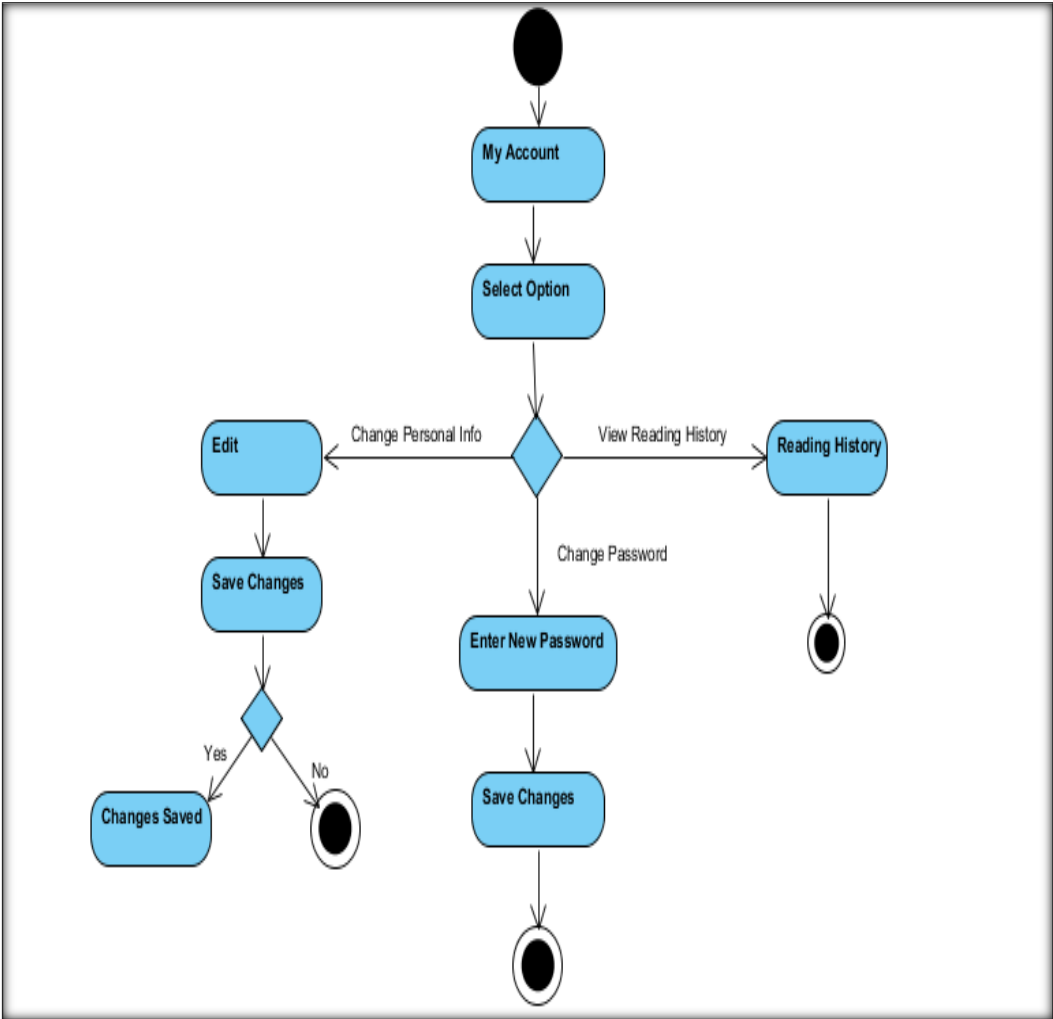


Figure 44 : Notification manager-Activity diagram

Account

Figure 45 : Account-Activity diagram



- Customization and Help

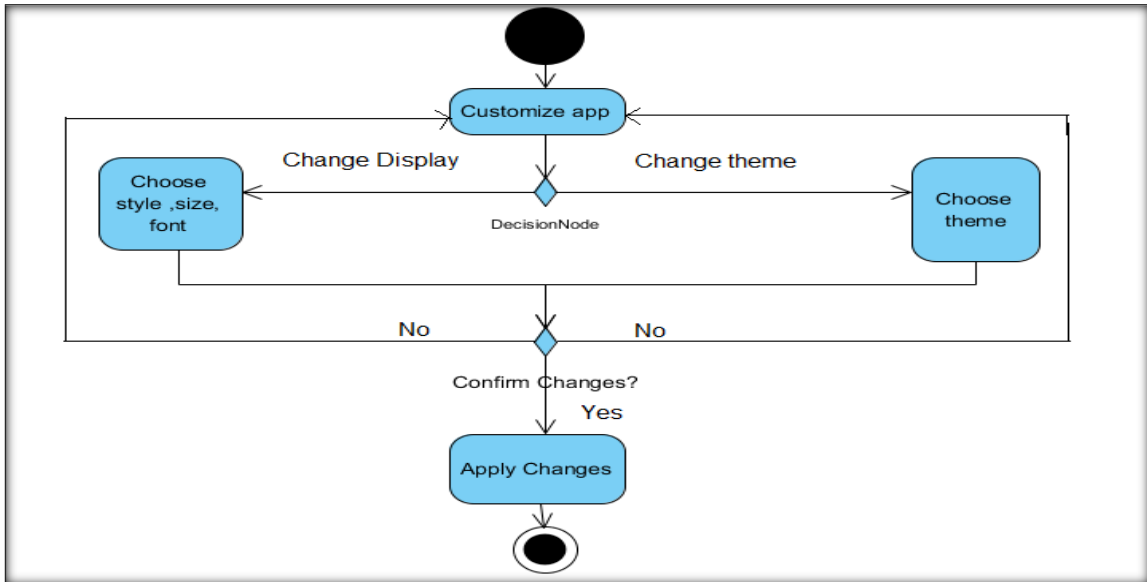


Figure 46: Customization –Activity diagram

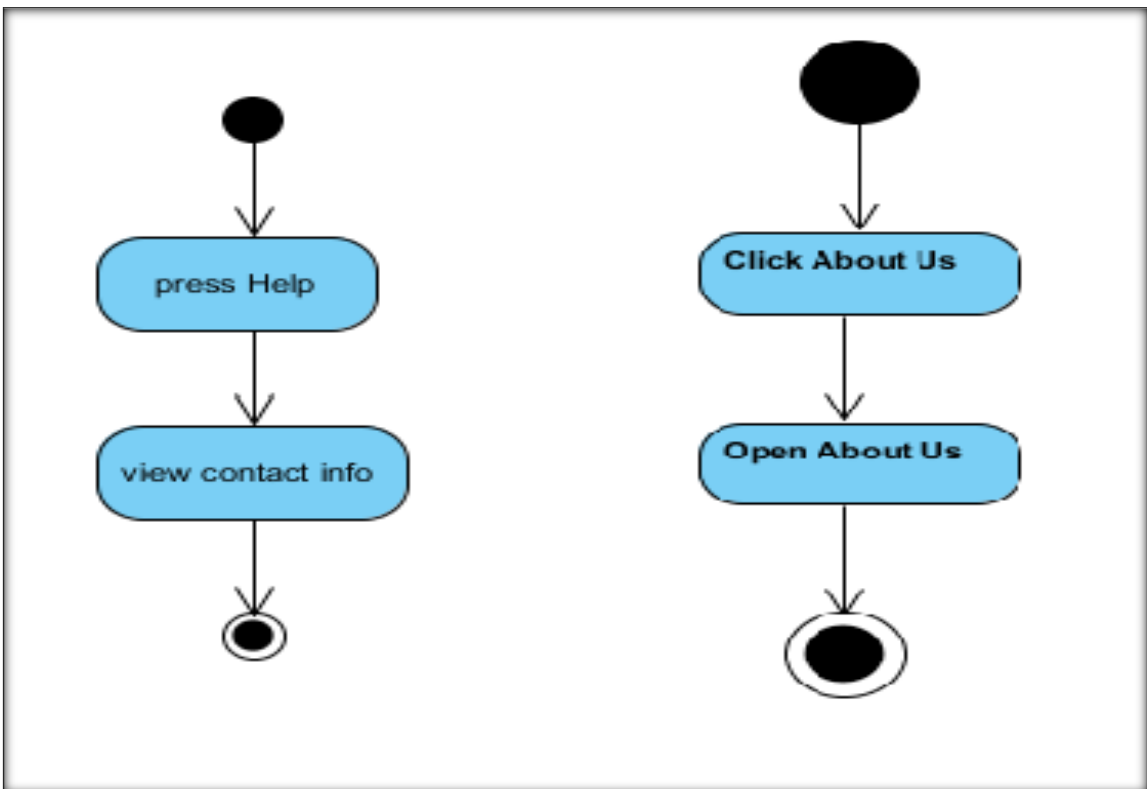


Figure 47 : Help-Activity diagram

4.5 Sequence diagrams

Login and signup

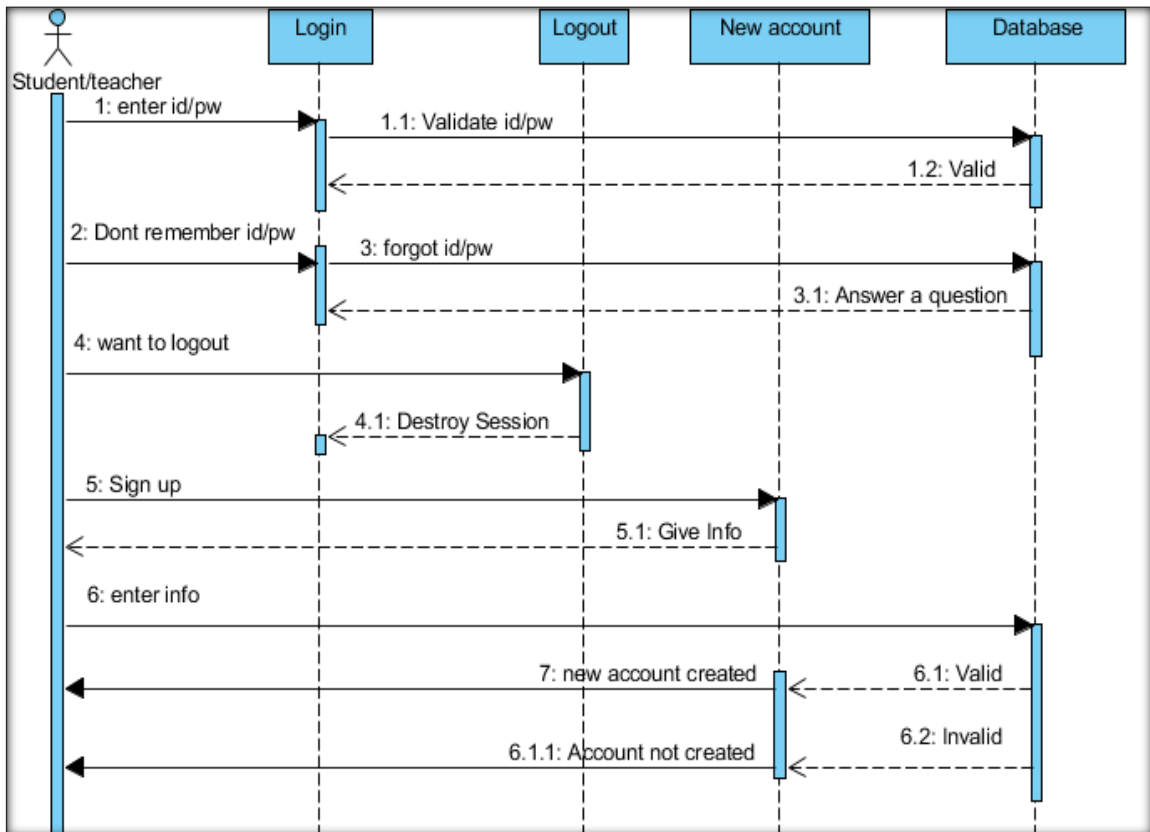


Figure 48 : Login and signup-Sequence diagram

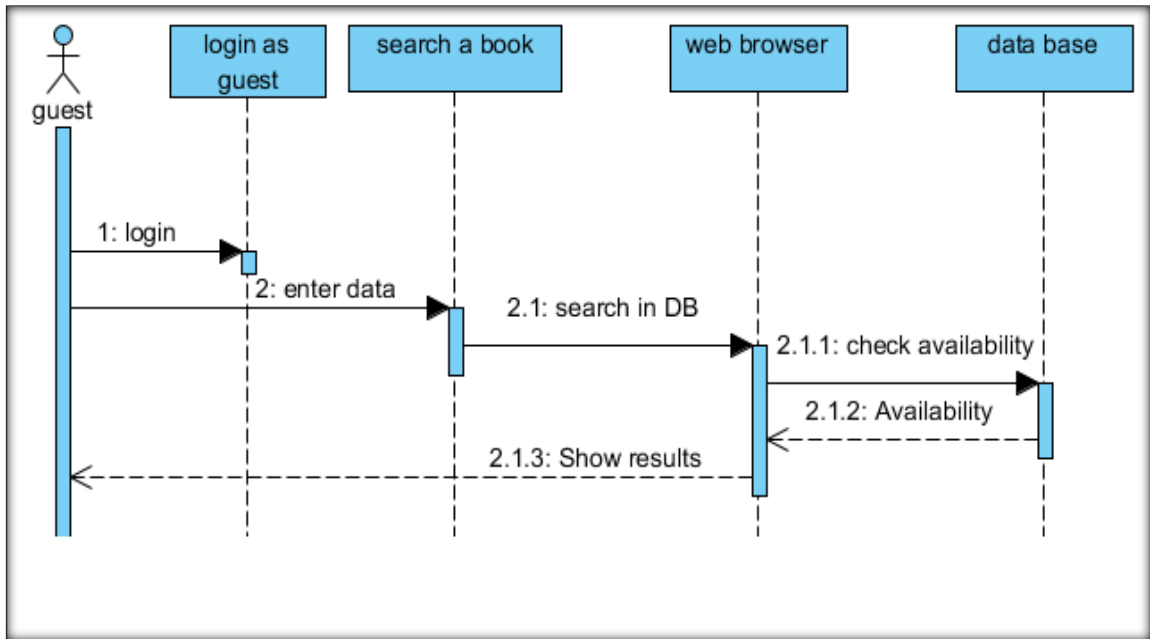


Figure 49 : Guest-Sequence diagram

Search

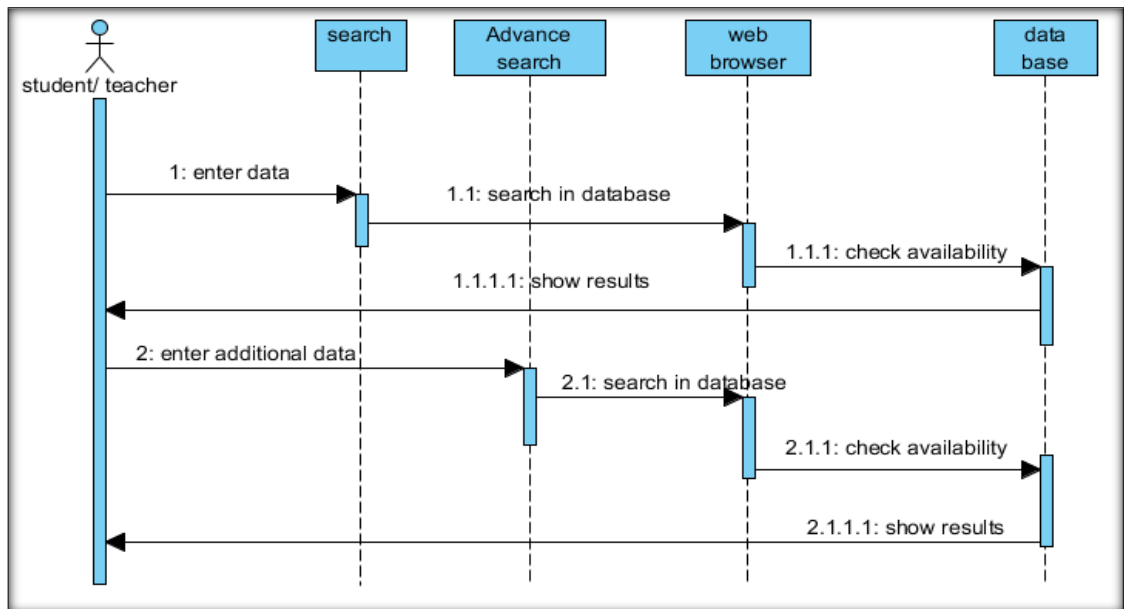


Figure 50 : Search-Sequence diagram

Download and order results

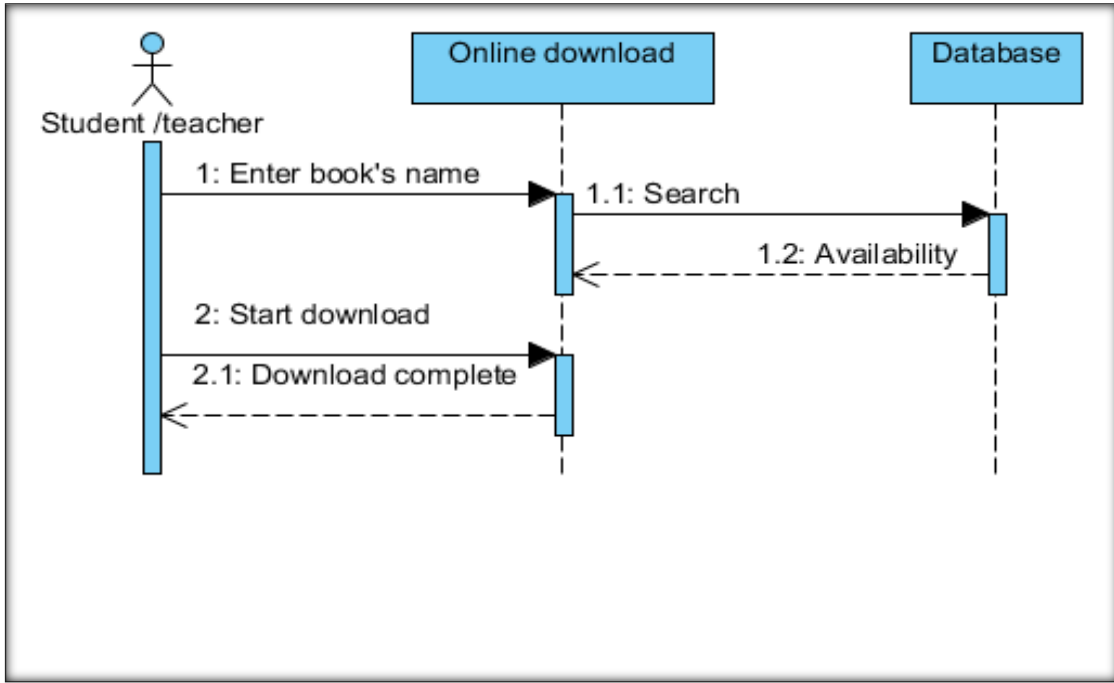


Figure 51 : Downloading-Sequence diagram

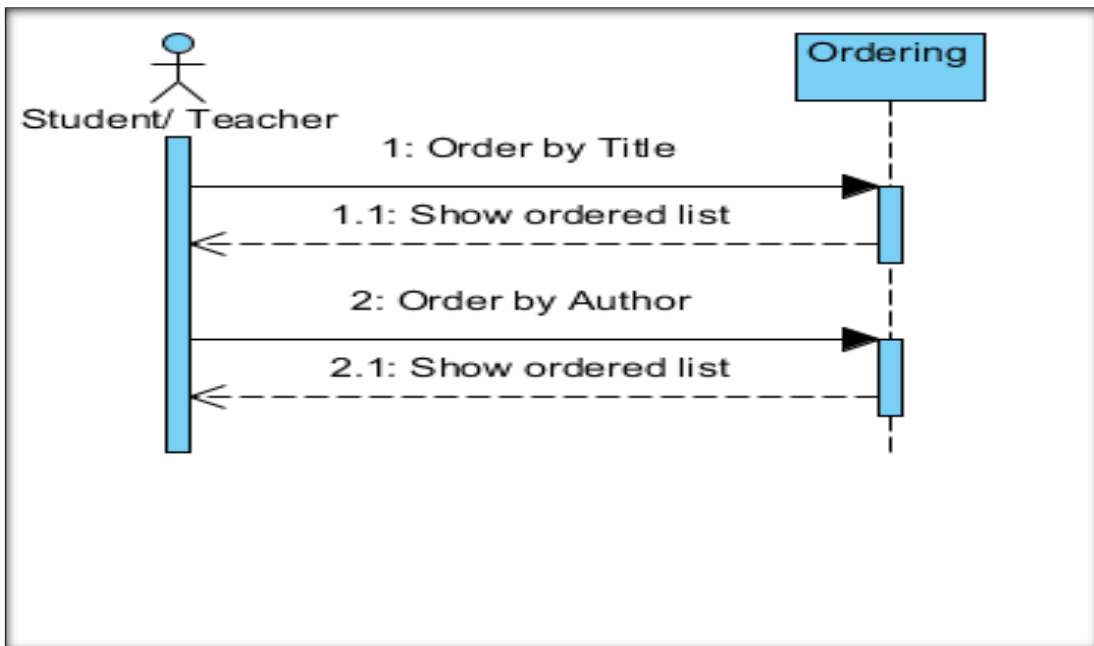


Figure 41: Order results-Sequence diagram

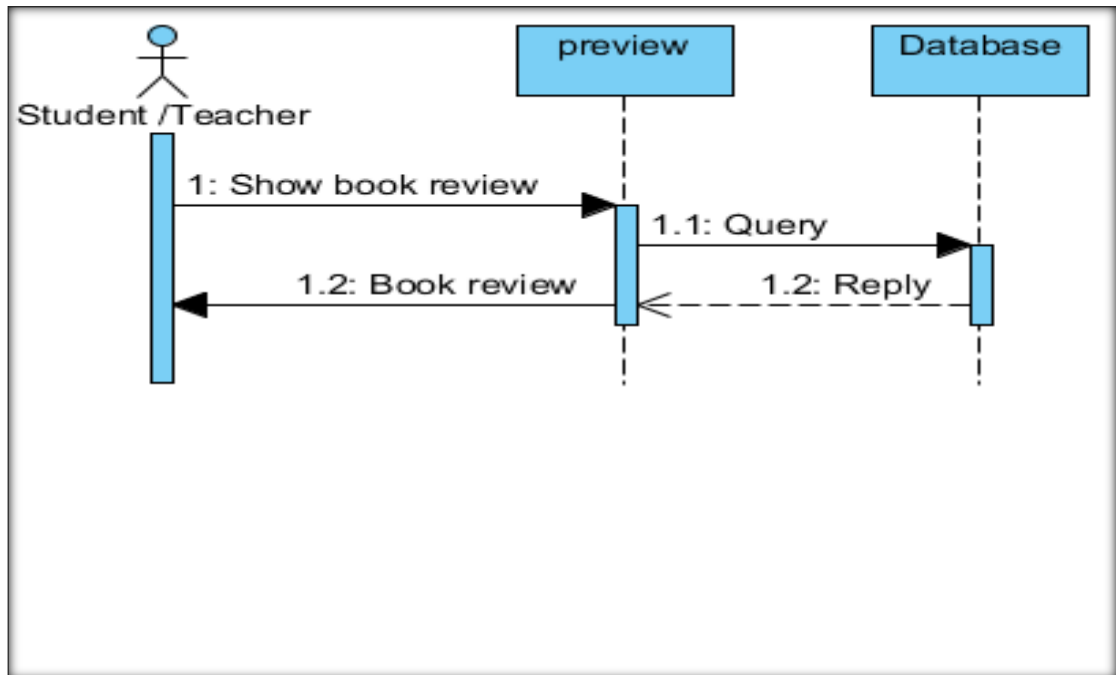


Figure 52 : Preview-Sequence diagram

Reserve /Reissue

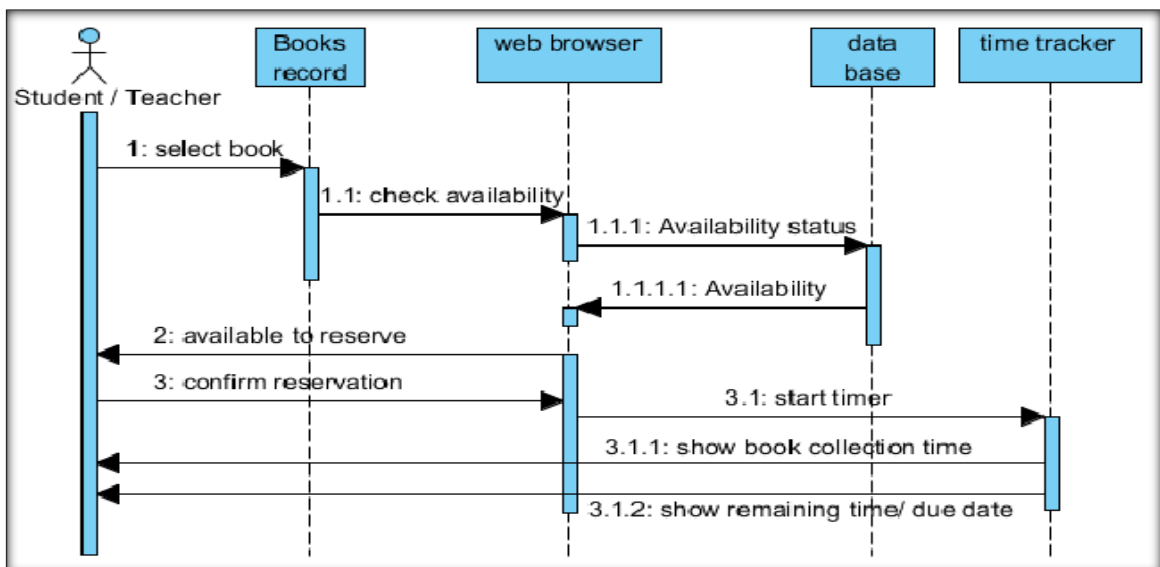


Figure 53 : Reserve-Sequence diagram

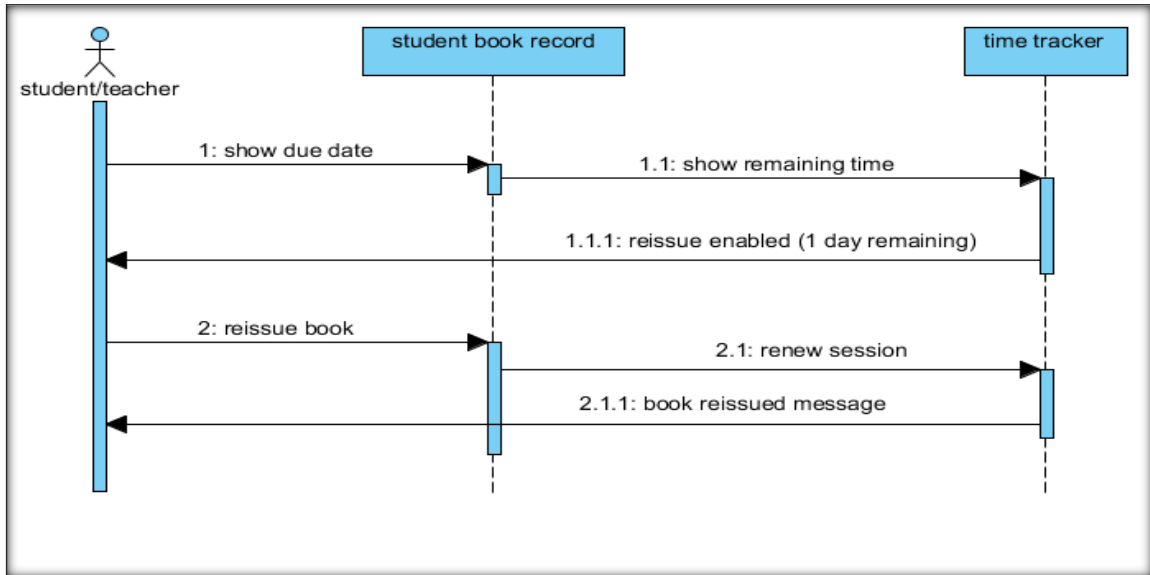


Figure 54 : Reissue-Sequence diagrams

Bookmark

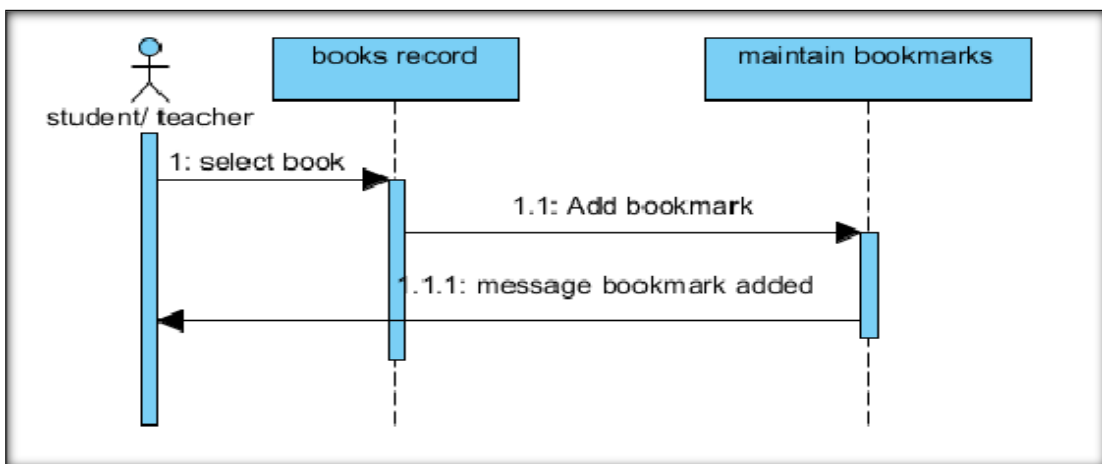


Figure 55 : Bookmarks-Sequence diagram

Tag and share

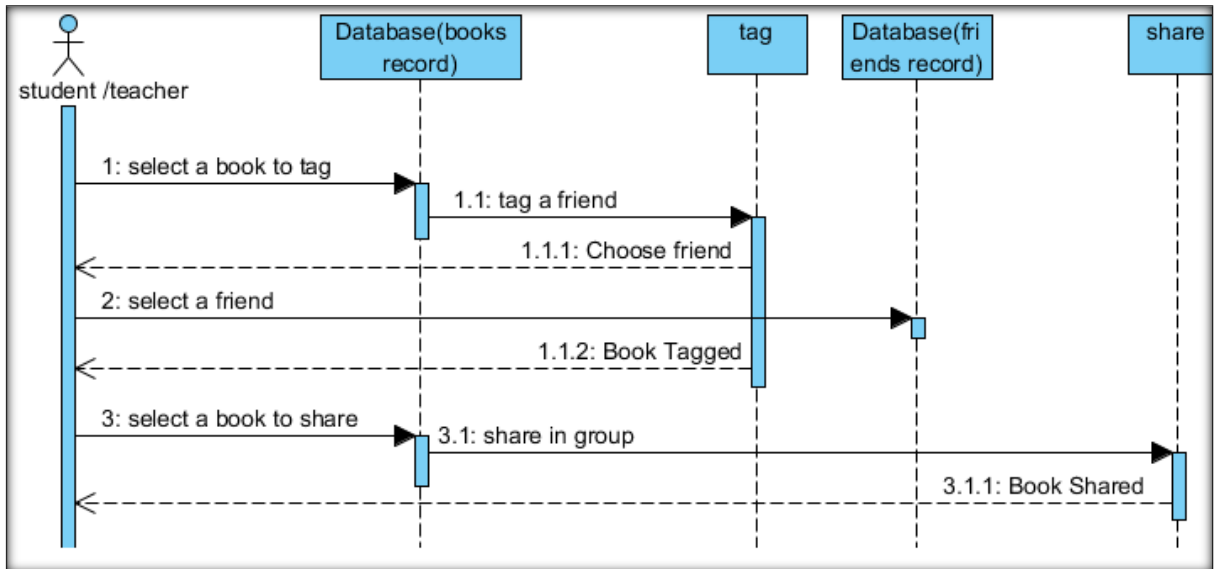


Figure 56 : Tag and share-sequence diagram

Course group

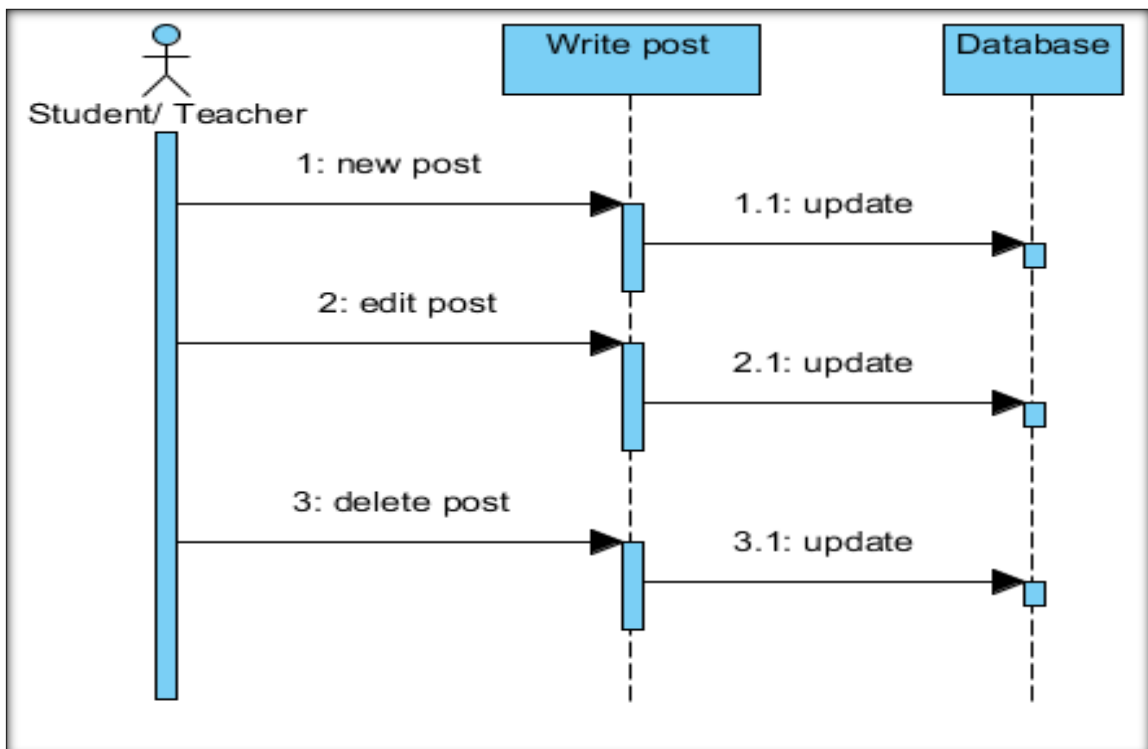


Figure 57 : course group-Sequence diagram

Mini library

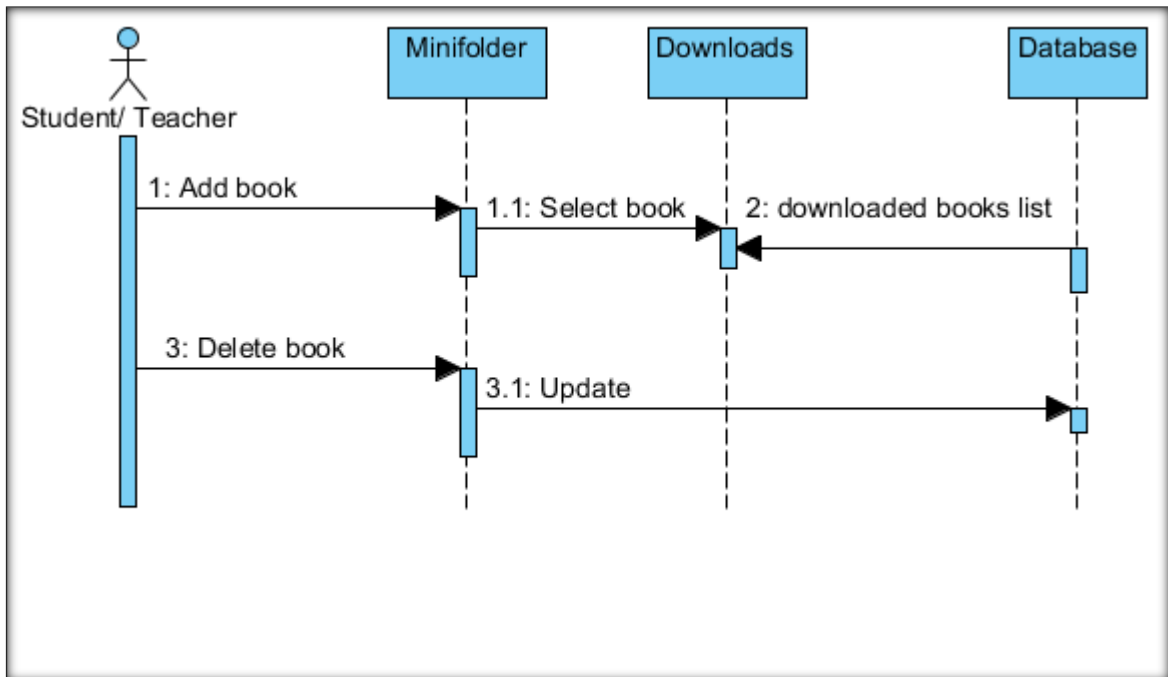


Figure 58 : Mini library-Sequence diagrams

Notification manager

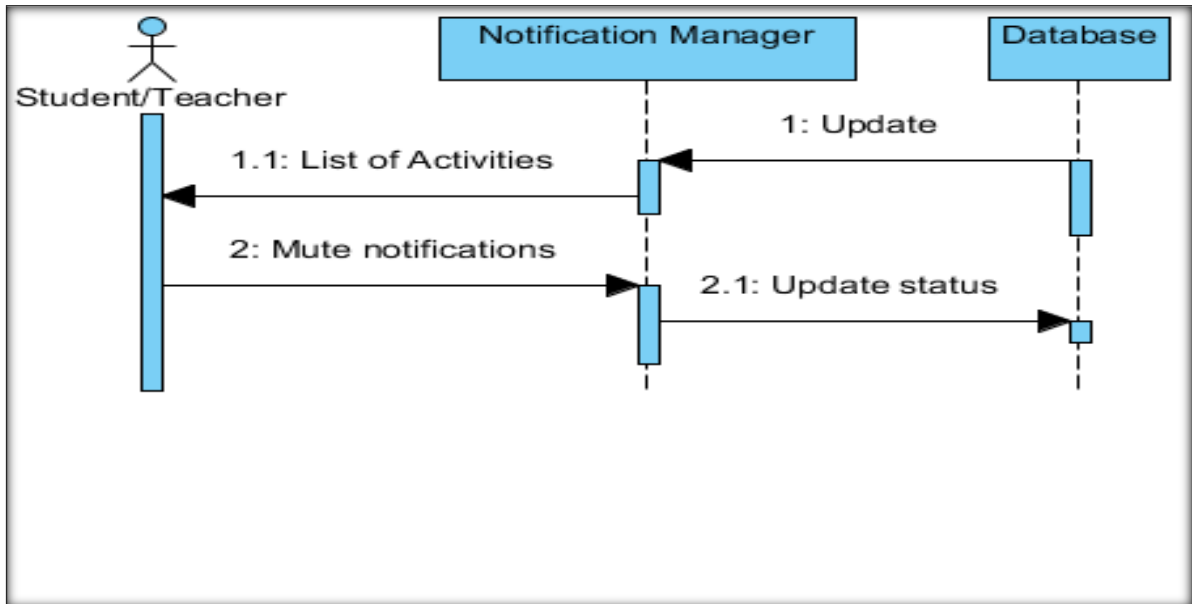


Figure 59 : Notification manager-Sequence diagram

Account

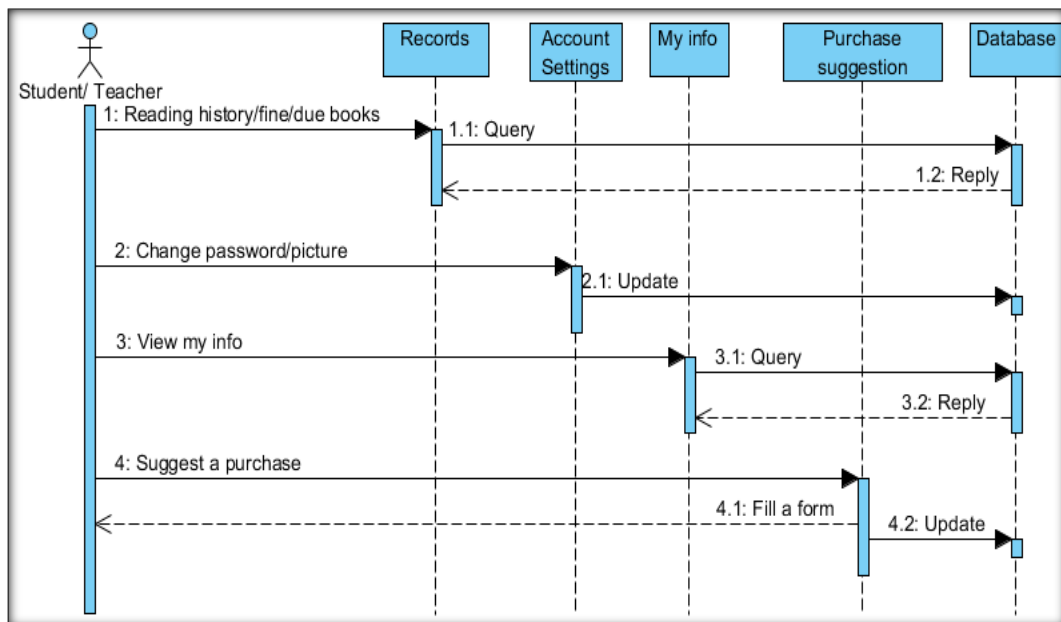


Figure 60 : Account-Sequence diagram

Customize app

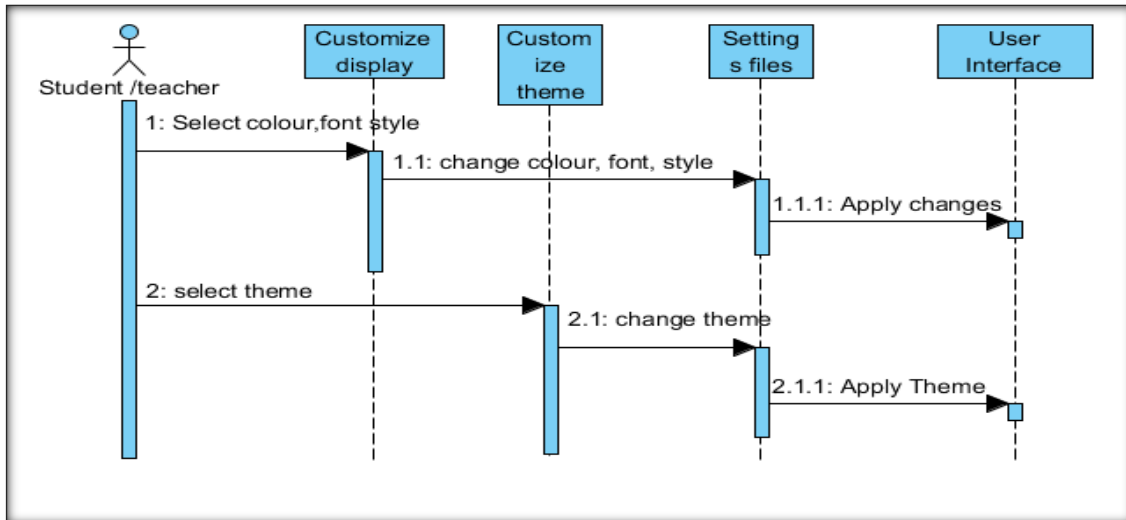


Figure 61 : Customization-Sequence diagram

Help

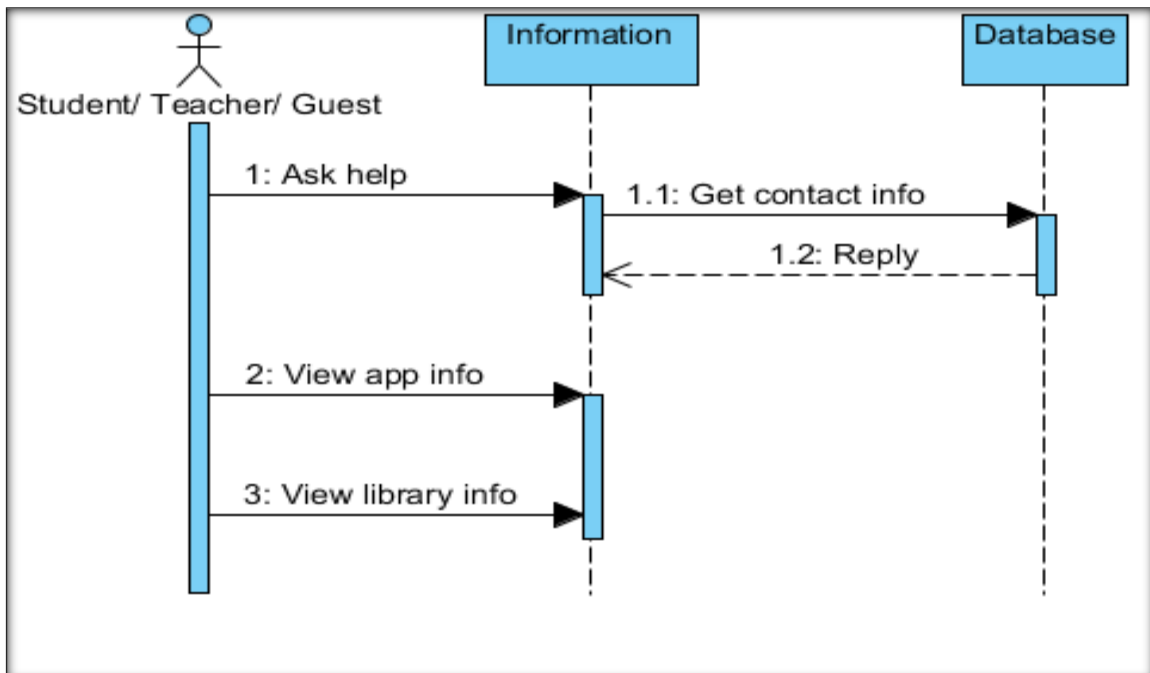


Figure 62 : Help-sequence diagram

4.6 Class diagram

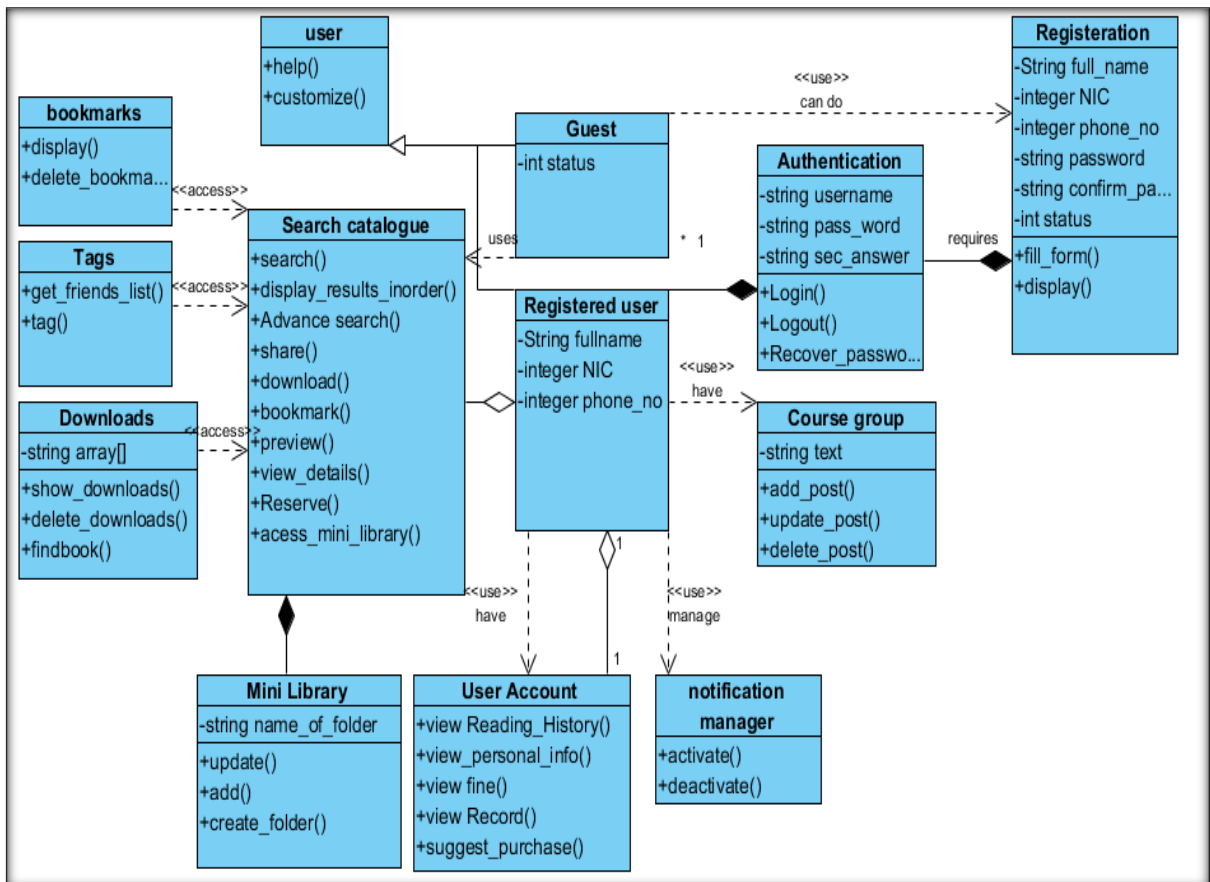


Figure 63 : Class diagram

Chapter 5

Implementation

5.1 Introduction

This chapter describes about different phases of implementation in detail. Mobile Library app has been implemented in the following phases:

User Interface

Backend Functionalities

Database Configuration

Integration of Database with Android App

5.2 User Interface

First the appearance of app at user side is set according to the functions and services that app offers for the user. The screens are linked in the order of their appearance. Icons and buttons are categorically placed presenting individual function of the app. **Refer to figures under section user interface 3.5.**

5.3 Backend Functionalities:

The functionalities and their logic are implemented in android development environment in code that is not visible to the user. The flow of logic makes a function work properly through a number of inputs, queries, checks , interacts with database and outputs the desired results.

5.4 Database Configuration:

The data that interacts with the app is placed in database tables of MySql from where it is fetched as needed. It is configured on an online server at same time as functionalities are

being coded. The various tables are being updated, as the user requires them to be changed. Some tables can only be changed by administration side.

5.5 Integration of Database with Android App:

Server side holds both database and php script files. The php script files process the queries from database and delivers the results to the user through json objects.

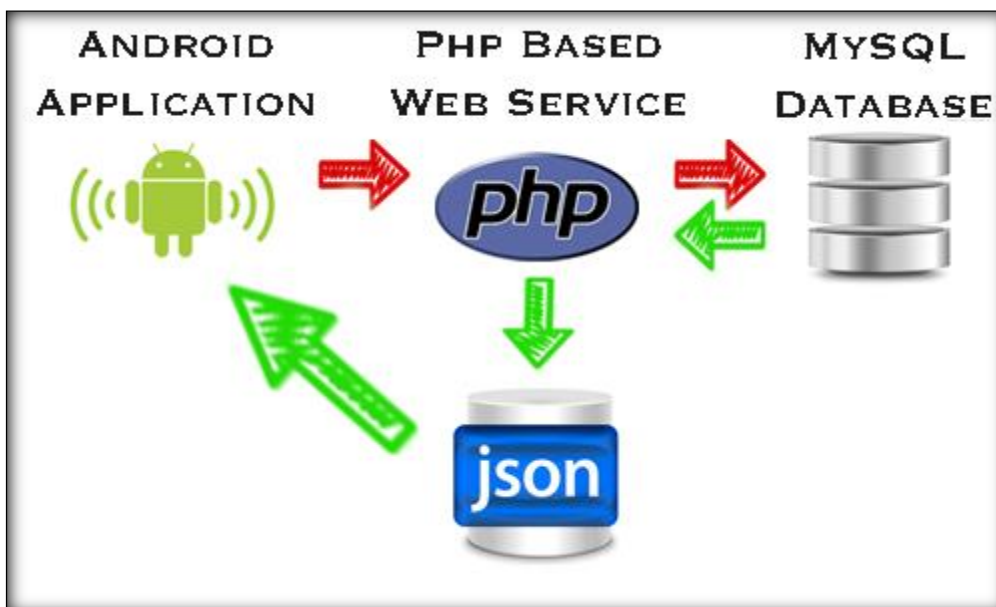


Figure 64: implementation overview

Chapter 6

Testing and Analysis

6.1 Introduction

Software testing and analysis is one of the most crucial part of software development life cycle. This can be termed as element of a broader topic that is referred to as verification and validation. Software testing can also be stated as the process of validating and verifying that a software program/product:

Meets the business and technical requirements that guided its design and development ;

Works as expected ; and

Can be implemented with the same characteristics.

Software testing , depending on the testing method employed ,can be implemented at any time in the development process .however mostly the test efforts occurs after the requirement have been defined and the coding process has been completed. As such the methodology of the test is governed by the software development methodology adopted.

6.2 Verification and validation

Verification and validation is intended to be systematic and technical evaluation of the system and its processes. To effectively deal with the increased complexity and functionality , systems needs practical techniques that can help improve software quality using the validation and verification process.

6.3 Functional testing

Software development organization with an effective functional testing practice has a objective way to determine whether each functional requirement is actually implemented in the code. With functional testing, the team translates functional requirement into executable test cases that confirm how well the code satisfies the requirement at any given time.it provides unprecedented objectives insight into requirement status and

prevents the missing or incorrect functionality implementations that can lead to countless rewrites ,user dissatisfaction and project failure.

Functional testing involves 5 steps:

- a) The identification of functions that the software is expected to perform.
- b) The creation of input data based on the function's specification.
- c) The determination of output based on the function's specification.
- d) The execution of test case.
- e) The comparison of actual and expected outputs.

6.4 Functional testing plans

Testing goals

To validate application functionality ,many of the features will be tested to ensure that those function provides the expected output. Testing will be completed over consecutive two days. Testing will complete when all the features have passed all the associated test cases-no exceptions.

Scope

This section details the feature that will be included in the functional testing phase(s) and those that will be excluded.

Testing approach and tools

The project is using an XP approach. At the end, the application will be moved to testing environment for functional testing. The testing cycle will be two days long .Result will be recorded using appropriate template and faults will be communicated to the development team at the end of the testing cycle using the given form.

Functional test cases and their execution

The following table shows the features that are subject into testing phase .test cases will be create and executed against these features.

Table 9:Feature ID

Feature ID	Feature name
1	Registration page
2	Login page
3	Search catalogue
4	Reserve
5	Renew
6	Bookmark
7	Tag
8	Course group
9	Account
10	Searched results
11	Help

Feature ID-1:Registration**Table 10:Registration**

S#	Test data	Expected O/P	O/P	Status
1	Iqra 12345678 12345678 6110198090877 iqrafaqraz@gmail.com	Message “You are successfully registered ”	Message “You are successfully registered ”	Success
2	Iqra 1234567 1234567 6110198090877 iqrafaqraz@gmail.com	Message “password must be at least 8 characters long ”	Message “password must be at least 8 characters long ”	Success

3	Iqra 1234567 1234567 6110198090877	Message “please check your fields ”	Message “please check your fields ”	Success
4	Iqra 12345678 12345679 6110198090877 iqrafaqraz@gmail.com	Message “password do not match”	Message “password do not match”	Success
5	Iqra 12345678 12345678 611019809087 iqrafaqraz@gmail.com	Message “invalid NIC”	Message “invalid NIC”	Success
6	Iqra 12345678 12345678 6110198090873 Iqrafaqraz	Message “invalid NIC”	Message “invalid NIC”	Success

Feature ID-2: Login

Table 11:Login

S#	Test data	Expected O/P	O/P	Status
1	Iqra 12345678	Home page “successfully login”	Home page “successfully login”	Success

2	12345678	“ Sorry, fields are empty”	Sorry, fields are empty”	Success
3	Iqra 12345688	“Login not Successful. Invalid Credentials!”	“Login not Successful. Invalid Credentials!”	Success
4	Forgot password	Answer security question	Answer security question	Success
5	“Mano”	Password change page	Password change page	Success
6	“kitty”	“Wrong answer. contact system administrator for password recovery”	“Wrong answer. contact system administrator for password recovery”	Success
7	“bini”	“wrong answer. Your account has been locked for a short time ”	“wrong answer. Your account has been locked for a short time ”	Success
8	Login as Guest	Main menu page	Main menu page	Success
9	Bookmark	“Dear user, kindly login to continue ”	“Dear user, kindly login to continue ”	Success

Feature ID-3:Search Catalogue

Table 12:Catalogue

S#	Test data	Expected O/P	O/P	Status
1	“A tail of ”	A tail of two cities	A tail of two cities	Success
2	“”Cryptography ”	“Message “Book not found ”	“Message “Book not found ”	Success
3	Naseem Hijazzi	“Khuda or Muhabbat” “Duniya gol ha”	“Khuda or Muhabbat” “Duniya gol ha”	Success
4	William stalling	“Book not found”	“Book not found”	Success
5	123456789	“Khuda or Muhabbat”	“Khuda or Muhabbat”	Success
6	123456780	“Book with specified ISBN not found”	“Book with specified ISBN not found”	Success

Feature ID-4: Reserve books

Table 13:Reserve

S#	Test data	Expected O/P	O/P	Status
1	Reserve book “A tale of two cities”	“No hard copies available ”	“No hard copies available ”	Success
2	Reserve book “Khuda or	“Your book has been reserved for	“Your book has been reserved for	Success

	muhabbat”	2 weeks “	2 weeks”	
3	Reserve book “Khuda or muhabbat”	“Book is already reserved .Due date is 1/1/2015”	Book is already reserved .Due date is 1/1/2015”	Success

Feature ID-5:Renew

Table 14:Renew

S#	Test data	Expected O/P	O/P	Status
1	Renew book	“book has been renewed .due date is 5/1/2015”	“book has been renewed .due date is 5/1/2015”	Success
2	Renew book	“max limit reached . see help for more info”	max limit reached . see help for more info”	Success

Feature ID-6:Bookmark

Table 15:Bookmark

S#	Test data	Expected O/P	O/P	Status
1	Bookmark “A tale of two cities ”	“Successfully Bookmarked ”	“Successfully Bookmarked ”	Success
2	Bookmark list	A tale of two cities	A tale of two cities	Success

3	Delete bookmarked item	“Bookmark deleted ”	“Bookmark deleted ”	Success
---	------------------------	---------------------	---------------------	---------

Feature ID-7:tag

Table 16:Tag

S#	Test data	Expected O/P	O/P	Status
1	tag	“Enter friend name” dialogue box appears	“Enter friend name” dialogue box appears	Success
2	Aroona ashkenaz OK	“successfully tagged ”	“successfully tagged ”	Success
3	Waleed bin haider OK	“Person not found”	“person not found”	Success

Feature ID-8:Course group

Table 17:Course group

S#	Test data	Expected O/P	O/P	Status
1	Course group	Course group page appears	Course group page appears	Success
2	Write post	Text editor appears for input	Text editor appears for	Success

			input	
3	“AoA ,HRM quiz tomorrow ” OK	“your post has been created”	“your post has been created”	Success
4	Update post	A dialogue box appears with two options Edit or Delete	A dialogue box appears with two options Edit or Delete	Success
5	Edit	Text editor appears	Text editor appears	Success
6	Made changes	“Updated ”	“Updated ”	Success
7	Delete	“Deleted ”	“Deleted ”	Success

Featre ID-9:Account

Table 18:Account

S#	Test data	Expected O/P	O/P	Status
1	Change password	Change password page	Change password page	Success
2	Input : 12345678 12345678	“successfully changed ”	“successfully changed ”	Success
3	12345678 123456789	“Mismatched passwords”	“Mismatched passwords”	Success
4	1234 1234	“password must be at least 8 characters long”	“password must be at least 8 characters long”	Success
5	Iqra faqraz	“Mismatched		Success

	Iqra faqra	passwords ”		
6	Logout	Login page	Login page	Success

Feature ID-10:Searched results

Table 19: functionality on book

S#	Test data	Expected O/P	O/P	Status
1	Tap on book “A tale of two cities ”	Dialogue box appears with different options on it	Dialogue box appears with different options on it	Success
2	Book info	Displays Book name :A tale of to cities Author name:johne Clarke ISBN number :xxxxxxx No.of Hard copied available:5	Displays Book name :A tale of to cities Author name:johne Clarke ISBN number :xxxxxxx No.of Hard copied available:5	Success
3	Reserve book	“Are you sure ,you want to reserve this book?”	“Are you sure ,you want to reserve this book?”	Success
4	Yes	“book successfully	“book successfully	Success

		reserved .Due date is 1/1/2015”	reserved .Due date is 1/1/2015”	
5	No	Dialogue box disappears	Dialogue box disappears	Success
6	Reserve book	“No hardcopy available right now ”	“No hardcopy available right now ”	Success
7	Share book information to group	“Successfully shared book info”	“Successfully shared book info”	Success

Feature ID-11:Help

Table 20:Help

S#	Test data	Expected O/P	O/P	Status
1	Help	Help page appears	Help page appears	Success
2	Tap on make call	Dialing...	Dialing...	Success
3	Back	Main menu page	Main menu page	Success

6.5 Unit testing

In computer programming, unit testing is a procedure to validate that individual units of source code are working properly. A unit is the smallest testable part of an application. Unit testing concentrates on each unit of the software as implemented in source code. The goal of unit testing is to isolate each part of the program and show that individual parts are correct.in this application each module as developed is individually tested. The each

function performed on the module is tested. Each operation and function is individually tested so as to check for possible errors that could occur.

6.6 Integration testing

Integration testing is the phase of software testing in which individual software modules are combined and tested as a group. It follows unit testing and precedes system testing in integration testing focus is on design and the construction of software architecture .all the modules have been combined and tested to ensure that they work according to the user requirements.

Modules to be integrated

Integration of android application with database was important. Initially both were tested individually. Database was running properly. To make the communication possible between android and database, PHP scripts were written on the server side . These scripts were used as the medium of their communication.

To check whether, they are integrated properly, we made some tests.

User entered his information for registering himself. The data entered by him was successfully stored by the database and hence the user was able to log in to the application by providing his credentials. That means the database is working properly and gets updated when required.

Similarly all the features stated in UNIT testing were tested to check whether everything was working properly in parallel and the result was a success.

Entry Criteria

Code is reviewed and approved.

Unit test plan for each modules is written ,reviewed and executed.

All the unit tests passed.

Exit Criteria

All code is completed and frozen and no more modules are to be integrated .

All of the system integration tests passed.

No major defect is outstanding

All the moderate defects found in SIT phase are fixed and retested.

System integration method

Big Bang techniques have been use to integrate the system.

In the big bang approach ,all the modules are individually tested. Next all those modules are put together to construct the entire system which is tested as a whole.

the software meets all the functional ,behavioral and performance requirements.

6.7 System testing

In software testing phase overall system is tested as a whole. The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together. System testing is a more limiting type of a testing. it seeks to detect defect within the system as a whole.

6.8 Summary

The system is tested thoroughly .All the modules have been tested for validation and results were found 95% accurate . Some tests cases with failure status were reviewed again and the missing functionality was added .

Chapter 7

Future Work

7.1 Introduction

There is scope for a number of advanced features that can still be added in Mobile Library in order to make it more interactive, and efficient for use.

Adding more mediums to the library

The user shall be able to look for material of his interest in a number of available mediums like soft copy, hard copy, C-Ds, videos, etc that he can issue according to their availability.

Chatting

A module for chatting with members in course group can help users to keep in touch with their fellows. They shall be able to send and receive messages for purpose of group study .

Uploading Audio/Video lectures and files in Course Group:

The user shall be able to post lectures (PDF, word doc),videos , audios in course group for the sake of sharing stuff with other group members.

Online Quizzes and Assignments:

The user shall be able to submit his quizzes and assignments. The course instructor can then access them through his account having slightly advanced privileges than students' accounts.

Customization using animations:

The user shall be able to customize the app beyond applying themes like using animations, for transition between screens . this way , the interface will be more interactive and user friendly

Appendix A

Assumption: A statement that is believed to be true in the absence of proof or definitive knowledge.

Availability: Present and ready for use; at hand; accessible.

Mb: megabytes

Class: A description of a set of objects having common properties and behaviors, which typically correspond to real-world items (persons, places, or things) in the business or problem domain.

Compatibility: Capable of orderly, efficient integration and operation with other elements in a system with no modification or conversion required.

Constraint: A restriction that is imposed on the choices available to the developer for the design and construction of a product.

API (application programming interface): specifies how some software components should interact with each other. In addition to accessing databases or computer hardware, such as hard disk drives or video cards, an API can be used to ease the work of programming graphical user interface components.

Wi-Fi: Wi-Fi is the name of a popular wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections

OS: Operating System

Dependency: A reliance that a project has on an external factor, event, or group outside its control.

External interface requirement: A description of an interface between a software system and a user, another software system, or a hardware device.

Feature: A set of logically related functional requirements that provides a capability to the user and enables the satisfaction of a business objective.

Flowchart: An analysis model that shows the processing steps and decision points in the logic of a process or of a program.

Functional requirement: A statement of a piece of required functionality or a behavior that a system will exhibit under specific conditions.

Hardware: A computer and the associated physical equipment directly involved in the performance of data-processing or communications functions.

Hardware Interface: The logical and physical characteristics of each interface between the software product and the hardware components of the system.

Implementation: Execution of a plan, idea, model, design, specification, standard, algorithm, or policy.

Interface: A point where two systems, subjects, organizations, etc., meet and interact.

Nonfunctional requirement: A description of a property or characteristic that a software system must exhibit or a constraint that it must respect, other than an observable system behavior.

Operating Environment: The circumstances surrounding and potentially affecting something that is operating.

Operating System: A collection of software that manages computer hardware resources and provides common services for computer programs.

Perspective: The way in which objects appear to the eye.

Procedure: A written description of a course of action to be taken to perform a given activity, describing how the activity is to be accomplished.

Process: A sequence of activities performed for a given purpose. A process description is a documented definition of those activities.

Quality attribute: A kind of nonfunctional requirement that describes a quality or property of a system. Examples include usability, portability, maintainability, integrity, efficiency, reliability, and robustness. Quality attribute requirements describe the extent to which a software product demonstrates desired characteristics, not what the product does.

References: List of any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.

Response: A reaction, as that of an organism or a mechanism, to a specific stimulus.

Requirement: A statement of a customer need or objective, or of a condition or capability that a product must possess to satisfy such a need or objective. A property that a product must have to provide value to a stakeholder.

Requirements specification: See software requirement specification and specification, requirement.

Scope: The portion of the ultimate product vision that the current project will address. The scope draws the boundary between what's in and what's out for the project.

Software requirements specification: A collection of the functional and nonfunctional requirements for a software product.

Specification, requirements: The process of documenting a system's requirements in a structured, shareable, and manageable form. Also, the product from this process.

Stimulus: Something causing or regarded as causing a response.

Supplementary Information: Something added to complete the information.

System requirement: A top-level requirement for a product that contains multiple subsystems, which could be all-software or software and hardware.

Usability: Fit for use; convenient to use.

Use case: A description of an interaction between an actor and a system that results in an outcome that provides value to the actor.

Use case diagram: An analysis model that identifies the actors who can interact with a system to accomplish valuable goals and the various use cases that each actor will perform.

User: A customer who will interact with a system either directly or indirectly (for example, using outputs from the system but not generating those outputs personally). Also called end user.

User class: A group of users for a system who have similar characteristics and requirements for the system.

User Interface: the logical characteristics of each interface between the software product and the users.

User requirement: User goals or tasks that users must be able to perform with a system, or statements of the user's expectations of system quality.

Validation: The process of evaluating a work product to determine whether it satisfies customer requirements.

Verification: The process of evaluating a work product to determine whether it satisfies the specifications and conditions imposed on it at the beginning of the development phase during which it was created.

Modifiability: the effort required to make changes in the software

Portability: The effort required to move the software to a different target platform

Reliability: dependable, how often the software fails.

Usability: how difficult it will be to learn and operate the system

Flexibility: ability to adapt to possible or future changes in its requirements.

Availability: something is available for use when it is needed

Appendix B

Activity diagram: An analysis model that shows a dynamic view of a system by depicting the flow from one activity to another. Similar to a flowchart.

Assumption: A statement that is believed to be true in the absence of proof or definitive knowledge.

Class: A description of a set of objects having common properties and behaviours, which typically correspond to real-world items (persons, places, or things) in the business or problem domain.

Class diagram: An analysis model that shows a set of system or problem domain classes and their relationships.

Compatibility: Capable of orderly, efficient integration and operation with other elements in a system with no modification or conversion required.

Constraint: A restriction that is imposed on the choices available to the developer for the design and construction of a product.

API (application programming interface): specifies how some software components should interact with each other. In addition to accessing databases or computer hardware, such as hard disk drives or video cards, an API can be used to ease the work of programming graphical user interface components.

Wi-Fi: Wi-Fi is the name of a popular wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections

Wi-Fi Router: used to provide access to the Internet.

OS: Operating System

Dependency: A reliance that a project has on an external factor, event, or group outside its control.

External interface requirement: A description of an interface between a software system and a user, another software system, or a hardware device.

Feature: A set of logically related functional requirements that provides a capability to the user and enables the satisfaction of a business objective.

Flowchart: An analysis model that shows the processing steps and decision points in the logic of a process or of a program.

Interface: A point where two systems, subjects, organizations, etc., meet and interact.

Non-functional requirement: A description of a property or characteristic that a software system must exhibit or a constraint that it must respect, other than an observable system behaviour.

Operating Environment: The circumstances surrounding and potentially affecting something that is operating.

Operating System: A collection of software that manages computer hardware resources and provides common services for computer programs.

Perspective: The way in which objects appear to the eye.

Post condition: A condition that describes the state of a system after a use case is successfully completed.

Precondition: A Condition that must be satisfied before a use case may begin.

Procedure: A written description of a course of action to be taken to perform a given activity, describing how the activity is to be accomplished.

Process: A sequence of activities performed for a given purpose. A process description is a documented definition of those activities.

References:List of any other documents or Web addresses to which this SDS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.

Response:A reaction, as that of an organism or a mechanism, to a specific stimulus.

Scope: The portion of the ultimate product vision that the current project will address. The scope draws the boundary between what's in and what's out for the project.

Software requirements specification: A collection of the functional and non-functional requirements for a software product.

Specification, requirements: The process of documenting a system's requirements in a structured, shareable, and manageable form. Also, the product from this process.

Stimulus:Something causing or regarded as causing a response.

System requirement: A top-level requirement for a product that contains multiple subsystems, which could be all-software or software and hardware.

Usability:Fit for use; convenient to use.

Use case: A description of an interaction between an actor and a system that results in an outcome that provides value to the actor.

Use case diagram: An analysis model that identifies the actors who can interact with a system to accomplish valuable goals and the various use cases that each actor will perform.

User: A customer who will interact with a system either directly or indirectly (for example, using outputs from the system but not generating those outputs personally). Also called end user.

User class: A group of users for a system who have similar characteristics and requirements for the system.

User Interface: the logical characteristics of each interface between the software product and the users.

User requirement: User goals or tasks that users must be able to perform with a system, or statements of the user's expectations of system quality.

Verification: The process of evaluating a work product to determine whether it satisfies the specifications and conditions imposed on it at the beginning of the development phase during which it was created.

PC: personal computer

Glossary

API: Application Programming Interface

APP: Application

ML: Mobile Library

OS: Operating System

LMS: Library management system

SDLC: Software Development Life Cycle

UML Unified Modeling Language

PC: Personal Computer