

CSY2030 Individual Project to create a GUI based Research Article Management System

| Date for Submission: | July 12, 2019 |
|---|---------------|
| Name | UN Id |
| Tsering Khando Lama | 18406499 |
| video: https://northampton.mediaspace.kaltura.com/media/Tseringkhando-18406499-+demo/0_ybde8geu | |
| Username: admin Password: admin | |

Table of Contents

| | |
|---|-------------------------------------|
| 1. Introduction..... | 3 |
| 2. System Design | 4 |
| 2.1 Use – Case Diagram..... | 4 |
| 2.1.1 Human Resource Administrator | 4 |
| 2.1.2 Researchers | Error! Bookmark not defined. |
| 2.2 Class Diagram | 7 |
| 3. System Description | 8 |
| 3.1 User Guide | 8 |
| 3.1.1 Admin Sign In..... | 8 |
| 3.1.2 Add Researcher..... | 10 |
| 3.1.3 View, Edit, Delete and Query Researcher | 10 |
| 3.1.4 Add an Article..... | 12 |
| 3.1.5 View, Edit, Delete, Query and Hire Articles | 12 |
| 3.1.6 Hired Articles..... | 14 |
| 3.1.7 Researcher Sign In | 14 |
| 4. Evidence of Testing | 16 |
| 4.1 Test Logs(Black Box Testing) | 16 |
| 4.2 White Box Testing | 17 |
| 4.2.1 View Conference Article | 17 |
| 4.2.2 View Hired Articles | 18 |
| 4.3 List of bugs and weaknesses: | 18 |
| 5. Conclusion | 19 |
| References..... | 20 |
| Appendix..... | 21 |

1. Introduction

The task given was to develop a stand-alone Research Article Management System which could be accessed using graphical user interface (GUI) for the Northampton Research Centre's Library. The system was to be built using object-oriented principles in Java. The report describes the prepared new system along with design and tests implementation. Design section includes Unified Modelling Language (UML) use-case and class diagrams and the tests include black and white- box testing.

The application was developed as per the requirements provided to the author and achieved features of the new system are as follow:

For the administration or the human resource-

- 1 Sign in onto the system with the username and password, and sign out.
- 2 Add new researcher, book chapters, journal articles and conference articles.
- 3 Change details of all the records.
- 4 Delete all the records.
- 5 Hire out all those articles to the researchers.
- 6 Return the hired articles from the researchers.
- 7 View the lists of articles hired and the researchers who hired them.
- 8 Search the researcher with their ID number and view the details including the articles they have borrowed.
- 9 Search all the articles based on their titles and view their details.

For the Researchers-

- 1 Sign in with the username and password given by the administration.
- 2 View the articles they have borrowed only.
- 3 View all the available articles, i.e., which are not hired by the researchers.
- 4 Sign out of the system.

The functionalities were achieved with the use of following concepts of Java:

- File handling (binary objects)
- Exception handling
- Model View Controller
- Collections

2. System Design

Before coding, to decide the user interface, the visualization of the system is important. This refers to the modelling of the system diagrammatically which was done using the UML diagrams, Use-Case and Class diagrams.

2.1 Use – Case Diagram

This shows what the human resource staff and the researchers can do with the system.

2.1.1 Human Resource Administrator

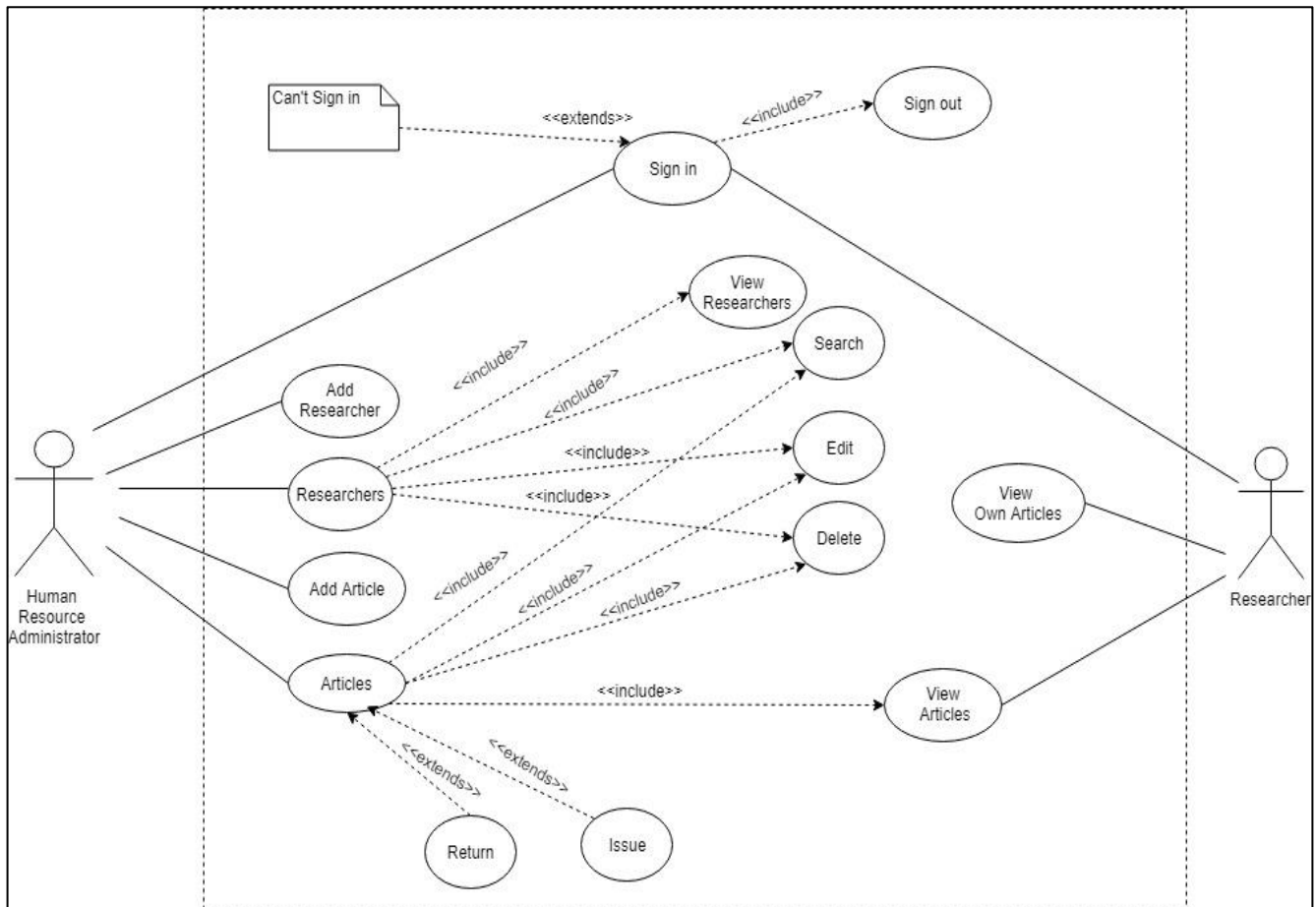


Figure 2.1: Use- Case diagram

Description

1 UC1 SIGN IN

Name: UC1 Sign IN

Initiator: Human Resource Administrator, Researcher

Goal: To sign in into the system.

Pre-Condition: Correct username and password.

Post-condition: Successfully signed in.

Assumptions: Admin and researcher both are valid.

Main Success Scenario

- The users are signed in.
- Windows are opened according to admin or researcher.

Extends: Can't Sign in

Includes: Sign Out

2 UC2 SIGN OUT

Name: UC2 Sign Out

Initiator: Human Resource Administrator

Goal: To be signed out of the system.

Pre-Condition: Administrator or Researcher must be logged in first.

Post-condition: Admin or Researcher is signed out.

Assumptions: Admin or Researcher is currently logged in into the system.

Main Success Scenario

- Admin or Researcher has signed out and the login view opens.

3 UC3 ADD RESEARCHER

Name: UC3 Researchers

Initiator: Human Resource Administrator

Goal: To add new researcher.

Pre-Condition: Admin must be signed in.

Post-condition: Admin adds a new researcher

Assumptions: Admin is signed in.

Main Success Scenario

- New researcher is added.

4 UC4 RESEARCHERS

Name: UC4 Researchers

Initiator: Human Resource Administrator

Goal: To edit, delete, view and query details of each.

Pre-Condition: Admin must be signed in.

Post-condition: Admin is able to perform all the functions.

Assumptions: Admin is signed in.

Main Success Scenario

- Admin performs all the work with no errors.

Includes: Edit, Delete, View Researchers, Search

5 UC5 ADD ARTICLE

Name: UC5 Add Article

Initiator: Human Resource Administrator

Goal: To add new article.

Pre-Condition: Admin must be signed in.

Post-condition: New article is added.

Assumptions: Admin is signed in.

Main Success Scenario

- Add has added new articles.

6 UC6 ARTICLES

Name: UC6 Articles

Initiator: Human Resource Administrator, Researcher

Goal: To edit and delete articles, list them, query each and hire each to researcher and return to the library.

Pre-Condition: Admin or researcher must be signed in. Researcher can only view articles.

Post-condition: Admin does all the work. Only the articles not on hire are listed to researcher.

Assumptions: Admin or researcher is logged in.

Main Success Scenario

- Admin does all the operations successfully.
- Researcher is shown only the articles available for hire.

Includes: View, Edit, Delete, Hire, Return

Extends: Return, Issue

7 UC7 VIEW OWN ARTICLES

Name: UC7 View Own Articles

Initiator: Researcher

Goal: To view the articles hired by the researchers themselves only.

Pre-Condition: Researcher must be logged in and must have borrowed article.

Post-condition: Articles borrowed are listed.

Assumptions: Researcher is registered.

Main Success Scenario

- Researcher can view only those articles borrowed by them.

2.2 Class Diagram

This shows the back end of the system which shows the relationships between the objects used.

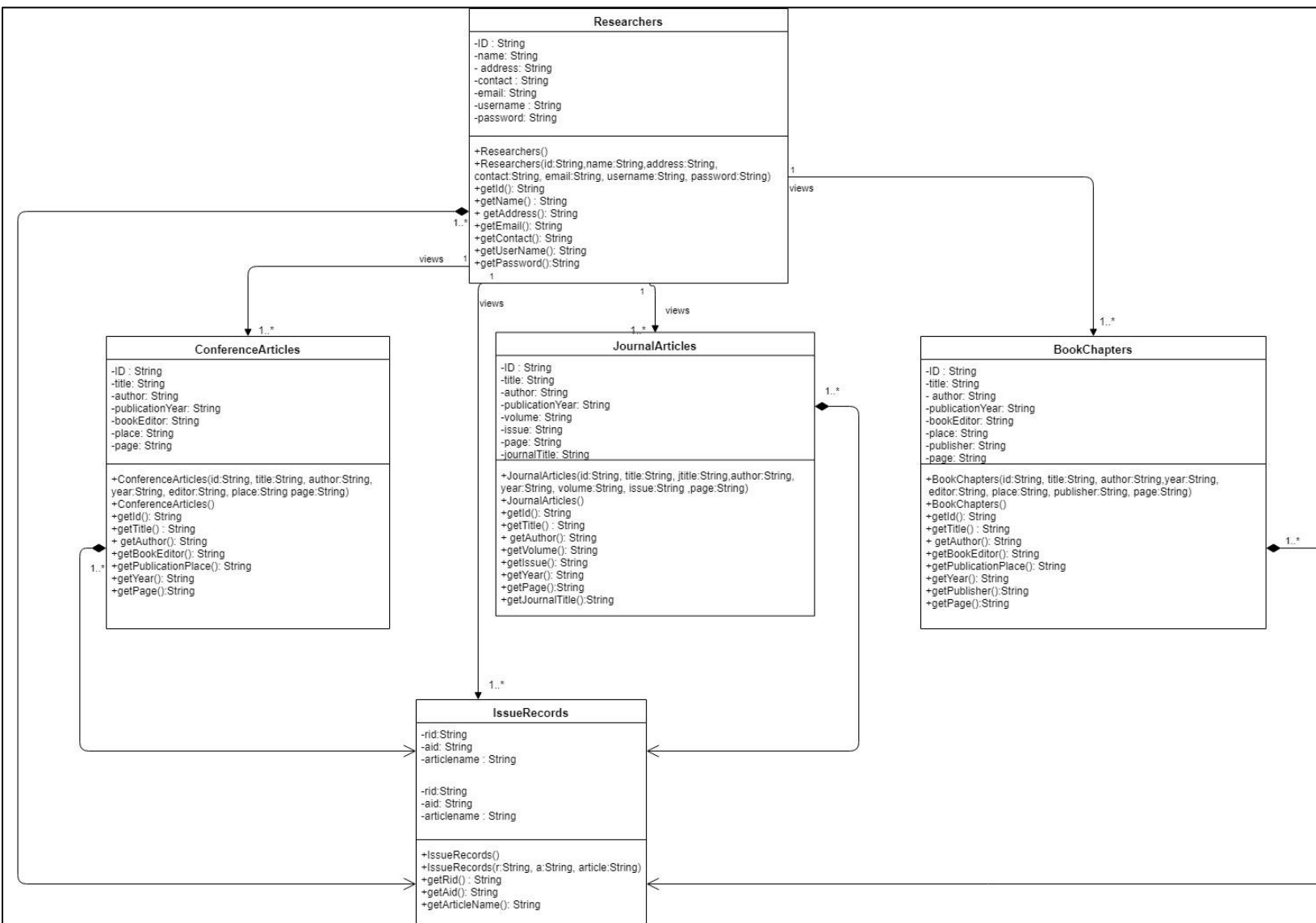


Figure 2.2: Class diagram

3. System Description

The very first interface of the system would be the ‘Sign In’ view which is shown in the fig3.1 and further, guidance to use the system has been provided.

Both, the admin and researcher will sign in from the same area.

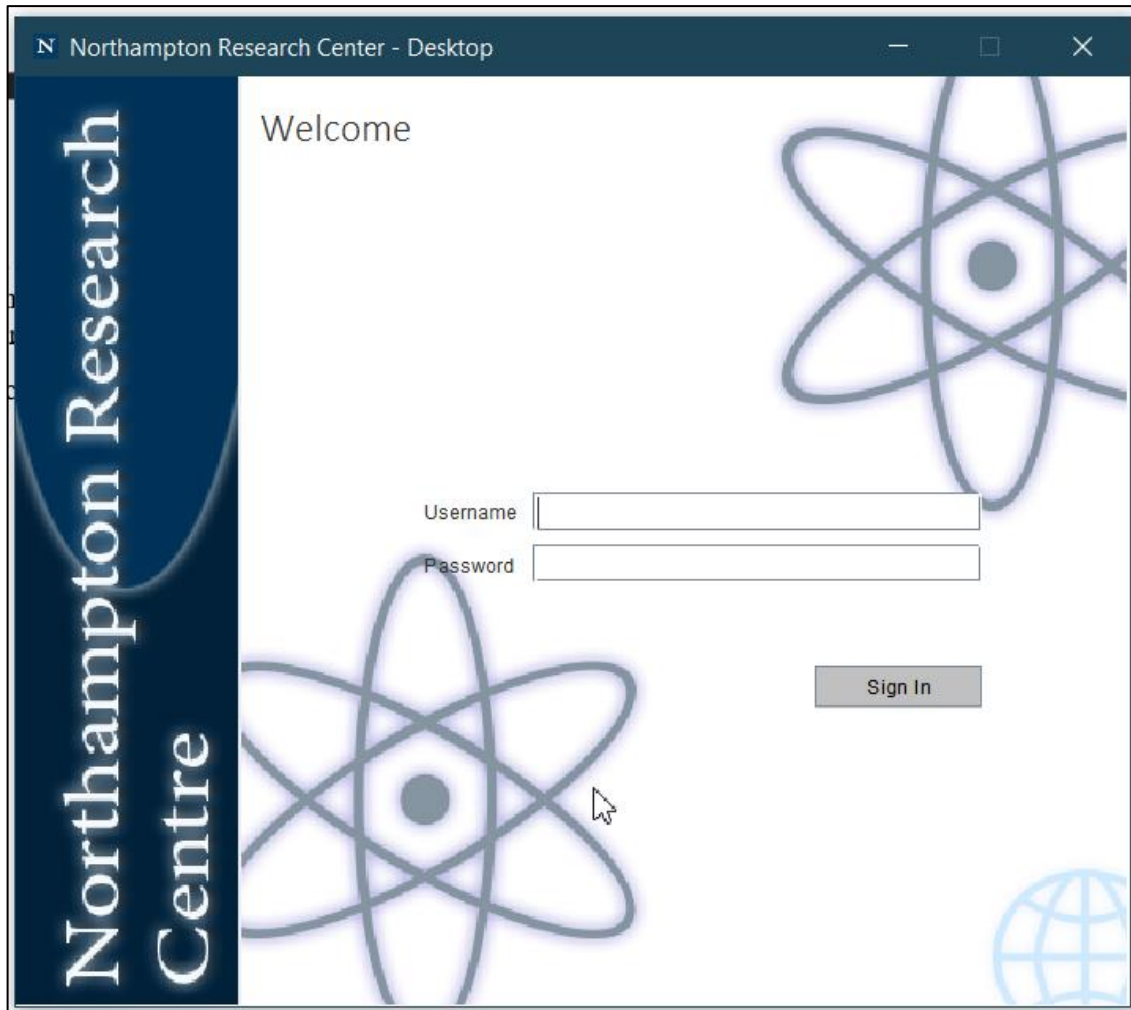


Figure 3.1: Sign in view

3.1 User Guide

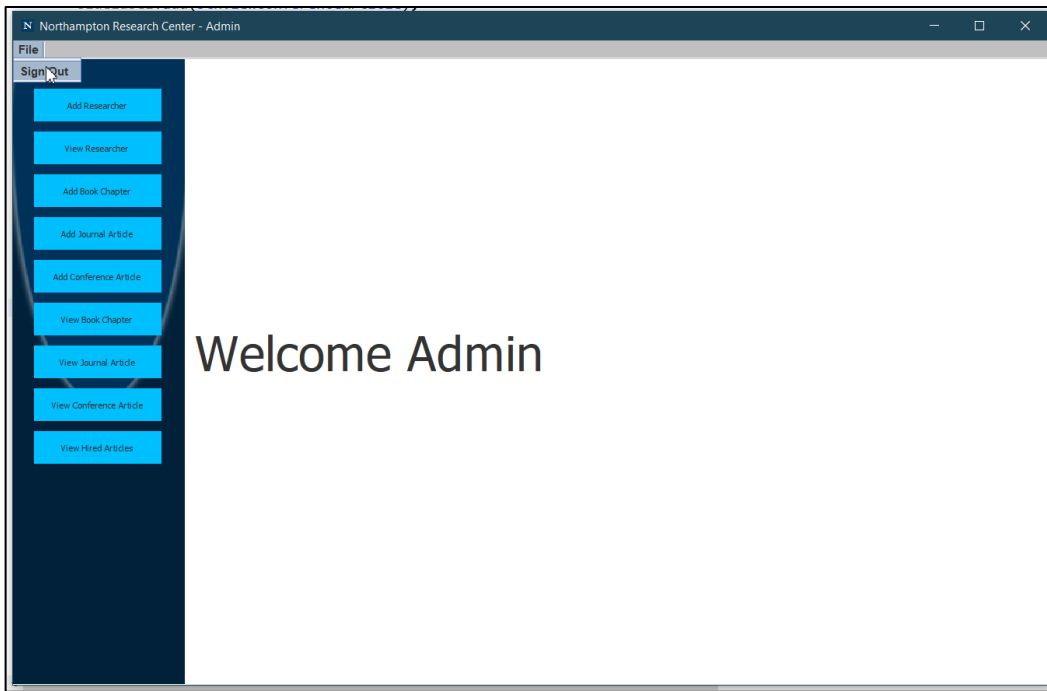
The admin and researcher are identified by their username from the login page.

3.1.1 Admin Sign In

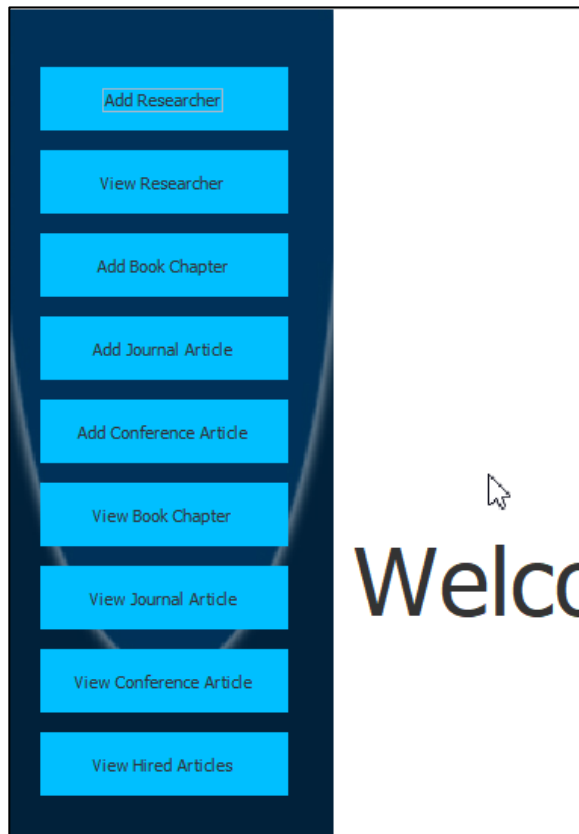
The human resource administrator's current username is "admin" and password is "admin". So, the admin needs to enter them in the text fields and click the "Sign In" button as shown:



Once you are logged in, the admin view opens:



- The navigation menu is on the left panel where the admin can click and perform operations.
- The user can sign out by clicking on the “File” menu on the top-left corner and clicking “Sign Out”.



These are the list of buttons that the admin will have access to, and the admin needs to click to navigate to different sections of the system.

3.1.2 Add Researcher

The admin first should click on the “Add Researcher” button and the form opens up.

Add New Researcher

ID No.

Name

Address

Contact

Email

[\(Create username and password\)](#)

Username

Password

- In the form, the ID no. is auto generated, and the rest of the information need to be filled by the admin.
- All the information must be filled correctly and the “Add Researcher” button is to be clicked.

3.1.3 View, Edit, Delete and Query Researcher

Once the researcher is added, the admin must click the “View Researcher” button which opens a page with list of researchers where the information is shown and can be edited or deleted.


To edit or delete:

- The required row should be clicked, and the details are shown in the text fields below the table.
- Then the changes can be made by clicking the “Save” button.
- Or clicking the “Delete” button will simply remove the researcher.

To query:

- Enter the ID of the researcher in the top- right text field and click the search icon.
- It will open the details page (Fig 3.2).

Researchers



| SN | ID | Name | Address | Contact | Email | Username | Password |
|----|--------|----------------|---------|------------|------------------|----------|----------|
| 1 | 100001 | adfasdf | fasfa | +123213213 | c@g.com | khando | k |
| 2 | 100002 | Tsering Khando | jorpati | +977 9999 | tsering@emai.com | test | test |

Details

ID

2

Name

100002

Address

Tsering Khando

Contact

jorpati

Email

+977 9999

Username

tsering@email.com

Password

test

Save

Delete

Fig 3.1: View Researchers

Researcher Details

ID

100002

Name

Tsering Khando

Address

jorpati

Contact

+977 9999

Email

tsering@email.com

Username

test

Password

test

Articles / Chapters Hired:

Back

Fig 3.2: Queried researcher's details

3.1.4 Add an Article

To add any article or chapter, the admin can open the form by clicking the “Add Book Chapter” or “Add Journal Article” or “Add Conference Article” as required. For example, journal article has been added in the given picture:



Add Journal Article

| | |
|------------------|-----------|
| ID | 300001 |
| Article Title | test |
| Journal Title | test2 |
| Publication Year | 1998 |
| Volume | 3 |
| Issue no. | 9 |
| Author(s) | test.test |
| Page No. | 99-111 |

- The ID here, is also auto generated.
- User needs to enter complete information and click “Add Article”.

3.1.5 View, Edit, Delete, Query and Hire Articles

Any article can be edited, deleted, queried and hired or issued to a researcher. Already hired article cannot be issued.

To edit or delete same steps of modifying or deleting researcher can be done.

To query:

- Enter the title of the article in the top- right text field and click the search icon.
- It will open the details page (Fig 3.4).

To Hire:

- There is dropdown list of researcher’s id and a button labelled “Issue” below it (Fig 3.3).
- Select the required row of article from the table and the researcher’s id from the dropdown list and click “Issue”.

Journal Articles

| SN | ID | Title | Journal Title | Publication Year | Volume | Issue No. | Author(s) | Page |
|----|--------|-------|---------------|------------------|--------|-----------|-----------|--------|
| 1 | 300001 | test | test2 | 1998 | 3 | 9 | test,test | 99-111 |

Details

ID

300001

Select Research ID to issue:

100002

Issue

Article Title

test

Journal Title

test2

Publication Year

1998

Volume

3

Issue no.

9

Author(s)

test,test

Page No.

99-111

Save

Delete

Fig 3.3: View journal articles

Article Details

ID

300001

Article Title

test

Journal Title

test2

Publication Year

1998

Volume

3

Issue No.

9

Author(s)

test,test

Page No.

99-111

Back

Fig 3.4: Query details

3.1.6 Hired Articles

To view the list of articles hired and the researchers who hired, the admin must go to the “View Hired Articles” and the list shows up. The admin can then select the row and return the article back to library by clicking the “Return” button.

Hires

| Researcher ID | Article No. | Article Title |
|---------------|-------------|---------------|
| 100002 | 300001 | test |

Return

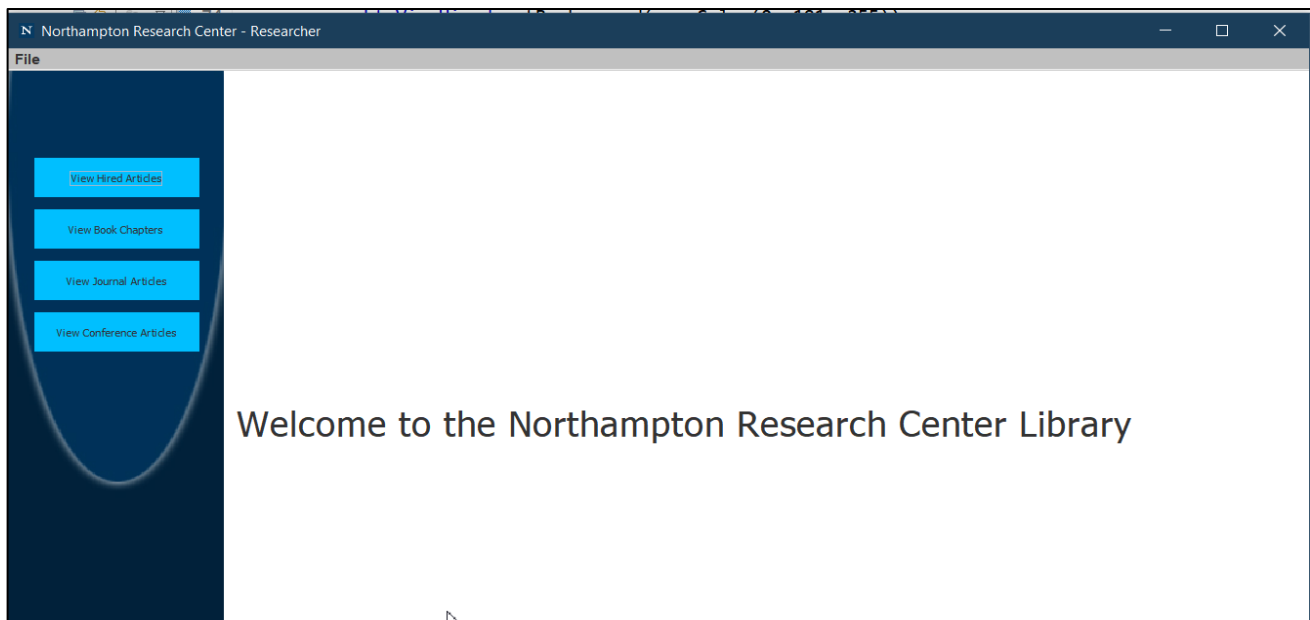
3.1.7 Researcher Sign In

The researcher can sign in with the username and password created by the admin and sign in. This opens the researcher’s window.

Username

Password

Sign In



Here, the researcher can list the articles hired by themselves only and the list of available articles only.

- By clicking “View Hired Articles” the list of articles hired by the logged in user is shown.

Hires

| Article No. | Title |
|-------------|-------|
| 300001 | test |

- Clicking the “View Book Chapters”, “View Journal Articles” or “View Conference Articles” will open the list of articles which are not borrowed by any researcher. Borrowed articles will not be shown.

Book Chapters

| ID | Title | Author(s) | Publication Year | Book Editor | Place of Publication | Publisher | Page |
|--------|-----------|-----------|------------------|-------------|----------------------|-----------|------|
| 200001 | new btest | test | 1939 | test | NEPAL | ASD | |

[Details](#)

| | |
|----------------------|-----------|
| ID | 200001 |
| Title | new btest |
| Author(s) | test |
| Publication Year | 1939 |
| Book Editor | test |
| Place of Publication | NEPAL |
| Publisher | A SD |

- The researcher can sign out just like the admin does.

4. Evidence of Testing

The system was tested on the basis of the functionalities implemented and code implemented.

4.1 Test Logs(Black Box Testing)

This includes all the information on tests carried out and whether they worked or not with screenshots.

| SN | Input | Expected Output | Actual Output | Remarks |
|-----|---|--|--|---------|
| 1. | Incorrect username and password | Show pop up message and cannot log in | Appendix 1 | Working |
| 2. | Correct username and password | Sign in successfully | Appendix 2,3 | Working |
| 3. | Sign out | Signs out and open Sign in window | Appendix 4 | Working |
| 4. | Add new researcher with empty fields | Prompts to enter complete information | Appendix 5 | Working |
| 5. | Add new researcher with invalid contact and email | Prompts to enter valid contact and email | Appendix 6,7 | Working |
| 6. | Add new researcher with correct data | Records successfully | Appendix 8 | Working |
| 7. | Add new book chapter with incomplete data | Prompts to enter complete information | Appendix 9 | Working |
| 8. | Add new book chapter with required data | Saves data | Appendix 10 | Working |
| 9. | Add new Journal Article with incomplete data | Prompts to enter complete information | Appendix 11 | Working |
| 10. | Add new Journal Article with required data | Saves data | Appendix 12 | Working |
| 11. | Add new Conference Article with incomplete data | Prompts to enter complete information | Appendix 13 | Working |
| 12. | Add new Conference Article with required data | Saves data | Appendix 14 | Working |
| 13. | Search researcher with correct ID | Shows details page | Appendix 15 | Working |
| 14. | Search researcher with incorrect ID | Prompts "Not found" | Appendix 16 | Working |
| 15. | Search articles with correct titles | Shows details page | Appendix 17 | Working |
| 16. | Search articles with incorrect titles | Prompts "Not found" | Appendix 18 | Working |
| 17. | Hire a non-hired article | Hires | Appendix 19 | Working |
| 18. | Hire already hired article | Unsuccessful | Appendix 20 | Working |
| 19. | Return article | Returns | Returned successfully | Working |
| 20. | Researcher can view all the articles hired | Shows the list | Appendix 21 | Working |
| 21. | Researcher can only view articles not on hire | Shows only the articles not hired | Shows only the articles not hired, otherwise empty table | Working |
| 22. | Edit or delete any data of researcher or article | Successful | Information modified or deleted | working |

| | | | | |
|-----|---|--|---------------------------------|---------|
| 23. | Click on Edit or Delete or Issue without selecting the row from table | Prompt select row | Prompts select row. Appendix 22 | Working |
| 24. | Click search icon with empty field | Prompts enter id or title | Prompts enter id or title | Working |
| 25. | Details of article can be viewed by the hired list by researcher | Hired articles should show all the details | Only ID and name are shown | Partial |

4.2 White Box Testing

Two methods from the code implementations having two or more decision points were taken.

4.2.1 View Conference Article

First, the method with displayed conference articles to the researcher was tested which is shown below:

```
//conference Article
public void showConferenceArticles()
{
    if(reader.readConferenceData()!=null)
    {
        boolean found = false;
        conferenceArray = reader.readConferenceData();
        ConferenceArticles reObj = new ConferenceArticles();
        for(int i1=0; i1<conferenceArray.size(); i1++)
        {
            if(reader.readHireDetails()!=null)
            {
                issueArray = reader.readHireDetails();
                IssueRecords check = new IssueRecords();
                for(int i=0; i<issueArray.size(); i++)
                {
                    reObj=conferenceArray.get(i1);
                    check=issueArray.get(i);
                    if(!reObj.getId().equals(check.getAid()))
                    {
                        found = false;
                    }
                    else
                    {
                        found = true;
                        break;
                    }
                }
            }
            if(found==false)
            {
                reObj=conferenceArray.get(i1);
                Object[] data= {reObj.getId(), reObj.getTitle(), reObj.getAuthor(), reObj.getYear(), reObj.getBookEditor(), reObj.getPublicationPlace(), reObj.getPage()};
                conferenceTableView.addRow(data);
            }
        }
        Object tableHead[] = {"ID","Title","Author(s)","Publication Year","Editor(s)","Conference Location","Page"};
        conferenceTableView.setColumnIdentifiers(tableHead);
    }
}
```

The above picture shows that there are 5 decision points. The cyclomatic complexity is calculated as 1+ number of decision points which shows: 1+5=6. Therefore, the cyclomatic complexity of the above code is 6. Hence, there is no risk.

Table 4.1:Test case

| S.N. | Path | Test Case |
|------|--|--|
| 1. | Decision point 1 false | reader.readConferenceData() = null |
| 2. | Decision point 1 true, decision point 2 true, decision point 3 false | reader.readConferenceData() != null, reader.readHireDetails()= null, found = false |
| 3. | Decision point 1 true, decision point 2 true, decision point 3 true, decision point 4 true, decision point 5 true | reader.readConferenceData() != null, reader.readHireDetails() != null, found = false |
| 4. | Decision point 1 true, decision point 2 true, decision point 3 true, decision point 4 true, decision point 5 false | reader.readConferenceData() != null, reader.readHireDetails() != null, found = true |

4.2.2 View Hired Articles

This is the method to display the articles hired by the researcher oneself.

```
//articles
public void showHires() {
    if(reader.readHireDetails()!=null)
    {
        issueArray = reader.readHireDetails();
        IssueRecords reObj = new IssueRecords();
        for(int i=0; i<issueArray.size(); i++)
        {
            reObj=issueArray.get(i);
            if(reObj.getRid().equals(getResearcherUsernamefromHire())) {
                Object[] data= {reObj.getAid(), reObj.getArticleName()};
                hireTableView.addRow(data);
            }
        }
        Object tableHead[]= {"Article No.", "Title"};
        hireTableView.setColumnIdentifiers(tableHead);
    }
}
```

There are 3 decision points. Thus, the cyclomatic complexity is $1+3=4$. It means there are no risks.

Table 4.2: Test case

| S.N. | Path | Test Case |
|------|--|---|
| 1. | Decision point 1 false | reader. readHireDetails () = null |
| 2. | Decision point 1 true, decision point 2 true, decision point 3 false | reader. readHireDetails () != null, i< reader. readHireDetails ().size() |
| 3. | Decision point 1 true, decision point 2 true, decision point 3 true | reader. readHireDetails () != null, i< reader. readHireDetails ().size(), view data |

4.3 List of bugs and weaknesses:

There are few limitations to the new system which are as follow:

- The system does not show complete details of articles hired out when the admin queries researchers accounts i.e view details of a specified researcher based on his/her researcher identification numbers. It only shows the article numbers.
- Similar limitation exists when the researcher tries to list the articles they have borrowed, only id and title are shown.

5. Conclusion

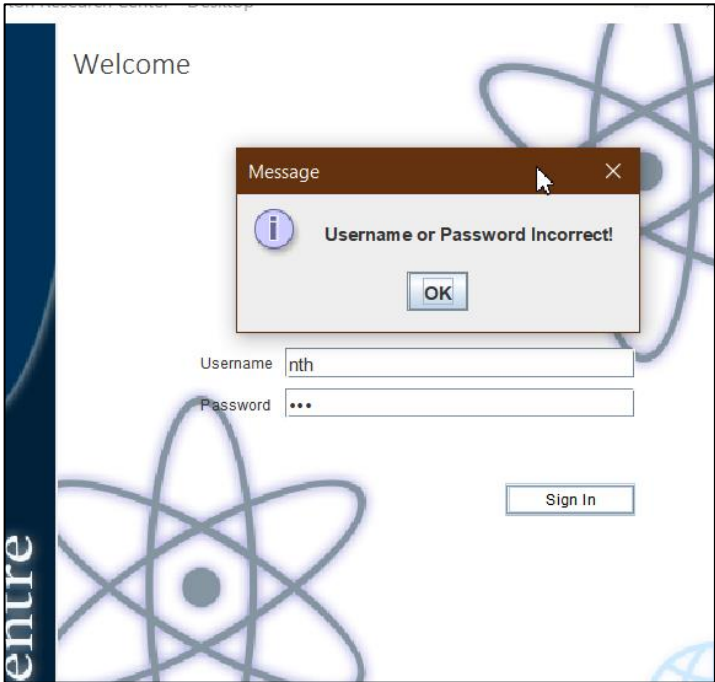
The given project was completed in the given time and with the instructed tools. The new system was developed with the MVC concept in Java.

The required basic functionalities have been met and the tests of the implementation were carried out as per required. However, the author, being aware of the few limitations, wishes to achieve them and keeps no limitation to the system.

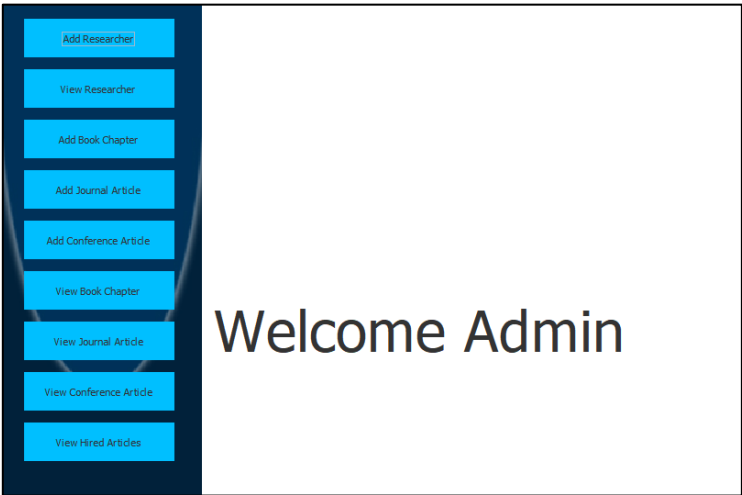
References

1. Code referred to validate email address (AdminController.java line 78):
TutorialsPoint,2019. Validate Email address in Java. [ONLINE] Available at:
<https://www.tutorialspoint.com/validate-email-address-in-java> [Accessed 20 June, 2019].
2. Code to validate contact number(AdminController.java line 84):
HowToDoInJava, 2019. Java Regex Tutorial. [ONLINE] Available at:
<https://howtodoinjava.com/regex/java-regex-validate-international-phone-numbers/> [Accessed 20 June, 2019]
3. Code to set table column width (AdminModel.java line 224):
stackoverflow, 2012. Java JTable setting Column Width. [ONLINE] Available at:
<https://stackoverflow.com/questions/953972/java-jtable-setting-column-width> [Accessed 25 June, 2019]

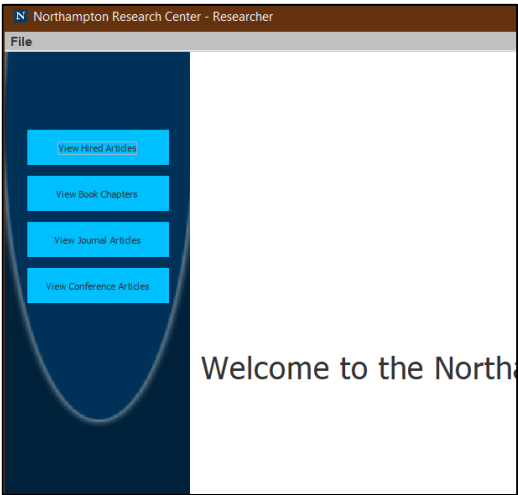
Appendix



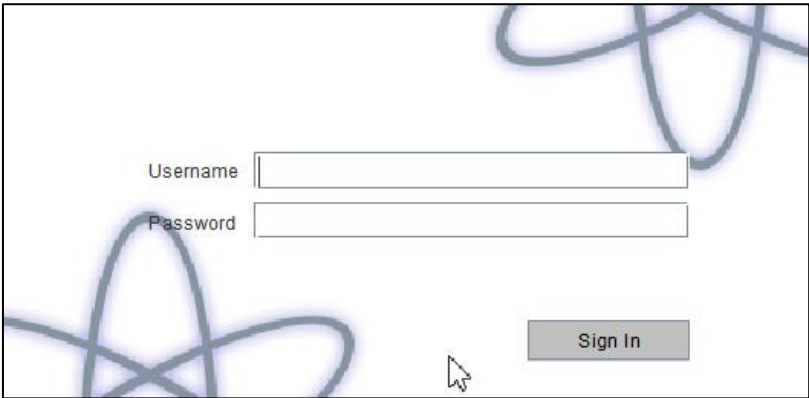
Appendix 1



Appendix 2



Appendix 3



Appendix 4

Add New Researcher

ID No.

Name

Address

Contact

Email

[\(Create username and password\)](#)

Username

Appendix 5

Address

Contact

Email

[\(Create username and password\)](#)

Name

Address

Contact

Email

Appendix 7

Appendix 6

[username and password\)](#)

Appendix 8

Add a Book Chapter

ID: 200002

Title:

Author(s):

Publication Year:

Book Editor:

Message

i Enter complete information

OK

Appendix 9

Add a Book Chapter

ID: 200002

Title: test

Author(s):

Publication Year:

Book Editor:

Message

i Book Chapter Added

OK

Appendix 10

ID: 300003

Title: test

Author(s): teest

Publication Year: 2014

Message

i Journal Article Added

OK

Appendix 12

Add Journal Article

ID: 300002

Title:

Author(s):

Publication Year:

Message

i Enter complete information

OK

Appendix 11

Add Conference Article

ID: 400003

Title:

Author(s):

Publication Year:

Message

i Enter complete information

OK

Appendix 13

Title: test

Message

i Conference Article Added

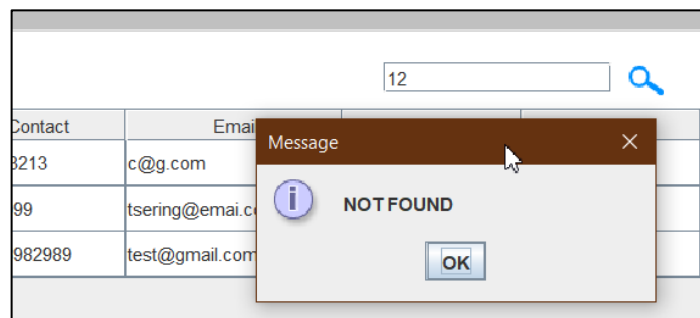
OK

Appendix 14

Researcher Details

| | |
|----------------------------|-------------------|
| ID | 100002 |
| Name | Tsering Khando |
| Address | jorpati |
| Contact | +977 9999 |
| Email | tsering@email.com |
| Username | test |
| Password | test |
| Articles / Chapters Hired: | |

Appendix 15

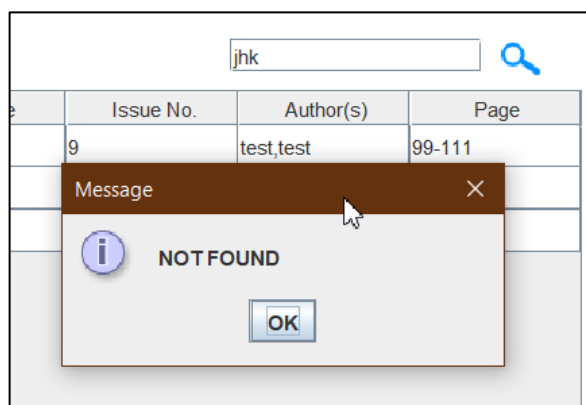


Appendix 16

Article Details

| | |
|------------------|-----------|
| ID | 300001 |
| Article Title | test |
| Journal Title | test2 |
| Publication Year | 1998 |
| Volume | 3 |
| Issue No. | 9 |
| Author(s) | test,test |
| Page No. | 99-111 |

Appendix 17



Appendix 18

Journal Articles jkh

| SN | ID | Title | Journal Title | Publication Year | Volume | Issue No. | Author(s) | Page |
|----|--------|-------|---------------|------------------|--------|-----------|-----------|--------|
| 1 | 300001 | test | test2 | 1998 | 3 | 9 | test,test | 99-111 |
| 2 | 300002 | test | test | 1999 | 4 | 4 | test | 5-9 |
| 3 | 300003 | test | teest | 2014 | 2 | 3 | | |

ID:

Article Title:

Journal Title:

Select Research ID to issue: 100002

Message ✕

Article Issued

Appendix 19

ID:

Article Title:

Journal Title:

Publication Year:

Select Research ID to issue: 100002

Message ✕

Article cannot be issued to more than one resesarcher.

Appendix 20

Hires

| Article No. | Title |
|-------------|-------|
| 300001 | test |

Appendix 21

| SN | ID | Title | Journal Title | Publication Year | Volume | Issue No. | Author(s) | Page |
|----|--------|-------|---------------|------------------|--------|-----------|-----------|--------|
| | 300001 | test | test2 | 1998 | 3 | 9 | test,test | 99-111 |
| | 300002 | test | test | 1999 | 4 | 4 | test | 5-9 |
| | 300003 | test | teest | 2014 | 2 | 3 | | |

ID:

Article Title:

Journal Title:

Select Research ID to issue: 100001

Message ✕

Select a row first.

Appendix 22