



Faculty of Computing and Information Technology

BAIT1083 Visual Programming Assignment 2020/2021

Programme : RSD 2

Tutorial Group : 11

System Title : TARK Library System

Tutor : Dr. Goh Ching Pang

Submission Date : 4/12/2020

Student Name	Registration No.	Module Handled
Soon Yi Hong	20WMR09372	Lend and Return Module
Douglas Lee Yue Zhang	20WMR09251	Login and Registration Module
Kelvin Lim Ding Qi	20WMR09274	Book Management Module
Chong Jia Jun	20WMR09234	Book Fine Module & Student Management

BAIT1083 Visual Programming - Assignment Evaluation Form					
Student Name		Soon Yi Hong	Douglas Lee Yue Zhang	Kelvin Lim Ding Qi	Chong Jia Jun
Assessment Criteria		Marks Awarded			
CLO	Criteria				
3	Description of System and Targeted Users (5 marks)				
3	Detailed Functional Modules Descriptions (10 marks)				
2	GUI and Navigation Design (10 marks)				
2	Program / Business Logic Design (10 marks)				
2	Exception Handling (10 marks)				
2	Database Integration (10 marks)				
2	System Reports (10 marks)				
3	Teamwork (10 marks)				
3	Final Documentation (15 marks)				
3	Presentation (10 marks)				
	Total				

1. Description of System and Targeted Users

This system will allow students to lend books and return books from the library. Librarians, the main user of this library system will handle all of the operations of the system in order that students can successfully lend and return books. Librarians can also do other basic operations to manage the library. Librarians can manage all of the data of books through the book management module in the system. Librarians can use this system to add, edit, delete and update the data in the library as the system provides these basic functions. Secondly, librarians will use this system to help students to lend and borrow books. Librarians will use the lend and borrow books module to achieve this so the system will record which students borrowed the book. Lastly, there are a lot of cases that students will return the books after the due date. This will result in a fine for them due to not returning the books on time. The system will also provide a Book Fine and Penalty module to keep track of the books being lent out to students. The system will help librarians to automatically calculate the total fine that the students have to pay after not returning the books on time. Last but not least, the system will generate reports for each module in order to keep track of the activities of the library.

The main targeted users of this library system will be librarians. Librarians will be conducting most of the operations to help out students to lend or return the books successfully. Librarians will also have to manage the data of the books in the library through the book management module. For example, if there is an arrival of new books, librarians have to register all of the relevant details of the books into the system. Another user of this system will be the admin of the library. Admin can access all of the modules in the system. Only the admin can generate reports of each module to keep track of the activities in the library. Admin also can only add new librarians by helping them create new accounts for them.

2. Detailed Functional Module Description:

1. Login and Registration module -

In this module, the system will allow users to register accounts and login to the system to perform other actions later. The purpose of this module is to authenticate and authorize users. It is one of the important modules in the entire library system as it serves as a primary security of the system. Users will have to input their Username and password that they registered correctly to login to the system. The module will also allow users to recover their passwords if they have forgotten, and edit their passwords.

For registering steps, users must ensure that their input must meet with the requirements set by the system. For example, passwords must have at least 8 characters and email addresses must be in the correct format. If one of the requirements are not met, the system will notify the user to make corrections. If everything meets the requirements, the system will notify the users that it is successful and it will automatically save the details into the system's database. Users will just have to use the username and password that they just registered to login to their respective accounts.

If users have clicked on the “Print report” function, “Students” function or “Roles” function, the system will pop up an error message saying that the users are not authorized to use these functions. Only the admin can access these functions.

The module will have 2 roles which is librarian and admin.

Example of each roles and what can these roles do:

Admin : - Access all modules, register new accounts, generate reports, delete accounts.

Librarian : - Update account login details, access book management module, Lend and Return module, Fine and Penalty module.

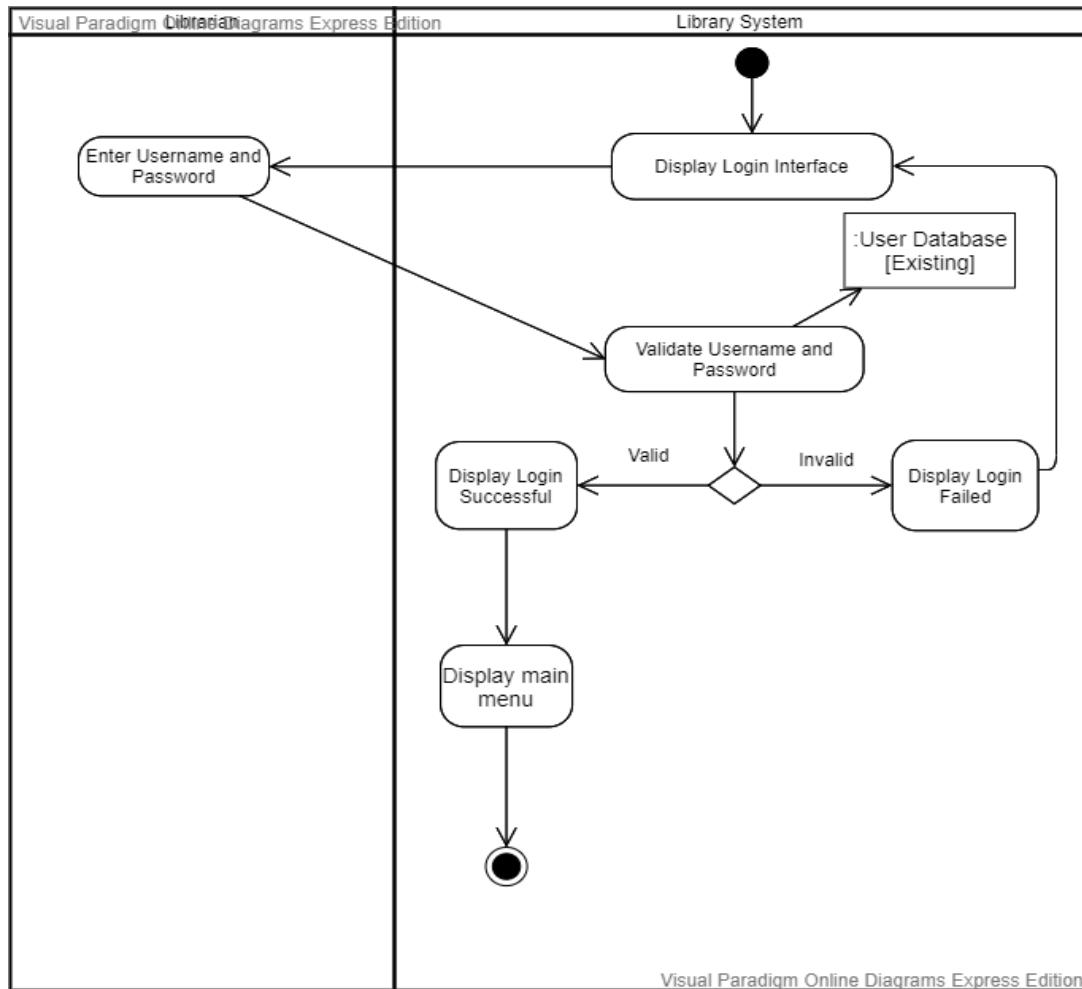


Diagram 1.1 The activity diagram shows the flow of the login module which will validate the user's username and password.

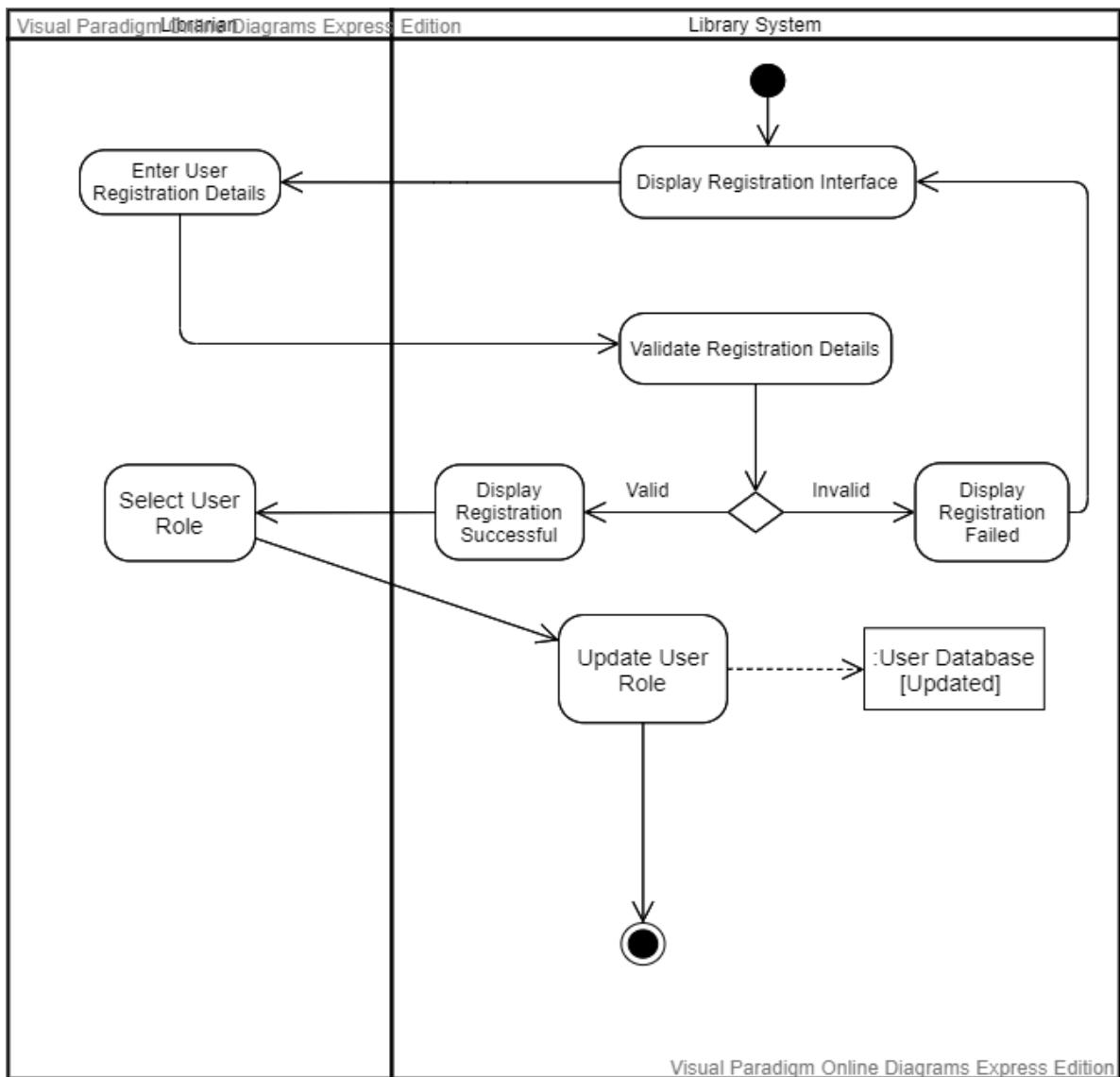


Diagram 1.2 The activity diagram above shows the registration of a new account process.

2. Book Management module - In this module, the system will allow only the staff to access this service and make the respective changes towards the existing book details. The purpose of this module is to allow the staff to always keep the book details up to date or make any necessary changes needed for the book details. First of all, there are two choices for the staff to choose which action they would like to perform which are book registration or book searching. If the staff chooses book registration, it will be directed to a book registration page where it will require the user to key in the book details of the new books. The book ID will be auto generated by the system where each book will have their own unique book ID. Then, the book details entered will be validated by the system and if it is valid, the new book details will be added to the system. Moreover, if the staff chooses book searching, they will be directed to the book searching interface which will require the staff to enter the book details of the book that they want to search such as book ID or book name. Then, after it has validated the entered details, it will display the searched book details and ask the staff to choose which action they would like to perform either edit or delete the existing book details. After having finished performing the respective action, the existing book details will be updated to the latest book details. The newest book details that have been added to the system will be allowed to be seen immediately after successfully added by any staff or students.

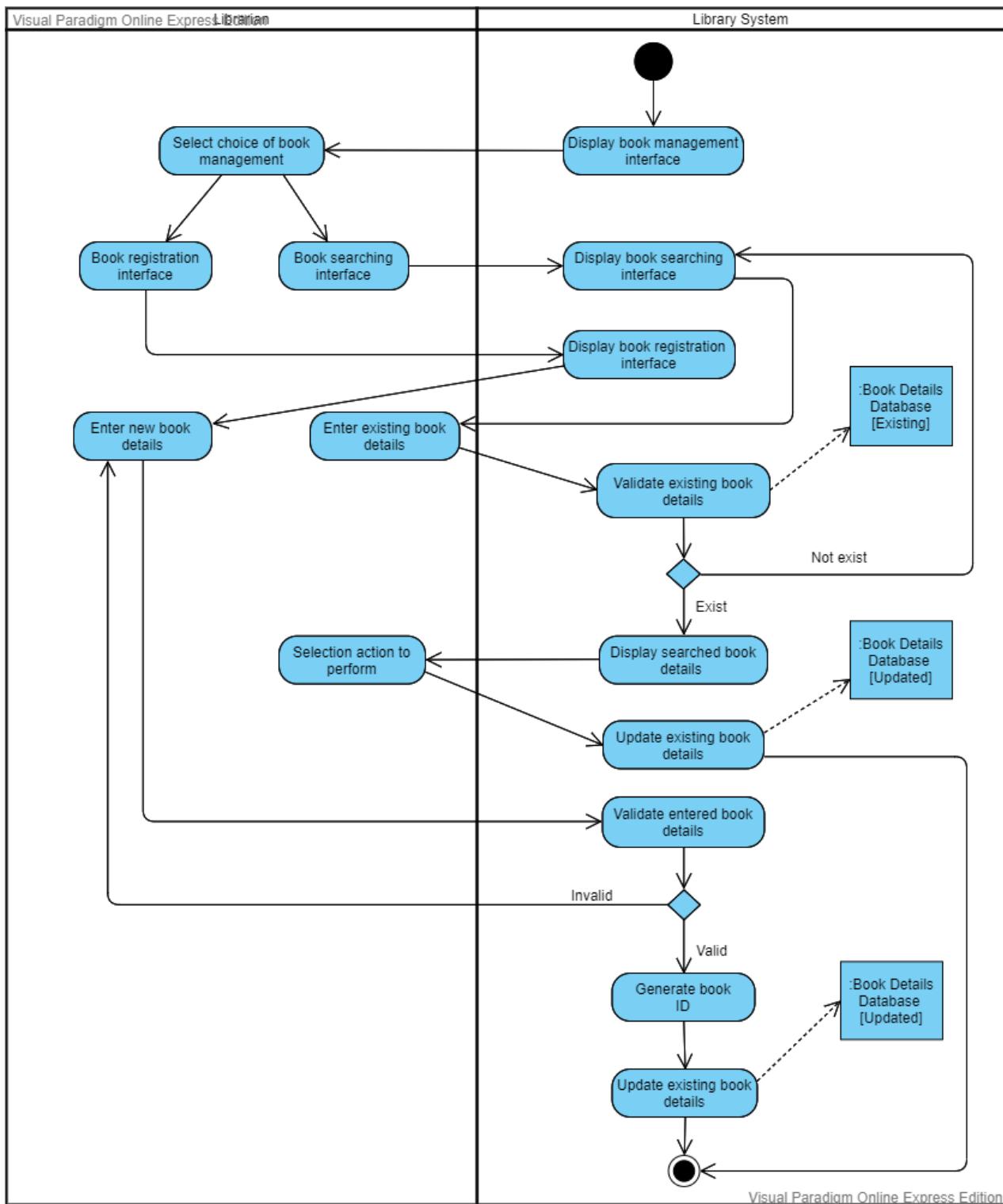


Diagram 2 The activity diagram above shows the registration of a new account process.

3.Lend and return module - Lend and return modules having some similar processes. First of all, the Return module required user credentials to login before accessing the Lend and Return module. The purpose of login is to let the system give authentication and authorize the user base on the user. If the user wants to use the return function, they will only be required to insert the book returning date of the lender. As the information of the lender is already taken before at the Lending function. So, we no longer need to repeat keys in the previous obtained details and information. But, in order to obtain the status librarians need to search for the lender ID. This module will let librarians lend and return books through the system. The system will also record all the books lending and returning detail as well as some brief lender's information.

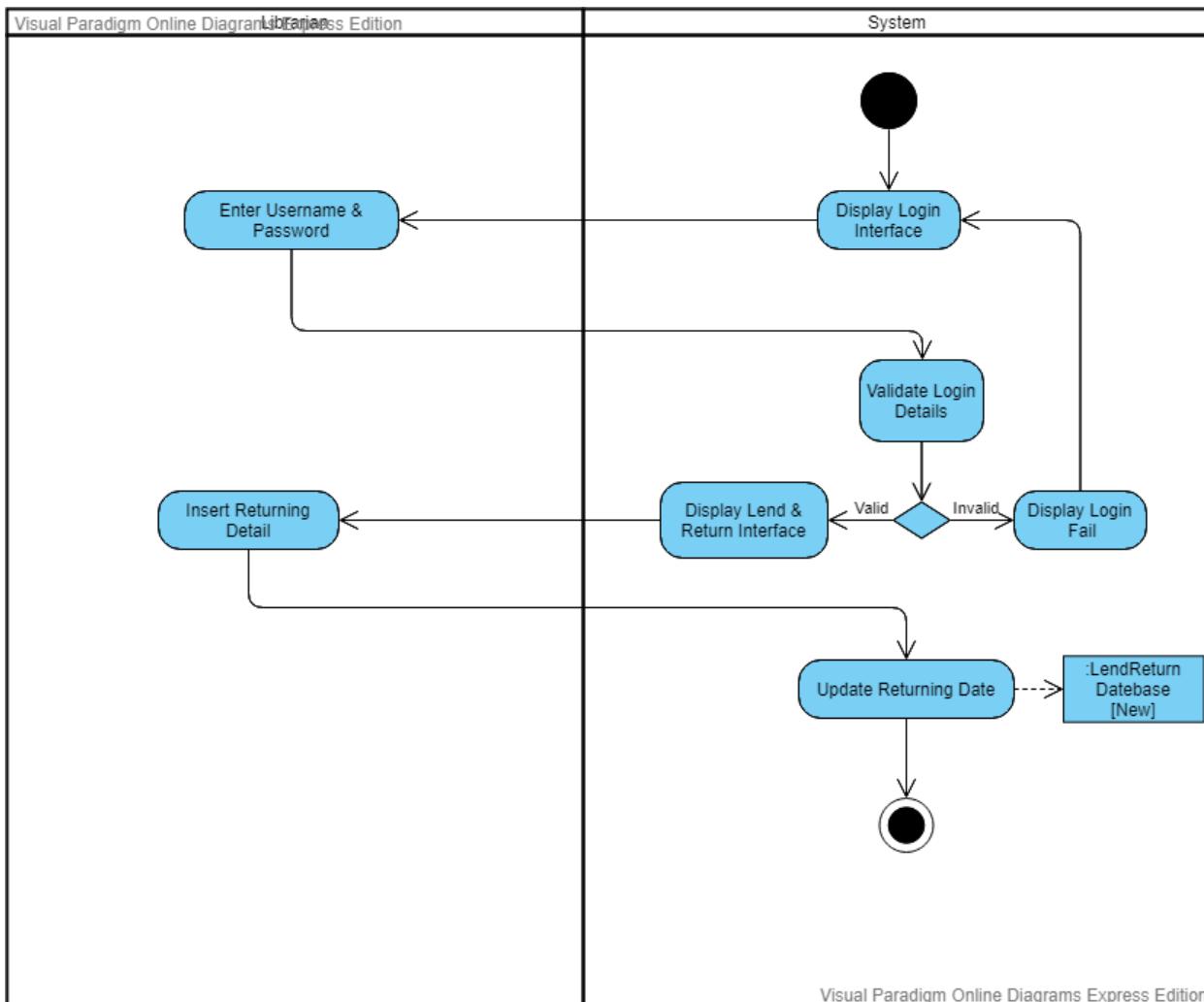


Diagram 3 The activity diagram shows the flow of the Lend & Return module which will let user's lend and return books.

4. Book Fine Module- When the borrower returns the book , the librarian has to enter the returnee as well as the borrowed book details. The system will then proceed to validate and check for late returns based on the book's borrowed date from the database. Fine will then be given to those returnees that return the book past the deadline, this module will help to calculate the amount of days that the deadline has passed as well as how much is the fine. If the borrower returns the book within the deadline, then there will be no fine issued. The librarian will then select the payment method that the borrower will pay their fine in , and then have the choice of whether the borrower wants to print the receipt for the payment.

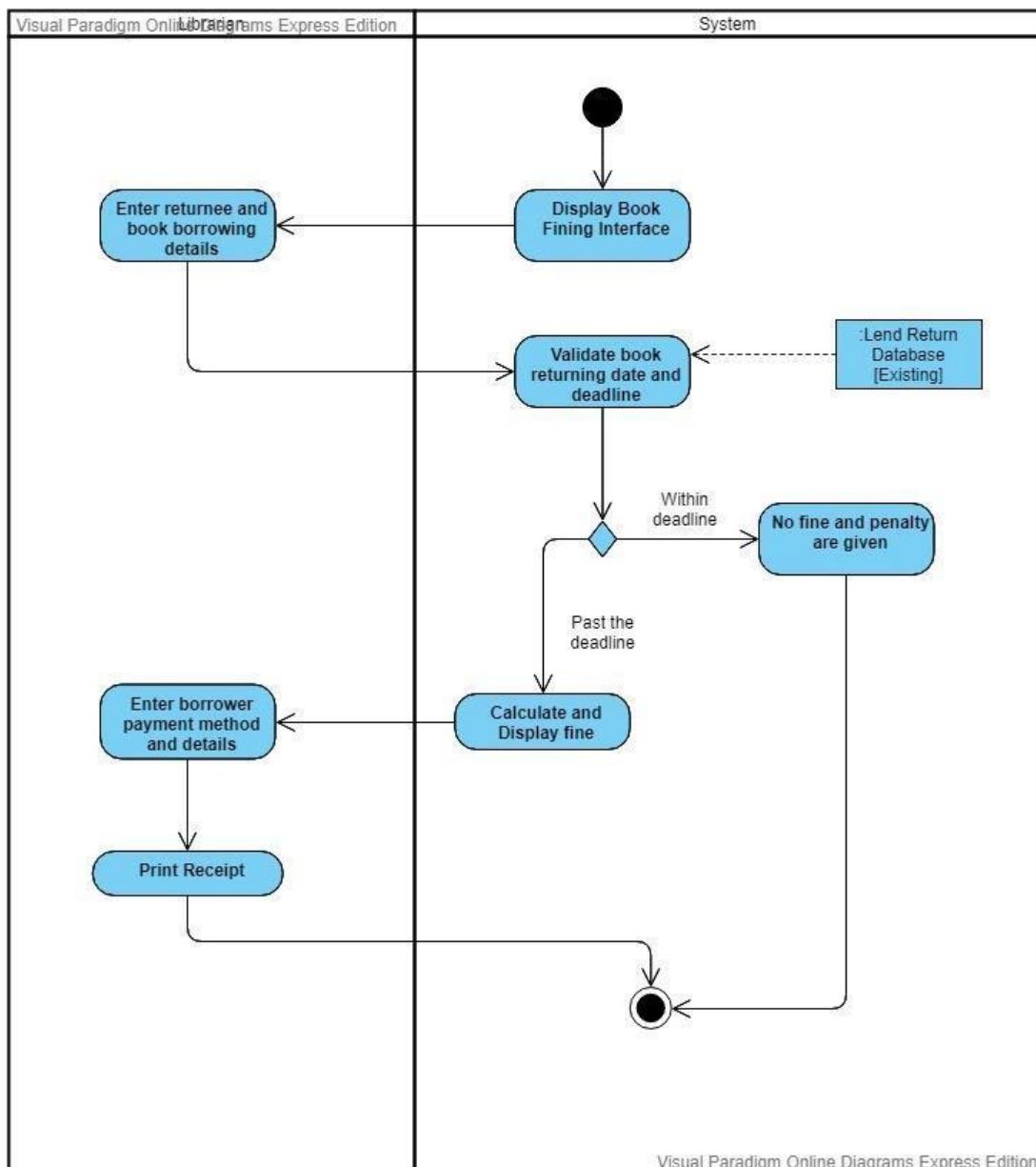


Diagram 4 This activity diagram shows the book fining process of the borrower when they exceed the deadline.

5. Student Management Module - Student management module allows the librarian to edit a student's information. Firstly, a student must be selected in order to view their student information. In the student management screen there will be 5 buttons, add allows the librarian to input new information, edit allows the librarian to edit and overwrite current existing student information with new data, delete allows the librarian to delete a student's information, clear allows the librarian to clear all data in the student information , and lastly exit which quits the form.

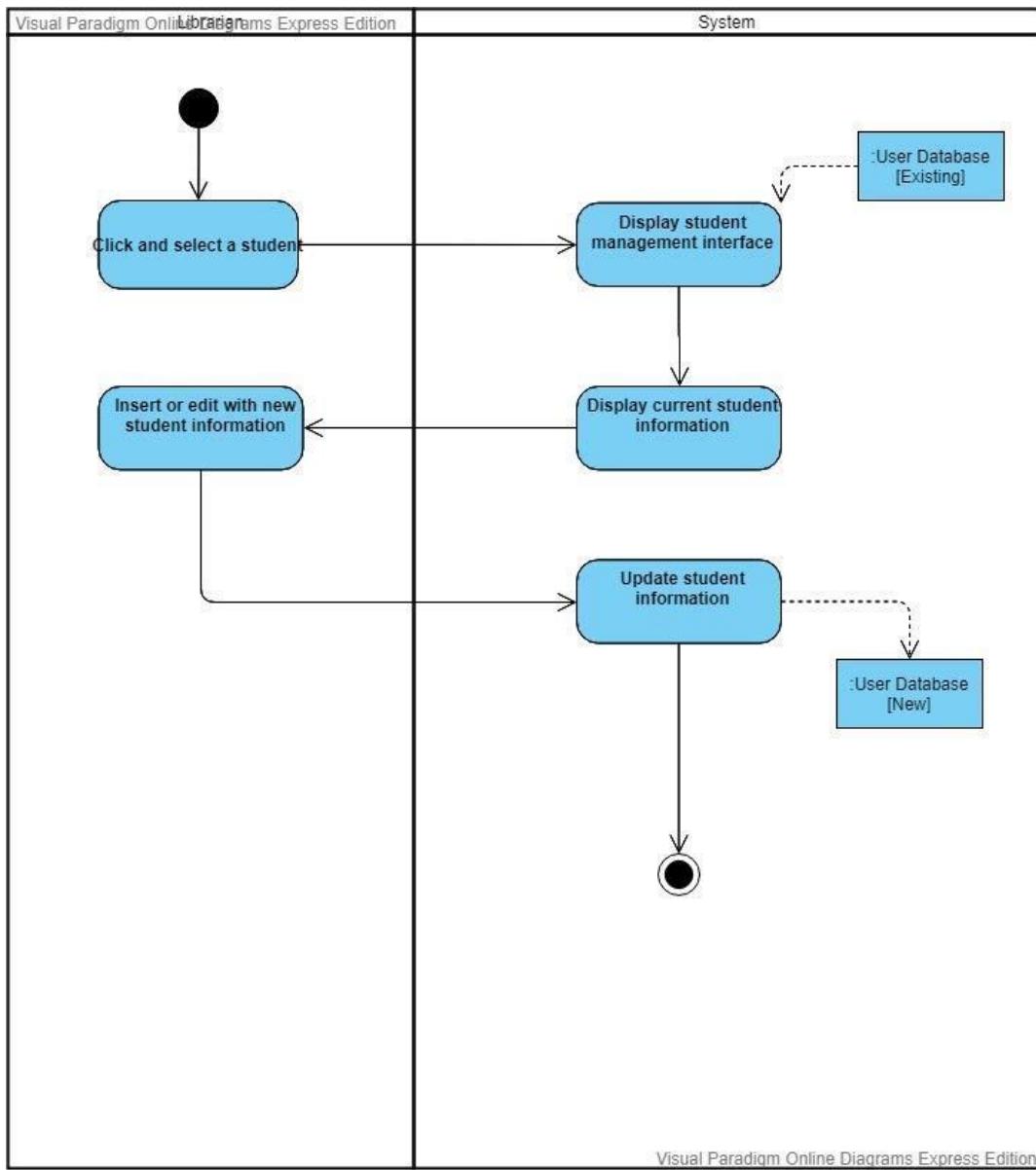


Diagram 5 The activity diagram shows the flow of the student management process in the library management system.

Allocation of Modules: Book management module - Kelvin Lim Ding Qi

Login module - Douglas Lee Yue Zhang

Lend and return module - Soon Yi Hong

Book Fine and Student Management module - Chong Jia Jun

3.User Manual

User Manual 3.1 - Login and Registration Module by Douglas Lee Yue Zhang

TARK Library system login page.

When the system is launched, the user will be greeted with the login page. If the user forgot his password, the forgotten password button can be pressed and the user will be redirected to the password recovery page. Users will be prompted to input again if the provided username or password did not match the database.

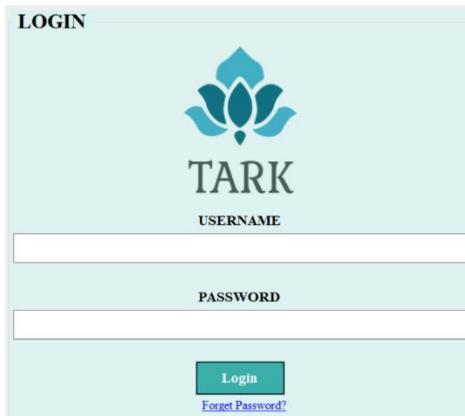


Figure 3.1.1 TARK library login page



Figure 3.1.2 Login Validation

Password Recovery

Users that have forgotten their password have to enter their login username as well as their email address. Their password will then be sent to their respective email address. The email address must be the same as the one that is being used to create the library system account and matches the username to prevent the new password from being in the wrong hands.

The screenshot shows a light blue 'Password Recovery' page. It contains two input fields: 'USERNAME' and 'EMAIL', both represented by black rectangular boxes. Below these fields is a teal-colored 'Confirm' button.

Figure 3.1.3 Password Recovery Page

The screenshot shows the same 'Password Recovery' page as Figure 3.1.3, but with different input values. The 'USERNAME' field contains 'james123' and the 'EMAIL' field contains 'james123@hotmail.com'. A small error message dialog box is overlaid on the page, displaying 'Username or Email Not Found' with an 'OK' button.

Figure 3.1.4 Incorrect registered email or username

The librarian or admin can change their password after logging into the homepage by clicking Account > Change Password.

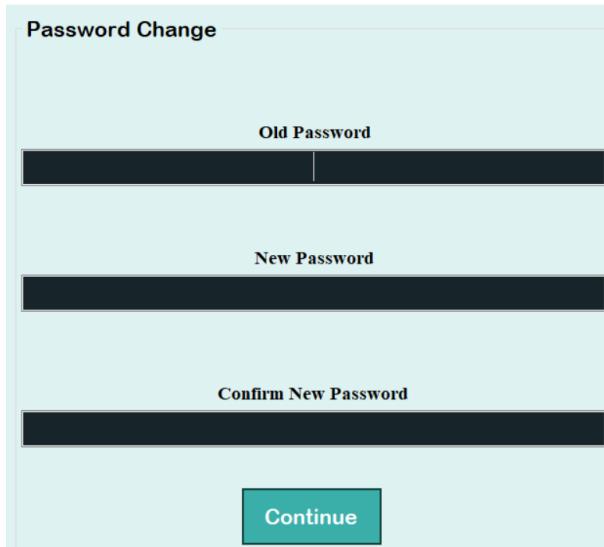


Figure 3.1.5 Change password page

Authorization

Users cannot access the “Print Report”, “Students” or “Roles” function. If they click on the buttons it will pop out an error message box and the system will disable those buttons so they would understand they could not access those functions.

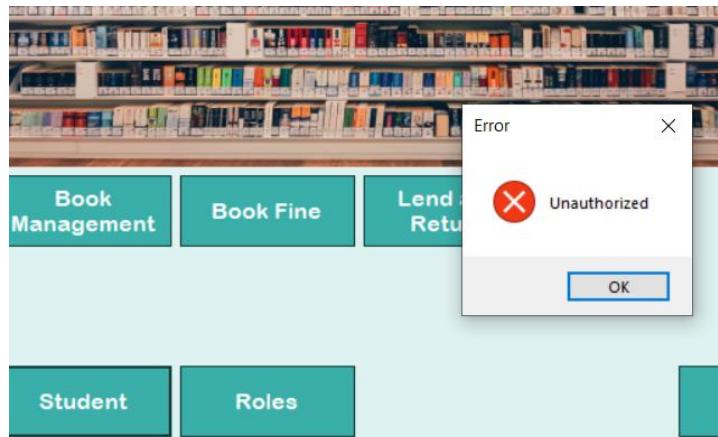


Figure 3.1.6 Unauthorized error message box



Figure 3.1.7 Disabled Buttons

Login Report Printing

Only Admin can then print the login report by clicking File > Print Report in the homepage.

LOGIN REPORT

Search By
Date From :

Date To :

	AccountID	Username	LoginTime
1	Z1001	ADMIN	3/12/2020
2	Z1001	ADMIN	3/12/2020
3	Z1001	ADMIN	3/12/2020
4	Z1001	ADMIN	3/12/2020
5	Z1001	ADMIN	3/12/2020
6	Z1001	ADMIN	3/12/2020

Figure 3.1.8 Login Report printing page



TARK
Summary Log In Report
=====
Prepare by TARK Library System
Date : 03/12/2020
Page : 1

Account ID	Username	Log In Time
Z1001	ADMIN	3/12/2020

--- End ---

Figure 3.1.9 Login report

Account Management

Account management allows the admin to manage the users that are being registered to the system. Admin can access the account management form by clicking on the “Roles” button at the homepage.

Account Management

Account Information					
Account ID :	Z1011				
Roles :	<input style="width: 100px; height: 25px; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;" type="button" value="User"/>				
Username :	<input style="width: 100%; height: 25px; border: 1px solid #ccc; border-radius: 5px;" type="text"/>				
Password :	<input style="width: 100%; height: 25px; border: 1px solid #ccc; border-radius: 5px;" type="text"/>				
Email :	<input style="width: 100%; height: 25px; border: 1px solid #ccc; border-radius: 5px;" type="text"/>				
<input style="width: 50px; height: 25px; border: 1px solid #007bff; border-radius: 5px; background-color: #007bff; color: white; font-weight: bold; padding: 2px 10px;" type="button" value="Add"/>	<input style="width: 50px; height: 25px; border: 1px solid #007bff; border-radius: 5px; background-color: #007bff; color: white; font-weight: bold; padding: 2px 10px;" type="button" value="Delete"/>	<input style="width: 50px; height: 25px; border: 1px solid #007bff; border-radius: 5px; background-color: #007bff; color: white; font-weight: bold; padding: 2px 10px;" type="button" value="Edit"/>	<input style="width: 50px; height: 25px; border: 1px solid #007bff; border-radius: 5px; background-color: #007bff; color: white; font-weight: bold; padding: 2px 10px;" type="button" value="Clear"/>	<input style="width: 50px; height: 25px; border: 1px solid #007bff; border-radius: 5px; background-color: #007bff; color: white; font-weight: bold; padding: 2px 10px;" type="button" value="Exit"/>	

	AccountID	Username	Password	Roles	Email
1	Z1001	ADMIN	zadminadmin	Admin	yihongsoo@
2	Z1002	JAMES	password123	User	james20@
3	Z1003	JIAJUN	password123	Admin	jiajunchon@
4	Z1004	XIAO MING	password123	User	xming123@
5	Z1005	PETER LIM	password123	User	peter_lim@
6	Z1006	JONATHAN TAN	jonathan999	User	jonathan@
7	Z1007	DAVID SIM	david47533	User	david_see@
8	Z1008	ELIZABETH KO...	elizabeth122	User	EzKoay@h
9	Z1009	NAM HUAT	simnamhuat12	User	namhuatsi@
10	Z1010	DICKSON TEOH	01212345678ab	User	dicksonTeoh@

Figure 3.1.10 Account Management Form

After filling in all the account information including selecting the role of the user, the ‘Add’ button can be pressed to insert a new account record into the database.

	AccountID	Username	Password	Roles	Email
Z1001	ADMIN	2adminadmin	Admin	yihongssoo	
Z1002	JAMES	password123	User	james20@h	
Z1003	JIAJUN	password123	Admin	jiajunchon	
Z1004	XIAO MING	password123	User	xming123@h	
Z1005	PETER LIM	password123	User	peter_lim@h	
Z1006	JONATHAN TAN	jonathan999	User	jonathan@h	
Z1007	DAVID SIM	david47533	User	david_see@h	
Z1008	ELIZABETH KO...	elizabeth122	User	EzKoay@h	
Z1009	NAM HUAT	sinnamhuat12	User	namhuat@h	
Z1010	DICKSON TEOH	01212345678ab	User	dicksonTeoh@h	
Z1011	ABU	abupassword123	User	abu123@h	

Figure 3.1.11 Account Management Add Function

Validation is done here as it requires the user to fill in all required fields and appropriate email format, appropriate password format.

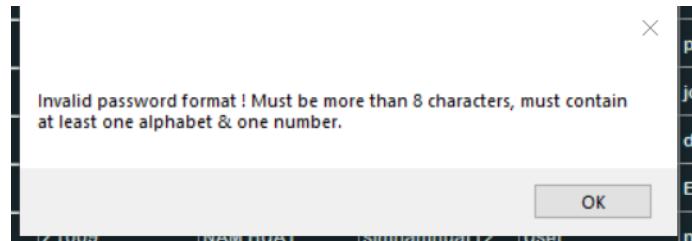


Figure 3.1.12 Invalid password format message box

×

Invalid Email Format !

OK

Figure 3.1.13 Invalid email format message box

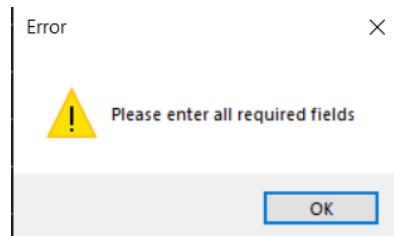


Figure 3.1.14 Enter all required fields error message box

The admin can also select a user from the database that they wish to edit. After clicking at a user, the existing information will then appear on the form. After the user is done changing the details, the “Edit” button can be pressed to update the new information regarding the user into the database.

Account Management

AccountID	Username	Password	Roles	Email
Z1001	ADMIN	2adminadmin	Admin	yihongsoon@
Z1002	JAMES	password123	User	james20@
Z1003	JIAJUN	password123	Admin	jiajunchon@
Z1004	XIAO MING	password123	User	xming123@
Z1005	PETER LIM	password123	User	peter_lim@
Z1006	JONATHAN TAN	jonathan999	User	jonathan@
Z1007	DAVID SIM	david47533	User	david_see@
Z1008	ELIZABETH KO...	elizabeth122	User	EzKoay@h
Z1009	NAM HUAT	simnamhuat12	User	namhuatsi@
Z1010	DICKSON TEOH	01212345678ab	User	dicksonTeoh@
Z1011	ABU	abupasword123	User	abu123@h

Account Information

Account ID : **Z1004**

Roles : **User**

Username : **Xiao Li**

Password : *********

Email : **xming123@hotmail.com**

Add **Delete** **Exit**

Edit **Clear**

	Z1004	Xiao Li	password123	User	xming123@
--	--------------	----------------	--------------------	-------------	------------------

Figure 3.1.15 Account Management Edit Function

The admin can also delete unwanted accounts from the database by selecting it and then press the “Delete” button.

Account Management

AccountID	Username	Password	Roles	Email
Z1001	ADMIN	2adminadmin	Admin	yihongsoon@
Z1002	JAMES	password123	User	james20@
Z1003	JIAJUN	password123	Admin	jiajunchon@
Z1004	Xiao Li	password123	User	xming123@
Z1005	PETER LIM	password123	User	peter_lim@
Z1006	JONATHAN TAN	jonathan999	User	jonathan@
Z1007	DAVID SIM	david47533	User	david_see@
Z1008	ELIZABETH KO...	elizabeth122	User	EzKoay@h
Z1009	NAM HUAT	simnamhuat12	User	namhuatsi@
Z1010	DICKSON TEOH	01212345678ab	User	dicksonTeoh@
Z1011	ABU	abupasword123	User	abu123@h

Account Information

Account ID : **Z1011**

Roles : **User**

Username : **ABU**

Password : *********

Email : **abu123@hotmail.com**

Add **Delete** **Exit**

Edit **Clear**

Would you like to delete the selected role details?

Figure 3.1.16 Account Management Delete Function

The “Clear” button can be used to reset all the account information on the form.

The screenshot shows a Windows-style application window titled "Account Management". The title bar has a close button. Below the title bar is a section labeled "Account Information". Inside this section are five input fields with labels: "Account ID : Z1012", "Roles : User", "Username : [redacted]", "Password : [redacted]", and "Email : [redacted]". At the bottom of the window are six buttons arranged in two rows: "Add", "Delete", "Edit", "Clear", and "Exit".

Figure 3.1.17 Account Management Clear Function

Report Printing

The report of current existing accounts registered in the database can be printed by selecting File > Print Report in the Account Management Form.

The screenshot shows a report titled "ROLES REPORT". At the top right of the report area is a "Print" button. The main part of the screen is a table with the following data:

AccountID	Username	Password	Roles	Email
Z1001	ADMIN	2adminadmin	Admin	yihongsoon@y...
Z1002	JAMES	password123	User	james20@yahoo...
Z1003	JIAJUN	password123	Admin	jiajunchong20...
Z1004	Xiao Li	password123	User	xming123@hotmail...
Z1005	PETER LIM	password123	User	peter_lim@yahoo...
Z1006	JONATHAN TAN	jonathan999	User	jonathan@gma...
Z1007	DAVID SIM	david47533	User	david_see12@hot...
Z1008	ELIZABETH KO...	elizabeth122	User	EzKoay@hotmail...
Z1009	NAM HUAT	simnamhuat12	User	namhuatsim@...
Z1010	DICKSON TEOH	01212345678ab	User	dicksonTeo@g...
Z1011	ABU	abupasword123	User	abu123@hotmail...

Figure 3.1.18 Report printing page



TARK

Summary Account Report
=====

Prepare by TARK Library System
Date : 03/12/2020
Page : 1

Account ID	Roles	Username
Z1001	Admin	ADMIN
Z1002	User	JAMES
Z1003	Admin	JIAJUN
Z1004	User	Xiao Li
Z1005	User	PETER LIM
Z1006	User	JONATHAN TAN
Z1007	User	DAVID SIM
Z1008	User	ELIZABETH KOAY
Z1009	User	NAM HUAT
Z1010	User	DICKSON TEOH
Z1011	User	ABU

--- End ---

Figure 3.1.19 Account report

Exception Handling

```

Try
    If lblAccountID.Text <> "" And cmbRoles.Text <> "" And txtUsername.Text <> "" And msktxtPassword.Text <> "" Then

        'Dim isDuplicated As Boolean = db.Items.Any(Function(o) o.Id = i.Id)'
        'Dim isDuplicated As Boolean = db.Accounts.Any(Function(o) o.Username.ToUpper = txtUsername.Text.ToUpper)'
        'If isDuplicated = True Then
        '    Console.WriteLine("ID already exist!", MessageBoxIcon.Error)'
        '    MessageBox.Show("Username Already Exist")'
        If IsValidPasswordFormat(msktxtPassword.Text) = False Then
            MessageBox.Show("Invalid password format ! Must be more than 8 characters, must contain at least one alphabet & one number.")
        ElseIf IsValidEmailFormat(txtEmail.Text) = False Then
            MessageBox.Show("Invalid Email Format !")
        ElseIf IsValidUserNameFormat(txtUsername.Text) = False Then
            MessageBox.Show("Invalid UserName, only alphabets are allowed !")
        Else
            i.Roles = cmbRoles.Text
            i.Username = txtUsername.Text
            i.Password = msktxtPassword.Text
            i.Email = txtEmail.Text
            db.SubmitChanges()
            MessageBox.Show("Record Updated!", "Info", MessageBoxButtons.OK, MessageBoxIcon.Information)
            BindData()
            IDload()
        End If
    Else
        errmsg = "Please enter all required fields"
        MessageBox.Show(errormsg, "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning)
    End If
Catch ex As Exception
    MessageBox.Show("Please fill in the blank/ Insert wrong data type!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning)
End Try

```

User Manual 3.2 – Book Management Module by Kelvin Lim Ding Qi

Book Management

The book management module allows librarians to enter a new book, edit existing book information and deletion of book information.

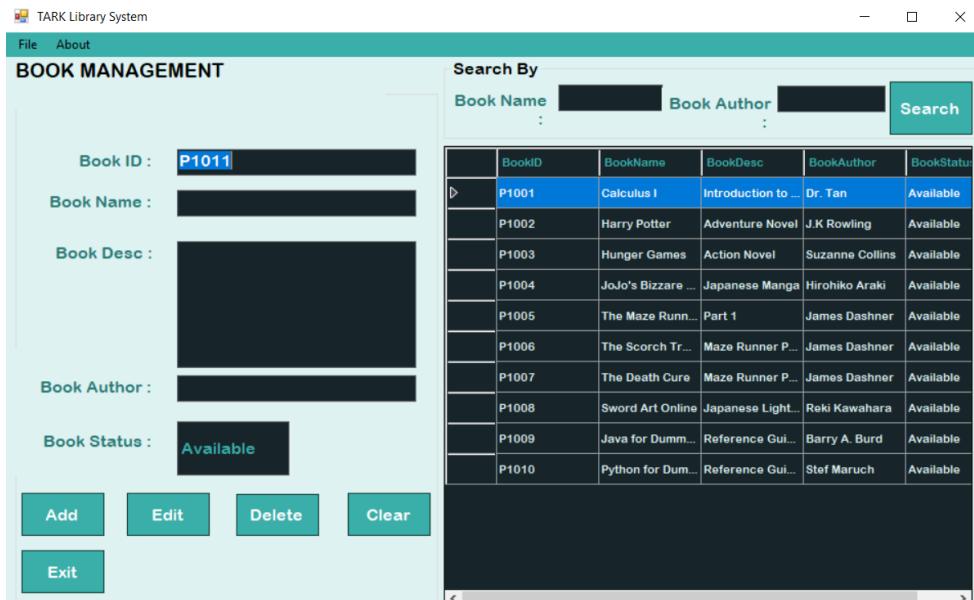


Figure 3.2.1 Book Management Page

The librarian can add a new book into the system. After entering the new book information into the form, the “Add” button can be pressed to enter the book into the system. All fields must be filled in order to enter the new book into the system.

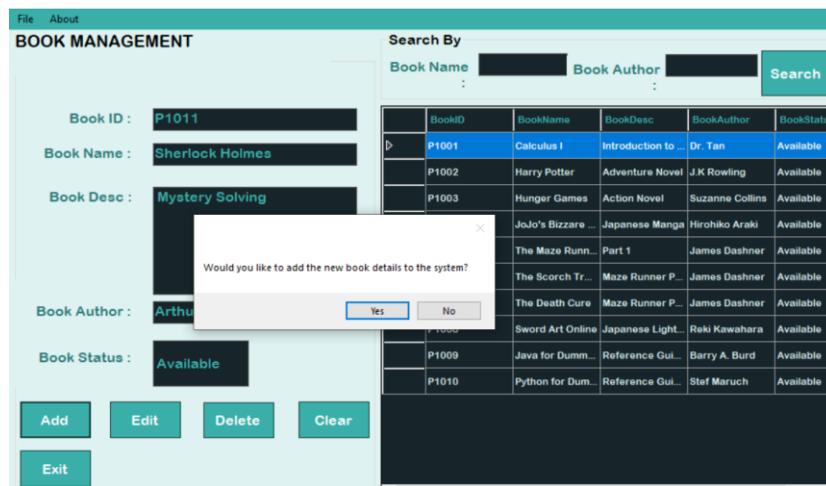
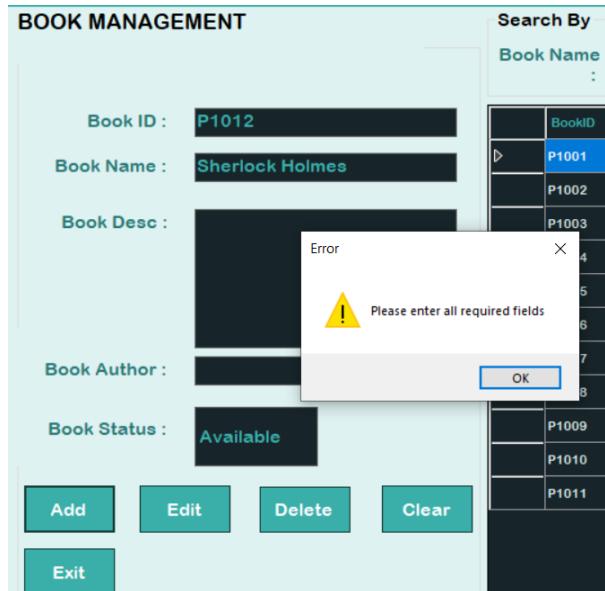
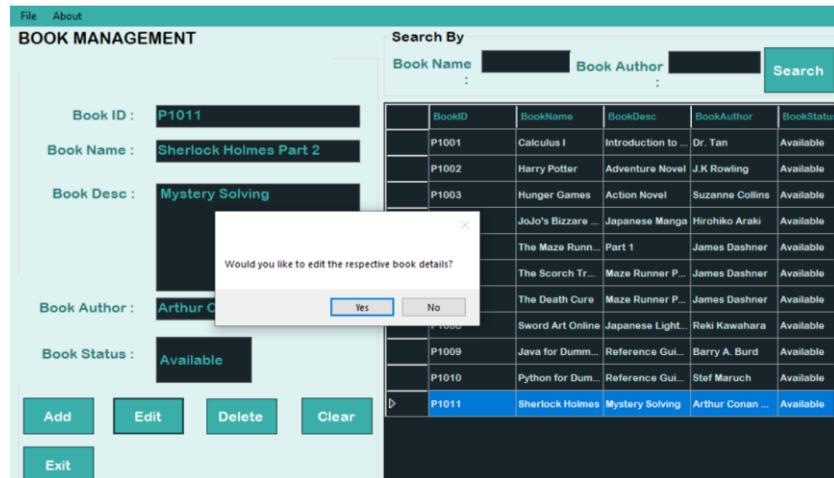


Figure 3.2.2 Book Add Function

Figure 3.2.3 “Enter all required fields” error

The librarian can edit existing book information by clicking on the book, the existing book information will then appear on the text boxes. Users can then make changes and then click on the “Edit” button to update the new book information into the database.

Figure 3.2.4 Book Edit Function

Any unwanted book records can be deleted by clicking on the respective book, and then proceed by clicking on the “Delete” function. That existing record will then be deleted from the database.

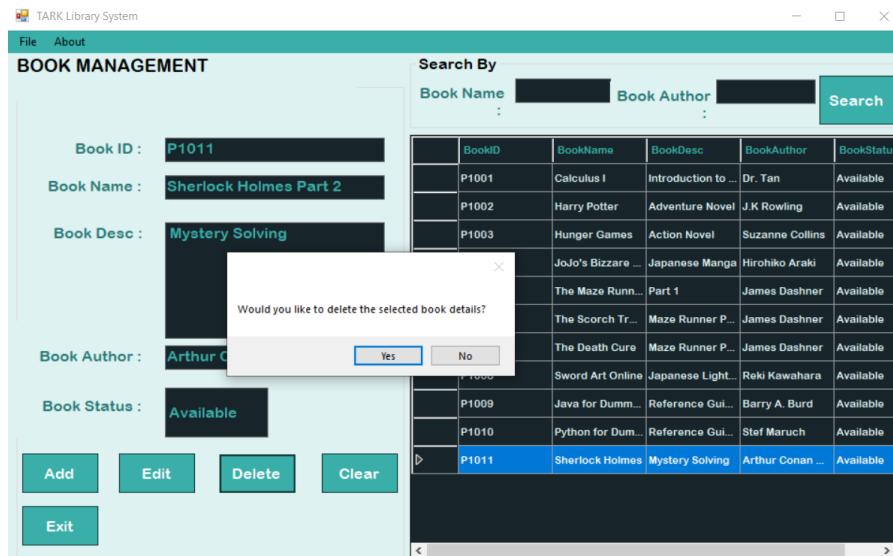


Figure 3.2.5 Book Delete Function

	BookID	BookName	BookDesc	BookAuthor	BookStatus
▷	P1001	Calculus I	Introduction to ...	Dr. Tan	Available
	P1002	Harry Potter	Adventure Novel	J.K Rowling	Available
	P1003	Hunger Games	Action Novel	Suzanne Collins	Available
	P1004	JoJo's Bizzare ...	Japanese Manga	Hirohiko Araki	Available
	P1005	The Maze Runn...	Part 1	James Dashner	Available
	P1006	The Scorch Tr...	Maze Runner P...	James Dashner	Available
	P1007	The Death Cure	Maze Runner P...	James Dashner	Available
	P1008	Sword Art Online	Japanese Light...	Reki Kawahara	Available
	P1009	Java for Dumm...	Reference Gui...	Barry A. Burd	Available
	P1010	Python for Dum...	Reference Gui...	Stef Maruch	Available
	P1011	Sherlock Holmes	Mystery Solving	Arthur Conan Doyle	Available

Figure 3.2.6 After delete records

The “clear” button allows the librarian to clear all the records on the text boxes.

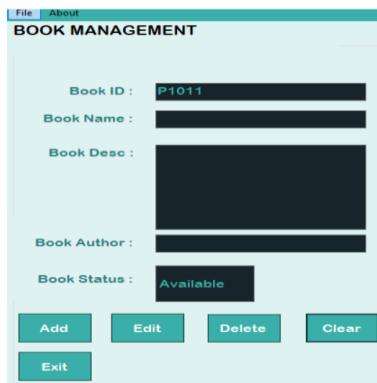


Figure 3.2.7 Clear function

The librarian can search for respective books in the database by typing in the book name and book author in the text box, and then click the “Search” button.



Figure 3.2.8 Book Searching

Report Printing

The admin can print the report by selecting File> Print Report on the Book Management form. All the existing records of books in the database will then appear on the report and ready to be printed.



Figure 3.2.9 Book Management Report printing page

TARK
Summary Books Report
=====
Prepare by TARK Library System
Date : 03/12/2020
Page : 1

Book ID	Book Name	Book Author	Book Status
P1001	Calculus I	Dr. Tan	Available
P1002	Harry Potter	J.K Rowling	Available
P1003	Hunger Games	Suzanne Collins	Available
P1004	JoJo's Bizzare Adver...	Hirohiko Araki	Available
P1005	The Maze Runner	James Dashner	Available
P1006	The Scorch Trails	James Dashner	Available
P1007	The Death Cure	James Dashner	Available
P1008	Sword Art Online	Reki Kawahara	Available
P1009	Java for Dummies	Barry A. Burd	Available
P1010	Python for Dummies	Stef Maruch	Available

--- End ---

Figure 3.2.10 Book Management Report

Exception Handling

```

Try
    If txtBookID.Text <> "" And txtBookName.Text <> "" And txtBookDesc.Text <> "" And txtBookAuthor.Text <> "" And lblBookStatus.Text <> "" Then
        i.BookName = txtBookName.Text
        i.BookDesc = txtBookDesc.Text
        i.BookAuthor = txtBookAuthor.Text
        i.BookStatus = lblBookStatus.Text
        db.SubmitChanges()
        MessageBox.Show("Record Updated!", "Info", MessageBoxButtons.OK, MessageBoxIcon.Information)
        LoadData()
    Else
        errmsg = "Please enter all required fields"
        MessageBox.Show(errmsg, "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning)
    End If

    Catch ex As Exception
        MessageBox.Show("Please fill in the blank/ Insert wrong data type!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning)
    End Try

```

User Manual 3.3 – Lend and Return Module by Soon Yi Hong

Lend and Return

The librarian can help students to lend out books through the Lend and Return module.

LendReturnID	BookID	StudentID	LendFrom	LendTo
L1001	P1002	S1001	3/12/2020	10/12/2020
L1002	P1001	S1002	3/12/2020	8/12/2020
L1003	P1008	S1007	1/12/2020	8/12/2020
L1004	P1005	S1004	5/12/2020	13/12/2020
L1005	P1003	S1005	5/12/2020	13/12/2020
L1006	P1004	S1010	4/12/2020	17/12/2020
L1007	P1006	S1003	7/12/2020	14/12/2020
L1008	P1009	S1006	9/12/2020	17/12/2020
L1009	P1010	S1009	10/12/2020	18/12/2020
L1010	P1007	S1008	19/12/2020	26/12/2020

Figure 3.3.1 Lend and Return Page

The librarian must enter all the lending information before they can add the record into the database. When a student wants to lend a book, the librarian must fill up all the fields including select the date of the lending date and the returning date, and then proceed by pressing the “Add” button.

LendReturnID	BookID	StudentID	LendFrom	LendTo
L1001	P1002	S1001	3/12/2020	10/12/2020
L1002	P1001	S1002	3/12/2020	8/12/2020
L1003	P1008	S1007	1/12/2020	8/12/2020
L1004	P1005	S1004	5/12/2020	13/12/2020
L1005	P1003	S1005	5/12/2020	13/12/2020
L1006	P1004	S1010	4/12/2020	17/12/2020
L1007	P1006	S1003	7/12/2020	14/12/2020
L1008	P1009	S1006	9/12/2020	17/12/2020
L1009	P1010	S1009	10/12/2020	18/12/2020
L1010	P1007	S1008	19/12/2020	26/12/2020

Figure 3.3.2 Lend and Return Function

Lend And Return

Lending Information

Lend Return ID :	L1011
Book ID :	P1011
Student ID :	[REDACTED]
Lend From :	18/12/2020
Lend To :	26/12/2020
Return Date :	[REDACTED]
Overdue Date :	[REDACTED]

Add Edit Clear Exit

Search By

Student ID [REDACTED] Date [REDACTED]

LendReturnID	BookID
L1006	P1004
L1007	P1006
L1008	P1009
L1009	P1010
L1010	P1007

Error

Please enter all required fields

OK

Lend And Return

Lending Information

Lend Return ID :	L1011
Book ID :	P1001
Student ID :	ABC
Lend From :	3/12/2020
Lend To :	3/12/2020
Return Date :	[REDACTED]
Overdue Date :	[REDACTED]

Add Edit Clear Exit

Search By

Student ID [REDACTED]

LendReturnID	BookID
L1004	P1004
L1006	P1006
L1008	P1009
L1009	P1010
L1010	P1007

Info

Student Not Found!

OK

Figure 3.3.3 Lend and Return Validation

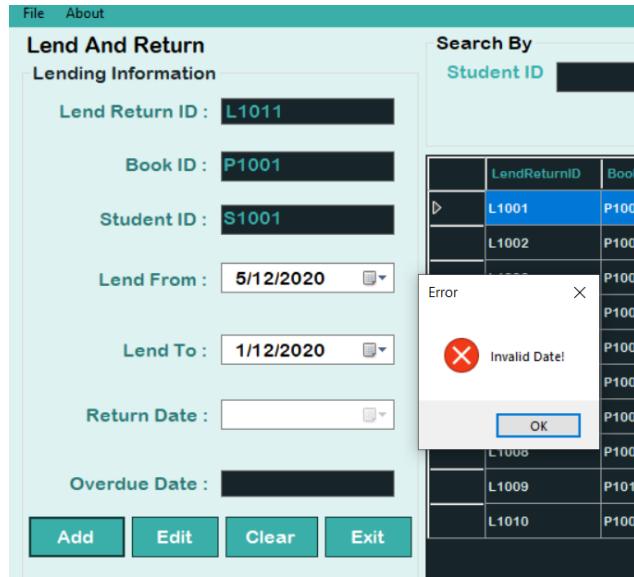


Figure 3.3.4 Lending Date Validation

When the student returns their book, the librarian can then click on the lending record in the database and then update their return date. After changing the return date, click on the “Edit” button to update into the database. The overdue date will then be showed on the label.

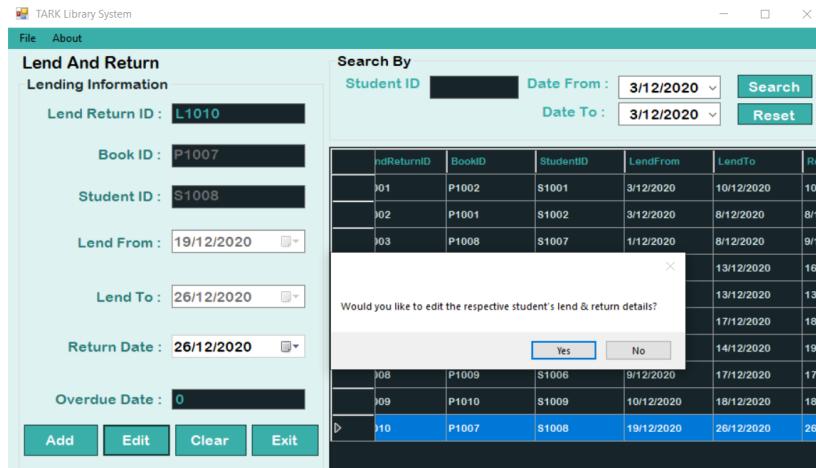


Figure 3.3.5 Lend and return Edit Function

	L1010	P1007	S1008	19/12/2020	26/12/2020
--	-------	-------	-------	------------	------------

Overdue Date : 0

Figure 3.3.6 After updating into the database

The “Clear” button will remove all characters in the text box.

Lend And Return	
Lending Information	
Lend Return ID :	<input type="text" value="L1011"/>
Book ID :	<input type="text"/>
Student ID :	<input type="text"/>
Lend From :	<input type="text" value="3/12/2020"/> <input type="button" value="▼"/>
Lend To :	<input type="text" value="3/12/2020"/> <input type="button" value="▼"/>
Return Date :	<input type="text"/> <input type="button" value="▼"/>
Overdue Date :	<input type="text"/>
<input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Clear"/> <input type="button" value="Exit"/>	

Figure 3.3.7 Clear Function

The librarian can then search lending information from the database by searching for Student ID and the dates. The “Reset” button will reset to the default which is display all the records.

Search By						
Student ID	<input type="text" value="S1004"/>	Date From :	<input type="text" value="3/12/2020"/> <input type="button" value="▼"/>	<input type="button" value="Search"/>		
		Date To :	<input type="text" value="24/12/2020"/> <input type="button" value="▼"/>	<input type="button" value="Reset"/>		
D	BookID	StudentID	LendFrom	LendTo	ReturnDate	
▷	P1005	S1004	5/12/2020	13/12/2020	16/12/2020	

Figure 3.3.8 Search Function

Search By					
Student ID	Date From :	3/12/2020	Search		
	Date To :	3/12/2020	Reset		
D	BookID	StudentID	LendFrom	LendTo	ReturnDate
▷	P1002	S1001	3/12/2020	10/12/2020	10/12/2020
	P1001	S1002	3/12/2020	8/12/2020	8/12/2020
	P1008	S1007	1/12/2020	8/12/2020	9/12/2020
	P1005	S1004	5/12/2020	13/12/2020	16/12/2020
	P1003	S1005	5/12/2020	13/12/2020	13/12/2020
	P1004	S1010	4/12/2020	17/12/2020	18/12/2020
	P1006	S1003	7/12/2020	14/12/2020	19/12/2020
	P1009	S1006	9/12/2020	17/12/2020	17/12/2020
	P1010	S1009	10/12/2020	18/12/2020	18/12/2020
	P1007	S1008	19/12/2020	26/12/2020	26/12/2020

Figure 3.3.9 Reset Button

Report Printing

The admin can then proceed to print all the lend and return report by clicking File> Print Report in the Lend and Return form.

LEND AND RETURN REPORT							
Search By		Date From :	3/12/2020	Search	Print		
		Date To :	3/12/2020	Reset			
D	LendReturnID	BookID	StudentID	LendFrom	LendTo	ReturnDate	Ov
▷	L1001	P1002	S1001	3/12/2020	10/12/2020	10/12/2020	0
	L1002	P1001	S1002	3/12/2020	8/12/2020	8/12/2020	0
	L1003	P1008	S1007	1/12/2020	8/12/2020	9/12/2020	1
	L1004	P1005	S1004	5/12/2020	13/12/2020	16/12/2020	3
	L1005	P1003	S1005	5/12/2020	13/12/2020	13/12/2020	0
	L1006	P1004	S1010	4/12/2020	17/12/2020	18/12/2020	1
	L1007	P1006	S1003	7/12/2020	14/12/2020	19/12/2020	5
	L1008	P1009	S1006	9/12/2020	17/12/2020	17/12/2020	0
	L1009	P1010	S1009	10/12/2020	18/12/2020	18/12/2020	0
	L1010	P1007	S1008	19/12/2020	26/12/2020	26/12/2020	0

Figure 3.3.10 Lend and Return report printing page



TARK

Summary Lend/Return Report**=====****Prepare by TARK Library System****Date : 03/12/2020****Page : 1**

Lend/Return ID	Book ID	Student ID	Lend From	Lend To	Return Date	Overdue D
L1001	P1002	S1001	3/12/2020	10/12/2020	10/12/2020	0
L1002	P1001	S1002	3/12/2020	8/12/2020	8/12/2020	0
L1003	P1008	S1007	1/12/2020	8/12/2020	9/12/2020	1
L1004	P1005	S1004	5/12/2020	13/12/2020	16/12/2020	3
L1005	P1003	S1005	5/12/2020	13/12/2020	13/12/2020	0
L1006	P1004	S1010	4/12/2020	17/12/2020	18/12/2020	1
L1007	P1006	S1003	7/12/2020	14/12/2020	19/12/2020	5
L1008	P1009	S1006	9/12/2020	17/12/2020	17/12/2020	0
L1009	P1010	S1009	10/12/2020	18/12/2020	18/12/2020	0
L1010	P1007	S1008	19/12/2020	26/12/2020	26/12/2020	0
--- End ---						

Figure 3.3.11 Lend and Return reportException Handling

```

Try
If lblLendReturnID.Text <> "" And txtBookID.Text <> "" And txtStudentID.Text <> "" And dateFrom.Text <> "" And dateTo.Text <> "" Then
    Dim datediff As Integer = CInt((dateTo.Value.Date - dateFrom.Value.Date).Days)
    'Dim isDuplicated As Boolean = db.Items.Any(Function(o) o.Id = i.Id)'
    Dim isDuplicated As Boolean = db.LendReturns.Any(Function(o) o.LendReturnID = lblLendReturnID.Text)
    If isDuplicated = True Then
        MessageBox.Show("ID Already Existed", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    ElseIf datediff < 0 Then
        MessageBox.Show("Invalid Date!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        Return
    Else
        Dim forBook As Book = db.Books.FirstOrDefault(Function(o) o.BookID = txtBookID.Text)
        Dim forStudent As Student = db.Students.FirstOrDefault(Function(o) o.StudentID = txtStudentID.Text)
        Dim isDuplicatedBookStatus As Boolean = db.Books.Any(Function(o) o.BookStatus = "Lending" And o.BookID = txtBookID.Text)
        Dim isDuplicatedStudentStatus As Boolean = db.Students.Any(Function(o) o.StudentStatus = "Lending" And o.StudentID = txtStudentID.Text)
        Dim isDuplicatedStudentFine As Boolean = db.Students.Any(Function(o) o.StudentFine = "Yes" And o.StudentID = txtStudentID.Text)
        If forBook Is Nothing Then
            'Console.WriteLine("Record Not Found!")
            MessageBox.Show("Book Not Found!", "Info", MessageBoxButtons.OK, MessageBoxIcon.Information)
        ElseIf forStudent Is Nothing Then
            MessageBox.Show("Student Not Found!", "Info", MessageBoxButtons.OK, MessageBoxIcon.Information)
        ElseIf isDuplicatedBookStatus = True Then
            MessageBox.Show("Book Already Lended", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        ElseIf isDuplicatedStudentStatus = True Then
            MessageBox.Show("Student Already Lending Book", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        ElseIf isDuplicatedStudentFine = True Then
            MessageBox.Show("Student Need to Pay Book Fine First!!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End If
    End If
End If

```

```
        Else
            i.LendReturnID = lblLendReturnID.Text
            i.BookID = txtBookID.Text
            i.StudentID = txtStudentID.Text
            i.LendFrom = dateFrom.Value
            i.LendTo = dateTo.Value
            i.ReturnDate = dateReturn.Value
            i.OverdueDate = lblOverdue.Text
            i.AccountID = Login.sessionAccountID
            afterBook.BookStatus = "Lending"
            afterStudent.StudentStatus = "Lending"
            db.LendReturns.InsertOnSubmit(i)
            db.SubmitChanges()
            MessageBox.Show("Successfully Added!")
            txtBookID.Text = ""
            txtStudentID.Text = ""
            lblOverdue.Text = ""
            BindData()
            IDload()
            txtBookID.Enabled = True
            txtStudentID.Enabled = True
            dateFrom.Enabled = True
            dateTo.Enabled = True

        End If
    End If
Else
    errmsg = "Please enter all required fields"
    MessageBox.Show(errormsg, "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning)
End If
Catch ex As Exception
    MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning)
End Try
```

User Manual 3.4 – Book Fine Module by Chong Jia Jun

The librarian can use the book fine module to access for the student's overdue fine records.

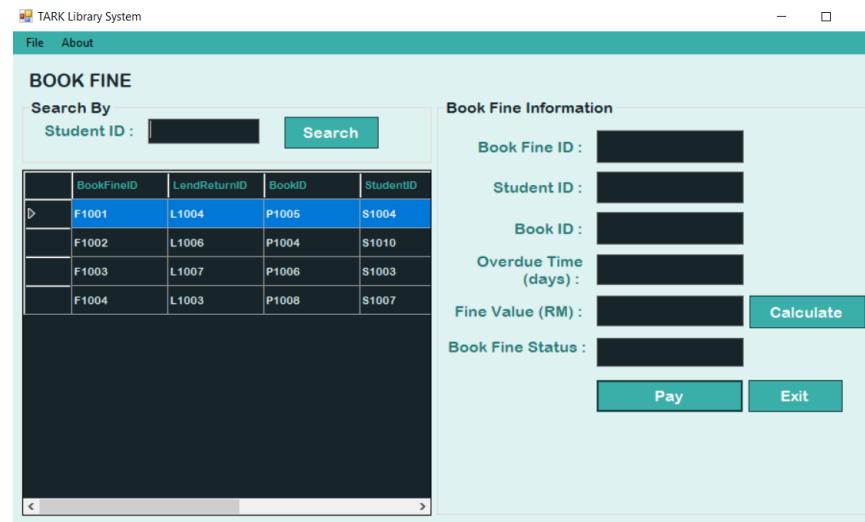


Figure 3.4.1 Book Fine Page

All of the records that have an overdue date are eligible for book fining. The lend and return information of the user that is eligible for fining will then appear. The book fine can be calculated by clicking on the respective record and then click on the “Calculate” Function.

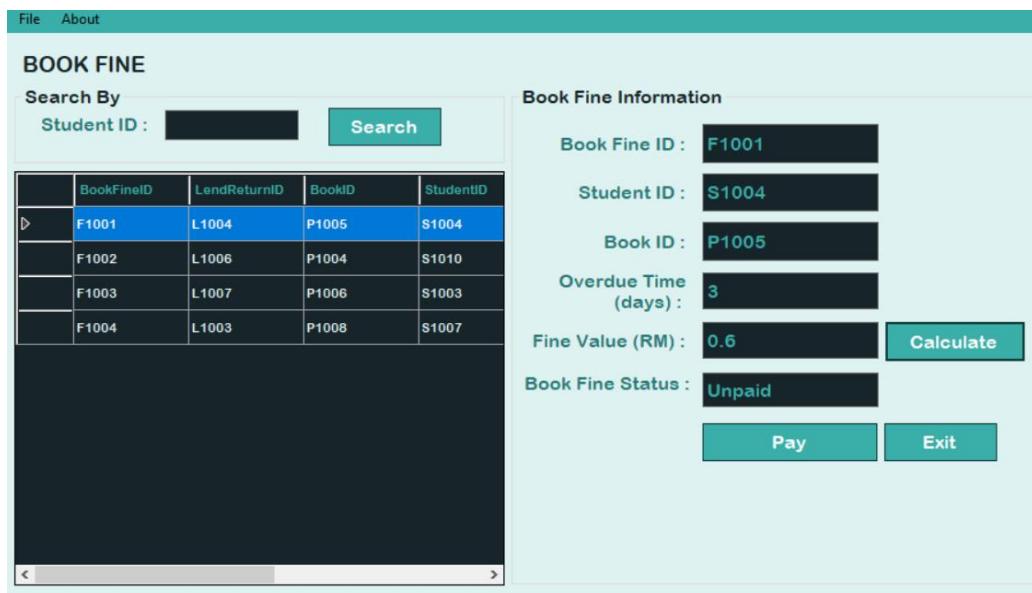


Figure 3.4.2 Book Fine Calculate Function

After the student paid for their fine, the librarian can proceed to click on “Pay” button, and it will remove the record from the book fine module, which means that the fine is settled.

	BookFineID	LendReturnID	BookID	StudentID
▷	F1002	L1006	P1004	S1010
	F1003	L1007	P1006	S1003
	F1004	L1003	P1008	S1007

Figure 3.4.3 After Paying

The librarian can search for fine records by typing in the Student ID.

BOOK FINE

Search By

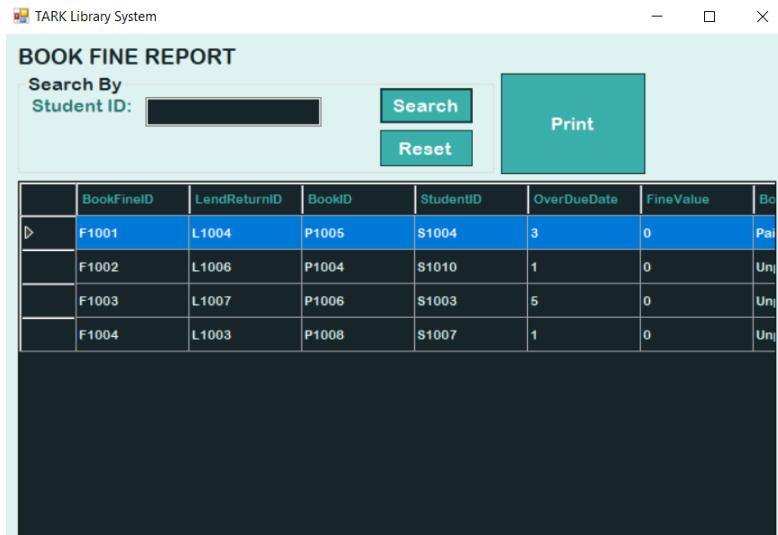
Student ID : **S1003** **Search**

	BookFineID	LendReturnID	BookID	StudentID
▷	F1003	L1007	P1006	S1003

Figure 3.4.4 Book Fine Search Function

Report Printing

The admin can print the book fine report by clicking File > Print Report in the Book Fine Module.

Figure 3.4.5 Book Fine report printing page

TARK

Summary Book Fine Report
=====

Prepare by TARK Library System
Date : 03/12/2020
Page : 1

Book Fine ID	Lend Return ID	Book ID	Student ID	OverDue Date	Fine Value	Book Fine Status
F1001	L1004	P1005	S1004	3	0	Paid
F1002	L1006	P1004	S1010	1	0	Unpaid
F1003	L1007	P1006	S1003	5	0	Unpaid
F1004	L1003	P1008	S1007	1	0	Unpaid

-- End --

Figure 3.4.6 Book Fine Report

User Manual 3.5 – Student Management Module by Chong Jia Jun

An admin account must be logged in to access the Student Management module. The student management module can be accessed by clicking on the “Student” button in the main menu.

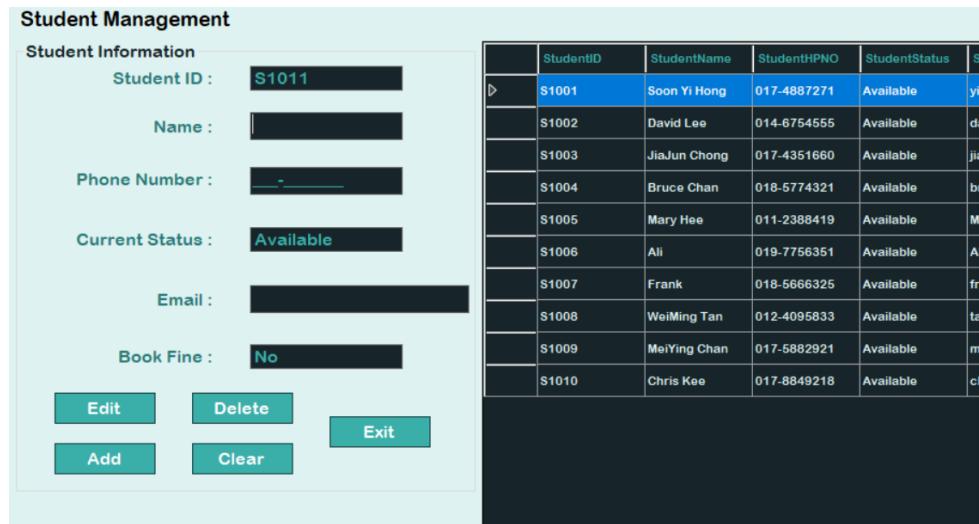


Figure 3.5.1 Student Management Menu

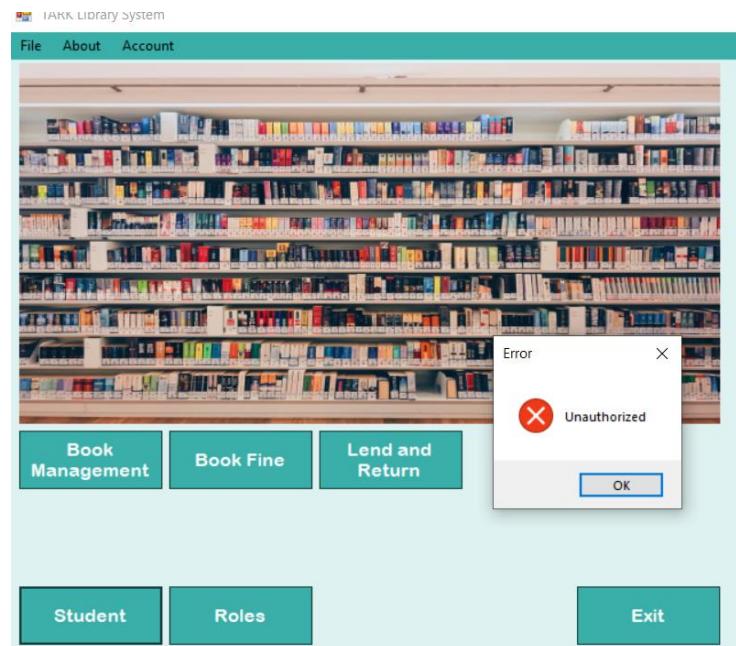


Figure 3.5.2 Unauthorized Access

When a student wants to register for a library account, the librarian can then enter the new student account information into the fields and then click on “Add” button to enter into the database.

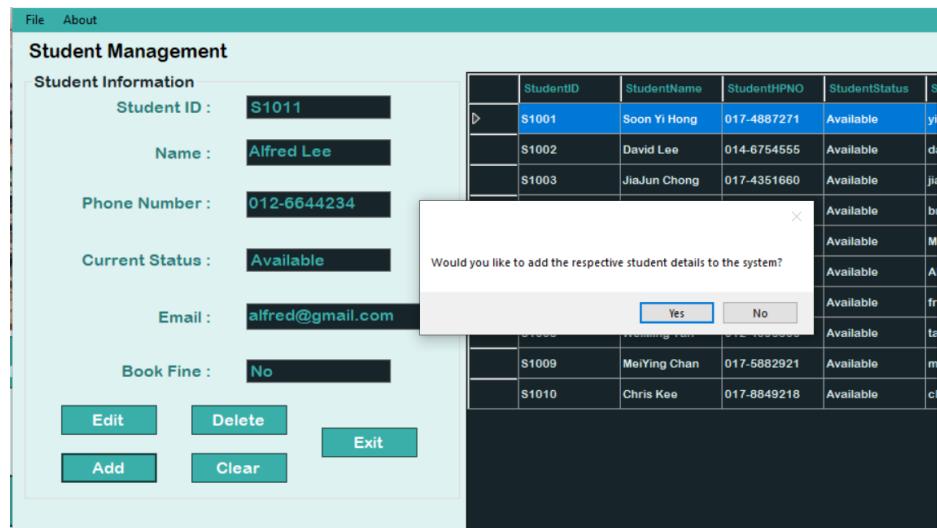
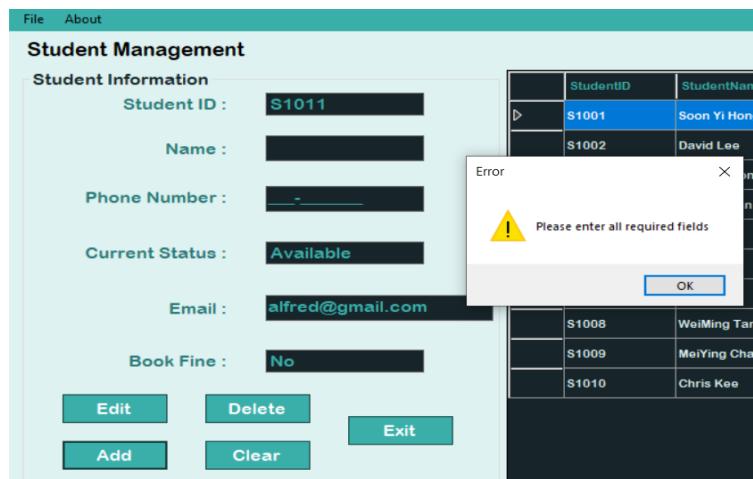
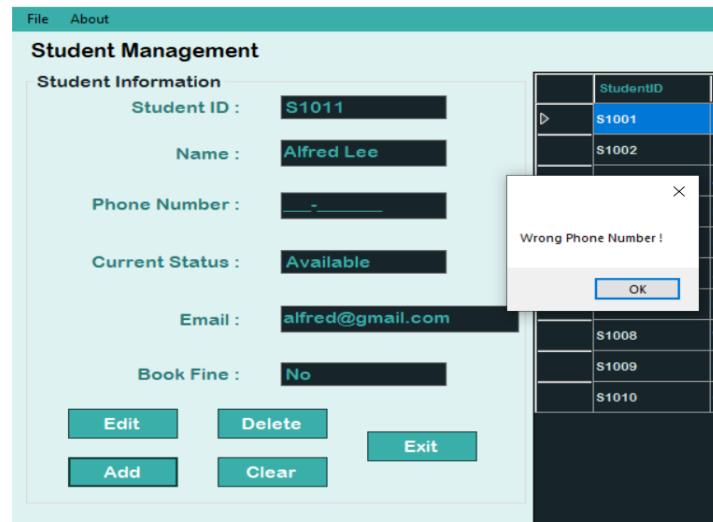
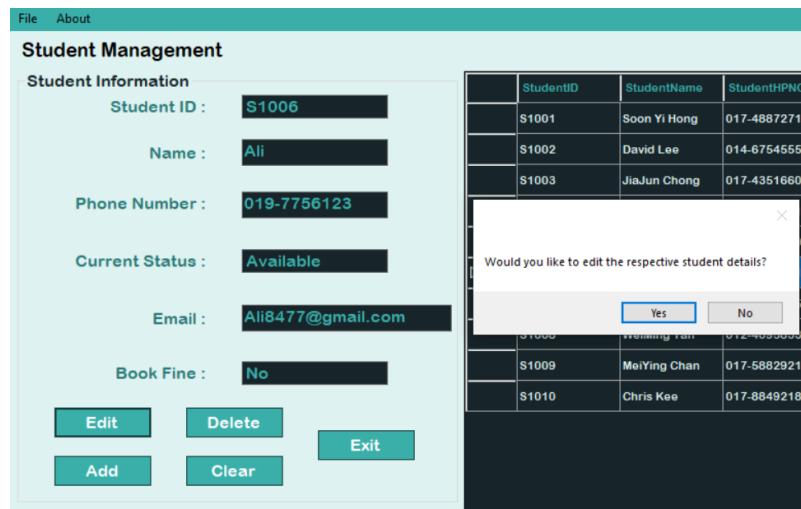


Figure 3.5.3 Student Management Add function



Figure 3.5.4 Input Validation

The librarian can then change the student's information by clicking on a student, and then change the information in the text boxes. After changing, click on the “Edit” button to update the new information into the database.

Figure 3.5.4 Student Management Edit Function

S1006	Ali	019-7756123	Available
-------	-----	-------------	-----------

Figure 3.5.5 After Edit

If the librarian wants to delete an unwanted student account information, just click on the respective student record and then click the “Delete” button to remove it from the database.

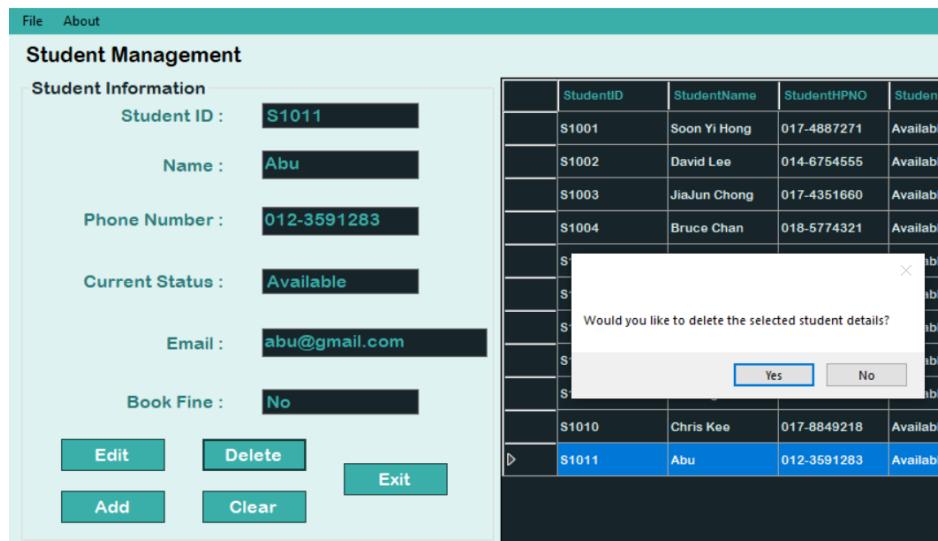


Figure 3.5.6 Student Management Delete Function

	StudentID	StudentName	StudentHPNO	StudentStatus	
▷	S1001	Soon Yi Hong	017-4887271	Available	yi
	S1002	David Lee	014-6754555	Available	da
	S1003	JiaJun Chong	017-4351660	Available	jia
	S1004	Bruce Chan	018-5774321	Available	br
	S1005	Mary Hee	011-2388419	Available	Ma
	S1006	Ali	019-7756123	Available	Al
	S1007	Frank	018-5666325	Available	fr
	S1008	WeiMing Tan	012-4095833	Available	ta
	S1009	MeiYing Chan	017-5882921	Available	me
	S1010	Chris Kee	017-8849218	Available	ch
	S1011	Abu	012-3591283	Available	ab

Figure 3.5.7 After Delete

The “clear” button will then remove all items in the text box.

The screenshot shows a window titled "Student Management". Under "Student Information", fields include "Student ID : S1011", "Name : [redacted]", "Phone Number : [redacted]", "Current Status : Available", "Email : [redacted]", and "Book Fine : No". Below these are buttons for "Edit", "Delete", "Add", "Clear", and "Exit".

Figure 3.5.8 Clear Button

Report Printing

Admin can print the student report by clicking File > Print Report in the student management module.

The screenshot shows a window titled "STUDENT REPORT". A "Print" button is at the top. Below is a table with 10 rows of student data:

	StudentID	StudentName	StudentIPNO	StudentStatus	StudentEmail	StudentFine
▷	S1001	Soon Yi Hong	017-4887271	Available	yihongsoon@y...	No
	S1002	David Lee	014-6754555	Available	david123@hot...	No
	S1003	JiaJun Chong	017-4351660	Available	jiajunchong20...	Yes
	S1004	Bruce Chan	018-5774321	Available	bruceC@yahoo...	No
	S1005	Mary Hee	011-2388419	Available	Mary948@gma...	No
	S1006	Ali	019-7756123	Available	Ali8477@gmail...	No
	S1007	Frank	018-5666325	Available	frankers@hotmail...	Yes
	S1008	WeiMing Tan	012-4095833	Available	tanweiming@h...	No
	S1009	MeiYing Chan	017-5882921	Available	mychan@yahoo...	No
	S1010	Chris Kee	017-8849218	Available	chrisK@hotmail...	Yes

Figure 3.5.9 Student Report Printing Page

	TARK			
Summary Student Report				
=====				
Prepare by TARK Library System				
Date : 03/12/2020				
Page : 1				
Student ID	Student Name	Phone Number	Student Email	Student Fine
S1001	Soon Yi Hong	017-4867271	yihongsoon@gmail.com	No
S1002	David Lee	014-6754555	david123@hotmail.com	No
S1003	JiaJun Chong	017-4351660	jiajunchong2000@hotmail.com	No
S1004	Bruce Chan	018-5774321	bruceC@yahoo.com	No
S1005	Mary Hee	011-2388419	Mary948@gmail.com	No
S1006	Ali	019-7756123	Ali8477@gmail.com	No
S1007	Frank	018-5666325	frankers@hotmail.com	Yes
S1008	Weiming Tan	012-4095833	tanweiming@hotmail.com	No
S1009	MeiYing Chan	017-5882921	mychan@yahoo.com	No
S1010	Chris Kee	017-8849218	chrisK@hotmail.com	Yes

Figure 3.5.10 Student Report

Exception Handling

```

Try
If lblStudentID.Text <> "" And txtName.Text <> "" And msktxtHPNO.Text <> "" And lblCurrentStatus.Text <> "" And txtEmail.Text <> "" And lblBookFine.Text <> "" Then

    'Dim isDuplicated As Boolean = db.Items.Any(Function(o) o.Id = i.Id)'
    Dim isDuplicated As Boolean = db.Students.Any(Function(o) o.StudentHPNO = msktxtHPNO.Text)
    Dim isDuplicatedEmail As Boolean = db.Students.Any(Function(o) o.StudentEmail = txtEmail.Text)

    If isDuplicated = True Then
        'Console.WriteLine("ID already exist!", MessageBoxIcon.Error)'
        MessageBox.Show("Phone Number Already Exist !")
    ElseIf isDuplicatedEmail = True Then
        MessageBox.Show("Email Already Exist !")
    ElseIf msktxtHPNO.MaskCompleted = False Then
        MessageBox.Show("Wrong Phone Number !")
    ElseIf IsValidEmailFormat(txtemail.Text) = False Then
        MessageBox.Show("Invalid Email Format !!")
    ElseIf IsValidNameFormat(txtName.Text) = False Then
        MessageBox.Show("Invalid name, only alphabets are allowed !!")
    Else

        i.StudentID = lblStudentID.Text
        i.StudentName = txtName.Text
        i.StudentHPNO = msktxtHPNO.Text
        i.StudentStatus = lblCurrentStatus.Text
        i.StudentEmail = txtEmail.Text
        i.StudentFine = lblBookFine.Text

        db.Students.InsertOnSubmit(i)
        db.SubmitChanges()
        MessageBox.Show("Successfully Added!")
        BindData()
        txtName.Text = ""
        txtEmail.Text = ""
        msktxtHPNO.Text = ""

    End If
End Try

```

```
        lblBookFine.Text = "No"
        lblCurrentStatus.Text = "Available"
        IDload()
    End If
Else
    errmsg = "Please enter all required fields"
    MessageBox.Show(errormsg, "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning)
End If
Catch ex As Exception
    MessageBox.Show(errormsg, "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning)
End Try
```