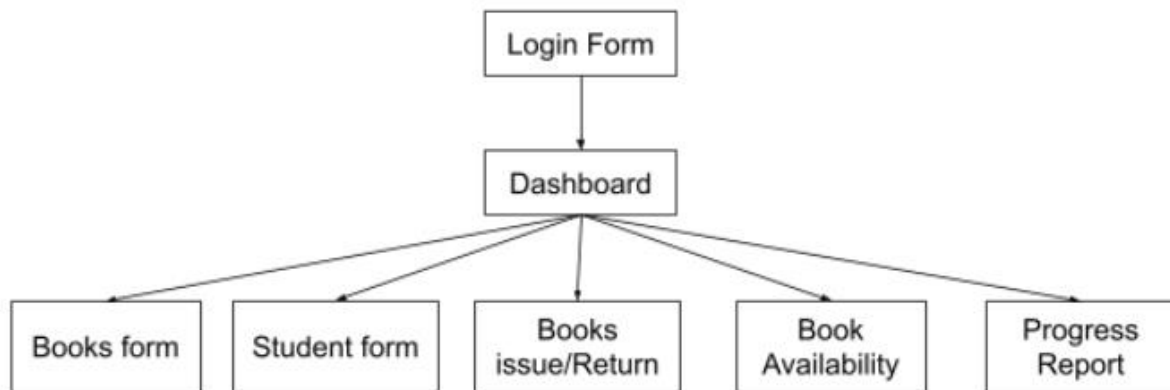


Digital Library management

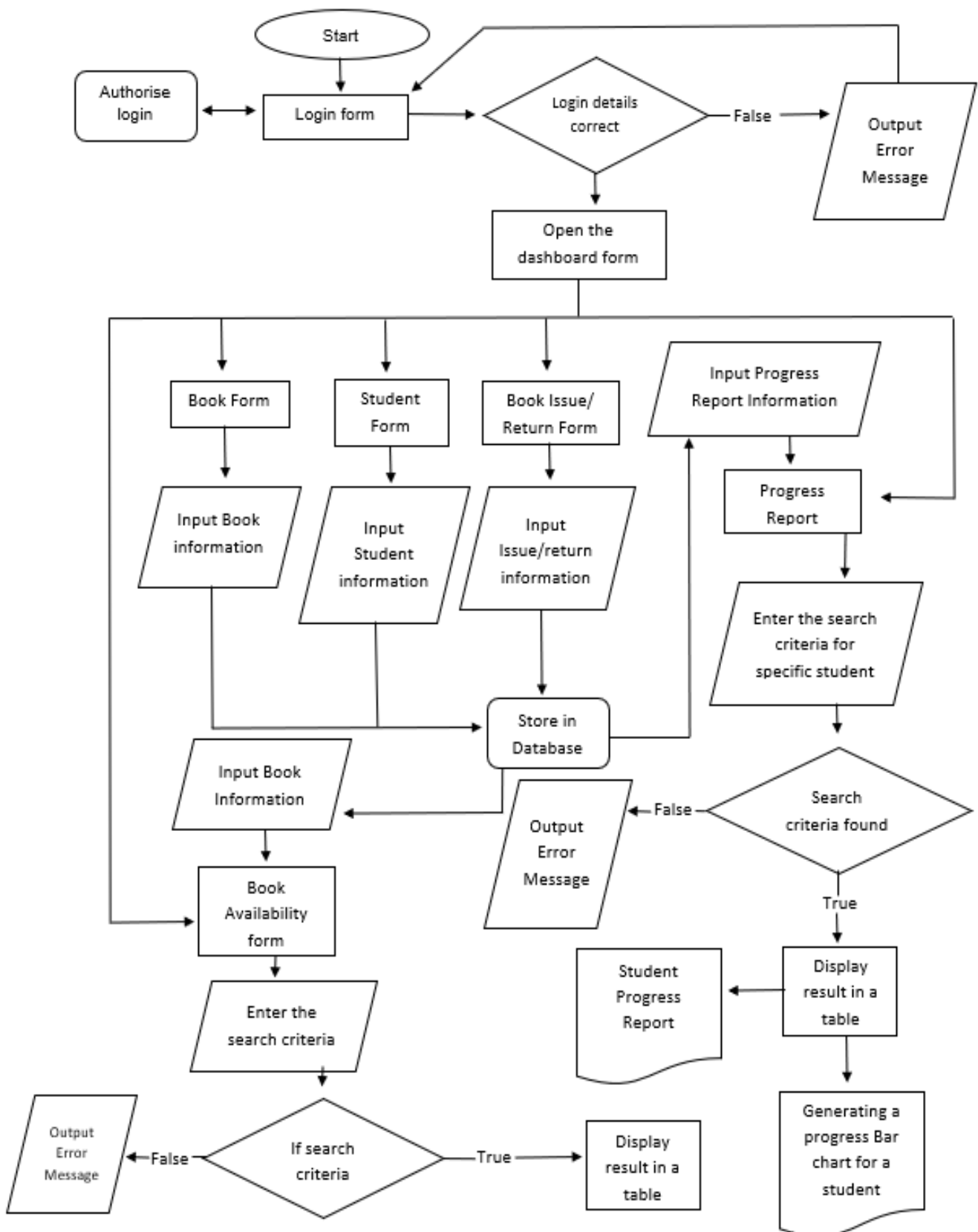
Criteria B - Design

Modular Design



System Flowchart

The System flowchart displays how the information produced by the client is stored in the database and after that the information from the database is used to output information to the other parts to output the progress report and generate a bar chart

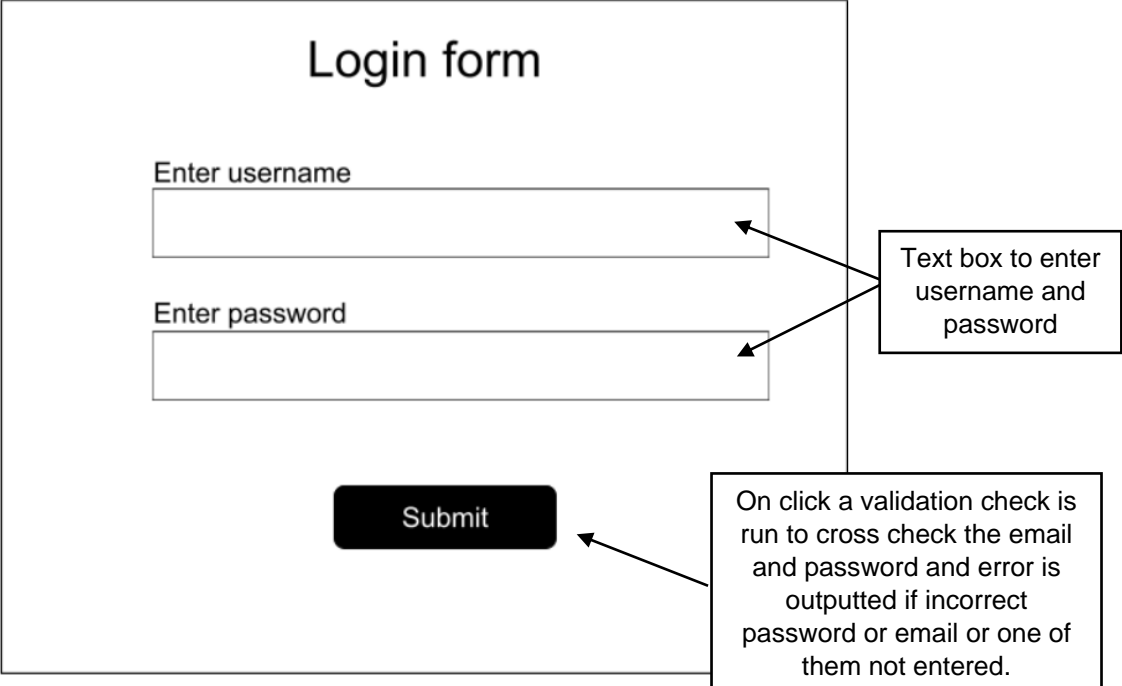


Forms Layout

These are the layout of the form which was presented to the client to get an overview of what the solution will look like and how the solution will function. Based on these layouts of the form I had received feedback from the client.

Login form:

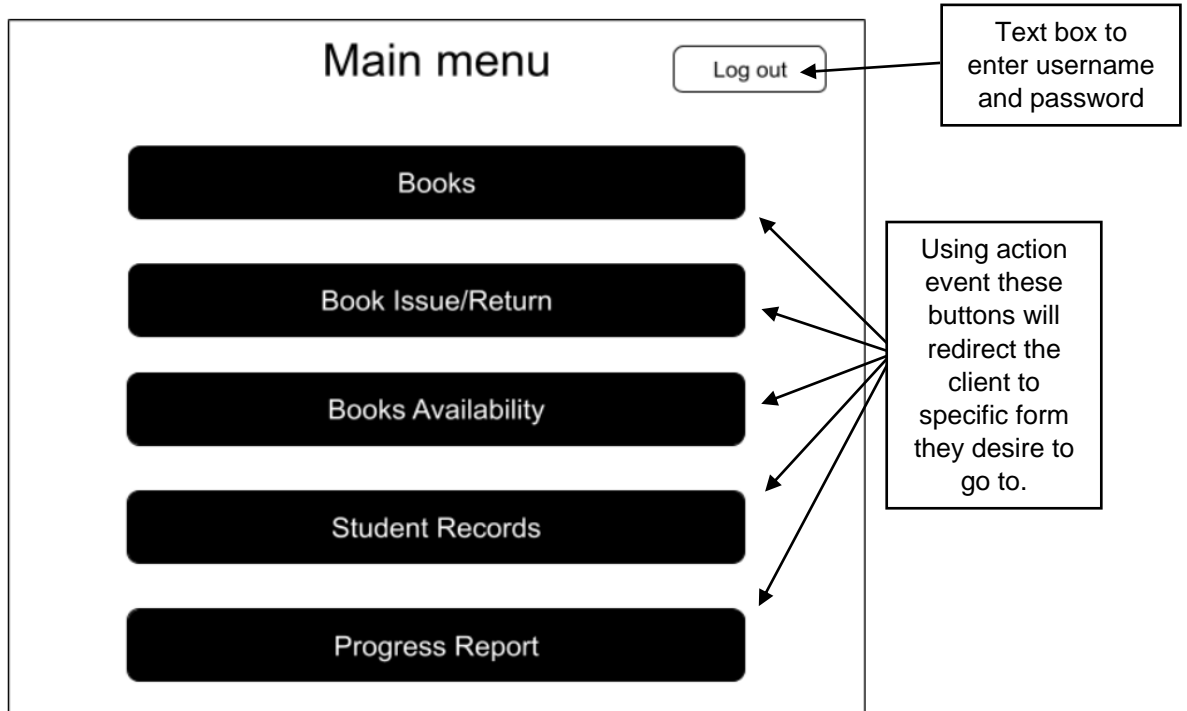
This form will be the first form when the client will open the application. The purpose of this form is to authentic the client before letting them access the software. The reason for this is to not let the student or any undesired individual have access to this solution except my client



The diagram illustrates the layout of a login form. It features a central rectangular box titled "Login form". Inside this box, there are two text input fields: the top one is labeled "Enter username" and the bottom one is labeled "Enter password". Below these fields is a dark rectangular button labeled "Submit". To the right of the form, there are two callout boxes. The first callout box, titled "Text box to enter username and password", has two arrows pointing to the "Enter username" and "Enter password" fields. The second callout box, titled "On click a validation check is run to cross check the email and password and error is outputted if incorrect password or email or one of them not entered.", has an arrow pointing to the "Submit" button.

Dashboard:

The dashboard will allow the client to easily navigate between various different forms and access the form needed by them.



Student form:

The student form allows the client to record data of all the new students who have joined the reading program. Also, it allows the client to update any student record or delete it. A table below is present for the client to go over all the data recorded stored and make it convenient for the client to just click on the table row that they want to update or delete and automatically the values will be updated in the text fields above reducing the need for the client to re-enter values.

Go back button let the client re-direct to the dashboard. This will reduce the need for the client to go back to repetitively close for changing between forms

Text box is used to enter the student ID, student class and student name

The add, update and delete buttons are respectively used to perform their task. Once the changes are done the database automatically updated.

The refresh button is used to automatically update value in tables from the database on the changes in the value.

The table depicts the values from the database records, and it also allows the user to on click of the row, automatically update values on the textboxes

Student Records

Student ID

Student Name

Student Class

Add Update Delete

Refresh

Student ID	Student Name	Student Class
01	John	04
02	Lily	02
03	Harber	03
04	Rhea	01

Books form:

The book form allows the client to add a new book in the database if a new book is bought for the library or to update any information related to book details or delete book details for the books removed from the library. The table is present to display information from database records and to make a user-friendly where the user can on click copy info from the table to the textbox.

Go Back

Text box is used to enter the Book ID, Book Name and Book Class

Text box is used to enter the Book Genre, Book Publisher and Book Author

Text box is used to enter the Book Genre, Book Publisher and Book Author

The add, update and delete buttons are respectively used to perform their task. Once the changes are done the database automatically updated.

The refresh button is used to automatically update value in tables from the database on the changes in the value.

The table depicts the values from the database records, and it also allows the user to on click of the row, automatically update values on the textboxes

Books

Book ID:

Book title:

Book Level:

Book Genre:

Book Publisher:

Book Author:

Add **Update** **Delete**

Refresh

Book ID	Book Title	Book Level	Book Genre	Book Publisher	Book Author
01	Beloved	10	Fiction	Simon	Narayan
02	1984	05	Novel	Hachette	Vikram
03	Persepolis	08	Poetry	Collins	Robert
04	Dalloway	15	Mystery	Schuster	Edgar
05	Jane Eyre	03	Article	Academic	William

Books availability form:

The books Availability form cross-checks with the database if a book is available to be issued to the student or not. To provide it with a more filtered manner - a range of criteria are given like book's level, book's genre, book's publisher, and book's author to look for the right match of books according to the demands of the client.

Go back button is used to re-direct the client to the dashboard

The client can use the dropdown option to choose from the four criteria: book's level, book's genre, book's publisher, and book's author.

Once the criteria is chosen, the client can click on the Okay button so the values can be filtered from the database related to the criteria.

Go Back

Books Availability

Choose a criteria

Okay

Search

Book ID	Book Title	Book Level	Book Genre	Book Publisher	Book Author
01	Beloved	10	Fiction	Simon	Narayan
02	1984	10	Novel	Hachette	Vikram
03	Persepolis	10	Poetry	Collins	Robert
04	Dalloway	10	Mystery	Schuster	Edgar
05	Jane Eyre	10	Article	Academic	William

The chosen criteria will be displayed here above the drop-down box.

In this textbox, the client can simply enter the initials or the value for the criteria set

On click of search button, the values are displayed on the table below.

The table depicts the values from the database after the criteria is set and the value is entered in the textbox. In this case, the criteria chosen was book level and the value entered was 10 so this is how the output looks like.

Books inventory form:

The book inventory form provides the client with a platform to issue and return books to students. The client can simply in clicks of buttons add the values in the table to issue a book. In the table below, the list of books issued is displayed. To return the book, the client can click on the row of the book issued in the table and then just click return automatically the book will be returned. The changes in the status of the book issued or returned will automatically happen in the database according to the actions performed by the client.

Go back button is used to re-direct the client to the dashboard

The Book ID and the student ID are informed of dropdown box so the client can type in the value instead of searching it.

The date of issue and date of return will have a calendar which will let the client choose the date of issue and date of return of a particular book.

The table presents the list of all the books which are currently issued and have not yet been returned.

The overdue books button lets the client view the list of books which are overdue by comparing it automatically with today's date and the table view changes to figure 1. For the client to view back the book which are currently issued they can simply click on book issued button.

The book name and student name will automatically be entered once the client chooses the values in the dropdown box above them respectively.

The issue button is used to store the book in the inventory database where all the books are issued. Also, before the book is issued a search and sorting algorithm will run to check if the chosen student ID has an overdue book or not. If there is overdue book with that student ID, then a error message will pop up restricting the client to issue a new book. Once the client clicks on return button then the books status is changed to "N" issued meaning not issued.

The title will also automatically change on overdue button click

To return to see the currently books issued click on the book issued button

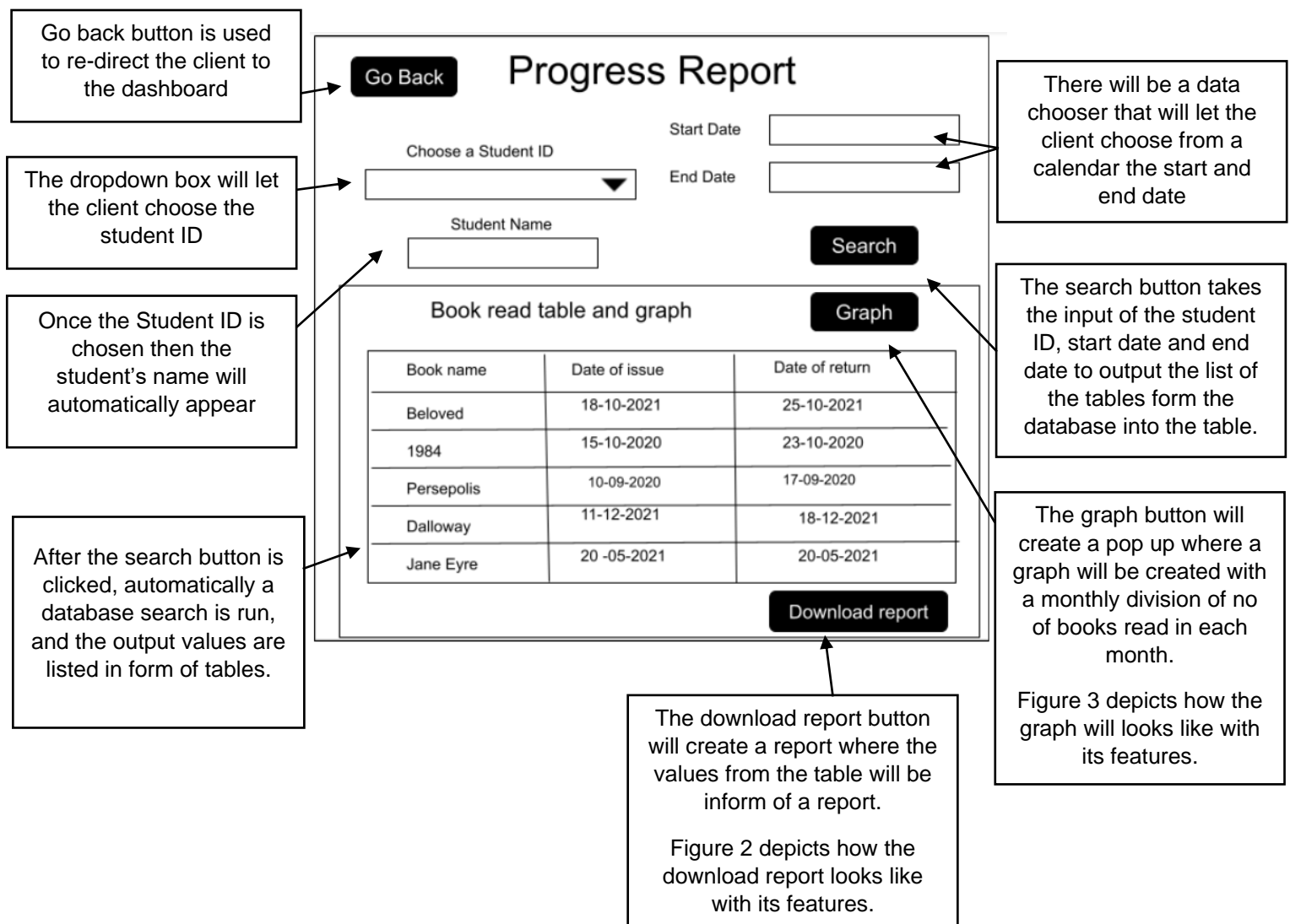
After clicking the Overdue button, the table view will change like this

Books currently overdue					
Student ID	Book ID	Student name	Book name	Date of issue	Date of Return
02	20	Miley	Beloved	01/03/2020	19/03/2020
03	40	Emma	Hamlet	19/03/2020	21/04/2020
05	11	Lucas	Sherlock home	11/06/2020	21/02/2020
08	33	Jimmy	Catch - 22	20/05/2020	12/03/2020

Figure 1

Progress Report form:

The progress report form will let the client perform analysis based on the performance of each student over a period. The client can simply add the student id with the start and end date. Automatically the list of the books with the details of the books for specific students which that student has read will be outputted in form of a table. Once the information is loaded in the table it will also be added into a temporary database where all the records will be stored. The purpose of this is to assist the program in automatically generating a table monthly to see the progress of the students over the chosen time frame by the client. The client can even convert the table in form of a report which can be downloaded by the client for keeping a personal record or sharing it with the parents.



Report of the Progress Report form:

The report depicts all the books the student has read over some time. The information is similar to the table in the progress report, however, the purpose of this is to present the same information with an option for the client to download it or print it or zoom in or zoom out.

The save button lets the client download the report in form of pdf. The print button will let the client print the report directly.

The zoom in and zoom out icon lets the client zoom into the report or zoom out according to the need of their device size.

The list of book details is sorted in form of ascending order according to the name and then according to the date of issue.

Progress Report of Student

List of books read by student

Book Name	Date of issue	Date of return
Anna Karenina	01/07/2021	15/07/2021
To kill the mockingbird	11/07/2021	13/08/2021
The great gatsby	11/08/2021	11/09/2021
Passage to India	14/08/2021	10/09/2021
Invisible man	29/08/2021	11/09/2021

Figure 2

Graph of the Progress Report form:

The Graph will appear once the client clicks on the graph button in the progress report form. The graph will appear as a pop-up. The graph will be outputted in form of a bar chart to make it easier for the client to analyse the progress of a specific student.

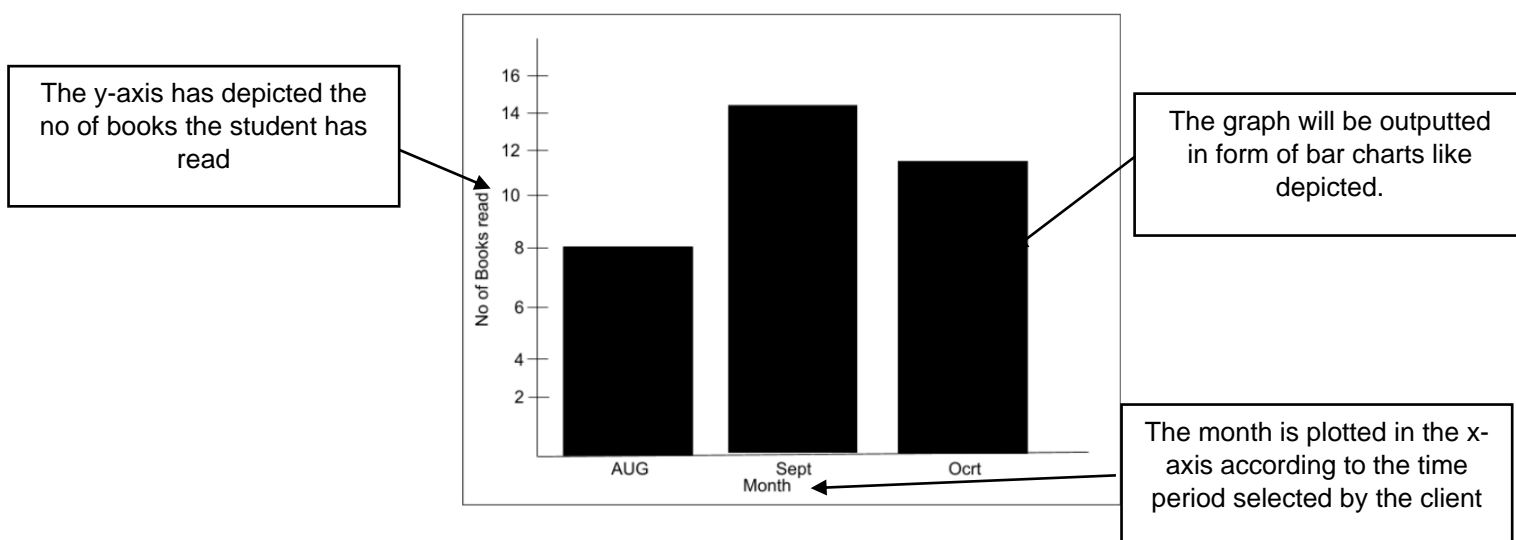


Figure 3

Database Tables

Student Table - This table is used to store the data of the students who are enrolled in the school and this table keeps a record of all the students that are part of the library program.

Attribute	Type	Primary key	Description
ID	Integer	Yes	A unique key assigned to every student to identify them uniquely amongst other participants
Name	Characters	No	The name of the student will be stored
Class	Integer	No	The grade will be stored to keep a record of which student belongs to which class

Book table - This table is used to store information about the books the library currently has bought for its usage and the books which are available and not available in the library

Attribute	Type	Primary key	Description
ID	Integer	Yes	This is a unique ID assigned to each book to distinguish them from the other books.
name	Characters	No	The name of the book is stored in this attribute
Level	Integer	No	The level of the book is stored in this attribute
Genre	Characters	No	The genre of the book is stored in this attribute
Publisher	Characters	No	The publisher of the book is stored in this attribute
Author	Characters	No	The author of the book is stored in this attribute
Issued	Characters	No	This takes to values either 'Y' or 'N' to decide whether the book in the library is issued or available to get issued

Inventory table - This table is used to keep a record of all the books assigned to each student every time they issue a book or return a book from the library.

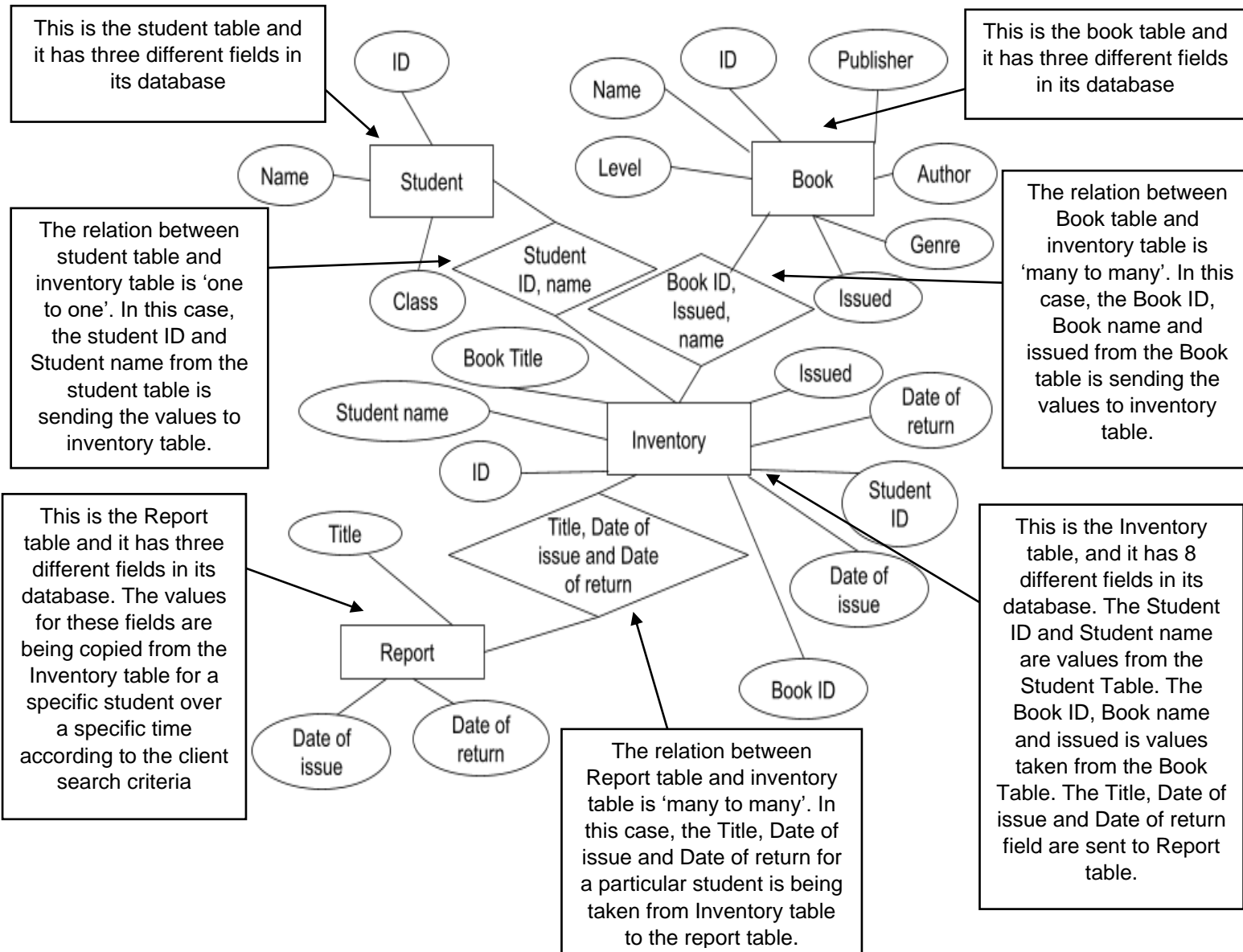
Attribute	Type	Primary key	Description
ID	Integer	Yes	This is used to uniquely identify each issue or return that takes place
IDS	Integer	No	This is the value taken from the ID of the book table and uniquely identifies the book issued
IDB	Integer	No	This is the value taken from the ID of the student table and uniquely identifies the student to whom the book is issued
title	Characters	No	This is to keep the record of the name of the book issued or returned by which student
name	Characters	No	This is to keep the record of the name of the student to whom a book was issued or returned
Datei	Date	No	This is used to keep the record of the day the book is issued
Dater	Date	No	This is used to keep the record of the possible day the book will be returned
Issued	Characters	No	This takes to values either 'Y' or 'N' to decide whether the book in the library is issued or available to get issued

Report - This table takes value from the inventory table to store records into the report table depending on the values set by the client to look for a specific student. Based on that, all the books issued or returned by the students will be recorded in this table to be displayed in form of a report.

Attribute	Type	Primary key	Description
Title	Characters	No	This is to keep the record of the name of the book issued or returned by the chosen student by the client search
Datei	Characters	No	This is the date on which the book was issued to the student
Dater	Characters	No	This is the date on which the book was returned by the student

Entity Relation diagram

Entity Relation diagram is used to depict how the database fields and values are related to different java forms. This depicts the relationship between the book, student, report and Inventory database tables.



Bubble sort

The bubble sort is used to sort an array in ascending or descending order. In this case, an array is created where the list of all students ID is added who have 1 or more books overdue. Bubble sort is used to sort the list of student ID in ascending order. This will allow the array to be used for Binary search as mentioned below. Also, we needed to sort the array as binary search needs a sorted array.

OverdueArr[] = 0

Temp = 0

For l -> 1 to length.overdueArr

For m = 1 to l < length.overdueArr

IF OverdueArr[l] > OverdueArr[m]

Then

Temp = OverdueArr[l]

OverdueArr[m] = OverdueArr[l]

OverdueArr[m] = Temp

End If

Next

Next

Binary search

On the array with the list of student ID whose books are overdue is sorted in ascending order. Then the binary search is used in the books inventory form. So, when all the information is added by the client in the books inventory form and the client clicks on the issue. A binary search is run and uses the student ID selected by the client to compare with the array of student ID whose book is overdue. If there is a match then the client is informed that this student has an overdue book and the book is not issued to that student.

Found = -1

StudentID = 0

First = 0

Mid = 0

Last = OverdueArr.length - 1

While last > first and found = -1

Do Mid = first+Last/2

If OverdueArr[mid] < Student_ID

Then First = mid+1

Else If Student_ID < Overdue[mid]

Then Last = mid-1

Else Found = Found + 1

End IF

End IF

End While

UML Class Diagram - Unified Modelling Language (UML) Class diagram is used to describe the structure of classes used in a program as well as the relationship between the objects used in the library management software.

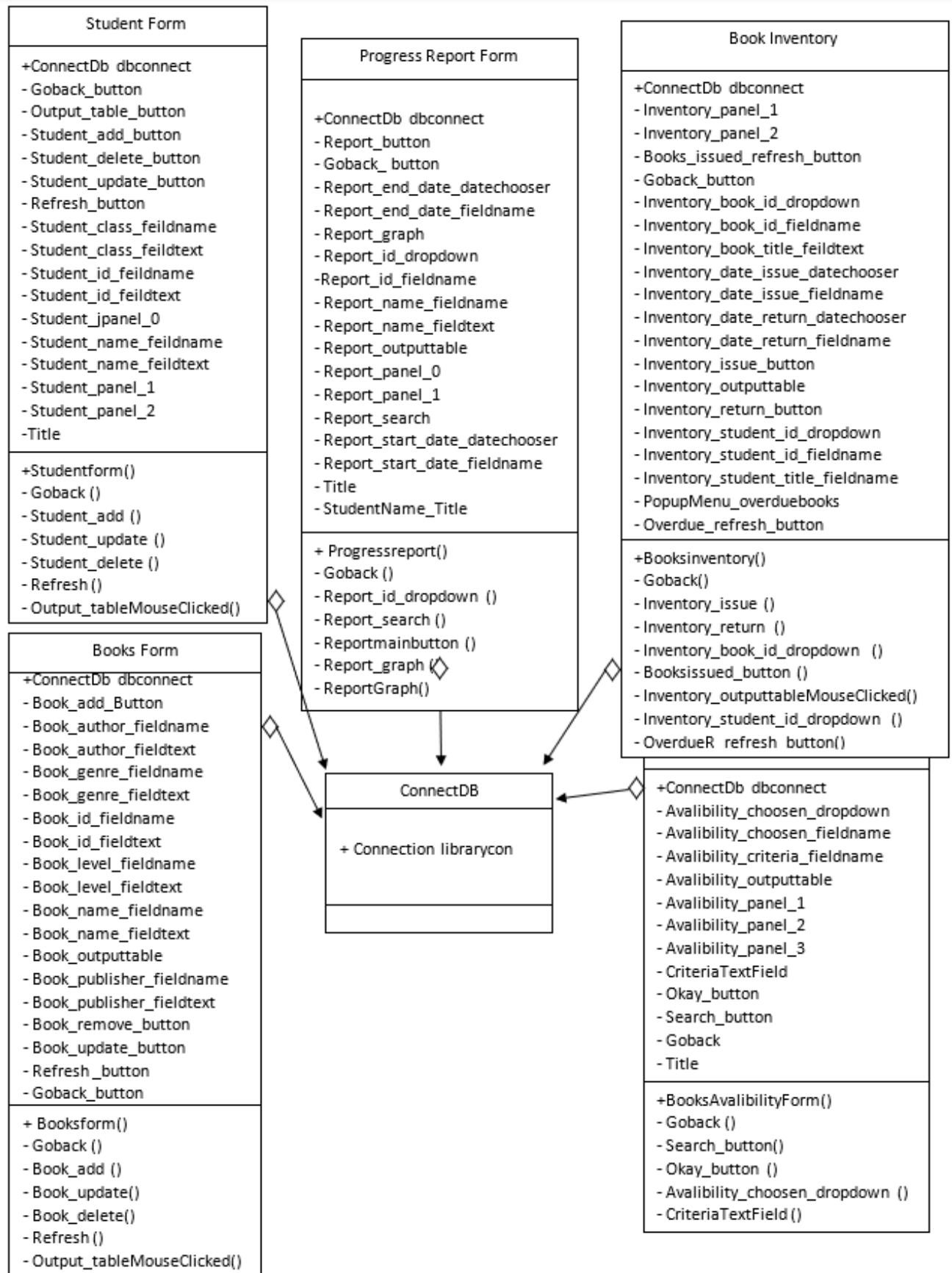


Chart
+ Chart (Map<Integer, String> graph) - CategoryDataset createDataset(Map<Integer, String> graph)

Login Form
- LoginButton - UsernameTextFeild - UsernameLabel - Inventory_login_button - Login_password_field - Login_password_fieldtext - Login_username_field - Login_username_fieldtext - Title
+ Loginform() - Login_button ()

Dashboard Form
- Booksform_button - BooksAvalibilityform_button - BookInventoryform _button - Logout_button - ProgressReportform_button - Studentform_button - Title
+ Dashboardform() - Booksform_button - BookInventoryform _button() - BooksAvalibilityform_button () - Studentform_button () - ProgressReportform_button () - Logout ()

Test plan

Test Type	Nature of test with	Example
The program starts and switch on. The client will input the appropriate username and password	Enter the username or password incorrect then the login fails, and an error message will appear.	In the login form, incorrect credentials were entered and a warning about entering wrong credentials is generated.
In the books and student forms, seeing if the data can be added/updated/deleted	Sample data will be added into the Textboxes to check if the added button functions. Also, random data such as with the wrong format to test if it accepts it or not. For update/delete we will see if clicking on the specific row where we want to update/delete automatically takes in data and update/delete all the info.	<p>In the book form, sample student details entered were student name "Arvind", student ID – "5" and student grade – "4" and add button is clicked.</p> <p>When the new student details were entered successfully and are visible in the table at bottom of the form.</p> <p>To check the functioning of update/delete a pre-entered info of student name lily is chosen and onclick of the row colour turns blue and the information gets copied in the textbox above.</p> <p>To delete just click on the delete button and change any information and click on the update information. This student information is deleted.</p>
In the book Availability form, choose the criteria and then enter the keywords to locate the availability of the desired book.	Choosing different criteria and seeing if, on the right-hand box, the criteria is visible at the top. Once it is visible, enter keywords related to the criteria to see if that list of books for example based on a specific genre and which all books are based on that genre are available. Also, sample data that is used is then cross-checked if the searched data gives the desirable data or not.	<p>In the book availability form, on choosing the criteria this is how it will look when the criteria are chosen and, for instance, the level is chosen.</p> <p>Once the "Okay" button is clicked then the level appears in the next panel, and we choose the level as "10" in this case.</p> <p>Once the search button is clicked so all the books with level 10 can be seen.</p>

<p>In the Book issue/return form, inputting appropriate information to issue a book and correct row click for self-input of the book to return.</p>	<p>Sample data will be entered and then the issue button will be clicked to check if the book is issued or not. Also, an erroneous form of information is entered to rule out the possibility that the system will accept the undesired form of information. To test the book return, the book which should be returned will be chosen from the list of books issued. Cross-check if the correct information from the table is self-inputted into the textbox and the book is successfully returned. Also, the book should no longer appear in the list of books below.</p>	<p>In the book inventory (book issue/return) form, book ID 1 is issued to student ID 2. Once they are chosen automatically the Book title and the student's name are loaded. Then the Date of issue and date of return is entered by the client.</p> <p>Once the client clicks on the issue button, the book details are issued and can be viewed by clicking on the book issued button.</p> <p>To return the book, the client as shown below can do it by clicking on the row so the information loads in. For instance, the student ID 2 to which book ID 1 is issued.</p> <p>On click of the return button and then on click on the book issued button once again to refresh we can see the book has been returned and has been removed from the table.</p>
<p>In the book issue/return form, on click, the table below values changes from overdue books to book issued list and visa-versa.</p>	<p>First, sample data will be inputted, and the list of books issued will be added. A few of the books from the sample data will be added to be depicted as an overdue list. Once this is done then with onclick of the button we can see if the table below changes, and the sample data matches. Also, we could cross-check with the sample data if the books listed as issue and overdue are correct or not?</p>	<p>On click of the books issued button, the list of all the books currently issued by clicking on the books issued.</p> <p>The client can view the list of overdue books by clicking on the overdue books button.</p>
<p>In the book issue/return form, on click of the issue button, an automatic verification will take place</p>	<p>A sample data of books issued and some of them will be overdue. Then a check will run crosschecking if an overdue book is there or not. We will run the check for a specific student who has an overdue book. If a warning message</p>	<p>In the book inventory (book issue/return) form, a book is already issued as seen with the student ID 1.</p> <p>If the same student Id is issued another book, then that book is</p>

checking if a book is overdue to a particular student and will restrict the book from being issued.	comes then the automated verification system is working correctly.	not issued, and a warning is generated.
In the progress report form, appropriate data will be inputted to search for all the books issued to the student over a period of time.	A sample data will be used for a particular student and the student ID will be inputted with the start and end date. First parameter will be to cross-check with the sample data that after inputting the Student ID does the correct student's name reflect or not. Then On click of the search button, the values in the table will be outputted. Then we will need to verify if the list of books issued over a certain time period matches the sample data used.	<p>In the progress report form, to test the functionality of this form the following information is entered. The chosen student Id is 1 and respectively the student's name loads in automatically. Also, the time period chosen is Jan 1st 2021 to 31st Aug 2021.</p> <p>Once the search button is clicked, automatically the list of the books this student read in this time period is visible in the table at the bottom of the form.</p>
In the progress report form, click the graph button automatically a bar graph should be generated showing the students' progress. Similarly, on click of the report button, a report should be generated letting the client view all the books issued.	Sample information will be inputted for a certain student, and this would result in values being generated and visible in the table. Once, the graph button is clicked an automatic graph should be generated and all the values from the sample should reflect on the graph. Similarly, on click of the report button, an automatically generated report should be visible and the values from the table/sample data should match with the values visible in the report.	<p>Based on student ID 2, from Jan 1st, 2021 to Feb 10 2020 a bar chart is automatically generated.</p> <p>Based on Student ID 1, the automatic progress report is generated on the list of the books that were read by student ID 1 from Jan 1st 2021 to April 31st 2021.</p>