s

Author: -

Husnain Awan

h

Library Management System

**Contents**

[**Introduction** 1](#_Toc83192228)

[**Windows Forms** 1](#_Toc83192229)

[**SQL Server** 2](#_Toc83192230)

[**Basic Working** 2](#_Toc83192231)

[**Classes** 2](#_Toc83192232)

[ **Books** 3](#_Toc83192233)

[ **Library** 3](#_Toc83192234)

[ **Security** 3](#_Toc83192235)

[ **Students** 3](#_Toc83192236)

[ **Students\_Security** 3](#_Toc83192237)

[**Forms** 4](#_Toc83192238)

[ **Login Form** 4](#_Toc83192239)

[ **Admin View Form** 5](#_Toc83192240)

[ **Add New Books Form** 5](#_Toc83192241)

[ **Add Existing Books Form** 6](#_Toc83192242)

[ **Display Books Form** 6](#_Toc83192243)

[ **Search Books Form** 7](#_Toc83192244)

[ **Add Admin User Form** 8](#_Toc83192245)

[ **Add Student Form** 9](#_Toc83192246)

[ **Display Student Form** 9](#_Toc83192247)

[ **Search Student Form** 10](#_Toc83192248)

[ **Student Profile Form** 11](#_Toc83192249)

[ **Issue Book Form** 12](#_Toc83192250)

[ **Return Book Form** 13](#_Toc83192251)

[ **Student View Form** 14](#_Toc83192252)

[**Comments** 15](#_Toc83192253)

**Library Management System**

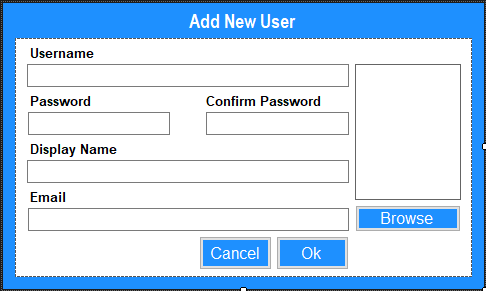
# **Introduction: -**

The project is related to the library management system using .NET Framework (WinForms) and C# with SQL Server data base. This is a complete software which can manage a library. The project is basically to test the skills of Object-Oriented Programming (OOP) in a software. The software includes following basic features: -

* Separate Interface for Staff and Other
* Encrypted Data base for Data Protection
* Separate Accounts for users
* Automatic Email Alerts
* Easy and user-friendly UI
* Add, Remove, Edit Books
* Issue, Return Books
* Restriction on Books or Users
* Automatic late fee calculator
* Separate User Profiles with Images
* Excellent Data Management
* Password Protected Profiles

# **Windows Forms: -**

Windows Forms is a Framework that lets you to generate GUIs (Graphical User Interface) with a user-friendly environment and without much of coding. The framework officially belonged to Microsoft and one can use WinForms on Visual Studio by Microsoft easily with C# programming language. C# is an Object-Oriented Programming language presented by Microsoft in .NET Framework. WinForms provides a very user-friendly environment to create an excellent GUI using simple features and commands. Here is an example of GUI made using WinForms.



**Figure 1: GUI Created by WinForms**

Thus, WinForms is an excellent choice for development of a good software using C# or C++. The basic purpose of using WinForms is to get rid of the coding while making a simple GUI.

# **SQL Server: -**

The SQL Server is a data base which we can use to store an manage programs data very accurately. The SQL Server use SQL based queries to manage the data stored in the specific data base. In this project SQL Server is used to create tables and manage the different types of data i.e. Accounts, Books and Users etc.

The tables can be used in the form of excel files but excel files are not used because data management in excel or Comma Separated Files is tough than that of SQL Server Data management thus for an easy approach SQL Server is a best option as a data base. Another property of the SQL Server is Encryption, tables can be encrypted and decrypted within the software and can’t be accessed outside the system.

# **Basic Working: -**

The project consists on several forms created by WinForms for several purposes for example a form for login page, for book entry or editing, for book issue or for user profile etc. These different forms are inherited to a main form which will be displayed after being logged into the program.

Now there are two different main forms one form the users and one form admins definitely admins will have more control over the management and user can only issue, return a book or see his/her profile which will display the books issued to him/her and other information.

First of all, a login form appears from where a domain will be selected “student” or “admin” then user should enter his login information and log into the software, if student was chosen Registration Number and CNIC should be entered and in case of admin the login parameters are username and password for the admin profile.

After logging into the software one can perform different tasks according to his/her domain. The following shows the restrictions for the normal user as compared to the admin.

|  |  |  |
| --- | --- | --- |
| **Operation** | **Admin** | **User** |
| Add, Edit, Delete Books | Yes | No |
| Add, Edit, Delete User | Yes | No |
| Search Book | Yes | Yes |
| Add, Edit, Delete Student | Yes | No |
| Search Student | Yes | No |
| Issue Book | Yes | Yes (Only for him/herself) |
| Return Book | Yes | Yes (Only for him/herself) |

# **Classes: -**

In this text the classes used in the project are explained briefly. The text explains how classes works in the project and why they are important here in this system. There is total 5 classes in the project and those are: -

* Books
* Library
* Security
* Students
* Student\_Security

All of these classes have their own purpose and works according to their requirement. There is an excellent use of concepts of OOPs in these classes all of these are explained below: -

## **Books: -**

The class mainly deals with the data of the books and the data travel between different forms. It manages the data entry checking, entering and retrieving from the table of data base. It manages how data is transferred from a form to other or from user to the data base. It sets different variables of the data base and saves data to it.

When a user wants to issue the book or return a book or to search books the class Books return the id, name and other parameter of the book and manages the data for user and other forms.

## **Library: -**

Library class manages the name of the library, logo of the library and other features i.e., quantity of books per rack. This class returns its parameter to the forms and other classes to process data and display library information on the system.

It shows the name and the logo of the library on the main screen while logging into the software and manages how many books can be assigned to a rack and quantity of books left for a rack.

## **Security: -**

It is an abstract class and polymorphism will be used to inherit another class from it explained in below text. It manages the security of the software it basically used to decrypted the encrypted columns of the data base named BlackBox which is used to store usernames and passwords of the admins. The encrypted data is decrypted by the class and then compared with the username and password entered by user while logging in thus on its basis results are made and user logged into the software if parameters are correct otherwise it shows the message according to the results.

## **Students: -**

The class mainly deals with the data of the students and the data travel between different forms. It manages the data entry checking, entering and retrieving from the table of data base. It manages how data is transferred from a form to other or from user to the data base. It sets different variables of the data base and saves data to it.

## **Students\_Security: -**

It is an inherited from the Security class and it delas with the student security it decrypts the student data and compare data while logging into the software. If a use wants to log into the software as a student, he/she should enter Registration number and CNIC which will be compared via this class without interference of other classes for security reasons.

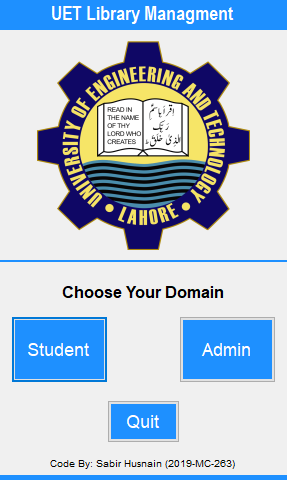
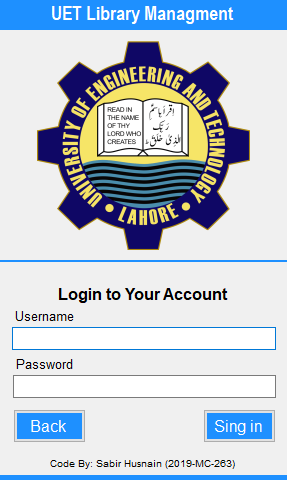
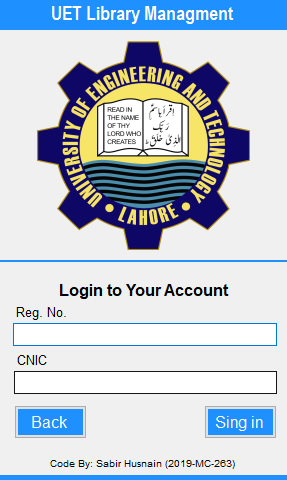
# **Forms: -**

The following text explains the forms used in the project. There are different forms used here for different purposes and for different domains. In the below text brief description and their UI is displayed. There is total 14 forms in the project and those are: -

* Login Form
* Admin View Form
* Student View Form
* Add New Books Form
* Add Existing Books Form
* Display Book Form
* Search Book Form
* Add Admin User Form
* Add Student Form
* Display Student Form
* Student Profile Form
* Search Student Form
* Issue Book Form
* Return Book Form

## **Login Form: -**

When someone starts the program, it displays the main login form and here one will choose the domain. The form displays the library information and two domains “student” and “admin”. The form is of following type: -

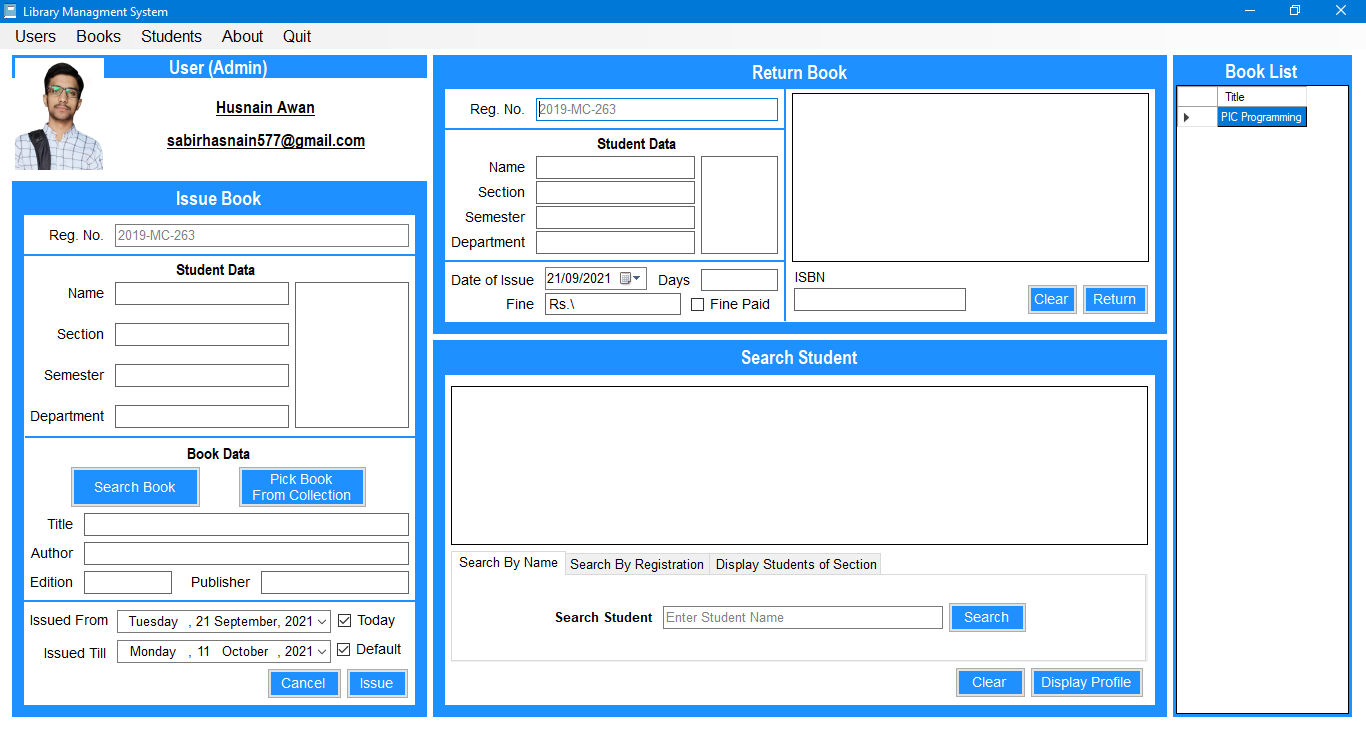
**Figure 2: (A) Choose Domain Form (B) Admin Form (C) Student Form**

The above figures show the different views of the form depending upon the user’s selection. User will enter the required data and proceed forward for the next main form. Here if any of the parameter is incorrect it will display and error message and one can’t log into the account.

Here a same form is in three different conditions for this panels are being used and depending on the user’s click the transition between panels took place and panels become invisible or visible to the user. This strategy saves a lot of work of making new and same panels for three different phases of the login form.

## **Admin View Form: -**

This is the main form if the user is an admin. The form controls all the tasks of the software i.e., add or search books, issue or return books etc. The form is basically an empty form with different panels on it but other forms are inherited to the main form. The form is shown below: -

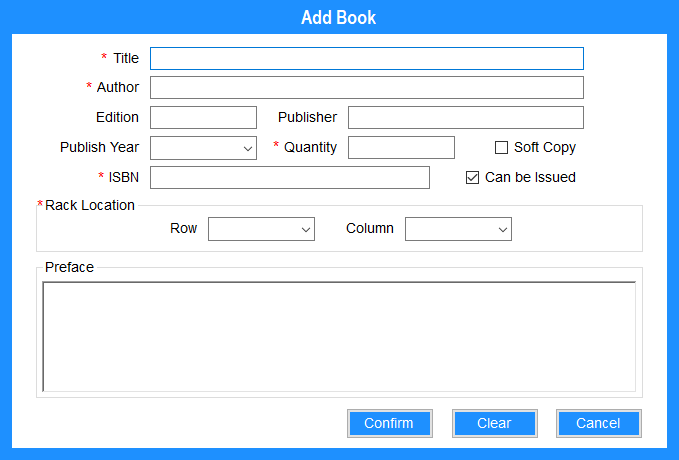


**Figure 3: Admin’s Main Form**

It displays some important functionalities of the software. One can directly issue, return or search books from here and a list of books saved into the database will be displayed on the right side of this form. The form is actually using other forms which will be explained below these are pinned here just for user’s ease.

* **Add New Books Form: -**

The below form is used to enter new book into the database of the system, it includes some basic information about the book, the data will flow to the class Books ad then to the data base.



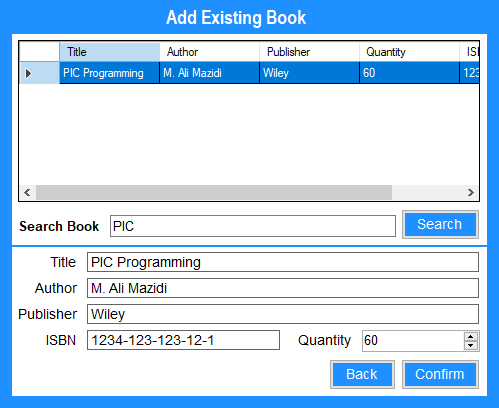
**Figure 4: Add Books Form**

Some quantities are labeled by a ‘\*’ symbol they are necessary parameters and should be entered to save the book otherwise it will display an error that the book can’t be saved due to incomplete information. Here the book quantity is selected and it is being whether the book is good to issue and is there a soft copy available of the book it will also record the position of the book in the respective row and column of the rack in which book will be kept.

User can use confirm button after entering the data or the book will be saved to the data base and can press clear to clear all the entered parameters in the boxes and cancel to go back to main windows of the software.

## **Add Existing Books Form: -**

This form is used to increment the quantity of the existing book here a book can be searched in the search bar by title of the book all the possible matches will be shown in the data grid view and one can select any of the book and increase its quantity.



**Figure 5: Add Existing Books**

This directly access the data base and compares entered title with the titles in the data base then those titles which are near to the entered text will be displayed in the grid view and one select any of the book from those or which he/she wants to increase in quantity. Thus, by selecting the book all the information of the book will be transferred to the labels placed down there this will be done by comparing the Book\_ID. Thus, by typing or using up down buttons one can change the quantity and it will be updated.

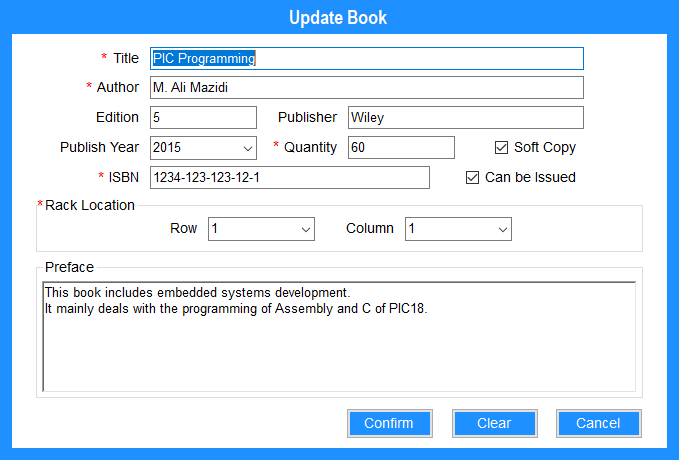
## **Display Books Form: -**

This form can display the list of the books saved in the data base with the complete information. User can edit or delete the books from there and can also add more books in the data base. It displays books on a data grid view. It basically gets all the books from the data base and moves them to the data grid now if a user wants to edit the books it will transfer the data flow to the class book and here the data will be saved in the variables in the class then in no time a new form opens in which one can edit the book. It will be the same form which is used to add books. Thus, a form is used for two purposes one for the addition of new books and secondly for the editing of existing book. The display book dorm is shown below: -



**Figure 6: Display Books Form**

It is the complete list of the saved books shown here as there is only one book in the data base so it is showing the same. User can search a book there, refresh the table after adding new data or in case of any malfunction, delete or update books or add new books. if user wants to delete the book, he/she will select the book which should be deleted and choose delete option. By selecting and pressing update book one can edit a book as shown below.

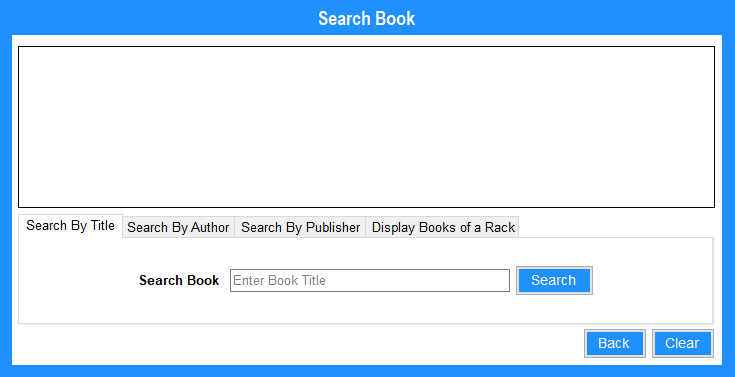


**Figure 7: Updating a book**

This is the interface if one wants to edit a book it is the same in which new book was added. It is being used as a book editor here by using polymorphism. The form is designed in such a way that it can understand whether data is being saved or being edited. User can enter new data and press confirm to update the data then a message will be displayed that your data has successfully been updated.

## **Search Books Form: -**

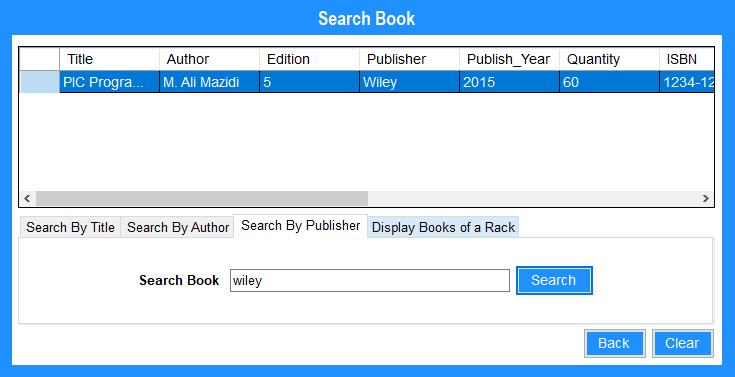
The form is used to search books out of a list of many books user can enter data according to the searching conditions and results will be displayed in the data grid view this feature.



**Figure 8: Add Existing Books**

This feature is also available on the main admin form basically that was its shadow on the main form the original form is here. In this form one can search book by title, author name, publisher or by the information of the rack.

In each case a list of possible books will be displayed in the data grid view. This also searches books directly from the data base. Entered text is compared with the data in the data base and results are shown. Following results are for the publisher-based search of books: -



**Figure 9: Add Existing Books**

## **Add Admin User Form: -**

In this form a new admin user can be added to the data base. one can choose this to create a new admin for the software and the form is displayed below: -

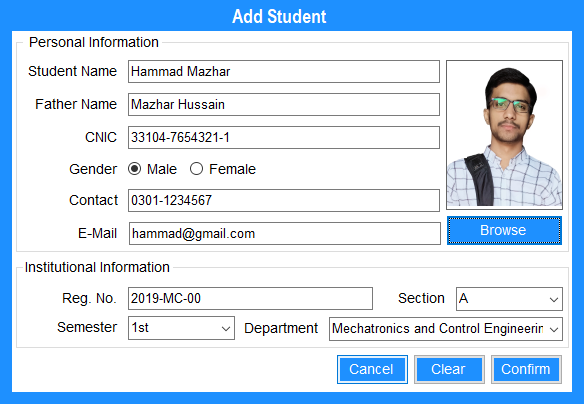


**Figure 10: Adding a new Admin user**

Thus, after filling the profile one can create a new admin for the software. Here the data will be saved to the data base and will automatically be encrypted in the form of random text. The email is used for the automatic email alerts. The image can be chosen from the computer storage and will be saved to the data base in the form of the byte format. An image when saved in the data base it is first converted to the text type format which is an array or buffer of the byte data type thus image is coded to a new format and then decoded when it is displayed to the user. Thus, a user can access the image even if it is not present in the computer storage but in the SQL Server.

## **Add Student Form: -**

In this form one can enter a new student user to the data base and can let him to access the library account. The form is shown below: -

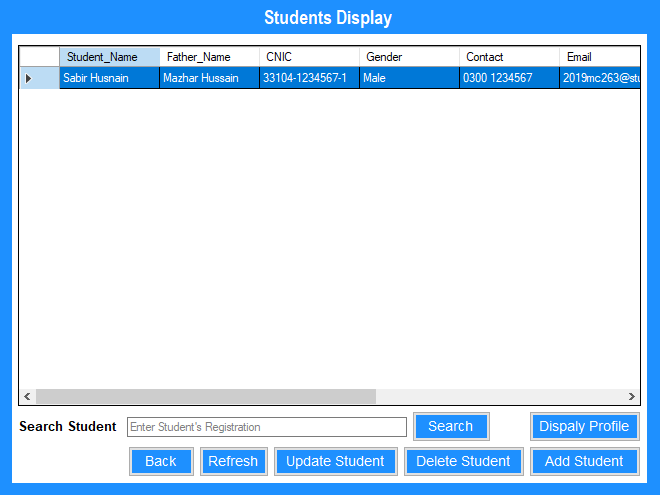


**Figure 11: Adding a New Student**

The above form is used to enter a new user to the system. The user information is entered and is saved to the data base. user can also upload an image of his/her to the software and thus he/she can be recognized in a better way. After adding a new user, he/she can access his/her account which will automatically be created and a digital student card will be issued to him/her. It will be explained in above forms. Same as above the image is converted to a buffer of bytes and thus saved to the SQL Server.

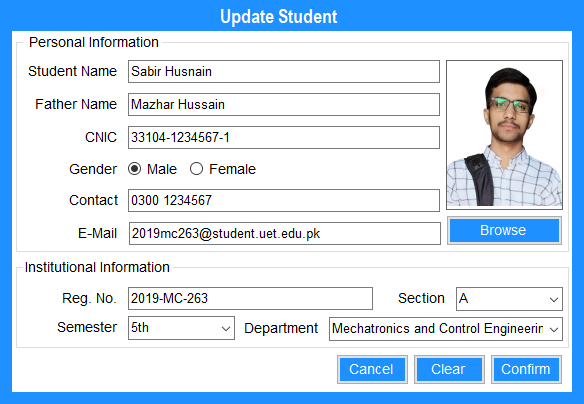
## **Display Student Form: -**

This form can display the list of the students saved in the data base with the complete information. User can edit or delete the student from there and can also add more students in the data base. It displays students on a data grid view. It basically gets all the students from the data base and moves them to the data grid now if a user wants to edit the student it will transfer the data flow to the class students and here the data will be saved in the variables in the class then in no time a new form opens in which one can edit the student. It will be the same form which is used to add student. Thus, a form is used for two purposes one for the addition of new books and secondly for the editing of existing student. The display student dorm is shown below: -



**Figure 12: Display Student Form**

It is the complete list of the saved student shown here as there is only one student in the data base so it is showing the same. User can search a student there, refresh the table after adding new data or in case of any malfunction, delete or update student or add new student. if user wants to delete the student, he/she will select the student which should be deleted and choose delete option. By selecting and pressing update student one can edit a student as shown below.

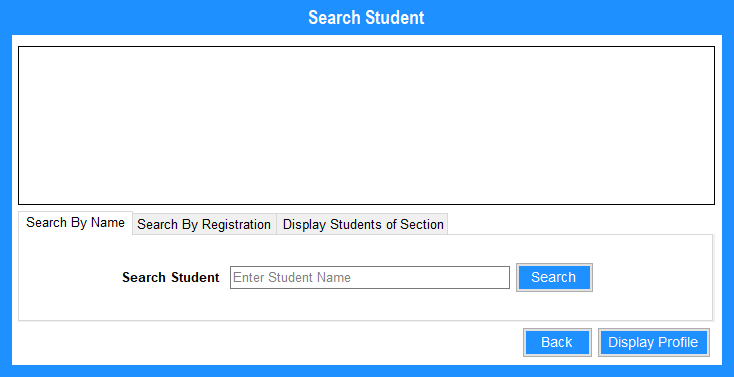


**Figure 13: Updating a student**

This is the interface if one wants to edit a student it is the same in which new student was added. It is being used as a student editor here by using polymorphism. The form is designed in such a way that it can understand whether data is being saved or being edited. User can enter new data and press confirm to update the data then a message will be displayed that your data has successfully been updated.

## **Search Student Form: -**

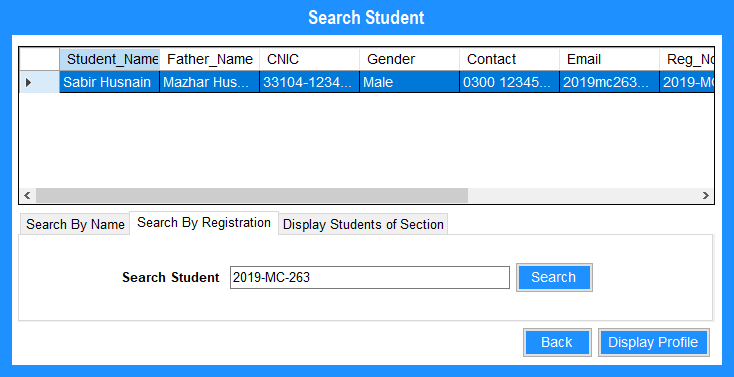
The form is used to search student out of a list of many students, user can enter data according to the searching conditions and results will be displayed in the data grid view this feature.



**Figure 14: Student Search Form**

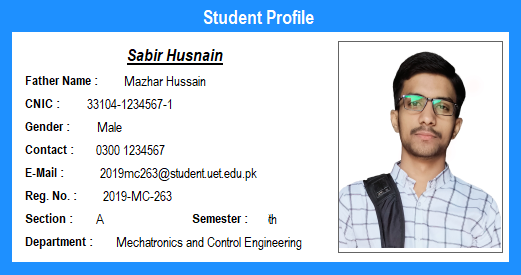
This feature is also available on the main admin form basically that was its shadow on the main form the original form is here. In this form one can search student by Name, registration number or by the academic information such as section and department etc.

In each case a list of possible students will be displayed in the data grid view. This also searches students directly from the data base. Entered text is compared with the data in the data base and results are shown. Following results are for the registration number-based search of student: -



**Figure 15: Add Existing Books**

## **Student Profile Form: -**



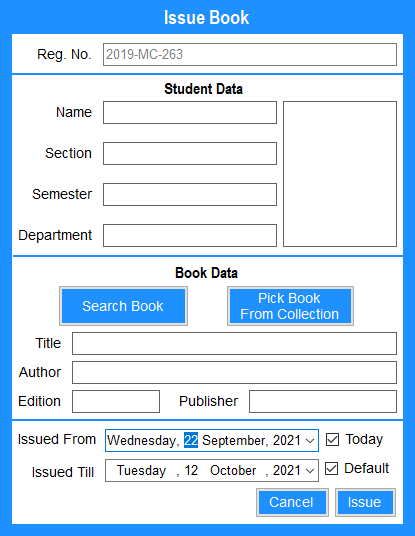
**Figure 16: Digital Student Card**

This is a unique form which can also be known as digital student card. When a student is added into the data base software create one for the student which is this form. It displays the complete information of the student along with image of the student. It displays information in a classical card format as shown above.

It basically retrieves the data form the data base by knowing the student id. When a student is searched in the search list and selected the class Students returns the id of the student on which basis student information is displayed in the profile or card. It has no buttons to keep its classical card appearance. To close the student card just click on any where on the card it will disappear from the screen. This operation is obtained by creating a click event for the card’s body and then closing that.

## **Issue Book Form: -**

This form is used to issue different books to students from the library data base. this form accesses the student, books and another table records issue data of the books. the form is shown below: -



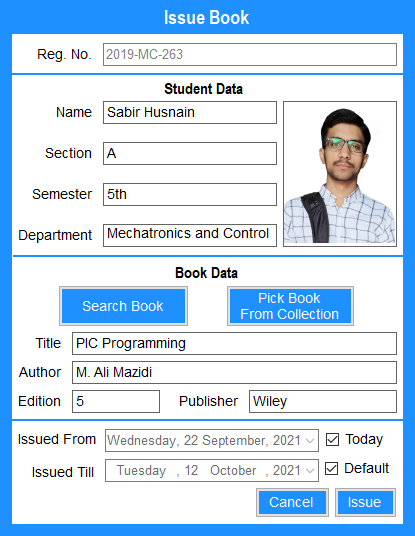
**Figure 17: Issue Book Form**

The form gets the registration number of the student and when user presses enter or moves the cursor away from the text box in which registration number was entered form automatically searches the student from the data base and fills the other records which will help the admin to issue book to a particular student.

Then there are two options of searching a book or picking a book from the collection. As the name suggests when a user clicks on search book, he/she can select a book by searching from the collection and if pick book form collection is selected, the complete list of the books will be displayed user should choose one of these books by double clicking it.

When search book is pressed it open the same form which is sued to search the books. it facilitates the user to search book by name, author name, publisher or by rack position. Thus, search book form is again used here in a different way with come changes which means polymorphism is also used here in this form. If a user selects pick book form collection it will display the form which is used to display books list thus it is also a form in which polymorphism is used which makes it very easy to manage the data flow and other things in the software.

Another important thing in this form is using of threads, here if the user searches or wants to pick the book the software will not continue and when the user double clicks the book it will generate a new thread in system and software will the books data in the form so that it can be saved to the record. it saves which book was issued, to whom it was issued, on which date and for how many days. It also decrements the quantity of the book from the collection and increment the number of issued books in students record thus in a single clicking many of the data is being changed.

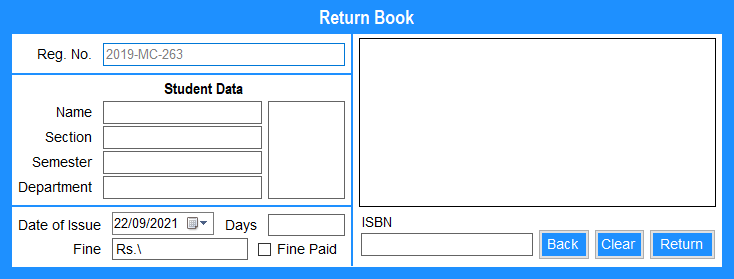


**Figure 17: A Book is being Issued to a Student**

The above figure displays how a book is issued to the student, here is today is selected against the issued from date it will automatically picks the current date which is on computer and if it was unchecked user can select any of the date. The check box Default against Issue Till will choose the date 20 days after Issued From and if it is unchecked any date can be picked. Theses dates will be saved to the data base and it will decide the late fee or fine for the user for current book which is being returned when returned.

## **Return Book Form: -**

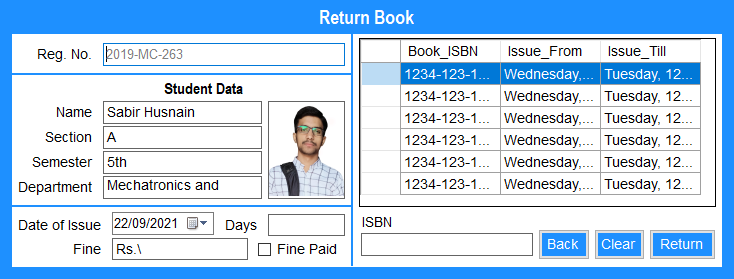
The return book is used to return the issued books to a specific user. One can search the roll number of the suer to view total number of books issued to the user and these are displayed in the data grid view. User can select a book which he wants to return and the system will remove the book from user’s profile if there are no remaining dues but if there are any dues, he/she can’t return the book before submission of dues. The form is shown below: -



**Figure 18: Return Book Form**

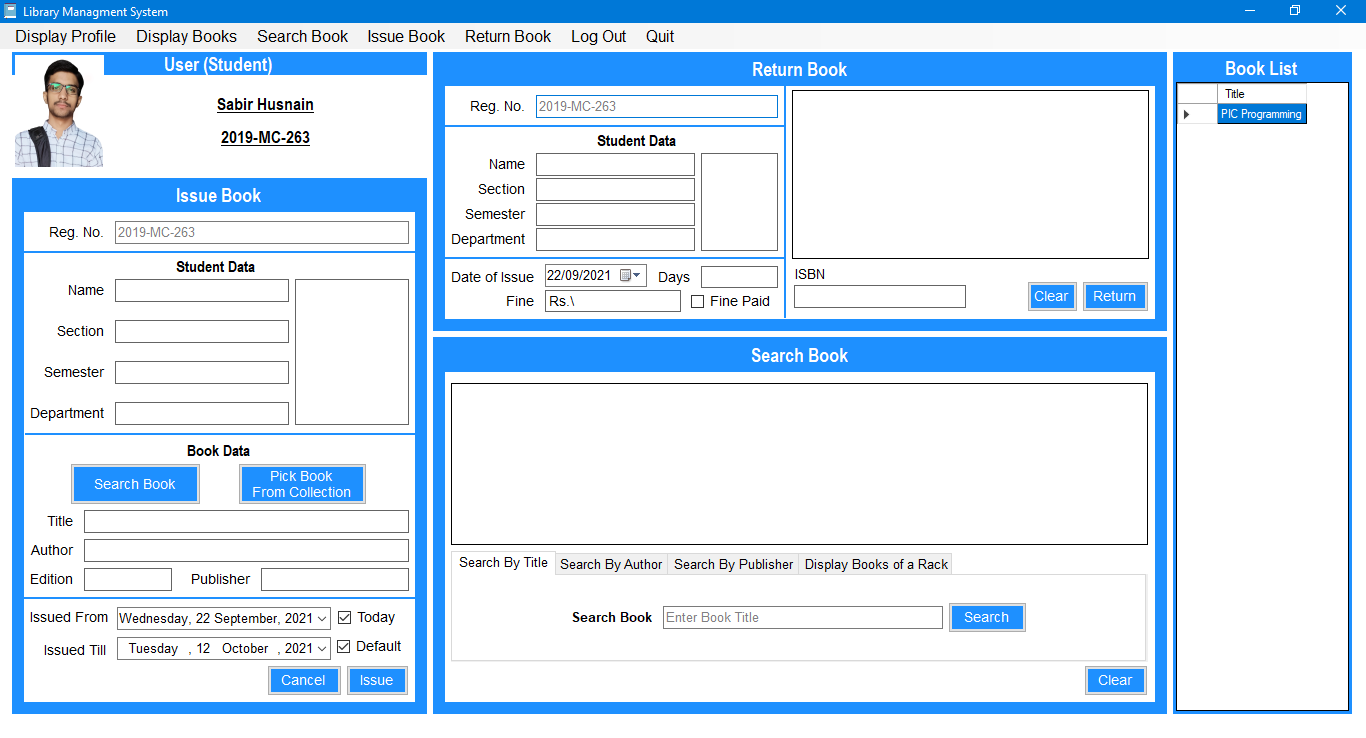
If the suer searches a student, he/she will be displayed in the form under Student Data tag. The issued books against the student profile will be displayed in the data grid view and user can select a book whose ISBN will be displayed below and press return to return the book.

The book will be removed from the student’s profile incremented in the books data and decremented from student data. Thus, there is also a use of threads and interrupts which are very useful in the process. The below figure shows how a book can returned.



**Figure 19: Returning a Book by a User**

## **Student View Form: -**



**Figure 20: Student View Form**

This form is used when a user login as a student this is the form which was used as admin view form but in case of admin, we haven’t any restrictions but, there are some restrictions in student view form. The restricted functionality for the student is explained above. The form is actually an empty form but some other forms are just pined here in the panels. This is just for users ease and student can access different forms from the same mane form.

# **Comments: -**

* The software is overall representing the skills used in Object Oriented Programming (OOP).
* The software uses a data base which is very easy to manage and many computers can be connected to a same server.
* There is a use of advanced encryption scripted by using Linux so it is also a secure software with respect to user’s privacy.
* The software is very straight forward and easy to operate.
* A basic computer user can use the software very easily without any advanced knowledge of computer and software or programming.
* On the end of programming, it is programmed using a .NET Framework on WinForms thus it is very easy to understand.