**Instructions on loading our project:**

1) Please restore the **SA47\_Team01ACADatabase.bak** in SSMS

2) Open the SA47\_Team01ACAProject.csproj (inside SA47\_Team01ACAProject folder) in Visual Studio

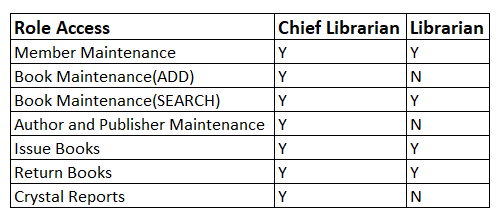
3) In Visual Studio, open the Server Explorer to add the database connection to **SA47\_Team01ACADatabase**.

4) Pick the Program.cs in Visual Studio Solution Explorer and press Start to run the project.

5a) If you want to log in as the Chief Librarian, key in your credentials (Username = **Venkat** and Password = **P@ssword1**).

5b) If you want to log in as a normal Librarian, key in these credentials (Username = justinjustin and Password = pa0s1s2w3o4r5d) or any other credentials in the UserMaster datatable.

5c)Access levels to the forms and actions are based on roles. Below is the table depicting the access roles(Y = yes, N = No) of librarians and chief librarian.



**Highlights of our project:**

**Book and Member Maintenance Form**

1) 2 indexing programming approaches are used in the Member and Book Maintenance forms.

2) In the Member and Book Maintenance forms, we have null-check mechanism to ensure textboxes are not null when inserting, updating and deleting the data.

3) We created a warning mechanism once the first and last records are reached.

4) We have null-check mechanism to ensure that when pressed, the Find button will not return anything when the Enter Member ID field is empty.

5) We created a warning mechanism to send a message when the system cannot find the member/ book in the database.

6) We created a warning mechanism to prevent user to navigate between records before a record has been loaded into the form

7) We created a warning mechanism to alert user before deleting the record

8) We limit character input to numeric for ISBN (can only enter 13 digits), Total Stocks and Number Loaned textboxes for Book Maintenance Form; Postal Code (can only enter 6 digits) and Contact Number (can only enter 7 digits) for Member Maintenance Form.

9) For Member Maintenance Form: member can only enter their birthdate if they are between age 7 (minimum member age) and age 120 (maximum member age)

10) We created a mechanism to prevent the user from having a Total Stocks amount less than Numbers Loaned and vice versa.

**Author and Publisher Maintenance Form**

1) When the form loads, it is programmed to automatically query the database for author and publisher data to fill the Author and Publisher datagrids.

2) We have a validation check at the Name textboxes to ensure that existing Authors/Publishers cannot be added.

3) If an Author/Publisher is found, the respective row in the corresponding datagrid will be highlighted.

4) Another validation check is done at the "Add" button to ensure that if a user still tries to add an existing Author/Publisher, he is prevented from doing so.

**Login Form**

1) We stored the user credentials (username & password) from our UserMaster data table into a Generic List and used it to verify login attempts.

2) For the verification process,

a) we find out if the credentials are valid by checking the textbox values with the <UserMaster> List;

b) if the credentials are valid, we run a Lamda expression to cross-check the Employee datatable for the employee's designation;

c) depending on the designation, the program will proceed to enable the user to access either the Chief Librarian or normal Librarian user interface.

**Landing Page**

1) When the page loads, it is programmed to automatically query the IssueTransaction datatable for books which have an "Extension Requested" remark.

2) Data of the books found will be used to populate the datagrid so the librarian will be able to review outstanding requests.

3) The librarian can thereby approve/reject the requests and in doing so, the database is updated accordingly.

4) After approving/rejecting any request(s), he can then refresh the datagrid to show the remaining outstanding requests.

[BooksOnLoan]

1) When the page loads, it is programmed to automatically query the IssueTransaction datatable for books which have an "out" status.

2) Data of the books found will be used to populate the datagrid so the librarian will be able to review books currently out on loan.

3) If a member asks the librarian for a book which is out on loan, the librarian can select the book title (via its BookID) and immediately check for its availability in stock.

4) When he does so, the bottom datagrid will be updated with details for that BookID.

**Multiple Search Form**

1) In the search for books with various criteria there are the options to search via single criterion or multiple options such as by category, author and genre. Created KeyPress events in order to handle the input of only numerical keys for search criteria of total stocks and loaned stocks. Users can input the name of the books into the search for book single criterion field for a search alternative other than scrolling the list. Added a "Reset" button for the multiple search option to restart selection whereas for the single criterion search, the filtered option in the field is immediately reset to empty value after clicking the "..." button and datagrid will display the listed results and number of records will be displayed at the bottom of the form.

**Issue Book**

1) Upon entering Member ID and Book ID, the corresponding Member and Book Name would be fetched from the database

2) We have validation check at the ID fields (Member and Book), to ensure that invalid IDs are not accepted

3) We have 2 Loan Periods- Normal and Special (radio boxes selected automatically according to the month. We used "DateTime.Now.Month" to check this) During normal period, each member can borrow a maximum of 5 books. During Special period (June and December), each member can borrow a maximum of 10 books We have used validation to limit users beyond this limit

4) Also, we have validation for checking if there are enough stocks of each book before issuing it. When "number loaned" for a particular book >= "total stock", the book can’t be issued.

5) We have used "Transaction Scope", so that when the system crashes unexpectedly after updating 1 table, the changes would be reversed.

**Return Book**

1) When a member wants to return a book, upon entering his/her member it automatically displays the list of books borrowed by him. The particular book to be returned could be selected.

2) We have validation for invalid member id.

3) We have added "Transaction scope" here too.

**DataBase VIEWs**

Views can be found in the database under the Views Folder. If you open this folder you will see 3 views.

1) DueBooksperMemberView

This view shows the number of books which are overdue and not returned. It also list the members and the datedue and the total days overdue

2) MembersBooksRecord

This view shows the activities of members who have borrowed from the library and the total number of times requesting e.g. for Extensions, overdues, or any rejections for extensions

3)Top10Popu**lar**BooksByYear

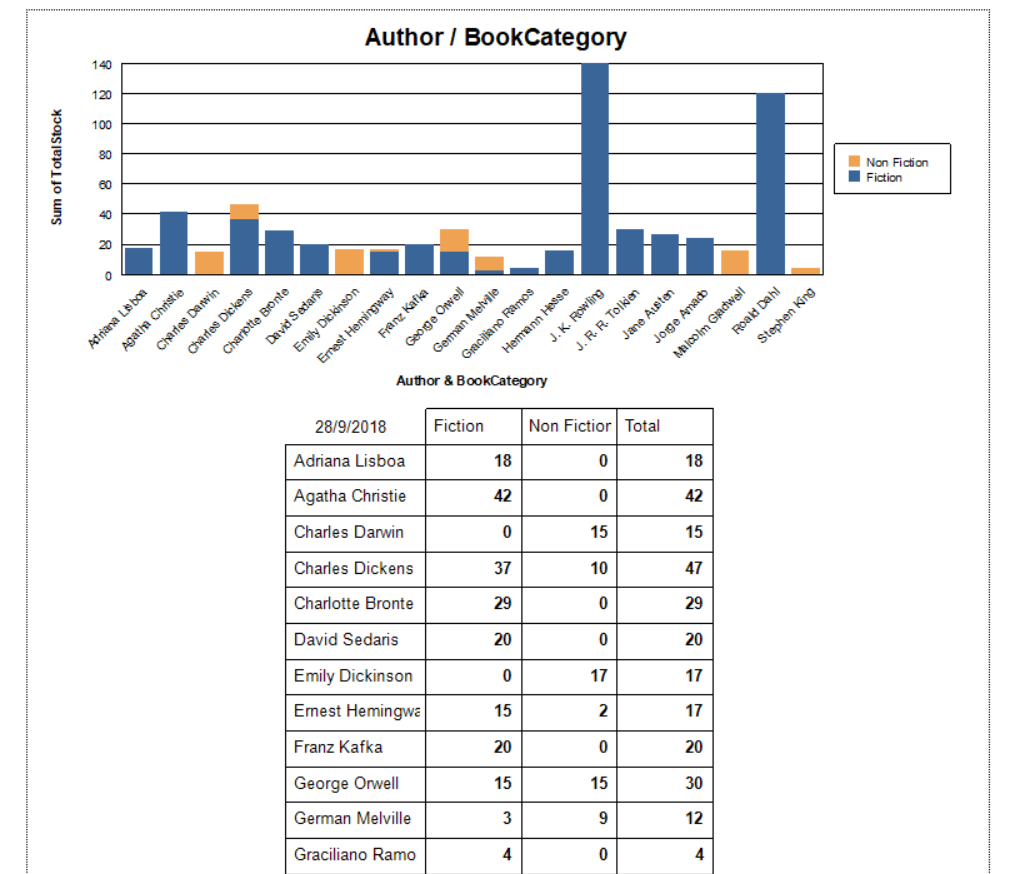
This view shows the top 10 more popular books borrowed in 2018

**Crystal Reports**

6 reports as followed:

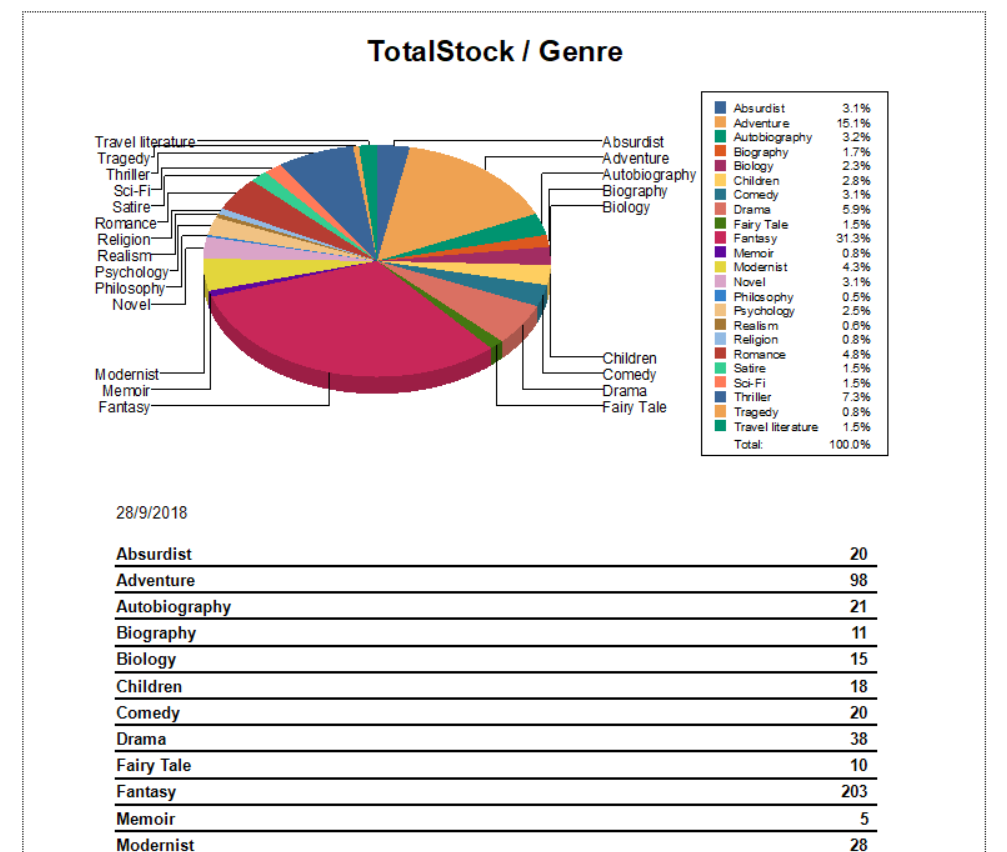
1. “AuthorByBookCategory” (Using Entity Framework)

This crosstab report shows per author the number of titles in each category(fiction or non- fiction) that they written in.

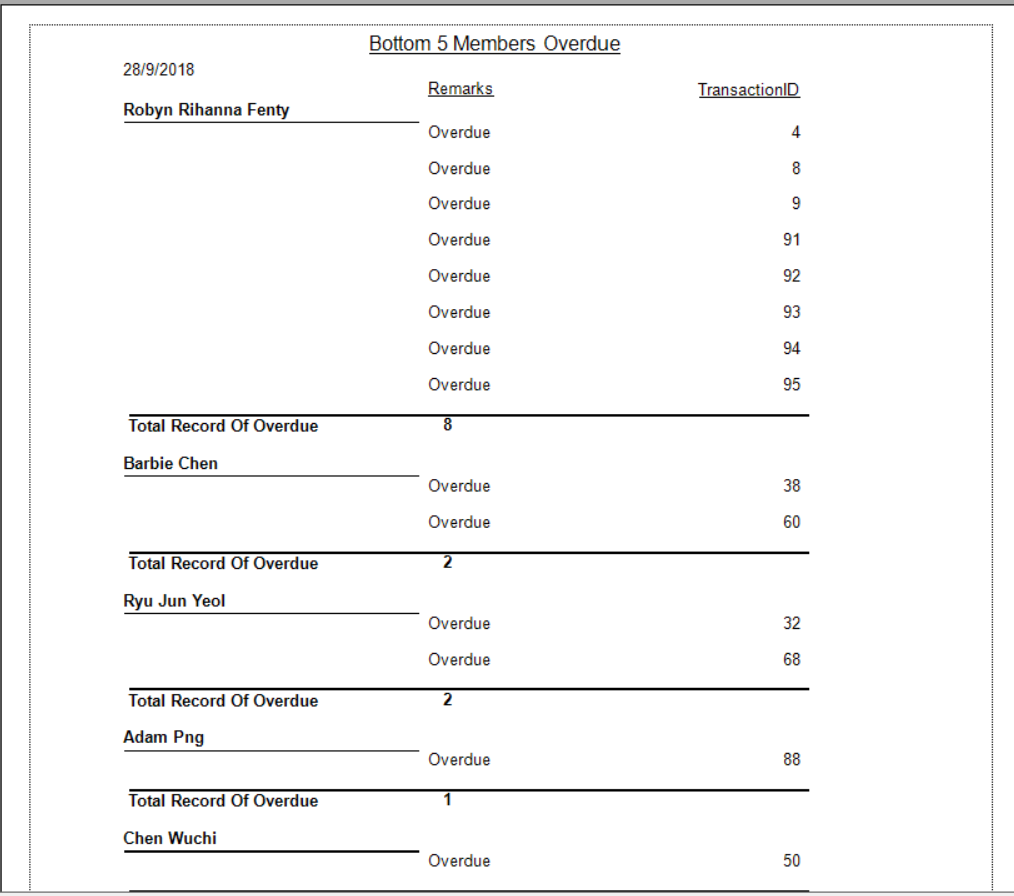


2)”BookStockByGenre” (Using Entity Framework)

This pie chart and report list shows the stock per genre.

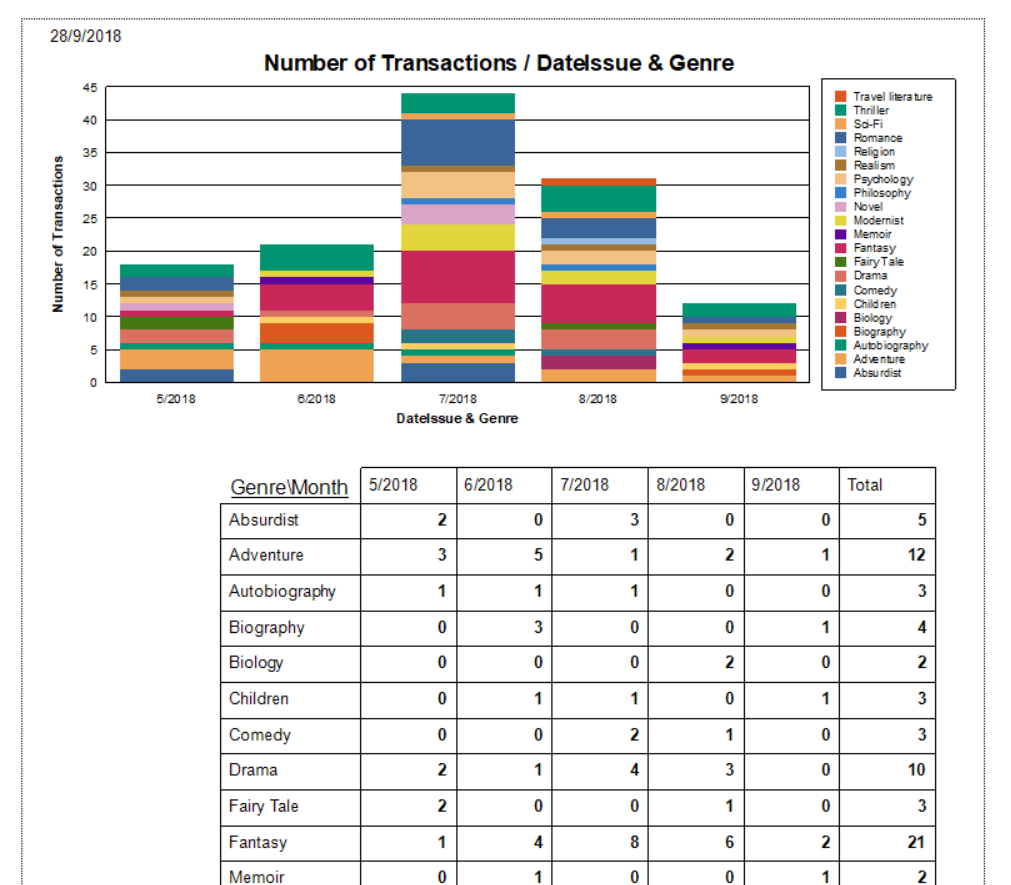


3) “Bottom5Members\_Overdue” (Using Typed Dataset)

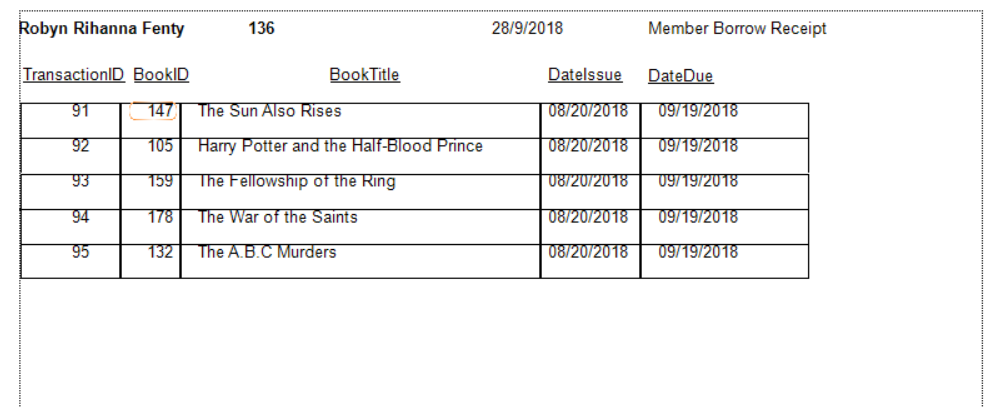
This report is tied to 2 tables and show the 5 members who have overdue records with the library

4) “TransactionByGenre” (Using Typed Dataset)

This crosstab report with stacked bar chart is tied to 2 tables and show the number of transactions against the genre and for each month.



5)”SingleBorrowReceipt” (Using Typed Dataset)

This report is tied to 3 tables and it shows the books borrowed detail for a single member.

6) “MembersBorrowreceiptList” (Using Typed Dataset)

This report is tied to 3 tables and it shows the books borrowed detail grouped by each member

