**EBook Management**

**Abstract:**

Ebook management system includes online and Physical end user experience which includes user login and author details to include their books in the Books storage. This system contains 5 tables namely Categories, Books, Users, BookBorrows, Book\_Borrow\_Master, Subscription details. Users have the option to enter their details when they login to the system. If the user wants to borrow book from the already existing books list in books table he/she can search according to the author, title, Publisher and language and then they can borrow when they want specific book in the resultset that are fetched from the Database.

User should return the book before the due date if not fine will be allocated to the user.

**Connected Architecture:**

* Connected architecture directly updates the Database according to the query execution without any external calling of method.
* It saves time when it doesn’t need to copy tables to another source.
* It is fast when compared to Disconnected architecture.

**Classes used:**

* **Program.cs :**

It contains the Main method of the namespace. From here all the classes are called by their object names.

It includes switch statements to navigate according to user input and preference.

* **DB.cs**

It contains the connection coding which is accepted as a string and the string is formed using **stringbuilder class** and it gets input from the user to which source it is going to connect. It is inherited in all the derived class mentioned below to open the connection to DB.

* **ChooseCategory.cs**

It includes the categories of the books to be inserted into the database and authors or admins can inserted according to the books category.

* **Add.cs**

Add classfile includes the details needed to add the book into DB. Values are get from the admin as per requirements.

* **User.cs**

User classfile includes the user details and if the user chooses online mode then the user id and their respective subscription details will be inserted into subscription table using **stored procedure CheckOnlineInsert.**

* **Search.cs**

Search class file is used in this context to search the books according to author, title, publisher and search results will be displayed. If user wants to borrow book then it will call the borrow object of Borrow class.

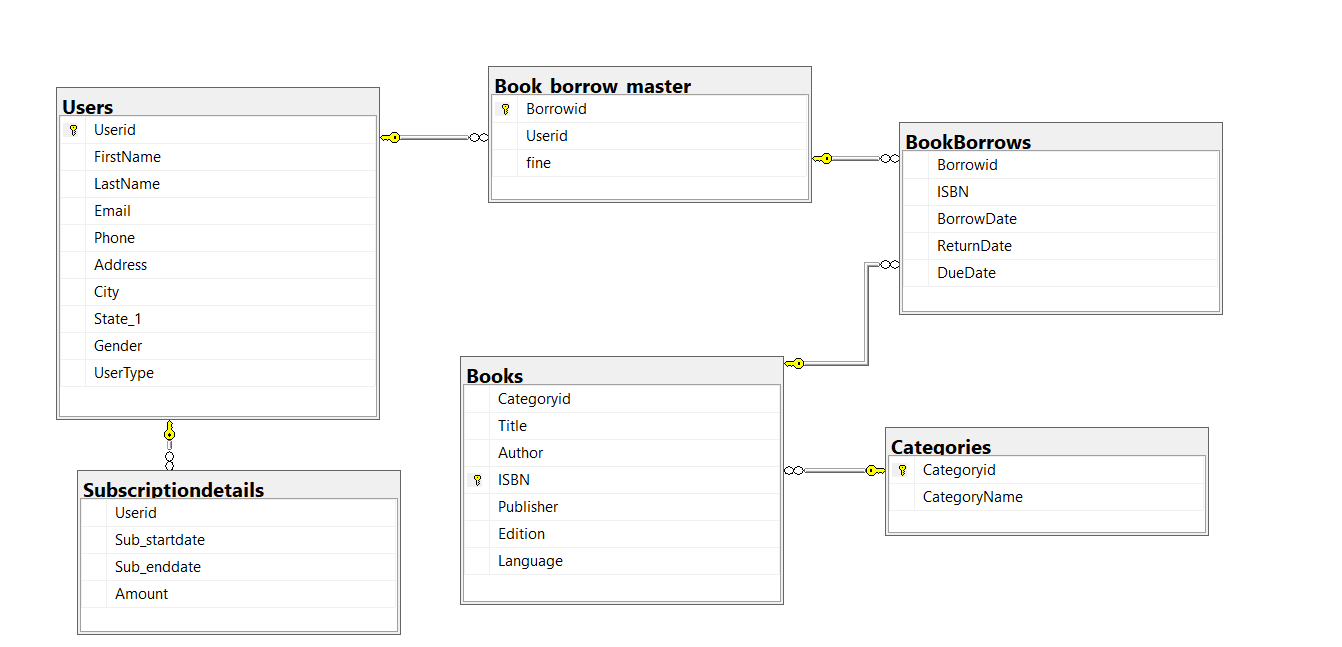
* **Borrow.cs**

Borrow class includes the **stored procedure BorrowBook**  to insert the values in the BookBorrows and Book\_Borrow\_Master table.

* **GiveBack.cs**

It includes the queries to return the book that are borrowed by the user . It checks whether the user returns the book within the due date if not user will be fined.

**DB Schema:**

****