**BOOK STORE MANAGEMENT Winform Project Documentation**

**1. Actors**

**Staff**

* Can manage books, publishers, and authors.
* Has CRUD (Create, Read, Update, Delete) permissions for books, publishers, and authors after logging in.

**Admin**

* Can manage staff.
* Can see the list of staff and create or disable specific staff after logging in.

**2. Database**

**Tables**

**User**

* UserId (PK) - int
* FullName
* Password
* Email
* Role
* Status

**Books**

* BookId (PK)
* BookName
* AuthorId (FK)
* PublisherId (FK)
* CategoryId (FK)
* Quantity
* Price

**Publishers**

* PublisherId (PK)
* PublisherName
* PublisherEmail
* PublisherDescription
* PublisherPhoneNo

**Authors**

* AuthorId (PK)
* AuthorName
* AuthorDescription
* AuthorAge

**Categories**

* CategoryId (PK)
* CategoryName
* CategoryDescription

**3. Screens (Components)**

**Authentication**

**Login**

* Fields: UserId, Password
* Button: "Forget Password?"

**Forgot Password**

* Fields: UserId, Password, Confirm Password
* Button: "Submit"

**Main Screen**

* Welcome Label
* Buttons: "Book Management", "Publisher Management", "Author Management", "Logout"

**Books**

**Book List**

* Data of books
* Search bar
* Button: "Add Book"

**Add Book**

* Fields to add a book (optional: add Publisher, add Author, add Category)

**Book Detail**

* Book Information
* Buttons: "Update Book", "Delete Book"

**Update Book**

* Fields to update a book (optional: add Publisher, add Author, add Category)

**Publishers**

**Publisher List**

* Data of publishers
* Search bar
* Buttons: "Add Publisher", "Delete Publisher", "Update Publisher"

**Add Publisher**

* Fields to add a publisher

**Update Publisher**

* Fields to update a publisher

**Authors**

**Author List**

* Data of authors
* Search bar
* Buttons: "Add Author", "Delete Author", "Update Author"

**Add Author**

* Fields to add an author

**Update Author**

* Fields to update an author

**Categories**

**Category List**

* Data of categories
* Search bar
* Buttons: "Add Category", "Delete Category", "Update Category"

**Add Category**

* Fields to add a category

**Update Category**

* Fields to update a category

**Admin**

**Staff List**

* Data of users with role Staff
* Buttons: "Add Staff", "Delete Staff", "Update Staff"

**Update Staff**

* Fields to update staff and change status

**Library**

* Data of books, publishers, and authors

**4. Flow**

**Default Screen: Login**

1. User navigates to the Login screen.
2. User enters UserId and Password.
3. User clicks "Login".
4. System verifies credentials:
   * If valid, redirects to the Main Screen.
   * If invalid, displays an error message.

**Forgot Password**

1. User clicks "Forget Password?" on the Login screen.
2. User is redirected to the Forgot Password screen.
3. User enters UserId, Password, and Confirm Password.
4. User clicks "Submit".
5. System updates the password if the UserId is valid and the passwords match.
6. User is redirected back to the Login screen.

**View Book Detail**

1. User navigates to the Book List screen.
2. User selects a book from the list.
3. System displays the Book Detail screen with information about the selected book.

**CRUD Books**

**Create Book**

1. User navigates to the Book List screen.
2. User clicks "Add Book".
3. User enters book details (optional: add Publisher, add Author, add Category).
4. User clicks "Submit".
5. System saves the new book and updates the Book List.

**Read Book**

1. User navigates to the Book List screen.
2. User views the list of books.
3. User can search and filter books using the search bar.

**Update Book**

1. User navigates to the Book List screen.
2. User selects a book to update.
3. System displays the Book Detail screen.
4. User clicks "Update Book".
5. User updates the book details (optional: add Publisher, add Author, add Category).
6. User clicks "Submit".
7. System saves the changes and updates the Book List.

**Delete Book**

1. User navigates to the Book List screen.
2. User selects a book to delete.
3. User clicks "Delete Book".
4. System prompts for confirmation.
5. User confirms the deletion.
6. System deletes the book and updates the Book List.

**CRUD Authors**

**Create Author**

1. User navigates to the Author List screen.
2. User clicks "Add Author".
3. User enters author details.
4. User clicks "Submit".
5. System saves the new author and updates the Author List.

**Read Author**

1. User navigates to the Author List screen.
2. User views the list of authors.
3. User can search and filter authors using the search bar.

**Update Author**

1. User navigates to the Author List screen.
2. User selects an author to update.
3. User clicks "Update Author".
4. User updates the author details.
5. User clicks "Submit".
6. System saves the changes and updates the Author List.

**Delete Author**

1. User navigates to the Author List screen.
2. User selects an author to delete.
3. User clicks "Delete Author".
4. System prompts for confirmation.
5. User confirms the deletion.
6. System deletes the author and updates the Author List.

**CRUD Publishers**

**Create Publisher**

1. User navigates to the Publisher List screen.
2. User clicks "Add Publisher".
3. User enters publisher details.
4. User clicks "Submit".
5. System saves the new publisher and updates the Publisher List.

**Read Publisher**

1. User navigates to the Publisher List screen.
2. User views the list of publishers.
3. User can search and filter publishers using the search bar.

**Update Publisher**

1. User navigates to the Publisher List screen.
2. User selects a publisher to update.
3. User clicks "Update Publisher".
4. User updates the publisher details.
5. User clicks "Submit".
6. System saves the changes and updates the Publisher List.

**Delete Publisher**

1. User navigates to the Publisher List screen.
2. User selects a publisher to delete.
3. User clicks "Delete Publisher".
4. System prompts for confirmation.
5. User confirms the deletion.
6. System deletes the publisher and updates the Publisher List.

**CRUD Categories**

**Create Category**

1. User navigates to the Category List screen.
2. User clicks "Add Category".
3. User enters category details.
4. User clicks "Submit".
5. System saves the new category and updates the Category List.

**Read Category**

1. User navigates to the Category List screen.
2. User views the list of categories.
3. User can search and filter categories using the search bar.

**Update Category**

1. User navigates to the Category List screen.
2. User selects a category to update.
3. User clicks "Update Category".
4. User updates the category details.
5. User clicks "Submit".
6. System saves the changes and updates the Category List.

**Delete Category**

1. User navigates to the Category List screen.
2. User selects a category to delete.
3. User clicks "Delete Category".
4. System prompts for confirmation.
5. User confirms the deletion.
6. System deletes the category and updates the Category List.

**Change Status of Staff**

1. Admin navigates to the Staff List screen.
2. Admin selects a staff member to update.
3. Admin clicks "Update Staff".
4. Admin updates the staff details and changes the status, either “Active” or “Disabled”.
5. Admin clicks "Submit".
6. System saves the changes and updates the Staff List.

**HÃY ĐỌC TRANG DƯỚI NẾU KO NHỚ NHỮNG THƯ VIỆN HOẶC CÂU LỆNH CHÚNG TA SẼ XÀI.**

**REFERENCES *(this session just for reference, student can use the other approach to do Practical Exam)***

***Working with DB connection string from JSON file****.*

* 1. In the Presentation layer (WindowsForms Project), you create *appsettings.json*and add ConnectionString same as the bellow to config the connection string to SQL Server Database.

*{*

*"ConnectionStrings": {*

*"DefaultConnectionStringDB": "server =(local); database=***AirConditionerShop2023DB***;uid=****sa****;pwd=****1234567890****; TrustServerCertificate=True"*

*}*

*}*

You can change **server**, **uid** and **pwd** to suitable with your local machine.

* 1. Set property "Copy to output Directory" of *appsettings.json* file to "Copy if newer"
  2. Using Manage Nuget packages to install packages the following in Windows Form Project

*Microsoft.Extensions.Configuration -Version 5.0.0*

*Microsoft.Extensions.Configuration.Json -Version 5.0.0*

* 1. Using *ConfigurationBuilder* to init Configuration object for reading *appsettings.json* file same as this code:

*private string GetConnectionString()*

*{*

*IConfiguration config = new ConfigurationBuilder()*

*.SetBasePath(Directory.GetCurrentDirectory())*

*.AddJsonFile("appsettings.json",true,true)*

*.Build();*

*var strConn = config["ConnectionStrings: DefaultConnectionStringDB "];*

*return strConn;*

*}*

* 1. After that, durring development, student can bypass the ConnectionString (which read from *appsettings.json*) to Data access layer by constructor or others

*public partial class AirConditionerShop2023DBContext: DbContext*

*{*

*public AirConditionerShop2023DBContext (string connectionString)*

*{*

*this.Database.SetConnectionString(connectionString);*

*}*

*}*

***Package using for .NET 5:***

*- Install package using Tools → NuGet Package Manager → Package Manager Console*

Install-Package Microsoft.EntityFrameworkCore.SqlServer -Version 5.0.17

Install-Package Microsoft.EntityFrameworkCore.Tools -Version 5.0.17

Install-Package Microsoft.EntityFrameworkCore.Design -Version 5.0.17

Install-Package Microsoft.Data.SqlClient -Version 3.0.1

- *Install package using CLI or Power Shell*

dotnet add package Microsoft.EntityFrameworkCore.SqlServer --version 5.0.17

dotnet add package Microsoft.EntityFrameworkCore.Design --version 5.0.17

dotnet add package Microsoft.EntityFrameworkCore.Tools --version 5.0.17

dotnet add package Microsoft.Data.SqlClient --version 3.0.1

Entity Framework Core

*- Install dotnet-ef for CLI*

dotnet tool install --global dotnet-ef --version 5.0.11

*- Use Entity Framework Core to generate Object Model from existing database – CLI*

dotnet ef dbcontext scaffold "Server=(local);uid=**sa**;pwd=**1234567890**;database=***AirConditionerShop2023DB***;TrustServerCertificate=True" Microsoft.EntityFrameworkCore.SqlServer --output-dir Models

*- Generate database from domain classes – CLI.*

dotnet ef migrations add "InitialDB"

dotnet ef database update

Entity Framework Core

*- Use Entity Framework Core to generate Object Model from existing database – Package Manager Console*

Scaffold-DbContext "Server=(local);uid=**sa**;pwd=**1234567890**;database=***AirConditionerShop2023DB***;TrustServerCertificate=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models

*- Generate database from domain classes – Package Manager Console*

Add-Migration "InitialDB"

Update-Database -verbose