

Project Proposal

Project Tools

1. Code Design:-

- In this small project, I used **Blazor Server App** mechanism.
- Programming Language is **C#**.
- One Interface and one repository in the services folder inside the Models folder.
- The connection to the DB has been done by using **Entity Framework 5.02**.
- I have created DB Context, three models, one for Users, one for Loss type, and one custom model for Users and Loss Type together.
- For Login Page, I have created a `CustomAuthenticationStateProvider` that inherited from `AuthenticationStateProvider` to validate the credentials Identity.

2. Architecture/ Structure:-

- I used the Blazor Architecture, which consists of “wwwroot” folder for css, Images, scripts, and files.
- I placed the project in the Presentation folder, and the Documentation folder has a text file to add documentation.
- Models folder for models, custom models, DB context, and services.

3. Error Handling:-

- All code blocks have been written in Try Catch block. Following this, the error will be logged by using Logger control. Finally, the user will be sent to the Error Page.

4. Testing, Maintenance, and Extensibility:-

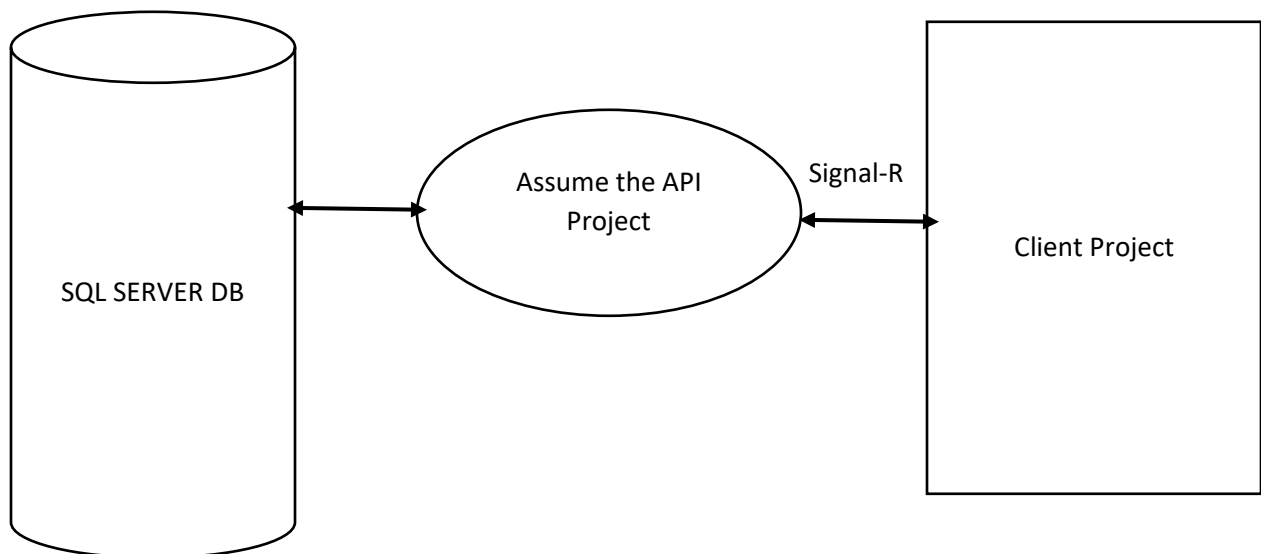
- The code has been tested successfully. Moreover, we can add extend our work by adding more models to the DB context, add more pages.

5. Design:-

- I have downloaded a template design, and then I added it to the Blazor server app project. I apologize if you will find some design problems.

Notes:-

- This project can be done using many other ways, such as using **Asp.net core only, MVC, or web forms**. However, since no methods have been mentioned, I used the Blazor because it is the **newest** methodology from Microsoft. Moreover, it will be the best choice to get real-time result as has been mentioned in the task “Extra credit: Our third party would like real-time updates on claim progress at each stage of its life cycle (creation, update, completion). Include this in your proposal”.
- The most important thing that I used the Entity Framework to connect to the Database. **This is not the best approach**. The **best approach is to use Web API**, and then connect the client project to the API project (Server Project), which will connect to the DB. I didn't use because when I upload it on the GitHub, the port number of the API will be changed, and you have to change it in the app.settings file in the client project. As a result, I decided to work on one project. However, if you want me to create and API, I can do that, it will return **JSON result**.
- I was thinking to use **Dabber** instead of Entity Framework to solve the Entity Framework performance issue, but I know that it will take a little bit time. Therefore, I was thinking if you want me to do it, I will do it on request because maybe it is better to use the Entity Framework at this stage.
- Important! For real-time result, we have to use the Signal-R in Blazor, which will interact with client browser. We can do this by adding end points in the services in the startup class (Please check the figure below).




Screen Design & Validations

1) Validation

Simple
Your tagline goes here

HomeView ClaimsLogin



Insurance Claims System
Company Name

Welcome to your system!

Login

Username

Please type your username.

Password


Please type your Password.

Login

2) Incorrect User

Simple
Your tagline goes here

HomeView ClaimsLogin



Insurance Claims System
Company Name

Welcome to your system!

Attention ! Your username or password is incorrect

Login


Ahmad

Login

3) Inactive User

Simple
Your tagline goes here

HomeView ClaimsLogin



Insurance Claims System
Company Name

Welcome to your system!

Attention ! Your username or password is incorrect

Login


bestg

Login

4) Active User

Simple
Your tagline goes here

HomeView ClaimsHello, Archie GemmellLogout



Insurance Claims System
Company Name

Welcome to your system!

Loss Type Code	Description	Last Updated By
F	Fire	Georgie Best
T	Theft	Archie Gemmell
O	Other	Archie Gemmell

HomeView Claim

Copyright Reserved for company, 2022 ©