Author

Dawid Noras

Directories:

- DataBase create SQL script – Here you will find script which will create database schema for you and populate it with some data.
- Executable – Here you will find the application, all needed Dll's and most important application config ("ElectoralCalculator.exe.config"). After you set up your database you will need to modify the connection string here to let application connect to your database.
- Source Code – Here you will find solution with all projects.

Database Setup:

First of all I was using Microsoft SQL with SQL Server 2014 Management Studio. After you will have that, you need to create your database by using my script (in "DataBase create SQL script" Directory). It all should be as easy as creating ne Query and just running it all. There might be some complications if you already have "Elections" database on your server, then you will need to change name of mine in script and in connectionString under "initial catalog=Elections;".

Speaking of Connection string, once you have your database up and running, in connection string find "provider connection string="data source=DAWID\SQLEXPRESS;" and change "DAWID\SQLEXPRESS" to your server address.

You might also put there login and password if your database requires it.

About Application:

I decided to use WPF for my user interface, because one of my goals for this and next year is to become very good and using it. Those 3 days of constant work gave me a lot of experience and I hope you will let me learn more from what you will say about my application.

There are many things and solutions there that I'm proud of. But there is one which I don't enjoy at all. I'm talking about my solution to a problem, when multiple application try to login with the same authentication data.

There are 2 ways of dealing with it (that I'm aware of). 1 is not to let second person login, when there is already other person logged on the same user. (that's what I ultimately choose). Second one is log off the first user, and let new one do what he likes.

I couldn't think of a better solution to this problem that would allow me to make application on time so my solution was to have a flag with user in his database record which tells me if someone is using this account at this moment. I set this flag when user login and set to 0 again when he logs off or he closes application. But there are cases that application get stuck (Exception or something else) that won't trigger log off code and user can no longer login without intervention.

Ultimately I think it should be done with 2 application. 1 to communicate with database and act like server to other applications (client ones). After login in server would send back some kind of token and respond to application only when shown it. That would fix a lot of problems, but again as I said for my experience I had no time to do this fully how I wanted to.

User Interface could use some improvement too 😊

Database consists of 4 tables:

- Electorate – those are users. Name, surname and pesel are all hashed with SHA512. I wrote above about "logged" flag. "voted" flag tells if user voted or not.
- LoginAttemps – all login attempt for statistics data
- Candidates – name and party of candidates. Pre created.
- Votes – all votes go here. Flag "valid" says if vote was valid or not (invalid vote is when you vote on 0 or more than 1 candidate). If flag "withRights" is checked, that mean someone without voting rights (under 18 years old or on blocked list) was trying to login to application.

Projects in solution:

- ElectoralCalculator – Main project with Windows and Pages. There are 2 windows in application. Login one, and main window. On main window I decided to use Pages to switch content to minimalize amount of opening and closing windows.
- BizzLayer – Business Layer used to get and put data to database with public API used by ElectoralCalculator
- DataLayer – Whole project made only for EF ORM.
- Model – I decided to separate my Model from this generated from database. (I'm not entirely sure if it was good or not. Most project have references for both of those.)
- ParsingUtility – Utility classes for parsing Json, Psesl. Serialize data. I've put here SHA too.

Summary:

If you have any questions or problems regarding application, don't hesitate to call me and ask.

I'm looking forward to your reply. I hope for the best.