IDEA P.C. NEKYA

Project: **Ce.R.T.H. B2B**

Author: Christos Lytras

✉ christos.lytras@gmail.com
✉ chris@nekya.com
✆ +30 698 037 2 501

Table of Contents

1. Specifications

1.1 Technologies & Frameworks

This application is built with:

- ASP.NET Core and C# for cross-platform server-side code
- Entity Framework Core for ORM with the database
- Identity Server for Authentication and Authorization
- React for client-side code
- Bootstrap for layout and styling

1.2 Requirements

1. Database server like SQL Server or MySQL Server
2. SSL certificate and permanent HTTPS redirection

   > ⚠ **Warning**
   >
   > The application and the Identity Server **cannot** operate with an invalid/expired or without a SSL certificate.
   >
   > There should be a scheduled SSL certificate renewal process, or an auto-scripted renewal.

3. An environment that has ASP.NET Core runtimes

   > ℹ **Information**
   >
   > Windows Server with an IIS web server are recommended but not limited to.

4. Git Bash for quick and automated update procedure

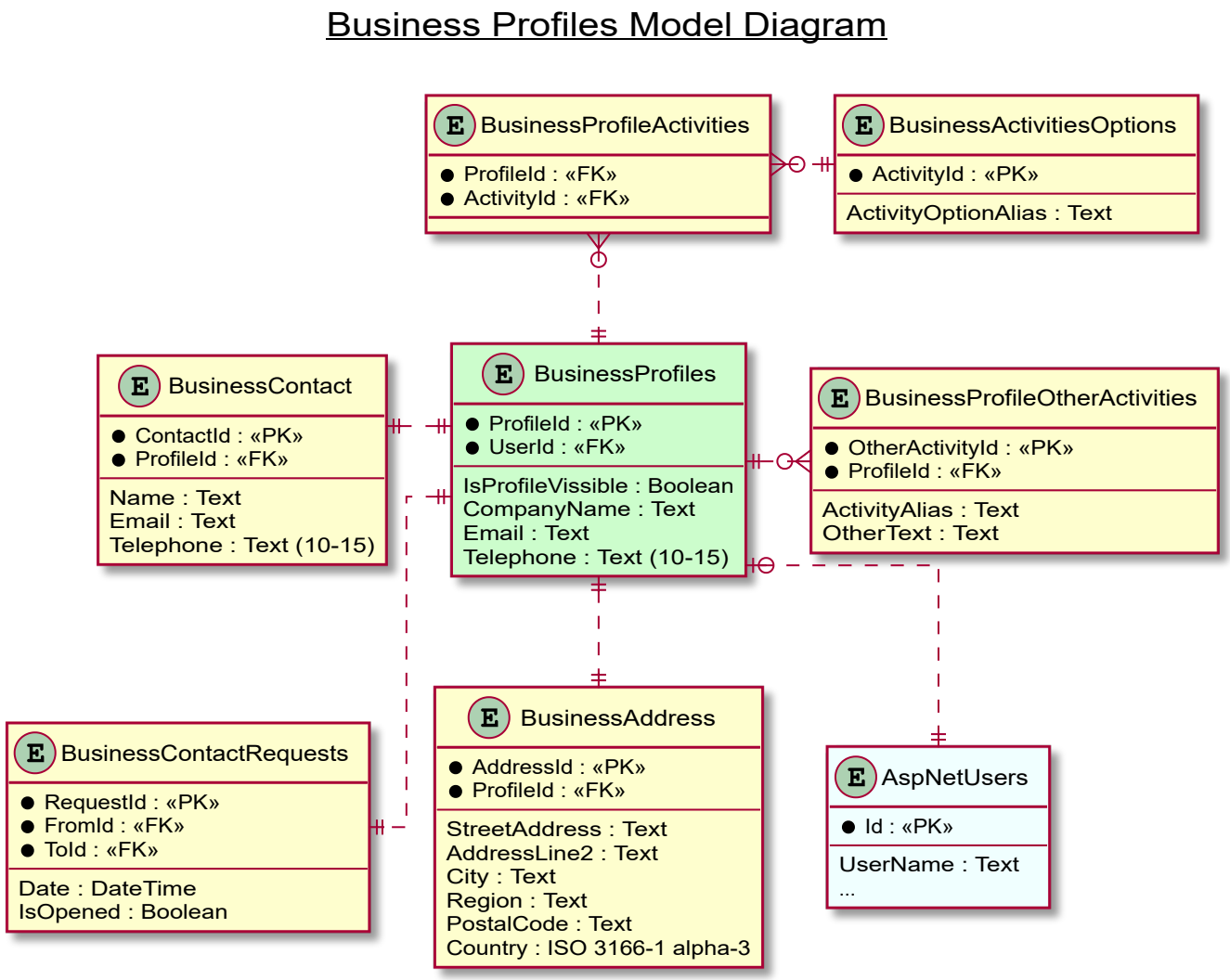
1.3 Compatibility

The application is compatible with the following minimum major browser versions

- Chrome 51
- Firefox 54
- Safari 10
- Edge 14
- Edge Chromium 80

Internet Explorer has major flows and standard CSS supporting issues. It also needs polypills to support ES6 standard features. For old browsers, there will be a warning message and if the application detects browsers that cannot be run into, there will be a notice to let final users know that their browser is old, insecure and propose upgrading to the latest secure versions.

1.4 Database Business Profiles Model Diagram



2. Installation

2.1 Production

2.1.1 ASP.NET Core Runtimes

- Download & install ASP.NET Core Runtime 3.1

> **Information**
>
> On Windows, it is recommended installing the **Hosting Bundle**, which includes the .NET Core Runtime and IIS support.

> **Warning**
>
> IIS Server **must be running** when the ASP.NET Core Hosting Bundle is installed.
>
> Restart IIS after the Hosting Bundle installation is finished.

2.1.2 Git Bash & Github

1. Download & install Git bash
   Follow the instructions under Auto-launching ssh-agent on Git for Windows to auto start the SSH agent every time Git bash is starting.
2. Install Github SSH keys for the deployment repository `CERTHB2BPublish`
   1. Copy SSH key files that will be provided (`certhb2bpublish_deploy` and `certhb2bpublish_deploy.pub`) into `~/.ssh` directory. If the `~/.ssh` directory does not exist create it.
   2. Create a `~/.ssh/config` file with the following contents

```
Host github.com
User git
Hostname github.com
IdentityFile ~/.ssh/certhb2bpublish_deploy
```

2.1.3 Application Repository Initialization

- Open Git bash and clone the `CERTHB2BPublish` repository to `C:\inetpub\CERTHB2B` using the following commands

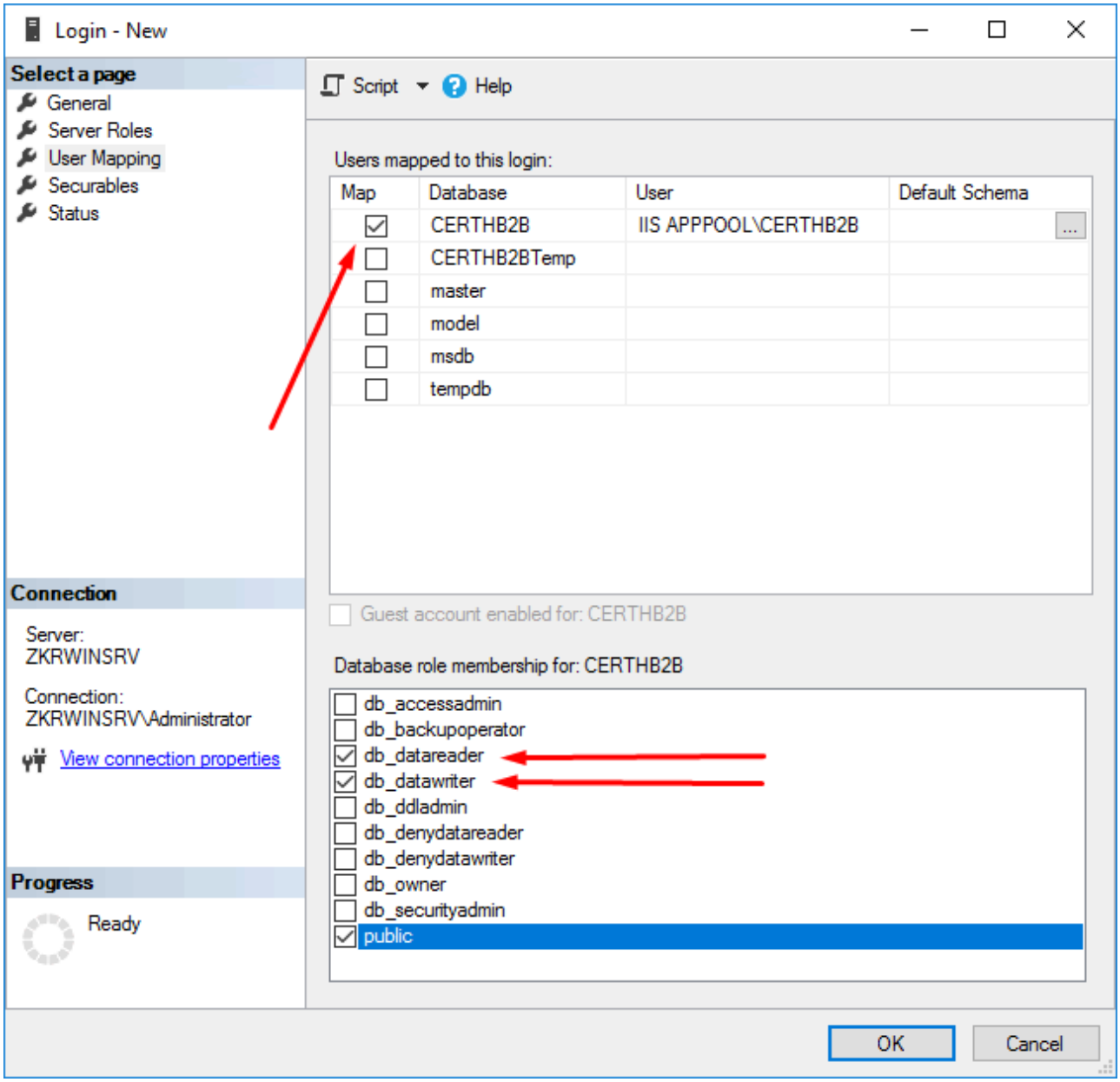
```
cd /c/initpub
git clone git@github.com:nekdev/CERTHB2BPublish.git CERTHB2B
```

2.1.4 IIS Site Creation

- Add a new IIS Application Pool and name it `CERTHB2B` with latest *.NET CLR version* and *Managed pipeline mode* set to `Integrated`
- Create an IIS site with name `CERTHB2B`, application pool the pool you created at the previous step, physical path `C:\inetpub\CERTHB2B` and binding for both HTTP (*port 80*) and HTTPS (*port 443*) ports
- Select the IIS site and open *Modules*; find and remove *WebDAVModule*

2.1.5 SQL Server database

1. Open SQL Server Management Studio (*install it if it's not already installed*) and create a database with name `CERTHB2B`
2. Create SQL Server login and apply it to the database
   Expand *Security* ➜ *Logins* (*not on the database, but under server*), right-click to *Logins* and click on *New Login...*. For `Login name` set `IIS APPPOOL\CERTHB2B` and under *User Mapping* page, check the newly created database and down at *Database role membership* check `public`, `db_datareader` and `db_datawriter`:



> ⚠ **Warning**
>
> The login name `CERTHB2B` in `IIS APPPOOL\CERTHB2B` **must be the same** as the IIS site name we created in the previous step.

3. Add connection string inside application root `appsettings.Production.json` file (*create the file if it does not exist*):

```
"ConnectionStrings": {
  "DefaultConnection":
"Server=GREENMINDB2B;Database=CERTHB2B;Trusted_Connection=True;MultipleActiv
eResultSets=true"
},
```

4. Initial Migrations
   Open Windows PowerShell (*as Administrator*), go to the application directory and run the application DLL with the `--Migrate all` CLI argument to apply initial migrations:

```
Set-Location -Path C:\inetpub\CERTHB2B\
dotnet CERTHB2B.dll --Migrate all
```

2.1.6 Identity Server Certificate

Use Microsoft Management Console (*MMC*) to import the Identity Server Certificate that will be provided.

1. Follow the instructions under View certificates with the MMC snap-in
2. Select **Run** from the **Start** menu, and then enter `certlm.msc`
3. Navigate to *Personal* → *Certificates* pane, right click within the *Certificates* panel and click *All Tasks* → *Import...* to start the Certificate Import Wizard
4. Follow the Wizard and select the `PFX` certificate file that will be provided
5. Enter or copy/paste the certificate password provided when asked and **make sure to check** *Mark this key as exportable. This will allow you to back up or transport your keys at a later time.*
6. When asked for *Certificate Store*, select *Place all certificates in the following store* with store name set to `Personal`
7. Click *Next* and *Finish* and you should get a successful message indicating the certificate is now imported. You can check the certificate under *Personal* → *Certificates* with the `CN` subject set to `CERTHB2BIdentityServerSPA`
8. Right click to the certificate *All Tasks* → *Manage Private Keys...* and add *IIS_IUSRS* with *Full control* and *Read* permissions

2.1.7 **Outgoing Emails**

The application is using SendGrid cloud API to send emails regarding:

- New user's registrations email confirmation
- User's reset password confirmation
- Profile to profile contact requests

Email settings are configured inside application root `appsettings.Production.json` file with the following properties:

- `SendGridUser`: The username of the SendGrid account
- `SendGridKey`: The API key created in SendGrid panel for the application
- `EmailSendFrom`: The email that users will see as "*From email*" field (*noreply@b2bplatform.imet.gr*)
- `EmailSendAs`: The name that users will see as "*From name*" field (*IMET B2B*)

2.2 Development

2.2.1 User Secrets

1. Init user secrets under the project directory
   `dotnet user-secrets init`

2. Add items to user secrets
   `dotnet user-secrets set SendGridUser <user>`
   `dotnet u1er-secrets set SendGridKey <key>`

3. List items in users secrets
   `dotnet user-secrets list`

3. Updates

Updates are applied using the release `CERTHB2BPublish` repository after new commits.

1. Open IIS manager and stop the web site
2. Open Git bash and pull new commits to `C:\inetpub\CERTHB2B` using the following commands:

```
cd /c/initpub/CERTHB2B
git pull
```

3. Open `powershell`, navigate to `C:\inetpub\CERTHB2B` and apply any new migrations:

```
PS C:\Users\Administrator> cd C:\inetpub\CERTHB2B
PS C:\inetpub\CERTHB2B> dotnet CERTHB2B.dll --Migrate all
```

4. Finally start the site again inside IIS manager