

2 Report za letný semester

2.1 Úvod

V nadväznosti na zimný semester bol projekt **EnergyProject** počas letného semestra rozšírený o pokročilé funkcionality. Dôraz sa kládol najmä na bezpečnosť, monitorovanie spotreby v reálnom čase, riadenie prístupu na základe rolí a vylepšenie používateľského rozhrania prostredníctvom analytiky a grafov.

2.2 Ciele semestra

Hlavné ciele pre druhý semester zahŕňali:

- **Overovanie a autorizácia:** nasadenie ASP.NET Core Identity a rozdelenie aplikácie pre roly Klient a Zamestnanec (Admin).
- **Monitorovanie v reálnom čase:** získavanie odpočtov z meračov a ich živá aktualizácia na webovom dashboarde.
- **Bezpečnosť a logovanie:** ošetrovanie vstupov, logovanie systémových udalostí a vynútenie bezpečnej komunikácie.
- **Vylepšenia UI a analytiky:** grafické znázornenie spotreby a tvorba prehľadných štatistík pre koncových používateľov.
- **Testovanie a stabilizácia:** odladenie chýb, oprava kaskádového mazania a stabilizácia databázových operácií.

2.3 Implementované časti

Overovanie a autorizácia (Identity & RBAC)

- **ASP.NET Core Identity:** Kompletná integrácia Identity pre správu používateľov (registrácia, prihlásenie, správa hesiel a osobných údajov).
- **Roly (Roles):** Zavedenie dvoch základných rolí – Admin (Zamestnanec) a Client (Klient).
- **Riadenie prístupu:** Obmedzenie prístupu k Admin Area (`\[Authorize(Policy = "AdminOnly")\]`) a Client Area (`\[Authorize(Policy = "ClientOnly")\]`). Logika zobrazenia navigačného menu je podmienená aktuálnou rolou používateľa.

Modul monitorovania v reálnom čase

- **SignalR namiesto gRPC:** Na prenos dát v reálnom čase zo servera na klientsky dashboard bola zvolená technológia **SignalR** vďaka jej výbornej integrácii do ASP.NET Core prostredia a podpore WebSockets.
- **Ukladanie dát:** Na pozadí beží `MeterBackgroundService`, ktorý v pravidelných intervaloch simuluje príchod nových odpočtov z meračov. Tieto odpočty sa okamžite ukladajú do databázy.
- **Živé aktualizácie:** Dashboard na strane klienta (`ViewMeterReadings`) využíva SignalR klienta na okamžité zobrazenie nového odpočtu (status indikátor, posledná hodnota) a prekreslenie grafu bez potreby obnovovania stránky.

Bezpečnosť a logovanie

- **Validácia vstupov:** Striktné overovanie na strane servera (Data Annotations) a ošetrovanie na strane klienta (pomocou `_ValidationScriptsPartial` a `jQuery Validation`) pre zabránenie chybných formulárov a zlepšenie UX.
- **Logovanie (Serilog):** Nasadenie knižnice Serilog pre štruktúrované logovanie kritických operácií, chýb a informačných správ do konzoly a textového súboru (`log.txt`).
- **Soft Delete:** Ochranné opatrenie pre finančné dáta. Odstránenie kreditnej karty nezmaže záznam fyzicky (aby zostali zachované historické väzby pre platby/faktúry), ale zmení stav `IsActive` na `false`. Zároveň bol upravený dizajn databázových väzieb na zabránenie nechceného kaskádového zmazania dôležitých entít.

Vylepšenia UI a analytiky

- **Grafy spotreby:** Využitie knižnice `Chart.js` pre vizualizáciu historickej spotreby. Používateľ má možnosť filtrovať zobrazenie (denné, mesačné, ročné).
- **Prehľadné štatistiky:** Zavedenie moderných KPI (Key Performance Indicator) kariet v klientskej časti, ktoré zobrazujú celkovú dlžnú sumu, uhradené sumy a stav merača ("Live", "Offline").
- **Opravy a čistý kód:** Opravy chýb v štruktúre HTML tabuliek, asynchrónne optimalizácie v repozitároch a ošetrovanie "null" výnimiek vo frontend kóde.

2.4 Záver

Počas druhého (letného) semestra sa prototyp vyvinul na bezpečnú, stabilnú a interaktívnu aplikáciu. Používatelia teraz majú prístup do svojho účtu chráneného pomocou Identity. Odborné miesta prijímajú a vizualizujú dáta v reálnom čase pomocou technológie SignalR, čo dodáva aplikácii moderný a profesionálny vzhľad. Vďaka prepracovanému logovaniu, soft-delete logike a vylepšenému grafickému rozhraniu systém spĺňa všetky požiadavky kladené na záverečnú fázu projektu.

Dodatkové informácie

Testovacie dáta a prístup Pri prvom spustení a aplikovaní migrácií sa databáza automaticky naplní testovacími dátami (tarify, adresy, merače, faktúry). Pre otestovanie aplikácie môžete použiť predpripravené účty:

- **Admin:** email: `admin@gmail.com`, heslo: `1Qwer$78`
- **Klient:** email: `client@gmail.com`, heslo: `1Qwer$78`

Ako spustiť projekt

- Nainštalovať .NET SDK 8.
- Príprava databázy: Na OS Windows aplikácia využíva predvolený SQL Server (LocalDB). Keďže LocalDB nie je podporovaná na Linux/macOS, najjednoduchšou cestou je spustiť MS SQL Server pomocou Dockeru a následne upraviť connection string v `appsettings.json`.
- Otvoriť riešenie `EnergyProject.sln` vo Visual Studio / Rider / VS Code.
- V koreňovom priečinku projektu spustiť migrácie: `dotnet ef database update` (prípadne vo VS: Update-Database).
- Spustiť aplikáciu: `dotnet run --project EnergyProject/EnergyProject` alebo tlačidlom Run v IDE.
- Po spustení sa aplikácia otvorí v prehliadači (napr. <https://localhost:7073> alebo <http://localhost:5222> podľa profilu).