

GYMNASIUM JANA KEPLERA

Parléřova 2/118, 169 00 Praha 6



Synchronizace účtů na škole

Maturitní práce

Autor: Erik Stoklasa

Třída: 4.B

Školní rok: 2021/2022

Předmět: Informatika

Vedoucí práce: Emil Miler

Praha, 2022



GYMNASIUM JANA KEPLERA
Kabinet informatiky

ZADÁNÍ MATURITNÍ PRÁCE

Student: Erik Stoklasa

Třída: 4.B

Školní rok: 2021/2022

Platnost zadání: 30. 9. 2022

Vedoucí práce: Emil Miler

Název práce: Synchronizace účtů na škole

Pokyny pro vypracování:

Cílem práce je vytvořit software na synchronizaci uživatelů napříč různými školními systémy. Software by měl centralizovaně spravovat systémy školní knihovny, tiskárny, vstupní dveře a systém Bakaláři.

URL repozitáře:

<https://github.com/erikstoklasa/SchoolAccountSync>

student

vedoucí práce

V Praze dne 24. 3. 2022

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů. Nemám žádné námitky proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Praze dne 25. března 2022

Erik Stoklasa

Poděkování

Chtěl bych poděkovat Emilu Milerovi za všechny konzultace, které mi ochotně poskytl. Zároveň bych chtěl poděkovat Františku Šimordovi za představení všech školních systémů.

Abstrakt

Maturitní projekt synchronizace školních systémů kopírek, Bakalářů, knihovny a vstupů.

Klíčová slova

synchronizace, školní systémy, databáze, .NET Core, PostgreSQL, MSSQL, Firebird

Abstract

Maturita project dealing with synchronization of school systems ranging from copiers, Bakalari system to library and entrance systems.

Keywords

synchronization, school management systems, database, .NET Core, PostgreSQL, MSSQL, Firebird

Obsah

1	Teoretická část	3
2	Implementace	5
2.1	.NET	5
2.2	Rešerše	5
2.3	UX návrh	5
2.4	Napojení na systém tiskáren	5
2.5	Napojení na systém Bakaláři	6
2.6	Napojení na systém knihovny	6
2.7	Napojení na systém vstupních dveří	6
2.8	Budoucí vylepšení	6
3	Technická dokumentace	7
	Závěr	9
	Seznam použité literatury	11

1. Teoretická část

Na Gymnáziu Jana Keplera se nachází mnoho školních systémů, které se starají o chod školy. Od knihovny, přes tiskárny, až po systém, který se stará o to, aby se do budovy dostali jen ti povolaní, tedy vstupní systém. Školní administrátor pak pro každého studenta musí vytvořit a spravovat účet v daném systému. Toto by nebylo problém, pokud by na škole bylo šedesát studentů, každopádně při škole, která je domovem pro přibližně 600 studentů je nutnost pracovat s vícero systémy velice časově náročná. Cílem této maturitní práce je vytvořit aplikaci, která tyto systémy spravuje a školní administrátor pak nemusí ztrácet svůj čas rutinní prací.

2. Implementace

Projekt byl zpracován v .NET 6, což je open source framework pro vytváření programů, které lze spustit na operačních systémech Windows, Linux a macOS. Byl použit programovací model Razor pages, který je alternativou ke klasickému MVC. Zároveň je celý projekt zabalený v kontaineru díky platformě Docker. Pro lokální databázi jsem zvolil PostgreSQL.

2.1 .NET

Framework .NET jsem si pro vypracování vybral, jelikož je oproti alternativám jako je třeba Node.js mnohem rychlejší a zároveň není závislý na platformě, na které bude spuštěn, což byla jedna z neznámých, při zakládání projektu. V této platformě se také dobře pracuje s daty díky LINQ, který jsem často používal. Zároveň na všechny databáze, na které jsem se potřeboval připojit, existovaly drivery, které jsem mohl využít.

2.2 Rešerše

Tato část práce, musím bohužel říct, trvala přibližně stejně, ne-li více času, než celková implementace. Vyznat se v hordách tabulek školních systémů, na které se musím připojit, a které jsou zároveň pojmenované naprosto nesmyslnými názvy (ACCARD, AULEA, SYS0901, SYS0902, atd.) bylo pro mě naprosto ubíjející. Zároveň jsem měl ale naprostou radost, jakmile jsem po desítkách hodin studování různých databázových tabulek přišel, jak opravdu fungují. Vše bylo ale nutné udělat ještě před tím, než začnu implementovat celý projekt.

2.3 UX návrh

Po rešerši jsem se pustil do designového návrhu projektu. Pracoval jsem v programu Figma, kde jsem si postupně vytvořil stránky dashboardu, konkrétního zobrazení studentů nebo zobrazení změn uživatelů. Často mi také pomáhalo si před implementací kódu jen ve Figmě načrtnout, jak by daná funkce měla vypadat, implementace byla pak jednodušší a rychlejší.

2.4 Napojení na systém tiskáren

Tiskárny na Gymnáziu Jana Keplera fungují na databázovém serveru PostgreSQL. Pro napojení k tomuto serveru jsem využil knihovny Npgsql. Zároveň musím zmínit, že i přes velký počet tabulek v databázi (asi 150) dávalo smysl jejich pojmenování a tím pádem bylo pochopení struktury databáze mnohem jednodušší, než u ostatních školních systémů.

2.5 Napojení na systém Bakaláři

Ze systému Bakaláři stahuji data, je to tedy takový centrální systém pravdy. Pokud se něco v systému Bakaláři změní, bere se to za pravdivý údaj. Struktura tabulek v tomto systému byla také poměrně složitá, ale alespoň byly poměrně smyslně pojmenovány.

2.6 Napojení na systém knihovny

Připojení na systém knihoven mi trvalo zprovoznit poměrně dlouho, jelikož se v průběhu mojí implementace změnil název databáze a zároveň se ze strany dodavatelské firmy systém znepřístupnil určitě tabulky. Bylo tedy možné naimplementovat jen jednoduchý způsob synchronizace se systémem, ale nebylo možné provést plnou synchronizaci.

2.7 Napojení na systém vstupních dveří

Asi nejstarší a také nejsložitější systém, se kterým jsem musel kdy pracovat. Stovky tabulek s nepsmylnými názvy, které mezi sebou nemají ani žádné vztahy, obsahují duplikáty různých údajů na spoustě míst, tedy systém, se kterým je opravdu složité pracovat. Nakonec se mi povedlo zprovoznit jednoduchou synchronizaci, která dokáže aktualizovat RFID na základě lokálních dat.

2.8 Budoucí vylepšení

Potenciálních vylepšení v systému vidím opravdu hodně, například přidání autentifikace, tak aby mohla aplikace běžet na serveru a nejen lokálně na jednom počítači. Dalším z vylepšení je možné hledání kolizí v RFID záznamech, systém vás sice upozorní, jakmile se budete snažit synchronizovat s duplikovaným RFID, ale bylo by lepší uživatele upozornit ještě předtím.

3. Technická dokumentace

Pro spuštění projektu je nutné vytvořit si PostgreSQL server, na kterém si musíte vytvořit tabulku dle souboru dbinit.sql. Dále je nutné spustit projekt s příkazem docker run. Přidáme environment variables, jak uvádí README soubor a aplikace je spuštěná.

Závěr

Projekt splnil zadání v plné míře, dokáže spravovat dané systémy, i když bylo těžké s nimi někdy pracovat. V rámci projektu jsem se toho naučil opravdu hodně, hlavně s různými typy databází a zpracování dat.

Seznam použité literatury

- [Cor] Tim Corey. *Intro to AppSettings in .NET Core - Appsettings.json, secrets.json, and more.* URL: https://www.youtube.com/watch?v=_2_qksdQKCE.
- [Tea] The Npgsql Development Team. *Documentation Npgsql.* URL: <https://www.npgsql.org/doc/>.