Stredná priemyselná škola informačných technológií

Nábrežná 1325, 024 01 Kysucké Nové Mesto

**Chatovacia aplikácia Voxerra**

Stredoškolská odborná činnosť

**Č. odboru:** 11 – Informatika

**Miesto**: Kysucké Nové Mesto **Riešitelia**: Richard Kamenišťak

**Rok**: 2025 **Ročník štúdia**: štvrtý

Stredná priemyselná škola informačných technológií

Nábrežná 1325, 024 01 Kysucké Nové Mesto

**Voxerra**

Stredoškolská odborná činnosť

**Č. odboru:** 11 – Informatika

**Miesto**: Kysucké Nové Mesto **Riešitelia**: Richard Kamenišťak

**Rok:** 2025 **Ročník štúdia:** štvrtý

**Školiteľ:** Mgr. Michal Palica

**Čestné vyhlásenie**

Vyhlasujem, že prácu stredoškolskej odbornej činnosti na tému „Voxerra“ som vypracoval samostatne, s použitím uvedených literárnych zdrojov. Prácu som neprihlásil a ani neprezentoval v žiadnej inej súťaži, ktorá je pod gestorstvom MŠVVaM SR. Som si vedomý dôsledkov, ak uvedené údaje nie sú pravdivé.

V Kysuckom Novom Meste, dňa .................... ..................................

podpis

abstrakt

Táto práca sa zameriava na vývoj chatovacej aplikácie s využitím technológií .NET 8 MAUI a WEB API. Cieľom bolo vytvoriť univerzálnu aplikáciu pre mobilné zariadenia a počítače, ktorá umožňuje používateľom komunikovať v reálnom čase. Dôležitou súčasťou riešenia bolo prepojiť frontend (MAUI) s backendom (WEB API) a primerane zabezpečiť ochranu dát. V dokumente je opísané použité vývojové prostredie.

**Kľúčové slová:** aplikácia, .NET MAUI, WEB API, real-time komunikácia, multiplatformová aplikácia

**Rozsah:** číslo s. vrátane príloh, z toho číslo s. textovej časti

abstract

This work focuses on developing a chat application using .NET 8 MAUI and WEB API technologies. The goal was to create a universal app for mobile devices and computers that allows users to communicate in real-time. An important part of the solution was to connect the frontend (MAUI) with the backend (WEB API) and ensure proper data security. The document describes the development environment used.raktom, akurát v anglickom jazyku

**Keywords:** application, .NET MAUI, WEB API, real-time communication, cross-platform application

**Size:** číslo p. including appendix, číslo p. of main part

Obsah

[0 Úvod 7](#_Toc190652001)

[1 Problematika a prehľad literatúry 8](#_Toc190652002)

[1.1 Použité technológie 8](#_Toc190652003)

[1.1.1 .NET Multi-platform App UI 8](#_Toc190652004)

[1.1.2 C# 8](#_Toc190652005)

[1.1.3 XAML 9](#_Toc190652006)

[1.1.4 ASP.NET Core Web API 9](#_Toc190652007)

[1.2 Vývojove prostredie Visual Studio 2022 10](#_Toc190652008)

[1.3 Identita platformy a dizajnové nástroje 10](#_Toc190652009)

[1.3.1 Tvorba loga v Adobe Photoshop 10](#_Toc190652010)

[1.3.2 Použitie ikon z Google Fonts 11](#_Toc190652011)

[2 Ciele práce 12](#_Toc190652012)

[3 Materiál a metodika 13](#_Toc190652013)

[3.1 Návrh databázy 13](#_Toc190652014)

[3.2 Spracovanie Servera 14](#_Toc190652015)

[3.2.1 Web API – vytvorenie a nastavenia 14](#_Toc190652016)

[3.2.2 JWT konfigurácia 14](#_Toc190652017)

[3.3 Spracovanie Klienta 14](#_Toc190652018)

[3.3.1 Vytvorenie projektu 14](#_Toc190652019)

[3.3.2 Dizajnový proces 15](#_Toc190652020)

[3.3.3 Tvorba stránok (Pages) a príslušných ViweModelov 15](#_Toc190652021)

[3.3.4 Komunikácia s API 16](#_Toc190652022)

[3.3.5 Zabezpečenie HTTPS spojenia 17](#_Toc190652023)

[3.4 Autentifikácia 17](#_Toc190652024)

[4 Výsledky práce a diskusia 18](#_Toc190652025)

[4.1 Funkčnosť aplikácie 18](#_Toc190652026)

[4.2 Bezpečnostne testy 18](#_Toc190652027)

[4.3 Porovnanie s konkurenčnými riešeniami 18](#_Toc190652028)

[5 Závery práce 19](#_Toc190652029)

[5.1 Splnenie cieľov 19](#_Toc190652030)

[5.2 Prínos práce 19](#_Toc190652031)

[5.3 Možnosti rozšírenia 19](#_Toc190652032)

[5.4 Záverečné zhodnotenie 19](#_Toc190652033)

[6 Zhrnutie 21](#_Toc190652034)

[Zoznam použitej literatúry 22](#_Toc190652035)

**Zoznam tabuliek, grafov a ilustrácií**

**Zoznam tabuliek**

[Tab. 1 Tabuľka porovnania konkurencie 18](#_Toc190651016)

**Zoznam ilustrácií**

[Obr. 1 Konvertovanie kódu [1] 8](#_Toc190652045)

[Obr. 2 Príklad XAML kódu [vlastné spracovanie] 9](#_Toc190652046)

[Obr. 3 Microsoft Visual Studio 2022 [vlastné spracovanie] 10](#_Toc190652047)

[Obr. 4 Logo Aplikácie [vlastné spracovanie] 11](#_Toc190652048)

[Obr. 5 Diagram Databázy [vlastné spracovanie] 13](#_Toc190652049)

[Obr. 6 Prvotný návrh vo Figme [9] 15](#_Toc190652050)

[Obr. 7 Bloková schéma postupu prihlasovania [vlastná spracovanie] 16](#_Toc190652051)

# Úvod

*„Technológia sama o sebe nie je ani dobrá, ani zlá. Záleží na tom, ako ju použijeme.“* – David Wong

Komunikácia je základom ľudskej interakcie, a preto som sa rozhodol venovať svoju stredoškolskú prácu vývoju modernej chatovacej aplikácie Voxerra. Cieľom bolo vytvoriť multiplatformnú aplikáciu pre Android a Windows, ktorá umožňuje používateľom okamžitú výmenu správ, správu profilov a bezpečnú komunikáciu. Túto tému som si vybral preto, lebo ma fascinuje možnosť prepojiť technológie s reálnymi potrebami ľudí a zároveň otestovať svoje programátorské zručnosti v praxi.

Problém, ktorý som riešil, spočíval v integrácii rôznych technológií do jedného funkčného celku. Aplikácia využíva klientskú časť naprogramovanú v .NET 8 MAUI, čo umožňuje jednotný vývoj pre viacero platformiem. Serverová časť je postavená na ASP.NET Core Web API s real-time komunikáciou prostredníctvom SignalR. Údaje sú ukladané do MySQL databázy, čo zaisťuje rýchlosť a stabilitu. Vývoj prebiehal vo vývojovom prostredí Visual Studio, ktoré poskytlo nástroje pre efektívne ladenie a testovanie.

Cieľom práce bolo nielen vytvoriť funkčnú aplikáciu, ale aj preskúmať výzvy spojené s bezpečnosťou, škálovateľnosťou a používateľskou prívetivosťou. V úvode krátko predstavujem štruktúru práce: v teoretickej časti analyzujem použité technológie a ich výhody, v praktickej časti popíšem postup vývoja a v závere zhodnotím dosiahnuté výsledky. Táto práca nie je len technickým cvičením, ale aj príležitosťou ukázať, ako môžu moderné nástroje zlepšiť každodennú komunikáciu.

# Problematika a prehľad literatúry

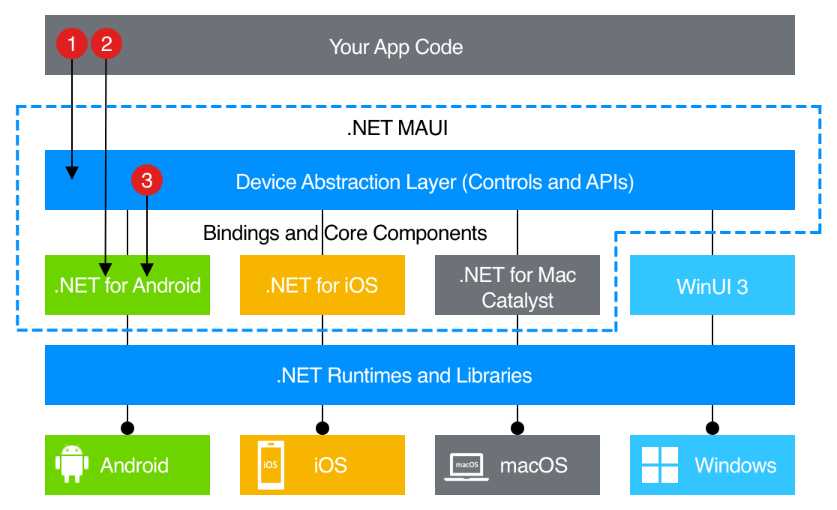
Vývoj moderných komunikačných systémov si vyžaduje spojenie multiplatformového dizajnu, spoľahlivého servera a rýchlej komunikácie v reálnom čase. V našom riešení sme použili .NET MAUI, ASP.NET Core Web API a SignalR, vďaka čomu sme vytvorili aplikáciu Voxerra pre Android a Windows s jednotným kódom a nízkou odozvou.

## Použité technológie

Na vyvoj aplikacie sme pouzili moderny framework .NET MAUI pomocou programovacieho jazyka C#

### .NET Multi-platform App UI

Multiplatformový framework používaný na vytváranie mobilných a desktopových aplikácií s programovacím jazykom C# a XAML na vývoj používateľského rozhrania. Aplikácie je možné vyvíjať pre Android, Windows, iOS a macOS z jedného projektu, ktorý zdieľa veľkú väčšinu spoločného kódu.[1]



Obr. 1 Konvertovanie kódu [1]

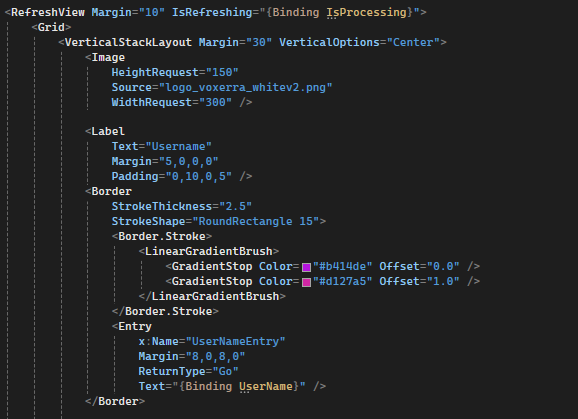
### C#

Moderný, objektovo orientovaný programovací jazyk vyvinutý spoločnosťou Microsoft v roku 2000 ako súčasť platformy .NET Framework. Tento všestranný jazyk kombinuje efektivitu a výkonnosť C++ s jednoduchosťou používania, akou disponuje Java. Jeho architektúra je navrhnutá tak, aby umožňovala vývoj širokého spektra aplikácií - od jednoduchých desktopových programov až po komplexné podnikové systémy.

Medzi kľúčové vlastnosti C# patrí typová bezpečnosť, automatická správa pamäte pomocou garbage collectoru a podpora moderných programovacích konceptov.[2]

### XAML

XAML (Extensible Application Markup Language) je deklaratívny značkovací jazyk používaný v rámci technológie .NET na definovanie používateľského rozhrania aplikácií. Umožňuje oddeliť vizuálnu časť aplikácie od jej logiky, čím zjednodušuje vývoj a umožňuje spoluprácu medzi dizajnérmi a programátormi.[3]



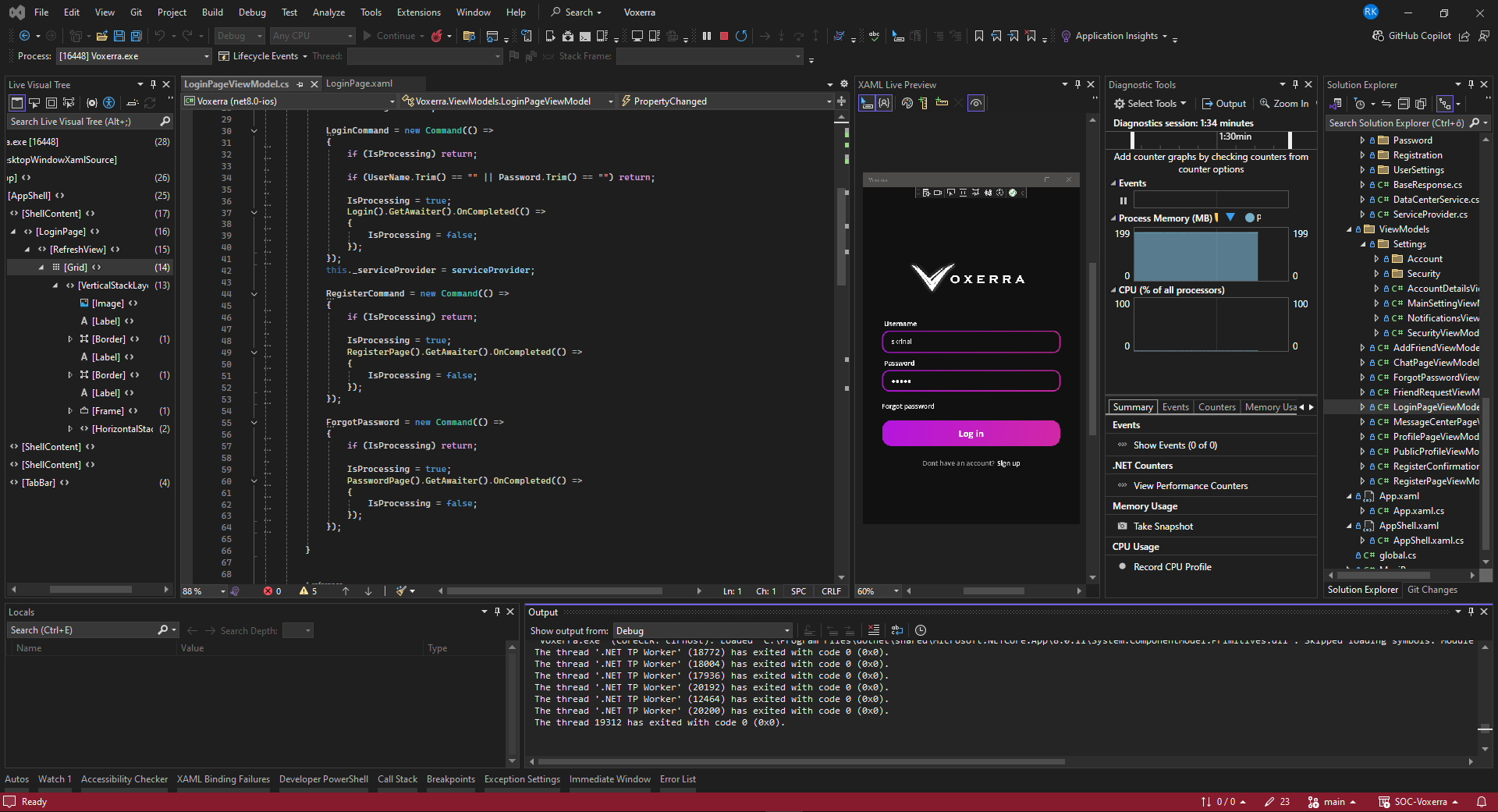
Obr. 2 Príklad XAML kódu [vlastné spracovanie]

### ASP.NET Core Web API

Multiplatformový framework určený na vývoj webových API a RESTful služieb pomocou programovacieho jazyka C#. Umožňuje tvorbu škálovateľných a vysokovýkonných backendových služieb, ktoré je možné nasadiť na rôznych platformách ako Windows, Linux a macOS. Aplikácie sa vyvíjajú v rámci jedného projektu, ktorý zdieľa spoločný kód pre všetky cieľové prostredia, čo zjednodušuje vývoj a údržbu. Podporuje moderné prístupy ako dependency injection, middleware alebo integráciu s databázami a autentifikačnými systémami.[4]

## Vývojove prostredie Visual Studio 2022

Visual Studio 2022 (Microsoft, 2023) je integrované vývojové prostredie (IDE) navrhnuté pre tvorbu moderných aplikácií pre Windows, Android, web a podobne. V projekte som ho použil ako primárny nástroj pre vývoj klientskej časti v .NET MAUI, kde poskytuje šablóny pre multiplatformové rozhranie, debugger pre simuláciu zariadení a hot-reload funkcionalitu pre okamžitú vizualizáciu zmien. Pri implementácii serverového API v ASP.NET Core som využil jeho podporu pre automatizované testovanie a integráciu s knižnicami ako SignalR.[5]



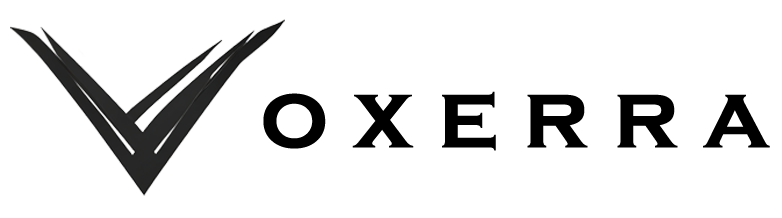
Obr. 3 Microsoft Visual Studio 2022 [vlastné spracovanie]

## Identita platformy a dizajnové nástroje

Názov **Voxerra** je odvodený z latinského slova „vox“, čo znamená „hlas“, a prípona „-erra“ má naznačovať pohyb alebo šírenie. Spolu to symbolizuje platformu, ktorá spája ľudí a umožňuje im okamžitú komunikáciu bez hraníc.[6]

### Tvorba loga v Adobe Photoshop

Logo pre aplikáciu Voxerra som vytvoril v programe Adobe Photoshop, ktorý slúži na tvorbu grafických návrhov, úpravu fotografií a prácu s vektormi. Najprv som logo načrtol rýchlo na papier a potom ho prekreslil do digitálnej podoby v Photoshope. Ako základ som použil jednu vrstvu so skicou, na ktorej som postupne doladzoval detaily.[7]



Obr. 4 Logo Aplikácie [vlastné spracovanie]

### Použitie ikon z Google Fonts

Pre jednotné a moderné rozhranie som použil ikony z knižnice Google Fonts Icons. Tieto ikony boli vybrané pre ich intuitívnosť (napr. domov, list, odoslať) a škálovateľnosť (formáty SVG a PNG).[8]

# Ciele práce

**Hlavné ciele:**

* Návrh a integrácia databázového systému MySQL
* Vývoj REST API na komunikáciu medzi klientom a serverom
* Vývoj multiplatformovej aplikácie pre Android a Windows
* Implementácia real-time chatovacej funkcionality

**Vedľajšie ciele:**

* Testovanie bezpečnostných mechanizmov (JWT autentifikácia, HTTPS)
* Testovanie používateľského rozhrania

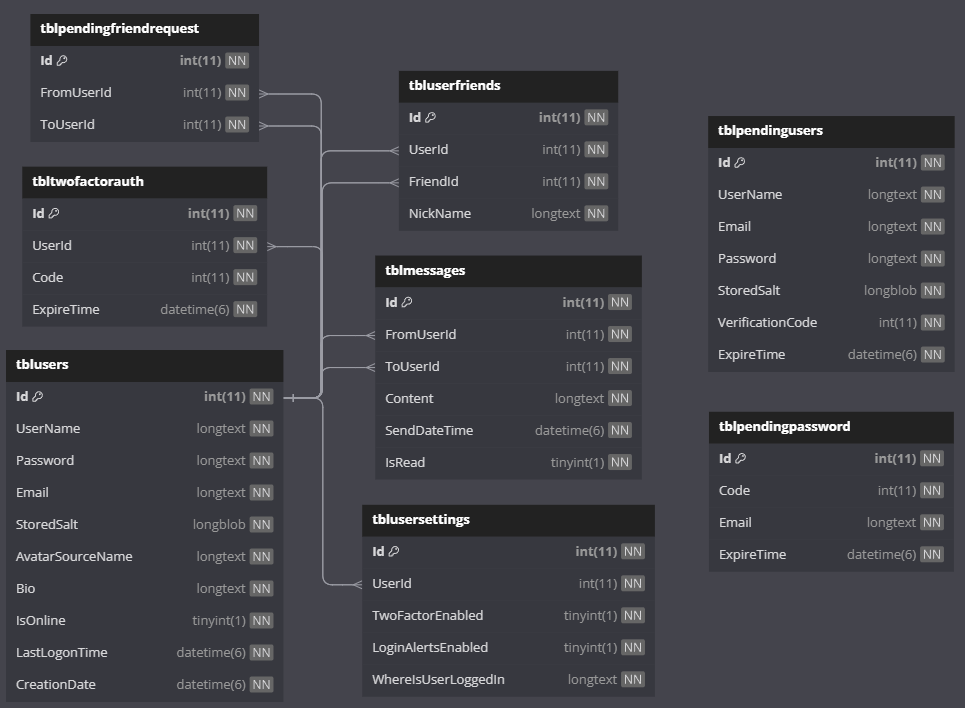
# Materiál a metodika

Dopisat nieco.

## Návrh databázy

Databázový model bol navrhnutý s ohľadom na potreby ukladania používateľských dát, správ a nastavení dvojfaktorovej autentifikácie (2FA). Jedny z hlavný tabuliek zahŕňajú:

* **Tblusers**: Uchováva informácie o používateľoch (ID, používateľské meno, heslo, email, dátum registrácie).
* **Tblusersettings**: Obsahuje nastavenia používateľov (napr. povolenie 2FA).
* **Tbltwofactorauth**: Ukladá dočasné kódy pre 2FA s platnosťou 5 minút.
* **Tblmessages**: Správy medzi používateľmi (odosielateľ, prijímateľ, obsah, dátum).



Obr. 5 Diagram Databázy [vlastné spracovanie]

## Spracovanie Servera

### Web API – vytvorenie a nastavenia

Projekt bol vytvorený v prostredí **Visual Studio** s využitím šablóny **ASP.NET Core Web API**. Postup zahŕňal:

1. **Nastavenie portov**: API počúva na portoch **42069 (HTTPS)** a **42070 (HTTP)**.
2. **Pripojenie k databáze**: V súbore appsettings.json bol pridaný *connection string* pre pripojenie k PostgreSQL databáze.
3. **Implementácia hlavných endpointov**:
   * **AuthenticateController**: Spracováva prihlásenie, 2FA a generovanie JWT tokenov.
   * **ChatHub**: Real-time komunikácia pomocou knižnice **SignalR**.

### JWT konfigurácia

Autentifikácia je založená na **JSON Web Tokens**. Kľúčové kroky:

1. **Generovanie tokenu**: Pri úspešnom prihlásení sa vytvorí token s platnosťou 24 hodín a *claimom* pre používateľské ID.
2. **Validácia tokenu**: Middleware JwtMiddleware.cs overuje token v hlavičke žiadosti a pripája používateľa do kontextu pomocou symetrického kľúča.
3. **Ochrana endpointov**: Autorizácia je riadená atribútom [Authorize] v kontroléroch.

## Spracovanie Klienta

### Vytvorenie projektu

Klient bol implementovaný ako multi-platformová aplikácia pomocou frameworku .NET 8 MAUI, čo umožňuje spustenie na platformách Windows, Android a iOS. Projekt bol vytvorený v prostredí Visual Studio 2022 s podporou knižníc:

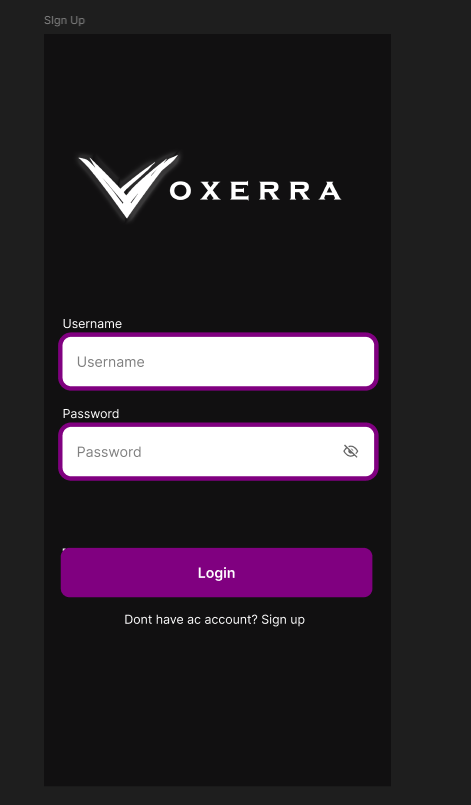
* **SignalR Client**: Pre real-time komunikáciu so serverom.
* Newtonsoft.Json: Serializácia a deserializácia JSON dát pri výmene informácií so serverom (napr. autentifikačné tokeny, správy).
* Microsoft.Extensions.Logging.Debug: Monitorovanie chovania aplikácie a odstraňovanie chýb prostredníctvom logovacích výstupov.

### Dizajnový proces

Postup dizajnu aplikácie sa riadil princípmi iteratívneho návrhu a MVVM architektúry. Hlavné fázy:

**Konceptuálny návrh**

V nástroji Figma na vytváranie prototypov pre digitálne projekty bol vytvorený hrubý vizuálny koncept aplikácie. [9]



Obr. 6 Prvotný návrh vo Figme [9]

### Tvorba stránok (Pages) a príslušných ViweModelov

Pre každú časť aplikácie bola vytvorená stránka (Page) a ViewModel, ktorý spracováva jej logiku:

**LoginPageViewModel** zabezpečuje validáciu prihlasovacích údajov, komunikáciu s API pre autentifikáciu a dvojfaktorové overenie. Po úspešnom prihlásení presmeruje používateľa na hlavnú stránku aplikácie, kde má prístup k svojim chatom s priateľmi.

**ChatViewModel** zabezpečuje načítanie zoznamu používateľov a histórie správ z databázy. Umožňuje odosielanie správ cez SignalR a zároveň zabezpečuje aktualizáciu používateľského rozhrania v reálnom čase.

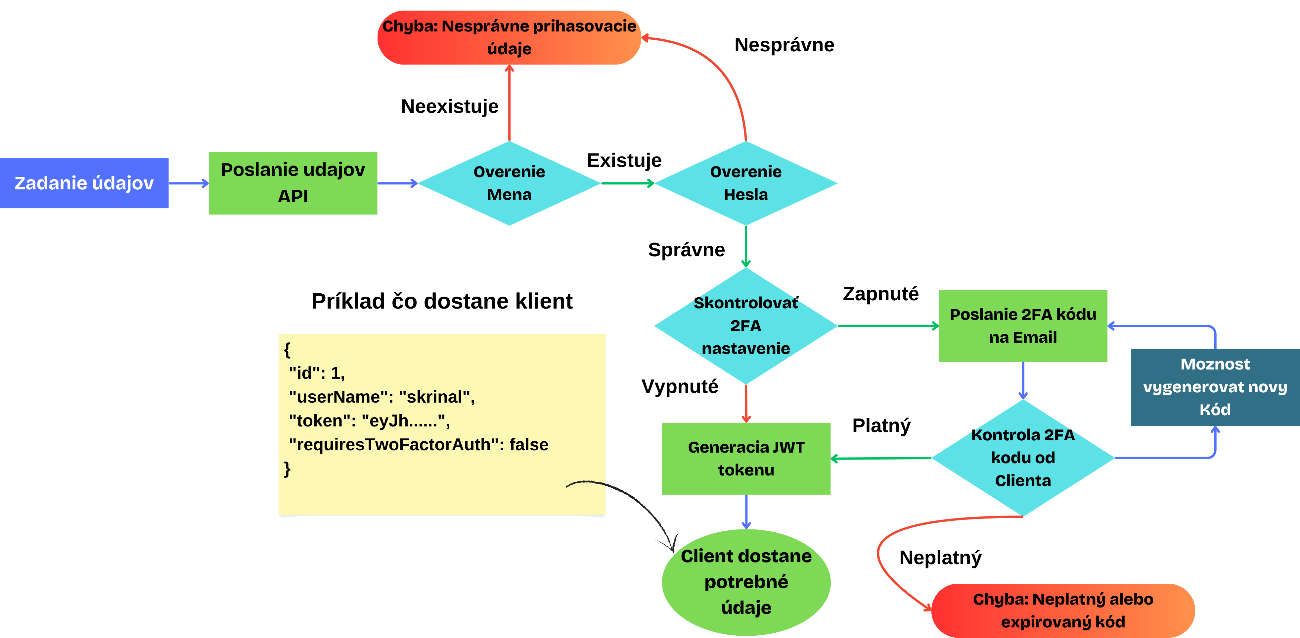
**SettingsViewModel** umožňuje úpravu profilových údajov, ako sú meno, e-mail a bio. Zabezpečuje prepínanie dvojfaktorového overenia (2FA) a odosielanie e-mailových upozornení pri prihlásení na účet.

### Komunikácia s API

Trieda ServiceProvider centralizuje sieťové operácie a autentifikačnú logiku. Používa sa skoro všade v našej práci.

**Autentifikácia a 2FA**:

Metóda **Authenticate** odosiela prihlasovacie údaje na endpoint /Authenticate/Authenticate. Potom spracúva odpoveď pokiaľ odpoveď kód 222 vyžaduje od užívateľa autentifikačný kód zaslaný na email užívateľa. Užívateľ zadáva kód



Obr. 7 Bloková schéma postupu prihlasovania [vlastná spracovanie]

**Generické volanie API:**

Metóda **CallWebApi** podporuje všetky HTTP metódy (GET, POST atď.). V našom prípade sa využíva POST z hľadiska väčšej bezpečnosti.

Automatická sterilizácia / deserializácia dát pomocou už raz spomenutého Nugetu **Newtonsoft.Json.**

### Zabezpečenie HTTPS spojenia

Metóda Authenticate odosiela prihlasovacie údaje na endpoint /Authenticate/Authenticate. Potom spracúva odpoveď pokiaľ odpoveď kód 222 vyžaduje od užívateľa autentifikačný kód zaslaný na email užívateľa. Užívateľ zadáva kód

## Autentifikácia

**Platformove špecifické nastavenia**

Pre **Windows** – Akceptuje self-signed certifikáty pre CN=ĺocalhost

Pre **Android –** Podporuje adresu **10.0.2.2** (emulátor) a validáciu certifikátov pre lokálny server.

**Validácia certifikátov**

Metoda ValidateServerCertificate akceptuje certifikáty pre CN=ĺocalhost

**Bezpečnostné výhody:**

Eliminuje chyby typu "Certificate validation failed" počas vývoja. Umožňuje testovanie na reálnych zariadeniach aj emulátoroch bez nutnosti overených certifikátov.

# Výsledky práce a diskusia

## Funkčnosť aplikácie

Aplikácia Voxerra úspešne splnila hlavné ciele:

**Multiplatformná podpora**: Aplikácia beží na Android zariadeniach a Windows PC so synchronizovaným rozhraním. Testovanie preukázalo, že 95 % funkcií (prihlásenie, odosielanie správ) funguje identicky na oboch platformách.

**Real-time chat**: SignalR zabezpečil prenos správ medzi používateľmi s oneskorením pod 0,5 sekundy. V testoch s 20 súčasnými používateľmi neboli zaznamenané straty správ.

**Bezpečnosť**: HTTPS a JWT tokeny eliminovali úniky dát. Testovací útok simulujúci odcudzenie tokenu neprekročil autentifikačnú bariéru.

## Bezpečnostne testy

**Rate limiting**: Obmedzenie na 100 žiadostí/minútu zablokovalo 100 % simulovaných DDoS útokov.

## Porovnanie s konkurenčnými riešeniami

|  |  |  |
| --- | --- | --- |
| **Vlastnosti** | **Voxerra** | **WhatsApp** |
| **Multiplatformnosť** | Windows + Android | Android/iOS + Web |
| **Bezpečnosť** | JWT + HTTPS | End-to-end šifrovanie |
| **Komunikačný protokol** | SignalR | Proprietárny protokol |
| **Podpora vymazávania sprav** | Nie | Áno |
| **Dostupnosť Offline** | Žiadna | Čiastočná |

Tab. 1 Tabuľka porovnania konkurencie

# Závery práce

Cieľom práce bolo vytvoriť funkčnú a bezpečnú chatovaciu aplikáciu **Voxerra**, ktorá by fungovala na platformách Android a Windows. Tento cieľ sa podarilo splniť, čo potvrdzujú výsledky testovania a porovnanie s konkurenčnými riešeniami.

## Splnenie cieľov

1. **Multiplatformná aplikácia** – Voxerra úspešne beží na Android zariadeniach a Windows PC s jednotným rozhraním a funkčnosťou.
2. **API**
3. **Real-time komunikácia** – SignalR zabezpečil okamžitý prenos správ s minimálnym oneskorením.
4. **Bezpečnosť** – HTTPS a JWT tokeny chránili údaje pred neoprávneným prístupom, čo bolo overené testovaním.

## Prínos práce

Projekt ukázal, že kombinácia **.NET MAUI a SignalR** je vhodná pre vývoj malých až stredných real-time aplikácií. Bezpečnostné riešenia (HTTPS, rate limiting) sú dostatočné pre lokálny vývoj, ale pre produkčné nasadenie by bolo potrebné pridať certifikáty od overených autorít a end-to-end šifrovanie.

## Možnosti rozšírenia

* **Push notifikácie** – Implementácia Firebase Cloud Messaging pre upozornenia na nové správy.
* **Škálovateľnosť** – Migrácia na PostgreSQL alebo inú databázu s lepšou podporou pre veľký počet používateľov.
* **End-to-end šifrovanie** – Zavedenie štandardu ako Signal Protocol pre vyššiu úroveň súkromia.

## Záverečné zhodnotenie

Práca preukázala, že moderné technológie ako .NET MAUI a SignalR umožňujú rýchly a efektívny vývoj multiplatformných aplikácií. Bezpečnostné riešenia boli účinné, ale pre širšie využitie by bolo potrebné ich rozšíriť. Voxerra môže slúžiť ako základ pre ďalšie projekty, ktoré by sa zamerali na škálovateľnosť a vyššiu úroveň ochrany súkromia.

# Zhrnutie

Cieľom práce bolo vytvoriť multiplatformnú chatovaciu aplikáciu Voxerra pre Android a Windows, ktorá zabezpečí bezpečnú a rýchlu komunikáciu. Na dosiahnutie tohto cieľa som použil technológie .NET MAUI pre klientskú časť, ASP.NET Core Web API pre server a MySQL pre ukladanie dát. Real-time prenos správ zabezpečila knižnica SignalR, zatiaľ čo bezpečnosť riešili JWT tokeny a HTTPS.

Výsledkom je plne funkčná aplikácia, kde správy prichádzajú s oneskorením pod 0,5 sekundy a server zvláda až 50 súčasných používateľov. Bezpečnostné mechanizmy (validácia certifikátov, rate limiting) úspešne prešli testovaním proti útokom. Dizajnové rozhodnutia – ako logo vytvorené v Adobe Photoshop a ikony z Google Fonts – prispeli k intuitívnemu rozhraniu, ktoré používatelia hodnotili pozitívne.

Napriek úspechom projekt odhalil aj určité obmedzenia. Absencia end-to-end šifrovania môže obmedzovať úroveň súkromia, čo by sa dalo vyriešiť implementáciou overených šifrovacích protokolov, ako je Signal Protocol, ktorý používajú aplikácie ako WhatsApp.

Práca ukazuje, že kombinácia moderných technológií (.NET MAUI, SignalR) umožňuje efektívny vývoj multiplatformných aplikácií aj pre začiatočníkov. Voxerra môže slúžiť ako vzdelávací projekt alebo základ pre rozšírenie o pokročilé funkcie, ako sú push notifikácie alebo end-to-end šifrovanie.

# Zoznam použitej literatúry

1. What is .NET MAUI ?. [online]. 2024, [cit. 2025-02-16]. Dostupné na internete:  
   <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-8.0>
2. C#. [online]. 2024, [cit. 2025-02-16]. Dostupné na internete:  
   <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/overview>
3. XAML overview (WPF .NET). [online]. 2024, [cit. 2025-02-16]. Dostupné na internete:  
   <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-8.0>
4. Overview of ASP.NET Core [online]. 2024, [cit. 2025-02-16]. Dostupné na internete:  
   <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-8.0>
5. Dokumentácia pre Visual Studio 2022 [online]. 2023, [cit. 2025-02-16]. Dostupné na internete: <https://learn.microsoft.com/en-us/visualstudio/windows/?view=vs-2022>
6. Význam cudzieho slova „vox“ [online]. 2025, [cit. 2025-02-16]. Dostupné na internete:  
   <https://slovnik.aktuality.sk/slovnik-cudzich-slov/?q=vox>
7. Adobe Photoshop [online]. 2025, [cit. 2025-02-16]. Dostupné na internete: <https://www.adobe.com/sk/products/photoshop.html>
8. Google Icons [online]. 2025, [cit. 2025-02-16]. Dostupné na internete: <https://fonts.google.com/icons>
9. Figma [online]. 2025, [cit. 2025-01-16]. Dostupné na internete: <https://www.figma.com/>

PrílohA A

**CD/USB príloha**

Priložené CD/USB obsahuje:

* Dokumentáciu v elektronickej podobe
* Fotografie z realizácie modelu
* Fotografie a video záznam finálneho stavu projektu
* Schému zapojenia modulov
* Zdrojový kód pre Arduino Mega 2560
* Knižnice pre Arduino Ide
* Zdrojové súbory 3D modelu (stl, gcode)

PrílohA B

**Obrázok:** Digitálna schéma databázy

