

# **STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST**

**Obor č. 18: Informatika**

## **Program pro chat**

**Patrik Dvořáček**  
**Kraj Vysočina**

**Třebíč 2018**

# STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

**Obor č. 18: Informatika**

**Program pro chat**

**Chat program**

**Autoři:** Patrik Dvořáček

**Škola:** Střední průmyslová škola Třebíč, Manželů Curieových 734,  
674 01 Třebíč

**Kraj:** Kraj Vysočina

**Konzultant:** Ing. Ladislav Havlát

Třebíč 2018

# Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod 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 Třebíči dne 21. března 2018 .....

Patrik Dvořáček

## **Poděkování**

Rád bych poděkoval panu inženýrovi Havlátovi za jeho pomoc po celou dobu práce na tomto projektu. Velké díky také patří Jaroslavu Hortovi a Zbyňkovi Juřicovi za jejich věcné připomínky a neúnavné testování nedokončených částí projektu.

## **Anotace**

Tématem práce je vytvoření programu pro konverzaci přes lokální síť. Dokáže přenášet textové zprávy a soubory. Je vytvořen v programovacím prostředí Visual Studio 2017 a programovacím jazyce C#.

## **Klíčová slova**

Chat, C#, .NET

## **Annotation**

The theme of this work is creation of a program for conversation over local network. It can send text messages and files. It has been created in Visual Studio 2017 integrated development environment and C# programming language.

## **Keywords**

Chat, C#, .NET

## OBSAH

Obsah .....	5
1 Úvod.....	6
2 Použité technologie.....	7
2.1 Visual Studio.....	7
2.2 Programovací jazyk C# .....	7
2.3 Framework .NET.....	7
3 Rozbor programu .....	8
3.1 Přehled programu .....	8
3.2 Komunikace .....	8
4 Spojení mezi serverem a klientem .....	9
4.1 Princip spojení.....	9
4.2 Hlavička zprávy .....	9
4.3 Přenos souborů .....	10
5 Struktura programu .....	11
5.1 Design .....	11
5.1.1 Material Design.....	11
5.2 Server .....	12
5.2.1 Úvod.....	12
5.2.2 Rozšířená nastavení .....	14
5.2.3 Hlavní okno.....	16
5.3 Klient.....	18
5.3.1 Úvod.....	18
5.3.2 Rozšířená nastavení .....	19
5.3.3 Hlavní okno.....	20
6 Závěr .....	22
7 Seznam obrázků a tabulek .....	23
8 Použité zdroje .....	24
Příloha A Use Case Diagram .....	25

# 1 Úvod

Téma softwaru pro chat jsem si zvolil, protože se mi komunikace po síti zdála jako obtížný a zároveň splnitelný úkol. S komunikací po síti jsem jako programátor žádné předchozí zkušenosti neměl, během práce jsem díky tomu mohl nasbírat mnoho zkušeností s jazykem C# a prostředím .NET.

Cílem této práce bylo vytvořit program, který bude schopný posílat po lokální síti více klientům najednou. Hlavní způsob komunikace bylo posílání textových zpráv, chtěl jsem ale vytvořit i možnost pro posílání dalších souborů.

Samozřejmě existuje mnoho lepších a široce používaných řešení a nepočítám s tím, že by je můj projekt mohl nahradit.

## **2 POUŽITÉ TECHNOLOGIE**

### **2.1 Visual Studio**

Visual Studio je integrované vývojové prostředí od společnosti Microsoft. Hlavním programovacím jazykem, který lze ve Visual Studiu použít, je C#. Microsoft zároveň podporuje mnoho dalších jazyků jako C, C++, Visual Basic, F# nebo Javascript. Je také možné si doinstalovat další balíčky dle potřeb.

První verze vznikla v roce 1997, kdy Microsoft spojil dohromady více vývojových nástrojů do jednoho balíčku. Původně se ve vývojovém prostředí nacházely jen jazyky Visual Basic, Visual C++ a J++ pro Javu. S dalšími verzemi rostly i schopnosti Visual Studia, které je nyní schopné podporovat projekty od webových prezentací, přes tvorbu databází až po vývoj mobilních aplikací.

### **2.2 Programovací jazyk C#**

Programovací jazyk C# je vysokoúrovňový a objektově orientovaný jazyk vyvinutý společností Microsoft v roce 2000. Má širokou škálu použití, dokáže totiž pracovat s webovými službami i jako běžný program. Syntaxe jazyku C# se schválně podobá jazykům C, C++ a Java, aby se jej i programátoři se základními znalostmi mohli rychle naučit.

### **2.3 Framework .NET**

Framework .NET je soubor technologií, který zastřešuje mnoho programovacích jazyků a platform. Bez ohledu na původní jazyk, ve kterém je program napsán, jej vždy kompilátor převede na Common Language Infrastructure (CLI), což je kód, který je schopen zpracovat jakýkoliv systém podporující .NET. Není proto potřeba programy z dalších vysokoúrovňových jazyků převádět na nejrozšířenější C# nebo naopak.



## 3 ROZBOR PROGRAMU

### 3.1 Přehled programu

Hlavní myšlenkou projektu bylo vytvořit dvě aplikace. První je aplikace serveru, která zajišťuje připojení mezi klienty, přijímání a odesílání zpráv. Jedná se o hlavní část programu, bez níž není komunikace možná. Server nastavuje administrátor a při spuštění může zvolit, na jakém portu bude server komunikovat s klienty nebo kolik klientů se může najednou připojit.

Program klienta je mnohem jednodušší a hardwarově nenáročný, protože komunikuje jenom se serverem. Klient se po zadání správné IP adresy a portu aplikace připojí na server, odkud přijímá data a soubory. Skrze stejné připojení zároveň data odesílá na server, který je zpracuje a rozešle je všem klientům.

### 3.2 Komunikace

Všechna komunikace probíhá skrze server, klienti mezi sebou přímo nekomunikují. Každý klient naváže spojení se severem a po odeslání dat je server přijme, zpracuje a následně odešle všem klientům. Komunikace je díky tomu synchronizována a všem klientům se zobrazují stejné zprávy ve stejný čas.

Spojení mezi serverem a klientem je realizováno skrze sockety, jež jsou součástí sady .NET. Projekt tím není vázán jen na programovací jazyk C# a v budoucnu je tedy možné program přenést i na mobilní platformy nebo z něj učinit webovou aplikaci.

```
if (_prichozikomunikace.Pending() && MaximalniPocet()  
{  
    var klient = _prichozikomunikace.AcceptTcpClient();  
    ++_pocetPripojzeni;  
    var byteJmeno = new byte[1024 * 1024 * 2];  
    var cteniJmena = klient.GetStream();  
    cteniJmena.Read(byteJmeno, 0, klient.ReceiveBufferSize);  
    var jmeno = Encoding.Unicode.GetString(byteJmeno)  
        .TrimEnd('\0');  
  
    if (KontrolaJmena(jmeno))  
    {  
        _seznamKlientu.Add(jmeno, klient);  
        Invoke((MethodInvoker) (() => VypisKlientu.Items.Add(jmeno)));  
        Invoke((MethodInvoker) (() => Vysilani("SERVER", jmeno + " se připojil(a)"));  
        AktualizaceSeznamu();  
  
        OdeslatHistorii(jmeno);  
  
        var vlaknoKlienta = new Thread(() => ObsluhaKlienta(jmeno, klient))  
        {  
            IsBackground = true  
        };  
  
        vlaknoKlienta.Start();  
    }  
}
```

Obrázek 3.1 Kód serveru pro příjem klientů

## 4 SPOJENÍ MEZI SERVEREM A KLIENTEM

### 4.1 Princip spojení

Klient i server využívají třídu `TcpClient`. Server po spuštění na zadaném portu spustí použije třídu `TcpListener`, která hlídá příchozí žádosti o připojení. Po přijetí požadavku a kontrole, zda není přesážen maximální počet připojení nebo klient nepoužívá duplikátní jméno, je spojení přijato.

Mezi serverem a klientem se tak vytvoří obousměrné spojení. Server i klient jsou proto schopni přijímat i odesílat data zároveň. Přes toto spojení je ale možné odesílat jen data v podobě bytů, každá zpráva i soubor proto musí být před spojením převedena na pole bytů a po přijetí znovu převedena zpět do původní podoby.

### 4.2 Hlavička zprávy

Před odesláním zprávy nebo souboru se do pole bytů přidá hlavička, aby odeslaná data mohl přijímající program správně zpracovat. V hlavičce se nachází značka v podobě číslice. Data se rozpoznávají podle této tabulky:

Tabulka 1 Hlavičky přenášených dat

Znak	Druh dat	Použití
0	Běžná textová zpráva	Pracují s ní obě strany
1	Obrázek	Pracují s ním obě strany
2	Soubor	Pracují s ním obě strany
3	Seznam klientů	Přijímá ho jen klient, odesílá pouze server
4	Žádost o odpojení	Přijímá ho jen server, odesílá pouze klient
5	Historie	Přijímá ji pouze klient, odesílá ji pouze server

Po přijetí dat program z pole bytů zkopíruje první tři byty a následně je převede na znak v kódování Unicode. Po jeho přečtení data zpracuje podle jejich druhu. Server ani klient nepočítají, že by se v komunikaci mohl objevit jiný první znak, a proto se v programu nenachází ošetření.

Znak dat a běžný text je oddělen znakem  $\phi$ . Program jej používá, protože se v běžné komunikaci nevyskytuje, neměl by tedy narušit odeslané zprávy.

### 4.3 Přenos souborů

Přenos souborů funguje stejně jako přenos textových zpráv, avšak využívá rozšířené hlavičky. Kromě znaku druhu dat se v ní nachází také velikost souboru, název souboru a jeho přípona. Program po přijetí těchto dat podle hlavičky z přijatého pole bytů vyjme data souboru a poté je uloží pod zadaným názvem a příponou.

Odesílání souborů a obrázků je bohužel omezeno na soubory s maximální velikostí 2 MB. Soubory s větší velikostí se při použití způsobu přenosu obsaženém v programu přenesou neúplně či poškozeně a uložený soubor poté nelze otevřít.

```
private void ZpracovaniSouboru(byte[] data, string nazev, string pripona, string odesilatel, string znak)
{
    var druh = 1;

    if (znak == "Soubor") druh = 2;

    UlozitSoubor(data, nazev, pripona, znak);

    var hlavicka = Encoding.Unicode.GetBytes(druh + "ϕ" + data.Length + "ϕ" + nazev + "ϕ" + pripona);
    var odesilanaData = new byte[2048 * 2048 + 128];

    Array.Copy(hlavicka, 0, odesilanaData, 0, hlavicka.Length);
    Array.Copy(data, 0, odesilanaData, 128, data.Length);

    foreach (var klient in _seznamKlientu)
    {
        var odeslani = new Thread(() => OdeslaniDat(odesilanaData, klient.Key));
        odeslani.Start();
    }

    Vysilani(odesilatel, "Poslal obrázek " + nazev + pripona);
}
```

Obrázek 4.1 Kód funkce zpracování souboru

## 5 STRUKTURA PROGRAMU

### 5.1 Design

V projektu je použit designový balíček MaterialSkin for .NET WinForms od belgického vývojáře Ignace Maes. Jeho cílem je přenést designový jazyk Material Design od Googlu do formulářových programů na platformě .NET. Tento balíček je volně stažitelný pod licencí MIT, je tedy možné jej použít i v komerčních produktech pod podmínkou, že je jeho použití uvedeno v dokumentaci.

Do projektu byl tento balíček přidán, protože přináší jednoduchý a čistý vzhled. Oproti základním komponentám formulářových aplikací obsahuje animace a nabízí i jednoduché úpravy vzhledu jako jsou změny barev napříč všemi formuláři najednou. Zároveň je možné v něm intuitivně využít výrazné barvy známé z Material Designu.

MaterialSkin for .NET WinForms má ovšem své nevýhody. Především jeho komponenty často nemají všechny funkce a vlastnosti, které jsou běžné u základních komponent. Vývoj na tomto projektu je už pozastaven, určité komponenty jsou kvůli tomu nedokončeny a nebylo možné je při tvorbě maturitní práce použít.

#### 5.1.1 Material Design

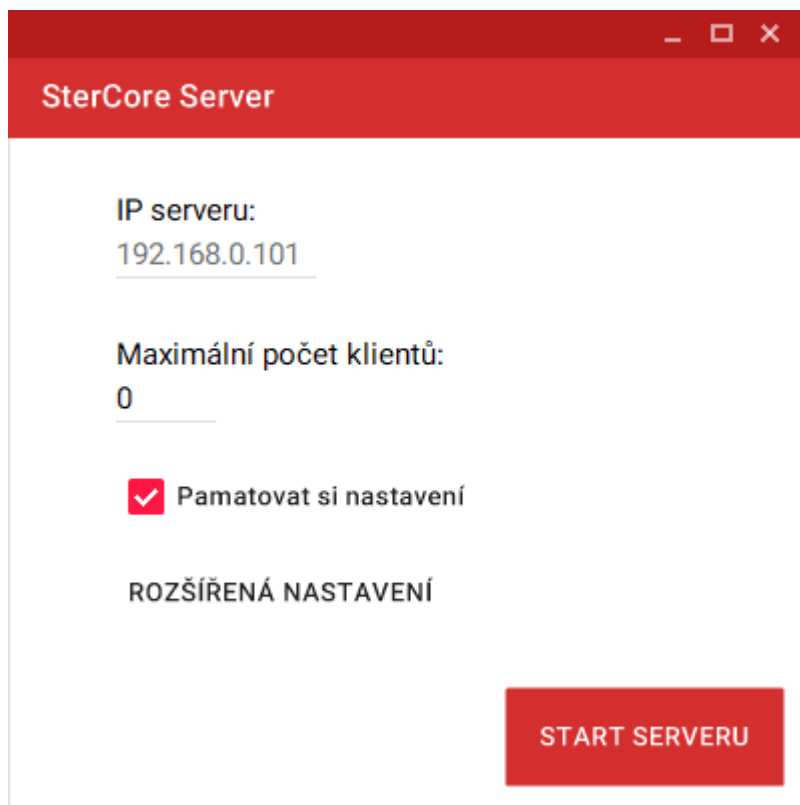
V projektu je použit designový balíček MaterialSkin for .NET WinForms od belgického vývojáře Ignace Maes. Jeho cílem je přenést designový jazyk Material Design od Googlu do formulářových programů na platformě .NET. Tento balíček je volně stažitelný pod licencí MIT, je tedy možné jej použít i v komerčních produktech pod podmínkou, že je jeho použití uvedeno v dokumentaci.

Do projektu byl tento balíček přidán, protože přináší jednoduchý a čistý vzhled. Oproti základním komponentám formulářových aplikací obsahuje animace a nabízí i jednoduché úpravy vzhledu jako jsou změny barev napříč všemi formuláři najednou. Zároveň je možné v něm intuitivně využít výrazné barvy známé z Material Designu.

MaterialSkin for .NET WinForms má ovšem své nevýhody. Především jeho komponenty často nemají všechny funkce a vlastnosti, které jsou běžné u základních komponent. Vývoj na tomto projektu je už pozastaven, určité komponenty jsou kvůli tomu nedokončeny a nebylo možné je při tvorbě maturitní práce použít.

## 5.2 Server

### 5.2.1 Úvod



Obrázek 5.1 Úvodní okno serveru

Úvodní okno serveru se zobrazí po spuštění programu. Administrátor může vidět:

- IP adresu
- Maximální počet klientů
- Políčko pro zapamatování nastavení
- Tlačítko pro otevření rozšířených nastavení
- Tlačítko pro spuštění serveru

Administrátor může server rychle spustit, aniž by měnil jakákoliv nastavení. Pokud byl program již dříve spuštěn a bylo uloženo nastavení, program automaticky nahraje předchozí nastavení a vloží je do programu.

#### IP serveru

IP serveru je hodnota, která se automaticky nahraje při spuštění programu. Administrátor ji nemůže změnit, program ji zjistí od DNS serveru, k němuž je počítač připojen. Pokud není počítač připojen k lokální síti, nastaví se hodnota 127.0.0.1.

**Maximální počet klientů**

Tato hodnota určuje maximální počet klientů, který se na server může najednou připojit. Pokud je nastaven počet na nulu, server žádný maximální počet připojení nepoužívá. V tom případě je počet připojení omezen jen propustností lokální sítě a hardwaru, na němž program běží.

**Pamatovat si nastavení**

Pokud je tlačítko zaškrtnuto při spuštění serveru, jsou aktuální nastavena do souboru Nastaveni.txt uloženém ve složce \Dokumenty\SterCore soubory. V opačném případě je uložení ignorováno a soubor s nastavením se buď nevytvoří, nebo zůstává nezměněn.

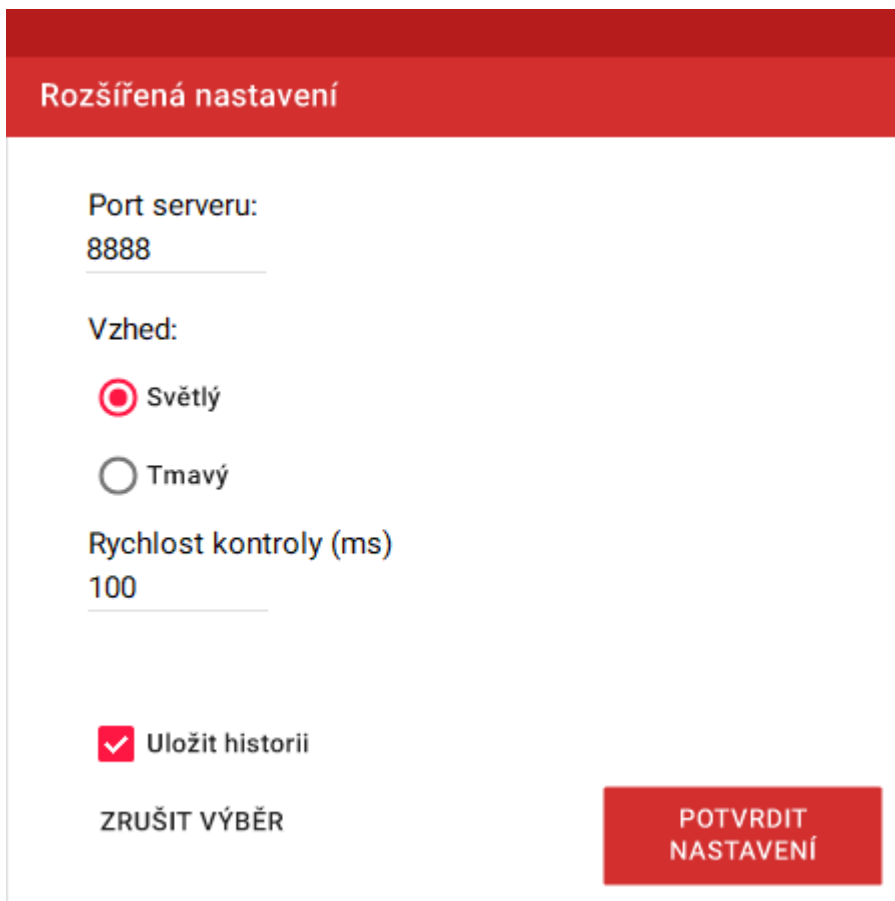
**Rozšířená nastavení**

Stisknutí tlačítka otevře okno s rozšířenými nastaveními. Otevření tohoto okna není pro chod programu povinné a administrátor jej může ignorovat.

**Start serveru**

Administrátor stisknutím tohoto tlačítka spustí chod serveru na zobrazené IP adrese a portu nastaveném v rozšířených nastaveních.

### 5.2.2 Rozšířená nastavení



Rozšířená nastavení

Port serveru:  
8888

Vzhed:

☒ Světlý

☐ Tmavý

Rychlost kontroly (ms)  
100

☒ Uložit historii

ZRUŠIT VÝBĚR

POTVRDIT NASTAVENÍ

Obrázek 5.2 Rozšířená nastavení serveru

Okno pro rozšířená nastavení umožňuje hlubší změny v programu, jejichž úprava není pro chod serveru důležitá. Zároveň je zde možné změnit vzhled programu. Administrátor v okně vidí:

- Port serveru
- Změnu vzhledu
- Rychlost kontroly v milisekundách
- Políčko pro uložení historie
- Tlačítko pro zrušení výběru
- Tlačítko pro potvrzení nastavení

### **Port serveru**

Port serveru určuje síťový port, na kterém bude server s klienty komunikovat. Síťový port je hlavním identifikátorem při komunikaci programů za pomoci protokolů TCP a UDP. Základní hodnota je nastavena na 8888, administrátor má přesto možnost tuto hodnotu změnit.

Hodnota portu serveru musí být stejná u serveru i všech klientů. Pokud by klient při pokusu o připojení zadal špatný port, bylo by jeho připojení zamítnuto. Server totiž poslouchá jen na jediném síťovém portu.

### **Změna vzhledu**

Změna vzhledu dovoluje administrátorovi přepínat mezi tmavým a světlým vzhledem. Díky balíčku MaterialSkin for .NET WinForms je možné bez obtíží změnit světlé pozadí a tmavé písmo na tmavé pozadí a světlé písmo a naopak.

### **Rychlost kontroly**

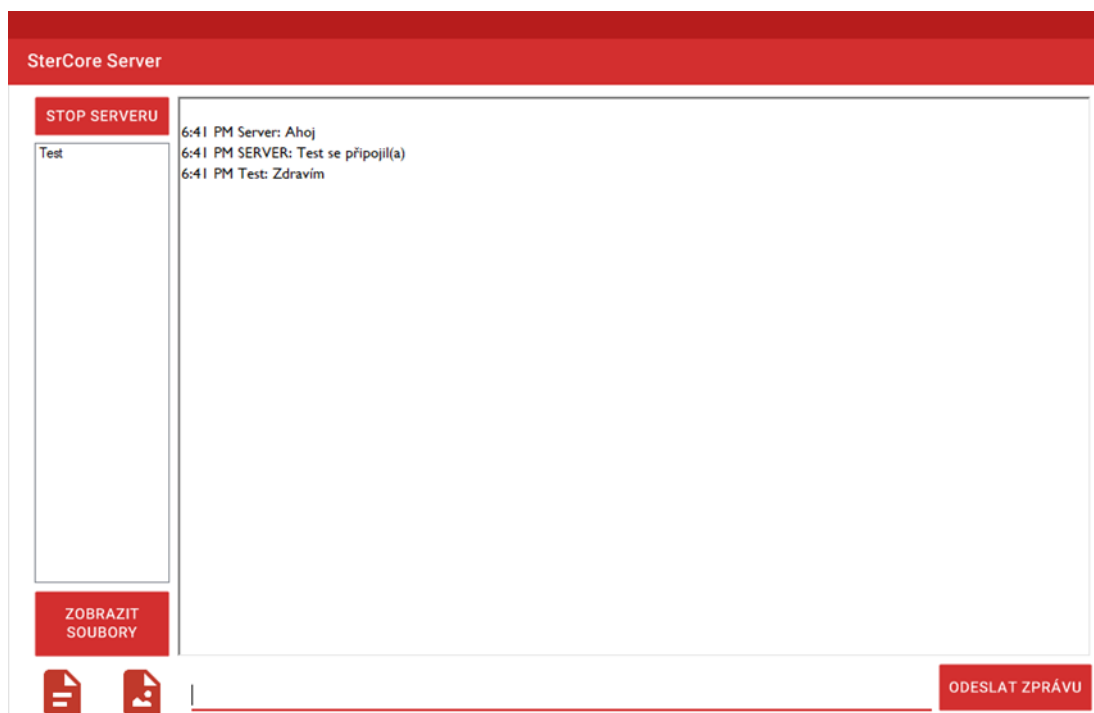
Rychlost kontroly je hodnota udávaná v milisekundách, jenž udává dobu mezi odesláním zpráv ze serveru. Základní hodnotou je 100 ms, program tedy počká 100 ms, po odeslání dat, než pošle další. Pokud je připojení v lokální síti rychlé, je možné tuto hodnotu snížit i na jednotky milisekund.

### **Políčko uložení historie**

Pokud administrátor ponechá toto políčko zaškrtnuté, historie konverzace se po vypnutí programu uloží do souboru Historie.txt. Po opětovném spuštění serveru se tento soubor načte do okna chatu.



### 5.2.3 Hlavní okno



Obrázek 5.3 Hlavní okno serveru

V hlavním okně probíhá veškerá komunikace mezi serverem a klienty. Administrátor zde může prohlížet konverzaci a zároveň se jí může zúčastnit. Administrátor má také možnost odesílat soubory a obrázky, má možnost sledovat počet a jména připojených klientů.

Okno je rozděleno do pěti částí:

- Oblast s chatem
- Odesílání souborů a obrázků
- Odesílání zpráv
- Seznam připojených
- Tlačítko zastavení serveru

#### Oblast s chatem

V této komponentě se zobrazuje veškerá komunikace, jenž probíhá mezi klienty a serverem. Zároveň se zde objeví oznámení, pokud někdo odeslal soubor nebo obrázek. Server zde také vypisuje chyby, avšak klientům se nezobrazí, pokud nejsou nahrány v rámci historie.

Zprávy mají v chatu jednoduché formátování. Jako první se zobrazí čas, kdy byla zpráva přijata. Následně server vypíše jméno odesílatele dané zprávy. Poté již následuje samotný obsah.

Text je ukládán v kódování Unicode. Komponenta s chatem dovoluje ukládání celého obsahu včetně formátování, aplikace serveru ale pro jednoduchost ukládá text bez formátování běžného textového souboru.

### **Odesílání souborů a obrázků**

V této oblasti se nachází tlačítka pro odeslání souborů. Jejich maximální velikost je omezena na 2 MB. Po stisknutí příslušného tlačítka se zobrazí okno pro výběr souboru nebo obrázku. Při úspěšném zadání je soubor odeslán všem klientům a jeho přenos následně oznámen v okně chatu.

Soubory i obrázky se ukládají do složky \Dokumenty\SterCore Soubory\. Při odesílání se soubor rovnou do této složky uloží a přenáší se jenom klientům. Administrátor má k této složce běžný přístup, může ale využít tlačítka Zobrazit soubory, které složku otevře v jeho prohlížeči.

### **Odesílání zpráv**

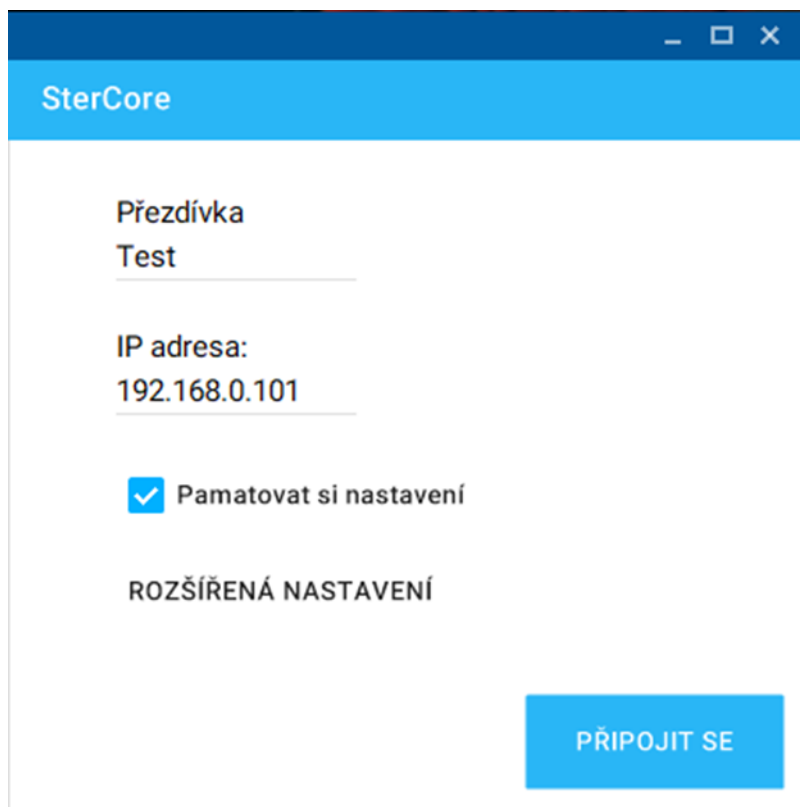
V této oblasti se nachází komponenta pro zadávání textu v kódovaném ve formátu Unicode. Vedle ní se nachází tlačítko pro odeslání zprávy, není ale potřeba jej využít. Při dopsání zprávy stačí jen zmáčknout klávesu Enter a zpráva se odešle.

### **Seznam připojených**

Seznam připojených slouží k informování, kdo je aktuálně na server připojen. V okně se zobrazují přezdívky klientů a okno se s každým připojením nebo odpojením aktualizuje. Do výpisu není zahrnut server.

## 5.3 Klient

### 5.3.1 Úvod



Obrázek 5.4 Úvodní okno serveru

Úvodní okno klienta se zobrazí okamžitě po spuštění programu. Jeho design a funkce je mnohem jednodušší než serveru, uživatel proto nemusí mít technické znalosti jako u serveru.

V okně se nachází:

- Pole s přezdívkou
- Pole s IP adresou
- Políčko pro zapamatování nastavení
- Tlačítko pro rozšířená nastavení
- Tlačítko pro připojení

#### Přezdívka

Do pole přezdívka se zadává jméno, s nímž chce uživatel v chatu vystupovat. Zadaná hodnota může obsahovat maximálně 20 znaků a nemůže být prázdná. Také nelze použít jméno vytvořené jen z mezer nebo prázdných znaků. Přezdívka po prvním spuštění prázdná, v případě, že klient uložil nastavení, se načte ze souboru s nastavením.

### IP adresa

Do tohoto pole zadává uživatel IP adresu serveru. Adresu musí dopředu znát, jinak není připojení na server možné. I tato hodnota je po prvním spuštění prázdná a také se načítá ze souboru s nastavením.

### Pamatovat si nastavení

Pokud je políčko zaškrtnuto při připojení, uloží se nastavení do složky \Dokumenty\SterCore soubory. V opačném případě se s nastavením nic nestane a při dalším spuštění se buď načtou základní hodnoty, nebo předchozí soubor s nastavením.

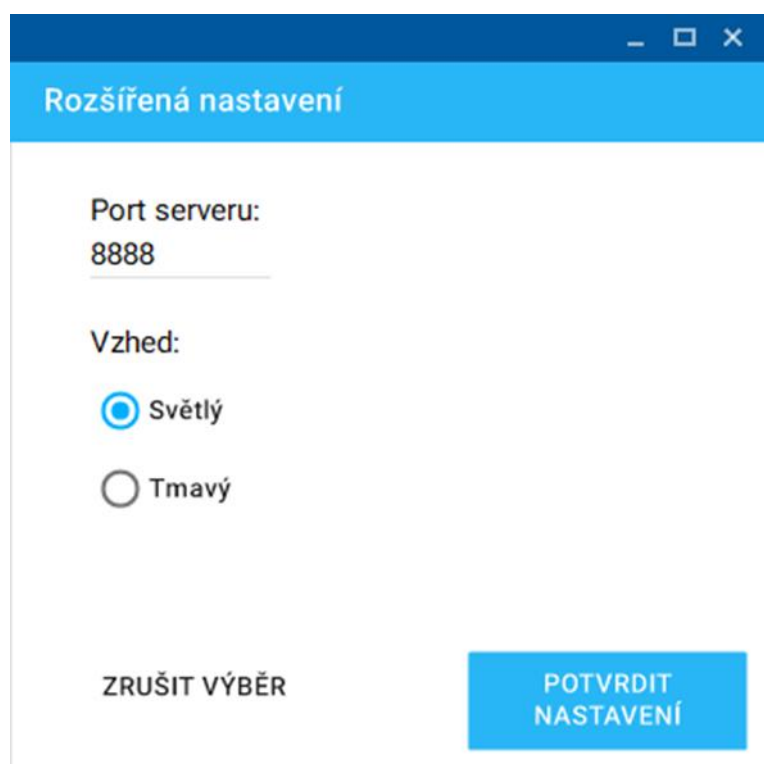
### Rozšířená nastavení

Otevře okno s volitelnými nastaveními. Použití tohoto okna není pro chod programu povinné.

### Připojit se

Po stisknutí program zkontroluje zadaná údaje a pokusí se o připojení na server. Při úspěšném připojení se otevře hlavní okno a uživatel může komunikovat. V opačném případě se zobrazí chybová hláška.

## 5.3.2 Rozšířená nastavení



Obrázek 5.5 Rozšířená nastavení klienta

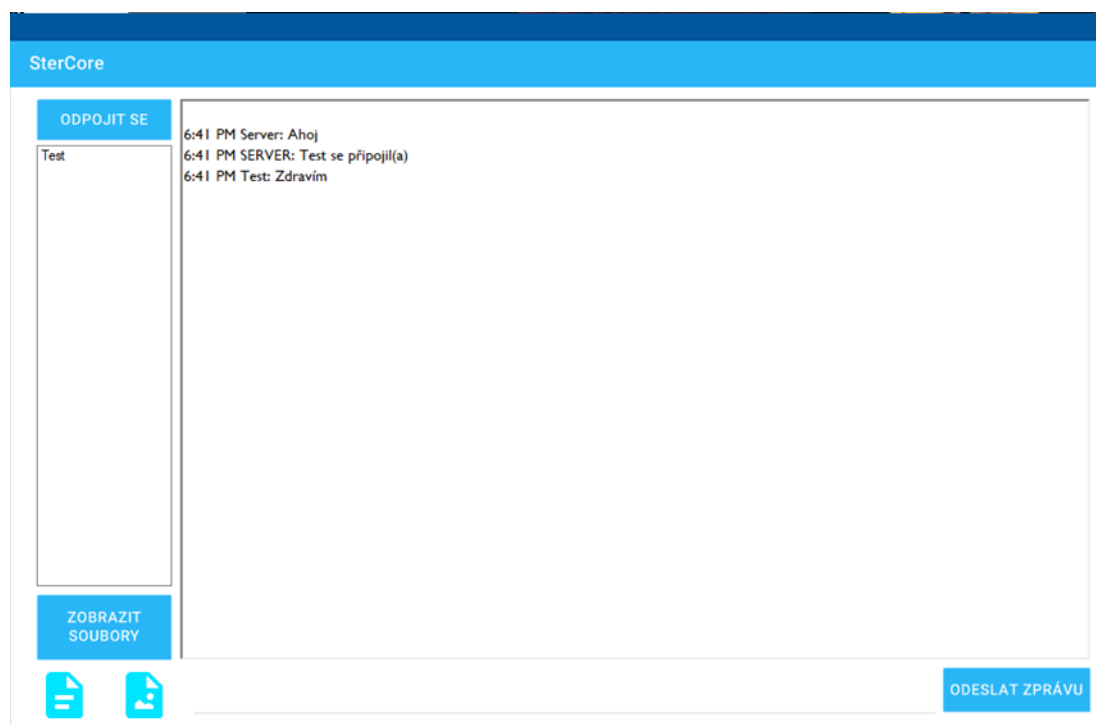
### Port serveru

Port serveru určuje síťový port, skrze který bude klient se serverem komunikovat. Zadanou hodnotu není potřeba měnit, pokud není změněna na straně serveru.

## Vzhled

Pomocí balíčku MaterialSkin for .NET WinForms změní vzhled oken a komponent, aby odpovídal volbě uživatele.

### 5.3.3 Hlavní okno



Obrázek 5.6 Hlavní okno klienta

Hlavní okno klienta je rozvrženo naprosto stejně jako hlavní okno serveru. Opět platí rozdělení na pět různých oblastí:

- Oblast s chatem
- Odesílání souborů a obrázků
- Odesílání zpráv
- Seznam připojených
- Tlačítko odpojení

Jediným rozdílem je nahrazení tlačítka pro vypnutí serveru tlačítkem pro odpojení. Způsob připojení bohužel nedovoluje kontrolu stavu připojení bez opakovaných požadavků, které by zbytečně zahlcovali server a lokální síť. Avšak pokud je klient odpojen a pokusí se odeslat zprávu, odpojení je oznámeno, hlavní okno se uzavře a uživatel se vrátí na úvodní okno.

**Odpojit se**

Tlačítko odpojit se odešle serveru požadavek na ukončení spojení. Po zpracování zprávy server ukončí spojení a hlavní okno klienta se uzavře. Ostatním připojeným se zobrazí zpráva o odpojení, aktualizuje se seznam připojených a klient se vrátí na úvodní okno programu.

## 6 ZÁVĚR

Samotná tvorba aplikace byla mnohem komplikovanější, než se na začátku projektu předpokládalo. Samotný základ aplikace, tedy spojení mezi serverem a klientem a přenos dat, musel být čtyřikrát přepsán od začátku. Hlavním problémem bylo samotné síťové spojení a druhou překážkou bylo použití threadingu. I přes různé potíže a neočekávané chyby se povedlo projekt úspěšně dokončit.

Při vývoji se čerpalo převážně z dokumentace jazyku C# a frameworku .NET přímo na webových stránkách Microsoftu. Skvělou inspirací byly také otázky na fóru StackOverflow, i když se většinou netýkali tvorby formulářových aplikací.

Od původní vize programu se bohužel liší dvě funkce programu. Zprvé se nepodařilo implementovat zobrazování obrázků přímo do okna chatu. Druhým nedostatkem je přenášení souborů s omezenou velikostí, přestože by při použití třídy TcpClient mělo být možné posílat soubory s neomezenou velikostí.

## 7 SEZNAM OBRÁZKŮ A TABULEK

Obrázek 3.1 Kód serveru pro příjem klientů .....	8
Obrázek 4.1 Kód funkce zpracování souboru .....	10
Obrázek 5.1 Úvodní okno serveru .....	12
Obrázek 5.2 Rozšířená nastavení serveru .....	14
Obrázek 5.3 Hlavní okno serveru .....	16
Obrázek 5.4 Úvodní okno serveru .....	18
Obrázek 5.5 Rozšířená nastavení klienta .....	19
Obrázek 5.6 Hlavní okno klienta .....	20
Tabulka 1 Hlavičky přenášených dat .....	9



## 8 POUŽITÉ ZDROJE

- [1] LEVY, Ken. History of the VS IDE. Code Magazine [online]. 2008, (5) [cit. 2018-03-18]. Dostupné z: <http://www.codemag.com/Article/0710022/History-of-the-VS-IDE>
- [2] Visual Studio IDE overview. Microsoft Docs [online]. 5.2. 2018 [cit. 2018-03-18]. Dostupné z: <https://docs.microsoft.com/en-gb/visualstudio/ide/visual-studio-ide>
- [3] Introduction to the C# Language and the .NET Framework. Microsoft Docs [online]. 20.7. 2015 [cit. 2018-03-18]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>
- [4] TcpListener Class. Microsoft Developer Network [online]. [cit. 2018-03-18]. Dostupné z: [https://msdn.microsoft.com/cs-cz/library/system.net.sockets.tcpllistener\(v=vs.110\).aspx](https://msdn.microsoft.com/cs-cz/library/system.net.sockets.tcpllistener(v=vs.110).aspx)
- [5] MAES, Ignace. MaterialSkin for .NET WinForms. Github [online]. 2015 [cit. 2018-03-19]. Dostupné z: <https://github.com/IgnaceMaes/MaterialSkin>
- [6] Material Design. Material Design [online]. [cit. 2018-03-19]. Dostupné z: <https://material.io/>
- [7] Co je to port. Správa sítě – slovník pojmů [online]. 2016 [cit. 2018-03-19]. Dostupné z: <https://www.sprava-site.eu/port/>

## PŘÍLOHA A USE CASE DIAGRAM

