

# **Alternatywny system synchronizacji.8**

Autorzy:

Jakub Ciałkowski  
Łukasz Pijarczyk  
Przemysław Sendek  
Tomasz Wójcik  
Paweł Zembrzusi

Warszawa, dn. 08.05.09

## **Spis treści:**

### **1. Informacje o systemie.**

#### **1.1. Specyfikacja systemu.**

#### **1.2. Przegląd zastosowanych narzędzi.**

##### **1.2.1. Język programowania C++.**

##### **1.2.2. Język programowania C#.**

##### **1.2.3. Język programowania MySQL.**

##### **1.2.4. Język programowania Django.**

### **2. Struktura systemu.**

#### **2.1. Opis głównych struktur systemu.**

##### **2.1.1. Klient.**

##### **2.1.2. Serwer.**

##### **2.1.3. Portal WWW.**

##### **2.1.4. Baza danych.**

#### **2.2. Kody źródłowe.**

### **3. Literatura i źródła.**

### **4. Załączniki**

#### **4.1. Przebieg testów programu**

#### **4.2 Changelog Serwera.**

#### **4.3. Terminarz**

## 1. Informacje o systemie.

Program ASS.8 powstał jako projekt na zajęcia Pracownia Projektowania Zespołowego w Wyższej Szkole Informatyki Stosowanej i Zarządzania w Warszawie. Głównym zadaniem programu jest umożliwienie współdzielenia plików między użytkownikami oraz synchronizacja danych za pośrednictwem serwera. Użytkownik ma możliwość współdzielenia danych dzięki klientowi usługi natomiast portal WWW daje mu możliwość logowania się i aktywnego uczestnictwa w społeczności tworzonej przez użytkowników.

### 1.1. Specyfikacja systemu.

System składa się z czterech części:

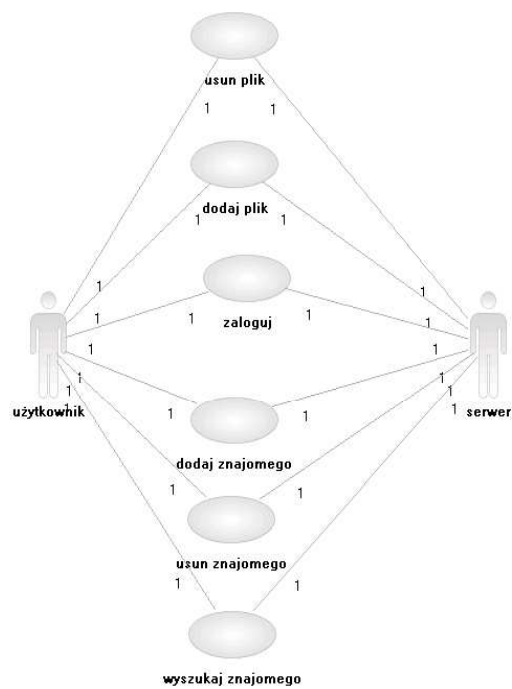
- klienta – programu instalowanego na komputerze użytkownika. Program ten ma za zadanie ułatwiać organizację plików, określanie ich praw dostępu dla innych użytkowników systemu oraz wyszukiwanie innych użytkowników systemu.
- serwera – który łączy się z bazą danych oraz jest pośrednikiem między klientem a bazą danych.
- bazy danych – gdzie przechowywane informacje o użytkownikach i ich plikach.
- serwisu WWW – strony na której dany użytkownik systemu może się zalogować, dzięki czemu uzyskuje dostęp do swojego profilu widocznego dla innych użytkowników. Osoba niezalogowana może przeglądać profile zarejestrowanych użytkowników i ma dostęp do plików oznaczonych jako publiczne.

Pliki mają trzy poziomy dostępności dla użytkowników:

- publiczny – każdy użytkownik systemu oraz osoba niezarejestrowana ma dostęp do pliku
- prywatny – dostęp do pliku mają tylko przyjaciele danego użytkownika
- osobisty – dostęp do pliku ma tylko jego właściciel

Zgodnie z przedstawionym poniżej diagramem system daje użytkownikowi następujące możliwości:

1. Logowanie się do systemu.
2. Wyszukiwanie znajomych.
3. Dodawanie znajomych.
4. Usuwanie znajomych.
5. Dodawanie plików.
6. Usuwanie plików.



## 1.2. Przegląd zastosowanych narzędzi.

W celu osiągnięcia jak najlepszych efektów podczas tworzenia systemu ASS.8, do jego wykonania zostały wykorzystane następujące narzędzia:

- Języki programowania C++ i C# przy tworzeniu serwera oraz klienta usługi.
- PostgreSQL przy tworzeniu bazy danych.
- Framework Django przy tworzeniu serwisu WWW.

### 1.2.1. Język programowania C++.

C++ to język programowania ogólnego przeznaczenia.

Umożliwia abstrakcję danych oraz stosowanie kilku paradygmatów programowania: proceduralnego, obiektowego i generycznego. Charakteryzuje się wysoką wydajnością kodu wynikowego, bezpośrednim dostępem do zasobów sprzętowych i funkcji systemowych, łatwością tworzenia i korzystania z bibliotek (napisanych w C++, C lub innych językach), niezależnością od konkretnej platformy sprzętowej lub systemowej (co gwarantuje wysoką przenośność kodów źródłowych) oraz niewielkim środowiskiem uruchomieniowym. Podstawowym obszarem jego zastosowań są aplikacje i systemy operacyjne.

C++ został zaprojektowany przez Bjarne'a Stroustrupa jako rozszerzenie języka C o obiektywne mechanizmy abstrakcji danych i silną statyczną kontrolę typów. Zachowanie zgodności z językiem C na poziomie kodu źródłowego pozostaje jednym z podstawowych celów projektowych kolejnych standardów języka.

Źródło: <http://pl.wikipedia.org/wiki/C%2B%2B>

### 1.2.2. Język programowania C#.

C# - obiektowy język programowania zaprojektowany przez zespół pod kierunkiem Andersa Hejlsberga dla firmy Microsoft.

Program napisany w tym języku kompilowany jest do języka Common Intermediate Language (CIL), specjalnego kodu pośredniego wykonywanego w środowisku uruchomieniowym takim jak .NET Framework, Mono lub DotGNU. Wykonanie skompilowanego programu przez system operacyjny bez takiego środowiska nie jest możliwe.

Cechy języka:

Język C# ma wiele cech wspólnych z językami programowania Object Pascal, Delphi, C++ i Java.

- osadzane języki proceduralne wykonywane przez bazę danych (plperl, pl/perl, plpgsql, plpython, pltcl, pl/tcl)
- obiektowość z hierarchią o jednym elemencie nadrzędnym: podobnie jak w Javie, kod programu jest zbiorem klas. W C# podobnie tak jak w Javie/Object Pascalu hierarchia dziedziczenia opiera się na istnieniu jednej klasy object (System.Object), która stanowi element nadrzędny tej hierarchii. W szczególności oznacza to, że również typy proste (int, double, itd.) są obiektami z właściwymi sobie metodami, np. `int i=1; string s = i.ToString();`
- odśmiecanie pamięci: zarządzaniem pamięcią zajmuje się środowisko uruchomieniowe. Oznacza to, że nie ma konieczności ani potrzeby samodzielnego zajmowania się zwalnianiem pamięci po obiektach, które przestają być używane
- refleksje i atrybuty klas: w czasie pracy programu istnieje możliwość analizy struktury kodu z poziomu tego kodu. Umożliwia to tworzenie wysoce uniwersalnych mechanizmów operujących na strukturze kodu nie znanej w czasie kompilacji. Mechanizm ten wykorzystywany jest m.in. w bibliotekach ORM, narzędziach do analizy i weryfikacji kodu czy rozszerzeniach AOP. Mechanizm atrybutów został z C# zapożyczony do języka Java w wersji 1.5 (adnotacje), jakkolwiek samo Reflection API istniejące od pierwszego wydania języka stanowiło inspirację dla twórców C#
- typy ogólne (generics - dostępne od wersji .NET 2.0): mechanizm zbliżony swoją ogólnością do szablonów w C++, jednak tu typ ogólny jest przenoszony do modułu binarnego i możliwy jest do wykorzystania bez konieczności posiadania kodu źródłowego (szablony w C++ to w uproszczeniu rozbudowane makrodefinicje)
- dynamiczne tworzenie kodu: biblioteki .NET umożliwiają dynamiczne tworzenie kodu w czasie działania programu i włączanie go do kodu aktualnie wykonywanego. Możliwe jest zarówno dynamiczne tworzenie kodu wykonywalnego ze źródeł C# jak i tworzenie dynamicznych modułów w języku pośrednim (MSIL).
- bogata biblioteka klas BCL, umożliwiająca rozwijanie aplikacji konsolowych, okienkowych (System.Windows.Forms), bazodanowych (ADO.NET), sieciowych (System.Net), w architekturze rozproszonej (WebServices) czy dynamicznych aplikacji internetowych (ASP.NET)

Źródło: [http://pl.wikipedia.org/wiki/C\\_Sharp](http://pl.wikipedia.org/wiki/C_Sharp)

### 1.2.3. Język programowania MySQL.

MySQL był pisany raczej z myślą o szybkości niż kompatybilności ze standardem SQL – przez dłuższy czas MySQL nie obsługiwał nawet transakcji, co było zresztą głównym argumentem przeciwników tego silnika bazodanowego. MySQL obsługuje większą część obecnego standardu ANSI/ISO SQL (tj. SQL:2003). Wprowadza również swoje rozszerzenia i nowe elementy języka

Serwer MySQL dostępny jest dla wszystkich popularnych platform systemowych i różnorodnych architektur procesorów. Jest dostępny także w wersji źródłowej, co umożliwia skompilowanie go dla dowolnej innej platformy.

Źródło: <http://pl.wikipedia.org/wiki/MySQL>

#### **1.2.4. Język programowania Django.**

Django - wysokopoziomowy, opensource'owy framework służący do tworzenia aplikacji internetowych, napisany w Pythonie. Powstał w środowisku dziennikarskim więc maksymalnie skupia się na szybkości tworzenia takich aplikacji oraz automatyzacji powtarzających się czynności tego procesu. Jego nazwa pochodzi od imienia gitarzysty Django Reinhardta.

- automatycznie generowany i kompletny panel administracyjny, z możliwością dalszego dostosowywania.
- przyjazne adresy dokumentów z możliwością dowolnego ich kształtowania.
- prosty lecz funkcjonalny system szablonów czytelny zarówno dla grafików jak i dla programistów.
- oddzielenie logiki aplikacji (kontroler) logiki biznesowej (model) wyglądu (szablony) oraz baz danych.
- wsparcie dla wielojęzycznych aplikacji.
- bardzo duża skalowalność i wydajność pod obciążeniem.
- wydajne systemy keshowania, obsługa memcache.
- własny, prosty serwer do testowania aplikacji.
- współpracuje z Apache poprzez mod\_python oraz z innymi serwerami poprzez protokoły FastCGI i SCGI.
- DRY czyli zasada „nie powtarzaj się” w odniesieniu do tworzenia aplikacji, (np. strukturę baz myśla django generuje ze zwykłych klas pythona).
- posiada ORM wysokiego poziomu pozwalający na łatwe i bezpieczne operowania na bazach danych bez użycia SQL.
- obsługuje PostgreSQL, MySQL, SQLite oraz Oracle.
- rozpowszechniany jest na liberalnej licencji BSD.

Źródło: <http://pl.wikipedia.org/wiki/Django>

## **2. Struktura systemu.**

Poniższy diagram klas przedstawia uproszczoną strukturę systemu ASS.8. Klasy typu baza, baza kont i baza plików nie są klasami w systemie w systemie lecz stanowią one fizyczną bazę danych.



## 2.1. Opis głównych struktur systemu.

Głównymi strukturami systemu ASS.8 są :



- klient – program instalowany na komputerze użytkownika. Program ten ma za zadanie ułatwiać organizację plików, określanie ich praw dostępu dla innych użytkowników systemu oraz wyszukiwanie innych użytkowników systemu.
- serwer – który łączy się z bazą danych oraz jest pośrednikiem między klientem a bazą danych.
- baza danych – gdzie przechowywane są pliki i informacje.
- serwis WWW – strony na której dany użytkownik systemu może się zalogować, dzięki czemu uzyskuje dostęp do swojego profilu widocznego dla innych użytkowników. Osoba niezalogowana może przeglądać profile zarejestrowanych użytkowników i ma dostęp do plików oznaczonych jako publiczne.

### **2.1.1. Klient**

Klient po wpisaniu loginu i hasła wysyła je do serwera czekając na odpowiedź czy jest on poprawny czy nie, jeżeli jest poprawny zapisuje otrzymany od serwera id sesji, który jest wysyłany przy każdej operacji. Po zalogowaniu okno programu minimalizuje się. Po zalogowaniu uruchamiany jest watek, który co ustalony czas sprawdza pliki na serwerze i na dysku. Najpierw pobiera listę plików z serwera, potem porównuje to z listą plików zapisanych w lokalnym pliku xml i z plikami rzeczywiście na dysku, w zależności od różnic wykonuje wysyłanie plików na serwer, ściąganie i usuwanie plików z serwera i z katalogu lokalnego.

### **2.1.2. Serwer**

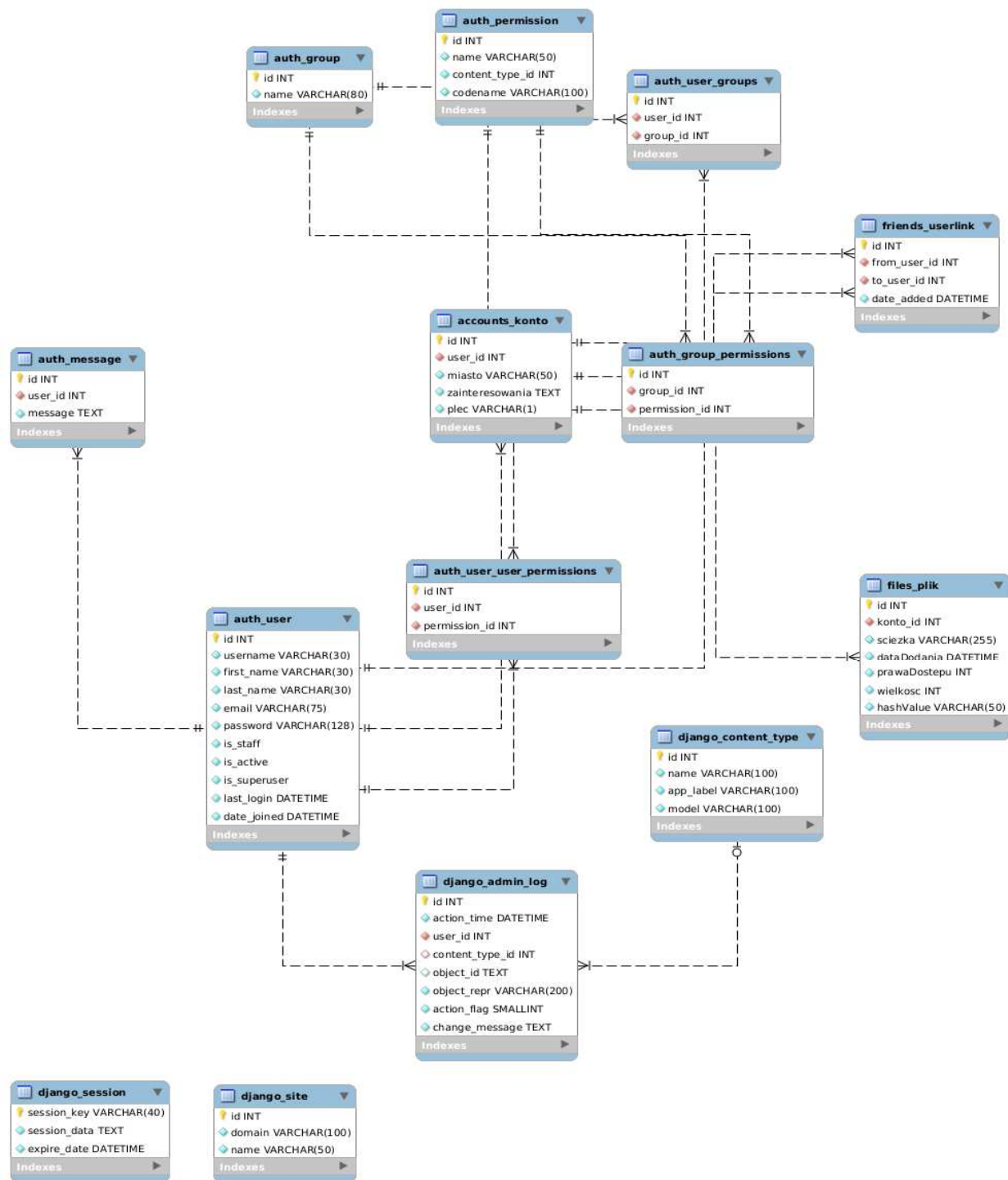
Serwer napisany w C++ jest obsługiwany przez system Linux. Ma strukturę monolityczną i jest synchroniczny. Potrafi obsłużyć wielu klientów jednocześnie i działa na wątkach. W protokole przesyłania danych wykorzystywany jest XML.

### **2.1.3. Baza danych.**

Poniższy diagram przedstawia strukturę bazy danych. Część z tabel bazy została wygenerowana automatycznie dzięki frameworkowi django. Tabele te mają za zadanie usprawnić pracę z frameworkiem oraz przechowują informacje takie jak lista stron i szablonów. Ich dokładną dokumentację można znaleźć na: <http://docs.djangoproject.com/en/dev/>.

Tabela `accounts_konto` zawiera wszystkie informacje o użytkowniku oraz klucz obcy do tabeli `auth_users`, gdzie przechowywane są szczegółowe informacje dotyczące konta użytkownika.

Tabela `files_pliki`, przechowuje informacje o plikach, zaś tabela `friends_userlink` informacje dotyczące znajomych użytkownika oraz o dane dotyczące udostępnianych plików.



## 2.1.4. Portal WWW.

Portal WWW został wygenerowany za pomocą frameworka django. Składa się on z 3 aplikacji.

- Aplikacji accounts odpowiedzialnej za zarządzanie kontami użytkowników.
- Aplikacji friends odpowiedzialnej za zarządzanie znajomymi.
- Aplikacją files odpowiedzialną za zarządzanie plikami.

Każda z aplikacji związana jest z osobną tablicą w bazie danych i korzysta ze wspólnej bazy jednak każda działa samodzielnie i korzysta z własnej logiki.

Przewidywana funkcjonalność aplikacji:

Dzięki aplikacji accounts można dodawać i usuwać konta użytkowników, logowanie i wylosowywanie, autoryzację użytkownika oraz wyszukiwanie kont innych użytkowników a również edycje swoich danych i oglądanie profilu.

Aplikacja friends pozwala nam na dodawanie i usuwanie użytkowników do listy znajomych oraz wyświetlanie listy znajomych.

Aplikacja files umożliwia usuwanie plików, wyświetlanie listy plików oraz zmianę praw dostępu do plików.

## 2.2. Kody źródłowe.

### Kod źródłowy klienta Windows:

ASS8 – Konfiguracja

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Net;
using System.Threading;

namespace ASS8.Klient
{
    public partial class ASS8___Konfiguracja : Form
    {
        komunikacja k;
        private bool close = false;
        Form loginForm;
        zarzadca z;
        private void automatyczneSpr()
        {
            while (true)
```

```

        {
            z.szukajZmian();
            Thread.Sleep(60000);
        }
    }
    public ASS8___Konfiguracja(Form log, komunikacja kom)
    {
        InitializeComponent();
        loginForm = log;
        KomClass = kom;
        z = new zarzadca();
        z.kom = k;
        Thread th = new Thread(automatyczneSpr);
        th.Start();
    }
    public komunikacja KomClass
    {
        set
        {
            k = value;
        }
    }
    private void btnSciezka_Click(object sender, EventArgs
e)
    {
        FolderBrowserDialog fbd = new
FolderBrowserDialog();
        if (fbd.ShowDialog() == DialogResult.OK)
        {
            txtSciezka.Text = fbd.SelectedPath;
        }
    }

    private void ASS8___Konfiguracja_FormClosing(object
sender, FormClosingEventArgs e)
    {
        if (!close)
        {
            this.Visible = false;
            e.Cancel = true;
        }
    }

    private void tcmsKonfiguracja_Click(object sender,
EventArgs e)
    {
        this.Visible = true;
    }

    private void tcmsZnajomi_Click(object sender,
EventArgs e)

```

```

        {
            MessageBox.Show(":)");
        }

e)    private void cmsWyloguj_Click(object sender, EventArgs
        {
            close = true;
            this.Close();
            loginForm.Show();
        }

        private void tcmsZakoncz_Click(object sender,
EventArgs e)
        {
            Application.Exit();
        }

e)    private void tcmsStop_Click(object sender, EventArgs
        {
            tcmsStart.Enabled = true;
            tcmsStop.Enabled = false;
        }

        private void tcmsStart_Click(object sender, EventArgs
e)
        {
            tcmsStart.Enabled = false;
            tcmsStop.Enabled = true;
        }
    }
}

```

ASS8 - Logowanie

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Security.Cryptography;
using System.Threading;

namespace ASS8.Klient
{
    public partial class ASS8___Logowanie : Form
    {
        private komunikacja k;
    }
}

```

```

        public static string wersja = "0.3.9";
        public ASS8___Logowanie()
        {
            InitializeComponent();
            k = new komunikacja();
        }

        private void btnLoguj_Click(object sender, EventArgs
e)
        {
            k.Haslo = haslo;
            k.Login = login;
            k.login();
            ASS8___Konfiguracja konfiguracja = new
ASS8___Konfiguracja(this, KomClass);
            this.Hide();
            //k.downloadFiles(new string[] { "plik1", "plik2"
}, "ja");
            //MessageBox.Show("Aa");
        }
        public komunikacja KomClass
        {
            get
            {
                return k;
            }
        }
        public string login
        {
            get
            {
                if (txtLogin.Text.Length == 0) return "";
                return txtLogin.Text;
            }
        }
        public string haslo
        {
            get
            {
                if (txtHaslo.Text.Length == 0) return "";
                return "-1";
                /*MD5CryptoServiceProvider md5 = new
MD5CryptoServiceProvider();
                Byte[] haslo;
                Byte[] hash;
                haslo =
ASCIIEncoding.Default.GetBytes(txtHaslo.Text);
                hash = md5.ComputeHash(haslo);
                return
ASCIIEncoding.Default.GetString(hash);*/
            }
        }
    }
}

```

```

    }
}
}

```

## Komunikacja

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using System.IO;
using System.Net;
using System.Net.Sockets;
using System.Xml.Serialization;
using System.Windows.Forms;

namespace ASS8.Klient
{
    public class komunikacja
    {
        private enum operacje
        {
            lista = 100, pobieranie, wysylanie, usuwanie
        }
        private enum odpowiedzi
        {
            bledne_zapytanie = 400, bledny_numer_sesji,
            plik_istnieje, blad_serwera, plik_nie_istnieje,
            blad_odbierania_plikow, wszystko_ok
        }
        string folder = "pliki";
        XmlSerializerNamespaces names;
        private int sessionID;
        private String serverIP = "127.0.0.1";
        private int serverPort = 2000;
        private TcpClient serwer;
        private string log;
        private string haslo;
        public komunikacja()
        {
            names = new XmlSerializerNamespaces();
            names.Add("", "");
            log = "";
            haslo = "";
        }
        public int login()
        {
            if (log == "" || haslo == "") return -2;
            serwer = new TcpClient(serverIP, serverPort);

```

```

        if (!serwer.Connected) return -1;
        NetworkStream networkStream = serwer.GetStream();
        try
        {
            StringWriter stringWriter = new
StringWriter();
            klientLogowanie logowanie = new
klientLogowanie(log, haslo, ASS8___Logowanie.wersja);
            XmlSerializer xml = new
XmlSerializer(typeof(klientLogowanie));
            xml.Serialize(stringWriter, logowanie, names);
            string strToWrite = stringWriter.ToString() +
"\r\n";

networkStream.Write(Encoding.ASCII.GetBytes(strToWrite.ToCharA
rray()), 0, strToWrite.Length);
        }
        catch (Exception)
        {
            System.Windows.Forms.MessageBox.Show("Nie
udalo sie zalogowac");
            return -1;
        }
        Byte[] buffor = new
Byte[serwer.Client.ReceiveBufferSize];
        int dlugoscOdp = 0;
        StringBuilder strRead = new StringBuilder("");
        do
        {
            int i = networkStream.Read(buffor, 0,
serwer.Client.ReceiveBufferSize);
            dlugoscOdp += i;

strRead.Append(Encoding.ASCII.GetString(buffor));
            if (dlugoscOdp >= 2)
                if (strRead[dlugoscOdp - 1] == '\n' &&
strRead[dlugoscOdp - 2] == '\r') break;
        } while (true);
        string str = strRead.ToString().Substring(0,
dlugoscOdp - 2);
        serwerLogowanie odpSerwera = new
serwerLogowanie();
        try
        {
            XmlSerializer xml = new
XmlSerializer(typeof(serwerLogowanie));
            StringReader stringReader = new
StringReader(str);
            odpSerwera =
(serwerLogowanie)xml.Deserialize(stringReader);
        }

```



```

        catch (Exception ex)
        {
            MessageBox.Show("Nie udalo sie odebrac
odpowiedzi od serwera: " + ex.ToString());
        }
        if (odpSerwera.odpowiedz == 0)
            sessionID = odpSerwera.sesja;

        return odpSerwera.odpowiedz;
    }

    public int downloadFiles(string[] nazwyPlikow, string
uzytkownik)
    {
        if (uzytkownik.Length == 0) uzytkownik = ".";
        if (!serwer.Connected) return -1;
        NetworkStream stream = serwer.GetStream();
        try
        {
            downloadPliku download = new
downloadPliku(sessionID, (int)operacje.pobieranie, uzytkownik,
nazwyPlikow);
            XmlSerializer xml = new
XmlSerializer(typeof(downloadPliku));
            StringWriter stringWriter = new
StringWriter();
            xml.Serialize(stringWriter, download, names);
            string strToWrite = stringWriter.ToString() +
"\r\n";

            stream.Write(Encoding.ASCII.GetBytes(strToWrite.ToCharArray())
, 0, strToWrite.Length);
        }
        catch (Exception)
        {
            MessageBox.Show("Nie mozna wyslac zapytania o
pliki");
        }
        int dlugoscOdp = 0;
        StringBuilder strRead = new StringBuilder("");
        try
        {
            do
            {
                Byte[] bytes = new
Byte[serwer.Client.ReceiveBufferSize];
                int i = stream.Read(bytes, 0,
serwer.Client.ReceiveBufferSize);
                dlugoscOdp += i;

                strRead.Append(Encoding.ASCII.GetString(bytes));
            }
            while (stream.Read(bytes, 0,
serwer.Client.ReceiveBufferSize) > 0);
        }
        catch (Exception)
        {
            MessageBox.Show("Nie mozna odczytac odpowiedzi");
        }
    }
}

```

```

        if (dlugoscOdp >= 2)
            if (strRead[dlugoscOdp - 1] == '\n' &&
strRead[dlugoscOdp - 2] == '\r') break;
        } while (true);
    }
    catch (Exception)
    {
        MessageBox.Show("Nie mozna ustanowic
polaczenia z serwerem");
    }
    serwerPliki plikiNaSerwerze = new serwerPliki();
    try
    {
        XmlSerializer xml = new
XmlSerializer(typeof(serwerPliki));
        StringReader stringReader = new
StringReader(strRead.ToString().Substring(0, dlugoscOdp));
        plikiNaSerwerze =
(serwerPliki)xml.Deserialize(stringReader);
    }
    catch (Exception)
    {
        MessageBox.Show("Odebrano bledne dane od
serwera");
    }
    if (plikiNaSerwerze.operacja !=
(int)operacje.pobieranie) return -3;
    if (plikiNaSerwerze.odp ==
(int)odpowiedzi.wszystko_ok)
    {
        List<plikInfo> pliki = plikiNaSerwerze.plik;
        foreach (plikInfo p in pliki)
            if (p.rozmiar >= 0)
            {
                StringWriter stringWriter = new
StringWriter();
                klientOdpDownload kod = new
klientOdpDownload(sessionID, (int)operacje.pobieranie, "ok");
                XmlSerializer xml = new
XmlSerializer(typeof(klientOdpDownload));
                xml.Serialize(stringWriter, kod,
names);
                string str = stringWriter.ToString() +
"\r\n";

                stream.Write(Encoding.ASCII.GetBytes(str.ToCharArray()), 0,
str.Length);

                FileStream streamWriter = new
FileStream(folder + "\\\" + p.nazwa, FileMode.OpenOrCreate,
FileAccess.Write);

                int rozmiar = p.rozmiar;

```

```

        int tempRozmiar = rozmiar;
        while (tempRozmiar > 0)
        {
            byte[] bytes = new byte[256];
            int readSize = bytes.Length;
            if (tempRozmiar < readSize)
            {
                readSize = tempRozmiar;
            }
            stream.Read(bytes, 0, readSize);
            tempRozmiar -= readSize;

        MessageBox.Show(tempRozmiar.ToString() + " " +
        readSize.ToString());

            streamWriter.Write(bytes, 0,
        readSize);

            streamWriter.Close();
        }
        streamWriter.Close();
        strRead.Remove(0, dlugoscOdp);
        dlugoscOdp = 0;
        try
        {
            do
            {
                byte[] bytes = new
        byte[serwer.Client.ReceiveBufferSize];
                int i = stream.Read(bytes, 0,
        serwer.Client.ReceiveBufferSize);
                dlugoscOdp += i;

        strRead.Append(Encoding.ASCII.GetString(bytes));
                if (dlugoscOdp >= 2)
                    if (strRead[dlugoscOdp -
        1] == '\n' && strRead[dlugoscOdp - 2] == '\r') break;
                } while (true);
            }
            catch (Exception)
            {
                MessageBox.Show("Blad podczas
        odbierania hasha pliku");
            }
            try
            {
                XmlSerializer xmlHash = new
        XmlSerializer(typeof(serwerHash));
                serwerHash shash = new
        serwerHash();
                shash =
        (serwerHash)xmlHash.Deserialize(new

```

```

StringReader(strRead.ToString().Substring(0, dlugoscOdp -
2)));
        }
        catch (Exception)
        {
            MessageBox.Show("Blad podczas
parsowania hasha z pliku");
        }
    }
    return (int)plikiNaSerwerze.odp;
}

public int uploadFile(string nazwa, DateTime data, int
rozmiar)
{
    if (!File.Exists(folder + "\\\" + nazwa)) return -
4;

    if (!serwer.Connected) return -1;
    NetworkStream networkStream = serwer.GetStream();
    try
    {
        StringWriter stringWriter = new
StringWriter();
        XmlSerializer xml = new
XmlSerializer(typeof(klientUpload));
        klientUpload upload = new
klientUpload(sessionID, (int)operacje.wysylanie, nazwa, data,
rozmiar, 1);
        xml.Serialize(stringWriter, upload, names);
        string str = stringWriter.ToString() + "\r\n";
        MessageBox.Show(str);

        networkStream.Write(Encoding.ASCII.GetBytes(str.ToCharArray())
, 0, str.Length);
    }
    catch (Exception)
    {
        MessageBox.Show("Nie mozna wyslac pliku");
    }
    int dlugoscOdp = 0;
    serwerBase odpSerwera = new serwerBase();
    try
    {
        StringReader stringReader = new
StringReader("");
        StringBuilder stringBuilder = new
StringBuilder("");
        while (true)
        {

```

```

        Byte[] bytes = new
Byte[serwer.Client.ReceiveBufferSize];
        int i = networkStream.Read(bytes, 0,
serwer.Client.ReceiveBufferSize);
        dlugoscOdp += i;

StringBuilder.Append(Encoding.ASCII.GetString(bytes));
        if (dlugoscOdp >= 2)
            if (StringBuilder[dlugoscOdp - 1] ==
'\n' && StringBuilder[dlugoscOdp - 2] == '\r') break;
        }
        XmlSerializer xml = new
XmlSerializer(typeof(serwerBase));
        odpSerwera = (serwerBase)xml.Deserialize(new
StringReader(StringBuilder.ToString().Substring(0, dlugoscOdp
- 2)));
    }
    catch (Exception)
    {
        MessageBox.Show("Blad podczas odbierania
danych z serwera");
    }
    if (odpSerwera.operacja !=
(int)operacje.wysylanie) return -3;
    if (odpSerwera.odp ==
(int)odpowiedzi.plik_istnieje)
    {
        string odp;
        if (MessageBox.Show("Plik " + nazwa + "
istnieje na serwerze, czy zastapic go?", "",
MessageBoxButtons.YesNo) == DialogResult.Yes)
        {
            odp = "ok";
        }
        else
        {
            odp = "abort";
        }
        StringWriter stringWriter = new
StringWriter();
        XmlSerializer xml = new
XmlSerializer(typeof(klientOdpDownload));
        klientOdpDownload kod = new
klientOdpDownload(sessionID, (int)operacje.wysylanie, odp);
        xml.Serialize(stringWriter, kod, names);
        string str = stringWriter.ToString() + "\r\n";

networkStream.Write(Encoding.ASCII.GetBytes(str.ToCharArray())
, 0, str.Length);
        try
        {

```

```

        StringReader stringReader = new
StringReader("");
        StringBuilder stringBuilder = new
StringBuilder("");
        while (true)
        {
            Byte[] bytes = new
Byte[serwer.Client.ReceiveBufferSize];
            int i = networkStream.Read(bytes, 0,
serwer.Client.ReceiveBufferSize);
            dlugoscOdp += i;

stringBuilder.Append(Encoding.ASCII.GetString(bytes));
            if (dlugoscOdp >= 2)
                if (stringBuilder[dlugoscOdp - 1]
== '\n' && stringBuilder[dlugoscOdp - 2] == '\r') break;
        }
        XmlSerializer x = new
XmlSerializer(typeof(serwerBase));
        odpSerwera = (serwerBase)x.Deserialize(new
StreamReader(stringBuilder.ToString().Substring(0, dlugoscOdp
- 2)));
    }
    catch (Exception)
    {
        MessageBox.Show("Blad podczas odbierania
danych z serwera");
    }
}
if (odpSerwera.operacja !=
(int)operacje.wysylanie) return -3;
if (odpSerwera.odp == (int)odpowiedzi.wszystko_ok
|| odpSerwera.odp == (int)odpowiedzi.plik_nie_istnieje)
{
    int rozmiarUp = 256;
    int rozmiarTmp = rozmiar;
    FileStream fileStream = new FileStream(folder
+ "/" + nazwa, FileMode.Open, FileAccess.Read);
    while (rozmiarTmp != 0)
    {
        StringReader stringReader = new
StringReader("");
        if (rozmiarTmp < rozmiarUp)
            rozmiarUp = rozmiarTmp;
        byte[] bytes = new byte[rozmiarUp];
        fileStream.Read(bytes, 0, rozmiarUp);

        MessageBox.Show(Encoding.ASCII.GetString(bytes));
        networkStream.Write(bytes, 0, rozmiarUp);
        rozmiarTmp -= rozmiarUp;
    }
}

```

```

        fileStream.Close();
        try
        {
            XmlSerializer xml = new
XmlSerializer(typeof(klientHash));
            klientHash khash = new
klientHash(sessionID, (int)operacje.wysylanie, "-1");
            StringWriter sw = new StringWriter();
            xml.Serialize(sw, khash, names);
            string str = sw.ToString() + "\r\n";

networkStream.Write(Encoding.ASCII.GetBytes(str.ToCharArray())
, 0, str.Length);
        }
        catch (Exception)
        {
            MessageBox.Show("Blad podczas wysylania
hasza");
        }
    }
    return odpSerwera.odp;
}

public List<plikInfo> downloadListy(string uzytkownik)
{
    if (!serwer.Connected) return null;
    NetworkStream networkStream = serwer.GetStream();
    if (uzytkownik.Length == 0) uzytkownik = ".";
    try
    {
        listaPlikow lista = new listaPlikow(sessionID,
(int)operacje.lista, uzytkownik);
        XmlSerializer xml = new
XmlSerializer(typeof(listaPlikow));
        StringWriter stringWriter = new
StringWriter();
        xml.Serialize(stringWriter, lista, names);
        string str = stringWriter.ToString() + "\r\n";

networkStream.Write(Encoding.ASCII.GetBytes(str), 0,
str.Length);
    }
    catch (Exception)
    {
        MessageBox.Show("Nie mozna pobrac listy
plikow");
    }
    serwerPliki pliki = new serwerPliki();
    try
    {

```

```

        XmlSerializer xml = new
XmlSerializer(typeof(serwerPliki));
        StringReader stringReader = new
StringReader("");
        int dlugoscOdp = 0;
        StringBuilder stringBuilder = new
StringBuilder("");
        while (true)
        {
            byte[] bytes = new
byte[serwer.Client.ReceiveBufferSize];
            int i = networkStream.Read(bytes, 0,
serwer.Client.ReceiveBufferSize);
            dlugoscOdp += i;

stringBuilder.Append(Encoding.ASCII.GetString(bytes));
            if (dlugoscOdp > 2)
                if (stringBuilder[dlugoscOdp - 2] ==
'\r' && stringBuilder[dlugoscOdp - 1] == '\n') break;
            }
            string str =
stringBuilder.ToString().Substring(0, dlugoscOdp - 2);
            pliki = (serwerPliki)xml.Deserialize(new
StringReader(str));
        }
        catch (Exception)
        {
            MessageBox.Show("Blad podczas odbierania listy
plikow");
        }
        if (pliki.operacja != (int)operacje.lista) return
null;
        if (pliki.odp != (int)odpowiedzi.wszystko_ok)
return null;
        return pliki.plik;
    }
    public int usunPliki(List<pojedynczyPlik>
plikiDoUsuniecia)
    {
        if (!serwer.Connected) return -1;
        NetworkStream stream = serwer.GetStream();
        List<string> plikiDoUs=new List<string>();
        foreach(pojedynczyPlik p in plikiDoUsuniecia)
            plikiDoUs.Add(p.nazwa);
        try
        {
            klientUsun usun = new klientUsun(sessionID,
(int)operacje.usuwanie, plikiDoUs.ToArray());
            XmlSerializer xml = new
XmlSerializer(typeof(downloadPliku));

```



```

        StringWriter stringWriter = new
StringWriter();
        xml.Serialize(stringWriter, usun, names);
        string strToWrite = stringWriter.ToString() +
"\r\n";

stream.Write(Encoding.ASCII.GetBytes(strToWrite.ToCharArray())
, 0, strToWrite.Length);
    }
    catch (Exception)
    {
        MessageBox.Show("Nie mozna wyslac zapytania o
pliki");
    }
    for (int j = 0; j < plikiDoUsuniecia.Count; j++)
    {
        int dlugoscOdp = 0;
        StringBuilder strRead = new StringBuilder("");
        try
        {
            do
            {
                Byte[] bytes = new
Byte[serwer.Client.ReceiveBufferSize];
                int i = stream.Read(bytes, 0,
serwer.Client.ReceiveBufferSize);
                dlugoscOdp += i;

                strRead.Append(Encoding.ASCII.GetString(bytes));
                if (dlugoscOdp >= 2)
                    if (strRead[dlugoscOdp - 1] ==
'\n' && strRead[dlugoscOdp - 2] == '\r') break;
            } while (true);
        }
        catch (Exception)
        {
            MessageBox.Show("Nie mozna ustanowic
polaczenia z serwerem");
        }
        serwerBase odpSerw = new serwerBase();
        try
        {
            XmlSerializer xml = new
XmlSerializer(typeof(serwerBase));
            string str =
strRead.ToString().Substring(0, dlugoscOdp - 2);
            odpSerw = (serwerBase)xml.Deserialize(new
StringReader(str));
        }
        catch (Exception)
        {

```

```

        MessageBox.Show("Bład podczas usuwania");
    }
    if (odpSerw.odp ==
(int)odpowiedzi.blad_serwera || odpSerw.odp ==
(int)odpowiedzi.bledny_numer_sesji) return odpSerw.odp;
    }
    return 1;
}

public string Login
{
    set
    {
        log = value;
    }
}
public string Haslo
{
    set
    {
        haslo = value;
    }
}
}
}

```

Program.cs

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace ASS8.Klient
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();

            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new ASS8___Logowanie());
        }
    }
}

```

XmlPlik

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using System.Xml.Serialization;

namespace ASS8.Klient
{
    public class pojedynczyPlik
    {
        private string pNazwa;
        private string pHash;
        public pojedynczyPlik() { }
        public pojedynczyPlik(string p, string h)
        {
            nazwa = p;
            hash = h;
        }
        [XmlAttribute]
        public string nazwa
        {
            get
            {
                return pNazwa;
            }
            set
            {
                pNazwa = value;
            }
        }
        [XmlAttribute]
        public string hash
        {
            get
            {
                return pHash;
            }
            set
            {
                pHash = value;
            }
        }
    }

    public class pliki
    {

```

```

private List<pojedynczyPlik> pPliki;
public pliki() { }
public pliki(List<pojedynczyPlik> p)
{
    plik = new List<pojedynczyPlik>();
    foreach (pojedynczyPlik tmp in p)
        plik.Add(tmp);
}
[XmlElement]
public List<pojedynczyPlik> plik
{
    get
    {
        return pPliki;
    }
    set
    {
        pPliki = value;
    }
}
}
}

```

XmlRequestsClass

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using System.Xml.Serialization;

namespace ASS8.Klient
{
    [XmlRoot("klient")]
    public class klientBase
    {
        private int idSesji;
        private int op;
        public klientBase() { }
        public klientBase(int id, int oper)
        {
            idsesji = id;
            operacja = oper;
        }
        [XmlAttribute]
        public int idsesji
        {
            get
            {
                return idSesji;
            }
        }
    }
}

```

```

        set
        {
            idSesji = value;
        }
    }
    [XmlAttribute]
    public int operacja
    {
        get
        {
            return op;
        }
        set
        {
            op = value;
        }
    }
}
[XmlRoot("plik")]
public class plikBase
{
    private string pNazwa;
    public plikBase() { }
    public plikBase(string n)
    {
        nazwa = n;
    }
    [XmlAttribute]
    public string nazwa
    {
        get
        {
            return pNazwa;
        }
        set
        {
            pNazwa = value;
        }
    }
}
[XmlRoot("plik")]
public class plikInfo : plikBase
{
    private DateTime pData;
    private int pRozmiar;
    private int pDostep;
    public plikInfo() { }
    public plikInfo(string n, DateTime d, int r, int dos)
        : base(n)
    {
        pData = d;
    }
}

```

```

        rozmiar = r;
        pDostep = dos;
    }
    [XmlAttribute]
    public DateTime data
    {
        get
        {
            return pData;
        }
        set
        {
            pData = value;
        }
    }
    [XmlAttribute]
    public int rozmiar
    {
        get
        {
            return pRozmiar;
        }
        set
        {
            pRozmiar = value;
        }
    }
    [XmlAttribute]
    public int dostep
    {
        get
        {
            return pDostep;
        }
        set
        {
            pDostep = value;
        }
    }
}
[XmlRoot("logowanie")]
public class klientLogowanie
{
    public klientLogowanie() { }
    public klientLogowanie(string l, string h, string v)
    {
        Login = l;
        Haslo = h;
        wersjaKlienta = v;
    }
    [XmlAttribute("login")]

```

```

public string Login
{
    get
    {
        return login;
    }
    set
    {
        login = value;
    }
}
[XmlAttribute("haslo")]
public string Haslo
{
    get
    {
        return haslo;
    }
    set
    {
        haslo = value;
    }
}
[XmlAttribute("wersja")]
public string wersjaKlienta
{
    get
    {
        return vKlienta;
    }
    set
    {
        vKlienta = value;
    }
}

private string login;
private string haslo;
private string vKlienta;
}
[XmlRoot("klient")]
public class listaPlikow : klientBase
{
    private string uzyt;
    public listaPlikow() { }
    public listaPlikow(int id, int oper, string u)
        : base(id, oper)
    {
        uzytkownik = u;
    }
    [XmlAttribute]

```

```

        public string uzytkownik
        {
            get
            {
                return uzyt;
            }
            set
            {
                uzyt = value;
            }
        }
    }
    [XmlRoot("klient")]
    public class downloadPliku : listaPlikow
    {
        private List<plikBase> pPlik;
        public downloadPliku() { }
        public downloadPliku(int id, int oper, string u,
string[] nazwa)
            : base(id, oper, u)
        {
            plik = new List<plikBase>();
            foreach (string s in nazwa)
                plik.Add(new plikBase(s));
        }
        [XmlElement("plik")]
        public List<plikBase> plik
        {
            get
            {
                return pPlik;
            }
            set
            {
                pPlik = value;
            }
        }
    }
}
[XmlRoot("klient")]
public class klientOdpDownload : klientBase
{
    private string pAction;
    public klientOdpDownload() { }
    public klientOdpDownload(int id, int oper, string act)
        : base(id, oper)
    {
        action = act;
    }
    [XmlAttribute]
    public string action
    {

```



```

        get
        {
            return pAction;
        }
        set
        {
            pAction = value;
        }
    }

}

[XmlRoot("klient")]
public class klientUpload : klientBase
{
    plikInfo pPlik;
    public klientUpload()
    {
    }
    public klientUpload(int id, int oper, string n,
DateTime d, int r, int dos)
        : base(id, oper)
    {
        plik = new plikInfo(n, d, r, dos);
    }
    [XmlElement]
    public plikInfo plik
    {
        get
        {
            return pPlik;
        }
        set
        {
            pPlik = value;
        }
    }
}

[XmlRoot("klient")]
public class klientHash : klientBase
{
    private string pHash;
    public klientHash() { }
    public klientHash(int id, int oper, string h)
        : base(id, oper)
    {
        hash = h;
    }
    [XmlAttribute]
    public string hash
    {
        get

```

```

        {
            return pHash;
        }
        set
        {
            pHash = value;
        }
    }
}
[XmlRoot("klient")]
public class klientUsun : klientBase
{
    List<plikBase> pPlik;
    public klientUsun() { }
    public klientUsun(int id, int oper, string[] n)
        : base(id, oper)
    {
        plik = new List<plikBase>();
        foreach (string s in n)
            pPlik.Add(new plikBase(s));
    }
    [XmlElement]
    public List<plikBase> plik
    {
        get
        {
            return pPlik;
        }
        set
        {
            pPlik = value;
        }
    }
}

```

```

[XmlRoot("serwer")]
public class serwerBase
{
    private int pOdpowiedz;
    private int pOp;
    public serwerBase() { }
    public serwerBase(int o, int op)
    {
        odp = o;
        operacja = op;
    }
    [XmlAttribute]
    public int odp

```

```

    {
        get
        {
            return pOdpowiedz;
        }
        set
        {
            pOdpowiedz = value;
        }
    }
    [XmlAttribute]
    public int operacja
    {
        get
        {
            return pOp;
        }
        set
        {
            pOp = value;
        }
    }
}

[XmlRoot("serwer")]
public class serwerLogowanie
{
    private int pOdpowiedz;
    private int pSesja;
    private string pWersja;
    public serwerLogowanie() { }
    public serwerLogowanie(int odp, int ses, string wer)
    {
        odpowiedz = odp;
        sesja = ses;
        wersja = wer;
    }
    [XmlAttribute]
    public int odpowiedz
    {
        get
        {
            return pOdpowiedz;
        }
        set
        {
            pOdpowiedz = value;
        }
    }
    [XmlAttribute]
    public int sesja

```

```

    {
        get
        {
            return pSesja;
        }
        set
        {
            pSesja = value;
        }
    }
    [XmlAttribute]
    public string wersja
    {
        get
        {
            return pWersja;
        }
        set
        {
            pWersja = value;
        }
    }
}

[XmlRoot("serwer")]
public class serwerPliki : serwerBase
{
    private List<plikInfo> pPlik;
    public serwerPliki() { }
    [XmlElement]
    public List<plikInfo> plik
    {
        get
        {
            return pPlik;
        }
        set
        {
            pPlik = value;
        }
    }
}

[XmlRoot("serwer")]
public class serwerHash : serwerBase
{
    private string pHash;
    public serwerHash() { }
    public serwerHash(int o, int op, string h)
        : base(o, op)
    {
        hash = h;
    }
}

```

```

    }
    [XmlAttribute]
    public string hash
    {
        get
        {
            return pHash;
        }
        set
        {
            pHash = value;
        }
    }
}
}
}

```

Zarzadca

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using System.Xml.Serialization;
using System.IO;
using System.Windows.Forms;
using System.Security.Cryptography;
namespace ASS8.Klient
{
    class zarzadca
    {
        private komunikacja k;
        string folder = "pliki";
        XmlSerializerNamespaces names;
        public zarzadca()
        {
            names = new XmlSerializerNamespaces();
            names.Add("", "");
            if (!File.Exists("pliki.xml"))
            {
                TextWriter sw = new StreamWriter("pliki.xml",
false);
                pliki p = new pliki(new
List<pojedynczyPlik>());
                XmlSerializer xml = new
XmlSerializer(typeof(pliki));
                xml.Serialize(sw, p, names);
                sw.Close();
            }
        }
    }
}

```

```

        // List<plikInfo> sprawdzNoweLokalnePliki(plikInfo
plik) { }
        private List<plikInfo> sprawdzRoznice(List<plikInfo>
plikidoPorownywania, pliki plikiPorownywane)
        {
            List<plikInfo> ret = new List<plikInfo>();
            foreach (plikInfo p in plikiDoPorownywania)
            {
                bool jest = false;
                foreach (pojedynczyPlik p2 in
plikidoPorownywane.plik)
                {
                    int comp = p.nazwa.CompareTo(p2.nazwa);
                    if (comp == 0)
                    {
                        jest = true;
                        break;
                    }
                    if (comp < 0) break;
                }
                if (!jest)
                    ret.Add(p);
            }
            return ret;
        }

        private List<pojedynczyPlik> sprawdzRoznice(pliki
plikidoPorownywania, List<plikInfo> plikiPorownywane)
        {
            List<pojedynczyPlik> ret = new
List<pojedynczyPlik>();
            foreach (pojedynczyPlik p in
plikidoPorownywania.plik)
            {
                bool jest = false;
                foreach (plikInfo p2 in plikiPorownywane)
                {
                    int comp = p.nazwa.CompareTo(p2.nazwa);
                    if (comp == 0)
                    {
                        jest = true;
                        break;
                    }
                    if (comp < 0) break;
                }
                if (!jest)
                    ret.Add(p);
            }
            return ret;
        }
    }

```

```

        private List<pojedynczyPlik> sprawdzRoznice(pliki
plikiDoPorownywania, pliki plikiPorownywane)
        {
            List<pojedynczyPlik> ret = new
List<pojedynczyPlik>();
            foreach (pojedynczyPlik p in
plikiDoPorownywania.plik)
            {
                bool jest = false;
                foreach (pojedynczyPlik p2 in
plikiPorownywane.plik)
                {
                    int comp = p.nazwa.CompareTo(p2.nazwa);
                    if (comp == 0)
                    {
                        jest = true;
                        break;
                    }
                    if (comp < 0) break;
                }
                if (!jest)
                    ret.Add(p);
            }
            return ret;
        }

        private void zapiszInfoPlikow()
        {
            DirectoryInfo di = new DirectoryInfo(folder);
            FileInfo[] files = di.GetFiles();
            List<pojedynczyPlik> pp = new
List<pojedynczyPlik>();
            foreach (FileInfo fi in files)
            {
                FileStream fs = new FileStream(folder + "/" +
fi.Name, FileMode.Open);
                MD5 md5 = new MD5CryptoServiceProvider();
                byte[] hash = md5.ComputeHash(fs);
                fs.Close();
                pp.Add(new pojedynczyPlik(fi.Name,
"")); //Encoding.ASCII.GetString(hash));
            }
            StreamWriter sw = new StreamWriter("pliki.xml",
false);

            pliki p = new pliki(pp);
            XmlSerializer xml = new
XmlSerializer(typeof(pliki));
            xml.Serialize(sw, p, names);
            sw.Close();
        }
    }

```

```

        private void odbierzPliki(List<plikInfo>
plikidoSciagniecia)
        {
            List<string> str = new List<string>();
            foreach (plikInfo pi in plikidoSciagniecia)
                str.Add(pi.nazwa);
            k.downloadFiles(str.ToArray(), "");
        }
        private void wyslijPlik(List<pojedynczyPlik>
plikidoWyslania)
        {
            foreach (pojedynczyPlik p in plikidoWyslania)
            {
                FileInfo fi = new FileInfo(folder + "/" +
p.nazwa);
                k.uploadFile(p.nazwa, fi.LastWriteTime,
(int)fi.Length);
            }
        }
        /*private void usunPlikiSerwer(List<pojedynczyPlik>
plikidoUsuniecia)
        {
            foreach (pojedynczyPlik p in plikidoWyslania)
            {
                FileInfo fi = new FileInfo(folder + "/" +
p.nazwa);
                k.uploadFile(p.nazwa, fi.LastWriteTime,
(int)fi.Length);
            }
        }*/
        private pliki plikiZapisane()
        {
            pliki plikiLokalnie = new pliki(new
List<pojedynczyPlik>());
            try
            {
                TextReader plikXml = new
StreamReader("pliki.xml");
                XmlSerializer xml = new
XmlSerializer(typeof(pliki));
                plikiLokalnie =
(pliki)xml.Deserialize(plikXml);
                plikXml.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Nie mozna bylo odczytac pliku
konfiguracyjnego: " + ex.ToString());
            }
            return plikiLokalnie;
        }
    }

```



```

private pliki plikiKatalog()
{
    pliki p = new pliki(new List<pojedynczyPlik>());
    try
    {
        DirectoryInfo di = new DirectoryInfo(folder);
        FileInfo[] files = di.GetFiles();
        foreach (FileInfo fi in files)
            p.plik.Add(new pojedynczyPlik(fi.Name,
""));
    }
    catch (Exception) { MessageBox.Show("Nie mozna
pobrac listy plikow"); }
    return p;
}
public void szukajZmian()
{
    List<plikInfo> plikiNaSerwerze =
k.downloadListy("");
    if (plikiNaSerwerze == null) return;
    pliki plikiLokalnieZapis = plikiZapisane();
    pliki plikiLokalnieKat = plikiKatalog();
    if (plikiLokalnieKat == null || plikiLokalnieZapis
== null) return;
    List<plikInfo> roznicaSerwerLok =
sprawdzRoznice(plikiNaSerwerze, plikiLokalnieZapis);
    List<pojedynczyPlik> roznicaLokSerwer =
sprawdzRoznice(plikiLokalnieKat, plikiNaSerwerze);
    List<pojedynczyPlik> roznicaLokKat =
sprawdzRoznice(plikiLokalnieZapis, plikiLokalnieKat);
    if (roznicaSerwerLok.Count != 0)
        odbierzPliki(roznicaSerwerLok);
    if (roznicaLokSerwer.Count != 0)
        wyslijPlik(roznicaLokSerwer);
    /*if(roznicaLokKat.Count!=0)
        usunPliki(roznicaLokKat);*/
    zapiszInfoPlikow();
}

public komunikacja kom
{
    set
    {
        k = value;
    }
}
}
}

```

## Kod źródłowy klienta wx:

ASS8\_Klient\_v1App.h

```

/*****
**
* Name:      ASS8_Klient_v1App.h
* Purpose:   Defines Application Class
* Author:    Paweł, Zembrzuski ()
* Created:   2009-04-17
* Copyright: Paweł, Zembrzuski ()
* License:
**
*****/

/

#ifndef ASS8_KLIENT_V1APP_H
#define ASS8_KLIENT_V1APP_H

#include <wx/app.h>

class ASS8_Klient_v1App : public wxApp
{
    public:
        virtual bool OnInit();
};

#endif // ASS8_KLIENT_V1APP_H
```

ASS8\_Klient\_v1Main.h

```

/*****
**
* Name:      ASS8_Klient_v1Main.h
* Purpose:   Defines Application Frame
* Author:    Paweł, Zembrzuski ()
* Created:   2009-04-17
* Copyright: Paweł, Zembrzuski ()
* License:
**
*****/

/

#ifndef ASS8_KLIENT_V1MAIN_H
#define ASS8_KLIENT_V1MAIN_H

//(*Headers(ASS8_Klient_v1Frame)
#include <wx/sizer.h>
#include <wx/menu.h>
```

```

#include <wx/textctrl.h>
#include <wx/panel.h>
#include <wx/button.h>
#include <wx/frame.h>
#include <wx/statusbr.h>
/**)

class ASS8_Klient_v1Frame: public wxFrame
{
    public:

        ASS8_Klient_v1Frame(wxWindow* parent,wxWindowID id = -
1);
        virtual ~ASS8_Klient_v1Frame();

    private:

        /**)Handlers(ASS8_Klient_v1Frame)
        void OnQuit(wxCommandEvent& event);
        void OnAbout(wxCommandEvent& event);
        /**)

        /**)Identifiers(ASS8_Klient_v1Frame)
        static const long ID_TEXTCTRL1;
        static const long ID_TEXTCTRL2;
        static const long ID_BUTTON1;
        static const long ID_PANEL1;
        static const long idMenuQuit;
        static const long idMenuAbout;
        static const long ID_STATUSBAR1;
        /**)

        /**)Declarations(ASS8_Klient_v1Frame)
        wxButton* Button1;
        wxPanel* Panel1;
        wxStatusBar* StatusBar1;
        wxTextCtrl* TextCtrl2;
        wxTextCtrl* TextCtrl1;
        /**)

        DECLARE_EVENT_TABLE()
};

#endif // ASS8_KLIENT_V1MAIN_H

```

Konfiguracja.h

```

#ifndef KONFIGURACJA_H
#define KONFIGURACJA_H

```

```

#ifndef WX_PRECOMP
    //(*HeadersPCH(Konfiguracja)
    #include <wx/sizer.h>
    #include <wx/textctrl.h>
    #include <wx/panel.h>
    #include <wx/frame.h>
    //*)
#endif
//(*Headers(Konfiguracja)
//*)

class Konfiguracja: public wxFrame
{
    public:

        Konfiguracja(wxWindow* parent=0,wxWindowID
id=wxID_ANY,const wxPoint& pos=wxDefaultPosition,const wxSize&
size=wxDefaultSize);
        virtual ~Konfiguracja();

        //(*Declarations(Konfiguracja)
        wxPanel* Panell;
        wxTextCtrl* TextCtrl1;
        //*)

    protected:

        //(*Identifiers(Konfiguracja)
        static const long ID_TEXTCTRL1;
        static const long ID_PANEL1;
        //*)

    private:

        //(*Handlers(Konfiguracja)
        //*)

        DECLARE_EVENT_TABLE( )
};

#endif

```

NewFrame.h

```

#ifndef NEWFRAME_H
#define NEWFRAME_H

#ifndef WX_PRECOMP
    //(*HeadersPCH(NewFrame)
    #include <wx/sizer.h>
    #include <wx/stattext.h>

```

```

#include <wx/textctrl.h>
#include <wx/panel.h>
#include <wx/bmpbtn.h>
#include <wx/dirdlg.h>
#include <wx/frame.h>
/**)
#endif
/**(*Headers(NewFrame)
#include <wx/spinctrl.h>
/**)

class NewFrame: public wxFrame
{
    public:

        NewFrame(wxWindow* parent=0,wxWindowID
id=wxID_ANY,const wxPoint& pos=wxDefaultPosition,const wxSize&
size=wxDefaultSize);
        virtual ~NewFrame();

        /**(*Declarations(NewFrame)
wxSpinCtrl* SpinCtrl1;
wxStaticText* StaticText2;
wxPanel* Panel1;
wxStaticText* StaticText1;
wxStaticText* StaticText3;
wxBitmapButton* BitmapButton1;
wxSpinCtrl* SpinCtrl3;
wxDirDialog* DirDialog1;
wxSpinCtrl* SpinCtrl2;
wxTextCtrl* TextCtrl1;
/**)

    protected:

        /**(*Identifiers(NewFrame)
static const long ID_TEXTCTRL1;
static const long ID_BITMAPBUTTON1;
static const long ID_STATICTEXT1;
static const long ID_SPINCTRL1;
static const long ID_STATICTEXT2;
static const long ID_SPINCTRL2;
static const long ID_STATICTEXT3;
static const long ID_SPINCTRL3;
static const long ID_PANEL1;
/**)

    private:

        /**(*Handlers(NewFrame)
/**)

```

```

        DECLARE_EVENT_TABLE()
};

#endif

pobieranie_plikow.h

#ifndef POBIERANIE_PLIKOW_H
#define POBIERANIE_PLIKOW_H

#ifndef WX_PRECOMP
    /**(*Headers(pobieranie_plikow)
    #include <wx/sizer.h>
    #include <wx/textctrl.h>
    #include <wx/panel.h>
    #include <wx/frame.h>
    /**)
#endif
/**(*Headers(pobieranie_plikow)
#include <wx/treectrl.h>
/**)

class pobieranie_plikow: public wxFrame
{
    public:

        pobieranie_plikow(wxWindow* parent=0,wxWindowID
id=wxID_ANY,const wxPoint& pos=wxDefaultPosition,const wxSize&
size=wxDefaultSize);
        virtual ~pobieranie_plikow();

        /**(*Declarations(pobieranie_plikow)
        wxPanel* Panell1;
        wxTreeCtrl* TreeCtrl1;
        wxTextCtrl* TextCtrl1;
        /**)

    protected:

        /**(*Identifiers(pobieranie_plikow)
        static const long ID_TEXTCTRL1;
        static const long ID_TREECTRL1;
        static const long ID_PANEL1;
        /**)

    private:

        /**(*Handlers(pobieranie_plikow)
        /**)

```

```

        DECLARE_EVENT_TABLE( )
};

#endif

wx_pch.h

/*****
**
* Name:      wx_pch.h
* Purpose:   Header to create Pre-Compiled Header (PCH)
* Author:    Paweł, Zembrzuski ( )
* Created:   2009-04-17
* Copyright: Paweł, Zembrzuski ( )
* License:
**
*****/

/

#ifndef WX_PCH_H_INCLUDED
#define WX_PCH_H_INCLUDED

// basic wxWidgets headers
#include <wx/wxprec.h>

#ifdef __BORLANDC__
    #pragma hdrstop
#endif

#ifndef WX_PRECOMP
    #include <wx/wx.h>
#endif

#ifdef WX_PRECOMP
    // put here all your rarely-changing header files
#endif // WX_PRECOMP

#endif // WX_PCH_H_INCLUDED

ASS8_Klient_v1App.cpp

/*****
**
* Name:      ASS8_Klient_v1App.cpp
* Purpose:   Code for Application Class
* Author:    Paweł, Zembrzuski ( )
* Created:   2009-04-17
* Copyright: Paweł, Zembrzuski ( )
* License:
**
*****/

```

```
*****
/
```

```
#include "wx_pch.h"
#include "ASS8_Klient_v1App.h"
```

```
/*(*AppHeaders
#include "ASS8_Klient_v1Main.h"
#include <wx/image.h>
/*)
```

```
IMPLEMENT_APP(ASS8_Klient_v1App);
```

```
bool ASS8_Klient_v1App::OnInit()
{
    /*(*AppInitialize
    bool wxsOK = true;
    wxInitAllImageHandlers();
    if ( wxsOK )
    {
        ASS8_Klient_v1Frame* Frame = new ASS8_Klient_v1Frame(0);
        Frame->Show();
        SetTopWindow(Frame);
    }
    /*)
    return wxsOK;

}
```

```
ASS8_Klient_v1Main.cpp
```

```
/******
**
```

```
* Name:      ASS8_Klient_v1Main.cpp
* Purpose:   Code for Application Frame
* Author:    Paweł, Zembrzuski ()
* Created:   2009-04-17
* Copyright: Paweł, Zembrzuski ()
* License:
```

```
*****
/
```

```
#include "wx_pch.h"
#include "ASS8_Klient_v1Main.h"
#include <wx/msgdlg.h>
```

```
/*(*InternalHeaders(ASS8_Klient_v1Frame)
#include <wx/intl.h>
#include <wx/string.h>
```



```

/**)

//helper functions
enum wxbuildinfoformat {
    short_f, long_f };

wxString wxbuildinfo(wxbuildinfoformat format)
{
    wxString wxbuild(wxVERSION_STRING);

    if (format == long_f )
    {
#ifdef __WXMSW__
        wxbuild << _T("-Windows");
#elif defined(__UNIX__)
        wxbuild << _T("-Linux");
#endif

#ifdef wxUSE_UNICODE
        wxbuild << _T("-Unicode build");
#else
        wxbuild << _T("-ANSI build");
#endif // wxUSE_UNICODE
    }

    return wxbuild;
}

/**(*IdInit(ASS8_Klient_v1Frame)
const long ASS8_Klient_v1Frame::ID_TEXTCTRL1 = wxNewId();
const long ASS8_Klient_v1Frame::ID_TEXTCTRL2 = wxNewId();
const long ASS8_Klient_v1Frame::ID_BUTTON1 = wxNewId();
const long ASS8_Klient_v1Frame::ID_PANEL1 = wxNewId();
const long ASS8_Klient_v1Frame::idMenuQuit = wxNewId();
const long ASS8_Klient_v1Frame::idMenuAbout = wxNewId();
const long ASS8_Klient_v1Frame::ID_STATUSBAR1 = wxNewId();
/**)

BEGIN_EVENT_TABLE(ASS8_Klient_v1Frame,wxFrame)
    /**(*EventTable(ASS8_Klient_v1Frame)
    /**)
END_EVENT_TABLE()

ASS8_Klient_v1Frame::ASS8_Klient_v1Frame(wxWindow*
parent,wxWindowID id)
{
    /**(*Initialize(ASS8_Klient_v1Frame)
    wxStaticBoxSizer* StaticBoxSizer2;
    wxMenuItem* MenuItem2;
    wxMenuItem* MenuItem1;
    wxMenu* Menu1;

```

```

wxBoxSizer* BoxSizer1;
wxMenuBar* MenuBar1;
wxStaticBoxSizer* StaticBoxSizer1;
wxFlexGridSizer* FlexGridSizer1;
wxMenu* Menu2;

Create(parent, id, wxEmptyString, wxDefaultPosition,
wxDefaultSize, wxSTAY_ON_TOP|wxCAPTION|wxCLOSE_BOX, _T("id"));
BoxSizer1 = new wxBoxSizer(wxHORIZONTAL);
Panell = new wxPanel(this, ID_PANEL1, wxDefaultPosition,
wxDefaultSize, wxTAB_TRAVERSAL, _T("ID_PANEL1"));
FlexGridSizer1 = new wxFlexGridSizer(0, 1, 0, 0);
FlexGridSizer1->AddGrowableCol(0);
FlexGridSizer1->AddGrowableCol(1);
FlexGridSizer1->AddGrowableCol(2);
FlexGridSizer1->AddGrowableCol(3);
FlexGridSizer1->AddGrowableRow(0);
FlexGridSizer1->AddGrowableRow(1);
FlexGridSizer1->AddGrowableRow(2);
FlexGridSizer1->AddGrowableRow(3);
StaticBoxSizer1 = new wxStaticBoxSizer(wxHORIZONTAL,
Panell, _T("Login"));
TextCtrl1 = new wxTextCtrl(Panell, ID_TEXTCTRL1,
wxEmptyString, wxDefaultPosition, wxDefaultSize, 0,
wxDefaultValidator, _T("ID_TEXTCTRL1"));
StaticBoxSizer1->Add(TextCtrl1, 1,
wxALL|wxEXPAND|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTIC
AL, 5);
FlexGridSizer1->Add(StaticBoxSizer1, 1,
wxALL|wxEXPAND|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTIC
AL, 5);
StaticBoxSizer2 = new wxStaticBoxSizer(wxHORIZONTAL,
Panell, _T("Hasło"));
TextCtrl2 = new wxTextCtrl(Panell, ID_TEXTCTRL2,
wxEmptyString, wxDefaultPosition, wxDefaultSize,
wxTE_PASSWORD, wxDefaultValidator, _T("ID_TEXTCTRL2"));
StaticBoxSizer2->Add(TextCtrl2, 1,
wxALL|wxEXPAND|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTIC
AL, 5);
FlexGridSizer1->Add(StaticBoxSizer2, 1,
wxALL|wxEXPAND|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTIC
AL, 5);
Button1 = new wxButton(Panell, ID_BUTTON1, _T("Zaloguj"),
wxDefaultPosition, wxDefaultSize, 0, wxDefaultValidator,
_T("ID_BUTTON1"));
FlexGridSizer1->Add(Button1, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
Panell->SetSizer(FlexGridSizer1);
FlexGridSizer1->Fit(Panell);
FlexGridSizer1->SetSizeHints(Panell);

```

```

        BoxSizer1->Add(Panell1, 1,
wxALL|wxEXPAND|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        SetSizer(BoxSizer1);
        MenuBar1 = new wxMenuBar();
        Menu1 = new wxMenu();
        MenuItem1 = new wxMenuItem(Menu1, idMenuQuit,
_("Quit\tAlt-F4"), _("Quit the application"), wxITEM_NORMAL);
        Menu1->Append(MenuItem1);
        MenuBar1->Append(Menu1, _("&File"));
        Menu2 = new wxMenu();
        MenuItem2 = new wxMenuItem(Menu2, idMenuAbout,
_("About\tF1"), _("Show info about this application"),
wxITEM_NORMAL);
        Menu2->Append(MenuItem2);
        MenuBar1->Append(Menu2, _("Help"));
        SetMenuBar(MenuBar1);
        StatusBar1 = new wxStatusBar(this, ID_STATUSBAR1, 0,
_T("ID_STATUSBAR1"));
        int __wxStatusBarWidths_1[1] = { -1 };
        int __wxStatusBarStyles_1[1] = { wxSB_NORMAL };
        StatusBar1->SetFieldsCount(1,__wxStatusBarWidths_1);
        StatusBar1->SetStatusStyles(1,__wxStatusBarStyles_1);
        SetStatusBar(StatusBar1);
        BoxSizer1->Fit(this);
        BoxSizer1->SetSizeHints(this);

```

```

Connect(idMenuQuit,wxEVT_COMMAND_MENU_SELECTED,(wxObjectEventF
unction)&ASS8_Klient_v1Frame::OnQuit);

```

```

Connect(idMenuAbout,wxEVT_COMMAND_MENU_SELECTED,(wxObjectEvent
Function)&ASS8_Klient_v1Frame::OnAbout);
    /**)
}

```

```

ASS8_Klient_v1Frame::~ASS8_Klient_v1Frame()
{
    /**(*Destroy(ASS8_Klient_v1Frame)
    /**)
}

```

```

void ASS8_Klient_v1Frame::OnQuit(wxCommandEvent& event)
{
    Close();
}

```

```

void ASS8_Klient_v1Frame::OnAbout(wxCommandEvent& event)
{
    wxString msg = wxbuildinfo(long_f);
    wxMessageBox(msg, _("Welcome to..."));
}

```

```
}
```

```
Konfiguracja.cpp
```

```
#include "wx_pch.h"
```

```
#include "Konfiguracja.h"
```

```
#ifndef WX_PRECOMP
```

```
    //(*InternalHeadersPCH(Konfiguracja)
```

```
    #include <wx/intl.h>
```

```
    #include <wx/string.h>
```

```
    //*)
```

```
#endif
```

```
//(*InternalHeaders(Konfiguracja)
```

```
//*)
```

```
//(*IdInit(Konfiguracja)
```

```
const long Konfiguracja::ID_TEXTCTRL1 = wxNewId();
```

```
const long Konfiguracja::ID_PANEL1 = wxNewId();
```

```
//*)
```

```
BEGIN_EVENT_TABLE(Konfiguracja,wxFrame)
```

```
    //(*EventTable(Konfiguracja)
```

```
    //*)
```

```
END_EVENT_TABLE()
```

```
Konfiguracja::Konfiguracja(wxWindow* parent,wxWindowID
```

```
id,const wxPoint& pos,const wxSize& size)
```

```
{
```

```
    //(*Initialize(Konfiguracja)
```

```
    wxBoxSizer* BoxSizer1;
```

```
    wxStaticBoxSizer* StaticBoxSizer1;
```

```
    wxFlexGridSizer* FlexGridSizer1;
```

```
    Create(parent, id, wxEmptyString, wxDefaultPosition,  
wxDefaultSize, wxDEFAULT_FRAME_STYLE, _T("id"));
```

```
    SetClientSize(wxDefaultSize);
```

```
    Move(wxDefaultPosition);
```

```
    BoxSizer1 = new wxBoxSizer(wxHORIZONTAL);
```

```
    Panell = new wxPanel(this, ID_PANEL1, wxDefaultPosition,  
wxDefaultSize, wxTAB_TRAVERSAL, _T("ID_PANEL1"));
```

```
    FlexGridSizer1 = new wxFlexGridSizer(0, 3, 0, 0);
```

```
    StaticBoxSizer1 = new wxStaticBoxSizer(wxHORIZONTAL,  
Panell, _T("Label"));
```

```
    TextCtrl1 = new wxTextCtrl(Panell, ID_TEXTCTRL1,  
_T("Text"), wxDefaultPosition, wxDefaultSize, 0,
```

```
wxDefaultValidator, _T("ID_TEXTCTRL1"));
```

```
    StaticBoxSizer1->Add(TextCtrl1, 1,  
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
```

```
    FlexGridSizer1->Add(StaticBoxSizer1, 1,  
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
```

```

        Panell1->SetSizer(FlexGridSizer1);
        FlexGridSizer1->Fit(Panell1);
        FlexGridSizer1->SetSizeHints(Panell1);
        BoxSizer1->Add(Panell1, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        SetSizer(BoxSizer1);
        BoxSizer1->Fit(this);
        BoxSizer1->SetSizeHints(this);
        /*)
    }

Konfiguracja::~Konfiguracja()
{
    /*(*Destroy(Konfiguracja)
    /**)
}

```

NewFrame.cpp

```

#include "wx_pch.h"
#include "NewFrame.h"

#ifdef WX_PRECOMP
    /*(*InternalHeadersPCH(NewFrame)
    #include <wx/intl.h>
    #include <wx/string.h>
    /**)
#endif
/*(*InternalHeaders(NewFrame)
#include <wx/bitmap.h>
#include <wx/button.h>
#include <wx/image.h>
/**)

/*(*IdInit(NewFrame)
const long NewFrame::ID_TEXTCTRL1 = wxNewId();
const long NewFrame::ID_BITMAPBUTTON1 = wxNewId();
const long NewFrame::ID_STATICTEXT1 = wxNewId();
const long NewFrame::ID_SPINCTRL1 = wxNewId();
const long NewFrame::ID_STATICTEXT2 = wxNewId();
const long NewFrame::ID_SPINCTRL2 = wxNewId();
const long NewFrame::ID_STATICTEXT3 = wxNewId();
const long NewFrame::ID_SPINCTRL3 = wxNewId();
const long NewFrame::ID_PANEL1 = wxNewId();
/**)

BEGIN_EVENT_TABLE(NewFrame, wxFrame)
    /*(*EventTable(NewFrame)
    /**)
END_EVENT_TABLE()

```

```

NewFrame::NewFrame(wxWindow* parent,wxWindowID id,const
wxPoint& pos,const wxSize& size)
{
    ((*Initialize(NewFrame)
    wxStaticBoxSizer* StaticBoxSizer2;
    wxFlexGridSizer* FlexGridSizer4;
    wxFlexGridSizer* FlexGridSizer3;
    wxFlexGridSizer* FlexGridSizer2;
    wxStaticBoxSizer* StaticBoxSizer3;
    wxBoxSizer* BoxSizer1;
    wxStaticBoxSizer* StaticBoxSizer1;
    wxFlexGridSizer* FlexGridSizer1;
    wxStdDialogButtonSizer* StdDialogButtonSizer1;

    Create(parent, id, wxEmptyString, wxDefaultPosition,
wxDefaultSize,
wxSTAY_ON_TOP|wxCAPTION|wxCLOSE_BOX|wxMINIMIZE_BOX, _T("id"));
    SetClientSize(wxDefaultSize);
    Move(wxDefaultPosition);
    SetMinSize(wxSize(200,300));
    BoxSizer1 = new wxBoxSizer(wxHORIZONTAL);
    Panell1 = new wxPanel(this, ID_PANEL1, wxDefaultPosition,
wxDefaultSize, wxTAB_TRAVERSAL, _T("ID_PANEL1"));
    FlexGridSizer1 = new wxFlexGridSizer(0, 1, 0, 0);
    FlexGridSizer1->AddGrowableCol(0);
    FlexGridSizer1->AddGrowableCol(1);
    FlexGridSizer1->AddGrowableCol(2);
    FlexGridSizer1->AddGrowableCol(3);
    StaticBoxSizer1 = new wxStaticBoxSizer(wxHORIZONTAL,
Panell1, _T("Synchronizowany katalog"));
    FlexGridSizer2 = new wxFlexGridSizer(0, 3, 0, 0);
    FlexGridSizer2->AddGrowableCol(0);
    FlexGridSizer2->AddGrowableCol(1);
    FlexGridSizer2->AddGrowableCol(2);
    FlexGridSizer2->AddGrowableCol(3);
    TextCtrl1 = new wxTextCtrl(Panell1, ID_TEXTCTRL1,
wxEmptyString, wxDefaultPosition, wxDefaultSize, 0,
wxDefaultValidator, _T("ID_TEXTCTRL1"));
    FlexGridSizer2->Add(TextCtrl1, 1,
wxTOP|wxBOTTOM|wxLEFT|wxEXPAND|wxALIGN_CENTER_HORIZONTAL|wxALI
GN_CENTER_VERTICAL, 5);
    BitmapButton1 = new wxBitmapButton(Panell1,
ID_BITMAPBUTTON1,
wxBitmap(wxImage(_T("/home/pawel/Dokumenty/Uczelnia/grupappz/S
ource/ASS8-Klient_v1/bitmaps/open.bmp"))), wxDefaultPosition,
wxSize(35,35), wxBU_AUTODRAW, wxDefaultValidator,
_T("ID_BITMAPBUTTON1"));
    BitmapButton1->SetMaxSize(wxSize(35,35));

```

```

        FlexGridSizer2->Add(BitmapButton1, 1,
wxTOP|wxBOTTOM|wxRIGHT|wxSHAPED|wxALIGN_RIGHT|wxALIGN_CENTER_V
ERTICAL, 5);
        StaticBoxSizer1->Add(FlexGridSizer2, 1,
wxALL|wxEXPAND|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTIC
AL, 5);
        FlexGridSizer1->Add(StaticBoxSizer1, 1,
wxALL|wxEXPAND|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTIC
AL, 5);
        StaticBoxSizer2 = new wxStaticBoxSizer(wxHORIZONTAL,
Panell1, _("Ograniczenia Pobierania i wysł,ania (kB/s)"));
        FlexGridSizer3 = new wxFlexGridSizer(0, 2, 0, 0);
        StaticText1 = new wxStaticText(Panell1, ID_STATICTEXT1,
_("Maksymalna prędkość Uploadu (0 - nieograniczona)"),
wxDefaultPosition, wxDefaultSize, 0, _T("ID_STATICTEXT1"));
        FlexGridSizer3->Add(StaticText1, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        SpinCtrl1 = new wxSpinCtrl(Panell1, ID_SPINCTRL1, _T("0"),
wxDefaultPosition, wxDefaultSize, 0, 0, 100, 0,
_T("ID_SPINCTRL1"));
        SpinCtrl1->SetValue(_T("0"));
        FlexGridSizer3->Add(SpinCtrl1, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        StaticText2 = new wxStaticText(Panell1, ID_STATICTEXT2,
_("Maksymalna prędkość Downloadu (0 - nieograniczona)"),
wxDefaultPosition, wxDefaultSize, 0, _T("ID_STATICTEXT2"));
        FlexGridSizer3->Add(StaticText2, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        SpinCtrl2 = new wxSpinCtrl(Panell1, ID_SPINCTRL2, _T("0"),
wxDefaultPosition, wxDefaultSize, 0, 0, 100, 0,
_T("ID_SPINCTRL2"));
        SpinCtrl2->SetValue(_T("0"));
        FlexGridSizer3->Add(SpinCtrl2, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        StaticBoxSizer2->Add(FlexGridSizer3, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        FlexGridSizer1->Add(StaticBoxSizer2, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        StaticBoxSizer3 = new wxStaticBoxSizer(wxHORIZONTAL,
Panell1, _("Częstotliwość aktualizacji"));
        FlexGridSizer4 = new wxFlexGridSizer(0, 1, 0, 0);
        StaticText3 = new wxStaticText(Panell1, ID_STATICTEXT3,
_("Co ile czasu ma być sprawdzana aktualność plików na
serwerze i lokalnie (s)"), wxDefaultPosition, wxDefaultSize,
0, _T("ID_STATICTEXT3"));
        FlexGridSizer4->Add(StaticText3, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        SpinCtrl3 = new wxSpinCtrl(Panell1, ID_SPINCTRL3, _T("0"),
wxDefaultPosition, wxDefaultSize, 0, 0, 100, 0,
_T("ID_SPINCTRL3"));
        SpinCtrl3->SetValue(_T("0"));

```

```

        FlexGridSizer4->Add(SpinCtrl3, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        StaticBoxSizer3->Add(FlexGridSizer4, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        FlexGridSizer1->Add(StaticBoxSizer3, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        StdDialogButtonSizer1 = new wxStdDialogButtonSizer();
        StdDialogButtonSizer1->AddButton(new wxButton(Panell,
wxID_OK, wxEmptyString));
        StdDialogButtonSizer1->AddButton(new wxButton(Panell,
wxID_CANCEL, wxEmptyString));
        StdDialogButtonSizer1->Realize();
        FlexGridSizer1->Add(StdDialogButtonSizer1, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        Panell->SetSizer(FlexGridSizer1);
        FlexGridSizer1->Fit(Panell);
        FlexGridSizer1->SetSizeHints(Panell);
        BoxSizer1->Add(Panell, 1,
wxALL|wxEXPAND|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        SetSizer(BoxSizer1);
        DirDialog1 = new wxDirDialog(this, _("Select directory"),
wxEmptyString, wxDD_DEFAULT_STYLE, wxDefaultPosition,
wxDefaultSize, _T("wxDirDialog"));
        BoxSizer1->Fit(this);
        BoxSizer1->SetSizeHints(this);
        /*)
}

NewFrame::~NewFrame()
{
    /*(*Destroy(NewFrame)
    /**)
}

```

pobieranie\_plikow.cpp

```

#include "wx_pch.h"
#include "pobieranie_plikow.h"

#ifdef WX_PRECOMP
    /*(*InternalHeadersPCH(pobieranie_plikow)
    #include <wx/intl.h>
    #include <wx/string.h>
    /**)
#endif
/*(*InternalHeaders(pobieranie_plikow)
#include <wx/button.h>
/**)

```



```

//(*IdInit(pobieranie_plikow)
const long pobieranie_plikow::ID_TEXTCTRL1 = wxNewId();
const long pobieranie_plikow::ID_TREECTRL1 = wxNewId();
const long pobieranie_plikow::ID_PANEL1 = wxNewId();
//*)

BEGIN_EVENT_TABLE(pobieranie_plikow,wxFrame)
    //(*EventTable(pobieranie_plikow)
    //*)
END_EVENT_TABLE()

pobieranie_plikow::pobieranie_plikow(wxWindow*
parent,wxWindowID id,const wxPoint& pos,const wxSize& size)
{
    //(*Initialize(pobieranie_plikow)
    wxBoxSizer* BoxSizer1;
    wxFlexGridSizer* FlexGridSizer1;
    wxStdDialogButtonSizer* StdDialogButtonSizer1;

    Create(parent, id, wxEmptyString, wxDefaultPosition,
wxDefaultSize,
wxCAPTION|wxSYSTEM_MENU|wxRESIZE_BORDER|wxCLOSE_BOX|wxMAXIMIZE
_BOX|wxMINIMIZE_BOX, _T("id"));
    SetClientSize(wxDefaultSize);
    Move(wxDefaultPosition);
    BoxSizer1 = new wxBoxSizer(wxHORIZONTAL);
    Panell1 = new wxPanel(this, ID_PANEL1, wxDefaultPosition,
wxDefaultSize, wxTAB_TRAVERSAL, _T("ID_PANEL1"));
    FlexGridSizer1 = new wxFlexGridSizer(0, 1, 0, 0);
    FlexGridSizer1->AddGrowableCol(0);
    FlexGridSizer1->AddGrowableCol(1);
    FlexGridSizer1->AddGrowableCol(2);
    FlexGridSizer1->AddGrowableCol(3);
    FlexGridSizer1->AddGrowableRow(1);
    TextCtrl1 = new wxTextCtrl(Panell1, ID_TEXTCTRL1,
wxEmptyString, wxDefaultPosition, wxDefaultSize, 0,
wxDefaultValidator, _T("ID_TEXTCTRL1"));
    FlexGridSizer1->Add(TextCtrl1, 1,
wxALL|wxEXPAND|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTIC
AL, 5);
    TreeCtrl1 = new wxTreeCtrl(Panell1, ID_TREECTRL1,
wxDefaultPosition, wxDefaultSize, wxTR_DEFAULT_STYLE,
wxDefaultValidator, _T("ID_TREECTRL1"));
    FlexGridSizer1->Add(TreeCtrl1, 1,
wxALL|wxEXPAND|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTIC
AL, 5);
    StdDialogButtonSizer1 = new wxStdDialogButtonSizer();
    StdDialogButtonSizer1->AddButton(new wxButton(Panell1,
wxID_OK, wxEmptyString));
    StdDialogButtonSizer1->AddButton(new wxButton(Panell1,
wxID_CANCEL, wxEmptyString));

```

```

        StdDialogButtonSizer1->Realize();
        FlexGridSizer1->Add(StdDialogButtonSizer1, 1,
wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        Panell1->SetSizer(FlexGridSizer1);
        FlexGridSizer1->Fit(Panell1);
        FlexGridSizer1->SetSizeHints(Panell1);
        BoxSizer1->Add(Panell1, 1,
wxALL|wxEXPAND|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL, 5);
        SetSizer(BoxSizer1);
        BoxSizer1->Fit(this);
        BoxSizer1->SetSizeHints(this);
        /*)
    }

pobieranie_plikow::~pobieranie_plikow()
{
    /*(*Destroy(pobieranie_plikow)
    /**)
}

```

### Kod źródłowy serwera:

baza.cpp

```
#include "baza.hpp"
```

```

void Baza::connect(const char *server, const char *login,
const char *pass, const char *db)
{
    //const char db[]="ass8";
    info("Łączenie z Bazą.....");
    if (conn.connect(db, server, login, pass))
    {
        info("Połączono");
        return;
    }
    else
    {
        Eline2("Błąd połączenia z bazą!", conn.error());
        exit(1);
    }
}

std::string Baza::get_passwd(std::string login)
{
    info("pobieranie hasła użytkownika")
    std::string zapytanie="select password from auth_user
where username='";

```

```

        zapytanie.append(login);
        zapytanie.append("'");
        mysqlpp::Query query = conn.query(zapytanie);
        mysqlpp::StoreQueryResult res = query.store();
        if (res.num_rows()>=1)
        {
            info("Zapytanie poprawne");
            return std::string(res[0]["password"]);
        }
        else
        {
            Eline2("BŁĄD zapytania do bazy danych:
",query.error());
            return std::string("ERROR");
        }
    }
    ///Zapytanie o listę plików użytkownika po id użytkownika z
    bazy accounts_konto
    mysqlpp::StoreQueryResult Baza::getFilesList(int user_id)
    {
        info("pobieranie listy plików usera o ID = 'id'");
        std::string zapytanie="select * from files_plik where
konto_id='";
        zapytanie+=user_id;
        zapytanie.append("'");
        mysqlpp::Query query = conn.query(zapytanie);
        mysqlpp::StoreQueryResult res = query.store();
        if (res)
        {
            info("Zapytanie Poprawne");
            return res;
        }
        else
        {
            Eline2("BŁĄD zapytania do bazy danych:
",query.error());
            mysqlpp::StoreQueryResult res;
            return res;
        }
    }
    ///Zapytanie o listę plików użytkownika o nazwie podanej w
    zmiennej user
    mysqlpp::StoreQueryResult Baza::getFilesList(std::string user)
    {
        info("pobieranie listy plików usera");
        int id=getUserId(user);
        if (id>=0)
        {
            info("id poprawne");
            return getFilesList(id);
        }
    }

```

```

    }
    else
    {
        info("ID NIEPOPRAWNE!");
        mysqlpp::StoreQueryResult res;
        return res;
    }
}

///Zapytanie o ID uzytkownika o loginie 'user' ale nie o id z
auth_user tylko o id z accounts_konto
int Baza::getUserId(std::string user)
{
    info("pobranie id usera");
    //Najlpierw zapytanie o ID usera z tabeli auth_user
    std::string zapytanie="select id from auth_user where
username='";
    zapytanie.append(user);
    zapytanie.append("'");
    info2("zaytanie: ",zapytanie.c_str());
    mysqlpp::Query query = conn.query(zapytanie);//Wyslanie
zapytania do bazy
    mysqlpp::StoreQueryResult res =
query.store();//umieszczenie wynikow zapytania w zmiennej res
    //A potem zapytanie o id z tabeli accounts_konto
    if (res)
    {
        zapytanie="select id from accounts_konto where
user_id='";
        zapytanie.append(res[0]["id"]);
        zapytanie.append("'");
        mysqlpp::Query query2 = conn.query(zapytanie);
        mysqlpp::StoreQueryResult res2 = query2.store();
        if (res)
        {
            info("Zapytanie Poprawne");
            return atoi(res2[0]["id"]);
        }
        else
        {
            Eline2("BŁ,Ä...d zapytania do bazy danych:
",query.error());
            return -1;
        }
    }
    else
    {
        Eline2("BŁ,Ä...d zapytania do bazy danych:
",query.error());
        return -1;
    }
}

```

```

}

mysqlpp::StoreQueryResult Baza::getFileInfo(std::string file,
std::string user)
{
    info("pobranie info o pliku usera");
    int id=getUserId(user);
    if (id>=0)
    {
        info("id poprawne");
        return getFileInfo(file,id);
    }
    else
    {
        info("ID NIEPOPRAWNE!");
        mysqlpp::StoreQueryResult res;
        return res;
    }
}

mysqlpp::StoreQueryResult Baza::getFileInfo(std::string file,
int user_id)
{
    info("pobranie info o pliku usera o ID = 'id'");
    std::string zapytanie="select * from files_plik where
konto_id='";
    zapytanie+=user_id;
    zapytanie.append("' AND sciezka='");
    zapytanie+=file;
    zapytanie.append("'");
    mysqlpp::Query query = conn.query(zapytanie);
    mysqlpp::StoreQueryResult res = query.store();
    if (res)
    {
        info("Zapytanie Poprawne");
        return res;
    }
    else
    {
        Eline2("BŁĄD zapytania do bazy danych:
",query.error());
        mysqlpp::StoreQueryResult res;
        return res;
    }
}

void Baza::addFile(std::string nazwa, std::string konto, int
wielkosc, int hash, int prawa, int data)
{
    info("dodanie pliku do bazy")
    std::string zapytanie="SELECT count(*) from files_plik";

```

```

    int ilosc;
    int id=getUserId(konto);
    mysqlpp::Query query = conn.query(zapytanie);
    mysqlpp::StoreQueryResult res = query.store();
    if (res)
    {
        ilosc=res[0]["count(*)"];
    }
    //SELECT count(*) from files_plik;
    zapytanie="INSERT INTO files_plik VALUES('"+ilosc;
    zapytanie+=", '"+id;
    zapytanie+=", '"+nazwa;
    zapytanie+=", CURRENT_DATE";
    zapytanie+=", '"+prawa;
    zapytanie+=", '"+wielkosc;
    zapytanie+=", '"+hash;
    zapytanie.append("'");
    conn.query(zapytanie);
    info("plik dodany do bazy");

}

```

baza.hpp

```

#ifndef BAZA_HPP
#define BAZA_HPP
#include <string>
#include <mysql++.h>
#include <iostream>
#include <iomanip>
#include "debug.hpp"
class Baza
{
    private:
        mysqlpp::Connection conn;
    public:
        std::string get_passwd(std::string login);
        Baza(){};
        void connect(const char *server, const char *login,
const char *pass, const char *db);
        mysqlpp::StoreQueryResult getFilesList(int user_id);
        mysqlpp::StoreQueryResult getFilesList(std::string
user);
        mysqlpp::StoreQueryResult getFileInfo(std::string
file, std::string user);
        mysqlpp::StoreQueryResult getFileInfo(std::string
file, int user_id);
        int getUserId(std::string user);
        void addFile(std::string nazwa, std::string konto, int
wielkosc, int hash=-1, int prawa=-1, int data=-1);

```

```
};
```

```
#endif//BAZA_HPP
```

```
debug.hpp
```

```
#ifndef DEBUG_HPP
#define DEBUG_HPP
#include <cstdio>
#include <cstdlib>
#define DEBUG
#ifdef DEBUG
#define deb(arg,arg2) fprintf(stderr,(arg),(arg2));
#define line fprintf(stderr,"I: %s : %d\n",__FILE__,__LINE__);
#define Eline(arg) fprintf(stderr,"E: %s : %d : %s\n",__FILE__,__LINE__,arg);
#define Eline2(arg,arg2) fprintf(stderr,"E: %s : %d : %s\n",__FILE__,__LINE__,arg,arg2);
#define info(arg) fprintf(stderr,"I: %s : %d : %s\n",__FILE__,__LINE__,arg);
#define info2(arg,arg2) fprintf(stderr,"I: %s : %d : %s => %s\n",__FILE__,__LINE__,arg,arg2);
#else
#define info(arg)
#define deb(arg,arg2)
#define line
#define Eline
#define Eline2(arg1,arg2);
#define info2(arg,arg2)
#endif
```

```
#endif//DEBUG_HPP
```

```
Main.cpp
```

```
//
// daytime_server.cpp
// ~~~~~
//
// Copyright (c) 2003-2008 Christopher M. Kohlhoff (chris at
kohlhoff dot com)
//
// Distributed under the Boost Software License, Version 1.0.
(See accompanying
// file LICENSE_1_0.txt or copy at
http://www.boost.org/LICENSE_1_0.txt)
//

#include <iostream>
#include <string>
#include <cstdio>
```

```

#include <cstdlib>

#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>

#include <boost/asio.hpp>
#include <boost/thread/thread.hpp>
#include <boost/bind.hpp>

#include "version.h"
#include "parser.hpp"
#include "version.h"
using boost::asio::ip::tcp;

int main(int argc, char *argv[])
{
    ///Zmienna przechowujÄ...ca port na ktÄłrym serwer
nasÄł,ucuje
    unsigned int port;
    ///Zmienne przechowujÄ...ce parametry podÄł,Ä...czenia do bazy
danych
    const char *server = 0, *user = 0, *pass = 0, *db=0;
    ///Zmienna przechowujÄ...ca nazwÄ™ bazy w bazie danych
    if (argc<6)///JeÄłeli jest mniej niÄł 5 argumentÄłw
    {
        std::cout<<"ZÄł,a liczba argumentÄłw:\n\
        server <adres_serwera_mysql> <login> <haslo> <db>
<port_nasÄł,uchiwania_aserwera_ass8>"<<std::endl;
        std::cout<<"Ass8
v"<<AutoVersion::FULLVERSION_STRING<<"
"<<AutoVersion::STATUS<<std::endl;
        exit(1);///To koÄł„czy program
    }
    else
    {
        std::cout<<"Ass8
v"<<AutoVersion::FULLVERSION_STRING<<"
"<<AutoVersion::STATUS<<std::endl;
        server=argv[1];
        user=argv[2];
        pass=argv[3];
        db=argv[4];
        std::sscanf(argv[5], "%u", &port);
    }
    try
    {
        info("ustawianie...");
        ///Potrzebne do poÄł,Ä...czenia z klientem
        boost::asio::io_service io_service;

```



```

    ///Potrzebne do poŁĄczenia z klientem
    tcp::endpoint endpoint(tcp::v4(), port);
    ///Potrzebne do poŁĄczenia z klientem
    tcp::acceptor acceptor(io_service, endpoint);
    ///Watek ktory bedzie usuwal skonczone forkie
    boost::thread w1(&eat_zombie);
    info("Początek pŁtli glownej");
    while (true)///Nieskonczona pŁtla
    {
        ///WyjŁcie/WejŁcie socketa
        tcp::iostream stream;
        info("oczekiawnie na klienta");
        acceptor.accept(*stream.rdbuf());
        line;
        if (fork()==0)
        {
            info("nowy klient");
            ///utworzenie parsera w forku (dla kaŁdego
klienta jeden taki jest tworzony);
            parser p(stream,server,user,pass,db);
            p.start();
        }
    }
}
catch (std::exception& e)
{
    std::cerr << e.what() << std::endl;
}

return 0;
}

```

Parser.cpp

```

#include <iostream>
#include <string>
#include <cstdlib>
#include <cstdio>
#include <vector>

#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>

#include "version.h"
#include "parser.hpp"
#include "xml.hpp"
#include "debug.hpp"

void parser::parsuj(std::string do_parsowania)

```

```

{
    line;
#ifdef LIBXMLCPP_EXCEPTIONS_ENABLED
    try
    {
#endif //LIBXMLCPP_EXCEPTIONS_ENABLED
        xmlpp::TextReader reader((unsigned
char*)do_parsowania.c_str(),do_parsowania.size());
        while (reader.read())
        {
            info("reader.read()");

info2("reader.get_name()",reader.get_name().c_str());
            if (reader.get_name().compare("logowanie")==0)
///Logowanie - Trzeba pamiętać że parser umieszcza
dodatkowo spację na końcu sparsowanych stringów
            {
                info("Logowanie...");
                if (reader.has_attributes()) ///Sprawdzanie
ilosci atrybutów
                {
                    info("reader.has_attributes==true");
                    if (reader.move_to_attribute("login"))
///Przesun się do atrybutu "login"
                    {
                        login=reader.get_value(); ///Zapisz
login
                        info2("login",login.c_str());
                    }
                    else ///Gdy brak takiego atrybutu to
konczymy rozmowe
                    {
                        Eline("Brak loginu");
                        odpowiedz_login(3);
                    }
                    if (reader.move_to_attribute("haslo"))
///Przesun się do atrybutu "haslo"
                    {
                        haslo=reader.get_value();
                        info2("haslo",haslo.c_str());
                    }
                    else ///Gdy brak takiego atrybutu to
konczymy rozmowe
                    {
                        Eline("Brak hasla");
                        odpowiedz_login(3);
                    }

                    if
(logowanie(login,haslo)==true)///Sprawdzamy czy login i haslo
pasują...

```

```

        {
            info("Logowanie - OK");
            odpowiedz_login(0);///Jeżeli tak to
dajemy klientowi ID sesji
        }
        else
        {
            info("login i hasło nie pasują");
            odpowiedz_login(1);///Jeżeli nie to
informujemy go o tym i kończymy
        }
    }
    else ///Brak loginu i hasła,a
    {
        info("Brak loginu i hasła,a");
        odpowiedz_login(3);
    }
}
else if
(reader.get_name().compare("klient")==0)///Jeżeli polecenie
od klienta
{
    int operacja;
    std::string t_uzytkownik;
    if (reader.has_attributes()) ///Sprawdzanie
ilości atrybutów
    {
        info("reader.has_attributes==true");
        if (reader.move_to_attribute("idsesji"))
///Przesun się do atrybutu "idsesji"
        {
            ///Do debugowania
            char temp[128];
            std::string uzytkownik;
            sprintf(temp,"%u",id_sesji);
            info("Sprawdzanie id_sesji");

            info2(temp,reader.get_value().c_str());
            if
            (id_sesji==atoi(reader.get_value().c_str())) ///Sprawdzamy czy
zgadza się id_sesji
            {
                info("id_sesji POPRAWNE");
                if
                (reader.move_to_attribute("operacja"))///Sprawdzamy czy
istnieje argument "operacja"
                {

                    info2("operacja",reader.get_value().c_str());

                    operacja=atoi(reader.get_value().c_str());

```

```

    }
    else
    {
        info("BŁĄDNE ZAPYTANIE");
        Odpowiedz(400); ///BŁĄDNE
zapytanie
    }
    if (operacja==100 ||
operacja==101)
    {
        if
        (reader.move_to_attribute("uzytkownik"))
        {
            info2("uzytkownik",reader.get_value().c_str());
            uzytkownik=reader.get_value();
        }
        else
        {
            info("BŁĄDNE
ZAPYTANIE");
            Odpowiedz(400); ///BŁĄDNE
zapytanie
        }
    }
    switch (operacja)
    {
    case 100:
        info("OPERACJA 100");
        lista_plikow(uzytkownik);
        break;
    case 101:
        wysylanie_plikow(reader,uzytkownik);
        info("OPERACJA 101");
        break;
    case 102:
        info("OPERACJA 102");
        odbieranie_plikow(reader,login);
        break;
    case 103:
        info("OPERACJA 103");
        usun_pliki(reader,login);
    default:
        info("BŁĄDNY kod operacji");
        Odpowiedz(400);
        break;
    }
}

```

```

    }
    else ///Niezgodne id_sesji
    {
        info("ID SESJI JEST NIEZGODNE");
        Odpowiedz(401);
    }
}
else ///Brak id_sesji
{
    info("BRAK ID_SESJI");
    Odpowiedz(401);
}
}
else///Brak atrybutów
{
    info("brak atrybutów");
    Odpowiedz(400);
}
}
else//Jeżeli polecenie nie zaczyna się od
<klient
{
    info("nierozpoznano");
    Odpowiedz(400);
    return;
}
#ifdef LIBXMLCPP_EXCEPTIONS_ENABLED
}
catch (const std::exception& e)///Jeżeli wyjątki w
LIBXMLCPP są... włączane i wystąpi błąd parsowania
{
    info2("BŁĄD PARSOWANIA XML",e.what());
    wyslij("<?xml version=\"1.0\"?>\n
    <server xml_stream_error=\"true\"/>"); ///To wysyłana
jest o tym informacja
    exit(0);///I zamykamy połączenie
}
#endif //LIBXMLCPP_EXCEPTIONS_ENABLED

}
void parser::wyslij(std::string w)///Funkcja wysyłająca dane
przez SOCKET
{
    stream<<w<<std::endl;
}
bool parser::logowanie(std::string login, std::string
haslo)///Funkcja sprawdzająca czy dla podanego loginu hasło
jest prawidłowe, jeżeli tak to generuje id_sesji
{

```

```

        info("Logowanie()");
        info2(baza.get_passwd(login).c_str(),haslo.c_str());
        //if
        (baza.get_passwd(login).compare(0,haslo.size(),haslo)==0)///Tr
        zeba pamiÄ™taÄ™ ÄŒe zmienna haslo tak naprawde zawiera haslo i
        na koÅ„cu spacjÄ™.
        if (baza.get_passwd(login).compare(haslo)==0)///Trzeba
        pamiÄ™taÄ™ ÄŒe zmienna haslo tak naprawde zawiera haslo i na
        koÅ„cu spacjÄ™.
        {
            info("Generowanie ID sesji");
            srand(time(NULL));
            id_sesji=(unsigned int)rand();
            return true;
        }
        else
        {
            Eline("Logowanie(): niepoprawne haslo");
            return false;
        }
    }
    ///Krotka odpowiedz gdzie i - kod odpowiedzi
    void parser::Odpowiedz(int i,int nr_operacji)
    {
        sprintf(bufor,"<?xml version=\"1.0\"?>\r\n\
        <serwer operacja=\"%d\" odp=\"%d\"/>",nr_operacji,i);
        wyslij(bufor);
    }
    ///OdpowiedÅŸ z dodatkowymi informacjami  gdzie i - kod
    odpowiedzi
    void parser::Odpowiedz(int i,int nr_operacji,std::string odp)
    {
        std::string temp;
        sprintf(bufor,"<?xml version=\"1.0\"?>\r\n\
        <serwer operacja=\"%d\" odp=\"%d\">",nr_operacji,i);
        temp.append(bufor);
        if (!odp.empty())
            temp.append(odp);
        else
            temp.append("\r\n");
        temp.append("</serwer>");
        wyslij(temp);
    }
    void parser::odpowiedz_login(int i)///Funkcja wysyÅŸ,ajÄ™ca
    odpowiedziÅŸ po proÅŸbie o zalogowanie
    {
        sprintf(bufor,"<?xml version=\"1.0\"?>\r\n\
        <serwer odpowiedz=\"%d\" sesja=\"%u\" wersja=\"%s
        %s\"/>",i,id_sesji,"-
        1","aa");//AutoVersion::FULLVERSION_STRING,AutoVersion::STATUS
        _SHORT);

```

```

        wyslij(bufor);
        if (i!=0)
        {
            stream.close();
            exit(0);
        }
    }
}

void parser::start () ///Funkcja wywoł, ywana tylko raz,
rozpoczyna pobieranie informacji z SOCKETA i przekazuje je
parserowi
{
    //stream<<"ASS8 Server v
numer_wersji"/*<<AutoVersion::FULLVERSION_STRING*/<<"\r\n\r\n"
;

    std::string a;
    std::string temp;
    while (true)
    {
        a=" ";
        temp=" ";
        int zabezp=0, zabezp2=0;
        info("PETLA ZBIERAJACA");

        do
        {
            boost::system::error_code error;
            if (error == boost::asio::error::eof)
            {
                info("Klient disconnected");
                exit(0);
            }
            //info("OCZEKIWANIE NA PUSTÄ, LINIE");
            std::getline(stream,temp);
            /*if(zabezp>25)
            {
                stream.close();
            }*/

            /*if(temp.empty() || temp.compare(" ")==0)
            {
                info("linia pusta");
                //stream.clear();
            }*/
            if (temp.find_first_of("<")<=temp.size())
            {
                zabezp=0;
            }
            info2("Otrzymalem: ",temp.c_str());
            if (zabezp>3 || zabezp2>100)
                exit(0);
            ++zabezp;

```

```

        ++zabezp2;
        char x[256];
        sprintf(x, "zabezp= %d zabezp2 = %d", zabezp,
zabezp2);
        info(x);

        //std::cout<<(int)temp[0]<<"
"<<(int)temp[1]<<std::endl;
        a.append(temp+"\r\n");
    }
    while (!(temp[0]==13 && temp[1]==0));
    //zabezp=0;
    info("ROZPOCZYNAM PARSOWANIE");
    //std::cout<<a<<std::endl;
    parsuj(a);
}
stream.close();
exit(0);
}

void parser::lista_plikow(std::string uzytkownik)
{
    info("Lista_plikow");
    if (uzytkownik.compare(".")==0)
        uzytkownik=login;
    mysqlpp::StoreQueryResult
res=baza.GetFilesList(uzytkownik);
    if (res.num_rows()<1)
    {
        info("res.num_rows()<1");
        std::string odp="";
        Odpowiedz(406,100,odp);
        return;
    }
    else
    {
        info("Tworzenie listy");
        std::string odp="";
        std::string temp;
        //char temp[1024];
        for (unsigned int i=0;i<res.num_rows();++i)
        {
            //sprintf(temp,"<plik nazwa=\"%s\" data=\"%d\"
rozmiar=\"%d\"
dostep=\"%d\"/>\n",res[i]["sciezka"],res[i]["dataDodania"],res
[i]["wielkosc"],res[i]["prawaDostepu"]);
            temp="<plik nazwa=\"" +res[i]["sciezka"];
            temp+="\" data=\"" +res[i]["dataDodania"];
            temp+="\" rozmiar=\"" +res[i]["wielkosc"];
            temp+="\" dostep=\"" +res[i]["prawaDostepu"];
            temp+="\r\n";

```



```

        odp.append(temp);
    }
    info("Lista utworzona");
    Odpowiedz(406,100,odp);
}

std::vector <std::string>
parser::pobieranie_listy_plikow(xmlpp::TextReader &reader)
{
    info("Odbieranie listy plikow");
    std::vector <std::string> nazwy_plikow;
    while (reader.read())
    {
        if (reader.get_name().compare("plik")==0)//czy mamy
plik?
        {
            info("mamy <plik ");
            if (reader.has_attributes())//Czy mamy atrybuty
            {
                info("mamy atrybuty");
                if
(reader.move_to_attribute("nazwa"))//PrzejdĹś do atrybutu
"nazwa"
                {
                    info("mamy atrybut \"nazwa\");
nazwy_plikow.push_back(reader.get_value());
                }
                else//Brak atrybutu "nazwa"
                {
                    Eline("Brak Atrybutu 'nazwa'");
                    //Odpowiedz(400);
                    //return;
                }
            }
            else//Brak atrybutĂłw
            {
                Eline("Brak AtrybutĂłw");
                //Odpowiedz(400);
                //return;
            }
        }
        else//nieoczekiwane dane
        {
            Eline2("Nieoczekiwane dane:
",reader.get_name().c_str());
            //Odpowiedz(400);
            //return;
        }
    }
}

```

```

    }
}
info("pobieranie/parsowanie listy plików zakończone");
return nazwy_plikow;
}

/*void wyslij_plik(xmlpp::TextReader &reader, std::string plik)
{

}*/
void parser::odbieranie_plikow(xmlpp::TextReader &reader,
std::string uzytkownik)
{
    info("Odbieranie plikow");
    reader.read();
    reader.read();
    info2("plik", reader.get_name().c_str());
    if (reader.get_name().compare("plik")==0)
    {
        info("plik jest ok");
        if (reader.has_attributes())
        {
            info("ma atrybuty");
            if (reader.move_to_attribute("nazwa"))
            {
                std::string nazwa = reader.get_value();
                info2("jest nazwa", nazwa.c_str());
                if (reader.move_to_attribute("rozmiar"))
                {
                    int
rozmiar=atoi(reader.get_value().c_str());
/*mysqlpp::StoreQueryResult
res=baza.getFileInfo(reader.get_value(), login);
if(res.num_rows()<1)
{*/

                std::string odp="<?xml version=\"1.0\"?>\n
<serwer operacja=\"102\"
odp=\"404\"/>\n";
                wyslij(odp);
                std::string sciezka=login+"/";
                sciezka+=nazwa;
                FILE *plik=fopen(sciezka.c_str(), "w+");
                if (!plik)
                {
                    Eline("PLIK SIE NIE OTWORZYŁ!!!!");
                    return;
                }
                std::string temp="";
                info("plik otwarty do zapisu");
                int size=0;

```

```

int size_old=0;
/*do
{
    size_old=size;
    info("zapis fragmentu");
    std::getline(stream,temp);
    fprintf(plik,"%s",temp.c_str());
    size+=temp.size();
    info2("odebrano",temp.c_str());
    if(size_old==size)
        break;
}while (size<rozmiar);*/
info("zapis pliku");
//char *a=new char(4);
int ile_czytac=32;
char a[33]={0};
char t[256];
for (int i=0;i<rozmiar;i+=32)
{
    if(rozmiar-size<32)
        ile_czytac=rozmiar-size;
    stream.read(a,ile_czytac);
    a[ile_czytac]=0;
    info2("odebrano",a);
    fprintf(plik,"%s",a);
    size+=ile_czytac;
    sprintf(t,"Odczytano %d bajtów z %d
bajtów, pozostało do odczytania %d bajtów\nteraz będzie czytał
%d bajtów",size,rozmiar,rozmiar-size,ile_czytac);
    info(t);
}
//char *temp2=new char(rozmiar);
//stream.read(temp2,rozmiar);
//info2("odebrano",temp2);
//fprintf(plik,"%s",temp2);
fclose(plik);
info("plik odebrany");
char tmp[128];
sprintf(tmp,"Pobrano %d z
%d",size,rozmiar);
info(tmp);
//na razie nie sprawdzam hasha itd
//std::getline(stream,temp);
//std::getline(stream,temp);
//std::getline(stream,temp);
info("Odebrano końcowego xmla");
baza.addFile(nazwa,uzytkownik,rozmiar,-1,-
1,-1);
//}
}
else

```

```

        {
            info("brak rozmiaru pliku");
            Odpowiedz(400,102);
        }
    }
}
else
{
    Eline("brak atrybutów");
    Odpowiedz(400,102);
}

}
else
{
    Eline("niepoprawnie wyslany xml");
    Odpowiedz(400,102);
    return;
}

}

void parser::wysylanie_plikow(xmlpp::TextReader &reader,
std::string uzytkownik)
{
    info("Wysylanie plikow");
    std::vector <std::string>
pliki=pobieranie_listy_plikow(reader);
    std::vector <std::string>::iterator it;
    for (it=pliki.begin();it<pliki.end();it++)
    {
        wyslij_plik(*it,uzytkownik);
    }
    info("Wyslano pliki");
}

void parser::wyslij_plik(std::string plik,std::string
uzytkownik)
{
    info("Wyslij plik");
    if (uzytkownik.compare(".")==0)
        uzytkownik=login;
    mysqlpp::StoreQueryResult
res=baza.getFileInfo(plik,uzytkownik);
    if (res.num_rows()<1)
    {
        info("res.num_rows()<1");
        std::string odp="<plik nazwa=\"" +plik;
        odp+="\" data=\"" -1\" rozmiar=\"" -1\" dostep=\"" -
1\" />\r\n";
        info("res.num_rows()<1");
        Odpowiedz(101,402,odp);
    }
}

```

```

        return;
    }
    else
    {
        info("wysylanie info o pliku")
        std::string odp="<plik nazwa=\""+res[0]["sciezka"];
        odp+="\" data=\""+res[0]["dataDodania"];
        odp+="\" rozmiar=\""+res[0]["dataDodania"];
        odp+="\" dostep=\""+res[0]["prawaDostepu"];
        odp+="\"/>\r\n";
        Odpowiedz(101,406,odp);
        info("Info o pliku wyslane");
    }
    info("Czekamy na zgode klienta na wysyl pliku");
    std::string od_klienta,temp;
    do
    {
        info("OCZEKIWANIE NA PUSTÄ„ LINIE");
        std::getline(stream,temp);
        //std::cout<<(int)temp[0]<<"
" <<(int)temp[1]<<std::endl;
        od_klienta.append(temp+"\r\n");
    }
    while (!(temp[0]==13 && temp[1]==0));
    info("sprawdzamy co wyslal klient");
    xmlpp::TextReader reader_temp((unsigned
char*)od_klienta.c_str(),od_klienta.size());
    reader_temp.read();
    if
(reader_temp.get_name().compare("klient")==0)//Sprawdzamy czy
odpowiedź nazleŹy do klienta
    {
        info("wyslal to napewno klient");
        if (reader_temp.has_attributes())//Sprawdzamy czy
odpowiedz ma atrybuty
        {
            info("SÄ... atrybuty");
            if
(reader_temp.move_to_attribute("action"))//Przechodzimy do
atrybutu "action"
            {
                info("mamy atrybut action");
                if
(reader_temp.get_value().compare("ok")==0)//JeŹeli odpowiedz
jest ok to wysylamy plik
                {
                    info("action=ok");
                    std::string sciezka=login+"/";
                    sciezka+=plik;
                    info("Otwieram plik");

```

```

FILE
*plik=std::fopen(sciezka.c_str(),"r");
    if (plik==NULL)
    {
        Eline2("NIE UDALO SIE OTWORZYC PLIKU:
",sciezka.c_str());
        Odpowiedz(402);
        return;
    }
    info("no to wysył,amy");
    char temp[BUFSIZE2];
    while ( (fread ( &temp, 1,BUFSIZE2,plik )
)> 0 )
    {
        info("Trwa wysył,anie pliku, proszÄ™
czekaÄ™...");
        stream<<temp;
    }
    /*std::fstream plik(plik,"r");
    if(plik.is_iopen())
    {*/
    stream<<std::endl;
    info("Plik zostaÅł, wysł,any");
    std::string odp="<?xml version=\"1.0\"?>\
<serwer operacja=\"101\" odp=\"kod_odp\"
hash=\"-1\" />";
    wyslij(odp);

    }
    else//Klient nie zgadza sie na przeslanie
    {
        Eline("Compare != OK");
        return;
    }
}
else//Brak atrybutu "action"
{
    Eline("Brak atrybutu \"action\"");
    Odpowiedz(400);
    return;
}

}
else//Brak atrybutu
{
    Eline("Brak atrybutu");
    Odpowiedz(400);
    return;
}
}
else//Odpowiedź nie od klienta

```

```

        {
            Eline("Odpowiedz nie od klienta");
            Odpowiedz(400);
            return;
        }
    }
}

void parser::usun_pliki(xmlpp::TextReader &reader, std::string
uzytkownik)
{
    info("NOT IMPLEMENTED YET!!!");
}

void eat_zombie()
{
    while (true)
    {
        if (wait(NULL)==-1)
        {
            sleep(1);
        }
    }
    //waitpid ( -1, NULL,WNOHANG);
}

```

Parser.hpp

```

#ifndef PARSER_HPP
#define PARSER_HPP

#include <boost/asio.hpp>

#include <string>

#include <fstream>

#include <libxml++/libxml++.h>
#include <libxml++/parsers/textreader.h>
#ifdef HAVE_CONFIG_H
#include <config.h>
#endif

#include "baza.hpp"
#include "debug.hpp"

#define BUFSIZE 1024
#define BUFSIZE2 1024*2
using boost::asio::ip::tcp;
class parser
{

```

```

private:
    tcp::iostream & stream;
    std::string login;
    std::string haslo;
    //std::string uzytkownik;
    int id_sesji;
    char bufor[BUFSIZE];
    Baza baza;

    void parsuj(std::string do_parsowania);
    bool logowanie(std::string login, std::string haslo);
    void odpowiedz_login(int i);
    void Odpowiedz(int i, int numer_operacji=-1);
    void Odpowiedz(int i,int nr_operacji,std::string odp);

    void wyslij(std::string w);
    void lista_plikow(std::string uzytkownik);
    void odbieranie_plikow(xmlpp::TextReader &reader,
std::string uzytkownik);
    void wysylanie_plikow(xmlpp::TextReader &reader,
std::string uzytkownik);
    void usun_pliki(xmlpp::TextReader &reader,std::string
uzytkownik);
    void wyslij_plik(std::string plik,std::string uzytkownik);
    std::vector <std::string>
pobieranie_listy_plikow(xmlpp::TextReader &reader);
public:
    parser(tcp::iostream &stream, const char* server, const
char* user, const char *pass,const char *db):stream(stream)
    {
        info("Parser Konstruktor PoczÄ...tek");
        login.clear();
        haslo.clear();
        id_sesji=0;
        baza.connect(server,user,pass,db);
        info("Parser Konstruktor Koniec");
    };
    void start();

};

//void odbieracz (tcp::iostream &stream);
void eat_zombie();
#endif//PARSER_HPP

```

Version.h

```

#ifndef VERSION_H
#define VERSION_H

namespace AutoVersion{

```



```

//Date Version Types
static const char DATE[] = "08";
static const char MONTH[] = "05";
static const char YEAR[] = "2009";
static const double UBUNTU_VERSION_STYLE = 9.05;

//Software Status
static const char STATUS[] = "Beta";
static const char STATUS_SHORT[] = "b";

//Standard Version Type
static const long MAJOR = 0;
static const long MINOR = 3;
static const long BUILD = 16;
static const long REVISION = 112;

//Miscellaneous Version Types
static const long BUILDS_COUNT = 280;
#define RC_FILEVERSION 0,3,16,112
#define RC_FILEVERSION_STRING "0, 3, 16, 112\0"
static const char FULLVERSION_STRING[] = "0.3.16.112";

//These values are to keep track of your versioning state,
don't modify them.
static const long BUILD_HISTORY = 0;

}
#endif //VERSION_H

```

Xml.hpp

```

//#define XML_STREAM_ERROR "<?xml
version='1.0'?><xml_stream_error/>"

```

### **Kod źródłowy bazy danych:**

```

CREATE DATABASE dev;
USE dev;
CREATE TABLE accounts_konto (
    id integer NOT NULL PRIMARY KEY,
    user_id integer NOT NULL REFERENCES auth_user (id),
    miasto varchar(50) NOT NULL,
    zainteresowania text NOT NULL,

```

```

        plec varchar(1) NOT NULL
    );
CREATE TABLE auth_group (
    id integer NOT NULL PRIMARY KEY,
    name varchar(80) NOT NULL UNIQUE
);
CREATE TABLE auth_group_permissions (
    id integer NOT NULL PRIMARY KEY,
    group_id integer NOT NULL REFERENCES auth_group (id),
    permission_id integer NOT NULL REFERENCES auth_permission
(id),
    UNIQUE (group_id, permission_id)
);
CREATE TABLE auth_message (
    id integer NOT NULL PRIMARY KEY,
    user_id integer NOT NULL REFERENCES auth_user (id),
    message text NOT NULL
);
CREATE TABLE auth_permission (
    id integer NOT NULL PRIMARY KEY,
    name varchar(50) NOT NULL,
    content_type_id integer NOT NULL,
    codename varchar(100) NOT NULL,
    UNIQUE (content_type_id, codename)
);
CREATE TABLE auth_user (
    id integer NOT NULL PRIMARY KEY,
    username varchar(30) NOT NULL UNIQUE,
    first_name varchar(30) NOT NULL,
    last_name varchar(30) NOT NULL,
    email varchar(75) NOT NULL,
    password varchar(128) NOT NULL,
    is_staff bool NOT NULL,
    is_active bool NOT NULL,
    is_superuser bool NOT NULL,
    last_login datetime NOT NULL,
    date_joined datetime NOT NULL
);
CREATE TABLE auth_user_groups (
    id integer NOT NULL PRIMARY KEY,
    user_id integer NOT NULL REFERENCES auth_user (id),
    group_id integer NOT NULL REFERENCES auth_group (id),
    UNIQUE (user_id, group_id)
);
CREATE TABLE auth_user_user_permissions (
    id integer NOT NULL PRIMARY KEY,
    user_id integer NOT NULL REFERENCES auth_user (id),
    permission_id integer NOT NULL REFERENCES auth_permission
(id),
    UNIQUE (user_id, permission_id)
);

```

```

CREATE TABLE django_admin_log (
    id integer NOT NULL PRIMARY KEY,
    action_time datetime NOT NULL,
    user_id integer NOT NULL REFERENCES auth_user (id),
    content_type_id integer NOT NULL REFERENCES
django_content_type (id),
    object_id text NULL,
    object_repr varchar(200) NOT NULL,
    action_flag smallint unsigned NOT NULL,
    change_message text NOT NULL
);
CREATE TABLE django_content_type (
    id integer NOT NULL PRIMARY KEY,
    name varchar(100) NOT NULL,
    app_label varchar(100) NOT NULL,
    model varchar(100) NOT NULL,
    UNIQUE (app_label, model)
);
CREATE TABLE django_session (
    session_key varchar(40) NOT NULL PRIMARY KEY,
    session_data text NOT NULL,
    expire_date datetime NOT NULL
);
CREATE TABLE django_site (
    id integer NOT NULL PRIMARY KEY,
    domain varchar(100) NOT NULL,
    name varchar(50) NOT NULL
);
CREATE TABLE files_plik (
    id integer NOT NULL PRIMARY KEY,
    konto_id integer NOT NULL REFERENCES accounts_konto (id),
    sciezka varchar(255) NOT NULL,
    dataDodania datetime NOT NULL,
    prawaDostepu integer NOT NULL,
    wielkosc integer NOT NULL,
    hashValue varchar(50) NOT NULL
);
CREATE TABLE friends_userlink (
    id integer NOT NULL PRIMARY KEY,
    from_user_id integer NOT NULL REFERENCES accounts_konto
(id),
    to_user_id integer NOT NULL REFERENCES accounts_konto
(id),
    date_added datetime NOT NULL,
    UNIQUE (to_user_id, from_user_id)
);
CREATE INDEX accounts_konto_user_id ON accounts_konto
(user_id);
CREATE INDEX auth_message_user_id ON auth_message (user_id);
CREATE INDEX auth_permission_content_type_id ON
auth_permission (content_type_id);

```

```

CREATE INDEX django_admin_log_content_type_id ON
django_admin_log (content_type_id);
CREATE INDEX django_admin_log_user_id ON django_admin_log
(user_id);
CREATE INDEX files_plik_konto_id ON files_plik (konto_id);
CREATE INDEX friends_userlink_from_user_id ON friends_userlink
(from_user_id);
CREATE INDEX friends_userlink_to_user_id ON friends_userlink
(to_user_id);

```

### **Kod źródłowy portalu:**

```

###WARSTWA LOGIKI
##KOD ZWIĄZANY Z KONTAMI UŻYTKOWNIKÓW:
#admin.py - kod odpowiedzialny za wygląd panelu
administracyjnego
from django.contrib import admin
from accounts.models import Konto

admin.site.register(Konto)

#forms.py - opis formularzy
from django import forms

plec_choice = (('K', 'Kobieta'), ('M', 'Mężczyzna'))

class LoginForm(forms.Form):
    login = forms.CharField(label="Login")
    password = forms.CharField(label="Hasło",
widget=forms.PasswordInput)

class RegisterForm(forms.Form):
    login = forms.CharField(label="Login")
    haslo = forms.CharField(label="Hasło",
widget=forms.PasswordInput)
    re_haslo = forms.CharField(label="Powtorz hasło",
widget=forms.PasswordInput)
    e_mail = forms.EmailField(label="e-mail")
    plec = forms.ChoiceField(label="Plec",
choices=plec_choice)

class SearchForm(forms.Form):
    search = forms.CharField(label="Szukaj")

class EditForm(forms.Form):
    imie = forms.CharField(label="Imie", required=False)
    nazwisko = forms.CharField(label="Nazwisko",
required=False)
    miasto = forms.CharField(label="Miasto", required=False)

```

```

    zainteresowania = forms.CharField(label="Zainteresowania",
widget=forms.Textarea, required=False)
    e_mail = forms.EmailField(label="e-mail", required=False)
    stare_haslo = forms.CharField(label="Stare haslo",
widget=forms.PasswordInput, required=False)
    nowe_haslo = forms.CharField(label="Nowe haslo",
widget=forms.PasswordInput, required=False)
    re_nowe_haslo = forms.CharField(label="Powtorz haslo",
widget=forms.PasswordInput, required=False)

```

```

#models.py - opis modelu danych wykorzystywanego w bazie
from django.db import models
from django.contrib.auth.models import User

```

```

class Konto(models.Model):
    user = models.ForeignKey(User)
    miasto = models.CharField(max_length=50)
    zainteresowania = models.TextField()
    plec = models.CharField(max_length=1)

    def __unicode__(self):
        return self.user.username

    def save(self, *args, **kwargs):
        self.user.save(*args, **kwargs)
        super(Konto,self).save(**kwargs)

    class Meta:
        verbose_name_plural = "Konta"

```

```

#urls.py - opis składni adresów URL i związanych z nimi
funkcjami
from django.conf.urls.defaults import *

```

```

urlpatterns = patterns('accounts.views',
    url(r'details/(?P<username>[\w-]+)/$', 'profile_view'),
    url(r'edit/(?P<username>[\w-]+)/$', 'profile_edit'),
    url(r'save/(?P<username>[\w-]+)/$', 'profile_save'),
    url(r'delete/(?P<username>[\w-]+)/$', 'profile_delete'),
    url(r'register/$$', 'register'),
    url(r'login/$$', 'user_login'),
    url(r'logout/$$', 'user_logout'),
    url(r'latest/$$', 'latest_users'),
    url(r'search/$$', 'search'),
)

```

```

#views.py - logika systemu, funkcje wywoływane w momencie
wpisanie odpowiedniego adresu URL
from django.shortcuts import render_to_response,
get_object_or_404

```

```

from django.contrib.auth.decorators import login_required
from django.http import Http404, HttpResponseRedirect
from django.contrib.auth.models import User
from django.contrib.auth import login, logout, authenticate
from django.template import RequestContext
from accounts.models import Konto
from accounts.forms import *
from friends.models import UserLink

MESSAGE_CODES = ('Warning', 'Information', 'Error')

class Message(object):
    def __init__(self, type, content):
        self.type = MESSAGE_CODES[type]
        self.content = content

    def __unicode__(self):
        return "%s : %s" % self.type, self.content

def user_login(request):
    msg=None
    try:
        requestUser = User.objects.get(username =
request.user.username)
        requestKonto = Konto.objects.get(user = requestUser)
    except User.DoesNotExist:
        requestKonto = None
    except Konto.DoesNotExist:
        requestKonto = None
    if request.POST:
        username = request.POST['login']
        password = request.POST['password']
        user = authenticate(username=username,
password=password)
        if user:
            try:
                konto = Konto.objects.get(user = user)
            except Konto.DoesNotExist:
                konto = Konto(user = user, plec ="M")
                konto.save()
            login(request, user)
            return
    HttpResponseRedirect(request.GET.get("next") or
"/accounts/details/"+username+"/")
    msg = Message(2,"Zly login lub haslo")
    f = LoginForm()
    return render_to_response("accounts/logowanie.html", {
        "form":f,
        "msg": msg,
        "requestKonto":requestKonto,
    })

```

```

def user_logout(request):
    if request.user.is_authenticated:
        logout(request)
    return HttpResponseRedirect("/")

def index(request):
    try:
        requestUser = User.objects.get(username =
request.user.username)
        requestKonto = Konto.objects.get(user = requestUser)
    except User.DoesNotExist:
        requestKonto = None
    except Konto.DoesNotExist:
        requestKonto = None
    return
render_to_response("index.html", {"requestKonto":requestKonto})

def register(request):
    try:
        requestUser = User.objects.get(username =
request.user.username)
        requestKonto = Konto.objects.get(user = requestUser)
    except User.DoesNotExist:
        requestKonto = None
    except Konto.DoesNotExist:
        requestKonto = None
    if request.method == "POST":
        f = RegisterForm(request.POST)
        if not f.is_valid():
            msg = Message(2, "Wypelnij wszystkie pola!")
            return
render_to_response("accounts/register.html", {"form":f,
"msg":msg})
        else:
            if not request.POST["haslo"] ==
request.POST["re_haslo"]:
                msg = Message(2, "Hasla musza byc takie same!")
                return
render_to_response("accounts/register.html", {"form":f,
"msg":msg})
            try:
                user =
User.objects.get(username=request.POST["login"])
                msg = Message(2, "Uzytkownik o podanym loginie
juz istnieje")
                return
render_to_response("accounts/register.html", {"form":f,
"msg":msg})
            except User.DoesNotExist:

```

```

        user = User(username = request.POST["login"],
                    password = request.POST["haslo"],
                    email=request.POST["e_mail"],
                    )
        user.set_password(request.POST["haslo"])
        user.save()
        konto = Konto(user = user, plec =
request.POST["plec"])
        konto.save()

login(request,authenticate(username=request.POST["login"],
password=request.POST["haslo"]))
        msg = Message(1,"Rejestracja zakonczona
sukcesem")

        return
render_to_response("accounts/detail.html",{ "msg":msg,
"requestKonto":konto, "viewKonto":konto})
        f = RegisterForm()
        return render_to_response("accounts/register.html",
{"form":f})

@login_required
def profile_view(request, username):
    requestKonto = Konto.objects.get(user = request.user)
    viewUser = get_object_or_404(User, username=username)
    viewKonto = Konto.objects.get(user=viewUser)
    add = True
    if requestKonto.user.username != viewKonto.user.username:
        try:
            ul = UserLink.objects.get(from_user=requestKonto,
                                     to_user = viewKonto)
            add = False
        except UserLink.DoesNotExist:
            add = True
    context = {
        'requestKonto':requestKonto,
        'viewKonto': viewKonto,
        'add':add,
    }
    return render_to_response("accounts/detail.html", context)

def latest_users(request):
    try:
        requestUser = User.objects.get(username =
request.user.username)
        requestKonto = Konto.objects.get(user = requestUser)
    except User.DoesNotExist:
        requestKonto = None
    except Konto.DoesNotExist:
        requestKonto = None

```



```

        latestUsers = []
        tempUsers = User.objects.all().order_by('-
date_joined')[:25]
        try:
            for u in tempUsers:
                k = Konto.objects.get(user=u)
                latestUsers.append(k)
        except Konto.DoesNotExist:
            pass
        return render_to_response("accounts/latestUsers.html",

{"requestKonto":requestKonto,"latestUsers":latestUsers})

@login_required
def search(request):
    requestKonto = Konto.objects.get(user=request.user)
    if request.POST:
        f = SearchForm(request.POST)
        if f.is_valid:
            accounts=[]
            userList = User.objects.filter(username__contains
=
                request.POST["search"])
            for user in userList:
                k = Konto.objects.get(user=user)
                accounts.append(k)
            count =len(accounts)
            if count == 1:
                msg = Message(1, "Znalezione 1 uzytkownika")
            else:
                msg = Message(1,"Znalezione "+repr(count)+"
uzytkownikow")
            return render_to_response("accounts/search.html",{
                "form":f,"requestKonto":requestKonto,
"users":accounts, "msg":msg})
            f=SearchForm()
            return
    render_to_response("accounts/search.html",{ "form":f,
'requestKonto':requestKonto})

@login_required
def profile_edit(request, username):
    requestKonto = Konto.objects.get(user=request.user)
    if requestKonto.user.username != username:
        msg = Message(2,"Mozesz zmieniac tylko swoj profil!")
        konto = Konto.objects.get(user=requestKonto)
        return render_to_response("accounts/detail.html",
{"requestKonto":requestKonto,"viewKonto":requestKonto,
"msg":msg})
    f = EditForm()
    konto = Konto.objects.get(user=request.user)

```

```

        return
    render_to_response("accounts/edit.html", {"form": f, "requestKonto": requestKonto,
        "editKonto": requestKonto})

@login_required
def profile_save(request, username):
    requestKonto = Konto.objects.get(user=request.user)
    if requestKonto.user.username != username:
        msg = Message(2, "Mozesz zapisywac tylko swoje konto!")
        return render_to_response("accounts/edit.html",
            {"requestKonto": requestKonto, "editKonto": requestKonto,
            "msg": msg})
    if request.POST:
        f = EditForm(request.POST)
        requestKonto.user.first_name = request.POST["imie"]
        requestKonto.user.last_name = request.POST["nazwisko"]
        requestKonto.miasto = request.POST["miasto"]
        requestKonto.zainteresowania =
request.POST["zainteresowania"]
        requestKonto.user.email = request.POST["e_mail"]
        if request.POST["nowe_haslo"]:
            if request.POST["stare_haslo"] !=
requestKonto.user.password:
                msg = Message(2, "Podales zle stare haslo")
                return
            render_to_response("accounts/edit.html",

{"requestKonto": requestKonto, "editKonto": requestKonto,
            "msg": msg})
            if
request.POST["nowe_haslo"] != request.POST["re_nowe_haslo"]:
                msg = Message(2, "Nowe haslo i powtorzone nowe
haslo musza byc
                takie same")
                return
            render_to_response("accounts/edit.html",

{"requestKonto": requestKonto, "editKonto": requestKonto,
            "msg": msg})

        requestKonto.user.password = request.POST["nowe_haslo"]
        requestKonto.save()
        msg = Message(1, "Zmiany zostaly pomyslnie zapisane")
        return render_to_response("accounts/edit.html",
            {"form": f,
            "requestKonto": requestKonto, "editKonto": requestKonto,
            "msg": msg})

@login_required
def profile_delete(request, username):

```

```

        requestKonto = Konto.objects.get(user=request.user)
        if requestKonto.user.username != username:
            msg = Message(2, "Mozesz usunac tylko swoje konto!")
            return render_to_response("accounts/detail.html",

{"requestKonto":requestKonto, "viewKonto":requestKonto,
"msg":msg})
        logout(request)
        requestKonto.user.delete()
        requestKonto.delete()
        msg = Message(1, "Twoje konto zostalo usuniete")
        return render_to_response("index.html", {"msg":msg})

##KOD ZWIĄZANY Z PLIKAMI UŻYTKOWNIKÓW:
#models.py - opis modelu danych wykorzystywanego w bazie
from django.db import models
from accounts.models import Konto

from datetime import datetime

class Plik(models.Model):
    konto = models.ForeignKey(Konto)
    sciezka = models.CharField(max_length=255)
    dataDodania = models.DateTimeField(default =
datetime.now())
    prawaDostepu = models.IntegerField()
    wielkosc = models.IntegerField()
    hashValue = models.CharField(max_length=50)

    def __unicode__(self):
        return self.sciezka+" - user:"+ self.konto

##KOD ZWIĄZANY Z ZARZĄDZANIEM ZNAJOMYMI:
#admin.py - kod odpowiedzialny za wyglad panelu
administracyjnego
from django.contrib import admin
from friends.models import UserLink

admin.site.register(UserLink)

#helpers.py - kod opisujący funkcje pomocnicze
from accounts.models import Konto
from friends.models import UserLink

#znajomosc: ja udostepniam komus
def get_my_followers(user):
    ul = UserLink.objects.filter(from_user =
        user).values('to_user').order_by('-date_added')
    return Konto.objects.filter(id__in=[i['to_user'] for i in
ul])

```

```

#znajomosc: ktos udostepnia mi
def get_my_following(user):
    ul = UserLink.objects.filter(to_user =
                                user).values('from_user').order_by('-date_added')
    return Konto.objects.filter(id__in=[i['from_user'] for i
in ul])

#znajomosc w dwie strony
def get_mutual(user):
    followers = UserLink.objects.filter(from_user =
                                        user).values('to_user').order_by('-date_added')
    following = UserLink.objects.filter(to_user =
                                        user).values('from_user').order_by('-date_added')
    followers_set = set([i['to_user'] for i in followers])
    following_set = set([i['from_user'] for i in following])
    return
Konto.objects.filter(id__in=followers_set.intersection(followi
ng_set))

```

```

#models.py - opis modelu danych wykorzystywanego w bazie
from django.db import models

```

```

from accounts.models import Konto
from datetime import datetime

```

```

class UserLink(models.Model):
    from_user = models.ForeignKey(Konto,
related_name='following_set')
    to_user = models.ForeignKey(Konto,
related_name='follower_set')
    date_added = models.DateTimeField(default=datetime.now)

    def __unicode__(self):
        return "%s udostepnia pliki dla %s" %(self.from_user,
self.to_user)

    def save(self, *args, **kwargs):
        if self.from_user== self.to_user:
            raise ValueError("Nie mozesz dodac siebie do
znajomych")
        super(UserLink, self).save(**kwargs)

    class Meta:
        unique_together = (('to_user', 'from_user'),)

```

```

#urls.py - opis skladni adresow URL i zwiazanych z nimi
funkcjami
from django.conf.urls.defaults import *

```

```

urlpatterns = patterns('friends.views',
    url(r'index/(?P<username>[\w-]+)/$', 'friends_manage'),

```

```

        url(r'followers/(?P<username>[\w-
]+)/$', 'friend_list', {'list_type': 'followers'}),
        url(r'following/(?P<username>[\w-
]+)/$', 'friend_list', {'list_type': 'following'}),
        url(r'mutual/(?P<username>[\w-
]+)/$', 'friend_list', {'list_type': 'mutual'}),
        url(r'add/(?P<username>[\w-]+)/$', 'add_friend'),
        url(r'delete/(?P<username>[\w-]+)/$', 'del_friend'),
    )

```

#views.py - logika systemu, funkcje wywoływane w momencie wpisania odpowiedniego adresu URL

```

from django.shortcuts import render_to_response,
get_object_or_404
from django.http import Http404, HttpResponseRedirect
from django.contrib.auth.models import User
from django.contrib.auth.decorators import login_required
from django.template import RequestContext

```

```

from friends.helpers import *
from friends.models import UserLink
from accounts.models import Konto
from accounts.views import Message

```

```

FRIEND_FUNCTION_MAP = {
    'followers': get_my_followers,
    'following': get_my_following,
    'mutual': get_mutual,
}

```

@login\_required

```

def friends_manage(request, username):
    requestKonto = Konto.objects.get(user = request.user)
    viewUser = get_object_or_404(User, username = username)
    viewKonto = Konto.objects.get(user = viewUser)
    followers = FRIEND_FUNCTION_MAP['followers'](viewKonto)
    following = FRIEND_FUNCTION_MAP['following'](viewKonto)
    mutual = FRIEND_FUNCTION_MAP['mutual'](viewKonto)
    context = {
        'followers': followers[:10],
        'following': following[:10],
        'mutual': mutual[:10],
        'requestKonto': requestKonto,
        'viewKonto': viewKonto
    }
    return render_to_response("friends/index.html", context)

```

@login\_required

```

def friend_list(request, list_type, username):
    requestKonto = Konto.objects.get(user = request.user)
    viewUser = get_object_or_404(User, username = username)

```

```

viewKonto = Konto.objects.get(user=viewUser)
context = {
    'list_type':list_type,
    'friends':FRIEND_FUNCTION_MAP[list_type](viewKonto),
    'requestKonto' : requestKonto,
    'viewKonto' : viewKonto,
}
return render_to_response("friends/list.html", context)

@login_required
def add_friend(request, username):
    requestKonto = Konto.objects.get(user = request.user)
    addUser = get_object_or_404(User, username = username)
    addKonto = Konto.objects.get(user = addUser)
    lista = get_my_followers(requestKonto)
    if addKonto in lista:
        msg = Message(2,"Uzytkownik jest juz dodany do Twoich
znajomych")
        return render_to_response("accounts/detail.html",
{"requestKonto":requestKonto, "viewKonto":addKonto, "msg":msg
})
    else:
        link =
UserLink(from_user=requestKonto,to_user=addKonto)
        link.save()
        msg = Message(1,"Uzytkownik pomyslnie dodany do
znajomych")
        return render_to_response("accounts/detail.html", {
            "requestKonto":requestKonto,
            "viewKonto":addKonto,
            "msg":msg })

@login_required
def del_friend(request, username):
    requestKonto = Konto.objects.get(user = request.user)
    delUser = get_object_or_404(User, username = username)
    delKonto = Konto.objects.get(user = delUser)
    try:
        ul = UserLink.objects.get(from_user=requestKonto,
            to_user=delKonto)
    except UserLink.DoesNotExist:
        msg = Message(1,"Nie mozesz usunac tego uzytkownika.")
        followers =
FRIEND_FUNCTION_MAP['followers'](requestKonto)
        following =
FRIEND_FUNCTION_MAP['following'](requestKonto)
        mutual = FRIEND_FUNCTION_MAP['mutual'](requestKonto)
        context = {
            'msg':msg,
            'followers':followers[:10],
            'following':following[:10],

```

```

        'mutual':mutual[:10],
        'requestKonto':requestKonto,
        'viewKonto':requestKonto
    }
    return render_to_response("friends/index.html",
context)
    ul.delete()
    msg = Message(1,"Uzytkownik pomyslnie usuniety ze
znajomych")
    return
HttpResponseRedirect("/friends/index/"+requestKonto.user.usern
ame+"/")

```

###WARSTWA PREZENTACJI - SZABLONY

##SZABLONY OGÓLNE

#base.html - bazowy szablon który rozszerzają inne szablony

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl"
```

```
lang="pl">
```

```
<head>
```

```

    <meta http-equiv="Content-type" content="text/html;
charset=utf-8" />

```

```
{%block extra_head%} {%endblock%}
```

```

    <title>{%block title%}Alternatywny System
Synchronizacji 8{%endblock%}</title>

```

```

    <link rel="stylesheet" type="text/css"
href="/css/style.css" />

```

```
{%block extra_style%} {%endblock%}
```

```

    <script type="text/javascript"
src="/js/jquery.js"></script>

```

```
<script type="text/javascript">
```

```

    $(document).ready(function(){
        $(".close").click(function(){
            closeMsg();

```

```
        });
```

```
        showMsg();
```

```
    });
```

```
function closeMsg()
```

```
{
```

```
    $("#warning").fadeOut(800);
```

```
    $("#info").fadeOut(800);
```

```
    $("#error").fadeOut(800);
```

```
}
```

```
function showMsg()
```

```
{
```

```
    $("#warning").fadeTo(800,0.7);
```

```
    $("#info").fadeTo(800,0.7);
```

```
    $("#error").fadeTo(800,0.7);
```

```
}
```

```

        </script>
        {%block extra_script%}{%endblock%}
    </head>

    <body>
        {%block extra_body%} {%endblock%}
        <!-- Container -->
        <div class="container">
            <!-- Header -->
            <div class="header">
                <h1>Alternatywny system synchronizacji -
ASS.8</h1>
            </div>
            <!-- END Header -->

            <div class="navbar">
                <ul>
                    {%if requestKonto.user.is_authenticated%}
                    Witaj, {{requestKonto.user.username}}
                    {%endif%}
                    <li><a href="/">Czym jest ASS.8?</a></li>
                    <li><a href="/accounts/latest/">Ostatnio
zarejestrowani</a></li>
                    <li><a
href="/accounts/search/">Szukaj</a></li>
                    {%if requestKonto.user.is_authenticated%}
                    <li><a
href="/accounts/details/{{requestKonto.user.username}}/">Profi
l</a></li>
                    <li><a
href="/accounts/logout/">Wyloguj</a></li>
                    {%else%}
                    <li><a href="/accounts/register/">DoŁ,Ä...cz do
nas</a></li>
                    <li><a
href="/accounts/login/">Zaloguj</a></li>
                    {%endif%}
                </ul>
            </div>

            <!-- Content -->
            <div class="content">
                {%if msg%}
                    {%ifequal msg.type "Warning" %}
                        <div id="warning">{{msg.content}}<br /><a
class="close">zamknij</a></div>
                    {%endifequal%}
                    {%ifequal msg.type "Information"%}
                        <div id="info">{{msg.content}}<br /><a
class="close">zamknij</a></div>

```



```

        {%endifequal%}
        {%ifequal msg.type "Error"%}
            <div id="error">{{msg.content}}<br /><a
class="close">zamknij</a></div>
        {%endifequal%}
    {%endif%}
    {%block main_content%} {%endblock%}
</div>
<!-- END Content -->

    <div class="footer">(c) <a
href="http://www.grupappz.fsl.pl">grupappz</a> 2009 {% block
footer %}{% endblock %}</div>
</div>
<!-- END Container -->
</body>
</html>

```

#index.html - szblon głównej strony portalu

```

{%extends "base.html"%}
{%block title%}{{block.super}} - Strona gŁ,Źłwna{%endblock%}

```

```

{%block main_content%}
    <h1>Czym jest ASS.8?</h1>
    <p>
        ASS.8 czyli Alternatywny System Synchronizacji.8 powstał,
        jako projekt na zajÄ™cia
        Pracownia Projektowania Zespoł,owego w
        <a href="http://www.wit.edu.pl">WyŁŁszej Szkole
        Informatyki Stosowanej i ZarzÄ...dzania</a>
        w Warszawie. GŁ,Źłwnym zadaniem programu jest
        umoŁŁliwienie wspÄłł,dzielenia plikÄłw
        miÄ™dzy uŁŁytkownikami oraz synchronizacja danych za
        poŁ>rednictwem serwera.
        UŁŁytkownik ma moŁŁliwoŁ>Äť wspÄłł,dzielenia danych
        dziÄ™ki <a href="">klientowi</a>
        usŁŁ,ugi, natomiast portal daje moŁŁliwoŁ>Äť <a
        href="/accounts/register/">rejestracji</a> i
        aktywnego uczestnictwa w społ,ecznieŁ>ci tworzonej przez
        uŁŁytkownikÄłw.
        Projekt ASS.8 jest usŁŁ,ugÄ... cał,kowicie darmowÄ... i zawsze
        takÄ... pozostanie.
        Jeł>li masz jułŁ konto <a href="/accounts/login/">zaloguj
        siÄ™</a>.
    </p>

```

```

    <h1>Co oferuje ASS.8?</h1>
    <p>KałŁdy uŁŁytkownik otrzymuje miejsce na swoje pliki.
    To ty decydujesz ktÄłre pliki chcesz udostÄ™pniÄť innym
    zarejestrowanym uŁŁytkownikom,

```

które swoim znajomym, a które mają... być widoczne tylko dla Ciebie.

</p>  
{%endblock%}

##SZBŁOY ZWIĄZANE Z KONTAMI UŻYTKOWNIKÓW  
#detail.html - szblon związany z wyświetlaniem szczegółów profilu

{%extends "base.html"%}  
{%block title%} {{block.super}} - Szczegóły profilu:  
{{viewKonto.user.username}} {%endblock%}

{%block extra\_style%}  
<style type="text/css">  
div.content .attr  
{  
 color:#66ba00;  
 margin:0px;  
 padding-left:10px;  
 padding-right:10px;  
}

div.content .value  
{  
 color:white;  
 margin:0px;  
 padding-left:10px;  
}  
</style>  
{%endblock%}

{%block main\_content%}  
 {%ifequal requestKonto.user.username  
viewKonto.user.username%}  
 <a  
href="/accounts/edit/{{requestKonto.user.username}}/">Modyfikuj profil</a><br />  
 <a  
href="/friends/index/{{requestKonto.user.username}}/">Zarządzaj znajomymi</a><br />  
 <a  
href="/accounts/delete/{{requestKonto.user.username}}/">Usuń profil</a><br />  
 <br />  
 {%else%}  
 {%if add%}  
 <a  
href="/friends/add/{{viewKonto.user.username}}/">Dodaj do znajomych</a><br />  
 <br />  
 {%else%}

```

        <a
href="/friends/delete/{{viewKonto.user.username}}/">Usuń, ze
znajomych</a><br />
        <br />
        {%endif%}
        {%endifequal%}
        <h1>Konto użytkownika {{viewKonto.user.username}}</h1>
        <span class="attr">Login:</span> <span
class="value">{{viewKonto.user.username}}</span><br />
        <span class="attr">Dołączył:</span> <span
class="value">{{viewKonto.user.date_joined.date|date:"d/m/Y"}}<
</span><br />
        <span class="attr">Ostatnie logowanie:</span> <span
class="value">{{viewKonto.user.last_login.date|date:"d/m/Y"}}<
/span><br />
        <span class="attr">Imię:</span> <span
class="value">{{viewKonto.user.first_name}}</span><br />
        <span class="attr">Nazwisko:</span> <span
class="value">{{viewKonto.user.last_name}}</span><br />
        <span class="attr">E-mail:</span> <span
class="value">{{viewKonto.user.email}}</span><br />
        <span class="attr">Zainteresowania:</span> <span
class="value">{{viewKonto.zainteresowania|escape}}</span><br
/>
        {%endblock%}
#edit.html - szblon związany z edycją profilu
{%extends "base.html"%}
{%block title%} {{block.super}} - Edycja profilu {%endblock%}

{%block main_content%}
        <h1>Edycja profilu</h1>
        <form action="/accounts/save/{{editKonto.user.username}}/"
method="POST">
                <h2>Zmiana hasła:</h2>
                <p><label for="id_stare_haslo">Stare hasło:</label>
<input type="password" name="stare_haslo" id="id_stare_haslo"
/></p>
                <p><label for="id_nowe_haslo">Nowe hasło:</label>
<input type="password" name="nowe_haslo" id="id_nowe_haslo"
/></p>
                <p><label for="id_re_nowe_haslo">Powtorz
hasło:</label> <input type="password" name="re_nowe_haslo"
id="id_re_nowe_haslo" /></p>
                <h2>Dodatkowe dane osobowe:</h2>
                <p><label for="id_imie">Imię:</label> <input
type="text" name="imie" id="id_imie"
value="{{editKonto.user.first_name}}"/></p>
                <p><label for="id_nazwisko">Nazwisko:</label> <input
type="text" name="nazwisko" id="id_nazwisko"
value="{{editKonto.user.last_name}}"/></p>

```

```

        <p><label for="id_miasto">Miasto:</label> <input
type="text" name="miasto" id="id_miasto"
value="{{editKonto.miasto}}"/></p>
        <p><label
for="id_zainteresowania">Zainteresowania:</label><br
/><textarea id="id_zainteresowania" rows="10" cols="40"
name="zainteresowania">{{editKonto.zainteresowania}}</textarea
></p>
        <p><label for="id_e_mail">e-mail:</label> <input
type="text" name="e_mail" id="id_e_mail"
value="{{editKonto.user.email}}"/></p>
        <input type="submit" value="Zmien" />
    </form>
{%endblock%}

```

#latestUsers.html - szblon związany z wyświetlaniem oastatnio zarejestrowanych użytkowników

```

{%extends "base.html"%}
{%block title%} {{block.super}} - Ostatnio dodani uzytkownicy
{%endblock%}
{%block extra_style%}
<style type="text/css">
    table.userList
    {
        margin:0px;
        padding-left:20px;
        padding-right:20px;

    }
    table.userList td
    {
        padding-left:20px;
        padding-right:20px;
    }
</style>

```

```

{%endblock%}
{%block main_content%}
    {%if latestUsers%}
        <h1>Ostatnio dodatni uzytkownicy:</h1>
        <table class="userList">
            {% for user in latestUsers %}
                <tr><td><a
href="/accounts/details/{{user.user.username}}/">{{user.user.u
sername}}</td><td>doŁ,Ä...czyŁ,
{{user.user.date_joined.date|date:"d/m/Y"}}</a></td></tr>
                {%endfor%}
            </table>
            <p></p>
            {%else%}
                <h1>Brak ostatnio dodanych uŁŁytkownikÄŁw</h1>

```

```

    {%endif%}

{%endblock%}

#logowanie.html - szablon odpowiedzialny za logowanie
użytkowników
{%extends "base.html"%}
{%block title%} {{block.super}} - Logowanie {%endblock%}
{%block extra_style%}
<style type="text/css">
    div.login
    {
        padding-top:30px;
        width:300px;
        margin-left:auto;
        margin-right:auto;
    }
    div.login_header
    {
        font-size:26px;
        background:#496e40;
        color:#fff;
        margin:0px;
        text-align: center;
    }
    div.login_form
    {
        padding:10px;
        border:1px solid black;
        border-top:none;
    }
</style>
{%endblock%}

{%block main_content%}
    {% if requestKonto.user.is_authenticated%}
        <h1>Jesteś już zalogowany.</h1>
        <a
href="/accounts/details/{{requestKonto.user.username}}/">Strona
a profilu</a>
    {%else%}
        <div class="login">
            <div class="login_header">
                Zaloguj się:
            </div>
            <div class="login_form">
                <form action="" method="POST">
                    {{form.as_p}}
                    <input type="submit" value="Zaloguj" />
                </form>
            </div>
        </div>
    {%endif%}

```

```

                <p>Jesli nie masz konta <a
href="/accounts/register/">zarejestruj sie</a>.</p>
            </div>
        {%endif%}
    {%endblock%}

```

```

#register.html - szablon odpowiedzialny z rejestrację
użytkowników
{%extends "base.html"%}
{%block title%} {{block.super}} - Rejestracja {%endblock%}

```

```

{%block main_content%}
    {%if requestKonto.user.is_authenticated%}
        <h1>Jestes zalogowany! Jesli chcesz zalozyc nowe konto
musisz sie najpierw wylogowac!</h1>
    {%else%}
        <h1>Zarejestruj nowego uzytkownika</h1>
        <h2>Wypełnij wszystkie pola:</h2>
        <form action="" method="POST">
            {{form.as_p}}
            <input type="submit" value="Zarejestruj" />
        </form>
    {%endif%}
{%endblock%}

```

```

#search.html - szablon odpowiedzialny za wyszukiwanie
użytkowników
{%extends "base.html"%}
{%block title%} {{block.super}} - Szukaj
użytkowników{%endblock%}
{%block extra_style%}
<style type="text/css">
    form
    {
        padding-left:20px;
    }
</style>

```

```

{%endblock%}
{%block main_content%}
<h1>Szukaj uzytkownikow</h1>
    <form action="" method="POST">
        {{form.search}}<br />
        <input type="submit" value="Szukaj" />
    </form>
    {%if users%}
        <ul>
            {%for user in users%}

```

```

                <li><a
href="/accounts/details/{{user.user.username}}/">{{user.user.u
sername}}</a></li>
                {%endfor%}
            </ul>
        {%endif%}
    {%endblock%}

```

##SZBLOY ZWIĄZANE Z ZARZĄDZANIEM ZNAJOMYMI

#index..html - szablon odpowiedzialny za wyświetlanie ostatnio dodanych znajomych

```

{%extends "base.html"%}
{%block title%} {{block.super}} - Lista znajomych:
{{viewKonto.user.username}} {%endblock%}

```

```

{%block main_content%}
    <a
href="/friends/followers/{{viewKonto.user.username}}/">Komu
udost&Ampniam pliki?</a><br />
    <a
href="/friends/following/{{viewKonto.user.username}}/">Kto mi
udos&Ampnia pliki?</a><br />
    <a
href="/friends/mutual/{{viewKonto.user.username}}/">Znajomo&Amp;
i w obie strony</a><br />
    <br />
    {%if followers%}
        <h2 class="value">10 ostatnio dodanych
u&Amp;ytownik&Amp;w:</h2>
        <ul>
            {%for u in followers %}
                <li><a
href="/accounts/details/{{u.user.username}}/">{{u.user.usernam
e}}</a></li>
            {%endfor%}
        </ul>
    {%else%}
        <h2 class="value">Brak uzytkownik&Amp;w. <a
href="/accounts/search/">Wyszukaj</a> znajomych kt&Amp;rym chcesz
udostepnia&Amp; swoje pliki</h2>
    {%endif%}

    {%if following%}
        <h2 class="value">10 uzytkownik&Amp;w, kt&Amp;rzy mnie ostatnio
dodali:</h2>
        <ul>
            {%for u in following %}
                <li><a
href="/accounts/details/{{u.user.username}}/">{{u.user.usernam
e}}</a></li>
            {%endfor%}

```

```

        </ul>
        {%else%}
        <h2 class="value">Brak uzytkowników którzy mnie ostatnio
dodali do znajomych.</h2>
        {%endif%}

        {%if mutual%}
        <h2 class="value">10 uzytkowników, z którymi
nawiązałem znajomość w dwie strony:</h2>
        <ul>
            {%for u in mutual %}
                <li><a
href="/accounts/details/{{u.user.username}}/">{{u.user.usernam
e}}</a></li>
            {%endfor%}
        </ul>
        {%else%}
        <h2 class="value">Brak uzytkowników z którymi
nawiązałem znajomość w dwie strony.</h2>
        {%endif%}

{%endblock%}

#list.html - szablon wyswietla wszystkich znajomych
{%extends "base.html"%}
{%block title%} {{block.super}} - Lista znajomych:
{{viewKonto.user.username}} {%endblock%}

{%block main_content%}
    <a
href="/friends/followers/{{viewKonto.user.username}}/">Komu
udostępniam pliki?</a>
    <a
href="/friends/following/{{viewKonto.user.username}}/">Kto mi
udostępnia pliki?</a>
    <a
href="/friends/mutual/{{viewKonto.user.username}}/">Znajomość
i w obie strony</a>

    {%ifequal list_type "followers"%}
        {%if friends %}
            <h1>Użytkownicy którym udostępniam pliki:</h1>
            {%for u in friends %}
                <a
href="/accounts/details/{{u.user.username}}/">{{u.user.usernam
e}}</a><br />
            {%endfor%}
        {%else%}
            <h1>Brak uzytkowników. <a
href="/accounts/search/">Wyszukaj</a> znajomych którym chcesz
udostępniać swoje pliki</h1>

```



```

        {%endif%}
    {%endifequal%}

    {%ifequal list_type "following"%}
        {%if friends %}
            <h1>Uzytkownicy ktorzy dodali mnie do znajomych:</h1>
            {%for u in friends %}
                <a
href="/accounts/details/{{u.user.username}}/">{{u.user.usernam
e}}</a><br />
                {%endfor%}
            {%else%}
                <h1>Brak uzytkownikĂłw.</h1>
            {%endif%}
        {%endifequal%}

        {%ifequal list_type "mutual"%}
            {%if friends %}
                <h1>Uzytkownicy z ktĂłrymi nawiĂ...zaŁ,em znajomoŁ>Ă† w
dwie strony:</h1>
                {%for u in friends %}
                    <a
href="/accounts/details/{{u.user.username}}/">{{u.user.usernam
e}}</a><br />
                    {%endfor%}
                {%else%}
                    <h1>Brak uzytkownikĂłw.</h1>
                {%endif%}
            {%endifequal%}
        {%endblock%}

```

###KOD ARKUSZA STYLÓW

#style.css

body

```

{
    background:#496f40;
    color:#fff;
}

```

a, a:active, a:visited

```

{
    color:#66ba00;
    text-decoration:none;
}

```

a:hover

```

{
    color:#66ba00;
    text-decoration:underline;
}

```

```
a.close, a.close:active, a.close:visited
{
    color:#fff;
    text-decoration:none;
}

a.close:hover
{
    color:#fff;
    text-decoration:underline;
}

#info
{
    text-align:center;
    margin: 0px;
    padding:15px 10px 15px 50px;
    position:relative;
    color: #fff;
    background-color: blue;
}

#warning
{
    text-align:center;
    margin: 0px;
    padding:15px 10px 15px 50px;
    position:relative;
    color: #fff;
    background-color: yellow;
}

#error
{
    text-align:center;
    margin: 0px;
    padding:15px 10px 15px 50px;
    position:relative;
    color: #fff;
    background-color: red;
}

div.container
{
    width:1024px;
    position:relative;
    min-width:760px;
    padding:0px;
    border: 1px solid black;
    background:#314a2c;
```

```

        margin-left:auto;
        margin-right:auto;
    }

div.container a, div.container a:visited, div.container
a:active
{
    text-decoration:none;
}

div.container a:hover
{
    text-decoration:underline;
}

div.header
{
    text-align:right;
    width:1024px;
    height:50px;
    margin: 0px;
    padding: 0px;
    background:#314a2c;
    overflow:hidden;
}

div.header h1
{
    font-size:40px;
    color:white;
    margin: 0px;
    padding: 0px;
}

div.navbar
{
    text-align:right;
    width:1024px;
    height:30px;
    margin:0px;
    padding:0px;
    background: #496e40;
    background-image: url('/gfx/headers.jpg');
    background-repeat: repeat;
}

div.navbar ul
{
    margin:0px;
    padding:0px;
    padding-top:3px;

```

```
}

div.navbar ul li
{
    display: inline;
    list-style-type:none;
    margin-right:10px;
    padding-left:50px;
}

div.navbar ul li a,div.navbar ul li a:visited, div.navbar ul
li a:active
{
    color:#fff;
    padding-left:10px;
    text-decoration:none;
}

div.navbar ul li a:hover
{
    color:#66ba00;
    text-decoration:none;
}

div.content
{
    min-height:200px;
    margin:0px;
    padding:0px;
    background:#314a2c;
    border:none;
}

div.content h1
{
    color:#66ba00;
    margin:0px;
    padding-left:10px;
    padding-right:10px;
}

div.content h2
{
    color:#66ba00;
    margin:0px;
    padding-left:10px;
    padding-right:10px;
}

div.content h3
{

```

```

        color:#66ba00;
        margin:0px;
        margin-bottom:5px;
        font-size:12px;
        padding-left:10px;
    }

div.content p
{
    text-align:justify;
    margin:0px;
    font-size:18px;
    padding:0px;
    padding-left:10px;
    padding-right:10px;
}

div.footer
{
    text-align:center;
    height:30px;
    background-image: url('/gfx/headers.jpg');
    background-repeat:repeat;
    color:#fff;
    padding:0px;
}

div.footer a, div.footer a:visited, div.footer a:active
{
    color:#fff;
    text-decoration:none;
}

div.footer a:hover
{
    color:#fff;
    text-decoration:underline;
}

```

### 3. Literatura i źródła.

„Szkola programowania Język C++” Stephen Prata – Książka traktująca o programowaniu w C++.

„Visual C# 2008. Projektowanie aplikacji. Pierwsze starcie” Jacek Matulewski. –Książka traktująca o programowaniu w C#.

[http://www.tibik.nazwa.pl/index.php?option=com\\_frontpage&Itemid=1](http://www.tibik.nazwa.pl/index.php?option=com_frontpage&Itemid=1) – Strona o programowaniu w C#.

[http://pl.docs.pld-linux.org/uslugi\\_bazydanych\\_mysql.html](http://pl.docs.pld-linux.org/uslugi_bazydanych_mysql.html) - strona o MYSQL  
<http://webmade.org/kursy-online/kurs-mysql.php> - strona o MYSQL  
<http://www.python.rk.edu.pl/> - Biblioteka Pytona -Dokumentacja i przewodniki w języku polskim.  
<http://www.djangoproject.com/> - Oficjalna strona Django.  
<http://www.djangobook.com/> - - Darmowa książka o Django.  
<http://www.django.pl/> - - Polska Społeczność Django.  
<http://pl.wikipedia.org/wiki/Django> - Strona Wikipedi poświęcona Django.

## 4. Załączniki

### 4.1. Przebieg testów programu.

Testy portalu:

Podczas testowania pojawiły się następujące błędy:

- Brak dodawania kont do bazy w momencie rejestracji.
- Możliwość usuwania kont bez odpowiednich praw dostępu.

Błędy zostały naprawione.

Testy klienta:

Podczas testowania pojawiły się następujące błędy:

- Brak możliwości połączenia z serwerem
- Problemy z parsowaniem XML.
- Wiele pomniejszych.

Błędy zostały naprawione.

Testy serwera:

Podczas testowania pojawiły się następujące błędy:

- Problem z dodawaniem plików do bazy danych
- Wiele pomniejszych.

Błędy zostały naprawione.

### 4.2. Changelog Serwera

08 maj 2009

released version 0.3.17 of Ass8-server

Change log:

- Fixed: wysyłanie listy plików
- Fixed: odbieranie plików
- Fixed: dodawanie plików do bazy (ponownie)
- Fixed: wiele pomniejszych

08 maj 2009

released version 0.3.16 of Ass8-server

Change log:

- Applied: Proteza na odbieranie pliku (przez telnet działa)
- Added: Kolejne informacje użyteczne przy odrobaczaniu
- Fixed: Parę małych poprawek

08 maj 2009

released version 0.1.10 of Ass8-server

Change log:

- Fixed: pobieranie listy plików z bazy
- Fixed: pobieranie id z bazy
- Fixed: usuwanie socketu gdy klient się rozłącza
- Fixed: wiele pomniejszych błędów
- Removed: przywitanie serwera

07 maj 2009

released version 0.0.9 of Ass8-server

Change log:

- Added: Wysyłanie pliku na serwer
- Added: Umieszczanie pliku w bazie
- Added: więcej informacji przy debugowaniu

07 maj 2009

released version 0.0.7 of Ass8-server

Change log:

- Updated: Parsowanie
- Added: Wysyłanie plików
- Added: Wysyłanie listy plików
- Added: Logowanie
- Added: Wiele więcej

02 maj 2009

released version 0.0.6 of Ass8-server

Change log:

- Added: odbieranie, kolejkovanie klientów
- Added: fork()
- Added: watek usuwający zombie

02 maj 2009

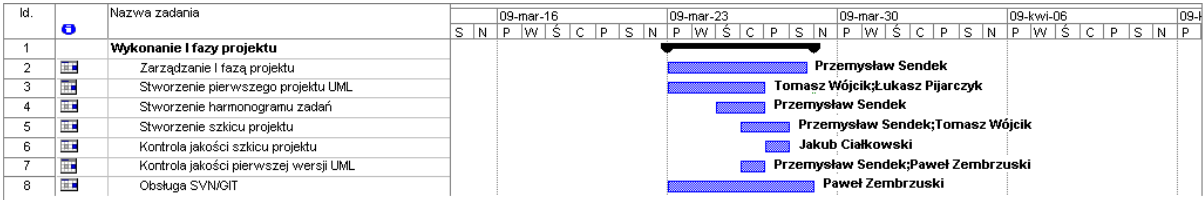
released version 0.0.1 of Ass8-server

Change log:

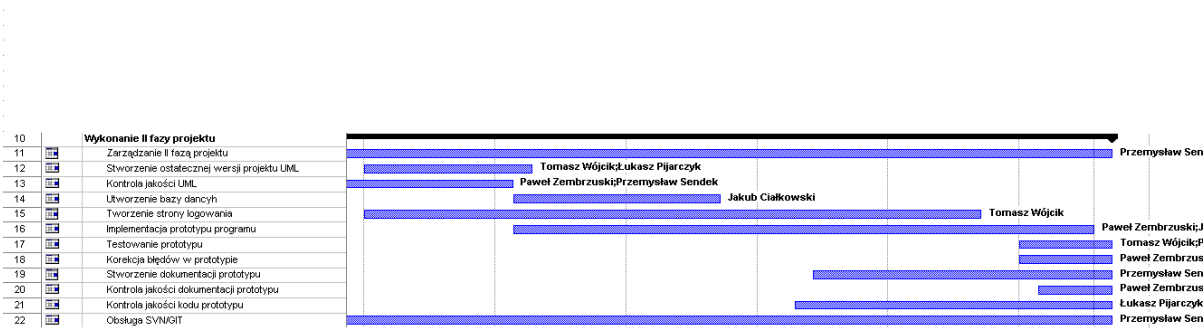
- Added: Zaczątki

### 4.3. Terminarz.

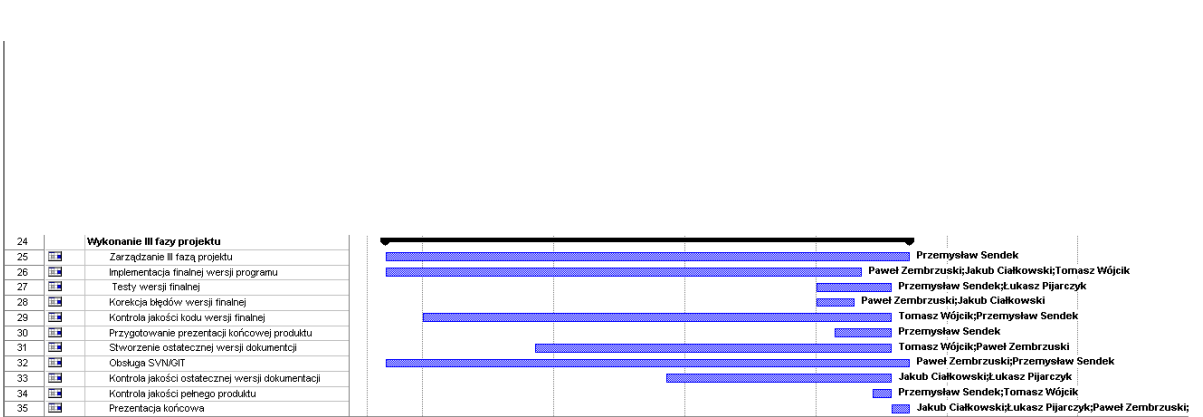
Wykres prezentujący I fazę projektu:



Wykres prezentujący II fazę projektu:



Wykres prezentujący III fazę projektu:



### Zadania

Nazwa zasobu	Nazwa zadania	Cz. trw.	Rozp.	Zak.
	Wykonanie I fazy projektu	6 dn	Pon, 09-03-23	Sob, 09-03-28
Przemysław Sendek	Zarządzanie I fazą projektu	6 dn	Pon, 09-03-23	Sob, 09-03-28



Tomasz Wójcik;Łukasz Pijarczyk	Stworzenie pierwszego projektu UML	4 dn	Pon, 09-03-23	Czw, 09-03-26
Przemysław Sendek	Stworzenie harmonogramu zadań	2 dn	Śro, 09-03-25	Czw, 09-03-26
Przemysław Sendek;Tomasz Wójcik	Stworzenie szkicu projektu	2 dn	Czw, 09-03-26	Pią, 09-03-27
Jakub Ciałkowski	Kontrola jakości szkicu projektu	1 dzień	Pią, 09-03-27	Pią, 09-03-27
Przemysław Sendek;Paweł Zembrzusi	Kontrola jakości pierwszej wersji UML	1 dzień	Czw, 09-03-26	Czw, 09-03-26
Paweł Zembrzusi	Obsługa SVN/GIT	6 dn	Pon, 09-03-23	Sob, 09-03-28
	<b>Wykonanie II fazy projektu</b>	41 dn	Nie, 09-03-29	Pią, 09-05-08
Przemysław Sendek	Zarządzanie II fazą projektu	41 dn	Nie, 09-03-29	Pią, 09-05-08
Tomasz Wójcik;Łukasz Pijarczyk	Stworzenie ostatecznej wersji projektu UML	9 dn	Pon, 09-03-30	Wto, 09-04-07
Paweł Zembrzusi;Przemysław Sendek	Kontrola jakości UML	9 dn	Nie, 09-03-29	Pon, 09-04-06
Jakub Ciałkowski	Utworzenie bazy danych	11 dn	Wto, 09-04-07	Pią, 09-04-17
Tomasz Wójcik	Tworzenie strony logowania	33 dn	Pon, 09-03-30	Pią, 09-05-01
Paweł Zembrzusi;Jakub Ciałkowski	Implementacja prototypu programu	31 dn	Wto, 09-04-07	Czw, 09-05-07
Tomasz Wójcik;Paweł Zembrzusi	Testowanie prototypu	5 dn	Pon, 09-05-04	Pią, 09-05-08
Paweł Zembrzusi	Korekcja błędów w prototypie	5 dn	Pon, 09-05-04	Pią, 09-05-08
Przemysław Sendek;Łukasz Pijarczyk	Stworzenie dokumentacji prototypu	16 dn	Czw, 09-04-23	Pią, 09-05-08
Paweł Zembrzusi;Tomasz Wójcik	Kontrola jakości dokumentacji prototypu	4 dn	Wto, 09-05-05	Pią, 09-05-08
Łukasz Pijarczyk;Przemysław Sendek	Kontrola jakości kodu prototypu	17 dn	Śro, 09-04-22	Pią, 09-05-08
Przemysław Sendek;Paweł Zembrzusi	Obsługa SVN/GIT	41 dn	Nie, 09-03-29	Pią, 09-05-08
	<b>Wykonanie III fazy projektu</b>	28 dn	Sob, 09-05-09	Pią, 09-06-05
Przemysław Sendek	Zarządzanie III fazą projektu	28 dn	Sob, 09-05-09	Pią, 09-06-05

Paweł Zembrzusi;Jakub Ciałkowski;Tomasz Wójcik	Implementacja finalnej wersji programu	25 dn	Sob, 09-05-09	Śro, 09-06-03
Przemysław Sendek;Łukasz Pijarczyk	Testy wersji finalnej	4 dn	Pon, 09-06-01	Czw, 09-06-04
Paweł Zembrzusi;Jakub Ciałkowski	Korekcja błędów wersji finalnej	2 dn	Pon, 09-06-01	Wto, 09-06-02
Tomasz Wójcik;Przemysław Sendek	Kontrola jakości kodu wersji finalnej	25 dn	Pon, 09-05-11	Czw, 09-06-04
Przemysław Sendek	Przygotowanie prezentacji końcowej produktu	3 dn	Wto, 09-06-02	Czw, 09-06-04
Tomasz Wójcik;Paweł Zembrzusi	Stworzenie ostatecznej wersji dokumentacji	19 dn	Nie, 09-05-17	Czw, 09-06-04
Paweł Zembrzusi;Przemysław Sendek	Obsługa SVN/GIT	28 dn	Sob, 09-05-09	Pią, 09-06-05
Jakub Ciałkowski;Łukasz Pijarczyk	Kontrola jakości ostatecznej wersji dokumentacji	12 dn	Nie, 09-05-24	Czw, 09-06-04
Przemysław Sendek;Tomasz Wójcik	Kontrola jakości pełnego produktu	1 dzień	Czw, 09-06-04	Czw, 09-06-04
Jakub Ciałkowski;Łukasz Pijarczyk;Paweł Zembrzusi;Przemysław Sendek;Tomasz Wójcik	Prezentacja końcowa	1 dzień	Pią, 09-06-05	Pią, 09-06-05

Grupa zasobów	Nazwa zasobu
Programista, Bazodanowiec, Kontrola jakości	Jakub Ciałkowski
Drugi projektant, Programista, Kontrola jakości, Tester	Łukasz Pijarczyk
Główny programista, Kontrola jakości, Obsługa SVN	Paweł Zembrzusi
Menadżer, Kontrola Jakości, Obsługa SVN, Tester, Programista, Projektant prezentacji końcowej, Twórca dokumentacji	Przemysław Sendek
Główny Projektant, Programista, Kontrola jakości, Twórca dokumentacji, Tester	Tomasz Wójcik

Zasoby i przydziały	Rozp.	Zak.
Jakub Ciałkowski	Pią, 09-03-27	Pią, 09-06-05
<i>Kontrola jakości szkicu projektu</i>	Pią, 09-03-27	Pią, 09-03-27
<i>Utworzenie bazy dancyh</i>	Wto, 09-04-07	Pią, 09-04-17
<i>Implementacja prototypu programu</i>	Wto, 09-04-07	Czw, 09-05-07
<i>Implementacja finalnej wersji programu</i>	Sob, 09-05-09	Śro, 09-06-03
<i>Korekcja błędów wersji finalnej</i>	Pon, 09-06-01	Wto, 09-06-02

<i>Kontrola jakości ostatecznej wersji dokumentacji</i>	Nie, 09-05-24	Czw, 09-06-04
<i>Prezentacja końcowa</i>	Pią, 09-06-05	Pią, 09-06-05
Łukasz Pijarczyk	Pon, 09-03-23	Pią, 09-06-05
<i>Stworzenie pierwszego projektu UML</i>	Pon, 09-03-23	Czw, 09-03-26
<i>Stworzenie ostatecznej wersji projektu UML</i>	Pon, 09-03-30	Wto, 09-04-07
<i>Stworzenie dokumentacji prototypu</i>	Czw, 09-04-23	Pią, 09-05-08
<i>Kontrola jakości kodu prototypu</i>	Śro, 09-04-22	Pią, 09-05-08
<i>Testy wersji finalnej</i>	Pon, 09-06-01	Czw, 09-06-04
<i>Kontrola jakości ostatecznej wersji dokumentacji</i>	Nie, 09-05-24	Czw, 09-06-04
<i>Prezentacja końcowa</i>	Pią, 09-06-05	Pią, 09-06-05
Paweł Zembrzusi	Pon, 09-03-23	Pią, 09-06-05
<i>Kontrola jakości pierwszej wersji UML</i>	Czw, 09-03-26	Czw, 09-03-26
<i>Obsługa SVN/GIT</i>	Pon, 09-03-23	Sob, 09-03-28
<i>Kontrola jakości UML</i>	Nie, 09-03-29	Pon, 09-04-06
<i>Implementacja prototypu programu</i>	Wto, 09-04-07	Czw, 09-05-07
<i>Testowanie prototypu</i>	Pon, 09-05-04	Pią, 09-05-08
<i>Korekcja błędów w prototypie</i>	Pon, 09-05-04	Pią, 09-05-08
<i>Kontrola jakości dokumentacji prototypu</i>	Wto, 09-05-05	Pią, 09-05-08
<i>Obsługa SVN/GIT</i>	Nie, 09-03-29	Pią, 09-05-08
<i>Implementacja finalnej wersji programu</i>	Sob, 09-05-09	Śro, 09-06-03
<i>Korekcja błędów wersji finalnej</i>	Pon, 09-06-01	Wto, 09-06-02
<i>Stworzenie ostatecznej wersji dokumentacji</i>	Nie, 09-05-17	Czw, 09-06-04
<i>Obsługa SVN/GIT</i>	Sob, 09-05-09	Pią, 09-06-05
<i>Prezentacja końcowa</i>	Pią, 09-06-05	Pią, 09-06-05
Przemysław Sendek	Pon, 09-03-23	Pią, 09-06-05
<i>Zarządzanie I fazą projektu</i>	Pon, 09-03-23	Sob, 09-03-28
<i>Stworzenie harmonogramu zadań</i>	Śro, 09-03-25	Czw, 09-03-26
<i>Stworzenie szkicu projektu</i>	Czw, 09-03-26	Pią, 09-03-27
<i>Kontrola jakości pierwszej wersji UML</i>	Czw, 09-03-26	Czw, 09-03-26
<i>Zarządzanie II fazą projektu</i>	Nie, 09-03-29	Pią, 09-05-08
<i>Kontrola jakości UML</i>	Nie, 09-03-29	Pon, 09-04-06
<i>Stworzenie dokumentacji prototypu</i>	Czw, 09-04-23	Pią, 09-05-08
<i>Kontrola jakości kodu prototypu</i>	Śro, 09-04-22	Pią, 09-05-08
<i>Obsługa SVN/GIT</i>	Nie, 09-03-29	Pią, 09-05-08
<i>Zarządzanie III fazą projektu</i>	Sob, 09-05-09	Pią, 09-06-05
<i>Testy wersji finalnej</i>	Pon, 09-06-01	Czw, 09-06-04
<i>Kontrola jakości kodu wersji finalnej</i>	Pon, 09-05-11	Czw, 09-06-04
<i>Przygotowanie prezentacji końcowej produktu</i>	Wto, 09-06-02	Czw, 09-06-04
<i>Obsługa SVN/GIT</i>	Sob, 09-05-09	Pią, 09-06-05
<i>Kontrola jakości pełnego produktu</i>	Czw, 09-06-04	Czw, 09-06-04

<i>Prezentacja końcowa</i>	Pią, 09-06-05	Pią, 09-06-05
Tomasz Wójcik	Pon, 09-03-23	Pią, 09-06-05
<i>Stworzenie pierwszego projektu UML</i>	Pon, 09-03-23	Czw, 09-03-26
<i>Stworzenie szkicu projektu</i>	Czw, 09-03-26	Pią, 09-03-27
<i>Stworzenie ostatecznej wersji projektu UML</i>	Pon, 09-03-30	Wto, 09-04-07
<i>Tworzenie strony logowania</i>	Pon, 09-03-30	Pią, 09-05-01
<i>Testowanie prototypu</i>	Pon, 09-05-04	Pią, 09-05-08
<i>Kontrola jakości dokumentacji prototypu</i>	Wto, 09-05-05	Pią, 09-05-08
<i>Implementacja finalnej wersji programu</i>	Sob, 09-05-09	Śro, 09-06-03
<i>Kontrola jakości kodu wersji finalnej</i>	Pon, 09-05-11	Czw, 09-06-04
<i>Stworzenie ostatecznej wersji dokumentacji</i>	Nie, 09-05-17	Czw, 09-06-04
<i>Kontrola jakości pełnego produktu</i>	Czw, 09-06-04	Czw, 09-06-04
<i>Prezentacja końcowa</i>	Pią, 09-06-05	Pią, 09-06-05