**WYŻSZA SZKOŁA INFORMATYKI**

**I ZARZĄDZANIA**

**Z SIEDZIBĄ W RZESZOWIE**

**Dokumentacja projektu**

Przedmiot: **Programowanie**

Tytuł projektu:

**Transfer plików**

Język C#

Prowadzący: Wykonawca:

dr. Marek Jaszuk Pavlo Ilynych

w55211

Roman Rukh

W55204

Kierunek: 4IIP-D

Rzeszów 2017

**Spis treści**

1. Wstęp 3
2. Forma File transfer 4
3. Forma managera plików………………...…………………………………………….9
4. Podsumowanie………………………………………………………………………..11
5. Literatura…………………………………………………………………….….….....12
6. Załączniki......................................................................................................................13
7. **Wstęp**

Oprogramowanie było stworzone na podstawie własnego zadania, które jest także opisane poniżej. Zadaniem do realizacji było napisanie programu dla transferu danych(serwer i klient) . Opracowane oprogramowanie składa się z dwóch części :

1. Forma klienta – możliwość połączenia z wybranym localhostem i transfer danych.
2. Forma serwera – możliwość stworzenia serwera z wybranymi parametrami oraz transfer danych.

Informacja o użytych technologiach oraz środowiskach:

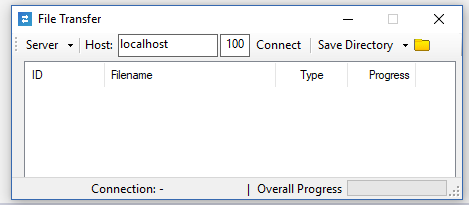
* Środowisko napisania – Microsoft Visual Studio 2017.
* Frameworks – WinForms, Metro.

Główne cele projektu:

1. Nauczyć się korzystać z biblioteki Metro.
2. Polepszyć widzę o tworzeniu klas oraz ich zastosowań.
3. Nauczyć się korzystać z różnych frameworków dla UI.

**2. Forma transferu**

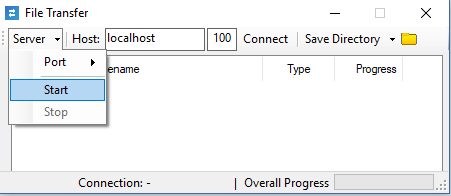
Forma serwera jak również forma klienta jest taka sama przez to, że z każdej formy możemy zrobić formę jaką chcemy. Forma transferu jest prosta, zrozumiała dla każdego użytkownika Internetu, i bardzo łatwa w użyciu.



Rys.1.Okno transferowe

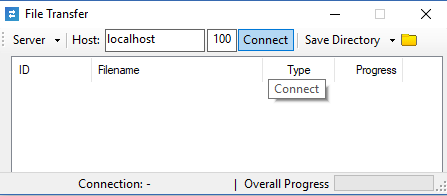
Uruchamianie aplikacji

Żeby uruchomić aplikację musimy otworzyć dwa okna tej samej aplikacji i na jednym zacząć pracę serwera, a na innej pracę klienta. W okienku “Host” wybieramy numer portu albo numer localhostu do którego chcemy się podłączyć i klikamy przycisk “Start”.

****

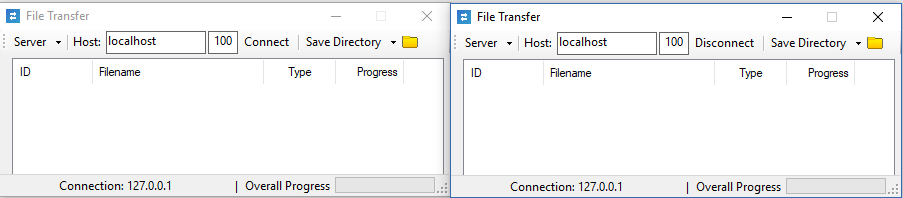
Rys.2. Start serwera

Dla tego, żeby uruchomić formę klienta, tak samo, wybieramy numer portu albo numer localhostu do którego chcemy się podłączyć, potem klikamy przycisk “Connect”.

****

Rys.3. Start klienta

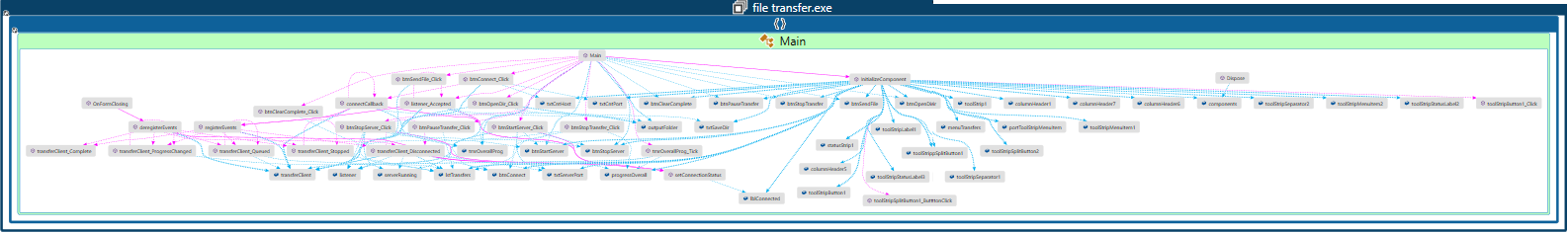
Po prawidłowym wprowadzeniu localhostu albo portu będzie dwa prawie takich samych okien. Okno, które startowało jako serwera nie zmienia się, a w oknie które startowało jako klient zmienia się przycisk “Connect” na “Disconnect”.



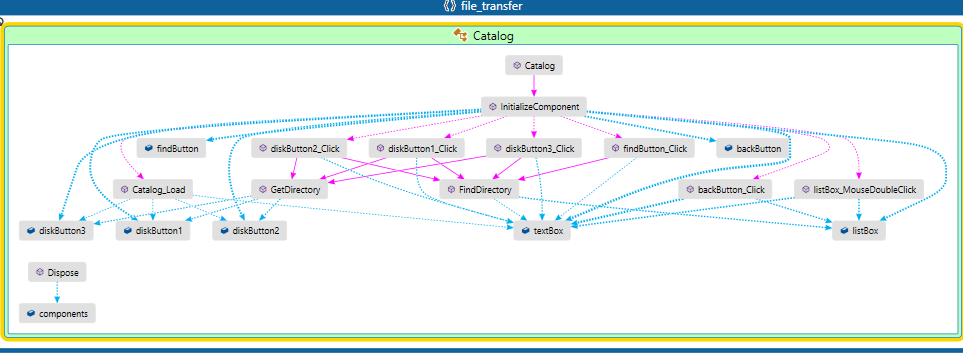
Rys.4. Po prawej stronie – okno serwera, po lewej – okno klienta

W załącznikach dołączam diagram przypadków użycia z wyróżnieniem aktorów projektu, diagram klasów i diagram Ganta.

Projekt składa się z dwóch części: Main (forma transferu danych), Catalog (Manager plików).



Rys.5. Diagram klas “Main”



Rys.6. Diagram klas “Catalog”

Specyfikacja wymagań

Wymagania funkcjonalne:

· Możliwość transferu danych .

· Możliwość wybory startu serwera albo klienta.

· Możliwość połączenia z wybranym localhostem albo portem.

· Możliwość wybory plików do ładowania.

· Możliwość wybory miejsca zapisu.

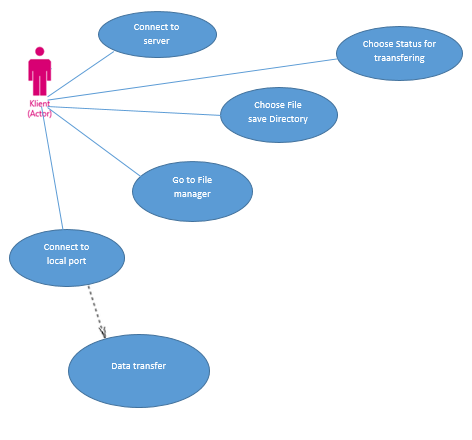
· Możliwość start, pause, clear ładowań.

Wymagania niefunkcjonalne:

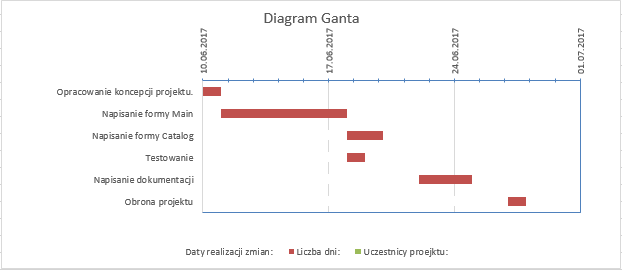
· Aplikacja musi zapewniać możliwość komunikacji na lokalnym komputerze.

· Framework do tworzenia form – WinForms, Metro.

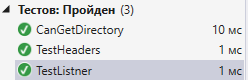
· Przechwytywania błędów(w razie jakichkolwiek błędów pokazać komunikat użytkownikowi)



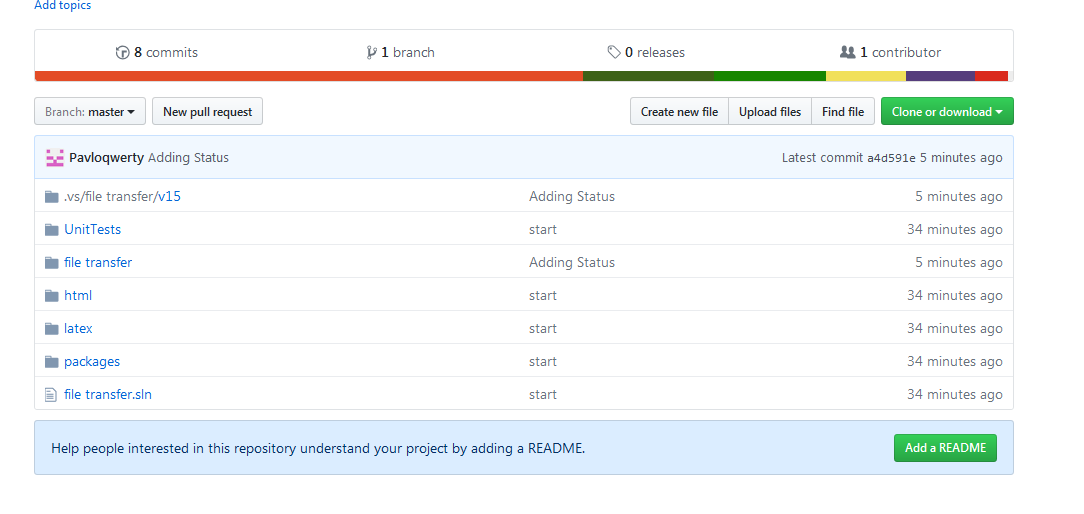
Rys.9. Use case diagram



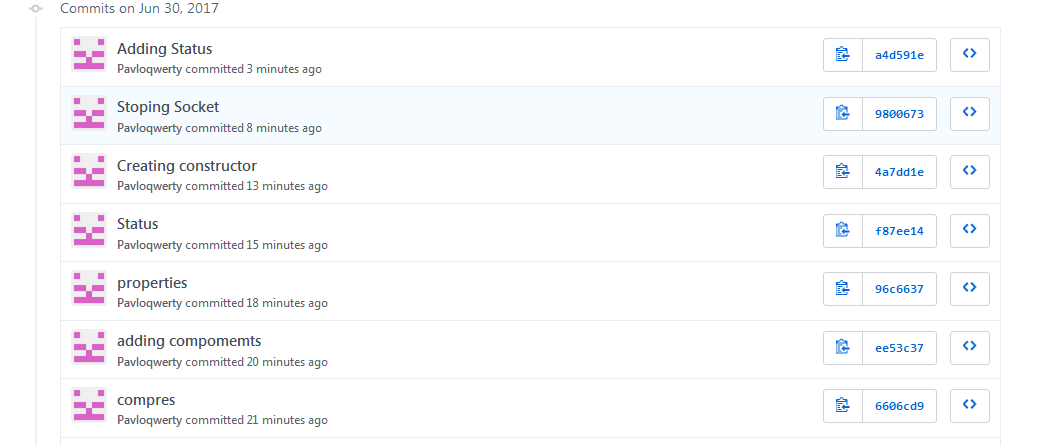
Rys.10. Diagram Gantta



Rys.11. Testy z pozytywnym wynikiem



Rys.12. Zrzut ekranu z Git

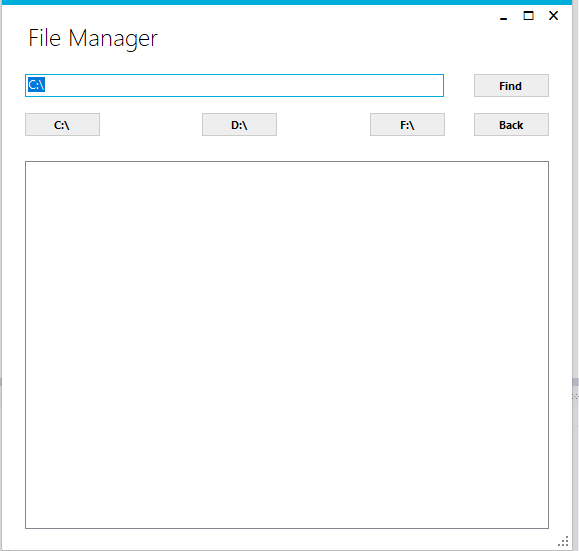


Rys.13. Zrzut ekranu z Git

**3. Forma managera plików**

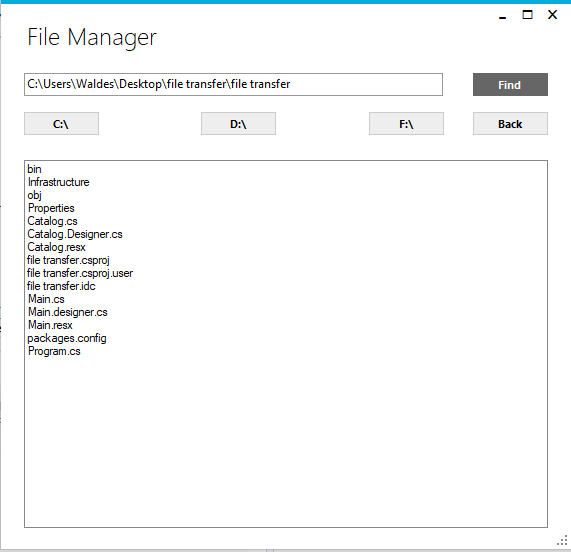
Na ten moment forma managera plików pracuje tak samo jak i manager plików Windows.

Użytkownik może używać ten manager zamiast zwykłego dla wyszukiwania potrzebnych plików.

****

Rys.15. Startowa forma managera plików

Ten manager plików otwiera wszystkie typy plików oraz te pliki, do których potrzebujemy specjalnych programów (na przykład : .pdf, .mp3, mpeg4).



Rys.16. Główna strona admin paneli

**4. Podsumowanie**

Podczas napisania pracy w praktyce użyłem swojej wiedzy na temat programowania obiektowego. Głębiej zrozumiałem podstawowe zasady programowania obiektowego. W praktyce użyłem podstawowych metod i środków programowania obiektowego. Nauczyłem się rozwijać hierarchię klas. Zastosowałem pojęcia inkapsulacja, dziedziczenie, przeładowanie funkcji, oraz nauczyłem się korzystać z dużej ilości nowych technologii.

**5. Literatura**

Źródła internetowe:

1. <https://msdn.microsoft.com/> (20.06.2017).
2. <https://github.com/Pavloqwerty/Project_File_transfer>
3. <https://metanit.com/> (20.06.2017).
4. <https://professorweb.ru/> (20.06.2017).
5. <https://gitter.im/> (20.06.2017).
6. <https://www.nuget.org/> (20.06.2017).
7. <http://stackoverflow.com/> (20.06.2017).
8. The C#Player`s Guide(RB Whitaker).
9. Murach`s C# 2012(Joe Murach, Anne Boehm)
10. C# 5.0 Unleashed(Bart De Smith).

**6. Załączniki**

Załącznik 1 – projekt w postaci kodu w55211.rar

Załącznik 1 – projekt w postaci kodu w55204.rar