**Systemy Równoległe i rozproszone**

**Sprawozdanie z Projektu**

Członkowie zespołu:

* Tomasz Tomala
* Konrad Tabiś
* Kamil Wasilonek
* Rafał Szyszka

1. **Cel projektu**

Celem projektu było zapoznanie się z funkcjonowaniem systemów rozproszonych. Na podstawie zgromadzonych danych teoretycznych, należało opracować aplikację umożliwiającą przesyłanie kodu źródłowego programu, sprawdzanie poprawności składni a następnie kompilację i wykonanie dostarczonego pliku. Po każdej części należało wysłać informację zwrotną o postępie programu. Wykonanie założeń projektu pozwoli lepiej zrozumieć działanie nowych i coraz popularniejszych technologii rozproszonych.

1. **Realizacja projektu**

Aby spełnić założenia projektu, początkowo każda osoba z grupy odszukała i zapoznała się z wiadomościami teoretycznymi na temat technologii z której korzystano podczas tworzenia aplikacji. Następnie zrealizowano następujące kroki implementacyjne:

Osoba 1: Stworzenie mechanizmu przesyłania kodu źródłowego pomiędzy komputerami.

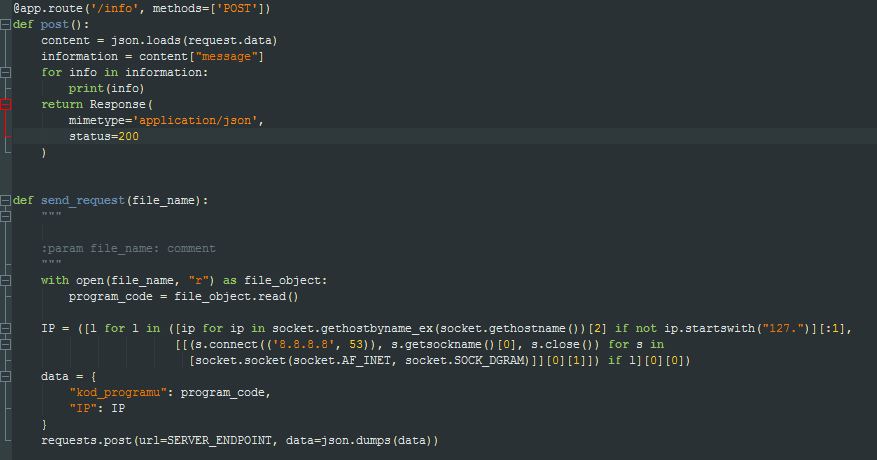
Osoba 2: Stworzenie systemu sprawdzania poprawności składni dostarczonego kodu oraz wysłania informacji zwrotnej.

Osoba 3: Wykonanie otrzymanego kodu źródłowego oraz wysłanie informacji zwrotnej.

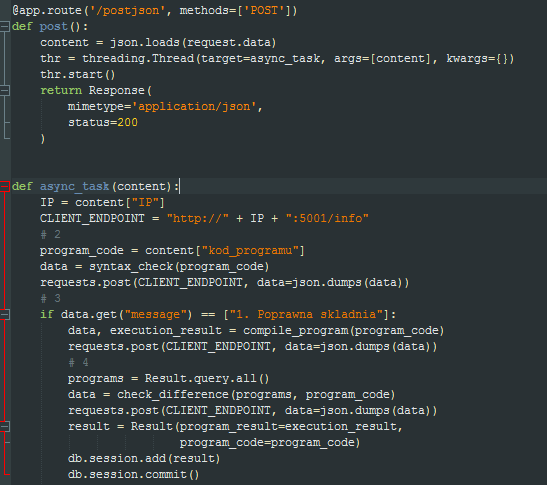
Osoba 4: Utworzenie mechanizmu składowania i porównywania przesłanych programów w wykorzystanie bazy danych.

Wszystkie osoby brały udział w przygotowaniu prezentacji z projektu (omawiając część programu realizowaną przed daną osobę) jak również opisu wykorzystanych technologii. Dodatkowo do każdej części programu napisane zostały testy sprawdzające poprawność stworzonej aplikacji.

1. **Listing kodu**

****

*Rys.1 Kod klienta*

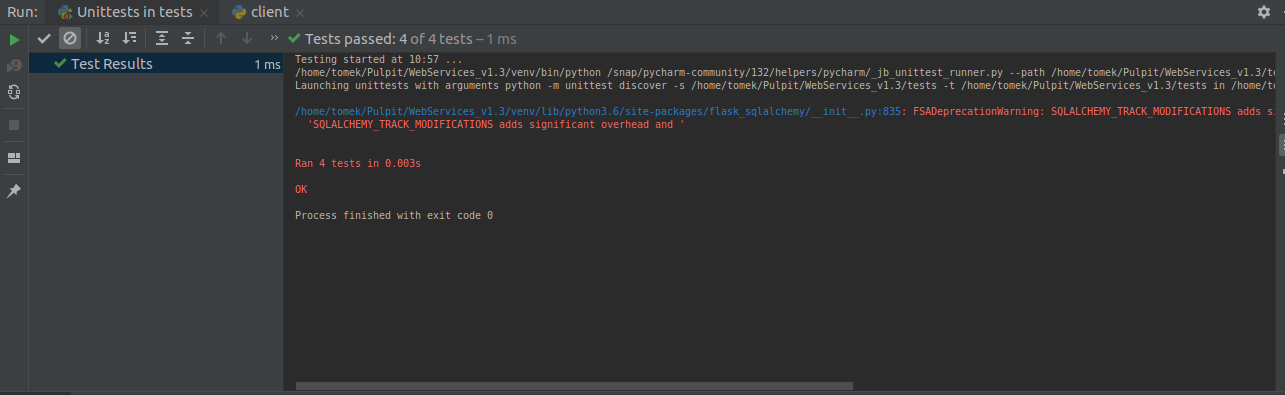
****

*Rys.2 Kod serwera*

****

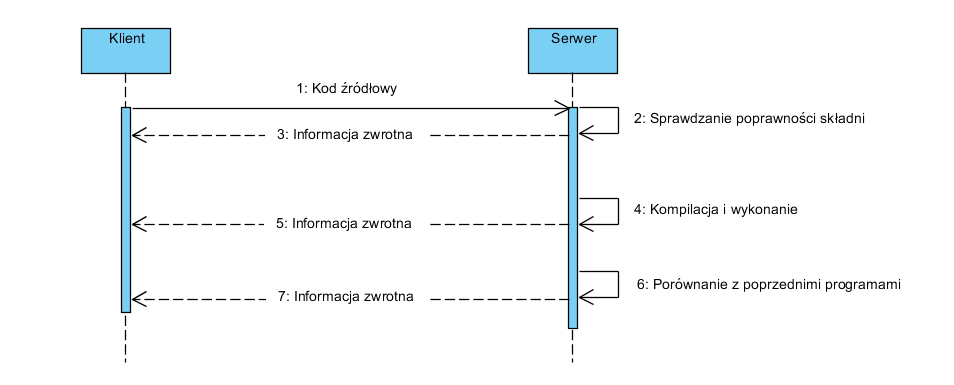
*Rys.3 Kod serwera*

1. **Wyniki testów**

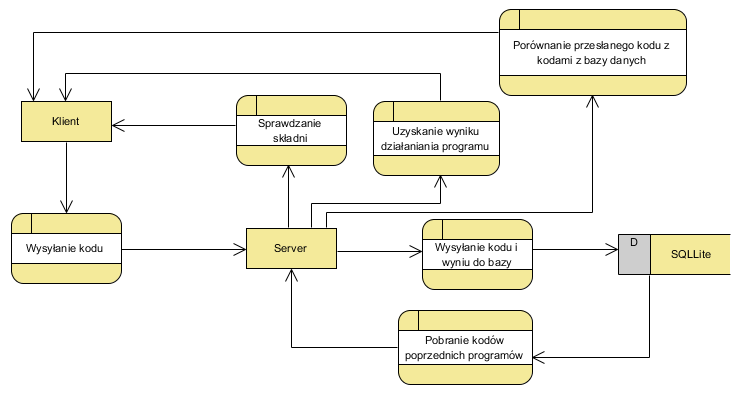
****

*Rys.4 Wynik testów*

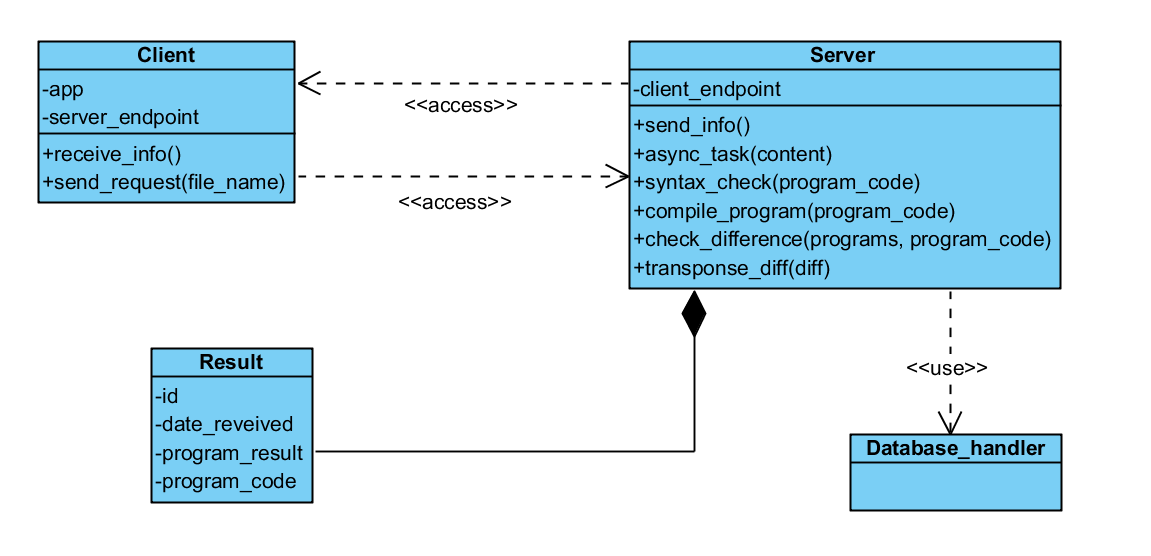
1. **Diagramy UML**

****

*Rys. 5 Diagram sekwencji*

****

*Rys. 6 Diagram przepływu danych*

****

*Rys. 7 Diagram klas*

1. **Nakład pracy**

Tab. 1 Nakład pracy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | K.Wasilonek | R.Szyszka | T.Tomala | K.Tabiś |
| przygotowanie diagramów | 1,5 | 1 | 1 | 1 |
| przygotowanie elementów prezentacji | 2 | 1,5 | 1,5 | 1,5 |
| wstęp teoretyczny | 1.5 | 1 | 1 | 1 |
| testowanie kodu i nanoszenie poprawek | 2 | 2,5 | 2,5 | 2 |
| implementacja | 3,5 | 4 | 3 | 4 |
| zapoznanie się z technologią | 1.5 | 2 | 3,5 | 2 |

*Rys. 8 Wykres nakładu pracy*

1. **Podsumowanie**

Na podstawie utworzonej aplikacji stwierdzono, że udało się osiągnąć postawione we wstępie założenia. Na podstawie wiedzy z zakresu działania systemów rozporoszonych opracowano poprawnie działający program do przesyłania kodu źródłowego, sprawdzania jego poprawności składniowej, kompilacji i wykonania. Każda część programu obsługuję również wysyłanie informacji zwrotnej, umożliwiającego monitorowanie kolejnych etapów wykonania. Dodatkowo dodane zostały zabezpieczenia gwarantujące niezawodną pracę wielu komputerów. Aby zapewnić niezawodną pracę oraz sprawdzić poprawność zaimplementowanej aplikacji wykonano testy jednostkowe kodu.

1. **Bibliografia**

* https://www.python.org/doc/, **oficjalna dokumentacja języka Python** (dostęp 08.06.2019)
* http://www.metal.agh.edu.pl/~banas/SRR/SRR\_W10\_CORBA\_WebServices\_Grids.pdf, **strona z materiałami dydaktycznym**, **dr hab. inż. Krzysztof Banaś, Akademia Górniczo-Hutnicza w Krakowie** (dostęp 08.06.2019)
* http://edu.pjwstk.edu.pl/wyklady/tin/scb/wyklad13/wyklad.html , **wykład na temat WebService, Polsko-Japońska Akademia Technik Komputerowych** (dostęp 08.06.2019)
* http://flask.pocoo.org/docs/1.0/testing/, **dokumentacja testów jednostkowych** (dostęp 08.06.2019)
* https://docs.python.org/3/library/unittest.html, **dokumentacja testów jednostkowych** (dostęp 08.06.2019)