Politechnika Krakowska

Wydział Inżynierii Elektrycznej i Komputerowej

Dokumentacja

**Programowanie w języku Java**

**Projekt: Aplikacja do nakładania filtrów na zdjęcia**

**Wykonali:**

Paweł Wądolny

Marcin Zieliński

PK WIEiK stacjonarne informatyka rok 3

Spis treści

[1. Cel i zakres projektu: 3](#_Toc504476943)

[2. Charakterystyka użytkowników 3](#_Toc504476944)

[3. Główne funkcje produktu 3](#_Toc504476945)

[4. Wymagania Funkcjonalne 3](#_Toc504476946)

[5. Wymagania Niefunkcjonalne 4](#_Toc504476947)

[6. Architektura i technologie 4](#_Toc504476948)

[7. Połączenie urządzeń 4](#_Toc504476949)

[8. GUI 5](#_Toc504476950)

[8.1 GUI Klient 5](#_Toc504476951)

[8.2 GUI Scheduler Server 5](#_Toc504476952)

[8.3 GUI Processing Server 5](#_Toc504476953)

[9. Klasy i metody 5](#_Toc504476954)

[10. Wyniki z działania systemu 5](#_Toc504476955)

[11. Raporty z szeregowania 5](#_Toc504476956)

[12. Tabele zbiorcze z testami wydajności 5](#_Toc504476957)

[13. Wnioski 5](#_Toc504476958)

1. Cel i zakres projektu:

Celem projektu jest stworzenie aplikacji w języku Java do nakładania filtrów na zdjęcia. Aplikacja będzie działała w następujący sposób: Host (klient) po wybraniu pliku graficznego oraz filtru będzie łączył się z serwerem kolejkującym (Scheduler Server), który sprawdzi dostępność serwerów przetwarzających (Processing Server) oraz ich obciążenie i prześle adres wybranego serwera do klienta, a ten następnie połączy się z serwerem przetwarzającym. Nastąpi przesłanie pliku obrazu do serwera, który po odebraniu obrazu nałoży na niego wybrany filtr, a następnie odeśle zmienioną fotografię do klienta.

Aplikacja musi być stabilna oraz prosta w obsłudze. Aplikacja będzie podzielona na 3 typy działania:

* Scheduler Server – Przechowuje adresy serwerów przetwarzających, komunikuje się z nimi w celu określenia ich obciążenia. Z tym serwerem łączy się klient, aby uzyskać dane serwera, do którego wyśle plik graficzny.
* Processing Server – serwer przetwarzający, jego zadaniem jest udzielenie informacji serwerowi kolejkującemu oraz odebranie, przetworzenie i odesłanie pliku graficznego klienta.
* Host – Aplikacja kliencka, umożliwia wybranie zdjęcia z dysku twardego oraz filtru do nałożenia. Aplikacja łączy się z Schedulerem w celu uzyskania adresu serwera przetwarzającego. Następnie łączy się z tym serwerem, wysyła wybrany plik oraz informację o wybranym filtrze. Po otrzymaniu pliku zwrotnego z serwera zapisuje go na dysku.

Zakres projektu obejmuje stworzenie aplikacji z intuicyjnym GUI. Aplikacja dodatkowo będzie posiadała symulator hostów/klientów w celu zasymulowania obciążenia serwera. Aplikacja ta będzie uruchamiana na jednym komputerze i przy wykorzystaniu wielowątkowości zostanie zasymulowane obciążenie serwerów.

2. Charakterystyka użytkowników

* **Administrator** – jego zadaniem jest połącznie serwerów oraz wpisanie w pliku konfiguracyjnym adresów serwerów (Schedulera i Processing), a następnie uruchomienie serwerów.
* **Użytkownik** – ma możliwość korzystania z aplikacji Host. Po wybraniu danego zdjęcia i filtru otrzyma ze serwera przetworzony plik graficzny.

3. Główne funkcje produktu

1. Serwer kolejkujący (Scheduler Server)
2. Serwer przetwarzający (Processing Server)
3. Aplikacja kliencka (Host)
4. Aplikacja symulująca klientów i testująca obciążenie serwerów (HostSimulator)
5. Przetwarzanie obrazu przy użyciu wybranego filtru

4. Wymagania Funkcjonalne

1. Łączność z serwerem kolejkującym
2. Łączność z serwerem przetwarzającym
3. Przesyłanie przez klienta pliku do serwera
4. Przetwarzanie obrazu na serwerze
5. Odesłanie wyniku do klienta

5. Wymagania Niefunkcjonalne

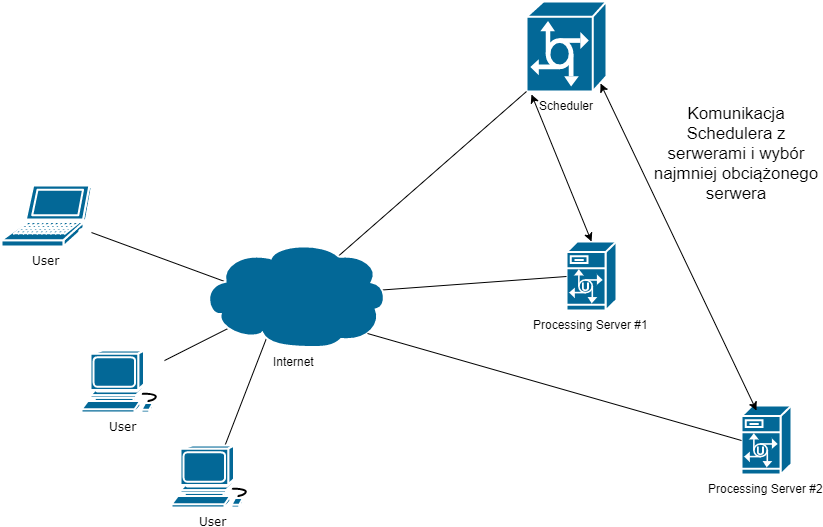
1. Stabilność – aplikacja musi działać niezawodnie
2. Intuicyjność – aplikacja powinna być przyjazna dla użytkownika; niedoświadczony klient nie może mieć problemów z obsługą
3. Łatwość instalacji – system powinien być możliwy do wdrożenia w jeden dzień roboczy

6. Architektura i technologie

1. Łączność pomiędzy serwerami i klientami będzie zrealizowana przy użyciu protokołu TCP/IP
2. GUI Hosta będzie stworzone przy użyciu JavaFX
3. Plik po przetworzeniu będzie kasowany z serwera
4. Aplikacja będzie obsługiwała typy plików graficznych: \*.jpg, \*.jpeg, \*.png, \*.bmp
5. Maksymalny rozmiar pliku będzie ograniczony do około 10-20MB, daje to około   
   10000px x 10000px dla formatów jpg, jpeg. Zabieg ten jest wprowadzony w celu podzielenia mocy obliczeniowej na klientów oraz uwarunkowany wystąpieniem błędu java.lang.OutOfMemoryError przy pracy z bardzo dużymi plikami (np. plik \*.bmp 600MB 20000px x 10000px).
6. Wstępnie dostępne filtry: Sepia, Negatyw, MirrorImage, Black&White, Red Image, Green Image, Blue Image.

Jeżeli znajdziemy inne modyfikacje obrazu godne uwagi zostaną one zawarte w projekcie.

7. Połączenie urządzeń



Symulacja aplikacji odbędzie się przy użyciu switcha zamiast sieci internet, ponieważ musielibyśmy posiadać publiczne ip, oraz w miarę szybki upload.

8. GUI

## 8.1 GUI Klient

Blabal

## 8.2 GUI Scheduler Server

Blabl

Babl

## 8.3 GUI Processing Server

dsadf

9. Klasy i metody

10. Wyniki z działania systemu

11. Raporty z szeregowania

12. Tabele zbiorcze z testami wydajności

13. Wnioski

Program działa poprawnie i jest odporny na błędy.