

RAPORT KOŃCOWY - POS

Artur Radko 188805

Marek Omilian 188759

Jan Reza 188866

Łukasz Saldat 188714

1.

1. Opracowanie wymagań, modele UML <ul style="list-style-type: none">• zidentyfikować klienta, użytkowników końcowych i inne zainteresowane strony tematu (stakeholders);• zgromadzić i przeanalizować szczegółowe wymagania funkcjonalne i pozafunkcjonalne zadań;• określić scenariusze użycia i stworzyć nadającą się do czytania specyfikację funkcjonalną;• opcjonalnie zawrzeć w niej diagramy UML scenariuszy użycia;• stworzyć model UML kluczowych elementów projektowanego rozwiązania;• przeanalizować możliwość użycia dodatkowych bibliotek i narzędzi do realizacji zadania;• opracować krótką specyfikację techniczną podsumowującą te informacje wraz z harmonogramem pracy;• przedstawić specyfikację do zaliczenia;• przygotować harmonogram pracy oraz kamienie milowe.	Microsoft Visio	10
---	-----------------	----

Opracowanie wymagań













2.

<p>2. Napisanie oprogramowania</p> <p>Oprogramowanie realizującego przetwarzanie obrazów (po uzgodnieniu z prowadzącym, można realizować inny, własny projekt)</p> <p>Założenia:</p> <ul style="list-style-type: none">• program uruchamiany z linii poleceń;• program z pliku *.ini powinien odczytać ścieżki źródłowe i docelowe (do analizy pliku ini można wykorzystać dowolną darmową bibliotekę, np. iniParser);• program odczytuje wszystkie pliki z obrazami ze ścieżki źródłowej;• program wykonuje przetworzenie odczytanych obrazów (wielowątkowo) - detekcja konturów;• przetworzone obrazy zapisujemy w folderze docelowym;• w folderze docelowym zapisujemy matrycę miniatur plików źródłowych oraz matrycę miniatur obrazów docelowych (jeden plik zawierać powinien sklejone ze sobą zmniejszone obrazki źródłowe, drugi plik - sklejone obrazki po przetworzeniu, aby można było szybko zobaczyć efekty działania programu). <p>Proponuję najpierw zapoznać się z poszczególnymi elementami projektu:</p> <ul style="list-style-type: none">• analiza pliku INI;• odczytywanie, zapisywanie i wyświetlanie obrazów w OpenCV;• zasady operacji na macierzach/obrazach w OpenCV;• wybranie i sprawdzenie metody przetwarzania obrazów. <p>Uwagi dodatkowe:</p> <ul style="list-style-type: none">• w pliku INI podajemy ścieżkę do folderu z plikami wejściowymi - program powinien przeczytać wszystkie pliki z tego folderu;• program powinien umożliwiać pracę z dużą liczbą plików >500;• obrazki miniaturki powinny mieć zachowane proporcje wysokość/szerokość; <p>Zdjęcia testowe do testowania oprogramowania (ZIP)</p>	<p>Visual Studio 2013 i C++ wraz z biblioteką OpenCV</p>	<p>20*</p>
---	--	------------

3.













3. Wykorzystanie narzędzia do pracy grupowej <ul style="list-style-type: none">• uzyskać dostęp do repozytorium systemu kontroli wersji (zewnętrzny hosting GitHub, Bitbucket.);• wyznaczyć osobę odpowiedzialną za utrzymanie porządku w repozytorium (maintainer);• zaplanować strukturę repozytorium mając na uwadze używane środowisko IDE, język programowania, dodatkowe artefakty (grafiki UI, specyfikacje itp.).	GIT + GitHub	10
--	--------------	----

- [GitHub](#)

 Rezoot final version 1.0 12e1b6b · 1 hour ago  27 Commits		
 build	Revert "coworker test"	5 days ago
 dokumentacja	documentation update	2 hours ago
 outImages	documentation update	2 hours ago
 testImages	documentation update	2 hours ago
 CMakeLists.txt	clean	2 days ago
 Datas.cpp	final version 1.0	1 hour ago
 DatasImages.h	final version 1.0	1 hour ago
 Images.cpp	final version 1.0	1 hour ago
 config.ini	final version 1.0	1 hour ago
 main.cpp	final version 1.0	1 hour ago

4.

4. Wykorzystanie systemu śledzenia błędów <ul style="list-style-type: none">• uzyskać dostęp do repozytorium systemu śledzenia błędów (zewnętrzny hosting GitHub, Bitbucket.);• wyznaczyć osobę odpowiedzialną za utrzymanie porządku w repozytorium (maintainer).• wykryte podczas cykli testowania błędy raportować do systemu śledzenia błędów;• opcjonalnie zastosować metodę test-driven development do zapewnienia jakości;• zademonstrować użycie testów jednostkowych do walidacji kluczowych elementów oprogramowania.	Bugzilla	10
--	----------	----

Discover our MantisHub Exclusive Modern UI! Access it via the "Modern UI" left nav icon or click here to make it your default experience! ×								
Zobacz 5 [Filtruj] ▾ ▾ 🔍								
P	Identyfikator	 	Kategoria	Ważność	Status	Przypisany do	Data modyfikacji 	Temat
✓	00002	1	General	drobny	 rozwiązany	 @jan	2025-06-11	Dodanie komentarzy
—	00005	1	General	ważny	 nowy		2025-06-09	matryca miniaturk oryginalnych obrazów nie wyświetla się
↑	00004		General	poprawka	 przypisany	 @administrator	2025-06-07	Brak tworzenia folderu wyjściowego.
—	00003		General	drobny	 przypisany	 @Lukasz	2025-06-07	Podział klas
^	00001		General	ważny	 przypisany	 @artur	2025-06-07	pliki wejściowe nie mają sprawdzonego rozszerzenia
<< < 1 > >>								

- [Bugzilla](#)

5.

5. Wykorzystanie wątków	Windows	10
-------------------------	---------	----

```
// Tworzenie instancji obiektu Images i listy watkow
Images images;
std::vector<std::thread> threads;
int count = 0;

// Przetwarzanie plikow z rozszerzeniem .jpg i .png
processFilesWithExtension("jpg", images, threads, count);
processFilesWithExtension("png", images, threads, count);

// Oczekiwanie na zakonczenie wszystkich watkow
for (auto& t : threads) {
    if (t.joinable()) {
        t.join();
        count++;
    }
}
```

6.

6. Wykorzystanie IPC	Windows	10
----------------------	---------	----

7.

7. Wykonanie dokumentacji projektu <ul style="list-style-type: none">• podczas implementacji zaangażować odpowiedni nakład pracy na dokumentowanie API (komentarze zgodne z Doxygen) oraz inne komentarze w kodzie, wyjaśniające szczególnie zagmatwane algorytmy, mechanizmy, zależności;• generować dokumentację HTML z komentarzy Doxygen na bieżąco, a nie po zakończeniu projektu;• przedstawić dokumentację do zaliczenia.	Doxygen	10
---	---------	----

Commits

History for POS_edge / dokumentacja on main

Commits on Jun 11, 2025

documentation update

marek12320 committed 2 hours ago

98ef779

Commits on Jun 7, 2025

dok

marek12320 committed 4 days ago

5887b13

first documentation draft

marek12320 committed 4 days ago

a6ee1b5

End of commit history for this file

POS_Edge 1.0

Main PageClassesFiles

POS_Edge

Classes

Class List

Datas

- Datas
- save2path
- fileINI
- inputDir
- outputDir

Images

- processImage
- showImageGrid

Class Index

Class Members

- All
- Functions
- Variables

Files

- File List
- File Members

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

C Datas

Forward declaration klasy Images

C Images

Klasa odpowiedzialna za przetwarzanie obrazów (np. detekcja krawędzi)

8.

8. Wykonanie profilowania projektu (hotspot'y) <ul style="list-style-type: none">• użyć profiler do wykrycia hot spotów w opracowanym programie i przeprowadzić próbę ich optymalizacji.	Visual Studio Profiler	10
---	------------------------	----

9.

9. Wykonanie wykrywania błędów <ul style="list-style-type: none">• Sprawdzić integralność danych w programie, np.:<ul style="list-style-type: none">◦ wykryć "memory leak"◦ wykryć wykraczanie wskaźników (np. korzystając z makra <code>_ASSERT(_CrtCheckMemory())</code>; w trybie DEBUG)	Visual Studio + CrtCheckMemory	10
--	-----------------------------------	----