

- 1: Wstęp napiszemy na końcu.
- 5: słowo zagajenia – jakieś zdanie wprowadzenia zanim pojawi się spis użytkowników
- 5: pauza ‘–’, nie dywiz ‘-’
- 5: Może byłoby lepiej tak jak niżej?
- 5: pauza ‘–’, nie dywiz ‘-’
- 6: słowo zagajenia
- 6: Rys. 3.1: Rysunek podzieliłbym na trzy osobne dla każdego aktora plus rysunek z relacjami pomiędzy aktorami.
- 8: W jaki sposób zweryfikujemy, czy to wymaganie zostało spełnione?
- 8: To jest wymaganie funkcjonalne.
- 8: W jaki sposób zweryfikujemy, czy to wymaganie zostało spełnione?
- 8: cytowanie dokumentacji, strony, ...
- 8: cytowanie dokumentacji
- 8: cytowanie dokumentacji
- 8: cytowanie dokumentacji
- 8: cytowanie dokumentacji
- 8: cytowanie dokumentacji



**Politechnika
Śląska**

PROJEKT INŻYNIERSKI

Tytuł pracy dyplomowej inżynierskiej

Imię NAZWISKO

Nr albumu: **⟨wpisać właściwy⟩**

Kierunek: **⟨wpisać właściwy⟩**

Specjalność: **⟨wpisać właściwą⟩**

PROWADZĄCY PRACĘ

⟨tytuł lub stopień naukowy oraz imię i nazwisko⟩

KATEDRA ALGORYTMIKI I OPROGRAMOWANIA

Wydział Automatyki, Elektroniki i Informatyki

OPIEKUN, PROMOTOR POMOCNICZY

⟨stopień naukowy imię i nazwisko⟩

Gliwice 2024

Tytuł pracy

Tytuł pracy dyplomowej inżynierskiej

Streszczenie

(Streszczenie pracy – odpowiednie pole w systemie APD powinno zawierać kopię tego streszczenia.)

Słowa kluczowe

(2-5 słów (fraz) kluczowych, oddzielonych przecinkami)

Thesis title

Thesis title in English

Abstract

(Thesis abstract – to be copied into an appropriate field during an electronic submission – in English.)

Key words

(2-5 keywords, separated by commas)

Spis treści

1	Wstęp	1
2	[Analiza tematu]	3
3	Wymagania i narzędzia	5
3.1	Wymagania funkcjonalne	5
3.1.1	Użytkownicy	5
3.1.2	Funkcjonalności	6
3.2	Wymagania niefunkcjonalne	8
3.3	Narzędzia	8
4	Specyfikacja zewnętrzna	9
5	Specyfikacja wewnętrzna	11
6	Weryfikacja i walidacja	13
7	Podsumowanie i wnioski	15
	Bibliografia	17
	Spis skrótów i symboli	21
	Źródła	23
	Lista dodatkowych plików, uzupełniających tekst pracy	25
	Spis rysunków	27
	Spis tabel	29

Rozdział 1

Wstęp

[Wstęp napiszemy na końcu.]

- wprowadzenie w problem/zagadnienie
- osadzenie problemu w dziedzinie
- cel pracy
- zakres pracy
- zwięzła charakterystyka rozdziałów
- jednoznaczne określenie wkładu autora, w przypadku prac wieloosobowych – tabela z autorstwem poszczególnych elementów pracy

Rozdział 2

[Analiza tematu]

- sformułowanie problemu
- osadzenie tematu w kontekście aktualnego stanu wiedzy (*state of the art*) o poruszonym problemie
- studia literaturowe [3, 4, 2, 1] - opis znanych rozwiązań (także opisanych naukowo, jeżeli problem jest poruszany w publikacjach naukowych), algorytmów,

Wzory

$$y = \frac{\partial x}{\partial t} \tag{2.1}$$

jak i pojedyncze symbole x i y składa się w trybie matematycznym.

Rozdział 3

Wymagania i narzędzia

- wymagania funkcjonalne i нефункционалне
- przypadki użycia (diagramy UML) – dla prac, w których mają zastosowanie
- opis narzędzi, metod eksperymentalnych, metod modelowania itp.
- metodyka pracy nad projektowaniem i implementacją – dla prac, w których ma to zastosowanie

3.1 Wymagania funkcjonalne

3.1.1 Użytkownicy

[słowo zagajenia – jakieś zdanie wprowadzenia zanim pojawi się spis użytkowników]

- U.1. Użytkownik zwykły: edytuje profil, tworzy, przegląda, modyfikuje, usuwa, udostępnia elementy.
- U.2. Administrator: posiada dostęp do wszystkich funkcjonalności użytkownika zwykłego, zarządza użytkownikami - [pauza ‘-’, nie dywiz ‘-’] nadaje i odbiera uprawnienia administratorskie, usuwa użytkowników.
- U.3. Użytkownik niezalogowany: loguje lub rejestruje się.

[Może byłoby lepiej tak jak niżej?]

U.1 Użytkownik zwykły: edytuje profil, tworzy, przegląda, modyfikuje, usuwa, udostępnia elementy.

U.2 Administrator: posiada dostęp do wszystkich funkcjonalności użytkownika zwykłego, zarządza użytkownikami - [pauza ‘-’, nie dywiz ‘-’] nadaje i odbiera uprawnienia administratorskie, usuwa użytkowników.

U.3 Użytkownik niezalogowany: loguje lub rejestruje się.

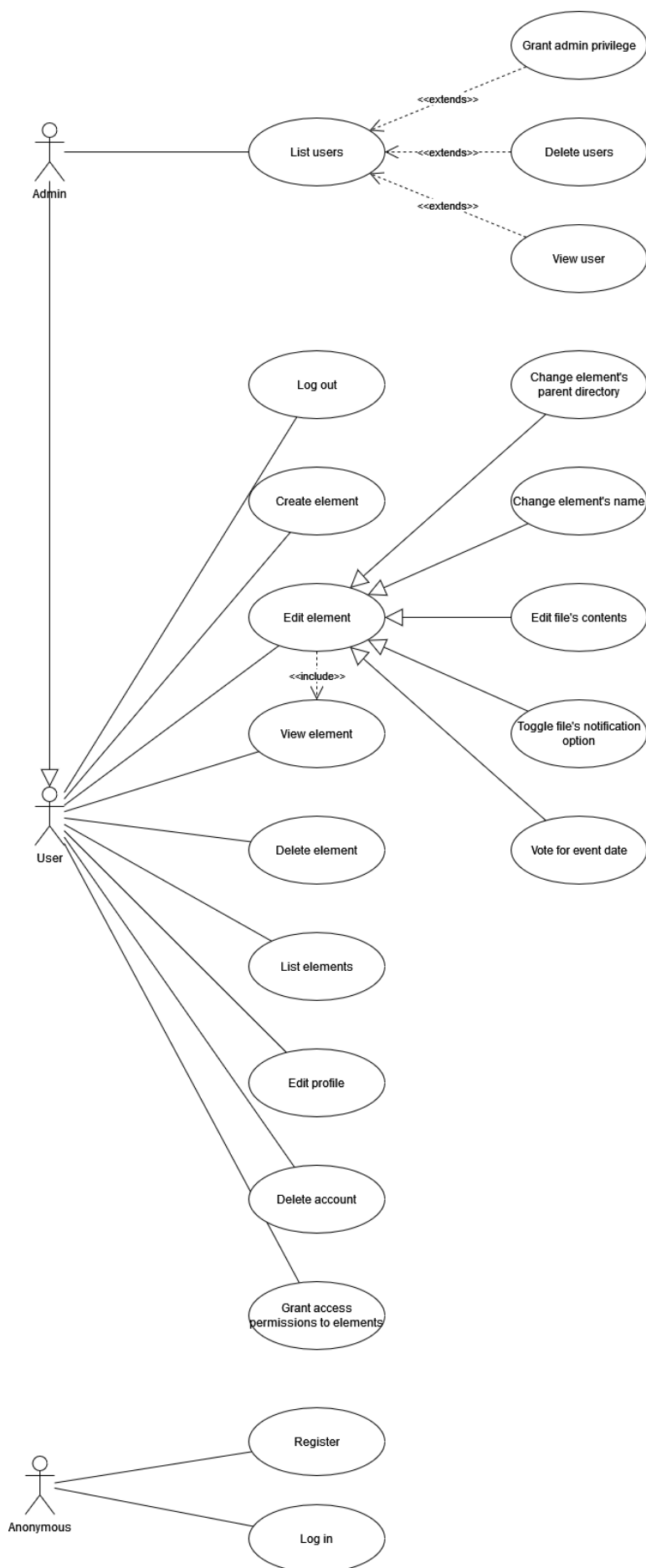
3.1.2 Funkcjonalności

[słowo zagajenia]

- F.1. Autentykacja użytkowników
 - F.1.1. Logowanie użytkownika U.3. na istniejące konto przy pomocy nazwy użytkownika oraz hasła.
 - F.1.2. Wylogowanie zalogowanego użytkownika (U.1., U.2.).
 - F.1.3. Rejestracja nowego użytkownika w systemie, umożliwiającą następnie zalogowanie (por. F.1.1.).
- F.2. Zarządzanie elementami.
 - F.2.1. Tworzenie nowego elementu.
 - F.2.2. Wyświetlenie elementu. Dla należącego do innego użytkownika - zależne od przydzielonego dostępu (por. F.2.5.).
 - F.2.3. Edycja elementu.
 - * F.2.3.1. Przeniesienie do innego katalogu.
 - * F.2.3.2. Zmiana nazwy.
 - * F.2.3.3. Modyfikacja zawartości. Dla należącego do innego użytkownika - zależne od przydzielonego dostępu (por. F.2.5.).
 - * F.2.3.4. Modyfikacja opcji powiadomień.
 - * F.2.3.5. Oddawanie głosu na termin wydarzenia.
 - F.2.4. Usunięcie elementu.
 - F.2.5. Przydział dostępu innym użytkownikom do elementów.
- F.3. Zarządzanie kontem użytkownika.
 - F.3.1. Edycja danych.
 - F.3.2. Usunięcie konta.
- F.4. Zarządzanie użytkownikami przez administratora (U.2.).
 - F.4.1. Wyświetlenie wszystkich użytkowników systemu.
 - F.4.2. Wyświetlenie jednego użytkownika.
 - F.4.3. Nadanie bądź odebranie użytkownikowi uprawnień administratora.
 - F.4.4. Usunięcie użytkownika.

Wymagania funkcjonalne przedstawiono na Rys. [rys.] 3.1.

[Rys. 3.1: Rysunek podzieliłbym na trzy osobne dla każdego aktora plus rysunek z relacjami pomiędzy aktorami.]



Rysunek 3.1: Diagram przypadków użycia aplikacji.

3.2 Wymagania нефunkcjonalne

Dla aplikacji wymagającej zalogowania przez użytkownika, służącej do przechowywania prywatnych informacji niezbędne jest zapewnienie bezpieczeństwa danych przed potencjalnym przechwyceniem. [W jaki sposób zweryfikujemy, czy to wymaganie zostało spełnione?]

Ze względu na dużą ilość danych wprowadzanych przez użytkownika, powinna zostać zapewniona odporność na nieprawidłowe dane wejściowe. [To jest wymaganie funkcjonalne.]

Istotna również jest intuicyjność interfejsu użytkownika. [W jaki sposób zweryfikujemy, czy to wymaganie zostało spełnione?]

3.3 Narzędzia

Do implementacji back-endu wybrano framework Spring [cytowanie dokumentacji, strony, ...], pozwalający m.in. na pisanie aplikacji z wykorzystaniem Java Persistence API (Spring Data JPA) oraz proste zaimplementowanie autentykacji i autoryzacji użytkowników (Spring Security). Za wybraniem tego frameworku zamiast np. Entity Framework [cytowanie dokumentacji] przemawiała również posiadana już wiedza o REST API oraz chęć pogłębienia znajomości samego Springa. Wspieraną przez Spring bazę danych MySQL [cytowanie dokumentacji] oraz sam back-end postanowiono uruchomić w kontenerze Dockera [cytowanie dokumentacji].

Do wykonania front-endu posłużył framework Angular [cytowanie dokumentacji]. Podobnie jak w przypadku Springa, za tym wyborem przemawiało pragnienie pogłębienia wstępnej wiedzy o frameworku. Subiektywną zaletą jest też, w przeciwieństwie do popularniejszego ¹ Reacta [cytowanie dokumentacji], wyraźny rozdział na pliki HTML, CSS (SCSS) oraz TypeScript w ramach danego komponentu. Wykorzystano również komponenty Angular Material, dzięki którym w prosty sposób uzyskać można estetycznie wyglądający interfejs graficzny, pozwalając na skupienie się na samej logice programu. Angular Material zawiera większość potrzebnych w projekcie komponentów, jednak brakuje w nim komponentu do wizualizacji wydarzeń w kalendarzu. W tym celu wykorzystano otwartoźródłowy (Apache License 2.0) pakiet DayPilot [cytowanie dokumentacji].

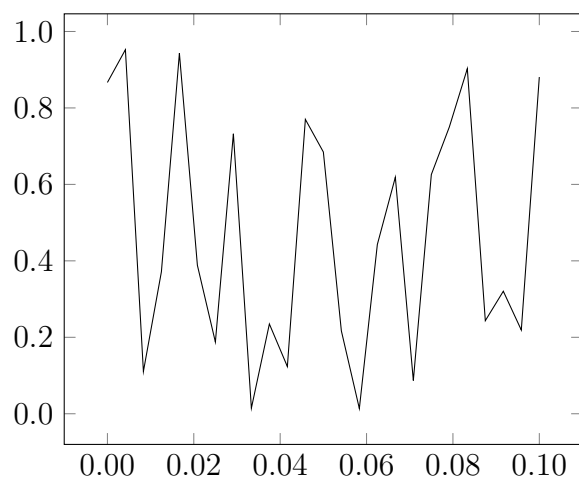
¹<https://trends.stackoverflow.co/?tags=reactjs,vue.js,angular,svelte,angularjs,vuejs3>, dostęp 08.11.2024

Rozdział 4

Specyfikacja zewnętrzna

Jeśli „Specyfikacja zewnętrzna”:

- wymagania sprzętowe i programowe
- sposób instalacji
- sposób aktywacji
- kategorie użytkowników
- sposób obsługi
- administracja systemem
- kwestie bezpieczeństwa
- przykład działania
- scenariusze korzystania z systemu (ilustrowane zrzutami z ekranu lub generowanymi dokumentami)



Rysunek 4.1: Podpis rysunku po rysunkiem.

Rozdział 5

Specyfikacja wewnętrzna

Jeśli „Specyfikacja wewnętrzna”:

- przedstawienie idei
- architektura systemu
- opis struktur danych (i organizacji baz danych)
- komponenty, moduły, biblioteki, przegląd ważniejszych klas (jeśli występują)
- przegląd ważniejszych algorytmów (jeśli występują)
- szczegóły implementacji wybranych fragmentów, zastosowane wzorce projektowe
- diagramy UML

Krótką wstawkę kodu w linii tekstu jest możliwa, np. **int a**; (biblioteka `listings`). Dłuższe fragmenty lepiej jest umieszczać jako rysunek, np. kod na rys 5.1, a naprawdę długie fragmenty – w załączniku.

```
1 class test : public basic
2 {
3     public:
4         test (int a);
5         friend std::ostream operator<<(std::ostream & s,
6                                         const test & t);
7     protected:
8         int _a;
9
10 };
```

Rysunek 5.1: Pseudokod w `listings`.

Rozdział 6

Weryfikacja i walidacja

- sposób testowania w ramach pracy (np. odniesienie do modelu V)
- organizacja eksperymentów
- przypadki testowe zakres testowania (pełny/niepełny)
- wykryte i usunięte błędy
- opcjonalnie wyniki badań eksperymentalnych

Tabela 6.1: Nagłówek tabeli jest nad tabelą.

ζ	metoda						
	alg. 1	alg. 2	alg. 3			alg. 4, $\gamma = 2$	
			$\alpha = 1.5$	$\alpha = 2$	$\alpha = 3$	$\beta = 0.1$	$\beta = -0.1$
0	8.3250	1.45305	7.5791	14.8517	20.0028	1.16396	1.1365
5	0.6111	2.27126	6.9952	13.8560	18.6064	1.18659	1.1630
10	11.6126	2.69218	6.2520	12.5202	16.8278	1.23180	1.2045
15	0.5665	2.95046	5.7753	11.4588	15.4837	1.25131	1.2614
20	15.8728	3.07225	5.3071	10.3935	13.8738	1.25307	1.2217
25	0.9791	3.19034	5.4575	9.9533	13.0721	1.27104	1.2640
30	2.0228	3.27474	5.7461	9.7164	12.2637	1.33404	1.3209
35	13.4210	3.36086	6.6735	10.0442	12.0270	1.35385	1.3059
40	13.2226	3.36420	7.7248	10.4495	12.0379	1.34919	1.2768
45	12.8445	3.47436	8.5539	10.8552	12.2773	1.42303	1.4362
50	12.9245	3.58228	9.2702	11.2183	12.3990	1.40922	1.3724

Rozdział 7

Podsumowanie i wnioski

- uzyskane wyniki w świetle postawionych celów i zdefiniowanych wyżej wymagań
- kierunki ewentualnych danych prac (rozbudowa funkcjonalna ...)
- problemy napotkane w trakcie pracy

Bibliografia

- [1] Imię Nazwisko i Imię Nazwisko. *Tytuł strony internetowej*. 2021. URL: <http://gdzies/w/internecie/internet.html> (term. wiz. 30.09.2021).
- [2] Imię Nazwisko, Imię Nazwisko i Imię Nazwisko. „Tytuł artykułu konferencyjnego”. W: *Nazwa konferencji*. 2006, s. 5346–5349.
- [3] Imię Nazwisko, Imię Nazwisko i Imię Nazwisko. „Tytuł artykułu w czasopiśmie”. W: *Tytuł czasopisma* 157.8 (2016), s. 1092–1113.
- [4] Imię Nazwisko, Imię Nazwisko i Imię Nazwisko. *Tytuł książki*. Warszawa: Wydawnictwo, 2017. ISBN: 83-204-3229-9-434.

Dodatki

Spis skrótów i symboli

DNA kwas deoksyrybonukleinowy (ang. *deoxyribonucleic acid*)

MVC model – widok – kontroler (ang. *model-view-controller*)

N liczebność zbioru danych

μ stopnień przyleżności do zbioru

\mathbb{E} zbiór krawędzi grafu

\mathcal{L} transformata Laplace’a

Źródła

Jeżeli w pracy konieczne jest umieszczenie długich fragmentów kodu źródłowego, należy je przenieść w to miejsce.

```
1 if (_nClusters < 1)
2     throw std::string ("unknown_number_of_clusters");
3 if (_nIterations < 1 and _epsilon < 0)
4     throw std::string ("You should set a maximal number of
        iteration or minimal difference — epsilon.");
5 if (_nIterations > 0 and _epsilon > 0)
6     throw std::string ("Both number of iterations and minimal
        epsilon set — you should set either number of iterations
        or minimal epsilon.");
```

Lista dodatkowych plików, uzupełniających tekst pracy

W systemie do pracy dołączono dodatkowe pliki zawierające:

- źródła programu,
- dane testowe,
- film pokazujący działanie opracowanego oprogramowania lub zaprojektowanego i wykonanego urządzenia,
- itp.

Spis rysunków

3.1	Diagram przypadków użycia aplikacji.	7
4.1	Podpis rysunku po rysunkiem.	10
5.1	Pseudokod w <code>listings</code>	11

Spis tabel

6.1	Nagłówek tabeli jest nad tabelą.	14
-----	--	----