

Uwagi

1: Wstępem zajmujemy się na końcu.

3: fleksja

3: cytowanie

3: bez przecinka

3: Rys. 2.1: Praca jest po polsku, to i rysunek musi być po polsku.

4: Rysunki, tabele i tym podobne elementy to w LaTeXu elementy pływające. Przy składzie tekstu LaTeX najpierw rozmieszcza je zgodnie z zasadami składu tekstów. Potem oblewa je tekstem. Tak się robi od czasów Gutenberga. Wstawianie elementów pływających na sztywno w konkretnym miejscu zwykle prowadzi do gorszego układu typograficznego. Mamy jako ludzkość bardzo duże doświadczenie (prawie pół tysiąclecia druku) i wiemy, co wygląda źle, a co wygląda dobrze. Wiemy, jaka powinna być długość wiersza, jakie powinno być wypełnienie strony tekstem (zaczernienie). LaTeX implementuje bardzo wiele algorytmów do optymalizacji składu zarówno makrotypograficznych (akapit, strona), jak i mikrotypograficznych (litery w słowie). Pierwotnie opcji [h] w LaTeXu nie było, bo jest niepotrzebna i szkodliwa, ale jakiś nieszczęśnik to doimplementował na zgubę swoją i innych.

4: (ang. *Computer-Assisted Language Learning*, CALL)

4: Skoro jest rozwijana, to musimy się podeprzeć jakimiś cytowaniami.

4: cytowanie

4: chorobę

4: po polsku

4: cytowanie

4: cytowanie

4: Co to znaczy: konwencjonalne algorytmy?

5: Skąd to wiemy? cytowanie

5: cytowanie

5: odnośnik do strony

5: czterdziestu trzech

5: odnośnik do strony

5: odnośnik do strony

5: Chyba trzeba pisać trochę ostrożniej. Pewnie znajdzie się ktoś komu nie zagwarantuje.

6: całym słowem

6: Tutaj napisałbym kilka słów o poziomach, ile znaków jest na poziomie itd.



**Politechnika
Śląska**

PROJEKT INŻYNIERSKI

Aplikacja mobilna do nauki znaków kanji

Aleksandra KYC

Nr albumu: 296410

Kierunek: Informatyka

Specjalność: Grafika Komputerowa i Oprogramowanie

PROWADZĄCY PRACĘ

dr hab. inż., prof. PŚ Krzysztof Simiński

KATEDRA Algorytmiki i Oprogramowania

Wydział Automatyki, Elektroniki i Informatyki

Katowice 2023

Tytuł pracy

Aplikacja mobilna do nauki znaków kanji

Streszczenie

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

Słowa kluczowe

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

Thesis title

Mobile application for Kanji learning

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
2.1	Nauka języków obcych wspomagana komputerowo	3
2.2	Analiza rozwiązań dostępnych na rynku	5
2.3	Nauka języka japońskiego i znaków kanji	5
3	Wymagania i narzędzia	7
4	[Właściwy dla kierunku – np. Specyfikacja zewnętrzna]	11
5	[Właściwy dla kierunku – np. Specyfikacja wewnętrzna]	13
6	Weryfikacja i walidacja	15
7	Podsumowanie i wnioski	17
	Bibliografia	19
	Spis skrótów i symboli	23
	Źródła	25
	Lista dodatkowych plików, uzupełniających tekst pracy	27
	Spis rysunków	29
	Spis tabel	31

Rozdział 1

Wstęp

[Wstępem zajmimy się na końcu.] Aby wyjaśnić cel tej pracy inżynierskiej, należy wspomnieć o kilku ważnych pojęciach związanych z językiem japońskim i jego nauką. Kanji razem z sylabariuszami hiragana i katakana oraz znanymi nam cyframi arabskimi i alfabetem rzymskim tworzą całość japońskiego alfabetu. Są to znaki logograficzne, co oznacza że przeciwnie do alfabetów składających się z głosek, każdy znak oznacza jakieś pojęcie. Kompleksowa lista wszystkich kanji nie istnieje, jednak ich liczbę szacuje się na około czterdziestu tysięcy. Jest to ogromna liczba, na którą składa się wiele znaków przestarzałych lub bardzo rzadko używanych. Zamiast niej częstym wskaźnikiem płynności językowej w kontekście kanji jest pojęcie *jōyō kanji*, które oznacza zestaw znaków wymagany przez japoński system edukacji. Po ukończeniu szkoły średniej Japończyk posługuje się 2136 znakami kanji. Podobna liczba potrzebna jest, aby zdać test znajomości języka japońskiego dla obcokrajowców (jap. *Nihongo nōryoku shiken*) na najwyższym poziomie oraz zawiera się w około 98 procentach tekstu pisanego.

Zaznajomienie się ze znakami kanji to nieodłączna część nauki języka japońskiego. Co prawda uparty uczeń może zaprzestać na nauce gramatyki, wymowy i zromanizowanego zapisu *rōmaji*, ale skutkuje to wątpliwą płynnością językową, gdyż nie będzie w stanie przeczytać nawet książki lub restauracyjnego menu. Dla początkującego tak duża ilość materiału może być odrzucająca, dlatego na rynku pojawiło się wiele rozwiązań, które pomagają użytkownikowi w nauce kanji. Jednym z takich rozwiązań jest aplikacja mobilna opracowana na rzecz tej pracy inżynierskiej.

Celem pracy jest stworzenie rozwiązania wspomagającego ucznia w nauce kanji jako aplikacji mobilnej. Program powinien posiadać skuteczny tryb nauczania, który potrafi dostosować się do poziomu ucznia. Jako że kluczowa dla zapamiętania tak dużej liczby znaków systematyczność jest trudna do osiągnięcia, aplikacja powinna motywować do kolejnych podejść i dostarczać różnorodne ćwiczenia. Ważnym zagadnieniem w rozwiązywaniu tego rodzaju jest też odpowiedni rozrzut pytań w czasie. Ta praca stara się wziąć pod uwagę fakt, że niektóre znaki mogą przyjść uczniowi łatwo, a inne trudno i należy odpowiednio dostosować algorytm dostarczający zadania do wykonania.

Rozwiązanie opisane w tej pracy to obok aplikacji mobilnej system wspomagający jej działanie. Na zakres pracy składa się sama aplikacja, baza danych dotyczących użytkowników oraz panel administracyjny jako aplikacja webowa. Rozwiązanie obsługuje trzy rodzaje użytkowników, którymi są uczeń, nauczyciel i administrator. Uczeń to użytkownik korzystający z takich funkcjonalności aplikacji jak przeglądanie słownika kanji, tworzenie zestawów znaków oraz tryb nauki. Aby ułatwić wybór znaków do nauki, aplikacja jest wyposażona w predefiniowane zestawy, które odpowiadają poziomom egzaminu językowego lub kolejnym klasom w japońskim szkolnictwie. Nauczyciel to użytkownik, który może tworzyć grupy uczniów, zadawać im zadania domowe i sprawdzać ich postępy. Takie rozwiązanie ma pomagać w nauce w trybie zajęć grupowych, gdzie nauczyciel spotyka się z uczniami kilka razy w miesiącu i chce motywować do dalszej pracy w domu.

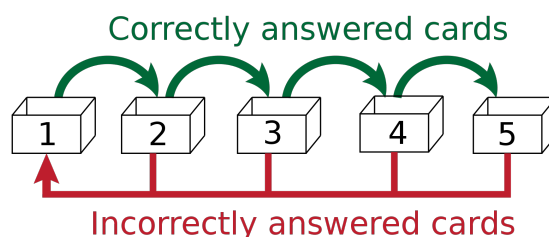
Rozdział 2

Analiza tematu

2.1 Nauka języków obcych wspomagana komputerowo

Próby usystematyzowania nauki, szczególnie gdy materiał wymaga od nas więcej zapamiętywania niż analizy problemów, są naturalnie występującym skutkiem konieczności żmudnego, wielokrotnego powtarzania. Uczniowie wykazujący się większym sprytem niż pracowitością szukają drogi na skróty, aby tą [fleksja] samą ilość materiału przyswoić w krótszym czasie. Tak powstały pierwsze metody technik zapamiętywania, na przykład fiszki, czyli niewielkie kawałki papieru z pytaniem na jednej stronie i odpowiedzią na drugiej. Taki system pozwala na kompaktowe przechowywanie materiału, co pozwala uczyć się również w krótkich chwilach czasu i daleko od domu, na przykład w autobusie.

Dziennikarz Sebastian Leitner w 1972 wymyślił sposób na dalsze usprawnienie nauki za pomocą fiszek [cytowanie]. W metodzie Leitnera, [bez przecinka] fiszki są podzielone na grupy względem tego, jak dobrze uczeń zna dany materiał. Jeżeli uczeń odpowie dobrze na dane pytanie, przenosi fiszkę do następnego pudełka. Jeśli odpowie źle, fiszka wraca na początek. Grupy różni to, jak często użytkownik jest zobowiązany powtarzać dany materiał. Im materiał lepiej znany, tym rzadziej uczeń musi go powtarzać. [Rys. 2.1: Praca jest po polsku, to i rysunek musi być po polsku.] [Rysunki, tabele i tym podobne elementy to w LaTeXu elementy pływające. Przy składzie tekstu LaTeX najpierw roz-



Rysunek 2.1: Ilustracja reprezentująca system wykorzystujący fiszki.

mieszcza je zgodnie z zasadami składu tekstów. Potem oblewa je tekstem. Tak się robi od czasów Gutenberga. Wstawianie elementów pływających na sztywno w konkretnym miejscu zwykle prowadzi do gorszego układu typograficznego. Mamy jako ludzkość bardzo duże doświadczenie (prawie pół tysiąclecia druku) i wiemy, co wygląda źle, a co wygląda dobrze. Wiemy, jaka powinna być długość wiersza, jakie powinno być wypełnienie strony tekstem (zaczerzenie). LaTeX implementuje bardzo wiele algorytmów do optymalizacji składu zarówno makrotypograficznych (akapit, strona), jak i mikrotypograficzny (litery w słowie). Pierwotnie opcji [h] w LaTeXu nie było, bo jest niepotrzebna i szkodliwa, ale jakiś nieszczęśnik to doimplementował na zgubę swoją i innych.]

Sposób Leitnera, mimo że skuteczny, jest żmudny do implementacji. Uczeń musi napisać ręcznie wszystkie fiszki, zrobić lub zakupić kilka pudełek o podobnym rozmiarze, rozkładać ten obszerny zestaw za każdym razem, gdy chce powtórzyć materiał. Tym samym traci mobilność, która jest dużą zaletą fiszek. W tym miejscu z pomocą przychodzi nauka języków wspomagana komputerowo (ang. Computer-Assisted Language Learning, skrót: CALL) [(ang. *Computer-Assisted Language Learning, CALL*)]. Jest to rozwijana [Skoro jest rozwijana, to musimy się podeprzeć jakimiś cytowaniami.] od lat pięćdziesiątych dwudziestego wieku dziedzina wykorzystująca komputery do poprawiania umiejętności językowych. Ten sposób podejścia do problemu zrzuca z barków uczniów część problemów z tworzeniem fiszek, a wraz z popularyzacją urządzeń takich jak smartfony czy tablety zwraca mobilność nauki.

Algorytm powtórek w interwałach (eng. *spaced repetition*), którego papierową implementacją jest wyżej wspomniany system Leitnera, został poddany wielokrotnym badaniom [cytowanie]. Wskazują one nie tylko na poprawę jakości zapamiętywania danego materiału, ale też na ogólną poprawę pamięci, szczególnie u osób chorujących na [chorobę] Alzheimera lub inne schorzenia związane z pogorszeniem umiejętności zapamiętywania[1]. Metoda *spaced repetition* [po polsku] jest szczególnie użyteczna w nauce słownictwa języków obcych. Jest to spowodowane faktem, że informacje do przyswojenia są liczne, lecz małe. Tę samą zasadę można wykorzystać przy nauce znaków kanji.

Analiza rozwiązań dostępnych na rynku i badań wskazuje na komercyjny i naukowy sukces metody *spaced repetition*. To zdroworozsądkowe podejście ma realny wpływ na szybkość i jakość przyswajania wiedzy. Przez ostatnie kilkadziesiąt lat próbowano udoskonalić ten względnie prosty algorytm z użyciem np. metod sztucznej inteligencji do wyliczania interwałów [cytowanie]. Takie podejście ma mieszane rezultaty według badań [cytowanie], nie jest jasne jak realny jest wpływ optymalizacji długości po pewnym marginesie osiągalnym przez konwencjonalne algorytmy [Co to znaczy: konwencjonalne algorytmy?].

2.2 Analiza rozwiązań dostępnych na rynku

Aktualnie nauczanie wspierane komputerowo jest szeroko rozpowszechnione [Skąd to wiemy? cytowanie]. Największą popularnością [cytowanie] cieszy się aplikacja Duolingo [odnośnik do strony], która umożliwia naukę pokazniej listy języków (aż czterdziestu trzech [czterdziestu trzech] na czas pisania tej pracy) z urządzenia mobilnego lub strony internetowej. Jednym z powodów dobrego przyjęcia Duolingo przez uczniów języków obcych jest podejście bogate w gry i zabawy, motywujące użytkownika przez punkty, nagrody i rankingi. Aplikacja nie wymaga od użytkownika dużego zaangażowania. Tworzenie własnych zestawów fiszek nie jest konieczne, wystarczy codziennie wykonać kilkanaście zadań wygenerowanych przez system.

Innym, bardziej angażującym użytkownika programem chętnie wykorzystywanym do nauki jest Anki [odnośnik do strony]. Jest to aplikacja oferująca większą możliwość dostosowania jej do własnych potrzeb, z tego powodu nie jest ograniczona do nauki języków obcych. Użytkownik ma możliwość stworzenia własnego zestawu do nauki złożonego z fiszek, który jest później przyswajany w interwałach zgodnie z zasadą *spaced repetition*. Zaletą Anki jest opcja eksportowania i dzielenia się zestawami między użytkownikami.

Wyżej wspomniane programy są często używane przez uczniów języka japońskiego i znaków kanji, jednak nie są wyspecjalizowane pod ich potrzeby. Przykładem aplikacji dostosowanej do amatorów japońskiego alfabetu jest WaniKani [odnośnik do strony]. Ta aplikacja webowa i mobilna gwarantuje [Chyba trzeba pisać trochę ostrożniej. Pewnie znajdzie się ktoś komu nie zagwarantuje.] uczniom nauczanie się 2000 znaków kanji i 6000 słów w nieco dłużej niż rok. Używa metody *spaced repetition* w swojej implementacji ustrukturyzowanego kursu, który jest przeznaczony dla początkujących i pozwala rozwinąć się do biegłego posługiwania się kanji. Aplikacja jest gorzej dostosowana do uczniów o zaawansowanym lub średniozaawansowanym poziomie, ponieważ nie da się dopasować jej pod własne potrzeby językowe.

2.3 Nauka języka japońskiego i znaków kanji

Ważną kwestią przy nauce kanji, zarówno metodami tradycyjnymi jak i z wykorzystaniem komputerowego wspomaganie, jest podejście do przyswajanego materiału. Najważniejszą i obowiązkową kwestią jest zapamiętanie wyglądu i pisowni znaku. Aby wspomóc ten proces, można użyć wskazówek mnemoniczych, czyli pewnego skojarzenia między znaczeniem znaku a jego wyglądem. Kolejną wskazówką do nauki mogą być klucze, czyli elementy znaku dzielące ich zbiór na podkategorie. Ten podział został stworzony na potrzeby słowników i charakteryzuje znaki na podstawie ich najważniejszego elementu (podznaku).

Wraz z opanowaniem pisowni należy zaznajomić się z czytaniem chińskim (jap. *onyōmi*)

i natywnym japońskim (jap. *kunyōmi*). Na nieszczęście uczniów j. [całym słowem] japońskiego znaki kanji nie są czytane jednoznacznie i większość z nich ma przynajmniej wyżej wspomniane dwa czytania (wiele z nich ma kilka w ramach *onyōmi* lub *kunyōmi*). Aby urozmaicić naukę czytania znaków kanji można połączyć ją z nauką słów, w których ów znak występuje. Takie podejście wzbogaca zasób słownictwa ucznia i ułatwia zapamiętanie czytań danego znaku przez różnorodne przykłady. Kolejnym możliwym sposobem na połączenie nauki znaków z nauką ogólnie rozumianego języka japońskiego jest załączenie ćwiczeń zawierających całe zdania. Umożliwi to użytkownikom o bardziej zaawansowanym poziomie powtarzać różne znaki w jednym ćwiczeniu oraz rozwinie ich ogólne umiejętności językowe.

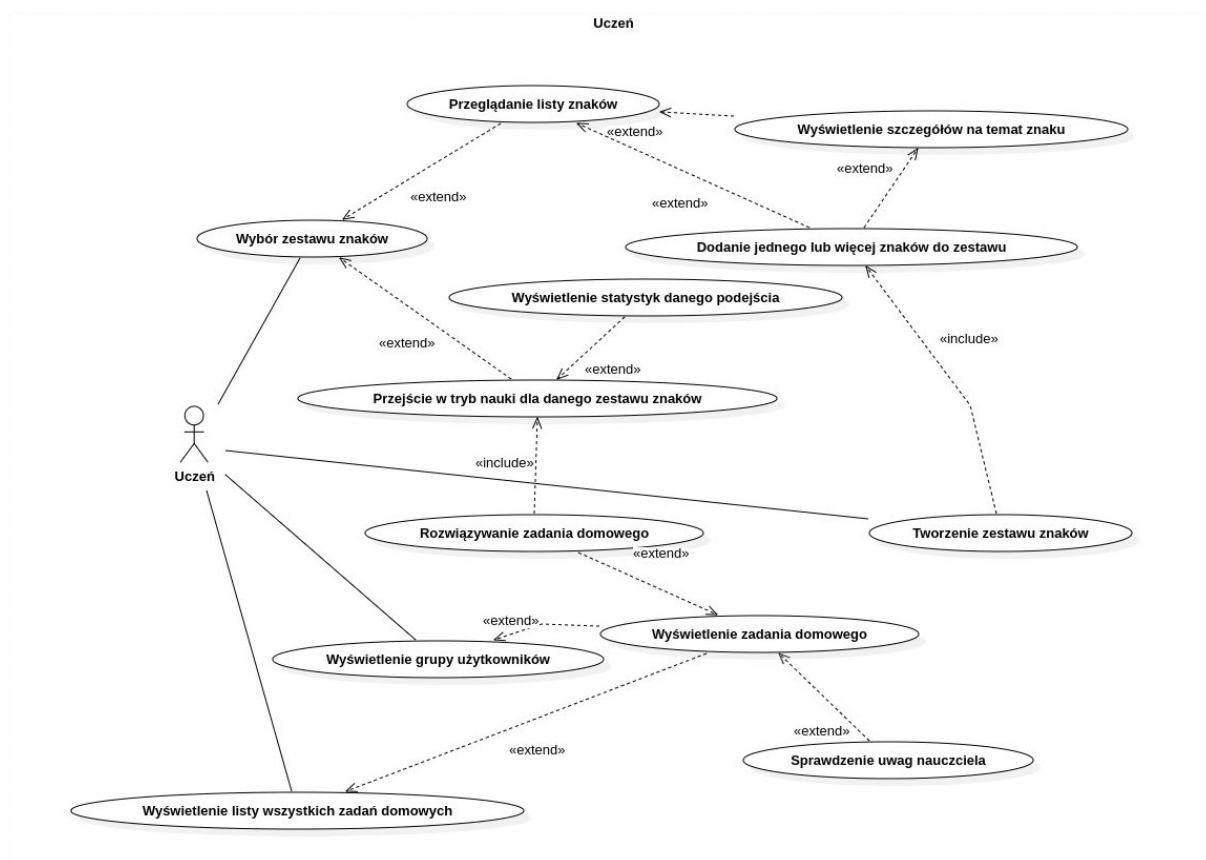
[Tutaj napisałbym kilka słów o poziomach, ile znaków jest na poziomie itd.]

Rozdział 3

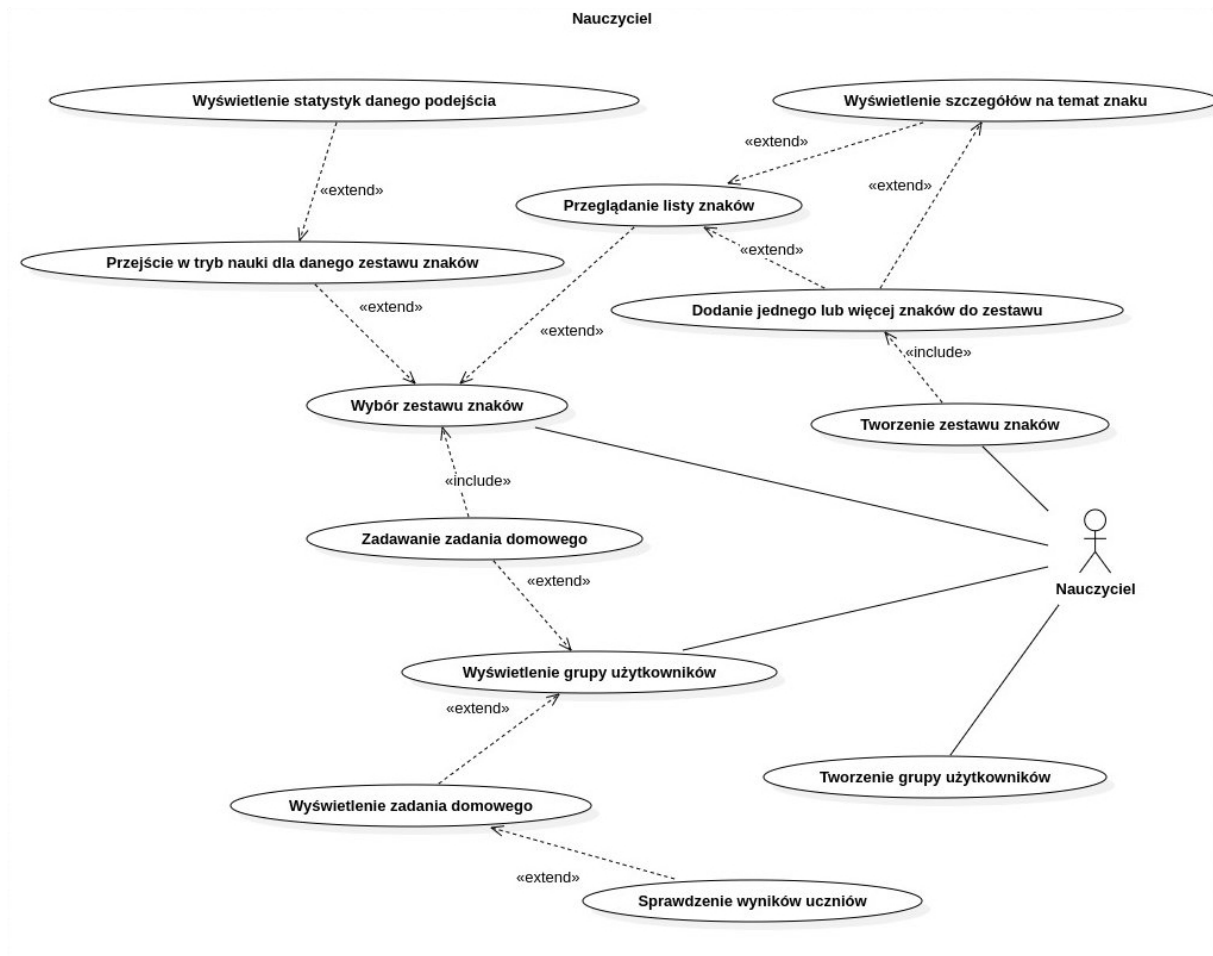
Wymagania i narzędzia

- 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

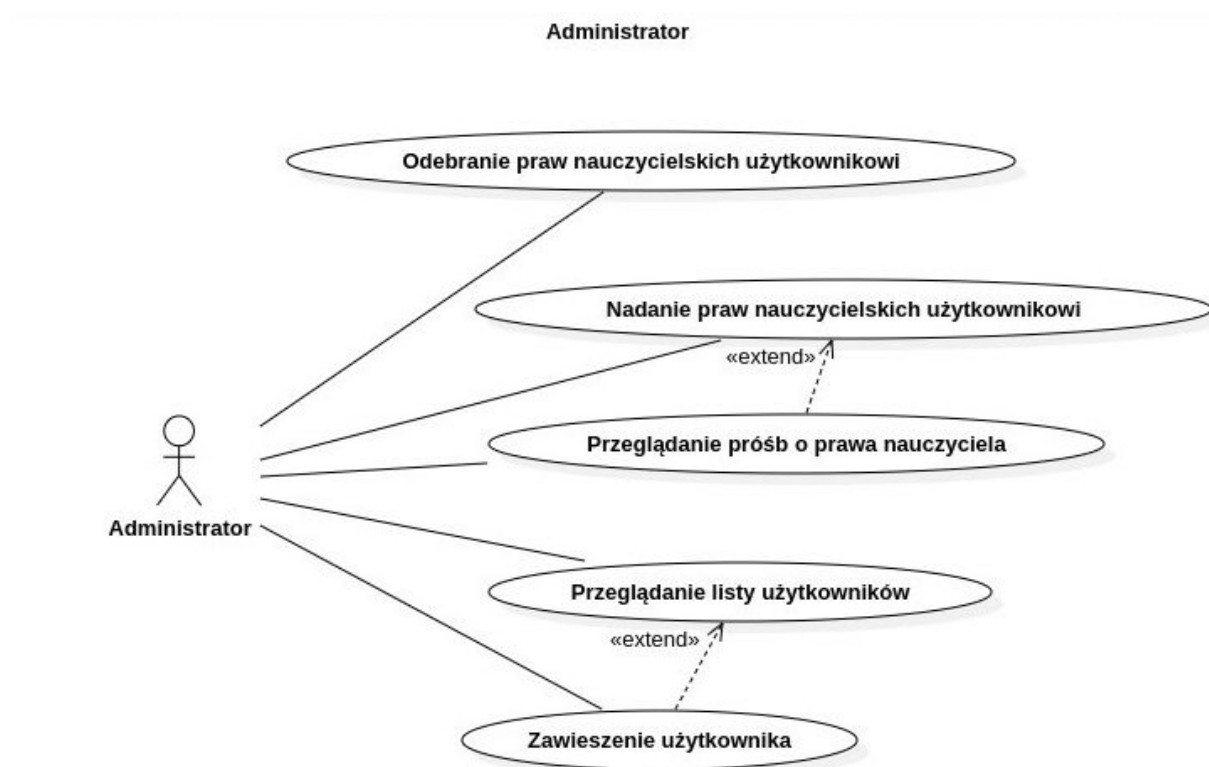
Przypadek użycia Priorytet



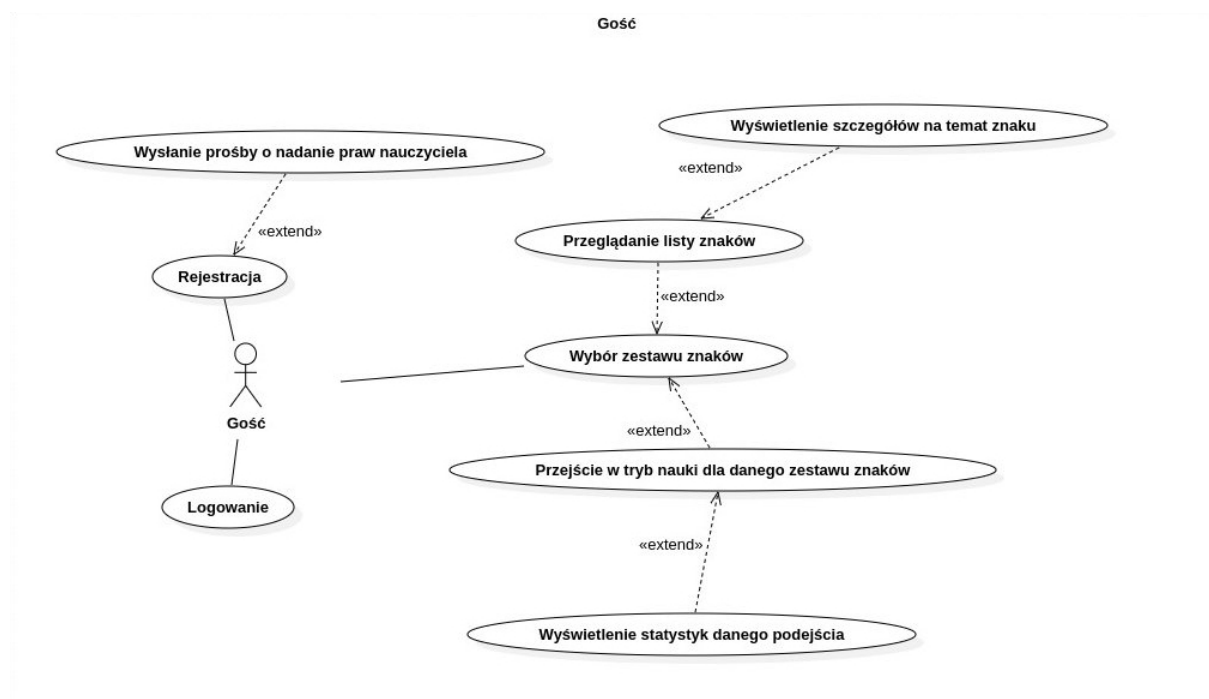
Rysunek 3.1: Podpis rysunku zawsze pod rysunkiem.



Rysunek 3.2: Podpis rysunku zawsze pod rysunkiem.



Rysunek 3.3: Podpis rysunku zawsze pod rysunkiem.



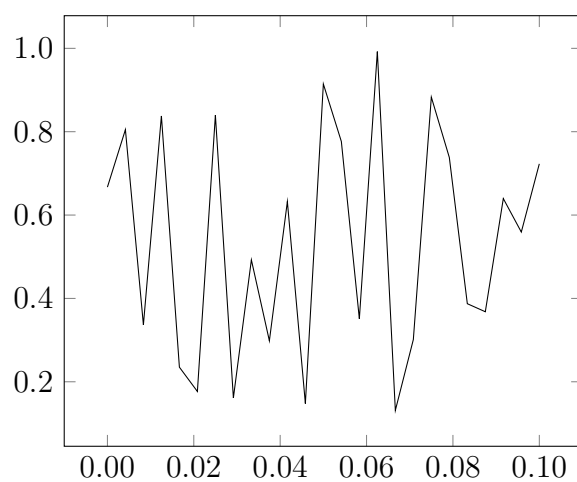
Rysunek 3.4: Podpis rysunku zawsze pod rysunkiem.

Rozdział 4

[Właściwy dla kierunku – np. 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

[Właściwy dla kierunku – np. 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 wstawka 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

Tablica 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] Erin M. Jackson Karri S. Hawley Emily O. Boudreaux. „A comparison of adjusted spaced retrieval versus a uniform expanded retrieval schedule for learning a name–face association in older adults with probable Alzheimer’s disease”. W: *Journal of Clinical and Experimental Neuropsychology* 30.6 (2008), s. 639–649.

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

2.1	Ilustracja reprezentująca system wykorzystujący fiszki.	3
3.1	Podpis rysunku zawsze pod rysunkiem.	8
3.2	Podpis rysunku zawsze pod rysunkiem.	9
3.3	Podpis rysunku zawsze pod rysunkiem.	10
3.4	Podpis rysunku zawsze pod rysunkiem.	10
4.1	Podpis rysunku po rysunkiem.	12
5.1	Pseudokod w <code>listings</code>	14

Spis tablic

6.1	Nagłówek tabeli jest nad tabelą.	16
-----	--	----