Uwagi

1: Wstępem zajmiemy się na końcu.

1: Może rōmaji tu jeszcze?

1: Angielskie słowa i frazy będziemy pisać w komendzie *english*. Ta komenda składa kursywą i (ważniejsze) przełącza LaTeX na angielską typografię, m. in. zasady dzielenia wyrazów itd.

3: Później zmienimy tytuł rozdziału na coś bardziej związanego z pracą niż dość ogólna «analiza tematu».

8: Trzeba napisać, jacy są autorzy.

12: np. czego? todo

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 slow (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

Rozdział 1

Wstęp

[Wstępem zajmiemy 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 romańskim 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. 日本語能力試験, [Może rōmaji tu jeszcze?] ang. *Japanese Language Proficiency Test* [Angielskie słowa i frazy będziemy pisać w komendzie *english*. Ta komenda składa kursywą i (ważniejsze) przełącza LaTeX na angielską typografię, m. in. zasady dzielenia wyrazów itd.]) 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 poskutkuje 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 zapamiętaniu tak dużej liczby znaków systematyczność jest trudna do osiągnięcia, aplikacja powinna motywować do kolejnych podejść i dostarczać różnorodne ćwiczenia. Ważnym zagadnieniem w rozwiązaniu 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

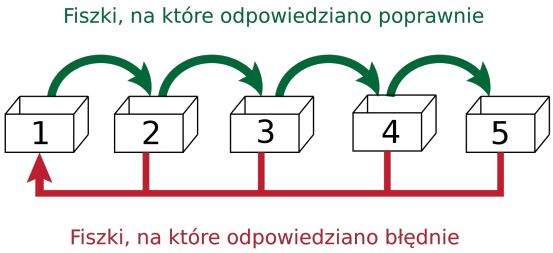
[Później zmienimy tytuł rozdziału na coś bardziej związanego z pracą niż dość ogólna «analiza tematu».]

2.1 Nauka języków obcych wspomagana komputerowo

Próby usystematyzowania nauki, szczególnie gdy materiał wymaga 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 tę 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 [20]. W metodzie Leitnera 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.

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*, CALL). Jest to rozwijana od lat pięćdziesiątych dwudziestego wieku dziedzina wykorzystująca komputery do poprawiania umiejętności językowych [12]. 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.



Rysunek 2.1: Ilustracja reprezentująca system wykorzystujący fiszki.

Algorytm powtórek w interwałach (ang. *spaced repetition*), którego papierową implementacją jest wyżej wspomniany system Leitnera, został poddany wielokrotnym badaniom [15, 16]. 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 [15]. Metoda powtórek w interwałach 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 powtórek w interwałach. 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 [14]. Takie podejście ma mieszane rezultaty, nie jest jasne jak realny jest wpływ optymalizacji długości interwału na jakość nauki [17, rozdział 6].

2.2 Analiza rozwiązań dostępnych na rynku

Aktualnie formą nauczania języków wspomaganą komputerowo cieszącą się szerokim zainteresowaniem jest jej poddziedzina, czyli nauczanie języków obcych wspomagane przez urządzenia mobilne (ang. *Mobile-Assisted Language Learning, MALL*) [19]. Największą popularnością [11, 19] cieszy się aplikacja Duolingo [1], która umożliwia naukę pokaźniej listy języków (aż 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 [19]. 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 [2]. Jest to aplikacja oferująca większą możliwość dostosowania jej do własnych

potrzeb, z tego powodu nie jest ograniczona do nauki języków obcych. Użytkownik ma możliwość stworzenie własnego zestawu do nauki złożonego z fiszek, który jest później przyswajany w interwałach zgodnie z zasadą powtórek w interwałach. 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 [3] [1]. Ta aplikacja webowa i mobilna obiecuje uczniom nauczenie się 2000 znaków kanji i 6000 słów w nieco dłużej niż rok. Używa metody powtórek w interwałach 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 wspomagania, jest podejście do przyswajanego materiału. Najważniejszą i obowiązkową kwestią jest zapamiętanie wyglądu i pisowni znaku. Aby wspomóc ten proces, można użyć wskazówek mnemonicznych, 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 najważniejszgo elementu (podznaku).

Wraz z opanowaniem pisowni należy zaznajomić się z czytaniem chińskim (jap. 音読み, *onyomi*) i rodzimym japońskim (jap. 訓読み, *kunyomi*). Na nieszczęście uczniów języka 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*). Na przykład można wziąć znak 人, który oznacza człowieka. Jeżeli chcemy powiedzieć poprostu *człowiek*, użyjemy czytania japońskiego - *hito*. Jeżeli zechcemy powiedzieć *Japończyk*, użyjemy czytania sinojapońskiego - 日本**人**, *Nihon**jin***. Jak można się zasugerować po powyższym przykładzie, rodzimego czytania często używa się tam, gdzie znak występuje w słowie samodzielnie, a sinojapońskiego tam, gdzie towarzyszą mu inne znaki. Jest to jednak tylko wskazówka, a nie zasada.

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

[1] `https://www.wanikani.com/`

poziomie powtarzać różne znaki w jednym ćwiczeniu oraz rozwinie ich ogólne umiejętności językowe.

Inną kwestią, często lekceważoną przez samouków, jest kolejność stawiania kresek przy pisaniu znaku. Ten sam znak może wyglądać zupełnie inaczej, jeżeli nieumiejętnie postawimy kolejne linie, z których się składa. Dlatego istotnym jest, żeby w swoją naukę wkomponować dużo pisania ręcznie, nawet jeśli w naszych czasach jest to coraz mniej popularne. Na wprost użytkownikom wychodzą tutaj też aplikacje, które umożliwiają ćwiczenie pisania przez na przykład rysowanie na ekranie. Uważnie wodząc palec lub długopis po ekranie według instrukcji można wyrobić dobre nawyki, które potem należy utrwalić na kartce.

Podobnie jak inne języki, japoński posiada swój własny system egzaminów przeznaczonych dla obcokrajowców. Egzaminy JLPT (jap. 日本語能力試験 ang. *Japanese Language Proficiency Test*) wydzielają pewne poziomy, które określają umiejętności językowe ucznia. W języku japońskim te poziomy oznacza się literą N oraz cyfrą od 1 do 5, gdzie N5 to najniższy, a N1 najbardziej biegły. Do zdania każdego z tych egzaminów potrzebny jest rosnący z poziomu na poziom zasób kanji. Zaczynając od niewielkich liczb jak około sto dla N5 i trzysta dla N4, kończymy na ponad dwu tysiącach dla pełnej biegłości w pisaniu i czytaniu według egzaminatorów.

Podział znaków kanji według trudności wygląda inaczej dla rodzimych mieszkańców Japonii. Jest on ułożony względem systemu szkolnictwa publicznego, podzielony na klasy w których uczniowie poznają dane znaki. Do zestawu dwu tysięcy znaków wymaganych w szkole, czyli 常用漢字, *jōyō kanji* dochodzi 846 znaków 人名用, *jinmeiyō*, które są dopuszczane w imionach i nazwiskach, mimo że nie należą do *jōyō kanji*.

Nauka znaków kanji to wieloetapowy proces. Aplikacje i inne pomoce naukowe przeznaczone do tego celu mogą być wysoko wyspecjalizowane, aby jak najbardziej ułatwić przyswajanie informacji.

Rozdział 3

Wymagania i narzędzia

3.1 Wymagania funkcjonalne i niefunkcjonalne

3.1.1 Wymagania funkcjonalne

Wymagania funkcjonalne zostały określone jako lista funkcjonalności z przypisanymi priorytetami. Najważniejsze są wymagania określone priorytetem Krytyczny, potem Ważny, następnie Niski, najmniejszą wagę mają te z oznacznikiem Opcjonalny.

3.1.2 Wymagania niefunkcjonalne

Wymagania niefunkcjonalne, w odróżeniu od funkcjonalnych, nie opisują konkretnych zachowań systemu a pewne cechy, które określają ogólne zachowanie systemu. Dla aplikacji stworzonej na potrzeby tej pracy wymagania niefunkcjonalne zostały określone w formie listy cech.

- Aplikacja posiada lokalną bazę danych ze znakami kanji, ich złożeniami i przykładowymi zdaniami nie przekraczającą 15MB
- Podstawowa funkcjonalność aplikacji jest dostępna offline
- System bezpiecznie przechowuje dane użytkowników w bazie danych hostowanej w chmurze
- Aplikacja jest skierowana do osób operujących językiem angielskim
- Aplikacja jest skierowana do użytkowników urządzeń mobilnych z systemem Android w wersji co najmniej 10.0

Tablica 3.1: Tabela przedstawiająca wymagania funkcjonalne zestawione z priorytetami.

przypadek użycia	priorytet
Użytkownik może przeglądać listę znaków kanji podzielonych na predefiniowane kategorie	Krytyczny
Użytkownik może uczyć się wybranego zestawu znaków według metody powtórek w interwałach, w której najtrudniejsze dla niego znaki będą się pojawiały w najkrótszych interwałach	Ważny
Użytkownik ma dostęp do statystyk danego podejścia	Niski
Użytkownik może zalogować się loginem i hasłem jako uczeń lub nauczyciel	Ważny
Użytkownik może się zarejestrować	Ważny
Użytkownik może poprosić przy rejestracji o uprawnienia nauczyciela	Niski
Zalogowany użytkownik może tworzyć własne zestawy znaków	Ważny
Nauczyciel może tworzyć grupy użytkowników, którzy uczą się razem	Ważny
Nauczyciel może zadać zadanie domowe wybranej grupie użytkowników	Ważny
Nauczyciel może sprawdzać postępy uczniów i komentować ich podejścia	Niski
Administrator ma dostęp do osobnego panelu administracyjnego, na którym może zarządzać uprawnieniami użytkowników	Ważny
Zalogowany użytkownik może dzielić się swoimi zestawami znaków za pomocą linku	Opcjonalny
W trakcie nauki uczeń może ćwiczyć pisanie rysując na telefonie	Niski

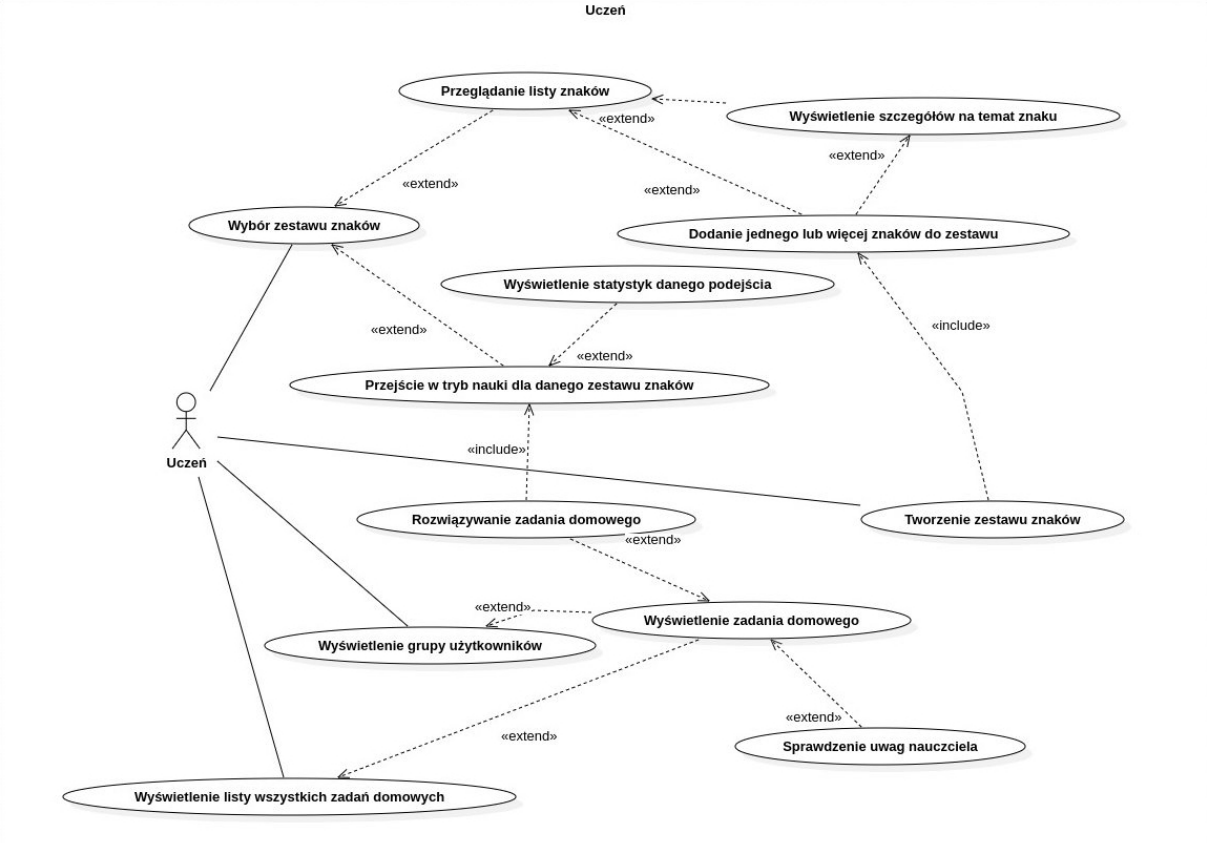
3.2 Diagramy przypadków użycia

Ze względu na dość dużą ilość przypadków użycia, zostały one zaprezentowane jako osobne diagramy UML dla każdego z aktorów. Gość to użytkownik niezalogowany. Uczeń do użytkownik zalogowany, a Nauczyciel ma dodatkowo nadane prawa nauczycielskie. Administrator to użytkownik specjalny, z dostępem do panelu administratora. [Trzeba napisać, jacy są autorzy.]

3.3 Opis narzędzi

3.3.1 Aplikacja mobilna

Aplikacja została napisana w języku programowania Java na platformę Android w wersji 10 lub wyższej. Programiści aplikacji przeznaczonych na Androida zwykle wybierają między językami Java i Kotlin (chociaż są interoperacyjne, więc można je łączyć). Kotlin jest językiem stworzonym przez firmę JetBrains, który działa na maszynie wirtualnej Javy. Język ten powstał między innymi aby wyeliminować niektóre problemy Javy, takie jak błędy odwołania (ang. *null pointer exceptions*). Kotlin w aplikacjach na Android ma rosnącą popularność. W 2017 został ogłoszony przez firmę Google jako oficjalny język oprogramowania na ten system [18], a w 2019 jako język preferowany [13]. Według dokumentacji ponad 60 procent profesjonalnych developerów w dziedzinie aplikacji mobilnych na Androida wybiera Kotlin [13].



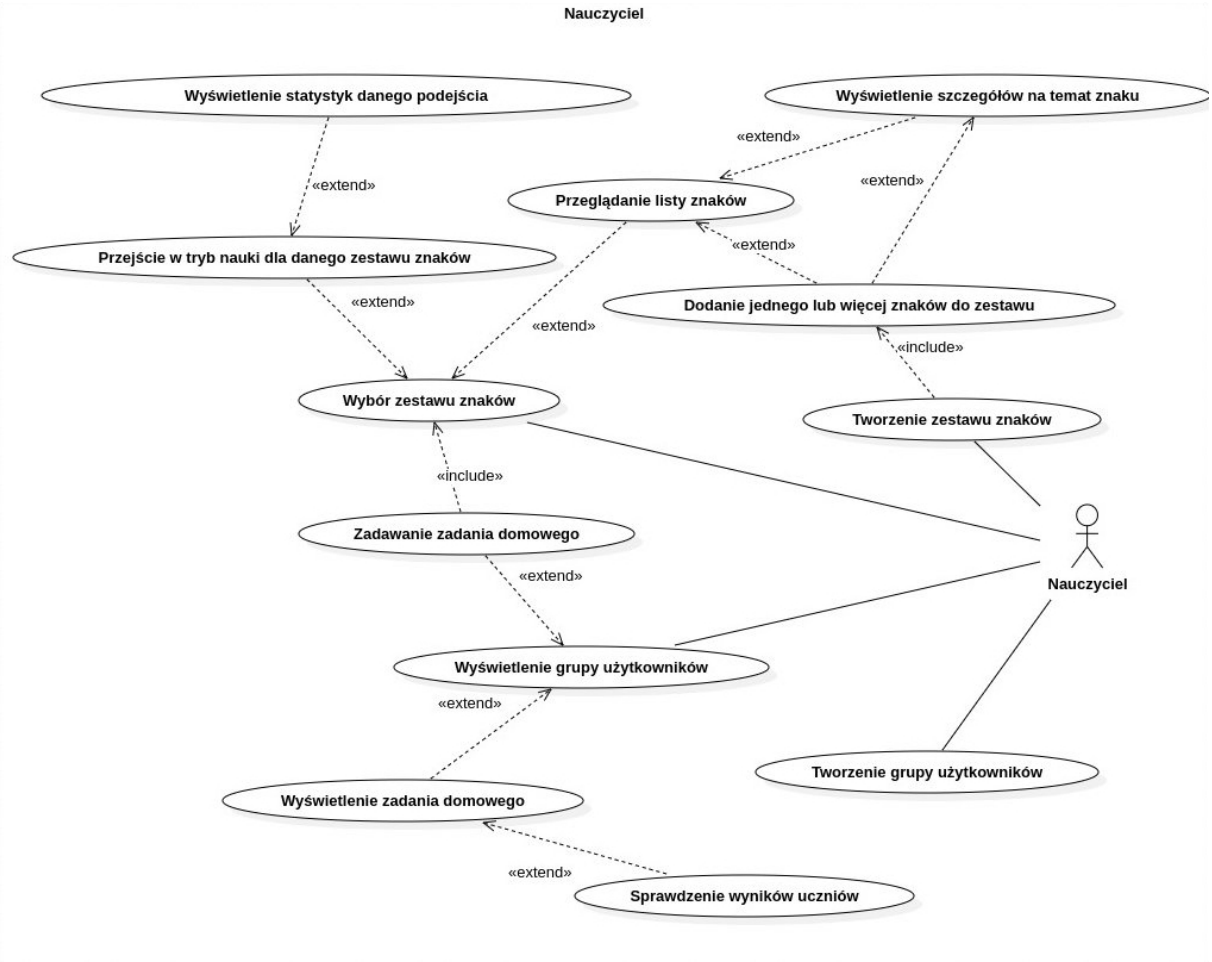
Rysunek 3.1: Diagram przypadków użycia dla Ucznia.

Mimo zalet tego nowocześniejszego języka, w aplikacji opisywanej w tej pracy postawiono na Javę. Jest to głównie spowodowane osobistymi pobudkami, takimi jak większa pewność w tym języku i sytuacja na rynku pracy. Dużą zaletą Kotlina i środowiska programistycznego Android Studio jest możliwość konwersji kodu w Javie na kod w Kotlinie. Ta cecha, wraz z interoperacyjnością tych dwóch języków, umożliwia dalszy rozwój aplikacji lub całkowitą migrację na Kotlina w przyszłości.

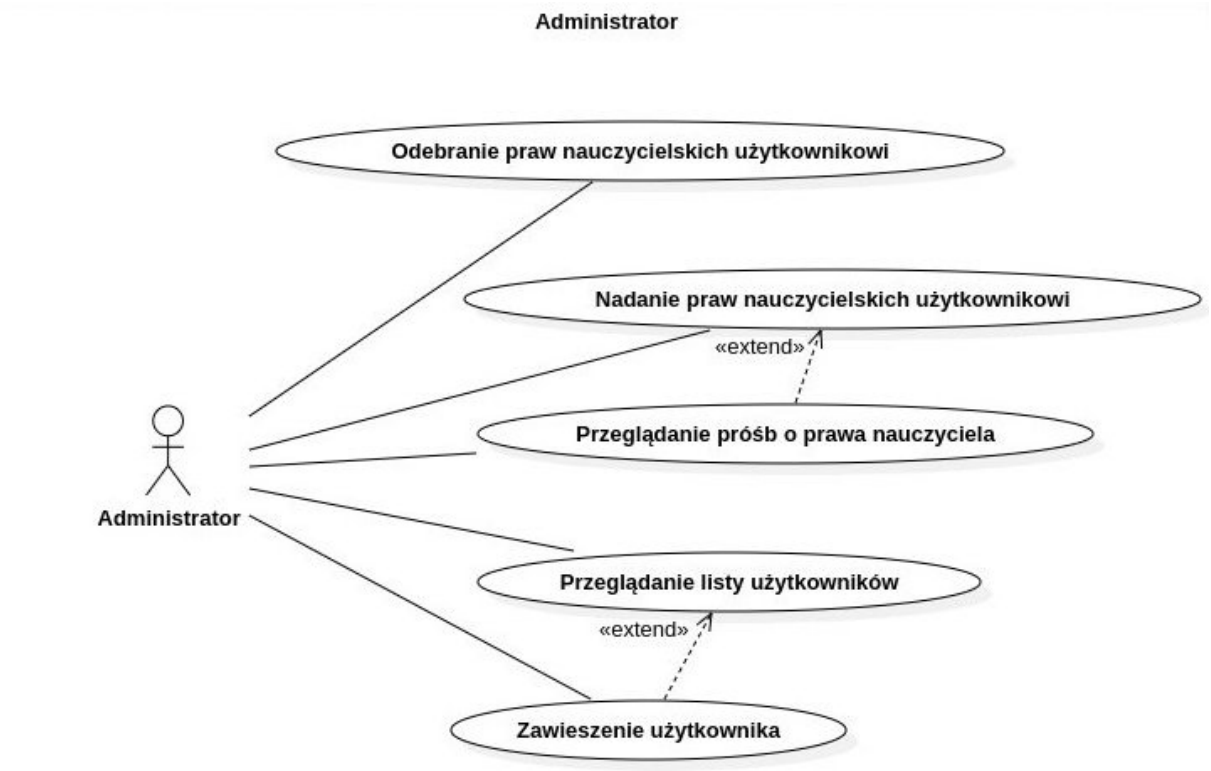
Przy tworzeniu aplikacji na platformę Android należy wybrać minimalną oraz docelową wersję zestawu narzędzi dla programistów (ang. *software development kit, SDK*). W aplikacji opisywanej w tej pracy minimalna wersja to 29 (Android 10), a docelowa 33 (Android 13). Wybór minimalnej wersji zestawu narzędzi jest uzasadniony tym, że 81,9 procent użytkowników tego systemu posiada minimalnie Androida 10 [10]. Docelowa wersja jest narzucona przez sklep Google Play.

Do tworzenia aplikacji użyto środowiska Android Studio Giraffe [4]. Android Studio to oficjalnie wspierane środowisko programistyczne dla developerów na platformę Android rozwijane przez firmę JetBrains. Zaprezentowane w 2013 roku, niedługo później zastąpiło wcześniej dominujące Eclipse. Android Studio oferuje bogaty wachlarz udogodnień dla programistów, takich jak:

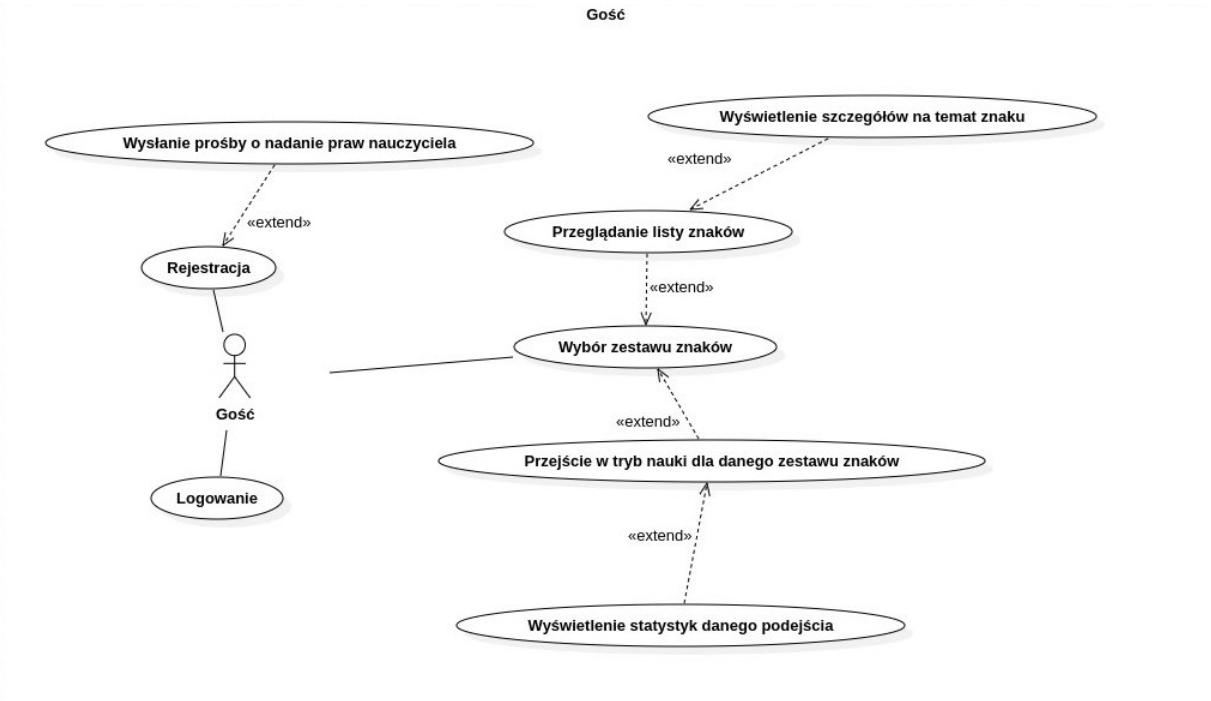
- sugestie poprawek kodu dostosowane do Androida,



Rysunek 3.2: Diagram przypadków użycia dla Nauczyciela.



Rysunek 3.3: Diagram przypadków użycia dla Administratora.



Rysunek 3.4: Diagram przypadków użycia dla Gościa.

- graficzny edytor wyglądu aplikacji z dynamicznym podglądem,
- wbudowany emulator urządzenia umożliwiający uruchomienie i debugowanie aplikacji.

3.3.2 Lokalna baza danych z informacjami dotyczącymi języka japońskiego

Do zrealizowania aplikacji opisywanej w tej pracy kluczowym elementem było znalezienie odpowiedniego źródła wiedzy na temat kanji, słów i zdań w języku japońskim. Aby zachować najważniejszą funkcjonalność aplikacji (przeglądanie znaków, uczenie się zestawów) w trybie offline, zdecydowano się na bazę danych przechowywaną lokalnie na urządzeniu każdego z użytkowników.

Stworzenie tej aplikacji i pracy inżynierskiej jest możliwe dzięki projektowi Kanjium [5]. Jest to darmowa baza danych złożona z kilku plików (EDICT, KANJIDIC, KRADFILE) należących do *Electronic Dictionary Research and Development Group (EDRDG)* [6]. Informacje na temat partykuł, fonetyki, homonimów, akcentu tonicznego i inne modyfikacje do wyżej wspomnianych plików są autorstwa Urosa Ozvatica. Na potrzeby aplikacji opisywanej w tej pracy baza została jedynie okrojona z pewnych niepotrzebnych części [np. czego? todo] , aby jej rozmiar umożliwiał lokalne przechowywanie na urządzeniach użytkowników. Baza danych Kanjium została wykorzystana w tej pracy zgodnie z licencjami grupy EDRDG oraz autora poprawek Urosa Ozvatica.

Ponieważ wyżej opisana baza danych zawiera pewne niezmieniające się informacje, jest dostępna dla aplikacji w trybie tylko do odczytu. W przypadku znaczących zmian w standardzie języka japońskiego wymuszających zmiany w bazie, należałoby ją zaktualizować w formie nowej wersji aplikacji.

3.3.3 Autoryzacja użytkowników i baza danych

Do zaimplementowania autoryzacji oraz bazy danych zawierającej informacje na temat zalogowanych użytkowników wykorzystano rozwiązanie Firebase [7]. Jest to komplet funkcjonalności oferowanych developerom aplikacji mobilnych, webowych, gier i innych. W aplikacji opisywanej w tej pracy z kompletu narzędzi wykorzystano mechanizm autoryzacji użytkowników i Firestore, czyli bazę danych NoSQL.

Autoryzacja użytkowników z użyciem Firebase dostarcza programistom łatwe w użyciu metody, które umożliwiają logowanie się użytkowników przez email i hasło lub konto na wybranych portalach społecznościowych. W celu dalszego ułatwienia implementacji Firebase oferuje gotowy interfejs użytkownika przeznaczony do logowania, jednak w tej pracy nie wykorzystano tej możliwości. Autoryzacja użytkownika oferowana przez Firebase została wybrana dla aplikacji opisywanej w tej pracy, ponieważ gotowe, bezpieczne rozwiązanie pozwoliło na poświęcenie cennego czasu na implementację bardziej kluczowych funkcjonalności. Warto zauważyć, że Firebase jest płatne od pewnych progów zużycia zasobów, jednak są one

praktycznie nieosiągalne dla aplikacji niedostępnej dla szerokiej publiki.

Firestore, czyli baza danych oferowana jako część zestawu narzędzi Firebase, to baza NoSQL zorientowana na dokumenty. Oznacza to, że dane są przechowywane w dokumentach podobnych do formatu JSON. Różnice między bazami danych SQL i NoSQL leżą w ustrukturyzowaniu przechowywanych danych. SQL było rozwijane w czasach, gdy przechowywanie danych było bardzo drogie. Z tego powodu SQL zwraca dużą uwagę na unikanie duplikacji rekordów. Nowocześniejsze rozwiązania NoSQL mogą zajmować więcej miejsca, jednak są wygodniejsze w użyciu dla programistów. Nie mają stale ustalonych kolumn i nie są relacyjne. Ponieważ istnieje wiele rodzajów baz danych NoSQL, należy uważnie wybrać dystrybucję w zależności od potrzeb. Dla aplikacji rozwijanej na potrzeby tej pracy wybrano bazę danych NoSQL zorientowaną na dokumenty, ponieważ jest to najpopularniejsza alternatywa dla klasycznych, tabularycznych baz danych. Dodatkowym atutem jest łatwość użytku z perspektywy programisty, szczególnie we wczesnych fazach rozwoju aplikacji, gdy często zmienia się pomysł na zapisywanie i wykorzystywanie danych.

3.3.4 Panel administratora

Panel administratora to aplikacja webowa umożliwiająca administratorowi systemu zarządzanie użytkownikami aplikacji mobilnej. Główną funkcjonalnością panelu administratora jest możliwość rozpatrywania wniosków o uprawnienia nauczyciela. Nauczyciel to użytkownik, który posiada dodatkowe przywileje zakładania grup i zadawania im zadań domowych. Użytkownik może poprosić o takie uprawnienia przy zakładaniu konta w aplikacji mobilnej, a administrator zaakceptować lub odrzucić prośbę. Oprócz tego panel administratora oferuje możliwość przeglądania listy użytkowników wraz z ich uprawnieniami i usuwania ich.

Panel administratora korzysta z tego samego systemu uwierzytelniania użytkowników co aplikacja mobilna. Dodatkowo, przy logowaniu sprawdza, czy użytkownik próbujący się zalogować ma uprawnienia administratora. Opcja dodawania użytkownika z uprawnieniami administratora nie jest przewidziana w panelu, ponieważ jest to narzędzie wewnętrzne i zakłada się, że osoba używająca konta administratora w razie problemów ma dostęp do bazy danych. Panel korzysta z tej samej instancji bazy danych NoSQL co aplikacja mobilna. Dzięki temu jest w stanie modyfikować dane użytkowników.

Jako framework ułatwiający budowę aplikacji webowej posłużył Angular [8] w wersji 17. Angular to otwartoźródłowa platforma służąca do tworzenia aplikacji webowych używająca języka programowania TypeScript i rozwijana przez Google. Ponieważ funkcjonalności panelu administratora są bardzo proste, wybór Angulara sprowadził się tylko do łatwości użytku związanej z wcześniejszą znajomością tej platformy. Dodatkowym atutem tego frameworku jest też wbudowana obsługa narzędzi Firebase, które są wykorzystywane w aplikacji. Do interfejsu użytkownika wykorzystano bibliotekę Bootstrap [9], która zawiera pewne zestawy szablonów CSS ułatwiających wykonanie estetycznie spójnej aplikacji.

Rozdział 4

[Właściwy dla kierunku – np. Specyfikacja zewnętrzna]

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

Krótka 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

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] URL: https://pl.duolingo.com/.

[2] URL: https://apps.ankiweb.net/.

[3] URL: https://www.wanikani.com/.

[4] URL: https://developer.android.com/studio.

[5] URL: https://github.com/mifunetoshiro/kanjium.

[6] URL: https://www.edrdg.org/edrdg/licence.html.

[7] URL: https://firebase.google.com/.

[8] URL: https://angular.io/.

[9] URL: https://getbootstrap.com/.

[10] Eugene Belinsky. *Android API Levels*. URL: https://apilevels.com/ (term. wiz. 19. 12. 2023).

[11] Laura Ceci. *www.statista.com*. 2023. URL: https://www.statista.com/statistics/1239522/top-language-learning-apps-downloads (term. wiz. 23. 10. 2023).

[12] Graham Davies. „Computer Assisted Language Learning: Where are we now and where are we going?" W: *E-Learning and Japanese Language Education: Pedagogy and Practice, Oxford Brookes University*. 2007, s. 5346–5349.

[13] Google? Android documentation? *Android' s Kotlin-first approach*. 2023. URL: https://developer.android.com/kotlin/first (term. wiz. 01. 11. 2023).

[14] Bartosz Dreger i Piotr Wozniak. *Implementing the repetition spacing neural network*. 1998. URL: https://web.archive.org/web/20220821061451/https://www.supermemo.com/en/archives1990-2015/english/ol/nn_train (term. wiz. 29. 10. 2023).

[15] Karri S. Hawley, Emily O. Boudreaux i Erin M. Jackson. „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.

[16] Jeffrey D. Karpicke i Henry L. Roediger III. „Expanding retrieval practice promotes short-term retention, but equally spaced retrieval enhances long-term retention". W: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 33.4 (2007), s. 704–719.

[17] James S. Nairne. *The Foundations of Remembering: Essays in Honor of Henry L. Roediger, III*. New York i Hove: Psychology Press, 2007. ISBN: 978-1-84169-446-7.

[18] Maxim Shafirov. *Kotlin on Android. Now official*. 2017. URL: https://blog.jetbrains.com/kotlin/2017/05/kotlin-on-android-now-official/ (term. wiz. 01. 11. 2023).

[19] Mitchell Shortt, Shantanu Tilak, Irina Kuznetcova, Bethany Martens i Babatunde Akinkuolie. „Gamification in mobile-assisted language learning: a systematic review of Duolingo literature from public release of 2012 to early 2020". W: *Computer Assisted Language Learning* 36.3 (2023), s. 517–554.

[20] Sander Tamm. *e-student.org*. 2023. URL: https://e-student.org/leitner-system/ (term. wiz. 30. 10. 2023).

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

Spis tablic