

**UNIwersYTET WARMIŃSKO MAZURSKI W  
OLSZTYNIE  
WYDZIAŁ MATEMATYKI I INFORMATYKI**

**Adrian Bachmura  
Informatyka**

**Projekt i implementacja aplikacji webowej do nauki znaków kanji.**

**Praca inżynierska  
wykonana w Katedrze Metod  
Matematycznych Informatyki  
pod kierunkiem dr Pawła Drozdy**

**Olsztyn 2026**

## Spis Treści

Streszczenie pracy .....	3
Abstract.....	4
1. Wstęp.....	5
1.1 Struktura pisma japońskiego.....	5
1.2 Metody wspomaganie nauki kanji .....	5
1.3 Istniejące rozwiązania.....	7
1.4 Cel pracy .....	8
2. Projekt systemu .....	9
2.1 Wymagania funkcjonalne i нефункционалне .....	9
2.2 Diagram przypadków użycia .....	10
2.3 Scenariusze przypadków użycia .....	11
2.4 Baza danych.....	16
3. Implementacja .....	20
3.1 Środowisko i narzędzia programistyczne .....	20
3.2 Użyte biblioteki i frameworki.....	20
3.3 Czysta architektura .....	20
3.4 Importowanie danych .....	23
3.5 Algorytm SRS.....	23
4. Interfejs użytkownika .....	25
4.1 Widok rejestracji.....	25
4.2 Widok logowania.....	26
4.3 Strona główna .....	27
4.4 Widok listy znaków kanji .....	28
4.5 Widok detali znaku kanji .....	29
4.6 Widok powtórek kanji .....	31
4.7 Ustawienia .....	34
5. Konteneryzacja aplikacji oraz możliwości dalszego rozwoju.....	35
5.1 Konteneryzacja aplikacji.....	35
5.2 Instrukcja uruchamiania aplikacji na własnej maszynie.....	36
5.3 Plany rozwoju aplikacji .....	36
6. Podsumowanie.....	38
Bibliografia.....	39
Spis ilustracji .....	40
Spis tabel .....	41

## Streszczenie pracy

W latach 2020-2025 znacząco wzrosło na świecie zainteresowanie Japonią. Przyczyn tego zjawiska można szukać w pandemii COVID-19, podczas której rozrywka stała się jeszcze istotniejszą częścią życia ludzi niż dotychczas, popkultura japońska w szczególności na tym skorzystała. Osłabienie jena japońskiego i generalnie tani koszt utrzymania także w znacznym stopniu przyczyniły się do przyciągania rekordowej ilości turystów z całego świata. Wraz z zainteresowaniem tym krajem wzrosło także zainteresowanie językiem japońskim, który znany jest z trudności w nauce. Jedną z największych barier dla osób próbujących go poznać jest pismo, a przede wszystkim znaki logograficzne znane jako kanji.

Ten projekt ma na celu kierowanie i wspomaganie procesu nauki właśnie tych znaków. Dotychczasowe podobne rozwiązania cierpią na niewystarczającą ilość danych pokazanych w słowniku albo nie stosują systemu powtórek rozłożonych w czasie (SRS). Powoduje to wymóg używania dwóch oddzielnych aplikacji do tych celów. Stworzony system próbuje ten problem rozwiązać poprzez połączenie systemu SRS z danymi z powszechnie używanego słownika JMdict i pliku z informacjami o kanji KANJIDIC tworzonych i utrzymywanych przez The Electronic Dictionary Research and Development Group (EDRDG) z systemem SRS.

W tej pracy przedstawiono implementację działającej fullstackowej aplikacji webowej, która ma za zadanie osiągnięcie tego celu. Dane zaimportowane ze słowników oraz dotyczące użytkowników zapisywane są w bazie danych MSSQL. Frontend strony został zrealizowany w TypeScript i używając biblioteki React, a komunikacją między warstwą frontend a bazą operuje backend w postaci WebApi napisanego w .NET. Szczególny nacisk położono na architekturę systemu – zastosowano czystą architekturę, która poprzez separację logiki biznesowej od infrastruktury ułatwia przyszłą rozbudowę o dodatkowe platformy i funkcjonalności.

## **Abstract**

Between 2020 and 2025, global interest in Japan has increased significantly. The reasons for this phenomenon can be found in the COVID-19 pandemic, during which entertainment became an even more important part of people's life than before, with Japanese pop culture especially benefitting. The weakening of the Japanese yen and the generally low cost of living also contributed significantly to attracting record numbers of tourists from all around the world. Along with interest in the country, interest in the Japanese language, which is known for being difficult to learn, has also grown. One of the biggest barriers for people trying to learn it is the writing system, especially the logographic characters known as kanji.

This project aims to guide and aid in learning these Japanese characters. Currently available similar solutions suffer from insufficient data shown in the dictionary or do not use a spaced repetition system (SRS). This forces the use of two separate applications for these purposes. The developed system tries to solve this problem by combining an SRS system with data from the widely used JMdict dictionary and KANJIDIC kanji information file created and maintained by The Electronic Dictionary Research and Development Group (EDRDG).

This paper presents the implementation of a working full-stack web application designed to achieve this goal. Data imported from dictionaries and user data are stored in an MSSQL database. The frontend of the website was implemented in TypeScript using the React library, and communication between the frontend layer and the database is managed by a backend in the form of WebApi written on .NET. Special emphasis was placed on the system architecture – clean architecture was used, which, by separating business logic from infrastructure, eases future expansion with other platforms and features.

# 1. Wstęp

## 1.1 Struktura pisma japońskiego

Język japoński stosuje jeden z najbardziej złożonych systemów pisma na świecie. Składają się na niego trzy główne elementy - dwa sylabariusze fonograficzne i zbiór znaków logograficznych.

Hiragana (jap. ひらがな) i katakana (jap. カタカナ) są fonograficznymi systemami pisma składającymi się z 46 bazowych znaków, które reprezentują sylaby. Powstały one setki lat po tym, jak Japończycy zapożyczyli kanji z Chin, aby ułatwić wyrażanie japońskiej gramatyki oraz dźwięków. Hiragana służy przede wszystkim do zapisu elementów gramatycznych oraz słów, które mają pochodzenie japońskie, a nie chińskie. Natomiast katakana jest stosowana do słów pochodzących z innych języków, takich jak angielski czy niemiecki oraz nazw własnych. Te dwa sylabariusze są istotne w kontekście tego projektu, gdyż ich znajomość jest potrzebna, żeby móc przeczytać fonetyczne zapisy kanji. Są one także zazwyczaj pierwszym krokiem w nauce tego języka i można je opanować w stosunkowo krótkim czasie, gdyż konceptualnie w pewnym stopniu przypominają alfabet łaciński.

Natomiast Kanji (jap. 漢字) są znakami logograficznymi, co znaczy, że nie reprezentują one wyłącznie dźwięku, ale także całe słowa lub idee. Jest ich bardzo wiele, często podawaną ilością w źródłach jest 50 000, ale znaczna większość nie jest używana w praktyce w dzisiejszej Japonii. Japońskie Ministerstwo Edukacji oficjalnie uznaje listę 2136 *jōyō* kanji, które są stosowane w codziennym życiu w Japonii. Dla tekstów akademickich liczba potrzebnych znaków jest wyższa, ale nie istnieje konkretny ich spis, co utrudnia stwierdzenie jakie znaki są bardziej użyteczne od pewnego momentu.

Złożoność tych znaków stanowi wysoką barierę wejścia w naukę tego języka. Są jednak metody, które ten proces ułatwiają.

## 1.2 Metody wspomagania nauki kanji

Jednym sposobem nauki który jest skuteczny przy nauce znaków kanji jest stosowanie technik mnemonicznych. Polega to na tworzeniu wizualnych skojarzeń, aby ułatwić zapamiętywanie. Metodę tą spopularyzował James Heisig w jego serii książek pełniących istotną rolę w środowisku nauki japońskiego – *Remembering the Kanji*. Metoda autora polega na rozkład złożonych znaków kanji na prostsze elementy składniowe które nazwał prymitywami. Choć w dużym stopniu bazują one na oficjalnie uznawanych w Japonii radykałach (部首, bushu), Heisig wprowadził dodatkowe elementy lepiej przystosowane do tworzenia mnemonik. Autor przedstawia łączenie znaczeń tych prymitywów w historie, które

są bardziej przyswajalne dla osób niemających doświadczenia z podobnym systemem pisma niż obco wyglądające symbole. Przykładem tego może być połączenie znaków 人(człowiek) i 木(drzewo) w 休(odpoczynek) przez wyobrażenie osoby odpoczywającej w cieniu drzewa. W podobny sposób można nauczyć się tysięcy tych znaków.

Jest to podejście uznawane za lepsze od metody mechanicznego zapamiętywania, czyli wielokrotnego przepisywania znaku. Tworzenie takich historii angażuje więcej obszarów naszej pamięci, ułatwia to też rozróżnianie podobnie wyglądających znaków. Pisanie wciąż ma jednak wartość, gdyż angażuje ono inne części pamięci, oraz jest wciąż okazjonalnie przydatną umiejętnością nawet w erze digitalizacji. Warto więc obie metody połączyć. Jednak niezależnie od tego, jaką metodę wybierzemy, każda wymaga wielokrotnego powtarzania uczonego materiału, żeby go nie zapomnieć. W tym celu warto zastosować system powtórek rozłożonych w czasie.

Krzywa zapominania to wykładniczy spadek retencji informacji w czasie odkryty w XIX wieku przez Hermanna Ebbinghaus. Obserwacje tego zjawiska spowodowały znaczny postęp w metodykach nauczania, czego jednym z efektów było powstanie koncepcji systemu powtórek rozłożonych w czasie (Spaced Repetition System, SRS). Polega on na powtarzaniu materiału w momentach przed spodziewanym zapomnieniem informacji. Interwały rosną progresywnie, żeby zmaksymalizować retencję przy jak najmniejszym nakładzie czasu nauki.

Systemy SRS wykorzystują fakt, że informacje uczone na przestrzeni wielu dni rozłożonych w czasie są łatwiej przyswajalne niż pojedyncza intensywna sesja. Nadaje się to szczególnie dla nauki języków obcych, ponieważ musimy zapamiętać wiele krótkich informacji. Zastosowanie SRS pozwala na systematyczną naukę przy jednoczesnym utrzymaniu wcześniej poznanych znaków w pamięci długotrwałej.

Jednym ze standardowych algorytmów wykorzystywanych w systemach SRS jest SM-2 (SuperMemo 2). Został on opracowany w 1988 roku przez Piotra Woźniaka, aby obliczać optymalne interwały powtórek dla użytkownika na podstawie tego jak dobrze odpowiadał na dane zagadnienie poprzednio. Użytkownik ocenia własne odpowiedzi w skali od 0 (kompletne zapomnienie, zazwyczaj wymagają powtórki w tej samej sesji) do 5 (łatwe). Wartości te wpływają następnie na interwał przed kolejnym pojawieniem się tego zagadnienia, a także na tzw. *Easiness factor*, który ma także wpływ na dalsze sesje.

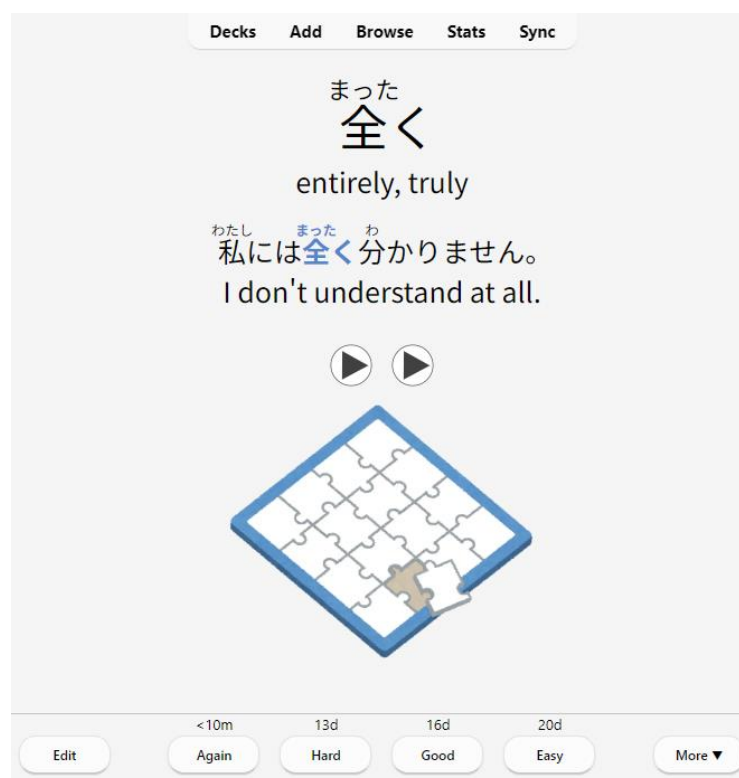
Algorytm SM-2 pozostaje standardem w aplikacjach edukacyjnych ze względu na prostą implementację oraz wysoką efektywność. Pomimo istnienia nowszych wersji tego algorytmu, takich jak SM-18, są to produkty komercyjne opracowane na potrzebę aplikacji

SuperMemo i nie są publicznie dostępne, co uniemożliwia ich wykorzystanie we własnych projektach. Z tego powodu wiele popularnych darmowych aplikacji edukacyjnych, takich jak Anki, stosuje algorytmy wzorowane na SM-2. W tym projekcie także przyjęto takie podejście ze względu na wspomniane zalety tego algorytmu.

### 1.3 Istniejące rozwiązania

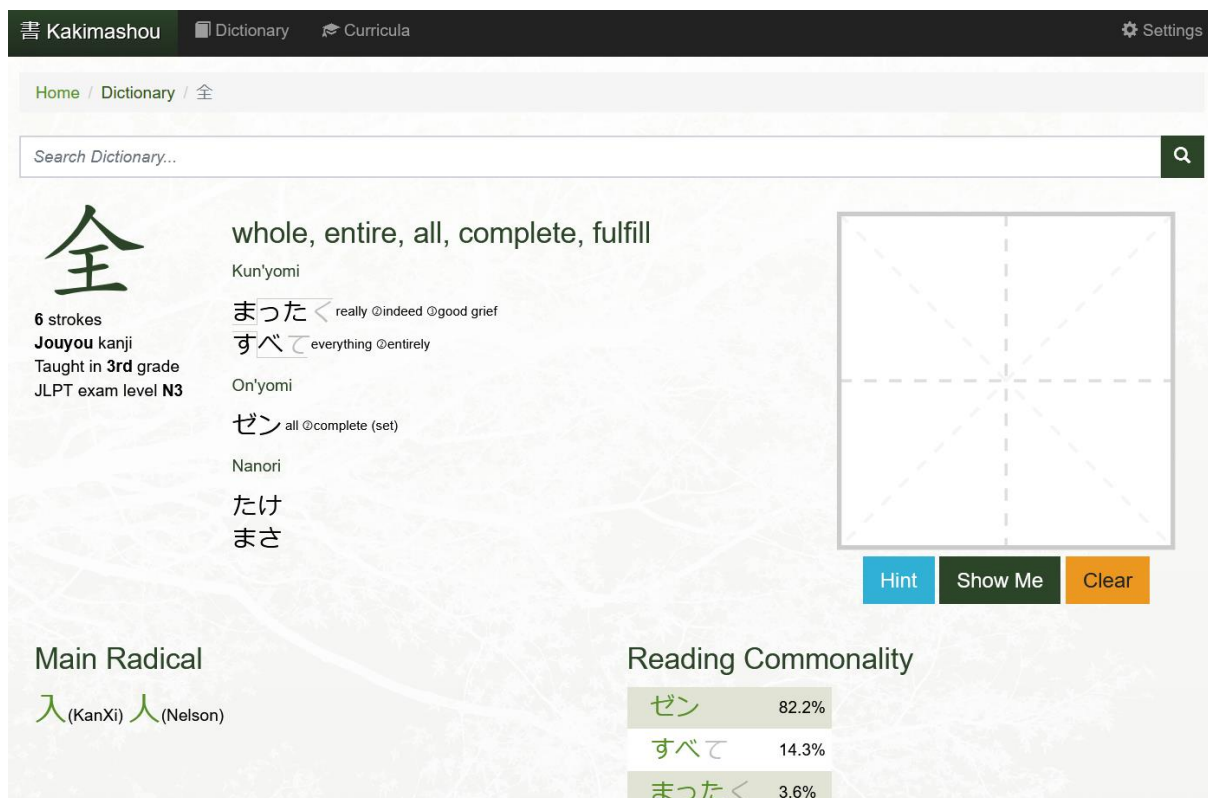
Oczywiście istnieje już wiele rozwiązań w dziedzinie nauki języka japońskiego, ale raczej dzielą się na słowniki i aplikacje do SRS, popularne rozwiązania nie oferują obu w jednej aplikacji. Dla części użytkowników nie jest to problem, ale dla osoby, która nie lubi ciągle nawigować między wieloma oknami jest to niedogodność.

Anki jest bardzo popularnym darmowym narzędziem SRS pozwalającym na dużą personalizację procesu uczenia się. Jednak natura tej aplikacji nie pozwala na tworzenie rozległego słownika dla kanji, a sam proces przygotowania i dostosowania samej aplikacji do działania jest dla wielu użytkowników skomplikowany i nieprzystępny.



Rysunek 1.1 - Interfejs aplikacji "Anki"

Kakimashou to dobry i naprawdę obszerny słownik japońsko-angielski z wieloma przydatnymi funkcjami, jak wspomagane rysowanie kanji na ekranie, rozkład elementów składniowych kanji, czy quizy skupione na konkretnych grupach kanji. Jednak nie oferuje on żadnego mechanizmu SRS, przez co nie nadaje się jako jedyne, długoterminowe narzędzie do nauki. Świetnie sprawdza się jednak jako aplikacja uzupełniająca inny system.



Rysunek 1.2 - Interfejs strony internetowej "Kakimashou"

## 1.4 Cel pracy

Celem pracy jest opracowanie i implementacja aplikacji webowej służącej do wspomagania nauki znaków kanji. Aplikacja ma być prosta w obsłudze, responsywna i zawierać obszerny słownik. Użytkownicy mają być w stanie dodawać do kanji notatki, edytować słowo-klucz dla fiszek oraz otrzymywać codziennie sesje powtórek z dotychczas poznanych znaków w systemie SRS.

Zakres pracy obejmuje zbudowanie fullstackowej aplikacji spełniającej te wymagania. Backend, opracowany w języku C# z użyciem technologii .NET, ma być zgodny z zasadami czystej architektury (Clean Architecture), a frontend napisany w TypeScript przy użyciu biblioteki React i narzędzia Vite ma oferować użytkownikom współczesny i przyjazny interfejs.



## **2. Projekt systemu**

### **2.1 Wymagania funkcjonalne i нефункционалне**

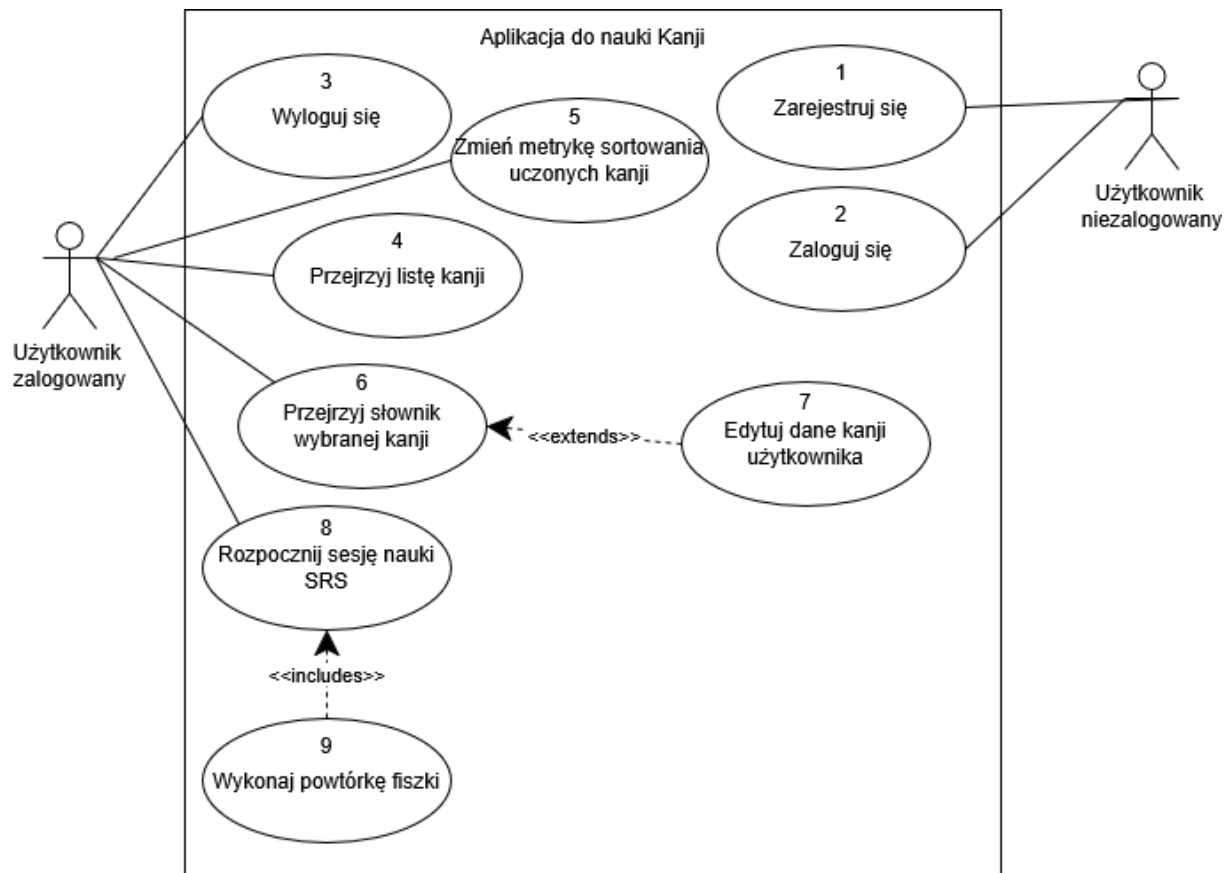
Wymagania funkcjonalne:

- Tworzenie własnego konta w aplikacji
- Logowanie się na własne konto w aplikacji
- Wylogowanie z aplikacji
- Przeglądanie listy kanji
- Sortowanie listy kanji według wybranych metryk
- Wyświetlenie szczegółowych informacji o kanji ze słownikiem
- Edycja słowa-klucza kanji używanego w fiszkach
- Dodawanie, edycja i usuwanie własnej notatki do kanji
- Stała zmiana sortowania uczonych kanji według wybranych metryk
- Dodanie wybranego kanji do aktywnej listy uczonych
- Wyświetlanie na stronie głównej informacji o liczbie dzisiejszych powtórek
- Przejście ze strony głównej do najwcześniejszego kanji spoza listy uczonych
- Uruchomienie sesji powtórek
- Wybór poziomu zapamiętania dla danego znaku kanji w trakcie sesji powtórek
- Odwracanie fiszki w celu wyświetlenia znaku kanji i notatki użytkownika
- Możliwość wczesnego zakończenia sesji powtórek i zapisania postępów
- Wsparcie dla ciemnego motywu interfejsu

Wymagania нефункционалне:

- Responsywność interfejsu
- Kompatybilna ze współczesnymi przeglądarkami internetowymi
- Łatwość utrzymania i rozbudowy systemu, wynikająca z zastosowania czystej architektury

## 2.2 Diagram przypadków użycia



Rysunek 2.1 - Diagram przypadków użycia

## 2.3 Scenariusze przypadków użycia

Lp.	PU-1
Nazwa:	Zarejestruj się
Aktor:	Użytkownik niezalogowany
Warunki początkowe:	Aktor wszedł na stronę aplikacji
Opis:	Aktor ma zamiar założyć konto w aplikacji
Ścieżka główna:	<ol style="list-style-type: none"><li>1. System wyświetla formularz rejestracji</li><li>2. Aktor wprowadza: nazwę użytkownika, email, hasło i powtórzenie hasła.</li><li>3. Aktor zatwierdza wpisane przez siebie dane poprzez kliknięcie przycisku „Sign up”</li><li>4. System waliduje dane otrzymane od aktora.</li><li>5. Walidacja jest pomyślna, system tworzy konto użytkownika</li><li>6. System wyświetla komunikat potwierdzający rejestrację aktora.</li><li>7. System przekierowuje aktora do formularza logowania.</li></ol>
Ścieżka alternatywna:	<ol style="list-style-type: none"><li>4a. Wpisane dane nie spełniają wymagań, wyświetlony jest odpowiedni komunikat.</li><li>4b. Hasła nie są identyczne, system wyświetla odpowiedni komunikat</li><li>4c. Email lub nazwa użytkownika znajdują się już w bazie danych, system wyświetla odpowiedni komunikat.</li></ol>
Warunki końcowe:	Rejestracja przeszła pomyślnie i aktor posiada nowopowstałe konto.

Tabela 2.1 PU-1 - Zarejestruj się

Lp.	PU-2
Nazwa:	Zaloguj się
Aktor:	Użytkownik niezalogowany
Warunki początkowe:	Aktor wszedł na stronę aplikacji
Opis:	Aktor chce się zalogować na swoje konto
Ścieżka główna:	<ol style="list-style-type: none"> <li>1. Aktor klika na przycisk „Log in” w nagłówku strony lub właśnie zakończył rejestrację.</li> <li>2. System wyświetla formularz logowania</li> <li>3. Aktor wprowadza: email lub login, oraz hasło</li> <li>4. Aktor zatwierdza wpisane przez siebie dane poprzez kliknięcie przycisku „Sign up” w formularzu.</li> <li>5. System waliduje dane otrzymane od aktora</li> <li>6. Walidacja jest pomyślna, system przekierowuje aktora do strony głównej.</li> </ol>
Ścieżka alternatywna:	5a. Dane nie są zgodne, aktor wraca do punktu 3.
Warunki końcowe:	Aktor został pomyślnie zalogowany

*Tabela 2.2 PU-2 - Zaloguj się*

Lp.	PU-3
Nazwa:	Wyloguj się
Aktor:	Użytkownik zalogowany
Warunki początkowe:	Aktor jest zalogowany na swoje konto
Opis:	Aktor chce wylogować się ze swojego konta
Ścieżka główna:	<ol style="list-style-type: none"> <li>1. Aktor klika na przycisk „Logout.</li> <li>2. System wylogowuje aktora</li> </ol>
Ścieżka alternatywna:	Brak
Warunki końcowe:	Aktor został pomyślnie wylogowany

*Tabela 2.3 PU-3 - Wyloguj się*

Lp.	PU-4
Nazwa:	Przejrzyj listę kanji
Aktor:	Użytkownik Zalogowany
Warunki początkowe:	Aktor jest zalogowany na swoje konto
Opis:	Aktor chce przeglądać listę wszystkich kanji oraz sortować je według dostępnych kryteriów.
Ścieżka główna:	1. Aktor klika na przycisk „Study” w nagłówku. 2. System wyświetla listę wszystkich kanji, sortowanych według metryki zmienianej w ustawieniach lub domyślnej „Heisig” 3. Aktor może zmienić metrykę sortowania lub nawigować na kolejną lub poprzednią stronę listy. 4. System aktualizuje wyświetlaną listę zgodnie z działaniami aktora
Ścieżka alternatywna:	Brak
Warunki końcowe:	Aktor przegląda listę wszystkich kanji

*Tabela 2.4 PU-4 - Przejrzyj listę kanji*

Lp.	PU-5
Nazwa:	Zmień metrykę sortowania uczonych kanji
Aktor:	Użytkownik Zalogowany
Warunki początkowe:	Aktor jest zalogowany na swoje konto
Opis:	Aktor chce zmienić metrykę sortowania uczonych kanji.
Ścieżka główna:	1. Aktor klika na ikonę zębatego koła w nagłówku strony. 2. System przekierowuje aktora do strony z ustawieniami. 3. Aktor wybiera preferowaną opcję sortowania uczonych kanji z rozwijanej listy. 4. Aktor klika przycisk „Save Kanji Order Preferences”. 5. System zapisuje wybraną preferencję aktora i wyświetla odpowiedni komunikat.
Ścieżka alternatywna:	3a. W przypadku wybrania opcji „Heisig” aktor może też zaznaczyć, czy chce używać kolejności z szóstej edycji Remembering the Kanji czy ze starszej wersji.
Warunki końcowe:	Preferencja sortowania została pomyślnie zaktualizowana.

*Tabela 2.5 PU-5 - Zmień metrykę sortowania uczonych kanji*

Lp.	PU-6
Nazwa:	Przejrzyj słownik wybranej kanji
Aktor:	Użytkownik zalogowany
Warunki początkowe:	Aktor jest zalogowany.
Opis:	Aktor chce przejrzeć słownik danej kanji.
Ścieżka główna:	1. Aktor wybiera kanji którego słownik chce przejrzeć w PU-4 lub klika przycisk „Learn This Kanji” na stronie głównej by zostać przekierowanym do najwcześniejszej nienauczonych kanji. 2. System przekierowuje aktora do słownika wybranej kanji 3. Użytkownik może nawigować do sąsiednich kanji lub przejść do PU-7
Ścieżka alternatywna:	Brak
Warunki końcowe:	Aktor znajduje się na stronie ze słownikiem wybranej kanji

*Tabela 2.6 PU-6 - Przejrzyj słownik wybranej kanji*

Lp.	PU-7
Nazwa:	Edytuj dane kanji użytkownika
Aktor:	Użytkownik zalogowany
Warunki początkowe:	Aktor jest w trakcie przeglądania słownika danej kanji.
Opis	Aktor chce dokonać edycji notatek, słowa-klucza lub dodać kanji do „nauczonych”.
Ścieżka główna:	1. Aktor klika w pole tekstowe notatek lub słowa-klucza. 2. Aktor edytuje treść wybranego pola i klika przycisk „Save”. 3. System zapisuje zmiany i dodaje kanji do „nauczonych” jeżeli była to pierwsza edycja dla tego znaku kanji.
Ścieżka alternatywna:	1a. Aktor klika w przycisk „Save” bez edycji pól tekstowych. System dodaje znak kanji do „nauczonych” 2a. Edytowane treści przekraczają maksymalną ilość znaków, system wyświetla odpowiedni komunikat
Warunki końcowe:	System pomyślnie zapisuje zmiany w kanji aktora i dodaje kanji do „nauczonych” jeżeli była to pierwsza edycja aktora dla tego znaku kanji.

*Tabela 2.7 PU-7 - Edytuj dane kanji użytkownika*

Lp.	PU-8
Nazwa:	Rozpocznij sesję nauki SRS
Aktor:	Użytkownik zalogowany
Warunki początkowe:	Aktor jest zalogowany i ma wyznaczone przez system dzisiejsze powtórki fiszek.
Opis:	Aktor chce rozpocząć sesję SRS
Ścieżka główna:	1. Aktor klika na przycisk „Start Reviews” na stronie głównej lub „Reviews” w nagłówku strony. 2. System przekierowuje aktora do strony z powtórkami wszystkich dzisiejszych fiszek.
Ścieżka alternatywna:	
Warunki końcowe:	Aktor przechodzi do realizacji PU-9 dla pierwszej kanji w aktualnej sesji.”

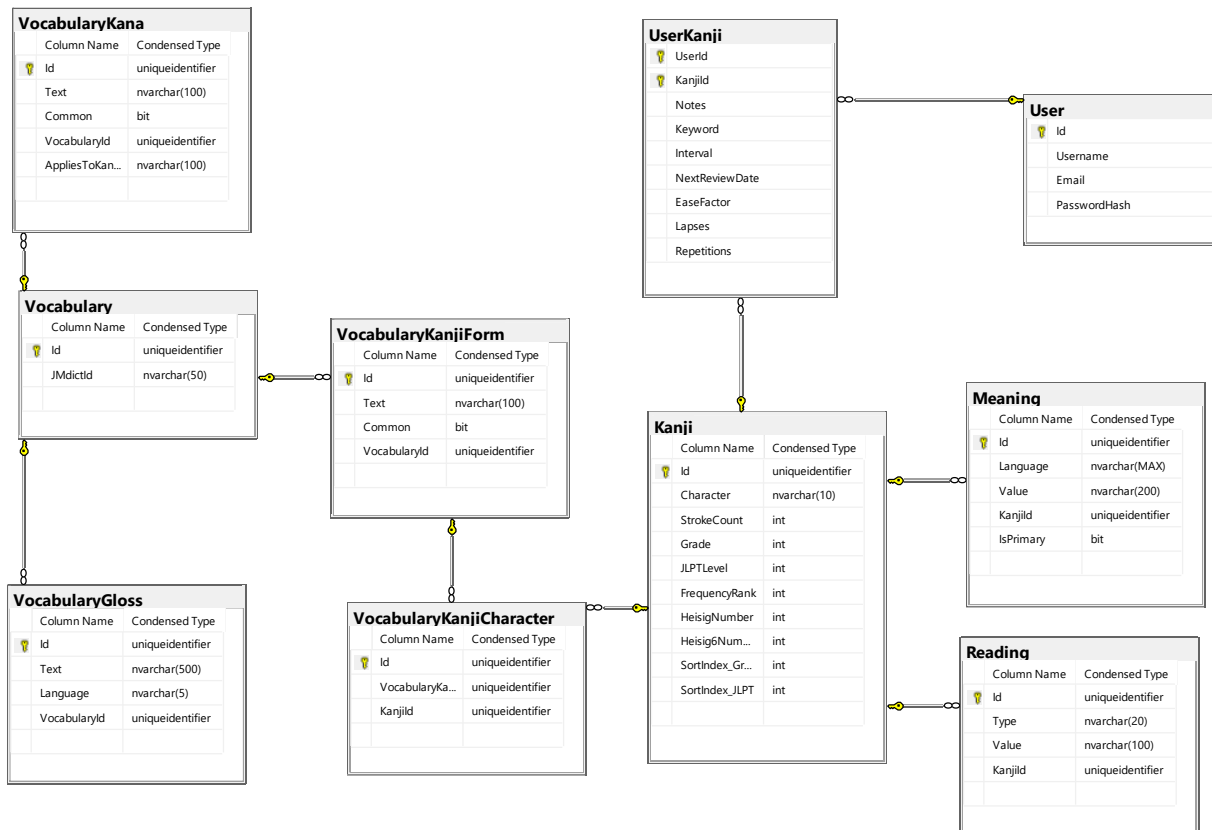
*Tabela 2.8 PU-8 - Rozpocznij sesję nauki SRS*

Lp.	PU-9
Nazwa:	Wykonaj powtórkę fiszki.
Aktor:	Użytkownik zalogowany
Warunki początkowe:	Aktor jest w trakcie sesji SRS
Opis:	Aktor wykonuje powtórkę fiszki.
Ścieżka główna:	1. System wyświetla dla aktora dzisiejszą fiszkę stroną zawierającą jedynie słowo-klucz. 2. Aktor klika przycisk odpowiadający poziomowi na jakim uważa, że zapamiętał kanji z danej fiszki. 3. Jeżeli w sesji są jeszcze dzisiejsze fiszki, następuje PU-9 dla kolejnej kanji.
Ścieżka alternatywna:	2a. Aktor może odwrócić fiszkę poprzez kliknięcie jej by wyświetlić znak kanji i odpowiadającą mu własną notatkę. 2b. Aktor może w dowolnym momencie zakończyć sesję wcześniej za pomocą przycisku „End Session”. System wyświetla odpowiedni komunikat i zapisuje dotychczasowy postęp. 3a. Jeżeli Aktor chce cofnąć swój wybór, może kliknąć przycisk „Undo” i wrócić do punktu 2. 3b. Jeżeli w sesji skończyły się dzisiejsze powtórki, system wyświetla komunikat o końcu sesji i zapisuje wyniki sesji.
Warunki końcowe:	Aktor przeszedł do kolejnej fiszki lub zakończył sesję SRS.

*Tabela 2.9 PU-9 - Wykonaj powtórkę fiszki.*

## 2.4 Baza danych

Baza danych wykonana została w systemie Microsoft SQL Server przy pomocy mapowania obiektowo-relacyjnego (ang. Object-Relational Mapping, ORM) oferowanego przez Entity Framework Core. Mechanizm ORM odwzorowuje utworzone modele domenowe aplikacji na bazodanowe tabele.



Rysunek 2.2 - Schemat bazy danych

Baza danych składa się z następujących tabel:

1. „User” – przechowuje informacje identyfikacyjne użytkowników aplikacji oraz dane niezbędne do procesu uwierzytelniania. Zawiera następujące pola:
  - „Id” – unikalny identyfikator użytkownika, klucz główny tabeli,
  - „Username” – unikalna nazwa użytkownika
  - „Email” – unikalny adres poczty elektronicznej użytkownika
  - „PasswordHash” – hasło użytkownika zapisane w postaci zahasowanej w celu zapewnienia bezpieczeństwa danych



2. „UserKanji” – tabela przechowująca personalizowane przez użytkownika dane kanji oraz wartości dotyczące progresu w nauce danego znaku potrzebne do SRS. Zawiera następujące pola:

- „UserId” – klucz obcy do tabeli „User”
- „KanjiId” – klucz obcy do tabeli „Kanji”, wraz z „UserId” tworzy klucz główny złożony
- „Notes” – notatki użytkownika
- „Keyword” – słowo-klucz używane w fiszkach
- „Interval” – aktualny interwał powtórki w systemie SRS
- „NextReviewDate” – data następnej powtórki
- „EaseFactor” – współczynnik łatwości z algorytmu SM-2
- „Lapses” – liczba pokazująca, ile razy użytkownik zapomniał tą kanji
- „Repetitions” – liczba poprawnych powtórek kanji z rzędu przez użytkownika

3. „Kanji” – tabela przechowująca dane o znakach kanji, połączona relacjami jeden do wielu z „Reading” ponieważ kanji mogą mieć wiele sposobów wymowy oraz „Meaning” ponieważ kanji może mieć kilka znaczeń zależnych od kontekstu

- „Id” – unikalny identyfikator kanji
- „Character” - znak kanji
- „Stroke Count” – liczba kresek w znaku
- „Grade” – poziom lub klasa, w której znak jest nauczany w japońskim systemie edukacji
- „JLPT” – poziom egzaminu JLPT (Japanese Language Proficiency Test)
- „FrequencyRank” - ranking częstotliwości występowania znaku
- „HeisigNumber” – numer znaku w starszych edycjach Remembering the Kanji autorstwa Heisiga
- „Heisig6Number” – numer znaku w szóstej edycji Remembering the Kanji
- „SortIndex\_Grade” – index znaku posortowanego najpierw przez klasę, a następnie częstotliwość występowania
- „SortIndex\_JLPT” – index znaku posortowanego najpierw przez poziom JLPT, a następnie częstotliwość występowania

4. „Meaning” - tabela przechowująca znaczenia znaków kanji
  - „Id” – unikalny identyfikator znaczenia, klucz główny tabeli
  - „Language” – kod języka znaczenia, obecnie jedynie „en” dla angielskiego
  - „Value” – treść znaczenia
  - „KanjiId” – klucz obcy do tabeli „Kanji”
  - „IsPrimary” – wartość binarna wskazująca na to, czy znaczenie jest określone jako główne
5. „Reading” – tabela przechowująca sposoby wymowy znaków kanji
  - „Id” – unikalny identyfikator sposobu wymowy, klucz główny tabeli
  - „Type” – typ sposobu wymowy, dzieli się je na *On’yomi* (oparte na historycznej wymowie chińskiej) i *Kun’yomi* (odczyty rodzimie japońskie)
  - „Value” – treść sposobu wymowy w postaci hiragany (dla odczytów *Kun’yomi*) lub katakany (dla odczytów *On’yomi*)
6. „VocabularyKanjiCharacter” – tabela pośrednia między dla relacji wiele-do-wielu między „VocabularyKanjiForm” i „Kanji”, ponieważ słowa japońskie mogą mieć więcej niż jedną formę zawierającą dany znak, a także zawierać w sobie różne kanji
  - „Id” – unikalny identyfikator, klucz główny tabeli
  - „VocabularyId” – klucz obcy do tabeli „Vocabulary”
  - „KanjiId” – klucz obcy do tabeli „Kanji”
7. „VocabularyKanjiForm” – tabela przechowująca formy zapisu słownictwa zawierające znaki kanji
  - „Id” - unikalny identyfikator, klucz główny tabeli
  - „Text” – forma słowa zawierająca kanji
  - „Common” - wartość binarna wskazująca na to, czy ta forma słowa jest powszechnie stosowana
  - „VocabularyId” – klucz obcy do tabeli „Vocabulary”
8. „Vocabulary” – tabela nadrzędna reprezentująca hasło słownikowe, łącząca 3 różne aspekty japońskiego słownictwa. Jedno hasło może posiadać wiele tłumaczeń przechowywanych w tabeli „VocabularyGloss”, wiele sposobów wymowy zapisanych w tabeli „VocabularyKana” oraz wiele form zapisu z użyciem znaków kanji przechowywanych w tabeli „VocabularyKanjiForm”.
  - „Id” - unikalny identyfikator w systemie, klucz główny tabeli
  - „JMdictId” – identyfikator ze słownika używanego w systemie, JMdict

9. „VocabularyKana” – tabela przechowująca odczyty słów w formie katakany lub hiragany, oraz słowa występujące jedynie w formach tych sylabariuszy.

- „Id” – unikalny identyfikator, klucz główny tabeli
- „Text” – sposób wymowy lub słowo zapisane w formie hiragany lub katakany
- „Common” - wartość binarna wskazująca na to, czy ta forma słowa jest powszechnie stosowana
- „VocabularyId” – klucz obcy do tabeli „Vocabulary”
- „AppliesToKanjiForm” – wskazuje, do której formy słowa zawierającego kanji odnosi się ten odczyt (jeżeli taka istnieje)

10. „VocabularyGloss” – tabela przechowująca tłumaczenia słów.

- „Id” – unikalny identyfikator tłumaczenia
- „Text” – treść tłumaczenia
- „Language” - kod języka znaczenia, obecnie jedynie „eng” dla angielskiego
- „VocabularyId” – klucz obcy do tabeli „Vocabulary”

### **3. Implementacja**

#### **3.1 Środowisko i narzędzia programistyczne**

Środowiskiem, w którym napisana została aplikacja było IDE od firmy Microsoft - Visual Studio 2022. Do zarządzania bibliotekami po stronie backend wykorzystano NuGet Package Manager, a dla zarządzania bazą danych wykorzystano SQL Server Management Studio. Do testowania endpointów web API stosowano narzędzie Postman. Po stronie frontendu użyto npm (Node Package Manager) do zarządzania pakietami oraz Vite jako serwer developerski. W projekcie do konteneryzacji zastosowano Docker Desktop wraz z narzędziem Docker Compose. Do zarządzania kodem źródłowym wykorzystano system kontroli wersji Git.

#### **3.2 Użyte biblioteki i frameworki**

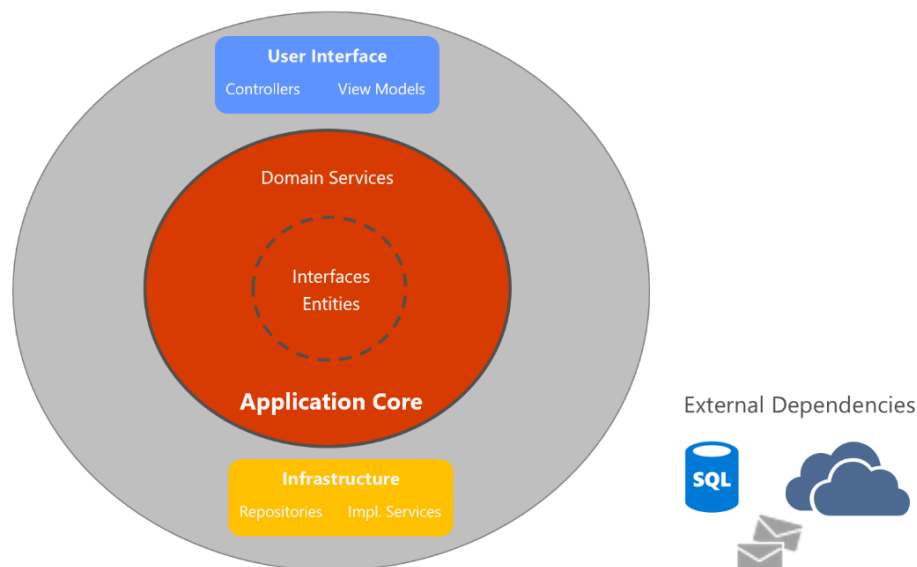
W implementacji zostały wykorzystane przede wszystkim następujące biblioteki i frameworki:

- .NET 8.0 – platforma aplikacyjna Microsoft do budowania aplikacji backendowych
- ASP.NET Core – framework do tworzenia Web API i aplikacji webowych
- Entity Framework Core – framework ORM do mapowania obiektów na struktury bazy danych
- FluentValidation – do definiowania reguł walidacji
- FluentResults - do obsługi wyników operacji i błędów w sposób eliminujący potrzebę rzucania wyjątków
- React – biblioteka JavaScript służąca do budowania interfejsów opartych na komponentach
- React router DOM – biblioteka do routingu w aplikacji po stronie klienta
- Axios – biblioteka HTTP do komunikacji z API

#### **3.3 Czysta architektura**

Wzorcem projektowym na którym był położony szczególny nacisk była tak zwana architektura czysta (Clean Architecture). Zapewnia ona separację odpowiedzialności między warstwami, niezależność logiki biznesowej od szczegółów implementacyjnych oraz łatwość rozwoju i utrzymania kodu. Zasady tego typu architektury zakładają, że kod dzielony jest na warstwy, gdzie warstwy znajdujące się bliżej wnętrza (Application Core) są niezależne od tych zewnętrznych.

# Clean Architecture Layers (Onion view)



Rysunek 3.1 - Warstwy w czystej architekturze

W projekcie wyszczególnione zostało pięć warstw:

1. Warstwa Domain – rdzeń aplikacji zawierający logikę biznesową. Warstwa ta zawiera:

- Entities – encje reprezentujące obiekty domenowe systemu
- Enums – typy danych o ograniczonym, zdefiniowanym zbiorze wartości
- Repositories – interfejsy repozytoriów
- Utilities – narzędzia pomocnicze

2. Warstwa Application – implementuje logikę aplikacyjną. Jest zależna od Domain.

Warstwa ta zawiera:

- Dtos – obiekty transferu danych(Data Transfer Objects) po stronie serwera w celu kontrolowania jakie dane są wymieniane między serwerem a klientem
- Interfaces – interfejsy serwisów
- Services – implementacje serwisów aplikacyjnych
- QueryFilters – filtry do filtrowania danych w serwisach

3. Warstwa Infrastructure – odpowiada za infrastrukturę związaną z bazą danych. Jest zależna od Domain i Application. Warstwa ta zawiera:

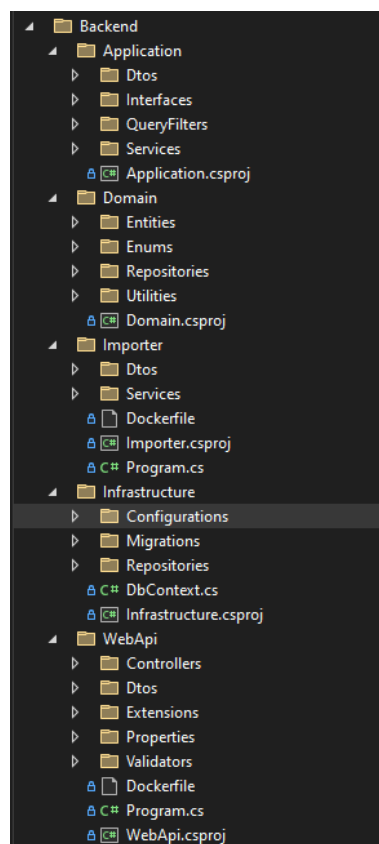
- Repositories – implementacje repozytoriów zdefiniowanych w Domain
- Configurations – konfiguracje tabel w bazie danych
- Migrations – migracje schematu bazy danych generowane przez Entity Framework Core
- DbContext.cs – kontekst bazodanowy Entity Framework Core

4. Warstwa WebApi – punkt wejścia dla aplikacji webowej opartej o RESTful API używającego ASP.NET Core. Jest zależny od wszystkich poprzednich warstw. Zawiera ona:

- Controllers - kontrolery API obsługujące żądania HTTP
- Dtos – obiekty transferu danych po stronie klienta
- Validators - walidatory danych wejściowych wykorzystujące bibliotekę FluentValidation
- Extensions – metody oparte o bibliotekę FluentResults w celu obsługi kodów statusu HTTP
- Program.cs – punkt wejścia i konfiguracji aplikacji

5. Warstwa Importer – warstwa zawierająca projekt importujący dane z plików ze słownikami do bazy danych. Zależna od wszystkich poprzednich warstw z wyjątkiem WebApi. Zawiera ona:

- Dtos – dedykowane obiekty umożliwiające transfer danych
- Services – serwisy zawierające skrypty importujące dane
- Program.cs - punkt wejścia importera



Rysunek 3.2 - Organizacja warstw w projekcie

### 3.4 Importowanie danych

Dane słownikowe importowane są do bazy danych przy użyciu dedykowanego projektu Importer, który uruchamia się jako osobny kontener w Docker Compose. Proces importu wykonywany jest automatycznie przy pierwszym uruchomieniu aplikacji, o ile baza danych nie zawiera jeszcze żadnych rekordów kanji.

Import przebiega w trzech etapach.

1. Importowane są dane o znakach kanji z pliku JSON wygenerowanego na podstawie słownika KANJIDIC2. Zapisywane są podstawowe informacje o kanji wykorzystywane w aplikacji.
2. Importowane są dane słownikowe z pliku JSON pochodzącego ze słownika JMdict. Słownictwo zawiera słowa w formie kanji oraz katakany i hiragany, oraz tłumaczenia na język angielski. Ze względu na dużą ilość danych, import realizowany jest partiami po 5000 rekordów w celu efektywnego zarządzania pamięcią.
3. Dla kanji generowane są dwa dodatkowe indeksy sortowania, pierwszy tworzony jest przez sortowanie najpierw po poziomie kanji w systemie edukacji japońskiej, a następnie po częstotliwości występowania. Drugi zaczyna od poziomów egzaminu JLPT, a następnie także po częstotliwości.

### 3.5 Algorytm SRS

System powtórek rozłożonych w czasie został zaimplementowany w oparciu o algorytm SM-2 (SuperMemo 2) opracowany przez Piotra Woźniaka. Algorytm ten oblicza optymalne interwały między kolejnymi powtórkami na podstawie oceny użytkownika dotyczącej tego, jak dobrze zapamiętał dany znak kanji.

Użytkownik podczas sesji powtórek wybiera jeden z czterech przycisków oceniających jego pamięć: "Again" (całkowite zapomnienie), "Hard" (trudne przypomnienie), "Good" (poprawne przypomnienie) oraz "Easy" (łatwe przypomnienie). W oryginalnym algorytmie skala wynosi od 0 do 5, jednak w tej implementacji wyżej wymienionym przyciskom odpowiadają wartości quality: 0, 3, 4 i 5, co jest bliższe współczesnym aplikacjom w systemach SRS. Na podstawie tej oceny system aktualizuje trzy kluczowe parametry: współczynnik łatwości (EaseFactor), określający względną łatwość zapamiętania danego znaku, Repetitions liczący poprawne powtórki z rzędu oraz Interval wskazujący liczbę dni do następnej powtórki.

W przypadku odpowiedzi z quality równym 0, interwał jest ponownie ustawiany na aktualny dzień. Dla poprawnych odpowiedzi interwał obliczany jest według zasady: 1 dzień po

pierwszej poprawnej powtórce, 6 dni po drugiej, a następnie jako iloczyn poprzedniego interwału i współczynnika łatwości (EaseFactor). Wartość EaseFactor modyfikowana jest według wzoru z algorytmu SM-2

$$EF' = EF + 0.1 - (5 - q)x(0.08 + (5 - q)x0.02)$$

Gdzie:

- EF' – obliczany nowy współczynnik EaseFactor
- EF – poprzedni współczynnik EaseFactor
- q – quality odpowiedzi wybranej przez użytkownika

Współczynnik łatwości ma ustawioną minimalną wartość 1.3, a współczynnik jest dodatkowo mnożony przez 0.9 dla odpowiedzi „Hard” lub przez 1.2 dla odpowiedzi „Easy” w celu lepszego dostosowania interwałów powtórek użytkownika

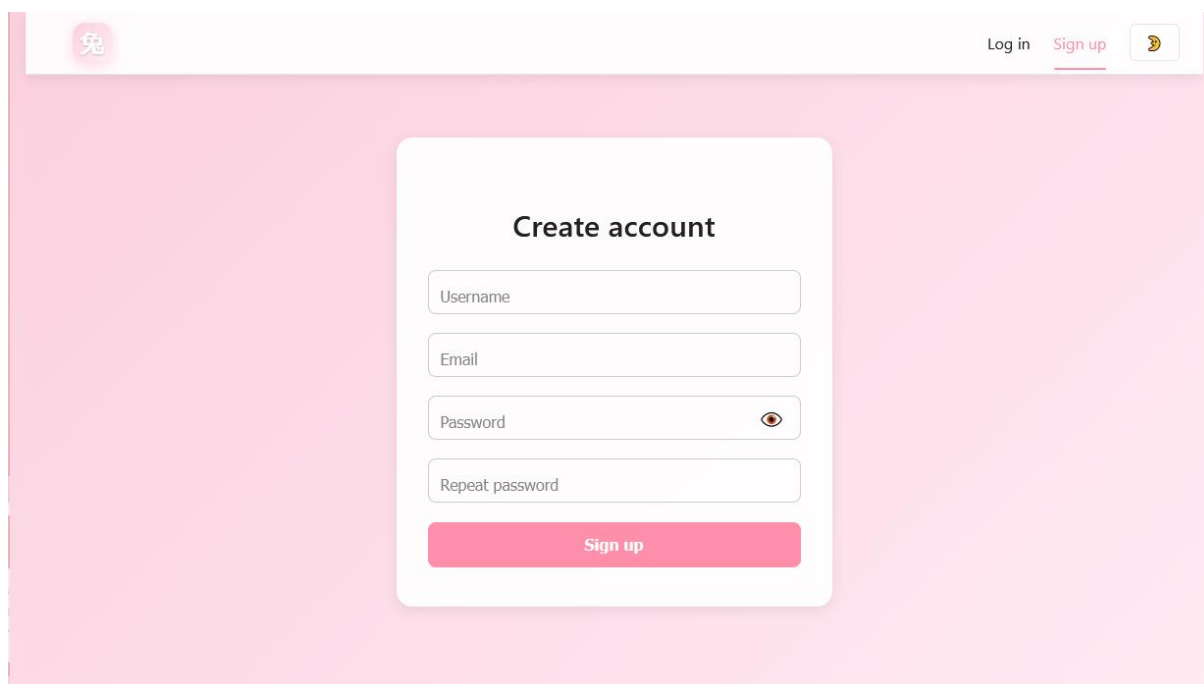


## 4. Interfejs użytkownika

Interfejs użytkownika aplikacji został zaprojektowany z myślą o intuicyjności i komforcie użytkowania. Kolorystyka aplikacji inspirowana jest liśćmi kwitnącej wiśni, nawiązując do kulturowego symbolu Japonii. Użytkownicy mogą również przełączyć interfejs na tryb ciemny według własnych preferencji. W kolejnych podrozdziałach przedstawiono szczegółowy opis poszczególnych widoków aplikacji wraz z ich funkcjonalnościami.

### 4.1 Widok rejestracji

Widok rejestracji umożliwia użytkownikowi utworzenie nowego konta w systemie poprzez formularz, do którego użytkownik wpisuje nazwę użytkownika, adres email oraz dwukrotnie hasło. Dodatkowo formularz oferuje możliwość przełączania widoczności hasła za pomocą ikony oka. Po potwierdzeniu przyciskiem, w przypadku wystąpienia błędów walidacji lub problemów z rejestracją, użytkownik informowany jest o nich poprzez komunikat wyświetlany nad formularzem. Jeżeli rejestracja przeszła pomyślnie system wyświetla odpowiedni komunikat i przekierowuje użytkownika do widoku logowania.



*Rysunek 4.1 - Widok rejestracji z niewypełnionym formularzem*

The screenshot shows a web application interface with a light pink background. At the top, there is a navigation bar with a logo on the left, and 'Log in' and 'Sign up' links on the right. The 'Sign up' link is underlined. Below the navigation bar, centered on the page, is a white rounded rectangle containing the 'Create account' form. The form has a title 'Create account' at the top. Below the title, a red error message states 'Password must be at least 8 characters long'. The form contains four input fields: a text field with 'user', an email field with 'user1@gmail.com', a password field with masked characters and a toggle icon, and another password field with masked characters. At the bottom of the form is a pink 'Sign up' button.

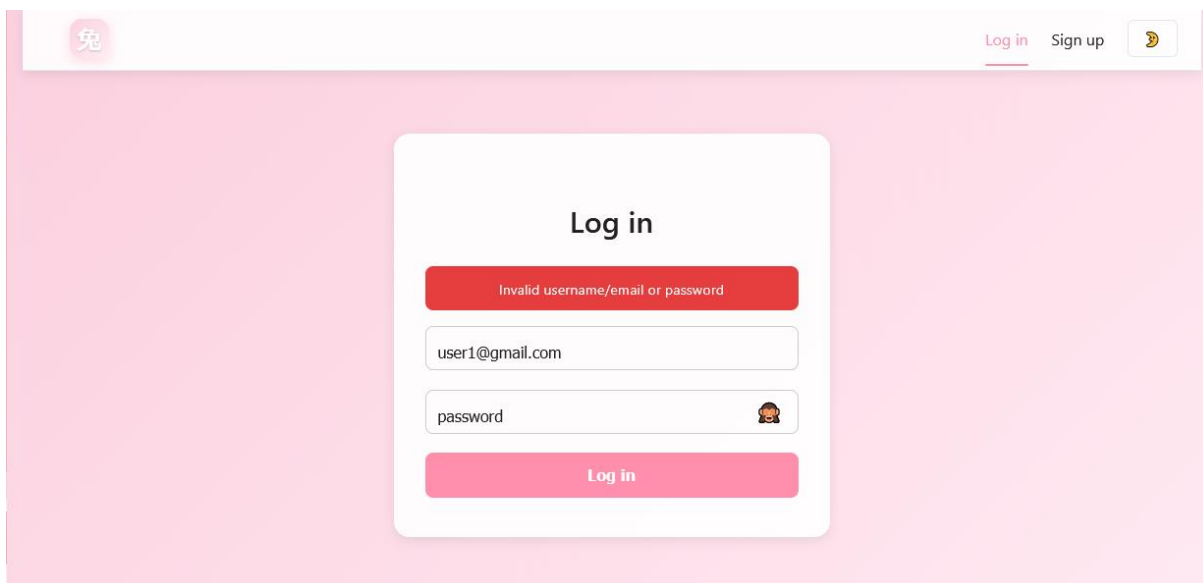
*Rysunek 4.2 - Widok rejestracji w przypadku błędu walidacji*

## 4.2 Widok logowania

Widok logowania umożliwia użytkownikowi zalogowanie się do systemu poprzez formularz, do którego użytkownik wpisuje nazwę użytkownika lub adres email, a także hasło. Podobnie jak w przypadku rejestracji, formularz oferuje możliwość przełączania widoczności hasła, a w przypadku błędnych danych użytkownik informowany jest o tym poprzez komunikat wyświetlany nad formularzem. Jeżeli logowanie przeszło pomyślnie, system przekierowuje użytkownika do strony głównej aplikacji.

The screenshot shows the same web application interface as Figure 4.2. The navigation bar is identical. The main content area features a white rounded rectangle with the 'Log in' form. The form has a title 'Log in' at the top. Below the title, there are two input fields: 'Username or Email' and 'Password'. The 'Password' field has a toggle icon. At the bottom of the form is a pink 'Log in' button.

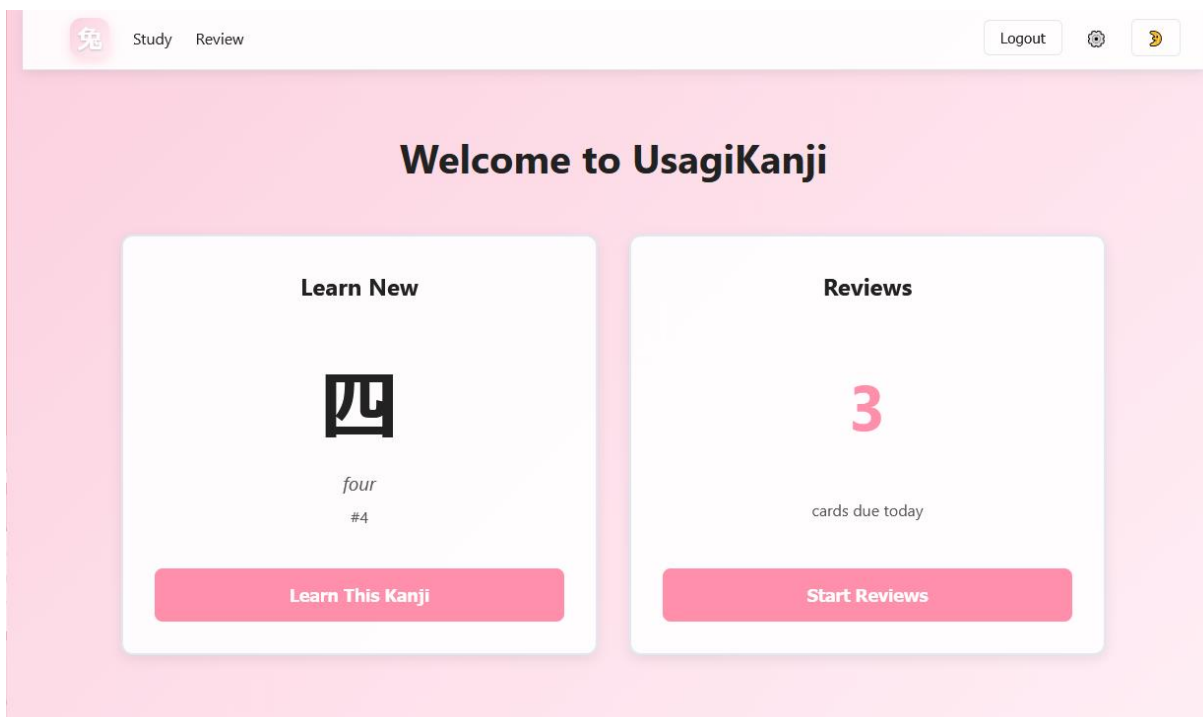
*Rysunek 4.3 - Widok logowania z niewypełnionym formularzem*



Rysunek 4.4 - Widok logowania po przesłaniu błędnych danych

### 4.3 Strona główna

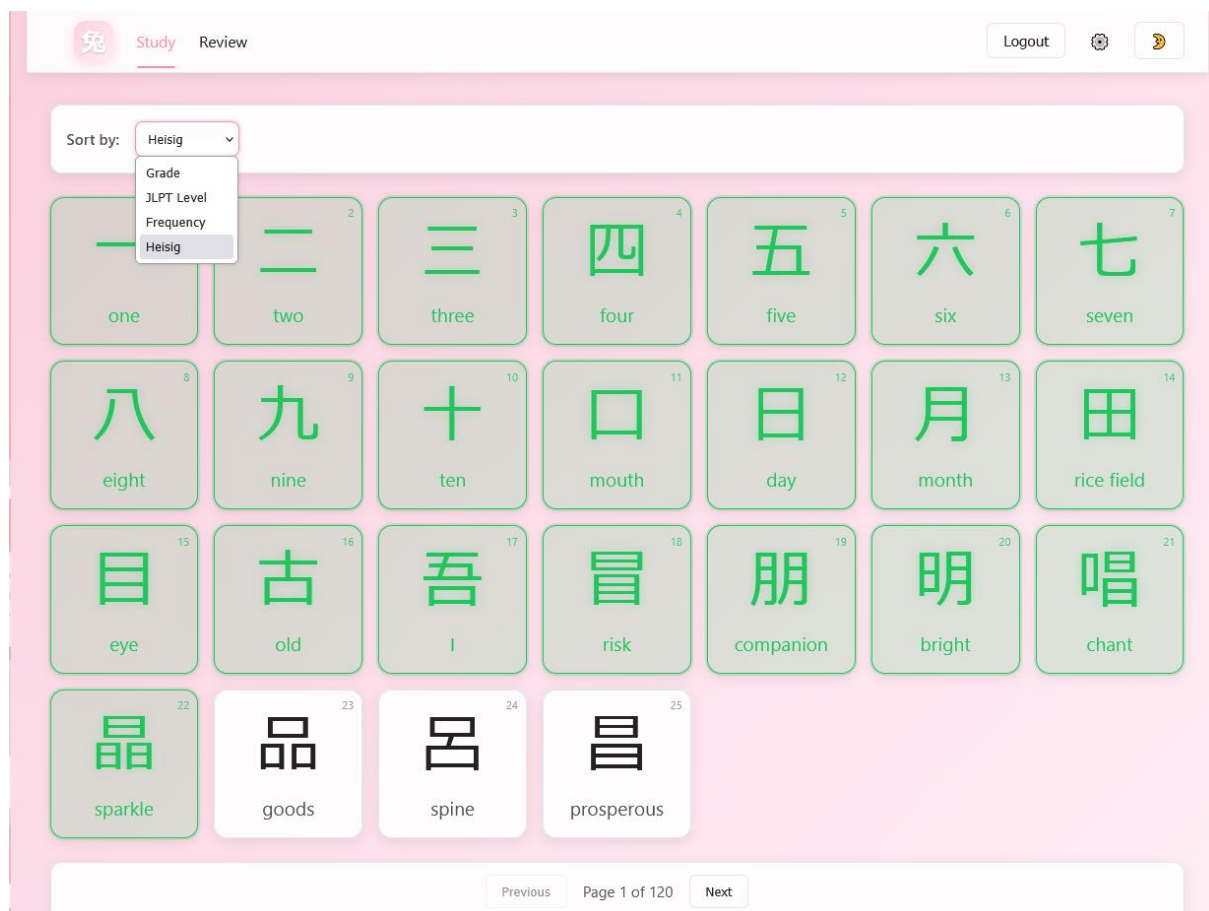
Strona główna aplikacji zawiera w sobie dwa główne elementy. Po lewej stronie pokazany jest następny znak kanji w kolejności wybranego toku uczenia, wraz z głównymi informacjami o niej - znak, główne znaczenie oraz pozycja. Znajduje się tam też przycisk przekierowujący do widoku szczegółów tego znaku. Po prawej użytkownik może zobaczyć, ile ma dzisiaj fiszek do powtórzenia wraz z przyciskiem przekierowującym do widoku powtórek. Do strony głównej można wrócić poprzez kliknięcie loga strony w nagłówku.



Rysunek 4.5 - Strona główna

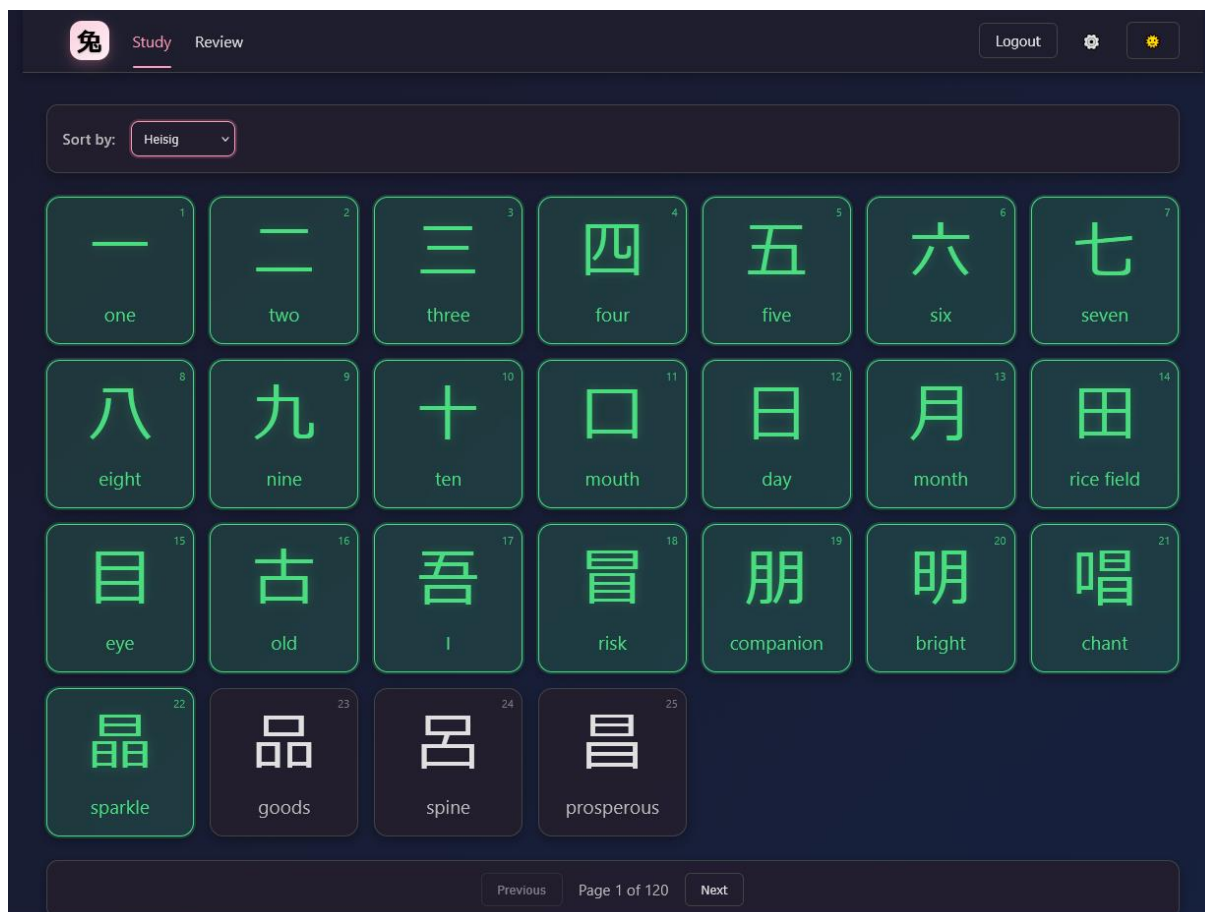
## 4.4 Widok listy znaków kanji

W tym widoku użytkownik może zobaczyć paginowaną listę kanji z danej metody sortowania oraz tymczasowo zmienić tą metodę w celu podglądu za pomocą rozwijanego menu. Kanji dodane przez użytkownika systemu SRS są podświetlone kolorem zielonym. W dolnej części widoku znajdują się numer strony z przyciskami do nawigowania na kolejne bądź poprzednie strony. Naciśnięcie znaku kanji przenosi użytkownika do widoku jego detali. W nagłówku widok ten znajduje się pod nazwą „Study”.



Rysunek 4.6 - Widok listy znaków kanji

Na następnej ilustracji (Rys. 4.7) możemy zobaczyć ten widok jako przykład wyglądu interfejsu po przełączeniu na ciemny motyw przy użyciu przycisku znajdującego się po prawej stronie nagłówka.



Rysunek 4.7 - Przykład ciemnego motywu aplikacji

## 4.5 Widok detali znaku kanji

Widok detali kanji jest miejscem, w którym użytkownik dodaje znaki kanji do systemu SRS. Z lewej strony znajduje się słowo-klucz, którego użyje fiszka oraz informacje o kanji takie jak sam znak, znaczenia, sposoby czytania On'yomi i Kun'yomi, ilość kresek, poziom w edukacji japońskiej, poziom w egzaminie JLPT oraz pozycja w rankingu częstotliwości występowania. Po prawej znajdują się pozycja, nawigacja do następnej bądź poprzedniego kanji, edytowalne notatki użytkownika oraz rozwijalny słownik ze słowami zawierającymi ten znak. Przycisk „Save” zapisuje zmiany i dodaje kanji do systemu SRS, jeżeli jeszcze jej tam nie ma.

兔

Study Review

Logout

month

< 13 >

月

MEANINGS:

month, moon

ADD YOUR NOTES HERE

Save

READINGS

Onyomi:

ガツ

ゲツ

Kunyomi:

つき

Strokes: 4

Grade: 1

JLPT exam level: N4

Frequency Rank: 23

VOCABULARY

九月	くがつ	ninth month of the lunar calendar, September
毎月	まいつき, まいげつ	each month, every month, monthly
月並み	つきなみ	trite, conventional, commonplace, hackneyed, every month
再来月	さらいげつ	month after next

Show more

Rysunek 4.8 - Widok detali znaku kanji

兔

Study Review

Logout

moon

< 13 >

月

MEANINGS:

month, moon

MONTHS ARE HISTORICALLY DERIVED FROM MOON CYCLES. THE CHARACTER LOOKS LIKE A LONGER KANJI FOR 'DAY'.

Keyword and notes saved successfully!

Save

READINGS

Onyomi:

ガツ

ゲツ

Kunyomi:

つき

Strokes: 4

Grade: 1

JLPT exam level: N4

Frequency Rank: 23

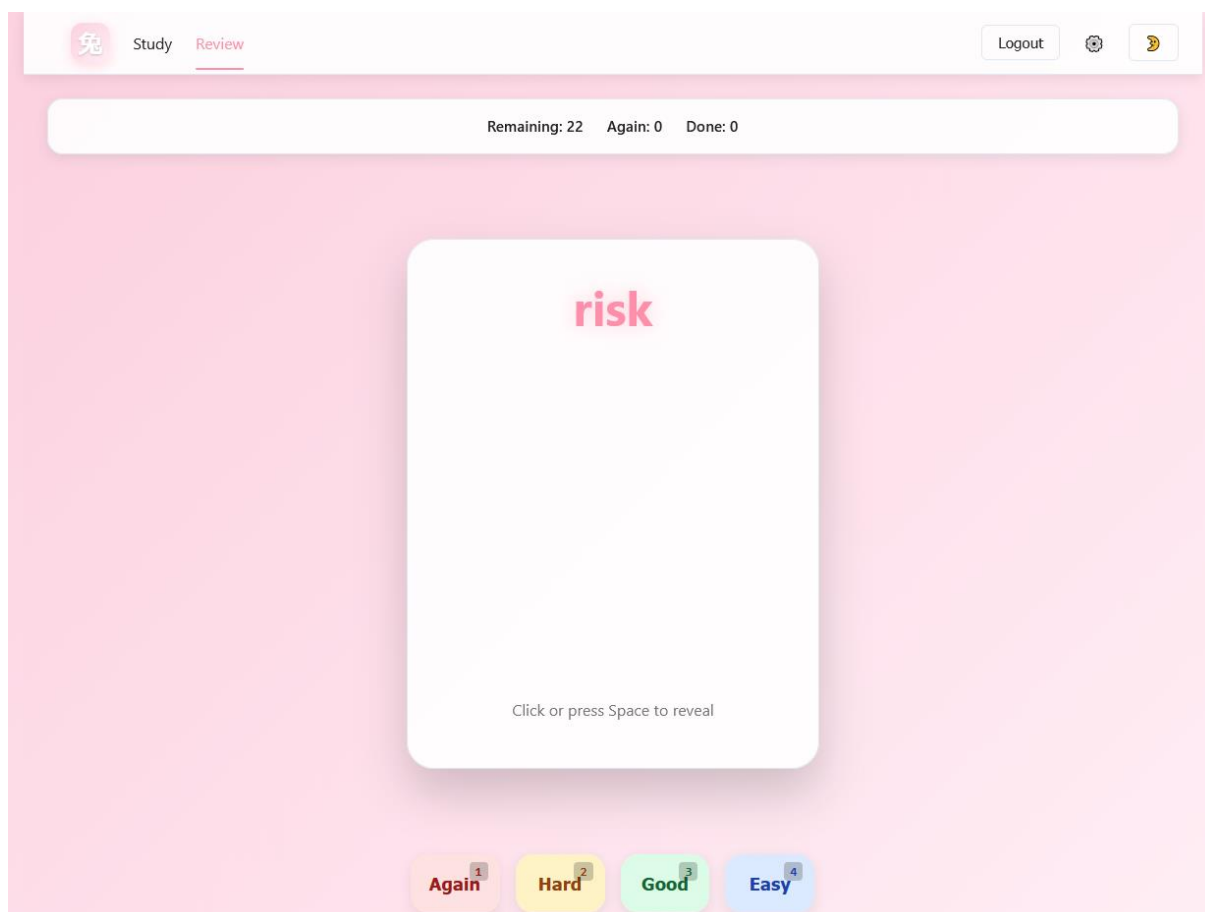
VOCABULARY

九月	くがつ	ninth month of the lunar calendar, September
毎月	まいつき, まいげつ	each month, every month, monthly
月並み	つきなみ	trite, conventional, commonplace, hackneyed, every month
再来月	さらいげつ	month after next

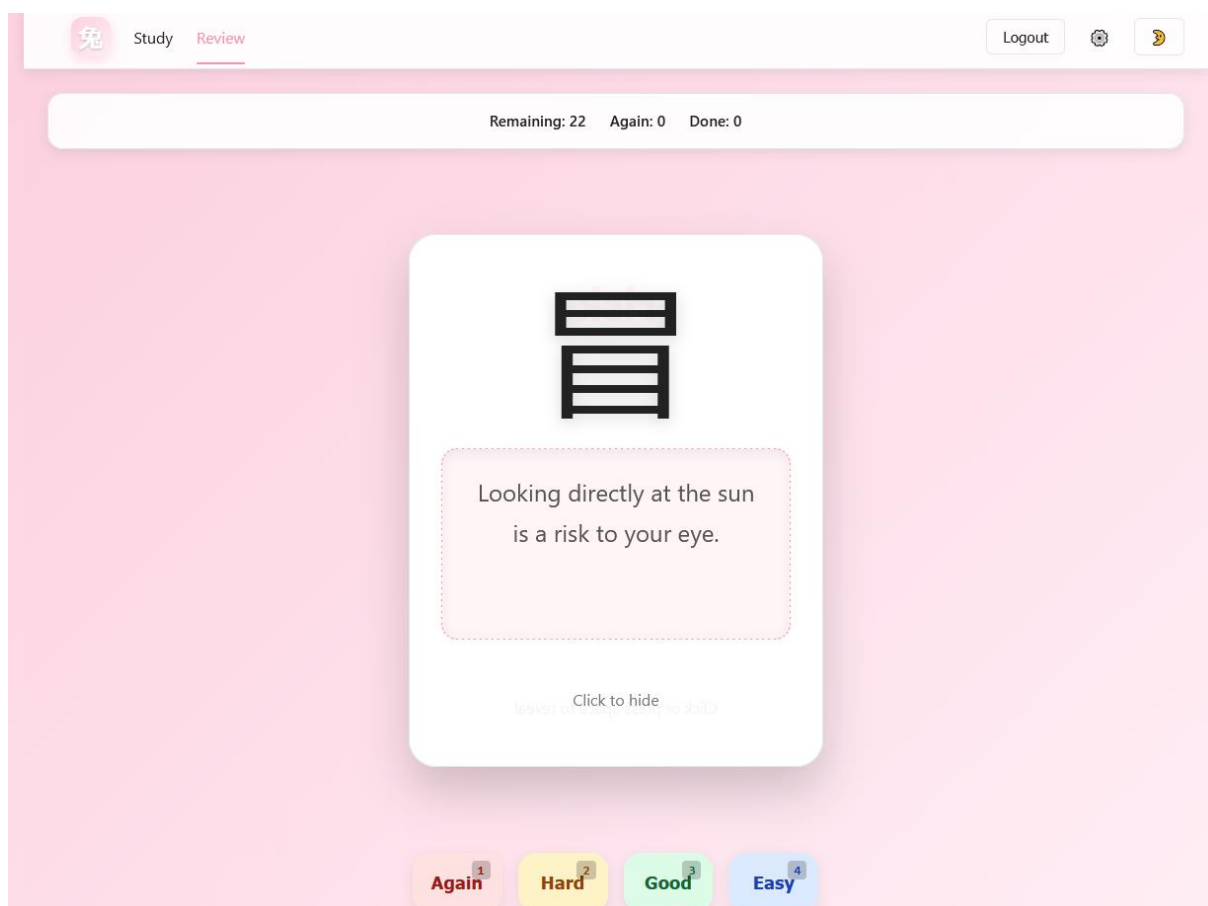
Rysunek 4.9 - Widok detali znaku kanji po edycji i rozwinięciu słownika

## 4.6 Widok powtórek kanji

W sekcji „Review” w nagłówku znajdują się dzisiaj powtarzane fiszki użytkownika. Na dole znajdują się przyciski odpowiadające poziomowi, na jakim użytkownik ocenia swoje zapamiętanie danego znaku. W górnej części wyświetlane są statystyki aktualnej sesji - liczba pozostałych fiszek, liczba fiszek oznaczonych jako „Again” (wymagających ponownej powtórki tego samego dnia) oraz liczba poprawnie zapamiętanych fiszek. Na środku widoku znajduje się sama fiszka ze słowem-klucz ustawionym przez użytkownika. Można ją odwrócić klikając na nią lub naciskając klawisz spacji. Odwrócona karta zawiera znak kanji oraz notatki użytkownika (Rys. 4.11). Od drugiej fiszki w sesji pojawiają się dodatkowe przyciski (Rys. 4.12) - „End session” pozwala zakończyć sesję wcześniej i zapisać postęp, a „Undo” cofa do poprzedniej fiszki. Jest to przydatne, gdy użytkownik przypadkowo kliknął niewłaściwy przycisk.

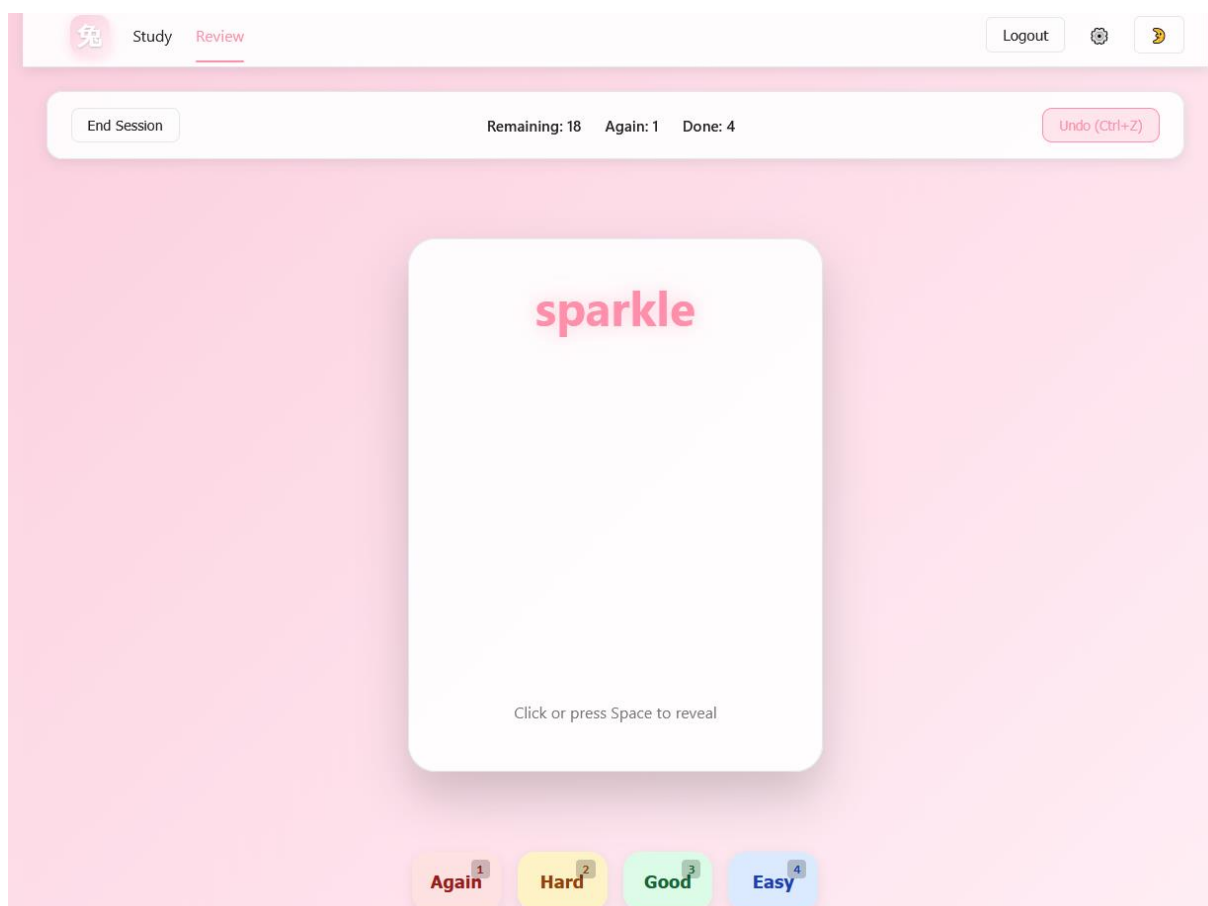


Rysunek 4.10 - Widok powtórki kanji na początku sesji SRS



*Rysunek 4.11 - Widok powtórki kanji po odwróceniu fiszki*

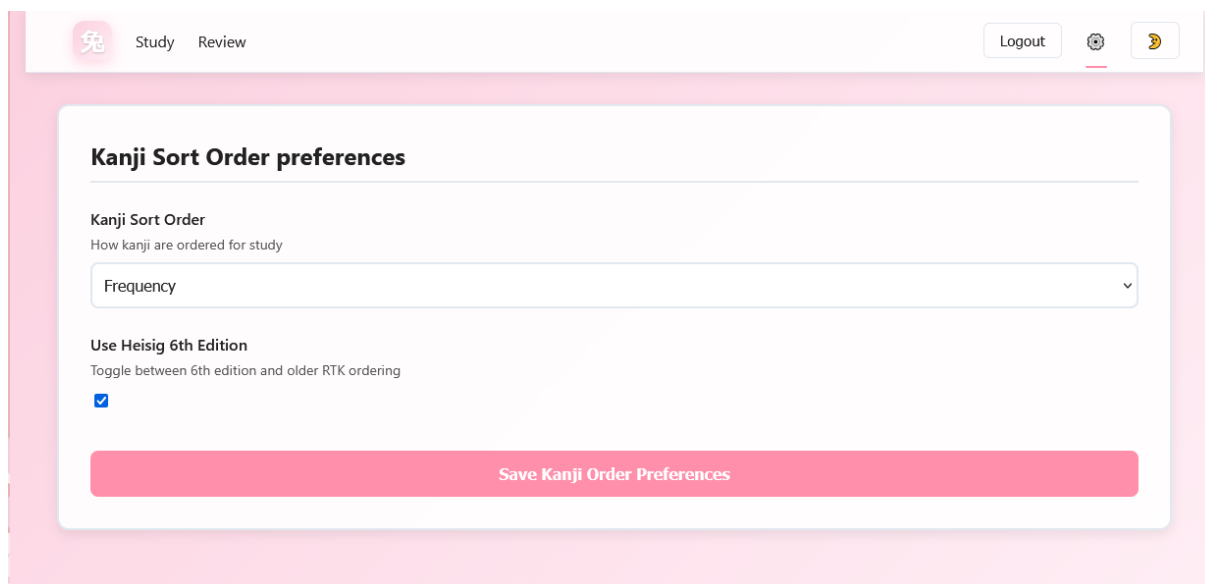




*Rysunek 4.12 - Widok powtórki kanji w trakcie sesji SRS*

## 4.7 Ustawienia

Do ustawień trafiamy poprzez kliknięcie ikony zębatki w prawej części nagłówka. Użytkownik może tutaj zmienić domyślną kolejność nauki znaków kanji na jedną z opcji sortowania dostępnych w widoku listy kanji (Rys. 4.6). Dodatkowo w przypadku wyboru ścieżki „Heisig” dostępna jest opcja wyboru czy chce używać numeracji z szóstej edycji książki Remembering the Kanji czy starszej.



Rysunek 4.13 - Ustawienia

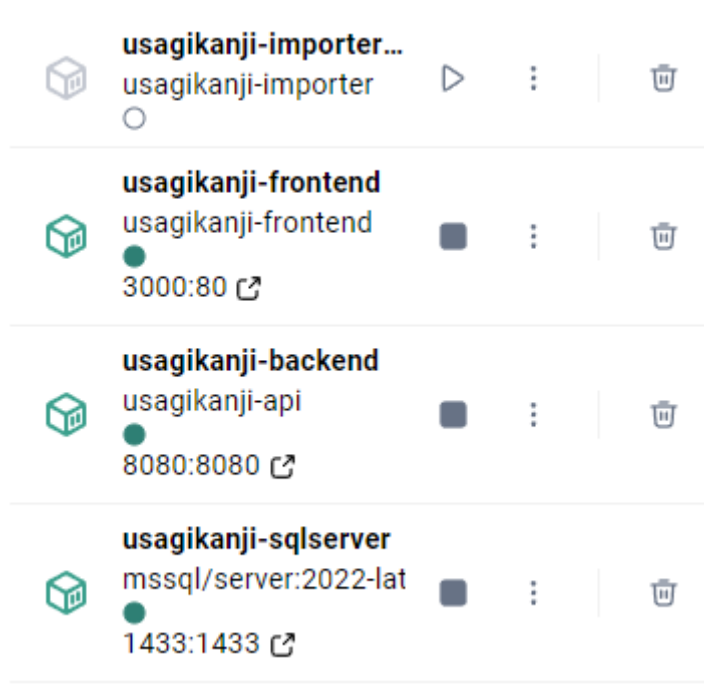
## 5. Konteneryzacja aplikacji oraz możliwości dalszego rozwoju

### 5.1 Konteneryzacja aplikacji

Aplikacja została skonteneryzowana przy użyciu technologii Docker, co zapewnia spójność środowiska między różnymi etapami rozwoju oraz ułatwia wdrożenie. Do zarządzania wielokontenerową architekturą aplikacji wykorzystano Docker Desktop wraz z narzędziem Docker Compose.

Aplikacja składa się z czterech kontenerów:

- „usagikanji-frontend” – kontener z aplikacją kliencką, odpowiedzialny za interfejs użytkownika
- „usagikanji-backend” – kontener z serwerem aplikacji, zawierający logikę biznesową i API
- „usagikanji-sqlserver” – kontener z bazą danych SQL Server, przechowujący dane aplikacji
- „usagikanji-importer” – kontener uruchamiający skrypty importujące dane ze słowników do bazy danych



Rysunek 5.1 - Widok kontenerów z interfejsu Docker Desktop

Wykorzystanie konteneryzacji w projekcie pozwala na izolację środowisk działania komponentów aplikacji oraz identyczne działanie na każdej maszynie. Poza tym ułatwia to uruchomienie wdrożenia aplikacji. Poza tym znacząco ułatwia to zarówno uruchomienie aplikacji podczas developmentu oraz deployment na produkcję.

## 5.2 Instrukcja uruchomienia aplikacji na własnej maszynie

Do uruchomienia aplikacji w kontenerach wymagany jest zainstalowany Docker Desktop oraz Git do sklonowania repozytorium.

1. Sklonowanie repozytorium tego projektu
2. Zmiana nazwy pliku „.env.example” na „.env”.
3. Do uzupełnienia bazy danych potrzebne są pliki JSON „kanjidic2-en” i „jmdict-eng”(najlepiej wersja common) które można znaleźć w repozytorium jmdict-simplified. Pliki z punktu umieszczamy w folderze „UsagiKanji/Backend/Importer/Data”
4. Będąc w katalogu głównym projektu(„/UsagiKanji”), uruchomienie kontenerów komendą „docker-compose up -d”
5. Aplikacja powinna być dostępna pod adresem „http://localhost:3000”

## 5.3 Plany rozwoju aplikacji

Na rynku istnieją rozwiązania do nauki japońskiego rozwijające się od wielu lat, dlatego też naturalnie istnieje wiele funkcjonalności brakujących w tym projekcie, a które poprawiłyby doświadczenie użytkownika w znaczącym stopniu. Istnieje więc duże pole do dalszego rozwoju tej aplikacji i pomysły na dodatkowe funkcje.

Jedną z takich funkcjonalności jest możliwość ręcznego pisania kanji bezpośrednio na ekranie. Zwiększyłoby to interaktywność aplikacji i pomogło w uczeniu się poprawnej pisowni japońskich znaków, zwłaszcza że te formy często różnią się od cyfrowych czcionek. KanjiVG jest projektem zawierającym obrazy grafiki wektorowej dla znaków kanji uwzględniającym poprawną kolejność pisania konkretnego znaku. Zastosowanie tych plików z biblioteką służącą do rysowania, taką jak na SVG.js, pozwoliłoby na zarówno wspomagane jak i samodzielne pisanie znaków.

Kolejną drogą rozwoju aplikacji jest dodanie większej ilości języków. Język angielski jest aktualnie jedynym dostępnym w aplikacji, co wynika z dużej dostępności źródeł(zwłaszcza JMdict) oraz preferencji autora względem do tego języka. Uwzględnienie języka polskiego mogłoby zwiększyć dostępność aplikacji dla użytkowników z Polski, wyróżnić projekt na tle innych narzędzi do nauki języka japońskiego, gdyż brakuje polskojęzycznych rozwiązań w tej dziedzinie.

Rozbudowa słownika byłaby kolejnym istotnym kierunkiem rozwoju aplikacji. Szczególnie przydatne byłoby dodanie rozkładu kanji na elementy je budujące, tak jak umożliwia to projekt AnimCJK. Tego typu funkcjonalność mogłaby wspomagać naukę poprzez

lepszego zrozumienia struktury znaków oraz ułatwiać łączenie ich w historie zgodnie z metodologią Remembering the Kanji.

Innym kierunkiem rozwoju byłoby też wprowadzenie gamifikacji, statystyk nauki oraz elementu społecznego w postaci udostępniania swoich notatek z kanji innym użytkownikom. Umożliwiłoby to dzielenie się wiedzą oraz pomogłoby to tym osobom, którzy mają problemy z wymyśleniem własnych historii. Gamifikacja i statystyki z kolei zwiększyłyby motywację użytkowników, co jest istotne, gdy nauka kanji jest tak żmudnym i długotrwałym procesem.

## 6. Podsumowanie

Celem niniejszej pracy inżynierskiej było zaprojektowanie i implementacja aplikacji webowej służącej jako miejsce do nauki znaków kanji. Miała ona zawierać obszerny słownik oraz system powtórek rozłożonych w czasie dla jak najlepszego doświadczenia użytkownika. Motywacją do realizacji projektu był brak jednego, spójnego rozwiązania integrującego wszystkie kluczowe funkcjonalności wspomagające naukę kanji.

Użytkownik ma możliwość utworzenia własnego konta, przeglądania znaków kanji wraz z ich szczegółowymi informacjami słownikowymi oraz dodawania wybranych znaków do własnej kolekcji fiszek. Fiszki te są następnie obsługiwane przez mechanizm SRS, który generuje codzienne sesje powtórek. Nie udało się w pełni spełnić wizji autora, lecz duża ilość pomysłów na dodatkowe funkcje i kod źródłowy zaprojektowany do ciągłego rozwoju ze względu na wybraną architekturę stanowią solidną podstawę na dalszy rozwój tej aplikacji.

## Bibliografia

1. Docker Inc., 2025. *Docker Documentation*.  
<https://docs.docker.com/>
2. Esposito D., 2024. *Clean Architecture with .NET*. Microsoft Press.
3. Heisig J. W., 2022. *Remembering the Kanji 1: A Complete Course on How Not to Forget the Meaning and Writing of Japanese Characters*, 6th edition. University of Hawaii Press, Honolulu.
4. Microsoft, 2025. *ASP.NET Core documentation*.  
<https://learn.microsoft.com/aspnet/core/>
5. Microsoft, 2025. *Entity Framework Core documentation*.  
<https://learn.microsoft.com/ef/core/>
6. Microsoft, 2025. *Modern Web Applications with ASP.NET Core and Azure – Common web application architectures*.  
<https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>
7. Microsoft, 2025. *TypeScript Documentation*.  
<https://www.typescriptlang.org/docs/>
8. React Team, 2025. *React – Official Documentation*.
9. The Electronic Dictionary Research and Development Group (EDRDG), 2024. *JMdict Japanese–Multilingual Dictionary*.  
<https://www.edrdg.org/wiki/index.php/JMdict>
10. Woźniak P., 1990. *Optimization of learning*. Praca magisterska. Politechnika Poznańska. Fragmenty dostępne Internet: <https://super-memory.com/english/ol.htm>
11. Vite Team, 2025. *Vite – Next Generation Frontend Tooling*.  
<https://vitejs.dev/>
12. Repozytorium projektu “jmdict-simplified” w serwisie GitHub.  
<https://github.com/scriptin/jmdict-simplified>

## Spis ilustracji

Rysunek 1.1 - Interfejs aplikacji "Anki" .....	7
Rysunek 1.2 - Interfejs strony internetowej "Kakimashou" .....	8
Rysunek 2.1 - Diagram przypadków użycia .....	10
Rysunek 2.2 - Schemat bazy danych.....	16
Rysunek 3.1 - Warstwy w czystej architekturze .....	21
Rysunek 3.2 - Organizacja warstw w projekcie .....	22
Rysunek 4.1 - Widok rejestracji z niewypełnionym formularzem.....	25
Rysunek 4.2 - Widok rejestracji w przypadku błędu walidacji.....	26
Rysunek 4.3 - Widok logowania z niewypełnionym formularzem.....	26
Rysunek 4.4 - Widok logowania po przesłaniu błędnych danych .....	27
Rysunek 4.5 - Strona główna .....	27
Rysunek 4.6 - Widok listy znaków kanji .....	28
Rysunek 4.7 - Przykład ciemnego motywu aplikacji .....	29
Rysunek 4.8 - Widok detali znaku kanji .....	30
Rysunek 4.9 - Widok detali znaku kanji po edycji i rozwinięciu słownika .....	30
Rysunek 4.10 - Widok powtórki kanji na początku sesji SRS.....	31
Rysunek 4.11 - Widok powtórki kanji po odwróceniu fiszki .....	32
Rysunek 4.12 - Widok powtórki kanji w trakcie sesji SRS .....	33
Rysunek 4.13 - Ustawienia.....	34
Rysunek 5.1 - Widok kontenerów z interfejsu Docker Desktop.....	35



## Spis tabel

Tabela 2.1 PU-1 - Zarejestruj się .....	11
Tabela 2.2 PU-2 - Zaloguj się .....	12
Tabela 2.3 PU-3 - Wyloguj się .....	12
Tabela 2.4 PU-4 - Przeglądaj listę kanji .....	13
Tabela 2.5 PU-5 - Zmień metrykę sortowania uczonych kanji .....	13
Tabela 2.6 PU-6 - Przeglądaj słownik wybranej kanji .....	14
Tabela 2.7 PU-7 - Edytuj dane kanji użytkownika .....	14
Tabela 2.8 PU-8 - Rozpocznij sesję nauki SRS .....	15
Tabela 2.9 PU-9 - Wykonaj powtórkę fiszki.....	15