

# Koncepcja wykonania systemu Liskool

Viktoria Kashpruk, Cezary Miłek

<b>1. Scenariusze przypadków użycia.....</b>	<b>3</b>
<b>2. Projekty ekranów.....</b>	<b>8</b>
Ekran startowy.....	8
Po kliknięciu "Zarejestruj się".....	8
Po kliknięciu "Zaloguj się".....	9
Ekran przesyłania materiałów.....	9
Po kliknięciu "Ustawienia zaawansowane".....	10
Po przesłaniu materiałów.....	10
Ekran po wygenerowaniu modułu.....	11
Ekran szczegółów modułu.....	11
Po kliknięciu "Quizy".....	12
Po wykonaniu quizu.....	12
Ekran monitorowania postępów.....	13
Ekran zarządzania użytkownikami i rolami.....	14
Po kliknięciu "Dodaj użytkownika".....	14
Po kliknięciu "Edycja użytkownika".....	15
Po kliknięciu "Usuń użytkownika".....	15
Po kliknięciu "Historia działań".....	16
Ekran raportów grupowych.....	16
Po kliknięciu "Generuj raport".....	17
<b>3. Architektura ogólna.....</b>	<b>18</b>
Komponenty:.....	18
Diagram bazy danych:.....	20
<b>4. Główne zasady kodowania.....</b>	<b>21</b>
Pisanie kodu w Pythonie:.....	21
Frontend mobilny (Java/Kotlin).....	21
Dokumentacja i komentarze.....	21
Wersjonowanie kodu.....	21
Wydajność i skalowalność.....	22
<b>5. Identyfikacja i zasady zarządzania ryzykiem.....</b>	<b>23</b>
1. Możliwość przeciążenia serwerów.....	23
2. Niedotrzymanie terminów realizacji projektu.....	23

3. Problemy z bezpieczeństwem danych użytkowników.....	23
4. Niska akceptacja użytkowników (UX/UI).....	24
5. Problemy z adaptacją algorytmów AI.....	24
6. Problemy z integracją systemów (backend, frontend, AI).....	24
7. Problemy związane z utrzymaniem infrastruktury.....	25
<b>6. Zgodność prac z tablicą koncepcyjną i specyfikacją.....</b>	<b>26</b>



# 1. Scenariusze przypadków użycia

## 1. Rejestracja i logowanie użytkownika

### Scenariusz główny:

- Użytkownik uruchamia aplikację Liskool.
  - Wybiera jedną z dwóch opcji: **Zarejestruj się** lub **Zaloguj się**.
  - **Rejestracja:**
    - Użytkownik klika przycisk **Zarejestruj się**, otwiera się nowy widok formularza rejestracyjnego.
    - Wypełnia pola, podając:
      - Imię i nazwisko.
      - Adres e-mail.
      - Hasło.
      - Rolę użytkownika (uczeń, nauczyciel, rodzic, administrator).
    - Po uzupełnieniu formularza klika przycisk **Zarejestruj**.
    - System sprawdza poprawność danych (np. liczby znaków, czy mail jest dostępny itp.).
    - W przypadku poprawnych danych użytkownik jest rejestrowany, a następnie automatycznie logowany.
  - **Logowanie:**
    - Użytkownik klika przycisk **Zaloguj się**, otwiera się widok logowania.
    - Wprowadza adres e-mail i hasło.
    - Klika przycisk **Zaloguj**.
    - System sprawdza dane logowania.
    - W przypadku poprawnych danych użytkownik uzyskuje dostęp do swojego panelu.
  - W przypadku błędnych danych system wyświetla komunikat o błędzie.
- 

## 2. Przesyłanie materiałów edukacyjnych

### Scenariusz główny:

- Zalogowany użytkownik przechodzi do panelu **Prześlij materiał**.
- Klika przycisk **Dodaj plik**, otwiera się widok wyboru pliku z urządzenia.
- Wybiera plik z obsługiwanych formatów (PDF, DOCX, PPT, MD, JPEG, PNG).
- Opcjonalnie klika przycisk **Ustawienia zaawansowane** i dostosowuje:

- Preferowany język.
- Poziom trudności.
- Typ materiałów wynikowych (notatki, quizy, wizualizacje).
- Użytkownik klika przycisk **Prześlij**.
- System:
  - Wykonuje OCR, jeśli to konieczne.
  - Analizuje treści, tworząc moduły edukacyjne.
  - Wyświetla komunikat o zakończeniu procesu wraz z przyciskiem **Przejdź do materiału**.
- Użytkownik klika przycisk i uzyskuje dostęp do wynikowego modułu edukacyjnego.

#### Scenariusz alternatywny:

- Plik w nieobsługiwanym formacie:
    - System wyświetla komunikat o błędzie i sugestiami rozwiązania.
- 

### 3. Generowanie modułów edukacyjnych

#### Scenariusz główny:

- Użytkownik (uczeń, nauczyciel lub rodzic) przesyła materiały do systemu, wybierając opcję **Prześlij materiały**.
- System przetwarza przesłane pliki:
  - Wykonuje OCR w przypadku plików graficznych lub PDF bez warstwy tekstowej.
  - Analizuje treści za pomocą algorytmów AI, identyfikując kluczowe tematy i terminy.
  - Tworzy strukturę modułu edukacyjnego, który może obejmować:
    - Interaktywne lekcje.
    - Quizy i ćwiczenia.
    - Podsumowania w formie notatek.
    - Wizualizacje kluczowych zagadnień (np. diagramy, wykresy).
    - Pliki audio na podstawie treści tekstowych.
- System dostosowuje moduł do poziomu trudności użytkownika, bazując na:
  - Profilu użytkownika.
  - Wynikach wcześniejszych quizów i ćwiczeń.
  - Dodatkowych preferencjach użytkownika (np. styl nauki, język).

- Użytkownik otrzymuje powiadomienie o zakończeniu procesu wraz z przyciskiem **Przejdź do modułu**.
- Po kliknięciu moduł zostaje otwarty w nowym widoku w sekcji **Moje moduły**.

#### Scenariusz alternatywny:

- Przesłany plik jest w nieobsługiwanym formacie:
    - System wyświetla komunikat o błędzie i proponuje rozwiązania.
  - Materiał jest zbyt ogólny lub niewystarczający do stworzenia wartościowego modułu:
    - System wyświetla sugestię dodania bardziej szczegółowych treści lub wybranie dodatkowych zasobów z biblioteki systemu.
- 

## 4. Generowanie quizów

#### Scenariusz główny:

- Zalogowany użytkownik wybiera moduł edukacyjny z listy, otwierając widok szczegółów modułu.
- Klika sekcję **Quizy**.
- System proponuje ustawienia quizu w nowym oknie.
- Jeśli ustawienia nie odpowiadają użytkownikowi, klika przycisk **Edytuj ustawienia quizu** i dostosowuje:
  - Liczbę pytań.
  - Typ quizu (zamknięty, otwarty, wielokrotnego wyboru).
  - Poziom trudności.
- Użytkownik zatwierdza ustawienia, klikając przycisk **Rozpocznij quiz**.
- System generuje quiz na podstawie materiału edukacyjnego.
- Po zakończeniu quizu system:
  - Wyświetla wynik.
  - Identyfikuje obszary wymagające poprawy.
  - Proponuje dodatkowe materiały z przyciskiem **Zobacz propozycje**.

#### Scenariusz alternatywny:

- Brak danych wystarczających do stworzenia quizu:
  - System sugeruje przesłanie dodatkowych materiałów lub akceptację ograniczonej wersji quizu z przyciskiem **Generuj mimo to**.

---

## 5. Monitorowanie postępów użytkownika

### Scenariusz główny:

- Użytkownik wybiera sekcję **Postępy** z głównego menu, otwiera się widok raportów.
  - System wyświetla raporty w formie wykresów i tabel:
    - Wyniki quizów.
    - Czas nauki.
    - Progres w opanowaniu materiału.
  - Użytkownik może filtrować dane według okresu czasu lub tematyki, klikając przycisk **Filtruj**.
  - W przypadku wykrycia trudności system sugeruje materiały do poprawy.
- 

## 6. Zarządzanie użytkownikami i rolami

### Scenariusz główny:

- Administrator loguje się do panelu zarządzania, wybierając opcję **Panel zarządzania** z głównego menu.
- System otwiera widok administracyjny.
- Administrator ma możliwość wykonania następujących akcji:
  - Kliknięcie przycisku **Dodaj użytkownika** otwiera formularz, gdzie:
    - Administrator wprowadza dane nowego użytkownika, takie jak imię, nazwisko, adres e-mail, hasło oraz rolę użytkownika.
    - Po uzupełnieniu klika przycisk **Zatwierdź**.
  - Kliknięcie przycisku **Edycja użytkownika** otwiera listę użytkowników.
    - Administrator wybiera konkretnego użytkownika, którego dane chce edytować, wprowadza zmiany.
    - Po uzupełnieniu klika przycisk **Zatwierdź**.
  - Kliknięcie przycisku **Usuń użytkownika** otwiera okno dialogowe z listą użytkowników.
    - Administrator zaznacza użytkownika, klika **Usuń**, a system wyświetla komunikat potwierdzający akcję z przyciskiem **Tak** lub **Anuluj**.

- System zapisuje wszystkie zmiany w dzienniku zdarzeń, a administrator może przeglądać logi w sekcji **Historia działań**.

#### **Scenariusz alternatywny:**

- Próba wykonania operacji przez użytkownika bez uprawnień:
    - System blokuje dostęp i wyświetla komunikat o braku uprawnień z przyciskiem **Powrót** do głównego menu.
- 

## **7. Raporty dla instytucji edukacyjnych**

#### **Scenariusz główny:**

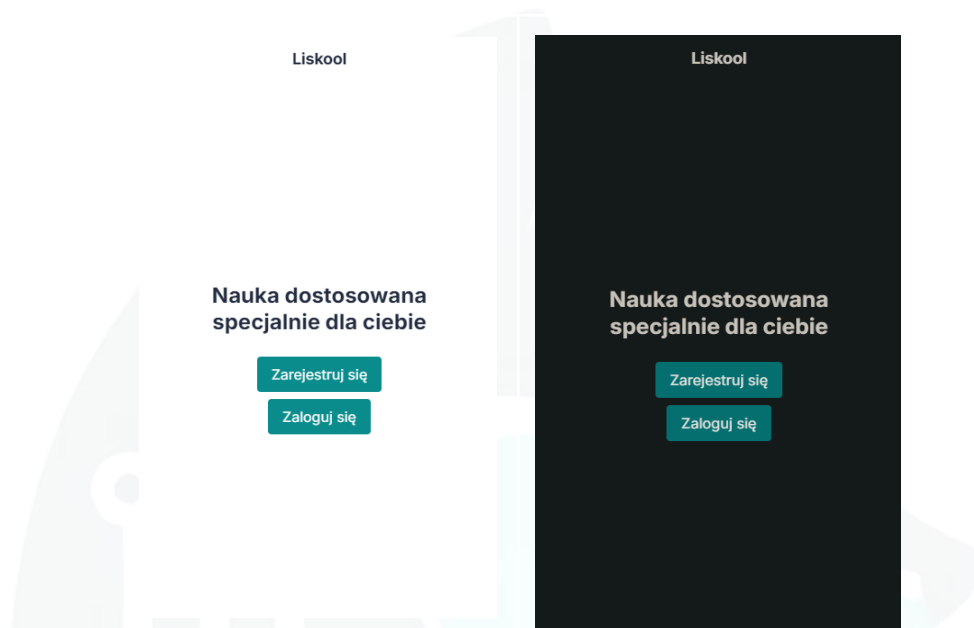
- Administrator instytucji loguje się do systemu i wybiera sekcję **Raporty grupowe** z głównego menu.
- System otwiera widok raportów, gdzie administrator może wybrać zakres danych:
  - Wybiera klasę, grupę lub użytkowników z listy rozwijanej.
  - Ustawia okres czasu, np. tydzień, miesiąc, rok, lub wybiera opcję **Zakres niestandardowy**, otwierając kalendarz do wyboru dat.
- Po ustawieniu kryteriów administrator klika przycisk **Generuj raport**.
- System przetwarza dane i prezentuje wyniki w formie:
  - Wykresów (np. wyniki klas).
  - Tabel (np. statystyki quizów).
  - Podsumowań tekstowych (np. efektywność nauki).
- Administrator może kliknąć przycisk **Eksportuj raport**, aby zapisać go w formacie PDF, Excel lub CSV.

#### **Scenariusz alternatywny:**

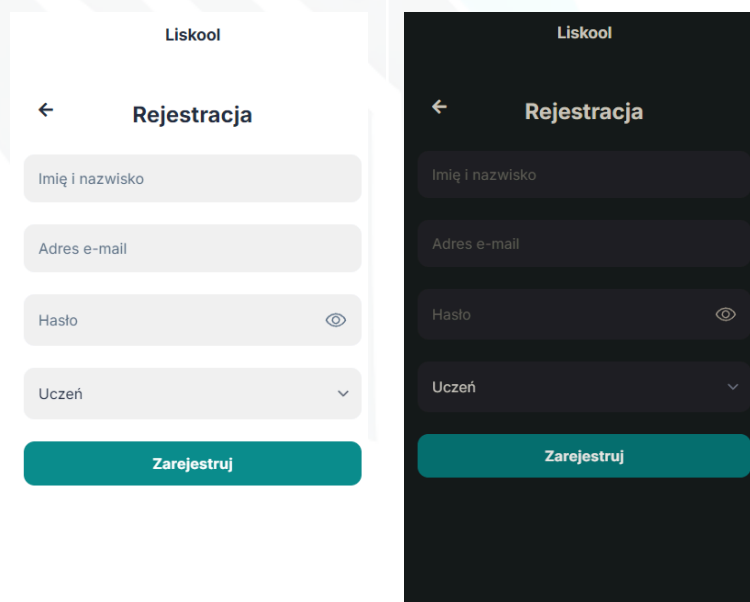
- Problemy z generowaniem raportu:
  - System wyświetla komunikat o błędzie z przyciskiem **Zgłoś problem** i linkiem do dokumentacji technicznej.
- Brak danych dla wybranych kryteriów:
  - System wyświetla komunikat **Brak danych dla wybranego zakresu** z przyciskiem **Zmień kryteria**.

## 2. Projekty ekranów

### Ekran startowy



### Po kliknięciu "Zarejestruj się"





## Po kliknięciu "Zaloguj się"

Liskool

← Logowanie

Adres e-mail

Hasło

[Zapomniałem hasła](#)

Zaloguj

Liskool

← Logowanie

Adres e-mail

Hasło

[Zapomniałem hasła](#)

Zaloguj

## Ekran przesłania materiałów

← Liskool

Prześlij materiał edukacyjny

Dodaj plik

Ustawienia zaawansowane

Prześlij

← Liskool

Prześlij materiał edukacyjny

Dodaj plik

Ustawienia zaawansowane

Prześlij

## Po kliknięciu "Ustawienia zaawansowane"

← Liskool

**Prześlij materiał edukacyjny**

[Dodaj plik](#)

**Ustawienia zaawansowane**

Preferowany język

Polski ▾

Poziom trudności

Łatwy ▾

Typ materiałów wynikowych

Notatki ▾

[Zastosuj ustawienia](#)

[Prześlij](#)

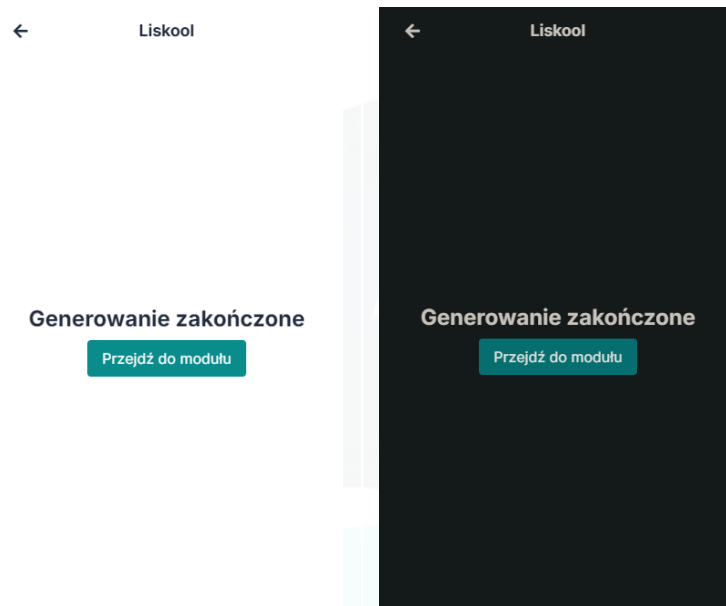
## Po przesłaniu materiałów

← Liskool

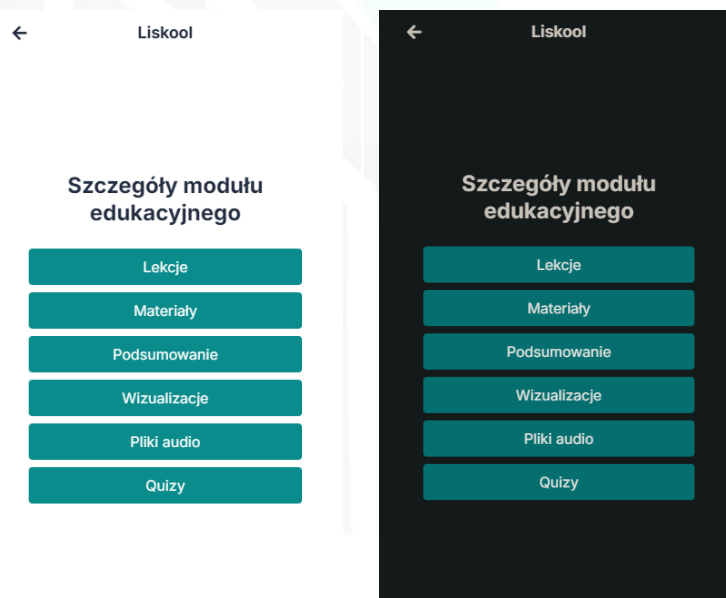
**Przesyłanie zakończone**

[Przejdź do materiału](#)

## Ekran po wygenerowaniu modułu



## Ekran szczegółów modułu



## Po kliknięciu "Quizy"

← Liskool

### Ustawienia quizu

Liczba pytań

10

Typ quizu

Zamknięty ▾

Poziom trudności

Łatwy ▾

Rozpocznij quiz

← Liskool

### Ustawienia quizu

Liczba pytań

10

Typ quizu

Zamknięty ▾

Poziom trudności

Łatwy ▾

Rozpocznij quiz

## Po wykonaniu quizu

← Liskool

### Wyniki quizu

Twój wynik: 8/10

Obszary wymagające poprawy:

- Temat 1
- Temat 2
- Temat 3

Zobacz propozycje

← Liskool

### Wyniki quizu

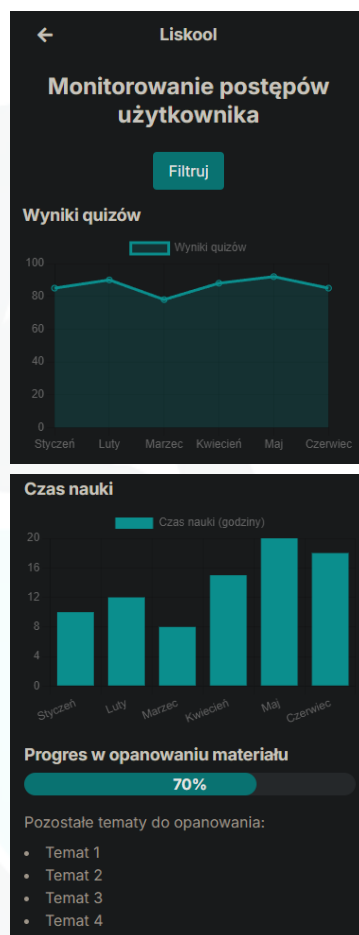
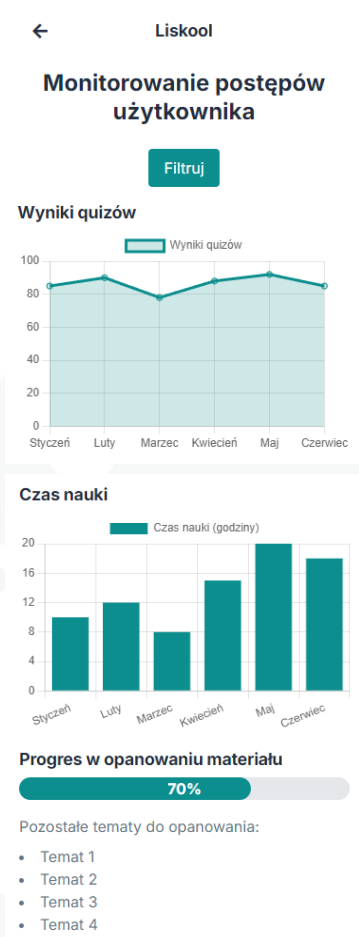
Twój wynik: 8/10

Obszary wymagające poprawy:

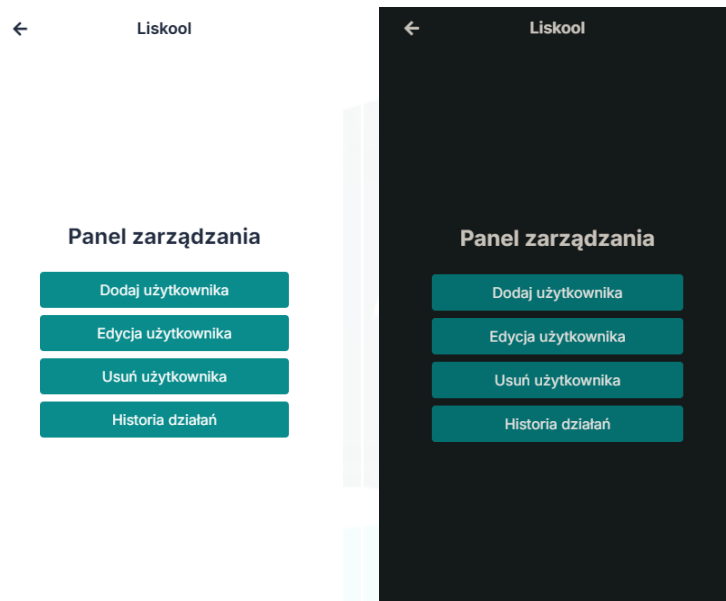
- Temat 1
- Temat 2
- Temat 3

Zobacz propozycje

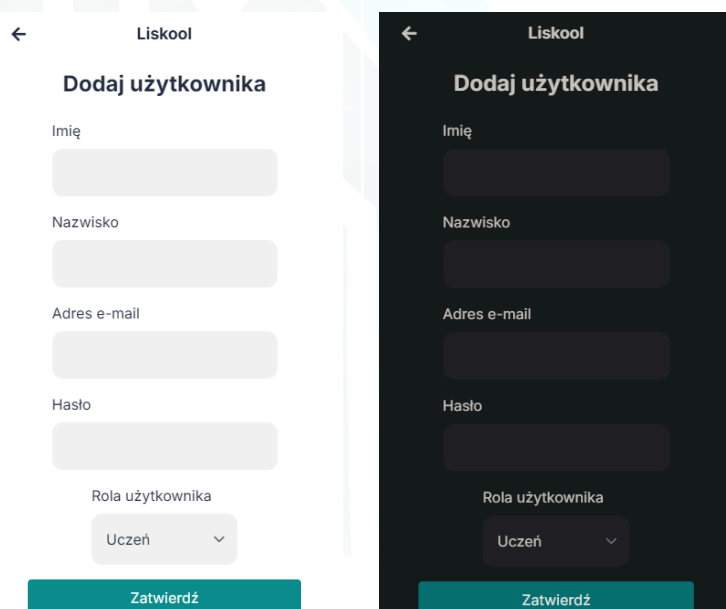
# Ekran monitorowania postępów



## Ekran zarządzania użytkownikami i rolami



## Po kliknięciu "Dodaj użytkownika"



## Po kliknięciu "Edycja użytkownika"

← Liskool

### Edycja użytkownika

Wyszukaj użytkownika

Wpisz nazwisko lub adres e-

Wybierz użytkownika

Użytkownik 1 ▾

Imię

Nazwisko

Adres e-mail

Rola użytkownika

Uczeń ▾

Zatwierdź

← Liskool

### Edycja użytkownika

Wyszukaj użytkownika

Wpisz nazwisko lub adres e-

Wybierz użytkownika

Użytkownik 1 ▾

Imię

Nazwisko

Adres e-mail

Rola użytkownika

Uczeń ▾

Zatwierdź

## Po kliknięciu "Usuń użytkownika"

← Liskool

### Usuń użytkownika

Wyszukaj użytkownika

Wpisz nazwisko lub adres e-

Wybierz użytkownika

Użytkownik 1 ▾

Usuń

← Liskool

### Usuń użytkownika

Wyszukaj użytkownika

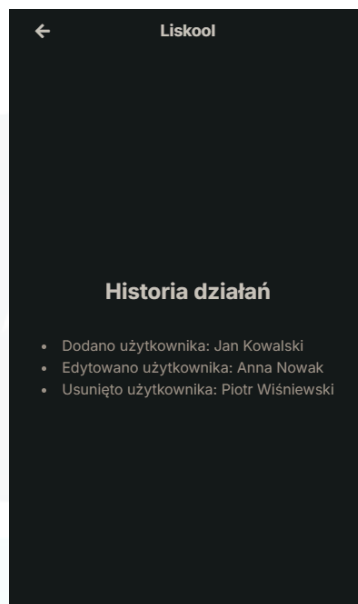
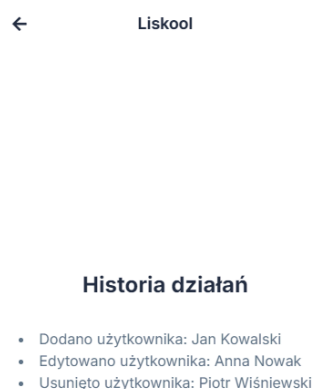
Wpisz nazwisko lub adres e-

Wybierz użytkownika

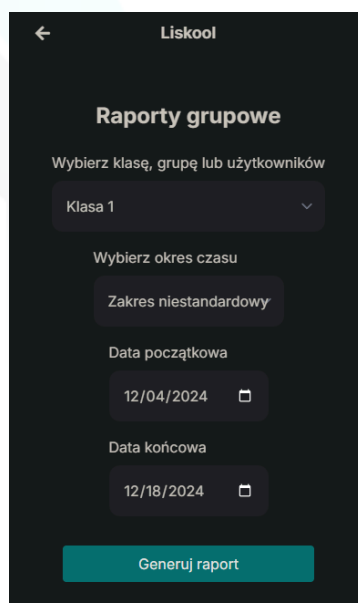
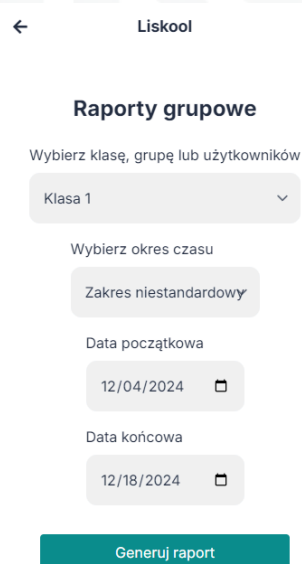
Użytkownik 1 ▾

Usuń

## Po kliknięciu "Historia działań"



## Ekran raportów grupowych





## Po kliknięciu "Generuj raport"



#### Statystyki quizów

Quiz	Średni wynik	Liczba uczestników
Quiz 1	85%	30
Quiz 2	78%	28
Quiz 3	92%	32

#### Efektywność nauki

Średnia efektywność nauki w wybranym okresie wynosi 85%.

Eksportuj raport



#### Statystyki quizów

Quiz	Średni wynik	Liczba uczestników
Quiz 1	85%	30
Quiz 2	78%	28
Quiz 3	92%	32

#### Efektywność nauki

Średnia efektywność nauki w wybranym okresie wynosi 85%.

Eksportuj raport

# 3. Architektura ogólna

## Komponenty:

### 1. Frontend (Aplikacja mobilna):

- **Technologia:** Kotlin/Java dla Androida
- **Funkcje:**
  - Interfejs użytkownika: logowanie/rejestracja, wybór kursów, dashboard z postępem nauki.
  - Wyświetlanie modułów edukacyjnych.
  - Obsługa interaktywnych ćwiczeń i quizów.
  - Synchronizacja danych z backendem, dodatkowo mechanizm buforowania danych lokalnie na urządzeniu użytkownika i synchronizacja po ponownym uzyskaniu dostępu do internetu.
  - Funkcje wspierające użytkowników z niepełnosprawnościami (np. VoiceOver).
- **Bezpieczeństwo danych użytkownika:**
  - Dodanie obsługi szyfrowania danych wrażliwych, np. klucze API, tokeny sesji.

### 2. Backend:

- **Technologia:** Python (Django/Flask/FastAPI).
- **Funkcje:**
  - Obsługa logiki biznesowej (np. przetwarzanie modułów edukacyjnych).
  - Monitorowanie kluczowych zdarzeń, np. logowań, ukończonych modułów edukacyjnych, błędów aplikacji.
  - Zarządzanie użytkownikami (profile, role).
  - API do komunikacji z aplikacją mobilną.
  - System kolejkowania zadań do obsługi czasochłonnych procesów, takich jak przetwarzanie danych.
  - Integracja z algorytmami AI.

### 3. Sztuczna Inteligencja (AI):

- **Moduły AI:**
  - Analiza treści (Natural Language Processing - NLP) do przekształcania materiałów dydaktycznych w moduły edukacyjne.
  - Silnik rekomendacji dostosowujący materiały do potrzeb ucznia.
  - Generowanie interaktywnych ćwiczeń na podstawie treści.

- Proces pozyskiwania i anonimizacji danych do uczenia algorytmów, z uwzględnieniem zasad RODO.
- Dynamiczne dostosowanie złożoności modeli w zależności od dostępnych zasobów sprzętowych.
- **Frameworki:** TensorFlow, PyTorch, spaCy, Hugging Face.

#### 4. Baza danych:

- **Technologia:** PostgreSQL (dla danych strukturalnych, np. użytkownicy, postępy), MongoDB (dla danych nieustrukturyzowanych, np. moduły edukacyjne).
- **Funkcje:**
  - Przechowywanie treści dydaktycznych.
  - Historia postępów uczniów.
  - Wyniki quizów i analiza danych.
  - Stosowanie szyfrowania danych w spoczynku (data-at-rest) oraz podczas transmisji (data-in-transit).

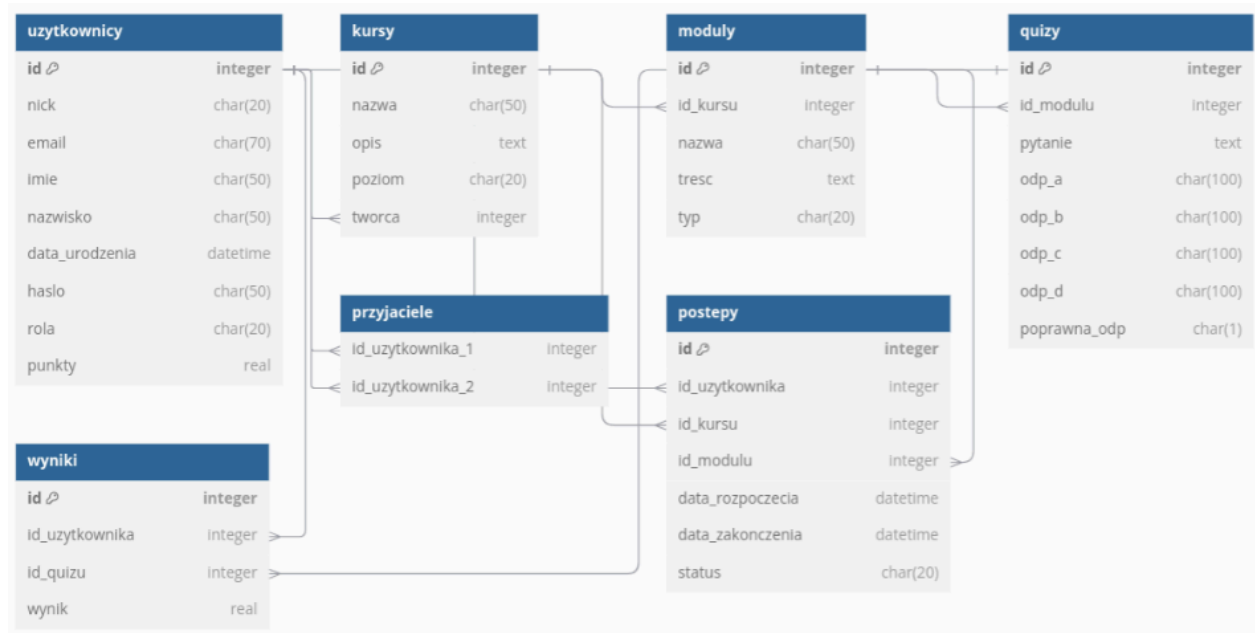
#### 5. Chmura i infrastruktura:

- **Chmura:** AWS/GCP/Azure.
- **Usługi:**
  - Przechowywanie plików: Amazon S3/Azure Blob.

#### 6. Testy i kontrola jakości:

- **Frontend:** Testy UI (np. Espresso), testy funkcjonalne.
- **Backend:** Testy jednostkowe (pytest), testy obciążeniowe (JMeter, Locust).
- **AI:** Walidacja jakości modeli (np. metryki dokładności, precyzji, odchylenia).
- **E2E:** Testy integracyjne, sprawdzające cały przepływ od użytkownika do przetworzenia w backendzie i zwrotu danych.

## Diagram bazy danych:



## 4. Główne zasady kodowania

Podczas tworzenia aplikacji LisKool będziemy korzystać z poniższych standardów.

### Pisanie kodu w Pythonie:

- **PEP 8 (Python):** Przy implementacji AI backendu w Pythonie trzymamy się standardów, takich jak użycie właściwej ilości spacji, długości linii, czy przejrzystych nazw zmiennych.
- **Nazwy przyjazne programistom:** Klasy, funkcje, zmienne i moduły powinny mieć nazwy jasno opisujące ich przeznaczenie, np. `generate_personalized_module` zamiast `genMod`. Konwencja *snake\_case* dla funkcji i zmiennych, a *Pascal/Case* dla klas.
- **Modularność:** Rozbijemy aplikację na moduły, np. oddzielne funkcje AI, API, bazy danych, obsługi quizów i frontend mobilny.

### Frontend mobilny (Java/Kotlin)

- **MVVM:** Używamy wzorca architektonicznego MVVM w aplikacji mobilnej dla łatwej separacji logiki od interfejsu użytkownika.
- **Spójny design:** Stosujemy design system (np. Material Design), aby aplikacja była intuicyjna i estetyczna.
- **Bezpieczeństwo:** Zabezpieczenie danych użytkownika za pomocą szyfrowania (np. AES do przechowywania danych lokalnych) oraz obsługa tokenów uwierzytelniających (np. OAuth 2.0).

### Dokumentacja i komentarze

- Dokumentację klas, metod i modułów w postaci **docstringów** w Pythonie.
- Wyraźne README.md dla całego projektu, aby każdy w zespole wiedział, jak uruchomić aplikację i jakie są jej założenia.
- Generujemy dokumentację API przy użyciu narzędzi takich jak Swagger lub Postman.

### Wersjonowanie kodu

- **Używamy Gita:** Kod na GitHubie lub GitLabie z jasno zdefiniowanymi gałęziami (`main`, `develop`, `feature/[nazwa_funkcji]`).
- **Commity:**
  - Stosujemy zwięzłe, opisowe nazwy commitów:

- `fix(login)`: poprawiono błędy uwierzytelniania.
- `feat(quiz)`: dodano obsługę nowego rodzaju pytania.
- **Code reviews**: Każda zmiana w kodzie powinna być przeglądana przez innego członka zespołu

## Wydajność i skalowalność

- **Optymalizacja algorytmów AI**:
  - W AI zadbamy o szybkie działanie algorytmów uczenia maszynowego, np. stosując wstępne przetwarzanie danych i narzędzia takie jak NumPy czy TensorFlow.
- **Caching**:
  - Wdrażamy cache dla często używanych danych, takich jak wyniki quizów czy szczegóły użytkowników.
- **Asynchroniczność**:
  - Backend powinien obsługiwać asynchroniczne operacje (np. przy obsłudze quizów lub analizy danych użytkownika).

## 5. Identyfikacja i zasady zarządzania ryzykiem

### 1. Możliwość przeciążenia serwerów

- **Opis ryzyka:**  
W przypadku nagłego wzrostu liczby użytkowników, np. po intensywnej kampanii promocyjnej, serwery mogą zostać przeciążone, co spowoduje problemy z dostępem do aplikacji. Może to skutkować negatywnymi opiniami użytkowników i utratą reputacji platformy.
- **Prawdopodobieństwo:** Średnie.
- **Środki zaradcze:**
  - Skalowalna infrastruktura (np. korzystanie z rozwiązań chmurowych takich jak AWS, Google Cloud, Azure).
  - Wprowadzenie mechanizmów balansowania obciążenia (load balancers).
  - Implementacja mechanizmów cache'owania.

### 2. Niedotrzymanie terminów realizacji projektu

- **Opis ryzyka:**  
Projekt może zostać opóźniony w wyniku złego zarządzania czasem, nieoczekiwanych problemów technicznych lub braku zasobów (np. odpowiedniej liczby programistów lub specjalistów AI).
- **Prawdopodobieństwo:** Średnie.
- **Środki zaradcze:**
  - Regularne sprinty (metodologia Agile) i przeglądy postępów prac.
  - Jasny podział zadań w zespole i użycie narzędzi do zarządzania projektami (np. Jira, Trello).
  - Analiza ryzyka technicznego na początku każdego sprintu.

### 3. Problemy z bezpieczeństwem danych użytkowników

- **Opis ryzyka:**  
Wyciek danych, ataki hakerskie lub słabe zabezpieczenia mogą spowodować utratę zaufania użytkowników i konsekwencje prawne.
- **Prawdopodobieństwo:** Niskie/Średnie.
- **Środki zaradcze:**
  - Szyfrowanie danych wrażliwych (np. hasła za pomocą bcrypt).

- Regularne audyty bezpieczeństwa i testy penetracyjne.
- Implementacja mechanizmów ochrony przed atakami DDoS.
- Wdrożenie uwierzytelniania wieloskładnikowego (MFA).

#### 4. Niska akceptacja użytkowników (UX/UI)

- **Opis ryzyka:**  
Aplikacja może nie spełniać oczekiwań użytkowników pod względem intuicyjności interfejsu lub spersonalizowanego podejścia do nauki.
- **Prawdopodobieństwo:** Średnie.
- **Środki zaradcze:**
  - Testy użytkowników w trakcie procesu projektowania (UX Research).
  - Regularne zbieranie opinii użytkowników i wdrażanie poprawek.
  - Wdrożenie uniwersalnych standardów projektowych (np. Material Design).
  - Dostosowanie aplikacji do różnych poziomów zaawansowania użytkowników.

#### 5. Problemy z adaptacją algorytmów AI

- **Opis ryzyka:**  
Algorytmy sztucznej inteligencji mogą nie działać zgodnie z założeniami, np. błędnie dobierać treści edukacyjne lub generować nieadekwatne ćwiczenia.
- **Prawdopodobieństwo:** Średnie/Wysokie.
- **Środki zaradcze:**
  - Wstępne testy na dobrze zbalansowanych zbiorach danych treningowych.
  - Ciągłe uczenie się algorytmu na podstawie nowych danych.
  - Monitorowanie wyników algorytmów i ręczna weryfikacja wyników w początkowych etapach wdrożenia.
  - Regularne aktualizacje modeli w oparciu o feedback użytkowników.

#### 6. Problemy z integracją systemów (backend, frontend, AI)

- **Opis ryzyka:**  
Może wystąpić problem z płynną komunikacją między warstwami systemu, co wpłynie na stabilność aplikacji.
- **Prawdopodobieństwo:** Średnie.
- **Środki zaradcze:**



- Jasno zdefiniowane API i formaty wymiany danych (np. JSON, REST API).
- Regularne testy integracyjne.
- Wprowadzenie pipeline'u CI/CD w celu automatycznego sprawdzania kompatybilności kodu.

## 7. Problemy związane z utrzymaniem infrastruktury

- **Opis ryzyka:** Problemy z serwerami, brak aktualizacji oprogramowania lub niewystarczające monitorowanie mogą prowadzić do awarii aplikacji.
- **Prawdopodobieństwo:** Niskie/Średnie.
- **Środki zaradcze:**
  - Monitorowanie stanu aplikacji i serwerów.
  - Regularne aktualizacje bibliotek i frameworków.
  - Wdrożenie systemu alertów i automatycznego restartu usług w przypadku awarii.

## 6. Zgodność prac z tablicą koncepcyjną i specyfikacją

Po dokonaniu analizy wykonanych prac oraz porównaniu ich z planem przedstawionym w tabeli koncepcyjnej oraz specyfikacji wymagań, możemy stwierdzić, że uzyskane wyniki są satysfakcjonujące.

Obecnie nie przewidujemy konieczności wprowadzania większych zmian w stosunku do tego, co zostało zawarte w tabeli koncepcyjnej i specyfikacji wymagań.

