

Faza Konstrukcji Systemu Liskool

Viktoria Kashpruk, Cezary Miłek

Testy funkcjonalne.....	2
Test: Przesyłanie materiałów edukacyjnych.....	2
Test: Generowanie quizów.....	3
Test: Monitorowanie postępów.....	5
Pomiary dla wymagań niefunkcjonalnych.....	7
Intuicyjność interfejsu użytkownika.....	7
Skalowalność.....	8
Wydajność.....	8
Bezpieczeństwo.....	9
Wielojęzyczność.....	10
Uniwersalność.....	10
Plan beta testowania.....	12
Faza 1: Testowanie podstawowych funkcjonalności przez wewnętrznego testera.....	12
Faza 2: Testy z udziałem zewnętrznych użytkowników.....	13
Faza 3: Finalne testy funkcjonalne przed premierą.....	13
Dokumentacja i narzędzia wspierające testy.....	14
Kluczowe wskaźniki sukcesu testów beta.....	14
Plan wykonania produktu.....	16
Fazy projektu.....	16
Faza 1: Stworzenie prototypu aplikacji.....	16
Faza 2: Zbieranie materiałów i testowanie.....	17
Faza 3: Testowanie we wczesnym dostępie.....	18
Faza 4: Dalszy rozwój i wdrożenie.....	19
Ocena zgodności wykonanych prac z wizją systemu i specyfikacją wymagań.....	20

Testy funkcjonalne

Test: Przesyłanie materiałów edukacyjnych

Szczegóły testu:

1. Użytkownik uruchamia aplikację i loguje się na swoje konto.
2. Użytkownik otwiera panel "*Prześlij materiał*" w głównym menu.
3. Klika przycisk **Dodaj plik**, co otwiera okno wyboru pliku.
4. Wybiera plik "*matematyka_wprowadzenie.pdf*" o rozmiarze 4 MB z urządzenia.
5. Opcjonalnie wchodzi w ustawienia zaawansowane:
 - **Wybiera język:** *Polski*.
 - Ustawia **typ materiałów wynikowych:** *Quiz i notatki*.
6. Klika przycisk **Prześlij**, co uruchamia proces przetwarzania pliku.
7. System wykonuje analizę pliku:
 - Wykonuje OCR, jeśli to konieczne.
 - Tworzy moduły edukacyjne na podstawie treści.
8. Po zakończeniu przetwarzania system wyświetla komunikat o sukcesie i przycisk **Przejdź do materiału**.
9. Użytkownik klika przycisk **Przejdź do materiału** i uzyskuje dostęp do wygenerowanych modułów w sekcji "*Moje moduły*".

Numer kroku	Opis akcji	Oczekiwania
1	Użytkownik uruchamia aplikację i loguje się na swoje konto.	Użytkownik zostaje przeniesiony do głównego menu aplikacji.
2	Użytkownik otwiera panel " <i>Prześlij materiał</i> " w głównym menu.	Wyświetla się widok panelu z przyciskiem Dodaj plik i opcjami ustawień zaawansowanych.

Numer kroku	Opis akcji	Oczekiwania
3	Klika przycisk Dodaj plik , co otwiera okno wyboru pliku.	Wyświetla się okno wyboru pliku na urządzeniu.
4	Wybiera plik "matematyka_wprowadzenie.pdf" o rozmiarze 4 MB.	Nazwa wybranego pliku pojawia się w interfejsie aplikacji.
5	Wchodzi w ustawienia zaawansowane i ustawia preferencje.	Wybrane opcje (język: Polski, typ: Quiz i notatki) zostają zapisane.
6	Klika przycisk Prześlij , uruchamiając proces przetwarzania.	System rozpoczyna analizę pliku i wyświetla wskaźnik postępu.
7	System przetwarza plik, wykonuje OCR i tworzy moduły edukacyjne.	Przetwarzanie kończy się sukcesem lub wyświetla komunikat o błędzie (np. nieobsługiwany format).
8	Po zakończeniu przetwarzania wyświetla się komunikat i przycisk Przejdź do materiału .	Użytkownik widzi komunikat "Przesyłanie zakończone".
9	Użytkownik klika Przejdź do materiału i otwiera wygenerowany moduł.	Moduł edukacyjny jest dostępny w sekcji "Moje moduły" i można go przeglądać.

Test: Generowanie quizów

Szczegóły testu:

1. Użytkownik uruchamia aplikację i loguje się na swoje konto.
2. Użytkownik otwiera sekcję "Moje moduły" z głównego menu.
3. Wybiera moduł edukacyjny o nazwie "Podstawy matematyki".
4. Klika sekcję **Quizy** w szczegółach wybranego modułu.
5. System wyświetla domyślne ustawienia quizu:
 - **Liczba pytań:** 10.
 - **Typ:** Zamknięte pytania z wielokrotnym wyborem.

6. Użytkownik opcjonalnie dostosowuje ustawienia, np. zmienia liczbę pytań na 15.
7. Klika przycisk **Rozpocznij quiz**, co uruchamia proces generowania pytań.
8. System analizuje treść modułu, generuje 15 pytań z czterema opcjami odpowiedzi.
9. Po zakończeniu quizu system wyświetla wynik użytkownika oraz listę zagadnień wymagających poprawy.
10. Wynik quizu zapisuje się w sekcji "*Postępy*", a użytkownik może wrócić do szczegółów modułu.

Numer kroku	Opis akcji	Oczekiwania
1	Użytkownik uruchamia aplikację i loguje się na swoje konto.	Użytkownik zostaje przeniesiony do głównego menu aplikacji.
2	Użytkownik otwiera sekcję "Moje moduły" w głównym menu.	Wyświetla się lista modułów edukacyjnych użytkownika.
3	Wybiera moduł edukacyjny o nazwie " <i>Podstawy matematyki</i> ".	Otwiera się widok szczegółów modułu z opcjami, w tym sekcją Quizy .
4	Klika sekcję Quizy .	System wyświetla domyślne ustawienia quizu (liczba pytań, typ pytań).
5	System wyświetla domyślne ustawienia quizu.	Domyślna liczba pytań wynosi 10, typ : <i>zamknięte z wielokrotnym wyborem</i> .
6	Użytkownik dostosowuje ustawienia quizu (np. zmienia liczbę pytań na 15).	Wybrane ustawienia zostają zapisane w interfejsie aplikacji.
7	Klika przycisk Rozpocznij quiz , uruchamiając proces generowania.	System rozpoczyna analizę modułu i generuje pytania na podstawie jego treści.
8	System generuje quiz na podstawie modułu.	Pytania są zgodne z materiałem, każde zawiera 4 opcje odpowiedzi (z czego jedna poprawna).

Numer kroku	Opis akcji	Oczekiwania
9	Użytkownik kończy quiz, a system wyświetla wynik i listę zagadnień do poprawy.	Wynik prezentowany jest w procentach, a lista wskazuje trudne zagadnienia.
10	Wynik quizu zapisuje się w sekcji "Postępy".	Wynik widoczny w sekcji "Postępy", a użytkownik może wrócić do szczegółów modułu.

Test: Monitorowanie postępów

Szczegóły testu:

1. Użytkownik uruchamia aplikację i loguje się na swoje konto.
2. Użytkownik otwiera sekcję **Postępy** z głównego menu.
3. W sekcji **Postępy** użytkownik wybiera okres czasu: "Ostatni tydzień".
4. Użytkownik ustawia filtr "Wszystkie moduły".
5. Klika przycisk **Filtruj**, aby zaktualizować widok raportu.
6. System generuje raport, który zawiera dane:
 - Wyniki quizów w wybranym okresie.
 - Czas poświęcony na naukę.
 - Progres w opanowywaniu materiałów edukacyjnych.
7. Użytkownik przegląda raport przedstawiony w formie wykresów i tabel.
8. Użytkownik klika przycisk **Eksportuj**, aby zapisać raport w formacie PDF.
9. System zapisuje plik na urządzeniu użytkownika w wybranej lokalizacji.

Numer kroku	Opis akcji	Oczekiwania
1	Użytkownik uruchamia aplikację i loguje się na swoje konto.	Użytkownik zostaje przeniesiony do głównego menu aplikacji.

Numer kroku	Opis akcji	Oczekiwania
2	Użytkownik otwiera sekcję Postępy w głównym menu.	Wyświetla się widok raportów z opcjami filtrowania i eksportu danych.
3	Użytkownik wybiera okres czasu "Ostatni tydzień".	Pole wyboru czasu zostaje zaktualizowane.
4	Użytkownik ustawia filtr "Wszystkie moduły".	Pole wyboru modułów zostaje zaktualizowane.
5	Klika przycisk Filtruj .	System przetwarza dane i aktualizuje widok raportu zgodnie z wybranymi kryteriami.
6	System generuje raport z wynikami quizów, czasu nauki i progresu.	Dane raportu są wyświetlane w formie wykresów i tabel.
7	Użytkownik przegląda raport na ekranie.	Raport zawiera przejrzyste informacje zgodne z wybranymi kryteriami.
8	Użytkownik klika przycisk Eksportuj .	System wyświetla okno wyboru lokalizacji dla zapisywanego pliku.
9	System zapisuje raport w formacie PDF na urządzeniu.	Plik PDF z raportem jest zapisany w wybranej lokalizacji i zawiera wszystkie dane raportu.

Pomiary dla wymagań niefunkcjonalnych

Intuicyjność interfejsu użytkownika

Opis wymagania: Użytkownik powinien łatwo poruszać się po aplikacji dzięki intuicyjnej nawigacji, czytelnym ikonom i opisom.

- **Metryki:**

- **Wskaźnik zrozumienia interfejsu:** $X = \frac{P}{T}$, gdzie:
 - P : liczba przypadków, w których użytkownik zgubił się lub potrzebował pomocy,
 - T : całkowity czas testów (w minutach).
- **Liczba błędnych kliknięć:** średnia liczba błędnych operacji wykonanych przez użytkownika w trakcie testów.
- **Czas wykonania kluczowego zadania:** średni czas realizacji operacji, np. przesłanie materiału lub rozpoczęcie quizu.

- **Metody pomiaru:**

- Testy użyteczności z udziałem co najmniej 20 użytkowników reprezentujących różne role (uczeń, nauczyciel, administrator).
- Obserwacja sesji użytkowników podczas realizacji typowych zadań.
- Ankiety satysfakcji użytkowników (w skali 1-5).

- **Kryteria akceptacji:**

- $X < 0.01$.
 - Średni czas realizacji kluczowych zadań: maksymalnie 10 sekund.
 - Poziom satysfakcji: minimum 4.0 w skali 1-5.
-

Skalowalność

Opis wymagania: System powinien obsługiwać do 10 000 użytkowników jednocześnie z możliwością dynamicznego skalowania.

- **Metryki:**
 - **Maksymalna liczba równoczesnych użytkowników:** testowana na poziomie 10 000 użytkowników.
 - **Czas odpowiedzi serwera:** średni czas odpowiedzi na żądania API (*ms*).
 - **Średnie obciążenie zasobów:** procent wykorzystania CPU, pamięci RAM oraz przepustowości sieci.
 - **Metody pomiaru:**
 - Testy obciążeniowe przy użyciu narzędzi takich jak JMeter, Locust.
 - Monitorowanie infrastruktury (AWS, Google Cloud) podczas testów.
 - Analiza logów serwera dla okresów maksymalnego obciążenia.
 - **Kryteria akceptacji:**
 - System obsługuje 10 000 równoczesnych użytkowników bez zauważalnego spadku wydajności.
 - Czas odpowiedzi serwera: $< 500\text{ ms}$ przy maksymalnym obciążeniu.
 - Obciążenie zasobów: $< 80\%$ dla CPU i pamięci.
-

Wydajność

Opis wymagania: Czas odpowiedzi na przesłanie materiałów:

- Dla plików $< 5\text{ MB}$: maksymalnie 10 sekund.
- Dla plików $> 5\text{ MB}$, ale $< 10\text{ MB}$: maksymalnie 20 sekund.
- Dla plików $> 10\text{ MB}$: maksymalnie 30 sekund.
- **Metryki:**
 - **Czas przetwarzania plików:** średni czas przesłania pliku do aplikacji.

- **Wskaźnik błędów przetwarzania:** procent operacji zakończonych błędem.
 - **Metody pomiaru:**
 - Testy przesyłania plików o różnych rozmiarach i formatach (PDF, DOCX, JPEG).
 - Automatyczne rejestrowanie czasu przetwarzania w logach systemu.
 - **Kryteria akceptacji:**
 - Czas przetwarzania zgodny z wymaganiami.
 - Wskaźnik błędów przetwarzania: poniżej 0.5%.
-

Bezpieczeństwo

Opis wymagania: Dane użytkowników są szyfrowane (AES-256), a połączenia odbywają się poprzez HTTPS. Dwupoziomowa autoryzacja dla użytkowników o podwyższonych uprawnieniach.

- **Metryki:**
 - **Wskaźnik naruszeń bezpieczeństwa:** liczba zgłoszonych incydentów.
 - **Czas reakcji na incydenty:** średni czas od zgłoszenia problemu do jego rozwiązania.
 - **Pokrycie testami bezpieczeństwa:** procent funkcji przetestowanych pod kątem bezpieczeństwa.
- **Metody pomiaru:**
 - Testy penetracyjne z udziałem zespołów bezpieczeństwa.
 - Symulacje ataków DDoS.
 - Automatyczne sprawdzanie kodu pod kątem luk (np. SonarQube).
- **Kryteria akceptacji:**
 - Brak naruszeń bezpieczeństwa podczas testów.
 - Pokrycie testami bezpieczeństwa: 100%.
 - Szyfrowanie danych potwierdzone audytem.

Wielojęzyczność

Opis wymagania: System obsługuje język polski i angielski z możliwością dodawania nowych języków.

- **Metryki:**
 - **Liczba wspieranych języków:** liczba przetłumaczonych elementów interfejsu.
 - **Wskaźnik pokrycia tłumaczeń:** procent przetłumaczonych elementów interfejsu.
 - **Metody pomiaru:**
 - Automatyczne testy poprawności tłumaczeń.
 - Ankiety wśród użytkowników dla oceny jakości tłumaczeń.
 - **Kryteria akceptacji:**
 - Wskaźnik pokrycia tłumaczeń: 100% dla języka polskiego i angielskiego.
 - Poziom satysfakcji z tłumaczeń: minimum 4.5 w skali 1-5.
-

Uniwersalność

Opis wymagania: System działa na urządzeniach z systemami Android 8.0 i wyższymi.

- **Metryki:**
 - **Wskaźnik zgodności:** $X = \frac{A}{N}$, gdzie:
 - A : liczba funkcji działających na urządzeniach z Androidem,
 - N : całkowita liczba funkcji systemu.
 - **Średni czas uruchamiania aplikacji:** czas od kliknięcia ikony do pełnego załadowania.
- **Metody pomiaru:**
 - Testy manualne i automatyczne na urządzeniach z Androidem.
 - Symulacje różnych środowisk w emulatorach.
- **Kryteria akceptacji:**

- $X = 1.0$ (wszystkie funkcje działają poprawnie).
 - Czas uruchamiania: < 5 sekund.
-



Plan beta testowania

Wstępne założenia

Celem testów beta jest weryfikacja poprawności działania systemu, wykrycie błędów w implementacji oraz ocena intuicyjności i funkcjonalności z perspektywy użytkownika końcowego. Proces testowania zostanie podzielony na trzy fazy, z uwzględnieniem stopniowego zwiększania liczby testerów oraz zakresu funkcjonalności.

Faza 1: Testowanie podstawowych funkcjonalności przez wewnętrznego testera

Czas trwania: 2 tygodnie

Zakres testów:

- Rejestracja i logowanie użytkowników.
- Przesyłanie materiałów edukacyjnych w różnych formatach (PDF, DOCX, JPEG itp.).
- Generowanie prostych modułów edukacyjnych i quizów.
- Podstawowe monitorowanie postępów użytkowników.

Cele:

- Wykrycie błędów krytycznych w podstawowych funkcjach systemu.
- Ocena stabilności systemu na wczesnym etapie.

Działania:

- Testy przeprowadza wewnętrzny tester z pełnym dostępem do systemu.
 - Błędy i uwagi dokumentowane w narzędziu do zarządzania projektami (np. Jira).
-

Faza 2: Testy z udziałem zewnętrznych użytkowników

Czas trwania: 3 tygodnie

Zakres testów:

- Wszystkie funkcjonalności z fazy 1.
- Tworzenie zaawansowanych modułów edukacyjnych z dostosowaniem do preferencji użytkownika (np. poziom trudności, język).
- Testowanie systemu przez grupę użytkowników reprezentujących różne role (uczniowie, nauczyciele, rodzice, administratorzy).
- Analiza intuicyjności interfejsu i ogólnej wydajności systemu.

Cele:

- Identyfikacja problemów związanych z użytecznością i wydajnością przy większym obciążeniu.
- Zbieranie opinii użytkowników końcowych dotyczących UX/UI.

Działania:

- Wyselekcjonowanie grupy 50 użytkowników spośród wolontariuszy.
- Zbieranie opinii za pomocą ankiet oraz sesji testowych z nagrywaniem ekranów.
- Monitorowanie logów i analizowanie zgłoszonych błędów.

Faza 3: Finalne testy funkcjonalne przed premierą

Czas trwania: 2 tygodnie

Zakres testów:

- Przegląd wszystkich funkcji systemu, w tym generowania raportów, zarządzania użytkownikami i integracji z aplikacją mobilną.

- Testy przeciążeniowe i wydajnościowe przy symulacji ruchu do 10 000 użytkowników jednocześnie.
- Weryfikacja kompatybilności systemu z różnymi urządzeniami i wersjami Androida oraz przeglądarkami.

Cele:

- Ostateczna weryfikacja zgodności systemu z wymaganiami funkcjonalnymi i нефункциональными.
- Upewnienie się, że system jest gotowy do wdrożenia w środowisku produkcyjnym.

Działania:

- Zaangażowanie wszystkich uczestników poprzednich faz testów.
- Przeprowadzenie testów w środowisku jak najbardziej zbliżonym do produkcyjnego.
- Przygotowanie raportu końcowego i lista poprawek do zaimplementowania przed wdrożeniem.

Dokumentacja i narzędzia wspierające testy

- Platforma do zarządzania zgłoszeniami błędów: Jira.
- Narzędzia do rejestrowania działań użytkowników: Hotjar.
- Ankiety opinii: Formularze Google.
- Symulacje obciążenia: Apache JMeter, Locust.

Kluczowe wskaźniki sukcesu testów beta

- Liczba błędów krytycznych zgłoszonych i rozwiązanych w fazach testów.
- Poziom zadowolenia użytkowników końcowych (ocena $> \frac{4}{5}$ w ankietach).

- Czas odpowiedzi systemu na przesyłanie materiałów i generowanie modułów edukacyjnych zgodny z wymaganiami нефункциональными.



Plan wykonania produktu

Fazy projektu

Nr	Nazwa fazy	Czas trwania	Początek	Koniec
1	Stworzenie prototypu aplikacji	3 miesiące	2025-01-15	2025-04-14
2	Zbieranie materiałów i testowanie	3 miesiące	2025-04-15	2025-07-14
3	Testowanie we wczesnym dostępie	3 miesiące	2025-07-15	2025-10-14
4	Dalszy rozwój i wdrożenie	3 miesiące	2025-10-15	2026-01-14

Faza 1: Stworzenie prototypu aplikacji

Nr	Zadanie	Czas trwania	Początek	Koniec
1	Wstępne spotkanie z zespołem	2 dni	2025-01-15	2025-01-16
2	Sfinalizowanie tablicy koncepcyjnej	5 dni	2025-01-17	2025-01-21
3	Projektowanie bazy danych	10 dni	2025-01-22	2025-01-31

Nr	Zadanie	Czas trwania	Początek	Koniec
4	Projektowanie API backendu	12 dni	2025-02-01	2025-02-12
5	Projektowanie interfejsu użytkownika	15 dni	2025-02-13	2025-02-27
6	Implementacja modułu przesyłania materiałów	20 dni	2025-02-28	2025-03-19
7	Weryfikacja architektury i testy integracyjne	15 dni	2025-03-20	2025-04-04
8	Analiza i optymalizacja prototypu	10 dni	2025-04-05	2025-04-14

Faza 2: Zbieranie materiałów i testowanie

Nr	Zadanie	Czas trwania	Początek	Koniec
1	Rozwój interfejsu przesyłania materiałów	15 dni	2025-04-15	2025-04-29
2	Implementacja modułu generowania modułów edukacyjnych	20 dni	2025-04-30	2025-05-19

Nr	Zadanie	Czas trwania	Początek	Koniec
3	Rozwój interfejsu zarządzania użytkownikami	15 dni	2025-05-20	2025-06-03
4	Optymalizacja algorytmów AI	15 dni	2025-06-04	2025-06-18
5	Testy wydajności i funkcjonalności aplikacji	15 dni	2025-06-19	2025-07-03
6	Usprawnienia oparte na testach	11 dni	2025-07-04	2025-07-14

Faza 3: Testowanie we wczesnym dostępie

Nr	Zadanie	Czas trwania	Początek	Koniec
1	Testy użytkowników wewnętrznych	14 dni	2025-07-15	2025-07-28
2	Testy beta z udziałem użytkowników zewnętrznych	30 dni	2025-07-29	2025-08-27
3	Zbieranie informacji zwrotnej i analiza wyników	15 dni	2025-08-28	2025-09-11

Nr	Zadanie	Czas trwania	Początek	Koniec
4	Iteracyjne poprawki na podstawie feedbacku	15 dni	2025-09-12	2025-09-26
5	Końcowe testy funkcjonalne i stabilności	18 dni	2025-09-27	2025-10-14

Faza 4: Dalszy rozwój i wdrożenie

Nr	Zadanie	Czas trwania	Początek	Koniec
1	Publikacja systemu na platformie produkcyjnej	14 dni	2025-10-15	2025-10-28
2	Monitoring i szybkie poprawki po wdrożeniu	21 dni	2025-10-29	2025-11-18
3	Przygotowanie infrastruktury serwerowej i zabezpieczeń	20 dni	2025-11-19	2025-12-08
4	Optymalizacja wydajności systemu	20 dni	2025-12-09	2025-12-28
5	Finalne testy systemu	17 dni	2025-12-29	2026-01-14

Ocena zgodności wykonanych prac z wizją systemu i specyfikacją wymagań

Wykonane prace w pełni odpowiadają wizji systemu oraz wymaganiom określonym w dokumentach projektowych. Zrealizowano główne funkcjonalności, takie jak przesyłanie materiałów, generowanie modułów edukacyjnych oraz monitorowanie postępów użytkowników.

System spełnia kluczowe wymagania niefunkcjonalne, w tym intuicyjność interfejsu, wydajność oraz kompatybilność z urządzeniami mobilnymi.

Rekomenduje się poszerzenie zakresu testów beta oraz dalsze doskonalenie algorytmów AI na podstawie wyników z testów.

