# Projekt Zespołowy 2 Grupa 4M

Spis treści

1 Wprowadzenie. 2

1.1 Cel projektu. 2

1.2 Wstępna wizja projektu. 2

2 Metodologia wytwarzania. 2

3 Analiza wymagań. 2

3.1 Wymagania użytkownika i biznesowe. 2

3.2 Wymagania funkcjonalne i niefunkcjonalne. 2

3.3 Przypadki użycia. 2

3.4 Potwierdzenie zgodności wymagań. 2

4 Definicja architektury. 3

5 Dane trwałe. 3

5.1 Model logiczny danych. 3

5.2 Przetwarzanie i przechowywanie danych. 3

6 Specyfikacja analityczna i projektowa. 3

7 Projekt standardu interfejsu użytkownika. 3

8 Specyfikacja testów.. 3

*9* *Wirtualizacja/konteneryzacja*. 4

10 Bezpieczeństwo. 4

11 Podręcznik użytkownika. 4

12 Podręcznik administratora. 4

13 Podsumowanie. 4

14 Bibliografia. 4

# **1** **Wprowadzenie**

## **1.1** **Cel projektu**

Celem projektu jest opracowanie aplikacji umożliwiającej zarządzanie sesjami burzy

mózgów. Aplikacja powinna odnotowywać start i koniec sesji, temat sesji, uczestników,

listę pomysłów. Aplikacja będzie dokonywała transkrypcji sesji, w trybie strumieniowym, i zapamiętywała ją w rekordzie sesji. Również pożądane

jest, aby aplikacja potrafiła dokonać podsumowania ustaleń z sesji (z wykorzystaniem

modeli językowych).

Główną motywacją do projektu jest w przyszłości możliwość analizy dynamiki pracy

zespołu w sesji kreatywnej i dzięki temu lepsze zarządzanie pracą kreatywną.

## **1.2** **Wstępna wizja projektu**

Nasza wizja projektu przewiduje aplikację desktopową z serwerem bazy danych. Użytkownik za jej pomocą po zalogowaniu się będzie mógł przeglądać wszystkie spotkania prywatne jak i grup do których należy uzyskując takie dane jak transkrypcja i podsumowanie spotkania

# **2** **Metodologia wytwarzania**

Projekt tworzymy w metodologii zwinnej. Spotkania zespołowe są zorganizowane w iteracjach. Spotkania przeprowadzamy 2 razy w tygodniu zdalnie na platformie Discord.

Stała komunikacja z właścicielem projektu za pomocą platformy Teams. Wtedy precyzujemy cele projektu, definiujemy niezbędne wymagania i pokazujemy

Organizacja pracy w projekcie

# **3** **Analiza wymagań**

## **3.1** **Wymagania użytkownika i biznesowe**

[- wymagania biznesowe: cele i potrzeby biznesowe, problemy do rozwiązania

wymagania biznesowe - Należy stworzyć program, który zajmowałby się transkrypcją nagrań dźwiękowych i na tej podstawie generował podsumowania. Dane nagrań mają być zapisane w bazie danych dostępnej dla klientów.

* Tworzenie podsumowań w języku polskim
* Baza danych powinna zawierać transkrypcję oraz podsumowanie
* Aplikacja powinna nawiązywać połączenie z mikrofonem

- wymagania użytkowe: potrzeby użytkowników i interesariuszy, cechy użytkowe

* Baza danych powinna umożliwiać dostęp do danych wyłącznie osobom, które uczestniczyły w spotkaniu
* Aplikacja powinna umożliwiać logowanie się
* Klienci powinni móc wykonywać zapytania na bazie danych.
* Klienci powinni móc dołączać do spotkania,

- wymagania systemowe: cechy rozwiązania]

## **3.2** **Wymagania funkcjonalne i niefunkcjonalne**

[specyfikacja;

mapowanie powiązań pomiędzy wymaganiami na poszczególnych poziomach]

Wymagania funkcjonalne:

* system prowadzi konta użytkowników
* system tworzy transkrypcji w czasie rzeczywistym
* system umie rozpoznawać użytkowników po ich głosie czytaj kto teraz mówi
* system może wygenerować podsumowanie spotkania
* system pozwala użytkownikom tworzyć nowe spotkania z zaplanowaną datą i godziną oraz listą uczestników jako całe grupy lub pojedyncze osoby
* system pozwala na przeglądanie informacji o zakończonych spotkaniach do których użytkownik miał dostęp
* system przechowuje informacje o użytkownikach i spotkaniach w bazie danych

Wymagania niefunkcjonalne:

## **3.3** **Przypadki użycia**

biznesowe przypadki użycia:

PB1 scenariusz główny utworzenie spotkania:

1.użytkownik loguje się

2.użytkownik tworzy spotkanie z datą i grupami lub użytkownikami mającymi brać udział w spotkaniu

Scenariusz udany: Spotkanie zostało utworzone i użytkownicy widzą na swoim koncie informacje o spotkaniu.

Scenariusz nieudany: Nie udało się utworzyć spotkania lub użytkownicy nie otrzymali informacji o spotkaniu lub otrzymali niepełne informacje np. brak daty spotkania

PB2 przeglądanie konta: scenariusz główny:

1 użytkownik loguje się

2 użytkownik wybiera grupę w obrębie której chce obejrzeć spotkania lub wchodzi w specjalną grupę inne czyli tam gdzie ma dostęp jako luźny użytkownik

3 wybiera konkretne spotkanie i przegląda jego szczegóły

Scenariusz nieudany: Nawet jak użytkownik brał udział w spotkaniu to nie ma dostępu do szczegółów tego spotkania.

PB3 korzystanie ze spotkania: scenariusz główny:

1 użytkownik loguje się

2 użytkownik rozpoczyna spotkanie

3 przebywa na spotkaniu

4. kończy spotkanie

5. sprawdza szczegóły spotkania

Scenariusz nieudany: Spotkanie się nie rozpoczyna lub nie kończy. Nie tworzą się szczegóły ze spotkania po spotkaniu.

Funkcjonalne przypadki użycia :

FU1: rejestracja użytkownika :scenariusz główny

1 wypełnia dane nazwę i hasło

2.dodania do bazy danych

3.akceptuje dane

rejestracja użytkownika : scenariusz alternatywny - zajęta nazwa użytkownika lub hasło niespełniające wymagania

podpunkty 1-2 jak w głównym

3 wyświetlenie komunikatu o zajętości nazwy lub niespełnianiu wymagań w zależnośći co zostało wykryte przez baze danych i powrót do 1

FU2: logowanie : scenariusz główny

1 wpisz dane nazwę i hasło

2 sprawdzenie poprawności z bazą danych

3 sukces udało się zalogować

logowanie : scenariusz alternatywny - niepoprawne hasło lub nazwa użytkownika:

podpunkty 1 i 2 jak w scenariuszu głównym,

3 .dane niewłaściwe wyświetlenie komunikatu i powrót do punktu 1

logowanie : scenariusz nieudany: Nawet po wpisaniu nieistniejącej nazwy użytkownika i odpowiedniego hasła nie udało się zalogować

FU3: utworzenie grupy użytkowników :scenariusz główny

1 wykonanie scenariusza FU2

2 wypełnia nazwe i dodaje użytkowników do grupy

3. próba dodania do bazy danych

utworzenie grupy użytkowników :scenariusz alternatywny - zajęta nazwa grupy:

podpunkty 1-2-3 jak w głównym

4 wyświetlenie komunikatu o zajętości nazwy lub niespełnianiu wymagań w zależnośći co zostało wykryte przez baze danych i powrót do 2

utworzenie grupy użytkowników : scenariusz nieudany - Nie udało się stworzyć grupy mimo spełnionych wymagań

FU4: dodanie użytkownika do istniejącej grupy :Scenariusz główny

1 wykonanie scenariusza FU2

2 wybranie grupy

3 wybranie użytkownika którego chcesz do niej dodać

FU5: usunięcie użytkownika : Scenariusz główny

1 wykonanie scenariusza FU2

2 wybranie konta do usunięcia i jego likwidacja

FU6: usunięcie grup użytkownika Scenariusz główny

1 wykonanie scenariusza FU2

2 wybranie grupy i usunięcie jej i wszystkich spotkań które się w niej odbyły

FU7:zarządzanie organizacją konta Scenariusz główny

1 wykonanie scenariusza FU2

2 wybranie opcji zarządzania organizacją

FU8: przeglądanie spotkania Scenariusz główny

1 wykonanie scenariusza FU2

2 wybranie grupy

3 wybranie spotkania

4 przejrzenie szczegółów i interesujących danych

FU9: utworzenie spotkania Scenariusz główny

1 wykonanie scenariusza FU2

2 wybranie opcji utwórz spotkanie

3 wybranie grup i luźnych użytkowników do danego spotkania

4 wybór daty spotkania

FU10: edycja parametrów spotkania Scenariusz główny

1 wykonanie scenariusza FU2

2 wybór spotkania

3 zmiana godziny lub dodanie/usunięcie użytkownika lub całego spotkania

FU11: rozpoczęcie i przebieg spotkania Scenariusz główny

1 wykonanie scenariusza FU2

2 wybór spotkania

3 rozpoczęcie generacji transkrypcji

4 zakończenie spotkania

5 pojawienie się podsumowania

FU12 :weryfikacja danych

1 wykonanie scenariusza FU2

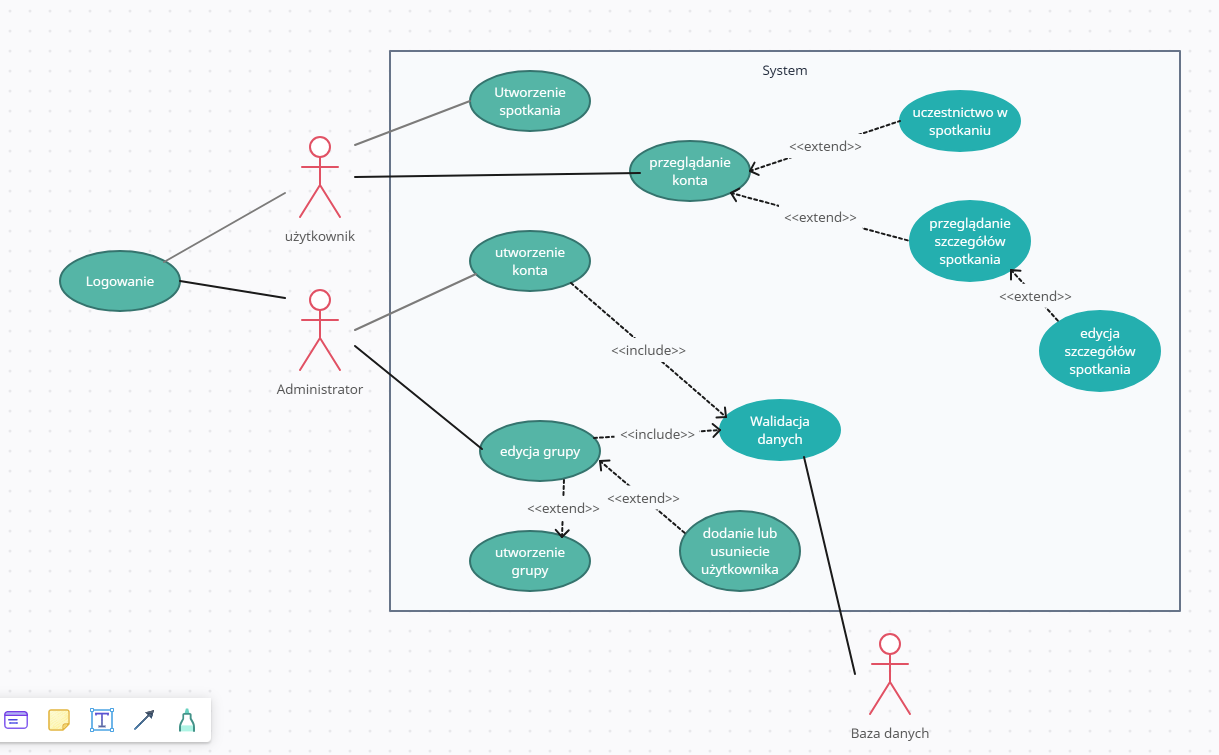
2 baza danych otrzymuje danych

3 otrzymujemy potwierdzenie

weryfikacja danych : scenariusz alternatywny - dane nie poprawne:

2 baza otrzymuje dane

3 otrzymujemy komunikat o niewłaściwych danych



## **3.4** **Potwierdzenie zgodności wymagań**

| Zatwierdzam specyfikację wymagań, jako spełniających potrzeby Klienta. | ………………………………………………………………….  Data i podpis Właściciela tematu |
| --- | --- |
| *Uwagi* | |

# **4** **Definicja architektury**

[plan struktury systemu – model komponentów, modułów, serwisów

zastosowane szablony architektoniczne (np. MVC, mikroserwisy, SOA, szyna usług)

interfejsy kluczowych elementów struktury oraz wyjaśnienie połączeń i interakcji pomiędzy nimi

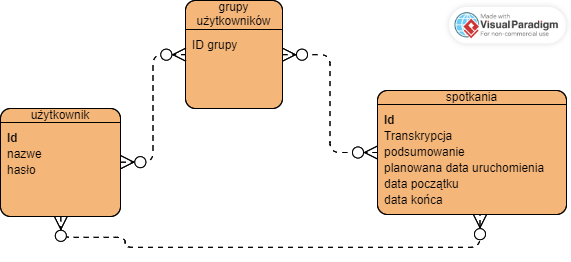
zdecydowaliśmy się na stosowanie następującej architektury:

* Pythona jako łącznik pomiędzy bazą danych a modułem AI oraz do tworzenia GUI ponieważ posiada doskonałą integrację z deepGramem oraz wiele bibliotek do GUI dodatkowo jest kompatybilny z bazą danych Mongo
* AI zdecydowaliśmy się na deepGram ponieważ posiada wszystkie potrzebne funkcje generowanie transkrypcji na żywo generowanie podsumowania oraz rozpoznawanie mówcy oraz jest dostępny do użytkowania w wersji darmowej
* W ramach bazy danych wybraliśmy bazę dokumentową mongo ponieważ dane które będziemy przechowywać to tabela użytkowników, tabele grup oraz tabela spotkań

poniżej jest jej diagram:

# **5** **Dane trwałe**

## **5.1** **Model logiczny danych**



## **5.2** **Przetwarzanie i przechowywanie danych**

W ramach bazy danych wybraliśmy bazę dokumentową mongo ponieważ dane które będziemy przechowywać to tabela użytkowników, tabele grup oraz tabela spotkań

spotkanie może zawierać zarówno luźnych użytkowników jak i całe ich grupy

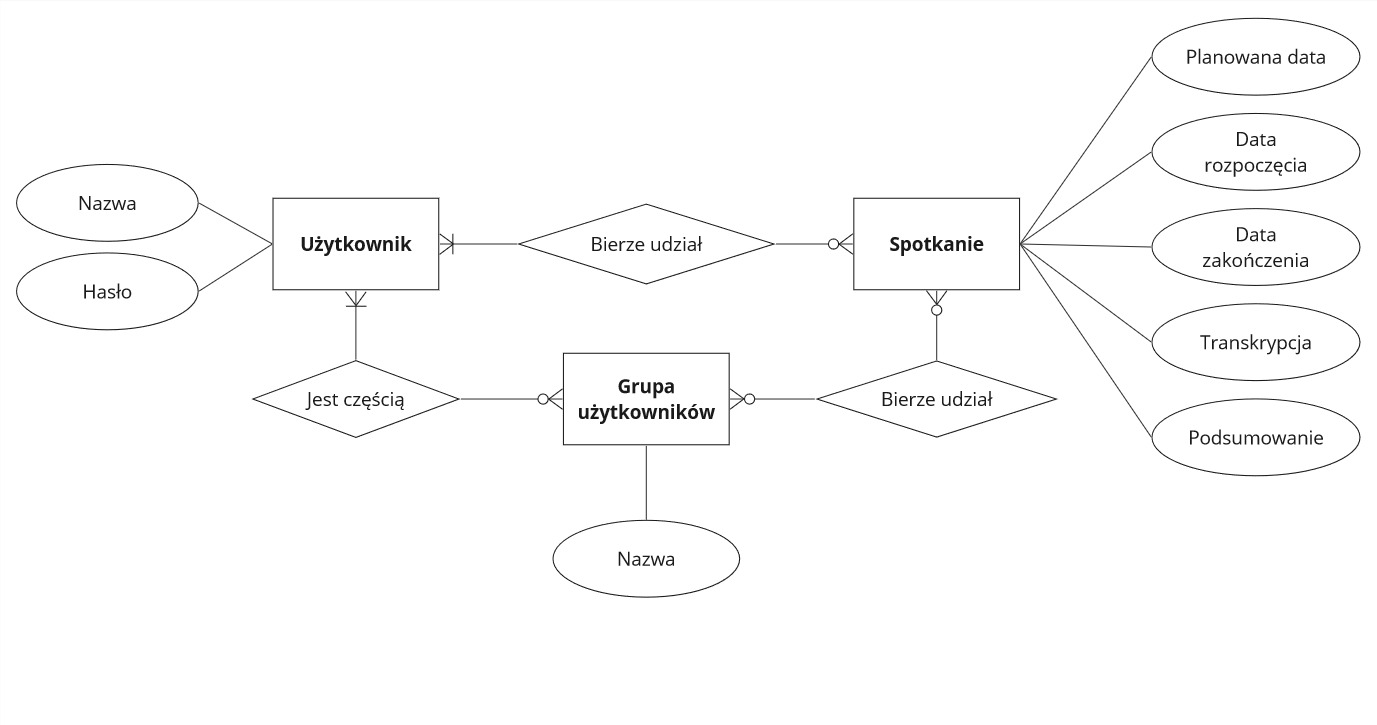
# **6** **Specyfikacja analityczna i projektowa**

Obowiązkowo odnośnik do repozytorium kodu (jedno repozytorium na projekt, jeżeli więcej proszę uzasadnić)

Obowiązkowo określenie metod realizacji: języki programowania, frameworki, środowisko programowania/ uruchamiania/ wdrażania, środowisko ciągłej integracji]

Obowiązkowo Diagram klas lub model pojęciowy struktury informacyjnej: E-R l

Diagram ER:



Opcjonalnie model struktury systemu (diagram wdrożenia)

Opcjonalnie specyfikacja realizacji przypadków użycia: diagramy sekwencji lub współpracy

Obowiązkowo statystyki: licaba plików, linie kodu, liczba testów jednostkowych

# **7** **Projekt standardu interfejsu użytkownika**

[wykorzystanie narzędzi do modelowania oraz tworzenia makiet warstwy prezentacyjnej (np. storyboards, wireframes, wireflows, mockups, prototypes etc)]

# **8** **Specyfikacja testów**

[standardy obsługi błędów i sytuacji wyjątkowych

rodzaje testów, specyfikacja i opis sposobu realizacji poszczególnych rodzajów testów, scenariusze testowe

miary jakości testów]

# ***9*** ***Wirtualizacja/konteneryzacja***

# **10** **Bezpieczeństwo**

# **11** **Podręcznik użytkownika**

[instrukcja użycia funkcjonalności systemu]

# **12** **Podręcznik administratora**

[- instrukcja budowy systemu z kodu źródłowego

- instrukcja instalacji i konfiguracji systemu

- instrukcja aktualizacji oprogramowania

- instrukcja zarządzania użytkownikami i uprawnieniami

- instrukcja tworzenia kopii zapasowych i odtwarzania systemu

- instrukcja zarządzania zasobami systemu]

# **13** **Podsumowanie**

[Krytyczna analiza osiągniętych wyników, mocne i słabe strony

Możliwe kierunki rozwoju]

# **14** **Bibliografia**

[Wykaz materiałów źródłowych, opis zgodny ze standardem sporządzania opisów bibliograficznych -<https://bg.pw.edu.pl/index.php/przypisy-i-bibliografia>]

| Zatwierdzam dokumentację. | …………………………………………………………………………………………………….  Data i podpis Mentora |
| --- | --- |