

Projekt Zespołowy 2 Grupa 4M

Spis treści

<u>Spis treści</u>	<u>1</u>
<u>1 Wprowadzenie</u>	<u>2</u>
<u>1.1 Cel projektu</u>	<u>2</u>
<u>1.2 Wstępna wizja projektu</u>	<u>2</u>
<u>2 Metodologia wytwarzania</u>	<u>2</u>
<u>3 Analiza wymagań</u>	<u>2</u>
<u>3.1 Wymagania użytkownika i biznesowe</u>	<u>2</u>
<u>3.2 Historyjki użytkowników:</u>	<u>3</u>
<u>3.3 Wymagania funkcjonalne i нефункционалне</u>	<u>4</u>
<u>3.4 Przypadki użycia</u>	<u>4</u>
<u>3.4 Potwierdzenie zgodności wymagań</u>	<u>20</u>
<u>4 Architektura</u>	<u>20</u>
<u>4.1 Definicja architektury</u>	<u>20</u>
<u>4.2 Specyfikacja analityczna i projektowa</u>	<u>23</u>
<u>5 Dane trwałe</u>	<u>26</u>
<u>5.1 Model ER</u>	<u>26</u>
<u>5.2 Model logiczny danych</u>	<u>27</u>
<u>5.3 Przetwarzanie i przechowywanie danych</u>	<u>27</u>
<u>6 Projekt standardu interfejsu użytkownika</u>	<u>28</u>

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 przeprowadzane 2 razy w tygodniu zdalnie na platformie Discord. Komunikacja z właścicielem projektu oraz mentorem jest realizowana za pomocą platformy Teams i jest stała. W trakcie tworzenia projektu występują spotkania z ekspertami, którzy pomagają nam w pracy w zakresie ich kompetencji. W połowie czasu przeznaczanego na realizację projektu organizowane jest spotkanie z komisją, która ocenia wykonaną prezentację śródsesjonalną, a w końcowej fazie projektu prezentację finalną.

3 Analiza wymagań

3.1 Wymagania użytkownika i biznesowe

Wymagania biznesowe:

- Program ma zajmować się transkrypcją nagrań dźwiękowych i na tej podstawie generować podsumowania. Dane nagrań mają być zapisane w bazie danych dostępnej dla klientów.
- Ma być możliwe tworzenie podsumowań w języku polskim
- Baza danych powinna zawierać transkrypcję oraz podsumowanie
- Aplikacja powinna nawiązywać połączenie z mikrofonem

Wymagania 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

3.2 Historyjki użytkowników:

- Uczestnik:
 - Jako uczestnik spotkania chcę móc się zalogować do aplikacji w łatwy sposób, podając nazwę użytkownika i hasło
 - Jako uczestnik chce móc przeglądać zaplanowane spotkania w przejrzysty sposób w żeby wiedzieć kiedy ono będzie i kto inny na nim będzie
 - Spotkania mają być pokazywane w postaci kalendarza dni, gdzie same spotkania mają być reprezentowane w postaci kafelków z datą i tematem spotkania
 - Jako uczestnik chce móc przeglądać ukończone spotkania aby wiedzieć kiedy był obecny na nim oraz co było mówione i poznać jego krótkie podsumowanie
 - Chcę, aby przegląd ukończonego spotkania był dostępny maksymalnie 10 minut po zakończeniu spotkania
- Moderator
 - Jako Moderator chce móc tworzyć spotkania i edytować ich szczegóły aby zarządzać nimi zgodnie z moimi wymaganiami
 - Nie chcę, aby do spotkania dołączył się użytkownik, który nie jest do niego zaproszony
 - Jako Moderator muszę móc rozpoczynać i kończyć spotkania
 - Nie chcę aby ktoś z użytkowników mógł rozpocząć spotkanie przedwcześnie ani go przedwcześnie skończyć
- Administrator
 - Jako Administrator chce móc tworzyć i edytować grupy aby zapewnić organizację uczestników
 - Chcę, aby łatwiej było moderatorowi dodawać grupy do spotkań, zamiast dodawać osobę po osobie wystarczy, że doda grupę do spotkania
 - Jako administrator chce móc tworzyć konta i nadawać im odpowiednie rolę aby zapewnić odpowiedni kształt organizacji
 - Chcę, aby odpowiednie osoby mogły mieć stworzone konta o wyższych uprawnieniach, nie mogą o tym sami decydować

3.3 Wymagania funkcjonalne i нефункционалне

Wymagania funkcjonalne:

Czyli za co odpowiada system:

- Prowadzi konta użytkowników
- Tworzy transkrypcje po spotkaniu
- Umie rozpoznawać użytkowników po ich głosie czytaj kto teraz mówi
- Potrafi wygenerować podsumowanie spotkania
- Pozwala użytkownikom tworzyć nowe spotkania z zaplanowaną datą i godziną oraz listą uczestników jako całe grupy lub pojedyncze osoby
- Pozwala na przeglądanie informacji o zakończonych spotkaniach do których użytkownik miał dostęp
- Przechowuje informacje o użytkownikach i spotkaniach w bazie danych

Wymagania нефункционалне:

- W spotkaniu może uczestniczyć maksymalnie 30 osób.
- Jednocześnie może się odbywać nieskończenie wiele spotkań, gdyż każde spotkanie odbywa się na urządzeniu klienta, a dane ze spotkania są umieszczane w bazie danych, która umożliwia jednoczesną aktualizację wyników.
- Dane są przechowywane bezpiecznie
- Aplikacja jest darmowa, ale po wyczerpaniu \$200 darmowych kredytów na platformie Deepgram należy utworzyć nowe konto, aby mieć ponowne \$200 darmowych kredytów.

3.4 Przypadki użycia

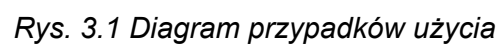
Nasza aplikacja powstała z myślą, aby była stosowana dla uczestników takich ośrodków jak szkoła letnia, dlatego w dalszej części dokumentacji mogą pojawić się do niej odniesienia.

W naszej aplikacji użytkownicy mogą mieć różne role. Poniżej znajduje się podstawowy ich opis, natomiast wszystkie role użytkowników oraz ich możliwe działania w aplikacji są przedstawione w diagramie przypadków użycia na Rys. 3.1. W dalszej części rozdziału znajdują się historyjki użytkowników z podziałem na ich role, a następnie biznesowe i funkcjonalne przypadki użycia. Baza danych naszej aplikacji jest zdalna, więc oczywistym wymaganiem każdego przypadku użycia jest dostęp do internetu, dlatego ten warunek nie będzie umieszczony w dalszej części dokumentacji.

Podstawowy opis przypadków użycia:

- Uczestnik (uczestnik szkoły letniej) może uczestniczyć w spotkaniu i przeglądać transkrypcję i podsumowanie ze spotkania.
- Moderator (który by był traktowany jako nauczyciel szkoły letniej) tworzy, rozpoczyna i kończy spotkania oraz edytuje szczegóły spotkania takie jak przypisywanie

- Administrator (kierownik szkoły letniej) tworzy i usuwa konta użytkowników i moderatorów oraz tworzy i edytuje grupy (np. klasa 2d w szkole letniej).



Rys. 3.1 Diagram przypadków użycia

Przypadki użycia podzielone względem aktorów:

- Uczestnik
 - Biznesowe przypadki użycia:
 - Przeglądanie konta (biz. przyp. nr 1)
 - Funkcjonalne przypadki użycia
 - Logowanie (fun. przyp. nr. 1)
 - Sprawdzanie terminów zapisanych spotkań
 - Przeglądanie spotkania (fun. przyp. nr. 2)
- Moderator
 - Biznesowe przypadki użycia:
 - Przeglądanie konta (biz. przyp. nr 1)
 - Zarządzanie spotkaniem (biz. przyp nr 2)
 - Funkcjonalne przypadki użycia
 - Wszystkie funkcjonalne przypadki użycia uczestnika
 - Utworzenie spotkania (fun. przyp. nr 3)
 - Usunięcie spotkania
 - Rozpoczęcie spotkania
 - Zakończenie spotkania
 - Przypisanie uczestników rozmowy do transkrypcji

- Edycja terminu spotkania
 - Edycja użytkowników lub grup spotkania
- Administrator
 - Biznesowe przypadki użycia:
 - Przeglądanie konta (przyp nr.1)
 - Zarządzanie kontem użytkownika (przyp nr.3)
 - Zarządzanie grupą
 - Funkcjonalne przypadki użycia:
 - Wszystkie funkcjonalne przypadki użycia moderatora
 - Tworzenie konta użytkownikowi (rejestracja użytkownika) (fun. przyp. nr. 4)
 - Usunięcie konta użytkownikowi
 - Tworzenie grupy (fun. przyp. nr 5)
 - Usunięcie grupy
 - Dodawanie użytkownika do grupy (fun. przyp. nr 6)
 - Usunięcie użytkownika z grupy
 - Edycja danych grupy
- Baza danych
 - Biznesowe przypadki użycia:
 - Zarządzanie bazą danych (przyp nr.4)
 - Funkcjonalne przypadki użycia:
 - Odczyt z bazy danych
 - Aktualizacja bazy danych
 - Weryfikacja danych (przyp. nr. 7)

Biznesowe przypadki użycia:

Biznesowy przypadek użycia nr.1	Przeglądanie konta	
Opis	Użytkownik powinien móc w łatwy sposób sprawdzać jakie spotkania już miał lub będzie miał oraz sprawdzać ich szczegóły.	
Wymagania	Użytkownik jest zalogowany.	
Warunek scenariusza udanego	Użytkownik pozyskał poszukiwane informacje.	
Warunek scenariusza nieudanego	Użytkownik nie może sprawdzić spotkania, w którym jest wpisany jako jego członek lub członek grupy lub nie ma dostępu do szczegółów tego spotkania.	
Aktorzy	Uczestnik, moderator i admin	
Wyzwalacz	Próba zobaczenia spotkań, w których użytkownik brał lub będzie brał udział lub chęć sprawdzenia informacji o nim takich jak transkrypcja lub podsumowanie.	
Scenariusz główny	Krok	Akcja
	1	Sprawdzenia spotkań w których użytkownik będzie brał udział
	2	Odczyt z bazy danych
	3	Dostarczenie informacji o utworzonych spotkaniach
Rozszerzenie scenariusza głównego	4	Próba sprawdzenia szczegółów wybranego spotkania
	2	Akcja kroku 2
	6	Dostarczenie informacji o wybranym spotkaniu

Biznesowy przypadek użycia nr.2	Zarządzanie spotkaniem	
Opis	Wszystkie działania związane z zarządzaniem spotkaniem	
Wymagania	Użytkownik jest zalogowany jako moderator lub administrator	
Warunek scenariusza udanego	Użytkownik może utworzyć lub usunąć spotkanie, rozpocząć i je zakończyć oraz edytować termin, użytkowników i grupy spotkania	
Warunek scenariusza nieudanego	Nie udało się osiągnąć przynajmniej jednego warunku scenariusza udanego.	
Aktorzy	Moderator lub admin	
Wyzwalacz	Brak	
Scenariusz główny 1	Krok	Akcja
	1	Próba utworzenia spotkania
	2	Poprawna walidacja danych
	3	Aktualizacja bazy danych
	4	Utworzenie spotkania
Scenariusz alternatywny do scenariusza głównego 1	1	Akcja kroku 1
	2a	Nie poprawna walidacja danych
	1, 2, 3, 4	Akcje kroków 1, 2, 3, 4
Scenariusz główny 2	1b	Próba usunięcia spotkania
	2, 3	Akcje kroków 2, 3
	4b	Usunięcie spotkania
Scenariusz alternatywny do scenariusza głównego 2	1b, 2a, 1b, 2, 3, 4b	Akcja kroku 1b, 2a, 1b, 2, 3, 4b
Scenariusz główny 3	1c	Próba rozpoczęcie spotkania
	2c	Rozpoczęcie spotkania

Rozszerzenie scenariusza głównego 3	3c	Próba zakończenia spotkania
	4c	Przypisanie uczestników rozmowy do transkrypcji
	3	Akcja kroku 3
Scenariusz główny 4	1d	Próba edycji terminu spotkania
	2, 3	Akcja kroków 2, 3
	4d	Edycja terminu spotkania
Scenariusz alternatywny do scenariusza głównego 4	1d, 2a, 1d, 2, 3, 4d	Akcje kroków 1d, 2a, 1d, 2, 3, 4d
Scenariusz główny 5	1e	Próba edycji użytkownika lub grupy biorącej udział w spotkaniu
	2, 3	Akcja kroków 2, 3
	4e	Edycja użytkownika lub grupy biorącej udział w spotkaniu
Scenariusz alternatywny do scenariusza głównego 5	1e, 2a, 1e, 2, 3, 4e	Akcja kroków 1e, 2a, 1e, 2, 3, 4e

Biznesowy przypadek użycia nr.3	Zarządzanie kontem użytkownika	
Opis	Wszystkie działania związane z zarządzaniem konta użytkownika	
Wymagania	Użytkownik jest zalogowany jako administrator	
Warunek scenariusza udanego	Użytkownik może stworzyć i usunąć konto użytkownika przy podaniu poprawnych danych.	
Warunek scenariusza nieudanego	Użytkownik nie potrafi stworzyć konta użytkownika mimo to, że podał poprawne dane lub użytkownik nie może usunąć konta użytkownika.	
Aktorzy	Administrator	
Wyzwalacz	Brak	
	Krok	Akcja
Scenariusz główny 1	1	Próba utworzenia konta dla nowego użytkownika
	2	Poprawna walidacja danych
	3	Aktualizacja bazy danych
	4	Utworzenie konta dla nowego użytkownika
Scenariusz alternatywny do scenariusza głównego 1	1	Akcja kroku 1
	2a	Nie poprawna walidacja danych, podanie nie poprawnego loginu lub hasła (powtarzającego się u innych użytkowników)
	1, 2, 3, 4	Akcje kroków 1, 2, 3, 4
Scenariusz główny 2	1b	Próba usunięcia konta użytkownika
	3	Akcja kroku 3
	4b	Usunięcie konta użytkownika

Biznesowy przypadek użycia nr.4	Zarządzanie bazą danych	
Opis	Wszystkie działania związane z zarządzaniem bazą danych przez bazę danych	
Wymagania	brak	
Warunek scenariusza udanego	Baza danych dokonała walidacji danych	
Warunek scenariusza nieudanego	Użytkownik nie potrafi stworzyć konta użytkownika mimo to, że podał poprawne dane lub użytkownik nie może usunąć konta użytkownika.	
Aktorzy	Administrator	
Wyzwalacz	Brak	
	Krok	Akcja
Scenariusz główny 1	1	Próba dokonania walidacji danych
	2	Wykonanie walidacji danych
Scenariusz główny 2	1a	Próba odczytania danych
	2	Odczytanie danych
Scenariusz główny 3	1b	Próba aktualizacji bazy danych
	1, 2	Akcje kroków 1, 2
	3b	Aktualizacja bazy danych
Scenariusz alternatywny do scenariusza głównego 3	1b, 1	Akcje kroków 1b, 1
	2c	Nie poprawna walidacja danych
	1b, 1, 2, 3b	Akcje kroków 1b, 1, 2, 3b

Funkcjonalne przypadki użycia :

Funkcjonalny przypadek użycia nr.1	Logowanie użytkownika	
Szczegółowy Opis	Zalogowanie się użytkownika za pomocą loginu i hasła.	
Wymagania	Posiadanie loginu i hasła w systemie	
Warunek scenariusza udanego	Użytkownik prawidłowo zalogował się i uzyskał dostęp do aplikacji	
Warunek scenariusza nieudanego	Scenariusz nieudany: Nawet po wpisaniu istniejącej nazwy użytkownika i odpowiedniego hasła nie udało się zalogować	
Aktorzy	Uczestnik	
Wyzwalacz	Uruchomienie aplikacji	
Scenariusz główny	Krok	Akcja
	1	Wyświetlenie okna umożliwiającego wpisanie loginu i hasła Wpisanie loginu i hasła (podać numer ekranu)
	2	Sprawdzenie poprawności danych
	3	Sukces - udało się zalogować (przejdzie do ekranu głównego)
Scenariusz alternatywny	2a	Niepoprawny login lub hasło. Konieczność ponownego wpisania danych (wypisuje się błąd)

Funkcjonalny przypadek użycia nr. 2	Przeglądanie szczegółów spotkania	
Szczegółowy Opis	Uczestnik i moderator powinni móc w łatwy sposób sprawdzać informacje zarówno o nadchodzących spotkaniach jak i odbytych	
Wymagania	Użytkownik jest zalogowany	
Warunek scenariusza udanego	Aktor pozyskał poszukiwane informacje	
Warunek scenariusza nieudanego	Nawet jak użytkownik brał udział w spotkaniu to nie ma dostępu do szczegółów tego spotkania.	
Aktorzy	Uczestnik	
Wyzwalacz	Chęć pozyskania informacji o spotkaniu	
	Krok	Akcja
Scenariusz główny	1	wybór grupy w obrębie której chce obejrzeć spotkania lub wchodzi w specjalną grupę inne czyli tam gdzie ma dostęp jako luźny uczestnik
	2	Wybór konkretnego spotkania i przeglądanie jego szczegółów
Scenariusz alternatywny	brak	brak

Funkcjonalny przypadek użycia nr.3	Utworzenie spotkania	
Dodatkowy opis	brak	
Wymagania	Użytkownik jest zalogowany jako moderator lub administrator.	
Warunek scenariusza udanego	Spotkanie zostało utworzone i jego przyszli uczestnicy widzą je na swoim koncie	
Warunek scenariusza nieudanego	Nie udało się stworzyć spotkania lub użytkownicy nie otrzymali informacji o spotkaniu lub otrzymali niepełne informacje np. brak daty spotkania	
Aktorzy	Moderator lub admin	
Wyzwalacz	brak	
Scenariusz główny	Krok	Akcja
	1	Kliknięcie w pole tworzenia spotkania na stronie głównej
	2	Podanie nazwy spotkania
	3	Podanie daty dziennej i godziny rozpoczęcia spotkania
	4	Podanie daty dziennej i godziny zakończenia spotkania
	5	Wybranie użytkowników i/lub grup biorących udział w spotkaniu

Funkcjonalny przypadek użycia nr.4	Rejestracja użytkownika	
Dodatkowy opis	Brak	
Wymagania	Rejestrować użytkowników do systemu może tylko administrator. To on decyduje kto się może zalogować z jakimi uprawnieniami.	
Warunek scenariusza udanego	Rejestracja się udała i użytkownik może się zalogować.	
Warunek scenariusza nieudanego	Użytkownik pomimo wpisania poprawnych danych nie może się zalogować.	
Aktorzy	Administrator	
Wyzwalacz	Administrator chce stworzyć konto użytkownikowi i nadać mu priorytet w aplikacji.	
Scenariusz główny	Krok	Akcja
	1	Kliknięcie w pole tworzenia konta na stronie głównej administratora.
	2	Podanie przyszłego loginu i hasła użytkownika
	3	Zatwierdzenie stworzenia konta
Scenariusz alternatywny	1	Akcja kroku 1
	2a	Podanie nie poprawnego loginu lub hasła (powtarzającego się u innych użytkowników)
	1, 2, 3	Akcje kroków 1, 2, 3

Funkcjonalny przypadek użycia nr.5	Tworzenie grupy użytkowników	
Dodatkowy opis	Brak	
Wymagania	Zalogowany jako administrator	
Warunek scenariusza udanego	Poprawnie utworzono nową grupę użytkowników	
Warunek scenariusza nieudanego	Nie udało się utworzyć grupy użytkowników	
Aktorzy	Administrator	
Wyzwalacz	Brak	
Scenariusz główny	Krok	Akcja
	1	Podanie nazwy dla nowej grupy
	2	Sprawdzenie poprawności danych
	3	Zapisanie nowej grupy z podaną nazwą
Rozszerzenie scenariusza głównego. Dodanie użytkowników do grupy	4	Pobranie użytkowników z bazy danych
	5	Wybranie użytkowników do dodania
	6	Dodanie użytkowników do grupy
Scenariusz alternatywny	3a	Zajęta nazwa grupy - powrót do kroku 1

Funkcjonalny przypadek użycia nr. 6	Dodanie użytkowników do grupy	
Szczegółowy Opis	Brak	
Wymagania	Zalogowany jako administrator, istnieje grupa, do której można dodać użytkowników	
Warunek scenariusza udanego	Dodano wybranych użytkowników do grupy	
Warunek scenariusza nieudanego	Nie udało się dodać użytkowników do grupy	
Aktorzy	Administrator	
Wyzwalacz	Brak	
	Krok	Akcja
Scenariusz główny	1	Wybranie grupy, do której chcemy dodać użytkowników
	2	Kliknięcie przycisku dodaj użytkowników
	3	Pobranie z bazy danych listy użytkowników możliwych do dodania
	4	Wybranie użytkowników z listy
	5	Wybranie użytkowników do dodania
	6	Dodanie użytkowników do grupy

Scenariusz alternatywny	4a	Brak użytkowników możliwych do dodania - w skład grupy wchodzi już wszyscy użytkownicy
-------------------------	----	--

Funkcjonalny przypadek użycia nr.7	Weryfikacja danych	
Szczegółowy Opis	Nasza baza danych weryfikuje poprawność wprowadzonych danych czy nie tworzą konfliktów i są zgodne z innymi ograniczeniami.	
Wymagania	Funkcja systemowa aktywowane automatycznie	
Warunek scenariusza udanego	Wszystkie dane są zgodne z ograniczeniami	
Warunek scenariusza nieudanego	W bazie danych pojawiły się niespójność.	
Aktorzy	Baza danych	
Wyzwalacz	Próba edycji lub dodania danych do bazy danych	
Scenariusz główny	Krok	Akcja
	1	Baza danych otrzymuje dane
	2	Sprawdzenie poprawności danych
	3	Potwierdzenie poprawności danych
Scenariusz alternatywny	2a	dane niepoprawne
	3a	Komunikat o błędzie

Do zastanowienia się:

FU5 Usunięcie użytkownika : Scenariusz główny

Warunki do przystąpienia: zalogowałeś się jako Administrator

1. Wybranie konta do usunięcia i jego likwidacja

Scenariusz udany: użytkownik został usunięty z bazy danych

Scenariusz nieudany: użytkownik wciąż istnieje

FU6 Usunięcie grupy użytkowników - Scenariusz główny

Warunki do przystąpienia: zalogowałeś się jako Administrator

1. Wybranie grupy i usunięcie jej i wszystkich spotkań które się w niej odbyły

Scenariusz udany: grupa została usunięta z bazy danych wraz z wszystkimi spotkaniami które były dostępne tylko w niej

Scenariusz nieudany: grupa została lub w bazie danych zostały zbędne spotkania

3.4 Potwierdzenie zgodności wymagań

Zatwierdzam specyfikację wymagań, jako spełniających potrzeby Klienta. Data i podpis Właściciela tematu
<i>Uwagi</i>	

4 Architektura

Architekturę podzieliliśmy na 2 podrozdziały. Pierwszy podrozdział określa definicję architektury. Tam przedstawione są szablony, których użyliśmy oraz składniki diagramu C4

przedstawiające poszczególne warstwy aplikacji. Drugim podrozdziałem jest specyfikacja analityczna i projektowa. W tym rozdziale przedstawiamy nasze decyzje architektoniczne z perspektywy programistycznej oraz ich uzasadnienie.

4.1 Definicja architektury

Zastosowaliśmy szablony:

- MVC

Interakcja klienta z widokiem powoduje modyfikację stanu modelu przez kontroler. Kontroler pobiera dane z modelu aby widok mógł je wyświetlić.

U nas model reprezentuje dane sesji burzy mózgów, użytkowników i grup.

Widok to GUI, a kontroler to systemy zarządzania danymi (zarządzanie sesjami burzy mózgów, uwierzytelnianiem, kontakt z bazą danych).

- Klient serwer

Klientem jest u nas aplikacja desktopowa, a rolę serwera pełni MongoDB w chmurze i zewnętrzne usługi takie jak Deepgram oraz Google Translate.

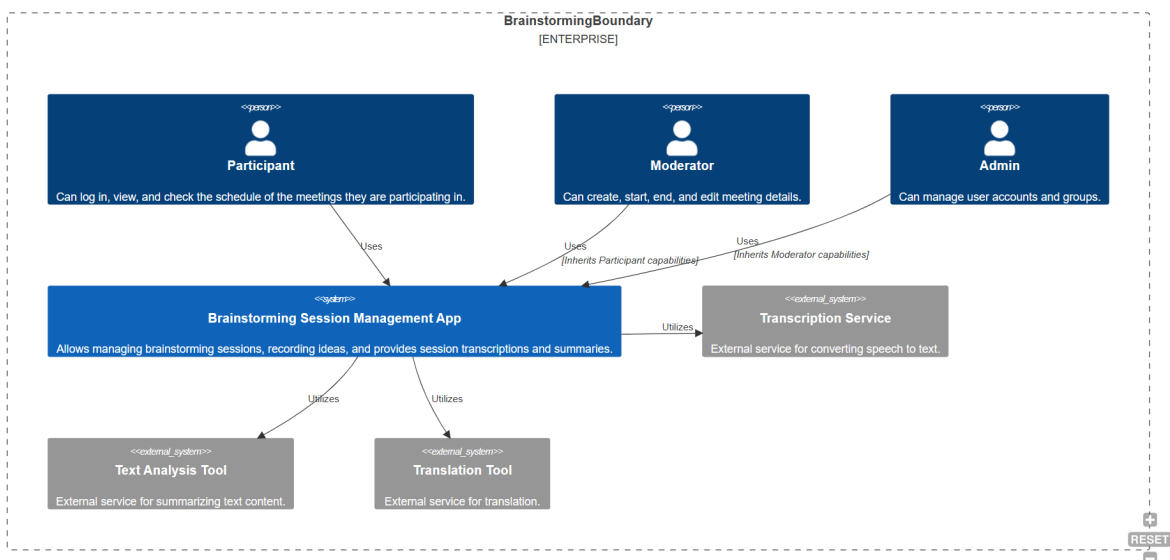
Poniżej znajdują się diagramy C4 poszczególnych warstw aplikacji. Pierwszy diagram dotyczy kontekstu systemu i jest przedstawiony na Rys. 4.1. Na tym poziomie diagramy dają odbiorcom widok z lotu ptaka na cały system. Szczegóły dotyczące wewnętrznego działania systemu i technologii nie są ważne na tym poziomie. Diagramy te powinny zawierać ogólny opis systemu — co robi, kto go używa i z jakimi innymi systemami współdziała.

Następnie mamy diagram kontenerowy przedstawiony na Rys. 4.2. Ten diagram powiększa następny poziom systemu, aby szczegółowo opisać kontenery, których system będzie używał, oraz sposób, w jaki będą się komunikować i współdziałać ze sobą. Kontener MongoDB Database odnosi się tylko do naszej bazy danych, w której będziemy uzupełniać dane, a nie do usługi hostującej bazę danych. Umieszczenie danych w tej bazie jest częścią aplikacji.

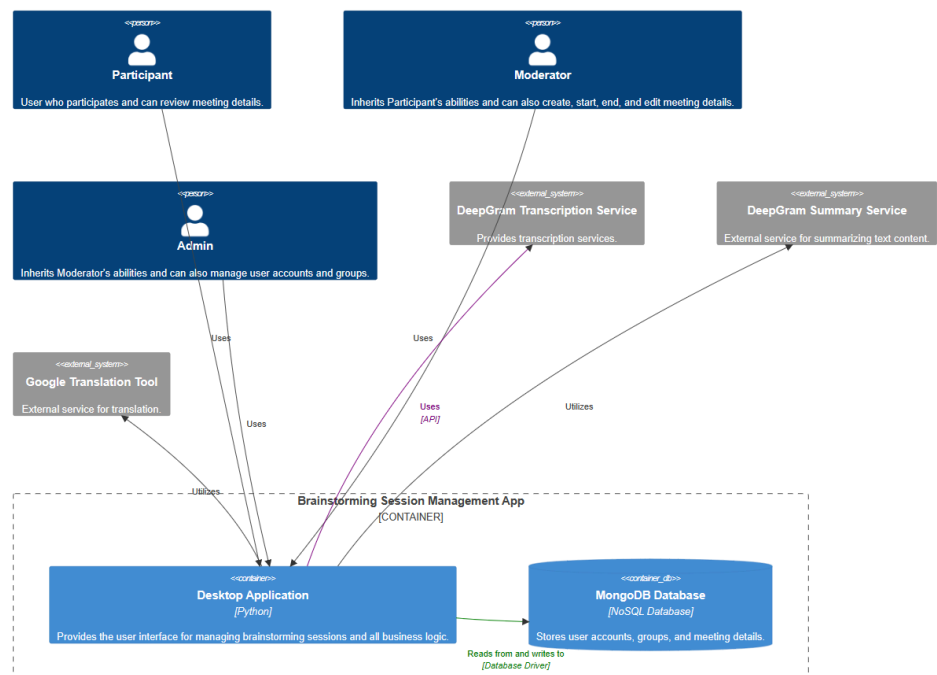
Kolejnym diagramem jest diagram komponentowy przedstawiony na Rys. 4.3. Schematy komponentów ilustrują szczegóły poszczególnych kontenerów. Diagramy te dzielą strukturalne bloki konstrukcyjne (kod, systemy plików, biblioteki) kontenera na poszczególne komponenty. Diagram szczegółowo opisuje, czym jest każdy komponent, jakie są jego obowiązki, jakie ma interakcje z innymi komponentami oraz szczegóły technologii/implementacji. Komponent Transcription Manager oprócz uzyskania transkrypcji zarządza także operacjami związanymi z transkrypcją, takimi jak podsumowanie lub translacja.

Na końcu mamy diagram wdrożeniowy zaprezentowany na Rys. 4.4. Diagram wdrożeniowy ilustruje, w jaki sposób architektura oprogramowania, zaprojektowana na poziomie

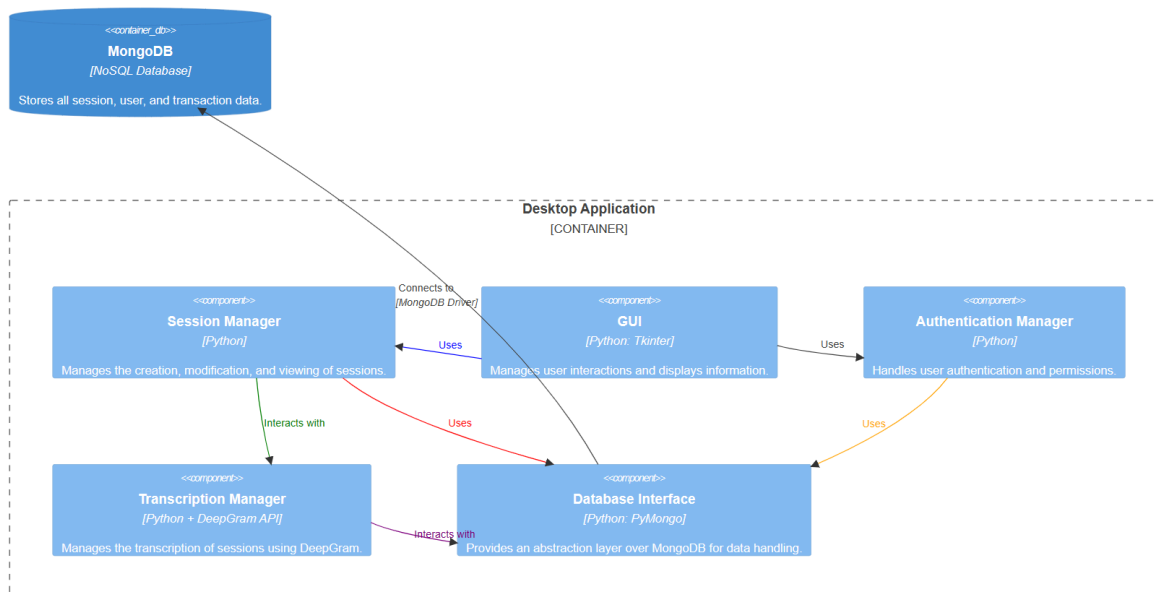
konceptyjnym, przekłada się na fizyczną architekturę systemu, w której oprogramowanie będzie działać jako węzły. Mapuje wdrażanie komponentów oprogramowania w węzłach sprzętowych i przedstawia ich relacje za pośrednictwem ścieżek komunikacyjnych, umożliwiając wizualną reprezentację środowiska wykonawczego oprogramowania w wielu węzłach.



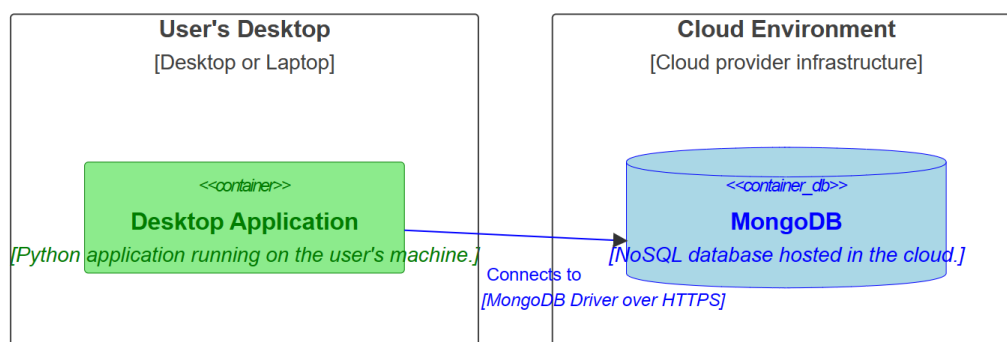
Rys. 4.1 Diagram kontekstowy



Rys. 4.2 Diagram kontenerowy



Rys. 4.3 Diagram komponentowy



Rys. 4.4 Diagram wdrożeniowy

4.2 Specyfikacja analityczna i projektowa

Link do repozytorium kodu: https://gitlab-stud.elka.pw.edu.pl/mpakulsk/pzsp2_24/

Podstawowe informacje:

- Aplikacja desktopowa
- Backend – Python

- Frontend – Python (biblioteka Tkinter)
- Transkrypcja – Deepgram
- Podsumowania – Deepgram
- Translacja – Tłumacz Google
- Baza danych – MongoDB

Decyzje wyboru:

Aplikacja desktopowa z backendem i frontendem w Pythonie:

- Wybraliśmy aplikację desktopową, ponieważ została nam ona polecona przez naszego mentora.
- Zastanawialiśmy się jeszcze nad aplikacją webową, w której backend napisany byłby w Pythonie z frameworkiem Flask, a frontend w JavaScriptcie z frameworkiem ReactJS.
- Myśleliśmy również o stworzeniu aplikacji desktopowej z backendem w Pythonie z frameworkiem Flask, a frontendem również w JavaScriptcie z użyciem frameworku Elektron, ale z tego zrezygnowaliśmy, ponieważ mentor nam zalecił napisanie frontendu w języku Python.
- Myśleliśmy jeszcze o aplikacji mobilnej z backendem w Pythonie za pomocą frameworku kivy i frontendem w JavaScriptcie za pomocą React Native lub backendem i frontendem w JavaScriptcie za pomocą React Native.
- Wybraliśmy Pythona zarówno również dlatego, że język ten posiada dobrą integrację z Deepgramem, jest kompatybilny z bazą danych MongoDB i jest stosunkowo prostym językiem.
- Przy tworzeniu GUI wykorzystamy bibliotekę Tkinter, ponieważ jest wbudowana w Pythona i również jest stosunkowo prosta w użyciu.

Transkrypcja:

- Wybraliśmy Deepgram jako narzędzie do transkrypcji, ponieważ posiada model Nova-2 z opcją meeting, która potrafi generować transkrypcję w wielu językach w stosunkowo dobrej jakości.
- Również ma wbudowaną możliwość diaryzacji (podziału wypowiedzianego tekstu na mówców).
- Deepgram nie jest bezpłatnym narzędziem, ale posiada darmowe \$200 kredytu, które starcza na długi czas. W przypadku wykorzystania kredytów, można założyć drugie konto i ponownie operować na \$200 darmowych kredytów.

- Testowaliśmy też inne opcje modelu Nova-2 jak general oraz inne modele jak Nova, Enhanced, Base.

Podsumowania:

Do podsumowania również wybraliśmy model Nova-2 z narzędzia Deepgram, ponieważ w porównaniu z innymi modelami prezentował dobrą jakość tworzenia podsumowań. Podsumowania, które są tworzone są abstrakcyjne, dzięki czemu opisują przebieg rozmowy własnymi słowami zamiast cytowania pewnych zdań z transkrypcji rozmowy.

Za pomocą biblioteki Transformers z HuggingFace testowaliśmy również inne modele do tworzenia podsumowań takie jak:

- allegro/herbert-base-cased,
- z-dickson/bart-large-cnn-climate-change-summarization – jedyny model z biblioteki Transformers, który potrafi podsumowywać bezpośrednio na język polski,
- facebook/bart-large-cnn.
- google-bert/bert-base-multilingual-cased

Stworzone podsumowanie nie jest idealne, ponieważ skupiałem się na tym, aby używać modeli, które są darmowe żeby aplikacja również była darmowa.

Translacja:

Do translacji użyliśmy narzędzia Tłumacz Google, ponieważ jest całkowicie bezpłatny oraz zapewnia może nie najdokładniejsze ale najbardziej kompletne tłumaczenie tekstu.

Za pomocą biblioteki Transformers z HuggingFace testowaliśmy również inne modele do tworzenia transkrypcji takie jak:

- gsarti/opus-mt-tc-en-pl
- facebook/m2m100_418M
- Helsinki-NLP/opus-mt-pl-en,

Niestety w większości przypadków tekst albo nie tłumaczył się do końca albo ten sam tekst był powtarzany wielokrotnie.

Baza danych:

Wybraliśmy bazę danych MongoDB ponieważ umożliwia korzystanie z bazy dokumentowej, która w naszym przypadku wydawała się najlepszą opcją. Do bazy danych miały być zapisywane między innymi dane z transkrypcji oraz podsumowania w języku takim w jakim

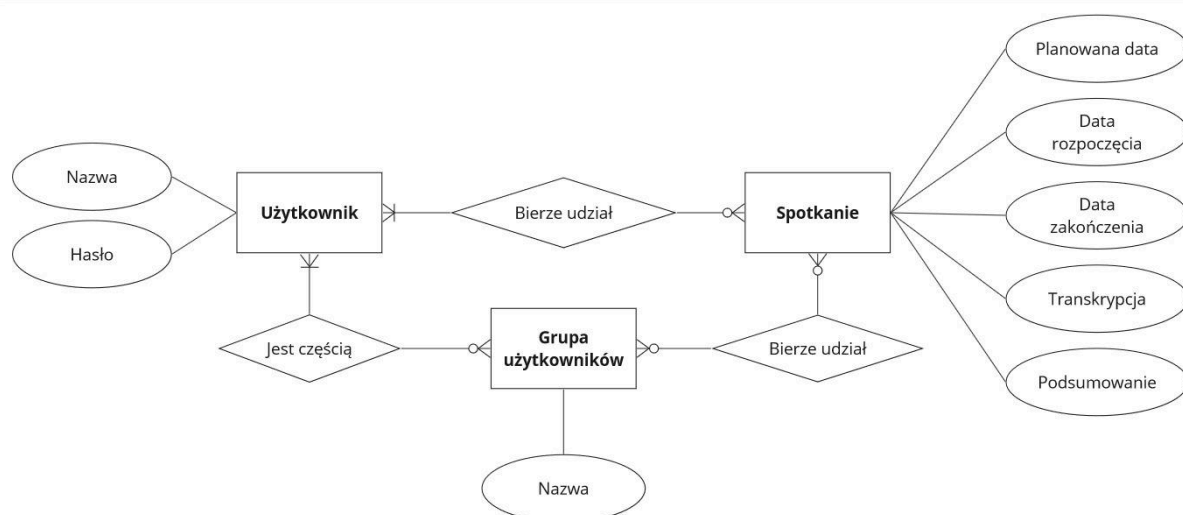
przebiegała rozmowa. Nie wybraliśmy bazy relacyjnej, ponieważ baza danych MongoDB jest prostsza w obsłudze i może pomieścić obszerne dane z transkrypcji rozmowy oraz nie ma potrzeby zadawania do bazy danych złożonych zapytań. Dodatkowo liczba uczestników spotkania w tabeli spotkań może być różna, a w bazie SQL przez to że tabele mają podział na kolumny trzeba by było trzymać identyfikator każdego uczestnika spotkania w oddzielnej kolumnie lub stworzyć dodatkową tabelę realizującą relację wiele do wielu.

5 Dane trwałe

Ten rozdział skupia się na sposobie wykorzystania bazy danych z MongoDB. Przedstawiony został model ER oraz relacyjny bazy danych

5.1 Model ER

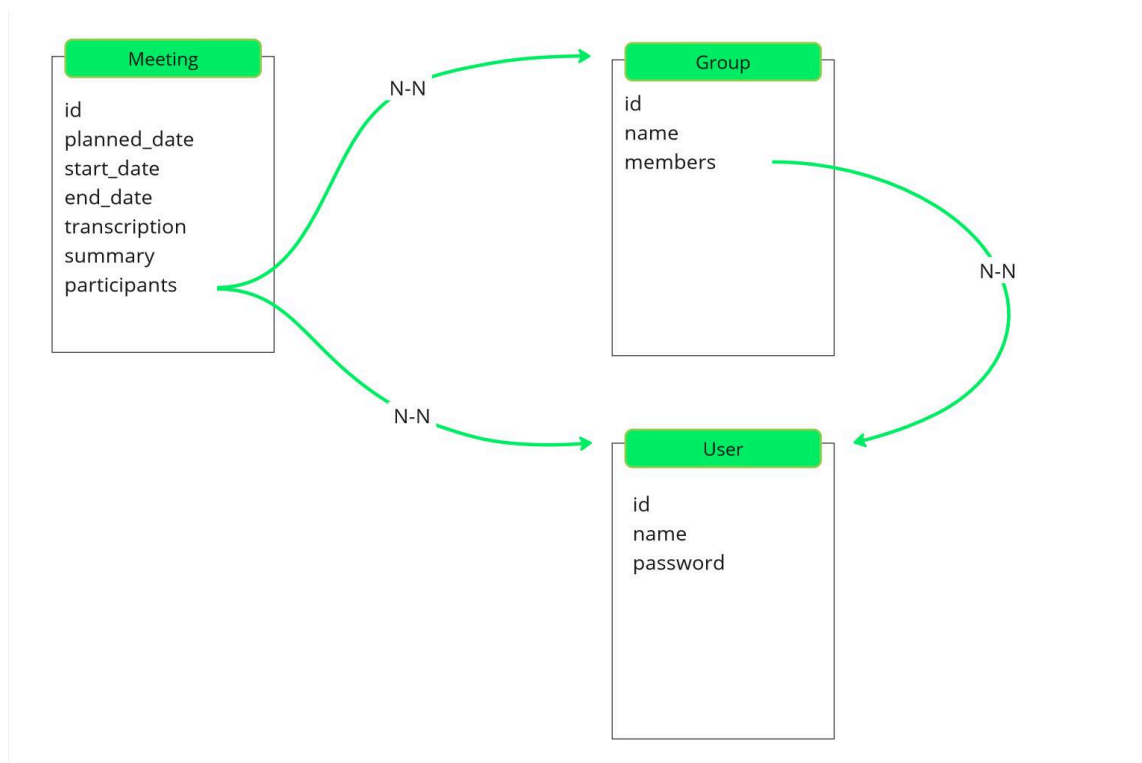
Diagram modelu ER (Rys 5.1) zawiera trzy encje występujące w projekcie, ich atrybuty oraz relacje między nimi przedstawione są poniżej.



Rys. 5.1 Diagram modelu ER

5.2 Model logiczny danych

Diagram modelu logicznego danych został przedstawiony na Rys. 5.2. Widać na nim sposób osadzania (embedding) dokumentów.



Rys. 5.2 Diagram modelu logicznego

5.3 Przetwarzanie i przechowywanie danych

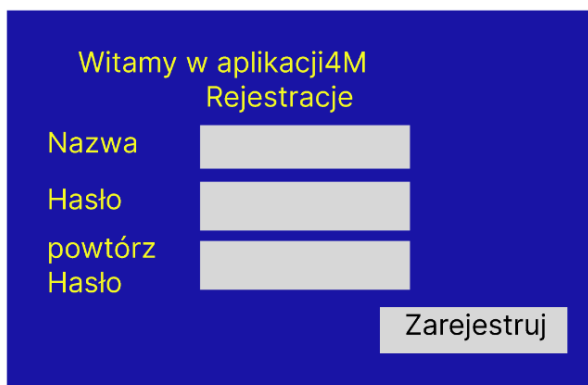
Łączenie z bazą danych MongoDB realizowane jest w aplikacji za pomocą biblioteki PyMongo. Nie planujemy projektować wyzwalaczy ani korzystać z bardziej zaawansowanych funkcjonalności dostępnych w bazie danych.

6 Projekt standardu interfejsu użytkownika

Szkice jak ma wyglądać GUI naszej aplikacji zostały zrobione w programie figma.com i dostępne są pod poniższym linkiem:

<https://www.figma.com/file/0C47bqVu2zl7MakocFkiwC/wizja-artystyczna?type=design&node-id=8-83&mode=design&t=sc4VR5RpVzpxq5VN-0>

Szkice te przedstawiają jak będzie wyglądała aplikacja z perspektywy typowego użytkownika. Ekrany rejestracji i logowania przedstawione są odpowiednio na Rys. 6.1 oraz Rys. 6.2. Po zalogowaniu użytkownik ma możliwość przeglądania dostępnych dla niego spotkań w formie kafelkowej (Rys. 6.3), gdzie będzie mógł wybrać filtry spotkań takie jak okres czasu (użytkownik podaje dwie daty od, do), grupy biorące udział w spotkaniach oraz stan spotkania (odbyte, w trakcie czy zaplanowane). Może również przejrzeć spotkania w formie miesięcznego kalendarza (Rys. 6.5). Po kliknięciu na spotkanie użytkownik przechodzi na ekran szczegółów spotkania (Rys. 6.6). Ten ekran uczestnik spotkania może jedynie przeglądać natomiast Moderator może go edytować. Podczas przebiegu spotkania wyświetlany jest ekran widoczny na Rys. 6.4.



Witamy w aplikacji4M
Rejestracje

Nazwa

Hasło

powtórz Hasło

Zarejestruj

Rys 6.1 Ekran rejestracji użytkowników

Witamy w aplikacji
4M- Logowanie

Nazwa

Hasło

Zarejestruj Zaloguj

Rys 6.2 Ekran logowania użytkowników

Grupa 1 wybrana
odbyte
zaplanowane
wszystkie

grupa2

grupa3

swobodne
spotkania

wszystkie

Filtr od do

Filtr zaplanowane

Nazwa użytkownika

Spotkanie z
08.04.2024 o 10:00

Spotkanie zaplanowane
na 11.04.2024 o 15:00

Spotkanie zaplanowane
na 15.04.2024 o 20:00

Rys 6.3 Ekran do przeglądania grup

Nazwa czas i wygaszacz ekranu (tło lub bąbelki)

Transkrypcja

Tekst próbny

Rys 6.4 Ekran przebiegu spotkania

Miesiąc i rok						
Poniedziałek	Wtorek	środa	czwartek	piątek	sobota	niedziela
		1	2	3	4	5
6	7 PZP2 15-18	8 mako 15-18 ANMA 18-20	9	10	11	12
13	14	15	16 9-12 13-14 10 15-18	17	18	19
20	21	22	23	24	25	26

Rys 6.5 Ekran kalendarza

Nazwa spotkania	nazwa spotkania
data i czas rozpoczęcia(wpierw zaplanowana następnie rzeczywista)	data i czas rozpoczęcia
lista członków	lista członków
grupy z dostępem	grupy z dostępem
data i czas zakończenia(po spotkaniu)	data i czas zakończenia
transkrypcja (po spotkaniu)	otwórz
Podsumowanie (po spotkaniu)	tekst

Rys 6.6 Ekran szczegółów spotkania

Zatwierdzam dokumentację.	
	Data i podpis Mentora