KONCERTY

Dokumentacja projektowa PZSP2

Wersja 2

25.11.2024r.

Semestr 2024Z

Zespół nr 14 w składzie:
Marcin Bagnowski
Aleksandra Buczma
Krzysztof Gólcz
Zofia Jasina

Mentor zespołu: dr inż. Krystian Radlak
Właściciel tematu: mgr inż. Katarzyna Nałęcz-Charkiewicz

Spis treści

1 Wprowadzenie

1.1 Cel projektu (z dokumentu od właściciela)

Projekt ma na celu stworzenie aplikacji internetowej wspomagającej zarządzanie zespołami muzycznymi oraz orkiestrami. Użytkownikami aplikacji będą członkowie orkiestr/zespołów oraz ich dyrygenci/kapelmistrzowie/chórmistrzowie (konieczny podział uprawnień użytkowników).

1.2 Wstępna wizja projektu

Aplikacja do zarządzania zespołami muzycznymi i orkiestrami ma na celu wsparcie grup muzycznych. Jej główne zadanie to ułatwienie komunikacji wewnątrz zespołu, koordynacji prób, koncertów oraz dostęp do zasobów, takich jak partytury czy nagrania.

Kluczowym elementem aplikacji jest system zarządzania uprawnieniami, który pozwala na przypisanie ról użytkowników zgodnie z ich pozycją w zespole. Dyrygenci, kapelmistrzowie i chórmistrzowie będą mieli pełne uprawnienia, umożliwiające im zarządzanie składem zespołu, zasobami muzycznymi, harmonogramem wydarzeń oraz ogłoszeniami. Będą mogli organizować np.: próby ogólne, jak również te dedykowane dla poszczególnych podgrup, takich jak instrumenty dęte, smyczkowe, czy chóry. Członkowie zespołów będą mieć dostęp do materiałów muzycznych przypisanych do ich roli oraz szczegółów dotyczących terminów prób i koncertów. Koordynatorzy z kolei, w odróżnieniu od zwykłych użytkowników, będą mogli zarządzać komunikacją poprzez dodawanie ogłoszeń.

Aplikacja zapewnia dystrybucję materiałów muzycznych, w której dyrygenci będą mogli udostępniać partytury, nagrania i notatki, dostosowane do potrzeb poszczególnych grup instrumentalnych oraz indywidualnych muzyków. Dzięki temu członkowie zespołów będą mieli stały dostęp do aktualnych zasobów, co umożliwi im samodzielne przygotowanie się przed próbami. System powiadomień przypomni o nadchodzących wydarzeniach, nowych materiałach czy zmianach w harmonogramie, umożliwiając bieżące śledzenie wszelkich aktualizacji.

Komunikacja w ramach aplikacji odbywać się będzie poprzez system ogłoszeń wewnętrznych, umożliwiający szybki kontakt między członkami zespołu a dyrygentem. Będzie to szczególnie przydatne w sytuacjach wymagających szybkich zmian lub dostosowania materiałów do potrzeb konkretnego koncertu.

Aplikacja będzie dostępna przez przeglądarkę internetową - przystosowana także pod kątem urządzeń mobilnych, dzięki czemu użytkownicy będą mogli korzystać z niej zarówno na komputerach, jak i smartfonach.

2 Metodologia wytwarzania

Podstawowym narzędziem pracy zespołowej jest repozytorium kodu na wydziałowym gitlabie:

- https://gitlab-stud.elka.pw.edu.pl/kgolcz/pzsp2-megalodony

Będzie ono wykorzystywane do przechowywania plików projektowych i wspólnej pracy nad nimi, a także do definiowania i przydzielania zadań. Przydzielanie zadań odbywać się będzie na bieżąco, biorąc pod uwagę aktualne możliwości oraz umiejętności członków zespołu.

Komunikacja między członkami zespołu odbywa się przy pomocy narzędzi takich jak Messenger oraz Discord. Do kontaktu z mentorem, klientem oraz ekspertami używane są także platformy Teams i Leon.

Platforma Miro służy jako przestrzeń robocza do tworzenia modeli i diagramów.

W celu określenia wrodzonych predyspozycji członków zespołu, przeprowadzony został test Belbina. Wynika z niego, że w zespole występują zróżnicowane role (3x Completer Finisher, 2x Shaper, Implementer, Monitor Evaluator, Plant, 2x Coordinator, Team Worker). Dotkliwym brakiem może okazać się brak naturalnego Resource Investigator-a.

Zgodnie z modelem kaskadowym pracy nad projektem, implementacja rozwiązania poprzedzona jest planowaniem, analizą wymagań oraz modelowaniem, a nastąpi po niej faza testowania.

3 Analiza wymagań

3.1 Wymagania użytkownika i biznesowe

- wymagania biznesowe:

Klient potrzebuje aplikacji internetowej wspomagającej zarządzanie zespołami muzycznymi oraz orkiestrami, aby ułatwić komunikację oraz dystrybucję zasobów między członkami.

- wymagania użytkowe:

Wymagania użytkownika, dostarczone przez klienta, prezentuje tabela 1.

Priorytet	Rola	Wymaganie
High	Kierownik grupy	Zarządzanie grupą (nazwa, opis, informacje dodatkowe takie jak link do strony internetowej czy profilów w mediach społecznościowych).
High	Kierownik grupy	Zarządzanie użytkownikami, w tym tworzenie grup i podgrup, zapraszanie nowych użytkowników do grup/podgrup.
High	Kierownik grupy	Wysyłanie ogłoszeń: do całej grupy, do podgrupy, do wybranych członków.
High	Kierownik grupy	Możliwość integracji ogłoszeń z mailem, co powoduje wysłanie ogłoszenia także na adresy e-mail członków grupy/podgrupy.
High	Kierownik grupy	Zarządzanie utworami (tytuł, opis, pliki z nutami, w tym możliwość dodawania plików per instrument, opcjonalne przypisanie do koncertu/koncertów).
High	Kierownik grupy	Planowanie koncertów i prób (tytuł, opis, repertuar, czas, miejsce, wybór całej grupy/podgrupy, bądź poszczególnych użytkowników).
High	Zwykły użytkownik	Oglądanie planu prób.
High	Zwykły użytkownik	Oglądanie planu koncertów.
High	Zwykły użytkownik	Przeglądanie repertuaru i pobieranie nut.
High	Zwykły użytkownik	Przeglądanie kalendarza wydarzeń.
Medium	Zwykły użytkownik	Przeszukiwanie repertuaru wg różnych kryteriów: nazwa utworu, koncert, do którego utwór jest przypisany, rodzaj instrumentu.
Low	-	Integracja kalendarza wydarzeń z Google Calendar.

Tab. 1. Wymagania użytkownika

- wymagania systemowe:

opisane przez wymagania funkcjonalne i niefunkcjonalne, patrz podpunkt 3.2.

- **aktorzy w projekcie:**

- główni: UŻYTKOWNIK (trzy poziomy uprawnień)
  - kierownik grupy (administrator, kapelmistrz, chórmistrz)
  - moderator (koordynator)
  - zwykły użytkownik (muzyk)
- drugorzędni: SYSTEMY ZEWNĘTRZNE
  - Google Sign-In API
  - Google Calendar
  - e-mail API
  - Google Drive API
  - Google Drive

3.2 Wymagania funkcjonalne i niefunkcjonalne

- wymagania funkcjonalne:

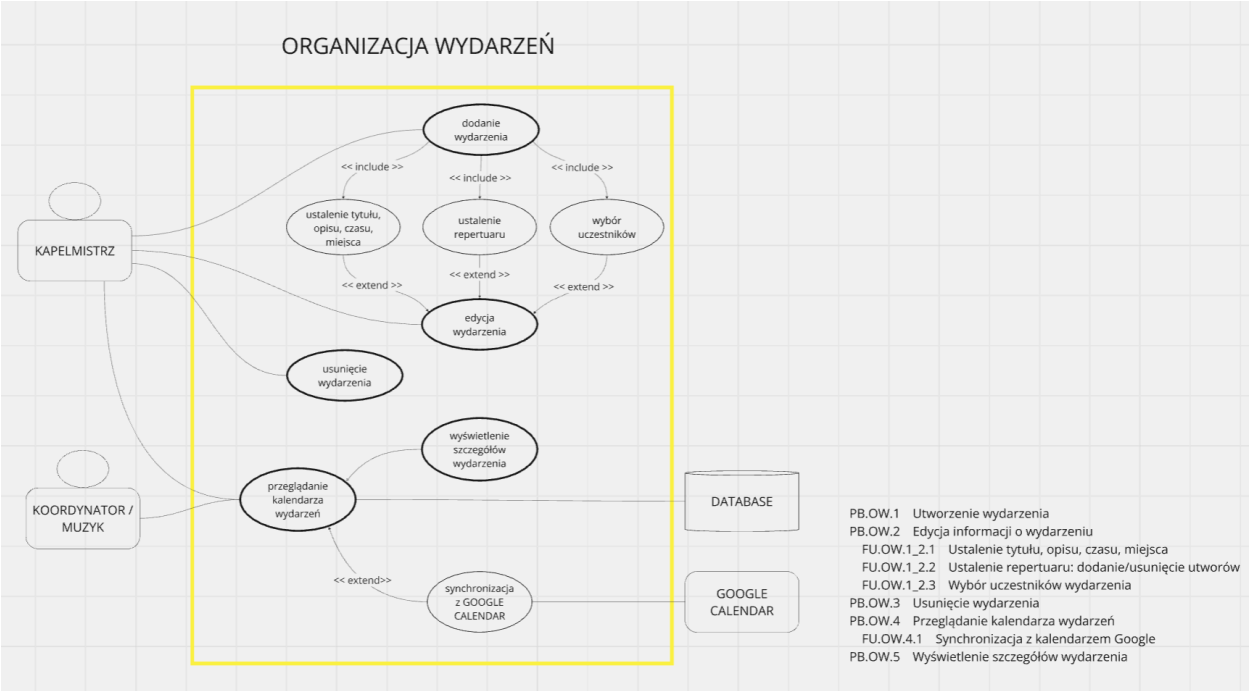
- Aplikacja powinna umożliwiać administratorom tworzenie grup i podgrup.
- Aplikacja powinna umożliwiać administratorom zarządzanie członkostwem użytkowników.
- Aplikacja powinna umożliwiać zarządzanie nazwą, opisem i dodatkowymi informacjami grup i podgrup (np. link do strony internetowej, profil w mediach społecznościowych).
- System powinien umożliwiać zapraszanie nowych użytkowników do grup i podgrup.
- Aplikacja powinna pozwolić administratorom i moderatorom na wysyłanie ogłoszeń do całej grupy, wybranej podgrupy lub pojedynczych członków.
- System powinien zapewnić wyświetlanie zamieszczonych ogłoszeń.
- System zapewni możliwość wysyłania powiadomień mailowych członkom grup/podgrup o nowym ogłoszeniu.
- System powinien umożliwiać dodawanie utworów muzycznych z nazwą, opisem, plikami nut oraz nagraniami do utworów.
- System powinien obsługiwać plik w formatach .mp4, .pdf, .m4a, .mid, .mus, .txt, .png, .musx, .jpg, .zip, .doc, .mp3, .wav, .wma, .tif.
- System powinien umożliwiać przypisanie pliku nut dla poszczególnych instrumentów oraz do poszczególnych użytkowników.
- System powinien umożliwiać przypisanie utworów do jednego lub kilku koncertów.
- Aplikacja powinna umożliwiać tworzenie wydarzeń takich jak koncerty i próby.
- Aplikacja zapewni zarządzanie wydarzeniami poprzez dodanie tytułu, opisu, repertuaru, daty, czasu i miejsca.
- System zapewni administratorowi możliwość wyboru uczestników wydarzenia (cała grupa, podgrupa lub poszczególni członkowie).
- System powinien udostępniać widok planu prób oraz planu koncertów.
- Aplikacja powinna umożliwiać członkom przeglądanie repertuaru wydarzenia.
- System zapewni filtrowanie katalogu utworów według nazwy utworu, koncertu, do którego utwór jest przypisany, oraz instrumentu.
- Aplikacja powinna umożliwiać pobieranie przypisanych nut uprawnionym członkom zespołu.
- System powinien posiadać funkcję kalendarza, pozwalającą użytkownikom przeglądać wszystkie zaplanowane wydarzenia.
- System powinien umożliwiać synchronizację kalendarza wydarzeń z Google Calendar.

- wymagania niefunkcjonalne:

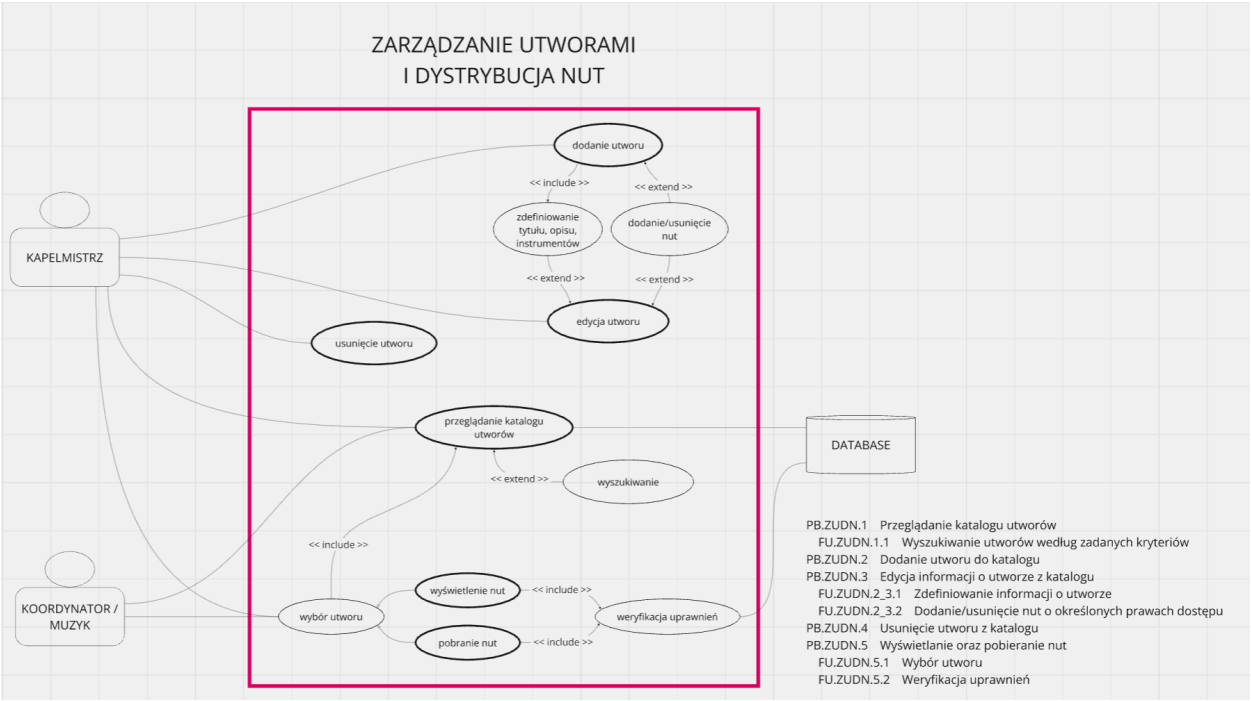
- Przenośność: Aplikacja powinna umożliwiać dostęp z różnych urządzeń i systemów operacyjnych, takich jak Windows, Linux, iOS i Android..
- Optymalizacja dla urządzeń mobilnych: Interfejs i wydajność aplikacji powinny być dostosowane do użytkowników korzystających z urządzeń mobilnych.
- Skalowalność: Aplikacja powinna wspierać rozbudowę i sprawne działanie dla 1000 użytkowników oraz kilku orkiestr (grup).
- Wydajność: Aplikacja powinna działać płynnie i bez zakłóceń nawet przy obciążeniu rzędu setek aktywnych użytkowników.
- Pojemność: Baza danych zapewni miejsce na pliki o łącznym rozmiarze do 15 GB (łącznie dla kilku grup, przy założeniu obsługi 5 grup każda po 2,6 GB średnio).
- Bezpieczeństwo: System powinien zapewniać ochronę danych użytkowników i być zgodny z obowiązującymi przepisami o ochronie danych osobowych.
- Wsparcie dla wielu języków: Interfejs aplikacji powinien być dostępny początkowo w języku polskim i angielskim. Z możliwością późniejszego łatwego rozszerzenia dla większej liczby języków.
- Dokumentacja: Aplikacja powinna posiadać pełną dokumentację techniczną oraz instrukcje instalacji i wdrożenia, umożliwiające łatwe wdrożenie i utrzymanie systemu.

3.3 Przypadki użycia

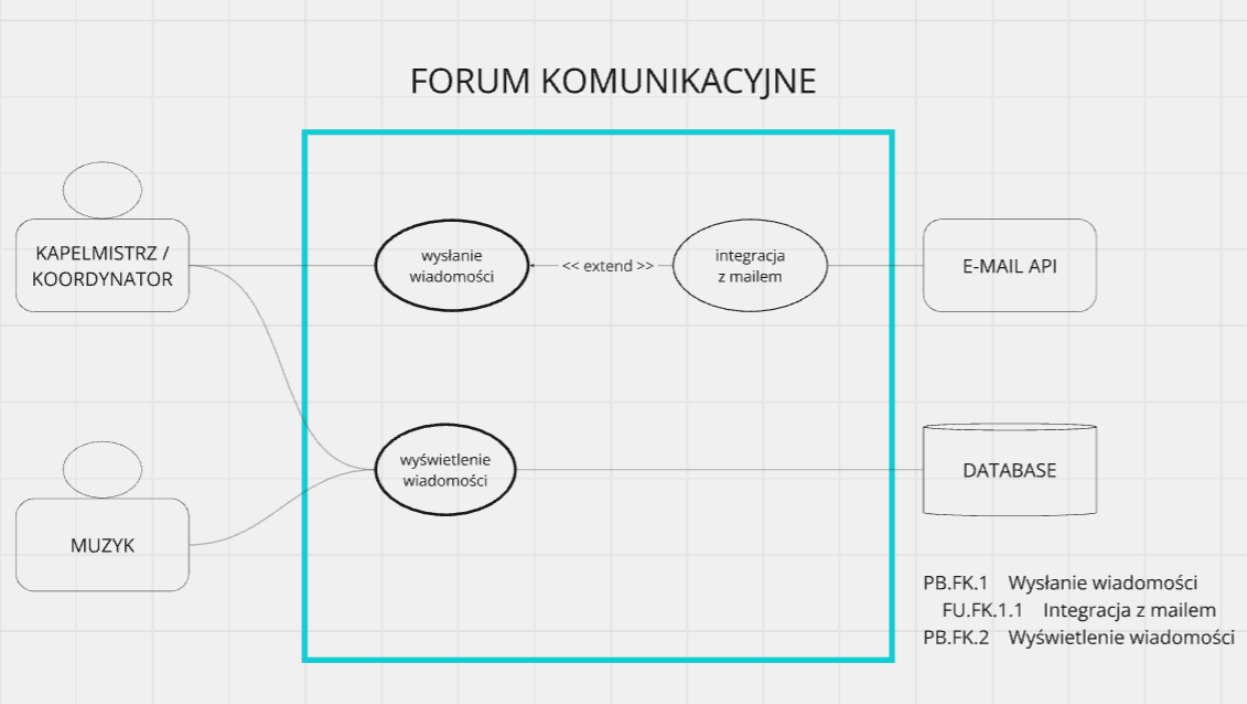
Najważniejsze przypadki użycia zamodelowane zostały na diagramach. Dotyczą one organizacji wydarzeń (zdj. 1.), zarządzania utworami i dystrybucji nut (zdj. 2.), forum komunikacyjnego (zdj. 3.) oraz zarządzania grupami (zdj. 4.).



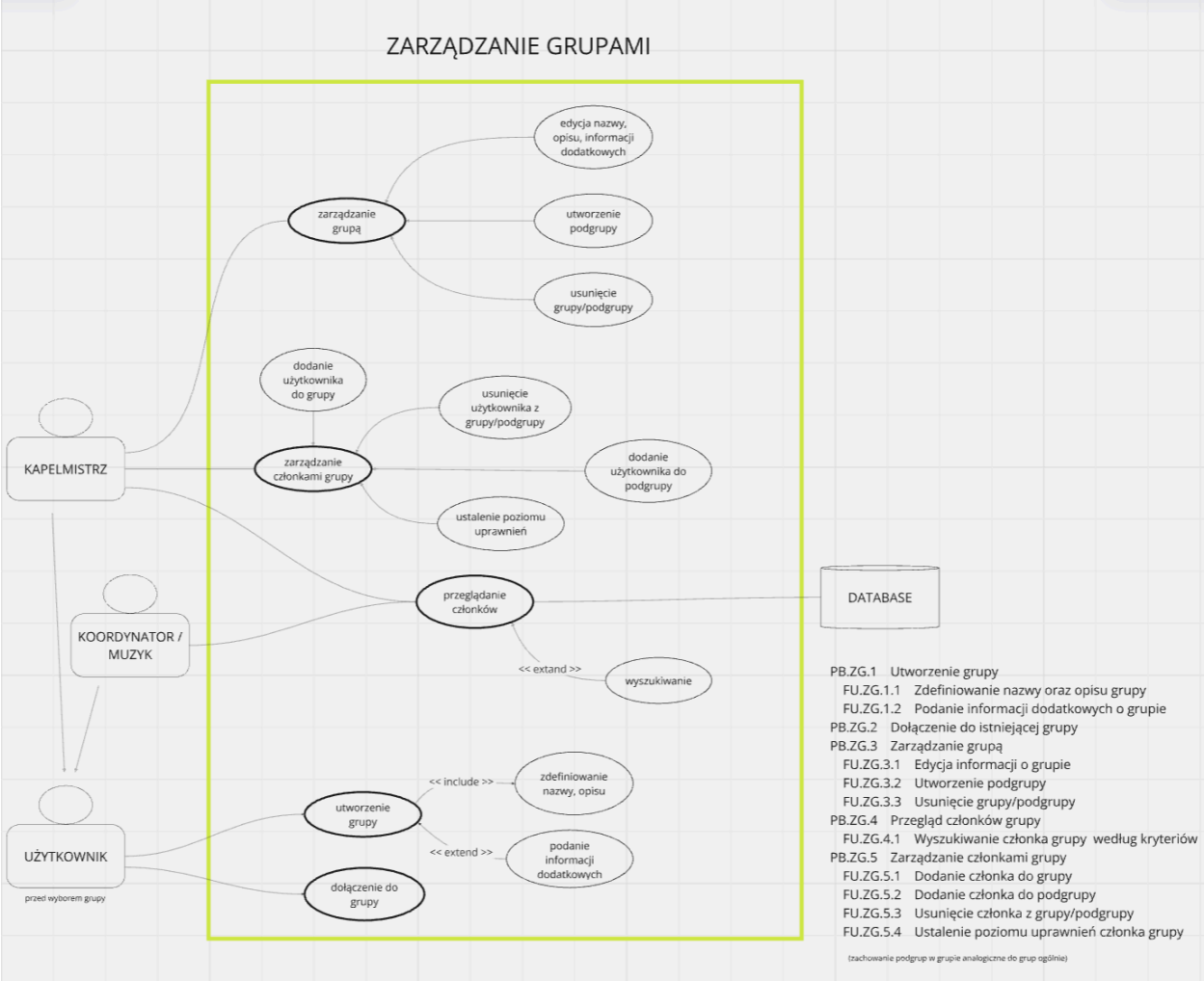
Zdj. 1. Diagram przypadków użycia - organizacja wydarzeń



Zdj. 2. Diagram przypadków użycia - zarządzanie utworami i dystrybucja nut



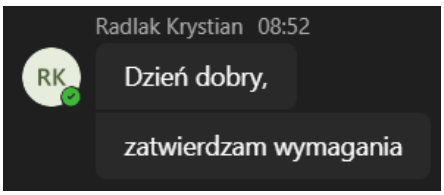
Zdj. 3. Diagram przypadków użycia - forum komunikacyjne

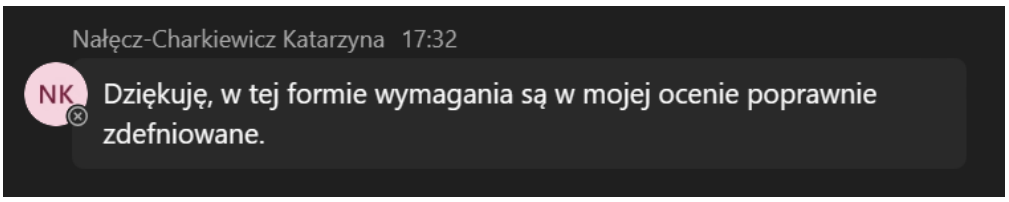


Zdj. 4. Diagram przypadków użycia - zarządzanie grupami

3.4 Potwierdzenie zgodności wymagań

Zatwierdzenie wymagań przez mentora oraz właściciela tematu



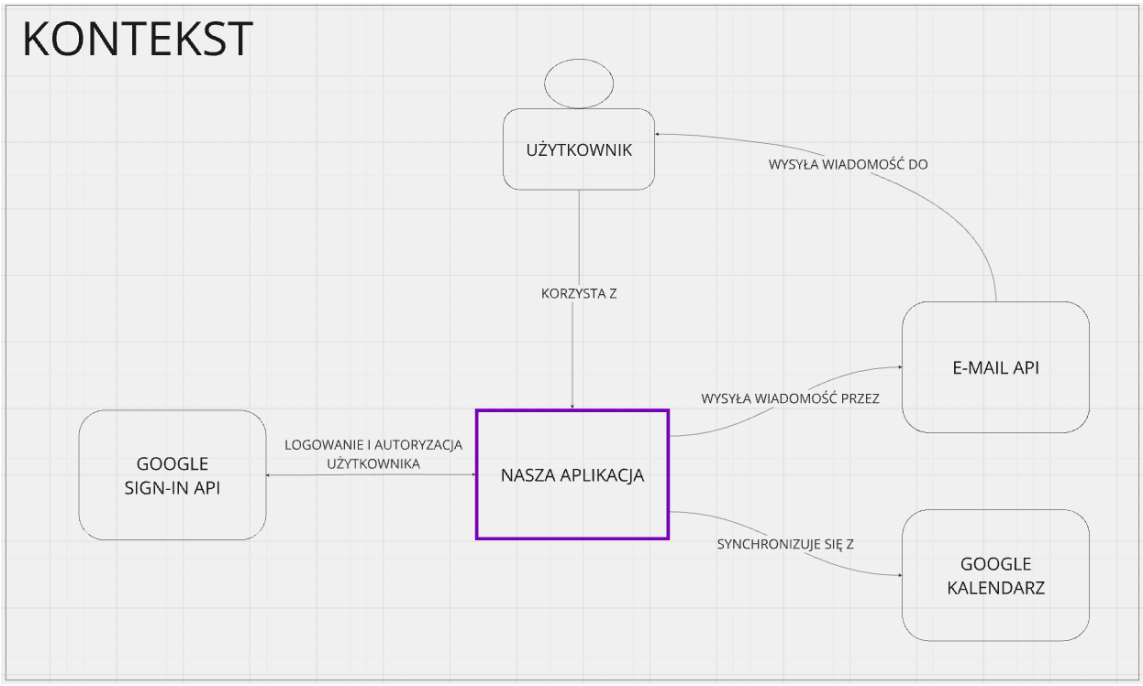


4 Definicja architektury

- Plan struktury systemu: c4

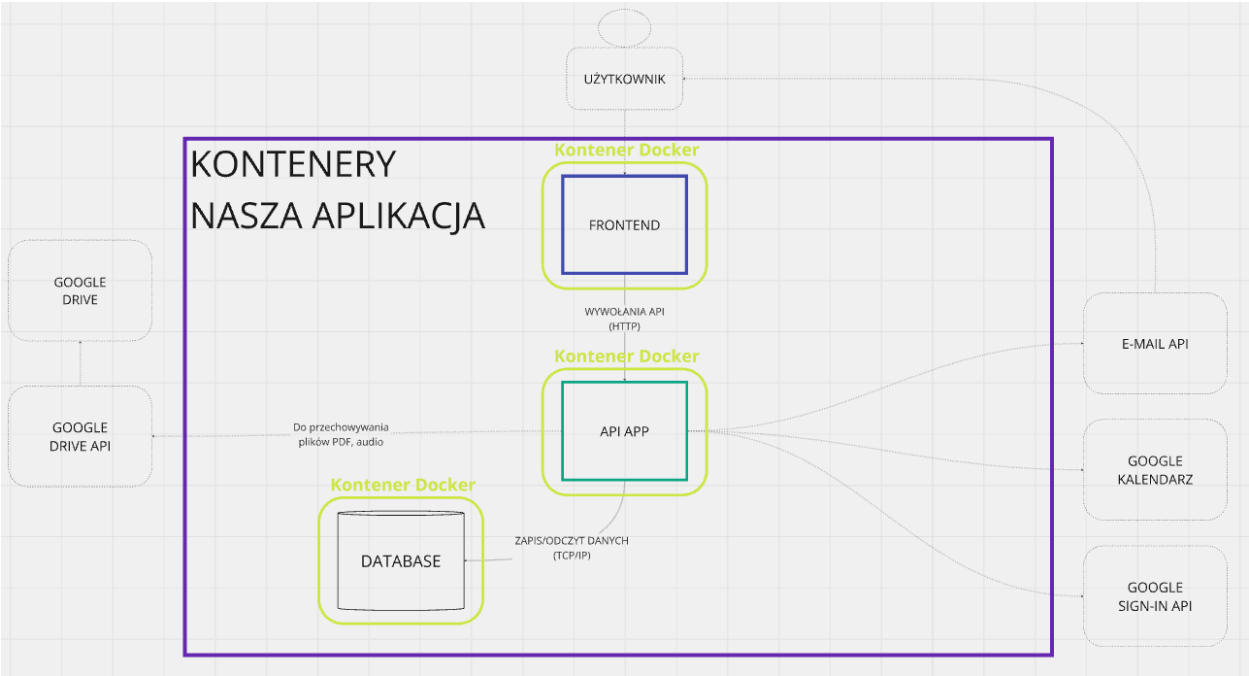
Link do Miro z całym modelem C4.

- Warstwa C1 - Kontekst (zdj 5.) - zewnętrzny kontekst opisu naszej aplikacji.
    - Użytkownik łączy się z naszą aplikacją z wykorzystaniem swojego konta google, które jest kluczowe w dalszym działaniu aplikacji
    - Użytkownik przy procesie logowania zostaje zautoryzowany przez naszą aplikację korzystając z Google Sign-In Api
    - Do przesyłania maili na konto Google (pocztę Gmail) użytkownika aplikacja korzysta z E-Mail Api
    - Aplikacja synchronizuje się z kalendarzem Google korzystając z Google Calendar Api



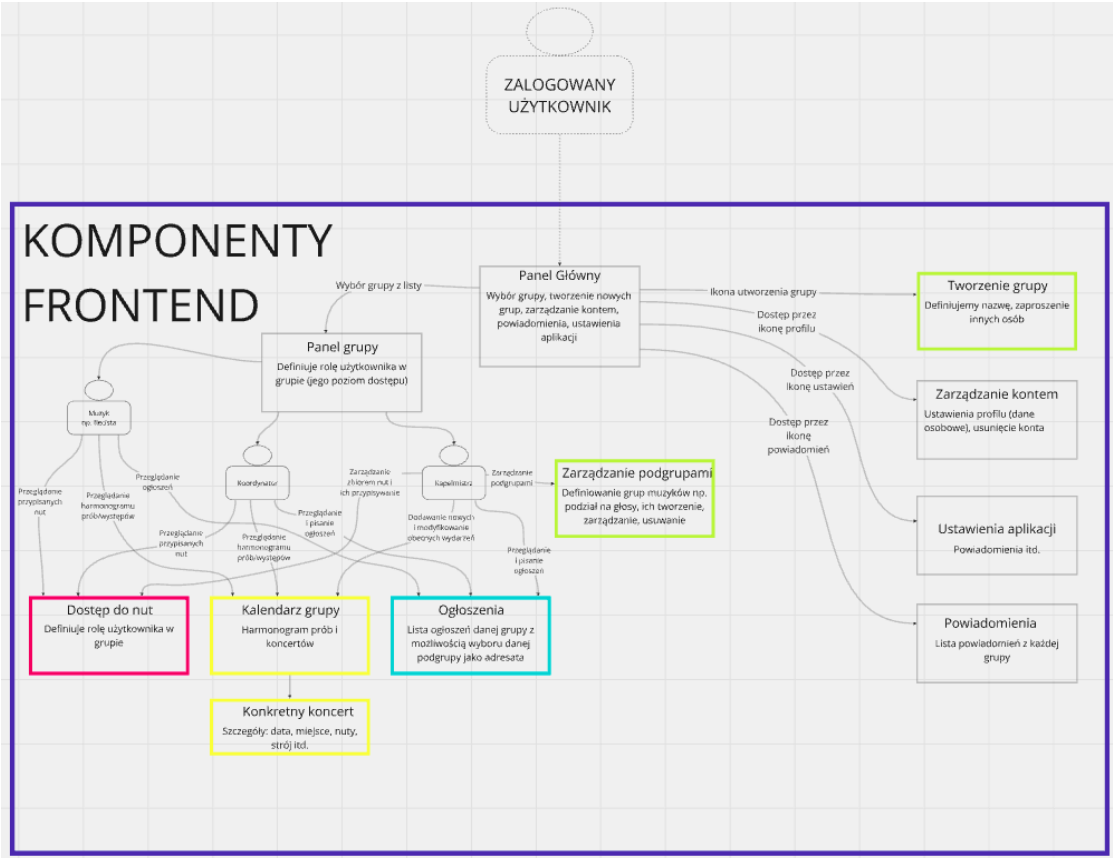
Zdj. 5. pierwsza warstwa modelu C4

- Warstwa C2 (Zdj. 6) - Kontenery - struktura wewnętrzna naszej aplikacji. Wykorzystamy **szablon architektoniczny architektury trójwarstwowej**. Każdy z trzech kluczowych komponentów aplikacji: Frontend, API (serwer), Baza danych znajduje się w oddzielnym kontenerze Docker. Te porozumiewają się między sobą z wykorzystaniem protokołu HTTP. Co warto zaznaczyć nasza aplikacja ma dwie bazy danych z czego pierwsza jest bazą relacyjną - przechowuje dane systemu, ale bezpośrednio nie przechowuje plików, a jedynie informacje o nich. Druga baza to chmura Google Drive, na której przetrzymywane są wszystkie pliki. Połączenie z nią jest realizowane za pośrednictwem Google Drive Api. Zatem nasza baza przechowuje klucz ID do plików trzymanych w chmurze Google Drive.



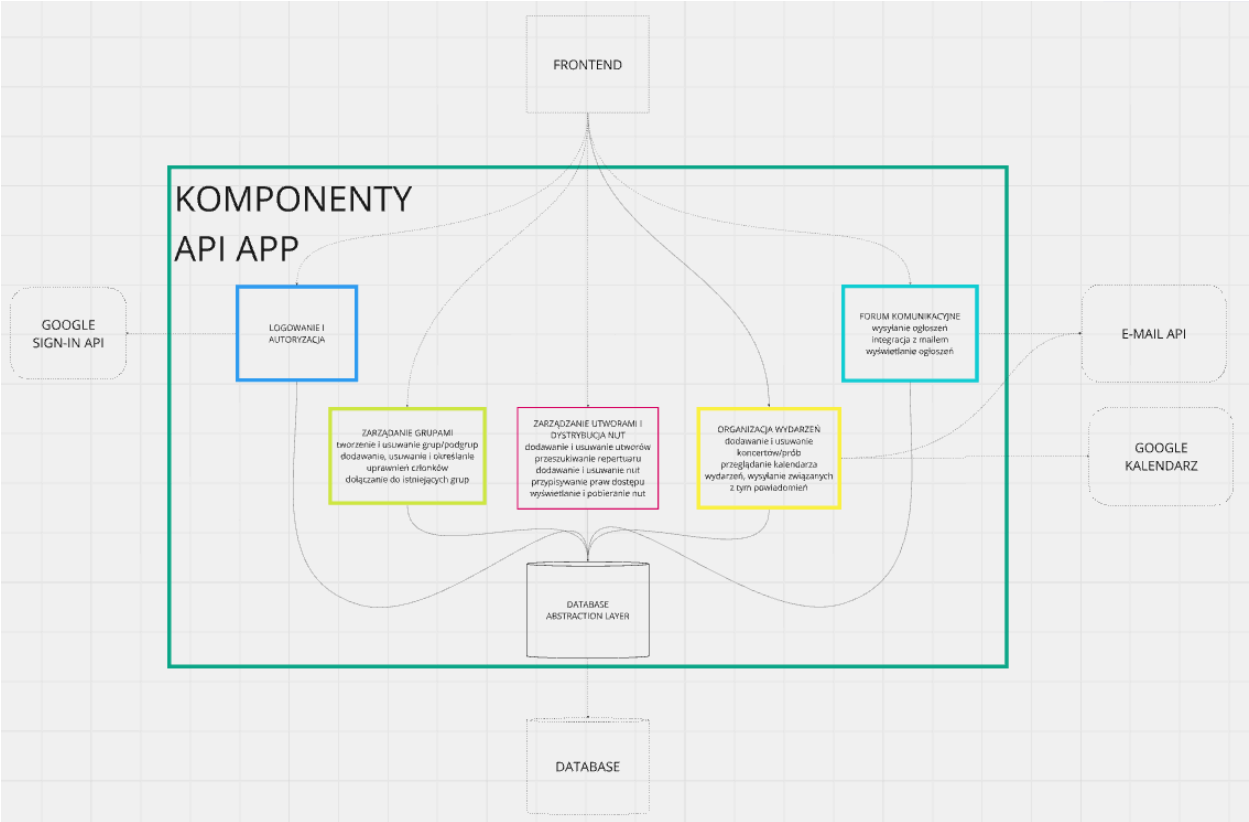
Zdj. 6. druga warstwa modelu C4

- Warstwa C3 - Komponenty:
    - Frontend (Zdj 7.) - służy do bezpośredniej interakcji z użytkownikiem. Wykonywane polecenia przez użytkownika wysyłane są do drugiego komponentu - API App, z którym się bezpośrednio komunikuje. Bazowym oknem jest tutaj Panel Główny, który pozwala na szybką orientację w systemie i zawiera odnośniki do poszczególnych funkcji:
        - Panel Grupy - sprawdza rolę użytkownika w grupie (Muzyk, Koordynator, Kapelmistrz) po czym udostępnia okna dotyczące konkretnej grupy:
            - Dostęp do nut - lista przypisanych nut oraz możliwość zarządzania przydziałem nut
            - Kalendarz grupy - wyświetla harmonogram prób i koncertów
            - Ogłoszenia - lista ogłoszeń grupy
            - Zarządzanie podgrupami - pozwala wyświetlać obecne grupy muzyków, nimi zarządzać oraz tworzyć nowe
        - Tworzenie nowej grupy - pozwala na utworzenie nowej grupy, zdefiniowanie jej nazwy oraz zaproszenie osób do grupy
        - Zarządzanie kontem - służy do wyświetlania i modyfikacji swoich danych w systemie oraz do usunięcia konta
        - Ustawienia aplikacji - pozwala na zmianę ustawień np. dotyczących wysyłania powiadomień
        - Powiadomienia - lista wszystkich powiadomień ze wszystkich grup oraz tych technicznych z aplikacji



Zdj. 7. komponent Frontend z trzeciej warstwy modelu C4

- API APP (serwer) (Zdj 8.) - odpowiada za techniczną realizację funkcjonalności systemu oraz wykonuje odpowiednie akcje w odpowiedzi na polecenia użytkownika. Łączy się bezpośrednio z frontendem protokołem HTTP oraz wykonuje odpowiednie polecenia do baz danych w zależności od wybranej akcji. Polecenia do baz danych są wykonywane przy użyciu warstwy abstrakcji (Data Abstraction Layer). Główne funkcjonalności to:
  - Logowanie i autoryzacja - weryfikuje użytkownika korzystając z Google Sign-In Api
  - Zarządzanie grupami - obsługuje mechanizm zarządzania grupami oraz ich tworzenia i usuwania, również odpowiada za zarządzanie poziomami dostępu poszczególnych użytkowników w grupie
  - Zarządzanie utworami i dystrybucja nut - realizuje dodawanie i usuwanie nut z systemu, przypisywanie praw dostępu do nut, wyświetlanie i pobieranie nut
  - Organizacja wydarzeń - dodawanie, usuwanie i modyfikacja wydarzeń, przeglądanie kalendarza, aktualizuje o wprowadzone modyfikacje kalendarz google użytkownika oraz wysyła mu stosowne powiadomienie mailowo i w aplikacji
  - Forum komunikacyjne - realizuje wysyłanie ogłoszeń w aplikacji oraz na maila, wyświetla ogłoszenia

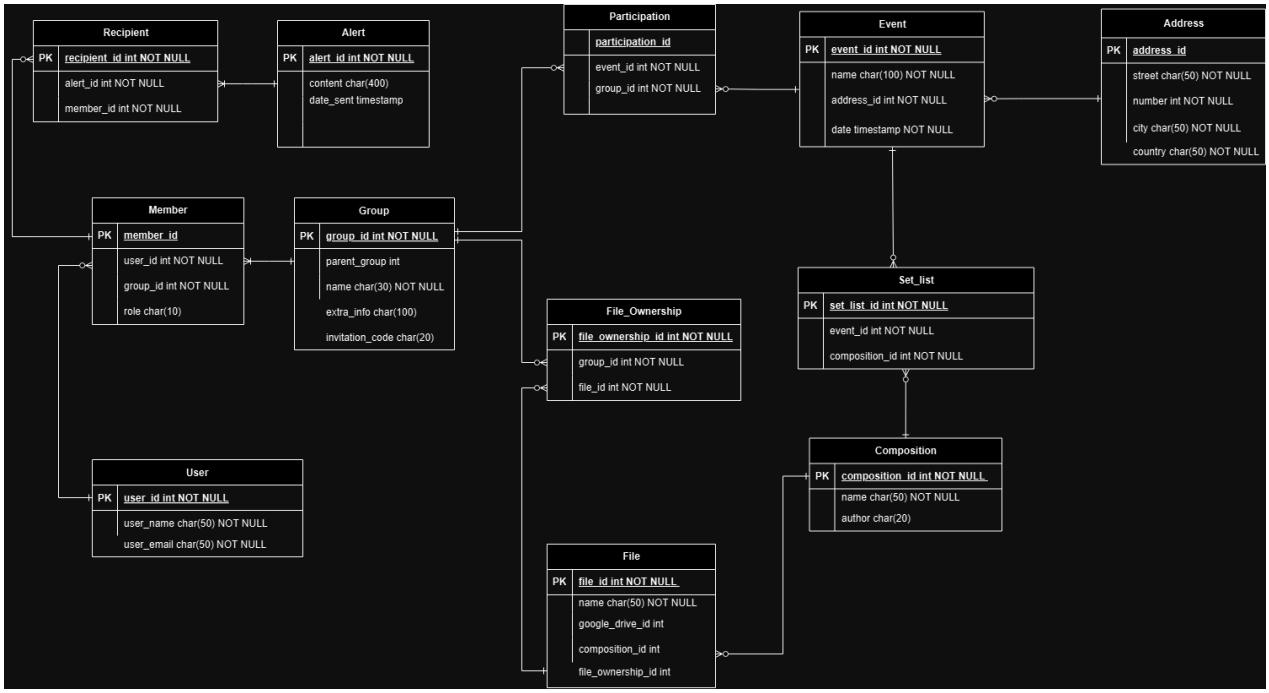


Zdj. 8 komponent API APP (serwer) z trzeciej warstwy modelu C4

- Baza danych - składa się z dwóch komponentów:
  - Baza relacyjna - przechowuje wszystkie informacje o systemie (grupy, użytkownicy, powiadomienia, informacje o plikach, wydarzenia) w odpowiednich tabelach, również przechowuje klucze ID plików z chmury Google Drive
  - Chmura Google Drive - przechowuje wszystkie pliki, do których można się odwołać przy użyciu ID konkretnego pliku, dostęp przy użyciu Google Drive Api

5 Dane trwałe

5.1 Model logiczny danych



Zdj. 9. Logiczny model relacyjnej bazy danych

5.2 Przetwarzanie i przechowywanie danych

- Planowana relacyjna baza danych: PostgreSQL

- Dostęp do bazy danych postgresql: SQLAlchemy

- Dodatkowa baza danych przeznaczona do obsługi plików: Postanowiliśmy użyć Google Drive do dodawania/pobierania plików. Polegać to będzie na tym, że w bazie relacyjnej wspomnianej wcześniej będzie tabela File. W tej tabeli będzie pole google_drive_id, które generowane jest przy każdym dodaniu pliku do Google Drive, przy użyciu Google API. Dzięki temu będziemy mogli wysyłać zapytanie z naszego serwera do naszej bazy danych, skąd otrzymamy google_drive_id, które zostanie użyte przez nasz serwer do wysłania zapytania do Google Drive API i na przykład pobrania pliku o danym id.

6 Specyfikacja analityczna i projektowa

- https://gitlab-stud.elka.pw.edu.pl/kgolcz/pzsp2-megalodony

- Metody realizacji:

- AngularJS/TypeScript - Frontend
- Python - Backend
- VScode, środowiska wirtualne (.venv)
- Środowisko wdrażania - docker

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

Opcjonalnie model struktury systemu (diagram wdrożenia)

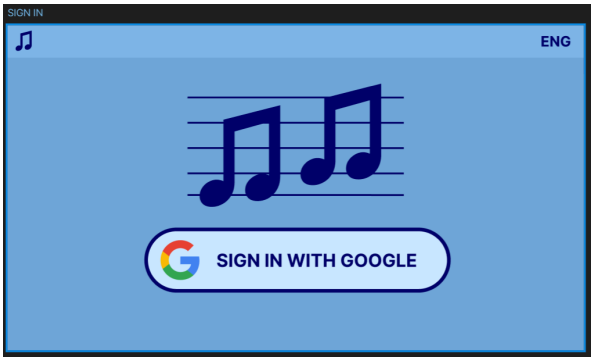
Opcjonalnie specyfikacja realizacji przypadków użycia: diagramy sekwencji lub współpracy

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

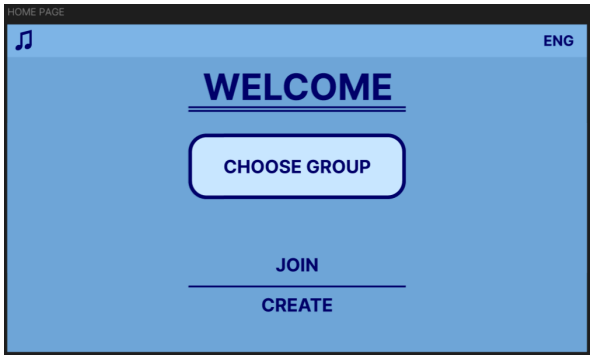
7 Projekt standardu interfejsu użytkownika

Interfejs użytkownika wstępnie zaprojektowany został z wykorzystaniem aplikacji figma.

Pierwsze, co użytkownik widzi, to logo aplikacji oraz możliwość zalogowania przez Google (zdj. 10.). Po udanym zalogowaniu ma możliwość wyboru grupy, do której już należy, bądź też dołączenia do lub utworzenia nowej (zdj. 11.). Przyciski te otwierają na stronie nakładki, oferujące wymienione funkcjonalności (zdj. 12.).



Zdj. 10. Ekran logowania

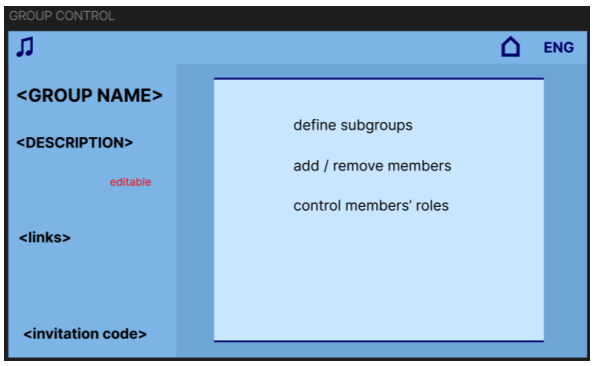


Zdj. 11. Ekran powitalny z wyborem grupy

Zdj. 12. Lista grup do wyboru, okienko dołączania, formularz utworzenia grupy

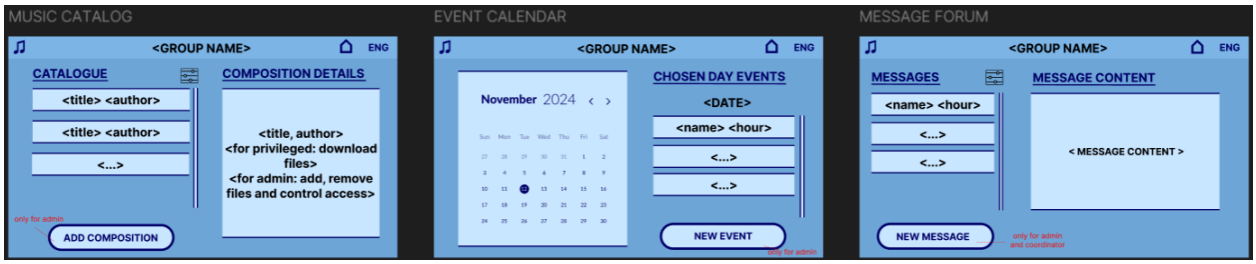
Po wybraniu grupy, użytkownik ma dostęp do kolejnego ekranu powitalnego (zdj. 13.), gdzie może zobaczyć dostępne mu informacje o grupie oraz wybrać, co chce zobaczyć dalej. Strona kontroli grupy (zdj. 14.) dostępna jest jedynie dla uprawnionych członków. Administratorzy mogą zobaczyć na niej szczegółowe informacje - jak kod zaproszeniowy, listę członków oraz podgrup - oraz nimi zarządzać.

Zdj. 13. Ekran powitalny grupy

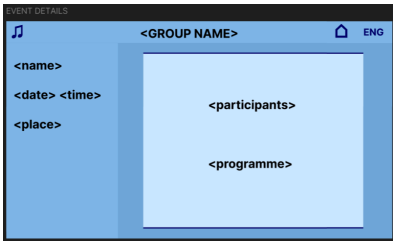


Zdj. 14. Strona kontroli grupy

Katalog muzyczny, kalendarz wydarzeń oraz forum komunikacyjne (zdj. 15.) dostępne są dla wszystkich, jednak uprawnieni członkowie mają na tych stronach większe możliwości - jak dodawanie utworów czy wydarzeń oraz pisanie ogłoszeń.



Zdj. 15. Katalog muzyczny, kalendarz wydarzeń, forum komunikacyjne



Z kalendarza wydarzeń przejść można także do strony ze szczegółami wybranego wydarzenia (zdj. 16.).

Zdj. 16. Ekran wydarzenia

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]

Rysunek (1 slajd) podsumowujący projekt do umieszczenia w repozytorium projektów PZSP2

ii. 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