



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Imię i nazwisko studenta: Andrzej Świerczyński

Nr albumu: 171573

Poziom kształcenia: Studia drugiego stopnia

Forma studiów: stacjonarne

Kierunek studiów: Informatyka

Specjalność: Aplikacje rozproszone i systemy internetowe

PRACA DYPLOMOWA MAGISTERSKA

Tytuł pracy w języku polskim: System do kooperacyjnej pracy nad tworzeniem muzyki

Tytuł pracy w języku angielskim: A system for collaborative work on creating music

Opiekun pracy: dr hab. inż. Julian Szymański

STRESZCZENIE

Tematem pracy jest zbadanie dostępnych na rynku rozwiązań do zdalnego nauczania i oferowanych przez nie funkcjonalności. Głównym założeniem projektowym jest zaprojektowanie i zimplementowanie aplikacji posiadającej wszystkie funkcjonalności niezbędne do takiego typu nauczania. Program ma być dostępny w formie przeglądarkowej oraz jako aplikacja mobilna. Wymagania projektu zostały zebrane poprzez wywiad jak również przez przegląd literatury, a następnie wielokrotne prototypowanie i eksperymenty.

Zbadana została większość dużych rozwiązań w zakresie zdalnego nauczania dostępnych na początku 2022 roku w celu ewaluacji dostosowania tych produktów do faktycznych wymagań stawianych przez edukację muzyczną. Przeanalizowano też postrzeganie przez studentów i pedagogów kursów zdalnych i tradycyjnych oraz przygotowanie instytucji do wprowadzenia zdalnego nauczania.

Stworzona aplikacja połączyła wiele funkcjonalności niedostępnych razem w jednym produkcie lub dotąd niespotykanych. Analiza produktów do zdalnego nauczania wykazała brak wielu właściwości niezbędnych do uczenia muzyki i konieczność używania dodatkowych rozwiązań.

Słowa kluczowe: nauka muzyki, zdalne nauczanie, WebRTC

Dziedzina nauki i techniki, zgodnie z wymogami OECD: 1.2 Informatyka i nauki informacyjne

ABSTRACT

The topic of the thesis is to investigate the remote teaching solutions available on the market and the functionalities they offer. The main project idea is to design and implement an application with all the functionalities necessary for this type of teaching. The programme is to be available in browser form as well as a mobile application. The requirements of the project were gathered through an interview as well as through a literature review, followed by multiple prototyping and experiments.

Most of the large-scale remote learning solutions available at the beginning of 2022 were examined in order to evaluate the adaptation of these products to the actual requirements of music education. Students' and educators' perceptions of remote and traditional courses were also analysed, as was the institution's preparedness for the introduction of remote teaching.

The application developed combined many functionalities not available together in one product or not available at all. The analysis of the remote teaching products showed that many of the features needed to teach music were missing and that additional solutions were needed.

Keywords: learning music, remote learning, WebRTC

Field of science and technology in accordance with OECD requirements: 1.2 Computer and information sciences

SPIS TREŚCI

WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW	9
1. WSTĘP I CEL PRACY	10
1.1. Cel i zakres pracy	10
1.2. Założenia projektowe	10
2. PRZEGŁĄD ISTNIEJĄCYCH ROZWIĄZAŃ	12
2.1. Uczenie zdalne	12
2.2. Aplikacje	14
2.2.1. Microsoft Teams	14
2.2.2. Zoom	15
2.2.3. Discord	17
2.2.4. Moodle	18
2.2.5. Sibelius	20
2.3. Technologie	21
2.3.1. WebRTC	21
2.4. Przegląd aktualnego stanu technologii	26
3. PROJEKT I IMPLEMENTACJA	29
3.1. Specyfikacja wymagań	29
3.2. Projekt rozwiązania	29
3.2.1. Serwer aplikacji	29
3.2.2. Frontend	29
3.2.3. Bazy danych	30
3.3. Interfejs użytkownika	30
3.4. Aplikacja mobilna	33
3.5. Rodzaje użytkowników	37
3.5.1. Uczeń	37
3.5.2. Nauczyciel	38
3.5.3. Administrator	38
3.6. Implementacja i użyte technologie	38
3.6.1. Spring Boot	38
3.6.2. Angular	40

3.6.3. PostgreSQL	41
3.6.4. MinIO	42
3.6.5. Docker	43
3.6.6. Android/Kotlin.	44
3.7. Stworzone komponenty	47
4. REZULTATY	49
4.1. Wytwarzanie projektu.	49
4.2. Porównanie istniejących rozwiązań ze stworzoną aplikacją.	49
5. PODSUMOWANIE	52
WYKAZ LITERATURY	55
WYKAZ RYSUNKÓW.	55

WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW

Jeśli w tekście nie wykazano inaczej, stosowane symbole należy rozumieć jako:

- API (ang. *application programming interface*) - interfejs programowania aplikacji
- REST (ang. *representational state transfer*) - styl architektury oprogramowania
- JSON (ang. *JavaScript object notation*) - lekki format wymiany danych
- JWT (ang. *JSON web token*) - tokeny webowe w formacie JSON
- IoC (ang. *Inversion of Control*) - wzorzec projektowy, w którym zarządzanie działaniem aplikacji i tworzeniem obiektów jest przeniesione do kontenera lub framework'u

1. WSTĘP I CEL PRACY

W ostatnich czasach zdalne nauczanie i informatyzacja szkół zaczęły stanowić coraz ważniejszą część procesu edukacji. Internetowe dzienniki oraz platformy edukacyjne były w użyciu już od dłuższego czasu, jednak pandemia COVID-19 stworzyła nowe i niespotykane do tej pory potrzeby w tym zakresie. Z dnia na dzień placówki oświatowe zostały zmuszone do przejścia na całkowicie zdalne zajęcia, na co wiele z nich nie było przygotowanych. Na rynku nie było odpowiednich rozwiązań mogących kompleksowo i w jednym serwisie obsłużyć wszystkie potrzeby jakie ta sytuacja stworzyła. Sytuacja ta miała miejsce jakiś czas temu i wiele firm opracowało rozwiązania odpowiadające współczesnej edukacji.

Pomysły te są jednak bardzo ogólne, nie ma możliwości dostosowania ich do potrzeb konkretnych placówek, co zmusza je do korzystania z dodatkowych aplikacji zewnętrznych. W konsekwencji sprawia to niepotrzebne problemy zarówno uczniom jak i nauczycielom. Jednym z takich obszarów jest nauka muzyki. Wymagane są do niej konkretne funkcjonalności, które nie są dostępne w istniejących platformach.

Jedną z takich funkcjonalności jest możliwość tworzenia nagrań. Posiadanie takiej funkcji wbudowanej w system jest bardzo wygodne ze względu na brak konieczności instalowania oddzielnego oprogramowania oraz oszczędność czasu. Nagrania mogą być również automatycznie tagowane według ucznia i kursu lub tematu na jaki je wysyła, co ułatwi nauczycielowi ich przeglądanie, odsłuchiwanie i ocenianie.

1.1. Cel i zakres pracy

Celem pracy jest zbadanie obecnych technologii i wykorzystanie ich do zdalnej nauki muzyki oraz jako szeroko pojęta pomoc dydaktyczna, a następnie użycie wyników tych badań do stworzenia aplikacji o jak najlepszym dopasowaniu.

Zakres pracy obejmuje kooperację z potencjalnymi użytkownikami w celu zebrania i analizy wymagań oraz stworzenie systemu na ich podstawie. Na koniec osiągnięte efekty zostaną opisane oraz przetestowane.

1.2. Założenia projektowe

Głównym założeniem pracy jest zaprojektowanie i zaimplementowanie aplikacji internetowej pozwalającej na zdalne prowadzenie lekcji muzyki. Podstawową funkcjonalnością ma być tworzenie kursów przez nauczycieli i dodawanie do nich uczniów. Mają one umożliwiać przesyłanie materiałów i tworzenie nagrań bezpośrednio w aplikacji zarówno przez wykładowcę jak i przez uczestników. Serwis ma również obsługiwać czat pomiędzy użytkownikami kursów oraz wideo-rozmowy. Co więcej, oprócz strony internetowej zostanie stworzona aplikacja mobilna udostępniająca takie same funkcjonalności jak jej przeglądarkowy odpowiednik, ale dostosowana do warunków urządzeń przenośnych.

Wraz z aplikacją powstanie obsługujący ją serwer. Ma on pozwalać na tworzenie i zapisywanie kont wielu użytkowników oraz ich danych i materiałów. Ponadto jego zadaniem będzie koordynacja wideo-rozmów oraz czatu. Wszystkie te działania mają być autoryzowane i uwierzytelniane.

2. PRZEGŁĄD ISTNIEJĄCYCH ROZWIĄZAŃ

2.1. Uczenie zdalne

Sytuacja na świecie zmusiła społeczeństwo by wiele codziennych czynności zacząć wykonywać zdalnie. Edukacja była jedną z nich. Wiele placówek i edukatorów wprowadziło jedynie awaryjne nauczenie zdalne (Emergency Remote Teaching), czyli przeniesienie nauczania do środowiska wirtualnego bez zmian w metodach i technikach pedagogicznych. Sprawiło to, że wyniki takich zajęć były gorsze niż przy zajęciach tradycyjnych. W takich sytuacjach konieczne jest zaplanowanie nauczania zdalnego, czyli takiego, które od początku było tworzone z myślą o możliwościach ale też i o ograniczeniach platform wirtualnych.

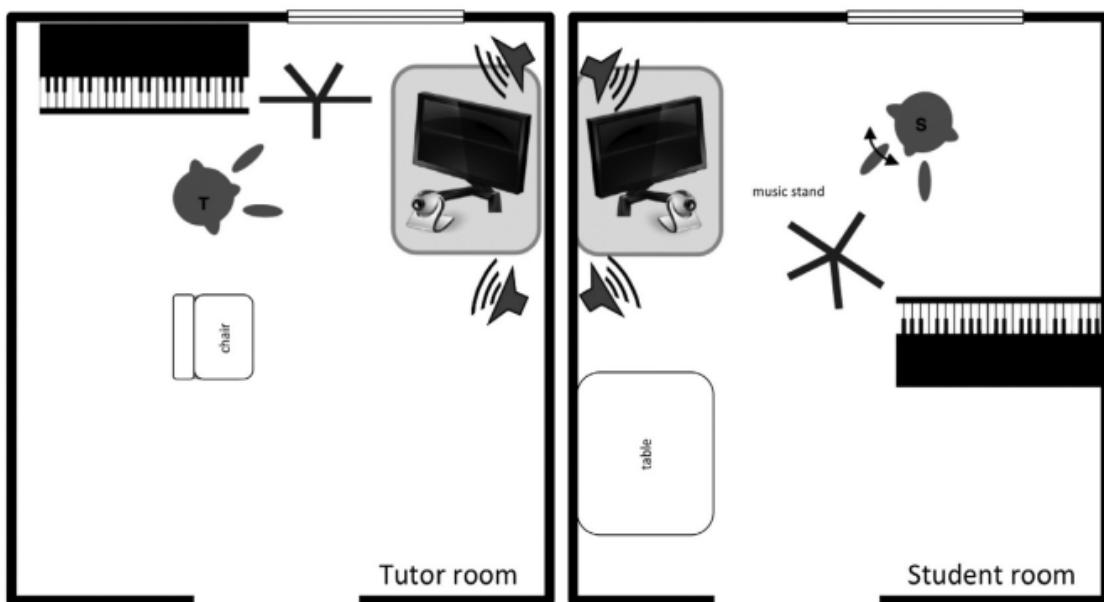
Badania wykazały również, że w takim środowisku nauczyciel przyjmuje poniekąd rolę konsultanta i moderatora dyskusji, przez co nauczanie, a szczególnie informacje zwrotne dla uczniów, muszą być skonstruowane w inny sposób. Innowacyjne metody pedagogiczne są potrzebne by zaangażować studentów i stymulować ich proaktywne zachowania, co jest trudne przy połączeniu wirtualnym. Szczególnie potrzebne są nowe sposoby na utrzymanie uwagi dzieci i ich zaangażowanie w działalności na ekranie. Konieczne jest również położenie większego nacisku na umiejętności techniczne nauczycieli, których braki okazały się sporym problemem. Wymienione niedociągnięcia były zdecydowanie mniej zauważalne na uniwersytetach, gdzie już wcześniej stosowano elementy nauczania zdalnego dla obniżenia kosztów [1].

Studenci inaczej postrzegają kursy zdalne i tradycyjne [2]. W nauczaniu zdalnym, jeszcze bardziej niż w stacjonarnym, ważna jest samoocena możliwości wykonania danego zadania [3]. Dla wielu studentów architektura kursów oraz to w jaki sposób będzie przeprowadzana praca i jak będą za nią oceniani ma duże znaczenie [4]. Zmiana trybu nauczania, pomimo że pozornie mogła się wydawać ułatwieniem dla studentów, została przez wielu przyjęta negatywnie. Jako największy z problemów wymieniane jest mniejsze zaangażowanie w postaci gorszego zainteresowania zajęciami i znajomymi z uczelni oraz poczucie wyobcowania ze społeczności akademickiej. Kolejnym problemem w oczach ankietowanych studentów była zwiększoną ilość pracy zlecaną im przez wykładowców połączona z gorszym jej zrozumieniem i słabszą komunikacją. Innym ważnym aspektem było przygotowanie technologiczne. Studenci bez adekwatnego połączenia z internetem oceniali zajęcia dużo gorzej niż tacy, którzy nie mieli z tym problemów. Szczególnie zwracali na to uwagę studenci muzyki, dla których nieregularne opóźnienia transmisji stanowiły duże utrudnienie [5]. Nauczanie zdalne oprócz wielu wad miało też swoje mocne strony. Najczęszszą wymienianą zaletą była możliwość nagrywania wykładów, umożliwiająca późniejsze odwołanie się bezpośrednio do słów wykładowcy lub też całkowite uczestniczenie w wykładzie asynchronicznie oraz mniejszy stres związany z zajęciami [2].

Również nauczyciele odmiennie postrzegają kursy tradycyjne i online. Duża część (57.8% ankietowanych) uważała, że zdalne nauczanie muzyki nawet w najlepszym przypadku nie jest w stanie przynieść tak dużych rezultatów jak zajęcia stacjonarne, natomiast 84% obawiało się że ich podopieczni nie otrzymywali zajęć o adekwatnej jakości. Wielu pedagogów stwierdziło też, że po-

ziom zaangażowania uczniów w pracę był niższy i że mieli oni większy problem z nadążaniem za instrukcjami prowadzącego. Do przeprowadzania zajęć korzystano z różnych systemów, najczęściej Zoom lub Google Meet, jak również z narzędzi pomocniczych, stworzonych na potrzeby edukacji muzycznej lub też przesyłania materiałów. Około połowa ankietowanych nie używała wcześniej żadnych narzędzi do zdalnego nauczania. Dużym problemem był też brak dostępu uczniów do instrumentów potrzebnych im do ćwiczeń, więc w wielu przypadkach nauka musiała być na jakiś czas przerwana [6].

Przy nauczaniu muzyki niezwykle ważne jest również umiejscowienie użytkowników w przestrzeni [2.1]. Podczas tradycyjnych zajęć uwaga nauczyciela jak i ucznia skupia się na statywie, przy czym w trakcie gry nadal pozostają w swoim polu widzenia, dzięki czemu są w stanie reagować na niewerbalne znaki znacznie ułatwiające komunikację. W środowisku wirtualnym możliwości te są bardzo ograniczone, co powoduje konieczność częstszego przerywania lekcji zarówno przez nauczyciela jak i przez ucznia. Jednym z proponowanych sposobów na rozwiązanie problemu jest odpowiednie ustawienie kamer tak, by uczestnicy widzieli siebie nawzajem jednocześnie. Umożliwiało to interakcje bardziej zbliżone do tych zachodzących w spotkaniach tradycyjnych [7].



Rysunek 2.1: Proponowany układ pomieszczeń dla zdalnych lekcji muzyki

Stwierdzono również, że sytuacja kryzysowa stanowiła dobrą okazję promowania niezależności i odpowiedzialności ze strony uczniów. Jednak jednym z głównych ograniczeń była utrata interakcji międzyludzkich między nauczycielami i uczniami, a także między samymi uczniami. Według ekspertów, choć korzystanie z udogodnień technologicznych jest pomocne dla uczniów na całym świecie, nie zastąpi to właściwej interakcji nauczyciel-uczeń. Aby złagodzić problemy związane z integracją eksperci sugerowali stosowanie podejścia mieszanego gdy tylko jest to możliwe. Blended learning jest definiowany jako "przemyślane połączenie nauki bezpośredniej i online" [8]. Umożliwia to zintensyfikowanie postrzegania czynników "ludzkich" i wzmacnia poczucie przynależności do społeczności. Z pewnością blended learning ułatwia interakcję, poprawia współpracę

i relacje społeczne między uczniami oraz między uczniami i nauczycielami [9]. W przyszłości, gdy normalne działania edukacyjne będą mogły zostać wznowione, należy ustanowić równoważące między nauką w szkole a nauką online, która stanowi narzędzie mogące uzupełniać lekcje bezpośrednie.

2.2. Aplikacje

2.2.1. Microsoft Teams

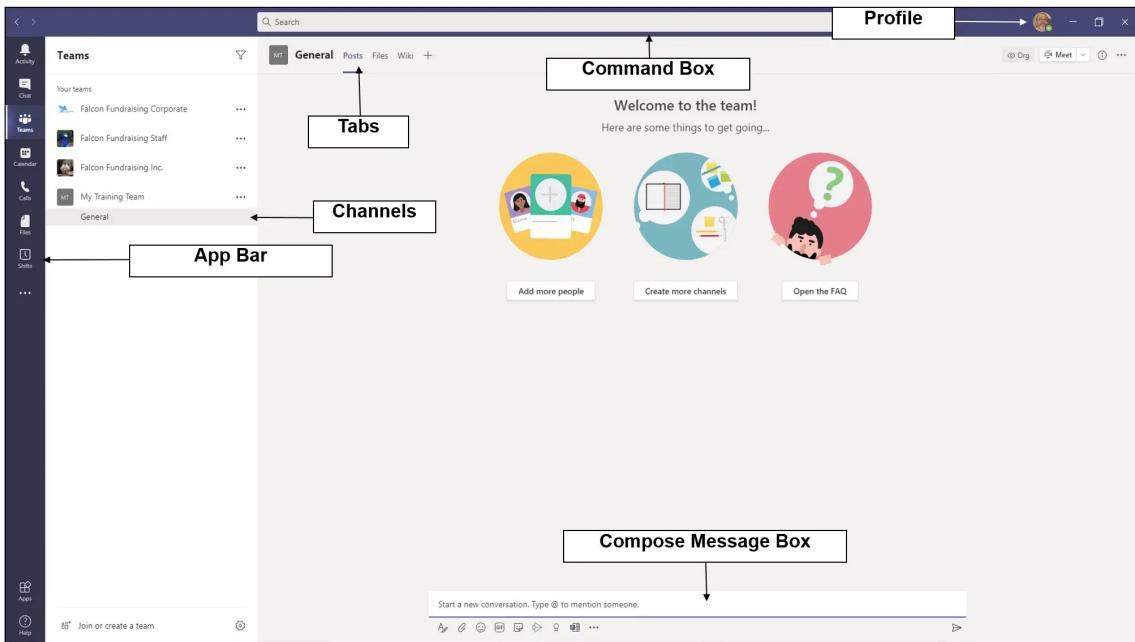
Microsoft Teams stał się wszechstronnym i kompleksowym narzędziem do nauczania, umożliwiającym nauczycielom płynną komunikację, współpracę i dostarczanie treści w dziedzinie zdalnego nauczania i edukacji online. Dzięki rozbudowanym funkcjom i integracji z innymi aplikacjami Microsoft, Teams zapewnia spójną i przyjazną dla użytkownika platformę zarówno dla nauczycieli jak i dla uczniów.

Jedną z kluczowych zalet Microsoft Teams jest możliwość ułatwienia komunikacji i współpracy w czasie rzeczywistym. Nauczyciele mogą tworzyć dedykowane zespoły dla swoich klas, umożliwiając uczniom dołączanie i uczestniczenie w dyskusjach, zadawanie pytań i udostępnianie zasobów. Funkcja czatu pozwala na natychmiastowe przesyłanie wiadomości i udostępnianie plików, promując wydajną i skuteczną komunikację między nauczycielami i uczniami. Ponadto integracja funkcji konferencji audio i video pozwala na wirtualne spotkania, umożliwiając interakcję twarzą w twarz, wykłady na żywo i dyskusje grupowe, które odwzorowują tradycyjne środowisko klasowe [2.2].

Aplikacja oferuje również szereg narzędzi wspierających dostarczanie i organizację treści. Nauczyciele mogą tworzyć i udostępniać zadania, quizy i materiały edukacyjne na platformie, zapewniając uczniom łatwy dostęp do zasobów edukacyjnych. Wbudowana funkcja udostępniania plików pozwala nauczycielom na płynną dystrybucję materiałów szkoleniowych, prezentacji i literatury. Co więcej, integracja z innymi aplikacjami Microsoft takimi jak OneNote i SharePoint umożliwia nauczycielom centralizację treści i usprawnienie przepływu pracy, poprawiając organizację i współpracę w wirtualnej klasie.

Innym godnym uwagi aspektem aplikacji jest jej wszechstronność w dostosowywaniu się do różnych stylów nauczania i podejść instruktażowych. Platforma obsługuje zarówno synchroniczne jak i asynchroniczne doświadczenia edukacyjne umożliwiając nauczycielom dostarczanie lekcji na żywo lub wcześniej nagranych wykładów, do których uczniowie mogą uzyskać dostęp we własnym tempie. Możliwość planowania i prowadzenia wirtualnych spotkań z podopiecznymi lub organizowania webinarów dla większych grup zapewnia elastyczność w dostosowywaniu się do różnych potrzeb dydaktycznych.

Ponadto Microsoft Teams zapewnia szereg funkcji ułatwień dostępu, aby zapewnić integrację wszystkim uczniom. Platforma obsługuje napisy i czytniki ekranu, dzięki czemu treści są bardziej dostępne dla osób niedosłyszających lub niedowidzących. Nauczyciele mogą również korzystać z funkcji czytnika immersyjnego, aby poprawić czytanie ze zrozumieniem i wspierać różne style uczenia się. Te funkcje dostępności pomagają stworzyć jednakowe doświadczenie edukacyjne dla wszystkich uczniów niezależnie od ich indywidualnych potrzeb.



Rysunek 2.2: Interfejs platformy Teams

Architektura zbudowana jest na platformie Azure i korzysta z jej możliwości skalowania, funkcji bezpieczeństwa i integracji z innymi serwisami firmy. Wiele z funkcjonalności Teams jak wideo-rozmowy, czat czy powiadomienia zostało przejęte po programie Skype. Aplikacje mobilne zostały zbudowane w języku natywnym dla danej platformy (Swift dla iOS i Java dla Androida). Wersje okienkowa i przeglądarkowa stworzone są jako jedna aplikacja oparta o framework Angular. Aplikacja desktopowa uruchamiana jest w środowisku electron, a następnie dodany jest natywny dla systemu kod dodający SSO oraz umożliwiający dzwonienie i przesyłanie plików [10]. Aplikacja zbudowana jest na platformie chmurowej Office 365, która zapewnia wysoki poziom bezpieczeństwa. Posiada autentykację dwuskładnikową zarówno na poziomie organizacji jak i poszczególnych zespołów, SSO przy użyciu Active Directory oraz enkrypcję danych przy przesyłaniu oraz po zapisaniu.

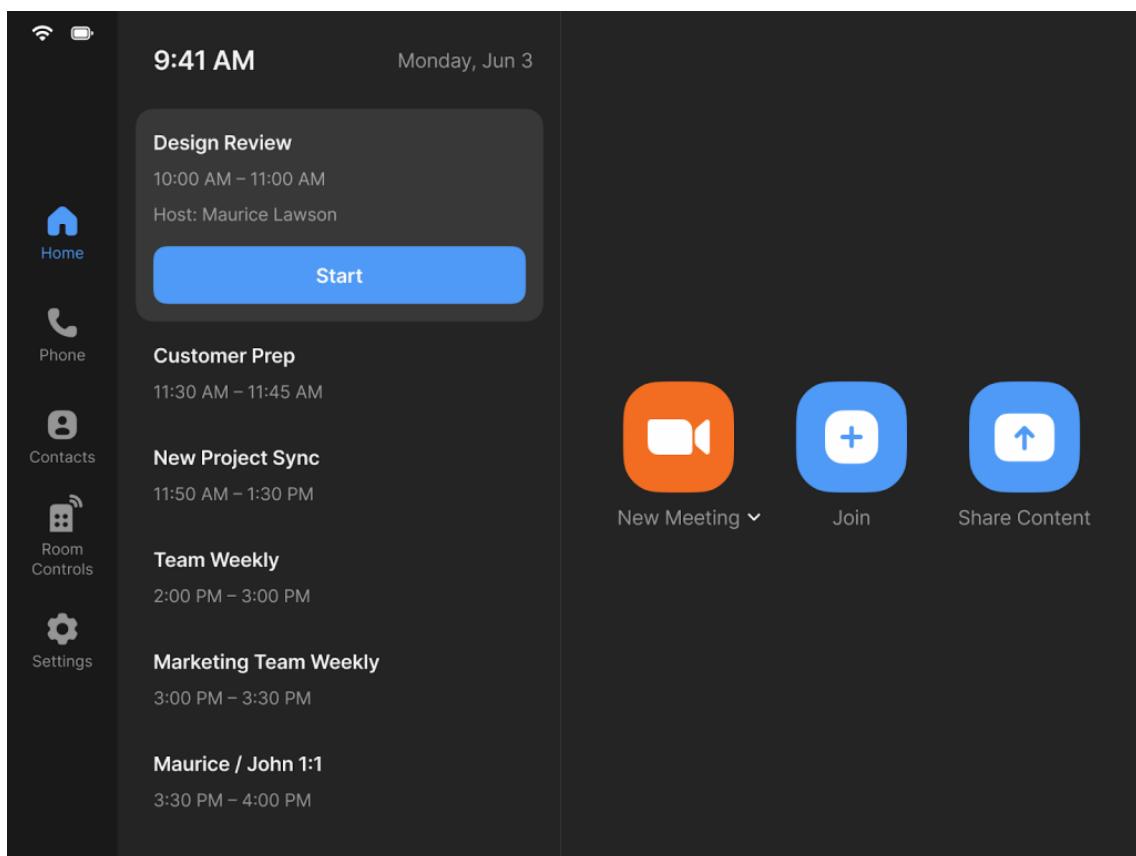
2.2.2. Zoom

Zoom to szeroko stosowana i bardzo popularna platforma do wideokonferencji, która stała się dla wielu niezbędnym narzędziem do zdalnej komunikacji i wirtualnych spotkań. Oferując płynny i przyjazny dla użytkownika interfejs, Zoom umożliwia osobom i zespołom łączenie się i współpracę z dowolnego miejsca na świecie.

Jedną z głównych zalet aplikacji jako narzędzia do nauczania jest jego zdolność do ułatwiania komunikacji i interakcji w czasie rzeczywistym [2.3]. Dzięki wysokiej jakości funkcji wideo i audio nauczyciele mogą prowadzić wykłady na żywo, które ściśle naśladują tradycyjne doświadczenie w klasie. Umożliwia to uczniom wizualne i głosowe angażowanie się z instruktorem, poprawiając ich ogólne doświadczenie edukacyjne. Dodatkowo, funkcja czatu Zoom pozwala na

natychmiastowe przesyłanie wiadomości i udostępnianie plików, wspierając komunikację w czasie rzeczywistym i współpracę między nauczycielami i uczniami. Uczniowie mogą zadawać pytania, prosić o wyjaśnienia i przekazywać informacje zwrotne, tworząc interaktywne i dynamiczne środowisko wirtualnej klasy.

Funkcja udostępniania ekranu Zoom jest kolejnym cennym atutem dla nauczycieli. Pozwala ona instruktorom na udostępnianie swoich ekranów ułatwiając prezentowanie pokazów slajdów, filmów i innych zasobów edukacyjnych. Niezależnie od tego, czy chodzi o prezentowanie pomocy wizualnych, czy demonstrowanie złożonych problemów, udostępnianie ekranu umożliwia nauczycielom dostarczanie dynamicznych i interaktywnych lekcji, które przyciągają uwagę uczniów i ułatwiają zrozumienie. Ponadto Zoom oferuje opcję dodawania adnotacji do udostępnianych ekranów umożliwiając nauczycielom i uczniom współpracę i interakcję bezpośrednio na prezentowanych treściach.



Rysunek 2.3: Interfejs platformy Zoom

Funkcja dodatkowych pokoi (breakout rooms) Zoom jest szczególnie korzystna dla pracy grupowej. Nauczyciele mogą podzielić swoje wirtualne klasy na mniejsze grupy umożliwiając uczniom angażowanie się w dyskusje, ćwiczenia i projekty. Promuje to interakcję między rówieśnikami, pracę zespołową i poczucie wspólnoty, zwiększając zaangażowanie uczniów i ich aktywny udział. Możliwość przemieszczania się między pokojami i monitorowania postępów grupy pozwala nauczycielom na udzielanie wskazówek i wsparcia w razie potrzeby.

Zoom oferuje również szereg dodatkowych funkcji, które zwiększą komfort nauczania. Funkcja nagrywania pozwala nauczycielom rejestrować sesje na żywo, udostępniając je uczniom do oglądu w dogodnym dla nich czasie. Służy to jako cenne źródło informacji dla uczniów, aby mogli powrócić do kluczowych pojęć, wzmacnić ich zrozumienie lub nadrobić zaległości w opuszczonych zajęciach. Zoom zapewnia również funkcję tablicy umożliwiającą nauczycielom ilustrowanie koncepcji i współpracę z uczniami w czasie rzeczywistym. Co więcej, integracja Zoom z systemami zarządzania nauczaniem i innymi narzędziami edukacyjnymi usprawnia zarządzanie kursami i udostępnianie treści, ułatwiając nauczycielom organizowanie materiałów klasowych, zadań i ocen.

Aplikacja ma też swoje wady. Zoom był często krytykowany za luki w zabezpieczeniach. Firma twierdziła, że używała enkrypcji end-to-end, natomiast w rzeczywistości używa jej tylko pomiędzy swoimi serwerami a nie klientami. W aplikacji były też luki bezpieczeństwa pozwalające na oglądanie udostępnianych ekranów i czytanie czatu bez wiedzy prowadzącego spotkanie, jak również przejmowanie kont [11]. Te oraz wiele innych problemów sprawiły, że platforma była badana przez organizacje rządowe, co więcej została złożony przeciwko niej pozew zbiorowy [12].

2.2.3. *Discord*

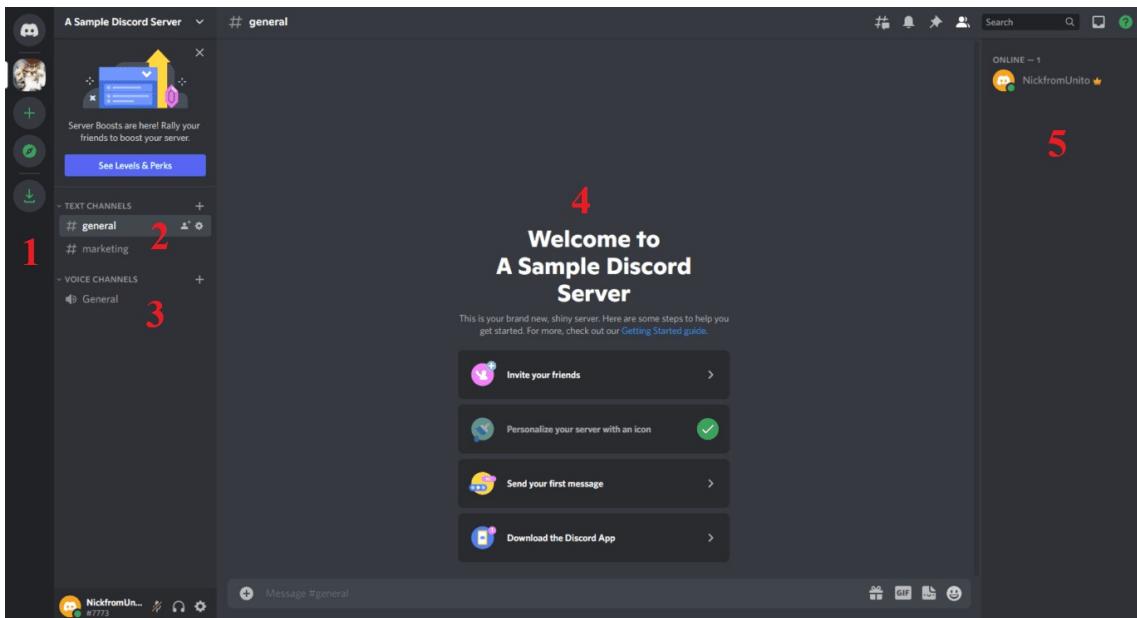
Discord to platforma społecznościowa VoIP i komunikator internetowy. Działa na systemach Windows, macOS, Android, iOS, iPadOS, Linux oraz w przeglądarkach internetowych. Od 2021 r. usługa ma ponad 350 milionów zarejestrowanych użytkowników i ponad 150 milionów aktywnych użytkowników miesięcznie [13].

Discord służy do tworzenia i zarządzania prywatnymi i publicznymi społecznościami. Daje użytkownikom dostęp do narzędzi takich jak połączenia głosowe i wideo, trwałe pokoje rozmów i integracje z innymi usługami dla graczy, a także ogólną możliwość wysyłania bezpośrednich wiadomości i tworzenia zamkniętych grup [2.4]. Pomimo iż serwis początkowo skierowany był głównie do graczy, w ostatnich latach nowe aktualizacje uczyniły z niego komunikator używany przez szersze grono użytkowników.

Poszczególne społeczności są zorganizowane w formie serwerów. Użytkownicy mogą zakładać je za darmo, zarządzać ich widocznością oraz tworzyć kanały głosowe, tekstowe i kategorie do ich sortowania. W każdym może być maksymalnie 800 000 użytkowników. Od 2017 roku istnieją również zweryfikowane serwery tworzone przez artystów, drużyny sportowe i twórców gier jako oficjalne społeczności fanów.

Głównym medium komunikacyjnym są kanały. Głosowe umożliwiają rozmowę oraz udostępnianie wideo, natomiast tekstowe wspierają przesyłanie plików. Wiadomości mogą być formatowane przy pomocy składni podobnej do języka Markdown. Jedną z funkcjonalności wprowadzonych stosunkowo niedawno mających na celu przyciągnięcie do platformy nowego grona odbiorców są sceny. Są to kanały umożliwiające moderowaną prezentację i dyskusję z kilkoma prezenterami i resztą słuchaczy, którzy mogą zabrać głos po zatwierdzeniu przez moderatorów [14].

Discord udostępnia również API dzięki któremu deweloperzy mogą pokazywać w profilu użytkownika co aktualnie robi w ich aplikacji lub zintegrować dołączanie do meczy i podobnych



Rysunek 2.4: Interfejs platformy Discord

1. Lista serwerów
2. Lista kanałów tekstowych
3. Lista kanałów głosowych
4. Widok wybranego kanału
5. Lista członków kanału

aktywności bezpośrednio przez komunikator. Dostępna jest też szeroka gama wtyczek i botów, od moderujących po odtwarzające muzykę. Istnieje również możliwość pisania własnych wtyczek, do których zostały stworzone dedykowane narzędzia takie jak discord.js [15].

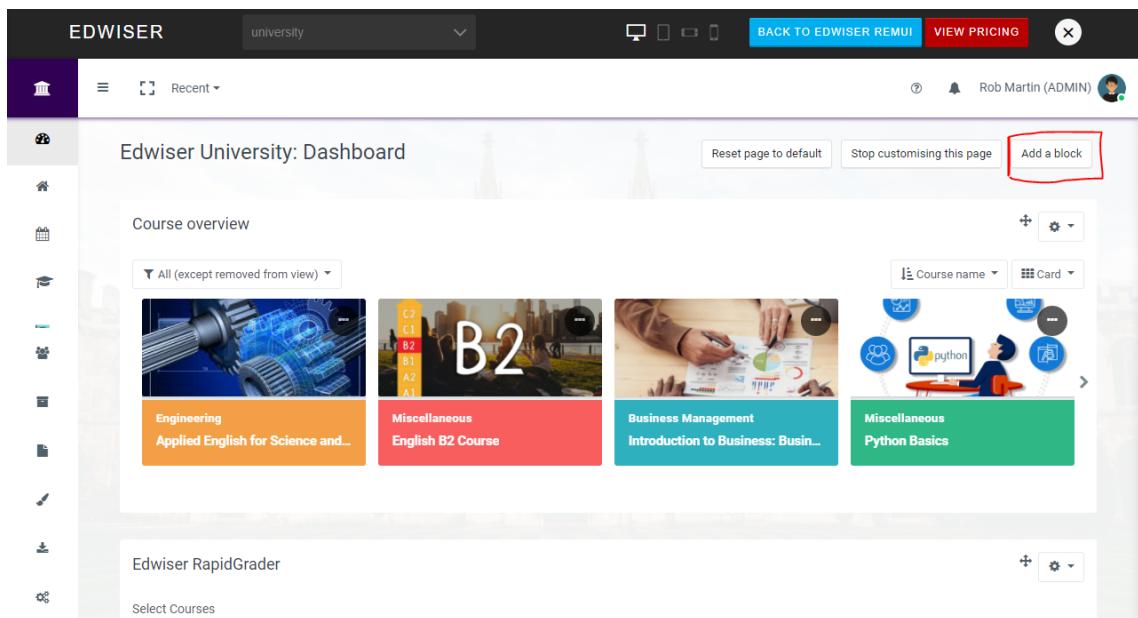
Discord oparty jest na bazie danych o ewentualnej spójności (Eventually consistent), a jego infrastruktura jest oparta na platformie GCP dla zapewnienia jak najmniejszych opóźnień dla klientów. Sama aplikacja napisana jest przy użyciu framework'a React, dla platform mobilnych React Native. Aplikacje desktopowe uruchamiane są w środowisku Electron. Backend został stworzony głównie w językach Elixir i Python, ale również Rust, Go i C++ [16].

2.2.4. Moodle

Moodle (Modular Object-Oriented Dynamic Language Environment) to system zarządzania treścią szkoleniową (LCMS) i środowisko wirtualnego nauczania (VLE). Stworzony początkowo jako otwarto-źródłowy projekt do wspomagania nauczania stacjonarnego. Ma rozbudowaną dokumentację i rozwija się w celu wspierania standardów IMS/SCORM [17]. Opiera się na konstruktywicznym podejściu do edukacji podkreślając, że uczniowie również mogą wnieść swój wkład w proces nauczania. Aktualnie zarejestrowanych jest 160 000 stron w 242 krajach [18].

Moodle oferuje przyjazny dla użytkownika interfejs, który ułatwia tworzenie i organizację kursów online. Nauczyciele mogą łatwo opracowywać materiały szkoleniowe, w tym treści multimedialne, zadania, quizy i fora dyskusyjne. Platforma obsługuje różne podejścia instruktażowe umożliwiając spersonalizowane i angażujące doświadczenie edukacyjne.

Jedną z kluczowych zalet Moodle jest jego wszechstronność i opcje dostosowywania.



Rysunek 2.5: Interfejs platformy Moodle

Platforma zapewnia szereg elastycznych formatów kursów, takich jak tygodniowe, tematyczne lub społecznościowe, umożliwiając nauczycielom dostosowanie struktury kursów do ich konkretnych potrzeb. Moodle obsługuje również integrację zewnętrznych narzędzi i zasobów umożliwiając płynną integrację z różnymi technologiami nauczania i systemami zarządzania treścią [2.5].

Wszechstronny zestaw narzędzi do oceniania i wystawiania ocen na platformie Moodle umożliwia nauczycielom skutecną ocenę wyników uczniów. Platforma oferuje różne rodzaje ocen, w tym quizy, zadania i interaktywne aktywności. Nauczyciele mogą ustalać kryteria oceniania, przekazywać informacje zwrotne i generować szczegółowe raporty na temat postępów uczniów, ułatwiając spersonalizowaną naukę i podejmowanie decyzji w oparciu o dane.

Współpraca i komunikacja są niezbędne w nauce online, a Moodle zapewnia kompleksowy zestaw funkcji ułatwiających interakcję między uczniami i nauczycielami. Fora dyskusyjne, systemy wiadomości i funkcje czatu w czasie rzeczywistym pozwalają na asynchroniczną i synchroniczną komunikację, wspierając współpracę, uczenie się od rówieśników i interakcję między instruktorem a studentem.

Co więcej, Moodle oferuje szereg funkcji administracyjnych, które upraszczają zarządzanie kontami użytkowników, rejestrację i administrowanie kursami. Nauczyciele mogą tworzyć grupy użytkowników, zarządzać uprawnieniami dostępu i śledzić aktywność użytkowników w systemie. Platforma obsługuje również kontrolę dostępu opartą na rolach, zapewniając, że tylko upoważnione osoby mają dostęp do określonych funkcji administracyjnych.

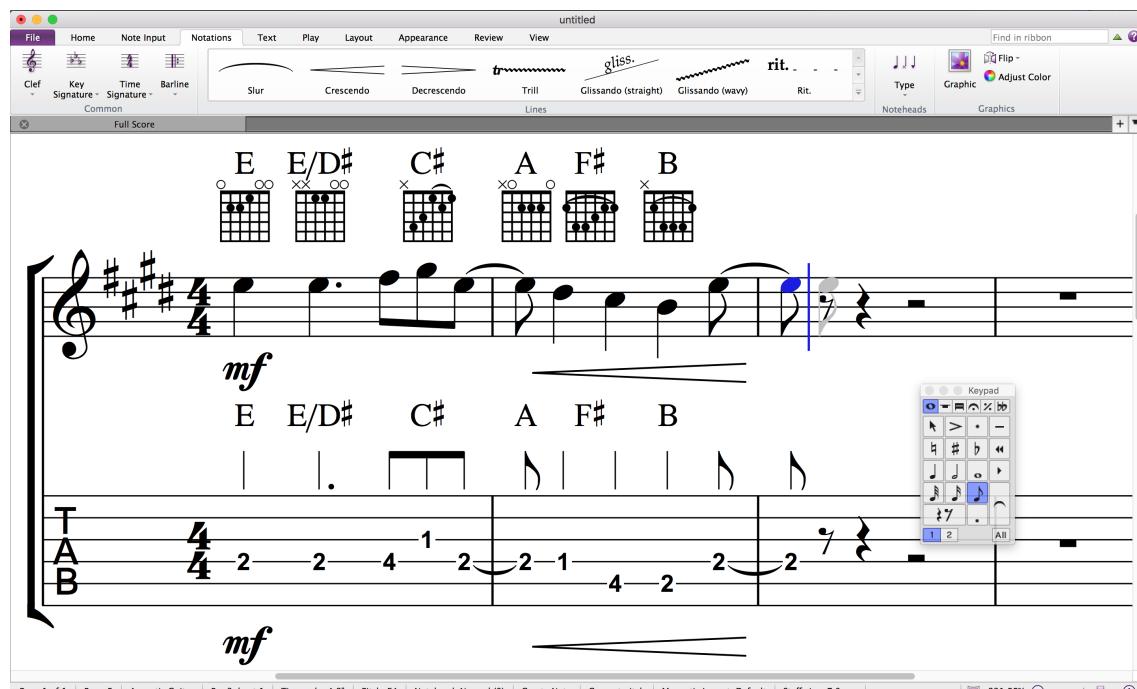
Kolejną istotną zaletą Moodle jest jego aktywna i wspierająca społeczność składającą się z nauczycieli, administratorów, programistów i entuzjastów, którzy przyczyniają się do ciągłego rozwoju i ulepszania platformy. Społeczność zapewnia obszerną dokumentację, fora i zasoby, ułatwiając użytkownikom szukanie pomocy, dzielenie się najlepszymi praktykami i współpracę

nad rozwojem Moodle.

Projekt napisany jest w języku PHP i wspiera kilka popularnych baz danych. Jest dostępny w formie gotowych paczek na wiele systemów operacyjnych. Istnieje również sieć hostingów Moodle'a z których zyski wspierają dalszy rozwój aplikacji. Moodle nie wspiera niektórych funkcjonalności bardzo przydatnych przy zdalnym nauczaniu takich jak wbudowane video-rozmowy, posiada jednak dużą bazę dostępnych wtyczek, których tworzenie jest relatywnie proste.

Platforma szczerzy się również wysokim poziomem bezpieczeństwa. Proces wytwarzania aplikacji, Secure by Design, stawia je na pierwszym miejscu w każdej nowej zmianie. Nie są zbiereane żadne dane użytkowników ani materiały edukacyjne. Wszystkie pluginy muszą implementować Privacy API by zapewnić zgodność z RODO. Moodle współpracuje również z platformą Bugcrowd, co pozwala ludziom z całego świata na ciągłe testowanie bezpieczeństwa platformy. Pozytywny wpływ ma też fakt, że system jest otwarty-źródłowy i ma olbrzymią liczbę użytkowników, którzy często zgłaszają zauważone błędy.

2.2.5. Sibelius



Rysunek 2.6: Interfejs aplikacji Sibelius

Sibelius, nazwany na cześć fińskiego kompozytora Jeana Sibeliusa, to największa na świecie aplikacja służąca do tworzenia partitur. Stworzony przez Avid Technologies program zapewnia użytkownikom rozbudowany, ale intuicyjny interfejs do precyzyjnego przedstawiania elementów muzycznych, od skomplikowanych struktur tonalnych po znuansowane artykulacje i dynamiczne wariacje. Centralnym elementem możliwości aplikacji jest zaawansowany silnik notacji, zaprojektowany w celu ułatwienia skrupulatnej transkrypcji pomysłów kompozytorskich. Ten

wszechstronny zestaw funkcji dostosowuje się do złożoności różnych idiomów muzycznych, oferując obszerną bibliotekę symboli notacji, ornamentów i artykulacji. Co więcej, silnik odtwarzania audio aplikacji, zintegrowany płynnie z funkcjami notacji, umożliwia odsłuchiwanie partytur w czasie rzeczywistym, wypełniając w ten sposób lukę między intencją kompozycji a jej słyszalną realizacją [2.6].

Integracja aplikacji Sibelius z bibliotekami instrumentów wirtualnych i bankami dźwięków opartymi na samplach jeszcze bardziej poprawia wrażenia kompozytorskie, wierne odzwierciedlając złożone orkiestracje. Funkcja ta uzupełnia proces komponowania zapewniając dokładne przybliżenie ostatecznego wyniku słuchowego, wspierając iteracyjny i wydajny przepływ pracy. Co ważne, oprogramowanie obsługuje różne standardowe formaty plików zapewniając interoperacyjność i kompatybilność z istniejącymi ekosystemami oprogramowania muzycznego. Sibelius wspiera też zapisywanie projektów w chmurze, co pozwala na zdalne udostępnianie i edycję kompozycji.

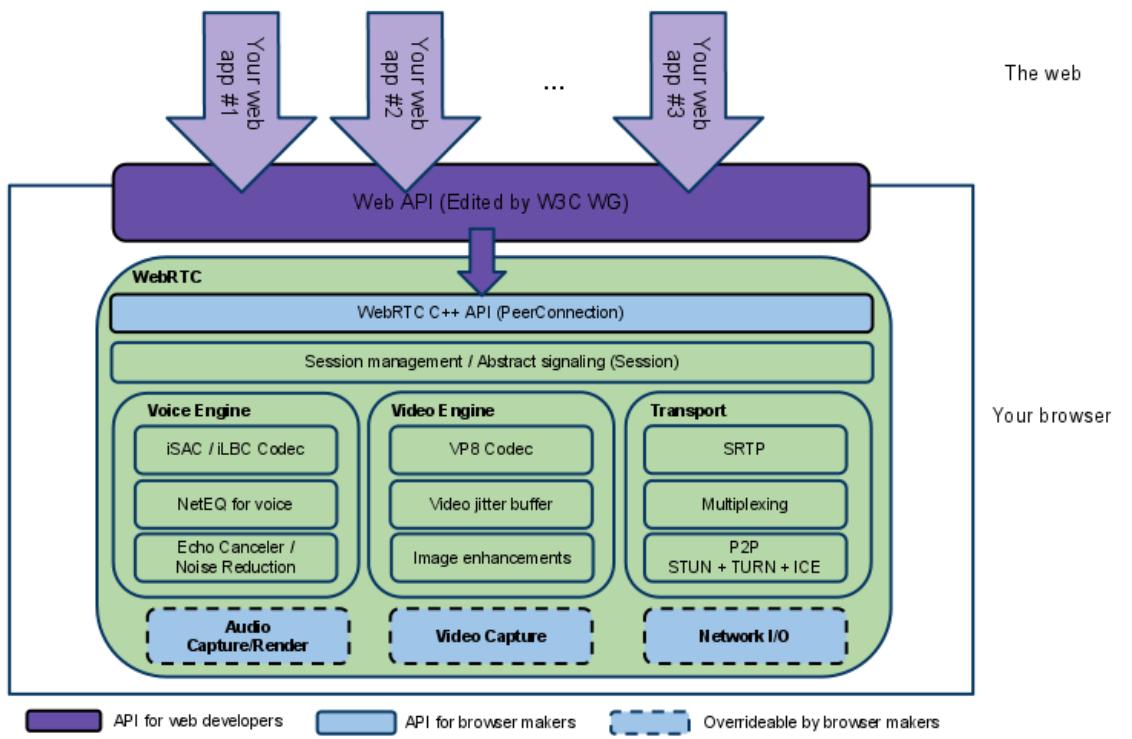
2.3. Technologie

2.3.1. WebRTC

WebRTC to technologia pozwalająca aplikacjom internetowym na przesyłanie strumieni audio i/lub wideo, jak również na wymianę dowolnych danych pomiędzy przeglądarkami bez konieczności korzystania z serwera pośredniczącego. Zestaw standardów, które mieści w sobie WebRTC, umożliwia przesyłanie informacji i przeprowadzanie telekonferencji peer-to-peer bez konieczności instalowania jakichkolwiek plug-inów lub dodatkowego oprogramowania przez użytkownika [2.7] [19]. Opublikowana po raz pierwszy w 2011 roku przez Google. WebRTC eliminuje potrzebę instalowania zewnętrznych wtyczek lub oprogramowania, dzięki czemu jest dostępny dla użytkowników na różnych platformach i urządzeniach.

Jedną z głównych zalet WebRTC jest jego zdolność do ułatwiania wysokiej jakości komunikacji audio i wideo. Wykorzystuje zaawansowane kodeki i adaptacyjne techniki strumieniania, aby zapewnić płynną i wydajną transmisję danych audio i wideo, nawet w trudnych warunkach sieciowych. Umożliwia to użytkownikom prowadzenie wyraźnych rozmów i wideokonferencji w czasie rzeczywistym bezpośrednio z poziomu przeglądarki internetowej, bez konieczności korzystania z aplikacji lub wtyczek innych firm. Platforma oferuje również wbudowane środki bezpieczeństwa w celu ochrony prywatności użytkowników i integralności danych. Wykorzystuje protokoły szyfrowania do zabezpieczania strumieni audio, wideo i danych, zapewniając poufność wrażliwych informacji podczas transmisji. WebRTC zawiera również mechanizmy zapobiegające podsłuchiwaniu, manipulowaniu i fałszowaniu tożsamości, zapewniając bezpieczne środowisko komunikacyjne dla użytkowników. Co więcej, jest wysoce wszechstronny i rozszerzalny, umożliwiając programistom integrację z różnymi aplikacjami i platformami. Obsługuje interoperacyjność między różnymi przeglądarkami internetowymi, umożliwiając użytkownikom płynną komunikację niezależnie od używanej przeglądarki. Ponadto WebRTC można łączyć z innymi technologiami internetowymi, takimi jak HTML5, JavaScript i CSS, aby tworzyć bogate i interaktywne doświadczenia użytkownika.

WebRTC znalazł zastosowanie w wielu branżach i przypadkach użycia. Jest szeroko stosowany do wideokonferencji, zdalnej współpracy i obsługi klienta online. Został również wykorzystany w rozwoju aplikacji do gier w czasie rzeczywistym, platform transmisji na żywo i usług telezdrowia. Elastyczność i wszechstronność WebRTC sprawiają, że jest to potężne narzędzie umożliwiające komunikację i współpracę w czasie rzeczywistym w aplikacjach internetowych.



Rysunek 2.7: Schemat architektury WebRTC

Główne komponenty API to:

- MediaStream - daje przeglądarkę dostęp do kamery i mikrofonu,
- RTCPeerConnection - tworzy rozmowy audio/video,
- RTCDataChannel - pozwala przeglądarkę na wysyłanie danych przez połączenie P2P.

WebRTC nie jest pojedynczym API, ale zbiorem API i protokołów definiowanych przez organizacje takie jak W3C i IETF. Wsparcie dla nich jest rozwijane niezależnie dla różnych przeglądarek i systemów operacyjnych [20].

MediaStream

API MediaStream udostępnia kontrolę nad strumieniami multimedialnymi. Zapewnia znormalizowany i ujednolicony interfejs dostępu do multimediiów z różnych źródeł, w tym kamer, mikrofonów, ekranów, a nawet wcześniej nagranych plików. MediaStreams mogą być przechwytywane

i przetwarzane w czasie rzeczywistym, umożliwiając aplikacjom przesyłanie treści audio i wideo bezpośrednio między użytkownikami w sesji WebRTC. MediaStream umożliwia programistom łatwy dostęp do strumieni multimedialnych i manipulowanie nimi, stosowanie filtrów, przetwarzanie audio i wideo oraz integrowanie ich z interfejsem użytkownika. Zapewnia metody kontrolowania przechwytywania multimedialów, takie jak uruchamianie, zatrzymywanie, pauzowanie i wznowianie strumieni, a także obsługę zdarzeń związanych ze zmianami strumienia multimedialów. Każdy obiekt MediaStream posiada wejście oraz wyjście. Wejściem może być fizyczne urządzenie takie jak kamera czy mikrofon, ale również źródło wirtualne. Wyjście może być przesyłane do wielu odbiorców takich jak element wideo w przeglądarce czy komponent PeerConnection. Zarówno wejścia jak i wyjścia są podzielone na ścieżki w postaci MediaStreamTrack. Każda taka ścieżka posiada sprecyzowany typ jak audio lub wideo. Są one reprezentacją typu danych pozyskiwanych ze źródła, przykładowo po podłączeniu mikrofonu dodany zostanie obiekt MediaStreamTrack (audio) [19].

RTCPeerConnection

API RTCPeerConnection umożliwia komunikację peer-to-peer audio, wideo i danych między przeglądarkami. Ustanawia i zarządza bezpośrednim połączeniem między dwoma urządzeniami równorzędnymi, umożliwiając im przesyłanie strumieni multimedialnych i wymianę dowolnych danych w czasie rzeczywistym. RTCPeerConnection obsługuje negocjowanie możliwości multimedialnych, konfiguracji sieci i parametrów bezpieczeństwa między urządzeniami równorzędnymi przy użyciu protokołów takich jak SDP i ICE. Wykorzystuje ICE do przechodzenia przez NAT i ustanawiania bezpośrednich połączeń peer-to-peer, gdy tylko jest to możliwe. RTCPeerConnection zapewnia elastyczną i konfigurowalną platformę dla programistów do integracji komunikacji w czasie rzeczywistym z aplikacjami internetowymi. Obsługuje funkcje takie jak kodeki audio/wideo, zarządzanie przepustowością, kontrola mediów i obsługa zdarzeń związanych ze stanem połączenia i strumieniem mediów.

RTCDataChannel

API RTCDataChannel umożliwia transmisję danych peer-to-peer między przeglądarkami w czasie rzeczywistym. Zapewnia niezawodny, uporządkowany i dwukierunkowy kanał wymiany dowolnych danych, takich jak tekst, pliki binarne lub strumienie multimedialne. RTCDataChannel działa w oparciu o podstawowe protokoły transportowe ustanowione przez WebRTC, umożliwiając bezpośrednią komunikację między instancjami przeglądarki bez potrzeby stosowania serwerów pośredniczących. Obsługuje konfigurowalne opcje niezawodności, umożliwiając programistom wybór między niezawodnym dostarczaniem (gdzie dane są gwarantowane w kolejności) lub niewiarygodnym dostarczaniem (gdzie szybkość dostarczania ma pierwszeństwo przed kolejnością).

Serwer sygnalizujący

Serwer sygnalizacyjny WebRTC odgrywa kluczową rolę w ułatwianiu komunikacji między równorzędnymi uczestnikami sesji. Podczas gdy WebRTC obsługuje rzeczywistą transmisję multimediiów, serwer sygnalizacyjny obsługuje wymianę komunikatów kontrolnych sesji, umożliwiając peerom wykrywanie i nawiązywanie bezpośrednich połączeń. Serwer sygnalizacyjny pomaga peerom w wymianie informacji, takich jak opisy sesji (SDP), kandydaci ICE i inne metadane wymagane do skonfigurowania połączenia. Działa jako mediator, umożliwiając negocjowanie możliwości multimedialnych, adresów sieciowych i ustanowienie początkowego uzgadniania połączenia. Serwer sygnalizacyjny może wykorzystywać różne protokoły, takie jak WebSocket lub HTTP, do przesyłania tych komunikatów między urządzeniami równorzędnymi. Po ustanowieniu początkowego połączenia rzeczywiste strumienie multimediiów mogą być przesyłane bezpośrednio między urządzeniami równorzędnymi bez potrzeby dalszego angażowania serwera sygnalizacyjnego. Należy zauważyć, że sam serwer sygnalizacyjny nie obsługuje transmisji multimediiów; jego podstawową funkcją jest umożliwienie nawiązywania i koordynowania połączeń między urządzeniami równorzędnymi.

Protokoły i framework'i używane przez WebRTC

ICE, skrót od Interactive Connectivity Establishment, to protokół używany w systemach komunikacji w czasie rzeczywistym do ustanawiania i utrzymywania połączeń między urządzeniami w sieciach. Jest on odpowiedzią na wyzwania stawiane przez firewalle, NAT-y i inne urządzenia sieciowe, które mogą utrudniać bezpośrednią komunikację peer-to-peer. ICE wykorzystuje kombinację technik, w tym STUN i TURN, aby określić najlepszą ścieżkę komunikacji między urządzeniami. Przeprowadza kontrole łączności, gromadzi informacje o sieci i ocenia dostępne protokoły transportowe oraz kandydatów do ustanowienia najbardziej wydajnego i niezawodnego połączenia. ICE dynamicznie dostosowuje się do warunków sieciowych, w razie potrzeby renegotując ścieżkę komunikacji i zapewniając nieprzerwaną transmisję audio, wideo lub danych. Inteligentnie wybierając optymalną trasę komunikacji ICE maksymalizuje szanse na udaną łączność i minimalizuje zależność od serwerów przekaźnikowych poprawiając ogólną jakość i wydajność komunikacji w czasie rzeczywistym w różnych środowiskach sieciowych [21].

NAT, czyli translacja adresów sieciowych, to technika stosowana w sieciach komputerowych w celu umożliwienia wielu urządzeniom współdzielenia jednego publicznego adresu IP. Jest ona powszechnie stosowana w sieciach domowych i biurowych w celu rozszerzenia ograniczonej dostępności adresów IPv4. NAT działa poprzez mapowanie prywatnych adresów IP używanych w sieci lokalnej na pojedynczy publiczny adres IP, który jest widoczny w Internecie. Proces ten umożliwia urządzeniom w sieci lokalnej dostęp do Internetu przy jednoczesnym zachowaniu poziomu bezpieczeństwa poprzez ukrywanie ich indywidualnych adresów IP przed podmiotami zewnętrznymi. NAT można wdrożyć na różne sposoby, takie jak statyczny NAT, dynamiczny NAT lub translacja adresów portów (PAT), która umożliwia wielu urządzeniom korzystanie z tego samego publicznego adresu IP poprzez przypisanie unikalnych numerów portów do każdego połączenia.

Podczas gdy NAT zapewnia skuteczne rozwiązanie do ochrony adresów IP i zabezpieczania sieci lokalnych, może wprowadzać komplikacje dla niektórych rodzajów komunikacji sieciowej, szczególnie w aplikacjach peer-to-peer i systemach komunikacji w czasie rzeczywistym. Protokoły takie jak STUN i TURN są często używane w połączeniu z NAT, aby sprostać tym wyzwaniom i ułatwić udaną komunikację między granicami sieci [21].

STUN, czyli Session Traversal Utilities for NAT, to protokół wykorzystywany w systemach komunikacji w czasie rzeczywistym do wykrywania i gromadzenia informacji sieciowych niezbędnych do nawiązywania połączeń peer-to-peer. Jest on szczególnie przydatny, gdy urządzenia znajdują się za NAT-ami lub zaporami sieciowymi, które uniemożliwiają bezpośrednią komunikację. STUN umożliwia urządzeniom określenie ich publicznego adresu IP, ocenę translacji adresów sieciowych i określenie typu używanego NAT. Wysyłając żądania STUN do serwera STUN urządzenia mogą uzyskać swój publiczny adres IP i wykryć obecność wszelkich urządzeń NAT na ścieżce komunikacyjnej. Informacje te mają kluczowe znaczenie dla negocjowania kanałów komunikacji i ułatwiania skutecznego przechodzenia przez NAT. STUN odgrywa istotną rolę w nawiązywaniu bezpośrednich połączeń peer-to-peer, zmniejszając zależność od serwerów przekaźnikowych i poprawiając wydajność i jakość komunikacji w czasie rzeczywistym [21].

TURN, czyli Traversal Using Relays around NAT, to protokół wykorzystywany w systemach komunikacji w czasie rzeczywistym, umożliwiający łączność pomiędzy urządzeniami znajdującymi się za translatorami adresów sieciowych (NAT) lub zaporami sieciowymi. W sytuacjach, gdy bezpośrednią komunikację peer-to-peer nie jest możliwa z powodu problemów z NAT, TURN służy jako serwer przekaźnikowy, który ułatwia transmisję danych między komunikującymi się stronami. Gdy nie można nawiązać bezpośredniego połączenia, klient wysyła swoje pakiety danych do serwera TURN, który następnie przekazuje je do zamierzonego odbiorcy. TURN odgrywa istotną rolę w zapewnieniu udanej komunikacji w scenariuszach, w których łączność peer-to-peer jest utrudniona przez restrykcyjne konfiguracje sieci. Pozwala na ustanowienie niezawodnych i bezpiecznych połączeń, umożliwiając transmisję audio, wideo lub danych w czasie rzeczywistym, nawet w złożonych środowiskach sieciowych. Dodaje to dodatkowy narzut czasowy i obliczeniowy, jest więc używane tylko jeżeli inne opcje są niedostępne [21]. Analiza z 2014 roku przeprowadzona przez 6 miesięcy wykazała, że 92% połączeń WebRTC korzysta bezpośrednio z połączenia P2P, a tylko 8% z serwerów TURN [20].

SDP, czyli Session Description Protocol, to standardowy protokół używany w systemach komunikacji w czasie rzeczywistym do opisywania sesji multimedialnych. Zapewnia on ustrukturyzowany format wymiany informacji o uczestnikach, typach mediów, kodekach, adresach sieciowych i innych parametrach niezbędnych do ustanowienia i zarządzania sesjami komunikacyjnymi. SDP odgrywa kluczową rolę w negocjowaniu możliwości i wymagań uczestniczących punktów końcowych, ułatwiając interoperacyjność różnych urządzeń i aplikacji. Wymieniając komunikaty SDP punkty końcowe mogą uzgodnić wspólny zestaw parametrów i ustanowić płynne połączenie dla transmisji audio, wideo lub danych [21].

2.4. Przegląd aktualnego stanu technologii

Przetwarzanie bezserwerowe wyłoniło się jako nowy trend, oferując nowatorskie podejście do rozwoju backendu. Platformy takie jak AWS Lambda, Azure Functions i Google Cloud Functions umożliwiają programistom skupienie się wyłącznie na pisaniu kodu, podczas gdy dostawca chmury obsługuje alokację zasobów i skalowanie. Taka architektura zwiększa elastyczność, zmniejsza koszty operacyjne i optymalizuje koszty. Architektura mikrousług nadal przekształca rozwój technologii backendowych, oferując modułowość, skalowalność i elastyczność. Dzieląc aplikacje na mniejsze, niezależne usługi, które mogą być rozwijane i wdrażane autonomicznie, mikrousługi sprzyjają łatwiejszej konserwacji i lepszemu wykorzystaniu zasobów. Kubernetes, platforma orkiestracji kontenerów, odgrywa kluczową rolę w zarządzaniu i skalowaniu mikrousług w spójny i zautomatyzowany sposób. GraphQL, język zapytań dla interfejsów API, zrewolucjonizował sposób pobierania danych i manipulowania nimi. Dzięki możliwości dostarczania dostosowanych odpowiedzi w oparciu o potrzeby klienta, GraphQL optymalizuje transmisję danych i ogranicza nadmierne pobieranie, przyczyniając się do skrócenia czasu ładowania i poprawy doświadczeń użytkowników. Technologie backendowe w coraz większym stopniu wykorzystują również możliwości uczenia maszynowego i sztucznej inteligencji. Integracja modeli uczenia maszynowego z aplikacjami umożliwia analitykę predykcyjną, zautomatyzowane podejmowanie decyzji i wgląd w dane.

Rozważanymi rozwiązaniami do stworzenia serwera aplikacji były Spring, Django oraz ASP.NET. Django to wysokopoziomowy framework webowy w języku Python, znany ze swojej filozofii "battery-included". Zapewnia deweloperom zaawansowany i wydajny zestaw narzędzi do szybkiego tworzenia aplikacji internetowych przy minimalnym powtarzaniu kodu. Podąża za wzorcem architektonicznym Model-Widok-Kontroler (MVC), który w Django jest reprezentowany jako Model-Widok-Szablon (MVT), ułatwiając rozdzielenie funkcji i utrzymanie czystego, zorganizowanego kodu. Django zawiera również system mapowania obiektowo-relacyjnego (ORM), upraszczający interakcje z bazą danych oraz wbudowany interfejs administratora do łatwego zarządzania danymi aplikacji. Nacisk na funkcje bezpieczeństwa, takie jak wbudowana ochrona przed typowymi lukami w zabezpieczeniach sieci jak SQL injection i cross-site scripting ułatwia tworzenie odpornych na ataki serwisów. Aktywna społeczność Django i obszerna dokumentacja dodatkowo przyczyniają się do jego popularności wśród programistów, czyniąc go doskonałym wyborem do tworzenia solidnych i skalowalnych aplikacji internetowych. Natomiast ASP.NET to wszechstronny i szeroko stosowany framework aplikacji internetowych opracowany przez firmę Microsoft. Umożliwia on programistom tworzenie dynamicznych, wysokowydajnych aplikacji i usług internetowych przy użyciu różnych języków programowania, w tym C# i VB.NET. ASP.NET opiera się na wzorcu architektonicznym Model-Widok-Kontroler (MVC), zapewniając ustrukturyzowane podejście do tworzenia stron internetowych, które poprawia organizację kodu i łatwość utrzymania. Dzięki bogatemu zestawowi bibliotek i narzędzi ASP.NET upraszcza typowe zadania związane z tworzeniem stron internetowych, od interakcji z bazą danych za pośrednictwem Entity Framework po uwierzytelnianie użytkowników za pomocą wbudowanego zarządzania tożsamością. Dodatkowo, ASP.NET Core, wieloplatformowa i open-source'owa ewolucja ASP.NET, rozszerza jego możliwości do działania na różnych platformach, w tym Linux i macOS [22]. Framework jest też silnie

zintegrowany z innymi platformami Microsoftu. Ostatecznie wybrany został jednak Spring z uwagi na wcześniejsze z nim zapoznajomienie.

W części frontendowej WebAssembly (Wasm) pojawił się jako przełom, umożliwiając programistom komplikowanie kodu z języków takich jak C++, Rust i innych do niskopoziomowego formatu binarnego, który może być wykonywany w przeglądarkach internetowych. Otwiera to nowe możliwości dla zadań wymagających dużej wydajności, takich jak przetwarzanie multimedii i gry, bez uszczerbku dla kompatybilności przeglądarki. Biblioteki zarządzania stanem, takie jak Redux i MobX, oferują skoncentrowane zarządzanie danymi, ułatwiając synchronizację danych w różnych częściach aplikacji i zapewniając spójne wrażenia użytkownika. Progresywne aplikacje internetowe (PWA) rewolucjonizują sposób tworzenia aplikacji internetowych, zapewniając natywne doświadczenia podobne do aplikacji w przeglądarkach internetowych. Wykorzystując technologie takie jak service workers i web manifests, PWA oferują dostęp offline, powiadomienia push i szybkie czasy ładowania, zwiększając możliwości użytkowników i dostępność.

Trzy technologie były brane pod uwagę przy tworzeniu części frontendowej: Vue.js, React oraz Angular. React to framework stworzony przez Facebooka. Znany ze swojego deklaratywnego i opartego na komponentach podejścia, React efektywnie zarządza interfejsem użytkownika (UI) aplikacji internetowych. Jego rdzeniem jest Virtual DOM, lekka reprezentacja rzeczywistego DOM w pamięci. React wykorzystuje go do obliczania najbardziej efektywnego sposobu aktualizacji interfejsu użytkownika w przypadku zmiany danych, minimalizując kosztową bezpośrednią manipulację rzeczywistym DOM. Komponenty Reacta są elementami składowymi interfejsu użytkownika, hermetyzującymi jego elementy i ich zachowanie. JSX (JavaScript XML), rozszerzenie składni JavaScript, pozwala definiować komponenty interfejsu użytkownika w formacie, który bardzo przypomina HTML [23]. Dzięki jednokierunkowemu przepływowi danych i ekosystemowi rozszerzeń, który obejmuje biblioteki zarządzania stanem, takie jak Redux, React ułatwia tworzenie złożonych i interaktywnych aplikacji internetowych przy jednoczesnym zachowaniu wydajności i organizacji kodu. Vue.js jest na wiele sposobów podobny do Reacta. Jego podstawowa architektura również koncentruje się wokół wirtualnego DOM, który umożliwia framework'owi efektywne śledzenie zmian w stanie aplikacji i selektywną aktualizację rzeczywistego DOM, co skutkuje optymalną wydajnością renderowania. Vue.js przyjmuje strukturę opartą na komponentach, w której interfejs użytkownika jest podzielony na samodzielne komponenty wielokrotnego użytku, które można łatwo komponować w celu tworzenia złożonych interfejsów. Wykorzystuje reaktywny system wiązania danych, umożliwiając programistom powiązanie elementów DOM ze stanem aplikacji i ich automatyczną aktualizację po zmianie danych. Vue.js zapewnia również kompleksowy zestaw narzędzi do routingu, zarządzania stanem i animacji [24].

W ostatnim czasie powstało również wiele rozwiązań do przechowywania dużych danych. Zdecydowanie najpopularniejszym i najbardziej znaczącym jest Amazon S3 (Simple Storage Service). Jest to jeden z podstawowych komponentów AWS (Amazon Web Services). S3 wykorzystuje płaską, rozproszoną architekturę, która pozwala na praktycznie nieograniczone przechowywanie danych [25]. Dane są zorganizowane w "wiadra", które są zasadniczo kontenerami dla obiektów, z których każdy jest identyfikowany przez unikalny klucz. Trwałość S3 osiąga się poprzez replikację danych w wielu centrach danych, zapewniając odporność i dostępność danych.

Zapewnia solidne funkcje bezpieczeństwa, w tym szyfrowanie danych w spoczynku i podczas przesyłania, a także oferuje precyzyjną kontrolę dostępu za pośrednictwem zasad AWS Identity and Access Management (IAM) i list kontroli dostępu (ACL). S3 płynnie integruje się również z usługami AWS, takimi jak Lambda, EC2 i CloudFront. Warto tutaj zwrócić uwagę na pewien problem. Pomimo tego, że API S3 jest bezspornie standardem wśród deweloperów, nie posiada on swojego standardu tj. nie został opisany w żadnym RFC ani innym tego typu dokumencie. Na dodatek jego wewnętrzne mechanizmy nie są publicznie znane. Inne produkty naśladując więc zachowanie, metody dostępu, funkcje S3, działa głównie na publicznie znanych założeniach, które Amazon może dowolnie modyfikować. Jednym z takich rozwiązań jest Ceph. Jest to rozproszony system pamięci masowej typu open-source, który obejmuje unikalną architekturę zapewniającą skalowalność i odporność w zarządzaniu przechowywaniem danych na dużą skalę. Składa się on z wielu komponentów, w tym RADOS (Reliable Autonomous Distributed Object Store), RBD (RADOS Block Device) i CephFS (Ceph File System), współpracujących ze sobą. U podstaw Ceph opiera się na klastrze RADOS, który jest zbiorem OSD (Object Storage Daemons), które przechowują dane w wielu węzłach. Klaster ten zapewnia redundancję danych poprzez automatyczną replikację między OSD i może obsługiwać jednocześnie odczyty i zapisy na dużą skalę. Rozproszony charakter Ceph oznacza również, że nie ma pojedynczego punktu awarii, co zwiększa dostępność danych i odporność na awarie. Serwis obsługuje różne metody dostępu, dzięki czemu nadaje się do różnych zastosowań. Na przykład RADOS Gateway zapewnia obiektową pamięć masową zgodną ze wspomnianym API S3, jak również Swift, podczas gdy RBD umożliwia blokową pamięć masową, która może być używana z maszynami wirtualnymi i kontenerami. Dodatkowo, CephFS oferuje rozproszony system plików zgodny ze standardem POSIX. Funkcje, takie jak warstwowanie danych i erasure coding, przyczyniają się do wydajności i bezpieczeństwa Ceph. Co więcej, adaptacyjność rozciąga się na obsługę różnych nośników pamięci masowej, w tym tradycyjnych dysków twardych i szybszych, bardziej trwałych dysków SSD [26].

3. PROJEKT I IMPLEMENTACJA

3.1. Specyfikacja wymagań

Wymagania dla aplikacji zostały starannie zidentyfikowane poprzez współpracę z klientem, Orkiestrą Modena Molti Mondi. Rozpoczęto od przeprowadzenia wywiadu, w trakcie którego zebrano podstawowe informacje dotyczące oczekiwania wobec planowanego muzycznego serwisu edukacyjnego. Ten etap pozwolił na określenie kluczowych funkcji, interfejsu użytkownika oraz ogólnego celu aplikacji.

Kolejnym krokiem było opracowanie wstępnego prototypu, który przedstawiał podstawowe struktury i funkcjonalności aplikacji. Prototyp ten został zaprezentowany klientowi, a po jego analizie zebrano cenne uwagi i opinie. Te informacje stanowiły cenny materiał do dalszej iteracji projektu. Uwzględniając te uwagi, przystąpiono do implementacji zmian i dodatkowych funkcji w prototype.

Ta ulepszona wersja została ponownie przedstawiona klientowi, a proces iteracyjny, w którym zbierano uwagi i wprowadzano zmiany, powtórzono kilkukrotnie. To cykliczne podejście pozwoliło na stopniowe dostosowywanie projektu do oczekiwania klienta oraz eliminowanie potencjalnych problemów czy niedoprecyzowanych elementów. W efekcie tych iteracji i uwzględniania sugestii klienta, uzyskano ostateczną specyfikację aplikacji, która stała się podstawą do implementacji gotowego projektu.

3.2. Projekt rozwiązania

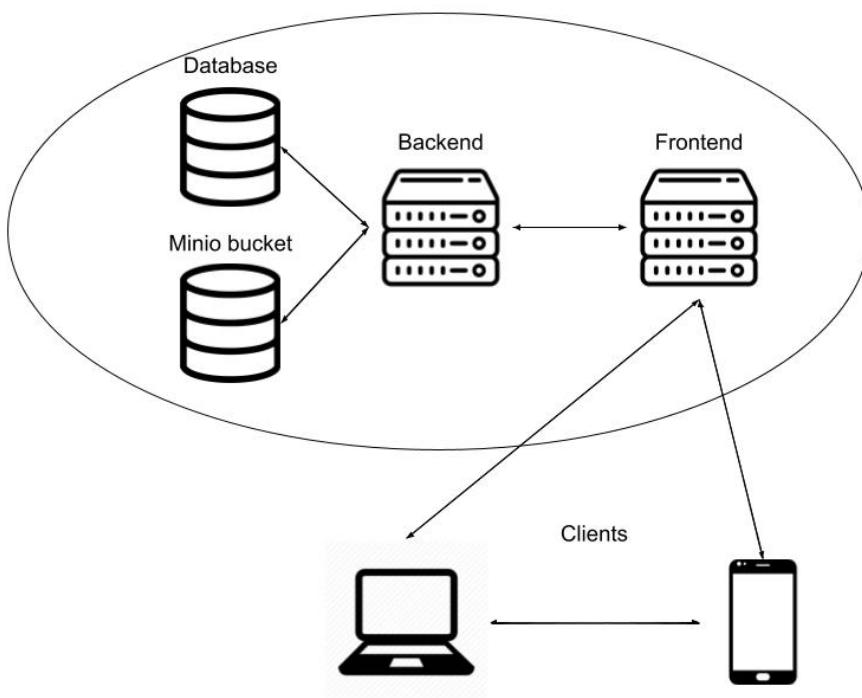
Aplikacja została podzielona na kilka komponentów [3.1], wszystkie są skonteneryzowane i uruchomione w środowisku dockerowym:

3.2.1. Serwer aplikacji

Najważniejszym z komponentów jest serwer aplikacji. Został on napisany z użyciem framework'a Spring. Z frontendem komunikuje się poprzez API RESTowe. Zajmuje się wszystkimi zapytaniami wysyłanymi przez aplikacje, służy również jako serwer sygnalizujący dla protokołu WebRTC. Do tego celu wykorzystany został protokół WebSocket. Serwer służy jednak tylko do znalezienia się rozmówców w sieci, sama rozmowa odbywa się zwykle bez udziału serwera co znacznie obniża jego potencjalne obciążenie. Uwierzytelnianie i autoryzacja użytkowników obsługiwane są za pomocą tokenów JWT.

3.2.2. Frontend

Użytkownicy mają dwie możliwe opcje korzystania z aplikacji. Pierwszą z nich jest strona internetowa napisana we framework'u Angular natomiast drugą jest aplikacja mobilna na Androïda stworzona w Kotlinie. Oba rozwiązania umożliwiają nagrywanie i odtwarzanie plików zapisanych



Rysunek 3.1: Architektura aplikacji

w kursach jak również uczestniczenie w wideo-rozmowach.

3.2.3. Bazy danych

W projekcie wykorzystane zostały dwie bazy danych. Pierwszą z nich jest tradycyjna baza SQL, w tym przypadku PostgreSQL. Służy ona do przechowywania danych o kursach, użytkownikach i innych danych tekstowych tego typu. Drugą jest MinIO, w której magazynowane są wszystkie pliki multimedialne takie jak nagrania uczniów czy materiały, których przechowywanie w zwykłej bazie danych byłoby nieefektywne.

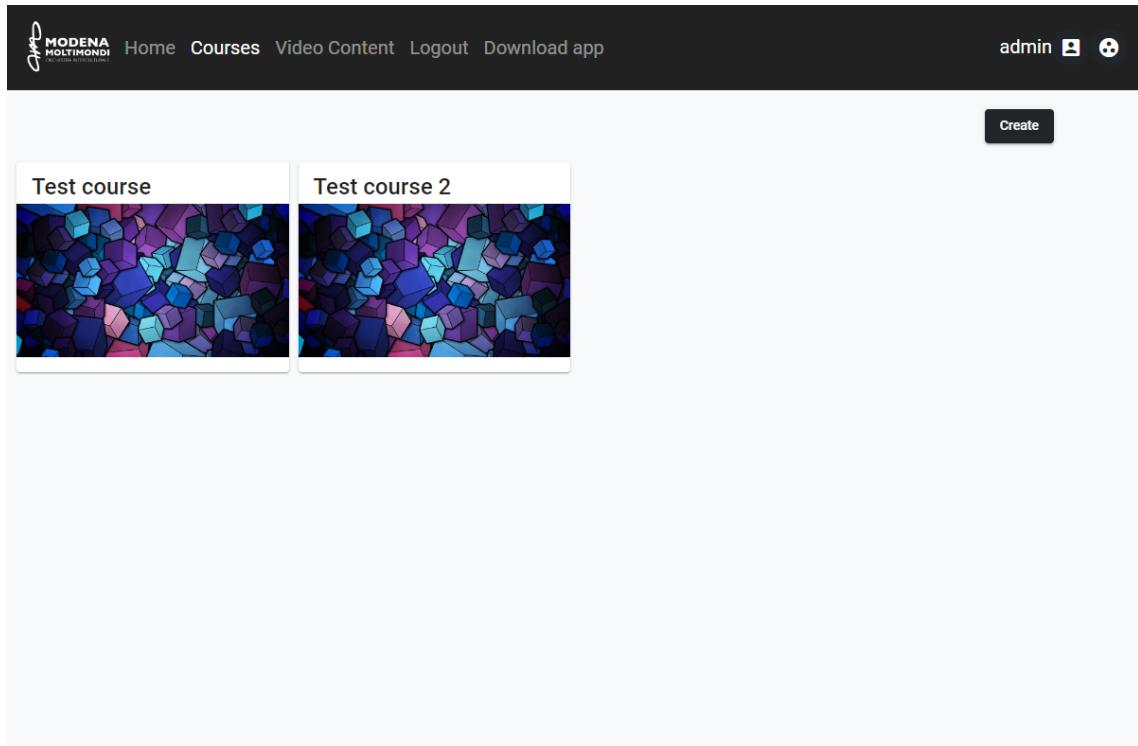
Dane uwierzytelniające do baz danych w domyślnym środowisku to:

Baza	Login	Hasło
PostgreSQL	musichubpgsql	musichubpgsql
MinIO	msi-minio	msi-minio

3.3. Interfejs użytkownika

Aplikacja składa się z kilku widoków. Użytkownik porusza się pomiędzy nimi za pomocą paska nawigacyjnego na górze strony, dostępnego z każdego miejsca w aplikacji. Widok pod-

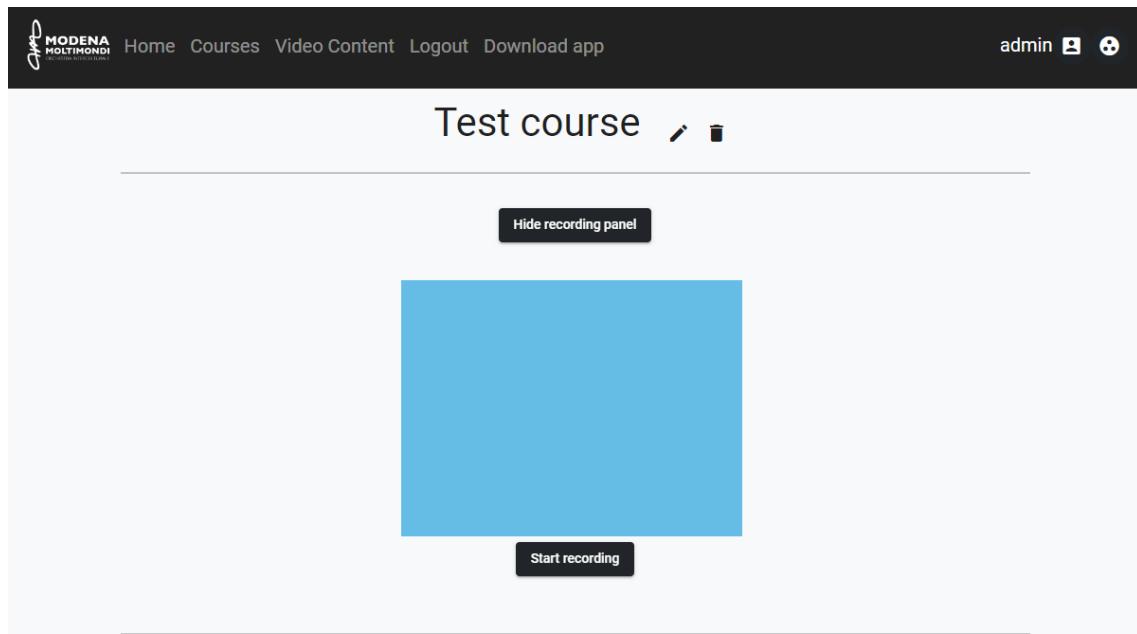
stawowy, dostępny również dla niezalogowanych użytkowników, zawiera jedynie obraz powitalny. Następymi widokami są widoku kursów i materiałów wideo, bardzo do siebie podobne [3.2].



Rysunek 3.2: Widok wszystkich kursów

Wyświetlona jest na nich lista wszystkich dostępnych dla zalogowanego użytkownika kursów / materiałów wideo, udostępniony jest też przycisk tworzenia nowych dla uprawionych do tego użytkowników. Kolejnym widokiem jest widok kursu, wyświetlony po wybraniu go z listy. To w nim dostępna jest większość głównych funkcjonalności aplikacji. Na górze strony umieszczone zostały panel nagrywania pozwalający na tworzenia i przesyłanie nagrań bezpośrednio z poziomu aplikacji [3.3].

Poniżej są sekcje kursu wraz z podsekcjami, w których użytkownicy może umieszczać załączniki [3.4]. Są one metodą na organizację i dzielenie kursów na części dla zwiększenia czytelności i ułatwienia pracy.

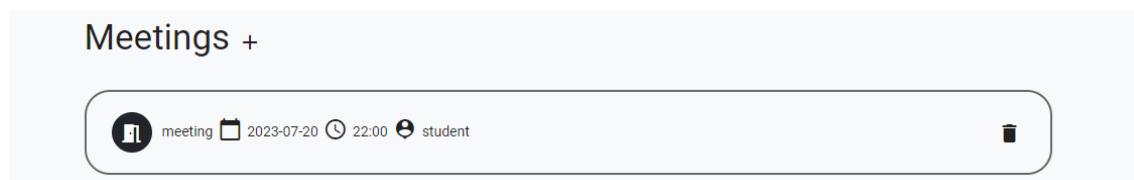


Rysunek 3.3: Panel nagrywania

A screenshot showing two nested sections. The outer section is titled "Sections +" and contains a single item named "test section". The inner section, titled "Subsections +", contains a single item named "subsection". Both sections have their own "Attachments" and "Participants' attachments" fields, each containing one file: "974268.jpg" and "20210326_143732.jpg" respectively. Each attachment row includes standard file management icons (trash, image, download) and a plus sign for adding more files.

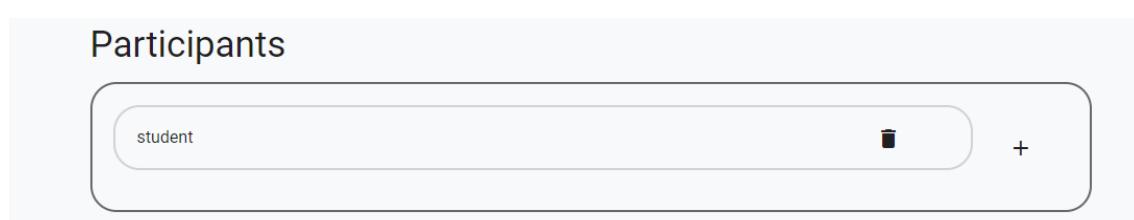
Rysunek 3.4: Sekcje i podsekcje

Następne są spotkania, wystawianie przez jednego z nauczycieli przypisanych do kursu dla danego ucznia na konkretną datę [3.5].



Rysunek 3.5: Spotkania

Kolejny jest panel członków kursu [3.6]. Widoczna jest w nim lista uczniów przypisanych w danym momencie do kursu. Daje on również możliwość ich dodawania lub usuwania.



Rysunek 3.6: Członkowie kursu

Ostatnią częścią widoku jest panel nauczycieli, widoczny jedynie dla właściciela kursu. Analogicznie do listy uczniów wyświetla on listę dodatkowych nauczycieli w kursie [3.7].



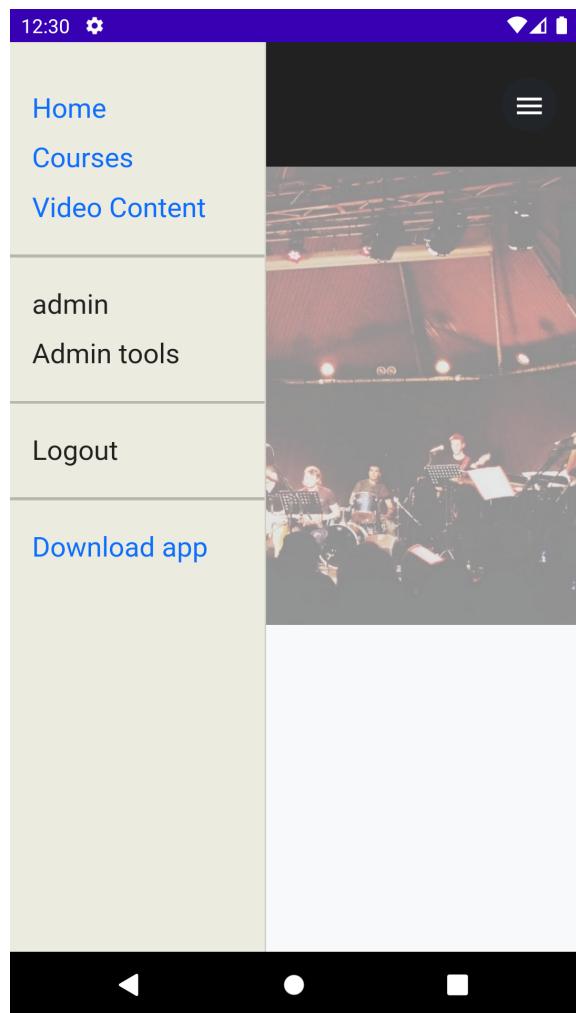
Rysunek 3.7: Nauczyciele

Widok materiałów wideo wygląda niemal identycznie jak widok kursu, jedyną różnicą jest brak panelu spotkań.

Jest również specjalny widok kontrolny dostępny dla admina aplikacji. Zawarta w nim jest lista wszystkich użytkowników i ich typów, możliwość tworzenia nowych nauczycieli jak również edycji istniejących kont.

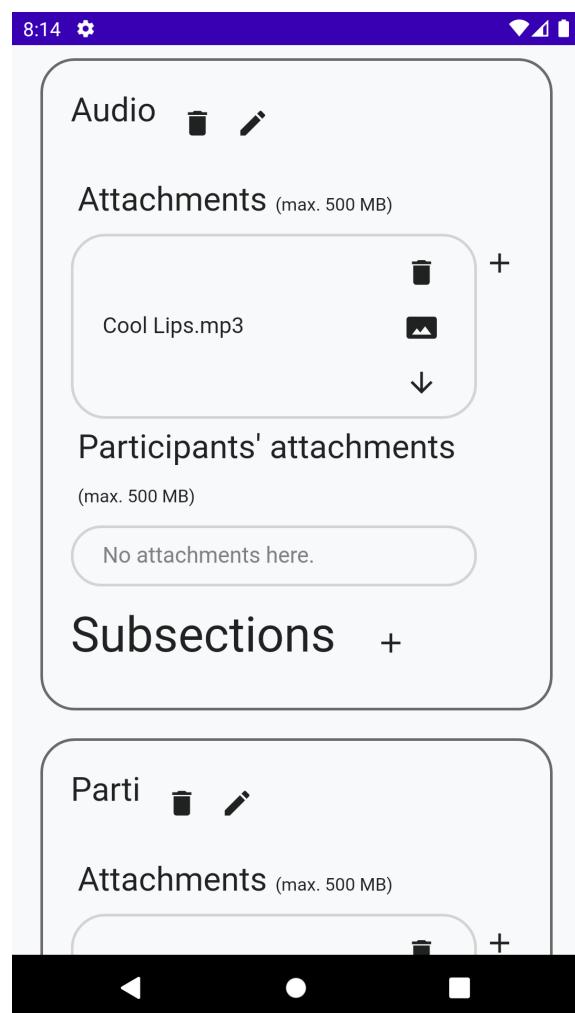
3.4. Aplikacja mobilna

Aplikacja mobilna oparta jest o system WebView. Jest to komponent, który umożliwia wyświetlanie treści internetowych bezpośrednio w aplikacji. Przy jego pomocy wyświetlany jest interfejs aplikacji, zmodyfikowany do wymiarów i możliwości urządzeń mobilnych. Pasek nawigacji został zamieniony na tzw. hamburger menu, co pozwala na dużo wygodniejszą obsługę strony na ekranie telefonu [3.8].

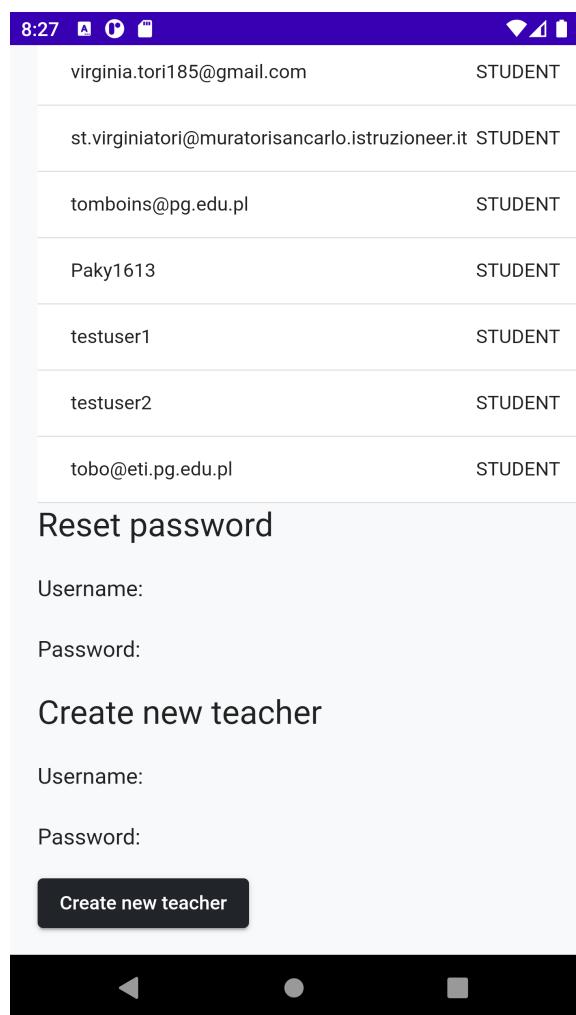


Rysunek 3.8: Menu na urządzeniu mobilnym

Widoki zostały zmienione na pionowe w celu uzyskania bardziej ergonomicznej obsługi [3.9][3.10]. Dodatkowo stworzone zostały skrypty uruchamiane przy niektórych funkcjach aplikacji. Udostępniają one warstwę pośrednią, dzięki której aplikacja przeglądarkowa jest w stanie korzystać z możliwości sprzętowych urządzenia poprzez wywoływanie odpowiednich funkcji i optymalizowanie ustawień w SDK Androida.

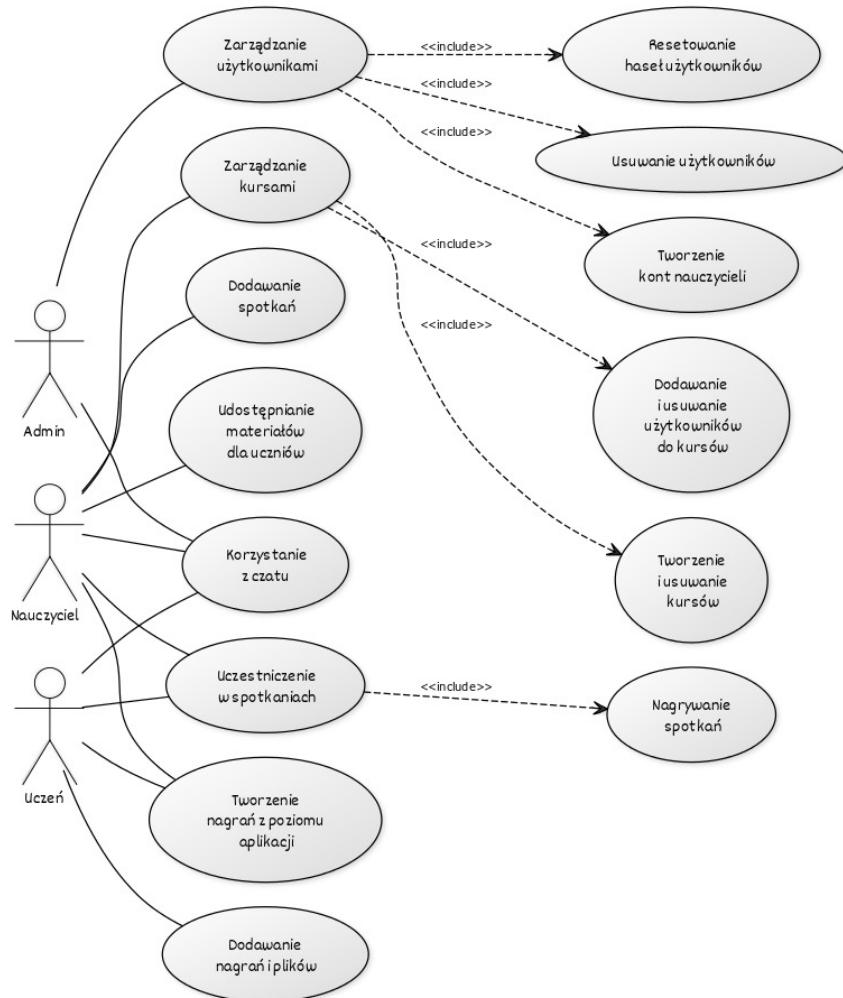


Rysunek 3.9: Widok sekcji na urządzeniu mobilnym



Rysunek 3.10: Panel narzędzie administratora na urządzeniu mobilnym

3.5. Rodzaje użytkowników



CREATED WITH YUML

Rysunek 3.11: Diagram przypadków użycia

3.5.1. Uczeń

Konto ucznia jest kontem o bardzo ograniczonych uprawnieniach. Jest ono tworzone samodzielnie za pomocą formularza rejestracyjnego a do każdego kursu musi zostać dodane przez jego właściciela. Student ma dostęp do materiałów dodawanych przez nauczyciela, może również dodawać własne nagrania do oceny. Co więcej, posiada dostęp do wystawionych przez pedagoga spotkań, nie ma jednak możliwości tworzyć ich samodzielnie.

3.5.2. Nauczyciel

Konta nauczycielskie dla uproszczenia zarządzania uprzywilejowanymi kontami są tworzone przez administratora, a następnie przekazywane odpowiednim użytkownikom. Eliminuje to potrzebę posiadania osobnego serwera mailowego, który byłby używany tylko w tym jednym celu. Konta te mogą tworzyć i zarządzać kursami poprzez dodawanie nowych sekcji i materiałów dla uczniów, mają również możliwość organizowania spotkań z podopiecznymi oraz wglądu do wszystkich przesłanych przez nich nagrań. Każdy nauczyciel może dodawać do swoich własnych kursów zastępów, którzy mają takie same przywileje jak on sam, nie mają jednak uprawnień do usuwania wcześniejszych utworzonych lekcji.

3.5.3. Administrator

Konto posiadające najszersze prawa ze wszystkich. Ma pełny dostęp do edytowania i usuwania wszystkich kursów jak również specjalny panel do zarządzania użytkownikami, tworzenia nowych kont nauczycielskich i resetowania haseł. Nie ma możliwości stworzenia nowego konta o takich samych uprawnieniach, istnieją jedynie te, które zostały zainicjalizowane w konfiguracji przed uruchomieniem aplikacji. W domyślnym środowisku login dla konta administratora to *admin*, natomiast hasło to *password*.

Wszyscy użytkownicy posiadają również dostęp do czatu wewnętrz každego kursu [3.11].

3.6. Implementacja i użyte technologie

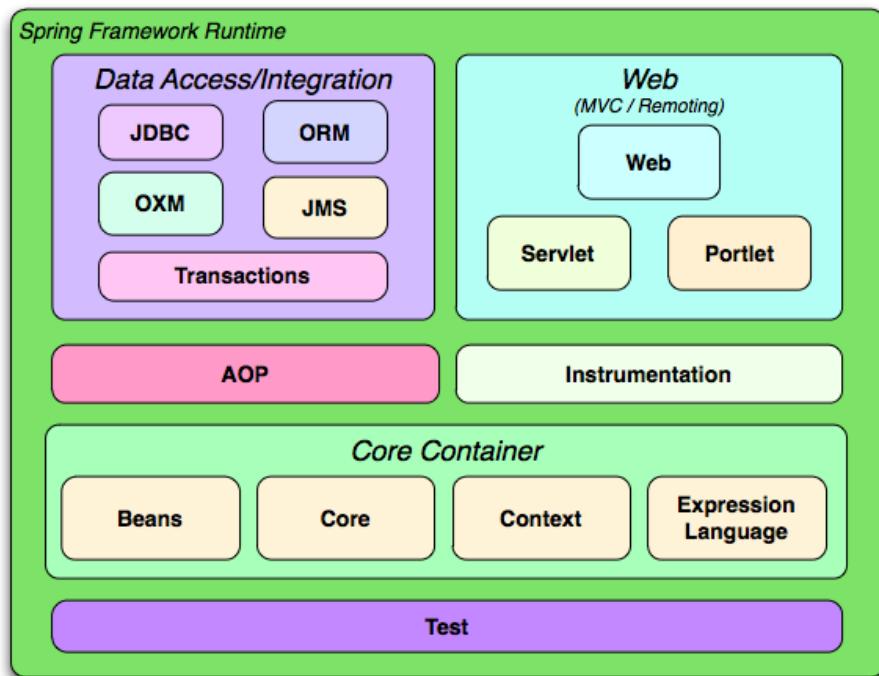
Kluczowym etapem w procesie tworzenia projektu był strategiczny wybór odpowiednich technologii, na których oparto całą konstrukcję. Wybór ten miał na celu zapewnienie solidnej podstawy dla projektu, opartej na wysokiej klasy rozwiązaniach, które gwarantują efektywne utrzymanie, skalowalność i niezawodną pracę systemu. Wszystkie zastosowane technologie zostały wybrane z myślą o spełnieniu wymagań projektu oraz długoterminowej wydajności. Każda z nich była wyselekcjonowana w oparciu o jej zdolność do integracji, elastyczność i gotowość do radzenia sobie z potencjalnymi wyzwaniami w przyszłości.

W przypadku komponentów, dla których istnieje wiele wysokiej jakości opcji, zdecydowano się na wykorzystanie tych najbardziej sprawdzonych i znajomych. To zapewniło nie tylko solidność technologiczną, ale także ułatwiło proces utrzymania, rozwoju i ewentualnych modyfikacji w przyszłości. Wybór takich komponentów pozwolił na łatwiejszą pracę oraz dostęp do szerokiej bazy wiedzy i zasobów online.

3.6.1. Spring Boot

Platforma Spring jest frameworkiem aplikacji oraz kontenerem inwersji kontroli (Inversion of Control - IoC) dla platformy Java. Wydana po raz pierwszy w 2002 roku jako alternatywa dla Java Enterprise Edition. Jest lekka (podstawowa wersja waży ok. 2MB) ale również wysoce konfigurowalna.

Rdzeniem frameworka Spring jest kontener Inversion of Control (IoC), znany również jako



Rysunek 3.12: Diagram framework'a Spring

kontener Spring. Kontener zarządza cyklem życia obiektów Java i ich zależnościami, zmniejszając złożoność tworzenia obiektów i ręcznego zarządzania zależnościami. Ta funkcja IoC promuje luźne sprzężenie i pozwala na bardziej modułowy i łatwy w utrzymaniu kod [3.12].

Rozbudowany ekosystem Spring obejmuje różne moduły i rozszerzenia, które zaspakajają różne potrzeby aplikacji. Na przykład moduł Spring MVC zapewnia architekturę model-widok-kontroler do tworzenia aplikacji internetowych, umożliwiając programistom tworzenie interfejsów API RESTful i dynamicznych stron internetowych. Moduł Spring Data upraszcza dostęp do danych, zapewniając spójny i intuicyjny interfejs API do pracy z bazami danych. Spring Security oferuje solidny framework do implementacji uwierzytelniania i autoryzacji w aplikacjach, zapewniając bezpieczny dostęp do zasobów.

Lista niektórych modułów Springa:

- Spring Core Container - podstawowy moduł zawierający komponenty niezbędne do działania aplikacji (BeanFactory i ApplicationContext),
- Spring Security - zawiera konfigurowalne procesy autoryzacji i autentykacji, wspierający szereg narzędzi, standardów i protokołów,
- Spring AOP - dodaje możliwość programowania aspektowego,
- Spring Data - umożliwia pracę z relacyjnymi bazami danych przy pomocy API JDBC, jak również z bazami NoSQL,

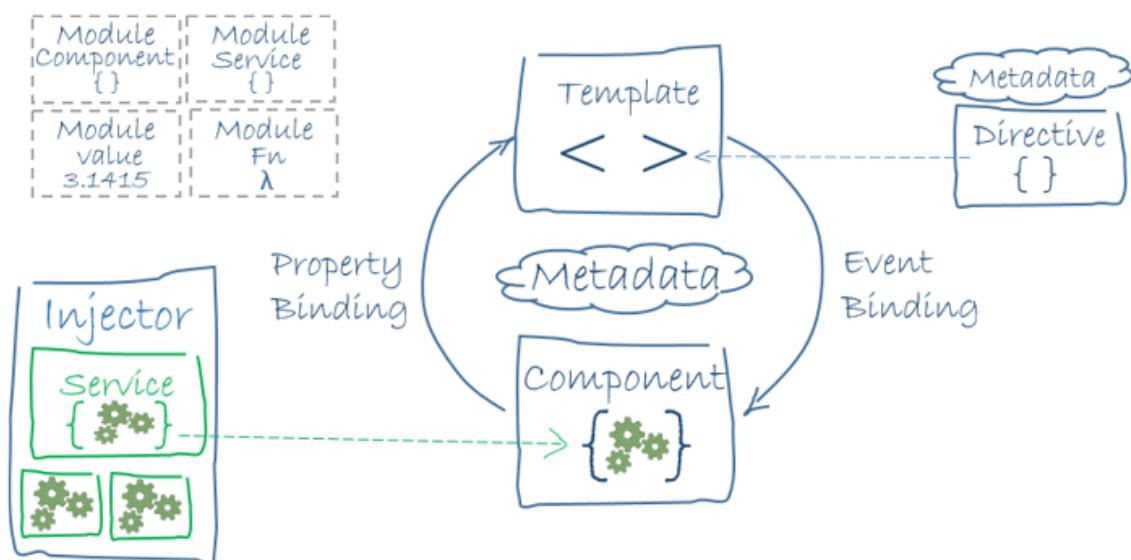
- Spring MVC - framework pozwalający na tworzenie aplikacji webowych w stylu Model-View-Controller oraz serwisów RESTowych,
- Spring Test - umożliwia testowanie komponentów Springowych przy pomocy platform takich jak JUnit lub TestNG. Zapewnia ładowanie odpowiednich kontekstów aplikacji oraz ich zapisywanie, zawiera również wiele przydatnych obiektów zaślepek (Mock).

Kolejnym kluczowym atutem Spring jest wsparcie dla integracji z innymi technologiami. Spring płynnie integruje się z popularnymi frameworkami, takimi jak Hibernate, JPA i Apache Kafka, umożliwiając programistom wykorzystanie istniejących narzędzi i bibliotek w aplikacjach. Spring zapewnia również szerokie wsparcie dla testowania, dzięki modułom takim jak Spring Test i Mockito, ułatwiając pisanie kompleksowych testów jednostkowych i integracyjnych dla komponentów.

Spring zapewnia też moduł programowania aspektowego. AOP umożliwia programistom modularyzację zagadnień przekrojowych, takich jak rejestrowanie, bezpieczeństwo i zarządzanie transakcjami, i zastosowanie ich do wielu komponentów aplikacji. Takie podejście promuje wielokrotne wykorzystanie kodu oraz separację problemów.

3.6.2. Angular

Istotnym elementem każdej aplikacji webowej jest jej interfejs. Istnieje wiele frameworków pozwalających na budowę samodzielnych modułów frontendowych zarówno niezależnych jak np. Vue.js, jak również zarządzanych przez wielkie firmy jak React od Facebooka i Angular od Google'a. Ten ostatni został wybrany jako narzędzie do stworzenia aplikacji. Najnowszą wersję frameworka jest 13.1.1 wydana 5 miesięcy temu.



Rysunek 3.13: Angular architecture

Jedną z kluczowych zalet Angulara jest jego system szablonów. Wykorzystując HTML

jako język szablonów, Angular pozwala programistom na płynne łączenie dynamicznej zawartości i statycznych znaczników HTML, ułatwiając tworzenie interaktywnych interfejsów użytkownika. Dwukierunkowe wiązanie danych umożliwia automatyczną synchronizację między modelem danych aplikacji a widokiem, zmniejszając potrzebę ręcznej manipulacji DOM i zwiększać wydajność programowania.

Kolejnym plusem frameworka jest jego modułowa architektura [3.13]. Aplikacje składają się z bloków zwanych komponentami. Zawierają one szablony HTML, style CSS i logikę związaną z określoną częścią interfejsu użytkownika. To modułowe podejście promuje możliwość ponownego wykorzystania kodu, łatwość konserwacji i testowania, ułatwiając tworzenie i utrzymywanie aplikacji.

Ponadto Angular oferuje kompleksowy zestaw narzędzi i bibliotek usprawniających proces programowania. Angular CLI (Command Line Interface) zapewnia mechanizmy automatyzujące typowe zadania, takie jak konfiguracja projektu, generowanie komponentów i optymalizacja komplikacji. Obszerna kolekcja komponentów Angular Material oferuje wstępnie stylizowane komponenty interfejsu użytkownika, które są zgodne z wytycznymi Material Design, umożliwiając programistom tworzenie atrakcyjnych wizualnie i spójnych interfejsów użytkownika przy minimalnym wysiłku. Framework korzysta z wielu technik optymalizacyjnych, takich jak lazy loading, komplikacja z wyprzedzeniem (AOT) oraz tree-shaking, dla zmniejszenia rozmiaru aplikacji i czasów ładowania.

3.6.3. PostgreSQL

PostgreSQL, znany również jako Postgres, jest darmowym i otwartym systemem zarządzania relacyjną bazą danych. Stworzony na Uniwersytecie Kalifornijskim w Berkeley w 1995 roku jako Ingres został przemianowany na PostgreSQL w 1996 aby zaznaczyć swoje wsparcie dla SQL. Przez wielu uważany za najbardziej zaawansowany otwarte-źródłowy system zarządzania bazami danych, co sprawia, że jest często używany do celów naukowych. Jest również wybierany jako baza dla komercyjnych produktów do analizy danych i zaawansowanych baz takich jak Greenplum czy Vertica [27].

Jedną z wyróżniających się cech PostgreSQL jest jego zgodność ze standardem SQL. Obsługuje szeroki zakres funkcji SQL, zapewniając kompatybilność z innymi systemami baz danych i ułatwiając migrację istniejących aplikacji do PostgreSQL. Co więcej, PostgreSQL wykracza poza standardowe możliwości SQL i oferuje wiele dodatkowych funkcji, które odróżniają go od innych baz danych.

PostgreSQL zapewnia obsługę zaawansowanych typów danych, w tym JSON, XML, a nawet typów zdefiniowanych przez użytkownika. Ta elastyczność pozwala programistom na wydajne modelowanie złożonych struktur danych i obsługę różnych typów informacji w jednej bazie danych. Dodatkowo, PostgreSQL zawiera potężne wbudowane funkcje i operatory, które umożliwiają złożone przetwarzanie i analizę danych.

Inną godną uwagi cechą PostgreSQL jest obsługa wyszukiwania pełnotekstowego. Zapewnia solidną i wydajną wyszukiwarkę tekstuową, która pozwala programistom na wbudowanie

zaawansowanych funkcji wyszukiwania w ich aplikacjach. Funkcja ta jest szczególnie przydatna w aplikacjach obsługujących duże ilości danych tekstowych, takich jak systemy zarządzania treścią, platformy handlu elektronicznego i bazy wiedzy.

Pod względem wydajności i skalowalności, PostgreSQL jest wysoko ceniony. Jego architektura została zaprojektowana do obsługi dużych obciążeń i dużych zbiorów danych, zapewniając doskonałą wydajność nawet przy dużych obciążeniach współbieżnych. Wsparcie PostgreSQL dla zaawansowanych technik indeksowania, optymalizacji zapytań i równoległego wykonywania zapytań przyczynia się do jego wyjątkowej wydajności.

Integralność i niezawodność danych są niezwykle ważne w każdym systemie baz danych. Postgres oferuje integralność transakcyjną, umożliwiając programistom utrzymanie spójności i integralności danych podczas wielu operacji. Zapewnia również obsługę wielowersyjnej kontroli współbieżności (MVCC), zapewniając wysoki poziom współbieżności bez poświęcania integralności danych. Co więcej, oferuje szereg opcji tworzenia kopii zapasowych i replikacji, umożliwiając programistom wdrażanie strategii ochrony danych i wysokiej dostępności.

Otwarty charakter PostgreSQL przyczynił się do powstania aktywnej społeczności. Składa się ona z programistów, administratorów baz danych i entuzjastów, którzy przyczyniają się do ciągłego rozwoju, ulepszania i wspierania bazy danych. Ta aktywna społeczność zapewnia, że PostgreSQL pozostaje na bieżąco z najnowszymi trendami i standardami branżowymi.

Ponadto PostgreSQL korzysta z bogatego ekosystemu rozszerzeń i narzędzi. Oferuje szeroką gamę rozszerzeń, które zwiększają jego funkcjonalność, takich jak obsługa dodatkowych typów danych, optymalizacje wydajności i integracja z innymi technologiami. Ponadto istnieje wiele graficznych interfejsów użytkownika, narzędzi administracyjnych i frameworków, które upraszczają zarządzanie bazami danych ich rozwój.

3.6.4. MinIO

MinIO to serwis pamięci masowej wysokiej wydajności oparty na 'bucketach' i kompatybilny pod względem API z Amazon S3. Posiada wiele kluczowych funkcjonalności koniecznych do bezpiecznego przechowywania dużych zbiorów obiektów [28].

Skalowalność

Jedną z kluczowych cech MinIO jest jego rozproszona architektura, która umożliwia skalowanie w poziomie na wielu serwerach. Oznacza to, że wraz ze wzrostem potrzeb organizacji w zakresie pamięci masowej, dodatkowe instancje MinIO mogą być płynnie dodawane w celu zwiększenia pojemności i wydajności.

Aktywna replikacja

MinIO wspiera synchroniczną i prawie synchroniczną replikację danych (w zależności od ograniczeń architektonicznych i tempa modyfikacji danych). W obu przypadkach głównym założeniem i celem jest aby replikacja była jak najbardziej spójna i przewidywalna (biorąc pod uwagę

przepustowość łącza i ilość aktualizacji).

Szyfrowanie

Aby zapobiec atakom typu 'Man-in-the-middle' używany jest protokół TLS v1.2+ pomiędzy wszystkimi komponentami w klastrze. Zapewnia to bezpieczeństwo danych zarówno w komunikacji wewnętrz systemu jak i z klientami zewnętrznymi. Implementacja protokołu jest zoptymalizowana na poziomie instrukcji procesora co daje minimalny narzut na wydajność. Dane zaszyfrowane są również podczas przechowywania. Dzieje się to automatycznie przed zapisaniem ich na dysk. Używany jest do tego schemat szyfrowania uwierzytelnionego (AEAD) który wspiera wiele popularnych algorytmów takich jak AES-256-GCM oraz ChaCha20-Poly1305 dla zapewnienia użytkownikowi jak największej elastyczności. Użycwanie możliwości sprzętowych takich jak funkcje SIMD sprawia, że narzut dla tego szyfrowania jest znikomy.

Kompatybilność z Amazon S3

Amazon S3 i jego API stało się niejako standardem dla rozwiązań chmurowych i zmusiło innych twórców do tworzenia kompatybilnych serwisów. MinIO bardzo kompleksowo wspiera to API, co pozwala na używanie go w wielu typach środowisk w zależności od potrzeb i sytuacji, jak również na płynne przechodzenie pomiędzy klastrami lokalnymi i chmurowymi.

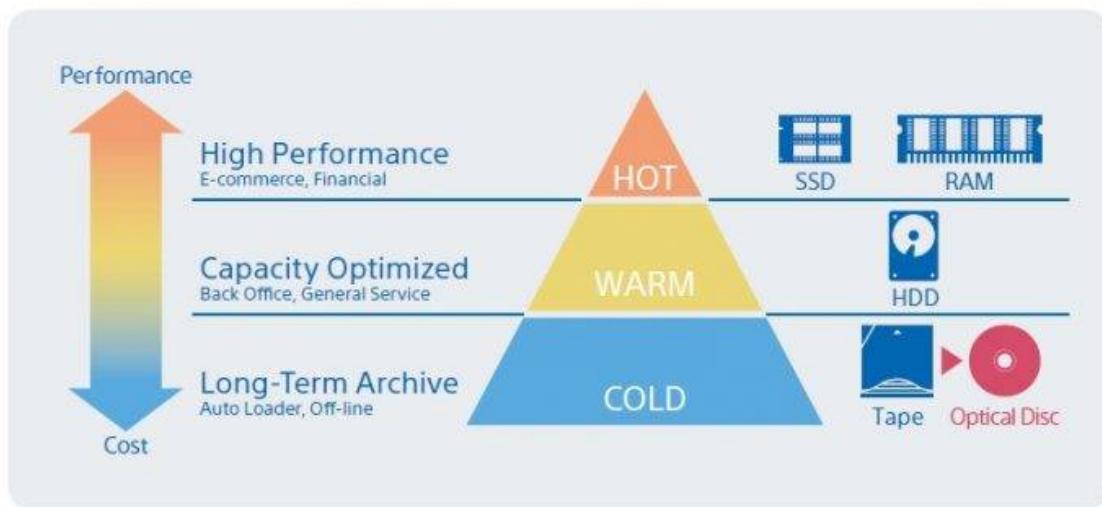
Poziomowanie danych

MinIO wspiera wiele sposobów poziomowania danych. Najprostszym jest rozkładanie danych na różnych typach nośników, zwykle w zależności od daty ich ostatniego użycia, jednak mogą tu być dowolne algorytmy. Taki system posiada najczęściej dwa lub trzy poziomy [3.14], gdzie najszybszym, a jednocześnie najmniejszym są dyski SSD (przeważnie NVMe). Drugie w kolejności są dyski SATA SSD i/lub HDD. Są one dużo większe lecz dużo wolniejsze stąd trzymane są na nich dane do których dostęp nie jest tak krytyczny. Stosuje się też poziomowanie z użyciem chmury gdzie często przejmuje ona rolę większych, ale wolniejszych dysków. MinIO wspiera też poziomowanie pomiędzy różnymi chmurami. Pozwala to na ujednolicenie dostępu do danych dla aplikacji, gdyż komunikuje się ona tylko z jednym endpointem, a serwer sam decyduje z którego miejsca pobrać dane oraz ustala polityki przechowywania różnych typów danych w poszczególnych poziomach według wytycznych.

3.6.5. Docker

Docker to platforma open-source, która umożliwia deweloperom automatyzację wdrażania i zarządzania aplikacjami w kontenerach. Kontenery są lekkimi, izolowanymi środowiskami, które zawierają aplikację i jej zależności, umożliwiając spójne i niezawodne wdrażanie na różnych środowiskach.

Jedną z kluczowych zalet Dockera jest jego zdolność do zapewnienia spójnego i powtarzalnego środowiska dla aplikacji. Dzięki Dockerowi deweloperzy mogą spakować swoje aplikacje



Rysunek 3.14: Poziomowanie danych

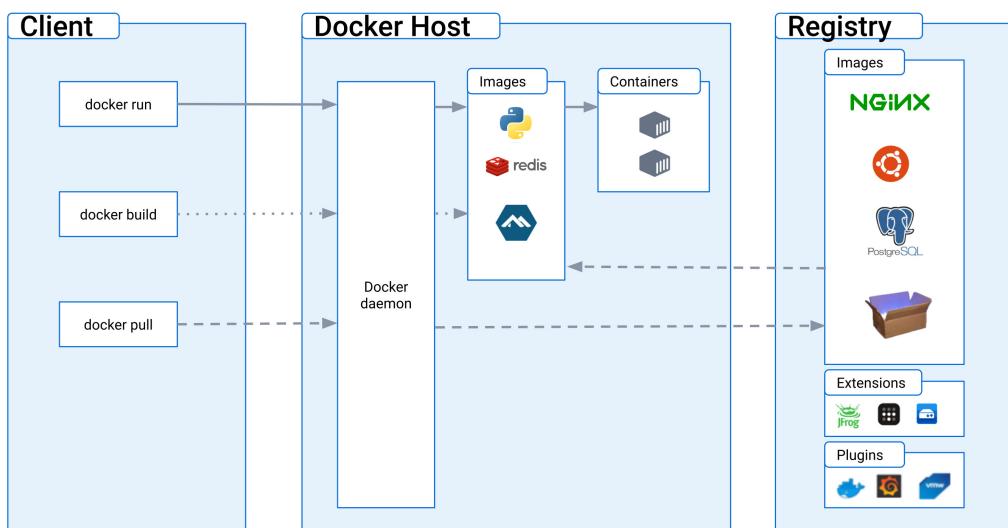
wraz z ich zależnościami do obrazu kontenera. Obraz ten zawiera wszystko, co jest potrzebne do uruchomienia aplikacji. Ta spójność zapewnia, że aplikacja zachowuje się w ten sam sposób niezależnie od systemu hosta czy infrastruktury.

Docker upraszcza proces tworzenia, udostępniania i wdrażania aplikacji kontenerowych. Programiści mogą korzystać z interfejsu wiersza poleceń lub GUI, aby łatwo tworzyć kontenery i zarządzać nimi. Obrazy dockerowe mogą być udostępniane i dystrybuowane za pośrednictwem rejestrów dockerowych, umożliwiając współpracę i usprawnienie procesów wdrażania. Działają one podobnie do repozytoriów oprogramowania umożliwiając pushowanie i pullowanie obrazów z jednego źródła. Wyróżnianie są dwa typy rejestrów: publiczne (np. Docker Hub) i prywatne. Dodatkowo, narzędzia do orkiestracji kontenerów, takie jak Docker Swarm czy Kubernetes, pozwalają na zarządzanie wieloma kontenerami w klastrze maszyn, zapewniając skalowalność i wysoką dostępność aplikacji.

Kolejną istotną zaletą Dockera jest jego wydajność i wykorzystanie zasobów. Kontenery są lekkie i współdzielą jądro systemu operacyjnego hosta, co skutkuje minimalnym narzutem [3.15]. Wydajność ta pozwala na wdrożenie wielu kontenerów na jednym hoście, maksymalizując wykorzystanie zasobów i umożliwiając efektywne skalowanie. Docker promuje również architekturę modułową i opartą na mikrousługach. Konteneryzując poszczególne komponenty aplikacji, deweloperzy mogą oddzielać problemy i skalować określone części niezależnie.

3.6.6. Android/Kotlin

Rozwój Androida i Kotlina zmienia krajobraz tworzenia aplikacji mobilnych. Android, jako system operacyjny opracowany przez Google i oparty o Linuxa, zapewnia wszechstronną platformę do tworzenia aplikacji na szeroką gamę urządzeń takich jak smartfony, tablety, telewizory i systemy wbudowane. Natomiast Kotlin to nowoczesny język programowania, pozwalający na zwiększenie produktywności i czytelności kodu. Od czasu zadeklarowania przez Google oficjal-



Rysunek 3.15: Architektura platformy Docker

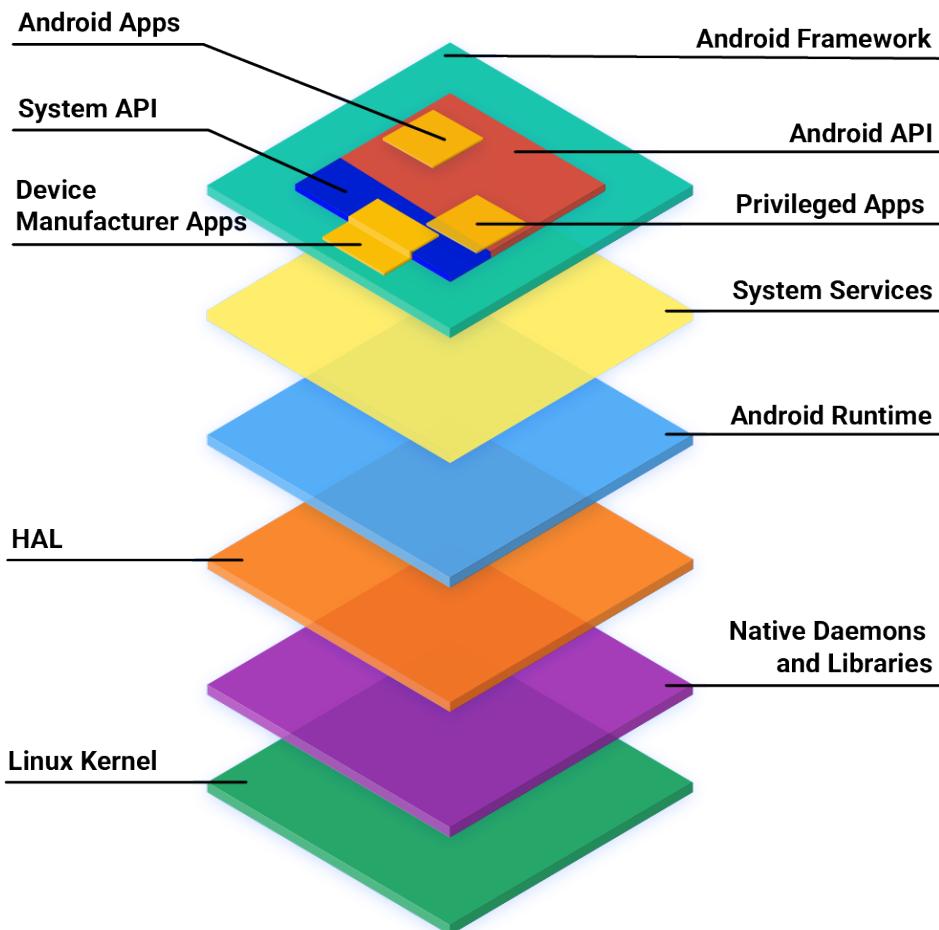
nego wsparcia w 2017 roku zdobył olbrzymią popularność wśród deweloperów na całym świecie.

Android oferuje kompleksowy zestaw narzędzi programistycznych (SDK) zawierający wiele bibliotek i interfejsów API pozwalających na wykorzystanie pełnego potencjału platformy [3.16]. Od budowania interfejsów użytkownika za pomocą XML i projektowania układów za pomocą potężnego Android Studio IDE, po dostęp do funkcji sprzętowych urządzenia, takich jak kamera, czujniki i GPS, Android zapewnia zasoby do dostosowywania aplikacji do dowolnych potrzeb.

Innym aspektem Androida jest jego bogaty ekosystem aplikacji. Sklep Google Play, oficjalny rynek aplikacji na Androida, zawiera miliony pozycji zaspakających różnorodne potrzeby użytkowników. Od sieci społecznościowych i narzędzi zwiększających produktywność po gry i wiele więcej, użytkownicy Androida mają dostęp do szerokiej gamy programów. Ta różnorodność stwarza deweloperom ogromne możliwości dotarcia do globalnej publiczności, zarabiania na swoich aplikacjach i przyczyniania się do rozwoju systemu.

Co więcej, otwarty charakter Androida zachęca do innowacji i współpracy. Platforma zapewnia deweloperom swobodę dostosowywania doświadczenia użytkownika, modyfikację komponentów systemu, a nawet tworzenia własnych ROM-ów. Ta otwartość sprzyja rozwojowi społeczności, w której to użytkownicy dzielą się wiedzą i tworzą własne modyfikacje systemu (często dając starym urządzeniom nowe życie). W ostatnich latach Android rozszerzył też swoje wsparcie dla nowych typów urządzeń, takich jak IoT, VR i AR.

Integracja Kotlin z Androidem zmieniła sposób tworzenia aplikacji mobilnych. Dzięki zwięzlej składni zwięzlej składni i rozbudowanym funkcjom język ten umożliwia programistom na pisanie czystego, wydajnego i łatwego w utrzymaniu kodu. Eliminuje tzw. "boilerplate code" i oferuje wiele mechanizmów abstrakcji, takich jak klasy danych, funkcje rozszerzające oraz wyrażenia



Rysunek 3.16: Architektura platformy Android

lambda, które upraszczają pracę i poprawiają organizację kodu. Kolejną kluczową zaletą Kotlina jest jego interoperacyjność z językiem Java. Płynnie integruje się on z istniejącymi projektami javaowymi, ułatwiając migrację i umożliwiając wykorzystanie istniejących bibliotek i framework'ów. Ta kompatybilność sprawia, że doświadczeni programiści Javy są w stanie bez większych trudności zacząć pracę z Kotlinem oraz pozwala na wykorzystanie swojego doświadczenia i wiedzy w tworzeniu aplikacji na Androida. Dostępny jest też bogaty ekosystem narzędzi, takich jak Android Studio, IDE stworzone specjalnie do tworzenia aplikacji androidowych.

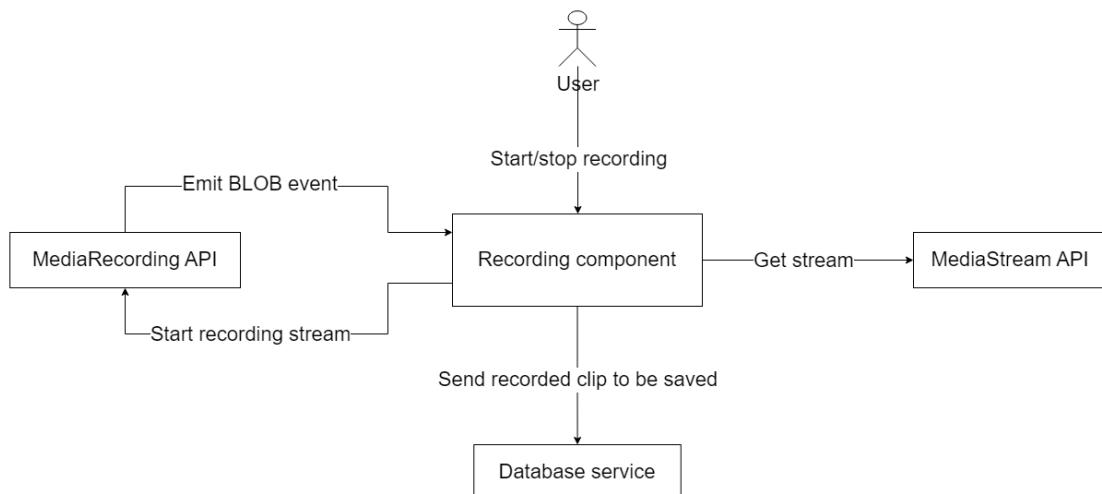
Istnieją również cross-platformowe rozwiązania, takie jak React czy Flutter, do tworzenia aplikacji mobilnych, ale pisanie aplikacji natywnych ma wiele zalet. Pozwala osiągnąć maksymalną wydajność wykorzystując pełną moc sprzętową. Dają programistom największe możliwości optymalizacyjne dla zapewnienia jak najbardziej płynnego działania. Mają bezpośredni dostęp do wszystkich możliwości urządzenia, funkcji i zasobów sprzętowych. Tworzenie natywnych aplikacji

zapewnia też większą integrację z ekosystemem androidowym. Narzędzia, takie jak Android Studio, zapewniają kompleksowe wsparcie i wiele dodatkowych funkcji, takich jak debuggery, emulatory i możliwości profilowania. Kolejną istotną zaletą jest zwiększoną kompatybilność. Tworzenie aplikacji korzystając z oficjalnych narzędzi wspieranych i promowanych przez Google minimalizuje ryzyko wystąpienia w przyszłości problemów ze starszymi wersjami aplikacji.

3.7. Stworzone komponenty

Jednym z kluczowych funkcjonalności wymaganych w projekcie było nagrywanie rozmów do późniejszego odtworzenia. Niestety nie znalezione zostały żadne dostępne open-sourcowe rozwiązania dla tego problemu. W związku z tym stworzony zostało komponent pozwalający na realizację tego zadania.

Źródło danych wybierane jest po nawiązaniu połączenia z drugim rozmówcą. Wszystkie ścieżki z nowego połączenia które posiadają obsługiwany typ MIME, a więc nadają się do nagrywania, są dodawane do strumienia mediów. API WebRTC udostępnia funkcję pozwalającą na rozpoczęcie i zakończenie emitowania specjalnych zdarzeń [29]. W nich zawarte są cząstki wideo z przypisanego strumienia od czasu ostatniego zdarzenia w postaci plików blob. Funkcja ta wywoływana jest po rozpoczęciu rozmowy i wyłączeniu przez użytkownika nagrywania. Odpowiedni serwis zaczyna wtedy czekać na zdarzenia a następnie pobiera i zapisuje emitowane dane do tablicy. Są tam one przechowywane aż do zakończenia rozmowy lub nagrywania. Istnieje również możliwość chwilowego wstrzymania nagrywania. Jeśli użytkownik zdecyduje się na zapis nagrania serwis łączy wszystkie elementy w pojedynczy plik blob a następnie wysyła go na serwer, gdzie przechowywany on jest w bazie MinIO [3.17]. Przykładowa aplikacja wykorzystująca ten komponent dostępna jest w repozytorium pod adresem <https://github.com/LeAndrejos/call-server>. Zawiera ona wszystkie elementy potrzebne do uruchomienia i demonstracji działania komponentu: serwer sygnalizujący WebRTC oraz podstawową stronę do wideorozmów wraz z funkcją nagrywania.



Rysunek 3.17: Schemat działania komponentu nagrywającego spotkania

Proces rozpoczęcia nagrywania prezentuje się następująco:

```
this.meetingRecorder = new MeetingRecorder(this.callService.remoteStream);  
this.meetingRecorder.startRecording();
```

Przekazany zostaje strumień wideo, który ma zostać nagrany, w tym wypadku strumień przychodzący od drugiego rozmówcy.

Nagrywanie można również wstrzymywać i następnie wznowiać. Odpowiedzialne są za to poniższe funkcje:

```
this.meetingRecorder.pauseRecording();  
this.meetingRecorder.resumeRecording();
```

Po zatrzymaniu nagrywania należy pobrać z komponentu gotowy obiekt File zawierający całe nagranie:

```
this.meetingRecorder.stopRecording();  
const file = this.meetingRecorder.getVideoFile('sample-name');
```

4. REZULTATY

4.1. Wytwarzanie projektu

Projekt był początkowo oparty na podstawie wymagań zebranych w formie wywiadu od klienta Modena Molti Mondi. Dalszy rozwój aplikacji oparty był o eksperymenty. Klientowi dostarczane były nowe funkcjonalności, a następnie zbierano opinie i uwagi które uwzględniane były w kolejnych wersjach. Praca odbywała się w luźnym oparciu o Scrum. Jest to zwinna metodologia zarządzania projektami używana głównie przy wytwarzaniu oprogramowania. Kładzie nacisk na współpracę, elastyczność i iteracyjny rozwój w celu dostarczania wartości klientom. Scrum dzieli pracę na backlog, który jest uszeregowaną pod względem ważności listą funkcji lub zadań. Podczas spotkania poświęconego planowaniu sprintu, zespół wybiera zestaw elementów z backlogu, które zobowiązuje się ukończyć w trakcie sprintu. Elementy te są następnie przenoszone do rejestru sprintu który służy jako przewodnik dla pracy zespołu. Pod koniec każdego sprintu zespół organizuje spotkanie podsumowujące sprint, aby zademonstrować ukończoną pracę interesariuszom i zebrać informacje zwrotne. Informacje zwrotne mają kluczowe znaczenie dla ciągłego udoskonalania i ulepszania produktu lub projektu. Dodatkowo, zespół organizuje spotkanie retrospektywne sprintu, aby zastanowić się nad swoimi procesami, zidentyfikować, co działało dobrze i omówić potencjalne obszary wymagające poprawy.

Praca odbywała się w formie około miesięcznych/dwumiesięcznych sprintów na początku których zbierane były informacje zwrotne od użytkowników i ich pomysły na dalszy rozwój. Na podstawie tych informacji tworzone były nowe funkcjonalności. Spotkania planowano co kilka tygodni by przedstawić postęp pracy wraz z końcowym wdrożeniem i dostarczeniem produktu do klienta. Dzięki temu aplikacja mogła być dostosowana do wymagań, które zidentyfikowano w późniejszym czasie przy korzystaniu z aplikacji. Przykładem takiej funkcjonalności jest moduł "Video Content".

Pewien problem sprawiła specyfikacja WebRTC, która do poprawnego działania wymaga połączenia HTTPS. Jest to zrozumiałe z powodów bezpieczeństwa, jednak znacznie utrudnia testowanie i zmusza do wystawienia lokalnego certyfikatu TLS.

4.2. Porównanie istniejących rozwiązań ze stworzoną aplikacją

Aplikacja została wykonana zgodnie z założeniami i spełnia wszystkie wymagania postawione przy jej projektowaniu. Jedna z najważniejszych funkcji, jaką są wideokonferencje, potencjalnie ma nieograniczone możliwości w kontekście ilości obsługiwanych połączeń. Zostało to osiągnięte dzięki skorzystaniu z technologii WebRTC, która umożliwia serwerowi bycie jedynie punktem identyfikacji dla obu klientów, po czym sama rozmowa odbywa się już bez jej udziału. Dzięki temu nawet jeśli z różnych powodów serwer przestałby reagować wszystkie rozpoczęte wideokonferencje mogłyby toczyć się dalej. Dzięki temu jest on również zdecydowanie odciążony obliczeniowo i nie odpowiada za transmisję video wszystkich połączeń, która byłaby bardzo kosztowna i wymagała wykorzystania maszyn o zdecydowanie większej mocy do obsługi tej sa-

mej liczby klientów.

Aplikacja może też zostać rozwinięta o wiele potencjalnych funkcjonalności jeszcze bardziej integrujących narzędzia muzyczne w jednym wygodnym miejscu. Jedną z nich jest metro-nom, stosunkowo proste, ale ułatwiające pracę przy braku fizycznego odpowiednika narzędzie, jak również zdalny edytor pięciolinii z możliwością adnotacji. Aktualnie aplikacja posiada możliwość udostępniania materiałów w różnych formatach jednak nie daje możliwości ich bezpośredniej edycji. Dodanie takiej funkcjonalności z pewnością okazałoby się przydatne zarówno dla nauczycieli jak i uczniów oraz pozwoliło im na pracę z utworami w dowolnym miejscu bez potrzeby noszenia papieru i długopisu. Wzorem dla takiego rozwiązania może być aplikacja Sibelius. Kolejnym potencjalnym rozwińciem może być wbudowany edytor nagrań umożliwiający przycinanie i łączenie nadesłanych plików. Dawałby on możliwość zsynchronizowania i odsłuchania wszystkich instrumentów tworzących dany utwór bezpośrednio w aplikacji i tworzenia różnych wersji z różnymi wykonawcami na danych instrumentach. Możliwa jest implementacja struktury drzewa w którym dla każdego utworu korzeniem byłby jego metadane a gałęziami i liśćmi kolejne kombinacje podkładów tworzących daną kompozycję [30].

Przegląd aplikacji dostępnych na rynku ujawnił, że pomimo wielu rozbudowanych rozwiązań żadne nie spełnia wszystkich wymagań stawianych przez zdalne nauczanie muzyki. Przy wszystkich z nich konieczne jest korzystanie z dodatkowych narzędzi by osiągnąć zamierzony efekt. Widać, że nastawione są one na szersze grono odbiorców. Posiadają możliwość integracji z różnymi zewnętrznymi aplikacjami, gdzie najczęściej możliwości w tym zakresie (ponad 1800 pozycji) zapewnia Microsoft Teams. Zoom posiada ich około 200. Discord jako platforma nastawiona na graczy posiada bardzo mało rozszerzeń o takim charakterze. Najlepszym wyborem pod względem czysto pedagogicznym jest Moodle które są platformą od początku tworzoną w celu nauczania. Nie jest ona jednak całkowicie wolna od wad. Nie posiada własnej funkcji video-rozmów, konieczne jest więc skorzystanie z innej platformy, która może być zintegrowana z Moodle'm czy to w formie pluginu czy też jako prosty link. Instalacja, dostosowanie platformy i zarządzanie nią również są w tym przypadku dużo bardziej trudniejsze i wymagają pewnej wiedzy technicznej nawet gdy używany jest jeden z hostingów Moodle'a. Wszystko to przekłada się na większe koszta użytkowania. Platformy oferujące rozwiązania typu plug & play niosą ze sobą zdecydowanie wyższe koszty niż Moodle, który będąc otwarto-źródłowym projektem na licencji GNU może być całkowicie bezpłatny. Stworzona aplikacja pod tym względem jest podobna do Moodle. Dostępny i otwarty kod umożliwia dowolne użycie we własnym środowisku, ale niezbędne są do tego podstawowe umiejętności techniczne, wersja plug & play nie istnieje.

Również w kwestiach bezpieczeństwa najlepszym wyborem jest Teams. Microsoft posiada duże doświadczenie w budowaniu komunikatorów dla firm o wysokich standardach zabezpieczeń przez tworzenie Skype for Business. Teams czerpie z tej aplikacji wiele kluczowych komponentów co czyni ją równie bezpieczną. Discord również jest uważany bezpieczną platformą, nie posiada jednak integracji z takimi funkcjami biznesowymi jak na przykład SSO, natomiast Zoom posiada reputację jako mało bezpieczna platforma. Na początku 2020 roku wykryto w niej wiele podatności, m.in.: możliwość dołączania do prywatnych konferencji bez zaproszenia czy przesyłanie danych, które według specyfikacji powinny być szyfrowane, w formie niezaszyfrowanej. Firma

twierdzi, że wszystkie problemy z bezpieczeństwem zostały już rozwiązane, jednak nadal znaleźć można listę problemów [31]. Stworzona aplikacja korzysta z tokenów JWT przesyłanych przez HTTPS dla zapewnienia bezpieczeństwa. Jest to rozwiązanie stosunkowo bezpieczne, jednak może być podatne na ataki CSRF. Stopień bezpieczeństwa jest niewątpliwie niższy niż w innych produktach, projektowanych i testowanych pod rygorystyczne standardy wymagane przez klientów biznesowych.

MusicHub, przez to że jest aplikacją hostowaną lokalnie, a nie w chmurze, mniej ogranicza użytkowników w kwestiach ilościowych. Jedyne limity do liczby kursów czy nagrani to możliwości zainstalowanego sprzętu, podobnie jak w Moodle. Wymagania minimalne nie są jednak wygórowane. Dla Moodle jest to 512MB RAMu oraz procesor 1GHz (zalecany dwurdzeniowy) oraz min. 200MB na samą aplikację plus miejsce na dane użytkowników. W przypadku MusicHub jest to ok. 128MB RAMu, procesor 2GHz dwurdzeniowy oraz min. 100MB na aplikację plus i dodatkowe miejsca na materiały.

Minusem MusicHub w porównaniu z innymi testowanymi aplikacjami jest liczba możliwych uczestników spotkania. Zoom oraz Teams już w wersjach darmowych pozwalają na 100 uczestników [32] [33], a wersjach płatnych ta liczba dochodzi do tysiąca [32]. Moodle nie posiada wbudowanych video konferencji, więc przy konfiguracji użytkownik może sam zdecydować jakiej liczby użytkowników się spodziewa i na tej podstawie wybrać zewnętrzny produkt. Ma to swoje zalety i wady, gdyż jest najbardziej wszechstronne, jednak wymaga od użytkownika więcej pracy, konfiguracji i dodaje potencjalne problemy i koszty. MusicHub zapewnia jedynie rozmowy dwóch użytkowników, jest to jednak kolejny aspekt, który może zostać poprawiony przy dalszym rozwoju aplikacji. WebRTC pozwala na połączenia wielu użytkowników, więc jest to kwestia kilku modyfikacji aplikacji i serwera sygnalizującego. Funkcjonalność czatu jest w podobnej sytuacji. Aktualnie istnieje jedynie możliwość czatu pomiędzy dwoma osobami, bez możliwości wysyłania załączników. Użyte biblioteki posiadają takie opcje, jest to więc kolejny aspekt pod którym projekt może być rozwijany. Zarówno Teams, Zoom jak i Moodle wspierają wieloosobowy czat (Zoom i Teams z takimi limitami jak przy video konferencjach), jednak Moodle nie wspiera domyślnie przesyłania w czacie załączników, taka funkcjonalność jest osiągalna jedynie przez zainstalowanie odpowiednich wtyczek.

Z punktu widzenia maszyn klienckich dużo zależy od wybranej wersji aplikacji. Przy wyborze programów desktopowych, tam gdzie są one dostępne (Teams oraz Zoom), wymagania sprzętowe są wielokrotnie wyższe niż wersje przeglądarkowe (4GB/128MB). Aplikacje webowe mają jednak pewne, choć nieliczne, ograniczenia. Wersje desktopowe często korzystają z akceleracji sprzętowej przy użyciu kart graficznych, posiadanie takowej może więc wpływać na dużo płynniejszą pracę. Również maksymalna rozdzielcość i niektóre integracje z innymi aplikacjami mogą być określone na stronach. Moodle oraz MusicHub dostępne są jedynie w formie aplikacji webowych, więc tam przez stronę można uzyskać dostęp do wszystkich funkcjonalności oferowanych przez platformę. Wymagania sprzętowe są dla przeglądarek bardzo niskie, pozwala to więc na dostęp do nich szerszemu gronowi odbiorców. Jednym elementem, który wymaga większej mocy obliczeniowej jest WebRTC, jednak nadal są to znakomite ilości.

5. PODSUMOWANIE

W bieżącym, niezwykle dynamicznym kontekście szybkiego postępu technologicznego oraz ewolucji w sposobie przekazywania wiedzy, zdalne nauczanie stało się niezmiennie obecne jako kluczowy element współczesnych narzędzi edukacyjnych. Analiza dostępnych na rynku platform ukazuje fascynujące spektrum możliwości, które umożliwiają naukę i współpracę na odległość. Niemniej jednak, analiza ta także ukazuje wyraźne braki, szczególnie w kontekście prowadzenia zajęć muzycznych. Większość istniejących platform edukacyjnych nie uwzględnia potrzeb związanych z nauczaniem muzyki i nie zapewnia odpowiednich narzędzi do skutecznego prowadzenia takich zajęć.

Choć istnieją pewne możliwości rozszerzenia funkcjonalności poprzez wtyczki, to dostępność takich rozszerzeń jest często ograniczona lub wymaga samodzielnego tworzenia. W tym kontekście, stworzenie dedykowanego narzędzia, które skupia się na potrzebach nauczycieli i uczniów w dziedzinie nauki muzycznej, jawi się jako odpowiedź na wyzwania wynikające z braku dedykowanych rozwiązań na rynku.

Niemniej jednak, obserwując trendy i kierunki rozwoju w dziedzinie edukacyjnej, można przypuszczać, że sytuacja ta zacznie się zmieniać. Rosnące zapotrzebowanie na specjalistyczne oprogramowanie do zdalnej nauki oraz dynamiczny rozwój projektów edukacyjnych skierowanych do konkretnych grup odbiorców, sugeruje, że w przyszłości powstaną narzędzia, które dokładnie odpowiadają specyficzny wymaganiom dziedziny, jaką jest nauka muzyczna.

Projekt stworzonej aplikacji stanowi istotny krok w kierunku wypełnienia tej luki. Dzięki zintegrowaniu wielu funkcjonalności, które brakują lub są niewystarczające w istniejących platformach, aplikacja dostarcza narzędzi do efektywnego prowadzenia lekcji muzycznych na odległość. Kursy, video-rozmowy z opcją nagrywania, a także zintegrowana opcja tworzenia i przesyłania nagrań, tworzą spójne i wszechstronne środowisko edukacyjne, które może dostosować się do różnorodnych potrzeb nauczycieli i uczniów.

Warto podkreślić, że projekt jest otwarto-źródłowy, co umożliwia implementację zaproponowanych rozwiązań przez szerokie grono deweloperów. To otwiera perspektywę na dalszy rozwój i doskonalenie platformy, aby sprostać rosnącym potrzebom edukacyjnym i zapewnić lepsze narzędzia do zdalnego nauczania muzycznego.

WYKAZ LITERATURY

- [1] Fernando Ferro, Patrizia Grifoni, and Tiziana Guzzo. Online learning and emergency remote teaching: Opportunities and challenges in emergency situations. *Societies*, 4, 2020.
- [2] Tenzin Doleck David John Lemay, Paul Bazelais. Transition to online learning during the covid-19 pandemic. *Computers in Human Behavior Reports*, 4, 2021.
- [3] Demei Shen, Moon-Heum Cho, Chia-Lin Tsai, and Rose Marra. Unpacking online learning experiences: Online learning self-efficacy and learning satisfaction. *Internet and Higher Education*, 19, 2013.
- [4] Liyan Song, Ernise S. Singleton, Janette R. Hill, and Myung Hwa Koh. Improving online learning: Student perceptions of useful and challenging characteristics. *Internet and Higher Education*, 7, 2004.
- [5] Ozlem Omur and Omer Bilgehan Sonsel. Covid-19 and digital literacy: Assessing pre-service music teachers' views on piano lessons provided in emergency remote teaching. *International Journal of Education Literacy Studies*, 9, 2021.
- [6] Ryan D. Shaw and Whitney Mayo. Music education and distance learning during covid-19: a survey. *Arts Education Policy Review*, 123, 2022.
- [7] Sam Duffy and Patrick Healey. A new medium for remote music tuition. *Journal of Music, Technology Education*, 10, 2017.
- [8] D. Garrison and N. Vaughan. Blended learning in higher education: Framework, principles, and guidelines. Jossey-Bass, 2008.
- [9] P. Grifoni T. Guzzo and F. Ferri. Blended Learning Environments for Adults: Evaluations and Frameworks. IGI Global, 2012.
- [10] Balu N. Ilag. Introduction: Microsoft Teams. Apress, 2018.
- [11] Paul Wagenseil. Zoom security issues. <https://www.tomsguide.com/news/zoom-security-privacy-woes>, 2022. [Online; accessed 2022-05-18].
- [12] Maya Yang. Zoom class-action lawsuit. <https://www.theguardian.com/technology/2022/apr/22/zoom-settlement-zoombombing-lawsuit>, 2022. [Online; accessed 2022-05-18].
- [13] Discord. What is Discord? <https://discord.com/safety/360044149331-What-is-Discord>, 2022. [Online; accessed 2022-05-18].
- [14] Jay Peters. Discord is making it easier to find interesting social audio rooms. <https://www.theverge.com/2021/5/13/22431438/discord-stage-channel-discovery-social-audio-threads-logo-clyde>, 2021. [Online; accessed 2023-05-16].

- [15] Discord team. API Docs for Bots and Developers. <https://discord.com/developers/docs/intro>, 2020. [Online; accessed 2023-05-16].
- [16] Jozsef Vass. How discord handles two and a half million concurrent voice users using WebRTC. <https://discord.com/blog/how-discord-handles-two-and-half-million-concurrent-voice-users-using-webrtc>, 2018. [Online; accessed 2023-05-16].
- [17] Ajlan Al-Ajlan and Hussein Zedan. Why moodle. 2008 12th IEEE International Workshop on Future Trends of Distributed Computing Systems, 2008.
- [18] Moodle statistics. <https://stats.moodle.org/>, 2023. [Online; accessed 2023-05-25].
- [19] MDN. WebRTC Description. https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API, 2022. [Online; accessed 2022-05-17].
- [20] Dragan Samardzija Branislav Sredojev and Dragan Posarac. Webrtc technology overview and signaling solution design and implementation. 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015.
- [21] MDN. WebRTC Protocols. https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API, 2022. [Online; accessed 2022-05-17].
- [22] O. Balci K. Schutt. Cloud software development platforms: A comparative overview. 2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA), 2016.
- [23] S. Aggarwal. Modern web-development using reactjs. International Journal of Recent Research Aspects, 2018.
- [24] E. Wohlgethan. Supporting web development decisions by comparing three major javascript frameworks: Angular, react and vue.js. 2018.
- [25] Amazon S3 documentation. <https://docs.aws.amazon.com/s3/index.html>, 2023. [Online; accessed 2023-08-13].
- [26] Ceph documentation. <https://docs.ceph.com/en/latest/>, 2023. [Online; accessed 2023-08-13].
- [27] A. Vannahme S. Juba and A. Volkov. Learning PostgreSQL. Packt, 2015.
- [28] MinIO Library. <https://resources.min.io/l/library>.
- [29] MDN. MediaStream recording API. https://developer.mozilla.org/en-US/docs/Web/API/MediaStream_Recording_API, 2023. [Online; accessed 2023-03-18].
- [30] Dionysios Politis Dimitrios Margounakis and Panagiotis Linardis. Towards a network – oriented system for cooperative music composition. Proceedings of the 7th WSEAS International Conference on Artificial intelligence, knowledge engineering and data bases, 2008.

- [31] Anthony Spadafora. Zoom security issues: What's gone wrong and what's been fixed. <https://www.tomsguide.com/news/zoom-security-privacy-woes>, 2023. [Online; accessed 2023-05-25].
- [32] Limits and specifications for Microsoft Teams. <https://learn.microsoft.com/en-us/microsoftteams/limits-specifications-teams>, 2023. [Online; accessed 2023-08-13].
- [33] Understanding meeting participant limit. <https://support.zoom.us/hc/en-us/articles/7492916830477-Understanding-meeting-participant-limit>, 2023. [Online; accessed 2023-08-13].

WYKAZ RYSUNKÓW

2.1	Propozycja układu pomieszczeń dla zdalnych lekcji muzyki	13
2.2	Interfejs platformy Teams	15
2.3	Interfejs platformy Zoom	16
2.4	Interfejs platformy Discord	18
2.5	Interfejs platformy Moodle	19
2.6	Interfejs aplikacji Sibelius	20
2.7	Schemat architektury WebRTC	22
3.1	Architektura aplikacji	30
3.2	Widok wszystkich kursów	31
3.3	Panel nagrywania	32
3.4	Sekcje i podsekcje	32
3.5	Spotkania	33
3.6	Członkowie kursu	33
3.7	Nauczyciele	33
3.8	Menu na urządzeniu mobilnym	34
3.9	Widok sekcji na urządzeniu mobilnym	35
3.10	Panel narzędzie administratora na urządzeniu mobilnym	36
3.11	Diagram przypadków użycia	37
3.12	Diagram framework'a Spring	39
3.13	Angular architecture	40
3.14	Poziomowanie danych	44
3.15	Architektura platformy Docker	45
3.16	Architektura platformy Android	46
3.17	Schemat działania komponentu nagrywającego spotkania	47

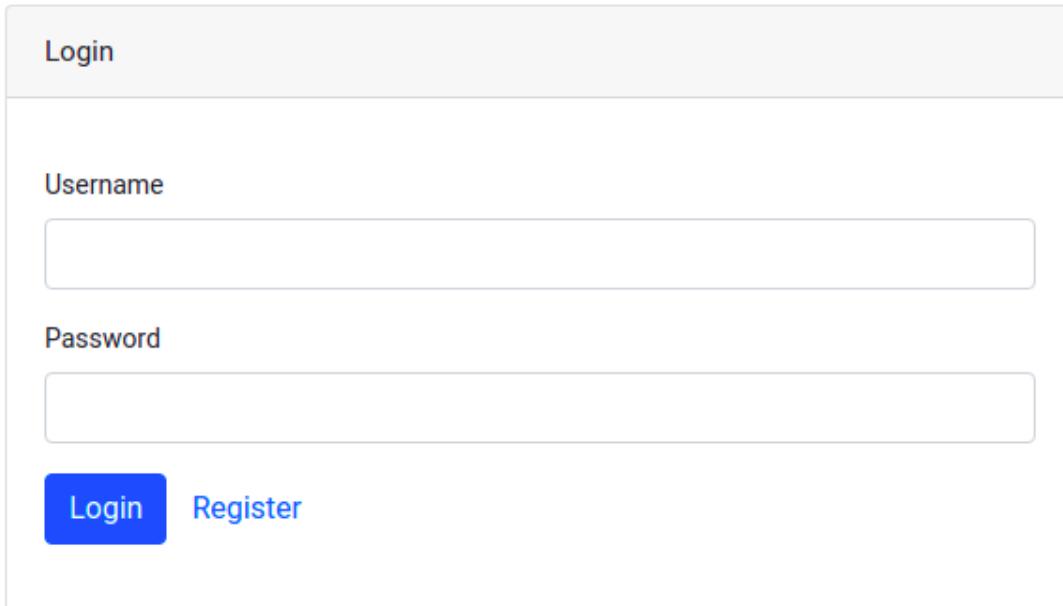
Wprowadzenie do instrukcji użytkownika

Poniżej zamieszczono instrukcję obsługi przygotowanej aplikacji - portalu MusicHub. Opisuje ona funkcjonalności dostępne dla poszczególnych grup użytkowników: uczniów, nauczycieli oraz administratorów. Tłumaczy również interfejs i przedstawia użytkownikowi przykładowe efekty poszczególnych działań. Została przygotowana, by ułatwić wdrożenie projektu u klienta - włoskiej orkiestry Modena Molti Mondi. Z tego też powodu, jak również by ułatwić korzystanie z niej innym zainteresowanym, całość instrukcji została spisana po angielsku.

MUSIC HUB USER GUIDE

1. Login and registration

Every user might create an account. Account has by default a student role. Teacher's account can be created only by the admin. When we enter <https://music-hub.site/> we will see the login page.



The image shows a login form with a light gray background. At the top left is the word "Login". Below it is a "Username" field represented by a white input box with a thin gray border. Underneath is a "Password" field, also a white input box with a thin gray border. At the bottom left is a blue rectangular button with the word "Login" in white. To its right is a blue link-like text "Register".

These are example credentials to admin accounts.

Username	Password
admin	password

Registration is as simple as login. Just enter credentials and your account with the student role will be created. After successful registration, you will be redirected to login screen with notification.

Registration successful



Login

Username

Password

[Login](#) [Register](#)

2. Creating courses and video content courses

Admins can create courses by going into the “Courses” tab in the navigation bar and clicking the “Create” button. Video content courses can similarly be created by going into the “Video Content” tab and clicking the “Create” button.

The screenshot shows the MODENA MULTIMONDI website. At the top, there is a dark header with the logo "MODENA MULTIMONDI ORCHESTRA INTERCULTURALE" and navigation links for "Home", "Courses", "Video Content", and "Logout". Below the header, the main content area has a dark background. In the center, there is a card-like element labeled "course" with a colorful geometric pattern. To the right of this card, there is a "Create" button with an upward-pointing arrow icon above it. At the bottom of the page, there is a dark footer bar with links for "Home", "Courses", and "Logout", and a user profile icon for "test_user1".

It will take them to a course creation page. Providing the course's name and description is required.

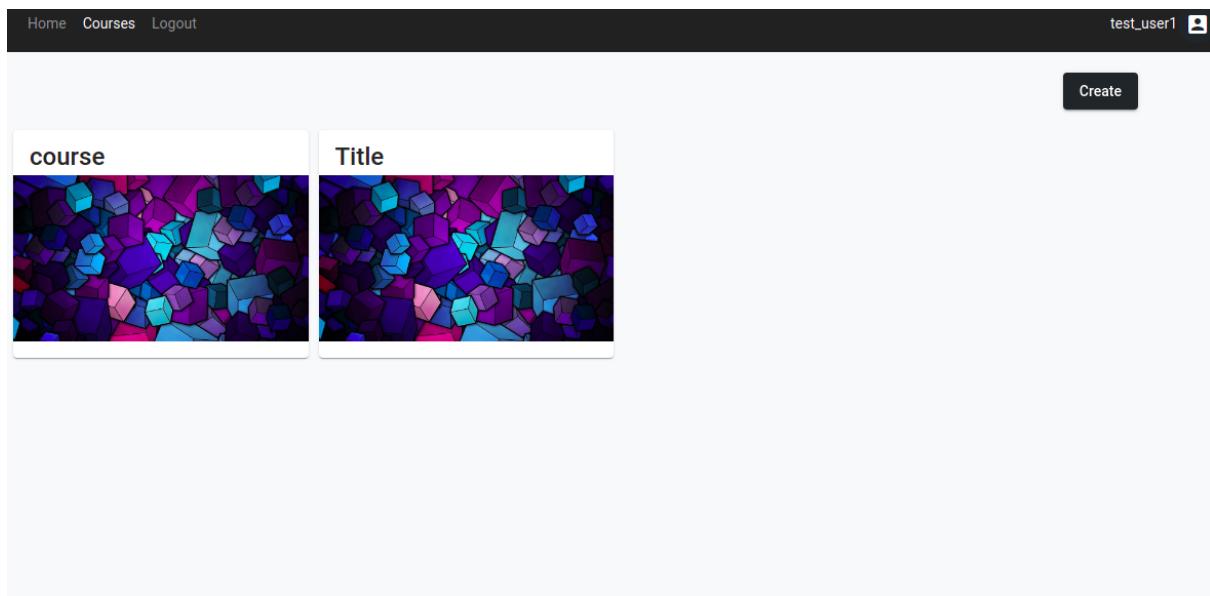
Create course

Title

Short Description

download.jpeg 

Clicking save creates the course and takes them back to the “Courses”/“Video Courses” page, where the newly created element will be already visible.

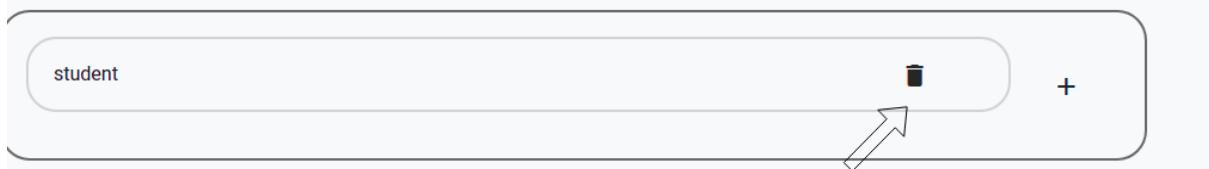


3. Adding participants to courses

Admins can add regular users to their courses by going to the desired course and scrolling to the “Participants” section.



A list of all added users is visible there, with the ability to remove them.



The “+” button shows the popup that allows the owner to add new users to the course.

Add a user to the course ×

piotr9

Add

A user must be selected and confirmed with the “Add” button. The user will be visible on the list immediately.

4. Adding sections

The “Add section” button in a course allows the owner to add a section to the course.

Sections

Add section

Name and description are required.

Add section

x

Name

Section nr 1

Description

Section Description

Add

Sections allow the owner to add materials and allow the students to send their recordings and materials.

Sections

Section nr 1

Section Description

Attachments

No attachments here.

+

Participants' attachments

No attachments here.

5. Adding attachments

The “+” button in the section card allows users to add their attachments. It shows a file selection popup. The name visible will be the original name of the file.

Section

Description

Attachments

+

Participants' attachments

Section nr 1

Section Description

Attachments

download.jpeg

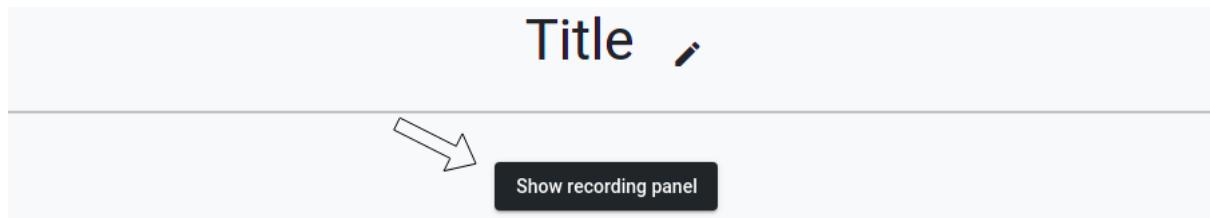


Participants' attachments

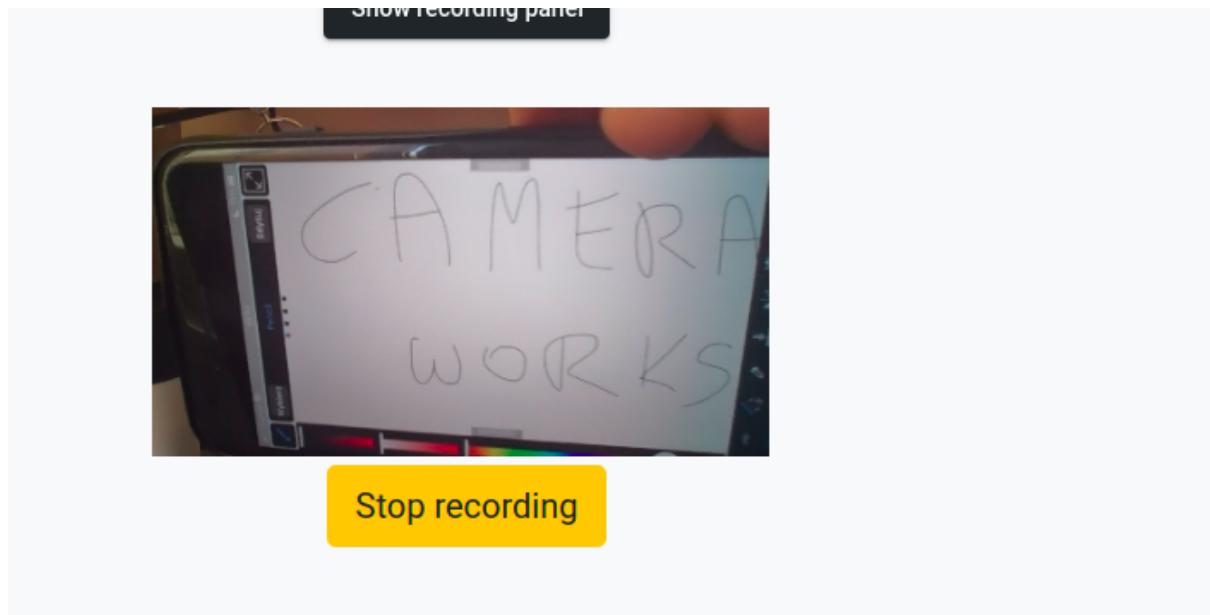
No attachments here.

6. Adding recordings

To add a recording one must first open the recording panel, with the button under the course's title.



There is a camera feed visible, to monitor the recording.



To

record/stop recording one must use the “Start recording”/“Stop recording” buttons.

When the recording is done the “Send recording” button allows the user to add the recording to a selected section.

Send your recording

X

Name of the recording

Record name

Select section

Section nr 1

Add

7. Adding meetings

To add a meeting one must click the “Add meeting” button.

Meetings



Add meeting

Schedule a meeting

X

Title



You must provide a title.

Date

 mm/dd/yyyy Calendar icon

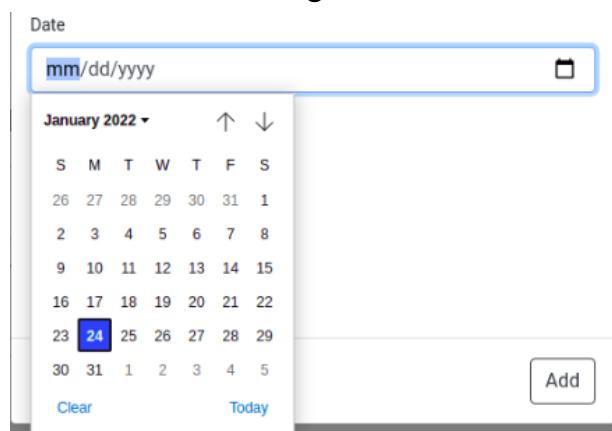
Time

Student

student

Add

After providing a name, date & time, and the user with whom the meeting will be, the meeting can be created with the “Add” button.



12:00



CANCEL

OK

Schedule a meeting

x

Title

Metting title

Date

01/11/2022



Time

12:00

Student

student

Add

8. Participating in meetings

Every user can see meetings they have planned in the course view. To start a meeting they must click on the button with the title of the desired meeting. That will take them to a call screen. The “Start call” button will initiate the call. When the other user initiates on their site the call will be started. The “Stop call” button stops the call.

Meetings

The screenshot shows a course interface with a "Meetings" section. At the top, there is a list of two meetings:

- Meeting 24.01 (Icon with a red circle), Date: 2022-01-24, Time: 20:00, Status: student
- Meeting 30.01 (Icon), Date: 2022-01-30, Time: 15:00, Status: student

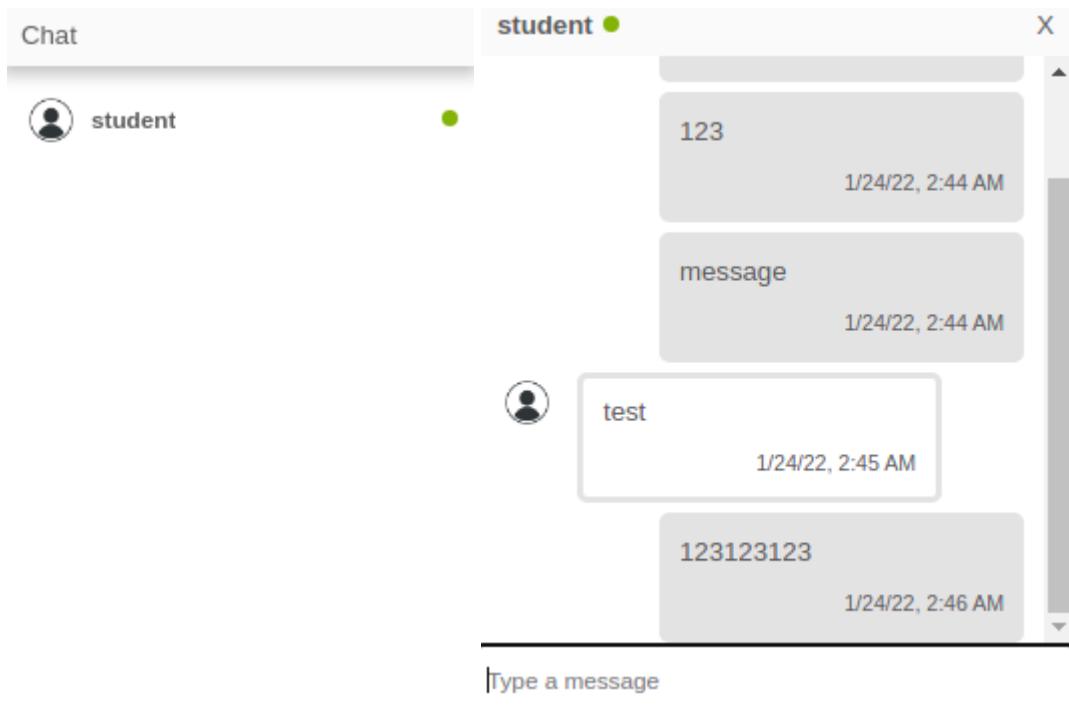
Below the list is a "Add meeting" button. The second meeting is expanded to show its details in a larger box:

Meeting 30.01 (Icon), Date: 2022-01-30, Time: 15:00, Status: student

At the bottom of this expanded view is a video player interface with a black background. It shows a play button, a progress bar at 0:00, and a "Start call" button.

9. Chat with course participants

Every user can chat with other course participants. When you enter the main course page, at bottom right there will be a small window with users that you can chat to. Just enter the conversation by clicking on user and write something.



10. Changing account data

Every user can change their username and password by clicking the upper right-hand corner.

Home Courses Logout test_user

test_course

Show recording panel

Sections

Add section

Meetings

Add meeting

Participants

student

In that panel you can change and save your account data.

Change account data

New username
test_user1

New password

Save

11. Admin tools

The admin account, besides from having access to all of the courses and materials on the site, has a suite of tools at their disposal.

Admin tools

Manage users

Username	Role	Delete	Reset Password
julian	STUDENT		Reset password
prova1	STUDENT		Reset password
8Lou46fb	TEACHER		Reset password

Reset password

Username: _____

Password: _____

Create new teacher

Username: _____

Password: _____

Create new teacher

It gives the ability to manage users, reset their passwords if they have forgotten it and create new teacher accounts which can be then distributed to the teachers.