



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Imię i nazwisko studenta: Maciej Zakrzewski

Nr albumu: 175573

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 tworzenia muzyki w modelu kooperacyjnym

Tytuł pracy w języku angielskim: System for composing music using cooperative approach

Opiekun pracy: dr hab. inż. Julian Szymański

STRESZCZENIE

Celem niniejszej pracy było zaprojektowanie, implementacja i zbadanie użyteczności systemu wspomagającego współpracę i edukację muzyczną na odległość, który nie wymaga od użytkowników specjalistycznego sprzętu oraz wysokich umiejętności technicznych. W ramach pracy przeprowadzono przegląd literatury i istniejących rozwiązań technicznych w dziedzinach pracy grupowej wspomaganej komputerowo oraz zdalnej współpracy i edukacji muzycznej. Na podstawie przeglądu zaproponowano koncepcję systemu, przeprowadzono analizę wymagań i określono architekturę rozwiązania. System został zaimplementowany jako progresywna aplikacja internetowa, która pozwala na tworzenie wirtualnych przestrzeni roboczych, w ramach których użytkownicy mogą współdzielić materiały cyfrowe, przeprowadzać wirtualne spotkania z możliwością wspólnej pracy nad utworami muzycznymi i ich odsłuchu oraz prowadzić dyskusję za pomocą forum. Zaimplementowany system został wdrożony i przetestowany pod kątem użyteczności. Przeprowadzone testy wskazały na użyteczność systemu w kontekście współpracy muzycznej i wskazały obszary, które wymagają dalszego rozwoju.

Słowa kluczowe: praca grupowa wspomagana komputerowo, oprogramowanie do pracy grupowej, zdalna współpraca muzyczna, zdalna edukacja muzyczna, aplikacje internetowe

Dziedzina nauki i techniki, zgodnie z wymogami OECD:

nauki o komputerach i informatyka

ABSTRACT

The purpose of this thesis was to design, implement, and examine the usability of a system supporting remote music collaboration and education that does not require users to have specialized equipment and high technical skills. The thesis included a review of the literature and existing technical solutions in the fields of computer-supported cooperative work, as well as remote music collaboration and education. Based on the review, a concept of the system was proposed, a requirements analysis was carried out and the solution's architecture was defined. The system was implemented as a progressive web-based application that allows the creation of virtual workspaces within which users can share digital materials, conduct virtual meetings with the ability to work on and listen to musical pieces together, as well as hold discussions via a forum. The implemented system was deployed and tested for usability. The tests carried out indicated the usability of the system in the context of music collaboration and identified areas that need further development.

Keywords: computer-supported cooperative work, groupware, remote music collaboration, remote music education, web applications

Field of science and technology in accordance with OECD requirements:
computer science and information technology

SPIS TREŚCI

SPIS TREŚCI	5
WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW	7
1. WSTĘP	8
1.1. Cel pracy	8
1.2. Zawartość pracy	9
2. PRZEGŁĄD LITERATURY I ISTNIEJĄCYCH ROZWIĄZAŃ	10
2.1. Praca grupowa wspomagana komputerowo	10
2.1.1. Macierz przestrzeni i czasu	10
2.1.2. Model MoCA	12
2.2. Edukacja zdalna	13
2.3. Zdalna edukacja i współpraca muzyczna	15
2.3.1. Sieciowe wykonywanie muzyki	16
2.3.2. Cyfrowe stacje robocze w pracy grupowej	17
2.4. Podsumowanie	18
3. PROJEKT, IMPLEMENTACJA I TESTY SYSTEMU	19
3.1. Analiza wymagań	19
3.1.1. Synchroniczność i rozkład fizyczny	19
3.1.2. Skala i liczba wspólnot praktyki	19
3.1.3. Nieugruntowanie, planowana stałość i obrót	20
3.1.4. Wymagania funkcjonalne	20
3.1.5. Wymagania niefunkcjonalne	20
3.1.6. Model domeny	21
3.2. Wysokopoziomowa architektura systemu	25
3.3. Wybór rozwiązań technicznych	27
3.3.1. Aplikacja kliencka	27
3.3.2. Serwer aplikacji	30
3.3.3. Dostawca tożsamości	32
3.3.4. Platforma wideokonferencyjna	35
3.3.5. Baza danych i magazyn plików	36
3.4. Implementacja	36
3.4.1. Dostawca tożsamości	36
3.4.2. Aplikacja kliencka	37
3.4.3. Serwer aplikacji	47
3.4.4. Platforma wideokonferencyjna	50
3.5. Wdrożenie systemu	51
3.6. Testy użytkowe	52
4. PODSUMOWANIE	55
4.1. Plany dalszego rozwoju systemu	55

WYKAZ LITERATURY	59
WYKAZ RYSUNKÓW	60
WYKAZ TABEL	61
Dodatek A: Instrukcja obsługi systemu	62

WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW

- | | |
|---------------|---|
| CSCW | – praca grupowa wspomagana komputerowo (ang. Computer-Supported Cooperative Work) |
| LMS | – system zarządzania nauką (ang. Learning Management System). |
| NMP | – sieciowe wykonywanie muzyki (ang. Networked Music Performance). |
| DAW | – cyfrowa stacja robocza (ang. Digital Audio Workstation). |
| API | – interfejs programowania aplikacji (ang. Application Programming Interface). |
| URL | – ujednolicony identyfikator zasobu (ang. Uniform Resource Locator). |
| HTML | – język znaczników do tworzenia dokumentów hipertekstowych (ang. HyperText Markup Language). |
| JSON | – format danych tekstowych w bazujący na języku programowania JavaScript (ang. JavaScript Object Notation). |
| JWT | – internetowy token w formacie JSON (ang. JSON Web Token). |
| WASM | – WebAssembly, format binarny dla aplikacji internetowych. |
| PWA | – progresywna aplikacja internetowa (ang. Progressive Web Application). |
| WebRTC | – otwarta platforma umożliwiająca komunikację w czasie rzeczywistym w przeglądarce internetowej (ang. Web Real-Time Communication). |
| IDP | – system zarządzający tożsamością użytkownika (ang. Identity Provider). |

1. WSTĘP

Dynamiczny rozwój technik komputerowych i sieciowych znaczco zmienił sposób komunikacji i współpracy międzyludzkiej zarówno w sferze prywatnej, edukacyjnej jak i zawodowej. Współczesne systemy pracy kooperacyjnej pozwalają na pracę zespołową i wymianę informacji niezależnie od czasu i lokalizacji użytkowników, dzięki czemu stają się niezastąpionymi narzędziami w wielu dziedzinach.

Edukacja na odległość w formie e-learningu, staje się coraz bardziej powszechną metodą dydaktyczną, która wspomaga, a w niektórych przypadkach całkowicie zastępuje tradycyjne formy nauczania. Systemy zarządzania nauką (ang. Learning Management Systems, LMS) pozwalają na dostarczanie treści dydaktycznych, monitorowanie oraz przeprowadzanie procesu edukacji w sposób elastyczny i zindywidualizowany. Dziedzina muzyki jest szczególnym wyzwaniem w kontekście edukacji i współpracy na odległość. Kształcenie muzyczne opiera się w dużej mierze na doskonaleniu umiejętności manualnych, wokalnych i słuchowych, co najczęściej wymaga bezpośrednich interakcji między nauczycielem a uczniem. Rozwiązania takie jak oprogramowanie wideokonferencyjne, choć są użyteczne w komunikacji i łatwe w obsłudze, nie spełniają wymagań dotyczących jakości i opóźnienia transmisji dźwięku, co w praktyce uniemożliwia zdalne wykonywanie utworów muzycznych w czasie rzeczywistym. Dedykowane oprogramowanie do sieciowego wykonywania muzyki (ang. Networked Music Performance, NMP) potrafi spełnić te wymagania, niestety wymaga często ponadprzeciętnych umiejętności komputerowych a bardziej wyrafinowane rozwiązania wiążą się z koniecznością posiadania specjalistycznego sprzętu i infrastruktury sieciowej. Cyfrowe stacje robocze (ang. Digital Audio Workstation, DAW) w formie aplikacji internetowych oferują alternatywne podejście do problemu, pozwalając na wspólną pracę nad utworami w formie asynchronicznej.

1.1. Cel pracy

Celem niniejszej pracy dyplomowej było stworzenie internetowego systemu wspierającego współpracę i edukację muzyczną. System ten ma umożliwić wspólną pracę nad utworami muzycznymi oraz zdальną naukę gry na instrumentach poprzez przyjazne użytkownikowi środowisko dostępne zarówno na komputerach stacjonarnych, jak i urządzeniach mobilnych.

Centralną funkcjonalnością systemu są wirtualne przestrzenie robocze, w których użytkownicy mogą przeprowadzać spotkania w formie wideokonferencji z możliwością częściowo synchronicznej pracy nad utworami muzycznymi. W trakcie spotkania dostępny jest współdzielony edytor audio, pozwalający na przyrostową pracę nad utworem i wspólny odsłuch rezultatów. W obrębie przestrzeni roboczej użytkownicy mogą współdzielić dowolne materiały cyfrowe niezbędne w procesie nauki i współpracy. Spotkania mogą zostać nagrane i automatycznie zarchiwizowane w przestrzeniach. System pozwala na organizowanie materiałów w katalogi oraz ich kategoryzację, ułatwiając użytkownikom znalezienie tego, czego potrzebują. Ponadto zapewnia mechanizm uprawnień dostępu do plików i katalogów, który ma na celu ochronę materiałów oraz ich strukturę przed modyfikacją, czy usunięciem przez nieuprawnionych użytkowników. Możliwa jest również komunikacja asynchroniczna pomiędzy członkami danej przestrzeni roboczej w formie forum dyskusyjnego z możliwością odniesienia się do zamieszczonych w niej materiałów.

1.2. Zawartość pracy

W ramach pracy została przeprowadzona analiza obecnego stanu wiedzy w dziedzinie zdalnej współpracy oraz edukacji muzycznej w formie przeglądu literatury. Na podstawie przeglądu zidentyfikowano najistotniejsze wymagania i problemy występujące w dziedzinie oraz wady i zalety istniejących narzędzi i rozwiązań technicznych. Następnie zaproponowano koncepcję systemu, który poprzez połączenie zalet istniejących rozwiązań ma na celu zaspokojenie najważniejszych potrzeb użytkowników i zminimalizowanie wpływu ograniczeń technicznych na proces współpracy i edukacji muzycznej.

Rozdział drugi zawiera przegląd literatury, w którym omówiono zagadnienia teoretyczne dotyczące współpracy i edukacji muzycznej z użyciem techniki komputerowej, a także opisano istniejące rozwiązania techniczne w tych dziedzinach, które posłużyły jako punkt wyjścia dla projektowanego systemu. Rozdział trzeci przedstawia projekt prototypu systemu, obejmujący analizę wymagań i dziedziny problemu, określenie architektury systemu, dobór narzędzi i rozwiązań technicznych oraz opis implementacji i wdrożenia systemu. Rozdział czwarty zawiera przedstawione przeprowadzone testy użytkowe systemu, mające na celu ocenę jego użyteczności. Praca jest zakończona podsumowaniem oraz propozcjami dalszego rozwoju powstałego systemu.

2. PRZEGŁĄD LITERATURY I ISTNIEJĄCYCH ROZWIĄZAŃ

W tej części pracy omówione zostały wybrane zagadnienia teoretyczne dotyczące pracy grupowej wspomaganej komputerowo, edukacji zdalnej oraz procesu tworzenia i nauczania muzyki w środowisku sieciowym. Przedstawiono również szereg istniejących narzędzi wykorzystywanych tych procesach.

2.1. Praca grupowa wspomagana komputerowo

Jedną z ram teoretycznych wykorzystywaną do zrozumienia w ogólności procesów współpracy z wykorzystaniem techniki jest praca grupowa wspomagana komputerowo (ang. Computer Supported Cooperative Work, skrót CSCW). Jest to interdyscyplinarna dziedzina nauki badająca sposoby wspierania współpracy międzyludzkiej poprzez zastosowanie techniki komputerowej [1]. Badania nad CSCW sięgają początku lat 80. XX wieku, wraz ze wzrostem popularności komputerów osobistych oraz rozwojem sieci komputerowych. Sam termin powstał w roku 1984 za sprawą Irene Greif oraz Paula Cashmana na potrzeby prowadzonych przez nich warsztatów. 2 lata później odbyła się pierwsza konferencja, a w roku 1992 zaczęto wydawać dziennik pod szyldem CSCW [2].

Najistotniejsze obszary badawcze, które składają się na CSCW to praca grupowa i procesy w niej zachodzące oraz technika ją wspierająca. Współpracę analizuje się w kontekście sposobów komunikacji międzyludzkiej, struktur wewnętrzgrupowych, planowania pracy oraz procesów podejmowania decyzji [1, 2, 3]. W przypadku techniki najistotniejszymi aspektami są sposoby zdalnej komunikacji, współdzielenia zasobów i informacji. W badaniach nad CSCW szczególną uwagę zwraca się na kwestie związane z interakcjami społecznymi oraz koordynacją pracy w grupie. Podczas analizy procesów współpracy istotne jest uwzględnienie aspektów kulturowych, organizacyjnych, ekonomicznych oraz socjotechnicznych, które mogą wpływać na efektywność i jakość pracy zespołowej. Z ideą systemów CSCW silnie związane jest również pojęcie oprogramowania do pracy grupowej (ang. groupware), czyli oprogramowania wspierającego pracę zespołową [4]. Obejmuje ono zarówno narzędzia komunikacyjne, takie jak czaty, wideokonferencje czy media społecznościowe, jak i bardziej zaawansowane rozwiązania, takie jak systemy zarządzania projektami, platformy do wspólnego tworzenia dokumentów czy oprogramowanie do organizacji pracy. Badania w dziedzinie CSCW mają również na celu identyfikację i analizę problemów wynikających ze stosowania techniki przy pracy grupowej, takich jak kwestie związane z prywatnością, bezpieczeństwem czy wprowadzaniem zmian w strukturach organizacyjnych.

Współpraca z wykorzystaniem techniki może obejmować różne kategorie działalności człowieka, takie jak edukacja, medycyna, nauka, inżynieria, biznes czy administracja. Dlatego też, oprócz badań nad techniką i rozwiązaniami specyficznych dla danej dziedziny, CSCW stawia na interdyscyplinarność i wymianę wiedzy pomiędzy różnymi sektorami. Tym samym przyczynia się do rozwoju efektywnych i innowacyjnych rozwiązań, które mogą wpływać na poprawę jakości współpracy w różnych obszarach. Z uwagi na swój szeroki zakres, dziedzina ta nieustannie ewoluje, obejmując najnowsze osiągnięcia nauk technicznych i społecznych.

2.1.1. Macierz przestrzeni i czasu

Jednym z najwcześniejszych podejść do modelowania systemów CSCW jest klasyfikacja rozwiązań technicznych w kontekście ich zastosowań w pracy zespołowej za pomocą dwóch kryteriów: czasu i przestrzeni [2, 4]. Klasyfikacja ta jest najczęściej przedstawiana w formie macierzy z osiami odpowiadającymi

nym kryteriom. Przykład takiej macierzy znajduje się na rysunku 2.1. Oś pozioma reprezentuje kryterium czasu i dzieli pracę grupową na synchroniczną i asynchroniczną. Przy pracy synchronicznej członkowie zespołu pracują razem w tym samym czasie, przy pracy asynchronicznej każdy może pracować w dowolnym momencie. Oś pionowa reprezentuje kryterium przestrzeni, a więc podział na współpracę lokalną i zdalną. Współpraca lokalna odbywa się w jednym miejscu, zdalna – w wielu miejscach. Na podstawie tych kryteriów określono cztery podstawowe formy współpracy:

- synchroniczna i lokalna,
- synchroniczna i zdalna,
- asynchroniczna i lokalna,
- asynchroniczna i zdalna.

Macierz CSCW, pomimo swojej prostoty jest pomocna w badaniach nad współpracą wspomaganą komputerowo. Klasyfikacja ta pozwala na zrozumienie podstawowych wymogów pracy zespołowej oraz na identyfikację odpowiednich narzędzi, które mogą ułatwić i ulepszyć komunikację oraz koordynację pracy w grupie. Może też być przydatny w procesie analizy istniejących rozwiązań i projektowania systemu wspierającego pracę grupową poprzez analizę potrzeb użytkowników w ujęciu form współpracy i dekompozycję systemu na części realizujące poszczególne formy [5].

Należy jednak pamiętać, że jest to tylko jedna z możliwych klasyfikacji, w praktyce granice między poszczególnymi formami mogą być płynne i zależne od kontekstu czynności wykonywanych przez zespół. Zadania realizowane przez grupę mogą wymagać współpracy w wielu formach. Wiele systemów CSCW łączy różne formy współpracy, oferując elastyczne rozwiązania, które mogą być dostosowywane do potrzeb zespołów i organizacji. Przykłady takich rozwiązań to platformy komunikacyjne integrujące czat, wideokonferencje i współdzielenie plików, czy systemy zarządzania projektami z funkcjami zarówno synchronicznymi, jak i asynchronicznymi. Z tego powodu większości współczesnych systemów CSCW nie można całościowo przypisać tylko do jednej z tych kategorii. Chociaż macierz CSCW jest użytecznym

		Czas	
		Synchroniczna	Asynchroniczna
Miejsce	Lokalna	Interakcje twarzą w twarz <ul style="list-style-type: none"> • narzędzia wspomagające grupowe podejmowanie decyzji, • systemy głosowania, • wyświetlacze • tablice interaktywne 	Zadania ciągłe w czasie <ul style="list-style-type: none"> • narzędzia wspomagające pracę zmianową, • publiczne wyświetlacze
	Zdalna	Interakcje zdalne <ul style="list-style-type: none"> • wideokonferencje, • współdzielenie ekranu, • komunikatory internetowe, • czaty, • edytory dla wielu użytkowników 	Koordynacja pracy <ul style="list-style-type: none"> • poczta elektroniczna, • kalendarz i harmonogramy elektroniczne, • fora, białetyki, wiki • systemy wersjonowania

Rysunek 2.1: Macierz klasyfikacji techniki wykorzystywanej w systemach CSCW według kryteriów przestrzeni i czasu [4].

narzędziem w badaniach i projektowaniu systemów wspierających współpracę, nie jest ona w pełni zgodna z rzeczywistością, ponieważ skupia się głównie na aspektach technicznych nie uwzględniając czynników społecznych takich jak struktura, dynamika, rozmiar czy charakter działań grupy. Czynniki te mogą wpływać na sposób, w jaki zespoły wykorzystują technikę i ostatecznie na efektywność pracy grupowej.

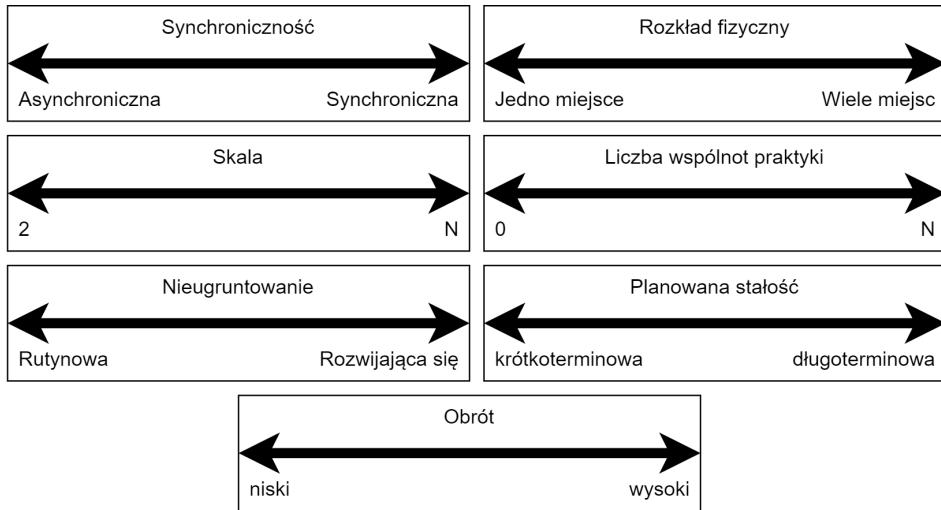
2.1.2. Model MoCA

Zakres badań CSCW zmieniał się wraz z rozwojem techniki, potrzebami i wzrostem różnorodności grup aktywnie ją wykorzystujących, a także pojawieniem się grup o znacznie większych rozmiarach i szerszych możliwościach. Jednym z nowszych podejść do systemów CSCW jest opublikowany w 2015 roku model koordynowanej akcji (ang. Model of Coordinated Action, skrót MoCA) [6], bazujący między innymi na opisanej wcześniej macierzy przestrzeni i czasu. Model ten rozkłada pracę wykonywaną w grupie na siedem ciągłych wymiarów. W przeciwieństwie do wcześniejszych podejść, które skupiały się głównie na aspektach technicznych, MoCA ma na celu opisanie tego procesu z bardziej holistycznego punktu widzenia, uwzględniając również aspekty społeczne [3, 5, 6]. Czynniki, które składają się na Model of Coordinated Action to [6]:

- Synchroniczność (ang. Synchronicity),
- Rozkład fizyczny (ang. Physical Distribution),
- Skala (ang. Scale),
- Liczba wspólnot praktyki (ang. Number of Communities of Practice),
- Niegruntowanie (ang. Nascence)
- Planowana stałość (ang. Planned permanence),
- Obrót (ang. Turnover).

Synchroniczność odnosi się do sposobu pracy grupy w kontekście czasu, określa w jakim stopniu członkowie grupy pracują w tym samym momencie. Rozkład fizyczny opisuje stopień rozproszenia fizycznego członków zespołu w trakcie pracy. Wymiary te są analogiczne do wymiarów z macierzy przestrzeni i czasu. Skala określa rozmiar grupy w ilości osób do niej należących. Wspólnota praktyki jest zbiorem osób zainteresowanych wspólną dziedziną, które wymieniają się wiedzą, umiejętnościami i doświadczeniem w jej zakresie w celu rozwinięcia swoich kompetencji. Ich ilość we współpracy grupie może zależeć nie tylko od ilości dziedzin, którymi się ona zajmuje, ale także od zróżnicowania kulturowego (np. normy społeczne lub języki) lub w poziomie zaawansowania. Wymiar ten opisuje grupy od przypadku, w którym nie istnieje żadna wspólnota praktyki (grupa współdzieli jedynie cel, ale żadna z osób nie posiada większego doświadczenia w dziedzinie), poprzez jednolite zespoły osób o podobnym poziomie doświadczenia i pochodzeniu, kończąc na zespołach mocno zróżnicowanych kulturowo lub pochodzących z wielu różnych środowisk tworzących własne wspólnoty praktyki. Niegruntowanie określa potencjał do zmian charakteru działań podejmowanych przez grupę. Charakterystyka działań może zmieniać się wraz ze zmianami w celach grupy, składzie jej członków, zmian w środowisku (np. na rynku konsumenckim lub pracy), w którym działa, czy też w technologii, której używa. Planowana stałość określa, jak długo grupa zamierza współpracować. Wymiar odnosi się do samych zamiarów grupy w tym zakresie ze względu na trudności w przewidywaniu rzeczywistej długości współpracy. Obrót określa, jak często grupa zmienia skład członków [5, 6].

Model koordynowanej akcji pozwala na bardziej precyzyjną analizę różnorodnych aspektów współpracy w grupie, uwzględniając nie tylko techniczne, ale także społeczne czynniki wpływające na skuteczność pracy zespołowej. Tego rodzaju podejście jest szczególnie istotne w kontekście rosnącej globalizacji



Rysunek 2.2: Wymiary modelu Model of Coordinated Action [6]

i złożoności współczesnych organizacji, które muszą się dostosować do szybkich zmian i dynamicznego otoczenia. W praktyce model ten może pomóc w projektowaniu bardziej efektywnych systemów CSCW poprzez analizę i uwzględnienie potrzeb oraz charakterystyki konkretnych grup użytkowników [5, 7]. Na przykład, w przypadku zespołów o dużej skali i dużej liczbie wspólnot praktyki, system CSCW powinien być w stanie skalować się wraz z rosnącą ilością użytkowników, a także zapewnić odpowiednie narzędzia do wspierania wymiany wiedzy i komunikacji między różnymi grupami specjalistów. W przypadku grup o wysokim nieugruntowaniu i planowanej stałości system powinien być elastyczny i łatwy w dostosowywaniu do zmieniających się potrzeb, celów oraz dostępnej techniki. MoCA może również służyć jako narzędzie diagnostyczne, umożliwiające analizę istniejących systemów CSCW oraz identyfikację obszarów, które wymagają poprawy lub dostosowania [5]. Dzięki uwzględnieniu szerszego spektrum czynników wpływających na współpracę model ten może pomóc organizacjom lepiej zrozumieć wyzwania i możliwości związane z wykorzystaniem techniki komputerowej w pracy zespołowej. Model ten stanowi podsumowanie wiedzy na temat współpracy wspomaganej komputerowo, wprowadzając bardziej całościowe podejście do analizy i projektowania systemów CSCW. W miarę rozwoju techniki, postępującej globalizacji oraz rosnącego zróżnicowania i złożoności współpracujących zespołów, takie podejście zdaje się kluczowe dla tworzenia innowacyjnych i efektywnych rozwiązań wspierających pracę zespołową.

2.2. *Edukacja zdalna*

W ciągu ostatnich lat można zauważać dynamiczny wzrost zainteresowania edukacją realizowaną w formie zdalnej. Klasyczny proces edukacji, który dotychczas opierał się na bezpośrednim kontakcie pomiędzy nauczycielem a uczniem, jest stopniowo rozszerzany o interakcje z różnego rodzaju platformami internetowymi a czasem nawet całkowicie do takiego środowiska przenoszony. Fenomen ten jest szczególnie widoczny w szkolnictwie wyższym, gdzie wraz z postępującą cyfryzacją coraz więcej uczelni przenosi materiały dydaktyczne na dedykowane do tego platformy internetowe, a także oferuje studia niestacjonarne w formie zdalnej [8]. Rosnące zapotrzebowanie na tego rodzaju rozwiązania sprawia, że na rynku pojawia się coraz więcej narzędzi służących do wspierania procesu edukacji w środowisku sieciowym. Są to zarówno zaawansowane platformy edukacyjne, jak i aplikacje pomagające nauczycielom w organizacji pracy. Epidemia COVID-19, która w krótkim czasie przerodziła się w pandemię, stała

się katalizatorem tego procesu. W wielu krajach, w tym w Polsce, wprowadzono czasowe restrykcje dotyczące między innymi funkcjonowania placówek oświaty w celu ograniczenia rozprzestrzeniania się wirusa. W efekcie szkoły, uczelnie wyższe oraz instytucje szkoleniowe zostały zmuszone do zawieszenia bezpośrednich spotkań dydaktycznych i przekształcenia ich w formę zdalną [9]. Koniecznością była więc błyskawiczna adaptacja do nowych warunków.

Najpopularniejszą formą edukacji na odległość w obecnych czasach jest często z nią utożsamiany e-learning [10, 11]. Kształcenie w tej formie polega na wykorzystywaniu techniki komputerowej i sieciowej. E-learning umożliwia efektywne łącznie różnorodnych formatów treści dydaktycznych, takich jak tekst, grafika, animacja, dźwięk czy wideo, tworząc multimedialne, angażujące środowisko nauki. Pozwala też na personalizację sposobu nauczania, dostosowanie tempa nauki do indywidualnych możliwości ucznia oraz kontrolę nad procesem nauki. Forma ta powszechnie opiera się na wykorzystywaniu systemów zarządzania nauką (ang. Learning Management Systems, LMS) [11], czyli specjalistycznego oprogramowania służącego do tworzenia, dostarczania i administrowania treści dydaktycznymi oraz monitorowania postępu użytkowników.

Systemy zarządzania nauką tworzą wirtualne środowisko klasowe, naśladujące strukturę i dynamikę tradycyjnej edukacji oraz dające dodatkowe korzyści wynikające ze stosowania cyfrowych formatów treści dydaktycznych takie jak interaktywność i dostępność [12]. Przykładami popularnych platform LMS są Moodle oraz Blackboard [8, 11]. Są to kompleksowe rozwiązania pozwalające na zarządzanie wszystkimi aspektami kształcenia zdalnego – począwszy od tworzenia kursów, poprzez ich dystrybucję, zarządzanie użytkownikami, aż po śledzenie postępów i ewaluację nabytej przez uczniów wiedzy. Główne funkcje systemów LMS obejmują [10, 11, 13]:

- Tworzenie kursów, testów oraz ich szablonów – nauczyciele sprawują kontrolę nad formą, strukturą i treścią materiałów dydaktycznych, mogą również przygotowywać testy kontrolne i egzaminy, ocenяjące stopień przyswojenia materiału przez uczniów.
- Przechowywanie i zarządzanie materiałami dydaktycznymi – rozwiązania LMS umożliwiają przechowywanie i udostępnianie multimedialnych materiałów dydaktycznych, takich jak prezentacje, dokumenty tekstowe, grafiki, wideo czy nagrania dźwiękowe. Pozwalają na ich indeksowanie, kategoryzację, a także śledzenie zmian.
- Monitorowanie i ocenianie procesu nauki – nauczyciele mają możliwość śledzenia postępów uczniów w obrębie kursów poprzez informację na temat materiałów i sekcji kursu, które przeglądali oraz wyników testów i zadań. Mają również możliwość ręcznego oceniania testów i prac nadesłanych przez uczniów.
- Zapewnienie środków komunikacji – użytkownicy mają dostęp do synchronicznych i asynchronicznych form komunikacji poprzez integrację z pocztą email, fora dyskusyjne, czat, videokonferencje czy webinarium.
- Zarządzanie przestrzenią do współpracy – nauczyciele i uczniowie mogą pracować w ramach wirtualnych grup w celu wspólnej realizacji zadań i projektów. Mogą wymieniać się materiałami, prowadzić dyskusję i wspólnie pracować nad zadaniami.
- Zarządzanie procesem edukacji – dostępne są narzędzia pozwalające na tworzenie harmonogramu zajęć, planowanie spotkań, ustalanie terminów oddawania prac i zadań.
- Dostarczanie narzędzi do nauki i pracy – uczniowie mają do dyspozycji narzędzia ułatwiające naukę, takie jak edytory treści, interaktywne materiały, wizualizacje czy quizy.

2.3. Zdalna edukacja i współpraca muzyczna

Nauka muzyki w formie zdalnej stanowi szczególnie wyzwanie ze względu na specyfikę tej dziedziny. Kształcenie muzyczne, zwłaszcza na poziomie podstawowym i średnim, opiera się nie tylko na wiedzy z zakresu teorii muzyki, lecz również w dużej mierze na ćwiczeniu zdolności manualnych w postaci gry na instrumencie, umiejętnościach słuchowych i wokalnych, a także umiejętności współpracy w tym zakresie z innymi muzykami [14]. Wymaga to najczęściej bezpośredniej interakcji pomiędzy nauczycielem a uczniem. Brakuje ogólnodostępnych i prostych w użyciu narzędzi, które pozwoliłyby na sprawne nauczanie muzyki w środowisku zdalnym, co stało się szczególnie widoczne w czasie pandemii za sprawą problemów, jakich doświadczyły placówki edukacyjne, nauczyciele, uczniowie i ich opiekunowie.

Badania przeprowadzone na środowiskach szkolnych w czasie pandemii pokazały, że edukacja muzyczna na odległość polegała głównie na wykorzystaniu platform wideokonferencyjnych (np. Zoom) [15, 16, 17], a więc na synchronicznej formie współpracy. Rozwiązań te nie umożliwiają efektywnego prowadzenia zajęć praktycznych ze względu na występującą w nich latencję, czyli opóźnienie w przesyłce danych, oraz niewystarczającą jakość dźwięku [15, 16, 18]. Wynika to z przystosowania tych rozwiązań głównie do przesyłu mowy i obrazu w kontekstach, które tych wymagań nie narzucają. W praktyce nie pozwala to na wspólne wykonywanie utworów. Latencja oraz jej zmiany w czasie często uniemożliwiały synchronizację pomiędzy uczestnikami. Niska wierność dźwięku utrudniała prawidłową ocenę umiejętności uczniów. Sytuację utrudniały również czynniki zależne od środowiska uczestników. Częstym problemem była niestabilność połączenia internetowego, rozpraszające otoczenie uczniów (np. domownicy) czy różnice w poziomie głośności uczestników spotkania [15, 16]. Wady te spowodowały, że część nauczycieli skupiła się na uczeniu wyłącznie teorii muzyki [15].

Przeważająca większość nauczycieli wykorzystywała w trakcie pandemii kombinację synchronicznych i asynchronicznych metod nauczania, co pozwoliło na zniwelowanie części problemów pierwszego podejścia za pomocą drugiego [15, 16, 17]. Najpopularniejsze podejście polegało na wymianie plików video zawierających nagrania własnych wystąpień pomiędzy nauczycielami a uczniami, które były następnie omawiane w sposób synchroniczny w trakcie wideokonferencji. Sposób ten pozwala na częściowe obejrzenie problemów występujących przy graniu na żywo w środowisku sieciowym [15]. Nagranie ma lepszą jakość dźwięku, jest niezależne od jakości połączenia internetowego, a także pozwala na wybór dogodnego czasu i miejsca wykonania. Możliwość wielokrotnego odtworzenia nagrań pozwala na lepsze poznanie uczonego repertuaru, dogłębiejszą analizę występu zarówno przez nauczycieli, jak i uczniów oraz bardziej precyzyjną ocenę umiejętności [19]. Znaczna większość nauczycieli nie angażowała uczniów w zajęcia takie jak komponowanie, aranżacja czy tworzenie wirtualnego zespołu, co może świadczyć o małej dostępności, bądź dużym poziomem skomplikowania oprogramowania, które na to pozwala [16]. Powszechnie było również wykorzystanie opisanych wcześniej systemów LMS [16].

W trakcie przymusowych zajęć zdalnych zdarzały się przypadki, w których umiejętności techniczne gry uczniów, zwłaszcza tych o mniejszym stopniu zaawansowania, stopniowo pogarszała się [15]. Taka forma zajęć znacząco ogranicza ekspresję artystyczną oraz możliwość korekty techniki gry przez nauczyciela [18]. Świadczy to o istotnym wpływie kontaktu fizycznego pomiędzy nauczycielem a uczniem na jakość nauki gry na instrumencie. Podczas nauki wyłącznie zdalnej uczniowie przejawiali niższy poziom motywacji i entuzjazmu w porównaniu do nauki tradycyjnej [20]. Synchroniczne środowisko online jest również mocno indywidualistyczne, obecne techniki komunikacji pozwalają na skupienie uwagi głównie na osobie aktualnie mówiącej, co utrudnia interakcję pomiędzy uczniami, powodując znudzenie, brak koncentracji i poczucie wykluczenia [19]. Istotną przeszkodą był również poziom umiejętności technicznych uczniów i nauczycieli w kontekście obsługi narzędzi komputerowych. W jednym z badań istotna

część uczniów i nauczycieli (odpowiednio około 34% i 19%) zadeklarowała, że ich umiejętności techniczne stanowiły średni bądź duży problem w trakcie tej formy nauczania [16].

Istniejące wady tej formy edukacji nie pozwalają jednak na zastąpienie nią w pełni podejścia tradycyjnego. Mimo tych wyzwań forma ta oferuje szereg zalet, które mogą zostać wykorzystane do wzbogacenia tradycyjnych metod nauczania. Jest to rozwiązanie pozwalające na większą dostępność, elastyczność i przenośność nauczania [19]. Możliwość nauki w dowolnym miejscu, czasie i tempie pozwala na dostosowanie procesu edukacji do indywidualnych potrzeb uczniów. Daje uczniom dostęp do lekcji prowadzonych przez bardziej różnorodne grono nauczycieli oraz możliwość nauki od ekspertów z innych regionów czy krajów [18]. Dostęp do materiałów dydaktycznych i wirtualnych spotkań w dowolnym miejscu i czasie przyczynia się do zmniejszenia kosztów podróży oraz zwiększenia dostępności nauczania [18, 19]. Jest to szczególnie ważne w przypadku osób mieszkających w obszarach wiejskich i nisko zaludnionych, gdzie dostęp do edukacji muzycznej może być utrudniony. Niektórzy nauczyciele w trakcie wymuszonych zajęć zdalnych zadeklarowali, że mieli znacznie więcej kontaktu z uczniami niż wcześniej, zwłaszcza w przypadku tych ponadprzeciętnie zaangażowanych [17]. Praktyki takie jak mentoring rówieśniczy połączony z klasyczną formą nauczania pozwalają na zwiększenie ciągłości procesu edukacji i mogą wspierać tworzenie wspólnot praktyki, w których uczniowie mogą wspólnie pracować nad projektami muzycznymi oraz wymieniać się wiedzą i doświadczeniem [21]. Współpraca muzyczna promuje rozwój osobisty, rozwija kreatywność, buduje poczucie przynależności, wspiera naukę techniki gry i poszerza wachlarz umiejętności muzycznych [22]. Tworzenie muzyki w nieformalnych wspólnotach pozaszkolnych sprzyja nie tylko nauce umiejętności technicznych, ale także społecznych [23].

Wykorzystanie środowisk zdalnych może być więc dobrym sposobem na rozszerzenie klasycznej formy edukacji muzycznej, pozwalając na częstszy kontakt pomiędzy uczniami i nauczycielem, wzbogacenie o dodatkowe materiały i formy współpracy, a także dostosowanie nauki do indywidualnych potrzeb uczniów. To samo można powiedzieć w kontekście edukacji i współpracy muzycznej odbywającej się poza szkołą np. warsztatów, zespołów czy wspólnot takich jak otwarte orkiestry czy chóry. Przemawia to za potrzebą dalszego rozwoju narzędzi wspierających naukę i współpracę muzyczną w środowisku sieciowym.

2.3.1. Sieciowe wykonywanie muzyki

Sieciowe wykonywanie muzyki (ang. Networked Music Performance, skrót NMP) to forma wspólnego wykonywania utworu muzycznego w czasie rzeczywistym przez grupę osób znajdujących się fizycznie w różnych miejscach z wykorzystaniem techniki sieciowej [24]. Badania obserwacyjne przeprowadzone w trakcie pandemii [15, 16, 17] nie wskazują na wykorzystanie systemów NMP w edukacji muzycznej. W literaturze brakuje informacji, które określają powszechność popularność systemów NMP w edukacji.

Systemy te są bardziej popularne w kontekście współpracy muzycznej, choć i tutaj są rozwiązaniemi niszowymi. Większość interakcji pomiędzy muzykami poprzez internet sprowadza się do wymiany plików w celu współpracy offline (np. wykonywanie partii do wcześniej nagranych ścieżek) a współpraca w czasie rzeczywistym ogranicza się najczęściej do monitorowania (np. kompozytor nadzorujący na żywo orkiestrę przygotowującą nagranie) [25]. Dziedzina systemów NMP jest wciąż rozwijana i badana w środowisku akademickim. Współpraca muzyczna w tej formie narzuca szereg istotnych wymagań na realizujące ją systemy. Dwie największe przeszkody to konieczność zapewnienia odpowiednio małego opóźnienia i jego wariancji oraz zagwarantowanie odpowiedniej jakości dźwięku [26]. Wyznaczone empirycznie wartości opóźnienia, które są tolerowane przez muzyków, mieszczą się w przedziale od 20 do 30 milisekund, choć faktyczna wartość uwarunkowana jest też umiejętnościami i stylem danego muzyka [27]. W kontekście jakości dźwięku przesyłanego pomiędzy użytkownikami ilość danych potrzebna

do przesyłania reprezentacji dźwięku rośnie wraz z jego rozdzielczością, co może skutkować wzrostem opóźnienia w zależności od warunków sieciowych, istotny jest więc odpowiedni kompromis między tymi parametrami. Wpływ na opóźnienie ma nie tylko infrastruktura sieciowa samego systemu, ale też uwarunkowania lokalnych sieci użytkowników.

Istnieją studia przypadków opisujące praktyczne wykorzystanie systemów NMP. Projekty takie jak LOLA (Low Latency audio visual streaming system) [28], czy DIAMOUSES [29, 30] umożliwiły transmisję dźwięku i obrazu o niskiej latencji i wysokiej jakości, co pozwoliło muzykom na synchroniczne wykonywanie utworów mimo znaczących odległości. Przykłady te demonstrują możliwość stworzenia użytecznej i niezawodnej implementacji systemu NMP, a także wskazują na potencjał takich rozwiązań we współpracy muzycznej, pozwalając na omijanie ograniczeń wynikających z dużej odległości geograficznej pomiędzy uczestnikami. Rozwiązania te wiążą się jednak z wysokimi wymaganiami technicznymi i kosztami, które ograniczają ich dostępność i praktyczne zastosowanie. Wymagają one wysokowydajnej infrastruktury sieciowej, specjalistycznego sprzętu i oprogramowania, które minimalizuje opóźnienia [24]. Obecnie istnieje szereg ogólnodostępnych aplikacji komputerowych pozwalających sieciowe wykonywanie muzyki również przez amatorów, takie jak JackTrip, Jamulus, JamKazam, SonoBus czy Deck 10 [31]. Rozwiązania te opierają się o transmisję dźwięku z urządzeń audio podłączonych do komputera użytkownika poprzez sieć. Wymagają one łącza internetowego o niskim opóźnieniu (w praktyce koniecznością jest łącze przewodowe) i dużej stabilności, a także wiedzy na temat konfiguracji sieciowej komputera.

Alternatywą dla przesyłania dźwięku cyfrowego w systemach NMP jest transmisja sygnałów MIDI [26]. Protokół MIDI (Musical Instrument Digital Interface) to standard komunikacji opracowany w 1983 roku, który umożliwia wymianę informacji muzycznych pomiędzy komputerami i instrumentami cyfrowymi [32]. MIDI nie przesyła dźwięku jako takiego, lecz sekwencje danych reprezentujących informacje o zdarzeniach związanych z instrumentami muzycznymi, takich jak naciśnięcie klawiszy, przycisków, pedałów oraz powiązanych z nimi parametrów takich jak prędkość czy użyta siła, a także zmiany w ich konfiguracji. Dzięki temu, dane MIDI są znacznie lżejsze od cyfrowego dźwięku, co pozwala na przesyłanie informacji z mniejszym opóźnieniem. Standard definiuje również format pliku, który pozwala na zapis i odtwarzanie sekwencji danych MIDI. Wykorzystanie standardu implikuje konieczność użycia kompatybilnych instrumentów cyfrowych, a także uniemożliwia przesyłanie wokalu [26]. Podejście to nie jednak popularne w dziedzinie NMP i ogranicza się do nielicznych projektów badawczych [33, 34].

2.3.2. Cyfrowe stacje robocze w pracy grupowej

Cyfrowe stacje robocze (ang. Digital Audio Workstation, skrót DAW) to oprogramowanie służące do nagrywania, tworzenia, edycji i mikowania dźwięku w formie cyfrowej. Są powszechnie wykorzystywane przez profesjonalistów w środowiskach do produkcji muzyki i dźwięku takich jak stacje radiowe, studia nagraniowe, studia postprodukcyjne, jak i przez twórców amatorskich [35, 36]. Standardowo oprogramowanie obsługuje jedna osoba naraz, a współpraca polega na fizycznej obecności uczestników w jednym miejscu (np. w studiu nagraniowym) [36].

Współczesne aplikacje DAW pozwalają również na zdальną pracę grupową. Narzędzia tego typu pozwalają na zdальną współpracę zarówno producentom, inżynierom dźwięku oraz muzykom, która może być realizowana na różnych etapach procesu produkcyjnego, od aranżacji, poprzez sesje nagraniowe, aż do mikowania i masteringu. Są to głównie rozszerzenia bądź samodzielne oprogramowanie pozwalające na transmisję dźwięku w czasie rzeczywistym z i do aplikacji DAW (np. Source-Connect Pro, VST Connect Pro) [37]. Najczęściej w tego typu rozwiązaniach dostęp do projektu dźwiękowego ograniczony jest do jednej osoby, pozostały uczestnicy mogą jedynie słuchać i nagrywać swoje partie w czasie rzeczywistym. Podobnie jak w systemach NMP wymaga to odpowiednio stabilnego i szybkiego łącza

internetowego oraz odpowiedniej konfiguracji sprzętu audio.

Istnieją również platformy internetowe takie jak BandLab czy SoundTrap, które odgrywają rolę DAW w postaci aplikacji przeglądarkowej [37, 38]. Są to rozwiązania dużo bardziej przystępne dla użytkowników amatorskich ze względu na prosty interfejs i dostępność z poziomu przeglądarki internetowej. Pozwalają one na pełny dostęp do projektu muzycznego przez wszystkich uczestników, a pliki projektowe przechowywane są w chmurze. Praca w takich aplikacjach jest częściowo synchroniczna bądź asynchroniczna, tj. każdy z użytkowników może w dowolnym momencie dodawać, edytować i usuwać niezależne od siebie ścieżki dźwiękowe, a następnie odsłuchać i pobrać wynikowy utwór. Pozwala to uniknąć problemów związanych z opóźnieniem i jakością dźwięku kosztem utraty możliwości w pełni synchronicznej współpracy. Aplikacje dostarczają również narzędzi do komunikacji takich jak czat i rozmowy video co pozwala na częściowo synchronczną pracę. Wykorzystanie cyfrowych stacji roboczych w edukacji muzycznej może być dobrym sposobem na rozwijanie kreatywności uczestników [39]. Mimo małej popularności, rozwiązania takie SoundTrap z powodzeniem zostały wykorzystane w edukacji i współpracy muzycznej w trakcie pandemii [38, 40].

2.4. Podsumowanie

Przedstawiony przegląd literatury i istniejących rozwiązań pokazuje złożoność problemu zdalnej edukacji i współpracy muzycznej. Modele komputerowej współpracy grupowej są użytecznym narzędziem w analizie istniejących systemów oraz analizie wymagań i projektowaniu nowych rozwiązań. Modele takie jak MoCA podkreślają potrzebę interdyscyplinarnego podejścia w projektowaniu systemów współpracy, które uwzględniają nie tylko wymagania i możliwości techniczne, lecz także uwarunkowania społeczno-kulturowe docelowych użytkowników. Jest to ważne zwłaszcza w dziedzinie muzyki, która oprócz wiedzy teoretycznej opiera się w dużej mierze na umiejętnościach praktycznych oraz bezpośrednich interakcjach pomiędzy członkami grupy.

Specyfika dziedziny muzyki w kontekście zdalnej współpracy jest też szczególnym wyzwaniem od strony realizacji technicznej. Wymaga ona zapewnienia odpowiednio niskiego i stabilnego opóźnienia oraz wysokiej jakości podczas transmisji dźwięku. Analiza istniejących narzędzi i systemów pokazuje, że spora część dostępnych rozwiązań jest zarezerwowana dla dobrze wyposażonych środowisk profesjonalnych i akademickich oraz osób obytych technicznie. Rozwiązania takie jak NMP i DAW są często skomplikowane w użyciu i wymagają od użytkowników odpowiedniego doświadczenia i sprzętu. Skutkuje to ograniczonym dostępem do tych technologii, które miało swoje odzwierciedlenie w wyborach narzędzi do zdalnego nauczania przez nauczycieli muzyki. Proste rozwiązania takie jak platformy wideokonferencyjne okazały się nieodpowiednie do prowadzenia zajęć praktycznych ze względu na niespełnienie wymagań wymienionych wcześniej dotyczących transmisji dźwięku. Skloniło to nauczycieli do szukania alternatywnych rozwiązań, np. w formie asynchronicznej wymiany plików video, a części z nich do ograniczenia zajęć wyłącznie do teorii. Obecny stan techniki nie pozwala na zastąpienie w pełni tradycyjnej nauki i współpracy muzycznej formą zdalną.

Obiecującymi alternatywami dla systemów NMP i klasycznych DAW są aplikacje takie jak SoundTrap i BandLab. Pozwalają one na asynchroniczną współpracę w formie cyfrowej stacji roboczej dostępnej za pomocą przeglądarki internetowej. Są to rozwiązania prostsze, bardziej przystępne i niewymagające dużej wiedzy technicznej. Brakuje jednak systemów łączących możliwości tego typu aplikacji z zaletami systemów LMS, które są powszechnie stosowane w edukacji zdalnej. Rozszerzenie przeglądarkowych aplikacji DAW o funkcjonalności systemów LMS, mogłoby stworzyć kompleksowy system wspierający naukę i współpracę muzyczną w środowisku sieciowym.

3. PROJEKT, IMPLEMENTACJA I TESTY SYSTEMU

Rozdział projektowy niniejszej pracy magisterskiej przedstawia wymagania, architekturę oraz implementację prototypu systemu. Określono w nim komponenty systemu, ich hierarchię, interakcje i przepływ danych pomiędzy nimi. Uzasadniono również wybory techniczne takie jak wybór środowiska i platform programistycznych, bibliotek, czy narzędzi, które zostały wykorzystane podczas realizacji projektu. Proponowany system nazwano *Concerto*.

3.1. Analiza wymagań

W odpowiedzi na zidentyfikowaną w przeglądzie literatury i istniejących rozwiązań lukę zaproponowano system łączący w sobie wybrane cechy systemów LMS i internetowych cyfrowych stacji roboczych. System nie wymaga od użytkowników znacznych umiejętności technicznych ani nakładów finansowych i pozwala na szybkie rozpoczęcie współpracy. Wymagania systemu zostały określone na podstawie przeglądu oraz ograniczeń wynikających z charakteru pracy magisterskiej i dostępnych zasobów. Celem systemu jest usprawnienie procesów wspólnego tworzenia, wykonywania i nauki muzyki poprzez stworzenie jednolitego środowiska do spotkań, wymiany informacji i współdzielenia materiałów. Do scharakteryzowania systemu i określenia wymagań posłużyły się siedmioma wymiarami modelu MoCA opisanego w rozdziale 2.1.2. Określone wymagania podsumowano dzieląc je na funkcjonalne i niefunkcjonalne. Wymagania funkcjonalne określają funkcje, które system powinien realizować, niefunkcjonalne określają cechy, które system powinien posiadać.

3.1.1. Synchroniczność i rozkład fizyczny

Z punktu widzenia wymiarów synchroniczności oraz rozkładu fizycznego współpracu muzyczna może być realizowana w sposób mieszany. Zdalna współpraca synchroniczna w formie wirtualnych spotkań, prób czy lekcji może być wartościowym zamiennikiem spotkań rzeczywistych w przypadku ograniczeń w ich częstotliwości. Wymaga to jednak dostosowania narzędzi i formy współpracy do obecnych możliwości technicznych aby spełnić wymagania związane z wiernością i opóźnieniem dźwięku. Proponowanym rozwiązaniem jest połączenie bazowych funkcjonalności systemów LMS takich jak wideokonferencje, forum dyskusyjne i współdzielenie materiałów z możliwością częściowo synchronicznego tworzenia muzyki analogicznego do internetowych stacji roboczych. Zintegrowanie współdzielonego edytora audio z wideokonferencją pozwoli na wspólną pracę nad utworami muzycznymi w trakcie spotkań. Możliwość przechowywania i współdzielenia materiałów, komunikacji asynchronicznej oraz dostęp do systemu z urządzeń mobilnych umożliwia użytkownikom pracę we własnym tempie niezależnie od czasu i miejsca. Pozwoli również na wzbogacenie zarówno spotkań zdalnych jak i rzeczywistych o możliwość dokumentowania i archiwizacji efektów pracy. W celu zwiększenia dostępności część funkcjonalności systemu powinna być dostępna także w formie aplikacji mobilnej. Aby pozwolić na potencjalną współpracę międzynarodową pomiędzy użytkownikami, podstawowa wersja systemu powinna być dostępna w języku angielskim ze względu na jego powszechność.

3.1.2. Skala i liczba wspólnot praktyki

W kontekście wymiarów skali i liczby wspólnot praktyki system jest skierowany do grup o małej wielkości tworzących jedną wspólnotę praktyki bądź większych grup podzielonych na mniejsze zespoły.

Przykłady takich grup obejmują amatorskie zespoły, szkoły muzyczne czy lokalne społeczności praktykujące muzykę takie jak chóry czy domy kultury. System powinien więc umożliwić pracę w niezależnych od siebie przestrzeniach aby zapewnić użytkownikom prywatność i swobodę w organizacji pracy. Ograniczenie skali systemu pozwoliło skupić się na aspektach funkcjonalnych i wstępnej ocenie przydatności proponowanego rozwiązania.

3.1.3. Nieugruntowanie, planowana stałość i obrót

Pod względem nieugruntowania system może być wykorzystywany zarówno przez rozwijające się jak i rutynowe grupy użytkowników. Niski próg umiejętności technicznych wymaganych do korzystania z niego pozwoli nowym grupom szybko rozpocząć współpracę, w przypadku grup rutynowych umożliwi łatwe dostosowanie się do nowego środowiska. Możliwość zarządzania użytkownikami, ich rolami oraz osobnymi przestrzeniami roboczymi pozwoli obsłużyć grupy o dowolnej planowanej stałości i obrocie.

3.1.4. Wymagania funkcjonalne

System musi:

- umożliwić organizowanie spotkań online z wieloma uczestnikami naraz,
- pozwalać na przeprowadzanie wirtualnych sesji tworzenia muzyki (np. lekcje muzyki, próby zespołowe, sesje improwizacyjne),
- umożliwiać nagrywanie spotkań i ich archiwizację,
- pozwalać na przechowywanie i współdzielenie dowolnych materiałów cyfrowych,
- pozwalać na tworzenie organizowanie materiałów w katalogi,
- dawać możliwość tworzenia i zarządzania niezależnymi przestrzeniami roboczymi, w obrębie których będą prowadzone spotkania i gromadzone materiały,
- pozwalać na zarządzanie użytkownikami i ich uprawnieniami w obrębie systemu, przestrzeni roboczych i katalogów,
- umożliwić użytkownikom forum do asynchronicznej dyskusji z możliwością odnoszenia się do zgromadzonych materiałów.

Wymagania funkcjonalne systemu zostały również przedstawione w formie dwuczęściowego diagramu przypadków użycia. Diagramy przedstawiają czynności, które mogą być wykonywane przez użytkowników na poziome systemu oraz przestrzeni roboczej (odpowiednio rysunki 3.1 i 3.2) oraz hierarchię ról w ich obrębie.

3.1.5. Wymagania niefunkcjonalne

System musi:

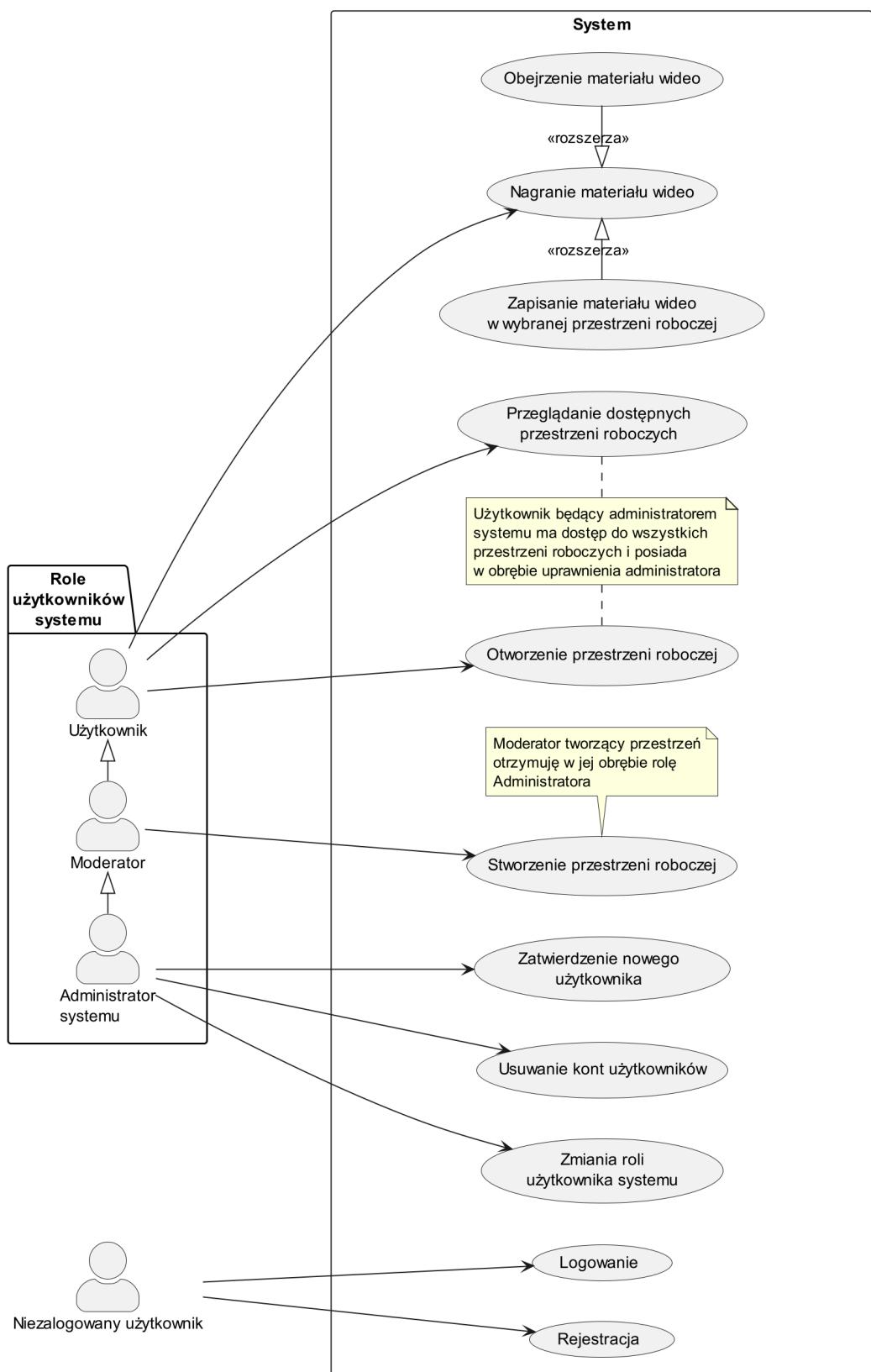
- być łatwy w obsłudze i intuicyjny dla użytkowników o różnym stopniu zaawansowania technicznego,
- być dostępny w formie aplikacji mobilnej oraz na komputerach osobistych,
- chronić materiały oraz dane użytkowników przed niepowołanym dostępem,
- być dostępny przez cały czas,
- zapewnić użytkownikom płynne doświadczenie na urządzeniach o przeciętnej wydajności.
- być dostępnym co najmniej w języku angielskim,

3.1.6. Model domeny

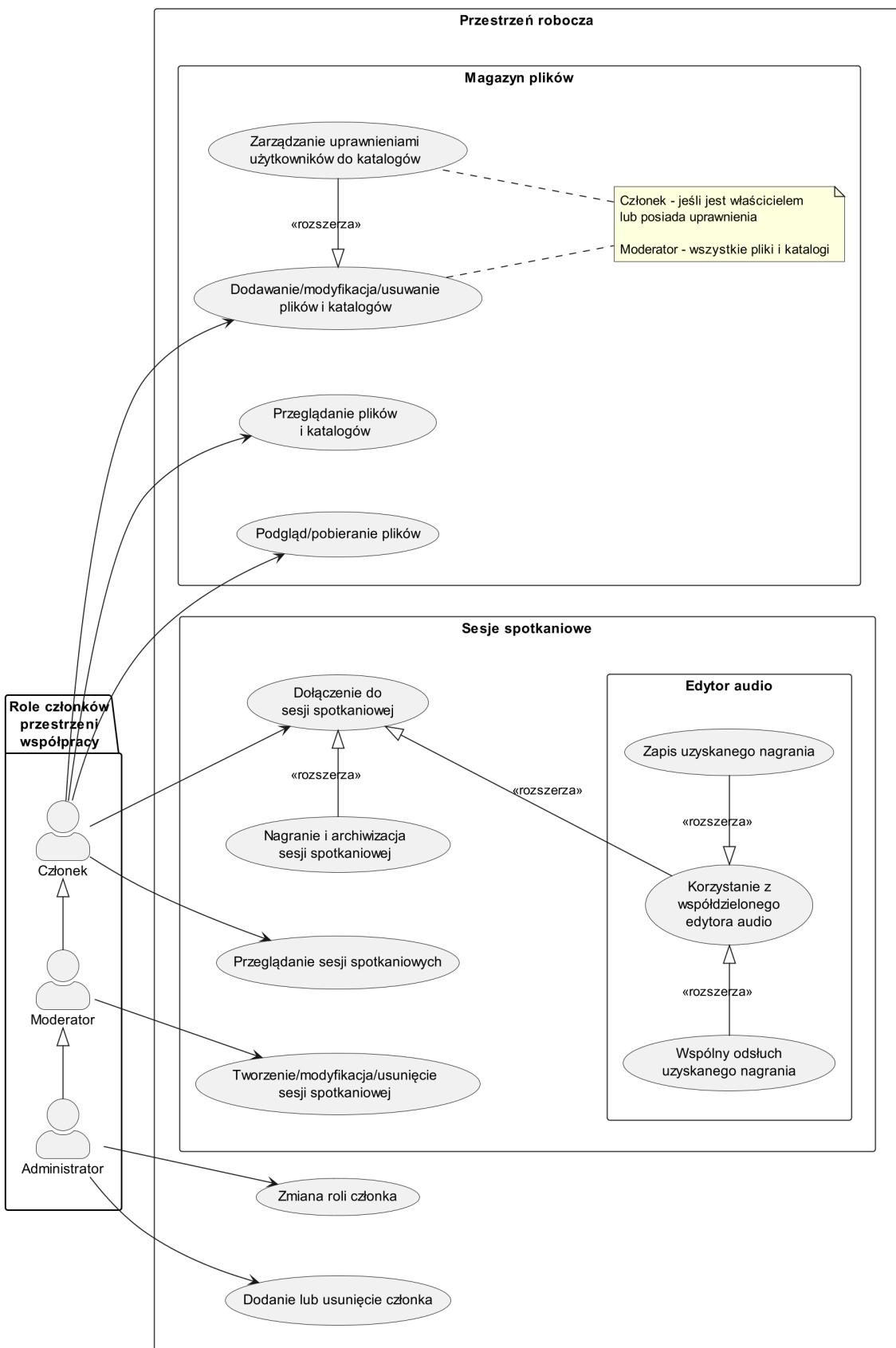
Na podstawie określonych wymagań zidentyfikowano domenę problemu i stworzono jej model przedstawiony w formie diagramu na rysunku 3.3. Model domeny przedstawia dziedzinę problemu, w której działa system za pomocą encji (bytów w nim wyróżnionych), ich atrybutów (właściwości, które posiadają) oraz relacji pomiędzy nimi. Jest to model koncepcyjny, który pozwala na zrozumienie struktury elementów systemu oraz ich powiązań niezależnie od jego faktycznej implementacji. Diagram wykorzystuje notację diagramu klas UML. Typy relacji wykorzystane w modelu to: asocja, kompozycja oraz agregacja. Asocja jest ogólną relacją łączącą dwie klasy. Kompozycja i agregacja są szczególnymi przypadkami asocjacji. Kompozycja oznacza, że jeden z elementów jest częścią drugiego i nie może bez niego istnieć. Agregacja oznacza, że jeden z elementów jest częścią drugiego, ale może istnieć niezależnie od niego. Relację określa również krotność, która mówi ile instancji jednego elementu, może być powiązanych z instancją drugiego elementu. Relacje, które same w sobie zawierają atrybuty, umieszczone w modelu jako osobne klasy asocjacyjne, przyłączone do relacji. Oznaczenia relacji i typów prostych zostały opisane w legendzie diagramu. Litera „E” w zielonym okręgu oznacza encję, litera „E” w czerwonym okręgu oznacza typ wyliczeniowy. Elementy modelu domeny zostały wymienione i opisane w tabeli 3.1.

Tabela 3.1: Elementy modelu domeny systemu

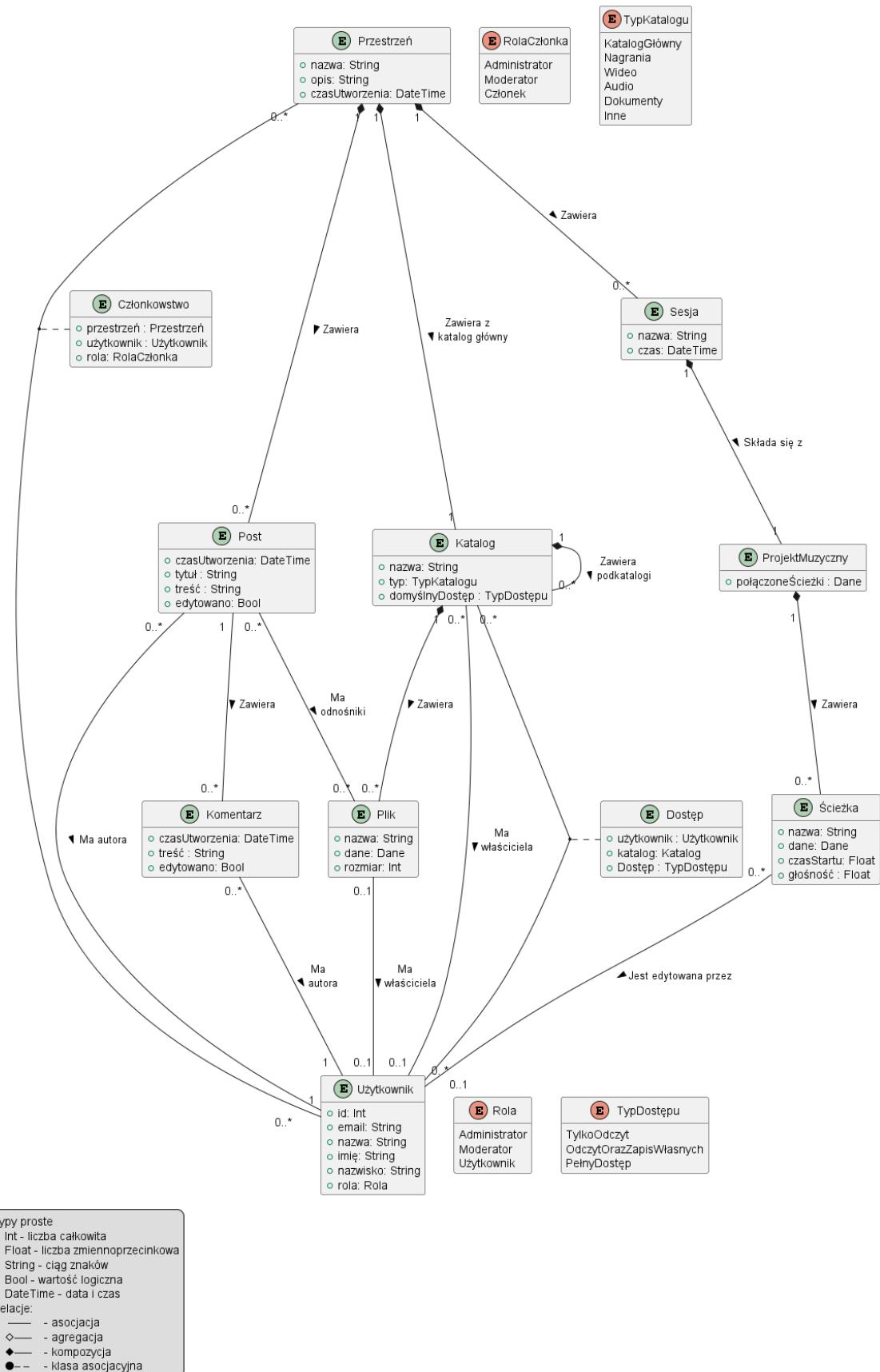
Nazwa elementu	Opis
Użytkownik	Osoba korzystająca z systemu. Każdy użytkownik posiada unikalny numer identyfikacyjny (ID), adres e-mail, nazwę, imię, nazwisko oraz rolę pełni w systemie.
Rola	Rola użytkownika w systemie. Określa uprawnienia użytkownika w kontekście całego systemu.
Przestrzeń	Niezależna przestrzeń robocza, w której odbywają się sesje spotkaniove, przechowywane są materiały i prowadzone są dyskusje. Przestrzeń robocza składa się z członków (podzbiór użytkowników), katalogu głównego oraz postów na forum dyskusyjnym.
Członkostwo	Relacja łącząca użytkownika z przestrzenią roboczą. Określa jego przynależność do przestrzeni oraz rolę jaką w niej pełni.
Rola członka	Typ wyliczeniowy określający rolę członka w obrębie przestrzeni roboczej. Rola określa uprawnienia użytkownika w obrębie przestrzeni.
Katalog	Katalog, w którym przechowywane są materiały. Katalogi mogą zawierać katalogi podrzędne oraz pliki tworzące w obrębie przestrzeni drzewiastą strukturę zaczynającą się od katalogu głównego. Katalogi mają przypisany typ oraz domyślny typ dostępu dla członków przestrzeni.
Typ katalogu	Typ wyliczeniowy określający kategorię materiałów zawartych w katalogu.
Plik	Plik przechowywany w systemie.
Dostęp	Uprawnienie użytkownika w obrębie katalogu. Nadpisuje domyślny typ dostępu w katalogu dla konkretnego użytkownika.
Typ dostępu	Typ wyliczeniowy określający uprawnienia użytkowników w obrębie katalogu. Użytkownicy mogą mieć jeden z trzech trybów dostępu: odczyt, odczyt i zapis tylko własnych plików i katalogów, odczyt i zapis wszystkich plików i katalogów.
Post	Post na forum dyskusyjnym przestrzeni roboczej. Zawiera tytuł, treść oraz opcjonalne odnośniki do plików. W obrębie posta mogą być dodawane komentarze.
Komentarz	Komentarz do posta na forum dyskusyjnym.
Sesja	Wirtualna sesja spotkaniova, do której mają dostęp członkowie przestrzeni. Podczas sesji użytkownicy uczestniczą w wideokonferencji i wspólnie tworzą projekt muzyczny.
Projekt muzyczny	Projekt muzyczny tworzony podczas sesji, składający się z osobnych ścieżek, które są potem łączone w całość. Wynikowy utwór można zapisać jako plik w katalogu i wspólnie odsłuchać podczas sesji.
Ścieżka	Ścieżka w projekcie muzycznym, przechowująca dane dźwiękowe. Ścieżka może być edytowana przez jednego użytkownika na raz. W czasie edycji użytkownik może nagrać bądź zimportować dźwięk, przesuwać względem innych ścieżek i zmieniać głośność.



Rysunek 3.1: Ogólny diagram przypadków użycia systemu



Rysunek 3.2: Diagram przypadków użycia przestrzeni roboczej



Rysunek 3.3: Diagram domeny systemu

3.2. Wysokopoziomowa architektura systemu

Na podstawie analizy wymagań zostały określone wstępne założenia architektoniczne systemu. Architektura systemu bazuje na modelu klient-serwer. Posiada również cechy powszechnych w inżynierii oprogramowania modeli architektonicznych – wielowarstwowego (ang. multi-layered) oraz wielopięnnego (ang. multi-tiered). Modele te opisują sposób organizacji systemu w aspektach odpowiednio: logicznym i topologicznym [41, 42]. Przedstawiony model architektury został wybrany ze względu na niski koszt czasowy implementacji i utrzymania oraz prostotę, które są przydatne w początkowych fazach rozwoju systemu i weryfikacji jego użyteczności. W dalszych etapach rozwoju systemu możliwe jest zastosowanie bardziej zaawansowanych modeli architektury lub ich kombinacji, takich jak architektura zorientowana na usługi (ang. service-oriented architecture) czy sterowana zdarzeniami (event-driven architecture) [41], które mogą przynieść korzyści w postaci większej elastyczności, skalowalności i niezawodności systemu. Architektura proponowanego rozwiązania składa się z kilku sprzężonych z sobą komponentów:

- klient aplikacji,
- serwer aplikacji,
- platforma wideokonferencyjna,
- dostawca tożsamości,
- baza danych,
- magazyn plików.

Oba podziały architektoniczne oraz komponenty systemu są widoczne na rysunku 3.2 przedstawiającym ogólną architekturę proponowanego systemu.

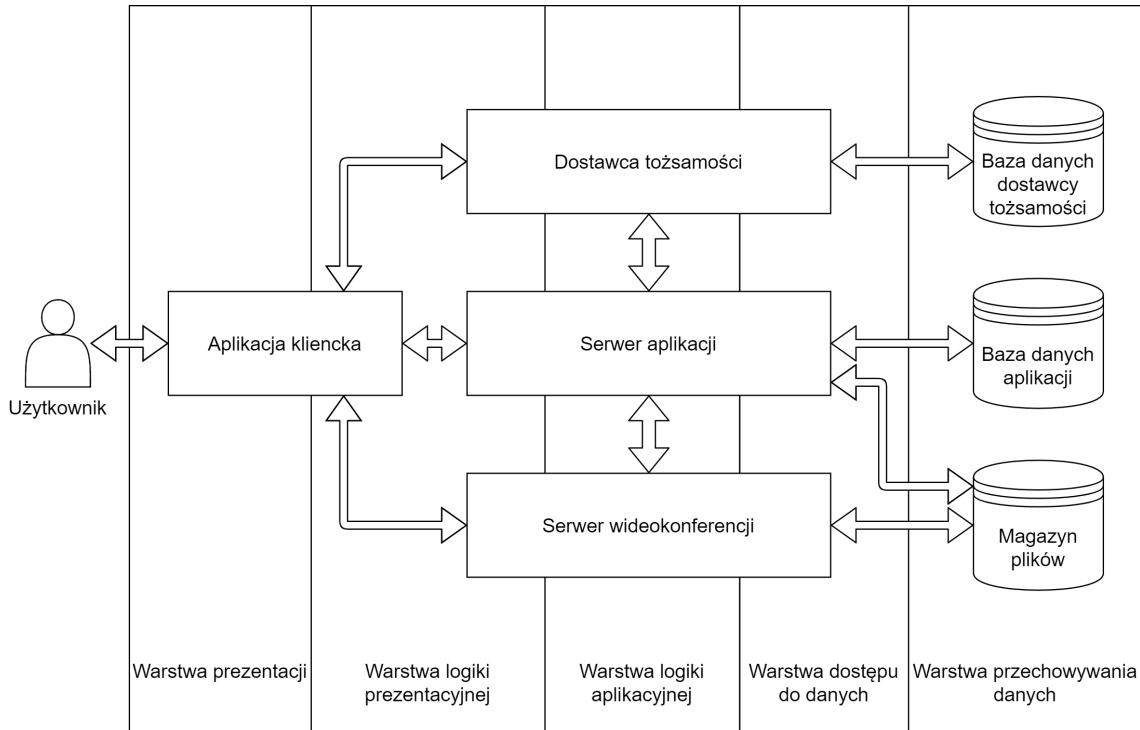
Aplikacja kliencka dostarcza interfejsu użytkownika i komunikuje się z serwerem aplikacji za pomocą interfejsu webowego. Jest zrealizowana jako progresywna aplikacja webowa (ang. progressive web application, PWA), która może zostać uruchomiona w przeglądarce internetowej oraz dodana do listy aplikacji na urządzeniu użytkownika imitując aplikację natywną zarówno na platformach mobilnych, jak i desktopowych. Aplikacja jest dostarczana do urządzenia przez serwer aplikacji w postaci plików statycznych.

Serwer aplikacji jest głównym komponentem systemu. Realizuje logikę domenową aplikacji, dostarcza webowego interfejsu programowania aplikacji (ang. Web Application Programming Interface, WebAPI), autoryzuje skierowane do niego żądania, zarządza bazą danych, dostępem do plików i usługi wideokonferencji. Odpowiada również za serwowanie aplikacji klienckiej.

Platforma wideokonferencyjna jest wykorzystywana do przeprowadzania spotkań online oraz ich nagrywania. Z uwagi na złożoność tego typu rozwiązań, ograniczenia czasowe oraz wymagania dotyczące niezawodności i jakości transmisji komponent został zrealizowany przy użyciu otwartoźródłowego rozwiązania wideokonferencyjnego, które zostało przystosowane na potrzeby projektu.

Dostawca tożsamości (ang. identity provider, IDP) jest odpowiedzialny za uwierzytelnianie użytkowników, zarządzanie ich kontami oraz dostarczenie informacji o ich tożsamości i rolach w systemie. Dostarcza podstawowych funkcjonalności związanych z kontami użytkowników, takich jak rejestracja, logowanie, zmiana i odzyskiwanie hasła, zarządzanie kontem, zarządzanie rolami. Dostawca tożsamości jest wykorzystywany zarówno przez serwer aplikacji, jak i aplikację kliencką. W celu zwiększenia bezpieczeństwa systemu oraz ułatwienia ewentualnej integracji z innymi systemami w przyszłości komponent został zrealizowany za pomocą gotowego rozwiązania z otwartym kodem źródłowym.

Baza danych odpowiada za utrwalanie i dostęp do wszelkich danych związanych z modelem domenowym aplikacji. Gromadzone są w niej informacje o użytkownikach, przestrzeniach roboczych i ich



Rysunek 3.4: Diagram ogólnej architektury systemu

zawartości, spotkaniach oraz innych klas domenowych. Z bazy danych korzystają zarówno serwer aplikacji, jak i dostawca tożsamości. W faktycznej implementacji komponent ten może być realizowany przez jedną lub kilka baz danych, w zależności od potrzeb i sposobu wdrożenia. Magazyn plików jest wykorzystywany do przechowywania faktycznej zawartości plików przesyłanych przez użytkowników oraz zarchiwizowanych nagrań z przeprowadzonych spotkań, które mają swoją reprezentację w bazie danych.

Model wielowarstwowy określa logiczną organizację systemu, w którym każda warstwa jest odpowiedzialna za określony, spójny zakres funkcjonalności, które nie muszą znajdować się w tej samej warstwie fizycznej. Podział ten Komunikacja powinna odbywać się wyłącznie w obrębie tej samej warstwy lub warstw sąsiednich [41, 42]. Główną korzyścią wynikającą z zastosowania modelu wielowarstwowego jest podział odpowiedzialności, każda z warstw realizuje ściśle określony zakres funkcjonalności, co ułatwia modyfikację i utrzymanie rozwiązania. Rozdzielnosc warstw skutkuje zmniejszonym powielaniem kodu, pozwala na jego wielokrotne wykorzystanie w obrębie systemu. W ujęciu tego modelu, w architekturze systemu możemy wyróżnić następujące warstwy:

- warstwa prezentacji,
- warstwa logiki prezentacji,
- warstwa logiki aplikacyjnej,
- warstwa dostępu do danych,
- warstwa przechowywania danych.

Warstwa prezentacji odpowiada za dostarczenie interfejsu użytkownika, a więc za przedstawienie mu informacji obecnych w systemie w sposób zrozumiały dla człowieka oraz odbieranie od niego danych wejściowych. Stanowi pomost między użytkownikiem a infrastrukturą i logiką systemu.

Warstwa logiki prezentacji obsługuje zdarzenia generowane poprzez interakcję użytkownika z interfejsem aplikacji. Ze względu na wielopienność systemu warstwa ta jest zrealizowana zarówno w aplikacji klienckiej, jak i serwerze. Po stronie serwera odpowiada za definicję i obsługę interfejsu webowego apli-

kacji (WebAPI), w tym autoryzację żądań, walidację przesłanych danych i wywoływanie odpowiednich operacji w warstwie logiki aplikacyjnej na ich podstawie. W aplikacji klienckiej odpowiada za komunikację z serwerem aplikacji i zmianę stanu interfejsu użytkownika na zdarzeń i danych otrzymanych od serwera lub użytkownika.

Warstwa logiki aplikacyjnej jest trzonem systemu, w którym realizowane są jego główne funkcjonalności. Odpowiada za wykonywanie operacji właściwych dla systemu określonych w modelu domeny, który implementuje. Stanowi granicę pomiędzy modelem domeny a danymi prezentowanymi użytkownikowi przekształcając je na odpowiednią formę, ukrywając tym samym szczegóły implementacyjne logiki domenowej. Zmiany w systemie będące wynikiem operacji przekazuje do warstwy dostępu do danych, która je utrwała.

Do obowiązków warstwy dostępu do danych należy zarządzanie komunikacją z bazą danych i magazynem plików oraz izolację ich szczegółów implementacyjnych, takich jak miejsce przechowywania danych, sposób ich reprezentacji i wykonywania zapytań. Pozwala ona na odczyt danych oraz utrwalanie zmian wprowadzonych do nich przez warstwę logiki aplikacyjnej.

Model wielopienny określa podział topologiczny systemu grupując jego komponenty w pnie (ang. tiers) sugerujące ich położenie fizyczne. Pnie ułożone są w sposób hierarchiczny, w którym pień poprzedni jest zależny od następnego. W przypadku rozważanego systemu pień kliencki jest zależny od pnia serwerowego, a ten z kolei od pnia danych. Implementacja poszczególnych pnii musi umożliwiać rozmieszczenie ich na różnym sprzęcie fizycznym, choć nie powinna wykluczać sytuacji, w której znajdują się one na tej samej maszynie. Możliwe jest również horyzontalne skalowanie systemu poprzez zwielokrotnienie komponentów składających się na dany pień, np. poprzez uruchomienie wielu instancji serwera aplikacji. Rozdzielenie fizyczne danych od komponentów serwerowych pozwala na ich bezpieczne przechowywanie na dedykowanym sprzęcie. W kontekście modelu wielopiennego proponowany system możemy podzielić na trzy pnie:

- pień kliencki,
- pień serwerowy,
- pień danych,

Pień kliencki znajduje się na urządzeniach użytkowników, na których uruchamiana jest aplikacja kliencka. Pień serwerowy zawiera serwery aplikacji, platformy wideokonferencyjnej oraz dostawcy tożsamości. W pniu danych mieszą się baza danych oraz magazyn plików.

3.3. Wybór rozwiązań technicznych

W ramach projektu zostało dobrane środowisko, platformy i biblioteki programistyczne, które wykorzystano później do zaimplementowania systemu. Najważniejszym kryterium wyboru była możliwość realizacji wymagań postawionych przed systemem. Z uwagi na holistyczny charakter projektu, ograniczenia w postaci czasu oraz zasobów (praca indywidualna), dodatkowe kryteria wyboru obejmowały również możliwość szybkiego i łatwego rozpoczęcia pracy nad systemem, minimalizację ilości różnych środowisk programistycznych, dostępność bibliotek oraz możliwość wykorzystania wiedzy i doświadczenia z poprzednich projektów. Poniżej przedstawiono poszczególne wybory techniczne dla każdego z komponentów systemu.

3.3.1. Aplikacja kliencka

Najistotniejszym wymaganiem dotyczącym aplikacji klienckiej w kontekście wyboru technologii było zapewnienie jej dostępności niezależnie od systemu operacyjnego użytkownika oraz rodzaju urządzenia.

Została ona zrealizowana w formie progresywnej aplikacji internetowej (ang. progressive web application, PWA), która może zostać uruchomiona w przeglądarce internetowej oraz dodana do pulpitu użytkownika imitując aplikację natywną, zarówno na platformach mobilnych, jak i desktopowych. Do zaimplementowania tej części systemu wybrany została platforma Blazor w modelu WebAssembly (wersja 7.0.5) oparta na platformie programistycznej .NET w wersji 7. Wybór ten jest powiązany między innymi z wykorzystaniem platformy ASP.NET Core w serwerze aplikacji, opisanym w podrozdziale 3.3.2.

Progresywne aplikacje internetowe

Aplikacje PWA umożliwiają uruchomienie ich w dowolnej platformie sprzętowej obsługującej przeglądarkę internetową. Celem tego typu aplikacji jest przybliżenie doświadczenia użytkowników do takiego, które oferują aplikacje natywne. Wykorzystują do tego mechanizmy dostępne w przeglądarkach internetowych, takie jak Service Worker i Web App Manifest [43].

Service Worker jest specjalnym skryptem języka JavaScript, który działa w dedykowanym wątku w tle i pośredniczy w komunikacji między aplikacją, przeglądarką i siecią. Potrafi przechwytywać żądania sieciowe w obrębie aplikacji, dzięki czemu pozwala programiście na zdefiniowanie sposobu ich obsługi. Skrypt ten może na przykład decydować o tym, czy dane mają zostać pobrane z sieci, pamięci podręcznej czy z użyciem mechanizmów takich jak Web Storage API lub IndexedDB. Mechanizmy Web Storage API oraz IndexedDB pozwalają na przechowywanie danych aplikacji w przeglądarce w formie klucz-wartość. Web App Manifest jest plikiem JSON, który zawiera metadane aplikacji, takie jak nazwa, adresy ikon, adres startowy czy preferowany sposób wyświetlania. Przeglądarki wykorzystują informacje zawarte w tym pliku do instalacji aplikacji w systemie użytkownika i wyświetlenia jej w sposób zbliżony do tego, w jaki wyświetla aplikacje natywne. Zainstalowana aplikacja PWA jest uruchamiana w dedykowanym oknie, które ukrywa elementy interfejsy typowe dla przeglądarki internetowej.

Progresywne aplikacje webowe zapewniają płynniejsze doświadczenie niż klasyczne strony internetowe. Interfejs użytkownika oraz jego logika mogą zostać w pełni załadowane przy uruchomieniu aplikacji. Wszelkie zmienne dane, które ma prezentować aplikacja, są ładowane asynchronicznie, dzięki czemu nie trzeba za każdym razem przesyłać wraz z nimi informacji określających sposób ich wyświetlania (np. dokumenty HTML, skrypty JavaScript). Poprzez wykorzystanie mechanizmów pamięci trwałej przeglądarki takich jak Cache API aplikacja może również działać w trybie offline.

WebAssembly

WebAssembly (często skracane jako WASM) jest otwartym standardem definiującym binarny format instrukcji wykonywanych przez maszynę wirtualną opartą na stosie, która może być uruchomiona w przeglądarce internetowej (standard nie ogranicza się do technik internetowych) [44]. Został zaprojektowany w celu zapewnienia wysokowydajnego, przenośnego i niskopoziomowego środowiska wykonawczego. Standard jest agnostyczny względem języków programowania i sprzętu komputerowego, co czyni go uniwersalnym, przenośnym rozwiązaniem i pozwala na wieloplatformowe zastosowanie. Definiuje izolowane, bezpieczne dla pamięci środowisko uruchomieniowe, które może działać jako samodzielna maszyna wirtualna lub zostać osadzone w innym środowisku (np. silniku przeglądarki internetowej). Jedną z głównych zalet WebAssembly jest jego wydajność. Format jest zaprojektowany tak, aby był wykonywany z prędkością zbliżoną do natywnej, co czyni go dobrze przystosowanym do aplikacji wymagających wysokiej wydajności lub wykonujących obliczenia o dużej skali. Ponadto, format WebAssembly jest kompaktowy, co pozwala na jego szybką transmisję i łatwe przechowywanie w formie zasobów statycznych, co ma szczególne znaczenie w przypadku wykorzystywania go w aplikacjach internetowych.

Blazor

Blazor to platforma programistyczna służącą do tworzenia interaktywnych aplikacji klienckich z oparta na platformie .NET oraz języku C# [45]. Jest to relatywnie nowe rozwiązanie, wprowadzone na rynek w 2018 roku przez firmę Microsoft, która dalej go rozwija i zapewnia długoterminowe wsparcie. Jest częścią opisanej w podrozdziale 3.3.2 platformy ASP.NET core. Główną zaletą tego rozwiązania w przypadku jednociennego wykorzystania platformy .NET po stronie serwerowej, jest możliwość używania tego samego języka programowania na każdym etapie tworzenia aplikacji, od tworzenia logiki biznesowej po interfejs użytkownika. Możliwe jest również wykorzystanie tych samych narzędzi i bibliotek oraz współdzielenie kodu między aplikacją kliencką a serwerową. Upraszczają to znaczaco proces wytwarzania aplikacji. Blazor pozwala również na komunikację z serwerem w czasie rzeczywistym za pomocą biblioteki SignalR.

Platforma pozwala na tworzenie interfejsu w sposób zorientowany na komponenty z użyciem składni Razor (połączenie składni HTML z językiem C#). Komponenty są niezależnymi częściami interfejsu użytkownika, które mogą być wykorzystywane wielokrotnie w obrębie aplikacji. Każdy komponent posiada swój własny widok, logikę i stan. Komponenty mogą być zagnieżdżane w innych komponentach, tworząc drzewiastą hierarchię, co pozwala na łatwiejsze tworzenie złożonych interfejsów użytkownika. Pomaga to również w organizacji kodu i zwiększa możliwość jego ponownego wykorzystania. Komponenty mogą przyjmować parametry od komponentów nadzorczych i emitować do nich zdarzenia. Dzięki temu możliwy jest dwukierunkowy przepływ danych w ich hierarchii. W przeciwieństwie do aplikacji bazujących na języku JavaScript, aplikacje Blazor WebAssembly nie mają bezpośredniego dostępu do interfejsu przeglądarki. W tym celu Blazor udostępnia mechanizm wywoływanego kodu JavaScript z poziomu kodu .NET i vice versa. Pozwala to na współpracę aplikacji z interfejsem programistycznym przeglądarki (np. dostęp do urządzeń peryferyjnych, takich jak kamera czy mikrofon) oraz integrację z istniejącymi bibliotekami i narzędziami napisanymi w języku JavaScript.

Wspierane są dwa podstawowe modele serwowania: Blazor WebAssembly oraz Blazor Server [46]. W przypadku modelu Blazor WebAssembly aplikacja kliencka jest pobierana w formie zasobów statycznych do przeglądarki użytkownika, gdzie jest uruchamiana w środowisku WebAssembly. Po pobraniu aplikacji komunikacja z jakimkolwiek serwerem jest opcjonalna i może odbywać się za pomocą dowolnego protokołu sieciowego obsługiwanej przez przeglądarkę (np. HTTP lub WebSocket). Kod środowiska .NET w aplikacji Blazor WebAssembly może być wykonywany na jeden z dwóch sposobów, z użyciem interpretera lub komplikacji „ahead-of-time” (AOT), który może określić programista. W przypadku użycia interpretera kod aplikacji wraz z wykorzystywanym podziobrem środowiska .NET jest kompilowany do języka pośredniego .NET – CIL (ang. Common Intermediate Language). Wraz z kodem aplikacji i innymi typowymi zasobami (np. pliki HTML, CSS, JavaScript, obrazami) do przeglądarki pobierany jest interpreter języka CIL w formacie WebAssembly, który pełni dla niego rolę środowiska uruchomieniowego. Odbywa się to więc w sposób analogiczny jak w przypadku klasycznych aplikacji platformy .NET. Istotną różnicą jest jednak fakt, że kod CIL aplikacji Blazor jest interpretowany. Standardowo kod źródłowy aplikacji platformy .NET jest kompilowany do CIL, który następnie jest przekształcany do kodu maszynowego metodą just-in-time przez środowisko uruchomieniowe platformy – CLR (Common Language Runtime). W przypadku komplikacji AOT znacząca część kodu aplikacji i platformy .NET jest kompilowana bezpośrednio do formatu binarnego WebAssembly wykonywanego przez silnik przeglądarki, co pozwala na ograniczenie wykorzystania interpretera CIL. Przekłada się to na lepszą wydajność aplikacji kosztem dłuższego czasu jej pierwszego ładowania (kod WebAssembly jest zazwyczaj bardziej obszerny niż kod CIL). Ograniczeniem tego modelu jest brak obsługi niektórych funkcjonalności interfejsu plat-

formy .NET, takich jak interoperacyjność z kodem niezarządzanym przez platformę, modyfikacja kodu poprzez mechanizm refleksji oraz niektórych zaawansowanych mechanizmów wielowątkowości.

W przypadku drugiego modelu kod aplikacji jest wykonywany na serwerze, a część aplikacji działającej w przeglądarce odpowiada za wyświetlanie interfejsu użytkownika oraz przekazywanie zdarzeń generowanych przez użytkownika do serwera, który przetwarza je i zwraca zaktualizowany widok interfejsu. Model ten w przeciwieństwie do WebAssembly zapewnia pełną funkcjonalność środowiska .NET oraz brak konieczności tworzenia publicznego API w serwerze aplikacji. Wymaga jednak stałego połączenia z serwerem ASP.NET Core ze względu na wykorzystanie do komunikacji protokołu WebSocket. W obu przypadkach Blazor obsługuje ten sam model programowania, programiści mogą więc tworzyć kod, który bez większych modyfikacji będzie działał niezależnie od wybranego sposobu serwowania (z wyjątkiem wspomnianych wcześniej nielicznych ograniczeń modelu WebAssembly).

3.3.2. Serwer aplikacji

Serwer aplikacji został oparty o platformę ASP.NET Core (wersja 7.0.5) ze względu rozbudowany zestaw funkcjonalności, wydajność oraz zalety wykorzystywanego w nim języka programowania – C# takie jak silne typowanie i nowoczesna składnia. Wykorzystanie platformy ASP.NET Core pozwoliło również na wykorzystanie silnie z nim sprzężonej platformy Blazor w celu implementacji aplikacji klienckiej, co pozwoliło zminimalizować ilość powielonego kodu i przyspieszyć proces wytwarzania oprogramowania.

ASP.NET Core

ASP.NET Core jest szeroko stosowaną w branży otwartoźródłową platformą do budowy nowoczesnych aplikacji i serwisów internetowych. Jest oparta na platformie programistycznej .NET (nazywanej wcześniej .NET Core). Pozwala to na uruchomienie jej na systemach operacyjnych Windows, Linux i MacOS oraz na tworzenie aplikacji w językach takich jak C#, F# i Visual Basic [47].

Architektura ASP.NET Core jest wysoce modularna, co pozwala na prostą konfigurację oraz łatwe dodawanie i usuwanie funkcjonalności w zależności od potrzeb projektu. Dostępna jest szeroka gama bibliotek i komponentów, zarówno oficjalnych, jak i tworzonych przez społeczność. ASP.NET Core oferuje wiele wbudowanych funkcjonalności, takich jak wstrzykiwanie zależności, obsługa uwierzytelniania i autoryzacji, walidacja modeli danych, routing czy mechanizmy komunikacji z aplikacjami klienckimi w czasie rzeczywistym. Pozwala na tworzenie dynamicznych aplikacji i stron internetowych z wykorzystaniem wzorca MVC (Model-View-Controller) oraz modelu programowania Razor Pages, a także interfejsów webowych. Umożliwia również tworzenie interaktywnych aplikacji jednostronnicowych przy użyciu, opisanej wcześniej w podrozdziale 3.3.1, platformy Blazor. Wszystkie te modele łączyć w ramach jednej aplikacji w zależności od potrzeb. Żądania HTTP są domyślnie obsługiwane przez lekki, wbudowany serwer Kestrel. Przetwarzanie żądań odbywa się w sposób potokowy i asynchroniczny, co przekłada się na wysoką wydajność i skalowalność aplikacji. Żądania i odpowiedzi przetwarzane są z wykorzystaniem wielu warstw oprogramowania pośredniego (middleware), które określają sekwencje operacji, które mają zostać wykonane w celu obsługi żądania. Wywołania kolejnych warstw są w pełni asynchroniczne, dzięki czemu możliwa jest jednocześnie obsługa wielu żądań naraz. Oprogramowanie pośrednie to moduły, które można umieścić w potoku przetwarzania żądań HTTP w celu obsługi konkretnych aspektów przetwarzania tych żądań. Mogą one przechwytywać, analizować, modyfikować lub przerywać przetwarzanie żądań HTTP oraz generować odpowiedzi. Domyślnie, platforma ASP.NET Core zawiera oprogramowanie pośrednie odpowiedzialne za takie funkcje jak: obsługa błędów, rejestrowanie zdarzeń, serwowanie plików statycznych czy obsługę kontrolerów MVC. ASP.NET Core pozwala na łatwą integrację z innymi technologiami platformy .NET, takimi jak wspomniany wcześniej Blazor,

czy opisana dalej biblioteka Entity Framework Core. Dzięki temu cała aplikacja może być tworzona w jednym, spójnym środowisku, co ułatwia proces wytwarzania oprogramowania. Platforma ASP.NET wspiera również konteneryzację, co pozwala na uproszczenie procesu wdrażania aplikacji i zarządzania infrastrukturą serwerową.

Entity Framework Core

Entity Framework Core (EF Core) to nowoczesna, otwartoźródłowa biblioteka do obsługi baz danych dla platformy .NET. Umożliwia ona programistom pracę z wieloma różnymi systemami zarządzania bazami danych w sposób obiektowy bez konieczności ręcznego pisania zapytań SQL [48]. Pozwala to na abstrakcję kodu aplikacji od specyfiki konkretnej bazy danych. Entity Framework Core odwzorowuje tabele i relacje bazy danych na obiekty .NET i odwrotnie za pomocą mechanizmu odwzorowywania obiektowo-relacyjnego (ang. Object-Relational Mapping, ORM), pozwalając na uniknięcie ręcznej implementacji warstwy dostępu do danych. EF core promuje podejście *code-first*, w którym programiści mogą definiować encje i relacje bazy danych w kodzie źródłowym aplikacji, a tworzeniem i aktualizacjami schematu bazy danych zajmują się biblioteka. Innymi słowy, EF Core pozwala na w pełni automatyczne zarządzanie schematem bazy danych, czyli tworzenie, modyfikowanie i usuwanie tabel, indeksów i innych elementów struktury bazy danych, na podstawie zdefiniowanych w kodzie źródłowym aplikacji klas modelu domeny. Biblioteka dostarcza również mechanizmy migracji i wersjonowania schematów bazy danych. Migracje pozwalają na automatyczne przekształcanie schematu bazy danych pomiędzy różnymi wersjami aplikacji. Takie podejście zapewnia, że logika i modele danych w aplikacji są zawsze aktualne względem schematu bazy danych, co czyni proces tworzenia aplikacji bardziej wydajnym i zmniejsza prawdopodobieństwo wystąpienia błędów. Jedną z zalet tej biblioteki jest obsługa zapytań LINQ (Language Integrated Query), które pozwala programistom pisać silnie typowane, czytelne zapytania bezpośrednio w języku C#. Biblioteka tłumaczy je na zapytania SQL, co zapewnia wysoki poziom abstrakcji i eliminuje potrzebę pisania przez programistów podatnego na błędy kodu SQL.

Wybór Entity Framework Core jako narzędzia do zarządzania danymi w projekcie jest uzasadniony ze względu na jego integrację z platformą .NET, co pozwala na ujednolicenie stosu technologicznego. Ponadto, EF Core jest elastyczny, wydajny i obsługuje wiele systemów baz danych, dając możliwość łatwej zmiany dostawcy bazy danych w przyszłości, jeśli zajdzie taka potrzeba. Dzięki temu aplikacja może być w prosty sposób dostosowana do różnych scenariuszy wdrożeniowych.

SignalR

Platforma ASP.NET Core pozwala na łatwą dwukierunkową komunikację w czasie rzeczywistym z aplikacjami klienckimi za pomocą otwartoźródłowej biblioteki SignalR [49]. SignalR działa w oparciu o mechanizm zdalnego wywoływanie procedur (ang. Remote Procedure Call, RPC) pozwalający na wzajemne wywoływanie metod między klientem a serwerem aplikacji umożliwiając tym samym szybką komunikację w obu kierunkach.

Komunikacja odbywa się przez jedną z kilku dostępnych technik, która jest automatycznie wybierana w zależności od możliwości sieciowych klienta aplikacji. Domyślnym sposobem jest wykorzystanie protokołu WebSocket pozwalającego na dwukierunkową komunikację w sposób ciągły za pomocą pojedynczego połączenia TCP. W przypadku braku możliwości utworzenia połączenia WebSocket SignalR próbuje wykorzystać technikę *Server-Sent Events* lub *Long Polling* dla komunikacji w kierunku serwera w połączeniu ze zwykłymi żądaniami HTTP dla komunikacji w kierunku klienta. Technika *Server-Sent Events* polega na utworzeniu żądania HTTP, które serwer utrzymuje otwarte przez długi czas i asynchronicznie wysyła do klienta kolejne porcje danych. *Long Polling* polega na otwarciu żądania HTTP

przez klienta, które serwer utrzymuje otwarte do momentu, aż nie pojawią się nowe dane do przesłania bądź nastąpi przekroczenie czasu na odpowiedź. Po zakończeniu żądania natychmiast jest tworzone kolejne w celu utrzymania połączenia.

3.3.3. Dostawca tożsamości

Do realizacji komponentu Dostawcy tożsamości wykorzystano gotowy otwartoźródłowy projekt – Keycloak [50] w wersji 21.1.0. Wybór ten jest uzasadniony koniecznością zapewnienia odpowiednio wysokiego poziomu bezpieczeństwa oraz ograniczenia ryzyka wystąpienia błędów i podatności.

Dostawca tożsamości (Identity Provider, IDP) to usługa lub system odpowiedzialny za scentralizowane tworzenie, zarządzanie cyfrową tożsamością użytkowników oraz ich uwierzytelnianie i autoryzację w innych systemach, usługach lub aplikacjach. Podstawową funkcją IDP jest uwierzytelnianie użytkowników poprzez weryfikację ich tożsamości przy użyciu zestawu unikalnych danych uwierzytelniających, takich jak nazwa użytkownika i hasło, lub innych mechanizmów uwierzytelniania, takich jak dane biometryczne, hasła jednorazowe czy tokeny. W kontekście aplikacji internetowych IDP odgrywają kluczową rolę we wdrażaniu rozwiązań *Single Sign-On* (SSO). SSO umożliwia użytkownikom dostęp do wielu usług lub aplikacji jednocześnie, używając tych samych poświadczeń, co upraszcza proces logowania. Przedsiębiorstwa, takie jak Google, Microsoft, Facebook czy Apple, udostępniają własne usługi IDP dla aplikacji innych firm, co zwiększa komfort użytkowników poprzez ograniczenie potrzeby pamiętania wielu haseł.

Istnieją także rozwiązania pozwalające na stworzenie własnej usługi IDP. Mogą to być biblioteki ułatwiające ich implementację, takie jak IdentityServer czy OpenIdict, lub gotowe produkty. Gotowe rozwiązania możemy dodatkowo podzielić na takie, które są wdrażane we własnym zakresie, na przykład Keycloak, oraz takie, które są dostarczane jako usługa (na przykład Auth0, Azure Active Directory i podobne). Zamiast implementować własne mechanizmy uwierzytelniania, aplikacje mogą polegać na sprawdzonym gotowym rozwiązaniu lub zewnętrznej usłudze, co pozwala na szybsze wdrożenie, mniejsze koszty utrzymania oraz niższe ryzyko wystąpienia błędów i podatności. Wykorzystanie dostawców tożsamości może przyczynić się do zwiększenia bezpieczeństwa, ułatwienia zarządzania tożsamościami oraz usprawnienia procesów uwierzytelniania i autoryzacji w różnych systemach.

Keycloak

Keycloak jest dostępny na wolnej licencji Apache License 2.0. Rozwój projektu jest aktywnie wspierany przez firmę Red Hat. Keycloak został stworzony na podstawie kilku kluczowych standardów i protokołów zapewniających bezpieczeństwo i interoperacyjność w zakresie zarządzania tożsamością. Ze względu na wykorzystanie standardowych protokołów, wybór ten pozwoli również na ewentualną zamianę własnego serwera IDP na inny, bez konieczności zmiany kodu aplikacji lub integracji systemu z zewnętrznym dostawcą tożsamości, na przykład w celu integracji z innymi systemami. Najważniejsze z nich, które są wykorzystywane w projekcie to: OAuth 2.0, OpenID Connect oraz Json Web Token (JWT).

OAuth 2.0

OAuth 2.0 to otwarty standard protokołu autoryzacji umożliwiający aplikacjom klienckim uzyskanie dostępu do określonych zasobów użytkownika w innej aplikacji, serwisie lub systemie [51]. Dostęp ten odbywa się bez konieczności przekazywania poświadczeń uwierzytelniających (np. loginu i hasła) do samej aplikacji klienckiej. Zamiast tego wykorzystuje ona kody autoryzacji i tokeny dostępu (ang. access tokens) uzyskane w wyniku autoryzacji użytkownika przez dostawcę tożsamości.

Typowy proces autoryzacji aplikacji klienckiej z użyciem IDP przebiega następująco. Aplikacja kliencka przekierowuje użytkownika do IDP, który go uwierzytelnia. W przekierowaniu przekazywany jest identyfikator aplikacji klienckiej oraz zakresy atrybutów (ang. scopes) określające dane, które mają zawierać później zwarcane tokeny (np. role użytkownika w systemie). Po pomyślnym uwierzytelnieniu IDP przekierowuje użytkownika z powrotem do aplikacji klienckiej wraz z kodem autoryzacji umieszczonym w adresie URL. Następnie aplikacja kliencka komunikuje się bezpośrednio z IDP w celu wymiany kodu dostępu na token dostępu. Opcjonalnie z tokenem dostępu zwarcany jest token odświeżający (ang. refresh token), który pozwala na uzyskanie nowego tokenu dostępu bez konieczności ponownego uwierzytelniania użytkownika (np. po jego wygaśnięciu).

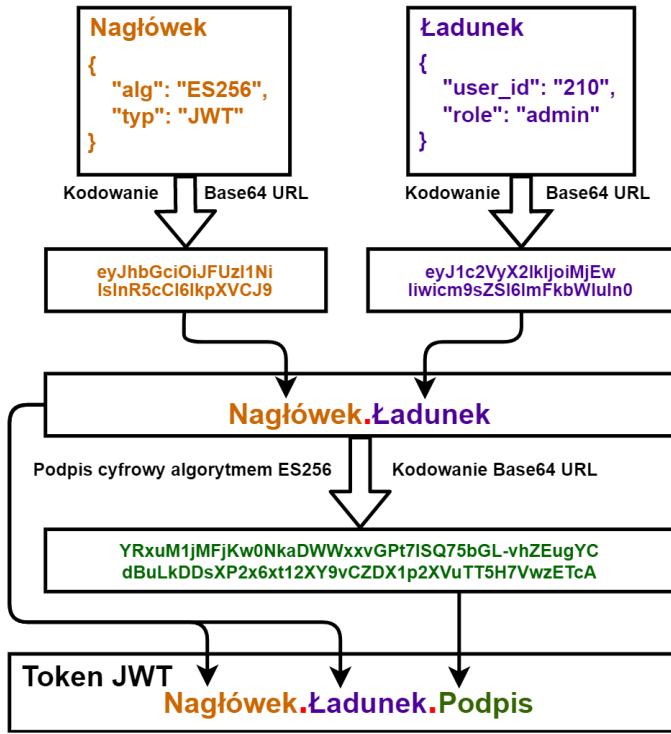
Tokeny te są zazwyczaj krótkotrwałe i wygasają po określonym czasie. Zapewnia to większe bezpieczeństwo, ponieważ w przypadku ich przechwycenia przez osoby trzecie, mogą one zostać wykorzystane tylko przez krótki czas. Są również opatrzone podpisem cyfrowym dostawcy tożsamości, co pozwala na weryfikację ich autentyczności przez system docelowy oraz uniemożliwia ich modyfikację i podrabianie przez osoby trzecie (poza sytuacją, w której wykradziony został klucz prywatny IDP). W celu uzyskania dostępu do chronionych zasobów aplikacja kliencka wysyła żądanie do serwera docelowego, załączając do niego token dostępu. Serwer docelowy weryfikuje poprawność tokenu i zwraca żądane zasoby. W przypadku gdy token dostępu wygaśnie, aplikacja kliencka może wykorzystać token odświeżający do uzyskania nowego tokenu dostępu bez konieczności ponownego uwierzytelniania użytkownika. W celu zwiększenia bezpieczeństwa cała komunikacja powinna odbywać się za pomocą protokołu HTTPS, który zapewnia szyfrowanie danych. Ma to na celu uniemożliwić przechwycenie kodu autoryzacji lub tokenów przez osoby trzecie.

OpenID Connect

OpenID Connect (OIDC) jest rozszerzeniem protokołu OAuth 2.0 o warstwę uwierzytelniania [52]. Protokół OAuth skupia się głównie na autoryzacji aplikacji klienckiej przesyłanych do serwera zasobów lub innego systemu za pośrednictwem dostawcy tożsamości, natomiast OpenID Connect definiuje sposób, w jaki aplikacja kliencka może uzyskać informacje o tożsamości użytkownika. Kluczowym elementem OIDC zawierającym wspomniane dane, jest token ID. Aplikacja kliencka może go otrzymać od IDP, wykorzystując opisany wcześniej kod autoryzacji. Przy przekierowaniu użytkownika do IDP, aplikacja kliencka musi w tym celu uwzględnić odpowiedni zakres atrybutów (domyślnie „openid”). Token ID może zawierać informacje takie jak nazwa użytkownika, adres e-mail, role użytkownika w systemie itp. Podobnie jak token dostępu jest podpisany cyfrowo przez IDP, co zapewnia jego autentyczność i uniemożliwia modyfikację przez osoby trzecie. Token ID jest przeznaczony do użytku wewnętrz aplikacji klienckiej i nie powinien być przesyłany wraz z żdaniami do serwerów.

JSON Web Token

JSON Web Token (JWT) jest standardem definiującym format tokenów wykorzystywanych do transferu roszczeń (ang. claims) pomiędzy zainteresowanymi stronami [53]. Standard ten jest często wykorzystywane w implementacjach protokołów uwierzytelniania i autoryzacji, takich jak OAuth 2.0 [54] i OpenID Connect [52]. Najczęściej wykorzystywany typ tokenów JWT to token podpisany cyfrowo, służący do weryfikacji integralności i autentyczności danych w nim zawartych. Rzadziej stosuje się tokeny zaszyfrowane, które mają za zadanie ukrycie zawartości przed stronami trzecimi. Token podpisany cyfrowo składa się z trzech części: nagłówka, ładunku (ang. payload) oraz podpisu. Część nagłówkowa zawiera informacje o typie tokenu i algorytmie wykorzystanym do jego cyfrowego podpisu. Część ładunku zawiera roszczenia, czyli zestaw stwierdzeń na temat podmiotu, którego dotyczy token. Zamieszczony



Rysunek 3.5: Przykładowy proces tworzenia tokenu JWT

jest tam również czas wygaśnięcia tokenu. Ostatnia część zawiera podpis cyfrowy tokenu przez podmiot wystawiający go, który ma zapewniać autentyczność, integralność i niezaprzecjalność zawartych w nim informacji. Autentyczność odnosi się do pewności co do źródła pochodzenia informacji. W kontekście JWT autentyczność zapewnia, że token pochodzi od oczekiwanej nadawcy. Integralność oznacza, że dane (w przypadku JWT – nagłówek i ładunek) nie zostały w żaden sposób naruszone od momentu ich wygenerowania przez podmiot wystawiający, zabezpiecza to przed nieautoryzowaną modyfikacją przesyłanych informacji. Niezaprzecjalność oznacza, że podmiot wystawiający nie może zaprzeczyć faktu wystawienia przez niego tokenu. Oprócz tego tokeny JWT są bezstanowe, nie ma potrzeby przechowywania żadnych informacji o konkretnym z nich, aby móc go zweryfikować bądź wykorzystać. Wszystkie informacje potrzebne do tego celu są zawarte w samym tokenie, jest on więc samowystarczalny. Wadą tej cechy jest jednak brak możliwości wycofania tokenu przed upływem jego ważności, co kompensuje się jego krótkim czasem ważności.

JWT jest oparty o kodowanie Base64 w wariancie bezpiecznym dla adresów URL, a także jak sama nazwa wskazuje format JSON (JavaScript Object Notation). Base64 to sposób kodowania danych binarnych w formie tekstu. JSON jest lekkim, czytelnym dla człowieka, niezależnym od języka programowania formatem służącym do przesyłania i przechowywania ustrukturyzowanych danych za pomocą par klucz-wartość. Konstrukcja tokenu JWT polega na zapisie nagłówka i ładunku w formacie JSON, a następnie zakodowaniu ich binarnej reprezentacji z użyciem schematu Base64. Wynikowe ciągu znaków są łączone za pomocą znaku kropki. Ostatnim krokiem jest dodanie podpisu. W tym celu uzyskany ciąg znaków zostaje podpisany cyfrowo przez podmiot wystawiający zazwyczaj z użyciem kriptografii asymetrycznej, którą cechuje istnienie pary kluczy – publicznego i prywatnego. Wiadomość zaszyfrowaną jednym z nich można odszyfrować tylko przy użyciu drugiego. W celu stworzenia podpisu cyfrowego podmiot wystawiający token szyfruje zakodowane i łączone z sobą nagłówek i ładunek z użyciem klucza prywatnego. Wynik działania algorytmu szyfrującego jest kodowany również z użyciem Base64, a następnie dołączony do ciągu dając w wyniku token o formacie „nagłówek.ładunek.podpis”. W celu weryfikacji to-

kenu odbiorca deszyfruje podpis z użyciem klucza publicznego wystawcy i porównuje go z jawną częścią tokenu. Jeśli są one identyczne, oznacza to, że token nie został po drodze zmodyfikowany i pochodzi od wystawcy. Przykładowy proces generowania tokenu JWT został przedstawiony na rysunku 3.5. Możliwe jest też podpisanie tokenu za pomocą kryptografii symetrycznej, w której szyfrowanie i deszyfrowanie odbywa się za pomocą jednego tajnego klucza znanego zarówno wystawcy, jak i odbiorcy tokenu. Tak podpisany token nie gwarantuje jednak niezaprzeczalności, ponieważ odbiorca jest w stanie wygenerować taki sam podpis jak wystawca. Użycie klucza symetrycznego jest szczególnie powszechne, gdy weryfikator i wystawca są tym samym systemem, ponieważ eliminuje to potrzebę zarządzania parą kluczy. JWT można dodatkowo zabezpieczyć poprzez zaszyfrowanie podpisanego tokenu, aby zapewnić jego poufność. W większości przypadków poufność jest wystarczającym stopniu zapewniana przez protokół HTTPS, który zapewnia szyfrowanie wszystkich danych przesyłanych pomiędzy stronami.

3.3.4. Platforma wideokonferencyjna

Możliwość realizacji komunikacji na odległość w czasie rzeczywistym jest jednym z kluczowych aspektów przedstawianego projektu. W celu zapewnienia odpowiedniej jakości połączeń video i audio oraz dostępności niezależnie od urządzenia użytkowników, zdecydowano się na wykorzystanie dojrzałego, otwartoźródłowego rozwiązania – Jitsi Meet [55] w wersji 8615, które zintegrowano z resztą systemu. Wybór ten jest uzasadniony brakiem wystarczających zasobów na stworzenie własnego rozwiązania wideokonferencyjnego, który zapewniłby jednocześnie odpowiednią niezawodność i jakość połączeń. Biorąc pod uwagę szerokie spektrum wymagań systemu oraz ograniczone zasoby, w realizacji projektu kluczowe było podejście efektywne i pragmatyczne.

Jitsi Meet

Jitsi Meet jest oparty o technologię WebRTC. WebRTC jest otwartym standardem pozwalającym na komunikację w czasie rzeczywistym w przeglądarce internetowej bez konieczności instalacji dodatkowego oprogramowania. Jest to technologia, która jest stale rozwijana i wspierana przez największych dostawców przeglądarek internetowych, takich jak Google, Mozilla, Microsoft i Apple. Rozwiązanie to pozwala na korzystanie z publicznie dostępnej instancji serwera, jednak w przypadku tego projektu zdecydowano się na wdrożenie własnej instancji w celu uzyskania większej elastyczności i kontroli. Wykorzystanie Jitsi Meet pozwoliło na realizowanie w systemie bezpiecznych połączeń video (połączenia są szyfrowane) o wysokiej jakości. Istotnym atutem tego rozwiązania jest jego elastyczność, która pozwoliła na dostosowanie funkcjonalności do specyficznych potrzeb projektu. Rozwiązanie to pozwoliło na osadzenie klienta platformy wideokonferencyjnej wewnętrz aplikacji klienckiej, personalizację interfejsu użytkownika, integrację z systemem autoryzacji, magazynem plików i kolaboracyjnym edytorem audio aplikacji.

Dodatkowo rozwiązanie można skalować poprzez zwiększenie liczby instancji usług składających się na Jitsi Meet, co pomaga sprostać rosnącym wymaganiom w przypadku zwiększenia liczby użytkowników i jednoczesnych połączeń. Jako rozwiązanie webowe nie wymaga instalacji dodatkowego oprogramowania, co ułatwia szybkie dołączanie do konferencji z dowolnego urządzenia z dostępem do internetu i przeglądarki. Jitsi Meet oferuje szeroki zakres funkcjonalności, takich jak współdzielenie ekranu, czat tekstowy, możliwość nagrywania spotkań, wirtualna tablica, integracja z kalendarzami czy możliwość synchronicznego odtwarzania multimedialnych w oryginalnej jakości. Dzięki temu, Jitsi Meet może być wykorzystywany w większości typowych scenariuszy wideokonferencyjnych. Użytkownicy mogą również dołączać do spotkań za pomocą dedykowanej aplikacji mobilnej. Własne wdrożenia Jitsi Meet działają z oficjalną aplikacją mobilną, choć istnieje również możliwość stworzenia własnej aplikacji mobilnej z

wykorzystaniem udostępnionych narzędzi programistycznych. Jitsi Meet jest aktywnie rozwijany przez społeczność oraz wspierany przez firmę 8x8, która dostarcza oparte na tym projekcie usługi.

Jitsi Broadcasting Infrastructure

Jitsi Broadcasting Infrastructure (Jibri) to komponent projektu Jitsi Meet, który umożliwia nagrywanie oraz transmisję na żywo prowadzonych w nim spotkań. Jibri działa jako osobna usługa, którą można skalować horyzontalnie w zależności od potrzeb. W miarę wzrostu liczby jednocześnie prowadzonych sesji nagrywania lub transmisji, do infrastruktury systemu można dodawać kolejne instancje Jibri. Integracja Jibri z Jitsi Meet jest stosunkowo prosta i sprowadza się do odpowiedniej konfiguracji obu komponentów. Po prawidłowej konfiguracji, użytkownicy Jitsi Meet zyskują dostęp do przycisków umożliwiających rozpoczęcie nagrywania lub transmisji na żywo.

3.3.5. Baza danych i magazyn plików

Do przechowywania danych aplikacji i dostawcy tożsamości wykorzystano system zarządzania relacyjnymi bazami danych PostgreSQL [56] w wersji 15.0. Wybór ten jest uzasadniony wydajnością, otwartością kodu źródłowego oraz wsparciem przez bibliotekę Entity Framework Core. Do zaimplementowania magazynu plików wykorzystano lokalny system plików serwera aplikacji. Pozwoliło to na prostą integrację z serwerem aplikacji i uproszczenie procesu wdrożenia kosztem mniejszej skalowalności i elastyczności systemu.

3.4. Implementacja

W tym podrozdziale przedstawiono implementację komponentów wysokopoziomowej architektury proponowanego systemu. Obejmuje ona opis realizacji aplikacji klienckiej i serwera aplikacji oraz ich wewnętrznej architektury. Omówiono również integrację z dostawcą tożsamości i platformą wideokonferencyjną oraz ich konfigurację niezbędną do poprawnego działania systemu. Komponenty zaimplementowano zgodnie z wymaganiami i modelem domeny określonymi w podrozdziale 3.1 oraz założeniami architektury systemu opisanymi w podrozdziale 3.2. Kod źródłowy systemu zaimplementowanego w ramach niniejszej pracy dostępny jest w repozytorium na platformie GitHub pod adresem <https://github.com/Etav99/Concerto>

3.4.1. Dostawca tożsamości

Rolę komponentu dostawcy tożsamości w systemie pełni uruchamiana w jego obrębie instancja serwera Keycloak opisanego w podrozdziale 3.3.3. Pierwszy krok konfiguracji serwera Keycloak to stworzenie dla systemu nowego obszaru (ang. realm) o nazwie *concerto*, będący zbiorem użytkowników, ich danych uwierzytelniających, klientów oraz konfiguracji związanych z bezpieczeństwem systemu. W obszarze włączono funkcje rejestracji użytkowników, resetowania hasła oraz weryfikacji adresu e-mail. Skonfigurowano również dostęp do serwera poczty, który jest wymagany do automatycznego wysyłania wiadomości związanego z IDP, takich jak linki do resetowania hasła czy weryfikacji adresu e-mail.

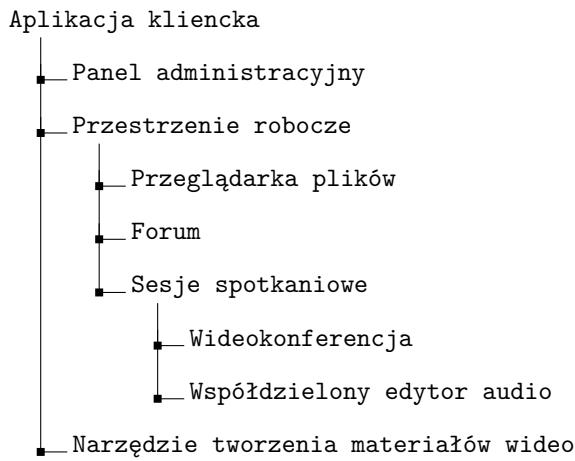
W obrębie obszaru zdefiniowano trzy role użytkowników: *admins*, *moderators* oraz *unverified*. Odpowiadają one rolom użytkowników określonych podczas analizy wymagań systemu (podrozdział 3.1). Rola *admins* reprezentuje administratora systemu, *moderators* – moderatora, *unverified* – użytkownika niezweryfikowanego przez administratora. Rola użytkownika nie posiada swojej reprezentacji w postaci roli w obszarze, wynika ona z braku posiadania roli *unverified*.

W obszarze zdefiniowano również dwa klienty: *concerto-client* z typem dostępu *public* oraz *concerto-*

server z typem dostępu *confidential*. Klient w kontekście dostawcy tożsamości jest dowolnym programem lub aplikacją, korzystającą z jego usług w ramach danego obszaru. Ma on określony zakres (ang. scope) dostępu do zasobów obszaru. Klienci z typem dostępu *public* są dostępne dla każdego, klienci typu *confidential* wymagają posiadania tajnego klucza dostępu. Klient *concerto-client* pozwala na uwierzytelnianie użytkowników w aplikacji klienckiej za pomocą tokenów ID oraz pobieranie tokenów dostępu przekazywanych do serwera aplikacji w celu autoryzacji żądań. Zakres dostępu klienta został skonfigurowany tak aby tokeny zawierały podstawowe dane użytkownika, takie jak nazwa, imię, nazwisko, adres e-mail oraz rolę użytkownika w systemie. Klient *concerto-server* jest odpowiedzialny za uwierzytelnianie i autoryzację serwera aplikacji w IDP. Za jego pomocą serwer aplikacji korzysta z API dostawcy tożsamości w celu weryfikacji tokenów JWT oraz zarządzania użytkownikami. Zakres dostępu klienta obejmuje uprawnienia potrzebne do zarządzania użytkownikami oraz ich rolami w obszarze *concerto* w celu realizacji funkcjonalności panelu administracyjnego.

3.4.2. Aplikacja kliencka

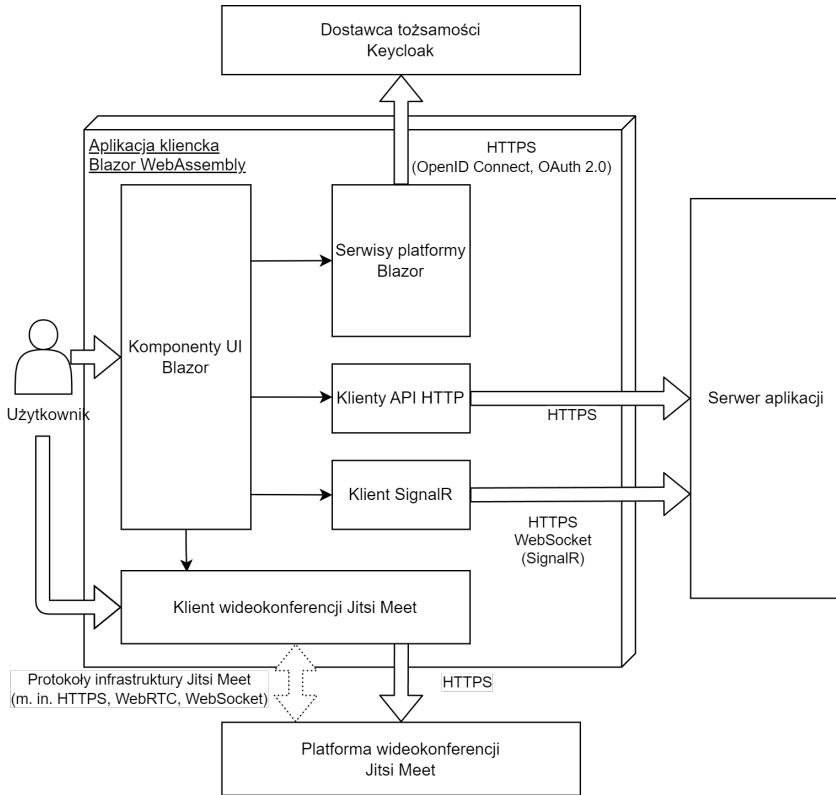
Aplikacja kliencka została zaimplementowana jako progresywna aplikacja internetowa z wykorzystaniem platformy *Blazor* w wariantie *WebAssembly*, rozwiązania te zostały opisane w sekcji 3.3.1. W odniesieniu do architektury systemu opisanej w sekcji 3.2 aplikacja kliencka realizuje warstwę prezentacji oraz częściowo warstwę logiki prezentacji. Z punktu widzenia interfejsu użytkownika oraz funkcjonalności w aplikacji klienckiej możemy wyróżnić elementy logiczne o następującej strukturze:



Elementy te są zrealizowane za pomocą komponentów Blazor (warstwa prezentacji) oraz klientów API i serwisów platformy Blazor dostarczanych do nich poprzez wstrzykiwanie zależności (warstwa logiki prezentacji). W celu integracji z bibliotekami i komponentami spoza platformy oraz z interfejsem przeglądarki internetowej, komponenty Blazor wykorzystują dostępny w platformie mechanizm *JSInterop* pozwalający na dwustronną komunikację z kodem JavaScript. Realizacja poszczególnych części została opisana w dalszych sekcjach.

Diagram 3.6 przedstawia ogólną architekturę aplikacji klienckiej. Szerokie strzałki z etykietami ozaczają komunikację sieciową z użyciem protokołów określonych w etykietach. Strzałki cienkie ozaczają komunikację w obrębie aplikacji klienckiej. Kierunek strzałek określa stronę inicjującą komunikację i kierunek zależności pomiędzy elementami. Wyjątek stanowi szeroka strzałka wyróżniona przerywaną linią stanowiącą wyjątek i ozaczającą wewnętrzną komunikację w obrębie infrastruktury Jitsi Meet z jej osadzonym w aplikacji klientem.

Nawigacja w aplikacji jest realizowana przez wbudowany w platformę *Blazor* mechanizm routingu,



Rysunek 3.6: Diagram architektury aplikacji klienckiej

który pozwala na przypisywanie komponentom adresów URI. Zdarzenia nawigacji do adresów URI w obrębie aplikacji są przechwytywane przez skrypt *Service Worker* (opisany w sekcji 3.3.1) platformy *Blazor* i przekazywane do obsługi przez wbudowany komponent *Router*, który wyświetla odpowiedni komponent w zależności od adresu. Do budowy interfejsu użytkownika wykorzystana została otwarta biblioteka MudBlazor dostarczająca zestaw gotowych komponentów zgodnych z wytycznymi standardu *Material Design* oraz szablon układu aplikacji zawierający komponenty do nawigacji. W celu ułatwienia nawigacji w aplikacji zaimplementowano dodatkowy komponenty nawigacyjne: *WorkspacesNavGroup* umieszczony w menu bocznym, która pozwala na szybki dostęp do przestrzeni roboczych, do których należy użytkownik oraz *NavigationToolbar* umieszczony w nagłówku aplikacji, który wyświetla ścieżkę nawigacji i pozwala na cofnięcie do poprzednich widoków. Układ aplikacji jest widoczny na rysunku 3.7 przedstawiającym stronę główną aplikacji. Funkcje związane z logowaniem i rejestracją są obsługiwane przez dostawcę tożsamości poprzez wzajemne przekierowanie z aplikacją kliencką. W celu integracji z dostawcą tożsamości w aplikacji klienckiej wykorzystano pakiet *Microsoft.AspNetCore.Components.WebAssembly.Authentication*. Pakiet ten pozwala na uwierzytelnianie użytkowników za pomocą protokołu OpenID Connect oraz automatyczne pobieranie tokenów ID i dostępu od dostawcy tożsamości.

Aplikacja kliencka komunikuje się z serwerem aplikacji za pomocą dwóch mechanizmów: żądań do API serwera za pomocą protokołu HTTP oraz komunikacji w czasie rzeczywistym za pomocą biblioteki *SignalR* wykorzystującej protokół *WebSockets*. W procesie komunikacji z użyciem API HTTP aplikacja kliencka korzysta z klas DTO (ang. Data Transfer Object). Klasy DTO definiują strukturę danych przesyłanych pomiędzy klientem a serwerem, pozwala to ograniczenie danych przesyłanych pomiędzy stronami do niezbędnego minimum oraz uniezależnienie API serwera od modelu domeny biznesowej serwera aplikacji. Kod źródłowy klas DTO jest wspólnie dostępnego dla aplikacji klienckiej i serwera.



Rysunek 3.7: Strona główna aplikacji

aplikacji, co minimalizuje ryzyko błędów wynikających z niespójności pomiędzy danymi oczekiwanyimi a otrzymanymi przez jedną ze stron oraz ułatwia zmiany w strukturze danych transportowych. Przesyłane obiekty klas DTO są automatycznie serializowane (zamienione na postać tekstową) do formatu JSON i deserializowane (odtworzone z postaci tekstowej) po stronie odbiorczej.

Panel administracyjny

Panel administracyjny jest dostępny dla użytkowników posiadających rolę administratora systemu. Pozwala na zarządzanie przeglądanie, zatwierdzanie i usuwanie użytkowników oraz zmianę ich ról w kontekście systemu. Panel podzielony jest na dwie zakładki. Pierwsza wyświetla listę niezatwierdzonych przez administratora użytkowników, którzy zarejestrowali się w systemie. Administrator ma możliwość zatwierdzenia lub usunięcia użytkownika z systemu. Druga zakładka wyświetla listę zatwierdzonych w systemie użytkowników wraz z ich rolami. Administrator ma możliwość zmiany roli użytkownika oraz usunięcia go z systemu. Widoki panelu administracyjnego zostały przedstawione na rysunkach 3.8a, 3.8b oraz 3.8c.

Przestrzenie robocze

Funkcjonalność przestrzeni roboczej jest realizowana przez kilka modułów dostępnych dla zalogowanych i zatwierdzonych użytkowników. Komponent *WorkspaceListPage* (Rysunek 3.9a) wyświetla listę przestrzeni roboczych, do których należy użytkownik, pozwala na przejście do wybranej przestrzeni oraz stworzenie nowej, jeśli użytkownik ma uprawnienia moderatora. Nową przestrzeń można również utworzyć poprzez operację klonowania przestrzeni, która pozwala na stworzenie przestrzeni zawierającej te same pliki lub ustawienia co przestrzeń źródłowa.

Komponent *WorkspacePage* odpowiada za widok pojedynczej przestrzeni roboczej. Składa z czterech podstron zapewniających dostęp do poszczególnych części przestrzeni:

- *Content* – przeglądarka plików przestrzeni (Rysunek 3.9b),
- *Sessions* – lista sesji spotkaniowych w przestrzeni (Rysunek 3.9c),
- *Forum* – forum dyskusyjne przestrzeni (Rysunek 3.9d),
- *Settings* – ustawienia przestrzeni (Rysunek 3.9e).

Podstrony można przełączać za pomocą zakładek znajdujących się w nagłówku komponentu. Podstrona *Content* pozwala na dostęp do plików zgromadzonych w przestrzeni i zarządzanie nimi za pomocą

(a) Widok listy niezatwierdzonych użytkowników

(b) Widok listy zatwierdzonych użytkowników

(c) Dialog ustawień zatwierzonego użytkownika

Rysunek 3.8: Panel administracyjny

komponentu *FileManager*. Pod zakładką *Sessions* możliwe jest wyświetlenie listy sesji spotkaniowych, dołączanie do nich oraz zarządzanie nimi z użyciem komponentu *SessionList*. Podstrona *Forum* wyświetla komponent o tej samej nazwie, który pozwala na korzystanie z forum dyskusyjnego przestrzeni. W przypadku użytkowników posiadających w przestrzeni uprawnienia administratora dostępna jest podstrona *Settings* pozwalająca na zmianę ustawień przestrzeni takich jak nazwa i opis oraz zarządzanie użytkownikami i ich rolami w jej obrębie. Widoki komponentu *WorkspacePage* zostały przedstawione na rysunku 3.9.

Przeglądarka plików

Komponent *FileManager* odpowiada za funkcjonalności związane z zarządzaniem plikami w przestrzeniach roboczych systemu. Jest wykorzystywany zarówno jako samodzielna przeglądarka plików, jak i komponent zintegrowany z innymi modułami systemu. Komponent został przedstawiony na rysunku 3.9b.

W trybie samodzielnej przeglądarki plików, *FileManager* umożliwia pełne zarządzanie zasobami plikowymi wskazanej przestrzeni roboczej. Użytkownik może przeglądać zawartość katalogów, pobierać, usuwać, zmieniać nazwy i uprawnienia dostępu plików oraz katalogów. Dodatkowo możliwe jest zaznaczanie elementów do zbiorczych operacji usuwania, kopiowania i przenoszenia. W przypadku posiadania odpowiednich uprawnień, użytkownik może przenosić i kopiować pliki również do innych przestrzeni. Ten tryb komponentu jest wykorzystywany w widoku przestrzeni roboczej (komponent *WorkspacePage*). W przypadku wybrania pojedynczego pliku, użytkownikowi zostaje przedstawiony podgląd pliku (jeśli dany format jest obsługiwany przez przeglądarkę) z możliwością pobrania go na urządzenie. Pozwala to na szybki dostęp do zawartości typowych materiałów, takich jak dokumenty PDF, obrazy oraz nagrania audio i wideo. Użytkownik ma również możliwość wyświetlenia postów na forum dyskusyjnym przestrzeni, które odnoszą się do wybranego pliku.

Workspaces		Upload	User
Your workspaces		Search	Actions
Workspace name		Creation date ↓	
Workspace 2		14.07.2024 13:21:38	⋮
Workspace 1		09.06.2024 09:34:05	⋮
Rows per page:		25	1-2 of 2 < < > >

(a) Widok listy przestrzeni roboczych

Workspaces > Workspace 1					Upload	User
CONTENT		SESSIONS	FORUM	SETTINGS	Actions	
Root						⋮
Root		Search			C	⋮
Name ↑	Format	Type	Size	Owner	Actions	
Rehearsals	folder	Video		Maciej Zakrzewski	⋮	⋮
Sessions	folder	Session recordings	-		⋮	⋮
Studio recordings	folder	Audio		Maciej Zakrzewski	⋮	⋮
Document	:pdf	Document	143,62 KB	Maciej Zakrzewski	⋮	⋮
Lorem Ipsum	:pdf	Document	76,81 KB	Maciej Zakrzewski	⋮	⋮

(b) Widok przeglądarki plików

Workspaces > Workspace 1					Upload	User
CONTENT		SESSIONS	FORUM	SETTINGS	Actions	
Sessions						⋮
Name	Date ↓				Actions	
Session 2	20.08.2024 18:00:00				⋮	⋮
Jam Session 1	16.08.2024 16:00:00				⋮	⋮

(c) Widok listy sesji

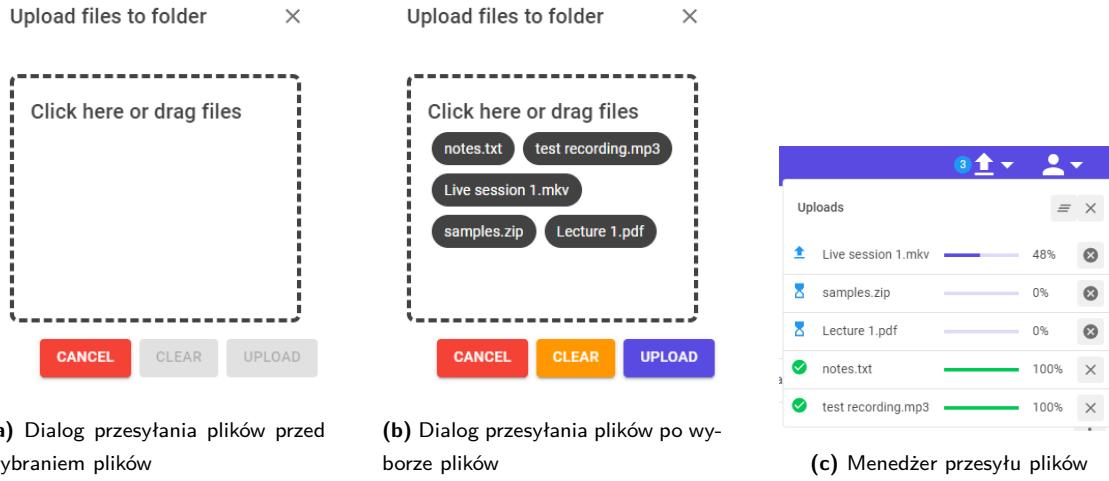
Workspaces > Workspace 1					Upload	User
CONTENT		SESSIONS	FORUM	SETTINGS	Actions	
Lorem ipsum						⋮
Files:	1	1			Actions	
Maciej Zakrzewski posted on 12.08.2024 07:38:34					⋮	⋮
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed porta metus lacus, sed aliquet lorem ullamcorper vita.						⋮
Jan Tymczasowy commented on 12.08.2024 07:39:40					⋮	⋮
Pron ac purus vitae lectus imperdiet venenatis. Etiam semper justo a odio consectetur, molestie tristique mauris pulvinar. Etiam vel lectus aliquet justo pretium feugiat eu ac magna.					⋮	⋮
<input type="button" value="ADD COMMENT"/>						⋮

(d) Widok forum

Workspaces > Workspace 1					Upload	User
CONTENT		SESSIONS	FORUM	SETTINGS	Actions	
DISCARD <input type="button" value="SAVE"/>						DELETE
Your role: Administrator						
Workspace name	T	Workspace 1				
Users	Add members	Search				
<input checked="" type="checkbox"/> Member	Name	Surname ↑	Username	Role	Actions	
<input checked="" type="checkbox"/> Krzysztof	Nowy		krzysztof22	Supervisor	⋮	⋮
<input checked="" type="checkbox"/> Jan	Tymczasowy		jan123	Member	⋮	⋮
<input checked="" type="checkbox"/> All	Legend: NEW MODIFIED DELETED EDITING					
Rows per page: 25 1-2 of 2 < < > >						⋮

(e) Widok ustawień

Rysunek 3.9: Przestrzeń robocza



Rysunek 3.10: Przesyłanie plików

FileManager jest również wykorzystywany innymi komponentami systemu w celu dostarczenia funkcjonalności wyboru plików i katalogów. Przykładowo we współdzielonym edytorze audio oraz narzędziu do tworzenia materiałów wideo użytkownik może wybrać katalog, w którym zostanie zapisany plik wynikowy, a w forum dyskusyjnym wybrać pliki do załączenia do posta.

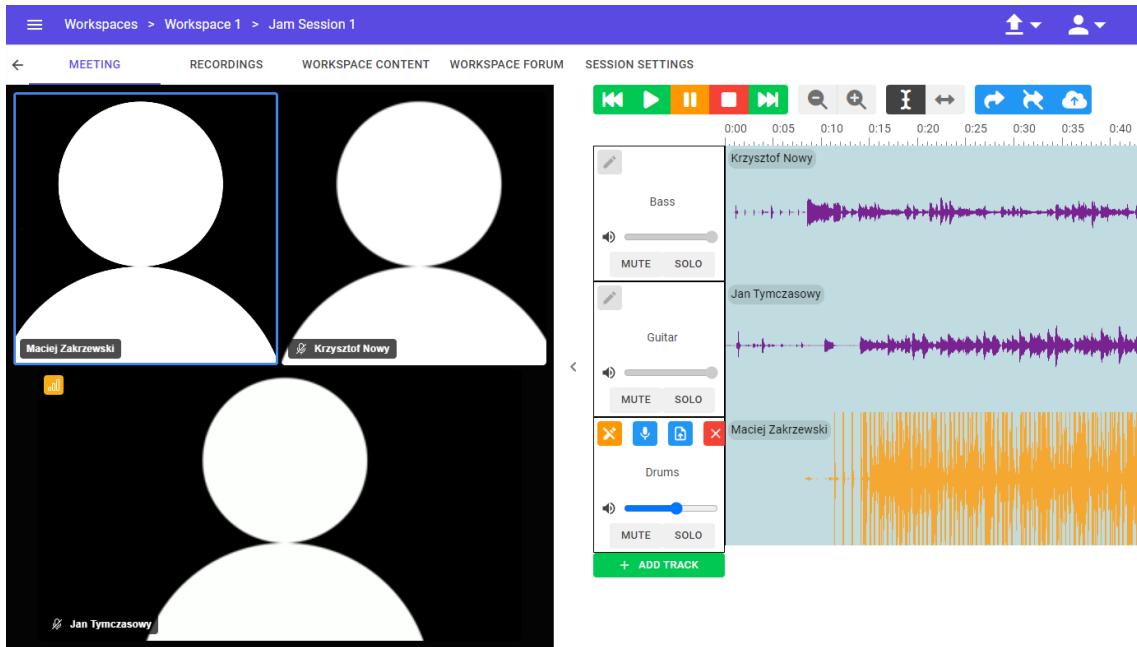
Aby nie blokować interfejsu użytkownika w trakcie przesyłania plików do przestrzeni roboczej, proces ten jest wykonywany w tle za pomocą instancji klasy *StorageService* dostarczanej przez wstrzykiwanie zależności. Po wybraniu opcji przesyłania plików wyświetlany jest dialog pozwalający na wybranie plików poprzez przeciągnięcie ich do niego lub za pomocą systemowego eksploratora plików (Rysunek 3.10a). Po wybraniu plików użytkownik widzi listę plików, których przesyłanie może następnie zatwierdzić bądź odrzucić (Rysunek 3.10b). Przesyłane pliki są wyświetlane w zwijanym komponencie *UploadManager* dostępnym w prawym górnym rogu interfejsu aplikacji (Rysunek 3.10c). Pozwala on na monitorowanie postępu przesyłania plików, anulowanie ich oraz wyświetlenie historii przesyłanych plików, z możliwością jej wyczyszczenia. W przypadku powodzenia lub błędu operacji przesyłania plików użytkownikowi wyświetlane jest nieinwazyjne powiadomienie w prawym dolnym rogu interfejsu.

Forum dyskusyjne

Komponent *Forum* w systemie Concerto pozwala użytkownikom na komunikację asynchroniczną w ramach przestrzeni roboczej. Forum pozwala na tworzenie postów, w których można odnosić się do plików przesłanych do przestrzeni. Każdy post może być komentowany, a zarówno posty, jak i komentarze mogą być edytowane przez ich autora, co umożliwia wprowadzanie poprawek lub dodatkowych informacji. Moderatorzy przestrzeni roboczej mają możliwość usuwania postów i komentarzy innych użytkowników. Rysunek 3.9d przedstawia widok forum dyskusyjnego przestrzeni roboczej.

Sesje spotkanio

Sesje spotkanio umożliwiają użytkownikom komunikację w czasie rzeczywistym w formie wideokonferencji zintegrowanej ze współdzielonym edytorem audio. Sesje spotkanio można nagrywać poprzez wybranie odpowiedniej opcji w interfejsie wideokonferencji Jitsi Meet. Nagrania są wykonywane na serwerze Jibri opisanym w sekcji 3.3.4, będącym częścią wdrożenia Jitsi Meet i automatycznie archiwizowane w przestrzeni roboczej po ręcznym przerwaniu nagrywania bądź opuszczeniu spotkania przez wszystkich uczestników (serwer Jibri jest zintegrowany z serwerem Concerto). Proces nagrywania



Rysunek 3.11: Widok głównej zakładki sesji spotkaniowej

jest w pełni zautomatyzowany i niezależny urządzeń użytkowników, co zapewnia niezawodność oraz jakość nagrań. Sesje spotkaniowe są dostępne dla wszystkich użytkowników przestrzeni roboczej, jednak tylko moderatorzy mają możliwość ich tworzenia i usuwania.

W trakcie sesji użytkownicy mogą korzystać ze współdzielonego edytora audio, który pozwala na grupową pracę nad utworem muzycznym w sposób inkrementalny. Praca z edytorem polega na nakładaniu na siebie kolejnych ścieżek dźwiękowych. Użytkownicy mogą tworzyć nazwane ścieżki, a następnie wybierać je do edycji co powoduje zablokowanie możliwości ich zmiany innym użytkownikom w celu uniknięcia konfliktów. Zmiany wprowadzane przez użytkowników są automatycznie utrwalane na serwerze i synchronizowane w czasie rzeczywistym z innymi uczestnikami sesji. W trakcie edycji ścieżki użytkownicy mogą nagrać ją za pomocą urządzenia wejściowego audio (np. mikrofonu bądź instrumentu muzycznego widocznego w systemie jako wejście audio) lub zimportować z pliku. W trakcie nagrywania odtwarzane są inne ścieżki w celu zapewnienia podkładu i stworzenia wrażenia wspólnej gry. Poziomy głośności sprzętu oraz opóźnienia występujące w sprzęcie użytkowników mogą się różnić, dlatego istnieje możliwość regulacji głośności oraz czasu rozpoczęcia poszczególnych ścieżek w celu dopasowania ich do siebie. Ścieżki dźwiękowe są wizualizowane w formie przebiegu amplitudy sygnału w czasie z możliwością regulowania skali wykresu, pomaga to w procesie ręcznej korekty synchronizacji ścieżek. Użytkownicy mogą również odtwarzać utwór lokalnie od dowolnie wybranego momentu w celu weryfikacji synchronizacji i brzmienia ścieżek. Możliwe jest również wyciszenie bądź oznaczenie ścieżek jako solo niezależnie od innych użytkowników, co pozwala wykluczyć niedopracowane jeszcze ścieżki bądź umożliwić nagrywanie do spersonalizowanego podkładu.

Edytor oferuje również możliwość wspólnego odsłuchu wynikowego utworu poprzez integrację z wideokonferencją. Utwór jest wtedy łączony w całość i serwowany przez serwer *Concerto*, który jest następnie odtwarzany u każdego z użytkowników poprzez zsynchronizowany odtwarzacz dostępny w Jitsi Meet. Dzięki temu użytkownicy mogą wspólnie odsłuchać efekt swojej pracy, z możliwością zatrzymania, przewijania i ponownego odtworzenia utworu. Rozwiążanie to ma pozwalać na sprawną dyskusję nad rezultatami, wspólną identyfikację obszarów do poprawy oraz szybką koordynację dalszej pracy. Dzięki uniknięciu strumieniowania dźwięku przez użytkowników, jakość dźwięku jest zachowana na wysokim

poziomie, a opóźnienia w transmisji nie wpływają na jakość odsłuchu. Wynikowy utwór można zapisać w wybranym katalogu przestrzeni roboczej.

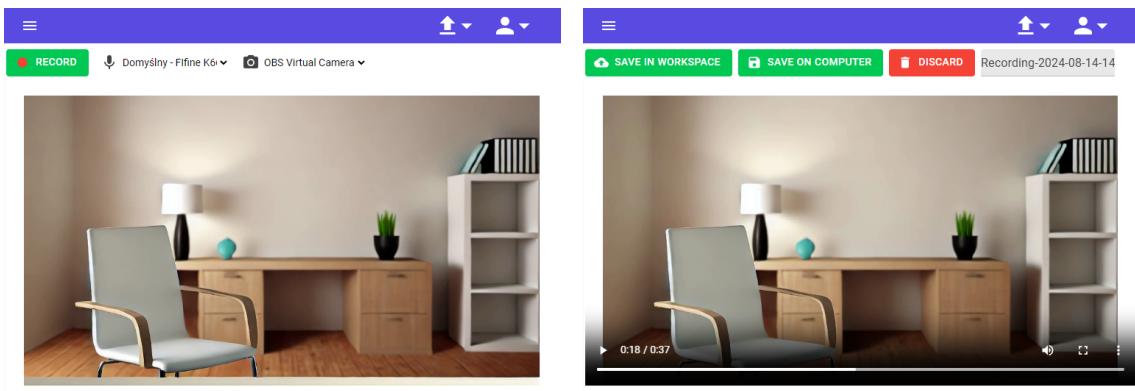
Na funkcjonalność sesji spotkaniowych składają się komponenty *SessionList* oraz *SessionPage*. Komponent *SessionList* wyświetla listę sesji spotkaniowych w danej przestrzeni roboczej. Moderatorzy przestrzeni mają możliwość dodawania nowych i usuwania istniejących sesji. Wybranie sesji z listy powoduje otwarcie komponentu *SessionPage*, który wyświetla widok danej sesji spotkaniowej. Widok ten jest podzielony na cztery zakładki:

- *Meeting* – wideokonferencja oraz współdzielony edytor audio,
- *Workspace content* – przeglądarka plików przestrzeni roboczej,
- *Workspace forum* – forum dyskusyjne przestrzeni,
- *Settings* – ustawienia sesji.

Główna zakładka *Meeting* zagnieździła w sobie komponent o tej samej nazwie, który wyświetla i integruje z sobą wideokonferencję Jitsi Meet oraz współdzielony edytor audio. Zakładki *Workspace Content* oraz *Workspace Forum* zapewniają dostęp do zasobów przestrzeni bez konieczności opuszczania sesji spotkaniowej. Zakładka *Settings* pozwala na zmianę ustawień sesji takich jak nazwa, opis i planowany czas rozpoczęcia przez moderatora przestrzeni. Widok zakładki *Meeting* został przedstawiony na rysunku 3.11.

Integracja z serwerem wideokonferencji polega na osadzeniu w komponencie *Meeting* klienta Jitsi Meet za pomocą elementu HTML *iframe*. Element ten jest tworzony dynamicznie za pomocą interfejsu dostępnego w skrypcie JavaScript *external_api.js*. Do klienta wideokonferencji przekazywane są unikalny identyfikator spotkania, imię i nazwisko użytkownika oraz wygenerowany przez serwer *Concerto* token JWT autoryzujący użytkownika w Jitsi Meet.

Komponent *Daw* zagnieźdzony w komponencie *Meeting* implementuje funkcjonalność współdzielonego edytora audio. Wykorzystuje on zmodyfikowaną na potrzeby systemu otwartoźródłową aplikację JavaScript *waveform-playlist* [57] autorstwa Naomi Aro, dostępną na licencji MIT, która pozwala na tworzenie playlisty ścieżek audio, ich edycję, nagrywanie, wizualizację oraz odtwarzanie w przeglądarce internetowej. Oryginalna aplikacja *waveform-playlist* pozwala jedynie na pracę lokalnie przez jednego użytkownika. Na potrzeby systemu *Concerto* zmodyfikowano i rozszerzono interfejs programistyczny oraz implementację *waveform-playlist* w celu umożliwienia pracy grupowej i synchronizację stanu edytora w czasie rzeczywistym. Dodano funkcjonalności takie jak blokowanie edycji poszczególnych ścieżek, możliwość dynamicznej podmiany danych źródłowych ścieżek oraz zmiany ich kolejności. Rozszerzono również zdarzenia generowane przez *waveform-playlist* w celu sygnalizacji zmian stanu edytora oraz przekazywania danych związanych z nagrywaniem ścieżek. Komponentem *Daw* komunikuje się z *waveform-playlist* za pomocą mechanizmu *JSInterop* platformy *Blazor*. Odpowiada on za ustalenie połączenia z hubem *SignalR* serwera *Concerto* i synchronizację stanu edytora z serwerem *Concerto* poprzez przesyłanie lokalnych zmian do serwera oraz obsługę powiadomień o zmianach otrzymywanych z serwera. Funkcjonalność wspólnego odsłuchu wynikowego utworu jest realizowana poprzez żądanie wygenerowania pliku audio na serwerze *Concerto* a następnie wywołanie funkcji wspólnego odtwarzania otrzymanego adresu do pliku w wideokonferencji za pomocą interfejsu *external_api.js*. Zapis utworu w przestrzeni roboczej jest realizowany poprzez wybór katalogu w przestrzeni w komponencie *FileManager* a następnie wysłanie żądania zapisu do serwera *Concerto*.



(a) Widok przed nagraniem

(b) Widok po nagraniu

Rysunek 3.12: Narzędzie do nagrywania materiałów wideo

Nagrywanie materiałów wideo

Komponent *RecordingTool* jest odpowiedzialny za realizację funkcjonalności narzędzia do nagrywania materiałów wideo. Pozwala on użytkownikom na wybranie wejściowego urządzenia audio i wideo (np. mikrofonu i kamery internetowej) i przygotowanie nagrania. W trakcie nagrania użytkownik widzi podgląd wybranego urządzenia wideo. Po zakończeniu nagrania użytkownik może odtworzyć nagranie w celu jego weryfikacji, a następnie odrzucić je bądź zapisać na swoim urządzeniu lub przesłać do wybranego katalogu w jednej z dostępnych przestrzeni roboczych.

Część komponentu odpowiedzialna nagrywanie i podgląd nagrania została zawarta w skrypcie JavaScript w formie klasy *RecordingManager* wykorzystującej następujące interfejsy przeglądarki internetowej:

- *MediaDevices* — wykrywanie i dostęp do urządzeń wejściowych,
- *MediaStream* — obsługa i łączenie strumieni audio i wideo,
- *MediaRecorder* — nagrywanie i kodowanie strumieni audio i wideo.

Klasa ta jest odpowiedzialna za konfigurację urządzeń wejściowych oraz połączenie i zarejestrowanie strumieni audio i wideo. Klasa komunikuje się z komponentem *RecordingTool* za pomocą mechanizmu *JSInterop* platformy *Blazor* w celu wymiany danych i zdarzeń generowanych przez użytkownika. W trakcie nagrywania wyświetlany jest na żywo podgląd z urządzenia wideo. Po zakończeniu zastępuje podgląd odtwarzaczem wideo prezentującym nagranie i przekazuje nagrany plik do komponentu *RecordingTool* w platformie *Blazor*. Komponent *RecordingTool* wykorzystuje komponent *FileManager* w celu zapisania nagrania w przestrzeni roboczej. Widoki komponentu *RecordingTool* przed i po nagraniu zostały przedstawione odpowiednio na rysunkach 3.12a oraz 3.12b.

Aplikacja mobilna

Aplikacje kliencką Blazor WebAssembly skonfigurowano do działania jako progresywna aplikacja internetowa. Widoki komponentów stworzono tak, aby były responsywne i dostosowywały się do wielkości ekranu urządzenia. Aplikację można zainstalować na urządzeniu mobilnym poprzez odwiedzenie witryny aplikacji i wybranie opcji instalacji w przeglądarce internetowej, a następnie korzystać z niej podobnie jak z aplikacji natywnej. Na urządzeniach mobilnych funkcjonalność aplikacji jest ograniczona względem wersji desktopowej. W szczególności nie jest dostępne narzędzie do nagrywania materiałów wideo oraz współdzielony edytor audio, ze względu na ograniczenia przestrzeni ekranu i dotykowego

interfejsu użytkownika. Obsługa wideokonferencji jest oddelegowywana do oficjalnej aplikacji Jitsi Meet dostępnej na platformy mobilnej. Wybrane widoki wersji mobilnej aplikacji zostały przedstawione na rysunku 3.13.

(a) Panel administracyjny

Workspace 2

ALL USERS		MORE
Users	owy	⟳
Surname	Nowy	
Name	Krzysztof	
Surname	Nowy	
Username	krzysztof22	
Email	xijifoc754@leacore.com	
Role	User	
Actions		⚙️ 🧑
Name	Jan	
Surname	Tymczasowy	
Username	jan123	
Email		siwemow3195@fuzitea.com

(b) Lista przestrzeni roboczych

Workspaces

Your workspaces		Search	⋮
Creation date	Created on	23.08.2024 16:23:22	⋮
Actions		⋮	
Name	Workspace 2	Created on	23.08.2024 16:12:39
Actions		⋮	
Name	Workspace 1	Created on	23.08.2024 16:10:13
Actions		⋮	
Rows per page: 25 1-4 of 4 < >			

(c) Ustawienia przestrzeni roboczej

Workspace 1

SETTINGS		DISCARD	SAVE	DELETE
Your role: Administrator				
Workspace name		T Workspace 1		
Users		Add members	Search	
Sort By		Surname		
Member		<input checked="" type="checkbox"/>		
Name	Krzysztof			
Surname	Nowy			
Username	krzysztof22			
Role	Member			
Actions		⚙️		
Member		<input checked="" type="checkbox"/>		
Name	Karolina			
Surname	Prasat			
Username	karolina			

(d) Przeglądarka plików

Workspace 1

CONTENT		SES:
Root		
Search		⋮
Sort By		
Name	Sessions	
Format	folder	
Type	Session recordings	
Size		
Owner		
Actions		
Name	Lorem Ipsum	.pdf
Format		
Type	Document	
Size	76,81 KB	
Owner	Maciej Zakrzewski	
Actions		

(e) Lista sesji spotkaniowych

Workspace 1

SESSIONS		FO	
Sessions		⟳ + ⋮	
Date		⋮	
Name	Jam Session 1	Scheduled for	22.08.2024 19:32:45
Actions		⋮	
Name	test	Scheduled for	22.08.2024 18:32:19
Actions		⋮	

(f) Forum dyskusyjne

Workspace 1

Maciej Zakrzewski posted on 31.08.2024 14:13:04

Post

Ipsum dolor sit amet, consectetur adipiscing elit. Sed porta metus lacus, sed aliquet lorem ullamcorper vitae. Nullam commodo accumsan leo in molestie. Proin lobortis fringilla nisl, sit amet pharetra felis tristique ut.

Maciej Zakrzewski commented on 31.08.2024 14:13:20

Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Suspendisse vitae tincidunt lectus, a accumsan massa.

+ ADD COMMENT

C CREATE POST

Rysunek 3.13: Wybrane widoki wersji mobilnej aplikacji klienckiej

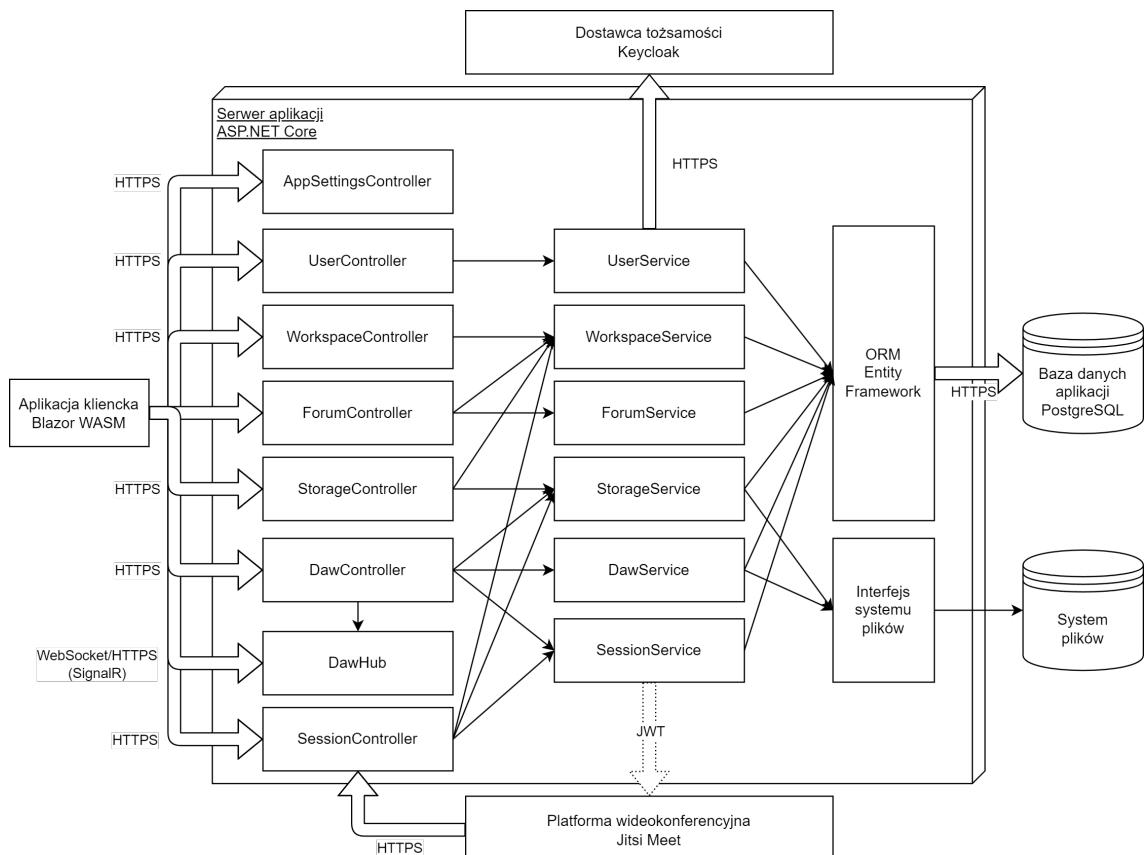
3.4.3. Serwer aplikacji

Serwer aplikacji został oparty na platformie ASP.NET Core. Implementacja serwera została podzielona na trzy warstwy odpowiadające architekturze systemu opisanej w sekcji 3.2. Są to:

- warstwa logiki prezentacji,
- warstwa logiki aplikacji,
- warstwa dostępu do danych.

Na warstwę logiki prezentacji składają się kontrolery API webowego obsługujące żądania HTTP, hub do komunikacji w czasie rzeczywistym oraz komponenty odpowiedzialne za uwierzytelnianie użytkowników i autoryzację ich żądań. Warstwa logiki aplikacji zawiera implementację modelu domeny opisanego w sekcji 3.1.6 oraz serwisy realizujące logikę z nim związaną. Warstwa dostępu do danych odpowiada za utrwalanie zmian wprowadzanych do systemu w bazie danych i magazynie plików.

Diagram architektury serwera aplikacji został przedstawiony na rysunku 3.14. Szerokie strzałki z etykietami oznaczają komunikację sieciową z użyciem protokołów określonych w etykietach. Szeroka strzałka o przerywanej linii oznacza niebezpośrednią wymianę informacji pomiędzy serwerem aplikacji a serwerem wideokonferencji za pomocą tokenu JWT. Strzałki cienkie oznaczają komunikację w obrębie serwera. Kierunek strzałek określa stronę inicującą komunikację i tym samym kierunek zależności pomiędzy elementami. Nazwy elementów wewnętrz obrazu serwera aplikacji odpowiadają nazwom klas w kodzie źródłowym. Kolejne rzędy elementów odpowiadają wymienionym wyżej warstwom.



Rysunek 3.14: Diagram architektury serwera aplikacji

Integracja z dostawcą tożsamości

Większość operacji w systemie wymaga przeprowadzenia procesu uwierzytelniania i autoryzacji użytkownika w celu weryfikacji jego uprawnień do poszczególnych zasobów. Aplikacja kliencka przesyła wraz z żądaniami token dostępu JWT otrzymany od dostawcy tożsamości. Zawiera on informacje potrzebne do określenia uprawnień użytkownika do każdego z zasobów serwera takie jak jego unikalny identyfikator oraz role w obrębie systemu. W celu obsługi tokenów JWT w serwerze aplikacji skonfigurowano wbudowany w platformę ASP.NET Core serwis *IAuthenticationService* weryfikujący tokeny. Serwis skonfigurowano poprzez ustawienie schematu uwierzytelniania na *JwtBearer* oraz wskazanie adresu dostawcy tożsamości Keycloak. Serwis pozwala na wykorzystanie dostępnego w platformie oprogramowania pośredniego, które pozwala na proste zabezpieczanie kontrolerów API przed nieautoryzowanym dostępem oraz łatwy dostęp do informacji zawartych w tokenie w trakcie obsługi żądania.

Model domeny i integracja z bazą danych

Model domeny systemu został zaimplementowany za pomocą biblioteki Entity Framework Core opisanej w sekcji 3.3.2. Zdefiniowane klasy odpowiadają encjom modelu domeny opisanego w sekcji 3.1.6 z różnicami wynikającymi z konieczności dostosowania modelu do możliwości wykorzystywanej środowiska programistycznego i standardów biblioteki Entity Framework Core oraz ułatwienia implementacji logiki aplikacji. Przykładem takiej różnicy jest dodanie do klas pól reprezentujących relacje w postaci kluczy obcych i referencji do innych encji w celu ich zdefiniowania i wykorzystania pełnych możliwości mechanizmu odwzorowywania relacyjno-obiektowego biblioteki Entity Framework Core. Niektóre zmiany wynikają z chęci optymalizacji modelu pod kątem wydajności. Przykładem jest zastąpienie danych binarnych plików i ścieżek dźwiękowych referencjami do zawierających je plików w systemie operacyjnym. Ma to na celu zmniejszenie obciążenia bazy danych oraz przyspieszenie operacji związanych z przesyłaniem i przetwarzaniem plików.

Biblioteka Entity Framework Core dostarcza implementacji klasy bazowej *DbContext*, która zarządza połączeniem z bazą danych oraz pozwala na wykonywanie operacji na encjach poprzez mechanizm odwzorowywania relacyjno-obiektowego. Klasa ta realizuje więc funkcjonalność warstwy dostępu do danych. Serwer aplikacji zintegrowano z bazą danych PostgreSQL poprzez stworzenie klasy *ConcertoDbContext* dziedziczącej po *DbContext* i zawierającej zbiory klas reprezentujących encje w bazie danych. Podczas uruchamiania serwera klasa ta jest rejestrowana w platformie ASP.NET Core wraz z konfiguracją połączenia z bazą danych. Instancja tej klasy jest zarządzana przez platformę ASP.NET Core i dostępna w serwisach warstwy logiki aplikacji poprzez mechanizm wstrzykiwania zależności.

Interfejs webowy

Interfejs webowy serwera aplikacji realizuje warstwę logiki prezentacji systemu. Został zaimplementowany przy użyciu kontrolerów API dostępnych w ASP.NET Core. Głównym komponentem używanym do budowy kontrolerów jest klasa bazowa *ControllerBase*, która automatyzuje standardowe operacje związane z obsługą żądań HTTP takie jak konfiguracja adresu kontrolera, ekstrakcja, deserializacja i walidacja parametrów żądania, obsługa błędów, autoryzacja oraz serializacja odpowiedzi. Interfejs został podzielony na wiele kontrolerów, z których każdy odpowiada za obsługę wydzielonej części funkcjonalności systemu po stronie serwera. Konwencja interfejsu kontrolerów jest oparta na akcjach w odróżnieniu od stylu REST (ang. Representational State Transfer) zorientowanego na zasoby. Tabela 3.2 przedstawia kontrolery serwera aplikacji składające się na jego interfejs webowy. Dla każdego kontrolera podano nazwę klasy kontrolera, opis funkcjonalności oraz serwisy logiki aplikacji, z których korzysta.

Tabela 3.2: Kontrolery API serwera aplikacji

Nazwa klasy	Opis	Serwisy
<i>AppSettingsController</i>	Dostarcza ustawień aplikacji klienckiej takich jak adresy serwera dostawcy tożsamości i serwera wideokonferencji.	–
<i>UserController</i>	Zarządzanie użytkownikami w systemie. Wykorzystywany przez panel administracyjny.	UserService
<i>WorkspaceController</i>	Zarządzanie i dostęp do przestrzeni roboczych.	WorkspaceService
<i>StorageController</i>	Zarządzanie plikami i katalogami oraz procesem przesyłania plików. Wykorzystywany przez przeglądarkę plików przestrzeni roboczej oraz serwer wideokonferencji w celu zapisu nagrani.	WorkspaceService StorageService
<i>ForumController</i>	Zarządzanie postami i komentarzami na forach dyskusyjnych przestrzeni roboczych.	WorkspaceService ForumService
<i>SessionController</i>	Zarządzanie i dostęp do sesji spotkaniowych przestrzeni roboczych. Odbiór i zapis nagrani wideokonferencyjnych.	WorkspaceService SessionService StorageService
<i>DawController</i>	Zarządzanie i dostęp do projektów muzycznych sesji spotkaniowych. Wykorzystywany przez współdzielony edytor audio do koordynacji pracy i synchronizacji stanu lokalnego z serwrem oraz współdzielony odtwarzacz multimedialny klienta wideokonferencji.	SessionService DawService StorageService

Współdzielony edytor audio w aplikacji klienckiej, oprócz korzystania z kontrolera *DawController*, wykorzystuje również hub biblioteki SignalR – *DawHub*. Odgrywa on kluczową rolę w procesie synchronizacji stanu edytora pomiędzy klientami w czasie rzeczywistym. Hub *DawHub* jest odpowiedzialny za powiadamianie podłączonych klientów o zmianach stanu projektu dokonywanych poprzez kontroler *DawController*. Po zestawieniu połączenia SignalR z hubem, połączenie jest rejestrowane w grupie związanej z identyfikatorem sesji spotkanowej. Po wykonaniu operacji wymagającej synchronizacji stanu edytora *DawController* wysyła za pośrednictwem huba powiadomienie o zmianie do wszystkich klientów w grupie. Klienci odbierają powiadomienie, a następnie aktualizują stan edytora za pomocą kontrolera *DawController*.

Serwisy logiki aplikacji

Warstwę logiki aplikacji tworzą serwisy realizujące logikę związaną z modelem domeny. Każdy z serwisów odpowiada za logikę związaną z jedną lub wieloma blisko powiązanymi encjami. Klasy serwisowe są rejestrowane w platformie ASP.NET Core i dostępne w kontrolerach API za pomocą mechanizmu wstrzykiwania zależności. Warstwa logiki aplikacji izoluje model domeny systemu od warstwy logiki prezentacji za pomocą klas DTO (sekcja 3.4.2) wymienianych między aplikacją kliencką a serwrem aplikacji. Do pobierania danych i utrwalania zmian w bazie danych serwisy wykorzystują opisaną wcześniej klasę *ConcertoDbContext*. Operacje związane z przechowywaniem danych binarnych plików wykorzystują interfejs systemu operacyjnego dostępny w platformie .NET. Operacje na plikach dźwiękowych korzystają z oprogramowania FFmpeg oraz biblioteki FFmpegCore. Tabela 3.3 przedstawia serwisy logiki aplikacji serwera aplikacji składające się na jego warstwę logiki aplikacji. Dla każdego serwisu podano nazwę klasy serwisu, opis funkcjonalności oraz encje modelu domeny (sekcja 3.1.6), na których

bezpośrednio operuje.

Tabela 3.3: Serwisy logiki domenowej serwera aplikacji

Nazwa klasy	Opis	Encje
<i>UserService</i>	Operacje związane z użytkownikami systemu i integracja z API dostawcy tożsamości. Pobieranie, zatwierdzanie, usuwanie, zmiana ról użytkowników.	Użytkownik
<i>WorkspaceService</i>	Operacje związane z przestrzeniami roboczymi. Pobieranie, tworzenie, modyfikacja, usuwanie przestrzeni. Dodawanie, usuwanie członków przestrzeni. Zmiana ról członków w obrębie przestrzeni. Weryfikacja członkostwa i roli w przestrzeni.	Przestrzeń Członkostwo
<i>StorageService</i>	Operacje związane z plikami i katalogami w przestrzeniach roboczych, integracja z systemem plików, zapis nagrani z wideokonferencji. Pobieranie, tworzenie, usuwanie, modyfikacja plików i katalogów. Zarządzanie danymi binarnymi plików w systemie plików.	Katalog Plik Dostęp
<i>ForumService</i>	Operacje związane z postami i komentarzami na forach dyskusyjnych przestrzeni roboczych. Tworzenie, dodawanie, modyfikacja postów i komentarzy. Weryfikacja uprawnień do modyfikacji i usuwania postów i komentarzy.	Post Komentarz
<i>SessionService</i>	Operacje związane z sesjami spotkanowymi przestrzeni roboczych. Pobieranie, tworzenie, modyfikacja sesji spotkanowych. Weryfikacja uprawnień do tworzenia, modyfikacji i dostępu do sesji. Generowanie tokenów dostępu do pokojów wideokonferencyjnych.	Sesja
<i>DawService</i>	Operacje związane z projektami muzycznymi sesji spotkanowych. Dodawanie, przejmowanie, zwalnianie, modyfikacja, usuwanie ścieżek. Generowanie pliku wynikowego projektu muzycznego. Zapis wynikowego pliku w wybranym katalogu.	Projekt muzyczny Ścieżka Plik Katalog

3.4.4. Platforma wideokonferencyjna

W celu zabezpieczenia platformy wideokonferencyjnej przed niepowołanym dostępem wykorzystano wbudowany w platformę Jitsi Meet mechanizm autoryzacji użytkowników za pomocą tokenów JWT. Platforma Jitsi Meet i serwer Concerto współdzielą tajny klucz symetryczny. Serwer Concerto generuje tokeny dostępu JWT w formacie wymaganym przez Jitsi Meet pozwalające jego posiadaczowi dołączenie do danego spotkania, a następnie podpisuje je za pomocą klucza symetrycznego. Podczas dołączania do pokoju wideokonferencyjnego danej sesji spotkanowej aplikacja kliencka pobiera token z serwera Concerto, który uprzednio weryfikuje czy użytkownik ma do tego uprawnienia, a następnie przekazuje go do klienta Jitsi Meet. Token zostaje przesyłany do platformy Jitsi Meet, która weryfikuje token i decyduje o przyjęciu użytkownika do spotkania.

Jednym z komponentów platformy Jitsi Meet jest Jitsi Broadcasting Infrastructure (Jibri), który pozwala na nagrywanie spotkań. Nagrania są zapisywane w systemie plików instancji Jibri wraz z metadanymi dotyczącymi spotkania. Po zakończeniu nagrywania Jibri uruchamia skrypt powłoki wskazywany przez zmienną środowiskową *JIBRI_FINALIZE_RECORDING_SCRIPT_PATH* przekazując mu ścieżkę do katalogu z nagraniem i metadanymi jako argument. W celu automatycznej archiwizacji nagrani w przestrzeni roboczej systemu Concerto przygotowano skrypt *finalize.sh*, który przesyła nagranie do ser-

wera Concerto za pomocą interfejsu webowego kontrolera *SessionController* autoryzując się za pomocą współdzielonego klucza. W żądaniu oprócz nagrania zawarty jest unikalny identyfikator spotkania, przy pomocy którego serwer Concerto identyfikuje przestrzeń roboczą, do której należy przypisać nagranie.

3.5. Wdrożenie systemu

W celu przetestowania zaimplementowanego systemu w warunkach rzeczywistych został on wdrożony na maszynie wirtualnej działającej w infrastrukturze Oracle Cloud, wykorzystując darmowy poziom usługi Oracle Free Tier. Maszyna wirtualna została skonfigurowana na bazie systemu operacyjnego Ubuntu 22.04 i wykorzystuje wszystkie dostępne w ramach darmowego poziomu usługi zasoby. Parametry maszyny wirtualnej zostały przedstawione w tabeli 3.4. W ramach usługi dostępny jest publiczny adres IP, który został wykorzystany do udostępnienia systemu w sieci Internet.

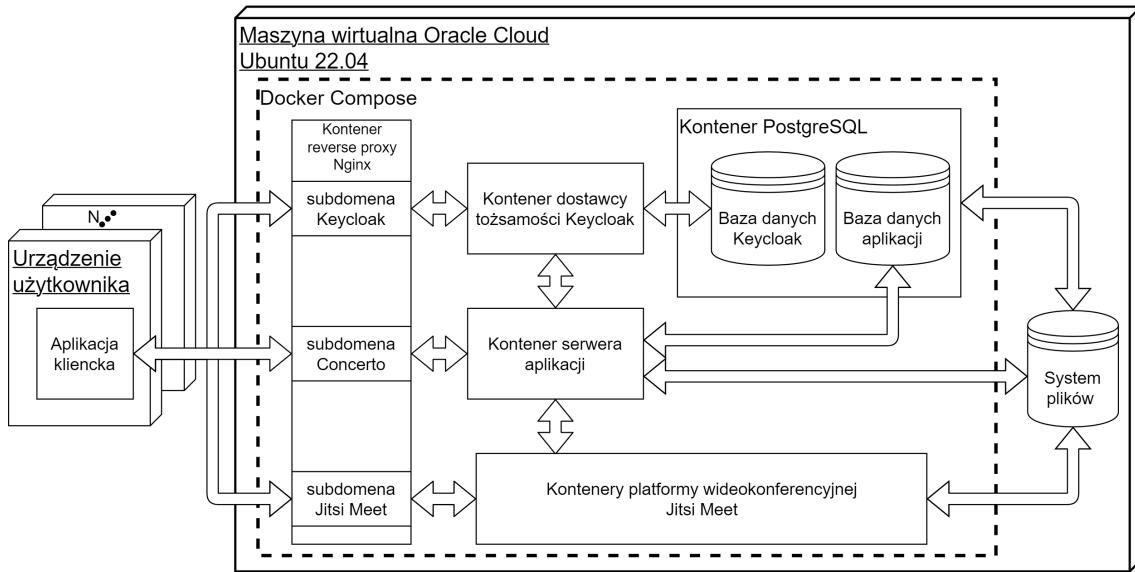
Tabela 3.4: Parametry maszyny wirtualnej

Parametr	Wartość
System operacyjny	Ubuntu 22.04
Procesor	4 × OCI Ampere A1 (maksymalnie taktowanie 3Ghz)
Pamięć RAM	24 GB
Przestrzeń dyskowa	200 GB
Przepustowość sieci	4 Gb/s

Aby odizolować środowisko uruchomieniowe komponentów systemu od systemu operacyjnego maszyny wirtualnej i ułatwić proces wdrażania kolejnych wersji systemu, zdecydowano się na wykorzystanie techniki konteneryzacji z pomocą narzędzia Docker. Kontenery są jednostkami oprogramowania, które zawierają wszystkie elementy niezbędne do uruchomienia danego programu takie jak kod wykonywalny, biblioteki, środowisko uruchomieniowe, konfigurację i inne zależności. Kontenery są uruchamiane przez system operacyjny w izolowanym środowisku i wykorzystują jądro systemu operacyjnego gospodarza. Dzięki temu są niezależne od konfiguracji systemu operacyjnego gospodarza, lekkie i zbliżone wydajnością do programów uruchamianych bezpośrednio w systemie operacyjnym.

Komponenty systemu zostały skonteneryzowane przy pomocy narzędzia Docker. Ze względu na to, że system składa się z wielu komponentów, zdefiniowany został plik konfiguracyjny narzędzia Docker Compose, które pozwala na zdefiniowanie, konfigurację i uruchomienie wielu kontenerów jednocześnie. W pliku konfiguracyjnym zadeklarowano kontenery dla każdego z komponentów systemu, zdefiniowano sieci pozwalające na komunikację między kontenerami, zmienne środowiskowe przekazywane do kontenerów oraz katalogi systemu plików maszyny wirtualnej udostępnione wewnętrz kontenerów. Konfiguracja poszczególnych komponentów systemu odbywa się przez ustawienie odpowiednich wartości zmiennych środowiskowych w pliku konfiguracyjnym.

Ze względu na wykorzystanie tych samych protokołów do komunikacji z klientem aplikacji przez wiele komponentów systemu, które mogą być uruchomione na tej samej maszynie, konieczne było zastosowanie mechanizmu odwróconego serwera pośredniczącego (ang. reverse proxy). Odwrócony serwer pośredniczący izoluje infrastrukturę sieciową systemu od urządzeń klienckich, przekierowując ruch sieciowy z zewnątrz do odpowiednich komponentów systemu. Wykorzystany został w tym celu serwer Nginx w formie kontenera Docker, skonfigurowany tak, aby serwer Keycloak, serwer aplikacji oraz platforma Jitsi Meet były dostępne z zewnątrz pod różnymi subdomenami. Rysunek 3.15 przedstawia diagram testowego wdrożenia systemu.

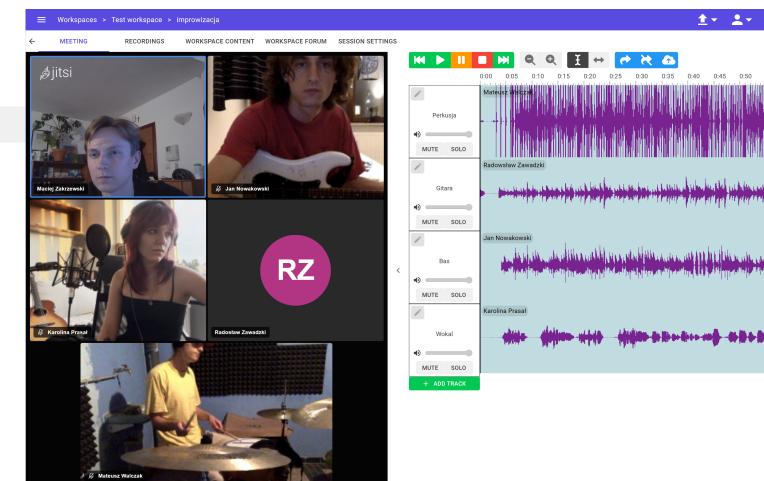


Rysunek 3.15: Diagram wdrożenia systemu

3.6. Testy użytkowe

Aby ocenić użyteczność zaimplementowanego systemu, przeprowadzono testy użytkowe z udziałem grupy 4 osób, które są aktywnymi uczestnikami zespołów muzycznych lub prowadzą działalność solową. Użytkownikom przedstawiono cele, jakie ma realizować system oraz oferowane przez niego funkcje. Następnie zostali poproszeni o przetestowanie funkcjonalności systemu i przeprowadzenie spotkania online z próbą muzyczną lub sesją improwizacyjną. Po zakończeniu testów użytkownicy wypełnili kwestionariusz skali użyteczności systemu (ang. system usability scale, SUS), który pozwala ocenić ogólną użyteczność systemu w skali od 0 do 100 punktów [58, 59]. Oprócz ankiety użytkownicy zostali poproszeni o zapisywanie swoich uwag i spostrzeżeń w trakcie korzystania z systemu. Rysunek 3.16 przedstawia sesję spotkanową w trakcie testów.

Kwestionariusz skali użyteczności systemu składa się z 10 pytań, na które użytkownicy odpowiadają w pięciostopniowej skali Likerta, gdzie 1 oznacza całkowicie się nie zgadzam, a 5 całkowicie się zgadzam.



Rysunek 3.16: Sesja spotkanowa w trakcie testów

Końcowy wynik dla każdego zestawu odpowiedzi jest obliczany na podstawie wzoru:

$$SUS = 2.5 \times (20 + (P_1 + P_3 + P_5 + P_7 + P_9) - (P_2 + P_4 + P_6 + P_8 + P_{10}))$$

gdzie P_1 do P_{10} to odpowiedzi użytkowników na poszczególne pytania. Wynik skali mieści się w przedziale od 0 do 100 punktów, wyższa wartość oznacza lepszą użyteczność systemu. System z wynikiem powyżej 68–70 punktów uznaje się za ponadprzeciętnie użyteczny [58, 59]. Tabela 3.5 przedstawia pytania z kwestionariusza wraz z indywidualnymi i usrednionymi odpowiedziami użytkowników oraz końcowe wyniki skali użyteczności systemu. Usredniony wynik skali użyteczności wyniósł 87.875 punktu, co wskazuje na wysoką użyteczność zaimplementowanego systemu. W szczególności pozytywnie oceniono łatwość użycia i niski poziom skomplikowania systemu, co było jednym z wymagań niefunkcjonalnych systemu.

Tabela 3.5: Wyniki kwestionariusza skali użyteczności systemu

Nr	Pytanie	Odpowiedzi użytkowników				
		1.	2.	3.	4.	Średnia
1	Myślę, że chciałbym często korzystać z tego systemu.	3	4	4	4	3.75
2	Uważam, że system był niepotrzebnie skomplikowany.	1	1	1	1	1
3	Uważam, że system był łatwy w użyciu.	5	5	4	5	4.75
4	Myślę, że potrzebowałbym wsparcia technicznego, aby móc korzystać z tego systemu.	1	1	1	1	1
5	Uważam, że różne funkcje w tym systemie były dobrze ze sobą zintegrowane.	3	4	4	3	3.5
6	Uważam, że w tym systemie było zbyt wiele niespójności.	2	2	2	1	1.75
7	Wyobrażam sobie, że większość osób nauczyłaby się korzystać z tego systemu bardzo szybko.	5	5	5	5	5
8	Uważam, że korzystanie z systemu było uciążliwe.	2	1	2	2	1.75
9	Czułem się bardzo pewnie korzystając z systemu.	5	4	4	4	4.25
10	Musiałem nauczyć się wielu rzeczy, zanim mogłem zacząć korzystać z tego systemu.	1	1	1	1	1
Wynik skali użyteczności systemu		85	90	85	87.5	86.875

Zebrane uwagi i spostrzeżenia przedyskutowano z użytkownikami i wykorzystano do dalszej analizy i planowania rozwoju systemu. Użytkownicy nie napotkali znaczących problemów z wydajnością i płynnością systemu. Pozytywnie oceniono łatwość obsługi funkcji związanych z zarządzaniem użytkownikami w systemie, przestrzeniami roboczymi i ich członkami. Doceniono możliwość przesyłania plików do przestrzeni tle bez zakłócania pracy oraz funkcję szybkiego podglądu plików bez konieczności ich, zabrakło natomiast możliwości szybkiego przenoszenia plików pomiędzy katalogami poprzez ich przeciąganie i upuszczanie. Użytkownicy stwierdzili, że system pozwala na sprawną wymianę informacji i materiałów cyfrowych wykorzystywanych w tworzeniu i nauce muzyki bez konieczności korzystania z wielu narzędzi i aplikacji.

Współdzielony edytor audio użytkownicy określili jako rozwiązanie użyteczne zwłaszcza do szybkiego przeprowadzania sesji improwizacyjnych, wymiany i sprawdzania pomysłów muzycznych pomiędzy spotkaniami na miejscu. Zwrócili również uwagę na potencjał systemu w prowadzeniu zajęć nauki gry na instrumentach muzycznych uzupełniających zajęcia stacjonarne. W kontekście prób muzycznych był postrzegany raczej jako dostateczny zamiennik w przypadku braku możliwości spotkania się na żywo. Jedną z sugestii użytkowników było dodanie możliwości nagrywania wideo wraz z dźwiękiem w edytorze audio, co mogłoby przełożyć się na stworzenie wierniejszego wrażenia wspólnego wykonywania utworu ze względu na możliwość reagowania na mowę ciała innych członków zespołu. Obecność graficznej wizualizacji dźwiękowej ułatwiała synchronizację ścieżek dźwiękowych nagrywanych w trakcie sesji. W trakcie nagrywania ścieżek konieczne było użycie słuchawek w celu uniknięcia nagrania dźwięku innych ścieżek z głośników, zasugerowano przy tym dodanie do edytora opcjonalnej funkcji odtwarzania aktualnie nagrywanego wejścia audio w celu lepszego monitorowania własnego brzmienia. Kolejną istotną sugestią było dodanie możliwości tworzenia i edycji zapisu nutowego oraz wyświetlania go w edytorze audio, co mogłoby ułatwić zadania takie jak nauka nowych utworów czy komponowanie. W edytorze zabrakło możliwość przełączenia urządzenia wejściowego, co przeszkodziło jednemu z użytkowników wykorzystaniu instrumentu z własnym wejściem audio do komputera, problem udało się obejść poprzez tymczasowe przełączenie domyślnego urządzenia wejściowego w systemie operacyjnym. Użytkownicy docenili możliwość wspólnego odsłuchu nagranego utworu w wideokonferencji, która pozwoliła na szybką wymianę uwag i pomysłów na temat nagranego materiału. Kolejną sugestią było odseparowanie sesji spotkaniowych od projektów audio, co pozwoliłoby na lepszą organizację pracy i możliwość przełączania się pomiędzy projektami bez konieczności zamknięcia i otwierania nowych sesji. Zwrócono również uwagę na brak możliwości pobrania wynikowego pliku audio na dysk bez konieczności zapisywania go w przestrzeni roboczej.

Forum zostało ocenione jako przydatne, choć zwrócono uwagę na kilka mankamentów. Użytkownicy zauważyli, że forum nie aktualizuje się w czasie rzeczywistym, co utrudniało szybką wymianę informacji. Zabrakło możliwości załączenia pliku do komentarzy oraz załączania plików spoza przestrzeni roboczej, co wymuszało dodatkowy krok, w postaci wcześniejszego wrzucenia pliku do magazynu przestrzeni roboczej. Użytkownicy zasugerowali również dodanie do strony głównej panelu z powiadomieniami, najnowszymi postami i nadchodzącyimi spotkaniami, co ułatwiłoby śledzenie aktywności w systemie. Narzędzie do nagrywania materiałów wideo zostało ocenione jako potencjalnie przydatny dodatek w kontekście edukacji, choć zabrało w nim możliwości edycji nagranego materiału wideo.

4. PODSUMOWANIE

Celem pracy było opracowanie i zaimplementowanie internetowego systemu wspomagającego współpracę i edukację muzyczną na odległość, który nie wymaga od użytkowników specjalistycznego sprzętu, infrastruktury sieciowej oraz wysokich umiejętności technicznych. W trakcie realizacji pracy przeprowadzono przegląd obecnego stanu wiedzy i rozwiązań technicznych w dziedzinie komputerowych systemów współpracy z naciskiem na współpracę i edukację muzyczną.

Na podstawie przeglądu zaproponowano koncepcję systemu i przeprowadzono analizę wymagań, wykorzystując do tego ramy teoretyczne dziedziny pracy grupowej wspomaganej komputerowo. Zaproponowany system ma zapewnić użytkownikom jednolite i zintegrowane środowisko do wymiany informacji i materiałów cyfrowych za pomocą znajomych narzędzi i interfejsów oraz pozwolić na wspólną pracę nad utworami muzycznymi. System stara się znieść wpływ opóźnień i zmiennej jakości transmisji dźwięku, będących główną barierą w zdalnym wykonywaniu utworów muzycznych, poprzez integrację internetowego edytora audio z wideokonferencją pozwalającą na przyrostową pracę nad utworem z możliwością wspólnego odsłuchu rezultatów w trakcie spotkania. Następnie określono wysokopoziomową architekturę systemu oraz dobrano rozwiązania techniczne i narzędzia programistyczne, które zostały wykorzystane do implementacji systemu.

W ramach implementacji został stworzony prototyp systemu, który realizuje cele niniejszej pracy, spełnia wymagania postawione w trakcie analizy oraz założenia architektoniczne określone na etapie projektowania. System został zaimplementowany jako progresywna aplikacja internetowa dostępna na urządzeniach stacjonarnych oraz mobilnych. Prototyp systemu wdrożono i udostępniono w sieci Internet w celu przeprowadzenia testów użytkowych w warunkach rzeczywistych. Testy użytkowe przeprowadzono z pomocą grupy czterech osób aktywnie związanych z dziedziną muzyki. Użyteczność systemu oceniono za pomocą kwestionariusza skali użyteczności systemu, w którym uzyskał wynik 86.875 punktu wskazujący na wysoką użyteczność. W ramach testów przeanalizowano również uwagi i spostrzeżenia użytkowników odnośnie do poszczególnych funkcjonalności systemu. Według użytkowników system jest prosty w obsłudze, pozwala na efektywną komunikację i wymianę materiałów cyfrowych wykorzystywanych w tworzeniu i nauce muzyki bez konieczności przełączania się między różnymi aplikacjami. Użytkownicy wskazali, że system znaczco ułatwia zdalne przeprowadzanie improwizacji i sprawdzanie pomysłów muzycznych oraz w mniejszym stopniu organizowanie prób muzycznych przygotowujących do występu. Zasugerowano również zmiany, jakie należałoby wprowadzić w celu polepszenia systemu w kontekście prób muzycznych, walorów edukacyjnych i ogólnej użyteczności, które zostały uwzględnione w planach dalszego rozwoju systemu.

4.1. *Plany dalszego rozwoju systemu*

Na podstawie wyników testów użytkowych i zebranych uwag użytkowników przygotowano plany dalszego rozwoju systemu. Jednym z głównych usprawnień, które należy wprowadzić, jest odseparowanie projektów muzycznych od sesji spotkaniowych oraz rozszerzenie możliwości edytora audio. W aktualnej wersji systemu, projekty są powiązane z konkretnymi sesjami spotkaniowymi, co ogranicza elastyczność pracy. Rozdzielenie projektów od sesji pozwoliłoby użytkownikom na kontynuację pracy na przestrzeni wielu spotkań. Edytor audio można by rozszerzyć o możliwość wersjonowania projektów i ścieżek, aby umożliwić użytkownikom cofanie i przywracanie zmian oraz porównywanie różnych wersji

utworu. Kolejnym krokiem rozwoju może być dodanie do edytora możliwości nagrywania ścieżek wideo wraz z dźwiękiem. Taka funkcjonalność mogłaby potencjalnie stworzyć lepsze wrażenie wspólnego wykonywania utworu oraz ułatwić synchronizację muzyków dzięki możliwości obserwacji mowy ciała innych członków grupy. Istotnym aspektem, który powinien zostać uwzględniony w dalszym rozwoju systemu, jest dodanie opcjonalnej możliwości monitorowania własnego wejścia audio w trakcie nagrywania ścieżek, co pozwoliłoby użytkownikom na lepsze kontrolowanie swojego brzmienia. W planach uwzględniono również rozszerzenie systemu o edytor zapisu nutowego i integrację go z edytorem audio. Taka funkcjonalność mogłaby znacząco ułatwić zadania takie jak nauka nowych utworów, tworzenie aranżacji czy komponowanie, które są istotnymi elementami edukacji i współpracy muzycznej. Warto zbadać również możliwość dodania do edytora obsługi protokołu MIDI, co pozwoliłoby na pracę z instrumentami elektronicznymi i wirtualnymi.

Aby poprawić doświadczenia użytkowników związane ze śledzeniem aktywności w systemie, planowane jest dodanie na stronie głównej aplikacji panelu przestawiającego informacje takie jak nadchodzące sesje spotkanie i aktywności innych użytkowników w przestrzeniach roboczych. Dodatkowo w celu polepszenia płynności pracy z systemem, planowane jest dodanie aktualizacji w czasie rzeczywistym do większej ilości komponentów systemu. Dotyczy to takich elementów jak forum dyskusyjne czy przeglądarka plików, które aktualnie odświeżane są za pomocą przycisku lub ponowne przejście do komponentu. Zgodnie z sugestiami użytkowników forum dyskusyjne powinno umożliwiać załączanie plików również spoza przestrzeni roboczych oraz załączanie ich do komentarzy. Forum można również rozszerzyć o możliwość tworzenia wątków w celu lepszej organizacji dyskusji, zwłaszcza w przypadku dużych grup. Przydatne mogą okazać się również funkcje takie jak oznaczanie w postach i komentarzach użytkowników.

Aby polepszyć walory edukacyjne systemu, planowane jest dodanie większej ilości funkcji znanych z systemów LMS, takich jak tworzenie i przypisywanie zadań z możliwością ich oceny, na przykład w formie zadań praktycznych polegających na zamieszczeniu przez ucznia materiału wideo z wykonaniem utworu. Funkcjonalność można zintegrować z dostępnym w systemie edytorem materiałów wideo, który należy rozszerzyć o możliwość edycji nagrania. Koniecznym dla dalszego rozwoju systemu krokiem jest przeprowadzenie testów na większych i bardziej zróżnicowanych grupach użytkowników, zwłaszcza w kontekście edukacyjnym. Testy te pozwolą na zebranie cennych informacji zwrotnych oraz ocenę funkcjonalności systemu w kontekście różnych scenariuszy użycia.

W kolejnych fazach rozwoju systemu warto również zastanowić się nad wykorzystaniem bardziej zaawansowanych modeli architektonicznych w celu zwiększenia wydajności, skalowalności oraz elastyczności systemu. Jednym z możliwych kierunków rozwoju jest zastosowanie architektury opartej na usługach w celu zdekomponowania logiki domenowej systemu na mniejsze, niezależne od siebie podsystemy, które mogą być rozwijane osobno i skalowane horyzontalnie w zależności od potrzeb.

WYKAZ LITERATURY

- [1] P. Wilson, „Computer supported cooperative work: an overview,” *Intelligent Tutoring Media*, t. 1, nr. 3, s. 103–116, 1990.
- [2] K. Schmidt, „Constructing CSCW: The first quarter century,” *Computer Supported Cooperative Work*, t. 22, s. 345–372, sierp. 2013.
- [3] A. Harris, D. Gómez-Zará, L. Dechurch i N. Contractor, „Joining Together Online: The Trajectory of CSCW Scholarship on Group Formation,” *Proceedings of the ACM on Human-Computer Interaction*, t. 3, s. 1–27, list. 2019.
- [4] „Chapter 11 - Groupware and Computer-Supported Cooperative Work,” *Readings in Human-Computer Interaction*, ser. Interactive Technologies, R. M. BAECKER, J. GRUDIN, W. A. BUXTON i S. GREENBERG, red., Morgan Kaufmann, 1995, s. 741–753.
- [5] R. McDermott, M. Daniels, Å. Cajander, J. Bass i J. Lalchandani, „A comparative analysis of two globally distributed group projects: A perspective from CSCW/CSCL research,” *2015 IEEE Frontiers in Education Conference (FIE)*, IEEE, 2015, s. 1–9.
- [6] C. P. Lee i D. Paine, „From The matrix to a model of coordinated action (MoCA) A conceptual framework of and for CSCW,” *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, lut. 2015, s. 179–194.
- [7] G. Petit i J. Soles, „From Two CSCW Frameworks to User Requirements Definition for a Retail Planning Collaborative Software,” *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2016, s. 690–699.
- [8] V. Shurygin, N. Saenko, A. Zekiy, E. Klochko i M. Kulapov, „Learning management systems in academic and corporate distance education,” *International Journal of Emerging Technologies in Learning (iJET)*, t. 16, nr. 11, s. 121–139, 2021.
- [9] D. J. Lemay, P. Bazelais i T. Doleck, „Transition to online learning during the COVID-19 pandemic,” *Computers in human behavior reports*, t. 4, s. 100130, 2021.
- [10] M. Ouadoud, N. Rida i T. Chafiq, „Overview of E-learning Platforms for Teaching and Learning,” *Int. J. Recent Contributions Eng. Sci. IT*, t. 9, nr. 1, s. 50–70, 2021.
- [11] R. Kraleva, M. Sabani i V. Kralev, „An analysis of some learning management systems,” *International Journal on Advanced Science, Engineering and Information Technology*, t. 9, nr. 4, s. 1190–1198, 2019.
- [12] V. M. Bradley, „Learning Management System (LMS) use with online instruction..,” *International Journal of Technology in Education*, t. 4, nr. 1, s. 68–92, 2021.
- [13] D. Turnbull, R. Chugh i J. Luck, „Learning management systems, an overview,” *Encyclopedia of education and information technologies*, s. 1052–1058, 2020.
- [14] T. Koutsoupidou, „Online distance learning and music training: Benefits, drawbacks and challenges,” *Open Learning: The Journal of Open, Distance and e-Learning*, t. 29, nr. 3, s. 243–255, 2014.
- [15] A. Ververis i A. Apostolis, „Online music education in the era of COVID-19: Teaching instruments in public music secondary schools of Greece during the 2020 lockdown,” *International Conference on Studies in Education and Social Sciences*, t. 3, 2020, s. 1–9.
- [16] P. M. Hash, „Remote learning in school bands during the COVID-19 shutdown,” *Journal of Research in Music Education*, t. 68, nr. 4, s. 381–397, 2021.

- [17] D. Calderón-Garrido i J. Gustems-Carnicer, „Adaptations of music education in primary and secondary school due to COVID-19: The experience in Spain,” *Music Education Research*, t. 23, nr. 2, s. 139–150, 2021.
- [18] A. M. Hernández, „Online learning in higher music education: Benefits, challenges and drawbacks of one-to-one videoconference instrumental lessons,” *Journal of Music, Technology & Education*, t. 13, nr. 2-3, s. 181–197, 2020.
- [19] M. D. Rucsanda, A. Belibou i A.-M. Cazan, „Students’ attitudes toward online music education during the COVID 19 lockdown,” *Frontiers in psychology*, t. 12, s. 753–785, 2021.
- [20] M. Wieser i F. Müller, „Motivation in Instrumental Music Instruction Before and During the Remote Learning Phase Due to COVID-19 Crisis,” *Music & Science*, t. 5, s. 1–13, list. 2022.
- [21] A. Goodrich, „Online peer mentoring and remote learning,” *Music Education Research*, t. 23, nr. 2, s. 256–269, 2021.
- [22] D. Kokotsaki i S. Hallam, „The perceived benefits of participative music making for non-music university students: a comparison with music students,” *Music Education Research*, t. 13, nr. 2, s. 149–172, 2011.
- [23] C. Koopman, „Community music as music education: On the educational potential of community music,” *International Journal of Music Education*, t. 25, nr. 2, s. 151–163, 2007.
- [24] L. Gabrielli, S. Squartini, L. Gabrielli i S. Squartini, *Wireless networked music performance*. Springer, 2016.
- [25] R. Wilson, „Aesthetic and technical strategies for networked music performance,” *AI & Society*, t. 38, nr. 5, s. 1871–1884, 2023.
- [26] C. Rottandi, C. Chafe, C. Allocchio i A. Sarti, „An overview on networked music performance technologies,” *IEEE Access*, t. 4, s. 8823–8843, 2016.
- [27] A. Carôt i C. Werner, „Fundamentals and principles of musical telepresence,” *Journal of Science and Technology of the Arts*, t. 1, nr. 1, s. 26–37, 2009.
- [28] C. Drioli i C. Allocchio, „LOLA: a low-latency high quality A/V streaming system for networked performance and interaction,” *Colloqui informatica musicale, Trieste*, 2012.
- [29] C. Alexandraki i in., „Towards the implementation of a generic platform for networked music performance: The DIAMOUSES approach,” *ICMC*, 2008, s. 251–258.
- [30] C. Alexandraki i D. Akoumanakis, „Exploring new perspectives in network music performance: The DIAMOUSES framework,” *Computer Music Journal*, t. 34, nr. 2, s. 66–83, 2010.
- [31] D. Kemack Goot i W. S. Deal, „A Spectrum of Online Rehearsal Applications: A Potential Means for Cultural Connection,” *Journal of Network Music and Arts*, t. 4, nr. 1, s. 4, 2022.
- [32] D. Kerr, „MIDI: the musical instrument digital interface,” *Cleveland, OH*, 2009.
- [33] R. Renwick, „Sourcenode: a network sourced approach to network music performance (nmp),” *ICMC*, 2012.
- [34] R. E. Saputra i A. S. Prihatmanto, „Design and implementation of beatme as a networked music performance (nmp) system,” *2012 International Conference on System Engineering and Technology (ICSET)*, IEEE, 2012, s. 1–6.
- [35] A. P. Kefauver i D. Patschke, *Fundamentals of digital audio*. AR Editions, Inc., 2007, t. 22.
- [36] M. Marrington i in., „Composing with the digital audio workstation,” *The singer-songwriter handbook*, s. 77–89, 2017.
- [37] S. Stickland, R. Athauda i N. Scott, „Design and evaluation of a scalable real-time online digital audio workstation collaboration framework,” *Journal of the Audio Engineering Society*, t. 69, nr. 6, s. 410–431, 2021.

- [38] J. Galván i M. Clauhs, „The virtual choir as collaboration,” *The Choral Journal*, t. 61, nr. 3, s. 8–19, 2020.
- [39] A. I. Uyub, „Digital Audio Workstation (DAW) as a Platform of Creative Musical Performance Experience,” *KUPAS SENI*, t. 10, s. 52–55, 2022.
- [40] K. Holdhus, C. Christophersen i H. Partti, „Soundtrapped? Socio-material perspectives on collaborative teaching within the music classroom,” *Research Studies in Music Education*, t. 45, nr. 3, s. 539–554, 2023.
- [41] M. Richards i N. Ford, *Fundamentals of software architecture: an engineering approach*. O'Reilly Media, 2020.
- [42] J. Ingino, *Software Architect's Handbook: Become a successful software architect by implementing effective architecture concepts*. Packt Publishing Ltd, 2018.
- [43] *Progressive Web Apps*, data dost. 12.02.2024, paź. 2023. adr.: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps.
- [44] *WebAssembly Introduction*, data dost. 12.02.2024, grud. 2022. adr.: <https://webassembly.github.io/spec/core/intro/introduction.html>.
- [45] *Blazor*, data dost. 25.02.2024, lut. 2024. adr.: <https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-7.0>.
- [46] *ASP.NET core blazor hosting models*, sty. 2023. adr.: <https://learn.microsoft.com/en-us/aspnet/core/blazor/hosting-models>.
- [47] *ASP.NET Core*, data dost. 26.02.2024. adr.: <https://github.com/dotnet/aspnetcore>.
- [48] *Entity Framework Core*, data dost. 12.03.2024. adr.: <https://github.com/dotnet/efcore>.
- [49] *SignalR*, data dost. 27.03.2024. adr.: <https://github.com/dotnet/aspnetcore/tree/main/src/SignalR>.
- [50] *Keycloak*, data dost. 13.05.2024. adr.: <https://www.keycloak.org>.
- [51] *The OAuth 2.0 Authorization Framework*, data dost. 12.04.2024, paź. 2012. adr.: <https://datatracker.ietf.org/doc/html/rfc6749>.
- [52] O. Foundation, *OpenID Connect*, data dost. 13.04.2024. adr.: <https://openid.net/developers/how-connect-works>.
- [53] *JSON Web Token (JWT)*, data dost. 12.04.2024, maj 2015. adr.: <https://datatracker.ietf.org/doc/html/rfc7519>.
- [54] *The OAuth 2.0 Authorization Framework: Bearer Token Usage*, data dost. 12.04.2024, paź. 2012. adr.: <https://datatracker.ietf.org/doc/html/rfc6750>.
- [55] *Jitsi Meet*, data dost. 19.03.2024. adr.: <https://github.com/jitsi/jitsi-meet>.
- [56] *PostgreSQL*, data dost. 12.05.2024. adr.: <https://www.postgresql.org/docs/15/index.html>.
- [57] N. Aro, *Waveform Playlist*, mar. 2023. adr.: <https://github.com/naomiaro/waveform-playlist>.
- [58] A. Bangor, P. T. Kortum i J. T. Miller, „An empirical evaluation of the system usability scale,” *Intl. Journal of Human–Computer Interaction*, t. 24, nr. 6, s. 574–594, 2008.
- [59] J. Sauro i J. R. Lewis, *Quantifying the user experience: Practical statistics for user research*. Morgan Kaufmann, 2016.

WYKAZ RYSUNKÓW

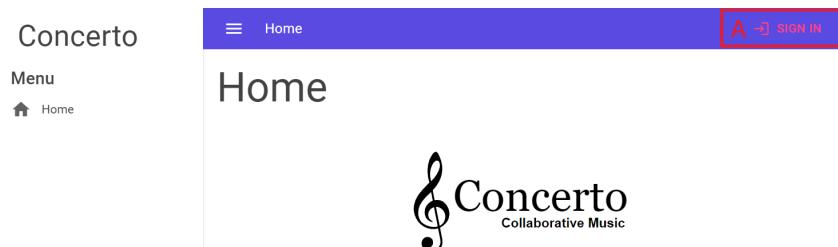
2.1	Macierz klasyfikacji techniki wykorzystywanej w systemach CSCW według kryteriów przestrzeni i czasu [4]	11
2.2	Wymiary modelu Model of Coordinated Action [6]	13
3.1	Ogólny diagram przypadków użycia systemu	22
3.2	Diagram przypadków użycia przestrzeni roboczej	23
3.3	Diagram domeny systemu	24
3.4	Diagram ogólnej architektury systemu	26
3.5	Przykładowy proces tworzenia tokenu JWT	34
3.6	Diagram architektury aplikacji klienckiej	38
3.7	Strona główna aplikacji	39
3.8	Panel administracyjny	40
3.9	Przestrzeń robocza	41
3.10	Przesyłanie plików	42
3.11	Widok głównej zakładki sesji spotkaniowej	43
3.12	Narzędzie do nagrywania materiałów wideo	45
3.13	Wybrane widoki wersji mobilnej aplikacji klienckiej	46
3.14	Diagram architektury serwera aplikacji	47
3.15	Diagram wdrożenia systemu	52
3.16	Sesja spotkaniowa w trakcie testów	52

WYKAZ TABEL

3.1	Elementy modelu domeny systemu	21
3.2	Kontrolery API serwera aplikacji	49
3.3	Serwisy logiki domenowej serwera aplikacji	50
3.4	Parametry maszyny wirtualnej	51
3.5	Wyniki kwestionariusza skali użyteczności systemu	53

DODATEK A: INSTRUKCJA OBSŁUGI SYSTEMU

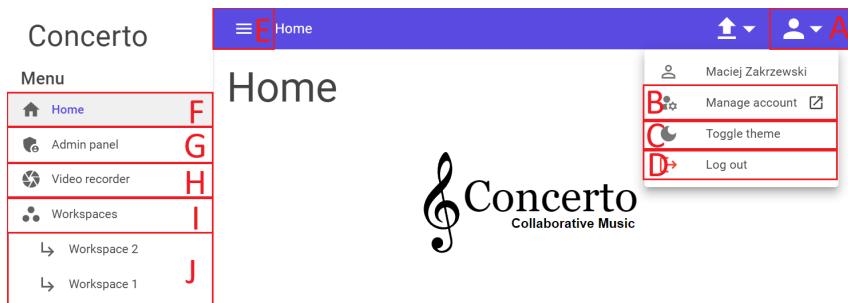
Dostęp do systemu



Rysunek A.1: Strona główna przed zalogowaniem

The screenshot shows a login form titled "Sign in to your account". It contains two input fields: "Username or email" (labeled A) and "Password" (labeled A). Below these fields are two buttons: "Remember me" (labeled B) and "Forgot Password?" (labeled E). At the bottom of the form is a large blue "Sign in" button (labeled C). At the very bottom, there is a link "New user? Register" (labeled D).

Rysunek A.2: Formularz logowania



Rysunek A.3: Strona główna po zalogowaniu

Rejestracja

W celu rejestracji w systemie należy wykonać następujące kroki:

1. Kliknąć przycisk „Sign In” znajdujący się w prawym górnym rogu strony (Rys. A.1 – etykieta A) w celu otwarcia formularza logowania.
2. Kliknąć przycisk „Register” znajdujący się na dole formularza logowania (Rys. A.2 – etykieta D) w celu otwarcia formularza rejestracji.

3. Wprowadzić dane do formularza rejestracji takie jak imię, nazwisko, adres e-mail, nazwa użytkownika oraz hasło.
4. Kliknąć przycisk „Register” znajdujący się na dole formularza rejestracji. W przypadku błędu należy wrócić do kroku 3. i poprawić błędy.
5. Po poprawnej rejestracji należy kliknąć link aktywacyjny otrzymany w wiadomości e-mail. Po kliknięciu linku konto zostanie aktywowane i użytkownik automatycznie zalogowany do systemu.
6. Po rejestracji należy zaczekać na zatwierdzenie konta przez administratora systemu w celu uzyskania dostępu do wszystkich funkcji systemu.

Logowanie

Aby zalogować się do systemu należy wykonać następujące kroki:

1. Kliknąć przycisk „Sign In” znajdujący się w prawym górnym rogu strony (Rys. A.1 – etykieta A) w celu otwarcia formularza logowania.
2. Wprowadzić nazwę użytkownika i hasło do formularza logowania (Rys. A.2 – etykieta A).
3. Opcjonalnie zaznaczyć pole „Remember me” (Rys. A.2 – etykieta B), aby przedłużyć czas trwania sesji.
4. Kliknąć przycisk „Sign In” (Rys. A.2 – etykieta C). Po poprawnym zalogowaniu użytkownik zostanie przekierowany do strony głównej systemu. W przypadku wyświetlenia komunikatu o błędnych danych należy wrócić do kroku 2. i poprawić błędy.
5. Po poprawnym zalogowaniu użytkownik zostanie przekierowany do strony głównej systemu w stanie zalogowanym. W przypadku konta niezatwierdzonego przez administratora systemu zostanie wyświetlony komunikat informujący o tym fakcie.

Resetowanie hasła

Aby resetować hasło do konta w przypadku jego utraty należy wykonać następujące kroki:

1. Kliknąć przycisk „Sign In” znajdujący się w prawym górnym rogu strony (Rys. A.1 – etykieta A) w celu otwarcia formularza logowania.
2. Kliknąć przycisk „Forgot Password?” znajdujący się na formularzu logowania (Rys. A.1 – etykieta E).
3. Wprowadzić adres e-mail lub nazwę użytkownika do formularza resetowania hasła i kliknąć przycisk „Submit”. Na adres e-mail związanego z kontem zostanie wysłana wiadomość z linkiem resetującym hasło.
4. Kliknąć link resetujący hasło w otrzymanej wiadomości e-mail. Po kliknięciu linku zostanie otwarty formularz zmiany hasła.
5. Wprowadzić nowe hasło do formularza zmiany hasła i kliknąć przycisk „Submit”. Po poprawnym zresetowaniu hasła użytkownik zostanie automatycznie zalogowany do systemu.

Wylogowanie

Aby wylogować się z systemu należy wykonać następujące kroki:

1. Rozwinąć menu użytkownika klikając na ikonę użytkownika w prawym górnym rogu strony (Rys. A.3 – etykieta A).

- Wybrać opcję „Log out” z rozwijanego menu (Rys. A.3 – etykieta D). Po kliknięciu opcji użytkownik zostanie wylogowany i przekierowany do strony głównej systemu w stanie niezalogowanym (Rys. A.1).

Panel administracyjny

Name	Surname ↑	Username	Email	Role	Actions
Jakub	Przykładowy	jakub	C yatod95515@obisims.com	Unverified	E ✓ X
Krzysztof	Testowy	krzysztof	D hosojiy646@janfab.com	Unverified	F

Rysunek A.4: Panel administracyjny – widok niezatwierdzonych użytkowników

Name	Surname ↑	Username	Email	Role	Actions
Jan	Nowakowski	janek	A jannowakowski665@gmail.com	Administrator	A B
Krzysztof	Nowy	krzysztof22	xijifoc754@leacore.com	User	B

Rysunek A.5: Panel administracyjny – widok zatwierdzonych użytkowników

Zatwierdzanie nowych użytkowników

Aby zatwierdzić nowego użytkownika w systemie należy wykonać następujące kroki:

- Zalogować się do systemu jako użytkownik z uprawnieniami administratora.
- Jeśli menu boczne nie jest widoczne, rozwinąć je klikając na ikonę menu po lewej stronie nagłówka strony (Rys. A.3 – etykieta E).
- Wybrać w menu bocznym opcję „Admin panel” (Rys. A.3 – etykieta G). Po kliknięciu opcji zostanie otwarty panel administracyjny.
- W panelu administracyjnym w domyślnej zakładce „Unverified users” (Rys. A.4 – etykieta A) zostaną wyświetleni niezatwierdzeni użytkownicy.
- Opcjonalnie można filtrować użytkowników po nazwie użytkownika, imieniu, nazwisku lub adresie e-mail wpisując odpowiednie dane w polu wyszukiwania (Rys. A.4 – etykieta G).
- Warunkiem koniecznym do zatwierdzenia użytkownika jest potwierdzenie przez niego adresu e-mail. Zielony symbol obok adresu e-mail (Rys. A.4 – etykieta C) oznacza, że użytkownik potwierdził adres e-mail, a czerwony symbol (Rys. A.4 – etykieta D) oznacza, że adres e-mail nie został potwierdzony.
- Kliknąć przycisk zielony przycisk z ikoną ptaszka (Rys. A.4 – etykieta E) znajdujący się obok wiersza użytkownika, który ma zostać zatwierdzony. Następnie potwierdzić chęć zatwierdzenia użytkownika w oknie dialogowym.

8. Nowych użytkowników można odrzucać klikając przycisk czerwony przycisk z ikoną krzyżyka (Rys. A.4 – etykieta F) znajdujący się obok wiersza użytkownika, który ma zostać odrzucony. Następnie potwierdzić chęć odrzucenia użytkownika w oknie dialogowym. Odrzucony użytkownik zostanie usunięty z systemu.

Zarządzanie użytkownikami

Aby zmienić rolę użytkownika w obrębie systemu lub usunąć go należy wykonać następujące kroki:

1. Zalogować się do systemu jako użytkownik z uprawnieniami administratora.
2. Jeśli menu boczne nie jest widoczne, rozwinąć je klikając na ikonę menu po lewej stronie nagłówka strony (Rys. A.3 – etykieta E).
3. Wybrać w menu bocznym opcję „Admin panel” (Rys. A.3 – etykieta G). Po kliknięciu opcji zostanie otwarty panel administracyjny.
4. Przejść do zakładki „All users” (Rys. A.4 – etykieta B). W tej zakładce znajdują się wszyscy zatwierdzeni użytkownicy systemu.
5. Aby zmienić rolę użytkownika, kliknąć niebieski przycisk z ikoną kółka zębatego (Rys. A.5 – etykieta A) znajdujący się obok wiersza użytkownika, którego rolę chcemy zmienić. W oknie dialogowym wybrać nową rolę użytkownika i potwierdzić zmianę.
6. Aby usunąć użytkownika z systemu, kliknąć czerwony przycisk z symbolem użytkownika z minusem (Rys. A.5 – etykieta B) znajdujący się obok wiersza użytkownika, który ma zostać usunięty. Następnie potwierdzić chęć usunięcia użytkownika w oknie dialogowym. Usunięty użytkownik zostanie usunięty z systemu.

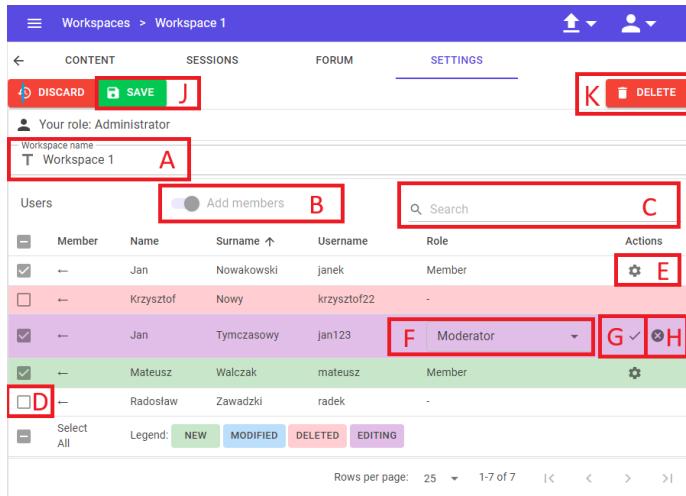
Przestrzenie robocze

Your workspaces		Search	Creation date ↓
	Workspace 2	23.08.2024 16:12:39	C
	Workspace 1	23.08.2024 16:10:13	
Rows per page: 25		1-4 of 4	< > >>

Rysunek A.6: Lista przestrzeni roboczych

Workspaces > Workspace 1				
A CONTENT	B SESSIONS	C FORUM	D SETTINGS	
	Root			
Name ↑	Format	Type	Size	Owner
	Folder		Other	Maciej

Rysunek A.7: Przestrzeń robocza



Rysunek A.8: Ustawienia przestrzeni roboczej

Tworzenie przestrzeni roboczych

Aby utworzyć nową przestrzeń roboczą należy wykonać następujące kroki:

1. Zalogować się do systemu jako użytkownik z uprawnieniami moderatora lub administratora.
2. Jeśli menu boczne nie jest widoczne, rozwinąć je klikając na ikonę menu po lewej stronie nagłówka strony (Rys. A.3 – etykieta E).
3. Wybrać w menu bocznym opcję „Workspaces” (Rys. A.3 – etykieta I). Po kliknięciu opcji zostanie otwarta lista przestrzeni roboczych.
4. Kliknąć przycisk z ikoną plusa (Rys. A.6 – etykieta A) znajdujący się w prawym górnym rogu listy przestrzeni roboczych. Po kliknięciu przycisku zostanie otwarty formularz tworzenia przestrzeni roboczej.
5. W formularzu wprowadzić nazwę przestrzeni roboczej i zatwierdzić działanie klikając przycisk „Create”.

Otwarcie przestrzeni roboczej

Aby otworzyć przestrzeń roboczą należy wykonać następujące kroki:

1. Jeśli menu boczne nie jest widoczne, rozwinąć je klikając na ikonę menu po lewej stronie nagłówka strony (Rys. A.3 – etykieta E).
2. Wybrać w menu bocznym przestrzeń roboczą, która ma zostać otwarta (Rys. A.3 – etykieta J).
3. Przestrzeń roboczą można również otworzyć z poziomu listy przestrzeni roboczych. W tym celu należy przejść do listy przestrzeni roboczych klikając w menu bocznym na opcję „Workspaces” (Rys. A.3 – etykieta I).
4. Opcjonalnie wpisać nazwę przestrzeni roboczej w polu wyszukiwania w celu filtrowania przestrzeni roboczych.
5. Kliknąć na wiersz przestrzeni roboczej, aby ją otworzyć (Rys. A.6 – etykieta C).

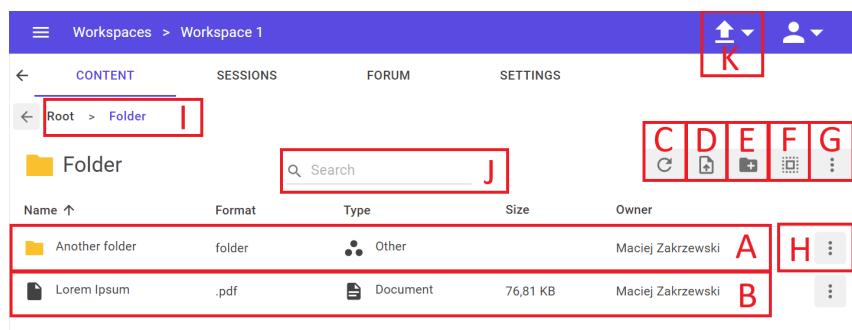
Zarządzanie przestrzenią roboczą

Aby zarządzać przestrzenią roboczą i jej członkami należy wykonać następujące kroki:

1. Otworzyć przestrzeń roboczą zgodnie z instrukcją z poprzedniej sekcji.

2. Wybrać zakładkę „Settings” (Rys. A.7 – etykieta D) znajdująca się w górnej części strony przestrzeni roboczej. Zostanie wyświetlony panel ustawień przestrzeni roboczej przedstawiający informacje o przestrzeni roboczej oraz listę jej członków.
3. Dalsze kroki opisują poszczególne możliwe działania w zakładce „Settings”. Aby zapisać wprowadzone zmiany, należy kliknąć przycisk „Save” (Rys. A.8 – etykieta J). Aby je odrzucić należy kliknąć przycisk „Discard” (Rys. A.8 – etykieta I).
4. Aby zmienić nazwę przestrzeni roboczej, wpisać nową nazwę w polu „Workspace name” (Rys. A.8 – etykieta A).
5. Aby usunąć przestrzeń roboczą, kliknąć przycisk „Delete” (Rys. A.8 – etykieta K). Po potwierdzeniu chęci usunięcia przestrzeni roboczej zostanie ona usunięta z systemu.
6. Aby przefiltrować listę członków przestrzeni roboczej, należy wpisać nazwę użytkownika, imię lub nazwisko w polu wyszukiwania (Rys. A.8 – etykieta C).
7. Aby włączyć możliwość dodawania nowych członków należy przełączyć kontrolkę „Add members” (Rys. A.8 – etykieta B). Na liście użytkowników pojawią się wpisy z użytkownikami systemu, którzy nie są jeszcze członkami przestrzeni roboczej. Aby dodać lub usunąć użytkownika z przestrzeni roboczej należy zaznaczyć/odznaczyć pole wyboru obok jego nazwy (Rys. A.8 – etykieta D). Wpis zostanie podświetlony na zielony jeśli użytkownik zostanie dodany lub na czerwony jeśli zostanie usunięty.
8. Aby zmienić rolę użytkownika w obrębie przestrzeni roboczej, należy kliknąć ikonę koła zębatego (Rys. A.8 – etykieta E) znajdującą się obok wiersza użytkownika, którego rolę chcemy zmienić. Edytowany użytkownik zostanie podświetlony na fioletowo. Następnie wybrać nową rolę użytkownika z listy rozwijanej (Rys. A.8 – etykieta F) i zatwierdzić zmianę za pomocą ikony ptaszka (Rys. A.8 – etykieta G). Zmianę można odrzucić za pomocą ikony krzyżyka (Rys. A.8 – etykieta H). Użytkownik ze zmodyfikowaną rolą zostanie podświetlony na niebiesko.

Katalogi i pliki



Rysunek A.9: Przeglądarka plików

Przeglądanie i zarządzanie plikami i katalogami

Aby przeglądać i zarządzać plikami i katalogami w przestrzeni roboczej należy wykonać następujące kroki:

1. Otworzyć przestrzeń roboczą zgodnie z instrukcją z poprzedniej sekcji lub wybrać zakładkę „Content” (Rys. A.7 – etykieta A) znajdująca się w górnej części strony przestrzeni roboczej. Zostanie

wyświetlona przeglądarka plików i katalogów przestrzeni roboczej.

2. Aby przejść do katalogu należy kliknąć na jego wiersz (Rys. A.9 – etykieta A). W ten sposób można nawigować po drzewie katalogów przestrzeni roboczej. Ścieżka do obecnego katalogu jest wyświetlana w górnej części przeglądarki (Rys. A.9 – etykieta I), za jej pomocą można wrócić do jednego z poprzednich katalogów.
3. Aby przefiltrować listę plików i katalogów należy wpisać nazwę pliku lub katalogu w pole wyszukiwania (Rys. A.9 – etykieta J).
4. Aby wyświetlić podgląd pliku i opcjonalnie pobrać go należy kliknąć na jego wiersz (Rys. A.9 – etykieta B). Zostanie wyświetlony dialog z podglądem pliku (jeśli wspiera go przeglądarka internetowa) oraz przyciskiem do pobrania pliku.
5. Aby przesyłać pliki do obecnego katalogu należy kliknąć przycisk z ikoną pliku i strzałki w góre (Rys. A.9 – etykieta D). Po kliknięciu przycisku zostanie otwarte okno dialogowe pozwalające na wybranie plików do przesłania poprzez przeciągnięcie bądź wybranie z systemowego eksploratora plików. Przesyłanie plików będzie odbywać się w tle, a postęp można monitorować w panelu dostępnym w nagłówku aplikacji (Rys. A.9 – etykieta K).
6. Aby utworzyć nowy katalog w obecnym katalogu należy kliknąć przycisk z ikoną katalogu i plusa (Rys. A.9 – etykieta E). Po kliknięciu przycisku zostanie otwarte okno dialogowe pozwalające na wpisanie nazwy nowego katalogu, wybranie kategorii oraz ustawienie praw dostępu do katalogu przez członków (Typy praw dostępu są opisane wewnętrz dialogu).
7. Aby edytować plik lub katalog należy otworzyć jego menu kontekstowe poprzez kliknięcie przycisku z ikoną trzech kropek (Rys. A.9 – etykieta H). Zostanie otwarte menu kontekstowe, w którym należy wybrać opcję „Settings”. Zostanie otwarty dialog pozwalający na zmianę nazwy pliku lub zmianę opcji katalogu analogicznie do tworzenia nowego katalogu.
8. Aby usunąć plik lub katalog należy otworzyć menu kontekstowe (Rys. A.9 – etykieta H) i wybrać opcję „Delete”. Po potwierdzeniu chęci usunięcia plik lub katalog zostanie usunięty z systemu.
9. Na plikach i katalogach można wykonywać operacje zbiorcze takie jak usuwanie, przenoszenie i kopiowania. W tym celu należy włączyć tryb zaznaczania poprzez kliknięcie przycisku z symbolem pola wyboru (Rys. A.9 – etykieta F). Następnie zaznaczyć pliki i katalogi, na których chcemy wykonać operacje. Po zaznaczeniu plików i katalogów w menu kontekstowym dowolnego pliku lub katalogu (Rys. A.9 – etykieta H) oraz przeglądarki plików (Rys. A.9 – etykieta G) pojawią się opcje odpowiednich operacji zbiorczych. Brak możliwości wybrania opcji oznacza brak uprawnień do wykonania danej operacji na zaznaczonych plikach i katalogach wynikający z ustawień katalogów.
10. Aby odświeżyć listę plików i katalogów należy kliknąć przycisk z ikoną strzałki okrężnej (Rys. A.9 – etykieta C).

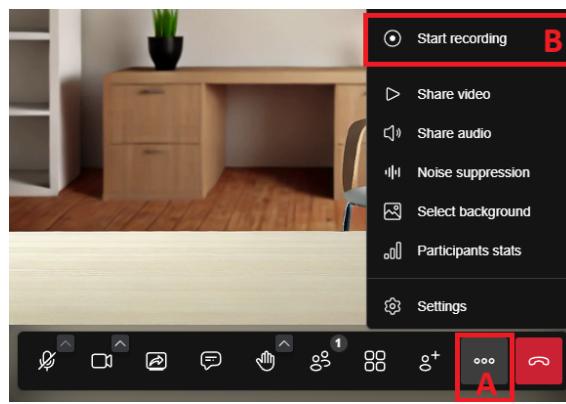
Sesje spotkaniowe

The screenshot shows a 'Sessions' page within a workspace. The top navigation bar includes 'Workspaces > Workspace 1', 'CONTENT', 'SESSIONS' (which is underlined), 'FORUM', and 'SETTINGS'. On the right side, there are five small red-bordered boxes labeled C, D, E, G, and H. Below the main list, there is a single entry 'test' with the date '22.08.2024 18:32:19'.

Rysunek A.10: Lista sesji spotkaniowych

The screenshot shows a 'Meeting' session titled 'Jam Session 1'. The interface includes a video feed of a room with a desk and chair, labeled 'A'. To the right is a control panel with three tracks: Bass, Guitar, and Drums. Each track has volume sliders and MUTE/SOLO buttons. The control panel is labeled 'B'. Above the video feed, there is a timeline from 0:00 to 0:35. The top navigation bar includes 'MEETING', 'RECORDINGS', 'WORKSPACE CONTENT', 'WORKSPACE FORUM', and 'SESSION SETTINGS'. To the right of the control panel, there are five small red-bordered boxes labeled C, D, E, F, G, and H.

Rysunek A.11: Sesja spotkaniowa



Rysunek A.12: Wideokonferencja

Tworzenie sesji spotkaniowej

Aby utworzyć nową sesję spotkaniową należy wykonać następujące kroki:

1. Otworzyć przestrzeń roboczą zgodnie z instrukcją z poprzedniej sekcji.

2. Wybrać zakładkę „Sessions” (Rys. A.7 – etykieta B) znajdująca się w górnej części strony przestrzeni roboczej. Zostanie wyświetlona lista sesji spotkaniowych przestrzeni roboczej.
3. Kliknąć przycisk z ikoną plusa (Rys. A.10 – etykieta D) znajdujący się w prawym górnym rogu listy sesji spotkaniowych. Po kliknięciu przycisku zostanie otwarty formularz tworzenia sesji spotkanowej pozwalający na wprowadzenie nazwy sesji, opisu oraz wybranie daty i godziny rozpoczęcia sesji.
4. Po wprowadzeniu danych kliknąć przycisk „Create” w celu utworzenia sesji spotkanowej.

Otwarcie sesji spotkaniowej

Aby otworzyć sesję spotkaniową należy wykonać następujące kroki:

1. Otworzyć przestrzeń roboczą zgodnie z instrukcją z poprzedniej sekcji.
2. Wybrać zakładkę „Sessions” (Rys. A.7 – etykieta B) znajdująca się w górnej części strony przestrzeni roboczej. Zostanie wyświetlona lista sesji spotkaniowych przestrzeni roboczej.
3. Kliknąć na wiersz sesji spotkanowej, którą chcemy otworzyć (Rys. A.10 – etykieta A). Po kliknięciu wiersza zostanie otwarta sesja spotkaniowa.
4. Domyślna zakładka „Meeting” pozwala na korzystanie z wideokonferencji i zintegrowanego z nią edytora audio. W zakładce „Recordings” można przeglądać katalog z nagraniami z sesji spotkanowych automatycznie tam archiwizowanych. W zakładce „Workspace content” można przeglądać pliki i katalogi przestrzeni roboczej bez konieczności opuszczania sesji spotkanowej.
5. W zakładce „Workspace forum” można korzystać z forum dyskusyjnego przestrzeni roboczej. W zakładce „Session settings” można edytować ustawienia sesji spotkanowej takie jak nazwa oraz data i godzina rozpoczęcia, a także usunąć sesję.

Korzystanie z wideokonferencji i edytora audio

Poniżej przedstawiono poszczególne działania możliwe do wykonania w zakładce „Meeting” sesji spotkaniowej:

1. Aby dołączyć do wideokonferencji należy kliknąć przycisk „Join meeting” w oknie wideokonferencji (Rys. A.11 – etykieta A) i zezwolić na dostęp do mikrofonu i kamery w przeglądarce.
2. Aby rozpocząć nagrywanie sesji należy otworzyć menu kontekstowe w oknie wideokonferencji (Rys. A.12 – etykieta A) i wybrać opcję „Start recording” (Rys. A.12 – etykieta B). Po zakończeniu nagrywania należy ponownie otworzyć menu kontekstowe i wybrać opcję „Stop recording”. Nagrywanie zatrzyma się również automatycznie po opuszczeniu wideokonferencji przez wszystkich uczestników. Nagranie zostanie automatycznie zapisane w katalogu o nazwie sesji spotkanowej umieszczonym w katalogu „Sessions” przestrzeni roboczej.
3. Aby dodać nową ścieżkę w edytorze audio należy kliknąć przycisk „Add track” w oknie edytora audio (Rys. A.11 – etykieta R).
4. Aby rozpocząć edycję ścieżki należy kliknąć przycisk z ikoną ołówka (Rys. A.11 – etykieta K) znajdujący się obok ścieżki. Po kliknięciu przycisku zostanie ścieżka zostanie zablokowana dla innych uczestników i pojawią się opcje edycji ścieżki.
5. Aby zakończyć edycję ścieżki należy kliknąć przycisk z ikoną przekreślonego ołówka (Rys. A.11 – etykieta M). Po kliknięciu przycisku ścieżka zostanie odblokowana dla innych uczestników.
6. Aby nagrać ścieżkę za pomocą urządzenia audio należy kliknąć przycisk z ikoną mikrofonu (Rys. A.11 – etykieta N) znajdujący się obok ścieżki. Po kliknięciu przycisku rozpocznie się

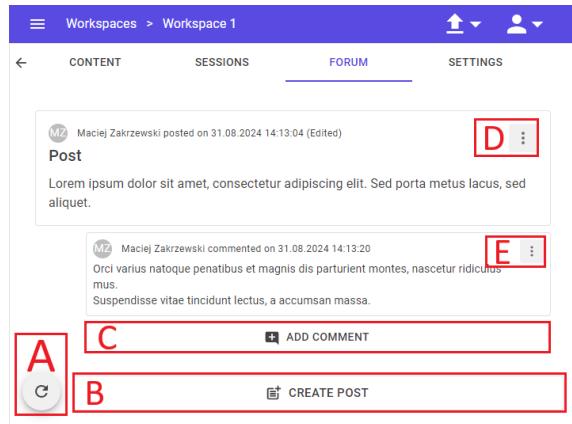
nagrywanie ścieżki. W trakcie nagrywania odtwarzane będą pozostałe ścieżki. Po zakończeniu nagrywania należy ponownie kliknąć przycisk, aby zakończyć nagrywanie. Nagrany dźwięk zostanie automatycznie przesłany na serwer i udostępniony innym uczestnikom. Aby przesyłać dane dźwiękowe z pliku należy kliknąć przycisk z ikoną pliku (Rys. A.11 – etykieta O) znajdujący się obok ścieżki. Po kliknięciu przycisku zostanie otwarte okno dialogowe pozwalające na wybranie pliku dźwiękowego z systemowego eksploratora plików. Po wybraniu pliku dźwiękowego zostanie on przesłany na serwer i udostępniony innym uczestnikom.

7. Aby usunąć ścieżkę należy kliknąć przycisk z ikoną krzyżyka (Rys. A.11 – etykieta P) znajdujący się obok ścieżki, która jest aktualnie edytowana.
8. Aby zmienić nazwę ścieżki należy kliknąć na jej nazwę (Rys. A.11 – etykieta Q) i wpisać nową.
9. Głośność edytowanych ścieżek można regulować za pomocą suwaka (Rys. A.11 – etykieta L).
10. W dowolnym momencie poza nagrywaniem można odtwarzać projekt lokalnie korzystając ze standardowego zestawu przycisków do odtwarzania dźwięku (Rys. A.11 – etykieta B).
11. Aby wyciszyć lokalnie dowolną ścieżkę lub wszystkie poza nią należy kliknąć odpowiednio przycisk „Mute” (Rys. A.11 – etykieta I) lub „Solo” (Rys. A.11 – etykieta J).
12. Aby zmienić poziom przybliżenia wizualizacji ścieżek należy użyć przycisków z ikoną lupy (Rys. A.11 – etykieta C).
13. Aby przejść do trybu przesuwania ścieżek względem siebie należy kliknąć przycisk z ikoną strzałki (Rys. A.11 – etykieta E). W trybie przesuwania można przesuwać edytowane ścieżki na osi czasu w celu synchronizacji z innymi ścieżkami.
14. Aby przejść do trybu zaznaczania należy kliknąć przycisk z ikoną kurSORA (Rys. A.11 – etykieta D). W trybie zaznaczania można zaznaczać na osi czasu fragmenty ścieżek do otworzenia lub zmieniać moment, od którego odtwarzane są ścieżki.
15. Aby odsłuchać wynikowy projekt wspólnie z innymi uczestnikami należy kliknąć przycisk z ikoną udostępniania (Rys. A.11 – etykieta F). Po kliknięciu przycisku w oknie wideokonferencji zostanie otwarte okno z odtwarzaczem audio, którego sterowanie jest zsynchronizowane z innymi uczestnikami. Aby zakończyć odsłuchanie należy kliknąć przycisk z przekreślona ikoną udostępniania (Rys. A.11 – etykieta G). Odtwarzacz i dźwięk będzie również dostępny na nagraniu sesji.
16. Aby zapisać wynikowy plik dźwiękowy w przestrzeni roboczej należy kliknąć przycisk z strzałki w chmurze (Rys. A.11 – etykieta H). Po kliknięciu przycisku zostanie otwarte okno dialogowe pozwalające na wpisanie nazwy pliku dźwiękowego oraz wybranie katalogu, w którym ma zostać zapisany plik.

Forum dyskusyjne

Poniżej przedstawiono poszczególne działania możliwe do wykonania w forum dyskusyjnym przestrzeni roboczej:

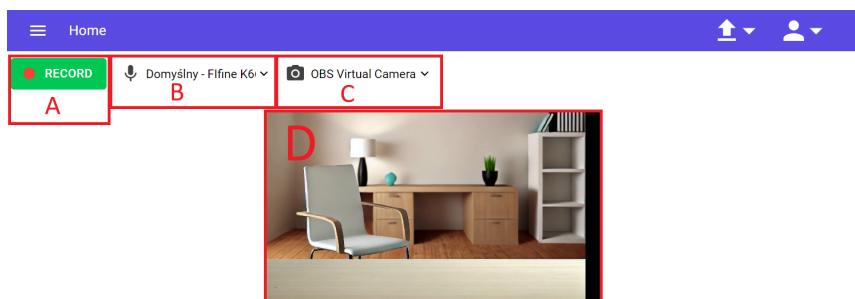
1. Aby otworzyć forum dyskusyjne należy otworzyć przestrzeń roboczą zgodnie z instrukcją z poprzedniej sekcji i wybrać zakładkę „Forum” (Rys. A.7 – etykieta C) lub zakładkę „Workspace forum” w sesji spotkanowej.
2. Aby odświeżyć stan forum należy kliknąć przycisk z ikoną strzałki okrężnej (Rys. A.13 – etykieta A).



Rysunek A.13: Forum dyskusyjne

3. Aby utworzyć nowy post należy kliknąć przycisk „Create post” (Rys. A.13 – etykieta B). Po kliknięciu w miejscu przycisku pojawi się formularz tworzenia postu, w którym należy wpisać tytuł postu oraz treść oraz opcjonalnie załączyć pliki z przestrzeni roboczej.
4. Aby dodać komentarz do postu należy kliknąć przycisk „Add comment” (Rys. A.13 – etykieta C) znajdujący się pod postem. Po kliknięciu w miejscu przycisku pojawi się formularz dodawania komentarza.
5. Posty i komentarze można edytować i usuwać za pomocą opcji dostępnym w menu kontekstowym postu lub komentarza. Menu kontekstowe można otworzyć klikając przycisk z ikoną trzech kropek (Rys. A.13 – etykiety D i E).

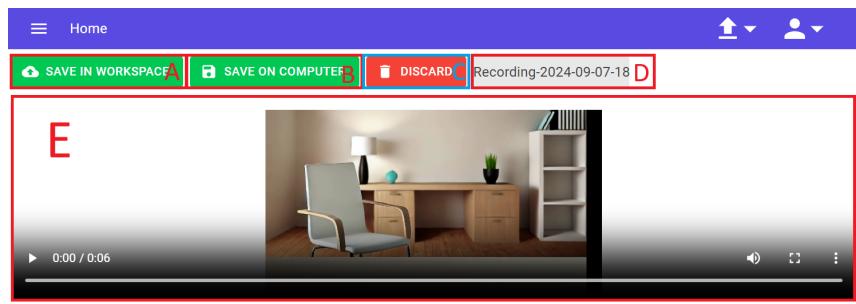
Tworzenie materiałów wideo



Rysunek A.14: Kreator tworzenia materiałów wideo przed nagraniem

Aby stworzyć materiał wideo należy wykonać następujące kroki:

1. Jeśli menu boczne nie jest widoczne, rozwinąć je klikając na ikonę menu po lewej stronie nagłówka strony (Rys. A.3 – etykieta E).
2. Wybrać w menu bocznym opcję „Video recorder” (Rys. A.3 – etykieta H). Po kliknięciu opcji zostanie otwarty kreator tworzenia materiałów wideo.
3. W kreatorze można wybrać urządzenie wejściowe audio i wideo za pomocą list rozwijanych (Rys. A.14 – etykiety B i C).
4. Aby rozpoczęć nagrywanie należy kliknąć przycisk „Record” (Rys. A.14 – etykieta A). Po kliknięciu



Rysunek A.15: Kreator tworzenia materiałów wideo po nagraniu

przycisku rozpoczęcie się nagrywanie materiału wideo. Aby zakończyć nagrywanie należy kliknąć przycisk „Stop” znajdujący się teraz w miejscu przycisku „Record”.

5. Po zakończeniu nagrywania można oglądać materiał za pomocą odtwarzacza wideo (Rys. A.15 – etykieta E).
6. W polu tekstowym (Rys. A.15 – etykieta D) można zmienić nazwę materiału wideo.
7. Aby zapisać materiał wideo na urządzeniu należy kliknąć przycisk „Save on computer” (Rys. A.15 – etykieta B). Po kliknięciu plik zostanie zapisany poprzez przeglądarkę internetową na urządzeniu.
8. Aby zapisać materiał wideo w przestrzeni roboczej należy kliknąć przycisk „Save in workspace” (Rys. A.15 – etykieta A). Po kliknięciu zostanie otwarte okno dialogowe pozwalające na wybranie przestrzeni roboczej oraz katalogu, w którym ma zostać zapisany plik.