



Politechnika Łódzka

Instytut Informatyki

## PRACA DYPLOMOWA INŻYNIERSKA

# SYSTEM WSPOMAGAJĄCY ZARZĄDZANIE SZKOŁĄ NAUKI TAŃCA

**Wydział Fizyki Technicznej, Informatyki i Matematyki Stosowanej**

**Promotor: dr inż. Bożena Borowska**

**Dyplomant: Maria Wichrowska**

**Nr albumu: 230035**

**Kierunek: Informatyka stosowana**

**Specjalność: Inżynieria baz danych**

Łódź, 2023



Instytut Informatyki

90-924 Łódź, ul. Wólczańska 215, budynek B9

tel. 042 631 27 97, 042 632 97 57, fax 042 630 34 14 email: office@ics.p.lodz.pl

# Spis treści

1. Wstęp.....	4
1.1. Cel i zakres pracy .....	5
1.2. Układ pracy .....	6
2. Analiza istniejących rozwiązań.....	7
2.1. Bookero .....	7
2.2. eFitness.....	9
2.3. WodGuru.....	11
2.4. Swiplo.pl .....	12
2.5. Własne rozwiązania .....	14
3. Zastosowane narzędzia i technologie.....	16
3.1. Środowisko programistyczne .....	17
3.2. Stos technologiczny .....	17
3.2.1. MongoDB .....	18
3.2.2. Node.js i Express.js.....	18
3.2.3. React .....	19
3.2.4. Inne biblioteki .....	20
4. Projekt aplikacji .....	21
4.1. Aktorzy systemu.....	22
4.2. Wymagania funkcjonalne.....	23
4.3. Wymagania нефункционалне.....	24
4.4. Architektura systemu .....	26
4.5. Diagram komponentów .....	27
4.6. Diagramy czynności.....	28
4.7. Projekt bazy danych .....	29
5. Implementacja systemu .....	31
5.1. Komunikacja z bazą danych.....	31
5.2. Szyfrowanie haseł .....	33
5.3. Moduł wysyłania wiadomości email.....	35
5.4. Routing.....	36
5.5. Hooki React.....	37
5.5.1. useState .....	38
5.5.2. useEffect .....	38
5.5.3. useRef .....	39
6. Dokumentacja użytkownika.....	39
6.1. Klient.....	40

6.1.1.	Zakładanie konta .....	40
6.1.2.	Logowanie .....	42
6.1.3.	Rezerwacja zajęć .....	43
6.1.4.	Przeglądanie rezerwacji .....	44
6.1.5.	Przeglądanie dostępnych grup .....	44
6.1.6.	Wylogowanie .....	45
6.2.	Pracownik .....	45
6.2.1.	Przeglądanie i tworzenie rezerwacji .....	46
6.2.2.	Wysyłanie wiadomości email .....	47
6.2.3.	Tworzenie raportów i list .....	48
6.2.4.	Lista obecności .....	50
6.3.	Właściciel szkoły tańca .....	51
6.3.1.	Dodawanie grup .....	51
6.3.2.	Modyfikowanie i usuwanie grupy .....	52
7.	Podsumowanie .....	53
7.1.	Możliwość rozwoju aplikacji .....	55
8.	Bibliografia .....	56
9.	Spis rysunków .....	57
10.	Spis tabel .....	58
11.	Spis listingów .....	59

# 1. Wstęp

Ćwiczenia fizyczne są codziennym nawykiem wielu osób w Polsce i na świecie. Kultura zdrowego trybu życia i związanej z tym aktywności fizycznej z roku na rok robi się coraz bardziej popularna. Osoby zdrowe chcą utrzymać sprawność, a osoby chore i otyłe starają się dzięki ćwiczeniom wrócić do zdrowia. Jednakże nie wszyscy lubią popularne ćwiczenia na siłowni lub bieganie na zewnątrz. Niektórzy poszukują aktywności, która nie tylko pozwoli utrzymać formę lub zrzucić zbędne kilogramy, ale również będzie sprawiać przyjemność. Jednym z takich wyborów jest taniec, który może być również formą terapii w sensie fizycznym, emocjonalnym oraz socjalnym [1].

Taniec jest znany od początku historii ludzkości [2]. Nie jest to jedynie sport, ale także forma ruchu. Jest powszechny nie tylko u ludzi, ale też u zwierząt, które w ten sposób wyrażają swoje emocje (jako taniec godowy lub rozwinięte popisy). W ostatnich dwudziestu latach taniec towarzyski i solo rozpowszechnił się jeszcze bardziej, głównie dzięki tanecznym programom telewizyjnym. Programy te pokazały, że taniec to nie tylko elitarne zawody sportowe dla wybranych osób, które trenują od najmłodszych lat [3]. To także pasja wielu ludzi, którzy mają ogromną radość z tańca. Od tego czasu szkoły z nauką tańca zaczęły pojawiać się masowo na mapie Polski.

Na podstawie wyników z przeprowadzonego badania Głównego Urzędu Statystycznego GUS z 2021 roku [4] stwierdzono, że w zajęciach sportowych lub rekreacji ruchowej uczestniczyło 38,8% osób, z czego 6,2% brało udział w zajęciach tanecznych. Oznacza to, że z takiej formy ruchu korzysta w Polsce ponad 920 tysięcy osób.

Branża szkół tańca rozwija się z roku na rok, a w związku z tym ich właściciele szukają jak najlepszego sposobu na zarządzanie nie tylko grafiką zajęć i zapisami na zajęcia taneczne, ale też swoimi pracownikami. Niestety obecny rynek aplikacji dostępnych dla tego typu przedsiębiorców jest bardzo ubogi i nie spełnia wymagań społeczeństwa XXI wieku. Aplikacje dla szkół tańca, które powstały do tej pory, są albo drogie, albo nie są przystosowane do zajęć tanecznych.

W dzisiejszych czasach aplikacje webowe są nieodzowną częścią wszystkich przedsiębiorstw. Służą nie tylko do marketingu internetowego oraz komunikacji z potencjalnym i obecnym klientem, ale również do zarządzania firmą i pracownikami. Obecnie przedsiębiorstwo bez strony internetowej jest mało wiarygodne. Taki biznes

nie ma możliwości dotrzeć do dużej liczby potencjalnych klientów. Przekaz internetowy jest głównym środkiem przekazywania informacji. Aby rozwinąć swoje przedsiębiorstwo, w tym szkołę tańca, właściciel musi skupić się także na tej części biznesu. Poleganie na przestarzałych metodach zapisu na zajęcia (poprzez kontakt telefoniczny lub wysyłanie maila) jest związane z podatnością danych na błędy. Mogą one wpłynąć na brak przekazu informacji, na przykład o odwołanych zajęciach, przez co klienci tracą czas na dojazd na miejsce zajęć. Przez takie niedociągnięcia przedsiębiorstwo może zrazić do siebie klientów. Dlatego szkoły tańca powinny zakładać systemy i bazy danych, które pomogłyby w zautomatyzowaniu procesu zapisów. Co więcej, taki system będzie internetową wizytówką firmy oraz świadectwem ich nowoczesności.

## **1.1. Cel i zakres pracy**

Celem pracy jest zaprojektowanie i implementacja aplikacji webowej wspomagającej zarządzanie szkołą nauki tańca. Dla klientów aplikacja będzie narzędziem do rezerwacji miejsc na zajęcia. Pracownikom i właścicielom posłuży do zarządzania rezerwacjami, zajęciami grupowymi, klientami oraz do tworzenia raportów na podstawie dostępnych w niej danych. Zostanie napisana w architekturze client – server w języku programowania Javascript z wykorzystaniem biblioteki React.

Zakres pracy obejmuje:

- Przegląd istniejących na rynku rozwiązań
- Opis narzędzi i technologii wykorzystanych do tworzenia aplikacji
- Opis środowiska programistycznego
- Omówienie architektury systemu
- Projekt aplikacji z podziałem na wymagania funkcjonalne i нефункционалне
- Stworzenie aplikacji z osobnymi widokami dla klienta, pracownika oraz właściciela szkoły tańca.
- Opracowanie dokumentacji użytkownika

## **1.2. Układ pracy**

Praca składa się z 11 rozdziałów. Rozdział pierwszy zawiera wstęp i wprowadzenie w tematykę oraz cel i zakres pracy dyplomowej. W rozdziale drugim znajduje się przegląd oraz analiza dostępnych na rynku rozwiązań. Wszystkie użyte narzędzia i technologie wymienione są w rozdziale trzecim. Problem projektowania aplikacji webowej omówiono w rozdziale czwartym. W rozdziale piątym zawarto implementację systemu według stworzonego projektu. Dokumentacja użytkownika została opisana w rozdziale szóstym. W rozdziale siódmym znajduje się podsumowanie pracy. Bibliografia, spis rysunków, spis tabel i spis listingów znajdują się kolejno w rozdziałach ósmym, dziewiątym, dziesiątym i jedenastym.

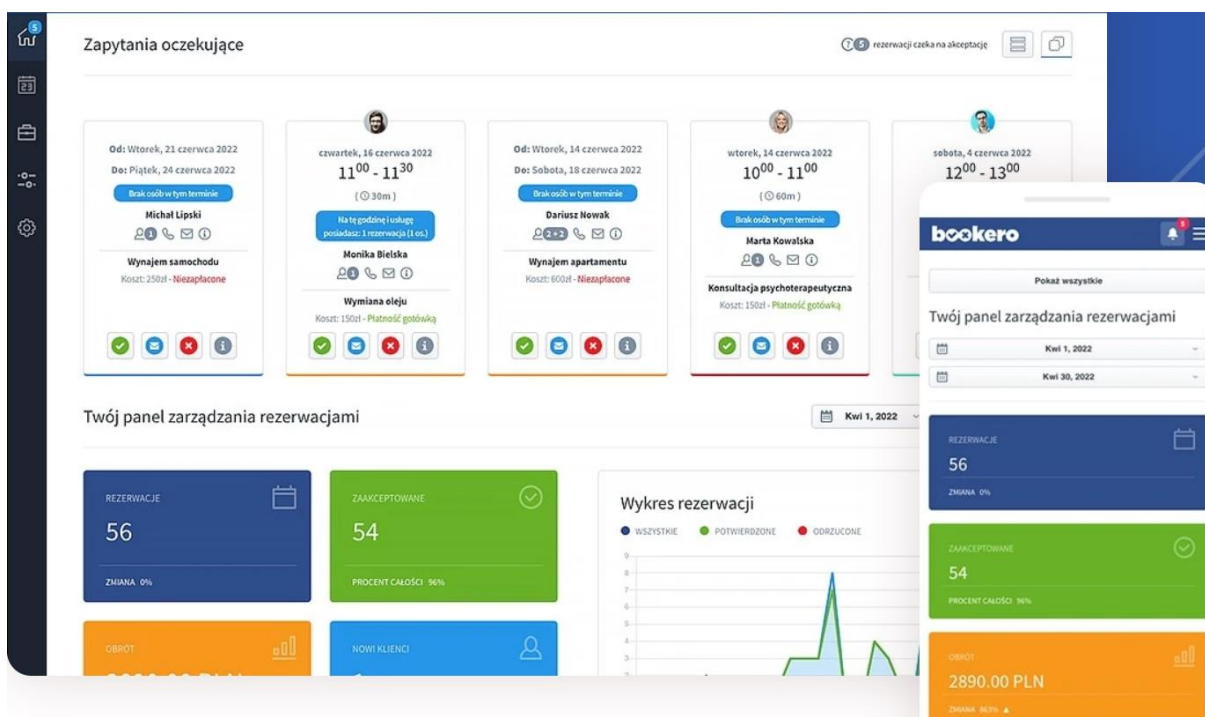
## **2. Analiza istniejących rozwiązań**

W Polsce znajduje się kilka tysięcy szkół tańca, a cała branża wyceniana jest nawet na 1 mld złotych [5]. Na potrzeby tej pracy dyplomowej przeprowadzono wywiad z właścicielami szkół tańca w największych polskich miastach, z którego jasno wynika, że na rynku widoczny jest brak odpowiedniego oprogramowania dla właścicieli tego typu działalności. Tylko dwie z czterech przytoczonych w tej pracy dyplomowej aplikacji są używane przez szkoły tańca, których przedstawiciele byli uczestnikami wywiadu. Przeprowadzone badanie wykazało, że niewielka liczba szkół tańca korzysta z aplikacji przeznaczonych do użytku przez siłownie i kluby fitness. Mimo wszystko, właściciele szkół decydują się również na takie rozwiązanie ze względu na brak aplikacji dostosowanych do ich potrzeb. Przyczyną mogą być również dodatkowe miesięczne koszty za korzystanie. Na polskim rynku istnieje kilkanaście aplikacji webowych do zarządzania szkołami tańca lub innymi ośrodkami sportowymi. Z badania przeprowadzonego na potrzeby tej pracy dyplomowej, ustalono, że istniejące narzędzia nie są najlepszej jakości.

### **2.1. Bookero**

Bookero.pl to start-up założony przez Pawła Nowakowskiego, obecnie prezesa podwarszawskiej spółki Bookero. Jest to system rezerwacji online. Widok głównego panelu aplikacji bookero.pl został przedstawiony na Rysunku 1. Strona daje klientom szkół tańca możliwość rezerwacji zajęć przez Internet. Według twórców aplikacji [6] pozwala „zarabiać więcej, sprawniej obsługiwać swoje rezerwacje i oszczędzać czas”. Aplikacja ma wiele funkcji, takich jak rezerwacje na dni oraz godziny, dodawanie wielu rezerwacji do koszyka, określanie warunków i limitów rezerwacji, tworzenie cenników i ustalanie reguł cenowych, ustalanie harmonogramu godzin otwarcia, zarządzanie parametrami rezerwacji, informacja o pełnym zakresie oferowanych usług, ustanawianie rezerwacji cyklicznych, tworzenie listy kodów rabatowych, umożliwienie płatności online, budowa własnej strony www w oferowanym systemie, wybór widoku formularza rezerwacji oraz zarządzanie kontami pracowników. Jest to zdecydowanie ciekawa propozycja również dla szkół tańca, jednak decydują się na nią nieliczni ze względu na małe przystosowanie dla tego typu działalności. Głównymi klientami aplikacji Bookero są siłownie, przez co niektóre funkcje są dostosowane dla tych

obiektów. Szkoły tańca korzystające z tej platformy mają zastrzeżenia co do kilku funkcjonalności. Nie jest możliwe oznaczenie daty startu kursu, co w przypadku zajęć tanecznych jest bardzo ważne, aby móc dobrze wypromować swoje wydarzenia z wyprzedzeniem. Klient chcący rozpocząć nowy kurs tańca, który rozpoczyna się za dwa tygodnie, a który został dodany już do aplikacji, widzi również wcześniejsze daty, mimo braku tych zajęć przed danym terminem. Natomiast w panelu tworzenia brakuje funkcji do sortowania alfabetycznego, przez co nowe kursy trafiają na koniec listy i muszą być ręcznie przenoszone na początek. Uczestnik wywiadu ze szkoły tańca w jednym z polskich miast przyznaje, że w jego opinii to również nie jest do końca przemyślane – skrolowanie w Bookero nie odbywa się w sposób ciągły i jeżeli lista kursów jest długa, należy przeciągać dany kafelek krok po kroku w górę strony internetowej. Według właścicieli szkół tańca cena tego systemu nie jest do końca adekwatna do jakości tego rozwiązania.



Rysunek 1. Panel główny aplikacji webowej Bookero [6]

Aplikacja Bookero jest płatna, w związku z tym właściciel szkoły tańca musi się liczyć ze stałym comiesięcznym wydatkiem za korzystanie z niej. Mimo tego, niektórzy decydują się na to rozwiązanie ze względu na brak lepszych alternatyw. Wady i zalety systemu Bookero zostały podsumowane w Tabeli 1.

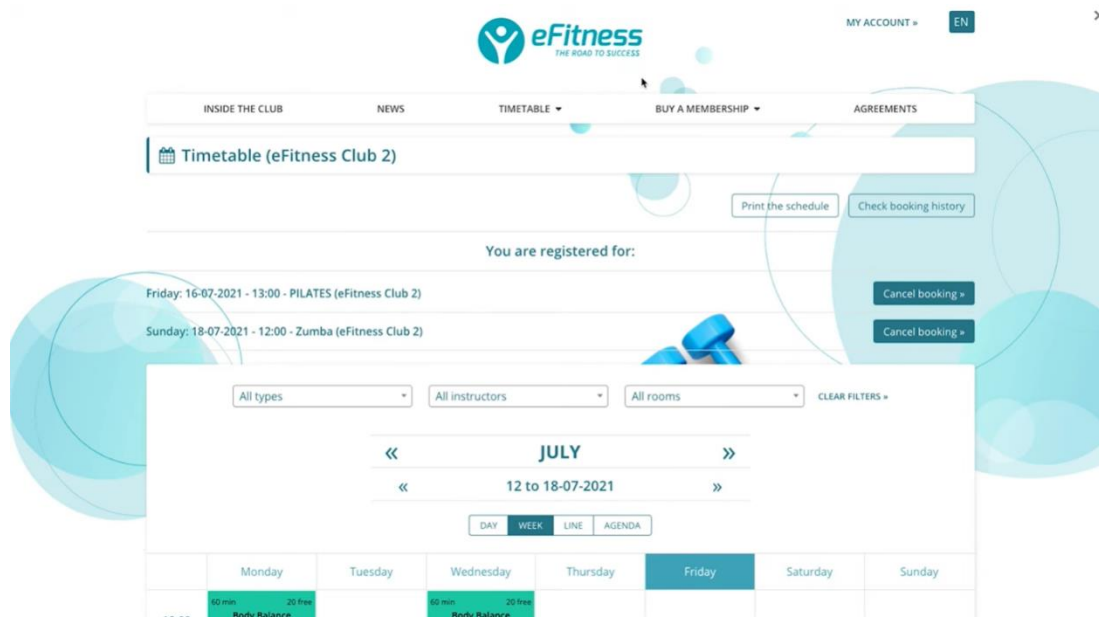


*Tabela 1. Zalety i wady aplikacji webowej Bookero*

Zalety	Wady
Otwarty system rezerwacji dla klientów	Comiesięczna płatność (12,90 – 69,90 PLN w zależności od wybranych funkcjonalności)
Płatności online z wieloma systemami płatności	Brak możliwości ustawienia daty początkowej zajęć na dalszą niż 7 dni
Katalog klientów z ich danymi	Kursy nie są sortowane, więc właściciel szkoły tańca musi manualnie ustawić najnowszy kurs na szczycie listy
Możliwość wykupienia opcji powiadomień SMS dla klientów o nadchodzących kursach lub rezerwacjach	Podczas ustawiania kursu na górze listy nie ma możliwości skrolowania, więc kafelek z kursem musi być przestawiany krok po kroku
Ankiety dostępne dla klientów	Wiele funkcji jest dodatkowo płatne

## 2.2. eFitness

eFitness.pl to aplikacja webowa stworzona przez firmę eFitness S.A. i przeznaczona do obsługi klubów fitness. Widok głównego panelu aplikacji eFitness został przedstawiony na Rysunku 2. Jest to narzędzie do zarządzania klubem. Posiada strefę klienta online i aplikację mobilną. Twórcy chwalą się na stronie internetowej swoim intuicyjnym interfejsem do zarządzania płatnościami, dostępem do danych o klubowiczach, rezerwacją zajęć i treningów personalnych, sprzedażą produktów i wieloma innymi funkcjonalnościami [7]. Po sprawdzeniu bazy klientów aplikacji eFitness okazało się, że niestety z tej aplikacji nie korzysta aktualnie żadna szkoła tańca w Polsce, a historycznie tylko jedna szkoła zdecydowała się w poprzednich latach na to rozwiązanie. Jest to związane z faktem, iż w tym przypadku również nie ma możliwości dodawania kursów tańca od konkretnej daty, na przykład z dwutygodniowym wyprzedzeniem. W takiej sytuacji klienci nie mogą zapisać się na interesujące ich zajęcia wcześniej niż w konkretnym tygodniu. Kursy są wypisane w tygodniowym kalendarzu.—Modyfikacja terminu konkretnego kursu powoduje zmianę wszystkich terminów, od najbliższego tygodnia, bez możliwości dodania daty zakończenia.



Rysunek 2. Panel kalendarza aplikacji webowej eFitness [7]

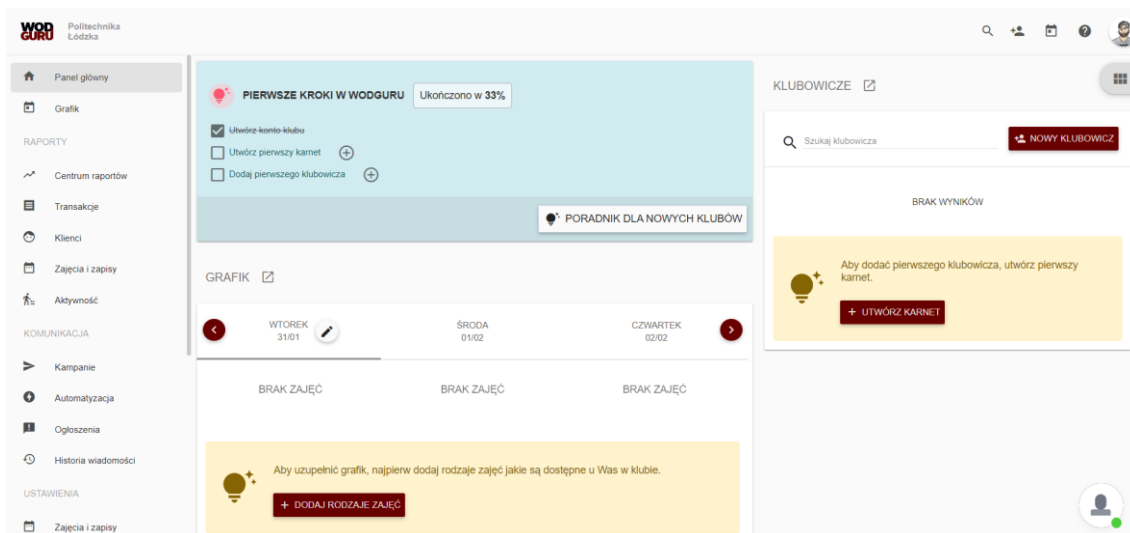
Aplikacja jest z góry płatna, a właściciel szkoły tańca nie ma dostępu nawet do wersji próbnej, aby sprawdzić czy to rozwiązanie będzie odpowiednie dla jego firmy. Dostępne są jedynie filmy pomocnicze w języku angielskim, które pokazują niektóre funkcjonalności systemu eFitness. Wady i zalety systemu eFitness zostały podsumowane w Tabeli 2.

Tabela 2. Zalety i wady aplikacji webowej eFitness

Zalety	Wady
Możliwość tworzenia karnetów dostosowanych do szkoły tańca	Comiesięczna płatność (cena zależna jest od liczby aktywnych użytkowników)
Płatności online	Dodatkowo płatne: niektóre funkcjonalności, karty dla klubowiczów, opcja aplikacji mobilnej, SMSy z powiadomieniami)
Dostępność raportów dotyczących zarobków klubu sportowego.	Brak wersji demonstracyjnej
Możliwość wykupienia aplikacji mobilnej dla klientów	Brak możliwości ustawienia daty początkowej zajęć na dalszą niż 7 dni
Możliwość wykupienia kart dla klientów (służących do odznaczenia wejścia na zajęcia)	Brak możliwości ustawienia daty końcowej kursu

## 2.3. WodGuru

WodGuru jest to system dla klubów fitness, który przyspiesza i automatyzuje ich pracę. Panel główny dostępny po zalogowaniu się przez pracownika szkoły tańca został przedstawiony na Rysunku 3. Aplikacja, jak oznajmia strona internetowa [8], pozwala zautomatyzować płatność za karnety, zapisy na zajęcia oraz marketing i sprzedaż. Oprócz aplikacji webowej, WodGuru posiada również opcję aplikacji mobilnej dla klubowiczów. Jest w niej dostęp do zapisów na zajęcia, opcja powiadomień o kończącym się karnecie, płatności online za karnet oraz historia treningów. Co więcej, pracownicy szkoły tańca mogą kontrolować obecność kursantów na zajęciach i sprawdzać, czy któryś z klubowiczów zalega z płatnościami.



Rysunek 3 Panel główny aplikacji webowej WodGuru [8]

Aplikacja WodGuru jest płatna. Cena to maksymalnie 350 PLN netto za miesiąc korzystania, a ostateczna płatność uzależniona jest od liczby kursantów zarejestrowanych na stronie. Co więcej, klient nie może się samodzielnie zarejestrować. Na początku osoba, która wyrazi potrzebę posiadania profilu użytkownika w aplikacji WodGuru, musi zgłosić się do szkoły tańca. Następnie pracownik musi wprowadzić dane klienta do systemu. Wady i zalety systemu WodGuru zostały podsumowane w Tabeli 3.

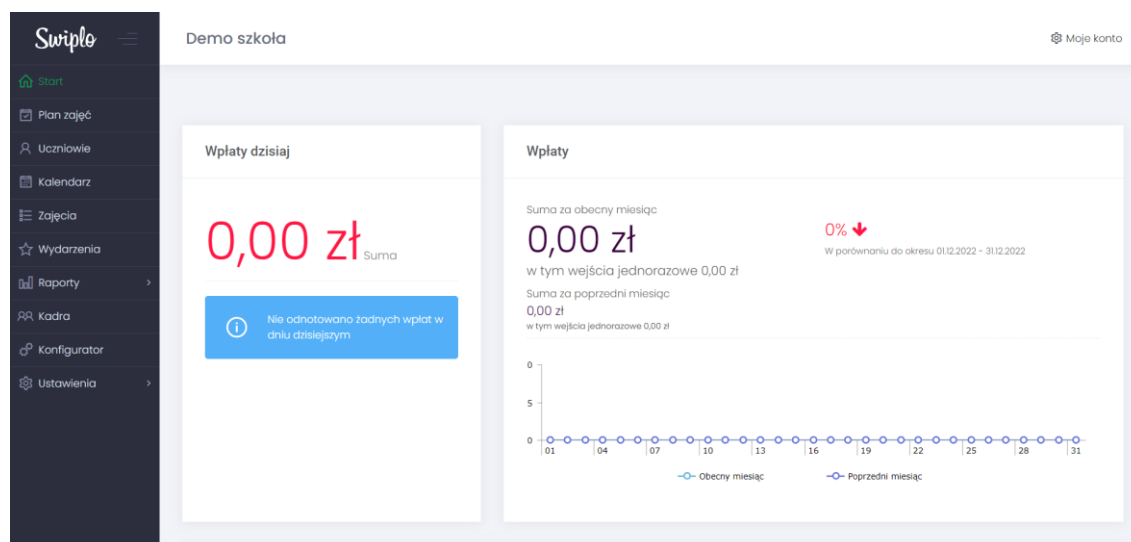
Tabela 3. Zalety i wady aplikacji webowej WodGuru

Zalety	Wady
Prosta obsługa zapisów i płatności online (dla klientów posiadających już konto w WodGuru)	Comiesięczna płatność (3 PLN za każdego aktywnego klubowicza, maksymalnie 350 PLN)
Dostępność aplikacji mobilnej dostosowanej do każdego przedsiębiorstwa indywidualnie	Dodatkowo płatne funkcjonalności: obsługa czytnika kart, obsługa drukarki fiskalnej, recepcja samoobsługowa
Klient może zobaczyć historię swoich treningów	Brak otwartego systemu rejestracji. Przed możliwością zapisu online na zajęcia, klient musi przyjść do szkoły tańca, a pracownik musi wprowadzić jego dane do systemu i aktywować profil
Możliwość zobaczenia raportów dotyczących aktywności i stanu swoich podopiecznych	Pracownik szkoły tańca musi manualnie sprawdzić ważność karnetu klienta
Lista obecności klientów na zajęciach	
Opcja odwołania zajęć jednym kliknięciem, wraz z poinformowaniem zapisanych klientów	

## 2.4. Swiplo.pl

Swiplo.pl to platforma do obsługi szkół tańca. Jej właścicielem jest spółka z ograniczoną odpowiedzialnością. Panel główny wersji demonstracyjnej jest widoczny na Rysunku 4. Dzięki połączeniu grafiku z platformy Swiplo.pl ze stroną internetową szkoły tańca, klienci zyskują możliwość łatwego zapisywania się na zajęcia. Wystarczy, że wypełnią formularz i zaakceptują zgody, a ich dane automatycznie znajdą się w systemie i pojawią na liście danych zajęć. Aplikacja jest zoptymalizowana do wyświetlania na różnych urządzeniach, dzięki czemu łatwo można dodawać nowe zajęcia, ustalać datę początku i końca kursu oraz liczbę osób, która może się zapisać na zajęcia. Swiplo.pl ma również możliwość wyszukiwania poszczególnych zajęć dzięki filtrom. W aplikacji nie zabrakło kalendarza, w którym można dokonywać rezerwacji sal na lekcje indywidualne oraz ustalać ważność karnetów. To prosta i czytelna funkcja, która pozwala na sprawdzanie aktywności karnetów i dat płatności. Wprowadzenie

płatności online umożliwiło wygodne zakupy karnetów widocznych od razu w systemie, a także zapisywanie się na kolejne zajęcia poprzez plan zajęć. eKarnet dla klientów korzystających z pakietu z aplikacją mobilną, pozwala w łatwy i szybki sposób odbić wejście poprzez zeskanowanie kodu QR przez pracownika recepcji. Powiadomienia push pozwalają szybko komunikować się z klientami wybranych zajęć lub ze wszystkimi uczniami szkoły.



Rysunek 4. Panel główny aplikacji webowej Swiplo.pl [9]

Aplikacja nie pozwala na zaznaczanie obecności na zajęciach. Dostępne są jedynie rezerwacje. Co więcej, karnety są przypisywane tylko na konkretne zajęcia, nie na liczbę wejść. Raporty to spis wykupionych karnetów, przeprowadzonych zajęć oraz pracujących instruktorów. Swiplo.pl nie posiada opcji pokazania danych kluczowych do kontaktu z klientami. Istnieje możliwość bezpłatnego korzystania z platformy przez 14 dni, bez umowy i zobowiązań. Wady i zalety systemu Swiplo.pl zostały podsumowane w Tabeli 4.

Tabela 4. Zalety i wady aplikacji webowej Swiplo.pl

Zalety	Wady
Możliwość połączenia grafiku z platformy Swiplo.pl ze stroną internetową szkoły tańca	Comiesięczna płatność (129-299 PLN, w zależności od wybranych funkcjonalności)
Możliwość utworzenia grupy zajęć tańca z	Brak listy obecności

datą początkową i datą końcową	
Płatności online	Brak możliwości zarządzania karnetami. Nie można stworzyć karnetów na przykład na liczbę wejść
Automatyczne powiadomienia w przypadku zakończenia ważności karnetu na zajęcia	Brak możliwości generowania listy numerów lub listy emaili
Dostępna aplikacja mobilna (z wykupioną droższą wersją aplikacji przez szkołę tańca)	Raporty są ograniczone do listy karnetów, listy instruktorów i listy zajęć grupowych

## 2.5. Własne rozwiązania

Jedynie kilka szkół tańca decyduje się na wyżej wymienione aplikacje i to pokazuje niszę rynkową. Większość firm ma swoje rozwiązania do zarządzania szkołą i recepcją. Z wywiadu przeprowadzonego z przedstawicielami szkół tańca wynika, że próby stworzenia własnego systemu informatycznego do zarządzania taką placówką kończą się niepowodzeniem. Ankietowani są świadomi, że systemy te zawierają wiele błędów, głównie ze względu na brak wystarczającej wiedzy w zakresie informatyki i tworzenia stron internetowych. Większość szkół tańca natomiast wciąż korzysta z kartek papieru i długopisu. Recepcjoniści zapisują wszystko w zeszytach lub na specjalnie wydrukowanych listach. Zgłoszenia uczestnictwa w kursie tańca przychodzą na pocztę elektroniczną, a pracownik przepisuje je ręcznie. Z tego powodu popełnianych jest wiele błędów ludzkich, takich jak literówki, czy niewłaściwe numery kontaktowe do klientów, a cały proces trwa o wiele dłużej, niż w przypadku gotowego systemu.



Brak comiesięcznych płatności	Brak bazy danych klientów, zajęć grupowych, karnetów, instruktorów i pracowników
Nie ma potrzeby dodatkowego szkolenia pracowników z systemu informatycznego	Błędy w danych (błędy ludzkie)
Brak narażenia na ataki hackerskie w celu zdobycia danych	Proces rezerwacji miejsca na zajęcia taneczne trwa długo

### 3. Zastosowane narzędzia i technologie

W poniższym rozdziale opisane zostały najważniejsze narzędzia i technologie, które wykorzystano do stworzenia aplikacji webowej dedykowanej szkołom tańca.



### 3.1. Środowisko programistyczne

Jako edytor tekstu został użyty Visual Studio Code. Jest to oprogramowanie typu open-source dostępne na macOS, Windows i Linux. Pozwala używać wbudowanych funkcji, które pomagają w programowaniu i zezwala na proste rozbudowywanie możliwości edytora dzięki wtyczkom opracowanym przez społeczność [10]. Visual Studio Code jest obecnie najpopularniejszym środowiskiem IDE do tworzenia oprogramowania w systemie Windows. Zostało również umieszczone na pierwszym miejscu najlepszych IDE dla użytkowników systemu Windows 10 [11]. Firma Microsoft aktywnie dba o rozwój oraz częste aktualizacje środowiska IDE.

Do kontroli wersji użyto narzędzia Git, które służy do zarządzania różnymi wersjami oprogramowania lub innymi plikami. Obejmuje również śledzenie następujących po sobie wydań tego oprogramowania oraz zarządzanie nimi [12].

Aby stworzyć logo aplikacji, użyto programu Adobe Illustrator. To aplikacja do tworzenia i edycji rysunków, ilustracji i dzieł sztuki w grafice wektorowej. Sprawdza się idealnie do zaprojektowania logo firm lub systemów.

### 3.2. Stos technologiczny

Zgodnie z wynikami ankiety StackOverflow [13], dziesiąty raz z rzędu wybrano JavaScript jako najczęściej używanego języka programowania. Natomiast Node.js i React.js to dwie najczęstsze technologie internetowe używane przez profesjonalnych programistów i osoby uczące się kodowania. Popularność tych technologii odzwierciedla znaczenie i rozpowszechnienie nowoczesnych systemów internetowych [14]. Stąd też wybór stosu technologicznego, który zawiera te technologie.

MERN Stack to połączenie technologii do tworzenia aplikacji webowych. Składa się na niego MongoDB, Express JS, React i Node JS, a nazwa stosu technologicznego powstała z połączenia pierwszych liter nazw tych technologii. Do stworzenia aplikacji użyty był też hipertekstowy język znaczników HTML, który służy do opisywania struktury dokumentu. W połączeniu z HTML użyto również kaskadowe arkusze stylów CSS. To język, który przedstawia wygląd i projekt dokumentu napisanego w HTML. CSS jest używany jako zbiór reguł, w których określone jest, jak dany element strony ma wyglądać [15].

### 3.2.1. MongoDB

MongoDB to dokumentowa nierelacyjna baza danych. Podstawową jednostką danych jest dokument, który ma specjalny unikalny klucz o nazwie „\_id”. Dokumenty są zebrane w kolekcjach, a kolekcje w bazach danych. Dokumenty w MongoDB są uporządkowanymi zbiorami kluczy z przypisanymi do nich wartościami i bardzo często w obrębie jednego dokumentu istnieje wiele par klucz-wartość. Wartości mają zapisane typy (String, Integer, Double, Boolean lub Array), a klucze są ciągami znaków [16].

Kolekcje w MongoDB są grupami dokumentów o dynamicznych schematach. Oznacza to, że kolekcje nie mają wyznaczonej struktury, jak to jest w przypadku relacyjnych baz danych. Oprócz grupowania dokumentów w kolekcje, MongoDB grupuje też kolekcje w ramach baz danych. Jedna instancja MongoDB może przechowywać wiele różnych baz danych, z których każda może być zbiorem różnych kolekcji. Przy korzystaniu z bazy danych pomocny był program MongoDB Compass. To interaktywne narzędzie do wysyłania zapytań, optymalizacji i analizowania danych MongoDB [17]. Pomaga przeprowadzić analizy schematów i stanowi interfejs graficzny dla bazy danych. Co więcej, do reprezentacji obiektów użyto Mongoose. To biblioteka do modelowania danych obiektowych dla MongoDB oraz Node.js. Służy do zarządzania relacjami pomiędzy danymi oraz zarządza reprezentacją obiektów między kodem a reprezentacją tych obiektów w bazie danych MongoDB [18].

### 3.2.2. Node.js i Express.js

Node.js jest środowiskiem uruchomieniowym dla kodu napisanego w języku Javascript. To narzędzie do tworzenia aplikacji po stronie serwera, tworzenia skryptów oraz bibliotek. Node.js umożliwia uruchamianie kodu JavaScript poza przeglądarką, co pozwala na pisanie aplikacji serwerowych oraz narzędzi w oparciu o język JavaScript. Jest asynchroniczny, dzięki czemu obsługuje dane wejściowe i wyjściowe bez blokowania. Co więcej, współpracuje bezpośrednio z systemem operacyjnym, dzięki czemu ma dostęp między innymi do plików, bibliotek systemowych, a nawet uruchomionych procesów [19]. Node.js jest bardzo wydajne i skalowalne, dzięki czemu idealnie nadaje się do tworzenia aplikacji internetowych, serwerów WWW, narzędzi wiersza poleceń, a także do budowy aplikacji czasu rzeczywistego, takich jak czaty lub

gry online. Node.js obsługuje wiele protokołów sieciowych, takich jak HTTP, HTTPS, TCP, a także pozwala na korzystanie z wielu bibliotek i modułów dostępnych w repozytorium npm (Node Package Manager). Node.js jest projektem open-source, co oznacza, że jest dostępne za darmo i każdy może korzystać z niego, rozwijać go i udostępniać swoje rozwiązania. Node.js zyskało popularność w krótkim czasie i jest wykorzystywane przez wielu programistów na całym świecie.

Express.js to popularny, minimalistyczny framework webowy dla Node.js, który ułatwia tworzenie aplikacji internetowych i API. Umożliwia szybkie i proste tworzenie aplikacji internetowych przy użyciu języka JavaScript. Jest łatwy w użyciu, umożliwia definiowanie tras, zarządzanie zapytaniami HTTP, obsługę plików statycznych i middleware [20]. Express.js jest jednym z najpopularniejszych frameworków webowych dla Node.js i jest wykorzystywany przez wiele firm i organizacji na całym świecie. Jego popularność wynika z łatwości użycia i elastyczności, która umożliwia programistom dostosowanie go do ich potrzeb. Express.js oferuje również wiele wbudowanych funkcji, takich jak obsługa zapytań HTTP, obsługa sesji, routing, middleware, renderowanie szablonów HTML, i wiele innych. Dzięki temu programiści mogą skupić się na pisaniu logiki biznesowej aplikacji zamiast poświęcania czasu na tworzenie tych funkcjonalności od podstaw.

### 3.2.3. React

React to biblioteka open-source dla aplikacji internetowych w języku oprogramowania Javascript do budowania interfejsów użytkownika. Pozwala tworzyć dynamiczne interfejsy, które są aktualizowane na stronie w czasie rzeczywistym. Dzięki niemu można też budować skomplikowane interfejsy składające się z wielu elementów, które można wykorzystywać wielokrotnie. React bazuje na deklaratywnym podejściu do programowania, co oznacza, że programista definiuje, co ma się wyświetlać, a nie jak ma to być wyświetlone. Jedną z zalet Reacta jest jego wydajność, ponieważ biblioteka umożliwia renderowanie tylko zmienionych elementów strony zamiast całej strony, co pozwala na szybsze ładowanie i wydajniejszą pracę aplikacji. Jest także przystępny do nauki oraz istnieje wiele dostępnych narzędzi i bibliotek, które ułatwiają pracę z Reactem. Znajduje się na drugim miejscu w rankingu dziesięciu najlepszych framework'ów języka Javascript [21].

### 3.2.4. Inne biblioteki

Jako dodatek do biblioteki React, użyto również biblioteki typu Promise - Axios, która służy do wysyłania żądań http z poziomu przeglądarki lub środowiska Node.js. Co więcej, zapewnia większą kontrolę nad funkcjami zapytań i wprowadza ujednolicony interfejs programowania aplikacji dla zdarzeń [22].

W celu zachowania bezpieczeństwa przy przechowywaniu haseł użytkowników, wykorzystano bibliotekę bcrypt, która wykorzystuje algorytm haszujący [23]. Dzięki tej bibliotece, hasła trzymane w bazie danych są wciąż bezpieczne, mimo potencjalnego dostępu do bazy przez osoby do tego nieupoważnione.

Nodemailer to biblioteka dla Node.js, który umożliwia programistom wysyłanie wiadomości e-mail z poziomu serwera. Jest to popularna biblioteka, która jest wykorzystywana w wielu aplikacjach internetowych do automatycznego wysyłania powiadomień e-mail, takich jak powiadomienia rejestracyjne, potwierdzenia zamówień, przypomnienia o aktywnościach, itp. Nodemailer oferuje wiele opcji konfiguracji, które umożliwiają dostosowanie wiadomości e-mail do indywidualnych potrzeb, w tym takie elementy jak nadawca, odbiorca, temat, treść czy załączniki.

Biblioteka dotenv dla języka JavaScript pozwala na ładowanie zmiennych z pliku .env do środowiska node.js. Pliki .env zawierają wartości zmiennych środowiskowych, takie jak hasła, klucze API i inne informacje konfiguracyjne, które są wykorzystywane przez aplikacje. Biblioteka dotenv jest użyteczna w aplikacjach, w których wiele zmiennych konfiguracyjnych jest wymaganych do uruchomienia aplikacji, ale te wartości nie powinny być publicznie dostępne lub przesyłane w kodzie źródłowym aplikacji. Dzięki wykorzystaniu pliku .env, który jest ignorowany przez system kontroli wersji, można bezpiecznie przechowywać te wartości poza kodem źródłowym aplikacji.

CORS to skrót od Cross-Origin Resource Sharing i odnosi się do mechanizmu bezpieczeństwa stosowanego przez przeglądarki internetowe, który kontroluje, czy strona internetowa może żądać zasobów z innego pochodzenia niż sama strona. Jeśli protokół, nazwa hosta lub numer portu różnią się między stronami, uważane są za pochodzące z różnych źródeł. Mechanizm CORS polega na dodaniu nagłówka HTTP Access-Control-Allow-Origin do odpowiedzi serwera, która określa, jakie strony internetowe mają dostęp do zasobów. W przypadku aplikacji internetowej frontend React, która komunikuje się z backendem napisanym w Node.js, konieczne jest

zezwoleń na dostęp do zasobów poprzez dodanie nagłówka Access-Control-Allow-Origin [24].

## **4. Projekt aplikacji**

Projekt aplikacji zawiera spis aktorów systemu, wymagania funkcjonalne przedstawione za pomocą diagramu przypadków użycia, wymagania niefunkcjonalne, projekt komunikacji technologii dla stosu technologicznego MERN, diagram komponentów, diagramy czynności oraz projekt nierelacyjnej bazy danych.

## 4.1. Aktorzy systemu

Projekt przewiduje stworzenie osobnych widoków dla klienta, pracownika i właściciela szkoły tańca. Widoki te są dostępne po wcześniejszej rejestracji i logowaniu. Aktorzy systemu, którzy będą się komunikować z systemem i korzystać z jego funkcjonalności są podzieleni na cztery role:

- Użytkownik niezalogowany
- Klient
- Pracownik
- Właściciel szkoły tańca

Użytkownik niezalogowany to osoba, która przegląda aplikację webową przed uwierzytelnieniem poprzez system loginu i hasła. W celu korzystania z funkcjonalności może zarejestrować się, a następnie w przypadku posiadania już konta w aplikacji, zalogować się.

Dla klienta dostępne są zakładki:

- ze spisem grup, na które może się zapisać,
- z formularzem rezerwacji zajęć,
- ze spisem swoich dotychczasowych rezerwacji.

Dla pracownika dostępne są zakładki:

- z rezerwacjami, gdzie dostępne są wszystkie ostatnio stworzone rezerwacje przez klientów i pracowników oraz możliwość zrobienia nowej rezerwacji dla klienta,
- ze spisem grup dostępnych w szkole tańca,
- z możliwością wygenerowania listy z numerami telefonów do klientów, listy z emailami do klientów i raportów, w których zawarte są dane liczbowe pozwalające na analizę obecnie prowadzonych zajęć tańca,
- z możliwością zaznaczenia obecności na zajęciach grupowych tańca.

Dla właściciela szkoły tańca dostępne są zakładki dostępne dla pracownika oraz dodatkowo:

- możliwość dodania nowej grupy,
- możliwość edycji grupy, która była wcześniej utworzona,
- możliwość usunięcia grupy.

## 4.2. Wymagania funkcjonalne

W ramach wymagań funkcjonalnych istnieje możliwość:

- zarejestrowania się klienta za pomocą adresu mailowego, hasła oraz danych kontaktowych i identyfikacyjnych,
- zarejestrowania się pracownika i właściciela szkoły tańca za pomocą adresu mailowego, hasła oraz danych identyfikacyjnych,
- zalogowania się użytkownika przy pomocy zarejestrowanego wcześniej adresu email i odpowiadającemu mu hasła,
- przeglądania dostępnych zajęć tanecznych przez użytkowników aplikacji webowej,
- dodawania zajęć tanecznych z dniem tygodnia i godziną zajęć, poziomem i nazwą zajęć oraz liczebnością grupy przez właściciela szkoły tańca,
- usuwania oraz edycji zajęć tanecznych przez właściciela szkoły tańca,
- zarezerwowania miejsca na zajęcia taneczne przez klienta na wybraną grupę tańca,
- zarezerwowania miejsca na zajęcia taneczne przy użyciu danych klienta przez pracownika lub właściciela szkoły tańca,
- zobaczenia rezerwacji dokonanych przez siebie w panelu klienta,
- zobaczenia wszystkich przeszłych rezerwacji klientów przez pracowników oraz właścicieli szkoły tańca,
- wygenerowania raportu liczebności kobiet, mężczyzn oraz osób niebinarnych na zajęciach tanecznych przez właściciela szkoły tańca,
- wygenerowania raportu średniej wieku klientów przez pracowników i właścicieli szkoły tańca,
- wygenerowania raportu z numerami telefonów lub/i adresami email klientów przez pracowników oraz właścicieli szkoły tańca,

- zaznaczenia obecności klienta na zajęciach tanecznych przez pracowników oraz właścicieli szkoły tańca,
- zobaczenia listy obecności klientów na zajęciach tanecznych przez pracowników oraz właścicieli szkoły tańca,
- zobaczenia listy obecności z wybranego dnia przez pracowników oraz właścicieli szkoły tańca,
- zobaczenia listy obecności dla wybranego klienta przez pracowników oraz właścicieli szkoły tańca.

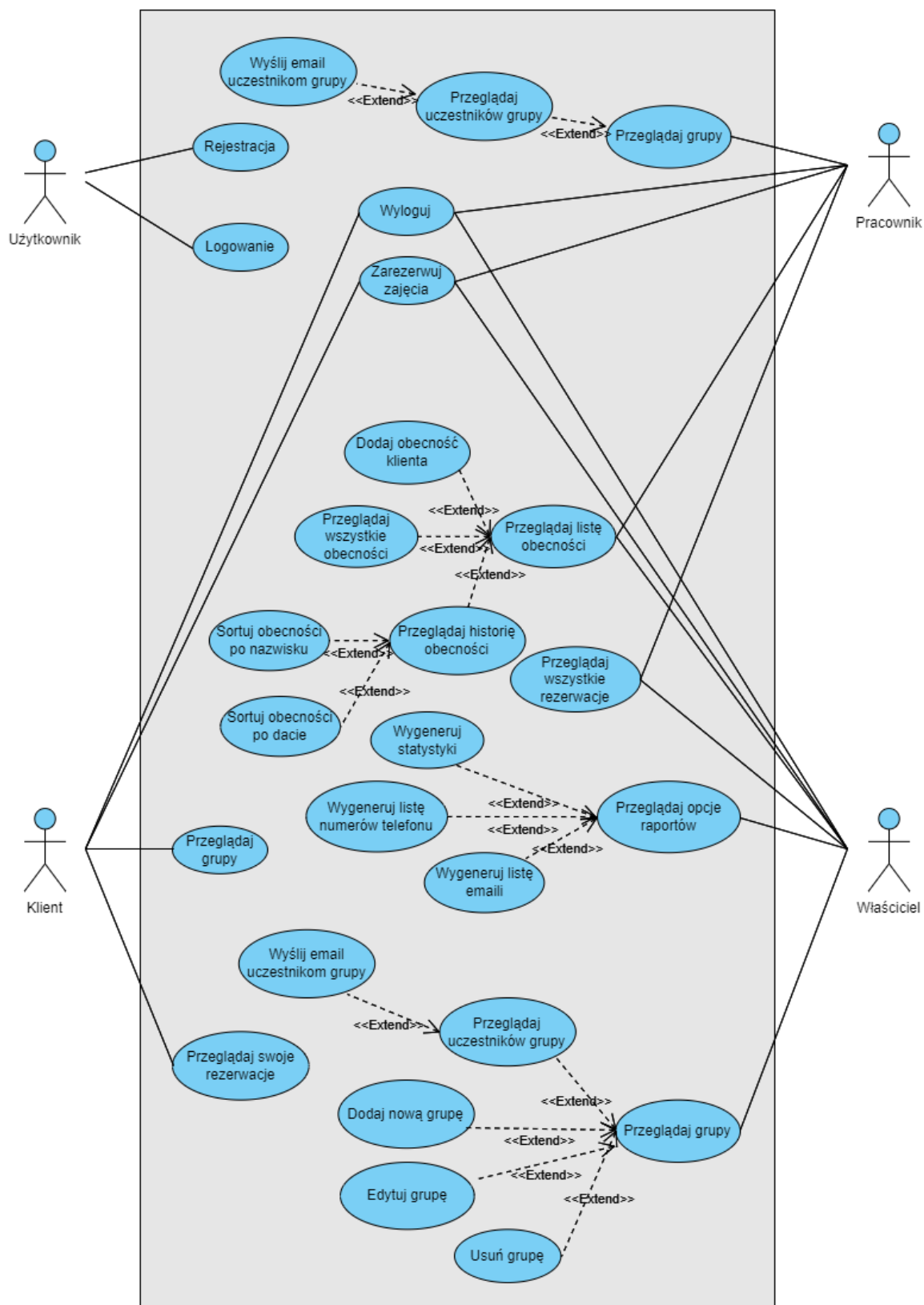
Wymagania funkcjonalne zostały przedstawione na diagramie przypadków użycia (Rysunek 7).

### **4.3. Wymagania нефunkcjonalne**

W ramach wymagań нефunkcjonalnych postanowiono, że:

- aplikacja jest tworzona w języku polskim,
- aplikacja musi mieć dostęp do Internetu,
- główne funkcjonalności aplikacji są dostępne jedynie dla użytkowników zalogowanych,
- nie można zmienić numeru, dnia, ani godziny grupy zajęć tańca,
- nie jest możliwe jednoczesne zarejestrowanie się tym samym adresem email więcej niż raz,
- hasło użytkownika musi mieć od 8 do 24 znaków, posiadać małą i wielką literę, cyfrę oraz znak specjalny,
- hasło musi zostać potwierdzone poprzez wpisanie go po raz drugi,
- hasło jest szyfrowane,
- pola w formularzu rejestracyjnym są walidowane.

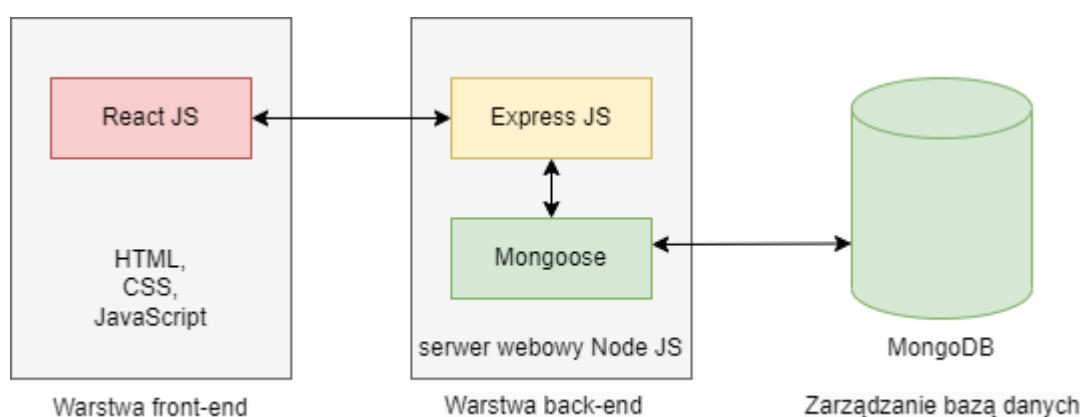




Rysunek 7. Diagram przypadków użycia UML

## 4.4. Architektura systemu

W celu stworzenia aplikacji, została wybrana architektura systemu client-server. Podobnie jak w przypadku typowej aplikacji webowej, stos technologiczny MERN dzieli się na dwa bloki: frontend i backend. Na Rysunku 8 przedstawiono diagram wizualizujący stos technologiczny MERN z frontendem (reprezentującym warstwę klienta) po lewej stronie i backendem (reprezentującym warstwę serwera) po prawej. Obok bloków zilustrowane zostało połączenie z bazą danych, która przechowuje dane używane w aplikacji.



Rysunek 8. Diagram przedstawiający komunikację w stosie technologicznym MERN [25]

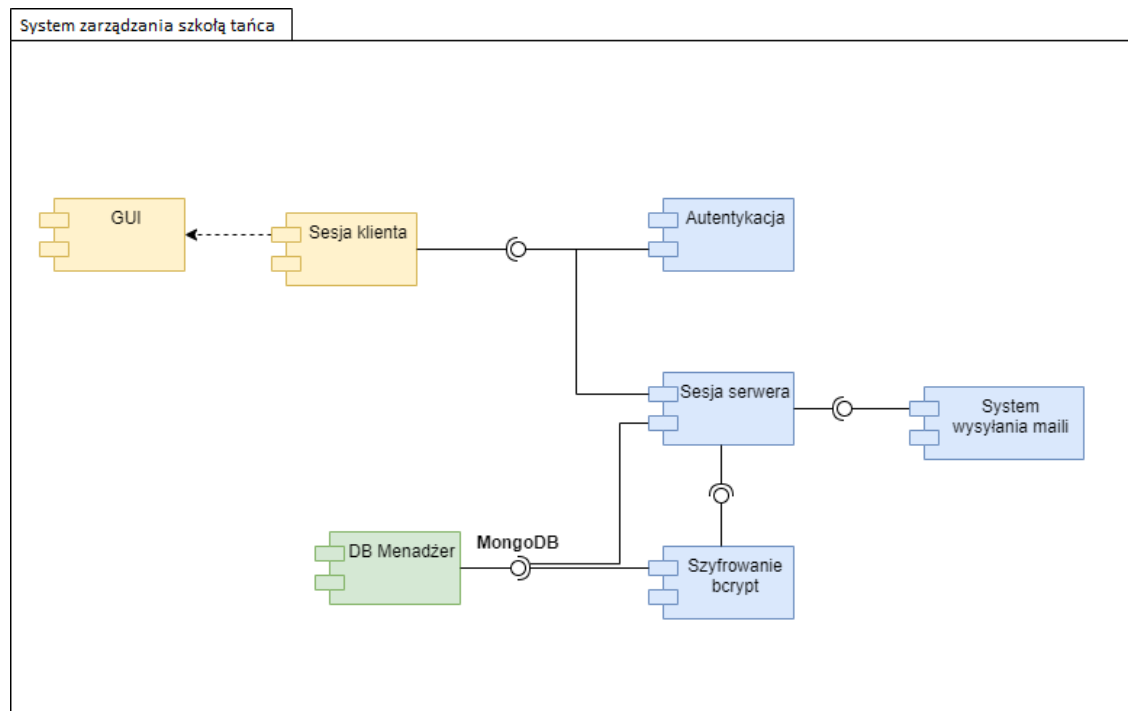
MongoDB, Node.js i Express.js są wykorzystane do stworzenia warstwy backend, która odpowiada za zarządzanie danymi, bazą danych, dokumentami HTML, żadaniami HTTP itp. [25]. Za jej pomocą są tworzone adresy URL, dzięki którym jest możliwe tworzenie, odczytywanie, modyfikacja i usuwanie danych przechowywanych w MongoDB. Adresy URL są reprezentowane przez funkcje, których inicjatorami są wywołania HTTP. Dane wysyłane są za pośrednictwem żądań, a serwer jest odpowiedzialny za modyfikację bazy danych i odesłanie odpowiedzi w formacie JSON, który jest używany przez wszystkie technologie tworzące stos technologiczny MERN.

React ma za zadanie wykonać żądania HTTP przy pomocy biblioteki Axios.js. Dzięki temu jest możliwe dynamiczne pobieranie danych bez konieczności przeładowywania strony internetowej. W rezultacie aplikacja webowa napisana przy pomocy framework'a React jest szybsza niż inne strony.

## 4.5. Diagram komponentów

Diagram komponentów UML jest narzędziem wizualizacji architektury systemów informatycznych. Są istotne przy projektowaniu i analizie systemów informatycznych. Pozwalają na przedstawienie, jak poszczególne elementy systemu są ze sobą powiązane. Dzięki nim łatwiej jest zidentyfikować elementy systemu, które wymagają testowania lub które są najbardziej narażone na awarie [26].

Podczas projektowania systemu do zarządzania szkołą tańca stworzono diagram komponentów, który użyto podczas implementacji. Diagram UML został przedstawiony na Rysunku 9, gdzie siedem komponentów zostało podzielone kolorystycznie na trzy warstwy systemu: klienta, serwera oraz bazy danych. Sesja klienta oraz komponent GUI odpowiadają za interfejs graficzny aplikacji. Sesja serwera została wykorzystana do przetwarzania żądań HTTP otrzymanych od klienta oraz obsługuje API. Komponent Autentykacja odpowiada za proces uwierzytelniania użytkownika. Szyfrowanie bcrypt przedstawia kodowanie haseł użytkowników, które następnie są przechowywane w bazie danych MongoDB. Natomiast System wysyłania maili, korzystając z sesji serwera ma za zadanie nadawać wiadomości poprzez aplikację.

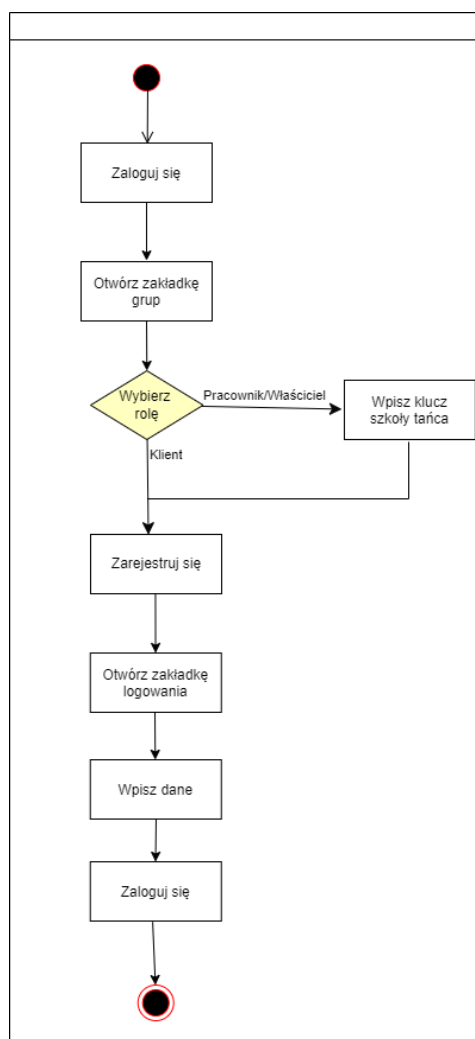


Rysunek 9. Diagram komponentów UML dla systemu zarządzania szkołą tańca

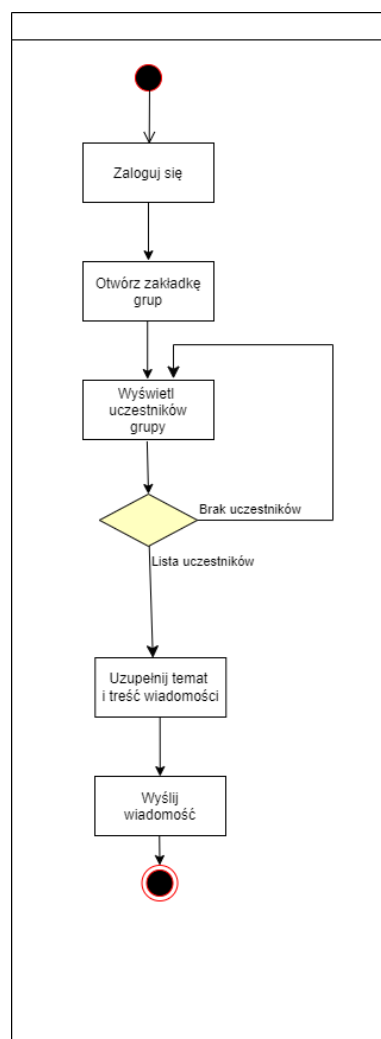
## 4.6. Diagramy czynności

Diagram czynności jest jednym z rodzajów diagramów UML. Służy do modelowania i przedstawiania zachowań systemów. Jest to graficzna reprezentacja sekwencji aktywności wykonywanych w systemie informatycznym. Dzięki diagramom czynności można analizować, projektować i udokumentować procesy oraz zachowania systemu. Co więcej, pozwala na zrozumienie procesu, poprawę jego efektywności i ułatwia jego ewentualne rozwijanie [26].

Podczas projektowania systemu stworzone zostały dwa diagramy czynności. Pierwszy z nich, widoczny na Rysunku 10, pokazuje proces rejestracji i logowania użytkownika. Drugi z diagramów, przedstawiony na Rysunku 11, opisuje działanie procesu wysyłania wiadomości email do uczestników grupy zajęć tańca.



Rysunek 10. Diagram aktywności dla procesu rejestracji i logowania użytkownika w systemie



Rysunek 11. Diagram aktywności dla procesu wysyłania wiadomości email przez pracownika/właściciela szkoły tańca

## 4.7. Projekt bazy danych

Projekt systemu wspomagającego zarządzanie szkołą tańca przewiduje wielu klientów, a także planuje się jego rozwój w przyszłości. Z tego powodu, do przechowywania danych wybrano dokumentową, nierelacyjną bazę danych. Tego typu bazy danych są bardzo przydatne w przypadku, gdy musimy przetwarzać duże ilości danych, a także zapewnić ich łatwą skalowalność.

Zaprojektowanie bazy danych wiąże się z zaplanowaniem kolekcji, które są potrzebne do projektu. Celem wcześniejszego zaprojektowania bazy nierelacyjnej jest szybsza implementacja danych. Nie jest to niezbędne, ponieważ MongoDB nie ma określonej sztywnej struktury. Natomiast jest to pomocne przy większych projektach. W tabelach 6, 7, 8, 9, 10 zostały umieszczone struktury kolekcji użytych do budowania bazy danych na potrzeby stworzenia aplikacji webowej.

*Tabela 6. Przedstawienie struktury kolekcji Users w MongoDB*

Nazwa kolekcji	Users
Opis kolekcji	Kolekcja przechowuje dokumenty, które zawierają dane użytkowników, którzy się zarejestrowali
Klucze kolekcji	<ul style="list-style-type: none"><li>• Email</li><li>• Hasło (w formie zakodowanej)</li><li>• Imię użytkownika</li><li>• Nazwisko użytkownika</li><li>• Rola (klient/pracownik/właściciel szkoły tańca)</li><li>• Numer telefonu</li><li>• Płeć (kobieta/mężczyzna/inna)</li><li>• Wiek</li></ul>

*Tabela 7. Przedstawienie struktury kolekcji Clients w MongoDB*

Nazwa kolekcji	Clients
----------------	---------

Opis kolekcji	Kolekcja zbierająca dokumenty na temat klientów, czyli użytkowników zarejestrowanych, którzy przynajmniej raz zarezerwowali miejsce na zajęciach tanecznych
Klucze kolekcji	<ul style="list-style-type: none"> <li>• Email</li> <li>• Imię użytkownika</li> <li>• Nazwisko użytkownika</li> <li>• Numer telefonu</li> </ul>

*Tabela 8. Przedstawienie struktury kolekcji Reservations w MongoDB*

Nazwa kolekcji	Reservations
Opis kolekcji	Kolekcja zawierająca dokumenty z danymi rezerwacji dokonanych przez klientów i pracowników szkoły tańca.
Klucze kolekcji	<ul style="list-style-type: none"> <li>• Email</li> <li>• Imię użytkownika</li> <li>• Nazwisko użytkownika</li> <li>• Numer telefonu</li> <li>• Rodzaj płatności</li> <li>• Numer i nazwa grupy</li> </ul>

*Tabela 9. Przedstawienie struktury kolekcji Groups w MongoDB*

Nazwa kolekcji	Groups
Opis kolekcji	Kolekcja mająca za zadanie zbierać dokumenty o utworzonych grupach przez właściciela szkoły tańca.
Klucze kolekcji	<ul style="list-style-type: none"> <li>• Nazwa/numer grupy</li> <li>• Poziom grupy</li> <li>• Maksymalna liczebność grupy</li> <li>• Dzień tygodnia</li> <li>• Godzina rozpoczęcia zajęć</li> </ul>

*Tabela 10. Przedstawienie struktury kolekcji PresenceLists w MongoDB*

Nazwa kolekcji	PresenceLists
Opis kolekcji	Kolekcja, w której znajdują się dokumenty z obecnościami klientów na zajęciach grupowych szkoły tańca.
Klucze kolekcji	<ul style="list-style-type: none"> <li>• Imię klienta</li> </ul>

	<ul style="list-style-type: none"> <li>• Nazwisko klienta</li> <li>• Nazwa/numer grupy</li> <li>• Data wprowadzenia obecności do systemu</li> <li>• Godzina wprowadzenia obecności do systemu</li> </ul>
--	--

## 5. Implementacja systemu

Do stworzenia aplikacji webowej typu fullstack został użyty głównie zbiór oprogramowania MERN Stack, czyli MongoDB, Express, React i Node.js. W poniższych podpunktach przedstawiono przykładowy kod odpowiedzialny za prawidłowe działanie aplikacji webowej.

### 5.1. Komunikacja z bazą danych

Połączenie z bazą danych zapewnione jest dzięki bibliotece mongoose i leży po stronie backend'owej aplikacji webowej. Kod zapewniający to połączenie został wyszczególniony na Listingu 1.

```
const mongoose = require("mongoose");
require("dotenv").config();

mongoose.connect(process.env.uri);
```

*Listing 1. Połączenie aplikacji webowej z bazą danych MongoDB*

Podczas połączenia z bazą danych użyta jest zmienna środowiskowa o nazwie „uri” i jest równoznaczna z danymi wpisanymi do schematu. Forma schematu jest pokazana na Listingu 2.

```
mongodb://[username:password@]host1[:port1][,...hostN[:portN]][/[defaulta
uthdb][?options]]
```

*Listing 2. Schemat zmiennej potrzebnej do połączenia aplikacji webowej z MongoDB*

Aby stworzyć kolekcje dokumentów w bazie danych użyto modeli i schematów, które zebrano w jednym folderze po stronie backend'owej. Przykład modelu został pokazany na Listingu 3.

```
const mongoose = require("mongoose");

const ClientSchema = new mongoose.Schema({
  email: {
    type: String,
    required: true,
  },
  name: String,
  surname: String,
  phone: String,
  payment: String,
});

const ClientModel = mongoose.model("Clients", ClientSchema);
```



```
module.exports = ClientModel;
```

*Listing 3. Model kolekcji MongoDB o nazwie "Clients"*

Przy implementacji systemu dla szkół tańca wykorzystano operacje CRUD, czyli zapisywanie, pobieranie, modyfikowanie i usuwanie danych z bazy. Uzyskano to przy pomocy metod dostępnych dla poniższych transakcji w MongoDB w połączeniu z metodami GET, POST, PUT i DELETE:

- `db.collection.aggregate()`
- `db.collection.count()`
- `db.collection.find()`
- `db.collection.findById()`
- `db.collection.findByIdAndRemove()`
- `db.collection.findOne()`
- `db.collection.sort()`

Przykład transakcji metodą GET oraz `db.collection.find()` został przedstawiony na Listingu 4.

```
app.get("/getClient", (req, res) => {  
  ClientModel.find({}, (err, result) => {  
    if (err) {  
      res.json(err);  
    } else {  
      res.json(result);  
    }  
  });  
});
```

*Listing 4. Transakcja MongoDB metodą GET pozwalająca odczytać wszystkie dokumenty znajdujące się w kolekcji Clients*

## 5.2. Szyfrowanie haseł

Szyfrowanie haseł to proces, dzięki któremu system przekształca jawne hasła na formy, które są trudne do bezpośredniego odczytania przez człowieka. Aby zapewnić bezpieczeństwo użytkownikom rejestrującym się w systemie, użyto biblioteki hashującej `bcrypt`. Dzięki tej bibliotece, podczas rejestracji hasło zapisywane jest w bazie danych jako skrót. Hasła szyfrowane są wraz z solą, czyli losowo wygenerowanym ciągiem znaków, który jest dodawany do hasła przed szyfrowaniem. Dzięki temu nawet jeśli dwóch użytkowników wpisze takie same hasło, to ich skróty

będą różne, ponieważ sole będą miały różne wartości. Tworzenie nowego użytkownika w bazie danych z zaszyfrowanym hasłem zostało przedstawione na Listingu 5.

```
app.post("/register", async (req, res) => {
  const user = req.body;

  bcrypt.hash(user.password, saltRounds, (err, hash) => {
    if (err) {
      console.log(err);
    }

    user.password = hash;
    const newUser = new UserModel(user);
    newUser.save();

    console.log("dodano użytkownika");
  });
});
```

*Listing 5. Metoda pozwalająca na rejestrację użytkownika przy użyciu szyfrowania hasła*

W przypadku logowania, hasło podane przez użytkownika również jest zaszyfrowane za pomocą bcrypt, a następnie porównane z tym, które znajduje się w bazie danych. Jeśli skróty się zgadzają, to użytkownik ma prawidłowe hasło, w przeciwnym razie, próba logowania kończy się niepowodzeniem. Logowanie zostało pokazane na Listingu 6.

```
app.post("/login", async (req, res) => {
  const email = req.body.email;
  const password = req.body.password;

  UserModel.findOne({ email: email }, (err, result) => {
    if (err) {
      res.send({ message: "Nieprawidłowy email" });
    }
    if (result) {
      bcrypt.compare(password, result.password, (error, response) => {
        if (response) {
          res.json(result);
        }
      });
    }
  });
});
```

```

        console.log("Zalogowano");
    } else {
        res.send({ message: "Niewłaściwe hasło!" });
    }
});
} else {
    res.send({
        message: `Nie istnieje użytkownik o podanym emailu: ${email}`,
    });
}
});
});

```

*Listing 6. Metoda pozwalająca na logowanie użytkownika przy użyciu szyfrowania hasła*

### 5.3. Moduł wysyłania wiadomości email

W celu zaimplementowania funkcjonalności pozwalającej na wysyłanie wiadomości na adresy email klientów zapisanych do konkretnych grup tańca, użyto biblioteki Nodemailer. Aby wysłać wiadomość z istniejącego adresu emailowego, najpierw połączono aplikację z kontem przypisanym do tego adresu. Następnie dostosowano konfigurację wiadomości, aby moduł wysyłał napisaną przez użytkownika treść oraz użyto metodę, która finalizuje cały proces. Funkcjonalność została uzyskana za pomocą kodu, który widoczny jest na Listingu 7.

```

const transporter = nodemailer.createTransport({
  service: "gmail",
  auth: {
    user: process.env.EMAIL,
    pass: process.env.PASSWORD,
  },
});

const mailOptions = {
  from: "info@wichrowskamaria.pl",
  bcc: `${emailString}`,
  subject: `${subject}`,

```

```

    text: `${message}`,
  };

  transporter.sendMail(mailOptions, function (error, info) {
    if (error) {
      console.log(error);
      res.json(error);
    } else {
      console.log("Email sent: " + info.response);
      res.json(info.response);
    }
  });
});

```

*Listing 7. Zmienne oraz metoda pozwalająca na wysłanie wiadomości email poprzez aplikację webową*

## 5.4. Routing

Aplikacja została zaimplementowana z pomocą biblioteki React Router, która odpowiedzialna jest za zarządzanie routingiem w aplikacjach wykorzystujących React. Pozwala ona na tworzenie jednostronicowych aplikacji (Single-Page Applications), które mają wiele widoków, lecz można je przeglądać bez ładowania całej strony ponownie. Przykładowo zaimplementowany routing w aplikacji został przedstawiony na Listingu 8.

`createBrowserRouter` to jedna z wielu metod, które są dostępne w React Router, a służy do tworzenia obiektu routera, który można skonfigurować z trasami w aplikacji. `Outlet` jest komponentem, który reprezentuje miejsce w aplikacji, gdzie będą renderowane dzieci komponentów, które pasują do bieżącej ścieżki URL. `RouterProvider` służy do dostarczania kontekstu routera dla innych komponentów w aplikacji.

```

import { createBrowserRouter, Outlet, RouterProvider } from "react-router-dom";

let Layout = () => {
  return (
    <div id="main-wrapper">
      <Navbar />
      <Outlet />
      <Footer />
    </div>
  );
};

```

```

    );
  });

const router = createBrowserRouter([
  {
    path: "/",
    element: <Layout />,
    children: [
      {
        path: "/",
        element: <MainPage />,
      },
      {
        path: "/login",
        element: <Login />,
      },
      {
        path: "/register",
        element: <Register />,
      }
    ],
  },
]);

function App() {
  return (
    <div className="app">
      <div className="container">
        <RouterProvider router={router} />
      </div>
    </div>
  );
}

export default App;

```

*Listing 8. Fragment kodu aplikacji webowej odpowiedzialny za routing*

## 5.5. Hooki React

Aby zaimplementować interfejs użytkownika, zastosowano bibliotekę React. Korzystając z funkcjonalności komponentów, użyto funkcji nazywanych hookami. Umożliwiają one dodawanie stanu do komponentów funkcyjnych, korzystanie z cyklu życia komponentów, zarządzanie stanem i wykonanie innych działań związanych z logiką biznesową. W projekcie użyto takich hooków jak: `useState`, `useEffect`, `useRef`.

### 5.5.1. useState

useState to jeden z hooków w React, który pozwala na korzystanie ze stanu w komponentach. Dzięki hookowi jesteśmy w stanie przechowywać wartości wewnątrz komponentu i aktualizować je podczas działania aplikacji. useState zwraca tablicę z dwoma elementami: aktualną wartością stanu oraz funkcją pozwalającą na zmianę stanu. Gdy wywołujemy tę funkcję, React przerysowuje komponent z nowym stanem i wywołuje zaktualizowaną funkcję render. Przykład użycia useState został przedstawiony na Listingu 9.

```
import { useState } from "react";

export const Register = () => {

  const [password, setPassword] = useState("");
  const [validPassword, setValidPassword] = useState(false);
  const [passwordFocus, setPasswordFocus] = useState(false);
}
```

*Listing 9. Fragment kodu komponentu z hookiem useState do rejestracji w aplikacji webowej*

### 5.5.2. useEffect

useEffect to hook w React, który umożliwia wykonywanie efektów ubocznych w komponencie. Efekty uboczne to operacje, które nie są związane z renderowaniem komponentu, takie jak wykonywanie zapytań do serwera, subskrypcja do strumieni danych, manipulacja DOM lub synchronizacja stanu między komponentami. useEffect pozwala na definiowanie funkcji, która będzie wykonywana po każdym renderowaniu komponentu lub tylko wtedy, gdy określona wartość zmiennej się zmieni. Funkcja ta może zwracać inny callback, który będzie wywołany po usunięciu komponentu z drzewa DOM lub przy następnym wywołaniu useEffect. Przykładowy fragment kodu implementacji systemu, który odpowiedzialny jest za aktualizowanie stanu poprawności wpisanych haseł (validPassword i validMatch) w zależności od zmian wartości hasła (password) i powtórnego hasła (matchPassword) jest widoczny na Listingu 10.

```
useEffect(() => {
  setValidPassword(PWD_REGEX.test(password));
```

```
setValidMatch(password === matchPassword);  
}, [password, matchPassword]);
```

*Listing 10. Fragment kodu komponentu z hookiem useEffect do rejestracji w aplikacji webowej*

### 5.5.3. useRef

useRef jest hookiem, który pozwala na tworzenie referencji do elementów w drzewie DOM lub do innego elementu, który jest utworzony przy pomocy hooków. Można go użyć do przechowywania i aktualizacji wartości, które mają przetrwać cykl życia komponentu, nie powodując re-renderingu. useRef pozwala na utworzenie obiektu referencyjnego, który może być wykorzystany do przechowywania wartości między renderowaniami. Przykładowy fragment kodu, w którym zmienna useRef przechowuje referencję do pola email, pokazany jest na Listingu 11. Przy pomocy useEffect, używając metody .focus() na useRef.current. To oznacza, że po załadowaniu komponentu, focus będzie ustawiony na polu email, a użytkownik będzie mógł od razu wpisywać adres e-mail.

```
const useRef = useRef();  
  
useEffect(() => {  
  useRef.current.focus();  
}, []);
```

*Listing 11. Fragment kodu komponentu z hookiem useRef do rejestracji w aplikacji webowej*

## 6. Dokumentacja użytkownika

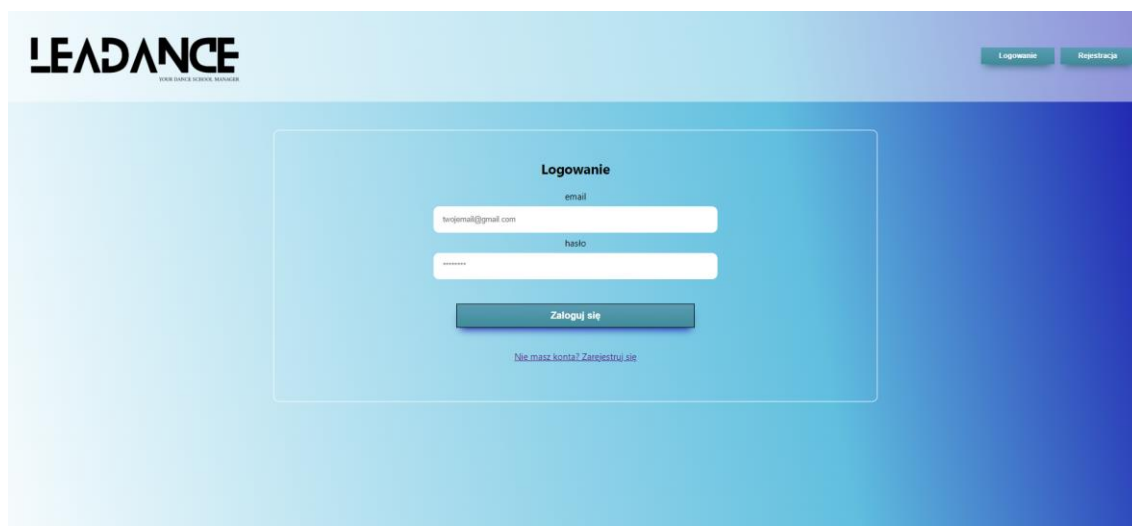
Dokumentacja użytkownika przedstawia działanie systemu aplikacji webowej o nazwie „Leadance”. Jest to aplikacja do zarządzania szkołą tańca, która pozwala przyspieszyć proces rezerwacji zajęć. Każdy zalogowany w systemie użytkownik ma przypisaną jedną z trzech ról: klient, pracownik, właściciel szkoły tańca. W zależności od roli, dostępne są różne funkcjonalności aplikacji.

## 6.1. Klient

Klient ma możliwość sprawdzenia w aplikacji dostępnych grup, dokonania rezerwacji oraz wglądu we wszystkie swoje dotychczasowe zapisy.

### 6.1.1. Zakładanie konta

W celu założenia konta należy wejść na stronę aplikacji webowej i wcisnąć przycisk Rejestracja. Widok użytkownika niezalogowanego widoczny jest na Rysunku 12. Następnie należy wypełnić formularz rejestracyjny, który przedstawiony jest na Rysunku 13.



Rysunek 12. Zrzut ekranu aplikacji webowej Leadance z widokiem użytkownika niezalogowanego.



**Rejestracja**

Email  
twojemail@gmail.com

Hasło  
\*\*\*\*\*

Potwierdź hasło  
\*\*\*\*\*

Imię  
Imię

Nazwisko  
Nazwisko

Numer telefonu  
500123456

Wiek  
Wiek

Płeć  
Kobieta

Kim jesteś?  
☐ Klientem
 ☐ Pracownikiem
 ☐ Właścicielem

**Zarejestruj się**

[Masz konto? Zaloguj się](#)

Rysunek 13. Formularz rejestracji aplikacji Leadance.

W trakcie wypełniania formularza pojawiają się kolorowe ikony pokazujące, czy wpisane wartości są prawidłowe, a jeżeli nie – podpowiedź jak wypełnić dane pole poprawnie. Obie sytuacje zostały przedstawione na Rysunku 14. Po wypełnieniu formularza rejestracji prawidłowymi danymi i zatwierdzeniu go przyciskiem „Zarejestruj się” pojawi się informacja o pomyślnym utworzeniu konta (Rysunek 15).

**Rejestracja**

Email ✓  
230035@edu.p.lodz.pl

Hasło ✗  
\*\*\*\*\*

**Hasło musi zawierać**  
 - 8-24 znaków  
 - małą i wielką literę  
 - cyfrę  
 - znak specjalny  
 Dozwolone znaki specjalne: ! @ # \$ %

Rysunek 14. Część panelu rejestracji z ikonami wskazującymi na poprawność danych oraz podpowiedzią w przypadku błędnego wpisania danych



*Rysunek 15. Komunikat po udanej rejestracji.*

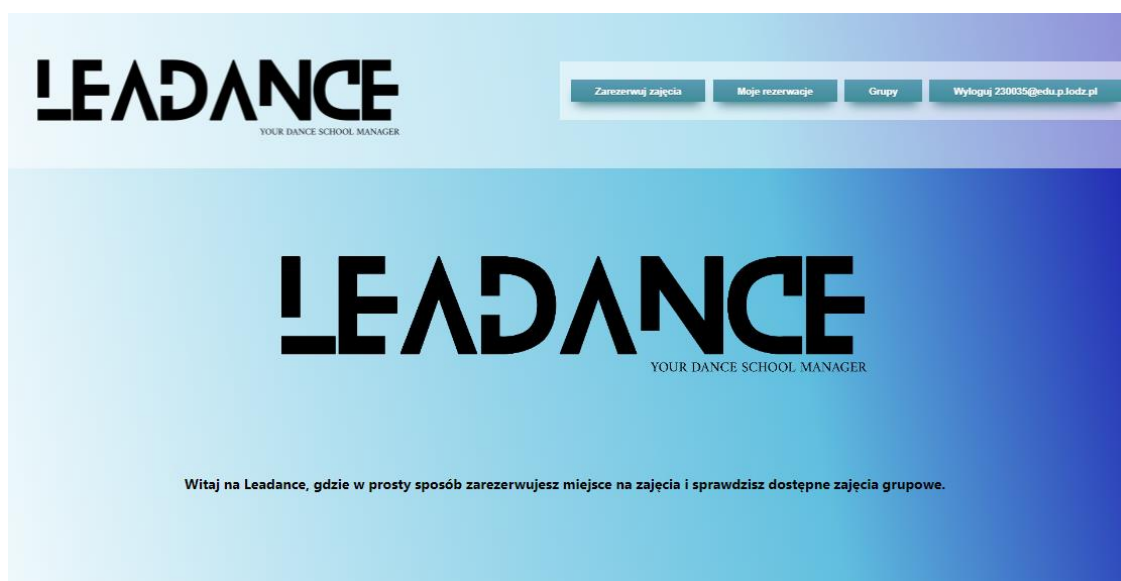
## 6.1.2. Logowanie

W celu zalogowania się do aplikacji webowej, należy wpisać adres email oraz hasło podane wcześniej podczas rejestracji i kliknąć przycisk „Zaloguj się”. Panel logowania widoczny jest na Rysunku 16.

A light blue rectangular box with a darker blue gradient at the bottom. In the center, the word "Logowanie" is written in a bold, black font. Below it, the word "email" is centered above a white input field with a black border, containing the text "twojemail@gmail.com". Below the input field, the word "hasło" is centered above another white input field with a black border, containing several asterisks. Below the password field, a dark blue button with a black border and the text "Zaloguj się" in white is centered. At the bottom, the text "Nie masz konta? Zarejestruj się" is centered and underlined in a smaller, black font.

*Rysunek 16. Panel logowania do aplikacji webowej Leadance*

Po zalogowaniu się do systemu następuje przekierowanie na stronę główną przedstawioną na Rysunku 17.



Rysunek 17 Strona główna dla klienta zalogowanego

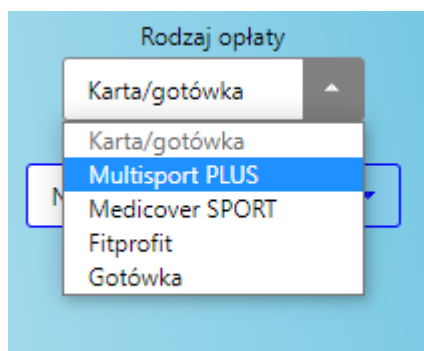
### 6.1.3. Rezerwacja zajęć

W celu zarezerwowania zajęć, klient musi wpisać podstawowe informacje o sobie. Widok formularza rezerwacji jest widoczny na Rysunku 18. Pole z adresem email wypełniane jest automatycznie. Pozostałe pola należy wypełnić ręcznie ze względu na to, że zarejestrowany klient może wprowadzić rezerwacje również dla innych osób.

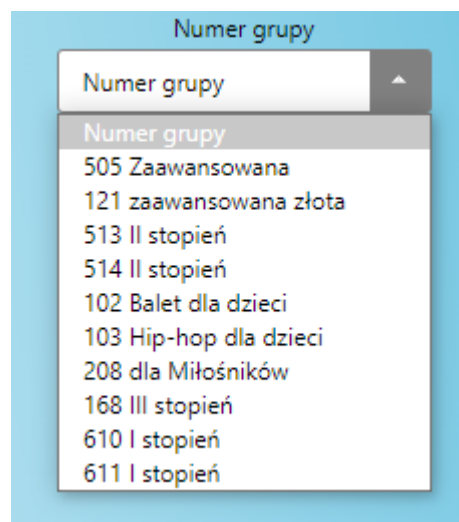
The screenshot shows the 'Rezerwacja' (Reservation) form on the Leadance website. The form is titled 'Rezerwacja' and contains several input fields: 'Email' (pre-filled with '230035@edu.p.lodz.pl'), 'Imię klienta' (Client's name), 'Imię' (First name), 'Nazwisko klienta' (Client's surname), 'Nazwisko' (Surname), 'Numer telefonu' (Phone number), 'Rodzaj opłaty' (Payment type) with a dropdown menu showing 'Karta/gotówka', 'Numer grupy' (Group number), and another 'Numer grupy' dropdown menu. At the bottom right of the form is a button labeled 'Potwierdzam rezerwację'.

Rysunek 18. Panel rezerwacji zajęć

Rodzaj opłaty to pole rozwijane, gdzie można wybrać jedną z obecnie dostępnych opcji płatności: jedną z kart sportowych lub gotówkę. Lista rozwijana pokazana jest na Rysunku 19. Numer grupy również jest listą rozwijaną. Lista grup jest widoczna na Rysunku 20.



Rysunek 19. Lista rozwijana rodzaju opłaty



Rysunek 20. Lista rozwijana z numerami grup

#### 6.1.4. Przeglądanie rezerwacji

Zalogowany klient może przeglądać wszystkie dokonane przez siebie rezerwacje. W tym celu wybiera z menu głównego, u góry strony, zakładkę „Moje rezerwacje”. Po kliknięciu przycisku wyświetlają się rezerwacje w formie tabeli, przedstawionej na Rysunku 21.

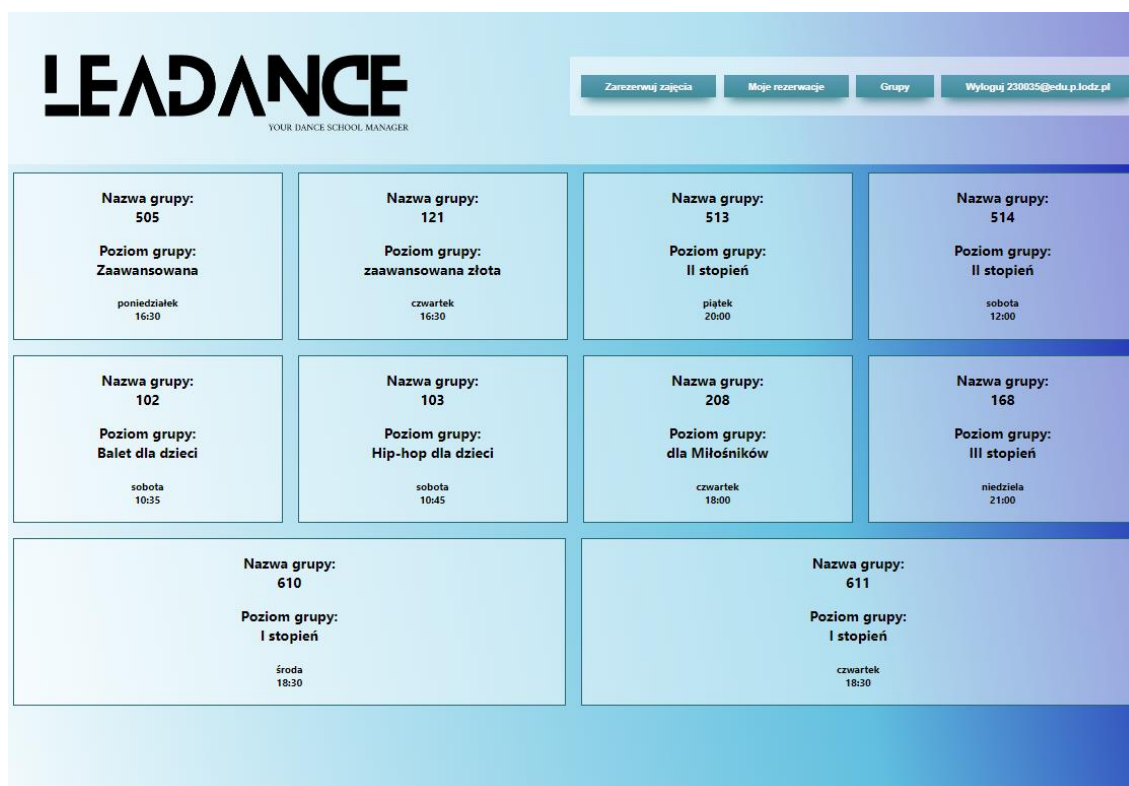


Rysunek 21. Widok zakładki "Moje rezerwacje"

#### 6.1.5. Przeglądanie dostępnych grup

Wyświetlenie wszystkich dostępnych grup, na które klient może się zapisać następuje po kliknięciu przycisku „Grupy” z menu głównego. Po wybraniu tej opcji,

poniżej menu wyświetlają się opisy grup wraz z poziomem i terminami zajęć. Spis dostępnych grup widoczny jest na Rysunku 22.



Rysunek 22. Widok zakładki "Grupy" dla zalogowanego klienta

### 6.1.6. Wylogowanie

Wylogowanie następuje po kliknięciu przycisku „Wyloguj” znajdującego się po prawej stronie menu głównego. Menu główne przedstawione zostało na Rysunku 23.



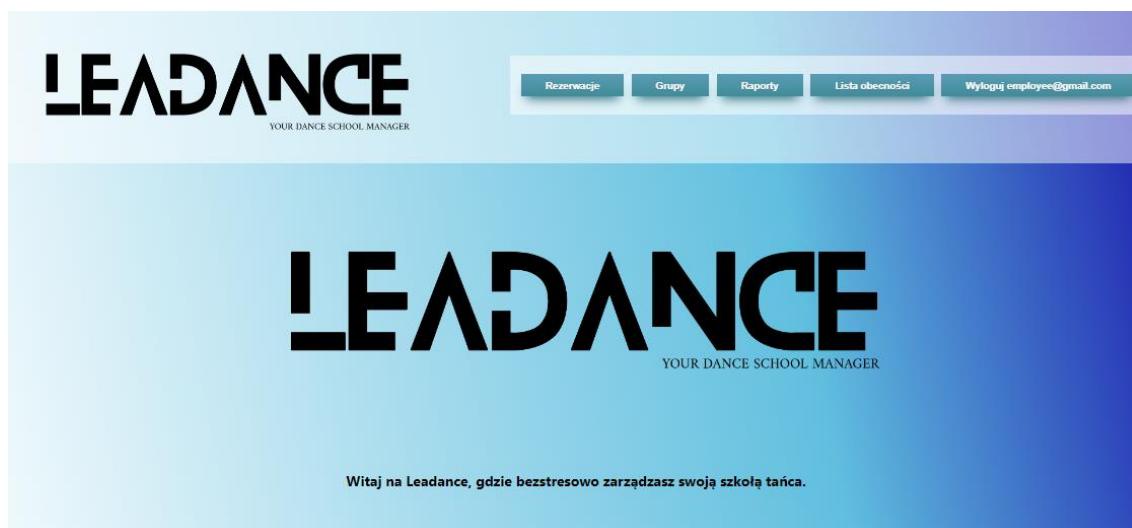
Rysunek 23. Menu główne dla zalogowanego użytkownika w roli klienta

Po wylogowaniu użytkownik zostaje przeniesiony do strony z panelem logowania.

## 6.2. Pracownik

Konto pracownika posiada więcej uprawnień niż konto klienta. Rejestracja i logowanie przeprowadzane są w ten sam sposób, jak w przypadku klienta.

Po zalogowaniu pracownikowi wyświetla się strona główna przypisana do jego roli, która przedstawiona jest na Rysunku 24.



*Rysunek 24. Strona główna po zalogowaniu się do aplikacji Leadance jako pracownik*

### 6.2.1. Przeglądanie i tworzenie rezerwacji

W przypadku, gdy klienci nie mają dostępu do aplikacji internetowej, mogą zostać zapisani przez pracownika. W tym celu pracownik wybiera zakładkę „Rezerwacje”, a następnie przycisk „Nowa rezerwacja”. Panel rezerwacji został przedstawiony na Rysunku 25.

Rysunek 25. Panel rezerwacji dla zalogowanego pracownika

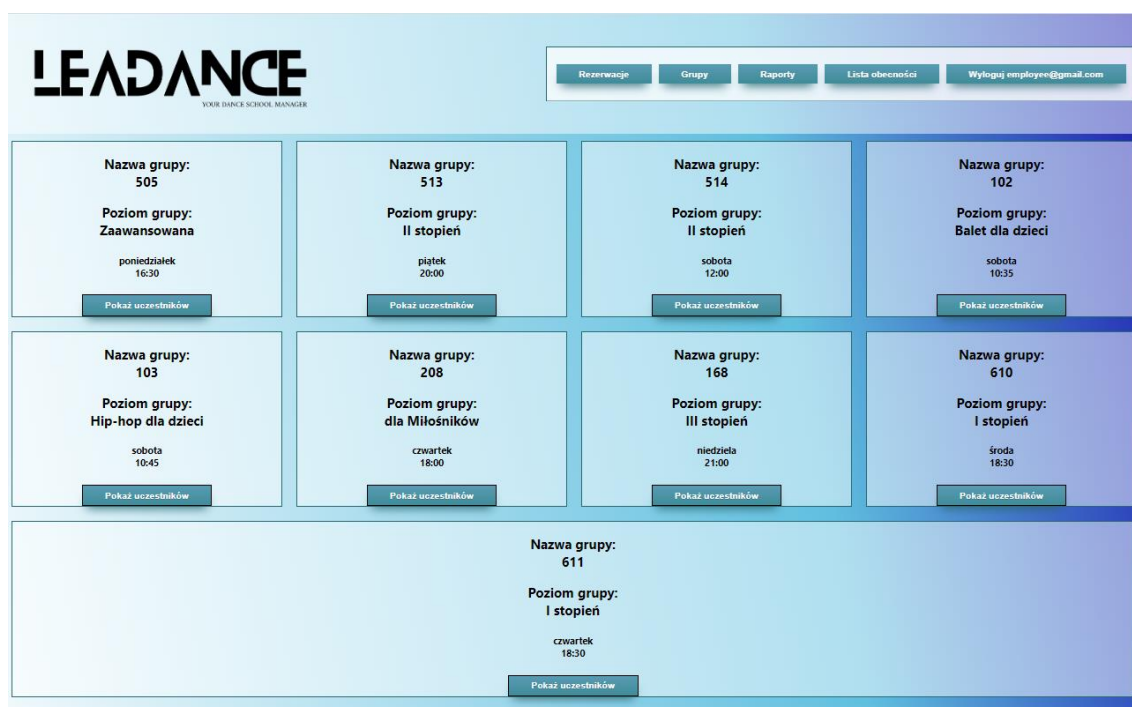
Po kliknięciu przycisku „Lista rezerwacji” wyświetla się tabela z rezerwacjami, widoczna na Rysunku 26.

Maria	Wichrowska	900900900	multisport	grupa 611 I stopień
Maria	Wichrowska	732718699	multisport	grupa 513 II stopień
Maria	Wichrowska	898989898	multisport	grupa 611 I stopień
Piotr	Kędziora	833833833	multisport	grupa 611 I stopień

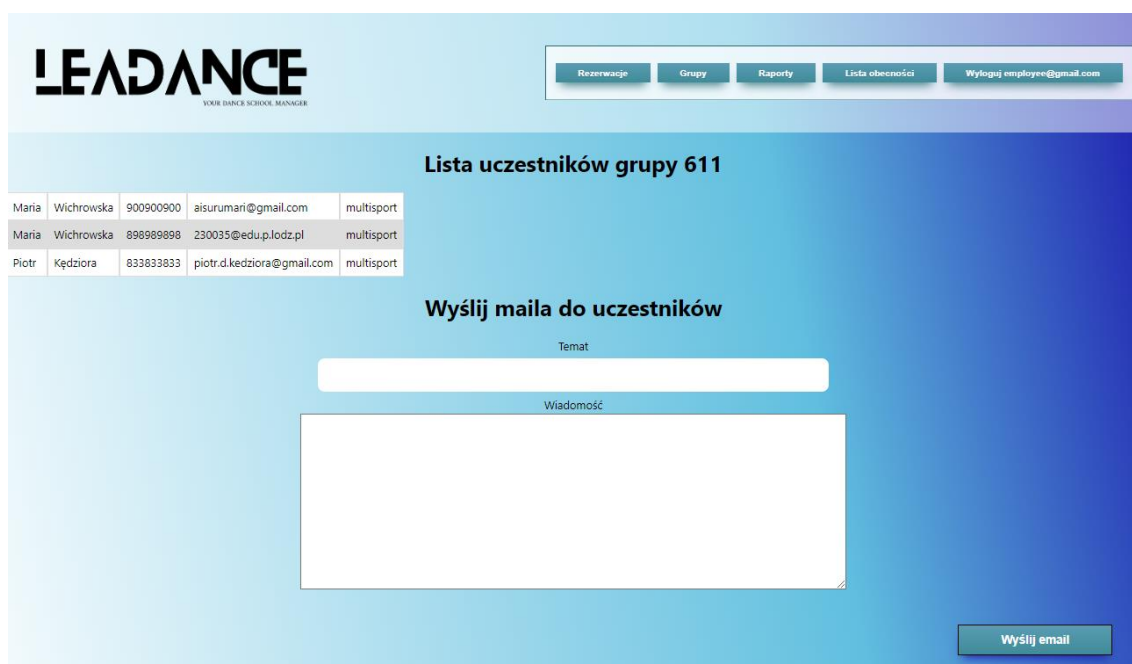
Rysunek 26. Lista rezerwacji widoczna dla zalogowanego pracownika

## 6.2.2. Wysyłanie wiadomości email

W celu wysłania wiadomości email do wszystkich uczestników grupy zajęć tanecznych, należy z panelu pracownika przejść z zakładki spisu grup do listy uczestników. Lista grup widoczna dla pracownika jest widoczna na Rysunku 27. Następnie ukazuje się widok listy ze spisem kursantów wraz z polem na wpisanie tematu oraz treści wiadomości, który pokazany został na Rysunku 28.



Rysunek 27. Lista grup widoczna dla pracownika



Rysunek 28. Lista uczestników wraz z panelem do wysyłania wiadomości email

### 6.2.3. Tworzenie raportów i list

Z panelu pracownika istnieje możliwość wygenerowania raportów. Są to raporty dotyczące średniego wieku klientów i liczebności kobiet, mężczyzn oraz osób



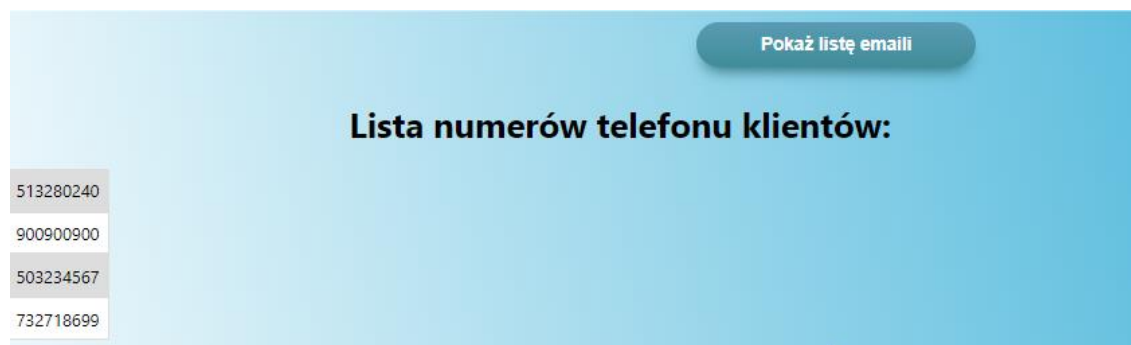
niebinarnych. Dzięki raportom można przeanalizować na jakich grupach docelowych klientów powinna się skupiać szkoła tańca.

W celu wygenerowania raportu, należy wcisnąć przycisk „Raporty” dostępny w menu głównym, a następnie przycisk „Pokaż statystyki”. Widok statystyk ukazany jest na Rysunku 29.

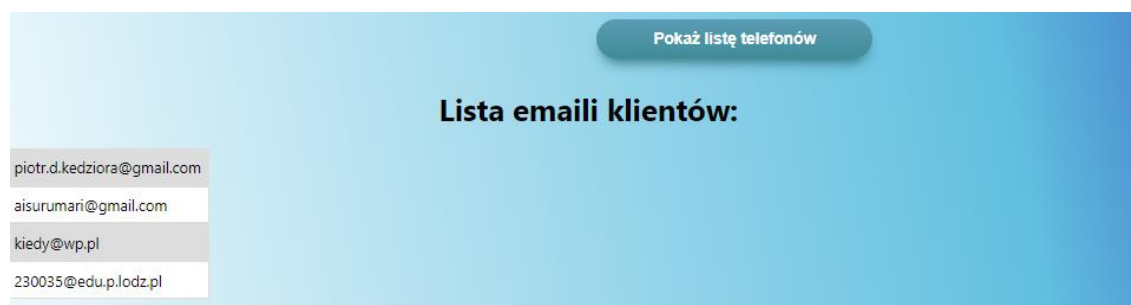
Pracownik szkoły tańca może również wygenerować listy z numerami telefonów klientów oraz listy z adresami email klientów. Aby wyświetlić te listy, użytkownik musi przycisnąć odpowiednio przyciski „Numery telefonów” i „Emaille”. Te widoki zostały pokazane na Rysunku 30 i Rysunku 31.



Rysunek 29. Raport ze statystykami dostępny dla zalogowanego pracownika



Rysunek 30. Wygenerowana lista telefonów klientów dostępna dla zalogowanego pracownika



Rysunek 31. Wygenerowana lista emaili klientów dostępna dla zalogowanego pracownika

## 6.2.4. Lista obecności

Pracownik może zaznaczać obecności klienta po kliknięciu zakładki „Lista obecności”. Obecności wyświetlane są w kolejności od ostatniej dodanej. Zakładka obecności widoczna jest na Rysunku 32.

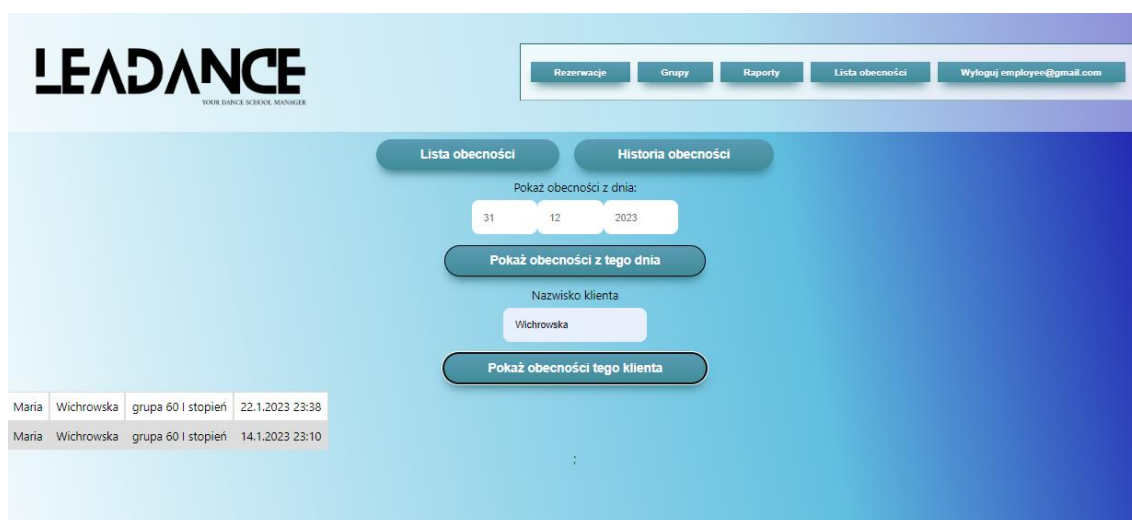
Historię obecności klientów filtrowaną po dacie lub po nazwisku klienta można zobaczyć po kliknięciu przycisku „Historia obecności”. Panel historii obecności przedstawiony został na Rysunku 33 oraz 34.

Imię	Nazwisko	Data i czas
Maria	Wichrowska	grupa 60 I stopień 22.1.2023 23:38
Adam	Lowak	grupa 60 I stopień 15.1.2023 20:36
Maria	Wichrowska	grupa 60 I stopień 14.1.2023 23:10

Rysunek 32. Panel listy obecności

Imię	Nazwisko	Data i czas
Adam	Lowak	grupa 60 I stopień 15.1.2023 20:36

Rysunek 33. Panel historii obecności względem daty



Rysunek 34. Panel historii obecności względem nazwiska klienta

## 6.3. Właściciel szkoły tańca

Właściciel szkoły tańca posiada szersze uprawnienia od pracownika. Prócz możliwości skorzystania z opcji stworzonych dla pracownika, może również modyfikować, dodawać i usuwać grupy zajęć tańca.

### 6.3.1. Dodawanie grup

W celu dodania nowej grupy zalogowany właściciel szkoły tańca musi przejść do zakładki „Grupy” z menu głównego i tam wprowadzić wymagane dane. Panel nowej grupy pokazany jest na Rysunku 35. Zatwierdzenie operacji następuje poprzez wciśnięcie przycisku „Potwierdzam stworzenie nowej grupy”.

**LEADANCE**  
YOUR DANCE SCHOOL MANAGER

Rezerwacje Grupy Raporty Lista obecności Wyloguj danceSchoolOwner@gmail.com

Nowa grupa Lista grup

**Nowa grupa**

Nazwa grupy  
485

Poziom grupy  
I stopień

Maksymalna liczba kursantów

Liczba osób

Dzień tygodnia  
Wybierz dzień tygodnia

Godzina  
17:00

Potwierdzam stworzenie nowej grupy

Rysunek 35. Panel nowej grupy widoczny dla zalogowanego właściciela szkoły tańca

### 6.3.2. Modyfikowanie i usuwanie grupy

Właściciel szkoły tańca ma możliwość modyfikacji poziomu dodanej wcześniej grupy. Aby to zrobić, należy przejść do widoku listy grup i wcisnąć przycisk „Edytuj grupę”. Aby usunąć grupę należy przycisnąć „Usuń grupę”. Widok listy grup dla zalogowanego właściciela szkoły tańca pokazany jest na Rysunku 36.

**LEADANCE**  
YOUR DANCE SCHOOL MANAGER

Rezerwacje Grupy Raporty Lista obecności Wyloguj maria@gmail.com

Nowa grupa Lista grup

<p>Nazwa grupy: 505</p> <p>Poziom grupy: Zaawansowana</p> <p>poniedziałek 16:30</p> <p>Usuń grupę Edytuj grupę</p> <p>Pokaż uczestników</p>	<p>Nazwa grupy: 513</p> <p>Poziom grupy: II stopień</p> <p>piątek 20:00</p> <p>Usuń grupę Edytuj grupę</p> <p>Pokaż uczestników</p>	<p>Nazwa grupy: 514</p> <p>Poziom grupy: II stopień</p> <p>sobota 12:00</p> <p>Usuń grupę Edytuj grupę</p> <p>Pokaż uczestników</p>	<p>Nazwa grupy: 102</p> <p>Poziom grupy: Balet dla dzieci</p> <p>sobota 10:35</p> <p>Usuń grupę Edytuj grupę</p> <p>Pokaż uczestników</p>
<p>Nazwa grupy: 103</p> <p>Poziom grupy: Hip-hop dla dzieci</p> <p>sobota 10:45</p> <p>Usuń grupę Edytuj grupę</p> <p>Pokaż uczestników</p>	<p>Nazwa grupy: 208</p> <p>Poziom grupy: dla Miłośników</p> <p>czwartek 18:00</p> <p>Usuń grupę Edytuj grupę</p> <p>Pokaż uczestników</p>	<p>Nazwa grupy: 168</p> <p>Poziom grupy: III stopień</p> <p>niedziela 21:00</p> <p>Usuń grupę Edytuj grupę</p> <p>Pokaż uczestników</p>	<p>Nazwa grupy: 610</p> <p>Poziom grupy: I stopień</p> <p>środa 18:30</p> <p>Usuń grupę Edytuj grupę</p> <p>Pokaż uczestników</p>

Rysunek 36. Lista grup widoczna dla zalogowanego właściciela szkoły tańca

## 7. Podsumowanie

Celem pracy było zaprojektowanie systemu do zarządzania szkołą nauki tańca. W związku z tym najpierw przeprowadzone zostało badanie ankietowe. Na podstawie wyników ankiety, sformułowane zostały założenia systemu oraz wymagania funkcjonalne i niefunkcjonalne. Następnie opracowany został projekt aplikacji. Według ankietowanych, najważniejsze w takim systemie są: dostępność danych o klientach, widok grafiku/grup zajęć tanecznych, raporty w celu możliwości analizy dotychczasowych strategii zarządzania i marketingu. W celu graficznej reprezentacji systemu wykorzystane zostały diagramy UML, które przedstawiają przypadki użycia, komponenty oraz dwie przykładowe czynności w trakcie korzystania z systemu.

Na podstawie projektu aplikacji, zaimplementowano wszystkie założone funkcjonalności za pomocą stosu technologicznego MERN do tworzenia aplikacji webowych. Aplikacja została napisana w architekturze client-server, w której system jest podzielony na klienta, czyli interfejs użytkownika, oraz serwer, w którym znajduje się logika biznesowa oraz baza danych. Klient przesyła zapytania do serwera, który udziela odpowiedzi.

Przy pomocy MongoDB powstała baza danych, dzięki której wszystkie dane o klientach i zajęciach grupowych w szkole tańca zostaną zapisane cyfrowo. Przy użyciu framework'a React stworzono cztery wersje aplikacji, w zależności od roli pełnionej w szkole tańca. Poprzez Node.js oraz Express JS utworzono serwer, który pozwala na komunikację ze wszystkimi elementami aplikacji webowej.

Bezpieczeństwo w sieci jest jednym z najważniejszych aspektów, które muszą być brane pod uwagę przy projektowaniu aplikacji internetowych. W dzisiejszych czasach, kiedy przechowywanie danych i przekazywanie ich w sieci staje się coraz bardziej popularne, należy zwrócić szczególną uwagę na ochronę danych, które są przekazywane i pozyskiwane przez użytkowników. Mając na względzie osoby korzystające z aplikacji Leadance, podczas rejestracji i logowania hasła są zaszyfrowane za pomocą biblioteki bcrypt, która zapewnia wysoki poziom bezpieczeństwa. Dzięki temu, w przypadku ewentualnego wycieku danych z aplikacji, hasła użytkowników będą bezpieczne i nie będzie możliwe ich odczytanie.

W kontekście projektu, wykorzystano trzy protokoły: HTTP, SMTP oraz TCP. Pierwszy z nich, czyli HTTP (ang. Hypertext Transfer Protocol) jest wykorzystywany do przesyłania informacji w postaci stron internetowych pomiędzy serwerem a klientem. Protokół ten określa sposób, w jaki informacje są przesyłane, jakie metody są dozwolone oraz w jaki sposób następuje nawiązanie i zakończenie połączenia. Drugim wykorzystanym protokołem jest SMTP (ang. Simple Mail Transfer Protocol), który jest standardowym protokołem internetowym wykorzystywanym do przesyłania poczty elektronicznej. Protokół SMTP określa sposób, w jaki wiadomości e-mail są przekazywane pomiędzy serwerami pocztowymi oraz między klientem poczty a serwerem pocztowym. Trzeci wykorzystany w projekcie protokół to TCP (ang. Transmission Control Protocol). TCP jest protokołem warstwy transportowej, który umożliwia nawiązanie połączenia pomiędzy aplikacjami działającymi na różnych urządzeniach w sieci. Protokół ten jest używany do zapewnienia niezawodnej i poprawnej transmisji danych poprzez dzielenie przesyłanych informacji na mniejsze pakiety i ich numerowanie, a także zapewnianie ich kolejności i potwierdzanie ich odbioru. W projekcie wykorzystano TCP do zapewnienia niezawodnego i bezpiecznego połączenia z bazą danych MongoDB.

Aby ułatwić proces tworzenia aplikacji webowej, wykorzystano łącznie osiem różnych bibliotek. Wśród nich znalazły się m.in. React, który jest popularną biblioteką JavaScript wykorzystywaną do budowania interfejsów użytkownika, oraz React Router,

który umożliwia tworzenie dynamicznego routingu w aplikacji webowej. Do komunikacji z serwerem wykorzystano bibliotekę Axios, która pozwala na tworzenie zapytań HTTP w łatwy sposób. Aby ułatwić komunikację z bazą danych MongoDB, wykorzystano bibliotekę mongoose. Do wysyłania maili użyto biblioteki nodemailer. Na rozwiązanie problemów związanych z "CORS" - Cross-Origin Resource Sharing, czyli problemem dostępu do zasobów między różnymi domenami pozwoliła biblioteka cors. Do zabezpieczania haseł wykorzystano bibliotekę bcrypt. Wszystkie wrażliwe informacje takie jak hasła i klucze API przechowywano w pliku konfiguracyjnym .env, a odczyt tych wartości zrealizowano przy pomocy biblioteki dotenv.

Cały projekt składał się z dziewięciu różnych komponentów oraz dziesięciu różnych widoków aplikacji webowej, co pozwoliło na stworzenie intuicyjnego interfejsu użytkownika, który pozwala na łatwe korzystanie z aplikacji i przeglądanie zawartości.

Po stronie wizualnej aplikacji webowej, wybrano nazwę systemu, która nie występuje jako znak towarowy w Urzędzie patentowym - Leadance. Część nazwy „lead” nawiązuje do ważnego elementu w tańcu towarzyskim – prowadzenia. Słowo nawiązuje również do „kierowania” w odniesieniu do zespołu lub pracowników. Natomiast angielskie słowo „dance” jest tłumaczone jako taniec. Dzięki takiej nazwie aplikacji webowej, przedsiębiorca posiadający szkołę tańca wie, że system jest skierowany właśnie dla niego. Następnie na podstawie tej nazwy stworzono logo i ustalono kolorystykę warstwy wizualnej. Jak wskazały badania, kolory wpływają na emocje i zachowania odbiorców[27]. Kolor niebieski odnosi się w psychologii kolorów do zaufania, profesjonalizmu, mądrości i prawdy [28].

## **7.1. Możliwość rozwoju aplikacji**

Aplikacja webowa Leadance ma szerokie perspektywy rozwoju. Przede wszystkim warstwa wizualna jest utworzona w zakresie podstawowym. Mogłaby być ulepszona przy pomocy zaprojektowanych makiet UX. Aplikacja stałaby się wtedy bardziej przyjazna dla użytkownika.

Następną możliwością rozwoju jest wprowadzenie płatności internetowych za zajęcia tańca. Klienci mogliby paroma kliknięciami opłacić swój karnet i mieć te dane na wyciągnięcie ręki w aplikacji. Co więcej, kolejnym krokiem rozwoju mogło by być stworzenie jej odpowiednika w wersji mobilnej dla klientów oraz pracowników i

właścicieli szkół tańca. W obecnych czasach, smartfon stał się mniejszą wersją laptopa lub komputera stacjonarnego. Aplikacja mobilna ułatwiłaby dostęp do danych, a rezerwację z telefonu można by zrobić w kilka sekund.

Do zarządzania szkołą tańca potrzebne są profile pracowników recepcji oraz właściciela szkoły. Wiele szkół tańca oferuje także zajęcia indywidualne u swoich instruktorów tańca. Osobny panel dla instruktora byłby przydatny w procesie rozwoju aplikacji. Taki pracownik mógłby otrzymywać opłaty za zajęcia przez płatności internetowe lub wpisywać swoją dostępność godzinową w określonych dniach, aby przyjąć kursantów. Ponadto, system mógłby zostać rozszerzony o system wbudowany i posiadać własne oprogramowanie do kart członkowskich i czytników użytych do ich czytania. Dzięki nim, obecności nie musiałyby być wpisywane do systemu, a klient sam miałby możliwość zaznaczenia udziału w zajęciach grupowych tańca.

Patrząc na aplikację Leadance z szerokiej perspektywy, ma ona wiele ścieżek i możliwości rozwoju. Nie tylko funkcjonalnych, ale też wizualnych.

## 8. Bibliografia

- [1] L. Allet i in., „Dance therapy combined with patient education improves quality of life of persons with obesity: A pilot feasibility study for a randomised controlled trial”, *Obesity Research & Clinical Practice*, 2017.
- [2] I. Turska, *Krótki zarys historii tańca i baletu*. PWM, 1983.
- [3] M. Tatarczuk, „Klient na rynku usług oferowanych przez szkoły tańca”, *Problemy Zarządzania, Finansów i Marketingu*, 2010.
- [4] „Uczestnictwo w sporcie i rekreacji ruchowej w 2021 r.”, Główny Urząd Statystyczny, 2022.
- [5] Opracowanie zbiorowe, *Czas na biznes. Szkoła tańca*. Colorful Media, 2013.
- [6] „<https://www.bookero.pl/>” (dostęp: 07.11.2022).
- [7] „<https://efitness.pl/>” (dostęp: 07.11.2022).
- [8] „<https://wod.guru/pl/>” (dostęp: 12.01.2023).
- [9] „<https://swiplo.pl/>” (dostęp: 12.01.2023).
- [10] A. D. Scott, *Wszechstronny JavaScript. Technologie: GraphQL, React, React Native i Electron*, Helion, 2020.
- [11] „<https://www.makeuseof.com/windows-10-best-ide/>” (dostęp: 03.02.2023)
- [12] J. Loeliger, M. McCullough, *Kontrola wersji z systemem Git. Narzędzia i techniki programistów.*, Helion, 2014.
- [13] „<https://survey.stackoverflow.co/2022/#most-popular-technologies-language>” (dostęp: 03.02.2023)
- [14] F. Ferreira, M. T. Valente, „Detecting code smells in React-based Web apps”, *Information and Software Technologies*, mar. 2023.
- [15] J. Kyrnin, R. Colburn, L. Lemay, *HTML, CSS i JavaScript dla każdego.*, 2016.
- [16] S. Bradshaw, E. Brazil, K. Chodorow, *Przewodnik po MongoDB. Wydajna i skalowalna baza danych*, Helion.
- [17] „<https://www.mongodb.com/products/compass>” (dostęp: 25.11.2022),



- [18] „<https://mongoosejs.com/>” (dostęp: 25.11.2022).
- [19] „<https://nodejs.org/en/about/>” (dostęp: 25.11.2022)
- [20] „<https://expressjs.com/>” (dostęp: 25.11.2022)
- [21] „<https://hackr.io/blog/best-javascript-frameworks>” (dostęp: 03.02.2023)
- [22] S. Xiong, X. Wang, Z. Lan, „Model Research of Visual Report Components”, *Procedia Computer Science*, sty. 2022.
- [23] P. Pant i in., „Authentication and Authorization in Modern Web Apps for Data Security Using Nodejs and Role of Dark Web”, *Procedia Computer Science*, sty. 2022.
- [24] „<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>” (dostęp: 16.02.2023)
- [25] F. Kaimer, P. Brune, „Return of the JS: Towards a Node.js-Based Software Architecture for Combined CMS/CRM Applications”, *Procedia Computer Science*, sty. 2018.
- [26] R. Miles, K. Hamilton, *UML 2.0. Wprowadzenie*. Helion, 2012.
- [27] S. Roohi, A. Forouzandeh, „Regarding color psychology principles in adventure games to enhance the sense of immersion”, *Entertainment Computing*, maj 2019.
- [28] B. Kilińska, „Wpływ koloru przekazu na jego percepcję”, *Dziennikarstwo i Media*, maj 2018.

## 9. Spis rysunków

Rysunek 1. Panel główny aplikacji webowej Bookero [6] .....	8
Rysunek 2. Panel kalendarza aplikacji webowej eFitness [7] .....	10
Rysunek 3. Panel główny aplikacji webowej WodGuru [8] .....	11
Rysunek 4. Panel główny aplikacji webowej Swiplo.pl [9] .....	13
Rysunek 5. Przykładowa lista do sprawdzania obecności na zajęciach tanecznych w łódzkiej szkole tańca .....	15
Rysunek 6. Przykładowa lista rezerwacji na zajęcia taneczne w łódzkiej szkole tańca .....	15
Rysunek 7. Diagram przypadków użycia UML .....	25
Rysunek 8. Diagram przedstawiający komunikację w MERN Stack,[25] .....	26
Rysunek 9. Diagram komponentów UML dla systemu zarządzania szkołą tańca .....	27
Rysunek 10. Diagram aktywności dla procesu rezerwacji i logowania użytkownika w systemie .....	28
Rysunek 11. Diagram aktywności dla procesu wysyłania wiadomości email przez pracownika/właściciela szkoły tańca .....	28
Rysunek 12. Zrzut ekranu aplikacji webowej Leadance z widokiem użytkownika niezalogowanego. ....	40
Rysunek 13. Formularz rejestracji aplikacji Leadance. ....	41
Rysunek 14. Część panelu rejestracji z ikonami wskazującymi na poprawność danych oraz podpowiedzią w przypadku błędnego wpisania danych .....	41
Rysunek 15. Komunikat po udanej rejestracji. ....	42
Rysunek 16. Panel logowania do aplikacji webowej Leadance .....	42
Rysunek 17. Strona główna dla klienta zalogowanego .....	43
Rysunek 18. Panel rezerwacji zajęć .....	43
Rysunek 19. Lista rozwijana rodzaju opłaty .....	44
Rysunek 20. Lista rozwijana z numerami grup .....	44

Rysunek 21. Widok zakładki "Moje rezerwacje" .....	44
Rysunek 22. Widok zakładki "Grupy" dla zalogowanego klienta.....	45
Rysunek 23. Menu główne dla zalogowanego użytkownika w roli klienta .....	45
Rysunek 24. Strona główna po zalogowaniu się do aplikacji Leadance jako pracownik .....	46
Rysunek 25. Panel rezerwacji dla zalogowanego pracownika .....	47
Rysunek 26. Lista rezerwacji widoczna dla zalogowanego pracownika.....	47
Rysunek 27. Lista grup widoczna dla pracownika .....	48
Rysunek 28. Lista uczestników wraz z panelem do wysyłania wiadomości email .....	48
Rysunek 29. Raport ze statystykami dostępny dla zalogowanego pracownika.....	49
Rysunek 30. Wygenerowana lista telefonów klientów dostępna dla zalogowanego pracownika.....	49
Rysunek 31. Wygenerowana lista emaili klientów dostępna dla zalogowanego pracownika.....	49
Rysunek 32. Panel listy obecności.....	50
Rysunek 33. Panel historii obecności względem daty .....	50
Rysunek 34. Panel historii obecności względem nazwiska klienta .....	51
Rysunek 35. Panel nowej grupy widoczny dla zalogowanego właściciela szkoły tańca.....	52
Rysunek 36. Lista grup widoczna dla zalogowanego właściciela szkoły tańca .....	52

## 10. Spis tabel

Tabela 1. Zalety i wady aplikacji webowej Bookero.....	9
Tabela 2. Zalety i wady aplikacji webowej eFitness .....	10
Tabela 3. Zalety i wady aplikacji webowej WodGuru .....	12
Tabela 4. Zalety i wady aplikacji webowej Swiplo.pl.....	13
Tabela 5. Zalety i wady własnych rozwiązań ankietowanych szkół tańca .....	15
Tabela 6. Przedstawienie struktury kolekcji Users w MongoDB .....	29
Tabela 7. Przedstawienie struktury kolekcji Clients w MongoDB.....	29
Tabela 8. Przedstawienie struktury kolekcji Reservations w MongoDB .....	30
Tabela 9. Przedstawienie struktury kolekcji Groups w MongoDB .....	30
Tabela 10. Przedstawienie struktury kolekcji PresenceLists w MongoDB .....	30

## 11. Spis listingów

Listing 1. Połączenie aplikacji webowej z bazą danych MongoDB .....	32
Listing 2. Schemat zmiennej potrzebnej do połączenia aplikacji webowej z MongoDB .....	32
Listing 3. Model kolekcji MongoDB o nazwie "Clients" .....	33
Listing 4. Transakcja MongoDB z metodą GET pozwalająca odczytać wszystkie dokumenty znajdujące się w kolekcji Clients .....	33
Listing 5. Metoda pozwalająca na rejestrację użytkownika przy użyciu szyfrowania hasła .....	34
Listing 6. Metoda pozwalająca na logowanie użytkownika przy użyciu szyfrowania hasła .....	35
Listing 7. Zmienne oraz metoda pozwalająca na wysłanie wiadomości email poprzez aplikację webową.....	36
Listing 8. Fragment kodu aplikacji webowej odpowiedzialny za routing .....	37
Listing 9. Fragment kodu komponentu z hookiem useState do rejestracji w aplikacji webowej .....	38
Listing 10. Fragment kodu komponentu z hookiem useEffect do rejestracji w aplikacji webowej .....	39
Listing 11. Fragment kodu komponentu z hookiem useRef do rejestracji w aplikacji webowej .....	39