**Uniwersytet Łódzki**

Wydział Matematyki i Informatyki

Jan Włodarski

Nr albumu: 408825

Pt. "Octopus" - mobilna aplikacja wspomagająca zarządzanie grafikiem Klubu Sztuk Walki przygotowana w technologii Android Studio

Praca licencjacka

na kierunku Informatyka

Praca wykonana pod kierunkiem

dr Roberta Plebaniaka

Katedra Analizy Nieliniowej

Łódź, 2025

Spis treści

[1. Wstęp 3](#_Toc200308305)

[2. Wykorzystane technologie 4](#_Toc200308306)

[2.1 Firebase Authentication 5](#_Toc200308307)

[2.2 Firebase Realtime Database 5](#_Toc200308308)

[2.3 Firebase Storage 5](#_Toc200308309)

[3. Założenia funkcyjne aplikacji 6](#_Toc200308310)

[3.1 Funkcjonalności ogólnodostępne 6](#_Toc200308311)

[3.1.1 Ekran główny 7](#_Toc200308312)

[3.1.2 Moduły informacyjne 8](#_Toc200308313)

[3.1.3 System rezerwacji 9](#_Toc200308314)

[3.1.4 Ustawienia i personalizacja 9](#_Toc200308315)

[3.1.5 Okno profilowe 10](#_Toc200308316)

[3.2 Funkcjonalności dostępne dla administratora 10](#_Toc200308317)

[3.3 Funkcjonalności dostępne dla trenerów 11](#_Toc200308318)

[4. Napotkane trudności w implementacji 12](#_Toc200308319)

[4.1 Bazy danych oraz trudności z ich implementacją 12](#_Toc200308320)

[4.1.1 Grafik zajęć oraz opisy treningów 13](#_Toc200308321)

[4.1.2 Statystyki dotyczące treningów 16](#_Toc200308322)

[4.2 Motyw oraz języki aplikacji 18](#_Toc200308323)

[4.2.1 Motyw graficzny 18](#_Toc200308324)

[4.2.2 Ustawienia języka aplikacji 20](#_Toc200308325)

[4.3 Zdjęcia oraz zarządzanie zasobami graficznymi 22](#_Toc200308326)

[4.4 Stoper dla trenerów 24](#_Toc200308327)

[5. Opis aplikacji 27](#_Toc200308328)

[5.1 Ekran główny 27](#_Toc200308329)

[5.2 Ekran logowania 29](#_Toc200308330)

[5.3 Funkcjonalności ogólne 31](#_Toc200308331)

[5.3.1 Grafik zajęć 32](#_Toc200308332)

[5.3.2 Cennik 34](#_Toc200308333)

[5.3.3 Rezerwacje 35](#_Toc200308334)

[5.3.4 Panel treningów personalnych 36](#_Toc200308335)

[5.3.5 Ustawienia 37](#_Toc200308336)

[5.4 Funkcjonalności z podziałem na rolę 38](#_Toc200308337)

[5.4.1 Użytkownik 38](#_Toc200308338)

[5.4.2 Trener 39](#_Toc200308339)

[5.4.3 Administrator 40](#_Toc200308340)

[6. Możliwe kierunki rozwoju aplikacji 42](#_Toc200308341)

[6.1 Panel Aktualności 42](#_Toc200308342)

[6.2 System powiadomień systemowych 43](#_Toc200308343)

[7. Bibliografia 44](#_Toc200308344)

# **Wstęp**

W czasach, gdy sport, a w szczególności sztuki walk staje się coraz popularniejszy, zarówno wśród młodzieży jak i starszej społeczności, a technologia mobilna oraz użytkowanie telefonów komórkowych rozwija się w zastraszającym tempie, aplikacje wspierające codzienne funkcjonowanie stają się nieodłącznym elementem naszego życia. Coraz więcej organizacji – w tym kluby sportowe – dostrzega potrzebę wprowadzania nowoczesnych narzędzi technologicznych, które nie tylko usprawniają komunikację z klientami, ale poprawiają też efektywność zarządzania.

Mobilne aplikacje organizacji, czy klubów sportowych mają na celu odpowiedzieć na zapotrzebowanie użytkowników na szybki dostęp do informacji czy wygodne rezerwacje usług, a jednocześnie pozwalają właścicielom i trenerom lepiej koordynować codzienne działania klubu.

Celem niniejszej pracy jest przedstawienie procesu projektowania oraz implementacji mobilnej aplikacji dedykowanej wsparciu działalności klubu sztuk walki Octopus Łódź, ze szczególnym uwzględnieniem aspektów użytkowych, technologicznych oraz organizacyjnych.

Aplikacja została zaprojektowana z uwzględnieniem potrzeb trzech głównych grup: właścicieli, trenerów oraz uczestników zajęć. Jej podstawowym zadaniem jest ułatwienie dostępu do kluczowych informacji oraz usług oferowanych przez klub, takich jak plan treningów grupowych, dostępność sprzętu, akcesoriów czy odzieży w magazynie klubowym oraz możliwość rezerwacji indywidualnych treningów z ulubionym trenerem. Dzięki temu aplikacja nie tylko zwiększa komfort korzystania z oferty klubu, ale również w sposób pozytywny wpływa na poziom zaangażowania i satysfakcji jego członków.

Opracowany projekt oferuje szeroki zakres funkcjonalności, będący odpowiedni zarówno na potrzeby nowych użytkowników, jak i stałych klientów klubu.

Wśród dostępnych usług znalazły się m.in. przeglądanie grafiku zajęć, zapoznawanie się z opisami poszczególnych treningów, rezerwacja personalnych spotkań z trenerami, a także rezerwowanie akcesoriów, sprzętu i odzieży sportowej dostępnych w magazynie klubowym.

Aplikacja umożliwia również przeglądanie cennika karnetów i kart sportowych, zarządzanie danymi profilowymi, konfigurację motywu, kontakt z administracją klubu za pośrednictwem wbudowanego okna kontaktowego oraz wyszukanie odpowiedzi na najczęściej zadawane pytania dotyczące klubu.

Oprócz funkcji skierowanych bezpośrednio do użytkowników, projekt oferuje również wsparcie dla procesów administracyjnych. Aplikacja pozwala na gromadzenie danych statystycznych dotyczących frekwencji na zajęciach oraz popularności poszczególnych produktów klubowych. Informacje te mogą być wykorzystane przez właścicieli do podejmowania bardziej świadomych decyzji biznesowych oraz organizacyjnych, m.in. w zakresie zamówień sprzętu, planowania oferty czy organizacji kadry trenerskiej oraz grafiku zajęć.

Dzięki wykorzystaniu nowoczesnych technologii oraz dostosowaniu funkcjonalności do realnych potrzeb użytkowników, opracowana aplikacja stanowi kompleksowe i intuicyjne narzędzie wspierające codzienną działalność klubu sportowego. Jej wdrożenie może znacząco przyczynić się do poprawy jakości obsługi klienta, zwiększenia efektywności zarządzania oraz budowania pozytywnego wizerunku klubu.

# **2. Wykorzystane technologie**

Cała aplikacja została stworzona przy wykorzystaniu narzędzia do pracy przy aplikacjach mobilnych jakim jest Android Studio. Jest to darmowe, oficjalne środowisko do kreowania aplikacji na urządzenia mobilne z systemem Android, obsługujące języki Java, Kotlin oraz C++. Oferuje ono narzędzia do projektowania interfejsu użytkownika, elastyczny system kompilacji, emulator Androida oraz inne funkcje wspierające rozwój aplikacji mobilnych, takie jak edytor kodu z zaawansowanym podpowiadaniem składni, system zarządzania zależnościami oparty o Gradle czy możliwość debugowania aplikacji na fizycznym urządzeniu. Całość, zarówno po stronie programistycznej, jak i wizualnej, była tworzona przy pomocy tego środowiska.

W projekcie zastosowany został język Kotlin, który obecnie jest najczęściej wykorzystywanym oraz zalecanym językiem do tworzenia aplikacji Android. Charakteryzuje się on nowoczesną składnią oraz pełną współpracą z językiem Java, co pozwala na integrację z istniejącymi już bibliotekami. Dodatkowo jest on przyjazny dla użytkownika – cechuje się prostotą, dzięki czemu nawet początkujący programiści są w stanie zrozumieć strukturę i działanie kodu napisanego w tym języku.

Aplikacja korzysta z usług platformy Firebase, która pełni funkcję wsparcia technicznego dostarczającego funkcjonalności serwerowe bez konieczności samodzielnego tworzenia infrastruktury bazodanowej. W projekcie wykorzystane zostały trzy główne fragmenty Firebase:

## **2.1 Firebase Authentication**

Moduł ten odpowiada za zarządzanie procesem rejestracji oraz logowania użytkowników. Uwierzytelnianie odbywa się za pomocą adresu e-mail oraz hasła, co zapewnia prosty oraz bezpieczny dostęp do aplikacji. Funkcja pozwala także na obsługę sesji użytkownika, zarządzanie kontami, czy przywracaniem dostępu do odpowiednich kont. Została ona wykorzystana przy logowaniu oraz rejestracji w aplikacji, a unikalne ID użytkownika tworzone przy powstawaniu konta wykorzystywane jest wielokrotnie w różnych operacjach na użytkownikach w aplikacji, takich jak zapisywanie rezerwacji na przedmiot, czy systemie powiadomień.

## **2.2 Firebase Realtime Database**

Ten moduł wykorzystywany jest jako główna baza danych aplikacji. Jest to nierelacyjna baza danych, w której dane przechowywane są w formacie JSON i synchronizowane w czasie rzeczywistym z klientami. Dzięki temu w aplikacji możliwe jest szybkie odczytywanie oraz zapisywanie istotnych informacji o użytkownikach, treningach, rezerwacjach, czy dostępności przedmiotów.

Automatyczna synchronizacja Firebase Realtime Database sprawia, że aplikacja od razu reaguje na zmiany w bazie, co pozytywnie wpływa na percepcję informacji przez użytkownika oraz eliminuje konieczność długiego oczekiwania na ich wyświetlenie.

## **2.3 Firebase Storage**

Funkcja wykorzystywana jest do przechowywania plików multimedialnych, takich jak zdjęcia, dokumenty, filmy oraz inne zasoby przesyłane przez użytkownika. Pozwala ona na łatwe i bezpieczne przechowywanie oraz udostępnianie plików z poziomu aplikacji mobilnej. W projekcie służy przede wszystkim do przechowywania zdjęć użytkowników (w tym trenerów) lub materiałów związanych z rezerwowanymi przedmiotami, czy opisem zajęć.

Firebase, jako kompleksowa platforma wspierająca rozwój aplikacji mobilnych, zapewnia również wysoki poziom bezpieczeństwa oraz integracji z innymi usługami jakich możemy doświadczyć w środowisku programistycznym, dlatego czyni ją to skutecznym rozwiązaniem zarówno dla prostych jak i bardziej zaawansowanych projektów. Wszystkie elementy aplikacji, począwszy od logiki biznesowej, przez warstwę wizualną, aż po system zarządzania danymi, zostały zaimplementowane przy użyciu opisanych powyżej technologii. Zastosowanie takich narzędzi programistycznych jak Android Studio, czy Firebase, pozwala na stworzenie miłej dla oka oraz wydajnej aplikacji mobilnej, spełniającej założone cele projektu.

# **3. Założenia funkcyjne aplikacji**

Projekt aplikacji w założeniu ma być skierowany do trzech głównych grup użytkowników: administratorów, trenerów oraz zwykłych członków klubu. Administratorzy powinni także mieć możliwość zmiany roli oraz nadawania jej wszystkim innym użytkownikom. Sama idea roli ma natomiast być skierowana jedynie do użytkowników zalogowanych.

Każda z tych grup powinna posiadać dostęp do odpowiednio dostosowanych dla nich funkcjonalności, co pozwaliłoby na lepsze dopasowanie interfejsu oraz zakresu dostępnych działań dla kompetencji danej roli użytkownika. Takie podejście umożliwia efektywniejsze zarządzanie zasobami klubu, organizację pracy trenerów oraz zapewnienie każdej z roli intuicyjnego, spersonalizowanego dostępu do informacji i usług.

## **3.1 Funkcjonalności ogólnodostępne**

Aplikacja w swojej podstawowej formie ma za zadanie oferować szeroki zakres funkcji dostępnych dla wszystkich osób z niej korzystających.

Celem głównym musi być zapewnienie szybkiego i przejrzystego dostępu do najważniejszych informacji związanych z funkcjonowaniem klubu, takich jak:

* najnowszy grafik zajęć,
* szczegółowe opisy konkretnych treningów oraz trenerów zajmujących się poszczególnymi grupami i prowadzących treningi personalne,
* panel odpowiedzi na najczęściej zadawane pytania, który umożliwiałby zaspokojenie potrzeb informacyjnych użytkownika bez konieczności angażowania pracowników klubu oraz bez potrzeby podejmowania przez użytkownika dodatkowych działań w celu nawiązania kontaktu.

Realizacja tych funkcjonalności miałaby na celu zminimalizowanie potrzeby bezpośrednie kontaktu osobistego czy telefonicznego z pracownikami klubu i zwiększenie samodzielności użytkownika co przekłada się na oszczędność czasu i wygodę, zarówno dla członków klubu, jak i jego personelu.

### **3.1.1 Ekran główny**

Ekran główny musi zapewniać prezentowanie nadchodzących zajęć, co pozwala użytkownikowi szybko zorientować się w bieżącym harmonogramie. Dodatkowo powinien zawierać on łatwy w obsłudze oraz intuicyjny system nawigacji pomiędzy interesującymi użytkownika kartami. Przyciski szybkiego dostępu do kluczowych sekcji aplikacji, takich jak: panel rezerwacji, trenerów personalnych czy cennik, powinny pomóc przy poruszaniu się po aplikacji. W swoich założeniach aplikacja powinna zawierać:

* **dolny pasek nawigacyjny -** umożliwia szybki dostęp do trzech głównych sekcji: Strony głównej, grafiku oraz ustawień aplikacji;
* **ikona powiadomień -** na ekranie głównym przenosi użytkownika do osobnego ekranu, gdzie wyświetlane są aktualne komunikaty, ogłoszenia oraz informacje;
* **menu boczne -** zawiera odnośniki do wszystkich istotnych sekcji aplikacji, w tym:
  + Grafik zajęć,
  + Cennik,
  + Rezerwacje,
  + Nasi trenerzy,
  + Pomoc i kontakt,
  + Ustawienia,
  + Logowanie (dla użytkowników niezalogowanych),
  + Wylogowanie (dla użytkowników zalogowanych).

### 3.1.2 Moduły informacyjne

Najważniejszym aspektem funkcjonalnym aplikacji jest prezentacja wszystkich informacji w sposób przejrzysty i łatwo dostępny. Można to umożliwić poprzez karty takie jak:

* **Grafik zajęć** – zawiera szczegółowy harmonogram treningów dostępnych w danym dniu. Użytkownik może w łatwy sposób sprawdzić godzinę rozpoczęcia zajęć, przypisaną grupę treningową oraz zapoznać się z opisem danego treningu. Opis ten obejmuje informacje dotyczące charakteru zajęć, poziomu ich zaawansowania, osoby prowadzącej oraz czasu trwania treningu
* **Cennik –** prezentowany jest w formie graficznej. Poza cenami karnetów dostępnych w klubie, zawiera również informacje o honorowanych kartach sportowych, takich jak Multisport, czy Medicover Sport. W tym module znajduje się również okno kontaktowe z podstawowymi danymi: numerem telefonu, adresem e-mail oraz odnośnikami do mediów społecznościowych.
* **Nasi trenerzy** – sekcja ta zawiera pełną listę trenerów oferujących treningi personalne w klubie, wraz z opcją filtrowania według grupy zajęciowej i poziomu zaawansowania. Po kliknięciu w profil trenera wyświetlany jest jego opis, zdjęcie oraz dane kontaktowe, a następnie, jeśli wybierzemy odpowiednią datę gdzie dostępne są treningi personalne z danym trenerem, możliwość rezerwacji takiego treningu.
* **Pomoc i kontakt** – moduł ten zawiera zbiór najczęściej zadawanych pytań w formie FAQ, które mają na celu rozwianie wątpliwości użytkowników bez konieczności kontaktu w formie fizycznej czy telefonicznej z obsługą klubu. Znajduje się tu również okno kontaktowe analogiczne do tego w sekcji Cennik.

Ważnym modułem dla aplikacji powinien stanowić też system rezerwacji:

### **3.1.3 System rezerwacji**

Strona ta ma za zadanie umożliwiać użytkownikom dostęp do asortymentu dostępnego w magazynie klubowym oraz, dla zalogowanych użytkowników, również rezerwację przedmiotów z tego zbioru. Powinien dzielić się on na dwa ekrany:

* **Sprawdź dostępność** – umożliwia przegląd i rezerwację sprzętu. Funkcja rezerwacji dostępna wyłącznie dla zalogowanych użytkowników.
* **Moje rezerwacje** – prezentuje historię oraz bieżący status rezerwacji dokonanych przez użytkownika.

Istotnym elementem aplikacji jest możliwość zarządzania jej wyglądem oraz sposobem działania, co umożliwia użytkownikowi dostosowanie interfejsu i funkcjonalności do własnych preferencji.

### **3.1.4 Ustawienia i personalizacja**

Moduł ten odnosi się do ustawień, które mają za zadanie umożliwić użytkownikowi dostosowanie aplikację do osobistych preferencji, takich jak:

* wybór motywu aplikacji (jasny/ciemny bądź domyślny), pozwala dostosować interfejs do trybu najbardziej odpowiadającego użytkownikowi,
* zmiana języka interfejsu (polski, angielski, ukraiński), pozwala na dostosowanie aplikacji dla ludzi różnej narodowości, którzy bardziej komfortowo czują się używając aplikacji w swoim rodzimym języku. Wybór języków jest uargumentowany ilością oraz różnorodnością narodowościową osób uczęszczających do klubu Octopus Łódź oraz językami którymi się posługują.

W celu zwiększenia komfortu korzystania z aplikacji oraz pogłębienia zaangażowania użytkowników, aplikacja daje możliwość modyfikacji oraz dodawania danych osób korzystających z niej w oknie profilowym:

### **3.1.5 Okno profilowe**

Funkcja ta, dostępna jedynie dla użytkowników po zalogowaniu, ma na celu umożliwienie ingerencji w dane użytkownika podane przy rejestracji, dodać zdjęcie profilowe czy dodatkowe informacje, jak numer telefonu, czy nazwisko, które będą niezbędne przy różnych procesach, takich jak rezerwacje asortymentu czy treningów personalnych. Dodatkowo, zakłada ona posiadanie listy rezerwacji użytkownika, dzięki czemu może on na bieżąco śledzić status zarezerwowanych przedmiotów oraz szczegóły tych rezerwacji.

## **3.2 Funkcjonalności dostępne dla administratora**

Aplikacja zapewnia administratorowi dostęp do edycji ról użytkowników, opisów zajęć, dodawania trenera oraz godzin dla trenera personalnego, dodawania przedmiotów do magazynu klubowego w aplikacji oraz możliwość akceptacji prośby o rezerwację przedmiotu, a ponadto mogą oni również korzystać z zaawansowanych funkcji zarządzania klubem. Są to:

* **Ekran statystyk** – zawiera dane dotyczące frekwencji na zajęciach, rezerwowanych przedmiotów bądź zarejestrowanych w aplikacji użytkowników, przedstawiane w postaci wykresów. Pozwoli to analizować popularność poszczególnych zajęć w określonych przedziałach czasowych.
* **Kalendarz klubowy** – umożliwia zapisywanie informacji dotyczących prowadzonych danego dnia treningów poprzez spisywanie takich danych jak:
  + Kto prowadził trening (opcjonalnie, czy wystąpiło zastąpienie trenera innym),
  + Ile osób brało udział w treningu,
  + Dodatkowa, opcjonalna notatka na temat danego dnia.

Zastosowanie takiego rozwiązania pozwoli zarówno pracownikom, jak i właścicielom klubu na efektywniejsze zarządzanie czasem oraz przestrzenią treningową. Umożliwia to m.in. optymalizację grafiku zajęć poprzez zwiększenie liczby treningów cieszących się większym zainteresowaniem oraz eliminację tych, które charakteryzują się niską frekwencją.

## **3.3 Funkcjonalności dostępne dla trenerów**

Trenerzy klubu, poza funkcjami ogólnodostępnymi, mają również zapewniony dostęp do specjalnego panelu, który pozwala na modyfikowanie danych wyświetlanych o nich dla użytkowników:

* **Dla trenerów** – ekran ten zawiera dane dotyczące trenera, takie jak zdjęcie, kontakt, czy opis trenera, które można modyfikować. Dodatkowo, istnieje możliwość dodawania pasujących wolnych godzin treningów personalnych dla danej daty, które następnie mogą ukazać się użytkownikom aplikacji. Panel ten wyposażony jest także w specjalny stoper:
  + **Stoper treningowy**  - funkcjonalność pomagająca trenerom w kontrolowaniu długości ćwiczeń. Dzięki stoperowi istnieje możliwość sterowania ilością rund treningowych, czy też ich długością, a także czasem trwania przerw. Zaimplementowanie stopera bezpośrednio w aplikacji ma na celu zwiększenie motywacji trenerów do regularnego korzystania z systemu. Integracja tej funkcjonalności eliminuje potrzebę stosowania zewnętrznych narzędzi, co umożliwia prowadzenie zajęć z wykorzystaniem jednej, kompleksowej aplikacji.

Aplikacja umożliwia również użytkownikom składanie próśb o treningi personalne z wybranym trenerem. Po złożeniu rezerwacji, system generuje powiadomienie zawierające szczegóły dotyczące proponowanej daty i godziny treningu, gdzie trafia do panelu trenera. Trener ma możliwość zatwierdzenia bądź odrzucenia zgłoszenia, z opcjonalnym podaniem przyczyny swojej decyzji. Takie rozwiązanie usprawnia komunikację oraz proces organizacji treningów indywidualnych.

# **4. Napotkane trudności w implementacji**

Proces tworzenia aplikacji mobilnej obejmował szereg zróżnicowanych funkcjonalności, mających na celu zapewnienie intuicyjnej obsługi, łatwego dostępu do informacji oraz sprawnej organizacji rezerwacji.

Pomimo ostatecznej realizacji wszystkich zaplanowanych funkcji, proces implementacji nie był pozbawiony wyzwań. Główne trudności napotkane podczas pracy nad aplikacją dotyczyły zarówno aspektów technicznych, takich jak integracja z bazą danych czy logika zarządzania dostępem w zależności od roli użytkownika, jak i problemów związanych z ergonomią interfejsu, synchronizacją rezerwacji oraz zapewnieniem niezawodnego systemu powiadomień.

Jednym z kluczowych wyzwań okazała się integracja oraz odpowiednie zarządzanie bazą danych.

## **4.1 Bazy danych oraz trudności z ich implementacją**

Projekt, do większości danych przechowywanych oraz odczytywanych w aplikacji, zakłada wykorzystanie chmurowej bazy danych – Firebase Realtime Database – ze względu na elastyczność tego rozwiązania oraz możliwości łatwej synchronizacji danych w czasie rzeczywistym.

Implementacja poprawnej i wydajnej komunikacji między interfejsem użytkownika a strukturą w bazie danych wymagała wielu poprawek i testów.

Pierwszym z problemów przy wprowadzeniu bazy danych do aplikacji okazało się odpowiednie połączenie różnych komponentów związanych z harmonogramem, tak aby jednocześnie spełniały one wymagania funkcjonalne, nie prowadziły do nadmiarowości danych w bazie oraz zapełniały łatwą skalowalność do działań w przyszłości.

### **4.1.1 Grafik zajęć oraz opisy treningów**

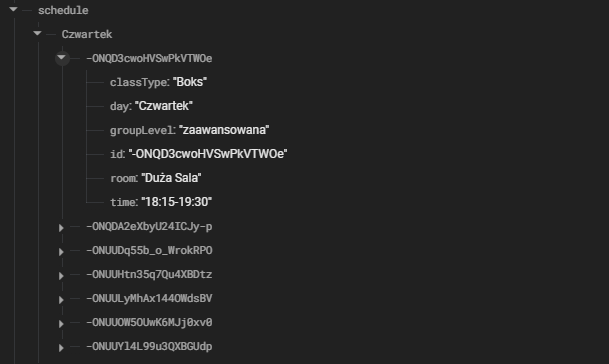
W aplikacji przewidziano dwa główne widoki grafiku – dynamiczny, wyświetlający najbliższe zajęcia na ekranie głównym, oraz drugi – statyczny, znajdujący się w oddzielnym ekranie.

Synchronizacja tych dwóch widoków oraz szczegółów ich dotyczących wymagała zaprojektowania odpowiedniej struktury danych, która umożliwiłaby łatwe filtrowanie zajęć według dnia, jednocześnie nie prowadząc do niepotrzebnego powielania informacji.

Próba takiego rozwiązania zarządzaniem bazą danych stworzyła problem, który szczególnie uwidocznił się w kontekście przechowywania opisów zajęć. Zajęcia tego samego typu, prowadzone o tej samej godzinie w różnych dniach tygodnia, posiadały identyczne opisy. Gdyby każdy wpis w grafiku zawierał własną kopię opisu, skutkowałoby to znaczącą redundancją danych. Aby tego uniknąć, oddzielona została warstwa opisowa od głównego drzewa danych zajęć.

Opisy zostały zorganizowane jako opisy w oddzielnej strukturze bazy danych, możliwe do edytowania niezależnie od właściwego harmonogramu.

Finalnie, wartości dotyczące godzin treningów wprowadzone zostały z dwoma różnymi zastosowaniami:

* Statyczny rzadko ulegający zmianom grafik zajęć – dotyczy on grafiku wyświetlanego na ekranie głównym, który obrazuje najbliższe treningi które odbywają się bądź odbędą w klubie, a także na tym z oddzielnym panelem, gdzie możemy znaleźć każde z zajęć, przewijając między dniami. Dane te znajdują się w bazie danych w drzewie „schedule” o gałęziach dzień\_tygodnia (przykładowo: Poniedziałek)/id\_zajęć (losowane jako id), gdzie znaleźć możemy podstawowe dane na temat każdego z zajęć:

Rysunek 1. Drzewo **schedule** w bazie danych Firebase Database

Opisy dla poszczególnych zajęć, tak jak wcześniej zostało to przedstawione, zostały zaimplementowane w oddzielnym drzewie, ze względu na łatwiejsze wprowadzanie i edytowanie tych danych oraz brak powielania opisów zajęć, gdy te same treningi odbywają się o tej samej godzinie, ale w innych dniach:

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 2. Drzewo **classDescription** w bazie danych Firebase Database

Takie rozwiązanie usprawniło zarządzanie treścią oraz uprościło proces aktualizacji informacji – zmiana opisu zajęć wymaga modyfikacji tylko jednego wpisu, bez konieczności ingerencji w wiele instancji rozproszonych po kalendarzu. Dodatkowym utrudnieniem była integracja grafiku ze statystykami:

### **4.1.2 Statystyki dotyczące treningów**

Statystyki mają za zadanie obrazować dane dotyczące frekwencji oraz ogólne informacje, przydatne dla właścicieli klubów, a uzupełniane przez pracowników, dotyczące zajęć danego dnia.

Najlepszym sposobem, aby znaleźć i automatycznie uzupełniać statyczne dane dotyczące zajęć w aplikacji, tak, aby pracownicy nie musieli każdej informacji wpisywać ręcznie, było wykorzystanie danych które baza już posiada w drzewie „schedule”. W tym przypadku jednak nie można było opierać się na automatycznie generowanych identyfikatorach wpisów, ponieważ każdorazowe dodanie lub usunięcie zajęć powodowało ich zmianę, co utrudniałoby ich identyfikację w sposób jednoznaczny. Zamiast tego, jako klucz do agregowania danych statystycznych, wykorzystano zestaw atrybutów takich jak nazwa zajęć, poziom trudności, dzień tygodnia oraz godziny, w których zajęcia się odbywają. Choć rozwiązanie to wymagało większej ostrożności przy wprowadzaniu danych, zapewnia większą trwałość oraz odporność na zmiany:

Obraz zawierający tekst, zrzut ekranu

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 3. Drzewo ScheduleStatistics w bazie danych Firebase Database

Tutaj każda z danych odpowiada za poszczególne funkcje w aplikacji, gdzie dzięki wartościom dayOfWeek, classType oraz groupLevel razem z kluczem który tworzony jest za pomocą godziny treningów, możemy zidentyfikować odpowiednie treningi i przypisywać do nich wartości, takie jak liczba uczestników, trener, który prowadził zajęcia oraz czy odebrał on wynagrodzenie za poprowadzony trening.

Cała ta struktura przypominać ma podręczny kalendarz, który wykorzystywany jest przez pracowników klubu do zapisywania danych dotyczących zajęć. Intencją przy projektowaniu tego komponentu było odwzorowanie tego komponentu w wersji cyfrowej, z myślą o ułatwieniu procesu rejestrowania danych w formie mobilnej.



Rysunek 4. Kalendarz dla pracowników oraz właścicieli klubu

To podejście bazodanowe oraz implementacyjne eliminuje konieczność prowadzenia dokumentacji papierowej oraz zwiększa wygodę użytkowania, zapewniając szybki dostęp do danych z poziomu telefonu bądź tabletu.

## **4.2 Motyw oraz języki aplikacji**

Kolejną z niewygodnych do implementacji funkcjonalności okazała się zmiana języka interfejsu użytkownika oraz motywu graficznego aplikacji.

Choć z pozoru są to funkcjonalności o charakterze estetyczno-użytecznym, ich prawidłowe zaimplementowanie w środowisku mobilnym wymagało zastosowania odpowiednich wzorców projektowych oraz zrozumienia mechanizmów działania systemu Android w kontekście zarządzania zasobami i stanem aplikacji.

Zaimplementowanie tych funkcjonalności uwzględniało wiele aspektów technicznych, takich jak dynamiczne ładowanie zasobów, przechowywanie preferencji użytkownika, unikanie błędów związanych z ponownym renderowaniem widoków oraz zapewnienie kompatybilności między różnymi wersjami systemu operacyjnego.

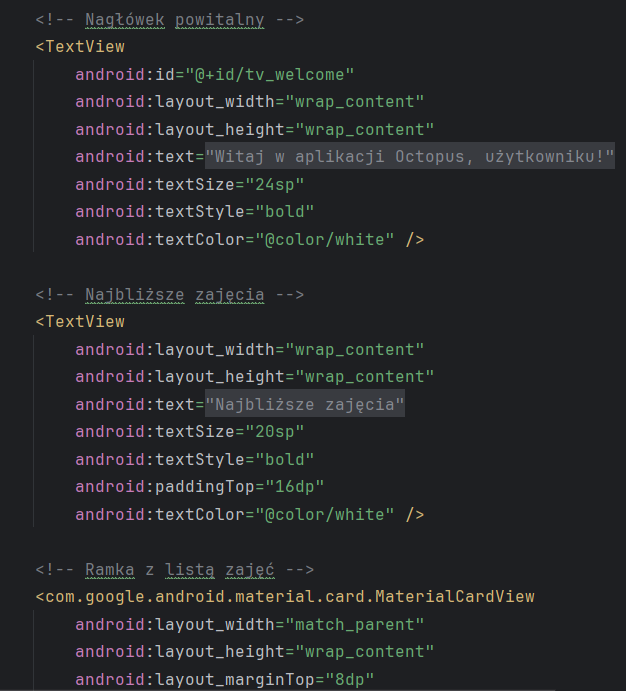
### **4.2.1 Motyw graficzny**

W aplikacji „Octopus” użytkownik ma możliwość wyboru jednego z trzech trybów wyświetlania: jasnego, ciemnego, oraz domyślnego (dopasowanego do ustawień systemowych urządzenia).

Problemem w tym wypadku okazało się dostosowanie trybów tak, aby każda z informacji i funkcjonalności była dla użytkownika czytelna, a interfejs – przyjazny.

Początkowo, aplikacja była tworzona jedynie dla jednego motywu – ciemnego, który był ustawiony domyślnie w emulatorze. Przez to też, po początkowym wprowadzeniu drugiego motywu, projekt był nieczytelny, czcionki były niewidoczne, bądź wtapiały się w tło, a przez co interfejs wydawał się chaotyczny oraz mało przyjazny użytkownikowi.

Niniejszy problem został rozwiązany poprzez dopasowanie kolorów do odpowiednich fragmentów – ponieważ każdy z paneli występuje na tym samym tle z czarno – szarym motywem, najbardziej komfortowym rozwiązaniem okazała się zmiana kodu dot. poszczególnych kolorów czcionek w danych fragmentach (przykładowo, ekranu głównego):



Rysunek 5. Fragment kodu układu graficznego głównego ekranu, przedstawiający ustawienia dotyczące pewnych widoków tekstowych

Dzięki takiemu rozwiązaniu, motywy działają poprawnie, zmieniając strukturę wizualną aplikacji, poprawiając czytelność dostosowaną do preferencji użytkownika, jednocześnie bez wpływu wyglądu czcionki na komfort użytkowania.

Mówiąc o możliwościach aplikacji, które może wykorzystać użytkownik, a które sprawiły kłopot w implementacji oraz użyciu, warto nadmienić również zmianę języka interfejsu:

### **4.2.2 Ustawienia języka aplikacji**

Wprowadzenie innowacyjności, jaką jest wybór preferowanego języka aplikacji, była kluczowa, bowiem do klubu sportowego uczęszczają klienci różnych narodowości, którzy mogą nie czuć się komfortowo użytkując aplikację w języku polskim.

Początkowo wszystkie komunikaty oraz elementy interfejsu były zapisane bezpośrednio w kodzie źródłowym – po polsku – co uniemożliwiało ich dynamiczne tłumaczenie.

Implementacja systemu językowego wymagała zatem przeniesienia wszystkich stałych tekstów do zasobów strings.xml oraz utworzenia osobnych plików dla każdego wspieranego języka – polskiego, ukraińskiego oraz angielskiego, a tak wyglądało to w kodzie na przykładzie języka angielskiego:

Obraz zawierający tekst, zrzut ekranu, menu, Czcionka

Zawartość wygenerowana przez AI może być niepoprawna.

Rysunek 6. Fragment kodu pliku strings.xml, gdzie ręcznie przypisywane są odpowiednie tłumaczenia dla danych słów w języku angielskim

Zadanie było czasochłonne, szczególnie biorąc pod uwagę zachowanie spójności kontekstowej komunikatów we wszystkich wersjach językowych.

Dodatkowym aspektem, o którym należałoby wspomnieć pod względem języka interfejsu, była dynamiczna zmiana języka, bez konieczności ponownego uruchamiania aplikacji. Początkowo za każdym razem po zmianie języka należało zrestartować aplikację, aby każda z wartości tekstowych zaktualizowała się.

Dzięki wykorzystaniu odpowiedniego mechanizmu zarządzania kontekstem oraz odświeżaniu widoków, udało się wprowadzić tą funkcjonalność, a stworzenie oddzielnej klasy zarządzającej wprowadzanym językiem pomogło w optymalizacji aplikacji:

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Zawartość wygenerowana przez AI może być niepoprawna.

Rysunek 7. Kod **App**, klasy odpowiadającej za dostosowywanie języka interfejsu przy ponownym uruchomieniu aplikacji

Wybrany język zapisywany jest w SharedPreferences, a jego ustawienie następuje podczas uruchamiania aplikacji w klasie App, dziedziczącej po Application. W metodzie OnCreate() aplikacja pobiera zapisany kod języka i ustawia odpowiednią lokalizację poprzez metodę setAppLocale(), która aktualizuje konfigurację zasobów zgodnie z wybranym językiem.

Dzięki takiemu podejściu wszystkie teksty interfejsu są ładowane bezpośrednio z odpowiedniego pliku strings.xml, co zapewnia spójne tłumaczenie treści w całej aplikacji.

Po wdrożeniu obsługi wielu języków kolejnym istotnym wyzwaniem podczas tworzenia aplikacji okazało się efektywne zarządzanie zdjęciami oraz zasobami graficznymi.

Obrazy pełnią ważną rolę w interfejsie użytkownika – zarówno w kontekście wizualnej identyfikacji trenerów, jak i uatrakcyjnieniu wyglądu aplikacji.

Ich obsługa wymagała odpowiedniego podejścia zarówno od strony technicznej, jak i organizacyjnej, szczególnie przy integracji z bazą danych oraz zapewnieniu płynnego działania aplikacji na różnego typu urządzeniach.

## **4.3 Zdjęcia oraz zarządzanie zasobami graficznymi**

Zasoby graficzne, przede wszystkim odczyt i zapis tych zasobów w bazie danych Firebase Storage, wykorzystywane są przy profilach, zarówno trenerów jak i użytkowników, a także przy zdjęciach dotyczących zajęć. Trudnością przy wprowadzeniu tej funkcjonalności była optymalizacja dodawania i wczytywania zdjęć, tak, aby były one odpowiednio wyskalowane i dostosowane do preferencji użytkownika.

Zaczynając od grafik odpowiednich zajęć, początkowo były one wprowadzane do aplikacji ręcznie jako zasoby graficzne w folderze drawable, co ograniczało ich personalizację i wymagało aktualizacji wersji aplikacji w przypadku każdej zmiany, a użytkownicy aplikacji nie mogli wprowadzać żadnych zmian – wszystko musiał dodawać administrator aplikacji.

Statyczne dodawanie zdjęć zostało zastąpione wczytywaniem oraz odczytywaniem zdjęć z bazy danych przy pomocy technologii Firebase Storage. Użytkownik może wybrać zdjęcie z pamięci swojego urządzenia, a aplikacja zapisuje je w bazie danych:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 8. Fragment kodu odpowiadającego za dodawanie zdjęcia trenera w szczegółach opisu grupy zajęciowej

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.oraz automatycznie ładuje w odpowiednich miejscach interfejsu:

Rysunek 9. Fragment kodu odpowiadający za załadowywanie zdjęcia w interfejsie szczegółowego opisu grupy zajęciowej

Dodatkowo, do dynamicznego wyświetlania obrazów wykorzystano bibliotekę Glide, która umożliwia szybkie i zoptymalizowane ładowanie zdjęć z chmury, jednocześnie oferując możliwość ich skalowania, a dodatkowa biblioteka uCrop pomaga w zaokrąglaniu i dopasowywaniu do układu w aplikacji. Tak wygląda zastosowanie Glide razem z uCrop w kodzie dotyczącym zdjęcia dla trenera w grafiku zajęć:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 10. Fragment kodu odpowiadający za przycinanie oraz skalowanie zdjęcia poprzez technologię uCrop oraz Glide

Funkcja ta pobiera zdjęcie od użytkownika, a następnie pozwala mu je wykadrować, po czym dodaje je do aplikacji.

Takie rozwiązanie znacząco poprawiło elastyczność zarządzania zdjęciami, estetykę oraz wydajność aplikacji.

Nieco mniejszym wyzwaniem okazała się, wymagająca dokładnej synchronizacji elementów graficznych z czasem – animacja stopera, której opracowanie wiązało się z kilkoma trudnościami implementacyjnymi.

## **4.4 Stoper dla trenerów**

Jednym z wymagających elementów projektu była implementacja animowanego stopera, który miał za zadanie odmierzać czas rund treningowych i przerw, co mogłoby pomóc trenerom w organizowaniu ćwiczeń na zajęciach.

Obraz zawierający tekst, zrzut ekranu, design, wzór

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.Oprócz samego licznika tekstowego, zdecydowano się na zastosowanie animacji, przy pomocy środowiska Lottie, którego animacje pomogły wizualnie zilustrować upływający czas. Główną trudność stanowiła konieczność synchronizacji trzech komponentów – licznika czasu, animacji oraz danych które podaje użytkownik (czas rundy, czas przerwy, liczba rund):

Rysunek 11. Zrzut ekranu funkcjonalności stopera treningowego

Początkowo, animacja nie działała poprawnie – błędy w synchronizacji skutkowały „rozjeżdżaniem” się animacji z realnym stanem licznika. Dodatkowo konieczne było też zarządzanie stanami pauzy, resetu oraz przejścia między rundami i przerwami.

Problematyczne okazało się to, że CountDownTimer – funkcjonalność użyta od odmierzania czasu – operuje na interwałach sekundowych (co 1000ms), natomiast ValueAnimator – funkcja która operuje animacją – potrzebuje dokładnego czasu trwania (w milisekundach), aby odtwarzać animację płynnie. Każdy błąd w synchronizacji, jak przykładowo wznawianie czasu po pauzie lub zmianie etapu z rundy na przerwę, skutkował „rozjechaniem” się animacji z realnym stanem licznika.

Dopiero precyzyjne odtworzenie animacji od momentu zatrzymania i odpowiednie zarządzanie jej czasem trwania, dzięki flagom dotyczącym startu, pauzy czy wznowieniu stopera, pozwoliło na osiągnięcie pożądanego efektu – płynnego i spójnego działania animacji z rzeczywistym odliczaniem czasu.

Każda sesja (rundy jak i przerwy) uruchamia nowy ValueAnimator, który stopniowo aktualizuje właściwość postępu (progres) widoku animacji (LottieAnimationView) w zakresie od 0 do 1 w notacji zmiennoprzecinkowej (float). Czas trwania animacji jest równy całkowitemu czasowi trwania sesji (totalTime), wyrażonemu w milisekundach:

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 12. Fragment kodu dotyczący stopera treningowego, odpowiadający za rozpoczęcie animacji

Dzięki temu animacja przebiega płynnie i kończy się dokładnie w momencie zakończenia odliczania.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.W przypadku pauzy aplikacja zapisuje aktualny postęp animacji w zmiennej pausedAnimationProgress oraz czas, jaki pozostał do końca (pausedTimeLeft): Po wznowieniu animacja odtwarzana jest od miejsca zatrzymania z odpowiednim czasem trwania:

Rysunek 13. Fragment kodu dotyczący stopera treningowego, odpowiadający za wznowienie animacji

Dzięki takiej operacji, użytkownik ma wrażenie ciągłości, nawet jeśli stoper został wznowiony po przerwie.

Dodatkowo, każda zmiana etapu powoduje zatrzymanie dotychczasowej animacji oraz rozpoczęcie nowej z odpowiednim czasem i zakresem.

Mechanizm ten działa analogicznie przy rozpoczęciu rundy jak i zapauzowaniu stopera:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 14. Fragment kodu dotyczący stopera treningowego, odpowiadający za rozpoczęcie rundy

Podsumowując, synchronizacja animacji z logiką czasu została osiągnięta. Umożliwiło to między innymi odpowiednie sterowanie dzięki wartości ValueAnimator oraz przez zapisywanie stanu pauzy i wznowienia. Takie rozwiązanie zapewnia zarówno estetykę działania animacji jak i pełną funkcjonalność stopera z możliwościami takimi jak pauza, czy reset czasu.

# **5. Opis aplikacji**

Mimo wszystkich niedogodności, jakie zostały napotkane podczas tworzenia aplikacji, założenia jej budowy zostały spełnione. W wyniku regularnych konsultacji z trenerami, pracownikami oraz właścicielami klubu, udało się dokładnie określić wymagania funkcjonalne aplikacji.

Na podstawie tych ustaleń zaimplementowano wszystkie kluczowe funkcjonalności, które były przewidziane na etapie planowania projektu. W niniejszym rozdziale przedstawiono szczegółowy opis aplikacji, z uwzględnieniem jej struktury, działania oraz poszczególnych modułów odpowiadających za realizację zadań, które zostały wcześniej zdefiniowane.

Kluczowym elementem interfejsu jest ekran główny, który pełni rolę punktu wyjścia do dalszej pracy z aplikacją. Od niego rozpoczyna się interakcja użytkownika z aplikacją, nawigacja pomiędzy poszczególnymi ekranami oraz podgląd najbliższych zajęć, które pomagają zorientować się użytkownikowi w treningach które odbędą się najbliżej czasu rzeczywistego:

## **5.1 Ekran główny**

Obraz zawierający tekst, zrzut ekranu, design

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 15. Zrzut ekranu, obrazujący wygląd ekranu głównego aplikacji

Głównym oknem które ma za zadanie przykuwać uwagę użytkownika jest panel najbliższych zajęć. Został on zaprojektowany tak, aby pokazywać najbardziej zbliżone do czasu rzeczywistego treningi. Jak widać na załączonym obrazku, najbliższe treningi to wszystkie te, które odbywają się w poniedziałek, ponieważ aplikacja odpalona jest w sobotę, gdy już wszystkie treningi się skończyły, a w niedziele nie ma żadnych zajęć.

Ekran główny oferuje również przyciski szybkiego dostępu, które skierują użytkownika w ekrany które, według rozmów z potencjalnymi klientami oraz pracownikami klubu, mogą być najczęściej odwiedzane.

Na ekranie widnieje również menu dolne oraz boczne – oba te rozwiązania mają ułatwić nawigację pomiędzy poszczególnymi oknami. Przykładowo, tak wygląda menu boczne, złożone z przycisków, po otwarciu:

Obraz zawierający tekst, zrzut ekranu, design

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 16. Zrzut ekranu, obrazujący widok panelu bocznego służącego do nawigacji między fragmentami

Dla niezalogowanego użytkownika, widnieje nagłówek „Witaj, gościu!”, natomiast po kliknięciu przycisków przeniosą nas one do wybranych ekranów.

Pierwszym z nich jest „Zaloguj się”, który pozwala niezalogowanemu użytkownikowi połączyć swoje konto z aplikacją, co otwiera szereg nowych możliwości:

## **5.2 Ekran logowania**

Obraz zawierający tekst, zrzut ekranu, design

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 17. Zrzut ekranu przedstawiający panel logowania

Panel logowania został stworzony specjalnie z myślą o dostosowywaniu pewnych funkcjonalności w zależności od roli użytkownika, która dopasowywana jest po utworzeniu konta i zalogowaniu. Przycisk przypominający logo androida pomaga dostać się do konta administracyjnego bez konieczności ręcznego wpisywania danych, jest on dostępny jedynie dla osoby programującej aplikację.

Logowanie odbywa się poprzez Firebase Authenticatication, które zostało wybrane ze względu na bezpieczeństwo, prostotę w integracji danych oraz współpracę z Android Studio – środowiskiem w którym aplikacja jest tworzona. W przypadku niniejszego projektu zdecydowano że potrzebnymi danymi są e-mail oraz hasło, które zakłada się przy rejestracji, do której możemy dostać się dzięki specjalnemu odnośnikowi poniżej przycisku „Zaloguj się!”:

Obraz zawierający tekst, zrzut ekranu

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 18. Zrzut ekranu przedstawiający panel rejestracji

Do poprawnego zarejestrowania użytkownika należy podać imię, e-mail, nazwę użytkownika oraz hasło. Dodatkowe pole, służące do potwierdzenia adresu e-mail, niweluje możliwość pomyłki użytkownika podczas wpisywania tej kluczowej do rejestracji danej.

Po zarejestrowaniu i zalogowaniu użytkownika, panel boczny powiększa swoje możliwości w zależności od roli jaką użytkownik posiada.

Sama aplikacja została stworzona z myślą o dostosowaniu jej dla trzech głównych grup odbiorców:

* Użytkowników – klientów klubu – „user”
* Trenerów – „trainer”
* Administratorów aplikacji bądź właścicieli klubu – „admin”

Rejestrując się jako nowy użytkownik, rola automatycznie zostaje dopisywana i każdy nowo rejestrujący się klient ma ustawioną jako „user”.

## **5.3 Funkcjonalności ogólne**

Dla niezalogowanych jak i zalogowanych gości aplikacja została zaprojektowana, aby zapewniać intuicyjny dostęp do szerokiego zakresu funkcjonalności.

Niezależnie od poziomu dostępu, oferuje zestaw modułów, których celem jest usprawnienie obsługi klienta, zwiększenie komfortu użytkowania oraz ułatwienie dostępu do informacji dotyczących klubu.

Wśród dostępnych opcji znajdują się:

* Grafik zajęć - Aplikacja oferuje także przejrzysty grafik zajęć, który można przeglądać w układzie tygodniowym – od poniedziałku do niedzieli, co pozwala planować swoją aktywność zgodnie z aktualną ofertą klubu.
* Cennik – zawiera on informację o dostępnych karnetach oraz honorowanych kartach sportowych
* Rezerwacje – pozwala na przeglądanie historii własnych rezerwacji, jak i na sprawdzenie dostępności oraz rezerwację wybranych przedmiotów z magazynu klubowego (ta opcja dostępna jest tylko dla zalogowanych użytkowników)
* Panel treningów personalnych – umożliwia umawianie się na zajęcia z trenerem personalnym.
* Ustawienia – daje możliwość dostosowania wyglądu oraz działania aplikacji, poprzez możliwość zmiany motywu graficznego oraz języka interfejsu.
* Pomoc i kontakt – znajduje się tutaj lista najczęściej zadawanych pytań wraz z odpowiedziami.

W kolejnych częściach podrozdziału szczegółowo omówiony zostanie każdy z wymienionych modułów, przedstawiając ich działanie oraz rolę w kontekście funkcjonowania całej aplikacji.

Przy rozwijaniu każdej części tego podrozdziału, grafiki w nich przedstawione ilustrować będą jedynie odpowiednie panele – bez widoku całej aplikacji łącznie z paskiem górnym czy dolnym, co umożliwi na precyzyjniejsze zobrazowanie całej funkcjonalności bez potrzeby skupiania się na zewnętrznych motywach – dostępnych w każdym panelu.

### **5.3.1 Grafik zajęć**

Pierwszą z opcji dostępnych ogólnie w menu bocznym (oraz możliwą do wybrania także w menu dolnym) jest grafik zajęć. Jest to panel stworzony w przyjaznym dla użytkownika interfejsie, pozwalający przeglądać plan zajęć w zależności od wybranego dnia tygodnia:

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 19. Zrzut ekranu przedstawiający grafik zajęć aplikacji

Pasek dnia tygodnia można przewijać w lewo bądź w prawo umożliwiając wybranie odpowiedniego dnia. Każde z zajęć zostało zaimplementowane w formie okien z czasem trwania, nazwą zajęć oraz poziomem jego zaawansowania.

Po kliknięciu w zajęcia pojawia się odpowiednie okno ze szczegółowym opisem aplikacji:

Obraz zawierający tekst, zrzut ekranu, osoba, ubrania

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 20. Zrzut ekranu przedstawiający przykładowy, uzupełniony szczegółowy opis zajęć BJJ grupy łączonej

Obraz zawierający tekst, zrzut ekranu, multimedia, Ludzka twarz

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.Pozwala to zobaczyć na czym dany trening polega, kto jest jego głównym trenerem, jaki jest czas zajęcia oraz poziom zaawansowania. Dodatkowo, po kliknięciu w panel z imieniem i nazwiskiem trenera, możemy zobaczyć kontakt razem z podglądem zdjęcia trenera w formie wizytówki:

Rysunek 21. Zrzut ekranu przedstawiający przykładowy, uzupełniony podglądu kontaktu do trenera w formie wizytówki

### **5.3.2 Cennik**

Obraz zawierający tekst, zrzut ekranu, design

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.Kolejną z opcji jest okno cennika karnetów oraz honorowanych kart sportowych wraz z danymi kontaktowymi do administracji klubowej. Jest to prosty panel w formie zdjęcia oraz interaktywnego okna danych które po kliknięciu odpowiedniego wiersza (numeru kontaktowego, facebooka lub instagrama) przenosi nas kolejno do kontaktów z wybranym już numerem telefonu do klubowej recepcji, strony na platformie Facebook bądź Instagram:

Rysunek 22. Zrzut ekranu przedstawiający wygląd fragmentu cennika

### **5.3.3 Rezerwacje**

Obraz zawierający tekst, zrzut ekranu, wzór, design

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.Kluczową w kontekście użytkowym dla klientów jest panel rezerwacji. Pozwala on użytkownikom na zapoznanie się z ofertą klubowych gadżetów bądź sprzętu oraz zarezerwowanie ich dla klientów zalogowanych:

Rysunek 23. Zrzut ekranu przedstawiający jedną z części ekranu rezerwacji - ekran dostępności

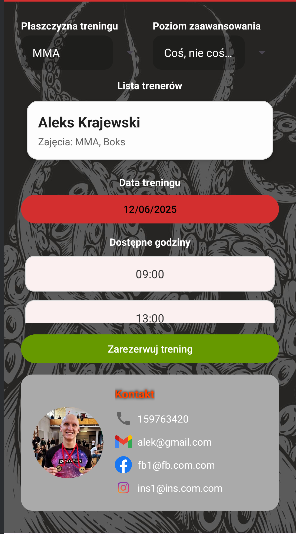
Okno to podzielone jest na dwa równorzędne panele – „Sprawdź dostępność” oraz „Moje rezerwacje”

W pierwszym segmencie klient ma możliwość sprawdzenia klubowej oferty z podziałem na sprzęt, akcesoria oraz ubrania oraz, będąc zalogowanym, także ich rezerwacji.

W drugiej części tego panelu natomiast użytkownik może sprawdzić jakie przedmioty zarezerwował, ich status oraz szczegóły rezerwacji które przesłał w formularzu, takie jak data zamówienia, czy charakterystyka zasobu.

### **5.3.4 Panel treningów personalnych**

Ważnym modułem dla interakcji użytkownika z aplikacją jest również panel treningów personalnych nazwany w menu bocznym „Nasi trenerzy”. Umożliwia on – poprzez wybranie odpowiedniej płaszczyzny treningu oraz poziomu zaawansowania – wyświetlenie listy dostępnych trenerów dla których klient może wziąć udział w treningu personalnym.

Poprzez wybranie daty, może on sprawdzić czy trener jest dostępny oraz w jakich godzinach, a po uzupełnieniu wszystkich danych, będąc zalogowanym, ma także możliwość zarezerwowania takiego treningu:

Rysunek 24. Zrzut ekranu przedstawiający przykładowy wybór trenera w panelu treningów personalnych oraz kontaktu

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.Płaszczyzny treningu, tak jak i poziom zaawansowania, są wyjątkowo odczytywane z bazy danych i automatycznie przypisywane do Spinnera w kodzie, co pozwala na późniejsze dodawanie lub usuwanie typów zajęć/poziomów zaawansowania:

Rysunek 25. Fragment kodu dotyczący przypisywania typów zajęć oraz poziomu zaawansowania w panelu treningów personalnych

### **5.3.5 Ustawienia**

W większości aplikacji, których jednym z celów podstawowych jest prezentowanie treści użytkownikowi, istotną rolę odgrywa możliwość dostosowania sposobu ich wyświetlania, dlatego też moduł ustawień stanowi integralną część opisywanej aplikacji, umożliwiając użytkownikowi zmianę motywu oraz dostosowanie języka interfejsu:

Obraz zawierający sztuka, tekst, wzór, design

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 26. Zrzut ekranu aplikacji przedstawiający wygląd panelu ustawień

Ustawienia te składają się z dwóch list (spinnerów) – motywu i języka.

Motyw możemy zmienić dla trybu jasnego oraz ciemnego, a także istnieje dodatkowa opcja – tryb domyślny, który dostosuje tą wartość do tego, jak jest to ustawione domyślnie w urządzeniu.

Język natomiast był jednym z trudniejszych założeń aplikacji, który udało się wykonać.

Aplikacja umożliwia dostosowanie języka w zależności od preferencji użytkownika – na język polski, ukraiński, bądź angielski.

Wszystko to udało się poprawnie zaimplementować, co zostało opisane w rozdziale 4.

Jak zostało wspomniane na początku rozdziału, aplikacja umożliwia podział każdego z użytkowników na odpowiednie rolę. W podrozdziale następnym opisanie zostanie każda z funkcjonalności dostępnych dla różnych rodzajów użytkowników:

## **5.4 Funkcjonalności z podziałem na rolę**

Każdy z użytkowników ma przydzieloną rolę po zarejestrowaniu się w aplikacji.

Projekt zakładał, że każda rola będzie posiadała dodatkowe panele (moduły) dostosowane odpowiednio do uprawnień. Założenie to udało się spełnić, a pierwszą i podstawową z ról jaka została utworzona był użytkownik (user).

## **5.4.1 Użytkownik**

Obraz zawierający tekst, plakat, zrzut ekranu, sztuka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.Jest to jedna z podstawowych przynależności osób korzystających z aplikacji. Poza funkcjonalnościami opisanymi w poprzednim podrozdziale, osoby z tą rolą mają dostęp do rezerwacji przedmiotów oraz treningów personalnych. Aby jednak było to w pełni możliwe, użytkownicy po pierwszej rejestracji muszą uzupełnić swoje dane w specjalnym dla nich oknie – „Mój profil” (Menu profilowe):

Rysunek 27. Zrzut ekranu obrazujący przykładowy ekran profilowy użytkownika, wypełniony tylko częściowo danymi

Poza możliwością edycji danych, okno to posiada również skrót do złożonych rezerwacji, które umożliwiają również wyświetlenie szczegółów po kliknięciu w odpowiednie zamówienie. Składać zamówienia można jednak jedynie wtedy, gdy uzupełnione są dane (rysunek pokazuje podgląd gdy dane zostały usunięte na rzecz pokazania jak działa podpowiedź instruktażowa z danymi jakie należy dodać, jednak wcześniej były one dodane, dlatego istnieją rezerwacje przypisane do konta).

### **5.4.2 Trener**

Projekt aplikacji w swoich założeniach uwzględnia również rolę trenera – osoby która ma większe uprawnienia od użytkownika, a dodatkowo posiada własny panel z danymi oraz dodatkowe funkcjonalności.

Obraz zawierający tekst, zrzut ekranu, design, projekt graficzny

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.Okno profilowe trenera różni się od tego dla osoby z rolą „user”. Opiekunowie poza możliwością edycji swoich danych otrzymują również możliwość dodawania godzin treningów personalnych dla odpowiednich dla siebie dat, a panel posiada również przycisk, który po kliknięciu przeniesie nas do zupełnie nowej funkcjonalności - stopera treningowego:

Rysunek 28. Zrzut ekranu przedstawiający przykładowo uzupełniony panel profilowy trenera, bez zdjęcia profilowego

Jak widać na załączonym obrazku, niektóre z godzin podświetlone są na zielono, to oznacza, że została złożona rezerwacja na trening, która została zaakceptowana przez trenera. Po kliknięciu takiej danej wyświetlą się szczegóły rezerwacji – kto zarezerwował trening oraz kontakt do tej osoby.

Dodatkowo, dzięki temu panelowi trener może przenieść się do stopera treningowego – funkcjonalności która umożliwia zarządzanie ćwiczeniami treningowymi, bez potrzeby organizowania własnej, dodatkowej aplikacji:

Obraz zawierający tekst, sztuka, plakat, design

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 29. Zrzut ekranu przedstawiający stoper treningowy dla trenera

### **5.4.3 Administrator**

Ostatnią z ról zaimplementowanych w projekcie jest rola admina, przystosowana dla twórcy aplikacji oraz właścicieli i pracowników klubu. Posiada ona najwięcej uprawnień. Umożliwia edycję poszczególnych zajęć w grafiku, ich szczegółowych opisów, edycję trenerów i dodawanie nowych bez konieczności tworzenia specjalnego konta.

Dodatkowo rola ta umożliwia zarządzanie strukturą administracyjną klubu w obrębie aplikacji – posiada ona specjalny panel administracyjny, pozwalający na bieżąco aktualizować dane dotyczące odbywających się treningów danego dnia, takie jak trener prowadzący zajęcia (ważne, jeśli odbyło się zastępstwo – główny trener nie mógł poprowadzić treningu i inny go zastąpił), czy ilość uczestników treningu.

Na podstawie takich danych, ale również złożonych rezerwacji czy zarejestrowanych użytkowników (dzięki zapisywanym w bazie danych czasom rejestracji), możemy wyświetlić statystyki, które odzwierciedlają zaangażowanie klientów w treningi, co jest szczególnie pomocne przy koordynacji codziennych działań klubu:

Obraz zawierający tekst, zrzut ekranu, design

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 30. Zrzut ekranu przedstawiający ekran statystyk dla administratora

Pokazane statystyki na podstawie zarejestrowanych użytkowników obrazują ilość użytkowników zarejestrowanych w danych dniach, co jest pomocne przy pokazaniu popularności aplikacji.

# **6. Możliwe kierunki rozwoju aplikacji**

Aplikacja w obecnym stanie spełnia swoje wszystkie podstawowe założenia. Umożliwia ona użytkownikom przeglądanie informacji na temat najnowszych zajęć, ogólny grafik przewidywany dla każdego dnia tygodnia, rezerwowanie przedmiotów dostępnych w magazynie klubowym oraz możliwości zapisania się do interesujących go trenerów w panelu treningów personalnych.

Niemniej jednak, wnioski z rozmów z trenerami oraz administracją i pracownikami klubowymi, a także rozwój technologii mobilnych, sprawia że warto rozważyć dalsze rozbudowywanie funkcjonalności projektu „Octopus”.

Wprowadzenie dodatkowych modułów i usprawnień może znacząco zwiększyć atrakcyjność i użyteczność aplikacji z perspektywy trenerów jak i ich podopiecznych.

W niniejszym rozdziale przedstawione zostały możliwe kierunki rozwoju aplikacji, których realizacja mogłaby wzmocnić więź z użytkownikami klubu sportowego.

Pierwszą z propozycji ulepszenia aplikacji w przyszłości jest panel aktualności:

## **6.1 Panel Aktualności**

Funkcjonalność, jaką byłby ten panel, umożliwiłaby użytkownikom śledzenie najnowszych informacji dotyczących klubu sztuk walk, publikowanych na oficjalnych profilach w mediach takich jak Facebook czy Instagram.

Takie rozwiązanie pozwoliłoby być na bieżąco z wydarzeniami, promocjami, komunikatami od trenerów czy zmianami organizacyjnymi, bez potrzeby opuszczania aplikacji.

Dzięki udostępnianiu przez Facebook oraz Instagram odpowiednich narzędzi, takich jak Graph API, możliwe byłoby pobieranie publicznych postów ze stron tych mediów. Ograniczeniem w tym rozwiązaniu jest wymaganie posiadania konta firmowego klubu oraz dostosowanie odpowiednich uprawnień, co jednak nie stanowi istotnej przeszkody, ponieważ klub dysponuje już takim kontem.

Podsumowując, panel aktualności pomógłby zwiększyć zaangażowanie użytkowników w stosowanie aplikacji oraz poprawiłby komunikację między klubem, a jego klientami, poprzez utrzymywanie stałego kontaktu.

Mimo wszystko, współczesne aplikacje mobilne nie ograniczają się jedynie do statycznego prezentowania treści – coraz częściej pełnią również rolę aktywnych narzędzi do komunikacji. Jednym z najbardziej efektywnych sposobów utrzymywania kontaktu z użytkownikiem są powiadomienia push, które umożliwiają przekazywanie informacji, takich jak chociażby powiadomienie o nowej aktualności na panelu, w czasie rzeczywistym.

## **6.2 System powiadomień systemowych**

Integracja powiadomień push pozwala na natychmiastowe informowanie użytkowników o istotnych zmianach, takich jak odwołanie treningu, zmiana godziny zajęć czy nagłe zastępstwo trenera. W połączeniu z panelem aktywności, może stanowić spójny system komunikacji między klubem, a jego klientami. Panel może służyć jako centrum najnowszych informacji, natomiast powiadomienia pełnią rolę sygnału, dzięki któremu użytkownik może zareagować na daną informację.

Z punktu widzenia technicznego, wdrożenie takiej funkcjonalności może zostać zrealizowane przy użyciu platformy Firebase Cloud Messaging, która pomogłaby w systemie dystrybucji wiadomości do urządzeń z systemem Android, co zapewniłoby szybkie i skuteczne przesyłanie najnowszych danych w aplikacji do użytkownika poprzez system mobilny.

W aplikacji „Octopus” udało zrealizować się wszystkie podstawowe założenia, jakie zostały postawione. Mimo tego, istnieje wiele możliwości rozwinięcia projektu poprzez wprowadzanie nowych usprawnień i technologii, takich jakie zostały przedstawione w tym rozdziale.

# **7. Bibliografia**

1. Josh Skeen, David Greenhalgh, Programowanie w języku Kotlin. The Big Nerd Ranch Guide, Helion 2019

2. Marcin Płonkowski, Android Studio. Tworzenie aplikacji mobilnych, Helion 2017

3. Craig Clifton, Learn Android Studio. Springer-Verlag Berlin and Heidelberg GmbH

& Co. KG 2015

4. Laurence Moroney, The Definitive Guide to Firebase. Build Android Apps on Google’s Mobile Platform, APress 2017

Źródła:

5.

# **8. Źródła ilustracji**

[Rysunek 1. Drzewo **schedule** w bazie danych Firebase Database 14](#_Toc200310233)

[Rysunek 2. Drzewo **classDescription** w bazie danych Firebase Database 15](#_Toc200310234)

[Rysunek 3. Drzewo ScheduleStatistics w bazie danych Firebase Database 16](#_Toc200310235)

[Rysunek 4. Kalendarz dla pracowników oraz właścicieli klubu 17](#_Toc200310236)

[Rysunek 5. Fragment kodu układu graficznego głównego ekranu, przedstawiający ustawienia dotyczące pewnych widoków tekstowych 19](#_Toc200310237)

[Rysunek 6. Fragment kodu pliku strings.xml, gdzie ręcznie przypisywane są odpowiednie tłumaczenia dla danych słów w języku angielskim 20](#_Toc200310238)

[Rysunek 7. Kod **App**, klasy odpowiadającej za dostosowywanie języka interfejsu przy ponownym uruchomieniu aplikacji 21](#_Toc200310239)

[Rysunek 8. Fragment kodu odpowiadającego za dodawanie zdjęcia trenera w szczegółach opisu grupy zajęciowej 22](#_Toc200310240)

[Rysunek 9. Fragment kodu odpowiadający za załadowywanie zdjęcia w interfejsie szczegółowego opisu grupy zajęciowej 23](#_Toc200310241)

[Rysunek 10. Fragment kodu odpowiadający za przycinanie oraz skalowanie zdjęcia poprzez technologię uCrop oraz Glide 23](#_Toc200310242)

[Rysunek 11. Zrzut ekranu funkcjonalności stopera treningowego 24](#_Toc200310243)

[Rysunek 12. Fragment kodu dotyczący stopera treningowego, odpowiadający za rozpoczęcie animacji 25](#_Toc200310244)

[Rysunek 13. Fragment kodu dotyczący stopera treningowego, odpowiadający za wznowienie animacji 26](#_Toc200310245)

[Rysunek 14. Fragment kodu dotyczący stopera treningowego, odpowiadający za rozpoczęcie rundy 26](#_Toc200310246)

[Rysunek 15. Zrzut ekranu, obrazujący wygląd ekranu głównego aplikacji 27](#_Toc200310247)

[Rysunek 16. Zrzut ekranu, obrazujący widok panelu bocznego służącego do nawigacji między fragmentami 28](#_Toc200310248)

[Rysunek 17. Zrzut ekranu przedstawiający panel logowania 29](#_Toc200310249)

[Rysunek 18. Zrzut ekranu przedstawiający panel rejestracji 30](#_Toc200310250)

[Rysunek 19. Zrzut ekranu przedstawiający grafik zajęć aplikacji 32](#_Toc200310251)

[Rysunek 20. Zrzut ekranu przedstawiający przykładowy, uzupełniony szczegółowy opis zajęć BJJ grupy łączonej 33](#_Toc200310252)

[Rysunek 21. Zrzut ekranu przedstawiający przykładowy, uzupełniony podglądu kontaktu do trenera w formie wizytówki 33](#_Toc200310253)

[Rysunek 22. Zrzut ekranu przedstawiający wygląd fragmentu cennika 34](#_Toc200310254)

[Rysunek 23. Zrzut ekranu przedstawiający jedną z części ekranu rezerwacji - ekran dostępności 35](#_Toc200310255)

[Rysunek 24. Zrzut ekranu przedstawiający przykładowy wybór trenera w panelu treningów personalnych oraz kontaktu 36](#_Toc200310256)

[Rysunek 25. Fragment kodu dotyczący przypisywania typów zajęć oraz poziomu zaawansowania w panelu treningów personalnych 36](#_Toc200310257)

[Rysunek 26. Zrzut ekranu aplikacji przedstawiający wygląd panelu ustawień 37](#_Toc200310258)

[Rysunek 27. Zrzut ekranu obrazujący przykładowy ekran profilowy użytkownika, wypełniony tylko częściowo danymi 38](#_Toc200310259)

[Rysunek 28. Zrzut ekranu przedstawiający przykładowo uzupełniony panel profilowy trenera, bez zdjęcia profilowego 39](#_Toc200310260)

[Rysunek 29. Zrzut ekranu przedstawiający stoper treningowy dla trenera 40](#_Toc200310261)

[Rysunek 30. Zrzut ekranu przedstawiający ekran statystyk dla administratora 41](#_Toc200310262)