**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 Liniowej

Łódź, 2025

1. 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 pracy jest przedstawienie projektu oraz implementacji mobilnej aplikacji stworzonej z myślą o funkcjonowaniu klubu sztuk walki – Octopus Łódź.

Aplikacja została zaprojektowana z uwzględnieniem potrzeb trzech głównych grup łączonych z klubem: 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 powiadomień push i 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, przy chociażby zapisywaniu 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, zarówno dla użytkowników zalogowanych jak i niezalogowanych, ma za zadanie oferować szeroki zakres funkcji dostępnych dla wszystkich osób 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, czy 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. Umożliwia użytkownikom sprawdzanie godzin, grupy zajęciowej, a także szczegółowego opisu treningu, który wyjaśnia, na czym polega trening, jaki jest poziom zaawansowania tego treningu, kto jest prowadzącym zajęcia w danym dniu oraz ile trwa dany trening.
* **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 na 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.

Kluczowe jest, aby użytkownik powinien móc również zarządzać tym, jak ma wyglądać aplikacja, aby dostosować ją do swoich potrzeb:

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:

* Włączenie lub wyłączenie powiadomień push, dotyczących ulubionych treningów, rezerwacji przedmiotów czy też zaakceptowania/odrzucenia rezerwacji na trening personalny z danym trenerem,
* 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 swoich danych w oknie profilowym:

3.1.5 Okno profilowe

Funkcja ta, dostępna jedynie dla użytkowników po zalogowaniu, ma pozwolić ingerować 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 dla administratorów 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

Dzięki takiemu rozwiązaniu, zarówno pracownicy klubu jak i sami właściciele mogą zaoszczędzić czas oraz lepiej zagospodarować miejsca na zajęcia, poprzez, przykładowo, dodanie większej ilości zajęć na które uczęszcza więcej osób bądź zastąpienie tych mniej popularnych.

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. Takie rozwiązanie ma również za zadanie zwiększyć motywację trenerów do korzystania z aplikacji. Stoper zaimplementowany w aplikacji może pomóc odrzucić wszystkie inne zewnętrzne rozwiązania w postaci aplikacji specjalnie do tego dostosowanych, dzięki czemu trenerzy będą mieli wszystko w jednym miejscu.

Aplikacja zapewnia również możliwość umawiania się z użytkownikami na trening personalny, dzięki możliwości zatwierdzania lub odrzucania rezerwacji z powodem w panelu powiadomień, do którego, gdy użytkownik pragnie złożyć prośbę o trening personalny, zostaje wysłana specjalna wiadomość, która informuje o tym, że została złożona rezerwacja na trening z konkretną datą i godziną.

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 i komunikacji między użytkownikami.

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.

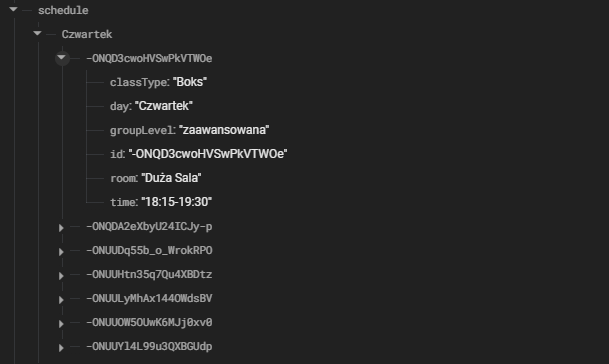
Synchronizacja tych dwóch widoków oraz szczegółów ich dotyczących wymagała zaprojektowania odpowiedniej struktury danych, która umożliwiłaby łatwe filtrowanie zajęć według dnia, jednocześnie nie prowadząc do niepotrzebnego powielania informacji.

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, skutkowało by 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.

Końcowo, wartości dotyczące godzin treningów wprowadzane 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:



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.

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.

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.



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.

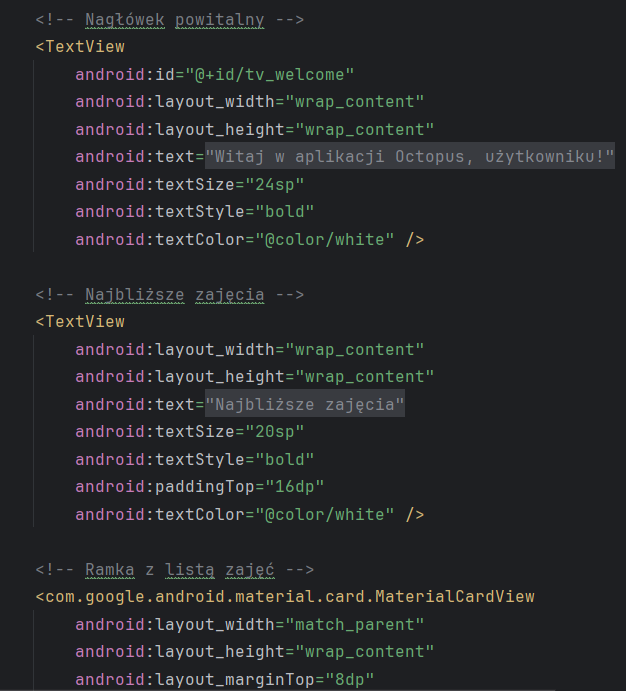
4.2.1 Motyw graficzny

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):



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.

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.

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 i automatycznie ładuje w odpowiednich miejscach interfejsu.

Dodatkowo, do dynamicznego wyświetlania obrazów wykorzystano bibliotekę Glide, która umożliwia szybkie i zoptymalizowane ładowanie zdjęć z chmury, jednocześnie oferując możliwość ich skalowania, zaokrąglania i dopasowywania do układu w aplikacji. Tak wygląda zastosowanie Glide w kodzie dotyczącym zdjęcia dla trenera w grafiku zajęć:

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