



Politechnika
Wrocławska



**Aplikacja wspomagająca wykonywanie treningu interwałowego w
kolarstwie szosowym – sprawozdanie**
PROGRAMOWANIE SYSTEMÓW MOBILNYCH

Autor:

GRZEGORZ SĘK 241778

Prowadzący zajęcia:

DR INŻ. MAREK PIASECKI

Termin zajęć:

Środa, 9:30 TP

Semestr zimowy 2021/2022

kod grupy: Y04-42b

19 stycznia 2022

Spis treści

1	Opis wstępny	2
1.1	Cel i zakres zadania	2
1.2	Funkcjonalny opis aplikacji	2
1.3	Założenia projektowe	2
2	Implementacja	3
2.1	Baza danych	3
2.2	Opis kolejnych okien (aktywności)	4
2.2.1	Menu główne	4
2.2.2	Dodawanie szablonu treningu	4
2.2.3	Wyświetlanie szablonów	5
2.2.4	Edytuj, uruchom, usuń	6
2.2.5	Edycja schematów treningów	6
2.2.6	Okno włączonego treningu	6
2.3	Napotkane problemy i sposoby ich rozwiązania	7
2.3.1	Typy zmiennych w bazie danych	7
2.3.2	Tabela nie zawiera takiej kolumny	7
2.4	Ekran uruchomionego treningu	8
2.5	Wykorzystane materiały	8
3	Podsumowanie	9
3.1	Wyniki	9
3.2	Co można dodać/poprawić?	9
3.3	Procentowy wykaz autorstwa poszczególnych modułów	9
4	DODATEK	11
4.1	Zrzuty ekranów	11
4.2	Opis projektu i założeń (wersja z ePortalu)	15
	Bibliografia	17

Rozdział 1

Opis wstępny

Opis wstępny aplikacji zawiera cel i zakres zadań wykonanych w trakcie realizacji projektu. Ponadto zostały opisane w nim również funkcjonalności aplikacji a także założenia projektowe.

1.1 Cel i zakres zadania

Celem tego projektu było wykonanie aplikacji mobilnej wspomagającej wykonywanie treningu interwałowego w kolarstwie szosowym.

Zakres prac:

1. wypisanie funkcjonalności aplikacji
2. wstępne ustalenie liczby okien aplikacji
3. zaprojektowanie okien aplikacji w środowisku Android Studio
4. połączenie bazy danych SQLite z aplikacją
5. testowanie aplikacji

1.2 Funkcjonalny opis aplikacji

Założenia funkcjonalne aplikacji:

- dodawanie schematu treningu
- edycja schematu treningu
- usuwanie schematu treningu
- uruchamianie schematu treningu
- możliwość pauzowania uruchomionego treningu
- możliwość restartu treningu

1.3 Założenia projektowe

Projekt zakładał stworzenie aplikacji mobilnej na smartfony z systemem android przy wykorzystaniu takich narzędzi jak:

- środowisko programistyczne na platformę Android – Android Studio
- system zarządzania relacyjną bazą danych – SQLite

Rozdział 2

Implementacja

Implementacja aplikacji została rozpoczęta od stworzenia jej graficznego interfejsu oraz połączenia aplikacji z bazą danych. Drobne napotkane problemy były Debugger’a dostępnego w Android Studio a także przy pomocy serwisu <https://stackoverflow.com/>.

Część mniej istotnych zrzutów ekranów aplikacji została przeniesiona do dodatku, aby te nie zaburzały czytelności sprawozdania.

2.1 Baza danych

Zgodnie z założeniami projektowymi aplikacja nie posiada wielu danych, na których mogłaby operować. W związku z tym zdecydowałem się na stworzenie w bazie danych SQLite tylko jednej tabeli. Dokładnie jest to tabela o nazwie *training_plan* (Rysunek 2.1) i zawiera one takie informacje jak:

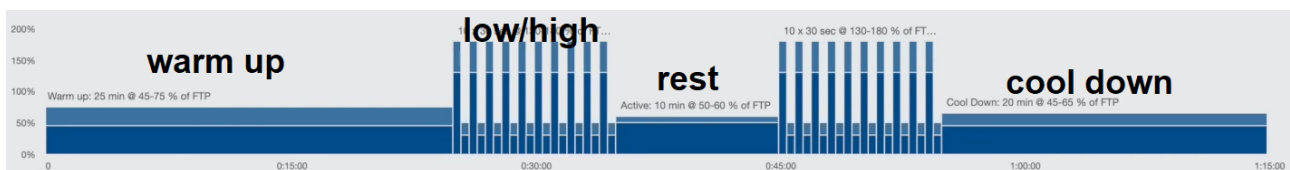
- *training__plan_id* – unikatowy numer dla każdego schematu
- *name* – czyli nazwę schematu treningu
- *warm_up* – czas trwania rozgrzewki
- *low* – czas trwania fragmentu treningu o niskiej intensywności
- *high* – czas trwania fragmentu treningu o wysokiej intensywności
- *sets* – czyli liczba powtórzeń sekcji *low/high*
- *rest* – czas trwania przerwy między kolejnymi cyklami interwałów
- *cool_down* – czas trwania ochłodzenia po zakończeniu wszystkich cykli interwałowych
- *cycles* – liczba cykli

training_plan	
training_plan_id	int
name	text
warm_up	text
low	text
high	text
sets	text
rest	text
cool_down	text
cycles	text

Rysunek 2.1. Projekt tabeli, źródło: własne

Jeśli chodzi o wykorzystane typy zmiennych to kolumna *training_plan_id* posiada typ *INTEGER* wraz z atrybutem umożliwiającym jego automatyczną inkrementację. Pozostałe kolumny posiadają typ zmiennej *TEXT* głównie z tego względu, że nie udało mi się znaleźć w jakiej formie powinienem zapisywać czas. Jednak w samym kodzie aplikacji część z otrzymanych wartości z bazy danych zostaje zmieniona na typy *INT* oraz *LONG*. W przypadku tej pierwszej wykorzystuję zwykłą funkcję dostępną w języku *java* do zamiany wartości *STRING* na *INT*. Natomiast dla zmiennych typu *LONG* konwersję wykonuję przy pomocy własnej funkcji oraz wbudowanej, które razem zamieniają dane z formatu *00:00* na typ *LONG*.

Aby lepiej zrozumieć, jak nazwy kolumn mają się do szablonu treningu, na rysunku (Rysunek 2.2) został przedstawiony przykładowy schemat treningu interwałowego.



Rysunek 2.2. Przykładowy schemat treningu interwałowego, źródło: [1]

2.2 Opis kolejnych okien (aktywności)

2.2.1 Menu główne

Jest to pierwsze okno, które wyświetla się po uruchomieniu aplikacji. Znajdują się tutaj dwa przyciski: *Add new training plan* oraz *Run*. Pierwszy przycisk przenosi do nowej aktywności, gdzie można dodać nowy schemat treningu. Natomiast ten drugi również powoduje uruchomienie nowej aktywności, lecz w tej wyświetlane są, dodane wcześniej, szablony treningów. Wygląd tego okna w orientacji pionowej jak i poziomej został przedstawiony kolejno na rysunkach (Rysunek 4.1), (Rysunek 4.2).

2.2.2 Dodawanie szablonu treningu

W tym oknie można stworzyć sobie wzór treningu interwałowego, który później będzie można uruchomić. Okno w orientacji *portrait* prezentuje się następująco (Rysunek 2.3). W przypadku orientacji

poziomej wygląda ono niemalże identycznie (Rysunek 4.3)

Rysunek 2.3. Okno dodawania schematu treningu, źródło: własne

2.2.3 Wyświetlanie szablonów

Okno to prezentuje wszystkie szablony treningów, a właściwie ich nazwy, znajdujących się w bazie danych. Przedstawione są w postaci listy. Naciśnięcie jednego z elementów tej listy powoduje otwarcie okna opcji, w którym możemy wybrać czy chcemy uruchomić ten trening, edytować go, czy też usunąć. Okno w postaci *portrait* wygląda następująco (Rysunek 2.4). Natomiast w orientacji *landscape* (Rysunek 4.4).



Rysunek 2.4. Lista dostępnych szablonów, źródło: własne

2.2.4 Edytuj, uruchom, usuń

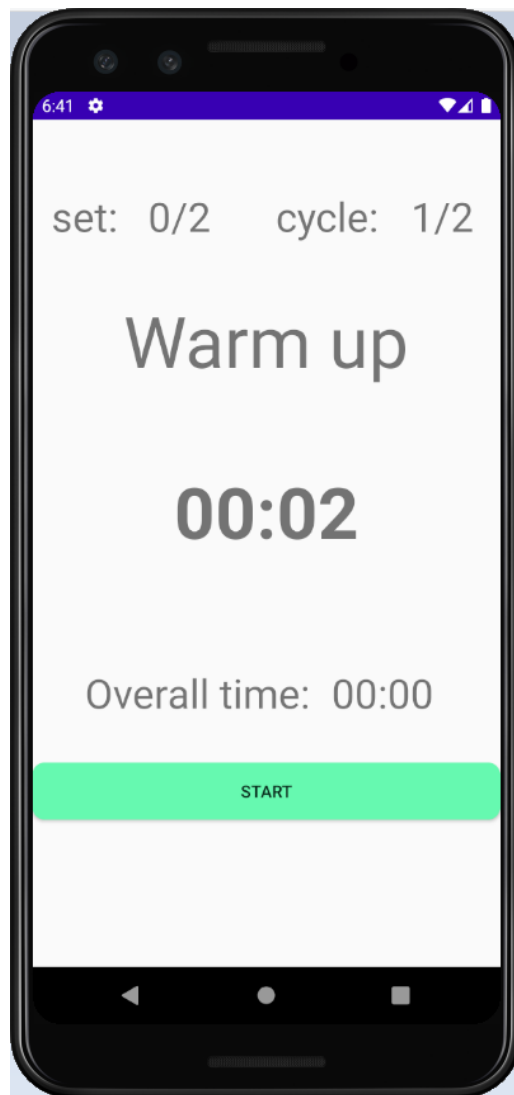
To okno otwiera się po uprzednim wybraniu szablonu z listy dostępnych. Znajdują się w nim trzy przyciski *EDIT*, *RUN* oraz *DELETE*, które wykonują operacje zgodne z ich nazwami. Okno w wersji pionowej przedstawione zostało na rysunku (Rysunek 4.5), natomiast w formie poziomej na rysunku (Rysunek 4.6).

2.2.5 Edycja schematów treningów

Jeśli chodzi o samą oprawę graficzną okno to jest identyczne jak w przypadku okna *dodaj schemat*. Jedyna różnica kryje się w kwestii programistycznej, gdzie polecenie SQL *INSERT* zostało zastąpione poleceniem *UPDATE*. Domyślnie także zostały wpisane dane w pola typu *EditText* zgodnie z tym co znajduje się rekordzie w bazie danych.

2.2.6 Okno włączonego treningu

Jest to okno, w którym tak naprawdę dzieje się najwięcej. Wyświetlane są tutaj takie informacje jak aktualna nazwa części treningu, licznik aktualnie wykonywanego elementu z serii, a także licznik całych serii. W trakcie działania, istnieje możliwość pauzowania aktualnego etapu oraz resetu całości. Ponadto kolor ekranu zmienia się w zależności od aktualnie wykonywanej części treningu 4.8. Okno w orientacji pionowej przedstawione zostało na rysunku (Rysunek 2.5), natomiast w orientacji poziomej na rysunku (Rysunek 4.7).



Rysunek 2.5. Okno wykonywanego treningu, źródło: własne

2.3 Napotkane problemy i sposoby ich rozwiązania

2.3.1 Typy zmiennych w bazie danych

Tak jak już wcześniej wspomniałem jednym z początkowych problemów był sposób zapisu czasu trwania poszczególnych sekcji treningu w bazie danych. Z tego względu, że nie znalazłem w jaki sposób zapisywać czas trwania to problem ten rozwiązałem poprzez zapisanie takiego czasu w formacie `00:00` zmiennej typu `TEXT`, który następnie w aplikacji zamieniany jest przy pomocy własnej funkcji oraz funkcji wbudowanych na zmienne typu `INT` oraz `LONG`.

2.3.2 Tabela nie zawiera takiej kolumny

Jest to problem nad którym spędziłem zdecydowanie za dużo czasu, gdyż szukałem pomyłki (*literówki*) w poleceniach SQL'owych, której tak naprawdę nie było. Po wyszukaniu błędu w serwisie <https://stackoverflow.com/> okazało się, że do tabeli w bazie danych prawdopodobnie została dodana nowa kolumna i wystarczy zmienić wersję bazy danych, co pozwoli na wywołanie metody `onUpgrade`, która usunie tabelę i stworzy ją na nowo. Innym sposobem rozwiązania tego problemu było usunięcie aplikacji i zainstalowanie jej ponownie. Oba sposoby wykorzystałem i oba równie dobrze zadziałały.

2.4 Ekran uruchomionego treningu

Jest to funkcjonalność aplikacji, której poświęciłem najwięcej czasu. Wymyślając temat projektu spodziewałem się, że wystarczą tutaj zagnieżdżone w sobie pętle *for*, jednak problem okazał się odrobinę trudniejszy. Początkowo próbowałem rozwiązać go wieloma instrukcjami warunkowymi *if*, lecz nie przyniosło to pożądanego efektu. Ostatecznie rozwiązałem problem stosując instrukcje *switch* i *case*, które pomogły w kontrolowaniu złożonych (przynajmniej dla mnie) operacji warunkowych.

2.5 Wykorzystane materiały

Jeśli chodzi o pomoce naukowe to korzystałem z poprzednich projektów, które realizowałem na tym laboratorium, jak i materiałów do wykładów. Poza wyżej wymienionymi posłużyłem się jeszcze trzema podstawowymi poradnikami dostępnymi w serwisie YouTube, a mianowicie:

- poradnik dotyczący posługiwania się SQLite (zapisywanie danych) [2]
- poradnik dotyczący posługiwania się SQLite (edycja i wyświetlanie danych) [3]
- poradnik prezentujący działanie klasy *Chronometer* w Java [4]

Rozdział 3

Podsumowanie

3.1 Wyniki

Ostateczne wyniki odrobinę się różnią od wstępnych założeń projektowych. Aplikacja posiada więcej okien niż przewidywałem na początku, lecz nie ma to żadnego wpływu na jej działanie czy też funkcjonalności.

W aplikacji brakuje wyświetlania takich danych jak: moc w trakcie trwania etapu treningu o niskiej i wysokiej intensywności, moc w trakcie przerwy oraz moc w trakcie etapu *cool down*. Jest to spowodowane ograniczonymi zasobami czasu na wykonanie tego projektu, aczkolwiek w przyszłości dodanie wyświetlania tych parametrów będzie wręcz banalne. Wystarczy, w analogiczny sposób, dodać te parametry do bazy danych oraz do interfejsu tak jak pozostałe informacje np.: liczba powtórzeń etapu *low/high* czy też liczbę cykli.

Z opcjonalnych części projektu dodałem do aplikacji zmianę koloru ekranu w trakcie zmiany etapu treningu. Z pewnością taka funkcjonalność będzie przydatna w trakcie jazdy na rowerze, dzięki temu wystarczy zerknąć na smartfon by wiedzieć o aktualnie wykonywanym etapie treningu.

Ostatecznie uważam projekt za średnio trudny, lecz jednak musiałem sporo czasu poświęcić na zapoznanie się z nowymi dla mnie elementami. Niewątpliwie osoba, która ma doświadczenie w tym temacie, taki projekt byłaby w stanie wykonać w te ustalone osiem godzin.

3.2 Co można dodać/poprawić?

- dodanie sygnału dźwiękowego w trakcie zmiany etapu treningu
- modernizacja interfejsu użytkownika, aby aplikacja była bardziej przyjazna dla oka
- dodanie wyświetlania mocy w trakcie każdego etapu treningu
- dodanie wyświetlania parametrów z pulsometru (komunikacja bluetooth)

3.3 Procentowy wykaz autorstwa poszczególnych modułów

Procentowy wykaz autorstwa poszczególnych klas:

- AddNewTrainingPlan (okno dodawania schematu) – 50%
- DatabaseHelper (klasa odpowiedzialna za obsługę BD) – 30%
- DisplayTrainingPlan (wyświetlanie listy schematów) – 50%
- EditPlan (okno edycji schematów) – 95%
- MainActivity (główne menu) – 100%
- PlanOptions (mini menu konkretnego schematu) – 100%

- RunActivity (ekran uruchomionego treningu) – 90%

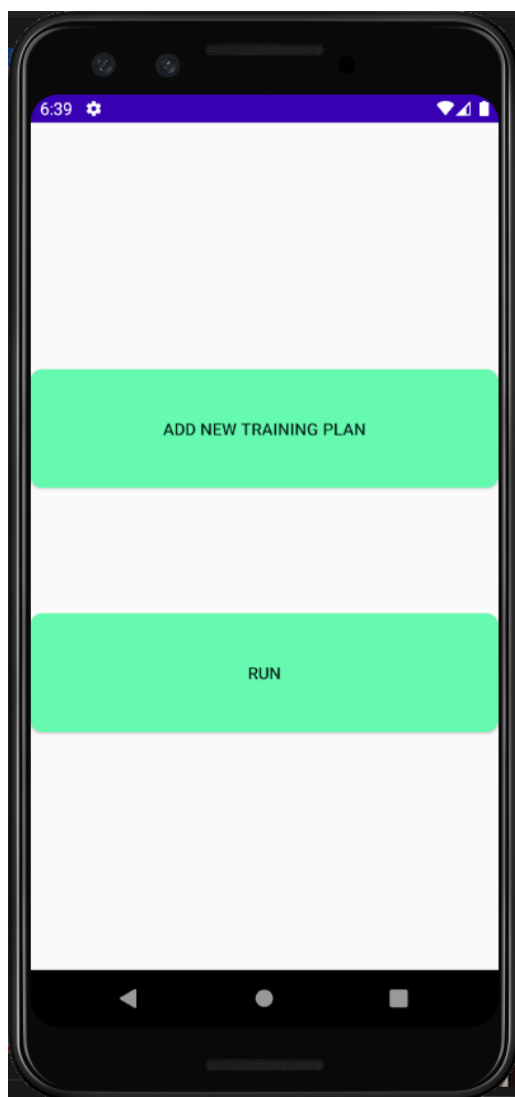
W przypadku interfejsu użytkownika (UI) jest to 100%

Rozdział 4

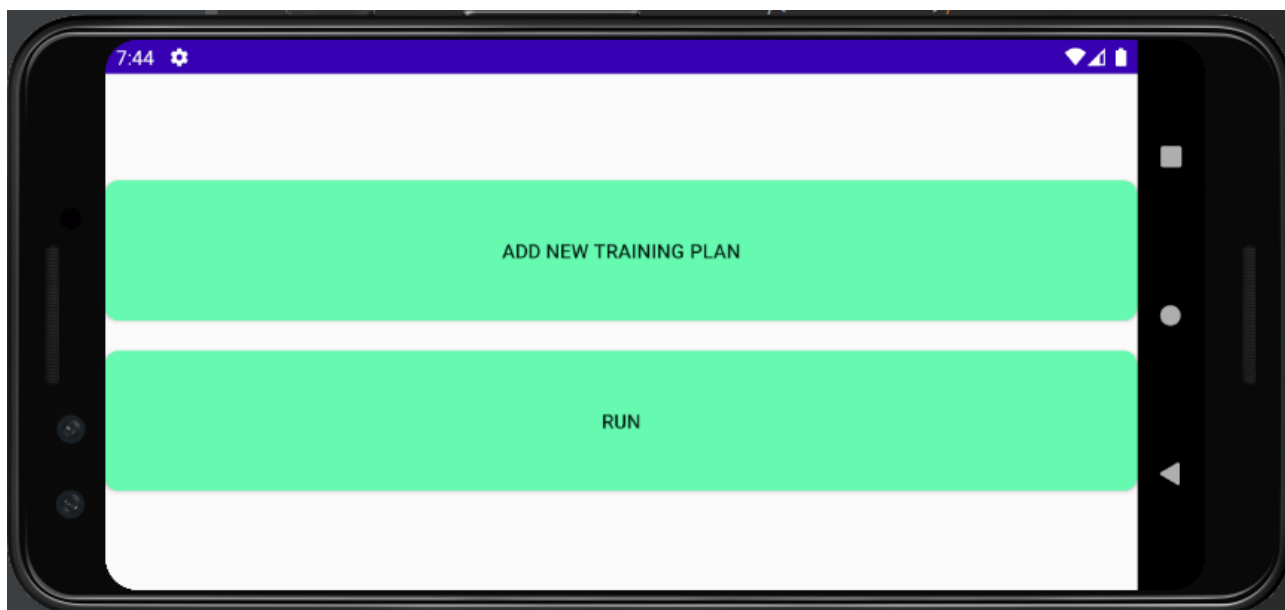
DODATEK

W dodatku zostały umieszczone zrzuty ekranów, które nie mogły być dodane do głównej części sprawozdania ze względu na swoje rozmiary. Na końcu również zostały umieszczone założenia projektu, które były umieszczone na ePortalu.

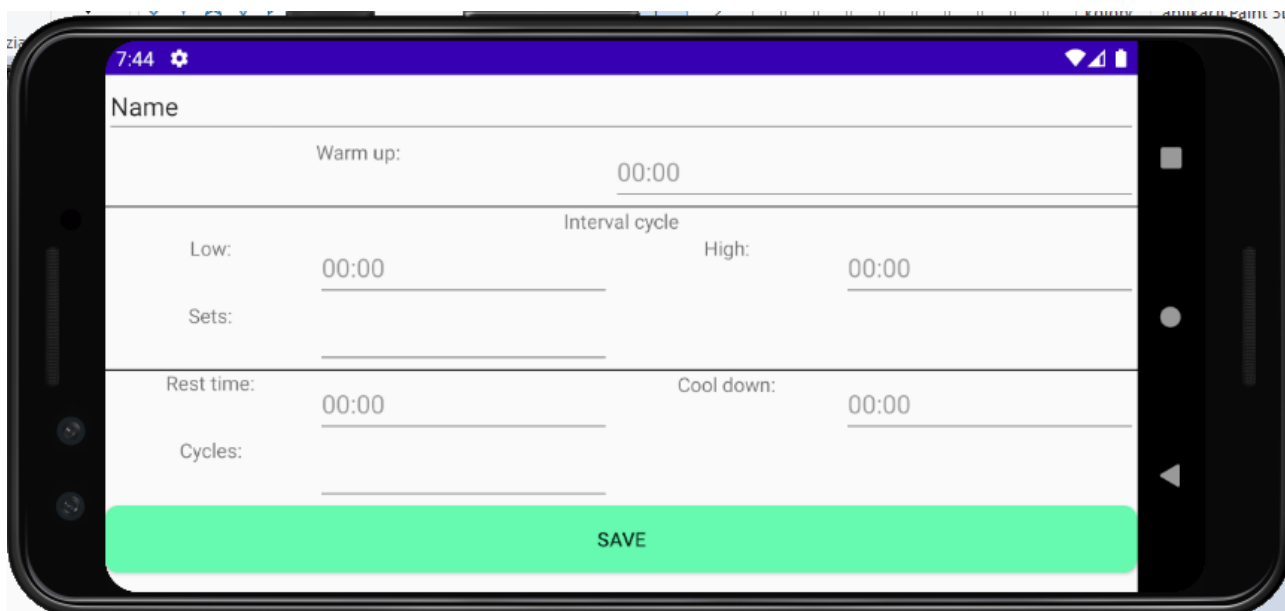
4.1 Zrzuty ekranów



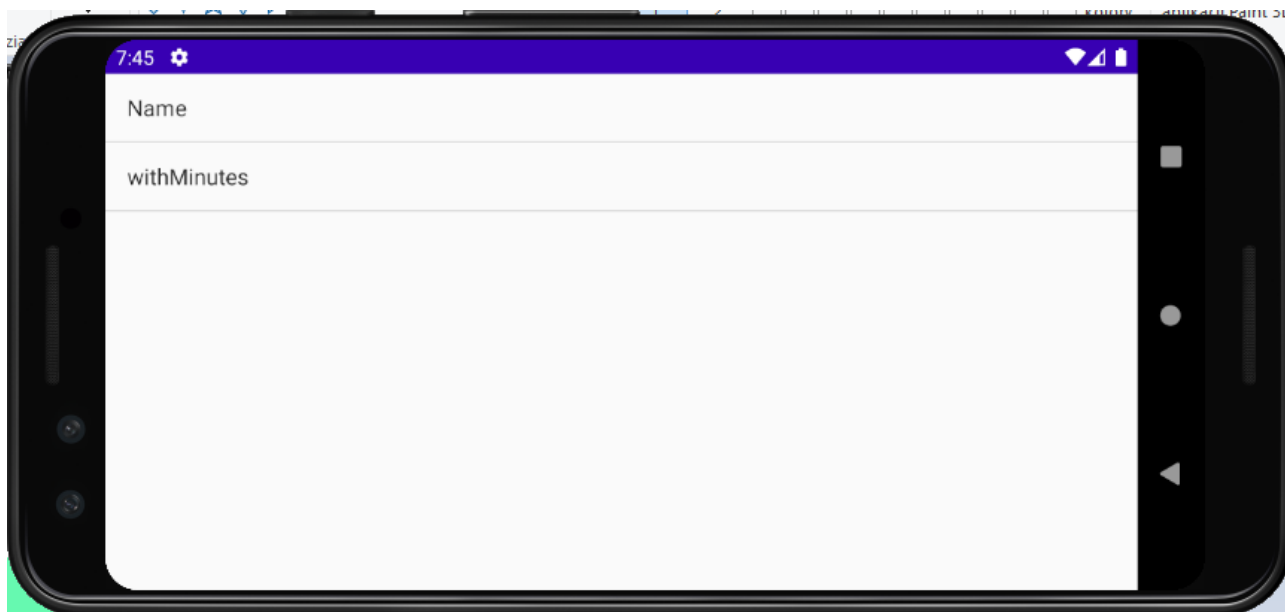
Rysunek 4.1. Główne menu, źródło: własne



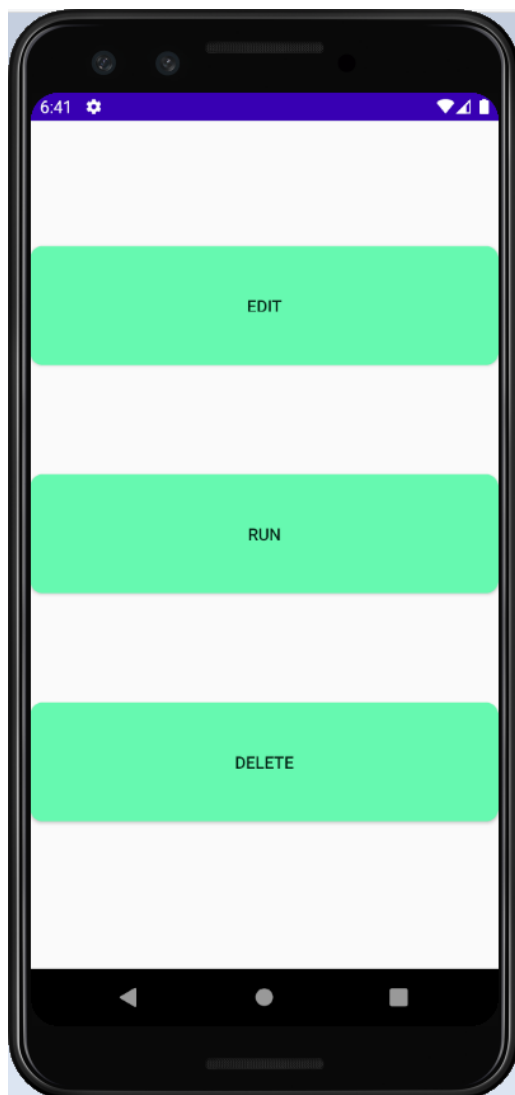
Rysunek 4.2. Główne menu, źródło: własne



Rysunek 4.3. Okno dodawania treningu, źródło: własne



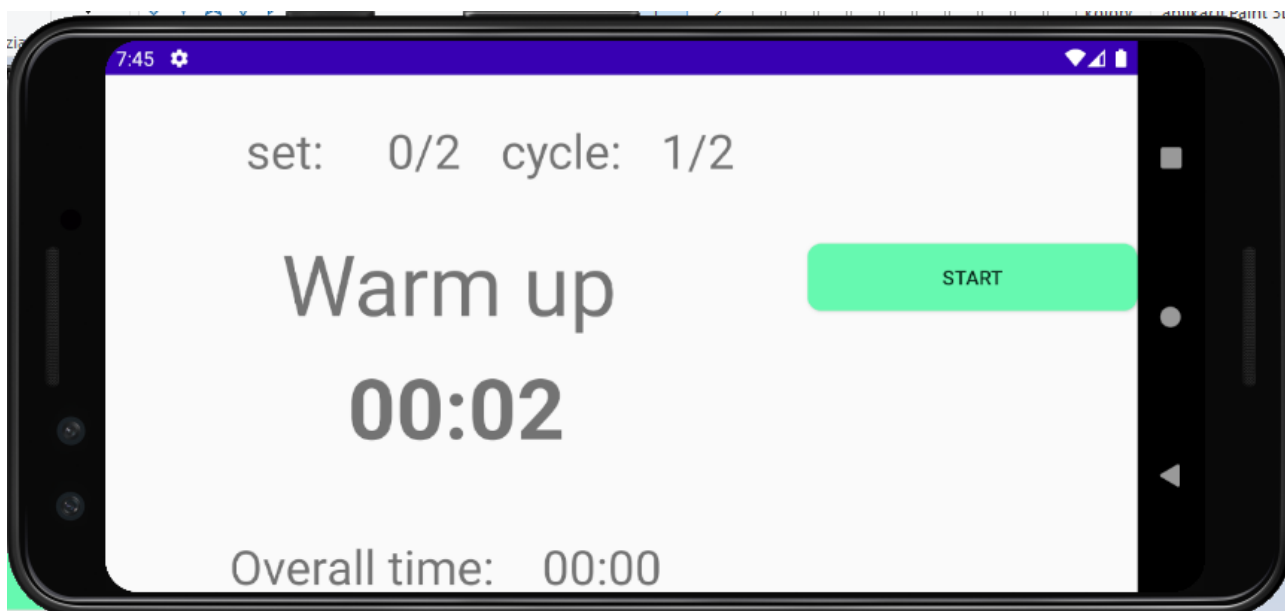
Rysunek 4.4. Okno wyświetlające dostępne treningi, źródło: własne



Rysunek 4.5. Opcje schematu, źródło: własne



Rysunek 4.6. Opcje schematu, źródło: własne



Rysunek 4.7. Okno uruchomionego treningu, źródło: własne



Rysunek 4.8. Przykład zmiany koloru, źródło: własne

4.2 Opis projektu i założeń (wersja z ePortalu)

Aplikacja będzie zapisywała schematy treningów w bazie danych SQLite.

Wstępnie przewiduję cztery okna (aktywności):

- menu główne
- dodaj plan treningowy
- uruchom plan treningowy
- aktywność uruchomionego treningu interwałowego

Zawartość okien:

- dodaj plan treningowy:
 - czas trwania rozgrzewki
 - czas trwania etapu treningu o niskiej i wysokiej intensywności
 - moc w trakcie trwania etapu treningu o niskiej i wysokiej intensywności
 - liczba powtórzeń tych etapów
 - liczba powtórzeń całego cyklu interwałowego
 - czas trwania przerwy między cyklami
 - moc w trakcie przerwy
 - czas trwania ochłodzenia (cool down)
- uruchom plan: W tym oknie będą wyświetlane zapisane w bazie danych plany treningów. będzie możliwość ich uruchomienia właśnie z tego okna
- aktywność uruchomionego treningu interwałowego:
 - wyświetlanie nazwy aktualnego etapu treningu
 - wyświetlanie czasu trwania aktualnego etapu
 - wyświetlanie aktualnego numeru cyklu np. 1/2

- wyświetlanie aktualnego numeru etapu treningu o niskiej aktywności np. 1/10
- wyświetlanie aktualnego czasu trwania treningu
- wyświetlanie wymaganej mocy w trakcie trwania danego etapu treningu
- (opcjonalnie) zmiana koloru ekranu w trakcie przechodzenia między etapem treningu o wysokiej intensywności a etapem treningu o niskiej intensywności np. czerwony ekran w trakcie wysokiej intensywności i zielony w trakcie niskiej

Bibliografia

- [1] Dylan Jhonson. What are the most effective intervals? hiit science. https://www.youtube.com/watch?v=YBgAr7kLsZY&t=736s&ab_channel=DylanJohnson.
- [2] CodingWithMitch. Save data into sqlite database [beginner android studio example]. https://www.youtube.com/watch?v=aQAIMY-HzL8&ab_channel=CodingWithMitch.
- [3] CodingWithMitch. Editing and deleting data from an sqlite database [beginner android studio example]. https://www.youtube.com/watch?v=nY2bYJyGty8&ab_channel=CodingWithMitch.
- [4] Coding in Flow. Chronometer / stopwatch with start, pause, resume, reset - android studio tutorial. https://www.youtube.com/watch?v=RLnb4vVkftc&ab_channel=CodinginFlow.