

PROJEKT

PROGRAMOWANIE SYSTEMÓW MOBILNYCH

Aplikacja natywna Moto Manager

Autor:

Adam KRZYKAŁA, 235411

Termin: poniedziałek 9:15

Prowadzący:

dr inż. Marek PIASECKI

25 stycznia 2021

Spis treści

1	Wstęp	2
1.1	Cel projektu	2
1.2	Założenia projektowe	2
1.3	Użyte technologie	2
2	Prezentacja aplikacji	3
2.1	Implementacja	3
2.2	Zaimplementowane funkcjonalności	5
2.2.1	Pobieranie danych o nowym pojeździe	5
2.2.2	Przejście pomiędzy częścią użytkową (garażem), a serwisem	5
2.2.3	Widok menu pojazdu	6
2.2.4	Tracker przejazdu	7
2.2.5	Aparat i zapisywanie zdjęć pod wskazanym adresem w pamięci urządzenia	7
2.2.6	Widok menu serwisu	9
2.2.7	Dodawanie nowej czynności serwisowej	10
2.2.8	Wyświetlanie wszystkich dotychczasowych napraw z możliwością udostępnienia przez e-mail	10
2.2.9	Ustawianie i zarządzanie powiadomieniami	11
2.2.10	Wyświetlanie lokalnego pliku PDF	12
2.3	Animacje	12
2.4	Uprawnienia	12
3	Responsywność interfejsu graficznego	13
4	Podsumowanie	13
	Bibilografia	13

1 Wstęp

1.1 Cel projektu

Celem projektu było stworzenie aplikacji na system Android, która wspomagałaby eksploatację pojazdu (w prezentowanej wersji motocyklu). Rozwiązanie ma za zadanie śledzić i zapisywać trasy przejazdów, a jednocześnie zapisywać wszelkie prace serwisowe i zgłaszać potrzebę ich przeprowadzenia.

1.2 Założenia projektowe

Projekt powstał w ramach przedmiotu Programowanie systemów mobilnych na Politechnice Wrocławskiej. Postawione założenia:

- stworzenie złożonej aplikacji na system Android wykorzystującej poznane technologie w ramach przeprowadzonych wykładów i laboratoriów,
- użycie przynajmniej jednej technologii mobilnej spoza prezentowanego zakresu,
- zaprogramowanie interfejsu użytkownika o średniej złożoności (3-4 widoki / aktywności lub fragmenty) dopasowujący się do konfiguracji i orientacji urządzenia.

1.3 Użyte technologie

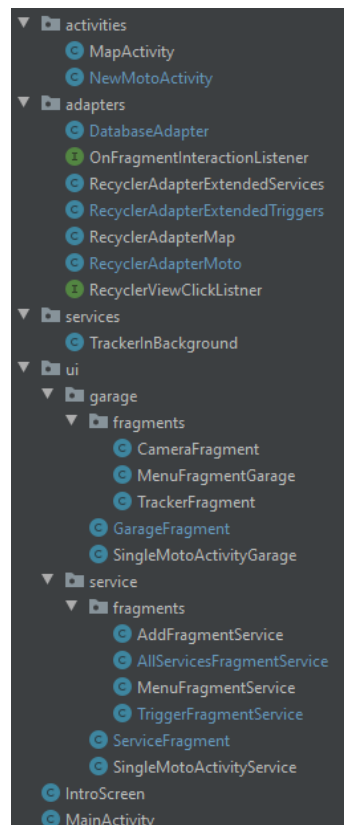
- Android Studio jako środowisko programistyczne,
- Firebase jako narzędzie do tworzenia aplikacji mobilnych - baza danych i chmura,
- SQLite w celu zapisywania interakcji użytkownika z aplikacją w lokalnej bazie danych, do późniejszego odczytu,
- Google API do obsługi map Google.

2 Prezentacja aplikacji

2.1 Implementacja

Projekt składa się z funkcjonalności, które zostały zaimplementowane w logice biznesowej aplikacji. Logika biznesowa składa się z:

1. aktywności
 - IntroScreen - aktywność 5 sekundowego animowanego pokazu slajdów
 - MainActivity - główna aktywność,
 - MapActivity - aktywność pozwalająca na korzystanie z map Google,
 - NewMotoActivity - aktywność pozwalająca pobrać pojazd z bazy danych Firebase,
 - SingleMotoActivityService - aktywność pozwalająca dokonywać operacji serwisowych,
 - SingleMotoActivityGarage - aktywność pozwalająca na korzystanie z pojazdu,
2. fragmenty - pozwalające na wydzieleniu operacji w ramach aktywności,
3. usługa pobierania w tle dokumentu PDF - serwisówki pojazdu,
4. adaptery
 - adaptery RecyclerView - klasy rozszerzające możliwości kontenerów RecyclerView,
 - DatabaseAdapter - klasa pozwalająca wraz z metodami na interakcję z bazą lokalną SQLite,
5. interfejsy
 - OnFragmentInteractionListener - interfejs umożliwiający komunikację między fragmentem a aktywnością,
 - RecyclerViewClickListener - interfejs pozwalający na podjęcie aktywności po kliknięciu w obiekt umieszczony w RecyclerView.

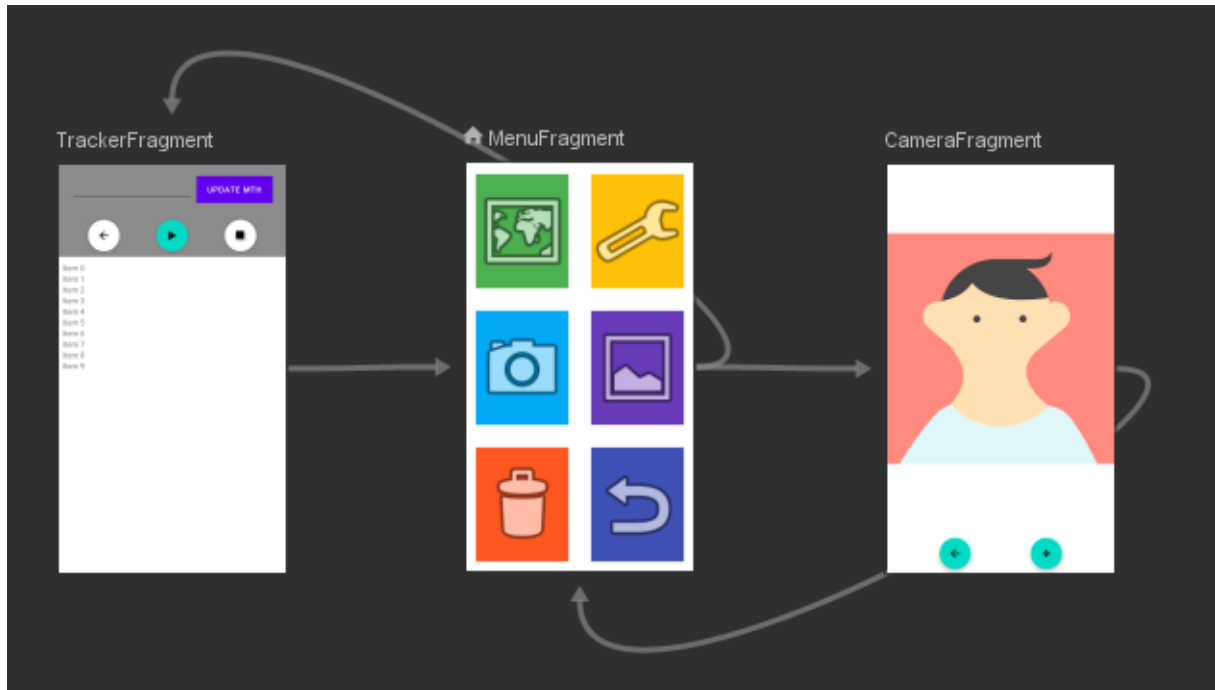


Rysunek 1: Drzewo projektu

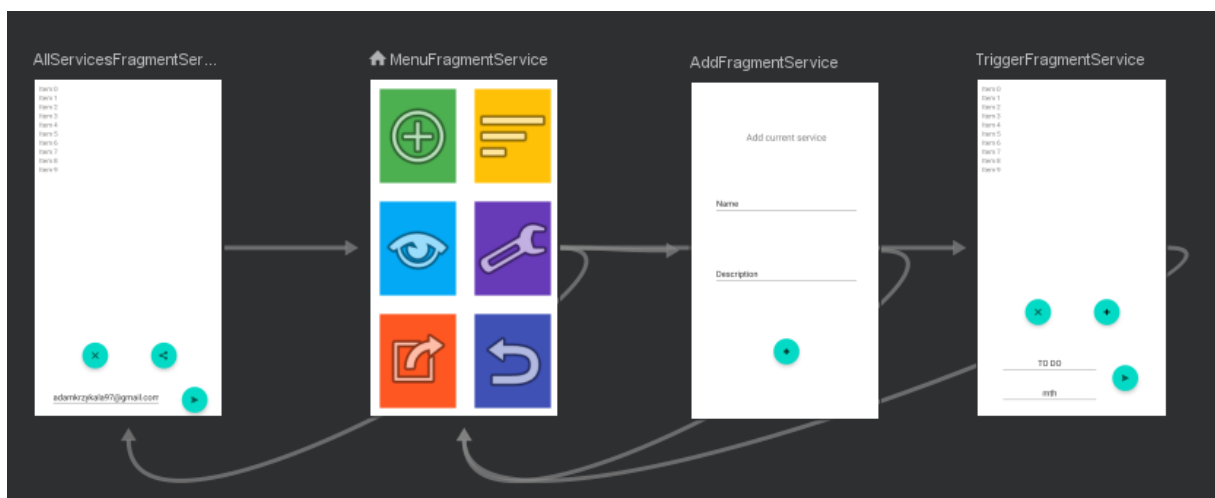
Pliki źródłowe składają się głównie z

- drawable - pomocniczych plików .XML,
- menu - pliki wysuwanego menu AppBarConfiguration,
- layout - wszystkie dostępne widoki projektu, zarówno dla różnych funkcjonalności, jak i różnych konfiguracji,
- navigation - pliki definiujące nawigację pomiędzy fragmentami.

Istnieją dwie główne grupy fragmentów:



Rysunek 2: Fragmenty po stronie części użytkowej - garaż

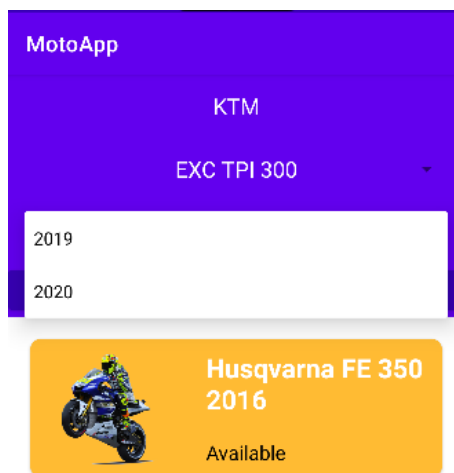


Rysunek 3: Fragmenty po stronie części serwisowej

2.2 Zaimplementowane funkcjonalności

2.2.1 Pobieranie danych o nowym pojeździe

Dane pobierane są dzięki usługom Firebase. Na zdalnym serwerze w bazie danych NoSQL umieszczone są dane o dostępnych w bazie pojazdach. Po uruchomieniu aktywności pozwalającej na dodawanie do aplikacji, pola wyboru uzupełniane są jako wynik połączenia z bazą i pobrania przykładowego modelu. Do możliwości wyboru natomiast ładowane są wszyscy producenci, modele, roczniki. Wybór ma charakter kaskadowy, co powoduje, że wybór producenta warunkuje treść pola wyboru modelu. Łączenie z bazą jest możliwe za pomocą przechwyconej instancji bazy i składowej jako `StorageReference`.



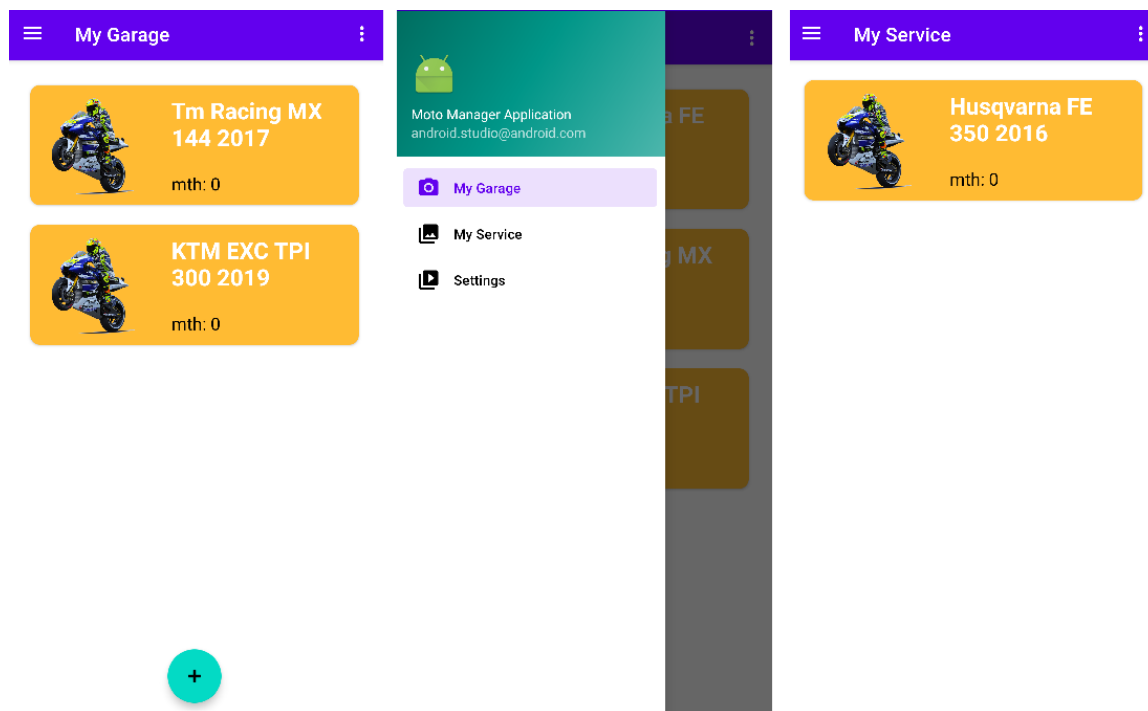
Rysunek 4: W bazie składowane są dwa roczniki z wybranego modelu

Kliknięcie wyświetlonej propozycji spowoduje dodanie pojazdu do bazy lokalnej (garażu) oraz pobranie serwisówki dla konkretnego modelu z dysku Firebase.

Pobieranie serwisówki w formie pliku PDF jest możliwe dzięki menadżerowi pobierania `DownloadManager`. Plik z danej w bazie lokalizacji (Firebase Storage) jest pobierany do pamięci urządzenia i zapisywany pod zadaną nazwą. Menadżer pobierania umożliwia przejście od razu do użytkowania aplikacji, pobierając plik w tle.

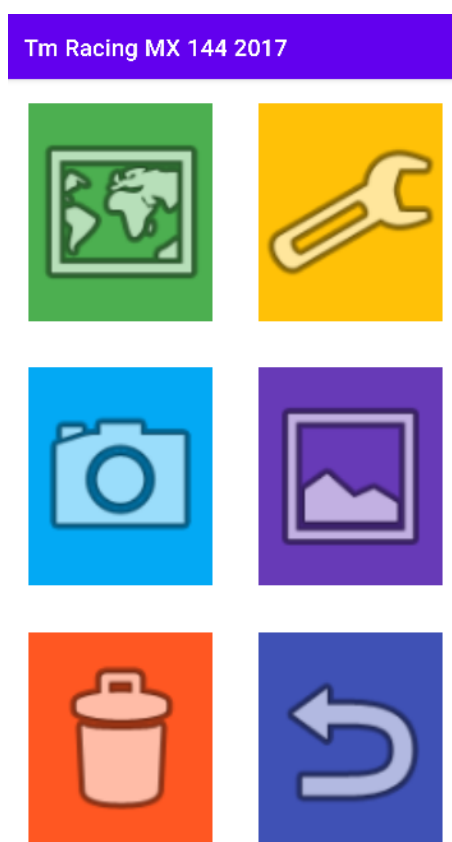
2.2.2 Przejście pomiędzy częścią użytkową (garażem), a serwisem

Po pobraniu pojazdu z bazy jest on dodawany do garażu. Wszystkie pojazdy są widoczne w liście. Przejście jest możliwe poprzez wysuwane menu boczne.



Rysunek 5: Widok garażu oraz części serwisowej

2.2.3 Widok menu pojazdu



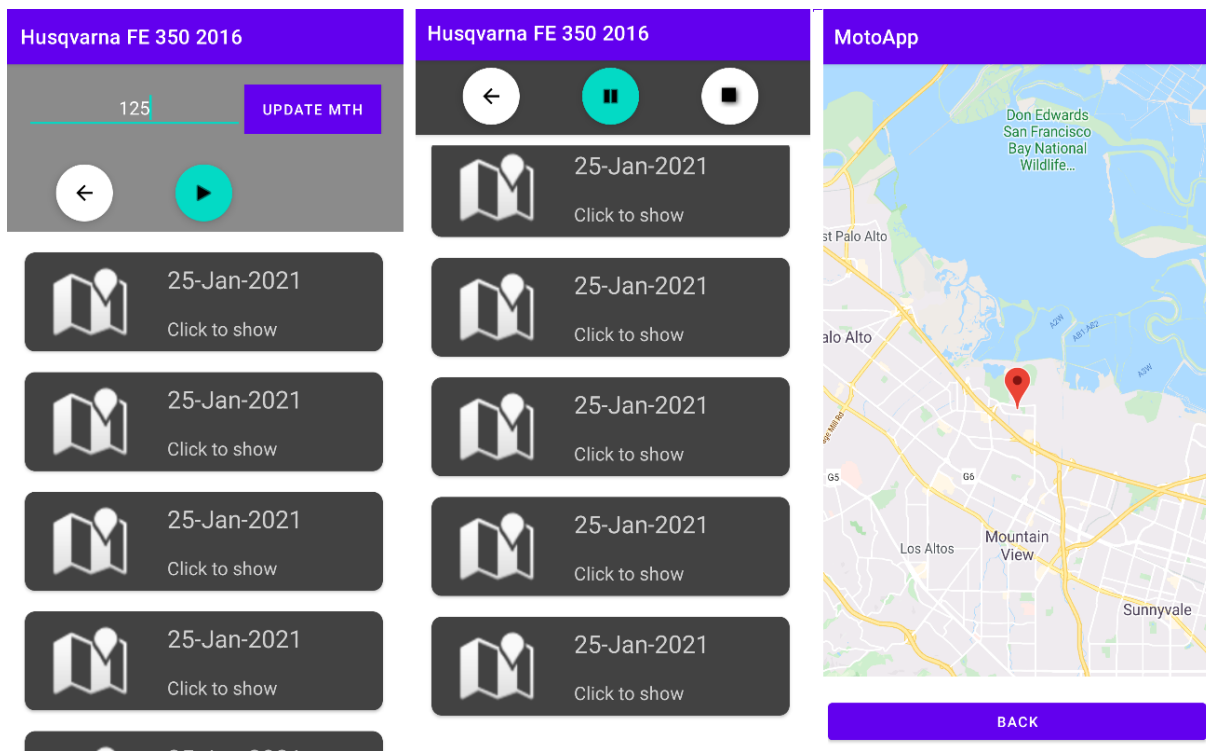
Rysunek 6: Widok menu pojazdu

Możliwe działania w obecnej aktywności

- Tracker przejazdu - aktualizacja motogodzin po jeździe oraz rejestrowanie położenia
- Service - dodanie pojazdu do części serwisowej
- Photo - możliwość robienia i zapisywania zdjęć w galerii w lokalizacji aplikacji
- Gallery - galeria zdjęć z lokalizacji aplikacji
- Delete - usunięcie pojazdu z bazy lokalnej
- Back - powrót do garażu i możliwość wyboru innego pojazdu.

2.2.4 Tracker przejazdu

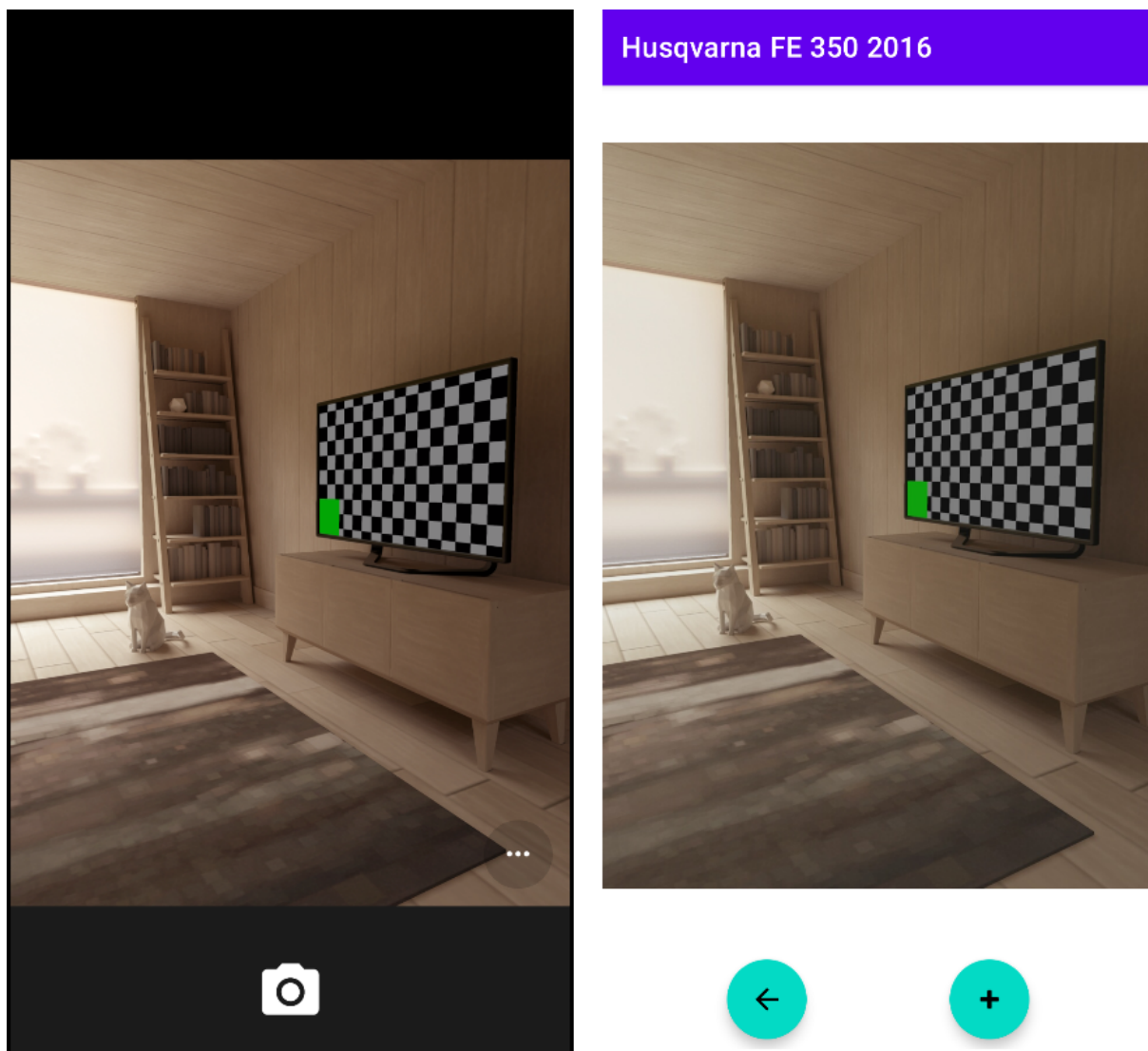
Sekcja ta pozwala na zapisywanie do bazy obecnego przejazdu w postaci punktów, a następnie dodawaniu do zapisanej listy. Istnieje możliwość podglądu zapisanej ścieżki przejazdu. Śledzenie i zapisywanie do bazy odbywa się w tle. Klasa trackera dziedziczy po klasie Service. Po każdej otrzymanej lokalizacji położenie jest zapisywane do tabeli, która odpowiada obecnemu przejazdowi. Każda taka tabela pojedynczego przejazdu wiąże się z tabelą zawierającą wszystkie przejazdy za pomocą klucza. Tabela wszystkie przejazdy umożliwia wyświetlenie ich na liście.



Rysunek 7: Widoki części trackera

2.2.5 Aparat i zapisywanie zdjęć pod wskazanym adresem w pamięci urządzenia

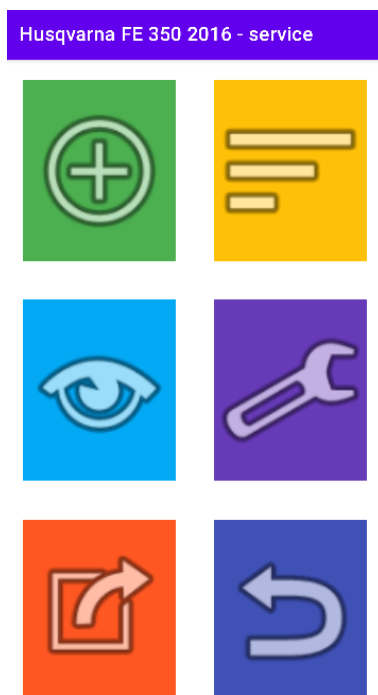
Aplikacja umożliwia szybki dostęp do aparatu i robienie powtarzalnych zdjęć oraz zapisywanie ich w pamięci urządzenia w wydzielonym folderze.



Rysunek 8: Widoki aparatu

Pozostałe elementy We fragmencie garażu można ponadto usunąć dany pojazd, przenieść go do serwisu lub wyjść do głównego menu.

2.2.6 Widok menu serwisu



Rysunek 9: Widoki menu serwisu

- Dodanie czynności serwisowej,
- Wyświetlenie wszystkich czynności i możliwe ich udostępnienie osobie zainteresowanej
- Dodawanie powiadomień
- Wyświetlenie noty serwisowej zapisanej lokalnie,
- Wysłanie pojazdu z powrotem do garażu,
- Powrót do spisu pojazdów.


2.2.7 Dodawanie nowej czynności serwisowej

SingleMotoActivity

Add current service

Name _____







Description _____



Rysunek 10: Dodawanie czynności serwisowej

Kiedy pojazd znajduje się w strefie serwisu, możliwe jest wykonanie na nim czynności, którą można nazwać, opisać i dodać do lokalnej bazy wszystkich napraw.

2.2.8 Wyświetlanie wszystkich dotychczasowych napraw z możliwością udostępnienia przez e-mail

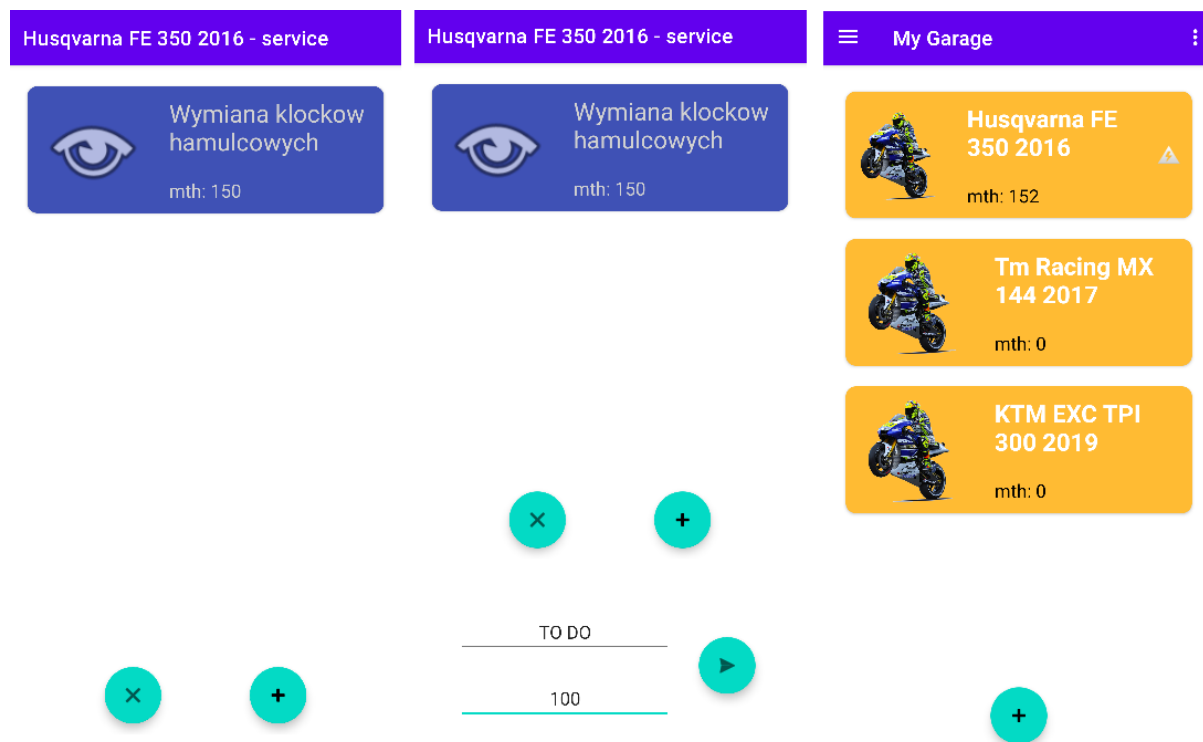
Husqvarna FE 350 2016 - service	Husqvarna FE 350 2016 - service	Husqvarna FE 350 2016 - service
 <p>Wymiana oleju</p> <p>25-Jan-2021</p>	 <p>Wymiana oleju</p> <p>25-Jan-2021</p> <p>Wymieniono olej na Belray SAE 80 odpowiednik Motul 10W80 przekładniowy</p> <p>DELETE</p>	 <p>Wymiana oleju</p> <p>25-Jan-2021</p> <p>Wymieniono olej na Belray SAE 80 odpowiednik Motul 10W80 przekładniowy</p> <p>DELETE</p>
 <p>Nowe sprzęgło</p> <p>25-Jan-2021</p>	 <p>Nowe sprzęgło</p> <p>25-Jan-2021</p> <p>Wymiana tarcz oraz kosza sprzęgłowego</p> <p>DELETE</p>	 <p>Nowe sprzęgło</p> <p>25-Jan-2021</p>

Rysunek 11: Widok panelu z wszystkimi naprawami

Interaktywny widok umożliwia rozszerzenie komponentów i wywołanie metody usuwania danej czynności serwisowej. Ponadto przycisk udostępniania daje możliwość otworzenia layoutu, przez który wprowadza się adres e-mail do wysyłki. Taka funkcjonalność przydatna jest szczególnie przy sprzedaży pojazdu. Udostępnić tym samym można całą historię remontów.

2.2.9 Ustawianie i zarządzanie powiadomieniami

Aplikacja wprowadza możliwość ustawienia powiadomień o potrzebie wykonania czynności serwisowej przy osiągnięciu wskazanej w powiadomieniu liczby motogodzin przez dany pojazd.



Rysunek 12: Widok panelu z wszystkimi naprawami

Kiedy wskazana liczba motogodzin zostanie przekroczona, wówczas użytkownik zostanie powiadomiony sygnałem ostrzegawczym przy pozycji pojazdu.

2.2.10 Wyświetlanie lokalnego pliku PDF

Bardzo ważną funkcjonalnością jest możliwość posiadania noty serwisowej, która jest niezbędna przy wszystkich pracach serwisowych.



Rysunek 13: Wyświetlanie pliku PDF z notą serwisową

Pozostałe funkcje w menu służą do oddania pojazdu z powrotem do garażu, bądź wyjścia do głównego spisu pojazdów.

2.3 Animacje

Do projektu dodano animowany pokaz slajdów jako intro aplikacji. Ponadto animowane są przejścia pomiędzy fragmentami. Te zostały wybrane spośród dostępnych w standardowej bibliotece.

2.4 Uprawnienia

Przed czynnościami wymagającymi uzyskanie uprawnień użytkownik jest pytany, czy wyraża zgodę. Wszystkie zapytania są kierowane do użytkownika bezpośrednio przed użyciem np. kamery,

Konieczne uprawnienia do prawidłowego działania wszystkich funkcji aplikacji

- `<uses-permission android:name=android.permission.WRITE_EXTERNAL_STORAGE"/>`
- `<uses-permission android:name=android.permission.READ_EXTERNAL_STORAGE"/>`
- `<uses-permission android:name=android.permission.INTERNET"/>`
- `<uses-permission android:name=android.permission.ACCESS_NETWORK_STATE"/>`
- `<uses-permission android:name=android.permission.ACCESS_FINE_LOCATION"/>`
- `<uses-permission android:name=android.permission.ACCESS_COARSE_LOCATION"/>`
- `<uses-permission android:name=android.permission.CAMERA"/>`

3 Responsywność interfejsu graficznego

Każdy widok projektu został przetestowany pod względem dostosowania się do orientacji oraz wielkości ekranu. Tam gdzie było to konieczne utworzono widoki wspomagające i ustawiające się w momencie zmiany wielkości ekranu bądź orientacji, Nie napotkano na żadne problemy z responsywnością i skalowalnością widoków.

4 Podsumowanie

W ramach projektu zrealizowano wstępne założenia. Napotkano problem zapisu bezpośrednio na karcie SD. Z tego względu obecny zapis odbywa się do pamięci wewnętrznej urządzenia. Ponadto udostępnianie informacji o pracach serwisowych zdecydowano się realizować w oparciu o wysyłanie wiadomości e-mail do wskazanego użytkownika. Wszystkie inne funkcjonalności są zaimplementowane w sposób umożliwiający ich użytkowanie. Z perspektywy osoby, która spotkała się z problemami, które aplikacja rozwiązuje (np. niedostępność not serwisowych, braku jednego miejsca na gromadzenie danych, brak systematyczności w prowadzonych pracach), rozwiązanie wydaje się być użyteczne. Aplikacja może być w łatwy sposób rozszerzona o inne pojazdy np. samochody osobowe, a tym samym możliwy jest jej potencjalny rozwój.

Literatura

- [1] Dr inż. Marek Piasecki. Wykłady, 2020/2021.
- [2] Techotopia. Android studio development essentials.