POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

KIERUNEK: INFORMATYKA (INF)

SPECJALNOŚĆ: INŻYNIERIA SYSTEMÓW INFORMATYCZNYCH (INS)

PRACA DYPLOMOWA

INŻYNIERSKA

Aplikacja mobilna wspomagająca zarządzanie kosztami eksploatacji pojazdów

Mobile application supporting management of vehicles operating costs

AUTOR:

Jakub Zagrobelny

PROWADZĄCY PRACĘ:

dr inż. Jarosław Mierzwa, Wydział Elektroniki

OCENA PRACY:

WROCŁAW, 2016

**Spis treści**

[Spis rysunków 5](#_Toc472072384)

[Spis tabel 6](#_Toc472072385)

[Spis listingów 7](#_Toc472072386)

[Skróty 8](#_Toc472072387)

[1. Wprowadzenie 9](#_Toc472072388)

[1.1. Wstęp 9](#_Toc472072389)

[1.2. Cel i zakres pracy 9](#_Toc472072390)

[2. Istniejące rozwiązania 10](#_Toc472072391)

[2.1. Fuelio 10](#_Toc472072392)

[2.1.1. Opis funkcji 10](#_Toc472072393)

[2.1.2. Przykładowe widoki 11](#_Toc472072394)

[3. Wymagania funkcjonalne i niefunkcjonalne 12](#_Toc472072395)

[3.1. Wymagania funkcjonalne 12](#_Toc472072396)

[3.1.1. Diagram i opis przypadków użycia 12](#_Toc472072397)

[3.2. Wymagania niefunkcjonalne 19](#_Toc472072398)

[3.2.1. Wykorzystane technologie i narzędzia 19](#_Toc472072399)

[3.2.2. Wymagania efektywnościowe i niezawodnościowe 20](#_Toc472072400)

[3.2.3. Wymagania dotyczące bezpieczeństwa 20](#_Toc472072401)

[4. Użyte technologie i narzędzia 21](#_Toc472072402)

[4.1. Android Studio 21](#_Toc472072403)

[4.2. Papyrus 21](#_Toc472072404)

[4.3. SQLite 21](#_Toc472072405)

[4.4. MPAndroidChart 21](#_Toc472072406)

[5. Projekt 22](#_Toc472072407)

[5.1. Projekt bazy danych 22](#_Toc472072408)

[5.2. Architektura aplikacji 22](#_Toc472072409)

[5.3. Interfejs graficzny 22](#_Toc472072410)

[5.3.1. Fragment główny 22](#_Toc472072411)

[5.3.2. Fragment dodawania kosztu 22](#_Toc472072412)

[5.3.3. Fragment statystyk 22](#_Toc472072413)

[5.4. Diagram klas 22](#_Toc472072414)

[6. Implementacja 23](#_Toc472072415)

[6.1. Opis kluczowych klas aplikacji 23](#_Toc472072416)

[6.1.1. Cost 23](#_Toc472072417)

[6.1.2. Vehicle 23](#_Toc472072418)

[6.1.3. Settings 23](#_Toc472072419)

[6.2. Algorytmy obliczeniowe 24](#_Toc472072420)

[6.2.1. Prognoza zasięgu pojazdu 24](#_Toc472072421)

[6.2.2. Obliczanie całkowitego kosztu kilometra 24](#_Toc472072422)

[6.2.3. Średnie spalanie 24](#_Toc472072423)

[6.3. Komunikacja z bazą danych 24](#_Toc472072424)

[6.3.1. Utworzenie bazy 24](#_Toc472072425)

[6.3.2. Nawiązanie połączenia z bazą danych 24](#_Toc472072426)

[6.3.3. Pobranie danych wybranego pojazdu 24](#_Toc472072427)

[6.3.4. Pobranie kosztów z ostatnich 30 dni 24](#_Toc472072428)

[6.3.5. Dodanie nowego kosztu 24](#_Toc472072429)

[6.3.6. Aktualizacja wybranego kosztu 24](#_Toc472072430)

[6.3.7. Usunięcie wybranego kosztu 24](#_Toc472072431)

[6.3.8. Dodanie pojazdu 25](#_Toc472072432)

[6.3.9. Usunięcie pojazdu 25](#_Toc472072433)

[6.3.10. Aktualizacja ustawień aplikacji 25](#_Toc472072434)

[6.3.11. Pobranie danych do eksportu 25](#_Toc472072435)

[6.3.12. Załadowanie danych zaimportowanych 25](#_Toc472072436)

[7. Testy aplikacji 26](#_Toc472072437)

[7.1. Testy jednostkowe z wykorzystaniem Mockito 26](#_Toc472072438)

[7.1.1. Opis wykonywanych testów 26](#_Toc472072439)

[7.1.2. Wnioski z testów 26](#_Toc472072440)

[7.2. Testy manualne na emulatorze 26](#_Toc472072441)

[7.2.1. Opis wykonywanych testów 26](#_Toc472072442)

[7.2.2. Wnioski z testów 26](#_Toc472072443)

[7.3. Testy manualne na urządzeniu mobilnym 26](#_Toc472072444)

[7.3.1. Opis wykonywanych testów 26](#_Toc472072445)

[7.3.2. Wnioski z testów 26](#_Toc472072446)

[8. Podsumowanie 27](#_Toc472072447)

[8.1. Wnioski 27](#_Toc472072448)

[8.2. Możliwości rozwoju 27](#_Toc472072449)

[9. Literatura i źródła 28](#_Toc472072450)

[10. Załącznik - instrukcja 29](#_Toc472072451)

# Spis rysunków

[Rysunek 1 Widok strony głównej 11](#_Toc471222961)

[Rysunek 2 Widok statystyk 11](#_Toc471222962)

[Rysunek 3 Diagram przypadków użycia 12](#_Toc471222963)

# Spis tabel

[Tab. 1. Przykład podpisu tabeli 3](#_Toc465685652)

# Spis listingów

[Listing. 1. Początkowe żądanie HTTP 21](#_Toc471160001)

# Skróty

**OGC** (ang. *Open Geospatial Consortium*)

**XML** (ang. *eXtensible Markup Language*)

**SOAP** (ang. *Simple Object Access Protocol*)

**WSDL** (ang. *Web Services Description Language*)

**UDDI** (ang. *Universal Description Discovery and Integration*)

**GIS** (ang. *Geographical Information System*)

**SDI** (ang. *Spatial Data Infrastructure*)

**ISO** (ang. *International Standards Organization*)

**WMS** (ang. *Web Map Service*)

**WFS** (ang. *Web Feature Service*)

**WPS** (ang. *Web Processing Service*)

**GML** (ang. *Geography Markup Language*)

**SRG** (ang. *Seeded Region Growing*)

**SOA** (ang. *Service Oriented Architecture*)

**IT** (ang. *Information Technology*)

1. Wprowadzenie
   1. Wstęp

Rozwój technologii komputerowych rozpoczynał się od ogromnych komputerów zajmujących powierzchnię nawet kilku pokoi, a obecnie urządzenia kilkaset razy mniejsze i jednocześnie kilkaset razy bardziej wydajne znajdują się w kieszeni prawie każdego człowieka. Popularność technologii mobilnych sprawiła, że człowiek może w każdej chwili wyszukać informację w sieci lub skorzystać z pomocnej aplikacji. Właśnie ta powszechność smartfonów, a także fakt niezadowolenia z dostępnych na rynku aplikacji wspomagających prowadzenie budżetu pojazdu, wpłynęła na mój wybór tematu pracy inżynierskiej.

* 1. Cel i zakres pracy

Celem pracy jest stworzenie aplikacji mobilnej działającej na smartfonach z systemem Android oraz udokumentowanie kolejnych etapów, czyli projektowania, implementacji oraz testowania. Celem pracy jest także poznanie podstaw technologii programowania na system Android w oparciu o język Java, a także zaznajomienie się ze środowiskiem programowania i jego narzędziami.

Niniejsza praca zawiera dokumentację projektu aplikacji mobilnej, opis jej testów oraz instrukcję obsługi. Na pracę składa się dziesięć rozdziałów. Pierwszy rozdział zawiera wstęp pracy. W rozdziale drugim znajduje się przedstawienie trzech już istniejących rozwiązań. Rozdziały od drugiego do piątego zawierają projekt techniczny aplikacji, czyli wymagania funkcjonalne i niefunkcjonalne, omówienie technologii oraz sam projekt struktury aplikacji oraz jej funkcji. Rozdziały szósty i siódmy omawiają implementację oraz testowanie aplikacji. W rozdziale ósmym jest podsumowanie projektu, a w dziewiątym rozdziale znajduje się wykorzystana literatura. Ostatni rozdział zawiera opis zawartości załączonej do pracy płyty CD oraz instrukcję instalacji oraz korzystania z aplikacji.

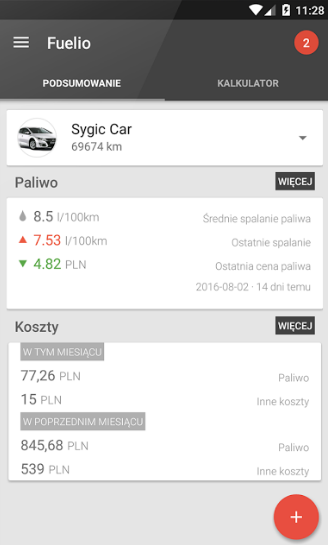
1. Istniejące rozwiązania

W niniejszym rozdziale przedstawię trzy wybrane popularne aplikacje zmierzające się tematem podobnym do mojego, dostępne w Sklepie Play. Na końcu podsumuję, które elementy poszczególnych aplikacji znajdą się w mojej aplikacji, a których nie będzie oraz co nowego wprowadzam, co nie znajduje się w omówionych rozwiązaniach.

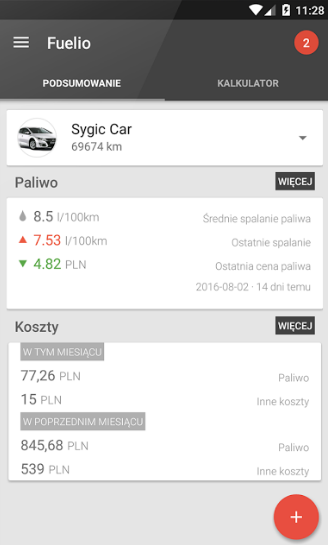
* 1. Fuelio
     1. Opis funkcji

Jest to aplikacja oparta na stylu aplikacji Material Art promowanym przez twórcę systemu Android, czyli firmę Google. Zawiera następujące funkcjonalności:

1. Wgląd w statystyki liczbowe i wykresy dla osobno tankowań oraz innych kosztów podzielonych na kategorie.
2. Wgląd osobno w dziennik tankowań i dziennik kosztów.
3. Generowanie raportów zawierających dane wybrane przez użytkownika.
4. Import i Eksport manualny lub automatyczny przy wykorzystaniu Google Drive, Dropbox albo lokalnie pliku CSV.
5. Kalkulator kosztów podróży.
6. Możliwość wyboru jednostek miar i waluty, języka, formatu daty, motywu kolorystycznego.
7. Opiniowanie stacji benzynowych z wglądem na ich lokalizację na mapie.
8. Wgląd w zestawienie tankowań na mapie ze względu na lokalizację stacji benzynowej.
9. Przypomnienia o przedłużeniu ubezpieczenia, wykonaniu przeglądu rejestracyjnego lub innego zdefiniowanego kosztu.
10. Możliwość dodawania wielu pojazdów i wyboru dla każdego osobnej formy zasilania samochodu, ilości źródeł, a także jednostek odległości, pojemności i średniego spalania.
    * 1. Przykładowe widoki



Rysunek 1 Widok strony głównej



Rysunek 2 Widok statystyk

1. Wymagania funkcjonalne i niefunkcjonalne
   1. Wymagania funkcjonalne

Aplikacja realizowana w ramach projektu będzie posiadała następujące funkcjonalności:

1. Zbieranie danych o kosztach związanych z eksploatacją danego pojazdu, danych samego użytkownika, jego pojazdów oraz preferencji dotyczących wyglądu aplikacji.
2. Możliwość dodania kilku pojazdów i przechowywania dla nich osobnych danych.
3. Analiza danych dotyczące kosztów i wyświetlanie statystyk.
4. Przeglądanie i edycja dodanych kosztów.
5. Eksportu do pliku stanu całej bazy danych w dowolnym momencie.
6. Import stanu bazy danych i zastąpienie obecnego.
7. Możliwość uruchomienia samouczka dla bieżącego widoku w jego wejściowej formie, który w krótki sposób przedstawia elementy danego widoku oraz ich działanie lub zawartość.
   1. Wymagania niefunkcjonalne
      1. Wymagania efektywnościowe i niezawodnościowe

Główne założenia efektywnościowe i niezawodnościowe aplikacji mobilnej realizowanej w ramach projektu:

1. Przystosowanie do wyświetlania na smartfonach z systemem mobilnym Android w wersji 4.1 (API 16) i wyższej. Aplikacja nie będzie przystosowana do wyświetlania na tabletach, więc uruchomienie na urządzeniu tego typu może spowodować nieestetyczne lub nieczytelne przeskalowanie elementów – dotyczy szczególnie tabletów z domyślnym trybem poziomym.
2. Prostota i intuicyjność widoków aplikacji umożliwiające sprawne zapoznanie się z aplikacją i wygodne korzystanie z niej – w przypadku trudności dostępne będą krótkie samouczki przedstawiające bieżący widok.
3. Uruchomienie aplikacji powinno trwać możliwie krótko - nie dłużej niż 10 sekund.
4. Przechodzenie pomiędzy widokami, ładowanie wykresów oraz powinno być płynne i szybkie.
5. Import i eksport powinien odbywać się szybko i bez błędów.
   * 1. Wymagania dotyczące bezpieczeństwa

Aplikacja ze względu na przechowywane dane będzie posiadał szyfrowaną bazę danych, a także archiwum zip powstałe w wyniku eksportu danych. Wejście do aplikacji nie będzie wymagało hasła, ale użytkownik może wykorzystać zewnętrzne aplikacji do hasłowania dostępu do aplikacji oraz szyfrowane odblokowanie telefonu.

Aplikacja nie będzie nigdzie wysyłać danych podawanych przez użytkownika bez jego zgody i potwierdzenia.

1. Użyte technologie i narzędzia
   1. Android Studio

Zintegrowane środowisko programowania służące do tworzenia aplikacji mobilnych na system Android stworzone przez Google, oparte na popularnej platformie o nazwie IntelliJ. Środowisko udostępnia wbudowane i wspierające tworzenie aplikacji narzędzia takie jakie jak graficzny edytor widoków, zintegrowany emulator urządzeń mobilnych, z których domyślnie są dostępne emulatory urządzeń serii Nexus z możliwością pobrania emulatora dowolnego urządzenia z wybranym systemem, debbuger, a także lista ToDo do planowania realizacji projektu. Jest to jedno z najbardziej popularnych środowisk służących do programowania na system Android. Wybrałem to środowisko ze względu na jego popularność oraz wsparcie podczas programowania. Podczas implementacji aplikacji oraz testowania korzystałem z wersji Android Studio 2.2.2.

* 1. Papyrus

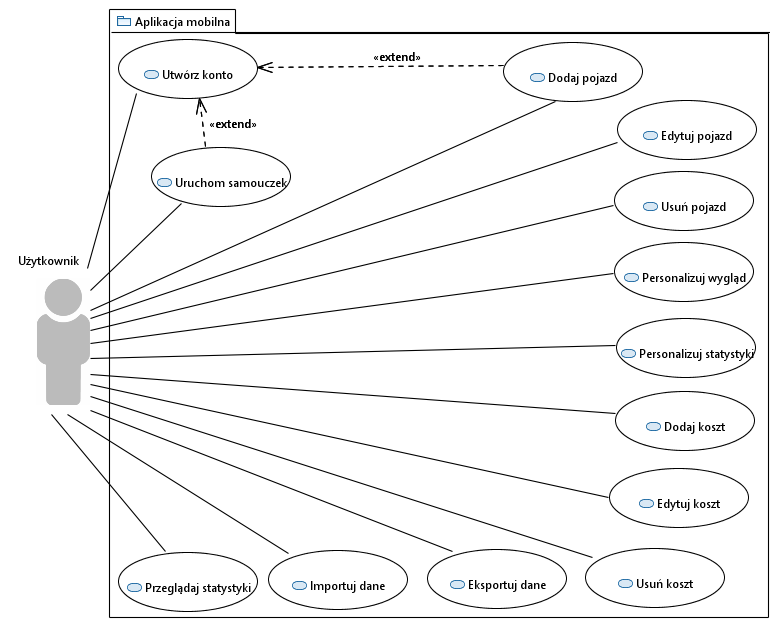
Środowisko służące do tworzenia diagramów UML oparte na środowisku programistycznym Eclipse. Wybrałem to środowisko ze względu na znajomość środowiska Eclipse oraz ze względu na czytelność. Podczas projektowania korzystałem z wersji Papyrus Neon (4.6.0).

* 1. SQLite
  2. MPAndroidChart

Biblioteka zawierająca gotowe klasy umożliwiające wygodne tworzenie estetycznych wykresów różnego typu tworzona przez Philippa Jahode. Wybrałem tę bibliotekę ze względu na wygląd wykresów, czytelną dokumentację oraz popularność. Podczas implementacji aplikacji korzystałem z wersji biblioteki MPAndroidChart v2.1.6.

1. Projekt
   1. Diagram i opis przypadków użycia

Funkcje aplikacji zostały zaprojektowane i przedstawione na poniższym diagramie przypadków użycia. Opis widocznych przypadków użycia znajduje się w dalszej części podrozdziału w osobnych tabelach.



Rysunek 3 Diagram przypadków użycia

Utwórz konto

|  |  |
| --- | --- |
| Kontekst zdaniowy | Pierwsze uruchomienie aplikacji i utworzenie bazy danych. |
| Warunki wstępne | Aplikacja nie została wcześniej uruchomiona lub nie istnieje baza danych aplikacji. |
| Warunek pomyślnego zakończenia | Utworzenie całej bazy danych przygotowanej pod dodawanie pojazdów wraz danymi użytkownika. |
| Warunek niepomyślnego zakończenia | Nieutworzenie bazy danych. |
| Aktor | Użytkownik. |
| Wyzwalacz | Wybór utworzenia nowego użytkownika przy pierwszym uruchomieniu aplikacji. |
| Główny przebieg | 1. Użytkownik wybiera opcję: utworzenie nowego konta przy pierwszym uruchomieniu aplikacji nowego pojazdu przy pierwszym uruchomieniu aplikacji.  2. Użytkownik wpisuje swój nick i zatwierdza.  3. Baza danych zostaje utworzona.  4. Opcjonalnie: Przejście do dodawania pojazdu.  5. Przejście do przypadku użycia **Uruchom samouczek** dla widoku głównego. |
| Rozszerzenie | 4. Po utworzeniu bazy jest możliwość dodania nowego pojazdu (**Extend::Dodaj pojazd**).  5. (**Extend::Uruchom samouczek**). |

Dodaj pojazd

|  |  |
| --- | --- |
| Kontekst zdaniowy | Dodanie nowego pojazdu, dla którego będzie prowadzona kontrola kosztów. |
| Warunki wstępne | Baza danych jest utworzona. |
| Warunek pomyślnego zakończenia | Dodanie pojazdu do bazy. |
| Warunek niepomyślnego zakończenia | Niedodanie pojazdu do bazy. |
| Aktor | Użytkownik. |
| Wyzwalacz | Potwierdzenie chęci dodania pojazdu po utworzeniu konta lub wybranie przycisku dodania nowego pojazdu w zakładce Pojazdy. |
| Główny przebieg | 1. Użytkownik wybiera dodanie nowego pojazdu przy pierwszym uruchomieniu aplikacji lub wybiera przycisk dodania nowego pojazdu w zakładce Pojazdy.  2. Użytkownik wpisuje wymagane dane pojazdu: markę, model, rodzaj paliwa/paliw i zatwierdza.  3. Pojazd zostaje dodany do tabeli pojazdy. |
| Rozszerzenie | Brak. |

Uruchom samouczek

|  |  |
| --- | --- |
| Kontekst zdaniowy | Uruchomienie samouczka aplikacji. |
| Warunki wstępne | Brak. |
| Warunek pomyślnego zakończenia | Poprawne wyświetlenie wszystkich ekranów samouczka i powrót do normalnego trybu użytkowania. |
| Warunek niepomyślnego zakończenia | Niewyświetlenie wszystkich ekranów samouczka lub brak powrotu do normalnego trybu użytkowania. |
| Aktor | Użytkownik. |
| Wyzwalacz | Utworzenie konta użytkownika przy pierwszym uruchomieniu aplikacji lub wybranie przycisku uruchomienia samouczka na górnym pasku aplikacji. |
| Główny przebieg | 1. Wyświetlenie jednego z ekranów samouczka.  2. Użytkownik klika w dowolny punkt ekranu.  3. Powtórzenie punktu 1 i 2 ilość razy zależną od widoku, w którym uruchomiono samouczek.  3. Powrót do normalnego trybu użytkowania. |
| Rozszerzenie | Brak. |

Edytuj pojazd

|  |  |
| --- | --- |
| Kontekst zdaniowy | Edytowanie danych pojazdu. |
| Warunki wstępne | Pojazd znajduje się w bazie danych. |
| Warunek pomyślnego zakończenia | Zaktualizowanie danych pojazdu w bazie danych. |
| Warunek niepomyślnego zakończenia | Niezaktualizowanie danych pojazdu w bazie danych. |
| Aktor | Użytkownik. |
| Wyzwalacz | Kliknięcie przycisku edycji znajdującego się przy nazwie pojazdu w zakładce Pojazdy. |
| Główny przebieg | 1. Użytkownik klika przycisk edycji znajdujący się przy nazwie pojazdu w zakładce Pojazdy.  2. Użytkownik edytuje wybrane dane pojazdu: markę, model, rodzaj paliwa/paliw, pojemność zbiornika/zbiorników i zatwierdza wprowadzone zmiany.  3. Pojazd zostaje zaktualizowany w bazie danych. |
| Rozszerzenie | Brak. |

Usuń pojazd

|  |  |
| --- | --- |
| Kontekst zdaniowy | Usunięcie danych pojazdu oraz danych z nim powiązanych z bazy danych. |
| Warunki wstępne | Pojazd znajduje się w bazie danych. |
| Warunek pomyślnego zakończenia | Poprawne usunięcie danych pojazdu oraz danych z nim powiązanych z bazy danych. |
| Warunek niepomyślnego zakończenia | Nieusunięcie z bazy danych wszystkich danych powiązanych z pojazdem. |
| Aktor | Użytkownik. |
| Wyzwalacz | Kliknięcie przycisku usunięcia znajdującego się przy nazwie pojazdu w zakładce Pojazdy oraz potwierdzenie chęci usunięcia w oknie dialogowym. |
| Główny przebieg | 1. Użytkownik klika przycisk usunięcia znajdującego się przy nazwie pojazdu w zakładce Pojazdy.  2. Użytkownik decyduje w oknie dialogowym  o eksporcie danych pojazdu przed jego usunięciem z bazy danych.  3. Użytkownik potwierdza w kolejnym oknie dialogowym chęć usunięcia pojazdu i danych.  4. Pojazd i dane z nim powiązane zostają usunięte z bazy danych. |
| Rozszerzenie | Brak. |

Personalizuj wygląd

|  |  |
| --- | --- |
| Kontekst zdaniowy | Personalizacja wyglądu aplikacji. |
| Warunki wstępne | Baza danych została założona. |
| Warunek pomyślnego zakończenia | Wprowadzenie zmian w wyglądzie aplikacji zadanych przez użytkownika. |
| Warunek niepomyślnego zakończenia | Niewprowadzenie zmian w wyglądzie aplikacji zadanych przez użytkownika. |
| Aktor | Użytkownik. |
| Wyzwalacz | Wybranie przycisku personalizacji wyglądu w zakładce Personalizacja. |
| Główny przebieg | 1. Użytkownik wybiera przycisk personalizacji wyglądu w zakładce Personalizacja.  2. Użytkownik wybiera z list rozwijanych dostępne parametry wyglądu aplikacji i zatwierdza zmiany.  3. Aplikacja zostaje uruchomiona ponownie w celu aktualizacji zmian.  4. Preferencje zostają zapisane w bazie danych |
| Rozszerzenie | Brak. |

Personalizuj statystyki

|  |  |
| --- | --- |
| Kontekst zdaniowy | Personalizacja wyświetlania statystyk. |
| Warunki wstępne | Baza danych została założona. |
| Warunek pomyślnego zakończenia | Wprowadzenie zmian w parametrach wyświetlania statystyk zadanych przez użytkownika. |
| Warunek niepomyślnego zakończenia | Niewprowadzenie zmian w parametrach wyświetlania statystyk zadanych przez użytkownika. |
| Aktor | Użytkownik. |
| Wyzwalacz | Kliknięcie przycisku personalizacji znajdującego się nad wybranym wykresem lub znajdującego się w zakładce Historia. |
| Główny przebieg | 1. Użytkownik klika przycisk personalizacji znajdujący się nad wybranym wykresem lub znajdujący się w zakładce Historia.  2. Użytkownik, w zależności od wyboru, ustawia zakres dat dla wykresu lub kategorie kosztów branych pod uwagę w zestawieniu.  3. Użytkownik zatwierdza zmiany.  4. Zamknięcie okna edycji. Wykres zostaje przerysowany w celu naniesienia zmian.  5. Preferencje zostają zapisane w bazie danych. |
| Rozszerzenie | 3.a. Użytkownik nie zatwierdza zmian.  4.a. Zamknięcie okna edycji.  5.a. Baza danych nie zostaje zaktualizowana. |

Dodaj koszt

|  |  |
| --- | --- |
| Kontekst zdaniowy | Dodanie nowego kosztu poniesionego z tytułu eksploatacji pojazdu. |
| Warunki wstępne | Baza danych została założona. |
| Warunek pomyślnego zakończenia | Dodanie kosztu do bazy danych. |
| Warunek niepomyślnego zakończenia | Niedodanie kosztu do bazy danych. |
| Aktor | Użytkownik. |
| Wyzwalacz | Krótkie lub długie kliknięcie przycisku „plus” widocznego w głównym widoku. |
| Główny przebieg | 1a. Użytkownik krótko klika przycisk plus (domyślny wybór kategorii dodawanego kosztu jako paliwo).  1b. Użytkownik długo klika przycisk plus (przejście do ekranu wyboru kategorii).  2. Użytkownik wpisuje wymagane wartości parametrów kosztu w zależności od wybranej kategorii i zatwierdza dodanie kosztu.  3. Koszt zostaje dodany do bazy danych. |
| Rozszerzenie | 1.b. Użytkownik wybiera jedną z wyświetlonych kategorii. |

Edytuj koszt

|  |  |
| --- | --- |
| Kontekst zdaniowy | Edycja parametrów wybranego kosztu. |
| Warunki wstępne | Baza danych została założona, koszt istnieje w bazie danych. |
| Warunek pomyślnego zakończenia | Zaktualizowanie parametrów kosztu w bazie danych. |
| Warunek niepomyślnego zakończenia | Niezaktualizowanie parametrów kosztu w bazie danych. |
| Aktor | Użytkownik. |
| Wyzwalacz | Wybranie kosztu z listy wyświetlanej w zakładce Historia, a następnie kliknięcie przycisku edycji. |
| Główny przebieg | 1. Użytkownik wybiera koszt z listy wyświetlanej w zakładce Historia, a następnie klika przycisk edycji.  2. Użytkownik zmienia wybrane wartości parametrów kosztu, które zależą od kategorii kosztu, a następnie zatwierdza zmiany.  3. Koszt zostaje zaktualizowany w bazie danych. |
| Rozszerzenie | 2.a. Użytkownik zmienia wybrane wartości parametrów kosztu, które zależą od kategorii kosztu, ale anuluje wprowadzone zmiany lub naciska przycisk wstecz.  3.a. Koszt nie zostaje zaktualizowany w bazie danych. |

Usuń koszt

|  |  |
| --- | --- |
| Kontekst zdaniowy | Usunięcie wybranego kosztu. |
| Warunki wstępne | Koszt znajduje się w bazie danych. |
| Warunek pomyślnego zakończenia | Usunięcie kosztu z bazy danych. |
| Warunek niepomyślnego zakończenia | Nieusunięcie kosztu z bazy danych. |
| Aktor | Użytkownik. |
| Wyzwalacz | Wybranie kosztu z listy wyświetlanej w zakładce Historia, a następnie kliknięcie przycisku usuwania. |
| Główny przebieg | 1. Użytkownik wybiera koszt z listy wyświetlanej w zakładce Historia, a następnie klika przycisk usunięcia.  2. Użytkownik potwierdza w oknie dialogowym chęć usunięcia kosztu.  3. Koszt zostaje usunięty z bazy danych. |
| Rozszerzenie | 2.a. Użytkownik anuluje usuwanie kosztu w oknie dialogowym.  3.a. Koszt nie zostaje usunięty z bazy danych. |

Eksportuj dane

|  |  |
| --- | --- |
| Kontekst zdaniowy | Eksport bazy danych do archiwum zip. |
| Warunki wstępne | Baza danych istnieje. |
| Warunek pomyślnego zakończenia | Wyeksportowanie wszystkich danych z bazy do archiwum zip. |
| Warunek niepomyślnego zakończenia | Niewyeksportowanie wszystkich danych z bazy do archiwum zip. |
| Aktor | Użytkownik. |
| Wyzwalacz | Wybranie przycisku Eksport w zakładce Import/Eksport. |
| Główny przebieg | 1. Użytkownik wybiera przycisku Eksport w zakładce Import/Eksport.  2. Użytkownik potwierdza eksport w oknie dialogowym.  3. Baza danych zostaje wyeksportowana do archiwum zip. |
| Rozszerzenie | 2.a. Użytkownik anuluje eksport w oknie dialogowym.  3.a. Baza danych nie zostaje wyeksportowana do archiwum zip. |

Importuj dane

|  |  |
| --- | --- |
| Kontekst zdaniowy | Import bazy danych z archiwum zip. |
| Warunki wstępne | Jest dostępne archiwum zip potrzebne do wykonania importu. |
| Warunek pomyślnego zakończenia | Zaimportowanie wszystkich danych z archiwum zip do bazy danych. |
| Warunek niepomyślnego zakończenia | Niezaimportowanie wszystkich danych z archiwum zip do bazy danych. |
| Aktor | Użytkownik. |
| Wyzwalacz | Wybranie przycisku Import w zakładce Import/Eksport lub wybranie opcji importu przy pierwszym uruchomieniu aplikacji. |
| Główny przebieg | 1. Użytkownik wybiera przycisku Import w zakładce Import/Eksport lub wybranie opcji importu przy pierwszym uruchomieniu aplikacji.  2. Użytkownik potwierdza import w oknie dialogowym z ostrzeżeniem, że wszystkie istniejące dane zostaną nadpisane.  3. Baza danych zostaje zaimportowana. |
| Rozszerzenie | 2.a. Użytkownik anuluje import w oknie dialogowym.  3.a. Baza danych nie zostaje zaimportowana. |

Przeglądaj statystyki

|  |  |
| --- | --- |
| Kontekst zdaniowy | Uruchomienie samouczka aplikacji. |
| Warunki wstępne | Brak. |
| Warunek pomyślnego zakończenia | Poprawne wyświetlenie wszystkich ekranów samouczka i powrót do normalnego trybu użytkowania. |
| Warunek niepomyślnego zakończenia | Niewyświetlenie wszystkich ekranów samouczka lub brak powrotu do normalnego trybu użytkowania. |
| Aktor | Użytkownik |
| Wyzwalacz | Potwierdzenie chęci dodania pojazdu po utworzeniu konta lub wybranie przycisku uruchomienia samouczka na górnym pasku aplikacji. |
| Główny przebieg | 1. Użytkownik wybiera dodanie nowego pojazdu przy pierwszym uruchomieniu aplikacji lub wybiera przycisk dodania nowego pojazdu w zakładce Pojazdy.  2. Użytkownik wpisuje wymagane dane pojazdu: markę, model, rodzaj paliwa/paliw i zatwierdza.  3. Pojazd zostaje dodany do tabeli pojazdy. |
| Rozszerzenie | Brak |

* 1. Projekt bazy danych
  2. Architektura aplikacji
  3. Interfejs graficzny
     1. Fragment główny
     2. Fragment dodawania kosztu
     3. Fragment statystyk
  4. Diagram klas

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

Rys. 1. Przykład podpisu rysunku

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Listing. 1. Początkowe żądanie HTTP

GET /script/Articles/Latest.aspx HTTP/1.1  
Host: www.codeproject.com  
Connection: keep –alive  
Cache -Control: max-age=0  
Accept: text/html ,application/xhtml+xml,application/xml|  
User -Agent: Mozilla/5.0 ...   
Accept -Language: en-US...  
Accept -Charset: windows -1251,utf -8...

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Przykład listy numerowanej:

1. A
2. Implementacja
   1. Opis kluczowych klas aplikacji
      1. Cost

asdfasdf

* + 1. Vehicle
    2. Settings
  1. Algorytmy obliczeniowe
     1. Prognoza zasięgu pojazdu
     2. Obliczanie całkowitego kosztu kilometra
     3. Średnie spalanie
  2. Komunikacja z bazą danych
     1. Utworzenie bazy
     2. Nawiązanie połączenia z bazą danych
     3. Pobranie danych wybranego pojazdu
     4. Pobranie kosztów z ostatnich 30 dni
     5. Dodanie nowego kosztu
     6. Aktualizacja wybranego kosztu
     7. Usunięcie wybranego kosztu
     8. Dodanie pojazdu
     9. Usunięcie pojazdu
     10. Aktualizacja ustawień aplikacji
     11. Pobranie danych do eksportu
     12. Załadowanie danych zaimportowanych

1. Testy aplikacji
   1. Testy jednostkowe z wykorzystaniem Mockito
      1. Opis wykonywanych testów
      2. Wnioski z testów
   2. Testy manualne na emulatorze
      1. Opis wykonywanych testów
      2. Wnioski z testów
   3. Testy manualne na urządzeniu mobilnym
      1. Opis wykonywanych testów
      2. Wnioski z testów
2. Podsumowanie
   1. Wnioski

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus. Aliquam erat volutpat. Integer maximus est turpis, ut bibendum ligula accumsan et. Ut eget vestibulum libero. Aliquam erat volutpat. Nullam placerat mauris a lectus tincidunt, et aliquet turpis aliquam. Etiam in malesuada lacus. Proin dignissim augue sit amet auctor elementum. Suspendisse potenti. Vivamus suscipit vulputate massa ac molestie. Suspendisse a justo porttitor, commodo mi at, placerat risus. Integer lobortis augue ac neque suscipit, vel sodales lacus fringilla.

* 1. Możliwości rozwoju

1. Literatura i źródła

[1] R. C. Martin. Czysty kod: Podręcznik Dobrego Programisty. Helion, Gliwice, 2014.

[2] Michał Gellert. Kurs programowania na platformę Android. https://www.youtube.com/playlist?list=PLTs20Q-BTEMNaj4UgOcQfSBsMvIH8bjuX.

[3] Damian Chodorek. Kurs Android. http://damianchodorek.com/category/kursy/android.

[4] Marco Vitas. Android Testing Tutorial: Unit Testing like a True Green Droid. https://www.toptal.com/android/testing-like-a-true-green-droid.

[5] Philipp Jahoda. MPAndroidChart. https://github.com/PhilJay/MPAndroidChart.

1. Załącznik - instrukcja

W załączniku znajduje się płyta CD wraz z wersją instalacyjną aplikacji w formacie \*.apk. Aby zainstalować aplikację na urządzeniu, należy skopiować plik instalacyjny do pamięci smartfonu, a następnie uruchomić go poprzez kliknięcie na niego i ewentualne wybranie aplikację, która przeprowadzi proces instalacji (zależy od urządzenia) oraz zatwierdzenie wymagań aplikacji odnośnie dostępu do elementów systemu (od wersji Androida 6.\* i wyższej zatwierdzenie dostępu występuje za pierwszym razem, gdy zostanie zgłoszona potrzeba skorzystania przez aplikację z danego elementu). Po kilkunastu sekundach instalacja będzie zakończona, więc będzie możliwość korzystania z aplikacji.