|  |  |
| --- | --- |
|  | **POLITECHNIKA ŁÓDZKA** |
|  | Wydział Elektrotechniki, Elektroniki, Informatyki  i Automatyki |
|  | Instytut Mechatroniki i Systemów Informatycznych |

Praca dyplomowa  
 magisterska

na temat:

**System na urządzenia mobilne wspomagający transport osób w miejscowościach turystycznych ( The system for mobile devices supporting the transport of people in tourist destinations** **)**

|  |  |
| --- | --- |
| Imię i Nazwisko: | **Viktor Kalashnykov** |
| Nr albumu: | **207 363** |
| Specjalność: | **Inteligentne Systemy Baz Danych** |
| Kierunek: | **Informatyka** |

Opiekun pracy:

Dr **Maciej Kacperski**

Łódź, styczeń 2019

Spis treści

[1 WSTĘP 3](#__RefHeading___Toc908_241114830)

[2 STAN WIEDZY 4](#__RefHeading___Toc552_2369495726)

[1.1 Rozwiązania znane dotychczas 4](#__RefHeading___Toc554_2369495726)

[1.2 System Operacyjny Android 7](#__RefHeading___Toc912_3645646421)

[1.3 Android Studio 8](#__RefHeading___Toc898_241114830)

[1.4 Java 9](#__RefHeading___Toc900_241114830)

[1.5 Serwis Google Maps 10](#__RefHeading___Toc556_2369495726)

[1.6 Google Firebase 12](#__RefHeading___Toc562_2369495726)

[3 SPECYFIKACJA PROJEKTU 13](#__RefHeading___Toc902_241114830)

[4 IMPLEMENTACJA PROJEKTU 14](#__RefHeading___Toc564_2369495726)

[4.1 Logowanie użytkownika 14](#__RefHeading___Toc690_1353847358)

[4.2 Moduł Kierowcy 14](#__RefHeading___Toc566_2369495726)

[4.2.1 Rejestracja nowego Kierowcy 14](#__RefHeading___Toc568_2369495726)

[4.2.2 Główny widok 17](#__RefHeading___Toc570_2369495726)

[4.2.3 Szczegóły Kierowcy 18](#__RefHeading___Toc572_2369495726)

[4.2.4 Dodanie nowej Trasy 20](#__RefHeading___Toc574_2369495726)

[4.2.5 Sterowanie Żądaniami od Klienta 22](#__RefHeading___Toc576_2369495726)

[4.2.6 Przepływ przejazdu Trasy przez Kierowcę oraz proces dowozu Klientów do Punktów Docelowych 25](#__RefHeading___Toc578_2369495726)

[4.3 Moduł Klienta 27](#__RefHeading___Toc580_2369495726)

[4.3.1 Rejestracja nowego Klienta 27](#__RefHeading___Toc582_2369495726)

[4.3.2 Główny widok dostępnych Tras 28](#__RefHeading___Toc584_2369495726)

[4.3.3 Szczegóły Klienta 30](#__RefHeading___Toc586_2369495726)

[4.3.4 Dodanie nowego Żądania do wybranego Kierowcy 31](#__RefHeading___Toc588_2369495726)

[4.3.5 Anulowanie Żądania przez Klienta 32](#__RefHeading___Toc590_2369495726)

[5 TESTOWANIE PROJEKTU 34](#__RefHeading___Toc904_241114830)

[6 WDRAŻANIE PROJEKTU 35](#__RefHeading___Toc906_241114830)

[7 PODSUMOWANIE 36](#__RefHeading___Toc592_2369495726)

[8 BIBLIOGRAFIA 37](#__RefHeading___Toc594_2369495726)

[9 SPIS RYSUNKÓW 38](#__RefHeading___Toc596_2369495726)

[10 STRESZCZENIE PRACY 39](#__RefHeading___Toc598_2369495726)

1 WSTĘP

Celem pracy dyplomowej jest przygotowanie aplikacji dla urządzeń na platfromę Android, która by umożliwiała przedstawicielom firm małego biznesu turystycznego oraz prywatnym kierowcom proponować swoje usługi dla prywatnych osób poprzez wbudowany mechanizm komunikacji. Klient może sprawdzić lokalizację autobusu, stan obciążenia oraz aktualną trasę. Także zadaniem była realizacja zgłaszania się klienta na przejazd od określonego przez niego punktu początkowego do punktu docelowego na mapie. Aplikacja miałaby ułatwić poszukiwanie klientów dla kierowcy.

W 2017 roku 18.9 mln. mieszkańców Polski (co stanowi 59 % ludności) uczestniczyli prawie w 45.9 mln podróży w kraju [1]. Najczęstszymi kierunkami są góry (Tatry, Sudety), morze Bałtyckie oraz miasta turystyczne, takie jak Wrocław, Kraków, Toruń, Zamość.

Rozwój technologii nie stoi nie miejscu i z przyjściem mobilnych platform, takich jak Android i  iOS, otwierają się nowe możliwości dla Polski. Z przychodem takich firm, takich jak Uber oraz BlaBlaCar, które wprowadzają innowacje w rynek przejazdu i zmieniają pogląd na sposoby interakcji przewoźników z klientami, podróżujący coraz chętniej korzystają z usług prywatnych firm.

Niestety, na rynku przewozu masowego pasażerów istnieje mało alternatyw. Branża jest zdominowana przez wielkie firmy (PKS), proponujący stałą trasę z brakiem możliwości dopasowania się do każdego indywidualnie w dowozie do miejsca docelowego. Także dla małych firm nie istnieje sprawdzonego i wygodnego kanału wyszukiwania oraz komunikacji z osobami chętnymi. Alternatywy, podobne do korzystania z platform OLX oraz BlaBlaCar, nie przynoszą odpowiednich wyników oraz nie mogą przedstawiać centralizowane rozwiązanie dla wyszukiwania podróżnika, śledzenia lokalizacji pojazdu oraz interakcji w czasie rzeczywistym z kierowcą. Dany projekt ma na celu zrewolucjonizować ten socjologiczny proces.

2 STAN WIEDZY

1.1 Rozwiązania znane dotychczas

Z rozwojem mobilnych technologii, przychodem coraz mocniejszego sprzętu większość zadań codziennych znachodzi nowe zaskakujące rozwiązania. Na rynku pojawia się coraz więcej aplikacji na mobilne urządzenia. Niektóre aplikacje potrzebują Internetu dla łączenia się z serwerem serwisów, zawierające jakieś dane niezbędne dla użytkownika oraz potrzebne dla prawidłowej logiki aplikacji (lokalizacja użytkownika, dane o pogodzie z serwisów prognozy pogody), a niektóre potrzebują tylko lokalnych danych (baza danych, pliki, pomiary z czujników takich, jak żyroskop, wilgotność). Aplikacje mobilne pomagają w organizacji życia, informują użytkownika oraz za pomocą nich firmy mogą pozyskiwać nowych klientów i komunikować się z osobami zainteresowanymi ofertą oraz pracownikami w bardziej wygodny sposób.

Do zalet rozwiązań mobilnych można zaliczyć: redukcja „pracy papierkowej” przez nadanie dostępu do formularzy, dokumentów handlowych czy kontraktów; krótszy czas obiegu dokumentów, zwiększenie satysfakcji klienta oraz lepsza komunikacja. [1]

W turystyce technologie informacyjne mają wszechstronne zastosowanie, jako w tym sektorze informacja ma największe znaczenie. Przez wprowadzenie technologii informacyjnych można uzyskać takie zalety, jak zwiększenie skuteczności, efektywności i jakości usług, diagnozowanie nowych rynków i produktów, a także monitorowanie istniejących, jak również tworzenie i dostarczanie produktu turystycznego [2]. To usprawnia obsługę klientów, pozwala na rozpowszechnienie informacji oraz ułatwić kontakty z przewoźnikami lub dostawcami, pośrednikami i klientami. Wielki zalety też są dla turystów którzy mogą szybciej, niż kiedykolwiek znachodzić potrzebną mu informację lub ob oferty od dostawcy (firm turystycznych, miejscowych i państwowych urzędów), dzięki nieograniczonemu dostępu do technologii oraz sieci Internet. Wielu firm lotniczych, hotelów i biur podróży za pomocą aplikacji oraz stron internetowych komunikują się z konsumentami w sprawie prezentacji produktu lub przeprowadzenia online rezerwacji. Także za pomocą technologii informacyjnych turyści mogą weryfikować oferty, proponowane przez firmy dostawców. Warto wspomnieć o portalach społecznościowych, gdzie turysty wymieniają się informacjami o ofertach i biurach. Ponadto technologie informacyjne, oraz mobilne technologie w szczególności, pozwalają na planowania rozwoju (dając wachlarz danych dla analizy oraz statystyk, niezbędnych dla weryfikacji na podstawie zdefiniowanego planu oraz strategii) i promocji (używając reklam i programów lojalnościowych w aplikacji). Dalej przedstawię informację o aplikację, które są używane przez różnych dostawców.

Pierwsza i najpopularniejszą ze wszystkich jest Street View od firmy Google, która oparta jest na platformie Google Maps oraz Google Earth. Za pomocą niego użytkownik może rozglądać się po wirtualnym świecie, oglądając panoramiczne zdjęcia wykonanych w wybranej lokalizacji oraz podróżować po nim w widoku „od pierwszej twarzy”. Obecnie aplikacja ma udostępnionych kilkaset obszarów w ponad 30 krajach i kontynentach. Służy jako promocja największych miast oraz miejsc turystycznych. Dla turystów daje możliwość sprawdzenia okolicy zarezerwowanego hotelu.

Google Earth to zaawansowany technicznie program dostępny dla każdego użytkownika-turysty bez żadnych opłat [3]. Na trójwymiarowym modelu kuli ziemskiej „z lotu ptaka” wyświetlana jest informacja o miejscach na Ziemi. Google Earth to dobry sposób dla promowania turystycznych przedsiębiorstw. Każda firma może oznaczyć się w miejscu, gdzie jest zlokalizowana. W tagu, przypisanym danemu miejscu, firma może podać informację, taką jak, na przykład, link do strony internetowej.

Kolejnym przykładem funkcji służącej rozwojowi internetowej turystyki są prezentacje multimedialne, zwane wirtualnymi wycieczkami, zawierające sferyczne panoramy wysokiej jakości [4]. Dzięki przedstawionej w taki sposób wydarzeń oraz turystycznego miejsca turysta weryfikuje wizję, dotyczącą danego zabytku za długo przed dokonaniem wycieczki. Bowiem czasami informacja, zawarta w ulotce od danego dostawcy oraz rekomendacja znajomych może okazać niewystarczająca dla osoby podróżującej. Wirtualna wycieczka może dać pewność, że miejsce albo wydarzenie jest warto odwiedzenia.

Google Goggles pozwala wyszukiwać informację o miejscach w Internecie za pomocą wizualizacji danych. Użytkownik robi zdjęcie aparatem fotograficznym, wbudowanym w smartfon, po analizie sfotografowanego obiektu , aplikacja łączy się z Internetem (serwisami Google), wprowadza zapytanie do wyszukiwarki i w ciągu kilku sekund zwraca wszystkie powiązane informacje dostępne na temat obiektu. Program oferuje również funkcję rozpoznawania tekstu, która umożliwia szybkie przetłumaczenie niezrozumiałych nazw [5].

Ostatnim przykładem aplikacji mobilnych jest tłumacz podróży, dzięki któremu bez problemu można załatwić wszystkie formalności związane z podróżą. Ze względu na możliwość tłumaczenia w czasie rzeczywistym brak znajomości obcego języka nigdy nie będzie stanowił bariery w komunikowaniu się podczas podróżowania [6].

Także wyposażenie telefonów w aparaty fotograficzne i kamery otworzyło aplikacje na wykorzystanie bardziej zaawansowanych funkcji, takich jak wykorzystanie kodów dwuwymiarowych (QR kody) co uwolniło turystę od wpisywanie długich adresów stron internetowych; oraz zastosowanie rozszerzonej rzeczywistości (ang. Augmented Reality, AR), czyli łączenie obrazu otrzymanego z kamery (zatem ze świata rzeczywistego) z elementami wykreowanymi przy wykorzystaniu technik informatycznych, sprawiającymi wrażenie rzeczywistych [7]. Telefony wyposażają się także w łączność bezprzewodową, na przykład BT (ang. Bluetooth) czy NFC (ang. Near Field Communication), co dodaje możliwość komunikowania się na bliskich odległościach. NFC, na przykład, jest szeroko używana do płatności zbliżeniowych oraz do wszelkich zastosowani, które potrzebują bezprzewodowego odczytu (synchronizacja z zewnętrznymi urządzeniami – np. głośnikami). Także popularne stają tz. Beacony – zewnętrzne urządzenia, komunikujące się ze smartfonem za pomocą BT oraz pełniące logikę biznesową danej aplikacji.

Bardzo popularnym rozwiązaniem w strefie turystyki są Audioprzewodniki. W takich systemach zapisane nagrania dotyczące konkretnych obiektów w różnych językach i formatach oraz zawierają treść opisu obiektu oraz wskazówek, takie jak „stojąc przed frontowym wejściem spójrz w lewo”. Takie rozwiązanie powszechnie są używane w muzeach (pierwsze wdrożenie miało miejsce w Amsterdamie w Stedelijk Museum w 1952 roku [8]).

Miejskie przewodniki mobilne (ang. city mobile guides, m-guides) to aplikacje mobilne, których głównym zadaniem jest ułatwienie wyszukiwania w trakcie wyjazdów turystycznych potrzebnych obiektów [9]. Mobilne przewodniki optymalizują czas wizyty turysty w mieście, oferują możliwość wyboru najbardziej interesujących atrakcji turystycznych z podaniem czasu ich zwiedzania, restauracje, hotele, ale także wydarzenia kulturalne. Zwykle udostępniają mapy i fotografie, czasami także materiały audio, pozwalają spojrzeć do rozkładów jazdy komunikacji miejskiej [10]. Dzięki GPS, też pozwalają na znalezienie najbliższego przystanku MPK. Takie aplikacje mogą być użyteczne nie tylko dla turystów, ale tez i mieszkańców.

Wśród polskich aplikacji dedykowanych turystyce warto przypomnieć audioprzewodniki miejskie takie, jak DIY Guide oferujący kilka tras po Krakowie, czy cykl Szlakiem warszawskich zabytków Visits in Poland dostępny w sklepie e-nexto, gdzie możliwy jest zakup całości lub wybranych nagrań. Inne – dostępne bezpłatnie – realizowane są na zlecenie administracji lokalnej, np. Audioprzewodniki po Poznaniu stowarzyszenia Poznańska Lokalna Organizacja Turystyczna (PLOT) powstały z inicjatywy Urzędu Miasta [11]. Kolejnym ważnym przykładem jest Droga Królewska, aplikacja, która dedykowana turystom niedowidzącym, wdrożona w Krakowie w roku 2011 na zlecenie Urzędu Miasta. W aplikacji jest przedstawiony zestaw z 12 nagrań w formacie MP3, zawierających opisy obiektów znajdujących się na jednej trasie zwiedzania, wraz z instrukcjami, jak dotrzeć do następnego punktu.

Następny przykład to aplikacja Odkoduj Łódź. W aplikacji mieści się 3 szlaki tematyczne, zawierające 2D kody, które użytkownik może zeskanować z zabytków na tych szlakach. Aplikacja funkcjonuje jako warstwa platformie rozszerzonej rzeczywistości Layar, udostępniona przez firmę MobileMS.

1.2 System Operacyjny Android

System Android nie potrzebuje przedstawienia. Jest to jeden z najpopularniejszych systemów na współczesne urządzenia mobilne (obok iOS i Windows Phone). Twórce danego systemu mieli intencję stworzenia zaawansowanego systemu dla kamer cyfrowych, jednak z wzrostem zapotrzebowań rynku na smartfony i, potem, tablety i innego rodzaju urządzeń mobilnych system był rozbudowany z uwzględnieniem preferencji użytkowników tych platform oraz jako konkurencję popularnym w tamte czasy platformy Symbian oraz Windows Mobile.

Android w wersji 1.0 po wyjściu na rynek mógł pochwalić się zunifikowaną witryną aplikacji, która była aplikacja Android Market (teraz Google Play), wbudowaną wyszukiwarka Google Search, aplikacją Gmail, synchronizacją kontaktów z serwisami Google oraz kalendarza z Google Calendar, dedykowaną aplikacją Google Maps z możliwością użycia API Google Directions oraz Street View, serwisem Google Talk Instant Messaging (teraz Google Hangouts) dla wymiany szybkimi wiadomościami, mechanizm notyfikacji przy zmianie w systemie oraz aplikacjach i aplikacja YouTube [12].

Aktualna wersja na czas pisania danego projektu (jest to 8.1 Oreo) zawiera takie funkcje, jak wskaźnik naładowania baterii urządzeń Bluetooth (czyli słuchawek, głośników oraz innych akcesoriów, które mogą być podłączone do smartfonu), który można przejrzeć z poziomu panelu z szybkimi ustawieniami, natywne wsparcie dla układów wspierających sztuczne sieci neuronowe (dedykowane jest urządzeniom, których procesory mają specjalne jednostki, odpowiedzialne za wykonywanie zadań związane ze sztucznymi sieciami neuronowymi, co aktywnie wykorzystuje się , na przykład, dla rozpoznawania twarzy w procesie logowania do urządzenia oraz dodaniu filtrów dla zdjęć), lepsze zarządzanie pamięcią RAM (system ma pobierać mniej pamięci dla aplikacji) oraz auto uzupełnianie formularzy na stronach internetowych oraz serwisach oraz program antywirusowy Google Play Protect. [13]

Dany projekt jest zorientowany na Android w wersji 4.2 (Jelly Bean), co pokrywa większość urządzeń, dostępnych na rynku (według wskazówek środowiska Android Studio jest to blisko 97 %).

1.3 Android Studio

Android Studio to oparte na IntelliJ kompletne środowisko (IDE) w którym zarówno początkujący jak i zaawansowani developerzy mogą pisać i projektować swoje aplikacje. [14]

Oprócz wsparcia dla języków programowania Java, C++ oraz Kotlin, narzędzi kompilacji i budowania kodu źródłowego w postaci pliku APK, który jest potrzebny dla instalacji gotowej aplikacji na testowych urządzeniach (wspierane są profile, takie jak profil Debug oraz Release, możliwe jest definiowanie własnego profilu) oraz archiwum AAR, zawierający kod źródłowy w postaci biblioteki, która może być użyta w kolejnych projektach, tworzonych zarówno na tym środowisku programistycznym, jak i na alternatywnych środowiskach (Eclipse, QT Creator, pakiet tworzenia aplikacji mobilnych Flutter) poprzez mechanizm linkowania (ang. link). Dla instalacji jest zintegrowane narzędzie przenoszenia, instalacji i uruchomienia aplikacji na urządzeniach do testowania. Na urządzeniach z Android API wersji 21 i wyżej (Android OS 5.0 +) jest wspierany mechanizm Instant Run, który ma na celu przyspieszenie budowania aplikacji, wykonując budowanie wyłącznie zmienionych części kodu źródłowego. Za proces budowania aplikacji w środowisku jest odpowiedzialne narzędzie Gradle, które zawiera zestaw skryptów, w którym w postaci DSL opisana jest poszczególna konfiguracja (minimalna wersja Android API, zestaw niezbędnych bibliotek i innych zależności, repozytorium kodu do zależności) oraz akcje, potrzebne dla budowania oraz przesłania paczki pod urządzenia z określonymi wersjami Android API.

Pobieranie narzędzi oraz rozszerzeń, pomagających dla implementacji zapotrzebowanej logiki biznesowej, dodatkowej funkcjonalności (użycie natywnych funkcjonalności do ABI – Aplikation Binary Interface, NDK Tools, pomagające w dostarczeniu kodu źródłowego, napisanym w języku C++, rozszerzenia do serwisów Google oraz Firebase SDK) jest proste. Do tego służy narzędzie Android SDK Manager. Za pomocą niego można też pobierać obrazy systemu Android z uwzględnieniem wersji Android API oraz architektury ( X86, ARM ). które są używane w wirtualnych urządzeniach, do przygotowania których służy AVD (Android Virtual Device) Manager.

AVD Manager jest używany w przypadku niezbędności testowania aplikacji w wirtualnym środowisku oraz w razie niemożliwości przetestowania aplikacji na danym modelu urządzenia od danej firmy (na przykład, w razie braku takiego urządzenia). W AVD Manager można skonfigurować wersję urządzenia (czyli model, taki jak Google Nexus 4) oraz wersję Android z odpowiedni poziomem API oraz architekturą, lub zdefiniować władne urządzenie ze swoją architekturą oraz technicznymi parametrami – ilość rdzeni CPU, RAM oraz rozdzielczość. AVD Manager pozwala na jednoczesne uruchomienie wielu wirtualnych urządzeń co znaczy, że nic nie stoi na przeszkodzie testować aplikację równolegle.

Dany projekt jest utworzony w IDE Android Studio wersji 3.3.1 z pluginem Gradle 3.3.1 oraz Build Tools w wersji 27.0.3.

Przetestowany projekt był na urządzeniu Honor 6X z wersją Android 7.0 Nougat oraz Nexus 4 z Android 5.0 Lollipop (wirtualne urządzeniu, utworzone w AVD Manager).

1.4 Java

Java jest obecnie jednym z najpopularniejszych języków programowania. Należy do grupy tzw. języków wysokiego poziomu, w których wiele rzeczy wykonywanych jest automatycznie i nie musi ich wykonywać programista. Początki Javy sięgają roku 1995 roku. Wtedy to grupa programistów pod kierownictwem Jamesa Goslinga, rozpoczęła pracę nad nowym obiektowym językiem. Inspiracją programistów stał się język C++. Był to dominujący język na platformę Windows w latach 90siątych. Do dziś cieszy się on wciąż ogromnym zainteresowaniem. Utworzenie Javy okazało się strzałem w dziesiątkę. Dzięki nowoczesnemu obiektowemu charakterowi, przenośności i niezależności od architektury, nowy język w krótkim czasie stał się hitem. [15]

Ważnym aspektem programowania obiektowego jest hermetyzacja. Hermetyzacja pozwala na ograniczenie dostępu do kodu źródłowego, niepowołanym obiektom. Szczególny nacisk położono tutaj na bezpośrednią modyfikację zmiennych składowych klasy.

Dziedziczenie przez wielu uważane jest za kluczowy element programowania obiektowego. Pozwala na tworzenie nowych klas w oparciu o istniejące klasy lub klasy abstrakcyjne. Dziedziczenie może być wykonywane wielokrotnie - zasadniczo nie ma żadnych ograniczeń.

Aby zdefiniować pojęcie polimorfizmu, musimy zrozumieć czym jest interfejs. Z pozoru jest to struktura, która przypomina klasę, jednak w odróżnieniu od klasy zawiera ona tylko nagłówki metod. Jak widać jest to podobna struktura do klasy abstrakcyjnej, z tym że tutaj żadna z metod nie może być wypełniona treścią.

Tworząc interfejsy wymuszono jest na programiście, który będzie implementował te interfejsy wypełnienie treścią zadeklarowanych metod. Dzięki temu można w określonej w grupie klas uzyskać te same metody, które będą wykonywać podobne działania.

Drugim czynnikiem, który zadecydował o dużej popularności Javy jest jej przenośność. Założenie języka jest proste: kod pisany jest raz i uruchamiany jest wszędzie. I tak właśnie jest w przypadku Javy. Środowisko uruchomieniowe Javy (bo tak właśnie tłumaczono jest rozwinięcie skrótu JRE), jest specjalną platformą, którą musi zainstalować każdy, kto chce skorzystać z aplikacji napisanych w Javie na swoim komputerze. Kod źródłowy napisany przez programistę jest kompilowany do kodu pośredniego nazywanego w Javie kodem bajtowym (z ang. byte-code). Właściwa kompilacja następuje na komputerze użytkownika za pomocą wbudowanej w JRE, wirtualnej maszyny Javy (z ang. JVM), którą można nazwać dla uproszczenia sprawy kompilatorem. Ponieważ operacja kompilacji dokonuje się w locie, ma to duży wpływ na wydajność Javy. Dlatego też mobilność kodu, wiąże się z wolniejszym jego działaniem.

Wszędzie tam gdzie możliwe jest zainstalowanie wirtualnej maszyny Javy, możliwe jest również korzystanie z Javy. Tak więc z aplikacjami napisanymi w tym języku można spotkać się miedzy innymi w telefonach komórkowych (tutaj nazywane midletami), palmfonach, palmtopach itp.

W projekcie został użyty JDK (Java Development Kit) w wersji 1.8. Dla zapełnienia nowej funkcjonalności oraz nowoczesnego wyglądu na starszych urządzeniach (z Android 4.2) użyta jest biblioteka Android Support v4 oraz Android Compat w wersji 27.0.0, Google Play Services w wersji 16.0.0 dla zapewnienia dostępu do serwisów Google Play, takich jak Maps, Places, Directions, Firebase Auth, Core, Database , UI z pakietu Firebase SDK w wersji 16.0.

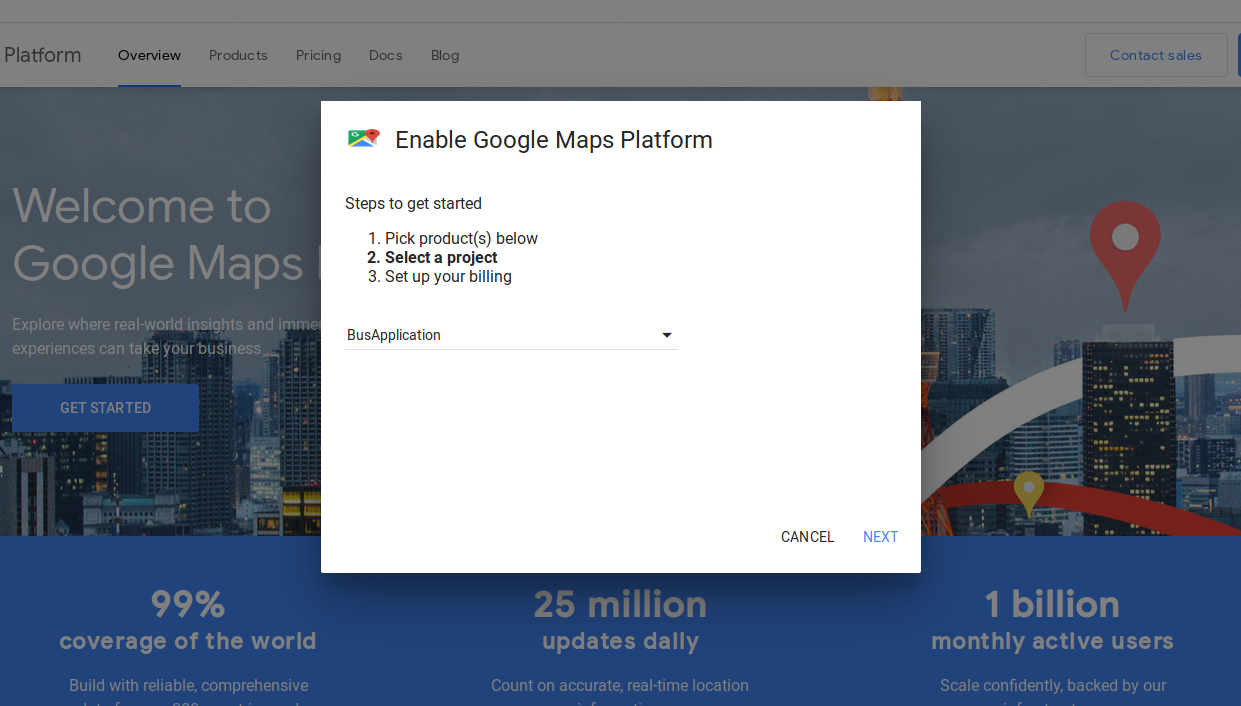
1.5 Serwis Google Maps

Google Maps to nowoczesna i zaawansowana usługa, oferowana pierwotnie wyłącznie w serwisie maps.google.com. Z czasem jej producent - firma Google - udostępniła pierwszą wersję API, która umożliwiała "osadzenie" mapy na dowolnej stronie internetowej. [16]

Obecnie API Google Maps dostępne jest w wersji stabilnej 3.x, oraz nierozwijanej 2. Pod względem możliwości znacznie przewyższa swoją pierwszą wersję, a mimo to jego obsługa jest bardzo prosta. Proste aplikacje to kwestia zaledwie paru linijek kodu - mechanizmy obsługujące mapę są po stronie Google, przez co autor mapy nie musi się o nie w ogóle troszczyć.

Interfejs map Google jest rozpoznawany przez miliony użytkowników stron Map Google - stworzone mapy są intuicyjne w obsłudze, mają szerokie pokrycie danymi kartograficznymi. Elementy interaktywne strony, takie jak na przykład mapa w Google Maps - są jednym z elementów trendu Web 2.0 - stron nowoczesnych, dynamicznych, przyciągających użytkowników.

Za pomocą Google Maps API jest możliwość wsadzenia widoku kartograficznego jak na stronach internetowych, tak i w natywnych klientach platform desktopowych (Windows, Linux, Mac ) i mobilnych rozwiązaniach (Android, iOS). Dla skorzystania z Google Maps na platformie Android wystarczy dodać projekt mapy na stronie [https://cloud.google.com/maps-platform/#get-started](https://cloud.google.com/maps-platform/" \l "get-started) (rys. 2.1) oraz wygenerować klucz API, który potem wystarczy dodać do projektu w środowisku Android Studio.



Rys. 2.1 Dodanie przykładowego projektu na Google Maps Platform [1]

Oprócz możliwością interakcji z Google Maps za pomocą natywnych klientów, dostarczanych poprzez Google Maps SDK, też komunikując się z REST API Google Maps.

W danym projekcie wykorzystany jest SDK do Google Maps w wersji 16.0.0

1.6 Google Firebase

Firebase to darmowe narzędzie od Google, które w swoim panelu zawiera kilkanaście funkcjonalności pomagającym developerom i marketerom w zarządzaniu aplikacją i jej monetyzacją. Firebase jest możliwy do zaimplementowania na systemach Unity, iOS, Android. Można z pewnością stwierdzić że jest następcą Google Analytics dla urządzeń mobilnych z dodatkowymi funkcjami. Narzędzie jest ciągle w fazie rozwoju. Posiada również pewne mankamenty, ale i tak otwiera wiele możliwości marketerom w szczególności tym, którzy korzystają z AdWords do promocji swojej aplikacji. W szczególności tym korzystając z zaawansowanych metod targetowania reklam wykorzystujących listy remarketingowe. Firebase ma również integrację z około 50 innymi sieciami reklamowymi, głównie działającymi na rynku amerykańskim. [17]

Oprócz funkcji dla analityki danych oraz zarządzania reklamami w aplikacji, Firebase dostarcza całkowity backend dla końcowego systemu klienckiego (w tym przypadku aplikacja Android). Między innymi w skład Firebase wchodzi Firebase Realtime Database – nowoczesna NoSQL w chmurze Google, zawierająca dokument, w którym znajdują się wejścia (wpisy). Każdy wpis może być dziedziczony od głównego katalogu ( „/” ) albo wpisu macierzystego. Do wpisów można odwoływać się w aplikacji za pomocą referencji. Użytkownicy danego systemu wyróżniają szybkość zapisywania oraz odczytu danych, który może trwać zaledwie setki milisekund, przez co jest bardzo użyteczny dla zarządzania tzw. „gorącymi” danymi (dane, które potrzebują częstej synchronizacji z backendem), którymi w danym projekcie są dane o lokalizacji użytkowników na mapach Google.

Szczególnej warto wspomnieć o Firebase Cloud Functions (w skrócie FCF), który stanowi backendową funkcjonalność dla aplikacji, dostarczając serwer na podstawie NodeJS instancji z możliwością dopisywania dodatkowej logiki w postaci „funkcji”, które są definiowane za pomocą języku JavaScript oraz pomocniczych narzędzi, za pomocą których można odwoływać się do innych serwisów Firebase (na przykład, Realtime Database) oraz móc zarządzać uruchomieniem tych funkcji (wysłanie do serweru FCF) oraz definiować w jaki moment musi być odpalana. Architektura FCF jest zorientowana na zdarzeniach, co oznacza, że można w łatwy sposób określić na co musi reagować serwer, żeby funkcja była uruchomiona. Można także skalować funkcję, uruchamiając je równolegle dla każdego użytkownika / zdarzenia i w tym momencie nowa instancja serwera z funkcją jest utworzona dla danego odpalenia funkcji.

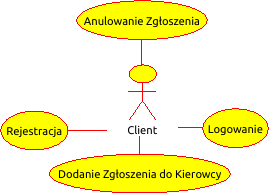
Dla realizacji mechanizmu autentykacji jest uzyte narzędzie Firebase Auth.

3 SPECYFIKACJA PROJEKTU

Dany projekt odpowiada następującym wymaganiom:

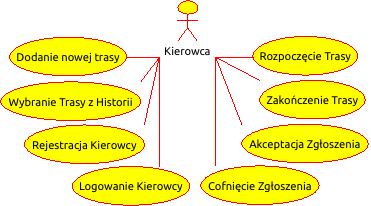
1. Realizacja interakcji między Kierowcą a Klientem przez aplikację oraz mechanizmu notyfikacji wewnątrz niej i Zgłoszenia, za pomocą których Klient i Kierowcą mogą wyświetlać szczegóły (lokalizacja początkowa oraz końcowa, zgłoszona przez Klienta).
2. Możliwość definicji Trasy przez Kierowcę w widoku mapy, używając serwisu Google Maps, w którym pierwszy wybiera po kolei punkty dla Trasy, a serwis Google Directions wylicza ścieżkę, dokonując jej optymalizacji pod kątem czasu przejazdu między poszczególnymi Punktami oraz obciążenia drogi (obecności korków na drogach).
3. Zapisanie Tras w bazie w historii przejazdów Kierowcy dla możliwości zobaczenie ich przez ostatniego w przyszłości oraz wyliczenie czasu przejazdu dla każdej z tras.
4. Użycie chmurowej architektury dla wolnego dostępu przez użytkowników do danych oraz oszczędzania kosztów w utrzymaniu własnych serwerów do autoryzacji i przechowywania danych i funkcjonalności.
5. Dopasowanie serwisu Firebase Auth, który służy dla autoryzacji użytkownika (Kierowcy lub Klienta). Dla autoryzacji używa się wbudowany mechanizm, który używa konto Google (domena gmail.com) oraz hasło dla weryfikacji informacji o użytkowniku.
6. Implementacja szybkiego dostępu do danych (zapisu, odczytu), co jest wbudowane w serwisie Firebase Realtime Database. W tej bazie, mianowicie, zapisane dane o bieżącej lokalizacji Kierowcy na mapie oraz Trasy, definiowane przez Kierowców i Zgłoszenia z aktualnym statusem.
7. Klient na głównym widoku w aplikacji widzi listę wszystkich Kierowców, który są partnerami danej aplikacji (czyli są zarejestrowane) oraz ich aktualne Trasy wraz ze szczegółami transportu, które włączają ilość wolnych miejsc, wolną przestrzeń bagażu w bagażniku oraz w miejsc w salonie na bagaż podręczny.

Klient ma następujące możliwości w aplikacji: rejestracja za pomocą serwisu Firebase Auth, logowanie do aplikacji dla dostępu do funkcjonalności, tworzenia Zgłoszenia do wybranego Kierowcy, anulowanie Zgłoszenia (rys. 3.1).



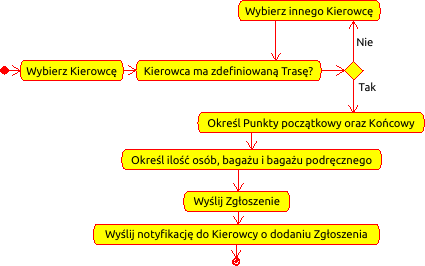
Rys. 3.1 Przypadki użycia dla Klienta [1]

Kierowca, w swoją kolej oprócz możliwości rejestrowania i logowania, dodaje nową Trasę, wybiera Trasę z historii, może zaakceptować lub wycofać zgłoszenie oraz rozpocząć jazdę Trasy i zadeklarować jej ukończenie (rys.3.2)



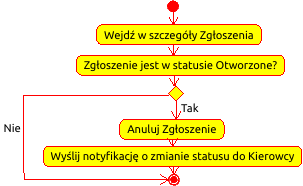
Rys. 3.2 Przypadki użycia dla Kierowcy [1]

Dla dodania Zgłoszenia Klientu trzeba wybrać Kierowcę, który ma zdefiniowaną Trasę dla przejazdu, dodać Punkt początkowy i Punkt końcowy Zgłoszenia, określić ilość osób, bagażu i bagażu podręcznego dla przejazdu i wysłać Zgłoszenie do wybranego Kierowcy, po czym Kierowca dostaje notyfikację o dodaniu nowego Zgłoszenia (rys.3.3).



Rys. 3.3 Przebieg tworzenia nowego Zgłoszenia przez Klienta [1]

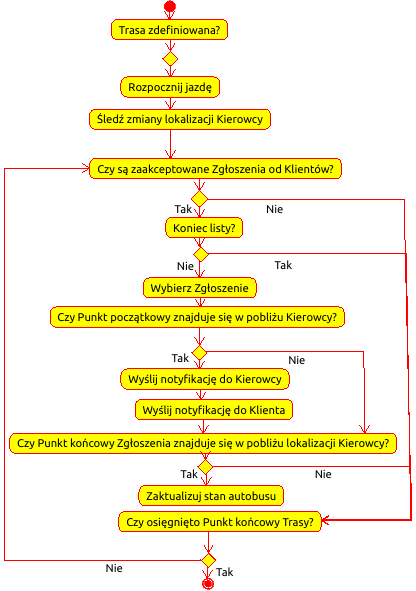
Dla anulowania Zgłoszenia Klientu wystarczy wejść w szczegóły Zgłoszenia oraz w przypadku otworzonego statusu Zgłoszenia Anulować go, po czym Kierowca dostaje notyfikację o anulowaniu (rys.3.4).



Rys. 3.4 Przebieg anulowania Zgłoszenia przez Klienta [1]

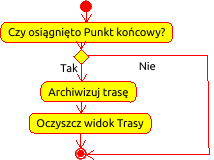
Cofnięcie i Akceptacja przez Kierowcę Zgłoszenia od Klienta wygląda podobnie jak wyżej wymieniony przypadek.

Przy rozpoczęciu Trasy przez Kierowcę status Trasy jest zmieniony na odpowiedni pod warunkiem, że Trasa jest zdefiniowana, potem rozpoczyna się śledzenie zmian lokalizacji Kierowcy na podstawie zaakceptowanych Zgłoszeń oraz aktualizacja szczegółów autobusu (ilość zajętych miejsc, bagażu oraz bagażu podręcznego w salonie) w przypadku osiągnięcia Punktu końcowego Zgłoszenia od Klienta, a także otrzymanie notyfikacji przez Klienta oraz Kierowcę, kiedy Kierowca znajduje się pobliżu Punktu początkowego Zgłoszenia (rys.3.5).



Rys. 3.5 Przebieg rozpoczęcia jazdy przez Kierowcę [1]

Przy zakończeniu Trasy przez Kierowcę, jest sprawdzany warunek, czy Punkt końcowy Trasy jest osiągnięty i w razie pozytywnej odpowiedzi Trasa jest archiwizowana (zapisywana do historii przejazdów) oraz widok Trasy jest oczyszczany (rys.3.6).

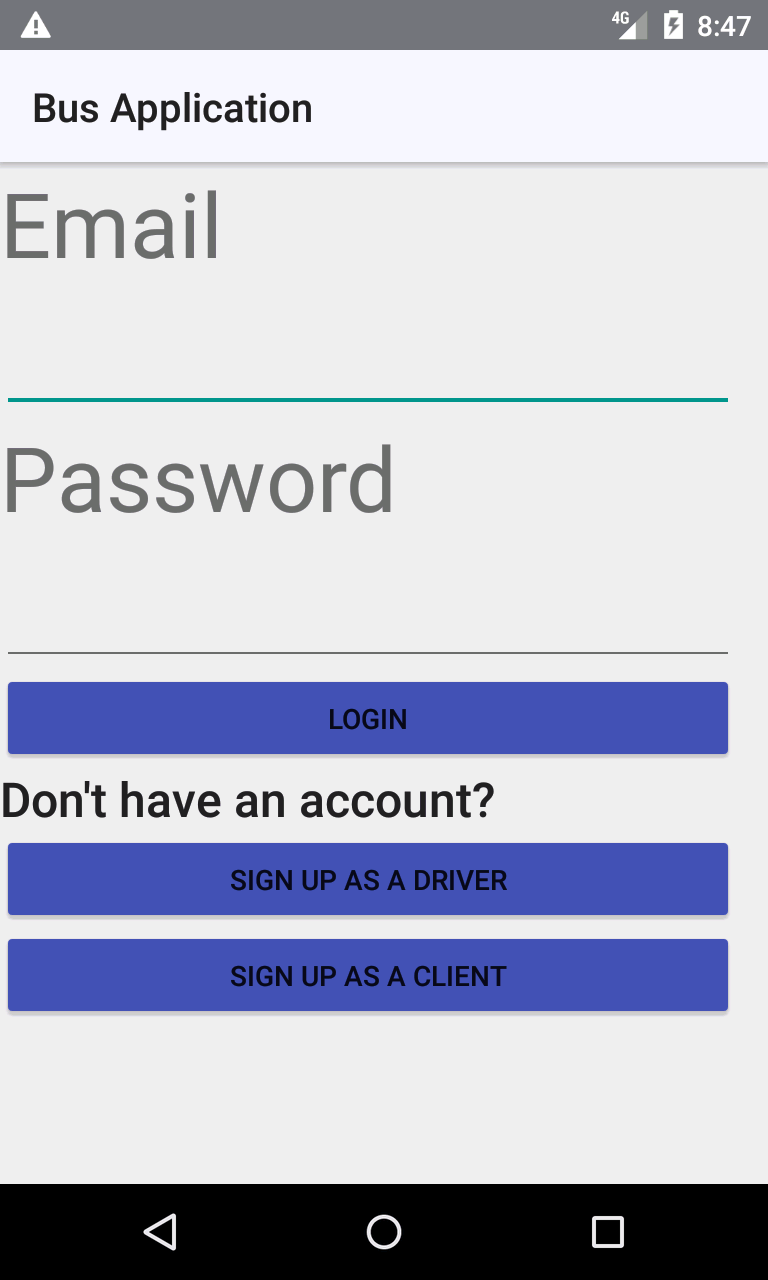


3.6 Przebieg zakończenia jazdy przez Kierowcę [1]

4 IMPLEMENTACJA PROJEKTU

4.1 Logowanie użytkownika

Przed Państwem widok logowania użytkownika (rys. 1.1)



Rys. 1.1 Widok logowania aplikacji [1]

Na nim zaprezentowany jest formularz logowania do modułów Kierowcy oraz Klienta. Informacja do którego modułu należy Użytkownik jest pobierana z listy Użytkowników bazy Firebase. Także w moment logowania pobierana jest lista atrybutów, która włącza szczegóły danych osobowych oraz żądania, a w przypadku Kierowcy dane o zdefiniowanych Trasach i szczegółach autobusu. O danych atrybutach można się dowiedzieć w następnych rozdziałach.

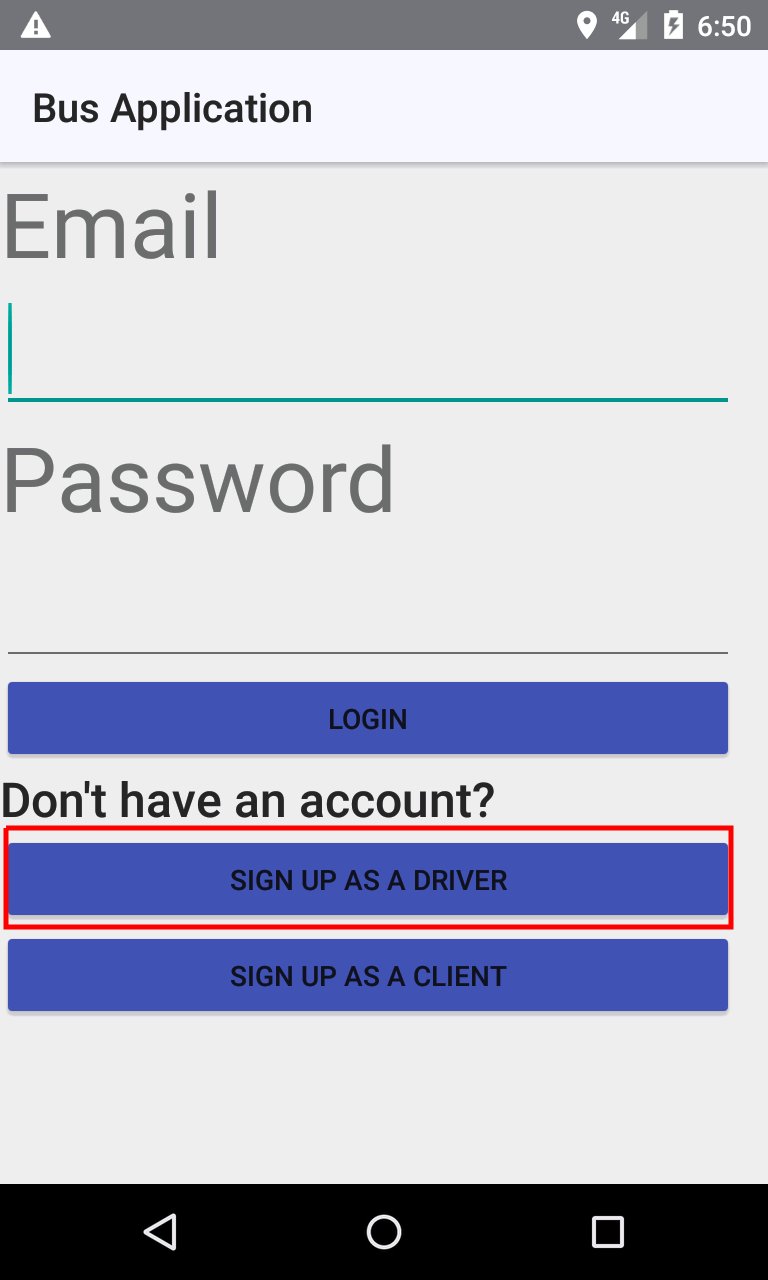
Także na danym widoku można zauważyć przyciski „Zarejestruj się jako Kierowca” oraz „Zarejestruj się jako Klient”, które przekierowują do strony Rejestracji Kierowcy (rozdział 3.2.1) oraz Rejestracji Klienta (rozdział 3.3.1 odpowiednio.

4.2 Moduł Kierowcy

4.2.1 Rejestracja nowego Kierowcy

Dla dostępu do możliwości aplikacji niezbędna jest autoryzacja ze strony użytkownika. Aplikacja po funkcjonalności i logice biznesowej dzieli się na moduły Kierowcy oraz Klienta. Proces rejestracji i logowania jest zrealizowany za pomocą systemu Firebase Auth, który jest częścią chmurowego serwisu Firebase. Wszystkie dane, podane pod czas rejestracji, są zapisywane w bazie Firebase Realtime Database w odpowiednich schematach (dla Kierowców to jednostka „drivers”, dla Klientów – „clients”).

Dla wejścia do strony rejestracji Kierowcy, trzeba wybrać z widoku Logowania punkt „Sign Up as Driver” (rys. 1.2).

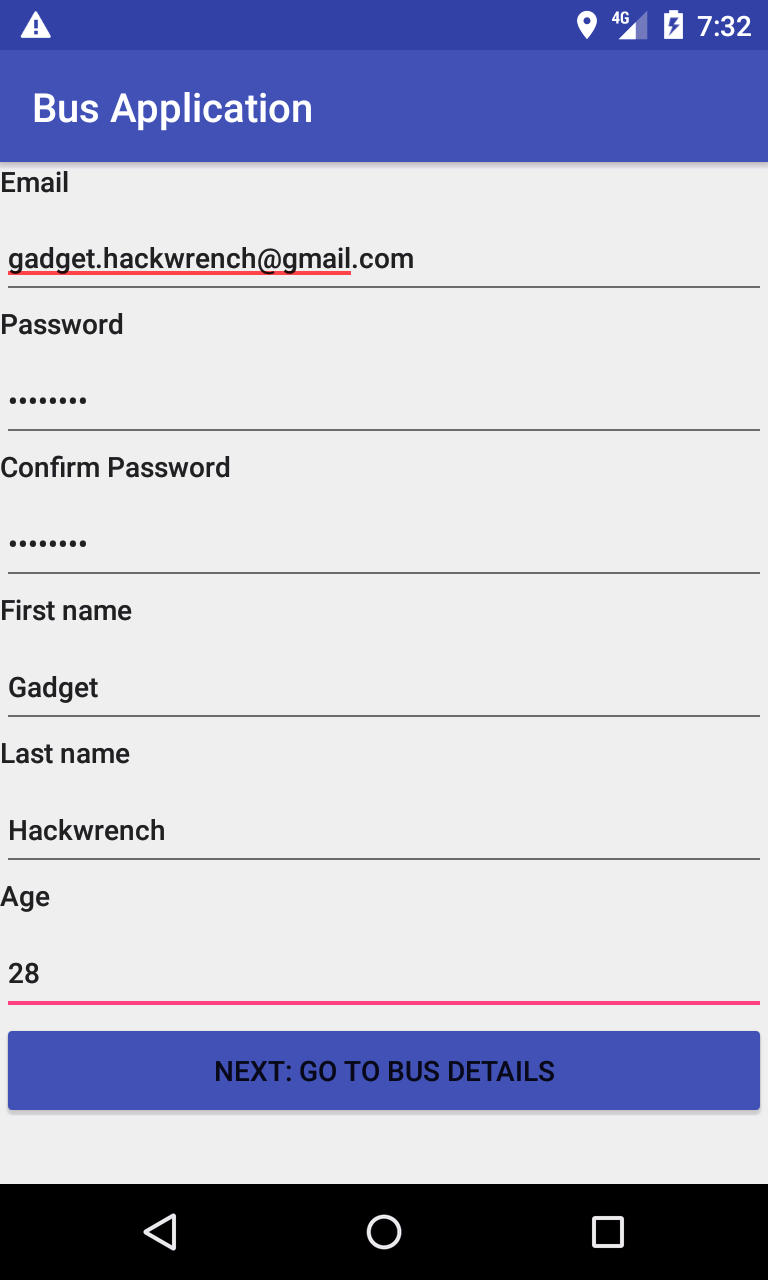


Rys. 1.2 Widok logowania aplikacji (*Przycisk, przekierowujący do rejestracji Kierowcy jest oznaczony czerwoną przestrzenią* ) [1]

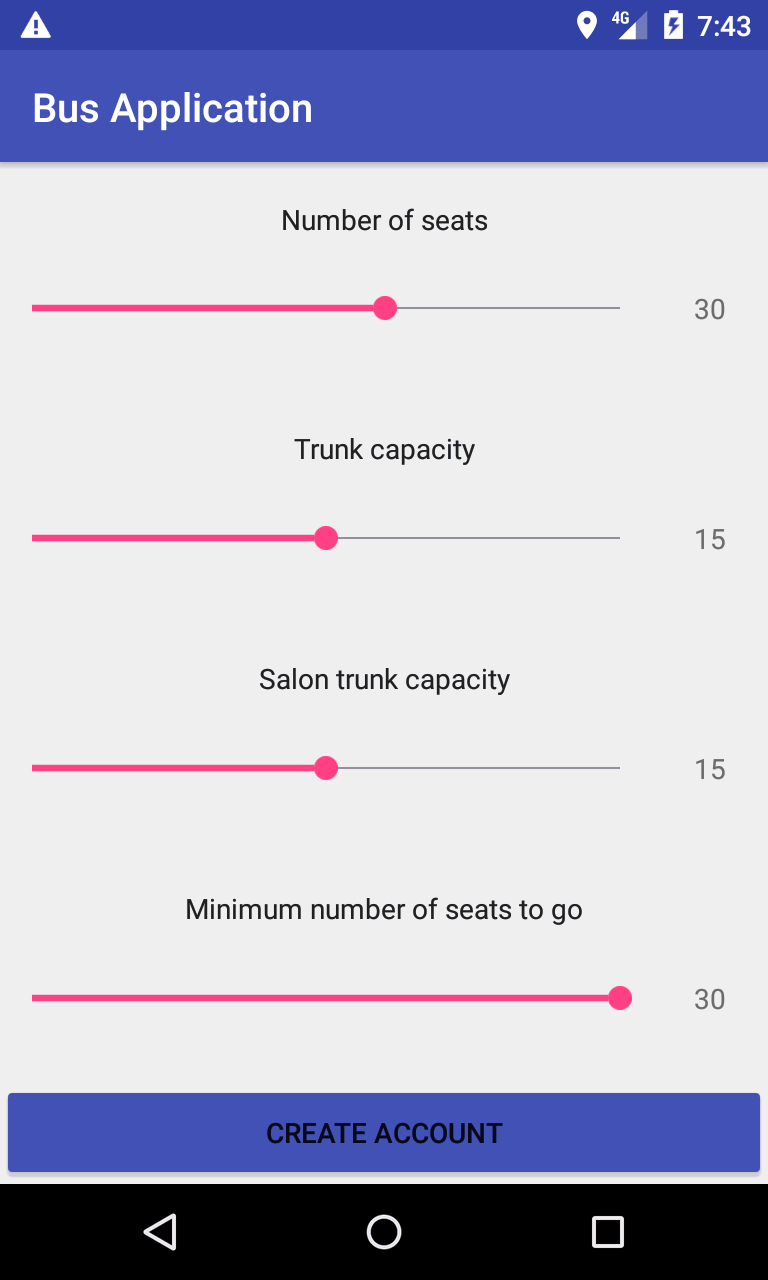
Dalej Kierowca uzupełnia email oraz hasło, powtórzenie hasła, dane osobowe – imię, nazwisko, wiek (rys. 1.3). Po kliknięciu „Go to Bus Details”, Kierowca przechodzi do widoku (rys. 1.4) gdzie on może wprowadzić szczegóły dotyczące autobusu – ilość miejsc w autobusie, pojemność bagażu, pojemność salonu na bagaż oraz minimalny próg zajętych siedzeń, niezbędnych dla rozpoczęcia jazdy. Dana informacja jest potrzebna dla możliwości Klientów mieć bieżące dane odnośnie bieżącego załadowania autobusu oraz żeby Kierowca mógł śledzić ilość zgłoszonych miejsc i bagażu w przyszłości.

Wszystkie szczegóły (oprócz emaila oraz hasła) mogą być zmienione na Widoku Szczegółów Kierowcy (rozdział 3.2.3 Szczegóły Kierowcy).

Po rejestracji Kierowca jest przekierowany do Głównego Widoku (rozdział 3.2.2)



Rys. 1.3 Widok Rejestracji Kierowcy, dane osobowe( Z uzupełnionymi przez Kierowcę danymi) [1]



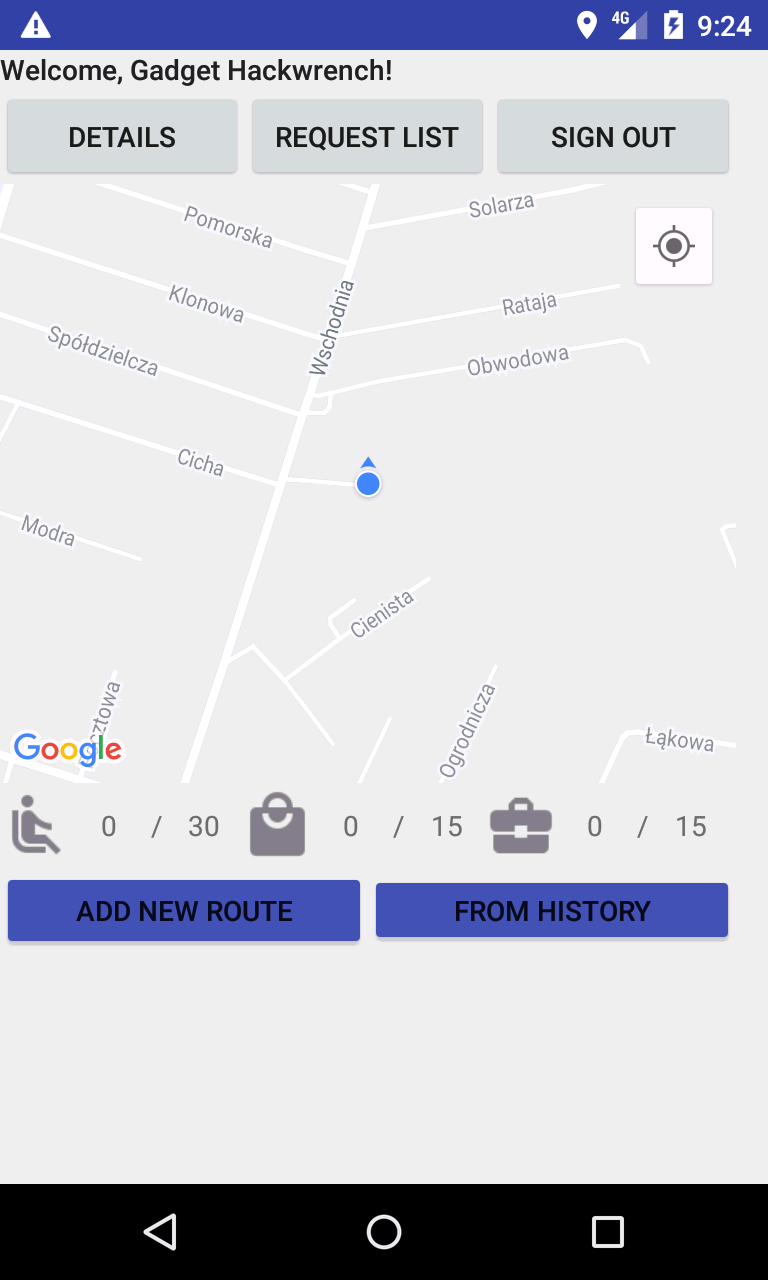
Rys. 1.4 Widok rejestracji Kierowcy, szczegóły autobusu [1]

Wychodząc ze specyfiki biznesu i pojemności przewozu, ilość możliwych miejsc w salonie jest ograniczona do 50, a pojemność bagażnika oraz bagażu w salonie – do 30. Także jest walidacja na polu „Minimalny próg miejsc dla przejazdu”, która pozwala na maksymalną wartość, która odpowiada ilości miejsc dla pasażerów w autobusie.

Po kliknięciu „Submit”, wszystkie detale Kierowcy są zapisywane w bazie Firebase Database pod wygenerowanym przez bazę kluczem w encjach „drivers”. Na późniejszych etapach szczegóły pobierane z tej samej encji.

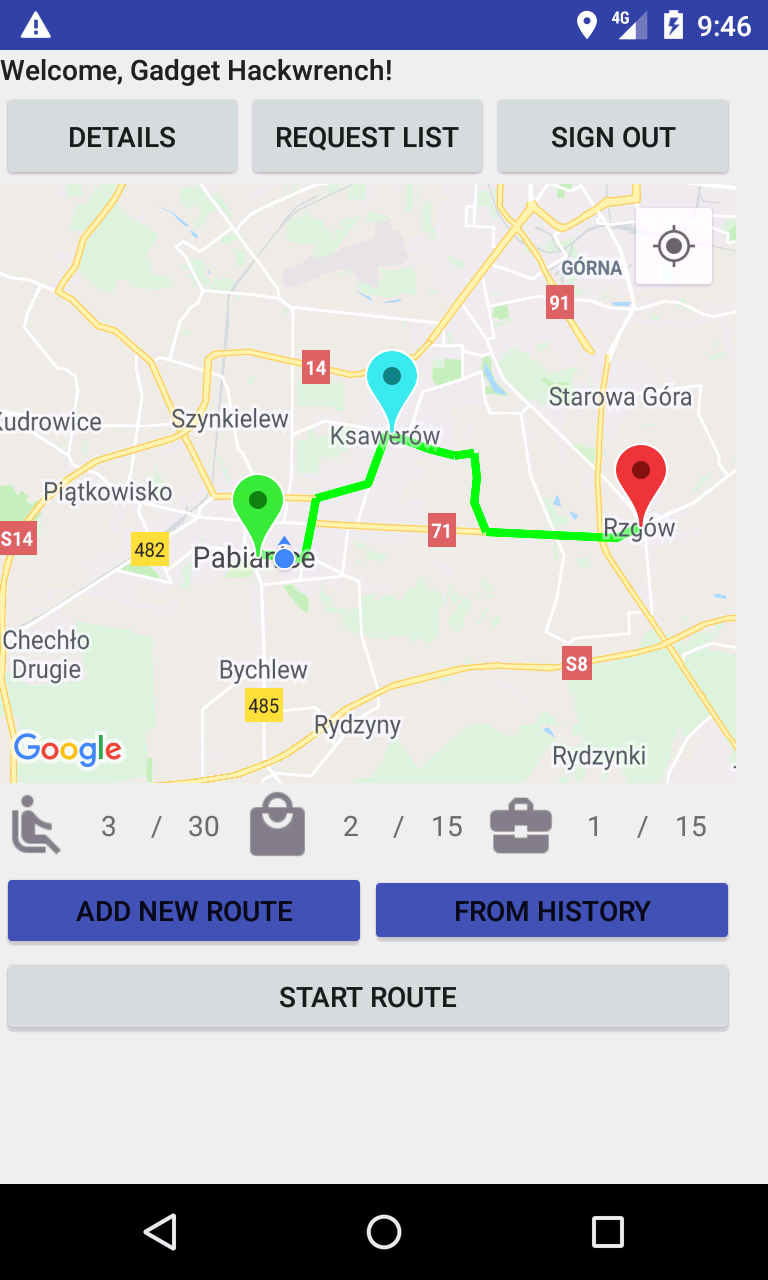
4.2.2 Główny widok

Po podaniu prawidłowych danych użytkownika, Kierowca jest przekierowany do Głównego Widoku (rys.1.5). Na nim przedstawiona jest mapa z bieżącą lokalizacją Kierowcy oraz przyciski „Dodaj nową ścieżkę” oraz „Wybierz ścieżkę z historii”. Dla potrzeb biznesowych mapa jest ograniczona do granic Polski. Aktualna trasa jest pobierana z bazy danych. Na widoku jest także widoczny status załadowania Autobusu (ilość zajętych miejsc, zgłoszona ilość bagażu oraz bagażu podręcznego). Na górze są widoczne przyciski Szczegóły, który przekierowuje do widoku Szczegółów Kierowcy (opisany w rozdziale 3.2.3 Szczegóły Kierowcy), listy przyjętych Żądań (rozdział 3.2.5) oraz Wyloguj, który wylogowuje Użytkownika z systemu Firebase Auth i przekierowuje do widoku Logowania.



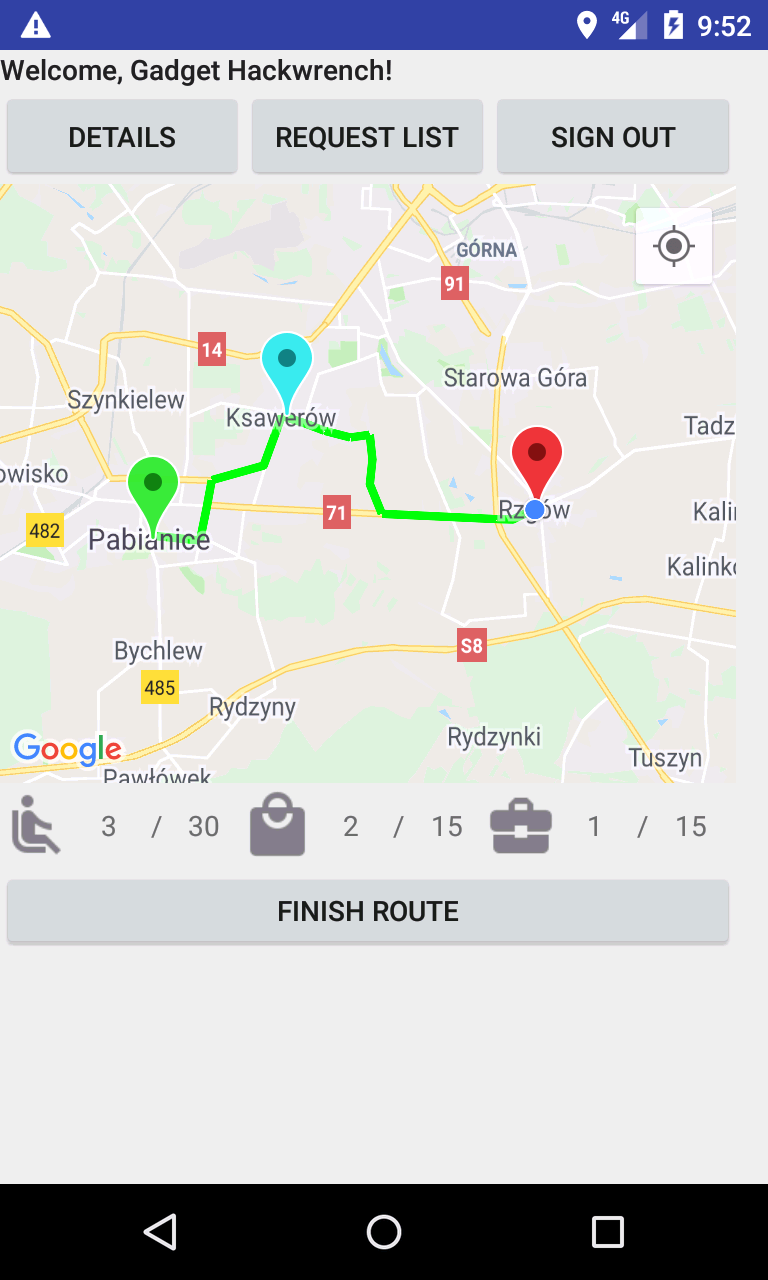
Rys. 1.5 Główny Widok Kierowcy, aktualna trasa nie jest zdefiniowana ( Z przyciskami Szczegóły, Lista Żądań oraz Wyloguj, widokiem mapy z bieżącą lokalizacją, statusem załadowania Autobusu oraz przyciskami dodania nowej trasy i pobrania Trasy z historii) [1]

W przypadku, gdy jest zdefiniowana Trasa dla przejazdu, ona jest pokazywana na mapie (rys.1.6). Pobierane z bazy są punkty przejazdu oraz na podstawie ich z API Google Directions rysowana jest Trasa, oraz listą Żądań od Klientów w statusie Zaakceptowano dla potrzeb definicji ilości osób, jednostek bagażu i bagażu podręcznego oraz punktów początkowych, skąd Kierowca podbiera każdego Klienta oraz punkty końcowe, gdzie Kierowca dowozi poszczególnego Klienta.



Rys. 1.6 Główny Widok Kierowcy, aktualna trasa jest zdefiniowana ( *Z zaktualizowanymi szczegółami Autobusu na podstawie Żądań, widoczny jest przycisk „Rozpocznij Jazdę”*) [1]

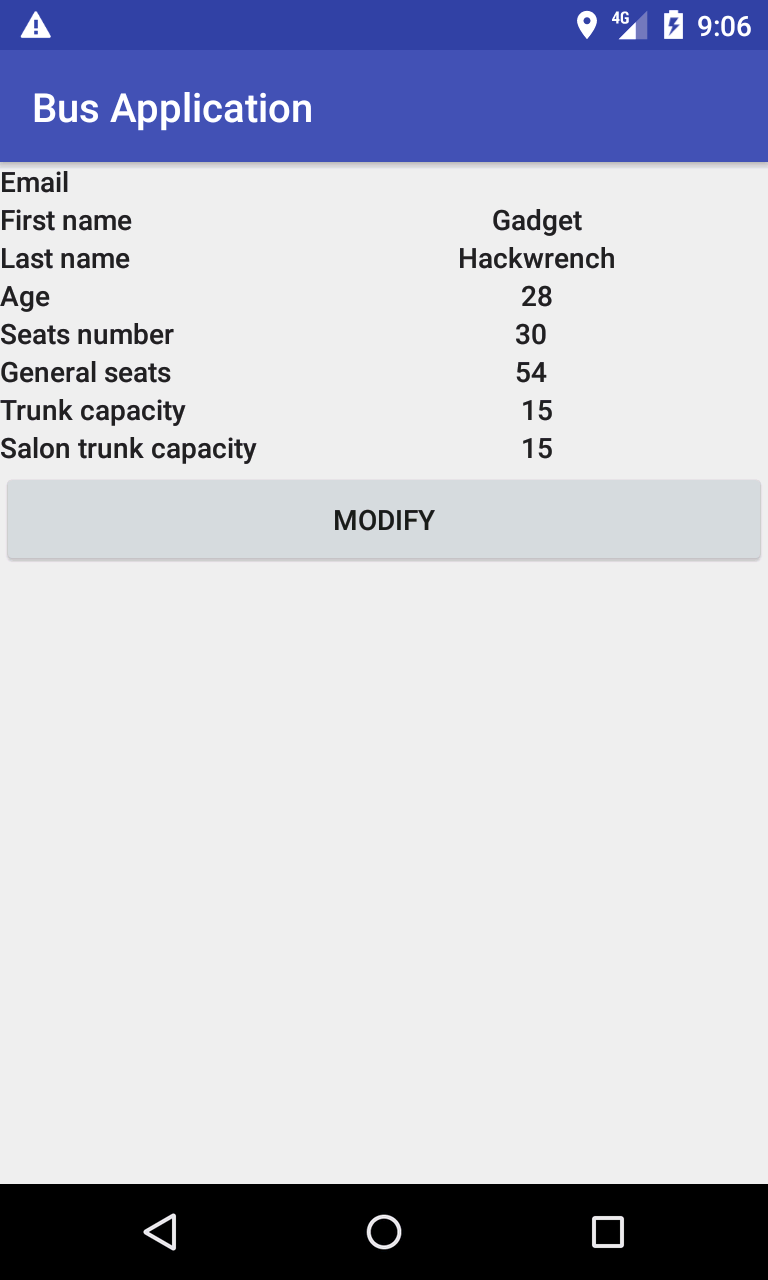
W tym momencie widoczny jest panel sterowania Trasą. Zależnie od statusu Trasy – czy jest ona otwarta prze Kierowcę, lub jest w trakcie realizacji, to jest możliwość rozpoczęcia Trasy w pierwszym przypadku (rys.1.6), a w drugim – jej zakończenie (rys. 1.7). Proces realizacji Trasy jest opisany w rozdziale 3.2.6.



Rys. 1.7 Główny Widok Kierowcy, aktualna trasa jest w statusie W trakcie ( *widoczny jest przycisk „Zakończ Jazdę”*) [1]

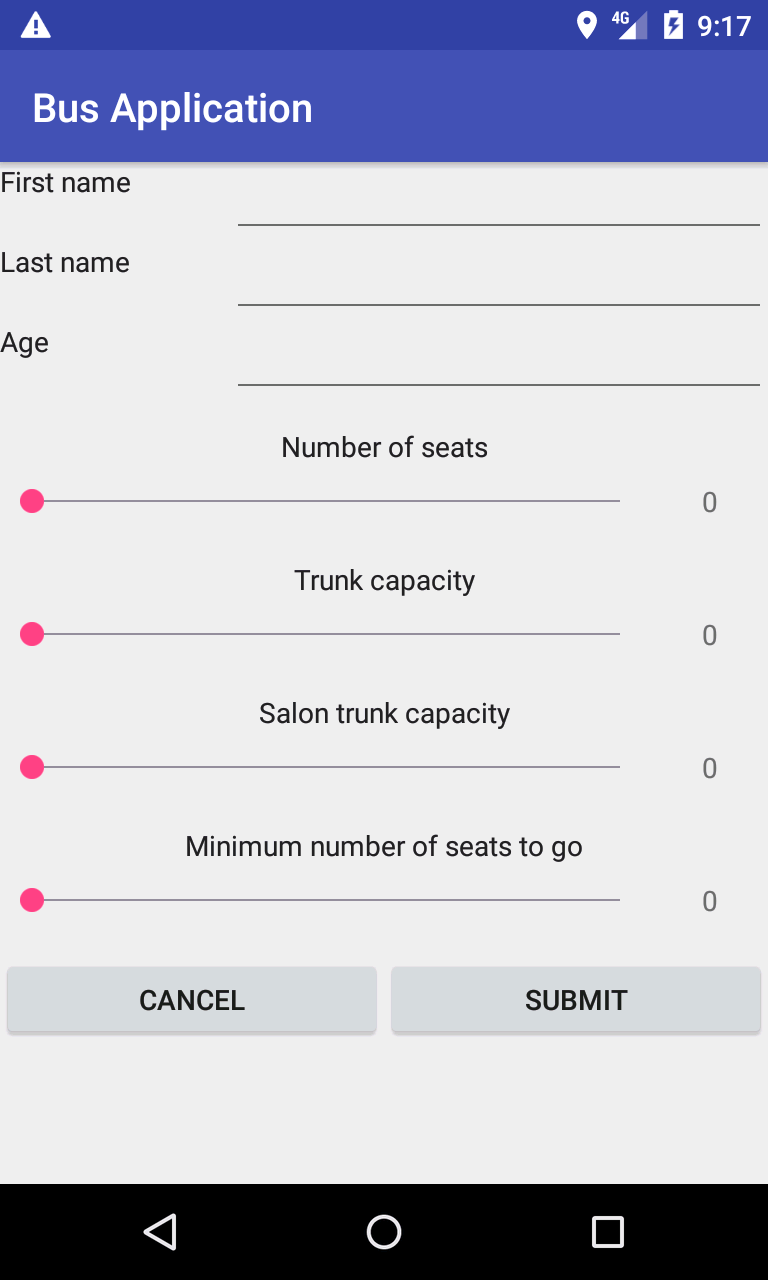
4.2.3 Szczegóły Kierowcy

Po wybraniu punktu „Szczegóły” na Głównym Widoku zostaje na stronie (rys. 1.8).



Rys. 1.8 Szczegóły Kierowcy [1]

Na widoku Kierowca widzi swoje dane osobowe (imię, nazwisko, wiek) oraz zdefiniowane wcześniej szczegóły Autobusu. Po kliknięciu przycisku „Modyfikuj”, Kierowca ma możliwość zmienić swoje dane (rys. 1.9). Proces odbywa się podobnie, jak opisane w rozdziale 3.2.1.



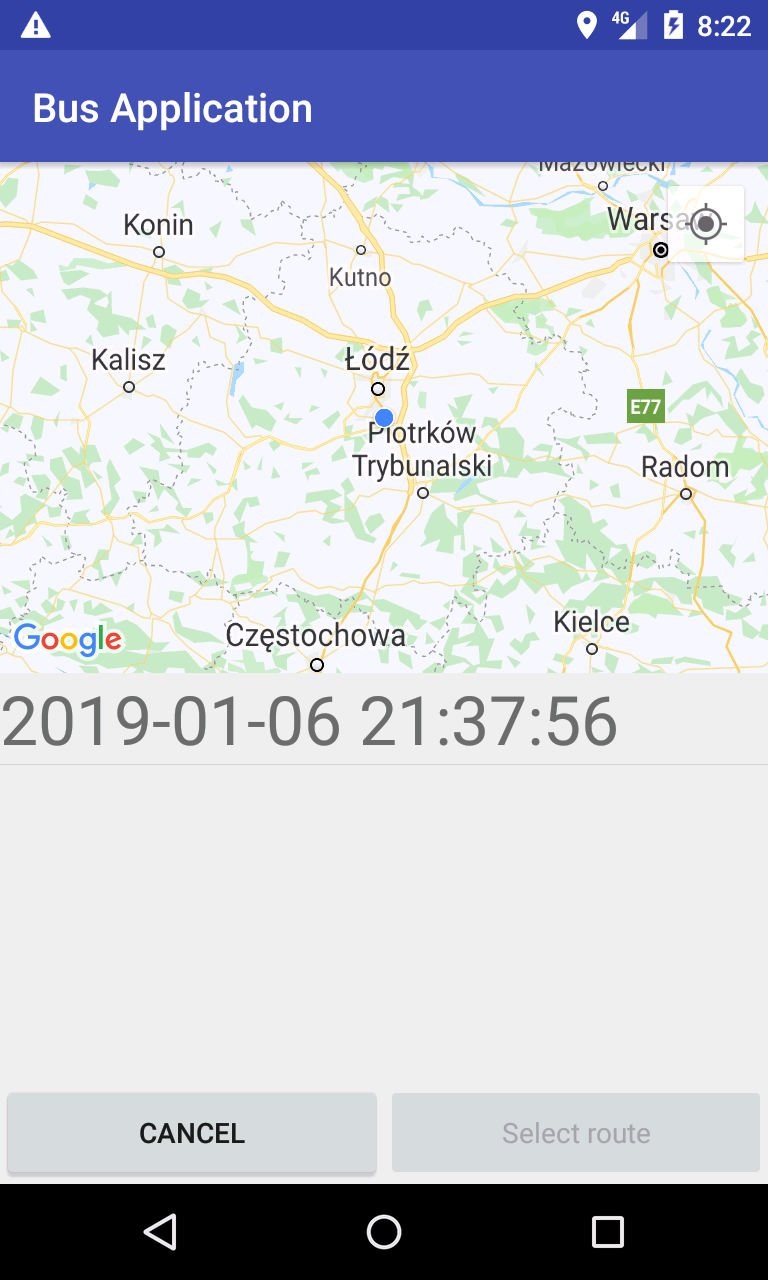
Rys. 1.9 Modyfikacja szczegółów Kierowcy [1]

Naciśnięcie przycisków „Wycofaj” oraz „Zapisz” powodują anulowanie modyfikacji szczegółów w pierwszym przypadku, a w drugim wypadku zapisanie zmienionych szczegółów w bazie danych.

4.2.4 Dodanie nowej Trasy

W aplikacji ma możliwość definicji aktualnej Trasy. Z Głównego Widoku on wybiera Trasę, która była zdefiniowana w przeszłości, gdy taka jest (przycisk „Dodaj Trasę z historii”) lub dodać nową (przycisk o odpowiedniej nazwie).

W pierwszym przypadku Kierowca jest przekierowany do widoku listy Tras z historii przejazdów (rys. 1.10).

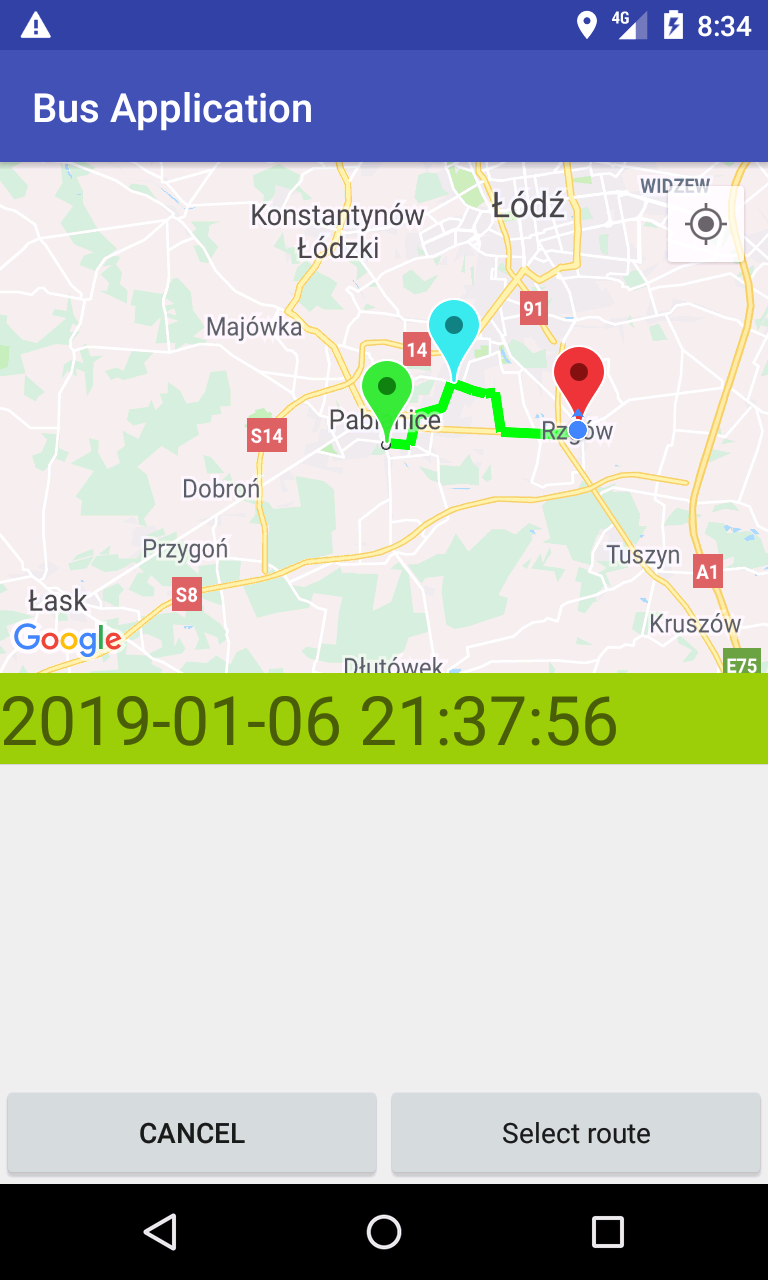


Rys. 1.10 Wybranie Trasy z historii ( *Żadna trasa nie jest wybrana*) [1]

Na Widoku prezentują się widok Google Maps oraz lista Tras, które były zdefiniowane wcześniej przez danego Kierowcę, z datami definicji poszczególnych Tras. Po wybraniu pozycji z listy, element zostaje zaznaczony kolorem, szczegóły Trasy są pokazywane na mapie, a dokładniej punkty przejazdu, a także za pomocą Google Directions API rysowana ścieżka Kierowcy. Także w ten moment aktywuje się przycisk „Wybierz Trasę” (rys. 1.11).

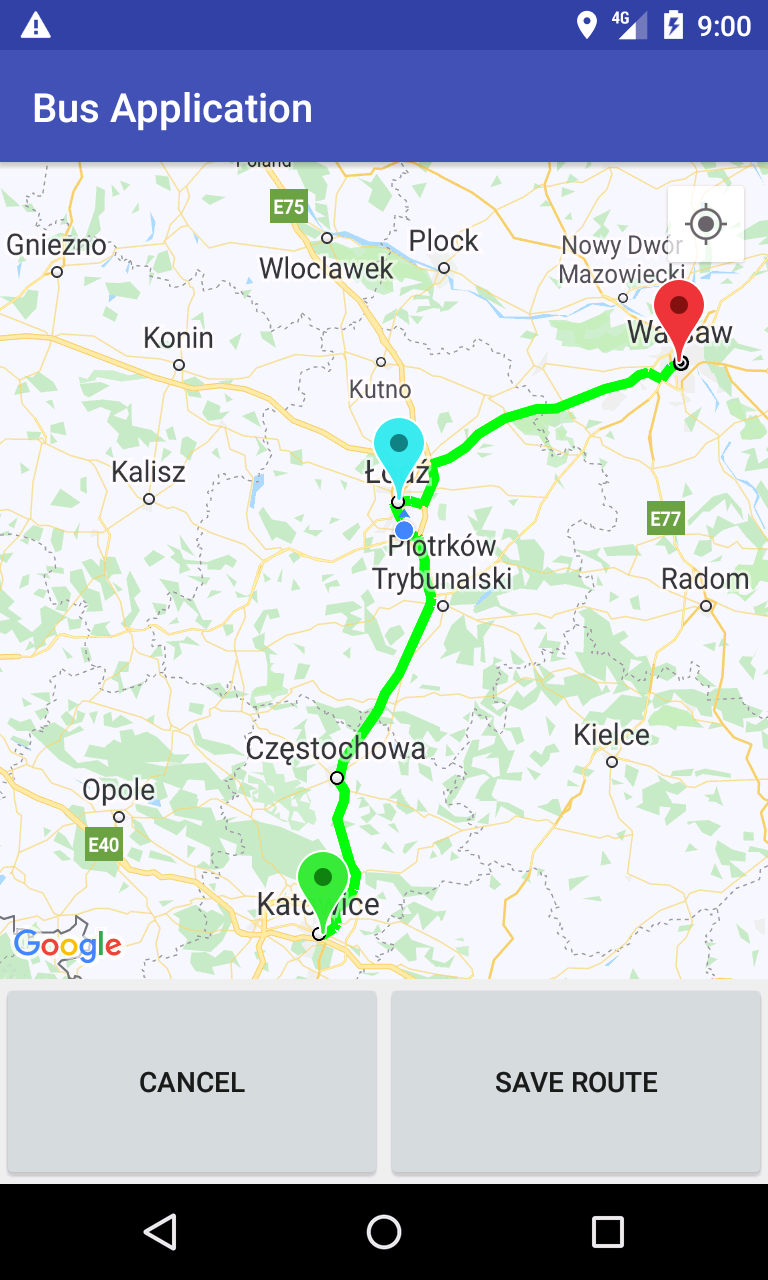
Za pomocą przycisku „Anuluj” dodanie jest wycofane, a przy naciśnięciu „Wybierz Trasę”, wybrana Trasa jest dodana do aktualnej ścieżki Kierowcy, co jest pokazane po automatycznym przejściu do Głównego Widoku przy zapisaniu na mapie (rys. 1.6).

Także po naciśnięciu na Głównym Widoku przycisku „Dodaj nową Trasę” wyświetlany zostaje się inny widok (rys.1.12) na którym Kierowca na mapie określa punkty przejazdu wybierając z mapy kliknięciem w odpowiedniej kolejności Markerów, z których rysowana jest Trasa w sposób, opisany dla Dodania Trasy z historii.



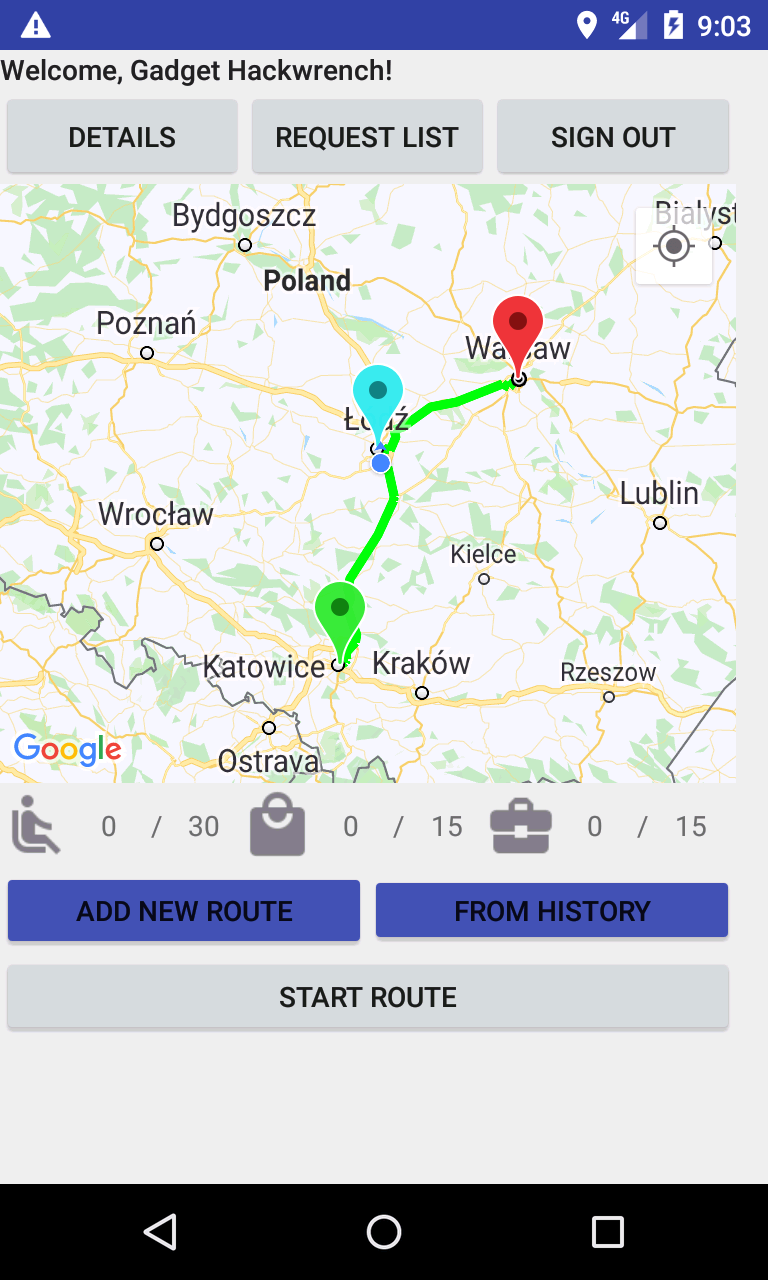
Rys. 1.11 Wybranie Trasy z historii ( Wybrana jest jedna z Tras*, jest podświetlona na rysunku*) [1]

Punkty wydzielone odpowiednim kolorem w zależności od pozycji na trasie: zielonym – punkt początkowy, niebieskim – punkt(y) pośredniczące oraz czerwonym – punkt docelowy. Po naciśnięciu przycisków „Anuluj” oraz „Dodaj Trasę” powoduje wycofanie dodawania aktualnej trasy oraz zapisywania danej Trasy w aktualnej ścieżce Kierowcy w bazie danych i wyświetlaniu jej na mapie Głównego widoku odpowiednio.



Rys. 1.12 Definicja nowej Trasy ( Dodanie Trasy do aktualnej ścieżki Kierowcy za pomocą przycisku jest możliwe gdy liczba punktów jest >=2) [1]

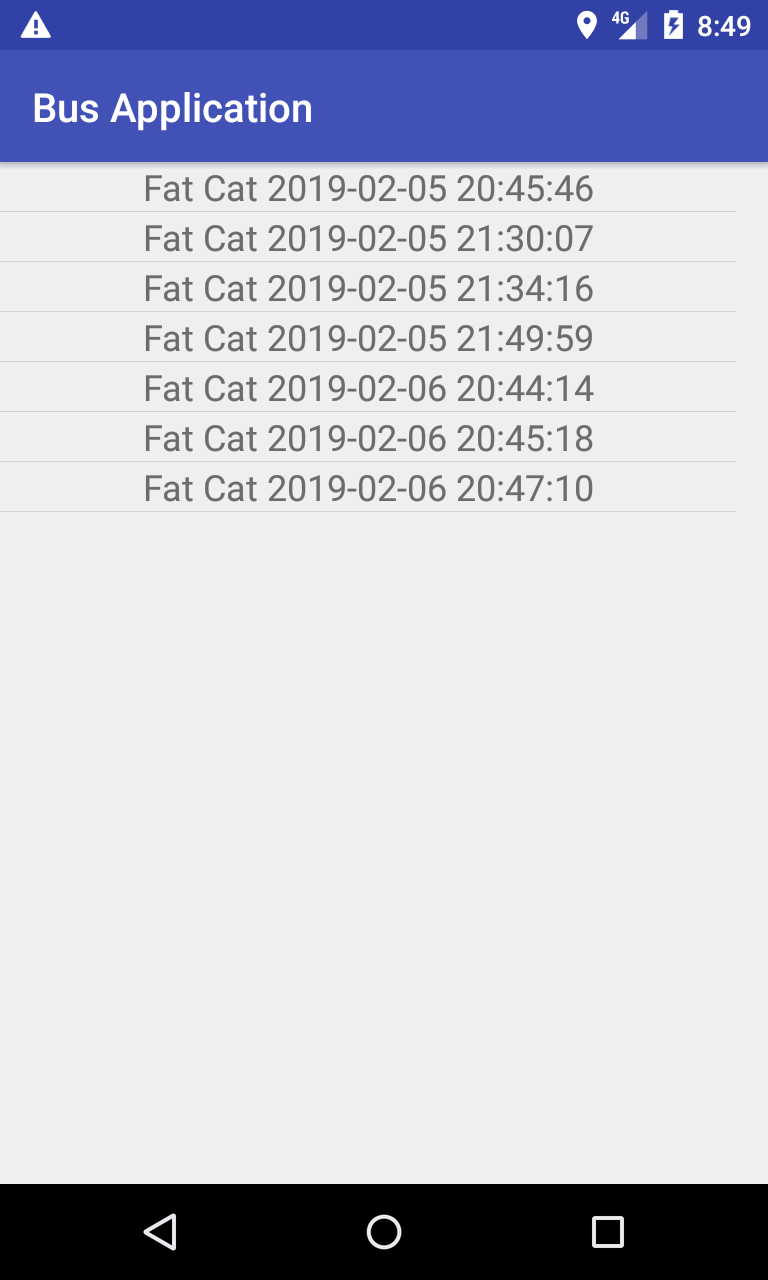
Jak można zauważyć na Głównym Widoku mapa jest aktualizowana, pokazując nowo dodaną Trasę (rys.1.13).



Rys. 1.13 Główny Widok Kierowcy po definicji nowej Trasy [1]

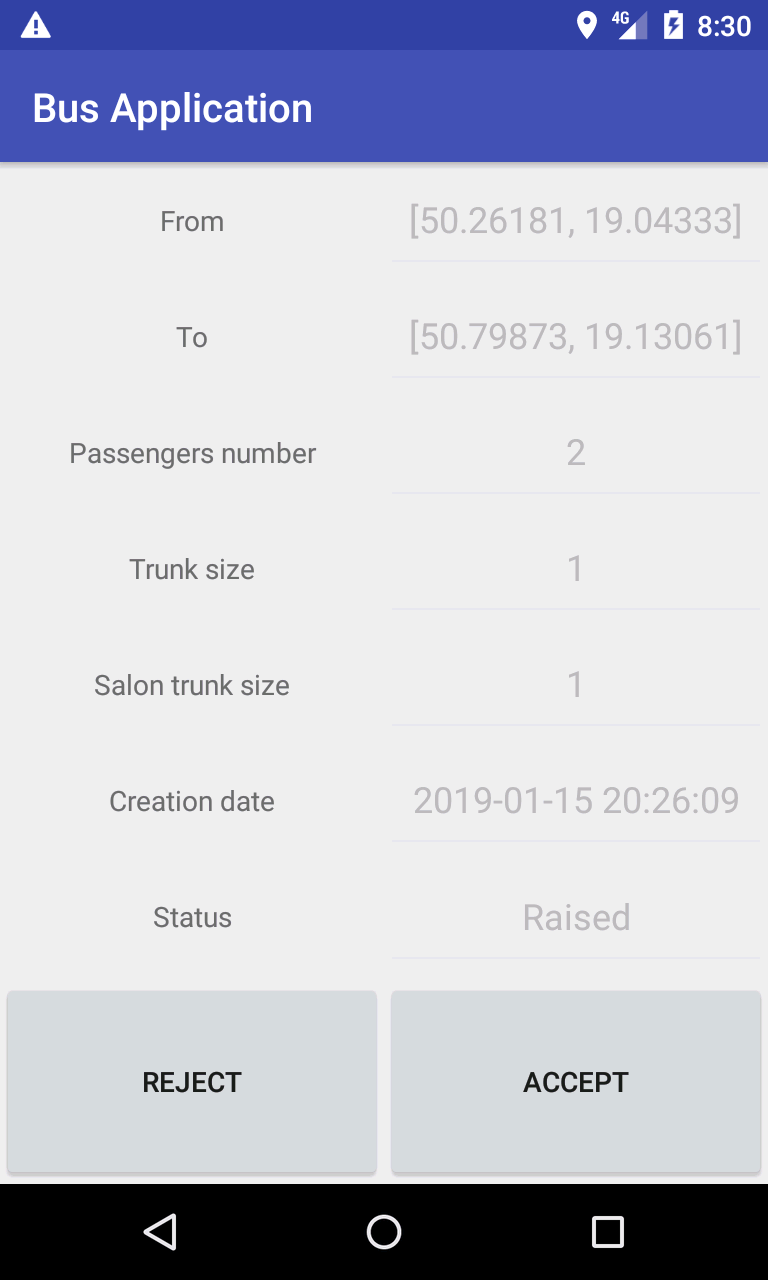
4.2.5 Sterowanie Żądaniami od Klienta

Po otrzymaniu żądania od Klienta, którego zgłoszenie jest opisane w rozdziale 3.3.4, Kierowca może zobaczyć dodane Żądanie z listy Żądań (rys. 1.14).



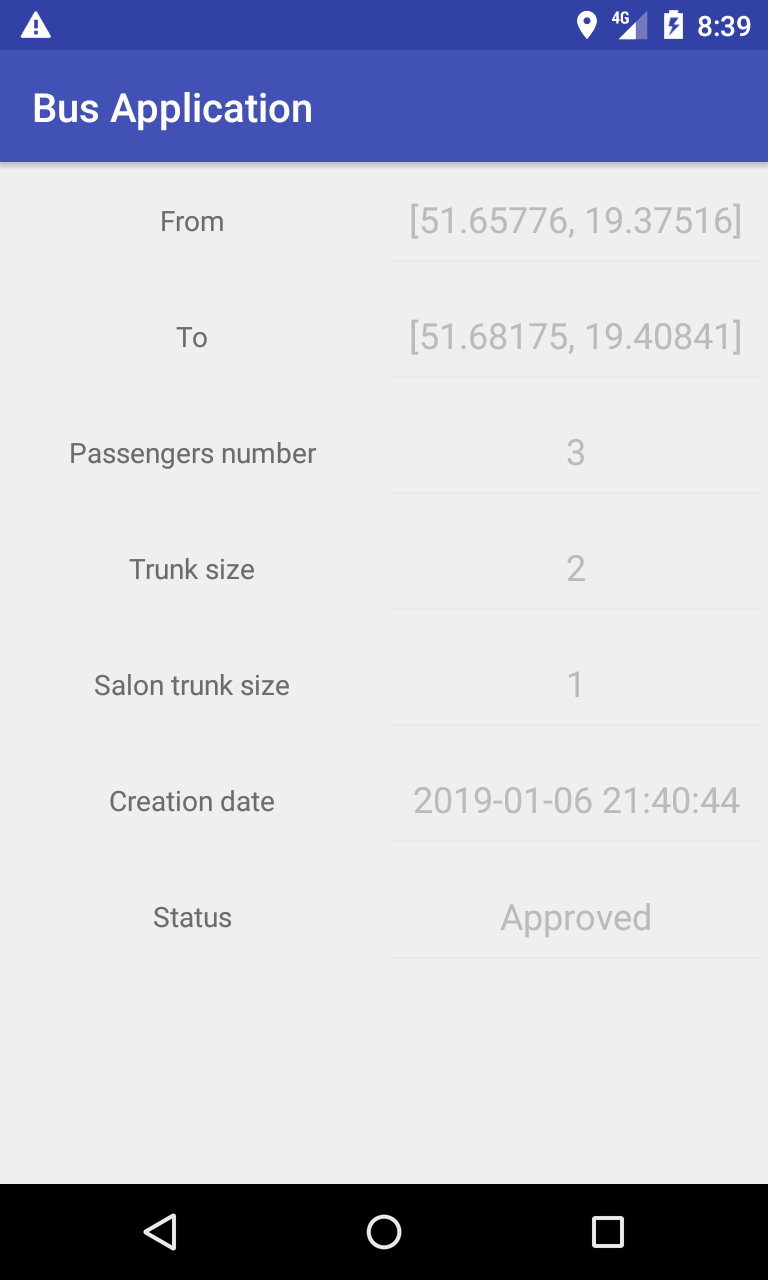
Rys. 1.14 Widok listy Żądań skierowanych do Kierowcy [1]

Na liście widoczne poszczególne Żądania z imionami Klientów, którzy zgłosili Żądanie wraz z datami ich utworzenia. Po wybraniu Żądania z listy Kierowca jest przekierowany do detali danego Żądania (rys. 1.15). W szczegółach Żądania Kierowca widzi Punkt, z którego Klient chcę rozpocząć jazdę, Punkt Docelowy, ilość pasażerów, ilość bagażu oraz bagażu w salonie, datę utworzenia żądania oraz jego status. Także w przypadku, gdy Żądanie jest w statusie „Otworzone”, Kierowca może jego cofnąć lub zatwierdzić za pomocą odpowiednich przycisków.



Rys. 1.15 Widok Szczegółów Żądania [1]

W przypadku, gdy Żądanie jest w innym statusie, to Kierowca nie ma możliwości sterowania im (rys.1.16).



Rys. 1.16 Widok Szczegółów Żądania, Żądanie w statusie Zaakceptowano [1]

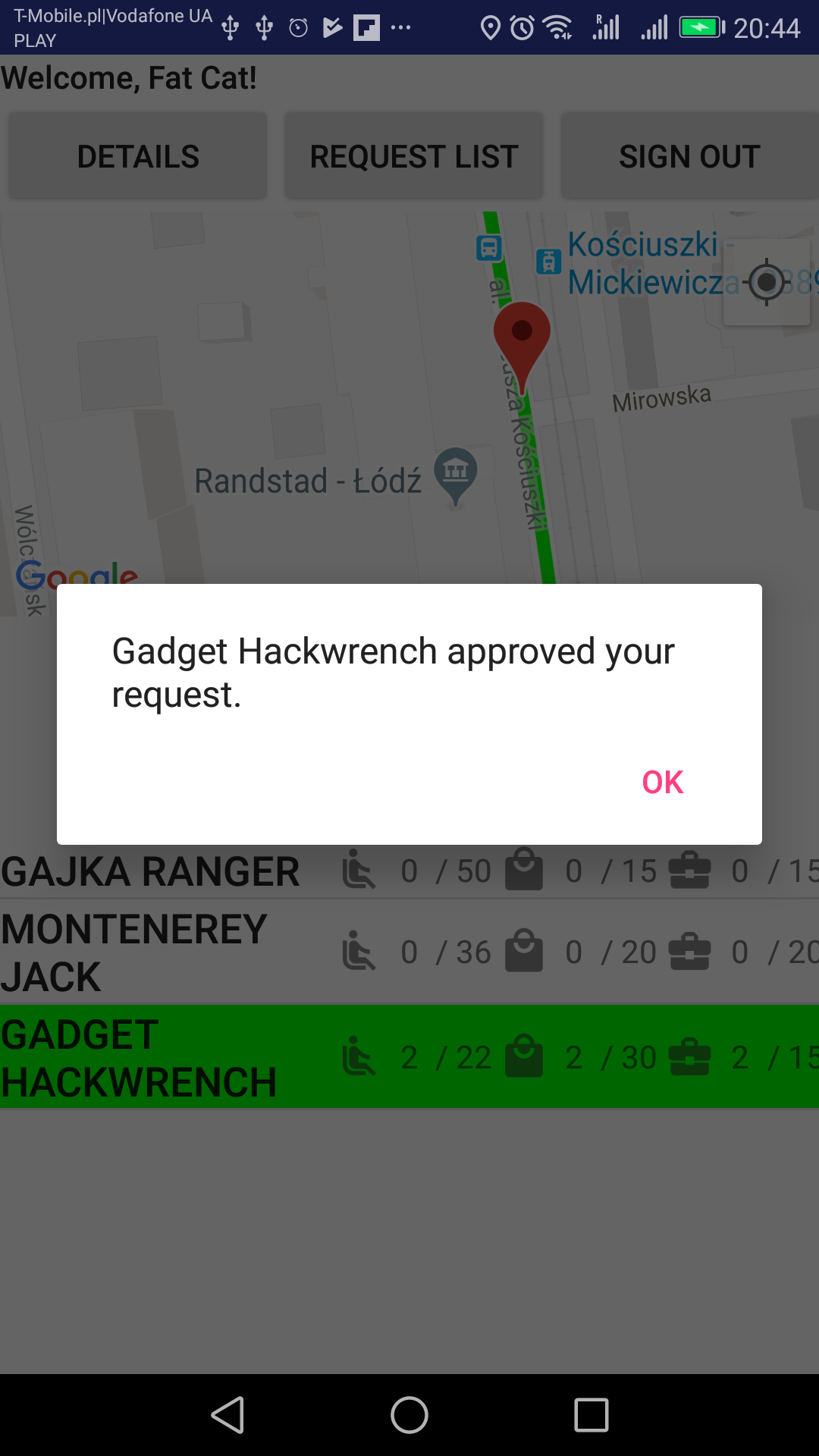
Po akceptacji Żądania, ono jest dodane do aktualnej Trasy u Kierowcy w bazie danych, a na Głównym widoku, dane autobusu są uzupełnione o informacje określone w Żądaniu (rys. 1.17).

Po akceptacji Żądania, jego status jest zmieniany na Zaakceptowano, co można zobaczyć, w jego szczegółach, jak na przykładzie (rys.1.16).



Rys. 1.17 Główny Widok Żądania, dane autobusu są aktualizowane na podstawie zaakceptowanego Żądania[1]

Także po akceptacji, Klient otrzymuje notyfikacje o akceptacji Żądanie prze kierowcę, co widać na rys. 1.18.



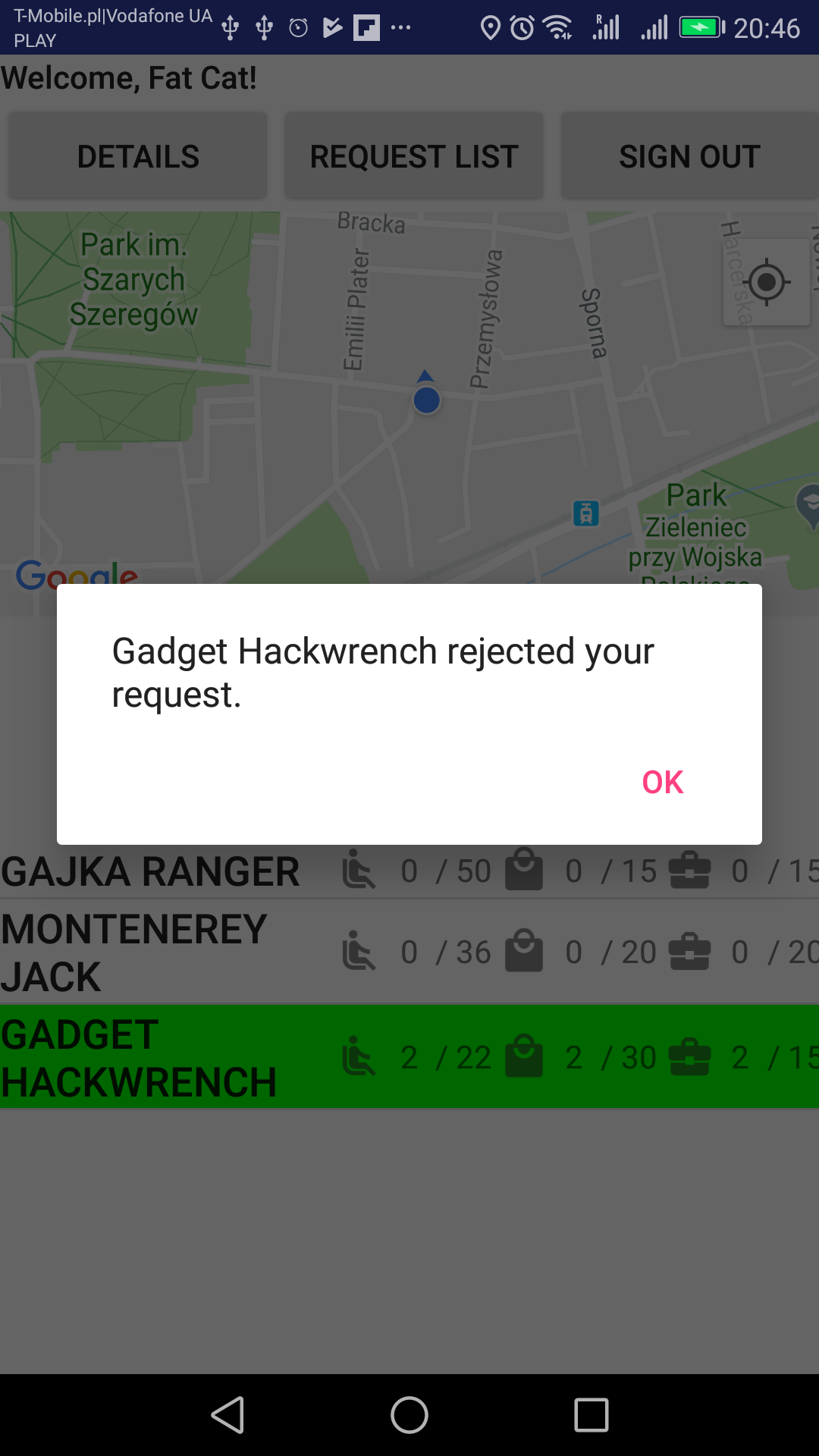
Rys. 1.18 Notyfikacja o akceptacji Żądania przez Kierowcę na widoku Klienta [1]

Trasa na mapie Kierowcę jest uzupełniana o punkty zgłoszenia (rys.1.19).Na danym widoku Punkt ciemno niebieski – Punkt początkowy Żądania, różowy – punkt Docelowy.

W przypadku, gdy Kierowca decyduje, że Żądanie jemu nie odpowiada, to ma możliwość jego anulowania. Wtedy Klient dostaje odpowiednią notyfikację (rys.1.20).



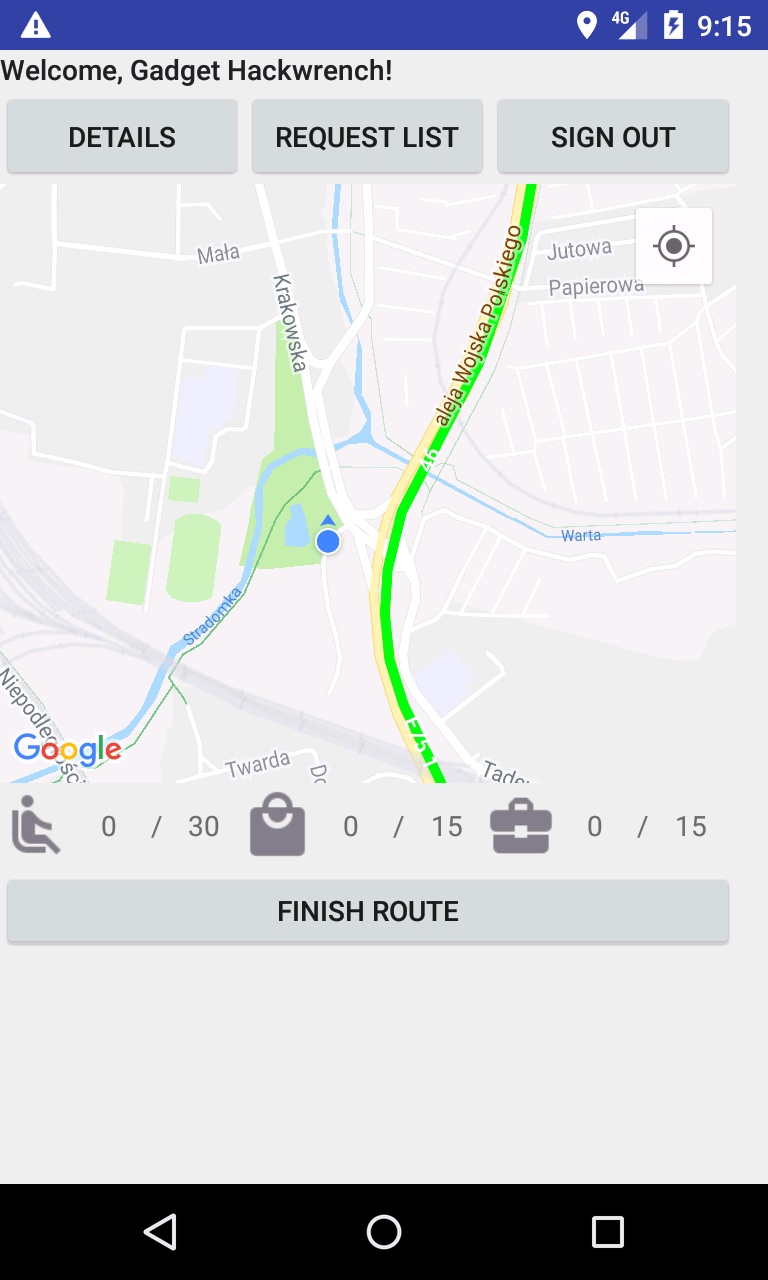
Rys. 1.19 Aktualizowana Trasa na widoku Kierowcy po akceptacji Żądania [1]



Rys. 1.20 Notyfikacja na widoku Klienta po wycofaniu Żądania przez Kierowcę [1]

4.2.6 Przepływ przejazdu Trasy przez Kierowcę oraz proces dowozu Klientów do Punktów Docelowych

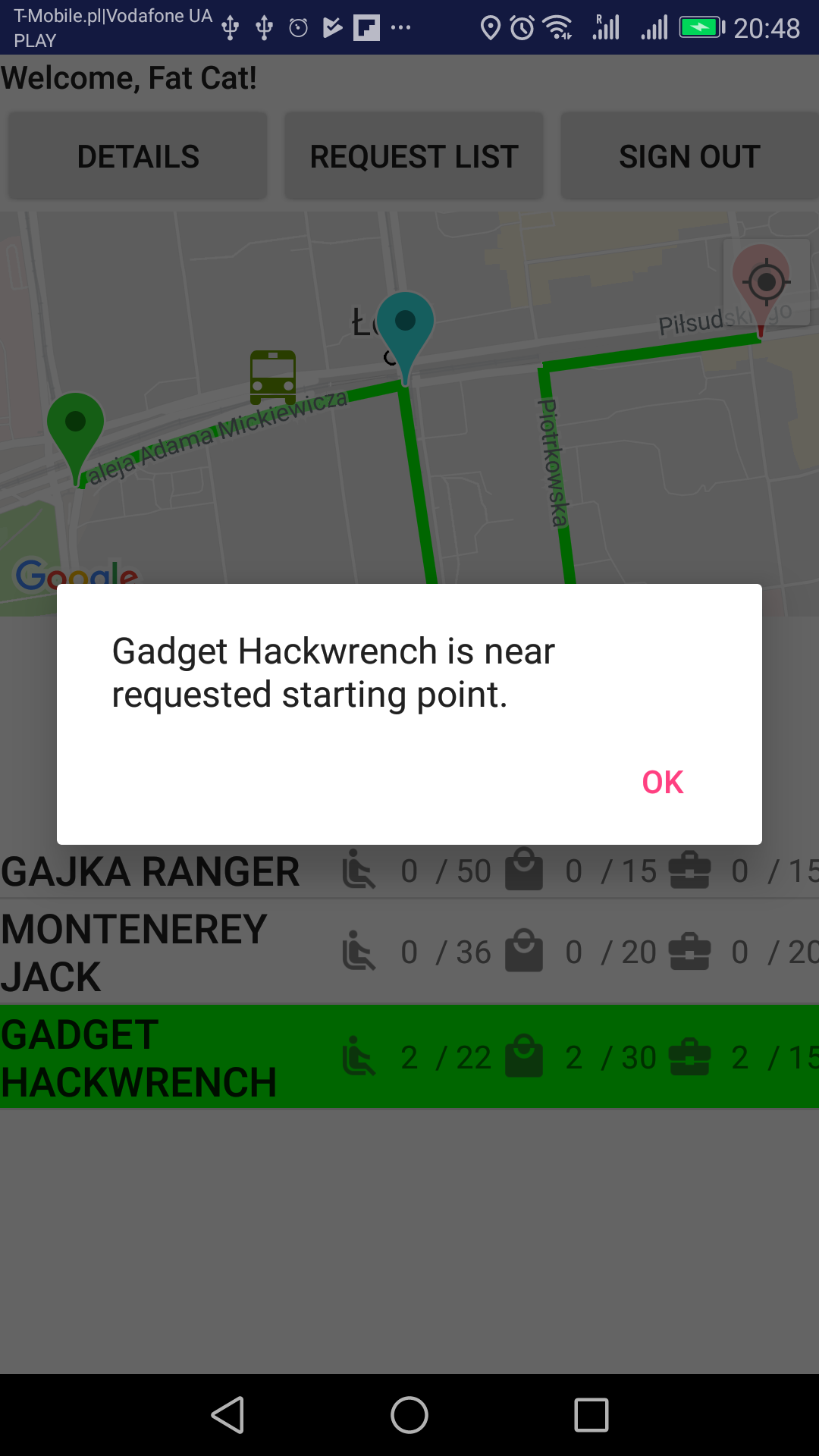
Kiedy kierowca dla aktualnej Trasy rozpoczyna proces realizacji jazdy (przez kliknięcie „Rozpocznij Jazdę”), aplikacja sprawdza bieżącą lokalizację Kierowcy i na podstawie zaakceptowanych żądań na Trasę, zwalniana jest zgłoszoną ilość miejsc oraz jednostek bagażu i bagażu podręcznego ze szczegółów Autobusu, gdy osiągnięty jest zdefiniowany punkt końcowy Żądania Klienta, co widoczne jest na rys.1.21.



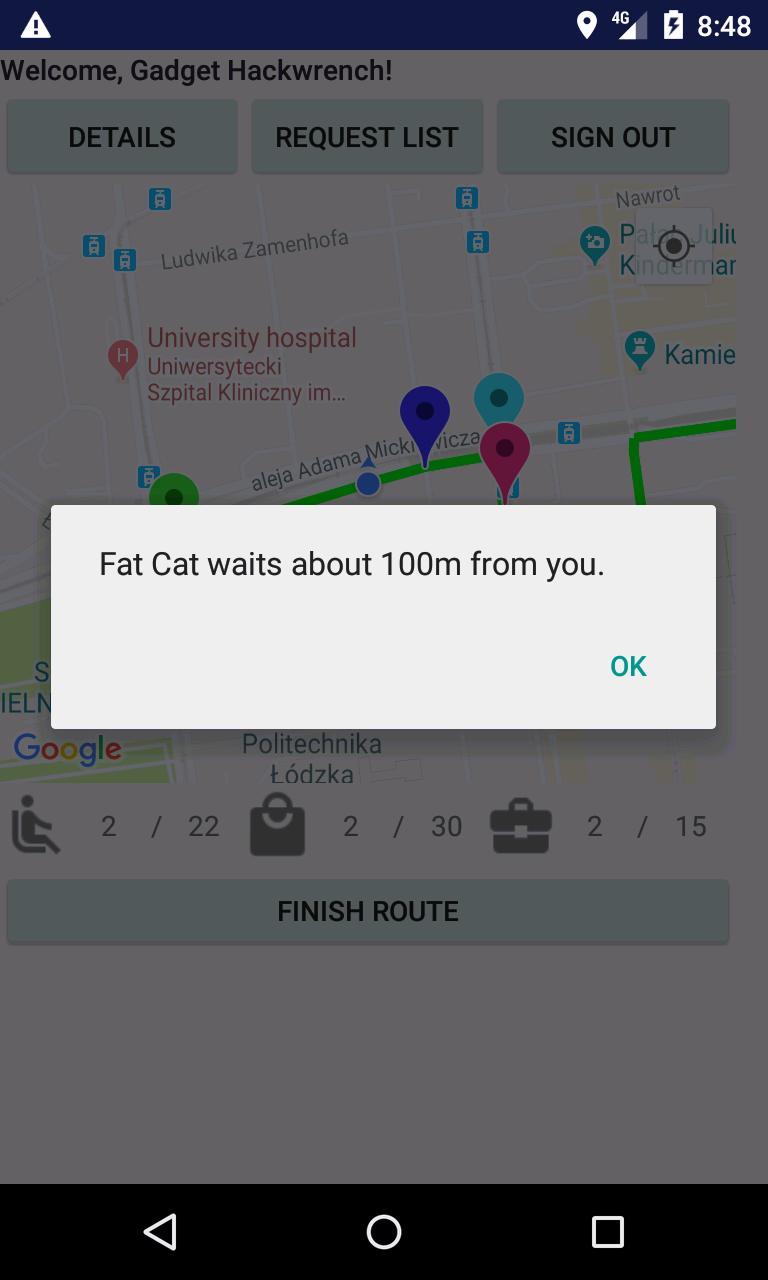
Rys. 1.21 Aktualizacja danych Autobusu na Głównym Widoku Kierowcy, po dojazdu Kierowcy do punktu końcowego Żądania[1]

W momencie, gdy Kierowca dojeżdża do Punktu Końcowego Trasy i deklaruje jej zakończenie (po przez przycisk „Zakończ Jazdę”) bieżąca Trasa jest archiwizowana w bazie, a mapa jest oczyszczana, widok jest przywrócony do stanu, widocznego na rys. 1.5.

Także warto dodać, że przy zbliżeniu się przez Kierowcę do Punktu początkowego Żądania (blisko 100 metrów), jak Klient, tak i Kierowca dostają komunikat (rys.1.22,1.23).



Rys. 1.22 Notyfikacja do Klienta o zbliżeniu się Kierowcy do Punktu początkowego Żądania [1]



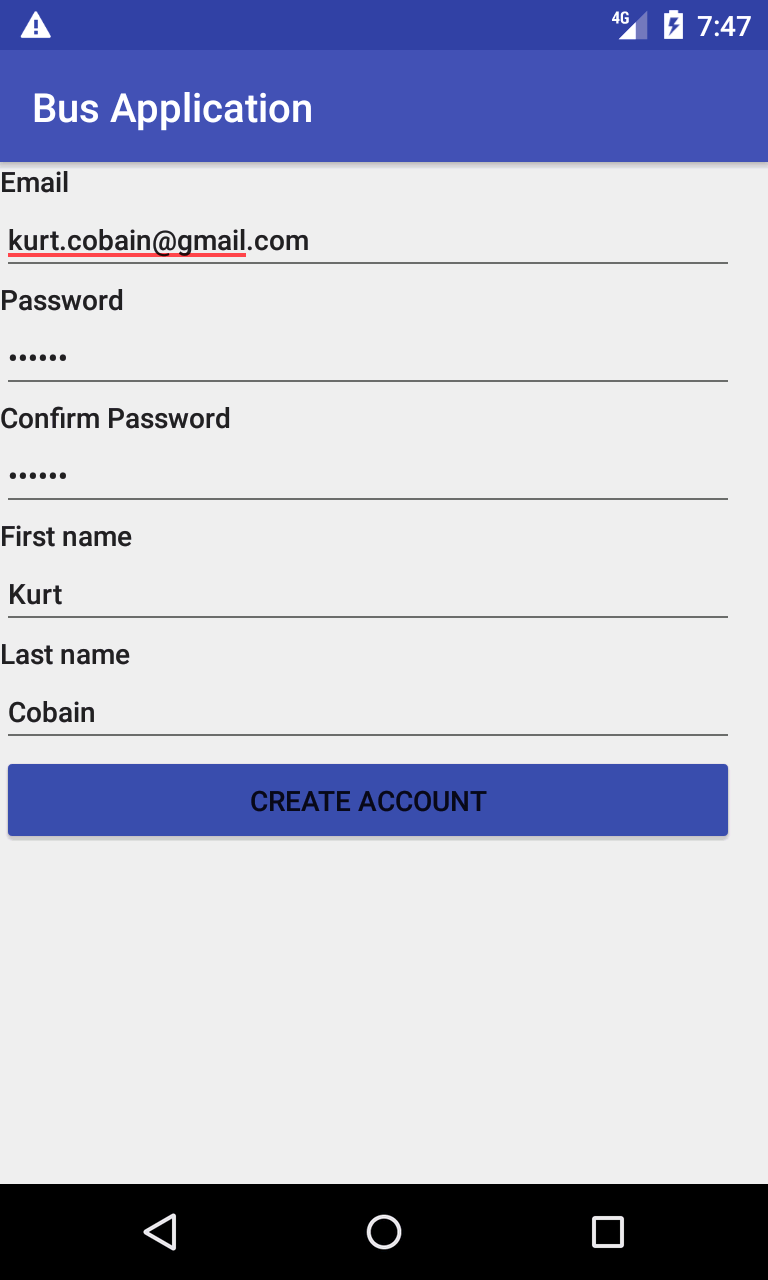
Rys. 1.23 Notyfikacja do Kierowcy o bliskości Klienta do bieżącej lokalizacji Kierowcy pod czas jazdy[1]

4.3 Moduł Klienta

4.3.1 Rejestracja nowego Klienta

Po kliknięciu przycisku „Zarejestruj się jako Klient” na widoku Logowania (rys. 1.2). Użytkownik jest przekierowany na widok Rejestracji Klienta (rys. 1.24)

Na danym widoku Klient uzupełnia dane personalne: email, hasło oraz imię i nazwisko. Po naciśnięciu „Utworz Konto”, dane zapisywane w węzłu „Klienci” w bazie danych Firebase, a nowo utworzony Klient, jest przekierowany do Głównego Widoku (rys. 1.25).



Rys. 1.24 Rejestracja Klienta[1]

4.3.2 Główny widok dostępnych Tras

Po logowaniu jako Klient lub rejestracji użytkownika spotyka następujący widok (rys. 1.25)

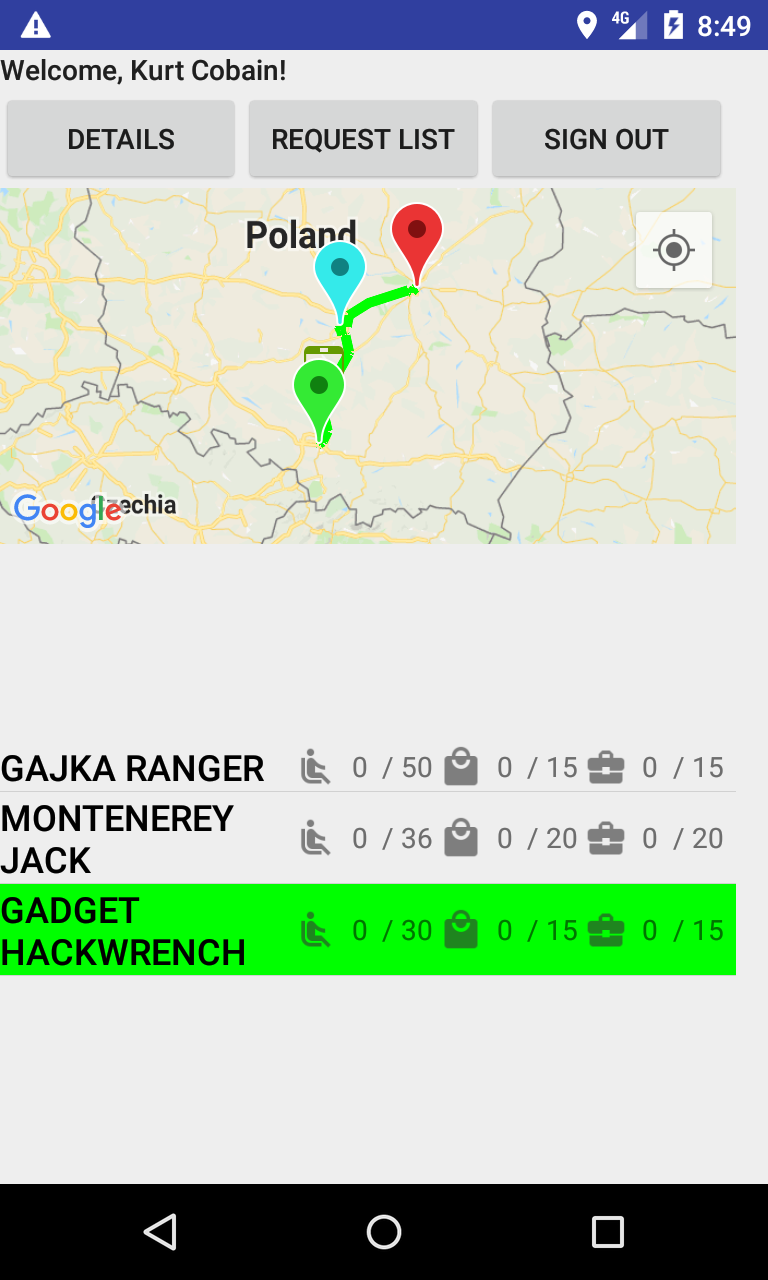


Rys. 1.25 Główny Widok Klienta[1]

Jak w przypadku z Głównym Widokiem Kierowcy (rozdział 3.2.2) na górze widoczne są przyciski: Szczegóły (prowadzi do widoku Szczegółów Klienta, rozdział 3.3.3), Lista Żądań oraz Wyloguj, działanie jakiego jest takie same jak w przypadku Głównego Widoku Kierowcy.

Reszta widoku zajęta jest przez Mapę oraz Listę Kierowców z dołu, na której Klient może zobaczyć aktualny stan autobusu poszczególnego Kierowcy – ilość zajętych miejsc w salonie pasażerów, zajęta objętość bagażnika oraz liczba bagażu w salonie. Jak i w przypadkach Kierowcy, opisanych w rozdziałach 3.2.2, 3.2.5 oraz 3.2.6, ta informacja jest aktualizowana na bieżąco, w zależności od przyjęcia przez Kierowcę żądań od Klientów oraz realizacji Przejazdów na Trasie.

Po wybraniu Kierowcy z listy na mapie zostaje wyświetlona aktualna trasa (rys. 1.26).



Rys. 1.26 Wybranie trasy z listy Kierowcy na Głównym Widoku Klienta[1]

Zielonym markerem jest zaznaczony Punkt Początkowy trasy, niebieskim – Punkt pośredni, a czerwonym – Punkt Końcowy.

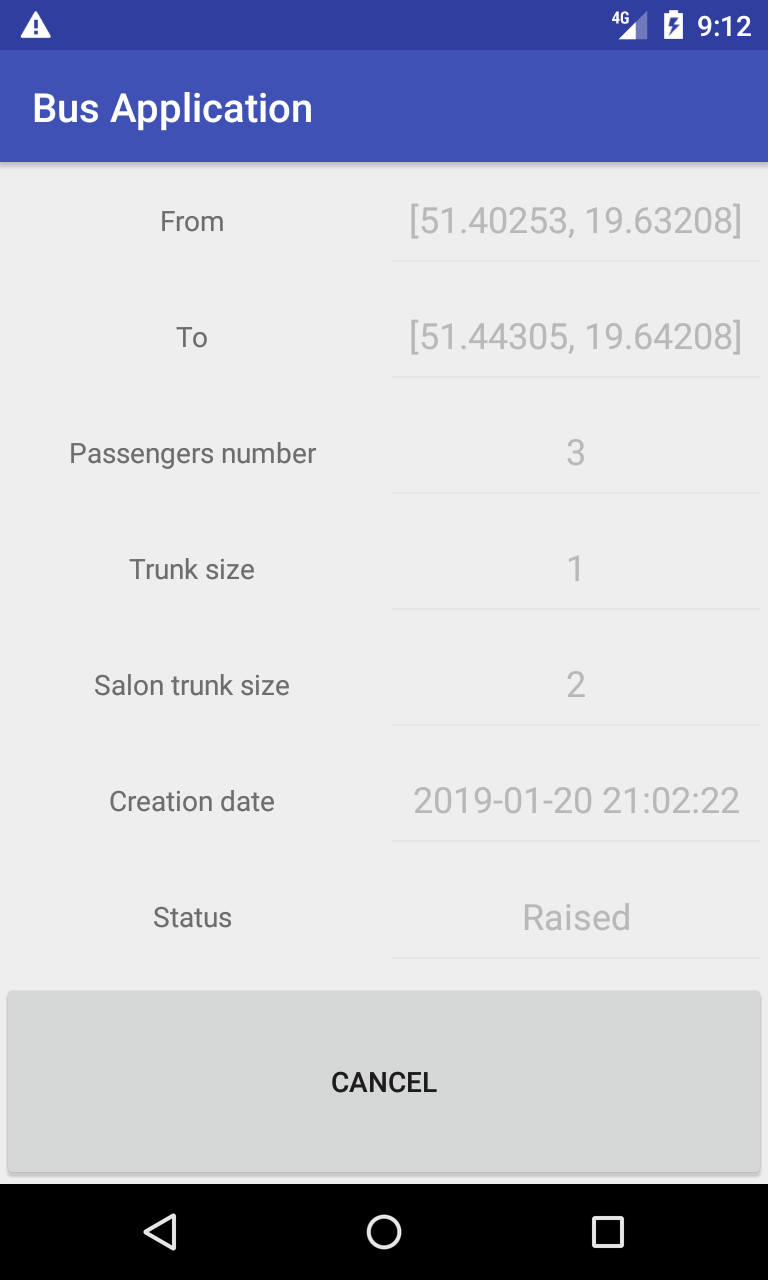
Po naciśnięciu przycisku „Lista Żądań” Klient widzi dodane przez niego żądania,, zidentyfikowane po imieniu Kierowcy oraz dacie dodania (rys. 1.27).



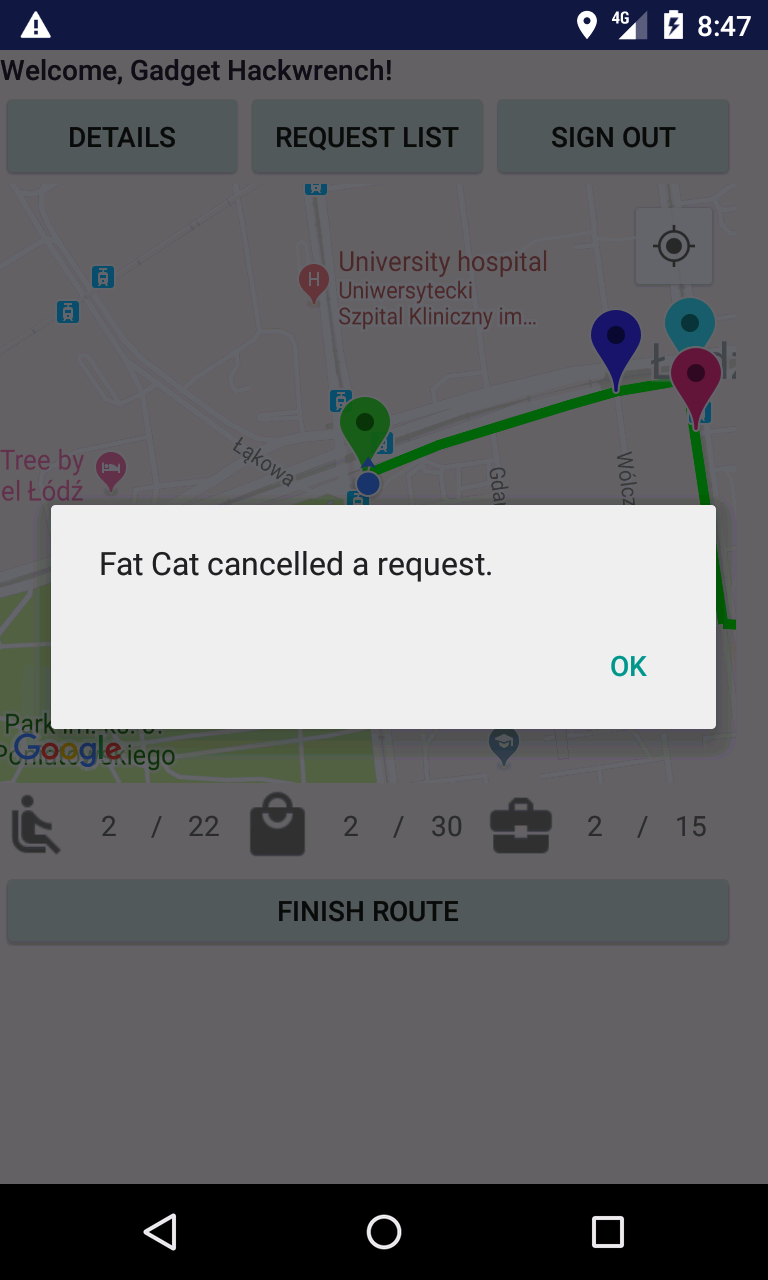
Rys. 1.27 Widok Listy Żądań od Klienta[1]

Proces dodania Żądania jest opisany w rozdziale 3.3.4. Po wybraniu Żądania Klient może zobaczyć jego szczegóły (rys. 1.28).

Na nim zaprezentowane są Punkty Początkowy i Końcowy Żądania, ilość miejsc, bagażu oraz bagażu podręcznego w zgłoszeniu, czas utworzenia Żądania oraz aktualny status. W przypadku, gdy Żądanie ma status „Otworzono”, to Klient ma możliwość anulowania Żądania (szczególniej w rozdziale 3.3.4). Po anulowaniu Żądania, Kierowcą dostaje wiadomość (rys.1.29).



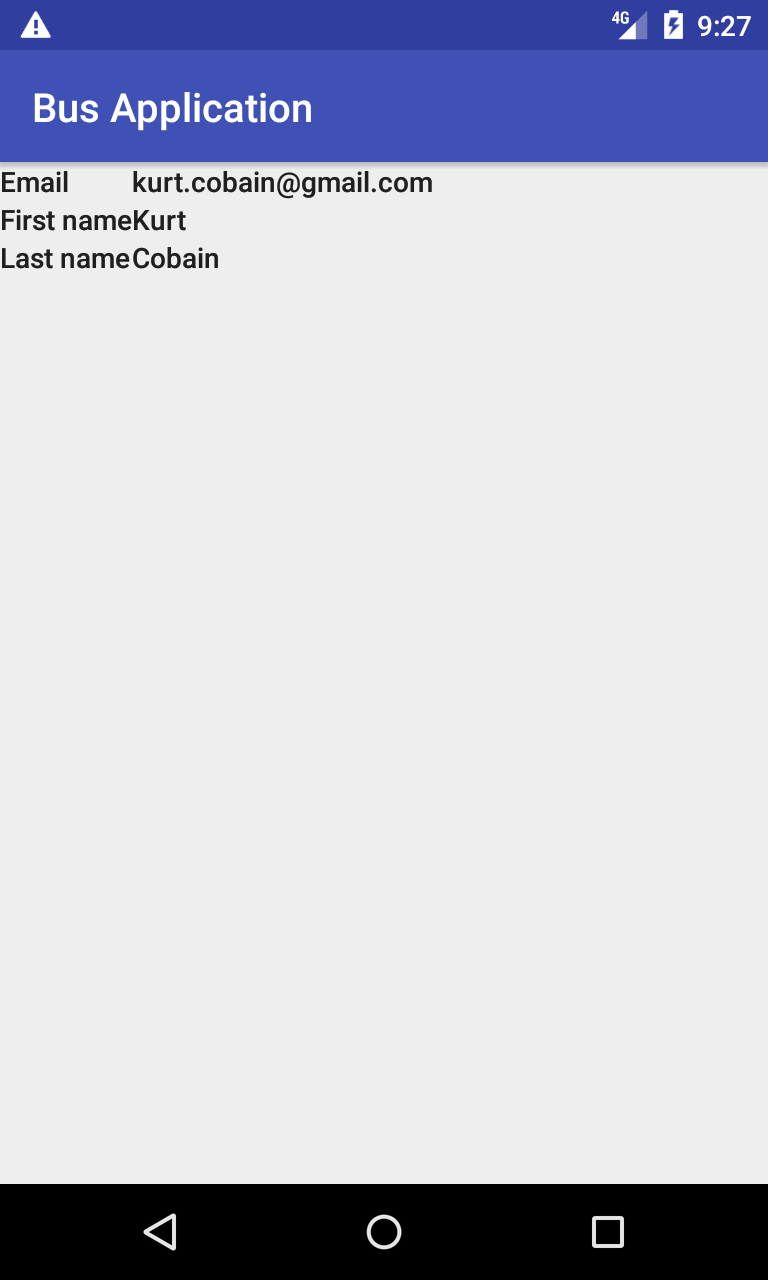
Rys. 1.28 Szczegóły żądania, złożonego przez Klienta [1]



Rys. 1.29 Komunikat o anulowaniu Żądania przez Klienta na widoku Kierowcy [1]

4.3.3 Szczegóły Klienta

Po przejściu z Głównego Widoku Klienta na Szczegóły, użytkownika spotyka widok (rys. 1.30).

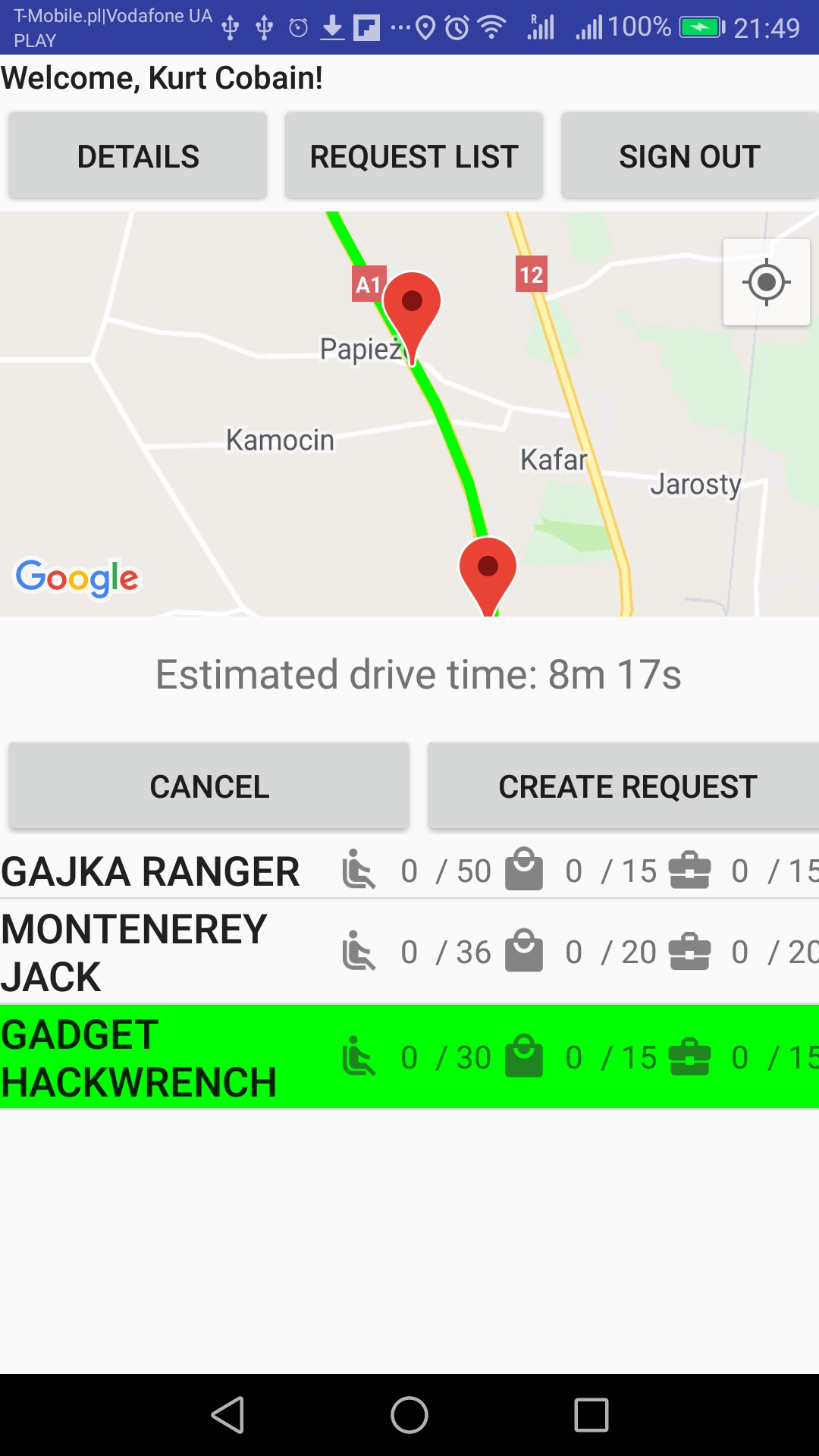


Rys. 1.30 Szczegóły Klienta [1]

Na tym widoku Klient może zobaczyć email, oraz dane osobowe, podane przy Rejestracji (rozdział 3.3.1)

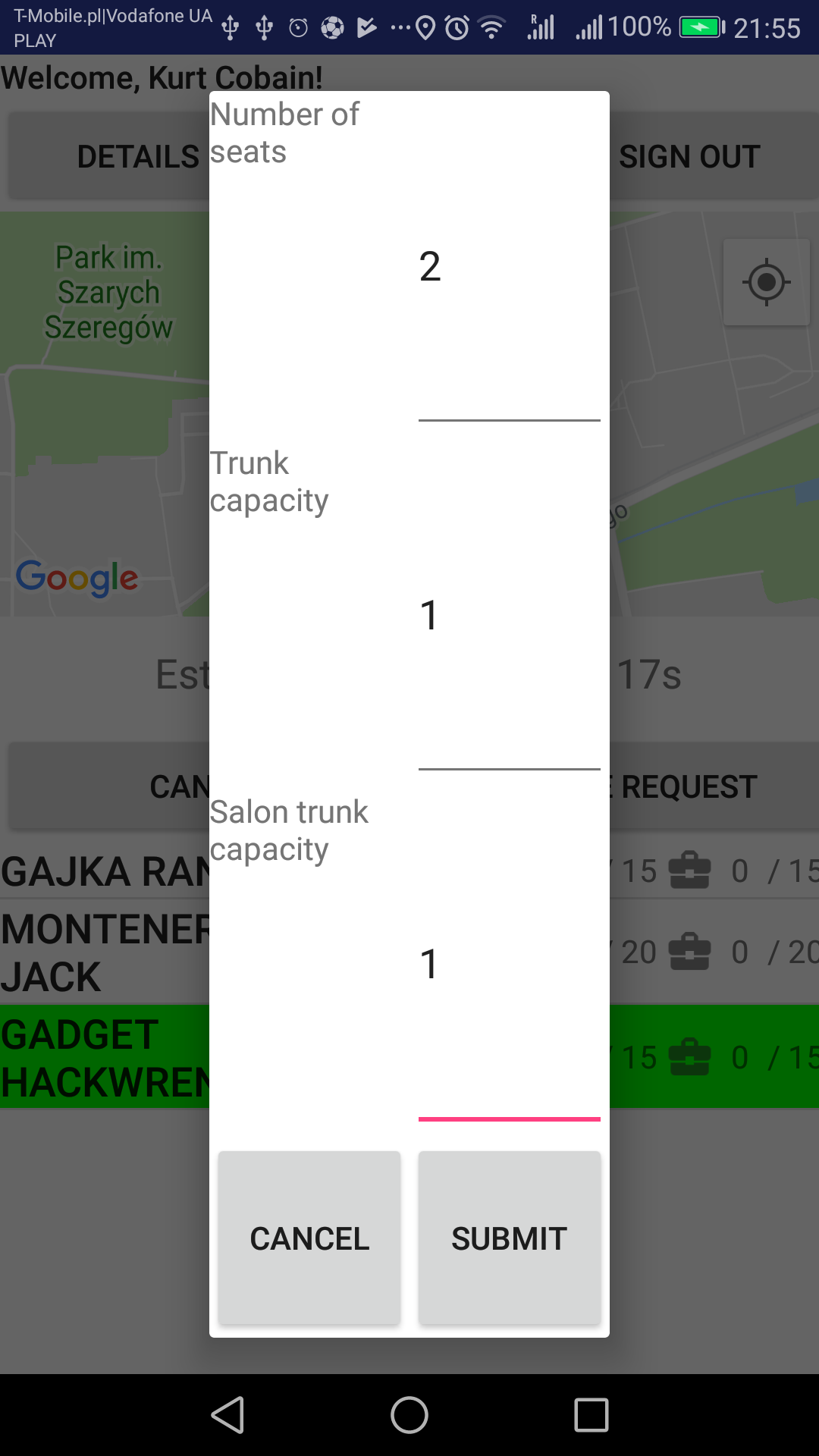
4.3.4 Dodanie nowego Żądania do wybranego Kierowcy

Po wybraniu Kierowcy z listy na Głównym Widoku (rozdział 3.3.2), Klient ma możliwość dodania nowego Żądania. Dlatego na mapie on wybiera Punkt Początkowy dla Żądania oraz Punkt Końcowy, które znajdują się na trasie, wtedy na Widoku pojawia się możliwość wycofania lub dodania Żądania (rys. 1.31).



Rys. 1.31 Dodanie Punktów Startu i Dojazdu dla Żądania na Głównym Widoku Klienta [1]

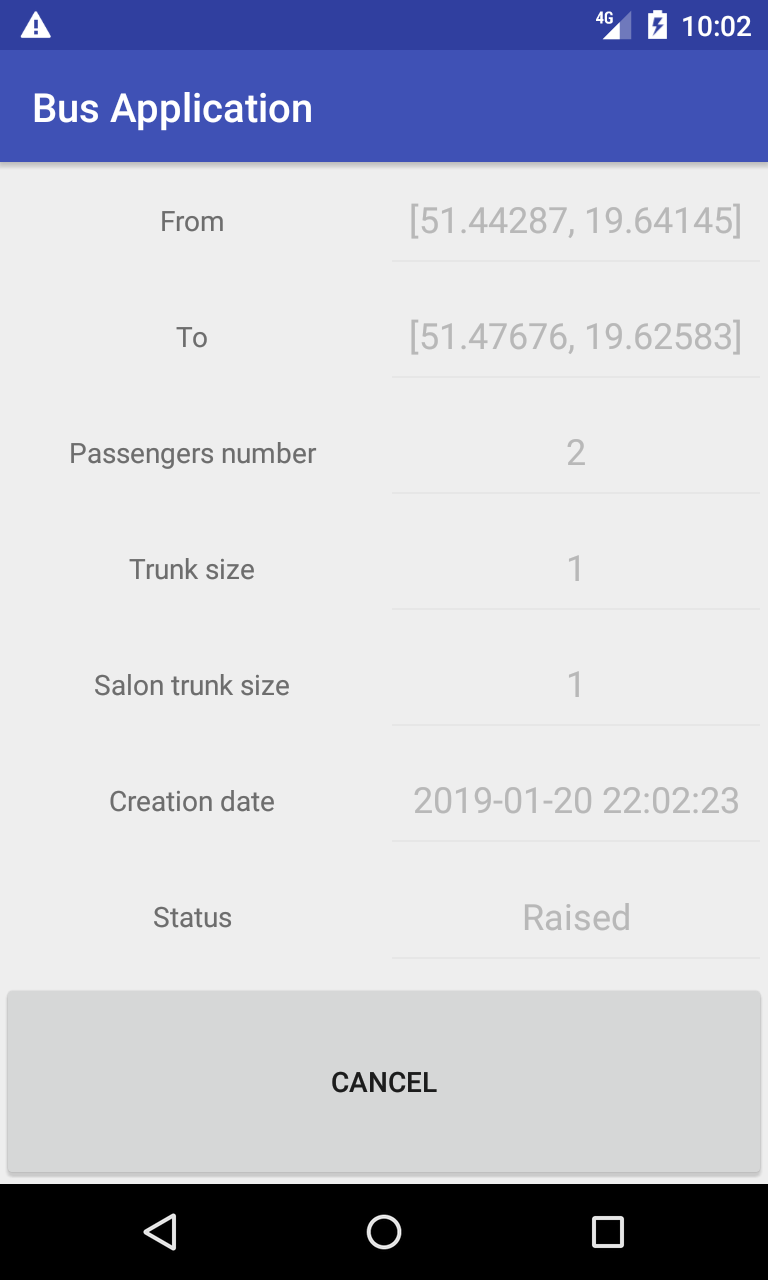
Gdy lokalizacje się zgadzają, Klient naciska „Utwórz Żądanie”. Wtedy pojawia się formularz, na którym on uzupełnia niezbędne dane dla Kierowcy – ilość miejsc, bagażu oraz bagażu podręcznego (rys. 1.32).



Rys. 1.32 Uzupełnienie danych przejazdu dla Żądania [1]

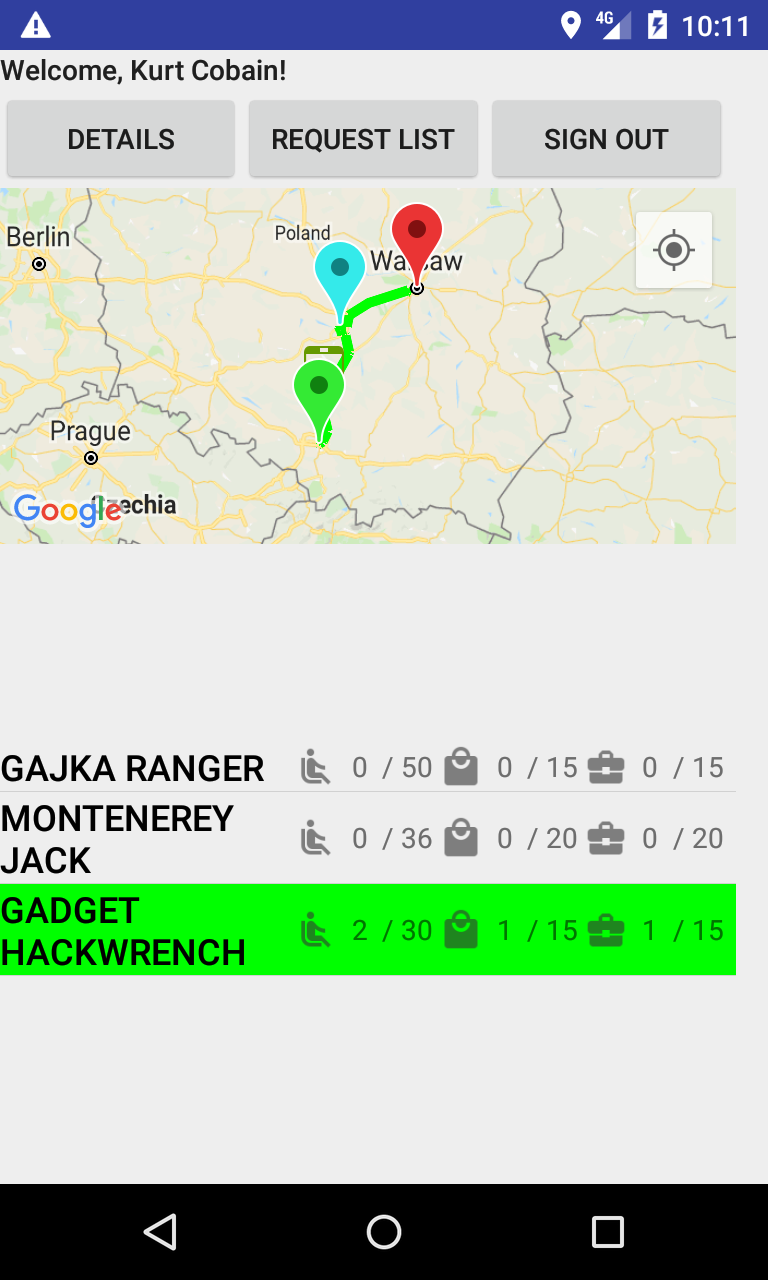
Gdy Klient nie jest pewien w poprawności lokalizacji wybranych na poprzednim etapie, to jest możliwość wycofania przez odpowiedni przycisk. Po wprowadzeniu danych i naciśnięciu „Zatwierdź”, Żądanie jest zapisane w Bazie Danych i jest wysłane do odpowiedniego Kierowcy.

Po zapisaniu, Klient może sprawdzić jego szczegóły (rys. 1.33). Automatycznie Żądania tworzy się w statusie „Otworzone”.



Rys. 1.33 Szczegóły zgłoszonego przez Klienta Żądania[1]

Po akceptacji Żądania przez Kierowcę, uzupełniają się dane stanu autobusu (zajęte miejsca, ilość bagażu w bagażniku oraz bagażu podręcznego w salonie) o informację, podaną w Żądaniu, co jest widoczne na rys. 1.4.



Rys. 1.34 Uzupełnienie danych autobusu Kierowcy po akceptacji Żądania, zgłoszonego przez Klienta[1]

4.3.5 Anulowanie Żądania przez Klienta

W przypadkach, ilustrowanych na rysunkach 1.28 oraz 1.33 Klient ma możliwość anulowania Żądania. Po naciśnięciu odpowiedniego przycisku przez Klienta, Żądania przechodzi w status „Anulowano przez Klienta” i zapisuje się w bazie danych. Kierowca w tym momencie jest powiadomiony o zmianie statusu Żądania Po przejściu na Szczegóły tego Żądania jest widoczna zmiana statusu oraz brak możliwości anulowania tego Żądania (rys.1.35).



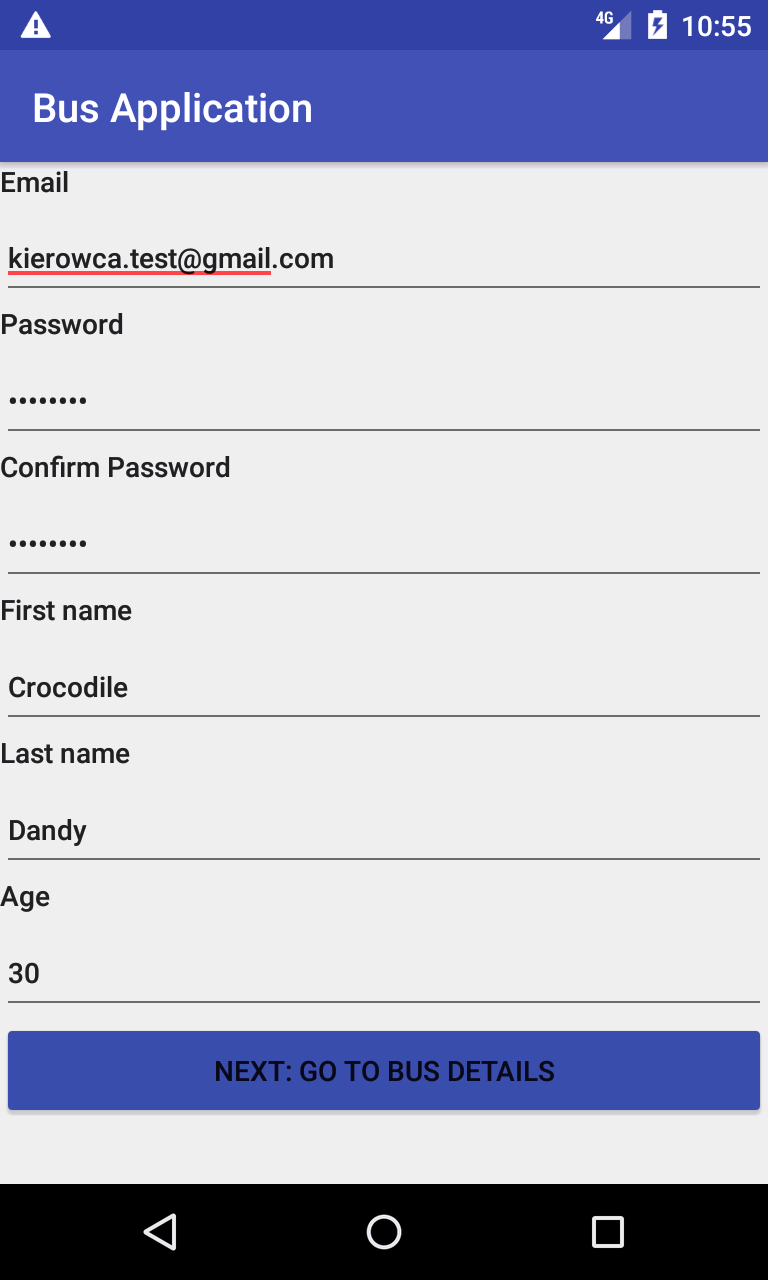
Rys. 1.35 Szczegóły Żądania Klienta po jego anulowaniu [1]

5 TESTOWANIE PROJEKTU

Zaczynamy od testowania przypadku rejestracji. Po otworzeniu aplikacji spotyka nas widok logowania (rys.5.1). Przy rejestracji Kierowcy wymagane są dane o użytkowniku (konto Google) oraz haśle (wraz z konfirmacją), potem dane osobowe – imię, nazwisko oraz wiek. Podajemy te dane (rys.5.2).

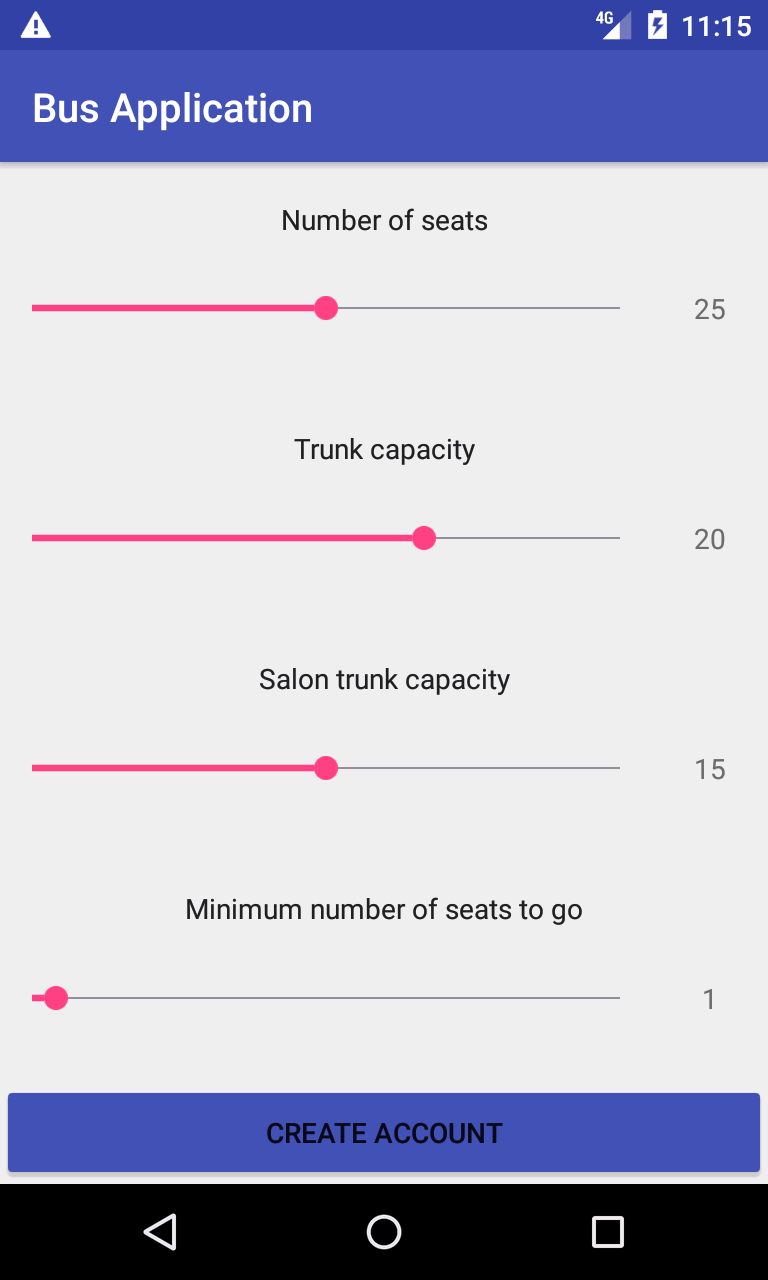


Rys.5.1 Widok startowy aplikacji [1]



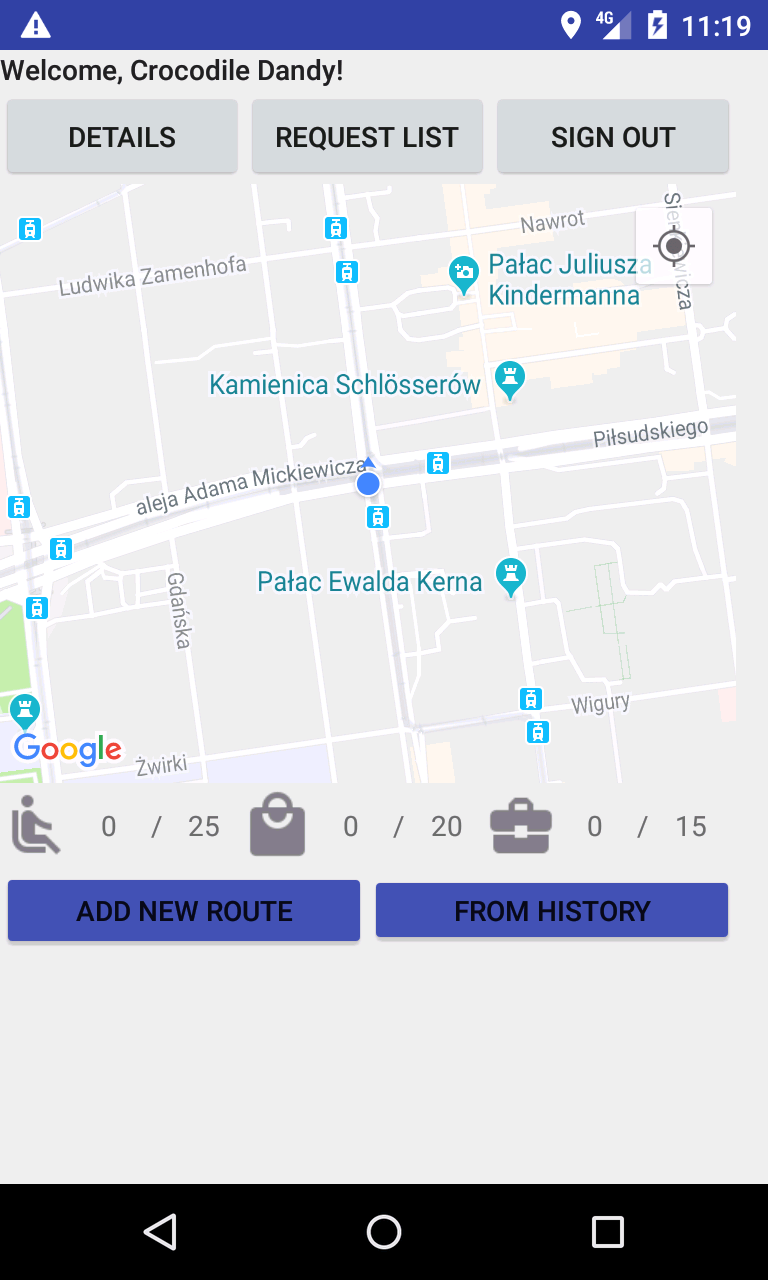
Rys.5.2 Testowanie przypadku rejestracji Kierowcy, użytkownik oraz dane osobowe[1]

Po określeniu danych parametrów, klikamy „Idź do szczegółów autobusu” (eng. „go to bus details”). Po naciśnięciu, trafiamy do widoku, gdzie określamy takie parametry, jak ilość miejsc w autobusie, ilość jednostek bagażu, które mogą się zmieścić w bagażniku, miejsce w salonie na bagaż oraz minimalny próg zajętych miejsc niezbędnych do rozpoczęcia jazdy zdefiniowaną Trasą (rys. 5.3).



Rys.5.3 Testowanie przypadku rejestracji Kierowcy, szczegóły autobusu [1]

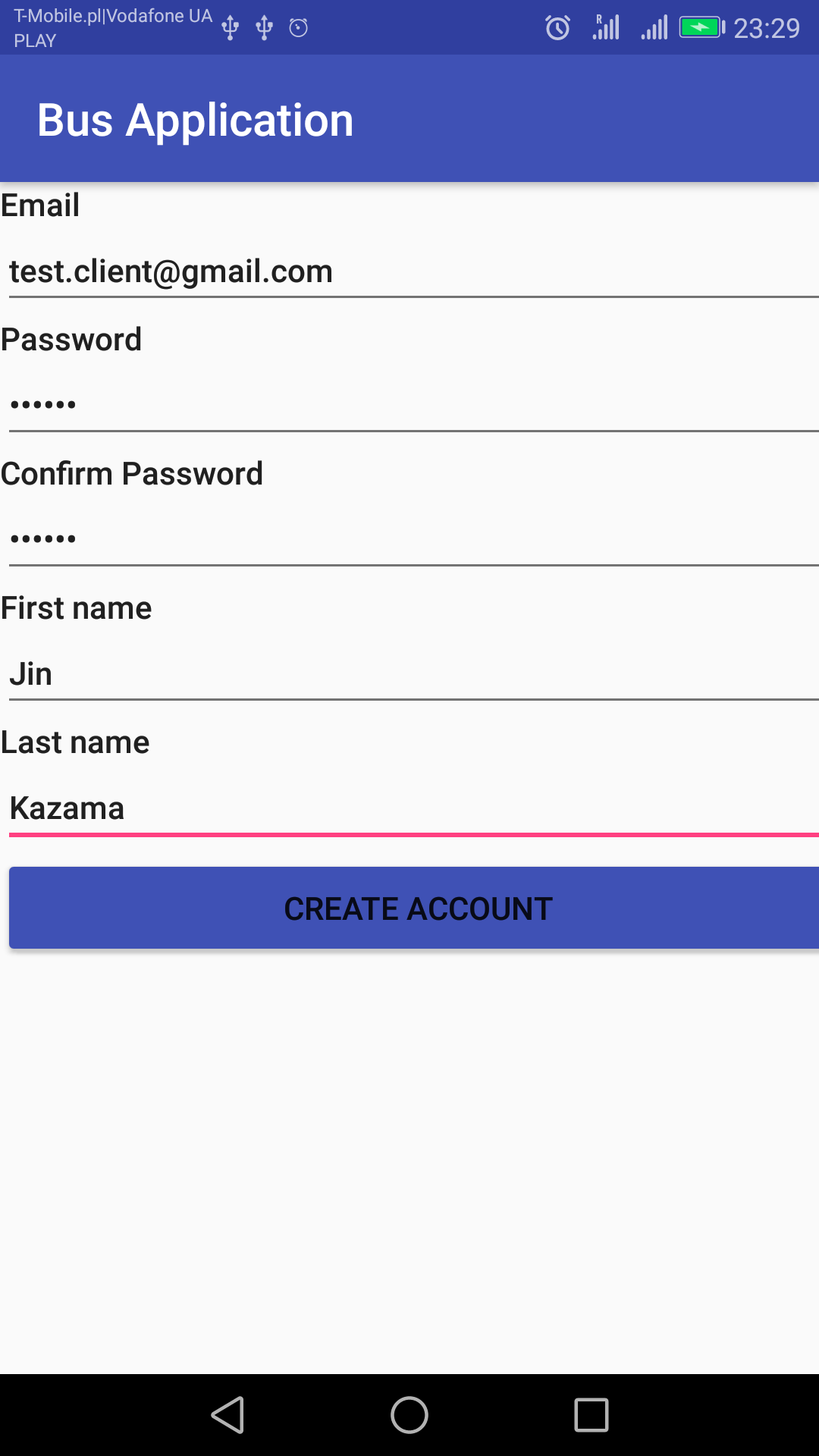
Po naciśnięciu przycisku „Create account”, jesteśmy przekierowani do głównej strony Kierowcy w aplikacji (rys.5.4).



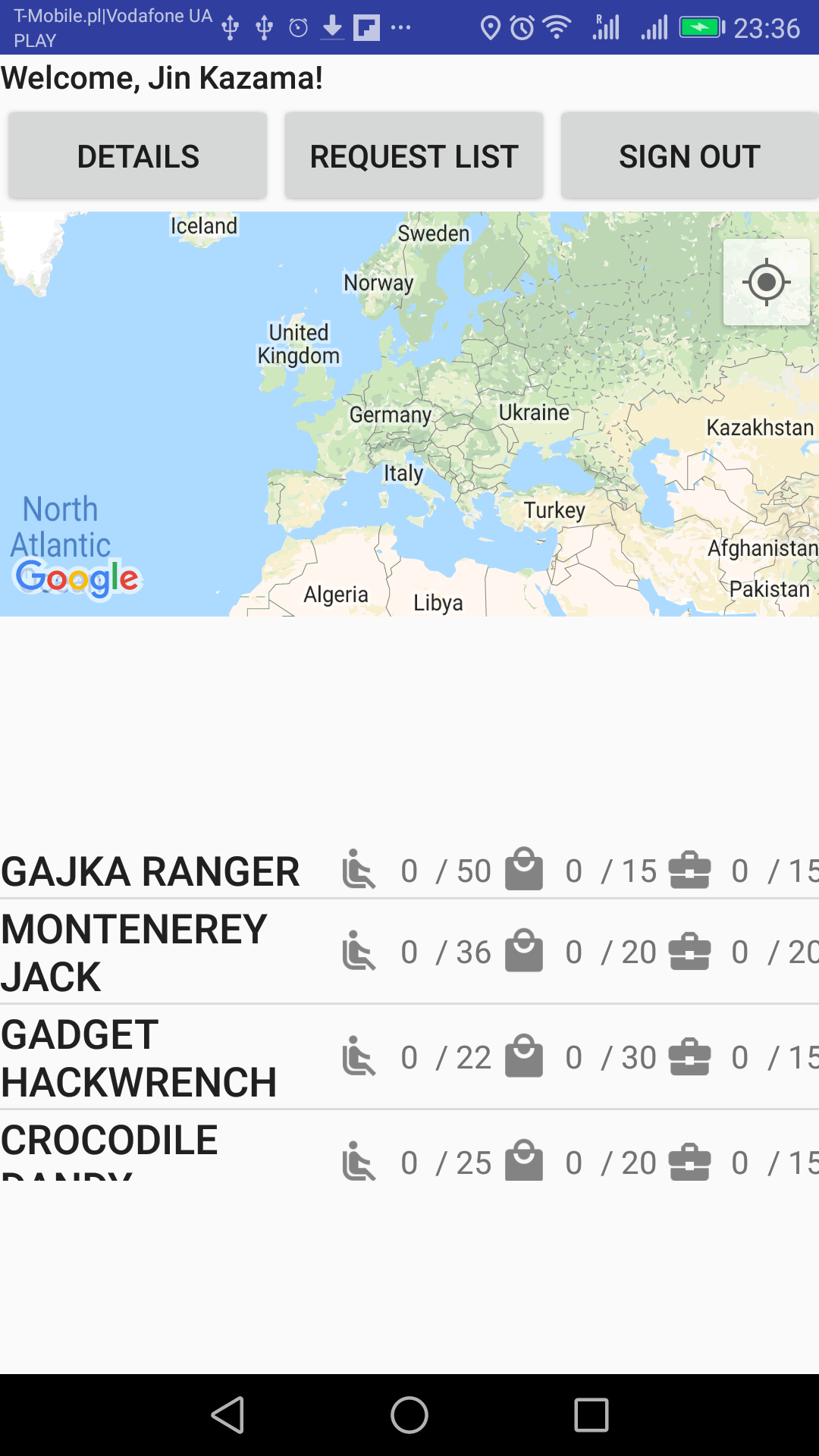
Rys.5.3 Główny widok Kierowcy [1]

W przypadku rejestracji Klienta, z ekranu powitalnego (rys.5.1) naciskamy „Sign in as client”, po czym w nowym widoku podajemy jak w przypadku Kierowcy użytkownika, hasło oraz imię, nazwisko i wiek (rys.5.4). Po naciśnięciu „Submit” przed nami pojawia się główny widok Klienta (rys.5.5). Na tym widoku także na liście Kierowców, można zauważyć niedawno zarejestrowanego użytkownika z poprzedniego kroku (na samym dole widoku).

Kolejnym krokiem będzie dodanie nowej trasy dla Kierowcy. Zalogujemy się z widoku autoryzacji (rys.5.1), podajemy email oraz hasło dla Kierowcy oraz klikamy „Login”, po czym z widoku głównego (rys.5.3) wybieramy punkt „Add new Route”.

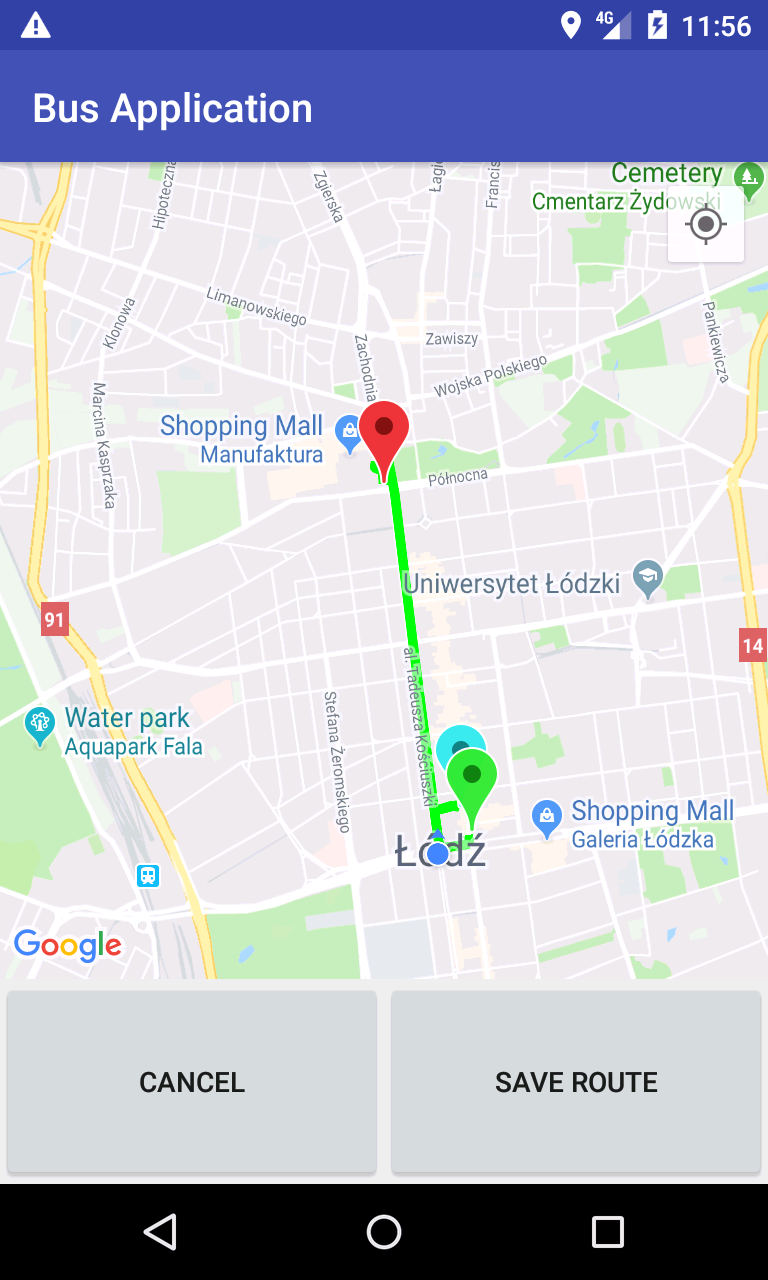


Rys.5.4 Testowanie przypadku rejestracji Klienta[1]

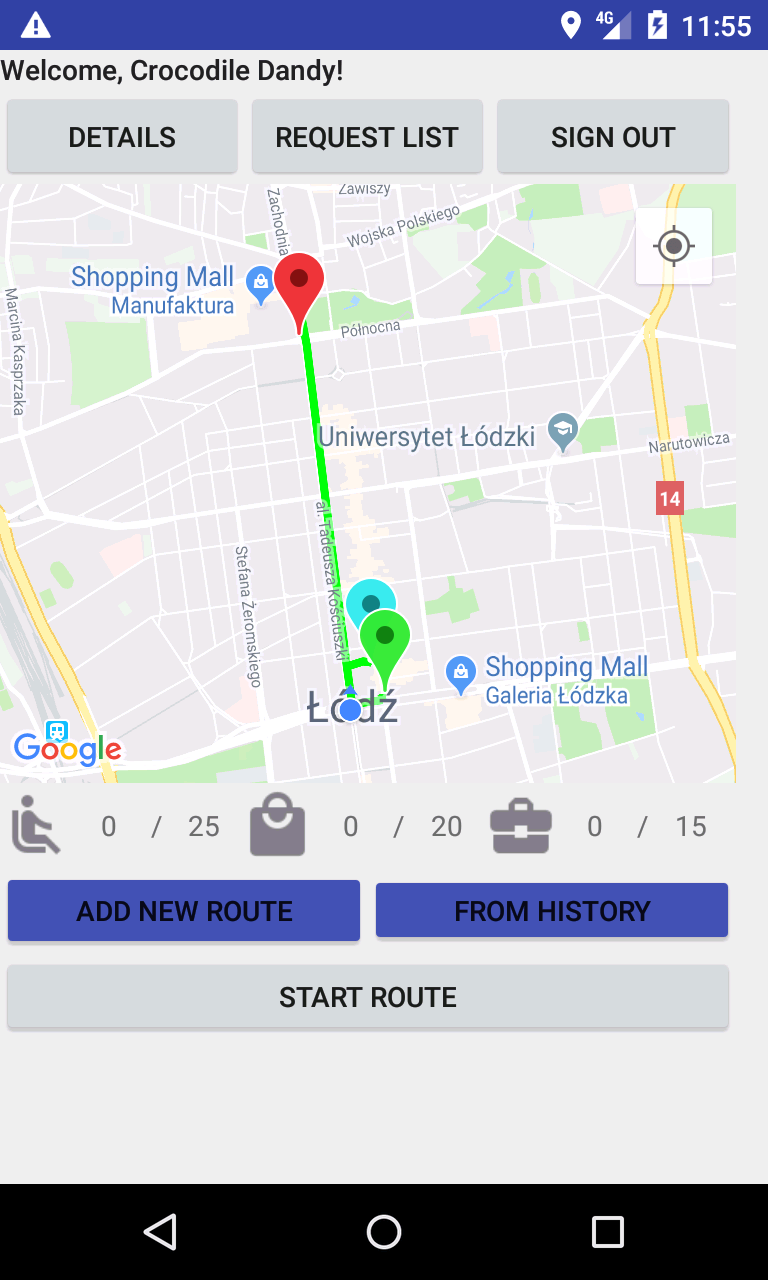


Rys.5.5 Główny widok Klienta [1]

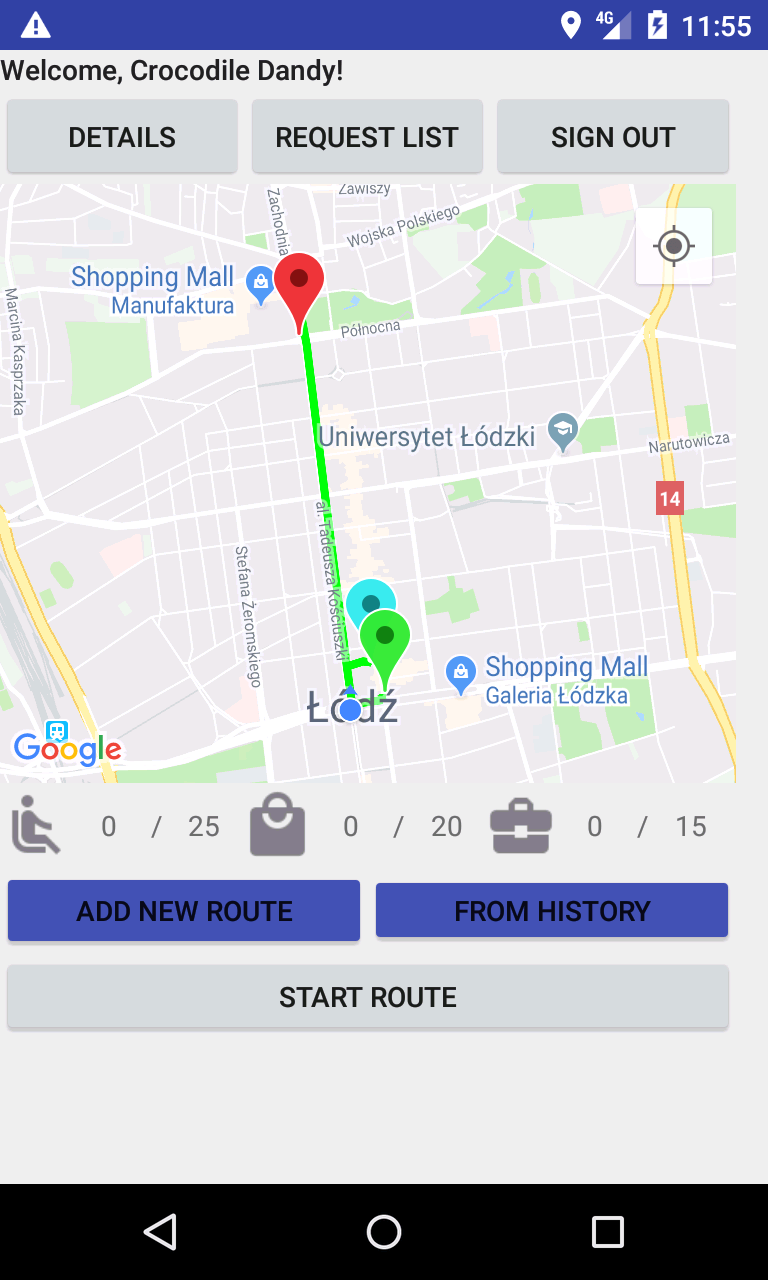
Na widoku mapy dodajemy po kolei punkty (założymy, że w danym przykładzie Kierowca organizuje przejazd między zabytkami „Kamienica Schlösserów”, „Pałac Juliusza Kindermanna” oraz „Muzeum Miasta” w mieście Łódź; rys.5.6) i klikamy „Save Route”. Widok Kierowcy jest automatycznie uzupełniony dodaną Trasą (rys.5.7). Po wybraniu danego Kierowcy z listy na widoku Klienta pokazuje aktualną Trasę oraz bieżącą lokalizację Kierowcy (rys.5.8).



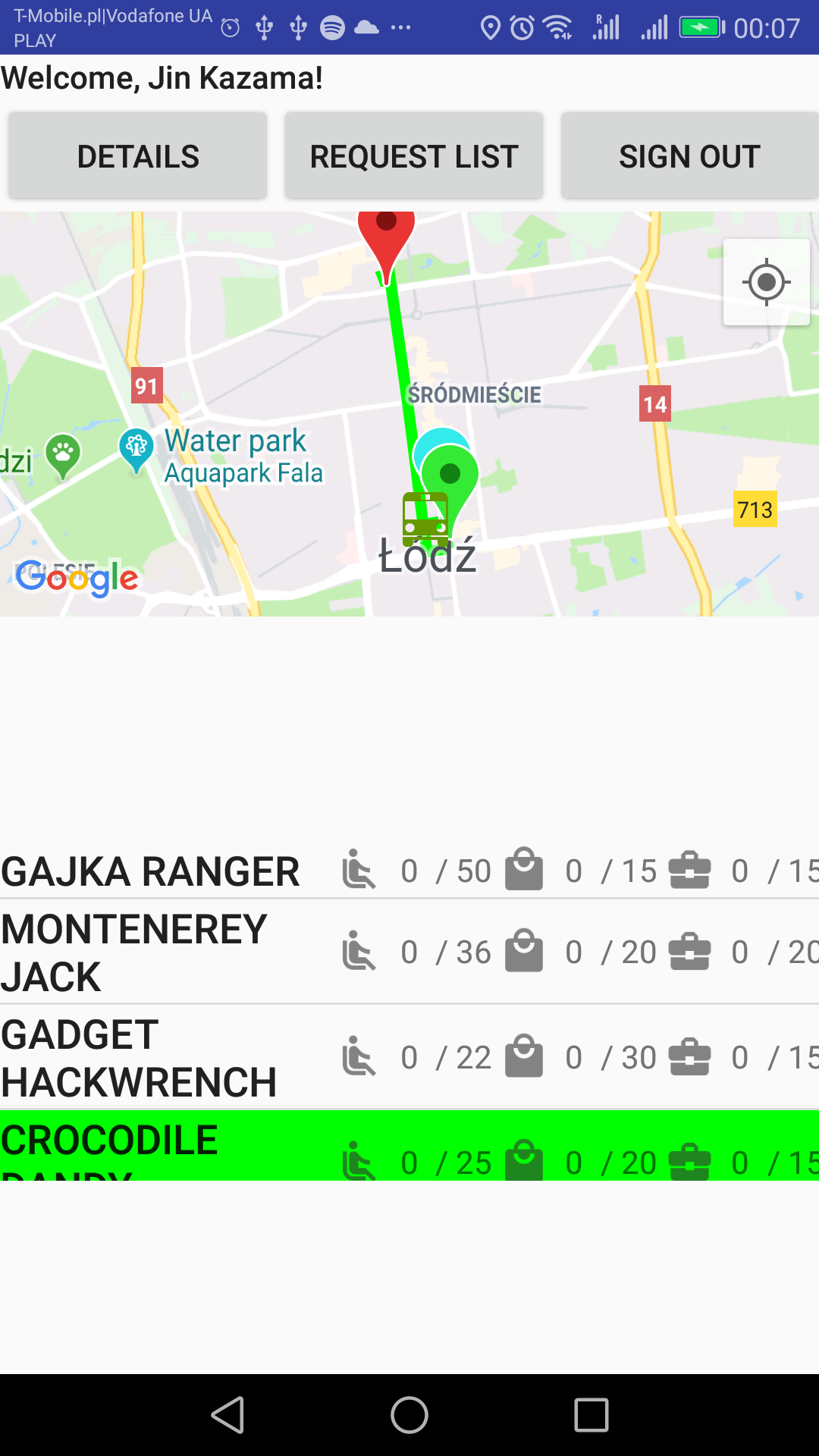
Rys.5.6 Testowanie dodania Trasy przez Kierowcę [1]



Rys.5.7 Widok Kierowcy po dodaniu nowej Trasy [1]



Rys.5.7 Widok Kierowcy po dodaniu nowej Trasy [1]



Rys.5.8 Widok Klienta po dodaniu przez wybranego Kierowcę nowej Trasy [1]

6 WDRAŻANIE PROJEKTU

7 PODSUMOWANIE

8 BIBLIOGRAFIA

[1]. https://www.msit.gov.pl/pl/turystyka/badania-rynku-turystycz/statystyka-komunikaty-i/7855,Charakterystyka-podrozy-mieszkancow-Polski-w-2017-r.html (sprawdzono dnia 2019.01.05)

[12] <https://android.com.pl/news/39909-historia-androida-w-pigulce-infografika-przedstawiajaca-najwazniejsze-informacje/> (sprawdzono dnia 14-02-2019)

[13] <https://komorkomania.pl/35271,android-8-1-oreo-co-nowego> (sprawdzono dnia 14-02-2019)

[14] <https://antyweb.pl/android-studio-to-swietne-narzedzie-dla-programistow-tworzenie-nowych-aplikacji-bedzie-teraz-znacznie-szybsze/> (sprawdzono dnia 14-02-2019)

[15] <http://www.altcontroldelete.pl/artykuly/java-charakterystyka-platformy-i-jezyka/> (sprawdzono 14-02-2019)

[16] <https://gmapsapi.com/> (sprawdzono 16-02-2019)

[17] <https://itiq.pl/marketing/czym-firebase-marketing-aplikacji-mobilnych-unity/> (sprawdzono dnia 16-02-2019)

9 SPIS RYSUNKÓW

10 STRESZCZENIE PRACY