Uniwersytet łódzki

Wydział Matematyki i Informatyki



PRACA DYPLOMOWA  
Aplikacja moblina do zapisywania wspomnień z wycieczek

Autor: Michał Mikła  
Kierunek: Informatyka  
Specjalizacja: Sieci Komputerowe i Przetwarzanie Danych  
NR Indeksu: 353764  
Promotor: dr Sebastian Lindner

Spis treści:

Wstęp  
Słowniczek  
Wybór narzędzia  
Struktura aplikacji  
Zaznaczanie pozycji na mapie

Wstęp

Aplikacje mobilne w dzisiejszych czasach mają bardzo szerokie zastosowanie. Dostarczają użytkownikom bardzo przydatną funcjonalność na przykład możliwości sprawdzenia rozkładu jazdy autobusu czy wyznaczenia trasy do konkretnego miejsca. Tematem mojej pracy jest aplikacja działająca w systemie mobilnym Android. Użytkownicy aplikacji będą mogli nanosić cyfrowe „wspomnienia” na mapę i odtwarzać je w dowolnej chwili. Dzięki napisaniu odpowiedniego kodu udało mi się stworzyć aplikację, która może służyć jako mobilny pamiętnik np. z wakacji lub wycieczek krajoznawczych gdzie możemy zaobserwować wiele ciekawych widoków godnych zapisania na dłużej i pokazania znajomym.

Wymagania sprzętowe:

Aby korzystać z aplikacji i mieć dostęp do jej pełnej funkcjonalności musimy posiadać Smartfon z funkcjami takimi jak:

- System Android 5.0

- Możliwość łączności bezprzewodowej z siecią internetową

- Aparat

- Możliwość łącznośco z systemem GPS

Słowniczek:

GPS (ang. Global Positioning System) – System bazujący na satelitach, który dzięki pomiarom odległości między odbiornikiem a satelitami (minimum trzema), wyznacza położenie na kuli ziemskiej.

Smart Watch – Jest to inteligentny, elektroniczny zegarek na rękę posiadający wbudowany system (np. Android lub IOS) mający dużo bardziej rozszerzoną funkcjonalność niż zwykłe zegarki. SmartWatch umożliwia np sprawdzenie od kogo otrzymaliśmy wiadomość lub odebranie połączenia czy nawet korzystanie z przeglądarki internetowej.

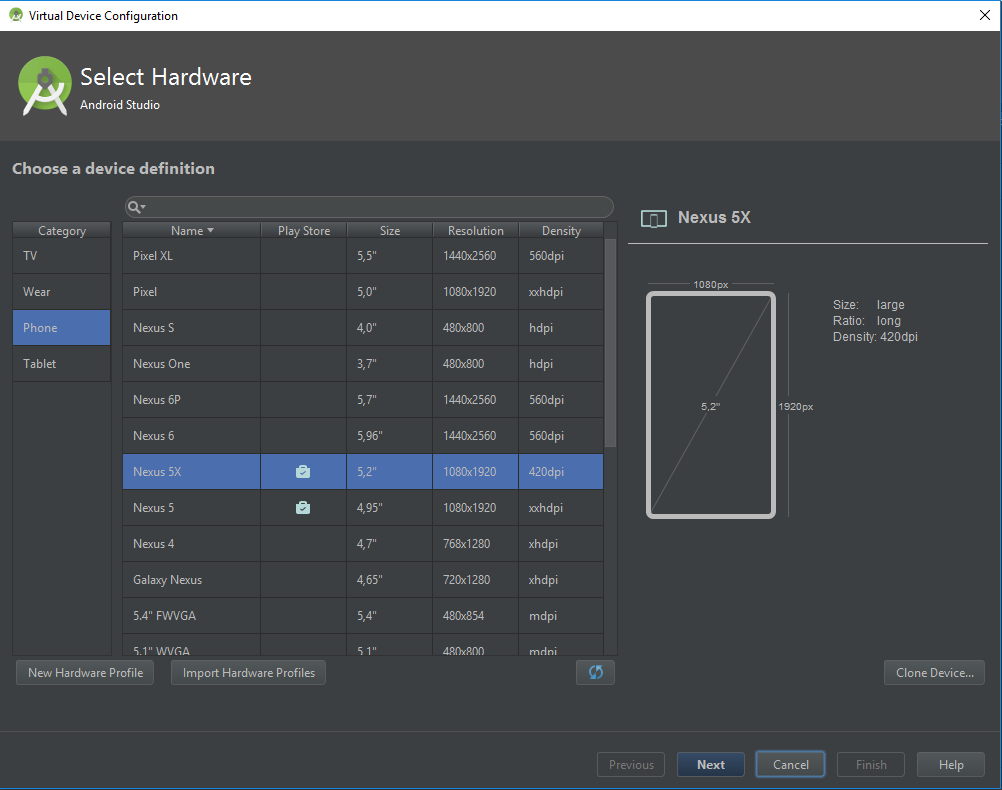
SQL – 8 strona

CRUD – 8 strona

INTENT

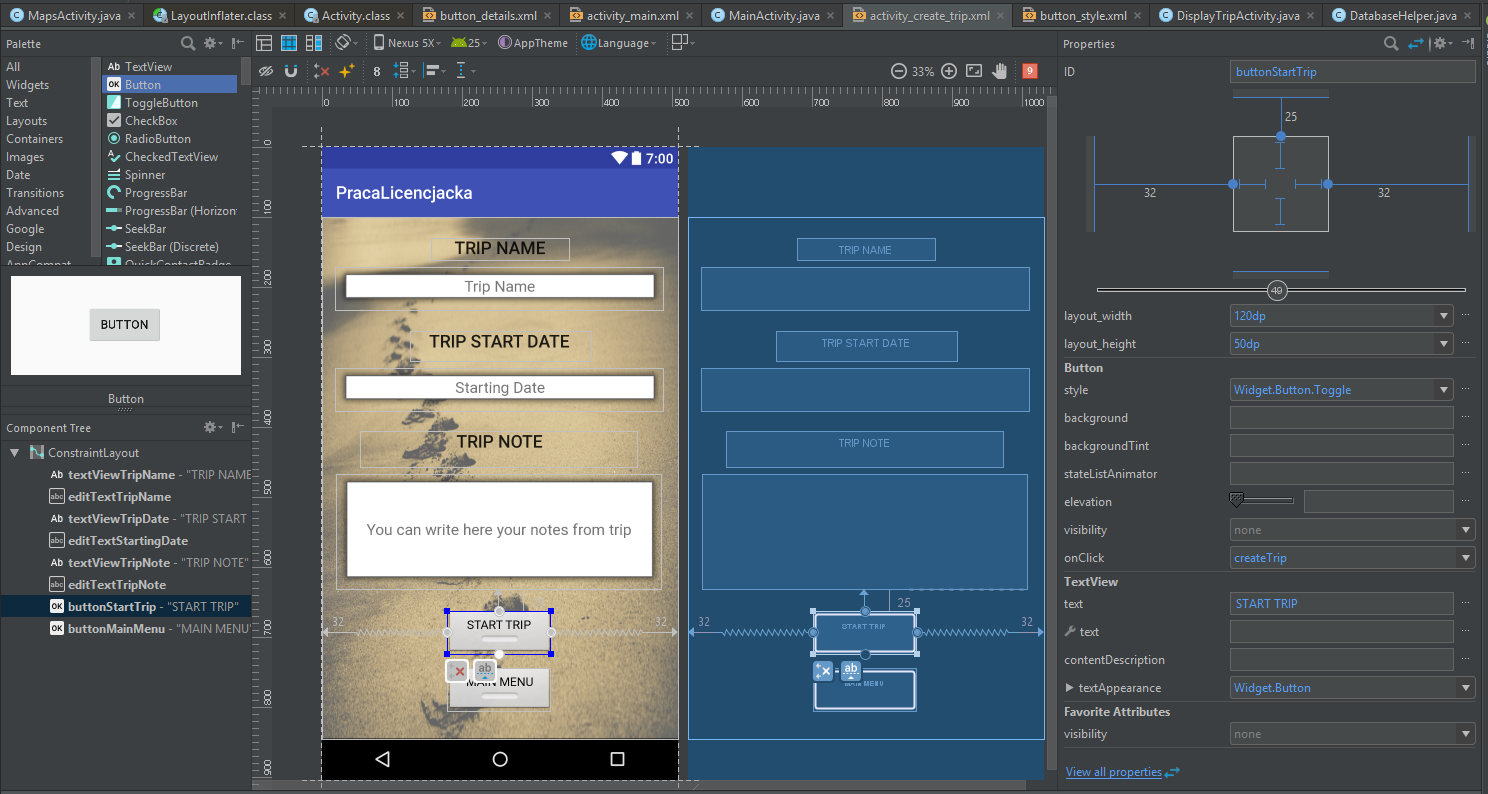
Wybór narzędzia

Narzędziem, które wybrałem do tworzenia mojej aplikacji jest stworzona przez Google platforma przeznaczona dla developerów Androida o nazwie Android Studio. Do wyboru miałem jeszcze Microsoft Visual Studio lecz zdecydowałem się wybrać TĘ pierwszą ze względu na jej duża przejrzystość oraz większą wydajność. Ogromną zaletą Android Studio jest to, że mamy dostęp do pełnej funkcjonalności całkowicie darmowo. Językiem programowania jest JAVA, tworzenie kodu jest znacznie ulepszone dzięki systemowi Inteligentnej Edycji Kodu który podpowiada nam ewentualne możliwości wybrania metod lub komponentów, czy też pomaga w zabezpieczeniu przed ewentualnymi błędami. Poniżej pozwolę opisać sobie kilka niezmiernie ciekawych i pożytecznych funkcjonalności.

Jedną z ciekawych funkcjonalności AndoidStudio jest Android Virtual Device Manager, dzięki któremu mamy możliwość stworzenia wirtualnego emulatora urządzenia na które będziemy tworzyć aplikację. AVD pozwala nam na emulowanie urządzeń takich jak SmartWatche, tablety, telewizory oraz telefony wraz z całą ich konfiguracją taką jak rozdzielczość, wersja systemu android czy wielkość ekranu.

Kolejną z ciekawych funkcjonalności Android Studio jest SDK Tools dzięki której możemy bez żadnych problemów pobierać i aktualizować za pomocą naszego kompilatora nowe wersje platform czyli wersji systemu android oraz często niezbędnych do rozszerzania działania naszych aplikacji bibliotek. W mojej pracy stosowałem SDK Tools w celu pobrania platformy w wersji 5, oraz takich bibliotek jak np. Google Play Services która umożliwia łączenie się z usługami takimi jak np. Mapy Google.

Ostatnią niezmiernie pożyteczną rzeczą jakie posiada nasz kompilator jest graficzny system umożliwający tworzenie wyglądu zewnętrznego tzw layoutu. Często tworzenie za pomocą kodu elementów takich jak przyciski, miejsca do wpisania lub wyświetlania tekstu jest uciążliwe ze względu na dużą ilość danych takich jak położenie elementu w konkretnym miejscu ekranu, jego wysokość, szerokość, stałe ustalenie względem krawędzi które musimy definiować. Graficzne tworzenie interfejsów usprawnia nam te żmudne zadania w znaczny sposób gdyż mamy możliwość wybrania sobie komponentu z listy, umieszczenia go na ekranie i dokładniego ustalenia położenia względem krawędzi ekranu lub innych komponentów czy wyglądu nie pisząc przy tym ani jednej linii kodu. Oczywiście jest możliwość tworzenia layoutów w trybie tekstowym ale moim zdaniem może to znacznie spowalniać pracę nad projektem.



Zalety Android Studio :

- Całkowicie darmowa dostępność do pełnej funkcjonalności.  
- Możliwość korzystania z systemu kontroli wersji.  
- Bardzo duża ilość informacji powszechnie dostępnych w internecie.  
- Możliwość emulowania wirtualnego urządzenia.  
- Łatwe pobieranie potrzebnych bibliotek i narzędzi za pomocą kompilatora.  
- Graficzne tworzenie layoutów.  
- Usprawnienia w pisaniu kodu.  
- Przejrzystość i stosunkowa łatwość obsługi nawet dla nowych użytkowników.

Wady Android Studio :

- Niestety Android Studio jest narzędziem znacznie obciążającym system, uruchomienie dodatkowego emulatora i kompilacja na nim aplikacji pobiera dużo pamięci operacyjnej więc sprawne korzystanie na słabszych komputerach jest utrudnione.

- Bardzo duża ilość aktualizacji, które zajmują sporo miejsca na dysku.

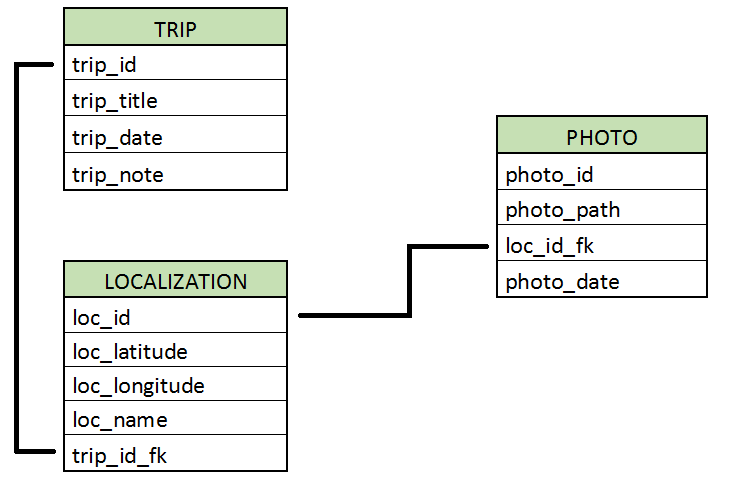
Baza danych

Pojęcie bazy danych

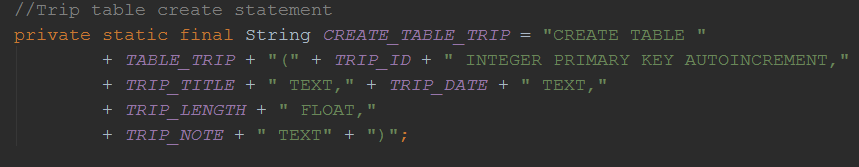
Baza danych to w najprostszym wytłumaczeniu pewnego rodzaniu kontener w którym możemy przechowywać informacje, najczęściej są one przechowywane w tabelach dzięki czemu mamy możliwość łatwego odwoływania się do nich za pomocą różnych poleceń zwanych kwerendami. Kwerendy są to polecenia służące do wyciągania konkretnych informacji z bazy danych ale i nie tylko, dzięki nim możemy również wstawiać dodatkowe dane, usuwać lub edytować. Dzięki zastosowaniu bazy mamy możliwość rozłożenia sobie konkretnych danych do konkretnych tabel co ułatwia późniejsze ich pozyskiwanie i formatowanie.

Struktura bazy danych w aplikacji

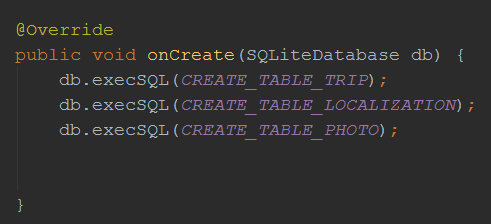
W tworzonej przeze mnie aplikacji również zastosowałem bazę danych ze względu na dużą ilość informacji do których potrzebuję mieć dostęp w trakcie działania programu, dzięki bazie mogę się do nich odwoływać w każdym momencie działania bez obawy, że zostaną one nagle utracone bez możliwości ich ponownego wydobycia. W mojej bazie danych zastosowałem 3 tabele które przechowują informacje o Wycieczce (trip), Lokalizacji (localization) oraz Zdjęciach (photo), tabele zostały połączone ze sobą za pomocą kluczy obcych dzięki czemu mam możliwość odczytu np lokalizacji z konkretnych wycieczki co bardzo usprawnia późniejsze działanie oraz obsługę aplikacji.



Tworzenie bazy danych w aplikacji

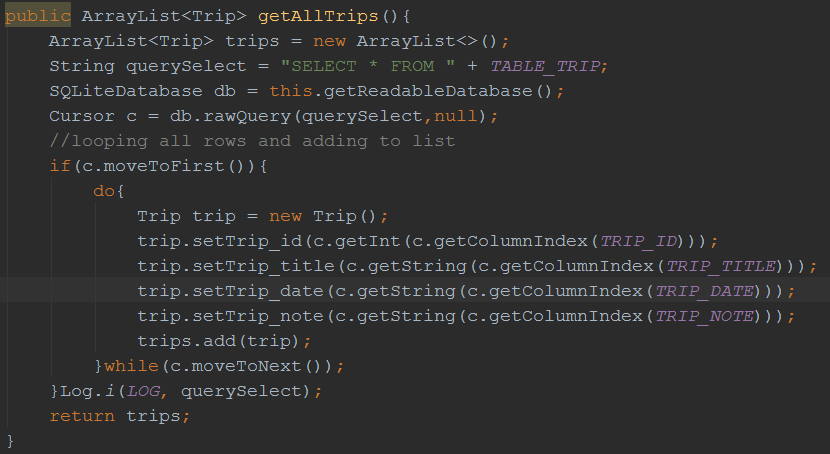
W mojej aplikacji postanowiłem stworzyć bazę danych lokalnie, czyli w pamięci urządzenia ze względu na to, że nie jest ona zbyt duża i przechowuję w niej tylko dane tekstowe więc nie ma problemu z przepełnianiem się pamięci. W celu jej obsługi utworzyłem klasę DatabaseHelper którą rozszerzyłem o klasę SQLiteOpenHelper dostarczającą mi dodatkowe metody do obsługi. Baza danych jest tworzona za pomocą kwerend SQL które są zapisane jako stałe w klasie dzięki czemu w każdej chwili mam możliwość edycji nazw tabel lub typów danych które przechowują. Poniżej przedstawię przykładową kwerendę do utworzenia tabeli TRIP:

Aby wywołać powyższe zapytanie używamy wbudowanej funkcji do obsługi kwerend sql, a mianowicie „execSQL” wywołuje nasze zapytanie dzięki czemu tworzy nam tabelę, można oczywiście w zmienną przekazywaną do funkcji całe zapytanie lecz byłoby to bardzo nieczytelne i ciężkie w edycji, w mojej klasie obsługi bazy danych Tworzenie bazy wygląda następująco:

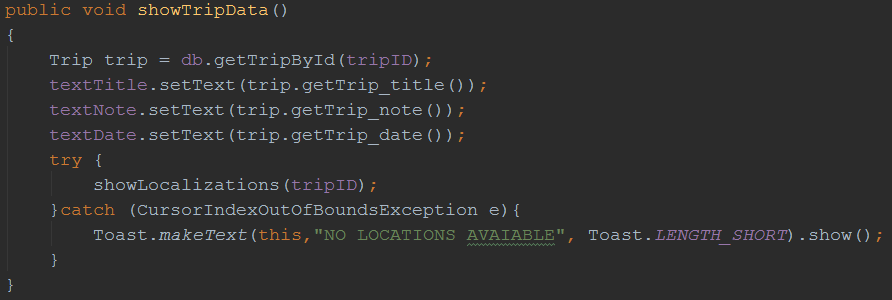


Tworzenie tabel zostało umieszczone we wbudowanej funkcji onCreate(), która jest uruchamiana za każdym razem gdy wywołuję klasę DatabaseHelper. Dzięki temu mam pewność że wywołanie klasy w każdym miejscu mojej aplikacji utworzy mi nową bazę danych lub w przypadku istnienia przywoła mi całą jej strukturę wraz z danymi.

Obsługa operacji w bazie danych odbywa się w podobny sposób jak jej tworzenie, w klasie DatabaseHelper umieściłem funkcje wykonujące operacje CRUD dzięki czemu w dalszej części programu mogę wyciągnąć na przykład listę wszystkich zapisanych wycieczek jedynie przy wywołaniu jednej metody.



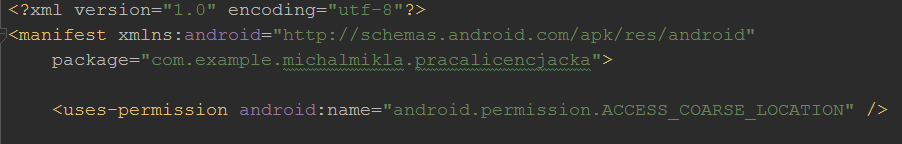
Działanie na danych z bazy

Wyżej przedstawione metody pokazują możliwość zapisu danych do bazy oraz wyciągania ich, w celu dobrego i przejrzystego zarządzania danymi utworzyłem w moim programie klasy odpowiadające tabelom istnięjącym w bazie danych. Klasy te dzięki składowym takim jak rekordy tabel pozwalają mi bez problemu utworzyć obiekt, którym mogę zarządzać w dowolny sposób dzięki implementacji metod takich jak na przykład Wyciąganie nazwy, ID, lub daty. Jest to duże ułatwienie, które zaprezentuję i opiszę na podstawie poniższego obrazu:

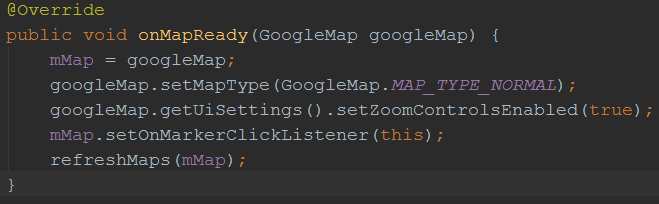
Utworzyłem obiekt klasy „Trip” i wyciągnąłem do niego odpowiednie dane konkretnej wycieczki znelzionej po numerze ID używając funkcji getTripById(). Dzięki temu mam możliwość odwoływania się do takich pól jak nazwa, notatka czy databez utworzenia bez konieczności wysyłania zapytań do w przypadku potrzeby uzyskania na przykład daty utworzenia Tripu o konkretnym numerze ID. Odnajduje to również doskonałe zastosowanie w przypadku wyciągania wielu obiektów na raz, dzięki jednemu zapytaniu wszystkie obiekty spełniające kryteria są umieszczane w liście do której mamy już później dostęp bez konieczności wykonywania wielorazowych zapytań.

GPS oraz Mapy Google

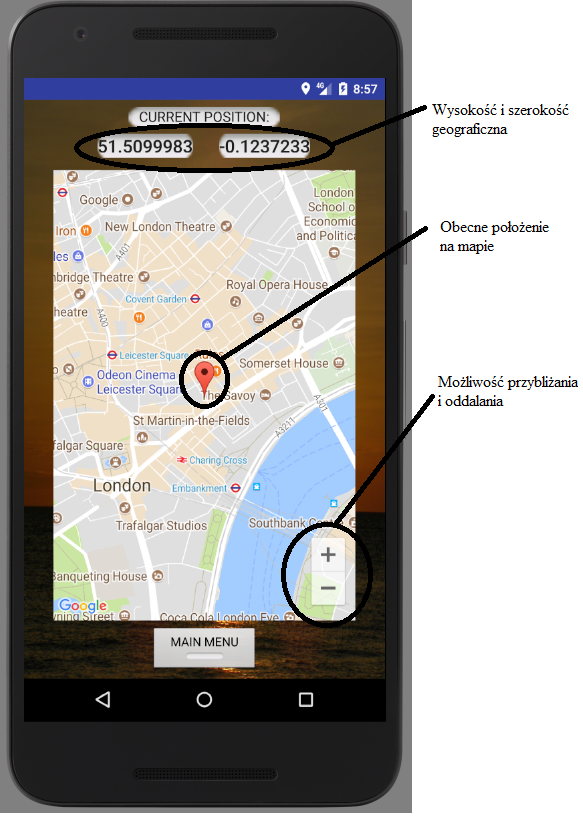
Niezbędnym elementem działania aplikacji jest GPS, dzięki jego zastosowaniu mamy możliwość obserwowania naszego położenia na mapie oraz zczytywania lokalizacji punktów. Dzięki wbudowanemu w język interfejsowi LocationListener, oraz dołączonym do niego metodom takim jak onLocationChanged możemy bez większych trudności pobierać dane, które są odczytywane przez nasz telefon z satelit i umieszczać naszą pozycję na mapie. Oczywiście niezbędnym elementem poprawnego działania aplikacji jest dodanie do pliku /manifest/AndroidManifest.xml pozwoleń na korzystanie przez aplikację z wbudowanych w urządzenie modułów lokalizacji:



Kolejnym niezbędnym elementem działania są Mapy Google na których wyświetlana zostaje nasza pozycja. Aby uzyskać dostęp do korzystania z map musimy wygenerować swój własny unikatowy klucz aplikacji na stronie <https://console.developers.google.com>, klucz ten również należy umieścić w pliku AndroidManifest.xml, oraz w pliku google\_maps\_api.xml znajdującym się w folderze values. Nieprawidłowe ustawienie klucza w aplikacji skutkuje tym, że w oknie gdzie powinna być wyświetlona struktura mapy mamy białe tło. Aby korzystać z map w programie musiałem zastosować wbudowane interfejsy takie jak OnMapReadyCallback czy ConnectionCallbacks które z kolei dostarczają metody umożliwiające zarządzeniem mapą na przykład metoda onMapReady(), która po zweryfikowaniu, że mamy poprawny klucz oraz połączenie z internetem umożliwia nam utworzenie mapy według własnych potrzeb:

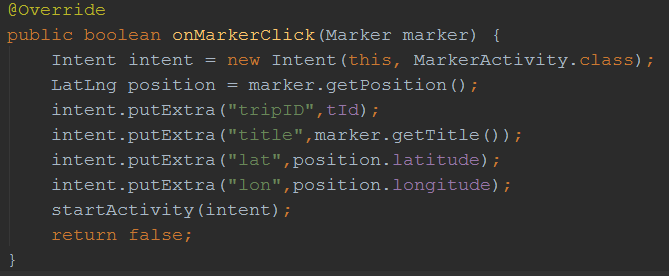


Powyższe zdjęcie przedstawia konfigurację mapy w programie, jej typ został ustawiony jako „MAP\_TYPE\_NORMAL”, czyli taki jak widzimy np w aplikacji Mapy google. Dodałem również możliwość zarządzania odległością z jakiej widzimy naszą mapę, oraz obsługę kliknięcia w nasz punkt znajdujący się na wyświetlanej mapie. W ostatniej linij możemy zauważyć napisną przeze mnie metodę „refreshMaps()”, której zadaniem jest odświeżenie naszego obecnego położenia na mapie, przesunięcie na nie kamery wyświetlającej mapę, oraz aktualizację pól tekstowych wyświetlających nasze położenie na mapie:



Zaznaczanie i zapisywanie pozycji na mapie

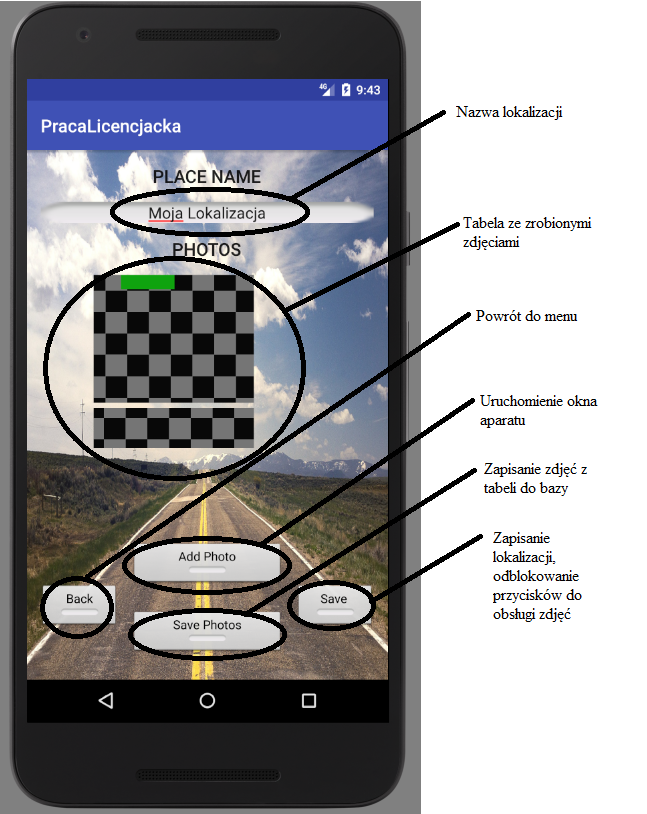
Moja aplikacja zapewnia również możliwość zaznaczenia naszego położenia na mapie i zapisania go do późniejszego obejrzenia. Wystarczy kliknąć na nasze obecne położenie na mapie i zostaniemy przeniesieni do nowego okna w którym będziemy mogli nadać nazwę tej lokalizacji, oraz dodać fotografie czegoś co nas w tym miejscu urzekło. Działanie to umożliwił mi interfejs OnMarkerClickListener, który dostarcza metodę onMarkerClick(). Kliknięcie na marker powoduje utworzenie intentu przenoszącego nas do nowego okna, dodatkowo jednoczenie przy przenoszeniu nas zostają przesłane dane pobrane z mapy takie jak wysokość i szerokość geograficzna oraz dane z bazy czyli numer identyfikacyjny obecnej wycieczki oraz jej tytuł, gdy dane te zostaną zadeklarowane w naszym intencie wystarczy za pomocą funkcji startActivity(intent) uruchomić przejście do drugiego okna czyli MarkerActivity.



Będąc już w MarkerActivity niezbędne jest zapisanie odebranych z poprzedniego okna danych, umożliwia to funkcja „getIntent()” która przechwytuje wysłany z poprzedniego Activity intentu wraz z całymi zapisanymi w nim danymi. Do zdekodowania ich utworzyłem własną metodę „recieveDataFromMap”, która przypisuje do zmiennych wartości, które zostały przesłane. Po nadaniu naszej lokalizacji tytułu możemy zapisać ją do bazy danych za pomocą przycisku „Save” uruchamiającego metodę saveLocalizationToDatabase(), od tej pory nasza lokalizacja będzie już istnieć w bazie co umożliwi nam późniejsze jej pokazanie na mapie oraz dodanie zdjęć.

Dodawanie i zapisywanie zdjęć

Niezbędną rzeczą umożliwiającą dodanie nam zdjęcia do naszej lokalizacji jest uprzednie jej zapisanie w bazie danych, dzięki temu mamy możliwość ustalenia klucza obcego wskazującego do której lokalizacji będzie należało zdjęcie. Jest to niezbędne w celu późniejszego wyświetlania sumarycznego wszystkich naszych wycieczek. Jeżeli lokalizacja zostanie już zapisana, zostają odblokowane przyciski do dodawania oraz zapisywania zdjęć. Dzięki kliknięciu w przycisk „Add photo” zostajemy przeniesieni do okna aparatu, gdzie mamy możliwość utworzenia pliku zdjęcia, zapisania go w pamięci naszego urzadzenia oraz uzyskania dokładnej lokalizacji. Posiadanie ścieżki lokalizacji do zdjęcia jest kluczowe, gdyż nasza baza nie przechowuje fizycznych plików ze zdjęciami a same ich lokalizacje w pamięci które dzięki funkcji „setPic” są dekodowane na zdjęcia i umieszczane w tabeli poglądowej. Zdjęć możemy dodawać dowolnie wiele, gdyż tabela poglądowa umożliwia przewijanie, ponadto przy utworzeniu pliku ze zdjęciem jest dodawana data jego utworzenia dzięki czemu później możemy wiedzieć dokładnie w jakim dniu i o której godzinie powstało nasze zdjęcie. Jeżeli mamy już zrobione zdjęcia, należy je zapisać za pomocą przycisku „Save photos”, w bazie danych zostają utworzone rekordy w tabeli „Photos”.



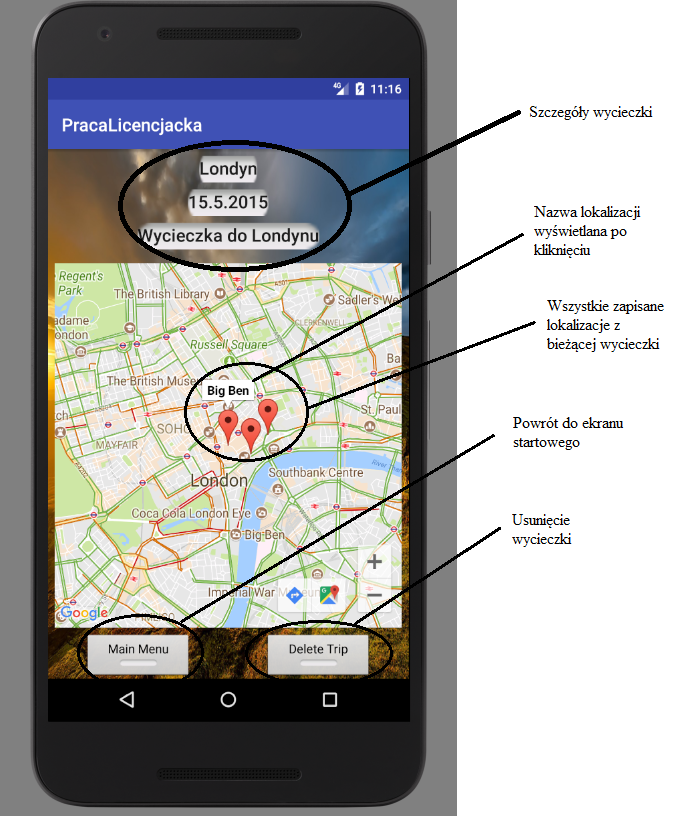
Ekran startowy

Ekran wyświetlany po uruchomieniu aplikacji (Main Activity) umożliwia nam utworzenie nowej wycieczki za pomocą przycisku „Create Trip”, który przenosi nas do kolejnego okna (Create Trip Activity) gdzie ustalamy pierwsze dane odnośnie naszej wycieczki. Drugą funkcjonalnością którą zapewnia nam to okno jest otworzenie sobie mapy bez tworzenia wycieczki, dzięki czemu możemy podejrzeć sobie nasze obecne położenie, które jest zaznaczone.   
 Kolejną funkcjonalnością jest wyświetlanie dotychczas zapisanych wycieczek. Po zakończeniu naszej wycieczki, zapisaniu wszystkich ciekawych miejsc wraz z fotografiami mamy możliwość ich odtworzenia. Wszystkie z naszych wycieczek są widoczne w tabeli menu głównego. Tabela jest wypełniana danymi zapisanymi w bazie danych, dzięki zapytaniu wybierającemu wszystkie wycieczki do listy, a następnie na podstawie listy dodwaniu ich po kolei do tabeli w startowym oknie. Tabela jest utworzona w taki sposób aby tytuły wycieczek symulowały napisy na białej tablicy w tle, natomiast przyciski dzięki którym możemy podejrzeć szczegóły konkretnych wycieczek są przeze mnie nieco zmienione. Nałożone została na nie tekstura drewnianego drogowskazu, który dodatkowo posiada animację po kliknięciu. Gdy naciśniemy na przycisk „SHOW DETAILS”, zostanie on przemieszczony w prawą stronę, uważam że zabieg ten zdecydowanie poprawia przyjemność obsługi oraz nadaje walory estetyczne. Animacja została stworzona dzięki wbudowanej funkcji AnimationListener, w której są dodakowo zaimplementowane metody badające start, koniec, oraz powtórzenie animacji. W przypadku mojego przycisku po zakończeniu animacji przesuwania zostanie wywoływana funkcja „goToShowTripActivity”, w której przesyłany jest numer ID wycieczki wyciągnięty uprzednio z bazy i uruchomione zostaje okno w którym będziemy mogli zobaczyć wszystkie szczegóły.



Podgląd szczegółów zapisanej wycieczki

Kliknięcie w drogowskaz z napisem „SHOW DETAILS” przenosi nas do nowego okna (DisplayTripActivity), gdzie możemy zobaczyć zapisane przez nas lokalizacje oraz podejrzeć zrobione tam zdjęcia i datę ich utworzenia. W przypadku braku zapisanych miejsc otrzymujemy komunikat, że nie ma dostępnych lokalizacji i mamy pogląd mapy bez naniesionych na nią punktów. W górnej części okna wyświetla się nazwa wycieczki, data jej utworzenia, oraz notatka która została nadana przy tworzeniu, wszystkie te dane są wyciągane z bazy danych na podstawie przesłanego po kliknięciu w przycisk numeru ID wycieczki. W przypadku braku lokalizacji możemy jedynie usunąć obecną wycieczkę, kliknięcie w przycisk „Delete Trip” powoduje usunięcie wpisu z bazy danych i przeniesienie nas do ekranu głównego gdzie wycieczka jest również kasowana z tabeli zapisanych pozycji. W przypadku gdy mamy zapisane lokalizacje, wyświetlana zostaje mapa wraz z naniesionymi na nią markerami które zapisalismy w trakcie podróżowania. Jednokrotne kliknięcie w punkt pokazuje nam nazwę którą nadaliśmy przy zapisywaniu.



Kolejną funkcjonalnością którą zapewnia okno szczegółów wycieczki jest możliwość obejrzenia zdjęć wykonanych w zapisanej lokalizacji, w tym celu należy kliknąć na lokalizację raz, a następnie po pojawieniu się jej nazwy należy w nią kliknąć. Efektem tego będzie uruchomienie kolejnego okna, w którym mamy możliwość oglądania zdjęć wraz z datami ich utworzenia oraz w przypadku gdy któreś z nich nam nie odpowiada usunięcia go. Ta funkcjonalność jest dostępna dzięki wbudowanej funkcji „setOnInfoWindowClickListener”, która tworzy nowy intent zawierający numer ID lokalizacji, oraz uruchamia okno gdzie możemy przeglądać zdjęcia (DisplayPhotoActivity).

Struktura Aplikacji

Moja aplikacja została stworzona na kilku klasach androidowych zwanych Activity, tworząc taką klasę automatycznie zostanie wygenerowany do niej plik layoutu z rozszerzeniem .xml dzięki któremu możemy zarządzać wyglądem konkretnego Activity. Postanowiłem podejść do tego tak, że każde wyświetlające się nowe okno aplikacji będzie nowym Activity i w jego ciele zapiszę metody odpowiednie do obsługi rzeczy w nim zawartych. Kolejnym elementem struktury jest folder „drawable” znajdujący się w katalogu „res”, w nim umieszczalem wszystkie elementy obrazów stosowanych na przykład jako tło w konkretnym activity lub zmiany wyglądu przycisku. Umieszczanie elementów graficznym w tym folderze w znaczny i łatwiejszy sposób ułatwia nam odwoływanie się do nich dzięki wpisaniu ([...@drawable/main\_background.jpeg](mailto:...@drawable/main_background.jpeg)).