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

Wstęp

Celem mojej pracy licencjackiej było stworzenie aplikacji na system mobilny android, dzięki której będziemy mieli możliwość zapisywania wycieczek uwzględniając w nich zdjęcia oraz lokalizacje ciekawych miejsc które napotkamy.

Aplikacja jest bardzo prosta w obsłudze, jedyne czego wymaga to posiadanie telefonu wersją systemu 5.0 (Kit Kat), aparatu, łącznością bezprzewodową z siecią internernetową oraz możliwością określania naszej lokalizacji przez GPS. Niestety tworzenie aplikacji mobilnych jest zadaniem dość problemowym ze względu na ogromną ilość rozbieżności między wersjami systemu co trzeba implementować w kodzie.

Dzięki połączeniu wszystkich tych funkcjonalności i napisaniu odpowiedniego kodu udało mi się stworzyć aplikację, która może służyć jako mobilny pamiętnik np. Z wakacji lub wycieczek krajoznawczych w których możemy zaobserwować wiele ciekawych widoków godnych zapisania na dłużej i pokazania znajomym.

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.

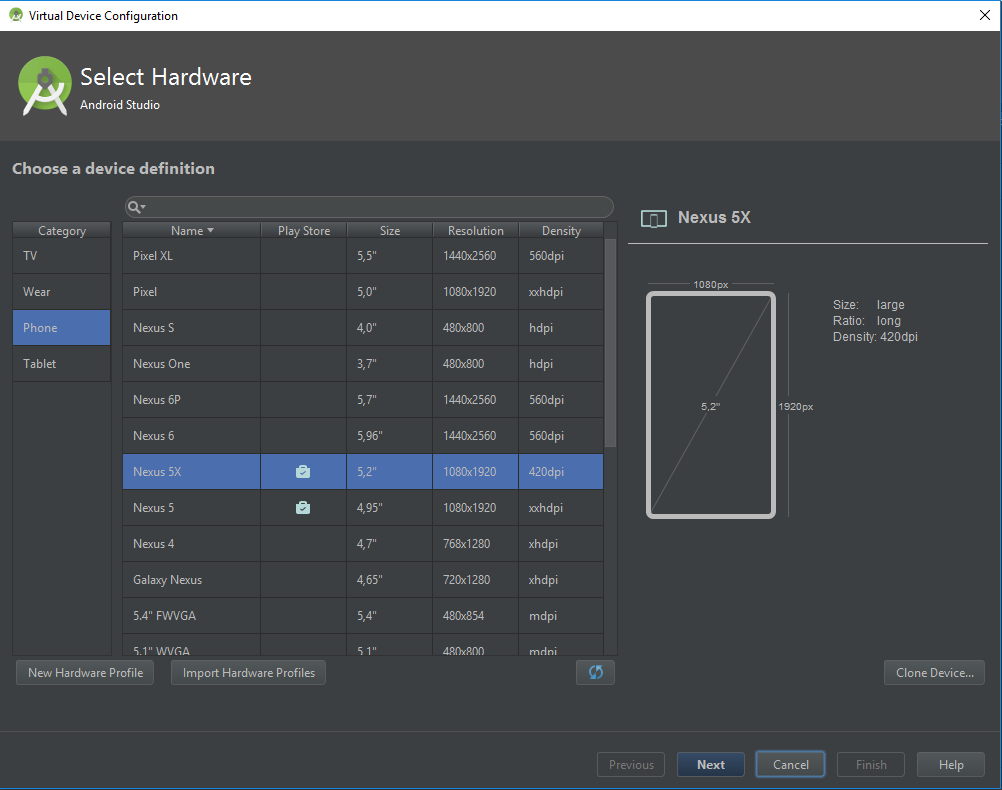
Smart Watch – Jest to inteligentny, elektroniczny zegarek na rękę posiadający wbudowany system (np. Android lub IOS) mający dużo bardziej rozszerzoną funkcjonalność niż zwykłe zegarki. SmartWatch umożliwia np sprawdzenie od kogo otrzymaliśmy wiadomość lub odebranie połączenia czy nawet korzystanie z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

SQL – 8 strona

CRUD – 8 strona

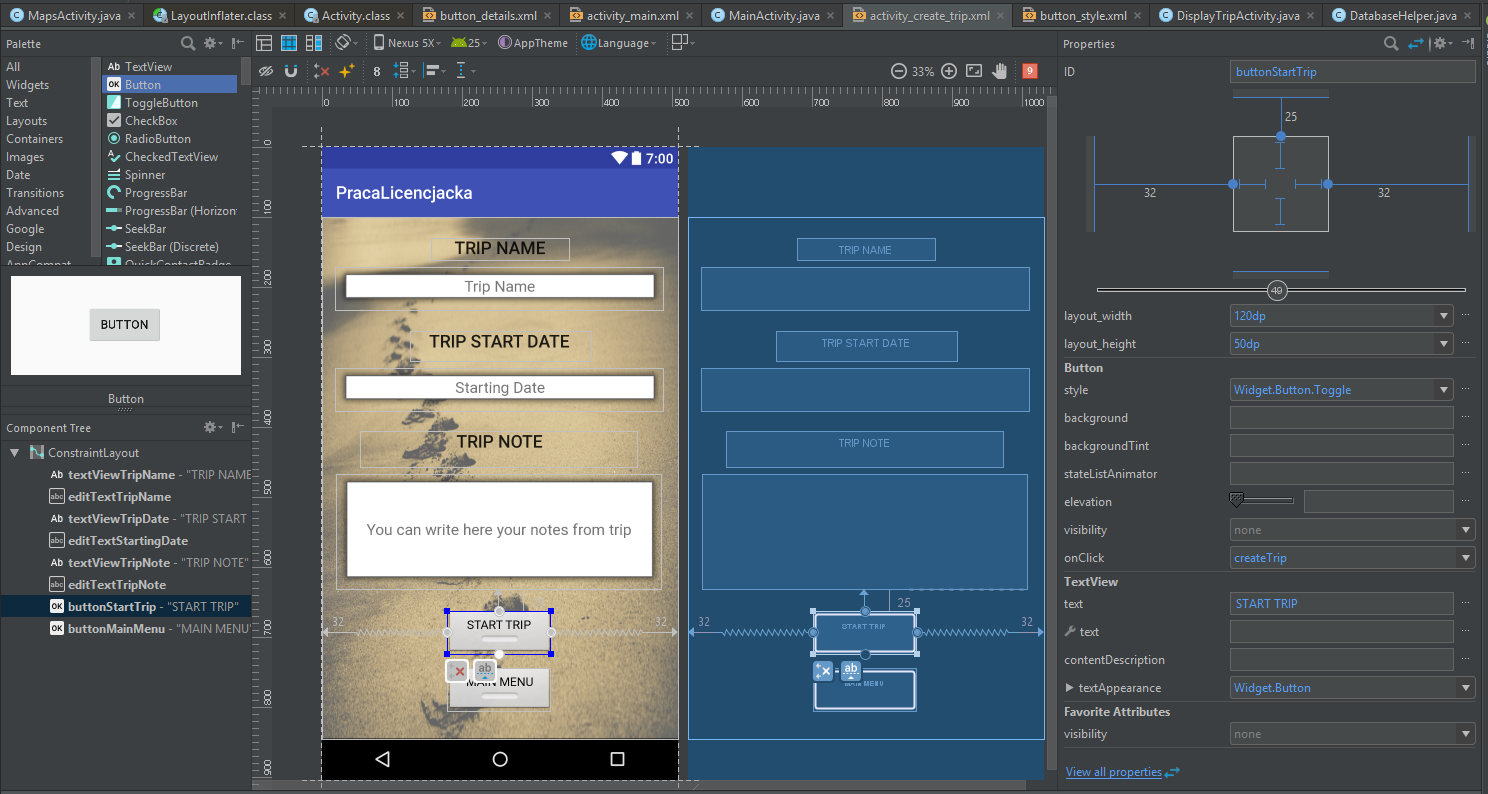
Wybór narzędzia

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 tą 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 i czasami mogą wprowadzać konflikty w tworzonej aplikacji

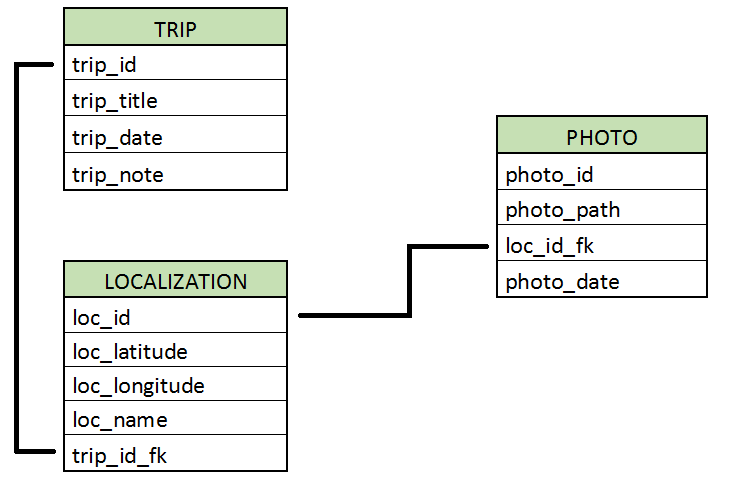
Baza danych

Pojęcie bazy danych

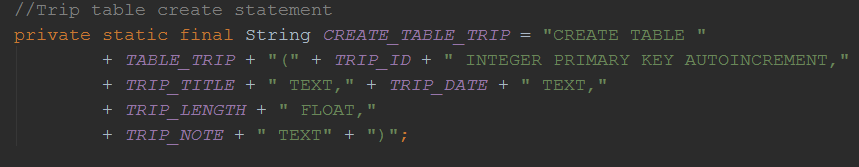
Baza danych to w najprostrzym wytłumaczeniu pewnego rodzaniu kontener w którym możemy przechowywać informacje, najczęściej są one przechowywane w tabelach dzięki czemu mamy możliwość łatwego odwoływania się do nich za pomocą różnych poleceń zwanych kwerendami. Kwerendy są to polecenia służące do wyciągania 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

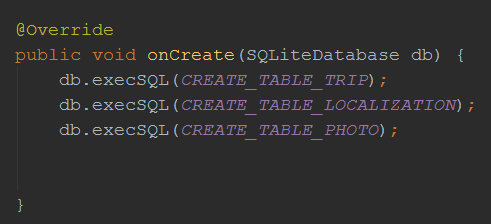
W tworzonej przeze mnie aplikacji również zastosowałem bazę danych ze względu na dużą ilość informacji do których potrzebuję mieć dostęp w trakcie działania programu, dzięki bazie mogę się do nich odwoływać w każdym momencie działania bez obawy, że zostaną one nagle utracone bez możliwości ich ponownego 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 i obsługa bazy danych w aplikacji

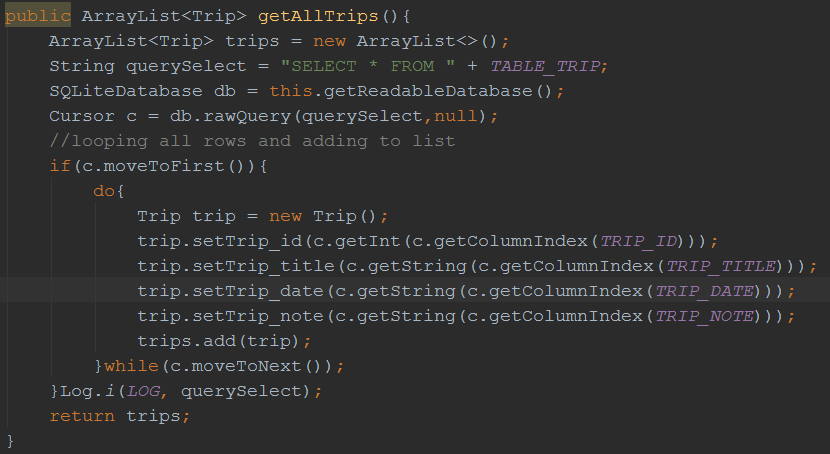
W mojej aplikacji postanowiłem stworzyć bazę danych lokalnie, czyli w pamięci urządzenia ze względu na to, że nie jest ona zbyt duża i przechowuję w niej tylko dane tekstowe więc nie ma problemu z przepełnianiem się pamięci. W celu jej obsługi utworzyłem klasę DatabaseHelper którą rozszerzyłem o klasę 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:

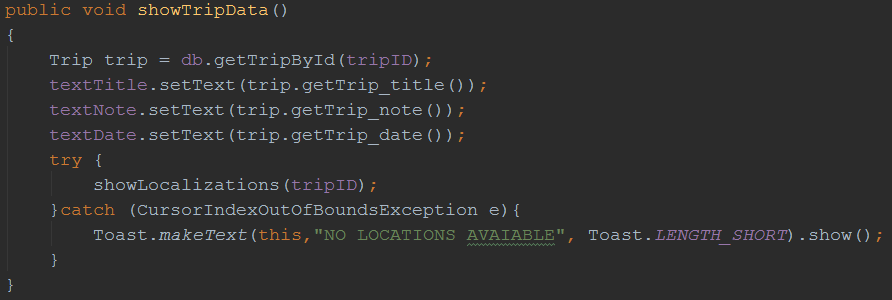


Tworzenie tabel zostało umieszczone we wbudowanej funkcji onCreate(), która jest uruchamiana za każdym razem gdy wywołuję klasę DatabaseHelper. Dzięki temu mam pewność że wywołanie klasy w każdym miejscu mojej aplikacji utworzy mi nową bazę danych lub w przypadku istnienia przywoła mi całą jej strukturę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

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 np do umieszczania ich 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)).