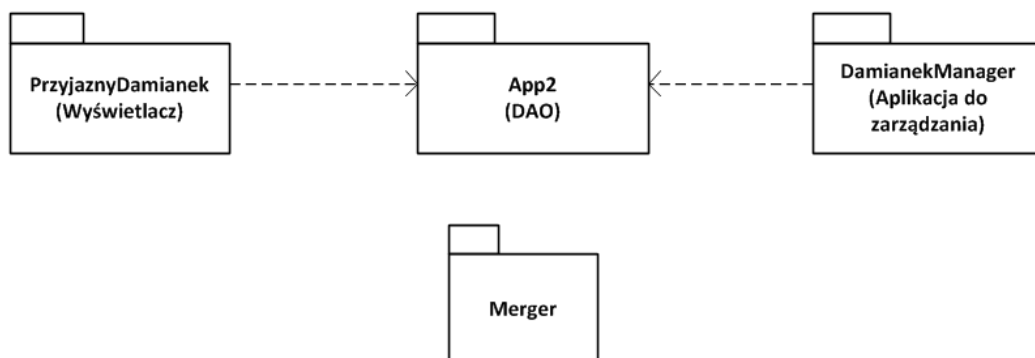


Architektura Systemu

Użyte Technologie

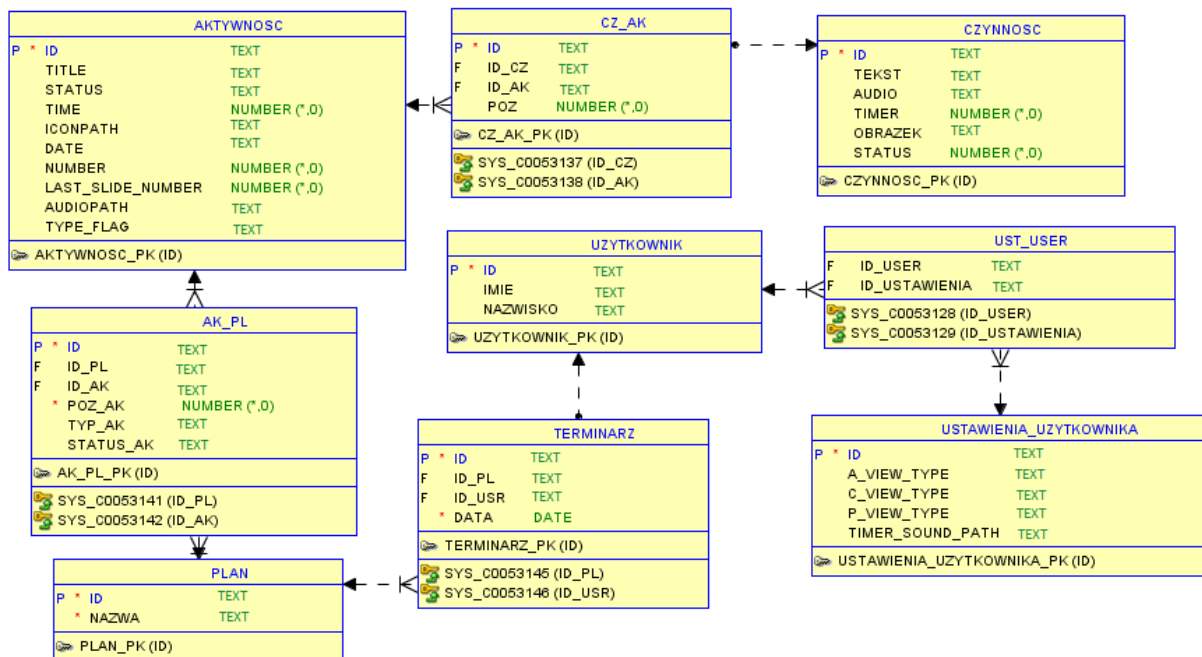
Język programowania: Java zgodny z Android API 7.0
JVM : Dalvik, Android API min 7.0
System operacyjny: Android OS
Baza danych: SQLite wersja 3.0
ORM: użyte rozwiązanie wbudowane w SDK Android
System kontroli wersji: Git
IDE: Android Studio

Architektura systemu



- App2 (Dao) - jest to wydzielony moduł pełniący rolę dostępu do bazy danych. Definiuje interfejsy dostępu do poszczególnych tabel. Ułatwia operację na bazie danych typu CRUD i nie tylko. Korzystają z niego moduły PrzyjaznyDamianek oraz DamianekManager. Również w nim zdefiniowane są klasy tworzące fizyczną bazę danych.
- Moduł PrzyjaznyDamianek zwany również Wyświetlaczem to osobna aplikacja która ma zadanie wyświetlać dane w sposób zdefiniowany przez użytkownika. Definiowanie widoku odbywa się w module DamianekManager.. Moduł zależny jest od modułu App2.
- DamianekManager to osobna aplikacja służąca do zarządzania użytkownikami oraz ich ustawieniami,, zarządzania planami aktywnościami oraz planami aktywności.
- Merger to osobna aplikacja, uruchamiana na komputerze a nie jak dwie pozostałe, na tablecie, która ma zadanie ułatwiać przenoszenie danych z bazy lokalnej do bazy centralnej oraz na odwrót. Ma zadanie ułatwiać dzielenie się utworzonymi planami pomiędzy użytkownikami i ułatwić pracę opiekunom, aby nie musieli tworzyć tych samych planów na wielu różnych tabletach.

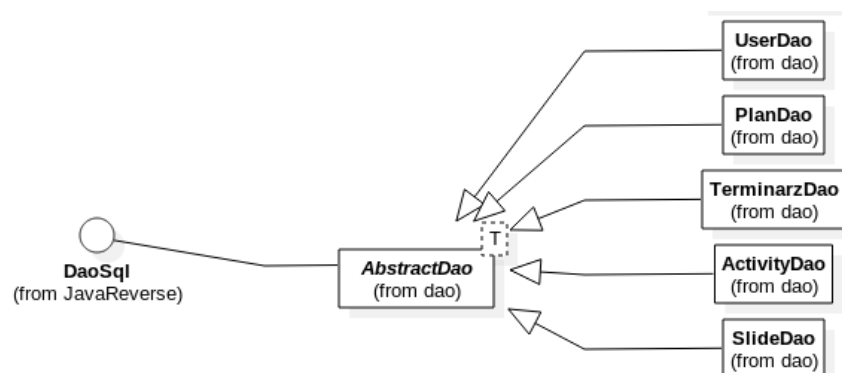
Diagram ERD Bazy danych



Wszystkie klucze główne są GUIDami (Globally Unique Identifier). Taka konstrukcja pozwala na przenoszenie danych z bazy lokalnej do bazy centralnej.

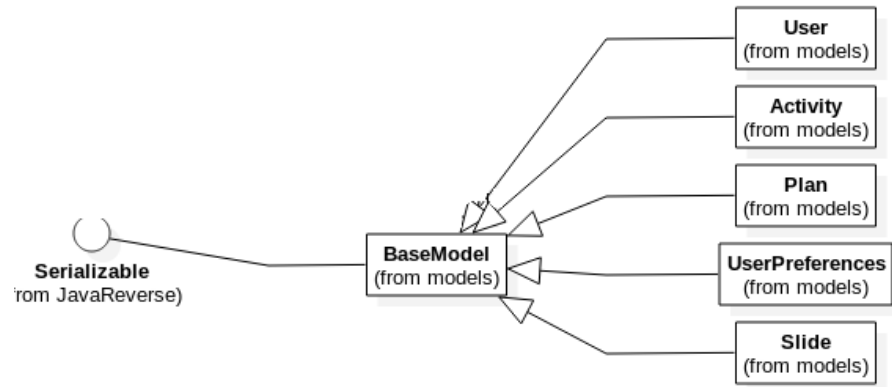
Baza lokalna jak i centralna posiada taki sam schemat jak powyżej. Baza lokalna posiada jedynie dodatkową tabelę wskazującą, który użytkownik jest w danej chwili wybrany. Tabela z jednym rekordem jest koniecznością aby elastycznie dostosowywać wyświetlanie danych z ustawieniami zdefiniowanymi per użytkownik.

Klasy modułu bazy danych



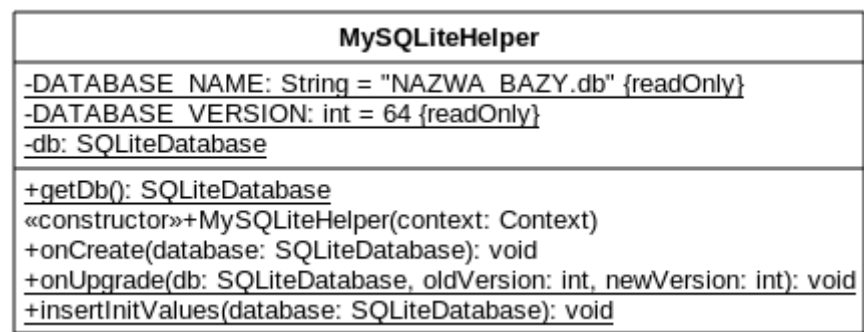
Hierarchia klas dostępu do warstwy danych

Konkretne implementacje AbstractDao pozwalają uzyskać dostęp do danych tabel. Dostarczają szereg metod pozwalających na elastyczne pobieranie, modyfikowanie oraz usuwanie danych



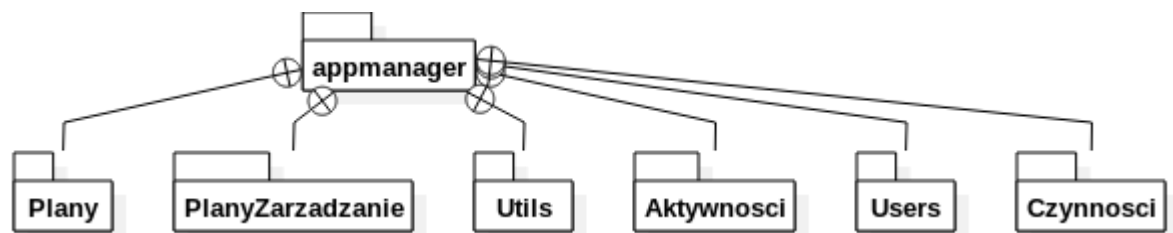
Hierarchia klas modeli

Są to klasy, które odwzorowują encję w bazie danych.



Najważniejsza klasa modułu. Jest to singleton, pozwala dostać uchwyt bazy danych poprzez operację getDB();

Diagram pakietów Aplikacji do zarządzania



Każdy pakiet realizuje wydzielone funkcjonalności:

Plany - odpowiada za tworzenie, edycje, usuwanie planów

PlanyZarzadzanie - odpowiada za edytowanie, planu aktywności na dany dzień

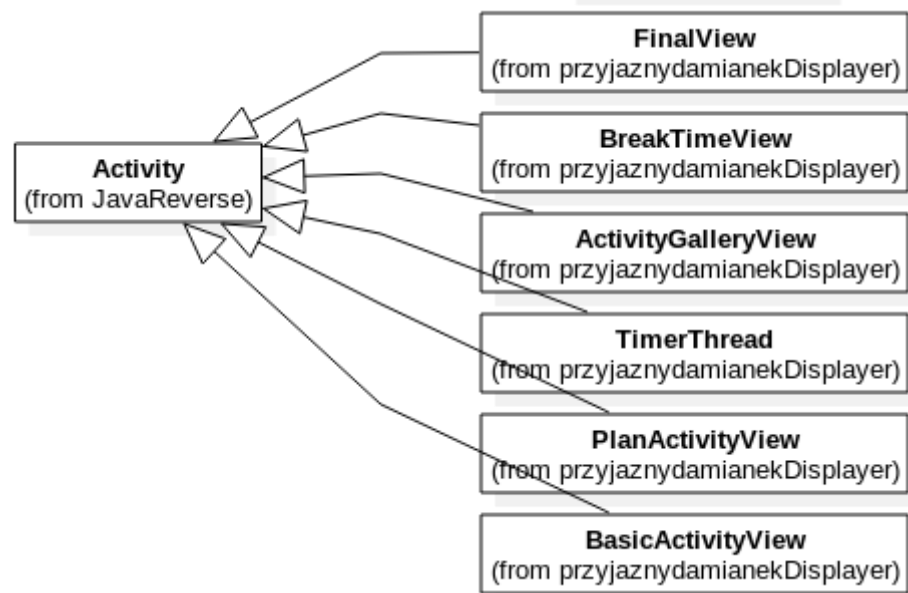
Utils - klasy pomocnicze (FileChooser itp)

Aktywnosci - odpowiada za tworzenie, edycje, usuwanie aktywności

Users - odpowiada za zarządzanie użytkownikami (dodawanie, edytowanie, zmiana ustawień)

Czynnosci - odpowiada za tworzenie, edycje, usuwanie czynności które wchodzą w skład aktywności

Najważniejsze klasy Wyświetlacza



Powyższy diagram obrazuje implementacje widoków w aplikacji.

BreakTimeView - widok przerwy

ActivityGalleryView - widok galerii aktywności

BasicActivityView - widok aktywności (w zależności od ustawień przyjmie widok slajdu albo listy)

PlanActivityView - widok planu aktywności

FinalView - Widok ekranu końcowego

TimerThread - zdezaktualizowana klasa timera

8. Aplikacja *Merger* - aplikacja składa się z jednej dosyć rozbudowanej klasy, w której wykonywane są głównie operacje na bazie danych oraz plikach, do tego w skład wchodzi kilka mniejszych klas - np. kontenery napisane dla aktywności czy metody odpowiadające za obsługę plików