Find Your Venue

Matteo De Francesco

May 30, 2020

1 Introduction

Find Your Venue is a simple application for Android based on localization, user photos and local database. The main idea is that the user can select an area from integrated Google Maps and will be able to explore all the venues in that area. The user can save his favourite venues to local DB and can add photos. He can also see information about that place and share it

2 Working Flow

At the *onCreate* in MainActivity, the application will request permission of localization and will ask the user to activate GPS. The data about position is stored only on the *SharedPreferences*, so the user will not have to worry about privacy issue. The flow of the application is explained in the following simple schema:

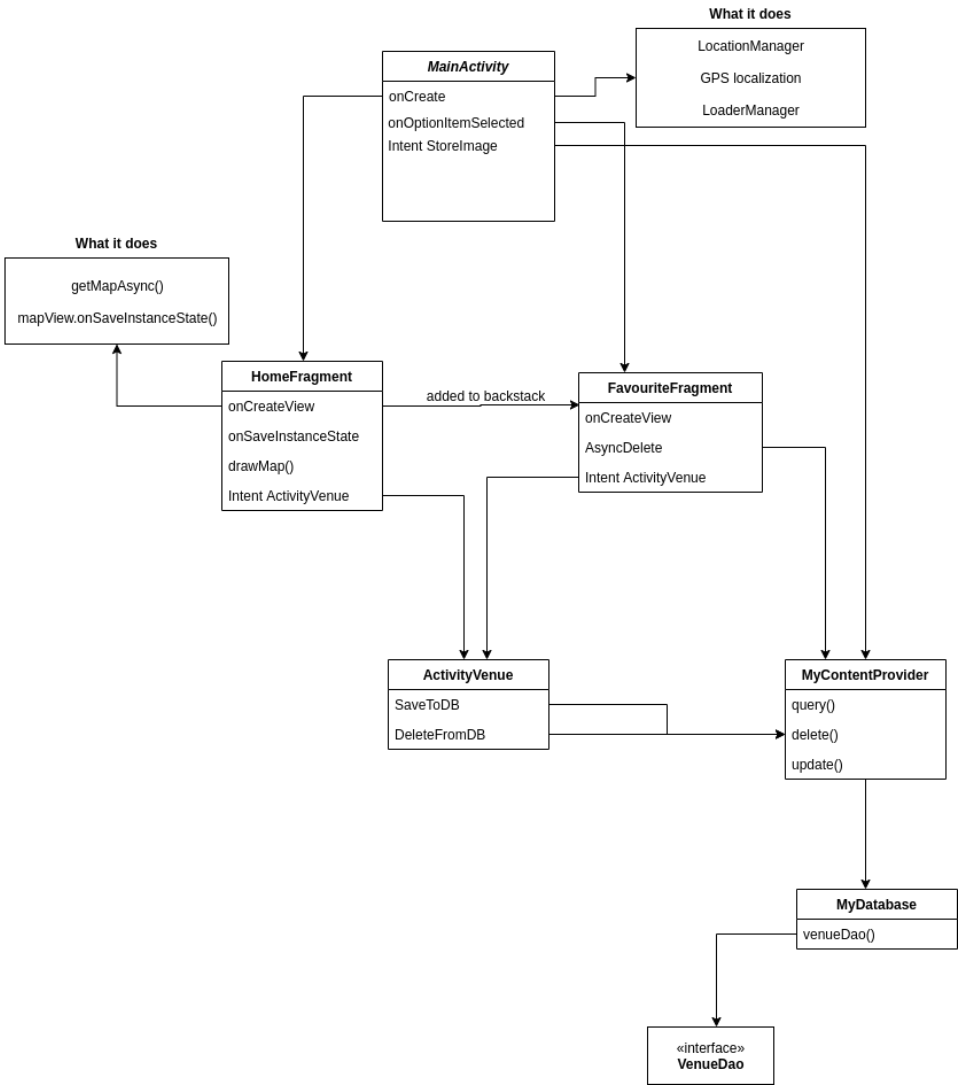

Figure 1: A simple schema

The **HomeFragment** is created at start and the map is loaded asynchronously. When the user select an area to explore, the *drawMap()* is called to draw the polygon. When the user selects the heart icon in the AppBar, the **FavouriteFragment** comes in and the HomeFragment is added to backstack, so the state of the fragment remain saved in memory. When the user select a venue from HomeFragment or FavouriteFragment, a new activity

ActivityVenue starts, and here he can save place to local DB or remove if he's in Favourites. With the *FloatingActionButton* of the MainActivity or the gallery icon in the AppBar, the user can take/select a photo and can add it to a saved venue with *Intent StoreImage*. This operations are done all asynchronously with *contentResolver*, that will use the URI specified in **MyContentProvider**, which contains all the logic for the operation of **MyDatabase** and interface **VenueDao**.

3 Other details

The RecyclerView of the HomeFragment loads data asynchronously, waiting for the call to Foursquare API. In MainActivity onCreate, a *LoaderManager* loads data from local Database, and keep a copy of the Cursor in the companion object, so in this way when FavouriteFragment comes in it doesn't need to load data again but it can use the shared cursor. However it has its own LoaderManager to reload data when the user modifyes the database inside this fragment.
To load photos asynchronously I have used **Picasso** library, it has automatic memory and disk caching and also it automatically handles recycling and download cancellation inside adapter.
In ActivityVenue I have used a ViewPager to display multiple photos, and also there is the option to listen to place description using **TextToSpeech** library of Android.
The missing permissions are requested at runtime using dialog pop-ups.
In the FavouriteFragment there is also a search icon with searchview, so he can filter the places by name, always using query of **MyContentProvider**.

4 Screenshots

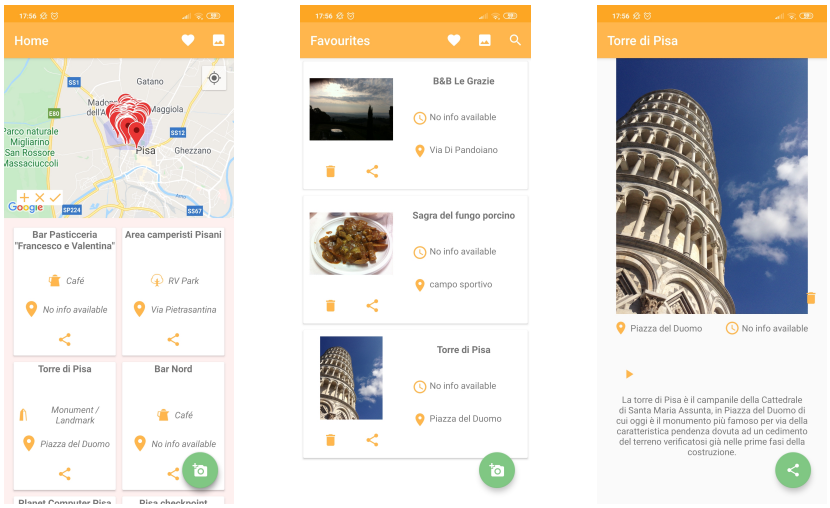

Figure 2: HomeFragment,FavouriteFragment and ActivityVenue

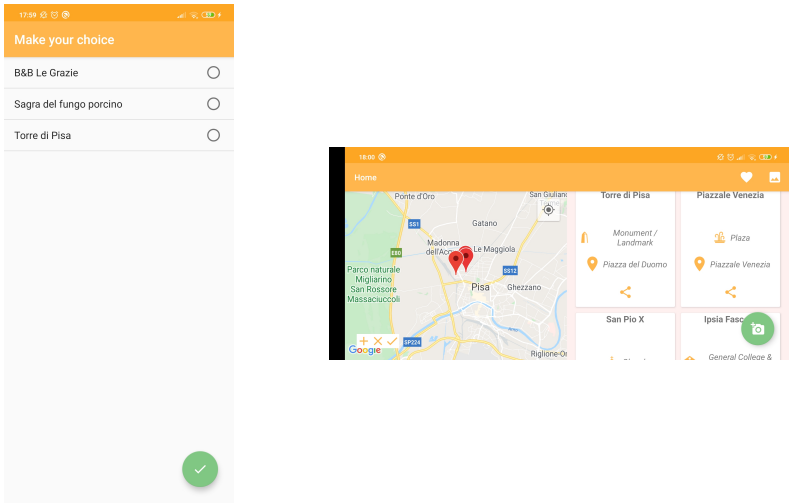

Figure 3: StoreImage and Main landed