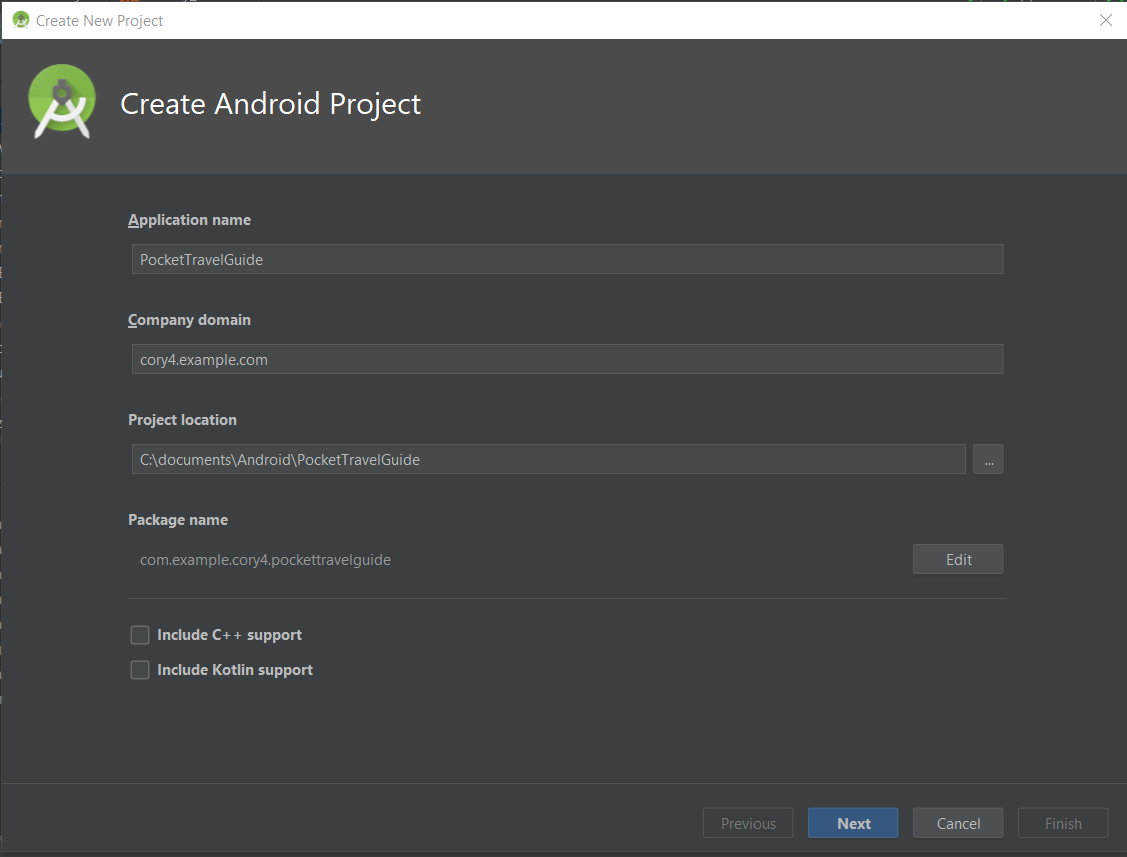
**Licenta 2019**

# Structura

1. Login: Firebase
2. Meniu:
   1. ~~Alegere limba (intre romana si engleza)~~
   2. Harta
   3. Despre (informatii despre oras generale + imagini 3D)
   4. Help (cum se foloseste aplicatia + chat unde se pot intreba diverse)
   5. Optiuni (sunet, notificari etc.)
   6. Recomandari (lista cu locatii recomandate pe categorii)
3. Harta:
   1. Urmaresti pe harta (cu ghidare audio)
   2. Marcaje acolo unde sunt obiectivele
   3. Posibilitatea alegerii traseului in functie de timpul care il ai
   4. In momentul in care treci pe langa o cafenea/restaurant/hotel/centru comercial partenere, vei fi atentionat (sub harta) de existenta lor in apropriere
   5. In momentul in care ajungi la un obiectiv poti scana codul QR pt a afla informatii despre acesta ( acestea vor putea fi si ascultate)

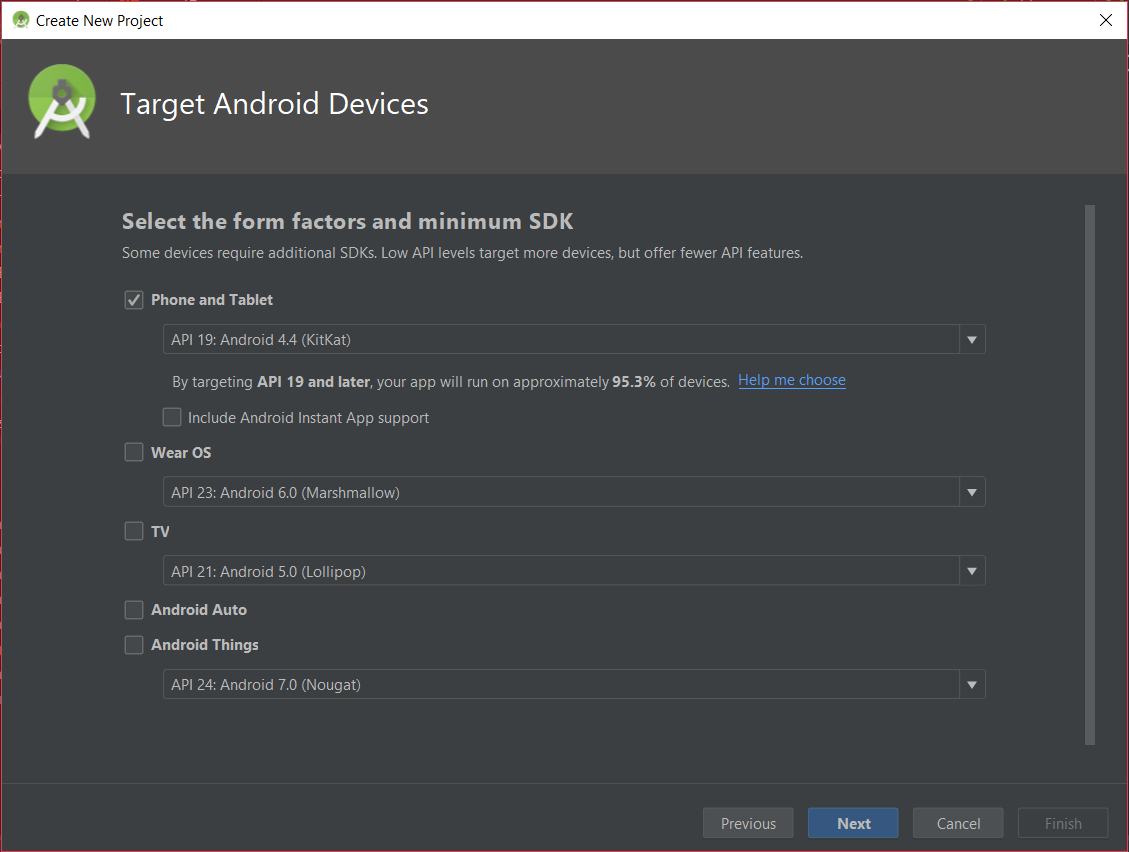
Creeare aplicatie

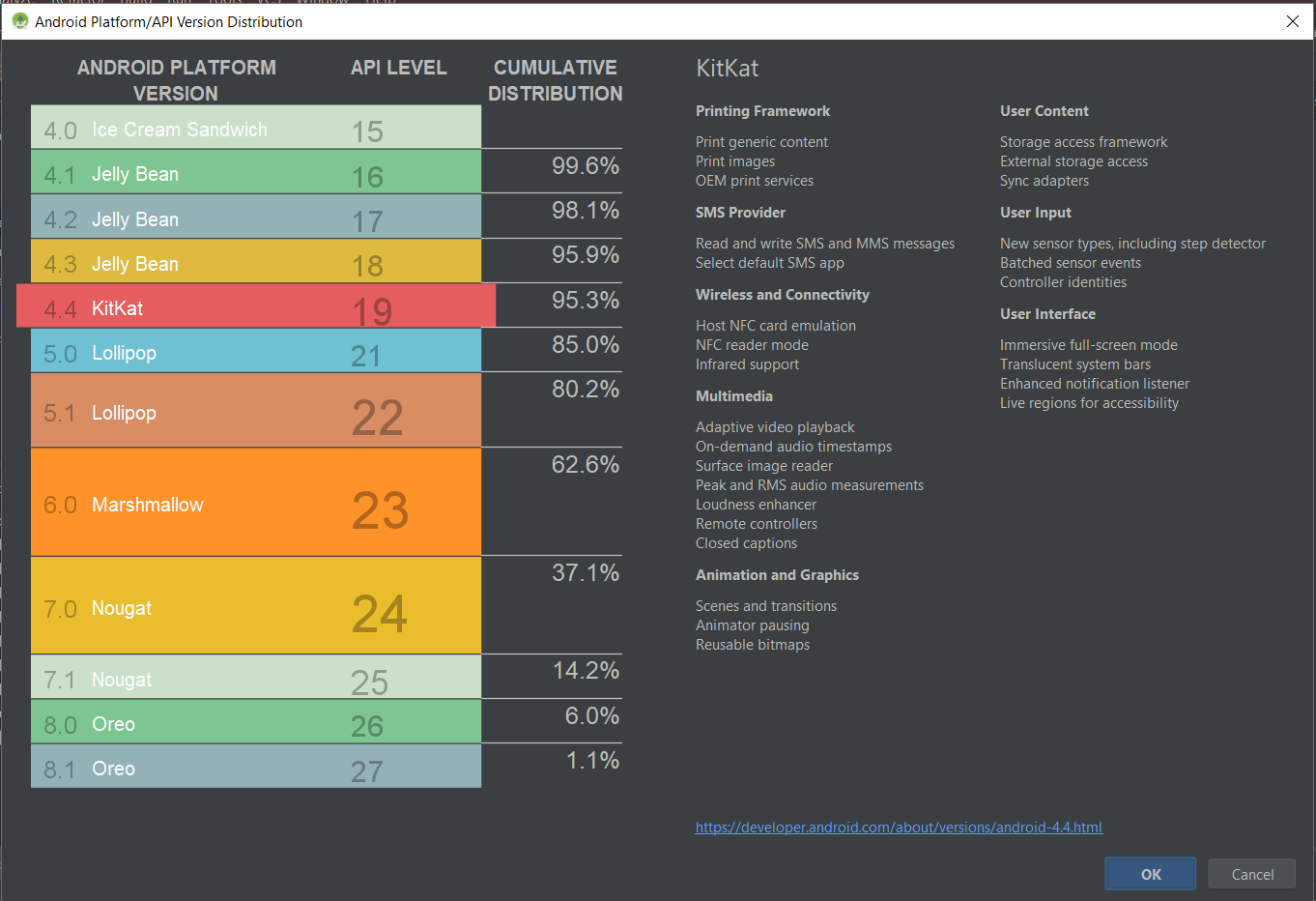
Numele aplicatiei este: Pocket Travel Guide



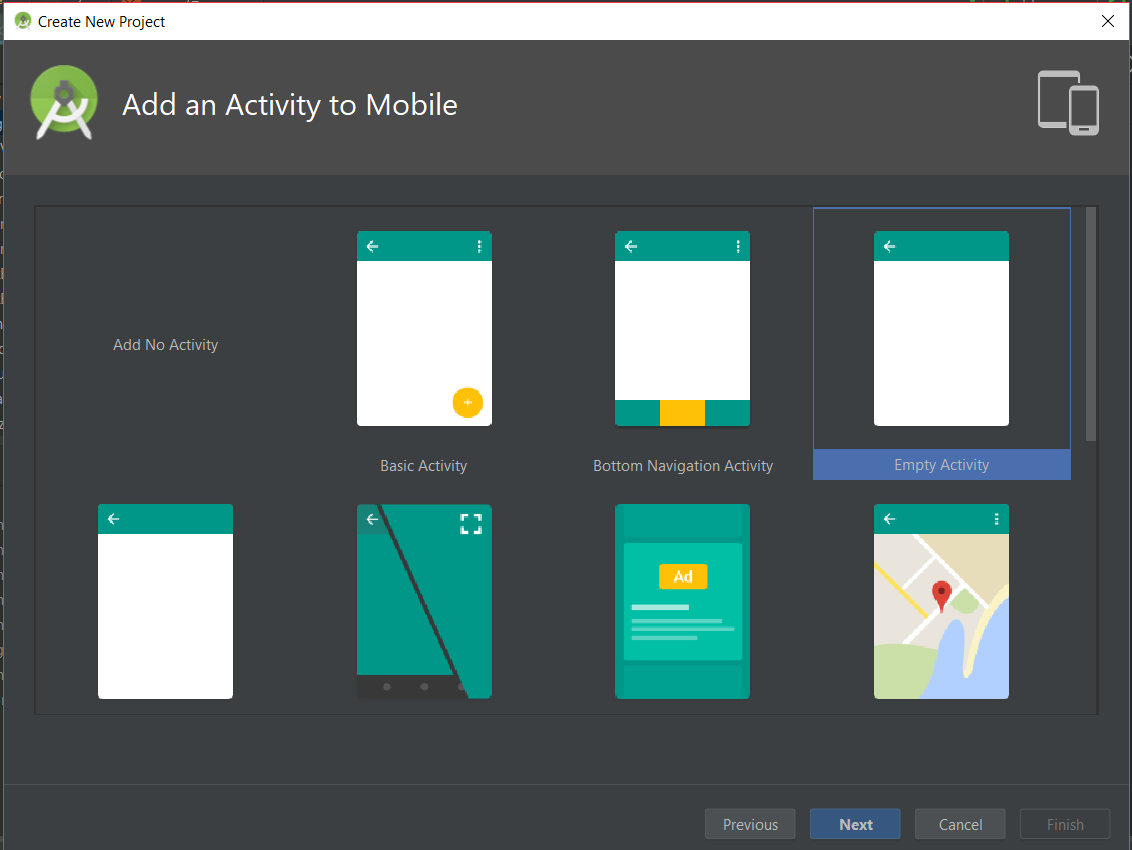
Versiunea android: In timp ce cu 4.1 (Jelly Bean) am atinge o acoperire de 99.6% din dispozitivele Android, hartiile necesita un minim de 4.4 (Kit Kat) pt a functiona, prin urmare vom avea o acoperire de 95.3%. (<https://support.google.com/maps/answer/3096703?co=GENIE.Platform%3DAndroid&hl=en>)

Pt. VR: 4.1 si 2 GB RAM (<https://www.quora.com/What-are-the-minimum-specifications-for-an-Android-phone-to-run-Cardboard-VR>)

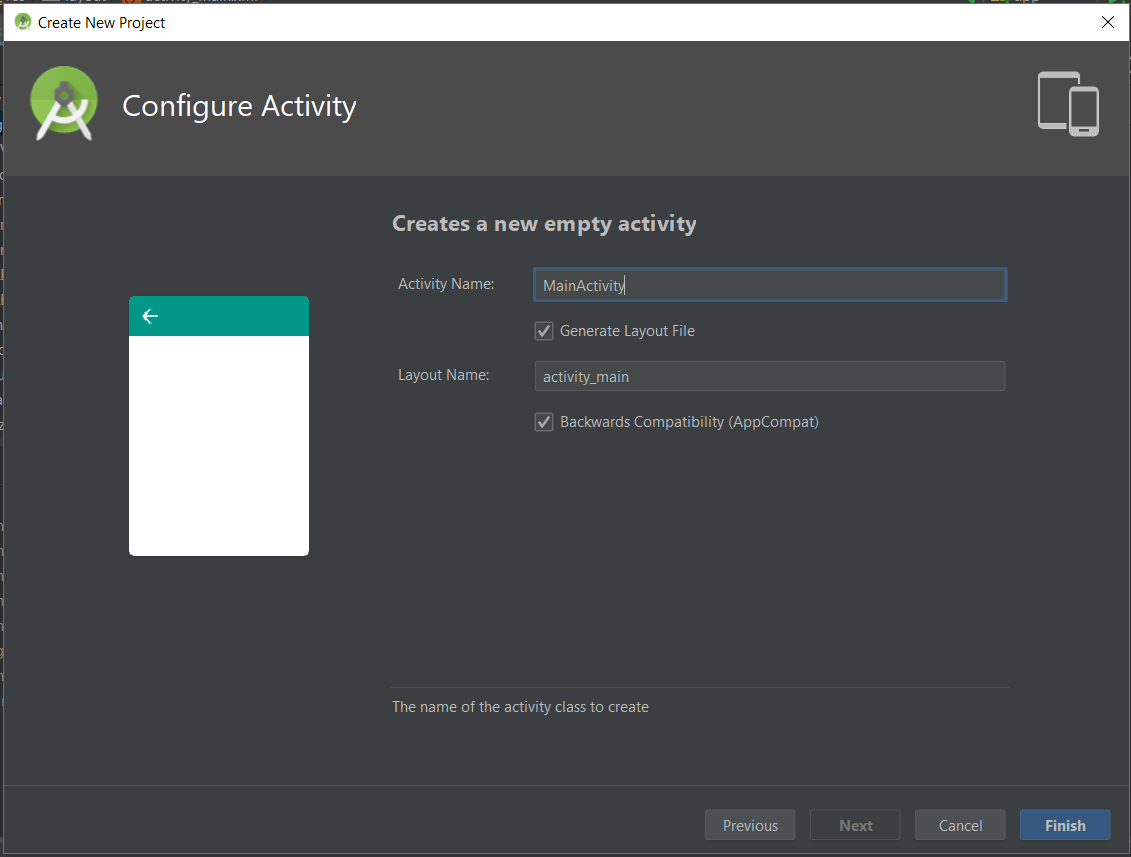




Prima activitate a facuto de tipul Empty Activity



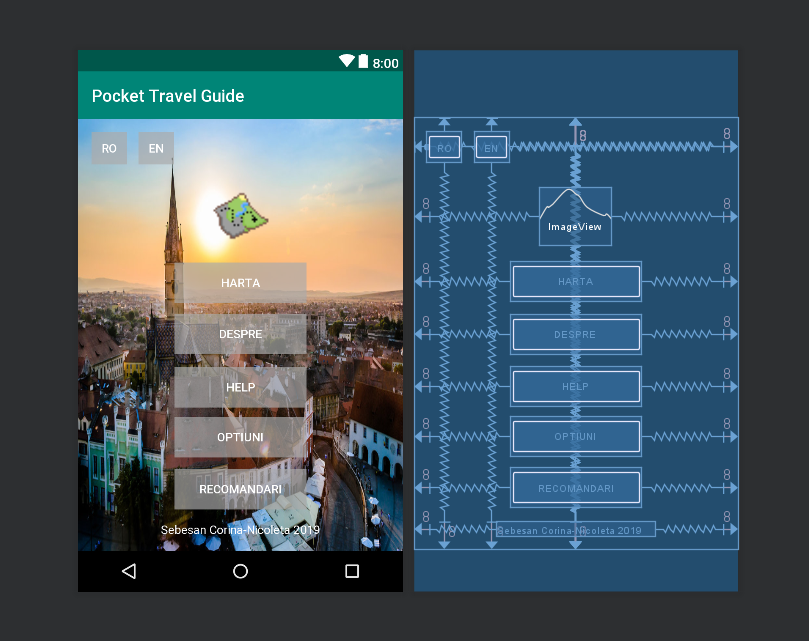
Aceasta va fi activitatea principala

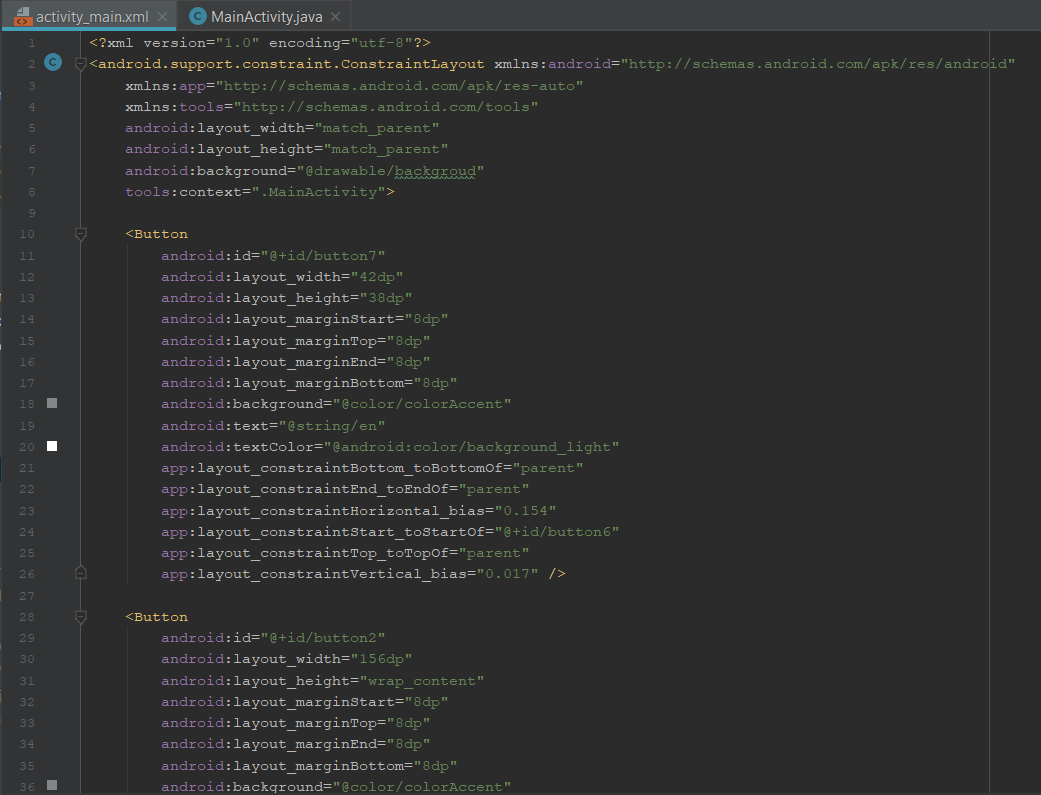


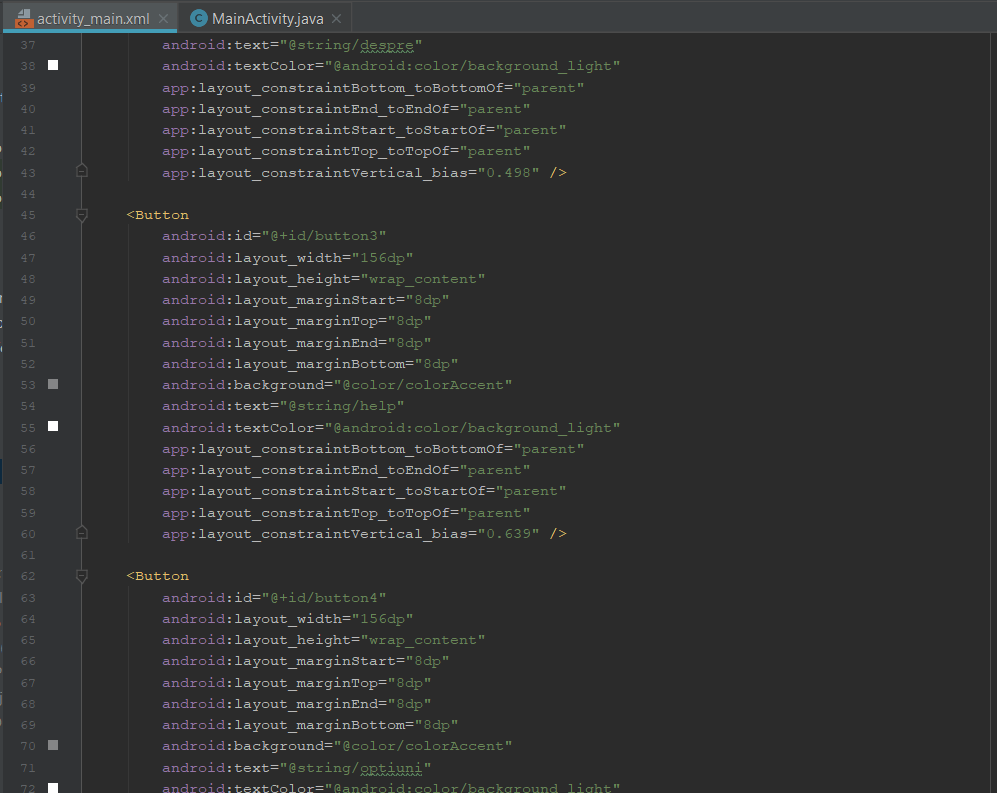
# Main activity

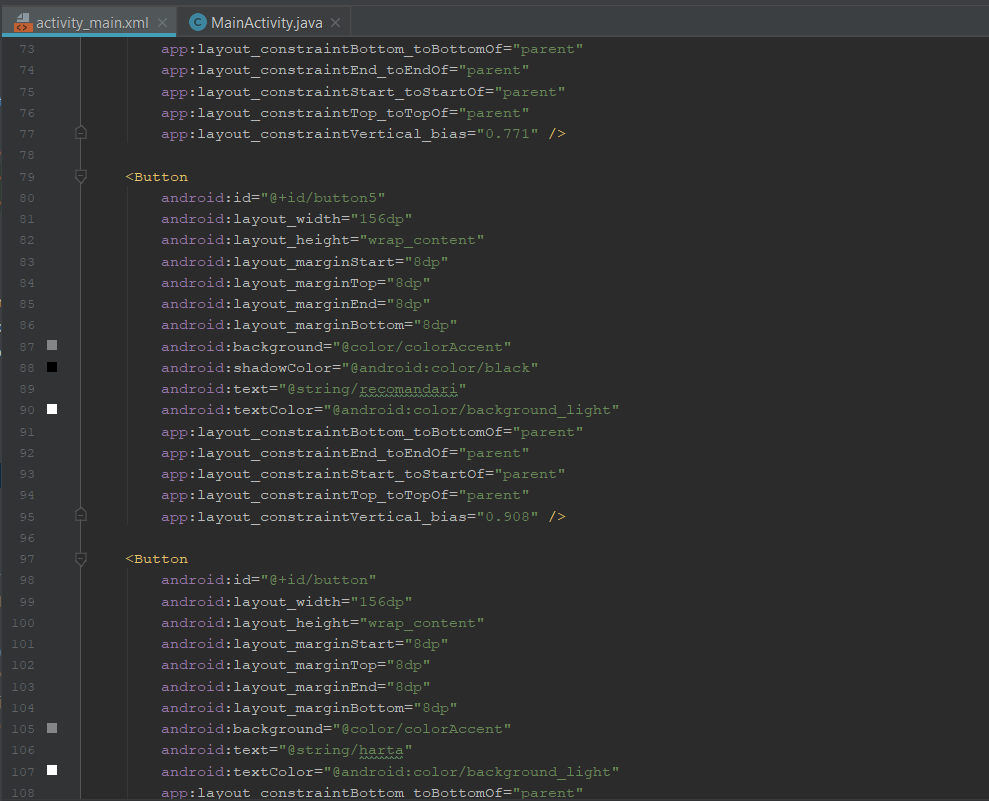
In meniu vom avea butoane pentru schimbat limba, harta, despre, help, optiuni, recomandari.

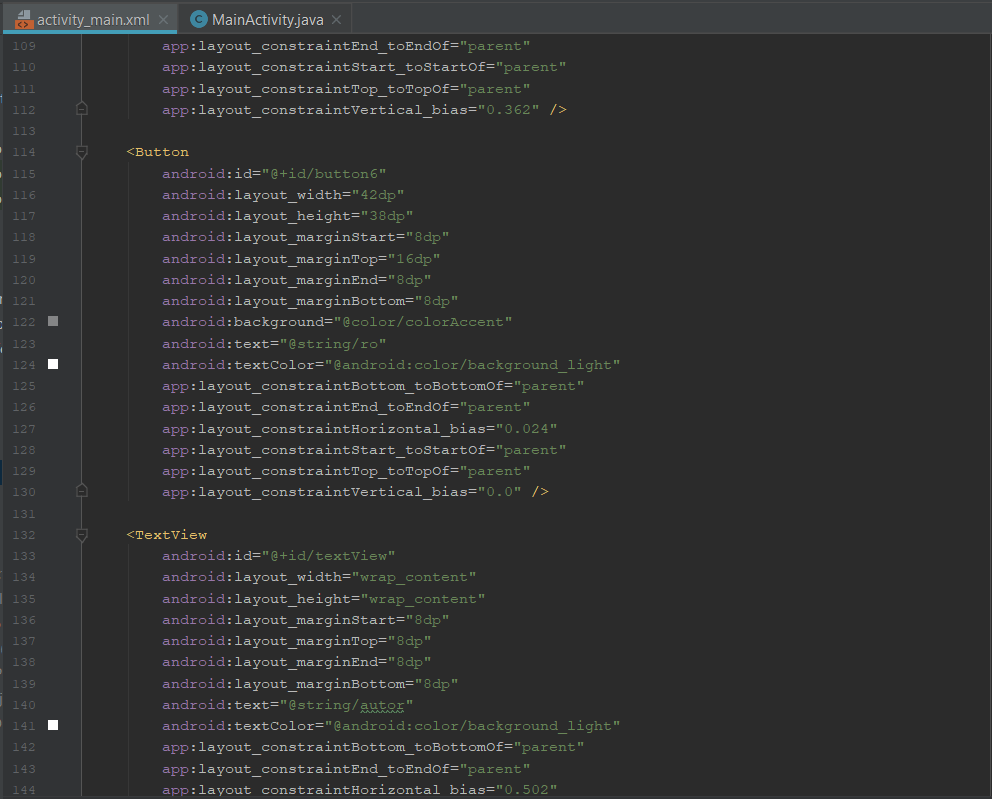
Logo-ul a fost realizat de mine cu site-ul: https://www.pixilart.com/draw

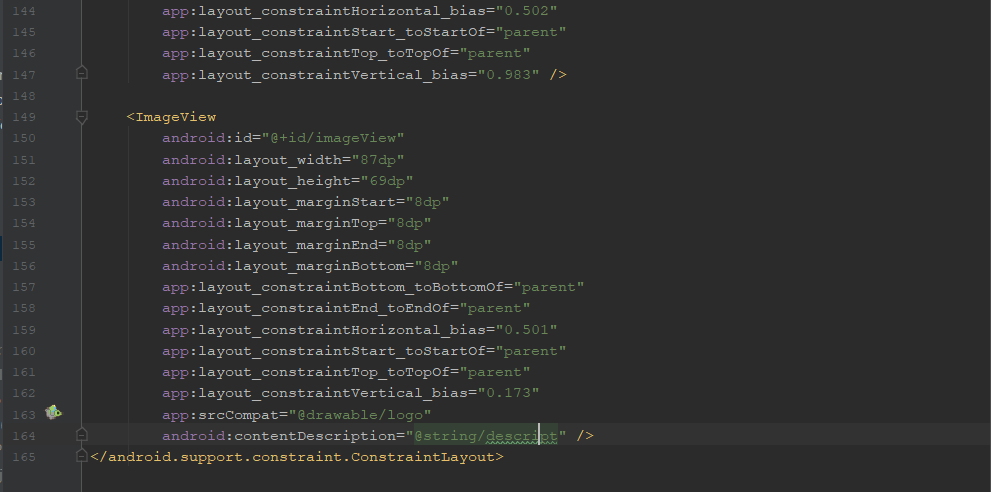












## Alegere limba (intre romana si engleza)

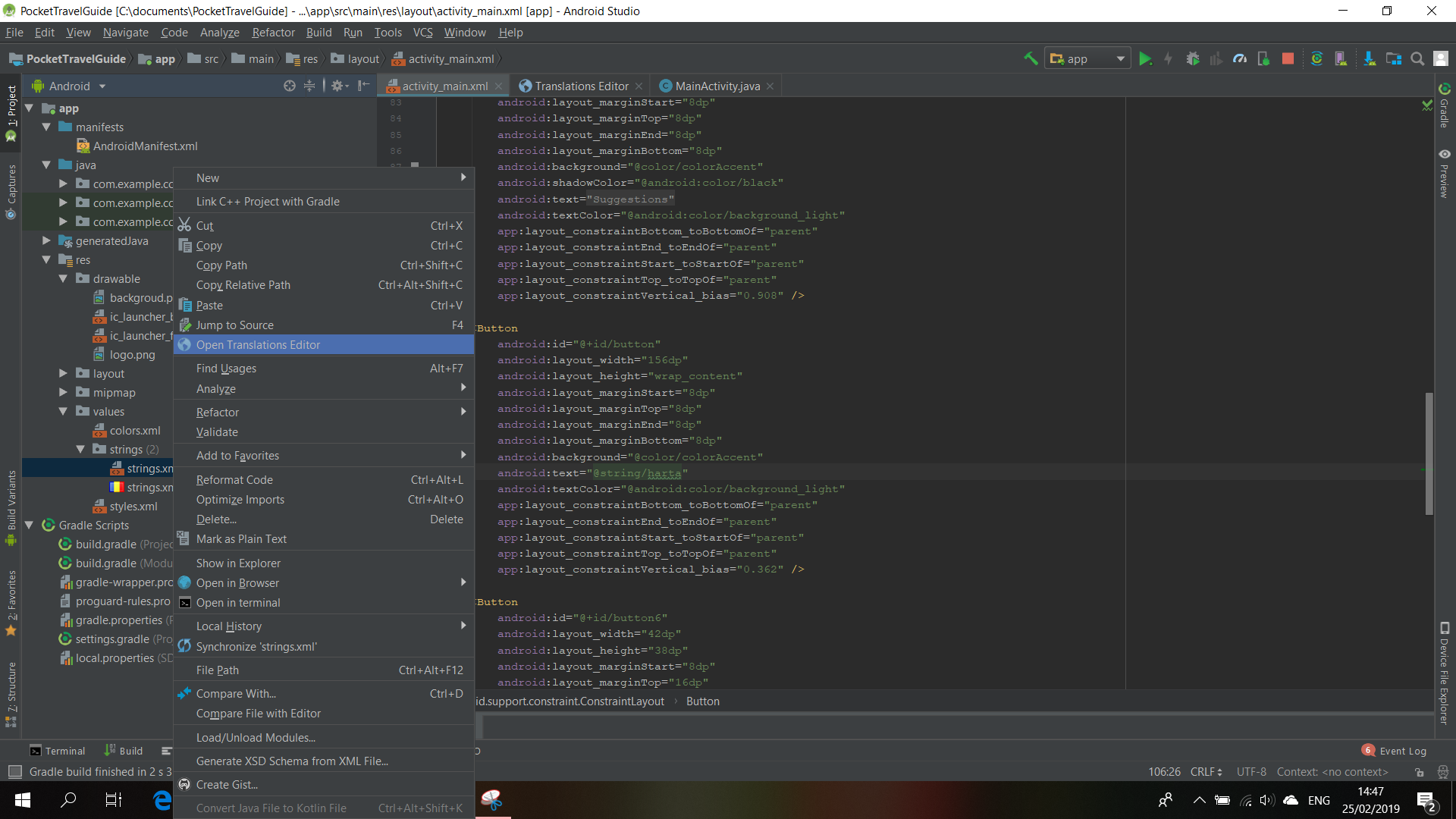
<https://developer.android.com/studio/write/translations-editor>

<https://www.youtube.com/watch?v=zILw5eV9QBQ>

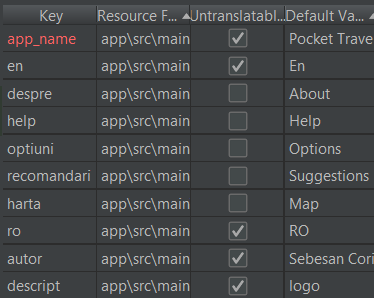
* 1. Se salveaza toate textele din layout in fisierul strings.xml (inlocuind text-ul cu @string/nume\_variabila)



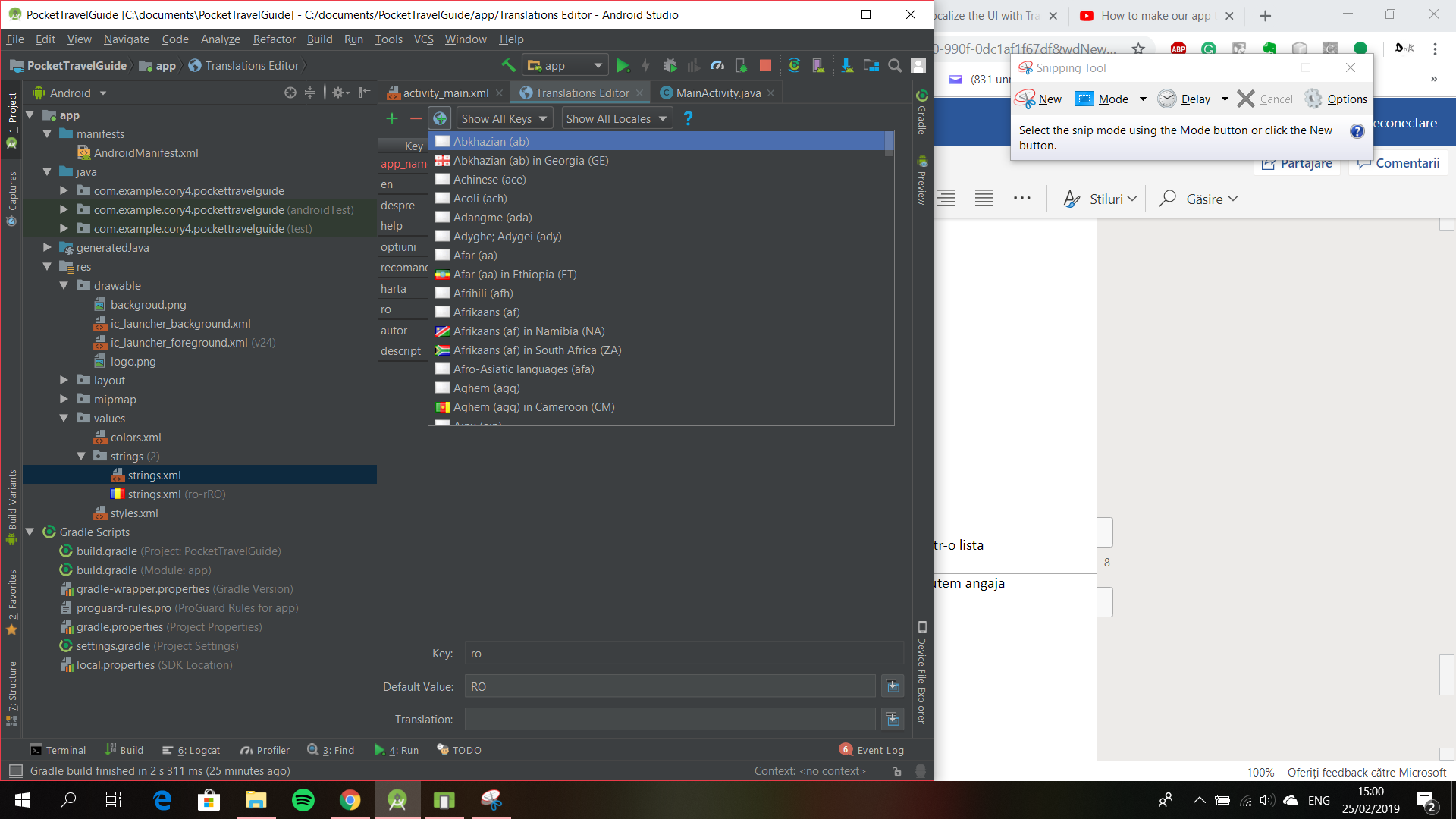
* 1. Se face click dreapta pe fisierul strings.xml si se Open Translations Editor



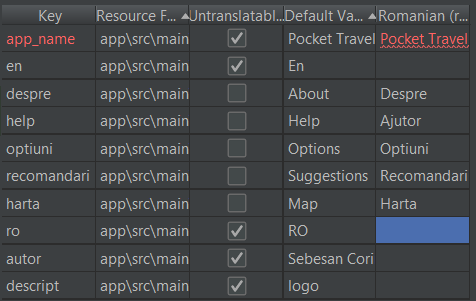
* 1. Aici va fi o tabel cu toate string-urile si traducerile lor (cea default e en-us). Exista si o coloana Untranslatable unde poti spune daca unele cuvinte nu trebuie traduse ( au aceeasi semnificatie universal)



* 1. Pt a adauga o noua limba se face click pe  si se selecteaza dintr-o lista limba dorita (in cazul nostru am selectata limba romana)



* 1. In noua coloana putem scrie traducerea cuvintelor personal sau putem angaja pe cineva care sa le traduca pentru noi



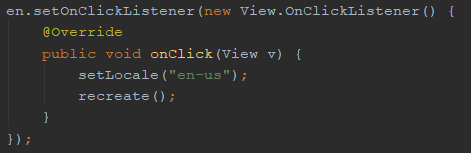
* 1. Creare variabila buton (<https://www.quora.com/What-is-the-work-of-findViewById-in-Android-studio>)



* 1. Functie la click ( <https://developer.android.com/reference/android/view/View.OnClickListener.html#onClick(android.view.View>)

Creeam o functie care sa seteze limba

Refacem activitatea in functie de limba aleasa (<https://stuff.mit.edu/afs/sipb/project/android/docs/training/basics/activity-lifecycle/recreating.html>)

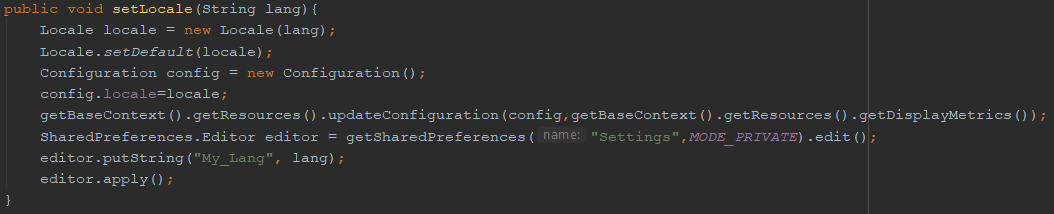


* 1. SetLocale: se creeaza un nou obiect de tip Locale care are ca limba cea selectata .

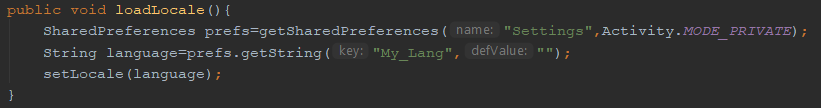
Se seteaza defaultul la instanta nu la system (<https://www.tutorialspoint.com/java/util/locale_setdefault.htm>)

Se creeaza o variabila de tip configuration si se seteaza locale ca fiing limba aleasa

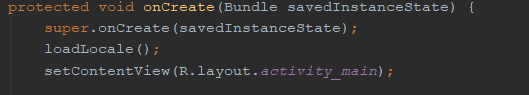
Salveaza datele la setarile comune



* 1. LoadLocale: incarca datele salvate in proprietatiile comune



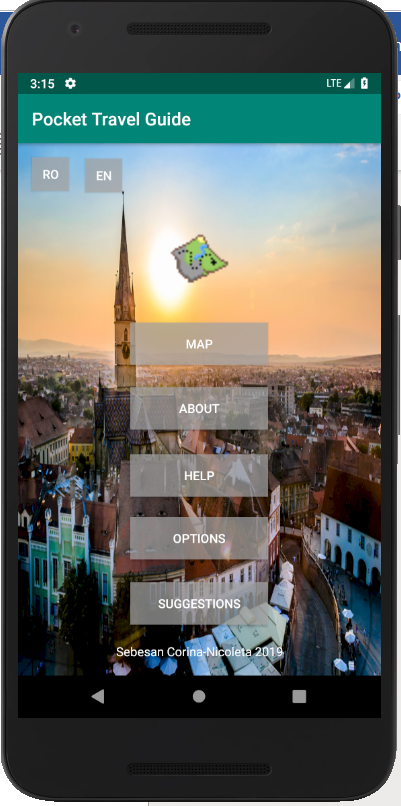
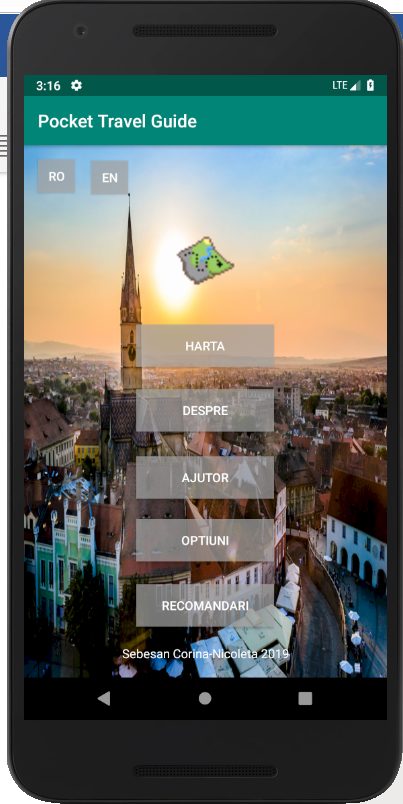
* 1. Apelare loadLocale in onCreate



* 1. Se schimba titlul action bar-ului (daca nu se schimba va fi in functie de limba default)



* 1. Testare

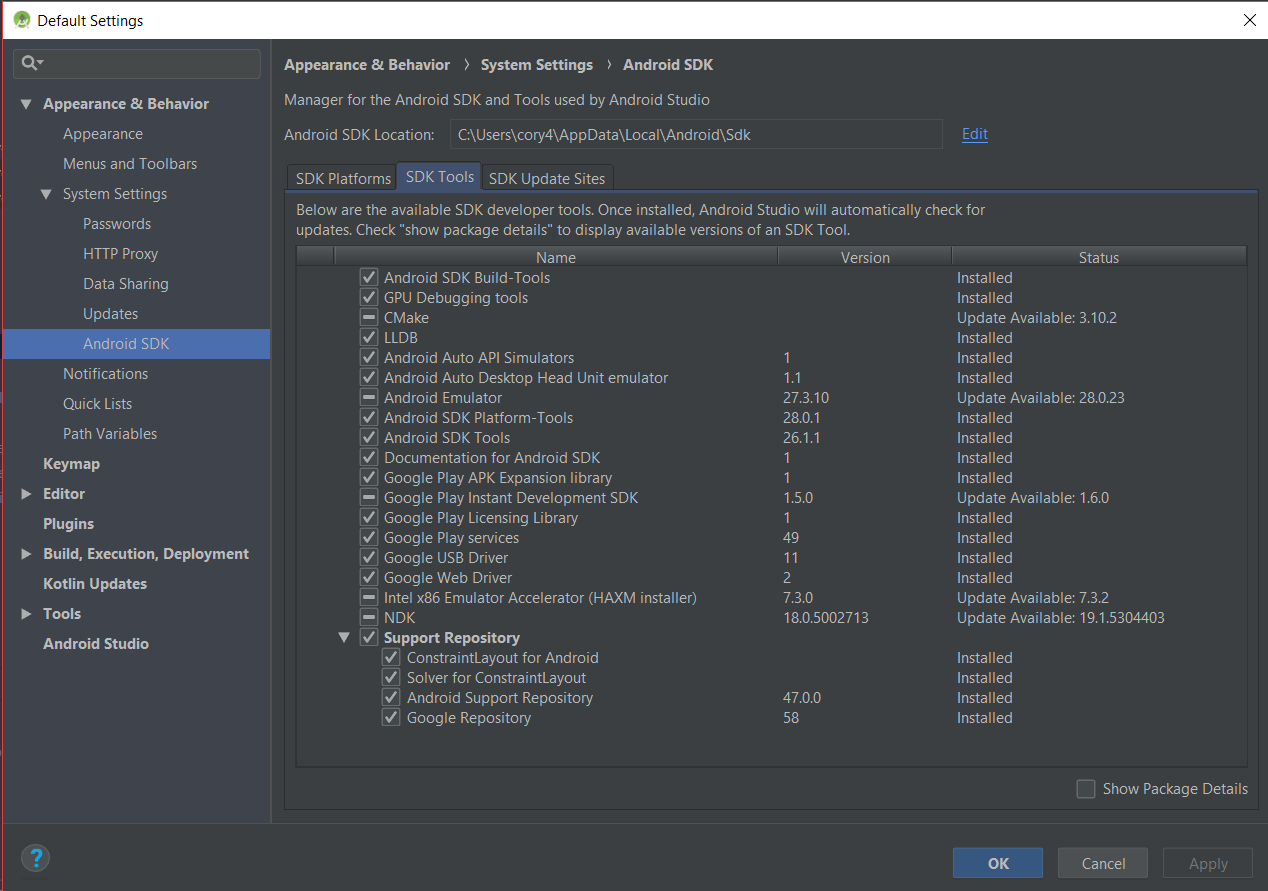
 

# Harta

## Urmarire pe harta cu ghidare audio

<https://www.youtube.com/watch?v=lchyOhPREh4>

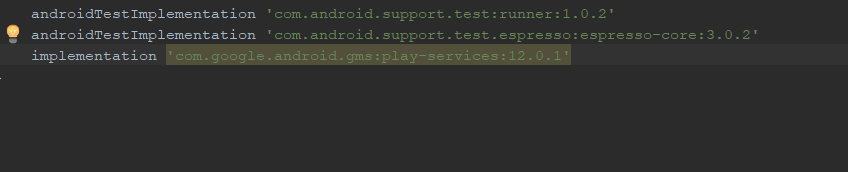
1. Adaugarea pachetului Google Play Services: in sdk manager -sdk tools



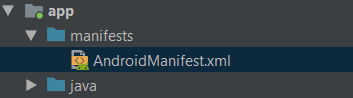
<https://developer.android.com/studio/intro/update.html#sdk-manager>

1. Mergem in build.gradel(Module:app) si spunem ca vrem ca google services sa fie compilate cu proiectul meu:

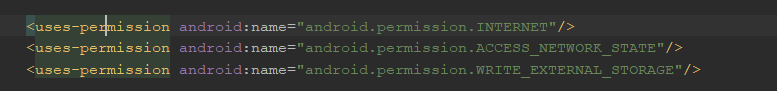




1. Trebuie adaugate niste permisiuni in fisierul manifest:



Avem nevoie de 3 permisiuni: INTERNET (fara internet nu poti accesa google maps), ACCESS\_NETWORK\_STATE ( ca sa vedem daca aplicatia poate accesa google services) si WRITE\_EXTERNAL\_STORAGE ( descarca harti si trebuie sa salveze datele in storage)



Da accesa aplicatiei mele la google api si cu nivelul de protectie (signature nu lasam pe nimeni sa aiba permisiunea asta in afara de aplicatie, este bazat pe semnatura cheii pe care o creez)



Urmatoarea este permisia care am creato ( com.example.cory4.pockettravelguide.permission.MAPS\_RECEIVE)



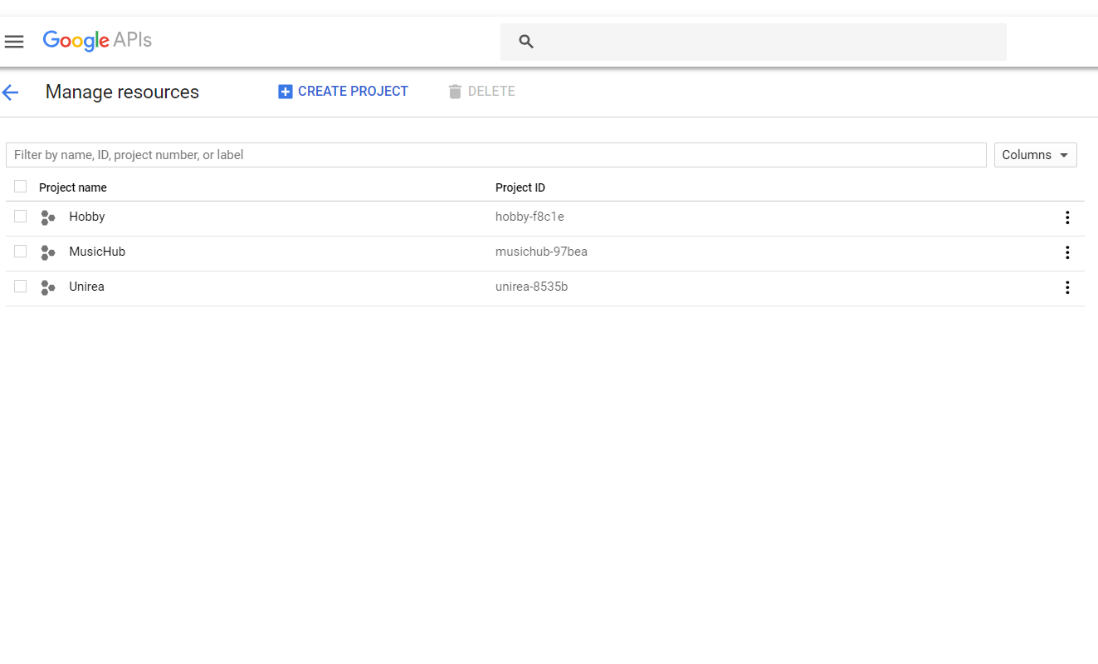
Dupa aveam cea de posibilitate de citire a serviciilor google



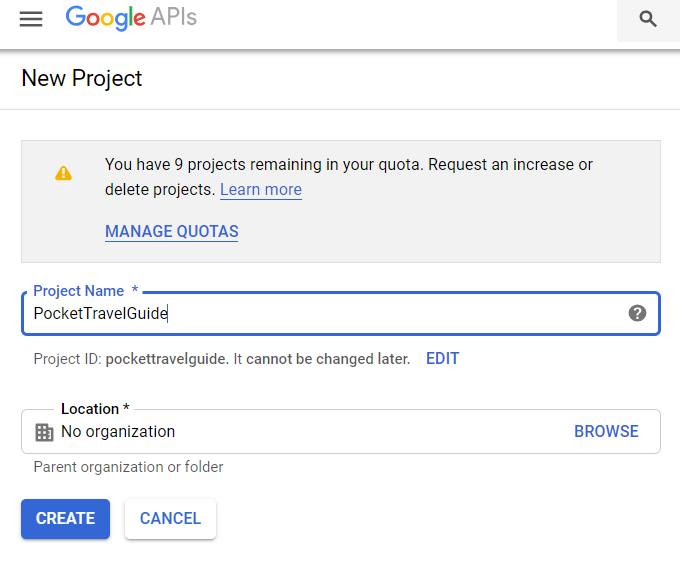
Pt a folosi Google Maps va trebui sa avem openGl pe dispozitiv (versiunea este scrisa in hexa)



1. Trebuie obtinut un api key de la google: trebuie sa mergem pe site-ul <https://console.developers.google.com/cloud-resource-manager?previousPage=%2Fprojects>



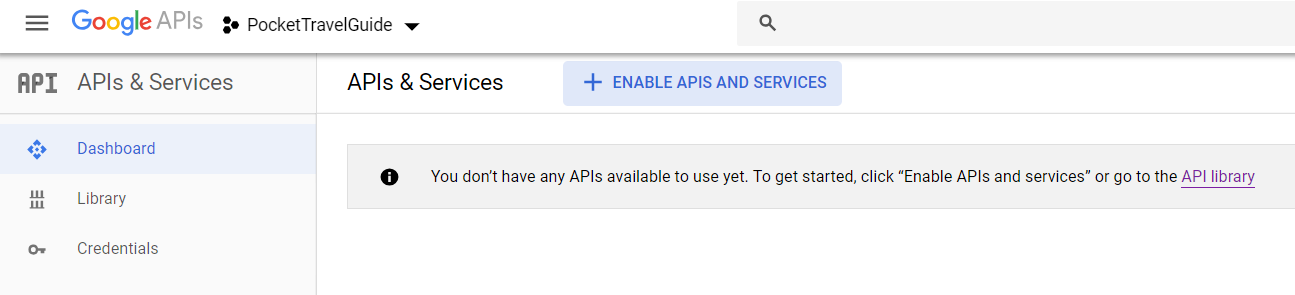
Trebuie sa facem click pe CREATE PROJECT pt a crea . Aici va trebui sa dam un nume noului proiect:



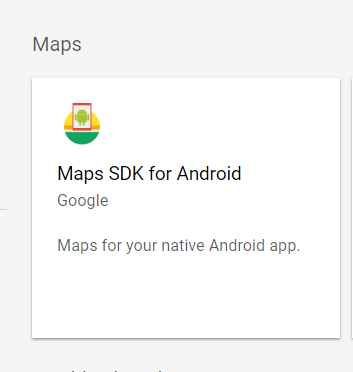
Si dupa facem click pe Create si ne va crea proiectul:



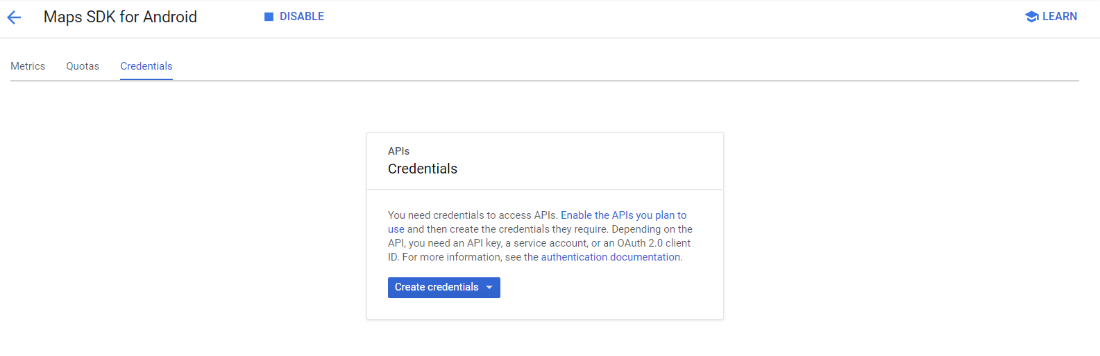
Dupa trebuie sa ma duc in dashboard si sa adaug apis



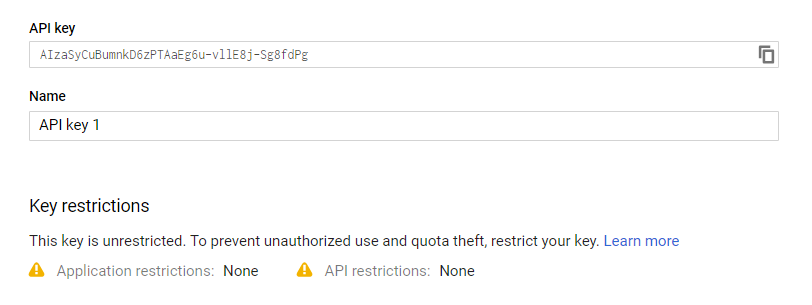
Pe noi ne intereseaza:



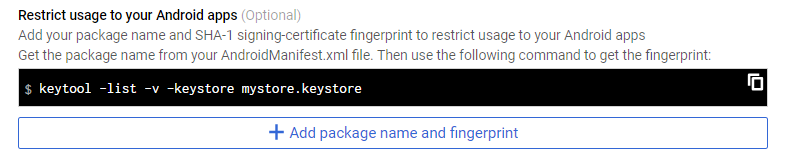
Dupa ce i-ai dat enable trebuie sa ii adaugi credentialele:



Selectam ca vrem sa creeam o cheie api si dupa o primim impreuna cu un nume si avem posibilitatea sa adugam niste restrictii:



Pentru a dauga restrictia de a acesa cheia numai de pe dispozitive android trebuie sa facem urmatoarele:



Adugam numele pachetului:

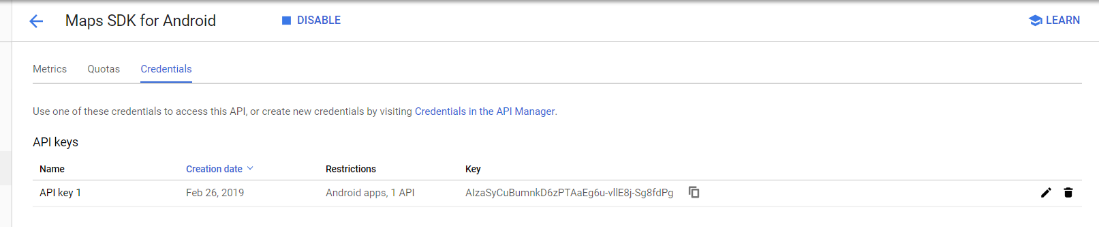


Pentru a obtine SHA-1 trebuie sa facem ce scrie aici: <https://stackoverflow.com/questions/27609442/how-to-get-the-sha-1-fingerprint-certificate-in-android-studio-for-debug-mode>

Dupa introducem SHA-ul pe site si setam si restrictiile pe api ca:



Dupa asta dam save si obtinem cheia:

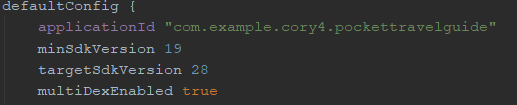


Vom copia cheia si in android studio in manifest vom adauga un tag de meta-data unde vom specifica cheia:



Dupa mai trebui sa facem un meta-data in care sa specificam ca vom folosi play services:



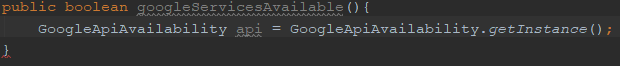
Ininte de a rula ca sa nu avem probleme cu numarul de referinte trebuie sa adaug in build.gradel(Module:app) urmatoarele(<https://developer.android.com/studio/build/multidex>):



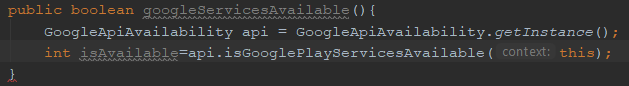
1. Verificarea ca utilizatorul aplicatiei are serviciile play:

Mergem in MainActivity.java si creeam o metoda care sa verifice daca exista:

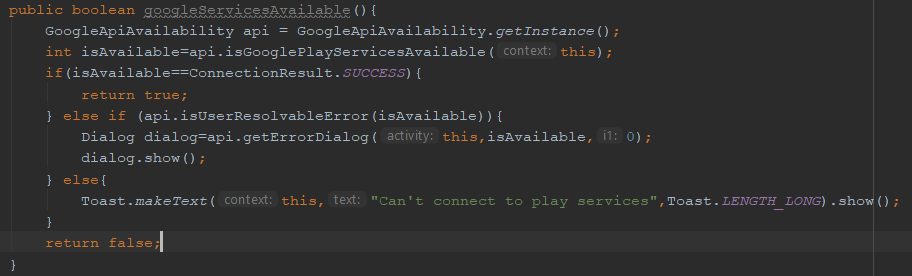
In aceasta metoda voi folosi o clasa care se numeste GoogleApiAvailability si luam o instanta a acesteia.



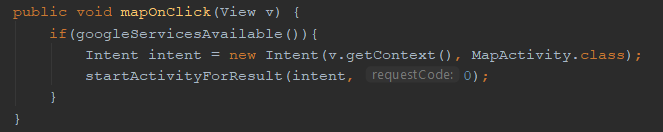
Dupa asta verificam daca este disponibila in contextul curent:



Daca este disponibila atunci se returneaza true. Si daca sunt probleme verificam daca se pot rezolva, daca da afisam o fereastra de dialog cu eroarea, daca nu anuntam pe utilizator. Daca nici una din chestii nu este valabila se va returna pur si simplu false.

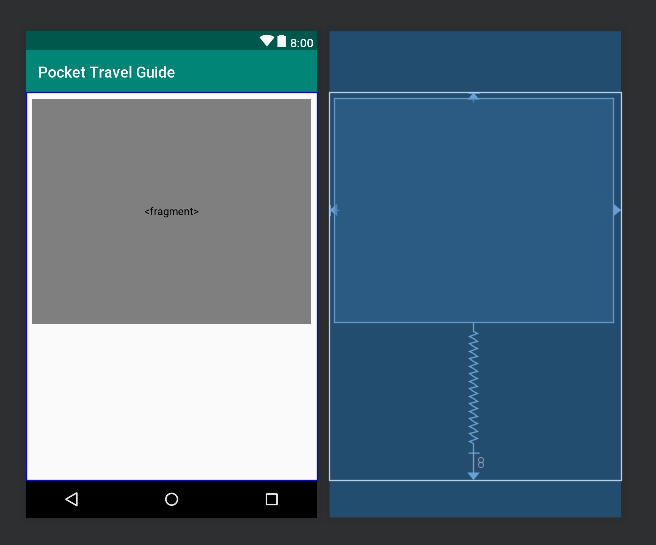


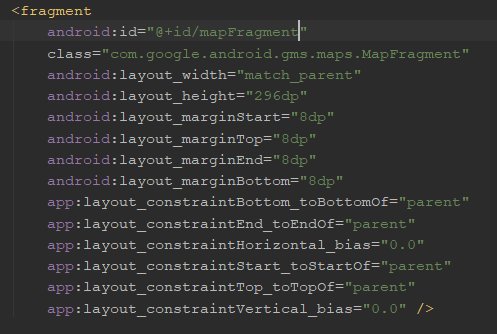
Si acesta metoda o vom apela din actiunea onClick a butonului Map si in cazul ca e de succes vom putea avea acces la harta prin apasarea butonului:



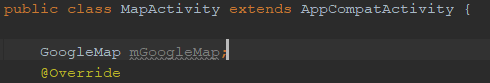
1. Trebuie sa creez un obiect cu care sa putem afisa google maps in aplicatie:

Vom face un fragment in interiorul activity\_map.xml de tipul MapFragment:

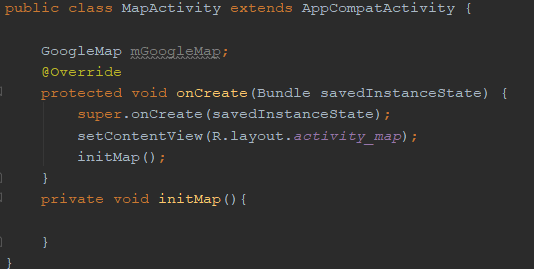




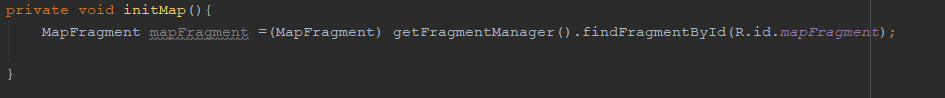
Dupa in MapActivity.java voi crea un obiect de tip GoogleMap:



Dupa creeam o noua metoda initMap care este apelata in onCreate



In aceasta metoda vom lua fragmentul:



Dupa incercam sa luam detaliile despre harta si sa indicam obiectul hartii prin functia getMapAsync(). Si trebuie sa fac o functie onMapReady. Vom astepta o vreme ca harta sa se incarce dupa vom apela functia asta si vom asoia parametrul ei obiectului nostru mGoogleMap

1. Ca sa punem ca harta sa arate catre o anume locatie trebuie sa mergem pe net la google maps sa cautam locatia si sa-i luam latitudinea si longitudinea si in metoda onMapReady sa scriem functia goToLocation de latitudine si longitudine (pe care o facem noi).

Prima data creeam un obiect LatLng. Mai creeam si un obiect de tip CameraUpdate. Dupa asta folosim obiectul google map mGoogleMap pt a muta camera in functie de update.

1. Daca vrem sa facem zoom in pe locatie facem o functie similara cu cea de sus.

# Despre (informatii despre oras generale + imagini 3D)

# Help (cum se foloseste aplicatia + chat unde se pot intreba diverse)

# Optiuni (sunet, notificari etc.)

# Recomandari (lista cu locatii recomandate pe categorii)