**Proiect SCD**

**Aplicație GPS**

**Student: Stoleru Cristian-Andrei**

**Grupa: 30142**

**Profesor Coordonator: Miron Radu**

**Cuprins**

[**Obiective** 3](#_Toc60757707)

[**Specificații** 4](#_Toc60757708)

[Funcționale 4](#_Toc60757709)

[Non-funcționale 4](#_Toc60757710)

[**Proiectare** 4](#_Toc60757711)

[**Detalii de implementare** 8](#_Toc60757712)

[**Testarea** 9](#_Toc60757713)

[**Concluzie** 12](#_Toc60757714)

# **Obiective**

In cadrul acestui proiect putem regăsi următoarele obiective necesare pentru realizarea sa:

* Realizarea unei baze de date ce sa retina pozițiile geografice a unui telefon;
* Realizarea unui serviciu web ce sa implementeze operațiile CRUD pentru baza de date;
* Testarea operațiilor printr-un client realizat cu arhitectura REST. In acest proiect a fost folosit Swegger-ul;
* Implementarea unei aplicații Android care sa:
  + Citească coordonatele GPS de la un telefon Android;
  + Generează unui Id unic pentru aplicație;
* Implementarea unei aplicatii WEB care sa:
  + Afișeze datele de la un terminal;
  + Afișeze pe o harta locațiile in care a fost telefonul cu acel terminal.

# **Specificații**

## Funcționale

Pentru punerea in execuție a acestui proiect sunt necesare următoarele resurse hardware:

* Unitate centrala, ce are rolul de a găzdui atât serviciul WEB cat si pagina WEB;
* Smartphone Android, pentru a transmite informațiile legate de coordonatele geografice si id-ul terminalului;
* Conexiune la Internet pe ambele dispozitive;

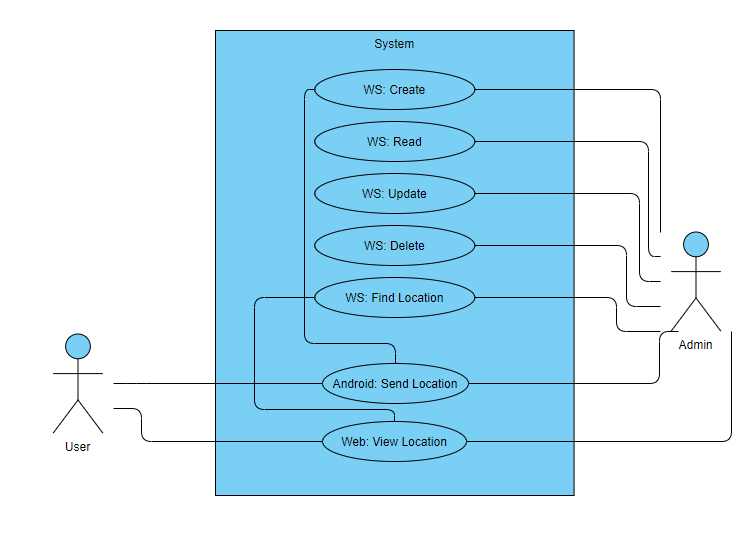
Clientul v-a transmite informații legate de coordonatele sale geografice prin intermediul telefonului personal informații către Web service. Apoi când acesta dorește sa verifice locațiile pe care le-a vizitat, va accesa un link unde va introduce id-ul terminalului telefonului (IMEI/MAE) si datele in care acesta va dori sa vizualizeze locațiile. Acestea vor fi prezentate utilizând o harta oferita de cei de la Google.

## Non-funcționale

Aplicația trebuie finalizata pana la sfârșitul semestrului 1 din anul 4 (12 Ianuarie) si sa respecte o arhitectura de tipul REST. Nu exista constrângeri de timp in ceea ce privește răspunsurile sistemului la diverse acțiuni sau interogări.

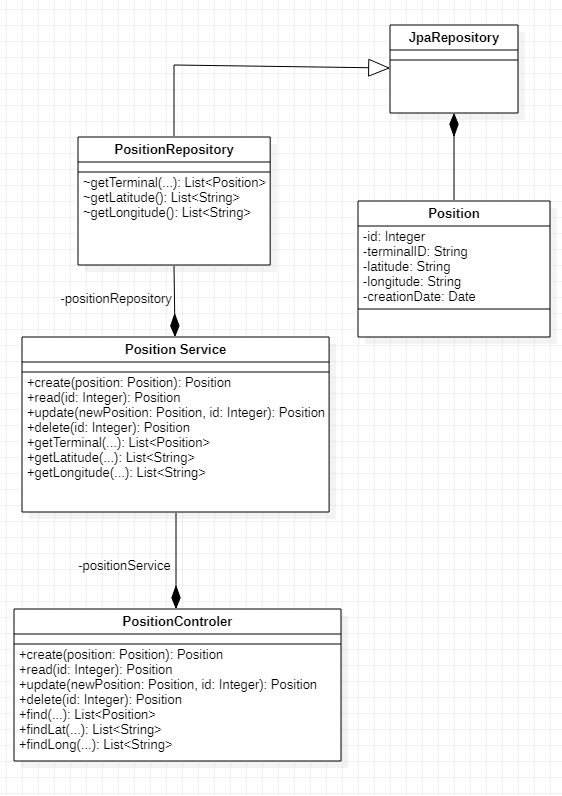
# **Proiectare**

In continuare voi prezenta o vedere asupra aplicatiei in care se observa modul in care utilizatorii au acces la ea.



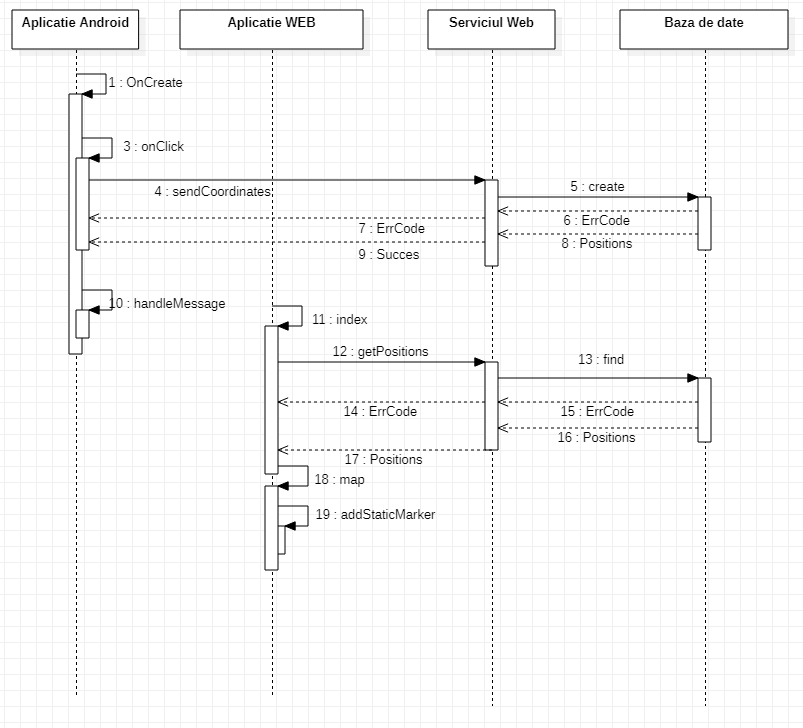
Figură Use case Diagram

Cea mai importanta partea a proiectului este serviciul Web. Pentru o înțelegere pe deplina a acestuia in continuare este prezentata diagrama claselor.

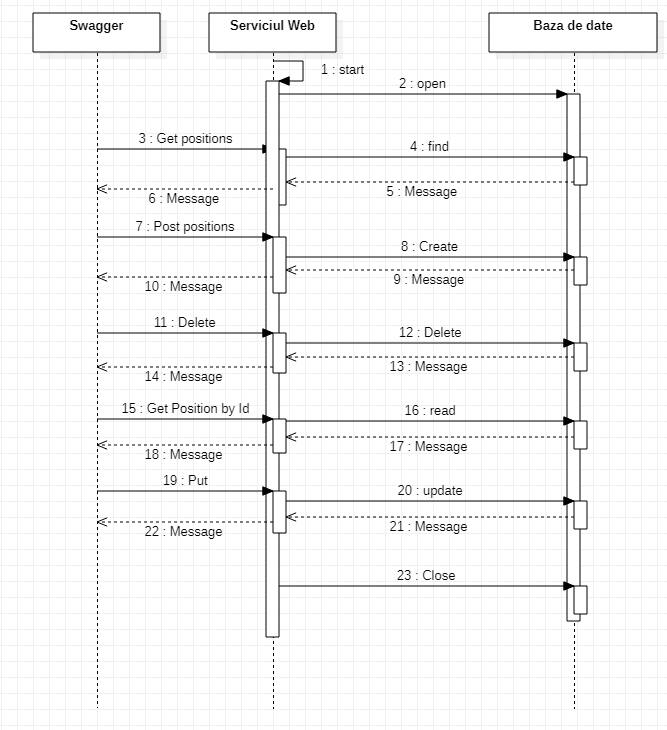


Figură Diagrama Claselor

Diagrama secventiala a acestei aplicatii este prezentata mai jos:



Iar pentru serviciul Web avem:



# **Detalii de implementare**

Modul cum am implementat aplicația li voi împarți in trei categorii, in funții de obiectivele pe care acestea le realizează:

* **Baza de date;**

Pentru realizarea bazei de date s-a folosit pachetul celor de la MySql unde am creat baza respectiva cu tabela positions folosind aplicația WorckBench 8.0.

Hostarea bazei de date s-a realizat local la portul 3306.

* **Serverul WEB;**

Pentru acesta a fost nevoie de folosirea unui mediu de dezvoltare integrat (IDE) pentru a putea scrie cod. Am decis sa folosesc IntelliJ cu versiunea 11 de Java.

Pentru a reduce timpul de proiectare a aplicației s-a folosit Swing, o interfața grafica cu utilizatorul al limbajului Java deținuta de Oracle.

Pentru testarea operațiilor serviciului web, am ales sa folosesc un API din pachetul Swing. Acesta a fost introdus prin adnotarea „@EnableSwagger2” la pornirea aplicației.

Hostarea serviciului este una locala, realizata pe portul 8082.

Acesta va realiza conexiunile intre baza de date si restul aplicațiilor ce au nevoie de acces la ea. In serviciul web au fost configurata operațiile CRUD clasice plus operații care se ne transmită informații in funcție de terminalId si doua date calendaristice.

* **Aplicația Android;**

Pentru a realiza respectiva aplicație a fost nevoie de un nou mediu de dezvoltare ce sa ne permită folosirea pachetelor de android a celor de la Google. IDE-ul folosit a fost Android Studio, unde interfața a fost scrisa in XAML iar codul in Java.

Pentru testarea aplicației a fost nevoie de instalarea unui emulator ce sa simuleze folosirea unui telefon mobil. Emulatorul ales de mine este cel al telefonului: Pixel\_3a\_API\_30\_x86.

Aplicația va cere utilizatorului următoarele permisiuni: accesul la internet, datele stării telefonului (aici putem extrage IMEI-ul) si locația (pentru a extrage longitudinea si latitudinea).

Datorita faptului ca folosim un emulator, codul IMEI transmis de acesta este null, iar locația de unde transmite informația este sediul Google din San Francisco (Googlepex).

Aplicația comunica informațiile obținute prin portul local 8082.

Interfața acesteia este formata dintr-un fundal alb si un singur buton. La apăsarea acestuia, se va prelua de pe telefon următorul set de date: terminalId, latitude si longitude. Acestea vor fi trimise către serviciul WEB ce va crea in baza de date o tranzacție ce va adaugă informațiile in tabela positions.

* **Aplicația WEB;**

Mediul de dezvoltare folosit a fost IntelliJ, însă limbajele de programare utilizate au fost HTML si Java Script.

Pentru testare si pentru debugging a fost folosita consola browser-ului Google Chrome.

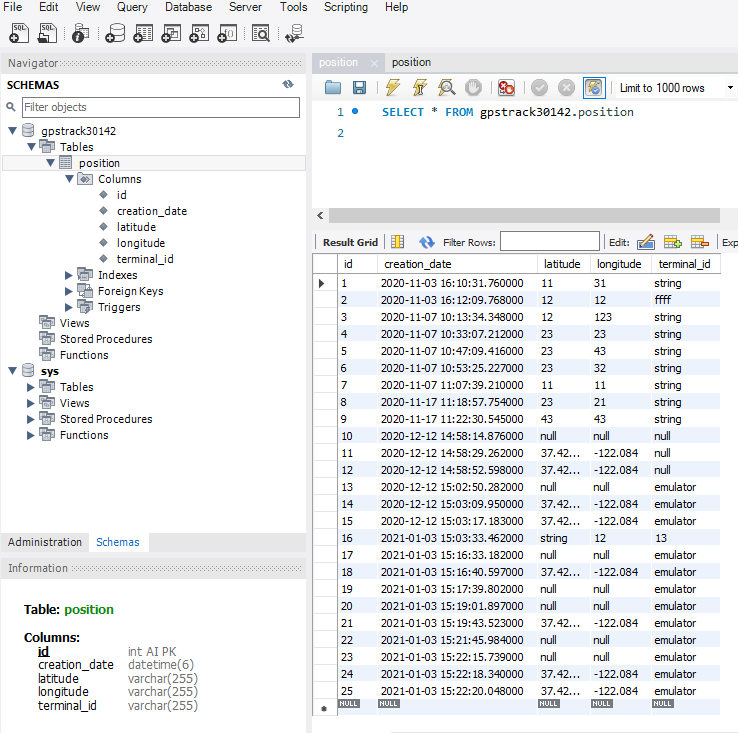
Aplicația este hostata pe portul local 8082.

Aceasta conține doua pagini web: index.html si map.html. In prima pagina regăsim 3 textbox-uri (in care putem introduce terminalId, startDate si endDate) si doua butoane. La apăsarea primului buton, acesta ne va transmite entitățile din baza de date ce au terminalId-ul introdus mai sus si care au fost create intre cele doua date calendaristice si le va afișa. La apăsarea celui de al doilea buton, textul afișat anterior va fi serializat si transmis următoarei pagini. Odată cu aceasta acțiunea se va deschide pagina map.html. In aceasta pagina se regăsesc doua butoane si o harta (Google Maps). Primul buton va lua datele transmise de pe prima pagina si le va prelucra astfel încât sa obțină latitudinea si longitudinea fiecărei entități si le va afișa pe harta utilizând un ping pe harta. Al doilea buton ne va întoarce la pagina index.html.

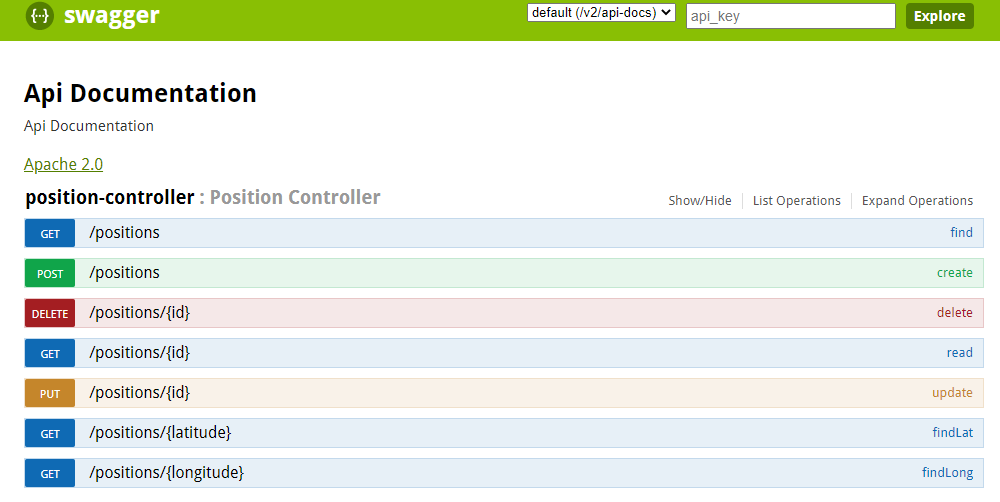
# **Testarea**

Pentru testarea datelor au fost realizate diverse operații pe toate cele 3 aplicații. In urma lor am obținut următoarele:

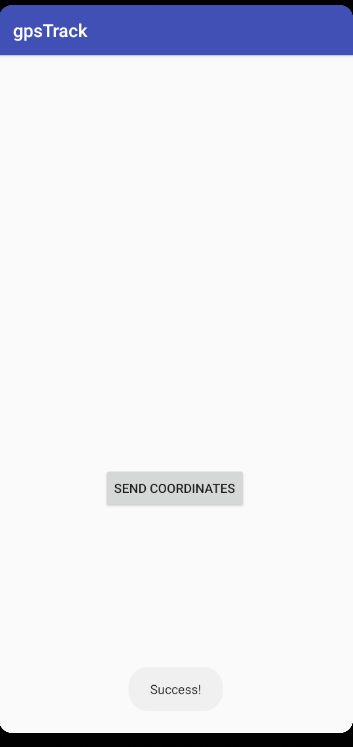
* Baza de date



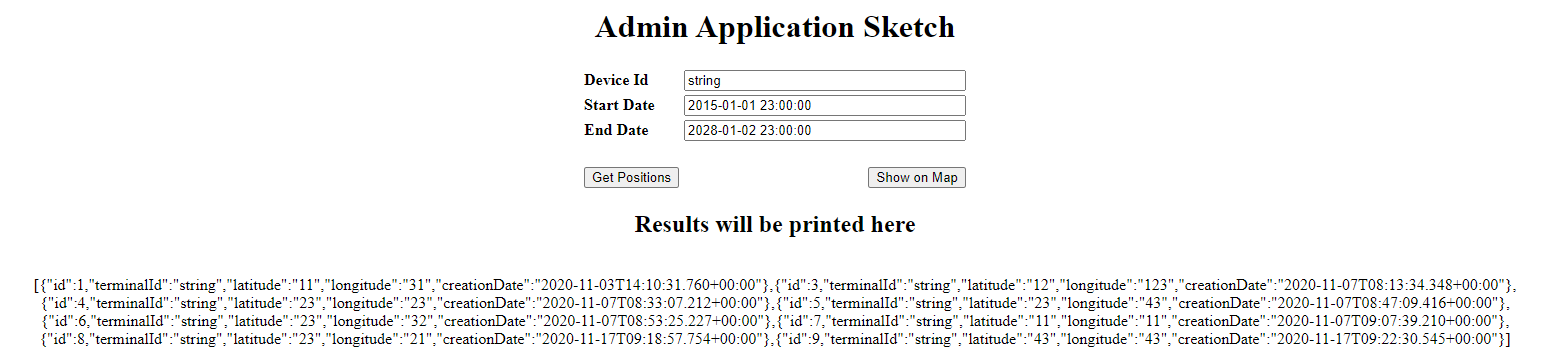
* Swagger

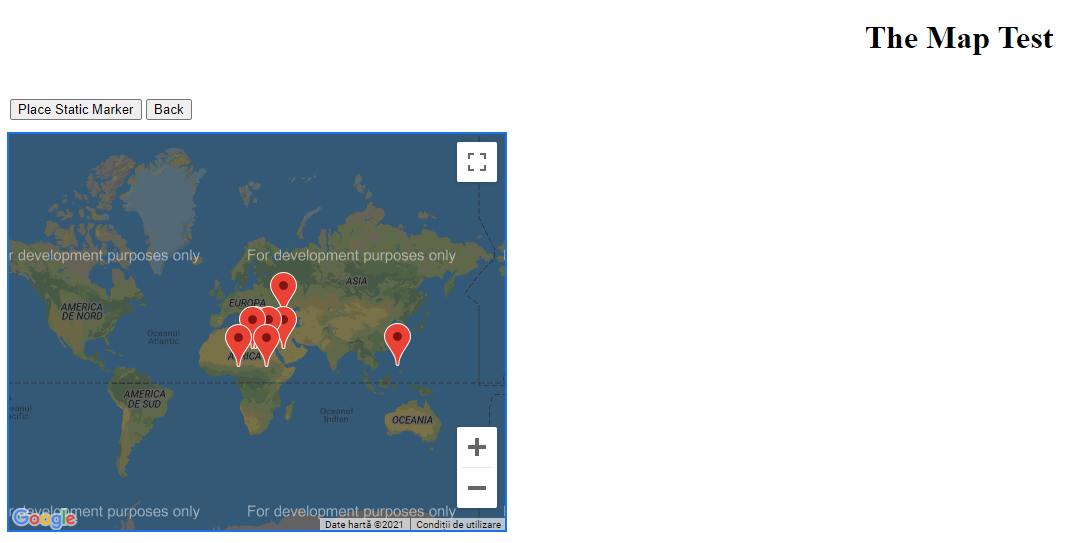


* Aplicatia Android



* Aplicatia Web



Figură Pangina Index.html

Figură 4 Pagina Map.html

# **Concluzie**

In urma acestui proiect am învățat sa lucrez cu o arhitectura de tipul REST si cum sa configurez un serviciu WEB cu ajutorul unui controler.

Pe viitor doresc sa îmbunătățesc aplicația dezvoltând-ui o interfața mai prietenoasa, atât pentru cea de android cat si pentru cea de WEB. Problema majora pe care o are partea de backend este aceea ca atât aplicația cat si utilizatorul au nevoie de codul IMEI pentru a le utiliza, acest lucru doresc sa îl înlocuiesc realizând o pagina de login/sign-up atât pe android cat si pe web.

O alta problema este aceea ca aplicațiile pot rula doar pe rețeaua locala, pentru a putea fi accesate global va fi nevoie de publicarea lor pe un server public.