Transport Public Timisoara

Transportul public reprezintă una din principalele modalități de deplasare în interiorul unei localități, punând la dispoziție publicului larg posibilitatea de a ajunge dintr-un loc în altul, fără a avea un mijloc de transport personal chiar și între distanțele mari din mediul urban. În același timp folosirea transportului public poate fi un factor important pentru reducerea cheltuielilor personale daca este folosit în schimbul autovehicolului. Alte beneficii ale transportului public ar fi: decongestionarea traficului, reducerea emisiilor de carbon, crearea de locuri de muncă și multe altele. Așadar observăm că transportul public nu ar fi benefic doar pentru noi din punct de vedere financiar cât și mediului înconjurător, totuși în utilizarea lui intervin două probleme esențiale:

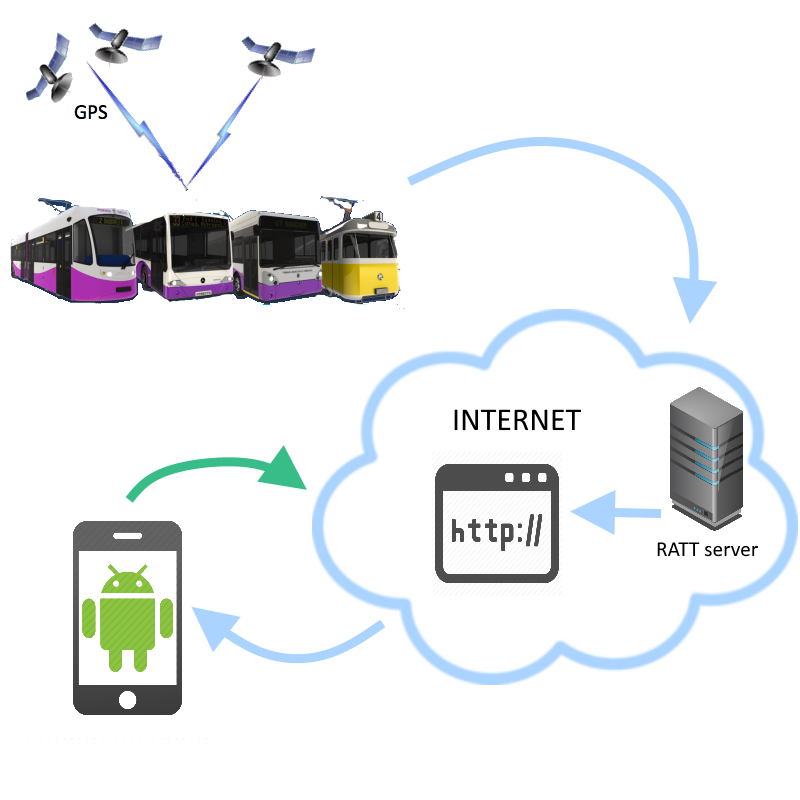
* cunoașterea traseele și stațiile mijloacelor de transport;
* cunoașterea momentului când mijlocul de transport ajunge într-o anumită stație.

Pentru rezolvarea primei probleme și anume informarea populației cu privire la rutele mijloacelor de transport în comun, cea mai frecventă metodă folosită este aceea de a afișa în majoritatea stațiilor și în interiorul mijloacelor de transport, o hartă ce prezintă toate traseele și statiile din acel oraș. A doua problema este rezolvată în mare parte ca și prima, fiind afisat un orar prestabilit cu intervalele de circulație sau in cel mai bun caz un afișaj cu informații în timp real despre momentul cand va ajunge mijlocul de transport în acea stație. Totuși chiar daca ar părea că nevoia de informare a populației ar fi îndeplinită, o întrebare ar fi: cât de repede se actualizează hărțile din stații din tot orașul la o modificare de traseu a unei linii? Daca ar fi o modificare temporară de traseu cât de accesibilă va fi aceasta informație? Sau cum putem folosi transportul în comun daca nu știm unde sunt stațiile sau care ar fi cea mai apropiata stație de noi? Observăm că doar aceste căi de informare aduc după sine îngreunarea folosirii transportului în comun prin incertitudinea unor informații și folosirea ineficientă a transportului prin asteptarea nedorită în stație. Pentru a intra în profunzimea problemei trebuie sa ne raportăm la un caz real și anume orașul Timișoara.

Orașul Timișoara dispune de tot ceea ce a fost descris mai sus cât și de mijloace suplimentare de informare. Site-ul Regiei Autonome de Transport Timișoara (RATT) conține o harta a transportului cu autobuze: urban, expres si metropolitan fără trasee deviate temporar, existând la momentul de față trasee neactualizate, mai conține o hartă a transportului electric tot fara trasee derivate temporar și nu în ultimul rând conține informații cu privire la timpul în care va ajunge un mijloc de transport într-o stație. Pe lânga aceasta mai există o aplicație mobilă android care ne oferă un acces mai ușor la informațiile afișate pe pagina oficială R.A.T.T.

Totuși pentru folosirea cea mai eficientă a transportului public în Timișoara ar fi nevoie de o sursa de încredere de informare, cu date actualizate în mod continu, să poată fi accesată din orice locație ne-am afla și in care sa gasim atât traseul actual al unei linii cât și timpii de sosire a mijloacelor de transport în comun pentru fiecare stație.

O soluție ar fi o aplicație mobilă ce ar avea acces întotdeauna la stațiile și traseele oficiale ale transportului în comun și la pentru fiecare stație timpul de sosire al fiecărui mijloc de transport.



Android Studio

Android Studio este platforma oficială de dezvoltare a aplicațiilor mobile Android, construită având la baza mediul de dezvoltare IntelliJ IDE creat de JetBrains. Este distribuită de către Google sub licența Apache Licence 2.0 ce oferă utilizatorilor dreptul gratuit de a folosi platforma gratuit atât în scopuri personale cât și în cele comerciale.

Fiecare proiect din Android Studio conține unul sau mai multe module ce conține urmatoarele foldere:

* *manifests*: conține fișierul AndroidManifest.xml ce prezintă informații esențiale despre aplicație, informații care trebuie să fie cunoscute de către sistem înainte de a rula codul. În conținutul fișierului este precizat numele pachetului Java al aplicației, aici se declara ce fel de permisiuni are nevoie aplicația pentru accesul parților protejate a unor diferite funcții API sau legături între alte aplicații, este descrisă lista de componente din aplicație: activități, servicii și în ce fel de condiții se rulează fiecare și lista de librarii folosite;
* *Java*: conține fișiere de cod sursă Java;
* *res*: conține fișiere non-cod cum ar fi scheme XML (layouts), anumite texte predefinite si imagini bitmap.

Mediul Android Studio folosește Gradle ca și fundație pentru sistemul de compilare, doar cu mai multe caracteristici configurate special pentru Android. Sistemul de compilare rulează ca o componentă integrată în meniul Android Studio și permite personalizarea, configurarea si extinderea procesului de compilare, crearea de multiple fișiere APK pentru aplicație, cu diferite caracteristici folosind acelasi proiect și aceleași module, de exemplu crearea de versiuni separate ale aplicației: o versiune gratuită si una platită. Fișierele de compilare Android Studio se numesc build.gradle și sunt fișiere text ce folosesc sintaxa Groovy pentru configurarea compilării. La importarea unu proiect existent, Android Studio genereaza automat fișierele necesare pentru compilare.

Android Studio asista la scrierea codului încât dispune de un dispozitiv de analiză de performanță a codului pe masură ce este scris. În modul de depanare ne ofera informații la fiecare linie cu privire la: valorile variabilelor, obiectele referențiate, valorile returnate de către metode. Tot în ajutorul dezvoltatorului, în vederea unei depanari și monitorizări cât mai exacte a aplicației, Android Studio dispune de informații cu privire la memorie si folosirea procesorului, gasirea obiectelor dezalocate, pierderilor de memorie, optimizarea performanței grafica si analizarea traficului internet.

Pentru a rula aplicația avem două opțiuni: folosind Android Emulator sau prin conectarea prin USB a unui dispozitiv cu un sistem de operare Android. Dupa selectarea uneia dintre opțiuni, se va instala aplicația pe emulator sau dispozitivul conectat, urmând ca la a doua rulare sa nu mai fie instalat din nou ci doar actualizat cu noile modificări efectuate, aceasta reducând semnificativ timpul de lansare pe dispozitiv sau emulator.

Google Maps Android API

Google Maps Android API permite dezvoltatorilor să înglobeze harți Google în orice aplicație. Interfața manipulează în mod automat accesul la serverele Google Maps prin descarcărcarea datelor necesare, afișarea hărții și raspunderea la schimbarea sau apropierea unei anumite locații pentru furnizarea unor detalii suplimentare. Deasemenea permite adaugarea de diferite componente optice pentru îmbunatățirea experienției utilizatorului cum ar fi: adaugarea anumitor puncte semnificative pe hartă (eng. marker), desenarea a diferitor linii sau poligoane pe harta și alte straturi grafice suprapuse peste harta de bază oferită de interfață.

Pentru a accesa interfața, aplicația are nevoie de o cheie API pentru a i se permite accesul la serverele Google Maps. Acest cod este generat gratuit și nu limitează numarul de utilizatori al aplicației.

În mod implicit, fișierul XML ce definește locul unde va fi poziționată harta în aplicație, res/layout/activity\_maps.xml, va conține următorul cod:

<fragment xmlns:android=<http://schemas.android.com/apk/res/android>   
    xmlns:tools=<http://schemas.android.com/tools>   
    android:layout\_width="match\_parent"   
    android:layout\_height="match\_parent"   
    android:id="@+id/map"   
    tools:context=".MapsActivity"   
    android:name="com.google.android.gms.maps.SupportMapFragment" />

Fișierul Java, MapsActivity.java, va implementa interfața OnMapReadyCallback care necesită suprascrierea a doua metode: onCreate si onMapReady. Prima metodă va fi apelată la pronirea aplicației și va porni procesul asincron de descărcare a hărților, iar a doua metodă se va executa atunci când hărțile s-au descărcat cu succes și cuprinde configurațiile de inceput ale harții cum ar fi: poziția de început a harții definită prin latitudine si longitudine, valoarea de apropiere față de coordonatele stabilite, metode pentru implementarea evenimentelor de selectare ale unor elemente grafice de pe hartă.

Referințe:

https://developer.android.com/studio/intro/index.html

https://developers.google.com/maps/documentation/android-api/