A gyakorlat első fázisában a SAP HANA rendszerrel ismerkedtünk meg. Ez abban merült ki, hogy anyagokat olvastunk a környezetről, mini projekteket próbáltunk beüzemelni a saját homokozónkban, és elkezdtük tervezgetni a későbbi projektet.

Második fázisnak nevezném a rákövetkező 2-3 hetet, ami alatt egy demó projektet kellett összeraknunk. A kapott adattáblából az adatokat feldolgoztam, és eltároltam java oldalon, majd JS oldalon vizualizáltam az adatokat. A projekttel az első számú célom az volt, hogy alapszintű JS tudásra tegyek szert, és hogy megismerkedjek egy térkép API-val. Ebben a projektben a Bing Map API-t használtam, és kellemes meglepetésként ért, hogy egyáltalán nem bonyolult. A végeredmény egy kezdetleges „saját” REST API-t használó alkalmazás volt, ami szépen minden címhez tartozó koordinátát lekérdez, majd a lekérdezések végén megjelenít egy térképen.

A projekt kezdetekor 3 fős csoportként szakadtunk le a többiekről. (Kiss János, Borbély Dávid, Vitrai Gábor) Innentől kezdve a csoport többi ragjától függetlenül dolgoztunk, de az adatbázis nem változott. Abban maradtunk, hogy Angular-t fogunk használni, hiszen fejlesztői szempontból nagyon sok kényelmes funkcióval segít ki minket. (Automatikus újra töltés, nodeJS)

A végső és legfontosabb fázis a tényleges alkalmazás kifejlesztése. Ezt Cserép Máté tanár úr témavezetésével tettük. A terv az volt, hogy két adatbázisból dolgozunk. Az egyikből a geoJSON adatokat kérdezzük le a térkép vizualizációjához, a másikból pedig az eredeti a Vincellér tanár úrtól kapott céges adatokat dolgoznánk fel. Ez meg is valósult. 2 Térkép megjelenítőt használtunk: OpenLayers-t és a Leaflet könyvtárakat. Igyekeztünk felosztani a munkát 3 főre, ehhez természetesen git verziókövetőt használtunk, és külön ágakon dolgoztunk.

A Leaflet-et arra használtuk, hogy egy megye kiválasztásánál, interaktívan megjelenjenek az ott található települések, zöld vagy piros pont formájában. Egy település akkor jelenik meg a térképen, ha az adatbázisban van cég azon a településen, míg pirosan, ha nincs. Ezek eldöntésére SQL lekérdezéseket használtunk, amik szerver oldalon kerültek kiszámításra, így a kliensnek kevesebbet kellett számolnia. A kiválasztott településnek a neve és statisztikái megjelenítésre kerültek a térkép alatt. Az OpenLayers -t arra használtuk, hogy a települések külterületi határait jelenítsük meg. Ezekből több is kijelölhető, és szintén a települések összforgalma alapján mutat statisztikát.

Összehasonlítás:

Kezdetben a Leaflet sokkal egyszerűbbnek tűnt, jobban kidogozottabbnak. Mint az később kiderült az OpenLayers objektum orientáltságának köszönhetően egyszerűbben kezelhető, és ami még fontosabb, jobban átlátható. Véleményem szerint mindkét térképkezelőt maximálisan kihasználjuk, és a projekt megfelelő közelségben maradt az SAP HANA környezettel. A szerver geoJSON formában exportálja számunkra a lekérdezések eredményeit, amit a JS nagyon könnyen fel tud dolgozni. Kód mennyiségét nézve nagyából megegyeznek, és hasonló funkciókat is látnak el, hiszen részben ez is cél volt, hogy összehasonlítható eszközöket használjunk.

A projekt vége egy nagyon látványos adatvizualizációs eszköz lett. Teljesítette az elvárásokat, kihasználta az SAP HANA által nyújtott eszközöket, két féle könyvtárat is megtanultunk kezelni, és az adatbáziskezelésben is otthonosabban mozgunk. Megtanultunk jobban csapatban dolgozni, és megismertük ennek nehézségeit, előnyeit. (Például: a tény, hogy valaki el tud készíteni egy szolgáltatást ami kell a másiknak, nagyban megkönnyíti a munkáját mindenkinek, de nagyon sok problémánk volt, még a git használatával is, számos konfliktust okozott, amiket meg kellett értenünk majd megoldanunk. A gyakorlat hasznosan telt, ezt ki merem jelenteni mindhármunk nevében.