**Kivonat! (Első két hét, részletesebb a haladás szempontjából)**

**2019.07.01. Hétfő:**

Első projektmegbeszélés. Érdeklődtem a Java rész és a Front-end iránt. Tisztáztuk a jelenléti és egyéb követelményeket, és a használandó Technológiákat. (SAP HANA)

**2019.07.02. Kedd:**

Terv: Tutorial-okat nézni, megismerkedni az Online Platformmal. Java alkalmazás készítésének utánanézése, és a program összekötése az SAP-val. Ha esetleg marad idő, Angular-nak utánanézni, kipróbálni.

Haladás: Számos kezdő, és középhaladó oktatóanyagot átnéztem és kipróbáltam köztük: Java, Java Web, Angular, Node.js. Megnéztük közösen hogy lehet adatbázist létrehozni. Az egyik tutorialt követve megkíséreltem beüzemelni egy Eclipse IDE-t hogy az képes legyen kommunikálni a SAP-al, de valamilyen port probléma miatt nem tudja letölteni a dependenciákat. Holnap ha ezt sikerül megoldanom megpróbálom összekapcsolni egy adatbázissal. (Esetleg Hibernate)

A nap végén Ismerkedtem az Angularral, de arra már csak kevés idő jutott.

**2019.07.03. Szerda:**

Terv: Megkérdezni, mi a projekt, git, todo kezelő? Port probléma megoldása, Online Spring kipróbálása, komplikáltabb java applikációt kipróbálni a platformon. Megtanulni JS-ezni.

Haladás: Adatbázishoz, SAP HANA-hoz kapcsolódó fogalmakról volt szó. Elkezdtünk gondolkodni azon hogy különböző platformokon hogyan valósítjuk meg a geo vizualizáció.

Felmerült hogy bitmap-et készítünk, és azt jelenítjük meg, hogy megjelenítjük a térképet és rá rajzolunk, pontokat, amikre ha kattintunk átirányít, továbbá felmerült külső 3rd party app bevonása. Haladtam tovább a tutorialokkal, elkezdtem megismerkedni a JS-el, és a SAPUI5-al. Elolvastam az újonnan kapott tutorialokat, HANA XS…

Megpróbáltam a Java Web Application-ra (Servlet működésének utánanéztem) komolyabb folyamatokat deploy-olni.

**2019.07.04. Csütörtök:**

Terv: Folytatni az új tutorialokat. Megpróbálni összehozni egy adatbázist egy egyszerű frontend lekérdezéssel, kapcsolat tesztelése.Java REST Controllerrel foglalkozó tananyag elvégzése, mivel szükség kell REST API-ra a szoftver fejlesztése során is.

Haladás: A reggel folyamán nekiláttam a tervezett tananyagnak, majd szoftveres problémába ütköztünk, ugyanis a SAP Cloud Service leállt, és nem tudtunk Buildelni adatbázist. Megkaptuk a mini projektet így mindenki elkezdett egy részével foglakozni. Volt aki a geokódolás után kutatott, volt aki front-endet kezdett írni. Mi az adatbázis feltöltésével próbálkoztunk viszont tekintettel arra hogy nem volt elérhető, megfelelő sémájú deployol-ható adatbázisunk, ezt nem tudtuk megtenni. Megoldásként az ELTE aramis adatbázis szerverén csináltunk egy táblát (ABIOWE.SAP) aminek köszönhetően páran neki kezdhettek a szükséges SQL kódok elkészítésének, hogy előállítsuk a szükséges kimutatásokhoz szükséges táblákat. Elkezdtem kutatni hogy hogyan érhető el az adatbázis kliensből, továbbá utánanéztem az ODATA-nak.

**2019.07.05. Péntek:**

Terv: Java kliens elkészítése és ResultSet-be összegyűjteni a lekért adatokat a szerverről (jdbc). Az ezzel kapcsolatos 2 tutorial végigolvasása (kb 3 óra). ODATA-ba beleásni magam (online/offline ?) Alkalmas Java kliens létrehozása, esetleg Java Web app deployolása.

Haladás: Ma végre sikerült Build-elni az adatbázist, így eléjük az üres táblát a megfelelő sémával, viszont továbbra sem tudunk adatot importálni a táblába, mert a felület nem működik jól..

Új tev: SAP HANA Egyetemi szervert használni, feltölteni a táblákat, onnan lekérni JSON-ba, és azt feldolgozni.

a) a teljes táblát JSON-a alakítani, és saját lekérdezéseket írni.

b) SAP oldalon megírni a lekérdezéseket/ létrehozni az új táblákat, és azt alakítani JSON-é. Ha ez megvan a front-end a JSON-t dolgozza fel, és minden kliens valamilyen segéd táblával feldolgozza/vizualizálja az adatokat. Minimum követelmény szint: <http://www.orszagleltar.gov.hu/faces/UI-tars2?_adf.ctrl-state=15mtndmp4o_3>

Később: Ha van működő SAP szerverünk, akkor ODATA segítségével fogjuk lekérdezni az adatokat, és a kliens tartalmazni fog egy StringBuildert szerű metódust aminek segítségével le fogunk tudni kérdezni adatot SQL szerű lekérdezéssel.

Azt a tanácsot kaptuk, hogy kezdjünk el dolgozni JSON-al aztán a későbbiekben ha működni fog az ODATA akkor álljunk át rá.

**2019.07.08. Hétfő:**

Terv: JSON adatok feldolgozása Java kliensen, tömbök feltöltése Java Reflexion segítségével.

Haladás: Összedolgozva egy másik hallgatóval, kitaláltuk, hogy Java platformon, felosztjuk a munkát. Én megoldom hogy a backend adatok sikeresen importálhatóak legyenek a szoftverbe, ő pedig megoldja hogy az adatvizualizáció jó legyen. Jelenleg egy JSONObject-eket tároló konténerből próbálom feldolgozható adat halmazzá átalakítani azt.

A Vincellér Zoltán tanár úrral átbeszéltük a tehendőket a hétre: mini projekt elkészítése (Táblák normalizálása->Odata->JSON->Geokódolás->vizualizáció) továbbá a projekt dokumentálása. A Reflexion API használata problémákba ütközött, holnap próbálkozom tovább.

**2019.07.09. Kedd:**

Terv: Odata specifikus konzultáció, Hozzáférés probléma megoldása, Elmaradt beszámoló megírása. Git beüzemelése. Java Reflexion API tanulás, debugolás.

Haladás: Átírtam sokkal hatékonyabbra a JSON beolvasást, ugyanakkor nem sikerült megoldanom hogy a magyar karaktereket is felismerje, az interneten nem találtam megfelelő megoldást rá, pedig órákon keresztül kerestem, ehhez segítséget kell kérnem valakitől. Megoldottuk hogy ODATA-ból ki tudjuk nyerni az JSON fájlt, holnap pedig megoldom hogy URL-ből tudjon olvasni, ha sikerült végre parse-olnom megfelelően.

**2019.07.10 Szerda:**

Terv: Régi JSON parser átírása működőképesre. Megoldani hogy URL-ből is tudjon olvasni, majd merge-lni a a geocode-os projekttel. Esetleg Dokumentáció írás.

Haladás: Kiderült hogy az encode-olási probléma fájl szintű így megoldható volt. Várhatóan nem fog jelentkezni amikor URL segítségével érjük el az adatbázist. Az elérés elméletileg jó ugyanakkor probléma hogy a szervernek nincsen tanúsítványa így az alkalmazás nem hajlandó elérni az ODATA-t, amiből ki tudnánk nyerni a JSON-t. Az adatfeldolgozást tökéletesítettem, és sikeresen mergel-tem Kovács Mátéval, és a lakcímek hosszas feldolgozása után, most már minden egyes címnek le tudjuk kérni a koordinátáit. Következő lépés hogy ezeket az adatokat egy API segítségével megjelenítjük egy front-end felületen. Ha ez megvan még 2-3 grafikont kéne megoldani, és egy Calculation View-et.

**2019.07.11. Csütörtök:**

Terv: 1. Java app kipróbálása Szervleten, 2. Vizualizáló API-t keresni, 3. megoldani a tanúsítvány problémát. 4. Front-end készítése

Haladás: Hosszas kínlódás után úgy döntöttem hogy a Bing ingyenes Map API-ját használom. A probléma ezzel az hogy az előre konfigurált fájlt nehéz előállítani így úgy döntöttem hogy a koordinátákat exportálom ki egy JSON fájlba amit egy JavaScript alkalmazás beolvas és feldolgoz. Erre azért volt szükség mert a Bing API-ja JS-t használ, és nem találtam módot a Java-ba való integrálásra. Ez után back-end oldalon elértem hogy a koordináták ne ismétlődjenek, pusztán optimalizálás céljából. (MIvel durván minden 3. koordináta megegyezik). Ez után a script segítségével megjelenítenem a megszerzett koordinátákat a térképen. Próbálkoztam tanúsítvány importálásával hogy URL-en keresztül el tudjam érni az OData állományt, de jelenleg nem jártam sikerrel.

**2019.07.12. Péntek:**

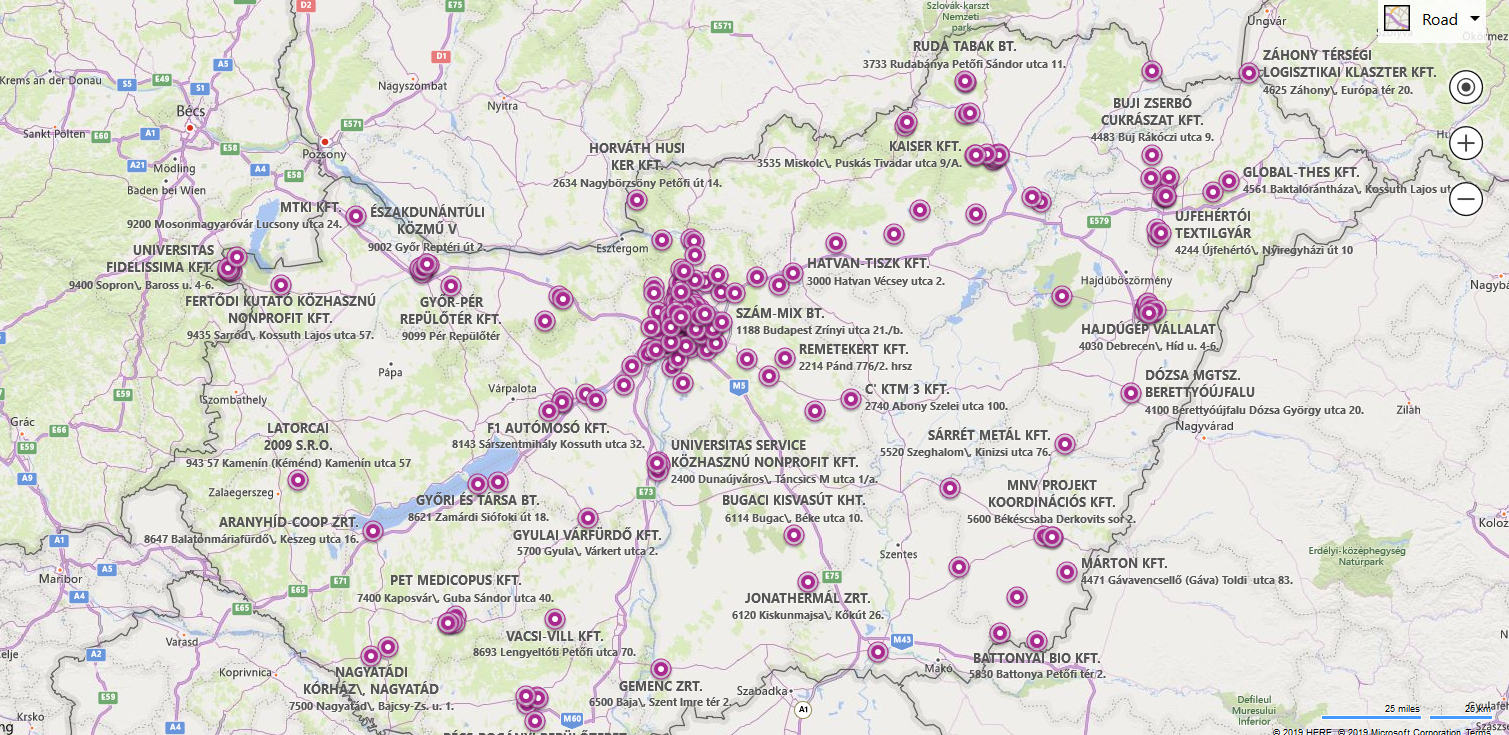
Terv: Megoldani a közvetlen ODATA elérést, esetleg hogy többet mint 1000 db-ot. Cégek nevét is kirakni Bing API-val, JFrame az app-nak, esetleg feltenni a homokozóba.

Haladás: Sikerült megoldani a vizualizálást továbbá készült egy mini JFrame felület is 2 gombbal, a funkciók működtetéséhez. (ProcessData, Visualise), továbbá sikerült elérni hogy a script a beolvasott koordináták alapján megjeleníti az összes pontot a térképen.

**A program működése tömören:**

1. Az első lépés az adatbázis elérése, ez ODATA-n keresztül történik. A Az odata-t lekéri a program JSON formátumban egy String-be majd GSON könyvtárral feldolgozza JAVA oldalon. (Jelenleg jogosultsági problémákba ütköztem, mivel a SAP HANA oldal-t nem hajlandó elérni a kliens mivel nincs tanúsítványa, és a tanúsítvány elfogadtatásához rendszergazdai jogosultsághoz lenne szükségem. Mindazonáltal a kód kész, és le van tesztelve, de jelenleg lokális .json fájlból olvassa be és dolgozza fel az adatokat, amik 100%-ban megegyeznek az URL segítségével lekérhetőkhöz.)
2. A json file parse-olása után az adatok mentésre majd feldolgozásra kerülnek, egy az adattábla sémájával megegyező struktúrájú konténerben. Innen egyszerűen és gyorsan elérhetőek.
3. A későbbiekben a Bing Map API-ját használom az adatbázisból beszerzett címek megjelenítésére. Ezzel az egyetlen probléma hogy egyszerre csak 1 címre lehet manuálisan keresni, ezért úgy döntöttem hogy még Java oldalon minden címhez lekérem a hozzájuk tartozó koordinátákat és azokat csak a lekérdezés után jelenítem meg vizuálisan a térképen. Erre szolgál egy osztály számos metódussal együtt. A getCoordinates() metódus megkapja a címet paraméterben és a Bing API egy szolgáltatása segítségével visszaadja a cím koordinátáit, majd eltárolja azokat.
4. Az idő szűkössége miatt egyszerűbbnek láttam hogy külön böngésző ablakot nyissunk a térképnek. A Font-end azonban JavaScriptet használ a megjelenítéshez, így Java oldalról a program exportálja a koordinátákat (a későbbiekben terv szerint a cég egyéb adatait is) és a böngésző megnyitásakor beolvassa azokat.
5. Ekkor már gyerekjáték megjeleníteni a koordinátákat a térképen.

Tehát a program a következő sémát követi:

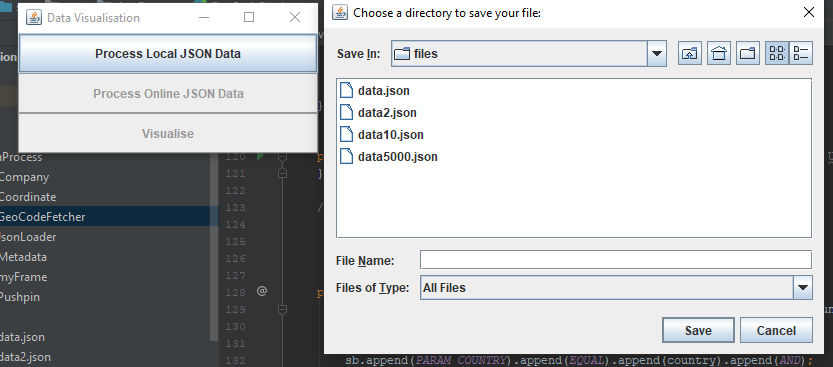
**SAP HANA → ADATBÁZIS → ODATA → JSON → JAVA → JS → BING API**

**2019.07.15. Hétfő:**

Terv: Megoldani a közvetlen ODATA elérést, esetleg hogy többet mint 1000 db-ot. Cégek nevét is kirakni Bing API-val. Indítófelület kiegészítése.

Haladás: Az ODATA állományt a szerverről, továbbra se érjük el, ez probléma, pláne hogy 1000-nél nagyobb adatothalmazt nem tudunk lekérdezni se. Otthon megpróbáltam megoldani Rendszergazdai jogosultsággal a tanúsítvány problémát, sikertelenül.

Kiegészítésre került a JFrame, immár két külön funkcióként elérhető az adatbeolvasás. Elérhető a (jelenleg, a fent említett hiba miatt) nem működő online lekérdezés, továbbá megmaradt a lokális JSON feldolgozás. Az egyszerűség kedvéért készült egy fájl kiválasztó menő, amivel tallózni lehet a feldolgozandó fájlt. Ezen felül a koordináták mellett a teljes, nyers cím és a cég rövid neve is exportálásra került a front-end felé, aminek az eredménye hogy a fenti, frissített képen látható módon megjelennek a címek, és a cégek nevei is a térképen.



**2019.07.16. Kedd:**

Terv: A Microsoft Pin-ek kattinthatóvá tétele., Dokumentáció írás majd projekt beadása GitHubra, JavaScript tanulás.

Haladás: Délelőtt, elkészítettem a projektet feltöltéshez, viszont nincs jogosultságunk a repóhoz. Fejlesztettem a térképet, és igyekeztem optimalizálni a Java kódot is. Délutánom nagy része pedig JavaScript tutorial-okkal ment el, hiszen Szerdától ha minden jól megy a Cserép Mátéval fogok dolgozni, egy JS alapú témában. Utána néztem a *Leaflet* könyvtárnak, és kezelésének. (<https://leafletjs.com/>) (<https://javascript.info>)

Későbbi projekthez.

FONTOS::HOGYAN TUDJUK AUTOMATIZÁLNI (SQL loader, SAP loader) A SCV IMPORTOKAT?

CV, OData, 1 FRONT-END

PÉNTEKIG: WEB-es idén feltöltve a projekt?