**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Informatikatudományi Intézet**

**Programozáselmélet és Szoftvertechnológia Tanszék**

Wireless Tools Analytics

Szerző: Témavezető:

Bezzeg András Benjamin Dr. Nikovits Tibor

Programtervező informatikus BSc. mesteroktató

**Budapest , 2023**

Ide kerül a hivatalos témabejelentő lap.

Tartalomjegyzék

[1. Bevezetés 1](#_Toc153374564)

[1.1. Rövid ismertető: 1](#_Toc153374565)

[1.2. Az alkalmazás: 1](#_Toc153374566)

[2. Felhasználói dokumentáció 3](#_Toc153374567)

[2.1. Futtatási lépések: 3](#_Toc153374568)

[3. Fejlesztői dokumentáció 4](#_Toc153374569)

[3.1. Leírás 4](#_Toc153374570)

[3.2. Fejlesztési terv- és folyamatok: 4](#_Toc153374571)

[3.3. Projektstruktúra leírása: 4](#_Toc153374572)

[3.4. Komponensek: 4](#_Toc153374573)

[3.4.1. Adatbázisrétegek 4](#_Toc153374574)

[3.4.2. 2. Adatfeldolgozás és szolgáltatás 5](#_Toc153374575)

[3.4.3. Felhasználói interfész 5](#_Toc153374576)

[3.5. Futtatási lépések fejlesztőknek: 5](#_Toc153374577)

[3.6. Adatbiztonság és Teljesítmény 5](#_Toc153374578)

[3.7. Összegzés 5](#_Toc153374579)

[4. Összefoglalás és további fejlesztési lehetőségek 7](#_Toc153374580)

[5. Irodalomjegyzék 8](#_Toc153374581)

[6. Melléklet 9](#_Toc153374582)

# Bevezetés

## Rövid kontextus:

A cégünk vezetéknélküli kommunikációs chipek tervezésével , szabadalmaztatásával és gyártásával foglalkozik. Mindemellett biztosit egy integrált fejlesztési környezetet ahol a felhasználó tesztelheti ,megfigyelheti, konfigurálhatja vagy egyéni kódot is írhat ezekre a hardware-ekre. (Továbbiakban ***Studio*** néven erre referálok). Ebben a Studio-ban számos saját és globális vezetéknélküli kommunikációs protokol támogatására van lehetőség, azonban nem mindenki képes hosszú és bonyolult C kód írására, a cég ezeket a felhasználókat is figyelembe véve hozta létre a *Wireless* *Tooling* csapatot aminek énis a tagja vagyok. A célunk átlagos vagy kevésbé tech savy felhasználók részére is lehetővé tenni, hogy egyszerűen generálhassanak hardware kódot. Ehhez különböző webes applikációkat hozunk létre ami az adott technológiához tartozó kódot képesek generálni egy SDK segítségével. Ezeket a webes applikációkat hívjuk ***Tool-***oknak.

## Motiváció:

Ezek a **Tool**-ok tehát egy külső alkalmazásba integrált (**Studio**) webes alkalmazások, aminek a használatáról a **Studio** logolást készít egy felhőben lévő adatbázisba. Mivel ezekenk a Tool-oknak a száma amivel a csapatunk foglalkozik mára bőven kétszámjegyűvé nőtt, elérkezett az igény, hogy ezeket a használati szokásokat feldolgozva, feltérképezhessük és láthatóvá tegyük melyik projekt milyen súlyú hasznot hoz a belefektetett munka arányában.

Ennek a kivitelezésével lettem megbízva a gyakornoki időszakom alatt és ez ihlette a szakdolgozat témáját.

## Az alkalmazás:

A célom tehát egy olyan alkalmazás elkészítése ami ezeket az adatokat felhasználva segíti és információval látja el először a csapatatot majd a menedzsmentet. Ehhez szintén egy webes alkalmazást fogok elkészíteni ami egy belső hálózaton való elérésre alkalmas legyen.

Tehát felhasználók kizárólag cég alkalmazottak lesznek.

Kisebb-nagyobb eltérések akadnak az eredeti verzióval szemben. Az adatokat véletlenszerűen generáltattam apró átalakításokkal , név maszkolással és jelentősen kisebb mennyiségben. A projekt struktúrájának kialakításakor és a technológiák kiválasztásakor a standalone lokálisan futtathatóság és a teljes és egyszerű verziókezelés volt a fő szempont.



1. ábra: Galaxis útikalauz stopposoknak könyv

Használjátok az automatikus számozást az ábrákhoz. Túl sok ábra esetén érdemes ábrajegyzéket beszúrni a dokumentumban az Irodalomjegyzék után. A folyó szövegben hivatkozzatok minden egyes ábrára és táblázatra is kereszthivatkozással (például így: 1. ábra, vagy táblázat esetén így: 1. táblázat).

1. táblázat: Példatáblázat

|  |  |  |
| --- | --- | --- |
| asdf |  |  |
| 1234 |  |  |

# Felhasználói dokumentáció

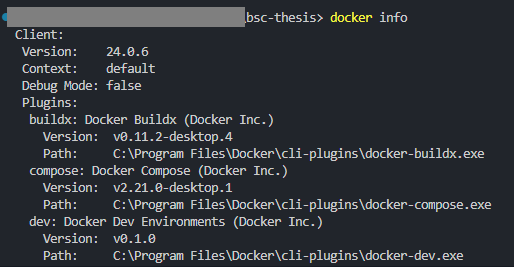
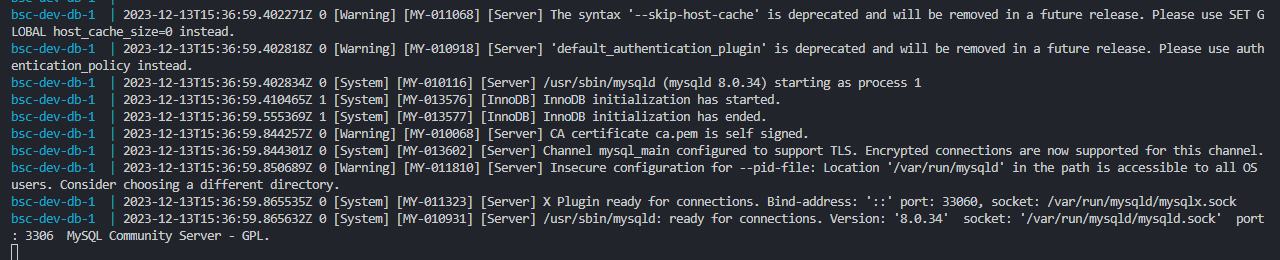
## Futtatási lépések:

A kicsomagolást követően a wireless\_tools\_analytics mappában található az alkalmazás forráskódjai. Három fontos dologgal kell rendelkeznie a felhasználónak:

* Node.js futtatási környezet
  + <https://nodejs.org/en/download>
* Npm package manager
  + <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>
* Futó docker daemon
  + Docker-Desktop
    - <https://www.docker.com/products/docker-desktop/>
  + Rancher Desktop , ha gond a liszenszelés
    - <https://rancherdesktop.io/>

### Leírás:

A futtatás egy többlépéses parancssor eredménye lesz: TODO KÉPEK

1. Indítsuk el a docker-container-t
   1. Nyissunk meg egy terminált
   2. Győződjünk meg, hogy fut a docker daemon
   3. Navigáljunk a db mappába
   4. Adjuk ki a ’docker-compose up’ parancsot

. ábra: Sikeres docker-compose utolsó üzenetei

1. Generáljuk le a segéd táblákat
   1. Sikeres docker container indítás után nyissunk egy új terminált ( az előzőt hagyjuk megnyitva!)
   2. Adjuk ki a `docker exec -it bsc-dev-db-1 bash -l` parancsot
   3. Ez nyit egy új terminált a korrábin belül, itt adjuk ki a `mysql -u root -p "bsc-dev-db" < runtime-scripts/helper-tables.sql` parancsot
   4. Sikeres futás esetén zárjuk be ezt a terminált
2. Fordítsuk le és futtassuk a szerver kódot
   1. Navigáljunk a server mappába
   2. Győződjünk meg róla, hogy rendelkezünk node és npm parancsokkal
   3. Adjuk ki az `npm ci` parancsot
   4. Adjuk ki az `npm run start` parancsot , ez elvégzi mindkét lépést
   5. Siker esetén hagyjuk nyitva ezt a terminált is
3. Futtasuk a kliens kódot
   1. Navigáljuk a client/tool\_analytics\_frontend mappába
   2. Adjuk ki az `npm ci` parancsot
   3. Adjuk ki az `npm run dev` parancsot
   4. Siker esetén ezt a terminált is hagyjuk nyitva

### run-script.js

Alternatívaként, készítettem egy extra modult, ami szinkronban elvégzi ezeket az utasításokat. Szükséges a fent említett három dependencia ebben az esetben is.

Használatához csak lépjünk be a fő mappába egy terminálból és adjuk ki a `node runtime-script.js` nevű parancsot. A program tájékoztatni fog éppen melyik komponensen dolgozik és hogy ez milyen eredménnyel zárult, ha valami rész sikertelenül zárul, az egész leáll. Felszabadítja maga után a portokat, egy terminált foglal el és egyedül a docker container-t nem törli, azt manuálisan lehet meg tenni ha szükséges.

### Extra tudni valók:

A db mappa tartalmaz egy markdown file-t ahol a docker-specifikus extra információk találhatóak.

## Felhasználói felület:

A felhasználói felület kevés interaktív tartalommal bír mivel az adatok kimutatása a fő cél. Azonban az alábbi lehetőségek adottak:

### Navigálás

Total page

Per Tool page

### Összehasonlítások

# Fejlesztői dokumentáció

## Leírás

Az Wireless Tools Analytics egy komplex (***háromrétegű kliens-szerver*** architektúrájú) rendszer, amely adatbázis-kezelést használva lehetővé teszi az adatok hatékony és strukturált kezelését. Az alkalmazás széles körű lehetőségeket biztosít az adatok lekérdezésére, feldolgozására és megjelenítésére különböző grafikonokon keresztül.

A tervezés során számos fontos kérdés merült fel a megvalósítást illetően. Ezek külön fel vannak tüntetve a rétegek részletes jellemzésénél. Fontos szempontnak találtam a könnyű és lokális kezelhetőséget ezért az alkalmazás kiegészült egy futtatást segítő szkrip-tel is. Ez segíti az egyébként aszinkronikus parancsok szinkronizált végrehajtását.

Az adatbázist egy lokális docker konténer fogja képezni, mivel így külsős service-k nélkül , offline is teljeskörű funkcionalitást érhetünk el. Emellett az adatbázis állapota is könnyen verziókezelhetővé vált inicializáló szkriptek segítségivel. Biztosítva a konzisztenciát különböző platformok és futtatási környezetek között.

## Fejlesztési terv- és folyamatok:

A fejlesztést három fázisra osztottam + verzió kezel

### Technológiák

### Megvalósítás

## Projektstruktúra leírása:

## Komponensek:

### Adatbázisrétegek

Az adatbázis réteg két fő részre bontható:

* **Első réteg:**A fő relációs adatbázisunk, ezek tartalmaznak minden beérkezett eseményt aminek a naplózásából a kimutatást végezzük, plusz a hozzájuk tartozó részletes információkat és azok kapcsolótábláit.  
  Ezek inicializálása létrehozása a docker konténer létrehozásakor automatikusan megtörténik.Ehhez a ***db/mock/data*** folder tartalmazza a szükséges sql parancsokat.  
    
  A réteg további 3-ra bontható:
  + EventLog tábla: Ez a fő táblánk, itt történik a naplózás minden esemény kiváltásakor.Megtudhatjuk belőle az esemény idejét, milyen parancsot és melyik eszközből kapott. Emelett refernciákat is tárol a felhasználó, lokáció és szoftver adatokról.
  + Információs táblák (User,Location,Software): Ezekben tároljuk a számontartott felhasználókat, lokációkat és szoftver információkat.
  + Kapcsoló táblák: Összekapcsolják a hozzájuk tartozó információs tábla egy elemét egy use session-el(StudioInstance). Ezen párok kerülnek referenciaként egy eseményhez.
* **Második réteg:**   
  A segéd táblák , ezek szolgálnak a kimutatásokhoz tartozó csoportosított lekérdezések ereményeivel. Erre azért volt szükség mivel az éles adatbázis nagyon nagymennyiségű adatot tárol, amik lekérdezése időbe telik és ha ez direkt a kliens kérésére történne az rontaná a reszponzivitását. Ehhez mountoljuk a db/mock/runtime-scripts mappa tartalmát, amit a konténer konzoljából futtathatunk a létrehozását követően.

### Adatfeldolgozás és szolgáltatás

Az alkalmazás lehetőséget biztosít az adatok dinamikus lekérdezésére a segéd táblákból és kisebb direkt lekérdezésekből. Főbb feladatai:

* **Kommunikáció az adatbázissal: a kapcsolat létrehozása , fenttartása**
* **Adatok feldolgozása: a tábla lekérdezések eredményeinek feldolgozása a megjelenítésnek alkalmas módon**
* **API biztosítása a kliens számára: ahol a feldolgozott adatokat lekérdezhetjük a megjelenítés oldalán**

Emelett session szinten saját magának is tárol segéd táblákat amik a feldolgozást és a címkézést segítik. Pl.: Évek , éppen használt eszközök , stb…

### Felhasználói interfész

Az alkalmazás egyszerű és felhasználóbarát felhasználói felülettel rendelkezik, amely lehetővé teszi a felhasználók számára az adatok intuitív és könnyű kezelését, valamint a kívánt grafikonok kiválasztását és testreszabását az adatok megfelelő megjelenítéséhez.

## Futtatási lépések fejlesztőknek:

## Adatbiztonság és Teljesítmény

Az alkalmazás maximális prioritást ad az adatbiztonságnak és az adatintegritásnak, miközben magas teljesítményt és gyors adatelérést biztosít a háromrétegű adatbázis struktúrájának hatékony kihasználásával.

## Összegzés

Az X alkalmazás lehetővé teszi a felhasználók számára, hogy könnyen és hatékonyan navigáljanak a háromrétegű adatbázisban tárolt információk között. A grafikonok változatos megjelenési lehetőségei révén az adatok vizualizációja segíti a felhasználókat a döntéshozatalban és az adatelemzés során.

# Összefoglalás és további fejlesztési lehetőségek

# Irodalomjegyzék

# Melléklet