

Követelményspecifikáció

GitHub repository statisztika vizualizáció

1. Feladatkiírás

A rendszer egy nyilvános webalkalmazás, amely GitHub repository-k statisztikai adatait tudja kinyerni vagy GitHub API-ról vagy a repository tartalmából. A hozzáféréshez nyilvános repository alapján nincs szükség, csak egy URL-re. Privát repository esetén egy GitHub-on készített access token megadás szükséges. A repository statisztikáit szövegesen és grafikusán mutatja az alkalmazás, amelyek közül pár példa: commitok száma, commitok időbeli eloszlása, commitok szerzőinek hozzájárulása, sorok számának alakulása, stb.

2. Fejlesztői csapat

Csapattag neve	NEPTUN-kód	E-mail cím
Németh Marcell	F2AVXH	nmarci2008@gmail.com
Morvai Dániel	JG6L5H	morvai.daniel96@gmail.com

A csapat tagjai között a feladatmegosztást a feladat technikai jellege miatt első iterációban front-end / back-end alapon tervezzük megvalósítani.

3. Részletes feladatléírás

A projekt során célunk egy olyan webalkalmazás készítése, amely képes egy felhasználó által választott GitHub repository-ról statisztikákat készíteni és vizualizálni. Az applikáció, mint kvázi BI eszköz, segítségével hatékonyabbá tehető a fejlesztői csapatok/egyének teljesítményének menedzselése, illetve esetleges javítása.

A kiválasztott repository életciklusa a webalkalmazás futása során a következő:

- Repository (továbbiakban tároló) URL kézi megadása (a kulcs validálása nem feladata a programnak, a felelősség a felhasználót terheli)
- A tároló felvételre kerül egy helyi metadata adatbázisba, amely egyfajta memóriaként, cache-ként működik egy adott időkereten belüli többszörös lekérdezések adatforgalom csökkentése érdekében.
- A feldolgozó motor (back-end) beolvassa a tároló tartalmát egy lokális memóriába és elkészíti a leíró statisztikákat, amelyek a következők:

- Kommitok száma: fejlesztők hozzájárulása a kódbázishoz
- Legtöbb committal rendelkező fejlesztő és egy másik repositoryba legtöbbet commitoló fejlesztő összehasonlítása
- Két repository fejlődésének üteme: hozzáadott commitok számának növekedése / időszak
- Kommitok időbeli eloszlása: év, hét, óra
- Összes kód hozzáadás, törlés száma: év, hét
- Statisztikák eltárolása az adatbázisban
- A tároló feldolgozásának időigényessége miatt, a kliens aszinkron vár a szerver válaszára az eredmény pollozásával, a lekérdezés idejétől a válasz megérkezéséig.
- Statisztikák megjelenítése, amelyek a következő vizualizációkat jelenti:
 - Hisztogram az egyes fejlesztők kommit számainak eloszlásáról
 - Pie-chart (kördiagram) a kód hozzáadás, törlés megoszlásáról
 - Line-chart (vonaldiagram) a kommitszámok időbeli változásáról

4. Technikai paraméterek

A definiált alkalmazást architektúrájából adódóan két részre osztottuk: back-end és front-end. A szerver feladata az üzleti logika megvalósítása: kapott URL alapján adatok biztosítása a webes kliens felé, amely ezeket megjeleníti. A két réteg egymással REST API hívásokkal kommunikál.

A back-end motort Spring Boot technológiával, Kotlin nyelven készítjük el annak érdekében, hogy több operációs rendszeren is lehessen használni. Az alkalmazás meta-adat adatbázisát egy Oracle SQL vagy hasonló adatbázis fogja tárolni.

A front-end megvalósításához az Angular ketrendszert fogjuk használni.

5. Szótár

Entitás	Magyarázat
GitHub	Nyilvánosan elérhető kódbázis, amely fejlesztői csapatok kódmenedzsmentjét segíti
Repository	Tároló alkönyvtár a GitHub kódbázisban, amely különálló projektek forráskódját tárolja
BI eszköz	Adatok beolvasását, tárolását, feldolgozását, elemzését megvalósító eszköz

Github access token	Github felhasználók által készíthető karaktersorozat, amelyből információk nyerhetők ki a felhasználó tárolójáról.
Metadata adatbázis	Az alkalmazás adatbázisa melyben tárol cachelni kívánt adatokat a gyorsabb kiszolgálás végett.
Front-end	A webes kliens program, amely kérésekkel fordul a szerverhez.
Back-end	Az alkalmazás szerver, amely egy távoli gépen fut és kliensek kéréseit képes feldolgozni.
Polling	A kliens program folytonos szinkron kérdése a szerver felé.
Commit	Tárolóba a kód jelenlegi állapotának rögzítése adott felhasználó által.

6. Fontosabb use-case diagrammok

