|  |  |  |
| --- | --- | --- |
|  | **Eötvös Loránd Tudományegyetem**  Informatikai Kar  Programozási Nyelvek és Fordítóprogramok Tanszék |  |

**P4 programok gráf alapú statikus elemzése**

|  |  |
| --- | --- |
| *Témavezető:*  Tóth Gabriella  Doktorandusz | *Szerző:*  Tábi Zoltán  Programtervező Informatikus, BSc |

Budapest, 2020

Tartalomjegyzék

[1. Bevezetés 4](#_Toc41345840)

[2. Felhasználói dokumentáció 5](#_Toc41345841)

[2.1. Program célja 5](#_Toc41345842)

[2.2. Használat feltételei (TODO elbírálás, hogy szükséges-e) 5](#_Toc41345843)

[2.3. Telepítés és futtatás 6](#_Toc41345844)

[2.4. Rendszer ismertetése 7](#_Toc41345845)

[2.4.1. Kezdő oldal 7](#_Toc41345846)

[3. Fejlesztői dokumentáció 9](#_Toc41345847)

[4. Összefoglalás 10](#_Toc41345848)

[5. További fejlesztések 11](#_Toc41345849)

[5.1. P4 fordítóprogram beépítése 11](#_Toc41345850)

[5.1.1. Egész projekt Linux-on történő futtatása 11](#_Toc41345851)

[5.1.2. Szerver a P4 fordítóprogramhoz 11](#_Toc41345852)

[5.2. Vizsgált résznyelv kiterjesztése 11](#_Toc41345853)

[5.3. Felhasználó azonosítása 12](#_Toc41345854)

[5.4. Gráf elemzés összesített verziója (TODO lehet kikerül) 12](#_Toc41345855)

[5.5. Felület optimalizálás 13](#_Toc41345856)

[5.5.1. Sötét téma 13](#_Toc41345857)

[5.5.2. Mobil használat 13](#_Toc41345858)

# Bevezetés

Szakdolgozatom célja egy olyan felhasználó barát weboldal elkészítése, ahol

# Felhasználói dokumentáció

## Program célja

A feladatom egy olyan felhasználóbarát és könnyen kezelhető weboldal elkészítése volt, amely P4-es programok gráf alapú elemzését végzi. A programkód bevitelét meg lehet tenni szövegbeviteli mező segítségével vagy fájlfeltöltés útján. A gráfokkal interaktív események hajthatóak végre. Többek között gráf bejárás szimulálása, csúcshoz és csúcsból vezető út kirajzolása és gráfok egymás közötti kapcsolatának megjelenítése. Ezek után meg lehet adni a kezdeti- és végértékeket, majd több megadása után csomagot összeállítani és végezetül az elemzett gráfok és eredmények megtekintését.

A weboldal főleg a P4 programnyelvvel foglalkozó közösség számára szól, de bárki számára használható az oldalon megtalálható fájlok segítségével, valamint a dokumentáció vagy a webhelyen található Súgó oldal átolvasásával.

## Használat feltételei (TODO elbírálás, hogy szükséges-e)

Egy weboldal lévén a felhasználásnak nincsenek rendszert érintő követelményei. A weboldal használható mobileszközről, tabletről, valamint számítógépről is, ezek közül ajánlott a számítógép használata nagyobb képernyő és átláthatóság érdekében. Valamint az alapvető webböngészés képességével is rendelkeznie kell a felhasználónak.

Támogatott böngészők:

* Google Chrome
* Mozilla Firefox
* Opera
* Microsoft Edge

## Telepítés és futtatás

A telepítés és futtatás menete:

* Node.js 12.16.1+ verzió letöltése és telepítése innen: <https://nodejs.org/>
* Visual Studio Community 2019 letöltése és telepítése innen: <https://visualstudio.microsoft.com/vs/>
* P4Analyst.sln fájl megnyitása a Visual Studio segítségével
* A solution explorer menü sávban az AngularApp projektre jobb klikkel kattintani és a Debug menü pontban a „Start New Instance” feliratra kattantani. Ekkor a projekt elindul és akár pár percbe is telhet mire a weboldal megjelenik és betöltődik, ha betöltés közben hiba lép fel, akkor az F5 billentyű segítségével rá kell frissíteni az oldalra.

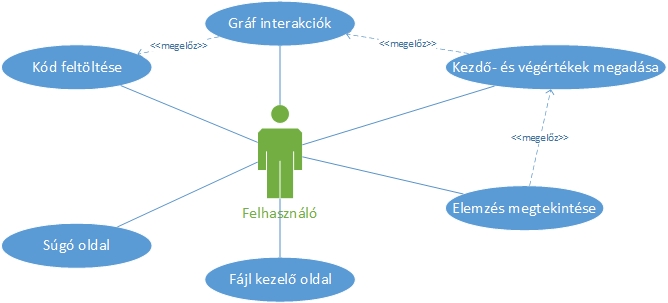
Azonban telepítésre nem feltétlenül van szükség, mivel az alábbi címen elérhető a weboldal: <https://p4analyst.azurewebsites.net/>

Ajánlott a számítógép használata nagyobb képernyő és átláthatóság érdekében.

Támogatott böngészők:

* Google Chrome
* Mozilla Firefox
* Opera
* Microsoft Edge

## Rendszer ismertetése

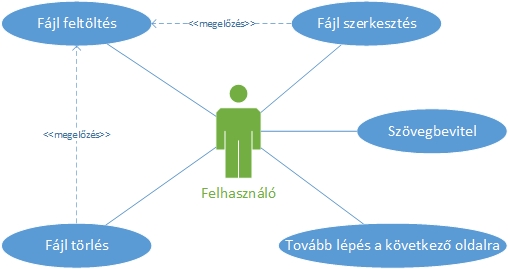


1. ábra – Általános felhasználói eset diagram

Az *általános felhasználói eset diagram* ábrán láthatóak a weboldal fő funkciói. Ez csak egy összefoglaló ábra, melynek funkcióit bővebben kifejtem később, amikor is a weboldal összes oldalát és azokban rejlő lehetőségeket részletesen leírom.

Az ábrán jól látható, hogy a weboldal egy része lineáris lefutással bír, vagyis egymásra épülő, ki nem kerülhető folyamatok követik egymást.

### Kezdő oldal



. ábra - Kezdő oldal felhasználói eset diagram

A *Kezdő oldal felhasználói eset diagram* ábráján látható, hogy a felhasználónak milyen funkciók és lehetőségek állnak rendelkezésére, amikor a kezdőoldalon van.

Ezen az oldalon adhatja meg elemzendő kódját, itt egy tab-os elrendezésben választhatja ki, hogy milyen módon adja át kódját. Az opciók kizárják egymást, szóval, ha feltöltött fájlt, addig nem léphet át a szövegbeviteli részre, amíg azt a fájlt ki nem törölte és fordítva is igaz, tehát, amennyiben a beviteli mező nem üres úgy nem léphet vissza a fájl feltöltő részlegre. A következő oldalra lépés, akkor érhető el, ha az egyik eset által adott meg szöveget. A szövegbeviteli mező egy egyszerű beviteli mező, ahova szöveget írhat be vagy illeszthet be a vágólapról.

#### Fájl feltöltés

A felhasználó ebben az esetben két különböző úton adhatja meg a kiválasztandó fájlt. Az első eset, hogy bal klikkelés után a fájlkezelő ablakból kiválasztja a feltöltendő fájlt. A második eset, hogy úgynevezett *drag and drop* módszerrel, vagyis a fájl odahúzásával és elengedésével adja meg a kiválasztott fájlt. Minden esetben ellenőrizzük, hogy a fájl megfelelő, .p4 vagy .txt kiterjesztésű-e, más kiterjesztéssel rendelkező fájl feltöltésére nincs lehetőség, ezt figyelmeztető üzenet formájában jelezzük a felhasználónak. A második módszer esetén a feltöltött fájlok mennyiségére is ellenőrzést hajtunk végre, mivel több fájl feltöltése nem megengedett, ekkor is megtagadjuk a fájl(ok) feltöltését és figyelmeztető üzenet formájában ezt jelezzük.

Az elfogadott fájl megadása után a képernyőn megjelenik a fájl neve, valamint mellette kettő akció gomb is. Az első esetben a fájl szerkesztését végezhetjük, ekkor egy felugró ablakban jelenik meg a fájl tartalma, ahol szerkeszthetjük. Az ablak bezárása négy módon történhet meg:

* Mentés gombra kattintás – a szerkesztést jóváhagyjuk, ezt egy felugró üzenettel erősítjük meg
* Mégsem gombra kattintás
* Bezáró gombra kattintás
* Az ablak területén kívülre kattintás

A törlés gombra való kattintás esetén a fájl feltöltése semmisnek tekintendő és lehetőség nyílik másik fájlt feltölteni, vagy a szövegbeviteli mezőre átlépni.

### Gráf megjelenítő oldal

# Fejlesztői dokumentáció

# Összefoglalás

# További fejlesztések

## P4 fordítóprogram beépítése

A vizsgált nyelv kibővítésének és a megfelelő hibajelzéseknek előfeltétele, hogy az elemzendő kód helyességét a P4 fordító segítségével ellenőrizzük. Mivel a P4 fordítót Linux operációs rendszeren lehet használni és az alkalmazás jelenleg Azure-ra van kitelepítve, ezért ennek beépítésére jelenleg két módszert látok megvalósíthatónak.

### Egész projekt Linux-on történő futtatása

Az első megoldás, hogy létrehozunk egy Linux operációs rendszerrel rendelkező virtuális gépet, amelyre telepítjük a P4 fordítóját és az elemzett programot lefuttatjuk rajta. A futtatást C#-ból meg lehet tenni parancssori parancsok futtatásával. Ezt a megoldást nem ajánlom, mivel így fejlesztői környezetben is ki kell alakítani egy virtuális gépet, ahol tesztelhetők a módosítások.

### Szerver a P4 fordítóprogramhoz

A második megoldás, hasonló az elsőhöz, de itt csak egy szerver applikációt telepítünk a virtuális gépre, aminek egy belépési pontja van, amit meg lehet hívni a futtatandó kóddal és visszatér a program helyességével. Ehhez a megoldáshoz is kell fejlesztői környezetet kialakítani, de itt csak egy applikációt kell tesztelni rajta és a jövőben kevés változtatást kell eszközölni rajta, így az új fejlesztőknek nem feltétlenül van szükség a környezet kialakítására. Így én ezt a megoldást javaslom.

## Vizsgált résznyelv kiterjesztése

A vizsgálat során kikötésekkel éltem, hogy milyen programkódot tud feldolgozni az elemző programok, ezen kikötések feloldása lenne a feladat. TODO FELOROLNI

## Felhasználó azonosítása

A felhasználó regisztrálhasson vagy belépjen meglévő fiókjába. A felhasználóbarát megközelítés miatt legyen lehetőség *Google, Facebook* vagy *Github* felhasználóval belépni. A felhasználó azonosítása nagyon sok lehetőséget von maga után. Lehetőség nyílik, hogy a felhasználó feltöltsön kódot és eltárolja saját maga, megjelölt másik felhasználók, vagy az egész közösség számára. Legyen lehetőség egymás kódjaira reagálni, ötleteket, megjegyzéseket adni, vagy kedvencnek jelölni, ezzel később is gyorsan megtalálható. A fájl kereső oldalon így bővíthetőek a keresési funkciók felhasználó és kedveltségi szint alapján.

A már feltöltött kód módosítását és előzmények megtekintését is meg lehet csinálni, valamint elmenteni a kód elemzéseit, hogy a felhasználó nyomon tudja követni, hogy mennyit fejlődött kódjának optimalizáltsága. Amennyiben a programba beépítésre kerül a [P4 fordítója](#_P4_fordítóprogram_beépítése), akkor a felhasználó tudja fordítani programját, ezzel ellenőrizve helyességét. Ezen funkciók segítségével, akár egy fejélesztői környezet kialakítása is eszközölhető lenne.

Projektek létrehozása, ahol egy vagy több felhasználó képes módosítani a kódot.

## Gráf elemzés összesített verziója (TODO lehet kikerül)

Az elemzés megtekintésénél jelenleg minden program lefutásnak, amit a különböző kezdeti értékek segítségével ad meg a felhasználó külön-külön kirajzoljuk a gráfot, de nincs egy olyan összesítő gráf, amely bemutatná a leggyakrabban használt kódrészleteket. Ennek megalkotására az adatok már rendelkezésre állnak, a különböző gráfokban szerepel, hogy egy rész hányszor lett használva, ezeket összesíteni kell és ezek alapján beállítani a csúcsok színezését.

## Felület optimalizálás

### Sötét téma

A felületen sötét téma került kialakításra, amely megfelelő kontrasztot és láthatóságot nyújt egy eset kivételével. A gráf éleinek színe alapértelmezetten fekete, ez a szín sötét téma esetén a sötét szürkén látható, de a szemnek fárasztó fókuszálni rá, ezért ezt módosítani kell. A gráfot teljesen újra kell építeni egy másik a sötét szürkével kontrasztban álló élszínezéssel.

### Mobil használat

A felület jól használható mobilos környezetben, de vannak szépítendő és a felhasználói élményt javító változtatási lehetőségek.

Mobil optimalizáció:

* Az oldalsáv menü megjelenítése elhúzással. Ezzel könnyítve a felhasználót a navigációs menü megnyitásában, hogy ne kelljen mindig a megnyitó gombot lenyomni. Fejlesztéskor figyelni kell, hogy az animációs gombot használjuk.
* Gráf megjelenítő oldalon a tab-os elrendezésben elhúzással (swipe) is tudjon oldalt váltani.
* A megjelenített gráfok mozgatásának és nagyításának megvalósítása. A felhasználó a gráfra nyomással nyisson meg egy felugró ablakot, ahol ezek a funkciók aktívak. Elegendő lenne csak a nagyítás funkció aktiválása, azonban a használt gráf ábrázoló könyvtár ezt a megoldást nem támogatja.
* A gráf valamiért nem a képernyő közepén jelenik meg és az újra méretezhetőség sem működik rajta teljes értékűen. Ezen esetekre nem tudtam megoldási ötletet találni.