

2. Követelmény, projekt, funkcionalitás

54 – *Override*

Konzulens:

dr. László Zoltán

Csapattagok:

| | | |
|-----------------|--------|----------------------------|
| Kriván Bálint | CBVOEN | balint@krivan.hu |
| Jákli Gábor | ONZ5G1 | j_gab666@hotmail.com |
| Dévényi Attila | L1YRH0 | devenyat@gmail.com |
| Apagyi Gábor | X8SG3T | apagyi.gabooo@gmail.com |
| Péter Tamás Pál | N5ZLEG | falconsaglevlist@gmail.com |

2011. február 19.

Tartalomjegyzék

| | | |
|----------|--|----------|
| 2 | Követelmény, projekt, funkcionalitás | 4 |
| 2.1. | Követelmény definíció | 4 |
| 2.1.1. | A program célja, alapvető feladata | 4 |
| 2.1.2. | A fejlesztőkörnyezet | 4 |
| 2.1.3. | A futtatáshoz szükséges környezet | 4 |
| 2.1.4. | A felhasználói felület | 4 |
| 2.1.5. | Minőségi tényezők | 4 |
| 2.1.6. | A software minősítése | 4 |
| 2.1.7. | A kibocsátás | 5 |
| 2.2. | Projekt terv | 5 |
| 2.2.1. | A fejlesztői csapat | 5 |
| 2.2.2. | Életciklus modell | 5 |
| 2.2.3. | Szervezési struktúra | 5 |
| 2.2.4. | Fejlesztési ütemterv | 6 |
| 2.2.5. | Határidők | 7 |
| 2.3. | Feladatleírás | 7 |
| 2.4. | Szótár | 8 |
| 2.5. | Essential use-case-ek | 10 |
| 2.5.1. | Use-case diagram | 10 |
| 2.5.2. | Use-case leírások | 10 |
| 2.6. | Napló | 11 |

Ábrák jegyzéke

| | |
|---|----|
| 2.1. MSN csoport a csapatnak | 6 |
| 2.2. Git történet | 6 |
| 2.3. Ticketek | 6 |
| 2.4. Essential use-case diagram | 10 |

2. Követelmény, projekt, funkcionalitás

2.1. Követelmény definíció

2.1.1. A program célja, alapvető feladata

Az általunk kifejlesztett program célja egy előre megadott digitális áramkör szimulációja és annak megjelenítése, grafikus mindenki számára könnyen kezelhető, átlátható formában. Az alkalmazás az áramköri elemekből felépített digitális hálózat működését szemlélteti úgy, hogy felhasználói interakciók során a rendszer bemenetei átkonfigurálhatóak.

2.1.2. A fejlesztőkörnyezet

A fejlesztéshez NetBeans 6.9.1 szoftvert választottuk. Az UML diagramok elkészítéséhez a Visual Paradigm for UML nevű alkalmazást használjuk, mely képes az osztály-diagramból Java forráskódot generálni és vice versa. Fejlesztés során szem előtt tartjuk, hogy a program kompatibilis legyen az Oracle által gondozott Java 1.6-os verziójával. Természetesen a cél az, hogy a digitális áramkört modellező program a Hallgatói Számítógép Központban rendszeresített JDK és JRE alatt fordítható és futtatható legyen. A dokumentumokat \LaTeX -hel készítjük el a Texmaker nevű alkalmazás segítségével, melyet PDF-be fordítunk le. A unit-tesztekre a JUnit csomagot fogjuk használni.

Mivel a fentebb felsoroltak közül mindegyik alkalmazás fut mind Windows, mind Linux operációs rendszeren, így az egész fejlesztés mindkét platformon megvalósítható.

2.1.3. A futtatáshoz szükséges környezet

Java Runtime Environment 1.6-os verziója, illetve az a számítógép, mely ezt futtatni képes. A modellező alkalmazás használatához billentyűzet, grafikus képernyő és egér szükséges.

2.1.4. A felhasználói felület

A program végső változata grafikus felhasználói felülettel rendelkezik. A programot a felhasználó az egér és a billentyűzet segítségével vezérelheti.

2.1.5. Minőségi tényezők

Teljesítmény: A cél az, hogy a digitális rendszermodellező szoftver használható legyen a fentebb meghatározott minimális rendszeren. A grafikus felületnél törekedni fogunk a folyamatos szimuláció megjelenítése.

Újrafelhasználhatóság: A cél az, hogy a grafikus felhasználói felületet a program többi részétől teljesen különválasszuk, így lehetővé téve azt, hogy később a grafikus felület egyszerűen és gyorsan változtatható legyen.

Rugalmasság: A rugalmasságot a fejlesztőkörnyezet biztosítja, a modellező szoftvernek ugyanis minden olyan környezetben futtathatónak kell lennie, melyben létezik megfelelő Java futtatókörnyezet.

Felhasználhatóság: A használat különösebb tanítást nem igényel, alapfokú számítástechnikai tudással akár a felhasználói kézikönyv elolvasása nélkül is használható.

2.1.6. A software minősítése

A kifejlesztett software akkor megfelelő, ha minél pontosabban megegyezik a fentebb leírtakkal. Ezt ellenőrizni lehet a program futtatásával és kipróbálásával, illetve a forráskód és a modell összevetésével, valamint a funkcionális tesztek futtatásával.

2.1.7. A kibocsátás

A program kibocsátása először a forráskóddal együtt a konzulens felé fog történni.

2.2. Projekt terv

2.2.1. A fejlesztői csapat

| Csapattag neve | feladatköre |
|-----------------|--|
| Kriván Bálint | csapatvezető, kód, dokumentáció, szervezés |
| Jákli Gábor | kód, dokumentáció, ticket-koordinátor |
| Dévényi Attila | kód, dokumentáció, GUI-felelős |
| Apagyi Gábor | kód, dokumentáció |
| Péter Tamás Pál | kód, dokumentáció |

2.2.2. Életciklus modell

A feladat először a program megtervezése, mely a dinamikus és objektum modelleket foglalja magába. Ha ez készen van, elkezdhető a skeleton implementálása. Ez a lépés már teljesen meghatározott, nem merülhet fel semmilyen komplikáció, ha a modellek megfelelőek voltak. A következő feladat a prototípus elkészítése. A programnak ebben az állapotban könnyen tesztelhetőnek kell lennie, hogy a programozási és funkcionalitási logikai hibák könnyen felismerhetők legyenek. Ha már a prototípus is megfelelő, akkor kezdődhet a grafikus felület megvalósítása. Itt is fontos a tesztelés és a kiértékelés, mert a jó megjelenés sokat számít a modellező használhatóságában. Ha ennek kifejeztése is sikeres, készen van a program első teljes változata. A kötelező feladat csak eddig tart. Ezt a változatot kell leadni a dokumentációval és a forráskóddal együtt.

2.2.3. Szervezési struktúra

A csapat öt emberből áll. A feladat szempontjából a tudásunk nem azonos, mindenki más-más területet érez a magáénak, illetve a feladat eltérő részeinek megoldásához van nagyobb kedvünk. Azt a felépítést választottuk, hogy mindenki az érdeklődésének és tudásának legmegfelelőbb részt kapja az egész feladatból. A feladatok szétosztását találkozókon, illetve az alább meghatározott kommunikációs csatornákon egyeztetjük, ahol az egyéni kívánságok mellett ügyelünk arra, hogy minden feladat kiosztásra kerüljön, valamint a csapattagok az egész feladat megoldásából nagyjából egyenlő mértékben vegyék ki a részüket. A találkozók keretében, mivel a szétosztott feladatok nagy mértékben függenek egymástól, javaslatokat teszünk egymásnak a feladat megoldásának körülményeit és a határidőt illetően.

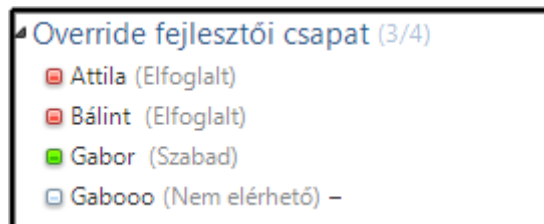
A forráskódot és minden a fejlesztés során elkészülő dokumentációt, illetve a projekthez tartozó egyéb fájlokat megegyezés alapján egy Git központi tárolóban tároljuk, melyhez a Codaset (<http://codaset.com>) nevű ingyenes szolgáltatását használjuk és erről mindenki egy saját klónt készít.

A kiosztott feladatokat a tulajdonosuk elvégzi a megbeszélt határidőig, de ha ez megváltozott funkcionalitást takar, akkor az adott csapattag köteles a megfelelő teszteseteket megírni, és azok sikeres lefutásáról meggyőződni. Abban az esetben, ha az alkalmazás nem fordul le, vagy valamelyik teszteset nem fut le sikeresen, az adott commit visszaállításra kerülhet annak kijavításáig, melyet a ticket-rendszerben jelezzük a másik felé.

Hogy a fejlesztés minél hatékonyabb és zökkenőmentesebb legyen, a következő eszközöket, technológiákat alkalmazzuk:

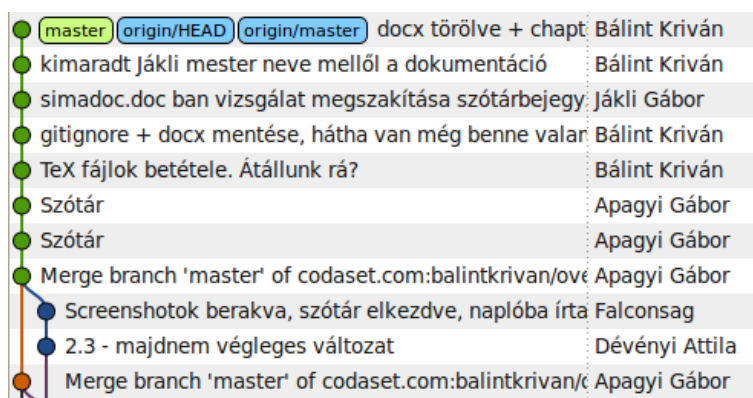
E-mail Az egymás számára fontos anyagokat, melyeket a találkozókön előzetesen megbeszéltünk, levélben küldjük el.

Msn Felvettük egymást a Microsoft Messenger-be, hogy szükség esetén egymástól is segítséget tudjunk kérni kisebb technikai problémák megoldásában. Természetesen ezek a feladat lényegét, a projektről hozott döntéseket nem érinthetik, de kivételes helyzetben akár az Interneten is tarthatunk találkozót.



2.1. ábra. MSN csoport a csapatnak

Git tároló A feladatok megoldása közben keletkezett anyagokat egy – kizárólag a csapat tagjai által hozzáférhető – helyen tároljuk (lásd fentebb). Így mindig elérhető a fejlesztések legfrissebb változata.



2.2. ábra. Git történet

Ticket-rendszer A fejlesztés során felmerülő problémákat, kérdéseket ticket formájában megírjuk egymásnak, amit később a kijelölt felelős személy megold, ha szükséges, akkor együtt konzultálunk a megoldás módjáról, menetéről.

| | SUMMARY | STATE | MILESTONE | ASSIGNEE | UPDATED ▼ |
|----|--|-------|---------------------------|-----------|--|
| #7 | Szervezési struktúra kialakítása created by falconsag 1 day ago | NEW | 2. Követelmény, projek... | gabooo | less than a minute ago by balintkrivan |
| #6 | Szótár elkészítése created by falconsag 1 day ago and has 1 comments | NEW | 2. Követelmény, projek... | falconsag | 1 minute ago by balintkrivan |
| #5 | Feladatleírás created by falconsag 1 day ago and has 1 comments | NEW | 2. Követelmény, projek... | devenyat | 1 minute ago by balintkrivan |
| #4 | Essential use-case-ek created by falconsag 1 day ago | NEW | 2. Követelmény, projek... | gabooo | 1 minute ago by balintkrivan |

2.3. ábra. Ticketek

2.2.4. Fejlesztési ütemterv

A program fejlesztésének három fő lépcsőfoka van. Ezek a következők:

Skeleton: A cél az, hogy mind a dinamikus, mind az objektum modell jól legyen kitalálva. Ha ezek elkészültek,

akkor a fejlesztés szempontjából sikeresen leraktuk az alapokat.

Prototípus: Ez már szinte a teljes változat, csak a grafikus felület elemei hiányoznak. Ez a változat tökéletesen megfelelő arra, hogy az objektumok, rutinok, függvények szemantikai helyességét vizsgáljuk.

Grafikus változat: A program teljes változata. Tulajdonképpen a prototípus a grafikus felülettel kiegészítve, esetleg kismértékben továbbfejlesztve.

2.2.5. Határidők

| | |
|-----------|--------------------------------------|
| febr. 11. | Csapat regisztráció |
| febr. 21. | Követelmény, projekt, funkcionalitás |
| febr. 28. | Analízis modell kidolgozása 1. |
| márc. 7. | Analízis modell kidolgozása 2. |
| márc. 14. | Szkeleton tervezése |
| márc. 21. | Szkeleton |
| márc. 28. | Prototípus koncepciója |
| ápr. 4. | Részletes tervek |
| ápr. 18. | Prototípus |
| ápr. 26. | Grafikus felület specifikációja |
| máj. 9. | Grafikus változat |
| máj. 13. | Összefoglalás |

2.3. Feladatleírás

Az általunk készített alkalmazás segítségével a felhasználó egy előre elkészített áramköri listából kiválasztott digitális áramkör szimulációját végezheti el grafikus megjelenítéssel. A program az alábbi alkatrészeket támogatja áramköri elemként: ÉS kapu, VAGY kapu, Inverter, Kijelző, Jelgenerátor, Kapcsoló, Vcc és Gnd. Ezek mindegyike egy vagy több ki- és/vagy bemenettel rendelkezik.

A komponensek (alkatrészek) részletezése:

1. **Általános komponensek:** A bemenet(ek)re érkező logikai érték(ek) alapján a kimenete(ke)n valamilyen logikai érték(ek)et produkálnak.

- Az ÉS kapu kettő vagy több bemenettel és egy kimenettel rendelkezik. A kimeneten a bemenetre kötött jelek logikai ÉS kapcsolata jelenik meg.
- A VAGY kapu kettő vagy több bemenettel és egy kimenettel rendelkezik. A kimeneten a bemenetre kötött jelek logikai VAGY kapcsolata jelenik meg.
- Az Inverter egyetlen kimenetén az egyetlen bemenetére kötött jel logikai negáltja jelenik meg.

2. **Megjelenítők:** Az ide tartozó elemek feladata a logikai értékek vizualizálása

- A Kijelző komponenssel a felhasználó a bemenetre kötött jelet vizuális formában tudja megjeleníteni.

3. **Jelforrások:** A harmadik csoport elemei melyeknek nincs bemenete csak kimenete, ez vagy előre definiált (gnd és vcc) vagy a felhasználó által módosítható (kapcsoló, jelgenerátor), ezáltal változtatva az áramkör működését.

- Jelgenerátor segítségével egy bitsorozatot tárolhatunk el, amelyet a szimuláció során az egyetlen kimenetén ciklikusan kiad.
- A Kapcsolónak egy kimenete van, melynek értéke a kapcsoló állásától függ. „Be” állásban igaz, „Ki” állásban hamis értékű.
- Gnd („föld”) konstans 0-át (logikai hamis) kiadó jelforrás.
- Vcc („tápfeszültség”) konstans 1-et (logikai igaz) kiadó jelforrás.

Az összes alkatrésze igaz, hogy nem lehet olyan bemenetük, amelyek szabadok, vagyis sehova sincsenek bekötve, ellenkező esetben a szimulációt nem lehet elindítani és a program figyelmezteti erre a felhasználót (ennek elkerülésére, a programmal szállított szimulálható áramkörök egyik komponensének sincs szabad bemenete).

A felhasználó a fentebb említett alkatrészekből összeállított digitális áramkör szimulációját végezheti. Az alkatrészek és azok egymáshoz kötései (összeköttetések) grafikus formában kerül megjelenítésre.

A szimuláció során bármelyik komponens pillanatnyi értékeit a felhasználó lekérdezheti az alkatrészeire való kattintással, ezzel egyidejűleg a szimulációt szünetelteti. Az áramkör vizsgálata közben a Kapcsolók értékeit szabadon változtathatja, melyek hatása valós időben megjelenik. Szimuláció elkezdésekor az összes áramköri elem kimenete hamis értéket vesz fel. Ha a vizsgálandó áramkör bizonyos idő alatt nem áll be stacionárius állapotba változatlan bemenetek mellett, akkor ez jelzésre kerül a felhasználó számára és a szimuláció automatikusan leáll. A szimuláció bármikor megszakítható majd újraindítható, illetve átválthat egy másik digitális áramkör vizsgálatára (az előre elkészített digitális áramkörök közül választva), amennyiben a jelenlegi áramkört nem kívánja tovább használni.

A szimuláció sebessége a felhasználó által konfigurálható, ezáltal a Jelgenerátor kimenetein kiadott jelek változásának sebessége változtatható.

A Kapcsolók, illetve a Jelgenerátorok gyors és egyszerű alap állapotba helyezése érdekében lehet törölni minden addig elvégzett beállítást egyetlen paranccsal, majd előlről kezdeni az egyes elemek konfigurálását. Lehetőség lesz továbbá a szerkeszthető jelforrások beállításainak elmentésére illetve későbbi visszatöltésére is. A konfiguráció sikeres betöltődéséhez teljesülnie kell annak a feltételnek, hogy abban az összes meghatározott elem szerepeljen az áramkörben, amire használni szeretnénk a beállításokat. Amennyiben ez nem áll fent, akkor a nem specifikált jelforrások alapállapotban lesznek. Előfordulhat még, hogy a konfigurációban olyan elemek szerepelnek, amelyek az áramkörünkben nem, ekkor hibaüzenet jelenik meg és a betöltés megszakad.

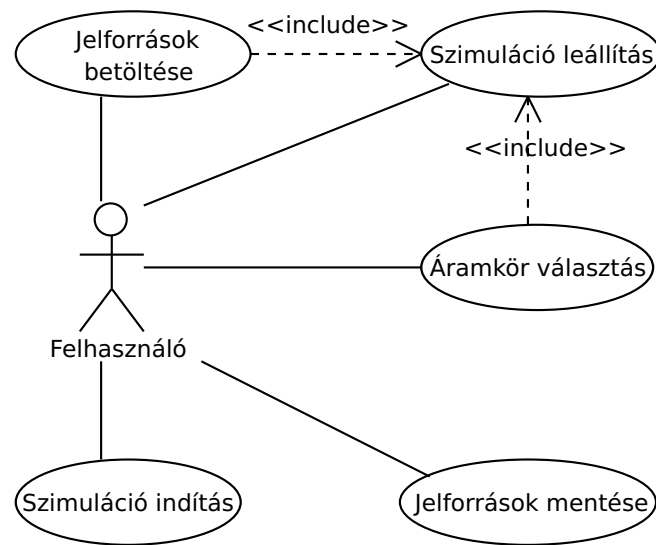
2.4. Szótár

| | |
|--------------------------------|--|
| Előre elkészített lista | Áramköröket tartalmazó előre elkészített gyűjtemény. |
| Áramkör | A komponensek egymáshoz kötéséből létrejövő rendszer. |
| Komponens | Az áramkör alapegysége, mely 3 fajtájú lehet: <i>általános komponens</i> , <i>megjelenítő</i> és <i>jelforrás</i> . |
| Alkatrész | komponens szinonimája |
| Általános komponens | Olyan komponens, mely a bemenet(ek)re érkező logikai érték(ek) alapján a kimenete(ke)n valamilyen logikai érték(ek)et produkál. |
| Bemenet | A komponensek olyan része, melyen keresztül logikai jeleket tudnak fogadni egy másik komponenstől és ezt valamilyen formában felhasználni. |
| Logikai jel | Az áramkörben levő komponensek által továbbított információ a többi komponens számára, mely az igaz illetve a hamis értéket veheti fel. |
| Igaz érték | A kétféle logikai jel egyike. (van amikor az '1'-es szimbólum jelöli) |
| Hamis érték | A kétféle logikai jel egyike. (van amikor a '0'-ás szimbólum jelöli) |
| Logikai negált | Az igaz érték logikai negáltja hamis, hamis értéké pedig igaz. |
| Kimenet | A komponens olyan része, melyen keresztül logikai jeleket tud továbbítani más komponenseknek. |
| Jel továbbítás | A logikai jel egyik komponenstől másik komponensig való áramlása. |
| Megjelenítő | Olyan komponens, mely a bemenet(ek)re érkező logikai jele(ke)t a felhasználó számára érzékelhető módon szemlélteti. |
| Jelforrás | Olyan komponens, mely bemenet nélkül továbbítja az áramkörben specifikált vagy a felhasználó által megadott jele(ke)t a kimenetén további komponens(ek) számára. |
| Gnd | Olyan komponens, mely konstans 0-át ad ki a kimenetén. |

| | |
|---|---|
| Vcc | Olyan komponens, mely konstans 1-et ad ki a kimenetén. |
| Komponensek egymáshoz kötése | Egy olyan folyamat, amely során 2 komponenst oly módon kapcsolunk össze, hogy az egyik komponens bemenetére a másik komponens kimenetének logikai jelét kapja meg. |
| Grafikus megjelenítés | Az áramkör felhasználó számára felfogható, érzékelhető megjelenítése. |
| ÉS kapu | <i>Általános komponens</i> , melynek a bemenetére érkező logikai jelek közt található hamis érték, akkor kimenetén hamis értéket, ellenkező esetben (vagyis minden bemenete logikai igaz) igaz értéket továbbít. |
| VAGY kapu | <i>Általános komponens</i> , melynek a bemenetére érkező logikai jelek közt található igaz érték, akkor kimenetén igaz értéket, ellenkező esetben (vagyis minden bemenete logikai hamis) hamis értéket továbbít. |
| Inverter | <i>Általános komponens</i> , mely a bemenetére érkező logikai jel negáltját továbbítja a kimenetén. |
| Kijelző | Egy darab logikai jelet váró <i>megjelenítő</i> , mely logikai igaz bemenet esetén világít (piros, kék vagy sárga színnel, melyet az áramkör leírója határoz meg), hamis esetén nem. |
| Áramkör leíró | Egy olyan szöveg, mely a program által elvárt módon leírja a teljes áramkört a komponensek és azok összeköttetéseinek definiálásával. |
| Jelgenerátor | Olyan <i>jelforrás</i> , mely előre megadott bitsorozatot ad ki ciklikusan a kimenetén. |
| Bitsorozat | Logikai jelekből létrejövő olyan sorozat, melynél az igaz értéket '1' szimbólum reprezentál, míg a hamis értéket '0'. |
| Kapcsoló | Olyan <i>jelforrás</i> , mely felhasználói interakció hatására kimenetén igaz, vagy hamis értéket továbbít. |
| Felhasználói interakció | Olyan esemény, melyet a felhasználó saját maga vált ki valamely tevékenysége során, ezzel potenciálisan befolyásolva az áramkör működését. |
| Szimuláció | Az a folyamat, mely során minden alkatrész kimenetének logikai jel értékét kiszámoljuk a bemenetére érkező logikai jelekből, vagy ha <i>megjelenítő</i> ről van szó, akkor a felhasználó számára a megjelenítő által meghatározott módon szemléltetjük a bemenetére érkező jeleket. Eközben a felhasználó által megadott időközönként (szimuláció sebessége) léptetjük a jelgenerátort. A szimuláció a felhasználó által indítható és megállítható. |
| Jelgenerátor léptetése | A jelgenerátorban tárolt bitsorozat következő elemére lépünk és azt adjuk ki a kimenetén, ha a végére értünk, akkor előlről indul. |
| Szimuláció sebessége | A jelgenerátor egyes állapotai közötti váltás sebessége. |
| Valós időben | A felhasználó számára lényegében érzékelhetetlen idő alatt. |
| Stacionárius | A rendszer egy stabil állapota, melyben olyan értékek jelennek meg a komponensek kimenetein, amelyek hatására (visszacsatolás esetén sem) változnak a rendszer komponenseinek kimeneti értékei (Jelgenerátor esetén nincs stacionárius állapot) |
| Szimuláció megszakítása/leállítása | A rendszer komponenseinek állapota nem változik, azok a megszakítás pillanatában felvett értékeket mutatják a következő indításig. |
| Állapot | Az áramkör komponenseinek aktuális tulajdonságainak (kimeneti/bemeneti értékek) összessége |
| Alap állapot | Az a kiindulási állapot, mely az áramkör betöltése után keletkezik. Ilyenkor az <i>általános komponensek</i> kimenetén a hamis érték van. |
| Jelforrások konfigurációja | A szerkeszthető jelforrások beállításainak (kapcsolók állapota, jelgenerátorok bitsorozata stb.) egy állapota. |

2.5. Essential use-case-ek

2.5.1. Use-case diagram



2.4. ábra. Essential use-case diagram

2.5.2. Use-case leírások

| Use-case neve | Szimuláció leállítás |
|---------------|---|
| Rövid leírás | A futó szimuláció leállítása |
| Aktorok | Felhasználó |
| Forgatókönyv | Az éppen futó szimulációt a felhasználó a "Stop" gombbal leállítja. |

| Use-case neve | Szimuláció indítás |
|---------------|--|
| Rövid leírás | A áramkör szimulációjának elindítása |
| Aktorok | Felhasználó |
| Forgatókönyv | A felhasználó, az általa korábban kiválasztott áramkör szimulációját a "Start" gombbal elindítja |

| Use-case neve | Áramkör választás |
|---------------|---|
| Rövid leírás | Szimulálni kívánt áramkör kiválasztása |
| Aktorok | Felhasználó |
| Forgatókönyv | <ol style="list-style-type: none"> 1. A felhasználó a Megnyitás menüpontra kapcsol. 2. Kiválaszt egy áramkört a felkínált listából. 3. Az áramkör betöltődik a rendszerbe. |

| Use-case neve | Jelforrások betöltése |
|---------------|-------------------------------------|
| Rövid leírás | Az áramkör jelforrásainak betöltése |

| Aktorok | Felhasználó |
|--------------|---|
| Forgatókönyv | <ol style="list-style-type: none"> 1. A felhasználó a "Jelforrások betöltése" menüpontra kapcsol. 2. Megadja, hogy melyik fájlt olvassa be a rendszer. 3. Ha a fájl megfelelő, akkor a betöltés megtörténik, egyéb esetben a felhasználót figyelmeztetjük. |

| Use-case neve | Jelforrások mentése |
|---------------|---|
| Rövid leírás | Az áramkör jelforrásainak mentése |
| Aktorok | Felhasználó |
| Forgatókönyv | <ol style="list-style-type: none"> 1. A felhasználó a "Jelforrások mentése" menüpontra kapcsol. 2. Megadja, hogy melyik fájlba történjen a mentés. 3. A mentés megtörténik, amennyiben ez valami oknál fogva nem sikerült (nincs joga, nincs elég terület stb.), a felhasználót figyelmeztetjük. |

2.6. Napló

| Kezdet | Időtartam | Résztevők | Leírás |
|-------------------|-----------|--|--|
| 2011.02.15. 21:30 | 2 óra | Kriván B. Dévényi A. Jákli G. | Értekezlet. Döntések: Megegyeztünk a feladat értelmezését illetően. Milyen kérdéseket teszünk fel a konzulensnek az első konzultáción? Ezeket a Apagyi G. és Péter T. számára továbbítottuk. |
| 2011.02.16. 09:00 | 2 óra | Kriván B. Dévényi A. Apagyi G. Péter T. | Értekezlet. Döntések: <ul style="list-style-type: none"> • Fejlesztői környezetben megállapodtunk (2.1.2) • Projekt szervezési struktúráját rögzítettük (2.2.3) • A felmerülő algoritmusokról is konzultáltunk. |
| 2011.02.16. 15:00 | 1 óra | Jákli G. | Tevékenység: Projekt terv bővítése, formázása, javítása (2.2) |
| 2011.02.17. 16:00 | 1 óra | Jákli G. Kriván B. Dévényi A. | Tevékenység: a 2.1 és a 2.2 alfejezet közös átdolgozása |
| 2011.02.17. 19:15 | 45 perc | Jákli G. Kriván B. Dévényi A. | Értekezlet. Döntések: A 2.3 alfejezet főbb gondolatait megfogalmaztuk, és meghatároztuk, hogy mik legyenek a mindenképpen lejegyezendő dolgok. |
| 2011.02.17. 20:00 | 50 perc | Jákli G. | Tevékenység: A megbeszéltek alapján elkezdte a 2.3 alfejezet megírását. |

| Kezdet | Időtartam | Résztevők | Leírás |
|-------------------|-----------|---|---|
| 2011.02.17. 23:00 | 30 perc | Péter T. | Tevékenység: Szervezési struktúra kiegészítése képernyőképekkel |
| 2011.02.18. 00:00 | 30 perc | Kriván B. Dévényi A. Péter T | Értekezlet (Msn megbeszélés). Döntések: A 2.3 alfejezet módosításának elhatározása és a szótárba (2.4) bekerülő szavak rögzítése |
| 2011.02.18. 00:30 | 30 perc | Péter T. | Az előző értekezleten meghatározott szavak felvétele a szótárba, még csak felsorolás szintjén |
| 2011.02.18. 14:00 | 1 óra | Péter T. | Szótárban lévő szavak magyarázatainak kitöltése |
| 2011.02.18. 15:30 | 30 perc | Apagyi G. | Szótárban lévő szavak megmagyarázásának folytatása |
| 2011.02.19. 12:00 | 2 óra | Kriván B. | Tevékenység: helyesírási hibák javítása, szótár (2.4) szerkesztése (sorrendek változtatása, további szavak bevezetése, meglévők magyarázatainak szerkesztése) |