

**Szegedi Tudományegyetem
Informatikai Intézet**

**GÉZA – Informatikus zárthelyi dolgozat
ütemező és menedzselő rendszer**

Szakdolgozat

Készítette:
Vad Avar
mérnökinformatika BSc
szakos hallgató

Témavezető:
Dr. Németh Gábor
adjunktus

Szeged
2025

Feladatkiírás

A témavezető által megfogalmazott feladatkiírás. Önálló oldalon szerepel.

Tartalmi összefoglaló

- *A téma megnevezése:*

GÉZA

- *A megadott feladat megfogalmazása:*
- *A megoldási mód:*
- *Alkalmazott eszközök, módszerek:*
- *Elért eredmények:*
- *Kulcsszavak:*

Tartalomjegyzék

Feladatkiírás	2
Tartalmi összefoglaló	3
Tartalomjegyzék.....	4
BEVEZETÉS.....	6
1. A VIZSGA MÓD	7
1.1. Szabály jelentése, alkalmazása.....	8
1.2. Saját könyvtár használata (home)	9
1.3. Feladat kiosztásra minden gépre	10
1.4. A rendszer jelenlegi gyengeségei.....	10
2. GÉPTERMI ZH ADMINISZTRÁTOR (GÉZA).....	11
2.1. GÉZA felépítése.....	11
2.2. Funkcionális követelmények	12
2.2.1. LDAP bejelentkezés	12
2.2.2. Órarendkezelés	12
2.2.3. Órarenden kívüli kérések.....	13
2.2.4. Gépterem, mint erőforrások	13
2.2.5. Kommunikáció	13
2.3. Nem funkcionális követelmények	14
3. TECHNOLÓGIAI ÁTTEKINTÉS.....	14
3.1. Frontend technológiák	14
3.2. Backend technológiák	14
3.3. Külső integráció.....	14
4. ADATBÁZIS FELÉPÍTÉSE	14
4.1. Normalizálási elemzés	14
4.2. Megkötések függőségek miatt	15
4.3. Adatbázis részletes elemzése.....	15

5. RENDSZERARCHITEKTÚRA	15
5.1. Szerver- és kliensoldali komponensek	15
5.2. Jogosultságkezelés	15
5.3. Modulok a rendszerben	15
5.3.1. Használható modulok.....	15
5.3.2. Modulok beillesztése.....	15
6. INTEGRÁCIÓ.....	16
7. POTENCIÁL A JÖVŐRE NÉZVE	16
7.1. Moduláris bővíthetőség.....	16
7.2. Ideális vízió.....	16
8. ÖSSZEGZÉS	16
Irodalomjegyzék.....	17
Nyilatkozat	18
Köszönetnyilvánítás	19

BEVEZETÉS

Az Informatika Intézetnél jelenleg dolgozó oktatók sokszor találkozhattak már a zárhelyi dolgozat íratás problémájával és az ezzel járó adminisztratív teendőkkel. Ezek közül néhányat, a teljesség igénye nélkül, kiemelve: dolgozat összeállítása, feladatok kitűzése, kitűzött feladatok ellenőrzése, pontozás kitalálása. A felsoroltakon kívül gondolkodni kell azon, hogy a dolgozat megírásához egy olyan környezetet kell biztosítani, amelyben a hallgatóknak minden rendelkezésre áll ahhoz, hogy a tudását tudja bizonyítani. Oktatói szempontból viszont ne legyen túl nagy feladat, bizonyos esetekben akár 60 hallgató, egyszeri felügyelete és a számokérés tisztességességének biztosítása.

Ezen feladat megkönnyítésére az Irinyi Kabinetben rendelkezésre állnak olyan eszközök, amelyek arra hivatottak, hogy dolgozatok során az oktátónak ne kelljen 60 ember mögött ott állnia annak érdekében, hogy biztosítsa a korrekt eredményeket. Ezen eszközcsoportot a hallgatók és oktatók is egyaránt „ZH mód” vagy „vizsga mód” néven ismerik. Az egyszerűség kedvéért ezen dolgozat hivatkozik ezekre az eszközökre ezeken a neveken is.

Ez a fajta üzemmód lehetővé teszi azt, hogy a Kabinet számítógépet távolról vezérelve bizonyos funkciókat (pl. internetelérés, szoftverek használata, operációs rendszerek stb.) a rendszert üzemeltető operátorok letiltsanak ezzel segítve a „steril” környezet biztosítását.

A fent említett adminisztratív elfoglaltságok közé tartozik az is, hogy ennek az üzemmódnak a beütemezését kérje az oktató a saját órájára. Ez a kérés jelenleg egy ímélváltást jelent az oktató és az üzemeltetés között. Ezen kérést elolvasva az operátor beállítja az oktató által kért beállításokat. Így leírva ez a fajta, évek alatt már megszokott, protokoll nagyon jól működik, elméletben. A gyakorlat viszont mást mutat.

Látható, hogy sok a lehetőség hibázásra, mivel a folyamat nagyrészt emberek végzik. Ha az oktató elgépél valami a kérdésben, megtörténhet, hogy rossz beállítást kapnak a hallgatók, ha az operátor gépel el valamit, előfordulhat, hogy nem is jó teremben indul el az ZH mód.

A dolgozat pedig pont ezt a problémát tűzte ki maga elé, létre kell hozni egy rendszert, amely hosszútávon képes minimalizálni vagy teljesen eltüntetni a hibák lehetőségét. Úgy, hogy közben megmarad az operátori rálátás a teljes rendszerre, és az oktatók pedig továbbra is képesek maradnak zárhelyi dolgozat közben olyan környezetet biztosítani a hallgatóknak, amely lefed minden felmerülő igényt, mindkét oldalról.

Ezen dolgozat keretében elkészült szoftver alapja lehet egy ilyen rendszernek, viszont várhatóan alakításra és fejlesztésre szorul majd. Addig is, amíg ezen módosítások

megtörténnek a program minimalizálja a jelenlegi hibalehetőségeket és teret ad automatizációk integrálásának.

1. A VIZSGA MÓD

Az Irinyi Kabinet gépteremiben jelenleg „ZH mód” vagy „vizsga mód” néven ismert speciális üzemmód hivatott néhány alapvető szabály betartatni a hallgatói számítógépeken a számonkérések alatt. A Kabinetben erre több eszköz is rendelkezésre áll, ezen eszközök összességét nevezzük ZH módnak, esetenként pedig vizsga módnak.

A termekben lévő számítógépeken, felhasználói szempontból futó operációs rendszer valójában egy virtuális rendszer. Ezt a felhasználó egy, a hoszt gépen futó, menüből tudja saját magának kiválasztani mikor használni szeretné a számítógépet. Ez az első pont, ahol a vizsga mód egyik eszköze bele tud nyúlni a hallgató által már megszokott folyamatba. Lehetőség van ugyanis távolról befolyásolni, hogy milyen választható opciók jelennek meg ebben a menüben. Így, ha szeretné az oktató, hogy egy dolgozat csak Linux rendszerről lehessen megoldható, akkor képes a többi választási lehetőséget innen eltüntetni.

Ez a virtualizált megoldás lehetőség ad olyan rendszerképfájl betöltésére, amely az általános menüben nincs benne. Így biztosítva azt, hogy ha szükséges akkor más szoftvereket, szoftverbeállításokat kapjanak a gépek bekapcsoláskor. Ezen megoldás arra is képes, hogy az elindult gépek mindegyikén rajta legyen egy feladatsor vagy kiindulási fájlok a dolgozat megkezdésekor.

A virtualizáció viszont kétélű fegyver. A fenti pozitív hatásai mellett megemlítenéd, hogy a virtuális rendszer kikapcsolásakor a rajta lévő adatok is eltűnnek. Ez adatvédelmi szempontból nagyon hasznos, nem kell azzal foglalkozni, hogy valaki esetleg bejelentkezve maradt, és a következő hallgató pedig kihasználva ezt a nevében cselekszik. Az érme másik oldalán pedig ott van az, hogy egy zárhelyi megírása alatt nem szerencsés eset az, ha valami miatt a számítógépet újra kell indítani (ilyen eset lehet mondjuk egy technikai hiba vagy áramkimaradás) akkor a hallgató munkája teljesen elveszik. Ez a probléma a kiküszöbölhető azzal, ha valamilyen online felületre kell feltölteni a hallgatóknak a megoldásaikat, ahonnan esetlegesen vissza is tudják azt tölteni újraindítás után. Viszont vannak olyan esetek, mikor ez nem megoldható vagy túlzott, ok nélküli, pluszmunkát okozna. A Kabinetben viszont van erre kidolgozott eszköz, amely képes megoldást kínálni erre. Ezt egy későbbi fejezet tárgyalja.

Hálózati szempontból vizsgálva a termék kiépítését észrevehető, hogy minden terem egy saját virtuális hálózatban van (VLAN). Ez lehetővé teszi egy újabb eszköz bevetését, amellyel távolról vezérelhető, hogy a számítógépek hogyan érik el az internetet. Ezt úgy kell

elképzelni, hogy az üzemmód bekapcsolásakor teremspecifikusan, azaz a terem virtuális hálózatára érve, vezérelhető, hogy pontosan milyen internetes címeket érnek el az eszközök. Így elérhető az, hogy a hallgatói számítógépek semmilyen módon ne férjenek hozzá az internethez, ha erre valaki igényt tart. Kizárva ezzel a külső kommunikációs csatornákat vagy segítő eszközöket.

1.1. Szabály jelentése, alkalmazása

Ezen dolgozat, ha ZH mód kontextusában, szabályra hivatkozik, akkor egy olyan beállításösszességre referál, amelyek egyszerre, együttesen vannak alkalmazva egyetlen terem számítógépeire.

Az üzemeltetéstől megtudott információk alapján a jelenlegi rendszerben 2, 3 vagy 4, az előző fejezetben leírt beállítás alkot egy szabályt:

- rendszerválasztó menüben elérhető opciók
- internet- és hálózatbeállítások
- választható módon: ZH könyvtár a hallgatóknak
- választható módon: használt operációs rendszer leállítása első és/vagy utolsó lépésként

Amióta ebben a jelenlegi architektúrát használja a Kabinet összegyűlte már jól bevált, tesztelt és valamennyire moduláris szabályok. Amennyiben ezek közül kell valamelyiket alkalmazni, az operátori feladat lényegében annyi, hogy a megfelelő helyre be kell illeszteni a megfelelő fájl elindítására szolgáló sort.

A szabályok alkalmazását legegyszerűbb oly módon áttekinteni, hogy egy konkrét példát veszünk, és a lépésként vizsgáljuk. Feltételezzük, hogy a kérés, ami érkezett egy Programozás Alapjai Gyakorlat tárgyhoz érkezett, és a kért szabály már létezik, azaz módosítás nélkül csak be kell állítani a megfelelő kurzus időpontjára.

Első lépésként egy belső, megfelelő szervert kell elérni az operátornak, amelynek az időzített feladati közé (cron) felvesz egy újat a kért időpontra, mely elindít egy scriptet. Ezen futtatható állomány továbbiakat hív meg, amelyek ténylegesen felelősek a szabály beállításáért.

Mikor elérkezik a kezdési időpont, elindul az állomány, mely valamilyen sorrendben elvégzi a beállításokat. Mivel egy Programozás Alapjai dolgozathoz kérték a beállításokat a következők történnek. A sorrendiségtől el lehet tekinteni ebben az esetben, és gondolhatunk az eseményekre úgy, hogy „egyszerre” történnek.

A választómenüben csak a Linux operációs rendszer lesz választható. A többi választható opció eltűnik. Amennyiben kérve volt az oktató által, hogy a jelenleg használt operációs rendszer leálljon, ez is megtörténik – tehát előre megnyitott weboldalak/programok bezáródnak.

A terem virtuális hálózatára olyan access-control list (ACL) beállítások kerülnek alkalmazásra, amelyek tiltják az internet általános elérését, és – feltételezve, hogy az új tárgyról beszélünk – csak a Bír3 érhető el. Legalábbis felhasználói szemmel ez látszik, valójában több dolog is elérhető az internetről, amelyek szükségesek ahhoz, hogy a használt szoftverek valójában működjenek. Ennek viszont felhasználói szempontból nincs jelentősége, mert az, hogy egy licenszserver elérhető alapfeltételezés a használatnak. Illetve természetesen a Bír3 sem egy olyan alkalmazás, amely önmagában áll, vannak külső függőségei, amelyek szintén át vannak engedve ezen a szűrőn.

Ezen felül az előre létre hozott „ZH home” könyvtárak lesznek felcsatolhatóak a „Home Mount” előre megírt script futtatásával. Ez biztosítja azt, hogy lehetőség van a Bír3 felületén kívül is adatmentésre a felmérés alatt. Ezen fájlok csak a dolgozat ideje alatt lesznek elérhetők, utána hallgatóként nem lehet hozzáférni.

Ebben az esetben mindhárom lépést elvégezte a szabály beállításához a rendszer, ez utóbbi opcionális, nem mindenhol van használatban.

1.2. Saját könyvtár használata (home)

Az intézet mindenki számára biztosít egy saját könyvtárat, amelyet a hallgató alapesetben bármikor bárholnan elérhet, és tárolhat rajta fájlokat. Ez a könyvtár elérhető bárholnan az internetről, csak a felhasználó nevet (ismertebb nevén a „h-s azonosítót”) és a hozzá tartozó jelszót kell tudni. Egy ilyen könyvtár azonban potenciálisan lehetőséget kínál azoknak, akik külső segítséget szeretnének használni egy dolgozat alatt.

Látható, hogy nem lenne szerencsés, ha ehhez a könyvtár hozzáférhető lenne a dolgozat megírása közben, ugyanis ebben elmenthető bármilyen állomány, akár otthonról, ami aztán a számonkérés alkalmával letölthető és használható. Éppen emiatt, a ZH üzemmódba állított számítógépek egy külön erre a célra létrehozott home könyvtárat csatolnak fel. Ez a könyvtár kezdetben üres, a dolgozat ideje alatt a hallgató hozzáférhet, menthet bele fájlokat, és ha újra kell indítani a számítógépét, akkor is megmaradnak. A felmérés után pedig kérés esetén ezen fájlokat az oktató megkaphatja, és szükség esetén átnézheti, kiértékelheti azokat.

Ez a megoldás lehetővé teszi, hogy teljesen leválasszák a hallgatókat az internetről, ha szükséges. Így nem áll fenn az a probléma, hogy nem tudják a hallgatók lementeni a

munkájukat, és valami miatt elveszne minden, amin dolgoztak. Technikai oldalról ennek a pontos megvalósítása nem lényeges jelen kontextusban.

1.3. Feladat kiosztásra minden gépre

Az utóbbi néhány évben talán egy kevésbé jellemző funkciója a vizsgamódnak az, hogy az oktatónak lehetősége van fájlokat feltölteni a számítógépekre, így mikor azok elindulnak a hallgatóknak már rendelkezésre állnak feladatok vagy kiindulóállományok.

Ezt az eszközt leváltotta jórészt a Coospace, illetve az ahhoz hasonló egyéb online, vagy belső hálózatról elérhető szolgáltatások. Példaképp a Bíró3 használatához nem szükséges ezt a funkciót használni, ugyanis az a feladat beadási felület mellett magával a feladattal is rendelkezik. Így fölösleges és csak plusz terhelés (esetlegesen hibaforrás) az, hogy a feladatok külön állományként is felkerüljenek a gépekre.

A múltban viszont használt eszköz volt ez, és mivel a Kabinet alapvető architektúrája azóta nem változott szignifikánsan, ezért ez továbbra is egy igénybe vehető lehetőség.

1.4. A rendszer jelenlegi gyengeségei

A fenti, igencsak komplex, rendszert áttekintve látható, hogy még akkor is, ha feltételezzük, hogy az évek óta megírt és használt beállítások, scriptek nem romlanak el, van néhány olyan pont, ahol az emberi beavatkozás miatt hibák léphetnek fel.

Az első ilyen rögtön a levél, amit az oktató ír az üzemeltetésnek, amennyiben abban az időpontot elrontja, vagy rossz szabályt kér akkor elképzelhető, hogy a számonkérés nem úgy fog történni, ahogy elképzelte. Azt lehet feltételezni, hogy amennyiben rossz teremszámot, vagy rossz időszávot (a hét napja, napon belüli óra) ad meg az oktató azt a túloldalon ellenőrzés után az operátor elutasítja, ugyanis beállítás előtt minden esetben ellenőrzésre kerül (manuálisan) az órarend is.

Természetesen az operátor is hibázhat, viszont ebben az esetben nagyobb problémák is adódhatnak. Egy rosszul beírt dátum, teremszám vagy időpont jelentheti az, hogy az eredetileg igényelt ZH mód nem kapcsol be, viszont egy másik teremben, ahol elképzelhető, hogy éppen oktatás zajlik, pedig újra indulnak a számítógépek. Szerencsére ezen hibák ritkán fordulnak elő, a lehetőség viszont adott rá.

Feltételezve, hogy hiba nélkül sikerült beállítani mindent. Az oktató az igényét elküldte, az alapján az operátor helyesen állította be az időzítéseket és a megfelelő scriptet használat. Ám, előre nem látható okok miatt változtatni kell valamelyik beállításon. Egy elképzelt forgatókönyv lehet, hogy a kiadott feladatsorban hiba van, ami miatt az oktató úgy dönt, hogy

plusz időt ad a hallgatóknak a megírásra. Mivel viszont a ZH mód fix időre van beállítva, ezért annak végeztével – beállítástól függően – kikapcsolhatnak a számítógépek. Ami azt jelenti, hogy a hallgató azon munkája, amit nem a home könyvtárában tárolt, vagy nem töltött fel egy online beadófelületre elveszik. Ezt viszont el lehet kerülni azzal, ha időben jelezzük a problémát az üzemeltetés felé. Ekkor ugyanis van lehetőség arra, hogy tovább maradjon ez az üzemmód a teremben. Viszont ez nem minden esetben egy kivitelezhető megoldás, különböző okok miatt (pl. felügyelet nélkül hagyott terem, szünetét töltő operátor, utolsó pillanatos változtatás).

Az előző eset feltételezte, hogy a kommunikáció a felhasználó és az operátor között lezajlott. Néhány esetben viszont ez nem egy ímélváltás oda-vissza, hanem levelezések hosszú sora, amit egy idő után nehéz követni. Ez pedig előidézhet, az alfejezet elején tárgyalt, hibákat.

2. GÉPTERMI ZH ADMINISZTRÁTOR (GÉZA)

Az előző fejezetben említett gyengeségeket felismerve, és azok hatásának csökkentésére jött létre a „Géptermi ZH Adminisztrátor”, röviden csak GÉZA névre keresztelt alkalmazás. Bizonyos szempontból tekinthető új üzemeltetési kollégának is. GÉZA egy olyan webalkalmazás, amely megkönnyíti mind operátori, mind oktatói szempontból a jelenlegi vizsgamód rendszer valamennyi aspektusát.

Oktatói szempontból egy felhasználóbarát, modern felületet kínál az igények benyújtására, amelyben a lehető legtöbb hibalehetőség elkerülésére van beépített védelem. Operátori szempontból pedig egy egységes, könnyen használható felületet kínál az igények kezelésére.

GÉZA az első lépése annak, hogy a vizsga mód rendszer jelenlegi robusztusságát megtartva, megfeleljen az igencsak növekvő igényeknek. Illetve, a mai elvárásokhoz mérten igyekezzon a hibák lehetőségét minimalizálni.

2.1. GÉZA felépítése

GÉZA felépítésének tervezésekor fontos szempont volt, hogy a háttérrendszer, ami a tényleges vizsgamód beállításokat kezeli már adott. Úgy kellett erre ráépíteni, hogy a jelenlegi funkcionalitás megmaradjon, és emellett egy könnyen kezelhető, biztonságos és rugalmas alkalmazást kapjunk.

Ennek eléréséhez GÉZA alapjául egy React alapú keretrendszer szolgál, a NextJS. A választás amiatt esett erre, mert a keretrendszer egyszerre kezeli a felhasználó oldali megjelenítéseket és a szerver oldali funkcionalitást is ellátja. Így fejlesztői szempontból

könnyen átlátható, kezelhetővé téve a projektet. Előnyei közé tartozik még az is, hogy egy régóta a piacon található könyvtár szolgáltatja az alapját, így hosszú távon nagy támogatottságra lehet számítani.

Fontos szempont volt, hogy amennyire lehet a jelenleg rendelkezésre álló információkat ne kelljen egy plusz felületre is felvinni. Emiatt több integráció is része a teljesen felépítésnek. Egyik ilyen például a beléptetés, amely a már megszokott Lightweight Directory Access Protocol (LDAP) azonosítókat használja, másik példa erre pedig a Belső Információs Rendszer (BIR), ami pedig az órarendi adatokat szolgáltatja.

2.2. Funkcionális követelmények

Az alábbi fejezetekben megfogalmazott követelmények egy tipikus felhasználási forgatókönyvet végig gondolva és azt szem előtt tartva születtek meg. A megfogalmazásuk során az operátorokkal – mint egyik végfelhasználó –, és néhány oktatóval – mint másik végfelhasználó – is történt egyeztetés, hogy a lehető legjobban le lehessen fedni az igényeiket.

2.2.1. LDAP bejelentkezés

Az internet egyre inkább eltávolodik a tipikus felhasználónév/ímélcím és jelszó párostól és egyre inkább a külső szolgáltatói azonosítás irányába mozdul a sztenderd. Ez sokkal biztonságosabb és hosszútávon fenntarthatóbb rendszereket eredményez.

Így GÉZA tervezésénél alapvetés volt, hogy saját azonosítási megoldás nélkül készüljön a rendszer. Az azonosítást teljes egészében a már létező, és az Intézetben belül használt, Lightweight Directory Access Protocol (LDAP) biztosítja. Ezzel az beléptetés problémáját teljes egészében levéve GÉZA-ról. Az LDAP rendszerében mindenkinek saját, egyedi és központi azonosítója van, így belépve a GÉZA-nak már minden szükséges adat rendelkezésére áll, ami ahhoz kell, hogy biztosítsa a zökkenőmentes használatot.

Ez megtiltja a jogosulatlan felhasználók általi rendszerhasználatot, nem hagy teret a feltörések által okozott biztonsági réseknek, és centralizálja a felhasználó kezelést. Ez utóbbi igen nagy előny egy akkora szervezetnél, mint az Informatika Intézet, ahol közel sem lenne kivitelezhető, hogy minden egyes oktatónak ide is külön felhasználót kelljen készíteni.

2.2.2. Órarendkezelés

Az előbb említett egyedi azonosító (továbbiakban „LDAP azonosító” néven is említve) lehetővé teszi azt, hogy egy másik, szintén gyakorlatban használt rendszert integráljunk be.

A Belső Információs Rendszer (BIR) évek óta szolgáltat funkciókat az oktatóknak és hallgatóknak. Ezen funkciók között van az órarend kezelés is. Ide minden félév elején felkerülnek az aktuális Kabineti órarendek – pontosan azok, amikre később ZH módot foglalnak majd az oktatók. Ezen adatokat pedig GÉZA fel tudja használni arra, hogy az oktatóknak egy olyan felületet adjon, amelyen kényelmesen és átláthatóan tudnak foglalni az órájukra vizsga módot.

Az integráció pontos részleteit egy későbbi fejezet taglalja, viszont ami lényeges, hogy az adatokat egyenesen ezen felület adatbázisából veszi át GÉZA. Így elég ide felvinni az adatokat, és foglalásokat már létre lehet hozni. Mivel a BIR tárolja azt is, hogy az egyes órához mely oktatók vannak hozzárendelve, ezért ezt az információt is át tudjuk venni.

Az átvett órarendi információk biztosítják, hogy az oktatók csak saját órájukra, és azon belüli időpontokra tudjanak csak foglalni. Ezzel is limitálva a hibák lehetőségét.

2.2.3. Órarenden kívüli kérések

Nem lehet teljes egészében a BIR-re hagyatkozni, ugyanis vannak olyan speciális esetek, amikor órarenden kívül is szükséges lehet ZH módot beállítani. Ilyen helyzetek tipikusan pótlás/vagy javítódolgozatok alkalmával vannak, esetlegesen vizsgaidőszakban, gyakorlati vizsgáknál.

Ezekre is valamilyen módon lehetőséget kell adnia a rendszernek. Az egy nagyon kaotikus megoldás lenne, ha bárki (bármelyik oktató) tudna egyéni kéréseket létrehozni, bármikor amikor csak akar. Ezért egy olyan megoldást kellett megvalósítani, amely elérhetővé teszi ezt a funkciót, anélkül, hogy átláthatatlan lenne a rendszer. Ez a megoldás pedig az eredetileg is használt, íméles kérések, formájában valósul meg. Az operátor tud rögzíteni egyéni kérést bármikor, ha az oktató ezt ímélnélben igényli tőle.

2.2.4. Géptermek, mint erőforrások

- időzített ZH-k áttekintése
- hibás foglalások jelzése

2.2.5. Kommunikáció

- ímél küldése foglalásról
- státuszváltozás közlése ímélnélben

2.3. Nem funkcionális követelmények

- gyorsaság
 - átláthatóság
 - könnyű kezelhetőség
 - modern kinézet (dark/light mód)
 - továbbfejleszthetőség

3. TECHNOLÓGIAI ÁTTEKINTÉS

3.1. Frontend technológiák

- react (nextjs)
 - tailwind
 - shadcn UI
 - miért pont ezek

3.2. Backend technológiák

- nextjs routes
 - better auth
 - mysql adatbázis (Drizzle)
 - miért pont ezek

3.3. Külső integráció

- LDAP
 - BIR
 - ímél

4. ADATBÁZIS FELÉPÍTÉSE

- használt táblák, és magyarázat hozzá
 - relációk leírása

4.1. Normalizálási elemzés

- mennyire normális az adatbázis
 - hogyan lehetne optimálisabb

4.2. Megkötések függőségek miatt

- better auth pl. saját id-t generál mindenképp
 - inkább legyen megbízható/elismert külső beléptetés, mint „tökéletes” adatstruktúra

4.3. Adatbázis részletes elemzése

- BIR adatbázis megemlítése, nézettábla
 - belső adatstruktúra
 - EK diagram
 - szabadszavas leírása a tábláknak
 - indexek/kulcsok magyarázata

5. RENDSZERARCHITEKTÚRA

- nextjs alapvető felépítése
 - logikai szétválasztása a komponenseknek

5.1. Szerver- és kliensoldali komponensek

- különbség
 - tényleges használata ezeknek az alkalmazásban példákkal

5.2. Jogosultságkezelés

- rangrendszer (3-4 felhasználótípus)
 - bővíthető jogkörökkel
 - moduláris felépítés

5.3. Modulok a rendszerben

- mit értek modul alatt
 - mitől modul egy modul

5.3.1. Használható modulok

- az elkészült modulok
 - hogyan működnek

5.3.2. Modulok beillesztése

- hogyan lehet ezeket hozzáadni

6. INTEGRÁCIÓ

- hogyan van/lesz integrálva a kabinetben
 - docker
 - webszerver
 - ímélezés

7. POTENCIÁL A JÖVŐRE NÉZVE

7.1. Moduláris bővíthetőség

- a jövőben hozzáadható modulok

7.2. Ideális vízió

- hogyan működne egy „teljesen” „kész” rendszer
 - mit kellene tudnia

8. ÖSSZEGZÉS

- bevezető csak másképp

Irodalomjegyzék

Nyilatkozat

Alulírott Vad Avar, mérnökinformatika BSc szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Képfeldolgozás és Számítógépes Grafika Tanszék Tanszékén készítettem, mérnökinformatikus BSc diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat a Szegedi Tudományegyetem Diplomamunka Repozitóriumban tárolja.

Dátum

Aláírás

Köszönetnyilvánítás

Dolgozatom nem készülhetett volna el témavezetőm, Dr. Németh Gábor nélkül, aki már az első ímélváltás óta nagyon támogatja és jó ötletnek gondolta egy ilyen rendszer kialakítását. Az ő segítségével, ötletei és szakmai meglátásai nélkül ezen rendszer talán sosem készült volna el ilyen minőségben. Türelméért és megértéséért különösen hálás vagyok neki.

Továbbá külön köszönet az Irinyi Kabinet munkatársainak, név szerint Angyal Tamás, Hemmert János, Kocsorné Sebő Marianna, [HIÁNYOZNAK MÉG NEVEK], akik többször szánták rám idejüket, hogy átbeszéljük a rendszer működését, és integrálását. A technikai hátterét a rendszernek ők biztosítják, és végsősoron ők üzemeltetik majd.

Ezen felül a rendszer előzetes átbeszélését, és az ideális vízió felállításáért köszönet illeti Goldmann Júliát és Gercsó Márkot. Remélem egyszer elkészül az ideális rendszer, amit együtt kitaláltunk.