

**Szegedi Tudományegyetem  
Informatikai Intézet**

**GÉZA – Informatikus zárhelyi 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:*

- *Alkalmasztott eszközök, módszerek:*

- *Elért eredmények:*

- *Kulcsszavak:*

## **Tartalomjegyzék**

<b>Feladatkiírás .....</b>	<b>2</b>
<b>Tartalmi összefoglaló .....</b>	<b>3</b>
<b>Tartalomjegyzék.....</b>	<b>4</b>
<b>BEVEZETÉS.....</b>	<b>6</b>
<b>1. A VIZSGA MÓD .....</b>	<b>7</b>
<b>1.1. Alkalmazható szabályok .....</b>	<b>7</b>
<b>1.2. Saját könyvtár használata (home) .....</b>	<b>7</b>
<b>1.3. Feladat kiosztásra minden gépre .....</b>	<b>7</b>
<b>2. GÉPTERMI ZH ADMINISZTRÁTOR (GÉZA).....</b>	<b>7</b>
<b>2.1. GÉZA felépítése.....</b>	<b>7</b>
<b>2.2. Funkcionális követelmények .....</b>	<b>7</b>
2.2.1. LDAP bejelentkezés .....	7
2.2.2. Órarendkezelés .....	8
2.2.3. Órarenden kívüli kérések.....	8
2.2.4. Géptermek, mint erőforrások .....	8
2.2.5. Kommunikáció .....	8
<b>2.3. Nem funkcionális követelmények .....</b>	<b>8</b>
<b>3. TECHNOLÓGIAI ÁTTEKINTÉS.....</b>	<b>8</b>
<b>3.1. Frontend technológiák .....</b>	<b>8</b>
<b>3.2. Backend technológiák .....</b>	<b>8</b>
<b>3.3. Külső integráció.....</b>	<b>8</b>
<b>4. ADATBÁZIS FELÉPÍTÉSE.....</b>	<b>9</b>
<b>4.1. Normalizálási elemzés .....</b>	<b>9</b>
<b>4.2. Megkötések függőségek miatt .....</b>	<b>9</b>
<b>5. RENDSZERARCHITEKTÚRA .....</b>	<b>9</b>
<b>5.1. Szerver- és kliensoldali komponensek.....</b>	<b>9</b>

<b>5.2. Jogosultságkezelés .....</b>	<b>9</b>
<b>5.3. Modulok a rendszerben .....</b>	<b>9</b>
5.3.1. Használható modulok .....	9
5.3.2. Beépíthető modulok .....	9
<b>6. INTEGRÁCIÓ.....</b>	<b>10</b>
<b>7. POTENCIÁL A JÖVŐRE NÉZVE .....</b>	<b>10</b>
<b>7.1. Moduláris bővíthetőség.....</b>	<b>10</b>
<b>7.2. Ideális vízió.....</b>	<b>10</b>
<b>8. ÖSSZEGZÉS .....</b>	<b>10</b>
<b>Irodalomjegyzék .....</b>	<b>11</b>
<b>Nyilatkozat .....</b>	<b>12</b>
<b>Köszönetnyilvánítás .....</b>	<b>13</b>

## **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 felsoroltokon 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 tisztességeségének biztosítása.

Ezen feladat megkönnyítésére az Irinyi Kabinetben rendelkezésre állnak olyan eszközök, amelyek arra hivatottak, hogy dolgozatok során az oktatónak 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 mászt mutat.

Látható, hogy sok a lehetőség hibázásra, mivel a folyamat nagyrészét emberek végzik. Ha az oktató elgépel valami a kérésben, 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 eltűntetni 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 igény, minden 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**

- virtuális gépek (működési elv)
  - operációs rendszerek
  - internetelérés, virtuális hálózatok
  - szoftverhasználat
  - ZH home könyvtár

### ***1.1. Alkalmazható szabályok***

- főmenüből választható opciók
  - internetszabályok
  - szoftverhasználat

### ***1.2. Saját könyvtár használata (home)***

- egyetlen módszer a géphibák miatti adatvesztés megelőzésére
  - létre kell ezeket hozni előre

### ***1.3. Feladat kiosztásra minden gépre***

- lehetősége ad fájlok (tipikusan zip, pdf) feltöltésére a ZH operendszerre
  - bíró/cs használatával kevésbé használt funkció

## **2. GÉPTERMI ZH ADMINISZTRÁTOR (GÉZA)**

- webalkalmazás
  - automatizáció első lépése

### ***2.1. GÉZA felépítése***

- modern webalkalmazás alapok (nextjs/react)
  - szerver és kliensoldal egyszerre
  - LDAP belépés
  - BIR kommunikáció

### ***2.2. Funkcionális követelmények***

- ide talán nem kell semmi, egyből a következő alfejezet jön

#### ***2.2.1. LDAP bejelentkezés***

- biztosítja a jogosultság ellenőrzést
  - könnyíti az adminisztratív teendőket

## **2.2.2. Órarendkezelés**

- BIR rendszerrel kommunikáció
  - szükségtelen újra felvenni a már meglévő adatokat
  - órák megosztása (fő és főfő gyakorlatvezető)

## **2.2.3. Órarenden kívüli kérések**

- felvihetők a rendszerbe, megtekintés céljából
  - a

## **2.2.4. Géptermek, mint erőforrások**

- időzített ZH-k áttekintése
  - hibás foglalások jelzése

## **2.2.5. Kommunikáció**

- ímél küldése foglalásról
  - státuszváltozás közlése ímélben

## **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

## **3.2. Backend technológiák**

- nextjs routes
  - better auth
  - mysql adatbázis (Drizzle)

## **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 mindenkorban
- inkább legyen megbízható/elismert külső beléptetés, mint „tökéletes” adatstruktúra

## 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. Beépíthető modulok

- a jövőben hozzáadható modulok

- hogyan lehet ezeket hozzáadni

## 6. INTEGRÁCIÓ

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

## 7. POTENCIÁL A JÖVŐRE NÉZVE

### 7.1. *Moduláris bővíthetőség*

- 5.3.2. bővebb kifejtése

### 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, ..... 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 / diplomamunkámat a Szegedi Tudományegyetem Diplomamunka Repozitóriumában tárolja.

Dátum

Aláírás

## **Köszönetnyilvánítás**

Dolgozatom nem készülhetett volna el témavezetőm, Dr. Németh Gábor nélkül, aki már az első ímélváltás óta nagyon támogatta és jó ötletnek gondolta egy ilyen rendszer kialakítását. Az ő segítsége, 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 háttérét a rendszernek ők biztosítják, és végső soron 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.