



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

PROGRAMOZÁSELMÉLET ÉS SZOFTVERTECHNOLÓGIAI  
TANSZÉK

## Shalendar

*Témavezető:*

Pintér Balázs

egyetemi adjunktus, PhD

*Szerző:*

Kertész János

programtervező informatikus BSc

*Budapest, 2025*

## SZAKDOLGOZAT TÉMABEJELENTŐ

### Hallgató adatai:

Név: Kertész János

Neptun kód: AM2VZ8

### Képzési adatok:

Szak: programtervező informatikus, alapképzés (BA/BSc/BProf)

Tagozat : Nappali

Belső témavezetővel rendelkezem

### Témavezető neve: Pintér Balázs

munkahelyének neve, tanszéke: ELTE IK, Programozásmélet és Szoftvertechnológia Tanszék

munkahelyének címe: 1117, Budapest, Pázmány Péter sétány 1/C.

beosztás és iskolai végzettsége: egyetemi adjunktus, PhD

A szakdolgozat címe: Shalendar

### A szakdolgozat témája:

(A témavezetővel konzultálva adja meg 1/2 - 1 oldal terjedelemben szakdolgozat témájának leírását )

A dolgozat témája egy mindennapi életben használható time management szoftver megvalósítása. A projekt felépítése három fő felületre és egy bejelentkezési oldalra oszlik, az utóbbi lehetővé teszi több felhasználó számára a profil kezelését és az egymás közötti naptár megosztást.

#### Főoldal:

Az oldalon egy naptár található, mellette pedig minimum egy oszlop, amely a feladatkezelő szoftverekből ismert lista formátumot követi. A felhasználó igényei szerint több, saját tematikáinak megfelelő oszlopot is felvehet. Az oszlopban kártyák helyezhetők el, amelyeket a felhasználó szabadon hozzáadhat vagy törölhet. A kártyákon kötelezően megadható cím, valamint opcionálisan kezdeti dátumok, határidők és prioritások. Ezek a kártyák a naptár megfelelő napjaira húzhatók, így segítve a feladatok ütemezését.

#### Napi nézet:

A naptár adott napjára kattintva megjelenik a nap részletezése. Itt két lista található: Az egyik lista egy időjelző sávval rendelkező feladatlista, amelyben a naphoz tartozó, időponthoz kötött feladatok jelennek meg. A másik lista olyan teendőket tartalmaz, amelyek nem kötöttek időponthoz. Mindkét listában a feladatok „elvégeztnek” jelölhetők.

#### Több naptár kezelése:

A naptár mellett található egy plusz gomb, amely lehetővé teszi több naptár létrehozását és kezelését. Ezekhez a naptárakhoz további felhasználók is hozzáadhatók, valamint lehetőség nyílik a naptárakból egyes kártyák vagy teljes naptárak importálására a saját naptárba.

Budapest, 2024. 10. 05.

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>2</b>
<b>2. Felhasználói dokumentáció</b>	<b>3</b>
<b>3. Fejlesztői dokumentáció</b>	<b>4</b>
3.1. Használati esetek . . . . .	4
3.2. Architektúra . . . . .	9
3.2.1. Adatbázis – Microsoft SQL Server . . . . .	9
3.2.2. Backend – ASP.NET WebAPI . . . . .	9
3.2.3. Fájrendszer felépítése . . . . .	11
3.2.4. Frontend – VUE.JS alapú webes felület . . . . .	11
3.2.5. Fájrendszer felépítése . . . . .	11
<b>4. Összegzés</b>	<b>12</b>

# 1. fejezet

## Bevezetés

## 2. fejezet

### Felhasználói dokumentáció

## 3. fejezet

# Fejlesztői dokumentáció

### 3.1. Használati esetek

Ezt latex táblázatok segítségével kéne megcsinálni, mert úgy jobban törik

Egy use case diagramm itt jól mutatna

### Bejelentkezési oldal funkciói

Regisztráció	
Given	A felhasználó megnyitotta az alkalmazást
When	A regisztrációs lehetőséget választja
Then	A rendszer belépteti a főoldalra, ahol láthatja a naptárát és a feladatlistáit.

Bejelentkezés	
Given	A felhasználó megnyitotta az alkalmazást
When	A bejelentkezés lehetőséget választja, majd bejelentkezik
Then	A rendszer belépteti a főoldalra, ahol láthatja a naptárát és a feladatlistáit.

## Főoldal funkciói

Új feladatlista létrehozása	
Given	A felhasználó a főoldalon van
When	A feladatlisták szerkesztése opción belüli feladatlista létrehozása opciót választja. Majd megfelelően paraméterezve létrehozza
Then	A feladatlista megjelenik a főoldalon.

Feladatlista törlése	
Given	A felhasználó a főoldalon van
When	A feladatlisták szerkesztése opción belüli feladatlista törlése opciót választja. Majd megfelelően paraméterezve törli
Then	A feladatlista valamint a hozzá tartozó kártyák törlődnek.

Kártya létrehozása az adott feladatlistához	
Given	A felhasználó a főoldalon van
When	A feladatlistán új kártya létrehozására kattint, majd megfelelően paraméterezi azt.
Then	A kártya megjelenik a kívánt oszlopban.

Kártya törlése az adott feladatlistáról	
Given	A felhasználó a főoldalon van
When	A kártyát kitörli
Then	A kártya törlődik a feladatlistáról

Kártya áthelyezése	
Given	A felhasználó a főoldalon van és van egy oszlop egy meglévő kártyával.
When	A felhasználó a kártyát egy másik oszlopba, vagy a naptár valamelyik mezőjére húzza.
Then	A kártya átkerül az új oszlopba vagy a naptárba.

Naptár napjának megnyitása	
Given	A felhasználó a főoldalon van
When	A naptár napjára kattint
Then	A naphoz tartozó napi nézet megnyílik.

További naptárak kezelése	
Given	A felhasználó a főoldalon van
When	A naptár mellett a további naptárak gombra kattint
Then	A további naptárak oldal megnyílik.



### Napi nézet funkciói

Kártya megjelölése elvégzettként	
Given	A felhasználó a napi nézet oldalán van
When	A kártyát megjelöli elvégzettként
Then	A kártya elvégzettként lesz megjelenítve

Kártya időpontra osztása	
Given	A felhasználó a napi nézet oldalán van és van legalább 1 kártya az időponthoz nem kötött feladatlistában
When	A felhasználó a kártyát az időponthoz kötött feladatlistába húzza
Then	A kártya a megfelelő időpontban megjelenik az időponthoz kötött feladatlistában

Kártya visszaküldése a főoldalra	
Given	A felhasználó a napi nézet oldalán van és van legalább 1 kártya ami a főoldal valamelyik feladatlistájában volt.
When	A felhasználó a kártyát visszaküldi a főoldalra
Then	A kártya visszakerül az eredeti feladatlistájába

### Több naptár kezelése funkciói

Új naptár létrehozása / csatlakozás	
Given	A felhasználó a több naptár fülön van
When	Az új naptár létrehozása / csatlakozás gombra kattint
Then	A rendszer létrehoz egy új naptárat, amelyhez további felhasználókat is hozzáadhat. / Csatlakozik a naptárhoz

Naptárból való importálás	
Given	A felhasználó a több naptár fülön van
When	A felhasználó a naptárból való importálás gombra kattint.
Then	A kívánt elemek importálásra kerülnek a saját naptárba

Naptár törlése vagy követésének megszüntetése	
Given	A felhasználó a több naptár fülön van
When	A naptár törlése/ leiratkozás gombra kattint
Then	A törlés / leiratkozás végrehajtásra kerül, az adott naptár többé nem látható az egyéb naptárak fülön

## 3.2. Architektúra

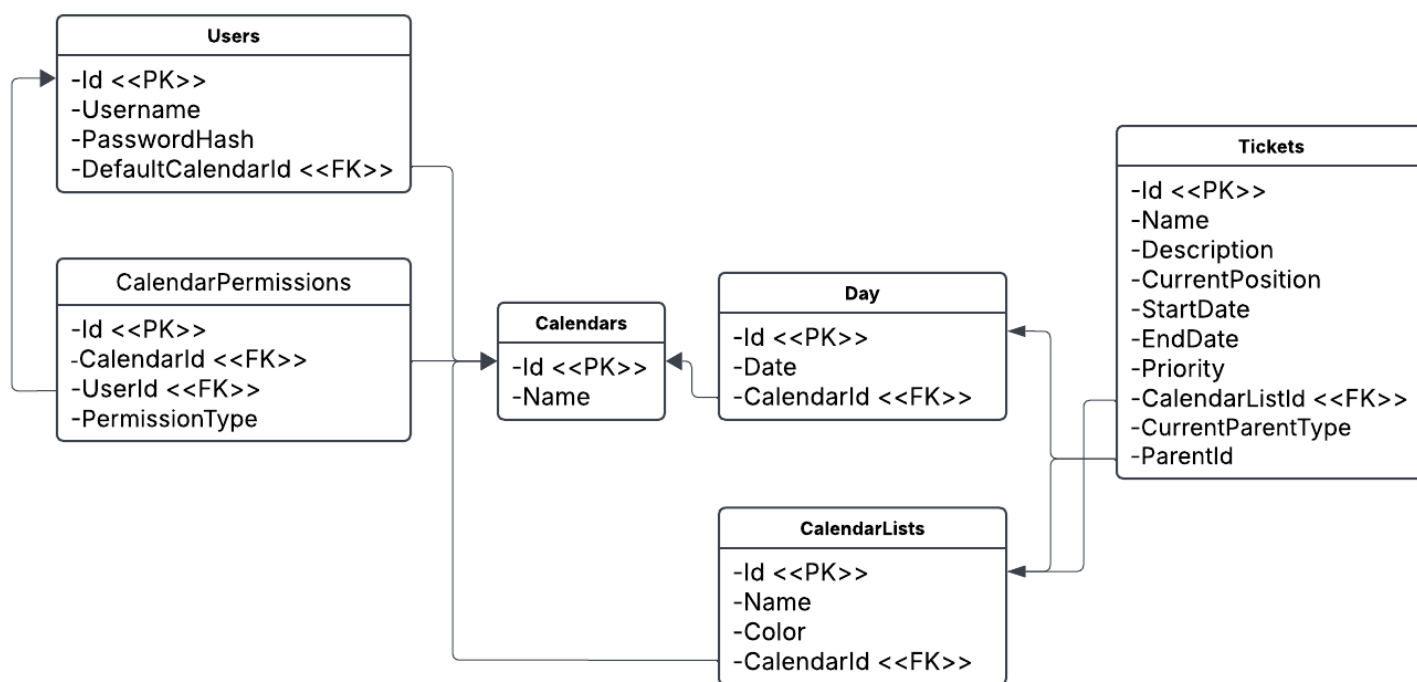
Az alkalmazás a kliens-szerver modellt követve 3 fő komponensből áll. A Microsoft SQL Server adatbázis, az ASP.Net WebAPI alapú backend és a VUE.JS-alapú frontend. Az első két komponens a szervert, míg a harmadik a klienst képviseli.

Az adatbázisban szereplő táblák feldolgozásában a szerver az Entity Framework-ot hívja segítségül majd RESTful API-kon keresztül szolgálja ki a kliens által intézett kéréseket. Ezeket a Vue.js keretrendszer segítségével dinamikusan feldolgozva és megjelenítve biztosítja a felhasználó számára.

### 3.2.1. Adatbázis – Microsoft SQL Server

Az adatbázis a backend által kezelt adatok perzisztens tárolását végzi, ideértve a felhasználók, naptárak, feladatok, kártyák és egyéb entitások adatait.

#### Entitások és kapcsolatok



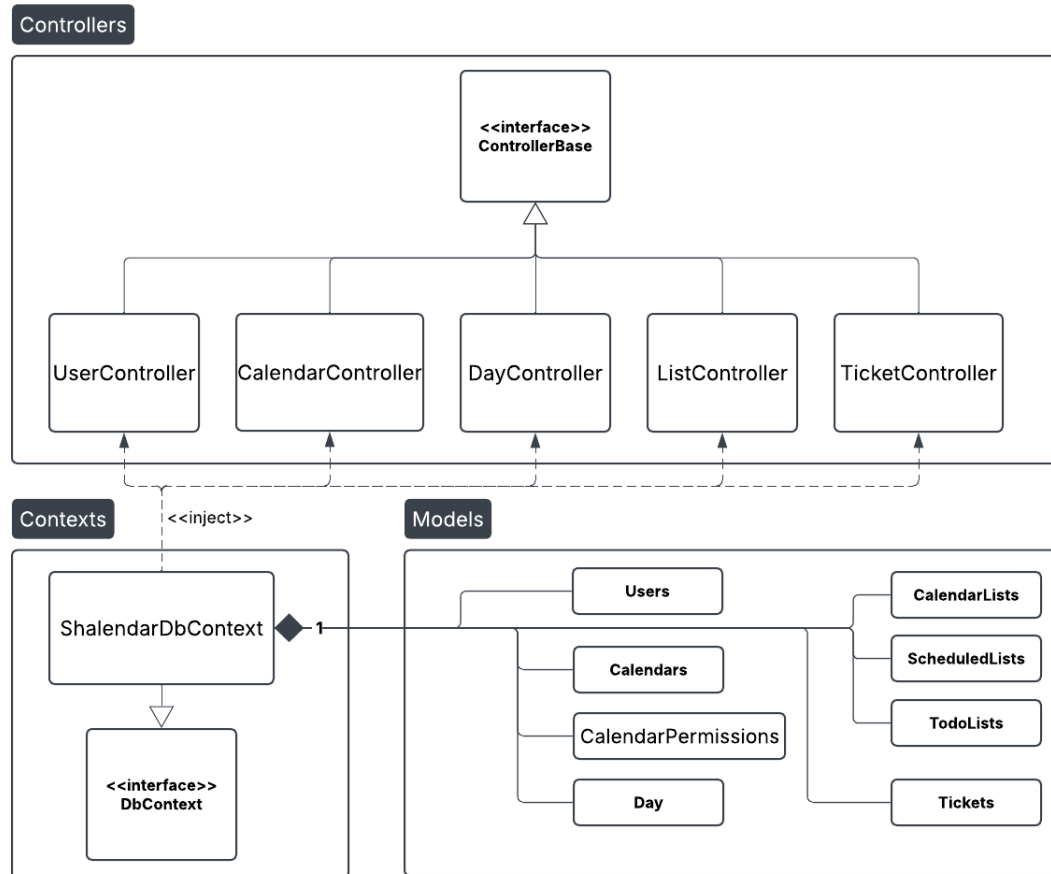
3.1. ábra. Adatbázis entitások és kapcsolatok

### 3.2.2. Backend – ASP.NET WebAPI

A backend feladata a kliens kérések fogadása és feldolgozása, az üzleti logika kivitelezése, az adatbázisban szereplő megfelelő adatok kinyerése Entity Framework

segítségével és kiszolgálása a front-end számára RESTful API-kon keresztül.

## Üzleti logika



3.2. ábra. Backend UML osztálydiagramm

Hitelesítés és  
jogosultságkeze-  
lés!! jwt token

## ORM használata

Az adatbázissal való kommunikáció során objektum-relációs leképztést használunk (Entity Framework), azaz az adatok kezelése objektumok formájában történik. Így az adatok kezelése egyszerűbb, pl.: LINQ segítségével történő adatkezelés.

## RESTful API-k

A RESTful API (Representational State Transfer) az interneten keresztüli kommunikációt egyszerűsíti, HTTP protokollra alapozva. Az adatok továbbítását json formátumban végzi a végpontok között.

### 3.2.3. Fájrendszer felépítése

ezt megírni

### 3.2.4. Frontend – VUE.JS alapú webes felület

A frontend felelős a felhasználói élményért, a vizuális megjelenítésért és a backenddel való kommunikációért, amelyeket HTML, CSS és a Vue.js JavaScript keretrendszer segítségével valósít meg.

#### Komponens-alapú architektúra

A VUE.JS keretrendszer komponensek segítségével építi fel az egyes felhasználói felületi elemeket.

- Bejelentkezési oldal: Regisztráció és bejelentkezés kezelése.
- Főoldal: Naptár és feladatlisták megjelenítése, kártyák hozzáadása, törlése, áthelyezése.
- Napi nézet: A nap részleteinek megjelenítése, feladatok állapotának módosítása.
- Több naptár kezelése: Naptárak létrehozása, csatlakozás, importálás, törlés.

#### API kommunikáció

Axios API használata a backend REST API végpontok eléréséhez.

### 3.2.5. Fájrendszer felépítése

esetleg Integráció és Kommunikáció section (de eléggé le van írva a fentebbi pontokban, max egy összefoglaló jöhet ide)

ezt megírni

## 4. fejezet

### Összegzés