

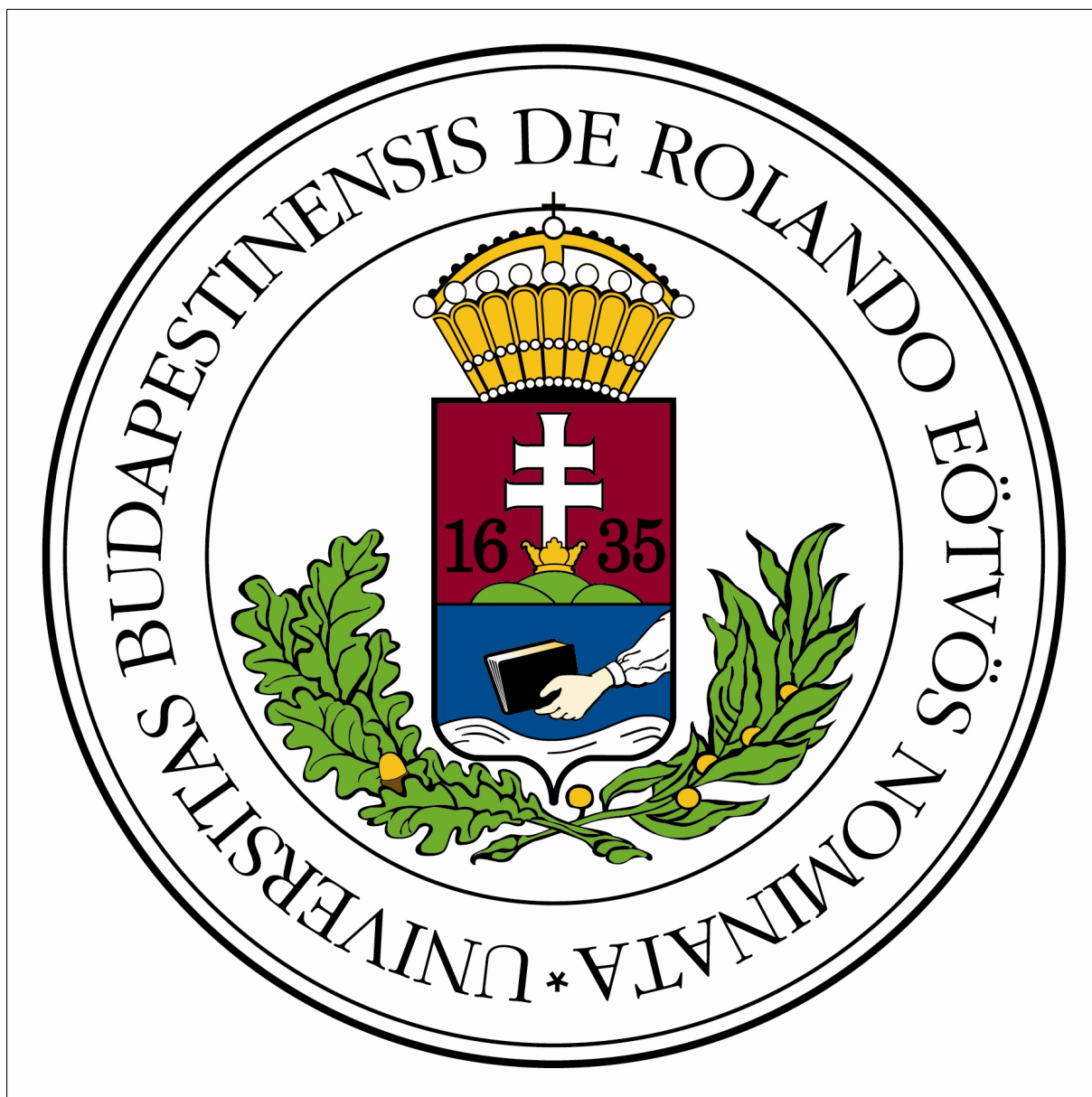
# **Szakedolgozat**

CompOffice - Kurzus és pénzügyi adminisztráció

Osztopáni Kristóf

2024-03-20

## Szakdolgozat



EÖTVÖS LORÁND TUDOMÁNYEGYETEM Informatikai Kar Média- és Oktatás informatikai Tanszék

CompOffice - Kurzus és pénzügyi adminisztráció

Témavezető: Dr. Menyhárt László Gábor, egyetemi adjunktus

Szerző: Osztopáni Kristóf, Programtervező informatikus BSC

# 1. Bevezetés

## 1.1 A dolgozat felépítése

A Szakdolgozat több szekcióra van osztva. Ezeket a címekben és alcímekben megjelenő első szám jelöli.

A szakdolgozat megírása még folyamatban van, így egyes részek hiányosak lehetnek.

Egyes szekciók tartalma:

- **1. Bevezetés:** Bemutatom a tervezett alkalmazást nagy vonalakban, szakdolgozat feléptését, az alkalmazás célközönségét valamint a szakdolgozat motivációját.
- **2. Felhasználói dokumentáció:** Az alkalmazás használatát mutatom be, a felhasználói felületet, a menüpontokat, az alkalmazás használatának lépéseit.
- **3. Fejlesztői dokumentáció:** Az alkalmazás fejlesztésének lépéseit mutatom be, a fejlesztés során használt technológiákat, a fejlesztési környezetet, a fejlesztési lépéseket valamint a kész arhitektúrát.

## 1.2 Témabejelentő

Megjegyzés: A témabejelentőm, nem a szó szerinti leadott verzió, de tartalmilag ekvivalens

A szakdolgozatom témája egy fullstack alkalmazás fejlesztése, amely egy cég belső adminisztrációját segíti, továbbá a fejlesztést és későbbi támogatást segítő technológiák felhasználása.

Az alkalmazás célja, hogy gyorsítsa az adminisztrációs folyamatokat, valamint a pénzügyi adatok kezelését. Ezek a folyamatok:

- A modulok(A legkisebb önmagukban is értékesíthető kurzusok) regisztrálása
- A termékek(Több modult tartalmazhat) regisztrálása
- Projektek (Egy kurzus amit adott időpontban tartunk és megrendelhető) regisztrálása
- A projektekre érkező rendelések céges információinak / résztvevők adatainak / a kurzust tartó oktatók adataink rögzítése, ellenőrzése
- A projektekhez tartozó pénzügyi adatok kezelése
- Általános pénzügyi adatok kezelése

Az alkalmazás érzékeny adatokat kezel így a biztonságra nagy hangsúlyt fektetek. Az alkalmazásban a Microsoft MFA autentikáció és Azure AD alapján a teljes céges okosizstémában is használt biztonságos bejelentkezés lesz jelen. A biztonság elvárások miatt kiemelten fontos a hogy a lehető legkevesebb

bug jusson csak ki a kódbázisból a kész termékbe. Ehhez teljes e2e típusbiztos rendszert alakítottunk ki, ami azt jelenti hogy a kommunikáció az Adatbázistól a szerverhez és a szervertől a klienshez végig típusellenőrizve lehet, amely reményeim szerint csökkenti az előforduló hibák számát. Továbbá a célom egy olyan tesztelési keretrendszer felépítése amely képes a teljes kódbázist a funkcionális követelmények alapján tesztelni.

Az alkalmazás fejlesztéséhez sok automatikus rendszer is tartozik, a fejlesztői élmény növelése érdekében. Ezeket a fejlesztői dokumentációban részletesen is bemutatom.

A felhasználók számára elérhető lesz egy kalendár és egy dashboard, ahol a tervezet és elmúlt megrendeléseket követhetik nyomon, valamint küldhetnek ki emlékeztetőket és összefoglalókat az oktatóknak az elkövetkező kurzusokról. Továbbá egy automatikus rendszer emlékezteti az adminisztrátorokat a közelgő eseményekről, hiányzó bejegyzésekről.

Az alkalmazás egy fullstack PWA lesz, amely ahol a szerver és kliens oldalt én valósítottam meg, az adatbázis pedig egy már létező adatbázisra épül, így ennek módosítása nem része a dolgozatomnak.

Ezt az alkalmazást 3 éve egy LowCode keretrendszerben már megvalósítottam, de ez a megoldás nem volt skálázható, így most egy teljesen új megközelítéssel szeretném megvalósítani.

## 3. Fejlesztői Dokumentáció

### 3.1 Áttekintés

#### 3.1.1 Élő dokumentáció

A dokumentáció nagy része “élő” dokumentációként lett tervezve, így a dokumentáció nagy része integrált a kódbázisban.

Erre egy példa hogy az adatbázis modelje a `prisma/schema.prisma` file amely leírja a konkrét adatbázis táblákat és kapcsolatokat, képes legenerálni a hozzájuk tartozó entity-relationship diagramot és leírásokat is. > Megjegyzés: A dokumentáció így “élő” hiszen csak a kód részét kell változtatnom és emiatt frissül a dokumentáció része is a projektnek.

A schema így néz ki:

```
/// @namespace Instructor
/// Oktatók cégének adatai
model instructor_company {
  /// Elsődleges kulcs
  ///
  /// @format uuid
  id          Int          @id(map: "PK__instruct__3213E83FA0B8D54C")

  /// Cég neve
  name        String       @unique(map: "UQ__instruct__72E12F1BD9C014F0") @db.VarChar
  instructor  instructor[]
}
```

És ebből Markdown formátumú dokumentáció készül. A kódbázis minden dokumentációja Markdown formátumú, pontosabban Github Favoured Markdown formátumú, és elérhető a [githubon](#). Ebből PDF is generálható.

#### 3.1.2 Dokumentáció előállítás

A dokumentációs fájlok:

- **/wiki/**: Tartalmazza az összes dokumentációt.
- **/wiki/wiki.md**: Az összevont dokumentáció

- **/wiki/docs.md**: A dokumentáció kialakításához használt gyökér fájl.
- **/wiki/dev/**: Tartalmazza a dokumentáció élő részét, ezek a generált dokumentációk.
- **/wiki/content/**: Tartalmazza a dokumentáció statikus részét

A Dokumentáció felépítése a kódbázisból a következőként történik:

- **1.**: Az adatbázis dokumentációjának legenerálása
- **2.**: A userstory-k dokumentációjának legenerálása
- **3.**: A root fájlt használva egy md-be összevonja a dokumentáció részeit.
- **4.**: Az összevont dokumentációból készítjük a PDF-et.

A megfelelő környezetben az `npm run docs` parancs ezeket hajtja végre

### 3.1.3 A használt eszközök

- Az adatbázis-t [Prisma](https://github.com/samchon/prisma-markdown) Orm-el managelem, és ehhez egy (prisma-markdown)[https://github.com/samchon/prisma-markdown] generátorral készítem a dokumentációt.
- A markdown dokumentációban található [Mermaid](#) formátumú diagrammokat a [mermaid-filter](#) cseréli le képre pdf esetén.
- A md-merge-lés egy saját python eszköz amely megtalálható a `/tools/md_merge.py` fileban
- A user-story-k a `/features` mappában találhatóak `.feature` kiterjesztéssel, ezek egy speciális Markdown formátumot követnek kifejezetten User-story-kra tervezve, a formátumot (Gherkin)[https://cucumber.io/docs/gherkin/reference/] nek hívják.
- A `.features` kiterjesztésű user story-k közönséges Markdown-ná alakítását a [gherkin2markdown](#) program végzi.
- A végső PDF a [Pandoc](#) programmal készül az [eisvogel](#) template-t használva.

## 3.2 Nem funkcionális követelmények

### 3.2.1 Hatékonyság

- A gyors válaszidő fontos, de nem prioritás.
- Az alkalmazás ahol tehet nagy mennyiségű adatokat pager-rel jeleníttmeg
- Az adat módosítás optimisztikus módon történik, azaz ha a kliens oldalon el lett fogadva akkor a kliens folytatja a működését még mielőtt a szerver válaszolna, a ritka alkalmankét amikor a szerver válasza, hogy helytelen a kliens oldali változás akkor a kliens oldalán is visszaállítja a változást.
- Amennyiben szükséges egy Gyorsítottár réteg is bevezethető, az adatbázis túlterheltségének csökkentésére. Feltehetően mivel az alkalmazást egyszerre csak 5-10 ember használja, ezért nem lesz szükség rá.

### 3.2.2 Biztonság

- Minden kommunikáció MS MFA Azure AD mögött történik.
- Az ismert támadási technikák (SQL Injection, XSS, stb.) nem működnek az alkalmazásban.

### 3.2.3 Megbízhatóság

- Az alkalmazás tRPC-t használ a kliens oldalon, de REST stílusú "publikus" api-t is biztosított. nem vezethet hibás viselkedéshez.
- Minden fontos esemény az alkalmazásban naplózásra kerül.

### 3.2.4 Felhasználói felület

- A felület ergonomikus és könnyen kezelhető.
- A felület kényelmes új felhasználók számára és lehetőséget ad a gyorsabb munkára a tapasztalt felhasználók számára.
- Az alkalmazásnak reszponzívnak kell lennie, azaz minden eszközön jól használhatónak kell lennie, habár az alkalmazás irodai környezetben való munkára (Számítógépen) készül.

### 3.2.5 Platformfüggetlenség

- Az alkalmazás kliens oldala Chromium alapú böngészőkön és Firefoxon működik.
- A server oldal tetszőleges linux szerveren működik.

### 3.2.6 Szükséges erőforrások

- **Hardveres erőforrások:**

- Az alkalmazás kliens oldala egy átlagos számítógépen is futtatható.
- Az alkalmazás szerver oldala egy átlagos szerveren is futtatható.
- Fejlesztés alatt a gyors fejlesztéshez egy erősebb számítógép ajánlott 8+ GB rammal és 4+ magos processzorral.

- **Szoftveres erőforrások:**

- Az alkalmazás kliens oldala egy átlagos böngészőn futtatható.
- Az alkalmazás szerver oldala egy átlagos linux szerveren futtatható.
- Fejlesztés alatt a package.json-ben devDependencies-ben megadott szoftverek szükségesek, valamint egyéb csomagok ezek:
  - git, nodejs, npm, pandoc, go, python3, texlive latex csomagok
  - fejlesztői környezet, lehetőleg ami integrálja a formázó és lintelő eszközöket pl.: VSCode vagy nvim
  - Adatbázis szerver
- **Üzemeltetés és karbantartás:**
  - Az karbantartáshoz egy naplózó rendszer beüzemelése előnyös de nem szükséges
  - Az üzemeltetés a számítógép költségén kívül nem igényel egyéb költséget.
  - A karbantartás egyszerűsítése a dolgozatom egyik fő célja. Reményeim szerint az erre fordított költségek minimálisak lesznek.

### 3.2.7 Megvalósítás és használt technológiák

TBD



### 3.3 Funkcionális követelmények

A rendszernek a következő funkciókat, reakciókat kell tudnia biztosítani.

Megjegyzés: Az alkalmazás belső használatra készül így a nyílt webről kereséssel való felfedezéssel nem kell foglalkozni.

#### 3.3.1 Általános funkciók

##### A weboldal megnyitása után:

- A felhasználók bejelentkezhetnek, akkor és csak akkor ha Azure AD-ben megengedett felhasználók.
- Ha a felhasználónak még van aktív bejelentkezése, akkor nem szükséges újra bejelentkeznie.
- Egy felhasználóhoz több aktív munkamenet is tartozhat.

##### A weboldal bezárása után:

- Az aktív munkamenet lezárul.

##### A weboldal minden "oldalán":

Megjegyzés: Az alkalmazás egy PWA tehát nem a hagyományos értelemben vett oldalakról beszélünk

- A felhasználók a menüben tudnak navigálni.
- A felhasználók vissza tudnak lépni a főoldalra.
- A felhasználók tudnak kijelentkezni.
- A felhasználók vissza tudnak lépni a jelenlegi munkamenetben a régebbi oldalra.
- A felhasználó meg tudja nyitni a kalendárt
- Minden listázó oldal frissíthető.
- Minden szerkesztő oldal menthető.
- Ha az éppen szerkesztett rekordot valaki más módosította, akkor a felhasználót értesíteni kell.

### 3.3.2 Felhasználók

**3.3.2.1 Jogosultsági szintek** A rendszerben a következő jogosultsági szinteket különböztetjük meg:

- **Super:** Az alkalmazás teljes körű használatára jogosult, admin felhasználó.
- **Test:** Az alkalmazás teljes körű használatára jogosult, fejlesztő és tesztelő felhasználó.
- **Finance:** A cég pénzügyi adatainak kezelésére jogosult felhasználó.
- **Operator:** A cég üzemi adatainak kezelésére jogosult felhasználó.
- **Coordinator:** (Legacy) Az Operátor felhasználókkal ekvivalens jogosultságú felhasználó.

Kezdetben különbözőek jogosultságokkal rendelkeztek de a kialakulás alatt a kért változások miatt a végeredményben ugyanazok a jogosultságokkal rendelkeznek, de különböző embereknek vannak kiosztva.

- **Instructor:** A cég egy belső oktatója, főként a felvitt project adatok ellenőrzésére jogosult.
- **Support:** A cég egy külsős oktatója, a jogosultságai az Instructor felhasználóéval jelenleg megegyeznek.
- **Bot:** Az automatikus rendszerekhez szükséges legszűkebb jogosultságú felhasználó / service account.

**3.3.2.2 Jogosultsági körök** Az alkalmazásban elvégezhető műveleteket a következő csoportokba lehet beosztani:

- **Email:** Az emlékeztető és értesítő levelek küldése
- **Info:** A Dashboard és a Kalendár megtekintése
- **Order:** A rendelések kezelése és új projectek kiírása
- **Finance:** A pénzügyi adatok kezelése, a banki backlogok kezelése
- **General:** Általános adatok kezelése

Ezek a csoportok tartalma megtekinthető az Adatbázis leírásában

Továbbá a jogosultságok megvalósításakkor a jogosultsági körök minden oldalra külön felülírhatóak lesznek.

**3.3.2.3 Műveletek** A jogosultsági körök által meghatározott műveletek a következők lehetnek:

- **Read(R)**: Adatok megtekintése
- **Write(W)**: Adatok módosítása
- **Restricted(-)**: Az adott adatokhoz nincs hozzáférés

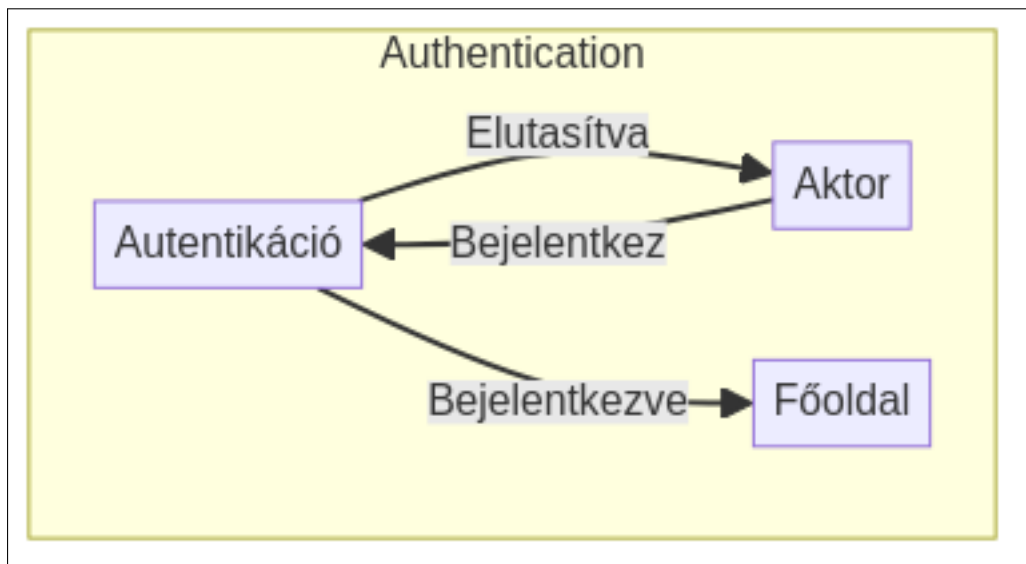
Jogosultságok Körök	Super	Test	Finance	Operator	Coordinator	Instructor	Support
Email	W	W	-	W	W	-	-
Info	W	W	W	W	W	R	R
Order	W	W	W	W	W	R	R
Finance	W	W	W	-	-	-	-
General	W	W	W	W	W	R	R

**3.3.2.4 Vendég jogosultságok** Minden adat érzékeny így nem engedünk vendég felhasználót, kizárólag a Céges Azure AD-ban meghatározott felhasználók látogathatják az oldalt

### 3.3.3 Use-case diagramok

#### 3.3.3.1 As An Un Authenticated User.

Bejelentkezési oldal látogatásakor a felhasználó, ha még nem jelentkezett be, akkor a bejelentkezési oldalra irányítódik át, ahonnan sikeres bejelentkezés után a főoldalra irányítódik át, sikertelen bejelentkezés esetén a bejelentkezési oldalon marad.

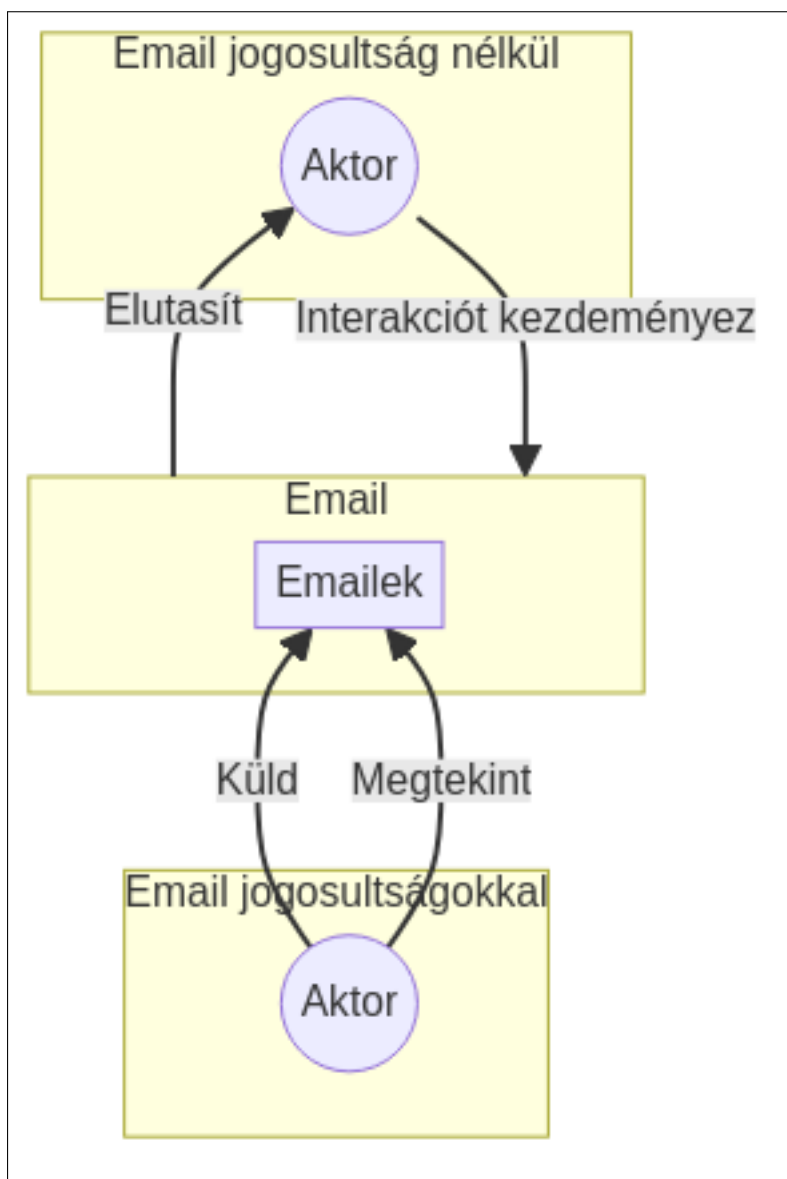


### 3.3.3.2 As An Authenticated User With Email privileges.

Amennyiben egy felhasználó rendelkezik az email(W) jogosultsággal akkor az alábbi funkciókat tudja elérni:

- Küldhet emailt
- Megtekintheti az emailküldés státuszát

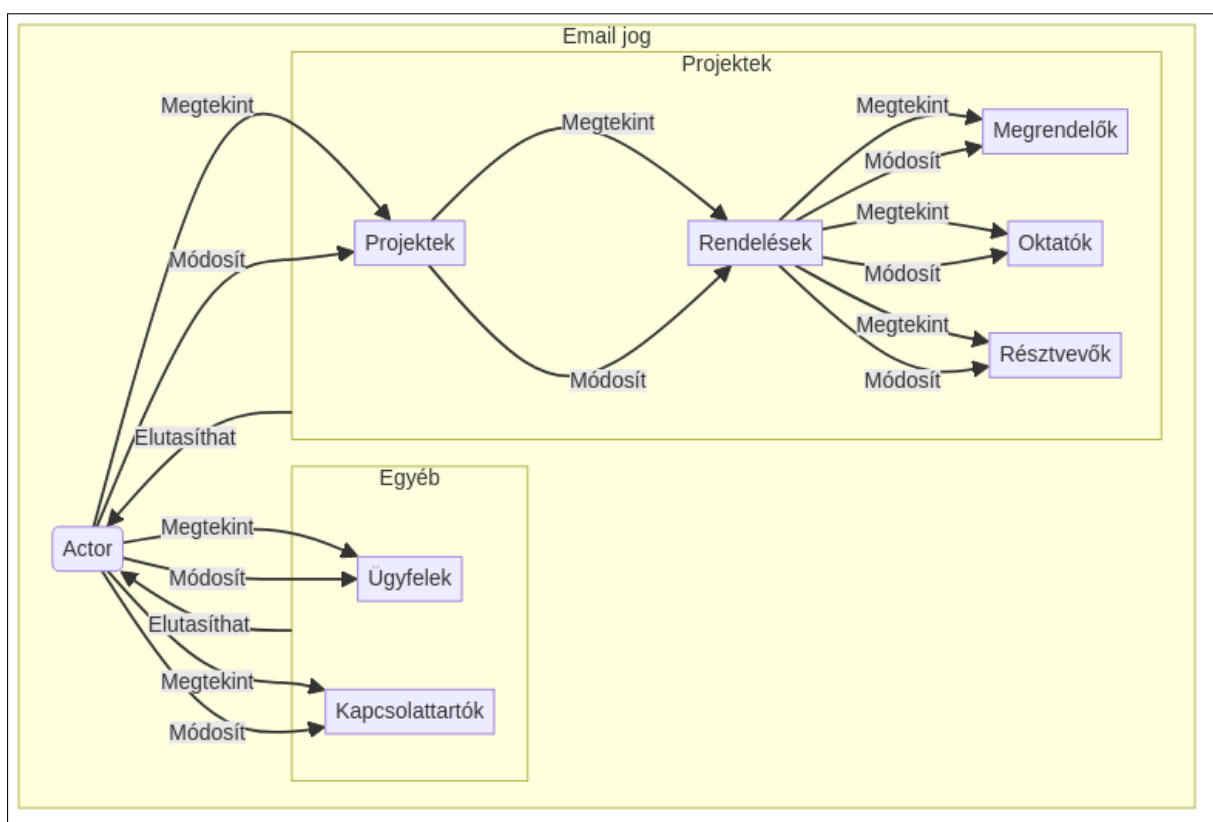
Ha csak olvasó jogosultsága van akkor csak megtekintésre nyithatja meg, amennyiben nincs jogosultsága, akkor nem tudja megnyitni az oldalt, a menüfülön nem is lesz látható az opció.



### 3.3.3.3 As An Authenticated User With Order privileges.

A felhasználó meg tudja tekinteni a futó projekteket és annak minden részletét beleértve a megrendelőket és a megrendelők résztvevőit, valamint a projektet tartó oktatókat. Ezen felül képes az ügyfeleket és kapcsolattartókat adatait módosítani.

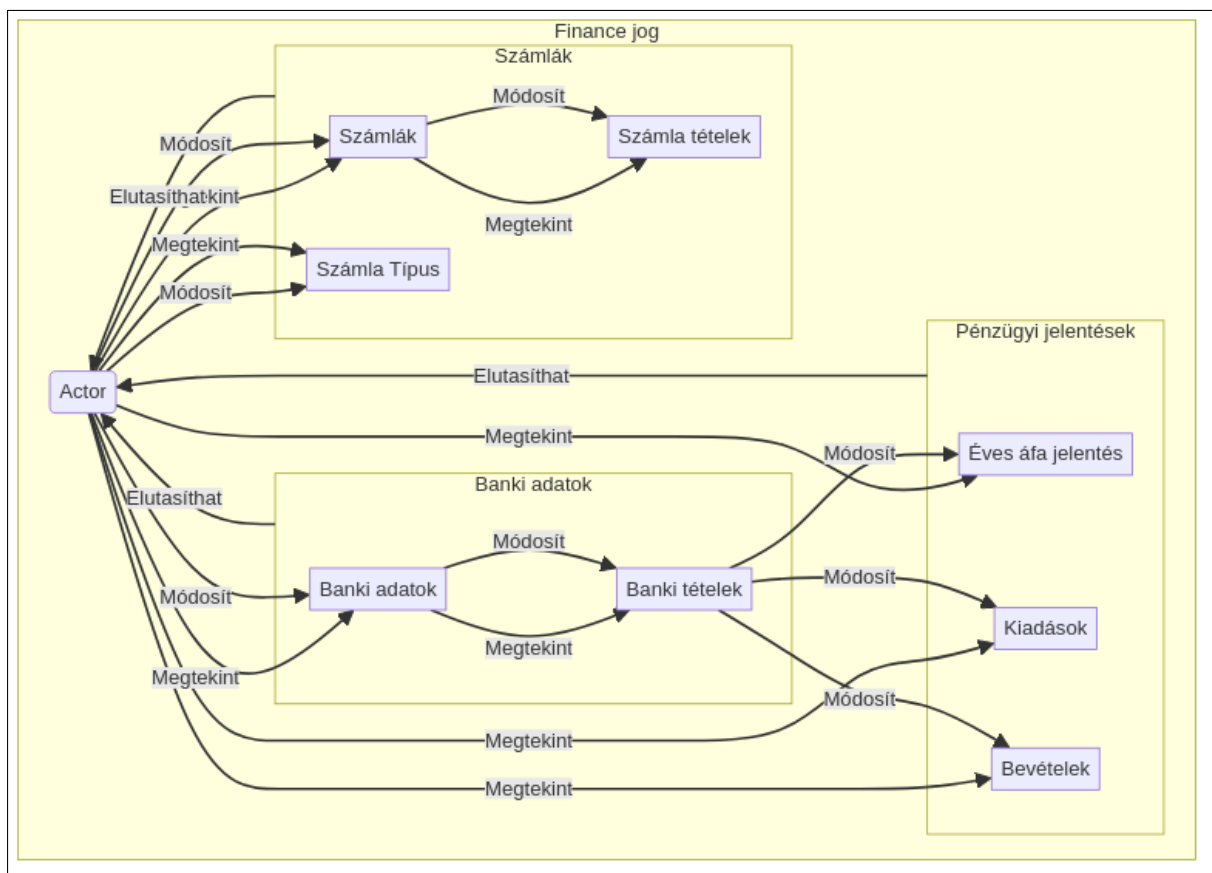
- Módosítás csak akkor ha van W jogosultsága
- Megtekintés csak akkor ha Olvasás(vagy írás) jogosultsága van.
- Ha nincs jogosultsága akkor nem tudja megnyitni az oldalt, a menüfűlön nem is lesz látható az opció, ha a felhasználó mégis oda navigál akkor az oldal elutasítja a kérést.



### 3.3.3.4 As An Authenticated User With Finance privileges.

A felhasználó hozzáfér a pénzügyi adatokhoz és azokat módosíthatja is. A pénzügyi adatok közé tartozik a költségek, bevételek, banki adatok, számlák és számla tételek, valamint az áfa jelentések.

- Módosítás csak akkor ha van W jogosultsága
- Megtekintés csak akkor ha Olvasás(vagy írás) jogosultsága van.
- Ha nincs jogosultsága akkor nem tudja megnyitni az oldalt, a menüfülön nem is lesz látható az opció, ha a felhasználó mégis oda navigál akkor az oldal elutasítja a kérést.

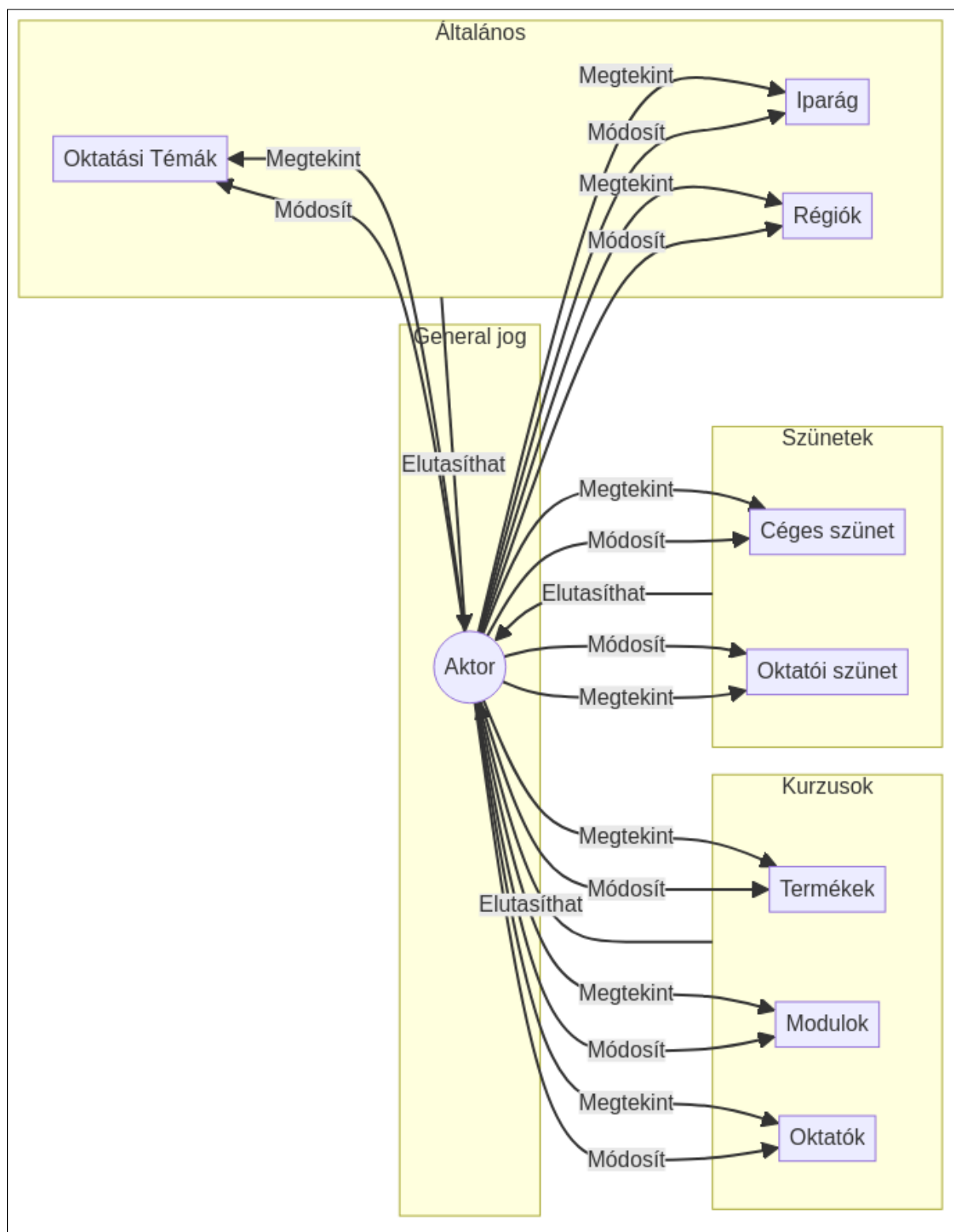


**3.3.3.5 As An Authenticated User With General privileges.**

A felhasználó hozzáfér az általános adatokhoz. Az általános adatok közé tartozik a modulok, termékek, oktatók, céges szünetek, oktatási témák, iparágak és régiók.

- Módosítás csak akkor ha van W jogosultsága
- Megtekintés csak akkor ha Olvasás(vagy írás) jogosultsága van.
- Ha nincs jogosultsága akkor nem tudja megnyitni az oldalt, a menüfűlön nem is lesz látható az opció, ha a felhasználó mégis oda navigál akkor az oldal elutasítja a kérést.



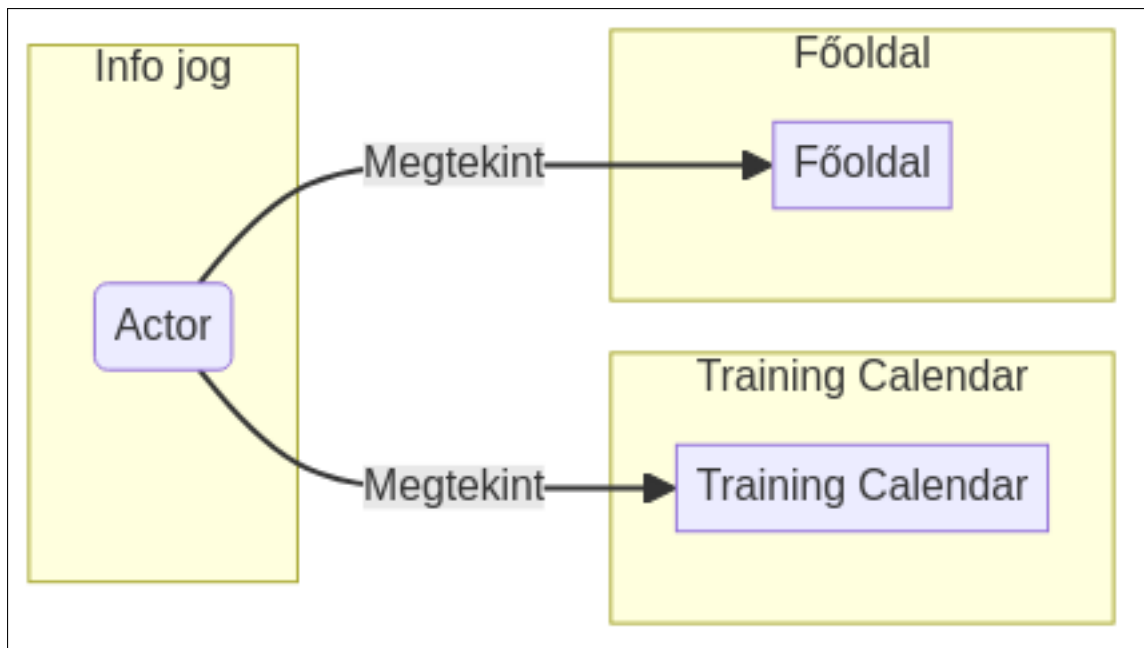


**As An Authenticated User With Info privileges.**

Minden felhasználó rendelkezik az Info jogosultsággal.

Az Info jogosultság csak megtekintésre jogosítja fel a felhasználót.

A Felhasználó megtekintheti a Training Calendar és a Főoldalon a közeljövőben esedékes eseményeket.



### 3.3.4 User-Story-k

**Itt vannak a user storyk::**

#### **Sanity Check**

**I am a user**

*Given* I have 42 cucumbers in my belly

*Then* The result is 42.