

DOKUMENTÁCIÓ:

Tudásadatbázis

Ács Koppány (Adatbázis alapú rendszerek gyakorlat)

Feladat Leírása / Program funkciói:

A feladat egy a wikipédiához hasonló tudásadatbázis létrehozás

Az alkalmazás tud:

- Bejelentkezés után lehet a táblákból adatot lekérdezni vagy felvinni/módosítani/törölni adatokat.

Követelmény	Státusz	Felelős	Dátum
1. Cikk kezelése			
2. Témakörök és altémakörök kezelése			
3. Felhasználók bejelentkeztetése			
4. Lektorok kezelése (a lektorok is felhasználók, de speciális adatokat, mint tudományos fokozat, intézet, szakterület is el kell tárolni)			
5. Szerző által létrehozott cikkek száma.			
6. Szerző hozzárendelése a cikkekhez (cikk létrehozó lesz a szerző)			
7. Különböző nyelvek kezelése (egy adott cikk több nyelven, különböző tartalommal jelenhet meg)			
8. Lektorokhoz nyelvek és nyelvi szintek rendelése			
9. Hibák bejelentése			
10. Legtöbbet módosított cikkek listája			

Megvalósítás:

Az alkalmazás java nyelven került fejlesztésre, illetve Oracle Database-t használ az adatok tárolására.

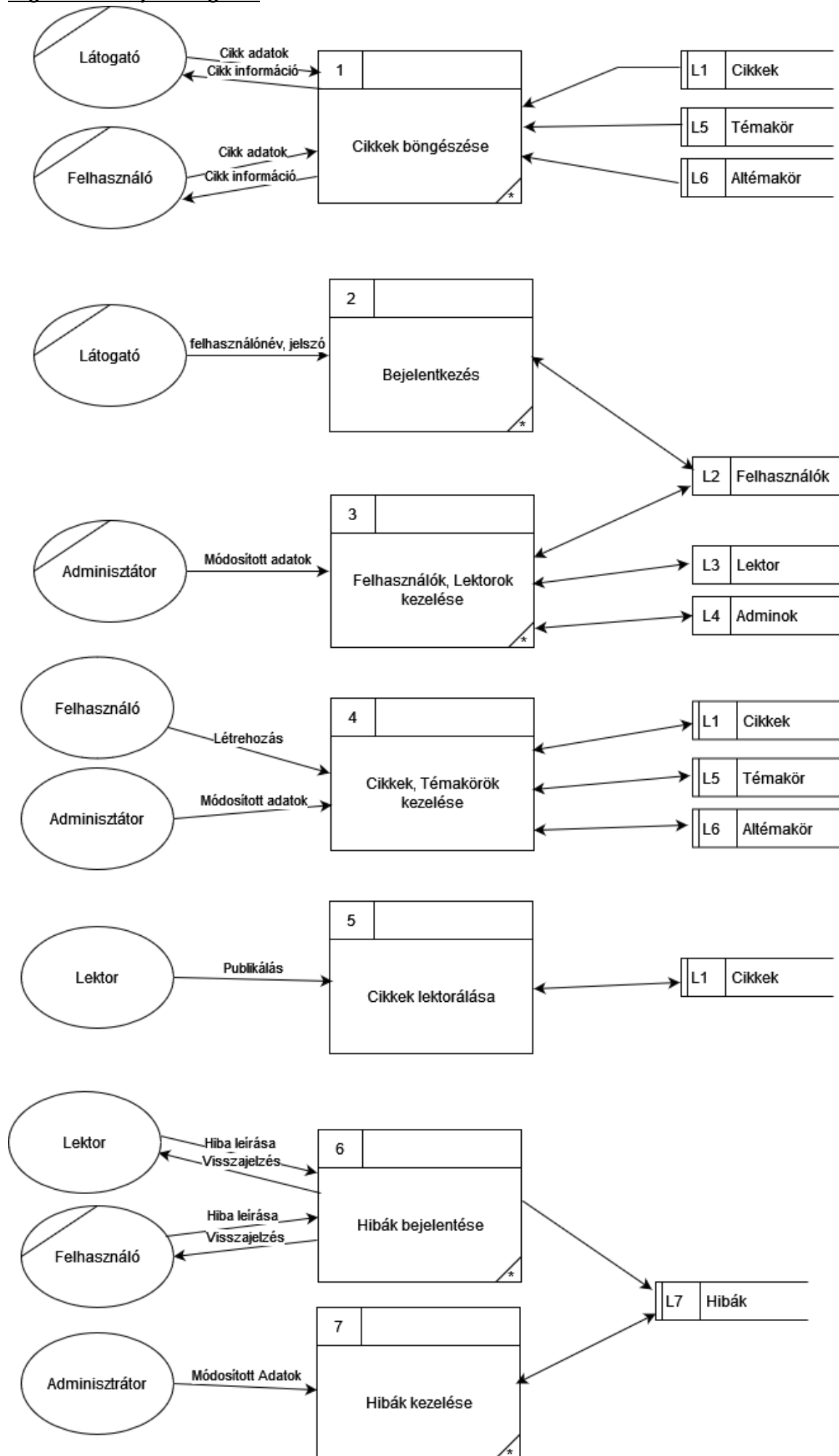
Munkafelosztás:

Percze Gábor: Logikai és Fizikai adatfolyam diagram

Ács Koppány: Szerep-Funkció Mátrix, Egyed-kapcsolat diagram, Egyedmodell, adatbázis létrehozó script, adat feltöltő script, alkalmazás.

Adatfolyam diagram:

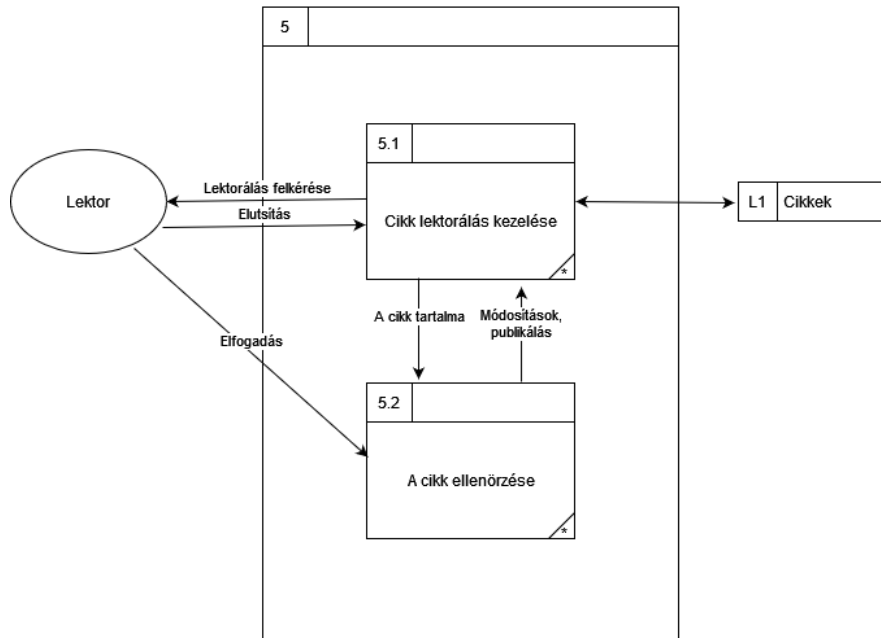
Logikai adatfolyam diagram:



Első lépésként felvettem a környezeti elemeket, majd felvettem a rendszer folyamatait. Meghatároztam azt is, hogy milyen adatokat kell eltárolnunk a rendszerben. Ezután berajzoljuk az adatfolyamokat. A rendszer működését 7 folyamatra bontottuk fel, ezek a cikkek böngészése, bejelentkezés, Felhasználók és Lektorok kezelése, Cikkek és Témakörök kezelése, Cikkek létrehozása, illetve a Hibák bejelentése és kezelése.

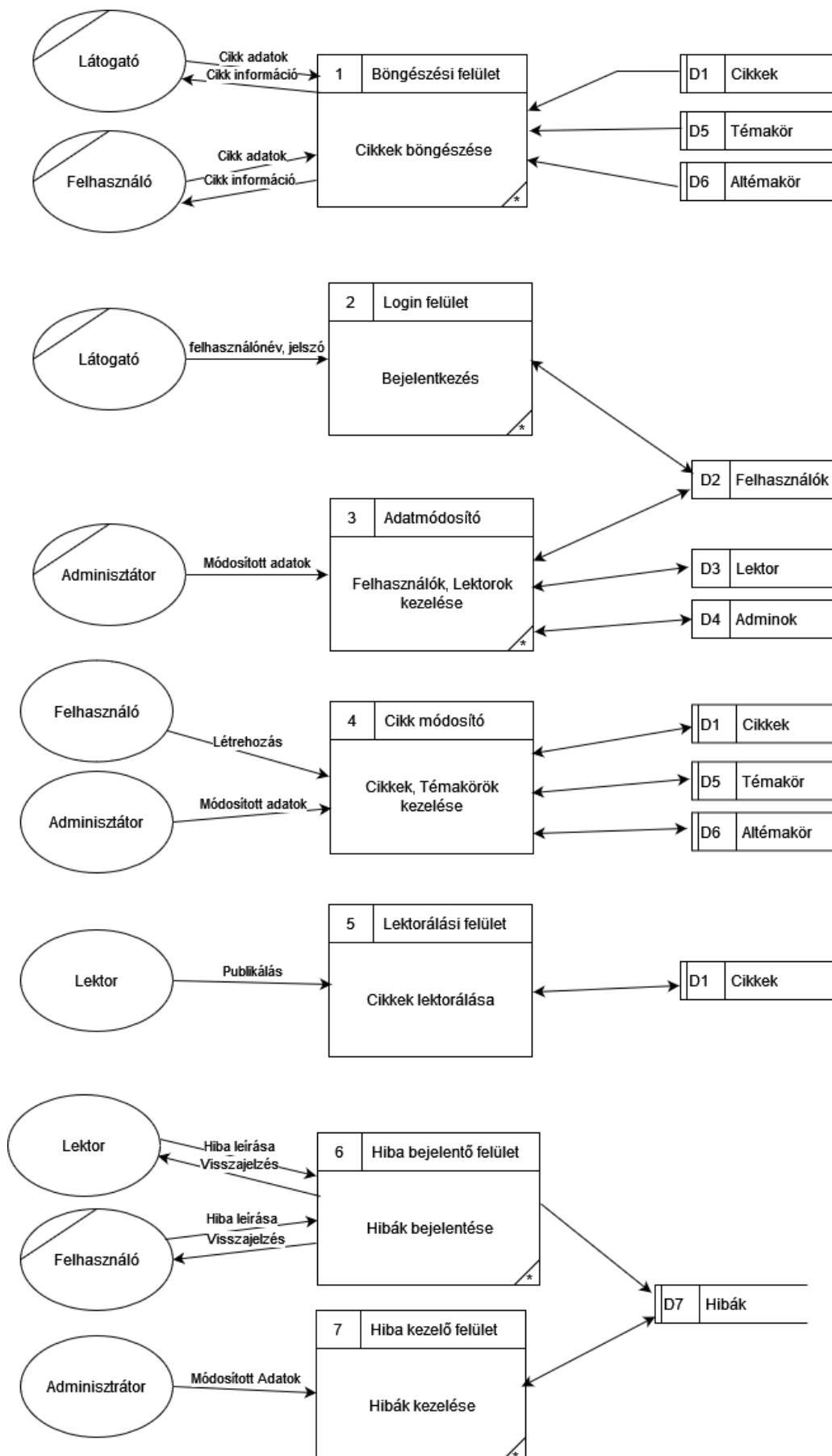
A cikk lektorálása nem végleges folyamatként vettük fel annak okán hogy ez összetettebb folyamat.

2. szintű adatfolyam diagramja :

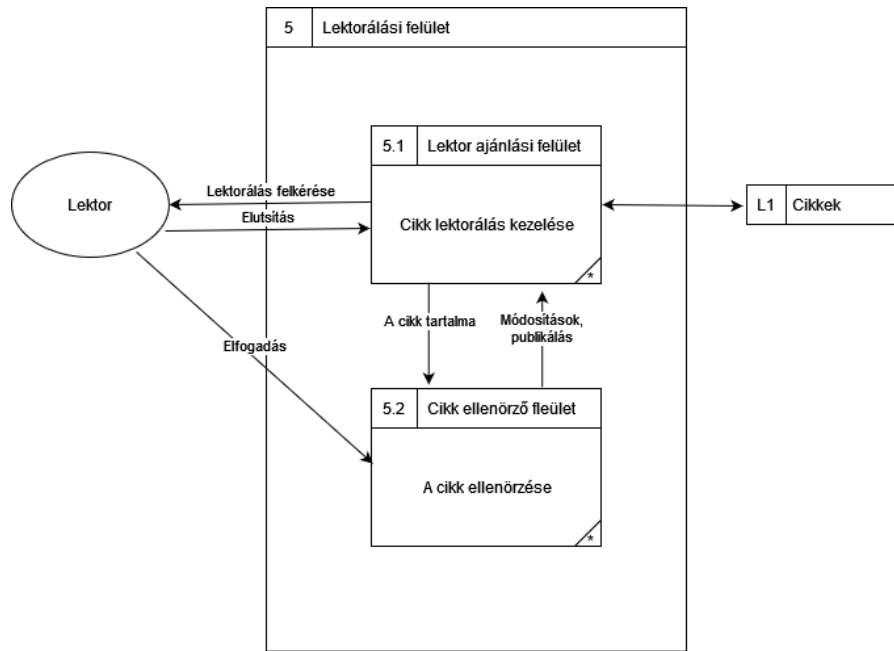


Fizikai adatfolyam diagram:

A fizikai adatfolyam diagrammokban jelöljük az adattárak típusát, azt, hogy az egyes folyamatok hol zajlanak (szervezeti egység), valamint ha rendszerben lenne fizikai anyagmozgás akkor azt is, viszont fizikai anyagmozgás nincs.



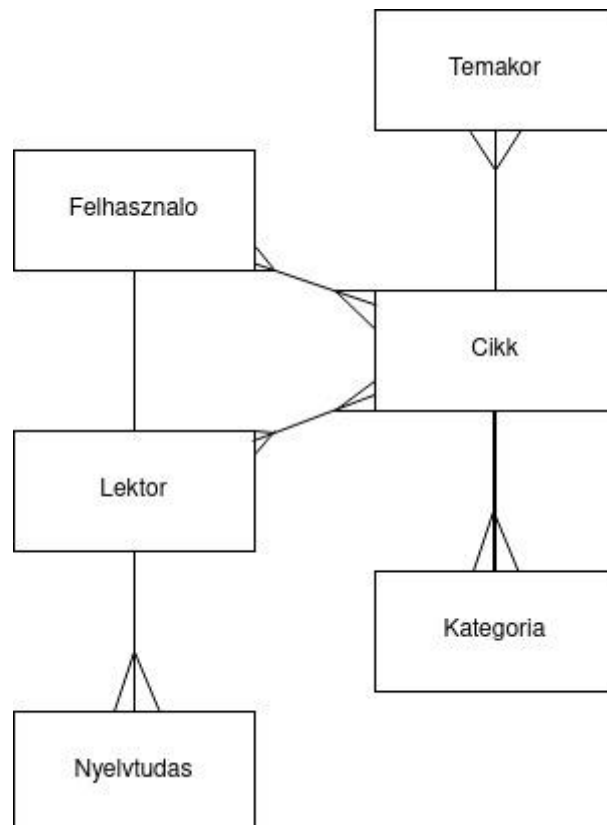
2. szintű fizikai adatfolyam diagramja :



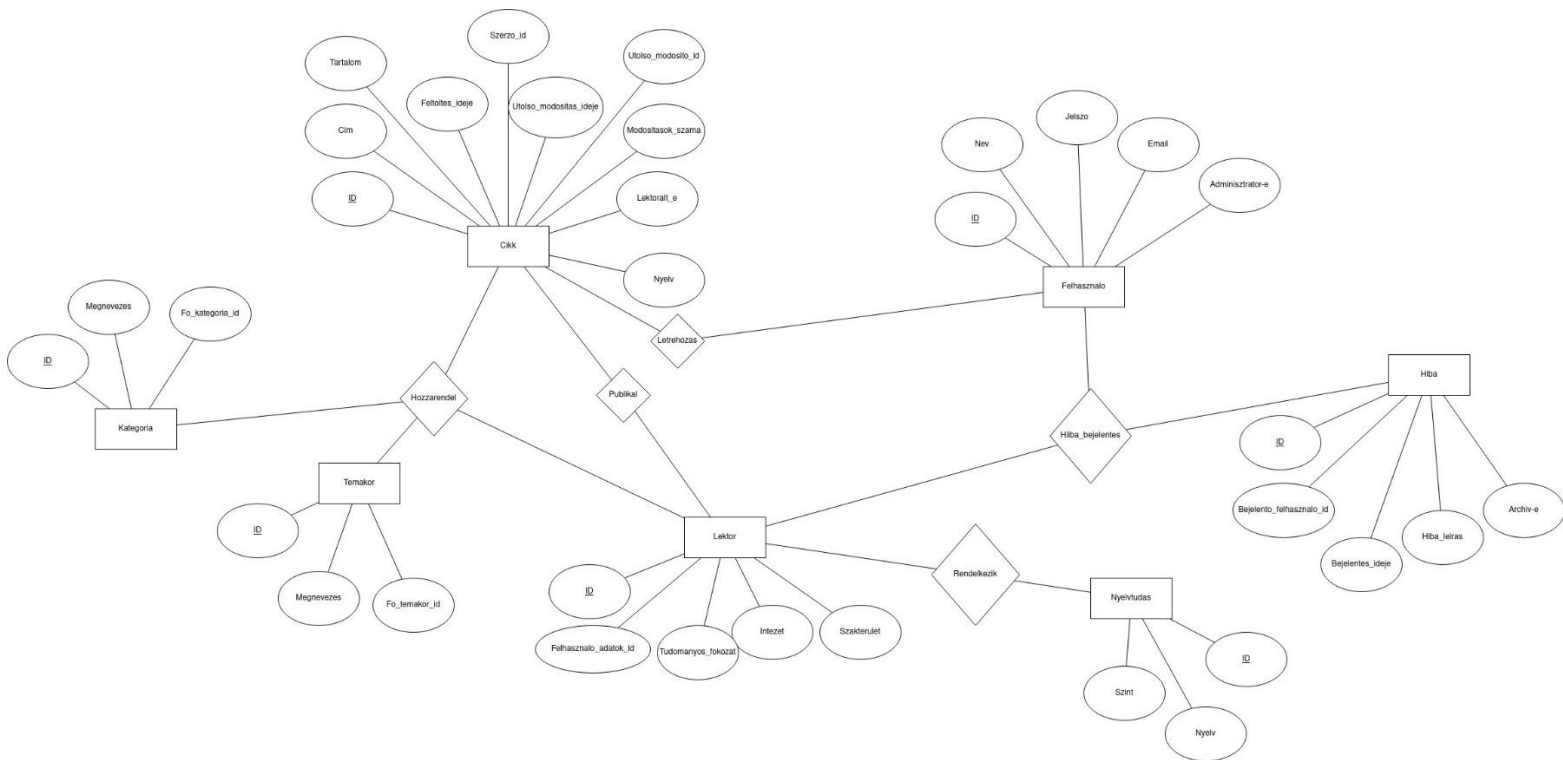
Szerep-Funkció Mátrix:

	Regisztráció	Bejelentkezés	Cikk létrehozása	Cikkek böngészése	Cikk lektorálása	Cikk jóváhagyása és megjelentetés	Témakörök kezelése	Lektorok kezelése	Kategóriák kezelése	Kulcsszavak rendelése	Szerző hozzárendelése	Felhasználók kezelése	Hiba bejelentés	Hiba kezelése
Szerep-funkció mátrix														
Felhasználó		X	X	X									X	
Lektor		X	X	X	X	X	X			X	X			
Adminisztrátor		X	X	X		X	X	X	X	X	X	X		X
Vendég(regisztráció nélküli)	X			X									X	

Egyedmodell:



EK Diagram:



EK LEKÉPZÉS:

Cikk(ID, cím, tartalom, szerző_id, feltöltés_ideje, utolsó_módosítás_ideje, utolsó_módosító_id, módosítások_száma, lektoralt-e, nyelv)

Kategória(ID, megnevezés, fő_kategória_id)

Témakör(ID, megnevezés, fő_témakör_id)

Felhasználó(ID, név, jelszó, email, adminisztrátor-e)

Lektor(ID, felhasználó_adatok_id, tudományos_fokozat, intézet, szakterület)

Nyelvtudás(ID, nyelv, szint, lektor_id)

Hiba(ID, bejelentő_felhasználó_id, bejelentés_ideje, hiba_leírás, archiv-e)

Cikk_kategória(ID, cikk_id, kategória_id)

Cikk_témakör(ID, cikk_id, témakör_id)

Normalizálás:

1NF: Minden relációséma 1NF-ben van, mivel nincs benne összetett attribútum.

2NF: Minden relációséma 2NF-ben van, mivel a sémákban minden másodlagos attribútum teljesen függ bármely kulcstól.

3NF: Minden relációséma 3NF-ben van, mert minden másodlagos attribútuma közvetlenül függ bármely kulcstól.

Egyed-Esemény mátrix:

[illegible]

Összetett SQL lekérdezések:

```
SELECT Lektor.id, Felhasznalo.nev, Lektor.tudomanyos_fokozat,
Lektor.intezet, Lektor.szakterulet, lektor.FELHASZNALO_ADATOK_ID FROM
Lektor JOIN Felhasznalo ON Lektor.felhasznalo_adatok_id = Felhasznalo.id;
```

```
SELECT n.id, f.nev, n.nyelv, n.szint, n.lektor_id FROM Nyelvtudas n JOIN
Lektor l ON n.lektor_id = l.id JOIN Felhasznalo f ON
l.felhasznalo_adatok_id = f.id;
```

```
SELECT hiba.id, hiba.bejelentes_ideje, hiba.hiba_leiras,
hiba.archiv_e, hiba.BEJELENTO_FELHASZNALO_ID, felhasznalo.nev FROM hiba LEFT
JOIN felhasznalo ON hiba.bejelento_felhasznalo_id = felhasznalo.id;
```

```
SELECT cikk.cim, kategoria.megnevezes, temakor.megnevezes
FROM Cikk cikk
LEFT JOIN Cikk_kategoria ck ON cikk.id = ck.cikk_id
LEFT JOIN Kategoria kategoria ON ck.kategoria_id = kategoria.id
LEFT JOIN Cikk_temakor ct ON cikk.id = ct.cikk_id
LEFT JOIN Temakor temakor ON ct.temakor_id = temakor.id;
```

```
SELECT f.nev, COUNT(*)
FROM cikk c
JOIN felhasznalo f ON c.szerzo_id = f.id
GROUP BY f.nev;
```

```
SELECT f.nev, COUNT(*)
FROM nyelvtudas n
JOIN lektor l ON n.lektor_id = l.id
JOIN felhasznalo f ON l.felhasznalo_adatok_id = f.id
GROUP BY f.nev;
```

```
select f.email, count(*) from hiba h
LEFT JOIN Felhasznalo f ON f.id = h.bejelento_felhasznalo_id
group by f.email;
```

Triggerek:

```
create or replace TRIGGER hibaTrig
BEFORE INSERT ON hiba
FOR EACH ROW
BEGIN
    :new.BEJELENTES_IDEJE := TO_CHAR(SYSDATE, 'YYYY-MM-DD');
END;
```

```
create or replace TRIGGER cikkTrig
BEFORE INSERT OR UPDATE ON cikk
FOR EACH ROW
BEGIN
    IF INSERTING THEN
        :new.FELTOLTES_IDEJE := TO_CHAR(SYSDATE, 'YYYY-MM-DD');
    ELSIF UPDATING THEN
        :new.UTOLSO_MODOSITAS_IDEJE := TO_CHAR(SYSDATE, 'YYYY-MM-DD');
        :new.modositasok_szama := :old.modositasok_szama+1;
    END IF;
END;
```

Alprogramok:

```
CREATE OR REPLACE PROCEDURE authenticate_user(
    p_email IN VARCHAR2,
    p_password IN VARCHAR2,
    p_user_id OUT NUMBER
) AS
BEGIN
    SELECT id INTO p_user_id FROM felhasznalo WHERE email = p_email AND
jelszo = p_password;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        p_user_id := -1; -- invalid credentials
END;
```