

KAPPA CSAPAT

Webtárhely, VPS, Levelező szolgáltatás-regisztráció

Csapattagok:

- Csupity Tamás - AGLU8T (Csapatvezető)
 - Gila Ferenc - GVPXRN
 - Mihály Marcell - FKNKDH

ADATBÁZIS ALAPÚ RENDSZEREK GYAKORLATA (IB152I)

Szerda: 14:00-16:00

2025. tavaszi szemeszter

A dokumentációt: Csupity Tamás készítette

Specifikáció:

Cél:

A projekt célja egy webalapú alkalmazás fejlesztése, amely lehetővé teszi a webtárhely, és domainek egyszerű és hatékony kezelését mind az ügyfelek, mind az adminisztrátorok számára. A rendszer automatizáltan biztosítja az számlázást és statisztikai elemzéseket.

A felhasználó regisztrálni tud majd bejelentkezni, előtű fizetni díjcsomagokra amelyek domain és webszerver szolgáltatások lehetnek. Megtudja nézni a gyakorta ismételt kérdéseket és az üzemeltetők válaszait. A rendszer számlázni fog az ügyfélnek aki ezeket ki tudja egyenlíteni. különböző érdekes statisztikákat tud megtekinteni amelyek a szolgáltató vagy a szolgáltatásokkal kapcsolatosak. Az admin felhasználót, számlát, díjcsomagot, tudástárat, statisztikát, domiant és webtárhelyet tud módosítani, hozzáadni és törölni.

Funkcionális követelmények

Alaprendszer funkciói:

- Felhasználókezelés: Regisztráció, bejelentkezés, felhasználói profilok kezelése, jogosultságok beállítása.
- Domainek kezelése
- Webtárhely kezelése
- Díjcsomagok (különböző tárhely és domain kapacitás)
- Számlakészítés.

Bővített funkciók:

- Tudástár (gyakran ismételt kérdések és válaszai a webhostinggal kapcsolatban)
- Számlák mentése havonta, számlák állapotváltozásainak kezelése az adatbázisba
- Bevételi statisztika, kimutatás készítés
- Korlátozás, esetleges kitiltás fizetési késedelem esetén
- Legnézettebb oldalak kiszűrése
- Legaktívabb felhasználók kiszűrése
- Legtöbbet fizető tulajdonosok kiszűrése

Feladatok kiosztása az 1. mérőföldkőben:

Gila Ferenc - GVPXRN: (33,3%)

- Egyedmodell
- Egyed-esemény mátrix
- Szerep-funkció mátrix

Mihály Marcell - FKNKDH: (33,3%)

- Logikai adatfolyam-diagramok
- Fizikai adatfolyam-diagramok

Csupity Tamás - AGLU8T: (33,3%)

- Egyed-kapcsolat diagram
- Egyed-kapcsolat diagram leképezése relációs adatbázissémákká
- Funkcionális függőségek felírása, relációsémák normalizálása 3NF-ig

Mindenki:

- Specifikáció, részletes feladatleírás, követelménykatalógus
- Feladat felosztás és dokumentum készítés

Feladatok kiosztása 2. mérőföldkőben:

Gila Ferenc - GVPXRN: (33,3%)

Mihály Marcell - FKNKDH: (33,3%)

Csupity Tamás - AGLU8T: (33,3%)

- A DDL parancsok létrehozása: Mindenki
- A DML parancsok létrehozása: Mindenki
- ON DELETE , kulcsok egyéb feltételek hozzáadása: Mindenki
- Feltöltés adatokkal: Mindenki

Feladatok kiosztása 3. mérfölkőben:

- Bejelentkezés és regisztráció megvalósítása, admin felvitel: (5p)
Mihály Marcell - FKNKDH: (50%)
Csupity Tamás - AGLU8T: (50%)
- Alapadatokat tartalmazó táblákhoz adatfelvitel, módosítás és törlés megvalósítása űrlapon keresztül: (0–1 pont (legfeljebb 12 tábláig))

Gila Ferenc - GVPXRN: 1 pont

A maradék fennmaradó ponton:

Mihály Marcell - FKNKDH: (50%)

Csupity Tamás - AGLU8T: (50%)

Feladatok kiosztása 4. mérföldkőben:

- Alapadatokat tartalmazó táblák adataihoz lekérdezések készítése (kötelező) – értékelés: táblánként 0–1 pont (legfeljebb 12 tábláig)

Gila Ferenc - GVPXRN: 0 pont

A maradék fennmaradó ponton:

Mihály Marcell - FKNKDH: 30%

Csupity Tamás - AGLU8T: 70%

- Triggerek írása a specifikációban jelölt funkciókhoz (kötelező*) – értékelés: triggerenként 0–3 pont

Gila Ferenc - GVPXRN: 2 pont

A maradék fennmaradó ponton:

Mihály Marcell - FKNKDH: 10 %

Csupity Tamás - AGLU8T: 90%

- Tárolt eljárások/függvények a megjelölt funkciókhoz (kötelező**) – értékelés: darabonként 0–3 pont

Gila Ferenc - GVPXRN: 2 pont

A maradék fennmaradó ponton:

Mihály Marcell - FKNKDH: 20%

Csupity Tamás - AGLU8T: 80%

- Funkciókat megvalósító összetett lekérdezések (kötelező***) – értékelés: lekérdezésenként 0–2 pont

Gila Ferenc - GVPXRN: 0

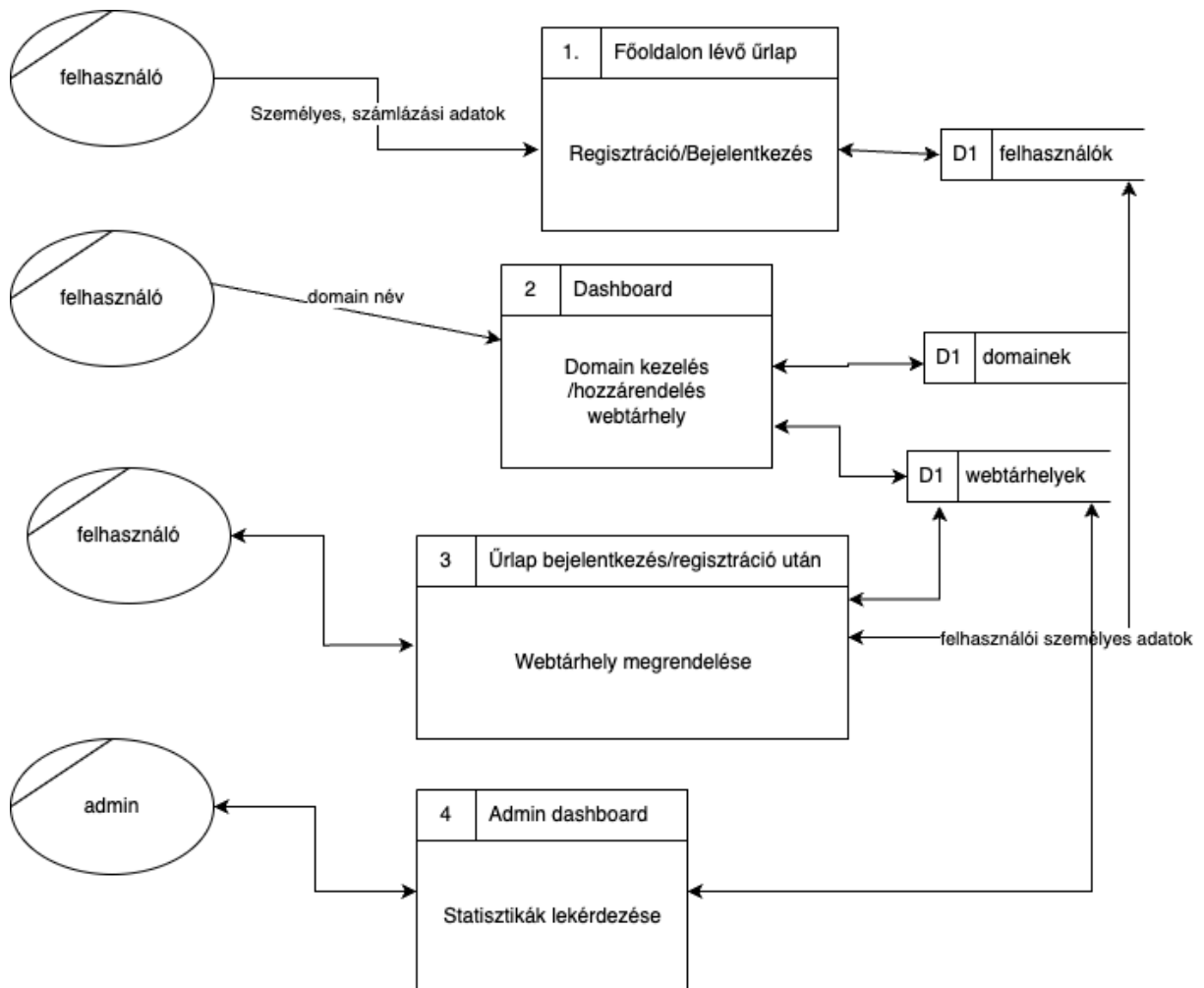
A maradék fennmaradó ponton:

Mihály Marcell - FKNKDH: 20%

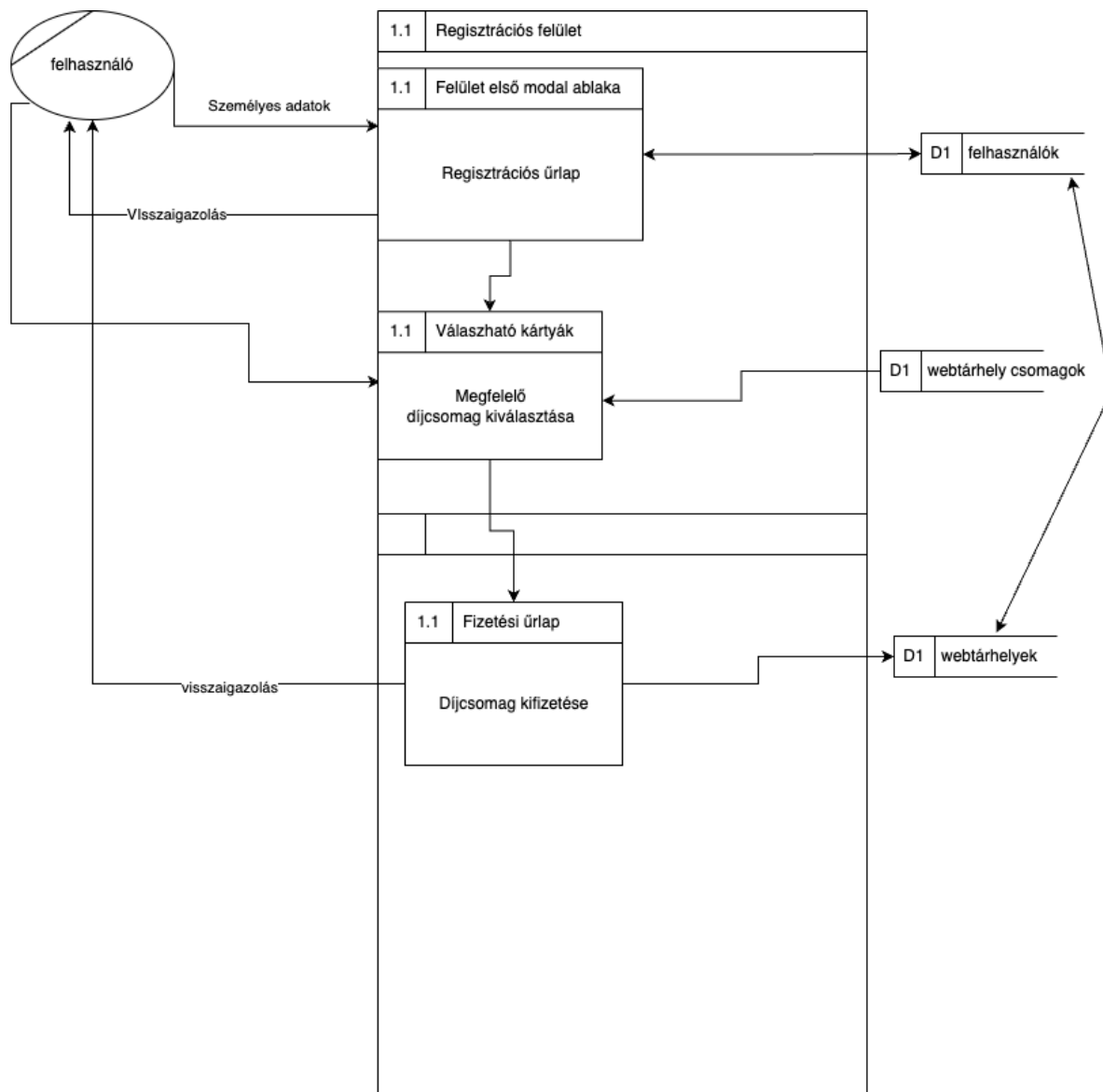
Csupity Tamás - AGLU8T: 80%

Tervezés, SSADM

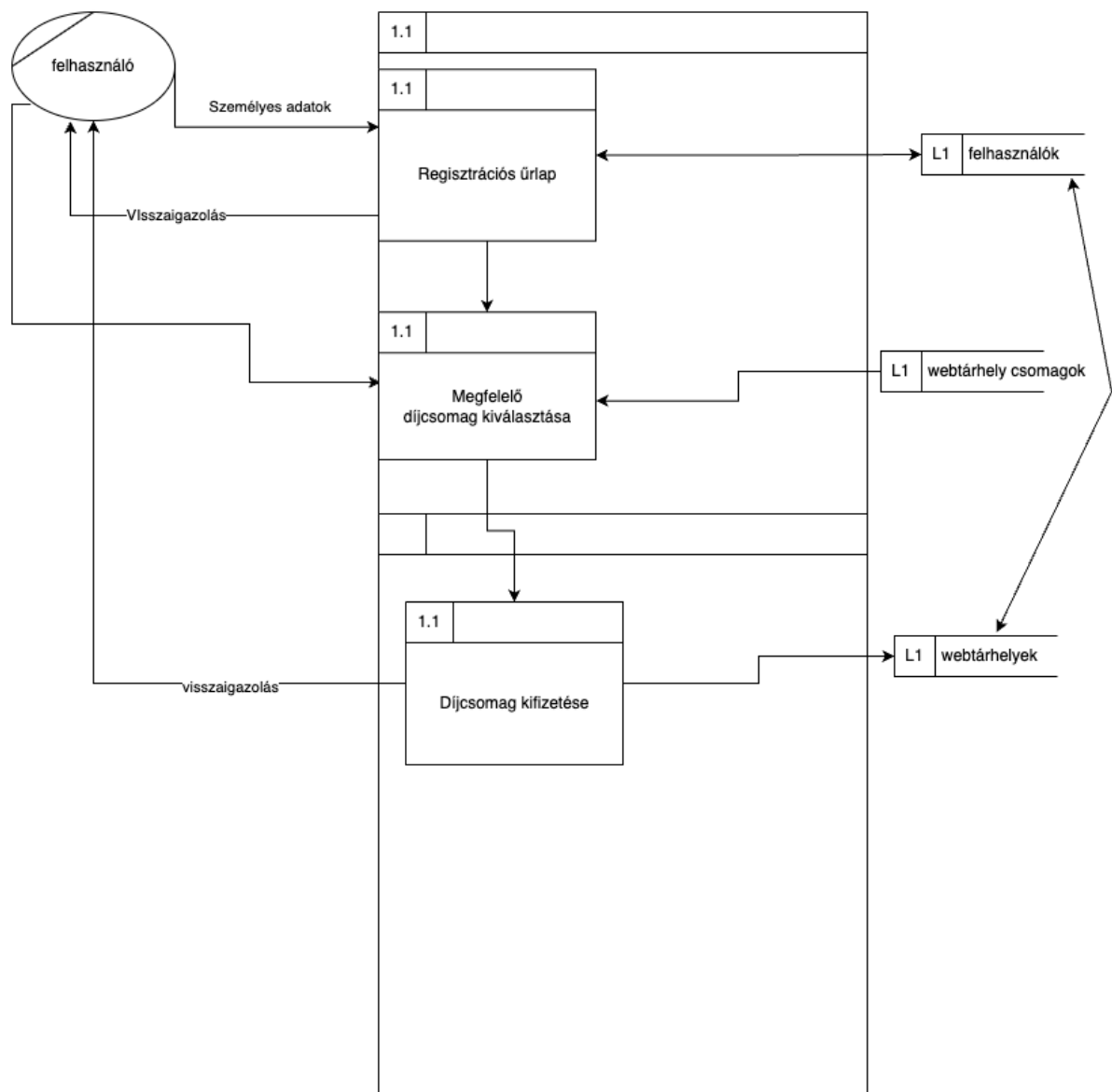
Fizikai adatfolyam: (Készítte Mihály Marcell)



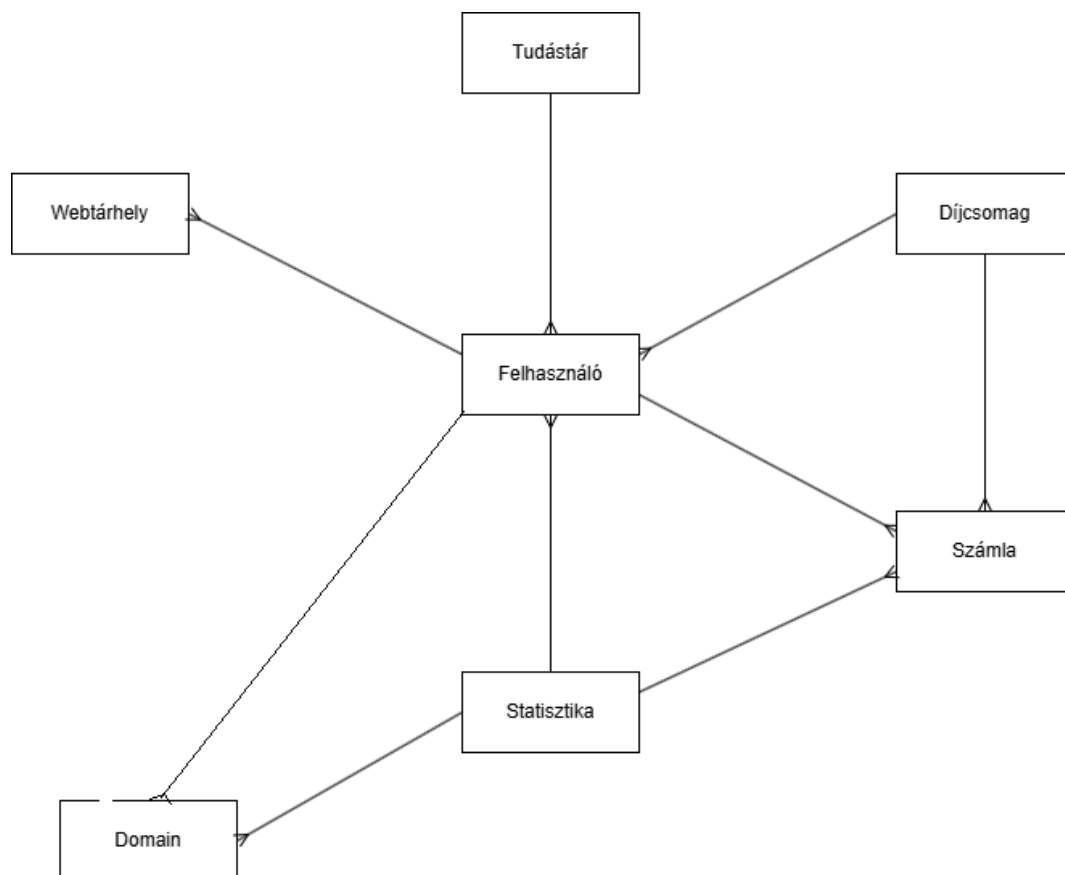
Logikai adatfolyam 1.: (Készítte Mihály Marcell)



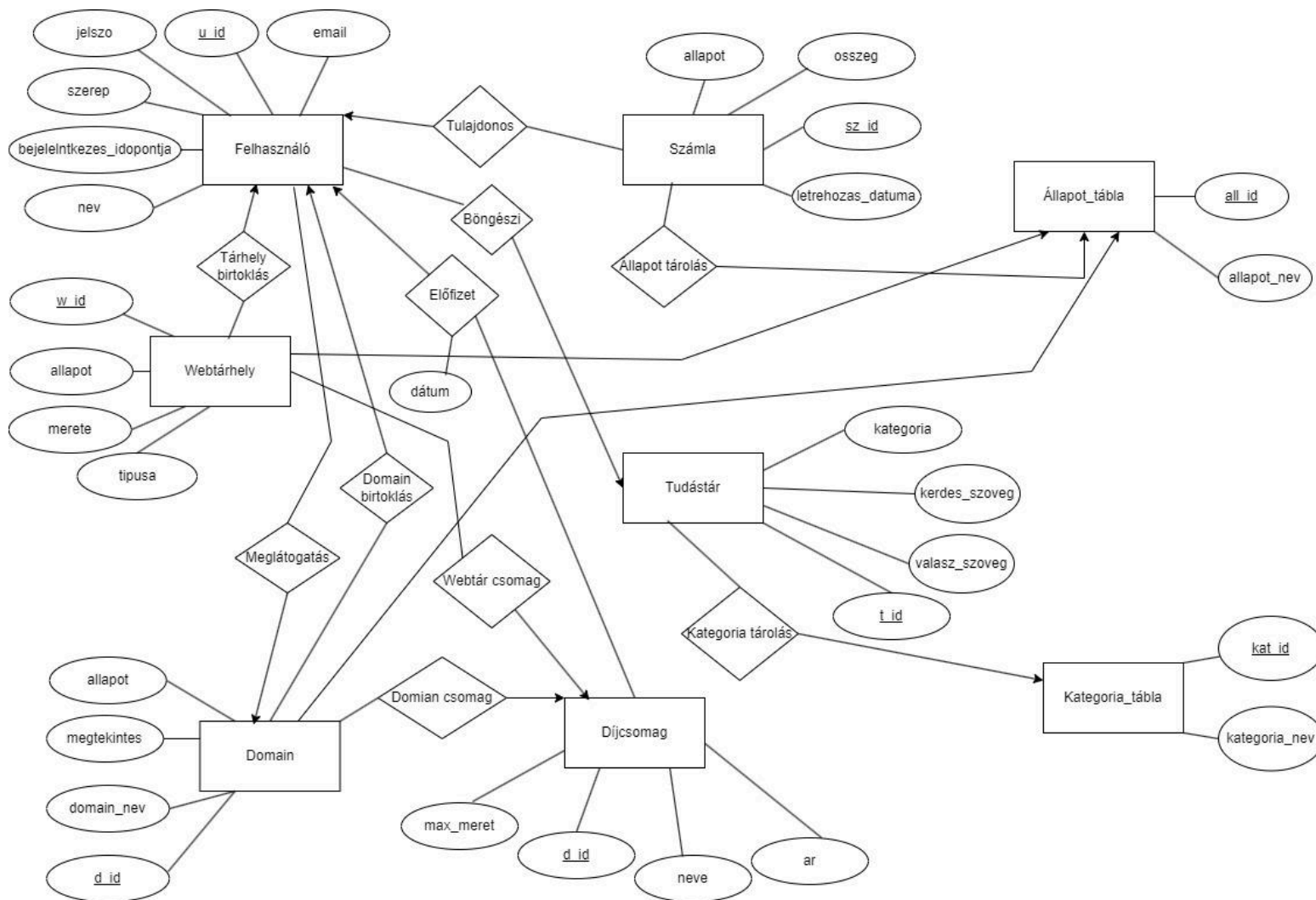
Logikai adatfolyam 2.: (Készítte Mihály Marcell)



Egyedmodell: (Készítette Gila Ferenc)



Egyed-kapcsolat:



Séma: (PK = Elsődleges kulcs, FK = külső kulcs)

Felhasználó (u_id [PK], jelszo, email, nev, szerep, bejelentkezes_idopontja)

Webtárhely (w_id [PK], allapot [FK], meret, tipus, u_id [FK], d_id [FK])

Domain (d_id [PK], domain_nev, status, megtekintes, u_id [FK], dij_id [FK])

Díjcsomag (d_id [PK], neve, max_meret, ar)

Számla (sz_id [PK], letrehozas_idopontja, osszeg, all_id [FK], u_id [FK])

Tudástár (t_id [PK], kategoria [FK], kerdes_szoveg, valasz_szoveg)

Előfizet (u_id [PK] [FK], d_id [PK] [FK], datum)

Állapot (all_id [PK], allapot_nev)

Kategória (kat_id [PK], kategoria_nev)

3NF-re hozás:

1NF-ben van, mert : Minden attribútum atomi.

2NF-ben van, mert: Minden nem kulcs attribútum teljes függőségben van az elsődleges kulccsal.

3NF-ben van mert: Nincs tranzitív függés.

Egyed-esemény-mátrix (Készítette Gila Ferenc)

[illegible]

Szerep-funkció mátrix (Készítette Gila Ferenc)

[illegible]

Funckiónális megvalósítás

Használt technológiák:

- Frontend: Svelte (JavaScript Web FrameWork)
- Backend: FastAPI (Python alapú Web FrameWork)
- Adatbázis: Oracle Cloud
- Verziókövetés: Github
- Diagramm készítés: Draw.io
- IDE: Visual Studio, PyCharm
- Dokumentum készítés: Google Docs

Megvalósított funkciók: [T] = Trigger [P] = eljárás

Felhasználókezelés: Regisztráció, bejelentkezés, felhasználói profilok

kezelése, jogosultságok beállítása: A rendszer képes felhasználót regisztrálni, bejelentkeztetni. A folyamatot jelszó titkosítás valamint autentikációs tokenek kísérik ezzel biztosítva a jogosultság kezelést is az elérhető API pontokban. Eljárással frssítésre kerül. [P]

Domainek kezelése : A rendszer képes domain egyedet tárolni és ezt képesek vagyunk CRUD valamint egyéb műveletekkel manipulálni.. A megtekintés attributumot egy eljárással kezeljük.

Webtárhely kezelése: A rendszer webtárhely egyedet tárol és ezt képesek vagyunk CRUD műveletekkel manipulálni.

Díjcsomagok (különböző tárhely és domain): A rendszer díjcsomag egyedet tárol ez határozza meg a számlázást és a webtárhely, domainek változását. Minden díjcsomagot csak egyszer vásárolhat meg a felhasználó.

Számlakészítés : Képesek vagyunk számlát létrehozni egy előfizetés megtörténetekor. [T]

Tudástár (gyakran ismételt kérdések és válaszai a webhostinggal

kapcsolatban): A felhasználói felületen elérhető a tudástár amiben a kérdések és válaszok találhatók.

Számlák mentése havonta, számlák állapotváltozásainak kezelése az

adatbázisba: A számlákat havonta minden hónap 1.-én automatikusan elkészíti a rendszer. Kezeli a fizetési határidő alapján az állapotokat valamint a számlát ki is lehet egyenlíteni. Triggerel kezeljük, hogy törölt állapotba is állítsuk. Egy eljárással kezeljük a havi kiállítást. [\[T\]](#), [\[P\]](#)

Bevételi statisztika, kimutatás készítés: A felhasználói felületen látható egy statisztika, itt megtalálható a bevétel is havi bontásban. [\[Nézettábla\]](#)

Korlátozás, esetleges kitiltás fizetési késedelem estén : A rendszer kezeli, hogyha a felhasználó nem időben fizet. Egy bizonyos idő után letiltja a vásárolt domainjeit és webtárhelyét. Triggerel és eljárásokkal kezeljük a késés állapot változást valamint a kitiltást. [\[T\]](#), [\[P\]](#)

Legnézettebb oldalak kiszűrése: A felhasználói felületen látható egy statisztika, itt megtalálható a megtekintés alapján rendezett domainek [\[Nézettábla\]](#)

Legaktívabb felhasználók kiszűrése: A felhasználói felületen látható egy statisztika, itt megtalálható a domainek, webtárhelyek, megtekintések és számlák alapján rendezett felhasználók. [\[Nézettábla\]](#)

Legtöbbet fizető tulajdonosok kiszűrése: A felhasználói felületen látható egy statisztika, itt megtalálható az összesített fizetett számlák összege alapján rendezett felhasználók. [\[Nézettábla\]](#)

Megvalósított triggerek:

1. Késedelmes számla állapotváltás (*TRIGGER*)

- **Funkció:** 2 hónap után automatikusan "késedelmes" állapotra állítja a **szamla** rekordot, ha nem lett kifizetve.
- **Trigger neve:** **szamla_fizetes_keses**
- **Használat:** **AFTER UPDATE** a számlak-ellenorzes eljárás meghívásán keresztül aktiválható.

```
create or replace TRIGGER ADMIN.szamla_fizetes_keses  
  
BEFORE UPDATE ON szamla  
  
FOR EACH ROW  
  
BEGIN  
  
    IF :OLD.all_id = 3 AND :NEW.all_id = 3 AND  
  
        MONTHS_BETWEEN(SYSDATE, :OLD.letrehozás_datuma) > 2 THEN  
  
        :NEW.all_id := 41; -- 41 = Késedelmes  
  
    END IF;  
  
END;
```


2. Számla létrehozása előfizetés esetén (*TRIGGER*)

- **Funkció:** Automatikusan számlát generál, ha új előfizetés (domain vagy webtárhely) történik.
- **Trigger neve:** `szamla_letrehozas`
- **Használat:** `AFTER UPDATE` az előfizet táblán automatikusan létrehoz egy számlát is erről.

```
create or replace TRIGGER ADMIN.szamla_letrehozas
AFTER INSERT ON elofizet
FOR EACH ROW
DECLARE
    v_ar NUMBER;
BEGIN
    SELECT ar INTO v_ar
    FROM dijsomag
    WHERE d_id = :NEW.d_id;

    INSERT INTO szamla (osszeg, letrehozas_datuma, u_id, all_id)
    VALUES (v_ar, SYSTIMESTAMP, :NEW.u_id, 3);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20002, 'A megadott d_id-hez nem található díjcsomag.');
```

3. Számla töröltre állítása (*TRIGGER*)

- **Funkció:** Ha 12 hónapja kifizetett számla, akkor az állapotot "töröltre" állítja. A trigger vagy időzített eljárással aktiválható.
- **Trigger neve:** `szamla_torolt_allapotra`
- **Használat:** `AFTER UPDATE` a számlak-ellenorzes eljáráson keresztül aktiválható.

```
create or replace TRIGGER ADMIN.szamla_torolt_allapotra
BEFORE UPDATE ON szamla
FOR EACH ROW
BEGIN
    IF :OLD.all_id = 4 AND :NEW.all_id = 4 AND
        MONTHS_BETWEEN(SYSDATE, :OLD.letrehozas_datuma) > 12 THEN
        :NEW.all_id := 5; -- 5 = Törölt
    END IF;
END;
```

Megvalósított eljárások:

1. Bejelentkezés időpont frissítése (*PROCEDURE*)

- **Funkció:** A login folyamat után frissíti a `bejelentkezes_idopontja` mezőt.
- **Eljárás neve:** `bejelentkezési_idopont_frissitese`
- **Meghívás helye:** `/auth/login` API

```
create or replace PROCEDURE ADMIN.bejelentkezes_idopont_frissitese (  
    p_user_id IN NUMBER  
)  
IS  
BEGIN  
    UPDATE felhasznalo  
    SET bejelentkezes_idopontja = SYSTIMESTAMP  
    WHERE u_id = p_user_id;  
END;
```

2. Fizetési hátralék kezelése, blokkolás (*PROCEDURE*)

- **Funkció:** Ha legalább 6 hónapja nem fizetett, akkor a `domain` és `webtarhely` rekordok blokkolásra kerülnek.
- **Eljárás neve:** `fizetesi_hatralek_kezelese`
- **Automatikus végrehajtás:** Időzítve hívja a rendszer.

```
create or replace PROCEDURE ADMIN.fizetesi_hatralek_kezelese (  
    p_felhasznalo_id NUMBER,  
    p_kimenet OUT VARCHAR2  
)  
AS  
    v_count NUMBER := 0;  
    e_nincs_fizetendo EXCEPTION;  
    PRAGMA EXCEPTION_INIT(e_nincs_fizetendo, -20002);  
BEGIN  
  
    -- Felfüggesztés + szolgáltatás blokkolása  
    SELECT COUNT(*) INTO v_count  
    FROM szamla  
    WHERE u_id = p_felhasznalo_id  
    AND all_id = 41
```

```

AND MONTHS_BETWEEN(SYSDATE, letrehozás_datuma) > 6;

IF v_count > 0 THEN
  UPDATE számla
  SET all_id = 21 -- Felfüggesztett
  WHERE u_id = p_felhasználó_id
  AND all_id = 41
  AND MONTHS_BETWEEN(SYSDATE, letrehozás_datuma) > 6;

  UPDATE webtarhely
  SET állapot = 0
  WHERE u_id = p_felhasználó_id;

  UPDATE domain
  SET állapot = 0
  WHERE u_id = p_felhasználó_id;
END IF;

-- Értesítés
p_kimenet := 'Fizetési hátralék miatt a szolgáltatásai korlátozva lettek. Kérjük, rendezze a tartozását!';

EXCEPTION
  WHEN e_nincs_fizetendo THEN
    p_kimenet := 'A felhasználónak nincs rendezetlen számlája.';
  WHEN OTHERS THEN
    p_kimenet := 'Hiba történt: ' || SQLERRM;
END;
```

3. Számla kiállítás minden hónapban (*PROCEDURE*)

- **Funkció:** Minden hónap 1-jén új számlát állít ki az aktív előfizetésekre.
- **Eljárás neve:** `haviszamlak_kiallitas`
- **Automatikus végrehajtás:** Időzítve hívja a rendszer.

```

create or replace PROCEDURE ADMIN.haviszamlak_kiallitas IS
BEGIN
  INSERT INTO számla (összeg, letrehozás_datuma, u_id, all_id)
  SELECT d.ar, SYSDATE, e.u_id, 3
  FROM elofizet e
  JOIN díjcsomag d ON e.d_id = d.d_id;

  COMMIT;
END;
```

4. Megtekintések növelése (*PROCEDURE*)

- **Funkció:** A domain megtekintés száma automatikusan nő, amikor a felhasználó lekér egy oldalt.
- **Eljárás neve:** `noveli_megtekinteseket`
- **Automatikus végrehajtás:** Időzítve hívja a rendszer.

create or replace PROCEDURE ADMIN.noveli_megtekinteseket IS
BEGIN

```
UPDATE domain d
SET megtekintes = ROUND(megtekintes * 1.02)
WHERE d.dij_id IN (
  SELECT d_id FROM dijcsomag
  WHERE LOWER(neve) LIKE '%alap%'
);
```

```
UPDATE domain d
SET megtekintes = ROUND(megtekintes * 1.10)
WHERE d.dij_id IN (
  SELECT d_id FROM dijcsomag
  WHERE LOWER(neve) LIKE '%plusz%' OR LOWER(neve) LIKE '%prémium%'
);
```

```
UPDATE domain d
SET megtekintes = ROUND(megtekintes * 1.20)
WHERE d.dij_id IN (
  SELECT d_id FROM dijcsomag
  WHERE LOWER(neve) LIKE '%vállalkozói%'
);
```

```
COMMIT;
END;
```

5. Számla állapotok kezelése (*PROCEDURE*)

- **Funkció:** Ellenőrzi a lejárt számlákat, és frissíti az állapotukat "késedelmes" vagy "törölt" állapotra.
- **Eljárás neve:** `szamlak_ellenorzes`
- **Megjegyzés:** A `szamla_torolt_allapotra,szamla_fizetes_keses` trigger ezt hívja segédként.

```
create or replace PROCEDURE ADMIN.szamlak_ellenorzes IS
BEGIN
```

```
    UPDATE szamla
    SET all_id = all_id
    WHERE all_id = 3
    AND MONTHS_BETWEEN(SYSDATE, letrehozasi_datuma) > 2;
```

```
    -- Lezárt számlák frissítése (szamla_torolt_allapotra trigger kiváltásához)
    UPDATE szamla
    SET all_id = all_id
    WHERE all_id = 4
    AND MONTHS_BETWEEN(SYSDATE, letrehozasi_datuma) > 12;
```

```
    COMMIT;
END;
```

A beütemezett adatbázis műveletek:

	JOB_NAME	ENABLED	STATE	LAST_START_DATE	NEXT_RUN_DATE
1	DOMAIN_MEGTEKINTES_NOVEL	TRUE	SCHEDULED	2025-05-15T08:00:06.	2025-05-15T09:00:00.467504Z
2	HAVISZAMLA_JOB	TRUE	SCHEDULED	(null)	2025-05-31T22:00:00Z
3	PIPELINE\$1	FALSE	DISABLED	(null)	(null)
4	PIPELINE\$2	FALSE	DISABLED	(null)	(null)
5	SZAMLA_ELLENORZES_NAPI	TRUE	SCHEDULED	2025-05-15T00:30:05.	2025-05-16T00:30:00.020652Z

Megvalósított lekérdezések:

A lekérdezéseket nézettáblákban tároltuk, ezeket funkcionálnak statsztikaként a felhasználói felületen.

1. NÉZETTÁBLA NEVE: **LEGTOBBET_FIZETOK**

- **Leírás:** Az összesíti és listázza a legtöbbet fizetett felhasználókat. A statisztikai kimutatáshoz. (statisztika_routes.py 28)
- **SQL:**

```
CREATE OR REPLACE VIEW legtobbet_fizetok AS
SELECT
  f.nev AS felhasznalo_nev,
  SUM(sz.osszeg) AS osszes_fizetes
FROM felhasznalo f
JOIN szamla sz ON f.u_id = sz.u_id
WHERE sz.all_id IN (2, 4) -- csak lezárt / teljesített
GROUP BY f.nev
ORDER BY osszes_fizetes DESC;
```

2. NÉZETTÁBLA NEVE: **LEGNEZETTEBB_DOMAIN**

- **Leírás:** Akik a legtöbb domaint és webtárhelyet vettek. A statisztikai kimutatáshoz. (statisztika_routes.py 37)
- **SQL:**

```
CREATE OR REPLACE VIEW legnezettebb_domain AS
SELECT
  f.nev AS felhasznalo_nev,
  SUM(sz.osszeg) AS osszes_fizetes
FROM felhasznalo f
JOIN szamla sz ON f.u_id = sz.u_id
WHERE sz.all_id IN (2, 4) -- csak lezárt / teljesített
GROUP BY f.nev
ORDER BY osszes_fizetes DESC;
```

3. NÉZETTÁBLA NEVE: **HAVI_BEVETEL_STAT**

- **Leírás:** Az összesített bevétel havonta. A statisztikai kimutatáshoz. (statisztika_routes.py 19)
- **SQL:**

```
CREATE OR REPLACE VIEW havi_bevetel_stat AS SELECT
    TO_CHAR(letrehozás_datuma, 'YYYY-MM') AS honap,
    SUM(összeg) AS bevétel
FROM számla
JOIN állapot_tábla állapot ON sz.all_id = állapot.all_id
WHERE all_id IN (2, 4) -- csak fizetett/lezárt számlák
GROUP BY TO_CHAR(letrehozás_datuma, 'YYYY-MM')
ORDER BY honap;
```

4. NÉZETTÁBLA NEVE: **LEGAKTIVABB_FELHASZNALOK**

- **Leírás:** A felhasználók listázása az általuk birtokoltak alapján. A statisztikai kimutatáshoz. (statisztika_routes.py 47)
- **SQL:**

```
CREATE OR REPLACE VIEW legaktivabb_felhasznalok AS
SELECT
    f.nev AS felhasznalo,
    COUNT(DISTINCT d.domain_nev) AS domainek_szama,
    COUNT(DISTINCT w.w_id) AS webtarhelyek_szama,
    COUNT(DISTINCT sz.sz_id) AS szamlak_szama
FROM felhasznalo f
LEFT JOIN domain d ON f.u_id = d.u_id
LEFT JOIN webtarhely w ON f.u_id = w.u_id
LEFT JOIN számla sz ON f.u_id = sz.u_id AND sz.all_id IN (2, 4)
GROUP BY f.nev
ORDER BY domainek_szama DESC;
```

További összetett lekérdezések:

(felhasznaloi_adatok_routes.py : 25-29)

A felhasználó számláit listázza:

```
SELECT sz.sz_id, sz.osszeg, sz.letrehozás_datuma, sz.all_id, a.allapot_nev
FROM szamla sz
JOIN allapot_tabela a ON sz.all_id = a.all_id
WHERE sz.u_id = :1
ORDER BY sz.letrehozás_datuma DESC
```

(felhasznaloi_adatok_routes.py : 44-48)

A felhasználó előfizetéseit listázza:

```
SELECT d.d_id, d.neve
FROM elofizet e
JOIN dijsomag d ON e.d_id = d.d_id
WHERE e.u_id = :1
ORDER BY d.neve
```

(szamla_routes.py 42-45)

A felhasználó számláit listázza:

```
SELECT s.sz_id, s.osszeg, s.letrehozás_datuma, s.u_id, s.all_id, a.allapot_nev
FROM szamla s
JOIN allapot_tabela a ON s.all_id = a.all_id
WHERE s.sz_id = (SELECT MAX(sz_id) FROM szamla)
```

(vasarlas_routes.py 42-47)

A felhasználó tárhelyét listázza:

```
SELECT w.w_id
FROM webtarhely w
JOIN allapot_tabela a ON w.all_id = a.all_id
WHERE w.allapot = 0 AND w.meret >= :1
ORDER BY w.meret
FETCH FIRST 1 ROWS ONLY
```