Háttéralkalmazások
Bevezetés

Imre Gábor

Q.B224

gabor@aut.bme.hu

imre.gabor@vik.bme.hu

BME AUT
Automatizálási és
Alkalmazott
Informatikai Tanszék

Tartalom

- Tantárgy információk
- Háromrétegű architektúra
- A félév témái
- Minta adatbázis

Tantárgy információk

Tantárgy információk

- Kari Moodle: https://edu.vik.bme.hu/
- Előadás pdf-ek, gyakorlat feladatok beadása

Előadások, gyakorlatok

- Előadás: QBF15, hétfőnként, 10:15-11:45
  - Pdf-ek mindig felkerülnek a tantárgy oldalára
  - FONTOS: május 5. pénteken is lesz előadás, a május 1-gyel való munkarend csere miatt
- Gyakorlatok: IL207, péntekenként 2 csoport: 10:15-11:45, 12:15-13:45
  - Nagypénteken kívül május 5-én sincs gyakorlat, a fenti munkarend csere miatt!
  - A leírások és beadási határidők a tantárgy oldalán
  - A beadandó feltöltése is a tantárgy oldalán

Követelmények

- Előadás
  - Nem ellenőrzünk jelenlétet, de minden vizsgaanyag, ami elhangzik
- Gyakorlat
  - 10 érdemi gyakorlat + 2 házi beszedés + 2 elmarad
  - A teljesítés feltétele: a beadandó anyag feltöltése határidőig a tanszéki portálra
  - **Az aláírás egyik feltétele: a 10 érdemiből 7 teljesítése**
- Házi feladat: 2 a félév során, mindkettő 20 pontos
  - **Május 11.** Adatmodell és adatelérési réteg fejlesztése
  - **Június 1.** Adatmodell, adatréteg és REST API fejlesztése (kisebb adatmodell)
  - Mindkét házi feladatot személyesen be kell mutatni a beadási határidők másnapjára eső gyakorlaton (önálló megoldás ellenőrzése kérdésekkel)
  - Az egyiket (szabadon választható) .NET, a másikat Java (Spring) platformon kell megvalósítani.
  - **Az aláírás másik feltétele: mindkét házi feladatra legalább 10-10 pont**
- Írásbeli vizsga: 60 pont, **min. 30**-at kell elérni
- Aláírás és sikeres vizsga esetén a félév végi pontszám (max. 100):
  - 1. házi pont + 2. házi pont + vizsgán elért pont

Háromrétegű architektúra

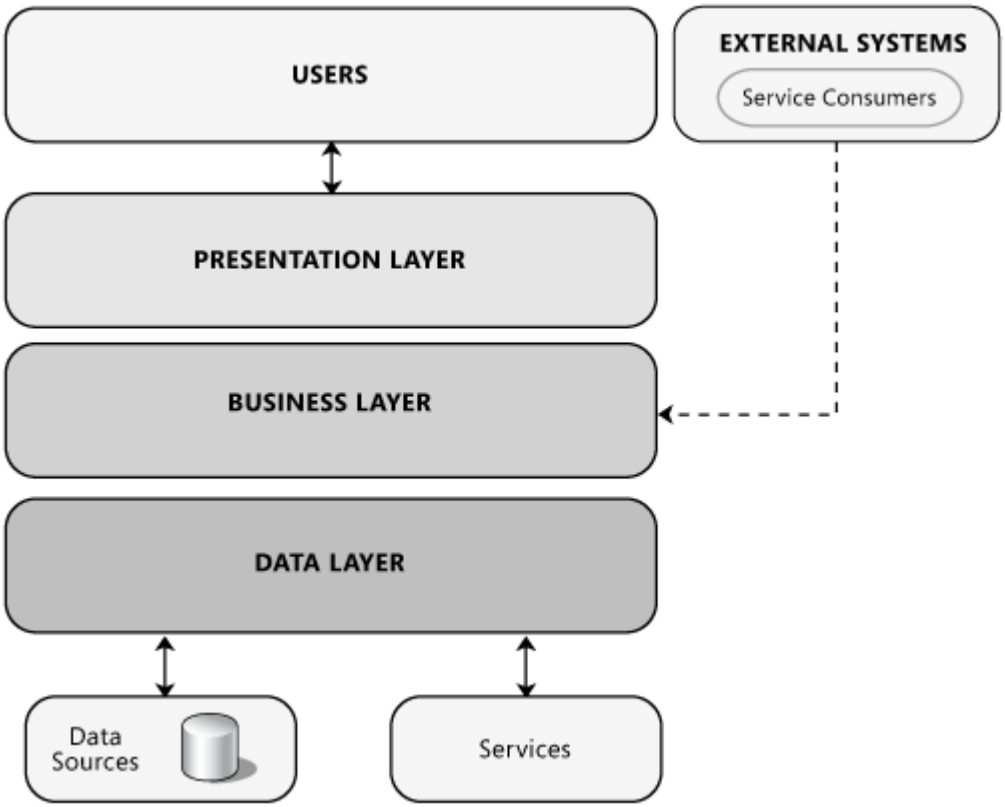
Separation of Concerns

- Szoftvertechnológia- és technikákból ismert tananyag (Architekturális tervezés)
- Separation of Concerns (SoC) → Rétegek → Háromrétegű architektúra vállalati információs rendszerekben
- SoC: a szoftver komponenseit válasszuk szét a felelősségeik szerint
- Az egyik legalapvetőbb szétválasztás: UI és logika, erre SoC-t alkalmazva:
  - Kód könnyebb megértése
  - Párhuzamos fejlesztés
  - Könnyebb bővítés, karbantartás
  - Jobb újrafelhasználhatóság
  - Izoláltan tesztelhetőség
- A UI komponensek meghívják a logikai komponenseket, de fordítva nem → **rétegekbe** szervezhetők

Perzisztens adatok

- Az alkalmazások nagy része perzisztens (megmaradó) adatokkal dolgozik
- A logikai réteg komponensein belül így további felelősség azonosítható: egy részük az adatok tárolásáért, eléréséért felelős → 3 rétegű architektúra
  - A rétegzést itt is az teszi lehetővé, hogy a függőség egyirányú: az üzleti logikai réteg hívja az adatréteget, nem fordítva

Háromrétegű architektúra



Microsoft Application Architecture Guide, 2nd Edition

https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ee658109%28v%3dpandp.10%29

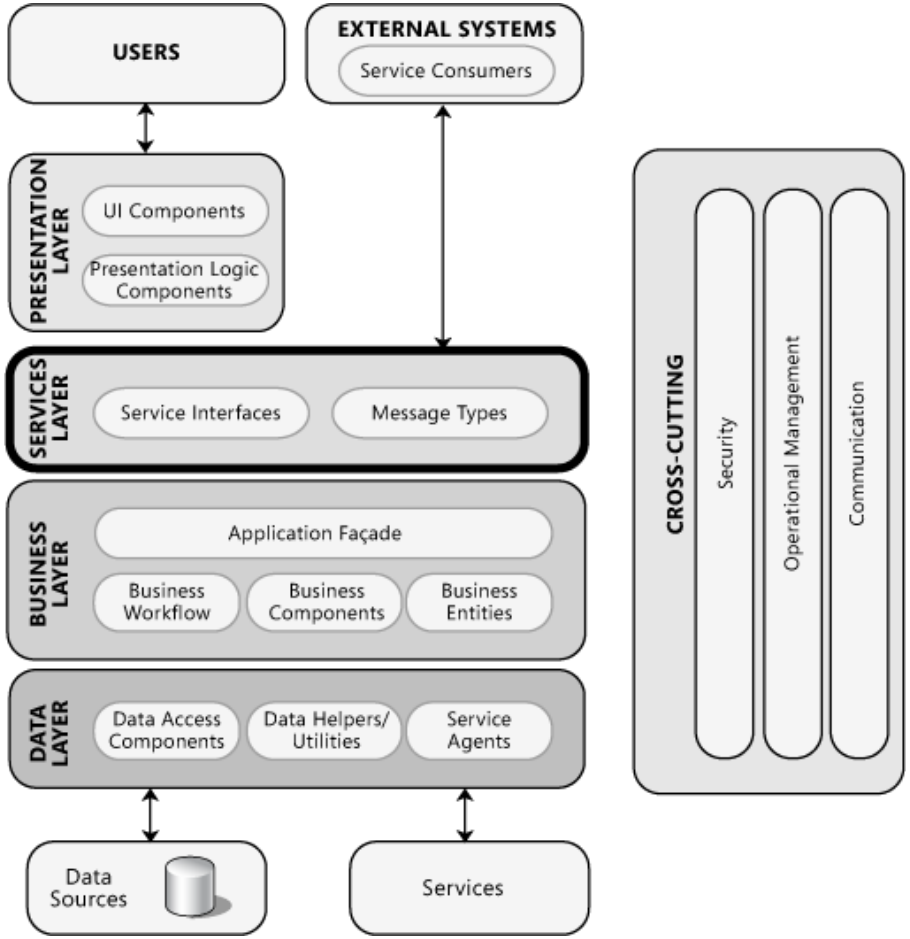
Az egyes rétegek feladatai

- Megjelenítési réteg (presentation layer, UI layer): felhasználói felület biztosítása, amely az adatok megjelenítését és a felhasználói interakciók kezelését teszi lehetővé
  - 2 fő típusa: vastag- és vékonykliens (böngésző)
- Üzleti logikai réteg (business layer, business logic layer, BLL): az alkalmazás funkcióinak megjelenítés-független megvalósítása, üzleti logikai, validációs szabályokkal együtt
- Adatréteg (data layer, data access layer, DAL, persistence layer): Hozzáférést biztosít az alkalmazás által használt (tárolt vagy lekérdezett) adatokhoz

Háromrétegű architektúra változatai

- Sok forrás a rendszer határain belülre sorolja az adatbázisokat, ilyenkor az általuk alkotott réteg neve az adatréteg (esetleg adattárolási réteg, adatbázis réteg), és az alkalmazás legalsó rétege adatelérési réteg (data access layer) néven szerepel → így már 4 réteg van
- Helyenként az adatelérési réteget az üzleti logikai réteg egy alrétegeként ábrázolják → így az adatbázis réteggel együtt is csak 3 réteg van
- Webes alkalmazások esetén a megjelenítés és az üzleti logika közé beékelődik a webréteg, amely a webes kliensekkel (pl. böngésző vagy mobil alkalmazás) való kapcsolattartásért felel → 4 vagy 5 réteg
- A rétegek számának képlékenysége miatt gyakori terminológia: n-rétegű architektúra
  - Fontos, hogy n!=2, a kétrétegű egész más világ

Rétegeken belüli komponensek



Microsoft Application Architecture Guide, 2nd Edition

https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ee658109%28v%3dpandp.10%29

Logikai vs. Fizikai rétegek

- Logikai réteg (layer): az eddig tárgyalt szoftveres rétegek
- Fizikai réteg (tier): a gép (esetleg virtuális), amin a logikai réteg(ek) fut(nak)
- Az egyes logikai rétegek nem feltétlenül lesznek külön fizikai rétegre telepítve
  - Pl. fejlesztői környezetben gyakran 1 fizikai rétegen fut minden logikai réteg
- Ha a logikai rétegek között fizikai réteghatár (vagy minimum processzhatár) is húzódik, a köztük lévő távoli elérést (hálózati kommunikációt) meg kell oldani
  - A megjelenítés és üzleti logikai réteg között ma leggyakrabban HTTP fölötti megoldások jellemzők, de vastag klienseknél előfordulnak más, bináris protokollok is (Java RMI, CORBA IIOP, .NET Remoting, WCF)
  - Az adatbázis réteg felé tipikusan gyártóspecifikus bináris protokollok
- Egy logikai réteget a fizikai rétegben több szerverre is telepíthetünk
  - Az ilyen típusú redundancia oka a jobb hibatűrés vagy magasabb teljesítmény elérése (vízszintes skálázás)
  - Ilyenkor meg kell oldani a redundáns szerverek közti terheléselosztást (load balancing)

Frontend-backend fogalma

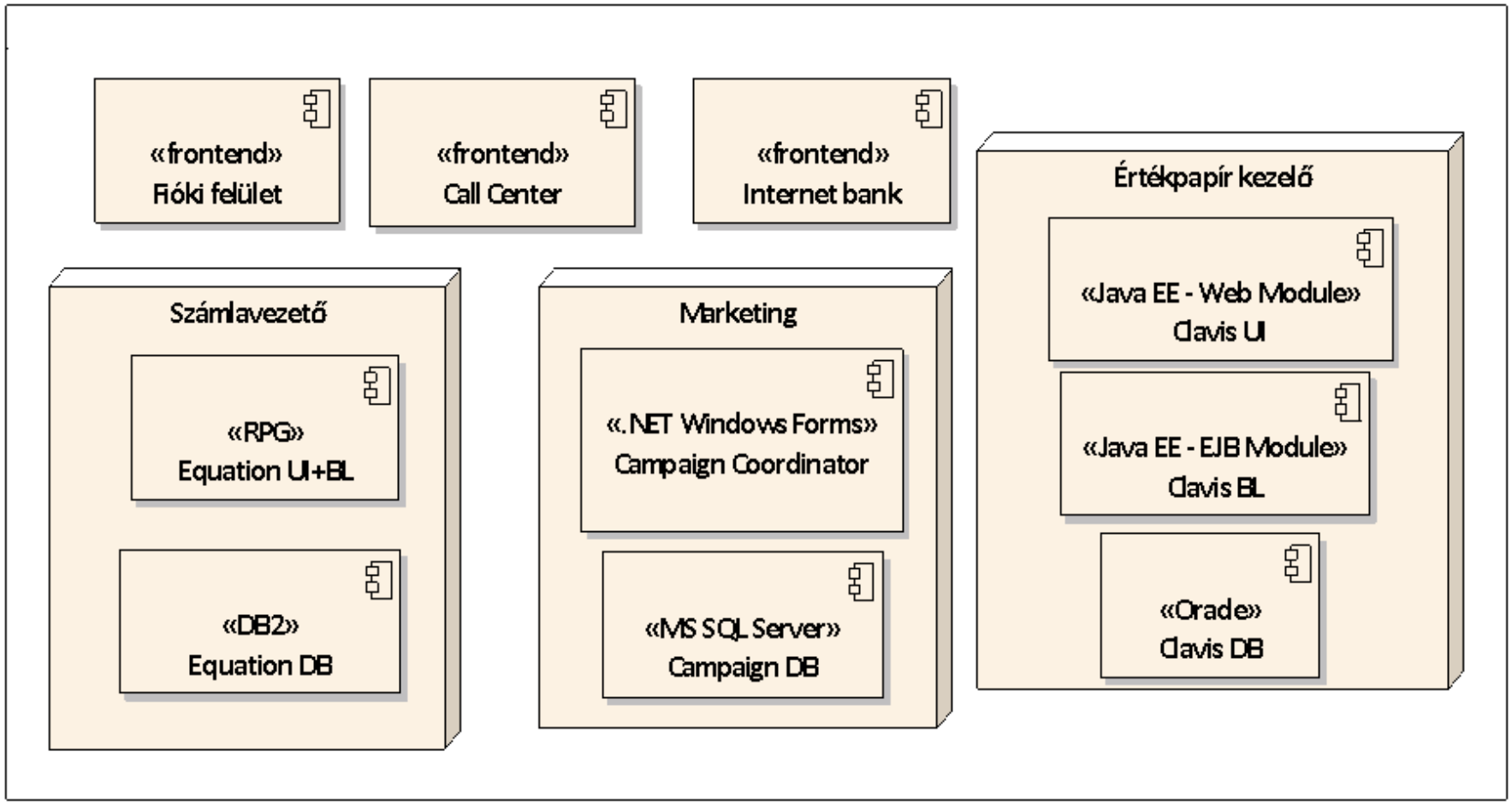
- Frontend: a Kliensalkalmazások tantárgy témája
  - A felhasználói felület előállításáért felelős komponensek, alkalmazások tartoznak ide, a megjelenítési réteggel esik egybe
  - Tipikus technológiák: HTML, CSS, JavaScript, TypeScript, Angular, React, Android, …
- Backend: a Háttéralkalmazások tantárgy témája
  - Minden, ami nem a megjelenítés: webréteg szerver oldala, üzleti logikai réteg, adatelérési réteg, adatbázisok
  - Tipikus technológiák:
    - Adatbázisok: relációs (SQL) és nem relációs (NoSQL)
    - Szerver oldalon futó üzleti logika
      - Java, Spring, Java EE,
      - C#, .NET technológiák,
      - Node.js,
      - számítási felhők

Elosztott megjelenítési réteg

- A megjelenítés valamekkora része kötődhet a szerver oldalhoz, főleg webalkalmazások esetén
- **Szerver oldal szerepe erős** (2000-es évek elejéig egyeduralkodó): a kliens oldali interakciók (pl. form elküldés, link kattintása) HTTP kérést váltanak ki, amelyre válaszul szerver oldalon generálódik ki a következő HTML, amit a böngésző megint megjelenít
- **Szerver oldal szerepe minimális**:
  - SPA (új alkalmazásoknál a leggyakoribb): szerver oldalról töltődik le statikus HTML+CSS+JS formájában egy ún. Single Page Application (SPA), onnantól a megjelenítés teljesen a kliens oldal (böngésző) feladata, csak adatokért fordul a backendhez, aszinkron módon
  - Böngésző plugin alapú megoldások (kihalófélben): a szerver oldalról bináris kód töltődik le, amelyet egy böngésző plugin kliens oldalon futtat: Java Applet, ActiveX control, Silverlight, Flash → a böngésző pluginek támogatása fokozatosan megszűnt
- **Köztes megoldás**: a szerver oldalról generált HTML-ek JS kódot is tartalmaznak, amelyek bizonyos események hatására adatokat vagy szerver oldalon generált HTML darabokat kérnek a szerver oldalról

Példa

- Többrétegű architektúrák integrációja
  - Ha minden rendszer eltárolja az ügyfél címét, hogyan védekeznénk az adatok szétcsúszása ellen?



A félév témái

Előadások

- Adatbázisok architekturális felépítése, tranzakciókezelés, MSSQL sajátosságok
- MSSQL programozása
- Adatelérési osztálykönyvtárak, objektum-relációs leképezés
- Entity Framework
- Java Persistence API
- Spring alapok
- Objektumok sorosítása: JSON, XML
- XML webszolgáltatások és REST API
- Webes felület szerver oldali generálása
- Felhő alapú rendszerek (Azure)
- Nem relációs adatbázisok
- Mikroszolgáltatások

Gyakorlatok

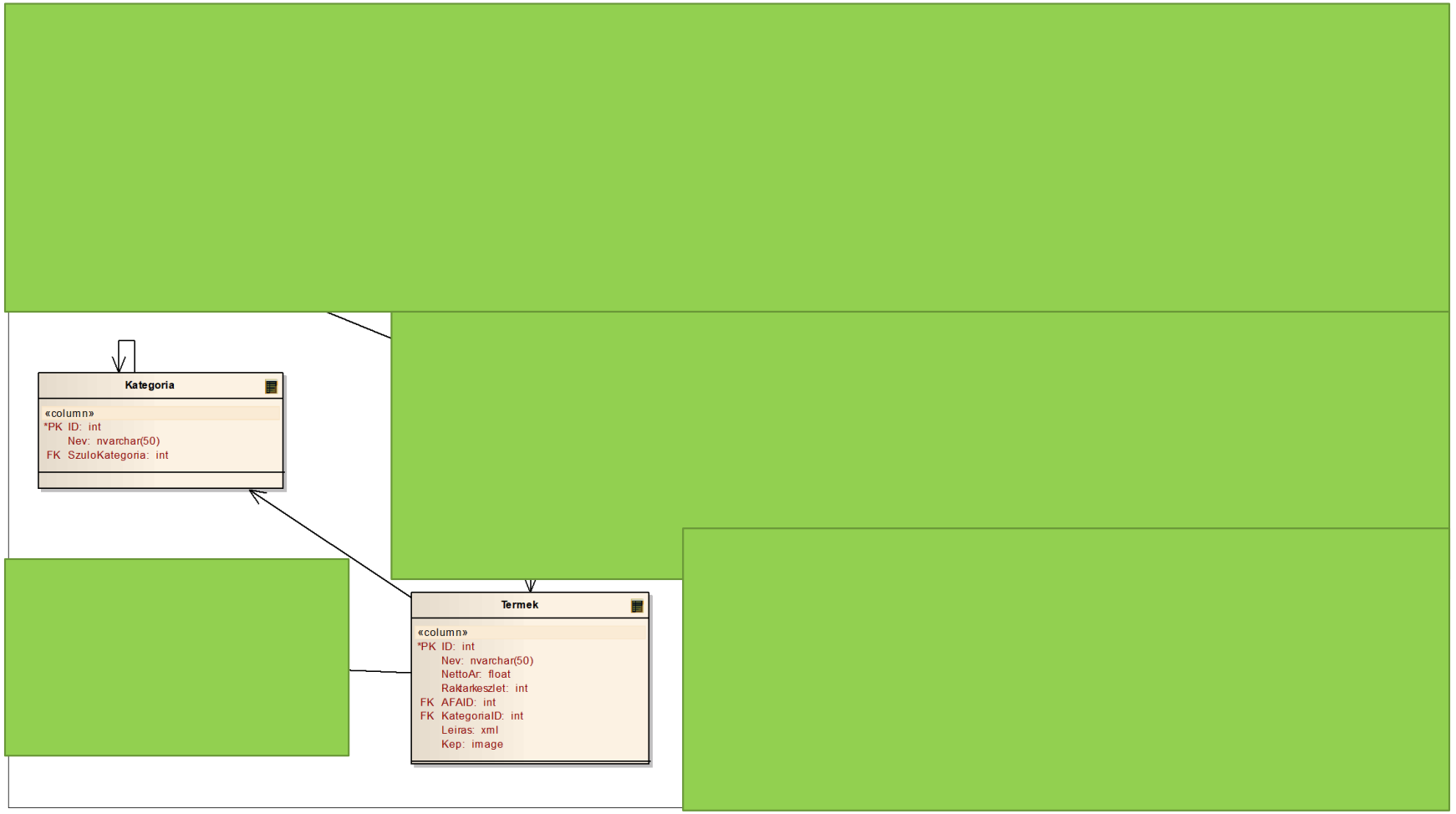
- SQL alapok
- Tranzakciók
- MSSQL szerver oldali programozása
- JDBC
- Entity Framework
- JPA
- REST API Spring platformon
- REST API .NET platformon
- 1. házi beadás
- Webes felület szerver oldali generálása ASP.NET-tel
- Azure
- XML webszolgáltatások (idén elmarad)
- 2. házi beadás

Minta adatbázis

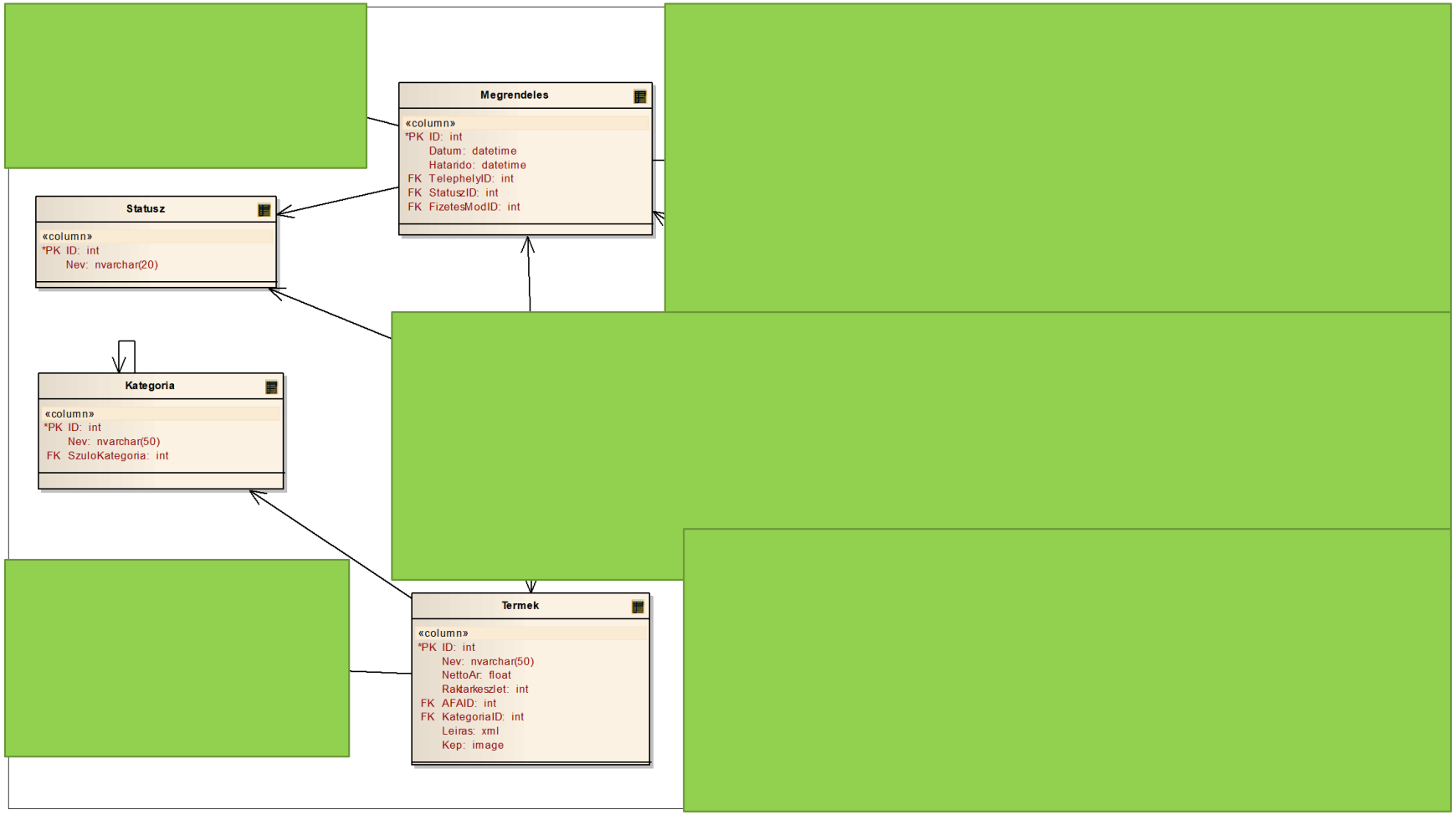
Minta adatbázis

- Több gyakorlat egy közös adatmodellt használ
- Kereskedelmi cég (Játékbolt)
- Készletek nyilvántartása
- Termék kategóriák hierarchikusan
- Vevők, telephelyek
- Megrendelések, megrendelés státusz
- Számlázás

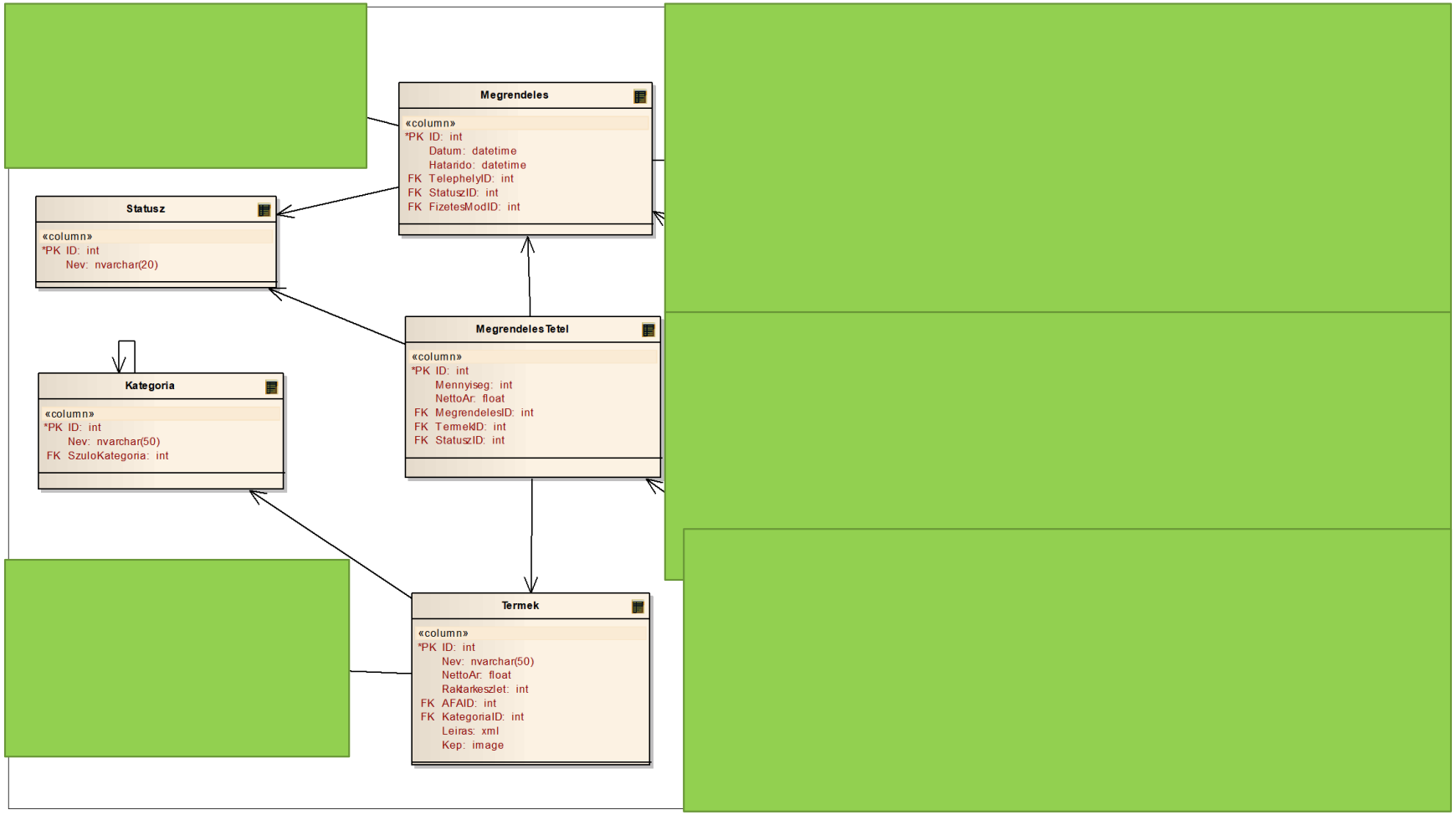
Adatmodell


Kategoria
«column»
*PK ID: int
Nev: nvarchar(50)
FK SzuloKategoria: int
Termek
«column»
*PK ID: int
Nev: nvarchar(50)
NettoAr: float
Raktarkeszlet: int
FK AFAID: int
FK KategoriaID: int
Leiras: xml
Kep: image

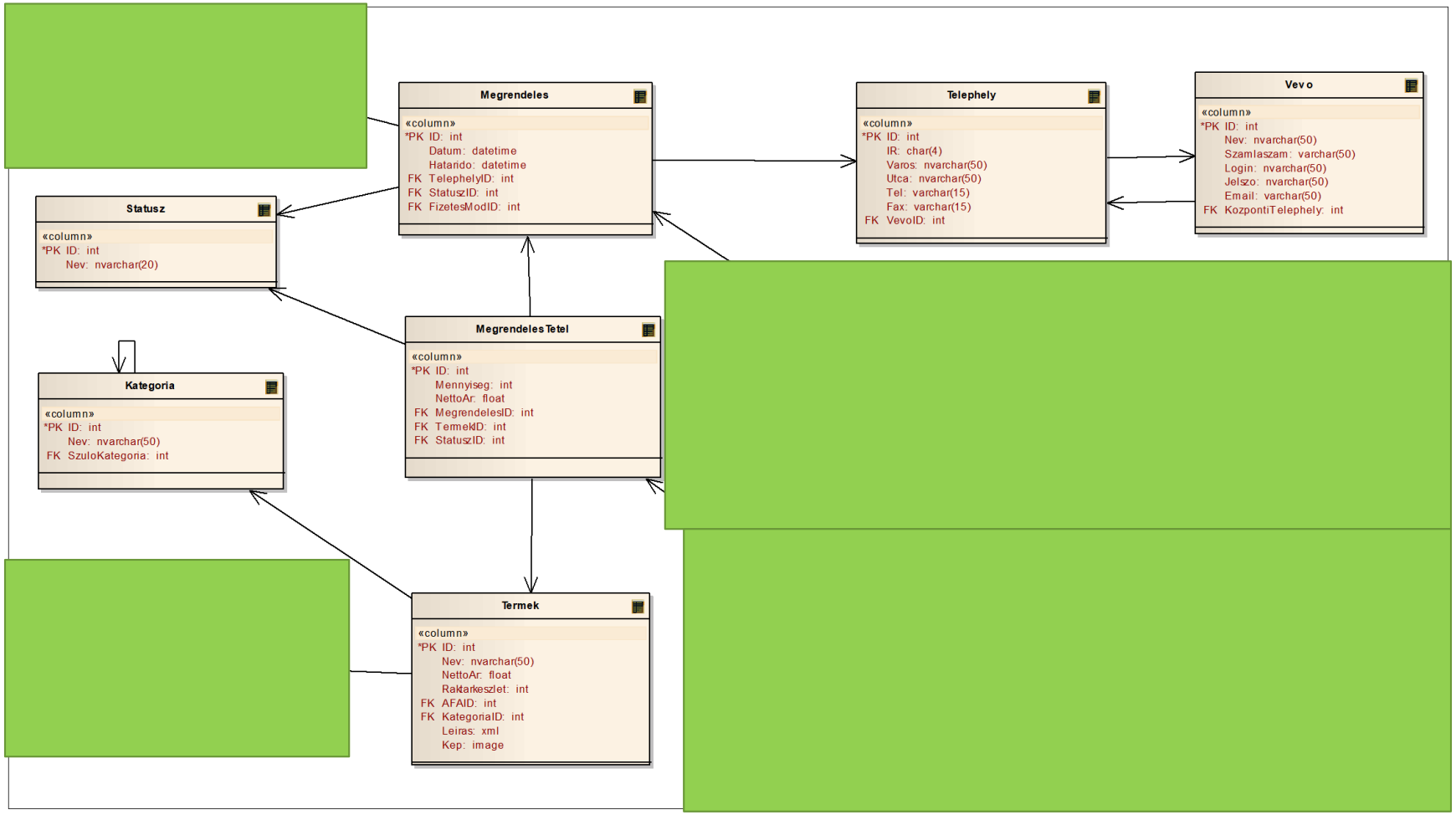
Adatmodell


Megrendeles
«column»
*PK ID: int
Datum: datetime
Hatarido: datetime
FK TelephelyID: int
FK StatuszID: int
FK FizetesModID: int
Statusz
«column»
*PK ID: int
Nev: nvarchar(20)
Kategoria
«column»
*PK ID: int
Nev: nvarchar(50)
FK SzuloKategoria: int
Termek
«column»
*PK ID: int
Nev: nvarchar(50)
NettoAr: float
Raktarkeszlet: int
FK AFAID: int
FK KategoriaID: int
Leiras: xml
Kep: image

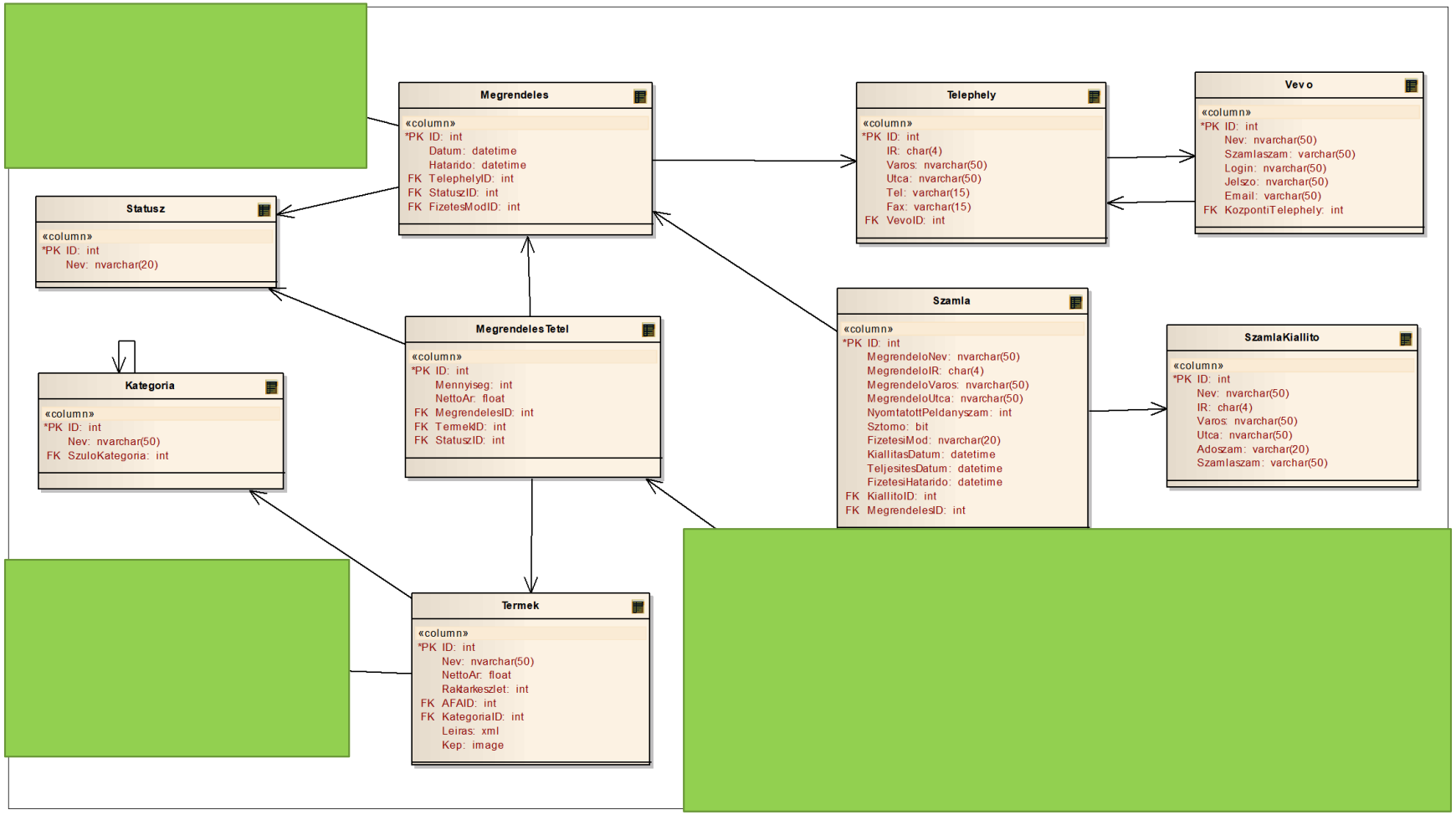
Adatmodell


Megrendeles
«column»
*PK ID: int
Datum: datetime
Hatarido: datetime
FK TelephelyID: int
FK StatuszID: int
FK FizetesModID: int
Statusz
«column»
*PK ID: int
Nev: nvarchar(20)
MegrendelesTetel
«column»
*PK ID: int
Mennyiseg: int
NettoAr: float
FK MegrendelesID: int
FK TermekID: int
FK StatuszID: int
Kategoria
«column»
*PK ID: int
Nev: nvarchar(50)
FK SzuloKategoria: int
Termek
«column»
*PK ID: int
Nev: nvarchar(50)
NettoAr: float
Raktarkeszlet: int
FK AFAID: int
FK KategoriaID: int
Leiras: xml
Kep: image

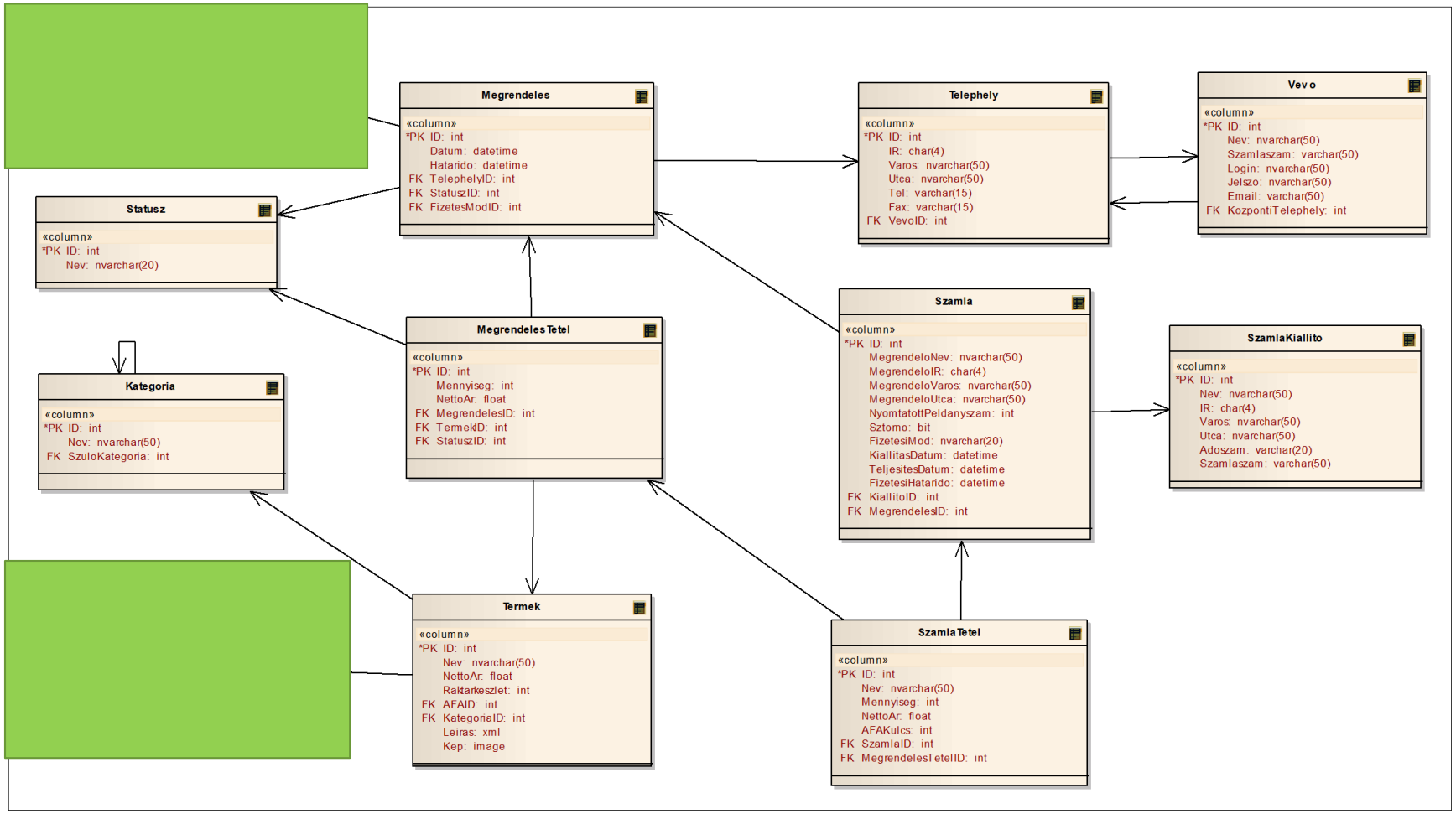
Adatmodell


Megrendeles
«column»
*PK ID: int
Datum: datetime
Hatarido: datetime
FK TelephelyID: int
FK StatuszID: int
FK FizetesModID: int
Telephely
«column»
*PK ID: int
IR: char(4)
Varos: nvarchar(50)
Utca: nvarchar(50)
Tel: varchar(15)
Fax: varchar(15)
FK VevoID: int
Vevo
«column»
*PK ID: int
Nev: nvarchar(50)
Szamlaszam: varchar(50)
Login: nvarchar(50)
Jelszo: nvarchar(50)
Email: varchar(50)
FK KozpontiTelephely: int
Statusz
«column»
*PK ID: int
Nev: nvarchar(20)
MegrendelesTetel
«column»
*PK ID: int
Mennyiseg: int
NettoAr: float
FK MegrendelesID: int
FK TermekID: int
FK StatuszID: int
Kategoria
«column»
*PK ID: int
Nev: nvarchar(50)
FK SzuloKategoria: int
Termek
«column»
*PK ID: int
Nev: nvarchar(50)
NettoAr: float
Raktarkeszlet: int
FK AFAID: int
FK KategoriaID: int
Leiras: xml
Kep: image

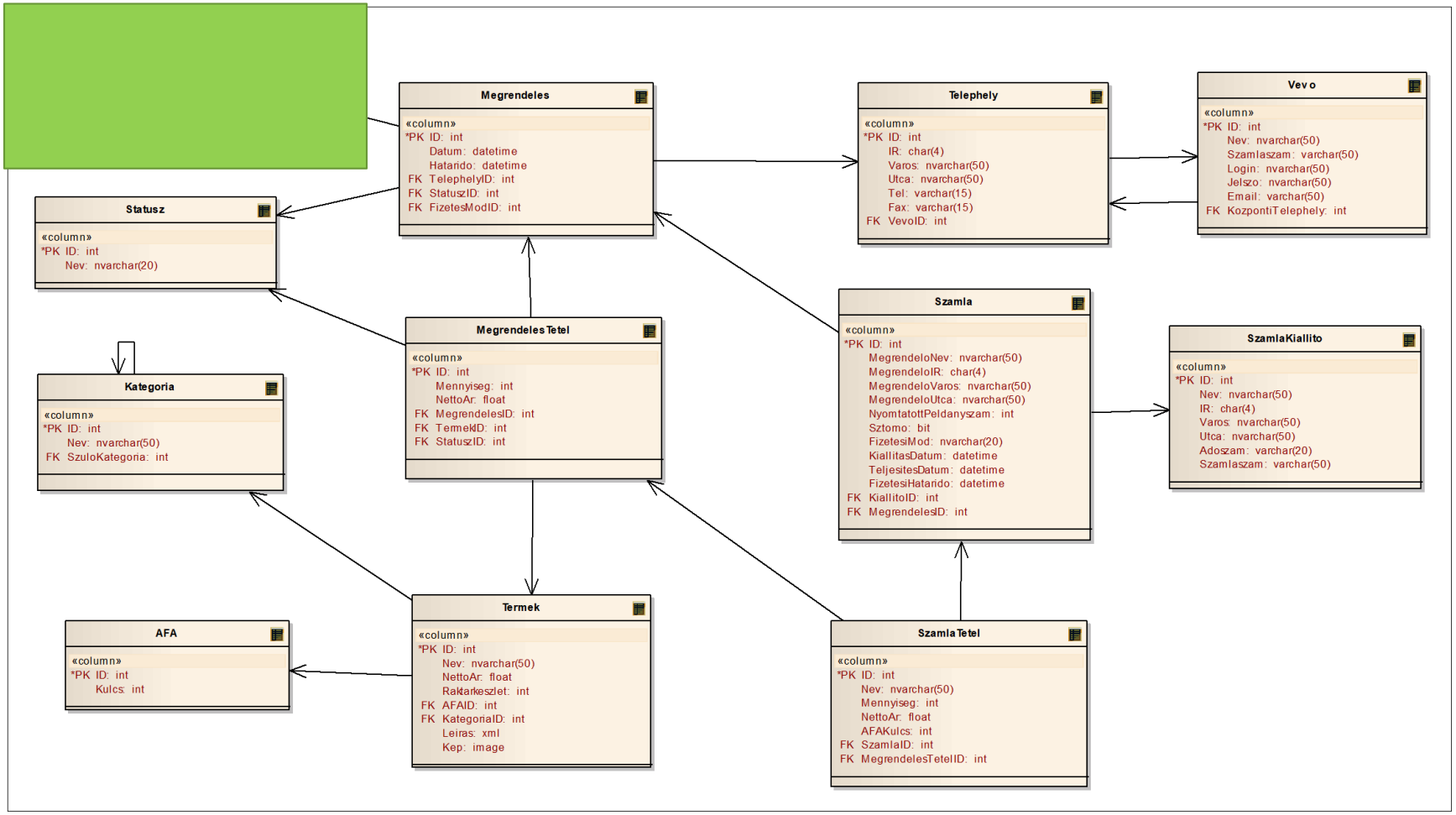
Adatmodell


Megrendeles
«column»
*PK ID: int
Datum: datetime
Hatarido: datetime
FK TelephelyID: int
FK StatuszID: int
FK FizetesModID: int
Telephely
«column»
*PK ID: int
IR: char(4)
Varos: nvarchar(50)
Utca: nvarchar(50)
Tel: varchar(15)
Fax: varchar(15)
FK VevoID: int
Vevo
«column»
*PK ID: int
Nev: nvarchar(50)
Szamlaszam: varchar(50)
Login: nvarchar(50)
Jelszo: nvarchar(50)
Email: varchar(50)
FK KozpontiTelephely: int
Statusz
«column»
*PK ID: int
Nev: nvarchar(20)
MegrendelesTetel
«column»
*PK ID: int
Mennyiseg: int
NettoAr: float
FK MegrendelesID: int
FK TermekID: int
FK StatuszID: int
Szamla
«column»
*PK ID: int
MegrendeloNev: nvarchar(50)
MegrendeloIR: char(4)
MegrendeloVaros: nvarchar(50)
MegrendeloUtca: nvarchar(50)
NyomtatottPeldanyszam: int
Sztorno: bit
FizetesiMod: nvarchar(20)
KiallitasDatum: datetime
TeljesitesDatum: datetime
FizetesiHatarido: datetime
FK KiallitoID: int
FK MegrendelesID: int
SzamlaKiallito
«column»
*PK ID: int
Nev: nvarchar(50)
IR: char(4)
Varos: nvarchar(50)
Utca: nvarchar(50)
Adoszam: varchar(20)
Szamlaszam: varchar(50)
Kategoria
«column»
*PK ID: int
Nev: nvarchar(50)
FK SzuloKategoria: int
Termek
«column»
*PK ID: int
Nev: nvarchar(50)
NettoAr: float
Raktarkeszlet: int
FK AFAID: int
FK KategoriaID: int
Leiras: xml
Kep: image

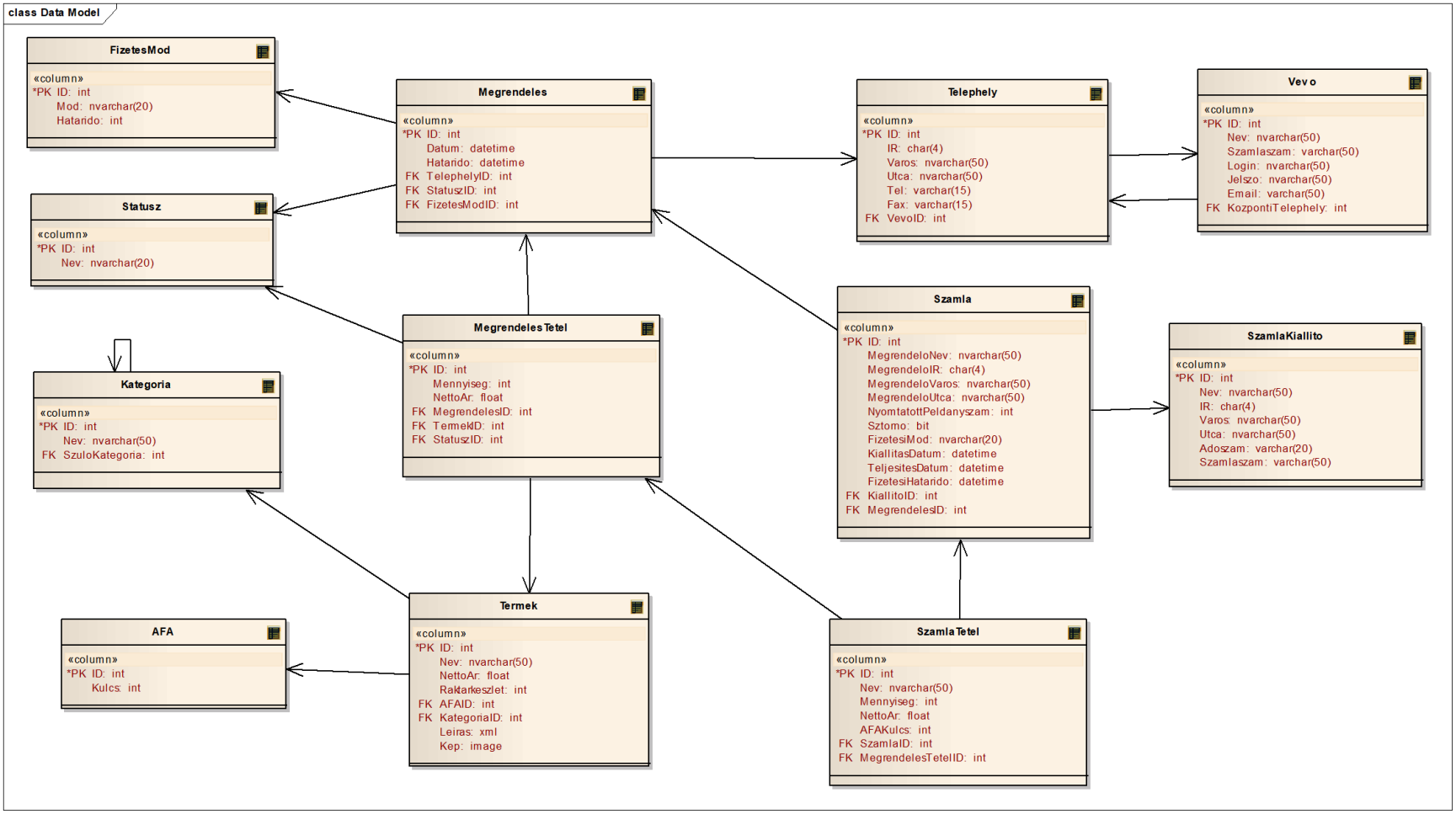
Adatmodell


Megrendeles
«column»
*PK ID: int
Datum: datetime
Hatarido: datetime
FK TelephelyID: int
FK StatuszID: int
FK FizetesModID: int
Telephely
«column»
*PK ID: int
IR: char(4)
Varos: nvarchar(50)
Utca: nvarchar(50)
Tel: varchar(15)
Fax: varchar(15)
FK VevoID: int
Vevo
«column»
*PK ID: int
Nev: nvarchar(50)
Szamlaszam: varchar(50)
Login: nvarchar(50)
Jelszo: nvarchar(50)
Email: varchar(50)
FK KozpontiTelephely: int
Statusz
«column»
*PK ID: int
Nev: nvarchar(20)
MegrendelesTetel
«column»
*PK ID: int
Mennyiseg: int
NettoAr: float
FK MegrendelesID: int
FK TermekID: int
FK StatuszID: int
Szamla
«column»
*PK ID: int
MegrendeloNev: nvarchar(50)
MegrendeloIR: char(4)
MegrendeloVaros: nvarchar(50)
MegrendeloUtca: nvarchar(50)
NyomtatottPeldanyszam: int
Sztorno: bit
FizetesiMod: nvarchar(20)
KiallitasDatum: datetime
TeljesitesDatum: datetime
FizetesiHatarido: datetime
FK KiallitoID: int
FK MegrendelesID: int
SzamlaKiallito
«column»
*PK ID: int
Nev: nvarchar(50)
IR: char(4)
Varos: nvarchar(50)
Utca: nvarchar(50)
Adoszam: varchar(20)
Szamlaszam: varchar(50)
Kategoria
«column»
*PK ID: int
Nev: nvarchar(50)
FK SzuloKategoria: int
Termek
«column»
*PK ID: int
Nev: nvarchar(50)
NettoAr: float
Raktarkeszlet: int
FK AFAID: int
FK KategoriaID: int
Leiras: xml
Kep: image
SzamlaTetel
«column»
*PK ID: int
Nev: nvarchar(50)
Mennyiseg: int
NettoAr: float
AFAKulcs: int
FK SzamlaID: int
FK MegrendelesTetelID: int

Adatmodell


Megrendeles
«column»
*PK ID: int
Datum: datetime
Hatarido: datetime
FK TelephelyID: int
FK StatuszID: int
FK FizetesModID: int
Telephely
«column»
*PK ID: int
IR: char(4)
Varos: nvarchar(50)
Utca: nvarchar(50)
Tel: varchar(15)
Fax: varchar(15)
FK VevoID: int
Vevo
«column»
*PK ID: int
Nev: nvarchar(50)
Szamlaszam: varchar(50)
Login: nvarchar(50)
Jelszo: nvarchar(50)
Email: varchar(50)
FK KozpontiTelephely: int
Statusz
«column»
*PK ID: int
Nev: nvarchar(20)
MegrendelesTetel
«column»
*PK ID: int
Mennyiseg: int
NettoAr: float
FK MegrendelesID: int
FK TermekID: int
FK StatuszID: int
Szamla
«column»
*PK ID: int
MegrendeloNev: nvarchar(50)
MegrendeloIR: char(4)
MegrendeloVaros: nvarchar(50)
MegrendeloUtca: nvarchar(50)
NyomtatottPeldanyszam: int
Sztorno: bit
FizetesiMod: nvarchar(20)
KiallitasDatum: datetime
TeljesitesDatum: datetime
FizetesiHatarido: datetime
FK KiallitoID: int
FK MegrendelesID: int
SzamlaKiallito
«column»
*PK ID: int
Nev: nvarchar(50)
IR: char(4)
Varos: nvarchar(50)
Utca: nvarchar(50)
Adoszam: varchar(20)
Szamlaszam: varchar(50)
Kategoria
«column»
*PK ID: int
Nev: nvarchar(50)
FK SzuloKategoria: int
AFA
«column»
*PK ID: int
Kulcs: int
Termek
«column»
*PK ID: int
Nev: nvarchar(50)
NettoAr: float
Raktarkeszlet: int
FK AFAID: int
FK KategoriaID: int
Leiras: xml
Kep: image
SzamlaTetel
«column»
*PK ID: int
Nev: nvarchar(50)
Mennyiseg: int
NettoAr: float
AFAKulcs: int
FK SzamlaID: int
FK MegrendelesTetelID: int

Adatmodell


class Data Model
FizetesMod
«column»
*PK ID: int
Mod: nvarchar(20)
Hatarido: int
Megrendeles
«column»
*PK ID: int
Datum: datetime
Hatarido: datetime
FK TelephelyID: int
FK StatuszID: int
FK FizetesModID: int
Telephely
«column»
*PK ID: int
IR: char(4)
Varos: nvarchar(50)
Utca: nvarchar(50)
Tel: varchar(15)
Fax: varchar(15)
FK VevoID: int
Vevo
«column»
*PK ID: int
Nev: nvarchar(50)
Szamlaszam: varchar(50)
Login: nvarchar(50)
Jelszo: nvarchar(50)
Email: varchar(50)
FK KozpontiTelephely: int
Statusz
«column»
*PK ID: int
Nev: nvarchar(20)
MegrendelesTetel
«column»
*PK ID: int
Mennyiseg: int
NettoAr: float
FK MegrendelesID: int
FK TermekID: int
FK StatuszID: int
Szamla
«column»
*PK ID: int
MegrendeloNev: nvarchar(50)
MegrendeloIR: char(4)
MegrendeloVaros: nvarchar(50)
MegrendeloUtca: nvarchar(50)
NyomtatottPeldanyszam: int
Sztorno: bit
FizetesiMod: nvarchar(20)
KiallitasDatum: datetime
TeljesitesDatum: datetime
FizetesiHatarido: datetime
FK KiallitoID: int
FK MegrendelesID: int
SzamlaKiallito
«column»
*PK ID: int
Nev: nvarchar(50)
IR: char(4)
Varos: nvarchar(50)
Utca: nvarchar(50)
Adoszam: varchar(20)
Szamlaszam: varchar(50)
Kategoria
«column»
*PK ID: int
Nev: nvarchar(50)
FK SzuloKategoria: int
AFA
«column»
*PK ID: int
Kulcs: int
Termek
«column»
*PK ID: int
Nev: nvarchar(50)
NettoAr: float
Raktarkeszlet: int
FK AFAID: int
FK KategoriaID: int
Leiras: xml
Kep: image
SzamlaTetel
«column»
*PK ID: int
Nev: nvarchar(50)
Mennyiseg: int
NettoAr: float
AFAKulcs: int
FK SzamlaID: int
FK MegrendelesTetelID: int