

# Webes alkalmazások fejlesztése

3. feladat - Hírportál

## Feladat

Készítsük el egy online portál hírkezelő rendszerét, ahol a munkatársak feltölthetik cikkeiket, amelyek megjelennek egy webes felületen.

2. részfeladat: az asztali grafikus felületet tehát a portál munkatársai használják a cikkek írására, illetve feltöltésére.

- A program használatához először be kell jelentkeznie a munkatársnak a felhasználónév és a jelszó megadásával. Ezt követően válnak elérhetővé a szerkesztési funkciók (illetve a kijelentkezés)
- A főablakban a saját cikkek listázódnak dátum szerint (cím, író, dátum), amelyeket módosíthatunk, illetve törölhetünk is.
- Új cikk felvitelénél meg kell adnunk a címet, az összefoglalót (max. 1000 karakter), valamint a teljes szöveget. Ezek kitöltése kötelező. A cikk beállítható vezető cikknek, ekkor azonban kötelező legalább egy képet feltölteni hozzá. Ezen felül feltölthetnek tetszőleges számú képet a cikkhez.
- Cikk módosításánál is ugyanezt a felületet kapjuk vissza, de már előre kitöltve.
- Cikk törlésénél a program megerősítést kér a felhasználótól

## Elemzés

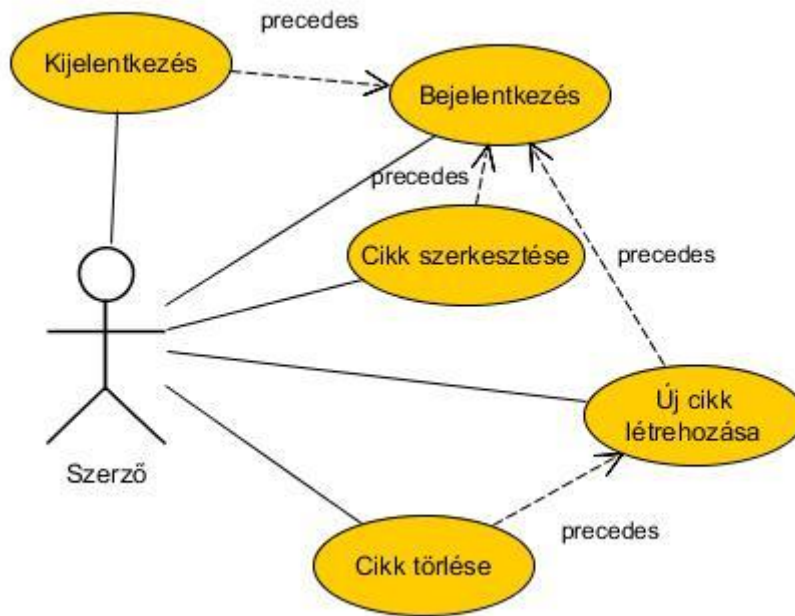
- A web api-t ASP.NET Core 2.0 MVC architektúrával készítjük el
- A perzisztencia réteg Transact-SQL, code-first Entity Framework 4.0 segítségével
- Az api frontendje Data Transfer Objectek lesznek, JSON formátumban szerializálva
- A felhasználók kezelését az ASP.NET Core-ban épített IdentityUser funkcionalitás segítségével oldjuk meg
- A kliens .NET Framework architektúrára támaszkodva WPF nézettel valósul meg.
- A kliens nem figyeli a bevitt adatok helyességét, viszont jelzi, ha a szerver visszautasította a kérést.
- Ha lejárt a munkamenet, a következő kérésnél a kliens kontroljai letiltódnak, és a program újraindítása után lehet újra bejelentkezni.

## Tervezés

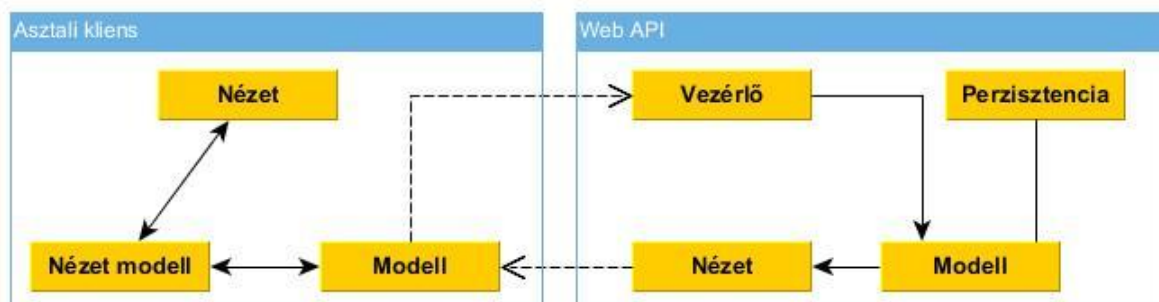
- Az előző megvalósítással ellentétben a képeket nem fájlban, hanem az adatbázisban tároljuk, hogy az api és a webes felület is elérje őket.
- Feltöltéskor ellenőrizzük, hogy a képek valóban kép formátumúak-e.

- A kliens MVVM architektúra mentén valósul meg, ahol a Model szerepét egy a webapi-val aszinkron módon kommunikáló osztály tölti be.
- A kliensben a változtatások azonnal feltöltődnek a szerverre.
- A képek átméretezése lusta módon történik lekéréskor, így sem a WPF kliensnek sem a webes API-nak nem kell foglalkoznia vele.

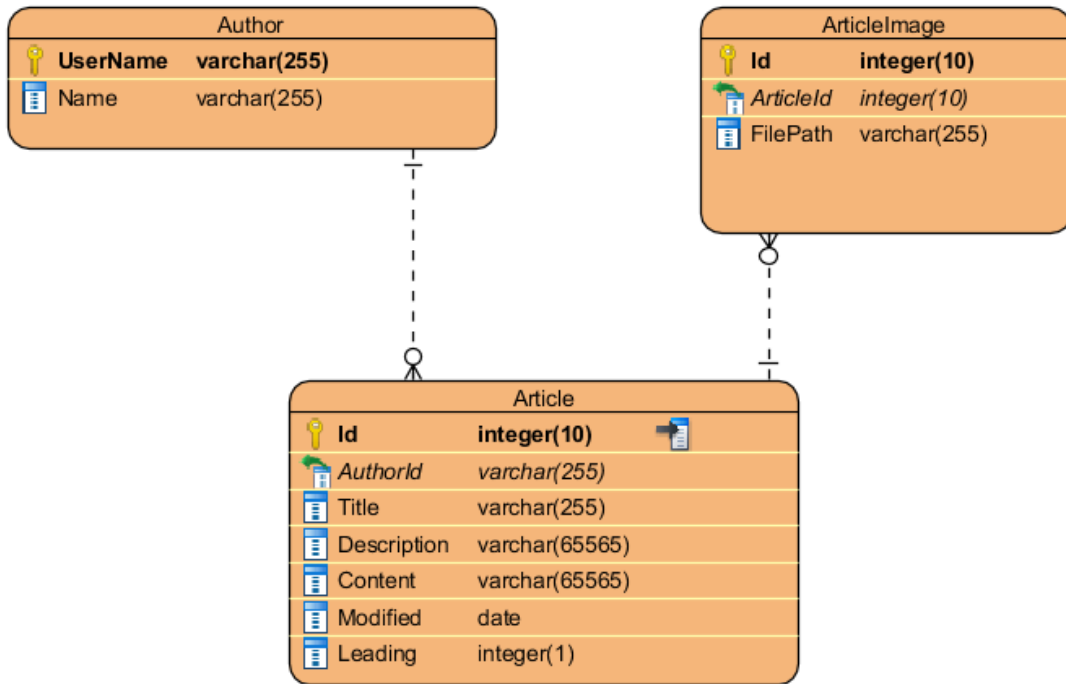
## Használati esetek



## Struktúradiagram



## Entitásdiagram



## Osztálydiagram

