

Webes alkalmazások fejlesztése

1. beadandó feladat - Hírportál (3.)

Feladat:

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

A webes felület tartalmazza magát a hírportált, ahol az olvasók tetszőlegesen böngészhetik a híreket.

- A főoldalon megjelennek a legfrissebb hírek (cím és összefoglaló, dátum szerint csökkenő sorrendben, legfeljebb 10), illetve a lap tetején kiemelten a vezető cikk (cím és összefoglaló) képpel (amennyiben több kép tartozik a cikkhez, az első jelenik meg, kicsinyített méretben).
- A címet kiválasztva megjelenik a teljes tartalom (beleértve a szerző nevét, illetve a bevitel, vagy utolsó módosítás dátumát) képpel (ha van kép a cikkhez rendelve, amennyiben több kép tartozik a cikkhez, az első jelenik meg, kicsinyített méretben). A képet kiválasztva megjelenik a cikkhez tartozó képgyűjtemény, ahol egyenként lapozhatunk a képek között, illetve visszaléphetünk a cikkhez.
- A hírportál tartalmaz egy archívumot, ahol dátum szerint csökkenő sorrendben listázódnak a hírek (cím és összefoglaló). Egy oldalon legfeljebb 20 hírt láthatunk, a többiért lapozni kell. Az archívumban lehet keresni is, megadott dátumra, cím(részlet)re, vagy tetszőleges szóra a cikk tartalmából.

A 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.

Az adatbázis az alábbi adatokat tárolja:

- felhasználó (név, azonosító, jelszó);
- cikkek (cím, szerző, dátum, összefoglaló, tartalom, vezető cikk-e);
- képek (cikk azonosító, kép).

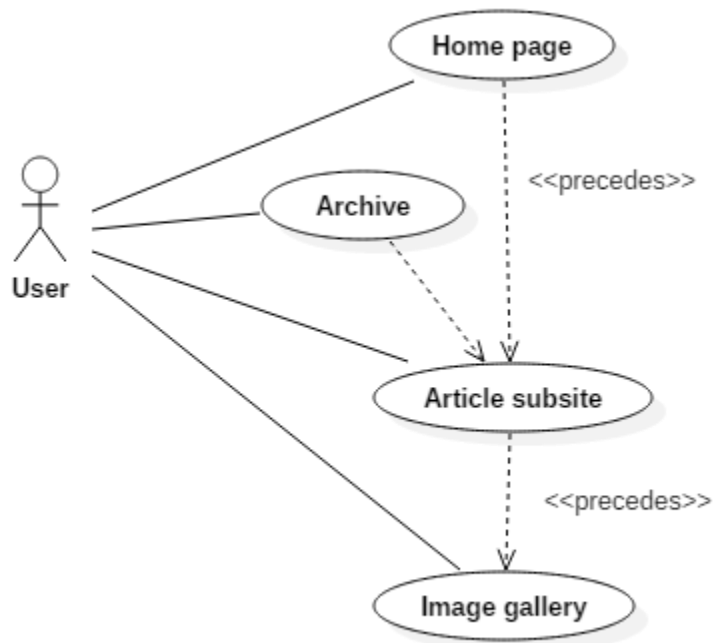
Feladat elemzése:

- ASP.NET keretrendszerben valósítjuk meg.
- 3 rétegű MVC architektúrát alkalmazunk.
- Az adatokat adatbázisban tároljuk, és a ADO.NET Entity Framework segítségével kezeljük.
- Code First megközelítést alkalmazunk az adatbázis létrehozására, ehhez 3 Modell osztályt kell definiálnunk: Account, Article, Image
- A weboldal nézetei HTML fájlok lesznek, ezeket jeleníti meg a böngésző.
- Minden controllerhez külön nézet tartozik.
- A kontrollerek felelősek azért, hogy adott URL címen melyik nézet jelenjen meg.
- A kontrollerek állítják össze a nézethez szükséges adatokat, illetve adatmódosítás esetén végrehajták a módosításokat a modellben.

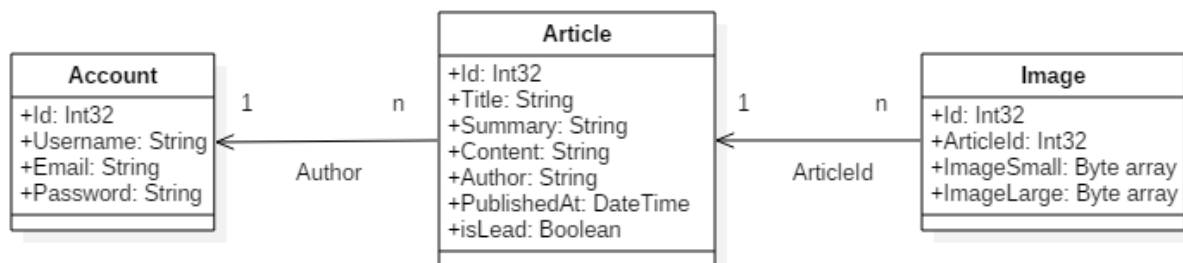
Article controller:

- **Index:** /Articles – Cikkek listázása dátum szerint csökkenő sorrendben az első 10 darabot, illetve a kiemelt cikk megjelítése kiemelten a cikkhez tartozó első képpel.
- **Read:** /Articles/Read/{id} – Adott Id-val ellátott cikk megjelítése (kép, szerző, dátum, szöveg)
- **Create:** /Articles/Create – Új cikk felvétele
- **Edit:** /Articles/Edit/{id} – Létező cikk szerkesztése
- **Delete:** /Articles/Delete/{id} – Cikk törlése
- **Archive:** /Articles/Archive – Cikkek listázása dátum szerint csökkenő sorrendben, szűrési lehetőséggel (keresés dátum, cím, összefoglaló, szöveg szerint)
- **Images:** /Articles/Images/{id} – Adott cikkhez tartozó fotók megjelítése lapozható formában.

Felhasználói esetek:



Adatbázis felépítése:



WebAPI:

Articles:

- GET /api/articles – Összes cikk lekérése
- POST /api/articles – Cikk feltöltése
- GET /api/articles/{id} – Adott id-val rendelkező cikk lekérése
- PUT /api/articles/{id} – Adott id-val rendelkező cikk frissítése
- DELETE /api/article/{id} – Adott id-val rendelkező cikk törlése

Images:

- GET /api/images/{id} – Adott id-val rendelkező kép lekérése
- POST /api/images – Kép feltöltése

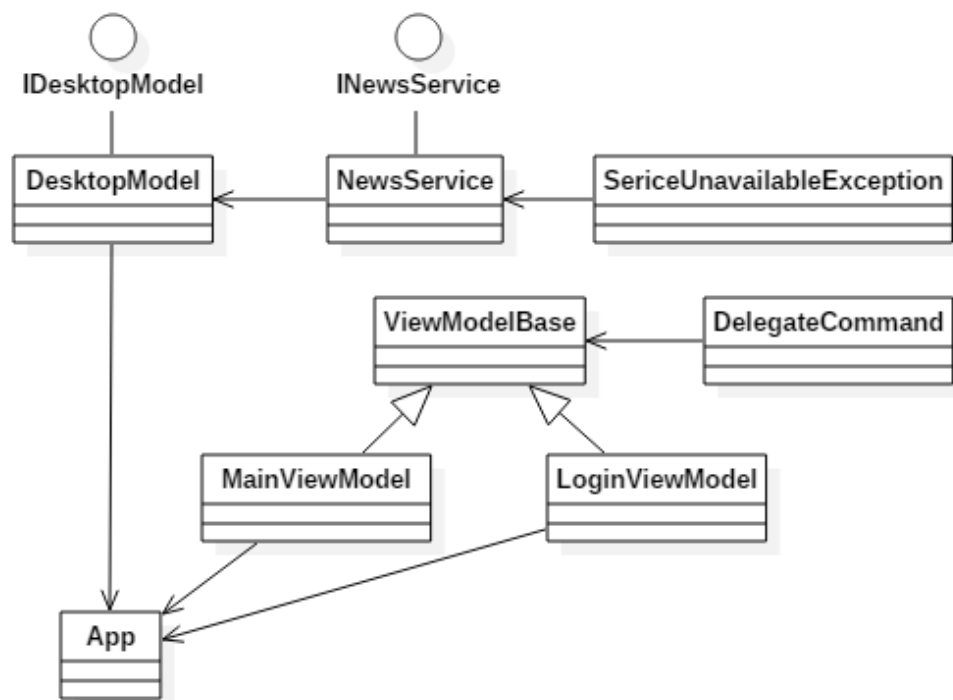
Authentication:

- GET /api/account/login/{userName}/{userPassword} - bejelentkezés
- GET /api/account/logout – kijelentkezés

Tesztesetek:

- cikk feltöltése (ellenőrizzük, hogy visszakapjuk)
- cikk szerkesztése (ellenőrizzük, hogy megváltoztak az adatok)
- cikk törlése (ellenőrizzük, hogy eltűnt)

WPF alkalmazás felépítése:



Soós Bálint
HDX9MU
2017.05.22.

WPF alkalmazás felhasználói esetei:

