

Webes alkalmazások fejlesztése

2. feladat – Aukciós portál

Feladat

Készítsünk egy aukciókkal foglalkozó online rendszert, ahol különböző tárgyakra licitálhatnak a felhasználók.

2. részfeladat: az asztali grafikus felületet az akciók hirdetői használhatják új tárgyak felvitelére, valamint a licitálás helyzetének lekérdezésére.

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)

- Hirdetőként bejelentkezhettek az alkalmazásba a felhasználónév és a jelszó megadásával, majd ezt követően lehetőségünk van a meghirdetett tárgyak követésére, új tárgy felvitelére, valamint kijelentkezésre.
- Új tárgy felvitelkor megadjuk az elnevezést, a kategóriát, a leírást, a kezdő licitösszeget, a licitálás lezárásának dátumát, továbbá feltölthetünk egy képet is a tárgyhöz.
- A meghirdetett tárgyak (a licitálás lezárási ideje szerint) listázódnak, és minden tárgynál megjelenik, hogy jelenleg mennyit ajánlottak érte. A tárgyat kiválasztva megkapjuk az összes információt képpel együtt, valamint az összes addigi licitet (dátum, név, összeg). Lehetőségünk van a licit azonnali lezárására (csak ha valaki már licitált rá, ekkor az aktuális licitáló viszi át a tárgyat).

Elemzés

- A web api-t ASP.NET Core 2.1 API 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, vagy korlátozottan figyeli a bevitt adatok helyességét, viszont jelzi, ha a szerver visszautasította a kérést. Helytelen művelet nem hajtható végre.
- Ha lejárt a munkamenet, a következő kérésnél a kliens kontroljai letiltódnak, és a program újraindítása után lehet újra bejelentkezni.

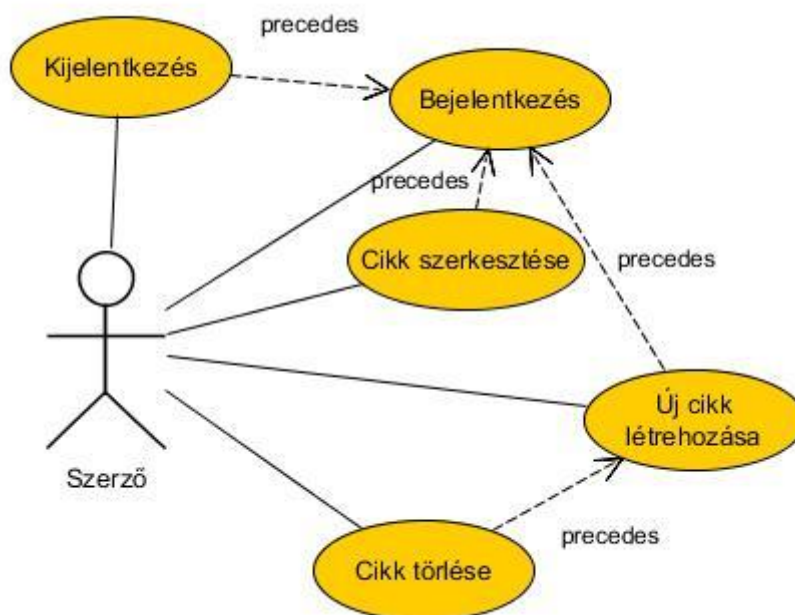
Tervezés

- 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.

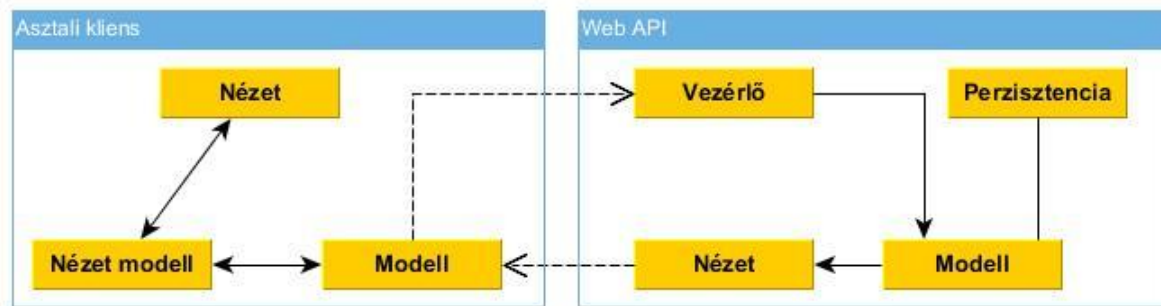
API leírás

- GET api/account/login/{username}/{password}
 - Bejelentkezés hirdetőként. Sikeres bejelentkezés esetén visszaadja a bejelentkezett felhasználó nevét (PublisherDTO típus)
- GET api/account/logout
 - Kijelentkezés. Csak bejelentkezés után lehetséges
- GET api/items
 - Csak bejelentkezés után érhető el.
 - Visszaadja a bejelentkezett felhasználó által hirdetett tárgyak előnézetét (ItemPreviewDTO tömb)
- GET api/items/{id}
 - Csak bejelentkezés után érhető el
 - Amennyiben az adott azonosítójú tárgy a bejelentkezett felhasználóhoz tartozik, visszaadjuk a részleteit (ItemDataDTO típus)
- POST api/items
 - Csak bejelentkezés után érhető el
 - A kérés törzse egy ItemDataDTO objektum, JSON formátumban szerializálva
 - Amennyiben a küldött objektum helyes, visszakapjuk az azonosítóját, ha pedig nem, akkor egy hibaüzenetet arról, hogy miért nem helyes az objektum
- PUT api/items/{id}
 - Csak bejelentkezés után érhető el
 - Amennyiben az adott azonosítójú tárgy a bejelentkezett felhasználóhoz tartozik, nem járt még le, és van licit, lezárja az adott tárgy hirdetését.
 - A művelet sikerességéről egy logikai változót kapunk válaszban

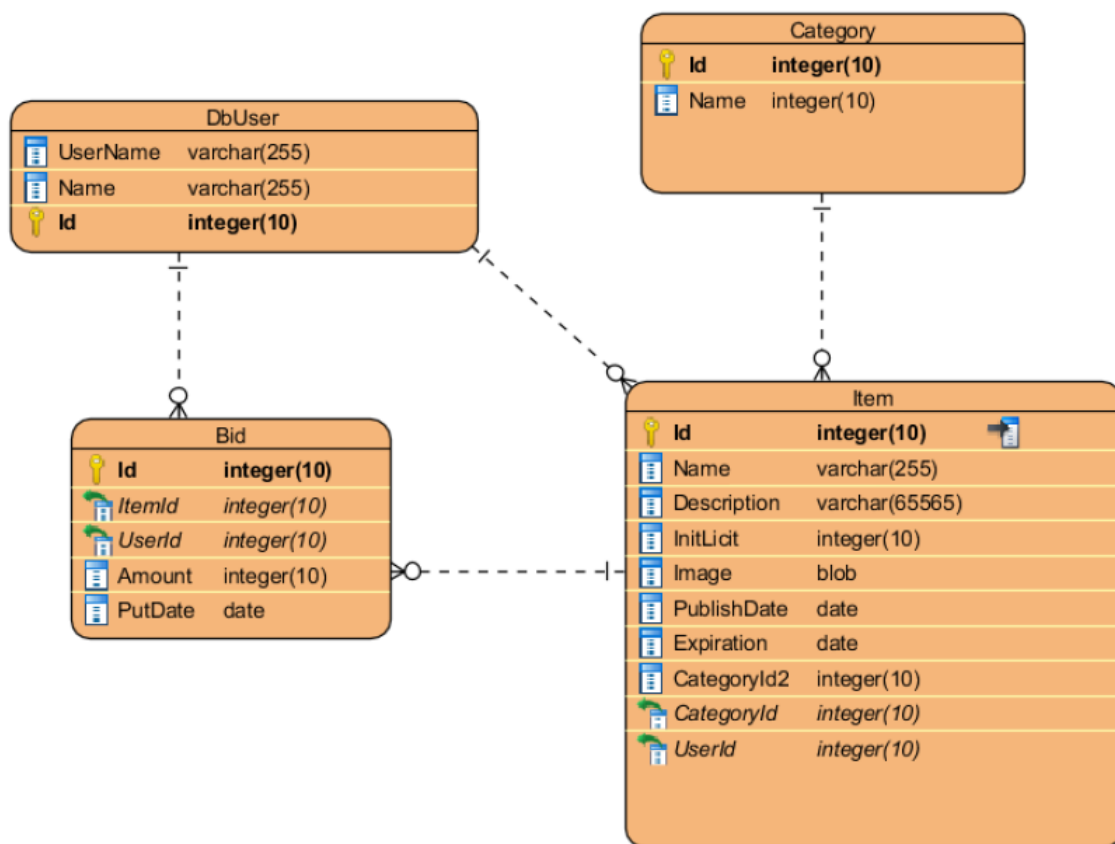
Használati esetek



Struktúradiagram



Entitásdiagram



Osztálydiagram

