

MODERN WEBALKALMAZÁSOK FEJLESZTÉSE .NET KÖRNYEZETBEN BEADANDÓ

VoteHub
Dokumentáció

Készítette:
Horánszki Patrik Donát
CJJ14N

Feladat

Látogatói felület: készítsünk REST architektúrájú webalkalmazást és hozzá webes felületű, asztali grafikus vagy mobil kliens alkalmazást, amelyen keresztül a felhasználók az aktív kérdésekben szavazhatnak, valamint megtekinthetők a korábbi szavazások eredményei.

- A felhasználók email cím és jelszó megadásával regisztrálhatnak, valamint jelentkezhetnek be. A portál további funkciói csak bejelentkezést követően érhetőek el.
- Bejelentkezést követő megjelenő felületen látható az aktív szavazások listája. Aktív az a szavazás, amely már elkezdődött, de még nem fejeződött be. A szavazásokat a befejező dátumuk szerint növekvő sorrendben kell listázni a kérdés szövegének, valamint a kezdő és befejező időpontnak a feltüntetésével. Vizuálisan legyen egyértelműen jelölve, hogy egy aktív szavazáson már szavazott-e a felhasználó vagy sem.
- Egy aktív szavazás kiválasztásával az alkalmazás jelenítse meg a kérdést és a válasz opciókat. Utóbbiak közül pontosan egyet kiválasztva lehet a szavazatot érvényesen leadni.
- A bejelentkezett felhasználók egy másik felületen kilistázhatják a már lezárt szavazásokat. Lezártnak tekintendő az a szavazás, amelynek befejező időpontja elmúlt. A szavazások listáját lehessen szűrni a kérdés szövegének részlete vagy időintervallum alapján.
- Egy lezárt szavazás kiválasztásával a weboldal jelenítse meg annak eredményét, azaz válasz opciónként a szavazatok száma és százalékos értéke.

Adminisztrációs felület: készítsünk Blazor keretrendszerre épülő kliens alkalmazást, amelyen keresztül új szavazásokat lehet kiírni a rendszerben. Bármely felhasználó írhat ki új szavazást.

- A felhasználók bejelentkezhetnek (email cím és jelszó megadásával) a programba. Sikeres bejelentkezést követően látja az általa kiírt korábbi szavazások listáját.
- Egy szavazást kiválasztva megjelenítésre kerül a feltett kérdés és a válasz opciók, valamint a szavazás kezdete és vége. Továbbá kerüljön megjelenítésre a szavazáson már részt vett felhasználók listája.
- Legyen lehetőség új szavazás kiírására a kérdés, a dinamikus számú (legalább 2) válasz opció, továbbá a kezdő és a befejező időpont

megadásával. A kezdő és a vég időpont létező kell legyen, továbbá mindkettőnek jövőbelinek kell lennie és a vég időpontnak legalább 15 perccel követnie kell a kezdő időpontot.

- A szavazás a kiírása után már nem módosítható.

Funkcionális elemzés

Regisztráció, bejelentkezés, kijelentkezés:

- Felhasználók email és jelszó megadásával regisztrálhatnak, majd bejelentkezhetnek. A hitelesítés JWT token alapú.

Aktív szavazások listázása, szavazás:

- A bejelentkezett felhasználók megtekinthetik az aktív szavazásokat, és szavazhatnak egy opcióra.

Lezárt szavazások eredményeinek megtekintése:

- A lezárt szavazások eredményei opciókra bontva, szavazatszámmal és százalékos aránnyal jelennek meg.

Szavazások szűrése:

- Lezárt szavazások szűrhetők kérdés szövege vagy időintervallum alapján.

Új szavazás létrehozása:

- Bármely bejelentkezett felhasználó új szavazást hozhat létre az admin felületen.

Felhasználói jogosultságok kezelése:

- Jogosultságok kezelése ASP.NET Identity-vel, szerepkörök támogatása.

Adatbiztonság:

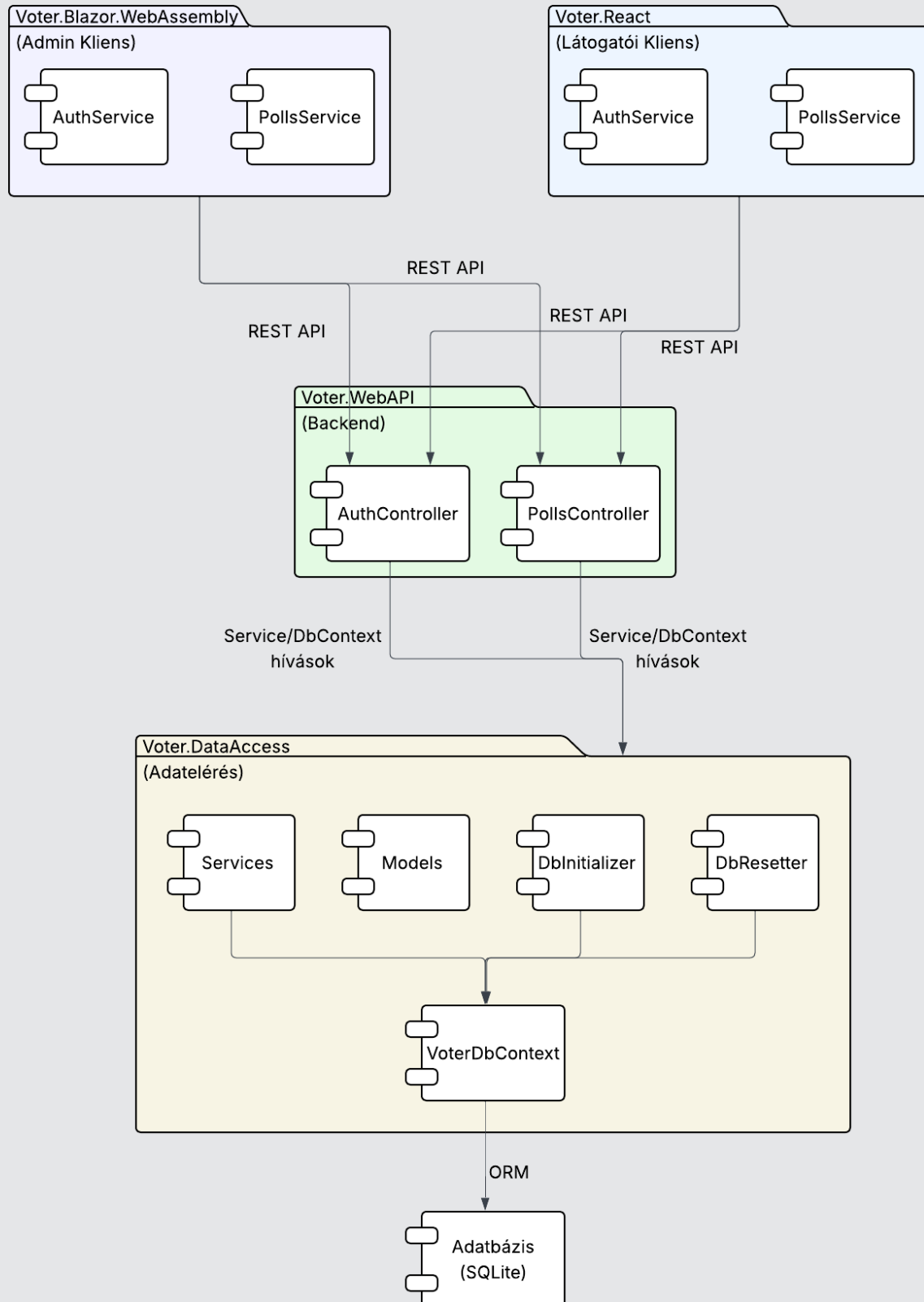
- Jelszavak hasítása, szavazatok titkos kezelése, felhasználók és szavazatok elkülönített tárolása.

Hibatűrés, hibakezelés:

- Hibás adatok, jogosulatlan hozzáférés, duplikált szavazás kezelése.

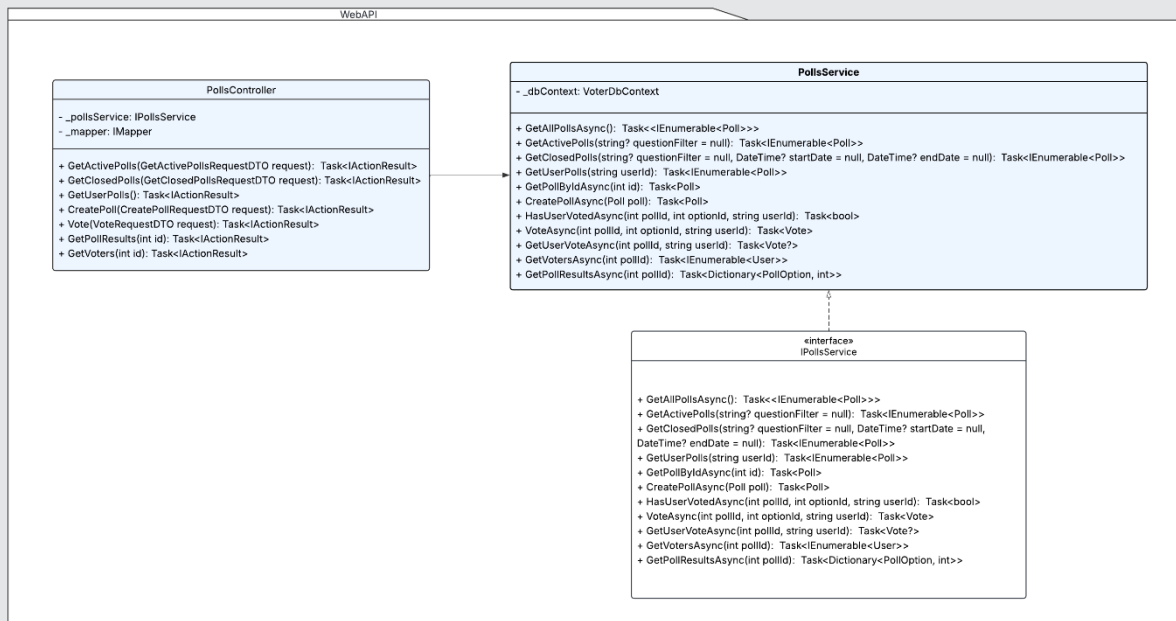
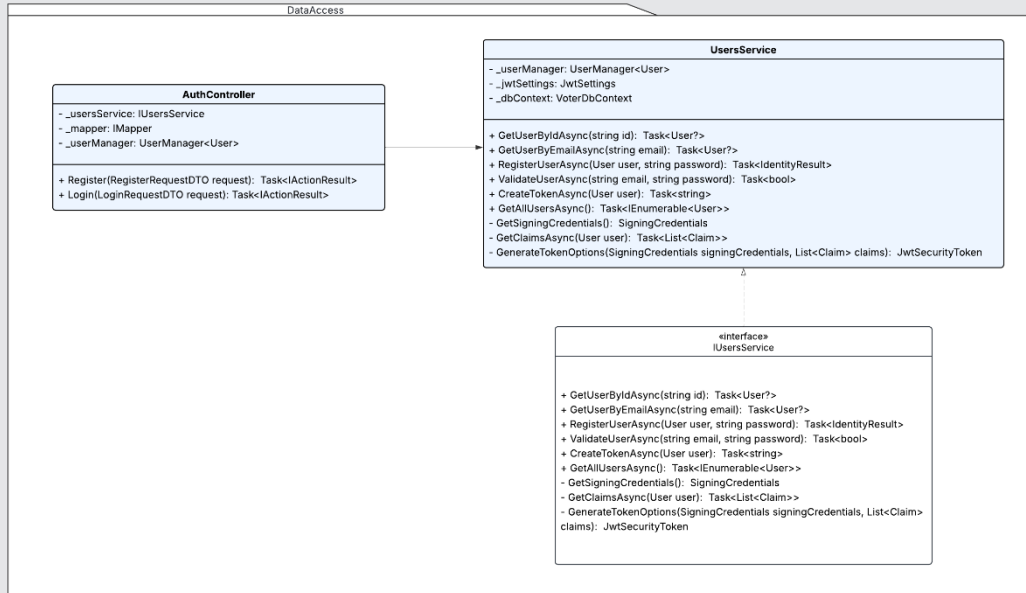
Komponensdiagram

UML



Osztálydiagram

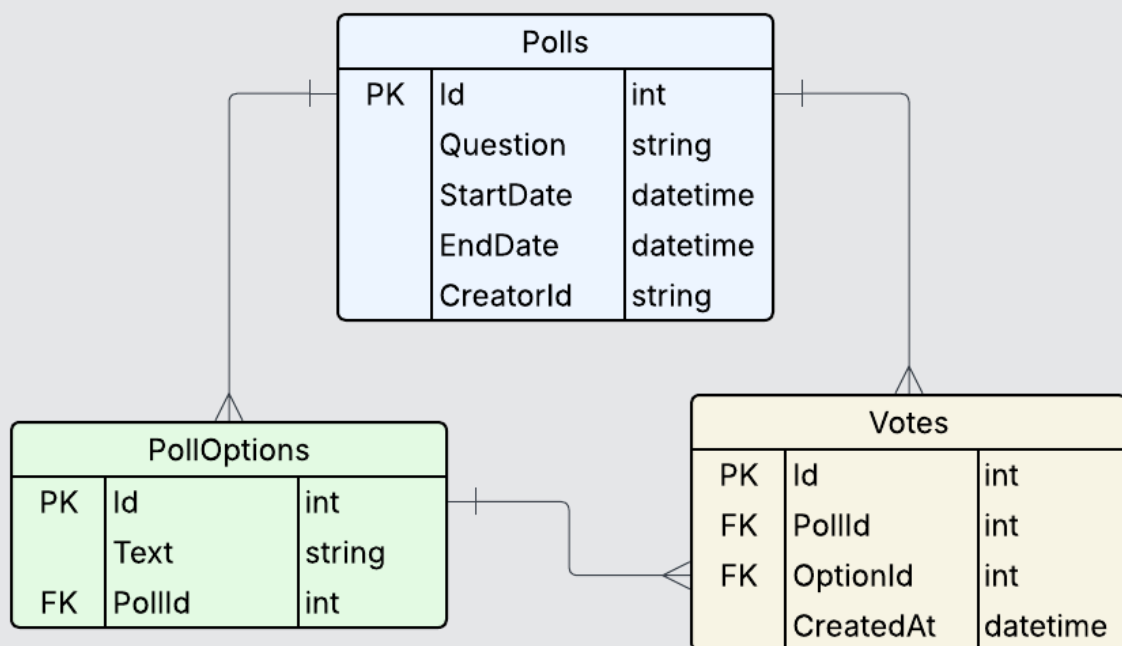
UML



Adatbázis felépítés

Az adatokat relációs adatbázisban, az ORM paradigmát követve, Entity Framework segítségével kezeljük. A főbb entitások:

- **User:** Felhasználó, ASP.NET Identity alapú, tartalmazza az azonosítót, felhasználónevet, email címet, jelszó hash-t, szerepköröket.
- **Poll:** Szavazás, tartalmazza a kérdést, kezdő és záró dátumot, létrehozó felhasználót, opciókat, szavazatokat.
- **PollOption:** Szavazási opció, egy szavazáshoz tartozik, tartalmazza az opció szövegét.
- **Vote:** Szavazat, egy felhasználó egy szavazásban egy opcióra szavazhat. A szavazat nem tartalmaz közvetlen felhasználói adatot a titkosság érdekében.



Interfészek

A backend ASP.NET WebAPI, REST architektúrával. A fő végpontok és azok részletes leírása:

AuthController

- POST /api/Auth/register – Regisztráció
 - **Bemenet:** RegisterRequestDTO (username, email, password, confirmPassword)
 - **Kimenet:** RegisterResponseDTO (success, errors)
- POST /api/Auth/login – Bejelentkezés
 - **Bemenet:** LoginRequestDTO (email, password)
 - **Kimenet:** LoginResponseDTO (success, token, user, errors)

PollsController

- GET /api/Polls/active – Aktív szavazások lekérdezése
 - **Kimenet:** PollDTO lista
- GET /api/Polls/closed – Lezárt szavazások lekérdezése (szűrhető)
 - **Bemenet:** GetClosedPollsRequestDTO (QuestionFilter, StartDate, EndDate)
 - **Kimenet:** PollDTO lista
- GET /api/Polls/user – Saját szavazások lekérdezése
 - **Kimenet:** PollDTO lista
- GET /api/Polls/{id} – Szavazás részletei
 - **Kimenet:** PollDTO
- POST /api/Polls – Új szavazás létrehozása
 - **Bemenet:** CreatePollRequestDTO (Question, StartDate, EndDate, Options)
 - **Kimenet:** PollDTO
- POST /api/Polls/vote – Szavazat leadása
 - **Bemenet:** VoteRequestDTO (PollId, OptionId)
- GET /api/Polls/{id}/results – Szavazás eredményei

- **Kimenet:** PollResultDTO
- GET /api/Polls/{id}/voters – Szavazók listája
 - **Kimenet:** UserDTO lista

Adatelérési réteg

A rendszerben a **Voter.DataAccess** projekt valósítja meg az adatelérési réteget (DAL). Ez a komponens felelős az adatbázis eléréséért, az entitások kezeléséért, valamint az üzleti logika és a perzisztencia szétválasztásáért.

Voter.DataAccess fő feladatai:

- Az adatbázis-kapcsolat és az Entity Framework konfigurációja (VoterDbContext)
- Entitásosztályok (User, Poll, PollOption, Vote, stb.) definiálása
- Migrációk kezelése (adatbázis séma verziózás)
- Adatbázis inicializálás (DbInitializer)
- Adatbázis resetelés, tesztadatok feltöltése (DbResetter)
- Adatelérési szolgáltatások (Services mappa): CRUD műveletek, lekérdezések, szavazás logika

A Data Access Layer biztosítja, hogy a backend (Voter.WebAPI) csak absztrakt módon, szolgáltatásokon keresztül érje el az adatokat, így a rendszer könnyen bővíthető és tesztelhető marad.

Backend réteg

A **Voter.WebAPI** projekt valósítja meg a rendszer backend rétegét. Ez a komponens felelős a REST API végpontok biztosításáért, a kliensalkalmazások kiszolgálásáért, a jogosultságkezelésért, valamint az üzleti logika és az adatelérési réteg összekapcsolásáért.

Voter.WebAPI fő feladatai:

- REST API végpontok megvalósítása (Controllers mappa: AuthController, PollsController)
- Jogosultságkezelés, hitelesítés (ASP.NET Identity, JWT token)
- Üzleti logika kezelése (a DataAccess szolgáltatások felhasználásával)
- DTO-k és entitások közötti leképezés (AutoMapper)
- Hibakezelés, válaszok egységesítése (pl. ProblemDetails)
- Swagger dokumentáció generálása
- Kliensalkalmazások (Blazor, React) kiszolgálása

A backend réteg biztosítja, hogy a kliensek biztonságosan, egységesen és hatékonyan érjék el a rendszer funkcióit, miközben elválasztja az üzleti logikát az adatelérési rétegtől és a felhasználói felülettől.

Tesztesetek

A projekt háromféle tesztet tartalmaz:

- **UnitTests:** Az egyes komponensek, szolgáltatások, modellek egységtesztjei. Ezek a tesztek izoláltan, külső függőségek nélkül ellenőrzik a különböző logikák helyes működését (pl. szavazás logika, validációk, DTO-mapping).
- **ControllerTests:** A WebAPI controllereinek tesztjei, amelyek ellenőrzik az egyes végpontok helyes működését, a bemeneti adatok validálását, a jogosultságkezelést és a válaszok helyességét. Ezek általában mock szolgáltatásokkal dolgoznak.
- **IntegrationTests:** A rendszer komponenseinek együttműködését vizsgáló tesztek, amelyek valós adatbázissal vagy in-memory adatbázissal futnak. Ellenőrzik a teljes folyamatokat (pl. regisztráció, bejelentkezés, szavazás leadása, eredmények lekérdezése), valamint az adatkonzisztenciát és a jogosultságokat.

Főbb tesztesetek:

- **Regisztráció, bejelentkezés, jogosultságkezelés:** Helyes és hibás adatok, duplikált email, hibás jelszó, szerepkörök kezelése.
- **Szavazás leadása:** Egy felhasználó csak egyszer szavazhat egy szavazáson, hibás opcióra nem lehet szavazni, jogosultságok ellenőrzése.
- **Szavazások listázása, szűrése:** Aktív és lezárt szavazások helyes szűrése, szavazott állapot megjelenítése.
- **Eredmények helyes megjelenítése:** Szavazatok száma, százalékos arány, saját szavazat kiemelése.
- **Admin funkciók:** Új szavazás létrehozása, opciók validálása (pl. minimum két opció), jogosultságok.
- **Hibakezelés:** Jogosulatlan hozzáférés, hibás adatok, nem létező szavazás vagy opció, megfelelő hibakódok és üzenetek.
- **Integrációs tesztek:** WebAPI végpontok helyes működése, adatkonzisztencia, teljes folyamatok tesztelése (end-to-end).
- **Felület tesztek (külön projektben):** Felhasználói élmény, hibakezelés, visszajelzések.