

Beadandó dokumentáció

Név: Antal Krisztián

Neptun-kód: LUJD4D

E-mail: antal.krisztian12@gmail.com

Feladatleírás:

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 mindkettőnek jövőbelinek kell lennie és a vég időpontnak legalább 15 perccel követnie kell a kezdő időpontot.
- A szavazás a kiírása után már nem módosítható.

Választott részfeladatok:

Több opció választása (1 pont):

Az adminisztrációs felületen a szavazások kiírója határozhassa meg a minimálisan és maximálisan megjelölhető válasz opciók számát. Legalább 1 válasz opció mindig legyen megadható, a maximális érték pedig nem lehet nagyobb a válasz lehetőségek számánál. A látogatói felületen minden szavazásnál kerüljön feltüntetésre a minimálisan és maximálisan megjelölhető válasz opciók száma és az alkalmazás ne engedjen érvénytelen szavazatot leadni.

Elfelejtett jelszó (0,5 pont):

A látogatói felületen legyen lehetőség jelszó emlékeztető kérni az email cím megadásával. A rendszer ilyenkor email üzenetben küldjön egy egyedi URL-t, amelyen az új jelszó beállítható. Az új jelszó beállításakor az email címet ne kelljen már újból megadni, az emailben küldött URL pedig csak bizonyos ideig legyen felhasználható. Megjegyzés: ha a látogatói felület nem webes, az URL helyett alternatív megoldásként egy speciális beállító kódot küldjünk, amellyel az asztali / mobil alkalmazásban az új jelszó beállítható. Ez a speciális beállító kód ekkor is rendelkezzen lejáratí idővel.

Szerkeszthető szavazások (0,5 pont):

Az adminisztrációs felületen legyen lehetőség a már kiírt saját szavazások szerkesztésére. Legyen minden beállítás módosítható, amennyiben a szavazás még nem kezdődött el a kezdeti időpont alapján. Ha már érkezett szavazat, csak a szavazás befejező időpontja legyen módosítható, és csak az aktuális beállításnál későbbi időpont legyen megadható.

Feladat funkcionális elemzése:

Voting System projekt - Funkcionális elemzés

1. Bevezetés

A Voting System projekt egy webes REST architektúrájú alkalmazás, amely két fő felhasználói szerepet támogat: látogatókat és szavazásokat kiíró felhasználókat. Az alkalmazás lehetővé teszi a felhasználók regisztrációját, bejelentkezését, szavazásokban való részvételt, valamint a szavazások eredményeinek megtekintését. Emellett biztosít adminisztrációs felületet új szavazások kiírására és szerkesztésére.

2. Látogatói felület funkcionalitása

- **Regisztráció és bejelentkezés:**

- A felhasználók email és jelszó megadásával regisztrálhatnak.
- A regisztrált felhasználók email és jelszó megadásával bejelentkezhetnek.
- Csak bejelentkezett felhasználók férhetnek hozzá a további funkciókhoz.

- **Aktív szavazások megjelenítése:**

- Az aktív (elkezdődött, de még nem befejeződött) szavazások listázása.
- Rendezés befejezési dátum szerint növekvő sorrendben.
- A listában megjelenik a kérdés szövege, kezdő- és végidőpontja.
- Vizuális jelölés, ha a felhasználó már szavazott.

- **Szavazás leadása:**

- Egy aktív szavazás kiválasztásakor megjelennek a válasz opciók.
- A felhasználó a szabályok szerint (pl. pontosan egy, vagy minél-meddig) szavazhat.
- Az alkalmazás csak érvényes szavazatot enged leadni.

- **Lezárt szavazások megtekintése:**

- Listázás szöveg-részlet és időintervallum alapján szűrhetően.
- Egy szavazás kiválasztásakor megjelenik az eredmény:

- Válaszonként a szavazatok száma és százalékos aránya.

- **Elfelejtett jelszó funkció:**

- Email megadásával lehet jelszóemlékeztetőt kérni.
- A rendszer egy egyedi URL-t vagy kódot küld, amivel az új jelszó beállítható.
- A link vagy kód lejáratí idővel rendelkezík.

3. Adminisztrációs felület funkcionalitása

- **Bejelentkezés:**

- Email és jelszó megadásával.
- Sikeres belépés után a felhasználó saját szavazásait látja.

- **Szavazás megtekintése:**

- Kiválasztva megjelenik:
 - A kérdés szövege
 - Válasz opciók
 - Kezdés és befejezés időpontja
 - Résztvevő felhasználók listája

- **Új szavazás kiírása:**

- Kérdés megadása
- Legalább 2 válasz opció dinamikus megadása
- Kezdő időpontnak jövőbelinek kell lennie
- Befejezési időpont legalább 15 perccel később kell legyen
- Kiírás után a szavazás nem módosítható

- **Szerkeszthető szavazások:**

- Csak akkor szerkeszthető, ha a szavazás még nem kezdődött el.

- Ha már van leadott szavazat:
 - Csak a befejezési időpont módosítható
 - Az új időpontnak későbbinek kell lennie
- **Több opció választása:**
 - Admin meghatározhatja a minimum és maximum kijelölhető opciók számát
 - A minimum legalább 1, maximum nem haladhatja meg az opciók számát
 - A látogatói felület is figyelembe veszi ezeket a szabályokat és érvénytelen szavazatot nem fogad el

4. Összegzés

A Voting System funkcionális specifikációja átfogóan lefedi a digitális szavazás folyamatait, az egyszerű felhasználói szavazástól kezdve az adminisztratív funkcionalitásig. A különböző szerepköröknek megfelelő felhasználói élményt biztosít, miközben érvényesíti a szavazás szabályait és időbeli korlátait is.

Felhasználói felület:

Technológiák:

- **React 18+** – komponensalapú felhasználói felület
- **Axios** – API hívásokhoz
- **React Router** – kliensoldali útvonalkezeléshez
-

Funkciók:

- Felhasználói bejelentkezés és regisztráció (login.tsx, register.tsx)
- Aktív és lezárt szavazások listázása (ActivePollsList.tsx, ClosedPollsList.tsx)
- Szavazás részleteinek megtekintése (PollsDetails.tsx)
- Szavazatok leadása (ActivePollsPage.tsx)

- Elfelejtett jelszó (ForgotPassword.tsx, ResetPassword.tsx)

API kommunikáció:

- A frontend az ASP.NET Core Web API-t használja adatlekéréshez és -küldéshez
- JWT token alapú autentikáció biztosítja a biztonságos kommunikációt

Adminisztrációs felület

Projekt neve/mappa: VotingSystem.AdminClient

Technológiák:

- **Blazor WebAssembly** – kliensoldali .NET alapú alkalmazás
- **.NET 8** – futtatókörnyezet
- **HttpClient** – REST API hívásokhoz
- **JWT autentikáció** – a biztonságos kommunikációhoz a backenddel

Funkcionalitás és fájlstruktúra

1. Bejelentkezés

- **Fájl:** Pages/Login.razor
 - Felhasználónév/jelszó beviteli mezők
 - Token alapú hitelesítés a backenddel
 - Navigáció sikeres belépés után

2. Admin főoldal / Admin által kreált szavazások listája

Fájl: Pages/MyPolls.razor

- Szavazások kezelése
- Szavazások részleteinek megtekintése
- Szavazások szerkesztése

3. Szavazások kezelése

- **Fájl:** Layout/EditPollModal.razor
 - Létező szavazások módosítása

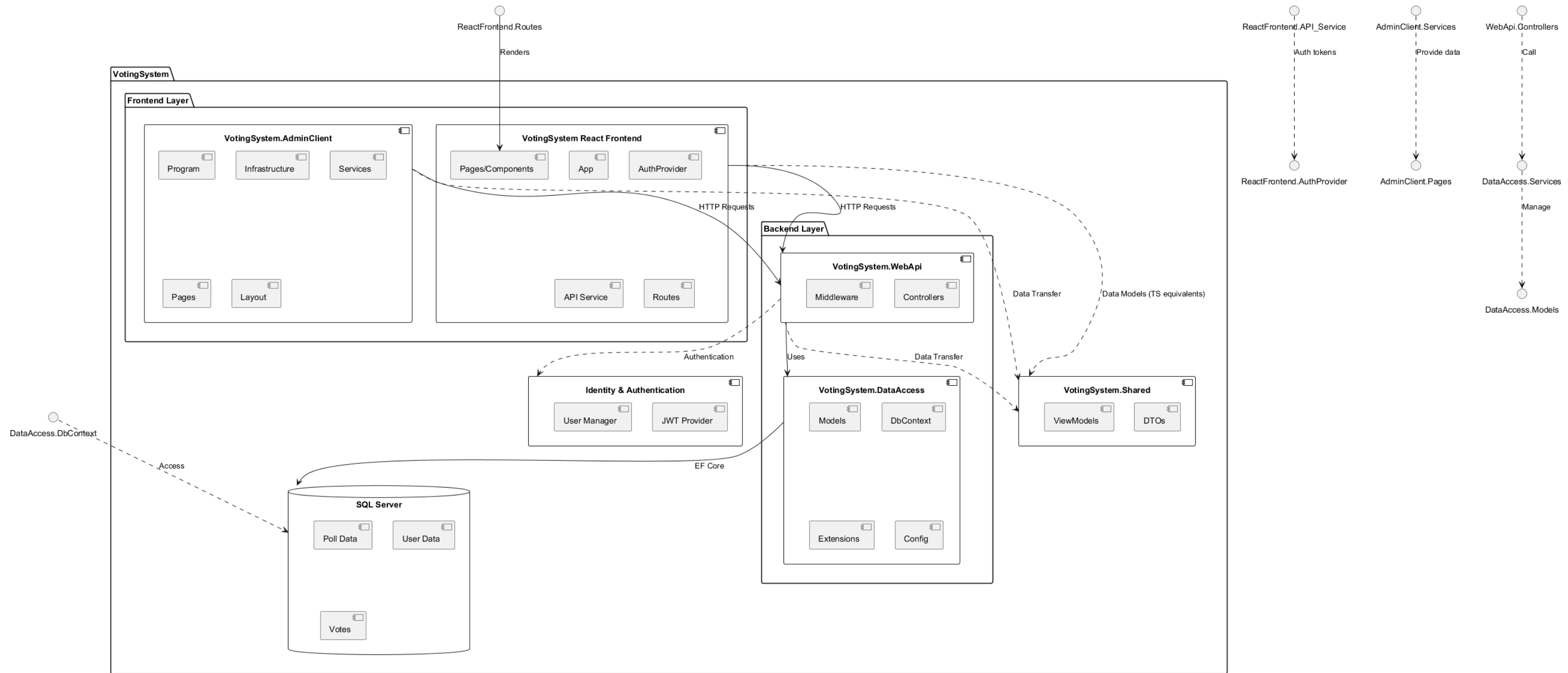
5. Navigáció és elrendezés

- **Fájl:** Layout/NavMenu.razor
 - Oldalsó vagy felső navigációs sáv
 - Linkek: Főoldal, Szavazások, Felhasználók, Kijelentkezés

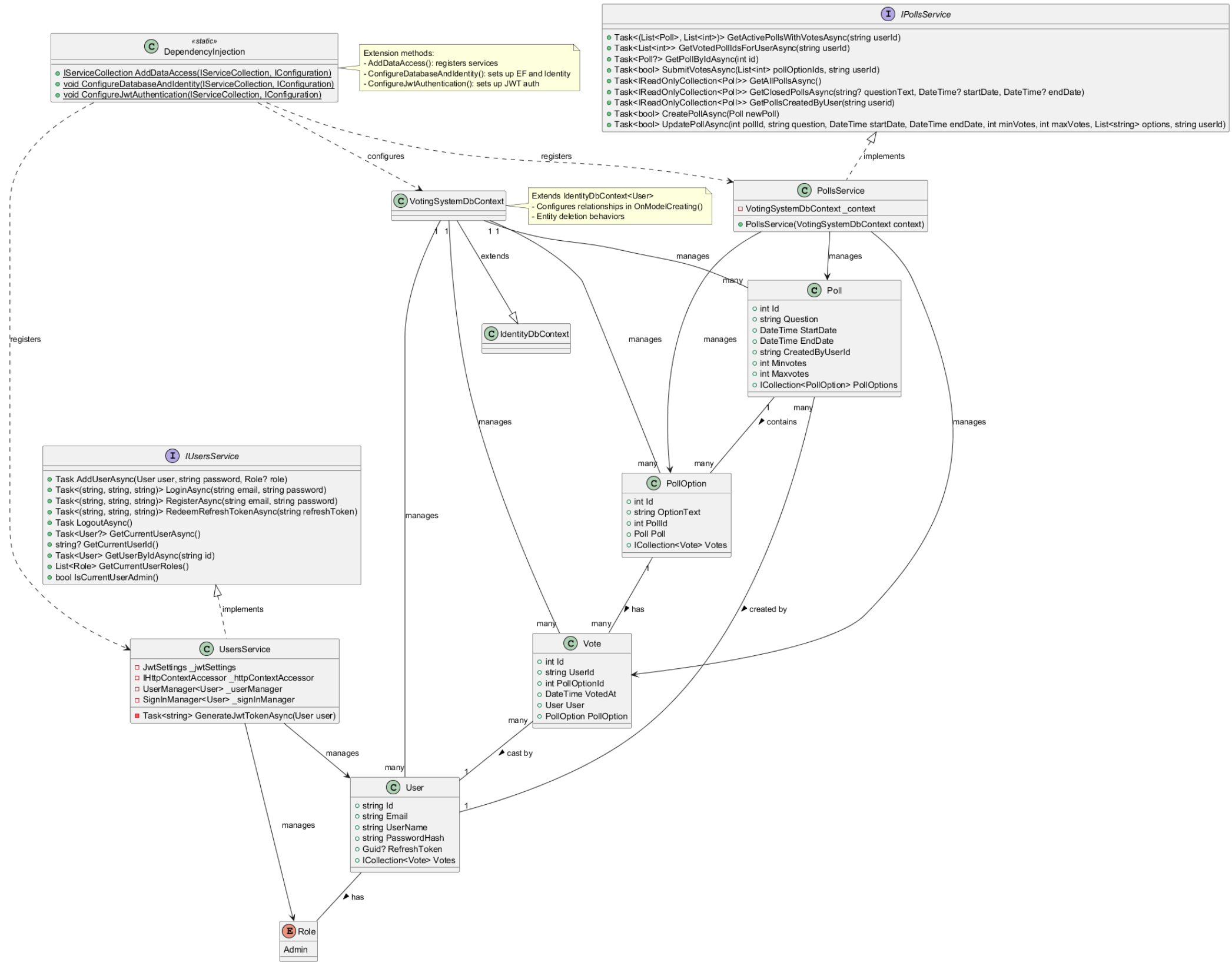
Autentikáció

- A frontend JWT tokenet használ, amit a bejelentkezés után tárol.
- Az API hívások Authorization: Bearer <token> fejléccel történnek.
- Tokenkezelés és validáció: Services/AuthService.cs

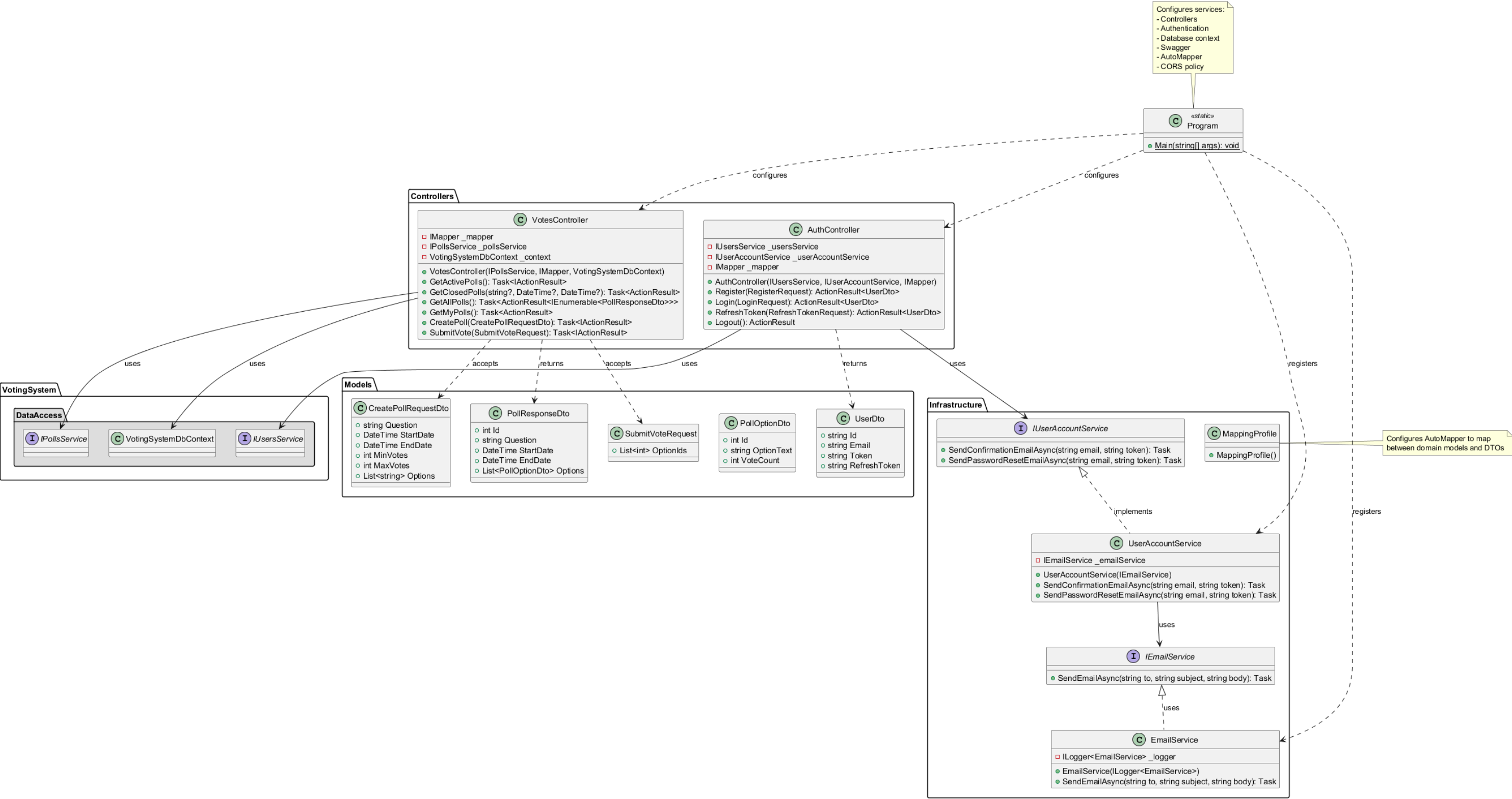
A projekt komponendiagramja



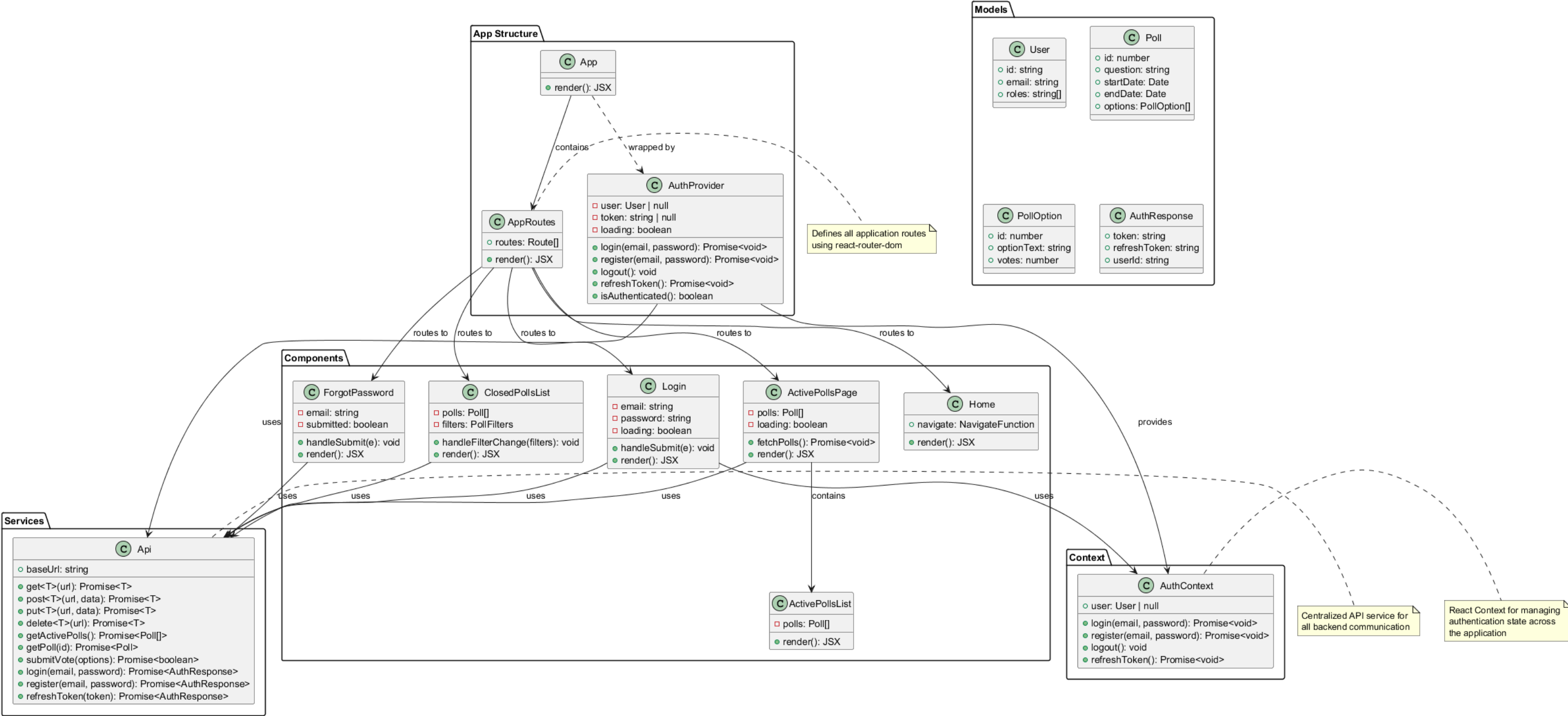
A DataAccess projekt osztálydiagramja:



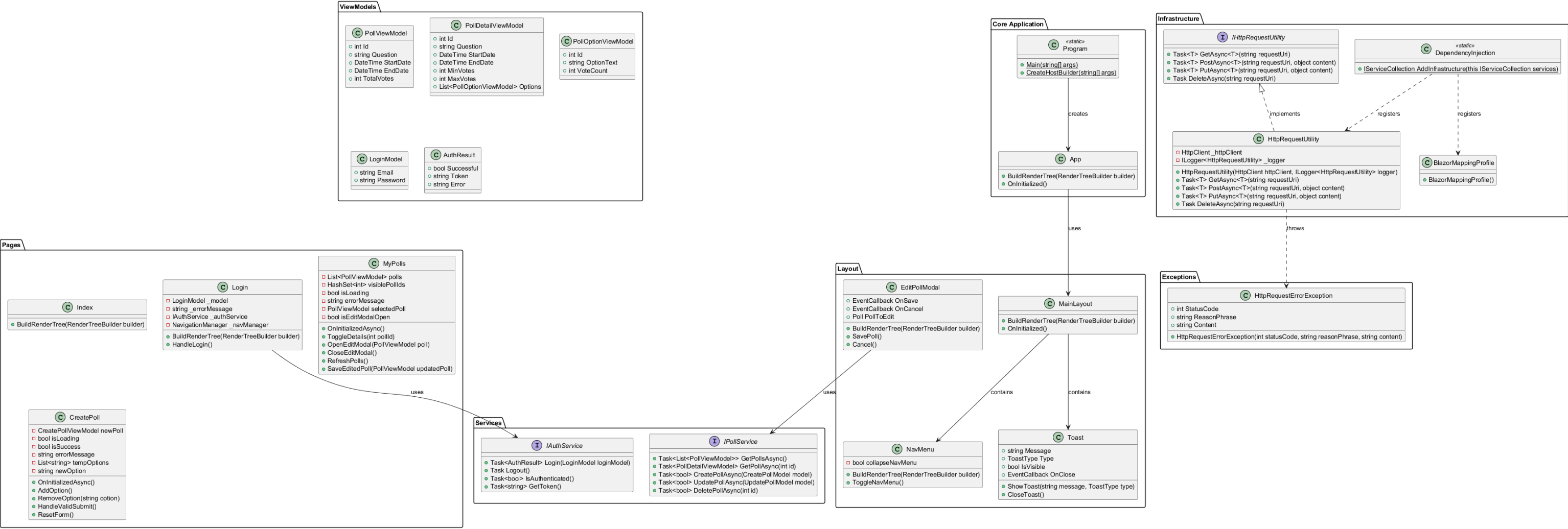
WebApi osztálydiagramja:



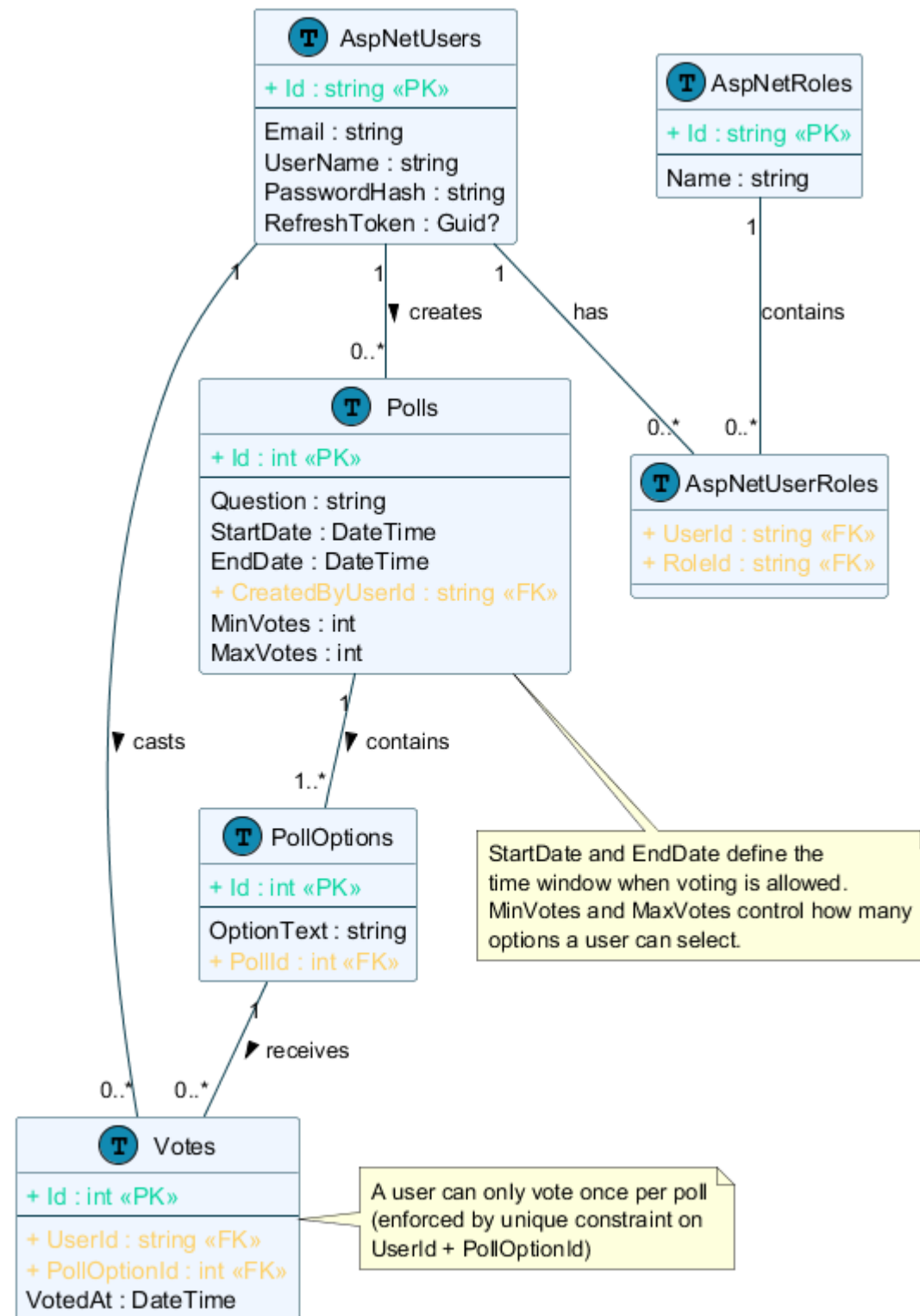
A felhasználói felület (React frontend) diagramja:



Az adminisztrációs felület (Blazor frontend) osztálydiagramja:



Az adatbázis sémadiagramja:



Szavazó rendszer webszolgáltatás felületének a leírása

Hitelesítési Végpontok

1. Felhasználói Regisztráció

- **URL:** /api/auth/register
- **Metódus:** POST
- **Leírás:** Új felhasználó regisztrálása a rendszerbe
- **Kérés Törzse:** {

```
"email": "felhasznalo@pelda.com",  
"password": "Jelszo123"  
}
```

Válasz:

```
{  
  "id": "felhasznalo-guid",  
  "email": "felhasznalo@pelda.com",  
  "token": "jwt-hitelesitesi-token",  
  "refreshToken": "frissitesi-token-guid" }
```

Státuszkódok: 200 Sikeres, 400 Hibás Kérés

2. Felhasználói Bejelentkezés

- **URL:** /api/auth/login
- **Metódus:** POST
- **Leírás:** Felhasználó hitelesítése és tokenek visszaadása
- **Kérés Törzse:**

```
{  
  "email": "felhasznalo@pelda.com",  
  "password": "Jelszo123"  
}
```

- **Válasz:** Azonos a regisztrációval
- **Státuszkódok:** 200 Sikeres, 401 Jogosulatlan

3. Token Frissítés

- **URL:** /api/auth/refresh
- **Metódus:** POST
- **Leírás:** Új JWT token kérése frissítési token használatával
- **Kérés Törzse:**

```
{  
  "refreshToken": "frissitesi-token-guid"  
}
```

- **Válasz:** Azonos a bejelentkezéssel
- **Státuszkódok:** 200 Sikeres, 401 Jogosulatlan

4. Kijelentkezés

- **URL:** /api/auth/logout
- **Metódus:** POST
- **Leírás:** A felhasználó tokenjének érvénytelenítése
- **Hitelesítés:** Bearer token szükséges
- **Státusz kódok:** 200 Sikeres, 401 Jogosulatlan

Szavazás Végpontok

1. Aktív Szavazások Lekérdezése

- **URL:** /api/votes/active
- **Metódus:** GET
- **Leírás:** Összes aktív szavazás lekérése az opciókkal együtt
- **Hitelesítés:** Bearer token szükséges
- **Válasz:**

```
[  
  {  
    "id": 1,  
    "question": "Szavazás kérdése?",  
    "startDate": "2025-05-01T12:00:00Z",  
    "endDate": "2025-05-15T12:00:00Z",  
    "minVotes": 1,  
    "maxVotes": 3,  
    "options": [  
      {
```

```
    "id": 101,  
    "optionText": "1. lehetőség"  
  },  
  {  
    "id": 102,  
    "optionText": "2. lehetőség"  
  }  
],  
"hasVoted": false  
}  
]
```

- **Státuszkódok:** 200 Sikeres, 401 Jogosulatlan

2. Lezárt Szavazások Lekérdezése

- **URL:** /api/votes/closed
- **Metódus:** GET
- **Paraméterek:**
 - questionText (opcionális): Szűrés kérdés szövege alapján
 - startDate (opcionális): Szűrés kezdési dátum alapján
 - endDate (opcionális): Szűrés befejezési dátum alapján
- **Leírás:** Lezárt szavazások lekérése eredményekkel
- **Hitelesítés:** Bearer token szükséges
- **Válasz:**

[

```
{
  "id": 2,
  "question": "Korábbi szavazás kérdése?",
  "startDate": "2025-04-01T12:00:00Z",
  "endDate": "2025-04-15T12:00:00Z",
  "options": [
    {
      "id": 201,
      "optionText": "1. lehetőség",
      "voteCount": 5
    },
    {
      "id": 202,
      "optionText": "2. lehetőség",
      "voteCount": 10
    }
  ]
}
```

- **Státuszkódok:** 200 Sikeres, 401 Jogosulatlan

3. Szavazás Részletei

- **URL:** /api/votes/{id}
- **Metódus:** GET

- **Leírás:** Részletes információ egy adott szavazásról
- **Hitelesítés:** Bearer token szükséges
- **Válasz:** Hasonló az aktív szavazásokhoz, de további részletekkel
- **Státuszkódok:** 200 Sikeres, 404 Nem Található, 401 Jogosulatlan

4. Szavazat Leadása

- **URL:** /api/votes/submit
- **Metódus:** POST
- **Leírás:** Szavazat leadása szavazási opciókra
- **Hitelesítés:** Bearer token szükséges
- **Kérés Törzse:**

```
{
  "optionIds": [101, 102]
}
```

- **Válasz:**

```
{
  "success": true
}
```

- **Státuszkódok:** 200 Sikeres, 400 Hibás Kérés, 401 Jogosulatlan

5. Szavazás Létrehozása (Csak Adminoknak)

- **URL:** /api/votes/create
- **Metódus:** POST
- **Leírás:** Új szavazás létrehozása
- **Hitelesítés:** Bearer token szükséges (Admin szerep)

- **Kérés Törzse:**

```
{  
  "question": "Új szavazás kérdése?",  
  "startDate": "2025-05-20T12:00:00Z",  
  "endDate": "2025-06-01T12:00:00Z",  
  "minVotes": 1,  
  "maxVotes": 2,  
  "options": ["1. lehetőség", "2. lehetőség", "3. lehetőség"]  
}
```

- **Válasz:**

```
{  
  "success": true,  
  "pollId": 3  
}
```

- **Státuszkódok:** 200 Sikeres, 400 Hibás Kérés, 401 Jogosulatlan, 403 Tiltott

6. Saját Szavazások Lekérdezése

- **URL:** /api/votes/mypolls
- **Metódus:** GET
- **Leírás:** Az aktuális felhasználó által létrehozott szavazások lekérése
- **Hitelesítés:** Bearer token szükséges
- **Válasz:** Szavazás objektumok tömbje
- **Státuszkódok:** 200 Sikeres, 401 Jogosulatlan

Hitelesítés

- JWT alapú hitelesítés
- Token az Authorization fejlécben: Authorization: Bearer {token}
- Frissítési token folyamat a token megújításához
- Szerepkör alapú hozzáférés-vezérlés az admin funkciókhoz

Hibakezelés

Minden végpont szabványos hibaválaszokat ad vissza:

```
{  
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",  
  "title": "Hiba címe",  
  "status": 400,  
  "detail": "Részletes hibaüzenet",  
  "errors": {  
    "mező": ["Hibaüzenet egy adott mezőre"]  
  }  
}
```

Tesztek

VotingSystem.ComponentTest

- CreatePollTests.cs
A CreatePoll komponens egységtesztjei: a űrlap megjelenítését, adatbevitel-ellenőrzést és a beküldési események kiváltását vizsgálja.
- MainLayoutTests.cs
Az alkalmazás fő elrendezését (header, footer, navigációs sáv) teszteli: a statikus és dinamikus elemek helyes renderelését ellenőrzi.
- MyPollsTest.cs
A „My Polls” lista-komponens működését vizsgálja: listaelemek betöltése, üres állapot kezelése és kattintási események tesztelése.
- NavMenuTests.cs
A baloldali navigációs menü tesztjei: menüpontok száma, URL-ek helyessége és az aktív elem kijelölése.

VotingSystem.IntegrationTest

- PollIntegrationTests.cs
A /api/polls végpontok integrációs tesztjei: GET, POST, PUT és DELETE hívások valódi HTTP-környezetben.
- PollsServiceTests.cs
A PollsService alapvető CRUD-műveleteit teszteli: adatlekérdezés, létrehozás, frissítés, törlés.
- PollsServiceEdgeCaseTests.cs
A PollsService szélsőséges és hibás bemenetek (pl. üres adatok, nem létező ID-k) kezelését ellenőrzi.
- UsersServiceTests.cs
A UsersService normál CRUD-folyamatait teszteli: felhasználó létrehozása, lekérdezése, módosítása és törlése.
- UsersServiceEdgeCaseTests.cs
A UsersService rosszindulatú vagy hiányos bemenetekre (hibás mezők, nem létező felhasználó) adott válaszait vizsgálja.

