

Dokumentáció

Rendszerterv

Jelölő, szavazó rendszer készítése

Szoftverarchitektúrák tárgy házi feladat

Készítették:

Barta Ágnes

Cseppentő Lajos

Tartalom

A rendszer célja, funkciói és környezete.....	3
Feladatkiírás.....	3
A rendszer által biztosítandó tipikus funkciók.....	3
A program környezete	4
Megvalósítás	5
Architektúra	5
Adatbázis réteg	6
Adatdefiníciók.....	6
Adathozzáférési réteg.....	7
Üzleti logikai réteg	7
REST Service	7
Tesztelés.....	9
Grafikus felület (Kliens).....	9
Felhasználói grafikus felület.....	9
Adminisztrátori grafikus felület	15
Adat- és adatbázis-terv	19
A program objektum modellje.....	19
Az objektumok (POCO) leírása	19
Az adatbázis	20
Administrator	20
User	21
Poll	21
Nomination	21
Vote.....	21
PollSubject.....	22
News.....	22
GUI terv	22
A program készítése során felhasznált eszközök.....	22
Összefoglalás.....	23

A rendszer célja, funkciói és környezete

Feladatkiírás

Egy olyan weboldal készítése a feladat, amellyel a jelölős-szavazós-díjnyerős események kezelhetők, pl. a rajongói sorozatos díjak (pl. <http://www.sorozatjunkie.hu/tag/jaws/>). A jelölési fázisban nem egy listából kell választani (mivel az túl hosszú lenne), hanem szövegdobozos beírással, amit validáció illetve auto-complete segít. A szavazási fázisban már csak a legtöbb szavazatot kapott jelöltek közül lehet választani. Lehesen követni a szavazás állását és a rendszer lehetőleg védje ki az egyszerűbb csalási módszereket

A rendszer által biztosítandó tipikus funkciók

A projekt során célunk egy olyan jelölő-szavazó alkalmazás készítése, amely segítségével a felhasználó képes filmeket és sorozatokat jelölni egy adott szavazásra, majd a szavazás megnyílása után képes ezekre szavazni.

A program fő funkciója a jelölési és szavazási folyamat megvalósítása. A folyamat a következő lépésekből áll:

- Egy adminisztrátor kiír egy jelölés-szavazást, melyet az összes látogató lát.
- A bejelentkezett felhasználók jelölhetnek a jelölési határidőig:
 - Meg kell nevezniük a jelöltet, valamint opcionálisan egy rövid indoklást is írhatnak.
 - Egy felhasználó tetszőleges számú filmet vagy sorozatot jelölhet egy szavazásra.
 - Jelölni csak létező filmet, ill. sorozatot lehet.
 - A jelölési időszak végéig a felhasználó módosíthatja a saját jelöléseit és akár törölheti is azokat.
- Jelölések összegzése, amely magában foglalja:
 - a beküldött jelölések felülvizsgálatát,
 - az irreleváns jelölések törlését és
 - az esetleges duplikátumok kiszűrését.
- Szavazás, ami magában foglalja:
 - a jelöltek nyilvánosságra hozását,
 - a szavazás aktiválását és
 - a szavazatok leadását, amely során minden bejelentkezett felhasználó pontosan egy jelöltre szavazhat.
- Szavazás lezárása:
 - Szavazni egy előre megadott időpontig lehet.
 - A határidő letelte után inaktívvá válik a szavazás.
- Eredményhirdetés:
 - Az eredményhirdetési időpont után a szavazás eredménye az összes látogató számára láthatóvá válik.

Az alkalmazásnak lehetővé kell tennie, hogy egyszerre párhuzamosan tetszőleges számú aktív jelölési-szavazási folyamat lehessen a rendszerben. Emellett a szoftverben egyéb funkciókat is meg kell valósítani, melyek a következők:

- Alapszintű védelem csalás ellen:
 - Facebook fiókos felhasználói beléptetés alkalmazása.
 - Gyanús felhasználói fiókok adminisztrátor általi kitiltásának a lehetősége.
- Hírek:
 - Lehesen híreket kiküldeni a kezdőoldalra, melyeket a látogatók szabadon olvashatnak.

- „Gyakran ismételt kérdések” (GYIK) és „Kapcsolat” címmel két statikus oldal.

A program környezete

A definiált alkalmazás felhasználói felülete *HTML5* alapokon készült el, így bármely modern böngészőből és okostelefonról is használható, ezzel biztosítva, hogy minél hatékonyabban minél szélesebb célközönséget érjünk el. A backend egy *adattárolási rétegből* és egy *REST API*-t kínáló webes csatlakozási pontból áll, utóbbit *.NET* platformon implementáltuk. Az alkalmazás a *Microsoft Azure* felhőben képes futni, garantálva a szoftver gyors skálázhatóságát és az üzemeltetés költséghatékonyságát. Az adattároláshoz a *Microsoft Azure* által nyújtott szolgáltatásokat használtuk.

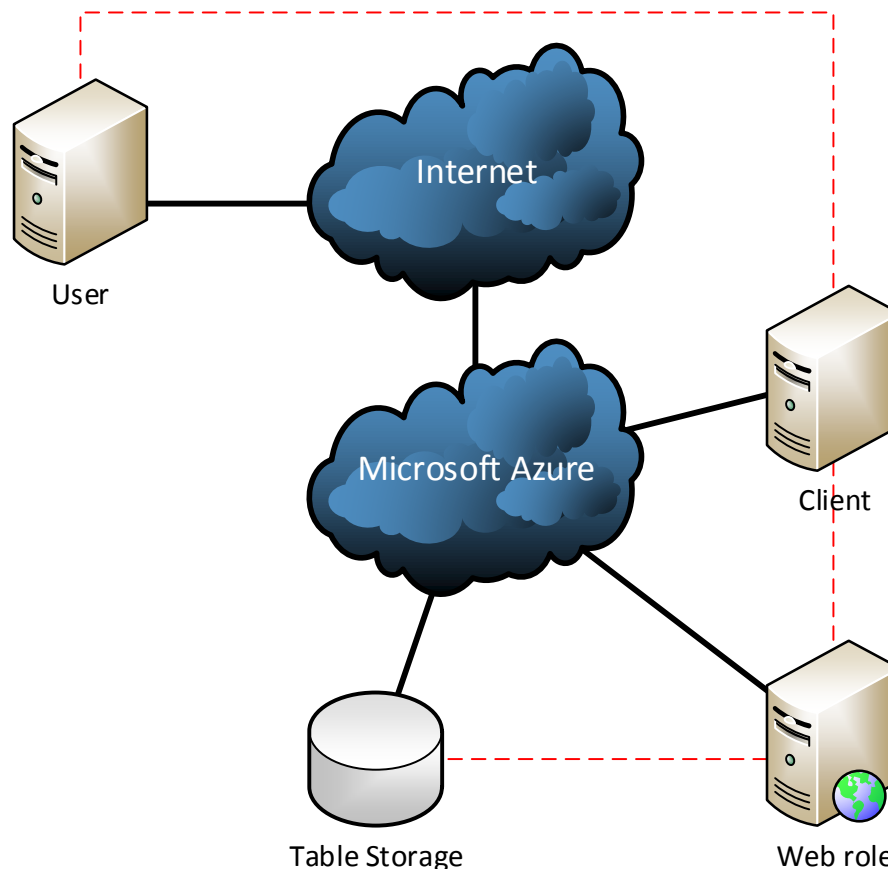
Megvalósítás

Az alkalmazást a feladatkiírásnak megfelelően egy többretegű alkalmazásként készítettük el. Az általunk elkészített programot „Nominate and Vote” névre kereszteltük, utalva a program funkcionalitására.

A fejezetben áttekintést adunk a program architektúrájáról, bemutatjuk az egyes komponensek feladatait és felelősségeit, továbbá részletesen ismertetjük a használt adatmodellt és a grafikus felhasználói felület felépítését.

Architektúra

A rendszer architektúrája az 1. ábrán látható. A felhasználó interneten keresztül tudja elérni az alkalmazást. Az alkalmazás a Microsoft Azure felhőben fut. Az adatok tárolására a Table Storage szolgáltatását használjuk a felhőnek, a webes API megvalósítására pedig az Azure Web role-ját. Az előbbieken kívül még használunk egy klienst, ami a kliens oldali kódot futtatja. A rendszer előnye, hogy a felhő használatának köszönhetően könnyen és jól skálázható. A rendszer működése során a kommunikáció az ábrán látható piros vonalak mentén történik. A felhasználó a klientsel kommunikál, ami a Web role web API-ját hívogatja, ami eléri a Table Storage-ban található adatokat.



1. ábra A szoftver komponensek kapcsolata

A Nominate and Vote architektúrája 5 különálló modulra bontható. Ebből négy a klasszikus N-rétegű alkalmazás architektúrájának felel meg:

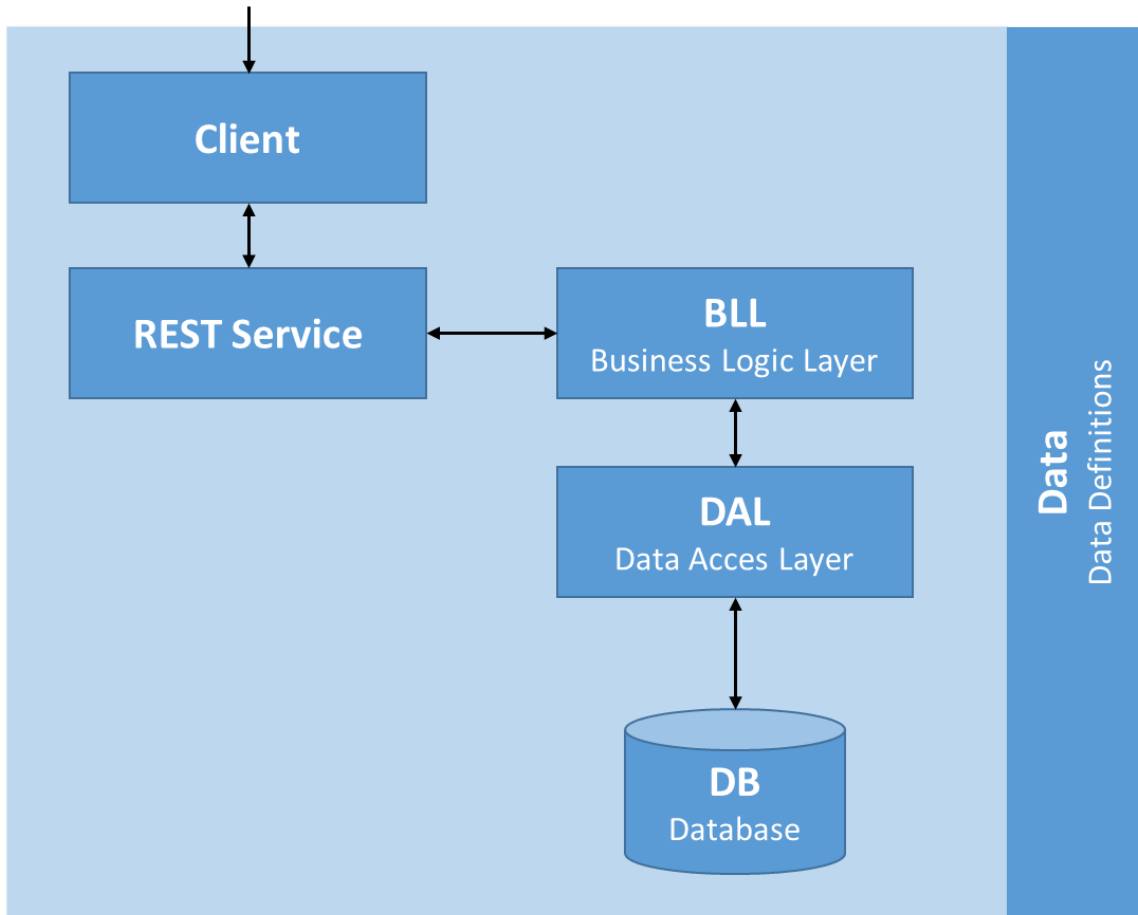
- adatbázisréteg (Database Layer, DB)
- adatelérési réteg (Data Access Layer, DAL)

- üzleti logikai réteg (Business Logic Layer, BLL)
- kliens (Client)

Ezeket kiegészítettük még egy különálló komponenssel:

- adatdefiníciók (Data Definitions).

Az egyes komponensek kapcsolatait mutatja be a 2. ábra. A fejezetben a fenn említett komponenseket mutatjuk be részletesen.



2. ábra A szoftver architektúrája

Adatbázis réteg

Az adatbázis réteg felel az adatok perzisztálásáért. Erre a célra mi a Microsoft Azure Table Storage rendszerét választottuk, a következők miatt:

- Az alkalmazás felhőben való futtatása lehetővé teszi, hogy a rendszer skálázható legyen.
- Elterjedt a gyakorlatban is gyakran alkalmazott megoldás.

Az adatbázis táblákban szereplő adatok reprezentálásához a programban különböző entitásokat hoztunk létre. Az entitások egy DataManager komponensnek köszönhetően leképződnek a programban használt objektumokká.

Adatdefiníciók

Annak érdekében, hogy a program minden komponensében elérhetők legyenek az egyes adatdefiníciók, ezt egy külön komponensként készítettük el.

Adathozzáférési réteg

Célja az adathozzáférés biztosítása a felsőbb rétegek számára. Ennek megfelelően a réteg feladata:

- üres adatbázis létrehozása,
- új entitások létrehozása az adatbázisban,
- igény szerinti adathozzáférés biztosítása felsőbb rétegnek az adatbázishoz.

Üzleti logikai réteg

Célja az adatelérési rétegtől kapott adatok alapján kommunikálni a REST Service-szel. A mi esetünkben az üzleti logikai réteg kevés feladatot lát el, a programban található POCO objektumokon hajt végre néhány műveletet.

REST Service

A REST Service az üzleti logikai réteg és a kliens közötti hívásokat valósítja meg. A következőkben az általunk készített REST API-t mutatjuk be.

A pollok kezeléséhez a következő API hívások megalkotására volt szükség:

Poll

API

GET api/Poll/GetPoll?pollId={pollId}

GET api/Poll/ListNominationPolls

GET api/Poll/ListVotingPolls

GET api/Poll/ListClosedPolls

Az admin funkciókat külön controllerben valósítottuk meg.

PollAdmin

API

POST api/PollAdmin/Save

Az adminisztrátor képes a felhasználókat tiltani és újra engedélyezni, ami a következő REST hívásokkal valósítható meg:

Admin

API

POST [api/Admin/BanUser?userId={userId}](#)

POST [api/Admin/UnBanUser?userId={userId}](#)

A jelölések kezeléséhez szükséges API hívások:

Nomination

API

GET [api/Nomination/GetForUser?userId={userId}](#)

POST [api/Nomination/Save](#)

DELETE [api/Nomination/Delete?nominationId={nominationId}](#)

A felhasználó be és ki tud jelentkezni a rendszerbe és a rendszerből.

Auth

API

POST [api/Auth/Login](#)

POST [api/Auth/Logout](#)

Az alábbi két controller a hírek manipulálásához, lekéréséhez szükséges API-t valósítja meg.

NewsAdmin

API

POST [api/NewsAdmin/Save](#)

DELETE [api/NewsAdmin/Delete?newsId={newsId}](#)

News

API

GET [api/News/ListNews](#)

Tesztelés

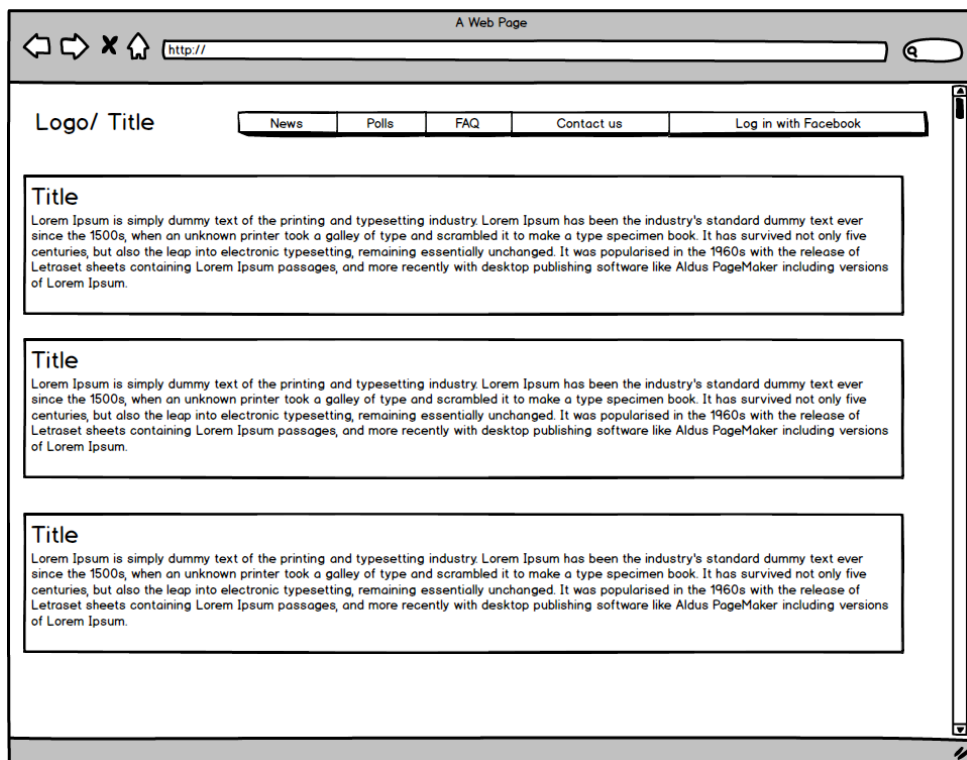
A grafikus felület megvalósítása előtt a rendszert egységtesztekkel és integrációs tesztekkel ellenőriztük. Nagy hangsúlyt fektettünk a POJO objektumok egységtesztelésére és a REST API integrációs tesztelésére. A tesztek elkészítését azért tartottuk fontosnak, hogy a grafikus felület elkészítése előtt megbizonyosodjunk arról, hogy az elkészült funkciók az elvárásnak megfelelően működjenek.

Grafikus felület (Kliens)

A szoftver tartalmaz egy felhasználói és egy adminisztrációs felületet. A felhasználói felület célja, hogy egy egyszerű, könnyen átlátható felületet nyújtsunk nekik, amely segítségével az összes funkció elérhetővé válik. Az adminisztrációs felület célja, hogy egyszerű, könnyen kezelhető legyen és segítségével az adminisztrátori teendők gyorsan és könnyedén elvégezhetőek legyenek. A tervezés során nagy hangsúlyt fektettünk a grafikus felületek megtervezésére, a felületek elkészítését mockup-ok tervezése és készítése előzte meg. A grafikus felületeket HTML5 alapokon készítettük el, azáltal lehetővé tettük, hogy a szolgáltatás akár mobiltelefonról is elérhető legyen. A felhasználói felület terveket az elkészült mockup-okkal mutatjuk be a következőkben. (A felhasználói felület a házi feladat leadási határidejére nem készült el, azonban a felület a tervek alapján fog elkészülni.)

Felhasználói grafikus felület

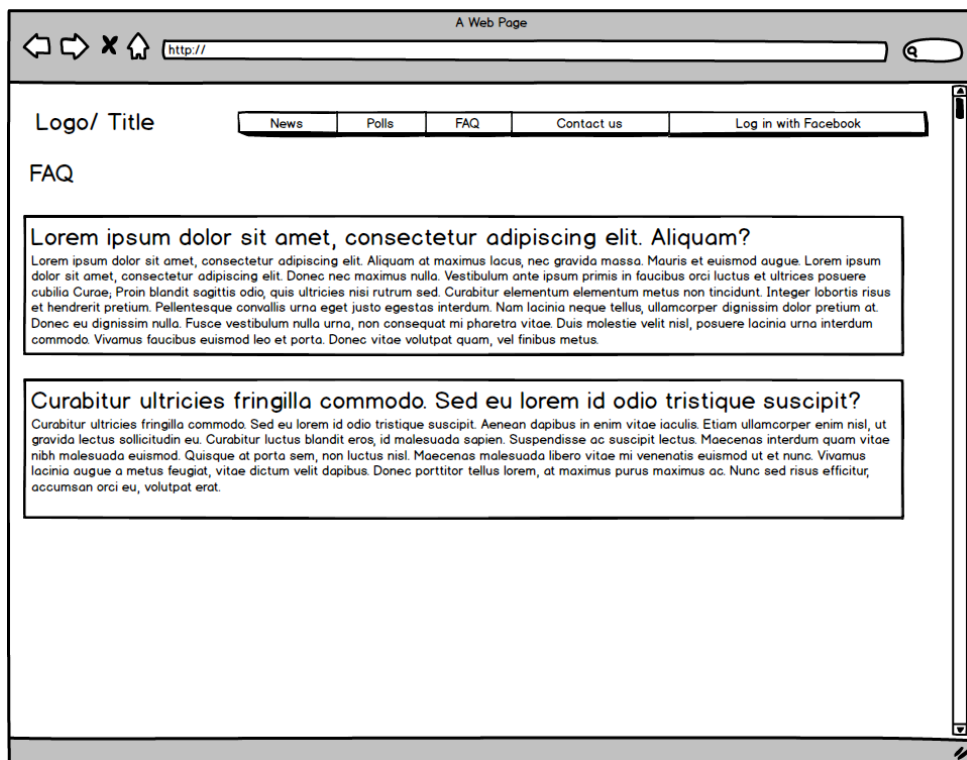
A program indulása után a felhasználót a böngészőben a kezdő képernyő várja, ami tartalmazza az eddig kiküldött híreket (2. ábra).



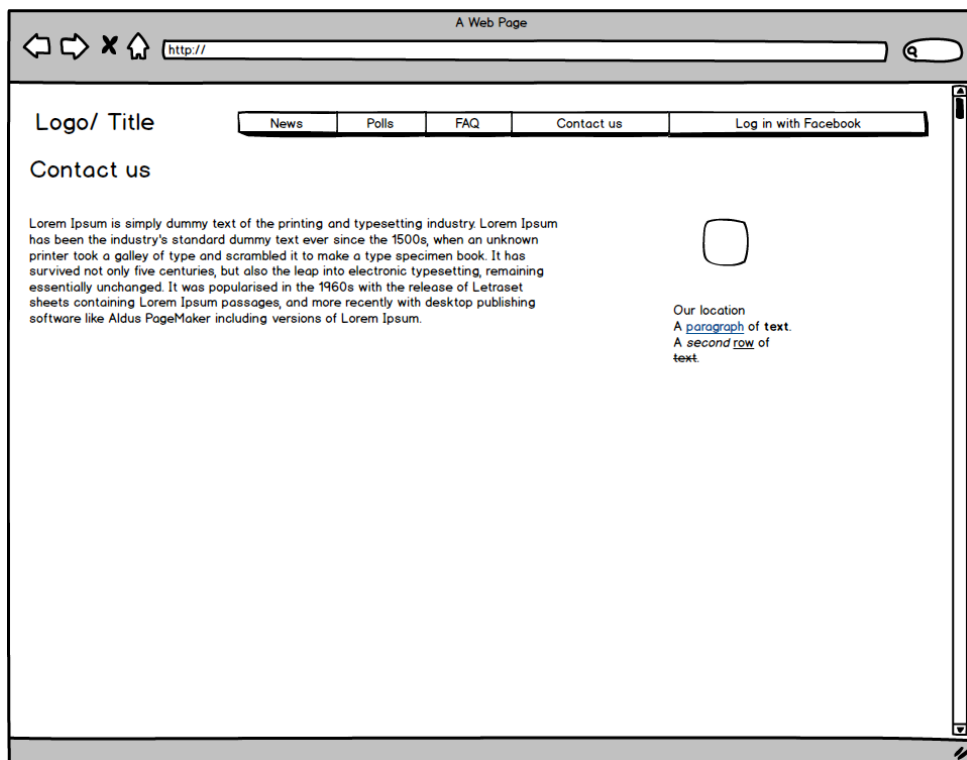
3. ábra Kezdőképernyő

A látogató számára a program biztosítja, hogy megtekinthesse a híreket, a lezárt szavazásokat, a statikus oldalainkat (gyakran ismételt kérdések, kapcsolat) és lehetőséget nyújt Facebook fiókkal történő belépésre.

A statikus oldalak egyszerűek, a legfontosabb információkat tartalmazzák szöveges formátumban. A gyakran ismételt kérdések a hírekhez hasonló formátumban jelennek meg. A kapcsolat oldal pedig tartalmazza az alkalmazással kapcsolatos legfontosabb információkat, a fejlesztők legfontosabb adatait, elérhetőségeit.

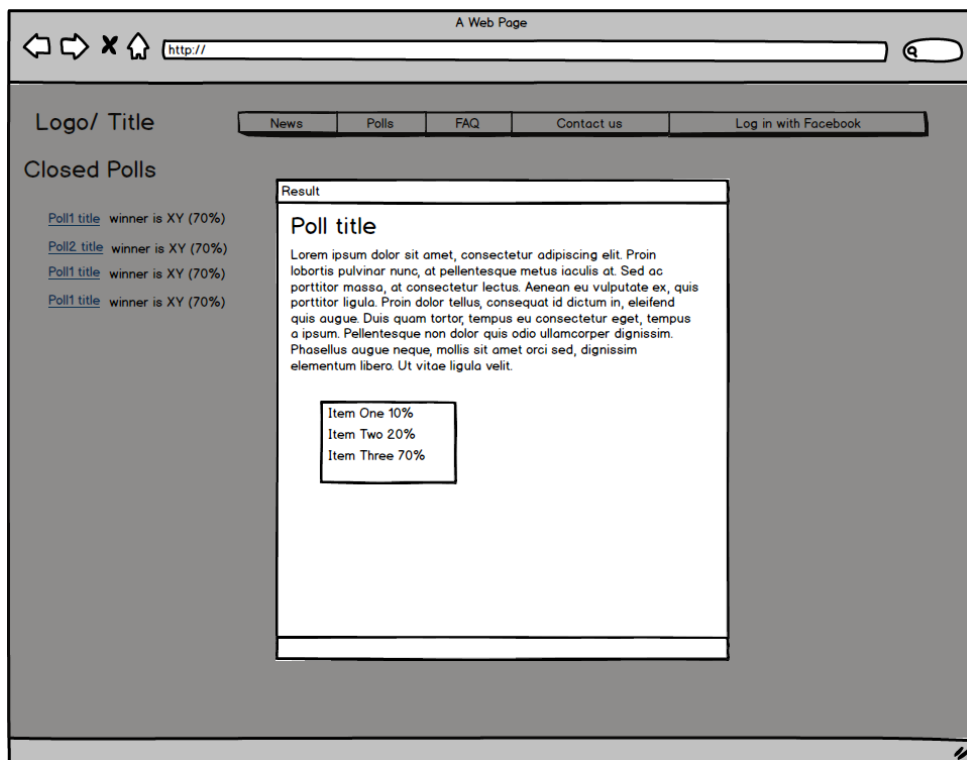


4. ábra Gyakran ismételt kérdések oldal

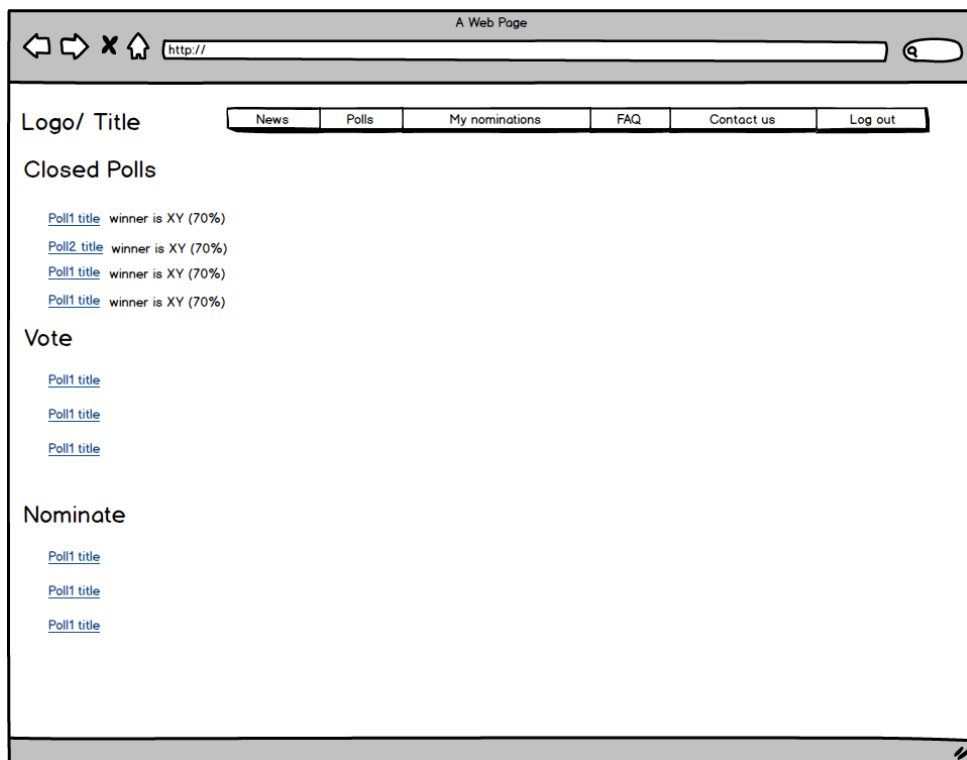


5. ábra Kapcsolat oldal

A látogatók meg tudják tekinteni a lezárt jelölés-szavazások eredményét. A jelölés-szavazások közül a szimpatikusot egy felsorolásból tudja kiválasztani, majd a megfelelőre rákattintva egy felugró ablakban tekintheti meg a szavazás végeredményét és a szavazatok százalékos megoszlását.



6. ábra Jelölés-szavazás eredménye



7. ábra Jelölés-szavazások listázása

A Facebook fiókkal történő belépés után a felhasználónak lehetősége van böngészni az éppen folyamatban lévő jelölés-szavazásokat, filmeket, sorozatokat jelölni, illetve egy-egy jelölés-szavazáson tud szavazni, ha ezt korábban nem tette meg. A jelölés-szavazások listázására létrehoztunk egy külön oldalt, ahol a jelölés-szavazások a státuszuk szerint csoportosítva jelennek meg.

A felhasználó képes filmet jelölni a „Nominate” státuszú jelölés-szavazásokra. Ebben az esetben a szimpatikus jelölés-szavazásra kell kattintani, majd a felugró popup ablakban egy-egy szövegdobozban meg kell adni a jelöltet és egy rövid indoklást. A jelölt megadását segíti, hogy a szövegdoboz auto-complete funkcióval van felruházva. Miután kitöltöttük mindkét mezőt a Send gombbal el tudjuk küldeni a jelölésünket.

The screenshot shows a web browser window titled "A Web Page" with a search bar containing "http://". The page has a navigation bar with links: News, Polls, My nominations, FAQ, Contact us, and Log out. The main content area is divided into three sections: "Closed Polls", "Vote", and "Nominate".

Closed Polls

- [Poll1 title](#) winner is XY (70%)
- [Poll2 title](#) winner is XY (70%)
- [Poll1 title](#) winner is XY (70%)
- [Poll1 title](#) winner is XY (70%)

Vote

- [Poll1 title](#)
- [Poll1 title](#)
- [Poll1 title](#)

Nominate

- [Poll1 title](#)
- [Poll1 title](#)
- [Poll1 title](#)

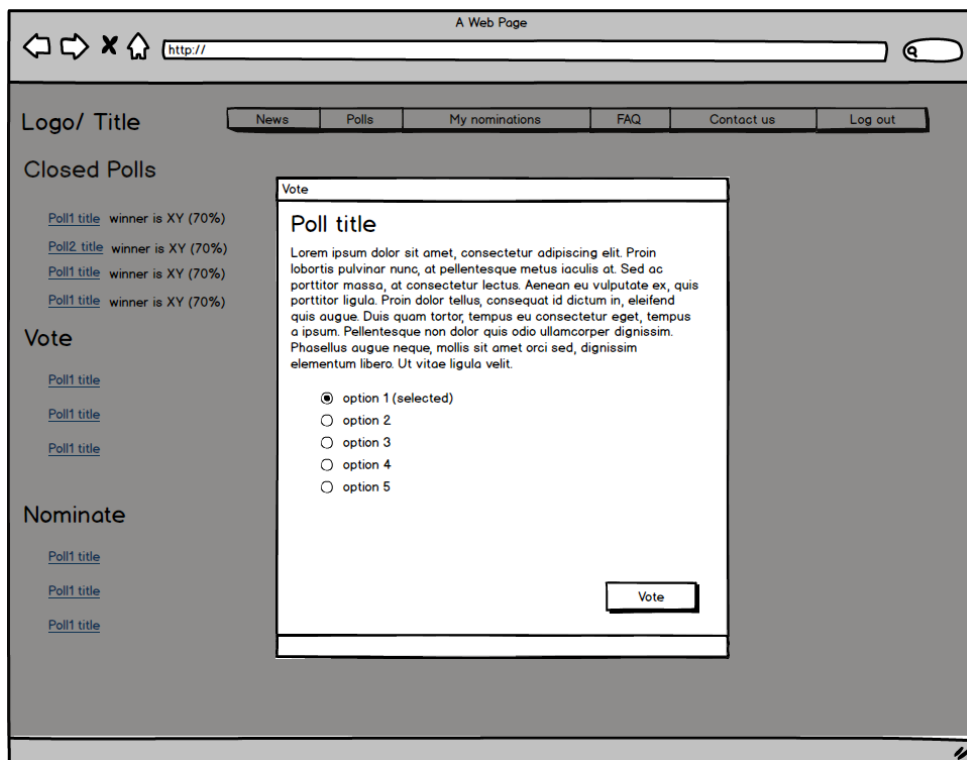
Nominate Form

The "Nominate" form is a modal window with the following fields:

- Poll title**: A text area containing placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin lobortis pulvinar nunc, at pellentesque metus iaculis at. Sed ac porttitor massa, at consectetur lectus. Aenean eu vulputate ex, quis porttitor ligula. Proin dolor tellus, consequat id dictum in, eleifend quis augue. Duis quam tortor, tempus eu consectetur eget, tempus a ipsum. Pellentesque non dolor quis odio ullamcorper dignissim. Phasellus augue neque, mollis sit amet orci sed, dignissim elementum libero. Ut vitae ligula velit."
- Poll subject**: A text input field.
- Motivation**: A text input field.
- Send**: A button to submit the nomination.

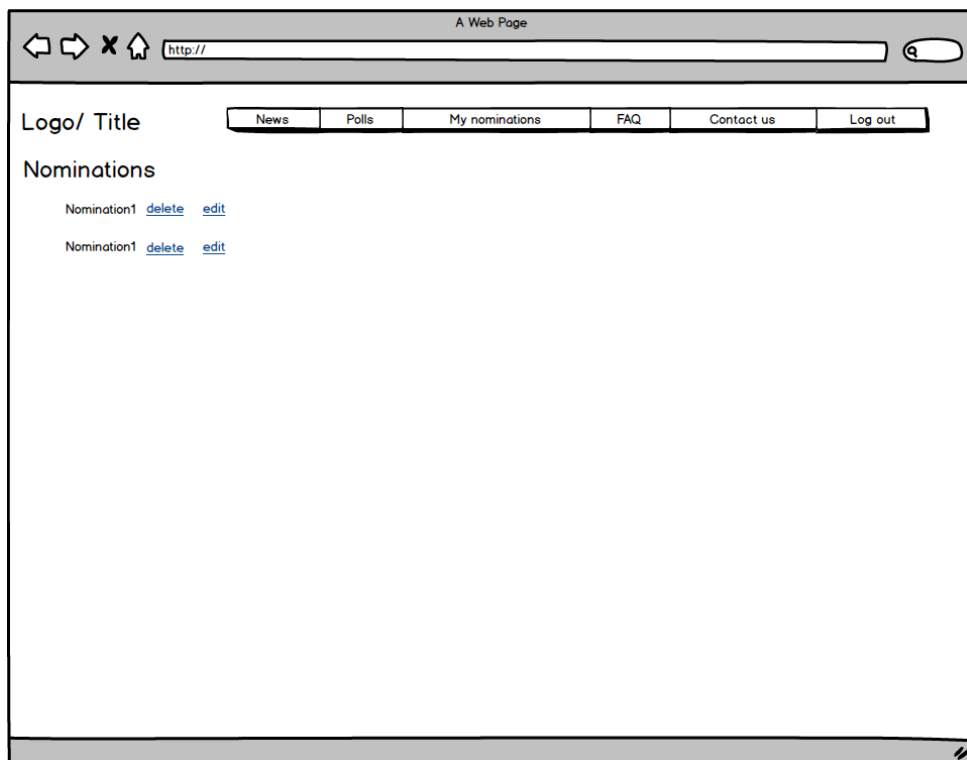
8. ábra Jelölés leadása

A felhasználónak lehetősége van szavazat leadására abban az esetben, ha korábban még az adott jelölés-szavazásra nem adott le voksot. Szavazni egy felugró ablakban tud, miután rákattintott a szimpatikus jelölés-szavazás nevére. A lehetőségek közül rádiós gombokkal tud választani, majd a Vote gombbal tudja véglegesíteni a szavazást.



9. ábra Szavazat leadása

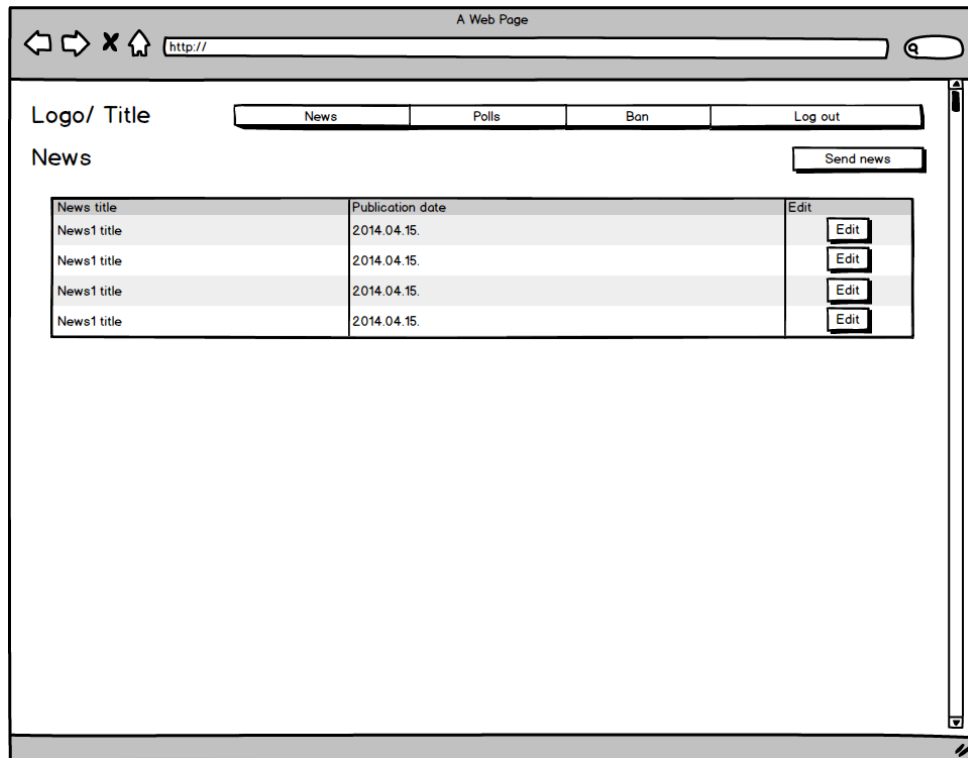
A felhasználónak lehetősége van a korábban leadott jelöléseit módosítani, törölni, amit a My nominations menüpont alatt a megfelelő jelölés kiválasztásával, szerkesztésével vagy törlésével tud elérni.



10. ábra Jelölések módosítása

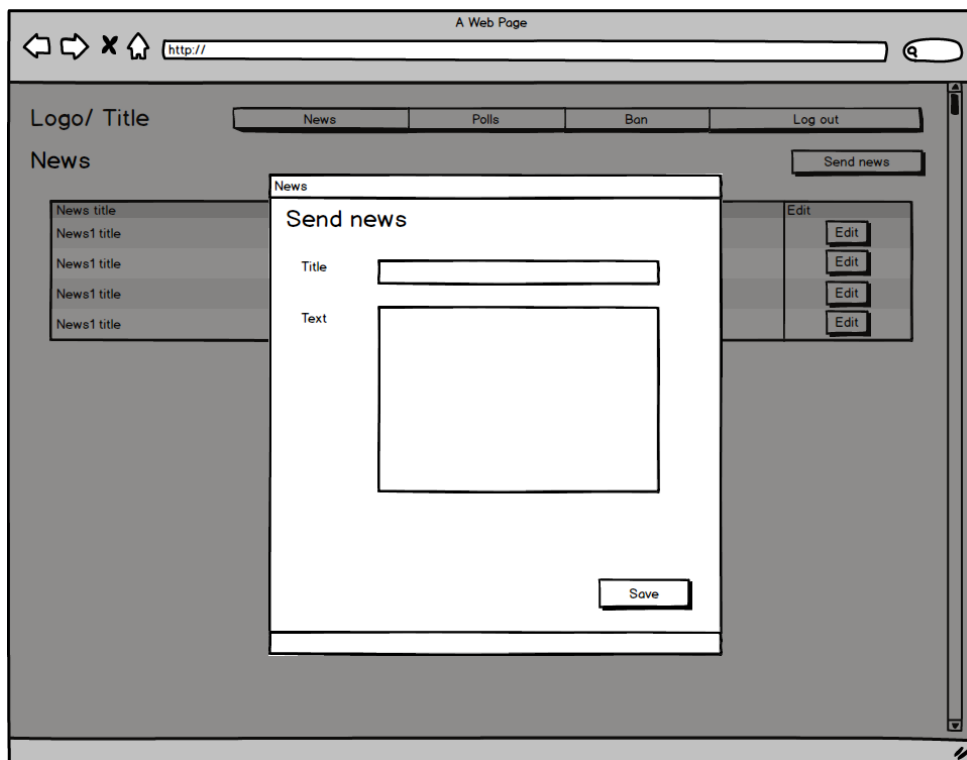
Adminisztrátori grafikus felület

Az adminisztrátorok is Facebook fiók segítségével léphetnek be a rendszerbe, a belépő felhasználók szerepkörét az autentikációs folyamat során határozzuk meg. A belépés után az adminisztrátor külön oldalon tudja a jelölés-szavazásokhoz, a hírekhez és a felhasználókhoz kapcsolódó feladatokat elvégezni. A híreket egymás alatt felsorolva tudja megtekinteni, amiket az edit gomb segítségével tud szerkeszteni, illetve a Send news gomb segítségével tud új hírt létrehozni.



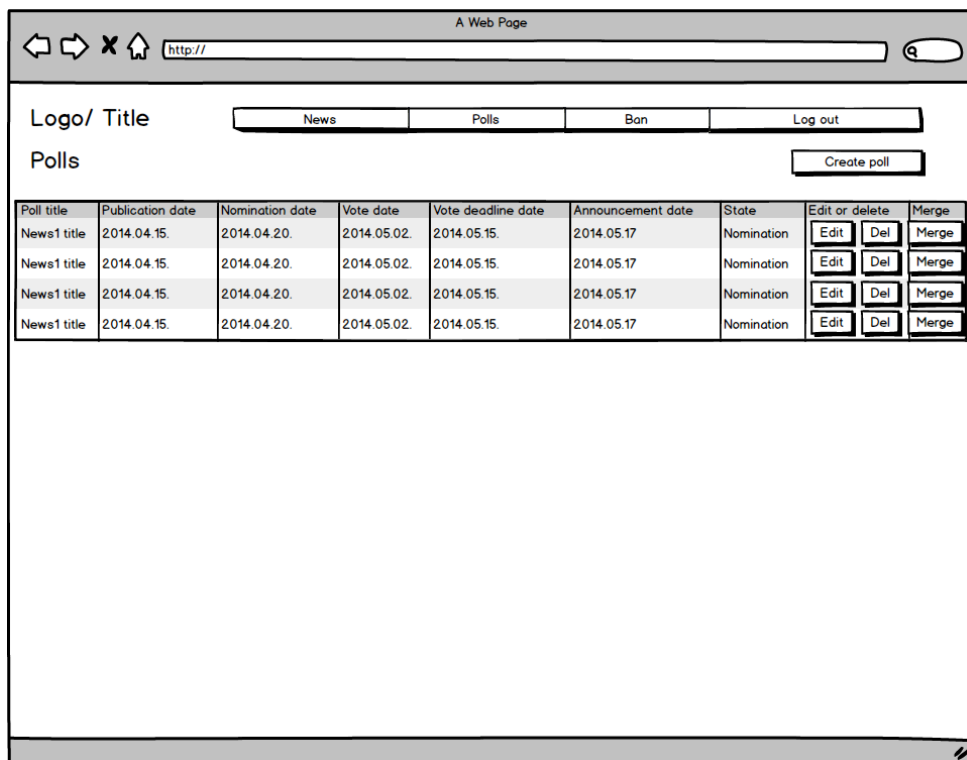
11. ábra Hírek karbantartása

Az új hír létrehozása és a hír szerkesztése teljesen hasonlóan egy-egy felugró ablakban történik. A hírnek meg lehet adni vagy módosítani lehet a címét és a szövegét.



12. ábra Új hír kiküldése

A jelölés-szavazásokat egy táblázatban tekintheti meg az adminisztrátor. Az adminisztrátor képes új jelölés szavazást kiírni (Create Poll) és a meglévőket tudja szerkeszteni vagy törölni. Ezen kívül lehetőség van még a leadott jelölések összefésülésére.



13. ábra Jelölés-szavazások admin oldal

Új jelölés-szavazást egy felugró ablak segítségével lehet létrehozni, ahol meg kell adni a címét és a hozzá tartozó leírást, illetve a különböző határidőket. A jelölés-szavazás szerkesztése egy hasonló felugró ablak segítségével történik.

The screenshot shows a web application interface for managing polls. A modal window titled 'Create poll' is open, allowing the user to add a new poll. The background page includes a navigation menu and a table of existing polls.

Poll title	Publication date	Nomina
News1 title	2014.04.15.	2014.04.
News1 title	2014.04.15.	2014.04.
News1 title	2014.04.15.	2014.04.
News1 title	2014.04.15.	2014.04.

Create poll

Title:

Text:

Publication date:

Nomination date:

Vote date:

Vote deadline:

Announce-ment date:

State:

Save

14. ábra Új jelölés-szavazás létrehozása

Az adminisztrátor feladata, hogy a jelölés-szavazásra leadott jelöléseket összefésülje, a nem releváns jelöléseket törölje, illetve ha ugyanazt a filmet többen is jelölték, akkor az indoklásokat egyesítse. Ezt a feladatot egy popup ablak segítségével tudja megvalósítani.

Logo/ Title

News Polls Ban Log out

Create poll

Poll title	Publication date	Nomina
News1 title	2014.04.15.	2014.04.
News1 title	2014.04.15.	2014.04.
News1 title	2014.04.15.	2014.04.
News1 title	2014.04.15.	2014.04.

Merge

Merge poll

Nomination1 title ☒ Include

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent ac purus vel est hendrerit ullamcorper. Cras elementum maximus arcu, id sodales dolor vestibulum vel.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent ac purus vel est hendrerit ullamcorper.

Nomination2 title ☐ Include

Save

State Edit or delete Merge

Nomination	Edit	Del	Merge
Nomination	Edit	Del	Merge
Nomination	Edit	Del	Merge
Nomination	Edit	Del	Merge

15. ábra Jelölések összefésülése

Logo/ Title

News Polls Ban Log out

Ban

Name: _____ Ban

ID	Name	Delete
1	Kiss Bela	Delete
2	Kiss Anna	Delete
3	Nagy Bela	Delete

16. ábra Felhasználók letiltása

A szoftver lehetőséget biztosít, hogy a gyanús felhasználókat vagy azokat, akik valamilyen visszaélést követtek el kitiltjuk. A tiltást a felületen úgy tudja az adminisztrátor megtenni, hogy egy auto complete támogatással rendelkező szövegdobozba begépeli a felhasználó nevét, majd a Ban gombra kattint. A felületen listaként jelennek meg a tiltott felhasználók, akik tiltását a törlésre való kattintással tudjuk feloldani.

Adat- és adatbázisterv

A program objektum modellje

A következőkben az adatmodellt mutatjuk be részletesen, aminek az osztálydiagramja a 17. ábrán látható. Az objektum modell tartalmazza az Administrator, User, Poll, Nomination, Vote, PollSubject, News osztályokat és a PollState enumot.

Az objektumok (POCO) leírása

A következőkben az ábrán látható osztályokat és azok tulajdonságait mutatjuk be részletesen.

Administrator: Az adminisztrátor felhasználót reprezentálja, az adminisztrátor egy szerepkör, amihez a megfelelő felhasználót az azonosítója segítségével kötjük hozzá.

User: Felhasználó, aki rendelkezik névvel, illetve be van nála jegyezve, hogy ki van-e tiltva a rendszerből.

Poll: Szavazás-jelölés, ami rendelkezik címmel és szöveggel, van egy PollState típusú állapota, illetve tartoznak hozzá különböző határidők.

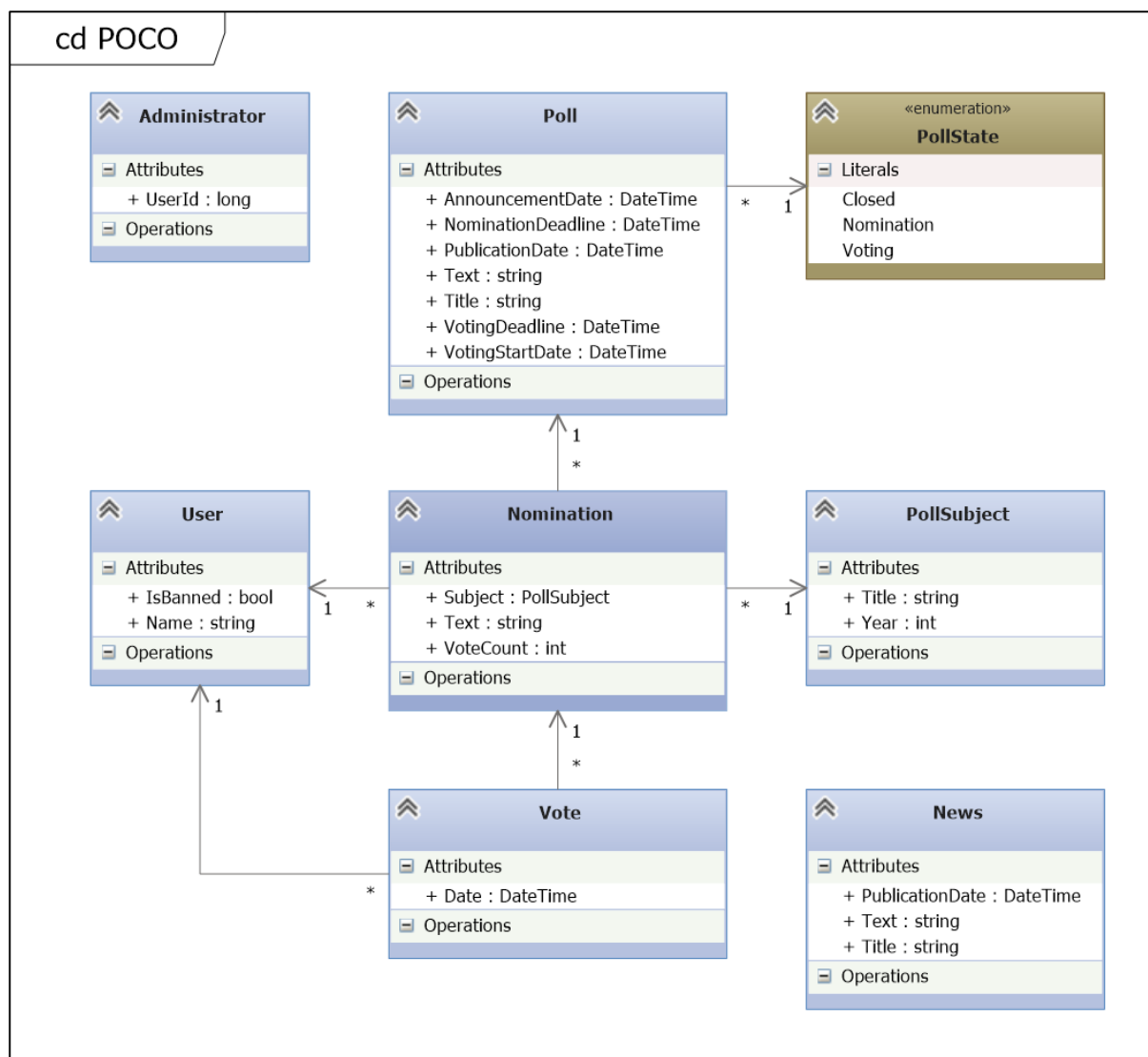
Nomination: Jelölés, ami tartalmazza a jelölt filmet vagy sorozatot (PollSubject), illetve egy a jelöléshez tartozó indoklást. Ezen kívül tároljuk, hogy melyik jelölés-szavazáshoz tartozik, illetve, hogy melyik felhasználó adta le a jelölést.

Vote: Szavazat, amit egy adott felhasználó ad le egy megadott jelölésre. Nyilvántartása azért fontos, mivel a szavazatok megvizsgálásával dönthető el, hogy egy felhasználó szavazott-e már az adott jelölés-szavazáson. A jelölés rendelkezik még egy dátummal, ami a leadás dátumát reprezentálja.

PollSubject: A jelölés-szavazásokra PollSubject-ek jelölhetők, amik a különböző filmeket és sorozatokat reprezentálják. Tartalmazza a film címét és a megjelenésének az évét.

News: A honlapra kikerülő hírek, amely tartalmazza a hír címét és a szövegét. A hírek nyilván tartjuk a keletkezésének idejét.

PollState: A jelölés-szavazás állapotait tartalmazza.



17. ábra POCO objektumok

Az adatbázis

Az adatbázis tárolására a Microsoft Azure Table Storage megoldását választottuk. A módszer előnye, hogy jól skálázható, ezáltal alkalmassá tettük a rendszert arra, hogy akár valós környezetben is jól használható legyen. A Table Storage-ban egy-egy entitást egy PartitionKey és egy RowKey határoz meg, amelyek egyedi azonosítóként funkcionálnak. A PartitionKey szerepe, az, hogy az azonos partíciókban található adatokat ugyanazon a csomópontok kell tárolni, azonban a különböző kulccsal rendelkezők szétszathatók a különböző csomópontok között. Ezen kívül még nyilvántart egy időbélyeget, ami az utolsó módosítás dátumát jelenti és a rendszer tart karban. Az adatbázisban az entitások a soroknak felelnek meg. Az entitások tulajdonságai típusos kulcs-érték párok, amik nevei megegyezik a táblában az oszlopok neveivel.

A következőkben a megtervezett adatbázis táblákat mutatjuk be, amelyek megtervezésének alapjául a program objektumai szolgáltak. A tervezés során figyelembe vettük, hogy a Table Storage-ból az adatok a PartitionKey alapján lekérdezhetőek.

Administrator

Az adminisztrátorok PartitionKey-e az adminisztrátor user ID-je.

Név	Típus	Megjegyzés
-----	-------	------------

PartitionKey	string	User ID
---------------------	--------	---------

User

A User tartalmazza a Name és az IsBanned tulajdonságokat, amiket már korábban definiáltuk. A tábla PartitionKey-e a felhasználókat azonosító ID.

Név	Típus	Megjegyzés
PartitionKey	string	ID
Name	string	
IsBanned	bool	

Poll

A Poll tartalmazza a korábban meghatározott határidőket, a jelölés-szavazás címét és leírását. A tábla PartitionKey-e a jelölés-szavazás állapota, a RowKey pedig a jelölés-szavazás azonosítója. A PartitionKey-nek azért a jelölés-szavazás állapotát választottuk, mivel a jelölés-szavazásokat a programunk jelölés-szavazás állapot alapján kérdezi le.

Név	Típus	Megjegyzés
PartitionKey	string	State
RowKey	string	ID
Text	string	
PublicationDate	DateTime	
NominationDeadLine	DateTime	
VotingStartDate	DateTime	
VotingDeadline	DateTime	
AnnouncementDate	DateTime	

Nomination

A Nomination tartalmazza a korábban meghatározott leírást, a szavazatok számát, illetve a jelölést leadó felhasználó ID-ját, a jelölést tartalmazó jelölés-szavazás azonosítóját és a hozzá tartozó jelölt azonosítóját. A táblában a PartitionKey a jelölés-szavazás azonosítója, ami azért praktikus választás, mert az adatbázisból általában csak az egy adott jelölés-szavazáshoz kapcsolódó jelöléseket kérdezzük le. A RowKey a jelölés ID-ja.

Név	Típus	Megjegyzés
PartitionKey	string	Poll ID
RowKey	string	ID
UserId	long	
SubjectId	string	
Text	string	
VoteCount	int	

Vote

A Vote tartalmazza a szavazatot leadó felhasználó azonosítóját, illetve annak a jelölésnek az azonosítóját, amelyre a szavazatot leadták. A szavazat tartalmazza még a leadásának dátumát. A táblában a PartitionKey a jelölés azonosítója, hiszen általában egy adott jelöléshez tartozó szavazatokat szeretnénk lekérdezni. A RowKey a szavazatot leadó felhasználó azonosítója.

Név	Típus	Megjegyzés
PartitionKey	string	Nomination ID
RowKey	string	User ID
Date	DateTime	

PollSubject

A PollSubject tartalmazza az addott film vagy sorozat nevét és megjelenésének évét. A táblában a PartitionKey a PollSubject azonosítója.

Név	Típus	Megjegyzés
PartitionKey	string	ID
Title	string	
Year	int	

News

A News tartalmazza a hír címét, szövegét és a hír publikálásának dátumát. A tábla PartitionKey-e a hír azonosítója.

Név	Típus	Megjegyzés
PartitionKey	string	ID
Title	string	
Text	string	
PublicationDate	DateTime	

Az objektummodell és az adatbázis közötti konverziót egy általunk készített DataManager osztály végzi.

GUI terv

A program tervezésekor az egyszerű felépítést tartottuk szem előtt. Ennek megfelelően a program érdemi működését egy egyszerű HTML alapú oldal látja el. A felhasználó egy egyszerű menüsor segítségével tud navigálni a funkciók között, a különböző szerkesztéshez, létrehozáshoz, módosításhoz tartozó funkciók használatához egy-egy popup ablakban található formot kell kitölteni. A grafikus felület részletes leírása a „Grafikus felületek” fejezetben található.

A program készítése során felhasznált eszközök

- Microsoft Visual Studio 2013
 - Felhasználás: fejlesztőkörnyezet
- Microsoft Azure Table Storage
 - Felhasználás: adatbázis-kezelő rendszer
- Microsoft Visio
 - Felhasználás: ábrák előállítása
- Microsoft Word
 - Felhasználás: dokumentáció elkészítése
- GitHub
 - Felhasználás: verziókezelés, csapatmunka támogatása

Összefoglalás

Munkánk során megterveztük, implementáltuk illetve dokumentáltuk a Nominate and Vote nevű szavazó-jelölő rendszert. Az elkészített alkalmazás segítségével szavazások írhatók ki, amikre a felhasználók tudnak jelölni filmeket és sorozatokat. A jelölési szakasz lezárulása után a felhasználók szavazni tudnak az adott jelölés-szavazás jelöltjeire, majd a szavazási időszak lezárása után megnézhetik az eredményeket.

A megvalósítás a következő rétegeket használja: adatbázis réteg, üzleti logikai réteg, REST Service és felhasználói felület. Az alkalmazás az adatokat a Microsoft Azure Table Storage-ban tárolja. A grafikus megjelenítést egy HTML 5 alapú webes felülettel oldottuk meg. Az alkalmazás biztosítja, hogy a jelölés-szavazási procedúra végigvihető legyen a jelölés-szavazás kiírásától az eredmény kihirdetéséig. A szoftver biztosítja, hogy az adminisztrátori teendők egyszerűen és könnyedén elláthatóak legyenek. A program Facebook fiókos hitelesítést tesz lehetővé. A rendszer belső működése különböző egységtesztekkel és integrációs tesztekkel ellenőrzött.

Munkánk során részletes terveket készítettünk – részük jelen dokumentum tartalmát képezik – és jelentős mennyiségű implementációs munkát is végeztünk. Ennek eredményeképpen egy jól működő és megbízható alkalmazást készítettünk el, amely az elvárt alapvető igényeknek megfelel, feladatát képes ellátni.