

# Rendszerbemutató dokumentáció

SZAVAZÁS KEZELŐ

Programrendszerek fejlesztése  
gyakorlat

## Tartalomjegyzék

|     |                                       |   |
|-----|---------------------------------------|---|
| 1   | Elképzeléseink .....                  | 3 |
| 2   | Csapatunk tagjai .....                | 3 |
| 3   | Rendszerspecifikáció .....            | 3 |
| 3.1 | Funkcionális követelmények .....      | 3 |
| 3.2 | Nem funkcionális követelmények .....  | 3 |
| 3.3 | Teljesítménybeli elvárások .....      | 3 |
| 4   | A rendszer képernyőtervei.....        | 4 |
| 5   | A munka feltételei .....              | 7 |
| 6   | Entitás-domain modell.....            | 7 |
| 7   | Menedzselt bean-ek és scope-juk ..... | 8 |

## 1 Elképzelések

Kitűzött feladatunk egy szavazás kezelő rendszer implementálása, a későbbiekben, továbbfejlesztés után rendelkezésre bocsátás. Célja, hogy interneten keresztül a felhasználók csoportja könnyen döntéseket tudjon hozni.

## 2 Csatatunk tagjai

Csatatunk három főből áll: Rózsa Dániel (RODRAAT.SZE), Juhász Péter (JUPSAAT.SZE), és Tóth László (TOLPAAT.SZE) személyében. A feladatokat egyenlően osztottuk fel egymás között, kölcsönösen segítünk egymásnak az adott részek elkészítésében.

## 3 Rendszerspecifikáció

Az online szavazáskezelést lebonyolító rendszer célja, hogy interneten keresztül a felhasználók könnyedén tudjanak döntéseket meghozni. A használathoz nem szükséges regisztráció, viszont a felhasználói csoport ebben az esetben tárolhatja is az információit a felületen.

### 3.1 Funkcionális követelmények

- Felhasználói felület
- Kérdések felvétele
- Válaszok felvétele
- Adatmódosítás
- Kérdések-válaszok nyomonkövetése

### 3.2 Nem funkcionális követelmények

A rendszer a következő környezetben kerül megvalósításra:

- Windows 7 operációs rendszer
- Netbeans 8.0 fejlesztői környezet
- Java EE
- JBoss Server
- Hibernate
- Primefaces

### 3.3 Teljesítménybeli elvárások

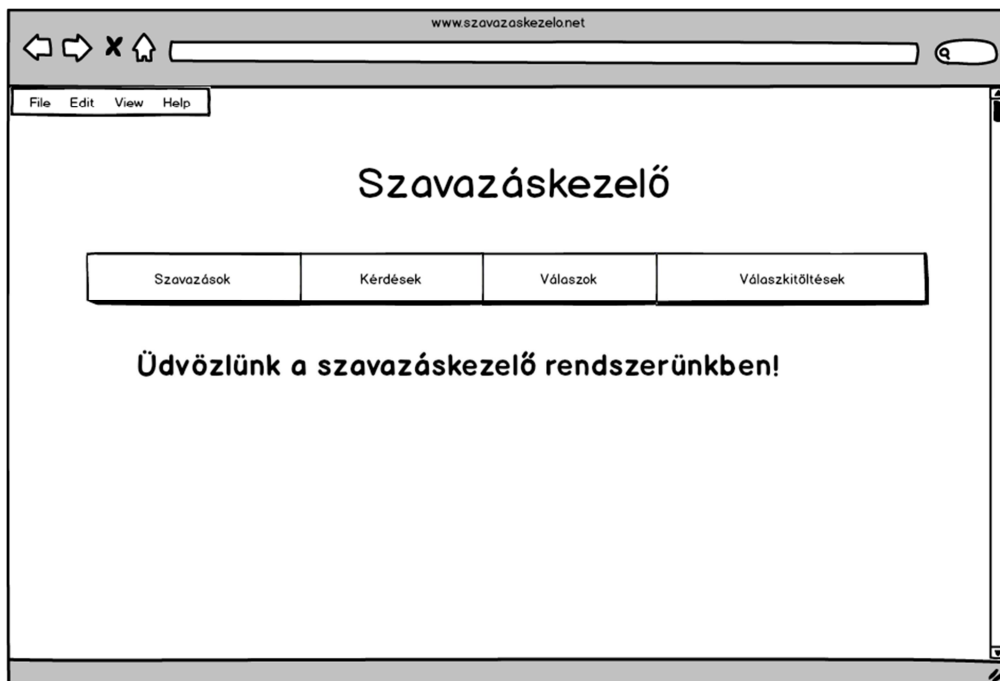
A tervezett rendszer interaktív mivolta miatt a következő elvárásoknak kell megfelelnie:

- Az adatelérési idő az 5 másodpercet nem haladhatja meg

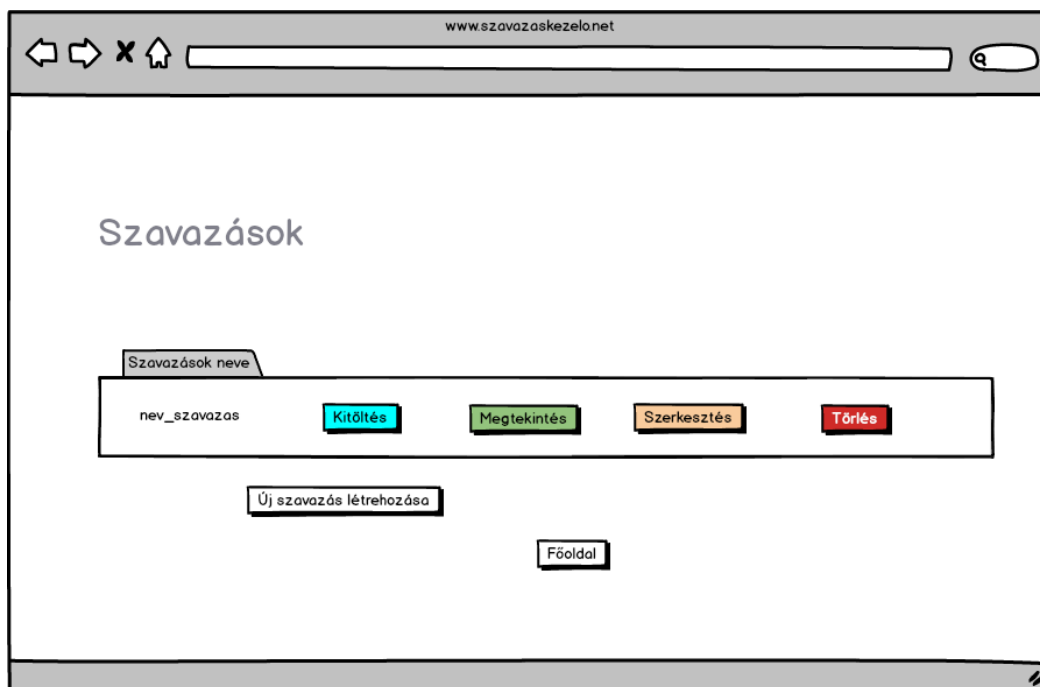
- Biztosítani kell, hogy a felhasználó legfeljebb 1 perc alatt meg tudja szemlélni a felhasználók, üzenetek és a csoportok listáját

## 4 A rendszer képernyőtervei

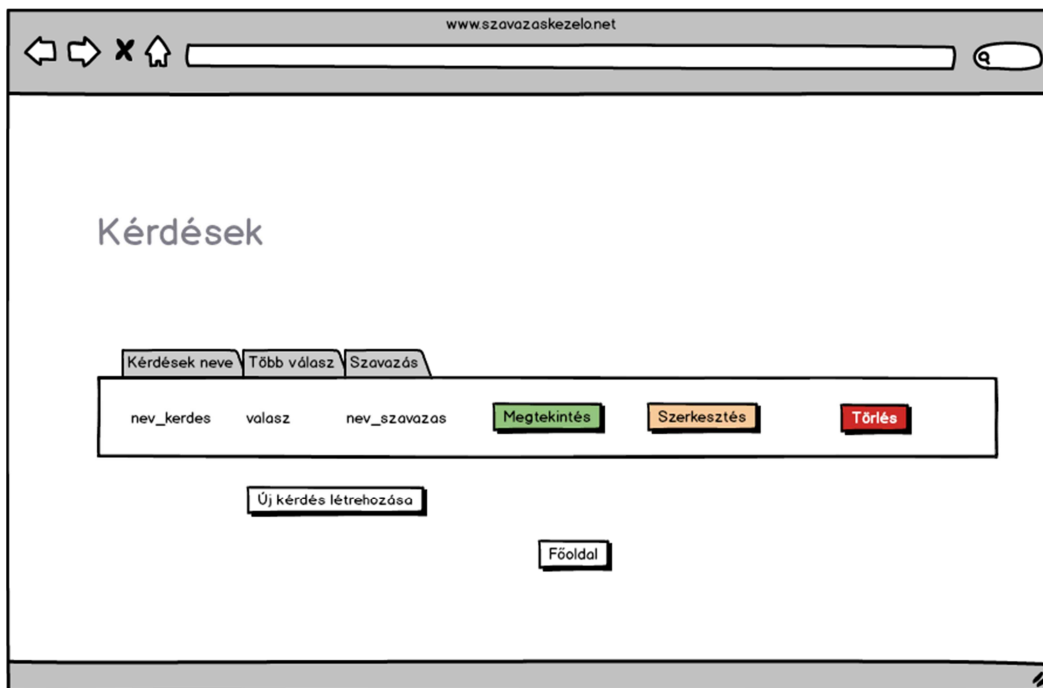
A bejelentkező oldal képernyőterve. Minden site-on a szellős, könnyen átlátható elrendezésre törekedtünk, amely akár más platformokon történő megjelenítés esetén is optimális elrendezésű.



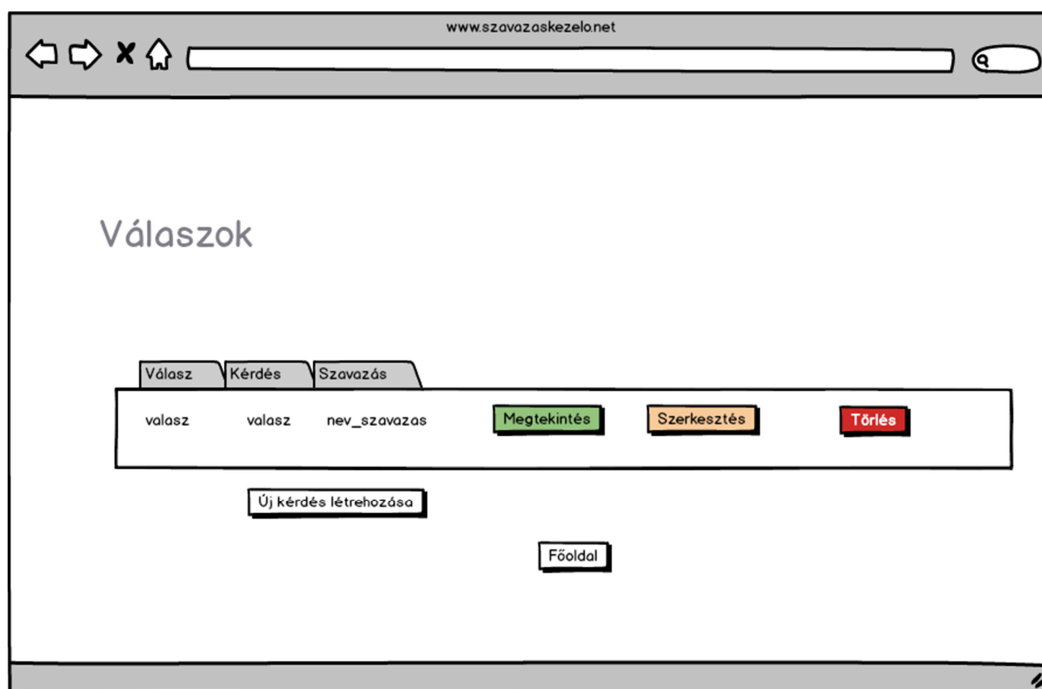
A szavazások menü felépítése.



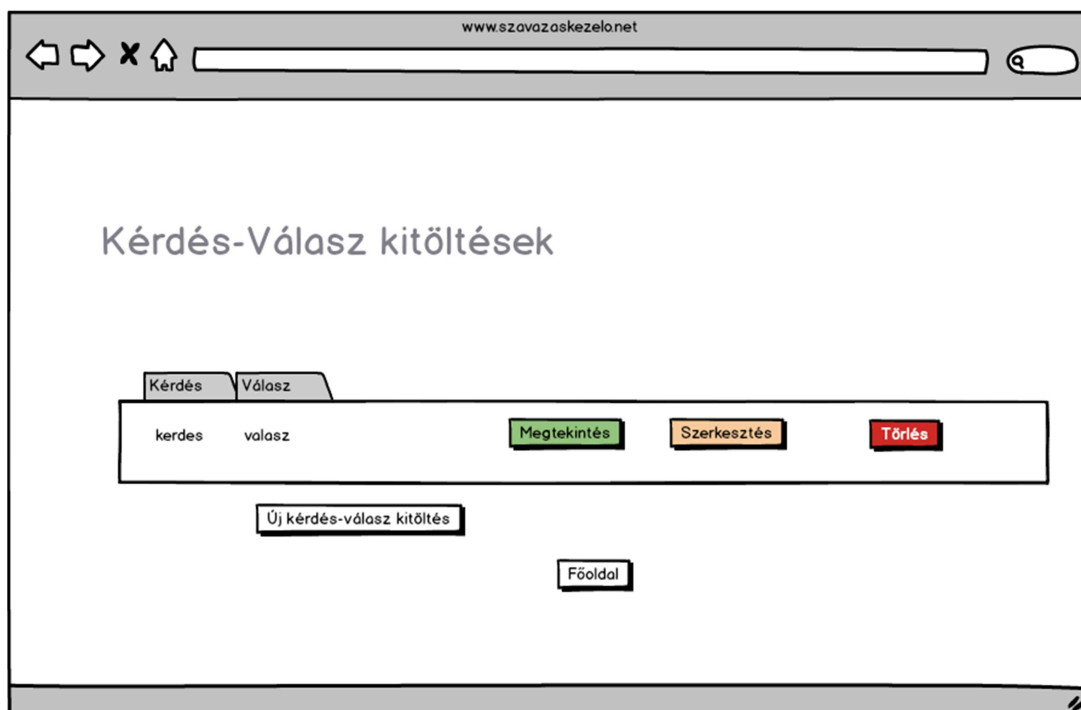
A kérdések menü tervezett kinézete:



A válaszok menü tervezett kinézete:



A Válaszkitöltések menü tervezett kinézete:



## 5 A munka feltételei

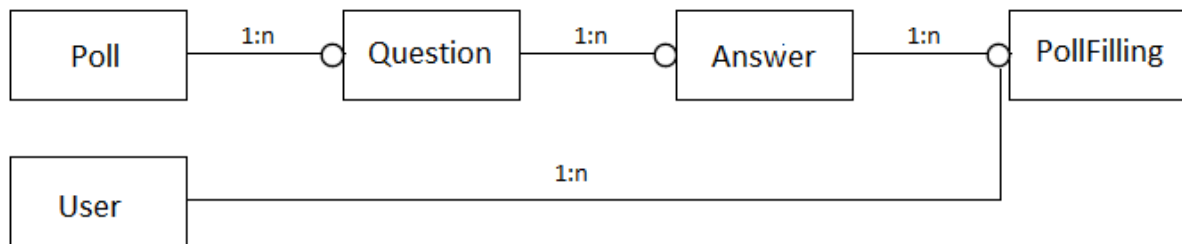
A projekthez a következő munkaállomásokat fogjuk használni a munka során:

- A projekt több munkaállomáson fog készülni.
- A munka során a Git verziókezelő rendszert használtuk
- Futás közben, a kész rendszer egyetlen munkaállomáson fog működni.

Szoftverkörnyezet:

A projekt megvalósítása közben Microsoft Windows rendszerekkel támogatott munkaállomásokon dolgozunk, Netbeans fejlesztői környezetet használva, Java programozási nyelven (Java EE). Szerverek a munkaállomásokon JBoss által, az adatbázis-kezelést pedig Hibernate segítségével oldjuk meg.

## 6 Entitás-domain modell



Ahogy az a modellünkből látszik, egy személyhez több kérdés, több válasz, és több, szavazás végeredményéből származó kérdés-válasz kapcsolat rendelhető. Egy szavazás mindig egy kérdéscsoport definiálása után jön létre, így előfordulhat, hogy egy kérdéshez, több válasz adódik, illetve egy válasz több kérdésre adható.

## 7 Menedzselt bean-ek és scope-juk

A *PollsController* és *PollsQuestionsController* osztályok életciklusa mindkét esetben *SessionScope*, amely esetén a scope az adott felhasználói munkamenet idejéig érvényes.

A *PollFillingsController* illetve a *PollAnswerController* osztályok életciklusa mindkét esetben *RequestScope*, amely esetén a scope az adott http kérés-válasz pár időtartamáig érvényes.

```
@ManagedBean(name = "pollsController")
@SessionScoped
public class PollsController implements Serializable {

    private Polls current;
    private DataModel items = null;
    @EJB
    private hu.prf.szavazaskezo.beans.PollsFacade ejbFacade;
    private PaginationHelper pagination;
    private int selectedItemIndex;
}

@ManagedBean(name = "pollQuestionsController")
@SessionScoped
public class PollQuestionsController implements Serializable {

    private PollQuestions current;
    private DataModel items = null;
    @EJB
    private hu.prf.szavazaskezo.beans.PollQuestionsFacade ejbFacade;
    private PaginationHelper pagination;
    private int selectedItemIndex;

    private String poll_question_id;
    private String poll_id;
}

@ManagedBean(name = "pollFillingsController")
@RequestScoped
public class PollFillingsController implements Serializable {

    private PollFillings current;
    private DataModel items = null;
    @EJB
    private hu.prf.szavazaskezo.beans.PollFillingsFacade ejbFacade;
    private PaginationHelper pagination;
    private int selectedItemIndex;
}

@ManagedBean(name = "pollAnswersController")
@RequestScoped
public class PollAnswersController implements Serializable {

    private PollAnswers current;
    private DataModel items = null;
    @EJB
    private hu.prf.szavazaskezo.beans.PollAnswersFacade ejbFacade;
    private PaginationHelper pagination;
    private int selectedItemIndex;
}
```