**Követelményspecifikáció**

Wikipedia-követő alkalmazás fejlesztése desktopra

Szoftverarchitektúrák házi feladat

# Feladatkiírás

A feladat egy olyan alkalmazás létrehozása, melynek segítségével a felhasználók feliratkozhatnak egy-egy wikipédia szócikkre és email értesítést kaphatnak, ha a cikk tartalma megváltozik. Állítható paraméterek: a frissítési gyakoriság és az "érzékenység" (mekkora módosítás számít relevánsnak). Egy felhasználó legfeljebb 5 szóra iratkozhat fel és legfeljebb napi egy frissítést kérhetnek, de a rendszergazda-jogosultságú felhasználók megnövelhetik ezeket a limiteket egy-egy felhasználó esetében.

# A fejlesztői csapat

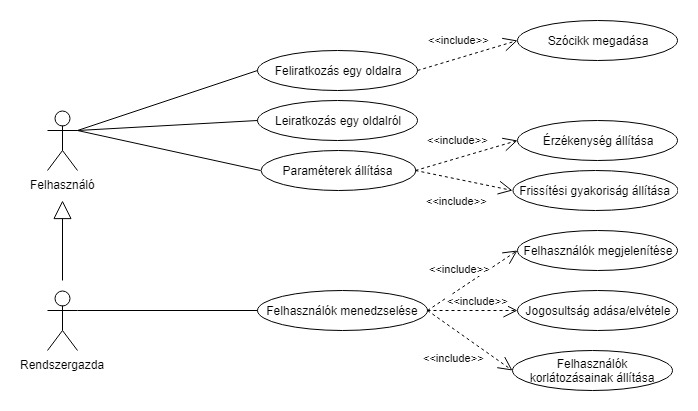
|  |  |  |
| --- | --- | --- |
| Csapattag neve | Neptun-kód | Email-cím |
| Kinyik Bence | WXI48Y | kinyik94@gmail.com |
| Tóth Lajos Gábor | QVMZHU | tothlajosg@gmail.com |

# Részletes feladatleírás

A projekt során célunk egy olyan alkalmazás készítése, amely képes egy felhasználót e-mail-ben értesíteni, amennyiben egy olyan wikipedia szócikk megváltozik, amelyre előzetesen feliratkozott. A felhasználók jogosultság szerint két csoportba oszthatók: egyszerű felhasználó és rendszergazda. Egy felhasználó bármely jogosultsági csoportba tartozik, képes a frissítési gyakoriságot és egy úgynevezett érzékenységi paramétert beállítani, azonban egy egyszerű felhasználó legfeljebb öt oldalra iratkozhat fel és legfeljebb napi egy frissítést kaphat, míg a rendszergazda-jogosultságú felhasználó módosíthatja is ezeket a limiteket egy már beregisztrált felhasználónál. A frissítési gyakoriság állításával a lekérdezés gyakoriságát, míg az érzékenységgel állításával az állítható be, hogy mekkora módosítás esetén küldjön értesítő e-mailt a program.

A program első lépésben azonosítja a felhasználót. Egyszerű felhasználó esetén lehetőséget ad egy adott oldalra való feliratkozásra egy Wikipedia honlap címének megadásával, a feliratkozott oldalak megjelenítésére, és leiratkozásra is. A rendszergazda ezen felül képes a regisztrált felhasználók megjelenítésére, egy adott felhasználónál a frissítési gyakoriság és az érzékenység átállítására, valamint egy egyszerű felhasználónak rendszergazdai jogot is adhat. Amennyiben egy olyan Wikipedia szócikk -amire több, mint egy felhasználó feliratkozott- megváltozik, és megváltozása átlépi a „érzékenységi küszöböt”, a program értesítést küld a feliratkozott felhasználó(k)nak arra az e-mail címre, amely a felhasználóhoz tartozik. A program adatbázisa két rendszergazdát (admin1, admin2) és 3 egyszerű felhasználót tartalmaz (user1, user2, user3).

## Use-case diagram



# Technikai paraméterek

Az alkalmazást C# nyelven a .NET keretrendszer segítségével valósítjuk meg Microsoft Visual Studio 2015-ös verziójú fejlesztőkörnyezetet használva, így a futtató számítógépnek előre telepített .NET keretrendszerrel kell rendelkeznie. Az előző ábrán látható use-case-ek Windows Formokon keresztül vihetők véghez és a fő Form lekicsinyítésekor a háttérben fut. Az implementálandó szoftver a futtatáshoz szükséges adatokat, egy helyi adatbázisban tárolja (felhasználónév, e-mail cím, jelszó, jogosultság, feliratkozott oldalak utolsó lekérdezés ideje, stb). A lekérdezésekhez és az oldalak megváltozásának a detektálásához a MediaWiki API-t használjuk, ami egy olyan webszolgáltatás, amivel különböző lekérések intézhetők a Wikipedia felé (korábbi verziók lekérdezése, egy oldal utolsó módosításának dátuma, stb.).

# Szótár

**Wikipedia oldal**: Egy olyan szócikk, ami a wikipedia.org domain név alá tartozik (például: *https://en.wikipedia.org/wiki/User:Szarch\_hf*).

**Feliratkozás egy oldalra:** Wikipedia szócikk hozzárendelése a feliratkozó felhasználóhoz. Az e-mail értesítések a szócikk megváltozásakor, de a feliratkozáskor meghatározott paraméterek függvényében kerülnek kiküldésre a felhasználóhoz tartozó e-mail címre.

**Leiratkozás egy oldalról:** Feliratkozás megszüntetése, azaz egy felhasználóhoz rendelt wikipedia szócikk megváltozásához rendelt értesítés küldésének eltörlése.

**Érzékenység:** Annak meghatározásan, hogy egy adott wikipedia szócikk hány betűjének megváltozása generál egy eseményt.

# Rendszerterv

**Wikipedia követő**

Szoftverarchitektúrák házi feladat

Kinyik Bence

Tóth Lajos

# A rendszer célja, funkciói és környezete

## Feladatkiírás

A feladat egy olyan alkalmazás létrehozása, melynek segítségével a felhasználók feliratkozhatnak egy-egy wikipédia szócikkre és email értesítést kaphatnak, ha a cikk tartalma megváltozik. Állítható paraméterek: a frissítési gyakoriság és az "érzékenység" (mekkora módosítás számít relevánsnak). Egy felhasználó legfeljebb 5 szóra iratkozhat fel és legfeljebb napi egy frissítést kérhetnek, de a rendszergazda-jogosultságú felhasználók megnövelhetik ezeket a limiteket egy-egy felhasználó esetében.

## A rendszer által biztosítandó tipikus funkciók

Vázlatosan az alábbi funkciók biztosítását várjuk el a rendszertől. (A funkciók részletes definíciója szintén a követelményspecifikáció dokumentumban olvasható.)

**Egyszerű felhasználó esetében:**

* Feliratkozás egy Wikipedia oldalra
* Leiratkozás egy Wikipedia oldalról
* Paraméterek állítása (érzékenység, frissítési gyakoriság)
* A már feliratkozott oldalak és azok paramétereinek megtekintése

**Admin esetében (az egyszerű felhasználó use-case-eit kiegészítve):**

* Regisztrált felhasználók megtekintése
* Regisztrált felhasználók jogainak állítása
* Felhasználók paramétereinek állítása (érzékenység, frissítési gyakoriság)

## A program környezete

A programot teljes egészében C# nyelven .Net keretrendszer segítségével írtuk, így a futtató számítógépnek előre telepített .NET keretrendszerrel kell rendelkeznie, az adatbázisba való beszúráshoz, lekérésekhez pedig egy helyi SQL szervert hoztunk létre, aminek kezeléséhez a LINQ könyvtárat választottuk.

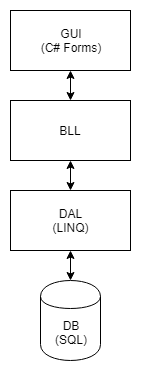
A program egy feltelepített 4.5-ös verziójú .Net keretrendszeren kívül, a manapság használatos számítógépek által nem teljesített követelményt nem támaszt.

# Megvalósítás

Az alkalmazást a feladatkiírásnak megfelelően egy négyrétegű alkalmazásként készítettük el.

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



## Adatbázis réteg

**Célja:** Az adatbázis réteg felel az adatok perzisztálásáért.

Erre mi a Microsoft SQL Server-t választottuk, hiszen egyszerűen létrehozható, módosítható és könnyen mozgatható a fejlesztés alatt (Visual Studio-ban is elérhető), ami a tesztelést nagyban megkönnyíti. Az adatbázisban található adatokat egy lokális mdf kiterjesztésű adatbázis fájlban tároljuk.

## Adathozzáférési réteg (Data Access Layer)

**Célja:** a külső, objektumrelációs leképezést (ORM) biztosító eszközzel együttműködve adathozzáférés biztosítása a felsőbb rétegek számára.

Ennek megfelelően a réteg feladatai:

* üres adatbázis létrehozása (az adatdefiníciós komponensben definiáltak alapján, az ORM eszköz felhasználásával)
* új entitások létrehozása az adatbázisban (az adatdefiníciós komponensben definiáltak alapján, az ORM eszköz felhasználásával),
* igény szerinti adathozzáférés biztosítása felsőbb rétegnek az adatbázishoz (az adatdefiníciós komponensben definiáltak alapján, az ORM eszköz felhasználásával).

**Megjelenés a kódban:**

**TODO**

## Üzleti logika

**Célja:** az adatelérési rétegtől kapott adatok alapján kiszolgálni a grafikus felhasználói felületet. Lényegében egy wrapper facade-ként viselkedik.

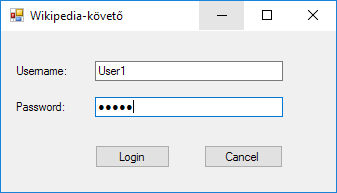
Esetünkben ez a réteg végzi a különböző verziójú wikipedia oldalakon a változások ellenőrzését és adott esetben email értesítést küld a feliratkozott felhasználónak, ezenkívül ebben a rétegben valósítjuk meg a MediaWiki API által szolgáltatott adatok feldolgozását is.

**Megjelenés a kódban:**  WikipediaPoller, WikiAPi, EmailSender osztályokban.

## Grafikus felhasználói felület

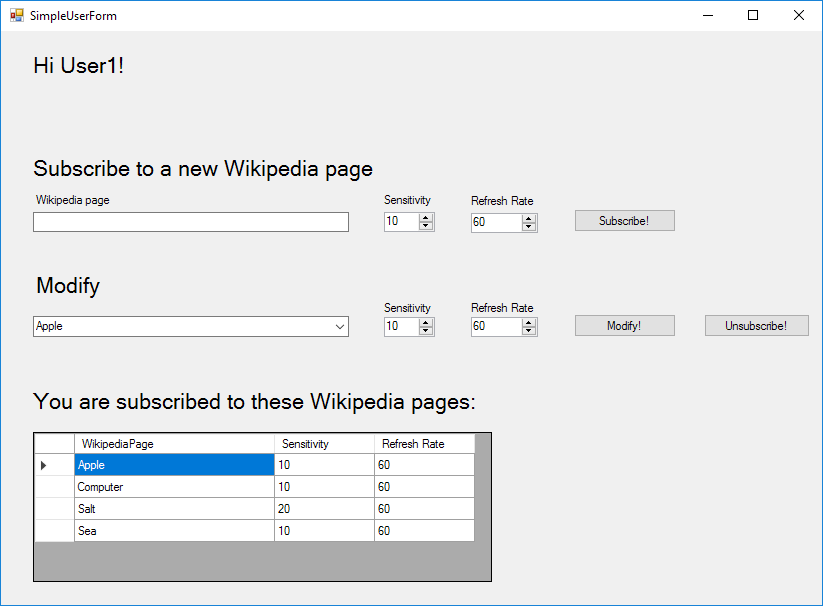
**Célja:** A felhasználók számára egyszerű, könnyen átlátható felületet nyújtani, az összes funkciót elérhetővé tenni.

A Wikipedia-követő felületét C# Formokkal valósítottuk meg. A grafikus felület három ablakot tartalmaz: egy bejelentkező ablakot, az egyszerű felhasználó ablakát és az admin ablakát.



1. ábra Bejelentkező form

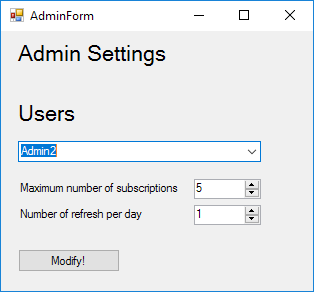
A bejelentkező form teszi lehetővé, hogy a regisztrált felhasználók bejelentkezhessenek és jogosultságuktól függően eljussanak a később bemutatott formokra, ahol a követelményekben definiált use-case-eket hajthatják végre. Helyes felhasználónév/jelszó beírásakor a Login gomb megnyomásával térhetnek át a felhasználók a jogosultságuknak megfelelő formokra.



2. ábra Egyszerű felhasználó form

Amennyiben egy egyszerű felhasználó jelentkezett be a 2. ábrán látható ablak fogadja. A felső sorban új oldalra tud feliratkozni a megfelelő érzékenység és a frissítési gyakoriság beállításával (amennyiben a feliratkozási limitje nem haladja meg a feliratkozások számát).

A középső sorban a bejelentkezett felhasználó a kiválasztott oldalhoz tartozó paramétereit tudja módosítani, illetve leiratkozni, amennyiben nem kíván több e-mail értesítést kapni az adott oldal megváltozásáról.

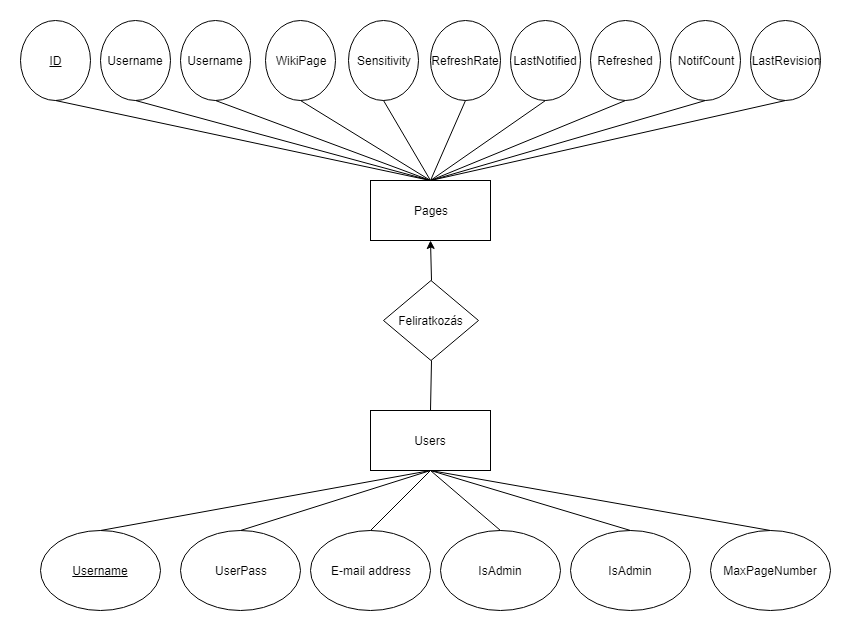


3. ábra Admin form

Amennyiben valaki admin jogosultsággal lép be a 3. ábrán látható ablak fogadja. Itt a kiválasztott felhasználók limitjeit tudja változtatni.

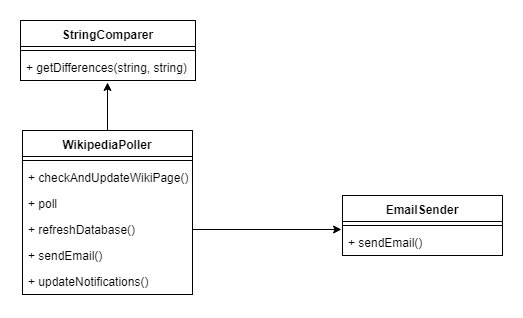
## E-K diagram

Az alábbi ábrán az egyed-kapcsolat diagram látható:



Ennek megfelelően két táblát hoztunk létre az SQL adatbázisunkban: Users és Pages. A Users táblában a felhasználókhoz tartozó adatokat tároljuk a Pages-ben, pedig a feliratkozásokhoz tartozó adatokat. Megjegyzendő, hogy a Users táblához csak a fejlesztők tudnak új sort felvenni.

## Osztálydiagram



Az alkalmazás indításakor egy szálat indítunk, így a program kezdetben két szálon fut. Az egyik szál a GUI-ért, a másik szálon a WikipediaPoller egy objektumának poll függvénye. Ez az objektum felelős azért, hogy adott időközönként (1 perc) végignézze az összes Wikipedia oldalt, amire a felhasználók feliratkoztak, összehasonlítsa az esetleges új állapotot az előző állapottal (StringComparer) és adott esetben értesítést küljdön a felhasználónak az EmailSender osztály segítségével.

# Telepítési leírás

A program .Net keretrendszerrel rendelkező számítógépeken azonnal futtatható, nem szükséges telepíteni. Az adatbázis példaadatokat tartalmaz, amivel rögtön használhatóvá válik.

# A program készítése során használt eszközök

* Microsoft Visual Studio 2015
  + Fejlesztőkörnyezet
* Microsoft SQL Server
  + Adatbázis-kezelő rendszer
* Microsoft Word
  + Dokumentáció írássa
* GitHub
  + Verziókezelés, csapatmunka támogatására
* draw.io weboldal
  + Ábrák készítése

# Összefoglalás

Munkánk során megterveztük, implementáltuk illetve dokumentáltuk a Wikipedia-követő rendszert. Az elkészített alkalmazással Wikipedia oldalakra tudunk feliratkozni és a megfelelő mértékű változás esetén e-mail értesítést küldeni a felhasználóknak.

A megvalósított alkalmazás négyrétegű architektúrát használ: adatbázis réteg, adatelérési réteg, üzleti logikai réteg és felhasználói felület. Az alkalmazás az adatokat egy helyi sql fájlban tárolja, ezt éri el az adatelérési réteg. A legnagyobb kihívást az üzleti logikai réteg implementálása volt, mivel erre a feladatra nem voltak előre implementált megoldások. A GUI-t C# formokkal oldottuk meg.

# Továbbfejlesztési lehetőségek

A wikipedia követő jelen állapotában egy jól működő, a specifikációt teljesítő program, azonban számos továbbfejlesztési lehetőség rejlik benne:

* Új felhasználók regisztrálása
* A program (adatbázis) távoli elérése és használata, mint szerveralkalmazás