



# HÍRPORTÁL

KÉSZÍTETTE:

[adam.janos.bukovics@gmail.com](mailto:adam.janos.bukovics@gmail.com)

# TARTALOMJEGYZÉK

Felhasználói dokumentáció.....	4
Osztályok statikus leírása.....	5
User.....	5
Felelőssége.....	5
Ősosztályok.....	5
Attribútumok.....	5
Metódusok.....	5
Content.....	6
Felelőssége.....	6
Ősosztályok.....	6
Attribútumok.....	6
Metódusok.....	6
Comment.....	7
Felelőssége.....	7
Ősosztályok.....	7
Attribútumok.....	7
Metódusok.....	7
Article.....	8
Felelőssége.....	8
Ősosztályok.....	8
Attribútumok.....	8
Metódusok.....	8
NewsPortal.....	9
Felelőssége.....	9
Ősosztályok.....	9
Attribútumok.....	9
Metódusok.....	9
UML osztálydiagramm.....	10
Összegzés.....	11
Mit sikerült és mit nem sikerült megvalósítani a specifikációból?.....	11
Mit tanultál a megvalósítás során?.....	11
Továbbfejlesztési lehetőségek.....	11
Képernyőképek a futó alkalmazásról.....	12

## Felhasználói dokumentáció

A program használata előtt be kell jelentkezni, saját felhasználónév és saját jelszóval, ezután tudod csak használni a program fő részét. Ez a „fő rész” tulajdonképpen minden felhasználónak ugyanaz, van egy engedély lista, amin ha nem vagy Author, akkor nem tudsz cikket írni, ilyenkor engem keress meg. A lényeg, hogy a menüpontoknak megfelelően választasz egy számot 0-6 -ig, ami mellé részletesen le van írva, hogy mit fog csinálni a program, de azért itt is leírom:

- A 0-ás számmal tudsz kilépni a rendszerből.
- Az 1-es számmal tudsz minden eddig felvett cikket listázni.
- A 2-es számmal tudsz minden felhasználót kilistázni. (Sajnos, ez nincsen jogokhoz kötve[ide terveztem az admint])
- A 3-mas számmal tudsz Új cikket írni, ha van hozzá jogod.
- A 4-es számmal tudsz egy adott cikkhez kommentezni.
- Az 5-ös számmal tudsz egy adott cikkhez tartozó kommenteket lekérdezni. (Sajnos ez sincsen letárolva [kellett volna egy txt pluszban])
- A 6-os számmal tudsz egy adott író szerint keresni a cikkek között.

Igazából, az egész működése nagyrészt csak egy kereső program, aminek a célja, a hatékony keresés a cikkek, vagy a kommentek között, esetlegesen az emberek között. Van ami hiányzik még, ahhoz, hogy teljesen tökéletes legyen. A programban figyelni kell, mivel nincsenek letárolva a kommentek, ezért minden egyes újraindításkor törlődnek az aktuálisan hozzáírt kommentek.

# Osztályok statikus leírása

## User

### Felelőssége

A lényege, hogy egy adott felhasználót mintázzon le, tehát tárolok benne egy nevet, egy hozzátartozó jelszót és egy role, amit majd engedélyeknél nézek meg.

### Ősosztályok

Nincsen ősosztálya.

### Attribútumok

#### *Privát*

Van egy username, egy password és egy role.

#### *Védett*

Nincsen ilyen attribútuma.

#### *Publikus*

Nincsen ilyen attribútuma.

### Metódusok

#### *Privát*

Nincsen ilyen metódusa.

#### *Védett*

Nincsen ilyen metódusa.

#### *Publikus*

Van két konstruktora, egy saját teljes, illetve egy másoló. Megvan az összes gettere / settere. Van egy saját kiírató függvénye a ShowDashboard, ami a username-hez tartozó role-t írhatja ki meg a username-eket. Vannak jogosultságot ellenőrző függvények, illetve egy teljes «operátor túlterhelés, ami a ShowDashboard-hoz hasonlóan, csak itt a password-öt is kiíratja.

## Content

### Felelőssége

A fő célja, hogy interfészként működjön a kommentnek és az article-nek.

### Ősosztályok

Nincsen ősosztálya.

### Attribútumok

#### *Privát*

Nincsen ilyen attribútuma.

#### *Védett*

Nincsen ilyen attribútuma.

#### *Publikus*

Nincsen ilyen attribútuma.

### Metódusok

#### *Privát*

Nincsen ilyen metódusa.

#### *Védett*

Nincsen ilyen metódusa.

#### *Publikus*

Van egy default destruktora, illetve van egy tisztán virtuális display függvénye. Meg egy teljesen virtuális getType-ja.

## Comment

### Felelőssége

Igazából ez a class azt a célt szolgálja, hogy egy adott cikkhez hozzáfűzhessük gondolatainkat.

### Ősosztályok

Content-ből származik le, ennek a class-nek az interfésze a Content.

### Attribútumok

#### *Privát*

Van egy text, ami maga a komment, van egy username ez az, hogy ki kommentelt, illetve egy az letárolva, hogy melyik cikkhez tartozik, ezt a cikk nevével oldottam meg.

#### *Védett*

Nincsen ilyen attribútuma.

#### *Publikus*

Nincsen ilyen attribútuma.

### Metódusok

#### *Privát*

Nincsen ilyen metódusa.

#### *Védett*

Nincsen ilyen metódusa.

#### *Publikus*

Van egy teljes konstruktora, mindenféle gettere, amik az adott tagváltozókat kérdezik le. Ugyanígy megvannak a setter-ek is, kivéve a Type-hoz, mert azt nem akarjuk módosítani. Emellett van egy «operátor túlterhelés, ami az adott kommentelő nevét és azt, hogy mit kommentelt kinyomja az std::cout-ra. Van egy display is, ami meg az összes dolgát nyomja ki.

## Article

### Felelőssége

Ennek a class-nak megfelelően tárolódik le a programban egy újság.

### Ősosztályok

Content-ből származik le, ennek a class-nek az interfésze a Content.

### Attribútumok

#### *Privát*

Egy újságcímet, egy szerzőt és magát a cikk szövegét tárolja le.

#### *Védett*

Nincsen ilyen attribútuma.

#### *Publikus*

Nincsen ilyen attribútuma.

### Metódusok

#### *Privát*

Nincsen ilyen metódusa.

#### *Védett*

Nincsen ilyen metódusa.

#### *Publikus*

Van egy teljes konstruktora, getter-ek és setter-ek a Comment-hez hasonlóan itt is megvannak. A display függvény megformázottan, az egészet kitolja az std::cout-ra. Az operátor« túlterhelés pedig csak a címet és az újság szövegét írja ki.

## NewsPortal

### Felelőssége

Ez a class tárolja az összes user-t az összes article-t és az összes comment-et. És ezen a felületen fogunk dolgozni az összes eddig megírt class-al.

### Ősosztályok

Nincsen ősosztálya.

### Attribútumok

#### *Privát*

Van 3 db STL tárolónk (vektorok), amikből az egyik a user-eket, a másik az újságokat, a harmadik pedig a comment-eket tárolja.

#### *Védett*

Nincsen ilyen attribútuma.

#### *Publikus*

Nincsen ilyen attribútuma.

### Metódusok

#### *Privát*

Nincsen ilyen metódusa.

#### *Védett*

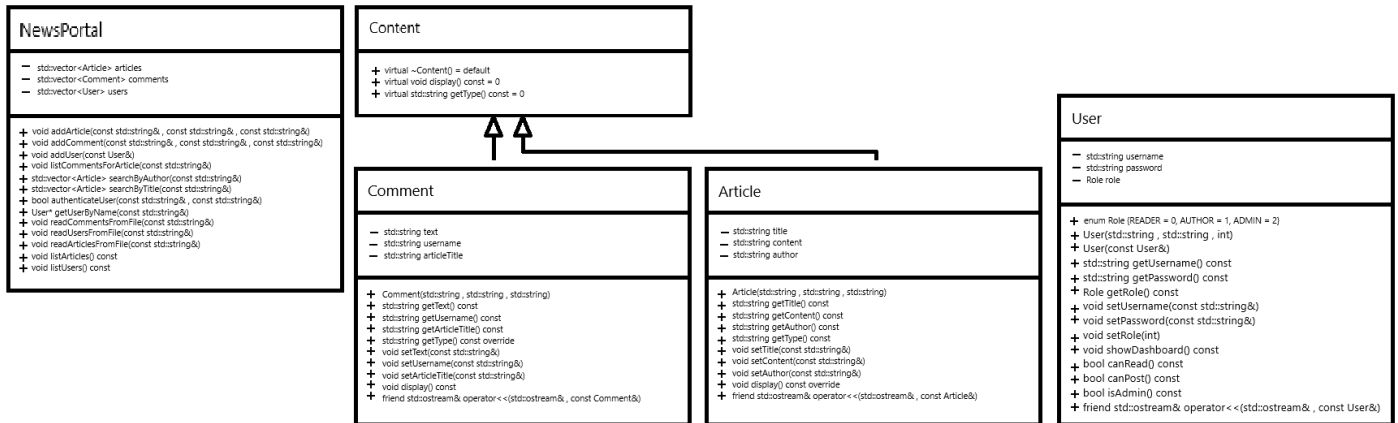
Nincsen ilyen metódusa.

#### *Publikus*

Van 3 db függvényem, ami a privát részben tárolt vektorokhoz adnak hozzá 1 elemet. Így lehet felvenni új user-eket, új újságokat és új kommenteket. Van listcommentsForArticle() függvényem, ami kap egy újság címet és visszaadja az összes kommentet hozzá. Ezen kívül újságot tudok író és cím szerint is keresni külön. Van egy a belépésnél használt függvény authenticateUser() ami eldönti, hogy jó-e a jelszó és a hozzá tartozó username. Ezen kívül van még 3 db olyan függvény, ami a teszt txt.-kből olvassa ki az összes adatot és azzal tölti fel a privát tagváltókat. Illetve van két függvényem, ami a cikkeket és a user-eket listázza ki egy vektorba.



# UML osztálydiagramm



# Összegzés

## Mit sikerült és mit nem sikerült megvalósítani a specifikációból?

Amit szerettem volna, azt elértem ezzel a programmal, de azért nem lett teljes körű ez a megoldás.

## Mit tanultál a megvalósítás során?

Hogyan kell STL tárolókat használni, milyen specifikusabb függvényei van a <vector>-nak. Meg az objektum orientált programozással ismerkedtem meg, ami sokat fejlesztett a programozói gondolkodásomon.

## Továbbfejlesztési lehetőségek

Mindenképpen lehetne még a választható opciókat bővíteni és ennek megfelelően többfajta lekérdezésre lenne képes a program. Lehetne nem csak kifejezetten (.txt) fájlokból beolvasni és egy egész adatbázisból. Lehetne komolyabb interface-t használni cmd helyett.



# Képernyőképek a futó alkalmazásról

```

 22 |
 23 |
 24 |
 25 |
 26 |
 27 |
 28 |
 29 |
 30 |
 31 |
 32 |
 33 |
 34 |
 35 |
 36 |
 37 |
 38 |
 39 |
 40 |
 41 |
 42 |
 43 |
 44 |
 45 |
 46 |
 47 |
 48 |
 49 |
 50 |
 51 |
 52 |
 53 |
 54 |

=====
Felhasználónév: |
=====

```

```

 22 |
 23 |
 24 |
 25 |
 26 |
 27 |
 28 |
 29 |
 30 |
 31 |
 32 |
 33 |
 34 |
 35 |
 36 |
 37 |
 38 |
 39 |
 40 |
 41 |
 42 |
 43 |
 44 |
 45 |
 46 |
 47 |
 48 |
 49 |
 50 |
 51 |
 52 |
 53 |
 54 |

===== Menü =====
0. Kilépés
1. Cikkek Listázása
2. Felhasználók Listázása
3. Új cikk hozzáadása
4. Komment hozzáadása
5. Kommentek Listázása egy cikkhez
6. Felhasználó szerinti cikk keresés
=====

Válassz egy opciót (0-6): |
=====

```