

SZAKDOLGOZAT



MISKOLCI EGYETEM

Alkalmazás fejlesztés, CMS rendszerek, Egyedi tartalomkezelő rendszer fejlesztése

Készítette:

Nagy László

2018-2021. Programtervező informatikus-szak

Témavezető:

Dr. Kovács László

MISKOLC, 2020

SZAKDOLGOZAT FELADAT

Nagy László LWI9Z1 programtervező informatikus jelölt részére.

A szakdolgozat tárgyköre: Alkalmazás fejlesztés, CMS rendszerek

A szakdolgozat címe: Egyedi tartalomkezelő rendszer fejlesztése

A feladat részletezése:

Szakdolgozat témája egy egyedi tartalomkezelő rendszer (CMS) kidolgozása.
Az alábbi főbb tevékenységpontokra szeretnék kitérni:

- A CMS rendszerek jellemzése, előnyei és hátrányai
- A CMS használatának biztonsági kérdései
- Néhány nyílt forrású tartalomkezelő rendszer összehasonlítása
- CMS motor funkcióinak megtervezése
- Egyedi CMS motor fejlesztése

Témavezető(k): Dr. Kovács László, Általános Informatikai Intézeti Tanszék vezetője, egyetemi tanár

Konzulens(ek):

A feladat kiadásának ideje: 2020. szeptember 30.

.....
szakfelelős

EREDETISÉGI NYILATKOZAT

Alulírott; Neptun-kód:
a Miskolci Egyetem Gépészmérnöki és Informatikai Karának végzős
szakos hallgatója ezennel büntetőjogi és fegyelmi felelősségem tudatában nyilatkozom
és aláírással igazolom, hogy
című szakdolgozatom/diplomatervem saját, önálló munkám; az abban hivatkozott szak-
irodalom felhasználása a forráskezelés szabályai szerint történt.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- szó szerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát megismertem, és tudomásul veszem,
hogy plágium esetén szakdolgozatom visszautasításra kerül.

Miskolc, év hó nap

.....

Hallgató

1.

szükséges (módosítás külön lapon)

A szakdolgozat feladat módosítása

nem szükséges

.....

dátum

.....

témavezető(k)

2. A feladat kidolgozását ellenőriztem:

témavezető (dátum, aláírás):

konzulens (dátum, aláírás):

.....

.....

.....

.....

.....

.....

3. A szakdolgozat beadható:

.....

dátum

.....

témavezető(k)

4. A szakdolgozat szövegoldalt

..... program protokollt (listát, felhasználói leírást)

..... elektronikus adathordozót (részletezve)

.....

..... egyéb mellékletet (részletezve)

.....

tartalmaz.

.....

dátum

.....

témavezető(k)

5.

bocsátható

A szakdolgozat bírálatra

nem bocsátható

A bíráló neve:

.....

dátum

.....

szakfelelős

6. A szakdolgozat osztályzata

a témavezető javaslata:

a bíráló javaslata:

a szakdolgozat végleges eredménye:

Miskolc,

.....

a Záróvizsga Bizottság Elnöke

Tartalomjegyzék

| | |
|---|-----------|
| 1. Bevezetés | 6 |
| 1.1. Célkitűzés | 6 |
| 1.2. Motiváció | 6 |
| 2. CMS-ek általános ismertetése | 8 |
| 2.1. CMS definíció | 8 |
| 2.2. Mikor használjunk tartalomkezelő rendszert? | 9 |
| 2.3. CMS felhasználókezelés | 9 |
| 2.4. CMS választás | 9 |
| 2.5. CMS típusok | 10 |
| 2.6. Nyílt forráskódú, vagy egyedi fejlesztés? CMS? | 12 |
| 2.7. Java alapú CMS-ek | 14 |
| 2.7.1. Hozzá tartozó definíciók | 14 |
| 2.7.2. Nyílt forráskódú Java portálok | 14 |
| 2.7.3. A Liferay bemutatása | 14 |
| 2.8. A CMS-ek információbiztonsági kérdései | 14 |
| 3. Fejlesztői dokumentáció | 15 |
| 3.1. Konceptió | 15 |
| 3.2. A Microservicekről általában | 16 |
| 3.3. Programspecifikáció | 16 |
| 4. Összefoglalás | 18 |
| Irodalomjegyzék | 19 |
| Adathordozó használati útmutató | 20 |

1. fejezet

Bevezetés

Manapság az internet világa nagy mértékben szerepel az életünkben. Az interneten keresztül szerzünk általában információkat a világban történt dolgokról, itt olvasunk különféle témában híreket, érdekes cikket, fórumokat, tehát leginkább az internet által tájékozódunk. Ezáltal, ha például egy magánszemély, vagy egy vállalat a saját termékét szeretné népszerűsíteni, nélkülözhetetlen hozzá egy weboldal, ahol ezt reklámozza. Ehhez alapvetőleg szüksége lenne webfejlesztői tudásra, viszont ebben a felhasználók nagy segítségére lehetnek a tartalomkezelő rendszerek, amiket megfelelő szaktudás nélkül is egyszerűen lehet használni, hogy a saját weboldalunkat létre tudjuk hozni.

1.1 Célkitűzés

Szakdolgozatomban szeretném feltárni a tartalomkezelő rendszerek használatának az előnyeit, illetve hátrányait, ezek típusait.

Egy külön fejezetben szeretném megválaszolni a CMS rendszerek információbiztonsági kérdéseit, majd fő célkitűzésként egy egyedi fejlesztésű CMS rendszer implementálását tűztem ki.

Ez a hagyományos szoftverarchitektúrájú rendszereken túl, egy sokkal eredményesebb, modulárisabb felépítésű rendszer lesz. Ez egy Microservice alapú megoldás segítségével fog létrejönni.

1.2 Motiváció

Azért esett a választásom ezen alkalmazás fejlesztésére, mivel a CMS rendszerek egy olyan megoldást nyújtanak, ami a webfejlesztést hatalmas mértékben megkönnyíti és meggyorsítja.

Manapság az emberekben is egyre jobban feltámad a vállalkozói ösztön, nagyon sokan döntenek úgy, hogy a saját ötleteiket meg szeretnék valósítani és egy vállalkozás

keretében ebből pénzt szeretnének csinálni.

A vállalkozásokon kívül is nagyon sokoldalú a tartalomkezelő rendszerek felhasználóinak köre. Magánszemélyként is létrehozhatunk weboldalakat például egy saját blog vezetéséhez, vagy fórumok kezeléséhez, vagy például egy-egy sportcsapatnak ez jelenti a legfőbb útját a kommunikációnak a szurkolók, érdeklődők felé.

Tehát elég széles körben használják a CMS-eket, ezért nagyon fontos, hogy minél jobban ki legyenek dolgozva ezek a rendszerek, minél több funkció szerepeljen benne, hogy mindenki megtalálja benne a számára megfelelő funkcionalitást.

Ez viszont azt eredményezi, hogy ezek a CMS-ek nagyon nagy rendszerekké válnak és a modularitás fontossága is egyre jobban csökken. Erre tud megoldást adni a Microservice technológia.

Microservice-k segítségével ugyanazokat a funkciókat sokkal kisebb egységekben tudjuk eltárolni, ezáltal sokkal jobban kezelhetőek lesznek ezek a rendszerek és erősíteni tudjuk a moduláris felépítést.

2. fejezet

CMS-ek általános ismertetése

2.1 CMS definíció

A CMS (Content Management System), olyan szoftver, amely speciális műszaki ismeretek nélkül segíti a felhasználókat egy weboldal tartalmának létrehozásában, kezelésében és módosításában.

Ezekhez a rendszerekhez a weboldal szerveren történő telepítésén kívül-nincs szükség fejlesztői tudásra, minimális szakértelem mellett üzemeltethető a weboldal. Egyszerűen tudunk felhasználókat, tartalmi elemeket létrehozni és módosítani anélkül, hogy ismer-nénk a webes nyelveket, kódokat. A CMS rendszerek célja az egyszerű tartalomfelvitel, gondolok itt mind szövegre, képre, vagy bármilyen csatolt állományra és ezek elérhető-vé tétele az oldalon a látogatók számára.

Egyik legfontosabb tulajdonsága ezeknek a rendszereknek a dinamizmus. Ez azt je-lenti, hogy megfelelő jogosultság birtokában az oldal tartalmát és annak megjelenését bármikor, bárholnan meg lehet változtatni.

A tartalomkezelő rendszerek template-k segítségével választják el egymástól a tartal-mat és a megjelenést. Ezek a sablonok úgymond keretrendszerként működnek, mi-vel esetleges arculatváltás esetén elegendő csak a template-t lecserélni és a tartalom-alkalmazkodva ehhez a sablonhoz-változtatlan formában, de új külsővel jelenik meg.

A későbbi bővítés is egyszerűen kivitelezhető, mivel ha új funkciókat szeretnénk integ-rálni a rendszerbe, akkor az grafikai átalakítással nem jár, nem kell újraszerkesztenünk az érintett részt.

Ezekre a rendszerekre továbbá jellemző a moduláris felépítés, valamint ezek többfel-használós rendszerek, tehát egyszerre többen is tudják szerkeszteni az oldalt. A legtöbb CMS keresésoptimalizált, így könnyen tudunk keresni a honlapon elhelyezett tartalmak között.

A CMS rendszerek legfőbb tulajdonságai:

- Könnyű kezelhetőség
- Szelektív hozzáférési jogosultságok

- Többnyelvűség
- Keresőoptimalizálási eszközök
- Reszponzivitás
- Nagyon kis költségűek

2.2 Mikor használjunk tartalomkezelő rendszert?

CMS rendszert leginkább olyan weboldalnál érdemes használni, ahol fontos az aktív kommunikáció, tehát ahol az oldalra látogatók számára folyamatos információt kell közvetítenünk, ezeket frissíteni kell. Ilyenek például a fórumok, blogok, portálok, webáruházak.

A tartalomkezelő rendszerek tehát összességében segítséget nyújtanak mind például a magánszemélyeknek a saját személyes blogoldaluk kialakítására, vagy pedig a vállalatoknak, cégeknek a weboldalának létrehozását, csökkentve ezzel a kiadási költséget. Nem kell megbízniuk egy webfejlesztő céget az oldaluk elkészítésével, ezt megtehetik maguknak, nagyobb erőfeszítés nélkül.

2.3 CMS felhasználókezelés

A CMS rendszerben nagyon fontos funkció a felhasználó kezelés. Minden rendszernek van egy adminisztrátora, aki teljes jogkörrel rendelkezik, bármit megtehet az oldalon. Ő osztja majd ki a többi felhasználó számára a megfelelő jogosultságokat. Az azonosítás regisztrálás és bejelentkezés útján történik.

Ezek alapján az oldalra látogatók két csoportba sorolhatók: azonosított felhasználók, akik már regisztrált tagjai a rendszernek, saját felhasználónévvel és jelszóval rendelkeznek, mellyel beléphetnek a webhelyre. Jogosultságaik korlátozottak, a jogokat az adminisztrátor rója ki.

És vannak az ismeretlen felhasználók, akik ugyan ellátogatnak a webhelyre, de még nem tagjai annak, nem rendelkeznek saját regisztrációval. Ezen felhasználók általában csak a közzétett tartalmakat olvashatják, illetve menthetik le, egyéb jogosultságuk nincs. Bizonyos esetekben számukra is nyitott néhány opció: publikus szavazásokban vehetnek részt, vagy anonimként hozzászólásokat fűzhetnek bizonyos tartalmakhoz. Ezekről szintén az adminisztrátor rendelkezik.

A weboldal típusától függően hozhatunk létre különböző jogosultsági köröket, így akár nem egy-egy személynek osztjuk ki a jogosultságokat, hanem a felhasználókat használat szerint csoportokba rakjuk és ezeknek a csoportoknak adjuk meg a jogosultságait.

2.4 CMS választás

Számos CMS platform közül tudunk választani, amik egymástól eltérő tulajdonságokkal rendelkeznek.

Dönthetünk akár a szabad licenc?, nyílt forráskódú rendszerek mellett, vagy akár a költségesebb egyedi fejlesztés mellett is.

Vannak egyszer?bb és komplexebb rendszerek, magyar nyelv?ek, vagy csak idegen nyelv?ek.

Különféle szerverkörnyezetet használnak, például Java, PHP, .Net. Vannak általános célú CMS-ek és speciális verziók, mint például az e-learning oktatási céllal.

Számos CMS rendszert a használata el?tt ki lehet próbálni, ez meghatározó lehet a döntésünkön, mert így tesztelni tudjuk az el?re telepített rendszereket. Ez nagy el?nyként számít a nyílt forráskódú CMS-eknél. Ezeket a CMS Award oldalon, illetve az Open-source CMS oldalon van lehet?ségünk tesztelni.

Magyarországon talán a legelterjedtebb CMS-ek a Drupal és Joomla, mint általános célú rendszer, a Moodle oktatási oldalak fejlesztéséhez és a Wordpress, ami a blogok esetén a legnépszerűbb.

Miel?tt kiválasztjuk a megfelel? CMS rendszert, tisztába kell lennünk a CMS-ek tulajdonságaival.

2.5 CMS típusok

Ebben a részben szeretném ismertetni a legnépszerűbb CMS-eket a felhasználók körében. Ezek pedig a Wordpress, Drupal és a Joomla.

Ez a három rendszer mer?ben eltér egymástól, de találunk közös vonásokat mind a működésükben, mind pedig felépítésükben, ugyanis mindhárom CMS mögött PHP programnyelv és MySQL adatbázis van.

Mindhárom személyre szabható, mert további kiegészít?kkel bővíteni tudjuk, és a weboldalak kinézetét gyorsan meg tudjuk változtatni a különféle sablonok telepítésével. Ez független a honlap tartalmától, mivel a weboldal adatai adatbázisban vannak eltárolva, így az esetleges módosítás nem veszélyezteti a már meglév? adataink elvesztését.

Most pedig szeretném egyesével ismertetni a három legkedveltebb tartalomkezel? rendszert.

Wordpress

2003-ban látta meg a napvilágot, leginkább blogok tervezésére fejlesztették, azonban napjainkra már rengeteg weboldal alapja, az egyszer? bemutatkozó oldalaktól kezdve webshopokon át, a teljes kör? szociális hálózatokig.

Népszerűségének kulcsa a könny? használhatóság, egyszer?ség, és a jól beállítható keres?optimalizálási paraméterek. Nincs szükség komolyabb programozói ismeretekre,

hogyan egy weboldalt létrehozzunk magunknak. A Wordpress egyszerre telepítésével már megoldható ez.

Alap weboldalunkat további kiegészítéssel bővíthetjük, amelyekkel növelhetjük a weboldal funkcionalitását, így akár webáruházzá is fejleszthetjük. A telepíthető sablonok segítségével gyorsan és egyszerre változtatható meg a weboldalunk kinézete. A widgetek, azaz kisebb kiegészítő programok pedig tovább javítják a Wordpress használhatóságát.

A Wordpress rendelkezik a legnagyobb fejlesztői közösséggel, így számtalan oktatóanyag is a rendelkezésünkre áll.

Drupal

A Drupal tartalomkezelő rendszer volt az első a három közül, amelyik megjelent. 2000-ben született meg a drop.org weboldal forráskódjaiból. Ez a CMS szakértők számára készült, így nem meglepő, hogy ez a legkevésbé használt webmotor.

Ugyanúgy megváltoztathatjuk weboldalunk kinézetét különféle sablonokkal, az alrendszer pedig modulok használatával tovább tudjuk fejleszteni, továbbá PHP és MySQL ismeretekkel egyedi fejlesztésű modulokat tudunk beintegrálni a weboldalunkba.

A Drupal rendszer közössége is nagyon aktív, számos fórumon tudunk keresgélni segítség után. A Drupal dokumentációja folyamatosan frissül, így minden információ rendelkezésre áll a CMS telepítésétől kezdve a weboldalépítés és a kiegészítők telepítésének folyamatáról.

Ez a rendszer nem javasolt egyszerűbb weboldalak esetében, mivel tartalmaz olyan kiegészítőket, amik komplexebbek és nem feltétlen van rá szükségünk a weboldalunkhoz, tehát a weboldalunk nem használná őket, de mégis lassulna ezek miatt.

Kevés ingyenes sablon áll rendelkezésünkre, amik viszont nem biztosítják a WordPress vagy a Joomla által kínált minőséget. A felhasználói felület is sokkal kevésbé felhasználóbarát.

2005-ben Mambo néven jött létre, így a három CMS közül ez számít a legfiatalabbnak.

Joomla

A Joomla fejlesztését egy nagy közösség végzi, saját MVC fejlesztői keretrendszerrel. Hatalmas előnye a Bootstrap integrálása és amiben különbözik a másik két rendszertől, és ez nagy előnyére is vált, az az, hogy ennél teljesen responszív adminisztrációs felülettel találkozunk. Az alap rendszert bővíthetjük további modulokkal, pluginekkel. A telepíthető sablonok segítségével pedig pillanatok alatt megváltoztatható a weboldal kinézete.

Nagyon aktív fejlesztői táborral rendelkezik, ezért könnyen elérhető oktatói anyaggal rendelkezik és a felmerülő kérdéseinkre gyors választ kaphatunk a fejlesztői fórumokon.

Azonban a WordPress-el összehasonlítva kevésbé felhasználóbarát és a weblap sablonok is kevésbé színvonalasak.



2.6 Nyílt forráskódú, vagy egyedi fejlesztés? CMS?

Miután arra jutott döntésünk, hogy weboldalunkhoz CMS rendszert fogunk alkalmazni, fel kell tennünk magunknak a kérdést, hogy nyílt forráskódú, vagy egyedi fejlesztés? tartalomkezel? rendszer mellett döntünk. Ezt több tulajdonság alapján érdemes eldönteni, ezeket szeretném bemutatni ebben a fejezetben.

El?ször is szeretném bemutatni a nyílt forráskódú CMS el?nyeit és hátrányait. Ez egy nyílt forráskódú tartalomkezel? rendszer, amelyet egy közösség fejleszt, és mind a termék, mind a teljes forráskód szabadon és ingyenesen bárki rendelkezésére áll. Ilyenek például a WordPress, Joomla, vagy a Drupal. A nyílt forráskódú CMS-ek nagyon elterjedtek, minden iparágban találkozunk ezzel a technológiával megvalósított weboldallakkal. Legnagyobb vonzereje talán az ingyenességben rejlik, és abban, hogy el?re ki tudjuk próbálni.

Ezzel szemben az egyedi fejlesztés? CMS egy olyan, jellemz?en zárt forráskódú rendszer, amelyet általában egy fejleszt?csapat, vagy cég saját maga fejleszt ki a nulláról, és a használatáért licenszdíjat kell fizetni, viszont a forráskódja rejtett. Legnagyobb el?nyük lehet talán az, hogy kifejezetten egy bizonyos weboldalra specializálják, így a vev? igényei szerint készül el az adott oldal és könnyen optimalizálható és egyedisége kit?nik a sok szabványos template-k közül.

A nyílt forráskódú CMS el?nyei lehetnek:

- Ezek a rendszerek önmagukban jellemz?en ingyenesek. Ebben az esetben csak azért a szolgáltatásért és szaktudásért kell fizetni, amellyel a fejleszt?k telepítik a rendszert és a kés?bbiekben esetlegesen frissíti.
- Továbbá fontos érv még, hogy ezek a rendszerek akár minimális szaktudással adminisztrálhatóak.
- Egy hozzáért? fejleszt? könnyen tudja b?víteni különféle egyedi funkcióval, mivel kódja nyílt, így bárki számára elérhet?.

- Ezeknek a rendszereknek a fejlesztésében több millió felhasználó vesz részt. Ez a hatalmas fejlesztői bázis is nagy előnynek számít.

A nyílt forráskódú CMS-ek ellen felhozott érvek:

- A rendszer folyamatosan frissül, ami viszont számunkra lehet, hogy nem szolgál kedvező eredménnyel. A frissítések során előfordulhat, hogy egy-egy korábban probléma nélkül működő plugin már nem fog működni, vagy hibázni fog. Ezek ellen nem sok mindent tudunk tenni.
- A nyílt forráskódú rendszerek népszerűsége jelenti egyben a sebezhetőségüket is. A nyitott forráskód miatt mindenki számára elérhető a kód, ezzel együtt a jelszavak tárolási módja, vagy a modulok felépítése, így elég sokan próbálkoznak azzal, hogy ezeket a rendszereket feltörjék. Látják ebben a pénzszerzési lehetőséget, ezért ezeknek a kódoknak a feltörésére számos támadási kísérletet tesznek, próbálnak rést találni a rendszerek biztonsági pajzsain.
- A CMS rendszerek tartalmazhatnak olyan file-okat, kódokat, plugineket, amik számunkra feleslegesek, de oldalunkat lassítják.
- Nehezebben optimalizálhatóak.

Az egyedi fejlesztésű CMS-ek előnyei:

- Az egyedi fejlesztés miatt ezek személyre szabott rendszerek, az ügyfél igényeire van összpontosítva. Nem kerülnek bele soha nem használt funkciók és menüpontok, amik lassítanák a rendszert, így a kívánt feladat leghatékonyabb megoldására fókuszál.
- Az esetleges hibák felmerülésekor a rendszer fejlesztőihez fordulva akár pár perceken belül megoldást találhatunk a problémákhoz, nem kell fórumokon keresgelnünk. A licenstdíjért cserébe megbízható, professzionális support-ot kapunk.
- Bármilyen ügyféligény könnyen megvalósítható az integritás megtartása mellett.

Az egyedi fejlesztésű CMS-ek ellen felhozott érvek:

- Egyik legnagyobb hátránya a költséges és időigényes beüzemelés. Egy egyedi fejlesztésű rendszer kiépítése sokkal több idővel és pénzzel jár, mint egy nyílt forráskódú rendszer használata.
- Egy másik hátránya, ami nagyon kis valószínűséggel fordul elő, viszont ezzel is számolni kell, mégpedig, hogy valamilyen okból elveszítjük a kapcsolatot a fejlesztő csapattal. Ilyenkor a weboldal továbbfejlesztése lehetetlenné válik.

2.7 Java alapú CMS-ek

2.7.1 Hozzá tartozó definíciók

2.7.2 Nyílt forráskódú Java portálok

2.7.3 A Liferay bemutatása

2.8 A CMS-ek információbiztonsági kérdései

3. fejezet

Fejlesztői dokumentáció

Ebben a fejezetben ismertetni fogom a fejleszteni kívánt rendszer tervezését, architektúrális tervét. A rendszer elkészítése szakértők tapasztalatai alapján és segítségével történik.

3.1 Konceptió

A webfejlesztésben egy nélkülözhetetlen gyors fejlesztést segít megoldás a cms. azonban a legismertebb piacon lévő megoldásoknál a hagyományos szoftverarchitektúrájú rendszerek vannak. A web-es területen azonban mára már szinte egyeduralkodóvá kezd válni az új fejlesztések területén a Front-end és back-end kettéválasztása. A back-end esetében pedig lehetőség nyílt, egy sokkal inkább sokoldalúbb modulárisabb fejlesztésre. Konkrét példával élve ha veszünk egy Drupal, Liferay, esetében egy webserverre van telepítve egy szolgáltatás. Az nyújtja az oldalak, tartalmak kezelését. Azonban a back-end esetében az architektúrára jellemzően vagy csak java vagy csak php-s bővíthetőség van.

Milyen új CMS architektúra dolgozható ki? Ha a fenn említett szempontokat figyelembe vesszük, akkor erre egy olyan microservice alapú megoldás adhat választ, ami könnyen telepíthető, bővíthetőre nyitott, módosításra zárt. Interface alapú megoldások. Jelen esetben az interfácenak a REST-es végpontokat is tekinthetjük.

A dolgozatban egy tartalom kezelő kerül kidolgozásra. A tartalomkezelő microservicek összessége, ami a hagyományos CMS-esek esetében egy adott oldal html tartalmát határozza meg, maximum néhány makró kifejezéssel.

Fontosabb funkciók:

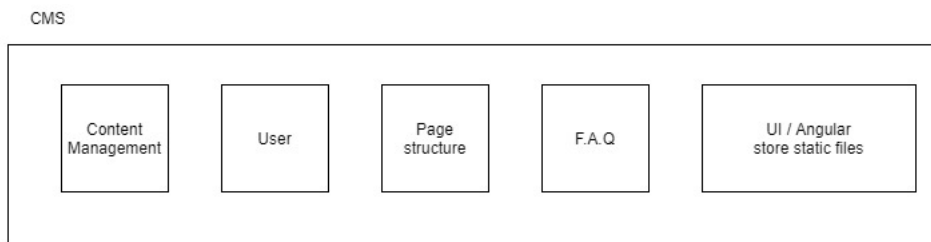
- Lehesse verziózni a tartalmakat (pl.: jogi nyilatkozat oldalát)
- Legyen staging mód (pizskozat kezelés, illetve élesítés előtt lehessen látni a leendő eredményt)
- Multi platformos (desktop, mobilos verziók kezelése)
- Makrók használati lehetőségének biztosítása (bővíthetőre nyitott, módosításra zárt elv)

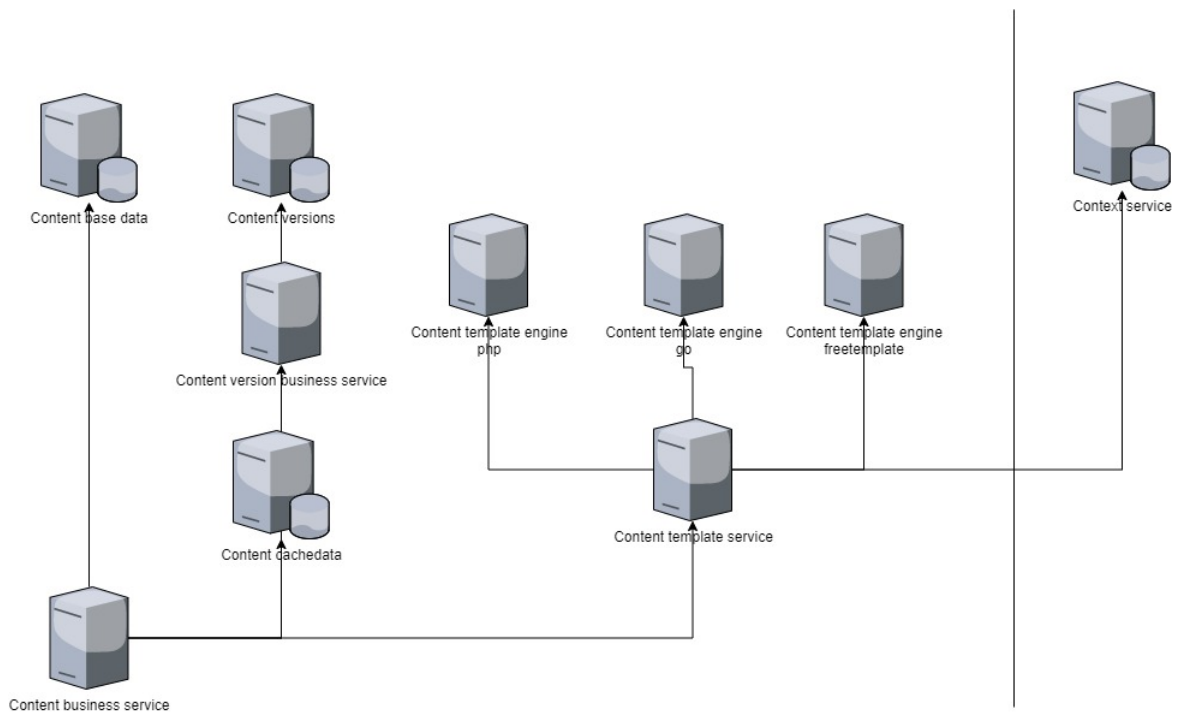
- Egyedi kontextus kezel?nek integrálhatósága (egy makró milyen adatokból dolgozhat. Rendszerhez igazított session kezelés)

Makrók, templatek el?nyei: Vannak esetek, mikor a tartalom nagy százalékban statikus, azonban vannak bizonyos elemei, akár a felhasználó neve, adott év, stb. Ezeket ne kelljen mindig módosítgatni, ezért valamilyen template leíróval kezelhet?. Azonban a tervezés során nem szeretnénk egy nyelvre korlátozni, sokkal inkább szabadon igényeknek megfelelően bővíthet?vé tervezni. ehhez kapcsolódik szorosan a kontextus kezel? amelyet a templatel összekapcsolva fog el?állni a tényleges tartalom.

3.2 A Microservicekr?l általában

3.3 Programspecifikáció





4. fejezet

Összefoglalás

Ebben a fejezetben kell összefoglalni a szakdolgozat eredményeit, sajátosságait és a témában való elhelyezkedését. A fejezet címe az „Összefoglalás” NEM módosítható! Lehet benne több alfejezet is, de nem ajánlott. Minimum 1 maximum 4 oldal a terjedelem.

Irodalomjegyzék

- [1] Andrew B. King: Website Optimization, O'Reilly Media, Inc. kiadó 2008
- [2] Nicolaos Theodossiou, Diamantis Karakatsanis, Ioannis Kougias: *The history of optimization. Applications in water resources management*, „Traditions and Culture” konferenciára készült cikk (Görögország 2014. március) 346-354.o.
- [3] optimizely.com: *Search Engine Marketing* -
<https://www.optimizely.com/optimization-glossary/search-engine-marketing> (2020.10.11)
- [4] bitacha.com (Daren Low): *6 Reasons Why Website Speed Matters & How Amazon Would Lose \$1.6 Billion If It Slowed Down*-
<https://www.bitcatcha.com/blog/6-reasons-why-website-speed-matters-how-amazon-would-lose-1-6-billion-if-it-slowed-down>
- [5] thinkwithgoogle.com: *Understanding the speed overview*
<https://www.thinkwithgoogle.com/intl/en-gb/feature/testmysite/faq/?section=2#section-2>
- [6] unbounce.com: *The page speed report*
<https://unbounce.com/page-speed-report/>
- [7] docs.microsoft.com: *A gyorsítótárzás működése*
<https://docs.microsoft.com/hu-hu/azure/cdn/cdn-how-caching-works/>
- [8] lordlikely.com: *Az Előterjesztés, Előzetes letöltés, Előcsatlakozás végrehajtása a WordPress programban*
<https://lordlikely.com/cat-wordpress/az-elterjesztes-elzetes-letoltes-elcsatlakozas/>
- [9] developers.google.com: *Introduction to HTTP/2*
<https://developers.google.com/web/fundamentals/performance/http2>
- [10] existek.com: *Top front end frameworks 2020*
<https://existek.com/blog/top-front-end-frameworks-2020>
- [11] hackr.io: *Best back end frameworks*
<https://www.keycdn.com/blog/best-backend-frameworks>
- [12] nyelvek.inf.elte.hu: *Best back end frameworks*
<http://nyelvek.inf.elte.hu/leirasok/ASP.NET/index.php>
- [13] <https://ormuco.com>: *Best back end frameworks*
<https://ormuco.com/blog/most-popular-databases>

Adathordozó használati útmutató

Ebben a fejezetben kell megadnunk, hogy a szakdolgozathoz mellékelt adathordozót (pl. CD) hogyan lehet elérni, milyen strukturát követ. Minimum 1 maximum 4 oldal a terjedelem. Lehet benne több alszakasz is. A fejezet címe nem módosítható, hasonlóan a következő részhez (Irodalomjegyzék).