
**UNIVERSITATEA SAPIENTIA DIN CLUJ-NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE,
TÎRGU-MUREȘ
PROGRAMUL DE STUDII CALCULATOARE**

Sapi3D tour – UI

PROIECT DE DIPLOMĂ

**Coordonator științific:
Ș.l.dr.ing. Szántó Zoltán**

**Absolvent:
Nagy-Serbán Tünde**

2021

UNIVERSITATEA “SAPIENTIA” din CLUJ-NAPOCA Facultatea de Științe Tehnice și Umaniste din Târgu Mureș Specializarea: <u>Calculatoarea</u>		Viza facultății:
LUCRARE DE DIPLOMĂ		
Coordonator științific: Ș.I. dr. ing. Szántó Zoltán	Candidat: Nagy-Serbán Tünde Anul absolvirii: 2021	
a) Tema lucrării de licență:		
b) Problemele principale tratate:		
c) Desene obligatorii:		
d) Softuri obligatorii:		
e) Bibliografia recomandată:		
f) Termene obligatorii de consultații: săptămânal		
g) Locul și durata practicii: Universitatea Sapientia, Facultatea de Științe Tehnice și Umaniste din Târgu Mureș Primit tema la data de: Termen de predare:		
Semnătura Director Departament	Semnătura coordonatorului	
Semnătura responsabilului programului de studiu	Semnătura candidatului	

Declarație

Subsemnata Nagy-Serbán Tünde, absolvent al specializării Calculatoarea, promoția 2021 cunoscând prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală, cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de specialitate sunt citate în mod corespunzător.

Târgu Mureș,

Data:

Extras

Extract

Cuvinte cheie:

**SAPIENTIA ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM
MAROSVÁSÁRHELYI KAR
SZÁMÍTÁSTECHNIKA SZAK**

Sapi3D tour - UI

DIPLOMADOLGOZAT

Témavezető:

**Dr. Szántó Zoltán
adjunktus**

Végzős hallgató:

Nagy-Serbán Tünde

2021

Kivonat

A dolgozat egy webalkalmazást mutat be, amely a Sapientia Erdélyi Magyar Tudományegyetem Marosvásárhely-i karának a 3D virtuális túráját foglalja magába.

A mai világban az emberek napjait nagyban befolyásolja a digitalizáció. A legtöbb embernél található legalább egy okostelefon, számítógép, laptop. Viszont nem elég, hogy a digitális eszközök nélkül az élet elképzelhetetlenné válik, mellette még ott van az Internet is.

A digitalizáció az emberek számára nagyon sok jó dolgot vezet be. Rengeteg problémát tudunk megoldani az internet és a digitális eszközök segítségével, mint például: számlák fizetése, távoli rokonokkal könnyebb a kapcsolattartás és nem utolsósorban a virtuális túrák segítségével el tudunk jutni olyan helyekre ahová nem biztos, hogy az életben lesz lehetőségünk.

A dolgozatban egy webes applikációról van szó amely felhasznál egy 3D modellt az egyetemről létrehozva az egyetem úgy nevezett virtuális túráját. A túrán bárki résztvehet ezáltal betekintést nyerhet az egyetem fala mögé. Ezen kívül informatív jelleggel is rendelkezik, mivel az alkalmazás számos információt megjelenít az egyetemmel kapcsolatban, mint például: különböző események (Sapi-Line-Tracer), szakkoordinatorok nevei, elérhetőségei.

Az alkalmazás egyik legfontosabb funkcionálitása az úgy nevezett "Vigyél el!" funkcionálitás, amely magába foglalja azt, hogy a felhasználót virtuálisan körbe viszi az egyetemen. A felhasználó megmondja hová szeretne eljutni és az alkalmazás egy adott útvonalon bemutatja az oda vezető utat. Ezen kívül arra is van lehetőség, hogy a felhasználók saját maguk lépeghessenek az egyetem modelljén így még jobban körbe tudják járni azt.

A rendszer webes felületre készült és ennek köszönhetően majdnem minden eszközön meglehetősen tekinteni, úgy a számítógépen mint a laptopen és nem utolsósorban a telefonon is. Az alkalmazás sok segítséget nyújthat az újonnan érkező egyetemistáknak, vendég diákoknak és a vendég tanároknak is az egyetem fő épületében való eligazodásnál.

Kulcsszavak: web, 3D modell, virtuális túra.

Abstract

Abstract

Keywords:

Tartalomjegyzék

1. Bevezető	1
2. Célkitűzések	4
3. Szakirodalom áttekintése	6
3.1. Telkom Egyetem kampusza 3D túrával	6
3.2. Mauritiusi Egyetemen 3D virtuális túra	7
3.3. Virtuális rendszerek az Old-Segeberg város házán	7
3.4. Kuba-i Nemzeti Művészeti iskola virtuális túrája	8
3.5. A játéktechnika alkalmazása virtuális túrák esetén	9
4. Követelmény specifikáció	10
4.1. Felhasználói követelmények	10
4.2. Rendszerkövetelmények	12
4.2.1. Fejlesztési rendszerkövetelmények	12
4.2.2. Használati rendszerkövetelmények	14
4.3. Funkcionalitások	14
5. Tervezés	16
5.1. Az egész rendszer architektúra	16
5.2. Webalkalmazás architektúra	17
6. Technológiai áttekintés	19

6.1. Adatbázisok	19
6.1.1. SQL	19
6.1.2. NoSQL	20
6.1.3. SQL vs NoSQL	20
6.2. Webes keretrendszerek	23
6.2.1. Angular	24
6.2.2. Vue.js	24
6.2.3. Angular vs Vue.js	25
6.2.4. NodeJs	25
6.2.5. Spring Boot	26
6.2.6. Spring Boot vs NodeJs	27
7. Használt technológiák	29
8. Felhasználói felület bemutatása	30
8.1. Megvalósítások	30
9. Összefoglalás	31
10. Továbbfejlesztési lehetőségek	32
Irodalomjegyzék	32

Ábrák jegyzéke

4.1. A rendszer használati eset diagramja	11
5.1. Az egész rendszer architektúrája	16
5.2. A webalkalmazás rendszer architektúrája	18
6.1. 6.000.000 rekord, 50% olvasás, 50% írás esetén	21
6.2. 6.000.000 rekord és 5.000 véletlenszerű olvasás esetén	22
6.3. 6.000.000 rekord 5.000 véletlenszerű frissítés esetén	22
6.4. Webes keretrendszerek Github és Stack Overflow pontozások alapján	24

Táblázatok jegyzéke

6.1. Előnyök Angular és Vue.js között	25
6.2. Hátrányok Angular és Vue.js között	25
6.3. Előnyök Spring Boot és NodeJS között	27

1. fejezet

Bevezető

A mai gyorsan fejlődő világunkban a digitális eszközök a mindennapok elengedhetetlen részei. Nem sok olyan háztartás van ahol nincs egyáltalán legalább egy telefon, számítógép, laptop, okos tv. Az elmúlt években a digitális világ annyira fejlett lett, hogy lassan már ki sem kell mozdulnunk a házból és mindent eltudunk végezni. Például ki tudjuk fizetni a számláinkat, be tudunk vásárolni. Ezekből is következtethetünk, arra hogy egy jó internet kapcsolattal és egy közepes teljesítményű számítógép mellett már egy életet is letudunk élni.

Az elmúlt két évtizedben a „virtuális múzeum” [1] fogalmának meghatározása a gyors technológiai fejlődés következtében megváltozott. A mai rendelkezésre álló 3D technológiák a virtuális múzeumok esetén már nem csupán a gyűjtemények bemutatása az interneten vagy egy kiállítás virtuális bemutatója panorámás fotózás segítségével. Egy virtuális múzeum nem csak az oda látogatóknak hasznos, hanem már tanórákon is feltudják használni a tanárok mivel nem mindig sikerül elvinni a diákokat egy adott városba így egy ilyen virtuális túra segítségével a tanár betudja mutatni az adott múzeum látványosságait. Sőt a gyerekek ezáltal otthon is eltudnak barangolni más országok, kontinensek múzeumaiba virtuálisan.

Világunk fejlődése próbálja biztosítani, hogy egyetlen ember se maradjon le azon helyekről ahová nem tud eljutni, így egyes múzeumokat, iskolákat, kastélyokat, látványosságokat is betudunk járni otthon a négy fal között a 3D virtuális túrák segítségével. De mit is jelent, mire is használják ezeket a 3D virtuális túra modelleket?

Több okot is fel lehet sorolni annak érdekében, hogy miért is használatosak ezek a 3D modellek-

kel megalkotott virtuális túrák. Első sorban az új diákok már otthonról be tudnak nézni az egyetem falai mögé így amikor elérkeznek az új év kezdéséhez akkor otthonosabban érezhetik magukat. Ez azért történhet meg mert már fogják tudni, hogy mit hol találnak nem kell segítséget kérjenek. Egy másik ok, hogyha az adott egyetem rendelkezik egy ilyen típusú modellel akkor jobban felhívhatja a figyelmet a diákok számára. Ez által az egyetem népszerűsítése is megtörténik. Ismert, hogy egyes szülők nehezen engedik el gyermeküket egyetemre mert féltik. Viszont az, hogy látnak egy képet az egyetemről megnyugtathatja őket így bátrabban biztathatják gyermeküket, hogy menjen el az adott egyetemre.

A 3D jelentése három dimenzió. Ez egy modell, amely tulajdonképpen matematikailag ábrázol egy háromdimenziós objektumot, amely lehet épület, virág, vagy akár egy ember is. A 3D modelleket széles körben használják az orvostudományban, magasabb szintű gyakorlati és elméleti kompetenciák elérésében. A 3D modellekkel nem csak tárgyakat hanem emberi folyamatokat is le lehet szimulálni.

Az orvos tudományban nagyon szépen letudják előre játszani a műtéteket így biztosabbak a dolgukba és a kockázati lehetőségeket is csökkenthetik. Sok esetben egy szimuláció/3D modell segít egy adott problémát jobban átlátni. Ilyen példa lehet egy hajóforgalom szimulációs rendszer [2] amely figyelembe veszi a hajókat, vízfelületet és az időjárási viszonyokat. Ezzel a rendszerrel próbálták megmutatni az olajszennyezést a tengerekben, óceánokban. Ezen modellek segítségével a világ látványosságai elérhetővé válhatnak azon emberek számára is, akik nem jutnak el az eredeti országba, városba, hogy megtudják tekinteni az adott látványosságot.

A számítógépes grafikát [3] sok területen alkalmazzák mint például az interaktív médiatervezésben, 3D túrák készítésében és videojáték iparban. Az elmúlt néhány év során számos technológia jelent meg a 3D virtuális túrák megjelenítésére az interneten. A virtuális túrák a felhasználóknak biztosítják az életszerű 3D környezeteket.

A tour (virtuális túra) szó jelentése utazás, kirándulás. A mai világban túrázni nem csak a való életben lehet hanem a virtuális világban is. Egy ilyen virtuális túra célja, hogy fejlettebb szimulációs technikák segítségével a nézők élethű 3D-s képet kapjanak a meglévő helyről. Egy híres példa a Second-Life [3] ahol a felhasználók közötti interakció avatárokon keresztül zajlik. Ezen alkalmazáson belül interaktív 3D tour-ok vannak leképezve.

A 3D és a tour (virtuális túra) szavak összetételéből jön ki a 3D tour (3D virtuális túra) amely azt

tükrözi, hogy a virtuális világban tudunk megnézni adott épületeket, kilátásokat, látványosságokat. Habár ezen virtuális 3D modelleken alapuló világ még nem tökéletes de folyamatosan fejlődik. A dolgozatomon belül egy ilyen modellről lesz szó amely a Sapientia Erdélyi Magyar Tudományegyetem Marosvásárhely-i karának egy részét tartalmazza. A projekt két részre van bontva, amely két államvizsga dolgozatot eredményez a 3D modell és a felhasználói felület. Közös munka a modellel történő attrakciók megoldása. Ilyen attrakció lehet a modellben való mozgás, közlekedés. Ezen dolgozatban a felhasználói felületről lesz szó.

A projekt célja, hogy a felhasználók a saját házukból is betekintést tudjanak nyerni az egyetem falai mögé is. Ez elsősorban az új felvételizőknek és az első éves egyetemistáknak lenne hasznos, mivel ez által már otthon neki foghatnak áttekinteni az egyetem különböző részeit mint például, hogy hol található egy adott tanszék, vagy hol található a dékáni hivatal, vagy hogy hol található a könyvtár. Ilyen alkalmazás hasznos lehet úgy a diákok mint a tanárok, szakkoordinátorok számára, hiszen rengeteg apró de gyakori kérdésben tud segítséget nyújtani. Ugyanakkor egy egyetemen belül sokszor megfordulnak kutatók, vendég tanárok, külföldi diákok (Erasmus) így számukra is hasznos lehet egy ilyen jellegű alkalmazás.

Egy ilyen alkalmazás esetében nem csak maga a modell jelenik meg hanem rajta kívül számos fontos információ, elérhetőség is. Dolgozatom célja bemutatni a Sapientia Erdélyi Magyar Tudományegyetem 3D Virtual Tour alkalmazás megalkotását, implementálását és nem utolsósorban a felhasznált technológiákat

2. fejezet

Célkitűzések

Elsődleges cél az volt, hogy legyen egy felület ahol a Sapientia Erdélyi Magyar Tudományegyetem Marosvásárhely-i karának a 3D modelljét megtudjuk jeleníteni. Ez mellett fontos volt az is, hogy az alkalmazás ne legyen bonyolult. Legyen minél egyszerűbb, átláthatóbb, használhatóbb. Hiszem azt, hogy egy egyszerűbb rendszer használatosabb mint egy bonyolultabb.

Elgondolkodtam azon is, hogy szükséges-e regisztráció majd bejelentkezés minden felhasználó részére. Idővel rájöttem, hogy ez nem szükséges hiszen ez az alkalmazás mindenki számára nyitott kell legyen, mivel az a cél, hogy megmutassuk az egyetemet belülről. Így a regisztrációs ötlet részt teljesen elvetettem.

Habár a regisztrációs részt teljesen elvettem eszembe jutott, hogy az alkalmazáson néha kell frissíteni ezért kellene egyedi felhasználók is amelyek elérik az alkalmazás azon részeit amit más nem. Így jött az ötlet, hogy csak kimondottan bejelentkezés lesz és az új user-eket egy adott státusszal rendelkező user felhasználó tud hozzá rendelni. Ez azt jelenti, hogy a visitorból(látogató) lesz egyedi felhasználó. Viszont ez a lehetőség nem mindenki számára biztosított.

Az egyedi felhasználók szempontjában is a legnagyobb cél az átláthatóság, egyszerűség, könnyen kezelhetőség. Ennek érdekében az a cél, hogy minden egyedi felhasználó számára csak az elérhető módosítási lehetőségek jelenjenek meg. Ezen azt kell érteni, hogy egy admin jogosultsággal rendelkező személy tud hozzáadni új egyedi felhasználót vagy akár törölni is, míg aki csak felhasználó jogosultsággal rendelkezik nem.

A bejelentkező személyek számára biztosítani szeretném, a felhasználói adatok biztonságos eltá-

rolását és kezelését is.

Cél az is, hogy az egyetemről egy link gyűjtemény kerüljön be az alkalmazásba. Ezt úgy kell érteni, hogy az adott tanszékekről, szakokról egy bővebb leírást mutatni, úgy hogy a linkeken keresztül átkerülünk olyan oldalakra ahol megjelennek bővebb információk az adott dologról. Ezáltal a diákok jobban eltudják dönteni, hogy az a szak amelyet kiválasztanak mennyire lesz jó számukra.

Mivel az alkalmazás első éves diákok számára készül és ők még nem ismerik az egyetem keretein belül szervezendő eseményeket sem, így az is a célok közé tartozik, hogy egy esemény naptárral is bővüljön az alkalmazás. Így az új diákok tisztában lehetnek, hogy milyen események lesznek, fogják tudni a helyszínt és dátumot is.

Az alkalmazás egyik legfőbb célja megvalósítani egy olyan funkciót, hogy "Vigyél el!". Ez a funkcionalitás azt jelenti, hogy az új egyetemista diák, vendég tanár, vendég hallgató vagy akár vendég kutató bejelöl egy adott helyiséget ahová szeretne eljutni az egyetem területén és az alkalmazás a modell segítségével megmutatja az oda vezető utat.

Szeretnék, egy olyan részt is biztosítani minden felhasználó számára ahol a saját véleményét tudja kifejezni. Ezen véleményeket figyelembe véve szeretném kijavítani az észlelt hibákat.

3. fejezet

Szakirodalom áttekintése

A világhalón való keresés során sok tanulmányt találtam, amely leírja hogy egy virtuális túra egyetemen, múzeumokban, különböző látványosságoknál mekkora befolyásoló képességgel rendelkezik. Megtudhattam, hogy sok egyetemnek van ilyen jellegű túrája más más típusban. Van amelyik egyetem a virtuális túrát képek sorozatában képzelte el, van amelyik panoráma képekben, van amelyik videóban és nem utolsó sorban jönnek azok akik 3D modellekkel valósították meg az egyetemük virtuális túráját. A következő alfejezetekben bemutatok pár hasonló témájú alkalmazást, megközelítést.

3.1. Telkom Egyetem kampusza 3D túrával

Indonézia egyik legnagyobb magánegyeteme a *Telkom University*¹ [4] is a virtuális 3D túrát alkalmazza az új diákok oda vonzására. A túrák tartalmazznak videó és képsorozatokat plusz 3d alapú modelleket is. Tulajdonképpen egy 3D web alapú túrát dolgoztak ki a 3D Vista használatával.

Miközben fejlesztették, ezt a túrát kutatásokat végeztek, hogy mivel lenne jó elkészíteni, más egyetemen milyen technológiákat alkalmaztak hazai területeken. A következő eredmények születtek:

- RF Rahmat: épületekről információ szolgáltatásokat kiviteleztek Universitas Sumatera Utara (USU) környezetben.
- Moloo: virtuális túrát hoz létre a WebGL és a SketchUp segítségével.

¹<https://tour.telkomuniversity.ac.id/>

- Fujita: Mobile robotokat használ virtuális túrák elkészítéséhez.

Szeliski R. aki a fotó technikákat mutatta be, szerint is a nagy felbontású fotók és a 3D modellek együttes használatával nagyobb érdeklődési kört lehet elérni, mint ha csak külön használnák őket. Az egyetem kiemelt kutatási stratégia terve volt, hogy ne csak a diákokat vonzzák magukhoz, hanem az IKT (Információs és kommunikációs technológia) technológiák fejlesztésében is érjenek el fejlődéseket. Ennek érdekében használtak 3D modelleket és nagy felbontású fotókat együttesen. A diákok számára nagy érdekességnek számított és ezzel elérték azt, hogy a diákok érdeklődését felkeltették.

3.2. Mauritiusi Egyetemen 3D virtuális túra

*Mauritius*²-i egyetem [3] is megalkotta saját 3D virtuális túráját. A megalkotást WebGL segítségével történt. A WebGL (Web Graphics Library) könyvtár 3D-s valós idejű megjelenítést kínál. A WebGL az OpenGL-ből származik, és API-t biztosít a 3D grafikához. A WebGL nem teljesen stabil és néhány algoritmus nem hatékonyan látja el a feladatait. Különböző böngészők memóriahasználata és végrehajtási ideje eltérő amelyek mellékhatásokat idézhetnek elő a megírt alkalmazásban.

Az egyetem is szintén valós időbeni megjelenítést alkalmaz és a megfelelő működés érdekében tesztelték az alkalmazást teljesítmény szempontjából, amikor több felhasználó csatlakozik egyidejűleg. A teszt eredményiből az egyetem arra következtetett, hogy minél összetettebb az objektum és ha még textúrával is rendelkezik akkor a teljesítmény nagyon csökken mivel a modell így nagyon nagy erőforrást igényel.

Az egyetem a megvalósításnak az egyik legegyszerűbb formáját választotta amellyel sikeresen elérték céljukat. Megalkották a virtuális túrájukat és ezt felhasználva felkeltették az jövőbeli egyetemisták érdeklődését.

3.3. Virtuális rendszerek az Old-Segeberg város házán

A Németország-i Old-Segeberg [1] város házának is készítettek ilyen virtuális túrát egy Windows alapú interaktív szoftvert és egy virtuális valóság alkalmazást HTC Vive rendszerekkel. Mindkét rend-

²<https://www.middlesex.mu/about-mdx-mauritius/campus-virtual-tour>

szert kipróbálták a látogatók és a visszajelzések alapján van jövője az ilyen jellegű alkalmazásoknak is. A virtuális múzeum csúcspontjai a műalkotások pontos ábrázolása.

A Windows alapú interaktív szoftvert egy PC számítógép segítségével tekinthették meg a felhasználók. A lényege az volt, hogy az oda látogató nem lépett be teljesen a virtuális világba csak kívülről tekintett bele míg a HTC Vive applikáció segítségével már a felhasználó végig ment a város házán virtuálisan. HTC Vive applikáció több interaktivitást nyújt a felhasználóknak viszont mindkét kifejlesztett rendszernek megvannak az előnyei és a hátrányai is. Majdnem minden háznál található egy PC számítógép így a Windows alapú rendszert könnyebben népszerűbbé lehet tenni mint a HTC Vive-ot.

3.4. Kuba-i Nemzeti Művészeti iskola virtuális túrája

Az innovatív technológiák új lehetőségeket biztosítottak a kulturális helyszínekről szóló információk gyűjtésére, elemzésére és megosztására. E technológiák közül a gömbszerű képalkotás és a virtuális túrakörnyezet segítségével megalkották a Kuba-i Nemzeti Művészeti iskolát [5] is, pontosabban a Nemzeti Balettiskolát. A virtuális túrán megnézhetőek a tantermek, a fő kupola és az előadó terem. A túrát azért hozták létre mivel az iskola már nagyon régi és rossz állapotban van így ezzel megtudják mutatni az érdeklődőknek, hogy milyen is volt régen az épület.

Ez a virtuális túra kompatibilis számítógépekkel, táblagépekkel és egyéb mobil eszközökkel. A virtuális túrában be vannak építve pdf-ek, linkek amelyek az adott részről bővebb információkat szolgáltatnak a túrázóknak. Összesen 14 gömbpanorámát, 96 hotspotot (műveletek pl. előre lépés, vissza lépés) és 40 linket. Ezen eszközök segítségével biztosítanak a felhasználóknak kényelmes virtuális túrázási lehetőséget. Az képernyőn megjelenik egy menü rendszer is amellyel tudjuk irányítani a túrát.

Ezen tanulmány szerint ennek a túranak a célja az volt, hogy az Balettiskola fennmaradjon a következő nemzedékek számára is legalább virtuális közegbe.

3.5. A játéktechnika alkalmazása virtuális túrák esetén

A virtuális túrák információt nyújtanak multimédiás úton a felhasználóknak, azt a benyomást keltve hogy valós időben navigálnak különböző helyeken. Egy sikeres túra jelentése, hogy a felhasználó el tudja hinni, hogy ő habár virtuálisan is de járt abban az adott helyiségben. Ahhoz hogy ez a benyomás létre jöjjön a modell pontos ábrázolást kell tartalmazzon az adott helyiségről. Ezen tanulmány szerint is az ilyen túrák felhasználhatóak létesítmények népszerűsítésére mivel egy interaktív élményt biztosítanak. Az ilyen jellegű túrákat össze lehet hasonlítani a számítógépes játékokkal is. Hiszen ezek a játékok annyiban különböznek, hogy nem valós időben és nem valós helyeken történnek. Persze vannak kivételek, ahol a játék egy része magába foglal egy valós helyszínt is.

Az Egyesült Királyság [6] számos egyeteme használ ilyen jellegű túrákat annak érdekében, hogy az emberek távolról is betekintést tudjanak nyerni az egyetemek világába. A kutatás szerint tizennégy brit egyetemet tekintve mindegyik weboldalán találtak állóképgalériát és videogalériát viszont meglepő számnak számított hogy 3600 interaktív túrát is találtak. Azért meglepő szám mivel ezek a túrák még nem igazán elterjedtek.

Ezen tanulmány felmérte az Egyesült Királyság egyetemei körében mennyire használatosak a virtuális túrák. Kiderült, hogy a virtuális egyetemi túrák interaktívabbá tették az egyetemek weboldalait és nagyobb fokú szolgáltatást biztosítottak a felhasználók számára hiszen nem csak képeket, videókat láttak az egyetemekről, hanem maga az egyetemet is. Az egyetemek ezen túrák megalkotásában teljes mértékben kihasználták a számítógépes játékokhoz használt grafikai eszközöket. Az modellek segítenek bemutatni az egyetemeket, plusz útvonalakat is tervezhetnek a modellben így már előre fogják tudni, hogy mit hol találnak a felhasználók. A jövőben valószínűleg majdnem minden egyetem fogja alkalmazni az ilyen jellegű túrákat.

4. fejezet

Követelmény specifikáció

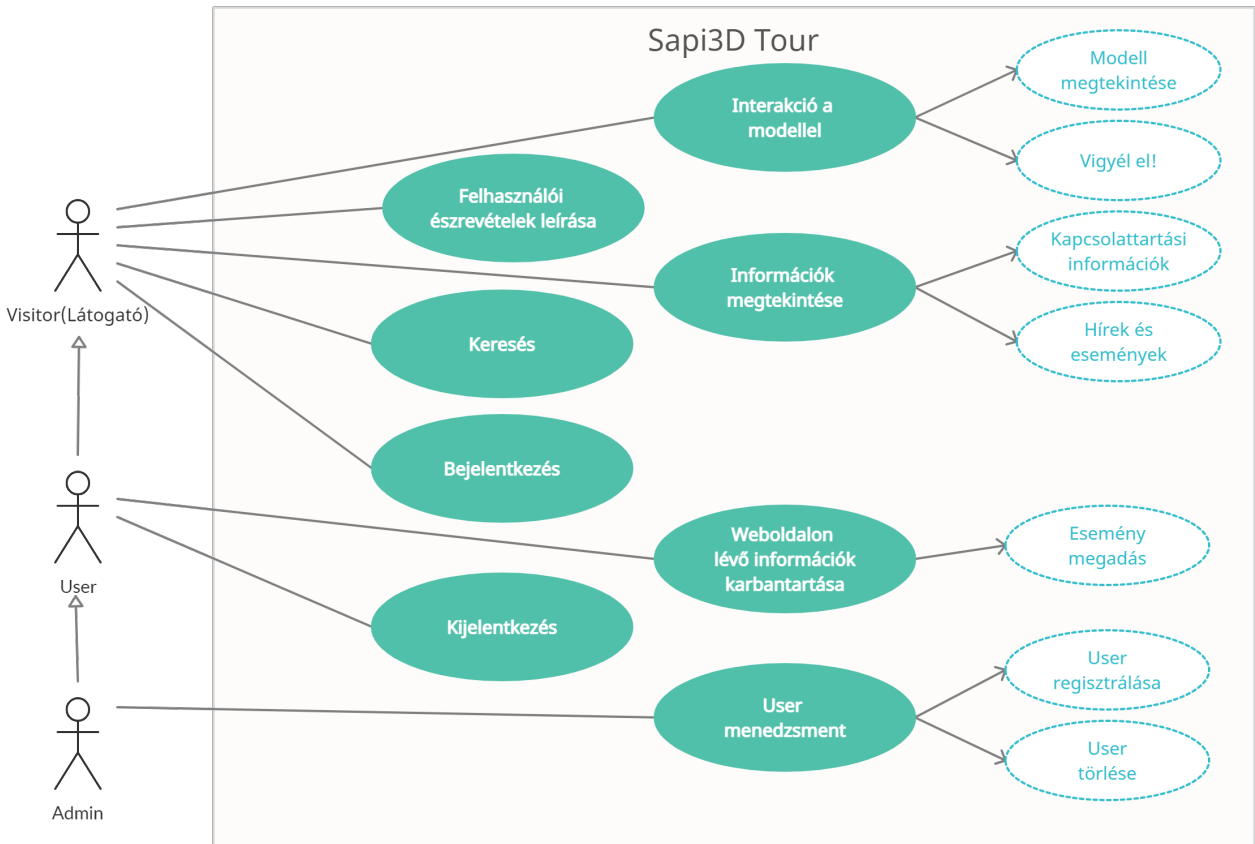
4.1. Felhasználói követelmények

A Sapi3D alkalmazás web alapú, ezért mindenki számára elérhető a fő célja, hogy egy 3D modellként jelenítse meg a Sapientia EMTE Marosvásárhely-i karának a fő épületét, illetve ennek fontosabb helyeit, mint pl. tanszékek, titkárság stb. A rendszer fontosabb funkcionalitásait és az ezeket igénybe vevő szerepköröket a 4.1 ábra szemlélteti.

Az elkészített alkalmazást bárki eltudja érni bárhol. Egyetlen feltételnek kell eleget tenni, amely az internet kapcsolat megvalósítása lenne. A rendszer megértésének érdekében tekintsük meg a 4.1 ábrát amely bemutatja a rendszert, amit három különböző felhasználó vehet igénybe. A következő felhasználói szerepkörök vannak: VISITOR, USER és ADMIN. A három típusú felhasználó közül kettőnek van lehetősége bejelentkezésre (ADMIN, USER).

Az első szerepkör a VISITOR(látogató), amelynek lehetősége van megtekinteni az elkészített weboldalt, tudja használni a "vigyél el" opciót és nem utolsó sorban saját kezűleg is végig tud menni az egyetem 3D modelljén. Vannak olyan VISITOR-ok aki betudnak jelentkezni így átalakulnak USER-ré. A második szerepkör a USER, aki felelős az alkalmazás karbantartásáért is. A harmadik szerepkör az ADMIN. Az admin felhasználó akinek az egész rendszerben van a legnagyobb felelőssége. Ő felel azért, hogy mely visitorok kaphatnak engedélyt a bejelentkezéshez. Ez mellett az ő hatáskörébe tartozik, hogy ki lesz kitörölve a rendszerből. Mindezek mellett az admin joggal rendelkező felhasználó is felelős az alkalmazás karbantartásában. A karbantartás alatt kell érteni azt, hogy az oldalon megjelenő

információk napra készek legyenek.



4.1. ábra. A rendszer használati eset diagramja

Bármely felhasználó, aki egy böngészőből megnyitja az oldalt, a VISITOR kategóriába kerül. Ez a fajta felhasználó megtekintheti a 3d modellt, körbe sétálhatja és el tud jutni pl. titkárságra. Amint megnyílt az oldal rögtön látható az egyetemről készített modell. Ezen a modellen tud nézelődni, esetleg körbe is tudja járni, vagy adott helységekre el is tud jutni (például: titkárság, adott tanszék). Ezen kívül lehetősége van információk, elérhetőségek, események részleteinek elolvasására is. Minden VISITOR ugyan akkor leírhatja saját véleményét, meglátásait az oldallal kapcsolatban is. A fent említett műveletek elvégzéséhez nem kell sem bejelentkezés, sem regisztráció.

Amennyiben a VISITOR bejelentkezik átkerül a USER kategóriába. A bejelentkezéshez szükséges megadni egy már regisztrált e-mail címet és egy már hitelesített jelszót is. A USER engedélyezése nem regisztráció alapján történik, hanem az admin joggal rendelkezők osztják ki, mivel a rendszer

úgy van megtervezve, hogy nincs direkt regisztráció, hanem csak egy ADMIN tudja beregisztálni az új USER-eket. Egy USER képes különböző műveletek elvégzésére, mint például: eseményeket megadni, az egyetemmel kapcsolatos információkat módosítani. Ezeken kívül mivel a VISITOR-ból lesz USER, így a USER-ből is tud lenni VISITOR a kijelentkezési opció esetén.

A regisztrációs feladatot egy admin jogosultsággal rendelkező felhasználó végezheti el. Nem csak a regisztrálás tartozik ehhez a feladatkörhöz, hanem a userek törlése is az ő feladata. Ezen plusz műveletek mellett szintén elvégezheti a USER és a VISITOR műveleteit is.

4.2. Rendszerkövetelmények

4.2.1. Fejlesztési rendszerkövetelmények

A rendszer egy webalkalmazásként legyen fejlesztve. Az alkalmazás fejlesztéséhez első sorban szükséges legalább Windows 10 operációs rendszer, vagy bármely operációs rendszer amely támogatja a Vue.js-t, Spring Boot-t, MongoDB-t.

Frontend

A Frontend részt HTML/JavaScript-ben kell megvalósítani, keretrendszernek pedig a Vue.js-t kell használni, mivel egy új, progresszív keretrendszer. A Vue.js használatához a következő utasításokat kell végrehajtani:

- nodejs és npm letöltése és installálása
- vue init webpack sapi3dtour
- npm install vue-router
- vue add vuetify - prototype
- npm install @fortawesome/fontawesome-free -D
- npm install --save axios

- `npm install vue-3d-model --save`

A Frontend a következő parancssor segítségével legyen indítható: `npm run dev`.

Backend

A Backend rész Java-ba legyen írva, keretrendszernek legyen használva a Spring Boot, mivel a segítségével nem kell külön foglalkozni a Maven és Gradle fájlok, Tomcat szinkronizálásával. Egy Java aplikáció a Spring Booton belüli létrehozásához a következő parancsokat kell teljesíteni:

- A legújabb Java (15.0.1) instalálása az operációs rendszerre
- Spring Boot keretrendszer telepítése az operációs rendszerre
- *Spring Initializr*¹ segítségével létre hozni a projektet különböző dependenciák megadásával.
- A következő dependenciák legyenek benne kezdetlegesen: MongoDB, Spring Security, Spring Web, Validation, Java Mail Sender
- Fejlesztés során ezeket a dependenciákat lehet bővíteni

Adatbázis

Az adatbázis legyen MongoDB NoSql adatbázis, amelyet az operációs rendszernek megfelelő Workbench-ben kezeljünk. Mivel az adatokat nem lehet egy séma alapján felállítani ezért van szükség a NoSQL adatbázisra.

Az adatbázis létrehozásához a következő lépéseket kell betartani:

- Telepíteni a MongoDB adatbázis kezelő rendszert.
- Telepíteni a MongoDB-hez tartozó Workbench-et
- Létrehozni az adatbázist a Workbench segítségével
- Kollektiókat nem itt hozzuk létre. Az a Spring Boot feladata.

¹<https://start.spring.io/>

4.2.2. Használati rendszerkövetelmények

A rendszer használatához a felhasználóknak szüksége van Internetre és egy digitális eszközre amelyen megtudják nyitni a webalkalmazást. A digitális eszköz lehet laptop, asztaligép, tablet vagy akár telefon is.

A felhasználók nincsennek egy adott operációs rendszerhez kötve. Az operációs rendszer lehet Windows, Linux, Android és iOS is. Külön telefonos applikáció még nincs, viszont a telefonok található böngészők bármelyikében meglehetősen tekinteni az alkalmazást.

4.3. Funkcionalitások

Alkalmazásomon belül több fajta funkció jelenik meg. Első sorban nincs regisztrációhoz kötve a felhasználó. Ez azt akarja jelenteni, hogy mindenki előtt nyitva áll az alkalmazás. Viszont vannak különleges joggal rendelkező felhasználók, akik be tudnak jelentkezni és ennek következtében több funkció bocsájtja a rendszer a rendelkezésükre.

A USER hozzáadás az ADMIN-on keresztül úgy történik, hogy megad minden adatot (név, e-mail cím, telefonszám) és egy gombra kattintva az új adatok bekerülnek az adatbázis rendszerbe. Közben az új felhasználó fog kapni egy email-t amellyel el tudja fogadni a jelentkezését. Ügyelni kell, hogy az email érvényessége időhöz kötött. A törlés csak egyszerűen kiválasztással történik. Egyetlen felhasználó sem tudja törölni saját magát még az admin joggal rendelkezők sem.

A következőkben a user joggal rendelkező felhasználók lehetőségeit részletezem. Eznek a szerepkörnek lehetősége van hozzáadni eseményeket, órarend linkeket. A tanszékek, szakok leírásának megadásai is lehetővé válik számukra. Sőt ha netán új szak indul azt is hozzá tudják adni.

A USER-ek számára természetesen van kijelentkező opció is. Tudni kell, hogy a modellnél, tanszékeknél, szakoknál és más fontos információknál elérhetőségeknél mindig az adatbázisban szereplő legújabb adat jelenik meg.

A következőkben tárgyalom a VISITOR felhasználó által elérhető funkciókat. Az első és legfontosabb, hogy megjelenik egy 3D modell az egyetem épületéről amelyen a felhasználók tudnak nézelődni esetlegesen körbe is tudják járni.

Mindenki számára megjelenik a bejelentkezési lehetőség is. Viszont ezt a funkciót csak azok használhatják akik rendelkeznek felhasználónévvel és jelszóval.

A második legfontosabb funkció a tanszékek, szakok megjelenítése. Ez egy külön oldalon lesz. A felhasználók itt láthatnak egy leírást és az elérhetőségeket egy adott tanszékről. Ezek mellett egy útvonalat is megtekinthetnek a 3D modellen belül így amikor oda érkeznek az egyetemre nem kell annyira keresgélni, hogy mi hol található.

Mindenki számára elérhető lesz az esemény naptár is. Ezen részen jelennek meg azok az események amelyeket az egyetem szervez a diákok részére. Látható lesz az események pontos dátuma, helyszíne, ára és itt is megjelenik egy olyan opció is amely a 3D modellen megmutatja az útvonalat. Ez azért jó mert ha egy idegen ember érkezik egy ilyen eseményre, például versenyre, akkor könnyebben elfog tudni igazodni, hogy hová is kell menjen.

Utolsó funkcionalitásnak egy vélemény nyilvánítást tettem be. Itt minden felhasználó, név nélkül tudja közölni az alkalmazással kapcsolatos észrevételeit. Ha valaki véleményt szeretne írni akkor értékelnie is kell egy skálán, hogy szerinte mennyire hasznos az alkalmazás. A vélemények lehetnek pozitívak és negatívak is mindezek mellett új ötlet javaslatokat is lehet írni. Figyelembe véve a véleményeket lehetőség van jobbra fejleszteni az alkalmazást. A felhasználók nem csak véleményt tudnak nyilvánítani, hanem a mások által adottakat el is tudják olvasni. A vélemények szerkesztésére nem lesz lehetőség.

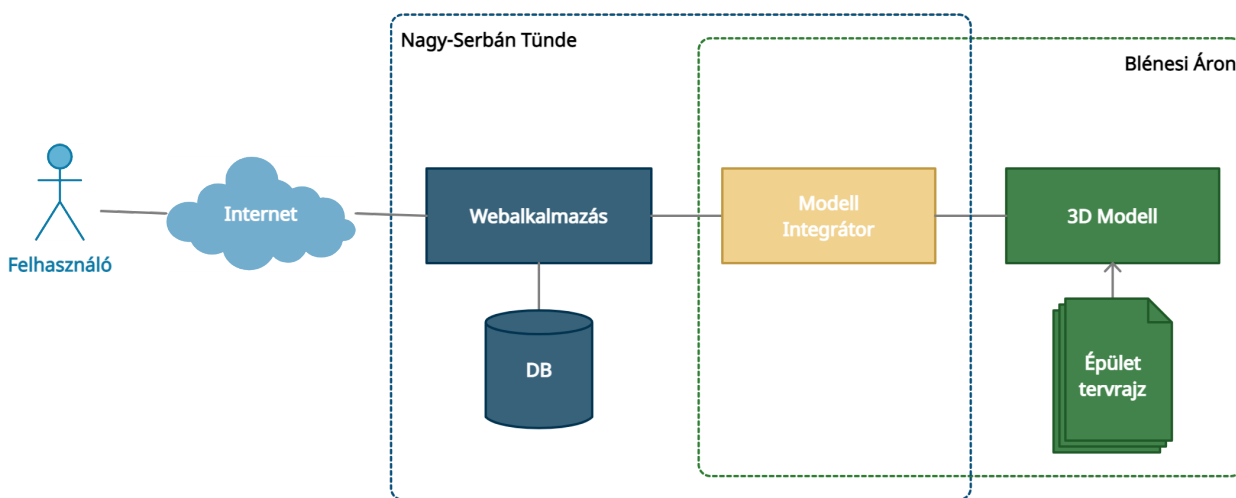
A funkcionalitásokból is látszik, hogy a céloknak megfelelően próbáltam megfelelni. Törekedtem az egyszerűsége, átláthatóságra. Kiderült, hogy a regisztrációs rész nem egyedi módon történik meg. Az alkalmazás megpróbál minden olyan lehetőséget magába foglalni ami a diákok számára elérhető kell legyen.

5. fejezet

Tervezés

5.1. Az egész rendszer architektúra

Az egész rendszer tervezésénél első lépésben összeállítottunk rendszerünk felépítéséről egy olyan listát ahol megjelennek a fő komponenseink. Az 5.1 ábrán látható ezeknek a komponenseknek az összesége. Megfigyelhetjük az ábrán, hogy nem a csak az elemek vannak rajta feltüntetve, hanem az elemek közötti kapcsolatok is megjelennek.



5.1. ábra. Az egész rendszer architektúrája

Amiután megszületett ez a lista rájöttünk, hogy a rendszert két részre fogjuk osztani. A 3D modell

tervezését, implementálását a kollégám, Blénesi Áron készíti el míg az webalkalmazást, adatbázis tervezést én. A két a projektet egy komponensen keresztül kötjük össze. Ez a komponens a web alkalmazáson belül van létrehozva, és általa jelenítődik meg a 3D modell.

Az 5.1 ábrán látható színeknek is megvan a maga jelentősége. A világos kék színnel látható a Felhasználó és az Internet. Ez a két komponens biztosítja a rendszer elérhetőségét és felhasználását, mivel az interneten keresztül érik el majd a felhasználók a weboldalt, ami által használatba tudják venni a rendszert.

A sötét kék színnel jeleltük a Webalkalmazást és az adatbázist (DB). Ezen komponensekről a további fejezetekben lesz szó bővebben. A sötét zöld szín a 3D modell felépítésére vonatkozik, amelyről egy másik dolgozat keretén belül lehet olvasni. (Blénesi Áron: Sapi3D tour - 3D modell). A narancs-sárga színnel jelölt rész lenne a két alrendszer összekötésének komponense, amelyről részletesebben lehet olvasni ezen dolgozat további fejezeteiben és a már említett kolléga dolgozatában is.

Az egész rendszer tervezésének befejezésekor kezdődött el az alrendszerek tervezésének fázisai. A Webalkalmazás és Adatbázis tervezéséről 5.2 alfejezetben olvashatunk.

5.2. Webalkalmazás architektúra

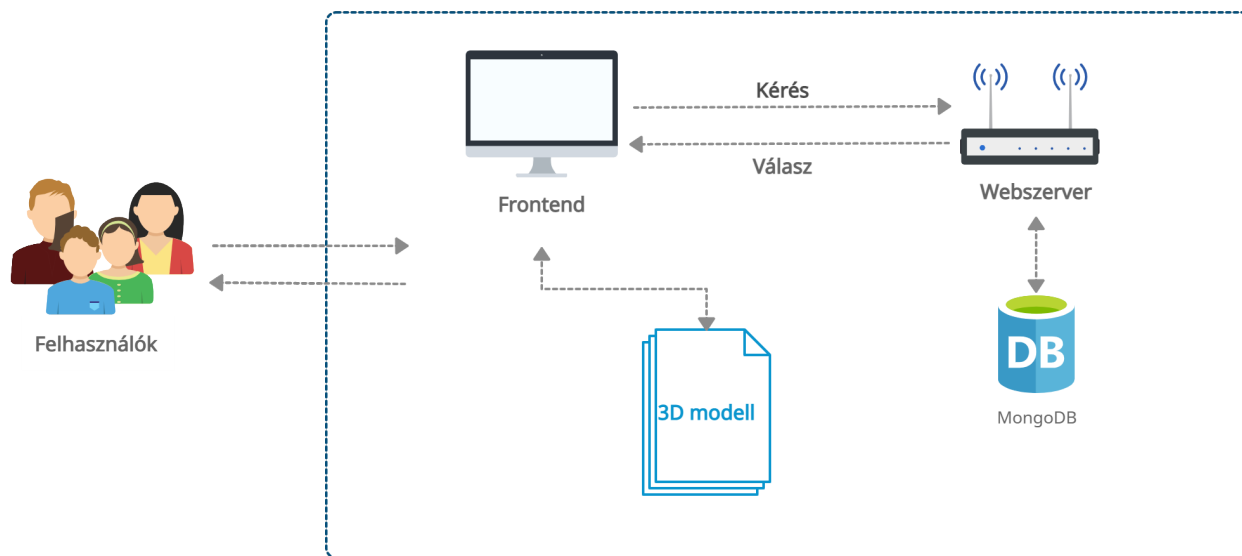
Az egész rendszer megtervezése után szüksége volt a felosztott részeknek külön alrendszer tervezésére, amely a 5.2 ábrán jelenik meg. A tervezés során kideült, hogy négy nagy komponensre lesz szükség: Frontend, Webszerver, Adatbázis és a 3D modell.

Amint az ábrán is látszik a Felhasználók a Frontenden keresztül érik el a teljes alkalmazást. A Frontend fogja biztosítani a felhasználói felületet, vagyis itt fognak megjelenni az adatok az adatbázisból és a 3D modell is ide fog betöltődni.

A Webszerver segítségével lesznek elérhetőek a Frontend számára az adatbázisban tárolt adatok. Tulajdonképpen a Webszerver biztosít egy kommunikációs csatornát a Frontend és az adatbázis között. A Frontend kéréseket (GET, PUT, POST) küld a webszervernek és a webszerver a kérésnek megfelelő eredményt szolgáltatja vissza.

Az adatbázis fogja tárolni a Frontendről érkező adatokat, valamint vissza is szolgáltatja azokat a megfelelő kérések esetén a Webszerveren keresztül.

A 5.1 fejezetben említett 3D modell egy statikus fájlban lesz elhelyezve, amelyet a Frontend ér el és onnan fogja betölteni a modellt egy Frontenden belüli komponensbe. Ezen a komponensen keresztül fog összekapcsolódni a 5.1 fejezetben említett két alrészleg, maga a webapplikáció és a 3D modell.



5.2. ábra. A webalkalmazás rendszer architektúrája

6. fejezet

Technológiai áttekintés

6.1. Adatbázisok

Adatbázisnak nevezzük azokat a nagy mennyiségű adatokat amelyek közös jellemzőkkel és struktúrákkal rendelkeznek. Az adatbázisokon belül több fajta műveletet tudunk elvégezni: karbantartás, tárolás, lekérdezés, szerkesztés, módosítás és nem utolsó sorban az adatok törlése is egy lehetőség. Ezen funkciókat egy adatbázis kezelő rendszer segítségével tudjuk elvégezni.[7] Azonban különbséget kell tennünk az SQL (Structured Query Language) és a NoSQL (Not only Structured Query Language) között.

6.1.1. SQL

Az elmúlt harminc évben a relációs adatbázis volt az alapértelmezett strukturált adatlekérdezési nyelv. Világunk fejlődése miatt egyre több és nagyobb információ adatok robbantak ki így az SQL alapú adatlekérdezés elveszítette hatékonyságát ezzel maga elé állítva azt a kihívást, hogy a nagyobb adatbázisok kezelése jóval megnehezedett. Ebből kifolyólag lehet arra következtetni hogy az SQL alapú szerverek hajlamosak nagy mennyiségű memóriát foglalni, biztonsági kockázatokat és teljesítmény-problémákat elkövetni.[8]

6.1.2. NoSQL

A NoSQL adatbázisokat azért alkották meg, mivel az SQL adatbázisok merev struktúrával rendelkeznek aminek következtében egy adott táblába nem tudunk kihagyni egy oszlopnak a feltöltését sem. Ennek következtében vagy fiktív adatokat, vagy üres mezőket kell megadjunk azokban a mezőkben amelyeket nem akarunk használni. A NoSQL adatbázisok sokkal rugalmasabbak lettek, fő céljuk az adatok könnyű tárolása és visszakeresése függetlenül a szerkezetektől és tartalomtól. Automatikusan kezelik az adatkezelést, hibajavítást amelyek költségmegtakarítás szempontjából is fontosak [8].

6.1.3. SQL vs NoSQL

A relációs adatbázisok az egyszerűség miatt leggyakoribb adatbázistípusok. Az adatok több táblára vannak bontva amelyekhez egyszerűen hozzá lehet férni. Az olyan műveletek, mint összeadás, létrehozás, visszakeresés, törlés stb. nagyon egyszerűen elvégezhetőek az SQL által megadott szintaxisok betartásával. Ilyen típusú adatbázis kezelő rendszerek a következők: Oracle, SQL Server, MySQL, PostgreSQL stb. A folyamatos adatmennyiség miatt a relációs adatbázisok hátrányba kerültek a nem relációs adatbázisokkal szemben. A nem relációs adatbázisok sokkal gyorsabban és hatékonyabban tudják elvégezni feladataikat mint a relációsak. Nem relációs adatbázis típusú rendszerek a következők lehetnek [9]: Firebase, MongoDB, GraphQL stb. A tanulmányok sokat segítenek abban, hogy egy adott projektben relációs vagy nem relációs adatbázist használjunk. Ha sok adatunk van és törekedünk a hatékonyságra akkor érdemesebb a nem relációs adatbázisokat használni.

Több tanulmány is szól az adatbázisok teljesítményéről. A következőkben bemutatnánk egyet. Ebben a tanulmányban három tesztet végeztek el. A kísérletben szerepelnek a következő NoSQL adatbázis kezelő rendszerek: Cassandra, HBase, Memcached, MongoDB, OrientDB, Redis és a Voldemort [10]. A tanulmányban három tesztet végeztek el. Mindhárom tesztben rendelkezésre állt 6.000.000 rekord. Ez a rekord szám nem is túl sok viszont nem is kevés, viszont mivel a tesztek az adatbázisok teljesítményét tesztelték olvasás, írás és frissítés esetén elegendőnek bizonyult ennyi adat is. A teszteket időre alpazták, hogy a különböző adatbázisok hány szekundum alatt végzik el a különböző teszteseteket.

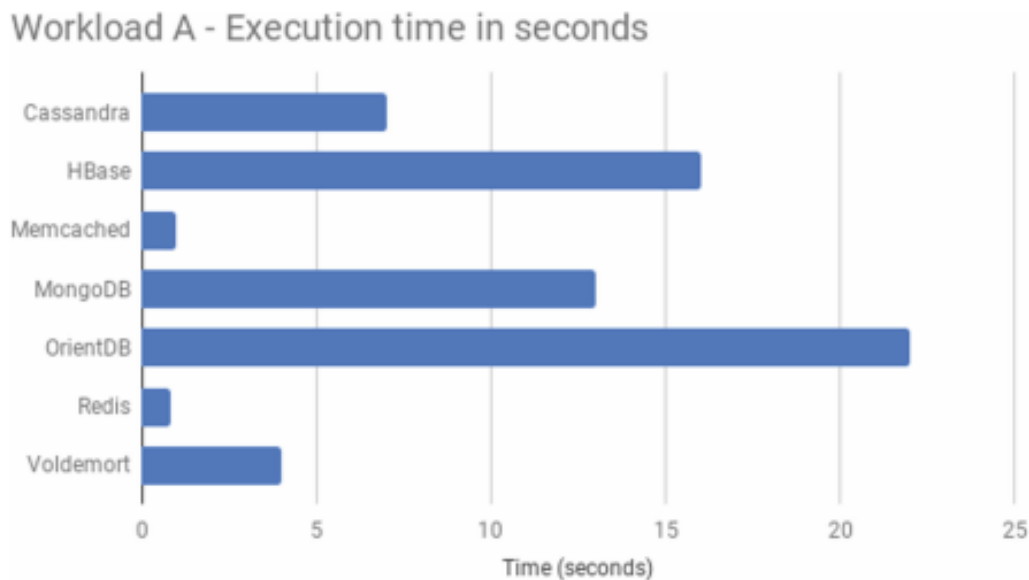
Az első tesztben a munkaterhelés fele olvasást és fele frissítést tartalmazott. Az eredmény a 6.1

ábrán látható. Az ábrát tekintve láthatjuk, hogy a legjobb időt a Redis és egy kevéssel lemaradva a Memcached teljesítette. A legrosszabb teljesítményt az OrientDB mutatta.

A második tesztnél a munkaterhelés 5.000 véletlenszerű olvasás volt. Az eredmények 6.2 ábrán láthatóak. Az első teszthez hasonlóan itt is a Redis és a Memcached teljesített a legjobban viszont utolsó helyre már a HBase került.

A harmadik tesztben a munkaterhelés 5.000 véletlenszerű frissítés volt. Az eredmények 6.3 ábrán láthatóak. Az első helyeket ismét a Redis és a Memcached szerezte meg míg az utolsó helyre visszakerült az OrientDB.

A tanulmányban leírt teszteket és NoSQL adatbázisokat tekintve a legjobb választás a Redis vagy a Memcached lenne. A legrosszabb választás az OrientDB lenne. A tanulmányban résztvevő többi adatbázis teljesítménye körülbelül megegyező volt, közepes teljesítménnyel, így a tanulmányt tekintve ezeknek a használata is megfontolandó.

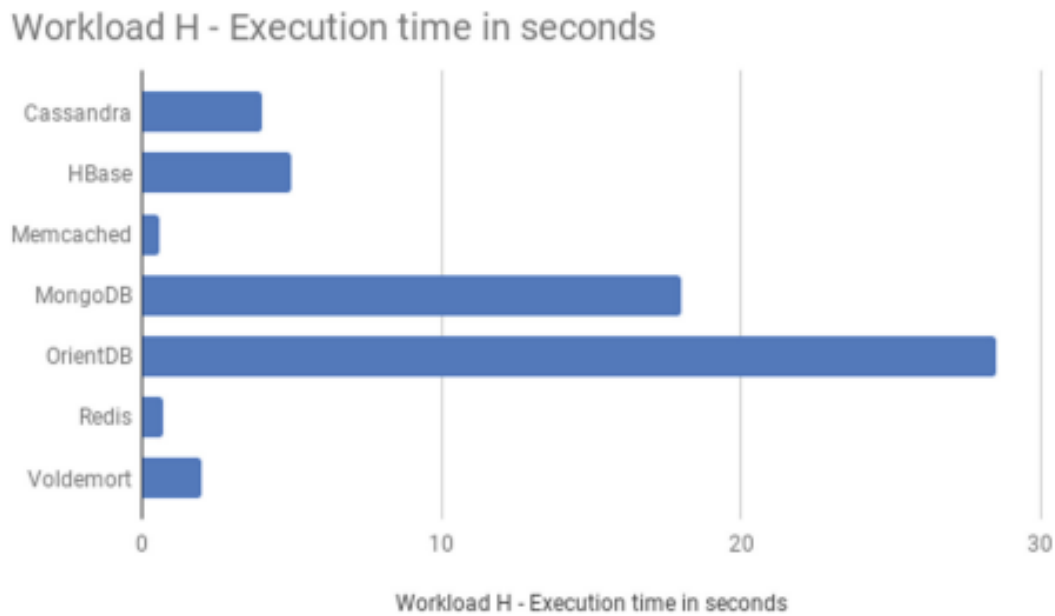


6.1. ábra. 6.000.000 rekord, 50% olvasás, 50% írás esetén¹

¹https://www.researchgate.net/figure/Workload-A_fig1_332028074



6.2. ábra. 6.000.000 rekord és 5.000 véletlenszerű olvasás esetén²



6.3. ábra. 6.000.000 rekord 5.000 véletlenszerű frissítés esetén³

²https://www.researchgate.net/figure/Workload-C_fig2_332028074

³https://www.researchgate.net/figure/Workload-H_fig3_332028074

6.2. Webes keretrendszerek

A keretrendszerek [11] napjainkban felkapott rendszerek lettek a felhasználók között. Ezek a rendszerek sokoldalúak, robusztusak és hatékonyak. A különböző alkalmazások fejlesztéséhez az ilyen jellegű rendszerek segítségével csak a magasabb szintű funkcionalitások elvégzésére kell koncentrálni. Erre az a magyarázat, hogy a keretrendszer gondoskodik az alacsonyabb funkcionalitásokról, amelyek már rengeteg tesztelt kódot tartalmaznak, így mi ezeket a funkcionalitásokat nem kell külön megírjuk és leteszteljük, hogy helyes-e a megírt kód. Számos előnyt lehet felsorolni, hogy miért jó ha használjuk ezeket a rendszereket.

- Elősegíti a tervezési minták megfelelő kialakítását.
- Biztonságosabb kódolás.
- A redundáns kód elkerülése.
- Következetes kódfejlesztés kevesebb hibával.
- Megkönnyíti a kód tesztelését és a hibakeresést is.
- Az alkalmazás fejlesztéséhez szükséges idő lecsökken.

Az 6.4 ábrán látható egy rangsorolás a Github és Stack Overflow által készített pontozások alapján a Front-end-et megvalósító keretrendszereknek. Látható hogy az első helyen a React van összesített 98 ponttal. A második helyen áll az ASP.NET MVC 95 ponttal viszont itt észlelhető az is, hogy a Githubról nem jött pontozás erről a keretrendszerről. Az ábrán látható további keretrendszerek ugyan annyi összpontozást kaptak. Az ábra alapján a legjobb döntés egy webes keretrendszer választásra a React lenne viszont a többi keretrendszer sincs sokkal lemaradva az első helyezettől.

Framework	Github Score	Stack Overflow Score	Overall Score
React	99	97	98
ASP.NET MVC		95	95
Angular	91	96	93
Ruby on Rails	87	99	93
AngularJS	90	97	93
Vue.js	100	87	93

6.4. ábra. Webes keretrendszerek Github és Stack Overflow pontozások alapján⁴

2021-ben a 10 leghasználatosabb Back-end keretrendszer a Spring - Spring Boot, NodeJS, Laravel, Django, Flask, Ruby, Play, Asp.Net, CakePHP és Symphony.[12][13]

A keretrendszerek világa igen nagy így a következőkben olvashatunk részletesebben néhány keretrendszerről mint például: az Angular, Vue.js, Spring Boot és a NodeJs.

6.2.1. Angular

A Google munkatársai 2008-ban fejlesztettek a JavaScript alapú Angular keretrendszert [14].Abban az időben a webhelyek zöme többoldalas alkalmazás megközelítésén alapult. A több oldalas alkalmazások lassúnak bizonyultak így bevezették az egy oldalas alkalmazásokat. Az egy oldalas alkalmazások abban jobbak a több oldalas alkalmazásoknál, hogy weboldal betöltése folyamán nem kell mindig mindent újra betölteni, hanem csak azok a részek töltődnek be ahol változások fognak megjelenni. Az Angular volt az egy oldalas alkalmazások első kerete. Az egyik fő előnye, hogy a felhasználók egyszerű struktúrával kell dolgozzanak. Megtanulják az Angular sajátos felépítését ezáltal gyorsan és optimálisabban tudnak benne fejleszteni. Az sem elhanyagolható, hogy a fejlesztők egy részletes és egyértelmű dokumentációval szolgáltak a felhasználóknak.

6.2.2. Vue.js

A Vue.js (röviden: Vue) [14] tekinthető az egyik legújabb keretrendszernek. Hasonlít az Angularhoz. Mindkettő TypeScript típusú. Használható kisebb, egyszerűbb projekteknel. Mindig egy oldalas alkalmazásokat lehet benne elkészíteni ami a felhasználói élményt segíti elő. Fő érdeme a skálázhatóság.

⁴<http://hotframeworks.com/>

Különlegessége, hogy egy nyílt forráskódú közösség fejlesztette ki, nem pedig egy nagyobb vállalat. Komponens alapú keretrendszer amely azt jelenti, hogy komponenseket különböztetünk és jelenítünk meg. Egy komponensen belül tudunk írni HTML, CSS és Script elemeket is. A megjelenítés egy oldalon történik ezért szükséges használni a útválasztást (rout).

6.2.3. Angular vs Vue.js

Mindkét keretrendszernek megvannak az előnyei 6.1 táblázat és a hátrányai is 6.2 táblázat.

6.1. táblázat. *Előnyök Angular és Vue.js között [15]*

	Angular	Vue.js
1	TypeScript használata	TypeScript használata, részletes dokumentáció
2	Részletes dokumentációval rendelkezik	Egy oldalas alkalmazások készítése
3	Gyorsítja a fejlesztést	Könnyű integráció a meglévő struktúrákba
4		Kihasználja virtuális DOM előnyeit
5		Sebessége és rugalmassága optimális

6.2. táblázat. *Hátrányok Angular és Vue.js között [15]*

	Angular	Vue.js
1	Számos különféle struktúrát kínál, nehezíti a tanulást	Kevesebb erőforrást kínál
2	Lassabb teljesítmény mert működik a reális DOM	

6.2.4. NodeJs

A NodeJs [16] egy olyan szoftver platform, amely a Chrome V8 JavaScript futási idején épül fel. Fontos tulajdonsága, hogy skálázható ezért is sokan használják. Eseményvezérelt, nem blokkoló

I/O modellt használ, amely könnyűvé és hatékonyá teszi a valós idejű alkalmazások megvalósítását, amelyek elosztott rendszeren futnak át. Tulajdonképpen közvetlenül a natív gépi kódra fordítja le a JavaScript-et, amelynek hatására az adatformátum egy lépése megszűnik, ezzel növelve az alkalmazás sebességét. Legtöbb esetben a NodeJs használatakor adatbázisnak NoSQL adatbázisokat választanak.

NodeJs a következő különlegességeket tartalmazza [17]:

- A NodeJS alkalmazások fejlesztésének elindítása könnyű.
- Az agilis fejlesztési módszertant követi, amely alkalmas a nagyon skálázható alkalmazásfejlesztési szolgáltatásokra.
- Nagy projekteknel gyorsabban működik mint a Java.
- Hatalmas erőforrás-készlet könyvtárakkal rendelkezik

6.2.5. Spring Boot

A Spring Boot [18] célja a Spring alkalmazás fejlesztés egyszerűsítése. Megtalálhatóak benne a következő tulajdonságok:

- Automatikus konfigurációk - Az alkalmazások Springként való működése érdekében.
- Indítófüggőségek - Biztosítja a felhasználóknak a szükséges függőségek(dependency) beimportálását. Ilyen lehet a Maven, Hibernate validátor, adatbázis eléréseket stb.
- Parancssori tolmács.
- Működtetés - A console-ba megjelennek az alkalmazás működésével kapcsolatos információk. Ilyen információ lehet a hiba, az elvégzett művelet stb.

Radikálisan gyorsabb és széles körben hozzáférhető, érthető Spring fejlesztést nyújt. Számos funkciót kínál: beágyazott szervereket, metrikákat, ellenőrzéseket, külső konfigurációkat. Saját struktúrával rendelkezik és nem tesz különbséget az adatbázisok között. A JAVA nyelvet használja.

Spring Boot a következő különlegességeket tartalmazza [17]:

- Egyszerű, minden eszköz és operációs rendszer támogatja.
- Beépített nyelvbiztonsági funkciókkal rendelkeznek, amelyeket a Java Compiler beágyaz.
- Robusztus kódot alkalmaz.
- Integrációs képesség jó.
- Beágyazott HTTP-kiszolgálókat, például Jetty, Tomcat használ és egyszerűen teszteli a webes alkalmazásokkal.

6.2.6. Spring Boot vs NodeJs

A Spring Boot és a NodeJs közötti néhány előnyt [19] az 6.3 táblázatban tudjuk megtekinteni.

6.3. táblázat. <i>Előnyök Spring Boot és NodeJS között</i>		
	Spring Boot	NodeJs
1	Java nyelv már nagyon ismeretes így ha elakadás van könnyű segítséget kérni/keresni	A JavaScript futásidejének köszönhetően gyors az adatfeldolgozásban
2	Többszálú programozás	Memória takarékos
3	Nagyszámú erőforrás áll rendelkezésre (például: az autentikáció már előre megvan írva)	NPM (Node Package Manager) -nek köszönhetően egyre több szolgáltatás érhető el.

Minden keretrendszer között találhatunk előnyöket is meg hátrányokat is. A Spring Boot és a NodeJs között előnyöket az 6.3 táblázatban már tárgyaltuk. A következőkbe néhány hátrányról [19] lesz szó:

- A NodeJS nem tud hatékonyan teljesíteni nagy számítások esetén.
- A NodeJS mivel még új technika ezért még nincs teljesen kifejlesztve, tesztelve így sok olyan hibába ütközhetünk amire nincs ideális megoldás.
- A Spring Boot legnagyobb hátránya, hogy a memóriaigénye nagyon nagy.

- A Spring Boot egy másik hátránya, hogy a hiba keresés meglehetősen nehéz mivel sok beépített kód található benne.

Összefoglalva mindkét technológia a maga módján megfelelő különböző alkalmazások tervezésénél. Ha az alkalmazás sok bemeneti/kimeneti feladatot (például: sok regisztráció) tartalmaz akkor mindenképp a NodeJs-t érdemes választani. Viszont ha biztonságos és önálló alkalmazást tervezünk amely intenzív CPU használatot igényel akkor a Spring Boottal érdemes foglalkozni.

7. fejezet

Használt technológiák

8. fejezet

Felhasználói felület bemutatása

8.1. Megvalósítások

9. fejezet

Összefoglalás

10. fejezet

Továbbfejlesztési lehetőségek

Irodalomjegyzék

- [1] T. P. Kersten, F. Tschirschwitz, and S. Deggim, „Development of a virtual museum including a 4d presentation of building history in virtual reality,” *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 361, 2017.
- [2] D. Dedov, M. Krasnyanskiy, A. Obukhov, and A. Arkhipov, „Design and development of adaptive simulators using 3d modeling,” *International Journal of Applied Engineering Research*, vol. 12, no. 20, pp. 10415–10422, 2017.
- [3] R. K. Moloo, S. Pudaruth, M. Ramodhin, and R. B. Rozbully, „A 3d virtual tour of the university of mauritius using webgl,” in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 2891–2894, IEEE, 2016.
- [4] D. Perdana, A. I. Irawan, and R. Munadi, „Implementation of a web based campus virtual tour for introducing telkom university building,” *International Journal of Simulation—Systems, Science & Technology*, vol. 20, no. 1, pp. 1–6, 2019.
- [5] R. Napolitano, I. Douglas, M. Garlock, and B. Glisic, „Virtual tour environment of cuba’s national school of art,” *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 42, no. 2, p. W5, 2017.
- [6] C. Maines and S. Tang, „An application of game technology to virtual university campus tour and interior navigation,” in *2015 international conference on developments of E-systems engineering (DeSE)*, pp. 341–346, IEEE, 2015.
- [7] „Dbms.” <https://www.tutorialspoint.com/dbms/index.htm>.

- [8] S. Venkatraman, K. Fahd, S. Kaspi, and R. Venkatraman, „Sql versus nosql movement with big data analytics,” *Int. J. Inform. Technol. Comput. Sci*, vol. 8, pp. 59–66, 2016.
- [9] A. Gupta, S. Tyagi, N. Panwar, S. Sachdeva, and U. Saxena, „Nosql databases: Critical analysis and comparison,” in *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, pp. 293–299, IEEE, 2017.
- [10] P. Martins, M. Abbasi, and F. Sá, „A study over nosql performance,” in *World Conference on Information Systems and Technologies*, pp. 603–611, Springer, 2019.
- [11] „Frameworks.” <https://hackr.io/blog/what-is-frameworks>.
- [12] „Top 10 emerging backend frameworks in 2021.” <https://www.decipherzone.com/blog-detail/top-10-backend-development-frameworks>.
- [13] „The most popular backend frameworks for web development in 2020.” <https://www.intagleo.com/blog/most-popular-backend-frameworks-for-web-development-in-2019/>.
- [14] E. Wohlgethan, *Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js*. PhD thesis, Hochschule für Angewandte Wissenschaften Hamburg, 2018.
- [15] „Vue and angular comparison.” <https://www.educative.io/blog/react-angular-vue-comparison>.
- [16] N. Js and N. JS, „Node.js,” *Tradução de: SILVA, AG Disponível em*, 2016.
- [17] „Nodejs vs spring boot.” <https://www.chapter247.com/blog/node-js-vs-springboot-java-which-one-to-choose-and-when/>.
- [18] Ž. Jovanović, D. Jagodić, and Vujičić, „Java spring boot rest web service integration with java artificial intelligence weka framework,” in *International Scientific Conference “UNITECH 2017*, pp. 323–327, 2017.
- [19] „Nodejs vs spring boot : Picking up the right technology.” <https://www.inexture.com/nodejs-vs-spring-boot-choosing-the-best-technology/>.