



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

Informatikai Kar

Információs Rendszerek Tanszék

Szakdolgozat/diplomamunka téma
nyilvántartó oldal SAP CAP
használatával

Témavezető:

dr. Vincellér Zoltán
mesteroktató

Szerző:

Papp Ádám Máté
programtervező informatikus BSc

Budapest, 2021

Tartalomjegyzék

Köszönetnyilvánítás	5
1 Bevezetés.....	6
1.1 A probléma bemutatása	6
1.2 Probléma megoldása.....	6
2 Felhasználói dokumentáció	8
2.1 Futó applikáció használata.....	8
2.1.1 Nyitóoldal.....	11
2.1.2 Témák.....	11
2.1.3 Témaszövegek megtekintése.....	13
2.1.4 Témaszövegek kulcsszavai.....	14
2.1.5 Téma beadása	15
2.1.6 Témák szerkesztése	16
2.1.7 Felhasználók megtekintése.....	17
2.1.8 Szakok szerkesztése	18
2.1.9 Karok szerkesztése	19
2.1.10 Határidők szerkesztése	19
2.1.11 Statisztikák	20
3 Fejlesztői dokumentáció.....	21
3.1 Az applikáció telepítése.....	21
3.2 Applikáció futtatása.....	23
3.3 Felhasznált technológiák	23
3.3.1 SAP Business Technology Platform	23
3.3.2 Cloud Foundry (CF)	24
3.3.3 SAPUI5	24
3.3.4 CAP	25
3.3.5 CDS	26

3.3.6	OData	27
3.3.7	MVC architektúra	28
3.3.8	Node.js	29
3.3.9	Felhasználó azonosítása, jogok kiosztása	29
3.3.10	Approuter	33
3.3.11	MTA.....	35
3.3.12	i18n	35
3.3.13	BAS fejlesztői környezet	36
3.3.14	Git verziókezelés.....	36
3.4	Adatmodell.....	37
3.5	Táblák.....	38
3.5.1	Theses	39
3.5.2	ThesisText.....	40
3.5.3	Admins.....	40
3.5.4	Teachers	40
3.5.5	Students.....	41
3.5.6	Faculties	41
3.5.7	Courses.....	41
3.5.8	Literatures	42
3.5.9	Prerequisites.....	42
3.5.10	Tags.....	42
3.5.11	Teachers2Theses.....	42
3.5.12	Students2Theses.....	43
3.5.13	Courses2Theses	43
3.5.14	Literatures2Theses	43
3.5.15	Prerequisites2Theses.....	44
3.5.16	Tags2Theses.....	44

3.6	Architektúra	44
3.7	MVC részek.....	46
3.7.1	Modellek.....	46
3.7.2	Nézetek.....	47
3.7.3	Kontrollerek	48
3.8	Tesztelés	55
3.8.1	Szerveroldal.....	55
3.8.2	Felhasználói felület - használati eset tesztek.....	58
	Összegzés	62
	Fejlesztési lehetőségek	62
	Irodalomjegyzék.....	62

Köszönetnyilvánítás

Köszönöm Vincellér tanár úrnak, hogy elvállalta a témavezetést, köszönöm Béleccki Andrásnak, hogy oly sok mindenben segített a fejlesztés során, valamint köszönöm minden rokonomnak és barátomnak, hogy mellettem voltak és támogattak a szakdolgozatírás és koronavírus által igencsak megváltozott napjaimban is.

1 Bevezetés

1.1 A probléma bemutatása

A felsőoktatási tanulmányok lezárásának a része, hogy a hallgatónak el kell készíteni egy összetett dokumentumot, ami egy, a hallgató által kiválasztott témához kapcsolódik. Úgy tartja a szokás, hogy ezt a dokumentumot alapszakon szakdolgozatnak, mesterszakon esetleg diplomamunkának hívják. Az egyszerűség kedvéért a későbbekben használt „*szakdolgozat*” szó alatt mindkettőt értem.

A szakdolgozati témaválasztás egy rendkívül komoly feladat, és igencsak stresszes is, különösen azoknak, akik teli vannak olyan kérdésekkel, mint például: „*Mi lehetne jó témának?*”, „*Mihez értek igazán?*”, „*Kit kérjek fel témavezetőnek?*”, „*Vajon kinek van még kapacitása velem foglalkozni?*”.

Én is hasonlóan tanácstalan voltam a választás előtt, és mint azt már akár Steve Jobstól is lehet tudni, „*Az emberek gyakran nem tudják, hogy mit is akarnak, amíg meg nem mutatod nekik.*” – éppen ezért igazán hasznos lett volna egy olyan szoftveres rendszer vagy platform, amelynél egy helyen megtekinthető lett volna, hogy mely oktató milyen témákkal foglalkozik, miről is szólna a téma, lehet-e még jelentkezni, van-e még szabad hely. Ilyen nyilvántartás sajnos nem volt elérhető, illetve jelen szakdolgozat írása közben sincs.

1.2 Probléma megoldása

Mivel még nem volt elérhető az, amire szükségem lett volna, és szakdolgozati témám sem volt, így ötvözve a kettőt jött az ötlet, hogy jelen szakdolgozatom szóljon egy szakdolgozati témamenedzser jellegű webes applikáció megvalósításáról, ahol még a szakdolgozati szöveget is be lehet adni, és a beadott szövegen pedig különböző eljárásokat lehetne alkalmazni, akár kulcsszavakat is keresni, és belőlük szófelhőt készíteni. Az alkalmazásban külön felületek volnának elérhetőek hallgatóknak, oktatóknak és az adminisztrátoroknak.

Egy ilyen alkalmazás nem zárná el azt a lehetőséget a hallgatók elől, hogy saját témát készítsenek, hisz az oktatóval történő előzetes egyeztetés után az oktató létrehozhat egy olyan témakiírást, ahová már akár a publikáláskor felveheti a hallgatót. Az viszont elkerülhető vele, hogy órák, napok teljenek el azzal, hogy a hallgatók az oktatók oldalait

kutatják fel témák után, illetve, hogy az email-es megkeresések valamilyen ok miatt nem érnek célba, és türelmetlen várakozással megy az idő, miközben a bejelentés határideje csak közeledik, és egyre inkább lekötődnek az oktatók szabad kapacitásai.

Az applikációmnak a Témamenedzser nevet adtam.

2 Felhasználói dokumentáció

2.1 Futó applikáció használata

Az alkalmazás eléréséhez internetkapcsolatra van szükség, illetve egy olyan böngészőre, ami támogatja a manapság szinte kivétel nélkül mindenhol használt modern webes technológiákat. Ilyen böngésző lehet például a Microsoft Edge, Google Chrome, Mozilla Firefox, Opera, Safari stb. Az, hogy milyen platformról (Android, iOS, Windows, macOS, Debian, Ubuntu stb.) fut a böngésző, a webes alkalmazás használata szempontjából mindegy.

Fontos szempont volt már a tervezéskor, hogy a weboldal reszponzív legyen, tehát hogy az alkalmazás ne csak nagy kijelzőjű eszközökön (pl. asztali számítógép) működjön az úgy, ahogy elvárná az ember, hanem mobil eszközökön is jól jelenjenek meg az elemek, működjön a navigáció és esztétikailag is szép látvány legyen.

A Témamenedzser sötét és világos témákat is támogat. Ennek meglétét is fontosnak találtam, hiszen manapság már rengeteg szoftverben elérhető ez a funkció.

Az alkalmazást három felhasználói csoport használhatja: hallgatók, oktatók és adminisztrátorok.

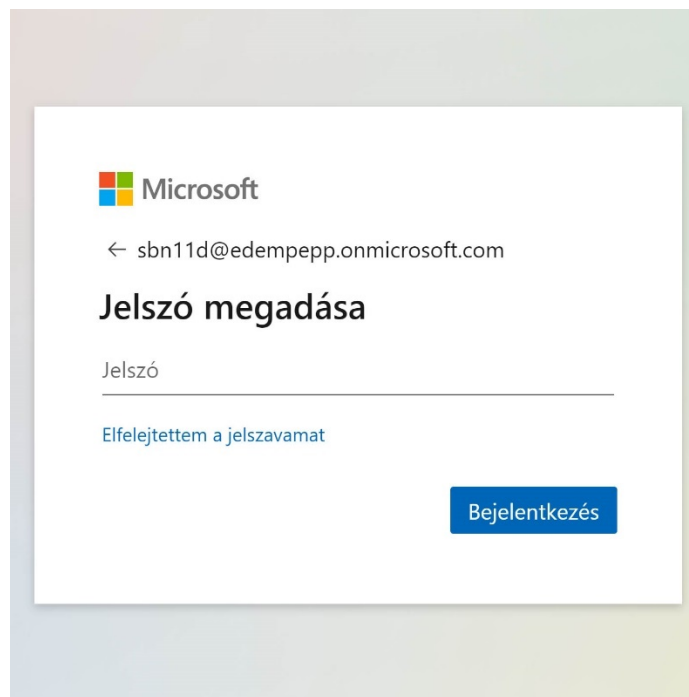
A hallgatóknak lehetőségük van nem archív szakdolgozati témákat böngészni, azokra feljelentkezni és az adott szakhoz tartozó határidőig egy szövegezést beadni. Megtekinthetik a már beadott szövegeket, illetve azokon kulcsszókeresést is elvégezhetnek. Az így megtalált kulcsszavakból, vagy esetleg a szakdolgozati szövegből véletlenszerűen kiválasztott szavakból tetszőleges méretű, betűtípusú és szószámú szófelhőket készíthetnek.

Az oktatóknak lehetőségük van szakdolgozati témákat kiírni, valamint azokat szerkeszteni. A szerkesztéskor beállíthatják, hogy a téma mettől meddig legyen elérhető. Ebben az időszakban jelentkezhetnek fel rá a hallgatók, de ezen az időszakon belül ki is lehet kapcsolni a témára jelentkezést. A témához létszámkorlátot lehet szabni, meg lehet adni, hogy mely szakoknak ajánlott az adott téma, hogy milyen irodalmak áttanulmányozását javasolja az oktató a témával kapcsolatban, illetve, hogy a jelentkezéshez milyen előismeretek kellenek. Külön beállítható, hogy csak az ajánlott szakokon lévő hallgatók jelentkezhessenek a témára, vagy szaktól függetlenül is lehet.

Az oktatók explicit is megadhatják, hogy mely hallgatók és társoktatók legyenek a témához kapcsolva. A témákat lehet archiválni, ezután nem módosítható semmi sem a témán, vagy ahhoz kapcsolódóan sem, tehát nem lehet a hozzá kapcsolódó hallgatókat, oktatókat, tárgyakat, címkéket stb. törölni vagy újakat hozzáadni.


Az adminisztrátorok megtekinthetik a felhasználókat rangok szerint szétbontva, új karokat és szakokat hozhatnak létre, valamint beállíthatják a szakonként különböző szakdolgozat leadására vonatkozó határidőt.





Az alkalmazás használatához először be kell jelentkezni, ezt pedig az alkalmazáshoz társított identitásszolgáltatón keresztül tehetjük meg a megfelelő felhasználónév és jelszó páros után. Az alkalmazás jellege miatt úgy gondoltam, nem teszem lehetővé a közvetlen regisztrációt. Új felhasználó csak azután léphet be, miután az azonosítási ügynököt kezelő személy hozzáadta a felhasználók listájához.

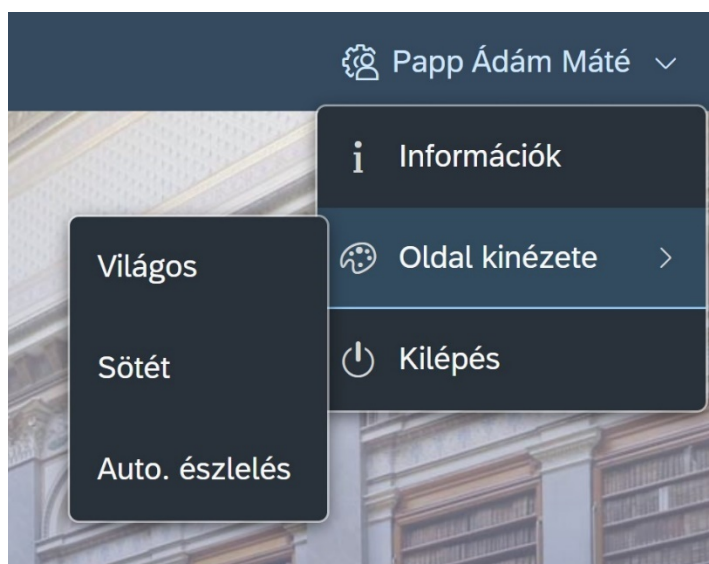


2.1. ábra. Az általam használt Azure Active Directory bejelentkező felülete.

Bejelentkezés után megjelenik az applikáció felhasználói felülete, aminek állandó része az oldal tetején lévő sáv, ahol a felhasználóra vonatkozó információk és beállítások érhetőek el, valamint egy menü, aminek a tartalma függ attól, hogy a felhasználó milyen

csoportba tartozik, vagyis, hogy hallgató, oktató vagy adminisztrátor-e, viszont minden csoportból elérhető a „¹ Nyitóoldal”, ami fixen a menü alján van.

Az állandó sáv jobb szélén van egy „” ikonnal kezdődő, majd a felhasználó nevével folytatódó gomb, aminek megnyomására egy kis menü jelenik meg. Ebben a menüben az első lehetőség az „ Információk” gomb, melynek megnyomására felugrik egy ablak a felhasználóra vonatkozó információkkal, mint például a neve, neptun kódja, email címe, beosztása, esetleg a hozzátartozó szak és kar azonosítója. A második opció az „ Oldal kinézete” gomb, ennél explicit kiválaszthatjuk, hogy világos, sötét, vagy esetleg az operációs rendszerével megegyező témájú legyen az applikáció. A megnyomására az applikáció minden eleme a kívántnak megfelelő színvilágot vesz fel. A harmadik és egyben utolsó opció a „ Kilépés”, aminek megnyomására elindul a felhasználó kijelentkeztetési folyamata, melynek végén a felhasználó az azonosítási ügynökből is kilép.



2.2. ábra. Menü a beállítások gombnál.

Az alábbiakban bemutatom, hogy mely oldalak érhetőek el az állandó sáv menü részéből, és hogy azokon milyen tevékenységet lehet végezni.

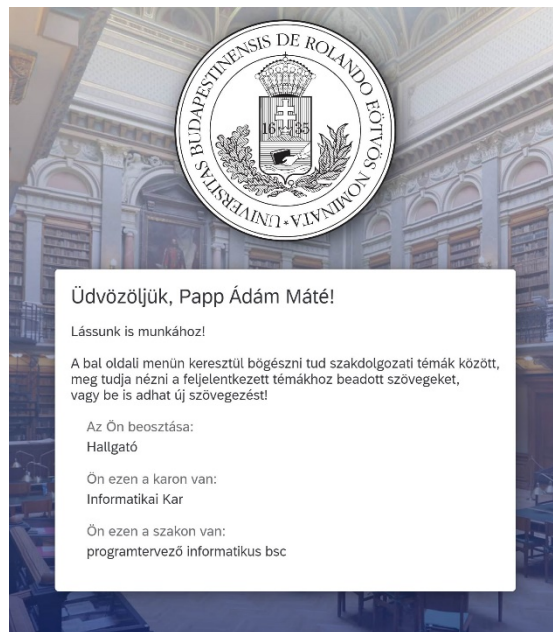
¹ Az ikonok az SAP Icons csomagban találhatóak meg

2.1.1 Nyitóoldal

„🏠” ikonnal társítva található meg a menüben. Ez az oldal nyílik meg az alkalmazás megnyitásakor is, hacsak nem hívtunk meg közvetlenül egy másik oldalt az alkalmazáshoz vezető linkben. Ezen az oldalon az ELTE Egyetemi Könyvtárának egyik terme látható háttérként², középen pedig az ELTE címere foglal helyet, alatta pedig egy ablakban van a felhasználó számára üdvözlő üzenet, rövid leírás a lehetséges tevékenységekről, valamint információk a felhasználóról.



2.3. ábra. Nyitóoldal sötét témával.

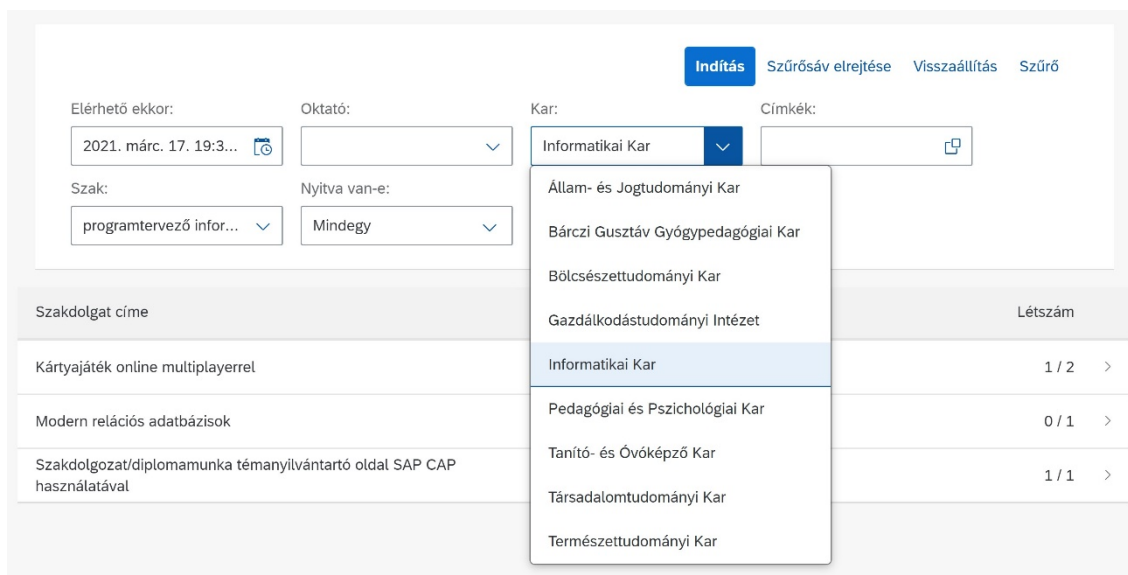


2.4. ábra. Nyitóoldal világos témával.

2.1.2 Témák

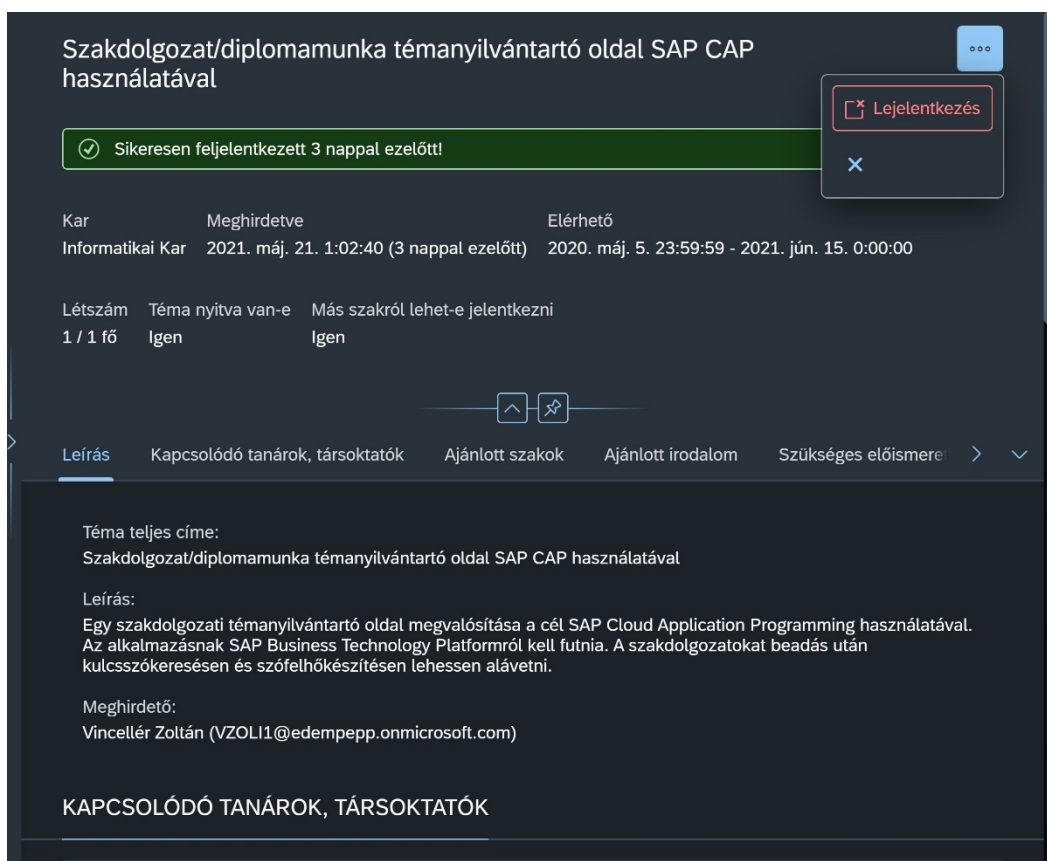
„🔍” ikon előzi meg a menüben. Ezt az oldalt csak oktatók és hallgatók érhetik el. Az oldal tetején egy keresősáv látható, aminek a használatával részletesen tudunk keresni témákra dátum szerinti elérhetőség, téma állapota szerinti elérhetőség, kapcsolódó oktató, kar, szak, téma neve és témához kapcsolt címkék alapján. A hibás keresési feltételeket (például rossz dátum formátum, nem létező oktató, szak stb.) egy piros keret jelzi annál az elemnél, aminek a tartalma hibás. A keresősáv alatt egy lista látható, ebben eleinte minden nem archivált téma fel van sorolva, keresés után pedig csak annak eredménye.

² A kép készítőjét nem sikerült kideríteni



2.5. ábra. Kereső és az eredménylista világos témával.


Ha rákattintunk egy témára, nagy kijelzőn nézve a képernyő jobb oldalán, mobil eszközről nézve pedig új oldalon megjelenik a téma részletes leírása, ami tartalmazza a téma teljes címét, leírását, a létszámadatokat, elérhetőséget, kapcsolódó oktatókat, ajánlott szakokat, megjelölt követelményeket, címkéket és irodalmi ajánlókat.



2.6. ábra. A kiválasztott témát részletező oldal sötét témával.

Ez a konstrukció egy úgynevezett „*master-detail*” (tükörfordításban mester-részlet) sablont valósít meg, ahol a fő (master) részen a témák felsorolása látható, a részletek (detail) pedig a kiválasztott témára vonatkozóan jelennek meg. Ha hallgatóként nézi a felhasználó ezt az oldalt, akkor egy extra funkció is elérhető, ugyanis itt lehet feljelentkezni a témára, amennyiben van még szabad hely rajta, elérhető az aktuális időpontban, nyitva van, a hallgató más témára nem jelentkezett még fel és megfelelő szakon van. Ha az éppen kiválasztott téma az, amire fel van jelentkezve (ezt egy kis üzenetcsík is jelzi a feljelentkezés dátumával), akkor egy olyan gomb jelenik csak meg, amivel le tud jelentkezni róla. Egy hallgató tehát egyszerre csak egy témára lehet feljelentkezve, kivéve, ha az oktatók több témára is feljelentkeztetik. Az oktatók be tudják állítani, hogy csak az ajánlott szakokról lehessen jelentkezni, vagy bármilyen másról is. Utóbbi esetben a hallgató irányába ezt egy figyelmeztető üzenetcsík jelzi, valamint jelentkezéskor meg kell erősítenie a szándékát.

2.1.3 Témaszövegek megtekintése

„” ikonnal ellátva van a menüben. Ezt az oldalt oktatók és hallgatók is elérhetik. A már korábban („2.1.2. Témák” bekezdés) említett „*master-detail*” koncepciót implementálva a menüpont kiválasztásakor egy olyan felület nyílik meg, aminél először a felhasználóhoz kapcsolódó témák listájából kell választani egyet, majd ezután nagy kijelzőn nézve a képernyő jobb oldalán, mobil eszközről nézve pedig új oldalon megjelenik a témához tartozó információk közül a téma teljes címe és leírása, alatta pedig a témához beadott szöveg, valamint a beadás ideje és a beadó hallgató email címe.

Megtekinthető szövegek:	
Szakdolgozat/diplomamunka témanyil...	<p>Téma teljes címe:</p> <p>Szakdolgozat/diplomamunka témanyilvántartó oldal SAP CAP használatával</p> <p>Leírás:</p> <p>Egy szakdolgozati témanyilvántartó oldal megvalósítása a cél SAP Cloud Application Programming használatával. Az alkalmazásnak SAP Business Technology Platformról kell futnia. A szakdolgozatokat beadás után kulcsszókeresésen és szöveghőkészítésen lehessen alávetni.</p> <p>Utoljára szerkesztve:</p> <p>2021. máj. 24. 5:46:15 (SBN11D@edempepp.onmicrosoft.com)</p> <p>Téma szövege:</p> <div> <p>EÖTVÖS LORÁND TUDOMÁNYEGYETEM</p> <p>Informatikai Kar</p> <p>Információs Rendszerek Tanszék</p> </div>

2.7. ábra. A témaszöveg megtekintése oldal, sötét témával.

Ha még nem adtak be a kiválasztott témához semmilyen szövegezést, úgy egy ablakban ezt jelzi az applikáció a felhasználónak. Ezen az oldalon csak megtekinteni lehet az információkat.

2.1.4 Témaszövegek kulcsszavai

„🔑” ikonnal szerepel a menüben. Ez az oldal oktatók és hallgatók által is elérhető, kialakításában hasonlít a „📄 Témaszövegek megtekintése” oldaléhoz. A kívánt téma kiválasztása után megjelenik egy olyan felület, ahol felsorolásra kerülnek a témához beadott szöveg kulcsszavai, valamint a beadott szövegezés is. Ez a kettő egy-egy összecusukható elembe van. A szófelhő generálást a d3-cloud külsős könyvtár végzi.




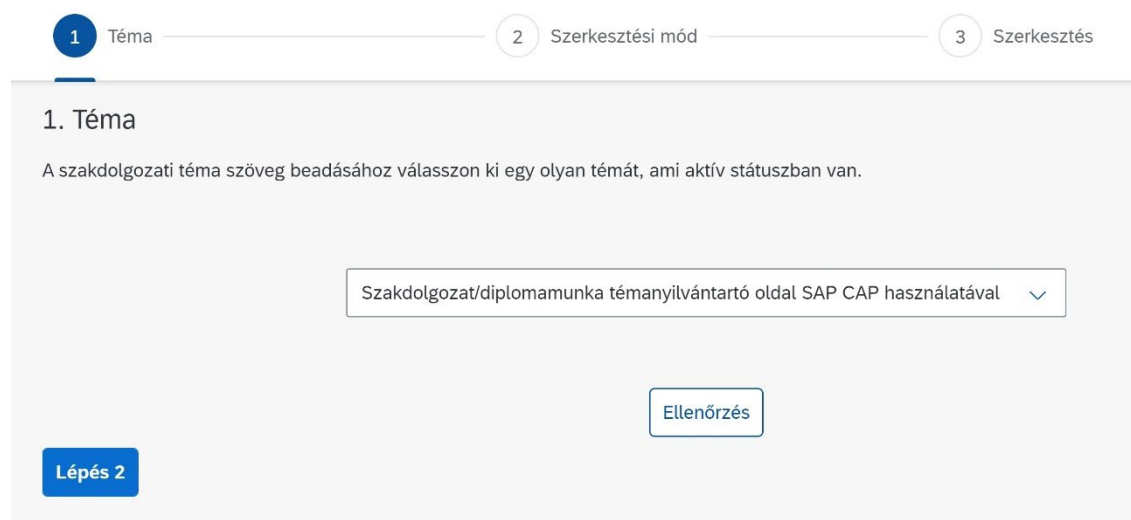
2.8. ábra. Egy kulcsszavakból generált szófelhő.

Az oldal tetején egy menüsáv van, benne a „⚙️ Szófelhő beállításai” gombbal. Ennek megnyomására egy ablak jön elő, amelyben beállítható a szófelhő magassága és szélessége amennyiben még nincs szófelhő létrehozva, valamint a szófelhőben megjelenő szavak száma és azoknak betűtípusa. Ebből az ablakból a „💾 Mentés” gombbal lehet kilépni. Ezen gomb mellett van a „🗑️ Szófelhő törlése” gomb is, amivel már legenerált szófelhőt lehet eltüntetni a felhasználói felületről. A menüsáv alatt a kiválasztott téma

címe foglal helyett, alatta pedig egy gomb van, aminek segítségével generálhatunk egy szófelhőt vagy a korábban megtalált kulcsszavakból, vagy véletlenszerűen választva a témához beadott szövegezésből. A szófelhő beállításáiban módosított értékek is csak esetleges újragenerálás után érvényesülnek.

2.1.5 Téma beadása

„” ikon van mellette a menüben. Ezt az oldalt csak hallgatók láthatják, és egy úgynevezett varázsló segítségével szövegezést adhatnak be szakdolgozati témákhoz. A varázsló három lépésben segít elvégezni ezt, első körben ki kell választani a témát, második körben pedig azt, hogy meglévő szöveget szeretne-e szerkeszteni, vagy teljesen újat szeretne beadni. Amennyiben a kiválasztott téma archív, vagy a hallgató szakán elmúlt a témabeadási határidő, egy piros üzenetcsík jelenik meg, ezeket a hibákat jelezve.



2.9. ábra. A varázsló első lépésénél a téma kiválasztása, világos témával.

Utolsó lépésben pedig megjelenik a témára vonatkozó információk közül az, hogy mi a téma teljes címe, leírása, mikor, milyen karon hirdették meg és hány fő részére. Továbbá van még egy névsor a feljelentkezett hallgatókkal, illetve a kapcsolódó tanárok és társoktatókkal kapcsolatban. Ezek alatt helyezkedik el egy szövegdoz, ami a varázslóban korábbiakban kiválasztottaknak megfelelően vagy tartalmazza már a szakdolgozat szövegét, vagy teljesen üres. Amikor a hallgató befejezte a szerkesztést, a következő lépésben megjelenik egy összesítő oldal, ahol fel van tüntetve a módosított szöveg, valamint lehetőség van visszamenni az előző lépésre, és ismételten szerkeszteni a szöveget. Ha a felhasználó úgy dönt, jóváhagyhatja a módosításokat és ekkor elmenti

az új szöveget, de bármikor kiléphet a szerkesztési folyamatból. Utóbbi kiválasztásakor a már meglévő szövegezéssel nem történik semmi.

2.1.6 Témák szerkesztése

„✎” ikonnal található meg a menüben. Ezt a menüpontot csak oktatók érhetik el. Ennél a „📄 Témaszövegek megtekintése” oldalhoz hasonló felépítésben meg lehet tekinteni a felhasználóhoz kapcsolódó témákat, valamint azokból egyet kiválasztva megjelenik az összes, témához tartozó információ. Az oldal tetején két gomb van, az egyik az „➕ Új téma kiírása”, a másik a „✎ Szerkesztés”.

➕ Új téma kiírása ✎ Szerkesztés

Szakdolgozati téma adatai - nézet

Téma egyedi azonosítója: f555f2b0-0389-4b4e-8a0b-f0d1da194a12

Téma teljes címe: Szakdolgozat/diplomamunka témanyilvántartó oldal SAP CAP használatával

Leírás: Egy szakdolgozati témanyilvántartó oldal megvalósítása a cél SAP Cloud Application Programming használatával. Az alkalmazásnak SAP Business Technology Platformról kell futnia. A szakdolgozatokat beadás után kulcsszókeresésen és szófelhőkészítésen lehessen alávetni.

Kar: Informatikai Kar

Maximum férőhely: 1

Téma nyitva van-e: Igen

Csak ajánlott szakról lehet-e jelentkezni: Nem

Téma archivált-e: Nem

Meghirdetve: 2021. máj. 21. 1:02:40 (1 héttel ezelőtt)

Elérhető: 2020. máj. 5. 23:59:59 - 2021. jún. 15. 0:00:00

Létrehozó: VZOL11@edempepp.onmicrosoft.com 2021. máj. 21. 1:02:40 (1 héttel ezelőtt)

Utoljára szerkesztve: VZOL11@edempepp.onmicrosoft.com 2021. máj. 24. 1:35:20 (4 nappal ezelőtt)

Feljekentkezett hallgatók:


Név	Feljelentkezés	Módosítások	Shak
Papp Ádám Máté	2021. máj. 21. 20:45:52 (7 nappal ezelőtt)	4	programtervező informatikus bsc

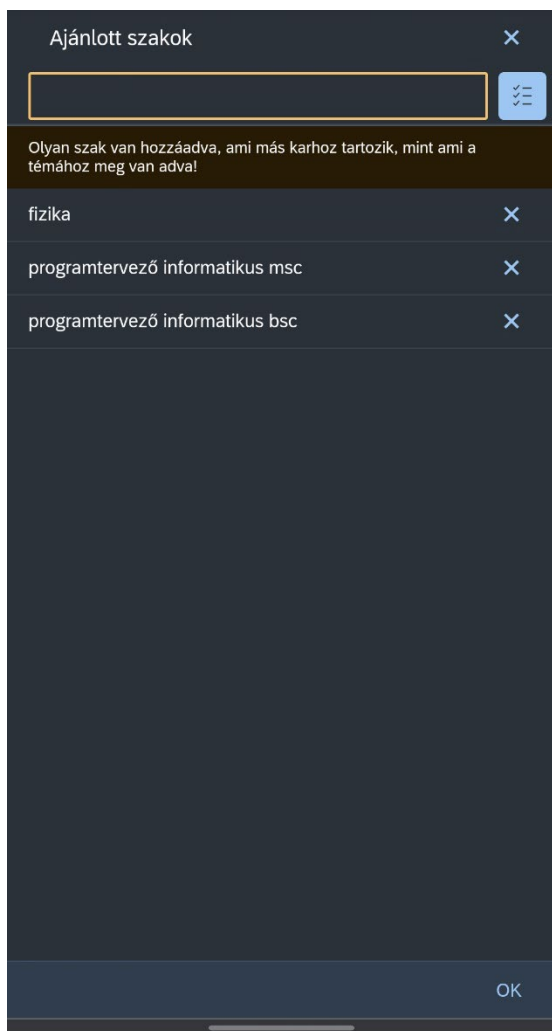
Kapcsolódó tanárok, társoktatók:

Név	Kar	E-Mail
Vincellér Zoltán	Informatikai Kar	VZOL11@edempepp.on microsoft.com

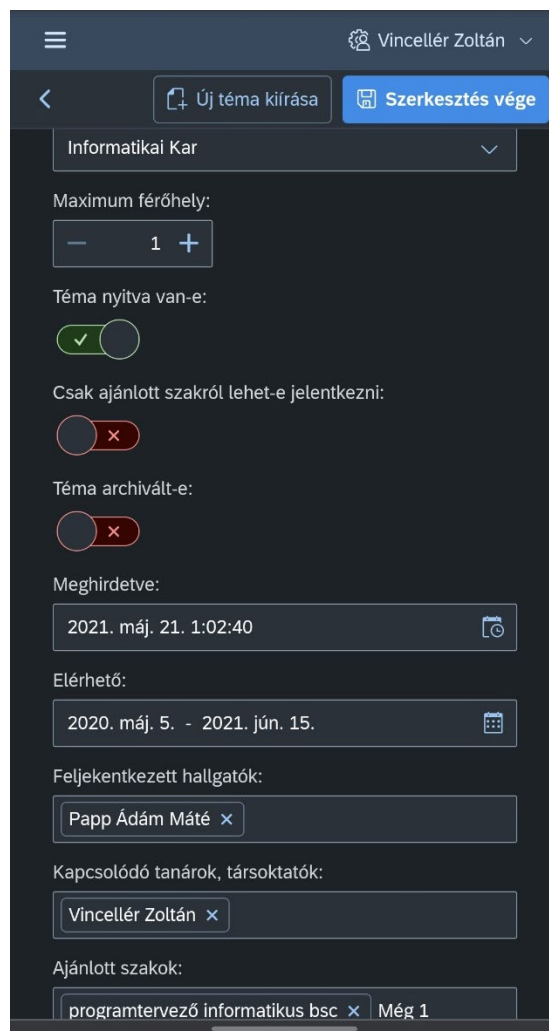
2.10. ábra. A kiválasztott téma információit összesítő oldal, világos témával.

A gombok megnyomása esetén megmaradnak az információk az oldalon, csupán szimpla szöveges sor helyett egy szerkeszthető mezőbe kerülnek át. Módosítani lehet a dátumoktól kezdve a hozzárendelt hallgatókon át a címkékig mindent. A szükséges előismeretek, ajánlott irodalmak, valamint címkék megadásánál ahogy a felhasználó elkezd gépelni, találatokat kap a már létező tartalmak közül. Teljesen újat is létrehozhat, ha előtte megnyomja a „★” ikonnal jelzett gombot a mező sorának végén. Az adatok automatikusan mentődnek a szerkesztés közben. Az archiv témákat nem lehet szerkeszteni, csak az ezt beállító mezőt. Ha a hallgatókat, oktatókat és szakokat felsoroló

mezőkben olyan elem van, aminek a kara nem egyezik a témáéval, egy sárga üzenet jelenik meg. Egy ugyanilyen struktúrájú oldalra visz az új témát létrehozó gomb is egy megerősítést kérő ablak után, annyi különbséggel, hogy valamennyi sor üres lesz, eltekintve a téma nevétől, ami „Új téma” lesz, a kar megegyezik az oktatóhoz tartozó karral, valamint a meghirdetés dátuma az akkori időponttá válik. A szerkesztési módból az oldal tetején lévő „ Szerkesztés befejezése” gomb megnyomásával lehet kilépni.










2.11. ábra. Figyelmeztetés a szakok szerkesztésénél, mobil nézetrel, sötét témával.



2.12. ábra. Szerkesztési oldal, mobil nézetrel, sötét témával.

2.1.7 Felhasználók megtekintése

Kizárólag adminisztrátorok által elérhető, „” ikonnal jelzett menüpont. Ennél az opciónál egy ikonokkal elválasztott táblázatsorozat jelenik meg, ami beosztás szerint szeparálja a felhasználókat: „” ikonnal a hallgatókat, „” jelzi az oktatókat, és „” kép van az adminisztrátoroknál. Ezen az oldalon csak megtekinteni lehet az olyan információkat, mint egy felhasználó neve, email címe, esetleg a kar és a szak, amihez hozzá van rendelve.

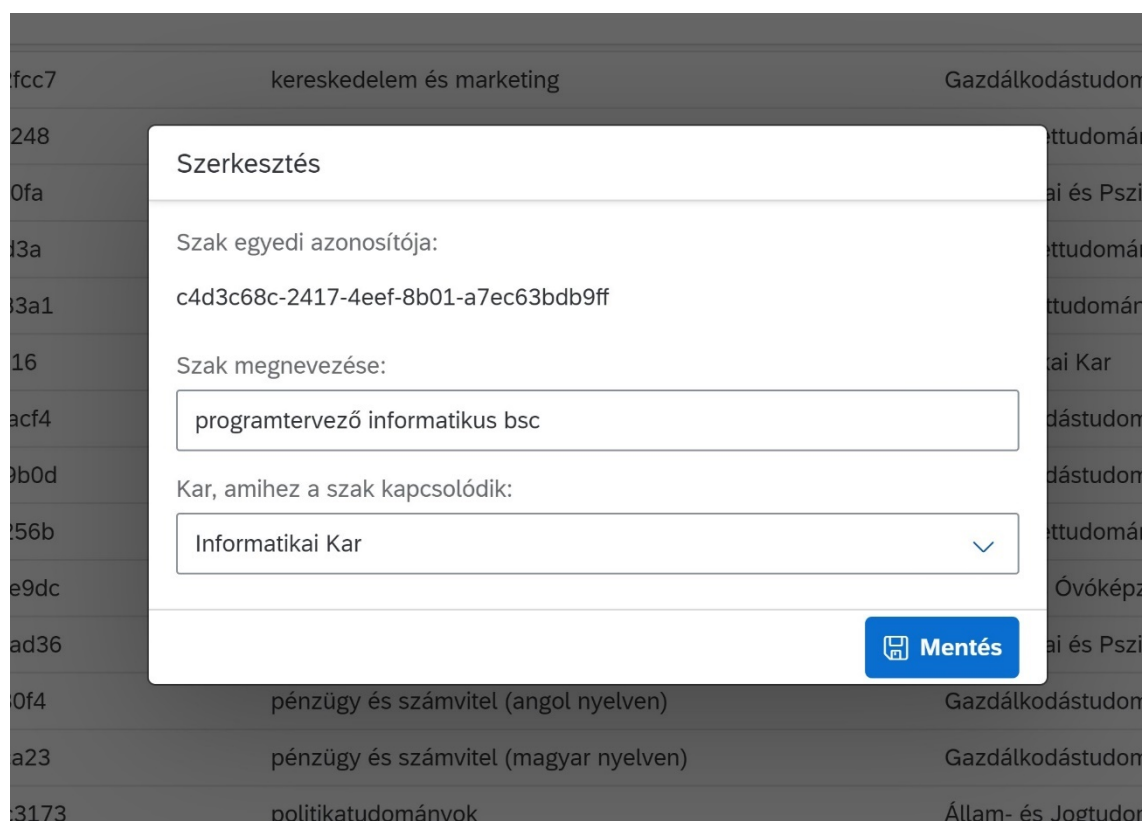
		
Hallgatók	Oktatók	Adminisztrát...

Név	Kar
Papp Ádám Máté	Informatikai Kar
Lukács László	Informatikai Kar

2.13. ábra. Felhasználók megtekintése oldal, részlet. Világos témával

2.1.8 Szakok szerkesztése

A menüben „☰” ikonnal társított, csak adminisztrátorok által megtekinthető oldal. Itt egy lista jelenik meg, ami tartalmazza az adatbázisban lévő szakok egyedi azonosítóit, neveit, illetve az, hogy melyik szak melyik karon van. Az oldal tetején bal oldalt van az „+ Új szak hozzáadása” gomb. Az utóbbi gombnak, vagy egy listaelemnek a megnyomására felugrik egy ablak, ami egy űrlapot tartalmaz, szerkeszthető mezőkkel. Új szak esetén ezek üresek, egyébként pedig a kiválasztott listaelem értékei vannak behelyettesítve. A szerkesztéskor automatikusan mentésre kerül a módosított adat.



fcc7

kereskedelem és marketing

Gazdálkodástudom

248

ttudomái

0fa

ai és Pszi

13a

ttudomái

13a1

ttudomár

16

ai Kar

acf4

dástudon

9b0d

dástudon

156b

ttudomái

e9dc

Óvóképz

ad36

ai és Pszi

0f4

pénzügy és számvitel (angol nyelven)

Gazdálkodástudon

a23

pénzügy és számvitel (magyar nyelven)

Gazdálkodástudon

173

politikatudománvok

Állam- és Jogtudor

Szerkesztés

Szak egyedi azonosítója:

c4d3c68c-2417-4eef-8b01-a7ec63bdb9ff

Szak megnevezése:

programtervező informatikus bsc




Kar, amihez a szak kapcsolódik:

Informatikai Kar



Mentés

2.14. ábra. Szak szerkesztése, világos témával.





2.1.9 Karok szerkesztése

A menüben „” ikonnal van jelölve, és csakis adminisztrátorok érhetik el. Ennél az oldalnál ugyanolyan formában jelennek meg adatok, mint a „ Szakok szerkesztése” opciónál, csak itt az adatbázisban lévő karok vannak megjelenítve, és a kar egyedi azonosítóját, valamint a kar nevét tartalmazza a táblázat. Az oldal tetején az „ Új kar hozzáadása” gomb a korábban említett oldal azonos funkciójú gombjával megegyező eredményt produkál.

2.1.10 Határidők szerkesztése

„” ikonnal van a menüben, és csak adminisztrátorok érhetik el. A „ Szakok szerkesztése” oldalhoz hasonló megjelenítést kapnak az adatok. Itt is a szakok töltődnek be egy listába, de itt minden listaelemnél van egy dátum mező, amiben a szakdolgozat beadására vonatkozó határidő van feltüntetve. Ezt a mezőre kattintva tudjuk szerkeszteni egy dátumkiválasztó ablakban.

Szakdolgozat leadási határidők szerkesztése



Szak megnevezése	Határidő																																																
anglisztika	2021. ápr. 8. 23:00:00 																																																
biológia	2021. máj. 5. 0:00:00 																																																
csecsemő- és kisgyermeknevelő	2021. máj. 22. 16:00:00 																																																
emberi erőforrások	2021. ápr. 15. 15:30:00 																																																
fizika	<div><div><div>< április 2021. ></div><table><tr><td></td><td>H</td><td>K</td><td>Sze</td><td>Cs</td><td>P</td><td>Szo</td><td>V</td></tr><tr><td>13</td><td>29</td><td>30</td><td>31</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>14</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td></tr><tr><td>15</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td></tr><tr><td>16</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td></tr><tr><td>17</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>1</td><td>2</td></tr></table></div><div><div>Óra</div><div>></div><div>13</div><div>14</div><div>15</div><div>16</div><div>17</div><div><</div></div></div>		H	K	Sze	Cs	P	Szo	V	13	29	30	31	1	2	3	4	14	5	6	7	8	9	10	11	15	12	13	14	15	16	17	18	16	19	20	21	22	23	24	25	17	26	27	28	29	30	1	2
	H	K	Sze	Cs	P	Szo	V																																										
13	29	30	31	1	2	3	4																																										
14	5	6	7	8	9	10	11																																										
15	12	13	14	15	16	17	18																																										
16	19	20	21	22	23	24	25																																										
17	26	27	28	29	30	1	2																																										
földtudományi																																																	
gazdálkodási és																																																	
gépészmérnök																																																	
germanisztika																																																	
gyógypedagógia																																																	
informatikus kön																																																	

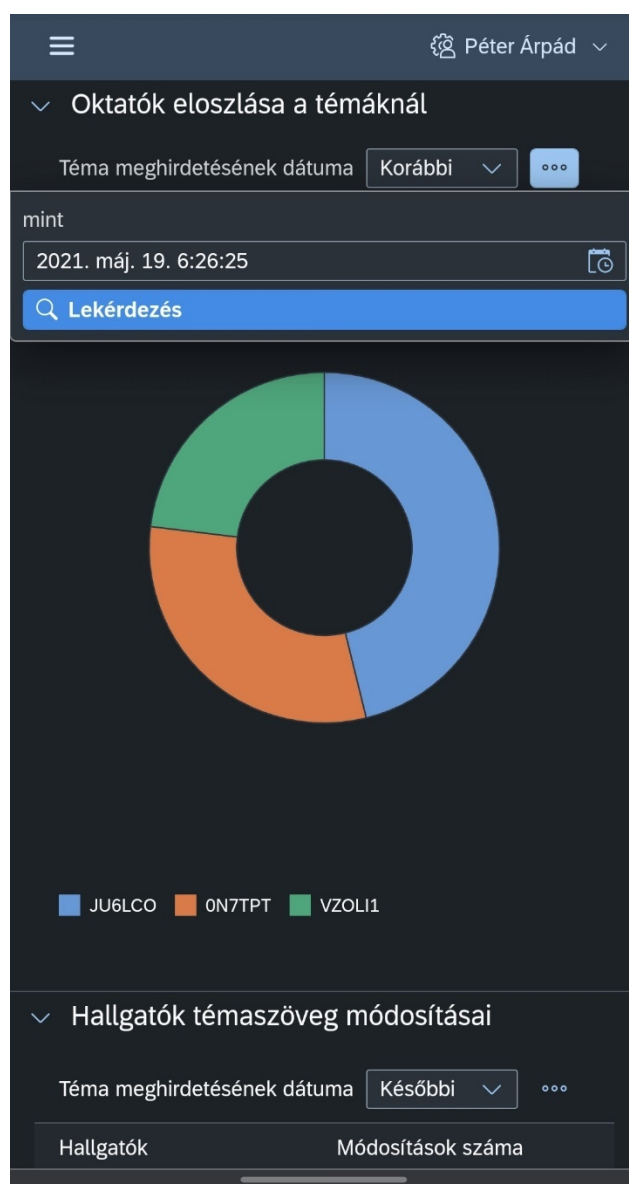
OK

Mégse

2.15. ábra. Egy határidő szerkesztése, világos témával.

2.1.11 Statisztikák

Ez az oldal kizárólag adminisztrátorok által elérhető, „” ikonnal található a menüben. Itt a különböző statisztikai mérések, diagramok és táblázatok foglalnak helyet, jelenleg csak kettő. Az egyik az oktatók témákhoz kapcsolódását összegző kördiagram, ahol meg lehet adni egy időpontot, és hogy azt megelőzően vagy azt követően nézzük, hogy egy oktató hány témánál szerepel, mint (társ)oktató. A másik a hallgatók témaszöveggel kapcsolatos módosításait összegző táblázat. Itt is meg lehet adni, hogy mely időpont előtt vagy után nézzük a szerkesztéseket. Mindkét kimutatást a „ *Lekérdezés*” gombra kattintva tudjuk frissíteni.



2.16. ábra. Statisztika oldalról egy részlet, mobil nézet, sötét témával.

3 Fejlesztői dokumentáció

3.1 Az applikáció telepítése

A webes applikációt az SAP Business Technology Platformjára (SAP BTP, az SAP Cloud Platform már nem létezik, mint különálló egység, hanem ennek a része) készítettem el, és a Platformba integrált Cloud Foundry (röviden CF) szolgáltatásról fut. Alapjában véve ezeket a szolgáltatásokat az SAP pénzért cserébe kínálja, viszont lehet kérni egy évig próbaidőszakot. Az alábbiakban bemutatom, hogy hogyan kell elvégezni a telepítést ebben a környezetben, amennyiben a birtokunkban vannak a forráskódok és konfigurációs fájlok.

- Ahhoz, hogy a Cloud Foundry környezetbe tudjuk telepíteni az alkalmazást, kell, hogy legyen egy terület (angolul space) az egyik SAP BTP-s végponton. Ha még nem rendelkezünk térrel, regisztrálnunk kell a <https://www.sap.com/> weboldalon, majd a <https://account.hanatrial.ondemand.com/> oldalon bejelentkezést követően az „Enter Your Trial Account” gombra kattintva be kell állítani egy régióhoz kötött alfiókot (angolul subaccount), és ha ez megvan, az alfiókra kattintva az „Enable Cloud Foundry” gombbal létrehozunk egy úgynevezett organisationt (magyarul szervezet) az általunk megadott aldomainen. Itt látható az az API végpont is, amin keresztül be tudunk lépni a Cloud Foundryba, és menedzselni tudjuk az ott létrehozott szolgáltatásainkat vagy applikációinkat. Végezetül kattintsunk a „Create Space” gombra, és adjuk meg, hogyan szeretnénk hívni a space-t.
- Az alkalmazás használatához kell egy adatbázis. Célszerű a Platformon létrehozni egy SAP HANA Cloud adatbázis példányt (angolul instance). Ehhez rá kell kattintani az előző lépésben létrehozott space-re, majd a bal oldali menüből a „SAP HANA Cloud” pontnál a „Create Database” gombra kattintva egy varázsló segítségével létrehozhatjuk az adatbázist. Ha ezzel megvagyunk, indítsuk el a HANA-t.
- Telepítsük a Cloud Foundry CLI-t (Command Line Interface – parancssori interfész) a számítógépükre a <https://docs.cloudfoundry.org/cf-cli/install-go-cli.html> oldalról, ügyelve arra, hogy a számítógépünkön futó operációs rendszernek megfelelő változatot válasszunk ki. Mivel nem egy manifest.yaml

fájlból történik a telepítés a CF környezetre, hanem `mta.yaml`-ből, ezért ehhez kell egy kiegészítő (plugin) a CF CLI-hez, a multiapps. Ezt a `cf install-plugin multiapps` parancs futtatásának segítségével le is tölthetjük. Ezután már nem csak a `cf push`, hanem a `cf deploy` parancs is elérhető a CLI-ből.

- Az applikáció telepítését elvégző eszköz letölti az applikációhoz társított Node.js csomagokat. Ehhez le kell tölteni a Node.js nevű futtató környezetet, valamint az ehhez alapértelmezettként beállított Node Package Managert (röviden npm), amit a <https://nodejs.org/en/download/> oldalról tehetünk meg. Ügyeljünk arra, hogy az operációs rendszerünknek megfelelő változatot válasszuk ki, illetve az LTS (long-term support, vagyis hosszútávon támogatott) verzió is megfelelő.
- Telepítsük a Cloud MBT-t (Multitarget Application Build Tool) a számítógépünkre a <https://github.com/SAP/cloud-mta-build-tool/releases> oldalról, vagy a korábban letöltött npm-et használva az `npm install -g mbt` paranccsal.
- Az MBT a Make nevű build rendszert is használja. A `make -v` parancs futtatásával tudjuk ellenőrizni, hogy fent van-e már a segédprogram. Amennyiben nincs, akkor Debian/Ubuntu esetén az `apt-get install make` parancs futtatásával telepíthetjük, Windows rendszerre pedig letölthetjük a <http://gnuwin32.sourceforge.net/packages/make.htm> oldalról, vagy a MinGW-t (Minimalist GNU for Windows) is telepíthetjük. Utóbbi esetben a bin könyvtárat, előbbinél pedig magát a make programot hozzá kell adni a PATH környezeti változóhoz.
- Nyissunk egy parancssori ablakot a forráskódok gyökérkönyvtárában (ahol az `mta.yaml` fájl található), és indítsuk el a build folyamatot az `mbt build` paranccsal.
- A build végén elkészül egy úgynevezett mta archive, egy `.mtar` kiterjesztésű fájl az `mta_archives` mappában. Lépünk be a Cloud Foundryba a `cf login` paranccsal, ehhez szükségünk van az első lépésben keletkezett API végpontra. Töltsük fel az imént generált `.mtar` fájlt a `cf deploy` paranccsal.
- Sikeres deploy után már csak egy identitásszolgáltatót kell az SAP BTP-hez kapcsolni, ami a felhasználókat tartalmazza. Az általam is használt Microsoft Azure Active Directoryval való összeköttetés a „3.3.9 Felhasználó azonosítása, jogok kiosztása” résznél olvasható.

3.2 Applikáció futtatása

A Cloud Foundryra történő sikeres telepítés végeztével elindulnak a szükséges szolgáltatások. A helyes működéshez az adatbázis szolgáltatás mellett alábbi applikációknak kell futni:

- app
- CAPapp-srv
- CAPapp-user

Ha valamiért leálltak, vagy akár a `cf stop` paranccsal leállítottuk, akkor a `cf start <app neve>` paranccsal el tudjuk indítani, és a parancssorból kiadott `cf apps` utasítással láthatjuk is, mely alkalmazások milyen elérhetőségen futnak. A keletkezett listából válasszuk ki az „app” nevű sor „routes” oszlopában lévő mezőjét, és írjuk be a böngészőbe. Ezen a linken keresztül érhető el az alkalmazás.

Az adatbázis szolgáltatást az SAP BTP Cockpiton keresztül is elérhető SAP HANA Cockpitban tudjuk elindítani, vagy a `cf update-service <adatbázis szolgáltatás neve> -c "{\"data\": {\"serviceStopped\":false}}` parancs kiadásával.

Ha már beléptünk az SAP BTP-be, akkor a korábban bejelentkezett fiókot fogja az alkalmazás használni. Fontos, hogy csak az alkalmazással kompatibilis identitásszolgáltatón keresztül használjuk a Témamenedzsert. Ha az oldal a „*Hiba a beazonosítás során*” kezdetű hibát írja ki, akkor más szolgáltatón keresztül kell bejelentkezni.

3.3 Felhasznált technológiák

3.3.1 SAP Business Technology Platform

Az SAP BTP a korábban SAP Cloud Platformnak nevezett szolgáltatás kiterjesztése. Nagyon sok kisebb SAP szolgáltatást gyűjt egy csokorba, és olyan területeken kíván kiemelkedően teljesíteni, mint az adatbázisok és adatbázis-kezelés (angolul database and data management), adat analitika, üzleti intelligencia (angolul business intelligence, BI), és ami a számomra a legfontosabb volt, applikáció fejlesztés és integrálás. Az SAP BTP nem kifejezetten a Google Cloud Platform, Microsoft Azure vagy esetleg Amazon AWS

vetélytársa, ugyanis a BTP teljesen elkülöníthető az alatta működő SAP felhő infrastruktúrától, és ezáltal mint egy összetett szolgáltatás, ezeken a platformokon is elérhető. A felhasználása erősen üzleti célokra van kitalálva, például olyan vállalatok részére, akik nagy mennyiségű adatokból szeretnének értékeket kinyerni valamilyen adatanalitikai vagy adatbányászati eljárásokkal, de a platform mégis tartalmazza azokat az alapvető építőelemeket, amik egy felhő alapú applikáció elkészítéséhez kellenek, így ilyen célokra is lehet használni, és a szakdolgozatban bemutatott szakdolgozati témamenedzser erre bizonyítékkal szolgál.

3.3.2 Cloud Foundry (CF)

Ez egy nyílt forráskódú (angolul open-source) platformszolgáltatás (angolul Platform as a Service, PaaS), ami különböző típusú applikációk és szolgáltatások telepítését és futtatását teszi lehetővé konténer alapokon úgy, hogy az elérhető legyen szélesebb körben is. Ennek egy enyhén módosított változata van integrálva az SAP BTP-ba,

Nagy előnye a CF-nek, hogy platformszolgáltatásként nem kell a fejlesztőknek az előre telepített (angolul on-premise) megoldásokkal ellentétben a szerverektől, tárhelyektől és hálózati beállításoktól kezdve az operációs rendszereken és futási környezeteken át a web- és adatbázis kiszolgáló konfigurálásáig mindenről gondoskodnia, ugyanis ezt megteszi a Cloud Foundry, tehát nagyban támogatja a teljes fejlesztési ciklust. Persze vannak helyzetek, amikor a fejlesztőnek mégis el kell végeznie bizonyos beállításokat például a futási környezeteknél, ezért bizonyos részek a közösen elérhetőek és irányíthatóak a felhőszolgáltatóval. Ez az irányítás a 3.1-es „Az applikáció telepítése” bekezdésben is említett parancssori interfészen keresztül érhető el.

3.3.3 SAPUI5

Ez egy olyan keretrendszer, amit felhasználói felületek (angolul user interface, UI) készítésére lehet használni, és az SAP nevű cég licenceli. A szintén SAP által fejlesztett úgynevezett Fiori dizájn irányelvekre épülve valósít meg reszponzív felületet úgy, hogy előre elkészített komponenseket tartalmaz, amiket fejlesztés során fel lehet használni, egymásba lehet ágyazni.

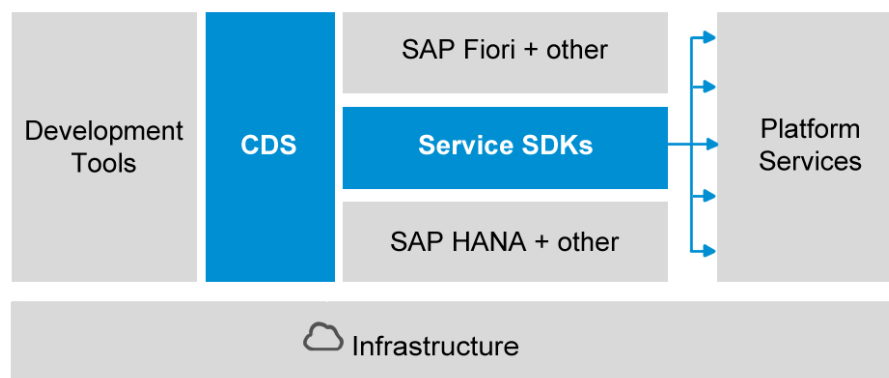


3.1. ábra. UI5 logó.

Nagy előnye, hogy a legalapvetőbb kezelőfelületi elemektől kezdve az egészen összetett elemekig konzisztens, ergonomikus és könnyedén alkalmazható UI-t kínál. A modell-nézet-vezérlő koncepcióban történő fejlesztést támogatja a leginkább. Több verziója is elérhető, folyamatos fejlesztés alatt áll, az alkalmazásom a jelenlegi legújabbat (1.89) támogatja. A SAPUI5 HTML5, CSS és JavaScript alapokon működik.

3.3.4 CAP

Az SAP Cloud Application Programming (CAP) modell az SAP cég által fejlesztett különböző nyelvekből, könyvtárakból és eszközökből álló keretrendszer felhő alapú technológiák használatához. Központi elemét képezi számos Cloud Foundryt használó SAP BTP szolgáltatásnak, mint például a HANA vagy a Fiori.



express



3.2. ábra. CAP által használt technológiák.

3.3.5 CDS

Az SAP CAP gerincét képező Core Data Service (CDS) egy univerzális, adatbázisfüggetlen leírónyelv. Számos dolgot el lehet végezni vele, a CRUD (create, read, update, delete – létrehozás, írás, módosítás, törlés) műveletektől kezdve a hitelesítésig (authentication).

A Témamenedzser db mappájában található a `data-model.cds`, ami egy CDS modell. Ebben vannak leírva azok az entitások, amiknek létezniük kell az adatbázisban, valamint meg vannak adva a mezőik, és a különböző táblák közti kapcsolatok.

A namespace `app.data`; kóddal a gyakorlatban „`app_data_`” prefixet tettem minden táblanévhez. Létrehoztam egy `CommonProps` aspectet is, ami minden típusú felhasználó (oktató, hallgató, adminisztrátor) legalapvetőbb mezőit (vezeték- és keresztnév, neptun kód, email) tartalmazza. Amikor a felhasználók entitásait kibővítjük ezzel az aspecttel, akkor hozzáadódnak az aspectben szereplő mezők. Léteznek az SAP által előre megadott aspectek is, mint például a `cuid`, ami egy UUID (Universally Unique Identifier, egy kötőjelekkel együtt 32 karakter hosszú, időbélyeget, véletlen számokat és MAC címet is felhasználó egyedi azonosító) típusú ID nevű kulcsot ad hozzá az entitáshoz, a managed aspect pedig a `createdBy`, `createdAt`, `modifiedBy`, `modifiedAt` mezőket, amiknél az utóbbi kettő minden módosításnál frissül.

A kapcsolatok az `Association` kulcsszóval szerkeszthetők, például a `Theses` nevű témákat tartalmazó tábla és a `Tags` nevű témákhoz rendelhető címkéket tartalmazó tábla közötti sok-sok kapcsolatot az alábbi módon lehet kivitelezni:

```
entity Tags : cuid {
  |   name : String;
  |
}
entity Theses : cuid, managed {
  |   //...
  |   tags : Association to many Tags2Thesis on tags.thesis = $self;
  |   //...
  |
}
entity Tags2Thesis {
  |   key tag : Association to Tags;
  |   key thesis : Association to Theses;
  |
}
```

3.3. ábra. Kapcsolatok megadása CDS használatával.

A CDS modellt a `cds build` paranccsal lehet buildelni, ahol meg lehet adni, hogy mivé legyen leképezve a modell, például SQL, JSON, HDBTABLE, stb. Az így generált fájlokat a `cds deploy` paranccsal lehet az adatbázisra felrakni. Amikor az alkalmazást lefordítjuk az `mbt build` paranccsal, akkor az automatikusan elvégzi ezeket a lépéseket.

A CDS nyelvet használva megadtam a `DataService` nevű service-t, ami OData protokollon keresztül elérhetővé tesz minden `data-model.cds`-ben lévő entitást, valamint a `StudentCount` nevű, kizárólag olvasható entitást, ami a `count` mezőjében tartalmazza az eddig feljelentkezett hallgatók számát és a `Theses` tábla valamennyi mezőjét. A gyakorlatban az entitások elérhetővé tétele az adatbázisban „*DataService_*” prefixű nézettáblák létrehozásával jár. Az entitások láthatósága és szerkeszthetősége függ attól, hogy a felhasználó milyen jogosultsági csoportba tartozik, valamint szerkesztésnél más szempontok is képbe jönnek.

3.3.6 OData

Az Open Data Protocol (OData) egy REST (Representational State Transfer) alapú API (Application Programming Interface). Legfrissebb és a Témamenedzser által is használt verziója az OASIS által standardizált v4.0. Az SAPUI5 és a CAP szorosan támogatják az adatok ezirányból történő elérését, és a Témamenedzserben is így, a `<kiszolgáló>/data/<OData lekérdezés>` úton lehet elérni a `DataService`-t.

Az API egy olyan interfész, ami meghatározza, hogyan lehet kapcsolatba lépni egy szoftver (jelen esetben az adatbázis) által kínált utasításkészlettel. A REST architektúrában a kliensek – jelen esetben – HTTP protokollon keresztül érik el a szerver ezen részeit úgy, hogy a kérésben lévő HTTP metódus (GET, POST/PATCH, DELETE, stb.) felel meg az elvégzendő CRUD műveletnek, az URI (Uniform Resource Identifier) mutatja meg, hogy mit kívánunk elérni, a header részben van a várt válasz formátuma, engedélyek és hitelesítő adatok, stb., a body részben pedig esetlegesen lehet küldeni úgynevezett payloadot, vagyis itt található mondjuk egy POST kérésnél az új érték.

A szerver szintén HTTP protokollon keresztül válaszol. A válasz elején van a státuszsor, ami a kérés eredményét összegzi (például `2xx` státuszkóddal sikeres volt a kérés, `4xx` kóddal a kliens, míg `5xx` kóddal a szerver oldal miatt hiba keletkezett), a headerben a

szerverrel kapcsolatos információk vannak, míg a bodyban a payload, vagyis például egy lekérdezett tábla mezői vannak.

A REST API állapotmentes, tehát a szerver nem tárol semmilyen információt a kliensekről. A kliensek tárolnak minden állapotot, és minden kérésnél a kliensek elegendő információt adnak a szervernek, hogy ki tudja őket szolgálni. Ez megbízhatóvá és jól skálázhatóvá teszi a szerveret.

Az OData protokollban az alábbi általam is használt lekérdezések is megadhatóak:

Lekérdezés	Leírás
\$select	Egy SQL SELECT utasításnak megfeleltethető.
\$filter	Egy SQL WHERE utasításnak megfeleltethető.
\$orderby	Egy SQL ORDER BY utasításnak megfeleltethető
\$count	Visszaadja az eredményben szereplő sorok számát.
\$expand	Mély olvasást végez a kapcsolt táblákban.
\$top, \$skip	Lapozással ellátott eredményeket ad vissza.
Lambda kif.	Egy gyűjteményen elvégzett lambda kifejezésnek nevezett logikai művelet.

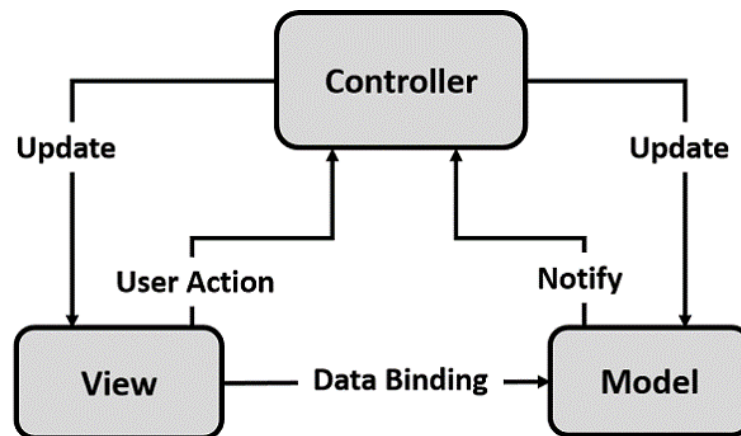
3.3.7 MVC architektúra

A modell-nézet-vezérlő (angolul model-view-controller, MVC) egy szoftvertervezési minta, melynek során az adatmodell úgy különül el a felhasználóval közvetlen kapcsolatban álló nézettől, hogy egy vezérlőt iktat közbe, ami lekezeli a nézetben kiváltott eseményeket, és elvégzi a modelleken az esetleges változtatásokat.

A modell az adatkezeléssel hozható egy kontextusba, menedzseli, frissíti és visszaadja a kívánt adatokat a vezérlőnek vagy a nézetnek. A modelleknek sok fajtáját lehet alkalmazni, én OData, JSON és `HttpContext.ResourceModel` modelleket használtam.

A nézet felelős a megjelenítésért (angolul presentation layer), és frissül a vezérlő által, esetleg a modellben bekövetkezett változások miatt, ha bizonyos adatokra adatkötéssel (angolul data binding) hivatkozunk. A nézeteket sokféleképpen el lehet készíteni, én XML alapú leírásokat készítettem, és ismertetőjelük a `*.view.xml` kiterjesztés a fájlnev végén.

A vezérlő elválasztja a megjelenítés logikáját az adatlogikától. Ez a réteg válaszol a felhasználói interakcióra úgy, hogy annak megfelelően szabályozza a nézetet és a modellt. A vezérlők csak JavaScriptben íródhatnak, és legfőbb jelük a `*.controller.js` kiterjesztés a fájlnev végén. Egy vezérlő általában egy nézethez tartozik, de több különböző nézet is használhatja, sőt akár más vezérlőbe is lehet meghívni funkciókat.



3.4. ábra. SAPUI5-ban használt MVC architektúra kapcsolatokkal

3.3.8 Node.js

Ez egy platformfüggetlen, nyílt forráskódú JavaScript futtató környezet (angolul runtime environment), ami a szerveroldali (angolul backend) mikroszolgáltatásokat futtatja a Google Chromium Project-en belül fejlesztett V8-as motor segítségével. Rengeteg nyílt, harmadik féltől származó Node.js program elérhető, ezeknek a kezelését pedig alapértelmezetten az npm (Node Package Manager) látja el. Ez egy CLI, aminek segítségével a node repositoryból telepíthetjük a kívánt Node programot. A konfigurációk, függőségek és futtatandó parancsok a package.json fájlban vannak, a package-lock.json fájlokban pedig egy adott verzióhoz lehet rögzíteni a csomagokat.

A Témamenedzser három futtatandó backend részből áll, a gyökér-, user és app mappákban kell kiadni az `npm start` parancsot (ezt megcsinálja a CF is).

3.3.9 Felhasználó azonosítása, jogok kiosztása

Kulcsfontosságú eleme az alkalmazásnak, hogy ki milyen felhasználóként kíván belépni, és hogy milyen jogok járnak neki. Az SAP terminológiájában ezt a folyamatot angolul user account and authorization (UAA) névvel illetik. Mivel az SAP BTP támogatja az OAuth 2.0 protokollal történő hitelesítést, úgy döntöttem, hogy az azonosítást egy külső résztvevő bevonásával, egy identitásszolgáltató (angolul identity provider, IdP)

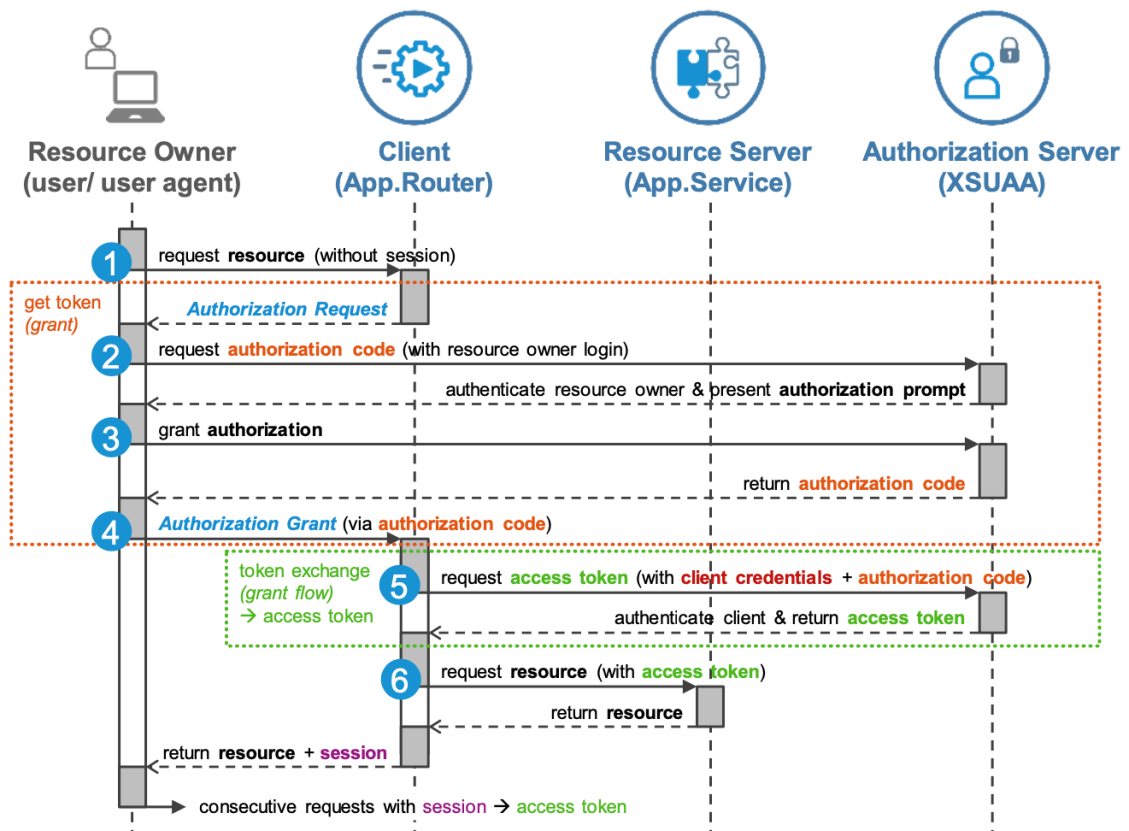
segítségével végzem el. A Microsoft Azure Active Directoryt (Azure AD) választottam erre a célra, mivel biztonságosnak, megbízhatónak és könnyen kezelhetőnek tartom. Az Azure AD-hez kapcsolódó felületen lehet létrehozni felhasználókat, beállítani jogosultságokat és egyéb más tulajdonságokat, mint például a név és a neptun kód.

A pontosabb működés megértéséhez az alábbi fogalmak bevezetése szükséges:

- Azonosítás (angolul identification): Ez alatt azt lehet érteni, hogy megmondjuk az applikációnak, kik is volnánk. Ez az alkalmazásomnál a neptun kódot tartalmazó email megadásával történik.
- Hitelesítés (angolul authentication): Itt alá is támasztjuk, hogy kik vagyunk, azzal, hogy megadunk egy olyan információt, amit jobb esetben csak mi tudunk (például jelszó), vagy felmutatunk valamit, ami egyedi és csak nekünk van (ilyen például az ujjlenyomat). Az alkalmazásom SAML 2.0 protokollt használ erre a célra.
- Engedélyezés (angolul authorization): Miután a két fenti tevékenység sikeresen végbement, az engedélyezés során dől el, hogy a bejelentkezett személy az applikáció mely részeihez fér hozzá, ezen ellenőrzés zajlik OAuth 2.0 protokollal.

Az SAML (Security Assertion Markup Language) 2.0 XML formátumú nyílt szabvány, ami az IdP (Azure AD) és egy szolgáltató (angolul service provider, jelen esetben az applikációm) közti hitelesítéssel kapcsolatos adatokat továbbítja. A legnagyobb jelentősége az egyszeri bejelentkezés (angolul single-sign-on, SSO) megvalósításában van, tehát hogy egyetlen bejelentkezéssel hozzáférjen a felhasználó a hozzá kapcsolt erőforrásokhoz és ne kelljen az alkalmazáson belül ismételt hitelesítenie magát.

Az OAuth 2.0 egy nyílt szabvány jogosultságok továbbításához. Az UAA ez alapján ad ki tokeneket, amik JSON Web Token (JWT) formátumúak. Ezek biztonságosan továbbítják az információkat a résztvevők között. Az információk megbízhatósága azon alapszik, hogy az engedélyezést végző szolgáltatás (UAA) a saját privát kulcsával digitálisan aláírta a tokenet, és így a hitelesség bármikor ellenőrizhető.



3.5. ábra. Jogosultságok kiosztásának folyamata.

A mechanizmus felállításához szükséges, hogy az SAP BTP subaccount felületéről a bal oldali menü „Security” részén belüli „Trust Configuration” opciót választva felállítsunk egy megbízható kapcsolatot az IdP és az SAP BTP között. Ehhez le kell tölteni a Platform SAML metaadatait („[↓ SAML Metadata](#)”), majd ezt az Azure AD-n belül létrehozott „SAP Cloud Platform” nevű vállalati alkalmazásának „Single sign-on” menüpontjánál lévő SAML résznél fel kell tölteni („[Upload metadata file](#)”). Ezután az AD ugyanezen felületéről kell letölteni az SAML metaadatokat („[Federation metadata XML download](#)”), és ezeket feltölteni a BTP korábban megnyitott felületén lévő „[↑ New Trust Configuration](#)” gomb megnyomása után megjelenő ablakba.

Az IdP-nek biztosítania kell, hogy az alkalmazás megkapja a kívánt információkat, mint például a felhasználó neve, email címe, rangja, a kar és a szak, amihez kapcsolódik. Ezek az információk attribútumként továbbítódnak, és az alábbi módon kell kinézniük:

Attribútum neve	Leírás
nameidentifier	Ez lesz az SAP BTP-n belül a felhasználó egyedi azonosítója és email címe. Az Azure AD-ban lévő felhasználó email címét adtam meg, ami tartalmazza a neptun kódot is.
first_name	A felhasználó keresztnéve.
last_name	A felhasználó vezetéknéve.
Groups	A felhasználó csoportjai.
mail	A felhasználó email címe.
userrole	A felhasználó rangja. Lehetséges értékek: Student, Teacher, Admin. Azure AD-n belül a „beosztást” (user.jobtitle) adtam meg továbbítandó értéként.
userfaculty	A kar, amin a felhasználó van. Az alkalmazáshoz társított adatbázisban lévő karok egyedi azonosítói lehetnek a lehetséges értékei, Azure AD-n belül a „cég nevét” (user.companyname) adtam meg. Hallgatóknak és oktatóknál szükséges megadni.
usercourse	A szak, amin a felhasználó van. Az alkalmazás adatbázisában lévő szakok egyedi azonosítóit veheti fel értéként, Azure-on belül pedig a „részleget” (user.department) továbbítja. Csak hallgatóknál kell megadni.

Fontos az is, hogy a frissen létrehozott felhasználó az Azure AD-n belül hozzá legyen rendelve ahhoz a vállalati alkalmazáshoz, amit az imént kapcsoltunk össze az SAP BTP-vel.

Ha már elvégeztük a „3.1 Az applikáció telepítése” részt, akkor kell léteznie egy XSUAA szolgáltatásnak, ami a konfigurációnak megfelelően létrehozott egy „Rank” nevű jogosultságot, amit a „capapp” nevű jogosultsági gyűjteménybe (Role Collection) szervezett. Ez a kollekció lesz közvetlenül a felhasználókhöz kapcsolva. A „Rank” jogosultságot még be kell állítani az SAP BTP/subaccount/Security/Roles menüpontban, hogy az IdP-ből továbbított értékekkel egyezzen. Amennyiben az IdP beállítását a fentebb

lévő táblázatnak megfelelően végeztük el, akkor az alábbi pároknak kell szerepelniük a jogosultságban:

Faculty	userfaculty
Course	usercourse
Rank	userrank

Amikor bejelentkezünk a Témamenedzser applikációba az IdP-n keresztül, akkor generálódik egy JWT token. Ehhez bizonyos funkciók elérése miatt hozzá kellett férnem, viszont biztonsági okok miatt erre a célra egy külön Node.js alkalmazást (mikroszolgáltatást) készítettem, ami hozzáfér a tokenhez, és visszaadja a szükséges információkat JSON formátumban. Ennek az JavaScript alkalmazásnak a neve CAPapp-user, és a Témamenedzseren belül a UserAttributes és UserInfo modellekben találhatóak meg a kinyert adatok.

3.3.10 Approuter

Az applikáció útválasztója (angolul röviden approuter) egy egységes belépési pontot biztosít minden, az alkalmazáshoz tartozó mikroszolgáltatáshoz. Ez egy Node.js csomag, ami a beérkező kéréseket továbbítja a megfelelő mikroszolgáltatáshoz, jogosultságokat ellenőriz, ha úgy van, bejelentkezteti a felhasználót, valamint kiszolgálja a statikus tartalom iránti kérélmeket.

Az xs-app.json fájlban keresztül lehet konfigurálni, és mivel Node.js alapú, a package.json tartalmazza az indításához szükséges parancsot, valamint az esetlegesen többi csomagtól való függőséget.

Az approuteren belül van a felhasználói felületnek is a helye. A UI különböző oldalai, valamint a navigációhoz kapcsolódó beállítások a `manifest.json` fájlban vannak megadva.

A Témamenedzserben az alábbi útvonalak léteznek:

Elérési út	Leírás
/static/logout.html	Kijelentkezési oldal (nem igényel azonosítást az elérése).
/data/	OData adatok lekérdezése.
/user/	Felhasználóról szóló információk elérése.
/ui/webapp/index.html	Kezdőoldal.
<i>Kezdőoldalt követően:</i>	
/base	Témák megtekintése.
/detail/{id}	Az {id} azonosítójú téma részleteinek megtekintése.
/viewThesesText	Témákhoz beadott szövegek megtekintése.
/viewKeywords	Kulcsszavak megtekintése.
/viewUsers	Felhasználók megtekintése.
/viewStats	Statisztikák és diagramok megtekintése.
/submitThesis	Témához szöveg beadása.
/editFaculties	Karok megtekintése és szerkesztése.
/editCourses	Szakok megtekintése és szerkesztése.
/editTheses	Témakiírások szerkesztése.
/editDeadlines	Beadási határidők megtekintése és szerkesztése.
/authFail	Sikertelen beazonosítást jelző oldal.
/permissionFail	Jogosultsági elégtelenséget jelző oldal.

Nem létező oldal felkeresése esetén egy speciális, `notFound` nevű oldal töltődik be.

Bár a felhasználói felület csak olyan oldalakat jelenít meg a felhasználónak, aminek az eléréséhez elegendő jogosultsággal bír, a fenti táblázatban lévő direkt linkek segítségével más, nem a saját jogosultsági körének megfelelő tartalmat is elérhetne. Ennek kiküszöbölése érdekében a jogosultság ellenőrzése minden megjelenítéskor megtörténik a kontroller által. Ha nem lehetett a jogosultságot leellenőrizni valami miatt, akkor a `/authFail` oldalra visz a router. Ha sikerült az ellenőrzés, és nem elégséges a

felhasználó jogosultsága, akkor a `/permissionFail` oldalra kerül. Megfelelő jogosultság esetén használhatóvá válik a kiválasztott oldal.

3.3.11 MTA

A multi target application (MTA) egyetlen alkalmazás, ami több különböző technológiával készült részből (modulból) áll. Egy ilyen alkalmazást az MTA modell határoz meg, ami platformfüggetlenül, YAML leírónyelven mondja meg, hogy milyen modulok, milyen paraméterekkel és függőségekkel rendelkezzenek. Ahhoz, hogy telepíteni (angolul *deploy*) lehessen az MTA-t, le kell fordítani (angolul *build*) az `mta.yaml` fájlt. Ezt a „3.1 Az applikáció telepítése” résznél említett Multitarget Application Build Tool (MBT) tehetjük meg. A build eredménye egy `.mtar` (MTA archive) fájl lesz, amiben tömörítve vannak a futtatáshoz szükséges állományok valamint a telepítést leíró fájlok. Ezt már lehet *deploy*olni a Cloud Foundry környezetre, ahol a CF automatikusan megszervezi és elvégzi a kívánt modulok telepítését, illetve azok összekapcsolását.

3.3.12 i18n

Az angol *internationalization* szóból ered ez a rövidítés (18 betű távolság van a szó elején és végén lévő betűk között), magyarul egyfajta honosítást, nyelvi lokalizációt jelent. Ez az informatikai eszközöknél azt jelenti, hogy ugyanaz a szoftver képes az adott régióknak megfelelően megjeleníteni nyelvet és mértékegységeket. Ezt a koncepciót a SAPUI5 támogatja, számos UI elem alapról rendelkezik ilyen tulajdonsággal. Az oldalon megjelenő szöveget is lehetséges oly módon dinamikussá tenni, hogy létrehozunk egy különleges *i18n* modellt a `Component.js` fájlban, valamint egy `i18n.properties` fájlt a `webapp/i18n` mappájában. Az `i18n.properties` név-érték párokat tartalmaz, a névre lehet a kódban hivatkozni, és a hozzá tartozó érték kerül majd megjelenítésre. Így igazán egyszerűen lehet új nyelveket hozzáadni az applikációhoz, hiszen a kódban semmit sem kell módosítani.

Minden országnak és régióknak megvan a maga nemzetközi rövidítése, ezeket szuffixumként alkalmazva csinálhatunk új `.properties` kiterjesztésű fájlokat, és azokban ugyanazon nevek mellé a lefordított értéket kell beírni. Például az `i18n_hu.properties` a magyar, az `i18n_en.properties` az angol nyelv észlelése esetén töltődik majd be.

A Témamenedzser jelenleg csak a magyar nyelvet támogatja.

3.3.13 BAS fejlesztői környezet

Az SAP Business Application Studio (BAS) az azt megelőző WebIDE következő generációja, ami egy modern fejlesztői környezet (angolul integrated development environment, IDE). Az „*Eclipse Theia*” nevű nyílt forráskódú projekten alapszik, ami a Microsoft Visual Studio Code nevű kódszerkesztőjének felhasználói élményét és belső működésének egy részét is átveszi. Ezt használtam a fejlesztést során, mivel támogatja valamennyi SAP Cloud szolgáltatást (SAP HANA adatbázis menedzsment, SAPUI5 nézetek, CDS modellezés stb.), VS Code által ismerős a UI, és megvan benne minden, ami a teljeskörű fejlesztéshez (angolul full stack development) kell. Ez magába foglalja a kliens (angolul frontend) és szerver (angolul backend) oldalt.

A Studion belül úgynevezett Dev Space-eket lehet létrehozni, amik testreszabhatók egy-egy területtel (például SAP Mobile Application, SAP Fiori, vagy amit én választottam, Full Stack Cloud Application) kapcsolatban, de alapjaiban mindegyik ugyanazt tudja. A Dev Space-ek Kubernetes (másnéven K8S, konténer alapú munkafolyamatok menedzselésére használt platform) pod-ok (a legkisebb K8S-re telepíthető egység), tehát van fájlrendszere, terminál CLI stb.

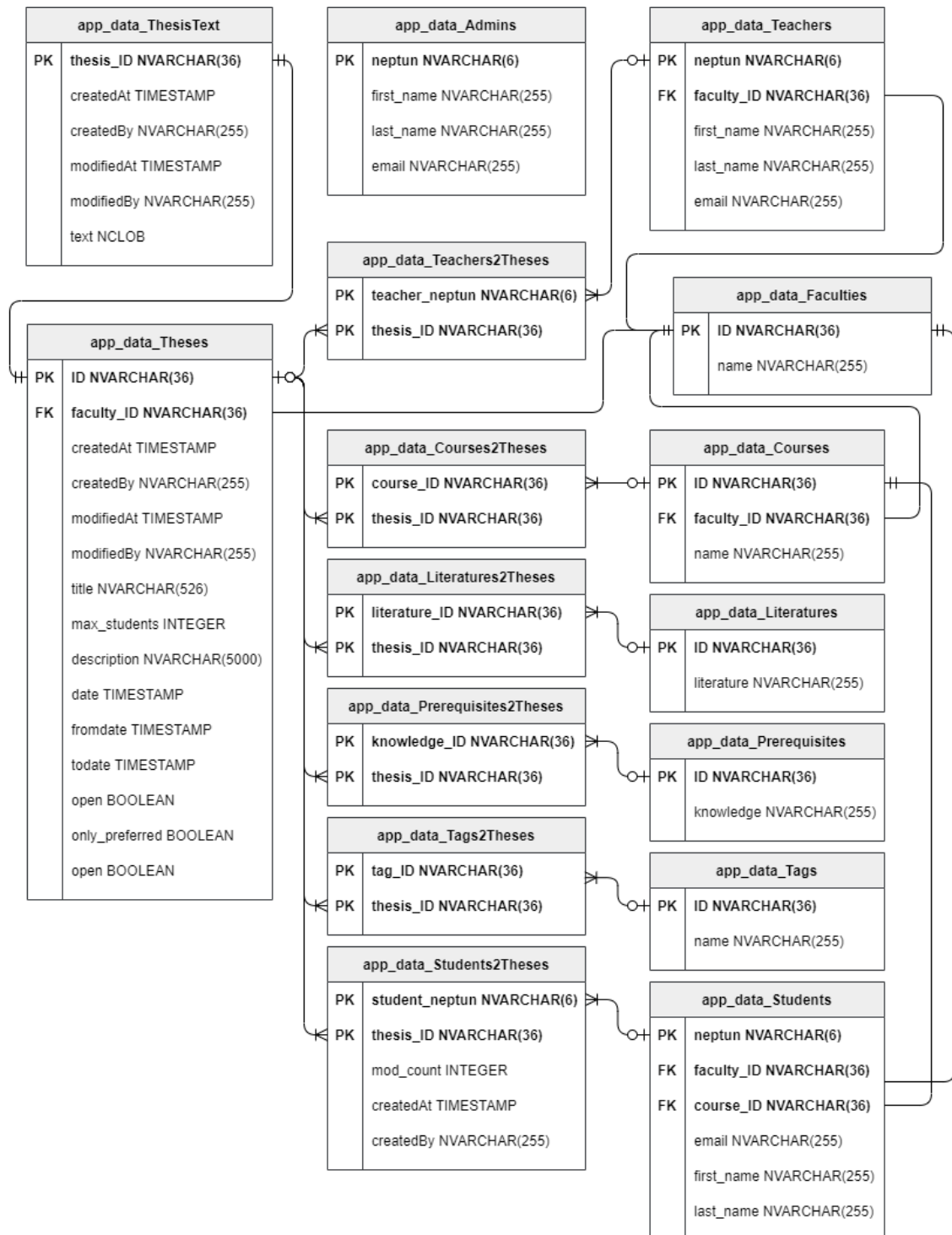
A BAS-ban történő fejlesztéssel járó tesztelés közben szükséges volt futtatni a Node.js programokat, akár a beépített debugger vagy a terminálból kiadott parancsok által. A futtatáshoz szükséges volt más, Cloud Foundry futó szolgáltatások elérése is, például a HANA adatbázisé és az XSUAA engedélyezőé. Ahhoz, hogy létrejöjjön a kapcsolat a külső szolgáltatásokkal, a mikroszolgáltatásokon belül a `default-env.json` fájlokban van megadva a csatlakozáshoz szükséges információ. Ez a fájl nem szükséges a Cloud Foundry telepítéskor, hisz abban a környezetben a telepítő automatikusan összekapcsolja a szolgáltatásokat.

3.3.14 Git verziókezelés

A webapplikációt a fejlesztés alatt a Githubon vezetem, a módosítások rendszeres committal lettek nyugtázva, illetve kiírásra került pár issue is. Ehhez össze kellett kötni a Business Application Studiot a Githubbal, a legegyszerűbbnek pedig az tűnt, hogy a Githubon belül generálok egy tokenet, amivel bizonyos jogosultságokkal felruházom a BAS-t. Ezzel a tokenel egybefűzve klónoztam a git repositoryt BAS-ra, majd onnan végeztem el a push műveleteket.

3.4 Adatmodell

Az alább látható az adatbázis tábláit, mezőit és az azok közti kapcsolatokat leíró adatmodell diagram. A PK az elsődleges (angolul primary), az FK a külső, idegen (angolul foreign) kulcsot jelöli.



3.6. ábra. Adatbázis adatmodell diagram.

A diagramban a mezők típusát a HANA adatbázisban használt típusokként adtam meg. Ezek jelentése az alábbi táblázatban látható:

Típus	Leírás
NVARCHAR	Unicode karaktereket tartalmaz, mérete a zárójelen belül van meghatározva.
BOOLEAN	Logikai érték, igaz vagy hamis.
INTEGER	Egész szám.
TIMESTAMP	Dátumot és időpontot tartalmaz.
NCLOB	Nagy mennyiségű unicode karaktereket tartalmaz.

3.5 Táblák

A DataService miatt ugyanezen prefixszel megtalálható minden az alábbiakban felsorolt tábla nézettáblaként létrehozva, valamint egy StudentCount nevű tábla is, ami a Theses táblából vett ID, title, max_students, faculty_ID, open, fromdate és todate mezőkön túl egy extra count mezővel is rendelkezik, amiben az éppen a témára feljelentkezett hallgatók száma van kiszámolva.

A DataService-ben elérhető négy, helyben megvalósított, SQL lekérdezések eredményeit visszaadó függvény: StudentsStats(), TeachersStats(), StudentsStatsWithDate(date, operator), TeachersStatsWithDate(date, operator). A date egy CDS DateTime típus, az operator paraméter pedig két karakter, lehetséges értékei: lt, gt. Ezek 0..n darab FIRST_NAME, LAST_NAME, NEPTUN, VALUE sorozatot adnak vissza eredményül. Csak adminisztrátor használhatja őket, hibás paraméterezés esetén 400-as hiba lesz a válaszban.

A gyakorlatban a CDS modellben meghatározottaknak megfelelően a táblák nevei előtt „app_data_” prefix van, ezt most az egyszerűség kedvéért elhagytam.

3.5.1 Theses

A szakdolgozati témákat tartalmazó tábla.

Mezőnév	Leírás
ID	A szakdolgozati témakiírás egyedi azonosítója (UUID), elsődleges kulcs, nem lehet NULL.
title	A téma címe.
description	A téma leírása.
max_students	A témára feljelentkezhető hallgatók maximális száma. Alapértelmezetten az értéke 1.
date	A téma meghirdetésének időpontja.
fromdate	A téma elérhetőségének kezdődátuma.
todate	A téma elérhetőségének befejező dátuma.
open	A téma állapotát (elérhető/lezárt) határozza meg. Alapértelmezetten hamis.
only_preferred	Meghatározza, hogy csak olyan szakról lehet-e a témára jelentkezni, ami szerepel az ajánlott szakok között, vagy bármilyenről. Alapértelmezetten hamis.
archived	Meghatározza, hogy archiválva van-e a téma, vagy sem. Alapértelmezetten hamis.
faculty_ID	Idegen kulcs, ami a <code>Faculty</code> tábla ID mezőjére mutat. Megadja, mely karhoz kapcsolódik a szakdolgozati téma.
createdAt	A téma létrehozásának dátuma.
createdBy	A téma létrehozójának email címe.
modifiedAt	A téma legutóbbi módosításának dátuma.
modifiedBy	A téma legutóbbi módosítójának email címe.

3.5.2 ThesisText

A szakdolgozati témákhoz kapcsolt szövegezéseket tartalmazza.

Mezőnév	Leírás
thesis_ID	A szöveg azonosítója, megegyezik a hozzá tartozó téma egyedi azonosítójával, elsődleges kulcs, nem lehet NULL.
text	A szövegezés.
createdAt	A szövegezés létrehozásának dátuma.
createdBy	A szövegezés létrehozójának email címe.
modifiedAt	A szövegezés legutóbbi módosításának dátuma.
modifiedBy	A szövegezés legutóbbi módosítójának email címe.

3.5.3 Admins

A már valaha bejelentkezett adminisztrátorokat tartalmazza.

Mezőnév	Leírás
neptun	Az adminisztrátor neptun kódja, egyedi azonosítóként szolgál, elsődleges kulcs, nem lehet NULL.
first_name	Az adminisztrátor keresztnéve.
last_name	Az adminisztrátor vezetéknéve.
email	Az adminisztrátor email címe.

3.5.4 Teachers

A már valaha bejelentkezett oktatókat tartalmazza.

Mezőnév	Leírás
neptun	Az oktató neptun kódja, egyedi azonosítóként szolgál, elsődleges kulcs, nem lehet NULL.
first_name	Az oktató keresztnéve.
last_name	Az oktató vezetéknéve.
email	Az oktató email címe.
faculty_ID	Idegen kulcs, ami a Faculty tábla ID mezőjére mutat. Megadja, mely karon van az oktató.

3.5.5 Students

A már valaha bejelentkezett hallgatókat tartalmazza.

Mezőnév	Leírás
neptun	A hallgató neptun kódja, egyedi azonosítóként szolgál, elsődleges kulcs, nem lehet NULL.
first_name	A hallgató keresztnéve.
last_name	A hallgató vezetéknéve.
email	A hallgató email címe.
faculty_ID	Idegen kulcs, ami a <code>Faculty</code> tábla ID mezőjére mutat. Megadja, mely karon van a hallgató.
course_ID	Idegen kulcs, ami a <code>Course</code> tábla ID mezőjére mutat. Megadja, mely szakon van a hallgató.

3.5.6 Faculties

A karokat tartalmazza.

Mezőnév	Leírás
ID	A kar egyedi azonosítója (UUID), elsődleges kulcs, nem lehet NULL.
name	A kar elnevezése.

3.5.7 Courses

A szakokat tartalmazza.

Mezőnév	Leírás
ID	A szak egyedi azonosítója (UUID), elsődleges kulcs, nem lehet NULL.
name	A szak elnevezése.
deadline	A szakdolgozat beadására vonatkozó határidő az adott szakon.
faculty_ID	Idegen kulcs, ami a <code>Faculty</code> tábla ID mezőjére mutat. Megadja, mely karon van az adott szak.

3.5.8 Literatures

A témákhoz ajánlott irodalmakat tartalmazza.

Mezőnév	Leírás
ID	Az irodalmi mű egyedi azonosítója (UUID), elsődleges kulcs, nem lehet NULL.
literature	Az irodalmi mű megnevezése.

3.5.9 Prerequisites

A témákhoz szükséges előismereteket, előzetesen megszerzett tudást tartalmazza.

Mezőnév	Leírás
ID	Az előismeret egyedi azonosítója (UUID), elsődleges kulcs, nem lehet NULL.
knowledge	Az előismeret megnevezése.

3.5.10 Tags

A témákhoz rendelt címkéket tartalmazza.

Mezőnév	Leírás
ID	A címke egyedi azonosítója (UUID), elsődleges kulcs, nem lehet NULL.
name	A címke elnevezése.

3.5.11 Teachers2Theses

A Theses és Teachers táblák közötti sok-sok kapcsolatot kialakító kapcsolótábla.

Mezőnév	Leírás
teacher_neptun	A Teacher tábla neptun mezőjére mutat. Az oktató neptun kódja, az elsődleges kulcs első fele, nem NULL.
thesis_ID	A Theses tábla ID mezőjére mutat. A téma egyedi azonosítója, az elsődleges kulcs második fele, nem NULL.

3.5.12 Students2Theses

A Theses és Students táblák közötti sok-sok kapcsolatot kialakító kapcsolótábla.

Mezőnév	Leírás
student_neptun	A Student tábla neptun mezőjére mutat. A hallgató neptun kódja, az elsődleges kulcs első fele, nem NULL.
thesis_ID	A Theses tábla ID mezőjére mutat. A téma egyedi azonosítója, az elsődleges kulcs második fele, nem NULL.
mod_count	A témaszöveg adott hallgató általi módosításainak számát mutatja, alapértelmezetten 0.
createdAt	A feljelentkezés létrehozásának időpontja.
createdBy	A feljelentkezést létrehozó felhasználó email címe.

3.5.13 Courses2Theses

A Theses és Courses táblák közötti sok-sok kapcsolatot kialakító kapcsolótábla.

Mezőnév	Leírás
course_ID	A Courses tábla ID mezőjére mutat. A szak egyedi azonosítója, az elsődleges kulcs első fele, nem NULL.
thesis_ID	A Theses tábla ID mezőjére mutat. A téma egyedi azonosítója, az elsődleges kulcs második fele, nem NULL.

3.5.14 Literatures2Theses

A Theses és Literatures táblák közötti sok-sok kapcsolatot kialakító kapcsolótábla.

Mezőnév	Leírás
literature_ID	A Literatures tábla ID mezőjére mutat. Az ajánlott irodalom egyedi azonosítója, az elsődleges kulcs első fele, nem NULL.
thesis_ID	A Theses tábla ID mezőjére mutat. A téma egyedi azonosítója, az elsődleges kulcs második fele, nem NULL.

3.5.15 Prerequisites2Theses

A Theses és Prerequisites táblák közötti sok-sok kapcsolatot kialakító kapcsolótábla.

Mezőnév	Leírás
knowledge_ID	A Prerequisites tábla ID mezőjére mutat. Az előismeret egyedi azonosítója, az elsődleges kulcs első fele, nem NULL.
thesis_ID	A Theses tábla ID mezőjére mutat. A téma egyedi azonosítója, az elsődleges kulcs második fele, nem NULL.

3.5.16 Tags2Theses

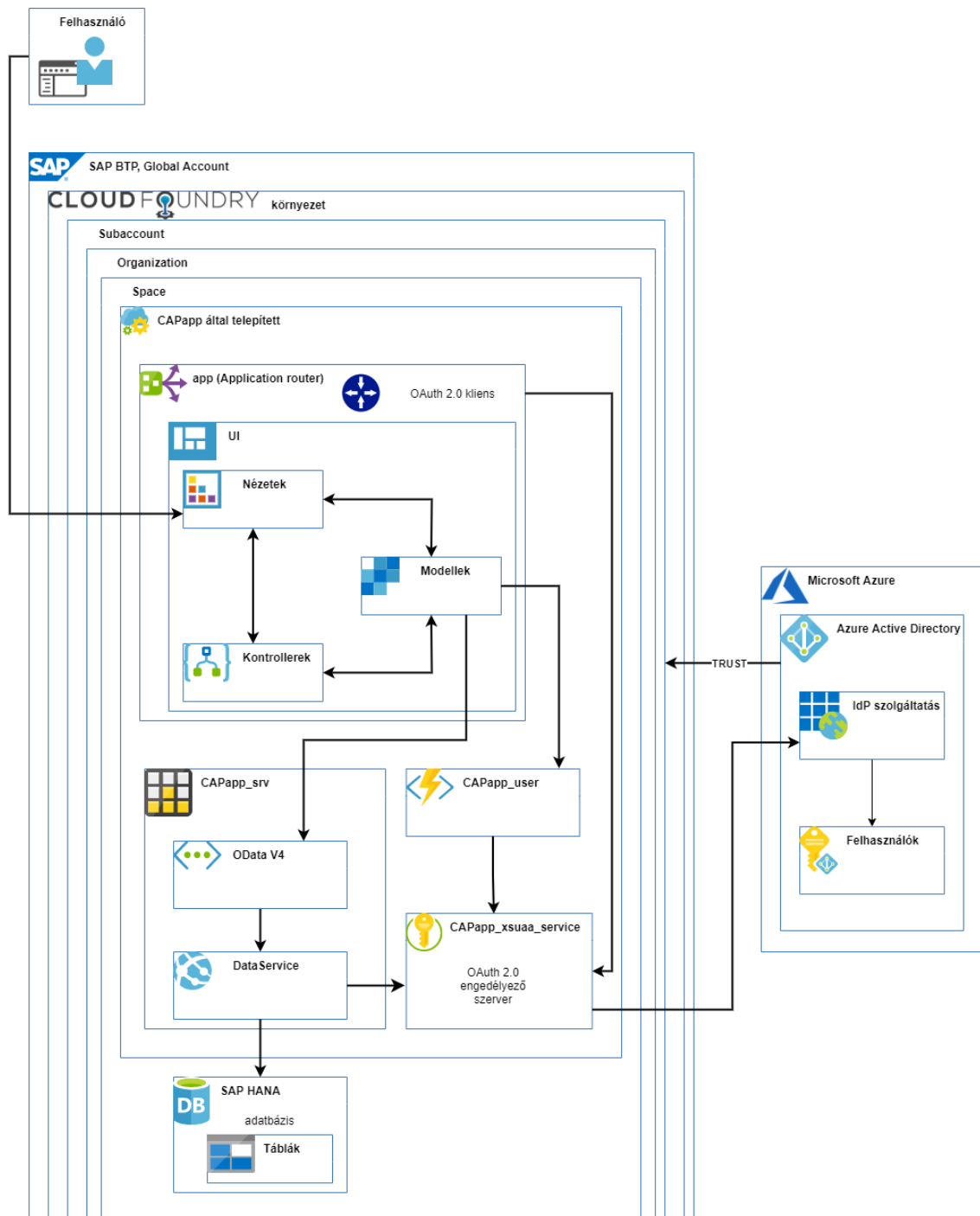
A Theses és Tags táblák közötti sok-sok kapcsolatot kialakító kapcsolótábla.

Mezőnév	Leírás
tag_ID	A Tags tábla ID mezőjére mutat. A címke egyedi azonosítója, az elsődleges kulcs első fele, nem NULL.
thesis_ID	A Theses tábla ID mezőjére mutat. A téma egyedi azonosítója, az elsődleges kulcs második fele, nem NULL.

3.6 Architektúra

Az architektúra diagramon látható, hogy hogyan is néz ki az egész rendszer, amiben a Témamenedzser (CAPapp projektnév alatt) benne van. A felhasználó közvetlenül a nézetekkel van kapcsolatban (az ehhez szükséges azonosítási és engedélyezési folyamatokról a „3.3.9 Felhasználó azonosítása, jogok kiosztása” résznél lehet olvasni). A modellek az OData API szolgáltatáson keresztül érik el az adatbázist, valamint a CAPapp_user mikroszolgáltatástól kapják meg az IdP bejelentkezés után átadott attribútumokat és felhasználói adatokat. Az OData kérések ellenőrzésére szolgáló DataService közvetlen kapcsolatban áll az adatbázissal. A DataService, a CAPapp_user és az app nevű approuter a CAPapp_xsuaa_service nevű engedélyezettési szolgáltatástól kéri le azokat az adatokat, amik az IdP szolgáltatásban történő azonosítás után keletkeztek. Az IdP a Microsoft Azure platformon fut, és egy megbízhatóság van kialakítva az Azure Active Directory és a Cloud Foundry környezetben lévő Subaccount között.

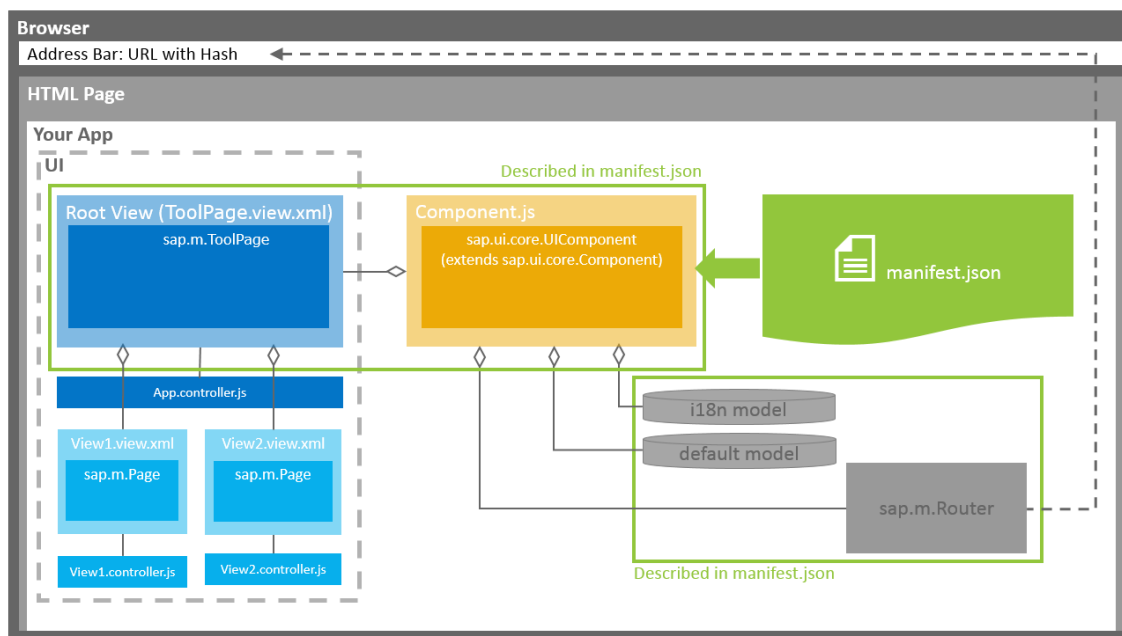
A diagramban csak a legrelevánsabb modulokat ábrázoltam, tekintve, hogy nagyon komplex platformokról van szó.



3.7. ábra. Architektúra diagram.

3.7 MVC részek

A Témamenedzser felhasználói felülete a „3.3.7 MVC architektúra” részben részletesen kifejtett Model-View-Controller mintára épül. A működést a leíró fájl (manifest.json), a komponens (component.js) és a gyökér nézet (ToolPage.view.xml) teszi lehetővé. Az applikáció jogosultság ellenőrzést követő működése az alábbi sablon szerint történik:



3.8. ábra. Az oldal általános működése.

A következőkben az MVC konkrét részeit mutatom be.

3.7.1 Modellek

- **Alapértelmezett modell:** A gyakorta használata miatt egyszerűbbnek tűnt név nélkül tartani. Ez a modell implementálja az OData V4 API szolgáltatást.
- **device:** A webappot használó böngészőről, eszközről információkat tároló JSON modell. Az `sap.ui.device` API-tól kapja az információkat.
- **i18n:** Egy speciális erőforrás, a nyelvi lokalizációért felelős `i18n.properties` fájl modellként történő reprezentálása.
- **UserAttributes:** Egy JSON modell, ami a CAPapp_user mikroszolgáltatásból tartalmazza a felhasználó beosztását, hallgatónál a szak egyedi azonosítóját, hallgatónál és oktatónál pedig a kar egyedi azonosítóját.
- **UserInfo:** Egy JSON modell, ami a CAPapp_user mikroszolgáltatásból tartalmazza a felhasználó nevét és email címét.

3.7.2 Nézetek

A nézetek `.view.xml` fájlvégződéssel bírnak és más nézetekkel egy szintben helyezkednek el. Az úgynevezett fragmenteknek `.fragment.xml` fájlvégződésük van, és lehetséges őket más oldalak vagy felhasználói felületek felett elhelyezni, például felugró ablakként. Nézetekből és fragmentekből az alábbiak léteznek:

Név	Leírás	Kontroller
AuthFail	Sikertelen beazonosítást jelző oldal.	AuthFail
Base	A Master és Detail oldalakat összekapcsoló láthatatlan oldal.	Base
Detail	Egy téma részletes információit megjelenítő oldal.	Detail
EditCourses	Szakok megjelenítésére és szerkesztésére biztosított oldal.	Faculty-CourseEdit
EditDeadlines	Szakedolgozat leadására vonatkozó szakonkénti határidők megjelenítésére és szerkesztésére biztosított oldal.	EditDeadlines
EditFaculties	Karok megjelenítésére és szerkesztésére biztosított oldal.	Faculty-CourseEdit
EditTheses	Témák megjelenítésére és szerkesztésére biztosított oldal.	EditTheses
Keywords	A kulcsszavakat és szófelhőt megjelenítő oldal.	Keywords
Stats	Diagramokat és statisztikákat megjelenítő oldal.	Stats
Master	A nem archív témákat megjelenítő oldal.	Master
NotFound	Nem létező oldalt jelző oldal.	-
PermissionFail	Elégtelen jogosultságot jelző oldal.	-
SubmitThesis	Szakedolgozat beadására biztosított oldal.	SubmitThesis
ToolPage	A menüt és a felső sávot tartalmazó oldal. Minden más oldal ennek a konténerében van.	ToolPage
Users	Felhasználókat megjelenítő oldal.	Users
ViewText	Felhasználóhoz kapcsolódó témák szövegezését megjelenítő oldal.	ViewText
Welcome	Nyitóoldal.	Welcome

Change	A szakdolgozati téma szerkesztését lehetővé tevő fragment.	EditTheses
Display	A szakdolgozati téma szerkesztő oldalon a témáról részletes információkat megjelenítő fragment.	EditTheses
EditCourseDialog	Egy kiválasztott szak szerkesztését lehetővé tevő fragment.	Faculty-CourseEdit
EditFacultyDialog	Egy kiválasztott kar szerkesztését lehetővé tevő fragment.	Faculty-CourseEdit
SearchValue-HelpDialog	Címkék közti keresést és kiválasztást lehetővé tevő fragment.	Master
WordCloud-Settings	Szófelhő beállításait megjelenítő fragment.	Keywords

3.7.3 Kontrollerek

A kontrollerek `.controller.js` fájlvégződéssel rendelkeznek és a kliensnél hajtódnak végre. A `sap.ui.define` API használatával jönnek létre ezek a modulok, és a `sap.ui.core.mvc.Controller` alosztályaiként vannak megadva. Gyakorta használt horog az `onInit`, ami a controller betöltésekor fut le. Az alábbi kontrollerek léteznek:

- **AuthFail.** Megvalósított metódusok:
 - `onLogoutPress`: Lekezeli a kijelentkezés gombra kattintást.
- **Base.** Megvalósított metódusok:
 - `onInit`: Hozzáadja a `patternMatched` eseményhez az `_onObjectMatched` metódust.
 - `_onObjectMatched`: Beállítja az oldal felosztását egyoszloposra.
- **Detail.** Megvalósított metódusok:
 - `onInit`: Beállítja a modelleket, hozzáadja a `patternMatched` eseményhez az `_onObjectMatched` metódust.
 - `onApplyClearButtonPress`: Témáról történő lejelentkezés gomb megnyomását kezeli le.

- `onApplyButtonPress`: Témára történő feljelentkezés gomb megnyomását kezeli le. Megjelenít egy megerősítő ablakot, és annak elfogadása esetén meghívja a `_onApplyConfirmed` metódust.
- `onHandleClose`: Összecsukja a Detail oldalt.
- `_onObjectMatched`: Az URL hashben megadott témához tartozó információkra frissíti az oldal tartalmát.
- `_checkStudentCourseInCourses`: Megjelenít egy figyelmeztető üzenetcsíkot, ha a hallgatóhoz tartozó szak nincs a téma által támogatott szakok között.
- `_refreshS2TmodelWithBtnUpdate`: Frissíti a témához feljelentkezett hallgatókat tartalmazó modellt és meghívja a `_onUpdateButton` metódust.
- `_onApplyConfirmed`: Lekezeli a feljelentkezési kérelmet.
- `_onUpdateButton`: Megvizsgálja, milyen kinézetet kellene adni a fel- vagy lejelentkezés gombnak.
- `_setDetailButtons`: Elvégzi a fel- vagy lejelentkezés gomb kinézettel kapcsolatos beállításait.
- `_getStudentCountByThesis`: Beállítja az oldalhoz tartozó modellben a feljelentkezett hallgatók számát.
- `_getText`: A paraméterében megadott névhez lekéri a `i18n` modellben lévő értéket.
- **EditDeadlines.** Megvalósított metódusok:
 - `onInit`: Hozzáköti a megfelelő eseményekhez a jogosultság ellenőrzést.
- **EditTheses.** Megvalósított metódusok:
 - `onInit`: Beállítja a modelleket, hozzáköti a megfelelő eseményekhez a jogosultság ellenőrzést.
 - `onListItemPress`: A kiválasztott témához tartozó információkra frissíti az oldal tartalmát.
 - `onBackToMasterPress`: A mobil eszközökön megjelenített vissza gomb megnyomását kezeli le.
 - `onFacultyChanged`: A kar megváltoztatásakor hívódik meg, meghívja a `_validateInputsWithFaculty` metódust.

- `handleNewThesisPress`: Új témát létrehozó gomb megnyomását kezeli le. Megjelenít egy megerősítő ablakot, és annak elfogadása esetén meghívja a `_createNewThesis` metódust.
- `onSelectableTokenUpdate`: Hallgatókat, oktatókat és szakokat tartalmazó mezők módosítását kezeli le.
- `onTokenUpdate`: Követelményeket, ajánlott irodalmakat és címkéket tartalmazó mezők módosítását kezeli le.
- `onEnableMyInput`: Az egyedi követelmények, ajánlott irodalmak és címkék megadását lehetővé tevő gombok megnyomását kezeli le.
- `handleEditPress`: A szerkesztés gomb megnyomását kezeli le, meghívja a `_validateInputsWithFaculty` és `_toggleButtonsAndView` metódusokat, utóbbit igaz paraméterrel.
- `handleSavePress`: A szerkesztés befejezése gomb megnyomását kezeli le, meghívja a `_toggleButtonsAndView` metódust hamis paraméterrel.
- `_bindItemsToMasterList`: Az oktatóhoz kapcsolódó témák címeit a menühöz rendeli.
- `_createNewThesis`: Lekezeli az új téma létrehozása iránti kérelmet.
- `_validateInputsWithFaculty`: Megvizsgálja a karral és a hallgatók, oktatók és ajánlott szakok kompatibilitását és szükség esetén a beviteli mezőkhöz figyelmeztető üzenetet tesz.
- `_toggleButtonsAndView`: A paraméterben kapott értéktől függően kezdeményezi a „*Change*” vagy „*Display*” fragment betöltését a `_showFormFragment` metódus meghívásával.
- `_getFormFragment`: Betölti a paraméterként kapott fragmentet.
- `_showFormFragment`: Meghívja a `_getFormFragment` metódust és frissíti az oldal tartalmát. „*Change*” paraméter esetén meghívja a `_validateInputsWithFaculty` metódust.
- `_getText`: A paraméterében megadott névhez lekéri a `il8n` modellben lévő értéket.

- **FacultyCourseEdit.** Megvalósított metódusok:

- `onInit`: Hozzáköti a megfelelő eseményekhez a jogosultság ellenőrzést.
- `onListItemPress`: Lekezeli a listaelemre kattintást, meghívja a `_showEditDialog` metódust „*Courses*” vagy „*Faculties*” paraméterrel.

- `onCloseEditDialog`: Bezárja a szerkesztési fragmentet.
 - `onNewItemPress`: Új szakot vagy kart létrehozó gomb megnyomását kezeli le. Megjelenít egy megerősítő ablakot, és annak elfogadása esetén meghívja az `_addNewItem` metódust.
 - `_showEditDialog`: Betölti a paramétertől függően a szak vagy kar szerkesztését lehetővé tevő fragmentet.
 - `_addNewItem`: Lekezeli az új kar vagy szak létrehozása iránti kérelmet.
 - `_getText`: A paraméterében megadott névhez lekéri a i18n modellben lévő értéket.
- **Keywords.** Megvalósított metódusok:
 - `onInit`: Beállítja a modelleket, hozzáköti a megfelelő eseményekhez a jogosultság ellenőrzést.
 - `onWordcloudSelect`: A szófelhő készítése gomb megnyomását kezeli le. Meghívja a `_generateWordCloud` és paramétertől függően a `_selectKeywords` metódusokat.
 - `onShowSettingsDialog`: A szófelhő beállításait megjelenítő gomb megnyomását kezeli le, betölti az ezért felelős fragmentet.
 - `onCloseSettingsDialog`: Bezárja a szófelhő beállításaiért felelős fragmentet.
 - `onDefaultWordcloudSelected`: Alapértelmezett esetben ez kezeli le a szófelhő készítése gomb megnyomását. Meghívja a `_generateWordCloud` és `_selectKeywords` metódusokat.
 - `onBackToMasterPress`: A mobil eszközökön megjelenített vissza gomb megnyomását kezeli le.
 - `onListItemPress`: Lekezeli a listaelemre kattintást, meghívja az `onCloudDelete` és `_selectKeywords` metódusokat.
 - `onCloudDelete`: A szófelhő törlése gomb megnyomását kezeli le, törli a szófelhőt.
 - `_bindItemsToMasterList`: A felhasználóhoz kapcsolódó témák címeit a menühöz rendeli.
 - `_prepareText`: Előkészíti a paraméterként kapott szöveget a kulcsszókiválasztáshoz.

- `_selectKeywords`: Feltölti a kulcsszavakkal az ezeket tartalmazó modellt. Ehhez meghívja a `_prepareText` metódust.
 - `_generateWordCloud`: Megjeleníti vagy frissíti a szófelhőt. Ehhez meghívja a `_wordCloud` metódust.
 - `_wordCloud`: Létrehozza a szófelhőt. Ehhez a `d3-cloud` nevű Node.js modult használja.
 - `_getText`: A paraméterében megadott névhez lekéri a `il8n` modellben lévő értéket.
- **Master.** Megvalósított metódusok:
 - `onInit`: Hozzáköti a megfelelő eseményekhez a jogosultság ellenőrzést.
 - `onListItemPress`: Listaelemre kattintást kezeli le, átvisz a kiválasztott téma Detail oldalára.
 - `onComboBoxChange`: Az oktató, kart és szakot kiválasztó kezelőkben lévő érték változásakor hívódik meg, validálja a beírt értéket.
 - `onDatePickerChange`: Az dátumot kiválasztó kezelőben lévő érték változásakor hívódik meg, validálja a beírt értéket.
 - `onReset`: A keresőmezők értékét alaphelyzetbe állító gomb megnyomását kezeli le, törli a mezők értékét.
 - `onSearch`: A keresőmezők tartalma alapján elvégzi a keresést a témákat tartalmazó listán.
 - `handleValueHelp`: A címkék kiválasztásához megjelenít egy címkéket kilistázó fragmentet.
 - `_handleValueHelpSearch`: A címkéket kilistázó fragment tartalmát frissíti a fragment keresőmezőjébe beírt értéknek megfelelően.
 - `_handleValueHelpClose`: Bezárja a címkéket listázó fragmentet.
 - `_getText`: A paraméterében megadott névhez lekéri a `il8n` modellben lévő értéket.
 - **SubmitThesis.** Megvalósított metódusok:
 - `onInit`: Beállítja a modelleket, hozzáköti a megfelelő eseményekhez a jogosultság ellenőrzést.
 - `onCheckReceivedData`: A kiválasztott témát ellenőrző gomb megnyomását kezeli le, ennek megfelelően validálja a varázsló első lépést vagy hibákat jelző üzeneteket jelenít meg.

- `onEditTypeSelected`: A szerkesztési mód kiválasztása gomb megnyomását kezeli le, validálja a kiválasztott módot és ennek megfelelően a varázsló második lépést.
 - `bindDataToForm`: A varázsló harmadik lépésére vezető gomb megnyomásakor hívódik meg, betölti a kiválasztott témára vonatkozó információkat az azt tartalmazó modellbe.
 - `wizardCompletedHandler`: A varázsló harmadik lépését befejező gomb megnyomását kezeli le, átvizsgálja a felülvizsgálati oldalra.
 - `backToWizardContent`: Visszavisz a varázslóra a felülvizsgálati oldalról.
 - `editThesisText`: A felülvizsgálati oldalon található „*Szerkesztés*” linke kattintást kezeli le. Meghívja a `_handleNavigationToStep` metódust 3-as értékkel.
 - `onWizardCancel`: A varázslóból való kilépésért felelős gomb megnyomását kezeli le. Megjelenít egy megerősítő ablakot, és annak elfogadása esetén meghívja a `_discardWizardProgress` metódust.
 - `onWizardSubmit`: A témaszöveg beadását lehetővé tevő gomb megnyomását kezeli le. Megjelenít egy megerősítő ablakot, és annak elfogadása esetén feldolgozza a kérelmet.
 - `_bindItemsToSelect`: A hallgatóhoz kapcsolódó témák címeit a listához köti.
 - `_discardWizardProgress`: Alaphelyzetbe állítja a varázslót, törli a megelőző lépésekben lévő eredményeket.
 - `_handleNavigationToStep`: A paraméterben kapott számú oldalra navigál a felülvizsgálati oldalról. Meghívja a `backToWizardContent` metódust.
 - `_getText`: A paraméterében megadott névhez lekéri a `il8n` modellben lévő értéket.
- **ToolPage.** Megvalósított metódusok:
 - `onInit`: Beállítja a modelleket, hozzáköti a megfelelő eseményekhez a jogosultság ellenőrzést.
 - `onSideNavButtonPress`: A menüt jelző gomb megnyomását kezeli le, kinyitja vagy összecsukja a menüt.

- `handleMenuItemClick`: Az oldal jobb felső részén lévő személyes beállítások megnyomását kezeli le.
- `onItemSelect`: A menü elemeire kattintást kezeli le. A kiválasztott oldalra navigál.
- `updateTheme`: Beállítja a paraméterben kapott témát.
- `onLogoutPress`: Lekezeli a kijelentkezés gombra kattintást.
- `showInfoDialog`: Megjeleníti a felhasználóval kapcsolatos információkat tartalmazó dialógust.
- `_loadMenusAndCheckUser`: Megjeleníti a felhasználó jogosultságának megfelelő menüpontokat a menüben és frissíti a felhasználóval kapcsolatos információkat az adatbázisban.
- `_setToggleButtonTooltip`: A paraméterben kapott értéknek megfelelően frissíti a menüt jelző gomb eszköztippjét.
- `_getText`: A paraméterében megadott névhez lekéri a `il8n` modellben lévő értéket.
- **Users.** Megvalósított metódusok:
 - `onInit`: Hozzáköti a megfelelő eseményekhez a jogosultság ellenőrzést.
- **ViewText.** Megvalósított metódusok:
 - `onInit`: Beállítja a modelleket, hozzáköti a megfelelő eseményekhez a jogosultság ellenőrzést.
 - `onBackToMasterPress`: A mobil eszközökön megjelenített vissza gomb megnyomását kezeli le.
 - `onListItemPress`: A kiválasztott témához tartozó információkra frissíti az oldal tartalmát.
 - `_bindItemsToMasterList`: A felhasználóhoz kapcsolódó témák címeit a menühöz rendeli.
 - `_getText`: A paraméterében megadott névhez lekéri a `il8n` modellben lévő értéket.
- **Stats.** Megvalósított metódusok:
 - `onInit`: Elvégez a diagramon pár beállítást, hozzáköti a megfelelő eseményekhez a jogosultság ellenőrzést.
 - `onQueryButtonPress`: Lekezeli a „*Lekérdezés*” gomb megnyomását, elvégzi a paraméter alapján a kívánt lekérdezést és a kívánt UI elem frissítését.

- `onDatePickerChange`: Az dátumot kiválasztó kezelőben lévő érték változásakor hívódik meg, validálja a beírt értéket.
- `_getText`: A paraméterében megadott névhez lekéri a i18n modellben lévő értéket.
- **Welcome.** Megvalósított metódusok:
 - `onInit`: Beállítja a modelleket, hozzáadja a `ThemeChanged` eseményhez a `_changeLogoTheme` metódust, hozzáköti a megfelelő eseményekhez a jogosultság ellenőrzést.
 - `_loadUserInfo`: Betölti a felhasználóval kapcsolatos adatokat az ezeket tároló modellbe.
 - `_changeLogoTheme`: Módosítja az ELTE címer színvilágát a témának megfelelően.

3.8 Tesztelés

Az alkalmazáshoz csak a sikeres azonosítás után lehet hozzáférni. Ezt megelőzően minden kérés a bejelentkező oldalra viszi a felhasználót, erről az approuter gondoskodik.

Azonosítást követően a generált JSON Web Tokennel lehet csak kéréseket indítani. Ennek a tokennek az élettartama jelenleg 15 perc tétlenség után megszűnik, és ekkor újból be kell jelentkezni.

A következőkben az aktív sessionnel bíró felhasználók lehetséges tevékenységeit vizsgálom.

3.8.1 Szerveroldal

Az adatbázishoz az OData és DataService szolgáltatásokon keresztül érkehetnek kérések, és attól függően, hogy milyen metódussal érkezik be a HTTP kérés, az alábbi műveleteket kerülnek elvégzésre:

Művelet	Metódus
CREATE	POST
SELECT/READ	GET
UPDATE	PATCH
DELETE	DELETE

Egy bejelentkezett felhasználó nem fér hozzá minden OData végponthoz. Az alábbi táblázatban látható, hogy mely rangokkal (O – oktató, H – hallgató, A – Adminisztrátor, * – mindegyik, X – egyik sem) milyen metódus engedélyezett végpontokként:

Végpont	POST	GET	PATCH	DELETE
Theses	O	H, O	O	X
Courses	A	*	A	X
Faculties	A	*	A	X
Students	H	*	H	X
Teachers	O	*	O	X
Admins	A	*	A	X
ThesisText	H	H, O	H	X
Tags	O	H, O	O	X
Literatures	O	H, O	O	X
Prerequisites	O	H, O	O	X
Courses2Theses	O	O	X	O
Students2Theses	H, O	H, O	X	H, O
Teachers2Theses	O	H, O	X	O
Tags2Theses	O	H, O	X	O
Literatures2Theses	O	H, O	X	O
Prerequisites2Theses	O	H, O	X	O
StudentCount	X	H, O	X	X
függvények	X	A	X	X

Illetéktelen GET lekérdezéseknél 200-as státusszal tér vissza a kiszolgáló (konkrét kulcsra hivatkozva 404-es státusszal), de a válasz törzse nem fog adatot tartalmazni.

Az OData a lekérdezéseknél automatikusan elvégzi a típusellenőrzést, és nem megfelelő típus esetén minden esetben 400-as hibával tér vissza.

Szintén 400-as hibával tér vissza akkor, ha már egy létező azonosítóval bíró elemet szeretnénk ismét kreálni, vagy egy nem létező mezőt szeretnénk módosítani.

404-es hibával tér vissza, ha a kérés helyes, de nem létezik az elérni kívánt rekord.

Pár példa az imént említett esetekre:

Lekérdezés				Válasz	
Végpont	Metódus	Törzs	Rang	Státusz	Törzs
/Theses	GET	-	Admin	200	value: []
/Theses	GET	-	Oktató	200	value: [...]
/Theses(valid ID)	GET	-	Admin	404	-
/Theses(valid ID)	GET	-	Hallgató	200	{...}
/Theses(invalid ID)	GET	-	Hallgató	404	-
/Theses(valid ID)	PATCH	open : false	Admin	403	-
/Theses(valid ID)	PATCH	open : false	Oktató	200	{...}
/Theses(valid ID)	DELETE	-	Admin	403	-
/Students	GET	-	Admin	200	value: [...]
/Students	POST	neptun : abc	Admin	403	-
/Courses(valid ID)	PATCH	name : szak	Admin	200	{...}
/Courses(valid ID)	PATCH	name : szak	Oktató	403	-
/ThesesText	GET	-	Oktató	200	value: [...]
/ThesesText(valid ID)	POST	text : abc, id: valid ID	Oktató	403	-
/ThesesText(valid ID)	POST	text : abc, id: valid ID	Hallgató	200	{...}
/ThesesText(már létező ID)	POST	text : abc, id: létező ID	Hallgató	400	{error, Entity already exists.}

A backend az authorization headerben érkező token által megtudhatja, hogy a küldőnek mi az email címe (neptun kódot tartalmazza), rangja, bizonyos rangok esetén pedig, hogy mi a szak és a kar azonosítója. Ezáltal fel van készítve az alábbi biztonsági kockázatot jelentő kérések kezelésére is:

- Ha a /ThesisText végponton hallgató POST vagy PATCH kérést indít
 - és lejárt a szakán a beadási határidő, 467 „A szakon lejárt a beadási határidő.” hibát kap.
 - és a téma archivált, 468 „A témaszöveg nem szerkeszthető, mert a téma archivált.” hibát kap.

- és nincs feljelentkezve a témára, 466 „*Nincs hozzárendelve <neptun> a témához.*” hibát kap.
- Ha a /ThesisText végponton hallgató vagy oktató GET kérést indít, csak azoknak a témáknak a szövegeit kapja meg, amikre feljelentkezett és tartozik hozzá témaszöveg.
- Ha a /Students2Theses végponthoz hallgató vagy oktató, a /Teachers2Theses, /Literatures2Theses, /Courses2Theses, /Prerequisites2Theses végpontokhoz oktató POST vagy DELETE kérést indít, és a téma archiválva lett, 468 „*A téma nem szerkeszthető, mert a téma archiválva lett.*” hibát kap.
- Ha a /Students2Theses végponthoz hallgató egy nem archív témával POST vagy DELETE kérést indít
 - és nem egyezik a neptun kódja a kérésben lévővel, 469 „*Küldő <neptun1> és a kérésben lévő <neptun2> nem egyezik.*” hibát kap.
 - és a téma zárva van, 470 „*A témára nincs engedélyezve feljelentkezés.*” hibát kap.
 - és a téma elérhetőségi idején kívül van, 471 „*A jelentkezési idő letelt.*” hibát kap.
 - és a témánál be van állítva, hogy csak preferált szakokról lehet jelentkezni, de a hallgatóé nincs köztük, 472 „*Csak ajánlott szakról lehet jelentkezni.*” hibát kap.
 - és a témára feljelentkezettek száma legalább annyi, mint a férőhelyek száma, 473 „*A szabad helyek beteltek.*” hibát kap.
- Ha a /Theses végponthoz oktató PATCH kérést indít, de a téma archivált, 468 „*A téma nem szerkeszthető, mert a téma archiválva lett.*” hibát kap.
- Ha a /Students, /Teachers, /Admins végpontokhoz a megfelelő jogosultsággal egy felhasználó PATCH kérést indít, de a neptun kódja nem egyezik a kérésben lévővel, 469 „*Küldő <neptun1> és a kérésben lévő <neptun2> nem egyezik.*” hibát kap.

3.8.2 Felhasználói felület - használati eset tesztek

- **Témák oldal:**
 - Ha oktatóként vagy hallgatóként kiválasztunk egy témát, az applikáció átnavigál a téma információit taglaló oldalra.
 - Ha hallgatóként fel vagyunk jelentkezve a témára, megjelenik egy „*Lejelentkezés*” gomb, valamint egy üzenetcsík a feljelentkezés időpontjával.

- Ha hallgatóként nem vagyunk feljelentkezve, és az elérhetőségi időn kívül vagyunk, és/vagy beteltek a szabad helyek, és/vagy zárva van a téma, és/vagy csak ajánlott szakról lehet jelentkezni, de a miénk nincs közte, és/vagy már feljelentkeztünk egy témára, megjelenik egy inaktív „*Jelentkezés*” gomb.
- Ha hallgatóként nem vagyunk feljelentkezve, és a jelentkezésnek nincs akadálya, megjelenik egy kattintható „*Jelentkezés*” gomb.
- Ha megnyomjuk az aktív „*Jelentkezés*” gombot, felugrik egy megerősítést kérő ablak. Ha az „*Igen*” gombra kattintunk, elküldi a rendszer a feljelentkezési szándékot a backendnek, és sikeres feljelentkezés esetén egy erről tájékoztató ablak ugrik fel. Ha a „*Nem*” gombra kattintunk, nem történik semmi.
- Ha hallgatóként tekintjük a témát, és nem jelentkeztünk még fel témára, és a szakunk nem szerepel az ajánlott szakok között, megjelenik egy ezt jelző üzenetcsík.
- **Témaszövegek megtekintése oldal**
 - Ha egy olyan témát választunk ki, amihez még nincs szövegezés beadva, felugrik egy erről tájékoztató ablak.
- **Témaszövegek kulcsszavai oldal**
 - Ha egy olyan témát választunk ki, amihez még nincs szövegezés beadva, felugrik egy erről tájékoztató ablak.
 - Ha rákattintunk a „*Szófelhő kulcsszavakból*” vagy „*Szófelhő témaszövegből*” gombra, akkor a szándéknak megfelelően a „*Szófelhő beállításai*ban” megadott mennyiségű szót tartalmazva vagy kulcsszavakból, vagy a témaszövegből véletlenszerűen kiválasztva generálódik egy szófelhő.
 - Ha még nem hoztunk létre szófelhőt, vagy töröljük a meglévőt a beállításokban lévő „*Szófelhő törlése*” gombbal, át lehet állítani a szófelhő magasságát és hosszát. Ha már meg van jelenítve egy szófelhő, ez az opció inaktív.
 - A szófelhőben lévő szavak száma és a betűtípus megváltoztatása csak a következő szófelhő generálástól lesz érvényes.
- **Téma beadása oldal**
 - Ha elmúlt a szakon a beadási határidő és megnyomjuk az „*Ellenőrzés*” gombot, megjelenik egy ezen hibát jelző üzenetcsík.

- Ha a kiválasztott téma archív és megnyomjuk az „Ellenőrzés” gombot, megjelenik egy ezen hibát jelző üzenetcsík.
- Egyéb esetben, ha megnyomjuk az „Ellenőrzés” gombot, megjelenik a varázsló következő oldalára vezető gomb.
- A varázsló második oldalán, ha a „Szerkesztés” opciót választjuk, de a témához még nincs beadott szöveg, akkor ezt a hibát egy felugróablak jelzi. Ha tartozik beadott szöveg a témához, akkor bármely opció kiválasztása után megjelenik a varázsló következő oldalára vezető gomb.
- A varázsló harmadik oldalán az előző lépésben kiválasztottaknak megfelelően vagy megjelenik a már beadott szöveg, vagy teljesen üres a szövegbeviteli mező.
- Az „Ellenőrzés” gombra kattintva a felülvizsgálati oldalra kerülünk.
- A felülvizsgálati oldalon a szövegmező melletti „Szerkesztés” gombra kattintva visszakerülünk a varázsló harmadik oldalára.
- A felülvizsgálati oldalon a „Jóváhagyás” gomb megnyomásával felugrik egy megerősítő ablak. Ha az „Igen” gombra kattintunk, elküldi a rendszer a témaszöveg beadási szándékot a backendnek, és sikeres beadás esetén egy erről tájékoztató ablak ugrik fel. Ha a „Nem” gombra kattintunk, nem történik semmi.
- A varázsló minden oldaláról elérhető a „Kilépés” gomb. Megnyomása esetén egy megerősítést kérő ablak ugrik fel. Ha az „Igen” gombra kattintunk, figyelmen kívül lesz hagyva az eddigi változtatás és a varázsló legelejére ugrunk. Ha a „Nem” gombra kattintunk, nem történik semmi.

- **Témák szerkesztése oldal**

- Ha kiválasztunk egy témát, aktívvá válik a „Szerkesztés” gomb.
- Ha az „Új téma kiírása” gombra kattintunk, létrehozunk egy „Új téma” nevű témát, aminek a kara meg fog egyezni a miénkkel, a meghirdetés és elérhetőség dátumai a jelenlegiek lesznek, és alapértelmezetten mi leszünk hozzáadva a témához, mint kapcsolódó oktatók.
- Ha a „Szerkesztés” gombot megnyomjuk, megjelenik a szerkesztési felület, és a „Szerkesztés” gomb helyett „Szerkesztés vége” gomb jelenik meg.
- Ha a „Szerkesztés vége” gombot megnyomjuk, akkor visszatérünk az adatokat megjelenítő felületre.

- Ha a kiválasztott téma archiválva van, akkor az archiválást módosító kapcsolón kívül minden elem inaktívvá válik.
- Ha a „★” gombot megnyomjuk, a hozzá tartozó mezőbe új, saját értékeket is felvihetünk.
- Ha a dátumokhoz nem valid értéket írunk be, azok a mezők nem mentődnek el.
- Ha az ajánlott szakoknál, kapcsolódó tanárok és feljelentkezett hallgatóknál olyan elemet adunk meg, ami nem ahhoz a karhoz kapcsolódik, ami a témához ki van választva, akkor ezt a beviteli mezőnél egy sárga háttérű üzenet jelzi.
- **Felhasználók megtekintése**
 - Ha az oldal tetején lévő kártyák közül változtatunk, az annak megfelelő rangú felhasználókat látjuk felsorolva.
- **Új szakok szerkesztése és Új karok szerkesztése oldalak**
 - Ha az oldal tetején lévő „Új szak/kar hozzáadása” gombra kattintunk, felugrik egy megerősítést kérő ablak. Az „Igen” gombra kattintunk, akkor létrehozásra kerül egy új szak/kar, és megnyílik egy ablak, amiben tudjuk a tulajdonságait szerkeszteni. Ha a „Nem” gombra kattintunk, nem történik semmi.
 - Ha rákattintunk egy listaelemre, megnyílik egy ablak, amiben tudjuk a tulajdonságait szerkeszteni.
- **Határidők szerkesztése**
 - Ha rákattintunk egy határidőre, felugrik egy dátumkiválasztó.
 - Ha nem valid dátumot írunk be, nem mentődik el az adat.
- **Navigációs oldalsáv, felhasználói beállítások**
 - Ha a navigációs oldalsáv gombjára kattintunk, az oldalsáv állapotától függően vagy összezsugorodik, vagy kinyílik.
 - Ha a navigációs oldalsávban rákattintunk egy elemre, az adott nevű oldalra visz minket az applikáció.
 - Ha a felhasználói beállítások „Információk” menüelemére kattintunk, felugrik egy ablak a felhasználónkkal kapcsolatos adatokat tartalmazva.
 - Ha a felhasználói beállítások „Oldal kinézete” menüeleménél kiválasztunk egy témát, a felhasználói felület felveszi az annak megfelelő színvilágot.

- Ha a felhasználói beállítások „*Kilépés*” menüelemére kattintunk, elindul a kijelentkeztetési folyamat.

Összegzés

A szakdolgozati applikációm elkészítése közben nagyon sok olyan számomra új dologgal foglalkoztam, amik az egyre inkább web központúvá váló világunkban történő fejlesztéshez elengedhetetlen kellékek. Bízom benne, hogy az itt megszerzett tudás is hozzájárul ahhoz, hogy kiaknázhassam az információs technológiákban rejlő lehetőségeket, és ezzel mások életébe kényelmet, hatékonyságot és pozitivitást tudjak vinni.

Fejlesztési lehetőségek

Úgy gondolom, egy szoftvert folyamatosan fejleszteni kell, ugyanis megváltozhatnak a hozzá társított igények. Jelenleg az alábbi elvégzendő fejlesztéseket tudom elképzelni:

- Szófelhő mentése képként.
- Backend oldalról fokozni a biztonságot.
- Értesítések küldése bizonyos esetekben, például oktatóknak, ha egy hallgató feljelentkezett a témára.
- Üzenetküldési lehetőség a felhasználók között.
- Témaszöveg beadása fájlfeltöltéssel.
- Az SAPUI5 rengeteg jó megoldást tartogat a különböző grafikonok megjelenítésére, a „*Statisztikák*” oldalra ezekből lehetne többet készíteni.

Irodalomjegyzék

- What is SAP Cloud Platform? - <https://www.graz.se/what-is-sap-cloud-platform/>
2021. 05. 23.
- Guide for User Authentication and Authorization in SAP Cloud Platform - <https://blogs.sap.com/2017/11/16/guide-for-user-authentication-and-authorization-in-sap-cloud-platform/>
2021. 05. 23.

- SAP HANA Cloud and Role Attributes - <https://blogs.sap.com/2020/12/21/sap-hana-cloud-and-role-collections/>
2021. 05. 23.