



Projektmunka II.

GKNB_INTM005

SZEMA–BETA Fejlesztési dokumentáció

**Kecskés D. Eszter
Lantos Zsuzsanna
Kamondy Tivadar
Csiszár Gergő
Litter Ádám**

Győr, 2021/2022/1

1. Az adatbázis tárolási stratégiája:

Az alkalmazás a Google Firebase elnevezésű szolgáltatására épül, amely a felhasználók kezelése és a hosting mellett egy cloud alapú NoSQL adatbázist, illetve fájlrendszert is biztosít a fejlesztők számára.

A választást a kiegészítésre szoruló telefonos applikáció döntötte el, mivel az már ezt a szolgáltatást használta. Továbbá fontos megemlíteni, hogy a Firebase egy remek, ingyenes megoldás kisebb projektek elkészítéséhez, később pedig skálázhatósága miatt nem szükséges lecserélni a már használt eszközöket.

1.1. Az adatbázis struktúrája:

Az adatbázis 6 darab bázis kollekcióból épül fel jelenleg, amelyek további kollekciókat és dokumentumokat tárolhatnak. Ezeket a következőkben egyesével mutatjuk be szerepük és tárolt értékeik szempontjából.

1.1.1. USERS kollekció

Minden tárolt dokumentum egy-egy felhasználó adatainak feleltethető meg.

Attribútum	Típusa	Leírása
AVATAR	STRING	A felhasználó google fiók profilképének elérési útja.
CONFIRMED	BOOLEAN	Visszaigazolt felhasználóról van-e szó.
EMAIL	STRING	Felhasználó google fiókjához tartozó email cím.
NAME	STRING	Felhasználó neve
TITLE	STRING	Felhasználó titulusa [Oktató,Fejlesztő,Tanuló]
GROUPS	STRING ARRAY	Felhasználóhoz rendelt csoportok azonosítói.

1.1.2. GROUPS kollekció

A felhasználók hozzárendelhetők az itt definiált csoportokhoz, annak érdekében, hogy a zárthelyi jellegű feladatsorok esetén csoportos hozzáférést biztosíthassanak nekik a tanárok.

Attribútum	Típusa	Leírása
ICON	STRING	Csoport ikonja.
OWNER	STRING	Csoport felelőse.
TITLE	STRING	Csoport neve.

1.1.3. NEWS kollekció

A fejlesztők és a tanárok kezelhetnek különböző híreket/bejegyzéseket, amelyeket a felhasználók a mobilos applikációban érnek el. Céljuk általában tájékoztató jellegű.

Attribútum	Típusa	Leírása
DATA	STRING	A poszt létrehozásának/módosításának időpontja (YYYY-MM-DD)
MESSAGE	STRING	A bejegyzés tartalma.
PROFILEIMAGE	STRING	A bejegyzéshez rendelt ikon elérési útja.
USER	STRING	A bejegyzést létrehozó/módosító felhasználó titulusa.

Az attribútumok elnevezései ez esetben az eredeti adatbázisból öröklődtek, ezért előfordulhat, hogy nem írják megfelelően körbe az adott érték jellegét.

1.1.4. QUESTIONS kollekció

A fejlesztők és a tanárok létrehozhatnak különböző típusú kérdéseket, amelyeket témakörökhöz és később feladatsorokhoz rendelhetnek.

Az egységes attribútumok:

Attribútum	Típusa	Leírása
ISPICTURE	BOOLEAN	Tartozik-e a kérdéshez ábra.
PICTURE	STRING	Az ábra elérési útvonala amennyiben létezik.
POINTS	NUMBER	A kérdés pontszáma.
TOPICNAME	STRING	A kérdés témaköre.
TYPE	NUMBER	A kérdés típusa
QUESTION	STRING	A kérdés tartalma.
ANSWERS	STRING ARRAY	A kérdésre adható válaszok.

A kérdések helyes válasza(i) számosságban és jellegük szerint is változnak típusonként. A típusok értékei a „0”, „1”, „3” értékek lehetnek. (Eredeti felépítés, úgy volt, hogy a jövőben lesz „2.” típus definiálva, de ez feltételezhetően rejtély marad.)

Típus	Helyes válasz leírása
0	A helyes válasz pozíciója + 1.
1	Boolean tömb a helyes és helytelen válaszok sorrendjének megfelelő tartalommal.
3	String tömb az elvárt válaszok értékeivel.

1.1.5. TOPICS kollekció

A kérdések rendelkeznek különböző témakörökhöz, a könnyebb szűrés és átláthatóság érdekében. A témakörrel nem rendelkező kérdések az „Alapértelmezett” témakör értékét kapják meg.

Attribútum	Típusa	Leírása
DESCRIPTION	STRING	A témakör leírása.
TOPICNAME	STRING	A témakör elnevezése.

1.1.6. QUIZES kollekció

A fejlesztők és a tanárok különböző jellegű feladatsorokat hozhatnak létre, amelyek több modulból állhatnak. A modulokhoz a korábban definiált kérdéseket rendelhetik. A tárolási forma ez esetben is az eredeti felépítést követi, azonban több módosítást végeztünk a kérdések tárolását illetően a duplikációkkal járó problémák elkerülésének érdekében. A különböző modulok, külön attribútumként szerepelnek a továbbiakban is, ezért a táblázatban „példa” jelleggel kerül egy bemutatásra.

Attribútum	Típusa	Leírása
„MINTA_MODUL”	STRING ARRAY	1. Tartalmazott kérdések azonosítói kettősponttal összefűzve. 2. Előfeltételként megjelölt modulok azonosítója kettősponttal összefűzve. 3. Modul ikonja. 4. Modul tartalmi leírása. 5. Modul neve.
TITLE	STRING	Feladatsor címe.
DESCRIPTION	STRING	Feladatsor leírása.
ICON	STRING	Feladatsor ikonjának elérési útja.
GROUP	ICON	A feladatsorhoz rendelt csoport, amennyiben zárthelyi dolgozat jellegű.
TIME	STRING ARRAY	A feladatsor elérhetőségének kezdete és vége, amennyiben zárthelyi dolgozat jellegű.
ZH	BOOLEAN	A feladatsor típusa.

MODULEIDS	STRING ARRAY	A feladatsor által tartalmazott modulok nevei.
-----------	--------------	--

2. A fájlrendszer tárolási stratégiája

Ahogy az már említésre került korábban, a Firebase szolgáltatás rendelkezik saját fájlrendszerrel. Ezt jelenleg elsősorban a különböző ikonok, illetve az egyes kérdésekhez tartozó ábrák tárolására alkalmazzuk.

Elérési út	Tartalom jellege
/post_icons/	A hirdetésekhez rendelhető ikonok.
/question_images/	A kérdések által tartalmazott ábrák.
/quizes/	A feladatsorokhoz és modulokhoz rendelhető ikonok.

3. Saját fejlesztésű React komponensek

A ReactJS library középpontjában az újra felhasználható saját komponensek állnak, amelyeket különböző funkciókkal és designnal ruházhatunk fel a fejlesztés során.

A SZEMA-BETA weblap fejlesztése során számos helyen alkalmaztuk a React MaterialUI nyújtotta sablon komponenseket, illetve számos egyéni megoldásra, illetve átdolgozásra is akad példa. Mivel nagyon sok komponensről van szó, ezért csak példa jelleggel mutatnánk be egy lényegesebb darabot a következőkben.

3.1. Modul komponens

3.1.1. Leírás

A modul komponens a feladatsorok összeállításánál alkalmazzuk, számos másik komponens tartalmaz, mint például a kérdések és a különböző modálok, amelyek segítségével módosítható akár mely tulajdonsága az adott elemnek.

Számos attribútummal rendelkezik, amelyeket ugyan lehetett volna egyetlen objektumban átadni, azonban a későbbi felhasználás során ez nagy problémát jelenthetne, mivel sokkal pontosabb dokumentációra lenne szükség az objektum kulcs-érték párojainak átlátásához. Valamint mi is csak a fejlesztés során ismerkedtünk meg a környezet adottságaival és korlátaival, ezért elszórtan lehet találkozni refactorálást igénylő részekkel.

3.1.2. Példa implementáció

A következő két képernyőkivágás a modul komponens egy gyakorlati implementálását mutatja, először a kódban, majd pedig a felhasználói felületen.

```

<>
<Modul
  setCanStep={this.props.handleStep}
  allModul={this.state.allModul}
  key={"modul-"+index}
  id={"modul-"+index}
  index={index}
  data={module}
  questions={
    module["data"][0].split(':').map((questionID)=>{
      let pos = this.props.questions.findIndex(obj => obj.id === questionID)
      if(pos !== -1){
        return this.props.questions[pos]
      }
    })
  }
}
allModulData = {
  this.props.modules.map((modul)=>{
    return {title: modul.title, "description": modul.data[3]}
  })
}
}
allQuestion={this.props.questions}
handleAddModul={this.handleAddModul}
handleDeleteModul={this.handleDeleteModul}
handleModifyModul={this.handleModifyModul}
handleModifyModulQuestions={this.handleModifyModulQuestions}
/>
</>

```

II. Modul

Módosítás X

Ez a második modul leírása.

ÚJ KÉRDÉS

What is the term to describe this code? [3 pont]

X

count, fruit, price = (2, 'apple', 3.5)

A. Lehetőség: tuple assignment

B. Lehetőség: tuple unpacking

Helyes

C. Lehetőség: tuple matching

D. Lehetőség: tuple duplication

ÚJ KÉRDÉS

What is an abstract class? [2 pont]

X

A. Lehetőség: An abstract class is a name for any class from which you can instantiate an object.

B. Lehetőség: Abstract classes must be redefined any time an object is instantiated from them

C. Lehetőség: Abstract classes must inherit from concrete classes.

D. Lehetőség: An abstract class exists only so that other "concrete" classes can inherit from the abstract class.

Helyes

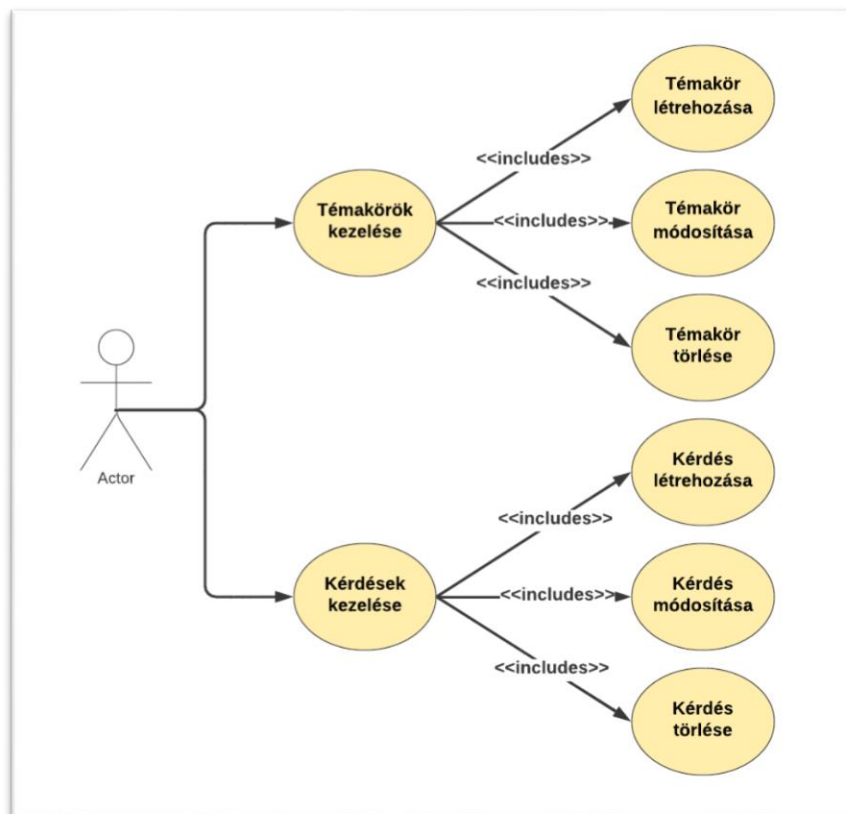
ÚJ KÉRDÉS

+

4. Felhasználói interakciók (megvalósított verzió)

Az előző félévben elkészítésre került egy dokumentáció, ahol a felhasználói interakciókat elemezve több folyamatára is létre lett hozva. Ezek többsége pontosan került implementálásra, azonban a feladatsor kezelés eltér az eredeti tervezettől, illetve a témakörök kezelésére kimaradt és közben a kérdésbázist kezelő oldalra került, ezért most ezeket a folyamatokat fogom részletezni a következőkben.

4.1. Témakörök kezelése



1. ábra: Kérdésbázis oldal funkcionalitása

4.2. Feladatsorok összeállítása

A feladatsor oldalon a felhasználónak három lehetősége nyílik:

- sablon nélküli, teljesen új feladatsor létrehozására,
- egy már létrehozott sablon felhasználására, illetve
- már meglévő feladatsor módosítására/törlésére.

A három folyamat az elérhető funkciók szempontjából csak annyiban különbözik, hogy a harmadik esetében elérhető a törlés lehetősége. Az egyes folyamatok lépéseiről részletes leírás található a felhasználói dokumentációban.