

Szókártya alkalmazás

Szoftverarchitektúrák házi feladat

Követelményspecifikáció

Készítették

Gyuris Bence Zsolt (MFSX9D)

Olasz-Szabó Bence (Q95A6S)

Konzulens

Nagy Ákos

Tartalom

1. Feladatkiírás	2
2. Funkciók	2
2.1. Felhasználókezelés.....	2
2.2. Szókártyák készítése	2
2.3. Szókártyák tematikus rendezése.....	2
2.4. Szókártyák megosztása	3
2.5. Teszt.....	3
2.6. Használati esetek.....	4
3. Nem funkcionális követelmények	5
3.1. Naplózás	5
3.2. Biztonság	5
3.3. Teljesítmény.....	5
4. Implementációs követelmények	6
4.1 Rendszer ábra	6
4.2 Kliens.....	6
4.3 Szerver	6
4.4. Architektúra.....	7
4.5 Adatbázis séma	7

1. Feladatkiírás

A szókártyák az idegen nyelvi szókincs bővítésének hatékony és játékos módszerei. A feladat egy olyan alkalmazás megvalósítása, ami ilyen módon segíti a nyelvtanulást.

Megvalósítandó feladatok:

- Felhasználókezelés
- Szókártyák készítése
- Szókártyák tematikus rendezése
- Megosztás közösségi oldalakon
- Teszt
- Egyéb, megbeszélés szerinti feladatok

Felhasználandó technológiák:

- A szerveroldali komponensek megvalósítása a vonatkozó Microsoft technológiákkal (.NET, Entity Framework, WCF/Web Api stb.)
- A kliensoldal megvalósítása Windows 8/8.1, Windows Phone 8/8.1 vagy Android alkalmazásként

2. Funkciók

Az alkalmazás a következő funkciókat fogja megvalósítani.

2.1. Felhasználókezelés

Az alkalmazásnak lehetővé kell tennie, hogy a felhasználók fiókokat hozhassanak létre. Ez a következő alfunkciókat foglalja magában: regisztráció, bejelentkezés, kijelentkezés.

- **Regisztráció:** Első használatkor új fiók létrehozása, e-mail, jelszó és felhasználói név megadásával.
- **Bejelentkezés:** E-mail és jelszó megadásával történő bejelentkezés.
- **Kijelentkezés:** Fiókból való kilépés, egyéb felhasználókba való bejelentkezést teszi lehetővé.

Az alkalmazás minden egyéb funkciójához szükséges, hogy a felhasználó bejelentkezve legyen a fiókjába. Ehhez aktív internet kapcsolat kell, így az alkalmazás offline működést nem tud megvalósítani.

2.2. Szókártyák készítése

Minden felhasználó létrehozhatja a saját szókártyáit. Egy szókártyának két oldala van, létrehozáskor a felhasználó ezt a két oldalt adja meg a megfelelő létrehozó oldalon. A kártya eleje az amely alapján ki kell találni a hátulját. A kártya hátulja egy idegen nyelvi szó (szöveg), az eleje lehet magyar nyelvű szöveg vagy kép. A kártyák hátulján lévő szó nyelvét meg kell adni (angol, francia, német stb.). A kártyák *törölhetőek* és *módosíthatóak* is.

2.3. Szókártyák tematikus rendezése

A felhasználók a saját kártyáikat kategóriákba (=csoportokba) rendezhetik. Létrehozhatnak névvel ellátott kategóriákat a megfelelő menüpont kiválasztásával. Minden kártyának

megadható kategória létrehozáskor, később ez módosítható. A kategóriának megadható nyelv, ez a nyelv lesz az alapértelmezett a kategória kártyáin. A kategóriák *törölhetőek és módosíthatóak* is.

2.4. Szókártyák megosztása

A felhasználók elérhetik egymás szókártya kategóriáit (csoportjait). A kategóriák lehetnek publikusak és privátak is. A kategóriákat meg lehet osztani más felhasználókkal közösségi oldalon keresztül (Facebook). A felhasználó ki tud tenni egy hivatkozást a közösségi oldalra, amely egy kategóriára mutat. Aki ezt a hivatkozást megnyitja, az később elérheti a kategória szókártyáit (a kártyák meg lesznek osztva vele). A hivatkozás létrehozásának és használatának technikai részleteit később definiáljuk.

Tehát, egy felhasználó elérheti azokat a szókártyákat/kategóriákat, amelyek:

- publikusak vagy
- megosztva vannak vele (pl. hivatkozásra kattintáson keresztül) vagy
- ő hozta létre őket (sajátjaik).

2.5. Teszt

Az alkalmazás lényege a nyelvtanulás (szókincs bővítés), ezért a legfontosabb funkció a nyelvi tesztelés. Bejelentkezés után elindítható többféle teszt, amelyben kártyák sorozatát kapjuk, ezek előlapjából kell kitalálni a kártyák hátlapját (kapunk egy képet/szót, ez alapján kell megadni a megfelelő idegen nyelvű szót).

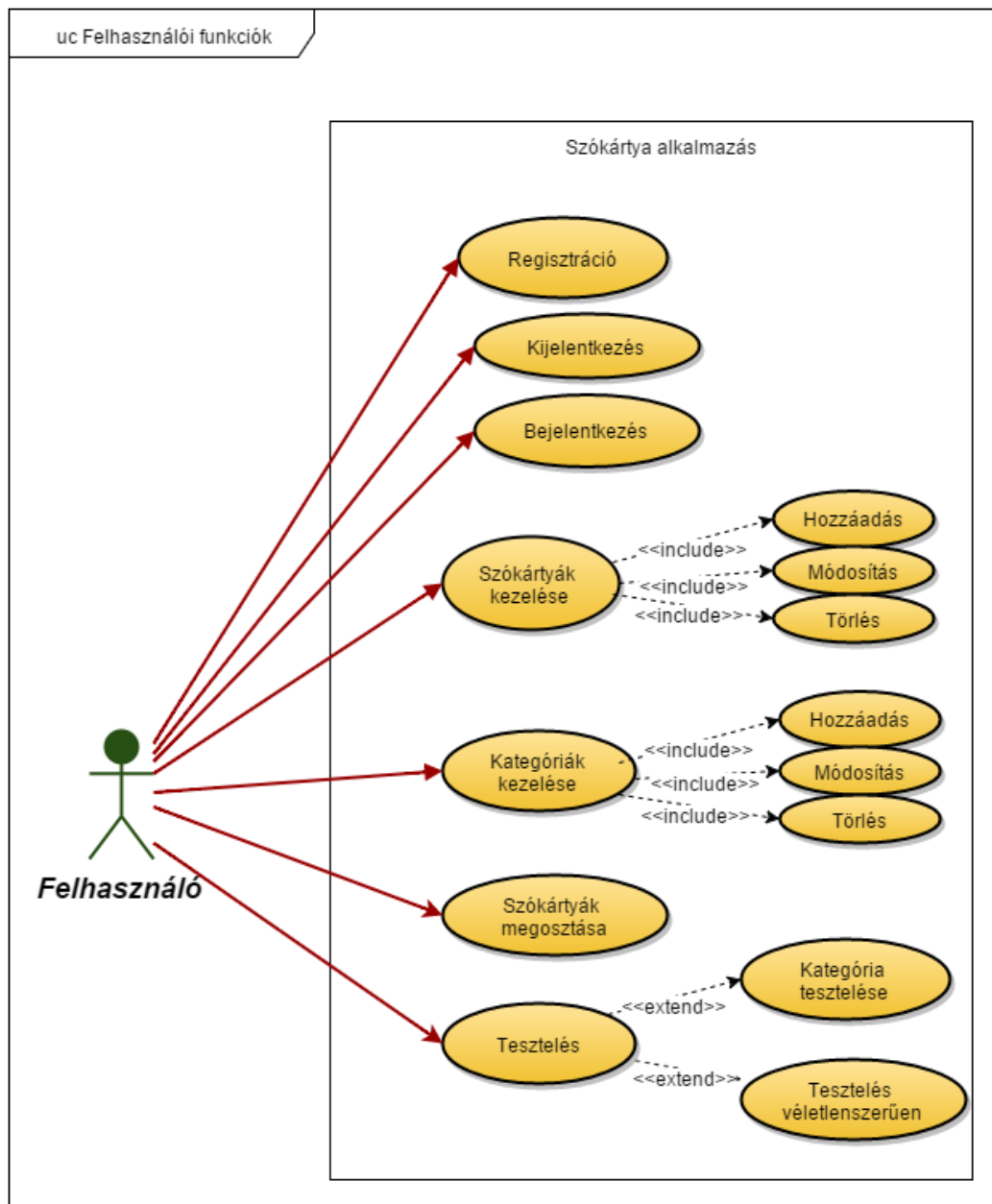
A kártyasorozat attól függ, hogy milyen típusú tesztet indítottunk el. A következők közül választhatunk:

- *Véletlenszerű teszt nyelv alapján:* Bármilyen kártyát kaphatunk a sorozatban (egyet maximum egyszer), amely a megadott nyelvhez tartozik. A teszt addig tart amíg a felhasználó le nem állítja, vagy el nem fogynak a kártyák a rendszerből.
- *Teszt egy kategórián belül:* A kártyák egy kiválasztott kategóriából fognak érkezni, addig tart a teszt, amíg minden kártyán végig nem érünk a kategóriából. A sorrend véletlenszerű.

A befejeződik a teszt (vége a sorozatnak vagy a felhasználó megszakítja) az alkalmazás kiírja a felhasználó eredményét (hány szót talált el).

2.6. Használati esetek

A felhasználói funkciókat a következő Use-case diagram foglalja össze:



1. ábra: Use case diagram

3. Nem funkcionális követelmények

Az alkalmazásnak egyéb nem funkcionális követelményeket is meg kell valósítania.

3.1. Naplózás

Az alkalmazásnak naplóznia kell a fontosabb eseményeket: hibák, leállások, bizonyos műveletek (pl. regisztráció, bejelentkezés, kártya feltöltés, törlés, stb.), rendszeresemények. A naplózást meg kell valósítani mind szerver, mind kliens oldalon. Ennek a célja, hogy a későbbi hibák oka kideríthető legyen, és a javítás könnyebb (lehetséges) legyen.

3.2. Biztonság

A felhasználó kliens programa és a szerver közötti kommunikáció megfelelő szintű *titkosítással* kell hogy legyen ellátva (HTTPS), más ne legyen képes elolvasni az üzeneteket. Legyen megfelelő a szerver oldali *jogosultságkezelés*: a felhasználó csak a saját szókártyáit és csoportjait tudja módosítani, csak a saját nevében tudjon létrehozni ilyeneket, csak a sajátjait tudja megosztani, más felhasználók adataihoz ne férjen hozzá.

Legyen megfelelő *hitelesítés* (authentication): A felhasználónak egyszer kell bejelentkeznie (első használatkor), utána mindig az ő nevében fog történni a kommunikáció (hitelesítő token szükséges).

3.3. Teljesítmény

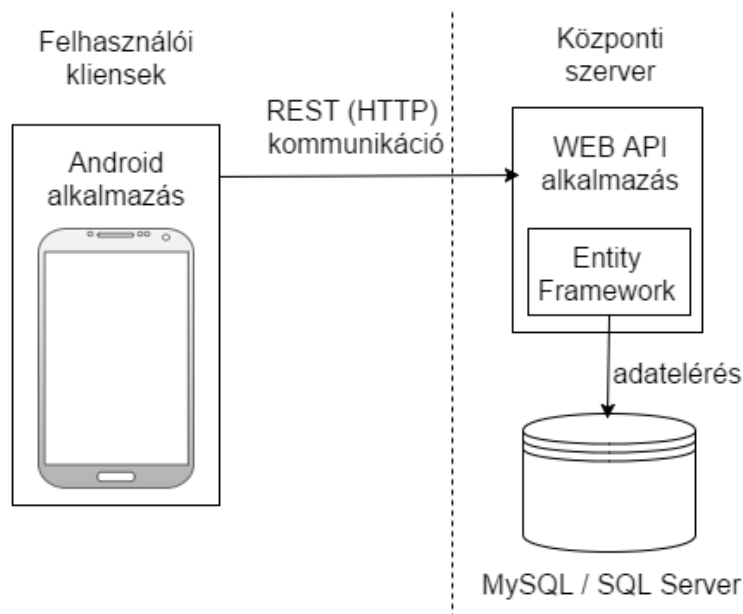
Az alkalmazás felé nincsenek külön teljesítmény igények, a kliens és szerver programok best effort szolgáltatást nyújtanak a minél jobb felhasználói élmény érdekében.

4. Implementációs követelmények

Az alkalmazás egy kliens alkalmazásból egy szerveralkalmazásból és az adatok tárolásáért felelős adatbázisból fog állni.

4.1 Rendszer ábra

A megvalósítandó szoftverrendszer komponenseinek áttekintése a következő ábrán látható:



2. ábra: Rendszer komponensek

4.2 Kliens

A kliens oldalt egy Android alapú mobilalkalmazás valósítja meg. Ez felhasználói input hatására az egyes funkciókhoz tartozó REST kéréseket elküldi HTTP csatornán keresztül, majd a kapott válaszok (Json/XML) alapján megjeleníti a felhasználónak a szükséges tartalmat.

4.3 Szerver

A szerver oldali logikát egy IIS szerveren futó WEB API alkalmazás fogja megvalósítani (C# nyelven kódolva). A perzisztens adatok tárolásáért egy adatbázis szerver lesz felelős. Ezt meg lehet valósítani SQL Server-rel (Microsoft) és MySQL adatbázis szerverrel is (MySQL előny: ingyen létrehozható Azure-ban hallgatói licence-szel, SQL Server előny: egyszerűbb integráció). Az alkalmazáslogika és az adatbázis közötti kapcsolatot az Entity Framework keretrendszer fogja megvalósítani. A szerver feladata, hogy az egyes funkciókhoz tartozó műveleteket a kliens kérésére végrehajtsa és a válaszukat visszaadja.

Felhasználókezelésre a beépített Individual User Accounts modult fogjuk használni.

4.4. Architektúra

Az alkalmazás ebből következően a háromrétegű architektúra modelljét fogja követni.

- *Megjelenítés*: Ennek a rétegnek a megvalósítása a kliens alkalmazás feladata.
- *Üzleti logika*: Ezt a réteget a szerveroldali vezérlőkben valósítjuk meg.
- *Adatelérés*: Ezt a réteget az Entity Framework által automatikusan generált leképzés fogja megvalósítani.

4.5 Adatbázis séma

A következő ábra áttekinti azt a sémát, amellyel lehetséges lesz a szókétyák és kategóriák eltárolása. A felhasználókezelést nem vettem bele, mert annak a sémáját az ASP.NET keretrendszer tartalmazza.



3. ábra: Adatbázis séma ER diagram