SZAKDOLGOZAT

Márta Krisztián

NYÍREGYHÁZA, 2023



**Angol szavak tanulását segítő alkalmazás fejlesztése Java nyelven**

Márta Krisztián

Programtervező informatikus BSc

Dr. Kerényi Gábor, Külső óraadó

**2023**

**Tartalomjegyzék**

[1. Bevezetés 7](#_Toc133786687)

[1.1. Témaválasztás indoklása 7](#_Toc133786688)

[1.2. Célkitűzés 7](#_Toc133786689)

[2. Nyelvoktató alkalmazások 8](#_Toc133786690)

[2.1. Történelmi áttekintés 8](#_Toc133786691)

[2.1.1. Az angol nyelv rövid történelmi áttekintése 8](#_Toc133786692)

[2.1.2. Az oktatás és a számítástechnika kapcsolatának fejlődése 9](#_Toc133786693)

[2.1.3. Az oktatás és a számítástechnika kapcsolata napjainkban 10](#_Toc133786694)

[2.2. Duolingo 11](#_Toc133786695)

[2.3. Quizlet 12](#_Toc133786696)

[3. Java programozási nyelv bemutatása 14](#_Toc133786697)

[3.1. Alkalmazásfejlesztés Javában 14](#_Toc133786698)

[3.2. A Java nyelv főbb jellemzői 15](#_Toc133786699)

[3.3. Adattípusok 16](#_Toc133786700)

[3.4. Vezérlési szerkezetek 17](#_Toc133786701)

[3.4.1. Feltételes utasítások 17](#_Toc133786702)

[3.4.2. Ciklusszervező utasítások 18](#_Toc133786703)

[4. A tervezés menetének leírása 20](#_Toc133786704)

[4.1. A követelmény specifikáció meghatározása 20](#_Toc133786705)

[4.1.1. Regisztrációs felület 20](#_Toc133786706)

[4.1.2. Bejelentkezési felület 21](#_Toc133786707)

[4.1.3. Menü felülete 22](#_Toc133786708)

[4.1.4. Tematikus szószedeteket tartalmazó felület 24](#_Toc133786709)

[4.1.5. Szókártyákat tartalmazó szótanulói felület 25](#_Toc133786710)

[4.1.6. Tesztfeladatsor felülete 26](#_Toc133786711)

[4.1.7. Szerkesztőnézet felülete 28](#_Toc133786712)

[4.1.8. Ranglista felület 31](#_Toc133786713)

[4.2. Használati eset diagram 31](#_Toc133786714)

[4.3. Adatbázis megtervezése 32](#_Toc133786715)

[4.3.1. Felhasználók tábla 32](#_Toc133786716)

[4.3.2. Témakörök tábla 32](#_Toc133786717)

[4.3.3. Szavak tábla 33](#_Toc133786718)

[4.3.4. Tananyag tábla 33](#_Toc133786719)

[4.3.5. Eredmények tábla 33](#_Toc133786720)

[5. A fejlesztés menetének leírása 34](#_Toc133786721)

[5.1. MVC modell alkalmazása 34](#_Toc133786722)

[5.2. Bejelentkezési felület – BejelentkezesJFrame.java 35](#_Toc133786723)

[5.3. Regisztrációs felület – RegisztracioJFrame.java 36](#_Toc133786724)

[5.4. Menü felülete – MenuJFrame.java 38](#_Toc133786725)

[5.5. Tematikus szószedeteket tartalmazó felület - SzoszedetekJFrame.java 39](#_Toc133786726)

[5.6. Szókártyákat tartalmazó szótanulói felület – SzotanuloJFrame.java 39](#_Toc133786727)

[5.7. Tesztfeladatsor felülete – TesztJFrame.java 40](#_Toc133786728)

[5.8. Szerkesztőnézet felülete – SzerkesztoJFrame.java 41](#_Toc133786729)

[5.9. Ranglista felület – RanglistaJFrame.java 44](#_Toc133786730)

[5.10. Fejlesztés során felhasznált szoftverek 45](#_Toc133786731)

[5.10.1. Apache NetBeans integrált fejlesztői környezet bemutatása 45](#_Toc133786732)

[5.10.2. XAMPP webszerver bemutatása 45](#_Toc133786733)

[5.10.3. Draw.io szoftver bemutatása 45](#_Toc133786734)

[6. Eredmények 46](#_Toc133786735)

[6.1. Bejelentkezési felület 46](#_Toc133786736)

[7. Összefoglalás 46](#_Toc133786737)

[8. Irodalomjegyzék 47](#_Toc133786738)

[9. Ábrajegyzék 49](#_Toc133786739)

[10. Táblázatjegyzék 50](#_Toc133786740)

# Bevezetés

## Témaválasztás indoklása

Tanulmányaim során az informatikai sokféle ágazatával megismerkedtem, melyek közül a grafikus felhasználói felülettel rendelkező alkalmazások tervezése és fejlesztése áll hozzám a legközelebb. Jó néhány ilyen alkalmazást készítettem már, kezdve az egyszerűbbektől a bonyolultabbakig, melyeket főként Java és C# programozási nyelvek segítségével fejlesztettem. Így szakdolgozatom témájának kiválasztásakor egyértelmű volt számomra, hogy ebbe az irányba induljak el.

Ezt követően döntést hoztam arról, hogy milyen nyelven és melyik integrált fejlesztői környezetben fejlesszek. A Java nyelvre és a Apache NetBeans fejlesztői környezetre esett a választásom, melyekkel a tanulmányaimból adódóan sok tapasztalattal rendelkezem, és célom, hogy még több tapasztalatot szerezzek a fejlesztés alatt.

Miután minden tisztázódott bennem a fejlesztési és technikai kérdésekkel kapcsolatban, már csak az alkalmazás tematikáját kellett kiválasztanom. Ekkor épp angolt tanultam, ebből adódóan jött az ötlet, hogy az angol tanulás köré építsem fel az alkalmazást, valamint mindenképp szerettem volna, olyan applikációt fejleszteni, mely hasznos és könnyen kezelhető a felhasználó számára. Így született meg, hogy a szakdolgozatom témája, ami nem más, mint angol szavak tanulását segítő alkalmazás fejlesztése Java nyelven.

## Célkitűzés

Célkitűzésem, hogy egy olyan alkalmazást készítsek, melynek segítéségével a felhasználó hatékonyan és könnyedén memorizálhatja az angol szavakat és azok magyar jelentését. A szavakat szószedetek segítségével csoportosítanám. Szeretném segíteni a tanulást képek segítségével, hiszen egy információ hosszútávon való raktározása sokkal egyszerűbb, ha egy képhez kötjük. Ezen felül az alkalmazásban lehetővé teszem a felhasználó számára, hogy személyre szabhassa a tanulási folyamatát, valamint próbára tegye megszerzett tudását teszt segítségével, melynek végén értékelést kap.

Több felhasználó számára is szeretném biztosítani a személyre szabott tanulási folyamatot, ezért többfelhasználós szoftver készítését tervezem.

# Nyelvoktató alkalmazások

A történelmi áttekintésnél fontosnak tartottam, hogy bemutassam az angol nyelv történetét, az oktatás és a számítástechnika kapcsolatát, valamint a digitalizáció szerepét a nyelvoktatásban.

A nyelvek tanulása közben egy szó vagy egy kifejezés jelentésének nagyobb értelmet adhatunk azáltal, ha ismerjük az eredetét. Ilyenkor a nyelv történelmének egy szakaszát akár részletesebben is megismerhetjük, mely nagyban elősegíti a nyelv elsajátítását. Ennek a folyamatnak a megalapozását szeretném elősegíteni, azzal, hogy bemutatom az angol nyelv történelmét röviden. [1]

Az emberek a mai világban már a digitális eszközöknek köszönhetően sokkal könnyebben tehetik magukévá a tudást, mint felmenőink. Sokan nem is tudják elképzelni, hogy az elődjeink, hogyan tudtak tanulni digitális eszközök nélkül, emiatt tartottam fontosnak bemutatni az oktatás és a számítástechnika kapcsolatának fejlődését, valamint a digitalizáció szerepét a nyelvoktatásban.

A történelmi áttekintés után pedig szeretnék bemutatni röviden két ismertebb nyelvoktató alkalmazást.

## Történelmi áttekintés

### Az angol nyelv rövid történelmi áttekintése

Az angol nyelv az indoeurópai nyelvcsaládba tartozik, és a nyelv a Brit-szigeteken alakult ki. Nagy-Britannia őslakossága a kelta népcsoport volt, akiket a Római Birodalom idején Julius Ceaser kiszorított a mai Anglia területéről a walesi és a skót hegyekbe. A rómaiak kivonulása után nem sokáig örülhettek a kelták, hiszen 450 körül jött 3 germán törzs (angelek, szászok, jüttök) és újra kiszorították a kelta népet a hegyekbe, akik magukkal is vitték a kelta nyelvet. Ekkor kezdődött el az angol nyelv fejlődése, mely 3 korszakra osztható. [1][2]

Az első korszak az Óangol, mely 450-től 1066-ig tehető. Másnéven angolszász nyelvnek is nevezzük, ennek elnevezése a két nagy törzs, az angol és a szász elnevezésből ered. Az angol nyelv őse a fentebb említett megszálló népek nyelve. A vikingek gyakran betörtek az országba, így kerültek be az óangol nyelvbe dán és norvég szavak. Ennek következtében történt, hogy elkezdték elhagyni a ragokat, hogy a dánok is megértsék mit mondanak. Jelentős hatása volt még az óangol nyelvre a latinnak, hisz az írni és olvasni tudó emberek latinul tudtak, ennek hatására rengeteg jövevényszó került a nyelvbe. [1][2]

Ezt követő korszak a középangol, mely 1066-től 1500-ig tartott. Hódító Vilmosnak köszönhetően Anglia normann uralom alá került, ami azt eredményezte, hogy a francia lett a hivatalos nyelv, és ennek erős hatása volt az angolra. Francia anyanyelvű vezetők irányították az állami hivatalokat, viszont az átlagemberek még angolul beszéltek, ezek keveredése miatt alakult ki a középangol. Az 1348-as nagy pestis járvány elpusztította a lakosság egyharmadát, a legtöbb vezetővel együtt, ezután újból angol anyanyelvű emberek kerültek az állami hivatalok élére. A hosszú időn át tartó francia behatásnak nagy mértékű hatása volt a nyelvre. [1][2]

A harmadik korszak a modern angol, amely a XV. századtól kezdve egészen napjainkig tart. Ennek az időszaknak a kezdetén már hasonlított a nyelv a maihoz, bár kisebb változások történtek. Készült egy egységes helyesírási rendszer, lefordították a Bibliát, változott a kiejtés, megszűnt a tegező forma, a nyelvtani szabályok leegyszerűsödtek és ezek mellett Shakespeare munkásságának is nagy szerepe volt, hogy kialakuljon a napjainkban használatos nyelv. [1][2]

### Az oktatás és a számítástechnika kapcsolatának fejlődése

Ősrégi technológiák az osztályteremben:

* 1650: A gyarmati korszakban népszerűek voltak a fából készült, nyomtatott leckével ellátott könyvek, amelyek segítették a diákokat a versek megtanulásában. [3]
* 1870: A diagépek elődje, a Magic Lantern vetítette ki az üveglemezekre nyomtatott képeket. [3]
* 1890: Megjelentek a krétatáblák és az iskolai táblák. [3]
* 1900-ban megjelentek a ceruzák, mellyel papírra írtak és elterjedése elavulttá tette az iskolai táblákat. [3]
* Az 1920-as években a tanulás új formája alakult ki, a rádiós tanórák révén. A New York-i Oktatási Tanács rádióállomáson keresztül közvetítette az órákat a hallgatóknak. [3]
* 1930-ban bevezették az írásvetítőt [3]
* 1940-ben megjelent a golyóstoll. [3]
* 1950-ben megjelent a fejhallgató, aminek segítségével a leckéket ismétléssel lehet megtanulni, ennek következtében jöttek létre a nyelvi laboratóriumokat, melyek napjainkban is használatosak. [4]
* 1970-ben megjelentek a kézi grafikus számológépek, melyek nagy sikert arattak a matematika területén. [4]
* 1972-ben megjelent a Scantron tesztelési rendszer, amely lehetővé tette a tanárok számára, hogy hatékonyabban és eredményesebben osztályozzanak. [3]

A személyi számítógépek kora:

* A számítógépek a 80-as években kerültek a mindennapi használatba. [3]
* 1981-ben mutatták be az első hordozható számítógépet. [3]
* 1984-ben az Egyesült Államokban az állami iskolákban átlagosan 92 diákra jutott egy számítógép. [3]
* Az 1985-ös évben jelent meg szintén a CD-ROM meghajtó. Egy lemezen rengeteg adatot lehetett tárolni. [3]
* 1996: Az internet eljutott az oktatási intézményekbe. [3]
* 1999-ben megjelent az interaktív táblák, melyek a kézzel írt jegyzetek és az interaktív technológia keverékei. A korai változatok vezetékkel csatlakoztak az asztali számítógépekhez, míg a legújabbak kompatibilisek okostelefonokkal és táblagépekkel. [3]
* 2007-2010: Az okostelefonok és a táblagépek kora. Megjelent az iPad, ami új magasságokba emeli az osztálytermi oktatást. [3]

### Az oktatás és a számítástechnika kapcsolata napjainkban

Az oktatás és a számítástechnika közötti kapcsolat nagyon fontos szerepet tölt be a mai modern társadalomban. A technológia rohamos fejlődése új módszerek és eszközök alkalmazására ad lehetőséget az oktatás területén. Az oktatási intézményekben az informatikai eszközök, valamint az oktatást segítő alkalmazások száma folyamatosan nő, melyek segítik a tanárok és a diákok életét.

A diákok már nem toll, papír és jegyzetfüzet segítségével jegyzetelnek az órákon, hanem okostelefonjukon, laptopjukon, tabletjeiken. Vannak olyan diákok is akik a távoktatás lehetőségével élve otthonról végzik tanulmányaikat. A tanárok pedig könnyebben és interaktívabban adhatják át a tudásukat a diákoknak, valamint digitális eszközökkel, interaktív táblákkal, prezentációkkal és videókkal teszik élményszerűbbé az oktatást. [5]

A mai digitalizált világban viszont már nem a tanár a legfőbb információforrás a diákok számára, hisz az internet segítségével bárhonnan, bármikor és bárhogyan hozzájuthatnak információhoz. [5]

Az interneten található számos információforrás, ilyenek például az alábbiak:

* Videómegosztó portálok
* E-learning platformok, rendszerek
* Ingyenes és fizetős kurzusok
* Szakmai fórumok
* Online szótárak, fordítók
* Tudásbázisok
* Alkalmazások mindenféle platformra
* Digitálisan elérhető könyvek, könyvtárak
* Podcastek
* Online lexikonok

Fontos, hogy ha ezeken a forrásokat vesszük igénybe tájékozódjunk arról, hogy mennyire hitelesek.

## Duolingo



1. ábra: Duolingo logó [6]

2013-ban a Duolingot (1. ábra) a legjobb nyelvtanuló alkalmazásnak választotta a Google. Az alkalmazásban több mint 40 nyelv közül választhatunk. A játékos oktatásra épül, ugyanis szinteket kell teljesíteni, melyek egy adott tematikára épülnek. A szintek végén a felhasználó próbára teheti tudását a szintugró tesztekkel. Található benne egy olyan funkció, amely a mindennapi és rendszeres tanulásra ösztönöz, ez pedig a széria számláló, amely megmutatja, hogy hány napot töltöttünk sorozatban tanulással. A példamondatok mindig egy-egy új szót tartalmaznak. Ezeket ismétlés útján megtanulhatjuk hallás utáni értéssel, kiejtéssel, fordítással, vagy akár feleletválasztós kérdésekkel is. [7]

Előny:

* Csak regisztrációra van szükség a tanulás elkezdéséhez.
* Azonnal leellenőrizhető a feladat.
* Számítógépen és okostelefonokon is működőképes.
* Nem kell egyetlen egy leckéért sem fizetni.
* Magyar nyelven is elérhető a felhasználói felület.
* Átlátható és könnyen kezelhető felhasználói felület.

Hátrány:

* A zajos tanulás zavarhatja a környezetünket.
* A mondatok elég gyakran nem életszerűek.

## Quizlet



2. ábra: Quizlet logó [8]

A Quizlet (2. ábra) segítségével lehetőségünk van szókártyákat készíteni. Idegen nyelv tanulásakor lehet a kártya egyik oldala az idegen nyelvű szó, másik pedig a magyar jelentése. Használhatjuk fogalmak és a hozzájuk tartozó definíciók gyakoroltatására is, ilyenkor viszont nagyon figyelni kell arra, hogy a hosszú válaszok beírása nehézkes, mert a program csak a pontosan megegyező karaktersorozatot fogadja el. Generálhatunk vele tesztfeladatsorokat, mely változatos feladatokat ad, valamint ki is nyomtathatjuk a szókártyákat. [9]

Előny:

* Csak regisztrációra van szükség a tanulás elkezdéséhez.
* Számítógépen és okostelefonokon is működőképes.
* A szókártyák addig gyakorolhatók, amíg minden szó rögzül.
* Könnyen kezelhető felhasználói felület.

Hátrány:

* Egy szókártyakészlet maximum 50 szót tartalmazhat.
* Csak angol nyelvű felhasználói felület.

# Java programozási nyelv bemutatása



3. ábra: Java programozási nyelv logója [10]

A Java programozási nyelvet (3. ábra) az 1990-es évek közepén fejlesztette ki a Sun Microsystems. A Java alapítója, James Gosling az Oak nevet szerette volna adni az új nyelvnek, de már létezett ilyen nevű programozási nyelv. A legenda szerint a Jáva szigetéről nevezték el a nyelvet, mivel a fejlesztés során rengeteg a szigetről származó kávét fogyasztottak el. [11]

A Java megnevezés nem csak egy programozási nyelvet jelöl, hanem a programok megírásához és futtatásához szükséges szoftverkörnyezetet, amit Java platformnak nevezünk. A platform részei a Java program, a Java fordító (compiler), a programokat futtató virtuális gép (JVM) és egy fejlesztői programcsomag (Java SDK). A JVM (Java Virtaul Machine) futtatja a számítógépen a programokat, mely biztosítja hardvertől és operációs rendszertől való függetlenséget, azaz a hordozhatóságot. A Java SDK (Java Software Development Kit), vagy ahogy napjainkban hívják JDK (Java Development Kit), egy előre kidolgozott sokoldalú fejlesztői programcsomag, mely szabadon felhasználható komponenseket és könyvtárakat tartalmaz. [11]

## Alkalmazásfejlesztés Javában

Amennyiben Java nyelven szeretnénk programok készíteni, csak egy egyszerű szövegszerkesztőre van szükségünk (Jegyzettömb, Notepad++), illetve a Java fordítóprogramra, amely lefordítja a forráskódot bájtkódra. A lefordított bájtkódot ezután a JVM tudja lefuttatni (4. ábra), melyet akár parancssorból is elvégezhetünk. Az említett módszer eléggé körülményessé teszi a fejlesztést, azonban sokan hívei még ennek. [12]



4. ábra: A forráskódtól a programig [13]

A legtöbb modern programozási nyelvhez elérhető egy IDE (Integrated Development Environment), magyarul integrált fejlesztői környezet, mely tartalmaz egy a nyelvre szabott szövegszerkesztőt, a fordítóprogramot, hibakeresőt, tesztelési lehetőségeket és számos egyéb, a fejlesztést segítő eszközt. A Javához is érhető el ilyen integrált fejlesztői környezet a legismertebbek a NetBeans, Eclipse és IntelliJ. Ezek közül majdnem mindegy melyikkel dolgozunk, hiszen ezek a fejlesztői környezetek minden segítséget megadnak a fejlesztéshez, nekünk csupán annyi dolgunk van, hogy szintaktikailag és szemantikailag is helyes Java forráskódot készítsünk, amelyet a fordító működőképes bájtkódra tud fordítani. [12]

A lefordított bájtkódot a JRE (Java Runtime Environment), futtatja le az adott hardverkörnyezetben. A JRE, vagyis a Java futtatható környezet magában foglalja a virtuális gépet (JVM), valamint a futtatáshoz szükséges legfontosabb osztályokat is. [12]

## A Java nyelv főbb jellemzői

* Platformfüggetlen, azaz hordozható: Akármilyen operációs rendszeren fejleszted az alkalmazásodat, azt egy másik operációs rendszeren is lehet majd futtatni, amennyiben az támogatja a Java futtatókörnyezetet. A fejlesztést követően a Java fordító lefordítja bájtkódra a kódot, melyet a JVM értelmezni tud. Mivel a virtuális gép minden platformon fut, a Java kódodat is tudod majd futtatni bárhol. [14]
* Objektumorientált: Alapja az objektum, melynek attribútumai jelentik az objektumban tárolt adatokat, a metódusok pedig a rajtuk értelmezhető és végezhető műveleteket. A második alappillér az öröklődés, mely azt jelenti, hogy egy osztályból létrehozhatunk egy másik osztályt, úgy, hogy az örökli az ősosztály minden attribútumát és metódusát, ezenfelül újabbakkal is ki tudjuk egészíteni. A harmadik alappillér a többalakúság, vagy másnéven polimorfizmus, ami annyit jelent, hogy egy adott metódus azonosítója közös lehet egy osztályhierarchián belül, viszont a hierarchia minden osztályában a tevékenységet végrehajtó metódus megvalósítása specifikus lehet. [12]
* Biztonságos: Megbízható, vagyis robusztus a Java, mert a programozási hibák egy részét megakadályozza, másik részét pedig kiszűri futás közben ezzel is segítve a fejlesztő munkáját. Megakadályozza a rosszindulatú programok rendszerbe kerülését is, mivel a virtuális gép csak a szigorú szabályoknak megfelelő bájtkódot fordítja le. Szintaktikailag követi a C++-t, viszont sokkal biztonságosabb és megbízhatóbb nyelvet építettek, köszönhető ez annak, hogy a Javában nincs goto utasítás, nincsenek mutatók, nincs többszörös öröklődés, nincs rekordtípus, és az operátorok nem felülírhatók. [11]

## Adattípusok

Egy fejlesztés során nagyon sok adattal kell dolgoznunk, és ezekről el kell döntenünk, hogy milyen értéket képviselnek és milyen műveleteket végezhetünk velük. Kétféle adattípust különböztetünk meg Javában, a primitív és a referencia adattípusokat.

I. táblázat: Primitív adattípusok [15]

|  |  |  |
| --- | --- | --- |
| Típus | Leírás | Méret/Formátum |
| byte | bájt méretű egész szám | 8 bit |
| short | rövid egész szám | 16 bit |
| int | egész szám | 32 bit |
| long | hosszú egész szám | 64 bit |
| float | egyszeres pontosságú lebegőpontos valós szám | 32 bit |
| double | dupla pontosságú lebegőpontos valós szám | 64 bit |
| char | Unicode karakter | 16 bit |
| boolean | logikai érték | true vagy false |

Referencia adattípusok közé tartozik a tömb, az osztályok, és az interfészek.

## Vezérlési szerkezetek

### Feltételes utasítások

if szerkezet: A benne lévő utasítások, csak akkor futnak le, ha az if-ben megadott feltétel teljesül. Ha egyetlen egy utasítás adunk meg az if szerkezetben, akkor nem kell utasításblokkot használnunk.

Szintaktika:

if (feltétel) {

utasítás(ok);

}

Az if szerkezet kibővülhet egy else ággal. Az else ágban lévő utasítások csak akkor futnak le, ha az if ág feltétele nem teljesül. Egy if utasításnak csak egyetlen egy else ága lehet.

Szintaktika:

if (feltétel) {

utasítás(ok);

} else {

utasítás(ok);

}

Az else if szerkezetnek köszönhetően több if ágat is egymásba lehet ágyazni. Ha az első feltétel teljesül, akkor az ahhoz tartozó utasítások fognak végrehajtódni, ha a második feltétel teljesül, akkor az ahhoz tartozó utasítások fognak végrehajtódni és ez így tovább. Amennyiben nem teljesül egyik feltétel sem, akkor a végén található else ág parancsai fognak lefutni.

Szintaktika:

if (feltétel) {

utasítás(ok);

} else if (feltétel) {

utasítás(ok);

} else if (feltétel) {

utasítás(ok);

} else {

utasítás(ok);

}

switch case szerkezet: A switch-case segítségével tudjuk egyszerűbbé tenni a hosszú if else szerkezeteket. A switch kap egy kifejezést, melyet először kiértékel. Ha talál egyezést, akkor az adott case ágban lévő utasításokat lefuttatja, majd az ezután következőket is. Azonban, ha csak a kifejezésnek megfelelő case ág utasításait szeretnénk végrehajtani, akkor minden case ágat megkell szakítani, melyet a break kulcsszó segítségével tudunk megvalósítani.A szerkezet legvégén található egy default ág, melynek utasításai akkor futnak le, ha a kifejezés értékét egyik case ág sem kezelte le.

Szintaktika:

switch (kifejezés) {

case érték:

utasítás(ok); break;

case érték:

utasítás(ok); break;

case érték:

utasítás(ok); break;

default:

utasítás(ok); break;

}

### Ciklusszervező utasítások

A ciklus vagy másnéven iteráció az ismétlődő utasítások végrehajtására szolgál. A ciklusok mindig tartalmaznak egy feltételt, és a ciklusmagban lévő utasítások újra ás újra lefutnak, mindaddig amíg a feltétel igaz. Háromféle ciklusszervező utasítást különböztetünk meg, melyeket röviden bemutatok.

while ciklus: A while ciklus egy feltételes ciklus, amely előltesztelő. A feltételes ciklusokat általában akkor használjuk, ha nem tudjuk a ciklusról, hogy hányszor kell lefutnia futtásidőben. A while ciklus megvizsgálja először a feltételt és ha igaz, akkor lefut a ciklusmag és kezdődik elölről, ha nem igaz kilép a ciklusból. Előfordulhat, olyan is, hogy a while ciklus egyetlen egyszer sem fut le.

Szintaktika:

while (feltétel) {

utasítás(ok);

}

do-while ciklus: A do-while ciklus úgyszintén feltételes ciklus, amely hátultesztelő, ebből következik, hogy a ciklusmagban szereplő utasítások egyszer mindenképpen végrehajtódnak. A ciklusmag legelső lefutása után kiértékeli a feltételt, amely, ha igaz, akkor újra fut a ciklus, ha nem akkor pedig kilép a ciklusból.

Szintaktika:

do {

utasítás(ok);

} while (feltétel);

for ciklus: A for ciklus egy számláló ciklus, de szokták még nevezni növekményes ciklusnak is.

Szintaktika:

for ( inicializálás; feltétel; ciklusváltozó növelése ) {

utasítás(ok);

}

Az inicializálás az, ahol deklaráljuk a ciklusváltozót és adunk neki egy kezdeti értéket. A feltétel a belépési feltétel, amely, ha igaz akkor, végrehajtódik a ciklusmag, egyébként kilép a ciklusból. Ciklusváltozó növelése alatt pedig a ciklusváltozó értékének változtatását értjük, ami nem feltétlen csak növelés lehet, hiszen csökkenthetjük is a ciklusváltozó értékét.

# A tervezés menetének leírása

## A követelmény specifikáció meghatározása

Egy szoftver tervezése és fejlesztése során a legfontosabb kezdeti lépés, a követelmények rögzítése, ezt nevezzük követelmény specifikációnak. A specifikációk a fejlesztés során nagyban elősegíti a munkánkat, hiszen tartalmaz minden olyan funkciót és annak leírását, amelyeket a programnak tudnia és tartalmaznia kell. Az alkalmazásom esetében ezen specifikációkat felületenként elkülönítve fogom kifejteni.

Az alkalmazás indulásakor a bejelentkezési felület jelenlen meg.

### Regisztrációs felület

Célkitűzéseim között említettem már, hogy szeretném, ha több felhasználó is tudná használni az alkalmazást személyre szabottan. Ebből kifolyólag az alkalmazásnak rendelkeznie kell egy regisztrációs felülettel, ahol a felhasználó adatait regisztrációkor eltudjuk tárolni adatbázisba.

A regisztráció során a felhasználónak az alábbi adatokat kell megadnia a beviteli mezők segítségével:

* e-mail cím
* felhasználónév
* jelszó
* jelszó újra

A jelszó és a jelszó újra beviteli mezők mellett található legyen egy-egy szem ikon, amely azt a célt szolgálja, hogy segítségével felfedhessük, majd újra elrejthessük a jelszó mezők tartalmát.

A regisztráció a „REGISZRÁCIÓ” feliratú gombra való kattintással, vagy az ENTER billentyűzet leütésére történjen meg, ha minden megadott adat helyes, hiszen megkell vizsgálnunk a beírt adatok helyességét. Ha hibásan megadott adatot észlel az alkalmazás akkor azt jelezze hibaüzenet segítségével.

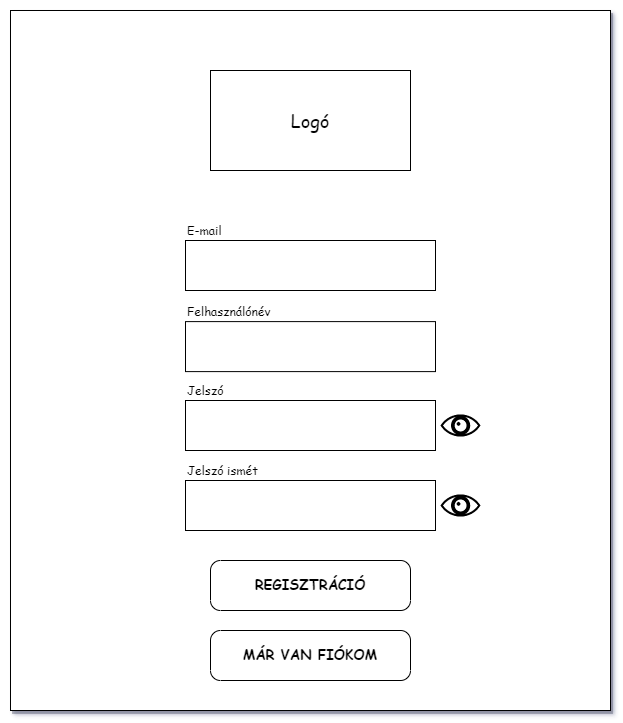
Hibás adatok az alábbiakat lehetnek:

* Ha üresen hagyta a felhasználó bármelyik mezőt.
* Ha a felhasználó által megadott e-mail cím formátuma nem megfelelő.
* Ha a felhasználó által megadott e-mail címről már regisztráltak.
* Ha a felhasználó által megadott felhasználónév már foglalt.
* Ha a felhasználó által megadott jelszó rövidebb, mint 8 karakter.
* Ha a felhasználó által megadott jelszó nem tartalmaz legalább egy kisbetűt, egy nagybetűt és egy számot.
* Ha a felhasználó által megadott jelszavak nem egyeznek.

Amennyiben minden megadott adat helyes, akkor az adatokat el kell mentse a megfelelő adatbázisba, majd tájékoztassa a felhasználót a sikeres regisztrációról és irányítsa át a bejelentkezési felületre.

A regisztrációs felületen biztosítani kell azt is, hogy visszatérhessünk a bejelentkezési felületre, ha mégsem szeretnék regisztrálni. Ez a „MÁR VAN FIÓKOM” feliratú gombra való kattintással történjen meg.

A felületen jelenlen meg az alkalmazás logója is.



5. ábra: Regisztrációs felület drótváza

### Bejelentkezési felület

A regisztrált felhasználóknak biztosítani kell a regisztráció során megadott felhasználónév és jelszó segítségével a bejelentkezést, melyet a bejelentkezési felület biztosít.

A jelszó beviteli mező mellett található legyen egy szem ikon, amely azt a célt szolgálja, hogy segítségével felfedhessük, majd újra elrejthessük a jelszó mező tartalmát.

A bejelentkezés a „BEJELENTKEZÉS” feliratú gomb való kattintással történjen meg, ha a megadott felhasználónév és jelszó helyes, azonban, ha hibás adatot észlel az alkalmazás azt jelezze a felhasználó felé hibaüzenet segítségével.

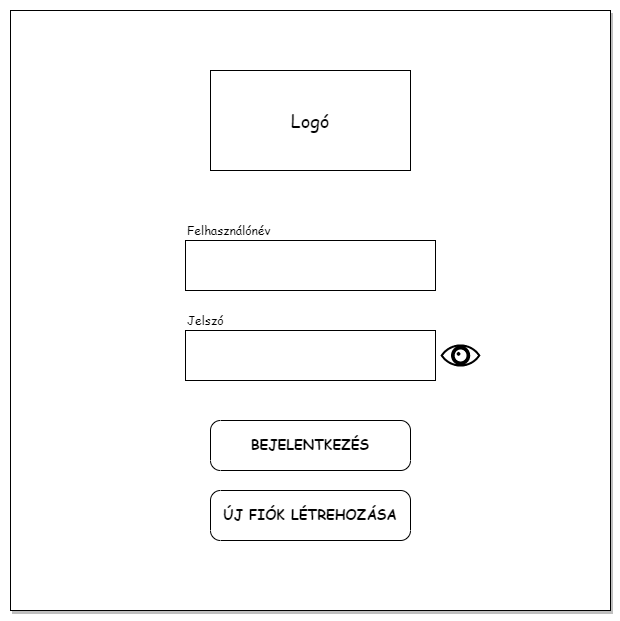
Hibás adatok az alábbiakat lehetnek:

* Ha üresen hagyta a felhasználó bármelyik mezőt.
* Ha a felhasználó által megadott felhasználónév nincs regisztrálva az adatbázisban.
* Ha a felhasználó nem a megadott felhasználónévhez tartozó jelszót adja meg.

Sikeres bejelentkezés esetén irányítsa át a felhasználót a menüt tartalmazó felületre.

A bejelentkezési felületen biztosítani kell a regisztrációs felület megnyitását, mely a „ÚJ FIÓK LÉTREHOZÁSA” feliratú gombra való kattintással valósuljon meg.

A felületen jelenlen meg az alkalmazás logója is.



6. ábra: Bejelentkezési felület drótváza

### Menü felülete

A sikeres bejelentkezést követően megjelenő menü felületének egy helyen kell tartalmaznia az összes elérhető funkciót és az ahhoz tartozó felület elérését a felhasználó számára.

Ezek a funkciók a következők:

* Tematikus szószedeteket tartalmazó felület elérése, valamint majd erről a felületről lehessen elérni a szókártyákat tartalmazó szótanulói felületet.
* Tesztfeladatsor felületének elérése.
* Szerkesztőnézet felületének elérése.
* Ranglista felületének elérése.
* Kijelentkezés.
* Kilépés az alkalmazásból.

A fent említett funkciók megvalósítása gombok segítségével történjen.

A „TEMATIKUS SZÓSZEDETEK” feliratú gombra való kattintáskor irányítsa át a felhasználót a tematikus szószedeteket tartalmazó felületre.

A „TESZT INDÍTÁSA” feliratú gombra való kattintáskor irányítsa át a felhasználót a tesztfeladatsor felületére.

A „SZÓSZEDETEK SZERKESZTÉSE” feliratú gombra való kattintáskor irányítsa át a felhasználót a szerkesztőnézet felületére.

A „RANGLISTA” feliratú gombra való kattintáskor irányítsa át a felhasználót a ranglista felületére.

A „KIJELENTKEZÉS” feliratú gombra való kattintáskor jelentkeztesse ki a felhasználót és irányítsa át a bejelentkezési felületre.

A „BEZÁRÁS” feliratú gombra való kattintáskor záródjon be az alkalmazás.

Ezeken felül a felület tartalmazzon még egy üdvözlő feliratot, melyben köszönti a program a felhasználót.



7. ábra: Menü felületének drótváza

### Tematikus szószedeteket tartalmazó felület

A tematikus szószedeteket tartalmazó felület jelenítse meg az adott felhasználóhoz tartozó összes szószedetet. Az alkalmazás alapból tartalmazzon öt darab beépített szószedetet, amelyek már a regisztráció után elérhetők legyenek a felhasználó számára, azonban a felhasználó hozhasson létre saját maga által személyre szabott szószedeteket is.

A szószedetek megjelenítése gombok segítségével legyen megvalósítva, a gombok tartalmazzák a szószedet nevét, illetve egy képet. Az öt alap szószedet egyedi képpel, míg a felhasználók által létrehozott szószedetek egy alapértelmezett képpel jelenjenek meg, mely az alkalmazás logója legyen.

A felület 3x3-as elrendezésben jelenítse meg a szószedeteket és amennyiben több mint 9 szószedet található meg a felületen, akkor tegye elérhetővé a függőleges gördítő sáv használatát.

Egy adott gombra való kattintáskor a gombnak megfelelő szószedet adataival együtt irányítsuk át a felhasználót a szókártyákat tartalmazó szótanulói felületre.

A felület bezárásakor irányítsa a felhasználót a menü felületére.



8. ábra: Szószedteket tartalmazó felület drótváza

### Szókártyákat tartalmazó szótanulói felület

Az alkalmazásban ezen felület segítségével kell biztosítani a szavak elsajátítását. A felület három panelből álljon, melyek váltják egymást. Ez a három panel az indítás panel, a tanulás panel és a vége panel.

Az indítás panel jelenjen meg a felület megnyitásakor. Itt lehetőséget kell adni a felhasználónak arra, hogy eldöntse milyen nyelven jelenjen meg a kifejezés a szókártyán. A választás rádió gombok segítségével legyen megvalósítva. A „KEZDÉS” feliratú gombra való kattintásra kerüljön elfogadásra a kiválasztott nyelv, majd váltsa az indítás panelt, a tanulás panelre.

A tanulás panel tartalmazzon egy tájékoztató szöveget, melyben a felhasználó arról értesül, hogy az adott szószedetből hány elsajátítandó szó van még hátra.

A szavak megjelenítésére szókártyák szolgáljanak, melyek egymás alatt helyezkedjenek el. A felső kártya a kifejezést (szó), míg az alsó a definíciót (szó jelentése) tartalmazza, de a definíció ne legyen látható a felhasználó számára. A szókártyákat gombok segítségével jelenítse meg. A beépített szószedetek szavainak kifejezés szókártyája tartalmazzon a szóhoz kapcsolódó képet. A definíció gombra kattintva a felhasználó bármikor megjelenítheti annak tartalmát, ekkor tudja leellenőrizni a felhasználó a szó jelentését.

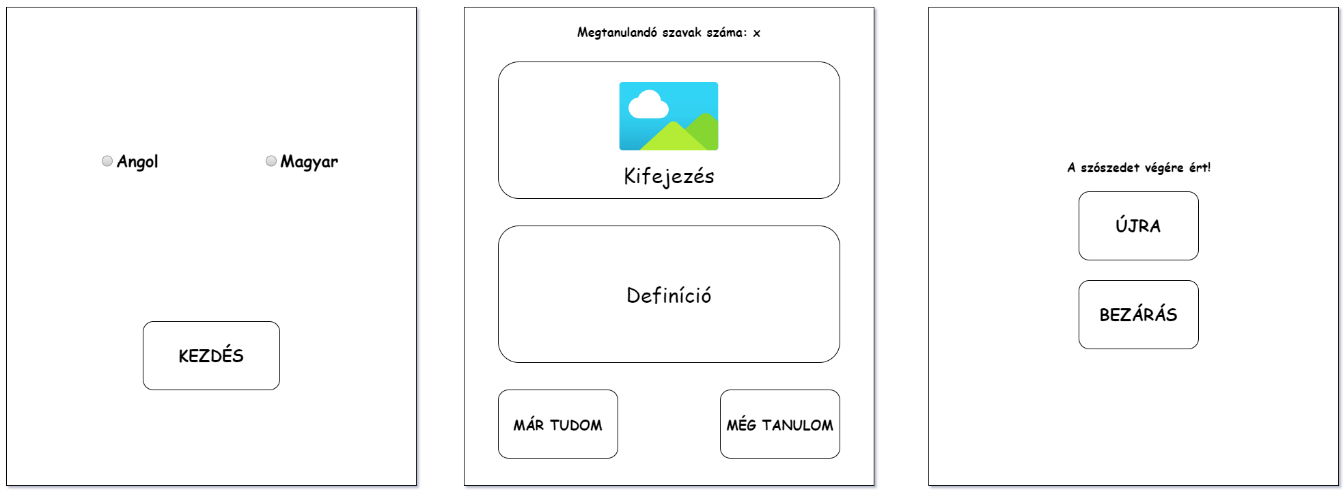
Lehetőséget kell adni arra, hogy a felhasználó eldöntse tudja-e az adott szó jelentését vagy sem. Erre két gomb kell szolgáljon, az egyik a „MÁR TUDOM” feliratú, míg a másik a „MÉG TANULOM” feliratú gomb legyen. A „MÁR TUDOM” feliratú gombra kattintva, távolítsa el a szót a tanulandó szavak közül, és lépjen a következő szóra. A „MÉG TANULOM” feliratú gombra kattintva, egyszerűen csak lépjen a következő szóra. A következő szó megjelenésével ismétlődik elölről a szó tanulási folyamata.

A szószedet tanulása addig induljon újra a még meg nem tanult szavakkal, amíg minden szót el nem sajátított a felhasználó, és erről kapjon tájékoztatást. Amikor az összes szót megtanulta a felhasználó jelenjen meg a tanulás panel helyett a vége panel.

A vége panelen értesüljön a felhasználó arról, hogy sikeresen elsajátította a szószedet összes szavát, valamint legyen lehetősége dönteni, hogy újból átnézi-e az adott szószedet szavait vagy visszalép a menübe.

Az „ÚJRA” feliratú gombra való kattintáskor ismét jelenítse meg az indítás panelt a vége panel helyett.

A „BEZÁRÁS” feliratú gombra való kattintáskor irányítsa át a felhasználót a menüt tartalmazó felületre.



9. ábra: Szókártyákat tartalmazó szótanulói felület paneljeinek drótvázai

### Tesztfeladatsor felülete

A felhasználó számára biztosítani kell, hogy a megszerzett tudását próbára tegye, erre szolgáljon a tesztfeladatsor felülete. A felület három panelből álljon, melyek váltják egymást. Ez a három panel az indítás panel, a teszt panel és a vége panel.

Az indítás panel jelenjen meg a felület megnyitásakor. Itt lehetőséget kell adni a felhasználónak arra, hogy eldöntse milyen nyelven jelenjen meg a kifejezés, valamint, hogy melyik szószedet szavaiból szeretne tesztet indítani. A nyelv kiválasztása rádió gombokkal, míg a szószedet kiválasztása legördülő listával legyen megvalósítva. A „KEZDÉS” feliratú gombra való kattintásra kerüljön elfogadásra a kiválasztott nyelv és szószedet, majd váltsa az indítás panelt, a teszt panelre.

A teszt panel tartalmazza a kifejezést a kiválasztott nyelven, valamint egy beviteli mezőt, hogy a felhasználó beírhassa a válaszát. A beépített szószedetek szavaihoz jelenítsen meg egy kapcsolódó képet. A válasz jóváhagyására a „VÁLASZ” feliratú gombra való kattintással legyen lehetőség, és ekkor az alkalmazás adjon visszajelzést, a pontszámról, valamint arról, hogy a válasz helyes volt-e. Amennyiben helyes volt a válasz, jelenjen meg a „HELYES” felirat, viszont, ha helytelen választ ad a felhasználó akkor a helyes válasz jelenjen meg. A válaszadás után jelenleg meg a „KÖVETKEZŐ” feliratú gomb, melyre ha kattintunk biztosítania kell a következő szó megjelenését.

Amikor a szószedet végére ért a felhasználó a teszt panel helyett jelenjen meg a vége panel, ahol kapjon tájékoztatást a felhasználó az elért eredményéről, az alkalmazás mentse el adatbázisba az eredményt, emellett legyen lehetősége eldönteni a felhasználónak, hogy indít-e újabb tesztet vagy visszalép a menübe.

Az „ÚJ TESZT” feliratú gombra való kattintáskor ismét jelenítse meg az indítás panelt a vége panel helyett.

A „BEZÁRÁS” feliratú gombra való kattintáskor irányítsa át a felhasználót a menüt tartalmazó felületre.



10. ábra: Tesztfeladatsor felület paneljeinek drótvázai

### Szerkesztőnézet felülete

Az alkalmazásnak a felhasználó számára az alábbi lehetőségeket kell biztosítani, hogy személyre szabhassa tanulási folyamatát:

* Új szószedet létrehozása.
* Szószedet szavainak módosítása, törlése.
* Szószedetek törlése.

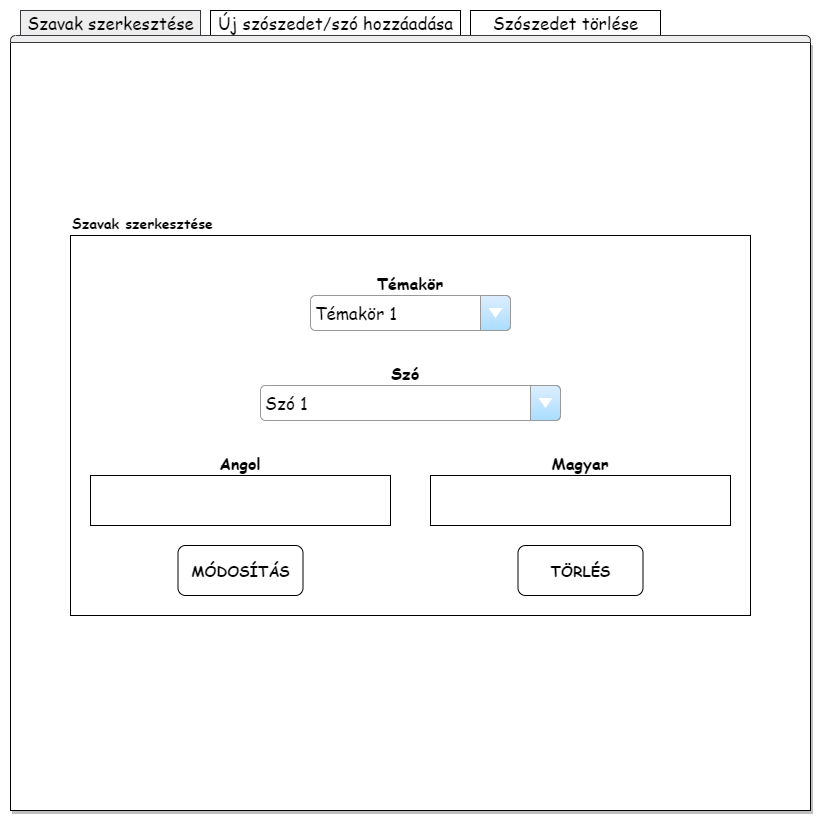
Biztosítani kell, hogy a felhasználó csak a saját maga által létrehozott szószedeteket törölhesse és módosíthassa.

A felületet három fő részre kell osztani, hogy áttekinthetőbb legyen a saját szószedetek menedzselése. Ez a három fő rész a szavak szerkesztése, új szószedet hozzáadása és a szószedet törlése. A három részre osztás füles ablakok (TabbedPane) segítségével legyen megvalósítva.

A szavak szerkesztése ablakban két darab legördülő lista segítségével lehessen kiválasztani, hogy melyik témakör melyik szavát szeretnék módosítani, vagy törölni. A szavak módosítására szolgáljon két beviteli mező, ahol a felhasználó beírhatja, hogy hogyan szeretné módosítani az adott szót.

A szó törlése a „TÖRLÉS” feliratú gombra való kattintva valósuljon meg, ekkor a szó törlése történjen meg az adatbázisban.

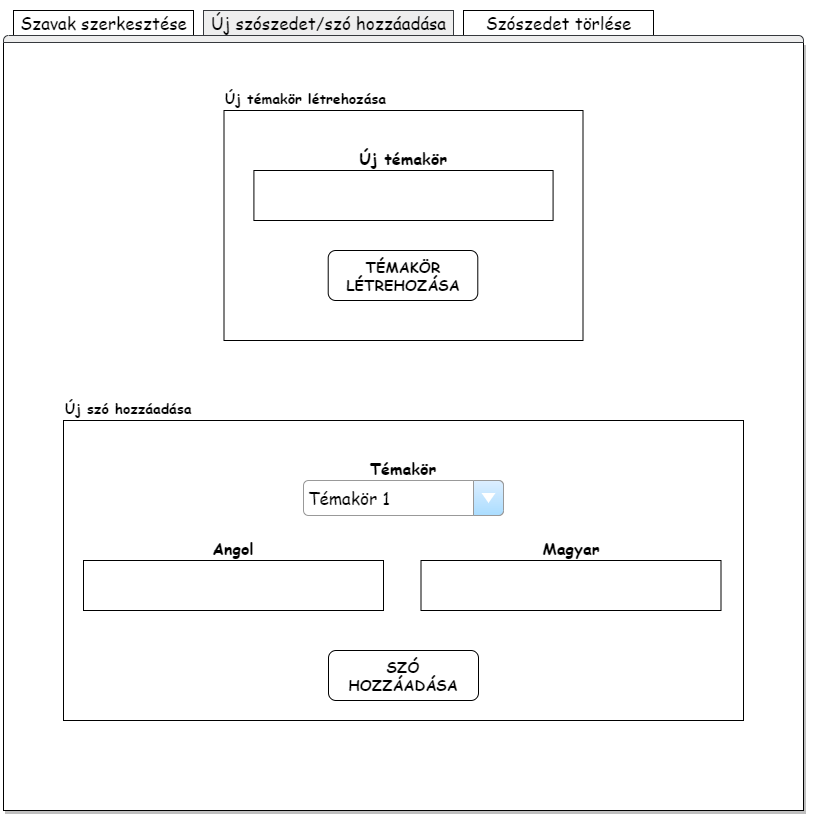
A szó módosítása a „MÓDOSÍTÁS” feliratú gombra kattintva valósuljon meg, ekkor a beviteli mezők értékére módosuljon a szó az adatbázisban.



11. ábra: Szavak szerkesztése ablak drótváza a szerkesztőnézet felületén

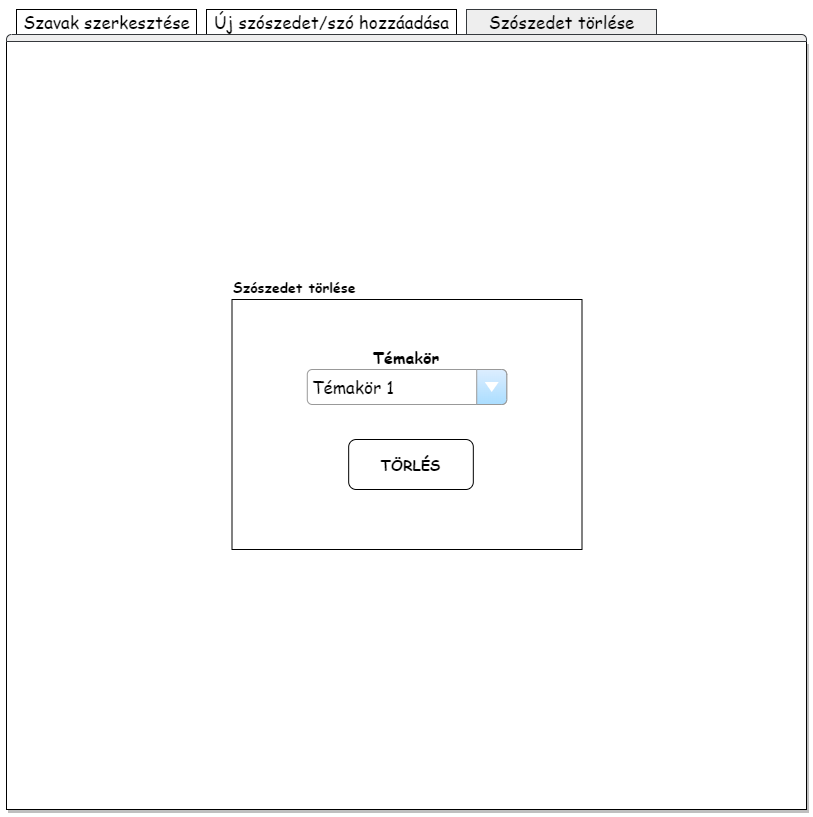
Az új szószedet/szó hozzáadása ablakban biztosítani kell a felhasználó számára, hogy létre tudjon hozni témaköröket és azokhoz új szavakat hozzáadhasson. Az új témakör megadására egy beviteli mező szolgáljon, és ennek adatbázisba való mentése a „TÉMAKÖR LÉTREHOZÁSA” feliratú gombra való kattintással történjen meg. A gombra való kattintáskor, ha üres a beviteli mező, ha már van ilyen témakör, vagy ha a beírt témakör neve hosszabb mint 20 karakter, akkor ezeket jelezze hibaüzenet segítségével.

Az új szó hozzáadásánál egy legördülő listából lehessen választani, hogy melyik szószedethez szeretnénk a szót hozzáadni, valamint beviteli mezők segítségével lehessen megadni a szó angol és magyar megfelelőjét. A „SZÓ HOZZÁADÁSA” feliratú gomb segítségével tárolja el az új szót az adatbázisban. Ha két beviteli mező közül bármelyik üres, akkor azt jelezze hibaüzenet segítségével.



12. ábra: Új szószedet/szó hozzáadása ablak drótváza a szerkesztőnézet felületén

A szószedet törlése ablakban lehetőséget kell adni a felhasználónak arra, hogy törölhesse bármelyik szószedetét. Egy legördülő lista segítségével lehessen kiválasztani, hogy melyik szószedetet akarja törölni. A „TÖRLÉS” feliratú gombra való kattintással törölje ki a szószedet összes szavát, valamint a témakörét az adatbázisból.

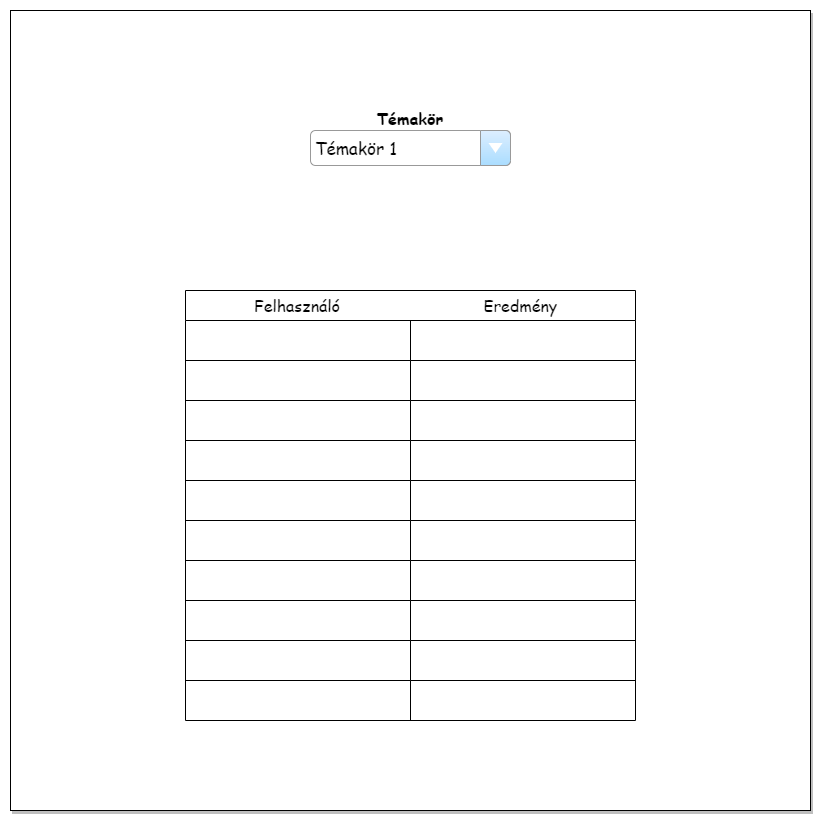


13. ábra: Szószedet törlése ablak drótváza a szerkesztőnézet felületén

### Ranglista felület

A felület jelenítse meg a kiválasztott témakörben elért eredményeket. A témakör kiválasztását legördülő listával, az eredmény megjelenítését táblázat segítségével valósítsa meg az alkalmazás. A táblázat az eredményt csökkenő sorrendben jelenítse meg, tartalmazva a felhasználó nevét, valamint százalékos eredményét.

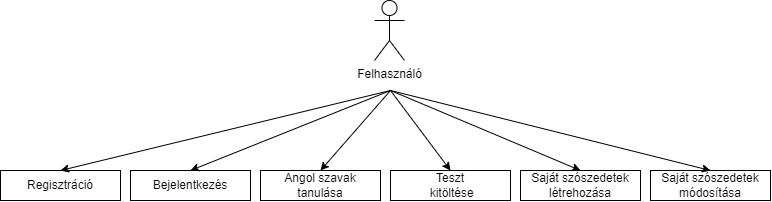
A felület bezárásakor irányítsa a felhasználót a menü felületére.



14. ábra: Ranglista felület drótváza

## Használati eset diagram

A használati eset diagram (15. ábra) modellezi a felhasználó által kiváltható eseményeket.

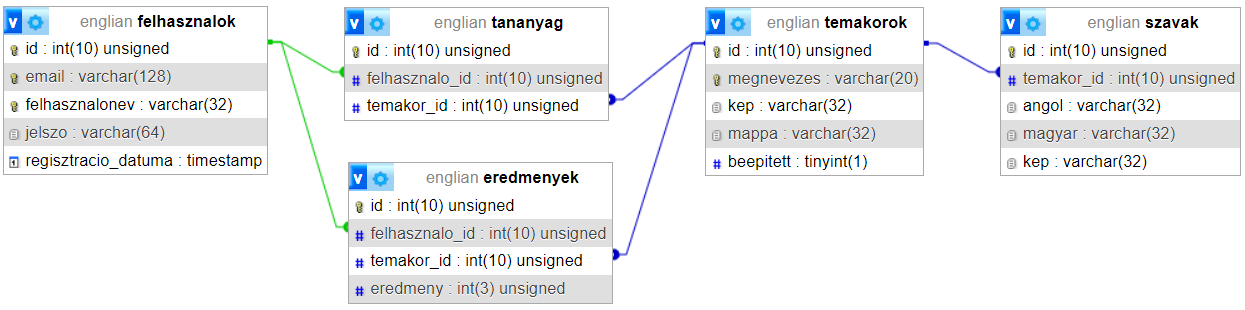


15. ábra: Használati eset diagram

## Adatbázis megtervezése

Az alkalmazásomban nagyon sok adattal kell dolgozni, melyeket el is kell tárolni. Ilyen adatok a felhasználók adatai, a témakörök, a szavak, a teszt eredmények, valamint a felhasználók és a témakörök összekapcsolása. Így ezek tárolását egy adatbázis segítségével oldottam meg, melyet a phpMyAdmin felület segítségével menedzseltem.

Az alkalmazás adatbázisa (16. ábra) *englian* néven lett létrehozva, mely öt táblát tartalmaz melyeket részletesen bemutatok.



16. ábra: Az adatbázis táblái és az azok közötti kapcsolatok

### Felhasználók tábla

A *felhasznalok* tábla tárolja a felhasználók adatait, melyek a következők:

id A felhasználó azonosítója, ez a kulcs. (szám)

email A felhasználó e-mail címe. (szöveg)

felhasznalonev A felhasználó felhasználóneve. (szöveg)

jelszo A felhasználó jelszava MD5 kódolással. (szöveg)

regisztracio\_datuma A felhasználó regisztrációjának ideje. (dátum)

### Témakörök tábla

A *temakorok* tábla tárolja a szószedetek témaköreinek adatait, melyek a következők:

id A témakör azonosítója, ez a kulcs. (szám)

megnevezes A témakör megnevezése. (szöveg)

kep A témakörhöz tartozó képfájl. (szöveg)

mappa A témakörhöz tartozó szavak képeit tartalmazó mappa

neve. (szöveg)

beepitett A témakör az alap öt szószedethez tartozik-e. (logikai)

### Szavak tábla

A *szavak* tábla tárolja a szavak adatait, melyek a következők:

id A szó azonosítója, ez a kulcs. (szám)

temakor\_id A témakör azonosítója. (szám)

angol A szó angolul. (szöveg)

magyar A szó magyarul. (szöveg)

kep A szóhoz tartozó képfájl. (szöveg)

### Tananyag tábla

A *tananyag* tábla tárolja a felhasználók és a hozzájuk tartozó témakörök összekapcsolását.

id A tananyag azonosítója, ez a kulcs. (szám)

felhasznalo\_id A felhasználó azonosítója. (szám)

temakor\_id A témakör azonosítója. (szám)

### Eredmények tábla

Az *eredmenyek* tábla tárolja a felhasználók által elért tesztek eredményét.

id Az eredmény azonosítója, ez a kulcs. (szám)

felhasznalo\_id A felhasználó azonosítója. (szám)

temakor\_id A témakör azonosítója. (szám)

eredmeny A teszt százalékos eredménye. (szám)

# A fejlesztés menetének leírása

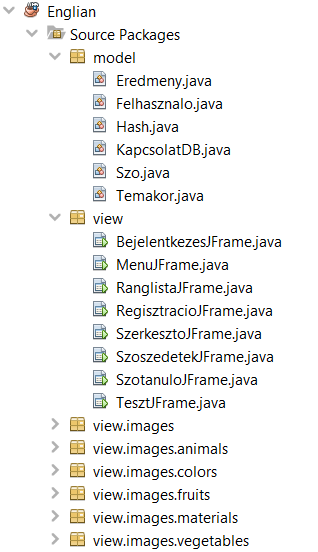
## MVC modell alkalmazása

Az alkalmazás fejlesztése során törekedtem arra, hogy a projekt felépítése az MVC modell alapjaira épüljön. Az MVC modell a Model-View-Controller rövidítése, melynek magyar megnevezése a Modell-Nézet-Vezérlés. Az MVC egy programtervezési minta, melynek célja, hogy az alkalmazásokban az adat és a felhasználói felület elkülönüljön egymástól, abból az okból, hogyha szeretnénk változtatni az adatkezelésen vagy akár újraszerveznénk az adatokat, akkor ne kelljen a felhasználói felülethez hozzányúlni. A modell három fő rétegből áll, melyek különböző feladatokat látnak el. [16]

A Model réteg felelős az adatszerkezetek definiálásáért, az adatok tárolásáért, kezeléséért és módosításáért, valamint az adatbázisban tárolt adatok lekérdezéséért. Ezen rétegben fogalmazódik meg az üzleti logika is. [16]

A View réteg nem mást jelent, mint a felhasználói felületet, melynek feladata, hogy megjelenítse a Model réteg tartalmát. [16]

A Controller réteg feladata, hogy kapcsolatot teremtsen a Model és a View réteg között. A Controller nincs mindig külön rétegként megvalósítva, előfordul néha, hogy összeolvad a View réteggel. [16]



17. ábra: Az alkalmazásom MVC struktúrája

A projekt *Model* rétegében találhatóak az adatok tárolására szolgáló osztályok, a hash függvényt tároló osztály, valamint az adatbázis-kapcsolatért felelős osztály. A *Model* réteg osztályait nem mutatom be külön, mivel azok nagy része az adatbázisból való rekordok tárolását szolgálja, valamint a KapcsolatDB osztály fontosabb SQL parancsokat végrehajtó metódusait az aktuális *View* rétegbeli osztályokban megemlítem.

A V*iew* réteg tartalmazza az össze felhasználói felületet, valamint az alkalmazás során használt képeket. A felhasználói felületek osztályait mutatom be részletesebben.

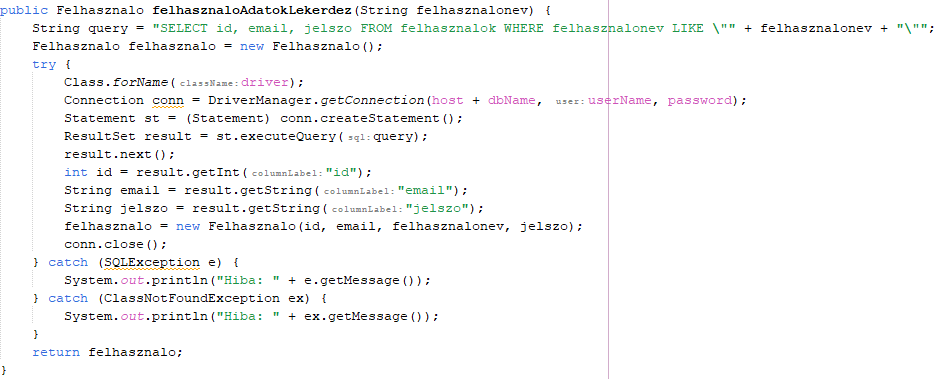
A *Controller* réteg jelen projekt esetén összeolvad a View réteggel.

## Bejelentkezési felület – BejelentkezesJFrame.java

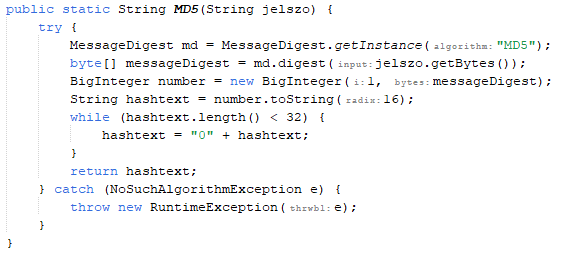
A bejelentkezés során ellenőrizni kell az adatok helyességét. A bejelentkezést megvalósító metódus (18. ábra) legelsőnek megvizsgálja, hogy a megadott felhasználónév és jelszó értékei üresek-e, és amennyiben valamelyik üres, azt egy párbeszéd ablak jelzi a felhasználónak. Azonban, ha egyik sem üres, akkor megvizsgálja, hogy a beírt felhasználónév be van-e regisztrálva, ha nincs akkor hibaüzenetet küld az alkalmazás a felhasználónak. Amennyiben a megadott felhasználónév szerepel az adatbázisban, akkor ellenőrzi, hogy a megadott jelszó a felhasználóhoz tartozik-e, és ha nem akkor azt jelzi a felhasználónak. Ellenkező esetben a bejelentkezés sikeres, a bejelentkezési felület bezárul és megjelenik a menü, valamint a felhasználó adatait lekérem az adatbázisból.



18. ábra: Bejelentkezést megvalósító metódus



19. ábra: Felhasználó adatainak lekérdezése adatbázisból



20. ábra: MD5 hash-függvény

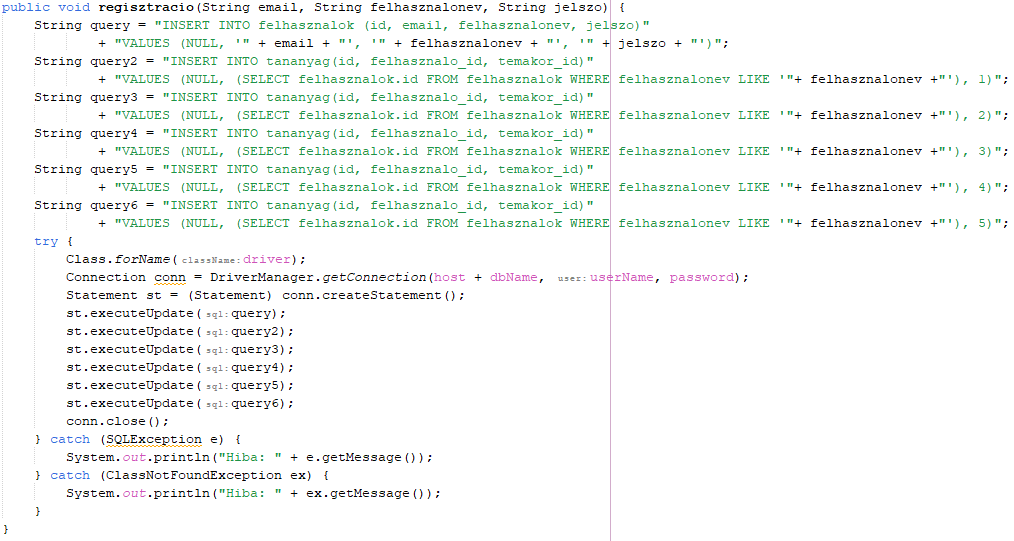
## Regisztrációs felület – RegisztracioJFrame.java

Felhasználó regisztrációja esetén megkell vizsgálni a megadott adatok helyességét. A regisztrációt megvalósító metódus (20. ábra) legelsőnek az e-mail cím validálását (23. ábra) végzi el, amelyre egy külön metódust írtam, ahol megvizsgálja az e-mail cím formátumát. Ezután megvizsgálja, hogy az e-mail cím be van-e regisztrálva. Az e-mail cím után megvizsgálja a felhasználónevet, hogy nem üres-e az értéke, valamint azt, hogy a felhasználónév nem-e foglalt. Ezt követően a jelszót validálja (24. ábra), melyre külön metódust írtam, ahol megvizsgálja a jelszó formátumát, majd ezután megvizsgálja, hogy a két beírt jelszó megegyezik-e.

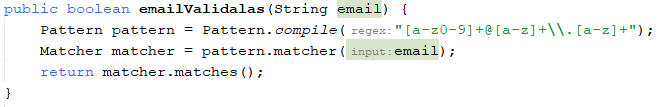
Bármelyik vizsgálatnál hiba lép fel, a rendszer párbeszéd ablak segítségével hibaüzenetet küld a felhasználónak. Azonban, ha semmilyen hiba nem lép fel az ellenőrzések során, akkor a regisztráció sikeres és ekkor minden adat tárolásra kerül az adatbázisban, majd ezután a regisztrációs ablak bezáródik és megjelenik a bejelentkezési felület.



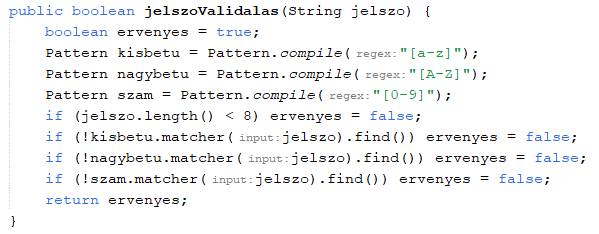
21. ábra: Regisztrációt megvalósító metódus



22. ábra: Regisztráció rögzítése az adatbázisba

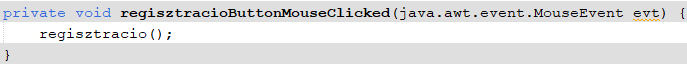


23. ábra: E-mail cím validálása



24. ábra: Jelszó validálása

A *regisztráció* gombra való kattintáskor történik meg a regisztráció.



25. ábra: Regisztráció gomb eseményvezérlője

## Menü felülete – MenuJFrame.java

A menü feladata, hogy egy helyről elérhessük a lehető legtöbb funkciót, ezeket gombok segítségével valósítottam meg, melyek eseményvezérlői bezárják a menüt és megnyitják a szükséget felületet a funkciók eléréséhez.



26. ábra: A menü felületének eseményvezérlői

## Tematikus szószedeteket tartalmazó felület - SzoszedetekJFrame.java

A felület feladata, hogy a felhasználó által elérhető tematikus szószedeteket egy helyről elérhetővé tegye. A szószedeteket gombok segítségével jelenítem meg, 3x3-as elrendezésben, melyhez a *GridLayout* elnevezésű elrendezést használtam. A *GridLayout* segítségével egy egyenlő téglalapokra osztott téglalaprácsban helyezhetjük el a komponenseket.



27. ábra: Tematikus szószedeteket tartalmazó felület elrendezésének beállítása

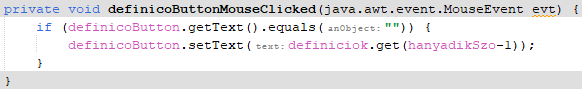
Mivel a szószedetek száma nem rögzített egy felhasználónál sem, mivel lehet létrehozni és törölni is azokat, így a gombokat a felület megnyitásakor helyezem el, valamint ekkor adom a gombokhoz az eseménykezelőket. Az eseményvezérlők kattintáskor megnyitják az adott témakör szavait tartalmazó szótanuló felületet.



28. ábra: Gombok és azok eseményvezérlőjének definiálása a tematikus szószedeteket tartalmazó felületen

## Szókártyákat tartalmazó szótanulói felület – SzotanuloJFrame.java

A felület segítségével tudjuk elkezdeni megtanulni a szókártyákon lévő szavakat. A nyelv kiválasztását rádió gombokkal oldottam meg. A szókártyákat gombok segítségével jelenítettem meg, ahol a felső gomb a kifejezést, míg az alsó gomb a definíciót tartalmazza. A definíció alapértelmezetten el van rejtve, de a felhasználó kattintással megjelenítheti.

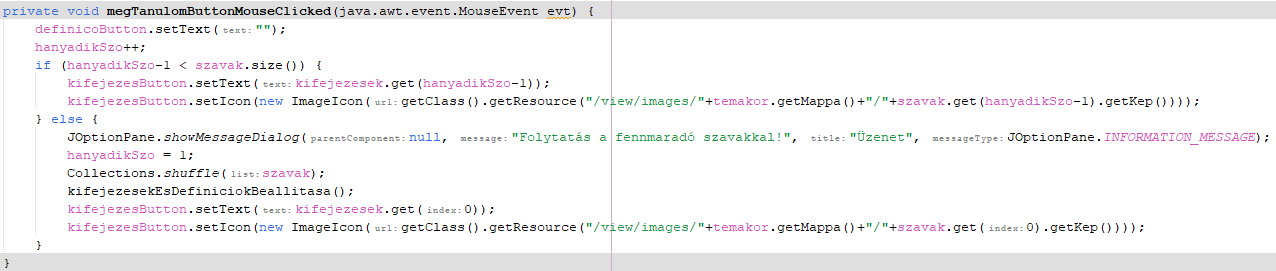


29. ábra: Definíció gomb tartalmának megjelenítése

A felhasználónak van lehetősége eldönteni, hogy az adott szót elsajátította-e vagy sem. Erre két gomb kell szolgál, az egyik a *már tudom* gomb, míg a másik a *még* *tanulom* gomb. Ezek eseménykezelője elvégzi a szükséges lépéseket, hogy a tanulási folyamat gördülékeny legyen.



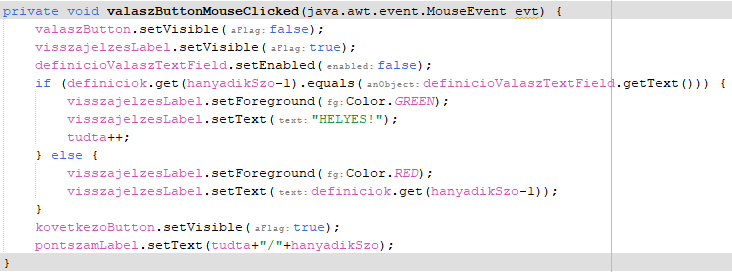
30. ábra: „Már tudom” gomb eseményvezérlője



31. ábra: „Még tanulom” gomb eseményvezérlője

## Tesztfeladatsor felülete – TesztJFrame.java

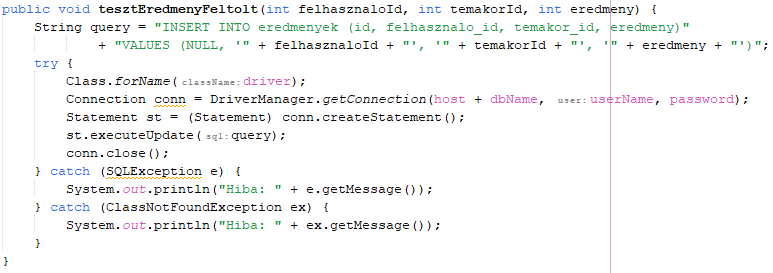
A felület segítségével tudjuk leellenőrizni a tanulási folyamatunk sikerességét. A nyelv kiválasztását rádió gombokkal oldottam meg. A teszt elindításakor megjelenítem az első kifejezést. A felhasználó a beviteli mezőbe írt válaszát a *válasz* gomb leütésével véglegesítheti, amely után visszajelzést kap, majd a *következő* gombra kattintva következik egy újabb szó. Amennyiben a teszt végére ért a felhasználó, visszajelzést kap a teszt eredményéről, valamint elmentem a született eredmény adatbázisba is.



32. ábra: Válasz gomb eseményvezérlője



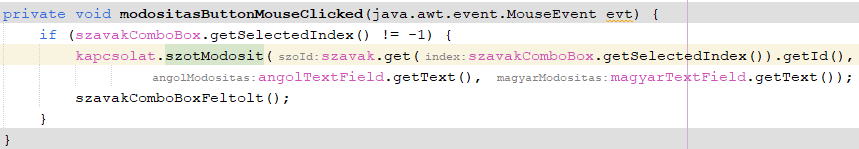
33. ábra: Következő szót megjelenítő gomb eseményvezérlője



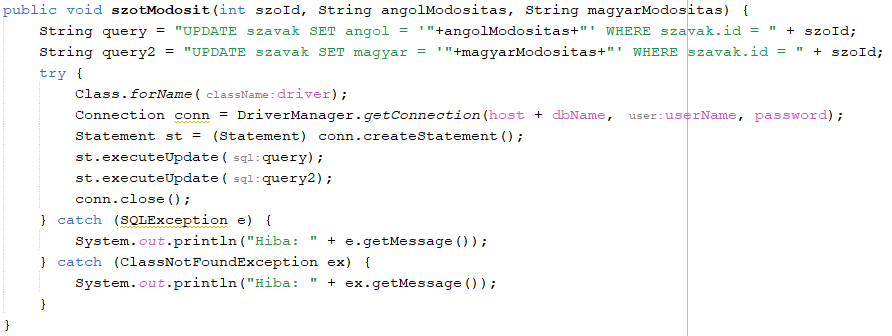
34. ábra: Teszt eredményének feltöltése adatbázisba

## Szerkesztőnézet felülete – SzerkesztoJFrame.java

A szószedetek szerkesztését három részre osztva valósítottam meg. A szavak szerkesztésekor két darab legördülő lista segítségével lehet kiválasztani, hogy melyik szót szeretnénk módosítani vagy törölni. Módosításkor a *módosítás* gomb eseményvezérlője a két beviteli mező értékére módosítja az angol és a magyar megfelelőket az adatbázisban.

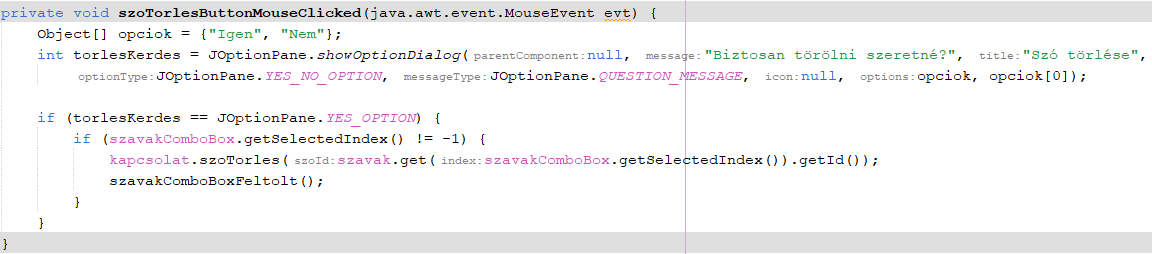


35. ábra: Szó módosítása gomb eseményvezérlője

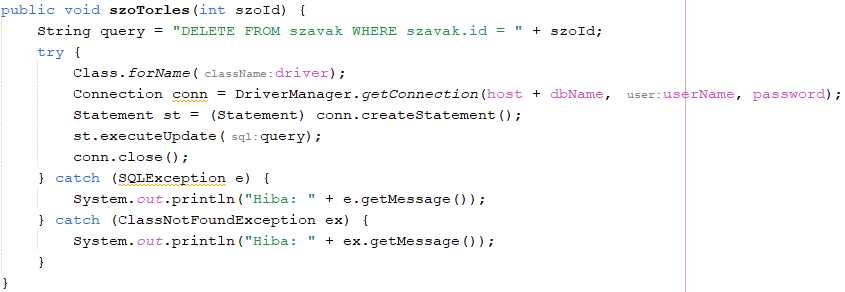


36. ábra: Szó módosítása az adatbázisban

A szó törlésekor a legördülő lista által kiválasztott szó törlése kerül az adatbázisból a *törlés* gomb eseményvezérlője által. A törlés előtt egy párbeszéd ablak segítségével biztosítom, hogy a felhasználó eldönthesse biztosan szeretné-e törölni az adott szót.

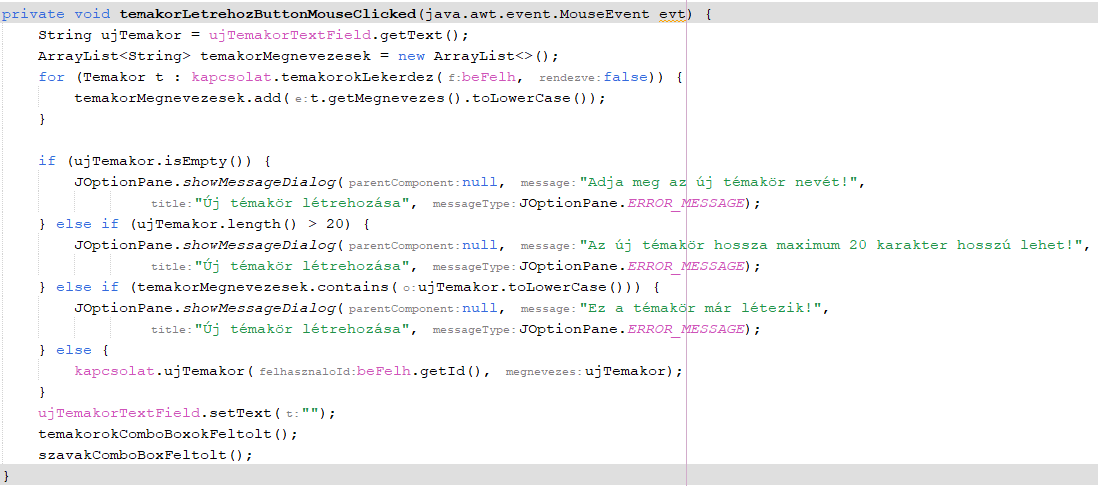


37. ábra: Szó törlése gomb eseményvezérlője

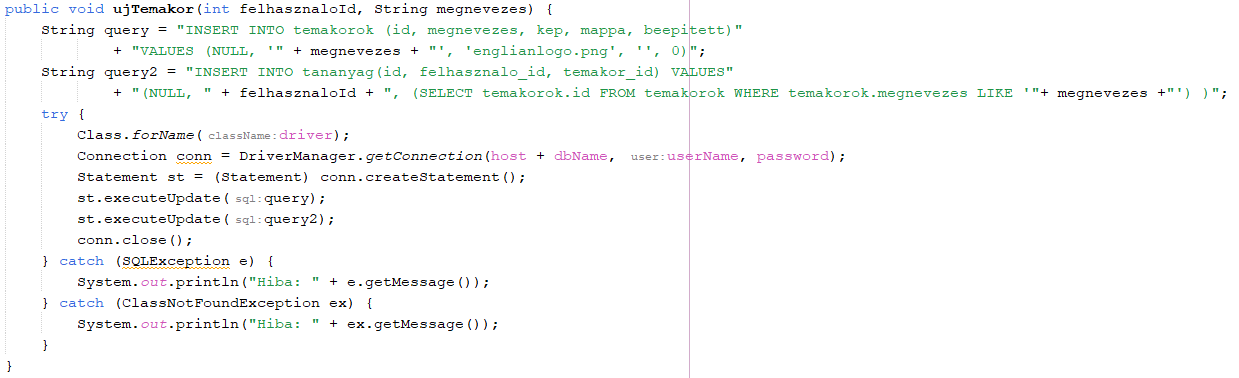


38. ábra: Szó törlése az adatbázisból

Új szószedet/szó hozzáadásakor a *témakör létrehozása* gomb eseményvezérlője a beviteli mezőbe beírt témakört létrehozza és eltárolja az adatbázisban. A témakör létrehozásakor esetlegesen felmerülő hibákat párbeszéd ablakok segítségével jelzem a felhasználónak.

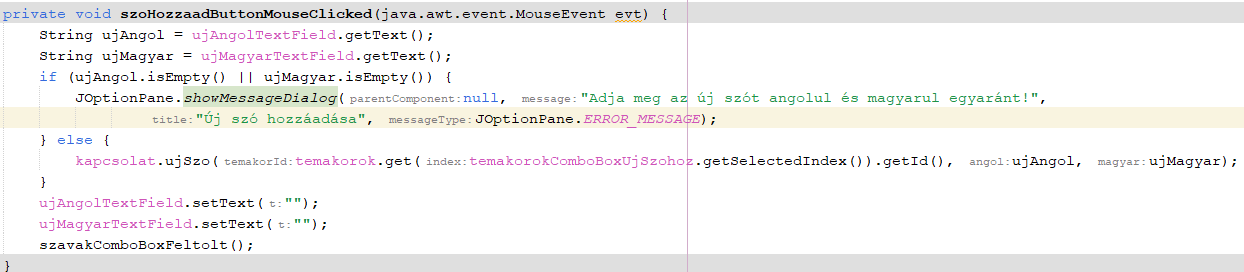


39. ábra: Témakör létrehozása gomb eseményvezérlője

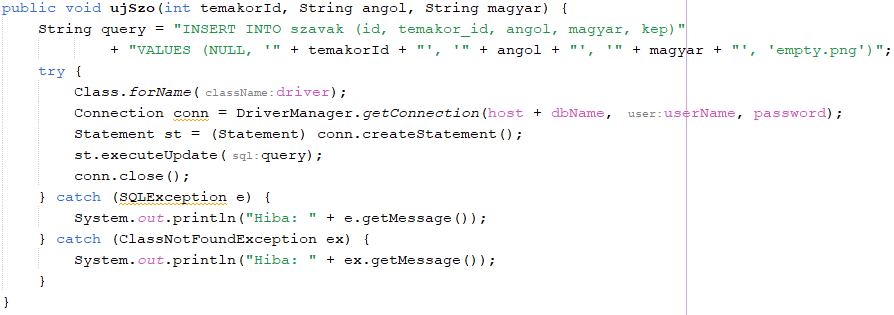


40. ábra: Új témakör feltöltése az adatbázisba

Új szó hozzáadásakor a *szó hozzáadása* gomb eseményvezérlője a legördülő lista által kiválasztott szószedethez adja hozzá a beviteli mezőkbe írt angol és magyar szavakat, melyeket adatbázisba eltárolja.

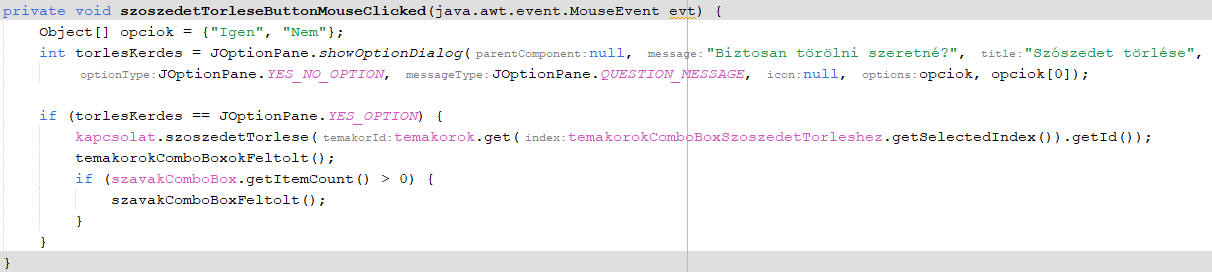


41. ábra: Új szó hozzáadása gomb eseményvezérlője

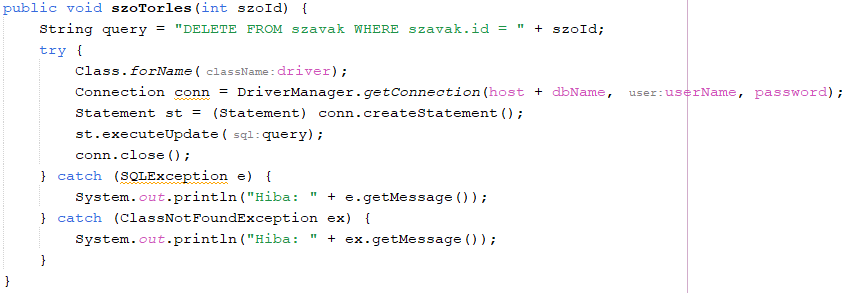


42. ábra: Új szó feltöltése adatbázisba

Szószedet törlésekor a *törlés* gomb eseményvezérlője a legördülő lista segítségével kiválasztott szószedetet törli az adatbázisból. A törlés előtt egy párbeszéd ablak segítségével biztosítom, hogy a felhasználó eldönthesse biztosan szeretné-e törölni az adott szószedetet.



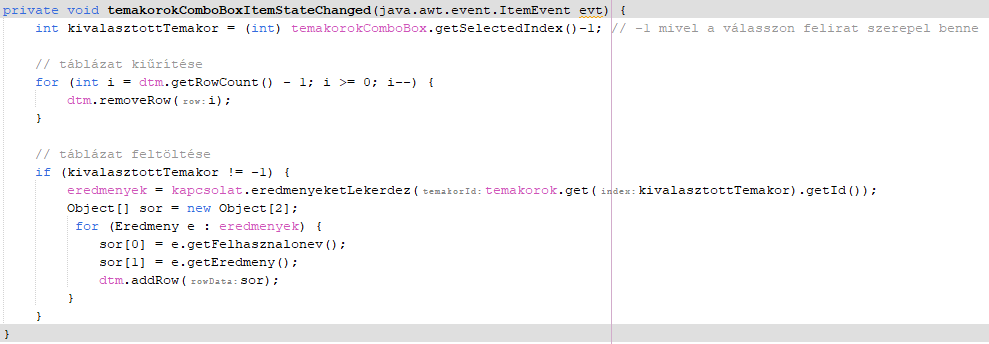
43. ábra: Szószedet törlése gomb eseményvezérlője



44. ábra: Szószedet törlése az adatbázisból

## Ranglista felület – RanglistaJFrame.java

A felület az eredményeket egy táblázatban jeleníti meg a legördülő listában szereplő témakör alapján. A táblázat az eredményt csökkenő sorrendben jeleníti meg, tartalmazva a felhasználónevet, valamint a százalékos eredményt.



45. ábra: Ranglista megjelenítése

## Fejlesztés során felhasznált szoftverek

### Apache NetBeans integrált fejlesztői környezet bemutatása

Az Apache NetBeans egy integrált fejlesztői környezet Java, PHP és más programozási nyelveken való fejlesztéshez. Az alkalmazásom fejlesztése során azért erre a fejlesztői környezetre esett a választásom, mert rendelkezik grafikus fejlesztői felülettel, ami nagyban megkönnyíti a felületek grafikus elkészítését.

Fejlesztés során használt verzió: Apache NetBeans IDE 16

### XAMPP webszerver bemutatása

Az alkalmazásom adatbázisban tárolja a felhasználók, a témakörök, a szavak, és az eredmények adatait. A XAMPP volt a segítségem ebben, amely egy nyílt forráskódú, ingyenesen használható webszerver. Az adatbázis elkészítéséhez a phpMyAdmin felületét használtam, melyet a localhost-on tudunk elérni. Az alkalmazás hibátlan működéséhez szükséges az Apache és a MySQL modulok futtatása, mivel ők felelnek az alkalmazás adatbázishoz való hozzáféréséért. [17]

Fejlesztés során használt verzió: XAMPP 8.2.4

### Draw.io szoftver bemutatása

A Draw.io egy diagramkészítő szoftver, mely online is elérhető, de akár telepíthető is a számítógépünkre. Készíthetünk vele diagramokat, grafikonokat, folyamatábrákat, UML diagramokat és ezeken kívül még nagyon sok mindent, melyek segíthetik a fejlesztési munkát. Szakdolgozatom során a szoftver segítségével készítettem el az alkalmazás használati eset diagramját, valamint megterveztem minden felület drótvázát.

Fejlesztés során használt verzió: draw.io 21.2.1

# Eredmények

## Bejelentkezési felület

Az alkalmazás indulásakor a bejelentkezési felület fogadja a felhasználót, ahol be tud lépni a regisztráció során megadott felhasználónévvel és jelszóval, ha a *bejelentkezés* gombra kattint, vagy ha lenyomja az *Enter* billentyűt. Amennyiben a felhasználó nincs regisztrálva, az *új fiók létrehozása* gombra kattintva, átirányítja a regisztrációs felületre. A szem ikonra kattintva a jelszó láthatóságát állíthatja.



# Összefoglalás

# Irodalomjegyzék

1. Az angol nyelv rövid történelmi áttekintése:

<https://behappynyelviskola.hu/az-angol-nyelv-tortenete/?cn-reloaded=1>

(2023. 04. 09.)

1. Az angol nyelv rövid történelmi áttekintése:

<https://tudnivalo.webnode.hu/angol/az-angol-nyelv-eredete/> (2023. 04. 09.)

1. Az oktatás és a számítástechnika kapcsolatának történelmi bemutatása:

<https://elearninginfographics.com/evolution-classroom-technology-journey-pen-keyboard-infographic/> (2023. 04. 09.)

1. Az oktatás és a számítástechnika kapcsolatának történelmi bemutatása:

<https://elearninginfographics.com/the-history-of-classroom-technology-infographic/> (2023. 04. 09.)

1. Horváth Ildikó, A digitális oktatás legújabb eszközei és módszerei, Széchenyi István Egyetem, 2017:

<https://www.hte.hu/documents/3102649/4239802/HT_2017_1_2_Horvath.pdf>

(2023. 04. 09.)

1. 1. ábra: Duolingo logó:

<https://d35aaqx5ub95lt.cloudfront.net/images/f2a2e608c854822ad2563a09595e7827.png> (2023. 04. 10.)

1. Duolingo bemutatása:

<https://www.angliaikisokos.com/az-5-legjobb-ingyenes-nyelvoktato-alkalmazas/>

(2023. 04. 10.)

1. 2. ábra: Quizlet logó:

<https://play-lh.googleusercontent.com/hiQHKRhpuGu4pWAFhpto9H7qWKSdX-BjKbDAtZYgm_jfoD0cN7MCllGOn6L3XWo-6Q> (2023. 04. 10.)

1. Quizlet bemutatása:

<http://refpedi.hu/sites/default/files/hir_kepek/A%20Quizlet%20feladatkészítő_2018_január_BMA_VÉGLEGES.pdf> (2023. 04. 10.)

1. 3. ábra: Java programozási nyelv logója:

<https://logos-download.com/wp-content/uploads/2016/10/Java_logo_icon.png>

(2023. 04. 10.)

1. Kovács Zsuzsanna, JAVA programozási nyelv NetBeans fejlesztőkörnyezetben, Observans Kft., Budapest, 2009:

<https://informatika.gtportal.eu/letoltes/SZINFO13_TJ.pdf> (2023. 04. 10.)

1. Tömösközi Péter, Programozás Javában, Eger, 2013:

<https://mek.oszk.hu/14200/14282/pdf/14282.pdf> (2023. 04. 10.)

1. 4. ábra: A forráskódtól a programig:

<https://www.scaler.com/topics/java/how-java-program-works/> (2023. 04. 10.)

1. A Java nyelv főbb jellemzői:

<https://bluebird.hu/java/> (2023. 04. 10.)

1. Nagy Gusztáv, Java programozás, 2007:

<https://nagygusztav.hu/sites/default/files/csatol/java_programozas_1.3.pdf>

(2023. 04. 11.)

1. MVC (Model-View-Controller):

<https://jankajanos.wordpress.com/2008/02/15/mvc-model-view-controller-tervezesi-minta/> (2023. 04. 27.)

1. XAMPP bemutatása:

<https://www.educba.com/what-is-xampp/> (2023. 04. 30.)

# Ábrajegyzék

[1. ábra: Duolingo logó [6] 11](#_Toc133786588)

[2. ábra: Quizlet logó [8] 12](#_Toc133786589)

[3. ábra: Java programozási nyelv logója [10] 14](#_Toc133786590)

[4. ábra: A forráskódtól a programig [13] 15](#_Toc133786591)

[5. ábra: Regisztrációs felület drótváza 21](#_Toc133786592)

[6. ábra: Bejelentkezési felület drótváza 22](#_Toc133786593)

[7. ábra: Menü felületének drótváza 24](#_Toc133786594)

[8. ábra: Szószedteket tartalmazó felület drótváza 25](#_Toc133786595)

[9. ábra: Szókártyákat tartalmazó szótanulói felület paneljeinek drótvázai 26](#_Toc133786596)

[10. ábra: Tesztfeladatsor felület paneljeinek drótvázai 27](#_Toc133786597)

[11. ábra: Szavak szerkesztése ablak drótváza a szerkesztőnézet felületén 29](#_Toc133786598)

[12. ábra: Új szószedet/szó hozzáadása ablak drótváza a szerkesztőnézet felületén 30](#_Toc133786599)

[13. ábra: Szószedet törlése ablak drótváza a szerkesztőnézet felületén 30](#_Toc133786600)

[14. ábra: Ranglista felület drótváza 31](#_Toc133786601)

[15. ábra: Használati eset diagram 31](#_Toc133786602)

[16. ábra: Az adatbázis táblái és az azok közötti kapcsolatok 32](#_Toc133786603)

[17. ábra: Az alkalmazásom MVC struktúrája 34](#_Toc133786604)

[18. ábra: Bejelentkezést megvalósító metódus 35](#_Toc133786605)

[19. ábra: Felhasználó adatainak lekérdezése adatbázisból 36](#_Toc133786606)

[20. ábra: MD5 hash-függvény 36](#_Toc133786607)

[21. ábra: Regisztrációt megvalósító metódus 37](#_Toc133786608)

[22. ábra: Regisztráció rögzítése az adatbázisba 37](#_Toc133786609)

[23. ábra: E-mail cím validálása 37](#_Toc133786610)

[24. ábra: Jelszó validálása 38](#_Toc133786611)

[25. ábra: Regisztráció gomb eseményvezérlője 38](#_Toc133786612)

[26. ábra: A menü felületének eseményvezérlői 38](#_Toc133786613)

[27. ábra: Tematikus szószedeteket tartalmazó felület elrendezésének beállítása 39](#_Toc133786614)

[28. ábra: Gombok és azok eseményvezérlőjének definiálása a tematikus szószedeteket tartalmazó felületen 39](#_Toc133786615)

[29. ábra: Definíció gomb tartalmának megjelenítése 40](#_Toc133786616)

[30. ábra: „Már tudom” gomb eseményvezérlője 40](#_Toc133786617)

[31. ábra: „Még tanulom” gomb eseményvezérlője 40](#_Toc133786618)

[32. ábra: Válasz gomb eseményvezérlője 41](#_Toc133786619)

[33. ábra: Következő szót megjelenítő gomb eseményvezérlője 41](#_Toc133786620)

[34. ábra: Teszt eredményének feltöltése adatbázisba 41](#_Toc133786621)

[35. ábra: Szó módosítása gomb eseményvezérlője 41](#_Toc133786622)

[36. ábra: Szó módosítása az adatbázisban 42](#_Toc133786623)

[37. ábra: Szó törlése gomb eseményvezérlője 42](#_Toc133786624)

[38. ábra: Szó törlése az adatbázisból 42](#_Toc133786625)

[39. ábra: Témakör létrehozása gomb eseményvezérlője 43](#_Toc133786626)

[40. ábra: Új témakör feltöltése az adatbázisba 43](#_Toc133786627)

[41. ábra: Új szó hozzáadása gomb eseményvezérlője 43](#_Toc133786628)

[42. ábra: Új szó feltöltése adatbázisba 44](#_Toc133786629)

[43. ábra: Szószedet törlése gomb eseményvezérlője 44](#_Toc133786630)

[44. ábra: Szószedet törlése az adatbázisból 44](#_Toc133786631)

[45. ábra: Ranglista megjelenítése 45](#_Toc133786632)

# Táblázatjegyzék

[I. táblázat: Primitív adattípusok [15] 16](#_Toc133750629)