

**RUANDER oktatási Kft.**

# **SZAKDOLGOZAT**

Radnóti Róbert  
Szoftverfejlesztő

Budapest  
2022

**RUANDER Oktatási Kft.**

# **ENGLISH Grammar**

## **Angol igeidő gyakorló program**

Konzulens:  
Kis Balázs

Készítette:  
Radnóti Róbert  
Szoftverfejlesztő

# Tartalomjegyzék

---

<b>1. BEVEZETÉS .....</b>	<b>4</b>
<b>3. FELHASZNÁLÓI DOKUMENTÁCIÓ.....</b>	<b>5</b>
3.1. Mire használható a szoftver? .....	5
3.2. A szoftver ablakai .....	5
3.2.1.Kezdőképernyő és indítás (Start) .....	5
3.2.2.Felhasználónév megadása (user name/felhasználó neve) .....	6
3.2.3.Főmenü .....	6
3.2.4.Eredmény lista (Score/Pontszámok) .....	7
3.2.5.Választásos feladatok (Multi choice) .....	8
3.2.6.Rendezéses feladatok (Arrange sentence).....	9
3.2.7.Feladattípusok kiválasztása .....	10
3.2.8.Hibaesetek .....	11
3.2.9.Rendszerkövetelmények.....	12
<b>4. FEJLESZTŐI DOKUMENTÁCIÓ.....</b>	<b>13</b>
4.1. Feladat specifikáció .....	13
4.1.1.Funkcionalitás .....	13
4.1.2.Kinézet.....	13
4.2. Adatbázis .....	14
4.3. Package Diagram .....	15
4.4. Use Case Diagram .....	16
4.5. Sztereotípus osztálydiagram.....	17
4.6. Normal osztálydiagram.....	18
4.7. Sequence diagram.....	21
4.8. State Machine diagram .....	25
4.9. Deployment Diagram .....	28
<b>5. ÖSSZEGZÉS .....</b>	<b>29</b>
5.1. Megvalósult project .....	29
5.2. Továbbfejlesztési lehetőségek .....	29
<b>6. IRODALOMJEGYZÉK.....</b>	<b>30</b>

# 1. BEVEZETÉS

---

Az angol nyelv gyakorlását segítő programom elkészítésének ötlete akkor fogalmazódott meg benne, amikor nyelvtanulásom során hiába kerestem olyan használható gyakorló megoldást, amivel olyan igeidőket is gyakorolni tudok, amelyeket nem vagy nagyon ritkán használunk. Szükségem volt egy laptopral használható gyors programra, ami kicsit más mint amiket eddig találtam. Ugyanis a magyar nyelvben három igeidőt, (múlt-, jelen és jövő időt), az angol nyelvben ehhez képest tizenkét igeidőt használhatunk. A három alap igeidőn kívül a beszélő mondandójának, alanyának és az aktuális helyzetének megfelelően még négy változatot különböztettük meg (egyszerű, folyamatos, befejezett, befejezett folyamatos). Ennek megértéséhez kellett valahogy összeállítanom egy táblázatot. Vízszintesen lehetnének a jelen-, a múlt-, és a jövő idő, függőlegesen pedig a többi négy hiszen ezek minden a három alapidőben pontosan ugyanúgy jelen vannak.

A gyakorláshoz szükséges programomban pontosan így helyeztem el őket. Így a táblázat viszonylag könnyen átlátható lett. A színek is kiemelik és egyértelműsítik a választási lehetőségeket. A pontszámos rendszert pedig azért illesztettem hozzá, hogy segítségével a haladás mértékét is jobban követhetem. Ezenkívül játékos motivációt is ad a gyakorlóknak. Mivel mindenkinél saját nevét, becenevét, azonosítóját is meg kell adni az indítást követően, így akár versenyezési lehetőséget is jelenthet a gyakorlóknak.

Kétféle gyakorlási módot próbálhatunk ki.

1. *Mondatrendezés:* A példamondatok összekevert szavait kell nyelvtanilag helyes sorrendbe állítani.
2. *Választásos feladatok:* A feladatmondatban egyes helyeken a szavak ki vannak húzva, és négy lehetséges megoldás közül kell kiválasztani a nyelvtanilag megfelelőt.

A programból való kilépés után az addig elérte aktuális pontszám elmentésre kerül, és amikor újra megcsináljuk a feladatsort, ez a pontszám frissül. Ez is motiváló tényező. A mondat egységeinek nyelvtani rendezése egy kicsit nehezebb feladat, mert itt először ki kell találni a mondat jelentését, utána pedig helyes sorrendbe rendezni őket. Ezért kétszer annyi időt terveztem a megoldásra, mint a választásosnál. Eleinte kicsit nehéznek tűnik, de gyakorlással már elérhető a szintidő.

### **3. FELHASZNÁLÓI DOKUMENTÁCIÓ**

---

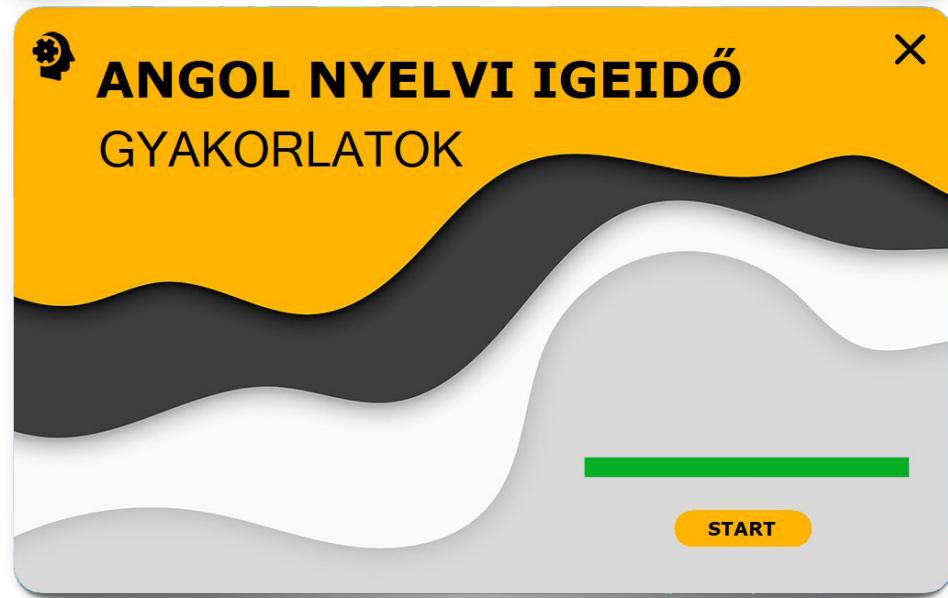
#### **3.1. Mire használható a szoftver?**

Az angol igeidők gyakorlásra, megtanulására illetve gyakorlására használható a program.

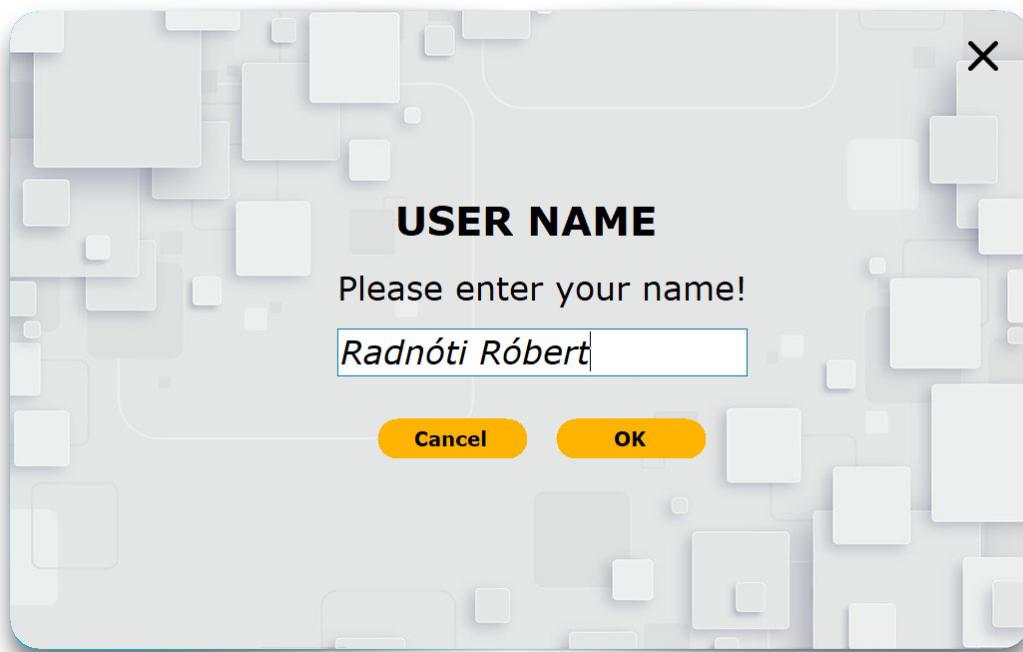
#### **3.2. A szoftver ablakai**

##### **3.2.1. Kezdőképernyő és indítás (Start)**

A Kezdőképernyőn lehet elindítani a programot, ha a végigfutó zöld csík eléri a maximális hosszát a Start gombbal. A cím felett elhelyezett fej ikon megnyomásával pedig át lehet kapcsolni a program feliratait magyarról angolra és vissza.



### 3.2.2. Felhasználónév megadása (user name/felhasználó neve)



Ezen az ablakon lehet megadni a **felhasználónevet**. Ezt az ablakot nem lehet üresen hagyni, mert addig nem megy tovább a program illetve hiba jelzést ad.

### 3.2.3. Főmenű

The image shows a main menu for English grammar. The title is 'English Gramar'. Below it is a greeting: 'Üdvözöllek Radnóti Róbert'. On the right, there is a 'Score' section with a bar chart showing 1 point. The main part is a grid of verb forms:

Tenses	Past	Present	Future
Simple	Past Simple	Present Simple	Future Simple
Perfect	Past Perfect	Present Perfect	Future Perfect
Continuous	Past Continuous	Present Continuous	Future Continuous
Perfect Continuous	Past Perfect Continuous	Present Perfect Continuous	Future Perfect Continuous

A főmenüben megjelenik egy üdvözet a felhasználó nevével. A **Score/Pontszámok** feliratú ikon megnyomásával megnézhetjük az eddig elért eredményeinket listába rendezve. A táblázat egyes elemire való klikkéssel indíthatjuk el a feladatokat. Ekkor kell kiválasztani a gyakorlatok típusát, melyek lehetnek választós illetve mondatrendezéses feladatsorok. A **Tenses/Igeidők** kiválasztásakor az összes igeidőből, a **Past, Present, Future menü** kiválasztásakor az alatta lévő oszlopban található igeidőből és a **Simple, Perfect, Continuous, Perfect Continuous**, kiválasztásakor a sorban lévő igeidőből fogunk véletlenszerűen kiválasztott feladatokat kapni. A főmenü bezárásával a felhasználó nevének beállítására jutunk vissza. Ekkor megadható akár egy másik felhasználó is, vagy újra lehet kezdeni a gyakorlást. Ha ezt is bezárjuk a akkor a főképernyőt kapjuk vissza.

### 3.2.4. Eredmény lista (Score/Pontszámok)

PONTSZÁMOK				
FELHASZNÁLÓ NEVE / IGEIDŐ / PONTSZÁM				
Robert	-	PastSimple	-	5
Katalin	-	PastSimple	-	8
Zsolt	-	PastSimple	-	15

Törlés Bezáras

A pontszámok a nevek és az igeidők szerint vannak rendezve. Ha ismét ugyanazon a néven és igeidőn gyakorlunk akkor nem kerül be újabb bejegyzés a listába, hanem felülíródik a már meglévő feladat pontszáma az új eredménnyel. Lehetőségünk van törölni is a listából ha már nagyon sok, vagy átláthatatlan. Ezt a sor kijelölésével és **Törlés/Delete** gombra klikkelve tehetjük meg.

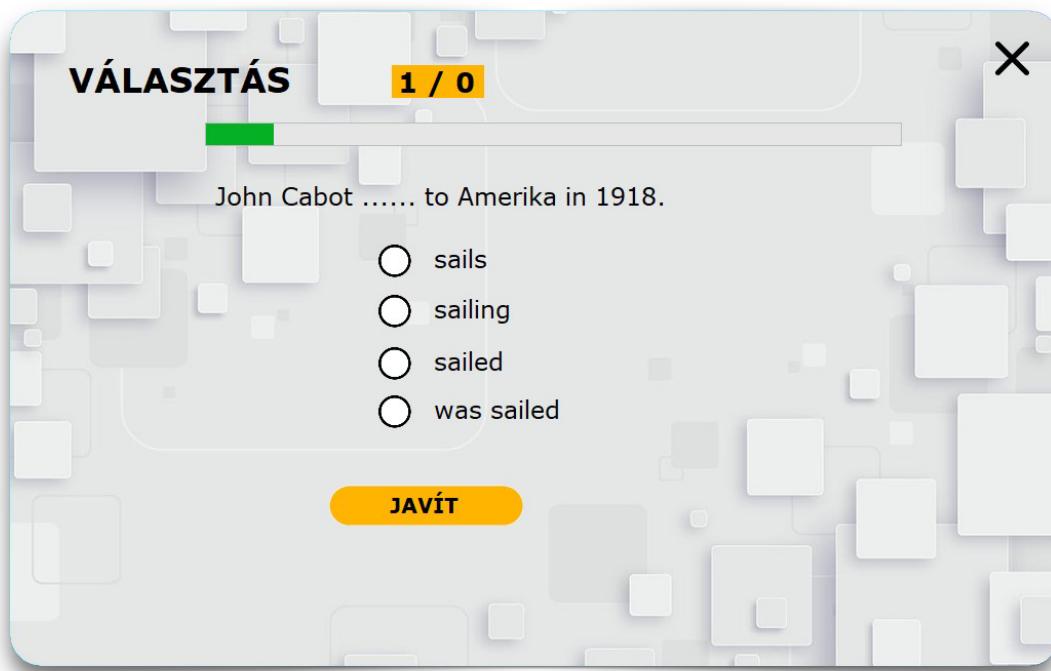
### 3.2.5. Választásos feladatok (Multi choice)

**VÁLASZTÁS** **1 / 0** **X**

John Cabot ..... to Amerika in 1918.

- sails
- sailing
- sailed
- was sailed

**JAVÍT**



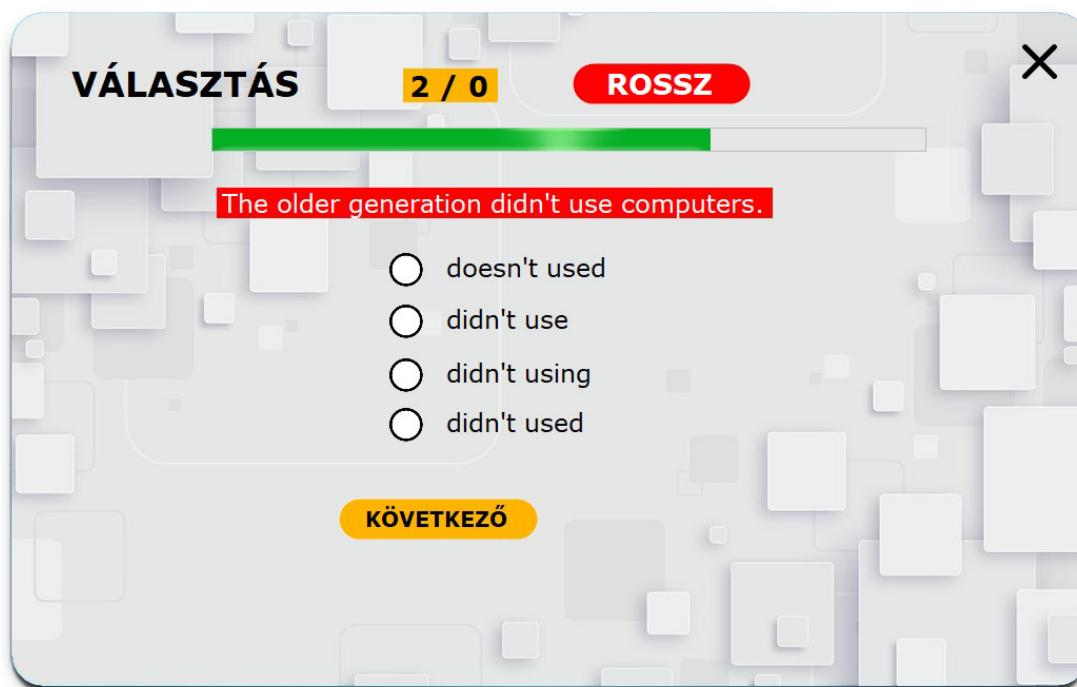
A választásos feladatoknál négy lehetséges válaszból kell kiválasztani a legmegfelelőbbet. Felül fut egy zöld csík, ami a megoldáshoz szükséges időt mutatja, illetve a számlálón a megoldott mondatok száma / helyes mondatok száma látható. A **Javít/Check** gombra kattintva az eredmény azonnal látható, a **Következő/Next** gombbal pedig tovább lehet menni. Ha elérünk a 20. kérdést az ablak automatikusan bezáródik és visszakerülünk a főmenübe. A feladat pontszáma elmentődik a listába.

**VÁLASZTÁS** **2 / 0** **ROSSZ** **X**

The older generation didn't use computers.

- doesn't used
- didn't use
- didn't using
- didn't used

**KÖVETKEZŐ**



### 3.2.6. Rendezéses feladatok (Arrange sentence)

**MONDATRENDEZÉS** **1 / 0**

Tedd a szavakat helyes sorrendbe!

in sailed John to Amerika Cabot 1918.

**JAVÍT**

A rendezéses feladatoknál a sorrendileg összekevert szavakat kell helyes sorrendbe tenni. A fent futó zöld csík a megoldáshoz szükséges időt mutatja, illetve a számlálón a megoldott mondatok száma / helyes mondatok száma látható. A **Javít/Check** gombra kattintva az eredmény azonnal látható a **Következő/Next** gombbal pedig tovább lehet menni. Ha elértek a 20. kérdést, az ablak automatikusan bezáródik és visszakerülünk a főmenübe. A feladat pontszáma elmentődik a listába.

**MONDATRENDEZÉS** **1 / 0** **ROSSZ**

Tedd a szavakat helyes sorrendbe!

John Cabot sailed to Amerika in 1918.

in sailed to Amerika Cabot 1918.

**KÖVETKEZŐ**

### 3.2.7. Feladattípusok kiválasztása

**PAST**  
(simple, perfect, continuous, perfect continuous)

**VÁLASZTÓS KÉRDÉSEK**

**MONDAT RENDEZÉS**

**BEZÁR**

A feladattípus kiválasztásánál egy rövid leírás után választhatunk a **Választós** és a **Rendezéses** feladatok közül. Ekkor indul el a feladat megoldásához szükséges idő mérése, és mivel a mondatrendezéses feladat nehezebb, ezért itt kétszer több idő áll rendelkezésre a kitöltéshez.

**TENSE**  
(Exercises all tenses)

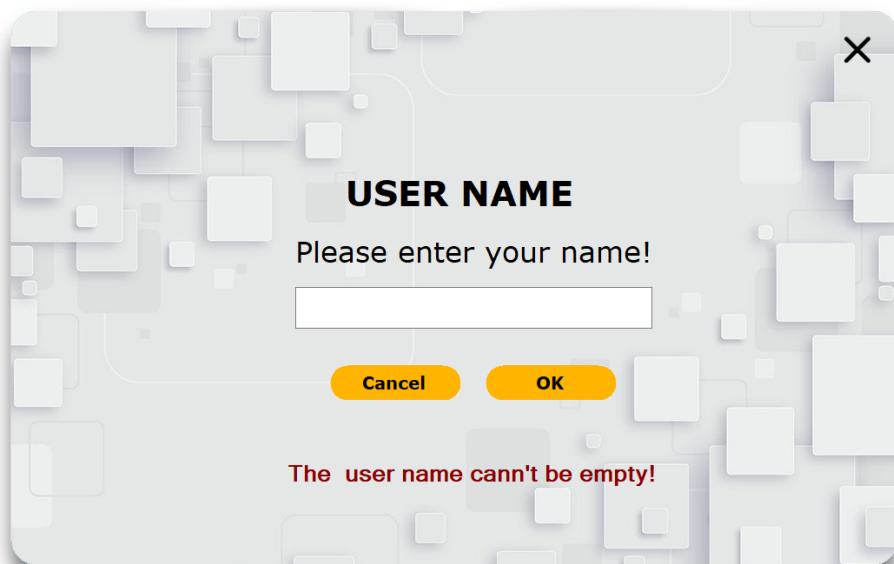
**VÁLASZTÓS KÉRDÉSEK**

**MONDAT RENDEZÉS**

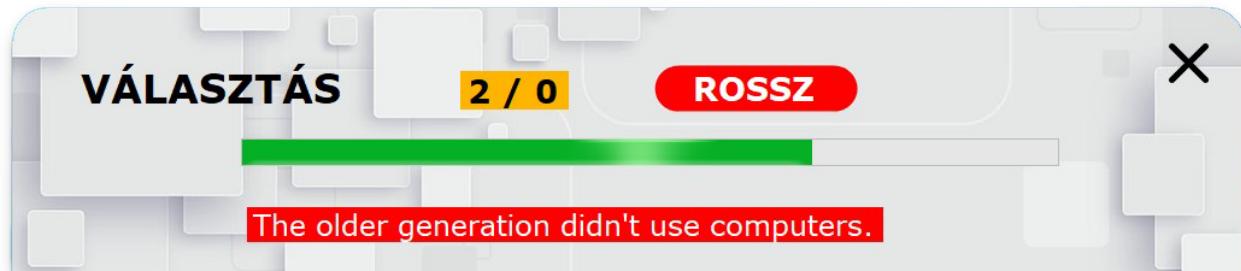
**BEZÁR**

### 3.2.8. Hibaesetek

Mivel a programban egy esetben van lehetőség be írni a **Felhasználó nevet/ User name** ezért itt üresen hagyva hibaüzenet kaphatunk.



Ezenkívül még a feladatokban, a helytelen a megoldásnál piros feliratban figyelmeztet a program, illetve szintén egy piros sávban egyidejűleg a helyes megoldás is megjelenik.



### **3.2.9. Rendszerkövetelmények**

A program MSSQL lokális adatbázist használ, ezért a Microsoft SQL Server Express telepítése szükséges. Mivel a program Microsoft Visual C# 2019-el készült Windows 10 operációs rendszer alatt ezért ennek a használata ajánlott.

#### **Minimális szoftverkövetelmények:**

Windows 10, MSSQL server

Processzor: 1 GHz vagy gyorsabb processzor

RAM: 1 GB 32-bit vagy 2 GB 64-bit

Tárhely: 16 GB a 32 bit-es windows-hoz vagy 20 GB a 64 bit-es windows-hoz

Grafikus kártya: DirectX 9

Display: 1024 x 768

#### **Ajánlott szoftverkövetelmények:**

Windows 10, MSSQL server

Processzor: 3 GHz vagy gyorsabb processzor vagy jobb

RAM: 4 GB 32-bit vagy 6 GB 64-bit vagy több

Tárhely: 50 GB a 32 bit-es windows-hoz vagy 80 GB a 64 bit-es windows-hoz vagy több

Grafikus kártya: DirectX 9 vagy jobb

Display: 1024 x 768

## 4. FEJLESZTŐI DOKUMENTÁCIÓ

---

### 4.1. Feladat specifikáció

#### 4.1.1. Funkcionalitás

Cél a windows 10 alatt futó angol nyelvi igeidők tanulását segítő program létrehozása MSSQL adatbázis kezelő segítségével. Az adatbázis a növekvő számú feladatok kezeléséhez szükséges. A szoftver kezelje a felhasználókat és nyílván tudja tartani a megoldott feladatok pontszámait. A szoftver több feladattípust tudjon használni és minden feladatra legyen időkorlát megszabva, amely lejárat esetében figyelmeztesse a felhasználót.

#### 4.1.2. Kinézet

Mivel nem számítógépes szakembereknek készül a szoftver, ezért könnyen érhető grafikus környezet kialakítása szükséges, Windows Forms segítségével. A lehető legegyszerűbb és könnyen kezelhető használata fontos szempont. A grafikus felület a GDI adta lehetőségek és saját Forms komponensek készítésével történt.

**CustomImageButton** egy leszármaztatott PictureBox komponens aminek az egér Hover és Leave tulajdonságai vannak megváltoztatva, hogy megváltozzanak, ha az egeret félévisszük, vagy pedig lehúzzuk rólá.

**CButton** egy leszármaztatott Button komponens, aminek egy radius tulajdonság lett hozzáadva. Ezzel lehet a gomb lekerekítések mértékét megadni, a GDI megjelenítés módosítása végzi el ezt a feladatot.

**DragControl** ha a Form-nak a FormBorderStyle tulajdonságát kikapcsoljuk - ami ahhoz kellett, hogy saját kinézetet adjunk neki -, elveszíti a form egérrel való mozgathatóságát is, amit ezzel a komponenssel lehet újra visszaadni.

**EllipseControl** pedig szintén a GDI segítségével módosítja a Form alakját négyzetesről egy radius megadásával lekerekített sarkúra.

## 4.2. Adatbázis

Az adatbázis MSSQL-ben készült. Négy táblából áll, az első az **Igeidők** tábla amelyeik egy azonosítóból (IgeidokID) és egy névből (IgeidokNeve) áll, ez tartalmazza az igeidők és a táblázat többi elemének azonosítóit, ettől függ az összes többi táblázat.

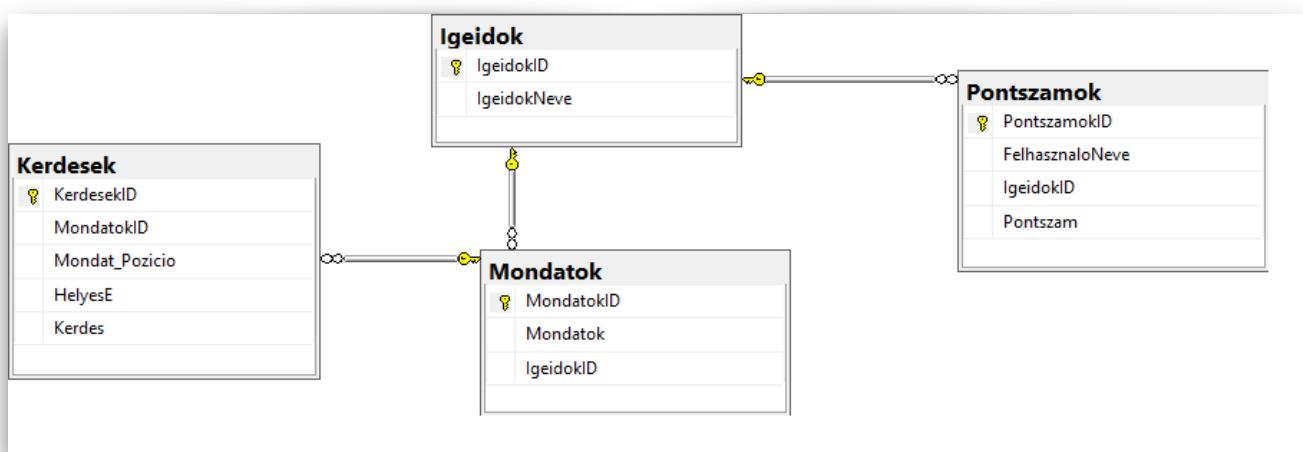
A **Pontszámok** tábla tartalmazza az feladatok eredményeit név (FelhasznaloNeve), az igeidőket (IgeidokID), és az elért pontszámot (Pontszam). A pontszámok tábla önállóan működik, csak az igeidőkhöz van benne egy azonosító a programban, lehet benne törölni és módosítani az adatokat.

A **Mondatok** tábla tartalmazza a feladatok szövegét (Mondatok) és az igeidő azonosítóját (IgeidokID) ezzel lehet be azonosítani melyik igeidőhöz tartozik mondat.

A **Kérdések** tábla tartalmazza választós feladatok kérdéseit (Kerdes), a helyes-e válasz (HelyesE) és a mondatok azonosítóját (MondatokID) illetve a kérdezni kívánt nyelvtani probléma pozícióját a mondatban (Mondat\_Pozicio), a pozíció szavakban van számolva.

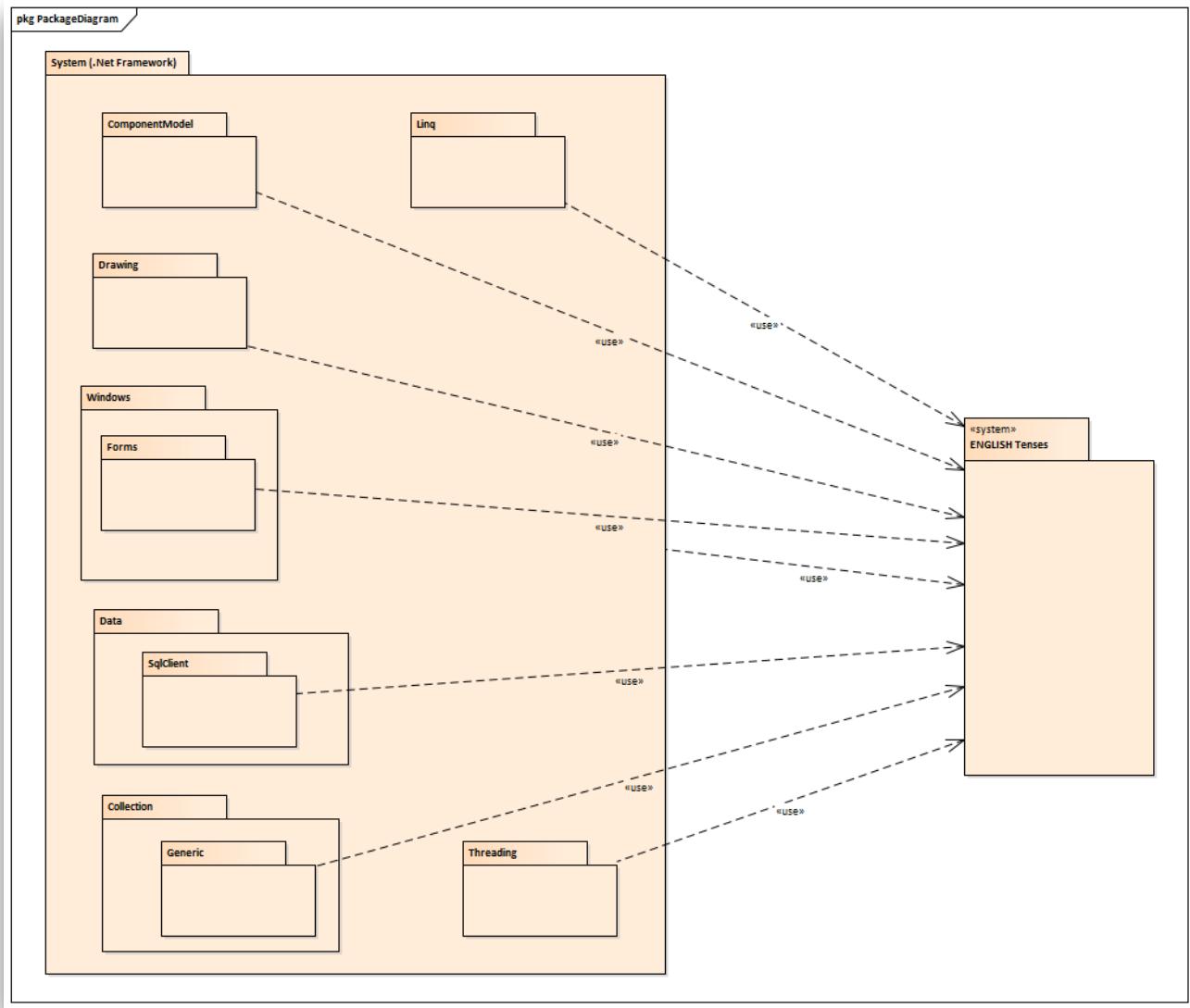
Az Igeidők, Mondatok, Kérdések tábla szorosan összefügg mert ezekből lehet pontosan összeszzedni a kétféle feladattípus kérdéseit és mondatait. A választós feladatnál az igeidők alapján lehet a mondatokat kiválasztani és az összeszedett mondatok alapján pedig a hozzátarozó kérdéseket. A rendezéses feladatnál pedig az igeidők alapján kiválasztott mondatokat lehet használni.

Minden feladattípusnál bármennyi mondat megadása lehetséges, mert a szoftver készít egy listát és ezekből összekeverve választja ki azt a 20 mondatot amit használni fog. A választásos feladatnál minden mondathoz négy kérdés tarozik.



## 4.3. Package Diagram

A program a Microsoft C# nyelven készült el, így kézenfekvő volt .NET keretrendszer adottságait fogja ki használni. A package diagram mutatja be ezeket az eszközöket.



**System.Threading:** A Timer osztályhoz van szükség rá mint például a feladatok időmérése alatt.

**System.Collection.Generic:** A gyűjtemények kezeléséhez kell.

**System.Drawing:** A Point struktúra kezeléséhez és az egyébb GDI műveletekhez szükséges.

**System.Windows.Forms:** Grafikus felület kezeléséhez szükséges.

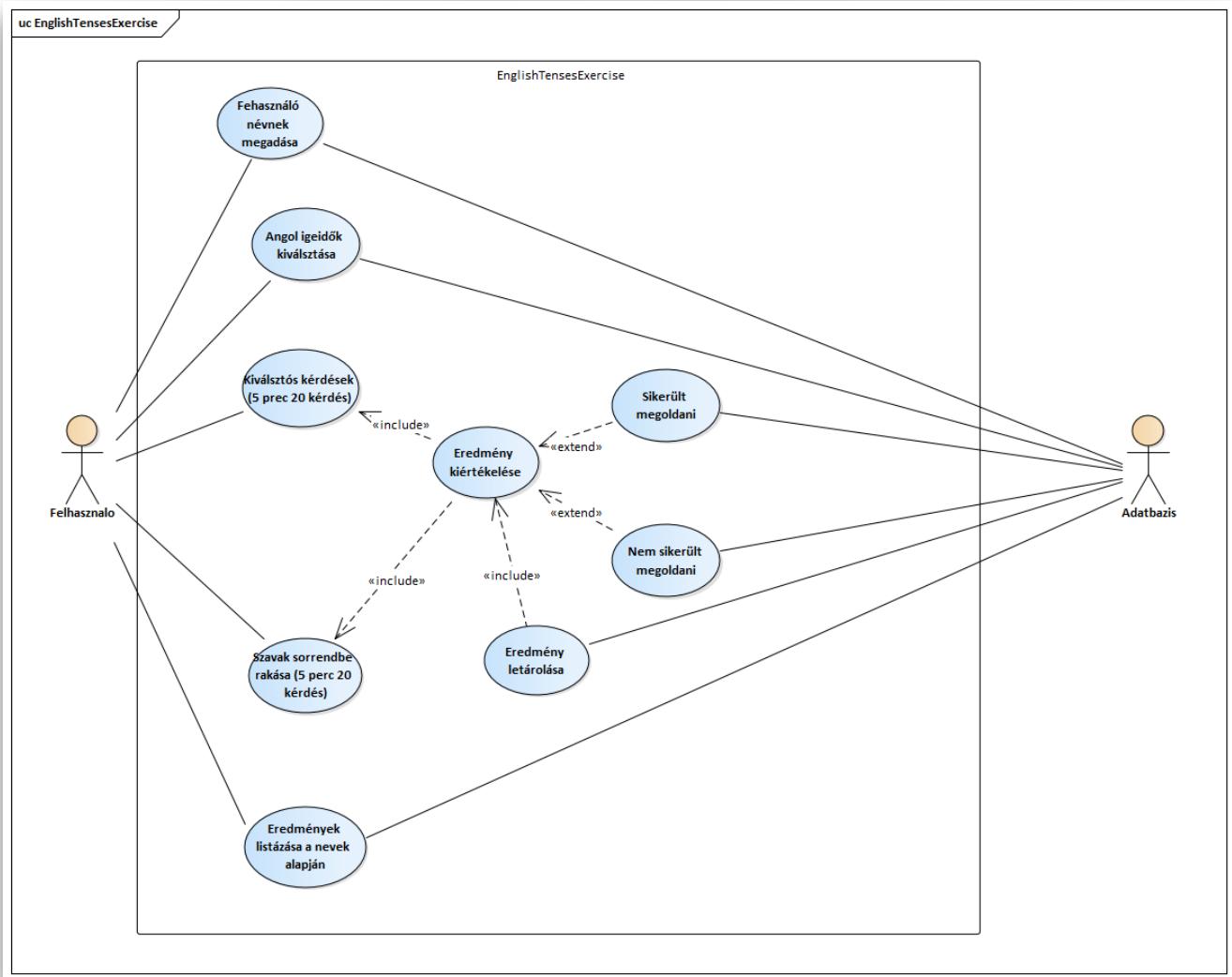
**System.Data.SqlClient:** Az adatbázis kapcsolódáshoz és a CRUD műveletekhez szükséges.

**System.Linq:** Nyelvi integrált lekérdezések névtere.

**System.ComponentModel:** Osztályokat biztosít, amelyek az összetevők és vezérlők futási és tervezési viselkedésének megvalósítására szolgálnak. Tartalmazza az alaposztályokat és interfészeket.

## 4.4. Use Case Diagram

Ez a diagram a funkciók működését írja le. Feladata, hogy leírja a rendszer szolgáltatásait, funkcionalitását és azt, hogy az adott funkcionalitást, szolgáltatást pontosan milyen felhasználók, vagy felhasználói csoportok érik el. Leírja, hogy a rendszernek pontosan mit kell majd tudnia. Követelmény feltáráshoz használatos.



A diagrammon az látszik, hogy a felhasználó meg tudja adni a nevét, és az tárolásra került az adatbázisban. Meg tudja nézni az igeidők táblázatát és ki tudja választani a gyakorolni kívánt esetet, a kiválasztott igeidők alapján kérdések jönnek létre. Két különböző feladattípus lehetséges kiválasztani, mindegyikre 5 perc idő áll rendelkezésre és 20 kérdést kell megválaszolni. Ha megoldás értékelhető, akkor letárolásra kerül az adatbázisban. A felhasználónak lehetősége van megnézni az elért eredményeket egy listában.

## 4.5. Sztereotípusos osztálydiagram

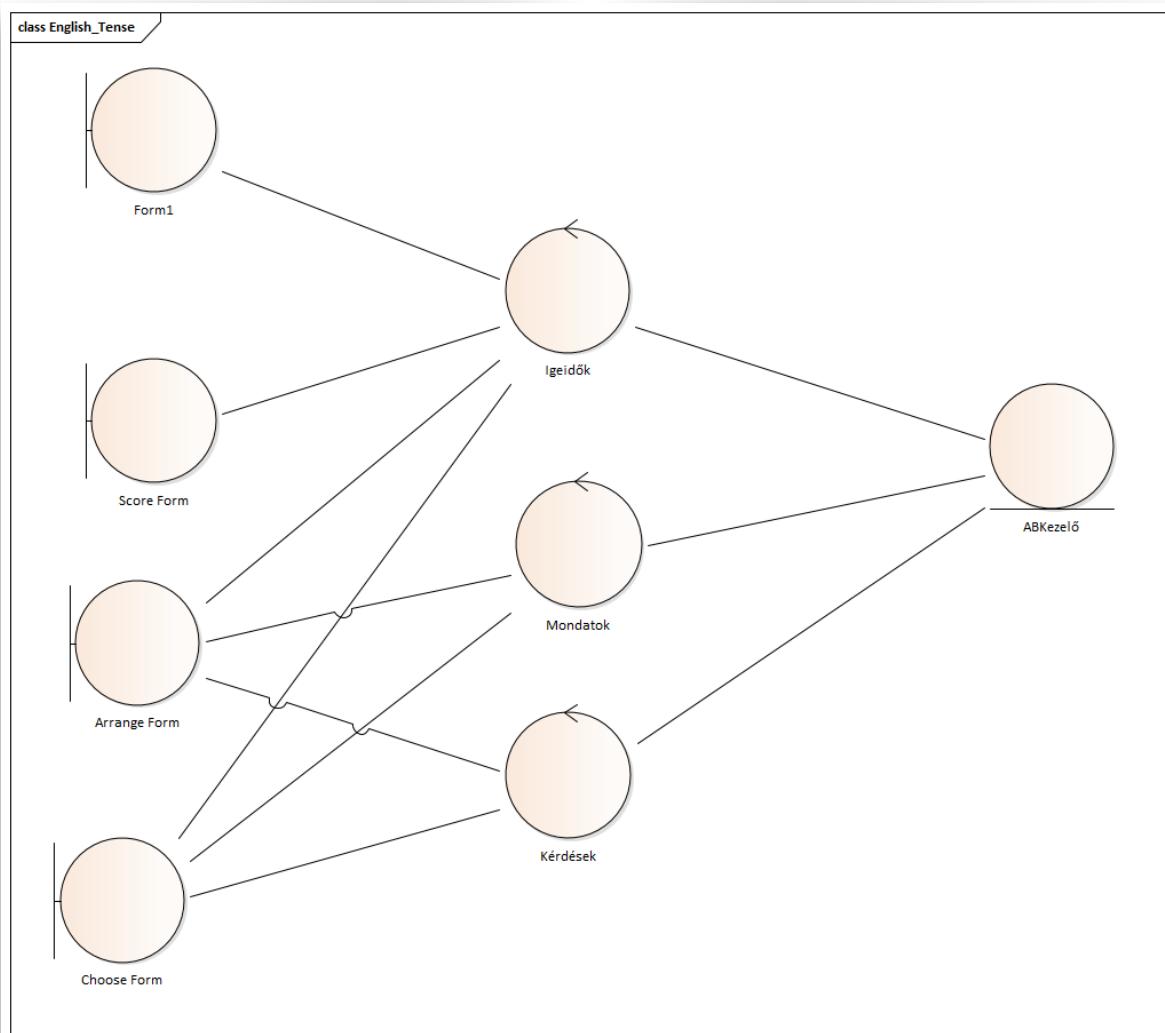
A sztereotípusos osztálydiagram a rendszer tagozódását, az egyes osztályok feladatköre szerint ábrázolja, így betekintést enged a rendszer osztállyszerkezeti felépítésébe.

Három típusa létezik:

**Boundary osztályok:** A felhasználóval való kapcsolatos osztályok. **Form1** - az alap megjelenítési forma, **Score** - A pontszámok és eredmények megjelenítése és szerkesztése, **Arrange** - a rendezős feladatok megjelenítése és használata, **Choose** - a választós feladatok megjelenítése és használata.

**Control osztályok:** Általános osztályok, ami adattárolást és vezérlést látnak el. **Mondatok** - a feladatok szövegeinek tárolása, **Kérésesek** - a feladatok kérdéseinek tárolása, **Igeidők** - a feladattípusok táblázatának tárolása.

**Entity osztályok:** Adattárolásért felelős osztály. **ABKezelő** - az alapvető CRUD (Create, Read, Update, Delete) feladatok ellátására létrehozott osztály.



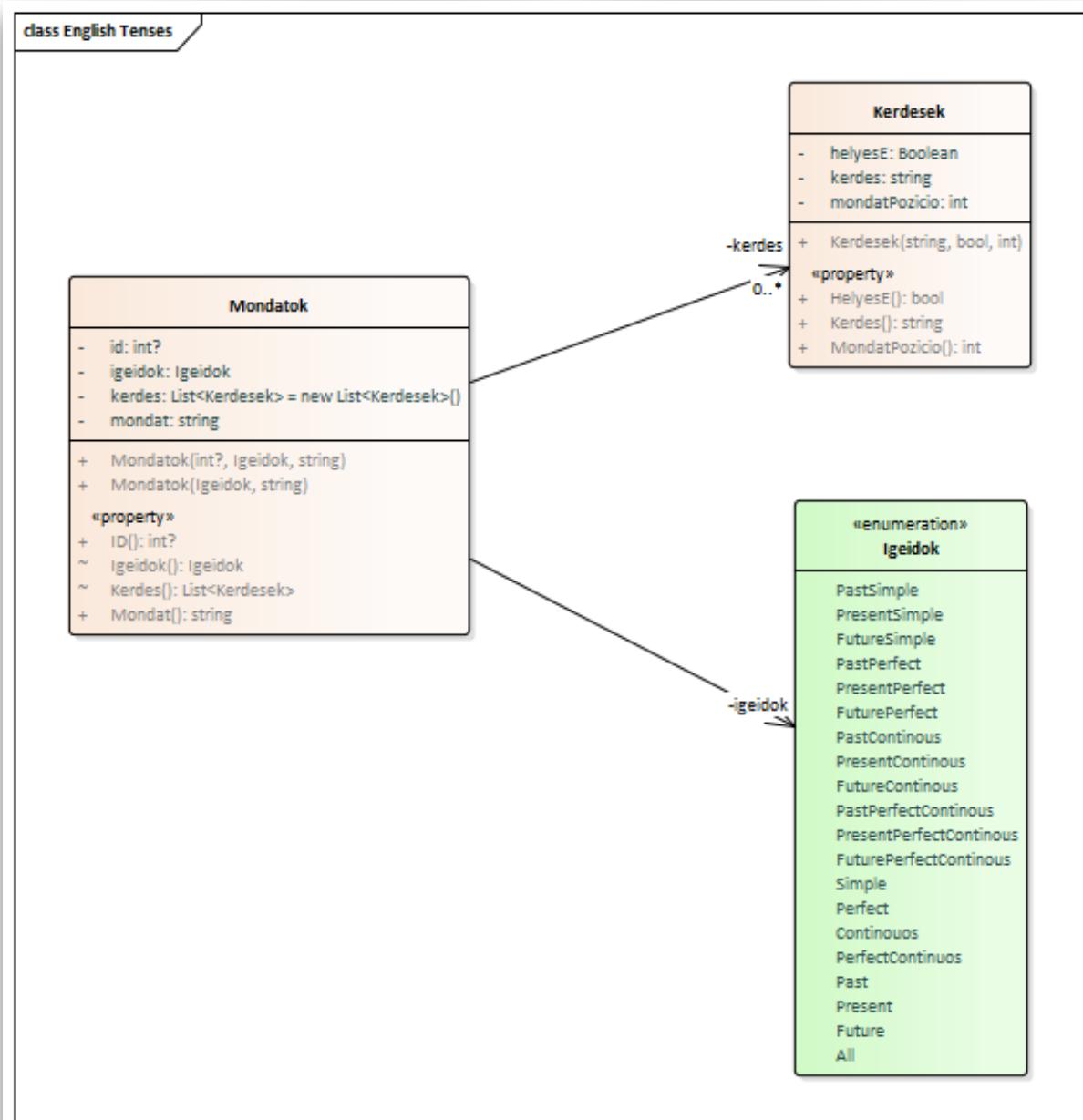
## 4.6. Normal osztálydiagram

Az objektumorientált modell bemutatása osztály diagrammal:

**Igeidők:** enum típus az igeidőket és a minden egyéb gomb nevét a táblázat tartalmazza.

**Kérdezek:** a választásos feladat kérdéseit (string), a válasz helyes vagy nem (bool) illetve a kérdés helye a szavakban mért pozíciója (int) található.

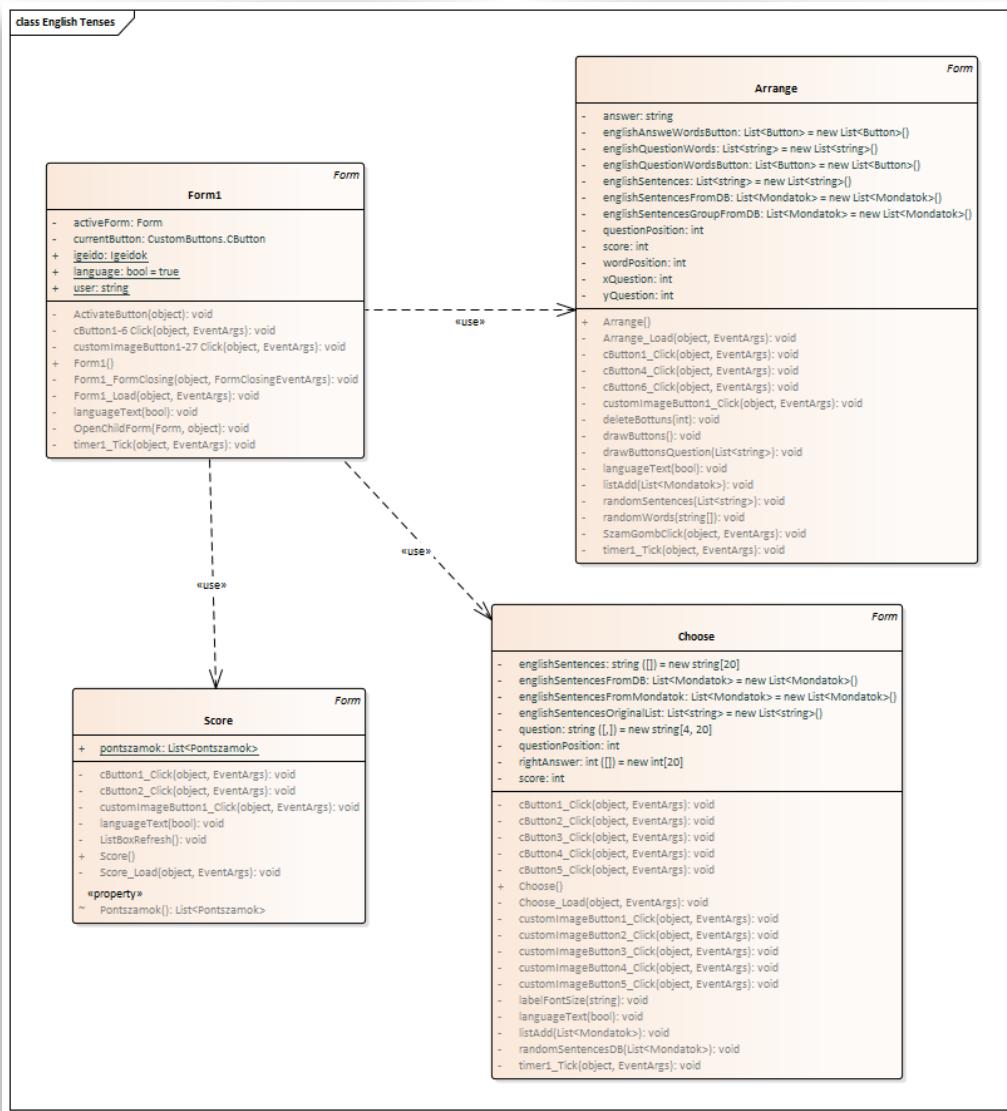
**Mondatok:** az Igeidők-et (enum), és a kérdéseket tartalmazó Lista (sering) és mondatokat (string) tartalmazza. A Mondatok osztálynak két konstruktora van a programban való létrehozáshoz illetve az adatbázisban való létrehozáshoz. Mindegyik osztály kivételekkel van ellátva ha hibás adatok kerülnek bele kivételeket dob így megvédi saját magát a hibás működéstől.



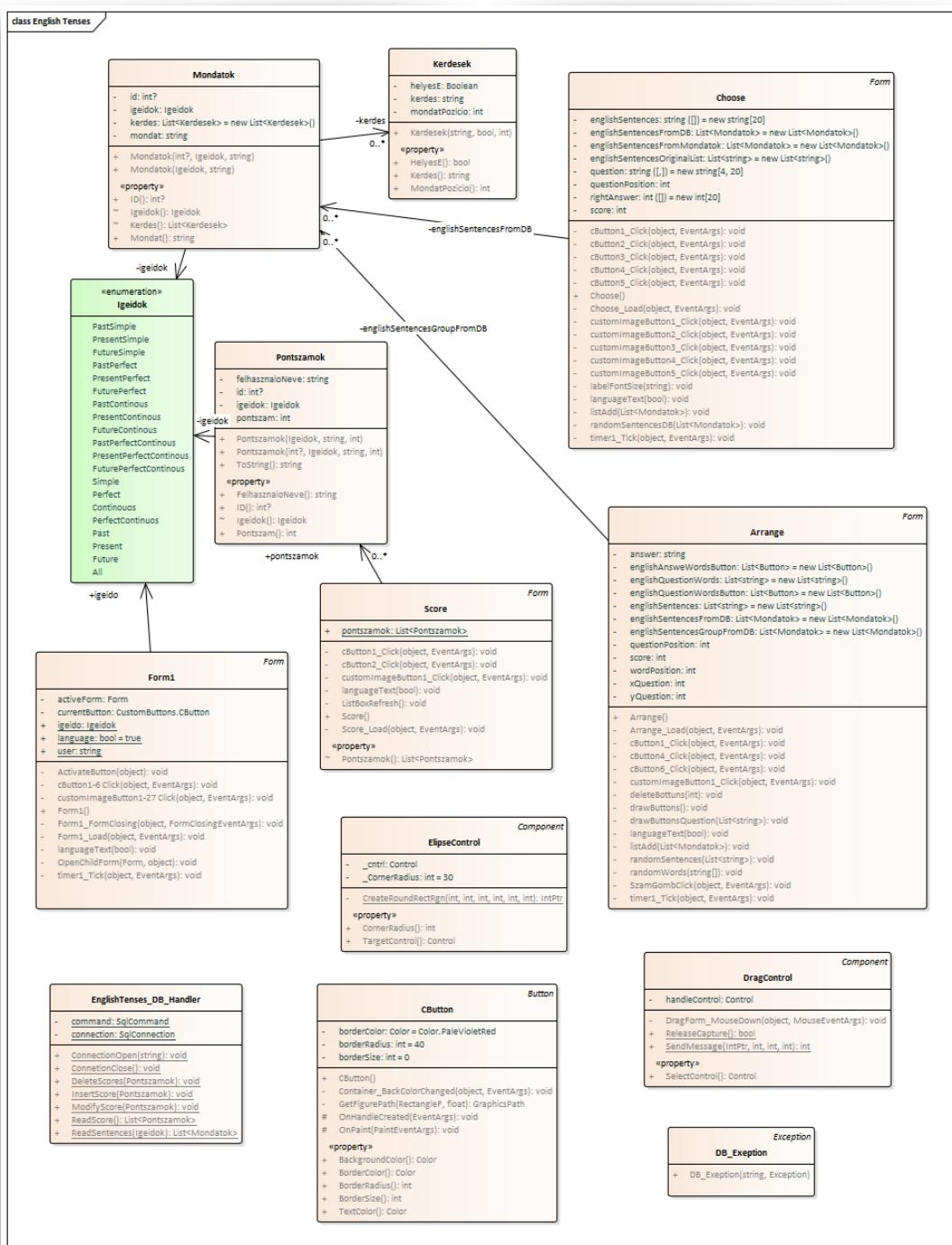
## A formok kapcsolatának bemutatása osztály diagrammal:

**Form1:** ez az alap megjelenítőablak. Ezen vannak rajta a feladattípus, a pontszámok, a főképernyő, a felhasználó nevének megadása. Mindegyik egy panelra van rátárolva, ezek egymás felett helyezkednek el. Az előtérbe helyezésüket az éppen aktuális események határozzák meg. A **Score**, **Arrange**, **Choose**, formok ugyanabba a panelba vannak betöltve. így tudtam azt megoldani, hogy ha esetleg később bővíteni szeretném a feladatokat, akkor csak egy újabb formot keljen megírnom és beillesztenem ebbe a panelba. Azért használtam paneleket, mert ezeknek a hátterét átlátszóra tudtam állítani, ami más egyéb komponenseknél nem sikerült.

**Arrange, Choose:** a feladatok form-jai ezek listákat tartalmaznak az adatbázisból beolvasott adatok tárolására illetve rendezésére. A rendezéses feladat a gombok elhelyezéséhez **xQuestion**, **yQuestion** és a pontok tárolásához a **score** változókat is tartalmaz. A választásos feladatban pedig van egy mondatlista, amiben egy négyelemű másik lista van. Ez tartalmazza a kérdéseket.



### A teljes osztálydiagram:

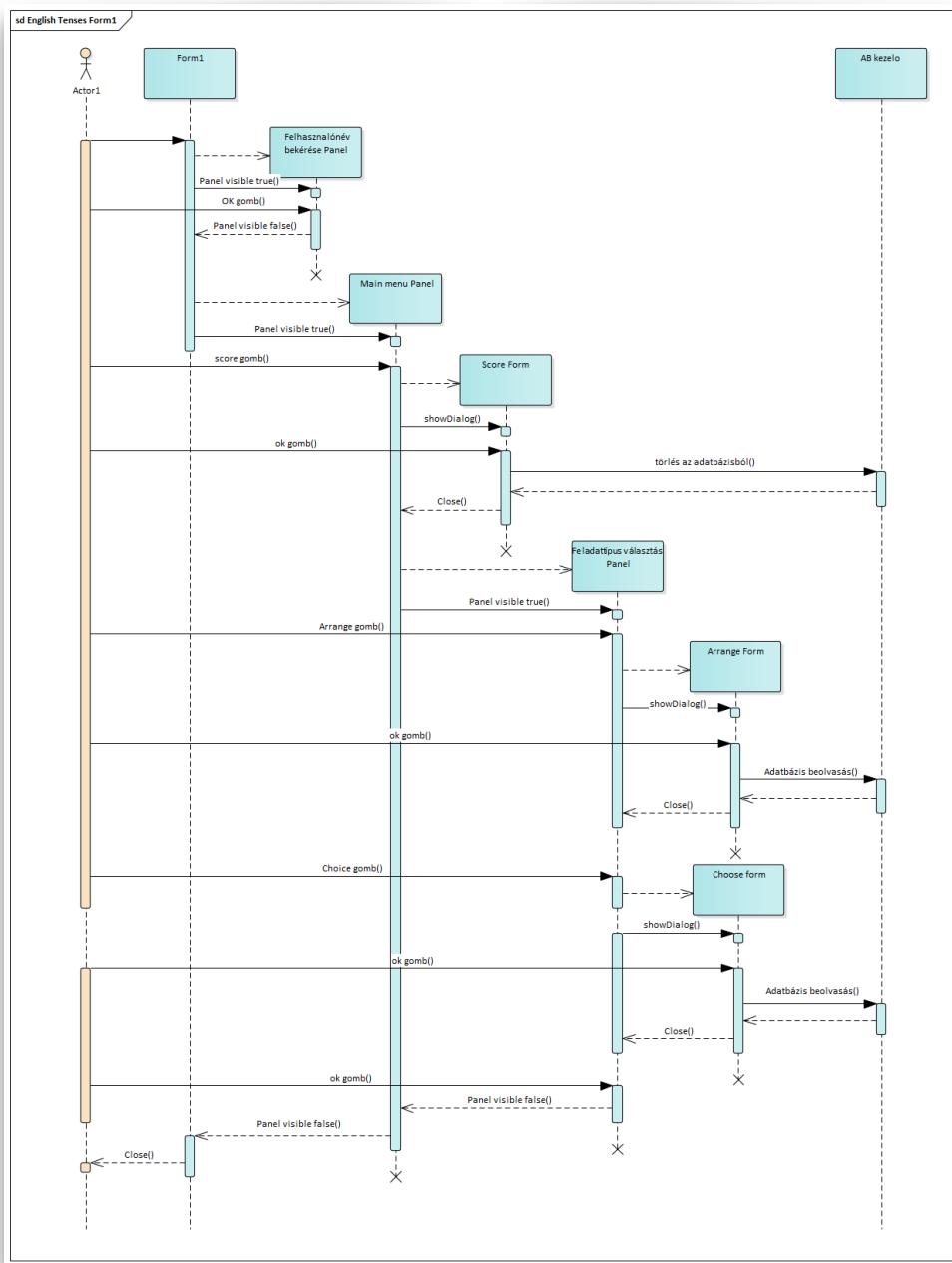


## *Az önálló osztályok:*

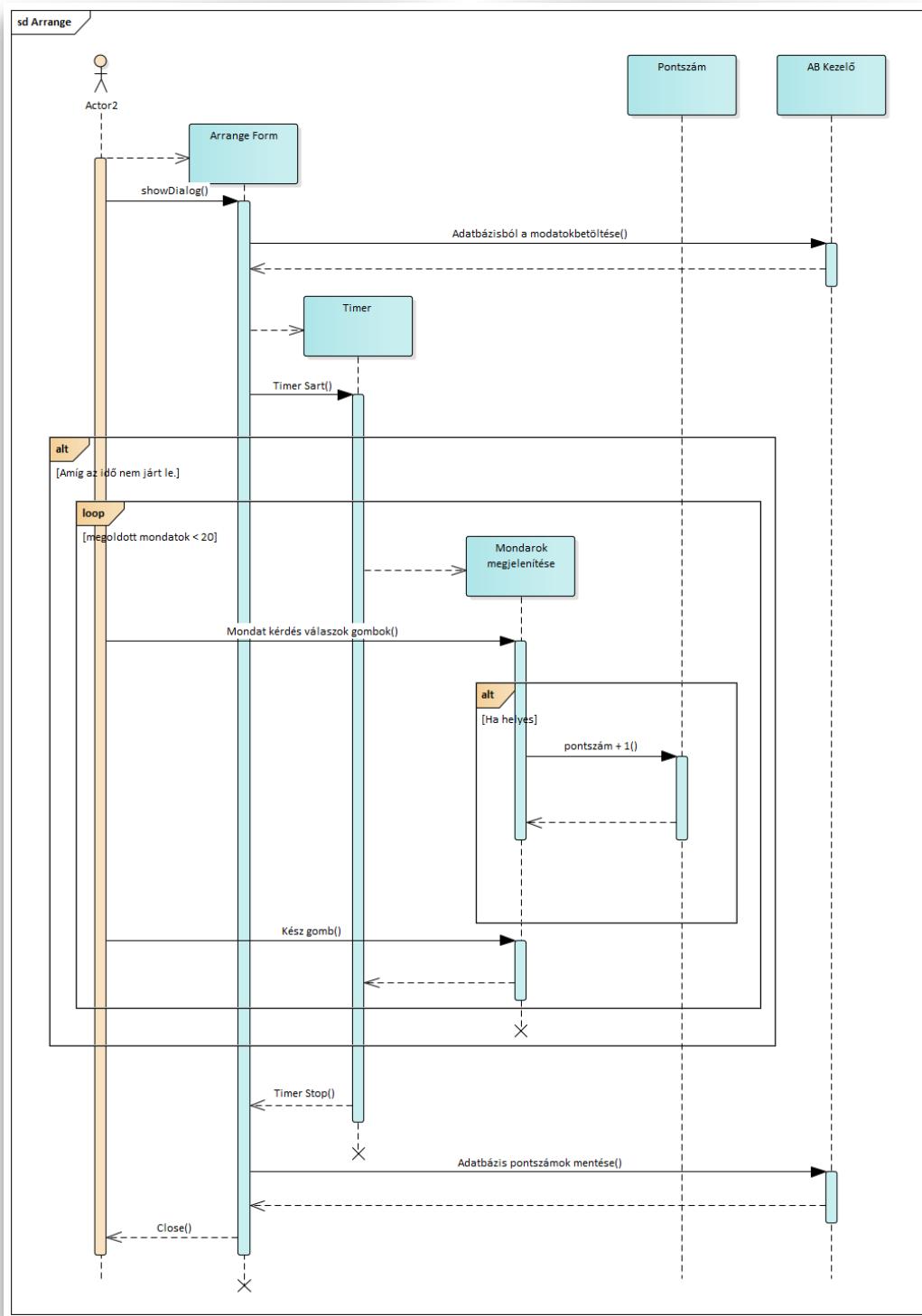
**EnglishTeneses\_DB\_Handler:** osztály az adatbázis CRUD feladatait és azoknak a hibakezelését végzi el, irányítja, **ElipseControl, CButton:** az ElipseControl osztály a formok megjelenítéséhez, pontosabban a négyzetes sarkok lekerekítésehez szükséges komponens, a CButton osztály szintén csak itt a gombokhoz használatos komponens, **DragControl:** osztály a formok egérrel történő mozgatásához szükséges komponens, **DB\_Exeption:** osztály az adatbázis hibakezeléséhez és üzeneteihez szükséges.

## 4.7. Sequence diagram

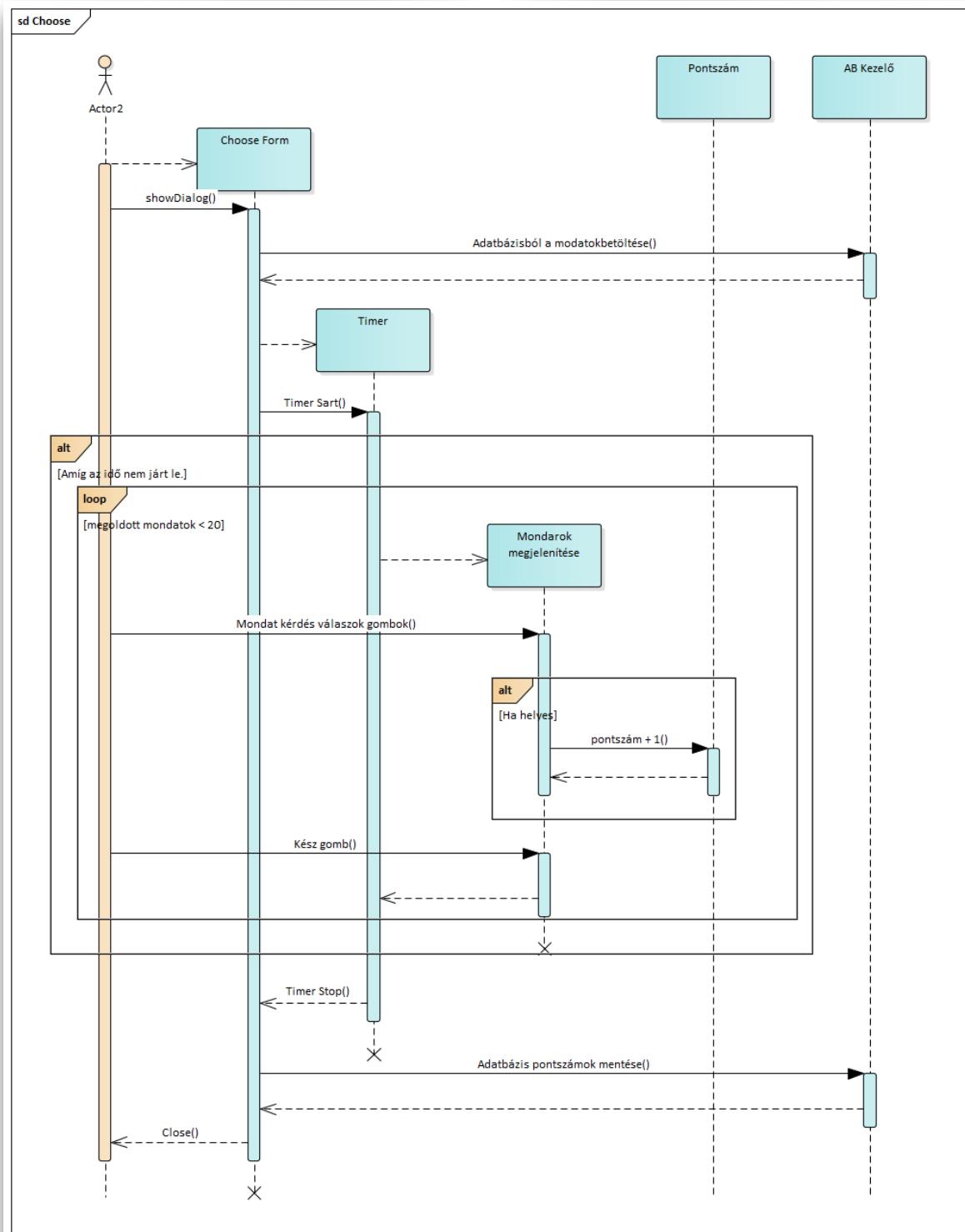
A szekvencia diagram egy viselkedést bemutató diagram, amelynek célja vizualizálni az egyes objektumok kapcsolatait azserint, hogy milyen interakciók szerint jönnek létre a programban. A diagram az osztályok kapcsolatait nem mutatja be. A diagram függőlegesen az időt, vízszintesen a kommunikáló objektumok életvonalait ábrázolja. Az első szekvencia diagram a teljes program működését mutatja meg. Itt jól látszódik az **Actor** - a felhasználó aki kapcsolatba lép mindegyik megjelenő form-al és hatására minden esetben történik valamilyen akció az adatbázisban illetve a program által tárolt adatokban. A **Form1**-en panelek vannak egymáson elhelyezve amelyek az aktiváláskor be- és bezárásukor kikapcsolódnak, illetve ha van még feladat akkor meghívnak egy formot és azok bezáródása után kapcsolódnak ki.



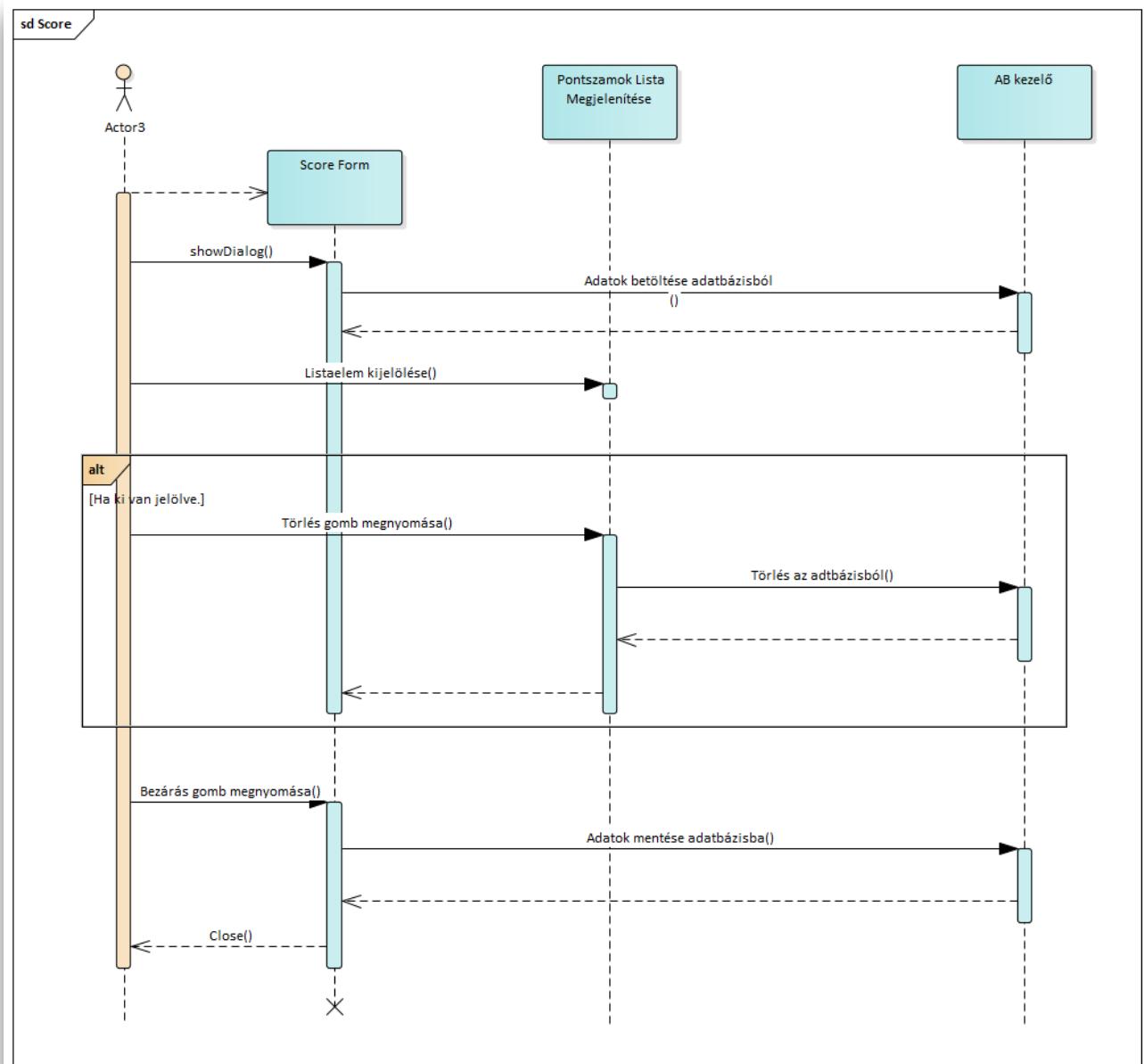
**Az Arrange form szekvencia diagramja** azt mutatja be hogyan működik a rendezős feladat Fromja. A megjelenés után betölti az adatbázisból a mondatokat, utána elindul egy **Timer** ami a feladat megoldásához szükséges időt fogja biztosítani, majd elindul egy ciklus ami a mondatokat és a kérdéseket jeleníti meg. Ha a felhasználó összerakta a mondatot és továbblép akkor ellenőrzi a helyességét. Ha elérte a 20. mondatot, kilép a ciklusból, leállítja a **Timer-t**. Ezután elmenti a pontszámokat és bezárja az Arrange form-ot.



A Choose form szekvenciadiagramja azt mutatja be, hogyan működik a választásos feladat From-ja. Hasonló a rendezéses feladathoz, csak itt a felhasználói beavatkozás a ciklusban egy megadott mondatban kitakart szavakat kell helyes megoldással helyettesíteni. Négy lehetséges kérdésből a helyesre kilkkelve kell eldönteni. Itt az elején az adatbázisból betöltődnek a mondatok és a kérdések, majd a form bezáródásakor elmenti az elért pontszámokat.



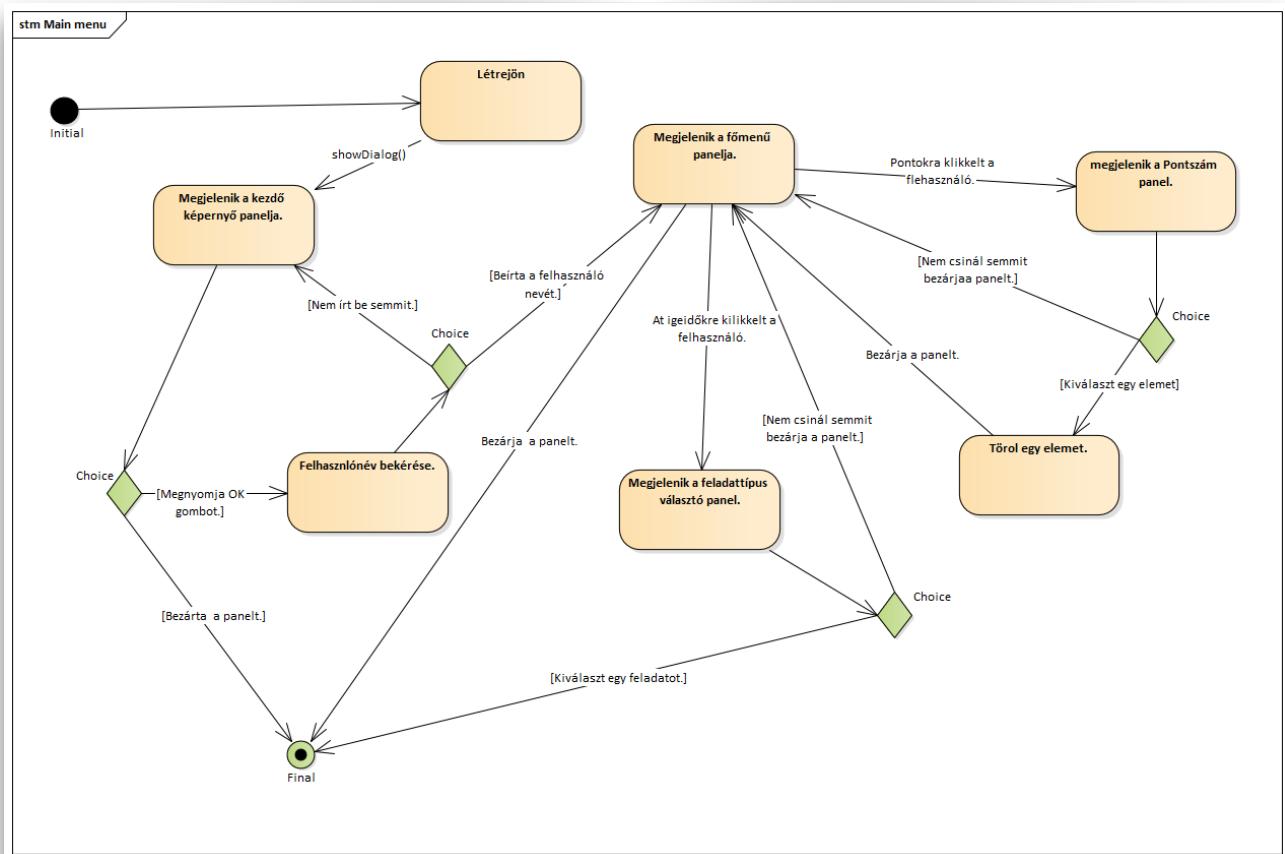
**A Score form szekvenciadiagramja** azt mutatja be, hogyan működik a pontszámlista From-ja. A Score form megjelenése után betölti az adatbázisból a pontszámokat. Itt két lehetőségünk van: ha megnéztük bezárjuk, illetve ha kijelölünk egy listaelemet és töröljük majd utána zártuk be a formot. Mielőtt a form bezáródna, elmenti az adatbázisba a pontszámlista adatait, illetve ha módosítás volt akkor azt is azonnal elmenti.



## 4.8. State Machine diagram

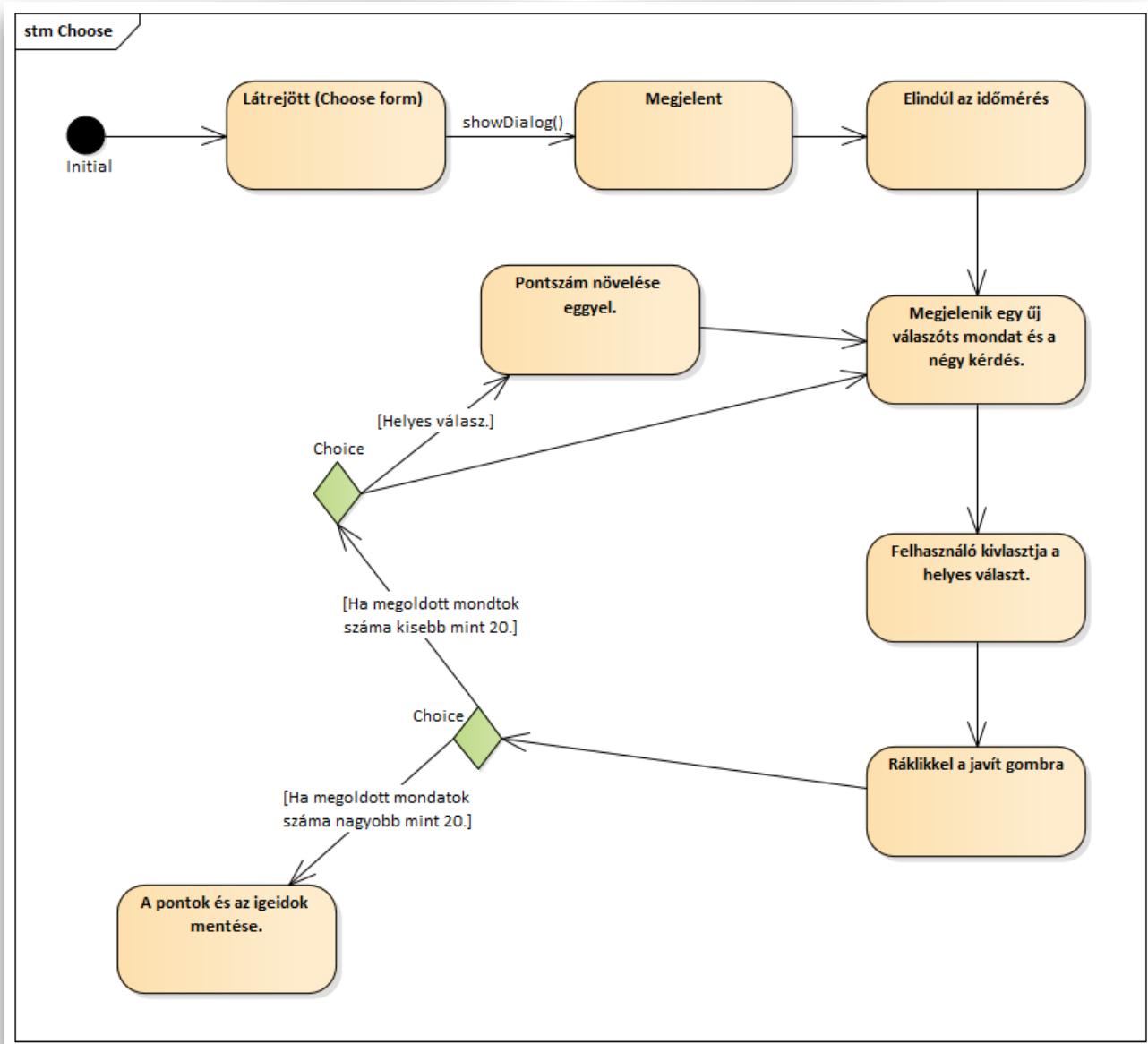
**Form1** diagramja:

A **Form1** form létrejötte után megjelenik a kezdőképernyő. Ha megjelent az **OK** gomb, ezzel lehet elindítani a következő panel megjelenítését. A felhasználópanelbe be kell írni nevet, ha üres marad akkor hibajelzést kapunk. minden panelt be lehet zárni és ilyenkor az előző panel kerül elő. Ha sikeresen beírtuk a felhasználónevet, akkor az **OK** gomba kiloppelünk, a főmenü panel jelenik meg. Innen kétfelé lehet továbbjutni a pontszámokra. Itt a listából lehetséges törölni, illetve ha megnézte felhasználó akár be is lehet zárni a formot. A feladatválasztó panel a táblázat elemeire való klikkelésre jelenik meg, ha kiválasztottunk a **Form1** bezáródik és elindulnak, illetve megjelennek a feladatokhoz tartozó formok.



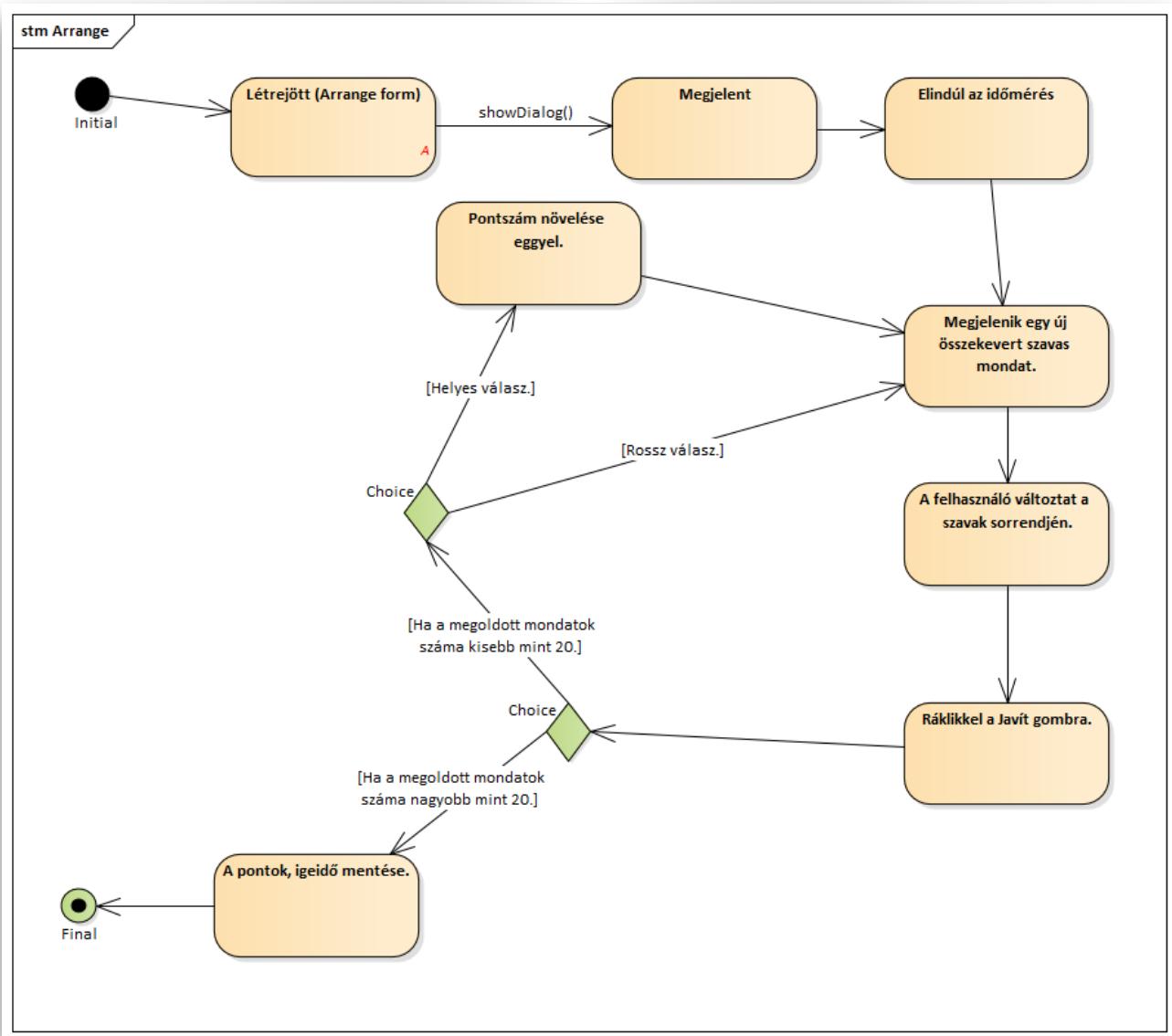
## A Choose form diagramja:

Ha létrejött a **Choose** form és megjelent elindul az időmérés. Egy **progressBar** mutatja pontosan mennyi van még hátra a feladatból. Megjelenik a választott mondat és alatta a négy választható és beilleszthető kérdés. Ha felhasználó kiválasztja a helyes választ és ráklikkel a **Javít** gombra, akkor - ha még nem elérte el a huszadik kérdést - és helyes a válasz, akkor megnöveli a pontszámot, ha nem helyes a válasz akkor újabb mondat töltődik be a kérdésekkel együtt. Ha elérte a huszadik kérdést akkor pedig bezáródik a Choose form.

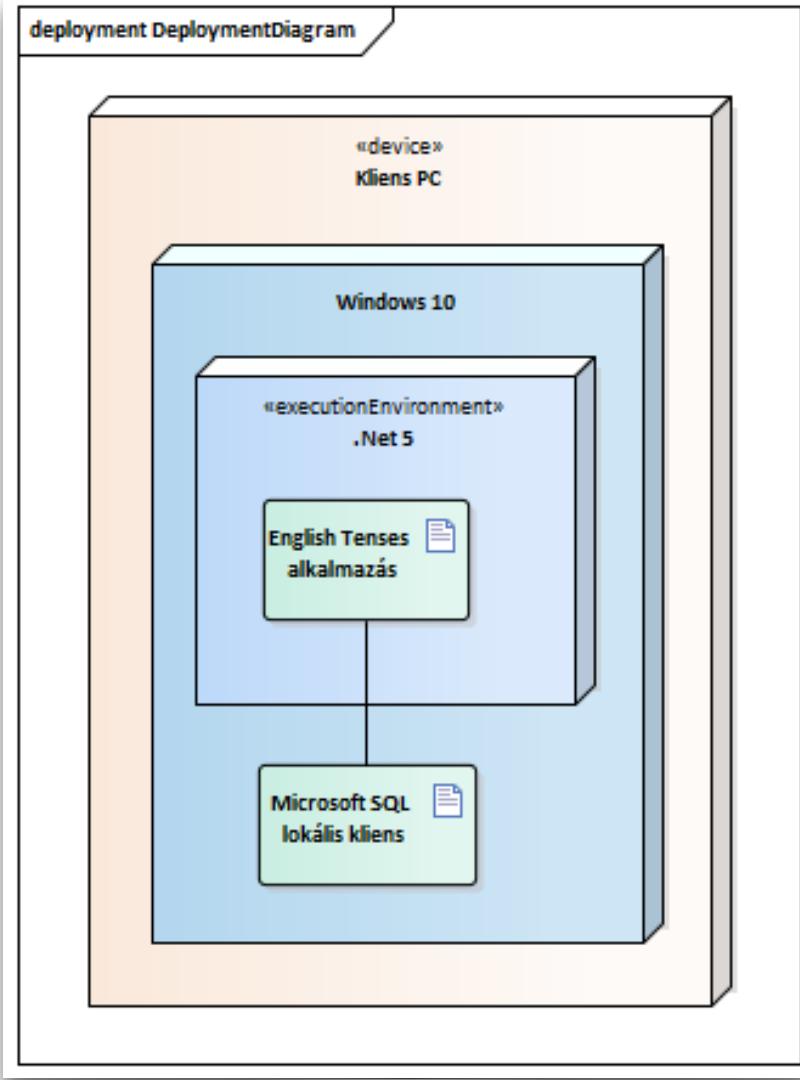


## Az Arrange form diagramja:

Ha létrejött az **Arrange** form és megjelent, elindul az időmérés. Egy **progressBar** mutatja pontosan mennyi van még hátra a feladatból. Megjelenik egy mondat az összekevert szavakkal. Ha megnyomták a Javít gombot vagy megjelenik egy újabb mondat vagy bezárjuk a form-ot. Mielőtt megjelenne egy új mondat, akkor ellenőrzi, hogy helyes-e a megoldás. Ha helyes, akkor a pontszámhoz hozzáad egyet, utána visszatér a mondatmegjelenítéshez. Ez egészen addig tart, amíg el nem érjük a 20. mondatot, itt kiléünk és a mentés után bezárjuk a form-ot.



## 4.9. Deployment Diagram



A szoftver lokális adatbázissal van kapcsolatban, ami azt jelenti, hogy a programból csak egy fut, csak az fér hozzá, és más külső adatbázissal nem kommunikál. A .NET keretrendszer a Windows 10 része és ez az egyik feltétele a szoftver működésének is. Szoftverének nincs különösebb erőforrásigénye, 26MB tárhelyet igényel a jelenlegi adatbázissal. minden olyan korábbi Windows-al elindítható amelyik tudja futtatni a .NET környezet és az MSSQL adatbázis kezelőt.

## **5. ÖSSZEGZÉS**

---

### **5.1. Megvalósult project**

A programom kidolgozásakor elvárás volt a használhatóság és a funkcionalitás, egy alapszinten használható nyelvgyakorló eszköz létrehozása. A program megfelel ezeknek a követelményeknek. Véletlenszerűen kiválasztott mondatokból áll, amit egy adatbázisból vesz ki. Képes nyelvi teszteket készíteni és azokat kikérdezni, ellenőrizni, illetve nyilvántartani az eredményeket. Magyar és angol nyelven van lehetőség használni a programot.

### **5.2. Továbbfejlesztési lehetőségek**

A szoftver későbbi fejlesztési lehetősége reményeim sok irányba fejlődhet még. További többféle feladattípushoz is bele lehet rakni, illetve szótanulási lehetőségeket is. Jelenleg a szoftver lokális adatbázist használ, amit esetleg egy külső szerverre lehetne tenni, valahol a webhálózaton. Ilyenformán az induláskor le tudná ellenőrizni az adatbázis módosításokat. Így állandóan változó szókészettel, vagy mondatokkal lehetne használni.

Az ilyen típusú programokkal az a gondom, ha sokat használom, egy idő után megtanulom a válaszokat és innen már nem érdemi gyakorlás lesz belőle. Külső változtatható adatbázissal ezt meg lehetne akadályozni. A program üzemeltetője akár egy weboldalon keresztül bővíthetné vagy módosítaná a meglévő adatbázist.

Lehetne belőle egy weboldalt készíteni, ami hasonlóan működne mint a program és itt is ki lehetne bővíteni hasonló megoldásokkal. Mivel a program C# működtetéséhez szükséges alap már megvan, ASP.NET segítségével ez meg is oldható.

Ha valaki az összes feladatot és az összes feladattípushoz helyesen kitölti, akkor lehetne egy pdf-ben generált bizonyítvány-szerű gratulációt készíteni a felhasználó nevével és eredményével feltüntetve, ami motivációt adhatna a használóknak.

A későbbiekbén lehetne nehézségi fokokat is beiktatni a fejlesztésbe, amivel játék- és verseny-helyzet-érzést is kaphat a felhasználó.

## **6. IRODALOMJEGYZÉK**

---

[stackwoverflow.com](#)

[geeksforgeeks.org](#)

Microsoft - [docs.microsoft.com/en-us/dotnet/csharp/](https://docs.microsoft.com/en-us/dotnet/csharp/)

[YouTube.com](#) — RJ Code Advance EN (speciális grafikus elemek és komponensek létrehozása)