|  |  |
| --- | --- |
|  | Eötvös Loránd Tudományegyetem  Informatikai kar  Média- és Oktatásinformatikai tanszék |

**Egyedi alkalmazás fejlesztése kvízjátékokhoz**

|  |  |
| --- | --- |
| *Témavezető:*  Dr. Illés Zoltán  habilitált egyetemi docens | *Szerző:*  Ömböli Csaba  programtervező informatikus BSc |

*Budapest, 2022*

Tartalomjegyzék

[1. Témabejelentő 4](#_Toc101375273)

[2. Bevezetés 5](#_Toc101375274)

[2.1. A dolgozat tartalma 5](#_Toc101375275)

[2.2. Mindent vagy semmit 5](#_Toc101375276)

[2.2.1. Az eredeti vetélkedő 5](#_Toc101375277)

[2.2.2. Saját szabályok 6](#_Toc101375278)

[3. Felhasználói dokumentáció 8](#_Toc101375279)

[3.1. Rendszerkövetelmények 8](#_Toc101375280)

[3.2. Telepítés 8](#_Toc101375281)

[3.3. A program helyes használata 8](#_Toc101375282)

[4. Fejlesztői dokumentáció 8](#_Toc101375283)

[4.1. Főbb tervezési irányvonalak 8](#_Toc101375284)

[4.2. Architekturális döntések 10](#_Toc101375285)

[4.3. Felhasznált programkönyvtárak, függőségek 12](#_Toc101375286)

[4.4. Indulás, aktiváció 12](#_Toc101375287)

[4.5. Navigáció 12](#_Toc101375288)

[4.5.1. Shell 12](#_Toc101375289)

[4.5.2. PageService 13](#_Toc101375290)

[4.5.3. NavigationViewService 13](#_Toc101375291)

[4.5.4. NavigationService 13](#_Toc101375292)

[4.6. Játék indítása 14](#_Toc101375293)

[4.7. Adatok bevitele és szerkesztése 14](#_Toc101375294)

[4.8. Az Adatelérési réteg 14](#_Toc101375295)

[4.9. Tesztelési terv 14](#_Toc101375296)

[4.9.1. Unit-tesztek 14](#_Toc101375297)

[4.9.2. Teljesítménytesztek 16](#_Toc101375298)

[4.9.3. End to end tesztek 16](#_Toc101375299)

[5. Összegzés 22](#_Toc101375300)

[6. További fejlesztési lehetőségek 22](#_Toc101375301)

[7. Hivatkozások 23](#_Toc101375302)

# Témabejelentő



# Bevezetés

## A dolgozat tartalma

A szakdolgozat keretében elkészített program alapjaiban a témabejelentőhöz illeszkedik, de a tervezési, és implementálási fázis során felmerülő új szempontok, körülmények és nehézségek miatt helyenként minimálisan eltérhet attól.

Az elkészült dolgozat 2 önálló részre bontható:

**Indító-alkalmazás**

Tartalmaz egy olyan indító-alkalmazást, amely keretet ad más programok rendszerezéséhez, és lehetővé teszi, a felhasználónak, hogy amennyiben az eszközén több játékkal rendelkezik, egy helyről kiválaszthassa, hogy melyiket szeretné elindítani. A program egyszeri konfigurálásával egy helyről, gyorsan indíthat el kvízjátékokat, kiválasztva a rendelkezésre állók közül a neki leginkább tetszőt.

**Mindent vagy semmit program**

A másik fő komponens, egy olyan alkalmazás, ami segítséget nyújt a Mindent vagy semmit televíziós vetélkedőben megismert játékmenet előkészítésében, zökkenőmentes lebonyolításában, és dokumentálásában. Az alkalmazás belekerült a korábban említett indító-alkalmazásba, de önállóan is használható.

A szakdolgozat két alkalmazást tartalmaz, de mivel mindkettőt ugyanolyan tervezési szempontok mentén, nagyon hasonlóan került megvalósításra, ezért a továbbiakban általánosan mindkettőre vonatkozóan írom a tudnivalókat. Ha valamely szempontnál a két alkalmazás eltér, akkor térek ki külön egyesével rájuk.

## Mindent vagy semmit

### Az eredeti vetélkedő

„A vetélkedő 1997-ben indult az akkor újnak számító TV2 képernyőjén, és minden hétköznap az esti műsorsávban jelentkezett. A játék fődíja egy autó volt, ezen kívül pénz- és tárgynyeremények sokasága talált gazdára minden adásban. Az egész az amerikai Joepardy című kvízműsorra épült. Minden adásban három játékos versenyzett, a nap győztese pedig eldönthette, hogy megtartja-e azt a pénzt, amit nyert és kiszáll, vagy visszatér a következő adásra. Aki egy bizonyos számú játékban részt vett az vihette haza a főnyereményt, az autót.” (Vollai, 2019)

A Vágó István nevével fémjelzett műsorban több játékmódban, általános műveltségi kérdésekre válaszolva juthattak közelebb a nyeremények elnyeréséhez a játékosok.

### Saját szabályok

Középiskolai éveim alatt megismerkedtem egy az eredeti vetélkedő szabályaival szinte teljesen megegyező változattal. Az általam elkészített program ennek a variánsnak a szabályrendszerét követi.

Ahhoz, hogy a játékot ebben a formában lehessen játszani, szükség van egy fizikai nyomógomb rendszerre, aminek segítségével eldönthető, hogy ki adhat választ az adott kérdésre.

**A meglévő rendszer technikai részletei**

A jelenlegi rendszer[[1]](#footnote-1) egy hálózati áramforrásra kötött központi egységből, és az ehhez csatlakoztatott négy darab nyomógombból áll. A központi egység biztosítja a gombok az számára az áramellátást, és kezeli az azok lenyomásából származó input jeleket. Ha egy nyomógombot lenyomunk, a gomb előtt található lámpa világítani kezd, a központi egység pedig (logikai áramkörök segítségével) biztosítja, hogy eztkövetően más gombok lenyomására ne világítson a saját lámpájuk, tehát mindig csak az elsőként lenyomott gombhoz tartozó lámpa világítson. A központi egységen található „nullázó” gomb lenyomásával alaphelyzetbe állíthatjuk a rendszert. (Ha van olyan lámpa, ami épp világít, az kialszik, és rendszer ismét inputra vár.)

**A játék menete**

A játékot az eredeti vetélkedőhöz hasonlóan egy műsorvezető/játékmester vezeti. Neki (általában kinyomtatott papíralapú, vagy Word formátumban) rendelkezésére állnak a játékban felhasználásra kerülő kérdéssorok.

A játékosok számától, és az alkalomtól függően lehet egyéni, vagy csapatos játék. Egyéni játék esetén mindenki egyedül áll ki a gombokhoz és versenyez, míg csapatos játéknál az egész csapat egyszerre használja ugyanazt a nyomógombot válaszadási szándékának jelzéséhez. (Az egyszerűség kedvéért a továbbiakban nem különböztetünk meg egyéni játékost, mivel az felfogható egy egyszemélyes csapatként is.) A csapatokat valamilyen szabály szerint[[2]](#footnote-2) körökre osztjuk. A körökre osztásnak fontossági sorrendben a következő preferenciái szoktak lenni:

1. minden csapat legalább n kört játsszon a döntőt nem számolva, (n > 0, n E N)
2. az egy körben szereplő csapatok száma minél egyenletesebb legyen (pl: 9 csapat, és n = 1 esetén rossz megoldás 2 négyfős és 1 egyfős kör kialakítása. Az optimális megoldás 3 háromfős kör.)
3. minél kevesebbszer játsszon egymás ellen két csapat

A körök kialakítása után a csapatok elfoglalják a helyüket egy-egy nyomógomb mögött, és a játék kezdetét veszi.

Egy kör két szakaszból áll. A csapatok feladata mindkét szakaszban az, hogy minél több pontot szerezzenek, a kérdésekre adott helyes válaszaikkal, fontos azonban, hogy a helytelen válaszokért a megválaszolt kérdés pont értékével megegyező levonás jár. (A csapatok akár negatív összpontszámot is elérhetnek.) Az első szakaszban ún. tematikus kérdéssorok kérdéseire lehet válaszolni. Egy kérdéssor 5 témából áll. Minden téma 6 egyre nehezebb kérdést tartalmaz, sorrendben a következő pont értékben: 1000, 2000, 3000, 4000, 5000, 8000. Mindig az előző kérdésre jó választ adó csapat kérheti a következő kérdést, ha ilyen nincs, akkor a legtöbb ponttal rendelkező csapat kérhet. Pont egyenlőség esetén a játékmester dönti el, hogy ki kérhet. A kiválasztott kérdést a játékmester elmondja, a csapatok pedig nyomógombjaik lenyomásával jelezhetik válaszadási szándékukat. (Bármelyik csapat nyomhat, nem csak az amelyik a kérdést kérte.) Mindig az a csapat adhat választ, amelyiknek világít a lámpája.

A második szakaszban „villámkérdésekre” lehet választ adni. Itt a játékmester sorban egymás után tesz fel véletlenszerű témákból kérdéseket. Ebben a szakaszban az összes kérdés 3000 pontot ér. Mivel itt nincsenek témák, amikből választani lehetne, ezért egy válsz elhangzása után a játékmester automatikusan folytatja a következő kérdéssel. Fontos, hogy míg a tematikus szakaszban a játékvezető megengedőbb, azzal kapcsolatban, hogy mennyi idő telik el a gomb lenyomása, és a válasz elhangzása között, addig a villámkérdéseknél a válasznak azonnal kell jönnie, különben a választ helytelennek ítéli meg.

A tematikus és a villámkérdéses szakasz lejátszására is korlátozott idő áll rendelkezésre. A csapatok által az egyes körökben megszerzett pontszám összeadódik, ez képzi majd az összpontszámot.

Az egyes körök végeztével kezdődik a sorban utánuk következő, egészen addig, amíg az összes kört le ne játszották a résztvevők. Ekkor következik a döntő kör, melynek szabályai megegyezni a korábbiakkal. A döntő résztvevői az első 4 legmagasabb összesített pontszámmal rendelkező csapat, akik a döntőben mind 0 pontról indulnak. A döntő győztese nyeri a játékot.

**A jelenleg használt segédprogram**

A játék során használt kérdéssorok megjelenítéséhez, a csapatok pontszámainak kijelzéséhez, és az idő méréséhez jelenleg egy általam ismeretlen személy által, a 2000-es években készített program van használatban. A programot egy ember kezeli, aki egyrészt figyeli a játékosok által kért kérdést, és mindig azt jeleníti meg a képernyőn, másrészt figyeli hogy a játékmester elfogadta-e a választ, és ez alapján adja meg/vonja le a pontokat a csapatoktól.

Az általam megvalósított Mindent vagy semmit program ennek a jól bevált működését veszi alapul, és ezen program hibáiból tanulva kísérel meg az elődjénél jobb felhasználói élményt nyújtani (A részleteket lásd a későbbi fejezetekben.)

# Felhasználói dokumentáció

## Rendszerkövetelmények

## Telepítés

## A program helyes használata

# Fejlesztői dokumentáció

## Főbb tervezési irányvonalak

Középiskolás koromtól kezdve, amikor barátaimmal még heti szinten aktívan használtam a kvízes rendszert (kiváltképp a programot), egészen mostanáig megvan bennem a szándék, hogy legyen egy olyan alkalmazás, ami jobban megfelel az eredeti program készítése óta eltelt évek támasztotta igényeknek. Nekem is volt sok ötletem ezzel kapcsolatban, de hogy a tényleges felhasználói igényeket felmérve tudjak nekiállni a tervezésnek, egy ehhez kapcsolódó kérdőív szétküldésével gyűjtöttem visszajelzéseket. Nem érkezett be annyi válasz, amennyire számítottam, de úgy vélem a felmérés, még így is reprezentatív.

Mivel a készítendő alkalmazásnál a funkciót tartottam elsődleges fontosságúnak, ezért az 1. és 2. ábrán látható visszajelzések fontos útmutatásként szolgáltak, hogy milyen mélységben kell „hozzányúlnom”, vagyis módosítanom, esetleg kivennem, az eddig használt funkciókhoz.



1. ábra A régi program funkcionalitására irányuló kérdésre kapott válaszok aránya

A képen szöveg látható

Automatikusan generált leírás

2. ábra A régi program esetleges haszontalan funkcióiról érdeklődő kérdés, és az erre kapott válaszok.

A meglévő funkciók átdolgozására, és újak hozzáadására igen sok javaslat érkezett, amelyek együttes megvalósítása (néhány általam kitalált funkcióval együtt) túlmutat a szakdolgozat keretein, ám többségük a jövőben vélhetően megvalósításra kerül. (Lásd: További fejlesztési lehetőségek c. fejezet.)

A funkció után a megjelenés volt számomra a következő prioritás. Véleményem szerint lehet egy alkalmazás bármennyire fejlett, és funkcionálisan tökéletes, ha nem frissítik a megjelenését, az évenkét változó UI/UX igények és trendek fokozatosan elavulttá teszik a felhasználók szemében. Bár a meglévő program elég egyedi a maga műfajában, de korábbi állításomat a 3. ábrán látható eredmény igazolni látszik.



3. ábra A régi program kinézetét érintő kérdés

Míg az előző két szempont talán felhasználóként a leglényegesebb, addig fejlesztőként a már a témabejelentőben is szereplő bővíthetőség kritikusan fontos. Mivel az alkalmazást a jövőben több területen tervezem továbbfejleszteni, ezért értékesebbnek tartok egy átláthatóan, és koherensen megírtat, egy extra funkciókkal telerakott, és/vagy legelkápráztatóbb UI megoldásokat használó, de kevésbé fejlesztő barát alkalmazásnál.

Összegezve: A fejlesztés során a prioritások sorrendben a következőként alakulnak:

1. Bővíthetőség, átláthatóság
2. Funkcionalitás
3. Kinézet

## Architekturális döntések

Az alkalmazás egy asztali alkalmazás. 2022-ben valóban szokatlan lehet egy dedikáltan asztali alkalmazást készíteni, ám ebben az esetben vannak olyan körülmények, amik ezt indokolttá teszik. A közösségünkben meglévő régi program (aminek az utódjaként ez készült) igénybevételeire, jelentős hányadban olyan környezetben kerül sor, ahol nincs, vagy csak akadozó internetkapcsolat áll rendelkezésre. Ezen okból kizárható a kizárólag online működtetett alkalmazás. A fennmaradó offline platformok közül a PC-s megvalósításon kívül minden más irreleváns lett volna.

A következő fontos döntés, a környezet kiválasztása volt. Mivel Windowson kívül más operációs rendszerre megvalósítani nem lett volna értelme, ezért kézenfekvő választás volt a .NET keretrendszer egy Windows specifikus szegmense.

A választott környezet eleinte az UWP[[3]](#footnote-3) volt, de idő közben a Microsoft kiadta az új MAUI[[4]](#footnote-4) rendszerbe illeszkedő WinUI 3[[5]](#footnote-5) (és ezzel együtt a Windows App SDK[[6]](#footnote-6)) első stabil verzióját. Ez az újszerű platform a korábban lefektetett szempontok közül a funkcionális helyességet nem befolyásolja, de a másik kettőben komoly előrelépés figelhető meg az UWP-hez képest. Általában elmondható, hogy egy újabb rendszer lévén jobban teljesíti az egyszerű bővíthetőséggel és fenntarthatósággal szemben támasztott követelményeinket, ám ezt csak fokozza, hogy a Microsoft megszűntette az UWP platform támogatottságát, a Windows App SDK javára, valamint hogy a Windows 11 operációs rendszer későbbi verzióiban várhatóan nem fognak futni az UWP-ra írt alkalmazások. A WinUI 3 tovább bővíti majd az előd amúgysem szegényes eszköztárát a felhasználói felület megvalósításához.

Ezek alapján az alkalmazás a következők szerint került megvalósításra:

* Windows App SDK-t és WinUI 3-at használ
* Szerkezete a platformhoz illő MVVM architektúrát követ
* A model és view-model részek C# nyelven, a view pedig XAML leíró nyelven kerültek megvalósításra
* Az alkalmazás .NET 5 és C# 9 szabványokat támogat

Az alkalmazás a Visual Studio 2019 fejlesztői környezeteben készült, és a további fejlesztői tevékenység zökkenőmentes folytatásához is ez javasolt. Régebbi és újabb verziók használata az esetlegesen fellépő átmeneti problémák miatt lassíthatja a fejlesztési folyamatot.

A projekt létrehozásakor a Visual Studioba kiegészítőként telepíthető Windows Template Studio-t vettem igénybe, amely néhány alapvető funkcióhoz szükséges sablonnal látta el az újonnan létrehozott alkalmazást. (Mivel a projektet eredetileg UWP projektként hoztam létre és csak később döntöttem a migrálásáról, ezért az akkor készült sablonok sem feltétlenül egyeznek meg egy újonnan generálttal.)

## Projektfelépítés

A forrsákód az egyes elemek logikai felépítésének szétválaszthatósága, és a könnyebb áttekinthetőség érdekében 3 projectbe rendeződik:

**AllOrNothing**

Ez az alkalmazás fő projektje, itt történik a legtöbb lényeges dolog. Ebben találhatóak meg a megjelenítésért felelős View-k, a hozzájuk kapcsolódó ViewModel-ek, az alkalmazás tagolt működését lehetővétevő Service-ek, és segítő osztályok.

**AllOrNothing.Repository**

Ez az elkalmazás adatelérési rétege. Itt kerül megvalóósításra a Repository és a UnitOfWork programtervezési minta, illetve itt található az EntityFramework működéséhez szükséges fájlok (migrációs fájlok, data-context fájlok stb…), az adatmodellek kivételével. (lsd.: Adatelérési réteg c. fejezet)

**AllOrNothing.Data**

Ebben a projektben kerültek definiálásra az adatok kezeléséhez és tárolásához kapcsolódó adatmodellek.

Test project

## Felhasznált programkönyvtárak, függőségek

## Indulás, aktiváció

Az alkalmazás belépési pontja a környzet által generált App.g.cs fájlban található, ahol az App osztály példányosításával át is tértünk a fejlesztői kódra.

## Navigáció

A program logikailag elkülöníthető funkciói az alkalmazáson belül külön oldalon kerülnek megjelenítésre. A következő oldalak léteznek jelenleg:

* főmenü
* játék beállítások
* Játék
* eredmények
* kérdéssorok-kezelő
* játékos kezelő
* alkalmazás szintű beállítások
* statisztikák (lsd.: További fejlesztési lehetőségek c. fejezet)

Ezen oldalak megjelenítése, és az oldalak közötti váltás, egy Frame, és egy navigációs szolgáltatás segítségével történik.

### Shell

A megjelenítés alapját a ShellPage és a hozzá tartozó ShellViewModel képzi. A SehellPage a Microsoft.UI.Xaml.Controls névtérben megtalálható Page osztályból származik, és funkcióit xaml kódon keresztül nyeri el. Az oldal tartalmát egyetlen, szintén a fenti névtérben megtalálható NavigationView objektum teszi ki. Ennek a MenuItems gyűjteményébe vehetjük fel azokat az elemeket, amiket a menüben szeretnénk látni, a megjelenítendő oldalt pedig a NavigationView Content property-jének értékül adott Frame objektum fogja a képernyőre renderelni. A későbbiekben ha a program oldalt vált (akár a navigációs menüre történő kattintás hatására, akár más okból), akkor mindig a ShellPage belsejében található Frame által megjelenítendő oldalt változtatjuk.

A navigáció másik két meghatározó része a NavigationService és a NavigationViewService.

### PageService

Az oldalak egyszerű betölthetősége érdekében kulcs-érték párokként összekötjük az egyes oldalakhoz tartozó ViewModel-leket, és az oldalakat. A későbbiek során így már elég lesz csak a betöltendő oldal ViewModel-jére hivatkozni, és ebből tudható lesz, hogy melyik oldalt akarjuk betölteni. Fontos, hogy ezen logika miatt viszont egy az egyhez reláció él az oldalak, és a hozzájuk tartozó ViewModellek között. Az előbb leírt folyamat a PageService osztályban történik.

### NavigationViewService

Ez a szolgáltatás a NavigationView által tartalmazott MenuItem-ek felé húzott absztrakciós réteg, ami segíti azok használatát. Referenciaként megkapja a ShellPage-en található NavigationView objektumot, majd a metódusaival ezt manipulálja. Legfontosabb funkciói, az új elem felvétele a navigációs menübe, valamint az egyes menüpontokra történő kattintás kezelése.

Az új elem ellenőrzött hozzáadásához tartozik, hogy ne lehessen többször ugyanazt a navigációs elemet a menühüz adni, valamint, hogy ne legyen a menüben két olyan elem, ami ugyanarra az oldalra vezet. Ez utóbbi ellenőrzéséhez szükséges a privát mezőként tárolt PageService példány.

A NavHelper osztály segítséget nyújt, hogy be tudjuk állítani az egyes menüelemekhez, hogy melyik oldalra vezessenek. A NavigationViewService osztály feliratkozik a NavigationView ItemInvoked eseményére, mely az egyes menü elemekre történő kattintás hatására kerül elküldésre. Az esemény fogadásakor a privát referenciaként tárolt NavigationService példány NavigateTo metódusának meghívásávlakezdjük meg a tényleges navigációt.

### NavigationService

Az osztály paraméterként kapja meg a ShellPage-en szereplő Frame-et, majd ezen objektum metódusai köré alakít ki biztonságos metódusokat, amik hívása biztosan elvárt eredményt ad. Legfontosabb feladata, a Frame NavigateTo metódusának meghívása, mely a saját azonos nevű metódusában történik meg. Mivel a Frame a céloldal típusát kéri, ezen metódus pedig a céloldal ViewModel-jének nevét kapja meg, ezért szükség van a privát PageService objektumra, hogy lekérjük a ViewModel-hez tartozó típust.

## Játék indítása

## Adatok bevitele és szerkesztése

## Az Adatelérési réteg

## Tesztelési terv

### Unit-tesztek

asd

Beolvasás fájlból:

* Helyes régi
* Helytelen régi
  + kevesebb téma
  + kevesebb kérdés egy témában
  + több téma
  + több kérdés
  + más elválasztó karakter
* Helyes új
  + nincs megadva szerző
  + csak szerző név van
  + csak szerző id van
  + csak intézmény van
  + nincs játékos az id-vel
  + van játékos az id-vel, de nem a megadott néven
  + nincs megadva leírás
  + sehol nincs megadva imeretkör
  + nem mindenhol van megadva ismeretkör
  + olyan ismeretkör van megadva, ami nem szerepel a rendszerben
  + egy témánál többször szerepel ugyanaz az ismeretkör
  + egy kérdéssoron belül többször szerepel ugyanaz az ismeretkör
* helytelen új
  + kevesebb téma
  + kevesebb kérdés egy témában
  + több téma
  + több kérdés
  + más elválasztó karakter
  + érvénytelen id van megadva (nem szám, túl nagy szám stb)

Csapat generálás

* normál eset => mindenki csak egyszer szerepel csapatban összesen
* normál eset => a legnagyobb csapat létszáma maximum M
* kevesebb játékos mint M
* több fős csapat => csapatnév jó-e
* egyfős csapat => csapatnév jó-e

Körök generálása

* minden körben különböző csapatok szerepel
* minden körben max M csapat szerepel
* minden csapat pontosan K körben szerepel
* egy csapat <K alkalommal szerepel bármely másik csapat ellen
* nincs csapat megadva
* kevés csapat van megadva (?)

Adatelérés tesztje

* játékos hozzáadása
* játékos módosítása
* játékos törlése
* kérdéssor hozzáadása
* kérdéssor módosítása
* kérdéssor törlése
* kérdés hozzáadása
* kérdés módosítása
* kérdés törlése
* ismeretkör hozzáadása
* ismeretkör módosítása
* ismeretkör törlése

### Teljesítménytesztek

### End to end tesztek

|  |  |  |
| --- | --- | --- |
| Player adding page test | | |
| id | Test-case | Expected result |
| 1.1 | Open the application | The application opens the main menu page, with 3 menu points. („Új játék”, „Kérdéssorok”, „Játékosok”) |
| 1.2 | Click ont he menu point „Játékosok” | The player editor page opens and the navigation bar indicates the switch. The existing players are listed on the left side, ont he right side is a text: „Válasszon játékost!” |
| 1.3 | Press the „Új játékos” button | The „Azonosító”, „Név”, „Becenév”, and „Intézmény” fields appears, all of the is empty except „Név”. The „Mentés” and the „Játékos törlése” buttons are visible. |
| 1.4 | Fill the fields „Név” : E2E Test1,  „Becenév”: E2E1, „Intézmény”: Testing | The fields are properly filled |
| 1.5 | Press the „Mentés” button. | A pop-up is shown with the „Sikeres mentés!” message. |
| 1.6 | Press the „Ok” button in the pop-up | The player named „E2E Test1” is present in the list, and there is a text on the right side: „Válasszon játékost!”. |
| 1.7 | Press the „Új játékos” button | The „Azonosító”, „Név”, „Becenév”, and „Intézmény” fields appears, all of the is empty except „Név”. The „Mentés” and the „Játékos törlése” buttons are visible. |
| 1.8 | Fill the fields „Név” : E2E Test2,  „Becenév”: E2E2, „Intézmény”: Testing | The fields are properly filled |
| 1.9 | Press the „Mentés” button. | A pop-up is shown with the „Sikeres mentés!” message. |
| 1.10 | Press the „Ok” button in the pop-up | The player named „E2E Test2” is present in the list, and there is a text on the right side: „Válasszon játékost!”. |
| 1.11 | Select the player named „E2E Test2” | The form is filled with the previously given data. The „Mentés” and „Játékos törlése” buttons are visible. |
| 1.12 | Click on the „Játékos törlése” button | A pop-up message appeares: „Biztosan törli a játékost?” |
| 1.13 | Press the „Mégse” button | The pop-up disappeares, and the player named „E2E Test2” is still avaible in the list. |
| 1.14 | Click on the „Játékos törlése” button | A pop-up message appeares: „Biztosan törli a játékost?” |
| 1.15 | Press the „Igen” button. | The player named „E2E Test2” is deleted from the list, and there is a text on the right side: „Válasszon játékost!”. |
| 1.16 | Repeate the previous steps, to have at least 16 players in the list | At least 16 players are avaible in the list |

|  |  |  |
| --- | --- | --- |
| Questionserie page test | | |
| Id | Test-case | Expected result |
| 2.1 | Open the application | The application opens the main menu page, with 3 menu points. („Új játék”, „Kérdéssorok”, „Játékosok”) |
| 2.2 | Click on the menu point „Kérdéssorok” | The questoin serie editor page opens and the navigation bar indicates the switch. The existing series are listed on the left side, ont he right side is a text: „Válasszon kérdéssort!” |
| 2.3 | Press the „Új kérdéssor” button | Ont he top of the form there is a field called „Kérdéssor címe”, with a valuo of „Új kérdéssor”. The form populates itself with 5 topics, all with the following values: Empty fields: „Téma címe”, „Leírás”. The seletion area named „Ismeretkör” is empty. 6 times the following: „Kérdés” (empty), „Válasz” (empty), „Érték” (0) |
| 2.4 | Change the field „Kérdéssor címe” to „E2E TestSerie1” | The field value is changed |
| 2.5 | Press the button „Mentés” | A pop-up is shown to warn you to fill all the rquiered fields. |
| 2.6 | Press the „Ok” button in the pop-up | The pop-up gets closed, and the form has the same values as before |
| 2.7 | Type some non numeric characters in the field „Érték”, and clikck elsewhere (or get rid of the focus on the field) | The field deletes the non-numeric characters, and set it’s value to 0. |
| 2.8 | Type 1000 to field „Érték” | The value is properly set |
| 2.9 | Type some non-numeric characters in the field „Érték”, and clikck elsewhere (or get rid of the focus on the field) | The field deletes the non-numeric characters, and set it’s value to 1000. |
| 2.10 | Type „t” in the field „Ismeretkör” | A list appears, and all values in it contain „t” |
| 2.11 | Click one of them | The competence gets added tot he area below |
| 2.12 | Press the X nex tot he competence | The competence is now removed from the area |
| 2.13 | Fill all the fields with some values excetp the „Ismeretkör” | Each field has a value, except „Ismeretkör” |
| 2.14 | At the first topic add competence „történelem” to the competence area | „történelem” is in the competence area |
| 2.15 | At the second topic add „történelem” to the competece area | „történelem” is in the competence area |
| 2.16 | At the last topic add „földrajz” to the competence area | „földrajz” is in the competence area |
| 2.17 | Press the „Mentés” button | A pop-up appears with the message „Sikeres mentés” |
| 2.18 | Close the pop-up by pressing „Ok” | The serie named „E2E TestSerie1” is visible in the list. Its „Dátum” field has the value of the current date, its „Ismeretkör” field has exactly the following values: „történelem”, „földrajz”. Ont he right side the „Válasszon kérdéssort” message is visible |
| 2.19 | Click on the previously created serie | The form gets populated with all the previously given values |
| 2.20 | Press the button „Kérdéssor törlése” | A pop-up shows with the message „Biztosan törli a kérdéssort?” |
| 2.21 | Press „Mégse” | The pop-up disappears, and the form’s values are unchanged. |
| 2.22 | Press the button „Kérdéssor törlése”, and after that press the button „Igen” | The same pop-up appears as before, and after that, the serie is removed from the list, and the form displays the message: „Válasszon kérdéssort!” |

|  |  |  |
| --- | --- | --- |
| Game test 1 | | |
| id | Test-case | Expected result |
|  | Open the application | The application opens the main menu page, with 3 menu points. („Új játék”, „Kérdéssorok”, „Játékosok”) |
|  | Click on the menu point „Új játék” | The game settings page opens, the navigation bar indicates the page-switch. |
|  | At the searchbox „Játékos keresése” search for the players you created at test-case 1.16. | A list is shown with the previously created players. |
|  | Click all of them one after another | All 16 players gets added to the area below |
|  | In the searchbox search for a player that is added to the area below | The player do not appers in the list |
|  | Click one of the X next to a player | The player is removed from the area |
|  | In the searchbox search for the player you removed in the prev. step | The player is visible in the list |
|  | Fill the field „Alkalom neve” with the value „Test Occasion1” | The vlaue is set properly |
|  | Fill the field „Körök száma” with some non numeric charackters, and click elsewhere | The field resets its value to 1 |
|  | Fill the field „Körök száma” with a nuumber that is less than 1, and click elsewhere | The field resets its value to 1 |
|  | Fill the field „Körök száma” with 3 | The value is set properly |
|  | Fill the field „Maximális csapatlétszám” with some non numeric charackters, and click elsewhere | The field resets its value to 1 |
|  | Fill the field „Maximális csapatlétszám” with a nuumber that is less than 1, and click elsewhere | The field resets its value to 1 |
|  | Fill the field „Maximális csapatlétszám” with 3 | The value is set properly |
|  | Press the „Csapatok sorsolása” button | At the „Csapatok” section many teams appears, all with a maximum of 3 player. |
|  | Change one of the teams team-name field to „TestTeam1” | The team-name is changed |
|  | Check the chekbox „Tematikus” | The checkbox is checked, and the time picker next to it gets enabled. It has a value of 1:00 |
|  | Set 0:00 at the time picker next to „Tematikus” | The value is set to 0:00, and the time picker gets disabled. The checkbox „Tematikus” gets unchecked. |
|  | Check the chekbox „Villám” | The checkbox is checked, and the time picker next to it gets enabled. It has a value of 1:00 |
|  | Set 0:00 at the time picker next to „Villám” | The value is set to 0:00, and the time picker gets disabled. The checkbox „Villám” gets unchecked. |
|  | Press the button „Kilépés” | A pop-up is shown wit the message „Biztosan kilép?” |
|  | Press the „Mégse” button | The pop-up disappears and everything is unchanged |
|  | Press the button „Tovább” | A pop-up is shown wit the message „Legalább egy játékmódot válasszon ki!” |
|  | Press the „Ok” button | The pop-up disappears and everything is unchanged |
|  | Check the chekbox „Tematikus” | The checkbox is checked, and the time picker next to it gets enabled. It has a value of 1:00 |
|  | Press the „Tovább” button | The page changes to the round setup page. There is no change in the navigation bar. Each team is listed 3 times in the section „Körök beosztása”. The first round is selected. |
|  | Select each round one after another from the list. | Each round has the same „Játékmód” state: „Tematikus” is checked and set to 1:00, „Villám” is unchecked |
|  | Set the „Tematikus” timepicker to 0:00 | The selected time is 0:00, and the timepicker gets disabled, and the checkbox unchecked. |
|  | Check the „Villám” checkbox | The checkbox is checked, and the time picker next to it gets enabled. It has a value of 1:00 |
|  | Set the „Villám” timepicker to 0:00 | The selected time is 0:00, and the timepicker gets disabled, and the checkbox unchecked. |
|  | Press the „Tovább” button | A pop-up shows up, with the message: „Válasszon ki egy kérdéssort!”, „Legalább egy játékmódot válasszon ki!” |
|  | Press the „Ok” button | The pop-up disappears, and everything is unchanged |
|  | //TODO: lejátszott kört tesztelni |  |
|  |  |  |
|  |  |  |

Adatelérési rétegről csak a Mindent vagy semmit programnál beszélhetünk, ebben az alfejezetben erről lesz szó.

# Összegzés

# További fejlesztési lehetőségek

# Hivatkozások

Vollai, G. (2019. május 15). *Retro TV #1 – Mindent vagy Semmit!* Forrás: ubulvilaga.com: https://ubulvilaga.com/2019/05/15/retro-tv-1-mindent-vagy-semmit/

1. Ez az általam elvégzett középiskolában található, és nem az én tulajdonomat képezi. [↑](#footnote-ref-1)
2. Ez lehet bármilyen tetszőleges szabály, akár véletlen szám generálással is történhet. [↑](#footnote-ref-2)
3. https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide [↑](#footnote-ref-3)
4. https://docs.microsoft.com/en-us/dotnet/maui/what-is-maui [↑](#footnote-ref-4)
5. https://docs.microsoft.com/en-us/windows/apps/winui/winui3/ [↑](#footnote-ref-5)
6. Korábban Project Reunion (bővebben: https://docs.microsoft.com/en-us/windows/apps/windows-app-sdk/) [↑](#footnote-ref-6)