



SZAKDOLGOZAT

Lippai féle határidő napló

Lippai Máté

konzulens: Bende Attila



2020-04-10

Tartalom

Eredetiségi nyilatkozat	2
KEZDETI ELKÉPZELÉS	3
FEJLESZTŐI DOKUMENTÁCIÓ	4
Fejlesztői környezetek	4
.Net	4
A Program felépítése	5
Adatbázis.....	6
Azure	6
Miért pont az Azure	7
Adatbázis felépítése	8
Funkciók	9
Bejelentkezés (frm_bejelentkezés).....	9
Regisztráció(frm_regisztráció).....	9
Súgó	9
Nyelvek	9
Események kezelése (frm_esemeny).....	11
Speciális megoldások	12
Rendszerkövetelmények.....	13
Diagrammok	14
Tovább fejlesztési lehetőségek	16
Munka folyamatok.....	16
Összehasonlítás	17
Munka értékelése	18
FELHASZNÁLÓI SEGÉDLET	19
Fontos tudni valók.....	19
Rendszer követelmények	19
Főmenü és a Regisztrálás	19
Regisztráció	19
Bejelentkezés.....	19
Nyelvek beállítása	20
Használati útmutató	21
Hivatkozás jegyzék.....	22

Eredetiségi nyilatkozat

NYILATKOZAT A SZAKDOLGOZAT EREDETISÉGÉRŐL

Alulírott a **BMSZC Verebély László Szakgimnáziuma és Szakközépiskolájának 54 213 05 OKJ Szoftverfejlesztői képzésében** részt vevő hallgatója büntetőjogi felelősségem tudatában nyilatkozom és aláírással igazolom, hogy

a című

..... című

szakdolgozat saját, önálló munkám, és abban betartottam az iskola által előírt, a szakdolgozat készítésére vonatkozó szabályokat.

Tudomásul veszem, hogy a szakdolgozatban plágiumnak számít:

- szó szerinti idézet közlése idézőjel és hivatkozás nélkül,
- tartalmi idézet hivatkozás megjelölése nélkül,
- más publikált gondolatainak saját gondolatként való feltüntetése.

E nyilatkozat aláírásával tudomásul veszem továbbá, hogy plágium esetén szakdolgozatom visszautasításra kerül.

Budapest, 2019. április 10.

Hallgató aláírása

KEZDETI ELKÉPZELÉS

- **ALAP ADATOK**

- A szakdolgozat témája: több felhasználós naptár.
- Felület: Asztali felület.

Az alkalmazás automatikusan elindulna miután a felhasználó bejelentkezett. Ez a funkció szabadon ki és bekapcsolható. A felhasználót egy főmenü fogadja, ahol a szükséges adatok megadása után a bejelentkezhet. Az új felhasználó a regisztráció gombra kattintva létrehozhatja a saját fiókját, amibe majd jelentkezik.

Bejelentkezés után a felhasználót a felhasználói felület fogadja, ahol láthatja az napi dátumot a pontos időt és az aktuális dátumot. Valamint a havi feladatokat, eseményeket, időpontokat.

A program lehetőséget ad a felhasználónak, hogy a naptarát személyre szabhassa

- **FUNKCIÓK A FŐMENÜ KÉPERNYŐN**

- bejelentkezés
- regisztráció

- **FUNKCIÓK**

- Esemény felvétele a naptárba
- Esemény módosítása és törlése
- emlékeztető beállítása és törlése

- **REGISTRÁCIÓHOZ SZÜKSÉGES ADATOK(ADATBÁZIS)**

- vezetéknév
- keresztnév
- felhasználónév
- jelszó
- születési dátum
- nem

- **ESEMÉNYEKET TÁROLÓ ADATBÁZIS**

- esemény kezdete
- esemény vége
- esemény neve, címe megnevezése

FEJLESZTŐI DOKUMENTÁCIÓ

Fejlesztői környezetek

- Az alkalmazás megírásakor a Microsoft Visual studio keretrendszert használtam.
- A választott programozási nyelv C#.
- Az adatbázis létrehozásához az AZURE portal
- A súgó HTML és CSS állománya a VScode alkalmazással készült
- Az alkalmazásban látható grafikai elemek a Krita nevű rajzoló programmal készült

.Net

„A Microsoft által készített .NET keretrendszer (a *.NET Framework*) gyors alkalmazásfejlesztést (RAD), platformfüggetlenséget és hálózati átlátszóságot támogató szoftverfejlesztői platform. A keretrendszert a korábbi platform, a COM leváltására szánták.

Eredetileg a .NET kifejezés nemcsak fejlesztői környezetet jelentett, hanem fejlesztőeszközök, szoftverek, sőt hardvereszközök összességét is. Az évek során a kép kitisztult, így mostanra a .NET alatt a keretrendszert értjük.

A .NET Framework eszköztára a szoftverfejlesztés szinte minden aspektusát (kliens-, illetve szerveroldali megoldások, adatbázisok kezelése, játékfejlesztés stb.) lefedi.

A .NET Framework alapját a CLI, vagyis a Common Language Infrastructure képezi. Ez nem más, mint azon szabályok halmaza, amelyek leírják egy nyelvfüggetlen fejlesztői környezetet, a futtatókörnyezetet, típusrendszert stb. A .NET Framework implementációját CLR-nek, Common Language Runtime-nak hívják. A CLI maga is négy fő részre oszlik:

Common Language Specification

A CLS a CLI része. Azokat a szabályokat írja le, amelyeket a CLI-kompatibilis nyelveknek be kell tartaniuk. Érdekes, hogy a legtöbb .NET nyelv – pl. a C# – tartalmaz olyan elemeket, amelyek nem felelnek meg a CLS specifikációnak (pl. a CLS nem engedi meg ulong típusú paraméterek használatát, míg a C# igen).

Common Type System

A CTS a CLI azon része, amely a típusokat, azok memóriabeli reprezentációját, illetve egymással való interakcióját írja le. A .NET minden nyelve ugyanazt a típusrendszert használja, és ugyan az egyes típusok megnevezése nyelvfüggő, viszont mindig ugyanarról a típusról van szó.

Common Language Runtime

A CLR vagy VES (Virtual Execution System) a CLI nyelven megírt programok betöltéséért és végrehajtásáért felel. Felelősséggel tartozik pl. a memóriamenedzsmentért, kivételkezelésért, illetve a kódbiztonságért is (itt a fontosabbak lettek említve).

Common Intermediate Language

A CIL (korábban MSIL) egy ún. köztes kód. Minden CLI nyelvben megírt program erre a kódra fordítódik le (ellentétben a hagyományos nyelvek natív kódjától). Ezt a kódot a tényleges futtatáskor az ún. jitter (Just in time compiler) fordítja le natív kódra, amelyet a processzor már tud kezelni. A CIL hasonlít a Java bytekódjára.”

A Program felépítése

Az alkalmazás három ablakból áll. Minden ablaknak külön célja van. Az alkalmazáshoz tartozik egy online adatbázis.

- **Fő menü**

Ez az ablak jelenik meg elsőnek. Itt lehet bejelentkezni, valamint megnyitni a súgót. A regisztráció gombra kattintva az ablak bezárul és megnyílik a regisztrációs ablak.

Név	toolbox (C#)	megjegyzés
tb_felhnev tb_jelszo	TextBox	sima szöveg doboz ha bele kattintanak üres lesz
bnt_bejel	Button	alaphelyzetbe le van tiltva lenyomásra végre hajtódik a bejelentkezés és új ablakot nyit meg
btn_reg	Button	lenyomásra új ablakot nyit meg
btn_sos	Button	lenyomásra megnyitja a sugo.html fájlt
lbl_marvan	Label	rákattintáskor használható a btn_bejel
pln_felhn pln_jelsz	Panel	A Textbox-ba kattintva kicserélődik
cmb_nyelvek	Combobox	A nyelv kiválasztására szolgál

- **Regisztráció**

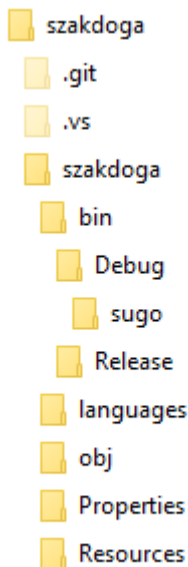
Ebben az ablakban lehet elvégezni a regisztrációt. Az adatok kitöltése után az ablak bezárul és megnyílik az esemény kezelő a program főrésze. A regisztráció során megadott adatokat a program egy adatbázisba menti el.

Név	toolbox (C#)	megjegyzés
tb_felhnev tb_jelszo tb_vnev tb_knev	TextBox	sima szöveg doboz ha bele kattintanak üres lesz
btn_ok	Button	új ablakot nyit meg lenyomáskor végre hajtódik a regisztráció
pln_felhn pln_jelsz pln_vnev pln_knev	Panel	A Textbox-ba kattintva kicserélődik

- **Esemény kezelés**

Az esemény kezelőben új eseményeket lehet létrehozni ezeket az eseményeket módosítani, valamint törölni lehet. Az újonnan létrehozott eseményeket és a módosításokat egy adatbázisba menti.

- mappa szerkezet



Adatbázis

Az adatbázist az Microsoft Azure rendszerével hoztam létre. Az adatbázisban kettő tábla található. Egyik a felhasználók adatait tárolja a másik a felhasználó által létrehozott eseményeket tárolja. Mivel az adatbázis a Microsoft szerverén van az alkalmazás online bárholnan használható.

Azure

„A Azure SQL Database felügyelt szolgáltatásként biztosított, általános célú, összehasonlítható adatbázis. Ezzel egy magas rendelkezésre állású és nagy teljesítményű adattárolási réteget hozhat létre az Azure-beli alkalmazások és megoldások számára. A SQL Database a különböző modern felhőalapú alkalmazások számára megfelelő választás lehet, mivel lehetővé teszi a kapcsolati és nem kapcsolati struktúrák, például a diagramok, a JSON, a térbeli és az XML feldolgozását.

Ez a Microsoft SQL Server adatbázismotor legújabb stabil verziójára épül. Használhatja a speciális lekérdezés-feldolgozási funkciókat, például a nagy teljesítményű memóriában lévő technológiákat és az intelligens lekérdezések feldolgozását. Valójában a SQL Server legújabb képességei SQL Database, majd SQL Server önmagára. A legújabb SQL Server képességek javítása és frissítése nélkül, több millió adatbázison keresztül tesztelve.

A SQL Database lehetővé teszi a teljesítmény egyszerű meghatározását és méretezését két különböző beszerzési modellen belül: egy virtuális mag-alapú vásárlási modellt és egy DTU-alapú vásárlási modellt. A SQL Database egy teljeskörűen felügyelt szolgáltatás, amely beépített magas rendelkezésre állást, biztonsági másolatokat és egyéb gyakori karbantartási műveleteket tartalmaz. A Microsoft az SQL és az operációs rendszer kódjának összes javítását és frissítését kezeli. Nem kell kezelnie a mögöttes infrastruktúrát.

Rugalmas készletek az erőforrások kihasználtságának maximalizálásához

Számos vállalkozás és alkalmazás számára elegendő az, ha önálló adatbázisokat tudnak létrehozni, majd azok teljesítményét szükség szerint felfelé és lefelé tudják skálázni – különösen akkor, ha a használati minták viszonylag jól jelezhetők előre. A kiszámíthatatlan használati szokások miatt a költségek és az üzleti modell is nehezen kezelhető. A rugalmas

készletek megoldást jelentenek erre a problémára. A teljesítmény adatokat egy készlethez kell hozzárendelni, és nem egy önálló adatbázishoz. A készlet kollektív teljesítménybeli erőforrásaiért kell fizetnie, nem pedig egyetlen adatbázis teljesítményére.

A rugalmas készletekkel az erőforrásigények ingadozásakor sem kell a az adatbázis teljesítményének fel- és leskálázására koncentrálnia. A rugalmas készletbe helyezett adatbázisok szükség szerint használják fel a teljesítmény-erőforrásokat. A rugalmas készletbe helyezett adatbázisok használják az adatbáziskészlethez hozzárendelt teljesítményt, de nem lépik túl az adatbáziskészlet teljesítménykorlátait, így költségei előre jelezhetők még akkor is, ha az egyéni adatbázis-használat nem jelezhető előre.

Hozzáadhat és eltávolíthat adatbázisokat a készlethez, az alkalmazást egy maroknyi adatbázisból több ezerre, az Ön által felügyelt Költségvetésen belül. A készlet adatbázisai számára elérhető erőforrások minimális és maximális mennyiségét is szabályozhatja, így biztosíthatja, hogy a készlet egyetlen adatbázisa se használja fel a készlet összes erőforrását, és hogy minden készletezett adatbázisnak garantált minimális erőforrása legyen. A rugalmas készleteket használó szoftveres (SaaS-) alkalmazások tervezési mintáinak megismeréséhez lásd: tervezési minták több-bérlős SaaS-alkalmazásokhoz SQL Database használatával.

Azure SQL Database: Önálló adatbázis

Az Azure SQL Database a Microsoft SQL Server-adatbázismotor legújabb, stabil verzióján alapuló relációsadatbázis-szolgáltatás (DBaaS). Az SQL Database egy nagy teljesítményű, könnyen használható, megbízható és biztonságos adatbázis, amellyel új alkalmazásokat, webhelyeket és mikró szolgáltatásokat hozhat létre egy tetszőleges programozási nyelven, infrastruktúrakezelés nélkül. Ismerje meg, hogyan használhatja az Azure SQL Database szolgáltatást a gyors útmutatóink, oktatóanyagaink és példáink segítségével.

Rugalmas készletekkel nem kell összpontosítania az adatbázis teljesítményének felfelé és lefelé állítására, mivel az erőforrások igénye ingadozik. A rugalmas készletbe helyezett adatbázisok szükség szerint használják fel a teljesítmény-erőforrásokat. A készletezett adatbázisok felhasználják, de nem lépik túl a készlet korlátait, így a díjszabás akkor is kiszámítható marad, ha az egyes adatbázisok használata nem.

Hozzáadhat és eltávolíthat adatbázisokat a készlethez, az alkalmazást egy maroknyi adatbázisból több ezerre, az Ön által felügyelt Költségvetésen belül. A készlet adatbázisai számára elérhető erőforrások minimális és maximális mennyiségét is szabályozhatja, így biztosíthatja, hogy a készlet egyetlen adatbázisa se használja fel a készlet összes erőforrását, és hogy minden készletezett adatbázisnak garantált minimális erőforrása legyen. A rugalmas készleteket használó szoftveres (SaaS-) alkalmazások tervezési mintáinak megismeréséhez lásd: tervezési minták több-bérlős SaaS-alkalmazásokhoz SQL Database használatával.”

Miért pont az Azure

Napjainkban az egyik legmodernebb szolgáltatások közé tartozik az Azure. Szerettem volna én is kipróbálni és a legmegfelelőbbnek a szakdolgozatot találatam. Számos kényelmi lehetőséget rejt és rengeteg hasznos plusz szolgáltatást, amelyekkel segítségével még hatékonyabb a fejlesztés.

A számomra az egyik legfontosabb talán a rugalmas adatbázis készlet. Amelyet nem használtam ennek az több oka is van. A legfontosabb, hogy a rugalmas adatbázis készletet üzleti célokra és jóval több adatbázis kezelésére.

Adatbázis felépítése

Az adatbázis táblái:

Tábla neve:	felhasznalok	esemenyek
Tábla tartalma:	a felhasználók adatai	az események adatai
Felhasználás:	bejelentkezés	események

Tábla: felhasznalok

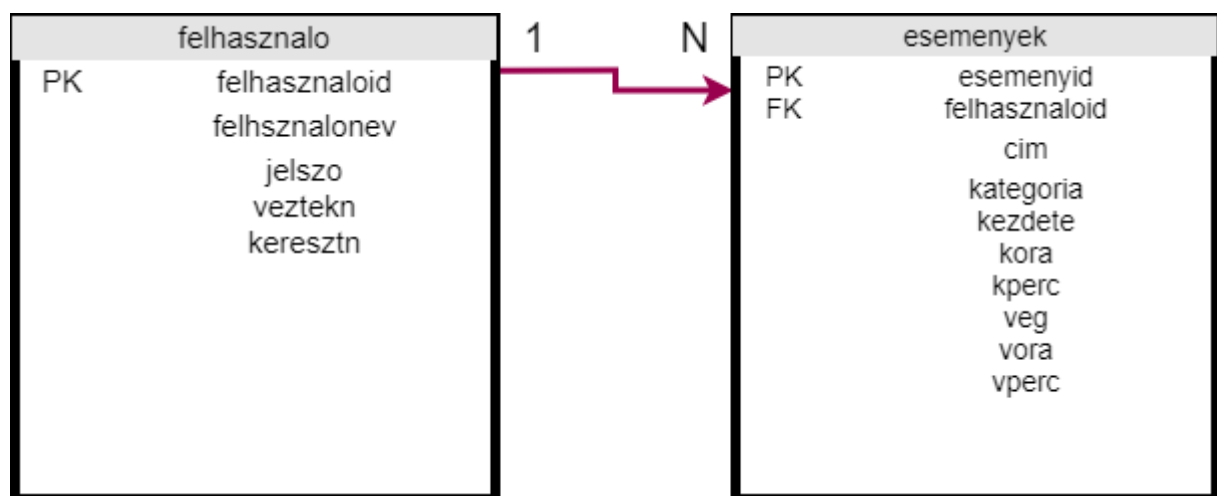
mező neve	mező leírása
felhasznaloid	A felhasználó azonosító száma elsődleges kulcs
felhasznalonev	A név amivel bejelentkezik
jelszo	A felhasználó jelszava
vezetecn	A felhasználó vezetéknév
keresztne	A felhasználó keresztnév

Tábla: esemenyek

mező neve	mező leírása
esemenyid	Az esemény azonosítója (sorszám) elsődleges kulcs
felhasznaloid	A felhasználó azonosítója idegenkulcs össze kapcsolja a táblákat
cim	Az esemény neve címe
kategoria	Az esemény kategóriája
kezdetek	Az esemény kezdetének dátuma
kora	Az esemény kezdetének az órája
kperc	Az esemény kezdetének perce
veg	Az esemény végének a dátuma
vora	Az esemény végének órája
vperc	Az esemény végének perce

A táblák kapcsolata:

A felhasználó és az esemény tábla között 1: N (egy a többhöz kapcsolat) van. Ugyanis minden felhasználóhoz több esemény is kapcsolódik.



Az Azure adatbázis rendelkezik egy tűzfallal, amely nélkül az alkalmazás nem éri el az adatbázist. Az adatbázis csak akkor működő képes, ha a felhasználó IP címét hozzá adjuk a tűzfalhoz. Sajnos nem találtam megoldást arra hogy a felhasználó IP címét automatikusan

Funkciók

Bejelentkezés (frm_bejelentkezés)

A bejelentkezés egy SELECT SQL utasítással hajtóik végre. A mezőkbe írt szöveget a program ellenőrzi, hogy szerepel-e az adatbázisba. Ezt egy elágazás vizsgálja, ha az egyezést talál, beengedi a felhasználót. Ellenkező esetben hiba üzenetet ír „a bejelentkezés sikertelen”.

A valóságban az utasítás egy if else elágazásban van. A feltétel az, hogy a felhasználónév és a jelszó mezők nem lehetnek az alapértelmezett. Az else ágban egy hiba üzenet jelenik meg: „a bejelentkezés sikertelen”. Ha a mezők mégse üresek, de az adatok hibásan lettek megadva akkor is kiírja a hiba üzenetet. A felhasználónév alapján az id-t lekérdezi az esemény(frm_esemény) ablakban.

feltétel ne legyen üres mező,

 ha feltétel = igaz

 lefut az SQL parancs, sikeres bejelentkezés

 ha feltétel = hamis

 hiba üzenet

Regisztráció(frm_regisztráció)

A regisztráció az INSERT INTO SQL paranccsal hajtóik végre. A parancs egy elágazásban szerepel. A mezők nem lehetnek üresek, ha a feltétel teljesül (nincs üres szöveg doboz) akkor megjelenik egy üzenet ezzel a szöveggel: „a regisztráció sikeres”. A program a felhasználói felületre ugrik. Ha van üres szöveg doboz megjelenik egy hiba üzenet: „a regisztráció sikertelen” és a program nem hagy, ha el a regisztrációs űrlapot.

Feltétel ne legyen üres mező,

 ha feltétel = igaz

 lefut az SQL parancs, menti az adatokat az adatbázisba sikeres regisztráció(üzenet)

 ha feltétel = hamis

 hiba üzenet

Súgó

A súgó egy gomb lenyomására jelenik meg a böngészőben. A súgó egy HTML oldal, ami rendelkezik egy CSS-el. A HTML-ben nincs, semmilyen funkciót nem tartalmaz. A súgó a felhasználói dokumentáció egy részét tartalmazza. A súgóban található képek és maga a súgó egy sugo nevű mappában található a bin/debug mappában. A HTML csak tageket tartalmaz minden formázás a CSS-ben van. A formázások nagy része osztályokban van megvalósítva.

Nyelvek

Az alkalmazásban kettő nyelv található angol és magyar. A nyelvek a langleges nevű mappában vannak. Minden nyelvhez tartozik egy .resx fájl. A nyelveket egy legördülő listából lehet ki választani. Alap értelmzetten a magyar van bállítva.

- eredeti
eredetileg a nyelvek beállítását egy-egy elágazás felelt az elágazás vizsgálta, hogy melyik érték van kiválasztva egy nyelvek nevű változó segítségével. A változó vizsgálta hogy mi van kiválasztva a legördülő listából az alapján választotta ki a megfelelő fájlt.
- A végleges változat
A végleges változat is egy változóval A remeg nevűváltozó már egyből a mappát vizsgálja. Nem szükséges az elágazás. Ennek az előnye, hogy további nyelvek hozzáadása egyszerűbben végezhetőel.

Angol nyelv eng.resx		magyar hun.resx	
befe	End	befe	Befejezés
bejel	Log in	bejel	Bejelentkezés
cim	Title	cim	Cím
esemeny	Events	esemeny	Események
felhnev	Username	felhnev	Felhasználónév
jelszo	Password	jelszo	Jelszó
kat	Category	kat	Kategória
kezd	Beginning	kezd	Beginning
kilep	Quit	kilep	Kilépés
knev	First name	knev	Keresztnév
marvan	I already have an account	marvan	már van fiókom
reg	Registration	reg	Regisztráció
sorsz	Number	sorsz	Sor szám
vnev	Last name	vnev	Vezetéknév

Események kezelése (frm_esemeny)

Három opció lehetséges mentés módosítás törlés. Mind három opció ez-egy SQL paranccsal hajtóik végre egy-egy gomb megnyomásával

Mentés	INSERT INTO SQL parancs
Módosítás	UPDATE SQL parancs
Törlés	DELETE SQL parancs

Az adatok megadása után a mentés gombra (kék) kattintva az adatok elmentődnek az adatbázisba és kiírja a DataGridView-ba. Eredetileg két oszlopodba mentődöttek volna az adatok elsőbe a sorszámok(id.) a másodikban a megadott adatok az alábbi sorrendben: név kategória kezdet vége. A kezdete és a vége három adatból áll össze. Minden adat külön oszlopba jelenik meg A törlés gomb (piros) melletti. Szövegdobozba beírt szám alapján jelöli ki a sort. A módosítás gombra (zöld) kattintva lehet módosítani a törlés gombra (piros) kattintva kitöröli a sort az adatbázisból. Az alábbi táblázat röviden leírja, hogy mi hogyan működik.

Név	toolbox (C#)	megjegyzés
tb_cim	TextBox	sima szöveg doboz, ha bele kattintanak üres lesz
tb_sorszam	TextBox	Ide kerül az esemény sorszáma módosítás és törlés funkciókhoz
cmb_kateg	ComboBox	az értékek a properties fülbe lettek beírva
cmb_kezd_o cmb_veg_o	ComboBox	az időponthoz az órákat tölti fel részletesebben a következő oldalon
cmb_kezd_p cmb_veg_p	ComboBox	az időponthoz az percekét tölti fel részletesebben a következő oldalon
dtp_kezdet dtp_veg	DateTimePicker	nincs semmi egyéb funkció
dvg_esemenyek	DataGridView	megjeleníti az eseményeket egy metódus segítségével
lbl_id	Label	A bejelentkezett felhasználó id-je jelenik meg ezzel lesznek a táblák összekötve.

Eseményke felépítése

cím:	a cím mezőbe beírt szöveg
kategória:	a legördülő listából kiválasztott egyik érték
Kezdet:	a naptárból kiválasztott dátum és a két legördülő listából kiválasztott szám érték
vége:	a naptárból kiválasztott dátum és a két legördülő listából kiválasztott szám érték

Az hibás értékek megadására minimális az esély. A kategóriákat kivéve mindenhol van alapértelmezett érték. Emiatt nem tartottam fontosnak a hibák vizsgálatát.

A DataGridView-t a tablafel() metódus tölti fel. Az adatai

Speciális megoldások

1. képek és szöveg színek
 - a. A paneleken látható képek a látvány kedvéért megváltoztatják a színüket. A mezőkre kattintva a program kicseréli a képet egy másik színűre.
2. szövegek a mezőkben
 - a. A mezők mellett nincs szöveg, hanem az alapértelmezett szövege segíti a felhasználót, hogy mit írjon bele. A mezőre kattintva az alap értelmezett lila szöveg eltűnik, és amit a felhasználó begépel, kék színű, jelenik meg
3. Bejelentkezés gomb
 - a. A Bejelentkezés gomb elinduláskor le van, tiltva csak a már van, fiókom szövegre kattintva lehet
4. Súgó
 - a. A súgó egy gomb lenyomására megjelenik egy HTML, ami tartalmazza a felhasználói dokumentáció egy részét.
5. óra
 - a. Az óra furcsán van, megadva az adatbázisban két oszlopban szerepel egy óra és egy perc oszlopban.

A legördülő mezők kettő számláló ciklussal vannak feltöltve az első az órákat a második a perceket tölti fel. Az órákat egy **metódus** tölti fel, ami a program betöltésekor lefut **orafell()**:

A metódusban két for ciklus található Minden ciklusban van egy elágazás, amikben két másik elágazás van (if-be ágyazott két másik if). Az első ciklus 0-tól 23-ig ki írja a számokat egyesével. A második ciklus 0-tól 55-ig írja ki a számokat. Az elágazások célja, hogy a kiírás eredménye olyan legyen, mint egy digitális óra.

eredeti metódus

Eredetileg az órák és a percek egy-egy változóban voltak letarolva, amiknek a kezdő értéke 0 volt. ciklusok ezeket a változókat növelték egyesével.

változtatás okai

Rájöttem, hogy változók nélkül is ugyan úgy működik.

Jobban áttekinthetőbbek a ciklusok.

könnyebb megvalósítani, hogy a 10-nél kisebb számok elé kiírjon egy 0-át

Pl.:05:12. Az eredetivel nem volt lehetséges

A percek esetében a teljesen felesleges egyesével kiírni a számokat elég csak ötösével

órát feltöltő ciklus	percek feltöltő ciklus
Ciklus 0-tól 24-ig feltétel $o \% = 0$ feltétel $o < 10$ kiír "0" + o feltétel $o \geq 10$ kiír o ciklus vége	Ciklus 0-tól 60-ig feltétel $p \% = 5$ feltétel $p \leq 5$ kiír "0" + o feltétel $p > 10$ kiír p ciklus vége

Rendszerkövetelmények

- CPU:
 - kezdetben a terhelés 28%
 - bármilyen módosítás esetén egy pillanatra felugrik 32-35% közé. Ez az érték függhet az eszköztől, amelyiken futtatjuk.
 - Alaphelyzetben folyamatosan 0-5%
- Memória:
 - induláskor 18MB ez a minimális memória igény, ami szükséges az indulás
 - Ha átváltunk egy másik ablakra az érték 19MB-ra emelkedik, de nem megy feljebb
 - Az érték 20MB fölé megy amikor az adatbázist kell használnia

Ezek nem a pontos értékek. ekkor ugyanis nincs megfelelő kapcsolat az adatbázissal. A regisztrációsorán mikor rákattintottam a regisztráció gombra megpróbálta elérni az adatbázist. A hiba üzenet után mutatta a 20MB értéket. Ez egy előzetes elemzés eredménye a végleges értékek várhatóan hasonlóak lesznek.

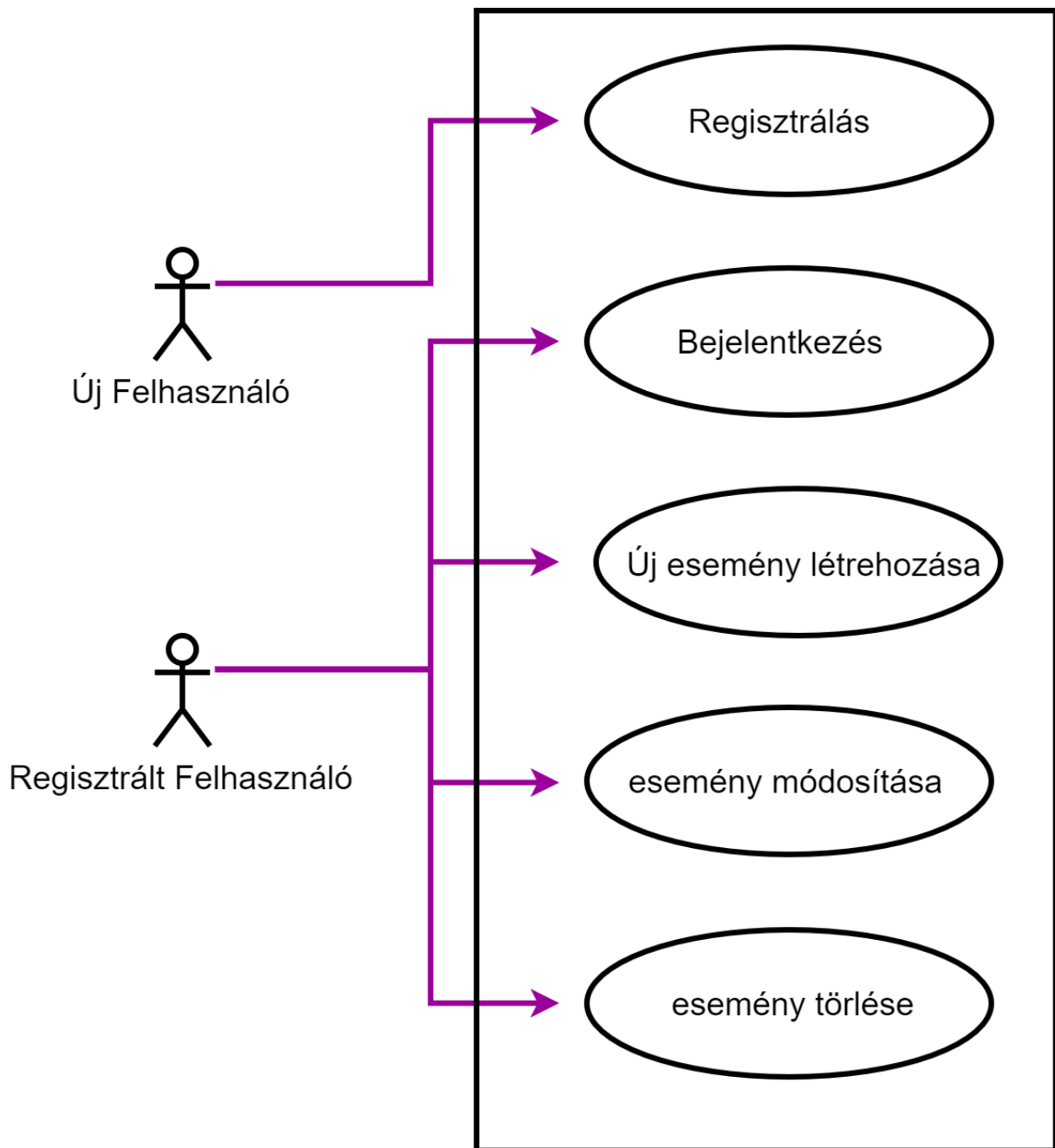
Végleges rendszerkövetelmények kerülnek majd a felhasználói dokumentációba

Fejlesztés során a fejlesztői eszköz újra lett telepítve. Ennek vényében a gép gyorsabb lett és az értékek is javultak.

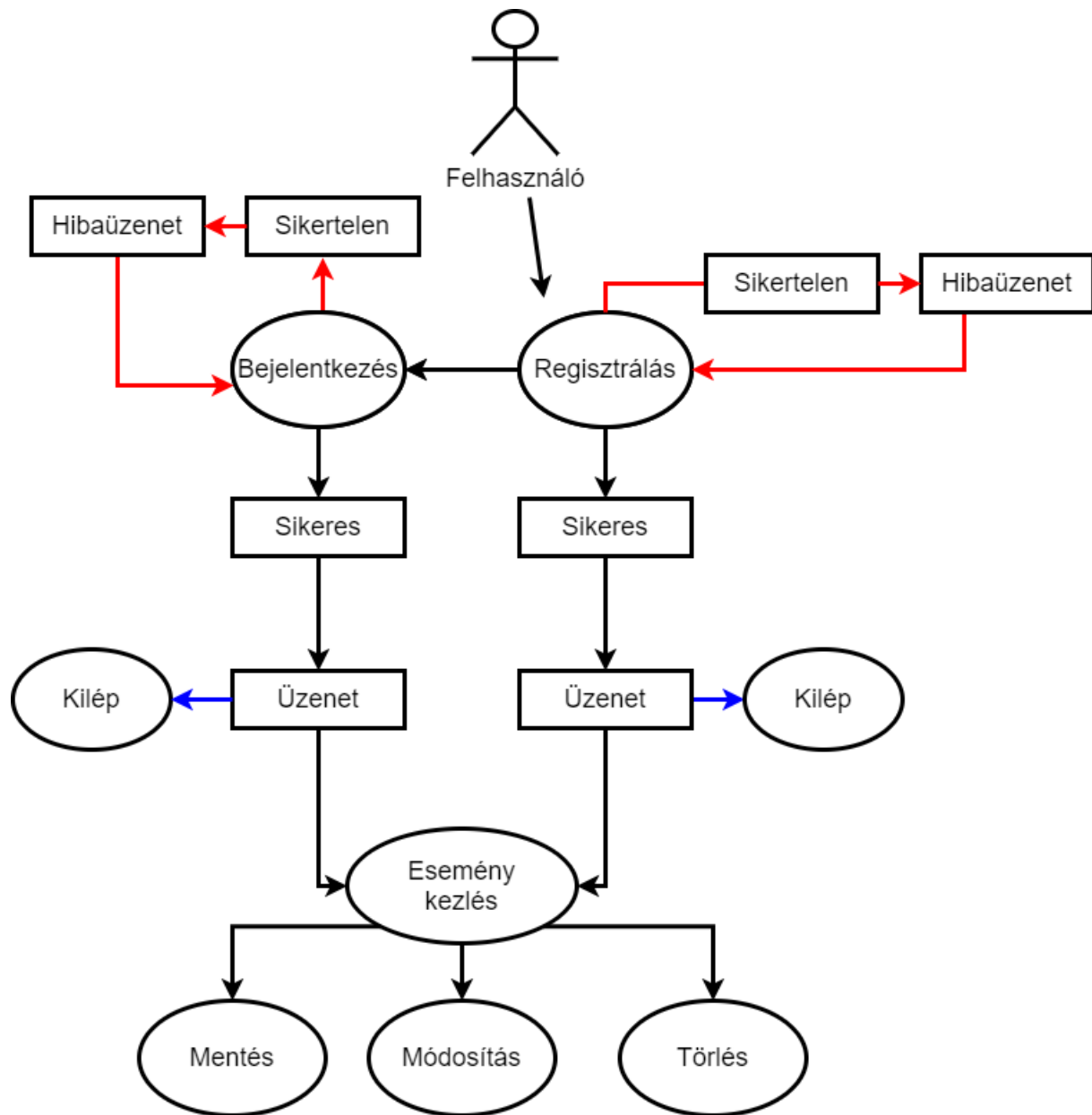
- CPU:
 - kezdetben a terhelés 21%
 - bármilyen módosítás esetén a maximális érték nem haladta meg a 15%-ot. Ez az érték az eszköztől függően változik
 - Alaphelyzetben folyamatosan 0-5%
- Memória:
 - induláskor 18MB ez a minimális memória igény, ami szükséges az indulás
 - Ha átváltunk egy másik ablakra az érték 19MB-ra emelkedik, de nem megy feljebb
 - Az érték 20MB fölé megy, amikor az adatbázist kell használnia

Ez egy körülbelüli értéket add, hogy egy gyengébb és egy erősebb eszköz esetén milyen teljesítményre lehet számítani. Ezek egy körülbelüli értékek a legjobb és a legrosszabb teljesítményre.

Diagrammok



Valójában egy aktoros az alkalmazás csak így tudtam szemléltetni, hogy regisztráció nélkül nem használható a program.



Tovább fejlesztési lehetőségek

- Személyesebbé tétel
 - profil kép hozzáadásának lehetősége
 - Helló <felhasználó kereszt neve>!
- adatbázis növelése
 - felhasználói szintek
főnök (magasabb szintű felhasználó)
beosztott (átlag felhasználó)
 - Felhasználói szintek és jogok megírása
 - Regisztráció kiegészítése, hogy az alkalmazás el tudja különíteni a szinteket
 - új tábla a beosztásoknak
 - Részletesebb felhasználói fiók→
 - a felhasználó tábla kiegészítése (lakcím szül. ...stb.)
 - b Új tábla a személyes adatoknak
- Alkalmazás fejlesztése
 - Új felület a felhasználók beosztásának
 - több nyelv felvétele
 - regisztrációs űrlap kiegészítése a személyes adatokkal
 - Új felület létrehozása a személyes adatok megadásához

Munka folyamatok

1. tervezési fázis
Ebben a szakaszban végig gondoltam mit is takar a választott téma. Kipróbáltam néhány hasonló alkalmazást, hogy legyen egy elképzelésem, hogy mit is kell létrehoznom
2. előzetes program tervezet
Csináltam egy-két próba programot. ezekben a programokban csak a gombokat és a szövegdobozokat használtam. Csak a gombokat programoztam le, hogy melyik gomb mikor mit csináljon
3. Adatbázis tervezés
Miután volt egy egyszerű programom, amiből ki lehetett indulni elkezdtem megtervezni az adatbázist. Végig gondoltam, hogy milyen adatokkal is dolgozik egy hasonló alkalmazás. A szükséges adatok össze gyűjtése után létrehoztam az adatbázist
4. Program tervezet adatbázissal
Már volt egy kezdetleges alkalmazásom és egy használható adatbázisom. Elkezdtem az alkalmazást összekapcsolni az adatbázissal. Megcsináltam a funkciók kezdetleges verzióit. Valamint ki alakítottam az alkalmazás végleges kinézetét.
5. Kezdeti dokumentáció
A már meglévő kezdeti funkciókról elkezdtem megírni a dokumentációt. Valamint a közeljövőben megvalósítandó funkciókból

6. Az alkalmazás befejezése
Befejeztem az alkalmazás kódolását. Megírtam az utolsó kódrészleteket kiegészítettem a meglévőket a hibákat kijavítottam.
7. Dokumentáció megírása
Befejeztem a fejlesztői dokumentációt, maradékát. A kiegészítettem és átírtam néhánydolgot. Miután készlettem a fejlesztői dokumentációval végig gondoltam mit is kéne a felhasználói dokumentációba írni. Végül megírtam a felhasználói dokumentációt

Összehasonlítás

- Elképzelés
 - Nyíljon meg az asztal indulásakor
 - Legyen bene regisztráció és bejelentkezés
 - Legyen egy adatbázis lokális adatbázis
 - Naptár személyre szabása
 - Események létrehozása módosítása törlése
 - Havi események kilistázása
- Ami megvalósult
 - regisztráció bejelentkezés
 - Események létrehozása módosítása törlése
 - Események kilistázása
- Ami megvalósult, de módosítva lett
 - Lokális adatbázis helyet Microsoft Azure SQL adatbázis
 - Az első néhány eseményt listáza ki
- Ami nem lett megvalósítva
 - Nyíljon meg az asztal indulásakor
 - Naptár személyre szabása
- A módosítások okai
 - Az adatbázis első oka az volt, hogy amikor az alkalmazást másik eszközön próbáltam elindítani nem találta az adatbázis fájlt. Ennek hatására jött az ötlet, hogy célszerűbb lenne egy online adatbázis.
 - A kilistázása azért lett módosítva, mert nem tudtam a dátumból csak a hónap értéket ki olvasni a lekérdezéshez
- A kihagyások oka
 - Nyíljon meg az asztal indulásakor időszükében hagytam ki
 - A naptár személyre szabása csak kezdetben tűnt jó ötletnek. A fejlesztés közben rá kellett jönnöm, hogy teljesen felesleg a személyre szabás lehetősége

Az alkalmazás a jelenlegi választható kategóriákat figyelembe véve esetleg munkahelyi felhasználásra is alkalmas, valamint iskolák számára diákoknak és pedagógusoknak egyaránt. Ugyan akkor használható különböző munkafolyamatok megtervezéséhez, időhatáros projektekhez tárgyalások megszervezéséhez és munkahelyi időbeosztások pontos tervezésére.

Az alkalmazás lehet, hogy nem tud többet, mint számtalan hasonló alkalmazás, viszont így alkalmam nyílt új technológiát amelyet eddig nem ismertem.

A dizájn kialakításkor elsődlegesen a saját igényeimet vettem alapul valamint a mai generáció felhasználási szokásainak ízlésének megféléően próbáltam kialakítani. Főként osztálytársaknál, barátokál láttam, hogy inkább a sötétebb színű háttérter részesítik előnybe ez a domináns. Ennek hatására jött ez a nagyon sötétszürke háttér ez elég sötét, de mégsem fekete ami (esetleg nyomasztó lehet). A kék és a lila színválasztás általam kedvelt színek. Az árnyalatokat próbáltam úgy kiválasztani, hogy esztétikus, áttekinthető legyen, ugyanakkor ne legyen zavaró a szemnek.

A project során gyakran ütköztem nehézségekbe fény derült szakmai hiányosságokra, és hogy mely területekkel kell többet foglalkoznom, tudásomat gyarapítani és gyakorolni. Ettől függetlenül a problémák megoldására pontosságra törekedtem. Lehetőségeimhez mérten igyekeztem magamból a maximumot kihozni és mások számára valami hasznosat alkotni, ami esetleg megkönnyíti az emberek munkáját.

Munka értékelése

A munkám lehetett volna jobb és kidolgozottabb ha az alábbiakra jobban figyelek és betartom.

- Korábban kezdek, hozzá a szakdolgozat megtervezéséhez
- Pontosabb adatbázis tervezés
- Több konzultáció

A munkám során tapasztalt pozitívumok

- Felmértem a saját hiányosságaimat
- megismertem több fajtamódszert
- új technológiákat próbáltam ki és ismerkedtem meg velük

FELHASZNÁLÓI SEGÉDLET

Fontos tudni valók

Használat előtt mindenképp szükséges a regisztráció. Ehhez kattintson a regisztráció gombra.

A bejelentkezés csak regisztráció után lehetséges.

Bejelentkezéshez kattintson a „már van fiókom” szövegre

Rendszer követelmények

- Memória: 19Mb
- CPU: 100Mb
- tárhely: 16Mb

Főmenü és a Regisztrálás

Regisztráció

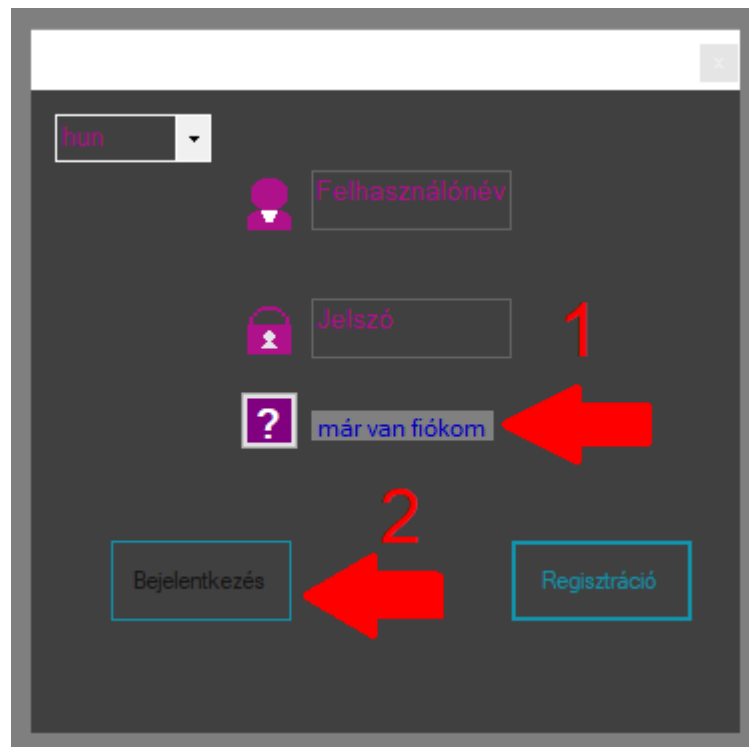
A regisztráció egy külön ablakban történik. A fő menüben lévő regisztráció gombra kattintva megjelenik a regisztrációs űrlap. A sikeres regisztráció után megjelenik egy felugró ablak azzal a szöveggel, hogy a regisztráció sikeres ellenkező esetben hiba üzenetet küld a program: a regisztráció sikertelen. A jelszó nem lehet hosszabb 10 karakternél.

- Regisztrációhoz szükséges:
 - felhasználó név
maximum 50 karakter hosszúságú lehet a hossza
nem ajánlott mind az 50 karaktert felhasználni
 - jelszó
maximum 10 karakter hosszúságú lehet,
A program nem igényel bonyolult felhasználónév és jelszó kombinációt
Nincsenek megkötések bármilyen speciális karakter elfogadható
 - vezetéknév
 - keresztnév
- Tippek a jelszóra
 - legyen könnyen megjegyezhető
 - Használjon kis és nagybetűket, számokat és speciális karaktereket
 - Lehetőleg ne legyen:
születési dátum
kisállat neve
családtag vagy önhöz közel álló személyneve

Bejelentkezés

A bejelentkezés a fő menüben történik. Bejelentkezni a felhasználónév és a jelszó megadásával lehetséges. A sikeres bejelentkezés után a Főmenü bezárul és megnyílik az eseménykezelő felület.

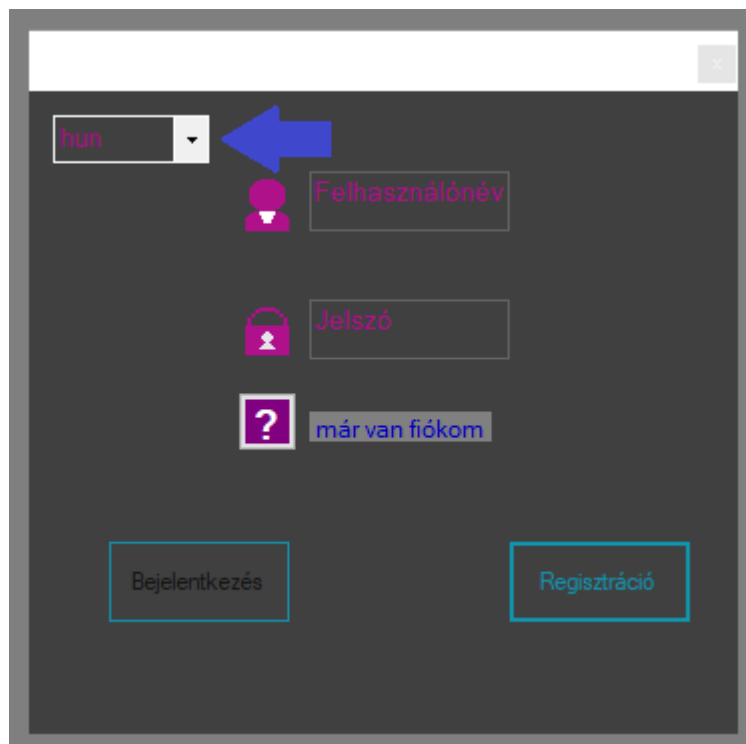
- Bejelentkezéshez szükséges adatok
 - felhasználónév
 - jelszó



Nyelvek beállítása

Két nyelvi beállítás választható: angol és magyar. Alapértelmezetten magyar nyelvű. A nyelv egy legördülőmenüből választhatók ki.

1. Kattintson a legördülő mezőre
2. Válassza ki a nevet



Használati útmutató

Új esemény létre hozása



1. írja be az esemény címét a cím mezőbe
maximum 50 karakter hosszúságú lehet a hossza
2. Adja meg a kategóriát a listából
3. állítsa be a kezdés idejét
4. állítsa be a végének az idejét
5. kattintson a mentés gombra

Esemény módosítása

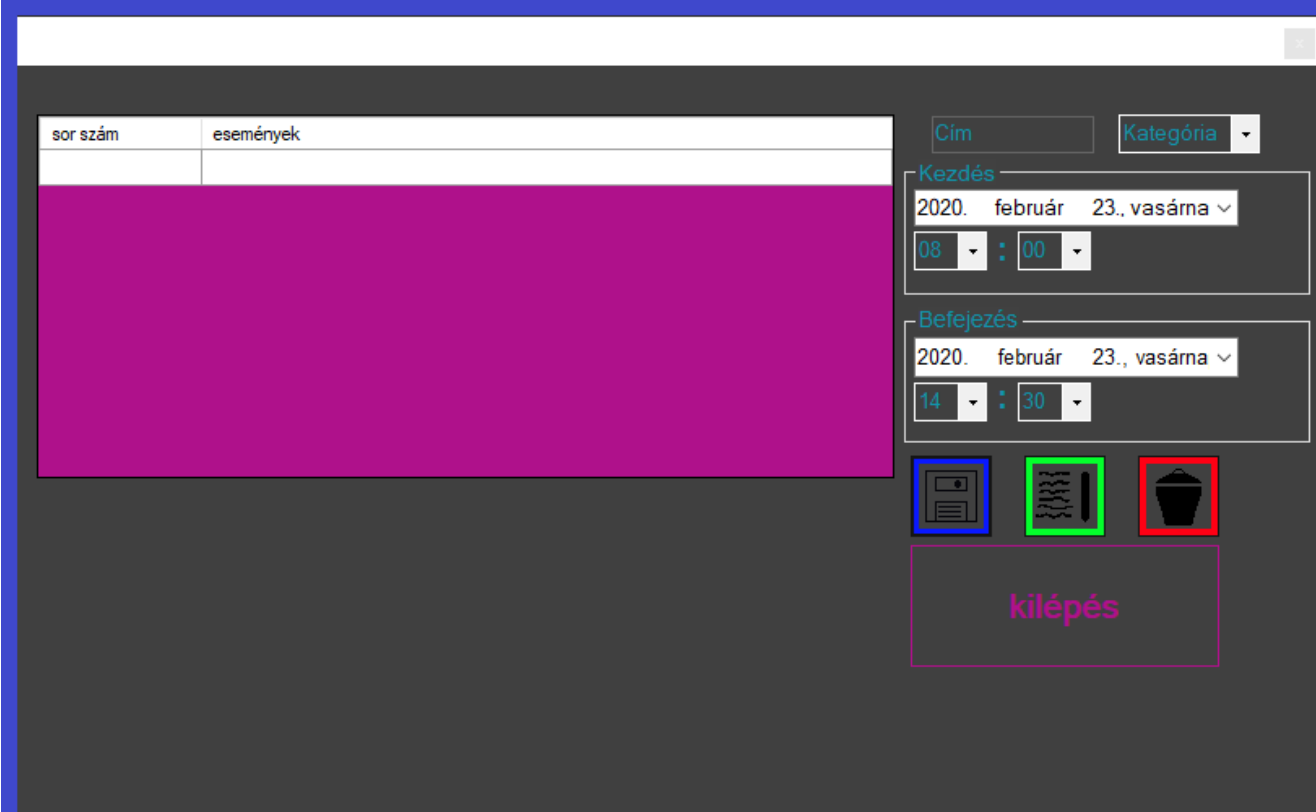
1. válassza ki a módosítani kívánt esemény: kattintson a sorszáma
2. módosítsa az adatokat
3. kattintson a módosítás gombra

Esemény törlése

1. válassza ki a módosítani kívánt esemény: kattintson a sorszáma
2. kattintson a módosítás gombra

gombok		
		
mentés	módosítás	törlés

Az alábbi kép egy régebbi változat az alkalmazásban kicsit módosult!



The screenshot shows a web application interface with a dark background. On the left, there is a table with two columns: 'sor szám' (row number) and 'események' (events). The table body is currently empty. To the right of the table is a form for creating or editing an event. The form includes a 'Cím' (Title) text input field, a 'Kategória' (Category) dropdown menu, and two time selection sections: 'Kezdés' (Start) and 'Befejezés' (End). Each time section has a date and time picker. At the bottom right, there are three action buttons: a blue 'mentés' (save) button with a floppy disk icon, a green 'módosítás' (edit) button with a pen icon, and a red 'törlés' (delete) button with a trash can icon. Below these buttons is a large pink button labeled 'kilépés' (exit).

Hivatkozás jegyzék

- A képek színének módosítása
YouTube videó
cím: MODERN DESIGN Visual Studio C# | TUTORIAL Slide Menu Login UI
feltöltő: [Satya Achmad](#)
- Nyelv beállítása
YouTube videó
cím: Multi language Form using ResourceManager and Culture Info - C#
feltöltő: Dev Bananaer
- .Net Framework
keretrendszer dokumentáció
Wikipedia:
https://hu.wikipedia.org/wiki/.NET_keretrendszer#Architektúra
Microsoft:
<https://devblogs.microsoft.com/dotnet/net-framework-november-13-2019-update-for-net-framework-4-8>
- Azure
Adatbázis dokumentáció
<https://docs.microsoft.com/hu-hu/azure/sql-database/sql-database-technical-overview>
SQL Database-dokumentáció
▼ Áttekintés
Mi az SQL Database?
Melyik SQL-beállítást válasszam?
> Kibocsátási megjegyzések
> Ár- és teljesítménybeállítások
▼ Rövid útmutatók
SQL Database-adatbázisok
Felügyelt SQL-példányok

metszet az oldal menüjéből