**DEBRECENI SZC BEREGSZÁSZI PÁL TECHNIKUM**

4032

Debrecen, Jerikó u.

17.

OM azonosító: 203033

*52*

*/*

*503*

*-*

*52/314*

*Fax:*

*150*

*-*

*204*

*http://www.dszcberegszaszi.hu*

*E*

*-*

*mail:*

*titkarsag@dszcberegszaszi.hu*



**Szakképesítés megnevezése: Szoftverfejlesztő**

**OKJ száma: 54 213 05**

**ZÁRÓDOLGOZAT**

**Munkaidő nyilvántartó rendszer**

**Készítette**: **Konzulens:**

Budai Gergő Gábor Boros Sándor

**Debrecen, 2023**

Tartalomjegyzék

[1 Bevezetés 1](#_Toc133566586)

[1.1 Záródolgozat célja 1](#_Toc133566587)

[1.2 Köszönetnyilvánítás 1](#_Toc133566588)

[2 Fejlesztői dokumentáció 2](#_Toc133566589)

[2.1 Fejlesztői környezet 2](#_Toc133566590)

[2.2 Az adatbázis 3](#_Toc133566591)

[2.2.1 Az adatbázis táblái és mezői 3](#_Toc133566592)

[2.3 A Program 5](#_Toc133566593)

[2.3.1 A program alapjai 5](#_Toc133566594)

[2.3.2 login\_form (bejelentkező felület) 6](#_Toc133566595)

[2.3.3 fooldal\_form 7](#_Toc133566596)

[2.3.4 alkalmazottak\_kezelese form 7](#_Toc133566597)

[2.3.5 jelenlet\_feljegyzese form 9](#_Toc133566598)

[2.3.6 jelenlet\_osszegzo form 10](#_Toc133566599)

[2.3.7 alkalmazottak\_megtekintese form 12](#_Toc133566600)

[2.3.8 fizetes\_kezelese form 12](#_Toc133566601)

[3 Felhasználói dokumentáció 14](#_Toc133566602)

[3.1 Hardware és software igény 14](#_Toc133566603)

[3.2 A program használata 14](#_Toc133566604)

[3.2.1 Bejelentkezés 14](#_Toc133566605)

[3.2.2 Főoldal 15](#_Toc133566606)

[3.2.3 Alkalmazottak kezelése 16](#_Toc133566607)

[3.2.4 Jelenlét feljegyzése 17](#_Toc133566608)

[3.2.5 Jelenlét összegző 18](#_Toc133566609)

[3.2.6 Alkalmazottak megtekintése 19](#_Toc133566610)

[3.2.7 Fizetés kezelése 20](#_Toc133566611)

[4 Összefoglalás 21](#_Toc133566612)

[5 Irodalomjegyzék 22](#_Toc133566613)

# Bevezetés

## Záródolgozat célja

A záródolgozatom célja, hogy bemutassam a képzés során megismert programozási, program tervezési módszereket és egy jól használható programot készítsek. Családon belül sokszor láttam, hogy mennyire időigényes és bonyolult a papíron való munkaidők feljegyzése és nyomon követése pedig majdnem minden alap feladathoz létezik egy számítógépes alkalmazás, ami vagy megoldja az adott feladatot, vagy a megoldás folyamatában nagy segítséget nyújt. Az egyik legfontosabb alapfeladat, ami minden vállalkozás életében előfordul, az a működése során keletkező dokumentumok tárolása, például a munkaidők nyilvántartása. A munkaidő nyilvántartó alkalmazást kifejezetten a vállalkozás munkaidő nyilvántartásának egyszerűsítésére és hatékonyságának növelésére terveztem elkészíteni. A záródolgozat részletesen leírja a programom működését, az adatok kezelését, emellett tartalmazza a felhasználói felület képernyőképeit és magyarázatot ad a funkciókra. Az program kizárólag a vállalkozó számára érhető el, és segít abban, hogy az ő munkaidejét és a dolgozók munkaidejét is hatékonyan nyilvántartsa. A használata egyszerű és intuitív, hogy még az informatikai ismeretekkel nem rendelkező személy is könnyen használhatja. A program lehetővé teszi a munkanapok felvételét, valamint a dolgozók teljesítményének nyomon követését.

## Köszönetnyilvánítás

Ezúton szeretném megköszönni Boros Sándor és Beke Béla Tanárúrnak, akik nélkül ez a záródolgozat nem jöhetett volna létre.

# Fejlesztői dokumentáció

## A képen szöveg, képernyő, megjelenítés látható Automatikusan generált leírásFejlesztői környezet

A Windows 10 operációs rendszert használtam a záródolgozatom elkészítéséhez mert ez az a rendszer, ami kompatibilis az alábbi programokkal.

A képen szöveg, aláírás, vektorgrafika, clipart látható

Automatikusan generált leírás

A programkódok írására és a design tervezésére a Microsoft Visual Studio fejlesztőkörnyezetét használtam.



A programban lévő képek szerkesztéséhez az Adobe Lightroom alkalmazást használtam. Ez az alkalmazás nagyon sok szerkesztési lehetőséget kínál.



Az adatbázis kezeléséhez, létrehozásához a phpMyAdmin-t használtam.  Ez egy nyílt forrású eszköz, ami képes az egész MySQL szerver menedzselésére az interneten keresztül.



A programom készítése közben a jegyzeteléshez a Microsoft Word szövegszerkesztő szoftvert használtam.

## Az adatbázis

### A képen asztal látható Automatikusan generált leírásAz adatbázis táblái és mezői

A programban lévő adatok tárolására a *munkaidonyilvantarto* adatbázis nyújt lehetőséget.

#### login tábla

A képen szöveg látható

Automatikusan generált leírásA login tábla a belépés adatainak eltárolására szolgál. Mivel a program saját felhasználási célra készült, nincs szükség regisztrációra, a bejelentkezési felületet csak az adatok védelme érdekében hoztam létre. A username a felhasználónév tárolására szolgál rövid szöveg típusban(varchar). A password mező a jelszó tárolására szolgál ez szintén rövid szöveg típusban van megadva.

#### Alkalmazottak\_adatai tábla

Ezt a táblát a programon belül felvett alkalmazottak adatainak a tárolása céljából hoztam létre. Első mezője az **Id** szám(integer) típusú és erre a mezőre AUTO-INCREMENT lett beállítva, tehát önmagától generálódik, ha egy új rekordot rögzítünk. A tábla második mezője a **Nev**, feladata a nevek tárolása, a mező típusa rövid szöveg (varchar). Az alkalmazottak lakhelyének tárolására a **Cím** mező szolgál. A **Nem** mező az alkalmazottak nemét tárolja varchar típusban. A **Pozicio** mező az alkalmazottak pozíció tárolására szolgál szintén rövid szöveg típusban. Az alkalmazottak születési dátumát a **Szuletesidatum** mező tárolja dátum(date) típusban. A következő mező a **Telefonszam** ennek a feladata a telefonszámok tárolása. A **Napifizetese** mező int típusban tárolja a dolgozók napi fizetését. Az utolsó mezőnek kettő értéke lehet , true vagy false.

Mivel az adatbázisban az alkalmazottak adatait hosszú távon terveztem tárolni egyéb okok miatt, ezért egy adott személy teljes kitörlése helyett ennek a mezőnek az értékét false-ra állítva jegyzem fel, hogy a dolgozó már nincs jelen a cégben. Ennek a mezőnek neve **aktiv**, típusa enum(’true’, ’false’).

#### Jelenletiiv tábla

A képen szöveg látható

Automatikusan generált leírásA jelenletiiv tábla célja az alkalmazottak jelenlétének vagy hiányzásainak a letárolása. Ebben a táblában az első mező az **Id** ami integer típusú, ehhez kapcsolati nézetben beállítottam az idegen kulcs megszorításoknál egy fk\_alkalmazott\_jelenlet nevű megszorítási tulajdonságot, amely ON DELETE RESTRICT, ON UPDATE RESTRICT funkcióval rendelkezik és ez az Id mező az alkalmazottak\_adatai táblában lévő Id oszlophoz kapcsolódik. A **datum** mező date típusú dátum tárolására szolgál.Ez az első két mező az Id és a dátum egyedire(unique) van beállítva ami lehetővé teszi hogy egy adott alkalmazottat ne legyen lehetséges ugyanarra a dátumra kétszer letarolni. A **jelenlet** mezőben rövid szöveg típusban lehet tárolni az alkalmazottak adott dátumához tartozó információt (megjelent, hiányzott stb.).

## A Program

### A program alapjai

Az adatbázis elkészítését követően Visual Studio telepítése következett, magának az alkalmazás vázának a létrehozása.

Első feladatom az adatbázis, összekapcsolása és megnyitása a programmal, ez a *MySqlConncetionStringBuilder* függvénnyel történt. A konnektor létrehozásához egy új osztályt hoztam létre Database névvel.

MySqlConnectionStringBuilder builder = new MySqlConnectionStringBuilder();

builder.Server = "localhost";

builder.UserID = "root";

builder.Password = "";

builder.Database = "munkaidonyilvantarto";

conn = new MySqlConnection(builder.ConnectionString);

comm = conn.CreateCommand();

Ezek a sorok tartalmazzák a szerver nevét mellyel az összekapcsolás létrejött és az adatbázis nevét is.

A következő feladatom, hogy létrehozzam és deklaráljam a Form-okat, a program azon részét, amit a felhasználó lát: *FormLogin, FormAlkalmazottak\_kezelese, FormFooldal, FormMegtekintés, FormJelenlétÖsszegző, FormFizetes*

A felsorolt Form-okat a következőképpen hívtam meg:

Login = new FormLogin();

Fooldal = new FormFooldal();

Megtekintés = new FormMegtekintés();

Alkalmazottak\_kezelese=new Alkalmazottak\_kezelese();

JelenletFeljegyzese = new FormJelenletFeljegyzese();

JelenletOsszegzo = new FormJelenlétÖsszegző();

Fizetes = new FormFizetes();

A Program.cs–ben statikus osztályok deklarálása is szükség volt, ez az alábbi programsorokban látható.

static public FormLogin Login = null;

static public Alkalmazottak\_kezelese Alkalmazottak\_kezelese = null;

static public FormFooldal Fooldal = null;

static public FormMegtekintés Megtekintés = null;

static public FormJelenletFeljegyzese JelenletFeljegyzese= null;

static public FormJelenlétÖsszegző JelenletOsszegzo= null;

static public FormFizetes Fizetes = null;

A programban létrehoztam még egy osztályt, ami az Alkalmazottak nevet kapta.

Az Alkalmazottak osztályban létrehozott változók az adatbázison belül az alkalmazottak\_adatai tábla oszlopainak neveinek felel meg. Ezeknek a változóknak értéket adtam és egy publikus Alkalmazott() metódus visszatérési értékeinek állítottam be.

string nev;

int id;

string nem;

string cim;

string pozicio;

DateTime szuletesidatum;

string telefonszam;

int napifizetese;

public string Nev { get => nev; set => nev = value; }

public int Id { get => id; set => id = value; }

public string Nem { get => nem; set => nem = value; }

public string Cim { get => cim; set => cim = value; }

public string Pozicio { get => pozicio; set => pozicio = value; }

public DateTime Szuletesidatum { get => szuletesidatum; set => szuletesidatum = value;}

public string Telefonszam { get => telefonszam; set => telefonszam = value; }

public int Napifizetese { get => napifizetese; set => napifizetese = value; }

Mivel a nevet később szövegesen is szeretném majd megjeleníteni, ezért ebbe az osztályba került egy override metódus is, aminek a visszatérési értéke a név.

### login\_form (bejelentkező felület)

A programot elindítva ezzel az ablakkal találkozik először a felhasználó. Ezen a Form-on történik a beléptetés. Mivel a programot saját célra írtam ezért regisztrációs felületet nem hoztam létre. Ezen a formon megtalálható két textbox a felhasználónév és a jelszó beviteléhez, amelyeket egyértelműen jelzek a felhasználónak. A beléptetés gyakorlatban úgy néz ki, hogy felvettem egy username és egy password változót. Ezeknek a változóknak az értékét az adatbázison belül lévő username és password oszlopokból olvasom ki is adom át. Tehát ez a két változó egyenlő lesz az adatbázisban lévő username és password oszlop értékeivel. Ez, a következő programsorokban látható.

string username = "";

string password = "";

database.comm.CommandText = "SELECT \* FROM login WHERE 1";

database.conn.Open();

MySqlDataReader dr = database.comm.ExecuteReader();

while (dr.Read()) {

username = dr["username"].ToString();

password = dr["password"].ToString();}

A felhasználónév és a jelszó megadása után lehetőség van belépni a programba a ”Belépés” gombbal vagy az enter lenyomásával, ezután az adatbázisból beolvasott sorokat összehasonlítja a program a szövegdobozokba beírt adatokkal egy összetett elágazás segítségével. Helytelen vagy nem megadott felhasználónév vagy jelszó esetén a program hibaüzenettel jelez a felhasználónak. Továbbá létrehoztam egy ”Mégse” gombot amivel ki lehet lépni a programból ez az *Enviroment.Exit* parancs meghívásával történik. Az ablak bezárásakor a program automatikusan kilép. A felhasználónév és a jelszót helyesen megadva és a Belépés gombra kattintva a szövegdobozoka beírt adatok kitörlődnek és a felhasználót a program a *FormFooldal* ablak megnyitásával fogadja.

### fooldal\_form

Sikeres belépés esetén ezzel az ablakkal találkozunk. Ez a Form lesz a Főoldal, amin megtalálható egy kilépés gomb és öt navigációs ikon (Alkalmazottak kezelése, Alkalmazottak megtekintése, Jelenlét feljegyzése, Jelenlét összegző, Fizetés kezelése).

Az ikonokra kattintva a programban megjelenik az adott ablak. Ez a *Window.Show()* metódussal történik. Az adott űrlap megnyitásakor a főoldalt szükség volt bezárni, pontosabban nem láthatóvá tenni ezért ehhez a *this.Hide()* metódust használtam. A kilépés gombra kattintva a program visszadob a bejelentkező felülethez.

### alkalmazottak\_kezelese form

Ezt a Formot megnyitva van lehetőségünk alkalmazottakat felvenni, törölni és szerkeszteni. Itt létrehoztam egy *clear()* nevű függvényt ami arra szolgál hogy a Form bezárásakor a szövegdobozba betöltött, beírt adatok kitörlődjenek és ha egy alkalmazottat kitörlünk vagy átszerkesztünk szintén törölje az adatokat a szövegdobozokból. Ez a következő programsorokkal történik:

textBoxCim**.**Text **=** string**.**Empty**;**

textBoxFizetese**.**Text **=** string**.**Empty**;**

textBoxId**.**Text **=** string**.**Empty**;**

textBoxNev**.**Text **=** string**.**Empty**;**

textBoxPozicio**.**Text **=** string**.**Empty**;**

textBoxTelefonszam**.**Text **=** string**.**Empty**;**

dateTimePicker1**.**Value **=** DateTime**.**Now**;**

comboBoxNem**.**Text **=** string**.**Empty**;**

Az ablakon megtalálható egy *listbox amit listBoxEmberek*-nek *neveztem el,* ebben a szerkeszteni vagy törölni kívánt alkalmazottat tudjuk kiválasztani. Az adatok feltöltése a listboxba egy *listaupdate()* nevű függvénnyel történik az adatbázisból amihez a *MySqlDataReader* osztályt használtam. A *listaupdate()* függvény a Form megnyitásakor fut le. A lista feltöltése következőképpen történik:

database**.**comm**.**CommandText **=** "SELECT `Id`,`Nev`,`Nem`,`Szuletesidatum`,`Telefonszam`,`Napifizetese`,`Cim`,`Pozicio` FROM `alkalmazottak\_adatai` WHERE `aktiv`='true'"**;**

database**.**conn**.**Open**();**

**using** **(**MySqlDataReader dr **=** database**.**comm**.**ExecuteReader**())**

**{**

**while** **(**dr**.**Read**())**

**{**

Alkalmazott uj **=** **new** Alkalmazott**(**dr**.**GetString**(**"Nev"**),** dr**.**GetInt32**(**"Id"**),** dr**.**GetString**(**"Nem"**),** dr**.**GetString**(**"Cim"**),** dr**.**GetString**(**"Pozicio"**),** dr**.**GetDateTime**(**"Szuletesidatum"**),** dr**.**GetString**(**"Telefonszam"**),** dr**.**GetInt32**(**"Napifizetese"**));**

listBoxEmberek**.**Items**.**Add**(**uj**);**

**}}**

**database.conn.Close();**

Az szövegdobozokba az adatok betöltése akkor történik meg ha kijelölünk egy alkalmazottat a listából. A Formon megtalálható 3 gomb az adatok szerkesztéséhez („Hozzáadás”,” Szerkesztés”, ”Törlés”) és egy ”Vissza” gomb.

A **hozzáadás** gomb megnyomását követően a program elküldi az adatbázisba az adatokat, ami úgy történik hogy a *database.comm.CommandText* hívódik meg. Miután a program hozzáadta az adatbázishoz az alkalmazottat azután a program újra meghívja a fentebb említett *listaupdat*e*()* függvény hogy a listában is megjelenjen az előbb felvett alkalmazott. Az adatbázisba való beszúrás az alábbi paranccsal történt:

INSERT INTO `alkalmazottak\_adatai`

(`Id`,`Nev`,`Cim`,`Nem`,`Pozicio`,`Szuletesidatum`,`Telefonszam`,`Napifizetese`) VALUES " **+** "

(Null, @nev, @cim, @nem, @pozicio, @szuletes, @telefonszam, @napifizetes);

A beszúráshoz szükséges paramétereket az ablakon megtalálható szövegdobozokból az veszem ki egyesével és az *AddWithValue* paranccsal adom át:

database**.**comm**.**Parameters**.**AddWithValue**(**"@nev"**,** textBoxNev**.**Text**);**

database**.**comm**.**Parameters**.**AddWithValue**(**"@nem"**,** comboBoxNem**.**Text**);**

database**.**comm**.**Parameters**.**AddWithValue**(**"@cim"**,** textBoxCim**.**Text**);**

stb…

A **szerkesztés** gomb használatához először szükséges kijelölni egy szerkeszteni kívánt alkalmazottat a listából. Arra az esetre hogyha a kijelölés nem történik meg létrehoztam egy elágazást, amelyben a program a *MessageBox.Show()* paranccsal jelzi a felhasználónak a szerkesztés sikertelenségét. A szerkesztéshez a következő kódsort használtam, majd a *listaupdate()* függvényt hívtam meg.

database.comm.CommandText =

"UPDATE `alkalmazottak\_adatai` SET `Nev`=@nev,`Cim`=@cim,`Nem`=@nem,`Pozicio`=@pozicio,`Szuletesidatum`= @szuletes,`Telefonszam`=@telefonszam,`Napifizetese`= @napifizetes

WHERE `Id`=@id";

A paramétereket itt is az *AddWithValue* segítségével adtam át.

Hasonlóan a szerkesztéshez, a **törlés** gomb megnyomása előtt szintén szükség van kijelölni egy alkalmazottat, amit törölni szeretnénk. A törlés az alábbi adatbázis paranccsal történt a programban:

database.comm.CommandText =

"UPDATE `alkalmazottak\_adatai` SET `aktiv`= 'false' WHERE `Id`=@id";

### jelenlet\_feljegyzese form

Ennek a Formnak a feladata, hogy az alkalmazottak munkanapjait feljegyezzük (ki és mikor dolgozott vagy mikor hiányzott). Ezen az ablakon is megtalálható egy listbox amit az előző Formhoz hasonlóan töltöttem fel. A Formon elhelyeztem még egy *DateTimePickert* amiben a dátumot tudjuk kiválasztani továbbá 4 gombot amikkel az adott jelenlét típust tudjuk elmenteni az adatbázisba. A 4 gomb működése lényegében ugyanaz, a megadott dátum és a kiválasztott személy alapján lementeni az adatbázisba az információt. A különbség csak az, hogy az alkalmazott az éppen megjelent, táppénzen volt, igazoltan hiányzott vagy igazolatlanul nem jelent meg. Az adatbázisba való lementéshez a következő programsorokat használtam.

database.comm.CommandText = "INSERT INTO `jelenletiiv` (`Id`, `datum`, `jelenlet`) VALUES ";

string values = "";

foreach (var item in listBoxDolgozok.SelectedItems)

{

Alkalmazott uj = (Alkalmazott)item;

values += (values.Length == 0 ? "" : ",") + $"({uj.Id}, '{dateTimePicker1.Value.ToString("yyyy-MM-dd")}', 'Megjelent')";

}

database.comm.CommandText += values;

database.comm.Parameters.AddWithValue("Datum", dateTimePicker1.Text);

Mivel a *jelenletiiv* táblában a név és a dátum *unique*-ra van állítva ezért a program nem engedi, hogy ugyanazzal a névvel ugyan arra a napra kétszer mentsünk le egy adatot. Emiatt szükség volt létrehozni egy hibakeresést a *try-catch* metódus segítségével. A hibakeresésen belül létrehoztam egy elágazást, amiben a program megpróbálja lementeni az adatot, ha a mentés sikeres akkor egy *MessageBox* segítségével jelzi a felhasználónak a sikerességet. Amennyiben a mentés sikertelen a program egy hibakódot ad vissza, ebben az esetben a program egy felugró ablakban kiírja, hogy sikertelen volt a mentés.

### jelenlet\_osszegzo form

Ennek az ablaknak a feladata, hogy az alkalmazottak munkanapjait jelenítse meg. Ezt megnézhetjük egy megadott hónapra és láthatjuk egy kijelölt dolgozó összes munkanapjait is. Ezen az űrlapon szükség volt létrehozni egy *DateTimePickert* aminek a segítségével meg lehet adni hogy melyik hónap adatait szeretnénk látni. Mivel havi összesítést szeretnénk ezért ennek a *DateTimePickernek* a formátumát beállítottam a *CustomFormat = "yyyy-MM"* paranccsal, hogy csak az évet és a hónapot jelenítse meg. Továbbá beállítottam még egy *ShowUpDown = true* parancsot is, aminek segítségével nyilakkal tudjuk állítani a dátumot. Mivel ez a Form listbox-okon alapul ezért létrehoztam egy listboxot és egy listaupdate() függvényt. Az előző Formhoz hasonlóan, e függvény meghívásakor töltődik fel az alkalmazottak neveivel a lista. A meghívás az ablak megnyitásakor történik. Miután az adatokat feltöltöttük, szükségünk van még 2 listára az egyikben a kiválasztott év és hónap a másikban pedig az összes munkanapot jelenítem meg. Ez a következő programsorral történt:

database.comm.CommandText = "SELECT `datum`,`jelenlet` FROM `jelenletiiv` WHERE `Id`=@id AND MONTH(datum)= @month AND YEAR(datum)= @year;";

database.comm.Parameters.AddWithValue("@id", kivalasztottalkalmazott.Id); database.comm.Parameters.AddWithValue("@month",dateTimePicker.Value.Month);database.comm.Parameters.AddWithValue("@year", dateTimePicker.Value.Year);

database.conn.Open();

using (MySqlDataReader dr = database.comm.ExecuteReader())

{ while (dr.Read())

{

string jelenlet = dr.GetString("jelenlet");

string a = $"{dr.GetDateTime("datum").ToString("yyyy-MM-dd ddd")} {" - " + (jelenlet)}";

listBoxNapok.Items.Add(a);

}

}

listBoxNapokOsszes.Items.Clear();

database.comm.CommandText = "SELECT `datum`,`jelenlet` FROM `jelenletiiv` WHERE `Id`=@id;" ;

database.comm.Parameters.Clear();

database.comm.Parameters.AddWithValue("@id", kivalasztottalkalmazott.Id);

using (MySqlDataReader dr = database.comm.ExecuteReader())

{

while (dr.Read())

{

string jelenlet = dr.GetString("jelenlet");

string a = $"{dr.GetDateTime("datum").ToString("yyyy-MM-dd ddd")} {" - " + (jelenlet)}";

listBoxNapokOsszes.Items.Add(a);

}

}

### alkalmazottak\_megtekintese form

Ez a Form az alkalmazottak adatainak megtekintésére ad lehetőséget. Itt szükség volt létrehozni egy listát amiból ki lehet választani a megtekinteni kívánt alkalmazottat. A működése nagyon hasonlít az alkalmazottak kezelése Form-ra. A listbox itt is a Form megnyitásakor történik, ugyanúgy mint a fentebb említett formoknál, a listaupdate() függvénnyel. Miután kiválasztottuk az adott alkalmazottat, az ablakon felvett textboxokba betöltődnek az adatok (név, cím, telefonszám , stb…).

### fizetes\_kezelese form

Ez a Form a dolgozók fizetését fogja kiszámítani, akár megadott hónapra vagy az egész évre. Itt első sorban létrehoztam egy *DateTimePickert* amivel kitudjuk választani a dátumot. Mivel itt is csak hónapokban vagy években van lehetőség az információt megjeleníteni ezért a formátumot a *CustomFormat = "yyyy-MM"* kódsorral beállítottam és az egyszerűség kedvéért meghívtam a *ShowUpDown = true* parancsot is, hogy nyilakkal legyen lehetőség a dátumot változtatni.

A fizetést a program minden alkalmazottnak az első Formnál megadott („Napi fizetése”) szám alapján fogja kiszámítani. A kiválasztott havi fizetéshez a következő adatbázis lekérdezést hívom meg.

database.comm.CommandText =

"SELECT SUM(`Napifizetese`) AS ertek

FROM `alkalmazottak\_adatai`NATURAL JOIN jelenletiiv

WHERE jelenletiiv.jelenlet='Megjelent'

AND [jelenletiiv.id=@id](mailto:jelenletiiv.id=@id)

AND MONTH(jelenletiiv.datum)=@month

AND YEAR(jelenletiiv.datum)=@year

GROUP BY jelenletiiv.id;";

A szükséges paramétereket a *DateTimePicker*-ből és a listában kiválasztott alkalmazottól adom át. Miután ez megtörtént a *MySqlDataReader* meghívásával kiíratom egy label-be a lekérdezés eredményét.

using(MySqlDataReader dr=database.comm.ExecuteReader())

{

while (dr.Read())

{

labelFizetes.Text = dr.GetInt32("ertek").ToString("#,##0 Ft");

}

}

Az éves fizetés kiíratása ugyanígy működik viszont az adatbázis lekérdezés különbözik:

database.comm.CommandText =

"SELECT SUM(`Napifizetese`) AS ertek

FROM `alkalmazottak\_adatai`NATURAL JOIN jelenletiiv

WHERE jelenletiiv.jelenlet='Megjelent'

AND [jelenletiiv.id=@id](mailto:jelenletiiv.id=@id)

AND YEAR(jelenletiiv.datum)=@year

GROUP BY jelenletiiv.id;

# Felhasználói dokumentáció

## Hardware és software igény

A C# programozási nyelv a .NET Framework-öt használja, ezért szükséges a számítógépen a megfelelő verzió telepítése a szoftver futtatásához, ez legalább egy 4.7.2-es verzió. Továbbá a szoftver adatbázis-szervert használ a felhasználók és a munkaidők nyilvántartására. Ez lehet például a MariaDB. A felsorolt szoftverek futtatásához szükség lesz egy számítógépre, amin Windows 10 operációs rendszer fut, ez lehet asztali számítógép vagy akár laptop. A program elindításához a Munkaido\_nyilvantarto.exe fájlt kell lefuttatni. A programba való belépéshez szükséges felhasználónév: user jelszó: user.

## A program használata

### Bejelentkezés

A képen szöveg látható

Automatikusan generált leírásAz program elindításakor a következő bejelentkező felület lesz a képernyőn.

Bejelentkezési felület

Az alkalmazás használatához elengedhetetlen a belépés biztonsági okok miatt. Ezen az ablakon kell megadni a felhasználónevet és a jelszót. Amennyiben valamelyiket hibásan adjuk meg akkor a program hibaüzenettel jelez, hogy mit rontottunk el. Miután ez megtörtént a belépés gombra kattintva vagy az entert lenyomva beléphetünk a programba. Ha a mégse gombra lépünk vagy bezárjuk a programot akkor az alkalmazás leáll.

### Főoldal

Navigációs felület

Miután sikeres volt a belépés a főoldal fogad minket.

Ebben az ablakban találhatóak meg a főbb navigációs ikonok, amik segítségével tudunk váltani az ablakok között. Ezek a következők:

* **Alkalmazottak kezelése**
* **Jelenlét feljegyzése**
* **Jelenlét összegző**
* **Alkalmazottak megtekintése**

Az alábbiakban bemutatásra kerül minden navigációs ikon feladata és működése. A **”Kilépés”** gombra kattintva vagy az oldal bázárásakor a program a bejelentkező felületre dob vissza.

### Alkalmazottak kezelése

A képen szöveg, képernyőkép, monitor látható

Automatikusan generált leírásAz alkalmazottak kezelésére kattintva tudunk felvenni, törölni és szerkeszteni alkalmazottat. Ez az ablak a következőképpen néz ki.

Alkalmazottak kezelése

Bal oldalon láthatóak az aktív alkalmazottak az ablak jobb oldalán pedig az adatait van lehetőség szerkeszteni. Az azonosító mezőt semelyik esetben nem szükséges kitölteni vagy átírni hiszen ezt a program magától állítja minden egyes dolgozónak.

Ha **hozzáadni** szeretnénk alkalmazottat, akkor az adatok beírása után a hozzáadás gombra kell lépni.

A **szerkesztéshez** ki kell jelölni egy alkalmazottat és a program magától betölti az adatokat a szövegdobozokba, ha nem választunk ki semmit és úgy szeretnénk szerkeszteni akkor az alkalmazás hibaüzenettel fog jelezni.

A **törléshez** szintén szükséges kijelölni egy alkalmazottat és ugyanúgy hibaüzenet jelenik meg ha nincs kijelölés.

Az oldal alján találunk még egy ”**Vissza**” gombot is amire kattintva a program a főoldalra navigál minket.

### Jelenlét feljegyzése

A jelenlét feljegyzése ikonra kattintva feljegyezhetjük az alkalmazottak jelenlétét vagy hiányzását a megadott dátumra.

Jelenlét feljegyzése

Itt lehetőség van egyszerre több alkalmazottat kijelölni a listából. Miután kijelöltünk egy vagy több nevet, a jobb oldalon lévő gombok segítségével tudjuk elmenteni a kijelölt napon dolgozó vagy hiányzó alkalmazottat a lista fölötti dátum megadásával. Hogyha ugyanarra a napra szeretnénk többször elmenteni egy dolgozót, a program nem fogja engedélyezni és hibaüzenettel jelez. Továbbá a dátum megadása mellett találunk még egy ikont is, ami arra szolgál, hogy egy kattintással visszaállítja a dátumot a jelenlegi dátumra. Az ablak alján megtalálható még egy ”Vissza” gomb, erre kattintva a program a főoldalra dob vissza.

### Jelenlét összegző

Ebben az ablakban az előbb bemutatott oldalon lementett munkanapjait tudjuk megtekinteni az alkalmazottaknak havi összegzésben, illetve meg tudjuk nézni az összes napjait is.

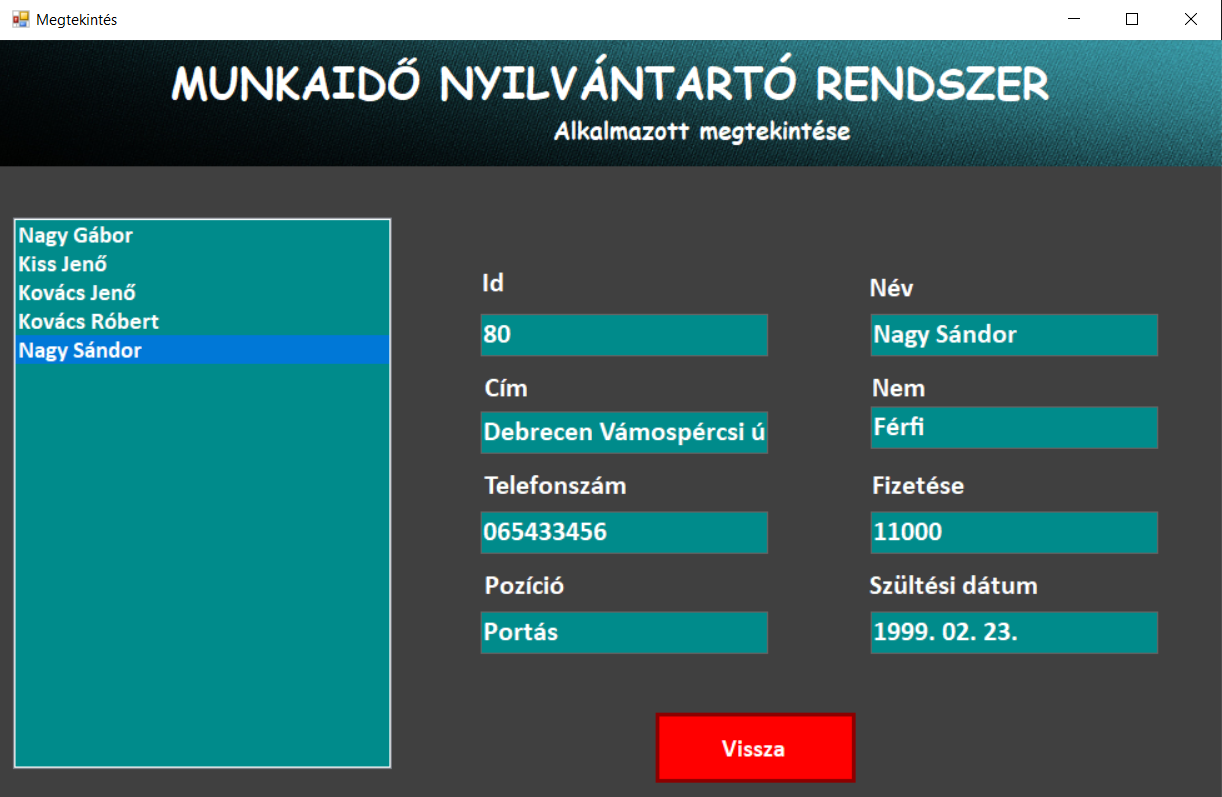
Jelenlét összegző

A képen szöveg látható

Automatikusan generált leírásA képen tér látható

Automatikusan generált leírásA bal oldalon találunk egy listát az alkalmazottak neveivel és egy dátum kiválasztót. A dátum kiválasztásához a jobbra lévő nyilakat lehet használni vagy akár billentyűzet segítségével is be lehet írni a dátumot. Miután kijelölünk egy alkalmazottat a program azonnal kiírja a havi összegzőjét és az összes munkanapját is. Ha dátumot szeretnénk változtatni nem szükséges újra kijelölni valakit mert a program egyből mutatja az eddig kijelölt dolgozó napjait.

### Alkalmazottak megtekintése

Ezen az oldalon van lehetőség megtekinteni az alkalmazottakat és az adatait. A bal oldali listából kiválasztva lehet megtekinteni az alkalmazottak adatait. A vissza gombra kattintva a program a főoldalra navigál minket.

Alkalmazottak megtekintése

### Fizetés kezelése

A képen szöveg, képernyőkép, monitor látható

Automatikusan generált leírásEnnek az ablaknak a feladata, hogy a dolgozók ledogozott napjai alapján havi és évi összegzésből számolja ki a fizetéseiket. A működése ennek is hasonlóan a többihez nagyon egyszerű, csak a megtekinteni kívánt alkalmazottat kell kiválasztani a listából.

# Összefoglalás

A záródolgozat készítésekor a célom az volt, hogy a képzés során megismert programozási technikákat egy saját elképzelés szerint felépített programban bemutassam. A fejlesztés során voltak dolgok, amiket nem tudtam részletesen bemutatni, például a nagyon hosszú kódsorok konkrét feladatait, de rövidebb módon minden kódrészletet és feladatot le tudtam írni. A program készítése közben legnagyobb nehézséget a jelenlétek típusainak feljegyzése jelentette számomra, ez volt az egyik legidőigényesebb része a feladatomnak. Továbbá a design tervezése is egy nehéz feladat volt számomra mert próbáltam minél egyszerűbbre tervezni az alkalmazást. Van néhány elképzelésem még a programmal kapcsolatban, mindenképpen szeretném a továbbiakban kiegészíteni az alkalmazást egy regisztrációs felülettel is mert szeretném a programot használhatóvá tenni az alkalmazottaknak is. Összességében úgy gondolom, hogy sikerült a tervezett alkalmazást elkészíteni. Bízom benne, hogy az általam készített alkalmazás megkönnyíti a mindennapjainkat és hozzájárul a vállalkozásunk hatékonyabb működéséhez.

# Irodalomjegyzék

Reiter István (2018). C# programozás lépésről lépésre

Illés Zoltán (2005). Programozás C# nyelven

[Hatvany Béla Csaba](https://bookline.hu/szerzo/hatvany-bela-csaba/65034?page=1) (2007). MySQL Server adatbázis-programozás .NET környezetben