**Szakképesítés neve:** **737/19T Szoftverfejlesztő képzés**

**OKJ száma:** **54 213 05**

SZAKDOLGOZAT

IMM (INDUSTRY MANUFACTURING MANAGER)

(Fejlesztői dokumentáció)

**Hidvégi Anikó Zeke Róbert**

Témavezető Tanuló

**Győr, 2021.**

Tartalom

[Bevezető 4](#_Toc65862502)

[Informatikai háttér 4](#_Toc65862503)

[Rendszerkövetelmények 4](#_Toc65862504)

[Hardver követelmények (minimum) 4](#_Toc65862505)

[Szoftver követelmények 4](#_Toc65862506)

[A program általános specifikációja 5](#_Toc65862507)

[Termék modul 5](#_Toc65862508)

[Gyártás modul 5](#_Toc65862509)

[Gép modul 5](#_Toc65862510)

[Raktár modul 6](#_Toc65862511)

[Felhasználó modul 6](#_Toc65862512)

[A program telepítése 6](#_Toc65862513)

[A program használatának részletes leírása 9](#_Toc65862514)

[Bejelentkezés 9](#_Toc65862515)

[Adatbázis kapcsolat beállítása 10](#_Toc65862516)

[Fő ablak 10](#_Toc65862517)

[Termék modul 11](#_Toc65862518)

[Minden termék gomb 11](#_Toc65862519)

[Termék kategóriák gomb 13](#_Toc65862520)

[Gép modul 14](#_Toc65862521)

[Gépek gomb 14](#_Toc65862522)

[Gyártás modul 19](#_Toc65862523)

[Munkarendek gomb 19](#_Toc65862524)

[Minden gyártás gomb 23](#_Toc65862525)

[Tovább fejlesztési lehetőségek 23](#_Toc65862526)

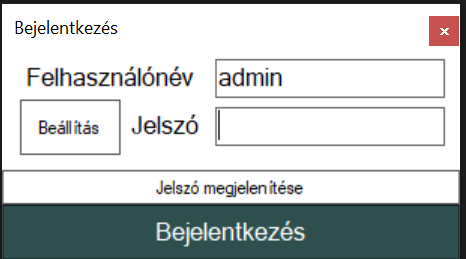
[Program frissítése 23](#_Toc65862527)

## Témaválasztás indoklása

Szerettem volna egy olyan vállalatirányítási rendszert megalkotni, amely figyelembe veszi a felhasználói igényeket. Amelyben egy apró módosításért nem kell a program frissítésére várni, hanem azonnal, akár egy gombnyomással meg lehet oldali. Az ihletet a munkahelyemen használt MES rendszerből szereztem. Összegyűjtöttem az általam ismert operátorok véleményét és próbáltam belefejleszteni az IMM-be. A szoftver fejlesztéséből időhiányban nem megvalósítható hiányosságok, ötletek a „Tovább fejlesztés és ötletek” fejezetben kerülnek kifejtésre. A programot folyamatosan fejlesztem és a fejlesztés követhető és letölthető github-ról.

## Adatbázis kapcsolat és bejelentkezés

A program indulásakor a bejelentkező képernyő fogad minket. Ezen a képernyőn lehet beállítani az adatbázis kapcsolatot is.



A kapcsolat beállítását a „Beállítás” gombra kattintva tehetjük meg. Ezután az alap csatlakozáshoz szükséges paramétert kell megadni, jelenleg egy „Alapértelmezett” gombot lenyomva az alap adatbázis kapcsolati stringet fogja beírni. A mentés gombra kattintva elmenthetjük a beírt csatlakozási stringet a program beállításaiba. Ezek a beállítások felhasználókra vannak szabva, tehát ha egy másik eszközön is futtatni szeretnénk a programot, ezt ott is kötelező elvégezni. Az alap csatlakozási fájl az program indító fájlal egy mappában található „imm.db” néven. A forrásban létrehoztam egy Database osztályt ami a későbbiekben kezeli az adatbázis kapcsolatát.

public class Database {

public static string Connection = Properties.Settings.Default.dbConn;

public static void checkConnectionState(SqlConnection con) {

if (con.State == System.Data.ConnectionState.Open) {

con.Close();

}

}

}

Ennek az osztálynak nincsen konstruktora. Úgy szerettem volna megcsinálni, hogy példányosítás nélkül is ki tudjuk olvasni éppen ezért publikus statikus stringet és metódust hoztam létre. A string felelős az elmentett csatlakozási paraméter kiolvasására. A metódus pedig ellenőrzi, hogy a megadott „SqlConnection” állapota nyitva, ha pedig nyitva van, akkor bezárja a kapcsolatot.

## Az adatbázis táblái

az adatbázis táblái ide wordbe táblázatba.

## Program felépítése (tabPaged, mdiChild)

első képek a programról és az mdiChild megoldásról, ennek a metódusairól, függvényeiről.

## Osztályok és metódusok

Az összes osztály felsorolása és kifejtése.

## Táblázatok és Fa nézetek

DatagridView automatikus frissítése, feltöltése, színezése illetve adatok adatbázisba adása.

## Frissítések

Az elképzelt frissítés kezelés

## Tovább fejlesztés és ötletek

A fejlesztésre nem kerülő ötletek kifejtése