

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Külső Konzulens

Sipos Fanni

NTT DATA Business Solutions Kft.

Belső Konzulens

Dr. Villányi Balázs

ETT Tanszék

Tartalomjegyzék

[1 Bevezetés 3](#_Toc198826015)

[1.1 Feladat bemutatása 3](#_Toc198826016)

[1.2 SAP Business One bemutatása 3](#_Toc198826017)

[2 Fejlesztői környezet 5](#_Toc198826018)

[2.1 Projekt felépítése 5](#_Toc198826019)

[2.2 Futtatás 5](#_Toc198826020)

[3 Megvalósítás 7](#_Toc198826021)

[3.1 Tervezési fázis 7](#_Toc198826022)

[3.1.1 Folyamat lépései 7](#_Toc198826023)

[3.1.2 Követelményelemzés 8](#_Toc198826024)

[3.2 Implementációs fázis 10](#_Toc198826025)

[3.2.1 Szükséges adatbázisszerkezet kialakítása 10](#_Toc198826026)

[3.2.2 Felhasználói mezők felvétele 10](#_Toc198826027)

[3.2.3 Form kialakítása 12](#_Toc198826028)

[3.2.4 Túratervezés menüpont létrehozása 13](#_Toc198826029)

[3.3 Tesztelés 15](#_Toc198826030)

[3.4 Telepítés 16](#_Toc198826031)

[4 Áttekintés 17](#_Toc198826032)

[4.1 Osztályok és leírásuk 17](#_Toc198826033)

[4.1.1 Program osztály 17](#_Toc198826034)

[4.1.2 TripPlanner osztály 18](#_Toc198826035)

[4.1.3 IFormService interfész: 19](#_Toc198826036)

[4.1.4 FormService osztály 19](#_Toc198826037)

[4.2 Bővíthetőség lehetősége 20](#_Toc198826038)

[4.2.1 Crystal Reports alapú nyomtatási funkció 20](#_Toc198826039)

[4.2.2 Vevői rendelések modul kiegészítése 20](#_Toc198826040)

[Irodalomjegyzék 21](#_Toc198826041)

[Függelék 22](#_Toc198826042)

# Bevezetés

## Feladat bemutatása

A tervezett fejlesztés célja, hogy az SAP Business One rendszerén belül egy új, felhasználóbarát funkcióval támogassa a vevői rendelések kiszállításának hatékonyabb szervezését és optimalizálását. A megoldás lehetőséget biztosít arra, hogy a felhasználók figyelembe vegyék a vállalat saját gépjárműflottájának kapacitását, valamint a kiszállítandó termékek összsúlyát, ezáltal megelőzve a túlrakodást és elősegítve a logisztikai folyamatok racionalizálását.

## SAP Business One bemutatása

Az SAP Business One egy integrált vállalatirányítási rendszer (ERP), amelyet kifejezetten kis- és középvállalkozások számára fejlesztett ki az SAP. A rendszer célja, hogy a vállalatok egyetlen, átfogó platformon keresztül kezelhessék alapvető üzleti folyamataikat, mint például a pénzügyek, értékesítés, beszerzés, készletgazdálkodás, gyártás és ügyfélkapcsolat-kezelés. Az SAP Business One lehetővé teszi a valós idejű adatkezelést és jelentéskészítést, így támogatja a gyors és megalapozott döntéshozatalt.

A rendszer egyik legnagyobb előnye az integrált működés, amely biztosítja, hogy minden adat egy közös adatbázisban kerüljön tárolásra. Ezáltal kiküszöbölhetők az adatredundanciák, csökken a hibalehetőség, és nő az átláthatóság. Egy értékesítési folyamat például teljes egészében végigvihető a rendszerben: az ajánlatból rendelés, majd szállítólevél, végül számla és könyvelési tétel generálható, mindezt automatikusan és egymással összefüggésben.

Az SAP Business One rugalmasan testreszabható, így a vállalat saját működési folyamataihoz igazítható. Támogatja az egyedi fejlesztések és kiegészítő modulok (add-on-ok) beépítését, valamint külső rendszerekkel – például logisztikai vagy webáruházi megoldásokkal – való integrációt is. A felhasználói felület könnyen kezelhető, grafikus kialakítása révén intuitív módon segíti a napi munkavégzést, míg az automatizált figyelmeztetések és riportok tovább növelik a hatékonyságot.



. Az SAP Business One széleskörű integrációt biztosít

# Fejlesztői környezet

Az Addon-ok fejlesztését a legegyszerűbb módon a Visual Studio (2019 vagy régebbi) segítségével tudjuk elvégezni. Ezek a verziók tartalmaznak egy minta projektet, amit kiválasztva már megkapjuk a szükséges alapokat. Jelenleg a sablon a .Net Framework 4.7.2-es verzióját használja alapértelmezetten.

Néhány kiegészítés a sablon esetén is szükséges: A „References” fülön található .dll fájlok közé fel kell venni két új fájlt. A SAPbouiCOM, ami az SAP Business One UI API-ja és SAPbobsCOM, ami az SAP Business One DI API-ja. Ezeknek a segítségével tud a program kommunikálni a grafikus felülettel, illetve a kliens mögötti adatbázissal.

## Projekt felépítése

A projektet elkészítve kapunk egy kezdeti főprogramot „Program” néven (main), illetve egy „Menu” névvel ellátott C# fájlt, ami tartalmazza egy üres form és új menüpont felvételét. Formot a projekt során SAP Business One Studio segítségével készült, azonban kódból is elkészíthető ugyanazon form. Ehhez tartalmaz egy form1.b1f nevű fájlt a projekt. Emelett megtalálható a fentebb említett „References” fül, ahol a felhasznált API-okat (.dll fájlok) lehet kezelni, illetve a „Properties” fül, ami maga a projekt felépítését írja le. Itt található az „Assembly.cs”, ahol megadható az elkészített Addon-hoz tartózó név, cégnév, esetleges copyright információk.

## Futtatás

Az első futtatás előtt fontos megjegyezni, hogy az Addon csak akkor tud kapcsolódni a klienshez, ha a Visual Studio és az SAP kliens is adminisztrátori módban fut. Ennek az oka, hogy az Addon telepítése előtt, kizárólag Debug módban futtatható a kód.

Futtatáshoz a fejlesztői környezet – jelen esetben Visual Studio – „Configuration Manager” menüpontja alatt kell felkonfigurálni a futtatás környezetét. Ehhez szükséges beállítani, hogy Debug módban fusson, illetve a „Platform” menüpont alatt az x64-es architektúrát kell beállítani.

A képen diagram, sor, szöveg, Párhuzamos látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

2. Kapcsolat létrehozása az SAP Business One klienssel

# Megvalósítás

## Tervezési fázis

### Folyamat lépései

A tervezési fázis a projekt alapvető szakasza, amely során a megrendelői igények és a funkcionális követelmények alapján meghatározásra kerül a fejlesztési folyamat menete és struktúrája. Ez a fázis kulcsfontosságú annak érdekében, hogy egy világos és strukturált megközelítést alakítsunk ki, amely iránymutatást nyújt a fejlesztés minden lépéséhez, biztosítva a hatékony implementációt és tesztelést.

A képen szöveg, képernyőkép, diagram, Párhuzamos látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

3. Fejlesztési folyamatot ábrázoló diagram

### Követelményelemzés

A folyamat kezdeteként részletes elemzés valósul meg a megrendelői igények azonosítására, amelynek célja a SAP Business One rendszerhez fejlesztett add-on által teljesítendő funkciók és üzleti célok pontos meghatározása. Ez a lépés magába foglalja a funkcionális követelmények összegyűjtését és részletes kidolgozását, amelyek az add-on fejlesztésének alapjául szolgálnak.

A követelményelemzés eredményeként Use Case diagramok készülnek, amelyek az add-on interakcióit és funkcionalitásait szemléltetik.

A képen szöveg, diagram, képernyőkép, sor látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

4. Vevői rendelések menühöz tartozó Use Case diagram

A képen képernyőkép, diagram, sor, Diagram látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

5. A kialakítandó menühöz tartozó Use Case diagram

## Implementációs fázis

### Szükséges adatbázisszerkezet kialakítása

Az Addon-hoz szükséges adatbázis szerkezet kialakítása során felhasználásra kerül az SAP Business One „Vevői rendelések”-hez tartozó rendszertáblája, illetve négy ún. „Felhasználó által definiált tábla”.

|  |  |  |
| --- | --- | --- |
| Tábla neve | Célja | Típusa |
| ORDR | A vevői rendelésekhez tartozó adatok tárolása | Rendszertábla |
| NTT\_TOUR | A Túratervezéshez kapcsolódó adatok tárolása | Felhasználó által definiált tábla |
| NTT\_SZALLKOR | Szállítási körök tárolása | Felhasználó által definiált tábla |
| NTT\_SOFOR | A vállalat sofőrjeinek tárolása | Felhasználó által definiált tábla |
| NTT\_RENDSZAM | A vállalat szállítási járműflottájának tárolása rendszám és kapacitás alapján | Felhasználó által definiált tábla |

### Felhasználói mezők felvétele

Az SAP Business One-ban a felhasználói mezők (User-Defined Fields, UDF) olyan egyedi adatmezők, amelyeket a felhasználók hozhatnak létre a rendszer testreszabásához. Ezek lehetővé teszik további adatok tárolását az üzleti objektumokban (pl. rendelések, számlák, partnerek) anélkül, hogy a rendszer alapvető szerkezetét módosítani kellene. Ezeket a mezőket az **Eszközök > Testreszabási eszközök > Felhasználói mezők létrehozása** menüpont alatt lehet definiálni. Létrehozás során a mező nevének és leírásának megadása mellett a típusát is meg kell adni. Használhatóak az alapértelmezett típusok (Numerikus, Alfanumerikus, Dátum), de a mezők értékei entitáshoz is kapcsolhatóak. A feladat elvégzéséhez 5 UDF mező definiálása szükséges.

|  |  |  |
| --- | --- | --- |
| **Mező neve** | **Rövid leírása** | **Kapcsolt entitás / Típus** |
| Sofőr | A szállítást végző sofőr neve | NTT\_SOFOR |
| Szállítási kör | A szállításhoz tartozó szállítási kör | NTT\_SZALLKOR |
| Rendszám | A szállítást végző jármű rendszáma | NTT\_RENDSZAM |
| Megjegyzés (Cím) | A szállítási címhez tartozó megjegyzés | Alfanumerikus (100) |
| Napok | Szállítási napok | Alfanumerikus (10) |

A felhasználói mezők megjelenítéséhez szükséges bekapcsolni a Nézet > Felhasználói adatok opciót. Miután a mezők rögzítésre kerülnek a Bizonylatok fejléc mezőihez, a Vevői rendelés modult megnyitva láthatóvá válnak a felvett mezők. Az esetleges elírások és hibák elkerülése érdekében az entitáshoz kapcsolt mezőket érdemes felhasználó által definiált lekérdezéssel kiválasztani. Így a felhasználó a mező kitöltésekor egy legördülő listában láthatja az entitásban (táblában) megadott rekordokat és azok közül választhat.

A képen szöveg, képernyőkép, képernyő, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

6. Vevői rendelések modul felhasználói mezői  
Értékadás a lekérdezés segítségével egy legördülő listából

### Form kialakítása

Az SAP Business One-ban a formok a grafikus felhasználói felület részei, amelyek lehetővé teszik az adatok megjelenítését és kezelését. Ezek az űrlapok az SAP B1 Studio segítségével testreszabhatóak, XML-alapú sablonok használatával. A formok támogatják az adatbevitelt, szerkesztést és lekérdezést. Tartalmazhatnak mezőket, gombokat, táblázatokat és egyéb vezérlőelemeket. A túratervezéshez használt form három beviteli mezőt tartalmaz a szűrési feltételek megadásához, egy gombot a szűrés elvégzéséhez és egyet a mentéshez. Emellett egy 13 oszlopból álló táblázatot az adatok megjelenítéséhez. Minden elemhez egy ún. Data Source-t kell rendelni. Ez teszi lehetővé később a mezőkben lévő adatok kódból való elérését, módosítását.

A képen szöveg, képernyőkép, szám, Párhuzamos látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

7. Túratervezéshez használt form

### Túratervezés menüpont létrehozása

#### Menüpont konfigurálása

A menüpont létrehozása a rendszer testreszabási felületén történik. Az SAPbouiCOM API-t használva egy új almenü ("Túratervezés") kerül definiálásra az Értékesítés modulhoz a createMenu metódussal. Ha a menüpont még nem létezik, egy MenuCreationParams objektum segítségével inicializálódik, megadva az egyedi azonosítót (TripPlanner), a nevet ("Túratervezés"), majd a menübe integrálódik.

#### Form betöltése

A menüpont kiválasztásakor a SBO\_Application\_MenuEvent eseménykezelő aktiválódik. Ellenőrzi a kiválasztott menü azonosítóját, majd betölti a az elkészített form-ot a megadott XML fájlból a FormService segítségével. A form betöltése előtt az esetleg nyitva lévő azonos UID-vel rendelkező form bezárásra kerül.

#### Szűrés gomb hatásának kezelése

A szűrés funkció a SBO\_Application\_ItemEvent metódusban valósul meg, az szűrés gomb lenyomására. A gombnyomáskor a form felhasználói mezőiből származó adatok alapján egy SQL lekérdezés (tárolt eljárás) hajtódik végre. A lekérdezés eredménye meghatározza, hogy a form mátrixa frissüljön-e adatokkal, vagy üzenet jelenjen meg, ha nincs találat.

#### Mátrix feltöltése adatokkal

A szűrés eredményeként a form-hoz tartozó ún. DataTable töltődik fel az adatbázisból lekérdezett rekordokkal. Ehhez szükséges, hogy a form létrehozásakor megadott DataSource-höz tartozó egyedi azonosítókat összekössük a lekérdezés eredményként kapott RecordSet egyes oszlopaival. Az algoritmus rekordontként végigmegy az eredményeken egy ciklussal és egyesével tölti fel a táblázat oszlopait, majd lépteti a RecordSet-et. Feltöltés után automatikusan átméretezi az oszlopokat a bennük lévő adatok alapján. Végül a mátrix megjeleníti az adatokat, és az állapotüzenet jelzi a sikeres feltöltést.

#### Adatmódosítás a mátrixban

A felhasználónak lehetősége van bizonyos adatokat módosítani a szűrés után megjelenő táblázatban. A [7. ábrán](#_Form_kialakítása) jól látható, hogy azok az oszlopok módosíthatóak, amelyek háttérszíne fehér. Mivel ezekben az oszlopokban olyan adatok szerepelnek, amelyek egy másik tábla kapcsolásával érhetőek el, ezért a módosítás során az esetleg hibák elkerülése miatt, nem kézzel adja meg a felhasználó az adatokat. Mindhárom oszlopban egy felhasználó által definiált lekérdezéssel érhetőek el a rekordok az oszlophoz tartozó táblából. A módosítani kívánt cellába kattintva egy nagyító jel jelenik meg, melyre kattintva egy felugró ablakban jelennek meg a rekordok.

#### Vevői rendelések form megnyitása

Az adatok sikeres betöltése után minden olyan sorban, ahol találhatóak adatok a legelső „Vevői rendelés” oszlopban egy sárga nyíl jelenik meg. SAP Business One-ban ez az objektum „LinkedObject”-ként definiálandó az XML módosításával. A megjelenítéshez az oszlop típusát módosítjuk az alapértelmezett 16-ról 116-ra. Ezek a típuskódok az SAP által értelmezett kódok. A 16-os kódhoz egyszerű szövegmező tartozik, míg a 116-os kódhoz a LinkedObject típus.

A nyílra kattintva a Vevői rendelések modul azon rekordja töltődik be, amelynek a kulcsa megegyezik az adott sorhoz tartozó kulccsal.

A képen szöveg, elektronika, képernyőkép, képernyő látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

8. Hivatkozás a Vevői rendelések rekordjára

#### Súlykorlátot túllépő rekordok kiemelése

Minden rendszámhoz tartozik egy súlykorlát. Ezt a korlátot vetjük össze betöltés után a túrasúllyal. Abban az esetben, ha a túrasúly túllépi a súlykorlátot a túrához tartozó sorok piros háttérszínt kapnak. Ehhez szükséges a túrasúly megfelelő kiszámítása. A vevői rendelések tábla alapvetően csak a rendelés súlyát tartalmazza. Miután egy rendelést kiegészítettünk a szállítási körrel, rendszámmal és sofőrrel kiszámítható a túrasúly. Ugyanazzal a szállítási nappal, körrel és rendszámmal rendelkező rendelések esetén az egyes rendelések súlyát összeadva kapható meg a túrasúly.

A képen szöveg, képernyőkép, szoftver, Weblap látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

9. Súlykorlátot túllépő rekordok kiemelése

## Tesztelés

Az SAP Business One add-on fejlesztése során a tesztelés kizárólag manuális módon történik, mivel a fejlesztett funkciók közvetlenül a kliens grafikus felületével és az SAP belső API-jaival integrálódnak. Ennek következtében a funkcionális ellenőrzések nem végezhetők el automatizált tesztkeretrendszerek (pl. unit test, integration test) segítségével, így minden újonnan fejlesztett vagy módosított funkció esetén manuális tesztelés szükséges. A manuális tesztelés minden esetben, teszt adatbázisban lévő adatokon, a végleges környezethez hasonló SAP konfigurációban történt.

A manuális tesztelést segíti, hogy a Debug módban futtatott programban, akár futás közben is ún. „Breakpoint”-ok helyezhetőek el. Ha futás közben egy ilyen ponthoz érünk a futás leáll és a felhasználó kezében van az irányítás. A fejlesztői környezetben láthatóak az éppen aktuális változók és értékeik, ezáltal egy esetleges hibás működés esetén végig követhetőek az adatok. Emellett a kliens rendelkezik egy „Státusz bár”-ral. Ez a kliensen belül a képernyő alján elhelyezkedő táblázat. A kódból az API-n keresztül a „SetStatusBarMessage” metódus segítségével könnyedén írathatóak ki változók értékeik, futási eredmények, kivételek, amelyek segítik a hiba okának megtalálását és javítását.

## Telepítés

Az SAP Business One AddOn telepítése során a fejlesztés befejezését követően szükség van egy telepítőcsomag előállítására, amely tartalmazza a futtatható állományokat és a szükséges konfigurációs fájlokat. Ehhez a Visual Studio környezetben a projekt összeállítása („Build”) után létrejön egy kimeneti mappa, amely az AddOn futtatásához szükséges .exe és .dll fájlokat tartalmazza.

A telepítőcsomag előkészítéséhez egy AddOnFiles.xml állomány kerül létrehozásra, amely felsorolja a projekthez tartozó összes fájlt. Ezt követően az SAPB1Installer segítségével elkészül a .zip alapú telepítőcsomag, amely a regisztrációs .ard fájlt is tartalmazza. A telepítés az SAP Business One kliens „AddOn-adminisztráció” menüpontján keresztül történik, ahol a felhasználó a .ard fájl kiválasztásával végrehajthatja az AddOn regisztrációját.

# Áttekintés

## Osztályok és leírásuk

A képen szöveg, képernyőkép, sor, diagram látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

10. UML osztálydiagram

### Program osztály

A Program osztály az alkalmazás belépési pontja, amely inicializálja a TripPlanner objektumot és elindítja az alkalmazást. Ez egy statikus osztály, amely az SAP Business One környezetben futó alkalmazás alapvető indítási logikáját kezeli.

Metódusok

* static void Main(): Létrehozza a TripPlanner objektumot, amely az SAP Business One integrációt és az üzleti logikát kezeli. Elindítja az alkalmazást az Application.Run() metódussal. A program leállítása előtt nullázza a globális oCompany változót.

### TripPlanner osztály

A TripPlanner osztály az alkalmazás magja, amely az SAP Business One integrációt és a túratervezési funkcionalitást valósítja meg. Kezeli az SAP-hoz való kapcsolódást, az egyéni menük létrehozását, az űrlapok betöltését és az adatkezelést.

Attribútumok:

* + static SAPbouiCOM.Application SBO\_Application : az SAP Business One UI API-hoz való kapcsolódást biztosító alkalmazás objektumot tárolja.
* - int[] docEntries : Tömb, amely a adatok mentéséhez szükséges kulcsokat tárolja. Adatok betöltésekor tárolódnak el a kulcsok, majd mentés után törlődnek.

Metódusok:

* + TripPlanner() : Inicializálja az SAP Business One UI és DI API kapcsolatot. Ellenőrzi a kapcsolat sikerességét és feliratkozik a szükséges eseményekre.
* - void createMenu(): Egyéni menüpontot ("Túratervezés") hoz létre az SAP Business One értékesítési moduljában.
* - void SBO\_Application\_MenuEvent: Kezeli a menüpont kiválasztását, betölti a túratervezési űrlapot XML-ből.
* - void SBO\_Application\_ItemEvent: Kezeli az űrlap eseményeit, például gombnyomásokat vagy mátrix linkekre kattintást.
* - void LoadMatrix: Lekérdezi és betölti az adatokat egy mátrixba az adatbázisból egy tárolt eljárás (NTT\_TURATERVEZES) segítségével.
* - void saveMatrix: Menti a mátrix adatait az adatbázisba és frissíti az értékesítési rendelések felhasználói mezőit.
* - void checkIfOverLimit: Ellenőrzi, hogy a túra súlya meghaladja-e a jármű kapacitását, és vizuálisan jelzi a problémás sorokat.
* - String GetCodeFromName: Segédmetódus, amely egy megadott tábla alapján a névhez tartozó kódot adja vissza.

### IFormService interfész:

Az IFormService egy interfész, amely meghatározza az űrlapkezelő osztályok által biztosítandó funkcionalitást. A FormService osztály implementálja ezt az interfészt. Standardizált interfészt biztosít az űrlapkezelő osztályok számára, lehetővé téve a rugalmas és egységes űrlapkezelést.

Attribútumok:

* Stream: Az űrlap XML tartalmát tartalmazó adatfolyam.

Metódusok:

* ExportToString: Visszaadja az űrlap XML tartalmát szövegként, opcionálisan testreszabva a megadott helyettesítésekkel.

### FormService osztály

A FormService osztály egy segédosztály, amely az SAP Business One űrlapok XML alapú betöltését kezeli. Az osztály felelős az űrlap XML tartalmának beolvasásáért és testreszabásáért, mielőtt az SAP környezetbe betöltődne. Megvalósítja az IFormService interfészt.

Attribútumok:

* + Stream Stream: Az űrlap XML tartalmát tartalmazó adatfolyam.

Metódusok:

* + FormSerice(): Egy megadott űrlap XML fájlt (formSRF) tölt be az alkalmazás beágyazott erőforrásaiból.
* + String ExportToString(): Beolvassa az űrlap XML tartalmát, és lehetővé teszi a megadott kulcs-érték párok alapján az XML módosítását (pl. űrlap típus vagy egyedi azonosító cseréje).
* + void Dispose(): Felszabadítja az erőforrásokat, például a StreamReader objektumot.

## Bővíthetőség lehetősége

### Crystal Reports alapú nyomtatási funkció

A Crystal Reports egy hatékony jelentéskészítő szoftver, amelyet az SAP fejlesztett, és lehetővé teszi strukturált adatok – például adatbázisokból vagy alkalmazásokból származó információk – formázott jelentésekbe történő átalakítását, nyomtatását, exportálását és megosztását.

A formot egy egyszerű „Nyomtatás” gombbal kiegészítve, és az ITEM\_EVENT metódusban a kattintását lekezelve meghívható az elkészített jelentés a megfelelő adatokat átadva. Ezáltal készíthető például szállító levél, jelentés a szállítói körökről, a súlykorlátot túllépő rekordokról.

### Vevői rendelések modul kiegészítése

A vevői rendelésekhez tartozó form kiegészíthető lenne egy „Túratervezés” gombbal, amelyre kattintva az éppen aktuális rekord szállítási dátuma, hozzátartozó rendszám vagy szállítási köre alapján a túratervezéshez tartozó form hívódna meg a kívánt szűrési feltételekkel, ezzel javítva a felhasználói élményt és az egyszerűbb kezelést.

A megvalósításhoz a gomb felvétele után az ITEM\_EVENT metódusban a vevői rendelésekhez tartozó form-hoz tartozó ID-t vizsgálva és a gomb kattintását lekezelve a UI API OpenForm() metódusának segítségével megnyitható a túratervezés form-ja és paraméterként átadhatóak az esetleges szűrési feltételek.

Irodalomjegyzék

1. Sap Business One... to Go 9.3, IDE KELL LATOGATASI DATUM<https://help.sap.com/doc/45e5277ba6584a59aa2f2b86969b5535/9.3/en-US/SAP_Business_One_To_Go_Release_9.3.pdf>
2. Data Interface API <https://help.sap.com/doc/saphelpiis_hc_b1_image_repository_development_training_basic_b1_90_tb1300_02_sol_pdf/9.0/en-US/B1_90_TB1300_02_Sol.pdf>
3. The User Interface API  
   <https://help.sap.com/doc/saphelpiis_hc_b1_image_repository_development_training_basic_b1_90_tb1300_03_pdf/9.0/en-US/B1_90_TB1300_03.pdf>

Függelék

SQL tárolt eljárás:

USE [SBODemoHU]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[NTT\_TURATERVEZES]

@rendszam NVARCHAR(15),

@szallKor NVARCHAR(15),

@szallDatum NVARCHAR(10)

AS

BEGIN

SET NOCOUNT ON;

DECLARE @dateParam DATE = NULL;

IF @szallDatum IS NOT NULL AND @szallDatum <> ''

BEGIN TRY

SET @dateParam = CONVERT(DATE, @szallDatum, 102);

END TRY

BEGIN CATCH

THROW 50001, 'Invalid date', 1;

RETURN;

END CATCH;

SELECT

ROW\_NUMBER() OVER(ORDER BY o.DocDueDate DESC) AS order\_id,

o.DocEntry as docEntry,

o.DocDueDate AS del\_date,

r.Name as car\_id,

s.Name AS driver,

k.Name as del\_circle,

o.CardName as order\_name,

o.Address2 AS del\_addr,

o.Weight as order\_kg,

SUM(o.Weight) OVER (

PARTITION BY o.DocDueDate, r.Code, k.Name

) as tour\_kg, -- total tour weight

o.Comments as desc\_ord,

o.U\_desc\_addr as desc\_addr,

o.U\_days as days,

r.U\_kg\_cap as car\_kg

FROM

dbo.ORDR o

LEFT JOIN dbo.[@NTT\_RENDSZAM] r ON o.U\_NTT\_RENDSZAM = r.Code

LEFT JOIN dbo.[@NTT\_SOFOR] s ON o.U\_NTT\_SOFOR = s.Code

LEFT JOIN dbo.[@NTT\_SZALLKOR] k ON o.U\_NTT\_SZALLKOR = k.Code

WHERE

(@rendszam IS NULL OR r.Name = @rendszam)

AND (@szallKor IS NULL OR k.Name = @szallKor)

AND (@dateParam IS NULL OR o.DocDueDate = @dateParam)

GROUP BY

o.DocDueDate, o.CardName, o.GrossBase, o.CardCode, o.DocEntry,

o.Address2, o.Weight, o.Comments, o.U\_desc\_addr,

o.U\_days, r.U\_kg\_cap, r.Name, r.Code, s.Name, k.Name

ORDER BY o.DocDueDate DESC;