

Szakdolgozat

Dézi Géza

Debrecen
2009

Debreceni Egyetem
Informatikai kar

AddOn fejlesztése SAP Business One vállalatirányítási rendszerhez

Témavezető:
Bérczes Tamás
egyetemi tanár

Készítette:
Dézi Géza
programozó matematikus

Debrecen
2009

Tartalomjegyzék

1. Bevezetés.....	4
2. Tervezés – követelmény specifikáció.....	6
2.1. Elvárások.....	6
2.2. Tervezés.....	8
3. Megvalósítás.....	10
3.1. A webes felület	10
3.2. A transzport réteg.....	25
3.3. Az ERP oldal – Sap Business One.....	34
3.4. Összefoglalás.....	40
4. Irodalomjegyzék.....	41
5. Függelék.....	42
5.1. Felhasznált rövidítések.....	42

1.Bevezetés

Az elmúlt évtizedekben egyre nagyobb hangsúlyt kaptak a cégek életében a vállalati erőforrás-tervezést megvalósító komplex vállalati szoftvercsomagok, azaz az ERP rendszerek. Ezen rendszerek segítségével a teljes belső ügyvitelt elektronikus formában lehet megszervezni. Az első ERP rendszerek a gyártási erőforrás-tervező rendszerekből (MRP) alakultak ki, azok kiterjesztéseként. Olyan modulokból épülnek fel, amelyek szoftveres megoldást kínálnak a termelés, a logisztika, az értékesítés, az emberi erőforrás gazdálkodás és a pénzügyi elszámolás tranzakcióinak valós idejű, egységes és integrált kezelésére. Egy jó ERP rendszer lefedi a vállalat teljes hierarchiáját, kellően rugalmas ahhoz hogy alkalmazkodni tudjon a meglévő folyamatokhoz, bármikor kész új modulok, fejlesztések fogadására.

A 90-es évektől egy új irányvonal kezdte el formálni az ERP rendszerek fejlődését (evolúcióját) ez pedig a folyamatszemplélet. A folyamatszemplélet modell szerint, a vállalat értéktermelési folyamata nem ér véget a vállalat határainál, a siker elérésében jelentős szerepe van az ügyfélkörrel való eredményes kommunikációnak és a beszerzési hálózat megfelelő szintű optimalizálásának. Az elektronikus kommunikáció terjedésével az ERP termékeket gyártó cégeknek új kihívásokkal kellett szembesülniük, olyan lehetőségek nyíltak meg, amelyek a korábban szinte elképzelhetetlenek tűntek. Új csatornák nyíltak meg az értékesítésben és a beszerzésben egyaránt, hatékonyabbá vált az üzleti partnerekkel való kapcsolattartás (B2B, Business to Business), a kereskedelmi értékesítés (B2C, Business to Customer) valamint a közigazgatási szférával való információcsere (B2A, Business to Administration).

Az elmúlt néhány év informatikai trendjében az üzleti intelligencia, a workflow rendszerek és az e-Business szerepel az első helyen. Néhány szóban tisztázzuk, mit is jelentenek ezek a fogalmak.

Üzleti intelligencia: a hagyományos ERP rendszerek által generált kimutatások, listák (reportok) ellátják információkkal a vezetőket, azonban nem alkalmasak mélyebb elemzések elkészítésére, összetett döntések támogatására. Ezekre a feladatokra önálló modulok készülnek, amelyek általában nagy mennyiségű adathalmazból különböző algoritmusok alapján nem triviális információkat szűrnék le, amelyek hozzájárulnak a megfelelő döntés

megszületéséhez. Az üzleti intelligencia eszközei: adattárházak, ügyfélkapcsolati rendszerek (CRM), beszállítói hálózat kezelők (SCM), vezetői információs rendszerek (MIS).

Workflow rendszerek: a nagyobb vállalatoknál egy-egy folyamat több emberen fut keresztül. Nem elég tisztázni az egyéni munkaköröket, meg kell adni a lépések időbeli sorrendjét is. A hagyományos ERP rendszer nem látja el feladatokkal a felhasználókat, van lehetőség lekérdezések futtatására, de nincs lehetőség a delegált feladat és felelősség átadásra. A workflow eszközök a meglévő rendszerek „fölé” helyezve automatizálják a folyamatokat, lerövidítik a várakozási időt, szabályozhatóvá és ellenőrizhetővé teszik a feladatok végrehajtását.

e-Business: néhány éve van a köztudatban ez a fogalom, ebbe a körbe tartozik az üzleti célú elektronikus levelezés, a marketing célból készült és interaktív lehetőségeket is biztosító Web oldal üzemeltetése, valamint a teljes vállalati működést vagy annak egy részét elektronikus, ill. Internet alapokra helyező cég is.

A mostani üzleti világban már jóval kevesebb esélye van a fennmaradásra egy olyan cégnek, amely nem használja ki az internetben rejlő lehetőségeket. Ezek a lehetőségek széles skálán mozognak, az egyszerű prezentációs jellegű weboldaltól kezdve az interaktív weboldalakon keresztül egészen az online vásárlást biztosító webáruházakig. Ezeket a lehetőségeket azonban nem olyan egyszerű kiaknázni, egy átlagos kis és középvállalkozás vezetője nem feltétlenül van tisztában a világháló által kínált lehetőségekkel, így erre a területre egy külön szakág specializálódott. Ebben a szakágban olyan cégek és marketing szakemberek dolgoznak, akik egyaránt jártasak az internetes világban és az üzleti életben is. Ezek a szakemberek képesek támogatást nyújtani a megfelelő eszközök kiválasztásában, bevezetésében és hatékony alkalmazásában.

Jelen szakdolgozatban az SAP Business One vállalatirányítási rendszer (későbbiekben SBO) e-Business modullal való kiterjesztését dokumentálom, a követelmények meghatározásától a tervezésen keresztül a megvalósításig. A szakdolgozatban leírtak Szerzői jogi védelem alatt állnak, minden további felhasználásuk az Ifsz Kft hozzájárulásával történhet.

2.Tervezés – követelmény specifikáció

2.1.Elvárások

A megrendelőnek a fejlesztés kezdetén volt egy SBO-s rendszere, valamint egy ettől függetlenül működő Webshopja. Ez a kettős megoldás sok plusz munkát jelentett valamint sok volt a hibalehetőség. Kézenfekvő volt a megoldás, egy olyan webshop kifejlesztése, amely képes automatikusan kommunikálni az SBO-val oda vissza, költséghatékony, tartalmazza a jelenlegi trendeknek megfelelő funkciókat például: PayPal és bankkártyás fizetési lehetőség, sokrétű kedvezményrendszer, dinamikus termékkezelés valamint alkalmazza a korszerű megoldásokat: keresőmotor optimalizálás (SEO), modell-view-controller séma (MVC), alkalmazás interfész használata (API).

Ezeknek a kívánalmaknak a CsCart nevű webshop motor felelt meg leginkább. Ez a Php és MySql alapokon megírt alkalmazás tartalmazta az előírt funkciók nagy részét, szerkezeti megoldásaiban megfelelt a fejlesztők igényeinek, valamint megfelelő dokumentációval és terméktámogatással rendelkezett a továbbfejlesztés támogatásához. A CsCart egy folyamatos fejlesztés alatt álló rendszer, jelenleg a 2.0.9-es verzióval tart. Általánosan elmondható, hogy hatékony a fejlesztők és a vevők közötti kommunikáció (fórum, help desk), így a frissítések alkalmával olyan új modulok és funkciók kerülnek be az alaprendszerbe, amelyekre a legnagyobb igény mutatkozik. Ezen kívül az alaprendszer kellően logikusan van felépítve ahhoz, hogy jelentősebb változtatás nélkül új funkcionalitással lehessen kiegészíteni. Ez a fejleszthetőség komoly előnyt jelent a webes alkalmazások piacán, hiszen így a megvásárolt terméket saját fejlesztőgárdával bármikor lehet módosítani.

Az ERP oldalról az SBO volt a továbbfejlesztés alapja. Az SBO kifejezetten kis- és középvállalatoknak fejlesztett egyszerveres vállalatirányítási rendszer. Az SAP ezen termékét 2004-ben vezette be a magyar piacra, ezáltal is erősítve a pozícióját a kis- és középvállalati szegmensben. Az SBO világszerte egyre több ügyfelet érint, egy 2008-as kimutatás szerint a 47 800 vállalatot számláló SAP ügyfélbázisból több, mint 35 700 szervezetet jelent. Íme egy leírás a SAP magyarországi weboldaláról, mire tervezték a Business One rendszert:

„A már stabilizálódott, de nagyobb célok felé törő néhány százmillió és pár milliárd forint közötti éves árbevételű vállalkozások számára az alapvető üzleti folyamatok optimalizálása jelenti a legfontosabb feladatot. Ez olyan komplex vállalatirányítási szoftvert kíván, amely gyorsan bevezethető, egyszerűen használható és hatékonyan használja ki a meglévő informatikai infrastruktúrát. Mindeközben a programnak megfizethetőnek is kell lennie, hogy megvásárlása ésszerű beruházásnak minősüljön.

Mindez együtt az SAP Business One. Az összetartó erő.,,

forrás: <http://www.sap.com/hungary>

A cég filozófiája szerint a Business One egy gyorsan bevezethető, felhasználóbarát és stabil rendszer, amely tartalmazza azokat az alapmodulokat, amelyekre egy vállalatnak szüksége lehet. Mivel nem lehet minden igényre felkészíteni egy rendszert, az SAP lehetőséget biztosít a bővítésre, az egyedi igények megvalósítására:

„...Bizonyos esetekben előfordulhat, hogy a rendszer alapfunkcionalitásán túlmutató bővítések alkalmazása szükséges. Ezek megvalósításához a rendszer tartalmaz egy integrációs eszközt, mely segítségével felület, adatbázis, illetve üzleti logika szinten integrált tetszőleges alkalmazások készíthetők az SAP Business One rendszerhez. „

forrás: <http://www.sap.com/hungary>

2.2. Tervezés

A tervezés első kérdése az alkalmazásokat kiszolgáló gép vagy gépek kérdése volt. Mivel az SBO önálló zárt szerveren fut, így a webes igények kiszolgálására egy önálló webszervert kellett üzembe helyezni. Mivel az SBO szervere nem elérhető a webszerről, így biztonságosan üzemeltethető, még akkor is, ha a webszervert külső támadás éri. A webszerveren egy Apache kiszolgáló látja el a beérkező igényeket. Az Apache http szerver egy nyílt forráskódú webkiszolgáló alkalmazás, amely nagy szerepet játszott a World Wide Web (WWW) elterjedésében. Robusztus, erőteljes és rugalmas webszerver, amely kompatibilis a HTTP/1.1 (RFC2616) protokollal. Egy 2008-as felmérés szerint a világ webhelyeinek 50,93%-át Apache szerver szolgálja ki.

Forrás: <http://news.netcraft.com/>

A második lényeges megszorítás az adatáramlás iránya volt. Mivel egy publikus weboldalról van szó, így a biztonsági kockázat is jóval nagyobb, mint egy belső hálózat esetén. Habár a korszerű biztonsági eljárások a betörések nagy része ellen védelmet nyújtanak (pl. SQL injection), mégis korlátozni kellett az adatok mozgásának irányát, azaz a webes felület nem kezdeményezhet adatcserét az SBO adatbázisával, csak az SBO képes olvasni és írni a CsCart alatt futó MySQL adatbázisban lévő adattáblákat. Így egy esetleges betörés esetén nem lehet olyan PHP szkripteket futtatni, amelyek bizalmas céges adatokhoz férnének hozzá, csak a MySQL adattáblákban lévő rekordokhoz férhet hozzá, amely szintén nem szerencsés, de kevesebb kárt okozhat mint az SBO törzstábláinak olvasása / felülírása. Szintén biztonsági okok miatt a webszerver szolgáltatásai közül az FTP protokollt használatát mellőztük, habár a fejlesztés menetét jelentősen egyszerűbbé tette volna. Helyette egy SSH csatornán keresztül Távoli asztallal kapcsolódtunk a szerverhez. A dolgozat több pontján kiemelésre került, hogy a fejlesztés során számtalan nyílt forráskódú alkalmazást és technológiát használtunk fel, az SSH is ebbe a csoportba sorolható. Segítségével több biztonsági rést sikerült megszüntetnünk,

mivel minden hálózati adatforgalmat kódol. Az SSH kapcsolathoz a copssh nevű ingyenes szoftvert alkalmaztuk.

Az előkészületek során derült fény egy újabb lényeges tényre: a CsCart és az SBO alapvetően más hierarchiába szervezi a termékek halmazát. A webes felületen egy többrétegű fa struktúra tartalmazza a termékek adatait, ez a szerkezet nagy mennyiségű adat esetén jelentősen megkönnyíti a termékek kezelését és a rendszerezését. Ellentétben az SBO-val, ahol a termékek csoportosítására szolgáló cikkcsoport nem fa struktúrájú. Egy másik lényeges eltérés, hogy az ügyfél a webshopban lényegesen több cikket kíván kezelni, mint az SBO-ban, ugyanis a webes felület egyfajta katalógusként működik. Minden termék szerepel benne amire rendelés érkezik, azonban az SBO-ba csak olyan termékek kerülnek be, amelyekre létezik tényleges rendelés a webshopban. Így a felhasználónak nem kell a CsCart-ba és az SBO-ba külön-külön feltölteni egy adott terméket, elegendő csak a webre feltenni, ha megrendelés érkezik rá, az automatikusan átkerül az SBO adatbázisába. Így az SBO termék adatbázisa jóval kisebb mint ami a weben megjelenik, ezáltal a napi munka során csak olyan termékekkel kerül kapcsolatba az ügyintéző amire valóban szüksége is van.

3.Megvalósítás

3.1.A webes felület

A fejlesztés internetes oldalának az alapját a CsCart adta. Mivel ennek a rendszernek a fejlesztése egy „élő” folyamat, így a benne lévő megoldások és lehetőségek a vásárlók igényeit követik. Minél több a vásárló, annál több igény jelentkezik a szoftver felé, így a vevők olyan funkciókat is megkaphatnak, amelyekre esetlegesen nem is gondoltak a vásárlás előtt, azonban korábbi igények alapján már bekerült a rendszerbe. Ilyen lehetőségek például a termékhez kapcsolt tulajdonságok és opciók kezelése, a bankkártyás és PayPal fizetési lehetőség biztosítása, a beépített captcha kódok a beléptetésnél, termékek importálása és exportálása xls vagy csv formátummal.

Opciók

A példa kedvéért: az adatbázisban van egy termék: az alma. Almából van zöld alma, piros alma és sárga alma. Kérdés, hogy ez hogy jelenjen meg a weboldalon. Az SBO adatbázisában egyértelmű volt a helyzet, minden almatípus külön-külön termékként szerepel egy adott cikksorozatban. A CsCart azonban ezt alapértelmezésként egy termékként tárolta, mellette feltüntetve a szín opciót (Options). Ez az opció egy legördülő listában jelent meg a termék adatlapján egy értéklistával és a rendeléskor ebből az értéklistából választotta ki az adott színvariáns kódját és ezt küldte tovább a termék azonosítója mellett. Ez a megoldás bizonyos esetekben logikus és jó, viszont ilyen struktúra esetén nem kezelhető a tételes, szín alapú raktárkészlet, valamint nehézséget okozott volna az SBO-val való összekapcsolása, webes oldalról egy termékazonosító + színvariáns kombinációt kellett volna leképezni egy SBO-s termékazonosítóra. Egy másik nagy hiányossága, hogy az opciókra a CsCart-ban nem lehetett szűrőket beállítani, azonban a tulajdonságoknál megvolt ez a lehetőség.

Tulajdonságok

Az tulajdonságoknál (Features) egy-egy kategóriához tulajdonságokat és ezekhez kapcsolódóan értéklistán rendel a CsCart. Például tekintsük az almák kategóriát. Ezen belül van három termék, zöld alma, piros alma és sárga alma. A kategóriához hozzárendeljük a szín tulajdonságot, három lehetséges értékkel: zöld, piros, sárga. Majd a termékeknél beállítjuk hogy melyik szín tartozik az adott termékhez. Természetesen egy adott színnel több termék is szerepelhet, például több fajta piros almánk van raktáron. Ez egy-két tulajdonságnál feleslegesnek tűnhet, azonban mikor egy termékhez 5-10 ilyen érték tartozik, már kifizetődő a használata. A másik nagy előnye, hogy a tulajdonságokra lehet szűrőket definiálni, így a vásárlás során a vevőnek meg lehet jeleníteni, hogy az alma cikkesoportban piros színű termékből van 5 darab, zöldből 2 és így tovább. Így ha a vevőt csak a piros alma érdekli, egyetlen kattintással listázhatja a termékeket (1. ábra). Valamint a másik nagy előnye, hogy így minden termék variáns külön-külön fog szerepelni az adatbázisban, egyedi azonosítóval, így az SBO-val való megfeleltetés is egyszerű és gyors lesz.

The screenshot shows a search results page for tiles. On the left, there is a navigation menu with categories like 'CSAPTELEPEK', 'FÜRDŐSZOBABÚTOROK', etc. Below the menu are sections for 'Vásárlási opciók' (Purchase options) showing a price range from 0 Ft to 250,000 Ft, and 'Információ' (Information) with links for 'Cégünkről' and 'Kapcsolat'. The main content area displays search results for 'Ares' tiles. The first result is 'Ares cemento 33 x 60 cm' with a price of 10 Ft and 'Nincs raktárkészleten'. The second is 'Ares pizarra 33 x 60 cm' with a price of [Lépjen kapcsolatba velünk az árért] and '1 cikk(ek)'. The third is 'Ares strattos beige 33 x 60 cm' with a price of [Lépjen kapcsolatba velünk az árért]. Each result includes a small image of the tile, a 'Részletek megtekintése' link, and buttons for 'KOSÁRBA HELYEZÉS' and 'Hozzáadás a kívánságlistához'.

1. ábra: szűkítés tulajdonságok alapján

Tulajdonságok megjelenítése


Alapértelmezésként a CsCart Tulajdonságokat a termék adatai között, felsorolásszerűen helyezte el „/” jellel elválasztva. Ez a megoldás egy nehezen átlátható összképet adott a kiválasztott kategória megjelenítésénél. Emiatt szükséges volt átalakítani a megjelenítést úgy, hogy az összkép átláthatóbb legyen, azonban az ügyfél számára a szükséges információk mégis elérhetőek maradjanak. Erre a célra egy egyszerű, de annál látványosabb Javascript kódot illesztettünk a CsCart JS keretrendszerébe. Ez a szkript Html kódokat képes megjeleníteni szövegbuborék formában (2. ábra). A háttérszíntől kezdve a megjelenési sebességig minden tulajdonságot be lehet állítani és a végeredmény tökéletesen illeszkedett a weblap arculatába.

Kiváló minőségű burkoló anyagok Olaszországból és Magyarországról

[Nagyobb kép megnévezése](#)

Család: Méret:

Szín:




[Ares beige 33 x 60 cm](#)

Termék tulajdonságai

Ár: 800 Ft

this product is electronically distributed

[KOSÁRBA HELYEZÉS](#) [Hozzáadás a kívánságlistához](#)





[Ares strattos beige 33 x 60 cm](#)

Termék tulajdonságai

[Lépjen kapcsolatba velünk az árért]

Raktáron: **1 cikk(ek)**

[Hozzáadás a kívánságlistához](#)

2. ábra: tulajdonságok listázása szövegbuborékban

Szűrés a Tulajdonságokra

A Tulajdonságok egyik legnagyobb fejlesztési kérdése a szűrés és csoportosítás volt. Adott egy cikkcsoport, ahol van 50 termék, 10féle tulajdonságtípus, mindegyik tulajdonságnál 3-4 különböző lehetséges értékkel. A vevő csak azokat a termékeket szeretné látni, amelyek megfelelnek az igényeinek, például első lépésben csak a magyar származású almák, azután a piros almák, majd végül azok, amelyeket nem kell hűvös helyen tartani. Ezt a szűrési folyamatot legördülő listákon keresztül kellett megvalósítani úgy, hogy minden lépés után a listák értéke csak olyan elemeket tartalmazzon, amelyek ténylegesen előfordulnak valamely terméknél. Ez az igény statikus adatoknál nem jelentett volna különösebb nehézséget, hisz ilyen szűrést statikus adatokkal számtalan helyen megvalósítottak már, elég csak a cím

szerinti szűréseknél az Ország – Megye – Város kiválasztó megoldásokat megfigyelni. Ide azonban egy olyan dinamikus módszer kellett, amely egy adott kategóriánál összegyűjti az összes lehetséges tulajdonság típust valamint az ezekhez kapcsolódó összes valóságban előforduló értéket. Ezen információk alapján szövegesen összeállít olyan Javascript tömböket, amelyeket a sablonkezelőn keresztül el lehet juttatni a kliens gépére, így a kliens gépén minden információ adott lesz a gyors szűréshez. Az információkat legördülő listákra képeztük le, majd a működés közben a listákat módosítottuk az aktuális kiválasztott értékek alapján. Először is a kontrollerben összegyűjtöttük az előforduló variánsokat egy tömbbe, egy másik tömbbe pedig az egyes termékek tulajdonság maszkját. Ezek után létrehoztunk egy újabb tömböt ami a szűrő aktuális értékeit tartalmazza. Ezekből a PHP-s tömbökből generáltunk Javascript tömböket, valamint ezekből generálta ki a sablonkezelő a SELECT tagokat. A futás során a SELECT tagok onchange eseményére be volt állítva egy függvény, ami az aktuálisan kiválasztott érték alapján beállította a szűrő tömböt, majd ezek után végigfutott a termékek tulajdonság maszkját tartalmazó tömbön, és ahol a szűrő értékétől eltérő elemet talált, azt megjelölte mint nem megjelenítendő elem. A függvény végén átkerült a vezérlés egy másik, AJAX hívást tartalmazó függvényhez, amely ellenőrizte, hogy mely termékeket kell megjeleníteni, majd ezeket elküldte egy háttérben futó függvényhez, amely kirajzolta a kívánt termékeket. Miután a szűrés után megjelentek a kért termékek, a többi legördülő lista értékkészletét úgy módosítottuk, hogy csak azok az értékek maradjanak benne, amelyek szerepelnek valamelyik megjelenített termék tulajdonság maszkjában (3. ábra). Kellott egy visszacsatolás is, mivel a szűrés során a már beállított értékeket tartalmazó SELECT tagokat nem módosítottuk, így előfordulhatott az az eset, amikor olyan szűrési feltételeket határozott meg a vevő, amelyre nincsenek megfelelő termékek. Ilyenkor a szűrési feltételeket lenulláztuk, az utolsó feltétel kivételével, majd az újonnan előállt szűrő tömb alapján jelenítettük meg az elemeket.

Csempék



Kiváló minőségű burkoló anyagok Olaszországból és Magyarországról

[Nagyobb kép megnévezése](#)

Méret: 33 * 60

Termékosalád: Kérem válasszon



[Ares beige 33 x 60 cm](#)

[Termék tulajdonságai](#)

Ár: 800 Ft

this product is electronically distributed

KOSÁRBA HELYEZÉS

[Hozzáadás a kívánságlistához](#)

Szín: Kérem válasszon

Kérem válasszon
beige
blanco
cemento
pizarra



[Ares strattos pizarra 33 x 60 cm](#)

[Termék tulajdonságai](#)

Ár: 10 Ft

Raktáron:

Nincs raktárkészleten

KOSÁRBA HELYEZÉS

[Hozzáadás a kívánságlistához](#)

3. ábra: dinamikus legördülő listák

Rendelések és számlák kezelése

A megrendelések kezelése gyakorlatilag a két összekapcsolt rendszer egyik legkritikusabb eleme volt. A követelmények meghatározásánál az egyik legfontosabb elem az volt, hogy a weboldalon leadott rendelések automatikusan kerüljenek át az SBO-ba elkerülve a többszöri adatrögzítést. Egy jól felépített és kényelmes weboldal jelentős mértékben növelheti a megrendelések számát. Mindkét rendszernek – CsCart és SBO – megvolt a saját megrendeléskezelő metódusa, ezt a két különböző rendszert kellett összekapcsolni és kiegészíteni az SBO-ban elkészült számlák megtekintésének lehetőségével a weboldalon. A megrendelések törzsadatait a weboldal hozta létre, majd a feldolgozást a SBO végezte el. A megrendelések adatbázisok közötti transzferjét egy automatikus folyamat látja el, amely minden 10 percben ellenőrzi hogy van e olyan megrendelés a webes adatbázisban, amely még nem került át az SBO rendszerébe, amennyiben talál ilyen rekordokat, azokból összeállít egy SBO-s megrendelést. Az érdekesebb része ezután következik, mivel a megrendelőnek vannak nagykereskedelmi partnerei, akiknél a vásárlás több száz tételt jelenthet, előfordulhat, hogy egy megrendelést készlethiány miatt nem egyszerre, hanem több körben szállítanak ki a megrendelőnek. Ilyenkor a weboldalon látnia kell a megrendelőnek, hogy az aktuális rendelésekből mennyi van teljesítve és mennyi a nyitott mennyiség, ami még szállításra vár. Az is előfordulhat, hogy a megrendelt tételeket utólag az SBO rendszerében az ügyintéző módosítja vagy törli. A szállítással párhuzamosan készülhetnek számlák is, amelyek egy adott rendelésből csak bizonyos tételeket tartalmaznak, illetve több rendelés tételeit fogják össze. Összefoglalva: az SBO-ban egy rendeléshez több számla is tartozhat és egy számlán több rendelésből is szerepelhetnek tételek, valamint egy webes megrendelés és a hozzá tartozó SBO-s megrendelés nem feltétlenül ugyanazokat a tételeket tartalmazza. Ezeket az információkat úgy kellett megjeleníteni a vevőknek, hogy bármikor egyértelműen el tudja dönteni, hogy mi történt az általa leadott megrendeléssel, mikor és milyen úton fizette ki a megrendelt tételeket. Erre a feladatra két menüpont készült el a weboldalon, az egyik, a számlákat tartalmazza táblázatos formában a másik leadott megrendeléseket a 4-es ábrán látható módon.

▲ Keresés

Összesen (Ft): -

Rendelés státusz:

Beszerzés alatt
 Lezárt
 Elutasított
 Sikertelen
 Visszavont
 Nyitott
 Feldolgozás alatt
 Kiszállítás alatt

Periódus: ▼

Adja meg a dátumot: -

Rendelés azonosító:

SBO kód:

KERESÉS

Oldalak kezelése: **1** 2 3 4

Azonosító	SBO kód	Státusz	Ügyfél	Dátum ↓	Összesen	Számlák
#36	#91000708	Nyitott	Készpénzes vevő	2009-11-19	775 Ft	
#41	#91000713	Nyitott	Készpénzes vevő	2009-11-19	1.150 Ft	
#37	#91000709	Nyitott	Készpénzes vevő	2009-11-19	1.163 Ft	
#33	#91000705	Nyitott	Készpénzes vevő	2009-11-19	1.413 Ft	
#42	#91000714	Nyitott	Készpénzes vevő	2009-11-19	1.150 Ft	
#38	#91000710	Nyitott	Készpénzes vevő	2009-11-19	1.163 Ft	
#34	#91000706	Nyitott	Készpénzes vevő	2009-11-19	1.413 Ft	
#39	#91000711	Nyitott	Készpénzes vevő	2009-11-19	125 Ft	

4. ábra megrendelések listázása

A vevő itt megtalálja a webes azonosítóját a megrendelésének, rákattintva egy részletező oldalon megnézheti pontosan mit rendelt, milyen dátummal és összeggel. Az SBO azonosító oszlopban a megrendeléshez tartozó SBO számot látja, rákattintva az SBO-ban szereplő tételeket találja meg, amelyek eltérhetnek a weben leadott listától. Látja a rendelés státuszát, dátumát, teljes összegét valamint egy értéklistát, hogy az adott rendelés tételeit mely számlákon találja meg. A táblázat fejlécében szereplő cellák rendezési feltételként működnek,

rákattintva a rekordokat dátum, végösszeg vagy azonosító szerint rendezheti az ügyfél csökkenő vagy növekvő sorrendben. A számlák oszlopban lévő számlaszámok szintén linkként működnek, rákattintva megtekintheti hogy az adott számla milyen tételeket tartalmaz. Mind a rendelés, mind a számla részletes megjelenítésénél szerepelnek az adott tételhez köthető számlák vagy megrendelések.

Ahhoz, hogy ezt az összetett kapcsolatot létrehozzuk, a megrendelés és a számla rekordjait tétel szinten kellett összekapcsolni. A rendelések felől nézve:

Minden rendelésnek van egy fejléce, a fejléchez külső kulccsal hozzá vannak kötve a rendelési tételek, valamint a rendelési tételek külső kulccsal hozzá vannak kötve a számla tételeihez. A számla tételei a megrendelésekhez hasonlóan egy fejléchez vannak kötve. A megrendelés és a számla fejléce tartalmazza a törzsadatokat, a tételek pedig a részletes adatokat. Ezzel a külső kulcsolási megoldással elérhetővé vált, hogy a megrendelő egy megrendeléshez tetszőleges számú számlát állítson ki, valamint hogy egy számla több megrendelésből is tartalmazzon elemet.

A megrendelések listázásánál szembesültünk egy újabb nehézséggel. Egy adott vevőnek több fajta megrendelése is lehet. Lehet hogy egy megrendelés csak a webes adatbázisban szerepel (SBO-ba még nem került át), lehet hogy mindkét adatbázisban létezik (SBO-ban feldolgozás alatt) vagy éppenséggel csak az SBO adatbázisában létezik (a vevő mondjuk faxon adta le a rendelést, és az lett rögzítve az SBO-ban). Természetesen ha egy vevőnek van egy ilyen lehetősége, akkor minden rendelését szeretné egy helyen látni, függetlenül attól, hogy milyen módon juttatta el a partneréhez. A programkód szempontjából ez azt jelentette, hogy a táblázathoz szükséges adatok két különböző táblában vannak elszórva. Az egyik tábla a cscart_orders, ahol a weboldal tárolja a saját megrendeléseit, valamint az ifsz_megrendelesek ahol pedig az SBO-ból érkező rekordokat tároljuk. Ahhoz, hogy a rendezés és a keresés megfelelően hatékony legyen, célszerű azt az adatbázis-kezelő rendszerre bízni. Erre a célra a MySql 5.0.1-es verziója már tartalmazza a nézettáblák kezelésének lehetőségét. Készítettünk egy olyan nézettáblát, amely mindkét forrás törzsadatait tartalmazza, és a MySql kezelni tudja

rajta a rendezéseket és a keresést.

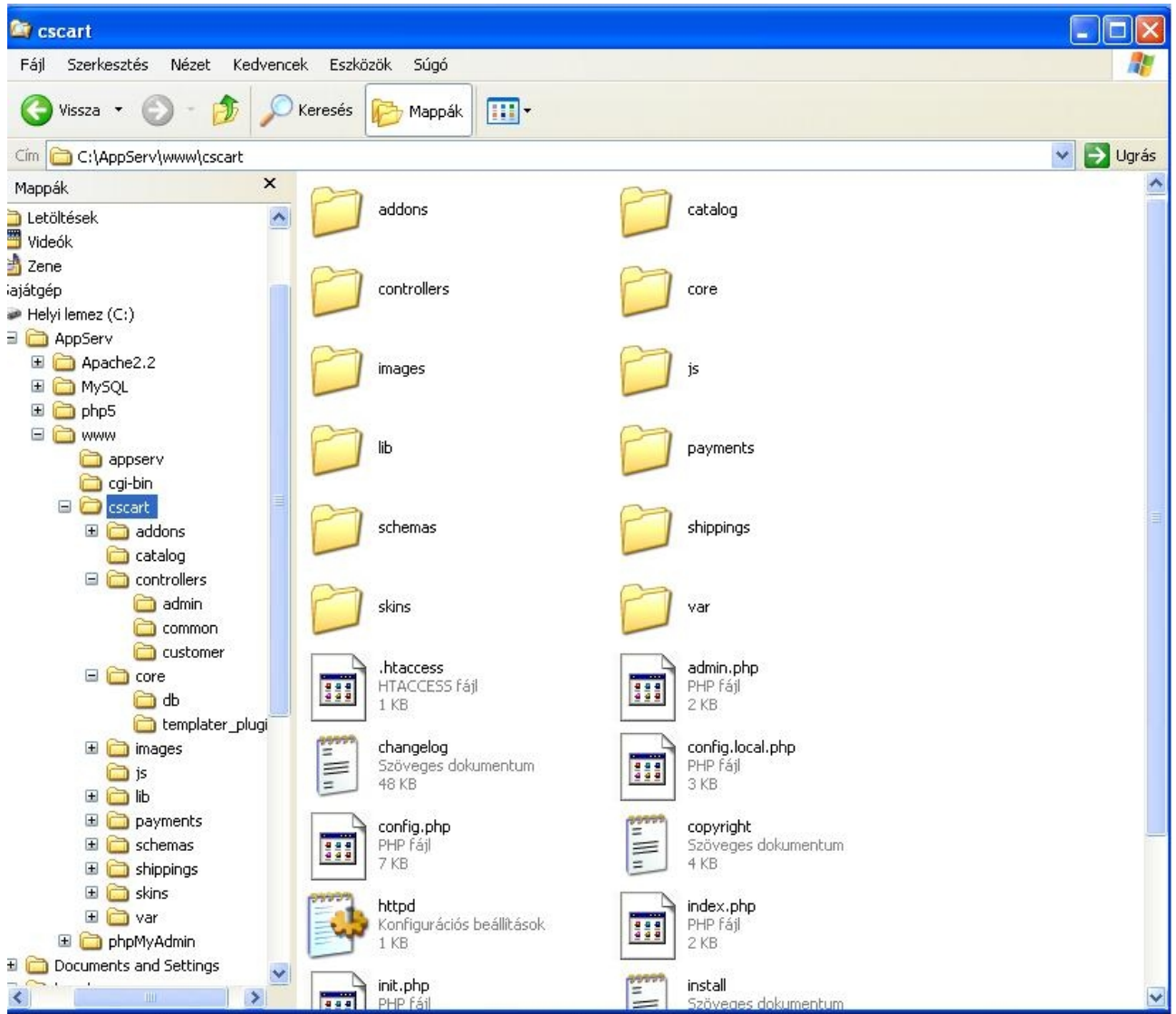
```
CREATE IFSZ_ORDERS_VIEW as
SELECT
cs.order_id, cs.user_id, u.SBO_ID, cs.status,
date_format( from_unixtime( `cs`.`timestamp` ) ,
_utf8 '%Y-%m-%e' ) AS datum, cs.total,
concat_ws( _utf8 ' ', `cs`.`firstname` , `cs`.`lastname` ) AS `cardname` ,
ifsz.docnum, ifsz.docentry, " AS WebMrs
FROM cscart_orders cs
LEFT JOIN cscart_users u ON cs.user_id = u.user_id
LEFT JOIN ifsz_megrendelesek AS ifsz ON cs.order_id = ifsz.WebMrs
UNION ALL
SELECT " , " , ifszm.cardcode, ifszm.DocStatus,ifszm.konyvelesi_datum, ifszm.Osszesen,
ifszm.cardname, ifszm.docnum, ifszm.docentry, ifszm.WebMrs
FROM ifsz_megrendelesek ifszm
WHERE (isnull( `WebMrs` )OR ( `WebMrs` = ""))
```

Általánosan elmondható, hogy az ingyenes megoldásokat készítő cégek igyekeznek tartani a versenyt a konkurenciával. A MySQL fejlesztői egyre több olyan eszközt építenek be, amely valamelyik fizetős adatbázis-kezelő rendszerben (Pl. Oracle vagy MSSQL) már bevált. Habár ezek az eszközök nem feltétlenül működnek annyira hatékonyan mint a fizetős megfelelőjük, illetve a dokumentációk sok esetben tartalmazzák a „...tudunk a hibáról és dolgozunk a megoldásán...” megjegyzéseket, hosszú távon az látszik, hogy az nyílt forráskódú rendszerek egyre nagyobb mértékben lefedik a fejlesztők igényeit. A fejlesztés során többször alkalmaztunk olyan programokat vagy eszközöket, amelyek nyílt forráskódúak, saját tapasztalat alapján elmondható, hogy a fizetős rendszereket (SBO) ingyenes forráskódú

eszközökkel kiegészítve az ügyfél számára hatékony és költségkímélő megoldásokat lehet előállítani.

Szállítási és fizetési címek

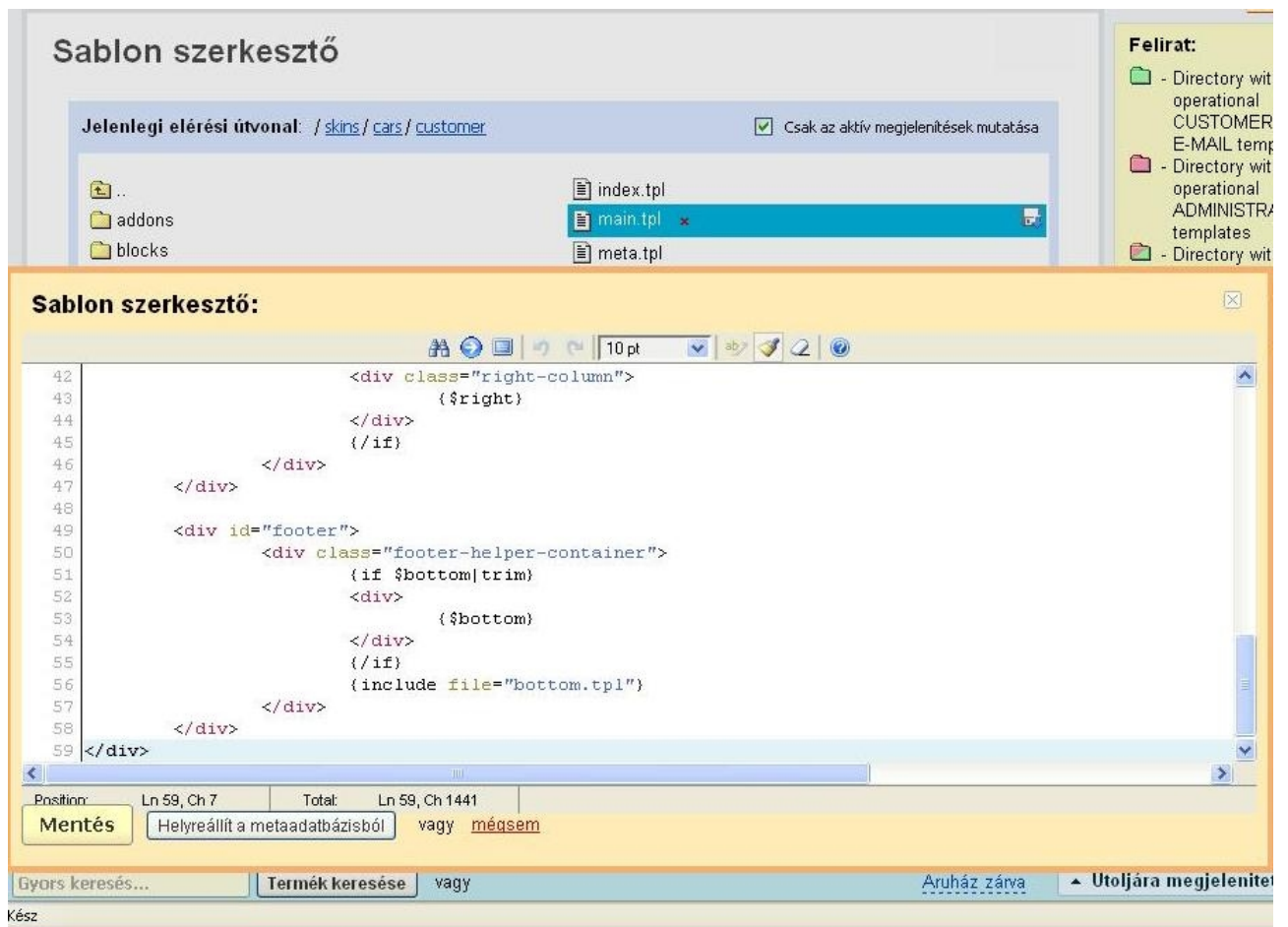
A következő fontos igény a megrendelő felől az volt, hogy egy vevőhöz tetszőleges számú postai és fizetési cím tartozzon, mivel a vevők többsége viszonteladóként szerepel a rendszerben. Egy viszonteladóhoz pedig több telephely és több központ létezhet. Erre a kérdésre azonban a CsCart nem tudott megoldást kínálni, csak egy postai és egy szállítási címet tárolt, amelyet lehetett ugyan rendelésenként módosítani, azonban a módosításokat csak a rendelés adatai között tárolta, nem volt lehetőség a már használt címek közötti választásra. Ennél a fázisnál kellett először kihasználni a CsCart moduláris felépítését azzal, hogy új kontrollert és új sablonokat illesztettünk a meglévő struktúrába (5-ös ábra).



5. ábra: a CsCart fájl szerkezete

Design módosítások – a CsCart felépítése

A CsCart készítői az alaprendszerhez verziótól függően 40-50 megjelenési sablont kínálnak, azonban ahhoz, hogy egy weboldal igazán szép és egyedi legyen, saját design tervet kell készíteni és implementálni. A megrendelő saját design tervet adott át számunkra, amely tükrözte a forgalmazott termékek minőségi színvonalát. Ennél a pontnál jött előtérbe a CsCart szerkezeti felépítése. A fejlesztők sikeresen implementálták az MVC sémát a webes környezetben. Minden funkcionalitáshoz saját kontroller kapcsolódott, valamint sablon fájlok halmaza, amelyek tartalmazták a megjelenéshez szükséges összes információt. A PHP sajátossága, hogy egy adott http kérésre adott válasz általában nem egy adott fájl futtatásával történik, hanem a programkód szét van darabolva több állományba és ezeket behívva áll össze a végleges forráskód. Ennek megvalósítása az include beépített paranccsal történik. Include állhat bárhol, ahol végrehajtható utasítás áll, hatására a paraméterként megadott állomány tartalma beépítésre kerül az hívási helyre. Ezt az elvet követte a sablonkezelő rendszer is. Azokat a weboldalelemeket amelyek önálló blokkot alkottak, külön sablonállományba helyezte el, így a módosítás során mindig csak egy kis részét módosítottuk a weboldalnak (6-os ábra). Előnye hogy könnyen és gyorsan lehet az egyes részelemeket szerkeszteni, hátránya hogy a fejlesztés során a fejlesztő sosem látja egészében a megjelenítő kódot, így egy szintaktikai hiba olyan problémákat is okozhat, amelyek más sablonfájlokban lévő kódok működését befolyásolják. Ezt a sablonkezelő rendszert a CsCart-ban a Smarty nevű keretrendszer integrálásával oldották meg a fejlesztők. A Smarty 2001 óta van jelen a weben, mint sablonkezelő és megjelenítő keretrendszer, nyílt forráskódú, szabadon felhasználható rendszer, amely a folyamatos fejlesztéseknek köszönhetően mára már számtalan hasznos funkciót tartalmaz, ilyen például a külső konfigurációs állományok használata, az adatok cache-selése, belső függvények használata, beépített hibakereső valamint a felhasználói add-on kiterjesztések. Itt is előjön, amit már korábban az Apache webkiszolgálónál, vagy a MySQL adatbázis-kezelő rendszerrel bemutattam, hogy jól összeválogatott ingyenes rendszerek kombinálásával olyan jól működő fejlesztői környezetet lehet felépíteni, amely felveszi a versenyt a szoftvergyártó cégek által kínált fizetős keretrendszerekkel.



6. ábra: a sablonszerkesztő működése a CsCart adminisztrációs felületén

A megjelenítés fejlesztése során alapvető követelmény volt, hogy a kapott weboldal lehetőleg minden platformon és böngésző programban ugyanúgy jelenjen meg. A weboldal fejlesztése során ez sok problémát okozhat, hiszen az internetes világban nincs elfogadott szabvány, csak egy „erőteljes” ajánlás arra vonatkozóan, hogy a HTML kódokat miként értelmezzék a böngészők. Ezt az ajánlást W3C konzorcium kontrollálja. Ez a konzorcium gyakorlatilag egy nemzetközi fejlesztői közösség, amely olyan szabvány terveket készít, amelyek segítségével ki lehet aknázni a webes világban lévő összes lehetőséget. A weboldal az XHTML 1.0 ajánlásait követi, amelynek a teljes leírását a W3C weboldala tartalmazza (<http://www.w3.org/TR/xhtml1/>). Ezen szabvány alkalmazásával a kész oldal a megrendelő igényei szerint jelenik meg a legnépszerűbb böngészők (Internet Explorer 6-7, FireFox 3, Opera 10, Google Chrome) mindegyikében.

3.2.A transzport réteg

A transzport réteg felelős az adatok áramlásáért a CsCart és az SBO között. A projekt elején lefektetett alapelvek szerint ezt a funkcionalitást csakis az SBO oldaláról lehetett megvalósítani, mivel a webservertől futtatott szkriptek nem férhetnek hozzá az SBO tábláihoz, fordítva viszont lehetséges a kommunikáció.

Az adatátvitelt Visual Basic nyelven megírt JOB-ok végzik, amelyek minden 10 percben lefutnak és leellenőrzik, hogy van-e olyan rendelés, amely még nem szerepel az SBO adatbázisában. Ezt az ellenőrzést egy állapotjelző mező segítségével valósítottuk meg, minden olyan rekord, amely átkerült az SBO-ba a transferred mezőbe I betűt kap.

Mivel ennek a Job-nak mindkét oldalon módosítania kell az adatbázist, így két különböző típusú adatbázis kapcsolatot kell kezelnie. A Visual Basic beépítve nem tartalmazza a MySQL kapcsolat felépítéséhez szükséges eszközöket, így azt külön telepíteni kellett. Az SBO felé egy MSSQL típusú adatbázis kapcsolatot kellett létrehozni, amely szintén egy önálló osztály beépítésével valósult meg.

Az adatbázis kapcsolatokon túl a JOB-nak csatlakoznia kell az SBO-hoz is. Erre a célra az SBO fejlesztői környezetét az SBO SDK –t használtuk. Ennek segítségével külön-külön lehet csatlakozni a felhasználói felülethez (User Interface – UI) valamint az adatkezelő réteghez (Data Interface – DI). Mivel a szóban forgó JOB-nak nincs szüksége felhasználói beavatkozásra a működéséhez, így a UI részét a kapcsolatnak nem fogjuk kihasználni, csak a DI részét. Nézzük meg, hogyan is működik a csatlakozás kódszinten:

Try

```
m_SboCompany = New SAPbobsCOM.Company

'gép neve ahol a server fut
m_SboCompany.Server = "localhost"
m_SboCompany.CompanyDB = "test_database"
'SBO-hoz kapcsolódó username és password
m_SboCompany.UserName = "sbo_test_user"
m_SboCompany.Password = "sbo_test_password"
'ők intézik az adatbázishoz való csatlakozást
m_SboCompany.DbUserName = "db_user"
m_SboCompany.DbPassword = "db_password"
m_SboCompany.DbServerType = BoDataServerTypes.dst_MSSQL2005
m_SboCompany.UseTrusted = True
```

```
If m_SboCompany.Connect() <> 0 Then
```

```

'A kapcsolódás során fellépő hibák kezelése
If 1 = 1 Then
    'catch error
    Dim ErrCode As Integer
    Dim ErrMessage As String

    m_SboCompany.GetLastError(ErrCode, ErrMessage)
    MSDataProvider.hiba_kiir(ErrMessage, "c:\temp\hiba.txt", True)
End If

Return False
End If

Return True

Catch excE As Exception
    MsgBox(excE.ToString)
End Try

```

Először is példányosításra kerül egy SAPbobsCOM.Company objektum. Ez a beépített eszköz fogja kezelni számunkra a kapcsolatot. Az SBO-hoz való csatlakozáskor nem elegendő az adatbázis hozzáféréseket megadni a programnak, hanem szükséges az SBO-hoz tartozós egyik licenc felhasználó nevét és jelszavát is megadni. Ennek segítségével a JOB, mint egy rendszerben regisztrált felhasználó fog tevékenykedni, így az általa elvégzett módosítások naplózásakor egyértelműen látszik, hogy az adott rekordot egy automatikus folyamat módosította. Ezek után már csak meg kell hívni a connect metódust és lekezelni az esetleges hibákat. Innentől kezdve a programunk csatlakozva van az SBO Data Interface-éhez és használhatjuk az általa kínált eszközöket.

A táblákhoz való hozzáférés során felhasználásra került még egy olyan eszköz is, amely nem része az SBO keretrendszerének, azonban használatával jelentősen csökkenthetjük a hibalehetőségek számát, illetve a hibakeresési folyamat is gyorsabb lesz. Ez az eszköz a DAO, azaz a Data Access Object. Ezeknek az osztálydefinícióját egy generáló kód hozza létre, paraméterként egy adatbázis tábla nevét kapja meg. A generálás során létrehoz egy olyan osztályt, amely tartalmazza azokat a DML utasításokat, amelyekre a munka során szükségünk lehet. Ezeket az utasításokat metódusok valósítják meg, minden metódusnak a paramétere egy adatrekord. Ehhez a rekordokhoz elkészíti és végrehajtja az SQL utasításokat. Miután egy-egy ilyen DAO osztályt legeneráltunk, igény szerint testre lehet szabni a metódusokat, így egy helyen lehet hozzáférni az adatbázist közvetlenül érintő utasításokhoz.

Íme egy példa egy DAO metódusra, amely az IFSZ_WEB_MEGRENDELESEK táblán végzi el az insert utasítást:

```
Public Function Insert(ByVal p_record As DataRow, ByRef p_message As String) As Integer

    Dim sqlQuery As New StringBuilder("Insert into [IFSZ_WEB_MEGRENDELESEK]
    (")
    Dim nfi As Globalization.NumberFormatInfo = New Globalization.NumberFormatInfo
    nfi.NumberDecimalSeparator = "."
    nfi.PercentGroupSeparator = ""

    If CType(p_record, DataRow).Table.Columns.IndexOf("ID") >= 0 Then
        If Not p_record.IsNull("ID") Then
            sqlQuery.Append("[ID]")
        Else
            p_message = ""
            Return -1
        End If
    Else
        p_message = ""
        Return -1
    End If

    ... ' további mezők kezelése
    'SQL utasítás lezárása
    sqlQuery.Append(");")

    ' Execute command
    Dim l_ret As Integer
    l_ret = MSDataProvider.ExecuteNonQuery(sqlQuery.ToString(), p_message)
    Return l_ret

End Function 'Insert
```

Egy rendelés létrehozása

Ahhoz, hogy a transzport eljárásunk sikeres legyen, szükségünk van arra, hogy új rekordokat tudjunk létrehozni az SBO adatbázisában. Az SBO biztosítja számunkra azt az eszközt, amellyel fel tudjuk vinni a saját rekordjainkat úgy, hogy az adatbázis konzisztenciája nem sérül. Erre a célra ebben az esetben az Items és az Documents osztályok példányait fogjuk használni. Az Items objektum az SBO raktárkezelő moduljának az eleme és egy raktárelemet reprezentál. A Documents objektum már egy kicsit érdekesebb, ugyanis az SBO-ban a számlák, rendelések, bizonylatok mind egy séma alapján vannak letárolva. Attól függően hogy milyen típusú a dokumentum, kiválasztja hogy melyik táblában tárolja el. Ezeknek a tábláknak a szerkezete minimális mértékben eltérhet ugyan egymástól, de általában ugyanazon elv szerint épülnek fel.

A rendelés létrehozásakor az első lépés, hogy összegyűjtjük azokat az adatokat amivel dolgozni szeretnénk. Ezt a feladatot a WebOrderProcess osztály feldolgoz metódusa látja el:

```
Public Sub feldolgoz()  
    Dim sqlQuery As String  
    Dim l_mrs_rows, l_mrst_rows As DataRowCollection  
    Dim l_mrs_row, l_mrst_row As DataRow  
    Dim l_bizonylatTipus As String  
    Dim l_autoBizonylat, l_listnum As String  
  
    'összegyűjtjük a fejadatokat  
    sqlQuery = "select * from IFSZ_WEB_MEGRENDELESEK where Statusz='O' order  
by id"  
    Try  
        l_mrs_rows = MSDataProvider.GetDataRecord(sqlQuery)  
    Catch ex As Exception  
        MSDataProvider.hiba_kiir(ex.ToString, "c:\temp\hiba.txt", True)  
    End Try
```

```

For Each l_mrs_row In l_mrs_rows
    'minden fejadathoz összegyűjtjük hozzá tartozó tételeket
    sqlQuery = "select * from IFSZ_WEB_MRS_TETELEK where mrs_id = " &
l_mrs_row.Item("ID").ToString & ""
    l_mrst_rows = MSDataProvider.GetDataRecord(sqlQuery)
    If l_mrst_rows.Count > 0 Then
        'az adatok feldolgozását a Megrendeles metódus végzi
        Me.Megrendeles(l_mrs_row, l_mrst_rows)
    End If
Next
End Sub

```

Miután összegyűjtöttük az aktuális rendeléshez tartozó tételek adatait, az egész adathalmazt átadjuk a Megrendeles metódusnak. Ebben a metódusban fognak megjelenni a korábban említett Items és Documents osztályok objektumai, amelyeknek az Add metódusa el fogja végezni az insert műveletet.

```

Public Sub Megrendeles(ByVal p_fej_row As DataRow, ByVal l_tetelek As
DataRowCollection)
    Dim i, l_ret As Integer
    Dim p_uj_sor As Boolean = True
    Dim l_docs_uj As SAPbobsCOM.Documents
    Dim l_item As SAPbobsCOM.Items
    ...
Try
    'Ha nincs valamilyen hiba miatt tétel, akkor a rendelés státuszát Törölt-re állítjuk
    If l_tetelek.Count = 0 Then
        Me.Update_Statusz(p_fej_row, "T")
        Exit Sub
    End If

    'l_docs_uj az SBO-s megrendeles bizonylat inicializálása
    l_docs_uj =
Me.m_parentaddon.SboCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.oOrders
)

    'Az SBO-s Item objektum inicializálása

```

```

    l_item =
Me.m_parentaddon.SboCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.oItems)

'Eldöntjük, hogy Nagyker vagy kisker wevöről van-e szó
If p_fej_row.Item("CardCode").ToString = "" Then
    l_docs_uj.CardCode = "WEB"
Else
    l_docs_uj.CardCode = p_fej_row.Item("CardCode")
End If

```

Ebben a metódusban valósul meg az az automatika, amely felelős a termékek transzportjáért. Ha egy olyan termékre érkezett megrendelés, amelyiknek az SBO_ID mezője üres, akkor az a termék még nem szerepel a SBO adatbázisában, így a folytatáshoz előbb fel kell vennünk új termékként:

```

If l_tetelek.Item(i).Item("ITEMCODE").ToString = "" Or
l_tetelek.Item(i).Item("ITEMCODE").ToString = "0" Then

    'Ha a cikk nem szerepel az SBO-ban, létrehozuk és az új Cikkszámmal a
rendelés tételt kiegészítjük.
    l_itemcode = Me.getItemCode(l_tetelek.Item(i).Item("ItmsGrpCod").ToString,
l_tetelek.Item(i).Item("FirmCode").ToString)
    'Így biztosan egy üres objektumot fogunk feltölteni
    If l_item.GetByKey(-1) = False Then
        l_item.ItemCode = l_itemcode
        l_item.ItemName = l_tetelek.Item(i).Item("WebItemName").ToString
        l_item.ManageBatchNumbers = SAPbobsCOM.BoYesNoEnum.tYES
        l_item.ItemsGroupCode = l_tetelek.Item(i).Item("ItmsGrpCod").ToString
        l_item.Manufacturer = l_tetelek.Item(i).Item("FirmCode").ToString
        l_item.UserFields.Fields.Item("U_katalogus").Value = "I"
        l_ret = l_item.Add
        If l_ret = 0 Then
            Me.ItemUpdate(l_tetelek.Item(i).Item("WebItemCode").ToString,
l_itemcode)
            Me.MrsItemUpdate(p_fej_row.Item("web_mrs_szam").ToString,
l_tetelek.Item(i).Item("WebItemCode").ToString, l_itemcode)
            l_docs_uj.Lines.ItemCode = l_itemcode
        Else
            Dim l_errmsg As String
            CType(m_parentaddon,
SBOAddon).SboCompany.GetLastError(l_error_code, l_errmsg)

            End If
        End If
    End If

```

Miután feldolgoztuk az összes rendelési tételt, már csak annyi a dolgunk, hogy a Document objektumra meghívjuk az Add metódust és lekezeljük az esetleges hibákat:

'Bizonylat létrehozása

```
l_error_code = l_docs_uj.Add()
If l_error_code <> 0 Then
    Dim l_text As String
    Dim l_errmsg As String

    CType(m_parentaddon, SBOAddOn).SboCompany.GetLastError(l_error_code,
l_errmsg)
    MSDataProvider.hiba_kiir(l_errmsg, "c:\temp\hiba.txt", True)
    Exit Sub
Else
    CType(m_parentaddon, SBOAddOn).SboCompany.GetNewObjectCode(l_docentry)
    Me.Update_Statusz(p_fej_row, "E", l_docentry)

End If

Catch ex As Exception
    MSDataProvider.hiba_kiir(ex.ToString, "c:\temp\hiba.txt", True)
    Me.m_parentaddon.SboCompany.EndTransaction(SAPbobsCOM.BoWfTransOpt.wf_
RollBack)
```

Összegezve még egyszer a folyamatot:

- minden 10ik percben elindul egy JOB
- a JOB csatlakozik az MSSQL és a MySQL adatbázisokhoz, valamint az SBO keretrendszer DI API-jához
- összegyűjti a MySQL adatbázisból azokat a rekordokat, amelyek még nem lettek áttöltve
- ezeket a rekordokat felveszi MSSQL adatbázisba
- leellenőrzi hogy van a rendelésekben olyan termék ami még nem szerepel az SBO-ban, ha talál ilyen akkor azt a terméket új termékként felveszi az SBO-ba az Items beépített osztály segítségével
- ha a rendelésekhez tartozó termékek mindegyike szerepel az SBO-ban, akkor a rendelésekből SBO-s rendeléseket generál a Documents beépített osztály segítségével
- a feldolgozott rekordokat update-eli a MySQL adatbázisban, a feldolgozva mezőbe I került ha a feldolgozás során nem történt hiba

Ez az egyik legfontosabb folyamat, amit az AddOn tartalmaz, megvalósítja a kapcsolatot két olyan szoftver között, amelyek más programnyelven íródtak, más adatbázis-kezelő rendszert használnak és más irányelvek szerint lettek megírva.

Árképzés

A megrendelőnek volt egy igen fontos igénye, amelyet szintén az adatok áramlásának fejezetéhez lehet kötni. Vannak olyan nagykereskedelmi partnerek, akik a weboldalon keresztül vásárolnak, viszont a részükre személyre szabottan kedvezményeket biztosít az eladó. Ez a kedvezmény lehet mennyiségi, fix vagy időszakos. Az SBO-ban ezzel nincs is probléma, hisz az árképzésre be vannak építve az eljárások, azonban a weboldalon ezek az eljárások nem léteztek. A másik probléma pedig az volt, hogy az árképzéshez szükséges adatok az SBO adatbázisában vannak letárolva, a weboldal pedig nem olvashatja az SBO-s táblákat. Megoldásként írni kellett egy olyan SQL függvényt, amely megkapja paraméterként a termék és a vásárló azonosítóját, az aktuális darabszámot és az aktuális dátumot. Ezek alapján lemodellezi az SBO árképzési folyamatát és meghatározza, hogy az adott vevőnek, az adott terméket milyen áron jelentesse meg a rendszer. Ehhez a folyamathoz szükséges volt olyan adatokat is áthozni a webes adatbázisba, amelyek nem kapcsolódnak közvetlenül a megrendelésekhez, hanem a vásárlóhoz rendelt árlisták rekordjait tartalmazzák. Itt ismét szerepet kapott a MySql 5.0.1-es verzióba bekerült újítás, a függvények használata.

```
DROP FUNCTION IF EXISTS `IFSZ_EGYSEGAR` $$
CREATE DEFINER=`root`@`localhost` FUNCTION `IFSZ_EGYSEGAR`(p_product_id
varchar(20), p_user_id varchar(20), p_date date, p_quantity INT) RETURNS decimal(10,2)
    DETERMINISTIC
BEGIN
DECLARE avg INT;
DECLARE pCardCode varchar(20);
DECLARE pItemCode varchar(20);
DECLARE ListaAr decimal(10,2);
DECLARE AkciosAr decimal(10,2);
DECLARE AkciosArMennyiseg decimal(10,2);
DECLARE PartnerSpecAr decimal(10,2);
DECLARE PartnerAkciosAr decimal(10,2);
```

```

DECLARE PartnerAkciosArMennyiseg decimal(10,2);
DECLARE CikkszoportAr decimal(10,2);
select SBO_ID into pCardCode from cscart_users where user_id = p_user_id;
select SBO_ID into pItemCode from cscart_products where product_id = p_product_id;
IF pItemCode > " then
    IF pCardCode = " THEN SET pCardCode = 'WEB' ;
END IF;
select Price into ListaAr
    from itm1 M1 join ifsz_partnerek cr on CR.listnum=M1.pricelist
    and CR.Cardcode=pCardCode
    and M1.ItemCode=pItemCode;
select case when SP.AutoUpdt='Y' then M1.Price*(1-P1.Discount/100) else SP.Price end into
AkciosAr
    from itm1 M1
    join ifsz_partnerek cr on CR.listnum=M1.pricelist and CR.Cardcode=pCardCode
    join OSPP SP on SP.Itemcode=M1.Itemcode and M1.pricelist=SP.listnum and
SP.Cardcode like '%*%'
    join SPP1 P1 on SP.Itemcode=P1.Itemcode and P1.listnum=SP.listnum and
P1.CardCode =SP.CardCode and SP.Expand='Y'
    and p_date between P1.fromdate and ifnull(P1.todate,'20990101')
    and M1.ItemCode=pItemCode;

```

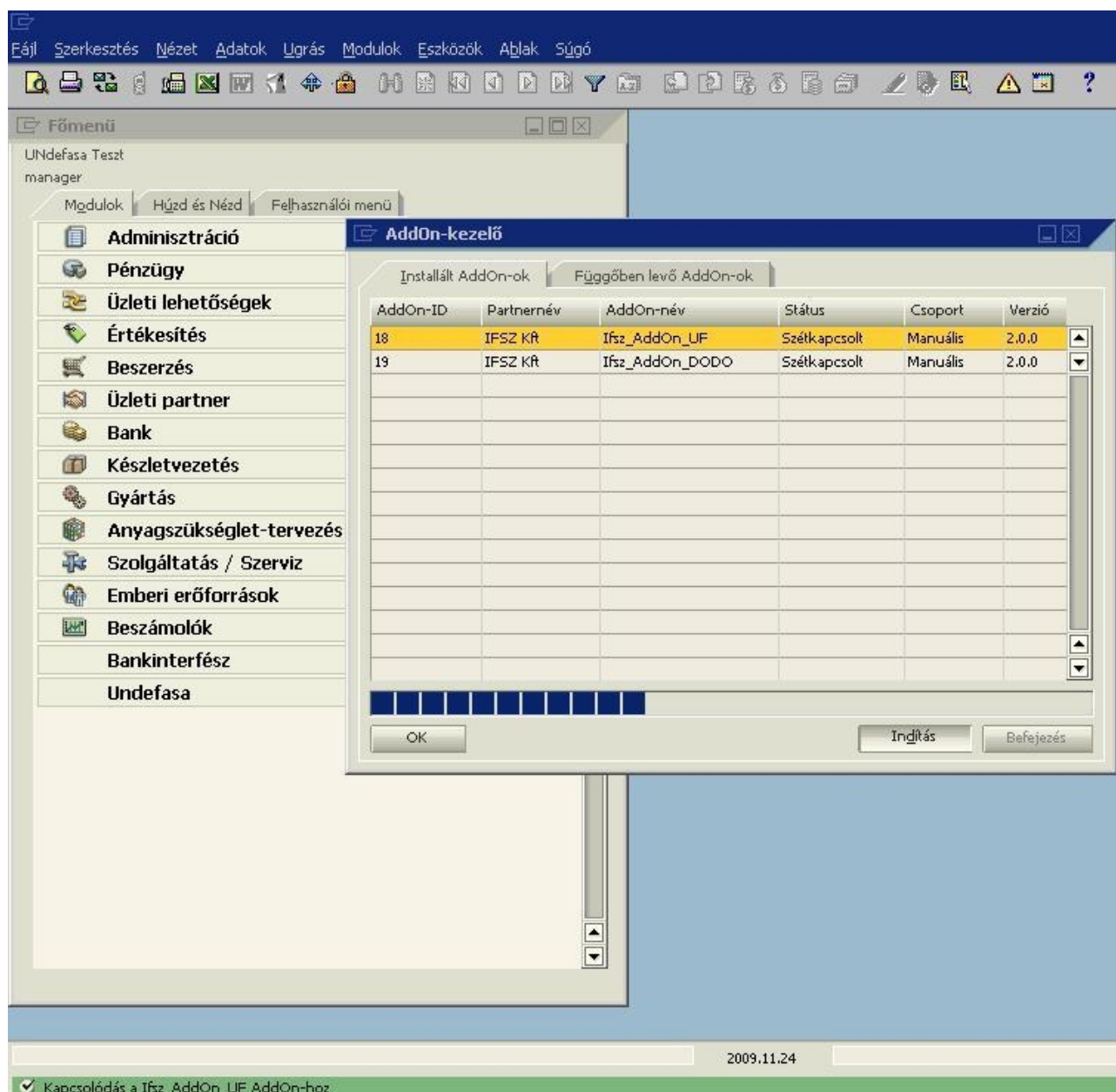
Ahhoz, hogy az árképzés szükséges adatok naprakészek legyenek, az SBO oldalon kell egy eljárás, ami szinkronban tartja a két adatbázisban lévő táblákat. Erre leginkább egy adatbázisbeli trigger lenne alkalmas, de mivel a két adatbázis eltérő helyen, eltérő kiszolgáló alatt fut, így ez a megoldás nem volt alkalmazható. Két megoldási lehetőség került szóba a fejlesztés során. Az egyik, hogy minden egyes DML utasítás után ugyanazt a változtatást a MySQL adatbázisban is elvégezzük, ez azonban elég sok hibalehetőséget hordoz magában. Több ponton lehetséges ezeket a rekordokat módosítani, ha ezek közül a pontok közül valamelyik kimarad, akkor sérül az adatbázisok szinkronja. A másik megoldás sokkal kézenfekvőbb volt, mivel módosított vagy új rekord csak az SBO adatbázisában jöhet létre, így elegendő bizonyos időközönként az SBO-ban módosult rekordokkal egyszerűen felülírni a MySQL-ben tárolt táblákat. Ezt a felülírási folyamatot a megrendelések transzferjéhez hasonlóan rendszeres időközönként futtatva biztosítani lehet a két adatbázis szinkronját.

3.3. Az ERP oldal – Sap Business One

Az fejlesztés során három nagy feladathalmaz került dokumentálásra, majd leprogramozásra. Az első két oldalról már volt részletesen szó, ez pedig a weboldal és a hozzá kapcsolódó fejlesztések valamint az adatok transzferre. A harmadik csoport az SBO-hoz kapcsolódó fejlesztések voltak. Ebben a fejezetben bemutatom, hogyan épül fel az SBO AddOn kezelő rendszere, illetve hogy a kész AddOnokat milyen felületen lehet elérni, milyen módon lehet velük elvégezni a mindennapi munkákat.

Az SBO felépítése

Az SBO kezelői felülete három alapvető ablaktípusból épül fel, ezek a főablak, az elsődleges és alárendelt ablakok valamint az üzenetek. Indításkor a főablak látható, bal oldalon az elérhető modulokhoz tartozó menüpontokkal. Az egyes modulok kezelőfelülete új ablakban nyílik meg, a főoldaltól függetlenül. Ezekben az ablakokban jelennek meg a modulokhoz tartozó formok és adattáblázatok. Beállítástól függően az SBO-hoz készített AddOn-ok indulhatnak automatikusan vagy manuálisan is. Manuális indítás során az Adminisztrációs modul AddOn kezelő részéből lehet elindítani a használni kívánt AddOn-t (7-es ábra).



7. ábra: SBO AddOn indítása

Miután az AddOn csatlakozott az SBO-hoz, elérhetővé válnak számunkra azok a formok és adattáblázatok, amelyek a kiegészítőnkhez tartoznak. A főablakban a menüsorban megjelent menüpont, amely a csatolt AddOn vezérlését végzi.

A 8-as ábrán azt az adattáblázatot láthatjuk, amely a CsCart-ból érkező megrendeléseket listázza és kezeli.

The screenshot shows a software application window titled "Webes megrendelések" (Web orders). The window has a menu bar with options: Fájll Szerkesztés Nézet Adatok Ugrás Modulok Eszközök Ablak Sűgő. Below the menu bar is a toolbar with various icons. The main area contains two tables and a control panel.

Table 1: Order Summary

Partner kód	Partner név	Azonosító	Rög. dátuma	Szállítási cím	SBO azonosító	Státusz
→ 10080	HALAJ F...	33	2009.11.19. 19:24		→ 91000038	Feldolgozott
→ WEB	Tamas Ber...	32	2009.11.19. 16:40		→ 91000037	Feldolgozott
→ WEB	Tamas Ber...	31	2009.11.19. 16:38		→ 91000036	Feldolgozott
→					→	

Table 2: Order Details

Cikk kód	Cikk név	Egységár	Mennyiség	Érték
→ 101bembem001	Cenefa dédalo ceniza 4 x 33 cm	100,00	1,00	100,00
→ 101bembem002	Decor 2 dédalo gris flor 33 x 60 cm	100,00	1,00	100,00
→				
→				
→				
→				
→				
→				

Control Panel:

Nyitott
 Elutasított
 Feldolgozott

Navigation Panel:

- Cikk sarzs lista
- Árazó
- Értékesítési árképző
- Beszerzési árképző
- Webes megrendelések

Status Bar:

2009.11.24 02:02

8. ábra: megrendelések listázása

Ezzel a képernyőképpel gyakorlatilag le is zárul az AddOn dokumentálása. A weboldalról érkező megrendelések sikeresen átkerültek az SBO-ba, innenől az SBO saját megrendelés (9-es ábra), bizonylat és számlakezelő moduljai képesek kezelni a rekordokat. A változtatásokat pedig a korábban kitérgyalt folyamatok visszavezetik a webes adatbázisba

Fájl Szerkesztés Nézet Adatok Ugrás Modulok Eszközök Ablak Súgó

vévi rendezés

Vevő: 10080
Név: HALAJ FERENC TÚZÉP
Tárgyalópartner:
Vevő ref.szám:
Saját pénznem:
Szám: 2009DEB 91000038
Státus: Nyitott
Könyvelési dátum: 2009.11.19
Szállítás dátuma: 2009.11.19
Bizonylatdátum: 2009.11.19
Manuális szám:
Tartalom Logisztika Pénzügy

Bizonylatfajta: Cikk Összefoglalás típusa: Nincs összefoglalás

#	Cikkszám	Cikk leírása	Kivánt menny...	Mennyiség	ME	Egységár	Adó kód	Öss...
1	101bembem001	Cenefa dédalo ceniza 4 x 33 cm	0,000	1		100,00 Ft	K20	
2	002kádegy004	NYMFA 160x70 kád + láb Alföldi	0,000	1	db	800,00 Ft	K25	
3							K20	

Értékesítő: -értékesítő nélkül-
Tulajdonos:
Megjegyzések: aa

Összesen (eng. nélkül): 900 Ft
Engedmény: %
 Kerekítés
Adó: 220 Ft
Összesen: 1.120 Ft

OK Mégsem Másolás forrása Másolás célja

9. ábra: megrendelés szerkesztő felülete az SBO bepített felületén

3.4.Összefoglalás

A projekt célja egy e-Business alkalmazás kifejlesztése volt az SAP Business One vállalatirányítási rendszerhez. Habár a szakdolgozat elkészültekor a projekt még nem zárult le, de a kezdeti célkitűzéseket nagy többségében elértük. Továbbfejlesztettük a CsCart nevű webáruház kezelő rendszert, amely így képes volt fogadni az SBO-ból érkező adatokat. Elkészítettünk egy olyan interfészt, amely összeköt két egymástól független rendszert, ezáltal az egyszerű weboldal szerves része lett a megrendelő üzleti stratégiájának és a vállalatirányítási rendszerének.

A megismert eszközök mindegyike napjainkban használt és elfogadott technológiákon alapult. Az Apache webservertől kezdve a MySQL adatbázis-kezelőn át Visual Basic nyelven megírt DAO objektumokig mindegyik eszköz hozzá tartozik egy piacképes programozó tudásbázisához. A munka során törekedtünk arra, hogy a felhasznált alkalmazások megfeleljenek az aktuális informatikai trendeknek és nyílt forráskódúak legyenek. Ezen alkalmazások segítségével sikerült összeállítani egy olyan fejlesztői környezetet, amely sikeresen tud támogatni egy olyan fejlesztést, amelynek a célpiaca a kis és középvállalkozások.

A munka során alkalmas volt megtapasztalni, hogyan lehet egy senior programozó irányítása alatt csapatmunkában dolgozni, illetve megismerhettem a gyakorlati alkalmazásait a rendszerfejlesztési technológiákról tanultaknak. Lehetőségem nyílt arra, hogy részt vegyek programozói megbeszéléseken valamint személyes ügyfél interjúkon. Ezek a tapasztalatok véleményem szerint szervesen be fognak épülni a pályafutásomba.

Ezúton szeretnék köszönetet mondani az Ifsz Kft-nek hogy részt vehettem a fejlesztésben, Bérczes Tamásnak, hogy vállalta a projekt irányítását illetve Navratil Zoltánnak hogy munkájával és szakértelmével részt vállalt a célok megvalósításában.

4. Irodalomjegyzék

- Michelberger Pál : Vállalati Információs Rendszerek Jövője
2006.07.27. <http://www.ecm-certification.com/virj.doc>
- Controlling prtál: ERP rendszereknek
http://www.controllingportal.hu/?doc=it_erp
- Wikipédia :Enterprise resource planning
http://en.wikipedia.org/wiki/Enterprise_resource_planning
- SAP Business One Általános ismertető
<http://www.sap.com/hungary/smallbusiness/solutions/overview/index.epx>
- Wikipédia: Material Requirements Planning
http://en.wikipedia.org/wiki/Material_Requirements_Planning
- CsCart Tudásbázisához
<http://kb2.cs-cart.com/>
- Apache http szerver dokumentációja
<http://httpd.apache.org/>
- XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)
<http://www.w3.org/TR/xhtml1/>
- MySql 5.0.1 dokumentáció
<http://dev.mysql.com/doc/refman/5.1/en/>
- Smarty dokumentáció
<http://www.smarty.net/manual/en/preface.php>
- PHP dokumentáció
www.php.net/manual

5. Függelékek

5.1. Felhasznált rövidítések

ERP : Enterprise Resource Planning, Vállalatirányítás rendszer

MRP: Manufacturing Resource Planning, Gyártási erőforrástervező rendszer

B2B: Business to Business, Cégek közötti üzleti kapcsolat

B2C: Business to Customer, Cégek és a vásárlók közötti üzleti kapcsolat

B2A: Business to Administration, Cégek és a közigazgatás közötti kapcsolat

CRM: Customer Relationship Management, ügyfélkapcsolatokat kezelő rendszerek

SCM: Supply Chain Management, ellátási lánc menedzsment rendszerek

MIS: Management Information System, vezetői információs rendszerek

SBO: SAP Business One, az SAP egyik vállalatirányítási rendszere

SEO: Search Engine Optimization, weboldalak szerkezeti optimalizálása

MVC: Modell-View-Controller, szoftverfejlesztési tervezési minta

API: Application Interface, olyan felület, amely segítségével külső alkalmazások csatlakozhatnak a szoftverhez

PHP: Hypertext Preprocessor, szerver oldali programozási nyelv

SAP: SAP AG, német szoftverfejlesztő cég, fő profilja az ERP rendszerek fejlesztése

SQL: Structured Query Language, relációs adatbázis-kezelő rendszerek programozási nyelve

FTP: File Transfer Protocol, hálózati állományátvitelre szolgáló szabvány

SSH: Secure Shell, szabványcsalád és protokoll, amit egy helyi és egy távoli számítógép közötti biztonságos csatorna kiépítésére fejlesztettek ki.

Html: HyperText Markup Language, leírónyelv, amelyet weboldalak készítéséhez fejlesztettek ki

AJAX: Asynchronous JavaScript and XML, interaktív weboldalak fejlesztésére szolgáló webfejlesztési technika

W3C: World Wide Web Consortium, nemzetközi közösség, amely a webszabványokat készíti

UI / DI: User Interface / Data Interface, az SBO által biztosított csatlakozási felületek, amelyek segítségével elérhető a felhasználó felület, vagy az adatállományok

DAO: Data Access Object, adattáblázatok hozzáférését biztosító objektumok

DML: Data Manipulation Language, programnyelvek egy családja, amely adatbázisműveleteket végez el