

SZAKDOLGOZAT



MISKOLCI EGYETEM

Androidos üzletkötő kliens (ügynök program) fejlesztése

Készítette:

Maradics Zoltán

2015. Programtervező Informatikus

Korszerű web-technológiák

Témavezető:

Szűcs Miklós

MISKOLC, 2014

SZAKDOLGOZAT FELADAT

Maradics Zoltán (DWOKX4) programtervező informatikus jelölt részére.

A szakdolgozat tárgyköre: szoftverfejlesztés

A szakdolgozat címe: Androidos üzletkötő kliens (ügynök program) fejlesztése

A feladat részletezése:

Az Android operációs rendszer világban elfoglalt helyzete, és az ezt futtató eszközök felhasználási körének ismertetése,

Androidos kliens fejlesztése, mely alkalmas a következő funkciók megvalósítására:

- külső forrásból érkező adatok alapján rendelések készítése, módosítása, tárolása, visszatöltése,
- ügyelve arra, hogy a program általános felhasználású legyen.
- Ismertesse a tervezés, a megvalósítási, a tesztelés fázisait, és a továbbfejlesztési lehetőségeket.

Témavezető: Szűcs Miklós Mérnökstanár

Konzulens: Dr. Karácsony Zsolt egyetemi docens

A feladat kiadásának ideje: 2014.09.22

A feladat beadásának határideje: 2014.11.21

.....
szakfelelős

EREDETISÉGI NYILATKOZAT

Alulírott Maradics Zoltán Neptun-kód: DWOKX4 a Miskolci Egyetem Gépészmérnöki és Informatikai Karának végzős Programtervező informatikus szakos hallgatója ezennel büntetőjogi és fegyelmi felelősségem tudatában nyilatkozom és aláírással igazolom, hogy Androidos üzletkötő kliens (ügynök program) fejlesztése című szakdolgozatom/diplomatervem saját, önálló munkám; az abban hivatkozott szakirodalom felhasználása a forráskezelés szabályai szerint történt.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- szószerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát megismertem, és tudomásul veszem, hogy plágium esetén szakdolgozatom visszautasításra kerül.

Miskolc, 2014.11.21.

.....

Hallgató

1.

szükséges (módosítás külön lapon)

A szakdolgozat feladat módosítása

nem szükséges

.....

dátum

.....

témavezető(k)

2. A feladat kidolgozását ellenőriztem:

témavezető (dátum, aláírás):

konzulens (dátum, aláírás):

.....

.....

.....

.....

.....

.....

3. A szakdolgozat beadható:

.....

dátum

.....

témavezető(k)

4. A szakdolgozat szövegoldalt

..... program protokollt (listát, felhasználói leírást)

..... elektronikus adathordozót (részletezve)

.....

..... egyéb mellékletet (részletezve)

.....

tartalmaz.

.....

dátum

.....

témavezető(k)

5.

bocsátható

A szakdolgozat bírálatra

nem bocsátható

A bíráló neve:

.....

dátum

.....

szakfelelős

6. A szakdolgozat osztályzata

a témavezető javaslata:

a bíráló javaslata:

a szakdolgozat végleges eredménye:

Miskolc,

.....

a Záróvizsga Bizottság Elnöke

Tartalomjegyzék

1. Bevezetés	7
1.1. Cégek bemutatása	8
1.1.1. 3Kéz Bt.	8
1.1.2. Bora Kft.	8
2. Téma elméleti kifejtése	10
2.1. Android	10
2.1.1. Android 1.0	11
2.1.2. Android 1.1	11
2.1.3. Android 1.5 (Cupcake)	11
2.1.4. Android 1.6 (Donut)	12
2.1.5. Android 2.0 és 2.1 (Eclair)	12
2.1.6. Android 2.2 (Froyo)	13
2.1.7. Android 2.3 (Gingerbread)	13
2.1.8. Android 3.0 (Honeycomb)	14
2.1.9. Android 3.1	14
2.1.10. Android 3.2 (Ice Cream Sandwich)	15
2.1.11. Android 4.0 (Ice Cream Sandwich)	15
2.2. Feladat Specifikálása	15
2.2.1. Összefoglalva	16
2.3. Modulokra bontás	17
2.4. Tervezés	18
2.4.1. Adatbázis	18
2.4.2. Felhasználói felületek	19
2.4.3. Forrás feldolgozás	21
2.4.4. Adatbázis megjelenítő	21
2.4.5. Rendeléskezelő	22
2.4.6. Kimenet előállító	23
2.4.7. Beállítások	23
3. Fejlesztői dokumentáció	24
3.1. Fejlesztés	24
3.1.1. Felületek	24
3.1.2. Adatbázis	25
3.1.3. Adatfeldolgozó	27
3.1.4. Adatbázis megjelenítő	29
3.1.5. Rendelés kezelő	30
3.1.6. Rendelések mentése	31

3.1.7. Beállítások	33
3.2. Tesztelés	34
3.3. Módosítások	34
3.3.1. Első módosítás	35
3.3.2. Második módosítás	35
3.4. Továbbfejlesztési lehetőségek	35
3.5. Felhasználói dokumentáció	36
3.6. Mellékletek	38
4. Összefoglalás	48
4.1. Összegzés	48
4.2. Summary	49
Irodalomjegyzék	50
Adathordozó használati útmutató	51

1. fejezet

Bevezetés

A mai modern világban sokan használnak okos telefont, tabletet ügyeik intézésére. Tehát adott a platform és felhasználóbázis. Egy jól megírt program gyorsabban és precízebben dolgozik, mint egy adminisztrátor. Az adatrögzítési feladatok automatizálása egy megfelelő célterület, ahol még vannak lefedetlen piacok.

Igény megjelenése:

Megfigyelést végeztem több értékesítő cégnél is, hogy zajlik a rendelés felvételi folyamat. Alkalmam volt betekinteni olyan rendszerbe, ahol a rendelés kiszállítás akár egy napon is történhet, de volt olyan is, ahol mindig másnap történt a kiszállítás. A rendelések papír alapon, telefonos vagy fax formájában érkeznek a megrendelőktől az adminisztrátorokhoz. Sok adatrögzítőt kell alkalmazni, ezen adatok digitális feldolgozásra, hogy lépést tudjanak tartani a beérkező rendelésekkel. A vállalkozások gyorsabb rendelés felvétellel hamarabb tudják előállítani a szállítandó (értékesítendő) áruk kiszállítási tervét, mind az útvonal és rakomány tekintetében. Ebből következik, ha automatizálnánk a rendelések felvételét, jelentős időt lehetne spórolni. Csökkentve a hiba lehetőségét ami, a kommunikációból és elírásból ered. A rögzítés és teljesítés közötti időminimalizálásra törekedve, sok kis költségvetésű vállalkozást érdekelne az alkalmazás. Ők lennének a legfőbb célcsoport, mert nem rendelkeznek olyan anyagi forrásokkal, ami egy komplex rendelés, számlázó és logisztikai szoftvert tudna megfizetni. A hozzá tartozó támogatás is havi vagy éves fizetése sem alacsony. Továbbá a komplett dobozos szoftverek nehezen lehet személyre szabni a vállalkozások igényei szerint. A felmerülő módosítások implementálása az egész terméket módosítaná és tartalmazhat olyan komponenseket is, amire az adott cégnek nincs szüksége (van alternatív szoftver).

Például: Az ügynök reggel elindul az előző nap felvett rendeléseivel a partnerhez. A nap folyamán számos cégnek szállít, és közben felveszi a másnapi rendelést (papír alapon). A nap végén mikor visszatér a telephelyre, átadja a felvett rendeléseket az adatrögzítőknek. A felvételt követően elkészülnek a szállítmány listák, amiket az este folyamán összeállítanak, hogy reggel újra kezdjék a kiszállítást és a rendelés felvételt.

Az alapötlet az volt, kínálni a cégeknek egy digitális felületet, amin a rendelés felvételt és adatcserét tudják végezni, valós időben. A nap végén megkapott rendelések helyett már nap közben lehet előkészíteni a következő szállítmányokat.

1.1. Cégek bemutatása

1.1.1. 3Kéz Bt.

1990-ben alakult a Vonalkód cég, amely vonalkód olvasókkal, számítógép szereléssel és egyéni programok megírásával foglalkozott. 1997-ben kettévált Vonalkód és a 3 Kéz Bt. cégekre. A Vonalkód továbbvitte a hardveres részét a vállalkozásnak, még a 3 Kéz Bt. a szoftveres területet fejlesztette tovább. A cég fő profilja az egyedi szoftverek készítése majd fejlesztése és karbantartása. Partnerei a kereskedelem minden részében megtalálhatók. Pl.:jövedéki kereskedés (cigaretta, alkohol), gyártó cégek, kis- és nagykereskedések, húskereskedések, kerékpárgyártó vállalatok. Használják még a programjait vízóra leolvasáshoz, munkaruha legyártásához szükséges anyagok összeszámlálásához (cérnahossz, anyagfelület mennyiség), almafelvásárlásban használatos mérlegek adatainak gépen rögzítéséhez.

1.1.2. Bora Kft.



A Bora Ital-nagykereskedelmi Kft. a '90-es évek elején kezdte meg működését, jelenleg Békés megye teljes területén illetve Jász-Nagykun-Szolnok megye jelentős részén van jelen szolgáltatásaival.

A vállalkozás fő tevékenysége élelmiszer, ital vegyes nagykereskedelme.

A Kft. a TEÁOR 4639 számon jelzett tevékenységen belül elsősorban sör, üdítőitalok, ásványvíz, és egyéb szeszes italok nagy- és kiskereskedelmével foglalkozik.

Szolgáltatásaink közé tartozik még rendezvények (a nagy volumenűtől egészen a kis családi összejövetelekig) teljes körű ital-, és eszköz ellátása is. Rendelkezünk minden olyan eszközzel, ami egy sikeres rendezvény lebonyolításához elengedhetetlen: hűtők, sörcsapók, sörszotalok, sörpadosok, óriás napernyők, sátrak a 3x3m-es mérettől egészen a több száz négyzetméteres nagysáorig.

Szakképzett területi képviselői csapatunk rendszeresen látogatják viszonteladói Partnereinket, akiktől a heti akciókról, aktuális promóciókról kapnak bővebb felvilágosítást.

Italkínálatunkban a legolcsóbb kategóriájú termékektől a legmagasabb prémium italok is megtalálhatók.

A Bora Kft. a multinacionális láncok saját márkás termékeire és az ehhez kapcsolódó árversenyre reagálva az italpiacon szinte minden szegmensben kifejlesztette és forgalmazza a Bora Kft. saját márkás termékeit. Ezek közül mára a Hunyadi Mátyás világos sör országosan ismert branddé vált a piacon.

A Bora-Ital Kiskereskedelmi Kft. mára 26 üzlettel rendelkezik Békés és Jász-Nagykun Szolnok megyében: Békéscsaba Corvin utca, Békéscsaba Szarvasi út, Békés, Battonya, Dévaványa, Doboz, Elek, Gyomaendrőd, Gyula, Karcag, Kétsoprony, Kondoros, Kunszentmárton, Mezőberény, Medgyesegyháza, Mezőkovácsháza, Mezőtúr, Nagyszénás, Orosháza, Sarkad, Szarvas, Szeghalom, Szolnok, Tótkomlós, Túrkeve, Újkígyós.

Célunk a viszonteladó partnerek és a lakossági vevők elérése és igényeik teljes körű kiszolgálása. Az italdiszkontoknál fontos szempont az egységes megjelenés és az üzlet nagyságához igazodó egységes külső és belső arculat.

Minden üzletünk a lakosság számára jól megközelíthető frekvenciált helyen található és parkolóval rendelkezik.

A Bora Kft. 2001 óta minden évben megrendezi a Csabai Sörfesztivál és Csülökparádét, amely Magyarország legnagyobb és leglátogatottabb sörfesztiváljává és egyetlen csülökparádéjává nőtte ki magát.[5]

2. fejezet

Téma elméleti kifejtése

2.1. Android

Az android azzal a céllal készült, hogy legyen egy egységes nyílt forráskódú operációsrendszer mobil eszközökre. Az alapötlet egy Linux operációs rendszer, amit úgy alakítanak át, hogy képes legyen kezelni a mobil eszközök hardverelemeit (érintőkép-ernyő, wifi, bluetooth).[4]

„A rendszernek egy ideig külön verziója létezett telefonokra (2.x) és táblaPC-kre (3.x) optimalizálva, ám a most megjelent (2012. január) 4.0-s verzió egyesíti ezt a két vonalat, így a jövőben várhatóan minden eszközön ugyanaz a verzió fut majd.”[1]

„Az Android napjaink egyik legnépszerűbb, és bátran állíthatjuk, hogy egyik legfejlettebb operációs rendszere. A népszerűség egyik fő oka a rengeteg médiareklám. Az Android előtt kevés mobileszközt reklámoztak az azt futtató operációs rendszerrel, így az Android mint operációsrendszer-név kellően újszerű volt a felhasználók számára. Megfigyelhető, hogy új mobileszköz vásárlásakor a felhasználók már nem is feltétlenül a készülékgyártó alapján, hanem operációs rendszer szerint keresnek eszközöket. A sikerhez természetesen az is hozzátartozik, hogy az Android mögött, az "IT-óriás", a Google áll. Az Android további előnye, hogy egy rendkívül egységes és kiválóan működő rendszerképet tükröz a felhasználóknak, és elfedi az egyes verziószámokat, valamint a köztük lévő különbségeket. Az Android az iOS-szel karöltve tulajdonképpen forradalmasította az operációs rendszerekről alkotott képet, és a felhasználói igények kielégítése is egyre nagyobb hangsúlyt kapott a mobil operációsrendszerek piacán. Az Android jellemzője, hogy nemcsak mobiltelefonokon, hanem táblagépeken is fut, ezek elterjedése pedig napjainkban rohamosan nő. Továbbá az Androidot úgy tervezték, hogy akár más eszközön is könnyen futtatható legyen: akár televíziókról, gépjárművek fedélzeti számítógépéről, ipari automatizálási rendszerről vagy háztartási eszközökről. Tulajdonképpen az Android minden olyan helyen kényelmesen használható, ahol relatíve limitáltak az erőforrások, és az adatbevitel nem feltétlenül egerrel és/vagy billentyűzettel történik.”[1]

2.1.1. Android 1.0

- 2008. október 21-én jelent meg.
- Apache-licenc.
- Egy szűk fanatikusokból álló rétegen kívül nem nyerte el igazán az átlag felhasználók tetszését.
- A platform stabilitása megfelelő volt, a használhatósága azonban nehézkesnek bizonyult.
- A UI nem volt megfelelő.
- A HTC által gyártott G1 is inkább csak koncepciótelefon volt, hogy elősegítse a fejlesztők munkáját, illetve felkeltse a cégek érdeklődését.[1]

2.1.2. Android 1.1

- 2009 februárjában jelent meg.
- Felkerült a G1 telefonokra (frissíthetőség tesztelése).
- Sok apró hibát javított, ezeket az 1.0-s kiadásától fogva gyűjtötték
- Látványos változtatásokat nem tartalmazott.[1]

2.1.3. Android 1.5 (Cupcake)

- 2009 áprilisában jelent meg.
- 2.6.27 verziójú Linux-kernelen alapul.
- A szoftveres billentyűzet automatikus kiegészítési funkciójával rendelkezik.
- A2DP Bluetooth-támogatása, illetve automatikus headset csatlakozása van.
- Új UI-komponensek jelentek meg benne.
- Animációkat vezettek be a képernyőváltások között.
- Feljavították másolás-beillesztés funkciót.
- Videók és képek közvetlen feltöltése vált lehetségessé a YouTube és a Picasa portáljaira.[1]

2.1.4. Android 1.6 (Donut)

- 2009 szeptemberében jelent meg.
- Az előző verzió javítása.
- Amlroid Market-javításokat tartalmaz.
- Feljavított galériefunkcionalitásokkal rendelkezik (több kép kijelölése közös művelethez).
- Hangfelismerésen alapuló funkciók vannak benne.
- Teljes platformban képes keresni az alkalmazás megjelenését.
- A használt technológiák frissítését, a WVGA-felbontás támogatását, egyéb optimalizálásokat tartalmaz.[1]

2.1.5. Android 2.0 és 2.1 (Eclair)

- 2009 októberében jelent meg.
- Nagyobb verzióváltás történt.
- 2.6.29-es Linux-kernel-támogatás.
- Hardveroptimalizációt is tartalmaz.
- Változatos képernyőméretek és felbontások támogatása (netbook- és táblagép-támogatás).
- Újraértelmezett grafikus felület HTML5 támogatással
- Multitouch támogatása.
- Bluetooth 2.1-es támogatás.
- "Élő" háttér megjelenése.
- A 2.0 kiadása után nem sokkal érkezett a 2.0.1-es verzió, amely több apró- de bosszantó - hibát javított.
- 2010 januárjában jelent meg a 2.1-es verzió, amely további javításokat hozott.
- Az Eclair legsikeresebb verziója a 2.1 lett, a 2.0 és a 2.0.1 minimálisan terjedt el, az ORA-tagok eszközeire csak 2.1 került fel.
- Néhány ismert eszköz: Samsung Galaxy Spica (GT-15700), Galaxy 3 (GT-15800), Galaxy S (GT-19000), Motorola Defy (Motorola MB525).[1]

2.1.6. Android 2.2 (Froyo)

- 2010 májusában jelent meg.
- Feljavított böngészője van: Flash 10.1 és akár háromszor gyorsabb JavaScript.
- JIT-támogatással rendelkezik, amely a CPU-igényes feladatokat 400-500 százalékkal gyorsíthatja.
- Stream és push támogatása.
- Ad hoc wifimegosztás.
- Teljesítménybeli és felületi javítások történtek.
- Az alkalmazások nagy részét a MicroSD-kártyára lehet másolni és ugyanígy vissza is helyezni.
- Hangalapú tárcsázás.
- Névjegymegosztás Bluetoothon keresztül.[1]

2.1.7. Android 2.3 (Gingerbread)

- 2010. december 6-án jelent meg.
- A Samsunggal közös Nexus S telefon.
- Új felhasználói interfésze van.
- Nagyobb felbontású kijelzőket támogat.
- 2.6.35.7 Linux-kernelt alkalmaz.
- Támogatja a WebM-videolejátszást.
- Near Field Communication (NFC) támogatása.
- Továbbfejlesztett másolás-beillesztés funkció.
- Átalakított gyári virtuális billentyűzet, multitouch támogatás.
- Javított energiagazdálkodás, hosszabb üzemidő.
- Optimalizáció (gyorsabb, hatékonyabb működés).
- Internethívás (VoIP) támogatása.
- Letöltéskezelő a hosszú ideig tartó HTTP-letöltésekhez.
- Új szenzorok (pl. giroszkóp) támogatása és kezelése.
- YAFFS helyett ext4-es fájlrendszer használata.[1]

2.1.8. Android 3.0 (Honeycomb)

- 2011 januárjában jelent meg.
- Táblagép-támogatással rendelkezik.
- Újragondolt felületet kapott.
- Táblagép PC-hez optimalizált kezelése van (pl. átalakított, megnövelt méretű virtuális billentyűzet).
- Többmagos processzorok támogatása.
- Teljes kompatibilitás a korábbi verziókra készült programokkal
- Fejlettebb szövegkijelölés, másolás-beillesztés.
- USB és Bluetooth külső billentyűzetének kezelése.
- Javított wifihálózat-keresés és Bluetooth-tethering.
- Felújított, kibővített gyári alkalmazások (böngésző, kamera, galéria, névjegyzék, e-mail).[1]

2.1.9. Android 3.1

- 2011 májusában jelent meg.
- Fejlettebb UI-effektek találhatók benne.
- Gyorsabb és látványosabb animációkkal rendelkezik.
- UI-elemek fejlesztése (szín, méret, kezelhetőség stb.).
- USB-eszközök támogatása (egér, billentyűzet, játékvezérlő, kamera stb.).
- Átméretezhető widgetek.
- Minden wifi access ponthoz külön HTTP-proxy-beállítás tartozik.
- Beépített alkalmazások fejlesztése.
- USB-host-API.
- Külső kameraintegráció (MTP -Media Transfer Protocol, PTP -
- Picture Transfer Protocol).
- RTP API (Real-time Transport Protocol): streaming támogatása.[1]

2.1.10. Android 3.2 (Ice Cream Sandwich)

- 2011 júliusában jelent meg.
- Kisebb frissítés.
- További optimalizáció táblagépek számára.
- Nagyítás támogatás kisebb kijelzőkre készített alkalmazások számára (iPhone-iPadhez hasonlóan).
- Media sync támogatása SD-kártyára.
- További támogatás a táblagép-UI fejlesztéséhez.[1]

2.1.11. Android 4.0 (Ice Cream Sandwich)

- 2011 októberében jelent meg.
- Új készenléti kijelzője, gyorsindítója és feladatkezelője van.
- Skálázható kezelőfelülettel rendelkezik.
- Az alkalmazások könnyen alkalmazkodhatnak az eltérő felbontású és fizikai méretű kijelzők adottságaihoz, amelyet az osztott képernyős megoldásokat támogató Fragments API is tovább segít.
- Az integrált arckövető megoldás révén a képernyőn megjelenő 3D-s alakzatok mindig a nézőnek megfelelő perspektívában jelennek meg.[1]

2.2. Feladat Specifikálása

A feladat szempontjából készíteni kell egy alkalmazást, ami képes külső forrásból kapott adatok alapján rendeléseket készíteni. Mobilon és/vagy tableten legyen elérhető. Kicsit bővebben:

A külső forrás egy szerver, ami az ottani rendszernek megfelelően előállított tömörített inputot bocsátja a rendelkezésünkre. Kicsomagolás után ebből a számunka fontos adatokat kell kinyerni és átalakítani a nekünk megfelelő bemeneti formára. Optimális esetben a szerver oldalon lehetne kialakítani egy olyan interfészt, ami a kliens által várt adatokat a megfelelő formában szolgáltatja. Csökkentve a kliens oldali adatfeldolgozási időt. Jelenleg a bemeneti adatok egy csomagolt állományban érkeznek, amiben megtalálhatók a partner- és cikktörzs adatok. Tehát a megvalósítandó cél ennek a csomagolt állománynak az átvétele kicsomagolása és a benne található szöveges állományokat feldolgozni. A szöveges állományok soronként tartalmazznak egy sort a nekik megfelelő táblából. A tervezés során figyelembe lehet venni ezt a tulajdonságot.

A szerver, a kapott szerkezethez nagyon hasonló formátumban várja a feldolgozandó rendeléseket. Egyetlen fájl vár, aminek szerkezete egy rendelés fejléc alatta a rendeléshez tartozó rekordok soronként. Ez ismétlődik, még van rendelés.

Miután megtörtént a feldolgozás a kliensnek rendelkeznie kell egy megtekintési felülettel, ahol az előzőleg beolvasott adatokat megtekintheti a felhasználó. Az egyes partnerek adatai és cikkek jellemzői megjelenítésre kerülnek.

A program rendeléskészítő modulja az egész legnagyobb részét adja. Az alap funkciók felett elvárt követelmények listáját az egyéni megrendelők adják. Tehát a felépítésnek olyannak kell lennie, hogy ezek elkülönüljenek, a későbbi módosítások könnyű kivitelezése érdekében.

Rendelkezzen egy általános felülettel, ahol a rendeléseket megtekinthetjük, jelenjenek meg alap információk a rendelésekről, a könnyebb beazonosíthatóság kedvéért. Tetszés szerint lehessen kiválasztani meglévő rendelést megtekintésre vagy módosításra.

Az új rendelés készítés első lépése a partner kiválasztás legyen. A partnerhez tartozó adatok közül a fizetés módja és a mozgáskód a felvételkor kerül kitöltésre. Lehessen megadni a rendeléshez megjegyzést. Kiválasztható legyen a teljesítés napja. Az új tétel felvitele után is legyen lehetőség annak adatait módosítani. A felületen legyen látható a rendelés mindenkor végösszege.

A felhasználónak legyen lehetősége egy adott dátumig, az addig felvett rendeléseket elküldeni a központnak. Végül nem utolsó sorban egy kilépés funkció, amely bezárja az alkalmazást.

2.2.1. Összefoglalva

Az alapvető funkciók, amiket a tervezett rendeléskészítő alkalmazásnak tudnia kell az alábbiak:

1. Az alkalmazásnak hordozhatónak kell lennie, ezért tablet vagy mobil eszközön kell megvalósítani.
2. Fogadni az adatokat tartalmazó tömörített állományt.
3. Kicsomagolni az állományt
4. Elkészíteni belőle egy könnyen hozzáférhető adattárat
5. Megjeleníteni az adattár tartalmát a felhasználónak megtekintésre.
6. Rendelések készítése, tárolása, későbbi felhasználás, módosítás tovább küldés céljával.
7. A rendeléshez tartozzon egy megjegyzés rész
8. Lehetőség a teljesítés dátumának megadására
9. A rendeléshez tartozó partner felvételnél a kiválasztás név alapján történjen.
10. A partner fizetés módjának és mozgáskódjának beállítási lehetőség.
11. A partner tiltott vagy örök készpénzes vagy az alapértelmezett fizetési módjától eltérővel készül a rendelés figyelmeztesse a felhasználót.

12. A tételek felvétele név alapját történjen.
13. A tétel vagy egész rendelés törlését csak megerősítés után végzi el.
14. A rendelés rendelkezzen végösszeggel
15. Az elkészített rendelésekből a szerver számára feldolgozható állomány előállítása.
16. Megadott dátumig készítse el a rendelésekből a szerver számára átadandó állományt
17. Lehetőség a kliens oldalon beállítani az ügynökkódot a források helyét.

2.3. Modulokra bontás

Az alkalmazás tervezés első lépése a feladat modulokra bontása. Ha végigkövetjük az alkalmazás egyes fázisait és lépéseit, akkor strukturáltan elkülöníthetők az egységek. Az alábbi részeket célszerű különválasztani:

- Fájl feldolgozó
- Adattárolás
- Adat megjelenítő
- Rendelés kezelő
- Eredmény fájl előállító
- Beállítások

Ezekben belül a rendelés készítő további kisebb részekre bontható.

- Rendelés partner felvevő
- Rendelési adat kitöltő
- Rendelési tétel felvevő
- Rendelési tétel módosítása
- Rendelés dátumválasztó

A szoftver tervezéséhez és fejlesztéséhez megközelítőleg ilyen szintű részekre bontás szükséges.

2.4. Tervezés

2.4.1. Adatbázis

Minden Android alkalmazás rendelkezik egy alap adatbázis kezelővel az SQLite-al. Nagyon gyors elérést tesz lehetővé az alkalmazások számára. Szintaktikája nagyban hasonlít az SQL szabványra. Manapság az adatok egyik legelterjedtebb tárolási formája relációs adatbázis. Egy meghatározott minta szerint tartalmazza az információt, amit bármikor megváltoztathatunk vagy lekérdezhetünk. Az Androidban megtalálható egy relációs adatbázis kezelő komponens az SQLite. Nyílt forráskódjának és kis méretének köszönhetően remekül alkalmazható kis erőforrású rendszerekben. C nyelven írták és lokálisan működik az eszközön, egy fájlban tárolódik ezért nincs szükség hálózati kapcsolatra. Az szoftver részeként funkcionál, működését az Android felügyeli, telepítést nem igényel.[2][3] Az SQLite (gyengén típusos) tárolási osztályai:

- NULL - ez a NULL érték
- TEXT - hasonló a Java String típusához
- INTEGER - hasonló a Java long típusához
- REAL - hasonló a Java double típusához
- BLOB - az inputtal egyezően tárolja az adatot[2][3]

Minden más adattípust előbb ezek valamelyikére kell átalakítani. Boolean - Integer 0 v 1. Nincs kimondott dátum vagy idő tárolására alkalmas típusa. Helyette a beépített dátum és idő függvények segítségével képes tárolni TEXT,REAL,INTEGER formátumban őket.[2][3]

- TEXT: az ISO8601 szabvány szerint karakterláncként:
YYYY-MM-DD HH:MM:SS.SSS
- REAL: A napok száma, ami eltelt időszámításunk előtt 4714 November 24 óta.
- INTEGER A Unix Time szerint tárolja a másodpercek számát, ami eltelt 1970.01.01.00:00:00 óta.[2][3]

Az általunk használni kívánt szövegek, számok, dátumok tárolására megfelelő.

A használatához szükséges két csomag neve: [2][3]

- android.database - általános adatbázis kezelő összetevőket tartalmaz
- android.database.sqlite - SQLite adatbázis használatához szükséges dolgokat tartalmazza[2][3]

A segéd osztályok:

- SQLiteOpenHelper - Ebből érdemes származtatni azt az osztályunkat, amely felelős lesz az adatbázis létrehozásáért. Az onCreate és onUpgrade metódusokat kell majd felüldefiniálnunk.

- SQLiteDatabase - Ennek segítségével tudunk majd dolgozni az SQLite adatbázissal. Az insert, delete, update metódusok mellett végrehajthatunk saját SQL utasításokat is a query, vagy a.rawQuery segítségével.[2][3]

Ezek felhasználásával kerül megtervezésre az adatbázis, ami az alkalmazás mögött helyezkedik el.

Szükség lesz az input állományoknak megfelelő táblák elkészítése. A tárolt rendeléseket tároló táblák megtervezése.

2.4.2. Felhasználói felületek

Az alkalmazás egy egészen egyszerű és letisztult felhasználói felülettel fog rendelkezni. A szoftver működéséhez az alábbi felületek kellenek:

- Főképernyő a funkciók gombjaival.
- Választó az adatbázis megjelenítőhöz.
- Partnerek adatai
- Cikkek adatai
- Rendeléseket megjelenítő felület.
- Partnerválasztó a rendeléshez
- A partner fizetismódjának és mozgáskódjának megadása
- A rendelés részleteit megjelenítő felület
- Dátumválasztó a rendeléshez
- Rendelést cikk kiválasztó
- A felvitt termék mennyiségének módosítása
- A felvitt termék vagy az egész rendelés törlését megerősítő ablak.
- Adatok mentése felület
- Beállítások képernyő

Ezek kezdetleges tervezete részben az előző céges szoftver kezelőfelületéből és a megrendelő által küldött tervezetek alapján készültek. Az új funkciók, melyek az előző szoftverben nem voltak benne, elemeinek elrendezésében a felhasználó barát elrendezés volt az irányadó.

Alkalmazás neve

Adatok betöltése

Adatok megjelenítése

Rendelések

Adatok mentése

Beállítások

Kilépés

(a) Főképernyő tervezet

Termék	Termék Leírás
Termék	
Termék	
Termék	
Termék	
Termék	
Termék	
Termék	
Termék	
Termék	
Termék	
Termék	

(b) Adatok megjelenítése

Új rendelés

Vissza

Rendelés

Rendelés

Rendelés

Rendelés

Rendelés

Rendelés

Rendelés

Rendelés

(c) Rendelések

2.4.3. Forrás feldolgozás

A partner- és cikktörzs csomagolt állományban érkezik egy előre azonosítható névvel. A fájl nevének szerkezete mindig utal a forrás és cél eszközre. Pl.:K-PDAD.zip Ebből látszik, hogy a központtól a PDA-ra, (a régi rendszert váltotta a mostani, előzőleg PDA-n működött) a dügynökkóddal érkezik. Kicsomagolás után az alábbi fájlokat tartalmazza:

- PARTNER.TXT
- RCIKK.TXT

Ezek az állományok soronként tartalmaznak egy-egy partner vagy cikk összes adatát. A sorok egy speciális karakterrel a `||` jellel vannak tagolva. A feldolgozás során ezt figyelembe lehet venni. A szöveges állományok feldolgozására a Java-ban két lehetőséget vizsgáltam meg:

- Scanner
- StringTokenizer

A termékek és partnerek megközelítőleg 6000 sort alkotnak. Tesztelésnek megírtam mindkét feldolgozóra, hogy a gyorsabbat használjam. A tesztekben az egész RCIKK.TXT-t százszor feldolgoztam és az abból kapott adatok a következők:

- Scanner : 118 ms
- StringTokenizer : 36 ms

Az eredményekből következik, hogy a StringTokenizer-t használom majd a feldolgozás során. A feldolgozott adatokat a nekik megfelelő táblákat egy rekordját reprezentáló osztályban célszerű tárolni, majd beilleszteni az adatbázisba.

A tesztek megtalálhatók a CD mellékleten

2.4.4. Adatbázis megjelenítő

Miután a feldolgozás végére az adatok bekerültek az adatbázisba. Szükséges lehet azok megtekintése, ezért erre egy külön felületen biztosítunk lehetőséget a felhasználónak. A megjelenítés előtt ki lehessen választani a megjelenítendő táblát (partner vagy cikk). Ezek után úgy kell megtervezni a megtekintő felületet, hogy mindkét táblát meg lehessen benne jeleníteni minimális változtatással (kód újra felhasználással). Az alap elképzelés egy lista megjelenítés minimális adatokkal a cikk vagy partnerről, amivel már beazonosítható. Mellette egy üres felület, amin kiválasztás után megjelennek a részletes leírások.

2.4.5. Rendeléskezelő

Ez a modul az alkalmazás legnagyobb és legfontosabb része, az egész központja. Először rendelkeznie kell egy alap felülettel, amin a már felvett rendelések listázódnak ki. Ha még nincs akkor üres. Egy új rendelés és a visszalépés gombot is kell készíteni. A már meglévő rendeléseket a listából kiválasztva tudjuk utólag módosítani. Újra felhasználhatóság szempontjából az új rendelés készítése és a rendelés módosítása rész ugyanazon a felületen is kivitelezhető. Ezért nincs szükség két külön felület kialakítására. A meglévő módosítása esetén annak adataival töltjük fel ugyanazt a felületet, amit akkor kapunk, amikor új rendelést készítünk.

Az új rendelés készítésekor a specifikáció szerint a partner kiválasztással kell kezdeni és annak mozgás és fizetémódjának beállításával. Figyelmeztetni kell a felhasználót, ha:

- a partner örökkészpénzes (azaz, minden rendelést készpénzfizetéssel rendez),
- esetleg tiltott (azaz, csak halasztott fizetési móddal lehet neki rendelést felvenni)
- vagy más fizetési móddal készül a rendelés, mint ami alapértelmezetten a partnerhez van rendelve.

A partner kiválasztása egy listából lehetséges, név alapján. Egy keresősáv áll rendelkezésre a név beírására, ami azonnal szűkíti a listát, ahogy beütünk egy újabb betűt. Az örökkészpénzeseknél nincs lehetőség fizetémód változtatására. A tiltott vásárlóknál csak a halasztott lehetőséget kínálja fel az alkalmazás. A partnerhez tartozó mozgáskód minden esetben az alapértelmezetten megadott "a" jelű legyen. A másik opció a "b" ami a cégen belüli árumozgást jelöli. Az ügynökök szoktak árut vinni egyik telephelyről a másikra. Az ilyen árukat egy "b" mozgáskódú rendelésbe teszik a partner pedig maga a cég. A partner kiválasztása után a rendelés felületre navigálunk. Itt megadható a rendeléshez tartozó egyéb módosítható adatok. Ami itt látszik:

- Rendelés azonosító, sémája utal az évre, ügynökkódra, rendeléssorszámra. (d4/R00001). Az újabb rendelésekkel a végén a szám rész növekszik. Az elmúlt évek tapasztalata szerint egy ügynök 100000 több rendelést nem készít évente. Módosítása nem lehetséges. a program automatikusan generálja.
- Rendelés teljesítési dátuma. Alapértelmezetten az aktuális napot állítja be. Nem lehetett általánosítani, előfordul a heti rendelés, másnapi vagy tetszőleges időpont. Felhasználóra bízunk a módosítást.
- Partner azonosítója és megnevezése. Később is módosítható a partner, ha rosszat választottunk ki. Előfordul azonos nevű partner más megnevezéssel (második név).
- Megjegyzés. Ez a mező a rendeléshez kapcsolódó egyéb információk megadására szolgál, egyben üzeni lehet a központban lévő adatrögzítőnek is.
- Végösszeg. A felvett tételek bruttó összegét számítja. Tagolja ezres egységekben és a "Ft" szöveget teszi a végére.

Ezekon kívül a megfelelő navigáló gombok:

- Ok (Rendelés felvétele, dátum és megjegyzés módosítások véglegesítése)

- Törlés (aktuális rendelés törlése)
- Vissza (Visszalépés, csak a dátum és megjegyzés módosítása vész el.)
- Újcikk. A rendeléshez új cikk adása.

Új cikk felvétele gomb lenyomásával megkezdődik a tétel hozzáadás. Az új felületen a cikkek listázva jelennek meg. Ugyanúgy, mint a partnerválasztásnál egy keresőben lehet szűkíteni a tételek listáját. Ahogy bevisszük a termék nevét betűnként, úgy csökken a lista mérete. Miután kiválasztottuk a terméket egy darabszámot kell megadni, aztán visszavigyalunk a rendelés felületre és látható a felvett rendelés a végösszeg ennek megfelelően módosul. Ezt ismételve tudunk újabb terméket felvinni a rendeléshez. A felvitt termékek rendelt mennyiségét utólag módosíthatjuk. Ha nem kell a termék a rendelésben, akkor törölhetjük. A termék módosításához csak ki kell választani a rendelési oldalon a listából. A módosító oldalon van lehetőség a törlésre.

2.4.6. Kimenet előállító

Hasonlóan a bemenethez itt is szöveges állomány lesz az adattovábbító. A neve az forrás fájl nevének fordítottja. Pl.:PDAD-K.TXT Tehát a tablet felől (PDA a régi rendszer miatt a feldolgozó ilyen állományra vár) megy a "K" központ felé és a "D" ügynök küldi. Benne a szerkezet mintája a kapott adatsémának. Kétféle sort különböztet meg.

- TR : A rendelés fejléce
- TETEL : A rendeléshez tartozó tétel

A TR sor leírja a rendelés jellemzőit (rendelés azonosító, megjegyzés, végösszeg, stb). Ha nem üres a rendelés, akkor ezt a tételek követik. TETEL kezdetű sorokkal. Ezek magukban foglalják a cikk adatait: cikkazonosító, rendelés azonosító, rendelt mennyiség, stb A sorokat itt is egy speciális karakter választja el: "|". Ezek a megkötések szerver oldalról érkeztek. A kimeneti fájl a forrás állományok mellett keletkezik. A megrendelő igényei szerint egy dátumot is meg lehet adni, hogy a rendelések meddig legyenek elküldve. Az első felvett rendeléstől a megadott dátumig az összesről elkészül a minta szerint a fájl. A megkapott adatokból a szerver csak a még fel nem vett rendeléseket rögzíti. Azért szükséges az összes adat küldése, mert a szerveren adatvesztés esetén így újra rögzítésre kerülnek a régebbi rendelések. Miután elkészül a kimeneti fájl azt levélhez csatolva vagy memória kártyára másolva lehet a központba eljuttatni.

2.4.7. Beállítások

Az alkalmazás rendeltetésszerű használatához először meg kell adni az ügynökkódot és a forrás állomány helyét. Az ügynökkódra a forrásfájl betöltése, továbbá a rendelés azonosítók és a kimeneti fájl előállításánál van szükség. A forrásállományok helyén keresi az input állományt és oda készíti a központnak szánt rendeléseket tartalmazó fájlt. Alapértelmezetten ne legyen ügynökkód megadva és a forrásfájlok helye egy előre megadott jegyzék legyen.

3. fejezet

Fejlesztői dokumentáció

3.1. Fejlesztés

3.1.1. Felületek

A felhasználói felületek az android fejlesztő környezetben teljesen elkülönülnek a program egyéb részeitől. Minden a képernyőn látható felület egyedi tervezés eredménye. A rajtuk található elemek (beviteli mező, megjelenítő mező, gomb, lista) mind saját leíróval rendelkezik, könnyen paraméterezhető a méret, hely, igazítás, stb. tekintetében. Felhasználva az android XML kifejezéseit könnyen és gyorsan lehet felhasználói felületeket tervezni. Minden layout-nak tartalmaznia kell egy gyöker elemet, aminek View vagy ViewGroup objektumnak kell lennie. Miután megadtuk a gyökérelemet hozzáadhatunk további layout objektumokat vagy widget-eket mint gyerek elemek. Például ez egy XML layout, vertikális LinearLayout felhasználásával, ami tartalmaz egy TextView (szövegmegjelenítő mező) és egy Button (gomb) elemet.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

3.1. ábra. LinearLayout példa

A tervezésnél feltüntetett lekészítendő felületek listájából pár érdekesebb. A rendeléskészítésnél a cikkek keresése a listában egy SearchView elem teszi lehetővé a bevitt adatok azonnali vizsgálatát gombnyomás nélkül, hogy a lista mérete csökken minden beírt betűvel.

Ugyanitt a rendelt mennyiséget már a felület szerkesztésekor lehet szűrni, hogy csak számok legyenek (plusz a "*" jel, de ez csak későbbi módosítás.).

A mellékletekben megtekinthetők az elkészült felhasználói felületek képei.

3.1.2. Adatbázis

Az alkalmazást körbeöleli egy Application, ami a legelső utasításokat tartalmazza a program indulásakor és a bezárásakor az utolsókat is. Amikor az alkalmazás elindul, kialakítja a kapcsolatot az adatbázissal és lezárja a kilépéskor. A megfelelő tároláshoz és gyorsasághoz 5 táblát alakítottam ki.

- cikktörzs (bejövő cikktörzs)
- partnertörzs (bejövő partnertörzs)
- rendelés (rendelések adatai)
- rendpartner (rendelési partnerek)
- rendcikk (rendelt cikkek)

A rendpartner és rendcikk táblák készítésekor még nem volt tisztázva, milyen adatok kellenek majd a forrás cikktörzs és partnertörzsből, értékeik közül melyeket kell majd megváltoztatni. A teljes másolásuk volt logikus a későbbi adatmódosítások így könnyen kivitelezhetők. (Arról is szó esett, hogy a rendelést készítő engedményt adhat az árakból egy előzetesen megbeszélte határok között. Nem volt tisztázva, hogy ez melyik árkategóriából számolandó. Végül ezek elmaradtak. Az első kiadott verzióig nem lett módosítva. Későbbi verzióban várható az adatbázis teljes átalakítása, hogy elkerüljük az adat többszörözést.) Az adatbázis az eszközön, saját területen található ahol idegen program nem tud hozzáférni. Ez a fájlszerkezetet tekintve /data/data/alkalmazás csomagnév/databases/adatbázis neve.db található.

A táblák szerkezete:

A cikktörzs

- id: elsődleges kulcs
- nev: A termék neve
- kod: A termék kódja
- egysegkod: A termék számolási egysége pl "db", "csomag"
- normalarnetto: Az alap eladási nettó ár
- normalarbrutto: Az alap eladási bruttó ár
- egyesarnetto : Az egyes árkategóriába tartozó partnerek eladási nettó ára
- kettesarnetto : A kettes árkategóriába tartozó partnerek eladási nettó ára
- kettesarbrutto : A kettes árkategóriába tartozó partnerek eladási bruttó ára
- harmasarnetto : A hármas árkategóriába tartozó partnerek eladási nettó ára
- negyesarnetto : A negyes árkategóriába tartozó partnerek eladási nettó ára

- otosarnetto : Az ötös árkategóriába tartozó partnerek eladási nettó ára
- otosarbrutto : Az ötös árkategóriába tartozó partnerek eladási bruttó ára
- beszerzesiarnetto : A termék beszerzési nettó ára
- afakod : A termék milyen áfakörbe tartozik
- egység_mennyiség : A termékből hány darab alkot egy egységet
- gongyoleg_kod : A terméket tároló rekesz kódja
- mennyiség : A termék raktár készlet mennyisége
- engedmeny: A termékre adható-e engedmény? "i" v "n"
- hatosarnetto : A hatos árkategóriába tartozó partnerek eladási nettó ára

A partner törzs

- id: elsődleges kulcs
- nev : partner elsődleges neve
- nev2 : partner másodlagos neve
- kod : partner kód
- irszam : irányítószám
- varos : cég székhelye
- utca_hsz : cég címe
- fizmod : fizetési mód
- esedekesseg : hány nap múlva érkezzen a rendelés
- arforma : árforma
- termekkedv : partner termék kedvezménye
- cigikedv : dohánytermék kedvezmény
- autotipus : szállító jármű típusa
- neszalltol : hány órától ne szállítsanak ki.
- nemszallig : meddig ne szállítsanak ki
- kps : örök készpénzes (i v n)
- tiltott : tiltott (i v n)

Rendelés tábla:

- id: elsődleges kulcs

- rendazon: rendelés azonosító
- osszar: végösszeg
- datum: mikorra dátum
- kelt: rendelés keltezése
- partnev: partner neve
- megjegyzes: megjegyzes

Az alkalmazás első elindításakor létrejönnek a táblák, és a program törléséig megőrzik a tartalmukat.

3.1.3. Adatfeldolgozó

Az adatok feldolgozásának menete:

- A beállításokban megadott ügynökkódot és forrásállomány helyet kiolvassuk.
- Ellenőrizzük, hogy létezik-e a megadott ügynökkódhoz tömörített állomány a megadott forrás állomány jegyzékben.
- Amennyiben megtaláltuk az állományt, kicsomagoljuk a források helyére.
- Megnézzük hány sorból áll a két forrásfájl (a feldolgozottság mutatóhoz kell)
- Ezután egy neki megfelelő AsyncTask meghívásával elkezdjük a feldolgozást.
- A feldolgozottság aktuális állását mindig mutatjuk a felhasználónak. ehhez egy ProgressDialog-t használunk.
- A feldolgozás végén eltávolítjuk a kicsomagolt fájlokat.

Részletesebben:

Felhasznált technológiák, csomagok:

- java.io.LineNumberReader - (input fájlok sorainak száma)
- java.util.StringTokenizer - (input fájlok darabolása)
- java.util.zip.ZipEntry (csomagolt állományban található fájlok)
- java.util.zip.ZipInputStream (csomagolt állomány feldolgozó)
- ProgressDialog (A feldolgozottság mutatója)
- AsyncTask (A feldolgozás ne akassza meg az alkalmazást, mutatható legyen a feldolgozottság.)

A ProgressDialog jelenik meg a képernyőn a feldolgozás ideje alatt. Ez tájékoztatja a felhasználót a feldolgozottségról. Nem lehet eltüntetni a felületről. Amíg a feldolgozás tart addig ez van előtérben.

Ez a programkód részlet mutatja a tömörített állomány kicsomagolásáért felelős komponenst.

```
try {
    FileInputStream = new FileInputStream(srcfile);
    ZipInputStream zipInputStream = new ZipInputStream(
        new BufferedInputStream(fileInputStream));
    ZipEntry zipEntry;
    while ((zipEntry = zipInputStream.getNextEntry()) != null) {
        String zipEntryName = zipEntry.getName();
        File file = new File(myDir + '/' + zipEntryName);
        if (file.exists())
            file.delete();
        if (zipEntry.isDirectory()) {
            file.mkdirs();
        } else {
            byte buffer[] = new byte[BUFFER_SIZE];
            FileOutputStream fileOutputStream = new FileOutputStream(
                file);
            bufferedOutputStream = new BufferedOutputStream(
                fileOutputStream, BUFFER_SIZE);
            int count;
            while ((count = zipInputStream.read(buffer, 0,
                BUFFER_SIZE)) != -1) {
                bufferedOutputStream.write(buffer, 0, count);
            }
            bufferedOutputStream.flush();
            bufferedOutputStream.close();
        }
    }
    zipInputStream.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
```

A forrásállomány-t megkapja egy `FileInputStream` ez képezi a bemenetet a `ZipInputStream`-nek. Létrehozunk egy `ZipEntry`-t, ami a töritett állományban található soron következő tartalom értékét veszi fel, egészen addig amíg a forrás nem lesz üres. Ezután a soron következő állományokat rendre létrehozzuk, ügyelve arra, hogy fájl vagy jegyzék. Végül ürítjük és zárjuk a bemenetet.

3.1.4. Adatbázis megjelenítő

Az alapötlet szerinte készítettem egy választóképernyőt az adatbázisok kiválasztásához. Majd a megfelelő kiválasztása után megjelenik egy képernyő, ami tartalmaz egy `ViewGroup`-t amiben van egy `Listfragment`. A lista egy sora az adatbázisban egy bejegyzés. Kiválasztva egy elemet mellette megjelenik annak minden adata. Az eszköz vissza gombjával tudunk visszatérni az adatbázis választó képernyőre.

3.1.5. Rendelés kezelő

A rendelési folyamat általános menete a következő:

- Az új rendelés gombra kattintva az alkalmazás legenerálja a rendelés azonosítót. Ehhez megnézi hány sorból áll eddig a rendelések tábla. Ha van már benne rendelés, akkor lekérdezi az utolsó azonosítóját és hozzáad egyet. Különben előállítja az első azonosítót. Szükséges tudni az évet és az ügynökkódot. Ha ezek rendben megvannak, akkor visszaadja a soron következő azonosítót (ez lehet az első is).
- Megtörténik a rendelés felvétele az adatbázisba az azonosítóval, keltezéssel, nullás végösszeggel, stb.
- Felveszi a rendelési partnert a megfelelő rendelési azonosítóval.
- Tovább lépés a rendelés felületre.
- A rendelés felület egyből tovább navigál a partnerfelvételre, ha ez egy új rendelés (nem meglévő módosítása)
- A partner felvétel előkeresi a meglévő partnert, amit a rendeléskészítésnél létrehoztunk a rendeléshez (üres partner)
- Ennek adataival feltölti a rendelési partner felület mezőit. (jelenleg ezek üresek lesznek.)
- A felhasználó a kereső sornak köszönhetően gyorsan ki tudja keresni a partnert. Kiválasztás után a megfelelő, akkor hozzá lehet adni a partnert a rendeléshez. (Nem tiltott, kiválasztottunk partnert.)
- A partner hozzáadás után visszavigáz a rendelés felületre. Innentől érvényes rendeléssel van dolgunk. A megegyezés szerint a partnerválasztás után már tárolni kell a rendelést, még ha üres is a terméklista.
- A rendelés felületen bármikor van lehetőség a partnermódosításra, csak megérintjük a kijelzőn a nevét és visszavigáz a partnerválasztó felületre.
- Ha a legelején nem választunk partnert (üres partner marad), hanem az eszköz vissza gombjával visszalépünk a rendelés felületre majd a vissza gombbal a rendelések listáig. Ebben az esetben törlődik a felvett rendelés a hozzá tartozó üres partnerrel. Új rendelés esetén újra generálja a fentebbi lépések szerint.
- Amikor a rendelés felületen vagyunk és rendelkezőnk megadott partnerrel. megkezdhetjük a tételek hozzáadását a rendeléshez.
- Az újcikk gombbal a rendelkezésre álló árucikkek közül tudunk választani.
- Az itt található kereső mező a partner hozzáadása felülethez hasonlóan, ahogy gépeljük, a termék nevét azonnal szűkíti a listát.
- A gyors tétel felvétel érdekében, ha kiválasztunk egy árucikket azonnal a mennyiség mezőre vált és bevihető a rendelni kívánt mennyiség.

- Miután a rendben gombbal felvettük a tételt visszatérünk a rendelés felületre, ahol a rendeléshez tartozó tétellistán megjelenik. A végösszeg ennek megfelelően változik.
- A tétel mennyiségi módosítására van lehetőség. Ha magát a terméket akarjuk kicserélni, a meglévőt eltávolítjuk (törlés) aztán a fent leírt lépésekkel felvesszük az újat.

A rendelések listából kiválasztva meglévő rendelést szintén a rendelések felületre visz az alkalmazás, ahol a rendelésben foglalt adatokkal (partner, megjegyzés, dátum, végösszeg, tételek).

Végösszeg számítása: A rendeléshez tartozó tételek listázása, négy féle áfa kategóriát különböztetünk meg a termékeknél:

- Áfa körön kívüli: 0
- Nullás kóddal rendelkezők: 5
- Egyes áfa kör: 18
- Kettes áfa kör: 27

Az áfák szerint szétválogatva összeadjuk a termék nettóárát szorozva a rendelt mennyiséggel. Majd a végén az egészre számoljuk az áfát. Ezt mind a négy kategóriában megcsináljuk, majd összeadjuk a bruttó értékeket és ebből áll össze a végösszeg.

3.1.6. Rendelések mentése

A kimeneti fájl előállításához szükség van egy dátumra a felhasználótól. Ezt a dátumot egy dátumválasztó segítségével tudja megadni. Alapértelmezetten a mindenkor dátum kerül beállításra.

A rendelések, rendelt cikkek és rendelő partnerekből listákat állítok elő. A rendeléseken végighaladva végignézem melyiknek a keltezés dátuma kisebb, mint a megadott dátum. A rendelés hozzá tartozó partner adataiból elkészítem a rendelés fejlécét. A rendeléshez tartozó rendelt cikkek listájából elkészülnek a tétel sorok.

A fájl elkészítéséhez a PrintWriter osztályt használom.

A TR rendelés fejrész az alábbiakat tartalmazza a rendelésről:

- rendelés azonosító
- partner kód
- rendelés dátum (keltezés)
- partner mozgáskód
- partner fizetésmód
- rendelés megjegyzés
- rendelés végösszeg
- partner autótípus

- partnernek mikortól nem lehet szállítani
- partnernek meddig nem lehet szállítani

A rendeléshez tartozó egy tétel sora a fájlban:

- rendelés azonosító
- rendelt termék kódja
- rendelt termék neve
- rendelt mennyiség
- rendelt termék egységkódja
- rendelt termék áfa kódja
- rendelt termék egységmennyisége
- rendelt termék nettó ára


```
try {
    PrintWriter pw = new PrintWriter(file);
    for (Rendeles rend : rendelesek) {
        if (rend.getKelt() < szam) {
            for (RendPartner part : rendpartnerek) {
                if (rend.getRendazon().equals(part.getRendazon())) {
                    pw.print("TR|" + rend.getRendazon() + "|" + " |"
                        + part.getKod() + "|"
                        + rend.datum() + "|"
                        + part.getMozgkod() + "|"
                        + part.getFizmod() + "|" + " |" + " |"
                        + rend.getMegjegyzes() + " |"
                        + rend.getOsszar() + "|"
                        + part.getAutotipus() + " |"
                        + part.getNeszalltol() + "|"
                        + part.getNemszallig() + "|"
                        + "0" + "\r\n");
                    for (RendCikk cikk : rendcikkek) {
                        if (rend.getRendazon().equals(
                            cikk.getRendazon())) {
                            pw.print("TETEL|" + cikk.getRendazon()
                                + "|" + cikk.getKod() + "|"
                                + cikk.getNev() + "|"
                                + cikk.getRendmennyiseg() + "|"
                                + cikk.getEgysegkod() + "|"
                                + cikk.getAfakod() + "|"
                                + cikk.getEgyseg_mennyiseg() + "|"
                                + cikk.getNormalarnetto()
                                + "\r\n");
                        }
                    }
                }
            }
        }
    }
    pw.flush();
    pw.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

Az elkészült szöveges állomány a

3.1.7. Beállítások

A program első elindításakor a beállításokban meg kell adni az ügynökkódot ami, azonosítja a felhasználót. A kód alapján lesz kiválasztva a tömörített állomány. A forrás állomány megadásánál a van alapértelmezett elérési út. Ide célszerű másolni a tömörített állományt, de ettől eltérő helyre is tehető, csak pontosan meg kell adni jegyzék

címét.

Az ügynökkód egyetlen betűből áll, a megadásnál nem kell figyelni, hogy kis- vagy nagybetűt adunk meg. A felhasználó azonosítója kisbetűként tárolódik.

Miután megadtuk a helyes adatokat a mentés gombbal visszatérhetünk a főképernyőre és megkezdhetjük az adatok beolvasását.

3.2. Tesztelés

Az alkalmazást a program írása közben teszteltem.

Az alábbi eseteket vizsgáltam:

- Ügynökkód és forrásállomány megfelelően tárolódik és kiolvasható.
- A kezdősorszám megadás helyessége.
- Kezdősorszám megadása után ténylegesen azzal hoz létre rendelőt.
- Helyes input esetén a beolvasás zavartalan
- Beolvasott adatok mind megjelennek.
- A felvitt rendelések biztos tárolása és visszaolvasása
- Rendelés azonosító generálása
- Rendelés részletek megfelelő tárolása
- Végösszeg pontos értéket mutat
- Tétel felvétel, módosítás, törlés
- Kimeneti szöveges állomány megfelelően elkészül.

A saját tesztelés után megkértem programozó barátomat, kollégáimat teszteljék az alkalmazást. A visszajelzések alapján kiadtam tesztelésre a megbízó cégnek ahol az alábbi módosításokat kérték.

3.3. Módosítások

Az első elkészült tesztelt verzió után hamar érkeztek az első módosítási igények. A megrendelőnek bármikor, bármilyen módosítási igénye megjelenhet a használat során kialakuló igény szerint. Olyan dolgok, amik a specifikációban nem szerepeltek vagy teljesen másként.

3.3.1. Első módosítás

Az első módosítás tulajdonképpen egy új funkció kérése volt. Az igény megjelenése: Mi történik akkor, ha a használt tablet munka során megsérül és elvesznek róla az adatok?

Az új tablet amit beállítanak rendeltetés szerűen az első sorszámmal kezdi majd az új rendelések felvételét. Beküldésnél a szerveren már létezik, azzal a sorszámmal rendelés ezért azt nem veszi fel. A megoldás egy kezdő sorszám beállításának lehetősége a beállításokba, ha az alapértelmezettől eltérőt szeretnénk.

A módosítások csak a beállítások felületét érintik egy új mezővel, ahol megadható egy sorszám. A lehetséges hibák:

- Nulla kezdeti sorszám (3.7. ábra)
- utolsótól kisebb (3.8. ábra)

Ezt ellenőrizve és eltávolítva az új rendelés ezzel a sorszámmal fog rendelkezni a felvételkor. (3.9. ábra)

3.3.2. Második módosítás

Az első módosítás után nem sokkal érkezett egy újabb igény. Ez tulajdonképpen egy figyelmeztető funkció a felhasználónak, amivel csökkenteni a későbbi egyeztetések számát.

Az italokat könnyebben és biztonságosabban tudják szállítani, ha azok teljes rekeszekben vannak. A nem teljes rekeszek elkerülése érdekében szeretnék volna, ha a felhasználó értesítést kap, amikor nem teljes rekesz mennyiséget vesz fel a rendelésben.

Pl.: 8 az egység mennyiség és a rendelésben 2 db vesz fel a felhasználó, akkor figyelmeztesse, hogy nem teljes rekesz. A felhasználónak a felvételkor legyen lehetősége a termékre jellemző rekeszmennyiséggel megadni nem cikk darabszámot, hanem rekesz mennyiséget.

Pl.: 6 az egység mennyiség a megrendelőnek 8 rekesz kell. A felhasználónak ne kelljen fejből tudni az egységmennyiséget vagy kikeresni, ezért ha a "*" jelet (3.10. ábra) a mennyiség elé teszi, a program kiszámolja azt. (3.11. ábra)

3.4. Továbbfejlesztési lehetőségek

Az alkalmazásban rejlő további lehetőségek a megrendelő igényeihez alakíthatók, melyek az alábbiak lehetnek:

- A bemenő állomány fájtípusa lehet XML vagy JSON a könnyebb, gyorsabb feldolgozás érdekében. Szabványosítás.
- A bemenő állomány érkezhethetne egyenesen a szerverről interneten keresztül.
- A cikktörzs legyen mindig aktuális, ne csak napi egy frissítés.
- Az alkalmazás elején bejelentkező felület kialakítása.

- Profil kialakítása
- Az adatbázis megjelenítőben kereső felület kialakítása mind a cikkek és partnerek részben.
- Az aktuális raktárkészlet megjelenítése.
- Az adatbázis újra tervezése

3.5. Felhasználói dokumentáció

Az apk telepítése után az alkalmazás első indításakor a főképernyő (3.2. ábra) fogadja a felhasználót. Az egyes menüpontok használata előtt az alkalmazás beállításával kell kezdeni. Ezt megtaláljuk a Beállítások (kék) gombbal érhetjük el.

Itt megadhatjuk az ügynökkódot, ami belekerül a rendelés azonosítóba és a szerver ez alapján azonosítja a rendelés készítőjét. Alatta a tömörített forrásállomány elérési útja, az első elinduláskor egy alapértelmezés szerint a belső gyökéren belül a bora jegyzékre mutat (létrehozza). A kezdő sorszám üres. Mindaddig nem kell módosítani, amíg a soron következő rendelés azonosítótól eltérőt nem szeretnénk a következő rendelésnek. Ezt úgy érjük el, hogy megadjuk a következő rendelés sorszámot, aztán a következő rendelés az adott sorszámot kapja, ha ez lehetséges. A Rendben feliratú gombbal véglegesíthetjük a beállításokat. Ezután visszakerülünk a főképernyőre. A Vissza gomb használatával a módosítások nem kerülnek elmentésre és ismét a főképernyőre térünk vissza.

Ezzel beállítottuk alkalmazásunkat. Megkezdhetjük az adatok betöltését(3.3. ábra). Az Adatok betöltése (türkiz) gombra nyomva, ha jól adtuk meg a csomagolt állomány helyét megkezdődik a háttér adatbázis feltöltése. A folyamat 1-2 percet vesz igénybe. Mivel ezt a műveletet naponta csak egyszer végezzük, nem okoz gondot, hogy hosszabb ideig tart. A folyamat a kijelző érintésével nem szakítható meg. A folyamat végeztével egy Elkészült felirat jelenik meg a kijelző alsó részén egy rövidebb ideig.

Az adatok betöltése után lehetőség van megtekinteni a felvett adatokat az Adatbázis (sárga) gombbal. Ezután egy választó képernyő következik, ahol kiválaszthatjuk a partner(3.4. ábra) vagy a cikk(3.5. ábra) törzset megtekintésre. Egy listában megjelennek az alkalmazás által tárolt tételek, partnerek. Bármelyiket kiválasztva jobb oldalt részletes információt kapunk róla. Visszalépni a választó képernyőre az eszköz vissza gombjával lehetséges. A választóképernyőről a kijelzőn található vissza gombbal tudunk újra a főképernyőre navigálni.

A rendelések gombbal a már meglévő rendeléseket listázhatjuk ki(3.6. ábra), ha már voltak. Az új rendelés felvétele az Új rendelés(zöld) után kezdődik. A partner felvétellel kezdődik. A felületen ábra begépelhető a partner neve, ami a gépeléssel azonnal csökken. A listáról kiválasztható a partner és utána megadható a fizetésmód és a mozgáskód. A fizetésmód a partnernek megadott alapértelmezett. Ettől eltérőt kiválasztva a program figyelmeztet. A választható lehetőségek: Készpénzes, Halasztott, Utalásos. A partner lehet tiltott, ha nem teljesítette a megelőző utalásos rendelést. Ha tiltott, akkor csak halasztással vehető fel neki rendelés, de erre figyelmeztet az alkalmazás.

Az örökkészpénzes partnereknek csak készpénzes rendelés készíthető. A felhasználó figyelmeztetést kap az ilyen partnerről is. Amint a megfelelő partnert válasszuk, ki a Rendben gombbal lehet felvenni. Ezután megjelenik a rendelés részletei.

Legfelül négy gomb található, balról, jobbra a következők:

- Ok
- Törlés
- Vissza
- Újcikk

Alatta a rendelés azonosító található, amit a rendszer automatikusan generál az ügynökkódból és az aktuális évből. A soron következő rendelés azonosító az előzőből származik, ha még nincs rendelés, akkor a nullás sorszámot kapja. Mellette a rendelés teljesítésének dátuma található. Itt adható meg, hogy mikorra szállítsák a termékeket. A következő sorban az előzőleg kiválasztott partner azonosítója és neve jelenik meg. Itt megváltoztatható a partner, ha véletlenül rosszat választottunk ki. A rendeléshez tartozó megjegyzés megadható alatta.

A rendelés végső részlete a végösszeg, amit az alkalmazás számol ki. Egy tétellista mutatja a rendeléshez tartozó cikkeket. Az új cikk felvétele gombbal a partner felvételhez hasonló. Egy kereső sávba be lehet gépelni a keresett cikk nevét, a lista a bevitt karakterekkel csökken. A kiválasztás után azonnal a rendelt mennyiséget lehet megadni. A mennyiségnél megadható a darabszám vagy akár rekesz darabszám is, ehhez a csillag karaktert kell a mennyiség előtt megadni. A program az egység mennyiség alapján kiszámolja a rendelt mennyiséget. A felvétel után a rendelés képernyőre térünk vissza, figyelmeztetést kapunk, ha nem teljes a rekeszmennyiség. A felvett cikk a mennyiséggel együtt a tétellistához adódik, módosítva a végösszeget. A felvett áru mennyiségét van lehetőség módosítani ha a tétellistán kiválasztjuk. A megjelenő új felületen a beviteli mezőben megadhatunk darab vagy rekesz mennyiséget, ugyan itt tudjuk törölni is a listáról.

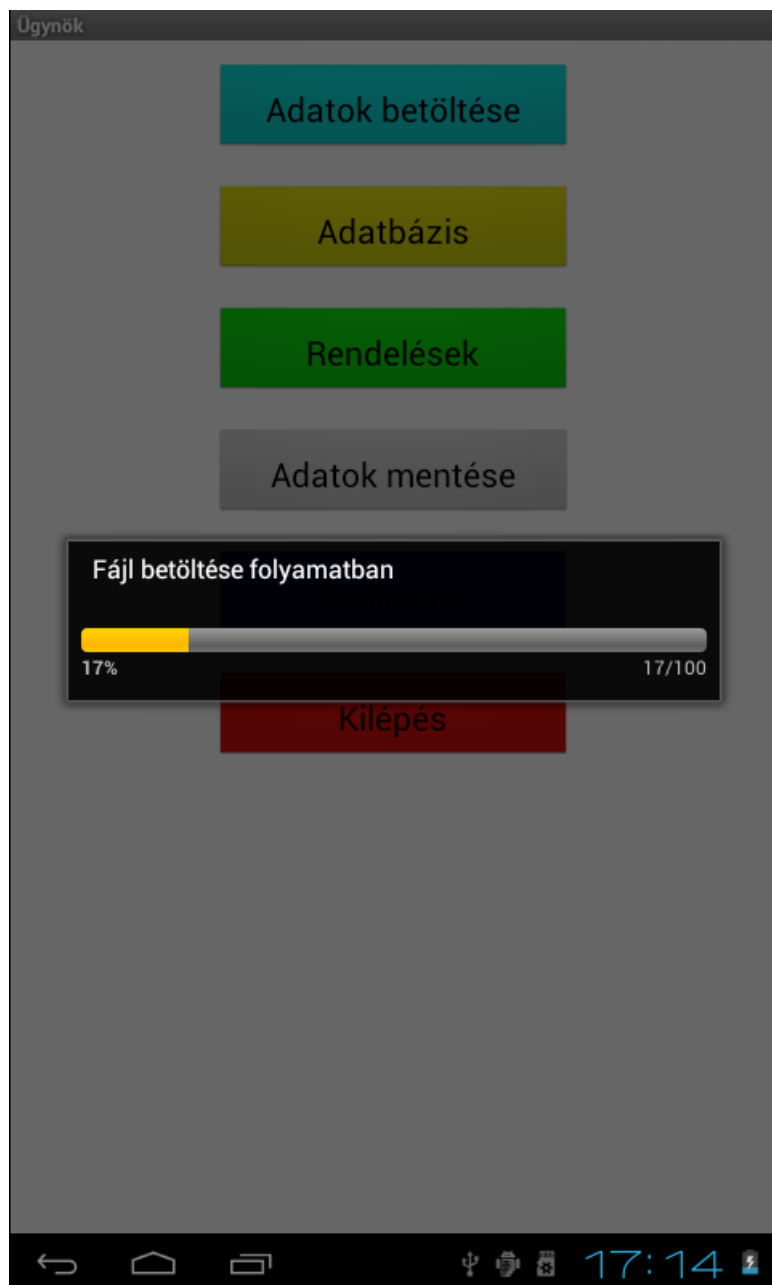
Miután végeztünk a tételek felvételével a rendelés felületen az oké gombbal visszatérhetünk a rendelések listájához. Később kiválasztva a kívánt rendelést azt módosíthatjuk vagy akár törölhetjük is. Fontos, hogy a sorszámgenerálás a mindenkor utolsó utánit adja az új rendelés azonosítójának.

A rendelések felvétele után visszatérve a fő menübe az adatok mentésével tudjuk elkészíteni a szerver számára a szöveges állományt. A mentés készítésekor megadható egy dátum, hogy meddig mentsük a rendeléseket.

3.6. Mellékletek



3.2. ábra. Főképernyő



3.3. ábra. Adatok betöltése

Partnerek	
8574"TIMUR és CSAPATA" Ker. és Szolg Bt	Név: ALI TRADE KFT Alnév: Sörkert Azon: 4088 Irszám: 5666 Varos: Medgyesegyháza Utca_hsz: Kossuth tér 25. Fizmod: H Esedékesség: 14 Árforma: Termekkedv: 0.0 Cigikedv: 0.0 Autotipus: Nemszállítható tol: 0 Nemszállítható ig: 0 Készpénzes: - Tiltott:-
-SPORT Büfé 5600Békéscsaba Árpád sor 3.	
2405Agrobert Kft. Monaco Söröző 5600Békéscsaba Trefort u. 13-17.	
40178Agrobert Kft. (doh.) Monaco Söröző 5600Békéscsaba Trefort u. 13-17.	
4088ALI TRADE KFT Sörkert 5666Medgyesegyháza Kossuth tér 25.	
9360Aranykonyha Kkt. 5600Békéscsaba Lencsési út 136.	
12666Autó Gumi Global Kft Vietnam söröző 5451Öcsöd Veresházi u.1.	
935Autó Gumi Trend Kft. Sztár Cukrászda 5540Szarvas Szabadság út 6-10.	
164Barna És Barna Kft. Speed Piza 5600Békéscsaba Jusst Gyula 1.	
9449Barna És Barna Kft. Speed Burger 5600Békéscsaba Jusst Gyula 1.	
3896Bartolák Ildikó Forintos Bolt 5674Kétsoprony Ady u. 1.	
855Bába Jánosné Élelmiszer Bolt 488-318	

3.4. ábra. Partner törzs

Árucikkek	
3BIT szelet 41gr	Név: Absinthe Pere
Cikk azonosító: M100171 Bruttóár: 94 Ft	Kermann's 0.7 L 60%
7 days Crois.Midi Kakaó 60gr	Azonosító: F518160
Cikk azonosító: M100210 Bruttóár: 87 Ft	Egység kód: db
7-Up Pet 0.5 L	Normál ár nettó: 3936.22
Cikk azonosító: B001130 Bruttóár: 189 Ft	Normál ár bruttó: 4999.0
7-Up Pet 2.5 L 1/8	Ár1 nettó: 4290.55
Cikk azonosító: B001230 Bruttóár: 315 Ft	Ár2 nettó: 3936.22
A.Tokaji Aszu éd.f.borkül 3p 0.5 L	Ár2 bruttó: 4999.0
Cikk azonosító: D900176 Bruttóár: 1180 Ft	Ár3 nettó: 4408.66
A.Tokaji Szamorodni éd.2009 0.5 L	Ár4 nettó: 4093.7
Cikk azonosító: D900177 Bruttóár: 795 Ft	Ár5 nettó: 3936.22
Ablaktisztító 500ml Privát	Ár5 bruttó: 4999.0
Cikk azonosító: M600052 Bruttóár: 195 Ft	Beszerezési ár netto: 3608.0
Absinthe Pere Kermann's 0.7 L 60%	Áfa kód: 2
Cikk azonosító: F518160 Bruttóár: 4999 Ft	Egység mennyiség: 6
Absinthe Rod.Black 70% 0.7L	Göngyöleg_kód: Z603006
Cikk azonosító: F034162 Bruttóár: 4990 Ft	Mennyiség: 0
Absinthe Rod.Bohème 60% 0.7L	Engedmény: i
Cikk azonosító: F034163 Bruttóár: 4190 Ft	Ár6nettó:3860.56
Absinthe Rod.Cannabis 70% 0.7L	
Cikk azonosító: F034161 Bruttóár: 4890 Ft	
Absinthe Rod.Strong 85% 0.7L	
Cikk azonosító: F034160 Bruttóár: 4603 Ft	
Absinthe Trenet 0.7 L 60%	
Cikk azonosító: F517160 Bruttóár: 5755 Ft	
Absinthe Tunel Black 80% 0.7L	
Cikk azonosító: F583162 Bruttóár: 6700 Ft	

3.5. ábra. Cikk törzs



3.6. ábra. Rendelések

Beállítások

Ügynökkód:

Forrás fájllok:

Kezdő sorszám:

3.7. ábra. Nullától nagyobb kezdeti sorszám

Beállítások

Ügynökkód:

Forrás fájllok:

Kezdő sorszám:

Az utolsó sorszámtól nagyobb kell

3.8. ábra. Utolsótól nagyobb sorszám

Beállítások

Ügynökkód:

Forrás fájl:

Kezdő sorszám:

3.9. ábra. Helyes kezdeti sorszám

Rendelések

Módosít

Vissza

Törlés

Cikk: Gacsó Egri Kékfrankos SZ 2L
Egységmennyiség: 6db
Aktuális mennyiség: 3db
Módosított mennyiség: *7 db

1	2	3	✕
4	5	6	-
7	8	9	+
*	0	.	↩

18:14

3.10. ábra. Rekesz mennyiség megadása

Rendelések

OK Törlés Vissza Újcikk

Rendazon: d4/R00001 Mikorra: 2014. 11. 3.

Partner: 40178 Agrobot Kft. (doh.)

Megjegyzés: Megjegyzés

Végösszeg: 25 902 Ft

Saga erdei gyüm.ízű tea
Mennyiség: 8 db Bruttó ár: 219.0 Ft Tétel ár: 1752.0Ft

Gacsó Egri Kékfrankos SZ 2L
Mennyiség: 42 db Bruttó ár: 575.0 Ft Tétel ár: 24150.0Ft

← ↻ 📱 18:16

3.11. ábra. Kiszámított mennyiség

4. fejezet

Összefoglalás

4.1. Összegzés

Ezelőtt még nem végeztem ilyen komplex program tervezését, fejlesztését, így ez a feladat kellő kihívást jelentett. Ahogy lépésről lépésre haladt a tervezés úgy értettem meg, hogy egy ilyen alkalmazás részekre bontás után akár párhuzamosan is fejleszthető. Voltak részei, amik jobban, könnyebben mentek (adatfeldolgozás), némelyik nehezebben (felhasználói felület).

A feladat kiírás szerint egy általános rendeléskészítő alkalmazást készítettem amit a megbízó módosításai szerint alakítottam át. Sokat tanultam a mobilalkalmazás fejlesztésről, a felhasználói felületek tervezéséről és az Android adattárolási lehetőségeiről. Megismertem a rendelés készítésétől a teljesítésig végbemenő folyamatokat, egy nagyvállalat működésében.

A megrendelő elégedett volt az elkészült alkalmazással. Hasznosnak ítélték a felhasználói dokumentációt. Bizalommal kérnek segítséget vagy további módosításokat. A további fejlesztési lehetőségben említett irányzatok szerint további funkciók implementálhatók bele. Cél egy közel univerzális termék előállítása kis- és közép vállalkozók részére. Úgy érzem a kitűzött célt elértem az alkalmazást használatba tudták venni. Jelenleg megközelítőleg 3-6 felhasználó használja napi rendszerességgel munkája végzése során.

4.2. Summary

I have never done the development and the coding of such a complex programme, thus this task has been a good challenge to me. While I was going on step by step improving and designing this application, I realized that such a programme can be divided into parts and developed simultaneously. There were parts which were easy to deal with, as data processing and also parts which were harder, for example, user interface.

According to the notice of set task I have prepared a universal order drafter application, and it was modified according to the clients' lights. I have learnt a lot about developing mobile phone applications, creating user interfaces and about the possibilities of Android's data storage. I have become familiar with the processes of a big entrepreneurship from order preparations to the last stages of fulfilment.

The customers were satisfied with the final form of the application. They have found the software documentation useful. They may ask for help or further modifications based on mutual trust.

According to the tendencies above mentioned in the possibilities of further development, new functions can be implemented. The purpose is to create a universal product for retailers and bigger entrepreneurships as well. I think, I achieved the desired goal and the application could have been put in practise. At the moment approximately 3-6 users use it at work on a daily basis.

Irodalomjegyzék

- [1] Ekler Péter, Fehér Marcell, Forstner Bertalan, Kelényi Imre: Android-alapú szoftverfejlesztés, Szak Kiadó, Veszprém, 2012.
- [2] Datatypes in SQLite Version 3,
<http://www.sqlite.org/datatype3.html>
- [3] SQLite - alapok, <http://codeguide.hu/2013/07/25/sqlite-alapok/>
- [4] Az Android platform bemutatása,
http://www.szak.hu/android-fejl/sample_chapters/chap1.pdf
- [5] Bora cég logó, <http://borakft.hu/index.php>
- [6] Felületek,
<http://developer.android.com/guide/topics/ui/declaring-layout.html>
- [7] Fehér Krisztián: Androidos szoftverfejlesztés alapfokon, BBS-INFO KÖNYVK. ÉS INFORM. KFT., 2014.
- [8] Kövesdán Gábor: Szoftverfejlesztés JavaSE platformon, Szak Kiadó, 2014.

Adathordozó használati útmutató

Az Eclipse fejlesztő környezetet használtam, ami ingyenesen letölthető a

<https://www.eclipse.org/downloads/>

linkről.

A Google által kiadott Android SDK plugin könnyen integrálható bele, de az android fejlesztői oldaláról letölthető egy hordozható verzió, amiben az Eclipse és az Android SDK együtt megtalálható. Ez letölthető az alábbi linkről:

<http://developer.android.com/sdk/index.html>

Középen kék felületen a letöltési lehetőség. Elfogadva az általános szerződési feltételeket és kiválasztva az operációs rendszer fajtáját (32 vagy 64 bit) elkezdődik a letöltés (354MB). Ez a kapcsolat gyorsaságától függően eltarthat néhány percig. A tömörített állományt kicsomagoljuk egy szabadon választott helyre. Benne található:

- Eclipse
- Android SDK

Az Eclipse a jegyzékben található eclipse.exe fájlal indítható. Elindításkor meg kell adni a programnak egy jegyzéket, ahova a projectek készüljenek. Itt két lehetőség felé lehet továbbhaladni.

- Emulátor készítése (lassú, akadozik, hosszas az elkészítés, a fejlesztők sem ajánlják használatát)
- Fejlesztő eszköz használata (gyors, nem igényel telepítést, fejlesztők is ezt használják)

Az első lehetőséget használva az alábbi lépéseket kell elvégezni az alkalmazás fordításához és futtatásához:

Az Eclipse-ben a Window menüpont alatt található Android SDK Manager és Android Virtual Device Manager.

Az Android SDK Manager megnyitása után az aktuálisan elérhető Android verziók láthatók. Itt ki lehet választani azt a verziót, amiből majd emulátort készítünk a futtatáshoz. Az alkalmazás futtatásához minimum 4.0 feletti Android verzió szükséges. A jelölő négyzetek segítségével kiválaszthatjuk a megfelelő verziót, aztán telepíthetjük. A csomagok letöltése után indítsuk el az Android Virtual Device Manager-t. A create gombra kattintva elkészíthető a saját emulátor.

Ezután importáljuk a project-et az Eclipse-be. A File menü import pontját választva a következő ablak jelenik meg. Kiválasztva az Androidon belül az Existing Android Code Into Workspace lehetőséget a továbbra kattintva ez az ablak jelenik meg.

Az importálás után bal oldalt megjelenik a teljes program struktúra. A projekt névén (Progv) jobb kattintással előugró menüben válasszuk a Run As lehetőségéből az Android Application-t. A program felhívja a figyelmünket, hogy nem érzékel fejlesztő eszközt csatlakoztatva ezért emulátort indít el. Az emulátor elindulása a gép teljesítményétől függően több percet is igénybe vehet. Elindulás után az alkalmazás megjelenik a képernyőn. Mielőtt bármit csinálunk, menjünk vissza az Eclipse-be. Az Emulátorra fel kell tenni a tömörített állományt, hogy az alkalmazás dolgozni tudjon vele.

Ennek menete:

A főképernyő jobb felső sarkában található ikonra kattintva hozzuk elő a DDMS-t itt tudunk betekintést nyerni az emulátorba működés közben. Az elindított emulátort kiválasztva, a jobb oldali ablakban ki lehet választani a fájl rendszert. Itt megkeressük az mnt jegyzéket azon belül az sdcard. Ha már egyszer elindítottuk az alkalmazást, akkor itt lesz egy bora jegyzék. Ebbe kell belemásolni a tömörített állományt. Ezt a jobb felső ikonnal tudjuk megtenni, amin egy piros nyíl mutat egy mobiltelefonra. Ezután újrafuttatva az alkalmazást már használatba is vehető. (lásd felhasználói utasítások)

Felhívom a figyelmet arra, hogy az emulátor elég lassan és akadozva is működhet. Mindenképpen ajánlom egy fejlesztőeszköz használatát a program gyors működéséhez.

Másik lehetőség, fejlesztő eszköz használata.

- Csatlakoztassuk az eszközt a számítógéphez. Az eszközillesztő programnak adjuk meg az sdk/extras/google/usb_driver mappában található illesztőt.
- Az Eclipse-ben futtatva az alkalmazást. A projekt névén (Progv) jobb kattintással előugró menüben válasszuk a Run As lehetőségéből az Android Application-t.
- A megjelenő ablakban válasszuk ki az eszközt. A program fordítása és telepítése után megjelenik az eszközön az alkalmazás.
- Ezután a fájlrendszerében az mnt/sdcard/bora jegyzékbe másoljuk be a tömörített forrásállományt. (Play Market-ben található Total Commander megkönnyíti a másolást.)
- Az alkalmazás használható.

A program használata során készített rendelések az eszközön kerülnek mentésre a későbbi felhasználásra. A készülék vagy emulátor fájlrendszerében megtalálhatók.

A cd-n található jegyzékek struktúrális felépítése:

- apk (mobil eszközre telepíthető állomány)
- Forrás_állomány (A teszt adatbázis)
- images (képek, ábrák)
- Latex (szakdolgozat latex változata)

- Program (A program forrása)
- test (A tesztek)
- UI (A felhasználói felületek végleges változata.)

Az alkalmazás használatát minimum 7 colos kijelzővel rendelkező és legalább 4.0 vagy feletti android rendszert futtató tableten javaslom.