



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Inventarizace majetku pomocí RFID ip a p enosného RFID tecího za ízení
Student:	Peter Mikolaj
Vedoucí:	Ing. Ján Mi kanin
Studijní program:	Informatika
Studijní obor:	Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce zimního semestru 2018/19

Pokyny pro vypracování

Cílem práce je systém pro evidenci a inventarizaci majetku za pomocí RFID ip a p enosného RFID za ízení dle specifických požadavk zadavatele.

Funk ní a nefunk ní požadavky na výsledný systém:

- synchronizace s existujícím ú etním programem - import a export evidence majetku,
- software pro p enosné RFID za ízení, který bude um t:
 1. spárovat existující evidenci majetku s RFID ípy na majetku,
 2. provád t inventarizaci majetku: v dané místnosti zkontrolovat p ítomnost všech evidovaných položek,
 3. vyhodnotit inventarizaci majetku: vypsát nalezené, nenalezené nebo p emístn né položky.
- PC aplikace, ve které bude náhled databáze majetku a nástroje pro její editaci.

Postupujte v krocích:

1. Formalizujte požadavky zadavatele.
2. Analyzujte procesy evidence majektu.
3. Navrhn te systém.
4. Diskutujte a zvolte vhodnou implementa ní platformu, implementujte první verzi systému.
5. ešení zdokumentujte a vhodným zp sobem otestujte.
6. Zhodno te výsledek.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 5. b ezna 2017

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTVÉROVÉHO INŽENÝRSTVÍ



Bakalárska práca

Inventarizace majetku pomocí RFID čipů a přenosného RFID čtecího zařízení

Peter Mikolaj

Vedúci práce: Ing. Ján Mičkanin

16. mája 2017

Pod'akovanie

Rád by som sa poďakoval Ing. Jánovi Mičkaninovi za ochotu pri riešení tejto bakalárskej práce, Ing. Michalovi Valentovi Ph.D. za vstriečný prístup nielen pri zadávaní témy bakalárskej práce a Petre Kudrnovej za veľmi rýchle vybavenie formalít na študijnom oddelení.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov. Ďalej prehlasujem, že som s ČVUT uzavrel dohodu, na základe ktorej sa ČVUT vzdalo práva na uzavrenie licenčnej zmluvy o používaní tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona. Táto skutočnosť nemá vplyv na ust. § 47b zákona č. 111/1998 Sb. o vysokých školách.

V Praze 16. mája 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 Peter Mikolaj. Všetky práva vyhradené.

Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.

Odkaz na túto prácu

Mikolaj, Peter. *Inventarizace majetku pomocí RFID čipů a přenosného RFID čtecího zařízení*. Bakalárska práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Bakalárska práca sa zaoberá analýzou postupov inventarizácie majetku a návrhom, implementáciou, testovaním, a nasadením informačného systému pre evidenciu a inventarizáciu majetku pomocou RFID čipov. Pre snímanie RFID čipov sa využíva *handheld* zariadenie. Prínosom práce je zjednodušenie repetitívnych úkonov zamestnancov pri inventarizácii majetku.

Kľúčová slova informační, systém, RFID, inventarizace, handheld, smart, Windows, CE

Abstract

Goal of the bachelor's thesis is to analyze property inventory processes and the design, implementation, testing and release of information system for property inventory using RFID tags. RFID tags are scanned by a handheld reader device. The advantage of using such a system is simplification of repetitive processes of property inventory.

Keywords information, system, RFID, inventory, handheld, smart, Windows, CE

Obsah

Úvod	1
1 Cieľ práce	3
2 Rešerš	5
2.1 Súčasný stav riešenia	5
2.2 Existujúce možnosti riešenia	5
2.3 Zariadenie Atid AT870	6
2.4 Windows CE 5.0	7
2.5 EPC UHF	7
3 Analýza a formalizácia požiadavkov	11
3.1 Synchronizácia s existujúcim účtovníckym programom	11
3.2 Software pre prenosné RFID zariadenie.	13
3.3 PC aplikácia	14
4 Návrh	15
4.1 Výber technológií	15
4.2 Schéma IS	17
4.3 Databázový model	18
4.4 Párovanie položiek s čipmi	20
4.5 Inventarizácia	21
5 Realizácia	23
5.1 Software pre presnosné RFID zariadenie.	23
5.2 API RFID modulu	24
5.3 Debugovanie a optimalizácia	26

5.4	Výsledná aplikácia	31
6	Testovanie aplikácie a školenie zamestnancov.	33
	Záver	35
	Literatúra	37
A	Zoznam použitých skratiek	39
B	Obsah priloženého CD	41

Zoznam obrázkov

2.1	zariadenie AT870	6
3.1	delenie majetku podľa dĺžky času použitia	11
4.1	Tok dát v informačnom systéme	17
4.2	Životný cyklus objektu Inventory	19
4.3	Flowchart priebehu párovania	20
4.4	Flowchart priebehu inventarizácie	21
5.1	usb konektor na kolíske	24
5.2	výnimka pri zaplnení virtuálneho pamäťového priestoru	27
5.3	virtuálna pamäť po importe prvých 300 položiek	28
5.4	Štruktúra virtuálneho pamäťového priestoru Windows CE	28
5.5	Štruktúra virtuálneho pamäťového 32 MB priestoru procesu Windows CE	29
5.6	virtuálna pamäť po štarte aplikácie (modrá - rezervovaný priestor, zelená - obsadený priestor, červená - voľný priestor)	29
5.7	virtuálna pamäť dvoch rôznych procesov - prvý využíva knižnicu A a B, druhý knižnicu C	30
5.8	Vyvolanie činnosti <i>garbage collectoru</i>	30
5.9	Okno inventarizácie s prehľadom položiek	31
5.10	Okno párovania s prehľadom položiek	31
5.11	Okno vyhodnotenia	32

Zoznam tabuliek

2.1	Rozdelenie RFID čipov podľa tried	7
2.2	Rozdelenie RFID čipov podľa frekvencie	9
3.1	Příklad tabulky	12
3.2	Příklad tabulky	13
4.1	Funkcie knižnice AT_UHF.h	18
5.1	Funkcie knižnice AT_UHF.h	25
5.2	Funkcie knižnice AT_UHF.h	25

Úvod

INMM, ako každá štátna inštitúcia, eviduje zoznam krátkodobého a dlhodobého majetku. Zoznam majetku INMM je uložený v účtovníckom programe Omega, ktorého licenciu si zakúpilo 19 563 firiem [10] a patrí medzi najpoužívanéjšie ekonomické softvéry na Slovensku. Každý rok prebieha inventarizácia tohoto majetku a je potrebné z databázy Omegy exportovať zoznam, vytlačiť ho, fyzicky prejsť každú miestnosť so zoznamom, hľadať a odškrtnúť jednotlivé položky. Každá položka, napr. stolička, softvér alebo zrkadlo má unikátne inventárne číslo a pri samotnej inventarizácii majetku je časovo náročné práve dohľadávanie položiek majetku, ktoré boli premiestnené v priebehu roka.

Ciel' práce

Cielom práce je analyzovať metodiku inventarizácie majetku, jeho evidencie, navrhnúť informačný systém, ktorý bude zjednodušovať úkony spojené s inventarizáciou využitím RFID čipov a prenosného RFID čítacieho zariadenia. Informačný systém by mal vedieť spolupracovať s existujúcou evidenciou majetku a splňať základné funkčnosti:

1. Párovanie existujúcej evidencie majetku s RFID čipmi na majetku.
2. Samotné prevedenie inventarizácie majetku, čo znamená v danej miestnosti skontrolovať prítomnosť všetkých evidovaných položiek označených RFID čipmi.
3. Vyhodnotenie inventarizácie majetku vypísaním nájdených, nenájdených a premiestnených položiek.

Rešerš

2.1 Súčasný stav riešenia

Aktuálne sa inventarizácia robí manuálne pomocou vytlačených zostáv majetku. Z účtovníckeho programu Omega, v ktorom je vedená evidencia majetku sa vytlačí predloha k inventarizácii, ktorá obsahuje evidenčné číslo majetku, názov a počet kusov. S touto papierovou podobou chodí inventarizačná komisia po celom objekte miestnosť po miestnosti a hľadajú sa položky, ktoré sa nachádzajú v zozname. Nájdené položky sa označia v zozname. V prípade nenájdenia položky vzniká manko. V prípade nájdenia nadbytočnej položky, napríklad takej, ktorá v minulosti bola vyradená ale fyzicky nebola zlikvidovaná, sa táto položka dopíše do zoznamu a vzniká prebytok. Najkomplikovanejšie a časovo najnáročnejšie je ale nájdenie premiestnených položiek, teda takých, ktoré sa v miestnosti nenachádzajú ale nachádzajú sa inde v objekte. Tieto položky je potrebné nájsť a premiestniť do miestnosti, v ktorej sú evidované alebo ich preevidovať do miestnosti, v ktorej momentálne sú.

2.2 Existujúce možnosti riešenia

Aktuálne je na slovenskom trhu niekoľko dodávateľov RFID zariadení a softvérových riešení. Tie sa dajú rozdeliť na:

Zákazky na mieru

Riešenia vyvinuté pre špecifické procesy napr. vo veľkých továrňach. Tieto zakázky vyžadujú osobitný prístup k projektu, ako zo strany dodávateľa, tak zákazníka. Informačný systém musí spĺňať konkrétne požiadavky a byť

2. REŠERŠ

čo najefektívnejší pre jednotlivé úkony, preto je integrovaný do existujúcich systémov.[2] Pre vytvorenie takéhoto informačného systému je potrebná analýza, návrh, vývoj, testovanie, prípadné zmeny a prijatie riešenia zákazníkom. V celom procese sú zapojené obe strany a prebieha v ňom neustála komunikácia, a to od zadania požiadavkov, zvolenia návrhu cez zmenové požiadavky pri vývoji a postupnom testovaní až po prijatie zákazníkom.

Univerzálne softvérové riešenie

Informačné systémy, ktoré sú vyvinuté pre obecné problematiky ako riadenie skladov, transport alebo objednávkový systém. Univerzálne informačné systémy vyžadujú väčšiu réžiu pri práci s existujúcimi systémami evidencie, nakoľko nie sú plne integrované. Vo všeobecnosti dokážu importovať a exportovať zostavy, medzi týmito dvoma krokmi je ale častokrát vyžadovaná konverzia dát. V reále to znamená importy a exporty zostáv jednotlivých súčastí, menej efektívnu prácu s počítačom a menšiu plynulosť a prehľadnosť prevádzaných úkonov. Ďalšou možnosťou integrácie je napr. REST API, teda webové aplikačné rozhranie [1], ktorého použitie je ale potrebné doimplementovať do existujúcej evidencie majetku, čo sú ďalšie náklady pre zákazníka a zbytočná nepriama réžia spojená s inventarizáciou. Príkladom takéhoto univerzálneho riešenia je Easyasset od firmy Barco [1].

Priamo v zadaní bakalárskej práce je zadané vypracovanie zákazky na mieru.

2.3 Zariadenie Atid AT870



Obr. 2.1: zariadenie AT870

Prístroj AT870 2.1, ktorý sa bude pri inventarizácii používať je *handheld* prenosné zariadenie, na ktorom beží operačný systém Windows CE 5.0. Jeho

súčasťou je UHF anténa na čítanie RFID značiek EPC druhej generácie. Obsahuje tiež kameru na snímanie jednodimenzionálnych čiarových alebo dvojdimenzionálnych QR kódov, GPS modul a GSM modul pre spojenie s mobilnou sieťou. Prenos dát na zariadenie je možný cez microSD slot. Pre prepojenie s počítačom slúži buď zabudovaný WLAN modul, alebo USB konektor vedúci priamo do zariadenia alebo do kolísky, do ktorej je možné zariadenie osadiť.

2.4 Windows CE 5.0

Windows CE 5.0 je v poradí tretie vydanie systému Windows CE .NET. Je to otvorený[7], kompaktný *real-time* operačný systém bežiaci na architektúrach x86, ARM a MIPS. Implementuje API, ktoré podmnožina Win32 API, preto sa na ňom nedajú bez úprav spustiť Win32 aplikácie skompilované pre plnohodnotné Windows operačné systémy. Obsahuje tiež Direct3Dmobile API pre prácu s 3D grafikou. [8]

2.5 EPC UHF

RFID, teda identifikácia pomocou rádiových vln sa využíva už od 2. svetovej vojny. Prvou aplikáciou v komerčnej sfére bolo označenie výrobkov pomocou EAS čipov, ktoré fungovali ako jednobitové systémy, ktoré indikovali prítomnosť čipu,[3] čo malo zabrániť krádežam výrobkov. V súčasnosti môžeme rozdeliť [4] RFID čipy na 5 tried (tab. 2.1).

Trieda	Názov	Pamäť	Zdroj napájania	Využitie
A	EAS	bez pamäte	Pasívny	Sledovanie tovaru
B	RO EPC	<i>Read only</i>	Pasívny	Identifikácia
C	EPC	<i>Read/Write</i>	Pasívny	Logovanie dát
D	Senzorové	<i>Read/Write</i>	Polopasívny	Senzory prostredia
E	Zrnkové	<i>Read/Write</i>	Aktívny	Ad Hoc siete

Tabuľka 2.1: Rozdelenie RFID čipov podľa tried

EAS

EAS čipy neobsahujú žiadnu informáciu, vysielajú jeden bit informácie o

2. REŠERŠ

svojej prítomnosti. Sú najlacnejšou variantou RFID čipov, ale keďže nie sú jednoznačne identifikované, nedá sa hovoriť o plnohodnotnej identifikácii.

RO EPC

Read only EPC obsahujú identifikačnú informáciu. Pretože sa na ne nedá zapisovať, je táto informácia, zväčša unikátne ID, zapísaná na čip už pri výrobe. Môžu sa využívať v jednoduchých čipových kartách, ktoré vysielajú jeden identifikačný reťazec, nie je to však bezpečné kvôli jednoduchému prečítaniu informácie iným čítacím zariadením. Pre jednoduchú funkčnosť je možné RO EPC navrhnúť bez mikročipu.

EPC

EPC, narozdiel od RO EPC, obsahujú zapisovateľnú alebo prepisovateľnú pamäť. Tá sa dá využiť na vytvorenie vlastného identifikátora alebo na uchovávanie informácií. EPC musia byť schopné čítať a zapisovať pamäť a preto obsahujú mikročip, kvôli ktorému sú nákladnejšie ako RO EPC čipy.

Senzorové

Senzorové RFID čipy sú schopné samostatne, teda bez prítomnosti RFID čítačky zaznamenávať dáta. Môže ísť o teplotu, zrýchlenie, zmenu orientácie, vibrácie, prítomnosť svetla alebo chemických látok. Aby mohli samostatne zbierať tieto dáta, musia mať zdroj energie mimo RFID čítačky, teda musia byť polopasívne alebo aktívne.

Zrnkové

Zrnkové, alebo anglicky *notes* RFID čipy sú schopné navzájom komunikovať alebo iniciovať spojenie s prijímačom. Ide o mikroprocesory s výpočtovým výkonom, preto sú rádovo drahšie ako predchádzajúce varianty RFID čipov.

Podľa frekvencie, na ktorej operujú sa RFID čipy v štandardizácii ISO 18000[5] rozdeľujú na LF, HF a UHF.[6] Každá frekvencia má špecifiká, ktoré obmedzujú jej použitie (tab. 2.2).

Označenie	Frekvencia	Dosah	Smerovosť signálu
LF	128 kHz	Centimetre	Všesmerový
HF	13.56 MHz	Desiatky cm	Všesmerový
UHF	860 960 MHz	5m	Všesmerový
2.4G	2.4 GHz	3m	Úzky

Tabuľka 2.2: Rozdelenie RFID čipov podľa frekvencie

Pre označenie majetku RFID čipmi som zvolil EPC druhej generácie, čo je čip pracujúci na ultra vysokej frekvencii, teda 860 - 960 MHz s dosahom do 5 metrov. Ide o variantu s najväčším dosahom spomedzi pasívnych RFID čipov s rozumnou jednotkovou cenou vzhľadom na počet inventárnych položiek, ktorých je cca. 3000, preto sa javí ako najvhodnejšia varianta na účely inventarizácie.

Analýza a formalizácia požiadavkov

Z ekonomického hľadiska nás bude zaujímať dlhodobý hmotný a nehmotný majetok a krátkodobý majetok v používaní (hmotný aj nehmotný), ktorý je evidovaný v operatívnej evidencii.[9]



Obr. 3.1: delenie majetku podľa dĺžky času použitia

3.1 Synchronizácia s existujúcim účtovníckym programom

Účtovnícky program Omega, ktorý obsahuje databázu majetku vedie evidenciu v databázovom prostredí Microsoft Office Access. Databázový súbor neobsahuje väzby serializovaných entít a aj keď sú na prvý pohľad zrejme stĺpce cudzích kľúčov a odkazované tabuľky, nie je bezpečné z hľadiska možných integritných obmedzení, ktoré sú implementované priamo v programe, ktorý obsluhuje databázu, používať tento .mdb súbor. Rovnaké stanovisko vyjadrila podpora programu Omega, kde v prípade priameho zásahu do databázy negarantujú funkčnosť programu.

3. ANALÝZA A FORMALIZÁCIA POŽIADAVKOV

Pre účely inventarizácie potrebujeme obojsmerný export a import. Z Omegy potrebujeme exportovať inventárne číslo, názov, miestnosť, dátum vyradenia a poznámku položky. Dátová definícia týchto stĺpcov je v tab. 3.1. Do poznámky inventárnej položky sa bude ukladať kód RFID čipu. Využitie tohoto stĺpca mi v e-mailovej korešpondencii odporúčala technická podpora Omegy. Od existujúcej poznámky bude kód oddelený oddeľovacími znakmi.

Tabuľka 3.1: dátové definície stĺpcov inventárnej položky v programe Omega

Názov stĺpca	Typ	Dĺžka
Inventárne číslo	short text	12
Názov	short text	50
Miestnosť	short text	30
Deň vyradenia	integer	4
Mesiac vyradenia	integer	4
Rok vyradenia	integer	4
Poznámka	long text	300

Pretože Omega pri importe položiek do tabuľky zmaže všetky stĺpce importovaného záznamu, pre úspešný opätovný import do programu Omega je potrebné importovať **všetky** stĺpce tabuľky inventárnych položiek. Počas celého procesu inventarizácie je teda nutné previesť min. 4 migrácie:

1. Exportovanie všetkých stĺpcov tabuľky inventárna položka vo formáte CSV do PC.
2. Importovanie stĺpcov zo súboru v PC, ktoré budeme používať alebo meniť do prenosného RFID zariadenia.
3. Exportovanie stĺpcov, ktoré mohli byť zmenené späť z prenosného RFID zariadenia do PC.
4. Importovanie aktualizovaného CSV súboru so všetkými inventárnymi položkami a stĺpcami do programu Omega.

Tabuľka vybraných stĺpcov exportovaného CSV súboru z Omegy (1537 položiek) 3.2.

Tabulka 3.2: exportovaný CSV súbor

Inv. č.	Názov	Umiestnenie	Dát. vyr.
1148	Nabíjačka batérií	1272 Vedúci EPČ	null
1149	Total Commander	1276 Sklad inf.	null
1183	Box plastový na zdrav. materiál	2006 sklad	null
1184	Tester na alkohol AL 7000	1273 Kancelária	null
1185	PC zostava VISION + Monitor 1037	1146 Ovládač 1	null
1186	Software k vyhodn. stanici VISION	1146 Ovládač 1	null

3.2 Software pre prenosné RFID zariadenie.

Zariadenie AT870 beží na platforme Windows Compact Edition 5.0 Professional, čo je kompaktná verzia systému Windows pre *embedded* (vstavané) systémy. Na implementáciu softvéru sa využije jeden z dostupných jazykov (Visual C++, Visual C#, Visual Basic, JScript, ASP.NET)

3.2.1 Spárovanie existujúcej evidencie majetku s RFID čipmi na majetku

Pre spárovanie je potrebné:

1. Importovať do zariadenia položku z CSV súboru, ktorý sa vyexportuje z Omegy.
2. Vybrať si na zariadení miestnosť, v ktorej položky ešte nie sú priradené k ich čipovým kódom.
3. Označiť majetok v miestnosti (vrátane tabličky miestnosti) RFID čipom, na ktorom je napísané inventárne číslo pre prípadnú manuálnu kontrolu.
4. Spárovať záznam položky v prenosnom zariadení s RFID čipom priblížením antény nastavenej na primeranú citlivosť a stlačením "pištole". Spárované položky sú od nespárovaných odlíšené farbou pozadia.
5. Importovať do Omegy kód RFID čipu spolu so záznamom inventárnej položky.

3.2.2 Prevedenie inventarizácie majetku

Samotná inventarizácia pomocou prenosného RFID zariadenia prebieha nasledovným spôsobom: ako prvé načítame čip miestnosti nachádzajúci sa za tabuľkou označenia danej miestnosti. Na zariadení sa zobrazí zoznam položiek v načítanej miestnosti (v prípade opakovanej inventarizácie sa v minulosti nájdené položky v tejto miestnosti odlišia v zobrazení farbou pozadia). Pri primeranej intenzite stlačíme „kohútik pištole“, čím naskenujeme všetky kódom označené položky v miestnosti. Prípadné nenájdené položky (pri nečakanej chybe prístroja, kódu...) sa dajú dodatočne nájsť manuálnym prečítaním kódu a samostatným odpísaním zo zoznamu v prístroji, ktorý je k dispozícii za akýchkoľvek podmienok. Na záver je nutné uložiť stav priebehu inventarizácie a môžeme pokračovať do ďalšej miestnosti.

3.2.3 Vyhodnotenie inventarizácie majetku.

Vyhodnotenie inventarizácie musí byť možné priamo na prenosnom RFID čítačom zariadení, a to ako kvôli kontrole kompletnosti prebiehajúcej inventarizácie tak aj kvôli možnosti (ručného) dohľadania nenájdených položiek, prípadne ich premiestnenia/previdovania do inej miestnosti. Prenosné zariadenie teda bude vedieť:

1. Zobrazit' nájdené a nenájdené položky
2. Presunúť nájdené/nenájdené položky do inej miestnosti
3. Inventarizácia miestnosti nie je finálna, takže spätné dohľadanie položiek v miestnosti je kedykoľvek možné a teda ručné dohľadanie položiek vyplýva z predchádzajúcej podsekcie 3.2.2

3.3 PC aplikácia

PC aplikácia by mala obsahovať prehľadný zoznam databázy majetku, nástroje pre jej editáciu a vyhodnotenie inventarizácie.

Za jednoduchú variantu tejto aplikácie sa dá považovať grafický SQL klient so sadou agregáčnych dotazov.

Návrh

4.1 Výber technológií

4.1.1 Prenosné RFID čítacie zariadenie

Prenosné RFID zariadenie AT 870 beží na platforme Windows CE 5.0. Pre vývoj softvéru pre handheld zariadenie sa dá použiť:

- Visual C++ - natívny low-level vývojový jazyk, kompiluje sa priamo do strojového kódu (assembler), neobsahuje réžiu pri vykonávaní kódu, takže sa program vykonáva rýchlejšie. Programátor má väčšiu zodpovednosť pri správe pamäte (alokácia a uvoľňovanie objektov), čo môže viesť ku kritickému zastaveniu programu, takže výsledkom je väčšia náchylnosť na pád pri chybe v kóde. Visual C++ ale nedisponuje takým množstvom knižníc pre prácu s grafickým prostredím ako Visual C# a preto je vývoj pomalší.
- Visual C# - menežovaný jazyk, podporuje knižnicu .NET Compact Framework (zjednodušený ekvivalent stdlib), čo je súbor tried pre často sa opakujúce úkony, ako napr. triedy pre manipuláciu so zoznamom objektov ale aj grafické prvky ako okná výnimiek alebo užívateľské dialógy. Súčasťou vývojových nástrojov pre C# je WYSIWYG dizajnér grafického rozhrania s intuitívnym ovládaním. Visual C#, pretože je menežovaný a programátor sa nestará o životný cyklus objektov, je odolnejší voči chybám v kóde a vhodnejší na malé, nenáročné aplikácie.
- Visual Basic .NET - menežovaný jazyk, "vývojovo príbuzný" s Visual C# - jazyky mali odlišné základy ale od roku 2000 ich firma Microsoft

vyvíjala s podporou .NET, takže ich funkcionalita sa veľmi nelíši. Visual Basic .NET je ale zastaralejší a obecné je C# populárnejší.

- JScript na klientskej strane - webový prehliadač, ktorý obsahujú zariadenia Windows Mobile podporuje JScript, takže je možné v okne prehliadača pomocou technológie AJAX dosiahnuť istý stupeň interakcie a zároveň komunikáciu so vzdialeným serverom. JScript aplikácie ale nemajú prístup k lokálnym dátam (inak, než cez cookies) a samotné prenosné RFID zariadenie by malo fungovať aj mimo dosah wifi sietí (sklady, odľahlé časti), takže sa toto riešenie nejaví ako vhodné.
- ASP.NET na serverovej strane - nevhodné z rovnakého dôvodu ako predošle riešenie - obmedzený dosah wifi sietí.

Pre vývoj aplikácie prenosného RFID zariadenia som zvolil jazyk Visual C# pre jeho jednoduchosť, robustnosť a menšiu náchylnosť na chyby v kóde.

4.1.2 PC aplikácia

Pre vývoje aplikácie na počítači som taktiež zvolil jazyk Visual C#, pretože využíva rovnaké nástroje ako aplikácia na zariadení a je kompatibilný s použitými technológiami.

4.1.3 Serializácia dát - databázové prostredie

Databáze podporované Windows CE:

- CEDB - veľmi odľahčená varianta perzistencie, každá databáza obsahuje len jednu tabuľku bez predom danej štruktúry. Záznamy majú rôzny počet stĺpcov a jedna databáza môže obsahovať len 4 zoradené stĺpce. Nie je vhodná na väčší počet záznamov (>1000), takže sa nehodí pre naše účely [11]
- EDB - vynovená a vylepšená verzia CEDB, navyše podporuje transakcie, prístup viacerých užívateľov, viacero zoradení stĺpcov, viacero kľúčových stĺpcov, viaceré databázy a zlepšený výkon pri väčších databázach Enhanced performance, especially with larger databases [12]
- SQLite - jednoduchá knižnica implementujúca databázové prostredie napísaná v jazyku C. Podporuje ACID transakcie, implementuje väčšinu zo štandardu SQL92, databáza je ukládaná v jednom súbore,

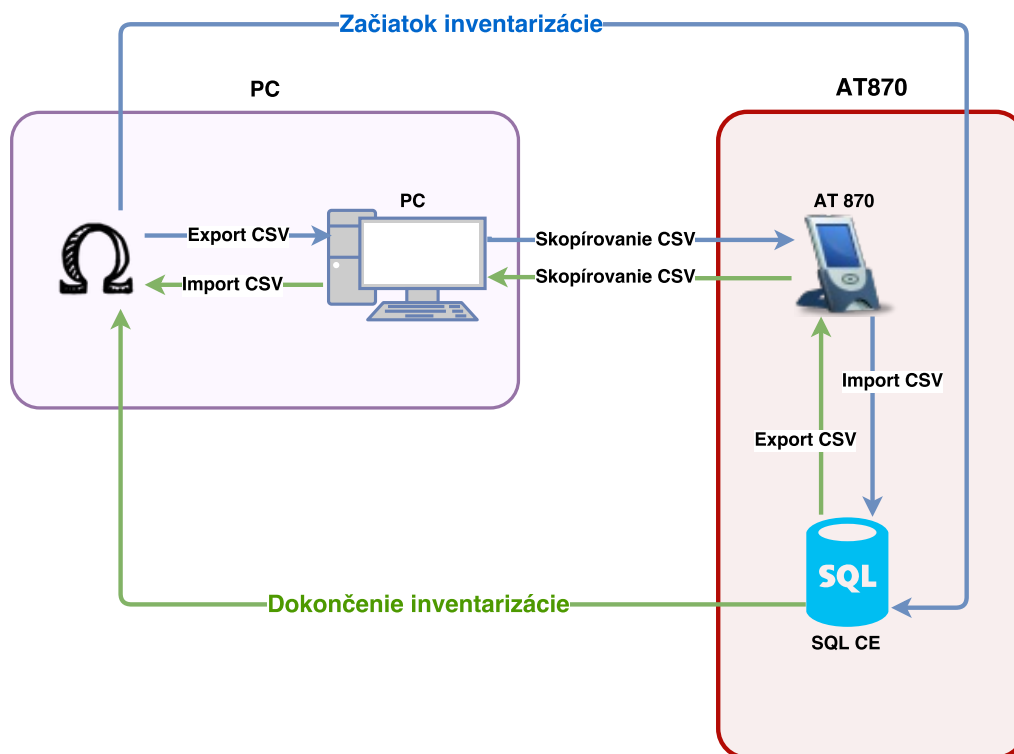
jednoduché api, bez ďalších závislostí. Podporuje stránkovanie, operácie v pamäti, poddotazy aj *views* (pohľady).[13]

- SQL CE - štandardné databázové prostredie od Microsoftu, nepodporuje stránkovanie, poddotazy ani *views* (pohľady).

Aj keď SQL CE zaostáva za SQLite ako výkonom tak funkcionalitou, vystačil som si s jej obmedzenými možnosťami a zvolil som si ju pre *out-of-the-box* podporu vývojovým prostredím Visual Studio.

4.2 Schéma IS

Schéma informačného systému a toku dát je zobrazená na obr.4.1.



Obr. 4.1: Tok dát v informačnom systéme

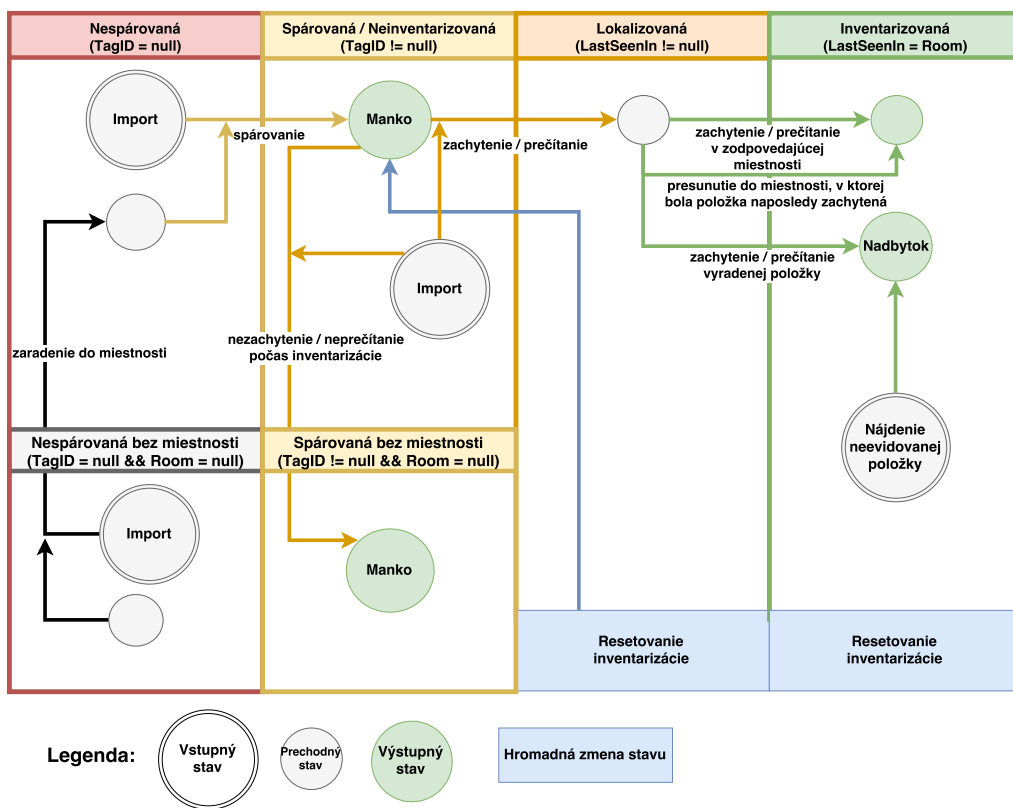
4.3 Databázový model

Databázový model je vzhľadom na možnosti importu a exportu z programu Omega veľmi jednoduchý. Databáza InventoryDB obsahuje jedinú tabuľku **Inventory**, ktorá má okrem atribútov, ktoré sú obsiahnuté pri exporte inventárnych položiek navyše ešte stĺpce TagID, Decommissioned a LastSeenIn, ktoré slúžia pre funkčnosť programu. Stĺpce sú popísané v tab.4.1

Názov stĺpca	Datový typ	Dĺžka	Môže obsahovať null	Unikátny	Primárny kľúč
InventoryID	nvarchar	16	Nie	Áno	Áno
Name	nvarchar	64	Áno	Nie	Nie
Room	nvarchar	8	Áno	Nie	Nie
TagID	nvarchar	24	Áno	Nie	Nie
Decommissioned	bit	1	Áno	Nie	Nie
LastSeenIn	nvarchar	8	Áno	Nie	Nie

Tabuľka 4.1: Funkcie knižnice AT_UHF.h pre prácu s RFID modulom

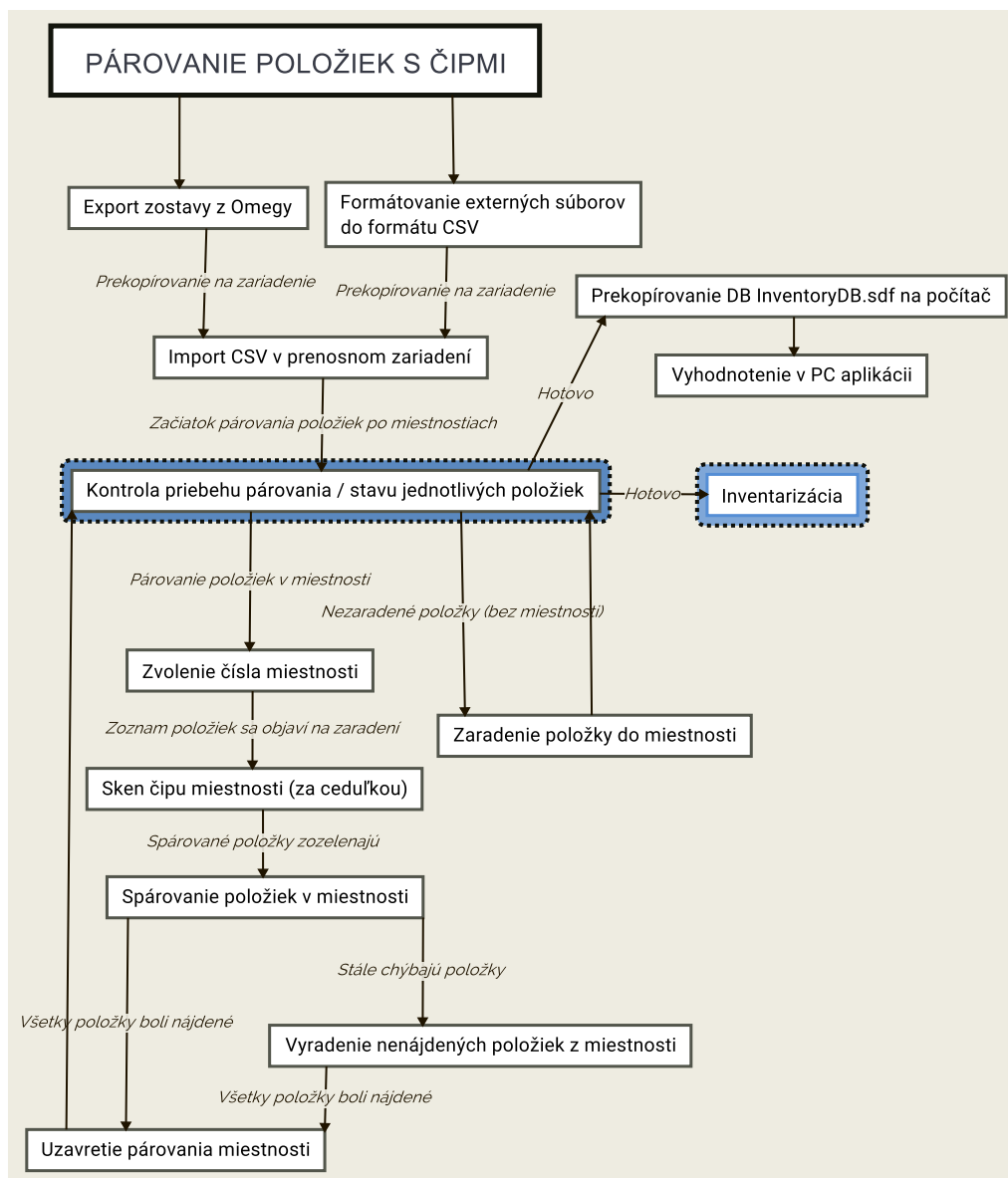
Životný cyklus objektu Inventory, teda inventárneho záznamu je popísaný v diagrame obr.4.2.



Obr. 4.2: Životný cyklus objektu Inventory

4.4 Párovanie položiek s čipmi

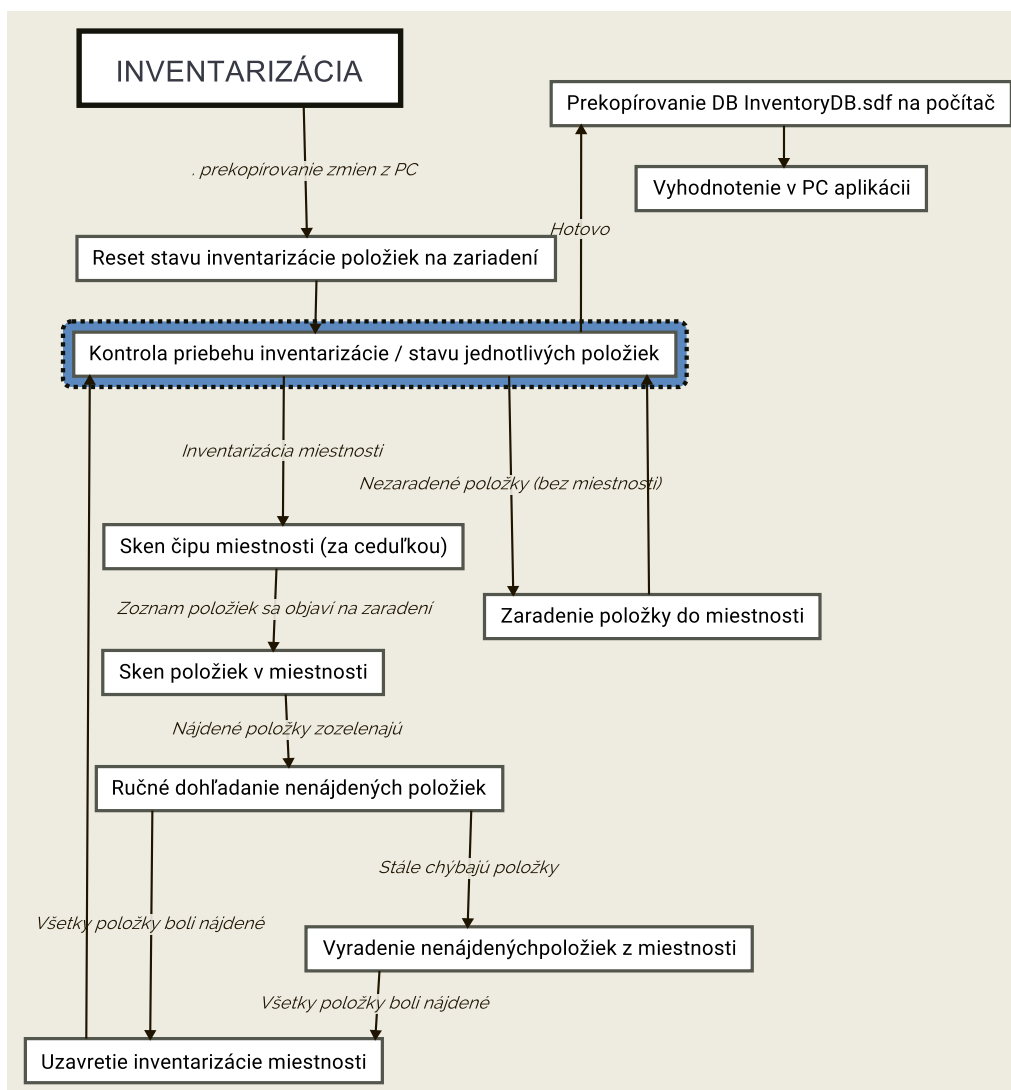
Párovanie položiek s čipmi sa začína importom zostavy vo formáte CSV do zariadenia. Pracovný cyklus, v ktorom prebieha párovanie sa nachádza medzi kontrolou priebehu párovania a začína buď párovaním miestnosti alebo zaradením individuálnej nezaradenej položky.



Obr. 4.3: Flowchart priebehu párovania

4.5 Inventarizácia

Podobne ako pri párovaní položiek, aj pri inventarizácii je pracovný cyklus v ktorom prebieha inventarizácia začína stále inventarizáciou miestnosti alebo párovaním položky.



Obr. 4.4: Flowchart priebehu inventarizácie

Realizácia

5.1 Software pre presnosné RFID zariadenie.

5.1.1 Výber vývojového prostredia.

Vývoj zariadení so systémom Windows CE 5.0 podporuje ako posledné vývojové štúdio Visual Studio 2008.

Funkčnosti VS 2008 a smart zariadení:

- Okamžité nasadenie aplikácie vrátane okamžitej inštalácie všetkých potrebných modulov a knižníc, ako sú Microsoft SQL Server CE alebo Microsoft .NET Compact Framework 3.5.
- Debugovanie aplikácie za behu.
- Funkčnosti vzdialených nástrojov Visual Studia 2008 - prehliadač súborov, prehliadač haldy, prehliadač procesov, editor registrov a špión (schématický prehľad GUI prvkov).
- Emulácia Windows smart zariadení s architektúrou armv4.

5.1.2 Prepojenie s vývojovým prostredím

Na využitie vyššie uvedených funkcií je potrebné zariadenie prepojiť buď pomocou usb *cradle* (kolísky), Bluetooth alebo tcp/ip pripojenia. Dodávateľ zariadenia AT870 nám dodal kolísku s nezodpovedajúcim USB káblom, nahliadni obr. 5.1.



Obr. 5.1: usb konektor na kólske

Konektor je atypický, objednať od dodávateľa sa nedá a výrobca by ho zaradil až do ďalšej výrobnjej várky. Pre prepojenie so *smart* zariadením pomocou Bluetooth sa do verzie operačného systému Windows Vista používala aplikácia ActiveSync, od OS Windows Vista ďalej sa používa Windows Mobile Device Center. WMDC sa mi z nezistených príčin a bez akýchkoľvek chybových hlások nepodarilo nainštalovať a preto som bol nútený použiť protokol tcp/ip. Zariadenie ATID870 neobsahuje *ethernet* port, a tak som vytvoril v systéme Windows 7 prístupový bod WiFi,

```
netsh wlan set hostednetwork mode=allow ssid=<ssid> key=<password>
netsh wlan start hostednetwork
```

na ktorý som sa pomocou zariadenia pripojil. Pre nadviazanie pripojenia bez ActiveSync je nutné skopírovať súbory z priečinka
<c:/ProgramFiles(x86)/CommonFiles/microsoftshared/CoreCon/1.0/Target/wce400/armv4/> do priečinka
<Windows/> na zariadenie manuálne, spustiť najprv program
<ConmanClient2.exe>, následne <CMAccept.exe> a do 3 minút iniciovať spojenie z programu Visual Studio.

5.2 API RFID modulu

Pre vývoj softvéru dodal výrobca Atid k modelu AT870 knižnice API a dva hlavičkové súbory AT_UHF.h a ATUTIL.h.

ATUTIL.h obsahuje definície funkcií pre ovládanie zariadenia, ako napr. ovládanie hlasitosti, napájania, fotoaparátu, podsvietenia, vibrátoru alebo modulu WLAN.

AT_UHF.h obsahuje definície funkcií pre prácu s RFID modulom:

Tabuľka 5.1: Funkcie knižnice AT_UHF.h pre prácu s RFID modulom

Funkcia	Súvislé sk.	Chyba/Úspech	Dĺžka sk.
EPC_GEN2_ONE_TAG_SELECT			
AT_UHF_ReadUID_Tag_Select()	Áno	Chybová správa	Neobmedzená
AT_UHF_ReadUID()	Nie	Chybová správa	SCANTIME
EPC_GEN2_MULTI_TAG_SELECT			
AT_UHF_ReadUID_TagSelect()	X	Chybová správa	SCANTIME
AT_UHF_ReadUID()	X	Chybová správa	SCANTIME
AT_UHF_ReadEPC_Continuous()	X	Chybová správa	Neobmedzená
ACCESS(R/W/LOCK/KILL)	X	Ch. spr./"Success"	SCANTIME

Tabuľka 5.2: Funkcie knižnice AT_UHF.h pre prácu s RFID modulom

Funkcia	Timeout	Návratová hodnota
EPC_GEN2_ONE_TAG_SELECT		
AT_UHF_ReadUID_Tag_Select()	X	"Not Detect"/"ÖK"
AT_UHF_ReadUID()	"Not Detect"/"ÖK"	"Not Detect"/"ÖK"
EPC_GEN2_MULTI_TAG_SELECT		
AT_UHF_ReadUID_TagSelect()	"Multi Read Stop"	"Multi Read Stop"
AT_UHF_ReadUID()	"Multi Read Stop"	"Multi Read Stop"
AT_UHF_ReadEPC_Continuous()	X	"Multi Read Stop"
ACCESS(R/W/LOCK/KILL)	"Not Detect"	"Not Detect"

Na prácu s RFID čipmi budem potrebovať dve funkcie: načítanie jedného tagu a nepretržité čítanie kódov v dosahu. Zjednodušená ukážka kódu 5.2 znázorňuje použitie týchto základných funkcií.

```
// Inicializacia RFID modulu
public static AT_UHF_NET.CUHFHost rfid;
Util_PowerOn_NET.CUtil_PowerOn PowerNotify;
PowerNotify = new Util_PowerOn_NET.CUtil_PowerOn(this);
PowerNotify.AT_PowerNotifyHWND();
PowerNotify.AT_PowerNotifyEvent();
rfid = new AT_UHF_NET.CUHFHost();
rfid.PowerOnInit(); // *** Must call this fuction. ***

// Nastavenie intenzity snimania
```

5. REALIZÁCIA

```
    rfid.SET_PowerControl(trackBarPower.Value);

// Spustenie snimania viacerych kodov
rfid.ActivatedForm = this;
rfid.ReadUID(AT_UHF_NET.UIDREAD_CODE.EPC_GEN2_MULTI_TAG
    );

// Spracovanie precitaneho kodu
public void GetAccesseEPC(string EPC) {
    if (EPC == null || EPC.Length < 28) {
        System.Windows.Forms.MessageBox.Show("Nacitany
            neplatny tag: " + EPC);
        AT_UHF_NET.CUHFHost.PlayFail();
        return;
    }
    EPC = EPC.Substring(4, 24);
    ...

// Ukoncenie snimania viacerych kodov
rfid.Stop();

// Nacitanie jednotlivého kodu
rfid.ReadUID(AT_UHF_NET.UIDREAD_CODE.EPC_GEN2_ONE_TAG);

// Vypnutie RFID modulu
PowerNotify.AT_PowerNotifyClose();
rfid.Close();
rfid.PowerOFF();
```

5.3 Debugovanie a optimalizácia

Okrem bugov v implementácii .NET, pre ktoré som párkrát musel preimplementovať časť GUI sa objavili pri behu prvej funkčnej verzie programu nedeterministické a zle reprodukovateľné chyby.

5.3.1 Pád programu pri importe.

Aplikácia na prenosnom zariadení pri importe väčšieho počtu položiek (>150) padá z nešpecifikovaných príčin - raz je to výnimka pri volaní knižnice .NET, inokedy nastane výnimka pri spracovaní SQL príkazu update:

```
at System.Data.SqlServerCe.SqlCeCommand.  
    ProcessResults ()  
at System.Data.SqlServerCe.SqlCeCommand.  
    CompileQueryPlan ()  
at System.Data.SqlServerCe.SqlCeCommand.  
    ExecuteCommand ()  
at System.Data.SqlServerCe.SqlCeCommand.  
    ExecuteNonQuery ()  
at INMM_RFID_Inventarizacia.CDatabase.updateQuery ()  
at INMM_RFID_Inventarizacia.CDatabase.updateQuery ()  
at INMM_RFID_Inventarizacia.CDatabase.ImportQuery ()  
at INMM_RFID_Inventarizacia.Porter.  
    buttonImport_Click ()  
at System.Windows.Forms.Control.OnClick ()  
at System.Windows.Forms.Button.OnClick ()  
at System.Windows.Forms.ButtonBase.WnProc ()  
at System.Windows.Forms.Control._InternalWnProc ()  
at Microsoft.AGL.Forms.EVL.EnterMainLoop ()  
at System.Windows.Forms.Application.Run ()  
at INMM_RFID_Inventarizacia.Program.Main ()
```

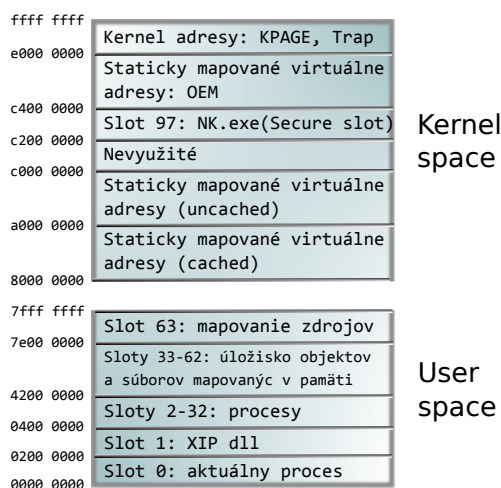
Obr. 5.2: výnimka pri zaplnení virtuálneho pamäťového priestoru

Na prvý pohľad nezmyselné chyby skrývajú závažný problém - zaplnenie virtuálneho pamäťového priestoru aplikácie, kvôli ktorému aplikácia padne pri volaní funkcie.

Každá aplikácia bežiaca na systéme Windows CE dostane pridelených 32 MB virtuálnej pamäte a má k dispozícii zdieľanú haldu maximálnej veľkosti 1 GB (pre zariadenie AT870 je to cca. 82 MB z celkových 128 MB RAM).



Obr. 5.3: virtuálna pamäť po importe prvých 300 položiek



Obr. 5.4: Štruktúra virtuálneho pamätového priestoru Windows CE

Pri importe položiek sa pri alokácii objektov na halde zdola obsadzuje voľná pamäť, až kým nenarazí zhora na priestor, ktorý obsadili knižnice - tzv. *DLL crunch* (obr. 5.3) a aplikácia začne hlásiť nesúvisiace chyby (obr. 5.2).

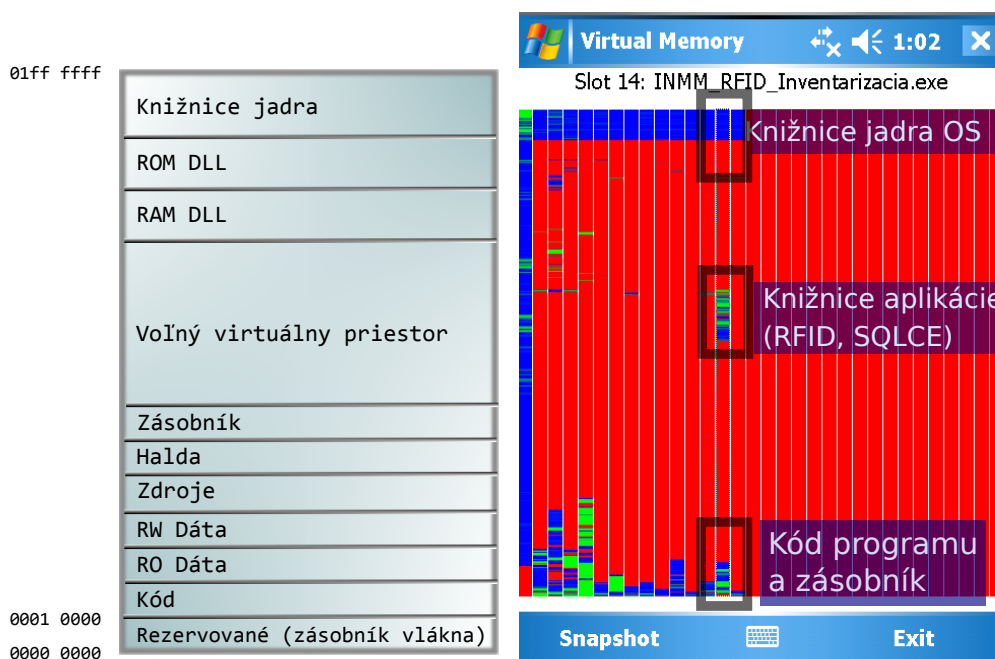
Virtuálna pamäť Windows CE 5.0

Celý virtuálny pamätový priestor systému Windows CE s 32 bitovou adresáciou má kapacitu 4 GB. Tento priestor sa delí na *kernel space*, ktorý je rezervovaný pre jadro operačného systému) a *user space*, kde sa alokujú nielen prostriedky užívateľských procesov. (obr. 5.4)

Virtuálna pamäť procesu vo Windows CE 5.0

User space sa ďalej delí na 64 slotov veľkosti 32 MB. Polovica týchto slotov obsahuje priestor pre jednotlivé bežiacie procesy. Celkovo teda na zariadení môže bežať paralelne 31 aplikácií a každá z nich má pridelený priestor 32 MB. (obr. 5.5)

Pri spustení nového procesu sa na zásobník rezervuje priestor veľkosti spustiteľného .exe súboru. Naviac sa zhora do pamäte procesu automaticky načítajú systémové DLL knižnice jadra operačného systému (obr. 5.6).



Obr. 5.5: Štruktúra virtuálneho pamäťového 32 MB priestoru procesu Windows CE

Obr. 5.6: virtuálna pamäť po štarte aplikácie (modrá - rezervovaný priestor, zelená - obsadený priestor, červená - voľný priestor)

Načítavanie knižníc dll do VM.

Knižnice jadra	Knižnice jadra
ROM DLL	ROM DLL
DLL A	DLL A
DLL B	DLL B
DLL C	DLL C
Halda č.2	
Vlákno č.2	
Halda	Halda
Hlavné vlákno	Hlavné vlákno
Proces č.1	Proces č.2
Rezervované)	Rezervované)

Obr. 5.7: virtuálna pamäť dvoch rôznych procesov - prvý využíva knižnice A a B, druhý knižnicu C

Priebeh alokácie pamäte:

- Spustí sa proces P1, ktorý využíva knižnice A,B s hornou adresou voľného virtuálneho pamäťového priestoru adr_{max}
- Knižnice A,B sa postupne alokujú na adrese $adr_{max} - size_a$ a $adr_{max} - size_a - size_b$
- Spustí sa proces P2, ktorý využíva knižnicu C s hornou adresou voľného virtuálneho pamäťového priestoru adr_{max}
- Knižnica C sa alokuje až od miesta $adr_{max} - size_a - size_b - size_c$, pretože je pre knižnice A,B vyhradený priestor na rovnakej pozícii ako v procese P1

(obr. 5.7).

Kód som najprv optimalizoval alokáciou objektov pred *for* cyklami, v ktorých som následne len menil ich atribúty. Toto nemalo žiaden dopad, takže pamäť zatažovali najskôr metódy z knižníc SQL CE. Následne som sa pokúsil periodickým uspávaním pracovného vlákna, v ktorom bežal *for* cyklus na krátky čas dať priestor *garbage collectoru*. Táto varianta pomohla, takže stačilo vyvolať činnosť GC (obr. 5.8).

```
GC.Collect();
GC.WaitForPendingFinalizers();
```

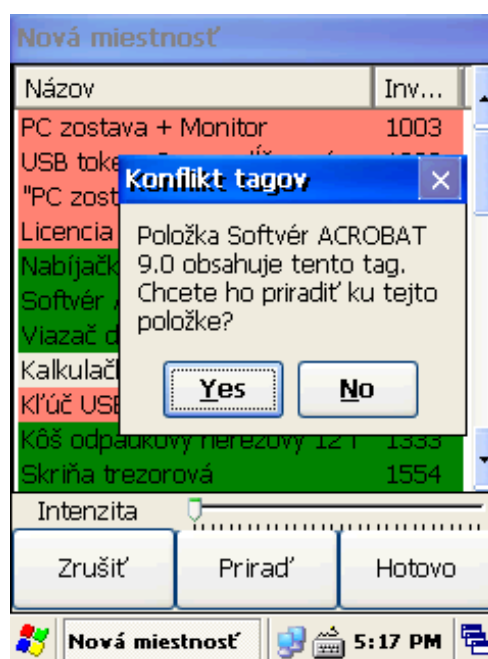
Obr. 5.8: Vyvolanie činnosti *garbage collectoru*.

5.4 Výsledná aplikácia

Po zmenových požiadavkách behom vývoja je grafické rozhranie aplikácie, vzhľadom na malé rozlíšenie obrazovky, navrhnuté čo najjednoduchšie. Na obrazovke figuruje zoznam položiek, jeden rad tlačidiel a posuvník pre nastavenie intenzity RFID modulu. Dodatočné funkcie, ako premiestnenie položiek alebo vyradenie položky z miestnosti, príp. jej presunutie do inej miestnosti tvoria funkčné tlačidlá umiestnené po bokoch prístroja.



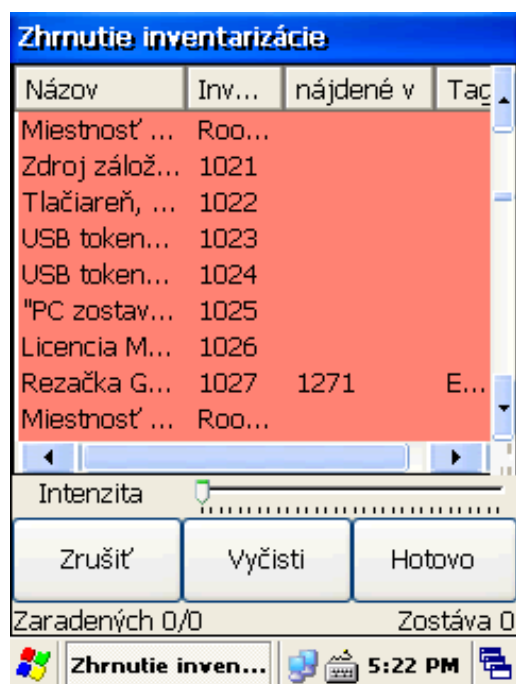
Obr. 5.9: Okno inventarizácie s prehľadom položiek



Obr. 5.10: Okno párovania s prehľadom položiek

V okne vyhodnotenia sa zobrazujú nenájdene položky a miestnosť, v ktorej boli naposledy zachytené, takže vieme, kde ich treba hľadať.

5. REALIZÁCIA



Obr. 5.11: Okno vyhodnotenia

Testovanie aplikácie a školenie zamestnancov.

V priebehu celého vývoja prebiehalo priebežné testovanie funkčných častí a zadávanie zmenových požiadavok povereným zamestnancom INMM. Týmto procesom zároveň získal zákazník hlbšie pochopenie problému implementácie IS a predstavu o možnostiach systému. Celý IS môžeme rozdeliť na 3 základné funkčné časti, ktoré boli testované v akceptačnom teste zadávateľom. Akceptačný test prebiehal tak, že poverený zamestnanec previedol za mojej asistencie a dohľadom tretej osoby - námestníka ekonomickej časti celý proces párovania čipov a inventarizácie od exportu z programu Omega až po opätovný import. Následne sa skontrolovala integrita dát v Omega a správny výsledok inventarizácie oproti fyzickému stavu inventarizovaných položiek v miestnostiach. Veľkosť testovacej vzorky boli 3 miestnosti, cca. 45 inventárnych položiek.

Párovanie položiek s čipmi

Pri párovaní položiek s RFID čipmi sa vyskytli problémy pri čipoch, ktoré boli umiestnené na kovovom povrchu. Kov značne utlmoval elektromagnetický signál, a tak bolo potrebné zvýšiť intenzitu RFID modulu na posuvníku v aplikácii alebo presunúť RFID čip na iné miesto na danom predmete. Pri osadovaní RFID čipov teda zadávateľ musí prihladnúť tiež na ich umiestnenie.

Inventarizácia

Pri inventarizácii sa pri plnej intenzite skenovania v miestnosti objavili opačné problémy, ako pri párovaní. Signál nie je úplne rušený stenami, preto sa pri inventarizácii objavilo upozornenie na zachytenú položku z vedľajšej

miestnosti. Po zmenšení intenzity RFID modulu priamo v aplikácii sa tento problém odstránil na úrovni, kedy zariadenie snímalo RFID čipy zo vzdialenosti cca. 4 metrov.

Vyhodnotenie v PC aplikácii

Pre striktnú kontrolu priebehu inventarizácie už v prenosnom zariadení, kedy nie je inventarizácia finálna, kým sa položka nenachádza v koncovom stave inventúry, teda nájdená, nenájdená, presunutá alebo vyradená, nie je potrebné kontrolovať stav inventarizácie v PC aplikácii. Na vyhodnotenie inventarizácie slúži Microsoft Excel, kde sa pri opätovnom importe jednoduchou funkciou dajú zobrazit položky, ktoré sa nenašli, teda nemajú priradenú miestnosť, alebo ktoré sa našli, ale boli vyradené, takže je u nich evidovaný dátum vyradenia. S týmito položkami sa ďalej pracuje v Omegae, prípadne vo výsledku inventarizácie.

Zhodnotenie

Napriek náročnejšiemu párovaniu a jeho občasým komplikáciám je inventarizácia miestnosti značne urýchlená. Čím viac je v miestnosti zhromaždených predmetov, a čím su hustejšie umiestnené, tým viac je efektívnejšie využívať RFID prenosné zariadenie. Ako vhodná aplikácia systému sa preto okrem inventarizácie javí sledovanie zásob na sklade.

Záver

Cieľom práce bolo zjednodušenie repetitívnych úkonov zamestnancov pri inventarizácii majetku vyvinutím informačného systému a softvéru pre RFID čítacie zariadenie. Tento cieľ sa mi podarilo naplniť vyvinutím softvéru pre prenosné zariadenie, ktorý je schopný načítať RFID značky, párovať ich s inventárnymi položkami a následne aj rýchle previesť kontrolu inventárnych položiek v miestnosti.

Praktický dopad je sprehľadnenie umiestnenia majetku ale hlavne pohodlnejšia a rýchlejšia inventarizácia majetku.

Ako ďalšie zlepšenia súčasného riešenia by som navrhoval priamu integráciu do účtovníckeho programu Omega, prípadne rozšírenie RFID technológie do ďalších procesov (evidencia liekov, pacientov, dohádzka zdravotníckeho personálu).

Literatúra

- [1] Barco, *evidencia majetku Easyasset*, <http://barco.sk/easyasset-evidencia-a-inventarizacia-majetku>, cit.10.5.2017
- [2] Barco, *Vývoj riešenia na mieru*, <http://barco.sk/vyvoj-software>, cit.10.5.2017
- [3] Mario W. Cardullo, *Genesis of the Versatile RFID Tag*, <http://www.rfidjournal.com/articles/view?392#back-from-modal>, cit.11.5.2017
- [4] Stephen A. Weis, *RFID: Principles and Applications*, http://www.rfid-off.com/uploads/4/5/1/2/45128343/rfid-article_mit_usa.pdf, cit.11.5.2017
- [5] International Organization for Standardization, *ISO 18000*, <https://www.iso.org/standard/62539.html>, cit.12.5.2017
- [6] Impinj, *Different Types of RFID Systems*, <https://www.impinj.com/about-rfid/types-of-rfid-systems/>, cit.12.5.2017
- [7] Microsoft, *Windows CE 5.0*, <https://msdn.microsoft.com/en-us/library/ms924466.aspx>, cit.26.4.2017
- [8] Stefan Berka, *Windows CE*, http://www.operating-system.org/betriebssystem/_english/bs-wince.htm, cit.26.4.2017

LITERATÚRA

- [9] Ing. Mgr.art. Roman Maroš, *Smernica evidovanie odpisovanie majetku*, <http://www.stupava.sk/materialy/smernice/Smernica-evidovanie-odpisovanie-majetku-mesta-92015.pdf>, cit.8.5.2017
- [10] Kros, *Referencie Omega*, <https://www.kros.sk/11036>, cit.26.4.2017
- [11] João Paulo Figueira, *CEDB.NET*, <https://www.codeproject.com/Articles/10199/CEDB-NET>, cit.26.4.2017
- [12] Microsoft, *EDB Database Support*, [https://msdn.microsoft.com/en-us/library/ee490607\(v=winembedded.60\).aspx](https://msdn.microsoft.com/en-us/library/ee490607(v=winembedded.60).aspx), cit.26.4.2017
- [13] Nuno Lucas, *SQLite for Windows CE*, <http://sqlite-wince.sourceforge.net/>, cit.27.4.2017

Zoznam použitých skratiek

GUI Graphical user interface

XML Extensible markup language

XLS Spreadsheet file format

CSV Comma separated values

RFID Radio frequency identification

INMM Inštitút nukleárnej a molekulárnej medicíny Košice

IČ Inventárne číslo

RFID Tag RFID značka

DB Databáza

Handheld prenosné RFID čítacie zariadenie

Windows CE Windows compact edition

SQL CE Microsoft SQL Server Compact

GC Garbage collector

WMDC Windows Mobile Device Center

OS Operačný systém

RAM Random access memory - operačná pamäť

VM Virtual memory - virtuálna pamäť

A. ZOZNAM POUŽITÝCH SKRATIEK

EPC Electronic product code

LF Low frequency

HF High frequency

UHF Ultra high frequency

EAS Electronic article surveillance

Obsah priloženého CD

	readme.txt	stručný popis obsahu CD
	src	
	impl	zdrojové kódy implementácie
	thesis.....	zdrojová forma práce vo formáte L ^A T _E X
	text.....	text práce
	thesis.pdf.....	text práce vo formáte PDF