

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

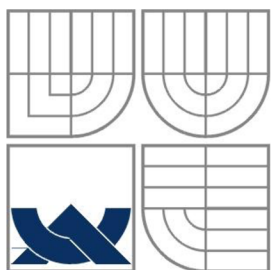
SAP MODUL PRO PLATBY FAKTUR S VYUŽITÍM QR
KÓDŮ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

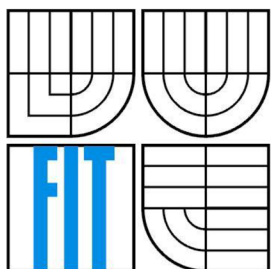
AUTOR PRÁCE
AUTHOR

Bc. BORIS ŠVANCAR

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SAP MODUL PRO PLATBY FAKTUR S VYUŽITÍM QR KÓDŮ

SAP MODUL FOR ENCODING INVOICES TO QR CODES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. BORIS ŠVANCAR

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. RADEK KOČÍ, Ph.D.

BRNO 2014

Abstrakt

Práce se věnuje kódování platebních údajů faktur do QR kódů podle standardu by square. Představený je zde systém SAP, postup kódování dat do QR kódů a standard PAY by square. Dle standardu je vytvořen funkční modul pro systém SAP v jazyce ABAP s využitím formulářů SmartForms, který umožňuje tyto QR kódy generovat na fakturu. Modul je následně podroben testování. Závěr práce tvoří zhodnocení dosažených výsledků a popis případného dalšího rozvoje.

Abstract

The master thesis deals with encoding payment data invoices into QR codes in accordance with the by square standard. The SAP system along with a process of encoding data into QR codes and by square PAY standard is introduced in my thesis. In compliance with the standard function module is designed for the SAP system which is in the ABAB language, using Smartforms form allowing the QR codes to generate the invoice. The module is then subjected to testing. The results are evaluated in conclusion of the masters thesis and a description of potential further development is assessed as a final step.

Klíčová slova

SAP, ABAP, SmartForms, QR kód, platba faktur, internetbanking, standard by square, PAY by square, XML, CRC32, komprese LZMA, konverze Base32hex

Keywords

SAP, ABAB, SmartForms, QR code, invoice payments, by square standard, PAY by square, XML, compression LZMA, conversion Base32hex

Citace

Boris Švancar: SAP Modul pro platby faktur s využitím QR kódů, diplomová práce, Brno, FIT VUT v Brně, 2014

SAP modul pro platby faktur s využitím QR kódů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Radka Kočího, Ph.D.

Další informace mi poskytli Ing. Tomáš Bartko a firma Forsys, a.s.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Bc. Boris Švancar
25. května 2014

Poděkování

Chtěl bych poděkovat svému vedoucímu Ing. Radkovi Kočí, Ph.D. za jeho ochotu, vedení firmy RKMC, s.r.o. za to, že mi umožnili zrealizovat tuto práci a firmě Forsys, a.s. za poskytnutí informací. Děkuji také rodině a přítelkyni za jejich podporu.

© Boris Švancar, 2014

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

Obsah.....	1
1 Úvod	3
2 Systém SAP	4
2.1 História SAP.....	4
2.2 Produkty a komponenty SAP	4
2.3 SAP ERP	5
2.4 ABAP Dictionary	7
2.5 Programovací jazyk ABAP a ABAP Workbench	9
2.6 SmartForms	10
2.7 Výstupné zariadenia	11
3 QR kódy	13
3.1 História čiarových kódov	13
3.2 QR kód	13
3.3 Kódovanie dát do QR kódu.....	14
3.3.1 Analýza dát.....	14
3.3.2 Kódovanie dát	15
3.3.3 Kódovanie opravy chyby	16
3.3.4 Štruktúrovanie finálnej správy	17
3.3.5 Umiestnenie modulov v matici.....	17
3.3.6 Maskovanie dát	20
3.3.7 Pridanie informácie o formáte a verzii	21
4 Štandard by square	22
4.1 Popis PAY by square.....	22
4.2 XML štruktúra so vstupnými dátami.....	23
4.3 Vytvorenie dátovej sekvencie	26
4.4 CRC32.....	29
4.5 LZMA kompresia.....	30
4.6 By square informačná hlavička	31
4.7 Konverzia Base32hex.....	31
4.8 Vytvorenie QR kódu	33
5 Návrh riešenia modulu pre systém SAP	34
5.1 Nastavenia systému	34
5.2 Návrh funkčného modulu pre vytvorenie by square sekvencie.....	34
5.3 Vytvorenie QR kódu	36
5.4 Návrh úpravy formulára a tlačového programu	36
6 Implementácia	38
6.1 Nastavenia systému	38
6.2 Funkčný modul pre generovanie by square sekvencie	39

6.2.1	Úpravy v ABAP Dictionary	39
6.2.2	Vytvorenie by square sekvencie.....	41
6.3	Generovanie QR kódu	42
6.3.1	Využitie špeciálneho výstupného zariadenia	42
6.3.2	Využitie PDF tlačiarne	44
6.3.3	Vytvorenie QR kódu samostatnou aplikáciou.....	44
6.3.4	Vytvorenie QR kódu vo formulári z matice	45
6.4	Úprava formulára a tlačového programu.....	46
7	Testovanie.....	47
7.1	Testovanie funkčného modulu pre generovanie by square sekvencie.....	47
7.2	Testovanie metód pre vytvorenie QR kódu.....	48
7.2.1	Využitie špeciálneho výstupného zariadenia	48
7.2.2	Využitie PDF tlačiarne	49
7.2.3	Vytvorenie QR kódu samostatnou aplikáciou.....	50
7.2.4	Vytvorenie QR kódu vo formulári z matice	50
7.3	Testovanie veľkosti QR kódu.....	50
7.4	Testovanie QR kódov na faktúrach mobilnými aplikáciami	51
7.4.1	Test základných údajov pre úhradu faktúry	52
7.4.2	Test kódovania viacerých bankových spojení.....	55
7.4.3	Test kódovania viacerých platobných príkazov do jedného QR kódu	56
7.4.4	Test kódovania trvalého príkazu	57
7.5	Výsledky testovania	58
7.6	Možnosti ďalšieho rozvoja.....	59
8	Záver.....	60
	Príloha A - Obsah DVD	63
	Príloha B - Uživatelská dokumentácia.....	64

1 Úvod

V dnešnej dobe je informačný systém samozrejmosťou takmer v každej firme. V mnohých ohľadoch uľahčuje a hlavne urýchľuje prácu pre všetkých ľudí vo firme. S rozšírením internetu sa naskytla možnosť prístupu ku firemnému informačnému systému aj prostredníctvom webu. Mnohé firmy začali webové rozhranie používať na to, aby umožnil prístup ku systému nie len pre zamestnancov, ale najmä na flexibilný prístup ku svojim službám pre zákazníkov. Príkladom takejto služby je internet banking alebo tiež internetové bankovníctvo. Internet banking ponúka väčšina bánk a umožňuje komunikáciu klienta s bankou prostredníctvom internetu 24 hodín denne. Prináša veľkú úsporu času ako pre klientov, tak aj pre samotné banky. Klienti majú možnosť zadávať jednorazové alebo pravidelné platby, sledovať svoje obraty a zostatok na účte a mnohé iné. Veľkú zmenu v elektronickom bankovníctve spôsobil príchod mobilných telefónov s možnosťou prístupu na internet - smartfónov. Ľudia vďaka nim môžu využiť prístup ku internet bankingu nie len cez počítač, ale kdekoľvek prostredníctvom svojho mobilného telefónu.

Potenciál smartfónov využila Slovenska banková asociácia a v spolupráci s firmou Forsys, a.s. navrhla v roku 2012 štandard s názvom by square, ktorý slúži pre kódovanie platobných údajov do QR kódov. QR kód vytvorený podľa tohto štandardu stačí pomocou aplikácie v internet bankingu pre mobilné telefóny iba oskenovať fotoaparátom telefónu a všetky platobné údaje sa v platobnom príkaze automaticky vyplnia. Na užívateľovi ostáva iba kontrola údajov a následné potvrdenie príkazu.

Krátko po nasadení štandardu do produktívnej prevádzky niekoľkými bankami začali prvé firmy posielat' svojim zákazníkom faktúry rozšírené o QR kód s platobnými údajmi. V súčasnosti však počet takýchto firiem nie je veľký, pretože na slovenskom trhu je veľmi málo informačných systémov, ktoré by podporovali priame generovanie faktúr s QR kódmi.

Nakoľko podnikový systém SAP ako jeden z celosvetovo najrozšírenejších firemných softvérov túto možnosť nepodporuje, rozhodol som vo svojej práci navrhnúť a implementovať takúto integráciu štandardu by square do systému SAP ERP.

V prvej časti tejto práce je predstavený systém SAP, jeho história a súčasť. Ďalšia časť popisuje kódovanie dát do QR kódov, štandard by square a jeho jednotlivé kroky. Tieto teoretické poznatky sú následne použité v návrhu SAP modulu umožňujúceho generovať QR kódy na základe platobných údajov z faktúry a navrhnutý modul je implementovaný v jazyku ABAP. Ku modulu je vytvorený užívateľský manuál s popisom jeho použitia. Riešenie je podrobené testovaniu a dosiahnuté výsledky spolu s možnosťami ďalšieho rozvoja sú popísané v závere práce.

2 Systém SAP

V prvej časti práce sa zoznámime so systémom SAP. Predstavíme si históriu spoločnosti SAP a jej základné produkty a komponenty. Z komponentov sa zameriame na SAP ERP, pri ktorom sa pozrieme bližšie na jeho moduly. Následne sa oboznámime s ABAP Dictionary, programovacím jazykom ABAP, so SmartForms a s nastavením tlačových zariadení, ktoré budú tvoriť dôležitý základ pre návrh modulu na tlač faktúr s QR kódmi podľa štandardu PAY by square. Informácie pre túto kapitolu som čerpal z [2], [8], [11], [13] a [14].

2.1 História SAP

Spoločnosť SAP bola založená v roku 1972 v Nemecku skupinou bývalých zamestnancov IBM. Názov SAP pochádza z angl. Systems, Applications and Products in Data Processing. Hlavným cieľom firmy bolo vyvinúť pre firmy softwarový balík, ktorý by pokryl všetky podnikové funkcie. Toto sa spoločnosti podarilo a vytvorila mnohonárodnú viacjazyčnú platformu, do ktorej je možné jednoducho zahrnúť nové podnikové procesy a postupy. Systém bol navrhnutý tak, aby ho bolo možné prevádzkovať na rôznych hardwarových platformách, operačných systémoch s využitím rôznych databáz. Takýto revolučný návrh spôsobil, že sa spoločnosť SAP za menej ako 20 rokov stala najväčším dodávateľom softvéru v Európe. V dnešnej dobe systém SAP používa viac ako milión užívateľov u viac ako 100 000 zamestnancov v 120 zemiach sveta.

2.2 Produkty a komponenty SAP

Teraz si predstavíme jednotlivé produkty a komponenty SAP. Najzákladnejším delením systému SAP je rozdelenie na SAP Business Suite a SAP NetWeaver. SAP Business Suite obsahuje všetky podnikové aplikácie a je určený pre najväčšie spoločnosti na svete. Komponenty umožňujúce prevádzku Business Suite ako napríklad webový portál, vývojové nástroje, a nástroje pre business intelligence obsahuje SAP NetWeaver. Okrem riešenia pre najväčšie firmy spoločnosť ponúka aj riešenia pre malé a stredné podniky: SAP Business One, SAP BusinessByDesign a SAP Business All-In-One, ktoré sú podmnožinou SAP Business Suite, ktorý sa skladá z 5 komponentov:

- SAP Enterprise Resource Planning (ERP)
- SAP Customer Relationship Management (CRM)
- SAP product Lifecycle Management (PLM)
- SAP Supply Chain Management (SCM)
- SAP Supplier Relationship Management (SRM)

Komponentu SAP ERP sa budeme venovať v samostatnej časti.

SAP CRM obsahuje podnikové funkcie pre podporu marketingu, predaja, služieb, podporu riadenia interaktívneho centra a poskytuje možnosť využitia webového kanálu.

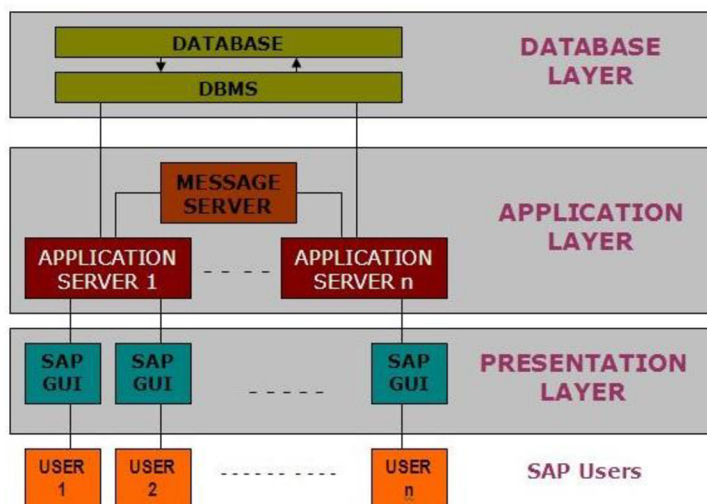
SAP PLM je vhodný pre podniky, ktoré potrebujú riešiť problém životného cyklu produktov, PLM pomáha podnikom vyvíjať nové produkty, odstraňovať organizačné obmedzenia, ktoré vedú k zníženiu produktivity a napomáha efektívnemu spojeniu odbytu, plánovania a výroby.

SAP SCM umožňuje podnikom plánovať a zjednodušiť podnikovú sieť logistiky a zdrojov, ktorých prepojením vzniká dodávateľský reťazec.

SAP SRM je určený pre riadenie a podporu procesu zaobstarania tovaru a služieb, ktoré každodenne podnik potrebuje. Umožňuje teda riadenie vzťahu podniku so svojimi dodávateľmi. Integrácia s PLM umožňuje vysoký stupeň spolupráce medzi kupcami produktov a dodávateľmi dielov.

2.3 SAP ERP

SAP ERP je základným produktom SAP. Vyvinul sa z aplikácie SAP R/3 a spadá do kategórie systémov pre online spracovanie transakcií (OnLine Transaction Processing - OTLP). Systém ERP má 3-vrstvovú klient/server architektúru, ktorá sa skladá z databázovej, aplikačnej a prezentačnej vrstvy (Obrázok 2.1). Spolu vytvárajú SAP Basis system, ktorý tvorí technologický základ SAP ERP aplikácií.



Obrázok 2.1 Architektúra SAP ERP [13]

Databázová vrstva pozostáva z databázy a systému riadenia bázy dát. Sú tu uložené všetky dáta ako obchodné dáta, systémové tabuľky, programy a iné. Databázové systémy, ktoré sú podporované sú napríklad: Oracle, Microsoft SQL Server, IBM DB/2, Siebel, Sybase, atď..

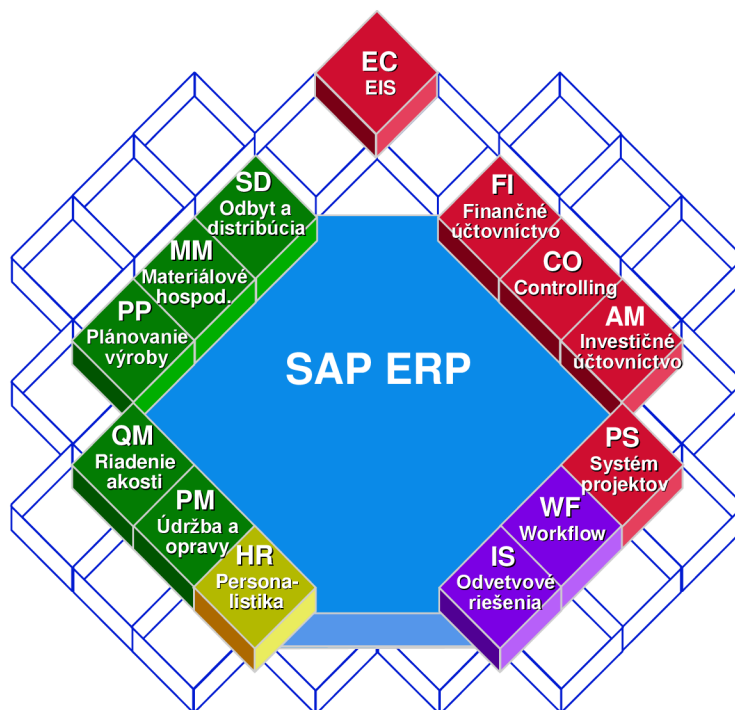
Aplikačná vrstva je miesto, kde beží SAP software. Skladá sa z jedného alebo viacerých aplikačných serverov, ktoré ponúkajú služby pre užívateľa. V prípade viacerých aplikačných serverov je za komunikáciu a vyvažovanie záťaže zodpovedný tzv. message server.

Prezentačná vrstva je software pre grafické užívateľské rozhranie - SAPgui. SAPgui prijíma vstupy od užívateľa, odosiela ich na aplikačný server a rovnako z neho dáta prijíma, formátuje ich a zobrazuje. Prezentačná vrstva je umiestnená na PC alebo notebooku užívateľa a využíva jeho zdroje (RAM, procesor, monitor, atď.).

Súčasťou SAP ERP je niekoľko riešení či podnikových scenárov:

- SAP ERP Financials (riadenie financií)
- SAP ERP Operations (prevádzkové operácie)
- SAP ERP Human Capital Management (riadenie ľudských zdrojov)
- SAP ERP Corporate Services (koncernové služby)

Každý z uvedených scenárov pozostáva z rôznych podnikových modulov. Príklady modulov sú uvedené na Obrázok 2.2.

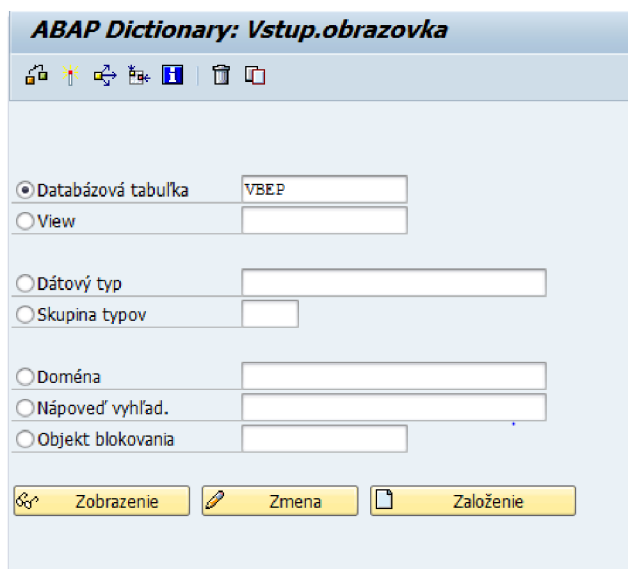


Obrázok 2.2 Príklady podnikových modulov SAP ERP [14]

Ďalej sa zameriame na tie časti SAP ERP, ktoré budú využité pri tvorbe modulu pre generovanie QR kódov podľa štandardu by square.

2.4 ABAP Dictionary

Všetky dátové definície systému SAP sú popísane a spravované cez ABAP Dictionary (SAP transakcia SE11). Najdôležitejšími objektmi uloženými v ABAP Dictionary sú domény, dátové typy, pohľady, databázové tabuľky, nápovede vyhľadávania a objekty blokovania (Obrázok 2.3).



Obrázok 2.3 ABAP Dictionary

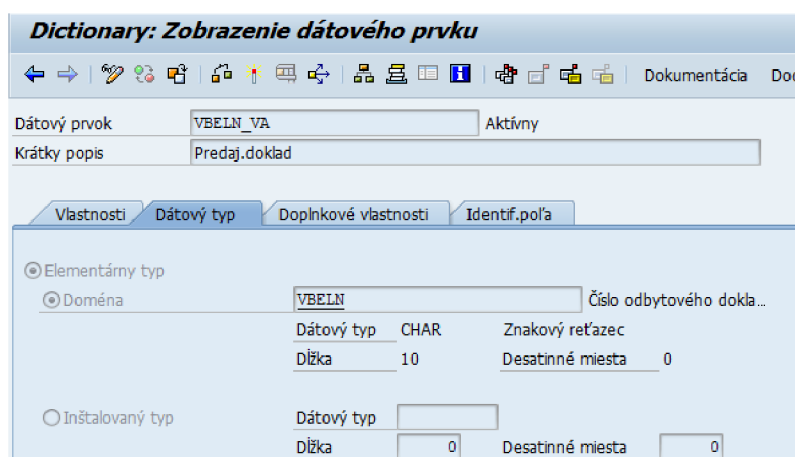
Domény slúžia na definovanie rozsahu hodnôt a sú definované jedným zo základných dátových typov a dĺžkou, prípadne počtom desatinných miest. Základné dátové typy sú definované v Tabuľka 2.1:

Dictionary type	Meaning	Maximum length
DEC	Calculation/amount field	1-31, 1-17 in tables
INT1	Single-byte integer	3
INT2	Two-byte integer	5
INT4	Four-byte integer	10
CURR	Currency field	1-17
CUKY	Currency key	5
QUAN	Amount	1-17
UNIT	Unit	2-3
PREC	Accuracy	16
FLTP	Floating point number	16
NUMC	Numeric text	1-255
CHAR	Character	1-255
LCHR	Long character	256-max

STRING.	String of variable length	1-max
RAWSTRING	Byte sequence of variable length	1-max
DATS	Date	8
ACCP	Accounting period YYYYMM	6
TIMS	Time HHMMSS	6
RAW	Byte sequence	1-255
LRAW	Long byte sequence	256-max
CLNT	Client	3
LANG	Language	internal 1, external 2

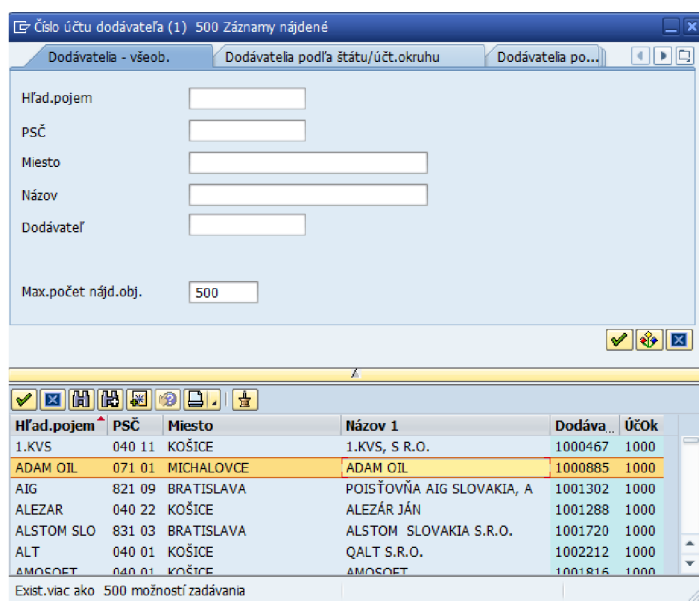
Tabuľka 2.1 Základné dátové typy SAP

Tieto domény sa využívajú pri definícii dátových elementov. Dátové elementy slúžia ďalej ako základ pre definovanie polí štruktúr a tabuliek. Taktiež môžu byť použité v ABAP programoch. Každému dátovému typu je priradený popis a jeho názov, ktorý sa použije pri zobrazení štruktúr alebo tabuliek (Obrázok 2.4).



Obrázok 2.4 Definícia dátového prvku v ABAP Dictionary

Štruktúry môžu byť vnorené do ľubovoľnej hĺbky. Tabuľkový typ definuje tzv. typ internej tabuľky, kde je ako riadok tabuľky použitá štruktúra. Tabuľky môžu byť definované v Dictionary nezávisle od databázy. V databáze sa fyzická tabuľka vytvorí až po tzv. aktivácii. Nápoveď vyhľadávania, v rámci SAPu nazývaná tiež matchcode, je štandardná funkcia systému SAP, kedy sú užívateľovi ponúknuté rôzne cesty vyhľadávania požadovanej hodnoty (Obrázok 2.5).



Obrázok 2.5 Nápvod' vyhľadávania (matchcode)

Objekty blokovania slúžia v systéme na synchronizáciu súčasného prístupu užívateľov k rovnakým dátam. V prípade, že jeden užívateľ pracuje s dátami nastaví sa zámok a nikto iný v tejto chvíli k dátam nemôže prísť. Teraz si predstavíme programovací jazyk ABAP.

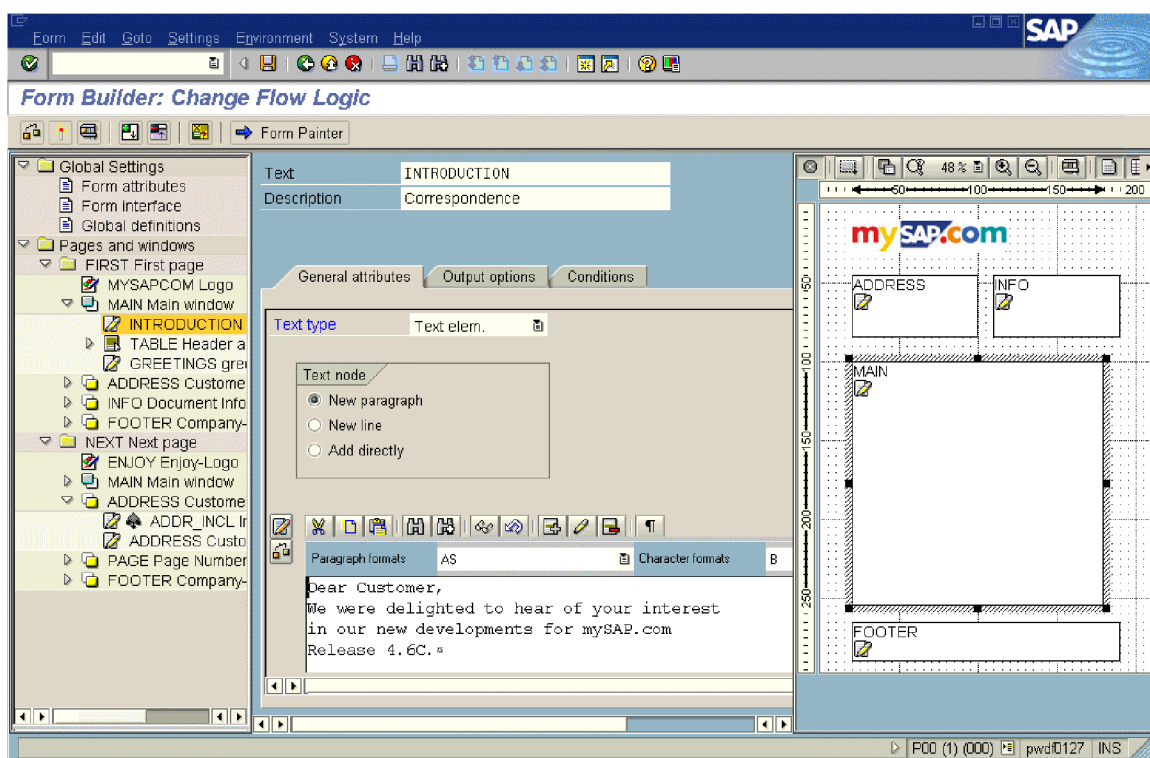
2.5 Programovací jazyk ABAP a ABAP Workbench

Základným nástrojom pre programovanie v systéme SAP je jazyk ABAP (angl. Advanced Business Application Programming). ABAP bol vyvinutý v 80. rokoch a slúžil ako základ pre programovanie v systéme R/3 a je v ňom napísaný takmer celý systém SAP ERP. Verzia ABAP Objects poskytuje možnosti objektovo - orientovaného programovania. ABAP umožňuje tvorbu spustiteľných programov ako reporty a dialógové programy. Je v ňom možné definovať typy, podprogramy, funkčné moduly, triedy, rozhrania a iné. Základný nástroj pre vývoj v jazyku ABAP je ABAP Development Workbench, ktorý poskytuje rozsiahlu sadu nástrojov, ktoré môžu vývojári použiť pri vytváraní a úprave programov systému SAP. Medzi jeho základné funkcie patrí:

- Object Navigator - prehľad a prepínanie medzi jednotlivými použitými objektmi
- ABAP editor - editor na písanie kódu v jazyku ABAP
- Function Builder - definícia funkčných modulov
- Class Builder - definícia tried
- Screen Painter - definícia obrazovky

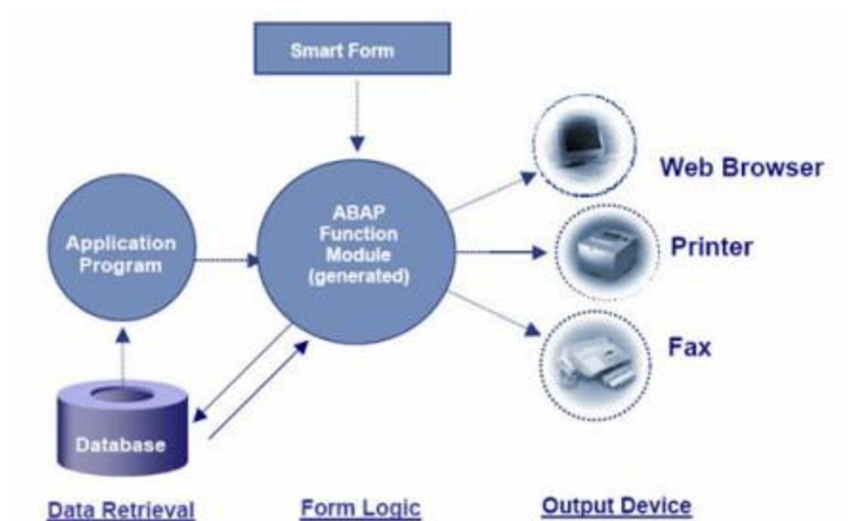
2.6 SmartForms

SAP SmartForms sú použité v systéme SAP na tvorbu tlačových formulárov ako napríklad faktúry, objednávky, dodávateľské listy atď. SmartForms boli zavedené v roku 1998 ako náhrada za SAP Script, ktorý mal množstvo obmedzení a bol zložitejší na údržbu. Veľkými výhodami SmartForms je, že pomáhajú prispôbiť formulár aj bez znalosti programovania vďaka plne grafickému užívateľskému rozhraniu cez Form Builder (Obrázok 2.6).



Obrázok 2.6 Form Builder pre vytváranie SmartForms [14]

Umožňujú vykonávať zmeny pomocou Drag & Drop, umožňujú vkladať statické a dynamické texty a tabuľky, definovať hlavičky tabuliek, medzisúčty, radenie dát a taktiež vkladať grafiku. Formuláre môžu obsahovať príkazy v jazyku ABAP a riadiace znaky tlačiarne. Rozvrhnutie oblastí na formulári je možné s využitím okien. Na definovanie štýlu textu slúžia Smart Styles zahŕňajúce napríklad odsadenie, riadkovanie, font, veľkosť, farbu a iné. Ku každému textu je tento štýl potrebné definovať. Pre posielanie dát do vnútra a z formulára je použité rozhranie formulára, v ktorom je možné definovať vstupné a výstupné parametre. Po aktivácii formulára sa automaticky vygeneruje funkčný modul na obsluhu formulára, ktorý je možné volať z akéhokolvek ABAP programu. Architektúru SmartForms popisuje Obrázok 2.7.



Obrázok 2.7 Architektúra SmartForms [11]

Ako výstupné médium je podporovaná tlačiareň, fax, e-mail alebo internet (XML) a je možné využiť aj tlač do PDF súboru. Veľkou výhodou SmartFormov je ich prenositeľnosť medzi systémami, nakoľko je ich možné exportovať do XML súboru.

2.7 Výstupné zariadenia

Pre výstup každého dokumentu musí byť v systéme SAP definované cieľové zariadenie. Na výstup dokumentov sa nemusí používať iba tlačiareň, ale napríklad aj archivačné zariadenie alebo PDF tlačiareň. Takéto zariadenia sa v SAP nazývajú výstupné zariadenia.

Každé výstupné zariadenie má priradený typ zariadenia. Systém používa informácie uložené v definícii typu zariadenia na prevod dokumentu z internej SAP znakovnej reprezentácie (spool požiadavka OTF alebo textový formát) na tlačový dátový tok konkrétneho zariadenia. Ako už názov napovedá, jedna definícia typu zariadenia môže byť použitá pre viaceré zariadenia s rovnakým typom. Atribúty typu zariadenia sú nasledovné:

- znaková sada
- riadiace znaky tlače
- formáty
- formát strany
- akcie

Znaková sada udáva kódovanie znakov vo výstupnom tlačovom toku pre zariadenie. Riadiace znaky tlače reprezentujú jednotlivé operácie tlačiarne ako zmena veľkosti písma, prechod na nový riadok a podobne. Tieto riadiace znaky sú nahradené príkazmi špecifickými pre tlačiarne pri

processe vytvárania výstupnej požiadavky. Formát určuje formát podporovaný systémom SAP. Na výber sú SAPscript formáty a ABAP list formáty. Formát strany tvorí rozhranie medzi skutočným formátom strany a SAPscript formátom. Špecifikuje rozmery papiera, ktoré sú použité na prepočet dĺžky riadka stĺpca pre prevod zo SAPscriptu. Posledným atribútom typu výstupného zariadenia sú akcie. Akcie sú príkazy pre výstupné zariadenie, ktoré sú vyžadované pre implementáciu daného formátu. Príkladom je akcia Inicializácia, ktorá definuje počet riadkov na stránke.

Vhodným nastavením týchto atribútov a definovaním príkazov v akciách je možné pomocou výstupných zariadení vykonať prevod obyčajného textu na čiarový kód.

3 QR kódy

Po zoznámení so systémom SAP sa v tejto časti budeme venovať QR kódom. Najskôr si v krátkosti predstavíme históriu čiarových kódov a následne sa budeme venovať popisu jednotlivých krokov kódovania dát do QR kódu. Informácie pre túto časť práce som čerpal z [7], [10], [15].

3.1 História čiarových kódov

Počiatky čiarových kódov siahajú až do roku 1949, kedy Norman Joseph Wooland vymyslel prvý čiarový kód, ktorý mal označovať každý tovar a urýchliť tak odbavovanie zákazníkov. Hoci patent na čiarový kód bol vydaný už 7. októbra 1952 trvalo viac ako 20 rokov kým došlo k jeho prvému komerčnému nasadeniu. Prvý kus tovaru s čiarovým kódom bol predaný v roku 1974 a išlo o balík žuvačiek. K masovému nasadeniu čiarového kódu v obchodoch došlo až v 80 rokoch, čo prospelo vylepšeniu štandardov a čítačiek. V súčasnosti je používanie čiarových kódov jedna z najpresnejších a najrýchlejších metód k registrácii väčšieho množstva údajov, pričom existuje veľké množstvo rôznych druhov čiarových kódov používaných v rôznych odvetviach. Najčastejšie sa vyskytujúcim čiarovým kódom je EAN-13, ktorým sa označujú jednotlivé druhy tovaru.



Obrázok 3.1 Čiarový kód EAN-13

Čoraz častejšie sa však na výrobkoch a dokumentoch vyskytujú tzv. 2D kódy. Ich veľkou výhodou je to, že dokážu pojať niekoľkonásobne viac informácií ako čiarové kódy na rovnakej alebo aj menšej ploche. Najtypickejším zástupcom 2D kódov je QR kód, ktorý si bližšie predstavíme.

3.2 QR kód

QR kód je dvojrozmerný štvorcový kód zložený z malých bielych a čiernych plôch (Obrázok 3.2) a jeho názov pochádza z anglického Quick Response (rýchla odpoveď), keďže je navrhnutý s ohľadom na rýchle dekódovanie. QR kód vyvinula v roku 1994 japonská spoločnosť Denso-Wave na označovanie súčiastok pri výrobe automobilov, no veľmi rýchlo sa rozšíril do rôznych

oblastí spoločnosti. V júni 2000 sa špecifikácia stala štandardom ISO 18004:2000 a v septembri 2006 bol štandard upravený podľa špecifikácie QR Code 2005 na ISO 18004:2006. Princíp QR kódu spočíva v tom, že QR kód sa zosníma kamerou alebo fotoaparátom mobilného telefónu a softvér dekóduje informáciu uloženú v kóde. Kód môže niesť akúkoľvek informáciu, no najčastejšie sa do neho kóduje webová alebo emailová adresa, kontaktné údaje alebo ako v našom prípade platobné údaje.



Obrázok 3.2 QR kód [6]

3.3 Kódovanie dát do QR kódu

Kódovanie dát do QR kódu je veľmi rozsiahly proces pozostávajúci z týchto krokov:

- analýza dát
- kódovanie dát
- kódovanie opravy chyby
- štruktúrovanie finálnej správy
- umiestenie modulov v matici
- maskovanie dát
- pridanie informácie o formáte a verzii

Teraz si popíšeme najdôležitejšie informácie ku jednotlivým krokom.

3.3.1 Analýza dát

Analýza dát slúži na výber kódovacieho režimu pre zakódovanie textového reťazca do QR kódu. Štandard definuje 4 kódovacie módy pre text popísané v Tabuľka 3.1:

Mode	Max. chars	Bits/char	Possible characters, default encoding
Numeric	7089	3 1/3	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Alphanumeric	4296	5 1/2	0-9, A-Z (upper-case only), space, \$, %, *, +, -, ., /, :
Binary/byte	2953	8	ISO 8859-1
Kanji	1817	13	Shift JIS X 0208

Tabuľka 3.1 Kódovacie módy QR kódu

Každý mód je optimalizovaný, aby generoval čo najkratšiu postupnosť pre reťazce obsahujúce dané znaky. Výber najefektívnejšieho módu sa riadi podmienkami:

1. Ak vstup obsahuje iba číslice 0 - 9 použije sa numerický mód.
2. Ak reťazec obsahuje znaky 0-9, A-Z, medzera, \$, %, *, +, -, ., /, : použije sa alfanumerický mód. Tento mód môže byť použitý iba pre veľké písmená.
3. Ak reťazec obsahuje znak, ktorý nie je možné použiť v alfanumerickom móde, ale môže byť zakódovaný v ISO 8859-1 použije sa bytový mód
4. Ak všetky znaky reťazca sú v Shift JIS znakovej sade (Kanji znaky) použije sa Kanji mód.

3.3.2 Kódovanie dát

Po výbere kódovacieho módu je na rade samotné kódovanie, pre ktoré treba zvoliť úroveň opravy chyby a najmenšiu verziu kódu pre dáta. Úroveň opravy chyby určuje koľko dát môže byť maximálne opravené ak dáta nie sú korektné načítané. Jednotlivé úrovne uvádza Tabuľka 3.2:

Error Correction Level	Error Correction Capability
L	Recovers 7% of data
M	Recovers 15% of data
Q	Recovers 25% of data
H	Recovers 30% of data

Tabuľka 3.2 Úroveň opravy chyby

Rôzne veľkosti QR kódov sú nazývané verzie (Obrázok 3.3). Dostupných je 40 verzií s rôznymi veľkosťami a kapacitami závislými na kódovacom móde a úrovni opravy chyby. Celková prehľadová tabuľka je k dispozícii v [10].



Obrázok 3.3 Ukážka verzií QR kódov [10]

Pre kódovanie je potrebné vybrať najnižšiu verziu, ktorej kapacita postačuje na uloženie dát. Po výbere sa začína s kódovaním dát. Kódované dáta musia začínať indikátorom zvoleného kódovacieho módu (numeric - 0001, alphanumeric - 0010, byte - 0100, kanji - 1000) a indikátorom počtu znakov, ktorý udáva počet znakov, ktoré budú kódované. Bitová dĺžka indikátora počtu znakov závisí od módu a verzie a zobrazuje ju Tabuľka 3.3:

Mode	Version 1 - 9	Version 10 - 26	Version 27 - 40
Numeric	10 bits	12 bits	14 bits
Alphanumeric	9 bits	11 bits	13 bits
Byte	8 bits	16 bits	16 bits
Kanji	8 bits	10 bits	12 bits

Tabuľka 3.3 Bitová dĺžka indikátora počtu znakov podľa módu a verzie

Tieto dva indikátory sú nasledované dátami zakódovanými príslušným módom. Presný popis kódovania dát je možné nájsť v [7] a [15]. Keďže špecifikácia vyžaduje, aby bola kapacita QR kódu vyplnená, v prípade, že tak nie je, je doplnený oddeľovač z núl. Maximálne však dĺžky 4. Ak ani po pridaní oddeľovača nie je kapacita vyplnená pridá sa taký počet núl, aby bola dĺžka vytvorených dát deliteľná 8. Ak stále nie je vyplnená celá kapacita kódu na jej doplnenie sa použije opakovanie bytov 11101100 00010001 resp. 236 a 17. Týmto sú dáta pripravené a je potrebné vygenerovať kódové slová opravujúce chyby.

3.3.3 Kódovanie opravy chyby

Kódové slová opravy chýb, umožňujú detekciu a opravu chyby v príslušnej časti kódu. Pred výpočtom musia byť dáta rozdelené do skupín a v rámci skupín do blokov. Zabezpečenie blokov pomocou K zabezpečovacích bytov je zaistené Reed-Solomonovými kódmi nad telesom $GF(2^8)$,

kde prvky telesa sú stotožnené s bytami tak, že bitový zápis je vnímaný ako koeficienty polynómov nad bitmi a tieto polynómy sú brané modulo polynóm s bitovými koeficientmi 100011101. Reed-Solomonove kódy vzniknú doplnením polynómu, kde najvyššie rady budú tvorené správou, s využitím takých koeficientov, aby bol výsledný polynóm deliteľný:

$$(x - \alpha^0) * (x - \alpha^1) * \dots * (x - \alpha^{(K-1)}), \text{ kde } \alpha = 2$$

Podrobný popis kódovania opravy chyby je v [7] a [15].

3.3.4 Štruktúrovanie finálnej správy

Po tom ako máme dáta a ich zodpovedajúce opravovacie kódy je potrebné určiť štruktúru výslednej správy. Štruktúrovanie je potrebné spraviť iba pre QR kódy, ktoré pozostávajú z viac ako jedného bloku dát. Prekladanie blokov sa vykonáva v tomto poradí: 1. dáta z 1. bloku, 1. dáta z 2. bloku až 1. dáta z posledného bloku, 2. dáta z 1. bloku a tak ďalej, kým sa nepoužijú všetky dáta všetkých blokov. V rovnakom poradí sa za dáta ukladajú aj kódové slová opravy chýb. Výsledná správa sa prevedie do 8-bitovej binárnej podoby, teda každé číslo sa prevedie na 8-bitové binárne číslo a správa sa spojí do jedného súvislého reťazca. Pre určité verzie kódov je potrebné doplniť takzvané zvyškové nulové bity, aby mala správa správnu dĺžku. Počet vyžadovaných zvyškových bitov je uvedený v Tabuľka 3.4.

Required Remainder Bits	0	3	4	7
QR Version	1, 7, 8, 9, 10, 11, 12, 13, 35, 36, 37, 38, 39, 40	14, 15, 16, 17, 18, 19, 20, 28, 29, 30, 31, 32, 33, 34	21, 22, 23, 24, 25, 26, 27	2, 3, 4, 5, 6

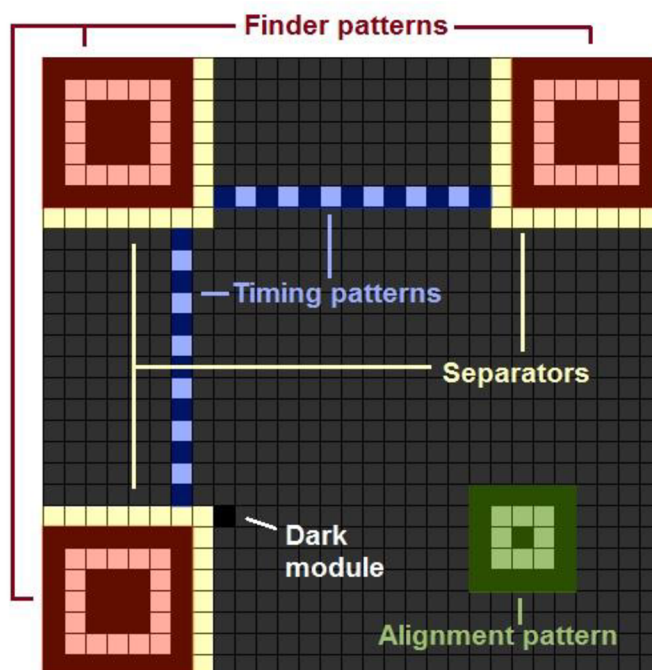
Tabuľka 3.4 Počet požadovaných zvyškových bitov

3.3.5 Umiestnenie modulov v matici

Keďže finálnu bitovú správu máme vytvorenú môžeme prejsť k jej umiestňovaniu do matice. Okrem tejto správy však QR kód obsahuje tzv. function patterns (Obrázok 3.4), ktoré sú:

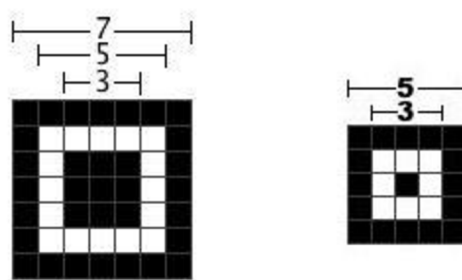
- Finder patterns
- Separators
- Alignment patterns
- Timing patterns

- Dark module



Obrázok 3.4 Umiestnenie function patterns v QR kóde [15]

Finder patterns pozostávajú z modulov veľkosti 7x7 a majú pre každú verziu QR kódu rovnaký tvar (Obrázok 3.5). Vždy sú v QR kóde 3 a sú umiestnené v rohoch (v ľavom hornom, pravom hornom a ľavom dolnom). Slúžia na určenie správnej orientácie kódu.



Obrázok 3.5 Finder pattern a Alignment pattern [15]

Separators sú čiary z bielych modulov umiestnené okolo finder patterns a oddeľujú ich od zvyšku kódu.

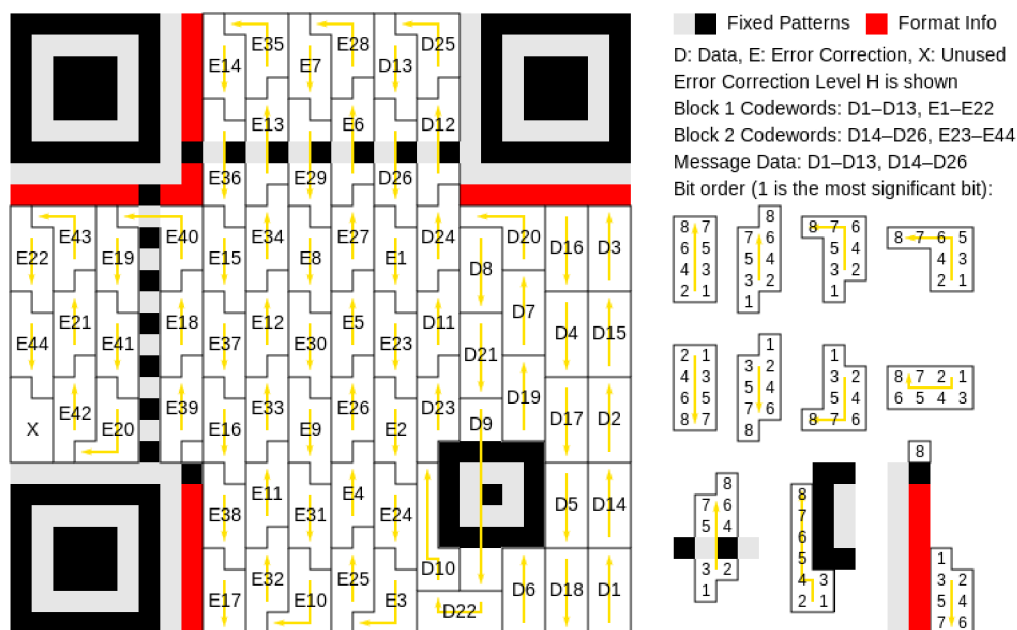
Alignment patterns sú vyžadované od verzie kódu 2 a vyššie a pozostávajú z 5x5 modulov čiernej farby vnútri s 3x3 modulmi bielej farby s čiernym modulom v strede (Obrázok 3.5). Ich pozícia je definovaná pre každú verziu kódu na str. 82 v [7].

Timing patterns sú dve čiary striedajúcich sa čiernych a bielych modulov. Ležia v šiestom riadku a šiestom stĺpci medzi separátormi. Vždy začínajú a končia čiernymi modulmi. Timing patterns môžu byť prekryté alignment patterns, pričom ich tmavé a svetlé moduly korešpondujú.

Všetky kódy musia mať dark module, ktorý je umiestnený vždy na súradnici:

$$((4 * V) + 9), 8), kde V je verzia QR kódu$$

Po umiestnení týchto vzorov je potrebné rezervovať miesto na informáciu o formáte a verzii a môžeme prejsť k umiestneniu samotných dát - bitov správy. Dáta sú umiestňované od spodného pravého rohu smerom hore v stĺpci širokom 2 moduly. Keď sa dosiahne okraj matice prechádza sa do vedľajšieho stĺpca a mení sa smer ukladania. Ak je príslušný modul už použitý alebo rezervovaný dáta sa umiestnia do nasledujúceho najbližšieho voľného modulu. Poradie je ukázané na Obrázok 3.6.

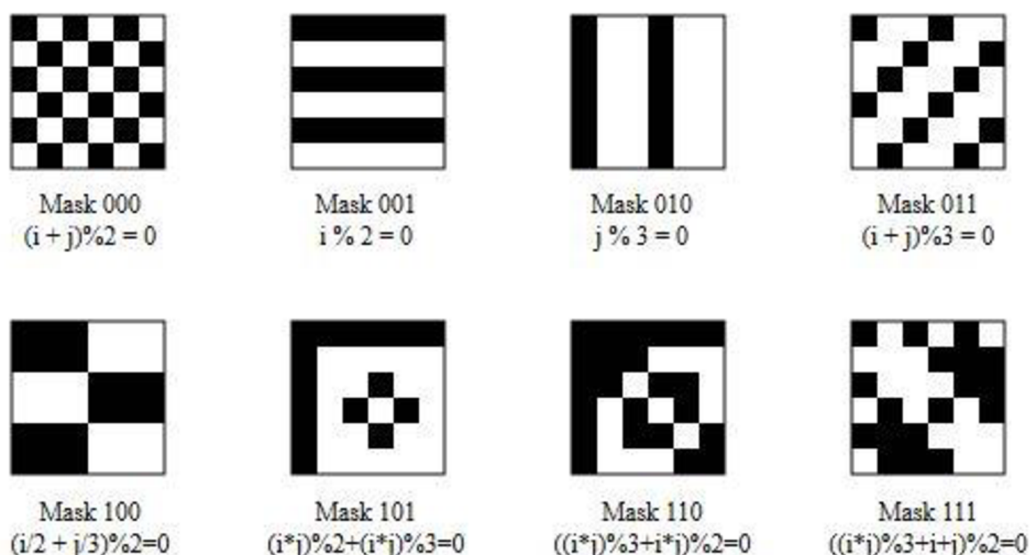


Obrázok 3.6 Poradie ukladania bitov správy do matice [10]

Po uložení modulov v matici je potrebné určiť najlepší maskovací vzor.

3.3.6 Maskovanie dát

Účelom maskovania je spraviť QR kód čo najčitateľnejší pre čítačky. Maskovanie modulu znamená zmenu jeho hodnoty na opačnú. Maskovacie vzory sú aplikované iba na moduly s bitmi správy nie na function patterns ani na rezervovane oblasti. Štandard pre QR kód definuje 8 rôznych maskovacích vzorov ukázaných na Obrázok 3.7.



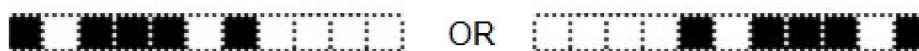
Obrázok 3.7 Maskovacie vzory pre QR kód [10]

Pre každú masku sa počíta tzv. penalty score na základe 4 pravidiel.

Prvé pravidlo počíta penalty score na základe výskytu skupiny 5 alebo viacerých modulov rovnakej farby. Za 5 modulov je penalty 3 a za každý modul na viac plus 1 ku penalty.

Druhé pravidlo počíta na základe výskytu modulov rovnakej farby s rozmermi 2x2 alebo väčšími. Penalty score sa pre blok $m \times n$ počíta ako $3 * (m - 1) * (n - 1)$.

Tretie pravidlo je založené na hľadani dvoch presne určených vzorov, a to konkrétne



Obrázok 3.8 Vzory použité pri výpočte penalty score [15]

Každý výskyt takéhoto vzoru zvyšuje penalty o 40.

Posledné pravidlo je založené na pomere bielych a čiernych modulov. Spočítajú sa tmavé moduly a vypočíta sa percento tmavých modulov v matici. Pre percento sa určí predchádzajúci a nasledujúci násobok čísla 5. Od oboch sa odpočíta 50 a určí sa absolútna hodnota. Tieto čísla sa vydedia 5.

Výsledok penalty score je menšie z týchto dvoch čísel vynásobené 10. Pre lepšie pochopenie si uvedieme príklad.

Príklad:

Percentuálny podiel tmavých modulov je 37%. Predchádzajúci násobok 5 je 35 a nasledujúci 40.

$$|35-50| = 15 \text{ a } |40-50| = 10$$

$$15/5 = 3 \text{ a } 10/5 = 2$$

Menšie z čísiel je 2 a výsledné penalty score je teda $2 * 10 = 20$.

Penalty score sa pre všetky 4 pravidlá sčíta a pre maskovanie kódu sa vyberie tá maska, pre ktorú je celkové skóre najnižšie.

3.3.7 Pridanie informácie o formáte a verzii

Posledným krokom pri vytváraní QR kódu je doplnenie informácie o formáte a verzii. Informácia o formáte zahŕňa úroveň opravy chyby a použitú masku a je kódovaná na 15 bitoch podľa postupu uvedeného na str. 76 v [7].

Informácia o verzii je kódovaná na 18 bitoch a zodpovedajúce hodnoty sú určené tabuľkou na str. 79 v [7]. Tieto informácie sú pridané namiesto rezervovaných oblastí, čím je vytvorený výsledný QR kód.

4 Štandard by square

V tejto kapitole bude predstavený štandard by square, ktorý popisuje kódovanie a dekódovanie platobných informácií do QR Code 2005. Veľkou výhodou používania kódov by square je úspora času a odstránenie možnosti chyby v platobných a fakturačných údajoch, nakoľko údaje sú načítané automaticky bez nutnosti manuálneho prepisovania.

Štandard by square vytvorila firma Forsys, a.s. pre Slovenskú bankovú asociáciu (SBA) a slúži pre všetkých jej členov. Bol vytvorený v roku 2012, no v reálnej prevádzke je od roku 2013. Hoci je v súčasnosti členom SBA 28 bánk by square vo svojich mobilných aplikáciách používa iba 6 bánk: Slovenská sporiteľňa, VÚB, Tatra banka, ČSOB, Prima banka a UniCredit bank. Podľa zástupcov firmy Forsys a.s. ostatné banky vôbec nemajú mobilnú aplikáciu a prípadne nasadenie by square zvažujú. S rastúcim počtom faktúr s QR kódmi v obehú sa však predpokladá, že implementácia tohto štandardu sa stane prioritou aj pre tieto banky.

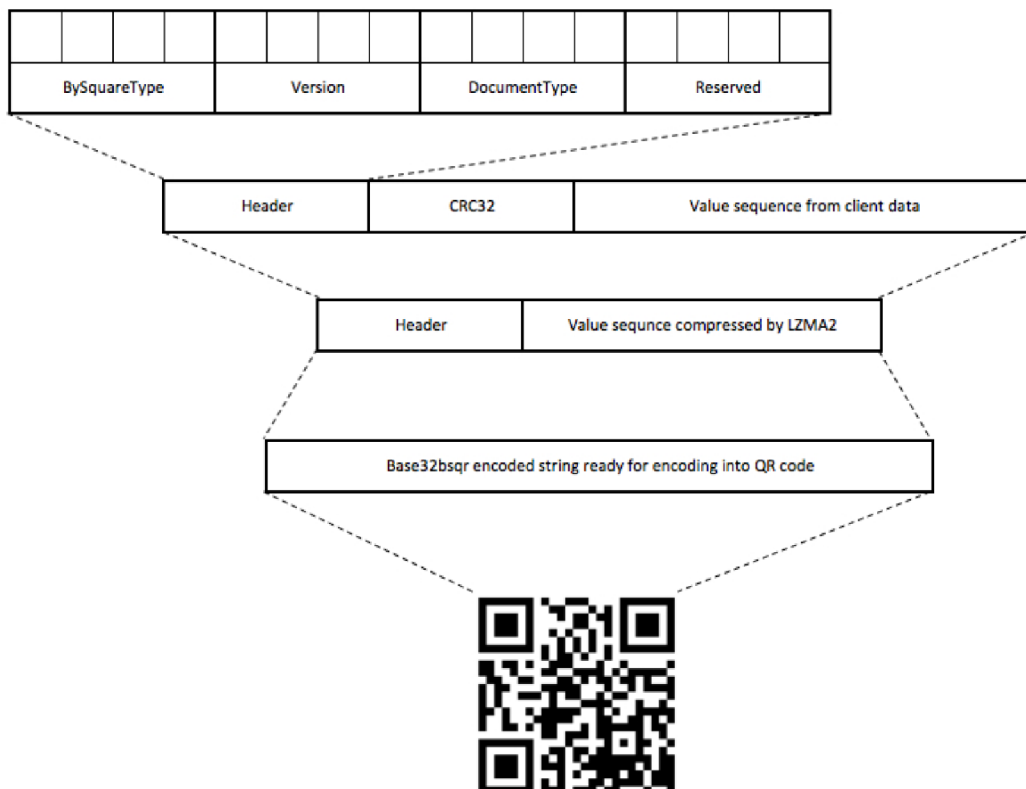
By square zahŕňa 2 produkty: PAY by square a INVOICE by square. PAY by square obsahuje údaje potrebné na vytvorenie platobného príkazu. Konkrétne môže ísť o platobný príkaz, hromadný platobný príkaz, trvalý príkaz a inkaso. Špecifikácia štandardu PAY by square je voľne prístupná.

INVOICE by square slúži na kódovanie a dekódovanie údajov potrebných na zaúčtovanie faktúr ako sú všeobecné fakturačné údaje, sumár faktúry a jednu položku faktúry. INVOICE je možné použiť ako faktúru, dobropis, ťarchopis, proforma faktúru alebo zálohovú faktúru. Produkt INVOICE by square zatiaľ nie je v reálnej prevádzke rozšírený a jeho špecifikácia nie je zverejnená, preto sa ďalej budem zaoberať iba štandardom PAY by SQUARE. Ako hlavný zdroj informácií pre túto kapitolu slúžilo osobné stretnutie s tvorcami štandardu by square z firmy Forsys, a.s. a zdroje [3], [4], [5], [6], [7], [15].

4.1 Popis PAY by square

Štandard PAY by square definuje formát dát, proces kódovania a dekódovania platobných príkazov do QR kódu. Platobné údaje sú uvedené podľa SEPA. SEPA (Single Euro Payments Area) predstavuje zjednotenie pravidiel pri realizácii bezhotovostných platieb v eurách v rámci krajín Európskej únie. Hlavným prínosom SEPA je odstránenie rozdielov medzi zahraničnými a tuzemskými platbami. Vstupné užívateľské dáta sú uložené vo formáte XML a na ich validáciu je definovaná XSD schéma. Kódovanie dát z XML súboru do QR Code 2005 následne prebieha v niekoľkých krokoch (Obrázok 4.1).

Prvým krokom je vytvorenie dátovej sekvencie z užívateľských dát. Pre dátovú sekvenciu sa následne vypočíta CRC32, ktoré sa pridá pred sekvenciu. Tieto dáta sú následne skomprimované algoritmom LZMA a doplnené o by square hlavičku. Predposledným krokom je prekódovanie dát konverziou Base32hex. Tieto dáta sa použijú ako vstup pre vytvorenie výsledného QR kódu. V nasledujúcich podkapitolách si predstavíme detailný popis jednotlivých krokov.



Obrázok 4.1 Proces kódovania PAY by square [6]

4.2 XML štruktúra so vstupnými dátami

Vstupné dáta pre kódovanie do QR kódu podľa štandardu PAY by square sú uvedené v XML súbore. Predstavíme si základnú štruktúru XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<Pay xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Pay"
xmlns="http://www.bysquare.com/bysquare">
  <InvoiceID minOccurs="0"/>
  <Payments>
    <Payment maxOccurs="unbounded">
      <PaymentOptions/>
      <Amount minOccurs="0"/>
      <CurrencyCode/>
      <PaymentDueDate minOccurs="0"/>
      <VariableSymbol minOccurs="0"/>
      <ConstantSymbol minOccurs="0"/>
      <SpecificSymbol minOccurs="0"/>
      <OriginatorsReferenceInformation minOccurs="0"/>
      <PaymentNote minOccurs="0"/>
      <BankAccounts>

```

```

        <BankAccount maxOccurs="unbounded">
            <IBAN/>
            <BIC/>
        </BankAccount>
    </BankAccounts>
    <StandingOrderExt minOccurs="0">
        <Day minOccurs="0"/>
        <Month minOccurs="0"/>
        <Periodicity/>
        <LastDate minOccurs="0"/>
    </StandingOrderExt>
    <DirectDebitExt minOccurs="0">
        <DirectDebitScheme/>
        <DirectDebitType minOccurs="0"/>
        <VariableSymbol minOccurs="0"/>
        <SpecificSymbol minOccurs="0"/>
        <OriginatorsReferenceInformation/>
        <MandateID/>
        <CreditorID/>
        <ContractID minOccurs="0"/>
        <MaxAmount minOccurs="0"/>
        <ValidTillDate minOccurs="0"/>
    </DirectDebitExt>
</Payment>
</Payments>
</Pay>

```

Bližšie si popíšeme význam jednotlivých tagov a ich povolené hodnoty. Hlavným tagom je <Pay>, ktorý označuje dokument platobného QR kódu podľa štandardu by square. Ak sú údaje súčasťou faktúry, číslo faktúry sa uvedie v <InvoiceID>, táto hodnota môže byť použitá tiež pre interné potreby vystavovateľa. Jednotlivé platby sú uvedené v <Payments>. Tag označujúci jeden platobný príkaz je <Payment>. Štandard PAY by square umožňuje kódovanie 3 typov platobných príkazov <PaymentOptions>:

- paymentorder - platobný príkaz
- standingorder - trvalý príkaz, údaje sa vyplnia do StandingOrderExt
- directdebit - inkaso, údaje sa vyplnia do DirectDebitExt

Tieto typy je možné v jednom platobnom príkaze kombinovať a na ich oddelenie sa používa medzera. Čiastka platby sa uvádza ako kladné číslo s bodkou ako oddeľovačom desatinnej časti v <Amount>. V prípade, že ide o dobrovoľný príspevok hodnota nemusí byť vyplnená. Mena platby <CurrencyCode> sa uvádza ako trojpísmenová skratka vo formáte ISO 4217 (napr. EUR, CZK, ...). Nepovinný údaj dátum splatnosti <PaymentDueDate> je uvedený vo formáte RRRR-MM-DD. Pri trvalom príkaze je tento dátum dňom prvej platby. Nasledujú tagy pre variabilný, konštantný a špecifický symbol <VariableSymbol>, <ConstantSymbol> a <SpecificSymbol>. Podľa SEPA sa však používa E2E referencia platiteľa <OriginatorsReferenceInformation>. Služi na identifikáciu platby a je obdobou variabilného symbolu, konštantného symbolu a špecifického symbolu. V rámci Slovenskej republiky sa odporúča uvádzať referenciu v tvare

VS[N10]/SS[N10]/KS[N4], napr. VS1234567890/SS1234567890/KS1234. Správa pre prijímateľa je definovaná v <PaymentNote>. Podľa štandardu PAY by square sa odporúča používať maximálne 140 Unicode znakov a správu uvádzať v tvare: *prijemca: predmet platby - meno klienta - dátum* napr. Telecom: internet - Svancar - 2014/01. Zoznam bankových účtov príjemcov je uvedený tagom <BankAccounts>. <BankAccount> reprezentuje jeden bankový účet v medzinárodnom formáte. Účet je definovaný medzinárodným číslom bankového účtu vo formáte IBAN <IBAN> a medzinárodným bankovým identifikačným kódom BIC (z angl. Bank Identification Code) <BIC>.

V prípade, že platobný príkaz obsahuje rozšírenie platobných údajov o údaje pre nastavenie trvalého príkazu, tento je uvedený <StandingOrderExt>. <Day> uvádza deň platby v závislosti na periodicite. V prípade dňa v týždni 1 = pondelok - 7 = nedeľa, pri mesiaci číslo od 1 do 31. Zoznam mesiacov, v ktorých sa má platba uskutočniť <Month> sa uvádza ako zoznam čísel mesiacov oddelených medzerou. Zodpovedajúce čísla mesiacov sú uvedené v Tabuľka 4.1. Periodicita opakovania príkazu sa uvádza ako jedno písmeno podľa Tabuľka 4.2. Na vymedzenie platnosti trvalého príkazu sa používa dátum poslednej platby v trvalom príkaze <LastDate>.

Month	Value
January	1
February	2
March	4
April	8
May	16
June	32
July	64
August	128
September	256
October	512
November	1024
December	2048

Tabuľka 4.1 Číselné hodnoty zodpovedajúce mesiacom

Rozšírenie platobných údajov o údaje pre nastavenie a identifikáciu inkasa je v <DirectDebitExt>. Inkaso môže používať jednu z dvoch schém <DirectDebitScheme>: SEPA - Inkaso zodpovedá schéme SEPA. other - iné. Typ inkasa <DirectDebitType> je jednorazové inkaso - one-off alebo opakované inkaso - recurrent. Hodnoty <VariableSymbol> a <SpecificSymbol> sa vyplňajú len v prípade, ak sa odlišujú od hodnôt VS a ŠS v platobnom príkaze. <OriginatorsReferenceInformation>

Periodicity	Value
Daily	d
Weekly	w
Biweekly	b
Monthly	m
Bimonthly	B
Quarterly	q
Semiannually	s
Annually	a

Tabuľka 4.2 Hodnoty zodpovedajúce periodicite

Pri SEPA inkase sa na definovanie inkasa používa mandát <MandateID>, čo je písomná zmluva medzi platiteľom a príjemcom platby. Na jednoznačnú identifikáciu príjemcu inkasa sa používa Creditor Identifier <CreditorID>. Zmluva medzi veriteľom a dlžníkom podľa SEPA je identifikovaná v <ContractID>. Pre inkaso je uvedená maximálna čiastka inkasa <MaxAmount>. Posledným údajom pri inkase je deň zániku platnosti inkasa <ValidTillDate>.

4.3 Vytvorenie dátovej sekvencie

Dáta obsiahnuté v XML súbore sú v procese kódovania predspracované a zoradené do dátovej sekvencie. Pri predspracovaní sú všetky dátumové polia v tvare RRRR-MM-DD prevedené na tvar RRRRMMDD. Desatinné čísla sú upravené tak, aby sa ako oddeľovač desiatin použila bodka. Typ platobného príkazu, schéma inkasa a typ inkasa sa nahradí podľa Tabuľka 4.3. Polia obsahujúce viacnásobné hodnoty ako napr. PaymentOptions alebo Month sú zosumované a nahradené jednou hodnotou. Príkladom je: <PaymentOptions>paymentorder standingorder</PaymentOptions> je nahradené hodnotou $1(\text{paymentorder}) + 2(\text{standingorder}) = 3$.

Payment options	Value
paymentorder	1
standingorder	2
directdebit	4
Direct debit type	Value
one-off	0
recurrent	1
Direct debit scheme	Value
other	0
SEPA	1

Tabuľka 4.3 Zodpovedajúce číselné hodnoty

Z predspracovaných dát sa vytvorí dátová sekvencie za použitia niekoľkých pravidiel.

- Poradie polí v sekvencie je definované v Tabuľka 4.4. V prípade, že niektoré voliteľné pole nie je použité, je v sekvencii uvažované ako keby malo dĺžku 0.
- Ako oddeľovač jednotlivých polí je použitý ASCII znak 9 - horizontálny tabulátor. Aby bolo zaručené, že sekvencia bude správna musia byť všetky tabulátory nájdené v hodnotách polí nahradené za znak medzery. To zaručí správnu rekonštrukciu poradia polí pri dekódovaní. Táto zmena sa týka najmä poľa so správou pre prijímateľa. Výskyt poľa s nulovou dĺžkou sa prejaví ako dva po sebe idúce tabulátory.
- Polia s atribútom *maxOccurs="unbounded"* sa môžu v sekvencii opakovať. Pred prvým výskytom je preto do sekvencie potrebné uviesť číslo udávajúce počet opakovaní oddelený tabulátorom. Číslo slúži pre dekóder, aby vedel určiť aký počet výskytov daného prvku sa má na vstupe očakávať. Pravidlo sa týka elementov Payment a BankAccount.
- Ak má pole komplexného typu atribút *minOccurs="0"* znamená to, že celý komplexný element nemusí byť v sekvencii zahrnutý. Pravidlo sa týka komplexných elementov StandingOrderExt a DirectDebitExt. Pred pridaním takéhoto elementu do sekvencie musí byť uvedené tabulátorom oddelené číslo 1 alebo 0 udávajúce jeho výskyt. Číslo 0 znamená, že všetky polia komplexného elementu sú ignorované a v sekvencii sa pokračuje nasledujúcim prvkom, teda v prípade že StandingOrderExt = 0 preskakujú sa všetky polia Day, Month, Periodicity a LastDate a pokračuje sa prvkom DirectDebitExt.

Order	Field name	Type	Priority	Max. length
1	InvoiceID	string	2	10
2	Payments (count)	integer	999	1
3	PaymentOptions	integer	999	15
4	Amount	decimal	999	3
5	CurrencyCode	string	999	8
6	PaymentDueDate	date	999	10
7	VariableSymbol	string	7	4
8	ConstantSymbol	string	5	10
9	SpecificSymbol	string	6	35
10	OriginatorsReferenceInformation	string	12	140
11	PaymentNote	string	1	34
12	BankAccounts (count)	integer	999	11
13	IBAN	string	999	1
14	BIC	string	999	2
15	StandingOrderExt	integer	999	4
16	Day	integer	999	1
17	Month	integer	999	8

18	Periodicity	string	999	1
19	LastDate	date	999	1
20	DirectDebitExt	integer	999	1
21	DirectDebitScheme	integer	999	10
22	DirectDebitType	integer	999	10
23	VariableSymbol	string	4	35
24	SpecificSymbol	string	3	35
25	OriginatorsReferenceInformation	string	11	35
26	MandateID	string	10	35
27	CreditorID	string	9	15
28	ContractID	string	8	8
29	MaxAmount	decimal	999	10
30	ValidTillDate	date	999	1

Tabuľka 4.4 Poradie polí v dátovej sekvencii

Tabuľka 4.4 obsahuje okrem iného aj stĺpce priority a max length, ktoré majú súvis s maximálnou dĺžkou dátovej sekvencie. Stĺpec max. length uvádza maximálnu odporúčenú dĺžku poľa v sekvencii. Celková dĺžka sekvencie sa odvíja od kapacity QR kódu a tvorcovia štandardu by square určili jej dĺžku na 550 UTF-8 znakov vrátane oddeľovačov. V prípade, že sa táto dĺžka prekročí sú postupne skracované, resp. vynechávané polia v poradí udanom stĺpcom priority. Ako môžeme vidieť, ako prvá je vynechávaná správa pre prijímateľa a ďalej polia, ktoré nemajú takmer žiaden vplyv na platobné údaje. V reáli sa však skracovanie neuplatňuje nakoľko 550 znakov postačuje na zakódovanie 5 platobných príkazov s viacerými bankovými účtami. Najčastejšie sa však využíva zakódovanie jedného platobného príkazu s jedným alebo viacerými bankovými účtami. Ukážeme si príklad vytvorenia dátovej sekvencie z takéhoto príkazu.

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<Pay xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Pay"
xmlns="http://www.bysquare.com/bysquare">
  <InvoiceID>001</InvoiceID>
  <Payments>
    <Payment>
      <PaymentOptions>paymentorder</PaymentOptions>
      <Amount>20.35</Amount>
      <CurrencyCode>EUR</CurrencyCode>
      <PaymentDueDate>2013-12-06</PaymentDueDate>
      <VariableSymbol>1200097151</VariableSymbol>
      <ConstantSymbol>0308</ConstantSymbol>
      <SpecificSymbol>012014</SpecificSymbol>
      <PaymentNote>UPC: internet - 2014/01</PaymentNote>
      <BankAccounts>
        <BankAccount>
          <IBAN>SK7911000000002628204091</IBAN>
        </BankAccount>
      </BankAccounts>
    </Payment>
  </Payments>
</Pay>
```



```

        <BIC>TATRSKBX</BIC>
      </BankAccount>
    <BankAccount>
      <IBAN>SK820900000000011424060</IBAN>
      <BIC>GIBASKBX</BIC>
    </BankAccount>
  </BankAccounts>
</Payment>
</Payments>
</Pay>

```

Dátová sekvencia:

```

001 11 20.35EUR20131206 12000971510308012014
UPC: internet - 2014/013SK7911000000002628204091 TATRSKBX
SK820900000000011424060 GIBASKBX SK6402000000002863006757 SUBASKBX 0 0

```

Po úvodnom čísle faktúry = 001 je uvedený počet platobných príkazov = 1 a typ príkazu paymentorder = 1. Pokračuje sa menou, sumou, VS, KS, ŠS. Keďže pole OriginatorsReferenceInformation sa v XML nenachádza, je nahradené nulovou dĺžkou, čo sa prejaví ako dva po sebe idúce tabulátory. Po poznámke je uvedený počet bankových účtov = 3 a IBAN a BIC pre všetky 3 účty. Dve nuly na konci značia, že rozšírenie o trvalý príkaz - StandingOrderExt ani rozšírenie o nastavenie inkasa - DirectDebitExt sa v XML nenachádzajú.

4.4 CRC32

Na začiatok pred vytvorenú sekvenciu sa doplní vypočítaný cyklický redundantný súčet CRC-32. Po dekodovaní je uložený súčet porovnaný s vypočítaným súčtom z dekodovaných dát na zaistenie integrity dát. Napriek tomu, že výsledný QR kód má svoj vlastný opravný mechanizmus tvorcovia štandardu použili CRC z dôvodu, že platobné údaje môžu byť ukladané alebo posielané vo forme by square sekvencie pričom môže dôjsť k porušeniu dát.

Princíp CRC je založený na delení v konečnom telese $GF(2^n)$. Výsledok je určený vstupnou postupnosťou bitov $M(x)$ a zvoleným polynómom $G(x)$ - kľúčom, ktorý je tiež postupnosťou bitov. Keď obe postupnosti interpretujeme ako polynómy nad telesom $GF(2^n)$ CRC je potom určené ako zvyšok po delení $R(x)$ vstupnej postupnosti zvoleným polynómom.

$$R(x) = \frac{M(x)}{G(x)} \quad (4.1)$$

Výsledok je polynóm, ktorý môže byť interpretovaný ako bitová postupnosť. Pri vhodne zvolenom polynóme má aj malá zmena vo vstupnej postupnosti za následok podstatne odlišný výsledok. Pri náhodnej zmene vstupnej postupnosti, pravdepodobnosť odhalenia chyby rastie so šírkou kľúča c a vieme ju definovať vzorcom:

$$P = 1 - 1/2^c \quad (4.2)$$

Pre kľúč dĺžky 32 bitov je pravdepodobnosť odhalenia chyby:

$$P = 1 - \frac{1}{2^{32}} \approx 0,99999999977 \quad (4.3)$$

Pre výpočet CRC32 je podľa štandardu by square použitý polynóm:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \quad (4.4)$$

Poradie bitov pri doplnení CRC pred dátovú sekvenciu si budeme demonštrovať na príklade:

CRC32: 0xD5 4E 9DCD



Výsledok: 0xCD 9D 4ED5 [dátová sekvencia]

Sekvencia s doplneným CRC sa v ďalšom kroku skomprimuje metódou LZMA.

4.5 LZMA kompresia

Komprimačná metóda LZMA je hlavná základná metóda použitá v kompresnom programe 7z. LZMA je slovníková metóda, ktorá bola navrhnutá tak, aby poskytovala vysokú kompresiu, rýchlu dekompresiu a nízke nároky na pamäť.

Princíp LZMA je založený na metóde LZ77 a intervalovom kódovaní (range encoding). LZ77 vyhľadáva vo vyhľadávacom bufferi najdlhšiu zhodu s predvídacím bufferom a do komprimovaného toku zapisuje trojicu (vzdialenosť, dĺžka, nasledujúci symbol). LZMA však v prípade, že nenájde zhodu vyšle symbol (hodnota 0 -255) a v prípade zhody vyšle dvojicu (dĺžka, vzdialenosť), ktorá sa zakóduje intervalovým kódom.

Detailný popis algoritmu LZMA nebol zverejnený, je však voľne k dispozícii SDK so zdrojovým kódom LZMA v jazykoch C, C++, C# a Java. Tvorcovia štandardu odporúčajú pre kompresiu použiť práve tieto zdrojové súbory a v štandarde uvádzajú požadované nastavenie parametrov:

- LC = 3
- LP = 0
- PB = 2
- DICTIONARY_SIZE = 128 KB (2¹⁷)

4.6 By square informačná hlavička

Po kompresii dát sa v tomto kroku pred skomprimované dáta doplní by square hlavička. Hlavička má 16 bitov a pozostáva zo štyroch štvorbitových hodnôt. Jej štruktúru popisuje Tabuľka 4.5.

Attribute	Number of bits	Possible Values	Note
BySquareType	4	0 - 15	by square type
Version	4	0 - 15	version of the by square type
DocumentType	4	0 - 15	document type within given by square type
Reserved	4	0 - 15	bits reserved for future needs

Tabuľka 4.5 by square hlavička

Prvá hodnota reprezentuje by square typ, čo je buď PAY by square alebo INVOICE by square. Verzia slúži na odlišenie jednotlivých verzií PAY by square a INVOICE by square a zaisťuje správnu interpretáciu by square kódu a spätnú kompatibilitu. Každá aplikácia pracujúca s by square štandardom by mala mať jednoznačne určenú najvyššiu podporovanú verziu. By square typ môže obsahovať niekoľko typov dokumentov. PAY by square však obsahuje iba jeden dokument. Posledné pole obsahuje 4 rezervované bity pre prípadné budúce použitie. Prehľad hodnôt jednotlivých atribútov hlavičky použitých v tejto práci je uvedený v Tabuľka 4.6.

by square type	by square type bit value	version	version bit value	document type	document type bit value
PAY by square	0000	0	0000	pay	0000

Tabuľka 4.6 Hodnoty by square hlavičky

Hlavička sa k dátam pridáva až po LZMA kompresii z dôvodu, aby bez potreby dekompresie bolo možné hneď určiť o aký by square typ, verziu a dokument ide.

4.7 Konverzia Base32hex

Posledným krokom pred samotným zakódovaním dát do QR kódu je alfanumerická konverzia Base32hex. Bitová sekvencia sa konverziou transformuje na ASCII znakovú postupnosť. Prekódovaná sekvencia je cca. o 10% väčšia, no spadá do znakovkej sady QR Code 2005 v alfanumerickom kódovacom móde, čo urýchľuje a uľahčuje zakódovanie do QR kódu. Konverzia Base32hex vyžaduje bitovú sekvenciu s dĺžkou deliteľnou 5. V prípade, že táto podmienka nie je

splnená, sú na koniec postupnosti doplnené zarovnávacie bity na najbližšiu dĺžku deliteľnú 5. Bitová sekvencia je následne rozdelená do 5 bitových blokov, ktoré sú prekódované na alfanumerické znaky podľa Tabuľka 4.7.

Character	4 (MSB)	3	2	1	0 (LSB)
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
4	0	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	0	0	0	0
9	0	0	0	0	1
A	0	0	0	1	0
B	0	0	0	1	1
C	0	0	1	0	0
D	0	0	1	0	1
E	0	0	1	1	0
F	0	0	1	1	1
G	1	1	0	0	0
H	1	1	0	0	1
I	1	1	0	1	0
J	1	1	0	1	1
K	1	1	1	0	0
L	1	1	1	0	1
M	1	1	1	1	0
N	1	1	1	1	1
O	1	1	0	0	0
P	1	1	0	0	1
Q	1	1	0	1	0
R	1	1	0	1	1
S	1	1	1	0	0
T	1	1	1	0	1
U	1	1	1	1	0
V	1	1	1	1	1

Tabuľka 4.7 Kódovacia tabuľka pre Base32hex

Výsledok konverzie sa nazýva by square sekvencia, ktorá môže byť braná ako výsledok kódovania a môže byť obsahom emailových príloh. Podľa štandardu sa odporúča súbor s by square sekvenciou ukladať so súborovou príponou .bsqr pre lepšiu identifikáciu obsahu súboru. Najčastejšie sa však by square sekvencia kóduje do QR kódu.

4.8 Vytvorenie QR kódu

Pre grafickú reprezentáciu zakódovaných dát by square štandard používa QR kód 2005. Popis QR kódu a proces zakódovania dát bol popísaný v predošlej kapitole. By square používa parametre kódovania uvedené v Tabuľka 4.8.

Parameter	Value	Note
code version	xL	x is number (from 1 to 17)
mode indicator	Alphanumeric	0–9, A–Z (uppercase only), space, \$, %, *, +, /, :

Tabuľka 4.8 Parametre kódovania do QR kódu 2005 použité v by square

Používa sa teda alfanumerický kódovací mód, QR kód s verziou 1 - 17 (21x21 - 85x85) s najnižšou úrovňou opravy chyby L, ktorá dokáže opraviť 7% kódového slova.

Štandard je navrhnutý tak, aby bolo možné definovať fyzickú veľkosť QR kódu bez závislosti na tom, ktorá verzia QR kódu je pre zakódovanie dát použitá. Fixná fyzická veľkosť v mnohých okolnostiach uľahčuje umiestňovanie týchto kódov na faktúry. Minimálna fyzická veľkosť QR kódu je definovaná na rozmer 30mm x 30mm, no odporúčaná veľkosť je 36mm x 36mm.

5 Návrh riešenia modulu pre systém SAP

Po zoznámení sa s teóriou týkajúcou sa systému SAP, vytvárania QR kódov a štandardu PAY by square sa dostávame ku ďalšej časti práce, ktorou je vytvorenie riešenia pre systém SAP, ktoré umožní generovanie faktúr s QR kódmi obsahujúcich platobné údaje z faktúry. V tejto kapitole si popíšeme štyri základné kroky, potrebné k vytvoreniu riešenia. Prvým krokom je tzv. customizing, čo je nastavenie štandardných funkcií systému SAP, druhým krokom je vytvorenie funkčného modulu, ktorý umožní vytvorenie by square sekvencie z dát faktúry, tretím krokom je zakódovanie sekvencie do QR kódu. Posledným krokom je rozšírenie formulára faktúry a tlačového programu, tak, aby bola umožnená tlač faktúr s platobnými údajmi uloženými v QR kóde.

5.1 Nastavenia systému

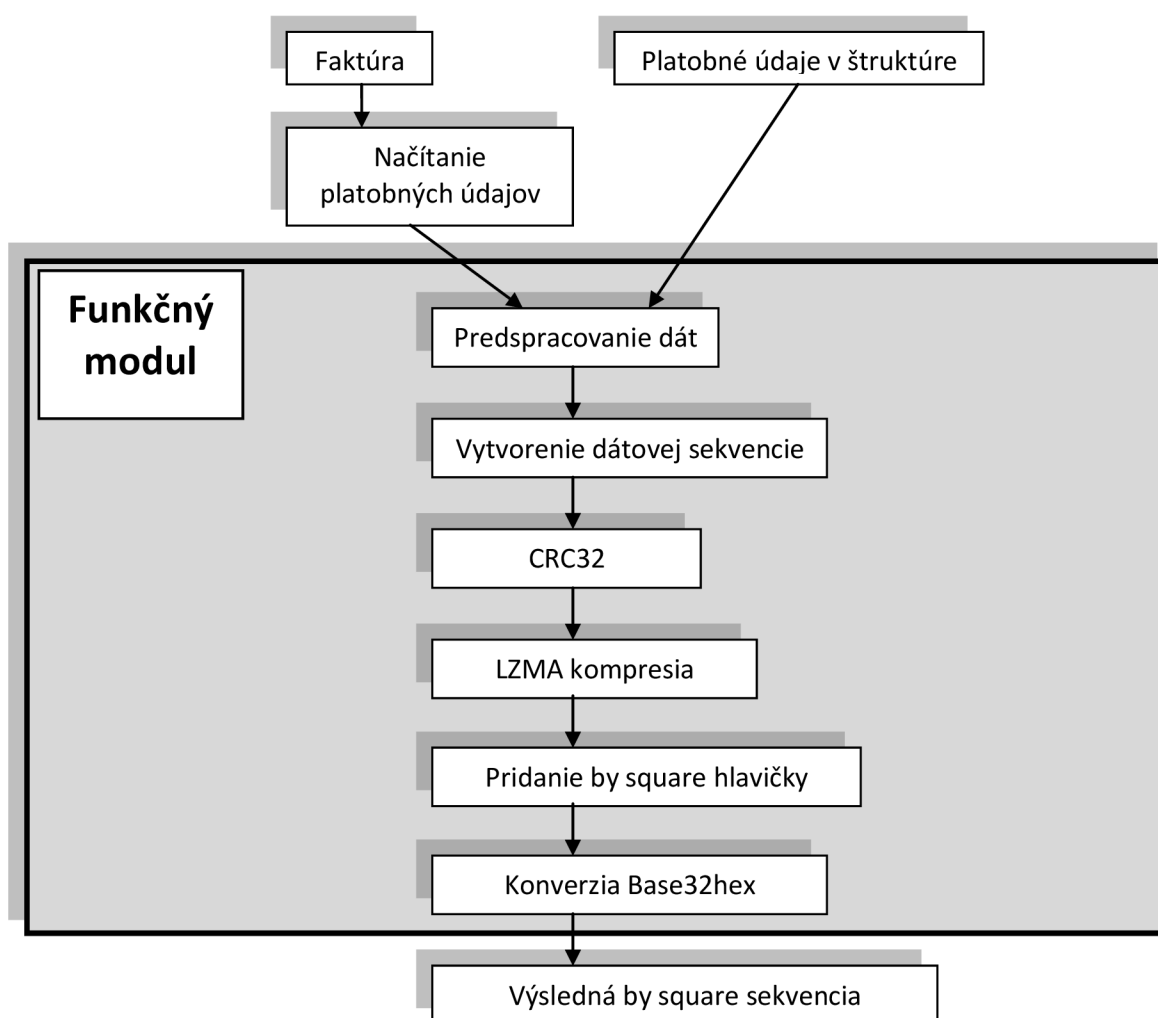
Pre správnu funkčnosť vytváraného modulu je potrebné, aby bola v systéme zabezpečená dostupnosť všetkých potrebných údajov. Základné fakturačné údaje ako číslo faktúry, suma, mena, variabilný symbol máme na základe štandardnej funkčnosti odberateľských faktúr v SAPE k dispozícii. Čo však štandardne nie je k dispozícii je bankové spojenie na subjekt, ktorý faktúru vystavuje. Toto je vyriešené založením tohto subjektu do systému v úlohe dodávateľa. Na karte dodávateľa je následne možné udržiavať bankové účty. Bankové účty môžu byť udržiavané buď vo formáte číslo bankového účtu / kód banky alebo vo formáte IBAN. Pre potreby štandardu by square je potrebný IBAN, teda ak v systéme chýba je potrebné ho pre daný účet vygenerovať. So zakladaním bankových účtov súvisí aj nutnosť mať v systéme založenú požadovanú banku, odkiaľ sa bude používať bankový identifikačný kód BIC. Všetky vyššie popísané kroky je nutné vykonať v rámci implementácie.

5.2 Návrh funkčného modulu pre vytvorenie by square sekvencie

Základnou časťou riešenia bude funkčný modul pre generovanie by square sekvencie. Nakoľko platobné údaje na faktúre obsahujú iba údaj pre jednoduchý platobný príkaz a PAY by square umožňuje kódovať aj trvalý príkaz a inkaso bude funkčný modul možné použiť dvoma spôsobmi. V prvom prípade bude funkčný modul vytvárať by square sekvenciu iba z dát na faktúre, to znamená, že jeho volanie bude zapracované do formulára faktúry, kde bude naplnený platobnými dátami. Druhý spôsob bude umožňovať funkčný modul využiť na kódovanie údajov definovaných vo vytvorenej štruktúre. Táto štruktúra bude zodpovedať formátu vstupného XML súboru popísaného pri štandarde. Tento režim bude okrem jednorazovej platby podporovať aj kódovanie

trvalého príkazu. Kódovanie inkasa nebude súčasťou vytvoreného riešenia, nakoľko kódovanie inkasa sa zatiaľ v reálnej prevádzke nevyužíva. Oba spôsoby využitia funkčného modulu môžeme popísať schémou uvedenou na Obrázok 5.1.

Prvým krokom v režime čítania dát z faktúry je načítanie platobných údajov do rovnamej štruktúry, ako sa používa pre vstup v druhom režime. Druhým krokom je predspracovanie dát, kde sú sformátované dátumy, sumy a poznámka pre prijímateľa. Predspracované dáta sa použijú pre vytvorenie dátovej sekvencie, na základe pravidiel štandardu. Pre dátovú sekvenciu je vypočítané CRC32, ktoré je pridané na začiatok sekvencie. Sekvencia rozšírená o CRC32 je následne skomprimovaná kompresnou metódou LZMA. Ku výsledku kompresie je doplnená by square hlavička a dáta sú prekonvertované konverziou Base32hex, čím vznikne by square sekvencia, ktorá je výstupom funkčného modulu.



Obrázok 5.1 Proces vytvorenia by square sekvencie

5.3 Vytvorenie QR kódu

Sekvenciu vytvorenú funkčným modulom je potrebné zakódovať do QR kódu. SAP štandardne nepodporuje generovanie QR kódov, preto v práci navrhujeme viaceré alternatívy, ktoré budú následne implementované a otestované.

Prvým riešením je dodatočná inštalácia typu tlačiarne, ktorá QR kódy umožňuje generovať. Pre vygenerovanie QR kódu na faktúru bude v SmartForme potrebné by square sekvenciu ohraničiť riadiacimi znakmi tlačiarne pre tlač QR kódu. V druhom riešení bude prvá alternatíva rozšírená o využitie PDF tlačiarne, ktorá vytvára QR kódy pri tlači formulára do PDF. Pri treťom riešení využijeme možnosť komunikácie systému SAP s externou aplikáciou. Táto externá aplikácia po predaní dát zo SAPu vytvorí obrázok QR kódu, ktorý následne uloží do určeného pracovného adresára. Riadiaci program v SAPE následne obrázok uloží do správy grafík, čím sa stane dostupný pre spracovanie na formulári. Poslednou alternatívou je riešenie, kedy externá aplikácia vráti iba maticu hodnôt, ktoré budú reprezentovať biele a čierne body QR kódu. QR kód bude následne vytvorený z tejto matice priamo na formulári. Všetky z popísaných alternatív budú v nasledujúcich častiach implementované a otestované.

5.4 Návrh úpravy formulára a tlačového programu

V poslednej časti návrhu sa zameriame na návrh úpravy formulára faktúry a tlačového programu. Formulár a tlačový program bude upravený tak, aby bolo možné využiť funkčný modul spolu s kódovaním do QR kódu pre zakódovanie platobných dát faktúry do QR kódu. Pre potreby tejto práce som na úpravu zvolil faktúru za teplo, ktorá je súčasťou SAPom certifikovaného riešenia pre odbyt tepla, ktoré vytvorila firma RKMC, s.r.o.. Faktúra má prehľadnú štruktúru aj rozhranie. Dôležitým krokom v návrhu je voľba umiestnenia QR kódu tak, aby nebol nutný rozsiahly zásah do rozloženia existujúceho formulára. Nakoľko dáta môžu byť v rámci formulára vyhodnocované dynamicky, je potrebné zvoliť správne miesto aj s ohľadom na chod programu, ktorý vykresľuje formulár. Aktuálne rozloženie formulára spolu s najvýhodnejším umiestnením QR kódu zobrazuje Obrázok 5.2. Okrem rozloženia formulára je potrebné upraviť aj jeho rozhranie a tlačový program tak, aby boli na formulári k dispozícii všetky údaje potrebné pre kódovanie platobných údajov podľa štandardu by square.

Teko Faktúra za Máj 2012
Doklad číslo: 4115530512

Číslo odberateľa: 60182 Číslo zmluvy: 43001094

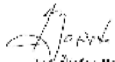
DODÁVATEĽ		ODBERATEĽ	
Tepláreň Košice, a.s.		Stredná odborná škola	
Teplárenská 3 042 92 Košice		Gemerská 1 040 11 Košice	
IČO: 36211541 DIČ: 2020048580 IČ DPH: SK2020048580 Banka: Československá obchodná banka, a.s. Účet: 256420997590		IČO: 17078423 DIČ: 17078423 IČ DPH: Banka: Slovenská pošta Účet: 70001204663160	
Konštantný symbol	0008	Dátum vyhotovenia	08.06.2012
Variabilný symbol	4115530512	Dátum odoslania	11.06.2012
Forma úhrady	Prevodný príkaz	Dátum splatnosti	25.06.2012
		Dátum dodania tovaru	31.05.2012

Fakturuje Vám dodávku tepelnej energie za mesiac Máj 2012.
Opakované dodanie tovaru podľa § 19 ods.3 zákona o DPH.

Fakturované veličiny	Množstvo	Jednotka	Jednotková cena	Cena bez DPH	DPH
Regulačný príkon (FIXNÁ ZLOŽKA)	13,8025	kW	99,92540	1,379,22 EUR	20 %
Teplo (VARIABILNÁ ZLOŽKA) Ostatní	13,057,30	kWh	0,03590	468,75 EUR	20 %
Množstvo doplnkovej vody	0,64	m3	1,53191	0,98 EUR	20 %
Cena spolu bez DPH				1.848,95 EUR	

Vyučtovanie	
Spolu bez DPH	1.848,95 EUR
Základ dane	1.848,95 EUR
DPH 20%	369,79 EUR
Spolu s DPH	2.218,74 EUR
Záloha	2.190,00 EUR
Celkom s DPH	2.218,74 EUR
Čast. vyrovnanie	0,00 EUR
K úhrade	28,74 EUR

QR kód


 Ing. Stefan Hačala
 vedúci odd. administratívny predaja tepla

Súčasťou faktúry je príloha č. 1
 KP 40.30.10 - Tepelná energia
 Tepláreň Košice, a.s. je zapísaná v Obchodnom registri SR pri OS Košice I v oddieli Sa, vložka č. 1204/V

Vystavil: Odd. 4014 - Administratíva predaja tepla
 Telefón: 055/6192013

Obrázok 5.2 Rozloženie faktúry s označením umiestnenia pre QR kód

6 Implementácia

Obsahom tejto kapitoly bude implementácia riešenia založená na uvedenom návrhu. Rovnako ako návrh aj implementácia je rozdelená na 4 časti a to: nastavenie systému, funkčný modul pre generovanie by square sekvencie, generovanie QR kódu a úprava formulára a tlačového programu. Všetky nové objekty boli zahrnuté do nového balíka ZQR a sú uložené v jednej samostatnej transportnej požiadavke. Takéto uloženie uľahčuje možný budúci prenos na iné ERP systémy.

6.1 Nastavenia systému

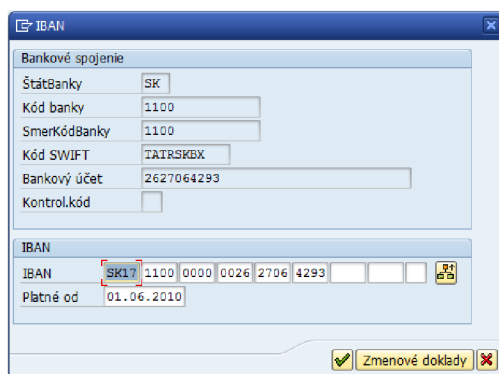
Pre implementáciu riešenia bol použitý interný firemný ERP systém. Na tomto systéme je dostupné aj riešenie pre odbyt tepla - ODT, konkrétne pre dodávateľa Tepláreň Košice, a.s.. Pre správnu funkčnosť musí byť dodávateľ založený v systéme. V našom prípade je založený pod číslom 1000039, čo bolo overené transakciou XK03 - Zobrazenie dodávateľa. V prípade, že dodávateľ nie je založený je možné založenie vykonať transakciou XK01. Pre daného dodávateľa bolo potrebné založiť bankové spojenia v transakcii XK02 - Zmena dodávateľa voľbou zobrazenia Platobný styk. Obrázok 6.1 zobrazuje obrazovku pre zadávanie bankových spojení.



Obrázok 6.1 Založenie bankových účtov dodávateľa.

Pre správne založenie bankového spojenia a vygenerovania IBAN je potrebné mať v systéme založenú banku cez transakciu FI01 a pre potreby štandardu by square je nutné v kmeňových údajoch banky vyplniť pole SWIFT kód.

Po založení účtov bol pre každý účet vygenerovaný IBAN stlačením tlačidla so šípku v stĺpci IBAN. Vygenerovaný IBAN sa užívateľovi zobrazí na potvrdenie (Obrázok 6.2).



Obrázok 6.2 Založenie IBAN

6.2 Funkčný modul pre generovanie by square sekvencie

Pre generovanie by square sekvencie bol vytvorený funkčný modul s názvom ZQR_CREATE_PAY. Funkčný modul ma dva vstupné (importné) parametre:

- gv_vbeln TYPE vbeln - číslo faktúry
- gs_pay TYPE zqr_pay- štruktúra reprezentujúca XML s údajmi o platbách

Výstupným (exportným) parametrom je o_bysquare TYPE zqr_bysquare reprezentujúci výslednú by square sekvenciu.

6.2.1 Úpravy v ABAP Dictionary

Pre potreby funkčného modulu bolo potrebné v dictionary založiť nové domény, dátové prvky, štruktúry a tabuľkové typy. Nová doména, ktorá bola založená je ZQR_PAYMENT_OPTIONS a reprezentuje dátový typ pre typ platby. Tabuľka 6.1 uvádza hodnoty, ktoré môže nadobúdať. Implementovaná je podpora pre jednorazový príkaz, trvalý príkaz alebo ich kombináciu.

Hodnota	Význam
1	jednorazová platba
2	trvalý príkaz
3	jednorazová platba + trvalý príkaz

Tabuľka 6.1 Hodnoty pre doménu zqr_paymen_options

Pre periodicitu opakovania trvalého príkazu bola založená doména ZQR_PERIODICITY. Hodnoty, ktoré môže doména nadobúdať a ich význam definuje Tabuľka 6.2.

Hodnota	Periodicita
d	Denne
w	Týždenne
b	Dvojtýždenne
m	Mesačne
B	Dvojmesačne
q	Štvrťročne
s	Polročne
a	Ročne

Tabuľka 6.2 Hodnoty domény zqr_periodicity

Nové dátové prvky sú založené na XSD schéme. Ich zoznam uvádza Tabuľka 6.3.

Dátový prvok	Doména
ZQR_AMOUNT	Suma
ZQR_BY SQUARE	By square sekvencia
ZQR_CONSTSYM	Konštantný symbol
ZQR_CURRCODE	Kód meny
ZQR_INVOICEID	ID faktúry
ZQR_ORGREFINF	Referencia platiteľa platby
ZQR_PAYMENT_OPTIONS	Typ platby
ZQR_PAYMENTNOTE	Poznámka ku platbe
ZQR_SPECSYM	Špecifický symbol
ZQR_VARSYM	Variabilný symbol
ZQR_STANDINGORDEREXT	Príznak rozšírenia o trvalý príkaz
ZQR_DAY	Poradové číslo dňa v rámci týždňa (1-7) alebo v rámci mesiaca (1-31)
ZQR_MONTH	Mesiac, v ktorom je trvalý príkaz realizovaný
ZQR_PERIODICITY	Periódna opakovania trvalého príkazu
ZQR_LASTDATE	Dátum ukončenia trvalého príkazu

Tabuľka 6.3 Nové dátové prvky

Pre uloženie bankového účtu bola vytvorená štruktúra ZQR_BANK_ACCOUNT s položkami IBAN a BIC. Pre ukladanie viacerých platieb slúži tabuľkový typ ZQR_BANK_ACCOUNTS. Celý jeden platobný príkaz je reprezentovaný štruktúrou ZQR_PAYMENT, ktorá obsahuje jednotlivé položky platobného príkazu podľa štandardu. Pre uloženie viacerých príkazov je vytvorený tabuľkový typ ZQR_PAYMENTS. Tento je následne použitý v štruktúre ZQR_PAY, ktorá zodpovedá XML formátu, ktorý je definovaný ako vstup pre PAY by square.

6.2.2 Vytvorenie by square sekvencie

Prvým krokom vykonávaným vo funkčnom module je vytvorenie dátovej sekvencie z dát predaných prostredníctvom importného parametra `gs_pay`. Dáta sú upravené na požadovaný tvar a sú spájané do reťazca dátovej sekvencie. Ako oddeľovač je použitý znak tabulátora definovaný ako konštanta `horizontal_tab` v triede `cl_abap_char_utilities`.

Vytvorená dátová sekvencia je s využitím štandardného funkčného modulu `SCMS_STRING_TO_XSTRING` prevedená zo znakového reťazca na reťazec hexa znakov. Tento reťazec v hexa je použitý ako parameter metódy `cl_abap_zip=>crc32` pre výpočet CRC32. Výsledok z metódy je následne upravený tak, aby bolo dodržané poradie bytov uvedené v 4.4. CRC je doplnené pred dátovú sekvenciu, ktorá je uložená s využitím štandardného funkčného modulu `GUI_DOWNLOAD` do lokálneho súboru. Tento súbor slúži na prenos dát medzi ABAPom a aplikáciou pre LZMA kompresiu skompilovanou ako .JAR archív.

Na základe odporúčanií tvorcov štandardu som pre kompresiu použil LZMA SDK s GNU LGPL licenciou dostupné na [9]. SDK zahŕňa knižnice pre programovacie jazyky C, C++, C# a Java. Ako najvhodnejšiu z týchto alternatív som zvolil Javu, nakoľko aj určité časti SAPu sú vyvinuté práve v tomto programovacom jazyku.

Pre kompresiu som vytvoril program, ktorý číta vstupnú sekvenciu pre kódovanie zo súboru, ktorého názov je definovaný ako `data_[cislo_faktury].crc`, kde číslo faktúry je parametrom programu. Načítaná sekvencia je prevedená na bytové pole metódou `public static byte[] hexStringToByteArray(String s)`. Pole je skomprimované metódou `public byte[] compress(byte[] data)` triedy `LZMACompressor()` z knižnice `LzmaAlone` s defaultným nastavením parametrov. Skomprimovaný výstup je uložený do výstupného súboru `data_[cislo_faktury].lzma`.

Program pre kompresiu je po uložení vstupných dát spúšťaný funkčným modulom `GUI_RUN` s číslom faktúry zadaným ako parameter. Nakoľko tento modul nečaká na ukončenie spúšťaného programu, je potrebné zabezpečiť, aby proces na výpočet by square sekvencie počkal na výstup programu kompresie. Toto je zabezpečené programovou slučkou testujúcou existenciu súboru po dobu 7 sekúnd. Skomprimovaný výstup je zo súboru načítaný metódou `gui_upload` triedy `cl_gui_frontend_services` a dočasné súbory sú zmazané volaním metódy `file_delete`. Dáta sú prevedené do binárnej podoby a doplnené o 16 bitovú by square hlavičku. Následne je na bloky dĺžky 5 bitov použitá konverzia `Base32hex` implementovaná v podprograme:

```
FORM base32hex
    USING lv_five_bits
    CHANGING lv_result TYPE c.
```

Pretože funkčný modul nesmie obsahovať podprogramy je podprogram umiestnený v programe `ZQR_CREATE_PAY` a je použité vzdialené volanie:

```
PERFORM base32hex(zqr_create_pay)
  USING lv_five_bits
  CHANGING lv_char.
```

Výsledok tejto konverzie tvorí by square sekvenciu a je uložený do exportného parametra funkčného modulu `o_bysquare`.

6.3 Generovanie QR kódu

By square sekvencia tvorí dáta ukladané do QR kódu. V rámci práce som implementoval viaceré možnosti vytvárania QR kódu, aby bolo možné porovnať výhody a nevýhody jednotlivých alternatív. Uvedieme si popis implementácie jednotlivých variant.

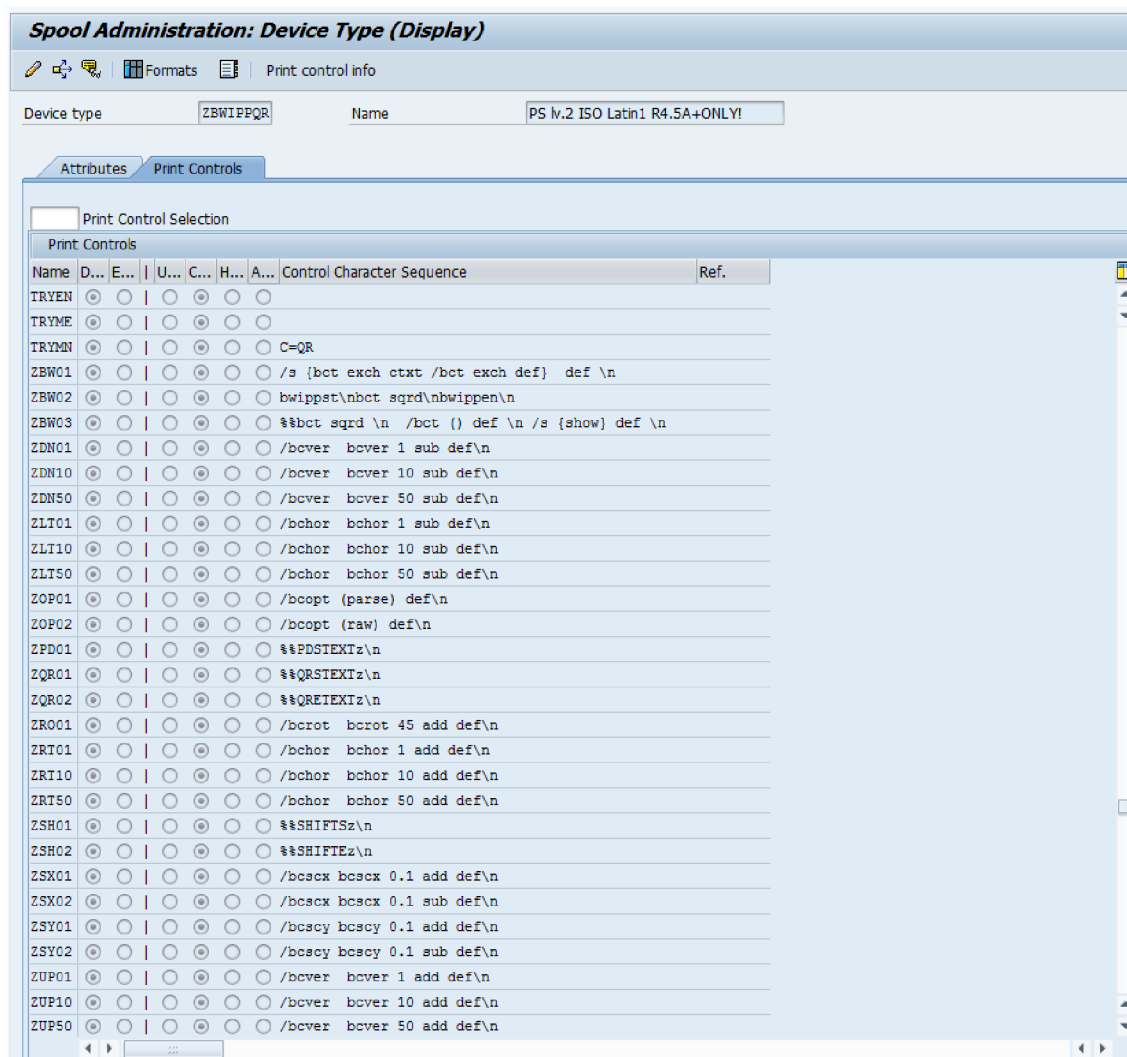
6.3.1 Využitie špeciálneho výstupného zariadenia

V tomto prípade bol pre vytvorenie QR kódu použitý open source softvér pre generovanie čiarových kódov Barcode Writer in Pure Postscript (BWIPP) použitý ako typ výstupného zariadenia a fyzická tlačiareň ako výstupné zariadenie.

V transakcii SPAD - Správa spoolu bol voľbou Utilities => For device types => Import naimportovaný typ zariadenia pre tvorbu čiarových kódov stiahnutý z [12] s názvom ZBWIPPQR. Tento typ zariadenia má v definícii špeciálne riadiace príkazy pre tlačiareň, ktoré umožňujú tvorbu QR kódu. Prehľad riadiacich znakov použitých pre tlač čiarových kódov vrátane QR kódu zobrazuje Obrázok 6.3.

Okrem príkazov tu musia byť definované jednotlivé akcie pre tlačiareň. Akcie sú naviazané na príslušný Postscript formát výstupu. V našom prípade je to formát DINA4, ktorý zodpovedá štandardnej strane A4. Akcie definované pre potreby QR kódov sú:

- ZBWIPP: CREDITS - základná definícia
- ZENC1: BWIPP encode qr 1 - príkazy pre kódovanie 1
- ZENC2: BWIPP encode qr 2 - príkazy pre kódovanie 2
- ZENC3: BWIPP encode qr 3 - príkazy pre kódovanie 3
- ZENC4: BWIPP encode qr 4 - príkazy pre kódovanie 4
- ZREND3: BWIPP rend renmatrix - vykreslenie QR kódu



Obrázok 6.3 Riadiace znaky tlačiarne pre typ zariadenia ZBIWPPQR

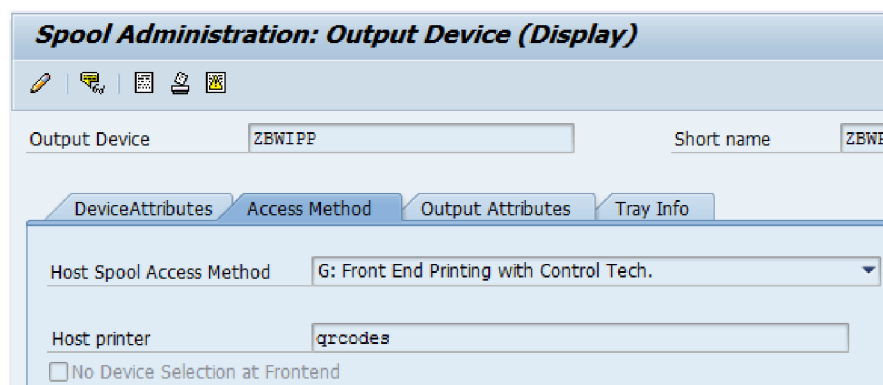
Zdrojové kódy týchto akcií sú obsiahnuté v súbore [12]. Na aplikovanie akcií v akomkoľvek výstupe bolo nutné pridať do akcie Printer initialization nasledovné referencie:

```
\i<ZBWIPPQR:DINA4:ZBWIPP>\n
\i<ZBWIPPQR:DINA4:ZREND3>\n
\i<ZBWIPPQR:DINA4:ZENC1>\n
\i<ZBWIPPQR:DINA4:ZENC2>\n
\i<ZBWIPPQR:DINA4:ZENC3>\n
\i<ZBWIPPQR:DINA4:ZENC4>
```

Tento novodefinovaný typ zariadenia je následne využitý pre výstupné zariadenie. Zariadenie bolo importované cez Utilities => For output devices => Import devices a má názov ZBIWPP. Na to, aby sa QR vytlačil musí byť pre tlač použité práve toto zariadenie. Pre definovanie QR kódu na formulári Smartforms je nutné použiť riadiace znaky tlačiarne ZBW01 a ZBW02, ktorými sa ohraničí text, ktorý bude obsiahnutý v QR kóde.

6.3.2 Využitie PDF tlačiarne

Fyzická tlačiareň použitá na vygenerovanie QR kódu je v tomto prípade nahradená virtuálnou PDF tlačiarňou Cute PDF writer. Inštalátor pre tlačiareň je dostupný na [1].Tlačiareň je potrebné nainštalovať s voľbou PS2PDF a po nainštalovaní je premenovaná z CutePDF Writer na qrcodes, aby bolo možné použiť importované nastavenie zariadenia ZBIWPP, ktoré je zobrazené na Obrázok 6.4.



Obrázok 6.4 Nastavenie výstupného zariadenia ZBIWPP

Pre okamžitú tlač do PDF je možné pre jednotlivých užívateľov cez transakciu SU01 - Údržba užívateľa nastaviť prednastavené výstupné zariadenie ZBIWPP a zaškrtnúť príznak Okamžitý výstup. QR kód je v tomto prípade vygenerovaný pri vytvorení výstupného PDF súboru.

6.3.3 Vytvorenie QR kódu samostatnou aplikáciou

Tretia implementovaná alternatíva využíva na generovanie QR kódu samostatnú Java aplikáciu QR_CODE.JAR. Aplikácia využíva pre generovanie QR kódu open source knižnicu ZXing podporujúcu spracovanie 1D a 2D čiarových kódov. Vstup tvoria dva parametre, a to by square sekvencia a číslo faktúry. Veľkosť vygenerovaného kódu je nastavená na 180 x 180 pixelov a úroveň opravy chyby je nastavená na úroveň L. Vytvorenie matice QR kódu je vykonané nasledovnými príkazmi:


```

//Nastavenie velkosti QR kodu
int size = 180;
Hashtable hintMap = new Hashtable();

//Nastavenie urovne opravy chyby
hintMap.put(EncodeHintType.ERROR_CORRECTION,
ErrorCorrectionLevel.L);

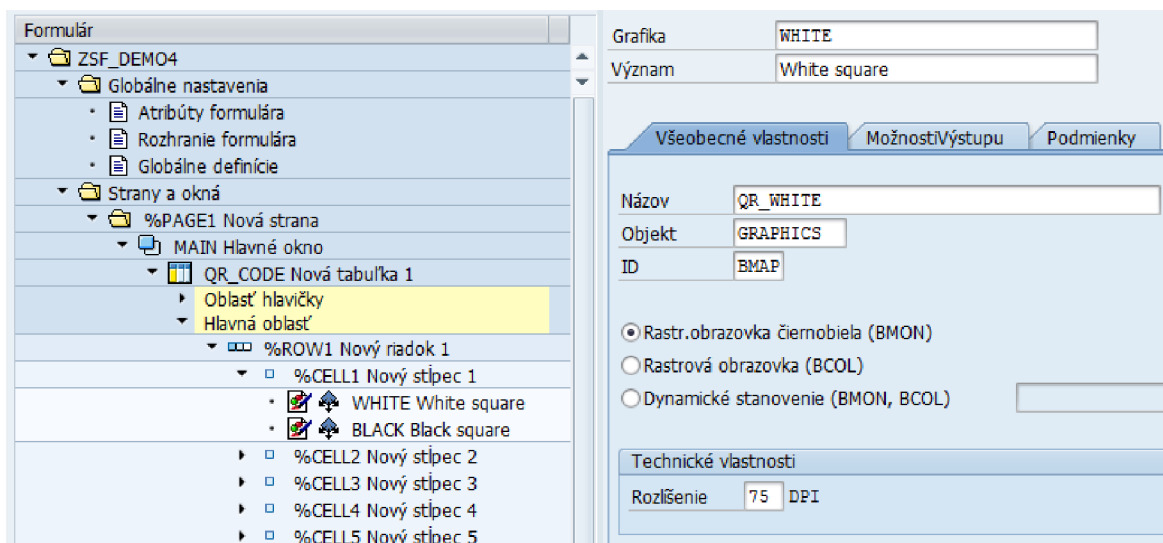
//Vytvorenie matice QR kodu
QRCodeWriter qrCodeWriter = new QRCodeWriter();
BitMatrix byteMatrix = qrCodeWriter.encode(
    qrCodeText, BarcodeFormat.QR_CODE, size, size, hintMap);

```

Matica je následne uložená ako BMP obrázok s názvom QR_[cislo_faktury].bmp. Táto aplikácia skompilovaná ako Java archív je z ABAPu volaná funkčným modulom GUI_RUN s príslušnými parametrami a vytvorený bitmapový obrázok je uložený medzi grafiky do systému SAP. Na jeho uloženie sa volá podprogram import_bitmap využívajúci volanie podprogramu import_bitmap_bds, ktorý je súčasťou SAP štandardu pre správu grafik. Pre uloženie obrázku je použitá čiernobiela reprezentácia. Logika na vytvorenie obrázku QR kódu a import obrázku do systému SAP bola v tejto časti doplnená do funkčného modulu ZQR_CREATE_PAY. Obrázok QR kódu je teda po volaní funkčného modulu možné nájsť v správe grafik cez transakciu SE78.

6.3.4 Vytvorenie QR kódu vo formulári z matice

V poslednej implementovanej alternatíve je na vygenerovanie matice QR kódu rovnako ako v predošlom prípade použitá Java aplikácia. Matica však nie je použitá na vytvorenie obrázku QR kódu, ale je odovzdaná prostredníctvom textového súboru na spracovanie späť do SAPu. Na reprezentáciu hodnôt v matici sú použité dva znaky: medzera - biely bod a znak X - čierny bod. Vytvorená matica je uložená ako interná tabuľka a do Smartform je prenesená rozhraním formuláru. Formulár je rozšírený o tabuľku, ktorá sa použije ako matica pre vykreslenie QR kódu. Každá bunka tabuľky obsahuje rozhodovaciu podmienku (Obrázok 6.5), ktorá vykreslí jednu z dvoch grafik QR_WHITE alebo QR_BLACK, teda biely alebo čierny štvorec. Postupným prechodom po jednotlivých riadkoch sa takto vykreslí celý QR kód.



Obrázok 6.5 Tabuľka pre vykreslenie QR kódu v Smartforme

6.4 Úprava formulára a tlačového programu

V práci som pre demonštráciu funkčnosti použil tlačový program ZODT_016 pre faktúry z riešenia pre odbyt tepla, ktorý využíva Smartform ZODT_002. Tlačový program bolo potrebné upraviť tak, aby vo formulári boli k dispozícii všetky potrebné údaje. Formulár obsahoval údaje o bankovom spojení dodávateľa, no však v tvare bankový účet / kód banky a nie v požadovanom formáte IBAN. V tlačovom programe bola doplnená logika, na načítanie chýbajúcich údajov o banke. Konkrétne o kľúč štátu banky `banks` a kontrolný kód banky `bkont`. O tieto atribúty bola rozšírená aj štruktúra rozhrania formuláru ZODT_S_037 (fakturácia - adresa odberateľa).

V druhom kroku bolo potrebné pripraviť všetky potrebné údaje pre funkčný modul generujúci by square sekvenciu. Vo formulári boli v časti globálna definícia vytvorené štruktúry `GS_PAY`, `GS_PAYMENT` a `GS_BANK_ACC` pre ukladanie platobných údajov. Logika formulára bola následne rozšírená o element Riadky programu, kde bolo zadané volanie štandardného funkčného modulu `READ_IBAN` pre vytvorenie IBAN, načítanie SWIFT kódu banky a naplnenie štruktúr dátami faktúry. Umiestnenie tejto logiky bolo zvolené s ohľadom na sekvenčné spracovanie elementov formulára tak, aby boli dostupné všetky potrebné údaje faktúry vyhodnocované na formulári.

Volaním funkčného modulu `ZQR_CREATE_PAY` je z fakturačných údajov vytvorený obrázok QR kódu a je umiestnený do správy grafík. Pre správne umiestnenie QR kódu na formulár bola rozšírená šablóna `VYUCTOVANIE_F`, ktorá slúži na vykreslenie tabuľky Vyúčtovanie. Do prvého riadka šablóny boli pridané dva stĺpce bez orámovania so šírkou 2,5cm a 6,5cm. Prvý je využitý pre umiestnenie by square loga a druhý pre samotný QR kód.

7 Testovanie

Po implementácii riešenia, ktoré umožňuje generovať faktúry s QR kódmi pre rýchlu úhradu prostredníctvom internetbankingu v mobilnej aplikácii bude riešenie v tejto kapitole podrobené testovaniu. Testovanie bude rozdelené na niekoľko častí v závislosti od toho, čo sa bude testovať. Ako prvý bude otestovaný proces generovania by square sekvencie z platobných údajov naplnených do štruktúry. Druhý test sa bude zaoberať porovnaním rôznych implementácií pre zakódovanie dát do QR kódu. V treťom teste budeme hľadať optimálnu veľkosť QR kódu a následne budeme testovať celý proces kódovania platobných údajov zapracovaný do fakturácie v module pre odbyt tepla. V poslednom teste bude riešenie otestované mobilnými aplikáciami štyroch rôznych bánk. Po zhodnotení výsledkov testov bude záver kapitoly tvoriť popis ďalších možností rozvoja vytvoreného riešenia.

7.1 Testovanie funkčného modulu pre generovanie by square sekvencie

Test funkčného modulu pre generovanie by square sekvencie je zameraný na testovanie jednotlivých krokov pri procese vytvárania tejto sekvencie. Konkrétne ide o vytvorenie dátovej sekvencie z údajov uložených v štruktúre zodpovedajúcej vstupnému XML súboru, výpočet CRC32, LZMA kompresiu a záverečnú Base32hex konverziu. Ako zdroj pre testovanie slúžili najmä súbory, ktoré mi poskytli tvorcovia štandardu. Poskytnuté testy pozostávali z piatich rôznych platobných dát, ktoré pokrývali rôzne kombinácie kódovaných údajov. Pre každé dáta tu bolo dostupných šesť súborov s rôznymi príponami:

- XML - platobné dáta vo formáte XML podľa XSD schémy definovanej štandardom
- TXT - dátová sekvencia vytvorená z platobných dát
- CRC - dátová sekvencia doplnená o CRC32
- LZMA - dátová sekvencia doplnená o CRC32 skomprimovaná algoritmom LZMA
- BIN - výsledok LZMA kompresie doplnený o by square hlavičku
- BSQR - výsledok base32hex konverzie - by square sekvencia

Pozrieme sa bližšie na jednotlivé platobné dáta. Prvé dáta obsahujú iba informáciu o bankovom účte príjemcu, typ platby a menu bez údaju o sume. Ich úlohou bolo otestovať najjednoduchšiu alternatívu, ktorú je možné v štandarde použiť. Druhé dáta obsahujú tri bankové účty, sumu, menu, dátum splatnosti, variabilný, konštantný a špecifický symbol a poznámku pre prijímateľa. Obsiahnuté platobné údaje zodpovedajú najčastejšie používanému formátu vstupných dát. Tretie dáta obsahujú okrem sumy všetky údaje ako druhé a reprezentujú darcovskú platbu, kedy si

užívateľ resp. darca môže po načítaní údajov vyplniť ľubovoľnú sumu. Štvrté dáta obsahujú štyri bankové účty: 2 slovenské, 1 rakúsky a 1 nemecký. Namiesto symbolov je tu použitá referencia platiteľa, ktorá podľa SEPA nahradí VS, KS a ŠS. V posledných piatich dátach sú obsiahnuté až tri platobné príkazy a slúžia na demonštráciu, že podľa štandardu, je možné do jedného QR kódu zakódovať viaceré platby.

Testovacie dáta boli postupne použité pri ladení aplikácie, kedy bol výsledok z funkčného modulu porovnávaný po každom kroku kódovania s testovacími súbormi. Jediná zistená odchýlka voči referenčným hodnotám nastala pri štvrtom teste. Rozdiel nastal hneď v prvom kroku po vytvorení dátovej sekvencie. Táto odchýlka bola však spôsobená tým, že poradie bankových účtov vo vzorovej dátovej sekvencii nezodpovedalo poradiu účtov uvedenému vo vzorovom XML súbore. Iné poradie účtov však nemá na funkčnosť a správnosť kódovania žiaden vplyv. Aby však zodpovedali výsledky jednotlivých krokov upravil som poradie účtov v testovacej aplikácii podľa poradia vo vzorovej dátovej sekvencii. Po tejto úprave všetky testy potvrdili správnosť všetkých krokov algoritmu. Výsledky testov zobrazuje Tabuľka 7.1.

Test	Dátová sekvencia	CRC32	LZMA	LZMA s by square hlavičkou v bin	Base32hex - by square sekvencia
1	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓
4	✗/✓	✗/✓	✗/✓	✗/✓	✗/✓
5	✓	✓	✓	✓	✓

Tabuľka 7.1 Výsledky testov funkčného modulu pre vytvorenie by square sekvencie

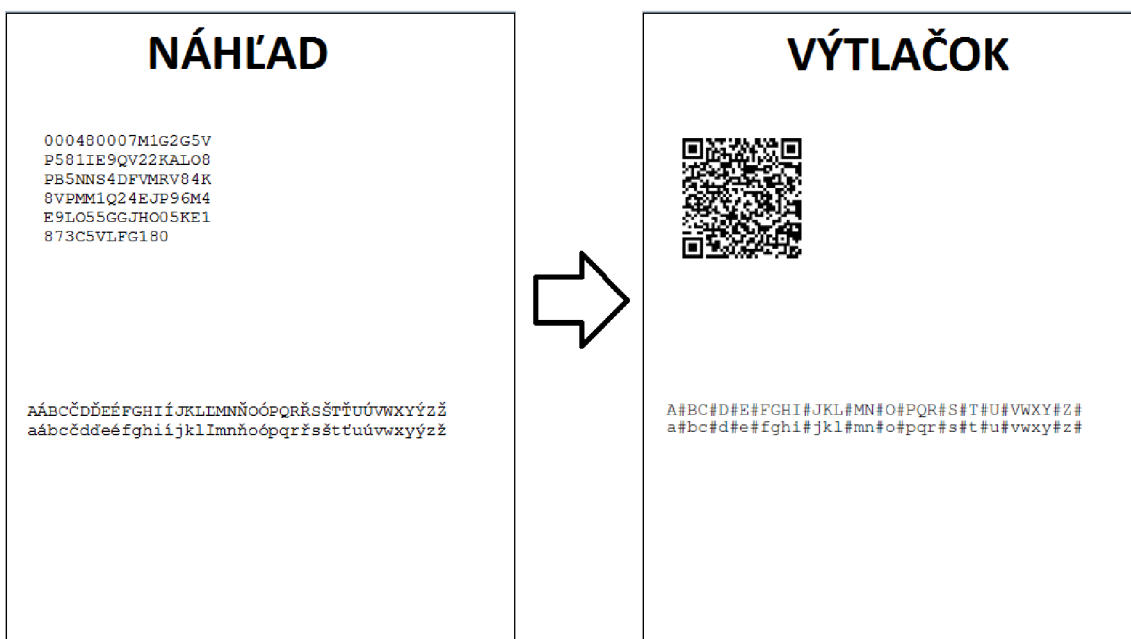
7.2 Testovanie metód pre vytvorenie QR kódu

Tento test bude zameraný na porovnanie rôznych metód vytvorenia QR kódu. Pri každej metóde budú popísané jeho výhody a nevýhody. Na základe takéhoto zhodnotenia bude vybraný jeden spôsob, ktorý bude použitý vo výslednom riešení.

7.2.1 Využitie špeciálneho výstupného zariadenia

Pre tlač QR kódu je v tomto prípade použitý špeciálny typ zariadenia. Výhodou tohto prístupu je, jeho rýchlosť, jednoduchosť a veľmi malá pamäťová náročnosť. U tohto riešenia však nevýhody prevládajú nad výhodami. Hlavnou nevýhodou je, že dáta sa do QR kódu kódujú až pri výstupe tlačiarne. Pri náhľade teda užívateľ nemá možnosť vidieť výsledný výstup. Riešenie nie je možné

využiť teda ani na posielanie faktúr v elektronickej podobe. Ďalšou veľkou nevýhodou je, že typ zariadenia ZBIWPP podporuje iba znakovú sadu Latin 1, kvôli čomu je pre tlač slovenských faktúr defacto nepoužiteľný. Rozdiel medzi tlačovým náhľadom a reálnym výtlačkom zobrazuje Obrázok 7.1. Poslednou nevýhodou je nutnosť inštalácie nového typu zariadenia a jeho potreba nastavenia ako predvolená tlačiareň. Na základe popísaných výhod a nevýhod, bola táto alternatíva vyhodnotená ako nevyhovujúca pre finálne riešenie.



Obrázok 7.1 Porovnanie tlačového náhľadu s reálnym výtlačkom

7.2.2 Využitie PDF tlačiarne

Predchádzajúce riešenie je v tejto alternatíve rozšírené o využitie frontendovej PDF tlačiarne. Pozrieme sa na zhodnotenie výhod a nevýhod tohto prístupu. Výhodami sú jednoduchosť a možnosť vidieť výstup kódovania do QR kódu priamo vo vzniknutom PDF bez nutnosti tlače. Využitie PDF tlačiarne pridáva nutnosť každú faktúru generovať do PDF, čo predlžuje čas generovania a spracovania faktúry a rovnako zvyšuje aj pamäťovú náročnosť riešenia. Vyžadovaná je tiež inštalácia PDF tlačiarne na každom počítači, kde budú faktúry generované. Nevýhodou tohto riešenia ostáva stále to, že QR kód nie je vidieť hneď v tlačovom náhľade a nepodporuje našu znakovú sadu, čo riešenie robí nepoužiteľným.

7.2.3 Vytvorenie QR kódu samostatnou aplikáciou

Vytvorenie QR kódu samostatnou aplikáciou, prináša voči predchádzajúcim alternatívam množstvo zmien. Medzi výhody tejto metódy patrí hlavne to, že QR kód je vytvorený v nezávislej aplikácii a potom je používaný ako obrázok, z čoho plynie, že je nezávislý na kódovaní formulára. Taktiež je možné výsledný QR kód vidieť už pri tlačovom náhľade. Ďalšou výhodou je, že tento prístup nevyžaduje nutnosť predefinovať nastavenia tlačových zariadení. Nevýhodou je však, že aplikácia na vytváranie QR kódu musí byť umiestnená na klientskom počítači alebo na serveri a komunikácia medzi ABAPom a aplikáciou prináša časovú režiu. QR kód vytvorený externou aplikáciou je umiestnený do správy grafík, čo zvyšuje pamäťovú náročnosť riešenia, no prináša so sebou aj možnosť neskoršej kontroly vygenerovaného kódu ako aj možnosť opakovaného použitia QR kódu bez nutnosti opätovného generovania. Táto alternatíva sa s ohľadom na svoje výhody a nevýhody ukázala ako najvýhodnejšia a je použitá vo výslednom riešení.

7.2.4 Vytvorenie QR kódu vo formulári z matice

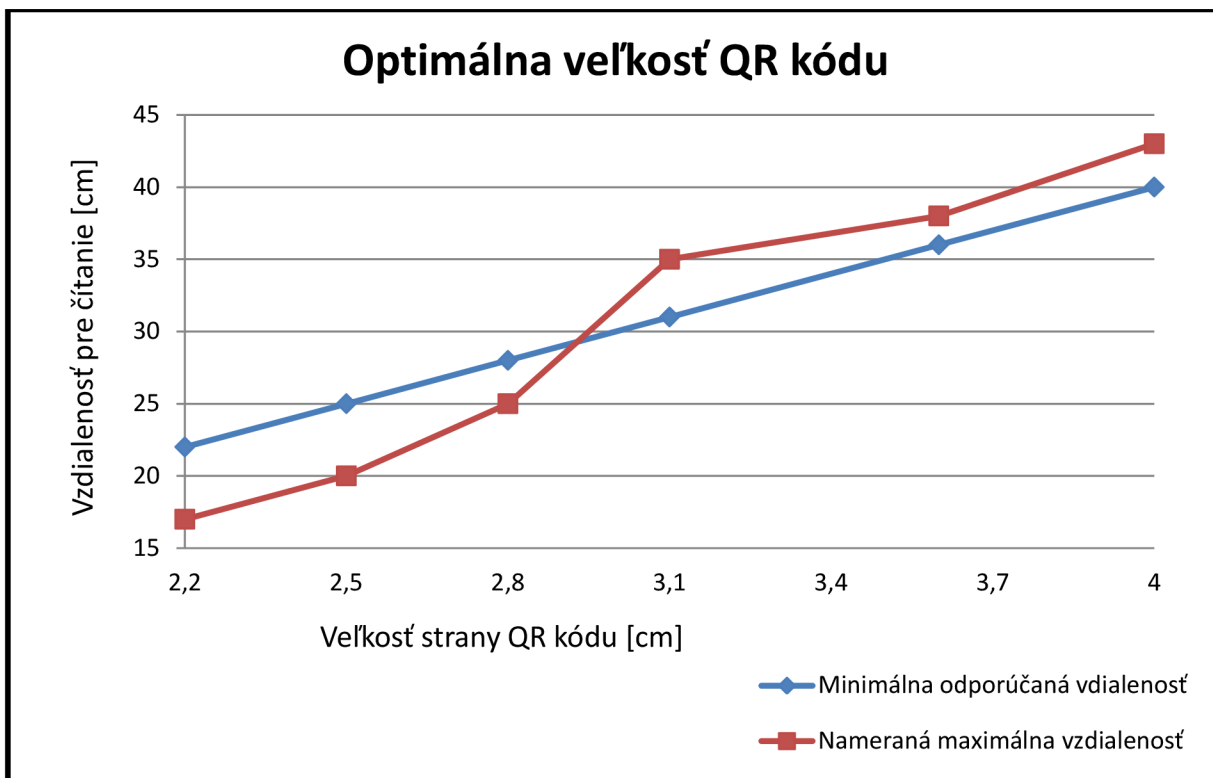
Posledná testovaná alternatíva je úpravou predošlého riešenia a bola navrhnutá so zámerom eliminovať časové a pamäťové nároky na vytvorenie a uloženie obrázka QR kódu do správy grafík. Pamäťová náročnosť riešenia bola zmenšená, no čas generovania kódu sa predĺžil keďže bolo potrebné vyhodnotiť cca. 1700 podmienok (pre každú bunku v matici 41x41) a rovnaký počet krát načítať obrázok bieleho alebo čierneho štvorca. Takéto generovanie QR kódu môže tvoriť alternatívu k predošlému riešeniu a záleží len na tom, či sa kladie väčší dôraz na použitú pamäť alebo čas generovania.

7.3 Testovanie veľkosti QR kódu

Po tom ako sme otestovali rôzne spôsoby generovania QR kódu a zvolili najvhodnejší pre výslednú aplikáciu tento test bude zameraný na hľadanie optimálnej veľkosti QR kódu. Pri jej určení budeme vychádzať z odporúčania, že minimálna veľkosť QR kódu je taká, aby sa dal prečítať zo vzdialenosti, ktorá je desaťnásobkom jeho veľkosti (strany). Na testovanie prvý krát využijeme mobilnú aplikáciu Slovenskej sporiteľne s názvom Platby. Tejto a aj ďalším aplikáciám sa budeme venovať v nasledujúcej podkapitole.

Test je založený na tom, že budeme mať k dispozícii QR kód veľkosti od 2,2 cm do 4 cm a budeme merať maximálnu vzdialenosť, z ktorej vie mobilná aplikácia QR kód prečítať. Namerané hodnoty ako aj odporúčaná minimálna vzdialenosť sú zobrazené v Graf 7.1. Môžeme vidieť, že pre malé veľkosti, je potrebné čítať kód pomerne z blízka. Zlom nastáva približne pri veľkosti 2,9 cm, kde sa pri čítaní dostávame nad minimálnu odporúčanú vzdialenosť. Hodnotu potvrdzujú aj údaje uvedené v by square špecifikácii, kde je ako minimum označená veľkosť 3 cm. Kód veľkosti 3,1 cm už je následne možné prečítať z maximálnej vzdialenosti až 35 cm, čo je viac ako 11-násobok veľkosti.

S ohľadom na dostupný priestor pre QR kód na faktúre som ako optimálnu veľkosť zvolil práve rozmer 3,1 x 3,1 cm.



Graf 7.1 Určenie vhodnej veľkosti QR kódu

7.4 Testovanie QR kódov na faktúrach mobilnými aplikáciami

Predchádzajúce testy boli predpokladom pre správnu funkčnosť kódovania platobných dát faktúr do QR kódov. V posledných testoch sa zameriame na otestovanie funkčnosti výsledného riešenia, a teda na testovanie QR kódov s platobnými údajmi, ktoré sú súčasťou faktúr za teplo, otestujeme prípady, kedy sú použité viaceré bankové spojenia v jednom príkaze a taktiež možnosť, kedy sú v jednom kóde obsiahnuté viaceré platby. Posledný test bude zameraný na overenie funkčnosti kódovania trvalého príkazu. Správnosť vytvorených QR kódov overíme prostredníctvom mobilných aplikácií rôznych bánk, pre ktoré sú QR kódy určené. Používať budeme aplikáciu Platby od Slovenskej sporiteľne, Smart Banking od ČSOB, BankAir od UnicreditBank a aplikáciu od Tatra banky. Prvé dve aplikácie vyžadujú, aby užívateľ mal reálne založený účet u danej banky a druhé dve poskytujú demo účty, na ktorých je možné vyskúšať funkčnosť internetbankingu. Teraz si popíšeme jednotlivé testy.

7.4.1 Test základných údajov pre úhradu faktúry

Test bude kontrolovať načítané údaje z QR kódu s údajmi uvedenými na faktúre. Údaje použité vo faktúre, ktoré nás budú zaujímať, sú na výreze faktúry (Obrázok 7.2). Zaujímáť nás z týchto údajov bude číslo bankového účtu a banka príjemcu, konštantný a variabilný symbol, dátum splatnosti, suma na úhradu, mena a poznámka pre príjemcu. Poznámka je v tvare "Faktura za teplo - Apríl 2012". V tomto teste budeme okrem správneho načítania údajov sledovať ako sa zachovávajú aplikácie pri dátume splatnosti, ktorý je v minulosti.



Banka: Československá obchodná banka, a.s.			
Účet: 25642093/7500			
Konštantný symbol	0008	Dátum vyhotovenia	10.05.2012
Variabilný symbol	4113370412	Dátum odoslania	11.05.2012
Forma úhrady	Prevodný príkaz	Dátum splatnosti	25.05.2012
		Dátum dodania tovaru	30.04.2012

Fakturujeme Vám dodávku tepelnej energie za mesiac Apríl 2012 .
Opakované dodanie tovaru podľa § 19 ods.3 zákona o DPH.

Fakturované veličiny	Množstvo	Jednotka	Jednotková cena	Cena bez DPH
Regulačný príkon (FIXNÁ ZLOŽKA)	0,0787	kW	99,92540	7,86 EUR
Teplo (VARIABILNÁ ZLOŽKA) Domácnosti	20,00	kWh	0,03590	0,72 EUR

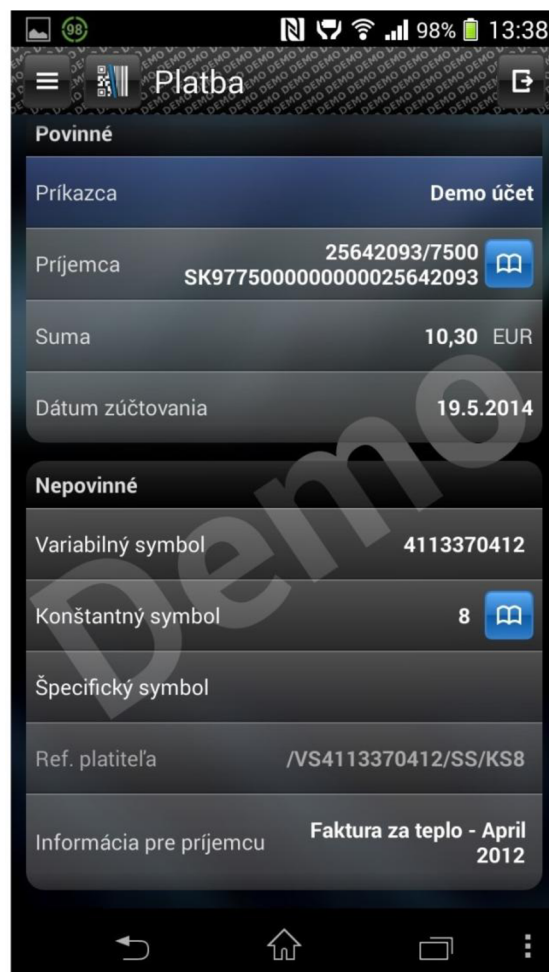
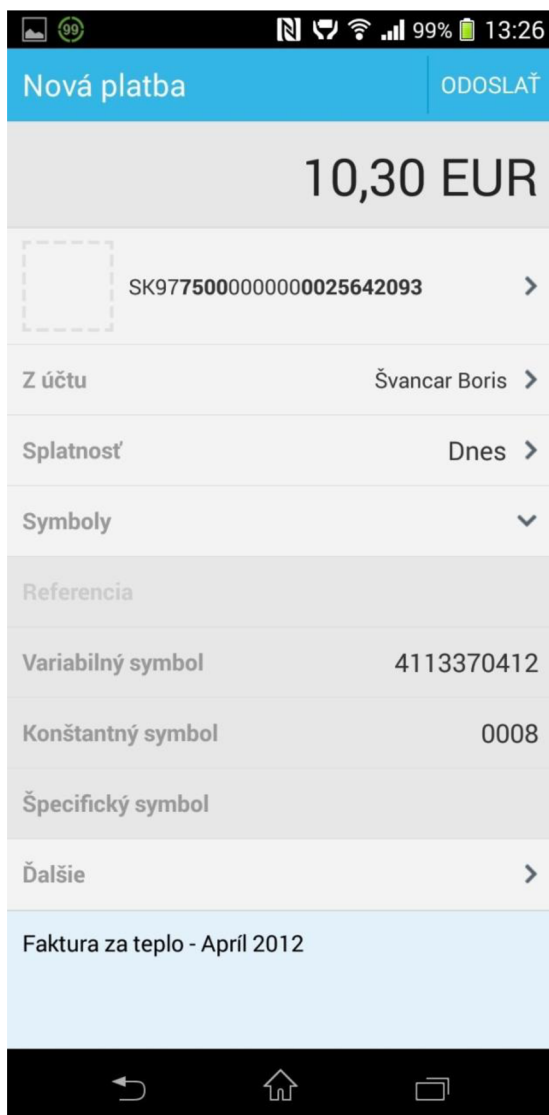
Cena spolu bez DPH 8,58 EUR

Vyúčtovanie	
Spolu bez DPH	8,58 EUR
Základ dane	8,58 EUR
DPH 20%	1,72 EUR
Spolu s DPH	10,30 EUR
Záloha	0,00 EUR
Celkom s DPH	10,30 EUR
Cent. vyrovnanie	0,00 EUR
K úhrade	10,30 EUR

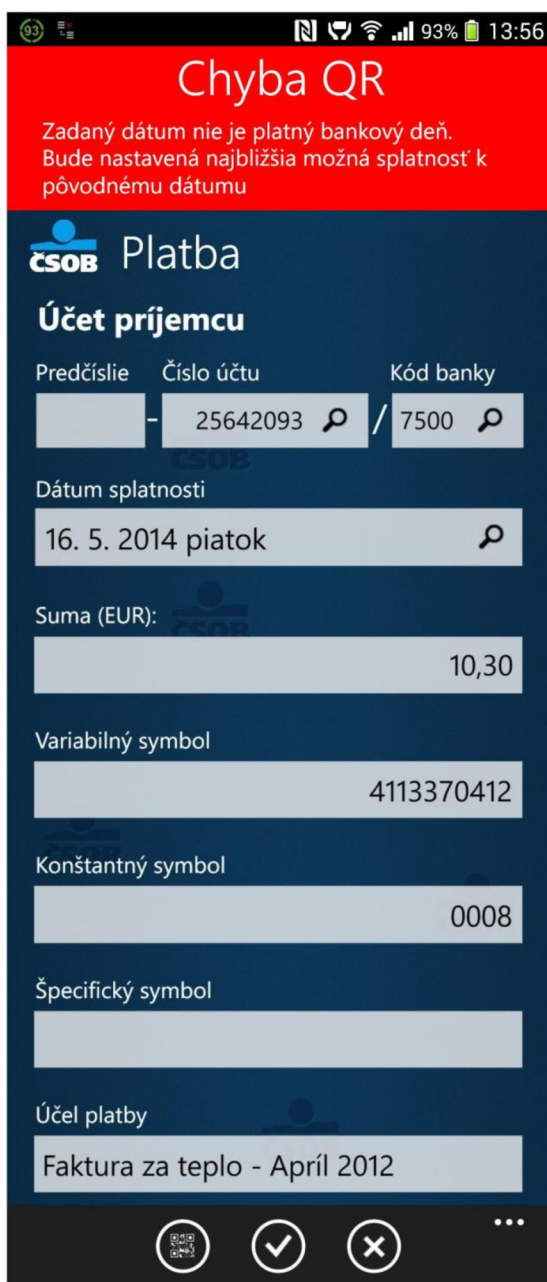


Obrázok 7.2 Výrez z faktúry za teplo

Uvedieme si postupne ukážky načítaných dát tejto faktúry v aplikáciách jednotlivých bánk. Obrázky Obrázok 7.3 a Obrázok 7.4 zobrazujú dáta načítané z QR kódu na faktúre. Na pravej strane obrázku 7.3 je aplikácia Slovenskej sporiteľne, na ľavej aplikácia Tatra banky a na obrázku 7.4 je na pravej strane aplikácia ČSOB a na ľavej UniCredit banky. Vo všetkých prípadoch načítané dáta zodpovedajú údajom na faktúre. Čo sa týka dátumu splatnosti, ten bol vo všetkých aplikáciách automaticky nastavený na aktuálny dátum. Jediná aplikácia, ktorá upozornila na skutočnosť, že dátum splatnosti je v minulosti, bola aplikácia ČSOB. Vo výsledku tento test potvrdil správnosť prepojenia vytvoreného funkčného modulu pre generovanie QR kódov s faktúrami za teplo.



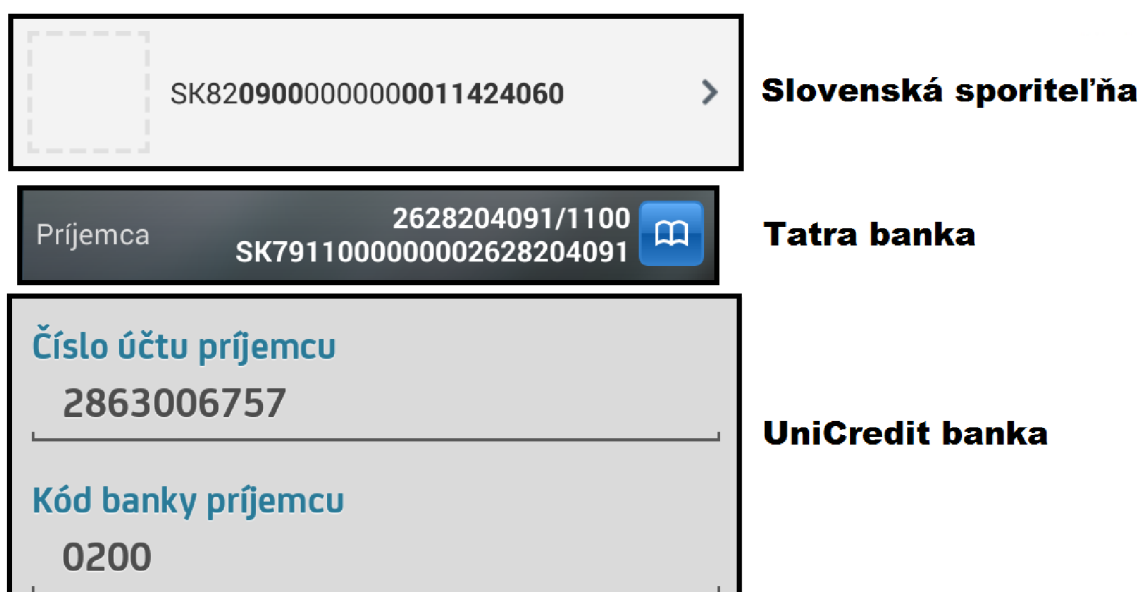
Obrázok 7.3 Dáta z QR kódu načítané mobilnou aplikáciou Slovenskej sporiteľne a Tatra banky



Obrázok 7.4 Dáta z QR kódu načítané mobilnou aplikáciou ČSOB a UniCredit banky

7.4.2 Test kódovania viacerých bankových spojení

Predchádzajúcim testom sme potvrdili funkčnosť kódovania platobných údajov v základnom tvare s jedným bankovým účtom príjemcu. Štandard by square však umožňuje, aby v časti s bankovým spojením príjemcu platby bolo uvedených viac bankových účtov. Viaceré bankové účty sú samozrejmosťou u mnohých firiem, ktoré poskytujú služby zákazníkom, preto sa v tomto teste zameriame na zakódovanie troch možných bankových spojení ku jednej platbe. Budeme skúmať ako sa jednotlivé aplikácie zachovajú pri načítaní QR kódu, ktorý bude obsahovať bankové spojenie po poradí na VÚB banku (kód banky 0200), na Slovenskú sporiteľňu (kód 0900) a na Tatra banku (kód 1100).

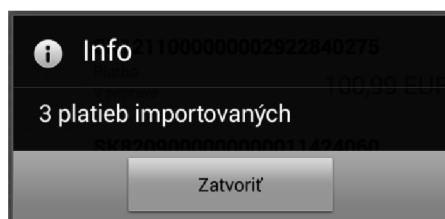


Obrázok 7.5 Načítanie QR kódu obsahujúceho 3 bankové účty

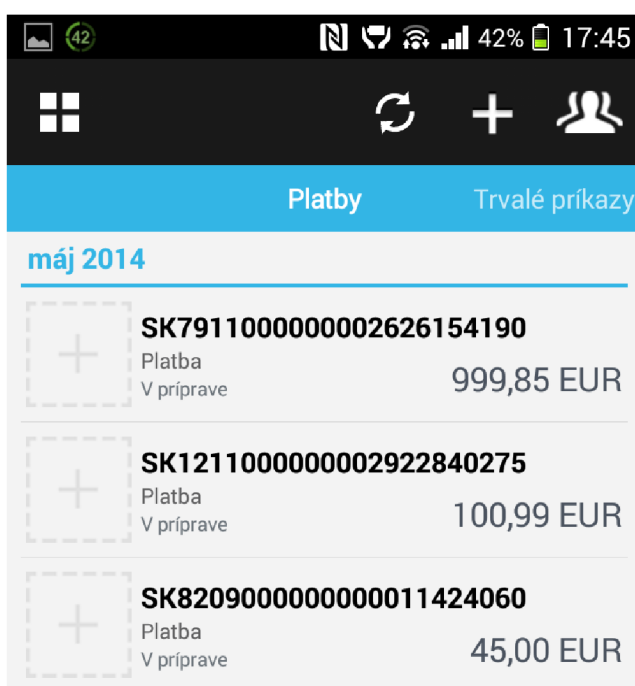
Obrázok 7.5 ukazuje dáta načítané jednotlivými aplikáciami. Vidíme, že každá aplikácia načítala iné bankové spojenie. Aplikácia Slovenskej sporiteľne načítala bankové spojenie na Slovenskú sporiteľňu, Tatra banka spojenie na Tatra banku. Keďže sa medzi údajmi nenachádzalo spojenie na UniCredit banku, jej aplikácia vybrala prvé uvedené bankové spojenie, a to na VÚB banku. Test preveril správnosť kódovania viacerých bankových spojení k jednej platbe a poukázal na správanie mobilných aplikácií, ktoré sa snažia vyhľadať spojenie v rámci svojej banky.

7.4.3 Test kódovania viacerých platobných príkazov do jedného QR kódu

Tretí test bude zameraný na uloženie viacerých platieb do jedného kódu a bude v ňom použitý QR kód kódujúci tri rôzne platobné príkazy. Po načítaní takéhoto kódu aplikáciou Slovenskej sporiteľne táto upozorní informáciou o 3 importovaných platbách (Obrázok 7.6) a tie sa následne zobrazia v zozname platieb so statusom "V príprave" (Obrázok 7.7).

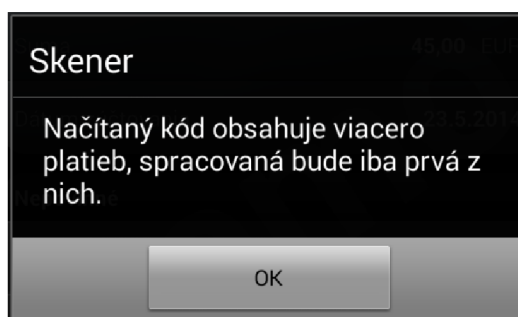


Obrázok 7.6 Informácia aplikácie Slovenskej sporiteľne



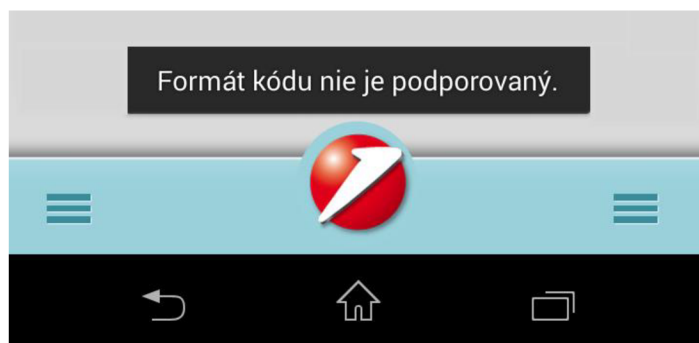
Obrázok 7.7 Tri platby načítané z QR kódu

Pri načítaní kódu aplikáciou Tatra banky sa zobrazí užívateľovi upozornenie, že načítaný kód obsahuje viacero platieb a spracovaná bude len prvá z nich (Obrázok 7.8).



Obrázok 7.8 Upozornenie aplikácie Tatra banky pri načítaní viacerých platieb

Aplikácia od UniCredit banky pri načítaní vypíše hlásenie, že formát kódu nie je podporovaný a nenačíta žiadne údaje (Obrázok 7.9).



Obrázok 7.9 Chyba pri načítaní viacerých platieb aplikáciou UniCredit banky

Z výsledkov testu vyplynulo, že napriek tomu, že štandard by square umožňuje kódovanie viacerých platieb do jedného QR kódu a táto funkčnosť je implementovaná vo funkčnom module neodporúča sa ju používať, nakoľko len jedna z troch testovaných aplikácií dokáže načítať správne všetky údaje.

7.4.4 Test kódovania trvalého príkazu

Posledný test bude testovať prípad, kedy bude QR kód obsahovať údaje pre vytvorenie trvalého príkazu. Pre tento test bol vytvorený vstup, ktorý obsahoval platobné údaje rovnako ako jednorazový príkaz, plus obsahoval príznak rozšírenia o trvalý príkaz, ktorého obsahom boli údaje o týždennom opakovaní trvalého príkazu po dobu jeden mesiac. QR kód vytvorený na základe tohto vstupu bol otestovaný mobilnými aplikáciami. Výsledkom načítania cez aplikáciu Platby Slovenskej sporiteľne bolo upozornenie, že aplikácia v súčasnej dobe nepodporuje platby typu trvalý príkaz ani inkaso (Obrázok 7.10). Aplikácia Tatra banky pri načítaní kódu zobrazí hlásenie

o chybe. Posledná aplikácia od UniCredit banky načíta údaje ako vo formáte, ako keby šlo o jednorazovú platbu. Takýto prístup môže byť pre užívateľa zavádzajúci, keďže sa deje bez jeho upozornenia.



Obrázok 7.10 Informácia aplikácie Slovenskej sporiteľne pri čítaní QR kódov s trvalým príkazom alebo inkasom

Test preukázal, že v súčasnej dobe banky vo svojich aplikáciách nemajú zabudovanú podporu QR kódov s trvalým príkazom ani s inkasom. Toto zistenie tvorilo hlavný dôvod, prečo v práci nebolo implementované práve kódovanie inkasa.

7.5 Výsledky testovania

Vytvorený funkčný modul bol podrobený dôkladnému testovaniu s využitím testovacích dát poskytnutých tvorcami štandardu by square. Na základe výsledkov testov bola potvrdená jeho správna funkčnosť. V druhom teste boli porovnané rôzne metódy generovania QR kódu. Metódy využívajúce špeciálny typ výstupného zariadenia boli vyhodnotené ako nepoužiteľné pre finálne riešenie a pre generovanie QR kódov bola zvolená alternatíva využívajúca pre generovanie externú aplikáciu. V treťom teste sme hľadali optimálnu veľkosť QR kódu na faktúre. Ako najvhodnejší bol vyhodnotený rozmer 31 x 31mm, ktorý zodpovedá údajom v špecifikácii by square. Štvrtý test bol rozdelený na niekoľko častí. Najskôr sme testovali využitie funkčného modulu pre kódovanie dát na faktúrach za teplo. Správnosť riešenia bola potvrdená načítaním QR kódu mobilnými aplikáciami štyroch rôznych bánk. Ďalej sme otestovali kódovanie viacerých bankových účtov k jednému platobnému príkazu a správanie aplikácií pri čítaní takéhoto vstupu. V záver sme ukázali správnosť kódovania viacerých platobných príkazov do jedného QR kódu a tiež kódovanie trvalého príkazu. Preukázali sme však, že v súčasnej dobe tieto dve možnosti nie je vhodné používať s ohľadom na obmedzenú funkčnosť mobilných aplikácií poskytovaných bankami.

7.6 Možnosti ďalšieho rozvoja

Vytvorené riešenie ponúka v mnohých smerov rozsiahle možnosti ďalšieho rozvoja. Pri štandarde by square je to možnosť rozšírenia PAY by square o INVOICE by square. Jeho špecifikácia nie je voľne dostupná, no chystá sa jeho komerčné nasadenie a dá sa teda očakávať aj jeho zverejnenie. Testy preukázali veľké rezervy v mobilných aplikáciách bánk, ktoré v súčasnej dobe zo štandardu PAY by square nepodporujú trvalé príkazy a inkasá. Po zdokonalení týchto aplikácií je možnosť riešenie rozšíriť práve o kódovanie inkasa.

Veľké možnosti rozvoja sú však pri kódovaní by square sekvencie do QR kódu. Tu je asi najlepšou alternatívou k súčasnému riešeniu využitie Adobe Forms, ktoré sú nástupcami formulárov SmartForms. Adobe Forms majú zabudovanú priamu podporu generovania QR kódov. Tieto formuláre však nie sú súčasťou štandardnej ABAP inštalácie SAP a nie sú zatiaľ u zákazníkov tak rozsiahle ako SmartForms. Inou alternatívou je využitie spoplatneného riešenia pre tlač čiarových kódov pre systém SAP ako napríklad Barcode DLL od firmy MW6 alebo obdobné riešenie od firmy TEC-IT. Pri úspešnom vyladení riešenia sa naskytajú možnosti toto riešenie zapracovať aj do iných faktúr, ktoré sú súčasťou štandardu SAP. Najlepšiu perspektívu pre ďalší rozvoj však poskytuje možnosť toto riešenie nasadiť do testovacej a následne aj produktívnej prevádzky u niektorého zo zákazníkov firmy RKMC, s.r.o. Faktúry s QR kódmi by mohli v tomto prípade uľahčiť úhrady stovkám ľudí.

8 Záver

V práci sme sa zoznámili so systémom SAP, s procesom generovania QR kódov a so štandardom by square. Pri systéme SAP som predstavil jeho históriu a základné produkty firmy SAP. Bližšie som sa zameril na systém SAP ERP, pre ktorý je určený aj modul pre generovanie platobných údajov do QR kódov, ktorý je riešením práce. So zameraním na požadované riešenie som bližšie popísal ABAP dictionary, programovací jazyk ABAP, nástroj pre vytváranie formulárov SmartForms a výstupné zariadenia v SAPe.

V druhej časti práce som popísal jednotlivé kroky v procese generovania QR kódov. Nakoľko podrobný popis všetkých krokov je veľmi rozsiahly uviedol som pri určitých krokoch najdôležitejšie informácie a detailný popis je možné nájsť v uvedenej literatúre.

Po QR kódoch som predstavil štandard by square Slovenskej bankovej asociácie pre kódovanie platobných údajov do QR kódov. Tento štandard sa skladá z dvoch častí a to INVOICE a PAY by square. Nakoľko INVOICE by square nie je verejne prístupný zameril som sa na PAY by square, pre ktorý som detailne popísal zakódovanie dát od štruktúry vstupného súboru až po vytvorenie výslednej by square sekvencie. Keďže štandard PAY by square neuvádza podrobné technické detaily niektorých krokov využil som možnosť osobného stretnutia s programátormi tohto riešenia, ktorí mi potrebné informácie poskytli.

Tieto teoretické predpoklady som následne využil pri návrhu riešenia modulu pre systém SAP, ktorý umožní platobné údaje faktúr uložené v systéme alebo zadané užívateľom zakódovať do by square sekvencie. Taktiež som navrhol niekoľko možných riešení, ktoré umožnia túto sekvenciu zakódovať do výsledného QR kódu.

Vytvorený návrh bol v rámci diplomovej práce implementovaný v jazyku ABAP a Java a následne podrobený testovaniu. Na testovanie funkčného modulu generujúceho by square sekvenciu som použil testovacie dáta poskytnuté autormi štandardu. Následne som vyhodnotil, jednotlivé alternatívy pre generovanie QR kódu. Z implementovaných alternatíva sa metóda využívajúca špeciálny typ tlačového zariadenia ukázala ako nepoužiteľná, a preto bola na generovanie kódov použitá externá aplikácia volaná zo systému SAP. Validita faktúr s platobnými údajmi v QR kódoch bola overená s využitím mobilných aplikácií od niekoľkých bánk. Rovnako bola overená správnosť kódovania viacerých bankových účtov ku jednej platbe, viacerých platieb do jedného QR kódu a kódovanie trvalého príkazu. Pri týchto alternatívach však treba brať ohľad na obmedzenú funkčnosť mobilných aplikácií.

Vytvorené riešenie poskytuje množstvo príležitostí pre ďalší rozvoj štandardu PAY by square ako aj možnosť rozšírenia o štandard INVOICE by square. Pre generovanie QR kódov je možnosť využitia spoplatnenej knižnice pre generovanie QR kódov priamo v systéme SAP, alebo využitie Adobe Forms.

Nakoľko riešenie by square pre systém SAP nebolo zatiaľ k dispozícii existuje reálna možnosť ho po úspešnej testovacej prevádzke nasadiť v budúcnosti do produktívnej prevádzky u niektorých zákazníkov využívajúcich systém SAP.

Literatúra

- [1] ACRO SOFTWARE INC. *Cute DPF writer* [online]. 2014 [cit. 2014-05-0].
Dostupné z: <http://www.cutepdf.com>
- [2] ANDERSON, George W. *Naučte se SAP za 24 hodin*. 1. vyd. Brno: Computer Press, 2012. ISBN 978-80-251-3685-0.
- [3] Cyklický redundantní součet. In: *Wikipedia: the free encyclopedia* [online]. 2013 [cit. 2013-12-22].
Dostupné z:
http://cs.wikipedia.org/wiki/Cyklick%C3%BD_redundantn%C3%AD_sou%C4%8Det
- [4] DRÁBEK, Vladimír. *Přednášky z předmětu Kódování a komprese*. 2013
- [5] EUROPEAN CENTRAL BANK. *SEPA* [online]. [cit. 2013-12-23].
Dostupné z: <http://www.ecb.europa.eu/paym/sepa/html/index.en.html>
- [6] FORSYS. *PAY by square specifications: Specifications for PAY by square standard* [online]. 2013 [cit. 2013-12-10].
Dostupné z: <http://www.bysquare.com/sk/wp-content/uploads/2013/02/by-square-PAY-specifications-1.0.0.pdf>
- [7] ISO 18004:2000. *Information technology — Automatic identification and data capture techniques — Bar code symbology — QR Code* [online].
Dostupné z: http://raidanii.net/files/datasheets/misc/qr_code.pdf
- [8] KÜHNHAUSER, Karl-Heinz. *ABAP: výukový kurz*. Vyd. 1. Brno: Computer Press, 2009. ISBN 978-80-251-2117-7.
- [9] PAVLOV, Igor. *LZMA SDK* [online]. 2013 [cit. 2014-05-01].
Dostupné z: <http://www.7-zip.org/sdk.html>
- [10] QR Code. In: *Wikipedia: the free encyclopedia* [online]. 2014 [cit. 2013-12-20].
Dostupné z: http://en.wikipedia.org/wiki/QR_code
- [11] RUNGTA, Krishna. *SAP Training Hub: All About Smart Forms* [online]. 2011 [cit. 2014-01-04].
Dostupné z: <http://www.saptraininghub.com/smart-forms/>

- [12] RUSSELL, Robert. *Barcode Writer in Pure Postscript - download device type* [online]. 2010 [cit. 2014-05-01].
Dostupné z: <http://bit.ly/zbwippqr>
- [13] SAP R3 [online]. [cit. 2014-01-03].
Dostupné z: <http://www.best-career-guide-for-erp.com/sap-r3.html>
- [14] SAP. *SAP Help Portal* [online]. [cit. 2014-01-03].
Dostupné z: <http://help.sap.com>
- [15] THONKY. *QR Code Tutorial* [online]. 2013 [cit. 2013-12-20].
Dostupné z: <http://www.thonky.com/qr-code-tutorial/introduction/>

Príloha A - Obsah DVD

Priložené DVD obsahuje tieto zložky:

- **bin/** - spustiteľné Java aplikácie
- **src_abap/** - zdrojové súbory pre ABAP
- **src_java/** - zdrojové súbory Java aplikácií
- **lib/** - knižnice využívané Java aplikáciami
- **bySquare/** - špecifikácia PAY by square
- **examples/** - testovacie príklady od tvorcov štandardu
- **word/** - zdrojový text diplomovej práce vo Worde
- **pdf/** - text diplomovej práce v .pdf

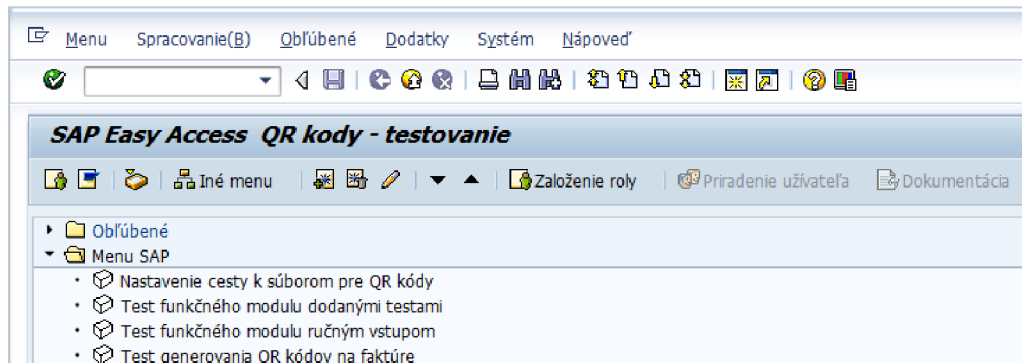
Príloha B - Užívateľská dokumentácia

Vytvorenie pracovného adresára

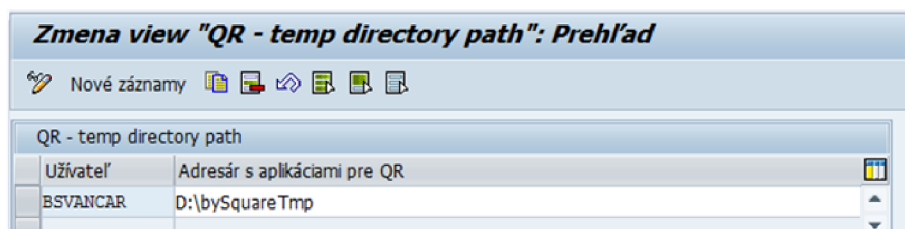
Prvý krok potrebný pre správnu funkčnosť generovania QR kódov je inštalácia externých programov, ktoré slúžia na LZMA kompresiu a generovanie QR kódu. V počítači, z ktorého sa bude spúšťať SAP GUI je potrebné vytvoriť priečinok s ľubovoľným názvom a do tohto priečinka nakopírovať programy LZMA.jar, QR_CODE.jar a adresár lib obsahujúci knižnice javase-2.0.jar, LzmaAlone.jar a zxing-core-2.0. Tento priečinok bude okrem iného využívaný aj na ukladanie vytvorených QR kódov.

Nastavenie cesty k pracovnému adresáru v systéme SAP

Po prihlásení do systému SAP, je možné spustiť menu, ktoré bolo vytvorené pre testovanie riešenia diplomovej práce (Obrázok 1). Menu je možné zobraziť zadaním transakcie ZQR do príkazového poľa v ľavom hornom rohu.



Obrázok 1: Menu v SAP pre testovanie QR kódov

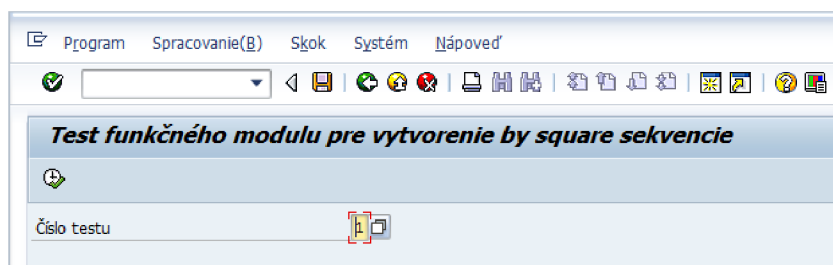


Obrázok 2: Definovanie cesty k pracovnému adresáru

Kliknutím na prvú položku v menu: *Nastavenie cesty k súborom pre QR kódy* alebo zadáním transakcie ZQR_PATH sa spustí program, kde je potrebné pre daného užívateľa definovať cestu k pracovnému adresáru (Obrázok 2).

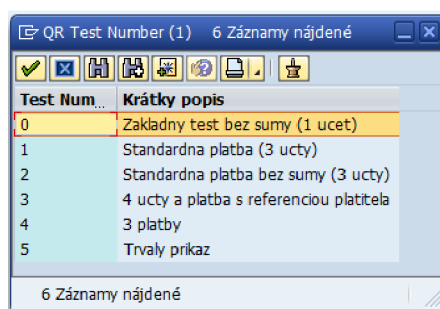
Testovacia aplikácia s preddefinovanými testami

Pre testovanie funkčného modulu je možné kliknutím na položku menu: *Test funkčného modulu dodanými testami* alebo transakciou ZQR_TEST spustiť testovanie generovania by square sekvencie a jej kódovania do QR kódu s využitím preddefinovaných testov. Spustením sa zobrazí program, kde je potrebné zadať číslo testu (Obrázok 3)




Obrázok 3: Program na testovanie funkčného modulu

Kliknutím na ikonu  alebo stlačením tlačidla F4 sa zobrazí nápoveda vyhľadavanie (matchcode)



Obrázok 4: Matchcode pre číslo testu

Po výbere testu sa kliknutím na  alebo stlačením tlačidla F8 spustí program, ktorý z preddefinovaných dát vytvorí by square sekvenciu a tu následne zakóduje do QR kódu. Výstupom programu je táto sekvencia a názov QR kódu uloženého v správe grafik. V spodnej časti výstupu je zobrazený preddefinovaný vstup, ktorý sa v danom teste použil. Výstup programu zobrazuje Obrázok 5.

```
Test funkčného modulu pre vytvorenie by square sekvencie

Test funkčného modulu pre vytvorenie by square sekvencie

Vygenerovana by square sekvencia:

000CI000E8FCI05HB5CD2DV6CFHXQ9QUDQE1CUU0RRD3KKBA33VI706M9U2Q00J6LLTR9FCORM2JSNNEMIRK5QH620PM23N8D26DBHHLHDIT5VU90AIEOMN4
DHKGS5AH00M3COE598FV243R9B76JAOISHRJRR048QD38IDJ2QDNGLQ3SP9IQBCLCASH8C04VL6GB1RHP1UCAQTCL1AB4R36G3BVGJT7PRUSCR7K2N08CP6NF
ULITEU4F

QR kod bol ulozeny do spravy grafik pod nazvom: QR12345_1

InvoiceID: 001

PaymentOptions: 1

Amount: 582.35

Currency: EUR

PaymentDueDate: 06.12.2012

VariableSymbol: 1200097151

ConstantSymbol: 0308

SpecificSymbol: 062012

OriginatorsReferenceInfo:

Note: Telefonica: mobile internet - 2012/03

IBAN: SK79110000000002628204091

BIC: TATRSKEX


IBAN: SK82090000000000011424060

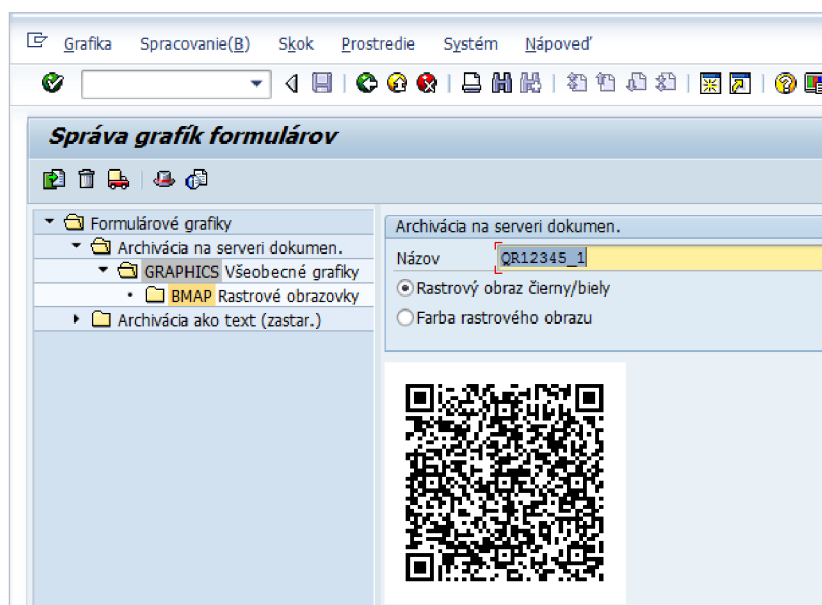
BIC: GIBASKEX

IBAN: SK64020000000002863006757

BIC: SUBASKEX
```


Obrázok 5: Výstup programu pre testovanie

Vygenerovaný obrázok QR kódu je možné nájsť pod vypísaným názvom v pracovnom adresári na lokálnom počítači alebo v správe grafik. Správu grafik je možné spustiť zadaním transakcie SE78 do príkazového poľa. Následne je potrebné v ľavej časti rozkliknúť voľbu GRAPHICS Všeobecné grafiky a následne dvojkliknutím na BMAP Rastrové obrázky sa zobrazí textové pole na zadanie názvu obrázka. Po zadaní názvu je možné obrázok zobrazit' kliknutím na ikonu  alebo stlačením klávesy F8.






Obrázok 6: Zobrazenie QR kódu cez správu grafík


Testovacia aplikácia s ručným vstupom

Pre testovanie funkčného modulu s využitím ručne zadaných dát je možné použiť transakciu ZQR_TEST_FM alebo voľbu v menu *Testovanie funkčného modulu ručným vstupom*. Po spustení sa zobrazí štandardný program systému SAP na správu funkčných modulov. Kliknutím na  alebo stlačením tlačidla F8 sa zobrazí obrazovka pre zadanie vstupných parametrov. Na prvej obrazovke (Obrázok 7) je potrebné zadať číslo faktúry, ktoré bude použité pri vytvorení názvu QR kódu a následne je potrebné vyplniť platobné dáta.



Testovanie funkč.modulu: Vstup.obrazovka

  Ladenie  Adresár test.dát










Test pre skupinu funkcií ZQR1
 Funkčný modul ZQR_CREATE_PAY
 Malé/veľké písmo

Parametre importu	Hodnota
GS_PAY	
GV_VBELN	<input type="text"/>

Obrázok 7: Obrazovka pre zadávanie vstupných dát funkčného modulu

Kliknutím na  sa dostaneme na najvyššiu úroveň dát, kde je potrebné zadať číslo faktúry. Po kliknutí na  je možné definovať dáta platobného príkazu (Obrázok 8).





Editor štruktúry: Zmena PAYMENTS





 Stĺpec
  Záznam
  Nový riadok
  Zdvojenie riadka
  Metadáta


1245 1 Záznam

P	AMOUNT	CURRE	PAYMENTDUE	VARIABLESY	CONS	SPECIFICSY	ORGIN_REF_INF	PAYMENTNOTE
1		EUR						

Obrázok 8: Zadávanie platobných dát


V platobnom príkaze je možné bankové účty definovať rovnako po kliknutí na  v stĺpci BANK_ACCOUNTS. V prípade, že užívateľ nevie určiť o aký stĺpec sa jedná, vie si jeho názov zobrazit' tlačidlom F1. Pridávanie nových záznamov pre bankové účty alebo pre nové platobné príkazy je možné tlačidlom . Návrat o úroveň vyššie je možná cez . Po vyplnení všetkých dát je potrebné vrátiť sa na najvyššiu úroveň na Obrázku 7. Odtiaľto je možné kliknutím na  alebo stlačením tlačidla F8 spustiť test funkčného modulu. Po jeho úspešnom vykonaní je zobrazená výsledná by square sekvencia (Obrázok 9) a v pracovnom priečinku a rovnako v správe grafik je vytvorený QR kód.

Testovanie funkč. modulu: Výsled.obrazovka



Test pre skupinu funkcií ZQR1
 Funkčný modul ZQR_CREATE_PAY
 Malé/veľké písmo


DobaChodu: 12.854.125 Mikrosekundy

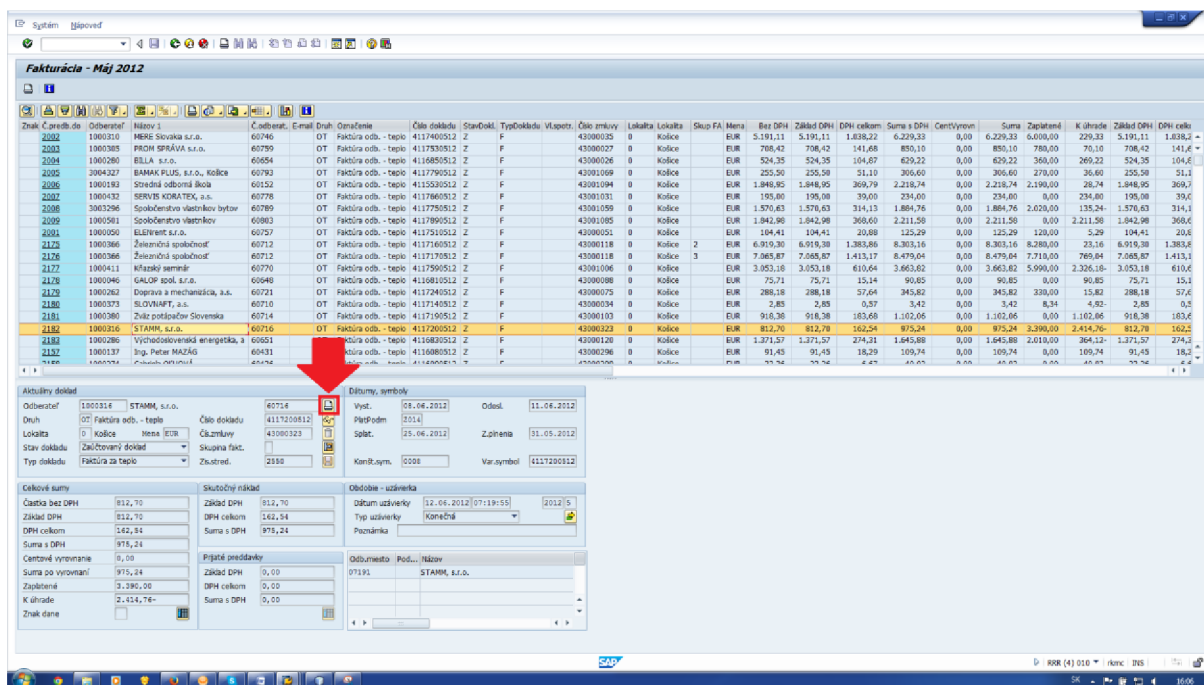
Parametre importu	Hodnota
GS_PAY	
GV_VBELN	ABCD

Parametre exportu	Hodnota
O_BYSQUARE	0004I00032SG115B11J3HGUBL4RFNSJUFIMRGLDGNVDSR9M35VNM77V0CEUV9Q1L9JLDR0UMFJEJIMQUI8QIN404EI9FSH4TUC9D95E8I800

Obrázok 9: Výsledok testovania funkčného modulu


Aplikácia pre fakturáciu s využitím QR kódov

Aplikácia pre fakturáciu za odbyt tepla, ktorá bola rozšírená o QR kódy je spustiteľná transakciou ZODT_013 alebo v menu voľbou *Test generovania QR kódov na faktúre*. Pri prvom spustení fakturácie je potrebné zadať obdobie fakturácie. To sa odporúča zvoliť na April 2012, nakoľko v systéme nie sú aktuálne platné dáta. Prípadná zmena obdobia je možná transakciou ZODT_000. Vstupná obrazovka fakturácie obsahuje údaj o aktuálnom zvolenom období a polia, na základ ktorých je možné obmedziť výber zobrazenia faktúr. Tieto polia môžu ostať nevyplnené. Prehľad faktúr sa zobrazí kliknutím na  alebo stlačením tlačidla F8. Prehľad je rozdelený do dvoch častí. Vrchnú časť tvorí zoznam faktúr a spodnú detail zvolenej faktúry (Obrázok 10).



The screenshot displays the 'Fakturácia - Máj 2012' window. The top section is a table listing various invoices. A red arrow points to a QR code icon in the 'Aktuálny doklad' section of the detailed view below. The detailed view includes fields for 'Objaviteľ', 'Druh', 'Lokalita', 'Stav dokladu', 'Typ dokladu', 'Číslo dokladu', 'Číslo zmluvy', 'Základ DPH', 'Suma s DPH', 'Centrálna vyravnávanie', 'Suma po vyravnání', 'Zaplatené', 'K úhrade', and 'Znak dňa'. It also features a 'Dátum udverejnenia' field and a 'Typ udverejnenia' dropdown menu.

Obrázok 10: Aplikácia na tlač faktúr s QR kódmi

Vytvorenie formulára faktúry je možné kliknutím na tlačidlo  označené v detaile faktúry. Následne sa zobrazí okno, kde je potrebné zadať výstupné zariadenie a zaškrtnúť príznak pre zobrazenie náhľadu. Po potvrdení tejto voľby sa zobrazí náhľad faktúry, ktorá obsahuje QR kód s platobnými údajmi.