



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**APLIKACE PRO GENEROVÁNÍ A TISK ŠTÍTKŮ V
SYSTÉMU SAP**

APPLICATION FOR GENERATING AND PRINTING OF LABELS IN THE SAP SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MAREK KALABZA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VLADIMÍR BARTÍK, Ph.D.

BRNO 2020

Zadání diplomové práce



Student: **Kalabza Marek, Bc.**
Program: Informační technologie a umělá inteligence Specializace: Informační systémy a databáze
Název: **Aplikace pro generování a tisk štítků v systému SAP**
Application for Generating and Printing of Labels in the SAP System
Kategorie: Databáze

Zadání:

1. Seznamte se a nastudujte SAP Adobe Form a SAP Smart Forms a generování tiskových formulářů v systému SAP. Dále prostudujte problematiku a možná řešení tisku čárových kódů 1D a 2D.
2. Seznamte se s komunikací a propojení systému SAP s tiskárnou.
3. Navrhněte způsob implementace generování a tisku různých typů štítků. Využijte k tomu vhodné modelovací techniky.
4. Navržené řešení implementujte.
5. Otestujte vytvořené řešení na vhodném vzorku dat.
6. Zhodnoňte použitelnost a navrhněte možná budoucí vylepšení a další pokračování tohoto projektu.

Literatura:

- Bandari, K.: Complete ABAP. SAP Press, 2016. ISBN: 978-1-4932-1272-9.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Bartík Vladimír, Ing., Ph.D.**

Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 19. května 2021

Datum schválení: 22. října 2020

Abstrakt

Cílem této diplomové práce je návrh a implementace aplikace pro generování a tisk adresních štítků v informačním systému SAP. Práce dále vysvětluje základní pojmy informačního systému SAP a popisuje technologie pro vytváření tiskových formulářů SAP Adobe Form a SAP SmartForm. Rovněž se zabývá i řešením tisku 1D a 2D čárových kódů a následně s nastavením a komunikací systému SAP s tiskárnou. Výsledkem této práce je analýza tvorby formulářů, návrh řešení pro generování formulářů a jejich implementace v systému SAP. V práci jsou konkrétně navržené a implementované tyto formuláře: formulář pro adresní štítky, formulář pro arch 21 adresních štítků, formulář pro podací lístky České pošty a formulář pro podací arch České pošty. Závěrem práce je testování vytvořeného řešení a návrh budoucího rozšíření.

Abstract

The goal of this diploma thesis is the design and implementation of an application for generating and printing address labels in the SAP system. The work also explains the basic concepts of the SAP system and describes the technologies for creating print forms as SAP Adobe Form and SAP SmartForm. Lastly, it describes the solution of printing 1D and 2D barcodes and setting up the communication between the SAP system and the printer. The result of this work is an analysis of form creation, subsequent design of a solution for generating forms as address label, sheet of 21 address labels, Czech Post delivery notes, Czech Post delivery sheet and its implementation in the SAP system. The conclusion of the work is testing the created solution and proposing future improvements.

Klíčová slova

SAP, SAP Adobe Forms, Adobe liveCycle Designer, SAP Smart Forms, Form Builder, NetWeaver, ABAP, ABAP Workbench.

Keywords

SAP, SAP Adobe Forms, Adobe liveCycle Designer, SAP Smart Forms, Form Builder, NetWeaver, ABAP, ABAP Workbench.

Citace

KALABZA, Marek. *Aplikace pro generování a tisk štítků v systému SAP*. Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vladimír Bartík, Ph.D.

Aplikace pro generování a tisk štítků v systému SAP

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Vladimíra Bartíka, Ph.D. a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Marek Kalabza
22. dubna 2021

Poděkování

Chtěl bych poděkovat vedoucímu diplomové práce Ing. Vladimíru Bartíkovi, Ph.D. za ochotu a rady při konzultacích. Dále bych chtěl poděkovat své rodině za podporu po dobu celého studia.

Obsah

1	Úvod	3
2	SAP	4
2.1	Historie	4
2.2	Architektura	5
2.2.1	SAP NetWeaver	6
2.2.2	Vývojové prostředí	6
2.3	ABAP Workbench	7
2.4	Advanced Business Application Programming - ABAP	7
2.4.1	ABAP Dictionary	8
2.4.2	Tabulky	8
2.4.3	Typy	8
2.4.4	Objekty	9
2.4.5	Funkční moduly	9
2.5	SALV	10
3	Analýza	11
3.1	SAP Script	11
3.1.1	Rozložení SAP Scriptu	11
3.1.2	SAP Script editor	13
3.1.3	SAP Script tisk	13
3.2	SAP Smart Form	14
3.3	Form Builder	16
3.4	Formuláře Adobe	16
3.5	Typy Adobe formulářů	17
3.5.1	Interaktivní formuláře	17
3.5.2	Formuláře bez možnosti úpravy	19
3.5.3	Statické formuláře	19
3.6	Adobe LiveCycle	20
3.7	Čárové kódy v systému SAP	21
3.8	Nastavení tisku v systému SAP	22
3.8.1	Založení výstupního zařízení	22
3.8.2	Vytvoření typu zařízení	26
3.9	ESim	26
4	Specifikace požadavků a návrh	27
4.1	Vyhledávací kritéria	27
4.2	Typy formulářů	27

4.2.1	Formulář pro tisk nalepovacího adresního štítku na tiskárně Intermec PC43t	27
4.2.2	Formulář pro tisk 21 nalepovacích adresních štítků	28
4.2.3	Formulář pro tisk podacího archu České pošty	29
4.2.4	Formulář pro tisk podacích lístků České pošty	29
4.3	Funkcionalita	30
4.4	Struktura adres	32
4.5	Objektové řízení tisku	33
4.6	Návrh postupu řízení tisku	34
5	Implementace	36
5.1	Programování v jazyce ABAP	36
5.2	Popis programu	36
5.2.1	Paket	37
5.2.2	Transakce	37
5.3	Výběrová obrazovka	38
5.4	Získávání dat	40
5.5	Dočasné tabulky	42
5.6	Zobrazení dat	42
5.7	PBO a PAI	44
5.8	Potvrzení tisku s počtem kopií	45
5.9	Formuláře	46
5.9.1	Implementace štítků pro tiskárnu Intermec PC43t	46
5.9.2	Implementace formuláře podacího lístku	47
5.9.3	Implementace formuláře pro 21 adresních štítků	48
5.9.4	Implementace formuláře podacího archu	49
5.10	Code inspector	51
5.11	Konfigurace tiskárny Intermec PC43t	52
6	Testování	55
6.1	Komunikace s tiskárnou	55
6.2	Přenos na testovací systém	56
6.3	Testování na datech	56
6.4	Chyba tisku během testování	57
6.5	Časová náročnost	57
6.6	Finální podoba tisku	58
7	Závěr	63
7.1	Zhodnocení	63
7.2	Možný budoucí vývoj	63
	Literatura	64
A	Obsah přiloženého CD	66

Kapitola 1

Úvod

V současné době velké i střední firmy pracují s obrovským množstvím dat informací, jež pak následně zpracovávají. K tomuto účelu využívají podnikové systémy. Systém pro řízení podniku zajišťuje úspěšný chod firmy, proto jsou tyto firmy na výše uvedených systémech zcela závislé. Podnikové systémy spadají do kategorie informačních systému, které mají tu vlastnost, že zrychlují předávání informačních dat a jejich zpracování. Kromě zrychlování předávání informací nám dovolují efektivně přizpůsobovat různé požadavky jak zákazníků, tak i dodavatelů. Podnikový systém většinou zastřešuje celý systém k řízení podniku. A to od plánování projektů, finančního účetnictví až po řízení a správu zaměstnanců. Podnikové systémy vyvíjí spousta velkých firem. Firmy bez podnikového systému můžou právě využít software od těchto firem nebo si můžou nechat vytvořit svůj vlastní podnikový systém. Mezi největší hráče patří podnikový systém od společnosti SAP, jenž má velkou výhodu v tom, že se jedná o zavedenou firmu s produkty, které jsou již dlouhá léta na trhu a produkty jsou tak stále vylepšovány. Společnost SAP a jejich produkty jsou celosvětově používané a také jsou celosvětově podporované různými firmami, jež se zabývají jejich implementací. Kromě toho taky společnost SAP spolupracuje s různými dalšími velkými firmami. Například s firmou Adobe, která umožňuje integraci s jejich typy formulářů. Systém společnosti SAP je navržen tak, aby zvyšoval produktivitu, poskytoval mnohem lepší správu zásob a materiálů, podporoval kvalitu, snižoval náklady na suroviny a především zvyšoval zisky firmy.

Ve své práci jsem se zaměřil na proces generování formulářů a štítků v systému SAP a jejich následný tisk v tiskárně. Tímto chci docílit částečné automatizace, ta usnadní práci ekonomickým pracovníkům, a to převážně z časového hlediska. Konkrétně se jedná o automatické doplňování adres z podnikového systému přímo do formulářů (štítků) a jejich následný tisk ve vybraných počtech kopií. Kromě toho se tato práce dále zaobírá doplňováním adres z podnikového systému do formulářů podacích lístků České pošty a podacího archu České pošty. Po vygenerování formulářů bude možné provádět tisk na lokálních tiskárnách. Má práce se textově skládá ze sedmi hlavních kapitol. V kapitole 2 se popisují základní informace o firmě SAP a systému SAP. Dále se zde popisuje architektura systému a vývojové prostředí, včetně programovacího jazyku ABAP. V kapitole 3 se nachází seznámení s různými typy formulářů a možnosti tisku 1D a 2D čárových kódů v systému SAP, včetně nastavování připojení k tiskárně. Kapitola 4 popisuje návrh aplikace z pohledu dat a návrh jednotlivých formulářů. V další kapitole 5 je popsána implementace aplikace, napojení tiskárny a napojení podnikových dat s jednotlivými formuláři. V předposlední kapitole 6 se popisuje testování a správnost tisku formulářů a štítků. V poslední kapitole 7 je závěrečná část se zhodnocením práce. Poslední kapitola zahrnuje zároveň návrh pro potenciální budoucí rozšíření této aplikace.

Kapitola 2

SAP

Společnost SAP[18] je německá firma, která má sídlo ve Walldorfu a patří mezi největší softwarové giganty na světě. Název zkratky SAP vznikl ze tří německých slovních spojení. A to konkrétně Systeme, Anwendungen a Produkte in der Datenverarbeitung. V překladu do češtiny jsou to tyto slovní fráze Systémy, Aplikace a Produkty na zpracování dat.

2.1 Historie

V Německu v roce 1972 mělo pět bývalých zaměstnanců IBM (Dietmar Hopp, Hasso Plattner, Hans-Werner Hector, Klaus Tschira a Claus Wellenreuther) vizi v potenciálu obchodních technologií.

Jejich myšlenkou bylo vytvořit standardní podnikový software, který by integroval všechny obchodní procesy a umožňoval zpracování dat v reálném čase. Zakladatelé a zaměstnanci společnosti SAP úzce spolupracovali se zákazníky - často seděli bok po boku se zaměstnanci v kancelářích zákazníků a učili se jejich obchodní potřeby a procesy. Koncem roku 1975 vytvořili aplikace pro finanční účetnictví, ověřování faktur a správu zásob. Mezi jejich první zákazníky patřila nylonová továrna ICI v německém Östringenu, Knoll, Burda, Linde a Schott. Kombinace zpracování, standardizace a integrace dat v reálném čase byly základem pro transformaci společnosti SAP z malé německé společnosti na světového lídra v obchodním softwaru. V roce 1979 společnost začala vyvíjet systém SAP R/2, který je druhou generací SAP softwaru. V roce 1980 se zaměstnanci společnosti SAP přestěhovali do své první vlastní kancelářské budovy v německém Walldorfu.

Přestože systém SAP R/2 měl obrovský prodejní úspěch, tak se po vstupu na burzu v roce 1988, vedoucí manažeři společnosti se poohlíželi už po třetí generaci tohoto softwaru. Příběh úspěchu SAP R/3 začal až v roce 1992, kdy software klient-server odkryl cestu ke globalizované ekonomice a proměnil SAP na globálního hráče s pobočkami a vývojovými centry po celém světě. V roce 1999 společnost SAP musela zareagovat na příchod internetu a novou ekonomiku. A tak spustili službu mySAP. O deset let později se společnost rozdělila na tři trhy budoucnosti: mobilní technologie, databázové technologie a cloud.

V roce 2011 začali první zákazníci používat in-memory databázi SAP HANA. Analýzy dat, které dříve trvaly dny nebo dokonce týdny, tak byly nyní dokončeny během několika sekund. O čtyři roky později společnost SAP uvedla na trh SAP S/4HANA, což je nejnovější generace obchodního softwaru, která běží zcela na SAP HANA. V roce 2019 společnost SAP získala americkou společnost Qualtrics, přední společnost v oblasti softwaru pro správu zkušeností zaměstnanců a zákazníků, čímž se společnost SAP stala špičkou v tomto rostoucím

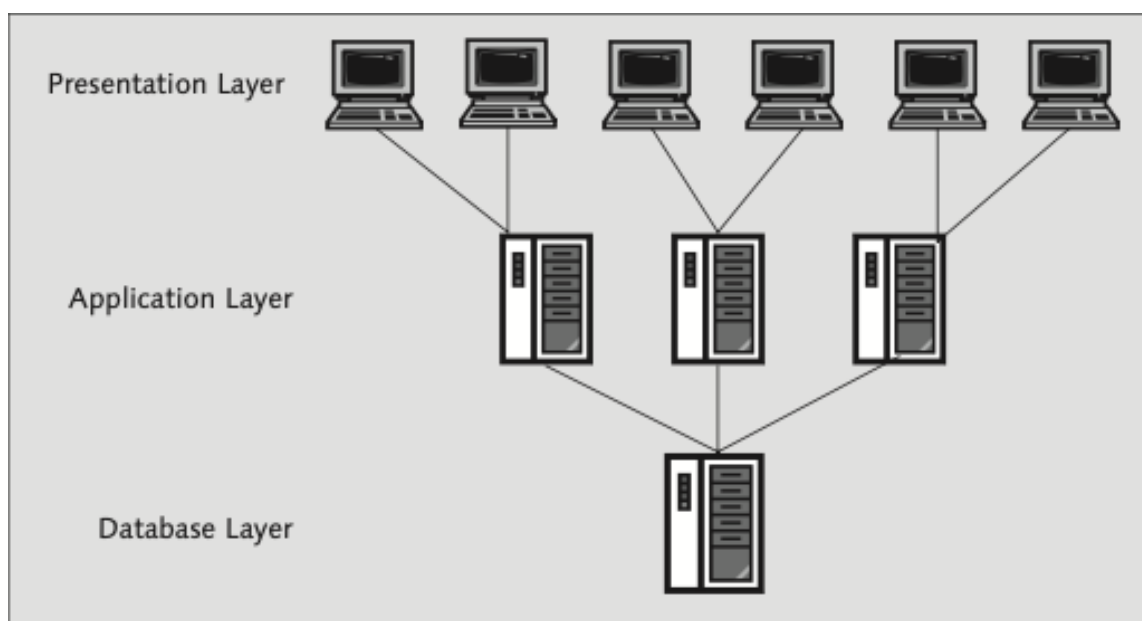
segmentu. Dnes společnost SAP vytváří řešení pro IoT a umělou inteligenci pro komplexní analýzy a blockchain. SAP HANA je nyní k dispozici na třech největších platformách veřejného cloudu. Integrované aplikace společnosti SAP nyní propojují všechny části podniku do inteligentní sady na digitální platformě.

SAP momentálně využívá přes 79 000 zákazníků. V této kapitole jsem čerpal z těchto zdrojů [16] a [17].

2.2 Architektura

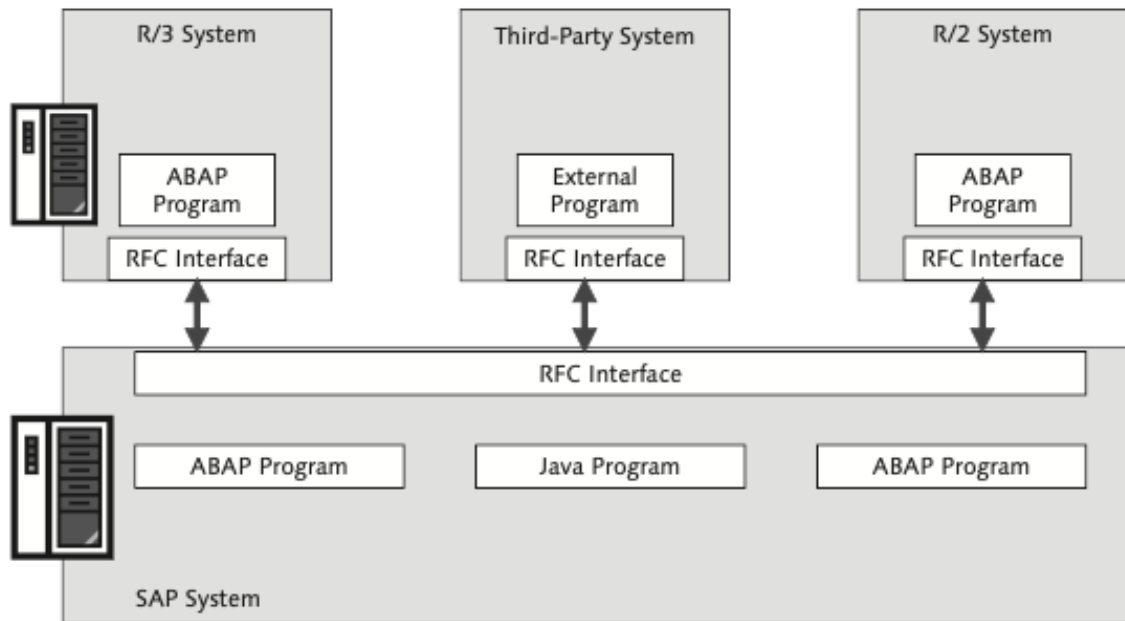
Architektura SAP běží na principu třívrstvé struktury klient-server:

1. prezentační vrstva
2. aplikační vrstva
3. databázová vrstva



Obrázek 2.1: Přehled SAP architektury[9].

V této fázi stačí mít pouze základní představu o architektuře klientského serveru. Tato architektura tedy představuje stejný softwarový pohled na architekturu SAP. Zprovoznění SAP systému se skládá z databázového serveru, obvykle na počítači nebo systému založeném na UNIXu a mnoho serverů zajišťující prezentační vrstvu. Databázový server je určen k poskytování služeb údržby a přístupu k datům. Programy se načítají do aplikačních serverů a spouštějí se v paměti RAM aplikačních serverů. Prezentační servery slouží k přístupu, provozu, interakci a dialogu uživatelů[13].



Obrázek 2.2: Přehled architektury SAP NetWeaver[9].

2.2.1 SAP NetWeaver

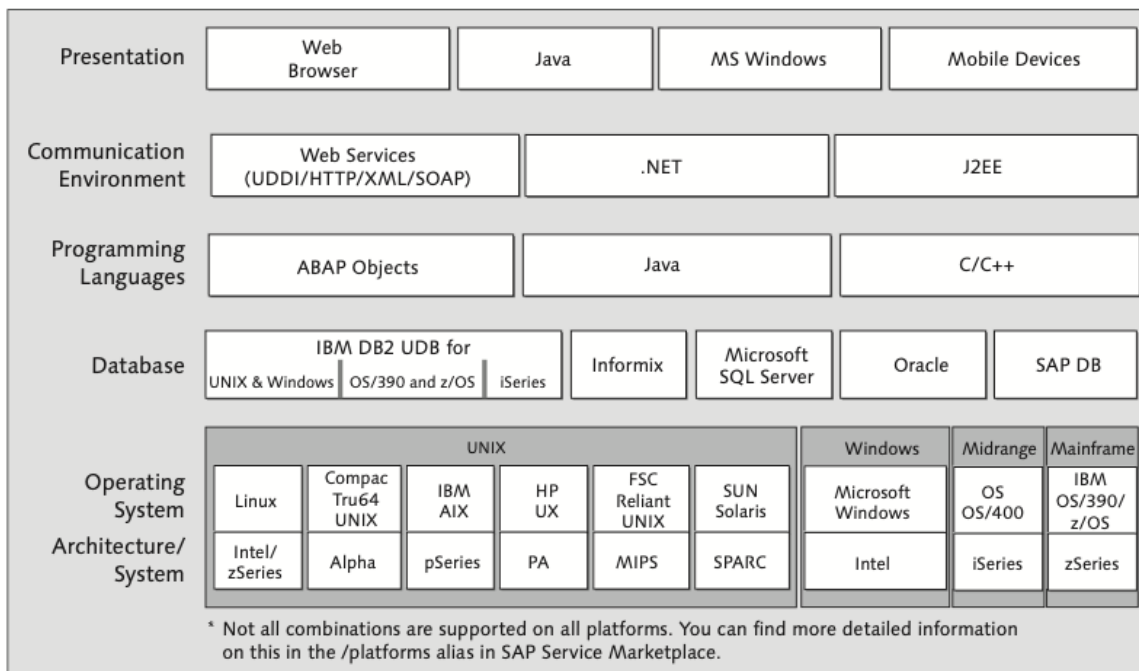
SAP NetWeaver je integrační framework, který byl navržen pro provoz servisně-orientované architektury (SOA). Kromě toho byl také navržen jako kooperativní technologie, která integruje funkční moduly SAP a také poskytuje připojení k externím systémům. NetWeaver je nástupcem systému SAP R/3. SAP NetWeaver obsahuje tedy komponenty, nástroje a aplikace. Aplikační server je součástí aplikace SAP NetWeaver. Samotný aplikační server NetWeaver se skládá z aplikačního serveru ABAP (AS ABAP) a aplikačního serveru Java (AS JAVA). Mezi komponenty SAP NetWeaver patří: obchodní sklad, správa obchodních procesů, správa obchodních pravidel, integrace procesů, správa hlavních dat, mobilní zařízení, portál, infrastruktura Auto-ID, správa identit a správa životního cyklu informací[13].

2.2.2 Vývojové prostředí

V prostředí SAP se informační systémy z podnikového pohledu dělí na tři typy:

1. Vývojový systém
2. Testovací systém
3. Produkční systém

Každý z těchto systémů je soběstačně funkční a nezávislý. Nezávislý, protože neobsahuje data závislá s jiným systémem. Kromě oddělených dat má taky každý systém své uživatelské přístupy, aby nedocházelo k neoprávněnému vstupu do produkčního systému. Jednotlivé systémy už podle názvu napovídají k čemu slouží. Vývojový systém je pouze určený na vývoj aplikací, které se poté mohou transportovat pomocí transportu do testovacího systému. Transport je obálka, která zaobaluje vytvořené části nebo změny projektu. Každý projekt musí být přidělený k nějakému transportu, pokud chce být transportován na jiný systém.



Obrázek 2.3: Přehled technologií podporující SAP[9].

Transport si může vytvářet každý uživatel sám. Tyhle transporty je pak možné uvolnit (tzn. potvrdit k přenosu, a tím je uzamknout) a následně odeslat na testovací systém. Testovací systém má už relativně reálnější data k testování a slouží výhradně k otestování funkčnosti jednotlivých částí aplikací nebo projektů. Kdežto ve vývojovém systému lze provádět úpravy kódu, tak v testovacím systému nelze provádět změny kódu jednoduše a je třeba provést úpravu pomocí nového transportu. Jakmile je kód řádně otestován, lze ho opět pomocí transportu transportovat do produkčního systému, který již obsahuje reálná data.

2.3 ABAP Workbench

ABAP Workbench je grafické prostředí v ABAP systému, které je přístupné pomocí SAP GUI a umožňuje vyvíjet aplikace v programovacím jazyku ABAP. Je to jedna z hlavních částí produktu SAP bussiness applications. SAP ABAP Workbench poskytuje mnoho usnadňujících nástrojů pro vytváření programů a zajišťuje podporu pro vytváření různých datových typů, struktur, objektů a obrazovek[7].

2.4 Advanced Business Application Programming - ABAP

ABAP je programovací jazyk určený pro speciální úkoly, který má být podobný přirozené lidské řeči. Tento jazyk vývojáři původně používali k vývoji platformy SAP R/3. Jazyk byl původně orientován pro technicky zdatné zákazníky se zkušenostmi s programováním. ABAP má nyní abstrakci mezi obchodními aplikacemi, operačním systémem a databází. Touto vlastností jazyka je zajištěno, že aplikace nezávisí přímo na konkrétním serveru nebo databázové platformě a lze je snadno přenést z jedné platformy na druhou. Tím se stal jazyk ABAP nezávislým prostředníkem komunikace.

2.4.1 ABAP Dictionary

ABAP Dictionary[10] se také nazývá datový slovník. Lze jej najít pod anglickou zkratkou DDIC nebo někdy jen jako slovník. Tento slovník slouží jako systémové úložiště pro celý systém, kde jsou vytvářeny a udržovány databázové objekty. Konkrétně se jedná o úložiště objektů jako jsou domény, datové prvky a transparentní tabulky. Každý program v systému SAP se dotazuje, zda slovník ABAP obsahuje definici objektu, aby se zajistilo, že všechny strany pracují s jedinou definicí objektu a nenastaly komplikace[15]. V ABAP Dictionary lze definovat tyto objektové typy:

- Tabulky
- Pohledy
- Typy
- Domény
- Nápoředy k vyhledávání
- Uzamykání objektů

2.4.2 Tabulky

Tabulky jsou kolekce tabulkových polí, které jsou organizované pomocí vertikálních sloupců a horizontálních řádků. V ABAP Dictionary jsou tabulky definované nezávisle na typu databáze. Pole tabulek jsou definované pomocí vytvořených datových typů nebo pomocí předdefinovaných datových typů. Existují následující tři kategorie databázových tabulek v ABAP Data Dictionary:

- Transparentní tabulky
- Sdílené tabulky
- Clusterové tabulky

Clusterové a sdílené tabulky slouží převážně k ukládání Dynpro sekvencí, dokumentů a nastavení systému. Tyhle dva typy databázových tabulek fungují na principu, že se skládají z mnoha jiných dat tabulek, které jsou právě uloženy ve sdílené tabulce nebo v clusterové tabulce. Obecně řečeno v clusterové tabulce se nachází malé množství velkých a rozsáhlých tabulek, kdežto ve sdílené databázové tabulce se nachází naopak velké množství tabulek s menším rozsahem dat. Ani jedna ze dvou zmíněných tabulek nekopíruje podobu fyzické databáze. Přesnou kopii fyzické databáze vyobrazují pouze transparentní databázové tabulky. Data samotných aplikací bývají většinou ukládána právě do transparentních databázových tabulek.

2.4.3 Typy

Datové typy se využívají v ABAP programech a lze je definovat v ABAP Dictionary. Konkrétně lze definovat tyto datové typy:

- Struktura
- Datový prvek
- Typ tabulky

Struktura

Struktury jsou komplexní typy, které se skládají z jednotlivých komponent, jež mohou být jiného datového typu. Komponenta může být pole ze základních typů, referenčního typu nebo opět struktury.

Datový prvek

Datový prvek je jasně daný jeden typ, který je vytvořen na základě standardního datového typu, jako je číslo, řetězec, čas nebo datum. Datový prvek může být využit jako typ pole tabulky, nebo jako typ proměnné v programu.

Typ tabulky

Typ tabulky je komplexní typ. Tento typ je vytvořen na základě struktury nebo datových prvků, který právě definuje jednotlivé pole tabulky řádku. Typ tabulek se využívá převážně pro vytváření interních tabulek, jež se definují lokálně. Tento typ lze použít pouze lokálně.

2.4.4 Objekty

ABAP Objects je objektově orientovaná část ABAP. Je to rozšíření o kompletní sadu jazykových prvků, které umožňují objektově orientované programování. Toto objektově orientované rozšíření ABAP navazuje na předchozí řadu jazyků a je s nimi plně kompatibilní. Obecně objektově orientované programování umožňuje vytvářet třídy, využívat polymorfismus a dědičnost tříd. Definování tříd lze provádět s příslušným integrovaným grafickým rozhraním[18].

2.4.5 Funkční moduly

Jsou to funkce, které se ve světě SAP nazývají funkční moduly. Funkční modul je schopen vrátit několik výsledků najednou. A to typu hodnota, tabulka, reference nebo struktura. Tyto funkce se vytvářejí pomocí uživatelského rozhraní v rámci nástroje SAP Object Navigator[7], který automaticky vytváří strukturu zdrojového kódu. Název funkce, který musí být v celém systému jedinečný, spolu se všemi parametry rozhraní a návratovými hodnotami pro chyby a výjimkami se zadávají prostřednictvím dialogů. Po zadání dialogu se následně objeví všechny parametry ve zdrojovém kódu a stačí je doplnit. Pro každou funkci mohou programátoři vytvořit samostatnou dokumentaci, jež může odkazovat i na jiné dokumenty. Při vytvoření funkce se funkce automaticky zařadí do pracovní fronty pro překlad dokumentace funkce. Funkce lze také velmi snadno zprovoznit pro vzdálené volání a lze je tedy volat z jiných systémů SAP a to ze všech aktuálních programovacích jazyků a operačních systémů, jako je Java, Windows nebo UNIX. Funkce jsou vždy seskupeny do funkčních skupin. V těchto funkčních skupinách můžou jednotlivé funkce přistupovat ke stejným globálním proměnnám[9].

2.5 SALV

ALV (ABAP List Viewer) je v podstatě chytřejší prohlížeč seznamu dat v podobě tabulky, která je jako sada součástí aplikačního rozhraní pro zobrazování velkého množství dat. ALV lze zobrazit buď v tabulkovém nebo hierarchickém formátu. Kromě toho umožňuje upravování vizuální prezentace a zpracovávání jednotlivých událostí. ALV je původní zkratka, která se stále používá, ale dnes ji najdeme v dokumentaci pod pojmem SAP List Viewer. V praxi se tedy stále označuje jako ALV. Kromě zmiňovaných možností ALV také nabízí uživatelům velmi přátelské rozhraní, které je navíc díky panelu nástrojů snadno upravitelné. Tento panel umožňuje uživatelům různě přizpůsobit prezentované rozložení podle jejich preferencí. Dále panel umožňuje snadno data filtrovat nebo třídít. Tato rozložení lze uživatelsky ukládat a tak každý uživatel může mít vlastní rozložení dle svých představ při každém spuštění aplikace.

SALV bylo strategicky zavedeno při implementaci objektově orientovaného programování ABAP v SAP ERP verze 4.6C. SALV je tedy první objektově orientovaný rámec ALV. Z vizuálního pohledu se nic nezměnilo. Implementace založená na třídách přinesla výsledky velmi podobné funkčnímu modulu. Přesto SALV přineslo jednu zásadní praktickou novinku. Umožňuje objekt ALV napojit na vlastní kontejner na obrazovce. Na základě této vymoženosti lze na jedné obrazovce prezentovat nejen jednu, ale i vícero ALV kontejnerů. Současná implementace ALV frameworku se tedy nazývá SALV a je k dispozici v SAP NetWeaver 7.0 a vyšší verzi. Tento framework je k dispozici v třídách, které začínají s názvem CL_SALV. Kromě klasické ALV tabulky podporuje taky ALV strom a hierarchicko-sekvenční seznam[5].

Kapitola 3

Analýza

Tato kapitola popisuje podrobnější analýzu možností tisku a generování formulářů. Vysvětlují se zde rozdíly mezi různými typy formulářů v systému SAP. V závěru kapitoly jsou vysvětleny důležité pojmy této práce.

3.1 SAP Script

Skript SAP[2] je individuální systém pro zpracování textu v systému SAP. Vypadá podobně jako jiné přední systémy pro zpracování textu. Používá se k tisku předformátovaného textu v předformátovaných formách. SAP Script se skládá z 5 hlavních komponent:

- Editor
- Programové rozhraní
- Nastavení stylů a rozvržení pro tisk
- Centrální výstupní modul
- Databázové tabulky pro ukládání textů, stylů a sad rozvržení

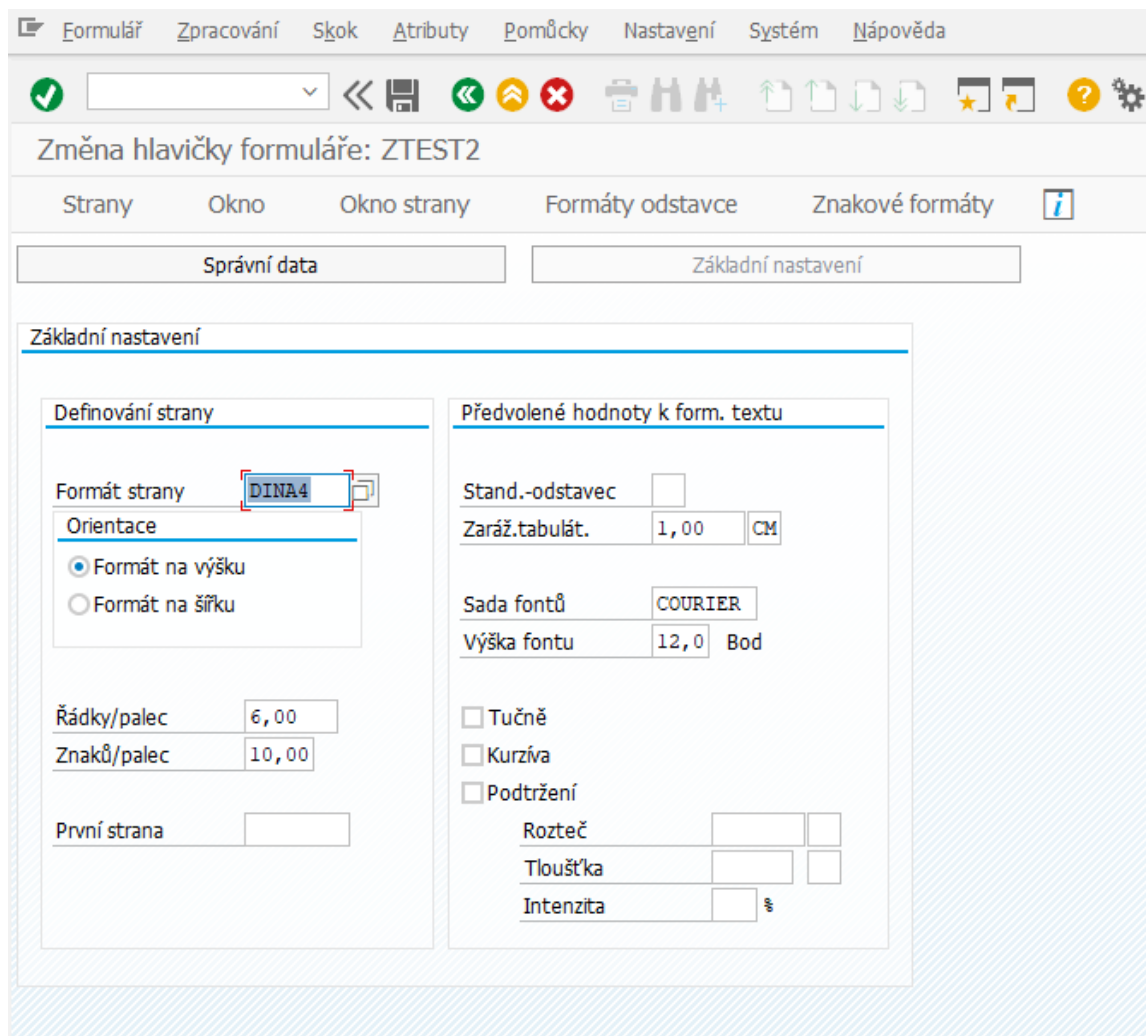
K výstupu dokumentů pomocí programovacího rozhraní potřebují aplikační programy R/3 takzvané sady rozvržení (druh formuláře). Ve SAP Scriptu nám grafické rozvržení popisuje samotné rozvržení jednotlivých tiskových stránek a pomocí textových prvků dodává definovatelné výstupní bloky, které může tiskový program volat. Grafické rozvržení může být taky definováno jako návrh stránky dokumentu. Samotné grafické rozvržení neobsahuje žádná data. Výběr dat pro dokument se provádí prostřednictvím uživatelem vytvořeného tiskového programu v systému SAP. Tiskový program vybírá data z databázové tabulky a dodává je do vytvořeného grafického rozvržení.

3.1.1 Rozložení SAP Scriptu

Rozložení SAP Scriptu se sestavuje z následujících komponent.

Data záhlaví

Data záhlaví se používají pro nastavení základních informací a nastavení řízení tisku. Definiuje se zde formát strany, orientace strany a úvodní strana. Dále se zde definují předvolené hodnoty k formátovému textu jako je výchozí odstavec, jazykové atributy a další.



Obrázek 3.1: Prostředí Sap Script po spuštění.

Formáty odstavce

Odstavce jsou formátovány pomocí různých atributů. Například standardní atributy odstavce určují levý nebo pravý okraj, zarovnání, řádkování a další atributy. Atributy písma určují font písma, velikost písma a další vlastnosti. Formáty znaků umožňují formátovat celé bloky textu v odstavcích.

Okna

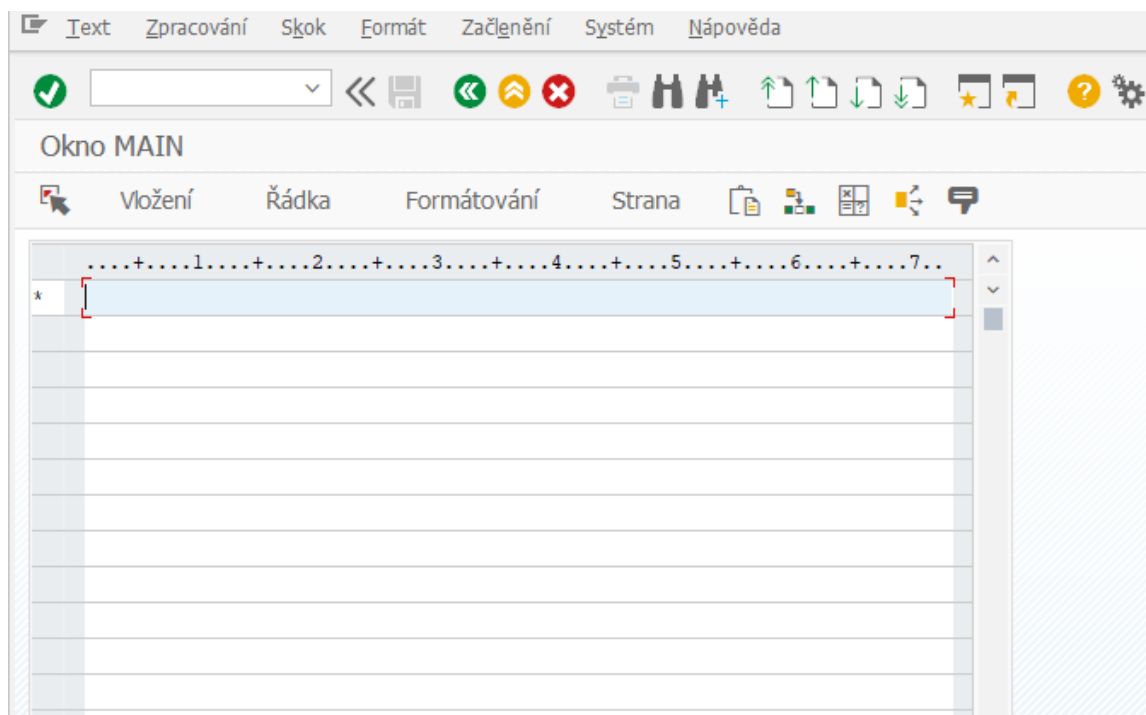
Okna strany jsou jednotlivé textové oblasti (adresa záhlaví, datum, zápatí) na stránce. Tyto okna nám pomáhají kombinovat informace obsažené v dokumentu do určitých skupin a zajistit, aby se každá skupina objevila na vytištěné stránce v samostatné oblasti. Pro každé okno můžeme definovat textové prvky. Tiskový program přistupuje k textovým prvkům podle názvu prvku. Následně je SAP Script formátuje a tiskne v příslušném okně. Ve formuláři musí být definován odstavec a použitý formát znaků.

Strany

Jsou to jednotlivé stránky dokumentu a musí mít jedinečný název. Často najdete různé grafické rozvržení pro jednotlivé stránky, kde první stránka faktury se liší od následujících stránek. Na těchto stránkách nemusíte opakovat obecné informace. Mezi ně patří například adresa nebo informace ohledně zákazníka.

Okna stran

Při definování oken a stránek jsme si neurčili polohu ani rozestup textů. A tyhle parametry nám právě určují vizualizaci, jak mají být vytisknuty. Proto existuje tahle záložka okna stránky, která definuje obdélníkovou oblast výstupu na výstupním médiu. A to prostřednictvím definování levého horního okraje oblasti výstupu s jeho velikostí šířky a výšky.



Obrázek 3.2: Editor Sap Script.

3.1.2 SAP Script editor

V SAP Script editoru se využívají speciální řídicí příkazy, které umožňují řídit výstupní formát. Tyto příkazy nejsou interpretovány editorem skriptů SAP, ale jsou předávány ke zpracování skriptu SAP. To zahrnuje například formátování stránek a řádků, formátování textu podle zadaných formátů odstavců a znaků.

3.1.3 SAP Script tisk

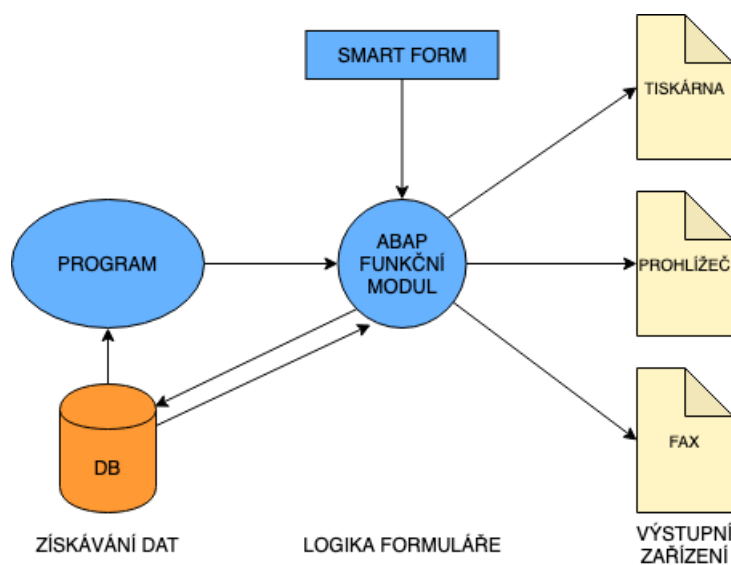
Provádění skriptu se provádí prostřednictvím programu ABAP, jenž se označuje jako tiskový program. Každý tiskový program by měl mít vlastní SAP Script formulář, který bude volán s parametry polí.

Tiskový program potřebuje k provedení skriptu buď všechny, nebo některé z následujících funkcí pro řízení zpracování formuláře:

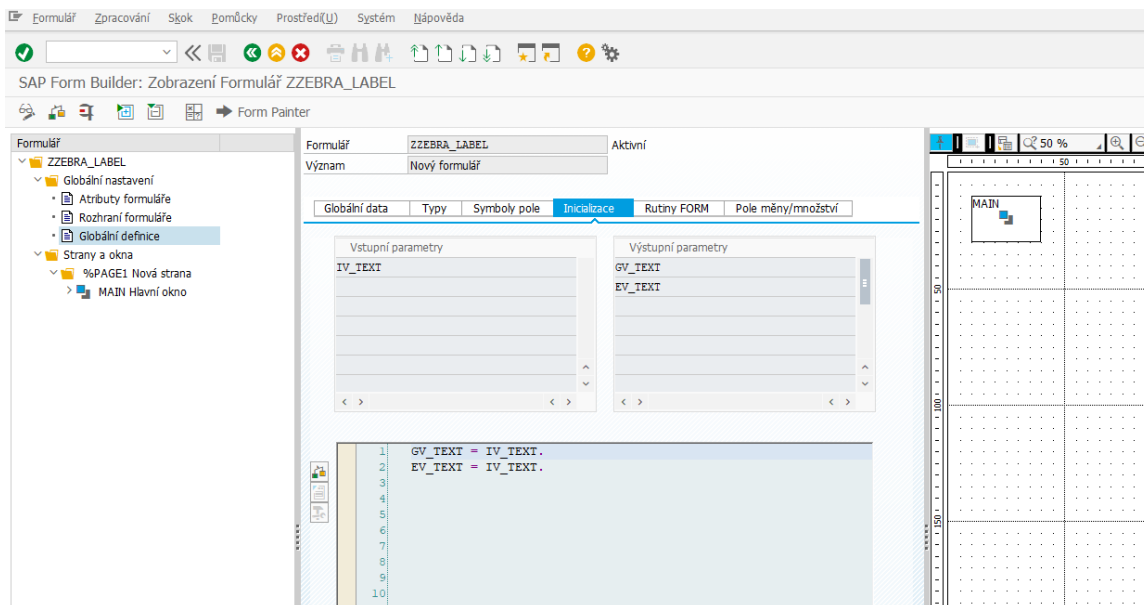
- OPEN_FORM - Otevře výstup sady rozvržení
- START_FORM - Spustí novou sadu rozvržení
- CLOSE_FORM - Ukončí výstup sady rozvržení
- WRITE_FORM - Volá prvek sady rozvržení
- END_FORM - Ukončí aktuální sadu rozvržení

3.2 SAP Smart Form

SAP Smart Forms[4] se využívá k vytváření a údržbě formulářů pro hromadný tisk v systémech SAP, který je již integrovaný v samotném systému. Pomáhá přizpůsobovat formuláře bez znalosti programování a to díky zcela grafickému uživatelskému rozhraní. Podporuje jako výstupní média tiskárnu, fax, e-mail nebo internet. Realizace probíhá pomocí generovaného výstupu XML. K SAP Smart Form vývojovému prostředí se dostaneme pomocí transakčního kódu SMARTFORMS. SAP Smart Forms byly zavedeny společností SAP za účelem zdokonalení schopností a klíčových funkcí nad skripty SAP. V SAP Smart Form je potřeba jen minimální tvorba kódování, když ho porovnáme se skripty SAP. K provedení jakýchkoli změn stačí použít Drag and Drop, Cut a Paste operace. Mezi tyto akce nepatří psaní kódovacích řádků ani používání skriptovacího jazyka. Kromě toho SAP Smart Form má snadnější vývoj a údržbu formulářů. Snadnější je přenos mezi systémy SAP, kde je možnost stáhnout a importovat tyhle formuláře. V porovnání se SAP Scripty lze ve SAP Smart Form definovat více formátů stránek a není třeba mít úvodní stránku. Kromě toho umožňuje vložit statické a dynamické tabulky, díky tomu je zcela výhodnější a použitelnější. SAP Smart Form si po aktivaci formuláře sám generuje funkční modul, který slouží pro generování výsledného formuláře.



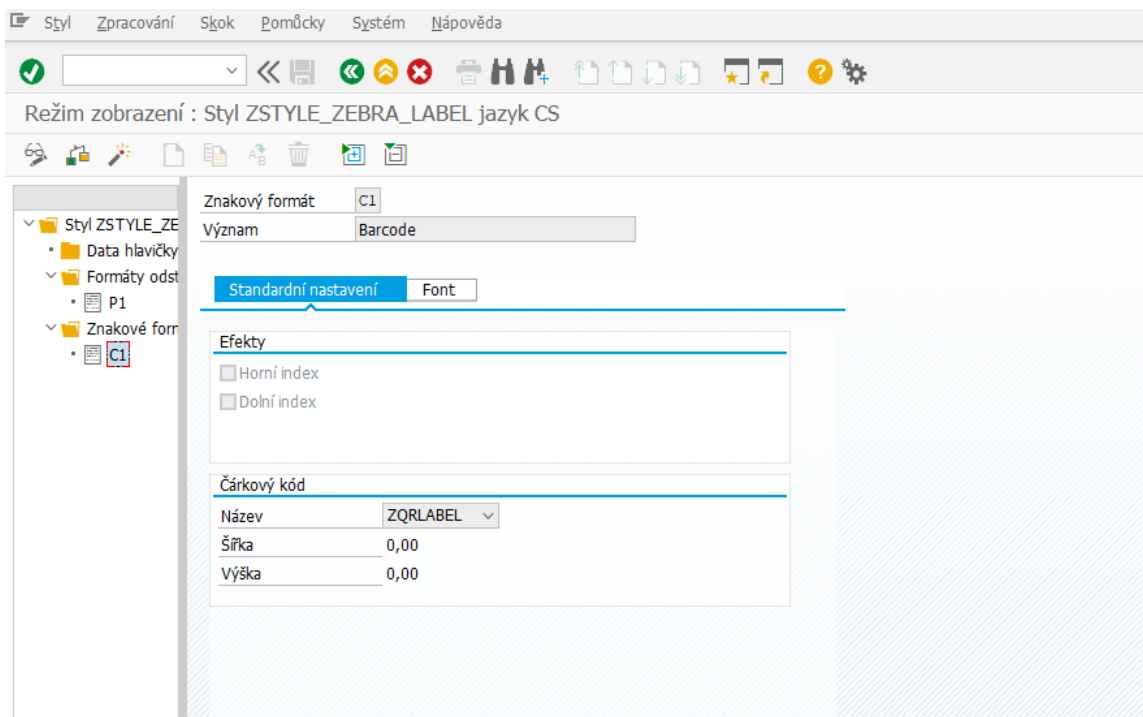
Obrázek 3.3: Architektura SAP Smart Form.



Obrázek 3.4: Prostředí SAP Smart Form.

SAP Smart Form vývojové prostředí je vizuálně odděleno na tři části. Na levé straně se nachází navigační menu, které umožňuje přepínat mezi jednotlivými nastaveními. Toto nastavení se nachází v prostřední části vývojového prostředí. Na pravé straně se zobrazuje vizuální návrh rozložení prvků formuláře, které lze snadno upravovat pomocí kurzoru. Navigační levá část se dále dělí na dvě hlavní sekce. První sekcí je globální nastavení. Kde je globální deklarace dat, jež slouží k definici dat, které jsou zapotřebí, případně jejich úprav pomocí programových rutin. Programová rutina je logika konkrétního formuláře, která slouží k doplnění nebo úpravě dat. V rámci těchto rutin formuláře nemůžeme přistupovat k žádným globálním datům, pokud je výslovně nepředáme rozhraním formuláře. Dále se zde nachází rozhraní formuláře, kde jsou definována všechna data, která budou předána do SAP Smart Formu z tiskového programu. V této části je možnost vytvořit si nové atributy formuláře. Druhou sekcí je vytváření stran a oken, včetně různých prvků formuláře.

Kromě vytváření samotné podoby formuláře a napojování dat z databáze je zapotřebí si definovat vlastní formáty odstavců a znaků. K tomuto účelu slouží styly SAP Smart Formu. Je to oddělená část vývojového prostředí SAP Smart Forms, kde se právě definují tyto formáty, jež se pak využijí v samotných formulářích. V této části lze definovat například typy pro vytvořené čárové kódy.



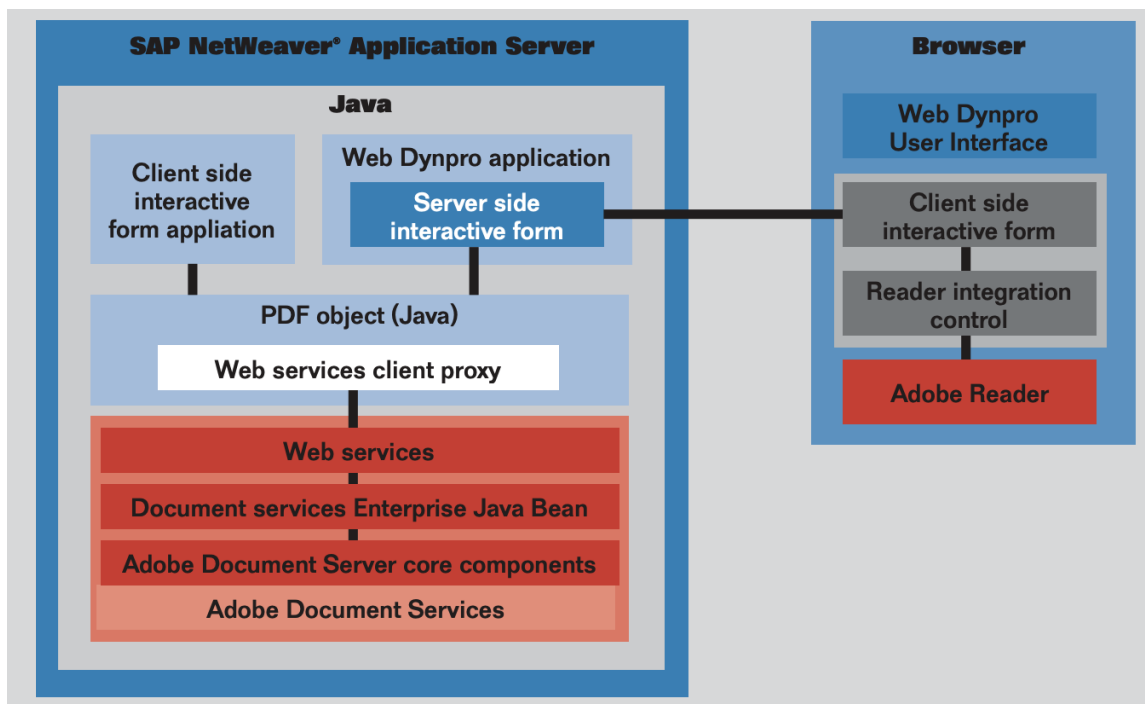
Obrázek 3.5: SAP Smart Form prostředí pro styly.

3.3 Form Builder

Form Builder je prostředí v ABAP Workbench, které sdružuje všechny vývojové nástroje potřebné pro použití SAP interaktivních formulářů od společnosti Adobe.

3.4 Formuláře Adobe

Prostředí SAP nabízí integrování Adobe interaktivních formulářů a to díky spolupráce firmy SAP se společností Adobe. Interaktivní formuláře založené na softwaru Adobe jsou k dispozici v mySAP ERP a SAP NetWeaver. K vývojovému prostředí Adobe formulářů se dostaneme pomocí transakčního kódu SFP. Interaktivní formuláře pomáhají organizaci nebo firmě odstraňovat vrstvy časově náročné byrokracie spadající do obchodních procesů a to pomocí automatizovaných procesů. Zaměstnanci mohou generovat vlastní formuláře přímo z aplikací SAP. S tímto řešením lze vytvářet procesy, které překonávají transakční systémy a eliminují nejtěžší aspekty tradičního ručního zpracování dokumentů. Protože interaktivní formuláře Adobe automatizují procesy založené na formulářích, a tak integrují formuláře a jejich data s podnikovými systémy, jsou tak schopné eliminovat nadbytečné a chybné ruční zadávání dat. Z hlediska uživatele je Adobe formulář dokument. Tento dokument lze upravovat nebo pouze zobrazit. Návrh a vzhled designu formuláře se vytváří prostřednictvím grafického vývojového prostředí Adobe LiveCycle [11]. Ve formuláři je nutné definovat kontext (datovou strukturu), který slouží následně k napojení konkrétních dat do příslušných prvků formuláře.



Obrázek 3.6: Architektura SAP interaktivních formulářů[20].

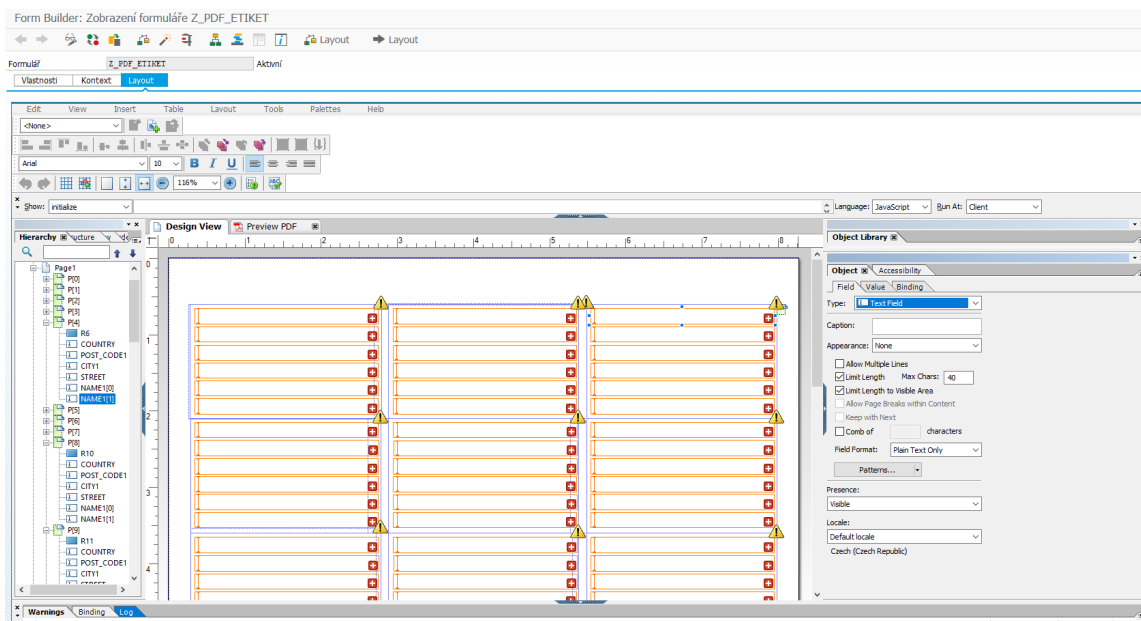
3.5 Typy Adobe formulářů

3.5.1 Interaktivní formuláře

Dokumenty PDF poskytují mnohem víc než jen známou funkci zobrazení a tisku. Mohou obsahovat interaktivní pole formuláře. Tato pole se nazývají interaktivní, protože uživatel může provádět změny pomocí myši nebo klávesnice. SAP interaktivní formuláře od Adobe nabízí vytvoření šablon jednotlivých formulářů, které obsahují loga nebo obrázky. Konkrétně generuje dokumenty způsobem, že kompletuje šablony formulářů s aktuálními systémovými daty. Tyto formuláře lze předem automaticky vyplnit konkrétními údaji z aplikací SAP a poté je pomocí bezpečných metod odeslat správným příjemcům. Interaktivní formuláře umí automaticky kontrolovat konzistenci formulářů a pracovat s digitálními podpisy a certifikacemi formulářů. Pro tento typ formulářů se používá technologie založená na XFA. Šablona formuláře je uložena ve formátu XML a data obsažená ve formuláři PDF jsou uložena jako datový záznam formátu XML.

Pro použití interaktivních formulářů v obchodních procesech existují dva různé typy scénářů. Online scénáře a offline scénáře. V tomto případě koncepty online a offline neoznačují, zda mají uživatelé přístup k síti, ale zda jsou uživatelé přihlášení k systému SAP nebo ne.

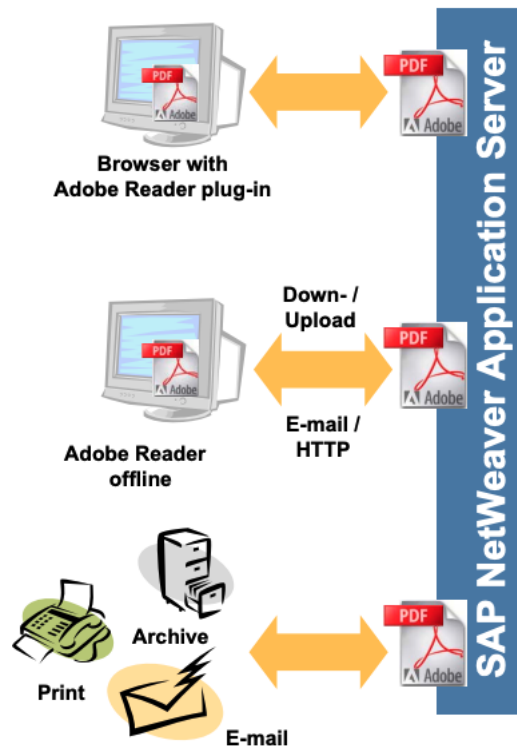
Online scénáře obsahují následující dvě hlavní charakteristiky. První je mít přímý přístup do systému SAP a umožnit se připojit do tohoto systému. To znamená, že během používání formuláře PDF je možné volat systém SAP. Myslí se tím například provádět ověřování, implementovat náповědu s hodnotami nebo provádět složité výpočty. To vyžaduje, aby byl uživatel v systému identifikován a zároveň byl přihlášen v systému SAP. Druhým požadavkem je potřeba zajistit integraci s dalšími technologiemi uživatelského rozhraní. Formuláře PDF se nikdy nezobrazují samostatně, protože jsou zobrazeny ve webovém pro-



Obrázek 3.7: Adobe interaktivní formulář.

hlížeči. A tento prohlížeč využívá jinou webovou technologii, jež volá Adobe Reader vložený na zobrazené stránce.

Offline scénáře lze charakterizovat následovně. Přístup do systému SAP není možný ani není vyžadován. To znamená, že uživatel při zpracování formuláře nemusí volat systém SAP. Proto musí být nápovědy s hodnotami, validace a jednoduché výpočty obsaženy již ve formátu PDF pomocí skriptovacího jazyka. A až po dokončení zpracování formuláře PDF se data převedou do systému SAP. Tento přenos se provádí dvěma způsoby. Nepřímo prostřednictvím e-mailu, který obsahuje formulář PDF, nebo přímo přihlášením k systému SAP. Následně se nahraje formulář PDF rovnou do aplikace. Pokud uživatel není přihlášen k firemnímu intranetu, obvykle není možné získat přístup k systému SAP. Pokud je uživatel přihlášen k intranetu, tak může odeslat formulář PDF nebo data v něm obsažená do systému SAP. Odeslání může realizovat prostřednictvím webové služby nebo přes HTTP volání. Další charakteristikou offline scénáře je nezávislost na dalších technologiích uživatelského rozhraní. Formuláře PDF se otevírají a zpracovávají výhradně v aplikaci Adobe Reader. Integrace s jinou technologií pro vytváření uživatelských rozhraní proto není nutná. Tyto dvě charakteristiky vyžadují pouze to, aby formulář PDF obsahoval všechny nezbytné informace (například data a nápovědy s hodnotami). To může mít za následek větší soubory PDF, jež mohou ovlivnit dobu přenosu[11].



Obrázek 3.8: Scénáře jednotlivých formulářů Adobe[19].

3.5.2 Formuláře bez možnosti úpravy

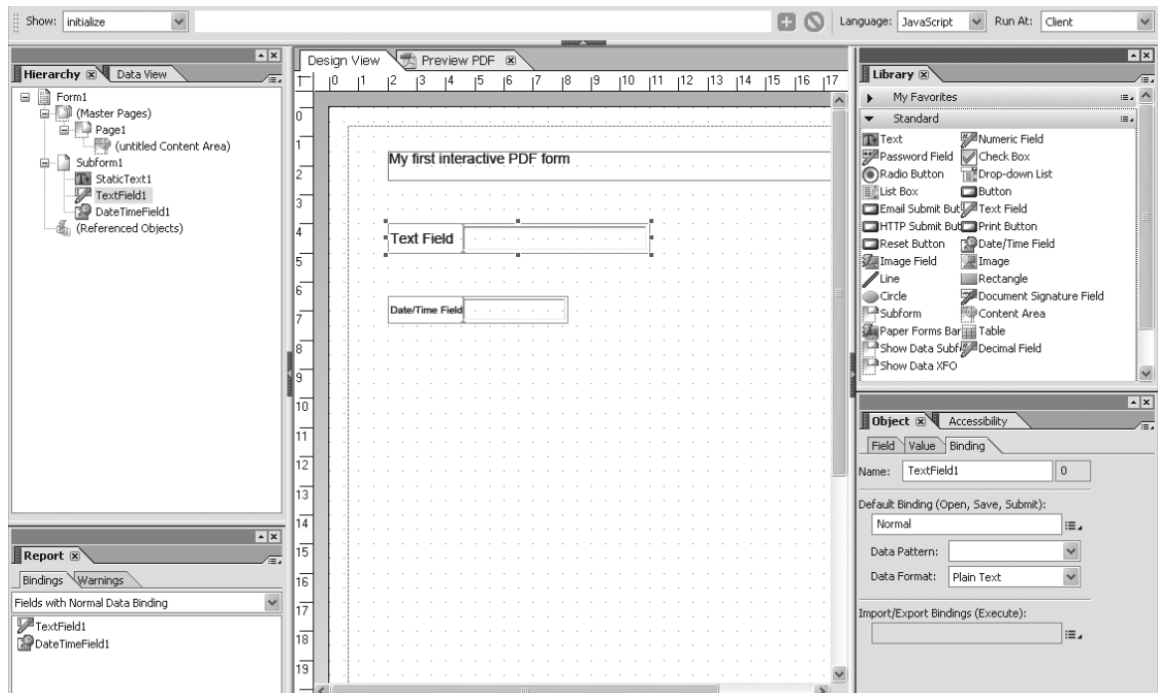
Navržené formuláře slouží k zobrazování informací bez možnosti změny formuláře. Mezi takové typy formulářů si lze představit například tisk vyplněné faktury, která byla doplněna daty z nějakých databází.

3.5.3 Statické formuláře

Jedná se o formuláře, jež převážně slouží jako šablona pro ruční dopsání po vytisknutí formuláře.

3.6 Adobe LiveCycle

Adobe LiveCycle Designer je nástroj, jenž využívá programátor k vytváření a rozvržení formulářů. Rozvržení tiskových formulářů založených na formátu PDF a interaktivních formulářů se vytváří stejným způsobem. Pro oba tyto typy formulářů se využívá nástroj Adobe LiveCycle Designer. Tyto formuláře lze snadno propojit s podnikovými daty a procesy[11]. Nástroj také umožňuje nahrání již vytvořeného Adobe interaktivního formuláře do prostředí SAP a pak jej libovolně upravovat nebo napojit na podnikový systém.



Obrázek 3.9: Prostředí Adobe LiveCycle[20].

3.7 Čárové kódy v systému SAP

V systému SAP existují dvě obecné třídy čárových kódů. Jednorozměrné (1D) a dvourozměrné (2D). Používají se v různých typech aplikací a v některých případech jsou skenovány pomocí různých typů technologií. Rozdíl mezi 1D a 2D skenováním čárových kódů závisí na rozložení a množství dat, které lze do každého kódu uložit. Oba typy lze efektivně použít v různých aplikacích k automatické identifikaci[6].

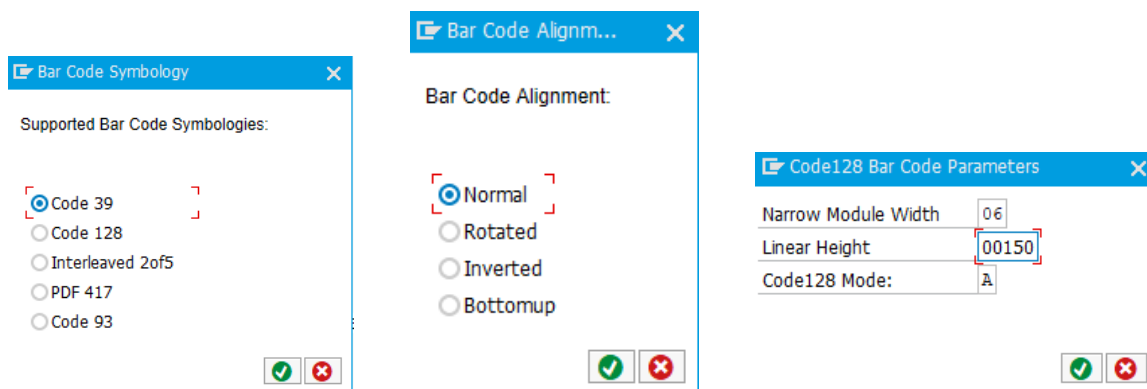
1D

1D čárové kódy jsou kódy podobné UPC kódu, jenž je běžně používán u spotřebního zboží. Kódy se používají k zakódování dat, a to pomocí řady čar a mezer s různou šířkou. 1D čárové kódy obsahují jen několik desítek znaků. S přidáváním dalších údajů se obecně fyzicky prodlužuje jejich velikost.

2D

2D čárové kódy se používají ke kódování dat v podobě vzorů čtverců, šestiúhelníků, teček a dalších tvarů. Díky své struktuře mohou 2D čárové kódy pojmout více dat než 1D čárové kódy, přičemž stále jsou fyzicky menší. Data jsou kódována na základě vertikálního i horizontálního uspořádání vzorů, takže jsou čtena ve dvou rozměrech. 2D čárový kód nekóduje pouze alfanumerické informace. Tyto kódy mohou také obsahovat obrázky, adresy webových stránek a další typy binárních dat. To znamená, že můžete tyto informace využít, ať už jste připojeni k databázi nebo ne. Mezi tyto čárové kódy patří například Data Matrix, QR Code nebo PDF417.

V systému SAP se musí nejprve nadefinovat vlastní formát čárového kódu než ho lze použít ve formulářích. Vytvářet a spravovat je lze v transakci¹ SE73. V sekci s volbou změna čárových kódů systému. V této transakci si tedy lze nadefinovat vlastní formát čárového kódu. SAP umožňuje výběr z pěti druhů čárových kódů (SAP ECC 6). A to konkrétně Code 39, Code 128, Interleaved 2of5, PDF 417 a Code 93. Po výběru druhu čárového kódu si lze poté zvolit orientaci čárového kódu a další vlastnosti vzhledu, které lze vidět na obrázku 3.10.



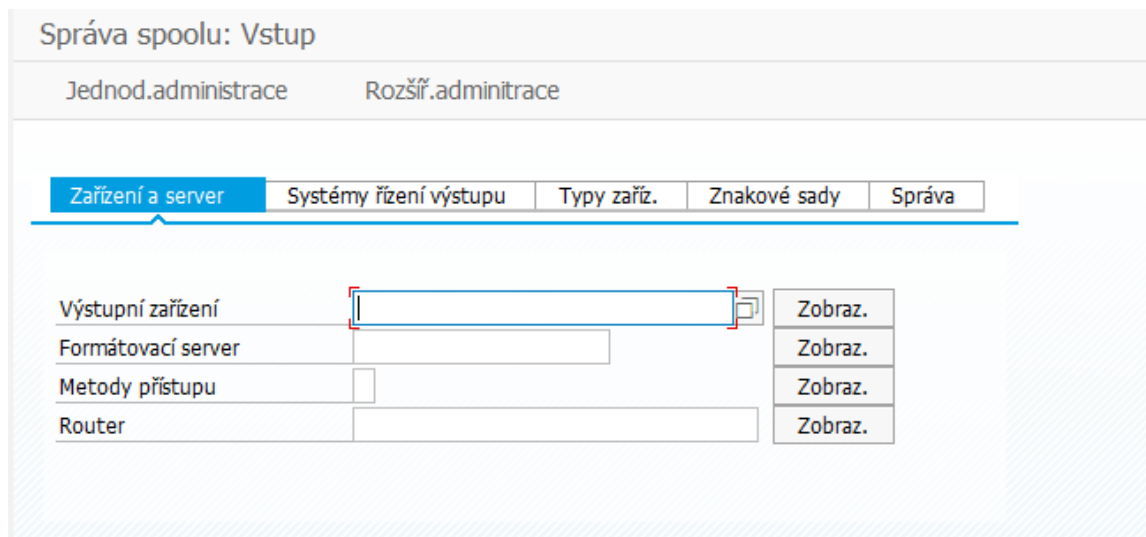
Obrázek 3.10: Možnosti čárových kódů v systému SAP.

¹Transakce je v terminologii SAP základní pojem, který představuje zkratku pro spustitelný program. V SAP GUI pomocí příkazové řádky lze tyto transakce spouštět. Běžné syst. transakce mají předponu SE.

3.8 Nastavení tisku v systému SAP

V systému SAP se nastavení všech tiskáren provádí v transakci SPAD.

3.8.1 Založení výstupního zařízení



Obrázek 3.11: Nastavení tisku v systému SAP.

Atributy zařízení

- Typ zařízení: Výběr z mnoha řad tiskáren - pro snadnější výběr je k dispozici průvodce nastavení.
- Formátovací server: Server, který bude vše zpracovávat.
- Třída zařízení: Zde se zadává, o jaký druh výstupního zařízení se jedná. Konkrétně lze zde vybírat z možností, jako je normální tiskárna, archiv. program, fax, pool zařízení, logické výstupní zařízení.
- Skupina oprávnění: Zde se může přiřadit skupina oprávnění.
- Model, stanoviště a zpráva: Tato pole nejsou pro samotný tisk důležitým parametrem a slouží k přehlednosti identifikace tiskáren v budově, a to v případě mnoha zařízení.

Správa spoolu: Založení výstupního zařízení

Výstupní zařízení: **Mojetiskarna** Krátký název: **MTIS**

Popis

Atributy zařiz. Spojení hostspoolu Výstupní atributy Info podav.

Typ zařízení: [včetně ikony] :

Formátovací server: [včetně ikony]

Popis serveru

Počítač: **ehpides1** Reálný server: [včetně ikony]

Třída zařízení: **Normální tiskárna**

Skupina oprávnění: [včetně ikony]

Model: [včetně ikony]

Stanoviště: [včetně ikony]

Zpráva: [včetně ikony]

Blok.tiskárny ze str.SAP

Obrázek 3.12: Atributy zařízení.

Spojení hostspoolu

- Druh připojení k systému hostspoolu: Udává, jak má spoolový workspace předat data pro tuto tiskárnu na spoolový systém hostspoolu.
 - Jestliže spoolový workproces pro tuto tiskárnu běží v systému hostspoolu, tak lze vybrat některou z následujících možností:
 1. C - jestliže spool server běží na počítači pracujícím pod Microsoft Windows NT, nebo tiskové požadavky pro tento výstup mají být předány lokálnímu nástroji Windows NT Print Manager
 2. E - pokud je tiskárna připojena přes externí OMS (Output Management System)
 3. F - pokud se mají výstupní požadavky pro tuto tiskárnu vytisknout lokálně na PC
 4. I - pokud je toto zařízení zařízením archivačním
 5. L - pokud spool server běží na systému UNIX a výstupní požadavky na tuto tiskárnu mají být předány lokálnímu spool hostu
 - Pokud probíhá formátování pro tisk na nějakém jiném systému, tak se vybírá mezi následujícími možnostmi zadání:

1. U - pokud má formátovací proces předat výstupní data na vzdálený systém prostřednictvím Berkeleyského protokolu. Formátovací proces předává výstupní data přímo prostřednictvím tohoto protokolu na spool host ve vzdáleném systému
 2. S - pokud má formátovací proces předat výstupní data na vzdálený systém prostřednictvím protokolu SAP
 3. X - pokud je zařízení typu SAPcomm
- Hostující tiskárna: Zde se nastavuje IP adresa tiskárny.

Správa spoolu: Založení výstupního zařízení

Výstupní zařízení: Krátký název:

Popis

Atributy zařiz. | **Spojení hostspoolu** | Výstupní atributy | Info podav.

Met.příst.k hostspoolu:

Host.tiskárna:

Počítač:

Zpětné hlášení statusu pomocí poolingu
 Bez dotazu na tiskové požadavky u hostspoolu

Obrázek 3.13: Spojení hostspoolu.

Výstupní atributy

- Krycí list SAP: Volba, zda se má na tomto výstupním zařízení tisknout titulní strana.
- Krycí list jazyka SAP: Určuje v jakém jazyce se realizuje výstup krycího listu.
- Krycí list hotspoolu: Volba, zda se má tisknout krycí list definovaný ve spool hostu.

- Kontrola pomocí infrastruktury monitoringu: Pokud je tato volba aktivní, pak se prostřednictvím infrastruktury monitoringu monitoruje zpracování jobů a požadavky spool hostu na tomto zařízení.
- Sekvenční zpracování požadavků: Pokud se toto pole označí budou požadavky zpracovávány v generovaném pořadí.
- Čítač kopií: Volba jak zpracovávat požadavky spoolu při tisku více kopií.
- Výstupní atributy pro ovladač sestavy a Smart Forms: Zde se definuje, zda tiskárna podporuje barevný tisk nebo speciální tiskový režim. A to včetně volby vkládání papíru.

Správa spoolu: Založení výstupního zařízení

Výstupní zařízení: Krátký název:

Popis: _____

Atributy zařiz. Spojení hostspoolu **Výstupní atributy** Info podav.

Zpracování výstupních požadavků

Krycí list SAP Krycí list jazyka SAP

Krycí list hostspoolu Netisknout krycí list host-spool

Kontrola pomocí infrastruktury monitoringu

Sekvenční zpracování požadavků

Čítač kopií Předání čítače kopií na spoolový systém host

Výstupní atributy pro ovladač sestavy a Smart Forms

Barev.tiskárna Tiskový režim DEFAULT: Implicitní nastavení tiskárny

Vkládání papíru Přednast.zásobník

Obrázek 3.14: Výstupní atributy.

Info podavače

- Podavač: Slouží pro starší typy zařízení, které potřebují informaci ohledně čísla podavače a konkrétního formátu strany při tisku sestav.
- Formát strany

3.8.2 Vytvoření typu zařízení

- Ovladač: Výběr ovladače, jenž je používán editorem SAPscript.
- Stránková tiskárna: Volba pro tiskárny, které jsou schopné formátovat celé stránky.
- Ovladač tiskárny: Výběr ovladače určený k výstupu tisku sestav ABAP.
- Argument: Argument určený pro ovladač tiskárny.
- Znaková sada: Definování znakové sady.

Správa spoolu: Založení typu zařízení

Typ zařízení Název

Vlastnosti

Verze

Zpracování pomocí SAPscriptu

Ovladač

Stránková tiskárna

Zpracování seznamů

Ovladač tiskárny

Argument

Znaková sada tiskárny

1. Znaková sada

Synt.znaková sada

Obrázek 3.15: Vytváření nového typu zařízení.

3.9 ESim

ESim je vestavěný protokol určený výhradně pro použití s tiskárnami štítků od výrobce Intermec. ESim protokol je navržen tak, aby podporoval jakýkoliv počítač, terminál, skener nebo klávesnici, která dokáže produkovat znaky ASCII pro ovládání tiskárny. S protokolem ESim lze použít libovolný editor pro ovládání tiskárny. A to přes USB nebo ethernetového rozhraní. Formát štítku se skládá z několika různých polí, které obsahují různé typy dat. Údaje v polích mohou určit počáteční pozici, rotaci, zvětšení a další vlastnosti[12].

Kapitola 4

Specifikace požadavků a návrh

Tato kapitola se zabývá detailním návrhem a specifikací požadavků mého řešení aplikace pro generování všech typů formulářů a štítků. Návrh byl konzultován s osobou pohybující se v ekonomické sféře, aby tyto formuláře a štítky odpovídaly návrhově skutečným potřebám v praxi a byly zároveň akceptovány společností České pošty.

4.1 Vyhledávací kritéria

Cílem je SAP aplikace, která bude vyhledávat data z personálních kmenových záznamů, adres, personálních kmenových záznamů dat o osobě, kmenových souborů zákazníků a kmenových souborů dodavatelů. Z těchto dat bude možné vybírat konkrétní adresní informace na základě pěti skupin vyhledávacích vstupních polí aplikace. A to konkrétně na základě účetního okruhu, všeobecného názvu, čísla odběratele, čísla dodavatele a čísla zaměstnance. U čísla zaměstnance bude navíc možnost zvolení druhu adresy.

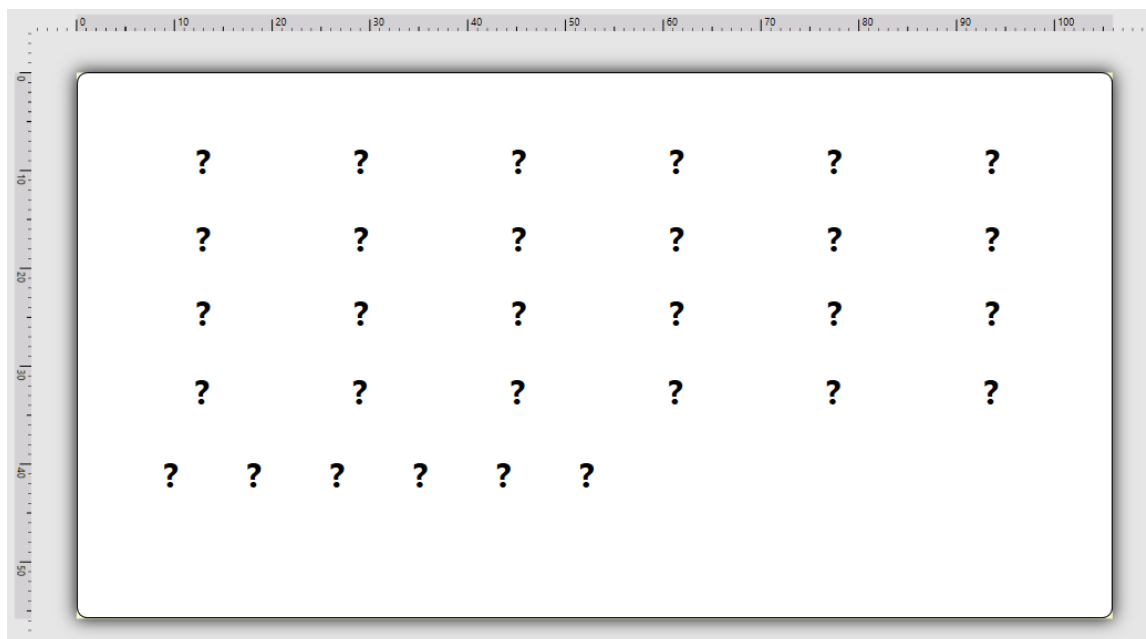
4.2 Typy formulářů

V této podkapitole jsou jednotlivé typy formulářů popsány z hlediska požadavků. Dále jsou u každého typu formuláře vytvořeny návrhy řešení.

4.2.1 Formulář pro tisk nalepovacího adresního štítku na tiskárně Intermecc PC43t

Návrh prvního formuláře byl vytvořen na základě zkušeností ekonomických pracovníků s častým zasíláním obálek. Adresy se musí ručně vyplňovat a zabírá to mnoho času. Také se často stává, že se obálka vyplní chybně, a to díky lidské chybě. Snadnější volbou je výběr adres v aplikaci podnikového systému SAP s aktuálními adresami a následně tisk samotných adres na lepící štítky. Tyto lepící štítky se poté jen snadno nalepí na příslušnou obálku. U tohoto typu formuláře je potřeba pro tisk speciální tiskárna. Konkrétně je potřeba tiskárna pro tisk lepících štítků. Pro tenhle účel byla vybrána tiskárna Intermecc PC43t. Tiskárna byla zvolena na základě nízké ceny a dostatečné velikosti štítků, které umí tiskárna zpracovat. Minimální velikost štítku byla zvolena na základě požadavků. Při specifikaci rozměrů adresních štítků bylo potřeba dobře rozhodnout o běžné maximální délce adres. Muselo se počítat s adresami v České republice, ale i v Německu. Mezi požadavky tisku bylo zapotřebí název společnosti, ulice, číslo popisné, město, poštovní směrovací číslo a zkratka

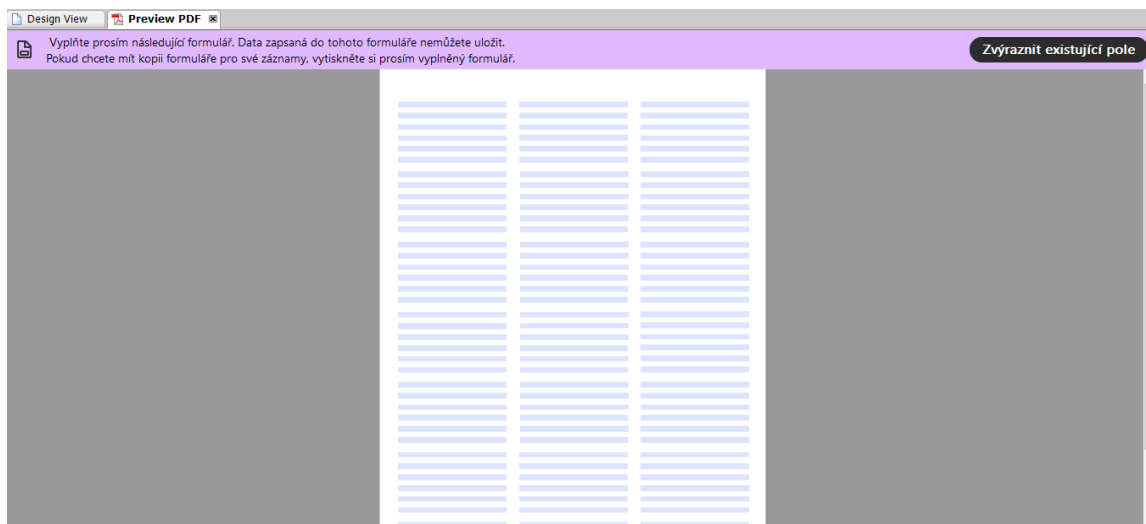
země. Z následujících požadavků byl vytvořen následující návrh takového štítku, který lze vidět na obrázku 4.1.



Obrázek 4.1: Návrh formuláře pro tisk lepicího adresního štítku.

4.2.2 Formulář pro tisk 21 nalepovacích adresních štítků

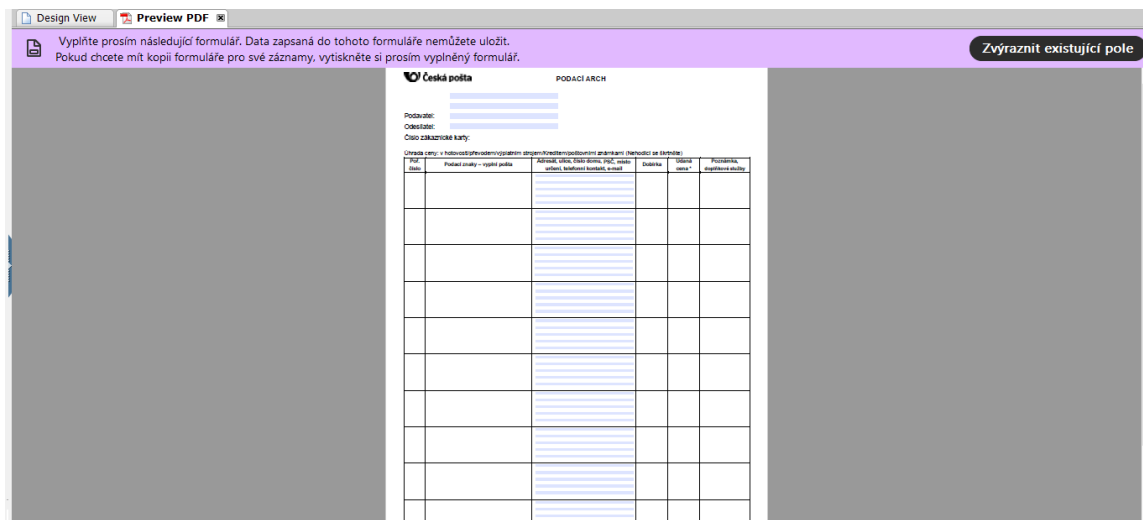
Tento formulář je velmi podobný předchozímu formuláři a taky z něj vychází. Tento formulář slouží pro hromadný tisk 21 kusů lepicích adresních štítků ve formátu stránky A4. Tento tisk má sloužit k tisku na klasické tiskárně se speciálním papírem, jenž obsahuje lepicí štítky. Vytvořený návrh formuláře lze vidět na obrázku 4.2.



Obrázek 4.2: Návrh formuláře pro tisk 21 nalepovacích adresních štítků.

4.2.3 Formulář pro tisk podacího archu České pošty

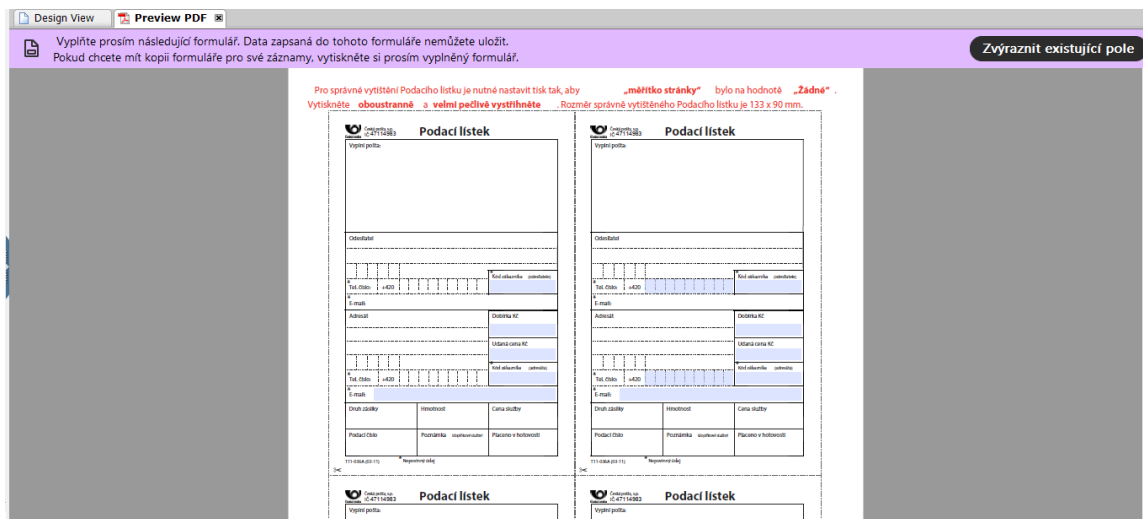
Podací arch České pošty je další papírová forma, která se využívá velmi často. Vyplnění adres zabere opět mnoho času, proto se rozhodlo pro automatizaci vyplňování adres i u tohoto formuláře. Zde se jedná o jednostranný tisk, jenž je potřeba tisknout na klasické tiskárně. Adresní požadavky, zde zůstávají naprosto stejné. Nově je zde, zapotřebí počítat se jménem a adresou odesílatele. Vytvořený návrh formuláře lze vidět na obrázku 4.3.



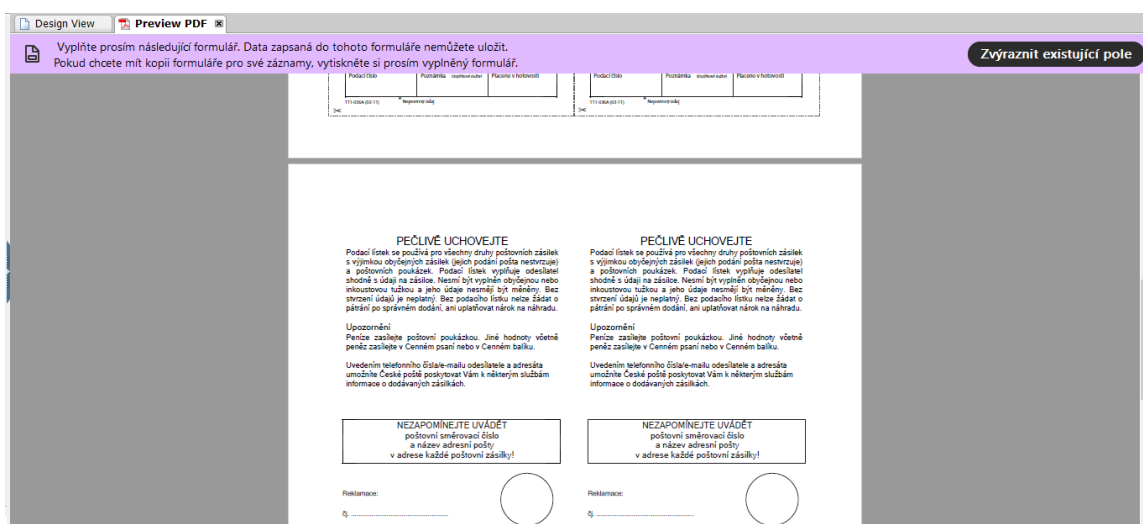
Obrázek 4.3: Návrh formuláře pro tisk archu České pošty.

4.2.4 Formulář pro tisk podacích lístků České pošty

Dalším formulářem jsou podací lístky České pošty. Požadavkem byl oboustranný tisk těchto štítků na klasické tiskárně. Na základě požadavků bylo zjištěno, že lze mít až 4 podací lístky na jedné straně A4. Proto byl vytvořen návrh dvou stran formuláře, kde první strana je přední část podacího lístku České pošty a druhá strana odpovídá zadní straně téhož podacího lístku. Formulář byl inspirován PDF souborem podacího lístku přímo ze stránek České pošty, aby byl vygenerovaný podací lístek akceptován touto společností. Adresní požadavky zde byly opět naprosto stejné, ale muselo se zde počítat opět i s názvem a adresou odesílatele, který je zde navíc. Vytvořený návrh formuláře lze vidět na obrázcích 4.4 a 4.5.



Obrázek 4.4: Návrh formuláře pro tisk podacích lístků České pošty - přední strana.

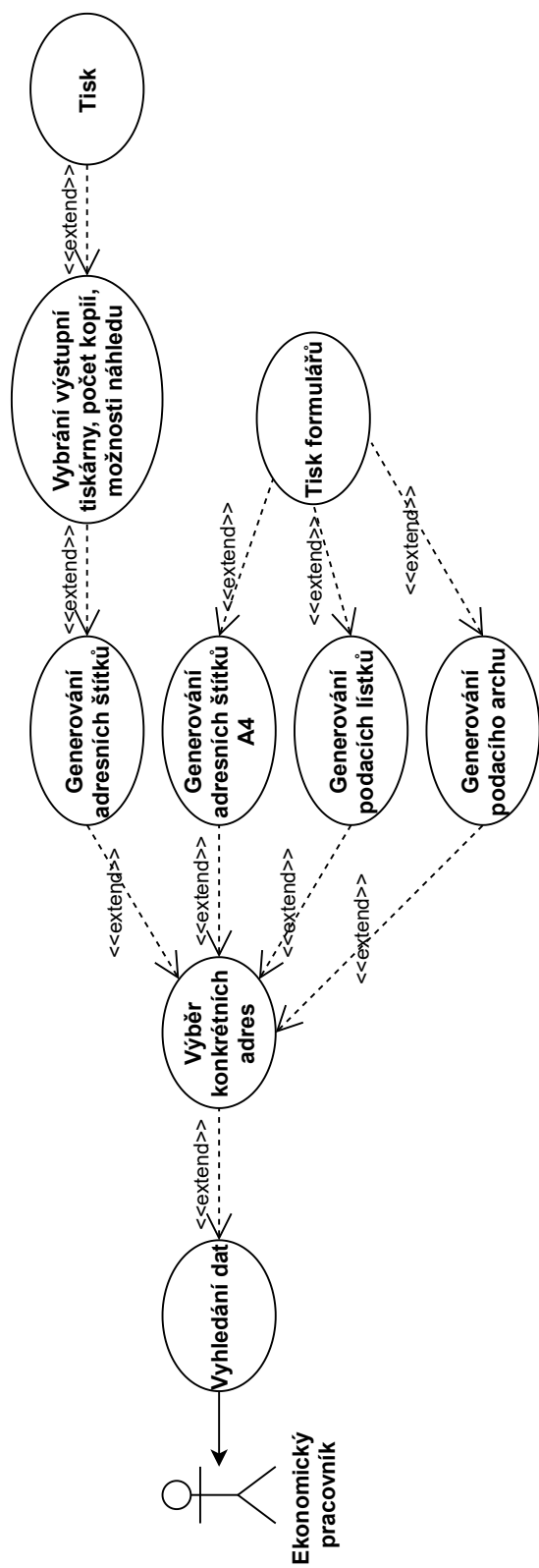


Obrázek 4.5: Návrh formuláře pro tisk podacích lístků České pošty - zadní strana.

4.3 Funkcionalita

U aplikace byla požadována výběrová obrazovka s vyhledávacími poli, která budou pokrývat všechna již zmíněná vyhledávací kritéria. S těmito požadavky bylo nutné počítat v rámci návrhu aplikace a její funkcionality. Výběr účetního okruhu s dalšími vyhledávacími parametry reprezentuje prvotní obrazovka programu. Spuštění této obrazovky je vyvoláno pomocí transakce, která je popsána v další kapitole. Podle vyplněných kritérií jsou tyto parametry uloženy a je spuštěn výběr dat z databáze, kde jsou uloženy patřičné datové záznamy. Po procesu vyhledání dat jsou tato data převedena do struktury pro zobrazení v SALV a vzápětí jsou zobrazena jako seznam dat v tabulce.

Uživatel má poté možnost si vybrat konkrétní záznamy dat, které se vygenerují do zvolených formulářů. Při výběru tisku štítků na tiskárně Intermec je zobrazováno vyskakovací



Obrázek 4.6: Diagram případu užití aplikace.

okno, jež umožňuje zvolit tiskárnu a počet kopií tisku. V případě ladění umí i zajistit náhled tisku.

V ostatních případech formulářů se formuláře rovnou generují a zobrazují se přímo v systému SAP. Zároveň při tomto zobrazení lze formulář tisknout rovnou na lokálních tiskárnách. Požadované funkce aplikace v systému SAP jsou přehledně graficky znázorněny v diagramu případu užití. Diagram případu užití je jedna část ze základních stavebních bloků modelovacího jazyka UML. Popisuje sekvenci kroků mezi účastníkem a systémem. Tento diagram případu užití lze vidět na obrázku 4.6. Uživatelem aplikace se předpokládá ekonomický pracovník firmy, který denně používá podnikový systém od firmy SAP. Tudíž v rámci návrhu nemuselo být počítáno se speciální vizuální podobou, ale stačilo vytvořit běžné prvky prostředí systému SAP.

4.4 Struktura adres

Základ všech adresních štítků nebo formulářů závisí na správném návrhu struktury, která bude sloužit pro všechny možné využití v rámci vyplňování adresy. Struktura pro tisk adresních štítků reprezentující všechny podstatné informace pro samotný tisk adresy byl zvolen ZPDF_PODACI_LISTKY_S. Tato struktura byla vytvořena za pomoci transakce SE11 v SAP Dictionary. Tato struktura je popsána a pro lepší přehlednost znázorněna v tabulce 4.1.

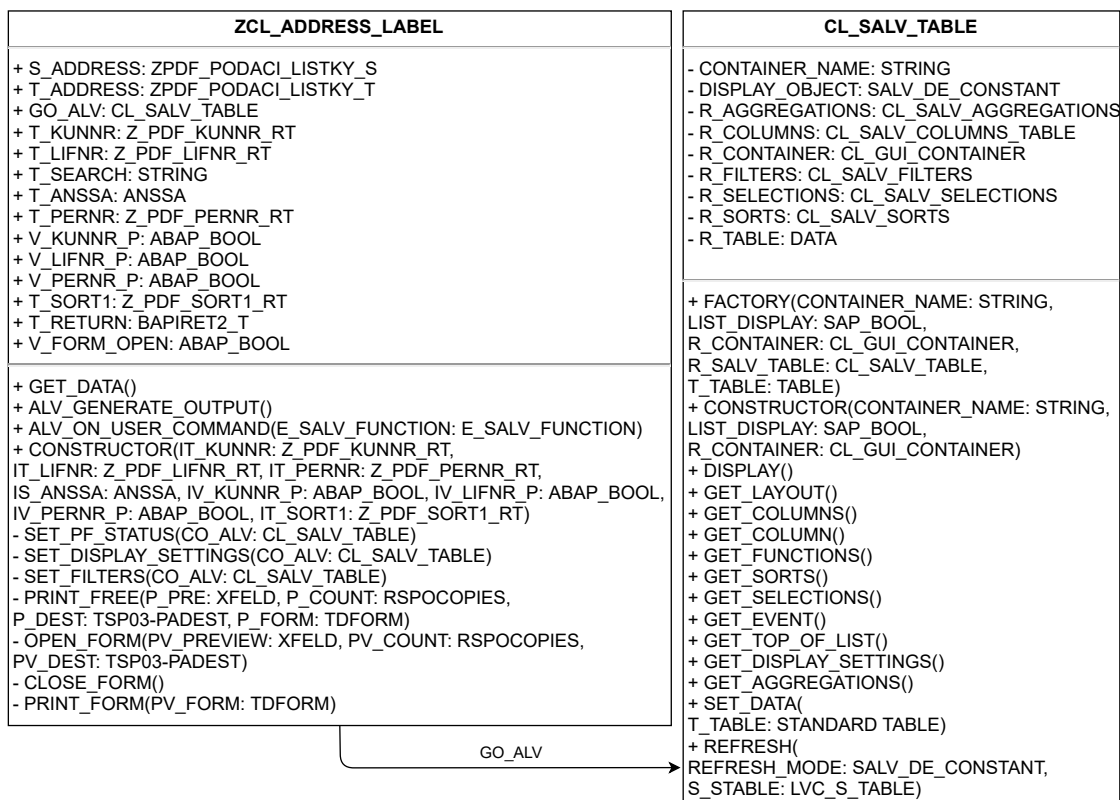
Komponenta	Ref. typ	Typ komp.	Datový typ	Délka	Des. č.	Krátký popis
KUNNR		KUNNR	CHAR	10	0	Číslo odběratele1
LIFNR		LIFNR	CHAR	10	0	Číslo účtu dodavatele
PERNR		PERSNO	NUMC	8	0	Osobní číslo
NAME1		AD_NAME1	CHAR	40	0	Jméno1
NAME2		AD_NAME2	CHAR	40	0	Jméno2
CITY1		AD_CITY1	CHAR	40	0	Místo
POST_CODE1		AD_PSTCD1	CHAR	10	0	PSČ obce
STREET		AD_STREET	CHAR	60	0	Ulice
HOUSE_NUM1		AD_HSNM1	CHAR	10	0	Číslo domu
COUNTRY		LAND1	CHAR	3	0	Klíč státu

Tabulka 4.1: Struktura ZPDF_PODACI_LISTKY_S.

Některé názvy atributů nemají příliš odpovídající název. U volby názvů atributů bylo nahlíženo na aktuálně používané názvy v jednotlivých originálních tabulkách kmenových dat. Běžný uživatel se s názvy atributů nesetká. Při případném hledání technických názvů polí se očekává, že aktuálním uživatelem je programátor a ten se vyzná v systémových názvech. Tato struktura byla využita tedy jako základ pro vznik dalších tabulek, jež se budou používat u různých typech formulářů. Tyto nové tabulky byly popsány níže, a to konkrétně v kapitole 5.

4.5 Objektové řízení tisku

Jazyk ABAP je poměrně starý jazyk, přesto jeho neustálý vývoj přinesl nedávno novou verzi, která zajistila integraci s objektivně orientovaným přístupem. Změna orientace vývojového pohledu na objekty nám zjednodušuje návrh softwaru, ale hlavně nám usnadňuje pochopení, údržbu a znovupoužití. Objektově orientované programování, neboli ve zkratce OOP, nám představuje a umožňuje jiný způsob myšlení při vývoji softwaru. Hlavní výhodou OOP spočívá v jeho jednoduchosti. Objektově orientované programování se tedy už dle názvu zaměřuje na vytváření a pracování se samotnými objekty, jež reprezentují konkrétní entity reálného světa nebo nějaké abstraktní entity. Každá entita, neboli objekt, má jasné definované charakteristické vlastnosti, které se uchovávají v podobě struktur a atributů daného objektu. Funkčnost objektu je popsán v podobě seznamu metod, jež tedy reprezentují chování samotného objektu. Objektivní model byl zvolen z hlediska budoucího vývoje dalších speciálních podob tisku, kdy je modifikace mnohem jednodušší než u imperativního vývoje. Na obrázku 4.7 je znázorněn diagram tříd.



Obrázek 4.7: Diagram tříd.

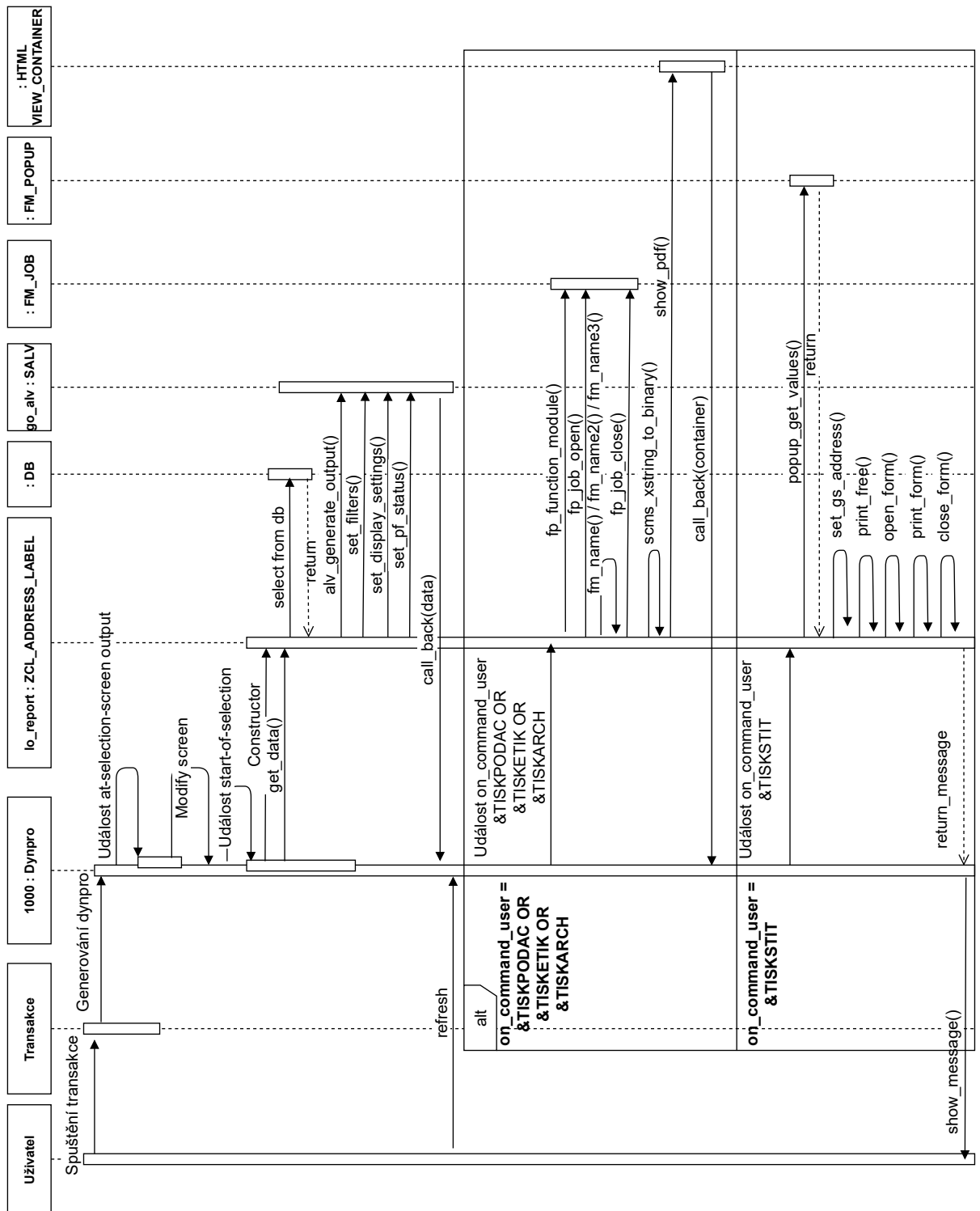
Znázorněná třída ZCL_ADDRESS_LABEL se skládá z atributů S_ADDRESS, T_ADDRESS, GO_ALV, T_KUNNR, T_LIFNR, T_SEARCH, T_ANSSA, T_PERNR, V_KUNNR_P, V_LIFNR_P, V_PERNR_P, T_SORT1, T_RETURN, V_FORM_OPEN. Názvy atributů byly sestaveny z logického hlediska, která se používají v praxi při vývoji v systému SAP. Ty jsou odvozeny z praktického hlediska a slouží hlavně pro jednodušší zorientování v kódu. Pokud název atributu začíná se dvěma znaky před podtržítkem, tak se většinou značí nejprve typ proměnné a dále následuje znak, jenž označuje datový typ. Pokud se nachází v názvu pouze jeden znak před podtržítkem, tak

se jedná pouze o datový typ. Co se týká datových typů, tak se většinou jedná o struktury, hodnoty, tabulky nebo objekty. Tyto atributy představují jednotlivé zadávací pole vstupní obrazovky. Třída byla sestavena z konstruktoru, který inicializuje atributy třídy na základě vyplněných vyhledávacích polí vstupní obrazovky. Metoda `GET_DATA` byla určena k získání všech nutných dat ze všech databázových tabulek. Další vytvořenou metodou byla `ALV_GENERATE_OUTPUT`, jež slouží k vytvoření SALV instance, nastavení SALV dat a zobrazení této tabulky. Metody `SET_DISPLAY_SETTINGS` a `SET_FILTERS` slouží k dodatečné úpravě již vytvořené tabulky SALV. Předposlední metoda `SET_PF_STATUS` zajišťuje zobrazení a funkční logiku uživatelských ikon této aplikace. Poslední metoda `ALV_ON_USER_COMMAND` se stará o logiku generování a řízení tisku na základě uživatelské interakce.

Druhou využívanou třídou je `CL_SALV_TABLE`. Je to systémová třída, sloužící pro obsluhu SALV tabulky. Tato třída pokrývá všechny potřebné funkce pro zobrazování dat v podobě tabulky a její úpravy. A to včetně funkcí pro zpracovávání uživatelských událostí v tabulce.

4.6 Návrh postupu řízení tisku

V podkapitole 4.5 byl sepsán a představen návrh pro implementaci objektivního řízení tisku pro tuto aplikaci. Kompletní schéma postupu řízení tisku od počátku zavolání aplikace až po samotný tisk reprezentuje sekvenční diagram, který lze vidět na obrázku 4.8. Postup řízení začíná po spuštění samotné aplikace a po výběru hledaných dat uživatelem. Po této události je následně vytvořen a volán objekt pro řízení generování a tisku jednotlivých formulářů. Na základě uživatelské interakce je potom zvolena příslušná akce s konkrétními adresními daty. V případě formulářů určené k tisku na klasických lokálních tiskárnách je volán funkční modul, který vytváří dynamický název nového funkčního modulu. Tento funkční model pak slouží pro generování zvoleného formuláře. Před samotným voláním tohoto nově vytvořeného funkčního modulu je vytvořen job. Job je proces, který běží na pozadí aplikace či systému. Do tohoto job procesu je právě přiděleno generování Adobe formuláře. Vygenerovaný formulář je následně převeden do binární podoby a předán prostřednictvím metody `SHOW_PDF` do HTML kontejneru, kde je zobrazen. V tomto kontejneru lze následně provést tisk. U varianty s tiskem štítků na tiskárně Intermecc PC45t je generování a řízení tisku zcela odlišné. V této sekvenci je voláno vyskakovací okno s možností tisku. Následně jsou data pomocí metody `SET_GS_ADDRESS` předána do SAP Script formuláře a je volán rovnou tisk na tiskárně. Celý proces tisku se skládá z provedení přípravy tisku metodou `PRINT_FREE()`, vytvoření SAP Script formuláře metodou `OPEN_FORM` a ve finále z provedení tisku za pomoci metody `PRINT_FORM`. Konec této sekvence uzavírá metoda `CLOSE_FORM`, která uvolňuje zvolený SAP Script formulář.



Obrázek 4.8: Sekvenční diagram řízení tisku formulářů.

Kapitola 5

Implementace

Tato kapitola popisuje konkrétní části při implementaci aplikace. V první části popisuje způsob a tvorbu programu. Dále se zaměřuje na vytvoření transakce pro tento program. V druhé části je popsán vývoj výběrové obrazovky se získáváním dat. Tato část navazuje dále na zpracování těchto dat pomocí SALV. V předposlední části se objasňuje detailní popis implementace jednotlivých typů formulářů. A v poslední části se vysvětluje způsob kontroly správnosti kódu a konfigurace tiskárny Intermec v SAP systému.

5.1 Programování v jazyce ABAP

Implementace programu byla vytvořena v jazyce ABAP. A to pomocí transakce SE80 v podnikovém systému SAP. Editor a vývojové prostředí v SAP systému neumožňuje zvýrazňování konstant, funkcí ani datových typů, jak je známo při jiných vývojových prostředích. Vývoj tedy není zcela jednoduchý a ani vývojářsky přívětivý.

5.2 Popis programu

Program neboli report, který je používán v terminologii SAP systému, sloužící ke spuštění samotné aplikace byl pojmenován Z_PDF_ADDRESS_PRINT. Zvolený program se skládá z vícero částí. Tyto části musí být naimportované ve správném pořadí a to hlavně kvůli úspěšnému překladu kódu. První část vkládá do programu všechny potřebné globální proměnné, konstanty, definované vlastní typy, obecné datové typy a struktury databázových tabulek. Pro definování nových proměněných se zde využívají nadefinované datové typy v SAP dictionary. Druhá část nám vkládá nadefinovanou vstupní výběrovou obrazovku, a to včetně jejího řízení a změny logiky na základě vyplněných vstupních dat. Vstupní parametry výběrové obrazovky jsou ukládány do globálních proměnných a je s nimi v dalších krocích nakládáno. Kromě toho se zde vkládá také logika řízení uživatelských příkazů jednotlivých tlačítek u výběrové obrazovky. V neposlední řadě jsou volány metody třídy ZCL_ADDRESS_LABEL, které nám zajišťují získání všech potřebných dat, generování jednotlivých formulářů a v neposlední řadě řízení kompletního tisku. Poslední část nám vkládá všechny potřebné podprogramy, jež slouží pro inicializaci dat, přípravu a zakládání jednotlivých formulářů. Tisk je zde realizován pomocí podprogramu vytvářející joby, které nám předpřipravuje třída tisku ZCL_ADDRESS_LABEL. Program se spouští pomocí klávesy F8 nebo kliknutím na ikonu s hodinami, jež mají zelenou fajfku. Tato ikona se nachází vlevo nahoře v hlavním panelu obrazovky.

5.2.1 Paket

Paket, nebo v anglické terminologii také package, je vývojový objekt, který tvoří základní moduly projektů ABAP. Lze je zobrazit v transakci SE80. Tyto pakety se zobrazují v průzkumníku projektu jako strom úložiště ABAP. Každý uzel paketu může obsahovat dílčí pakety, nebo vývojové objekty jako dílčí uzly. Hierarchie paketů je tedy reprezentována jako strom uzlů. Představuje technický pohled na architekturu z aplikace, nebo systém SAP. Pro tuto aplikaci byl založen paket s názvem Z_PDF. Všechny vytvářené objekty, textové zprávy a zdrojové kódy byly ukládány do tohoto paketu.

5.2.2 Transakce

Pro snadný způsob spuštění aplikace byla vytvořena transakce ZPOSTA. Této transakci byl přidělen report Z_PDF_ADDRESS_PRINT a konkrétní číslo výběrové obrazovky. Při zavolání této transakce je spuštěna kontrola autorizace, podle přihlášeného uživatele, a následně je zobrazena definovaná výběrová obrazovka. Úplná definice transakce lze vidět na obrázku 5.1.

The screenshot shows the 'Zobrazení Transakce reportu' (Transaction Report) screen in SAP. The transaction code 'ZPOSTA' is entered in the 'Kód transakce' field, and the package 'Z_PDF' is shown in the 'Paket' field. The 'Text transakce' is 'Tisk adresních štítků', the 'Program' is 'Z_PDF_ADDRESS_PRINT', and the 'Výběrová obrazovka' is '1000'. The 'Objekt oprávnění' field is empty, and there is a 'Hodnoty' button. The 'Klasifikace' section shows 'Profesionální uživ.transakce' selected, 'Transakce Easy Web' with 'Servis' in the adjacent field, and 'Pervasive enabled' unchecked. The 'Schopnost GUI' section has 'SAP GUI pro HTML' and 'SAP GUI pro Java' unchecked, and 'SAP GUI pro Windows' checked.

Kód transakce	ZPOSTA
Paket	Z_PDF
Text transakce	Tisk adresních štítků
Program	Z_PDF_ADDRESS_PRINT
Výběrová obrazovka	1000
Start s variantou	
Objekt oprávnění	
<input type="button" value="Hodnoty"/>	

Klasifikace

Klasifikace transakce

- Profesionální uživ.transakce
- Transakce Easy Web Servis
- Pervasive enabled

Schopnost GUI

- SAP GUI pro HTML
- SAP GUI pro Java
- SAP GUI pro Windows

Obrázek 5.1: Definice transakce ZPOSTA.

5.3 Výběrová obrazovka

Po spuštění reportu je zavolána a zobrazena výběrová obrazovka, která vyžaduje vyplnění vstupních formulářů. Tyto vyplněné parametry jsou velmi zásadní, jelikož se s nimi dále pracuje v dalších krocích při vykonávání samotného programu. Výběrovou obrazovku není vždy potřeba tvořit pomocí nástroje Screen Painter. Pokud nepoužijeme tento nástroj, musíme se spokojit pouze s jednou vstupní obrazovkou. A to s obrazovkou s číslem 1000. Bez použití nástroje Screen Painter je nutné využít příkazové řádky v kódu. A to pro samotné definování parametrů, bloků, vizuálních změn prvků (délka, typ, seskupení) nebo vyhledávacích nápověd pro pole. Na základě těchto ABAP příkazů obrazovky se vygeneruje vstupní obrazovka zcela sama. Zdrojový kód této definice obrazovky lze vidět na obrázku 5.2.

```
SELECTION-SCREEN BEGIN OF BLOCK b02 WITH FRAME TITLE text-002.
  PARAMETERS p_bukrs TYPE t001-bukrs DEFAULT 'KCT'.
SELECTION-SCREEN END OF BLOCK b02.

SELECTION-SCREEN BEGIN OF BLOCK b01 WITH FRAME TITLE text-001.
  SELECT-OPTIONS pt_sort1 FOR adrc-sort1 MODIF ID M4.
  PARAMETERS: p_kunnr RADIOBUTTON GROUP 1 DEFAULT 'X' USER-COMMAND flag.
  SELECT-OPTIONS pt_kunnr FOR knal-kunnr MODIF ID M1.
  PARAMETERS: p_lifnr RADIOBUTTON GROUP 1.
  SELECT-OPTIONS pt_lifnr FOR lfal-lifnr MODIF ID M2.
  PARAMETERS: p_pernr RADIOBUTTON GROUP 1.
  SELECT-OPTIONS pt_pernr FOR pa0006-pernr MATCHCODE OBJECT ewapernr MODIF ID M3.
  PARAMETERS: ps_ansa TYPE pa0006-ansa MODIF ID M3 DEFAULT '1'.
SELECTION-SCREEN END OF BLOCK b01.

AT SELECTION-SCREEN OUTPUT.

LOOP AT SCREEN.
  IF p_kunnr = abap_true AND screen-group1 = 'M1' OR
     p_lifnr = abap_true AND screen-group1 = 'M2' OR
     p_pernr = abap_true AND screen-group1 = 'M3'.
     screen-input = '1'.
  ELSEIF screen-group1 = 'M1' OR
         screen-group1 = 'M2' OR
         screen-group1 = 'M3'.
     screen-input = '0'.
  ENDIF.

  MODIFY SCREEN.

ENDLOOP.

START-OF-SELECTION.
```

Obrázek 5.2: Implementace výběrové obrazovky a její logiky.

Při implementaci bylo využito dvou základních vstupních typů parametrů. A to základní jednoduché vstupní pole pomocí příkazu PARAMETERS a rozšířené vstupní pole SELECT-OPTIONS, které umožňuje zadat seznam konkrétních hodnot, nebo výhradně jenom interval hodnot. V rámci druhé možnosti se hodnoty ukládají do takzvaných range tabulek. SAP je tímto typem tabulky zcela unikátní, jelikož žádné jiné programovací jazyky toto neumějí a neumožňují. Range tabulka je standardní tabulka se standardním klíčem a speciálně strukturovaným typem řádku. Využívá se hlavně pro použití v obecném logickém výrazu s relačním výrazem IN. Tento operand umí prohledávat hodnoty v rámci tabulky hodnot a využívá se hlavně v SQL dotazech. Tahle tabulka má pouze čtyři sloupce SIGN, OPTION, LOW a HIGH. Na základě těchto čtyř atributů umí tato tabulka uchovávat intervaly nebo seznamy hodnot. Na základě této vlastnosti byly implementovány některá vstupní pole na vstupní obrazovce aplikace. Možnost vyplňovat pouze jenom seskupené vstupní pole na vstupní obrazovce je implementováno pomocí ABAP příkazu LOOP AT SCREEN, která umožňuje jednotlivé prvky na vstupní obrazovce postupně procházet. Při procházení jednotlivých prvků se poté odchytní konkrétní přiřazené skupiny. Tyto skupiny jsou definované ve vlastnostech grafického prvku. Grafické prvky je možné poté v průběhu výběru dynamicky měnit. Pro požadovanou funkcionalitu skrývání jednotlivých grafických prvků byla využita vlastnost deaktivování, jež znemožní vyplnit příslušné vstupní pole. Tato implementační logika je nutná již v průběhu výběru, proto se využívá odchytní událostí v průběhu tohoto výběru. A to pomocí příkazu AT SELECTION-SCREEN OUTPUT, která nám právě umožňuje zachycení těchto vyvolaných událostí na dané obrazovce.

Pošta tisk adresních štítků

Účetní okruh

Účetní okruh

Vyhledání adresy

Hledat Do

Odběratel (zákazník)

Odběratel Do

Dodavatel

Dodavat. Do

Zaměstnanec

Osobní číslo 0 Do 0

Druh adresy 1

Obrázek 5.3: Výběrová obrazovka.

Samotná vstupní obrazovka se skládá z několika polí, která očekávají vstupní data. Jednotlivé pole slouží k účelům, jež jsou detailně popsány níže.

Vstupní výběrová obrazovka

- Účetní okruh: Výběr účetního okruhu, který slouží k výběru správné adresy hlavní pobočky jakožto adresa odesílatele na adresních štítcích.
- Hledat: Umožňuje vyhledávat skrze všechny tabulky a atributy tabulek (např. příjmení).
- Odběratel(zákazník): Výběr konkrétních čísel odběratelů (zákazníků).
- Dodavatel: Výběr konkrétních čísel dodavatelů.
- Zaměstnanec: Výběr konkrétních čísel zaměstnanců a to včetně druhu adresy.
- Osobní číslo
- Druh adresy

5.4 Získávání dat

Na základě vstupních dat na vyplněné vstupní obrazovce jsou data předána do třídy ZCL_ADDRESS_LABEL, která řídí celou logiku tisku. Pomocí této třídy je v první řadě volána metoda pro získání dat GET_DATA. Tato metoda umožňuje vyhledat příslušné adresy v konkrétních databázích na základě preference uživatele. Hledané adresy nejsou uloženy výhradně v jedné databázové tabulce, proto je třeba rozlišovat, které typy adres hledáme a v jakých tabulkách. Selekcce hledaných adres je proto rozdělena na tři následující části.

- Databázové tabulky pro vyhledávání zákazníků
 - ADRC - adresa (centrální správa adres)
 - KNA1 - kmenový soubor zákazníků (všeobecná část)

```
IF v_kunnr_p = abap_true.
```

```
SELECT *
  APPENDING CORRESPONDING FIELDS OF TABLE t_address
FROM adrc INNER JOIN kna1 ON ( adrc~addrnumber = kna1~adrnr )
WHERE adrc~sort1 IN t_sort1 AND
      kna1~kunnr IN t_kunnr AND
      adrc~date_to >= sy-datum.

SORT t_address BY kunnr.
```

Obrázek 5.4: Implementace vyhledávání zákazníků.

V implementaci byl využit speciální příkaz CORRESPONDING FIELDS OF TABLE, který je pro jazyk ABAP zcela charakteristický. Tento příkaz umožňuje uložit všechny možné hodnoty atributů záznamu databázové tabulky do stejnojmenných atributů struktury vybrané tabulky. Příkaz APPENIDNG značí, že tento záznam v podobě struktury

tabulky bude přidán na konec tabulky, a to jako nový záznam tabulky. To znamená, že tabulka nebude celá přepisována. Kromě správného propojení tabulek a vyhledávání na základě parametrů bylo nutné pohlídat, zda jsou tato získaná data aktuálně platná. Tuto podmínku zajistila kontrola koncového data platnosti adresy. Za tímto účelem byla využita systémová globální proměnná aktuálního datumu.

- Databázové tabulky pro vyhledávání dodavatelů
 - ADRC - adresa (centrální správa adres)
 - LFA1 - kmenový soubor dodavatelů (všeobecná část)

```
ELSEIF v_lifnr_p = abap_true.
```

```
SELECT *
  APPENDING CORRESPONDING FIELDS OF TABLE t_address
FROM adrc INNER JOIN lfa1 ON ( adrc~addrnumber = lfa1~adrnr )
WHERE adrc~sort1 IN t_sort1 AND
      lfa1~lifnr IN t_lifnr AND
      adrc~date_to >= sy-datum.

SORT t_address BY lifnr.
```

Obrázek 5.5: Implementace vyhledávání dodavatelů.

- Databázové tabulky pro vyhledávání zaměstnanců
 - PA0006 - Personální kmenový záznam Infotyp 0006 (Adresy)
 - PA0002 - Personální kmenový záznam Infoty 0002 (Data k osobě)

```
ELSEIF v_pernr_p = abap_true.
```

```
SELECT DISTINCT pa0006~pernr pa0006~name2 AS name1 ort01 AS city1 PSTLZ
  AS post_code1 stras AS street hsnmr AS house_num1 land1 AS country
  APPENDING CORRESPONDING FIELDS OF TABLE t_address
FROM pa0006 INNER JOIN pa0002 ON ( pa0006~pernr = pa0002~pernr )
WHERE pa0006~pernr IN t_pernr AND
      ( pa0002~nchmc IN t_sort1 OR pa0002~vnamc IN t_sort1 ) AND
      pa0006~anssa = t_anssa AND
      pa0006~endda >= sy-datum.

SORT t_address BY pernr.
```

Obrázek 5.6: Implementace vyhledávání zaměstnanců.

U vyhledávání adres pro zaměstnance byl problém, že názvy atributů měli odlišný název atributů než u zavedené struktury. Z toho důvodu bylo nutné tyto atributy správně

přejmenovat na názvy atributů struktury adres ZPDF_PODACI_LISTKY_S. Příkazová funkce SORT ve všech případech zajišťuje seřazení vyhledaných záznamů podle zvoleného kritéria a to pomocí pomocného příkazu BY. Jednotlivé tabulky je možné následně vidět v transakci SE11.

5.5 Dočasné tabulky

Dočasné tabulky (temporal tables) jsou v SAP terminologii nazývány jako interní tabulky. Interní tabulky existují jen za běhu spuštěného ABAP programu a používají se převážně při zpracování velkých objemů dat. Tato tabulka se skládá ze struktury a klíče, které jsou totožné u každého záznamu tabulky. Vyhledávání v interních tabulkách lze provádět pomocí indexu nebo klíče. V programech se tyto tabulky využívají pro ukládání dočasných dat, jež jsou následně různými způsoby zpracovávány. Jelikož tato tabulka je k dispozici pouze za běhu programu, tak její velikost je dynamicky měněna a průběžně ukládána do vyrovnávací paměti. Z tohoto pohledu je interní tabulka velmi univerzální a proto i často využívaná. Během implementace byly tyto tabulky hojně využity. Hlavně pro výsledky získaných dat a pro zobrazení zpracovaných dat v SALV.

5.6 Zobrazení dat

Standardní variantou zobrazování dat v SAP systému je zobrazení pomocí tabulky ALV. Jelikož bylo v návrhu navrženo OOP řízení tisku, tak bylo zvoleno využití objektové varianty SALV, která byla popsána v kapitole 2. Při implementaci byla vytvořena SALV komponenta se vstupními daty pro hlavní obrazovku s číslem 1000. Následně byla SALV komponenta upravena z hlediska zobrazovaných sloupců a jejich názvů. Byla přidána funkcionality odlišných barev řádků za sebou a byla zapnuta funkce pro vícenásobný výběr řádků. V poslední řadě bylo do SALV komponenty implementovány pomocné nástroje pro řazení řádků, filtrování dat a možnost uživatelské úpravy celé SALV tabulky. Naimplementovanou SALV tabulku lze vidět na obrázku 5.7.

Pošta - adresy Odběratel (zákazník)

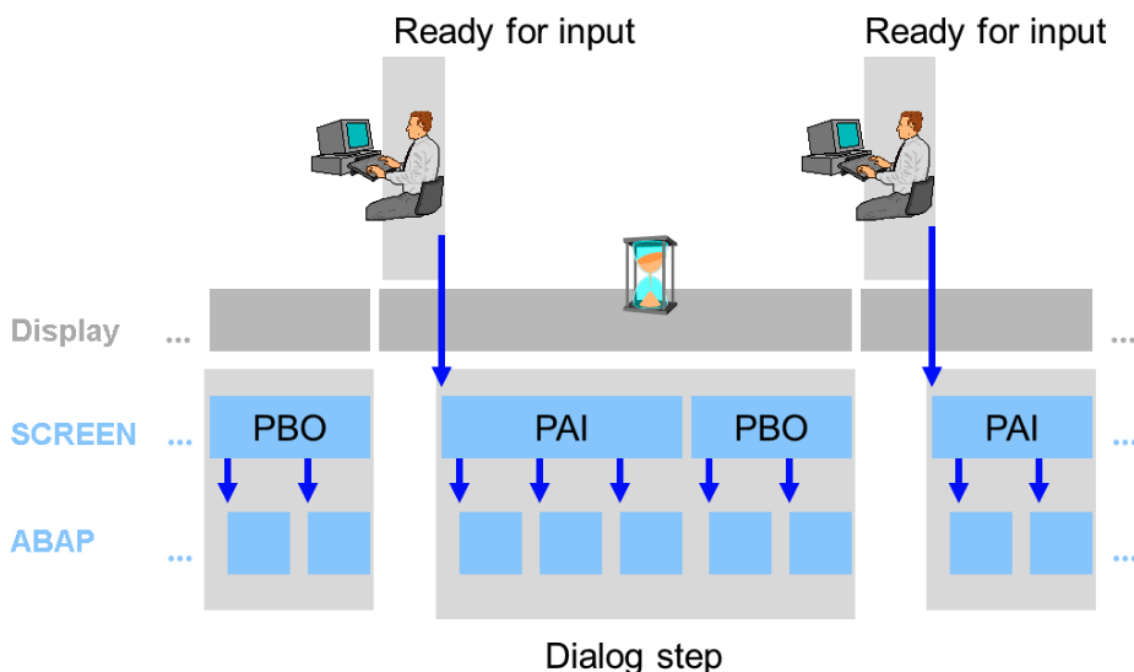
Tisk adresních štítků A4 Tisk adresních štítků Tisk podacích listků Tisk podacích listků Tisk podacích listků Tisk podacích listků

Odběratel	Jméno	Jméno 2	Místo	PSČ	Ulice	Č.domu	Stát
100000	SAP ČR spol. s r. o.	druhy radek k SAP CR - test	Praha 5	155 00	Pekařská	621/10	CZ
100004	SAP Hungary Kft.		Budapest	1123	Csörsz, MOM PARK, Gellért Irodaház	41	HU
100275	SAP Romania		praha	140 00	Calea Moșilor	51	CZ
100325	SAP Slovensko		Bratislava 5	850 00	Kutlíkova	17	SK
100326	SAP España		Madrid	28020	Plaza Pablo Ruiz Picasso Planta	4	ES
100395	SAP Deutschland AG & Co. KG	SAP Německo	Waldorf	69190	Neurottstrasse	15a	DE
100417	SAP Österreich GmbH		Wien	1021	Lassallestr.	7b	AT
100546	SAP Polska Sp. z o.o.		Warsaw	02-672	ul. Domaniewska	41	PL
100875	SAP ČR, spol. s r.o., Brno		Brno	639 00	Vídeňská	89a	CZ

Obrázek 5.7: SALV s daty nalezených výsledků.

5.7 PBO a PAI

V rámci implementace výběrové obrazovky a SALV bylo využito grafické prostředí Dynpro. Toto prostředí v ABAP programu umožňuje dynamicky měnit logiku před zahájením dané obrazovky (Process Before Output) a po akci uživatele (Process After Input). Toto rozhraní je definováno pro každou obrazovku Dynpra. To znamená, že každá definovaná obrazovka může mít odlišnou logickou hierarchii před zahájením a zároveň i po akci uživatele. Ve stručnosti se jedná o bloky kódu, které jsou volané definovanou událostí obrazovky. PBO je tedy událost zpracování, která je volána před zobrazením obrazovky. Mezi úkoly PBO patří inicializace polí obrazovky a dodání výchozích hodnot dat pro obrazovku. PAI je druhá událost zpracování, ke které dojde po opuštění obrazovky nebo po uživatelské interakci. Princip chování PBO a PAI v systému SAP je zobrazeno na obrázku 5.8.



Obrázek 5.8: Logika PBO a PAI v systému SAP[1].

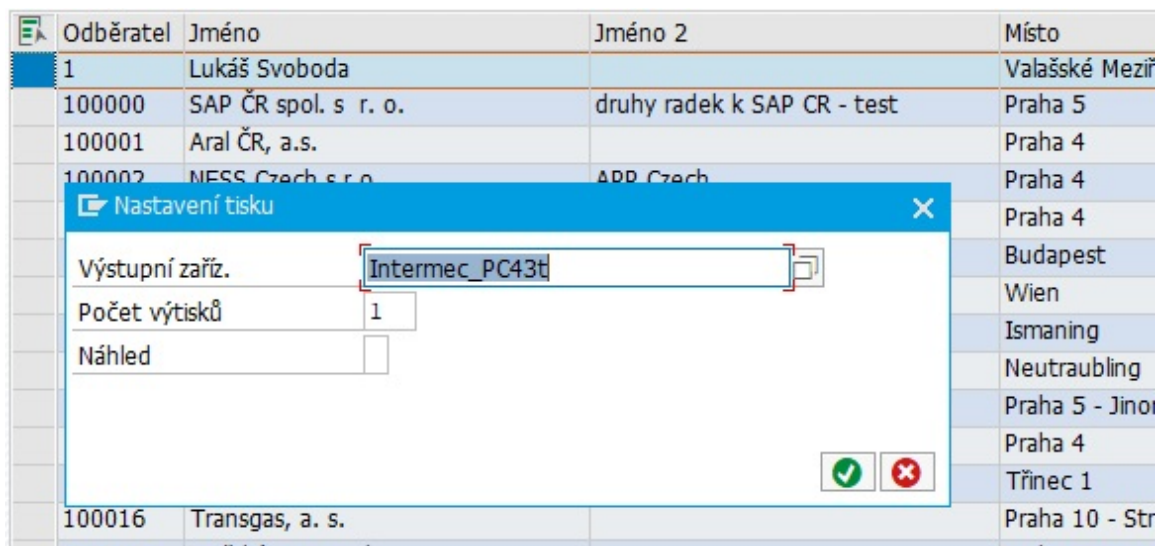
Byl zde zmíněn obrázek 5.7, který právě využívá obsluhu programu pomocí těchto událostí oken Dynpra. Všechny grafické prvky, které jsou zobrazeny na obrázku 5.7, jsou nutné zvláště implementovat. Pro implementaci tlačítek v horní liště bylo využito SAP GUI Status. SAP GUI Status je nezávislá součást programu ABAP, která umožňuje vytvářet grafické uživatelského rozhraní pro programy. Existují tři různé typy grafických uživatelských rozhraní:

- Status dialogu
- Dialogové okno
- Kontextová nabídka

Tlačítka byla tedy implementovaná pomocí status dialogu. Založení SAP GUI statusu proběhlo za pomoci transakce SE41. Následně byly vytvořeny grafické prvky v podobě tlačítek, včetně nastavení volajících funkcí a přiřazení ovládacích tlačítek klávesnice. Po vytvoření grafických prvků a nastavení jejich volání v programu bylo definováno v PAI samotné odhycení volání. A následně je zpracováván daný požadavek akce. Do horní lišty byly implementovány čtyři tlačítka. Prvním tlačítkem je tisk adresních štítků, jež spustí přípravu dat a vyvolá dialogové okno s možností tisku, které po potvrzení provede tisk adres přímo na štítky v tiskárně Intermecc PC43t. Druhým tlačítkem je tisk adresních štítků A4, jež umožní vygenerování 21 etiket ve formátu A4 a následně je zobrazí v SAP systému v podobě PDF. Předposledním tlačítkem je tisk podacích lístků, který spustí generování podacích lístků prostřednictvím SAP Adobe Form a následně je zobrazí v SAP systému opět v podobě PDF. Posledním tlačítkem je tisk podacího archu, jež vygeneruje a zobrazí podací arch s vybranými adresami. Všechna tato tlačítka byla vytvořena pouze s názvem akce bez ikony. Toto bylo zvoleno z důvodu přehlednosti a omezených systémových ikon, které by špatně vystihovaly danou operaci tlačítek.

5.8 Potvrzení tisku s počtem kopií

Tato funkcionality mohla být řešena dvěma způsoby. A to buď přes novou obrazovku Dynpra, nebo pomocí funkčního modulu. Nakonec byla zvolena při implementaci druhá varianta, a to pomocí funkčního modulu POP_UP_GET_VALUES. Toto rozhodnutí bylo zvoleno na základě jednodušší implementace v rámci této aplikace. V rámci této implementace bylo nutné zvážit vhodné parametry, které budou sloužit jako vstup na dialogovém okně. Nakonec se rozhodlo ke zvolení tří vstupních hodnot. A to výběr tiskárny, počet kopií tisku a náhled tisku. Tyto parametry musely být vytvořeny jako speciální struktura a uloženy do interní tabulky, která byla předána do volaného funkčního modelu pro dialogové okno. Tento funkční modul po zavolání zpětně vrací uživatelem zadané hodnoty. Kromě toho vrací i zpětnou reakci zvolených ovládacích tlačítek dialogového okna. Toto dialogové okno lze vidět na obrázku 5.9.



Obrázek 5.9: Dialogové okno pro potvrzení tisku štítků.

5.9 Formuláře

V této podkapitole je detailně popsána implementace jednotlivých typů navrhovaných formulářů a štítků, které byly navrženy v kapitole 4.

5.9.1 Implementace štítků pro tiskárnu Intermec PC43t

Pro implementaci tohoto štítku bylo potřeba založit nový typ formuláře. Kvůli komunikaci s tiskárnou Intermec PC43t byla zvolena možnost skrze SAP Script formulář, který je schopen odesílat příkazy v jazyce tiskárny. Konkrétně se tedy jedná o jazyk ESim. Pro založení tohoto typu formuláře byla využita transakce SE71. V této transakci byl následně založen formulář s názvem ZBAR_F_ADDRESS_L. V SAP Script formulářích bylo nutné definovat hlavičku a několik dalších částí formuláře. Nejprve se provedlo nadefinování hlavního okna o šířce okna 150 mm a výšce 50 mm. Toto okno bylo nazváno jako MAIN, jelikož se jedná o jediné okno v tomto formuláři. Tyto parametry odpovídají rozměrům štítků ve fyzické podobě. Dále byl definován formát odstavce DF. DF je zkratka pro název Default. Tento název byl zvolen z důvodu využití opět jediného typu odstavce. Komunikace s tiskárnou probíhá skrze ESim jazyk, a tak nebylo nutné definovat speciální další odstavce. Definice alespoň jednoho odstavce byla povinná, protože bez typu odstavce se nedalo zaktivovat formulář. Před samotnou implementací SAP Scriptu bylo vytvořena stránka formuláře. Tato stránka byla pojmenována jako PAGE1 a byla přiřazena v hlavičce formuláře jako prvotní strana po spuštění.

```
/E LABEL
* N
* ZT
* I8,B,001
* A780,400,2,2,2,2,N,"&GS_ADDRESS-NAME1&"
* A780,340,2,2,2,2,N,"&GS_ADDRESS-NAME2&"
* A780,280,2,2,2,2,N,"&GS_ADDRESS-STREET&"
* A780,220,2,2,2,2,N,"&GS_ADDRESS-CITY1&"
* A780,160,2,2,2,2,N,"&GS_ADDRESS-POST_CODE1&"
* A780,100,2,2,2,2,N,"&GS_ADDRESS-COUNTRY&"
* P1
```

Obrázek 5.10: Implementace SAP Script formuláře pro tiskárnu Intermec PC43t.

Za použití příručky ESIM jazyka, která je dostupná zde [12], byla vytvořena implementace formuláře v SAP Scriptu. Tato implementace je k vidění na obrázku 5.10. Pomocí příkazu \E v implementaci SAP Scriptu bylo vytvořeno konkrétní identifikační název elementu pro SAP Script. Na základě tohoto názvu, který byl pojmenován jako LABEL, je volán zápis těchto příkazů do formuláře. Následně i jeho tisk, jakmile dojde k ukončení otevřeného formuláře pomocí funkčního modulu FORM_CLOSE. Před samotným voláním tisku bylo nutné provést napojení hodnot do globálních proměnných v podobě struktury, které byly využity v tomto SAP Script formuláři. Tato funkce nastavení hodnot byla zajištěna pomocí funkce SET_GS_ADDRESS, která je volána před tímto zmíněným procesem tisku.

5.9.2 Implementace formuláře podacího lístku

Implementace byla zvolena pomocí SAP Adobe formuláře, jelikož byl k dispozici PDF formát originálního formuláře. Tento formulář byl volně k dispozici na stránkách České pošty. V transakci SFP byl založen formulář a jeho rozhraní se stejnojmenným názvem Z_PDF_PODACI_LISTEK. Na základě tohoto zmíněného PDF formuláře podacích lístků byl pak tento formulář importován do tohoto SAP Adobe formuláře. Import formuláře nebyl zcela dokonalý, a tak byly provedeny ruční úpravy vizuální stránky formuláře. Před zahájením napojování jednotlivých dat a nastavení chování formuláře při tisku bylo nutné nadefinovat a nastavit rozhraní vytvořeného formuláře. Definici a popis rozhraní lze vidět níže.

Rozhraní:

Import

Název parametru	Typizace	Označení typu	Nepovinné	Před. hodn.	Navrh. hodn.
/IBCWB/DOCPRARAMS	TYPE	SFPDOCPARAMS	X	X	
IS_COMPANY	TYPE	ZPDF_PODACI_LISTKY_S		X	
IT_ADDRESS	TYPE	ZPDF_PODACI_LISTKY_T		X	

Tabulka 5.1: Import v rozhraní Z_PDF_PODACI_LISTEK.

Export

Název parametru	Typizace	Označení typu	Před. hodn.
/IBCWB/FORMOUTPUT	TYPE	FIFORMOUT	X

Tabulka 5.2: Export v rozhraní Z_PDF_PODACI_LISTEK.

Globální definice

Název parametru	Typizace	Označení typu	Navrh. hodn.	Konstanta
GS_COMPANY	TYPE	ZPDF_PODACI_LISTKY_S		

Tabulka 5.3: Globální data v rozhraní Z_PDF_PODACI_LISTEK.

Inicializace

```
gs_company = is_company.  
CONCATENATE gs_company-street gs_company-house_num1  
INTO gs_company-street SEPARATED BY space.
```

Obrázek 5.11: Implementace inicializace v rozhraní Z_PDF_PODACI_LISTEK.

Jako vstupní parametr rozhraní byla zvolena struktura IS_COMPANY s obsahem informací ohledně firmy jako odesílatele a tabulka dat jednotlivých adres IT_ADDRESS. Struktura s informacemi ohledně firmy je v podobě struktury adres, která byla navržena a ukázána v předešlé kapitole. Tabulka s jednotlivými adresami využívá vlastní strukturu. Jedná se o strukturu, jež obsahuje další 4 struktury jako její atributy. Tyto 4 struktury vycházejí ze zmíněné základní adresní struktury. Tato implementace struktury nám zajistí uložení maximálně 4 adresních dat. Toto implementační rozhodnutí bylo zvoleno za cílem dosáhnout v tabulce IT_ADDRESS, že každý záznam v tabulce bude představovat jeden A4 formát podacích lístků. Čtyři adresní struktury tedy představují v originálním formuláři 4 podací lístky, které jsou právě v jednom A4 formátu podacích lístků. Díky tomu, že tato tabulka obsahovala záznamy adresních dat pro každý jeden A4 list, tak bylo možné využít funkcionality SAP Adobe formuláře, jež umožňuje na základě počtu záznamů v tabulce generovat další totožné formuláře za sebou. Tímto způsobem bylo tedy umožněno generovat najednou i větší počty podacích lístků jedním procesem generování formuláře. Následně byla vstupní data rozhraní napojena na příslušné pole pro správný tisk jednotlivých polí. V neposlední řadě byla nastavena maximální délka polí a jejich viditelnost s dalšími vlastnostmi polí formuláře. Po ověření všech propojení polí s daty byl formulář úspěšně zaktivován.

Při implementaci přípravy dat bylo zapotřebí vytvořit příslušný funkční modul pro generování tohoto formuláře. Za tímto účelem je volán funkční modul FP_FUNCTION_MODULE_NAME, který vytvoří nový funkční modul s rozhraním vybraného formuláře. Před zavoláním tohoto vytvořeného funkčního modulu jsou adresní data upravena. Konkrétně je upravován název ulice, jež je spojován s číslem popisným. V případě, že adresa je velmi dlouhá, je provedeno zkrácení adresy při tomto spojování. V poslední části je volán funkční modul formuláře s parametry adresních dat a celý proces je spuštěn jako job. Vytvořené PDF je následně převedeno do binární podoby a je distribuováno do HTML kontejneru v novém okně Dynpra. V tomto okně je pak zobrazeno nově vygenerované PDF, které lze tisknout na lokálních tiskárnách.

5.9.3 Implementace formuláře pro 21 adresních štítků

Implementace byla opět zvolena pomocí SAP Adobe formuláře. Tentokrát z důvodu využití práce s tiskem ve formátu A4 a funkcionalitou generování formuláře pomocí jednoho procesu. Digitální podoba těchto štítků ve formátu papíru A4 a formátu souboru PDF nebyla nikde k dispozici, proto musel být formulář vytvořen ručně. V transakci SFP byl založen další formulář i jeho formulářové rozhraní. Oba formuláře byly pojmenovány Z_PDF_ETIKET. Po založení bylo v tomto formuláři implementováno 21 etiket na formátu papíru A4, kde každá adresní etiketa měla rozměry 63,5 × 38,1 mm. Před zahájením napojování jednotlivých dat a nastavení chování formuláře při tisku, bylo nutné nadefinovat a nastavit rozhraní vytvořeného formuláře. Definicí a popis rozhraní lze vidět níže.

Rozhraní:

Import

Název parametru	Typizace	Označení typu	Nepovinné	Před. hodn.	Navrh. hodn.
/1BCWB/DOCPARAMS	TYPE	SFPDOCPARAMS	X	X	
IT_PDF_ETIKET	TYPE	ZPDF_ETIKET_T			

Tabulka 5.4: Import v rozhraní Z_PDF_ETIKET.

Export

Název parametru	Typizace	Označení typu	Před. hodn.
/1BCWB/FORMOUTPUT	TYPE	FPFORMOUT	X

Tabulka 5.5: Export v rozhraní Z_PDF_ETIKET.

Vstupním parametrem tohoto formuláře byla zvolena tabulka IT_PDF_ETIKET. Tato tabulka se skládá ze struktury ZPDF_ETIKET_S, která obsahuje 21 atributů. Každý tento atribut byl definován jako typ struktury ZPDF_PODACI_LISTKY_S. Během implementace této tabulky bylo zamýšleno využití každého záznamu tabulky jako celého souboru dat pro tento jeden formulář. Opět zde bylo umožněno generovat najednou i větší počty tohoto formuláře jedním procesem generování formuláře. V následující fázích implementace byla vstupní data rozhraní napojena na jednotlivé pole formuláře. Byly nastaveny maximální délky polí a jejich další vlastnosti. Po ověření nastavených propojení dat ve formuláři byl formulář úspěšně zaktivován.

Implementace zpracování dat vyžadovalo vhodně vyplnit data do struktury tabulky před samotným odesláním do rozhraní formuláře. Za tímto účelem byla implementována logika zpracování dat, která umí vhodně rozdělit data po 21 adresních datech a tato data vkládat jako jednu velkou strukturu v podobě záznamu tabulky. Kromě toho je prováděno ošetření délky některých adresních informací, jelikož podací lístky mají omezenou velikost vstupních polí. Poté je volán funkční modul formuláře s parametry adresních dat a proces je převeden do jednoho jobu. Vytvořené PDF je následně převedeno do binární podoby. V této datové podobě je pak předán do HTML kontejneru, který běží v novém okně Dynpra. V tomto kontejneru jsou pak data zobrazována jako PDF formát, jenž lze tisknout na klasické lokální tiskárně.

5.9.4 Implementace formuláře podacího archu

V tomto posledním formuláři bylo opět zvoleno využití SAP Adobe formuláře. Toto rozhodnutí bylo opět postaveno na tom, že tento formulář existuje již ve formátu PDF, které je volně ke stažení ze stránek České pošty. Na základě existence tohoto formuláře bylo nejlepší využít právě SAP Adobe formulář a právě tento stažený formulář importovat do SAP systému. I přesto, že formulář bylo možné importovat, tak import nebyl zcela zdařen a muselo být provedeno hodně vizuálních změn. Nejprve bylo pomocí transakce SFP provedeno založení formuláře a jeho formulářové rozhraní. Tyto dva objekty byly pojmenovány Z_PDF_PODACI_ARCH. Arch obsahuje 10 možných adresních polí k vyplnění. V případě více vybraných adresních dat by nastal problém tisku. Z tohoto důvodu byl SAP Adobe

formulář nastaven pro generování tohoto formuláře pro každý záznam v tabulce, která byla nainportována rozhraním tohoto formuláře. Díky tomuto nastavení byla vytvořena vhodná struktura tabulky, jež obsahuje 10 struktur adresních dat. Díky této definici potřebných tabulek a struktur bylo nadefinováno a nastaveno rozhraní vytvořeného formuláře. Definici a popis rozhraní lze vidět níže.

Rozhraní:

Import

Název parametru	Typizace	Označení typu	Nepovinné	Před. hodn.	Navrh. hodn.
/1BCWB/DOCPRARAMS	TYPE	SFPDOCPARAMS	X	X	
IS_COMPANY	TYPE	ZPDF_PODACI_LISTKY_S		X	
IT_PDF_PODACI_ARCH	TYPE	ZPDF_PODACI_ARCH_T		X	

Tabulka 5.6: Import v rozhraní Z_PDF_PODACI_ARCH.

Export

Název parametru	Typizace	Označení typu	Před. hodn.
/1BCWB/FORMOUTPUT	TYPE	FPFORMOUT	X

Tabulka 5.7: Export v rozhraní Z_PDF_PODACI_ARCH.

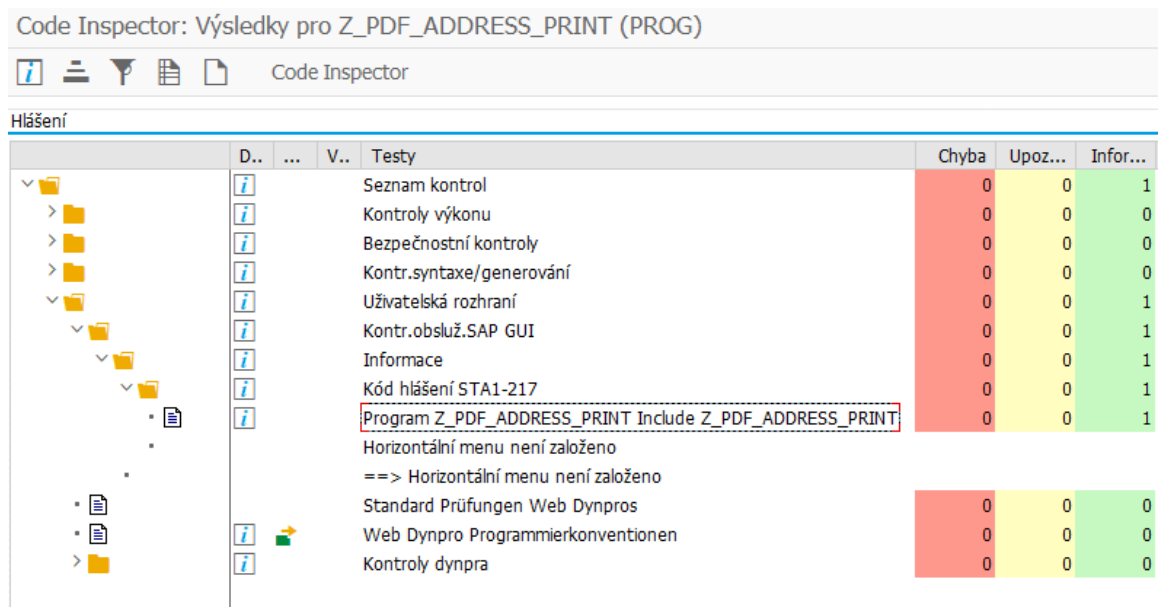
Rozhraní tohoto formuláře bylo vytvořeno z tabulky IT_PDF_PODACI_ARCH, která obsahuje všechna vybraná adresní data uživatele. Dále se skládá ze struktury IS_COMPANY, jež nese informace ohledně odesílatele. Odesílatel byl získán na základě výběru účetního okruhu při vyhledávání dat. Generování více počtů tohoto formuláře bylo zde opět vyřešeno jedním procesem. Vstupní data rozhraní byla napojena na jednotlivé pole formuláře. Následně byly nastaveny všechny požadované vlastnosti polí formuláře. Nakonec byla data zkontrolována a formulář byl aktivován.

Formátování dat a jejich příprava pro rozhraní formuláře byla připravena v podobné implementaci jako předchozí formulář. Pouze se využívají jiné tabulky a počet adresních struktur. Jinak je opět volán funkční modul formuláře s parametry adresních dat a strukturou odesílatele. Vytvořené PDF je převedeno do binární podoby a je odesláno do HTML kontejneru, kde je zobrazeno jako PDF formát. Opět lze z tohoto prostředí provést tisk na klasické lokální tiskárně.

5.10 Code inspector

Inspektor kódu je nástroj pro kontrolu objektů. Slouží tedy hlavně pro kontrolu správnosti zdrojového kódu. Kromě toho pomocí nástroje Code Inspector můžeme kontrolovat výkon, zabezpečení, syntaxi a dodržování konvencí názvů jednotlivých objektů nebo sad objektů. V inspektoru kódu můžeme definovat kontroly, které pomocí kontrolních variant prozkoumají určité sady objektů. Po zadání kontroly obdržíme informační zprávy, varovné zprávy nebo chybové zprávy o různých vlastnostech zkoumaných objektů. Chybové zprávy (Errory) jsou zprávy, které nám oznamují velmi důležitou chybu. Tato chyba může způsobit pád programu po spuštění aplikace. Dále existují zde varovné zprávy (Warningy). To jsou zprávy oznamující chybu v kódu, jež nezpůsobí pád během běhu programu, ale může například označovat zastaralé příkazy jazyku ABAP nebo nevhodné prohledávání databáze. Posledními zprávami jsou takzvané informační zprávy, které není zcela nutné řešit, jelikož nám neohroží běh aplikace. Slouží čistě jako dodatečně pomocné informace během vývoje[8]. Tento nástroj posloužil tedy během implementace za účelem kontroly správnosti kódu během vývoje a k bezproblémovému nasazení do produkce. Výsledek z tohoto nástroje lze vidět na obrázku 5.12.

- Varianta kontroly (Check Variant): Definuje pravidla, která se mají použít. Dále se zde určuje jaké kontroly mají být provedeny a nastavení těchto kontrol.
- Sada objektů (Object Set): Definuje vývojové objekty, které budou zahrnuty.
- Inspekce: Definuje kombinaci varianty kontroly a sady objektů. Popisuje jaké kontroly mají být použity a to na vybrané konkrétní vývojové objekty.



Code Inspector: Výsledky pro Z_PDF_ADDRESS_PRINT (PROG)						
Hlášení						
D..	...	V..	Testy	Chyba	Upoz...	Infor...
>			Seznam kontrol	0	0	1
>			Kontroly výkonu	0	0	0
>			Bezpečnostní kontroly	0	0	0
>			Kontr.syntaxe/generování	0	0	0
>			Uživatelská rozhraní	0	0	1
>			Kontr.obsluž.SAP GUI	0	0	1
>			Informace	0	0	1
>			Kód hlášení STA1-217	0	0	1
>			Program Z_PDF_ADDRESS_PRINT Include Z_PDF_ADDRESS_PRINT	0	0	1
			Horizontální menu není založeno			
			==> Horizontální menu není založeno			
			Standard Prüfungen Web Dynpros	0	0	0
			Web Dynpro Programmierkonventionen	0	0	0
			Kontroly dynpra	0	0	0

Obrázek 5.12: Kontrola kódu pomocí Code Inspector.

5.11 Konfigurace tiskárny Intermec PC43t

Tiskárny značky Intermec jsou zcela certifikované skrze SAP Printer Vendor Program. Integrované čárové kódy v prostředí SAP jsou optimalizované se všemi zařízeními Intermec. Konkrétně pro SAP Smart Forms a mySAP Business Suite. Společnost Intermec také nadále podporuje tisk štítků s čárovými kódy ze SAP Script v SAP R/3, tisk s podporou XML v SAP Auto-ID infrastruktúře a v poslední řadě návrh a tisk štítků třetích stran.

Pro správnou funkčnost tiskárny byla nastavena statická IP adresa na straně DHCP. Toto nastavení je řízeno na základě MAC adresy tiskárny. Tím bylo zajištěno, že tiskárna bude přístupna stále z jedné IP adresy. Tohle omezení bylo nutné provést, jelikož následující propojení tiskárny se systémem SAP vyžaduje konfiguraci se statickou IP adresou, a to ze strany systému SAP. Kromě toho byly před samotnou konfigurací SAP systému nainstalovány ovladače tiskárny pro SAP. V systému SAP se typ zařízení skládá ze všech atributů definovaných pro výstupní zařízení. Pro správný přístup musí být tyto informace nastaveny v systému SAP. Tyto ovladače pokrývají aspekty, jako jsou ovládací příkazy pro výběr písma, velikost stránky, výběr znakové sady, použitá znaková sada a další.

Pomocí transakce SPAD v sekci celá administrace a v záložce typ zařízení byl definován nový typ zařízení a jeho konfigurace. Název nového typu zařízení byl zvolen ZASCIIPR. Tento typ byl tedy založen a následně implementován pro tiskárnu Intermec s českou znakovou sadou. Tato implementace typu zařízení umožnila správný tisk českých názvů firem a jmen zaměstnanců. Typ tiskárny lze vidět na obrázku 5.13.

The screenshot shows the SAP SPAD configuration interface for a device type. The title is 'Správa spoolu: Typ zařízení (Zobrazení)'. Below the title are icons for editing, sharing, and deleting, followed by a 'Formáty' button. The main configuration area is divided into two tabs: 'Vlastnosti' (selected) and 'Řízení tisku'. Under 'Vlastnosti', there are several sections:

- Verze:** 7
- Zpracování pomocí SAPscriptu:**
 - Ovladač:** Line printer driver II
 - Stránková tiskárna
- Zpracování seznamů:**
 - Ovladač tiskárny:** Nepoužívejte ovladač tiskárny pro tisk seznamu ABAP
 - Argument:** (empty field)
- Znaková sada tiskárny:**

1. Znaková sada	1404	Microsoft Windows 1250 for Central Europe
Synt.znaková sada	0	

Obrázek 5.13: Typ zařízení ZASCIIPR.

Po úspěšném založení a implementaci typu zařízení ZASCIIPR byla implementace přesunuta do sekce správy spojení všech zařízení. A to pomocí transakce SPAD. V sekci zařízení a server byl následně založen nový typ výstupního zařízení. Výstupní zařízení bylo pojmenováno Intermecc_PC43t. Pojmenování bylo zvoleno na základě modelové řady požadující tiskárny. Po založení tohoto výstupního zařízení bylo správně nakonfigurováno síťové spojení a další atributy zařízení. Správnou konfiguraci lze vidět přehledně v tabulce 5.8.

Atributy zařízení	
Název atributu	Typ/Hodnota
Typ zařízení	ZASCIIPR
Třída zařízení	Normální tiskárna
Model	Intermec PC43T
Typ zařízení	ZASCIIPR
Spojení hostspoolu	
Název atributu	Typ/Hodnota
Met. příst. k hostspoolu	U: Tisk pomocí prot. Berkeley
Host. tiskárna	LPT1
Router	192.168.1.225
Výstupní atributy	
Název atributu	Typ/Hodnota
Krycí list jazyka SAP	Ne
Krycí list hostspoolu	Netisknout krycí list host-spool
Čítač kopií	Předání čítače kopií na spool. sys. host
Vkládání papíru	Přednast. zásobník

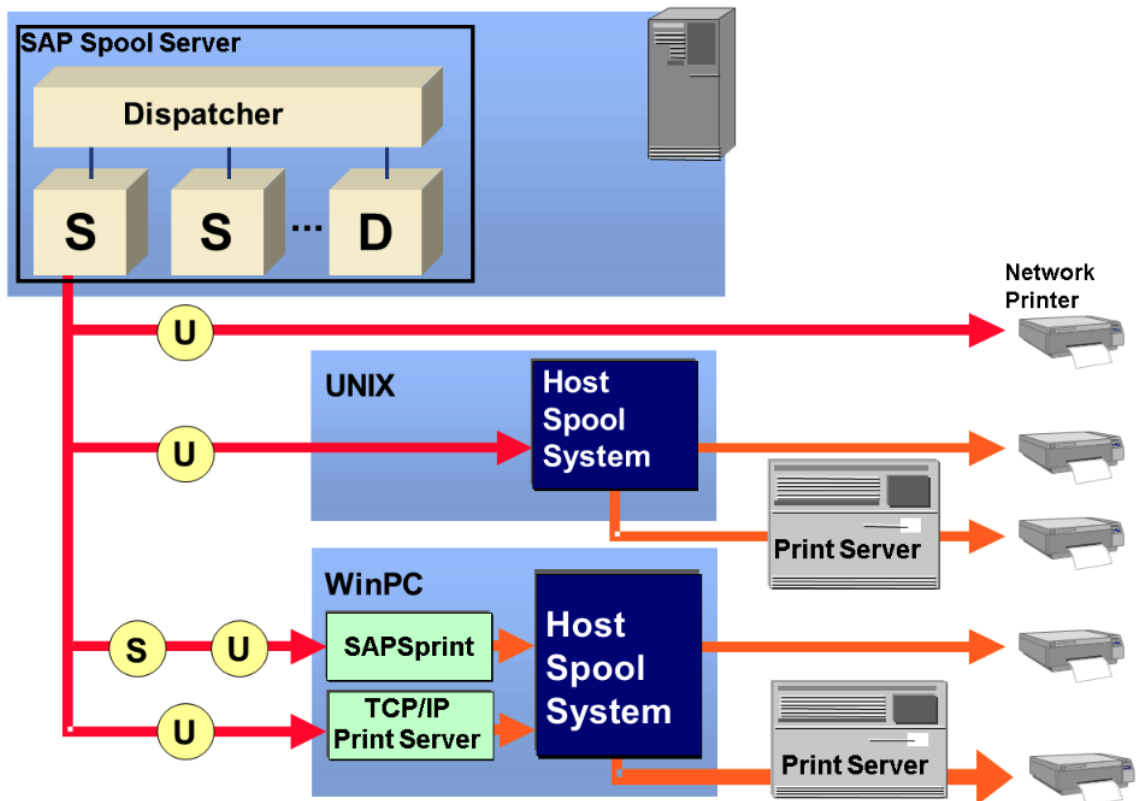
Tabulka 5.8: Konfigurace tiskárny v transakci SPAD pro zařízení Intermecc_PC43t.

Při konfiguraci výstupního zařízení v definici zařízení bylo určeno jakým způsobem je tiskárna připojena k systému SAP. Zvolený způsob je terminologicky nazván jako přístupová metoda. Metoda přístupu určuje například to, zda je použit místní nebo vzdálený tisk. Metoda přístupu je označena písmenem a informuje systém, která architektura je využívána a jaký operační systém je využíván. Tímto způsobem je systém schopen rozpoznat způsob přenosu dat[3].

System používá pro vzdálený tisk následující metody přístupu v závislosti na používaném operačním systému:

- Metoda přístupu U: Systémy UNIX nebo systém Microsoft Windows (pomocí tiskového serveru TCP/IP a služby SAPSprint).
- Metoda přístupu S: Systémy Microsoft Windows (pomocí SAPSprint).

Na obrázku 5.14 je znázorněna architektura vzdáleného tisku, která je zmíněná výše.



Obrázek 5.14: Architektura vzdáleného tisku[3].

Kapitola 6

Testování

Tato kapitola popisuje způsob přenesení aplikace na testovací systém a následně testování propojení tiskárny se SAP systémem, bezproblémové řízení tisku a správný výstup adresních dat v nezměněné podobě oproti náhledu.

6.1 Komunikace s tiskárnou

Před přenosem aplikace na testovací systém bylo velmi zásadní nejprve otestovat síťovou komunikaci SAP systému s nakonfigurovanou tiskárnou. Toto testování lze provést přímo v systému SAP za pomoci transakce ST06, kde v sekci menu detailní analýzy lze nalézt nástroj kontroly LAN připojení pomocí ping funkce. V tomto nástroji byla zadána IP adresa nastavené tiskárny a následně byl odeslán požadavek na otestování spojení. Vzápětí byla vrácena úspěšná zpráva, že je vše správně propojené a nastavené pro správnou komunikaci. Bližší informace ohledně připojení je zobrazeno na obrázku 6.1.

```
LAN Check by PING (Specific IP-Address)
Prezentační server  Aplikační server  Databázový server  Specific IP-Address

10 Pings
ping -a 192.168.1.225 -n 10 -l 500
ping -a 192.168.1.225 -n 10 -l 500

Pinging 192.168.1.225 with 500 bytes of data:

Reply from 192.168.1.225: bytes=500 time=8ms TTL=62
Reply from 192.168.1.225: bytes=500 time=9ms TTL=62
Reply from 192.168.1.225: bytes=500 time=9ms TTL=62
Reply from 192.168.1.225: bytes=500 time=8ms TTL=62
Reply from 192.168.1.225: bytes=500 time=8ms TTL=62
Reply from 192.168.1.225: bytes=500 time=8ms TTL=62
Reply from 192.168.1.225: bytes=500 time=10ms TTL=62
Reply from 192.168.1.225: bytes=500 time=8ms TTL=62
Reply from 192.168.1.225: bytes=500 time=8ms TTL=62
Reply from 192.168.1.225: bytes=500 time=8ms TTL=62

Ping statistics for 192.168.1.225:
Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in mill-seconds:
Minimum = 8ms, Maximum = 10ms, Average = 8ms
```

Obrázek 6.1: Kontrola spojení s tiskárnou Intermec PC43t.

6.2 Přenos na testovací systém

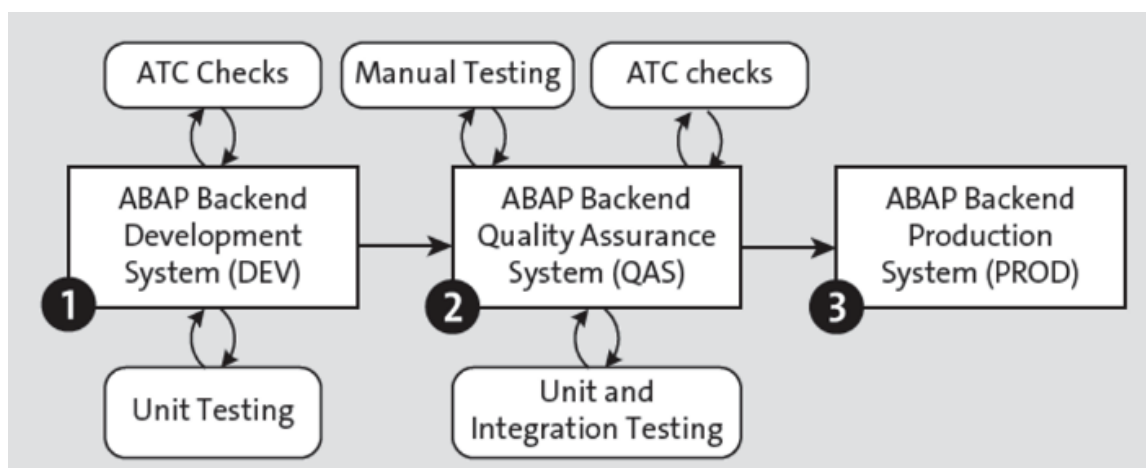
Pro testování s reálnějšími daty zaměstnanců, zákazníků nebo odběratelů bylo zapotřebí aplikaci transportovat na testovací systém. Za tímto účelem bylo nutné v transakci SE10 vybrat náš transportní požadavek a následně tento transportní požadavek uvolnit z fronty. Po úspěšném uvolnění požadavku byl následně odeslán požadavek na testovací systém KT6. Po odeslání bylo třeba provést přihlášení na účet v testovacím systému a požadavek potvrdit v transakci STMS. Poté již byla aplikace zcela přenesena na testovací systém a mohlo proběhnout samotné testování aplikace. Tento proces lze vidět na obrázku 6.2.

6.3 Testování na datech

V rámci spolupráce s firmou KCT data s.r.o jsem mohl vhodně testovat aplikaci přímo ve firmě. Proces testování tedy probíhal s pomocí reálného uživatele nad reálnými testovacími daty. Testované byly všechny možné scénáře a toto testování probíhalo po dobu dvou týdnů. Průběh testování měl podobný scénář, který je zmíněný níže, ale byla zároveň měněna vybraná data, vstupní parametry aplikace a možnosti tisku.

1. Spuštění transakce ZPOSTA
2. Vyplnění účetního okruhu
3. Vyplnění příslušného pole pro vyhledávání adres
4. Výběr konkrétních zobrazených dat pro tisk
5. Zvolení jednoho z možných tisků
6. Zvolení volby tisku (počet kopií, tiskárna)
7. Potvrzení a volání tisku

Výsledek: Tisk štítků/formulářů



Obrázek 6.2: Způsob testování a přenos aplikace[14].

6.4 Chyba tisku během testování

Během testování asi po 14 dnech intenzivního tisku byla objevena chyba, kdy tiskárna přestala tisknout požadavky ze systému SAP. Nejprve byly totožné operace nasimulovány v aplikaci a bylo provedeno důkladné procházení kódu, pomocí nástroje ABAP debugger. A to krok po kroku. Zde nebyla objevena žádná chyba. Požadavek byl správně vytvořen a předán do spoolu systému SAP. Poté tedy byla zkontrolována fronta požadavků ve spoolu systému SAP. Tam bylo zjištěno, že operace byly řádně dokončené a odeslané do tiskárny. Poslední možnou chybou v systému SAP mohla být přepsaná konfigurace připojení tiskárny se systémem SAP. Tato konfigurace nebyla změněna. Nezměněný stav byl zjištěn podle datumu poslední aktivace. Tudíž byla chyba na straně tiskárny. Provedlo se opětovné spuštění tiskárny. Po spuštění měla tiskárna správně nastavenou IP adresu i další parametry, které by mohly mít vliv na neprovedení tisku štítků. Tedy bylo nakonec zjištěno, že samotná aplikace nevykazuje chybu ani propojení samotného systému SAP s tiskárnou. Nakonec bylo provedeno rozebrání tiskárny a bylo zjištěno, že tiskárna má vyčerpaný ribon. Ribon je polyesterová páska, jež je oboustranná. Na jedné straně má nanesenou inkoustovou vrstvu, která je vlivem tepla roztavena přímo na folii štítků. Pokud tento ribon dojde, tak tiskárna pozastaví všechny operace a začne ukládat všechny nové požadavky do své paměti, pokud je stále volná. Na základě tohoto chování byli doplněny instrukce k obsluze této aplikace a byly taktéž s tím seznámeni i ekonomičtí pracovníci. Vše tedy bylo vyřešeno pouze instrukcemi v případě vyskytnutí se podobného problému. Poté již žádné incidenty nenastaly.

6.5 Časová náročnost

V rámci testování bylo vhodné také poukázat, zda tato aplikace opravdu přinesla zlepšení požadovaných činností. A to v oblasti zrychlení prováděných operací v rukou ekonomických pracovníků. Dle průměrného měření bylo naměřeno, že vyplnění jedné adresy zabere pracovníkovi zhruba 18 sekund. Před tímto zápisem, ale musí ekonomický pracovník nejprve danou adresu vyhledat v systému SAP, která zabere průměrně 15 sekund. Dále po vyhledání a zápisu provádí kontrolu, zda zapsal všechny údaje správně. Tahle činnost zabere další 2 sekundy. Dohromady se tedy dostáváme na časovou náročnost 45 sekund na jeden adresní štítek. Při využívání nově vytvořené aplikace bylo měření časové náročnosti měřeno pro rozdílné adresy pro každý štítek. Pokud by nebyly rozdílné, byl by čas konstantní v případě nové aplikace, jelikož u tisku stejných dat lze navolit počet kopií. Z tohoto důvodu bylo měření cíleno pro odlišné adresy, ať porovnání vůči běžnému ručnímu vyplnění je více objektivnější. Výběr vhodných kandidátů lze bez problémů zobrazit do 25 sekund. Dále výběr konkrétní adresy zabere přibližně 5 sekund. S těmito výpočty a daty se nadále počítalo v rámci zpracování grafu náročnosti. Tento graf lze vidět na obrázku 6.3.



Obrázek 6.3: Graf časové náročnosti vyplňování adres za jeden den.

Ekonomický pracovník vypisuje průměrně 50 adres denně. V tomto případě se jedná o 1100 adres měsíčně. Pokud tyto počty spojíme s časovou náročností výše, tak se dostáváme do zajímavých čísel, kdy ekonomický pracovník stráví nemalou část své pracovní doby pouhým vyplňováním. Pro přehlednější pohled byla sestavena tabulka 6.1, kterou lze vidět níže.

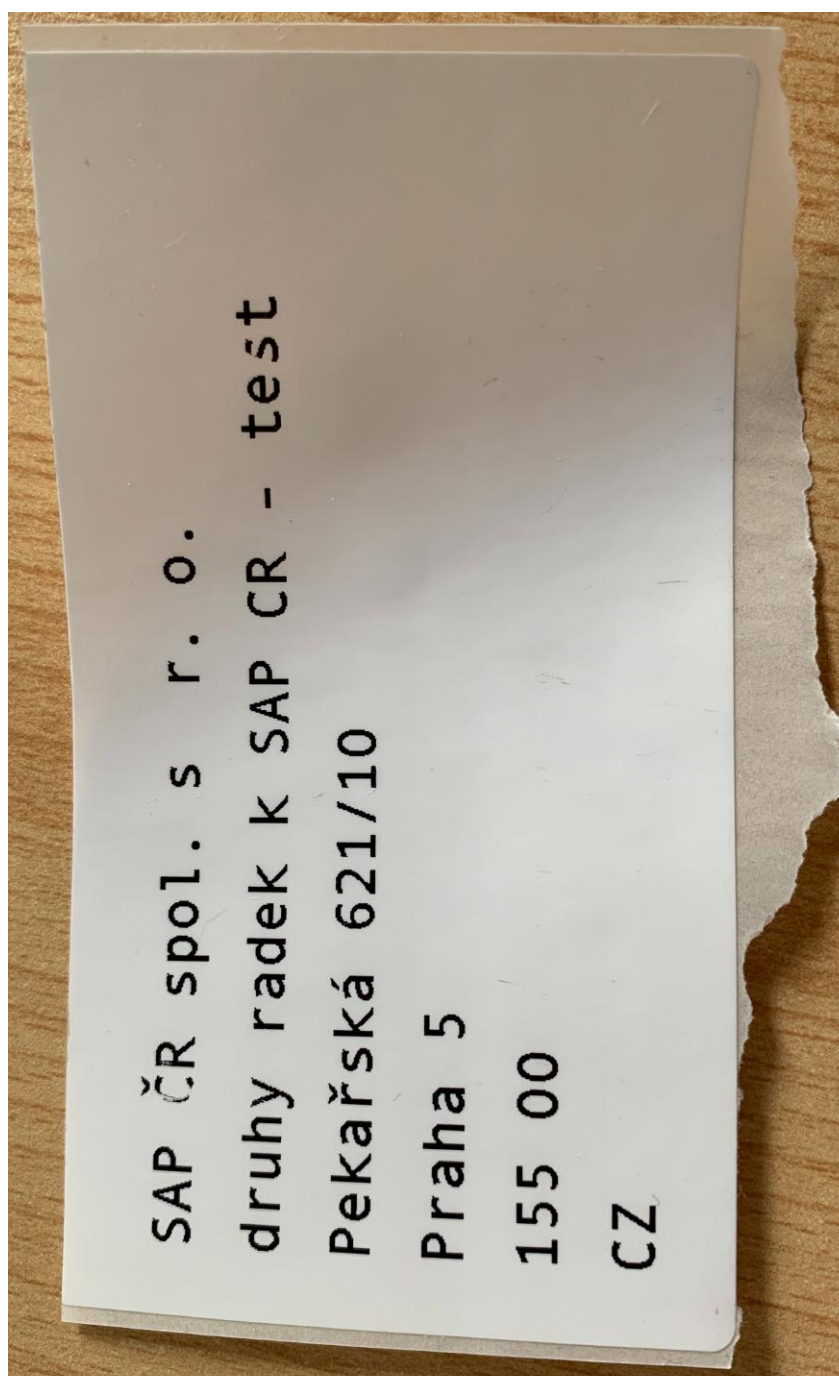
Strávený čas vyplňováním adres			
	za 1 den	za 1 měsíc	za 1 rok
Před	37.5 min	13.125 hod	157.5 hod
Po	4.5 min	1.575 hod	18.9 hod

Tabulka 6.1: Přehled časové vytíženosti ekonomických pracovníků.

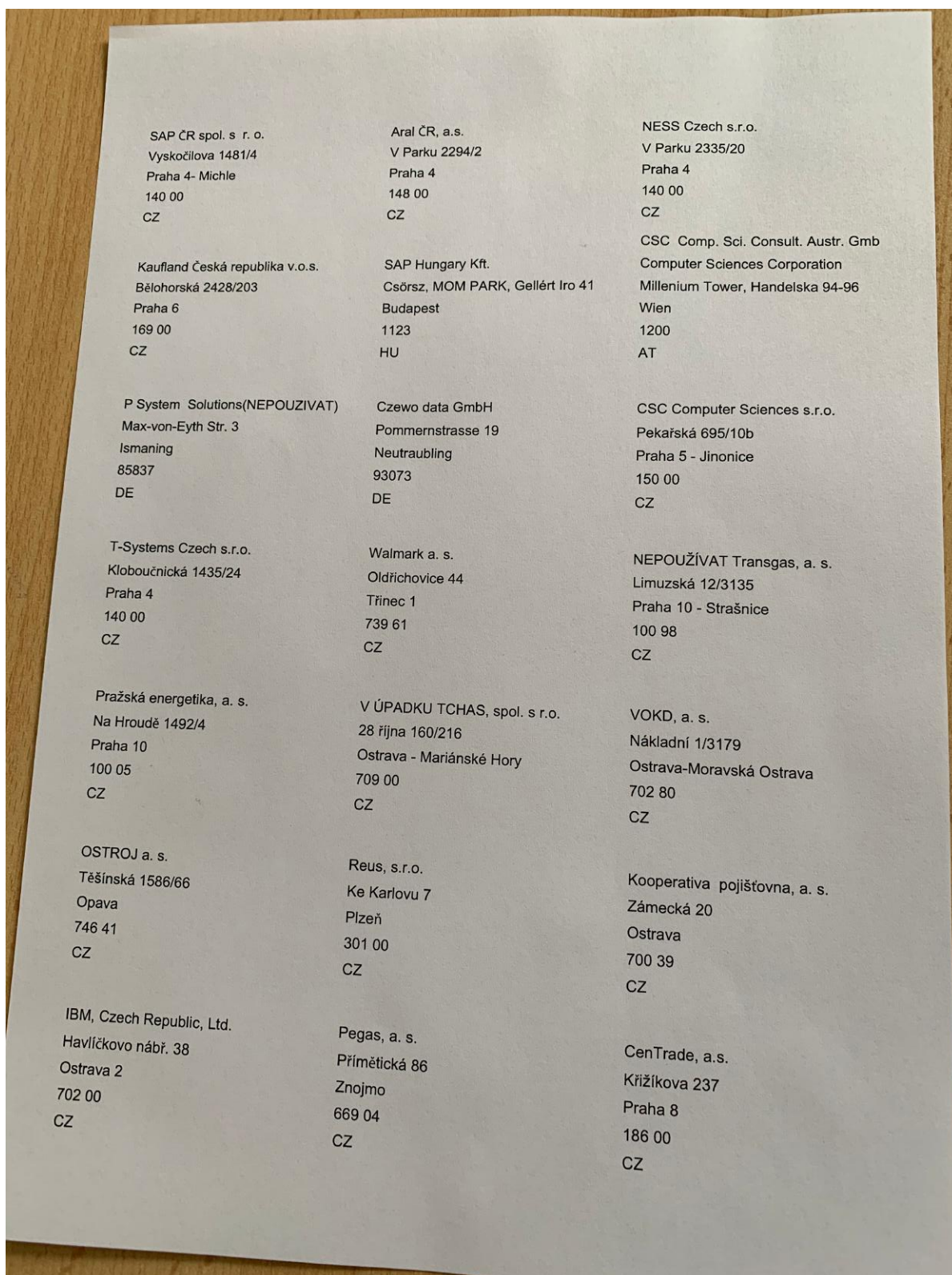
Na základě výše vypočtené tabulky jsme zjistili, že za použití této aplikace lze vyplňování adresních štítků urychlit až o 88 procent v ročním srovnání. Konkrétně se jedná o 138 a půl hodin méně tohoto úkonu v jednom pracovním roce. Díky této aplikaci se mohou tito pracovníci firem věnovat důležitějším úkonům.

6.6 Finální podoba tisku

V rámci testování byly odladěny chyby a odchylky během tisku formulářů. Testování bylo úspěšně dokončeno a výsledné formuláře byly zdokumentovány. Tyto zdokumentované jednotlivé finální výsledky tisku lze vidět na obrázcích 6.4, 6.5, 6.6 a 6.7.



Obrázek 6.4: Finální podoba adresního štítku na tiskárně Intermec PC43t.



Obrázek 6.5: Finální podoba 21 adresních štítků na formátu papíru A4.

KCT Data s. r. o.
Valčíkova 1369/17

Podavatel: Praha Libeň

Odesílatel: 182 00

Číslo zákaznické karty:

Úhrada ceny: v hotovosti/převodem/výplatním strojem/Kreditem/poštovními známkami (Nehodící se škrtněte)

Poř. číslo	Podací znaky – vyplní pošta	Adresát, ulice, číslo domu, PSČ, místo určení, telefonní kontakt, e-mail	Dobírka	Udaná cena *	Poznámka, doplnkové služby
		SAP ČR spol. s r. o. Vyskočilova 1481/4 Praha 4- Michle 140 00			
		Aral ČR, a.s. V Parku 2294/2 Praha 4 148 00			
		NESS Czech s.r.o. V Parku 2335/20 Praha 4 140 00			
		Kaufland Česká republika v.o.s. Bělohorská 2428/203 Praha 6 169 00			
		SAP Hungary Kft. Csörsz, MOM PARK, Gellért Iro 41 Budapest 1123			
		CSC Comp. Sci. Consult. Austr. Gmb Computer Sciences Corporation Millenium Tower, Handelska 94-96 Wien 1200			
		P System Solutions(NEPOUZIVAT) Max-von-Eyth Str. 3 Ismaning 85837			
		Czewo data GmbH Pommernstrasse 19 Neutraubling 93073			
		CSC Computer Sciences s.r.o. Pekarská 695/10b Praha 5 - Jinonice 150 00			
		T-Systems Czech s.r.o. Kloboučnická 1435/24 Praha 4 140 00			

Česká pošta, s.p., IČ 47114983

Potvrzení pošty:

* V případě zásilek, u kterých v souladu se zněním Poštovních nebo Obchodních podmínek České pošty odesílatel neuvádí udanou cenu, se sloupec Udaná cena nevyplňuje.

Obrázek 6.6: Finální podoba podacího archu České pošty s vyplněnými adresami.

Pro správné vytištění Podacího lístku je nutné nastavit tisk tak, aby „měřítko stránky“ bylo na hodnotě „Žádné“ .
 Vytiskněte **oboustranně** a **velmi pečlivě vystřihněte** . Rozměr správně vytištěného Podacího lístku je 133 x 90 mm.

Podací lístek

Vyplní pošta:

Odesílatel **KCT Data**
Valčíkova 1369/17
 1 8 2 0 0 : Praha Libeň Kód zákazníka (odesílatele)
 Tel. číslo: +420 *
 E-mail:

Adresát Dobírka Kč
SAP CR spol. s r. o.
Vyskočilova 1481/4 Udaná cena Kč
 1 4 0 0 0 : Praha 4- Michle Kód zákazníka (adresata)
 Tel. číslo: +420 *
 E-mail:

Druh zásilky	Hmotnost	Cena služby
Podací číslo	Poznámka (doplňkové služby)	Placeno v hotovosti

T11-036A (03-11) * Nepovinný údaj

Podací lístek

Vyplní pošta:

Odesílatel **KCT Data**
Valčíkova 1369/17
 1 8 2 0 0 : Praha Libeň Kód zákazníka (odesílatele)
 Tel. číslo: +420 *
 E-mail:

Adresát Dobírka Kč
Aral CR, a.s.
V Parku 2294/2 Udaná cena Kč
 1 4 8 0 0 : Praha 4 Kód zákazníka (adresata)
 Tel. číslo: +420 *
 E-mail:

Druh zásilky	Hmotnost	Cena služby
Podací číslo	Poznámka (doplňkové služby)	Placeno v hotovosti

T11-036A (03-11) * Nepovinný údaj

Podací lístek

Vyplní pošta:

Odesílatel **KCT Data**
Valčíkova 1369/17
 1 8 2 0 0 : Praha Libeň Kód zákazníka (odesílatele)
 Tel. číslo: +420 *
 E-mail:

Adresát Dobírka Kč
NESS Czech s.r.o.
V Parku 2335/20 Udaná cena Kč
 1 4 0 0 0 : Praha 4 Kód zákazníka (adresata)
 Tel. číslo: +420 *
 E-mail:

Druh zásilky	Hmotnost	Cena služby
Podací číslo	Poznámka (doplňkové služby)	Placeno v hotovosti

T11-036A (03-11) * Nepovinný údaj

Podací lístek

Vyplní pošta:

Odesílatel **KCT Data**
Valčíkova 1369/17
 1 8 2 0 0 : Praha Libeň Kód zákazníka (odesílatele)
 Tel. číslo: +420 *
 E-mail:

Adresát Dobírka Kč
Kaufland Česká republika v.o.s.
Bělohorská 2428/203 Udaná cena Kč
 1 6 9 0 0 : Praha 6 Kód zákazníka (adresata)
 Tel. číslo: +420 *
 E-mail:

Druh zásilky	Hmotnost	Cena služby
Podací číslo	Poznámka (doplňkové služby)	Placeno v hotovosti

T11-036A (03-11) * Nepovinný údaj

Obrázek 6.7: Finální podoba podacího lístku České pošty s vyplněnými adresami.

Kapitola 7

Závěr

7.1 Zhodnocení

Výstupem této práce v rámci první části bylo seznámení se systémem SAP, jeho prostředím a jednotlivými možnostmi vytváření formulářů v tomto systému. V druhé části bylo zanalyzováno jednotlivé možnosti tvorby čárových kódů v systému SAP, propojení s tiskárnou a možnosti tisku z prostředí systému SAP. V třetí kapitole byly popsány specifikace požadavků a návrh aplikace. Návrh obsahuje navržené řešení jednotlivých formulářů pro jednotlivé úlohy v systému SAP a základní modely s využitím modelovacích technik. V rámci čtvrté kapitoly byly sepsány specifikace požadavků aplikace a její návrh. V návrhu byly využity moderní modelovací techniky, jako je diagram případu užití a sekvenční diagram řízení tisku. V další kapitole byly sepsány implementační záležitosti, jako je popis programu, způsob vytvoření obrazovek, získávání dat, zachycování událostí obrazovek a jednotlivé implementace všech formulářů. Kromě toho zde byl zobrazen způsob konfigurace tiskárny Intermec PC43t. V předposlední kapitole této práce byly popsány způsoby testování a přenos aplikace na testovací server. Byly zde projednány chyby během testování aplikace a způsoby jejich zamezení. Rovněž byla zhodnocena použitelnost aplikace a bylo provedeno srovnání časové náročnosti daného úkonu uživatele s touto aplikací a bez ní, kde bylo zjištěno, že díky této aplikaci dokáže ekonomický pracovník zkrátit pracovní dobu vyplňování adres až o 88% ročně. Díky této aplikaci se během jednoho dne průměrný úkon vyplňování adres zkrátí z 37,5 minut na 4,5 minuty.

7.2 Možný budoucí vývoj

Při vývoji této aplikace byl použit objektivně orientovaný přístup, který umožňuje snadněji rozšířit tuto aplikaci v budoucnu. Tato aplikace momentálně poskytuje zázemí ve formě několika formulářů pro aktuální využití v praxi. V případě možnosti přidání nových typů formulářů je tato aplikace připravena poskytnout data a to včetně řízení tisku. V budoucnu tedy může nabízet tisk adres pro mnohem větší počet možných druhů formulářů. Například lepící formuláře na balíky od jiných dopravních společností. Dalším možným rozšířením je zde možnost tisknout 2D kód, který bude uchovávat tyto adresní informace v sobě. Tímto způsobem by se využitelnost těchto štítků rozšířila i do jiných segmentů podnikových procesů.

Literatura

- [1] *Calling ABAP Dialog Modules* [online]. [cit. 2021-03-03]. Dostupné z: <http://saphelp.ucc.ovgu.de/NW750/EN/4a/3a2f8f4275044ee10000000a421937/frameset.htm>.
- [2] *Introduction to SAP Scripts* [online]. [cit. 2020-11-11]. Dostupné z: <https://www.guru99.com/sap-script.html>.
- [3] *Remote Printing (Access Methods S and U)* [online]. [cit. 2021-03-03]. Dostupné z: <http://saphelp.ucc.ovgu.de/NW750/EN/4e/5d5fae719b4db0e10000000a42189b/content.htm>.
- [4] *Smart Forms in SAP ABAP* [online]. [cit. 2020-11-11]. Dostupné z: <https://www.guru99.com/smart-forms.html>.
- [5] *What is ALV? An Overview of ABAP List Viewer/SAP List Viewer* [online]. [cit. 2021-03-03]. Dostupné z: <https://blog.sap-press.com/what-is-alv-an-overview-of-abap-list-viewer-and-sap-list-viewer>.
- [6] *What Is the Difference Between 1D and 2D Barcode Scanning?* [online]. [cit. 2020-11-11]. Dostupné z: <https://lowryolutions.com/blog/what-is-the-difference-between-1d-and-2d-barcode-scanning/>.
- [7] BANDARI, K. *Complete ABAP*. SAP Press, 2016. ISBN 978-1-4932-1272-9.
- [8] BEEMANATI, J. *SAP Code Inspector (SCI) Tutorial* [online]. [cit. 2021-03-03]. Dostupné z: <https://blogs.sap.com/2011/02/11/sap-code-inspector-sci-tutorial/>.
- [9] FÄRBER, G. a KIRCHNER, J. *ABAP Basics*. 2. vyd. SAP Press, 2010. ISBN 978-1592293698.
- [10] GUPTA, T. *ABAP Data Dictionary*. SAP Press, 2011. ISBN 978-1592293797.
- [11] HAUSER, J., DEUTESFELD, A., SZÜCS, T. a REHMANN, S. *SAP Interactive Forms by Adobe*. 2. vyd. SAP Press, 2011. ISBN 978-1-59229-795-5.
- [12] HONEYWELL. *ESim* [online]. 2017 [cit. 2020-11-11]. Dostupné z: <https://www.honeywellaidc.com/en/-/media/en/files-public/technical-publications/printers/1common/937-024.pdf>.
- [13] MARKANDEYA, S. a ROY, K. *SAP ABAP*. 1. vyd. Apress, 2014. ISBN 978-1-4302-4804-0.

- [14] PRESS, S. *ABAP and Continuous Integration: the ABAP Test Cockpit and the ABAP Unit Runner* [online]. [cit. 2021-03-03]. Dostupné z: <https://blog.sap-press.com/abap-and-continuous-integration-the-abap-test-cockpit-and-the-abap-unit-runner>.
- [15] PUBLISHING, R. *ABAP Programming for SAP* [online]. [cit. 2020-11-11]. Dostupné z: <https://learning.sap-press.com/>.
- [16] SAP. *SAP History* [online]. [cit. 2020-11-11]. Dostupné z: <https://www.sap.com/corporate/en/company/history.html>.
- [17] SAP. *SAP History | About SAP SE* [online]. [cit. 2020-11-11]. Dostupné z: <https://www.sap.com/corporate/en/company/history.html?pdf-asset=70eee289-847d-0010-87a3-c30de2ffd8ff&page=1>.
- [18] SAP. *SAP Software Solutions | Business Applications and Technology* [online]. [cit. 2020-11-11]. Dostupné z: <https://www.sap.com/>.
- [19] SAP. *PDF-based Interactive Forms* [online]. 2005 [cit. 2020-11-11]. Dostupné z: https://www.adobe.com/nl/pdfs/7_SAP.pdf.
- [20] SAP. *SAP INTERACTIVE FORMS BY ADOBE* [online]. 2007 [cit. 2020-11-11]. Dostupné z: http://wwimages.adobe.com/www.adobe.com/enterprise/partners/pdfs/solution_in_detail_interactive_forms.pdf.

Příloha A

Obsah přiloženého CD

- Složka se zdrojovými kódy
- Složka s diplomovou prací ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
- Diplomová práce ve formátu PDF