

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

2014

Jakub Poremski

Zadání bakalářské práce

Student: **Jakub Poremski**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: CATHEDRAL Software, s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jan Gaura**

Konzultant bakalářské práce: Ing. Josef Šustr

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014




doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 5. května 2014

A handwritten signature in cursive script, appearing to read "J. Šamr", is written above a horizontal dotted line.

Rád bych na tomto místě poděkoval svému konzultantovi ing. Josefu Štrovi, že mi umožnil absolvování praxe ve firmě CATHEDRAL Software, s.r.o. a za vedení v jejím průběhu. Dále děkuji ing. Janu Gaurovi za vedení při tvorbě bakalářské práce.

Abstrakt

Tato bakalářská práce stručně pojednává o průběhu mé odborné praxe ve společnosti Cathedral Software, s.r.o. V této práci seznamuji s firmou a pracovním prostředím v kterém jsem praxi absolvoval. Hlavním obsahem je poté stručný popis úkolů, které jsem na praxi řešil a samotný postup řešení. Poté porovnávám zkušenosti a dovednosti využití a získané v praxi. V závěru hodnotím průběh praxe jako celek a její přínos.

Klíčová slova: Odborná praxe, Cathedral Software, Python, Dynamická maska, Django, XML, PyDev, Webové rozhraní, HTML

Abstract

This bachelor's thesis briefly discusses the course of my professional practice at company Cathedral Software, s.r.o. This paper introduces the company and work environment in which I practice graduated. The main content is then a brief description of the tasks that I have to deal with and procedure to resolve them. Then compare the experience and skills utilized and obtained during practice. In conclusion, I assess the progress of the practice as a whole and its benefits.

Keywords: Professional Practice, Cathedral Software, Python, Dynamic mask, Django, XML, PyDev, Web interface, HTML

Seznam použitých zkratk a symbolů

XML	– Extensible Markup Language
HTML	– Hyper Text Markup Language
PHP	– PHP Hypertext Preprocessor
ORM	– Object Relational Mapping
CSS	– Cascading Style Sheets

Obsah

1	Úvod	4
2	Zasazení praxe	5
2.1	Popis odborného zaměření firmy	5
2.2	Popis pracovního zařazení	6
3	Zadané úkoly a jejich časová náročnost	7
3.1	Studie: Možností jazyka Python	7
3.2	Komponenta web rozhraní pro zobrazení dynamické masky výrobku . . .	7
4	Řešení zadaných úkolů	8
4.1	Studie: Možností jazyka Python	8
4.2	Komponenta web rozhraní pro zobrazení dynamické masky výrobku . . .	10
5	Znalosti a dovednosti, které jsem využil v průběhu praxe	14
6	Znalosti a dovednosti, které jsem získal v průběhu praxe	15
7	Závěr	16
8	Reference	17

Seznam obrázků

1	Logo firmy	5
2	Příklad instalace pomocí manažeru PIP	11

Seznam výpisů zdrojového kódu

1	Struktura XML dokumentu	12
---	-----------------------------------	----

1 Úvod

Cílem této bakalářské práce je sestavení závěrečné zprávy o mém působení na praxi ve firmě CATHEDRAL Software s.r.o., kde jsem nastoupil začátkem října 2013 a do dubna 2014 se podílel na vývoji webového rozhraní pro firemní systém ArisCAT v jazyce Python.

Bakalářskou práci v podobě praxe jsem si vybral hlavně z toho důvodu, že jsem si chtěl ověřit jestli převážně teoretickou výuku jsem schopen využít i prakticky. A také pro získání mých prakticky prvních zkušeností s prací v oboru, který studuji.

V první kapitole vás seznámím jak s firmou CATHEDRAL Software s.r.o. a jejich systémem ArisCAT, tak s mým pracovním zařazením. V následujících dvou kapitolách uvedu zjednodušené zadání a postup řešení zadaného úkolu. Dále vás seznámím s prostředky, které jsem k úspěšnému řešení využil. Dále se budu zabývat znalostmi a zkušenostmi nabytými studiem na Vysoké škole báňské a také těmi získanými na praxi. Nakonec bude následovat celkové zhodnocení a závěr.

2 Zasazení praxe

2.1 Popis odborného zaměření firmy

Firma Cathedral Software s.r.o. se v současné době zabývá převážně vývojem informačních systémů. Především se pak jedná o jejich úspěšný modulární informační systém ArisCAT pro výrobce a prodejce stínící techniky.

2.1.1 Historie

Společnost byla založena v polovině roku 1996 jako sdružení fyzických osob Cathedral SoftwareHardware. Firma sídlí na Netušilově ulici v Prostějově.

Od roku 2000 se firma zaměřuje stále více na vývoj informačních systémů, což díky zkušenostem získaných při vývoji a implementaci různých produktů vyústilo vývojem informačního systému ArisCAT. Jenž je v současné době stěžejním produktem společnosti.

Za zmínku jistě stojí že díky úspěchům v oblasti vývoje informačních systémů byla navázána spolupráce se společností LCS International a.s. (dnes Asseco Solutions, a.s.), která patří mezi přední výrobce podnikových aplikací.

dne 16. 7. 2007 sdružení fyzických osob změnilo právní formu a tím byla založena společnost CATHEDRAL Software s.r.o.



Obrázek 1: Logo firmy

2.1.2 ArisCAT

„Systém podle vašich představ“ takto firma uvádí svůj hlavní produkt. Jedná se o komplexní software zaměřující se převážně na zpracování nabídek a konstrukci stínící techniky (žaluzie, rolety, markýzy, atd.). Systém je optimalizován pro zpracování velkého množství parametrů, jež tyto výrobky vyžadují.

Systém se vyznačuje velkou možností modulace pro specifická přání klienta, snadnou a rychlou instalací. Výhodou jistě je, že projekt se nejen prodává, ale stále vyvíjí a jsou dostupné stále nové aktualizace.

2.1.3 Další zaměření firmy

Mimo systém ArisCAT má firma Cathedral Software s.r.o. i další aktivity, mezi které nepochybně patří systém Helios Orange společnosti Asseco Solutions jejíž jsou autorizovaným

implementačním a integračním partnerem. S tímto systémem jsem se během praxe v podstatě nesetkal, ale domnívám se, že zaměření je podobné jako u informačního systému ArisCAT.

Třetím, spíše okrajovým zaměřením společnosti je správa a údržba počítačových sítí, či samotná dodávka počítačů a serverů, kabelážních systémů. Firma také nabízí instalaci IP kamerového systému.

2.2 Popis pracovního zařazení

Po počátečním pohovoru, který nekladl důraz na schopnosti nabyté při studiu, ale na ochotu a zájem tyto schopnosti získat během praxe, jsem se zúčastnil první schůzky, kde mi byly sděleny podrobnosti mého zařazení na pozici vývojáře. Vyžadováno bylo alespoň minimální povědomí o nástrojích pro vývoj webu, základy HTML a Pythonu.

Cílem mé praxe bylo přiblížit možnosti vývoje webového rozhraní v jazyce Python, společně s vytvořením plnohodnotné komponenty pro webové rozhraní systému ArisCAT v tomto jazyce.

3 Zadané úkoly a jejich časová náročnost

Má práce ve firmě Cathedral Software s.r.o. se skládala ze dvou úkolů. Po seznámení s kolektivem mi praktikant, který ve společnosti působil během prázdnin a v době mého nástupu končil, v rámci úvodního školení předvedl projekt, kterým se zabýval a přiblížil fungování firmy.

Na této práci jsem provedl určité množství úprav, především v oblasti kaskádových stylů. Šlo v podstatě o spřehlednění kódu a různých změn v pozicování. Protože jsem na začátku praxe nedisponoval vlastním notebookem, byl mi přidělen vlastní počítač, kde jsem musel provést řadu úprav a oprav původních instalací, abych byl schopen nainstalovat nám doporučené vývojové prostředí Eclipse a knihovny pro práci s Pythonem.

Později se ukázalo, že na dalším vývoji této aplikace se podílet nebudu a že sloužila v podstatě jako nástroj na získání orientace ve vývojovém prostředí, které nám bylo doporučeno i pro další využití v průběhu praxe. Z tohoto důvodu se této problematice dále nebudu věnovat.

3.1 Studie: Možností jazyka Python

Prvním úkolem bylo provést nezávislý rozsáhlejší průzkum možností využití jazyka Python, který měla firma zájem začít využívat. Cílem bylo porovnat jazyk Python v konkurenci jiných nástrojů pro vývoj webových aplikací, především pak s jazykem PHP, který firma využívá v současné době, ale od kterého plánuje v dalším vývoji upustit. Zjistit možnosti využití dostupných frameworků a knihoven pro jazyk Python. Tyto informace jsem měl dále využít a otestovat při následujícím úkolu.

3.2 Komponenta web rozhraní pro zobrazení dynamické masky výrobku

Cílem bylo vytvoření komponenty/frameworku web rozhraní pro zobrazení dynamické masky výrobku (formátování objektů do tabulky). Zadáním mělo být vytvoření univerzálního způsobu zobrazování dynamické zadávací masky na webové aplikaci. Podklady pro tuto masku jsou uloženy v databázi, obsahují informace o typu objektu, pozici na obrazovce a případných vazbách na ostatní objekty na obrazovce.

4 Řešení zadaných úkolů

4.1 Studie: Možností jazyka Python

Protože se firma Cathedral Software s.r.o. možnostmi vývoje v jazyce Python začala teprve zabývat, dostal jsem za úkol provést nazávislý průzkum jaké jsou vlastně jeho možnosti a jaké jsou výhody oproti jazyku PHP který firma momentálně využívá.

4.1.1 Řešení

Začal jsem pátráním v článcích a diskuzích zabývajících se tématem porovnávání Pythonu a PHP, byl jsem až překvapen, že i když má PHP značně větší zastoupení na trhu, že znám osobně spoustu PHP programátorů a vlastně žádného, který by využíval Python, jsem narazil na obrovské množství článků týkající se této problematiky. O volbě toho správného jazyka se vedou nekonečné spory a nedokážu říci co je vlastně lepší. Uvedu tedy jen pár postřehů a důvody proč bych Python volil já. Výhodou PHP je jeho všeobecná větší rozšířenost, frameworky pomocí kterých se dají jednoduché aplikace doslova naklikat a je tudíž jednoduché zvládnout základy. Navíc jej podporují všechny web hostingy. Hosting pro python stále ještě není samozřejmostí, i když se to v posledních letech zlepšuje. Výhodami Pythonu jsou obrovské množství volně dostupných knihoven, kvalitní Freeware vývojová prostředí a samotný vývoj, který je u PHP řekněme takový podivný. Kdy vývojáři docela často vydají něco, co vlastně pořádně nefunguje. A i když je PHP dá se říci všude, když se podívám po nějakém tutoriálu a chtěl bych začít od nuly, překvapivě jsem našel daleko více přehledných návodů k Pythonu a nabyl jsem dojmu, že je pro laika jednodušší sehnat pomoc pro základy, protože je na různých diskuzích mnoho nadšenců kteří vývoj v pythonu neberou pouze jako práci.

4.1.1.1 Frameworky Základem pro vývoj webu v jazyce Python je kvalitní framework. To je softwarová struktura, která slouží jako podpora pro programování jiných softwarových struktur. Pomáhá nám při organizaci kódu a některé věci řeší za nás. Příkladem takového frameworku pro jazyk PHP je třeba Nette, který má českého autora, ale já se chci zabývat Pythonem, takže jsem se zaměřil na 4 nejlépe hodnocené a nejčastěji se vyskytující frameworky na internetu. Prvním frameworkem je Django o kterém jsem povědomí získal už během studia, tento framework poskytuje zkrátka vše. Byl vyvinut v novinářském prostředí před několika lety, tak aby splňoval přísné požadavky vývojářů a zároveň intenzivní plnění termínů v redakci. Django má pro mě až neuvěřitelně obrovské možnosti doplňků, přičemž už v základním balíku se vyskytuje například vlastní šablonovací systém nebo ORM. Django vám vytvoří administrační systém, takže vám odpadne tato část programování. Protože to byl jediný framework pro vývoj v pythonu o kterém jsem věděl již dříve, zaměřil jsem se na něj. Zjistil jsem, že naše využití ve škole bylo opravdu jen nakouknutí pod povrch. A že je to tak masivní nástroj, že studium tohoto frameworku zabere velmi mnoho času a během poměrně krátké praxe to pro mě nemá využití. Zkušenějšímu vývojáři bych ho jistě doporučil, ale pro mou praxi jsem se rozhodl, že méně může znamenat více a že čas strávený snahou naučit se orientovat ve frameworku Dajngo

je daleko větší, než u jednodušších a méně rozsáhlých frameworků. Toto rozhodnutí jsem, jak se dočtete níže, později přehodnotil a rozhodl se právě pro Django. Kromě jeho komplexnosti bych se rád zmínil i o faktu, že podporuje instalaci pod vývojovým prostředím Eclipse, které firma využívá. Což mi oproti studiu dále zmíněných frameworků značně zjednodušilo základy.

Kolega na praxi poté přišel s frameworkem Web2Py. Možnost vzájemné pomoci při řešení problémů od instalace počínaje, implementací konče mi připadala jako dobrý nápad a proto jsem se tímto frameworkem začal podrobněji zabývat. Weby dělané v tomto frameworku vypadaly hezky a na internetu se vyskytuje velké množství videotutoriálů i jiných návodu použití. Z počátku jsem byl spokojen. Administrační rozhraní bylo vcelku intuitivní, jednoduché programování a úpravy se daly provádět přímo v něm, pro složitější je samozřejmě důležité vývojové prostředí, které sice Web2Py poskytuje, ale jeho možnosti jsou značně omezené nebo jsem je neobjevil. I zde jsem po chvíli pátrání našel návod, jak využít tento framework pod Eclipse. V podstatě je to svým rozsahem alternativa k Django. Jediné co se mi opravdu nelíbilo byl fakt, že Web2Py generuje sám od sebe poměrně velké množství souborů a kódu a tak celkově si dělá co chce. Ano, je fakt, že toto je podstata fungování frameworků, ale oproti výše zmíněného Django, je toto množství neúnosné. Při snaze zjistit, co vlastně dělá a chce jsem ztratil více času, než tutoriály které jsem si prošel pro pochopení základů tohoto frameworku. Protože se nerad potýkám při psaní aplikace s kódem, který se tvoří sám a kterému pořádně nerozumím, bych Web2Py rozhodně nedoporučil.

Dále už jsem provedl vlastní neovlivněný průzkum. Prošel větší množství webů, přečetl několik hodnocení a zkušeností uživatelů. Pro úplné základy se nejčastěji vyskytovala zmínka o frameworku Bottle, který je často uváděn jako studijní pomůcka a má minimum rozšíření. Z toho důvodu jsem se jím hlouběji nezabýval. Z množství frameworků na které jsem dále narazil mi vykrytalizovaly nakonec 2. Byly jimi Pyramid a Flask.

Flask je poměrně mladý microframework a proto není tak rozšířen jako dva výše zmíněné a materiálů k němu není tolik. Nicméně dokumentace k němu je podrobná a hlavně přehledná. Základní verze tohoto frameworku je jak napovídá označení micro velmi jednoduchá. Má ovšem velké množství dostupných rozšíření. Toto mě zaujalo, protože tak jako budu rozšiřovat své vědomosti a aplikace, mohou růst i možnosti mého frameworku. Flask jsem si právě pro svou počáteční jednoduchost rychle oblíbil. Pořád v sobě obsahuje vlastní server pro vývoj a debugger. Oproti výše zmíněným frameworkům nemá vlastní šablony, ale už v základním balíku je obsažen šablonovací systém jinja2. Instalace na OS Windows je náročnější, ale možná. Protože se mi ovšem zalíbil, věnoval jsem čas instalaci Linuxu a získání některých návyků k jeho používání. Později jsem si ale uvědomil, že pro další využití pro firmu není méně rozšířený framework ideálním řešením a proto jsem se včas vrátil k Django. Včas znamená, že změny, které jsem musel provést k vývoji pod Djangem byly minimální. Další nevýhodou byl vývoj pod aplikací Eclipse, který je mi ani přes velkou snahu nepodařil optimalizovat.

Framework Pyramid je rozsahově něco mezi Flaskem a například Djangem. Je to taktéž mladý framework, ale je nástupcem projektu Pylons, ze kterého bylo odstraněno

to špatné a pochopitelně něco nového přidáno. Stejně jako flask není, tak hodně rozšířen. Dokumentace k němu je přehledná a podařilo se mi optimalizovat vývoj pod prostředím Eclipse. Tutoriály k tomuto frameworku jsou také snadno dohledatelné jak na webech zaměřených vývojem pod tímto frameworkem, tak také na youtube, což mi umožnilo získat základní přehled. Nicméně mě Pyramid neoslovil, tak jako Flask ani není na vysoké úrovni Django.

4.1.1.2 Šablony Šablony nám vytvářejí kostru naší webové aplikace, nejsou nezbytné, ale velmi nám usnadňují práci a dodávají ji základní systém. Protože jsem byl v počátku rozhodnut pro Flask, zabýval jsem se i tímto tématem.

Prakticky jediné dvě zmiňované šablony s minimem negativního hodnocení pro tvorbu webových aplikací v Pythonu se při mém pátrání kromě vlastních šablon Django objevovaly šablony Mako a Jinja2. Úplně nejlépe hodnocená a méně problémová v mém průzkumu vycházela šablona Mako. Nicméně i šablona Jinja2 vyšla v hodnocení také velmi dobře. Jediné výtky byly k práci se seznamy. Protože jsem se prvně rozhodl k práci využít framework Flask, který využívá právě Jinja2, tak i přes to, že jsem našel poměrně jednoduchý návod, jak tyto šablony zaměnit, zůstal jsem u základního nastavení.

4.1.2 Využití

Všechny tyto poznatky se mi hodily při řešení dalšího úkolu a získal jsem díky nim širší náhled na problematiku tvorby webových aplikací v jazyce Python.

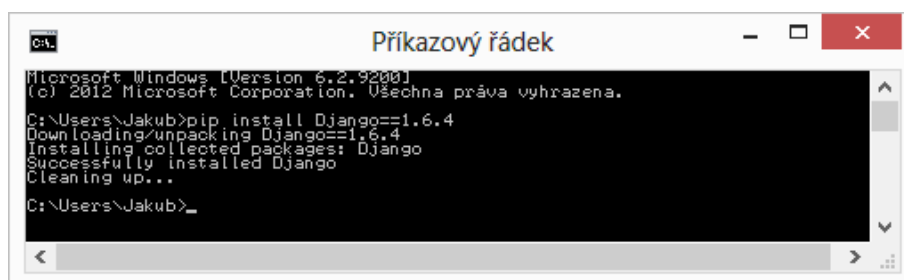
4.2 Komponenta web rozhraní pro zobrazení dynamické masky výrobku

Po skončení teoretického úkolu a získání základní orientace ve vývojovém prostředí a osvěžení si návyků pro tvorbu pomocí jazyka Python jsem dostal podrobnější specifikaci k mému druhému úkolu.

4.2.1 Použité technologie

Jako vývojové prostředí jsem využil Eclipse. Tento software je volně dostupný a byl nám doporučen při úvodním školení. Po instalaci Python IDE PyDev nám vytvoří plnohodnotné prostředí pro vývoj v jazyce Python. PyDev je oficiální doplněk k prostředí Eclipse a dá se tudíž snadno doinstalovat pomocí Eclipse samotného. PyDev podporuje vývoj pomocí frameworku Django, předpokladem ovšem je nainstalovaný Python interpret, v mém případě jsem zvolil verzi 2.7 a v něm nainstalovány všechny knihovny včetně například Django. Pro instalaci knihoven jsem zvolil pip, což je nástroj sloužící k instalaci a údržbě knihoven a balíčků pro Python. Možností instalace knihoven pro Python je více, nicméně tato mi připadá nejefektivnější.

Po velmi důkladné studii jsem pro tvorbu webové aplikace v Pythonu nejprve zvolil Flask framework. Díky faktu, že základní balíček tohoto frameworku obsahuje šablonu Jinja2, která je založena na šablonách Django jsem problém týkající se výběru šablon neřešil. Flask mi pro mou práci vystačil prakticky v základní verzi a nebylo třeba přidávat



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. Všechna práva vyhrazena.

C:\Users\Jakub>pip install Django==1.6.4
Downloading/unpacking Django==1.6.4
Installing collected packages: Django
Successfully installed Django
Cleaning up...
```

Obrázek 2: Příklad instalace pomocí manažeru PIP

nějaká rozšíření. Naštěstí jsem během prvních týdnů vývoje uvědomil, že osobní sympatie a spokojenost s méně rozšířeným frameworkem nejsou vhodným náhledem pro výběr technologie na které bude postavena aplikace, kterou odevzdám na konci mé praxe. V této době byl můj vývoj této komponenty ve fázi, kdy návrat k frameworku Django naštěstí znamenal jen minimum změn.

Společně se zadáním jsem dostal i příklad dynamické masky uloženou ve formátu XML, protože jsem se během studia se složitějšími XML soubory nesetkal a vždy sloužili jen jako okrajová ukázka, musel jsem vyhledat také nástroj pro práci s tímto druhem souborů. V tomto mi výborně posloužil online nástroj XMLGrid, který jsem si vybral právě z toho důvodu, že je volně k dispozici a nemusím nic stahovat do počítače. Tento nástroj jsem využil během úvodní studie XML dokumentu, ale později při samotném programování, kdy jsem zjistil že Eclipse nepodporuje vývoj webu a nechtěl jsem využívat více než jedno vývojové prostředí, jsem objevil rozšíření HTML editor pro Eclipse, který obsahoval i podobné, ne-li stejné možnosti jako XMLGrid.

4.2.2 Řešení

V první fázi jsem se potřeboval zorientovat v příkladu dynamické masky, který jsem obdržel zároveň se zadáním práce. K tomu mi skvěle posloužil již výše zmíněný nástroj XMLGrid a také několika stránková dokumentace popisující tuto masku, kterou jsem taktéž obdržel. Masku obsahovala 4 druhy tagů. Element mask byl rodičem definice jedné dynamické masky a jeho atributy popisovaly základní údaje o masce, tento tag se v XML vyskytuje prakticky vždy pouze jednou. Ten obsahuje elementy Object které popisují vždy jeden typ grafického výstupu(Label, Panel, Checkbox, atd.). Element Item který se týká grafického elementu typu ComboBox, dále obsahuje položky Action. Akce jsou triggerery ovlivňující vlastnosti jiných grafických elementů, například jejich viditelnost nebo nastavují jejich hodnotu.

Po pochopení jak maska funguje jsem přešel k samotné práci s ní. Nejprve jsem si rozparsoval jednotlivé elementy a pokusil se vypsat jejich atributy na konzoli. V nejčerstvější paměti jsem měl práci s javascriptem a JSON, který je konkurentem právě XML a proto mi tato část dělala ze začátku problémy.

```
<Masks>
  <Mask ID="1463" MaskNo="40" Name="Z90.GN" ...>
    <Object ID="85532" ID_MASK="1463" Name="CB_Ovlad.typ" ...>
      <Item ID="94323" ID_MASK="1463" ID_OBJECT="85532" ...>
        <Action ID="105432" ID_MASK="1463" ID_ITEM="94323" .../>
        <Action .../>
      </Item>
    </Object>
  </Mask>
</Masks>
```

Výpis 1: Struktura XML dokumentu

Poté jsem začal s atributy elementu mask, které v podstatě popisují základní grafické prvky stránky, jako je výška a šířka a také rozmístění jednotlivých oddílů. Protože element mask je v jednom XML dokumentu právě jeden, vkládání těchto hodnot do proměnných a jejich následné použití při tvorbě html dokumentu nebylo příliš složité. HTML dokument jsem si rozdělil do třech divů a jedním jsem je ještě obalil (Tag <div> rozděluje HTML dokument na části, se kterými se dá poté pracovat samostatně). Jeden oddíl bude obsahovat tabulku výrobků a je neměnný pro jednoho zákazníka, samozřejmě jiný zákazník už může mít své rozhraní nastaveno jinak. Další 2 jsou pro formuláře jednotlivých výrobků, to jsou MASTER pro zobrazení hlavních předvoleb objednávky a DETAILS pro podrobnější údaje. Tyto oddíly už mají pro každý výrobek proměnlivý vzhled, který získáme pomocí dynamické masky.

Dále jsem začal tvořit funkci, která měla za úkol začít zpracovávat jednotlivé objekty. Začal jsem nejjednodušším objektem typu text a pokusil se vykreslit všechny objekty masky s hodnotou InetVisible(bitový příznak určující zda li je prvek viditelný na webové aplikaci) nastavenou na 1 právě jako text. Hodnotu tohoto inputu jsem nastavil jako ID daného objektu abych si ověřil, že se mi zobrazují reálné data. V okamžiku kdy se mi to podařilo jsem začal rozlišovat jednotlivé druhy elementů. Stále jsem však pracoval pouze s atributy zodpovědnými za pozicování objektů a atributy určujícími viditelnost a přiřazujícím nadřazený formulář. Po HTML stránce jsem se nejprve pokusil tyto informace vkládat a pozicovat přímo do vytvořených oddílů. Toto řešení se ukázalo jako neefektivní a nedokonalé. Optimálním řešením se ukázalo vytvořit pro každý prvek vlastní oddíl, který se dal lépe pozicovat a nastavovat mu potřebné rozměry. Na závěr mi zbyly elementy typu ComboBox, s kterými bylo třeba pracovat dále a elementy typu Image, které měly speciální atributy.

Než jsem s nimi začal pracovat, musel jsem zpracovat všechny ostatní atributy elementu Object, protože už se mi sice vykreslovaly, ale jinak byly nefunkční, což bylo pro správné nastavení triggerů, které by měly tyto elementy nějakým způsobem upravovat zásadní. Když jsem byl hotov, mohl jsem se začít věnovat těmto spouštěčům. Jakmile jsem zpracoval fungování ComboBoxu a zobrazování itemů, přešel jsem na jejich funkčnost. Zpracování itemů jsem provedl tak, že jsem si pro každé ID comboboxu vytvořil seznam itemů, ty jsem poté v HTML vkládal jako možnosti pro určené záznamy Každý

item má několik akcí, protože jedna akce může ovlivnit pouze jeden objekt, ale jeden Item jich může ovlivňovat několik. Existují tři druhy akcí, proto jsem si vytvořil 3 funkce které je prováděly. Jsou to akce disable, která objekt znepřístupní pro změny. Akce hide zneviditelní objekt uživateli a akce set nastaví objektu určitou hodnotu.

Na úplný závěr jsem si ponechal objekt Image, protože měl několik specifických atributů a hlavně nebyl zastoupen v xml dokumentu, který jsem dostal jako příklad. Pomocí nástroje pro úpravu XML dokumentu jsem si proto upravil tuto masku, tak abych mohl problém obrázků řešit. Řešení z kterého jsem měl největší strach bylo nakonec až na práci s XML podobné, jako předchozí trigger. I obrázky totiž reagovaly na změnu itemů v daném comboboxu.

Práce byla zdánlivě hotova, ale opak byl pravdou, bylo třeba komponentu náležitě otestovat na příklady a situace, které nejsou v dynamické masce, kterou jsem měl k dispozici zahrnuta. K tomu mi opět posloužil již nástroj pro úpravu XML souboru, kdy jsem si po úpravě původní masky tyto situace vytvářel. Ukázalo se, že nejvíce chyb obsahovaly spuštěče.

Nedořešeným úkolem se stalo zobrazení upravené pro mobilní technologie, v tomto směru bych navrhl úpravu samotné masky, v které jsou pozice objektů pevně dané a rozmístěné do 2 větších celků (MASTER a DETAIL) a nejsou nijak vázány k sobě navzájem, ale umístěny na přesnou pozici v masce. Vypuštěním těchto pozic by se sice dalo docílit jiného uspořádání, ale ne seřazení a sdružení objektů které spolu souvisí. Momentálně při současné masce nevidím možnost rozumného pozicování na Mobilním zařízení.

4.2.3 Využití

Jestli se firma opravdu rozhodne vyvíjet webové rozhraní svého informačního systému v jazyce Python, věřím, že je tato komponenta plně využitelná a část dokumentace se dá taktéž využít jako návod instalace a práce s frameworkem Django

5 Znalosti a dovednosti, které jsem využil v průběhu praxe

Po dobu praxe jsem využíval převážně znalosti nabyté v předmětech volitelných. Největší měrou mi byl nápomocen předmět Skriptovací a programovací jazyky, který byl zaměřen na základy jazyka Python. Nutno ovšem dodat, že můj rozvoj v tomto ohledu končil zápočtem a bylo třeba vědomosti značně oprášit a rozšířit. Druhým opravdu cenným předmětem pro mě byl předmět Vývoj internetových aplikací, kde jsem se sice s pythonem jako nástrojem pro vývoj webových aplikací nesetkal, ale ve kterém jsem se setkal s HTML formuláři, pozicováním pomocí CSS a prací s XML soubory. Dále se mi hodil předmět Uživatelská rozhraní, kde část semestru byla zaměřena také na jazyk Python.

Tyto 3 volitelné předměty byly jakýmsi stěžejním základem pro praxi. Nicméně důležitá byla také práce s objekty, které se věnovalo hned několik předmětů, založených sice na jiném programovacím jazyce, ale naučily mě logiku myšlení. Předmět Softwarové inženýrství mě zase připravil na postupný vývoj software od zpracování a rozvrhnutí práce po postup vývoje a pravidel, což dalo mému snažení hlavu a patu, protože jsem na zadaném problému v podstatě pracoval samostatně od začátku až do konce.

6 Znalosti a dovednosti, které jsem získal v průběhu praxe

Školní předměty jsou samozřejmě koncipovány tak, aby studentovi umožnily náhled a základní povědomí k dané problematice. Takže jsem se v průběhu praxe musel zdokonalit a udělat si rozsáhlejší přehled o využití jazyka Python. Setkal jsem se s rozsáhlými XML soubory, které jsem v průběhu studia využil jen okrajově a naučil se v nich orientovat pomocí online editoru XMLGrid. Dále jsem se seznámil s web frameworky pracujícími na Pythonu a podrobněji s frameworkem Flask a jeho možnostmi.

Jakožto introvertní člověk jsem se v neposlední řadě naučil spolupráci a práci v kolektivu. Dále jsem rozvinul samostatné myšlení, protože při studiu je přece jen většina projektů předem rozvržena.

7 Závěr

Absolvování bakalářské práce formou praxe hodnotím určitě kladně. Práce v oboru je pro mě opravdu cenná zkušenost. A to i přes fakt, že během posledního roku bakalářského studia jsem svou budoucnost nasměroval směrem k sítím a studiu kurzů firmy CISCO poskytovaných na naší škole. Tyto technologie mě velice zajímají a odborné dovednosti získané na praxi v budoucnu pravděpodobně využiji jen zřídka.

Firma Cathedral Software s.r.o. je poměrně menší firma s příjemným kolektivem, tedy alespoň ta část kterou jsem poznal. Celkově největší zkušeností je pro mě právě práce ve firmě, která mě brala za plnohodnotného zaměstnance, protože bez ohledu na to, čím se budu v budoucnu zabývat, jsou praktické zkušenosti neocenitelné a jsem velmi rád, že jsem nějaké získal a zvolil si praxi. Tuto volbu bych určitě doporučil každému bez ohledu na výběr firmy.

Jakub Poremski

8 Reference

- [1] *Cathedral Software, s.r.o.* [online]. [cit. 2014-04-28]. Dostupné z: <http://www.cathedral.cz/>
- [2] *ArisCAT* [online]. [cit. 2014-04-28]. Dostupné z: <http://www.ariscat.com/>
- [3] GAURA, Jan. *Studijní opora k předmětu Skriptovací programovací jazyky a jejich aplikace: Skripta* [online]. 22. září 2013. [cit. 2014-04-28]. Dostupné z: <http://mrl.cs.vsb.cz/people/gaura/spja/skripta.pdf>
- [4] *Django project: web* [online]. [cit. 2014-04-28]. Dostupné z: <https://www.djangoproject.com>
- [5] *Python: web* [online]. [cit. 2014-04-28]. Dostupné z: <https://www.python.org/>
- [6] *W3Schools: the world's largest web development site* [online]. [cit. 2014-04-28]. Dostupné z: <http://www.w3schools.com/>