

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 Elektrotechnika a informatika

Studijní obor: 1802R022 Informatika a logistika

INTERAKTIVNÍ MAPA TECHNICKÉ UNIVERZITY V LIBERCI

**INTERACTIV MAP OF THE TECHNICAL UNIVERSITY
OF LIBEREC**

Petr Soukup

Vedoucí bakalářské práce: doc.RNDr. Pavel Satrapa, Ph.D.

Rozsah práce:

Počet stran textu... 46

Počet obrázků..... 5

Zadání bakalářské práce

(vložit originál)

Prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb. O právu autorském a o právech souvisejících s právem autorským).

Souhlasím s umístěním bakalářské práce v Univerzitní knihovně TUL.

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č.121/2000 Sb. o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užití své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

V Liberci dne 28. května 2009

.....
Podpis

Poděkování

Tato bakalářská práce vznikla pod dohledem pana doc. RNDr. Pavla Satrapy, Ph.D., kterému bych chtěl touto cestou poděkovat za neocenitelné připomínky a rady při vzniku této práce. Dále bych rád poděkoval svým rodičům za možnost studovat na vysoké škole a své přítelkyni za podporu a trpělivost.

Anotace

Tato bakalářská práce se zabývá návrhem a zhotovením interaktivní mapy Technické univerzity v Liberci. Cílem je zhotovit interaktivní mapu s WWW rozhraním. Výsledná aplikace je doplněna o vyhledávání učeben a pedagogů a je podpořena administrativním rozhraním.

Annotation

The bachelor's concerns desing and construction of the interactiv map of Technical university of Liberec. The aim is to make an interactive map with WWW interface. The resulting application is accompanied by a search of teachers and classrooms and is supported by an administrative interface.

Klíčová slova

Mapa

Internet

Webová aplikace

Framework

PHP

AJAX

Technická univerzita v Liberci

Key words

Map

Internet

Web application

Framework

PHP

AJAX

Technical university of Liberec

OBSAH

1	ÚVOD	9
2	TEORETICKÁ ČÁST	10
2.1	POSTUP PRO NÁVRH NOVÉHO PROJEKTU	10
2.1.1	<i>Zadání</i>	11
2.1.2	<i>Analýza</i>	11
2.1.3	<i>Design</i>	13
2.1.4	<i>Vývoj</i>	13
2.1.5	<i>Testování</i>	13
2.2	POUŽITÉ PROGRAMOVACÍ PROSTŘEDKY	14
2.2.1	<i>PHP a Framework</i>	14
2.2.2	<i>XHTML</i>	15
2.2.3	<i>AJAX - Asynchronous JavaScript and XML</i>	17
2.2.4	<i>MySQL</i>	17
2.3	VÝBĚR A ZPRACOVÁNÍ MAPOVÝCH PROSTŘEDKŮ	20
2.4	MAPOVÝ SERVER	22
3	PRAKTICKÁ ČÁST	24
3.1	ZADÁNÍ PROJEKTU	24
3.1.1	<i>Záměr projektu</i>	24
3.1.2	<i>Konkurence a inspirace</i>	24
3.1.3	<i>Cílová skupina projektu</i>	25
3.1.4	<i>Popis projektu</i>	25
3.2	ANALÝZA	26
3.2.1	<i>Klíčové funkce, procesy</i>	27
3.2.2	<i>Popis navazujících funkcí</i>	27
3.2.3	<i>CRON funkce</i>	28
3.2.4	<i>Konceptuální a fyzický model dat</i>	28
3.2.5	<i>Data flow diagramy</i>	29
3.2.6	<i>Analýza typů stránek existujících na webu</i>	30
3.2.7	<i>SEO specifikace</i>	30
3.2.8	<i>Sledované statistiky webu</i>	31
3.2.9	<i>Personalizační entity</i>	31
3.3	DESIGN	32
3.4	VÝVOJ	33
3.4.1	<i>Zpracování mapových podkladů</i>	34
3.4.2	<i>Použité programovací prostředky</i>	34
3.4.3	<i>Praktický popis programového řešení projektu</i>	35
3.5	TESTOVÁNÍ	44
4	ZÁVĚR	45
5	LITERATURA	46

SEZNAM POUŽITÝCH ZKRATEK

- HTML** *HyperText Markup Language* – Značkovací jazyk, který tvoří základ při tvorbě webové stránky.
- CSS** *Cascading Style Sheets* - Jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML.
- XHTML** *eXtensible HyperText Markup Language* - Značkovací jazyk pro tvorbu hypertextových dokumentů v prostředí webových stránek.
- XML** *eXtensible Markup Language* - Obecný značkovací jazyk, který umožňuje snadné vytváření konkrétních značkovacích jazyků pro různé účely a široké spektrum různých typů dat.
- Skript**..... Zdrojový kód programu, který je tzv. interpretován, tj. čten a spouštěn za běhu speciálním procesem, tzv. interpretem.
- SQL** *Structured Query Language* - Standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích.
- MySQL**..... Relační databázový systém.
- JavaScript**.... Multiplatformní, objektově orientovaný skriptovací jazyk.
- ASP** *Active Server Pages* - Skriptovací platforma společnosti Microsoft, primárně určená pro dynamické zpracování webových stránek na straně serveru.
- API**..... *Application programming interface* - Rozhraní pro programování aplikací.
- RSS** *Rich Site Summary* - Rodina XML formátů určených pro čtení novinek na webových stránkách a obecněji syndikaci obsahu.
- AJAX**..... *Asynchronous JavaScript and XML* - Obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich znovunačítání.
- ČÚZK**..... Český úřad zeměměřický a katastrální
- GIS**..... Geografický informační systém - Na počítačích založený informační systém pro získávání, ukládání, analýzu a vizualizaci dat, která mají prostorový vztah k povrchu Země.

1 ÚVOD

V dnešní době se stále častěji setkáváme s perfektními webovými prezentacemi, aplikacemi a službami. Naopak neustále ubývá těch průměrných a podprůměrných. Kvalitní provedení webových stránek už není prioritou jen velkých firem. Díky novým technologiím se prakticky každý může stát součástí virtuálního světa internetu, využívat nepřeberné množství programů a služeb. To, co se ještě nedávno zdálo nereálné, je dnes zcela běžné, technický a technologický pokrok žene svět velmi rychle dopředu. Proto je důležité snažit se držet krok, stále se zdokonalovat, vyvíjet nové programy a aplikace, a poskytovat uživatelům stále jednodušší a užitečnější služby. Jednou takovou službou by mohla být i „Interaktivní mapa Technické univerzity v Liberci“. Zjednodušení vyhledávání učitelů, učeben a přednáškových sálů jistě ocení jak současní, tak i budoucí studenti a pedagogové.

Tato práce, která má za cíl zhotovit interaktivní mapu budov Technické univerzity v Liberci, se skládá ze tří částí. Část teoretická nastíní nutnou analýzu a návrh internetového projektu. Dále jsou zde uváděny možnosti programového zpracování jednotlivých funkčních částí projektu, vhodnost použití jednotlivých programovacích technik a jejich správný výběr a použití. V kapitole věnované problematice programovacích technik jsou představeny nové trendy a srovnání jejich výhod oproti starším verzím. Praktická část je věnována popisu samotné aplikace. Je zde rozebrán fyzický model použité databáze, jsou vyzdviženy možnosti, přednosti a složení aplikací. Dále je podrobně analyzován účel jednotlivých funkcí a jejich samotný kód. Závěr práce je věnován zhodnocení projektu a jeho přínosu pro zlepšení orientace návštěvníků webových stránek Technické univerzity v Liberci a možnostem rozšíření projektu.

2 TEORETICKÁ ČÁST

Současný stav webové prezentace Technické univerzity v Liberci

V současné době Technická univerzita v Liberci (dále jen Technická univerzita nebo univerzita) disponuje kvalitní webovou prezentací univerzity, která obsahuje přehlednou navigaci a dostatek informací pro celou cílovou skupinu návštěvníků. Naopak možnosti vyhledávání na hlavní stránce Technické univerzity (www.tul.cz) jsou značně omezené. Vztahují se pouze na obsah základní prezentace školy. Je tedy vyloučen obsah jednotlivých kateder Technické univerzity. Funkcionalita telefonního seznamu jako pomocné aplikace je velice omezena aktuálností dat, například díky nedávné přestavbě bloku A. K neaktuálnosti dat také přispívá přirozená změna složení pracovníků Technické univerzity a minimální obnova databáze. V tomto roce dochází k modernizaci služby telefonního seznamu.

2.1 Postup pro návrh nového projektu

Jelikož účelem bakalářské práce je zhotovit novou webovou službu, je nutné vypracovat základní dokumentaci před zahájením samotného vývoje. K tomuto účelu slouží jistě bezpočet nástrojů a postupů. Po zvážení možností byl v této práci použit následující postup pro návrh nového projektu.

Složení dokumentace:

- zadání
- analýza
- design
- vývoj
- testování

2.1.1 Zadání

Zadání projektu se zabývá stanovením základních mantinelů. Nejdříve je nutné stanovit cíl projektu. Ten nám především udává ambice a směr. V našich nárocích bychom neměli být skromní, ale je nutné se vyhnout nemožným cílům. Po stanovení toho, čeho chceme dosáhnout, je dobré se inspirovat u již hotových projektů. To nám může přinést utřídění myšlenek a možné doplnění o funkcionality, které mohly být opomenuty. Již v první fázi návrhů je nutné porozhlédnout se po konkurenci. Závěry mohou být dva. Máme místo na trhu pro náš projekt nebo zjistíme, že konkurence je natolik silná a početná, že není rozumné pokračovat v dalším vývoji a investici zaměřit jinam. Toto lze z dokumentace tohoto projektu vypustit. Stejně jako příjmový model, který by následoval. Důležitou otázkou u nového projektu je zejména jeho cílová skupina. V konečné fázi se od ní odvíjí design, funkcionality i způsob propagace. Například pokud máme cílovou skupinu počítačových nadšenců, nemůžeme opomenout RSS kanál pro zprávy. Závěrem úvodní části dokumentace je vytvoření závazku. Ten nám zaručuje dodržení zadání funkcionality projektu. Zahrnuje v několika větách klíčové body.

2.1.2 Analýza

Analýza začíná definicí klíčových funkcí a procesů, které budou běžet na pozadí projektu. Ten doplníme o navazující funkce. Po funkční stránce zbývá dořešit *CRON* funkce. Tedy ty, které poběží v pravidelných intervalech. Jedná se například o generování RSS zpráv nebo *scriptů* běžících na serveru pro generování *sitemapy*. Dobré je rovnou stanovit i časový rozpis tak, aby se procesy navzájem nekryly. Po stanovení všech procesů přejdeme k návrhu konceptuálního a fyzického modelu dat. Tento moment může být klíčový a je nutné ho provést odborně. Špatný koncept může znamenat pomalou aplikaci v případě užívání několika sty uživateli, a v případě vyšší návštěvnosti může znamenat výpadek databáze a tím i celého projektu. V dalším kroku navážeme vytvořením *Data-flow diagramů* jednotlivých funkcí, které nám jasně zobrazí

tok dat.

Analýza typů existujících stránek otevírá druhou část. Na rozdíl od analýzy procesů se zaměříme na plynulou návaznost jednotlivých stránek. Analýza typů stránek dává základ grafikům pro tvorbu drátových návrhů. Touto analýzou navážeme na návrh šablon jednotlivých stránek a především rozmístění funkčních prvků na stránce, včetně případných reklamních bloků. Zde je potřeba myslet především na komfort uživatelů. V základu můžeme počítat s dvěma základními typy stránek. A to se stránkou statickou a stránkou dynamickou. Statická stránka zpravidla obsahuje neměnný obsah na základě uživatelské interakce. Dynamická je naopak závislá na chování uživatele a vrací obsah na základě uživatelské požadavku, například hledání na stránce. To svůj výsledek vrací na stejnou stránku, ale na základě vstupních informací generuje obsah.

SEO specifikace nám zahrnuje širokou problematiku optimalizaci stránky především pro vyhledávače. Jedná se o definici klíčových slov vhodných pro daný projekt. U klíčových slov je dobré pohlídat i konkurenci. Pokud na klíčové slovo „bazar“ cílí celá konkurence, je moudré cílit projekt na přesnější názvy, kde konkurence nebude tak vysoká. Dále zahrnuje definování názvů jednotlivých stránek a to každou stránku zvlášť. Dále i popis stránek. I dnes, kdy se pravidla vyhledávačů stránek stále zpřísňují, má svou váhu správně zvýrazněný text v nesémantických značkách a výstižné nadpisy psané s ohledem na analýzu klíčových slov. V neposlední řadě je nutné již při návrhu myslet na správné pojmenování stránek a zachování tzv. *hezke URL*. Tuto *URL* je například možné generovat z nadpisů stránky, který vytvoříme z obsahu a klíčových slov.

Tato dokumentace by měla obsahovat harmonogram vývoje, zejména pokud se na něm podílí tým lidí, kteří musí spolupracovat. Časový plán nám nejen zajistí efektivnost vývoje, ale určí i základní představu o ceně práce, na níž lze postavit základní rozpočet. Tyto dvě položky se z pochopitelných důvodů v návrhu projektu „Interaktivní mapa TUL“ neobjeví.

Závěr dokumentace analýzy tvoří procesy, které probíhají po spuštění projektu. Jedná se o sledování statistik webu a chování uživatelů. V beta provozu webu je kladen důraz na sledování statistik z nástroje *webmaster tools*, který nám dá konkrétní analýzu obsahu webu. Hlavně upozorní na nedostatky, jako jsou nedostupné stránky a ty, které na ně odkazují. Dále odhalí chybné a nedostupné adresy. Dokumentaci analýzy

zakončíme stanovením procesů údržby a zodpovědnosti. Tedy osob, které budou odpovídat za obsah, moderování diskuzí a jiného obsahu vkládaného uživateli.

2.1.3 Design

V dokumentaci designu v první řadě navrhujeme logo projektu. To by mělo souviset s projektem a jeho zaměřením. Také musíme brát v úvahu korporátní styl, pokud se jedná o firmu. V návaznosti na design loga je vytvořen *layout* základní stránky a posléze všech typových stránek dle analýzy typových stránek. Vše završí definice *CSS* prvků. Nadpisy, okraje, styly tabulek, barva textu, odkazů atd. Design stránky vypovídá více než z poloviny o její kvalitě a mnohem více utváří názor nového návštěvníka. Také se čím dál častěji tvoří účelově k povaze celého projektu.

2.1.4 Vývoj

Postup vývoje plyne z předešlé dokumentace. Prvním krokem je zpracování vybraných mapových podkladů. Dalšími kroky je návrh databáze a programová příprava. Do té lze zařadit přípravu *frameworku* v případě jeho použití, nebo jednotlivé funkce, které nepřímo souvisejí s aplikací. Například připojení k databázi. Do přípravy zahrneme i vytvoření *CSS* a *HTML* a vložení grafiky. Dále je dobré postupovat tvorbou jednotlivých funkčních bloků. Ty otestovat a pokračovat dále. Po dokončení všech částí výslednou aplikaci zkompletujeme do funkčního celku.

2.1.5 Testování

Důležitou fází je samotné testování. To bychom neměli podcenit a věnovat mu dostatečnou dobu. Po detailním testování ve skupině několika lidí, kteří mají odlišné

cítění, zkušenosti a odbornost, můžeme přejít ke spuštění *beta verze* projektu. Ta může například běžet současně se svojí starší verzí, jedná-li se o re-design, nebo s výrazným označením, že se jedná o *beta verzi* projektu. To do jisté míry poskytne dobu, po kterou uživatelé odpustí nějaký ten nedostatek. Není ani od věci poprosit uživatele o vyjádření jejich mínění.

2.2 Použité programovací prostředky

2.2.1 PHP a Framework

Základem celé aplikace byl zvolen skriptovací jazyk *PHP*. Ten se realizuje na straně serveru. Je ho tedy možno popsat jako interpret serveru. Tímto se liší například od jazyků jako *JavaScript*. Nejdůležitější předností skriptovacích jazyků, které běží na straně serveru, je vlastnost předávání prohlížeči pouze výsledku (HTML kódu) a samotný skript je tak uživateli skryt. Nevýhodou je nemožnost reagovat na akce na straně klienta. V minulosti bylo nutností kombinovat jazyk *PHP* s jiným jazykem, který na akce na straně klienta dokáže reagovat. Například *JavaScript*. Dnes se stále více používá technologie *AJAX*.

Rovnocenné volby jsou *CGI* a *ASP* technologie. Jelikož *PHP* není jedinou možností jak realizovat zadanou aplikaci, je v následujícím textu uvedeno několik důvodů, proč byl vybrán skriptovací jazyk *PHP*.

Argumenty jsou již nasbírané zkušenosti s touto technologií. Mezi přednosti jazyka *PHP* patří množství podporovaných technologií, nenáročnost na prostředky, snadná spolupráce s databázovým serverem. Ten je navíc zdarma a má také množství podporovaných protokolů. Pro budoucího provozovatele aplikace je také důležitá podpora *PHP* na většině free-hostingových serverů a serverů používaných Technickou univerzitou.

Obecně o Framework

Pod tímto pojmem si lze představit vývojový softwarový nástroj pro tvorbu nových projektů v *PHP*. Jeho součástí mohou být podpůrné programy nebo *API* knihovny. Mezi služby též patří komunitní poradna, doporučení a pravidla pro vývoj. Základní kámen tvoří *Zend Framework*, dále záleží na každém, zda použije tento základní *framework* nebo jiný, upravený na vývoj konkrétních služeb nebo projektů.

Odpůrci *frameworků* hovoří o jeho neefektivnosti při zpracování dat. Tento argument má své opodstatnění u specializovaných aplikací. Zde může používání obecných metod nebo tříd znamenat pomalejší *script*. Hlavním důvodem proč se firmy přiklánějí k řešení svých aplikací přes vlastní nebo volně dostupný *framework* je úspora času a tím i financí.

Struktura *frameworku* se dělí na *frozen spots* a *hot spots*. Zatímco první jmenované jsou zpravidla pevnou součástí všech *frameworků*, *hot spots* jsou závislé na úpravě programátora a jejich funkce a výstup se více či méně liší.

2.2.2 XHTML

Celý projekt „Interaktivní mapa TUL“ zasadíme do běžné webové stránky. Tu je nutné vytvořit v nějakém standardu. Máme na výběr od různých standardů *HTML*, tak i *XHTML*.

Co je XHTML

Definice říká, že *XHTML* je reformulací *HTML* jako aplikace *XML*. Definice nám tedy říká, že *XHTML* vychází z *XML*. *Extensible Markup Language* (rozšiřitelný značkovací jazyk) je značkovací jazyk pro vytváření dokumentů. Zjednodušeně se dá říci, že *XML* slouží k předávání strukturovaných informací ve srozumitelné formě napříč interpretem.

Ukázka XML

```
<?xml version="1.0" encoding="utf-8" ?> // verze xml a nastavení kódování
<table name="makers"> // element začátku tabulky
  <row idMakers="1" lat="50.77043916" lng="15.08728610" label="Koleje TUL" image="koleje.jpg" ulr="http://" /> // element
  záznamu jednoho řádku databáze s parametry obsahující data ve sloupcích
  <row idMakers="2" lat="50.77278700" lng="15.07210861" label="Budova A" image="" ulr="" />
  <row idMakers="3" lat="50.77277600" lng="15.07522861" label="Budova F" image="" ulr="" />
  .
  .
  .
  .
</table> // konečný element ukončující tabulku
```

XML má velmi přísná pravidla narozdíl od standardu *HTML*. Například v kódu rozlišuje malá písmena od velkých, takže značky je třeba psát přesně tak, jak byly definovány v konkrétním jazyce. Aplikací jazyka *XML* je velké množství. Jmenovitě můžeme uvést *Dockbook* nebo již zmíněné *XHTML*.

XHTML oproti *HTML* nic nového nepřináší. Jeho použití znamená striktnější dodržování pravidel psaní kódu.

Ukázka XHTML

```
<html xmlns="http://www.w3.org/2002/06/xhtml1/" version="XHTML 2.0" xml:lang="en"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/1999/xhtml
  http://www.w3.org/MarkUp/SCHEMA/xhtml1.xsd">

  <object id="pdf-object"
    name="pdf-object"
    type="application/pdf"
    data="http://www.w3.org/TR/xhtml1/xhtml1.pdf"
    width="100%"
    height="500">

  </object>
```


2.2.3 AJAX - Asynchronous JavaScript and XML

Termín *AJAX* se objevil již v roce 2005 v článku Jesse James Garreta. Článek s názvem „*AJAX: Nový přístup k webovým aplikacím*“ stanovil nový a komplexní pohled na již starší vyslovené myšlenky z roku 1996. Mezi tyto první myšlenky lze zařadit například *iframe*.

Technologie z významu svého názvu uvádí spojení technologií *JavaScript* a *XML* spojené asynchronní komunikací. V praxi to znamená možnost načtení libovolných dat bez nutnosti obnovení celé webové stránky.

AJAX udává nový směr v možnostech webových stránek. Jejich uživatelská přívětivost dosahuje nového rozměru. Uživatel získává dojem jednolité aplikace. Například u mapových aplikací to znamená načtení pouze viditelné části mapy, při posunu po mapě se nové části načítají bez obnovy zbytku stránky.

Technologie *AJAX* přináší několik nevýhod. První z nich je možnost zvýšení počtu *http* požadavků. Výsledkem může být zpomalení webové stránky. Je tedy nutné implementovat *AJAX* na správné funkce webu. Mnohé moderní webové stránky založené na technologii *AJAX* působí jako ucelené aplikace se složitou vnitřní logikou. V takovéto aplikaci nelze jednoduše navigovat vpřed a zpět. Dnes lze tento nedostatek částečně řešit ukládáním historie procházení stránky do globálních proměnných a zpětně je vyvolat. Nástrahou může být připojení s vysokou latencí. Je tedy dobré dát uživateli jasně najevo, že aplikace zpracovává data. Předejde se tak opakovanému požadavku ze strany uživatele a tím zatížení serverů.

2.2.4 MySQL

Použití tohoto typu databází vyplynulo z rozhodnutí použití skriptovacího jazyka *PHP*, které s *MySQL* spolupracují nejlépe. Oproti své konkurenci, jako jsou databáze typu *Oracle* nebo *MS Acces*, je řešení od *MySQL* zdarma.

MySQL

Z návrhu aplikace plynou jisté nároky na ukládání dat, které bude možno snadno

editovat v připraveném administračním prostředí. Ukládat budeme informace o mapových podkladech budov, informace o pracovnících a osazení jednotlivých místností. Dále je nutné zaručit co nejméně náročné vyhledávání. Tyto nároky by šly jen velice obtížně řešit jiným způsobem než za použití databází.

Správný návrh databází je alfou a omegou dnes již jakéhokoliv projektu, který databáze využívá. Co se při několika stech návštěvách může projevit pomalejšími reakcemi stránek, může u desítek tisíc dotazů na databázi v jednom okamžiku způsobit padání databázových serverů a nefunkčnost části nebo celého projektu. Nejen z těchto důvodů je nutné dodržovat několik základních pravidel při vytváření databází.

Základem je dobře si rozmyslet potřeby projektu. Nemíchat informace, které k sobě nepatří a podobně. Tabulky se vyplácí dělat co nejjednodušší. To samé platí o datových typech. Poskytnout jim pouze dostatečný rozsah. Zpravidla tabulky opatřovat automatickým indexem. Nepsaným pravidlem je také dodržovat kulturu jednotlivých nadpisů.

Pomoc nám mohou normální formy. Jedná se o pravidla, se kterými lze dosáhnout lepšího návrhu databázového systému. Dá se tedy říci, že čím je tabulka ve vyšší normální formě, tím je kvalitnější.

„Normální formy:

- **0NF** (nultá normální forma): Tabulka je v nulté normální formě právě tehdy, existuje-li alespoň jedno pole, které obsahuje více než jednu hodnotu.
- **1NF** (první normální forma): Tabulka je v první normální formě, jestliže lze do každého pole dosadit pouze jednoduchý datový typ (jsou dále nedělitelné).
- **2NF** (druhá normální forma): Tabulka je ve druhé normální formě, jestliže je v první a navíc platí, že existuje klíč a všechna neklíčová pole jsou funkcí celého klíče (a tedy ne jen jeho částí).
- **3NF** (třetí normální forma): Tabulka je ve třetí normální formě, jestliže každý neklíčový atribut není transitivně závislý na žádném klíči, neboli je-li ve druhé normální formě a zároveň neexistuje jediná závislost neklíčových sloupců tabulky.

- **BCNF** (Boyce-Coddova normální forma): Tabulka je v Boyce-Coddově normální formě, jestliže pro každou netriviální závislost $X \twoheadrightarrow Y$ platí, že X obsahuje klíč schématu R .
- **4NF** (čtvrtá normální forma): Tabulka je ve čtvrté normální formě, je-li ve třetí a popisuje pouze příčinnou souvislost (jeden fakt).
- **5NF** (pátá normální forma): Tabulka je v páté normální formě, pokud je ve čtvrté a není možné do ní přidat nový sloupec (skupinu sloupců) tak, aby se vlivem skrytých závislostí rozpadla na několik dílčích tabulek.¹

Mezi jednotlivými tabulkami databáze je nutné vytvářet určité vztahy. Těmto vztahům říkáme relace a můžeme využít třech základních typů:

- **Relace 1:1** je relace nejjednodušší. Ve své podstatě říká, že jeden primární index hlavní tabulky odpovídá jedinému záznamu (indexu) druhé tabulky. V praxi lze tento případ ukázat na faktuře a platbě za ní. Kdy jedna faktura bude pouze jednou zaplacená.
- **Relace 1:N** je dalším typem relace taktéž mezi dvěma tabulkami. Tentokrát jednomu primárnímu indexu hlavní tabulky odpovídá žádný nebo několik záznamů v druhé tabulce. Můžeme uvést v hlavní tabulce záznam o osobě a ve druhé telefonní čísla, kde nemusí mít přiřazeno žádné číslo, ale také jich může být několik.
- Posledním typem je **relace M:N**. Ta vzniká mezi tabulkami, kde každý záznam může odpovídat několika záznamům z druhé tabulky. Pro příklad lze uvést knihovnu. Zde může kniha mít několik autorů a jeden autor může napsat několik knih.

Data lze z databáze získat pomocí dotazu na databázi. K těmto dotazům se používají *SQL* dotazy. Kvalitu navržené databáze lze do jisté míry určit dobou, jakou databáze potřebuje pro odpověď na Váš dotaz.

¹ <http://www.rydval.cz/phprs/view.php?cislocclanku=2005123127>

2.3 Výběr a zpracování mapových prostředků

Výběr mapových podkladů a jejich zpracování

Jak se ukázalo v pozdější době, výběr mapových podkladů byl nejdůležitějším aspektem celé této práce. Původní předpoklady se diametrálně odlišují od reálné skutečnosti. Celá problematika je rozebrána v této části kapitoly.

Nabízené možnosti:

- Zapůjčení mapových podkladů od katedry geografie.
- Zapůjčení mapových podkladů od ČÚZK.
- Zapůjčení potřebných mapových podkladů od soukromého nebo státního subjektu.

Požadavky na mapové podklady

Z návrhu aplikace a zadání bakalářské práce „Interaktivní mapa TUL“ došlo k následujícím požadavkům na mapové podklady. Bylo zapotřebí získat geografickou mapu Liberce a přilehlého okolí pro první dvě úrovně přiblížení. Dále dvě vrstvy turistické mapy Liberce a okolí pro lepší orientaci ve městě. Neúspěšně dopadla žádost na katedře geografie Fakulty pedagogické TUL. Široká problematika vlastnických práv mapových podkladů nedovoluje tyto podklady pro tuto práci použít. A to zejména pro nákladnost a složitost jejich pořízení.

Český úřad zeměměřický a katastrální má vlastní program pro studenty bakalářského a magisterského studia. Za účelem závěrečných prací poskytuje mapové podklady České republiky. Omezení se týkají poskytnutím maximálně 10 mapových listů ZM10 a rastrové mapy pouze středních měřítek. Zejména dostupná měřítko jsou pro uživatele omezující. A pro tento projekt tedy nedostatečným řešením.

Poslední volbou byla společnost *Seznam.cz a.s.*, která je provozovatelem portálu www.mapy.cz. Tato společnost je častým zadavatelem bakalářských a diplomových prací a její ochota spolupracovat je velká. Bohužel není vlastníkem mapových podkladů provozovaného portálu. Těmi jsou společnosti *PLANstudio*, *GEODIS BRNO* a

SHOCart. Prosba o poskytnutí potřebných mapových podkladů byla neúspěšná.

V této fázi jsou vyčerpány veškeré možnosti získání dat pro vlastní mapovou aplikaci. Při podrobnějším přezkoumání i velké společnosti, vlastníci projekty obsahující mapové aplikace, používají jedno z následujících řešení. Jedná se o řešení použití mapového *API*. Taková vývojová prostředí pro mapové aplikace nabízí společnosti *Seznam.cz, a.s., Centrum holdings* nebo *Google*.

Srovnání možností mapových API

- *API mapy.cz* společnosti *Seznam.cz, a.s.* najdeme na adrese <http://api.mapy.cz/>. Rozsahem poskytuje mapu celé Evropy v pohledu základním, turistickém a také fotomapu. Použití tohoto řešení přináší několik omezení. Významná pro projekt „Interaktivní mapa TUL“ jsou následující:

- použití pro nekomerční provoz
- k dispozici jsou omezené mapové podklady
- provoz je omezen na 1000 zobrazení denně

Lze ho doporučit pro firemní stránky a podobné méně navštěvované stránky, kterým hlavní omezení na 1000 zobrazení nebude překážkou. Pro potřeby Technické univerzity je to nedostatečné.

API Amapy.cz společnosti *Centrum holdings*, poskytuje obdobné prostředí jako již zmíněné *API* Seznamu. Narozdíl od svého konkurenta na českém trhu neomezují své uživatele tisícem zobrazení za den. K velkému plusu můžeme přiřadit rychlost načítání a téměř neomezené funkce oproti konkurenci. Pro český trh lze použití tohoto API jen doporučit. Samozřejmě plně dostačuje i nárokům projektu „Interaktivní mapa TUL“.

Poslední možností je použití mapového *API* od světové jedničky společnosti *Google*. Možnosti tohoto řešení jsou prakticky neomezené. Můžeme také jistě vyzdvihnout velkou komunitu a velké množství příkladů a návodů pro vývojáře. Jistým omezením je absence češtiny v podporovaných jazycích a tedy i dokumentace v anglickém jazyce. Toto omezení již neplatí pro uživatele výsledné služby. *API* rozezná jazykovou mutaci systému uživatele a přizpůsobí se. Novinkou je uvolnění verze pro flash aplikace nebo možnost vkládat videa do map.

Konečný výběr *API* od *Google* proběhl na základě předchozích zkušeností s tímto prostředím. Do budoucna je to sázka na jistotu z pohledu dalšího vývoje a kvality map. Již dnes se některá města převádějí do 3D pohledu. A je tedy pravděpodobné, že i Liberec bude takto nafočen. Navíc se dá předpokládat technologický a obsahový náskok nad ostatní konkurencí.

2.4 Mapový server

Za spojením mapový server si každý z nás může představit mnoho věcí a také se jich za ním mnoho skrývá. V nejširším slova smyslu znamená kompletní řešení zobrazování map v síti zahrnující hardware, software a programové řešení. Jedná se o aplikaci, která pracuje principem klient-server. Služba umožňuje sdílet geografické informace v podobě rastrových map. Ty se zpravidla skládají z jedné nebo více vrstev. Výsledný obraz je předložen uživateli v libovolném formátu. Příkladem mohou být kompletní řešení našich předních serverů www.seznam.cz nebo www.centrum.cz v podobě služeb amapy.cz a mapy.cz

Princip funkčnosti:

- skript vracející obrázkové mapy na základě vstupních požadavků (*PHP*, *CGI*)
- skript vracející šablony hotové internetové stránky s mapou
- lze použít i hotová řešení např. *MapScript*, *PHP* nebo *Python*.

GIS

GIS neboli geografický informační systém je velice důležitý pojem představující systém obstarávající získávání, ukládání a vizualizaci mapových podkladů. Tato data jsou definována geometrií, topologií, atributy, a dynamikou. Na základě dat, která jsou nejdražší součástí systému, dokáže vygenerovat rozmanité modely. Ty mohou například sloužit pro plánování výstavby infrastruktury nebo mohou určit rozsah záplav.

Součásti GIS

- hardware
- software
- data
- pracovníci

Vektorové mapové vrstvy

Mapové aplikace mají možnost několika zobrazení. Mapu lze zobrazit jako foto-mapu, turistickou mapu nebo je lze kombinovat. Již standardem je také zobrazování aktuálních informací o dopravě, počasí nebo stavu vozovek.

Na rozdíl od foto-mapy, která se skládá z jednotlivých vrstev čtverců představujících části fotky, se například turistická mapa výrazně liší. Skládá se z vektorových mapových vrstev. Ty jsou ukládány v podobě bodů a čar. Bod definuje polohu a nemá žádný rozměr. Čára je spojnice těchto bodů. Tato vektorová data jsou ukládána v těchto modelech:

- špagetový model
- hierarchický model
- topologický model

Rastrové mapové vrstvy

Jedná se o rastry orientované do spojitých vrstev. Příkladem může být jedno podlaží budovy nebo nadmořská výška. Rastr je rozdělen na nekonečně mnoho malých stejných dílů. Tvar těchto dílů může být různý (čtverec, obdélník, trojúhelník, šestiúhelník). Nejčastěji se používá čtvercové dělení. Zpravidla jsou jednotlivým částem rastru přiděleny hodnoty. Dle typu hodnoty lze rastrové vrstvy rozdělit na:

- rastrové vrstvy výčtového typu
- rastrové vrstvy hodnotového typu

Rastrové vrstvy výčtového typu mají zpravidla celočíselnou hodnotu, která představuje kód sledované veličiny. Nedílnou součástí tohoto typu je dekodovací tabulka. Rastrové vrstvy hodnotového typu mají přiřazenou hodnotu dle číselného datového typu (integer atd.).

3 PRAKTICKÁ ČÁST

3.1 Zadání projektu

Zadání projektu lze rozdělit na několik částí. Vhodné je začít záměrem projektu, poté zhodnotit konkurenci, dále určit cílovou skupinu projektu a následně zhotovit popis projektu.

3.1.1 Záměr projektu

Záměrem projektu „Interaktivní mapa TUL“ je přinést studentům, pracovníkům a návštěvám Technické univerzity v Liberci možnost snadné orientace mezi budovami univerzity. Dále usnadnit navigaci v jednotlivých budovách. Vše za pomoci grafické interaktivní mapy a plánu jednotlivých budov. K možnostem projektu patří snadné nalezení kanceláře konkrétní osoby nebo učebny a jako výsledek zobrazení správného plánu budovy.

3.1.2 Konkurence a inspirace

Na začátku každého projektu má být dobrý nápad nebo potřeba nové služby. Inspiraci je vhodné čerpat u funkčně podobných projektů. V tomto konkrétním případě žádná předešlá aplikace, která by vyhovovala nárokům zadání, neexistuje. Do jisté míry by jí mohla být obecná mapa Liberce a okolí. Mezi inspirativní aplikace patří provedení světových map společností *Google Inc.* Konkrétně bezpočet aplikací využívajících *API* této společnosti. Jako příklad povedené mapové aplikace je prezentace www.bezrealitky.cz, která mapu využívá pro vyhledání volných bytů k prodeji na území České republiky. Tato prezentace využívá pouze možnosti mapového *API*. Na rozdíl od toho, interaktivní plán budovy je dostatečně zpracován v rámci prezentace města Jirkov

(<http://www.jirkov.cz/mestsky-urad/planek-uradu/>).

3.1.3 Cílová skupina projektu

Zde předpokládáme využití aplikace jako služby stránek Technické univerzity v Liberci. Máme přesně definovanou cílovou skupinu studentů, zájemců o studium a hostů Technické univerzity v Liberci. Této cílové skupině není nutno přizpůsobovat prostředí projektu nebo jeho funkcionalitu. A pokud se dodrží přehlednost aplikace a minimum funkčních prvků, bude možné vypustit ostatní podpůrné aplikace. Touto podporou je myšlena kontextová nápověda nebo „našeptávač“ ve vyhledávání.

3.1.4 Popis projektu

Projekt „Internetová mapa TUL“ je mapový systém zobrazující všechny budovy Technické univerzity, založený na *API* společnosti *Google Inc*. Základem je centrální mapa s možností posunu po mapě, přiblížení, oddálení. Všechny akce budou integrovány na ovládací prvky na okraji mapy. Zároveň bude možné vše ovládat myší. Nedílnou součástí bude přechod ze základní turistické, geografické či kombinované mapy do plánů jednotlivých budov Technické univerzity. Ty lze procházet od přízemí směrem vzhůru, vždy po jednotlivých patrech. Plány budov musí obsahovat popis všech důležitých místností. Funkce doplní jednoduché vyhledávání budov, místností a pracovníků Technické univerzity. Celý projekt podpoří administrace.

Použité techniky

- (X)HTML, XML, MySQL, PHP, AJAX, CSS

Vzor pro práci:

- <http://maps.google.cz>, www.mapy.cz

Funkce:

- mapa Liberce a okolí
- posun po mapě s částečným načítáním
- dva módy mapy: turistická mapa TUL mezi budovami a mapa budov
- plynulý přechod od jistého bodu zvětšení, nebo výběrem budovy

Vzhled:

- geografická mapa Liberce v takové velikosti, aby byly vidět všechny budovy TUL
- budovy TUL zvýrazněny
- po prvním zvětšení přechod na turistickou mapu
- 2D zobrazení map
- u módu zobrazení detailu budovy odpovídá jeden krok zvětšení resp. zmenšení přechodu na nižší resp. vyšší patro

Vyhledávání:

- budovy a místnosti řazeny v databázích pro snadné vyhledávání
- formuláře pro změnu nebo nastavení aplikace (jednoduché administrační prostředí)
- možnost vyhledávání jednotlivých místností
- možnost vyhledat jednotlivé pracovníky

3.2 Analýza

V této fázi je nutné analyzovat klíčové funkce a procesy, navazující funkce, CRON funkce. Dále je třeba stanovit konceptuální a fyzický model dat a vytvořit Data flow diagramy. Následně pak provést analýzu typů stránek existujících na webu, SEO specifikaci, obsáhnout sledování statistik webu a určit personální entity.

3.2.1 Klíčové funkce, procesy

Mezi klíčové procesy projektu patří zobrazení mapy Liberce a okolí. Je použito mapové *API* společnosti *Google Inc.* Zvětšení mapy je nastaveno na takovou hodnotu, která zaručuje viditelnost všech budov univerzity.

Druhý klíčový proces zobrazuje detail jednotlivých budov. Funkce generuje plán budovy, výsledný plán je rozdělen na několik stejných čtverců, ze kterých se generuje. Rozdělen je z důvodu rychlejšího načítání výsledného plánu a možnosti detail plánu generovat po menších částech.

Použité mapové *API* dovoluje zvětšení resp. zmenšení mapy. Dále načítání mapy po jednotlivých částech. Plán budovy nabízí jeden stupeň zvětšení a načtení části mapy. Při posunu po plánu pomocí ovládacích kláves načte vždy část mapy pomocí technologie *AJAX*.

Proces vyhledávání je řešen pomocí *SQL* dotazu v databázi. Umožňuje vyhledat pracovníka nebo místnost dle obsahu databáze. Pokud by testování ukázalo nutnost fulltextového vyhledávání, dalo by se toto řešit pomocí aplikace *Sphinx*. Jedná se o volně šiřitelné řešení fulltextového vyhledávání.

Celý projekt podporuje administrace. Je řešena přístupem přes zabezpečenou stránku. Šifrovací funkce používá porovnání zadaného hesla se záznamem v databázi. Je použita šifra *MD5*. Administrace obsahuje možnost změny středu mapy, vložení nových významných míst na mapu, včetně obsahu. Dále nabízí také správu plánů budov a kontaktů.

3.2.2 Popis navazujících funkcí

Mezi navazující funkce je počítáno s vložením *google analytics* pro kontrolu chování uživatelů a možné rychlé odhalení chyb uživatelského rozhraní. Dále s použitím *webmaster tools*, který web prověří z pohledu dostupnosti a existence jednotlivých stránek. Mimo jiné podá oznámení vyhledávači *Google* o existenci nového projektu a dojde k rychlejšímu zaindexování stránek.

3.2.3 CRON funkce

Projekt nebude obsahovat procesy, které poběží v pravidelných intervalech nebo stále na pozadí projektu.

3.2.4 Konceptuální a fyzický model dat

Základní entity

Aplikace je závislá na datovém zdroji stahovaném ze serverů společnosti *Google Inc.* Jedná se o kompletní mapové podklady. Jelikož je poskytnutý zdroj dat zdarma, nezaručuje poskytovatel žádnou podporou. Krátkodobé výpadky jsou možné. Dlouhodobý výpadek se nepředpokládá. V jeho případě by bylo možné v krátké době upravit aplikaci na použití jiného mapového zdroje. V této době by aplikace sloužila bez hlavní mapy, s přístupem k plánům budov přes přímý odkaz.

Zdrojem všech ostatních potřebných dat je vlastní databáze. Záloha dat bude řešena přímo na hostingu každodenní zálohou dle možností hostingového serveru. Jedná se o informace o pracovnících, jednotlivých místnostech v budovách a bodech zobrazených na hlavní mapě.

Pomocné upřesňující entity

Potřebné informace pro zobrazení hlavní mapy jsou GPS souřadnice středu této mapy. Střed mapy je nastaven na souřadnice N50.764911 E15.054016. Souřadnice jsou ve formátu *decimal*. Jedná o formát snadno převoditelný ze známějšího formátu *DMS*. Na rozdíl od tohoto formátu se snadněji ukládá v databázi a lépe se s ním pracuje. Poslední nutnou hodnotou pro nastavení základní mapy je velikost přiblížení. To může nabývat hodnot od 1 do 20. V případě tohoto projektu je základní hodnota nastavena na 13. Kombinací souřadnic a velikosti přiblížení je zobrazení mapy Liberce a okolí v takovém rozsahu, že jsou viditelné všechny současné budovy Technické univerzity. Každá budova zobrazená na hlavní mapě formou významného bodu potřebuje souřadnice polohy, opět uložené ve formátu *decimal*. Krom souřadnic je nutné uložit popis objektu, zobrazovaný obrázek, odkaz, a v případě více druhů budov také druh

grafické ikony na mapě.

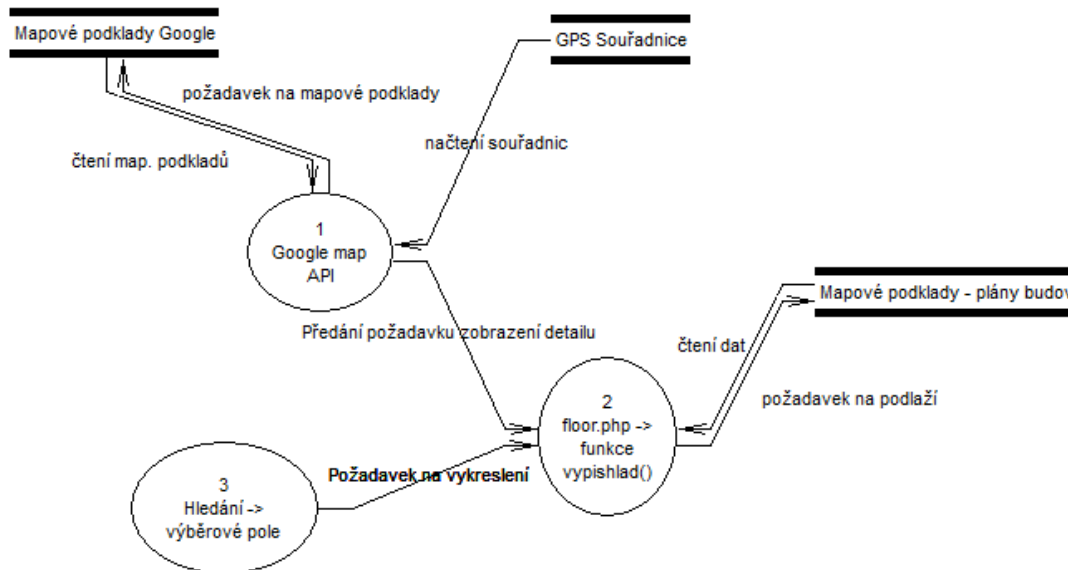
Do plánu každé budovy Technické univerzity je nutno uložit počet čtverců, ze kterých se plán skládá a číslo hladiny. Tyto informace jsou nutné pouze pro správné vykreslení hladiny (podlaží) plánu. Samozřejmostí jsou uloženy jednotlivé plány budov. Dělí se na jednotlivá patra (hladiny) a ta jsou rozdělena do shodných čtverců ve formátu *.gif*.

Poslední tabulka v databázi obsahuje data o pracovnících. Ke kontaktním údajům je nutné přiřadit informace o budově a místnosti, ve které pracovník sídlí. Z těchto údajů program určuje pozici pracovníka na mapě.

3.2.5 Data flow diagramy

Data flow diagram, čili diagram datových toků je jedním z nástrojů pro modelování funkcí systémů. Data flow diagram se skládá z procesů, toků, skladů a terminátorů.

Proces představuje část systému, která mění vstupy na výstupy. Symbolem takového procesu je kruh, ovál nebo obdélník. Tok znázorňuje přesun informací z jedné části systému do jiné. Jeho symbolem je šipka. Sklad slouží k uložení dat k pozdějšímu použití. Jeho symbolem jsou dvě vodorovné čáry. Terminátor představuje externí entity, které komunikují se systémem a stojí vně systému. Data flow diagram pro tuto práci lze vidět na *Obr. 3.1*.



Obr. 3.1 Data flow diagram

3.2.6 Analýza typů stránek existujících na webu

Projekt je dělen na statické a dynamické stránky. Hlavní stránka projektu je statická, obsahuje ovšem dynamický prvek hlavní mapy. Ten je řešen technologií *AJAX* a ve své podstatě při změně mapy nenačítá novou stránku, ale aktualizuje vyhraněnou část uzavřenou v *tagu div*.

Druhým typem stránky je plán budovy. Jedná se o dynamickou stránku, její obsah se mění v závislosti na požadavku uživatele. Zobrazuje jednotlivá patra budov dle předchozího požadavku.

3.2.7 SEO specifikace

Zde je dobré začít návrhem klíčových slov. U projektu, který neobsahuje větší množství textu, jako v případě mapové aplikace, je významnější SEO optimalizace obtížná. Navíc převážné množství textu je skryto v popisu mapy, který je vykreslován

ajaxovou nebo javaskriptovou funkcí a tudíž pro vyhledávací roboty je tento text skryt. Přesto zde pojmenujeme jednotlivé budovy s příznakem názvu univerzity. Př. „Budova A – Technická univerzita v Liberci“

Další optimalizace je zaměřena na výstižný titulek hlavní stránky a její popis.

```
<title>Interaktivní mapa Technické univerzity v Liberci</title> //titulek stránky
<meta name="keywords" content="mapa, Technická, univerzita, TUL, interaktivní, budovy TUL">
<meta name="description" content="Interaktivní mapa Technické univerzity umožňuje snadnou orientaci mezi budovami Technické univerzity a poskytuje přehled rozmístění místností v budovám.">
```

3.2.8 Sledované statistiky webu

Statistiky jsou sledovány službou *google analytics*. V této službě využijeme nastavení cílů. Jejich počet je omezen na 4. Budou tedy průběžně přenastavovány na všechny důležité cíle projektu. Těmi jsou jednotlivé hladiny plánů budov. K těmto cílům je zvolen přístup přes hlavní mapu a přes vyhledávání.

V nástroji *webmaster tools* jsou sledovány tyto statistiky:

- chyby v http
- nedostupné adresy URL
- nenalezené stránky, na které existuje odkaz

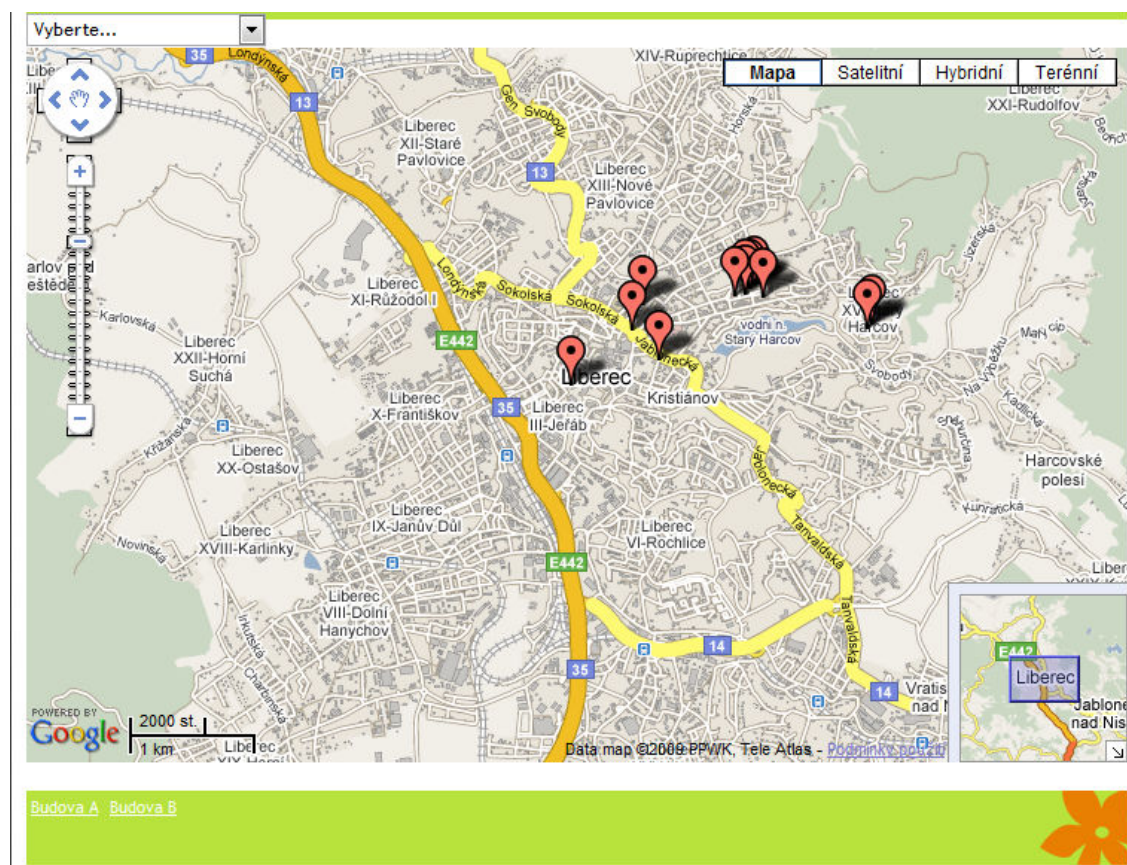
3.2.9 Personalizační entity

U aplikace je nutné oddělit možnosti běžných uživatelů od práv administrátora. K této potřebě přidáme dále „super uživatele“. Jeho práva se nebudou vztahovat k aplikaci, ale bude mít přístup k samotné databázi, administračnímu prostředí hostingu a ftp přístup. Druhý uživatel je takzvaný administrátor, který dostane přístup k administraci aplikace. Nejnižším stupněm je neregistrovaný uživatel, který má možnosti využívat všech služeb bez změny obsahu.

3.3 Design

Výsledný vzhled projektu je řešen pouze v *CSS*. Jedná se o vytvoření stylů pro *tag div*, který je zarovnán na střed stránky a je mu dán bílý okraj. Ve spodní části všech stránek je vložen *div* s přímým přístupem k jednotlivým interaktivním plánům budov. Obsahuje opět v *CSS* definované zelené pozadí a bílé ohraničení. Tento *div* je dobře patrný v levé spodní části na *Obr. 3.3*. Na *Obr. 3.4* je znázorněn detail budovy. Ten se zobrazí po přechodu z hlavní mapy.

Tato minimální grafika je zvolena za účelem snadného přizpůsobení k již hotové prezentaci univerzity nebo libovolného institutu, který by se rozhodl toto řešení použít.



Obr. 3.3 Zobrazení hlavní mapy



Obr. 3.4 Zobrazení detailu budovy

3.4 Vývoj

Vývoj projektu začneme zpracováním mapových podkladů a dále budeme pokračovat návrhem databáze a vytvořením potřebných tabulek dle předchozího návrhu. Výsledná databáze je obohacena o testovací data. Tato data jsou vložena s ohledem na výsledné možnosti databáze tak, aby obsahovala charakteristická data. Vývoj zakončíme výběrem programovacích technik. Jádro projektu je naprogramováno ve skriptovacím jazyku *PHP*. O databáze se postará *MySQL* a interpretaci dat technologie *Javascript*, *(X)HTML*, *CSS* s *AJAX*. Důvody a srovnání možných řešení lze nalézt v teoretické části práce.

Následně naprogramujeme základní funkce. Například funkce pro spojení k

databázi nebo základní třídu pro použití *AJAXU*. Aplikace je složena z jednotlivých funkčních částí. Ty programujeme zvlášť a po každém dokončení funkcionality otestujeme. Poslední fází je vytvoření *CSS* a finálního *HTML* kódu.

3.4.1 Zpracování mapových podkladů

Mapy budov

Druhou půlkou obsahu jsou jednotlivé budovy Technické univerzity a jejich procházení. Slepé mapové podklady všech budov a jejich pater byly součástí materiálu k této bakalářské práci.

Postup zpracování map budov

Jednotlivé mapy pater budov byly přesunuty na stejně velké rastry a byly z nich vytvořeny totožně velké čtverce. Následně byly rozděleny na jednotlivé čtverce. K tomu byl použit program *Corel Photo Paint*. Vzniklé obrázky jsou systematicky pojmenovány *RxCy.gif* kde „x“ představuje číslo řádku a „y“ číslo sloupce. Toto řešení usnadnilo načítání jednotlivých obrázků při změně mapy bez nutnosti ukládání informací o mapách do souboru či databáze. Zároveň byl zvolen formát, který umožňuje minimální velikost každého ze čtverců. Jednotlivé mapy pater byly doplněny o aktuální čísla místností v jednoduchém grafickém editoru.

3.4.2 Použité programovací prostředky

PHP

PHP je skriptovací programovací jazyk provádějící skripty na straně serveru. Nejčastěji se kombinuje s značkovacími jazyky jako jsou *HTML* nebo *XHTML*. V syntaxi jazyka lze hledat příbuznost u jazyků *C*, *Pascal* a také *Java*. Značnou výhodou je jeho nezávislost na platformách. K výhodám lze také přiřadit podporu mnoha knihoven nebo existence celých *frameworků*. Dále je zde také velká obliba

programátorů a z toho plynoucí vysoká komunitní podpora. V neposlední řadě je předností podpora freehostingových serverů nebo existence snadné instalace lokálních serverů.

Využití technologie AJAX v mapové aplikaci

Jak bylo již uvedeno výše, aplikaci „Interaktivní mapa TUL“ je rozdělena do dvou rovin. Mapa Liberce a okolí a plán jednotlivých budov Technické univerzity. Zatímco první část obstarává API vytvořené společností Gogole, druhá část je vytvořena nově.

Pokud vezmeme v úvahu rozsah mapových podkladů potřebných pro rozsah aplikace, bylo by nemyslitelné tato data načítat najednou. Zde využijeme možnosti *AJAXu* a budeme načítat pouze data, která aktuálně uživatel potřebuje, a to po jednotlivých čtvercích mapy. Tímto dosáhneme při každé akci vyvolané uživatelem nejmenšího přenosu dat.

3.4.3 Praktický popis programového řešení projektu

Celý program je rozdělen do několika částí:

- zdroj dat
- program obstarávající základní mapu
- program obstarávající detailní mapu
- administrační prostředí

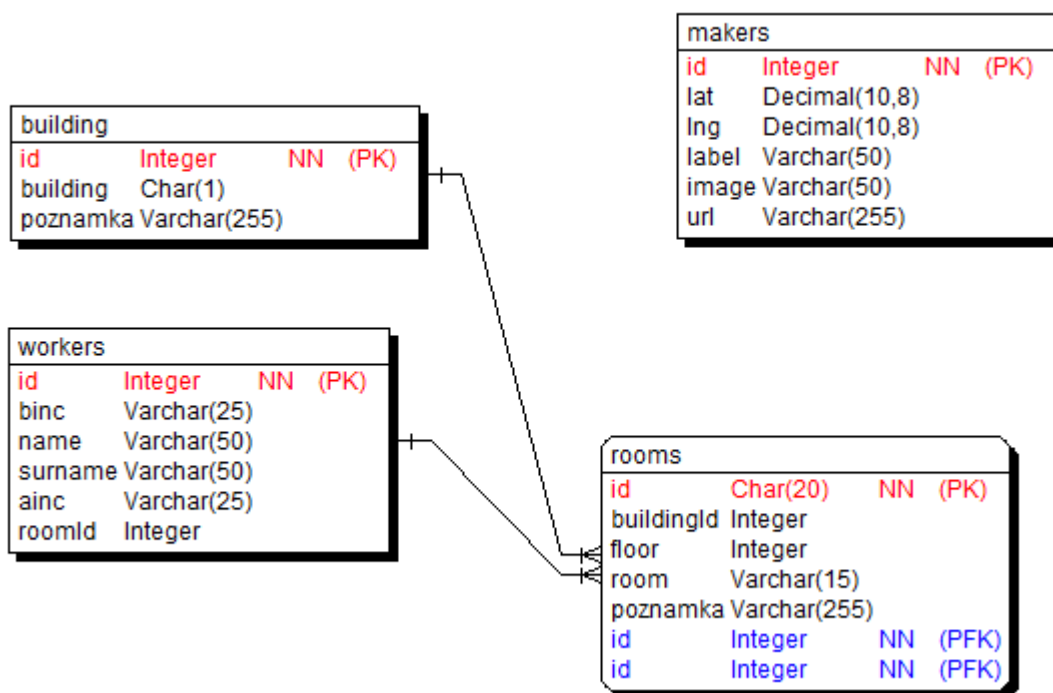
Zdroj dat

Zdrojem dat je již zmíněná databáze *MySQL*. Databáze je nazvaná „*mapa*“ a obsahuje následující tabulky. Tabulka „*makers*“ je hlavní tabulkou pro základní mapu. Obsahuje primární index *id* typu *int(4)*. Dále tabulka obsahuje sloupce *lat* a *lng* typu *decimal(10,8)*. Obsahují GPS souřadnice jednotlivých důležitých bodů na hlavní mapě. Dále obsahuje *label*, *img* a *text*. Všechny popisují obsah informační bubliny, která obsahuje nadpis, obrázek a krátký popis. Tuto kontextovou informaci získáme po

kliknutí na ikonu budovy.

Tabulka „*mapa*“ udržuje informace o základním nastavení hlavní mapy. Důležitá data jsou GPS souřadnice ve sloupcích *lat* a *lng* s typem *decimal(10,8)*. Dále sloupec *zoom* obsahující velikost přiblížení.

Záznamy nutné k vykreslení plánu budovy jsou uloženy v tabulce „*hladiny*“. Obsahují primární index *id(int3)* a sloupce *rozmer(int10)* a *pocet(int20)*. Tyto hodnoty jsou použity pro správné vykreslení hladiny mapy.



Obr. 3.5 Schéma základních tabulek databáze

Databáze jsou uvedeny jako hlavní zdroj dat, ale hlavní mapa svá data čerpá ze souborového zdroje *gen_data.xml*. Tento soubor je exportem popsané tabulky „*makers*“, která je součástí databáze. Datový soubor *gen_data.xml* je generován při přidání nového záznamu do databáze. Obsah souboru je smazán a vygenerován nově. Toto řešení bylo zvoleno s ohledem na zvýšení efektivity *scriptu*. K tomuto postupu vedla skutečnost pouze sporadické změny databáze obsahující data k vykreslení jednotlivých ikon na hlavní mapu. Dále je důvodem jednodušší kód *scriptu* při načítání dat přímo z databáze. V databázi jsou také uloženy informace o pracovnících. Tabulka „*workers*“ obsahuje

kromě primárního indexu *id*, také jméno, příjmení, tituly a sloupec *roomId*. Ten spojuje pracovníka s jeho umístěním na podlaží. Databáze dále obsahuje potřebné informace o budově a místnosti. Jejich obsah je zobrazen na *Obr. 3.5*.

PHP script převádějící databázi do xml výstupu

```
<?php
// pomocí hlavičky určíme mime typ text/xml
header('content-type: text/xml');

// vložíme třídu pro mysql a xml
include 'xml_generator_class.php';
include 'mysql_class.php';
// název tabulky
$tb_name = 'makers';
// vytvoříme instanci třídy tMysql a připojíme k db
$sql = new tMysql;
$sql->dbName = 'mapa';
$sql->dbHost = 'localhost';
$sql->dbUser = 'petsa';
$sql->dbPassword = 'petsa';
$sql->connect();
// vytvoříme instanci třídy c_xml_generator
$xml = new c_xml_generator;
// vytvoříme si nejvyšší element
$top = $xml->add_node(0, 'table', array('name' => $tb_name));
// vybereme data z požadované tabulky
$sql->query('export', 'SELECT * FROM '.$tb_name);
// pro každý prvek výsledku vytvoříme jeden element
// ---
while($row = $sql->fetch_assoc('export')) {
    $xml->add_node($top, 'row', $row);
};
// ---
$sql->free_result('export');
$data = $xml->create_xml();
$soubor = fopen("gen_data.xml", "a+");
fwrite($soubor, $data);
fclose($soubor);

?>
```

Tento soubor obsahuje dva vložené scripty. Prvním je *xml_generator_class.php*, který generuje samotnou strukturu *xml* souboru. Druhým je *mysql_class.php*, který

zajišťuje připojení k databázi a čtení dat. Tyto *scripty* jsou převzaty z článku magazínu www.interval.cz z článku „Tvorba XML výstupu v PHP“. [1]

Zobrazení hlavní mapy

```
<script type="text/javascript" src="http://www.google.com/jsapi?key=<?$googlemapcode?>"></script>
<script type="text/javascript">
  google.load("maps", "2.x");
  // Call this function when the page has been loaded
  var global_markers = new Array(); // pole značek
  var map;

  // Vytvoří značku
  function createMarker(point, content, cap) {
    // objekt obsahující vlastnosti značky
    var opt = new Object();
    opt.title = cap;
    var marker = new GMarker(point, opt);
    // obsluhu kliknutí na značku
    GEvent.addListener(marker, "click", function() {
      map.setCenter(point, 15); // vycentruje a zzoomuje
      marker.openInfoWindowHtml(content);
    });
    return marker;
  }

  // Inicializace mapy
  function initialize() {
    if (GBrowserIsCompatible()) {
      map = new GMap2(document.getElementById("imt"));
      map.addControl(new GLargeMapControl());
      map.addControl(new GMapTypeControl());
      map.addControl(new GOverviewMapControl());
      map.setCenter(new GLatLng(50.764911, 15.054016), 13);
      map.setUIToDefault();
    }
  }

  // Načtení XML souboru
  GDownloadUrl("gen_data.xml", function(data, responseCode) {
    var xml = GXml.parse(data);
    var s;
    var buildings = xml.documentElement.getElementsByTagName("row");

    for (var i = 0; i < buildings.length; i++) {

var point = new GLatLng(parseFloat(buildings [i].getAttribute("lat")),
```

```

parseFloat(buildings [i].getAttribute("lng")));

// html obsah informacního okna
s = "<h1>" + buildings [i].getAttribute("label") + "</h1><img src='"+ buildings [i].getAttribute("image")+"' alt='"+ buildings
[i].getAttribute("label")+"' width='190'><br><a href='"+ buildings [i].getAttribute("url")+"' target='_blank'>text odkazu </a>";

var marker = createMarker(point, s, buildings [i].getAttribute("label"));

// přidání značky do globálního pole
global _markers [global _markers.length] = marker;

// přidání značky na mapu
map.addOverlay(marker);
}
});
}
google.setOnLoadCallback(initialize);
</script>

```

Celý výše uvedený *script* je proveden v níže uvedeném těle stránky a *tagu div* s příznakem *imt*.

```

<body onload="initialize()" onunload="GUnload()">
  <div id="imt" style="width: 740px; height: 480px"></div>
</body>

```

Zobrazení plánu budovy

Druhá část práce zobrazuje jednotlivá podlaží budov dle požadavku uživatele. O vykreslení správného podlaží se stará *ajaxová* funkce. Ta vrací výstup funkce nebo celého souboru do určeného *divu*. *Ajaxová* funkce je uložena v souboru *ajax.js* a ten je pomocí funkce *include()*, vložený do potřebných souborů. [2]

```

<script type="text/javascript">
function ajax(stranka, kam) {
  var httpRequest;
  if(typeof window.ActiveXObject != 'undefined')
  {
    httpRequest = new ActiveXObject("Microsoft.XMLHTTP");

```

```

}
else
{
    httpRequest = new XMLHttpRequest();
}
httpRequest.open("GET", stranka, true);
httpRequest.onreadystatechange= function ()
{
    processRequest(httpRequest, kam)
};
httpRequest.send(null);
}

function processRequest(httpRequest, kam)
{
    if (httpRequest.readyState == 4)
    {
        if (( httpRequest.status >= 200 && httpRequest.status < 300 ) || httpRequest.status == 304)
        {
            if (typeof kam == 'string')
            {
                document.getElementById(kam).innerHTML = httpRequest.responseText;
            }
            if (typeof kam == 'function')
            {
                kam(httpRequest.responseText);
            }
        }
        else
        {
            alert("Chyba pri nacitani stanky " + httpRequest.status + " : " + httpRequest.statusText);
        }
    }
    else
    {
        if (typeof kam == 'string' && kam != "")
        {
            document.getElementById(kam).innerHTML = '';
        }
    }
}

function mojefunkce(parametr)
{
    alert(parametr);
}
</script>

```


O výpis jednoho podlaží budovy se stará funkce *vypishlad(\$budova,\$poschodi)*. Parametry jsou označení budovy a aktuálního čísla poschodí, které požadujeme vykreslit. Doplňkem k výpisu požadované hladiny mapy je výpis jednotlivých pracovníků, kteří na daném podlaží mají svoji kancelář. Tento výpis provádí funkce *workersOnFloor(\$budova,\$poschodi)*. Parametry funkce jsou totožné jako u předchozí popsané funkce. Obě funkce jsou umístěny v souboru *floor.php*. Tato separace obou funkcí je z důvodů volání tohoto souboru *ajaxovou* funkcí a poté vkládání výsledku obou funkcí do hlavního *divu content*.

```
function vypishlad($budova,$poschodi) {
    for ($i=1; $i < $rozmer; ++$i){
        for ($j=1; $j < $rozmer; ++$j){
            ?>
            <a href="#" onclick="ajax('index.php?akce=detail&ak=zoom&budova=B&poschodi=<?echo $poschodi;?>&radek=<?echo $i;?>&sloupec=<?echo $j;?>', 'content');" ></a>
            <?>><br><?>
        }
    }
}
```

Kód funkce pro výpis pracovníků na dané ploše [3]

```
function workersOnFloor($poschodi,$budova) {
    include ("odkazy.php");
    $spojeni = mysql_pconnect("$server", "$user", "$password");
    $db="$dba";

    $q = "SELECT bi.inc,
           w.name,
           w.surname,
           ai.ainc,
           r.room
    FROM `workers` w
    INNER JOIN `rooms` r
    ON w.roomid = r.id
    INNER JOIN `binc` bi
    ON bi.id = w.binc
    INNER JOIN `ainc` ai
    ON ai.id = w.ainc
    WHERE r.floor = '$poschodi' AND r.building = '$budova.' ";
    $result = MySQL_db_query($db, $q, $spojeni);

    echo("<table border='0'>");
    for ($i=0;$i<mysql_num_fields($result); $i++){
```

```

//echo("<td><strong>.mysql_field_name($result, $i).</strong></td>");
}
while (list($inc,$name, $surname, $ainc,$room) = mysql_fetch_row($result)){
echo("<tr><td>$inc</td><td>$name</td><td>$surname</td><td>$ainc</td> - <td>$room </td></tr>");
}
echo("</table>");
}

?><div id="menu"><?
echo ("<h4>Pracovníci na tomto podloží</h4>");
$d = workersOnFloor($_GET['budova'], $_GET['poschodi']);
?></div><?

```

Výpis obou funkcí

```

<table border="0" width="8%" id="table1">
<tr>
<td align="center"><button onclick="ajax('floor.php?akce=plus&ak=plus&poschodi=<?echo
$_GET['poschodi']+1;?>&budova=<?echo $_GET['budova']?>', 'content');"> + </button></td>
<td align="center"><button onclick="ajax('floor.php?akce=minus&ak=plus&poschodi=<?echo $_GET['poschodi']-
1;?>&budova=<?echo $_GET['budova']?>', 'content');"> - </button></td>
</tr>
</table>

```

Hlavní stránkou aplikace je *index.php*. Ta obsahuje hlavičku, potřebné *meta tagy* a základní strukturu stránky pomocí *tagu div*. Jejich vlastnosti jsou definovány v souboru *style.css*.

```

<body>
<div id="header">
<a href="index.php"><b>Interaktivní mapa TUL</b></a><br>
<form name="select_m" id="" action="index.php" method="get">
<select name="sel1" onchange="ajax('floor.php?akce=start&'+this.value, 'content');">
<?
    allWorker();
?>
</select>
</form>
</div>
<div id="page">
<div id="content">

<?
    if (isset($_GET['akce'])){
        //vypishlad($budova,$poschodi);
    }else{

```

```

        require ("gmap.php");
    }
    ?>
</div>
</div>
<div id="footer">
    <table>
    <tr>
        <td><a href="#" onclick="ajax('floor.php?akce=start&budova=B&poschodi=1','content');">Budova A</a><td>
        <td><a href="#" onclick="ajax('floor.php?akce=start&budova=B&poschodi=1','content');">Budova B</a><td>
    </tr>
    </table>
</div>
</body>

```

Problematika vyhledávání

Poslední funkcionalitou projektu je vyhledávání. Dle zadání a návrhu stránek má být řešeno pomocí *MySQL* dotazů. Nad rámec této práce je vývoj složitějšího fulltextového vyhledávání. Vstupním parametrem je jméno pracovníka nebo název místnosti. Výstupem by mělo být vypsání správného podlaží. Vyhledávání je řešeno dvěma boxy typu *select*. Jeden bude obsahovat jména pracovníků v databázi a druhý čísla místností. Po výběru pracovníka resp. místnosti z boxu předá funkce *onchange()* potřebné požadavky funkci *ajax()*. Její parametry jsou umístění *scriptu* zobrazující podlaží budovy a *id divu*, který má překreslit.

```

<form name="select_m" id="select_m" action="index.php" method="get">
    <select name="sel1" onchange="ajax('floor.php?akce=start&'+this.value, 'content');" style="font-size:12px;">
    <?
        allWorker(); // vypisuje všechny pracovníky v databázi v abecedním pořadí v divu <option>
    ?>
    </select>
</form>

```

Výstup bude dodržen dle návrhu a bude zobrazovat správné podlaží. Tímto bude docíleno minimalizování problémů s vyhledáváním, které by jednoduché *MySQL* vyhledávání představovalo. Zejména možnosti překlepů, interpunkce nebo shody jmen. Selektivní možnost vyhledávání bude představovat přesnější službu.

Administrace stránek

Součástí aplikace je jednoduchá administrace pro základní správu celé aplikace. Je vytvořena převážně tak, aby nebylo nutné zasahovat do databáze a nedocházelo k poškození dat.

Možnosti administrace:

- změna klíče hlavní mapy
- vložení, editace a odstranění významných značek na hlavní mapě a jejich obsahu
- vložení, editace a odstranění záznamů o pracovnících

Databáze je zabezpečena jednoduchým kódováním pomocí funkce MD5.

3.5 Testování

Samotné testování bylo rozděleno do několika fází. Testování samostatných komponent probíhalo po jejich dokončení. Následně byla otestována aplikace jako celek. Do testování aplikace byl zapojen programátor aplikace a následně dva studenti Technické univerzity, jeden programátor a projektový manager. Z tohoto testování vyšel požadavek na změnu systému vyhledávání. Po dohodě byl zvolen výběrový způsob. V konečné fázi proběhlo testování aplikace na různých prohlížečích pomocí služby na stránce <http://browsershots.org>.

4 ZÁVĚR

Výsledek práce odpovídá zejména dostupnosti potřebných dat. Problematika mapových podkladů výrazně omezila možnosti vlastní realizace a bylo nutné se spolehnout na mapové *API* společnosti *Google Inc.*

Druhým důležitým aspektem práce je samotné zpracování plánů budov. To se neukázalo ani tak problematické, jako spíše časově náročné. Není možné obsáhnout všechny pracovníky Technické univerzity, jelikož se často mění jak popis místností, tak i samotní pracovníci. V budoucnu bude záležet na provozovateli této aplikace, zda stanoví konkrétní osobu zodpovědnou za aktualizaci těchto dat.

Dále je do budoucna počítáno s připojením k databázi nové aplikace „Telefonní seznam TUL“. Aplikace neobsahuje funkce importu dat z již existující databáze kontaktů. Pro spojení s aplikací „Telefonní seznam TUL“ bude nutné upravit strukturu databázi s využitím primárního klíče ID místnosti. Do budoucna bude tedy možné snadno rozšířit aplikaci o další a kompletnější informace o pracovnících a učebnách.

Výsledná aplikace přispěje k přehlednosti navigace mezi budovami Technické univerzity a ulehčí hledání učeben a jednotlivých pracovníků univerzity.

Celá výsledná aplikace je umístěna na webové stránce <http://www.imt.webz.cz/> a kompletní zdrojový kód je nahraný na příloženém CD.

5 LITERATURA

- [1] <http://www.interval.cz/>
- [2] <http://citron.blueboard.cz/>
- [3] <http://www.root.cz/>