

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Virtuálna FIIT

Projektová dokumentácia

Vedúci projektu: Mgr. Alena Kovárová

Autori: Bc. Filip Hlaváček Bc. Michal Palček
Bc. Ján Hudec Bc. Rastislav Pečík
Bc. Pavol Mešťaník Bc. Ivan Polko
Bc. Matúš Novotný

Ak. rok: 2010/2011

Obsah

Obsah.....	ii
Zoznam obrázkov	iv
Zadanie.....	v
Úvod	vi
Prehľad dokumentu.....	vi
1 Riešenie minuloročného tímu – nesplnené požiadavky.....	1
2 Prehľad 3D technológií	2
2.1 Technológia O3D	2
2.2 Technológia WebGL.....	2
3 Knižnice pre JavaScript a PHP	5
3.1 JavaScript.....	5
3.2 PHP	5
4 Opis použitých nástrojov	6
4.1 Prehliadače - WebGL.....	6
4.1.1 Firefox 4.....	6
4.1.2 Chrome Canary.....	6
4.2 Modelovanie.....	7
5 Údaje v systéme	8
5.1 Prístup k údajom.....	8
5.2 Získavanie údajov z AIS.....	9
6 Analýza programového prihlásenia do AIS.....	11
7 Identifikácia funkcionálnych požiadaviek.....	14
7.1 Detekcia kolízií.....	14
7.2 Interaktivita – otváranie dverí.....	14
7.3 3D model	15
7.4 Potreba 2D a mobilného rozhrania	15
7.5 Navigácia	15
8 Špecifikácia riešenia	17
8.1 Špecifikácia požiadaviek	17
8.1.1 Funkcionálne požiadavky.....	17
8.1.2 Nefunkcionálne požiadavky.....	17

8.2	Charakteristika používateľov systému	18
8.3	Prípady použitia.....	18
8.3.1	Opis diagramu prípadov použitia	19
9	Architektúra systému	25
9.1	Server.....	25
9.2	Klient.....	26
10	Optimalizácia mobilnej verzie aplikácie	27
11	Návrh GUI aplikácie	28
11.1	Grafické rozhranie pre 3D verziu.....	28
11.2	Grafické rozhranie pre 2D a mobilnú verziu.....	29
12	Dátový model	30
12.1	Opis dátových entít	30
13	Priority implementácie	34

Zoznam obrázkov

Obr. 4.1 Plugin Firebug.....	6
Obr. 4.2 Nástroje pre vývojárov v prehliadači Chrome.....	7
Obr. 6.1 Artefakt, ktorý vznikol pri vykresľovaní modelu	15
Obr. 7.1 Diagram prípadov použitia	19
Obr. 8.1 Schéma architektúry systému	25
Obr. 10.1 Logický dátový model.....	32
Obr. 10.2 Fyzický dátový model	33

Zadanie

Pamätáte si, ako vždy začína každý semester? Zistíte si, aký je váš rozvrh, no v ňom sú záhadne zakódované čísla miestností, v ktorých máte cvičenia. A ako dlho vám trvá, kým nájdete miestnosť, v ktorej sa nachádza váš vedúci? A ako zistíte, kedy má váš prednášajúci konzultačné hodiny? Alebo ktorý cvičiaci má v danej miestnosti cvičenie hneď po vás?

Toto je len pár z mnohých problémov. Ich riešenie momentálne spočíva v tom, že si musíte otvoriť ten správny informačný zdroj a v ňom informáciu nájsť a aj tak vám nakoniec nikto nepovie, kde je miestnosť BX04. Nebolo by to krásne, keby ste jednoducho počítaču zadali číslo miestnosti alebo meno človeka a on vás k nemu virtuálne zaviedol? A čo tak keby to všetko fungovalo, až budeme mať novú budovu FIIT?

Vašou úlohou bude:

- zanalyzovať danú oblasť (napr. aj O3D vs. WebGL)
- vytvoriť skutočný 3D model novej budovy FIIT resp. opraviť existujúci
- navrhnuť a zimplementovať jeho interaktívne časti ako je napríklad funkčný výťah, informačné tabule, možnosť zadania otázky
- to všetko by samozrejme nešlo bez databázy a možnosti importovania do nej z iných existujúcich systémov
- riešenie bude potrebné optimalizovať tak, aby ho používateľ mohol používať aj cez jednoduché webové rozhranie
- na záver nesmie chýbať testovanie a vyhodnotenie použiteľnosti

Úvod

Predkladaný dokument obsahuje projektovú dokumentáciu vytvorenú v rámci predmetu Tímový projekt.

Cieľom tejto dokumentácie je vysvetliť riešenie celého projektu s názvom Virtuálna FIIT. Systém by mal slúžiť ako navigácia po novej budove fakulty a poskytovať informácie o jednotlivých miestnostiach.

Prehľad dokumentu

Dokument je rozdelený na niekoľko častí:

1. Úvod
 - Počiatočné informácie o dokumentácii
2. Analýza
 - Analýza použitých technológií, knižníc a vývojových nástrojov
 - Analýza súčasného stavu riešenia, jeho funkcionality a nedostatkov
3. Špecifikácia
 - Špecifikácia funkcionálnych a nefunkcionálnych požiadaviek systému
4. Návrh
 - Návrh architektúry systému
 - Návrh dátového modelu
 - Návrh GUI systému
 - Návrh 2D a mobilnej verzie systému

1 Riešenie minuloročného tímu – nesplnené požiadavky

Naše riešenie bude vychádzať z riešenia minuloročného tímu LOST¹. Podľa ich projektovej dokumentácie² uvádzame nasledujúce nesplnené, resp. nie úplne splnené, požiadavky:

- Nestihli namodelovať najspodnejšie poschodie, ktoré obsahuje prevažne skladové miestnosti.
- 3. a 4. poschodie nie je namodelované, ale iba skopírované 5.
- Nebol implementovaný výťah.
- Nezobrazuje sa aktuálna pozícia používateľa v 3D režime.
- V 3D režime je potrebné ošetriť prechádzania cez zatvorené dvere.
- Pri prechode medzi poschodiami sa niekedy vyskytne chyba.
- Databáza neobsahuje skoro žiadne údaje.

¹ <http://labss2.fiit.stuba.sk/TeamProject/2009/team03is-si/>

² http://labss2.fiit.stuba.sk/TeamProject/2009/team03is-si/docs/Projektova_dokumentacia_final.pdf

2 Prehľad 3D technológií

2.1 Technológia O3D

Minuloročný tím implementoval virtuálny model pomocou O3D plugin-u od spoločnosti Google, ktorý umožňuje zobrazovať 3D grafiku v internetovom prehliadači. S týmto riešením je spojených niekoľko problémov:

- O3D ako plugin je už ďalej nevyvíjaná technológia³. Namiesto toho Google prerobil technológiu O3D tak, aby fungovala ako knižnica nad WebGL.
- Nevyhnutnosť nainštalovať plugin do prehliadača. Používateľ nemá vždy možnosť nainštalovať plugin na počítači, za ktorým sa práve nachádza.

2.2 Technológia WebGL

Oproti minuloročnému tímu sme v inej situácii, pretože technológia WebGL je už použiteľná, čo dokazuje aj množstvo ukážok dostupných na internete. WebGL vychádza zo štandardu pre zobrazovanie 3D grafiky OpenGL ES 2.0⁴. V prehliadačoch, ktoré ho budú podporovať, nebude vyžadovať žiaden plugin, čo je veľkou výhodou. Zobrazená 3D scéna, je normálny HTML element a nad jeho plochu je možné umiestňovať ďalšie HTML elementy, slúžiace napr. na zobrazovanie ovládacích prvkov, alebo zobrazovanie informácií o označenej miestnosti.

Ešte nie je vydaný prehliadač s podporou WebGL, a musíme tak pracovať s vývojovými verziami Mozilla Firefox-u - verzia 4⁵ a Google Chrome – verzia Canary⁶. Firefox 4 by však mal výjsť na začiatku roka 2011, teda pri dokončení tímového projektu by už mal byť dostupný aspoň jeden prehliadač, v ktorom bude WebGL fungovať.

Samotné WebGL pracuje na pomerne nízkej úrovni. Aby sme urýchlili a uľahčili vývoj, rozhodli sme sa, že použijeme niektorú z nasledujúcich knižníc, ktoré sme preskúmali.

GLGE

GLGE je knižnica založená na JavaScript-e zjednodušujúca prácu s WebGL. Hlavným cieľom tvorby GLGE je rýchla práca s WebGL, čím sa vývojár môže sústrediť na bohatší obsah svojich stránok. Daná knižnica v súčasnosti ponúka širokú funkcionálnosť. Súčasná verzia knižnice je v. 0.5.2, čo nám hovorí o jej neúplnosti, no neustále sa rozširuje a nadobúda nové funkcie. Poslednou pridanou funkciou danej knižnice bol „level of details“, pričom autor danej knižnice na svojej stránke⁷ uvádza funkcie, ktoré plánuje v blízkej budúcnosti doplniť. Daná knižnica vie priamo pracovať s grafickým formátom Collada a podporuje animácie.

C3DL

³ <http://code.google.com/apis/o3d/>

⁴ <https://cvs.khronos.org/svn/repos/registry/trunk/public/webgl/doc/spec/WebGL-spec.html>

⁵ Ďalej len Firefox 4.

⁶ Ďalej len Chrome.

⁷ <http://www.glge.org/>

C3DL⁸ celým názvom *The Canvas 3D JS Library*, je ďalšou JavaScript knižnicou. Skoršie verzie využívali predchodcu WebGL (experimentálnu technológiu Canvas 3D, z ktorej sa neskôr vyvinulo WebGL), no od verzie 2.0 využíva technológiu WebGL. V súčasnosti sa web stránka danej knižnice prerába, a prístup k niektorým informáciám je neúplný. Ku knižnici je spravená jednoduchá webová aplikácia na prácu s 3D modelmi. Medzi klady danej knižnice patria súbory s matematikou na podporu 3D grafiky. C3DL je schopná importovať grafický formát Collada a samozrejmosťou sú animácie.

Copperlicht

Knižnica Copperlicht⁹ je založená na JavaScript-e. Ponúka veľké množstvo podporovaných grafických formátov. Má veľmi rýchle prekršovanie 3D modelov. K danej knižnici je vytvorený aj editor (CopperCube), ktorý je potrebný na načítanie podporovaných grafických formátov. Na rozdiel od samotnej knižnice je spolpatnený. Táto knižnica už obsahuje pomerne rozsiahlu verziu 1.2.3 (detekcia kolízií, fyzika v modeloch), ktorú pravidelne rozširujú a fixujú staršie chyby. Základnou výhodou je dostatok informácií v podobe dobrej dokumentácie.

SceneJS

Je ďalšou „mladou“ knižnicou založenou na JavaScript-e poskytujúcou flexibilné API založené na JSON (*JavaScript Object Notation*). 3D modely vie zmeniť na JSON objekty, s ktorými je schopná pracovať. Súčasná verzia danej knižnice je v.0.7.8 beta. Daná knižnica zatiaľ nie je funkčnosťou porovnateľná s predchádzajúcimi. Dokumentácia k danej knižnici je zatiaľ veľmi skromná¹⁰.

SpiderGL

JavaScript knižnica, sľubujúca veľmi rýchle prekršovanie¹¹. Súčasná dostupná verzia je len v.0.1.1.20100129. Táto verzia ponúka: matematické funkcie, nízku/vysokú úroveň lineárnej algebry funkcií a tried, geometrické štruktúry a triedy orientované na modelovanie miest, priame a nepriame prístupy k prekršovaniu vo WebGL, používateľské rozhranie, správu interakcií a asynchrónne načítavanie obsahu. Dokumentácia je pomerne rozsiahla a kompletná.

GwtGL

Technológia GwtGL umožňuje vyvíjať web stránky a web aplikácie s 3D obsahom pomocou programovacieho jazyka Java, namiesto JavaScript-u¹². Nadväzuje na technológiu GWT, ktorá sa používa na budovanie komplexných web aplikácií. Na domovskej stránke tejto knižnice je uvedené, že GWT kompilátor generuje optimalizovaný JavaScript kód, čím prispieva k vyššej rýchlosti. Posledná vydaná verzia je z 15. februára 2010, vzhľadom na to, že odvtedy sa WebGL špecifikácia zmenila, a preto demá na stránke nie sú funkčné vo vývojovej verzii Firefox-u 4, je táto technológia pre nás nepoužiteľná.

⁸ <http://www.c3dl.org/index.php/development-news/>

⁹ <http://www.ambiera.com/copperlicht/>

¹⁰ <http://scenejs.org/>

¹¹ <http://spidergl.org/>

¹² <http://code.google.com/p/gwtgl/>

O3D-WebGL

Prepísaním technológie O3D, tak aby fungovala nad WebGL vznikla knižnica O3D-WebGL. Niektoré vlastnosti technológie O3D však neboli prenesené do O3D-WebGL:

- Animácie
- Vytváranie vlastných textúr, čo je užitočné, napr. pri dynamickom vytváraní nápisov v modeli.
- Modely nie sú prenášané v skomprimovanom tvare, a teda sú väčšie.

Na stránke je uvedený návod¹³, podľa ktorého sme sa pokúšali prekonvertovať riešenie minuloročného tímu do O3D-WebGL, čo sa nám však nepodarilo.

Výsledok porovnania

Knižnica CopperLicht vyšla z porovnania najlepšie. Má stabilnú verziu, kvalitné návody, kvalitnú dokumentáciu a pri prípadných problémoch je možné na jej fóre získať rýchle odpovede.

Ku knižnici CopperLicht však budeme musieť implementovať konvertor do vlastného formátu, pretože priamo CopperLicht nepodporuje žiaden formát pre 3d modely (len cez platený editor). Z programu Autodesk 3ds max-u využijeme export do formátu Collada, pretože je to prehľadný XML súbor. Ten načítame a zobrazíme. Neskôr v rámci optimalizácie formát Collada upravíme tak, aby súbory s modelmi boli menšie a model sa tak načítal rýchlejšie.

Bolo by možné použiť aj priamo Collada formát, vlastný formát však umožní optimalizovať veľkosť súborov s modelmi.

¹³ <http://code.google.com/p/o3d/wiki/GettingStarted>

3 Knižnice pre JavaScript a PHP

3.1 JavaScript

Minuloročný tím využil knižnicu JQuery na napovedanie pri vyhľadávaní miestností. Je to najpopulárnejšia knižnica pre JavaScript¹⁴. Využijeme ju preto aj v našom projekte. Okrem napovedania JQuery použijeme aj na rôzne vizuálne efekty, ako sú vysúvacie panely alebo animované zobrazovanie informácií a pod. Tieto vizuálne efekty sa budú môcť odohrávať aj pred zobrazenou 3D scénou, čo je jedna z už spomenutých výhod WebGL.

3.2 PHP

Pri súčasne plánovanom rozsahu by PHP framework možno ani nebol potrebný, keďže jediný, čo bude naprogramované v PHP, je výber z databázy a konverzia vybraných informácií do vhodného formátu pre stránku. Ak však budeme uvažovať do budúcnosti, že by mohlo pridať administratívne rozhranie, vtedy by už bolo vhodné nejaký framework využiť.

Uvažovali sme nasledujúce technológie (vylúčili sme CMS, ktoré nehľadáme, knižnicu ezComponents, ktorá nepodporuje AJAX a knižnicu Fusebox, ktorá už nie je vyvíjaná):¹⁵

- CakePHP
- CodeIgniter
- Qcodo
- Symfony
- Yii
- Zend Framework

Vybrali sme si framework CodeIgniter¹⁶, pretože ako uvádzajú na stránke, je jednoduchý a rýchly, čo je na naše použitie ideálne, ďalej je dobre zdokumentovaný a jeden člen tímu už s ním má skúsenosti.

¹⁴ <http://trends.builtwith.com/javascript/JQuery>

¹⁵ http://en.wikipedia.org/wiki/Comparison_of_web_application_frameworks#PHP_2

¹⁶ <http://codeigniter.com/>

4 Opis použitých nástrojov

V tejto sekcii sú opísané nástroje, ktoré použijeme pri riešení projektu.

4.1 Prehliadače - WebGL

Prehliadače, ktoré podporujú WebGL, sú beta verzie Firefox-u 4, resp. jeho vývojové verzie s názvom Minefield, vývojové verzie Safari na Mac OS X a vývojové verzie Chrome - Chrome Canary¹⁷. Vzhľadom na to, že nikto z tímu nemá k dispozícii Mac, budeme 3D rozhranie skúšať na prehliadačoch Firefox 4 a Chrome Canary. Prednostne sa budeme orientovať na prehliadač Firefox 4, pretože očakávame, že sa bude najrýchlejšie rozširovať spomedzi prehliadačov podporujúcich WebGL.

4.1.1 Firefox 4

Vo Firefox-e 4 využijeme rovnako ako minuloročný tím plug-in Firebug¹⁸, ktorý nám umožní ladiť kód v JavaScript-e, ako aj prehľadne zobrazovať jednotlivé HTTP požiadavky tak, ako je znázornené na Obr. 4.1, čo určite využijeme pri optimalizácii.

URL	Status	Domain	Size	Timeline
Net panel activated. Any requests while the net panel is inactive are not shown.				
GET logging.g	200 OK	getfirebug.com	25.8 KB	390ms
GET screenLo	200 OK	getfirebug.com	7.2 KB	594ms
GET screenLo	200 OK	getfirebug.com	4.6 KB	375ms
GET screenLo	200 OK	getfirebug.com	4.6 KB	390ms
GET screenLo	200 OK	getfirebug.com	7.4 KB	578ms
GET screenLo	200 OK	getfirebug.com	5.8 KB	578ms
GET screenLo	200 OK	getfirebug.com	5.7 KB (7.6 KB)	1.16s
GET screenLo	200 OK	getfirebug.com	5.7 KB (11.7 KB)	1.14s
GET screenLo	200 OK	getfirebug.com	5.5 KB (8.8 KB)	1.14s
GET screenLo	200 OK	getfirebug.com	7.3 KB (10.2 KB)	1.14s
10 requests			79.5 KB	1.16s

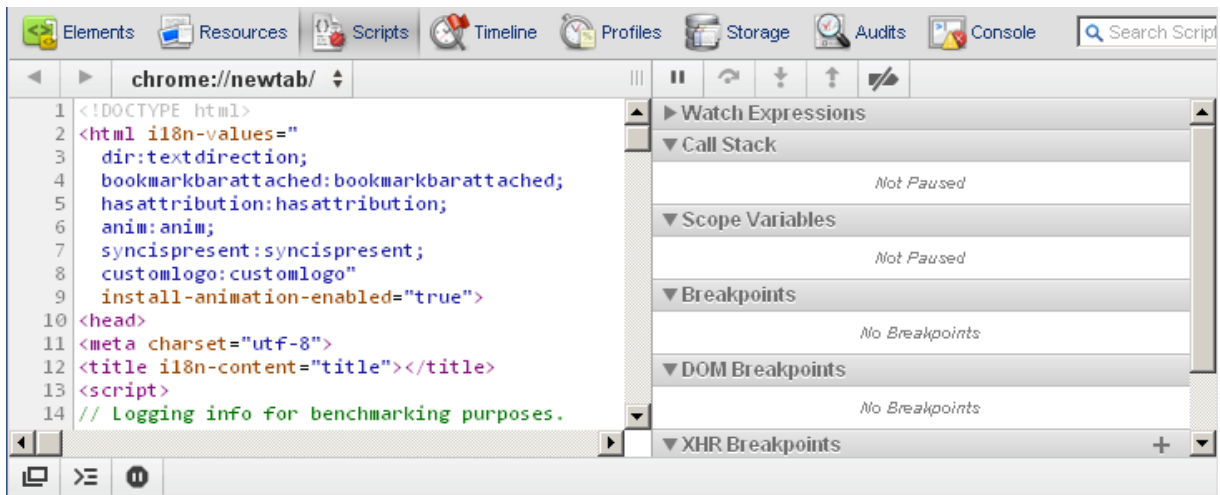
Obr. 4.1 Plugin Firebug

4.1.2 Chrome Canary

Prehliadač Chrome Canary je vývojová verzia prehliadača Chrome. Poskytuje zabudované nástroje pre vývojárov, s podobnou funkcionalitou ako Firebug. Ukážka týchto nástrojov sa nachádza na Obr. 4.2.

¹⁷ http://www.khronos.org/webgl/wiki/Getting_a_WebGL_Implementation

¹⁸ <http://getfirebug.com/>



Obr. 4.2 Nástroje pre vývojárov v prehliadači Chrome

4.2 Modelovanie

Minuloročný tím vytvoril svoj model v nástroji Autodesk 3ds max 2010, preto ho budeme používať aj my. V súčasnosti je najnovšia verzia 2011, ktorá sa dá stiahnuť a vyskúšať po dobu 30 dní. Pre študentov však ponúkajú možnosť zaregistrovať sa a stiahnuť si plnú verziu softvéru zadarmo.¹⁹

¹⁹ <http://students.autodesk.com/?nd=register&und=624>

5 Údaje v systéme

V rámci projektu bude potrebné ukladať a zobrazovať údaje v rôznej forme. Okrem samotného zobrazovania 3D, alebo 2D modelu novej budovy FIIT, prípadne aj načítavania iných súborov s údajmi chceme, aby Virtuálna FIIT slúžila aj ako vhodný informačný zdroj. V univerzitnom prostredí môžeme uvažovať o dvoch skupinách používateľov. Prvou skupinou sú študenti a pracovníci fakulty. Pre týchto sú relevantné najmä informácie o mapovaní rozvrhov na miestnosti a pracovniach jednotlivých zamestnancov fakulty. Druhou kategóriou sú návštevníci fakulty, pre ktorých sú dôležité informácie, ako názvy miestností, ich pozícia a opäť pozície pracovní zamestnancov fakulty.

Z tohto je možné vidieť, že v rámci nášho projektu budeme potrebovať získavať a zobrazovať nasledujúce informácie.

- názvy a umiestnenia miestností
- zoznam predmetov
- rozvrhové akcie v jednotlivých miestnostiach
- údaje o zamestnancoch (pracovňa, telefón, email, ...)

Prístup k týmto informáciám opisujeme v nasledujúcej časti.

5.1 Prístup k údajom

Všetky relevantné informácie potrebné pre náš projekt sa nachádzajú v databáze Akademického informačného systému Slovenskej technickej univerzity skratene AIS STU²⁰. Prístup k týmto informáciám je možný troma spôsobmi.

- priamy prístup do databázy AIS
- prístup do testovacej databázy AIS obsahujúcej vybrané pohľady na tabuľky
- parsovanie údajov z webovej stránky AIS

Najvhodnejší by bol prvý spôsob prístupu, teda priamy prístup do skutočnej databázy AIS, keďže by sme mohli pracovať rýchlo a efektívne s aktuálnymi údajmi. Nevýhodou a hlavným problémom je, že toto nie je z administratívneho dôvodu možné, keďže databáza obsahuje citlivé osobné údaje, ku ktorým nemôžeme mať prístup. Z tohto dôvodu by bolo nutné pre projekt virtuálnej FIIT vytvárať špeciálne pohľady, ktoré by obmedzovali prístup k informáciám, čo ale nie je možné. Teda prístup k skutočnej databáze AIS nie je pre tento projekt možný najmä z administratívnych dôvodov.

Druhou možnosťou je prístup k testovacej databáze AIS. V tejto testovacej databáze sa nachádzajú vybrané pohľady a kópie údajov zo skutočnej databázy a teda by mohla slúžiť ako vhodná náhrada za reálnu databázu. Nevýhodou je komplikovaný prístup, keďže táto testovacia databáza je v súčasnosti prístupná len z počítačov zamestnancov fakulty. Druhým problémom je nekompletnosť údajov. V tejto testovacej databáze sa nachádzajú len vybrané údaje a chýbajú v nej niektoré informácie, ktoré chceme v našom projekte zobrazovať. Tretím a jedným z najväznejších údajov je ale neaktuálnosť týchto údajov. V tejto testovacej databáze sa nachádzajú údaje staré jeden až dva roky. Tento prístup z praktických dôvodov nie je možný, keďže pre potreby projektu potrebujeme údaje, ktoré je možné aktualizovať minimálne v rozsahu mesiacov prípadne aj kratšom.

²⁰ <http://is.stuba.sk>

Poslednou možnosťou získania údajov je parsovanie údajov priamo zo stránky AIS. Tento spôsob získavania údajov síce nie je najrýchlejší, ani najvhodnejší, ale v súčasnosti sú v rámci AIS verejne prístupné všetky údaje potrebné pre náš projekt. Tieto údaje sú aktualizované okamžite pri každej zmene a sú prístupné z ľubovoľného miesta s pripojením k internetu. Preto sme sa rozhodli využiť tento spôsob prístupu. Podrobnejší opis prístupu k údajom sa nachádza v ďalšej časti.

5.2 Získavanie údajov z AIS

Základné informácie potrebné pre projekt sa dajú získať v podobe zdrojových kódov stránok z AIS STU. Ide konkrétne o stránku s informáciami o zamestnancoch fakulty²¹ a stránku s informáciami o rozvrhových akciách²², kde je nutné ešte zvoliť výber zobrazenia zoznamu rozvrhových akcií. Obe stránky obsahujú informácie vo forme tabuliek.

Prvá z týchto stránok obsahuje údaje o zamestnancoch fakulty a externých zamestnancoch fakulty, konkrétne ide o tieto:

- meno
- pracovisko
- kancelária
- telefón
- e-mail

Druhá tabuľka obsahuje údaje o rozvrhových akciách:

- deň
- začiatok rozvrhovej akcie
- koniec rozvrhovej akcie
- predmet
- typ akcie
- miestnosť
- vyučujúceho
- obmedzenia
- poznámku k rozvrhovej akcii

Tieto dve tabuľky poskytujú väčšinu podstatných údajov. Ďalšie konkrétnejšie údaje o zamestnancoch by bolo síce možné získať, ale nebolo by praktické ich všetky ukladať opätovne do našej databázy, keď už sú verejne prístupné v AIS. Príkladom takýchto informácií, ktoré sú často potrebné sú konzultačné hodiny. Tie sa nachádzajú v profiloch jednotlivých zamestnancov, ktoré si však spravujú zamestnanci jednotlivo a nie sú preto v rovnakej forme a neobsahujú všetky informácie. Preto takéto údaje zobrazované v profiloch zamestnancov bude efektívnejšie nezobrazovať priamo, ale len poskytnúť odkaz do AIS.

Samotný proces získavania údajov z AIS je možné realizovať prostredníctvom parsovania zdrojového kódu stránok AIS. Toto parsovanie bude vykonávať program, respektíve skript, ktorý bude v pravidelných intervaloch aktualizovať všetky údaje v databáze. Ako efektívne riešenie bolo uvažované použitie programu v jazyku Java a použitie PHP skriptu. Oba tieto prístupy by boli

²¹ <http://is.stuba.sk/pracoviste/zamestnanci.pl?id=70>

²² http://is.stuba.sk/katalog/rozvrhy_view.pl?konf=1;f=70

efektívne a funkčné. Ako vhodnejšie sa ale javí použitie PHP najmä z dôvodu, že PHP bude potrebné pre realizáciu iných častí projektu a preto nebude nutná inštalácia ďalších komponentov. Avšak použitie programu v jazyku Java by si vyžadovalo inštaláciu Java Virtual Machine, čo by bola ďalšia dodatočná požiadavka nášho systému. Preto je teda vhodnejšie použiť PHP, pomocou ktorého je možné získať stránku z AIS s potrebnými údajmi. Zo zdrojového kódu tejto stránky je ďalej možné vyparsovať údaje a tieto následne uložiť do vybraného databázového systému

6 Analýza programového prihlásenia do AIS

Poslucháči univerzity, ktorí majú pridelené prihlasovacie údaje, majú prístup do univerzitného informačného systému prostredníctvom internetu. Každý poslucháč, s ktorým sme sa stretli pristupoval do AIS-u prostredníctvom internetového prehliadača. Tak získava prístup nie len k všeobecne dostupným informáciám, ale aj ku svojim osobným údajom. To nás inšpirovalo a začali sme sa zaujímať o spôsoby, ako získať osobné údaje z AIS-u, bez prehliadača. Vyriešenie tohto problému by pomohlo pri riešení získavania dát z AIS-u automatizovane bez nutnosti manuálneho spracovania dát osobou.

Analýzu spojenia sme vykonali pomocou internetového prehliadača Mozilla Firefox s nainštalovaným doplnkom Firebug a Firecookie, ktoré umožňujú sledovať odosielanie požiadaviek a prijímanie odpovedí servera. Opakovane sme sa pokúšali prihlásiť do AIS-u a sledovali sme odpovede servera pri odoslaní správnych a nesprávnych prihlasovacích údajov. Tým sme zistili, aké HTTP hlavičky žiada a prijíma internetový prehliadač:

Vyžiadanie štartovacej stránky <http://is.stuba.sk/>

```
GET / HTTP/1.1
Host: is.stuba.sk
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; sk; rv:1.9.2.12)
Gecko/20101026 Firefox/3.6.12
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: sk,cs;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-2,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
```

Odpoveď servera na vyžiadanie štartovacej stránky:

```
HTTP/1.1 200 OK
Date: Fri, 29 Oct 2010 16:21:25 GMT
Server: Apache
Content-Language: sk
Expires: now
Pragma: no-cache
Cache-Control: no-cache
Content-Length: 12601
Connection: close
Content-Type: text/html; charset=iso-8859-2
```

Z vrátených hlavičiek sme zistili, že je používaný server Apache. Po zobrazení štartovacej stránky sme sa pokúsili prihlásiť so správnymi prihlasovacími údajmi, klikli sme na odkaz Prihlásenie do osobnej administratívy AIS²³, zobrazil sa prihlasovací formulár, zadali sme správne údaje a formulár odoslali.

Prihlásenie sa do AIS-u so správnymi údajmi:

```
GET /auth/ HTTP/1.1
Host: is.stuba.sk
```

²³ <http://is.stuba.sk/auth/>

```
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; sk; rv:1.9.2.12)
Gecko/20101026 Firefox/3.6.12
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: sk,cs;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-2,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://is.stuba.sk/
```

Odpoveď servera na prihlásenie:

```
HTTP/1.1 301 Moved Permanently
Date: Fri, 29 Oct 2010 16:25:27 GMT
Server: Apache
Location: https://is.stuba.sk/auth/
Content-Length: 330
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

Z vrátených hlavičiek sme zistili, že prihlásenie a prenos informácií po prihlásení sa realizuje cez HTTPS protokol. Prehliadač odpoveď 301 spracováva automaticky a následne prebehlo presmerovanie.

Odpoveď servera 301 spôsobila presmerovanie:

```
GET /auth/ HTTP/1.1
Host: is.stuba.sk
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; sk; rv:1.9.2.12)
Gecko/20101026 Firefox/3.6.12
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: sk,cs;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-2,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://is.stuba.sk/
Authorization: Basic *****
```

Prihlasovacie údaje a zneužiteľné informácie sú z bezpečnostných dôvodov v celej kapitole nahradené hviezdičkami.

Odpoveď servera na presmerovanie:

```
HTTP/1.1 200 OK
Date: Fri, 29 Oct 2010 16:25:33 GMT
Server: Apache
Content-Language: sk
Expires: now
Pragma: no-cache
Cache-Control: no-cache
Content-Length: 19554
Connection: close
Content-Type: text/html; charset=iso-8859-2
```

Prijatie hlavičiek po prihlásení bolo pre nás kľúčové. Zistili sme, že prehliadač po presmerovaní odoslal údaj Authorization a server po prihlásení neodoslal žiadne cookies. Následne sme aj ďalej po prihlásení sledovali komunikáciu a dospeli sme k záveru, ktorý sme odhalili už pri prihlásení – server

nepoužíva cookies a pri každom vyžiadaní obsahu po prihlásení sa odosiela údaj Authorization. Vyskúšali sme sa prihlásiť aj s nesprávnymi prihlasovacími údajmi a komunikácia prebiehala rovnako, prehliadač odosiela rovnaké hlavičky ako pri prihlásení so správnymi údajmi, ale údaj Authorization mal inú hodnotu. Server po neúspešnom prihlásení odoslal odpoveď 401.

Odpoveď servera na prihlásenie s nesprávnymi prihlasovacími údajmi:

```
HTTP/1.1 401 Authorization Required
Date: Fri, 29 Oct 2010 16:40:53 GMT
Server: Apache
WWW-Authenticate: Basic realm="AIS STU"
Content-Language: sk
Expires: now
Pragma: no-cache
Cache-Control: no-cache
Content-Length: 4926
Connection: close
Content-Type: text/html; charset=iso-8859-2
```

Z uvedenej analýzy vyplýva, že je možné vykonať programové prihlásenie sa do systému AIS, čo nám umožní získať údaje, ktoré sú dostupné iba po prihlásení, ako je napr. osobný rozvrh.

7 Identifikácia funkcionálnych požiadaviek

7.1 Detekcia kolízií

V analyzovanom riešení sú na detekciu kolízií použité výškové mapy. Princíp je taký, že na obrázku sú zakreslené červenou farbou miesta, kam sa používateľ pri prechádzaní modelu dostať nemôže a farbami od čiernej po bielu sú znázornené miesta, po ktorých sa pohybovať môže. Rôzne odtiene farby znázorňujú rôznu výšku podlahy. Dvere v mape vôbec nie sú zaznačené. Modrou farbou je znázornené miesto prechodu na vyššie poschodie.

Takýchto obrázkov je spolu 8. Problémom je ich veľkosť. Sú vo formáte PNG a spolu majú veľkosť 1,89 MB. Ich rozmery sú okolo 5000*1000 pixelov, čo pri načítaní do pamäte, ktoré je nevyhnutné na zisťovanie hodnôt jednotlivých pixelov, zaberie okolo 220MB (testované na prehliadači Chrome, zobrazená stránka, ktorá obsahuje iba tieto obrázky). Vhodné by bolo vyskúšať zmenšiť rozmer kolíznych máp, aby sme optimalizovali pamäťovú náročnosť, ale tak, aby to používateľ nepostrehol pri pohybe po modeli, napr. zaseknutím sa o stenu.

Možnosti na zobrazovanie miestnosti, v ktorej sa používateľ nachádza v 3D režime, sú nasledovné:

- Zafarbiť každú miestnosť inou farbou, a v tabuľke si udržiavať záznamy, ktorá miestnosť patrí ku ktorej farbe. Vytváranie farebnej mapy pre všetky poschodia je však časovo zdĺhavý proces. Nevýhodou je tiež pamäťová náročnosť. Výhodou je jeho rýchlosť.
- Využiť funkciu „picking“, ktorú poskytuje aj O3D. Vieme takto zistiť, aký objekt sa nachádza na danom bode obrazovky. Vo všeobecnosti vráti prvý objekt, ktorého polygóny sa pretínajú s priamkou určenou bodom obrazovky. Ak by sme namiesto toho použili priamku, ktorá smeruje do podlahy, vedeli by sme určiť v ktorej miestnosti sa používateľ nachádza. Hľadanie priesečníka s polygónmi je však výpočtovo náročné, a muselo by prebiehať pri každom vykreslení, aby informácia o miestnosti bola aktuálna. V súčasnom riešení sa táto funkcia využíva na označovanie miestností v prehliadacom režime.

Z hľadiska rýchlosti navrhujeme teda využiť už existujúcu výškovú mapu, zafarbiť miestnosti a vytvoriť mapovaciu tabuľku. Aby sme vedeli zaznačiť rôznu výšku podlahy v rôzne zafarbených miestnostiach, za predpokladu súboru s formátom PNG s farbou určenou 24 bitmi, využili by sme 16 bitov na identifikáciu miestnosti a 8 bitov na identifikáciu výšky podlahy.

7.2 Interaktivita – otváranie dverí

Otváranie dverí je v súčasnosti riešené pomocou funkcie „picking“. Je to výpočtovo náročné, preto je možnosť riešiť otváranie dverí pomocou klávesy, ako je to v mnohých počítačových hrách. Pri stlačení klávesy by sa zistili najbližšie dvere k používateľovi, ak by boli dostatočne blízko, tak by sa otvorili, resp. zatvorili. V blízkosti dverí by sa tiež zobrazila informácia pre používateľa, ktorým klávesom môže dvere otvoriť.

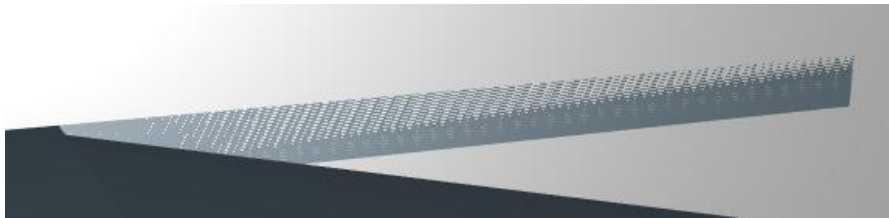
Aby nešlo prechádzať cez zatvorené dvere, bolo by vhodné zakomponovať ich do výškovej mapy, aby sme vedeli zistiť, že ideme prejsť cez dvere. Na zjednodušenie vytvárania výškovej mapy navrhujeme všetky dvere označiť jednou farbou. Ak pri prechádzaní modelom narazí používateľ na danú farbu, v zozname dverí budú nájdené dvere na danom poschodí, ktoré sú k nemu najbližšie. Podľa toho, či sú otvorené, bude dovolené používateľovi cez ne prejsť.

7.3 3D model

Model minuloročného tímu budeme musieť aktualizovať, tak aby zodpovedal najnovším dostupným plánom. Model je namodelovaný v programe 3ds max 2010 a je rozdelený do súborov podľa poschodí. Model poschodia sa skladá z jednotlivých miestností, dverí, skiel a ďalších komponentov.

Jednotlivé miestnosti musia byť v modeli vyčlenené ako samostatné pomenované objekty kvôli tomu, aby sa dala použiť funkcia picking, ktorá vráti objekt – miestnosť na ktorú používateľ klikol. Rozdelenie modelu do samostatných objektov však spomaľuje vykresľovanie. Súčasný riešenie ide plynulo, pri predbežných testoch s technológiou WebGL sme však zistili, že pri takomto rozdelení, je zobrazovanie modelu pomalšie. Pri spojení všetkých objektov do jedného, sa však model zobrazoval úplne plynulo. Pri spojení všetkých objektov, však strácame možnosť zvyrazňovať jednotlivé miestnosti. Vyriešiť by sa to dalo napr. zobrazovaním farebného priehľadného polygónu ako stropu miestnosti, čo by bol na pohľad rovnaký efekt ako zafarbenie celej miestnosti rovnakou farbou.

V modeli sú niektoré časti nesprávne namodelované, niektoré polygóny sa prekrývajú a vznikajú potom artefakty, ako je to ukázané na Obr. 7.1.



Obr. 7.1 Artefakt, ktorý vznikol pri vykresľovaní modelu

7.4 Potreba 2D a mobilného rozhrania

V súčasnosti oficiálne podporujú technológiu WebGL len vývojové verzie troch prehliadačov. Naším cieľom je však umožniť čo najväčšej skupine ľudí využiť našu aplikáciu a preto sme sa rozhodli rozšíriť pôvodne len 3D verziu Virtuálnej FIIT aj o 2D verziu, ktorá by fungovala aj na prehliadačoch nepodporujúcich WebGL. Ďalej sme predpokladali, že každý novo nastupujúci poslucháč vlastní mobilný telefón a to nás viedlo k myšlienke rozšíriť 2D verziu aj pre mobilné zariadenia. Hlavnou myšlienkou je jednoduchá a nenáročná navigácia po novej budove FIIT prostredníctvom mobilného telefónu.

7.5 Navigácia

Súčasťou analyzovaného riešenia je aj navigácia. Služi na nájdenie najkratšej cesty medzi dvoma miestnosťami. Uskutočňuje sa to pomocou grafu, ktorého uzly reprezentujú miestnosti. Tie sú poprepájané ohodnotenými cestami, ktoré reprezentujú vzdialenosti medzi jednotlivými miestnosťami. Ďalej obsahuje pomocné uzly nachádzajúce sa na možných križovatkách v budove. Graf je uložený v XML súbore.

V dokumentácii minuloročného riešenia je uvedené, že váhy hrán medzi susednými miestnosťami, medzi ktorými existuje okrem štandardných priechodov na chodbu navyše aj ďalší priechod, boli zvýšené tak, aby nedochádzalo k nekorektnému skracovaniu cesty cez susednú miestnosť. Po otestovaní vyhľadania takejto cesty sme však zistili, že nie všetky váhy takýchto hrán, kde mohlo dochádzať k nájdeniu nesprávnej cesty, boli správne prehodnotené. Preto bude potrebné upraviť niektoré váhy ciest v XML súbore.

Hľadanie najkratšej cesty je realizované pomocou Dijkstrovho algoritmu. Jedná sa o štandardný známy algoritmus, ktorý je pre naše potreby dostatočne efektívny.

8 Špecifikácia riešenia

8.1 Špecifikácia požiadaviek

Naše riešenie vychádza z riešenia minuloročného tímu, uvádzame preto iba požiadavky, ktoré sú v našom riešení nové, resp. tie, ktoré neboli v riešení minuloročného tímu implementované.

8.1.1 Funkcionálne požiadavky

- Prerobenie zobrazovania modelu novej budovy FIIT do technológie WebGL
- Vylepšenie interaktivity
 - o Funkčný výťah
 - o Nemožnosť prechádzania cez zatvorené dvere
 - o Zobrazovanie názvu miestnosti, v ktorej sa používateľ nachádza
- Aktualizovanie modelu podľa najnovších plánov budovy
- Oprava nedostatkov navigácie medzi miestnosťami
- Vytvorenie 2D verzie modelu, ktorá bude funkčná na bežne používaných prehliadačoch
- Vytvorenie verzie pre mobilné zariadenia
- Získavanie a zobrazovanie verejných informácií z AIS
- Získanie osobného rozvrhu z AIS po prihlásení
- Pre 2D a mobilnú verziu sú spoločné nasledovné funkcionálne požiadavky:
 - o Navigácia
 - s manuálnym zadaním cieľovej miestnosti,
 - s automatickým zistením miestnosti podľa rozvrhu (vyžaduje sa prihlásenie).
 - o Zobrazenie informácií o miestnosti.
 - o Zobrazenie informácií o vyučujúcom.
 - o Zobrazenie plánu budovy po jednotlivých poschodiach.

8.1.2 Nefunkcionálne požiadavky

- Systém musí podporovať desiatky súčasne pripojených používateľov
- 3D verzia systému musí byť použiteľná do 20 sekúnd pri priemernej rýchlosti spojenia²⁴, ktorá je na Slovensku 620kb/s (nemusia byť však načítané všetky poschodia, stačí iba aktuálne)
- 2D verzia systému musí byť použiteľná do 5 sekúnd pri priemernej rýchlosti spojenia.
- Mobilná verzia systému musí byť použiteľná do 10 sekúnd pri zvolení minimálnej úrovne detailov (v závislosti od vyťaženia a rýchlosti siete mobilného operátora).
- Všetky rozhrania by mali mať intuitívne ovládanie
- Aktualizácia uložených údajov v databáze by mala prebiehať každý deň v noci
- Prihlasovacie údaje užívateľa do AIS nesmú byť ukladané a ani logované.
- Prihlasovacie údaje musia byť zadávané cez zašifrované spojenie - HTTPS.
- Aktualizovanie údajov nesmie narušiť prácu používateľov so systémom
- Hardvérové a softvérové požiadavky na strane klienta:
 - o 3D rozhranie
 - Prehliadač s podporou WebGL
 - Samostatná grafická karta (pri integrovanej grafickej karte sa môžu objaviť problémy s plynulosťou prehliadania modelu)

²⁴ <http://www.dsl.sk/>

- 2D rozhranie
 - Štandardný prehliadač s podporou HTML, CSS a JavaScript.
- Mobilné rozhranie
 - Mobilné zariadenie s webovým prehliadačom a prístupom na internet.
 - Podpora JavaScript-u zariadením pre funkčnosť pokročilých interaktívnych možností.
- Hardvérové a softvérové požiadavky na strane serveru:
 - Webový server s podporou jazyka PHP (vyvíjané pod apache 2.2.9 s podporou PHP5)
 - Databázový systém MySQL 5.0
 - Operačný systém – každý, ktorý spĺňa predošlé požiadavky (vyvíjané pod OS GNU/Linux Debian Lenny s jadrom 2.6.26-2-686)

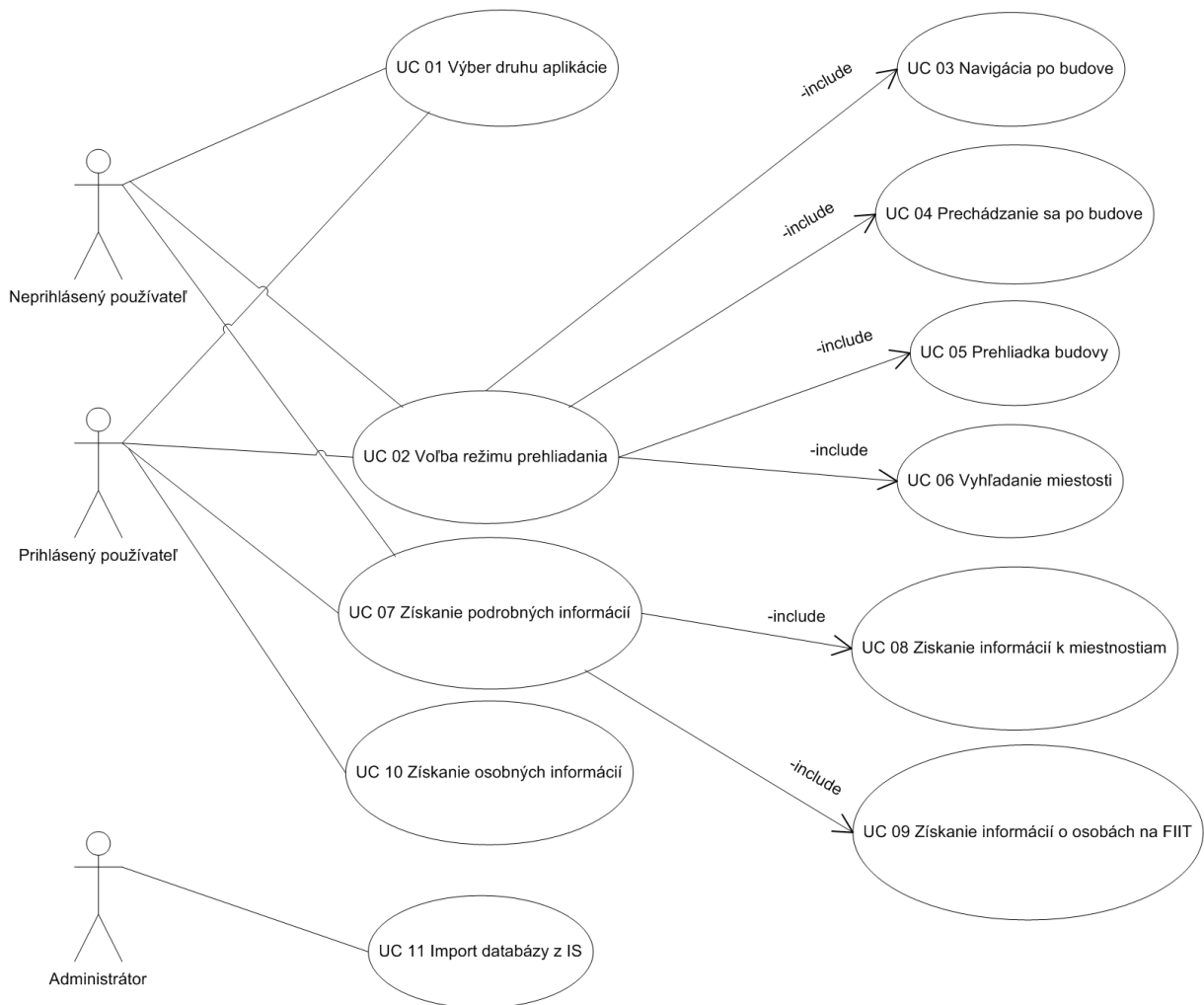
8.2 Charakteristika používateľov systému

Uvažujeme s tromi rolami používateľov:

- Administrátor – spravuje celý systém, vrátane modelu a údajov uložených v databáze.
- Používateľ – návštevník stránky, ktorý si chce prezrieť model budovy, alebo nájsť nejaké informácie.
- Prihlásený používateľ – návštevník stránky, ktorý zadal platné prihlasovacie údaje.

8.3 Prípady použitia

Na Obr. 8.1 je znázornený diagram prípadov použitia, ktorý znázorňuje funkčnosť aplikácie. Diagram bol vytvorený pomocou MS Visio.



Obr. 8.1 Diagram prípadov použitia

8.3.1 Opis diagramu prípadov použitia

8.3.1.1 UC1 Výber druhu aplikácie

Názov prípadu použitia:	Výber druhu aplikácie
Identifikácia prípadu použitia:	UC 01
Vstupná podmienka:	Korektné spustenie aplikácie v jednom z podporovaných zariadení
Kroky prípadu použitia:	<ol style="list-style-type: none"> 1. Používateľ spustí aplikáciu na podporovanom zariadení. 2. Aplikácia v prípade podpory viacerých módov zobrazenia (textová vetva aplikácie / 2D vetva / 3D vetva vo WebGL) sprístupní možnosť voľby módu aplikácie. Pri podpore len jednej z možností, ju aplikácia volí automaticky. 3. Používateľ si zvolí mód zobrazenia na základe možností použitého zariadenia. 4. Aplikácia načíta potrebné dáta pre daný mód.

Výstupná podmienka:	Aplikácia je spustená a zobrazená v móde podľa používateľovho výberu. V prípade zariadenia, ktoré podporovalo len jednu z módov zobrazenia, musí byť aplikácia korektne spustená v tomto móde.
----------------------------	--

8.3.1.2 UC2 Výber režimu prehliadania

Názov prípadu použitia:	Výber režimu prehliadania
Identifikácia prípadu použitia:	UC 02
Vstupná podmienka:	Aplikácia musí splniť UC 01.
Kroky prípadu použitia:	<ol style="list-style-type: none"> 1. Používateľ na základe voľby v UC 01 má k dispozícii niektoré, prípadne všetky, z režimov vizualizácie danej budovy (2D / 3D pohľad z vrchu / 3D pohľad s možnosťou lietania v modeli / 3D simulácia chôdze s interaktívnymi prvkami v budove FIIT). Používateľ klikne na ikonu predstavujúcu preferovaný režim náhľadu na scénu. 2. Aplikácia nastaví potrebné parametre prehliadania na hodnoty zodpovedajúce výberu.
Výstupná podmienka:	Režim podľa výberu je pripravený na prezeranie.

8.3.1.3 UC 03 Navigácia po budove

Názov prípadu použitia:	Navigácia po budove
Identifikácia prípadu použitia:	UC 03
Vstupná podmienka:	Aplikácia musí splniť UC 01.
Kroky prípadu použitia:	<ol style="list-style-type: none"> 1. Používateľ si zvolí možnosť navigácie v budove. 2. Systém otvorí možnosť výberu miestností (kliknutím / vpísaním názvu). 3. Používateľ si vyberie zo zoznamu miestností svoj počiatočný a cieľový bod. Možnosti je možné voliť aj kliknutím na danú miestnosť. Prípadne je možné cieľ a počiatočnú pozíciu načítať z iných prvkov v modeli. 4. Systém zobrazí priechodnú cestu medzi dvoma miestnosťami.
Výstupná podmienka:	Je zobrazená navigácia medzi dvoma miestnosťami, ak daná cesta existuje. Výsledné zobrazenie navigácie je závislé od módu zvoleného v UC 01.

8.3.1.4 UC 04 Prechádzanie sa po budove

Názov prípadu použitia:	Prechádzanie sa po budove
Identifikácia prípadu použitia:	UC 04
Vstupná podmienka:	Aplikácia musí splniť UC 01 a súčasne zariadenie, na ktorom je spustená, musí podporovať internetové prehliadače s podporou WebGL.
Kroky prípadu použitia:	<ol style="list-style-type: none"> 1. Používateľ si zvolí možnosť prechádzania sa po budove. 2. Systém umiestni kameru na počiatočné miesto v modeli. V prípade prechodu z iného náhľadu systém vráti pozíciu v modeli, ktorá je najbližšie k predchádzajúcemu náhľadu. 3. Používateľ má možnosť virtuálne sa prechádzať po budove FIIT. Otvárať dvere. Sledovať nástenky. Vozíť sa vo výťahu. 4. Systém nesmie dovoliť prechod cez steny ani lietanie medzi poschodiami, ale musí zabezpečiť adekvátne správanie prostredia voči pohľadu používateľa (<i>first person camera</i>).
Výstupná podmienka:	Používateľ sa môže virtuálne prechádzať po budove FIIT s interaktívnymi prvkami.

8.3.1.5 UC 05 Prehliadka budovy

Názov prípadu použitia:	Prehliadka budovy
Identifikácia prípadu použitia:	UC 05
Vstupná podmienka:	Aplikácia musí splniť UC 01.
Kroky prípadu použitia:	<ol style="list-style-type: none"> 1. Používateľ si zvolí možnosť prehliadky budovy. 2. Systém ponúka používateľovi možnosť voľby prehliadky budovy bez akýchkoľvek obmedzení (možnosť prechodu cez steny, poschodia, okná a dvere + pohyb v priestore bez ohľadu na zemskú príťažlivosť). 3. Používateľ po zvolení danej možnosti stráca spomínané obmedzenia a má možnosť prehliadky budovy.
Výstupná podmienka:	Používateľ má možnosť si virtuálne prezerať model bez obmedzení.

8.3.1.6 UC 06 Vyhľadanie miestnosti

Názov prípadu použitia:	Vyhľadanie miestnosti
Identifikácia prípadu použitia:	UC 06
Vstupná podmienka:	1. Aplikácia musí splniť UC 01.
Kroky prípadu použitia:	2. Používateľ si zvolí možnosť vyhľadania miestnosti.

	<ol style="list-style-type: none"> 3. Systém dovoľuje vyhľadanie miestnosti podľa mena, kliknutím, prípadne prostredníctvom možnosti zobrazenia miestnosti, v ktorej sa daná osoba v súčasnosti nachádza. 4. Používateľovi sa zobrazí daná miestnosť v modeli. Pokiaľ bola získaná pred vykonaním daného kroku súčasná pozícia používateľa je zobrazené aj cesta medzi danými dvoma bodmi.
Výstupná podmienka:	Je zobrazená požadovaná miestnosť.

8.3.1.7 UC 07 Získanie podrobných informácií

Názov prípadu použitia:	Získanie podrobných informácií
Identifikácia prípadu použitia:	UC 07
Vstupná podmienka:	Aplikácia musí splniť UC 03 a súčasne je spustená na zariadení s prehliadačom podporujúcim technológiu WebGL.
Kroky prípadu použitia:	<ol style="list-style-type: none"> 1. Používateľ klikne na niektorý z bodov ponúkajúcich informácie v modeli (napr. nástenka, výveska pred miestnosťou, ...). 2. Systém vyhľadá informácie vzťahujúce sa k danému objektu z databázy (export z informačného systému univerzity). 3. Používateľovi sa zobrazia načítané informácie v prehľadnej forme.
Výstupná podmienka:	Používateľovi sú zobrazené požadované informácie.

8.3.1.8 UC 08 Získanie informácií o miestnostiach

Názov prípadu použitia:	Získanie informácií o miestnostiach
Identifikácia prípadu použitia:	UC 08
Vstupná podmienka:	Aplikácia musí splniť UC 03 a súčasne je nutné aby bola spustená na zariadení s prehliadačom podporujúcim technológiu WebGL.
Kroky prípadu použitia:	<ol style="list-style-type: none"> 1. Používateľ klikne na interaktívnu nástenku v blízkosti vybranej miestnosti. 2. Systém vyhľadá informácie o danej miestnosti (napr. rozvrh danej miestnosti, osoba vedúca výučbu v čase prehliadky, ...). 3. Používateľovi sa zobrazia načítané informácie v prehľadnej forme.
Výstupná podmienka:	Používateľ získal bližšie informácie pomocou interaktívnych

	násteniek vo virtuálnej budove.
--	---------------------------------

8.3.1.9 UC 09 Získanie informácií o osobách na FIIT

Názov prípadu použitia:	Získanie informácií o osobách na FIIT
Identifikácia prípadu použitia:	UC 09
Vstupná podmienka:	Aplikácia musí splniť UC 03 a súčasne je nutné, aby bola spustená na zariadení s prehliadačom podporujúcim technológiu WebGL.
Kroky prípadu použitia:	<ol style="list-style-type: none"> 1. Používateľ klikne na interaktívnu nástenku v blízkosti zvolenej miestnosti, prípadne vojde do miestnosti, kde momentálne prebieha výučba, alebo do kancelárie. 2. Systém vyhledá informácie o osobách nachádzajúcich sa v danej miestnosti (napr. telefónne číslo na osobu, ktorá výučbu viedla, zoznam osôb v danej miestnosti, prípadne nájde miestnosť, kde by sa daná osoba mala nachádzať...). 3. Používateľovi sa zobrazia načítané informácie v prehľadnej forme.
Výstupná podmienka:	Používateľ získa bližšie informácií o osobách vo virtuálnej budove.

8.3.1.10 UC 10 Získanie osobných informácií

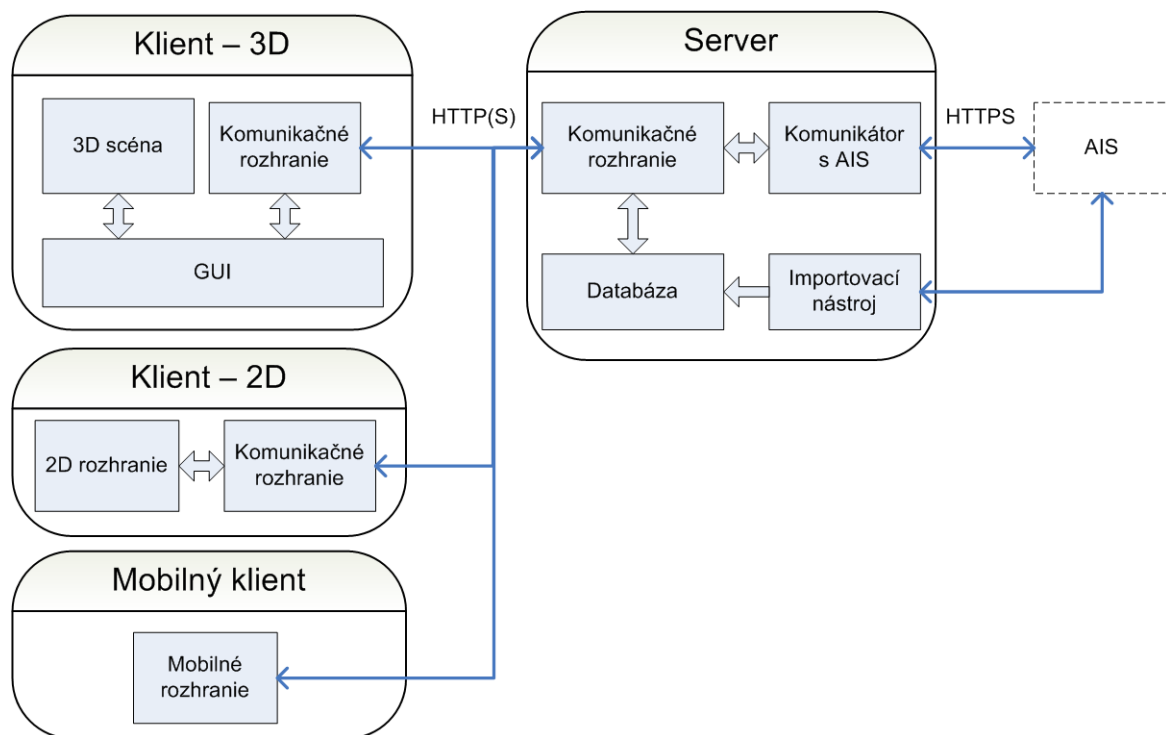
Názov prípadu použitia:	Získanie osobných informácií
Identifikácia prípadu použitia:	UC 10
Vstupná podmienka:	Aplikácia musí splniť UC 01.
Kroky prípadu použitia:	<ol style="list-style-type: none"> 1. Systém si vyžiada od používateľa prihlasovacie údaje do AIS-u. 2. Systém vyhledá informácie o štúdiu, miestnostiach a kolegoch, ktoré následne ponúkne používateľovi. 3. Používateľ konkretizuje svoje vyhľadávanie v systéme (hľadá kolegov, miestnosti v osobnom rozvrhu, jedáleň, atď.). 4. Systém prehľadne sprostredkuje používateľovi hľadané informácie.
Výstupná podmienka:	Používateľ získa požadované informácie.

8.3.1.11 UC 11 Import databázy z IS

Názov prípadu použitia:	Import databázy z IS
Identifikácia prípadu	UC 11

použitia:	
Vstupná podmienka:	-
Kroky prípadu použitia:	<ol style="list-style-type: none">1. Administrátor dá podnet na aktualizovanie údajov v databáze.2. Aplikácia nahradí údaje v databáze exportovanými dátami z akademického informačného systému.3. Aplikácia zobrazí kontrolné hlásenie o stave daného importu.
Výstupná podmienka:	Údaje v systéme budú zodpovedať aktuálnym údajom v databáze.

9 Architektúra systému



Obr. 9.1 Schéma architektúry systému

Naše riešenie vychádza z klient-server architektúry, ktorá je znázornená na Obr. 9.1. Klientom je prehliadač, v ktorom je zobrazené niektoré z rozhraní – 3D, 2D, alebo mobilné. Nasleduje opis základných komponentov.

9.1 Server

Implementačné technológie na strane servera sú PHP a MySQL.

Databáza

Uchováva všetky informácie, ktoré potrebujeme zobrazovať v modeli.

Importovací nástroj

Umožňuje načítavať údaje do databázy z AIS. Naprogramovaný bude v jazyku PHP.

Komunikačné rozhranie

Poskytuje rozhranie pre klientov na zisťovanie informácií. Komunikačné rozhranie dostane HTTP GET požiadavku, napr. na informáciu o určitej miestnosti. Požadované informácie vyberie z databázy a vráti klientovi vo formáte JSON, alebo v prípade mobilnej verzie vráti celú stránku s požadovanými informáciami. Naprogramované bude v jazyku PHP s framework-om CodeIgniter.

Komunikátor s AIS

Umožňuje programové prihlásenie do AIS a následné získanie osobného rozvrhu študenta.

9.2 Klient

Implementačné technológie na strane klienta sú JavaScript, HTML, CSS s využitím knižnice JQuery.

Komunikačné rozhranie

Poskytuje funkcie v JavaScript-e na zistenie informácií zo serveru. Technológiou AJAX zašle požiadavku v správnom tvare na server. Keď server odpovie, komunikačné rozhranie v klientovi oznámi, že prišla odpoveď. Tento proces prebieha asynchrónne, nečaká sa na odpoveď klienta. Komunikačné rozhranie je rovnaké pre 3D a 2D klienta.

GUI pre 3D rozhranie

Predstavuje používateľské rozhranie pre klienta, ktorý využíva 3D rozhranie. Obsahuje všetky prvky na stránke okrem samotného zobrazovania modelu a manipulácie s ním. Sú to prvky na vyhľadávanie informácií (miestností alebo osôb), na prepínanie medzi poschodiami, prepínanie medzi režimom prechádzania a prehliadania, zobrazovanie aktuálnej miestnosti (označenej, alebo tej, v ktorej sa používateľ nachádza) a prvky na navigáciu.

3D scéna

Zabezpečuje zobrazovanie modelu, jeho prezeranie, prechádzanie a interaktivitu. Využíva technológiu WebGL s pomocou knižnice CopperLicht. Komunikácia s doplnkovým UI predstavuje:

- Vzájomné informovanie o označenej miestnosti, resp. miestnosti v ktorej sa používateľ nachádza
- Aktualizácia polohy prvkov používateľského rozhrania, ktoré sa viažu k určitej 3D pozícii (napr. nad miestnosťou bude zobrazená nejaká informácia, ktorá sa však musí posunúť podľa toho ako bude používateľ pohybovať s modelom)
- Požiadavka na zmenu poschodia vyvolaná z GUI, alebo informácia, že používateľ prešiel na iné poschodie a GUI sa musí aktualizovať

2D a mobilné rozhranie

Zabezpečuje zobrazovanie pôdorysov poschodí v 2D režime, pre prehliadače bez podpory WebGL a mobilné zariadenia. Zabezpečuje prezeranie jednotlivých poschodí, zobrazovanie informácií o miestnostiach a navigáciu po budove.

10 Optimalizácia mobilnej verzie aplikácie

V 2D verzii určenej pre prehliadače môžeme počítať s vysokým rozlíšením displeja, (takmer) neobmedzeným množstvom prenášaných dát a vysokou pravdepodobnosťou dostupnosti technológie AJAX. Mobilná verzia je špecifická svojimi nasledujúcimi obmedzeniami:

- malý displej,
- obmedzenie množstva prenášaných dát,
- nedostupnosť technológie JavaScript,
- nedostupnosť technológie AJAX,
- nižšia prenosová rýchlosť.

Z obmedzení uvedených pre mobilnú verziu sme zostavili zoznam optimalizačných riešení, ktoré by mali pomôcť efektívne používať aplikáciu Virtuálna FIIT aj na starších modeloch mobilných zariadení:

- zvoliteľná veľkosť obrázku s plánom,
- rozkladanie obrázka s plánom (pri navigácii sa zobrazí len časť plánu budovy, po ktorej vedie cesta),
- textová navigácia,
- „server side“ navigácia (vykreslenie cesty bez potreby JavaScript-u)

11 Návrh GUI aplikácie

Projekt virtuálnej FIIT si vyžaduje tri grafické používateľské rozhrania. Rozhrania pre 3D a 2D verziu sa budú mierne líšiť. Hlavným rozdielom je spôsob interakcie s aplikáciou, ktorý má následne vplyv na používateľské rozhranie. Hlavné rozdiely, princípy a komponenty opíšeme v nasledujúcich častiach pre jednotlivé verzie aplikácie.

11.1 Grafické rozhranie pre 3D verziu

3D verzia ponúka používateľovi najviac interaktivity, preto bude GUI pre túto verziu ponúkať najviac možností. Rozhodli sme sa použiť pre GUI pre 3D verziu obdobný princíp ako využíva napríklad Google Maps. Jednotlivé komponenty ovládania sú vykresľované na neviditeľnú vrstvu nad samotným modelom budovy, prípadne budú vykresľované priamo do modelu. Týmto prístupom dosiahneme takmer nepozorovateľnú integráciu GUI so samotnou aplikáciou a vyhneme sa tak segmentácii aplikácie do časti s modelom, časti s vyhľadávaním a menu.

3D verzia našej aplikácie bude používateľovi poskytovať tri módy prehliadania modelu. Preto bude GUI obsahovať tri tlačidlá, ktoré budú meniť režimy zobrazenia:

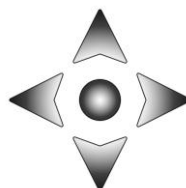
- tlačidlo na zobrazenie módu z pohľadu prvej osoby: model sa zobrazí v režime simulujúcom pohľad, ktorý by sa naskytol používateľovi ak by prechádzal chodbami budovy.
- tlačidlo na zobrazenie módu navigácie: mód navigácie zobrazí jedno, prípadne vrstvene viacero poschodí (podľa nastavenia), tak aby bolo možné prehľadne zobraziť vypočítanú trasu. Zo spomínaného dôvodu, bude rotácia do značnej miery obmedzená (používateľ nepotrebuje vidieť model napríklad zospodu)
- tlačidlo na zobrazenie módu takzvaného voľného letu: tento režim poskytuje používateľovi maximálnu voľnosť, používateľ môže lietať priestorom akýmkoľvek smerom a bez žiadnych zábran

Každý z týchto troch módov bude podporovať ovládanie myšou a klávesnicou, avšak bude možné ovládanie aj čisto len myšou, preto budú na najvyššej vrstve stále dostupné ovládacie prvky na manipuláciu s modelom (prehliadanie). Konkrétne ide o nasledovné tri elementy:

- rotácia modelu



- posun model



- zväčšenie / zmenšenie modelu



Samotný model bude podporovať interakciu s používateľom. Bude umožňovať kliknúť na konkrétnu miestnosť, kde sa následne zobrazí ponuka v jednoduchom a prehľadom dialógovom okne. Podobným spôsobom bude fungovať aj ostatná funkcionálna, ako je napríklad zobrazenie rozvrhu prihláseného používateľa.

11.2 Grafické rozhranie pre 2D a mobilnú verziu

Grafické rozhranie je pre 2D a pre mobilnú verziu aplikácie v podstate spoločné. Obe verzie sa od 3D verzie výrazne líšia. GUI je jednoznačne rozdelené do jednotlivých komponentov a to:

- 2D plán budovy: obrázok plánu jedného poschodia budovy
- textová navigácia: opisuje v krokoch ako sa používateľ dostane z bodu A do bodu B
- panel s nástrojmi: vyhľadávanie cesty, miestnosti, osoby, atď.

Komponent zobrazujúci 2D plán budovy bude tak, ako v 3D verzii obsahovať v najvyššej vrstve elementy na prácu s obrázkom:

- zväčšenie / zmenšenie plánu: bude poskytovať len pár úrovní, medzi ktorými sa bude môcť používateľ prepínať
- posun plánu

Komponent s textovou navigáciou bude obsahovať v bodoch opísaný postup ako má používateľ prechádzať budovou, tak aby trafil do hľadanej miestnosti. Panel s nástrojmi bude jednoduchý formulár, prípadne menu, ktoré poskytne používateľovi všetky špecifikované možnosti.

Normálna a mobilná verzia sa odlišujú sa len nasledovných bodoch:

- mobilná verzia musí byť dátovo nenáročná, preto bude kladený dôraz na prehľadnosť a jednoduchosť
- mobilná verzia bude zobrazovať naraz vždy len jeden komponent, kým normálna zobrazí súčasne všetky komponenty naraz

12 Dátový model

Ako databázový systém pre náš projekt sme vybrali databázu MySQL dôvodom je, že je dostupná bezplatne a najmä je tento systém potrebný aj pre iné časti projektu a bolo by zbytočné mať aktívne dva rôzne databázové systémy súčasne, najmä keď naše požiadavky na funkcionality spĺňajú prakticky všetky súčasné databázové systémy.

Z predchádzajúcej analýzy potrebných a dostupných dát sme vytvorili logický (Obr. 12.1) a fyzický (Obr. 12.2) dátový model. Oba sú vytvorené v IBM Rational Data Architect v7.

12.1 Opis dátových entít

Tu sa nachádza opis jednotlivých dátových entít. Všetky dátové entity majú ako primárny kľúč, stĺpec s menom ID typu Integer, preto ho nebudem pri jednotlivých entitách uvádzať. Dátové typy je možné vidieť v diagramoch preto ich opätovne nebudem uvádzať v prehľade dátových entít.

User

Táto entita predstavuje zamestnancov fakulty.

- First_Name – predstavuje krstné meno zamestnanca
- Surname – predstavuje priezvisko zamestnanca
- Title_Before_Name – tituly zamestnanca pred menom
- Title_After_Name – tituly zamestnanca za menom
- Ais_ID – ID číslo používané v AIS
- Room – pracovňa zamestnanca (ID cudzí kľúč na entitu Room)

Timetable

Táto entita predstavuje jednotlivé rozvrhové akcie.

- Time_From – čas odkedy trvá rozvrhová akcia
- Time_To – čas dokedy trvá rozvrhová akcia
- Note – poznámka k rozvrhovej akcii
- Restriction – obmedzenia rozvrhovej akcie
- Day – deň v týždni (ID cudzí kľúč na entitu Day)
- Course – predmet, ktorého sa rozvrhová akcia týka (ID cudzí kľúč na entitu Course)
- Teacher – učiteľ, ktorý vedie danú rozvrhovú akciu (ID cudzí kľúč na entitu User)
- Room – miestnosť v ktorej sa rozvrhová akcia koná (ID cudzí kľúč na entitu Room)

Phone

Táto entita predstavuje telefón zamestnanca fakulty.

- Phone – telefónne číslo
- User – zamestnanec, ktorému patrí telefónne číslo (ID cudzí kľúč na entitu User)

Email

Táto entita predstavuje e-mail zamestnanca fakulty.

- Email – e-mail
- User – zamestnanec, ktorému patrí e-mail (ID cudzí kľúč na entitu User)

Course

Táto entita predstavuje predmet, ktorý má vytvorené nejaké rozvrhové akcie.

- Name – názov predmetu

Day

Táto entita predstavuje deň v týždni

- Name – názov dňa v týždni
- Abb – skratka dňa v týždni

Room

Táto entita predstavuje miestnosti v univerzity.

- Name – názov miestnosti
- Building – budova v ktorej sa miestnosť nachádza (ID cudzí kľúč na entitu Building)

Building

Táto entita predstavuje budovy univerzity.

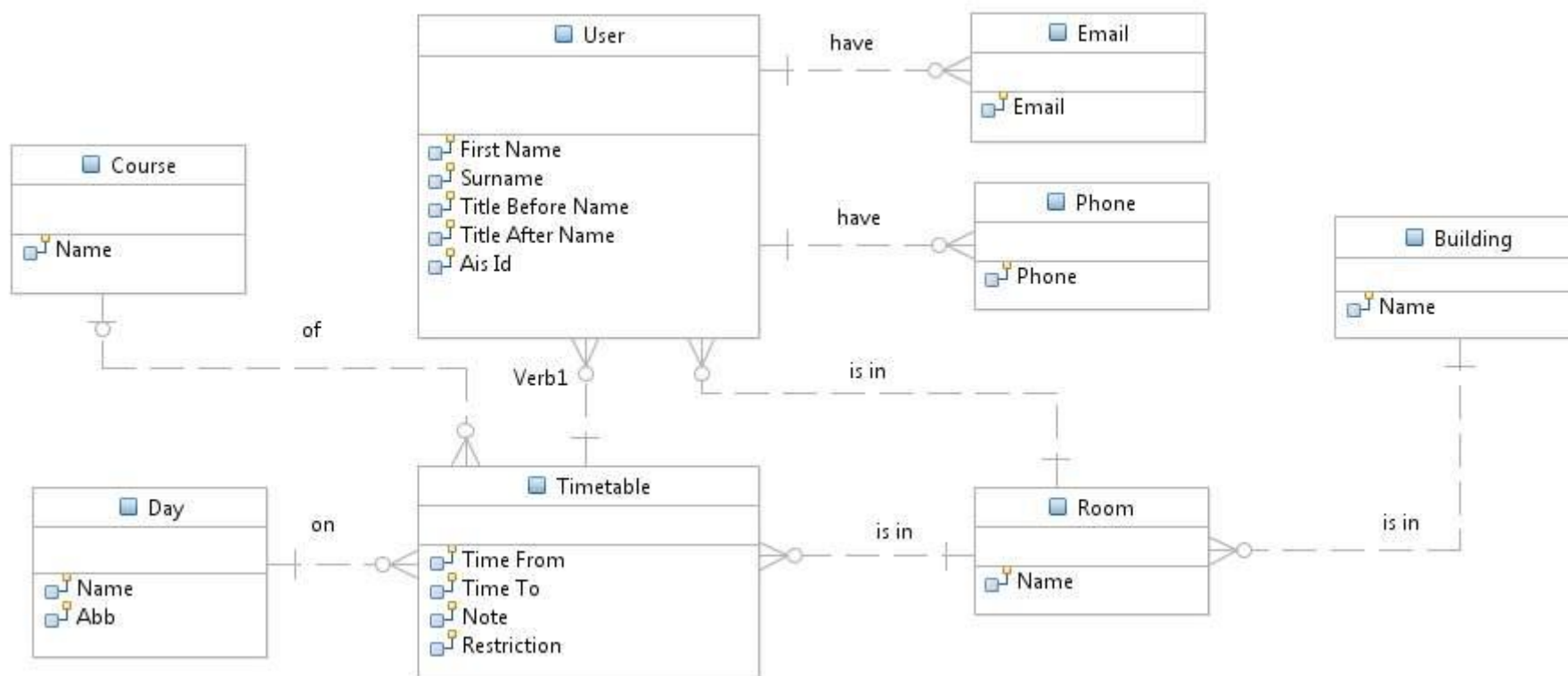
- Name – názov budovy

Okrem týchto dátových entít sa bude nachádzať v databáze ešte dočasná entita Room_FIIT. Ide o špeciálnu tabuľku na mapovanie existujúcich rozvrhov v reálnych miestnostiach budovy fakulty FEI na dosiaľ nedokončené miestnosti novej budovy fakulty FIIT. Po dokončení budovy a vytvorení skutočných rozvrhov v novej budove FIIT bude táto pridaná do tabuľky Building a miestnosti sa budú nachádzať v tabuľke Room. Táto entita nie je zobrazená v logickom ani fyzickom dátovom modeli, keďže ide len o dočasnú tabuľku, potrebnú, kým nebude dokončená budova FIIT a nebudú rozvrhové akcie prebiehať priamo v tejto budove.

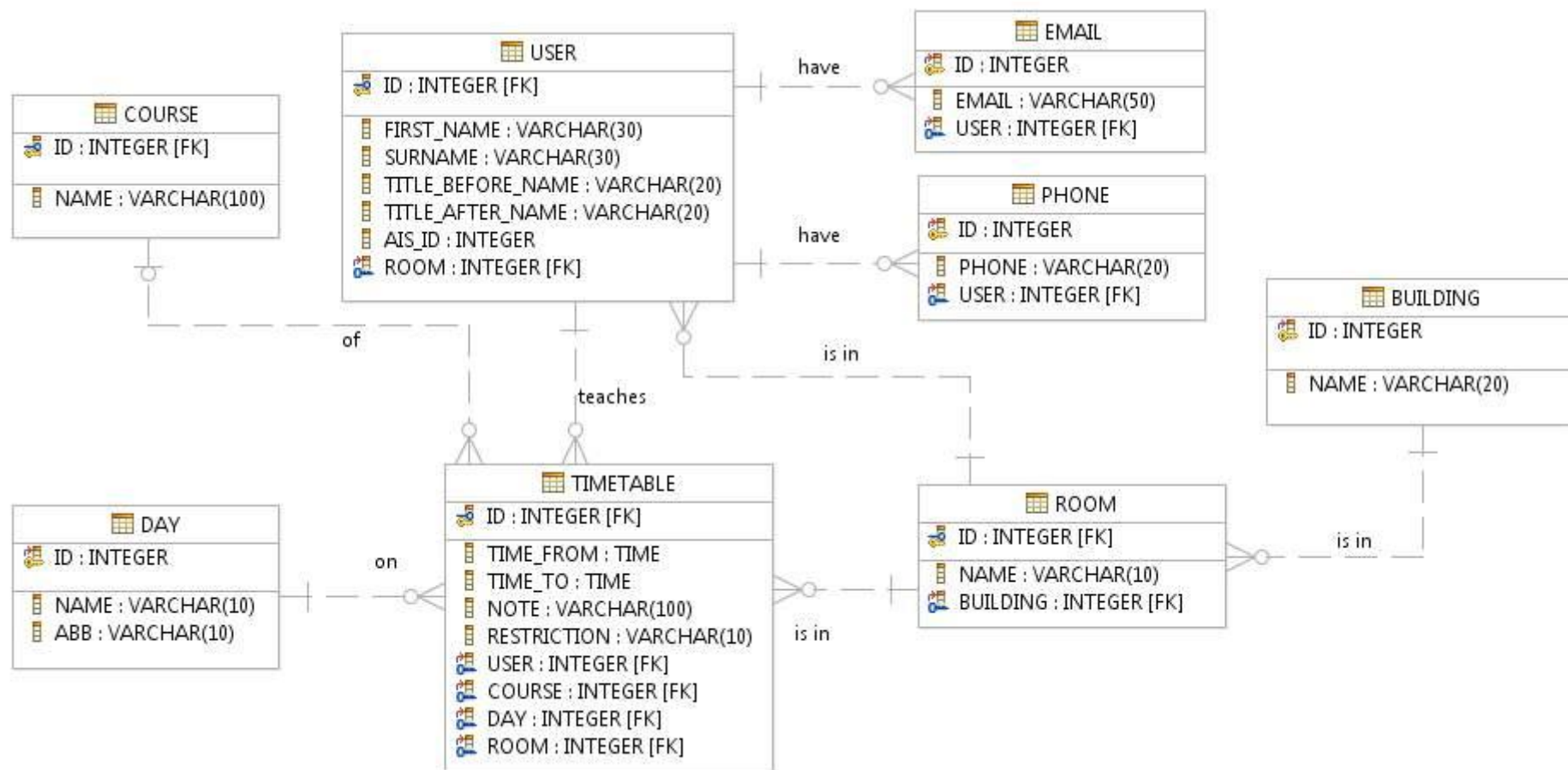
Room_FIIT

Táto entita predstavuje dočasné mapovanie dosiaľ neexistujúcich miestností na reálne miestnosti fakulty FEI.

- Name – názov miestnosti v novej budove FIIT
- Room – miestnosť v existujúcej budove fakulty FEI (ID cudzí kľúč na entitu Room)



Obr. 12.1 Logický dátový model



Obr. 12.2 Fyzický dátový model

13 Priority implementácie

V zimnom semestri budeme vyvíjať evolučný prototyp, z ktorého v letnom semestri vyvineme výsledný produkt. V prototypy sú naše priority implementácie nasledovné:

- 3D klient
 - o funkčnosť minuloročného riešenia - prehliadanie, prechádzanie, označovanie miestností
 - o neprechádzanie cez dvere
 - o výťah
 - o zobrazovanie informácií nad miestnosťou
- 2D klient
 - o vytvorený plán aspoň jedného poschodia
 - o označovanie miestností
 - o vyhľadávanie a zobrazovanie miestností
- Mobilný klient
 - o vytvorený plán aspoň jedného poschodia
- Ku všetkým rozhraniam klienta
 - o prototyp používateľského rozhrania
- Model
 - o spísané potrebné aktualizácie modelu
 - o aktualizované a upravené minimálne jedno poschodie
- Server
 - o import údajov z AIS
 - o komunikačné rozhranie na serverovej aj klientskej strane

Pre všetky uvedené časti platí, že v prototypy sa nezameriavame na používateľskú prívetivosť ani na kompletnosť. Stačí teda, keď v prototypy bude iba jedno poschodie, a nebudú importované a zobrazované všetky údaje z AIS, ale len vybrané.

V letnom semestri z hľadiska implementácie plánujeme:

- implementovať navigáciu
- domodelovať zvyšné poschodia
- vytvoriť zvyšné plány poschodí pre 2D a mobilného klienta
- implementovať funkcionality, ktorá bola v prototypy implementovaná iba čiastočne
- implementácia získavania osobných rozvrhov z AIS
- optimalizácia rýchlosti načítavania a množstva prenášaných údajov