

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Virtuálna FIIT

Projektová dokumentácia

Vedúci projektu: Mgr. Alena Kovárová, PhD.

Autori: Bc. Samo Forus Bc. Jozef Kujan
Bc. Igor Aufricht Bc. Juraj Tóth
Bc. Dušan Poizl Bc. Juraj Volentier
Bc. Matej Škoda

Kontakt: tim9@wx.sk

Ak. rok: 2011/2012

1	Zadanie	5
2	Úvod	5
3	Features.....	5
3.1	Mobilný klient	6
3.1.1	Multiplatformový mobilný klient.....	8
3.1.2	Mobilný klient - grafická navigácia	9
3.1.3	Mobilný klient - textová navigácia	11
3.2	Real-time komunikácia klienta so serverom	11
3.2.1	Server riešenie pre websockety	12
3.3	Menovky na dverách	12
3.3.1	Menovky - serverová strana.....	14
3.3.2	Optimalizácia menoviek	14
3.4	Vysúvací panel.....	15
3.5	Celoobrazovkový mód a navigačné prvky	17
3.5.1	Zobrazovanie iba celoobrazovkového módu	18
3.5.2	Úprava rozloženia ovládacích prvkov.....	18
3.5.3	Získanie údajov o prepojení miestností z navigačného grafu	20
3.5.4	Presun načítavania navigačného grafu pred načítavanie textúr	20
3.6	3D Model.....	21
3.6.1	Svetlá a tieň v 3D modeli	23
3.6.2	Výškové mapy.....	24
3.6.2.1	Neviditeľné zábradlia a chyby kolíznych máp	24
3.6.2.2	Vytvorenie polygónov typu Base.....	25
3.6.2.3	Vytvorenie výškovej (kolíznej) mapy modelu	26
3.6.2.4	Testovanie nových výškových máp a kolízií	26
3.7	WebGL	27
3.7.1	Aktualizácia WebGL	28
3.7.2	Antialiasing vo WebGL.....	28
3.8	Knižnice pre websockety.....	28
3.9	Testovanie importéra	28
4	Nájdeneé chyby	29
4.1	Výťah.....	29
4.2	Déjà vu	30
4.3	Zvuky chôdze a iné.....	31
4.4	Dvere	31
4.5	Chyby modelu	31
4.6	Webové rozhranie	34
4.7	Rozhranie.....	36
4.8	Iné drobné chyby	37

4.9	Oprava bugov – prepojenie poschodí	39
5	Mobilná aplikácia - PhoneGap	40
5.1	Editor navigačného grafu	44
5.1.1	Analýza požiadaviek	44
5.1.2	Návrh.....	44
5.1.3	Implementácia	44
5.2	Ohraničenie miestností	45
5.2.1	Analýza	45
5.2.2	Editor.....	45
5.2.3	Program.....	45
5.3	Zobrazenie osobného rozvrhu	46
5.3.1	Analýza	46
5.3.2	Návrh.....	46
5.4	Prevod bitmapových obrázkov do SVG.....	47
5.5	Zobrazenie mapy v HTML tabuľke.....	48
5.5.1	Zobrazenie mapy	48
5.5.2	Posúvanie mapy	48
5.5.3	Výber miestnosti	48
5.6	Vyhľadávanie osôb a miestností	50
5.6.1	Vyhľadávanie	50
5.6.2	Zobrazenie informácií o osobách, miestnostiach a predmetoch	50
5.6.3	Offline mód	50
5.7	Lokalizácia používateľa	51
5.7.1	Android	51
5.8	Mapy mobilnej aplikácie	52
5.9	Analýza aplikácie iTransit.....	54
5.9.1	Okno (záložka) departures (odchody autobusov)	54
5.9.2	Okno (záložka) around me (Kde som?)	55
5.9.3	Okno (záložka) stops (zastávky).....	56
5.9.4	Okno (záložka) lines (linky)	57
5.10	Zobrazenie cestovného poriadku	58
5.10.1	Zastávky mestskej verejnej dopravy v oblasti budovy FIIT	58
5.10.2	Spoje prichádzajúce/odchádzajúce z jednotlivých zastávok.....	58
5.10.3	Dostupnosť údajov	59
5.10.4	Import dát do databázy VirtualFiit	59
5.10.5	Dopyt dát mobilnou aplikáciou.....	59
5.10.6	Návrh tabuliek.....	60
5.10.7	Implementácia	60
5.11	Analýza použitia databázy na mobilných zariadeniach	61

5.12	Analýza možnosti použitia jedalen.stuba.sk	62
5.12.1	Dátový model	62
5.12.2	Opis dátových entít	63
5.13	Offline mód	64
5.14	Možné rozšírenia mobilnej aplikácie v budúcnosti	65
6	Zhodnotenie	66
7	Zdroje	67
A.	Mobilná aplikácia - používateľská príručka	68
A.1	Ovládanie	68
A.2	Mapa	68
A.3	Hlavná obrazovka	69
A.4	Odfot' kód	72
A.5	Info	72
A.5.1	Osobný rozvrh	73
A.5.2	MHD	73
A.5.3	FIIT	74
A.5.4	Jedálny lístok	74
A.5.5	AIS	74
A.6	Navigácia	74
A.7	Hľadať	74
A.8	Nastavenia	75
A.9	Vysvetlenie obrázkov na mape	76
A.10	Farebné označenie miestností na mape	77
B.	GraphEditor - používateľská príručka	78
B.1	Hlavné okno	78
B.2	Hlavné menu	79
B.2.1	Menu File	79
B.2.2	Menu Floor	79
B.2.3	Menu Elevators	79
B.3	Módy editora	81
B.3.1	Mód Panning	81
B.3.2	Mód Edit	81
B.3.3	Mód Move	82
B.3.4	Mód Room edit	82
C.	Príspevok na konferenciu IIT.SRC 2012	83
D.	Projektový poster zo súťaže TPCup	85

1 Zadanie

Virtuálna FIIT

Vedúci tímu: Mgr. Alena Kovárová, PhD.

Pamätáte si, ako vždy začína každý semester? Zistíte si, aký je váš rozvrh, no v ňom sú záhadne zakódované čísla miestností, v ktorých sú 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 treba 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?

Úlohou bude osvojiť si a vylepšiť predchádzajúce riešenie (napr. tiene, antialiasing, použitie nových WebGL knižníc, aktualizovať 3D model, atď.), do úvahy pripadajú viaceré rozšírenia systému. Napríklad nové interaktívne časti, ako sú napríklad informačné tabule, možnosť zadania otázky, špeciálnou výzvou je vizualizovanie používateľov vo virtuálnom priestore formou avatarov, ktoré môžu spolu komunikovať. Ďalším prínosom by bola personalizácia systému formou prihlásenia sa doň (napr. načítať fotku z AIS-u a namapovať ju na tvár avatara). To všetko by samozrejme nešlo bez databázy a možnosti importovania do nej z iných existujúcich systémov (nielen AIS ale napr. aj FIIT-kárske RSS, jedáleň a pod.). Riešenie bude potrebné optimalizovať tak, aby ho používateľ mohol používať aj cez jednoduché webové rozhranie, či na mobilnom zariadení. Bonusom je návrh a implementácia lokalizácie používateľa v priestore na základe miesta jeho pripojenia a jeho voliteľným zverejnením (napr. formou už spomínaného avatara).

Tento projekt nadväzuje na minuloročné, v prípade záujmu kliknite na [predminuloročný projekt](#) alebo [minuloročný projekt](#). Pozrite sa v akom stave je [momentálne zrealizovaná Virtuálna FIIT](#) (ak sa vám nenačítala stránka, nahraďte 159-tku 99-tkou).

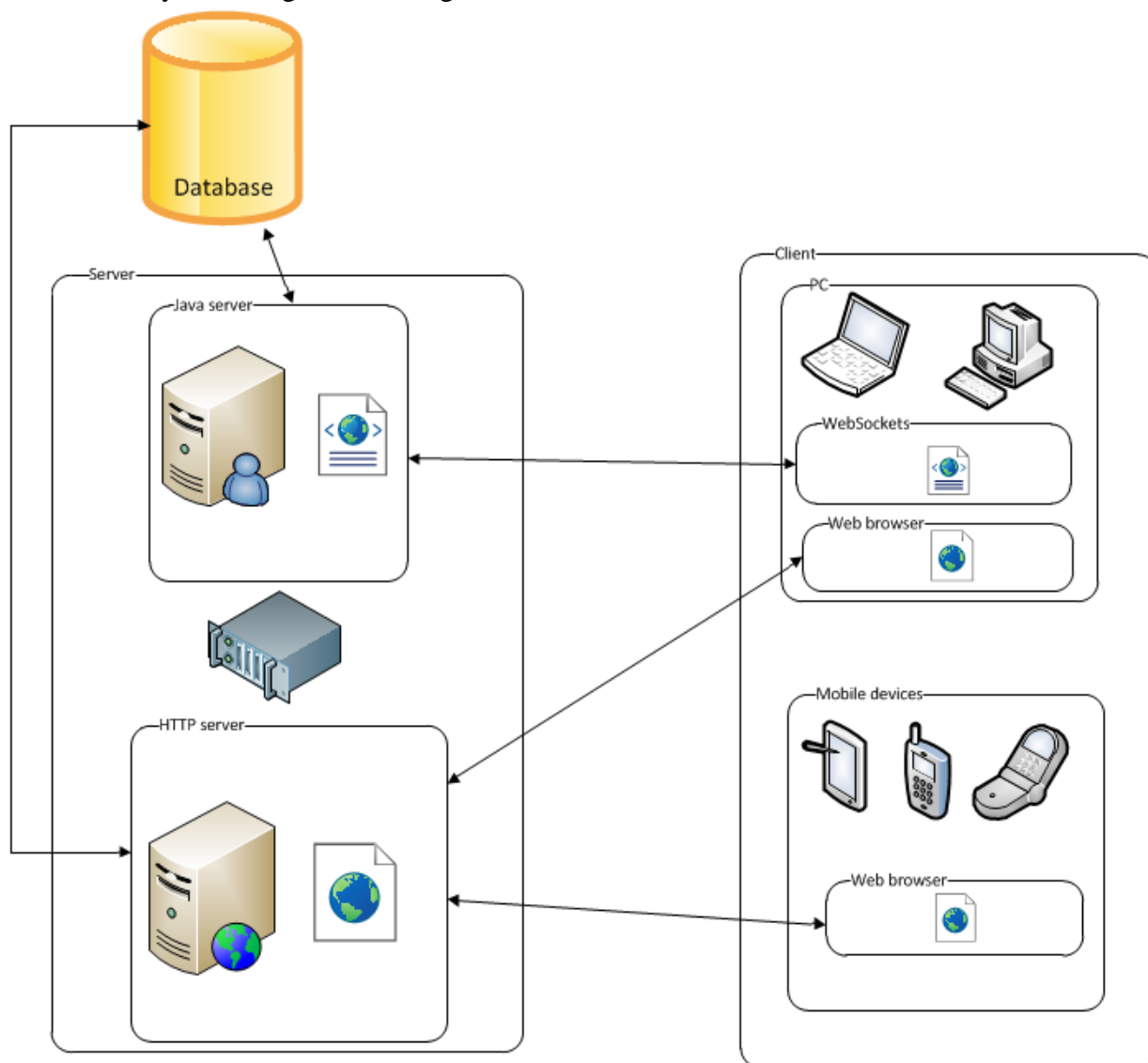
2 Úvod

Predkladaný dokument obsahuje projektovú dokumentáciu vytvorenú k predmetu Tímový projekt. Tento dokument slúži ako podklad k projektu s názvom Virtuálna FIIT, ktorý je pokračovaním projektu minuloročného tímu SW7D. V dokumente sa nachádza postup a informácie k jeho riešeniu.

3 Features

- Import modelu
- Analyzovať technológie pre real-time komunikáciu so serverom
- Analyzovať možnosti pre mobilného klienta
- Hľadanie bugov
- Antialiasing
- WebGL
- Mobilný klient – analýza
- 3D model – export, maxscript
- Menovky – klientská strana
- Upraviť vysúvací panel

- Analýza možnosti vytvorenia natívnej mobilnej aplikácie
- Oprava bugov – model
- Oprava bugov – kolízie
- Oprava bugov – prepojenie poschodí
- Spustenie mobilnej verzie na localhoste
- Menovky – serverová strana
- Mobilný klient – grafická navigácia
- Mobilný klient – textová navigácia
- Oprava bugov – výťah
- Lightmapy
- Neviditeľné zábradlia schodísk
- Výškové mapy
- Optimalizácia menoviek
- Redizajn webových prvkov



Obr. 1: Nákres novej plánovanej architektúry aplikácie

3.1 Mobilný klient

analýza súčasného stavu

Mobilný klient je vytvorený v jazyku PHP s frameworkom Codeigniter.

Prezentačná vrstva – kombinácia HTML a CSS

Plán poschodia – obrázok vo formáte GIF

Menu obsahuje 4 časti:

- Fakulta – všeobecné info o fakulte, zameraní, vzniku a kontaktné údaje
- Ľudia na fakulte – neobsahuje žiadne info
- Miestnosti – neobsahuje žiadne info
- Budova – prehliadanie pôdorysu poschodia

Prehliadanie plánu momentálne funguje v 2 režimoch:

- čiastočný – zobrazuje len časť budovy

šírka je nastavená na 200px

posun sa vykonáva tlačidlami na obrazovke

programovo sa generuje časť obrázka, ktorá sa má zobrazit'

- úplný – zobrazí celú budovu a posúvanie sa je ponechané na browser

Používateľské rozhranie je prispôsobené obmedzeným možnostiam mobilných zariadení.

Odporúčania pre budúcnosť

Zvažovali sme 2 možnosti:

- vylepšenie fungujúcej verzie
- vytvorenie nového natívneho klienta
 - Android – založený na Java
 - iPhone – objektové C (hlavný problém – je potrebné mať Mac)

Analýza možností na vytvorenie natívneho klienta

Flex podporuje nasledujúce mobilné platformy:

- Android
- iOS
- BlackBerry Tablet OS
- televízia

Do Flexu pre mobilné zariadenia bolo potrebné pridať niektoré veci a tými sú:

- mobilné komponenty
- mobilné témy vzhľadu
- výkon
- export aplikácie v natívnom formáte

Výhody Flexu:

- Produktívny vývoj – Flex je silný komponentový framework.
- Využívané jazyky sú blízke webovému vývoju (ľahko sa učia)
- Rovnaký jazyk aj API pre rôzne platformy
- Publikovanie pre rôzne platformy jedným klikom myši
- Flex je open source

Navigácia

Zdrojové kódy mobilného klienta sú obrázky. Tieto sa negenerujú na serveri, ale sú uložené na serveri už hotové a iba sa použije obrázok potrebného podlažia.

3.1.1 Multiplatformový mobilný klient

V poslednom čase sa dostávajú do popredia frameworky umožňujúce multiplatformový vývoj pre mobilné platformy. Najznámejší open-source framework podporujúci až 7 platforiem je PhoneGap.

PhoneGap podporuje tieto platformy:

- iOS
- Android
- Windows Mobile / Windows Phone
- Blackberry OS
- WebOS
- Symbian
- Bada

Tieto platformy sa v mnohom líšia a neumožňujú vytvárať spoločný natívny kód preto, že ich API je dostupné pre rôzne programovacie jazyky (napr. objective-C pre iOS či C/C++ pre Android).

PhoneGap sprístupňuje natívne funkcie pomocou JavaScriptu a moderných webových technológií HTML5 a CSS3. Aplikácia je teda vytváraná nezávisle na platforme a pri vybudovaní prostredníctvom PhoneGap Build je aplikácia (teda JS, HTML a CSS) obalená frameworkom a vzniká natívna aplikácia samostatne pre každú platformu.

PhoneGap usiluje o kompletne pokrytie natívnej funkcionality, ako môžeme vidieť na nasledujúcom obrázku:

	 iOS iPhone / iPhone 3G	 iOS iPhone 3GS and newer	 Android	 OS 4.6-4.7	 OS 5.x	 OS 6.0+	 WebOS	 WP7	 Symbian	 Bada
ACCELEROMETER	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
CAMERA	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
COMPASS	✗	✓	✓	✗	✗	✗	✗	✓	✗	✓
CONTACTS	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓
FILE	✓	✓	✓	✗	✓	✓	✗	✓	✗	✗
GEOLOCATION	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MEDIA	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗
NETWORK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (ALERT)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (SOUND)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (VIBRATION)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
STORAGE	✓	✓	✓	✗	⚠	✓	✓	⚠	✓	✗

Obr. 2: Natívna funkcionálna pokryť PhoneGapom

To aplikácii umožňuje využiť sieťovú komunikáciu či dokonca geolokáciu a na moderných platformách i kompas, kameru a podobne.

3.1.2 Mobilný klient - grafická navigácia

Vykreslenie smerovej šípky do plánu budovy.

Kedže je celá stránka naprogramovaná v programovacom jazyku php, prioritne sme analyzovali možnosti v PHP.

Knižnica GD umožňuje prácu s obrázkami vo viacerých formátoch (jpg, png, gif ...)

Obsahuje funkciu:

imageCopyMerge(\$dstImage, \$ptrImage, \$ptr_x, \$ptr_y, 0, 0, \$ptr_w, \$ptr_h, pct)

Kde \$dstImage a \$ptrImage sú obrázky ktoré daná funkcia spojí a ďalej sú súradnice kam sa má vložiť obrázok \$ptrImage. Táto knižnica je pomerne používaná a vyhovujúca aj pre náš projekt.

Roly používateľov

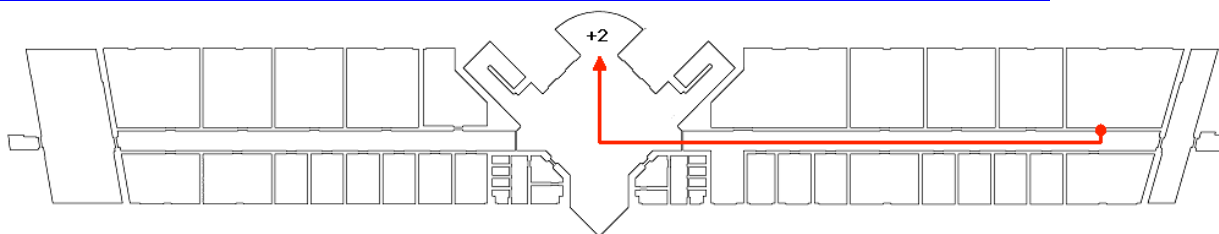
Číslo US	Rola	Cieľ	Dôvod
----------	------	------	-------

	používateľ'a		
1	Používateľ	Schopnosť cez mobil vyhľadať miestnosť, profesora, alebo miestnosť kde sa vyučuje predmet	Uľahčenie života používateľom
2	Používateľ	Vypísanie možných miestností a z nich možnosť si vybrať	Zlepšenie práce so systémom
3	Používateľ	Zobrazenie šípok pre navigáciu používateľ'a	Získať mapu na miesto určenia používateľ'a

Do budúcnosti sa uvažuje pridať ešte prihláseného používateľ'a ako samostatnú rolu.

Úlohy k jednotlivým US

Číslo úlohy	Popis úloh	Náročnosť	Priorita
1.1	Naštudovať ďalšie funkcie v php	nízka	nízka
2.1	Analyzovať možnosti ako najprehľadnejšie vypísať zoznam	nízka	nízka
3.1	Návrh a implementácia algoritmu na vyhľadávanie cesty	stredná	stredná
3.2	Vytvorenie mapy bodov, s ktorými bude algoritmus pracovať	stredná	stredná
3.3	Vytvoriť funkciu na kreslenie šípky	nízka	stredná



Obr. 3: Ukážka zobrazenia šípok grafickej navigácie

Na vyhľadávanie cesty sa využíva mierne upravený algoritmus hľadania do hĺbky, ktorý prehliada mapu bodov a získava body cez ktoré treba prejsť, aby sa dostal do cieľa. Následne sa vykreslia medzi jednotlivými bodmi na tejto mape čiary a šípka udávajúca smer, pričom sa kreslí len 4 smermi – hore, dole, doľava, doprava.

Toto pre 2D mapu postačuje pri vhodne zvolených bodoch.

Body treba zadávať vo formáte XTY, kde:

- X je číslo poschodia
- T je textové označenie miestnosti
- Y je číselné označenie miestnosti

Textové označenie vyjadruje, na akej svetovej strane sa miestnosť nachádza alebo či sa nachádza v centre. Napríklad:

3SW2 – 3. poschodie, juhozápadná miestnosť číslo 2

5E1 – 5 poschodie, východná miestnosť číslo 1

1NE3 – 1. poschodie, severovýchodná miestnosť číslo 3

5C1 – 5. poschodie, centrum číslo 1

2CW5 – 2. poschodie, miestnosť číslo 5 v západnom centre

3.1.3 Mobilný klient - textová navigácia

V existujúcej aplikácii je riešená navigácia pomocou grafu bodov v priestore. Z nich sa nedajú extrahovať vhodné dáta na slovnú navigáciu.

Roly používateľov

Číslo US	Rola používateľa	Cieľ	Dôvod
1	Používateľ	Získať inštrukcie ako sa dostať k cieľu	Slovné inštrukcie môžu byť zrozumiteľnejšie ako šípky. Taktiež sú vhodné pre nevidiacich keďže ich potom dokáže prečítať čítačka.

Úlohy k jednotlivým US

Číslo úlohy	Popis úlohy	Náročnosť	Priorita
1.1	Vytvorenie navigačných dát	stredná	vysoká
1.2	Napísanie algoritmu na vytvorenie cesty	stredná	vysoká
1.3	Vytvorenie používateľského rozhranie	nízka	stredná

3.2 Real-time komunikácia klienta so serverom

Nápad pridania avatarov a ich interakcie do existujúcej aplikácie virtuálnej FIIT vyžaduje obojsmerné spojenie so serverom. Obojsmerné spojenie musí byť dostatočne rýchle

a škálovateľné. Rozšírenie serverovej strany pre real-time komunikáciu s klientom teda vyžaduje zvolenie správnej technológie. Našli sme štyri rôzne spôsoby rozšírenia komunikácie. Nasledujúca tabuľka popisuje výhody a nevýhody jednotlivých možností (viď tabuľka 1).

Tab. 1: Výhody a nevýhody real-time komunikácie

Názov technológie	Výhody	Nevýhody
WebSocket	Technológia typu PUSH Server môže inicializovať požiadavku	Nová technológia Nie je ešte úplne schválená W3C
Ajax	Dobre podporovaná technológia	Server nemôže inicializovať požiadavku, iba klient
Plugin do prehliadača	Možnosť použitia natívneho kódu	Zložitejšia implementácia Nekompatibilita prehliadačov
Samostatná aplikácia	Natívna aplikácia – nie je obmedzené prehliadačom	Strata multiplatformovosti

Po diskusii sme sa rozhodli že najlepšou voľbou pre použitie našej komunikácie budú práve WebSockets. Napriek tomu, že sa jedná o mladú technológiu, s možnými zmenami do budúcnosti je dobre podporovaná v súčasných prehliadačoch. Poskytuje presne to, čo potrebujeme a jej implementácia na klientskej strane nie je vôbec náročná. Klientská implementácia bude doplnená do existujúceho klientskeho JavaScriptu.

3.2.1 Server riešenie pre websockety

Vybraté riešenie pre komunikáciu pomocou WebSocketov bude potrebovať implementáciu na strane servera. Rozhodli sme sa vybudovať druhý paralelný server s webovým serverom, ktorý bude práve obsluhovať websocket požiadavky. Bude napísaný v programovacom jazyku Java.

Základom riešenia bude použitie existujúcej knižnice pre komunikáciu cez websockety napísaná v Jave. V tejto oblasti sme našli dve existujúce riešenia. Prvým je WebSocket4J a druhé je jWebSocket. WebSocket4J má momentálne zastavený vývoj a je pod licenciou GPL. Na druhú stranu jWebSocket je vyvíjaný aktívnejšie, poskytuje viac dokumentácie a je pod voľnejšou licenciou LGPL. Preto sme sa rozhodli pre real-time komunikáciu s klientom použiť knižnicu jWebSocket.

3.3 Menovky na dverách

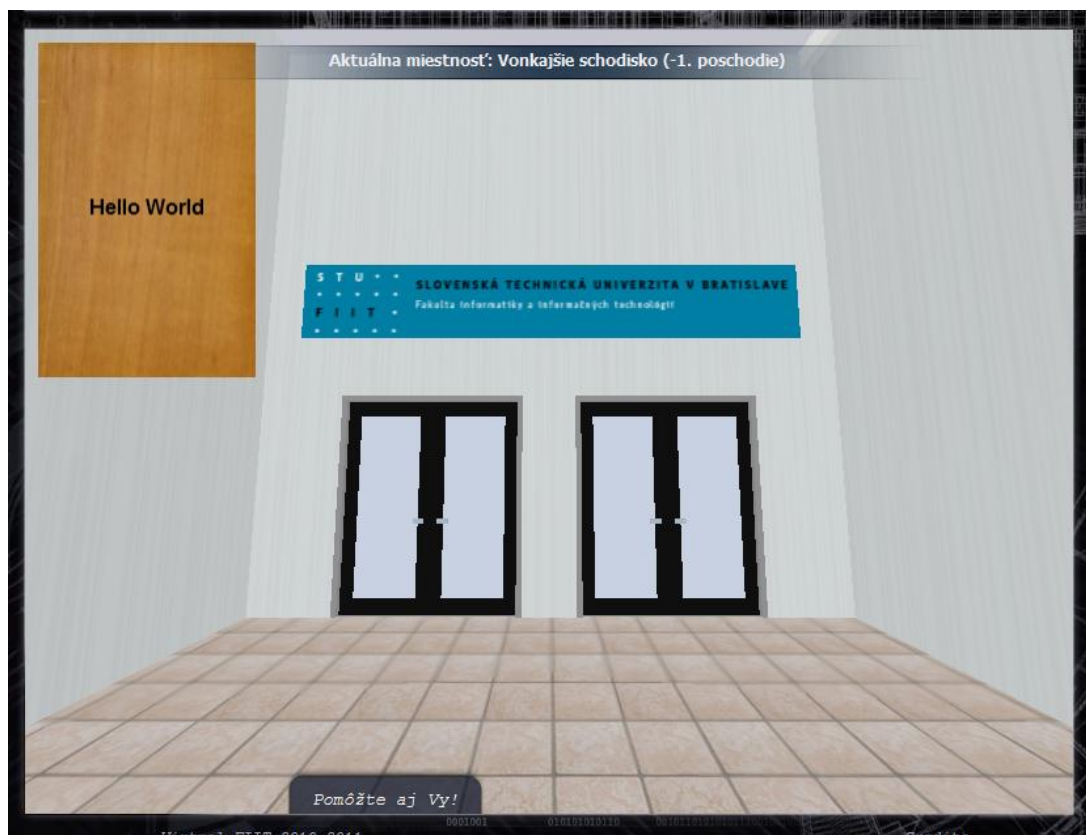
Menovky na dverách učební budú zobrazovať meno učiteľa, ktorému táto učebňa patrí.

Našli sme 3 možné riešenia ako pridať text do projektu:

- nápisy priamo v modeli
- nápisy na textúre
- nápisy dynamicky vo WebGL

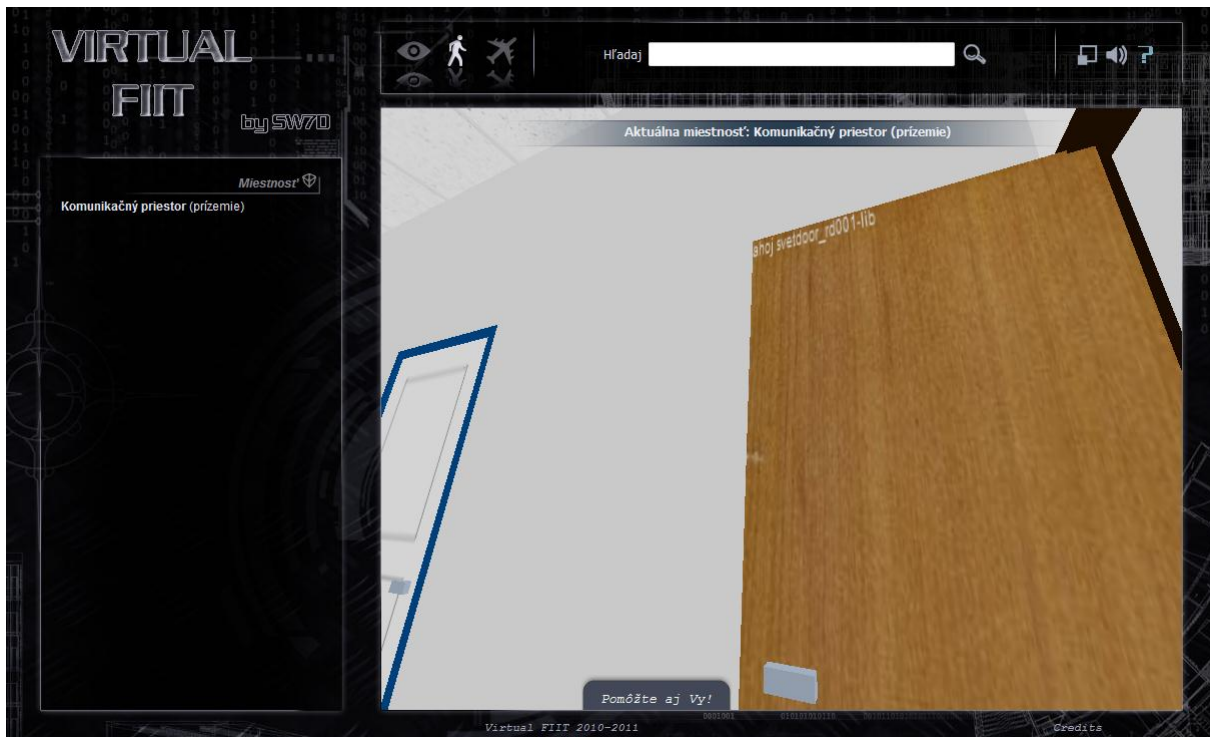
Predošlé predpokladané riešenie s pridaním textu pomocou WebGL sme zistili, že nebude možné vzhľadom na použitý engine Copperlicht. Nápisy priamo v modeli tiež nie sú vzhľadom na predpokladané zmeny dobrým riešením. Preto sme sa zamerali na prídanie menovky dynamicky na textúru.

Vytvorili sme Overlay 2D vrstvu (CL3D.Overlay2DSceneNode). Overlay 2D scéna podporuje prídanie textu spolu s pozadím inej textúry. Výsledok je vidno na obrázku. Následne sme chceli použiť konverziu na Textúru pomocou `createTextureFrom2DCanvas(canvas)`. Funkcia zbehne v poriadku, ale premenná `image` vnútri `Texture` je nastavená na `null`, čo znamená že nejde použiť v modeli a namiesto textúry spraví čiernu farbu. Problém bol reportovaný na fórum stránky enginu copperlicht.



Obr. 4: Prídanie textu s pozadím inej textúry na Overlay 2D vrstvu

Následne sme prišli s myšlienkou, nevytvárať Overlay2D, ale vytvoriť nápis pomocou `document.createElement("canvas")`. Tento spôsob fungoval oveľa lepšie a dokázali sme vypísať na dvere ich priradené ID. Zatiaľ to je viditeľné iba na otvorených dverách, keďže zavreté dvere majú iba jednu inštanciu pre všetky kvôli optimalizácii. Túto bude nutné upraviť, alebo úplne odstrániť.



Obr. 5: Nápis pomocou `document.createElement("canvas")`

Ďalším problémom je získanie ID miestnosti, ak vieme ID dverí. Keď sa nám podarí získať ID miestnosti, môže v databáze vyhľadať meno učiteľa. Následne sa vykreslí meno na dvere namiesto ID dverí.

3.3.1 Menovky - serverová strana

Pri riešení získavania dát o miestnostiach sme narazili na viacero problémov. Prvým z nich je nedostatočná dokumentácia údajov o modeli, čo veľmi sťažovalo riešenie problémov, keďže posielané modely obsahujú veľké množstvo dát. Narazili sme na ďalší problém, a síce že momentálne nemáme k dispozícii údaje o tom, ktoré dvere vedú do ktorých miestností. Máme k dispozícii zoznam dverí s ich id, taktiež máme k dispozícii zoznam miestností, avšak prepojenie chýba. Tieto údaje bude nutné pridať do dát z modelu, aby bolo možné úlohu úspešne dokončiť.

3.3.2 Optimalizácia menoviek

Implementovanie menoviek na dvere budovy vyžaduje zmenu textúr dverí. Vytvorené riešenie ale prinieslo so sebou problém spomalenia celej prezentácie. Pohyb po budove bol značne spomalený. Po preskúmaní problému sme zistili, že slabým článkom nie je grafická karta, ale samotný javascript a engine cooperlicht. Problémom je aj veľkosť celkových textúr. Tým, že každé dvere majú svoju vlastnú textúru, sa veľkosť spotrebovanej pamäte razantne zvýšila.

Na vyriešenie problému sme použili skrývanie dverí na ostatných poschodiach a nahradzovanie atlasom. Táto optimalizácia vylepšila vykresľovanie a zvýšila celkový počet snímkou za sekundu. Ostal ale problém s veľkosťou zaberanej pamäte. Na zníženie veľkosti textúr je nutné zmenšiť plochu polygónu. To sme sa rozhodli implementovať pridaním nového polygónu v tvare štvorca

priamo na dvere. Tu sme narazili na problém s exportovaním modelu s upravenými dverami. 3D max studio skripty zhadzujú 3D modeler. Túto chybu sa nám zatiaľ nepodarilo odstrániť.

3.4 Vysúvací panel

Roly používateľov

Číslo US	Rola používateľa	Cieľ	Dôvod
1	Používateľ	Zobraziť informačný panel v režime fullscreen	Aby boli informácie o miestnostiach a ľuďoch dostupné aj v režime fullscreen
2	Používateľ	Zobrazovať na dverách menovky	Aby mal používateľ základné informácie o miestnosti pred tým, ako do nej vstúpi

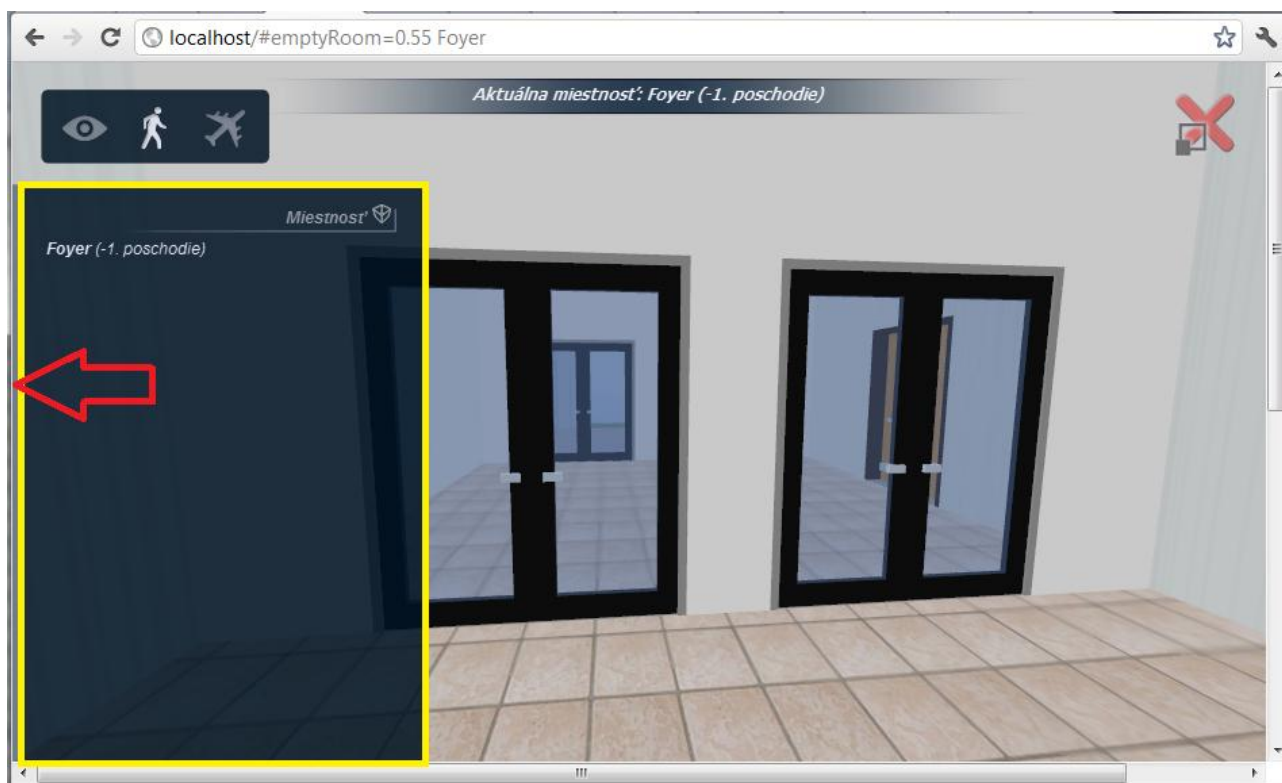
Úlohy k jednotlivým US

Číslo tasku	Popis tasku	Náročnosť	Priorita
1.1	Upraviť vysúvací panel	nízka	stredná
2.1	Menovky	stredná	stredná

Úlohou bolo vytvoriť vysúvací panel s informáciami pre režim fullscreen. Úloha si vyžiadala úpravy CSS, JavaScriptu a obrázkov.

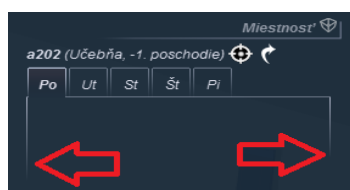
Konkrétne sme pridali element `<div>` s id `#fullscreenInfoPaneContainer` (viď žltý obdĺžnik na obr. 6), ktorý v režime fullscreen obsahuje samotný info panel, a element `<div>` s id `#toggleFullscreenInfoPane`, ktorý predstavuje prepínač na zobrazenie alebo skrytie informačného panelu (viď červená šípka na obr. 6 a 8). Uvedené elementy sú zobrazené len v režime fullscreen, inak sú skryté pomocou CSS.

Pri prepnutí do režimu fullscreen sa element informačného panelu s id `#info_pane` pomocou JavaScriptu dynamicky premiestni do vnútra elementu `#fullscreenInfoPaneContainer`, ktorý sa následne spolu s `#toggleFullscreenInfoPane` pomocou CSS zobrazí. Vysúvanie a zasúvanie panelu je riešené taktiež pomocou JavaScriptu, kedy pri kliknutí na prepínač `#toggleFullscreenInfoPane` sa `#fullscreenInfoPaneContainer` odsunie mimo obrazovky a po ďalšom kliknutí sa presunie nazad.

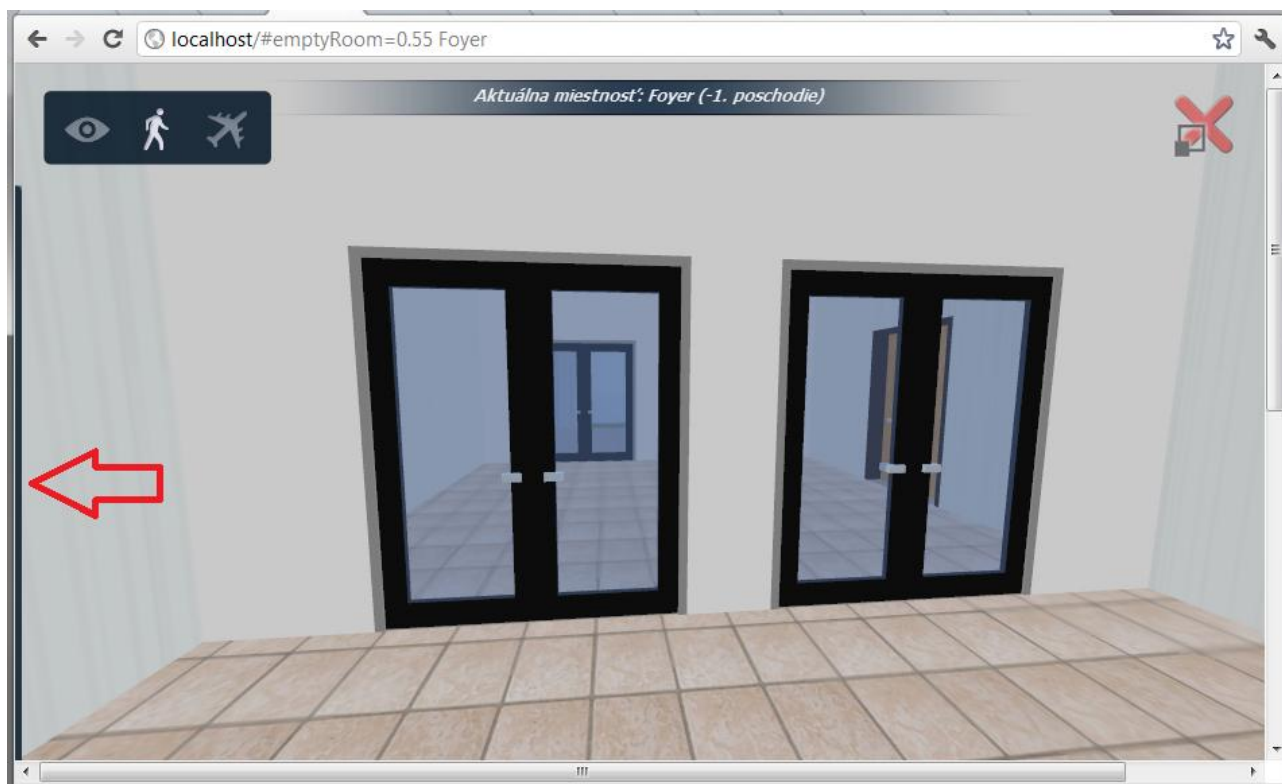


Obr. 6: Informačný panel je vysunutý

Bolo nutné vytvoriť obrázky pozadia oboch elementov a upraviť obrázok pozadia záložiek v informačnom paneli (viď červené šípky na obr. 3), keďže gradient prechádzal do čiernej farby namiesto priehľadnej.



Obr. 7: Záložky v informačnom paneli



Obr. 8: Informačný panel je zasunutý, zobrazí sa po kliknutí na prepínač, označený červenou šipkou

3.5 Celobrazovkový mód a navigačné prvky

Roly používateľov

Číslo US	Rola používateľa	Cieľ	Dôvod
1	Používateľ	Celobrazovkový mód	Aby bola vždy využitá celá plocha obrazovky a aby boli dobre rozložené ovládacie prvky
2	Používateľ	Získať údaje o tom, ktoré dvere spájajú ktoré miestnosti	Aby bolo možné zobrazit' na dverách menovky

Úlohy k jednotlivým US

Číslo tasku	Popis tasku	Náročnosť	Priorita
1.1	Zobrazovať iba celobrazovkový mód	nízka	stredná
1.2	Upraviť rozloženie ovládacích prvkov	nízka	stredná

2.1	Získať údaje o prepojení miestností z navigačného grafu	stredná	stredná
2.2	Presunúť načítavanie navigačného grafu pred načítavanie textúr	stredná	stredná

3.5.1 Zobrazovanie iba celoobrazovkového módu

Pôvodne existovali v aplikácii dva módy:

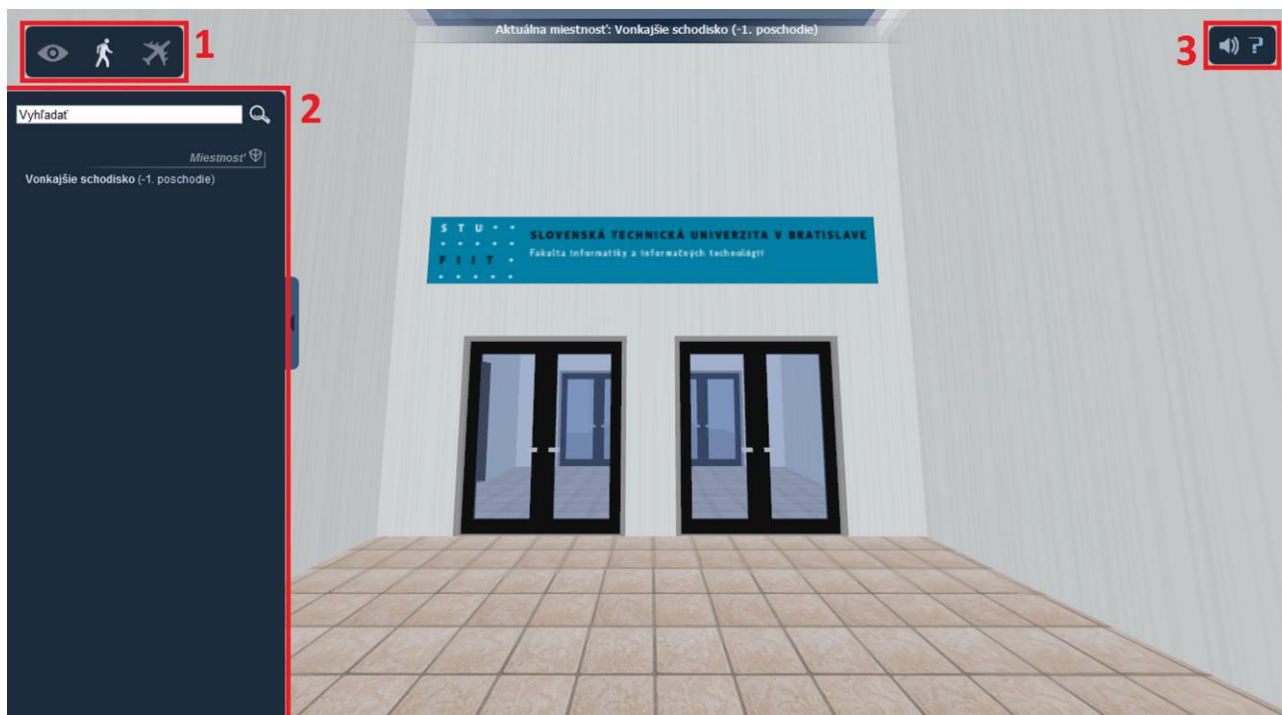
- normálny - zobrazovací panel mal pevnú veľkosť a bol umiestnený v strede stránky
- celoobrazovkový - zobrazovací panel zaberá celé okno

Rozhodli sme sa odstrániť normálny režim, keďže samotný zobrazovací panel zaberá len malú časť okna a zvyšný priestor bol takmer nevyužitý. Z HTML kódu boli odstránené všetky elementy, ktoré sa používali na zobrazovanie prvkov nesúvisiacich so zobrazovacím panelom. Ponechané boli panely #fullscreenPane (zobrazovací panel), #info_pane (zobrazuje informácie o miestostiach/ľuďoch), #searchBoxContainer (vyhľadávací formulár), #currentRoom (obsahuje názov aktuálnej miestnosti), #elevatorButtons (tlačidlá výťahu), #helpscreen (nápoveda), #fs_modes (tlačidlá na zmenu módu pohybu) a #fs_action_buttons (pomocné tlačidlá).

V CSS súboroch boli urovené drobné zmeny kvôli korektnému zobrazovaniu uvedených prvkov v celoobrazovkovom režime. Ďalšie úpravy sú popísané v nasledujúcich podkapitolách.

3.5.2 Úprava rozloženia ovládacích prvkov

Bolo zmenené rozloženie ovládacích prvkov na obrazovke (viď Obr. 9 a Obr. 10). Tlačidlá režimov pohybu (#fullscreenViewMode, #fullscreenWalkMode, #fullscreenFlyMode) boli presunuté k okraju bočného panelu a bolo upravené ich poradie (na obrázkoch označené ako 1). Bočný panel (#fullscreenInfoPaneContainer, na obrázkoch označený ako 2) bol rozťahnutý na celú výšku okna. Pomocné tlačidlá (#sound, #help, na obrázkoch označené ako 3) boli taktiež presunuté k okraju bočného panelu.



Obr. 9: Pôvodné rozloženie ovládacích prvkov

Bol pridaný nový panel s minimapou (#minimap, na obrázku označený ako 4). Minimapa je zatiaľ nefunkčná, avšak dizajn je pripravený do budúcnosti.



Obr. 10: Nové rozloženie ovládacích prvkov

3.5.3 Získanie údajov o prepojení miestností z navigačného grafu

Aby bolo možné zobraziť na dverách menovky, je potrebné získať informáciu o tom, ktoré miestnosti dané dvere spájajú. Tieto informácie je možné získať z navigačného grafu, ktorý sa načíta to poľa `self.navigationDisplayer.navigationGraph`. Keďže navigačný graf neobsahuje ID prvkov, ale farbu z kolíznej mapy, je ešte nutné načítať pre každé dvere farbu.

Údaje o miestnostiach sa teda získavajú nasledovne:

- pre všetky poschodia
 - pre všetky dvere
 - zisti z kolíznej mapy farbu dverí
 - zisti z navigačného grafu, ktoré miestnosti dvere spájajú
 - ulož túto informáciu do objektu dverí

3.5.4 Presun načítavania navigačného grafu pred načítavanie textúr

Aby bolo možné využiť informácie o prepojení miestností pri vytváraní textúr, je nutné presunúť načítavanie modelu poschodí (a tým pádom aj textúr) v čase až za vytvorenie navigačného grafu a tým pádom až za načítanie kolíznych máp poschodia. Presunuli sme načítavanie jednotlivých poschodí z konca metódy `vfiit.main()` do callbacku ako parametra vytvárania objektu triedy `vfiit.Engine`. Uvedený callback je zavolaný až po načítaní kolíznych máp poschodí a vytvorení navigačného grafu. Ešte bolo nutné oddeliť načítavanie kolíznych máp jednotlivých poschodí (metóda `vfiit.Engine.preloadFloor()`) od načítavania modelov poschodí (metóda `vfiit.Engine.loadFloor()`).

3.6 3D Model

Model minuloročného tímu je vytvorený vo veľmi dobrej kvalite. Tento model je vytvorený v softvéri 3ds Max.

Napriek vynikajúcej kvalite modelu bolo nájdených viacero chýb, medzi ktoré patria napríklad chyby textúr, chyby modelovania (posunuté a chýbajúce časti stien) a taktiež problémy vyskytujúce sa pri pohybe v aplikácií, ktoré spôsobujú najmä chyby výškových máp.

Úloha týkajúca sa 3D modelu je dlhodobá, lebo počas celého obdobia vývoju budú pravdepodobne nachádzané ďalšie chyby v modeli.

Roly používateľov

Číslo US	Rola používateľa	Cieľ	Dôvod
1	Používateľ	Upraviť existujúci model, aby obsahoval čo najmenej chýb	Aby mal používateľ cez web dostupný model bez chýb

Úlohy k jednotlivým US

Číslo úlohy	Popis úlohy	Náročnosť	Priorita
1.1	Import aktuálneho modelu do programu 3ds Max	nízka	vysoká
1.2	Export modelu z 3ds Max a jeho nasadenie v aplikácii	stredná	vysoká
1.3	Oprava nájdených chýb, aktualizácia modelu	podľa chyby	podľa chyby

Import aktuálneho modelu

Model minuloročného tímu sa podarilo importovať bez najmenších problémov. Časti modelu sú rozdelené do jednotlivých poschodí budovy, ktoré sú uložené ako samostatné 3ds Max súbory. Tieto súbory boli dostupné z repozitára minuloročného tímu.

Samotný import prebiehal ako jednoduché otvorenie týchto súborov v programe 3ds Max.

Export modelu a jeho nasadenie

Export modelu a jeho nasadenie prebieha nasledovne:

- export 3ds Max modelu vo formáte Collada (.dae)
- konverzia modelu vo formáte collada do formátu JavaScript Object Notation (JSON)
- nasadenie modelu v aplikácii

Export modelu vo formáte 3ds Max do formátu collada (.dae) sa vykonáva pomocou integrovaného pluginu (FBX exporter) priamo v programe 3ds Max. Model v tomto formáte je

uložený ako XML súbor.

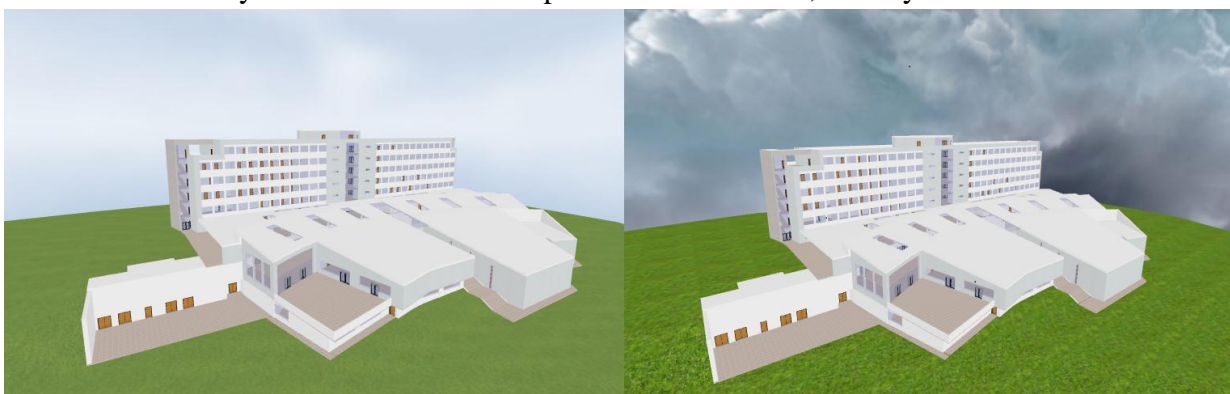
Collada súbor je následne potrebné konvertovať do súboru vo formáte JSON, ktorý predstavuje objekt v jazyku JavaScript. Na túto konverziu je používaný skript napísaný v jazyku Python, ktorý bol vytvorený minuloročným tímom. Skript bolo potrebné upraviť, keďže nebol kompatibilný s najnovšou verziou FBX exportéra. Ďalšia nepraktická vlastnosť skriptu bola, že finálny výstup (dáta vo formáte JSON) boli vypísané iba do konzoly. Preto jeho ďalšou úpravou bolo automatické vytvorenie JSON súboru a uloženie výstupu do neho.

Nasadenie modelu do aplikácie prebieha jednoduchým nahradením starého JSON súboru novým.

Aktualizácia textúr exteriéru modelu

Model budovy (3D scéna) je umiestnená vnútri v kocke (v tzv. skyboxe). Steny tejto kocky majú taktiež vlastnú textúru, ktorá sa prejavuje ako okolité pozadie 3D scény.

Originálne pozadie bolo v nízkom rozlíšení a pôsobilo príliš jednoducho. Toto pozadie bolo nahradené novovytvorenou textúrou, ktorá dodáva celej 3D scéne lepší vizuálny efekt. Taktiež bola nahradená nevýrazná textúra trávniku pred budovou novou, viac výraznou textúrou.



Obr. 11: Pohľad na budovu pred a po aktualizácii textúr

Chyba modelu – diera v schodisku

Pri hľadaní chýb modelu bola nájdená prvá chyba modelu v podobe diery v schodisku v miestnosti – aula. Táto chyba bola následne odstránená a bol exportovaný nový model bez tejto chyby. V modeli boli nájdené aj ďalšie chyby, ktoré zatiaľ ešte neboli opravené z dôvodu nutnosti štúdia programu 3ds Max a techník úpravy modelu. Tieto chyby budú opravené v najbližšej dobe.



Obr. 12: Chyba modelu pred a po oprave

3.6.1 Svetlá a tieňe v 3D modeli

Roly používateľ'ov

Číslo US	Rola používateľa	Cieľ	Dôvod
2	Používateľ	Upraviť existujúci model tak, aby jeho súčasťou bolo osvetlenie / tieňe	Aby bola aplikácia lákavejšia po grafickej stránke

Úlohy k jednotlivým US

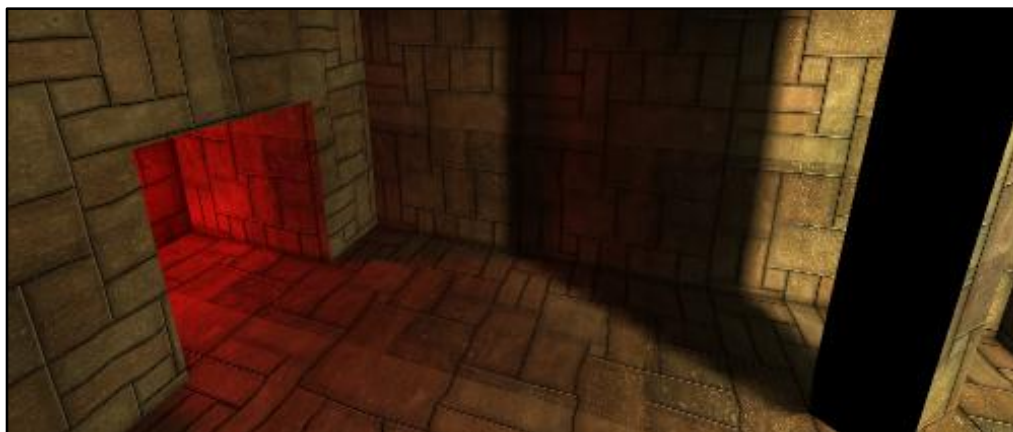
Číslo úlohy	Popis úlohy	Náročnosť	Priorita
2.1	Analýza možností pre vytvorenie svetiel a tieňov	stredná	vysoká
2.2	Aplikácia analyzovaných poznatkov a ich implementácia	vysoká	vysoká

Analýza možností pre vytvorenie svetiel a tieňov

Do modelu je možné pridať efekty osvetlenia dvoma spôsobmi:

- použitím dynamických svetiel (podpora priamo technológiou WebGL)
- mapovaním svetelných textúr (lightmapping)

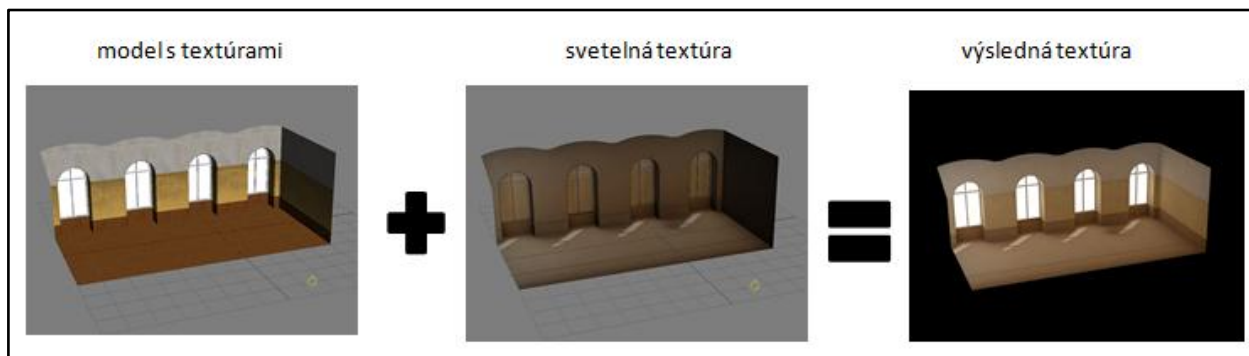
Dynamické svetlá sú pridané do modelu vo forme svetelných bodov. Tieto body vyžarujú svetlo, s ktorým priamo interaguje okolie a jednotlivé časti modelu. Dynamické svetlá sa rozdeľujú na bodové svetlá (vyžarujúce svetlo z jedného bodu do všetkých smerov) a na smerové svetlá (vyžarujú svetlo do určitého smeru).



Obr. 13: Model s dynamickým svetlom

Mapovanie svetelných textúr spočíva vo vytvorení textúr s efektom odrazeného svetla a ich

následné mapovanie priamo na originálne textúry modelu. Prekrytím týchto dvoch textúr vzniká dojem, že originálna textúra odráža určité množstvo svetla. Tieto textúry sú statické, sú používané pre statické objekty a neinteragujú priamo s okolím(vid' tabuľka 2).



Obr. 14: Aplikácia svetelných textúr

Tab. 2: Porovnanie metód osvetlenia

Metóda	Mapovanie svetelných textúr	Dynamické svetlá
Výhody	- minimálna náročnosť na hardvérové prostriedky	- dynamickosť (interakcia s okolím) - vizuálne pokročilejšie - jednoduchá implementácia (podpora knižnicou WebGL)
Nevýhody	- veľkosť textúr pre väčší model - vizuálne jednoduchšie - statickosť (žiadna interakcia s okolím)	- obrovská záťaž na prostriedky, hlavne pri väčšom počte dynamických svetiel

Napriek väčšiemu počtu výhod dynamického osvetlenia bude výhodnejšie použiť v projekte mapovanie svetelných textúr. Aplikácia nie je určená najmä pre vlastníkov modernejších grafických kariet, a preto je potrebné dbať na výkonnostné požiadavky. Použitím mapovania svetelných textúr nebudú tieto požiadavky extrémne zvýšené a zároveň sa dosiahne aj požadovaný efekt osvetlenia.

3.6.2 Výškové mapy

3.6.2.1 Neviditeľné zábradlia a chyby kolíznych máp

Vo vytvorenom modeli sa našla chyba v oblasti všetkých schodísk budovy, ktorá spôsobovala, že používateľ sa mohol pri prechode schodiskami ocitnúť o jedno poschodie vyššie alebo nižšie.

Táto chyba bola spôsobená chýbajúcimi časťami vo výškových mapách v oblasti schodísk.

Roly používateľov

Číslo US	Rola používateľa	Cieľ	Dôvod
1	Používateľ	Upraviť existujúci model, aby neobsahoval chyby schodísk a stien	Aby sa mohol používateľ pohybovať po budove bez problémov

Úlohy k jednotlivým US

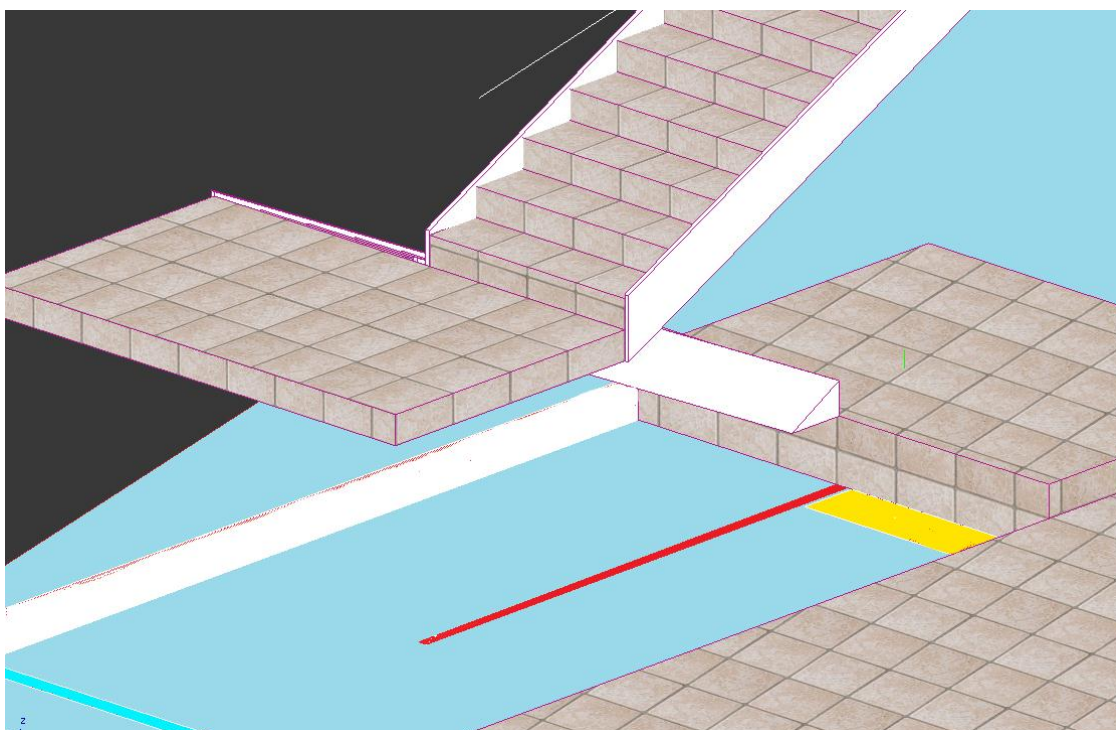
Číslo úlohy	Popis úlohy	Náročnosť	Priorita
1.1	Vytvorenie polygónov typu <i>Base</i>	stredná	vysoká
1.2	Vytvorenie výškovej (kolíznej) mapy z modelu	stredná	vysoká
1.3	Testovanie nových výškových máp a kolízií	stredná	stredná

3.6.2.2 Vytvorenie polygónov typu *Base*

Ako prvý krok bolo potrebné vytvoriť polygóny typu *Base*. Tieto polygóny sú pri exporte detegované ako základ kolíznych máp, ktoré sú nasledovne tvorené na ich základe. Jednotlivé polygóny, ktoré určujú kolízie musia byť tohto typu, inak sú export skriptami odignorované.

V modely sa nachádzajú dva typy schodísk. Sú to schody rovné (nachádzajúce sa pri hlavnom vchode) a schody otáčavé (nachádzajúce sa pri výťahoch).

Otáčavé schodiská v modely sú vytvorené z ľavej a pravej časti schodov. Na miesta medzi jednotlivými časťami schodiska boli vytvorené a pridané polygóny v tvare obdĺžnikov. Keďže boli tieto polygóny určené ako typ *Base*, nie sú v modely viditeľné. Tieto polygóny slúžia iba pri exporte kolíznych máp.



Obr. 15: polygón typu base (označený červenou farbou) umiestnený medzi ľavou a pravou časťou otáčavého schodiska

Ďalším problémom bolo prechádzanie cez okraje rovných schodísk nachádzajúcich sa priamo pri hlavnom vchode do budovy. Na okraje schodísk boli podobne pridané polygóny typu *Base*, ktoré zabránia prechodu cez tieto časti.

Pri pohybe po okraji rovného schodiska vedúceho do podzemia budovy (schodisko pri hlavnom vchode) bolo možné vidieť cez stenu. Na toto miesto bol taktiež pridaný polygón typu *Base*, ktorý zabráni pohybu v blízkosti tejto steny.

Na všetkých poschodiach sa nachádzajú pri výťahoch sklenené okná siahajúce od zeme až po strop. Pri pohybe veľmi blízko týchto okien občas nastala situácia, že používateľ z okna vypadol. Na týchto miestach sa nachádzali polygóny typu *Base*, avšak boli príliš úzke a pri intenzívnom pohybe sa mohlo stať, že kolízia nebola zachytená a používateľ prešiel cez toto okno. Rozšírením týchto polygónov bol tento problém odstránený na všetkých poschodiach.

3.6.2.3 Vytvorenie výškovej (kolíznej) mapy modelu

Samotné výškové mapy sú generované priamo z modelu v programe 3DS Max. Najprv sú z programu 3DS Max vyexportované mapy výšok (pomocou farebných škál) a mapy dverí a stien. Tieto dve mapy sú následne spojené pomocou python skriptu a je vytvorená finálna výšková (kolízna mapa).

Nasadenie máp do aplikácie sa vykoná obyčajným prekopírovaním súboru do príslušného priečinku daného poschodia.

Podrobnejší postup procesu exportu výškových máp sa nachádza v dokumentácii minuloročného tímu.

3.6.2.4 Testovanie nových výškových máp a kolízií

Testovanie bolo vykonané vo forme akceptačných testov (viď tabuľky 3, 4, 5).

Tab. 3: akceptačný test č. 1

ID	1	Názov	Neviditeľné zábradlia otáčavých schodísk	
Účel	Testovanie kolíznej mapy			
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia	
1	Presunieme sa k ľubovoľnému otáčavému schodisku v aplikácii	Žiadna	Zhoduje sa s očakávanou reakciou	
2	Prejdeme medzi poschodiami stredom otáčavého schodiska	Pohyb je zablokovaný (nie je možné prejsť)	Zhoduje sa s očakávanou reakciou	

Tab. 4: akceptačný test č. 2

ID	2	Názov	Kolízie okien	
Účel	Testovanie kolíznej mapy			
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia	
1	Presunieme sa k ľubovoľnému oknu veľkému od zeme až po strop	Žiadna	Zhoduje sa s očakávanou reakciou	
2	Snažíme sa prejsť cez okno	Pohyb je zablokovaný (nie je možné prejsť)	Zhoduje sa s očakávanou reakciou	

Tab. 5: akceptačný test č. 3

ID	2	Názov	Videnie cez steny	
Účel	Testovanie kolíznej mapy			
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia	
1	Presunieme sa k rovnému schodisku pri hlavnom vchode budovy vedúcemu do podzemia	Žiadna	Zhoduje sa s očakávanou reakciou	
2	Pri pohybe dole schodmi sa tlačíme na pravú stenu schodiska	Pri pohybe nie je možné vidieť cez stenu modelu	Zhoduje sa s očakávanou reakciou	

3.7 WebGL

Použitý framework copperlicht považujeme za vhodné zvolený:

- Voľne šíriteľný
- Výrazne uľahčuje vytváranie kódu VirtFIIT
- Podporuje množstvo 3D formátov: 3ds, obj, x, lwo, b3d, csm, dae, dmf, oct, irrmesh, ms3d, my3D, mesh, lmts, bsp, md2, stl a ďalšie
- Prehľadný, jednoduchý SDK
- Plne využíva najnovšie WebGL 1.0
- Veľmi rýchly

3.7.1 Aktualizácia WebGL

Minuloročná verzia používa copperlicht vo verzii 1.3.5, najnovšia verzia je 1.4.1

Obsahuje mnohé vylepšenia ako napríklad:

- Podporu dynamického osvetlenia
- Obojstranné materiály
- Podpora IE pomocou IEWebGL pluginu (overiť)
- Podpora kamery typu tretia osoba
- Optimalizované vykresľovanie
- Odstránené mnohé chyby, ako napr.:
 - vyťažovanie CPU
 - nesprávna poloha myši
 - chybné propagovanie kláves

Preto bola táto knižnica aktualizovaná v repozitári a ten nahraný do produkčnej verzie.

3.7.2 Antialiasing vo WebGL

Podľa špecifikácií WebGL je antialiasing štandardne zapnutý pokiaľ ho podporuje prehliadač. Firefox zatiaľ antialiasing nepodporuje. Od verzie 10 by ho ale mal podporovať. Chrome antialiasing podporuje.

3.8 Knižnice pre websockety

- GNU WebSocket4J – GPL2 <http://maarons.alwaysdata.net/software/GNU/WebSocket4J/>
- jWebSocket – LGPL www.jwebsocket.org/

Primárne by som sa zamerával na jWebSocket z dôvodu lepšej dokumentácie a množstva podporovaných aditívnych funkcionalít a voľnejšej licencie. WebSocket4J je momentálne aj zastavený vývoj pokiaľ nebudú websockety oficiálne schválené.

3.9 Testovanie importéra

Inštalácia aplikácie podľa návodu v dokumentácii minuloročného tímu bola bezproblémová.

Importér je funkčný, správne importoval údaje z AIS-u. Testovali sme nasledovné časti:

- Databáza
 - *Nainštalovať databázu (vykonať CREATE tabuliek)*
- *Import z AIS*
 - *Importovať údaje (zahŕňa import zamestnancov, miestností a predmetov)*

- *Importovať zamestnancov z AIS*
- *Importovať miestnosti z AIS*
- *Importovať predmety z AIS*
- *Import z rozvrhu*
- *Importovať osoby z rozvrhu fakulty*
- *Importovať miestnosti z rozvrhu fakulty*
- *Importovať predmety z rozvrhu fakulty*
- *Importovať rozvrh fakulty*

Všetky testované importy prebehli v poriadku, s výnimkou importov v časti *Import z rozvrhu*, kedy nám prehliadač pri spustení v časti *Import z rozvrhu* vrátil prázdnu stránku. V Apache error logu sa pre daný čas nachádzal iba nasledovný riadok:

```
zend_mm_heap corrupted
```

Po krátkom hľadaní na internete sme prišli na to, že daný stav bol spôsobený knižnicou Simple HTML DOM Parser pre PHP:

<http://stackoverflow.com/questions/3830187/zend-mm-heap-error-with-simple-html-dom>

Riešením bolo pridať premennú `USE_ZEND_ALLOC` s hodnotou 0 medzi premenné prostredia Apache, napríklad pridaním nasledovného riadku do súboru `.htaccess` nachádzajúceho sa vo webroote aplikácie:

```
SetEnv USE_ZEND_ALLOC=0
```

Následne už importy prebehli úspešne všetky.

4 Nájdene chyby

4.1 Výt'ah

Keďže výt'ah sa pohybuje medzi poschodiami, najviac sa tu prejavujú chyby v kolízii a v prepojení poschodí. Je možné napríklad predčasne vystúpiť (obr. 1), vypadnúť do šachty (obr. 2) či sa zaseknúť niekde mimo priestoru (obr. 3).

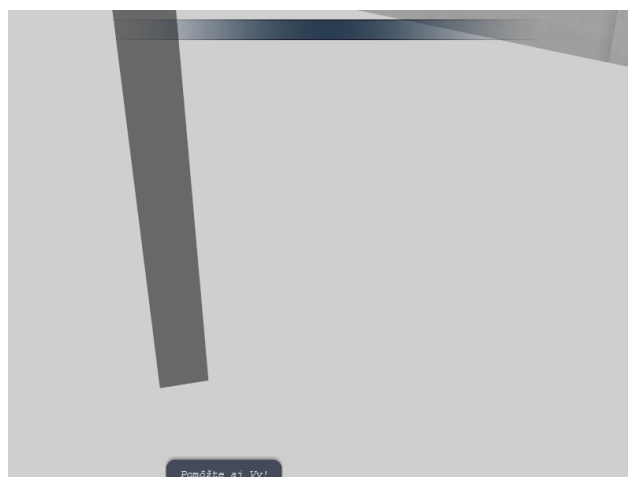
Ide o veľmi komplexné chyby, na ktorých odstránenie je potrebné prepísať kód pohybu osoby, výt'ahu, detekciu kolízie a pod., nebolo preto možné behom jedného šprintu chybu odstrániť. Chyba nebola spôsobená enginom `copperlicht`, nakoľko je prítomná aj po aktualizácii tohto enginu na najnovšiu verziu, tá však obsahuje prepracovanejšiu detekciu kolízie, čo pri reimplementácii spomínaných častí kódu môže pomôcť chybu odstrániť.



Obr. 16: Predčasné vystúpenie z výťahu



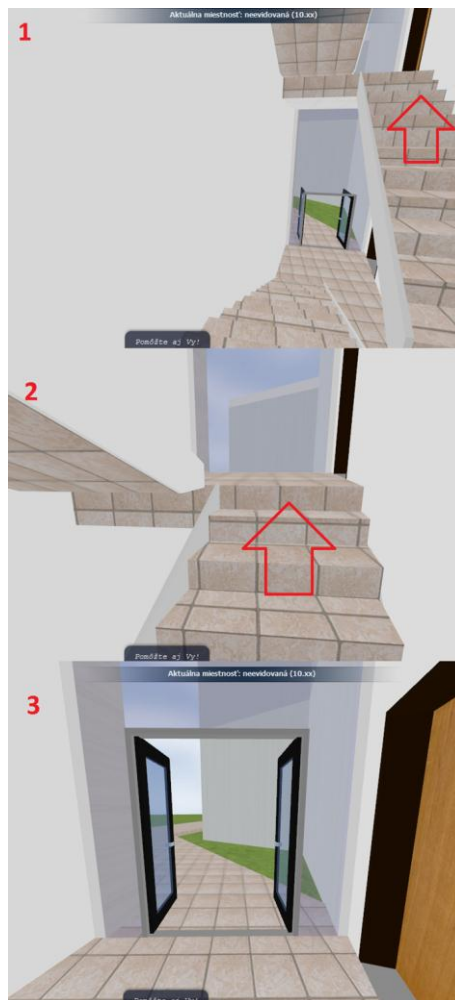
Obr. 17: Vypadnutie do šachty



Obr. 18: Zaseknutie sa (nie je možné sa pohnúť)

4.2 Déjà vu

Ako bolo spomenuté vyššie, niekedy nie sú poschodia správne prepojené a je možné sa napr. opätovne objaviť na tom istom poschodí, ako napr. tu na bočnom schodisku (obr. 4).



Obr. 19: Schodiskové deja vu

4.3 Zvuky chôdze a iné

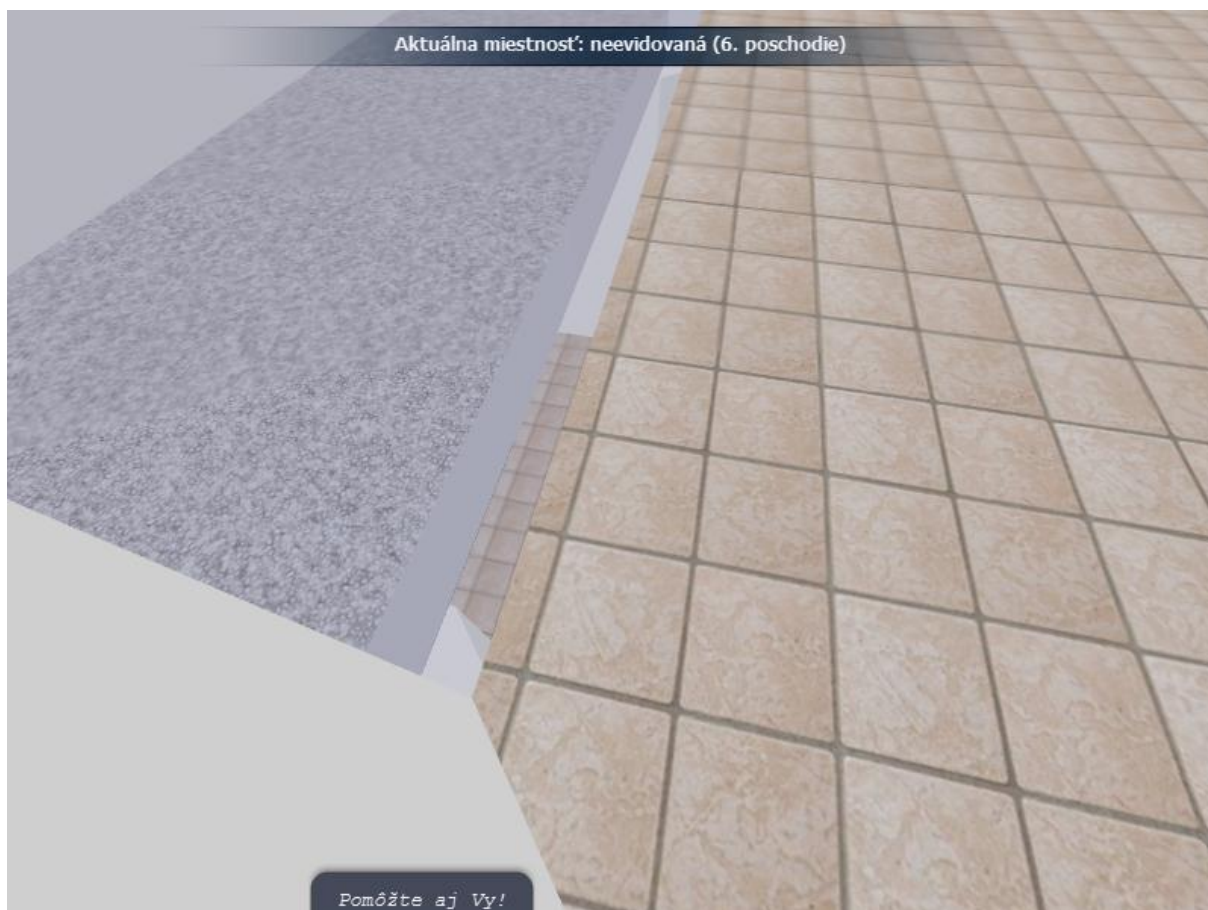
Chybné je i zapracovanie zvukov do aplikácie. Pri zaseknutí výťahu sa zvuk výťahu ďalej prehráva. Horšie je na tom však zvuk chôdze: tempo krokov vôbec nezodpovedá rýchlosti pohybu a nemení sa ani pri behu (klávesou shift). I keď ide o slabší nedostatok, treba naň myslieť.

4.4 Dvere

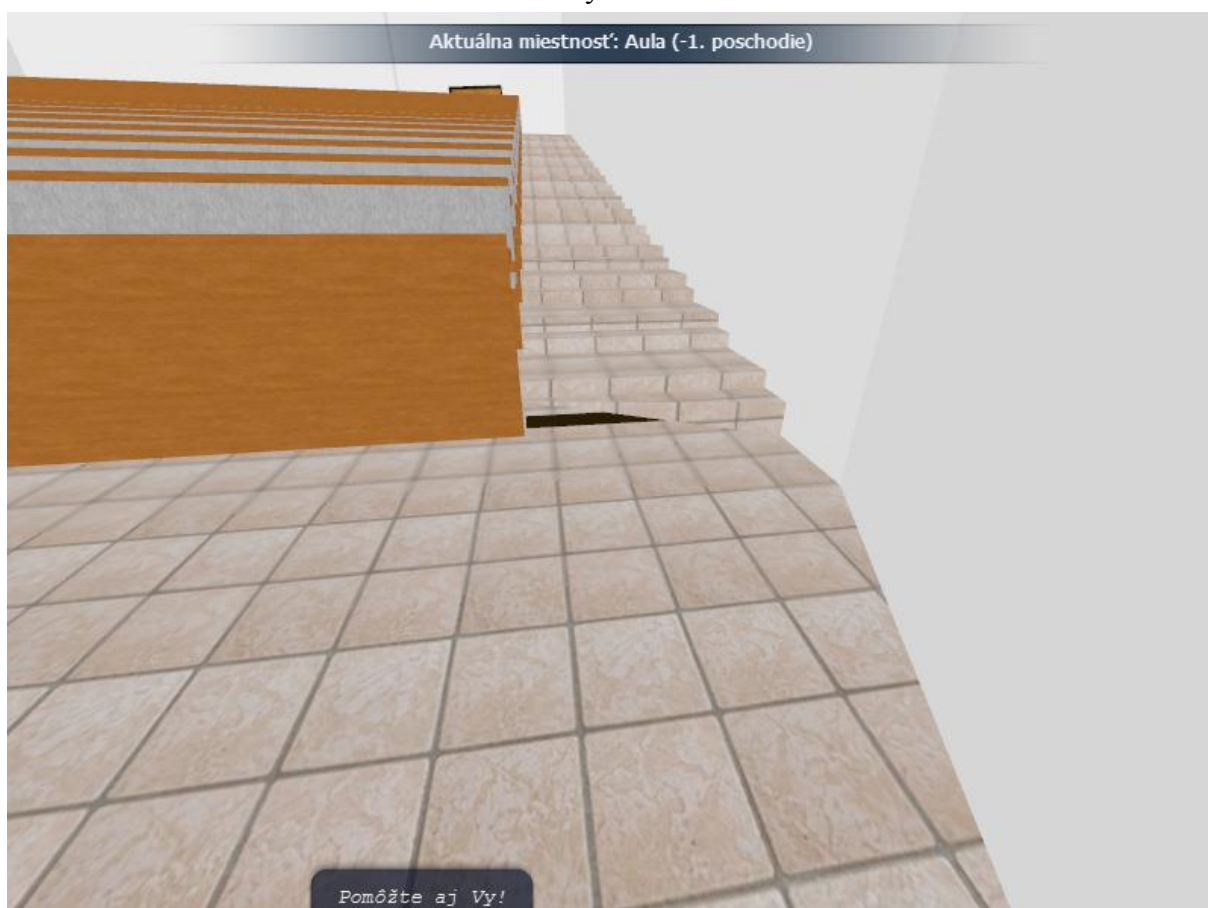
Dvere sa zatvoria, až keď sa iné dvere otvoria. To nie je vhodné a môže to byť veľmi chaotické keď, bude implementovaná real-time interakcia viacerých používateľov.

4.5 Chyby modelu

Chýb modelu je niekoľko, najčastejšie sa vyskytujú pri skladaní jednotlivých častí budovy (či už stien alebo jednotlivých poschodí), sú tu však aj iné chyby, spôsobené nedokončením niektorej steny objektu. Príklady chýb:



Obr. 20: Chyba modelu č. 1



Obr. 21: Chyba modelu č. 2



Obr. 22: Chyba modelu č. 3



Obr. 23: Chyba modelu č. 4

4.6 Webové rozhranie

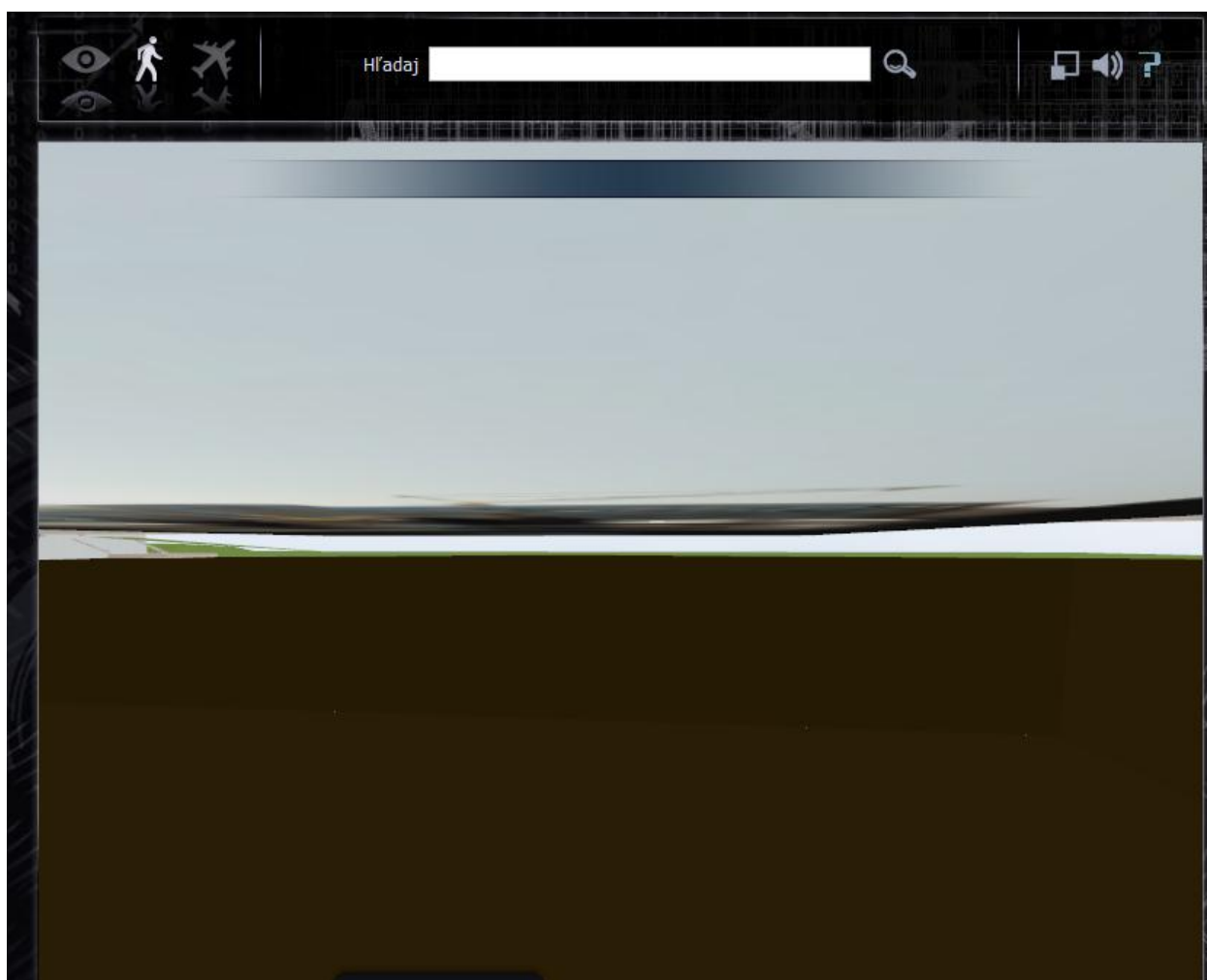
Credits idú zbytočne pomaly. Keďže sa tam nenachádza mnoho funkcií ani iných možností, na webovom rozhraní sme neobjavili do očí bijúce chyby.



Obr. 24: Hrubé okno vidno až na ďalšie poschodie



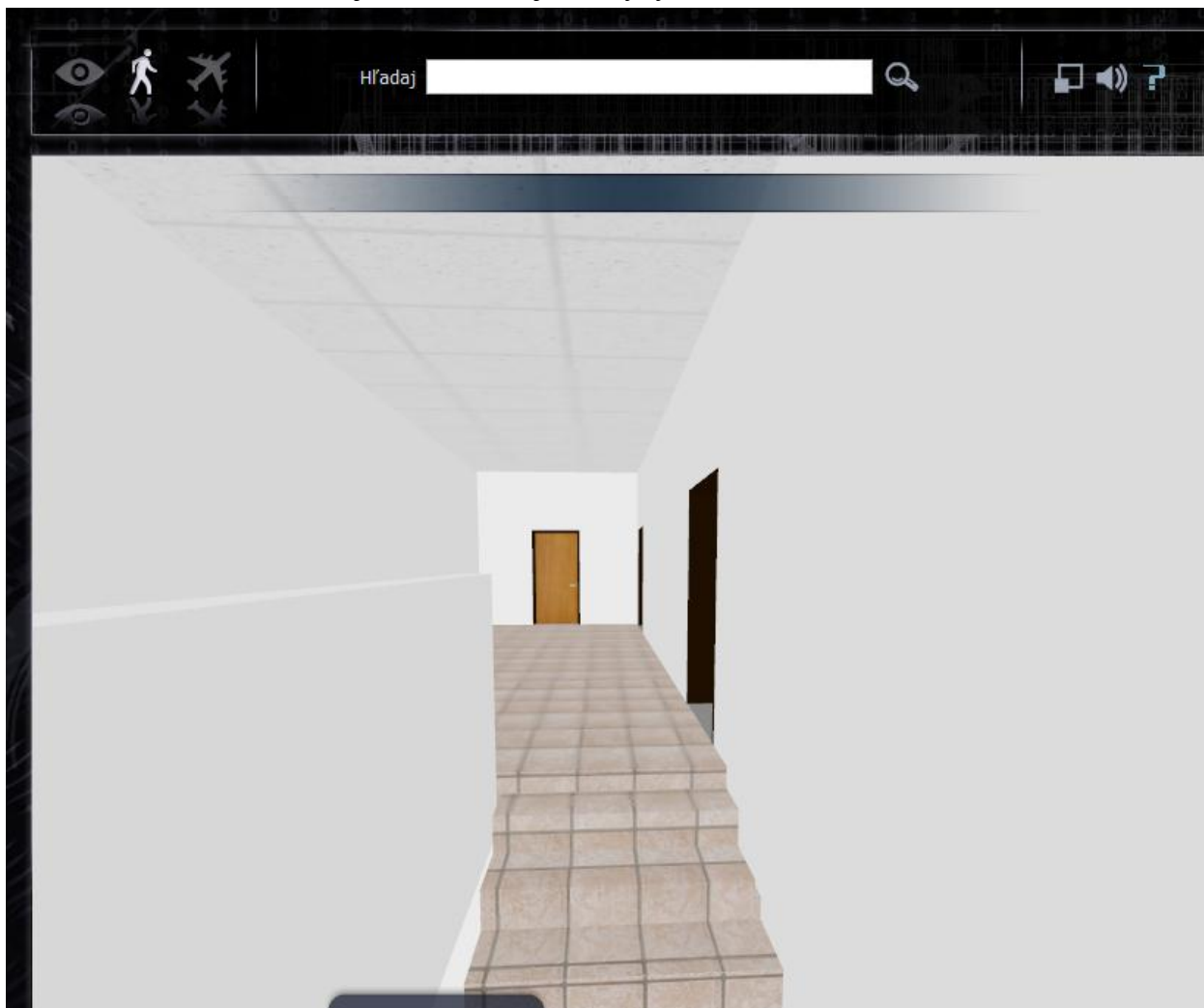
Obr. 25: Nedoliehajúca stena



Obr. 26: Prepadnutie pod schodmi v režime chôdze

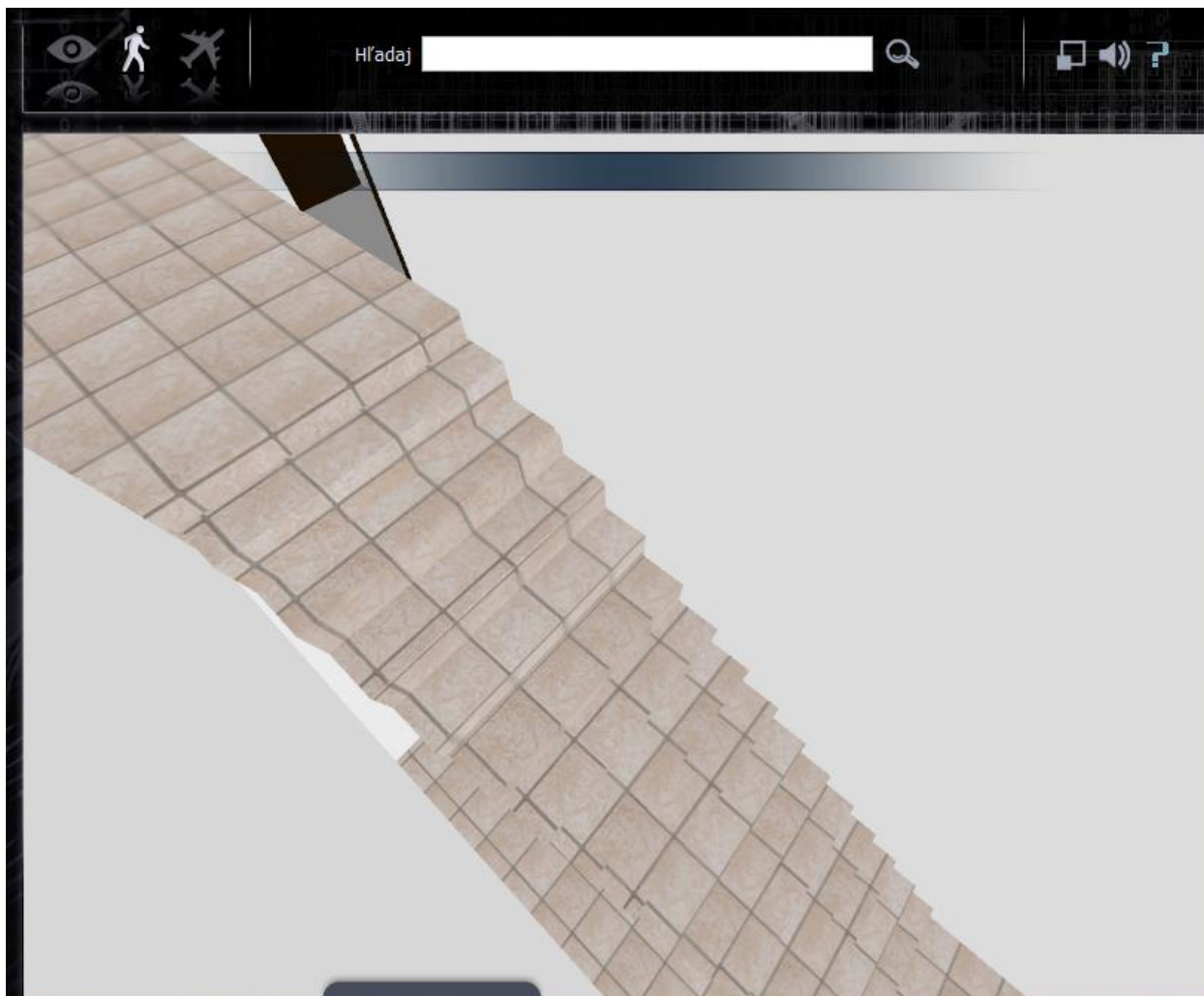
4.7 Rozhranie

Credits idú zbytočne pomaly. Keďže sa tam nenachádza mnoho funkcií ani iných možností, na webovom rozhraní sme neobjavili do očí bijúce chyby.



Obr. 27: Levitovanie

V režime chôdze levitácia nad schodmi. Obe predchádzajúce chyby nastali pri väčšej záťaži procesora iba raz za čas pri opakovanom prechádzaní jedného úseku.



Obr. 28: Levitácia

4.8 Iné drobné chyby

Bug #1:

Postup:

Pri chôdzi po schodoch zo spodného poschodia nahor sa prejde niekedy cez schody a zostane dolu.

Po prejdení schodami sa prázdny priestor (chodba) správa ako stena – nedá sa cez chodbu prejsť.

Bug #2:

Postup:

Po priblížení sa k stene na schodoch nadol zo spodného poschodia je možné vidieť za stenu.

Bug #3:

Postup:

Po prejdení prvého schodiska sa dá prejsť cez zábradlie. Následne je možné lietat' (akoby sa dalo chodiť po neexistujúcom podklade).

Bug #4: Prechádzanie cez otáčacie dvere

Postup:

1. prísť ku dverám
2. ísť cez dvere

Očakávané: nie je možné prejsť

Reálne: dá sa prejsť

Bug #5: Prechádzanie cez otvorené dvere

Postup:

- 1) prísť ku dverám, otvoria sa
- 2) snažiť sa prejsť cez krídlo otvorených dverí

Očakávané: nie je možné prejsť

Reálne: dá sa prejsť

Bug #6: Spodok schodiska

Postup:

- 1) prísť na spodok schodiska
- 2) motať sa na opačnej strane ako je začiatok schodiska, pri stene a smerom k zadnej stene

Očakávané: dá sa ísť pod schodisko

Reálne: nedá sa ísť pod schodisko, občas to hodí na vyššie poschodie

Bug #7: Prechádzanie medzi úrovňami schodiska

Postup:

1. prísť na ľubovoľné schodisko, zhruba do stredu schodov
2. ísť do boku do vedľajšieho schodiska

Očakávané: nedá sa prejsť, prípadne padnem na vedľajšie schodisko

Reálne: preskakujem z jedného na druhé, napriek tomu, že tam je veľký výškový rozdiel

Bug #8: Vidieť cez stenu

Postup:

1. prísť úplne naľavo od hlavného vchodu, kde sú dvere zasadené do zeme
2. šmýkať sa po stene

Očakávané: nič zaujímavé

Reálne: vidieť cez stenu do miestností

Bug #9: Prechádzanie cez sklo

Postup:

1. prísť do miestnosti s výťahmi (ľubovoľné poschodie okrem prízemie)
2. prísť úplne do zlomu presklenej steny medzi blokmi výťahov

Očakávané: nič zaujímavé

Reálne: ocitnem sa tesne za sklom na vonkajšej strane budovy, niekedy sa dá prejsť aj preč od budovy (vo vzduchu)

Bug #10: Vonkajšia časť budovy

Vonkajšia časť budovy sa dala prezerat' viacmenej iba pomocou režimu voľného letu a režimu náhľadu. Pre režim voľného letu neexistujú zábrany v podobe kolízií. V danom režime sme nenašli žiadne buggy.

Režim náhľadu je v podstate iba zobrazovanie jednotlivých poschodí z výšky bez väčších možnosti zmenit' pozíciu, možné je približovat' a menit' uhol. V danom režime sa vonkajší plášť dá prezerat' iba minimálne a neodhalili sme žiadne chyby.

Vonkajšia časť obsahuje chodník na ktorom sa nenachádzajú žiadne anomálie.

4.9 Oprava bugov – prepojenie poschodí

Z tejto úlohy som stihol vzhľadom na rozsiahlosť ostatných úloh iba spraviť analýzu. Bug naozaj v aplikácii existuje a najviac sa prejavuje na schodisku, ktoré nemá zábradlie. Riešením by mohlo byť pridaním do modelu zábradlia alebo neviditeľnú (priesvitnú) stenu cez ktorú by sa nedalo prechádzat', čím by blokovala pohyb zo schodiska.

5 Mobilná aplikácia - PhoneGap

V letnom semestri po vykonaných opravách a rozšíreniach desktopovej verzie projektu VirtualFIIT sme sa rozhodli vytvoriť mobilnú aplikáciu vyvíjanú na báze analyzovanej technológie PhoneGap zo zimného semestra (viď kapitola 3.1). Po dôslednej analýze sa nám podarilo vytvoriť zoznam jednotlivých features (backlog), na ktorých vývoji budeme pracovať v jednotlivých šprintoch.

- balíčkovanie (pre prenositeľnosť)
- Zobrazenie celoobrazovkovej mapy
 - farebný obrázok
 - označenia (čísla) miestností na mape
- Zobrazenie trasy na mape
- Vyhľadávanie osoby
 - zobrazenie výsledku vyhľadávania informácií
- Vyhľadávanie rozvrhu človeka
 - zobrazenie výsledku vyhľadávania informácií
- Načítanie pozície z QR kódu
 - alternatívne načítanie číslom
- Navigácia na zvolené miesto
 - zadanie cieľa
- Prihlásenie do AIS
- Zobrazenie osobného rozvrhu prihláseného študenta
 - potrebné prihlásenie do AIS
- Zobrazenie jedálneho lístka
- Zobrazenie odchodov MHD od budovy
- Používanie cache
 - vo vyhľadávaní
 - pre osobný rozvrh
 - pre rozvrhy vyhľadávaných ľudí
 - pre umiestnenie vyhľadávaných ľudí
 - pre MHD

V rámci zadania boli taktiež detekované aj nasledujúce user stories:

USC č. 1:

Rola	Cieľ	Dôvod
Používateľ	Zobrazenie trasy na mape	Umožní prehľadnú navigáciu.
Ďalšie detaily	Výrazná, ľahko viditeľná čiara. Symboly, napr. pri schodisku, výtahu.	
Opis testov	Používateľ zadá svoju polohu – skenovaním QR kódu alebo zadaním čísla miestnosti, zadá cieľovú polohu (číslo miestnosti). Zobrazená mapa bude obsahovať čiaru (a prípadné značky), ktorá bude viesť od aktuálnej polohy po cieľ.	

USC č. 2:

Rola	Cieľ	Dôvod
Používateľ	Navigácia na zvolené miesto	Umožní prehľadnú navigáciu.
Ďalšie detaily	Voľba aktuálnej polohy – QR kód alebo číslo miestnosti. Výrazná, ľahko viditeľná čiara. Symboly, napr. pri schodisku, výtahu.	
Opis testov	Používateľ zadá svoju polohu – skenovaním QR kódu alebo zadaním čísla miestnosti, zadá cieľovú polohu (číslo miestnosti). Zobrazená mapa bude obsahovať čiaru (a prípadné značky), ktorá bude viesť od aktuálnej polohy po cieľ.	

USC č. 3:

Rola	Cieľ	Dôvod
Používateľ	Používanie cache (pre rozvrhy, pre MHD)	Používateľ chce, aby aplikácia fungovala aj bez internetového pripojenia v režime offline. Používateľ chce, aby program v čo najväčšej miere šetril množstvo prijímaných dát cez internet.
Ďalšie detaily		
Opis testov	Aplikácia funguje v režime offline. Aplikácia zobrazuje aktuálne dáta rozvrhov a MHD aj v režime offline.	

USC č. 4:

Rola	Cieľ	Dôvod
Používateľ	Balíčkovanie	Zákazník chce, aby sa v aplikácii dali vymeniť podklady máp a miestností vo forme dátových balíkov pre iné budovy. Zákazník chce, aby bolo túto výmenu vykonať bez zásahu do zdrojových kódov.
Ďalšie detaily		
Opis testov	Všetky funkcie aplikácie správne fungujú aj so zmenenými podkladmi pre inú budovu.	

USC č. 5:

Rola	Cieľ	Dôvod
Používateľ	Vyhľadávanie rozvrhu človeka	Nájsť miestnosť kde je prednáška
Ďalšie detaily	<ul style="list-style-type: none"> - používateľ si chce nájsť miestnosť kde ma prednášku - neukážeme mu celý zoznam predmetov, ktoré existujú na škole ale iba jeho rozvrh - rozvrh treba načítať z AIS - je potrebné login na AIS. - získanie rozvrhu prebehne automaticky - zobrazenie aktuálneho rozvrhu vzhľadom na aktuálny čas 	
Opis testov	- test, či sa zobrazí rozvrh	

USC č. 6:

Rola	Cieľ	Dôvod
Používateľ	Vyhľadávanie osoby	Umožní získanie informácií o osobe.
Ďalšie detaily	<p>Možnosť vyhľadania osoby zadaním mena. Systém zobrazí základné údaje o hľadanej osobe (meno, sídlo, kontaktné informácie), predmety, ktoré vyučuje a rozvrh. Na poprednom mieste bude zobrazené miesto, kde by sa podľa rozvrhu osoba v danom čase (prípadne v najbližšom čase) mala nachádzať.</p>	
Opis testov	Používateľ zadá meno osoby a systém mu zobrazí uvedené informácie.	

USC č. 7:

Rola	Cieľ	Dôvod
Používateľ	Zobrazenie osobného rozvrhu	Umožní zobrazit' osobný rozvrh.
Ďalšie detaily	Ak je používateľ prihlásený do systému AIS (pozri Prihlásenie do AIS), systém mu zobrazí jeho osobný rozvrh, kde bude zvýraznená aktuálna časť rozvrhu.	
Opis testov	Používateľovi sa zobrazí jeho osobný rozvrh.	

USC č. 8:

Rola	Cieľ	Dôvod
Používateľ	Zobrazenie celoobrazovkovej mapy	Lepšia viditeľnosť na malom displeji
Ďalšie detaily	Mapa bude zobrazená ako farebný obrázok. <ul style="list-style-type: none"> • Miestností na mape budú označené aj svojim číslom. • Funkcionalita je plne dostupná aj v režime offline. 	
Opis testov	Úspech: <ul style="list-style-type: none"> • Zobrazila sa farebná mapa na celú obrazovku. • Na mape sú viditeľné čísla miestností. • Horeuvedené 2 body fungujú v režime offline. 	

USC č. 9:

Rola	Cieľ	Dôvod
Používateľ	Zobrazenie jedálneho lístka	Jednoduchý prístup k informáciám z jedného miesta
Ďalšie detaily	Prístup k jedálnemu lístku z jedného miesta (z ktorého sa prístupuje k iným informáciám). <ul style="list-style-type: none"> • Funkcionalita je dostupná iba v režime online. 	
Opis testov	Úspech: <ul style="list-style-type: none"> • Zobrazil sa jedálny lístok príslušného zariadenia. Chybové hlásenia: <ul style="list-style-type: none"> • „Nie je k dispozícii dátové“ 	

5.1 Editor navigačného grafu

Pre potreby zobrazenia trasy pri navigácii je potrebné vytvoriť navigačný graf ktorý bude určovať trasy po ktorých sa dá pohybovať po budove. V prototypy navigácie sme tento graf vytvorili ručne. Toto bolo aj pre jedno poschodie veľmi zdĺhavé. Preto vznikla potreba vytvoriť nástroj pomocou ktorého sa bude dať jednoducho a rýchlo tento navigačný graf vytvoriť.

5.1.1 Analýza požiadaviek

Pri navigácii sa bude kresliť nad mapu do HTML5 canvasu trasa vo forme čiary. Je teda potrebné mať k dispozícii body a prepojenia medzi týmito bodmi. Je taktiež potrebné rozlíšiť typ bodu teda či daný bod reprezentuje miestnosť, schodisko, výťah alebo prepojenie. Miestnosti budú totiž väčšinou koncové body grafu. Trasy medzi miestnosťami budú tvorené prepojeniami z bodov. Všetky body budú samozrejme tvorené hranami.

Hranám sa bude dať ručne nastaviť váha aby sa dali ošetriť prípadné rôzne špeciálne prípady. Jednotlivé body grafu bude treba odlišiť a preto sa každému bodu priradiť menovku. Okrem menovky budú jednotlivé typy bodov odlišené farbou.

Najprv sa vytvoria navigačné grafy v rámci jedného poschodia. Následne sa musia vytvoriť schodiská a výťahy medzi poschodiami. Teda spojiť jednotlivé body reprezentujúce schodiská a výťahy do reťaze.

5.1.2 Návrh

Funkcionalita bude rozdelená medzi nasledovné triedy.

Floor – obsahuje všetky body, hrany a polygóny jedného poschodia.

NavigationGraphView – trieda obsluhujúca takmer všetku funkcionality vzťahujúcu sa k grafu. Zabezpečuje posúvanie pohľadu, pridávanie bodov, polygónov a manažment poschodí.

MainWindow – trieda hlavného okna. Obsluhuje a vytvára menu aplikácie, panel nástrojov a samotný graf.

ElevatorChainDialog – trieda dialógu kde sa zreťazujú body reprezentujúce výťahy a schodiská.

GraphNode – bod v grafe. Môže byť 4 rôzne typy, miestnosť, schodisko, výťah a prepojovací. Každý bod bude mať odkazy na hrany ktoré z neho vedú aby mohol aktualizovať v prípade zmeny pozície bodu.

GraphEdge – predstavuje hranu medzi bodmi grafu. Obsahuje odkazy na dva body ktoré spája.

GraphPolygon – polygón reprezentujúci ohraničenie miestnosti.

GraphLabel – pomocná trieda zobrazujúca menovku pri bode alebo polygóne.

GraphXMLReader/Writer – dve triedy ktoré zabezpečujú načítanie a uloženie grafu do XML formátu.

5.1.3 Implementácia

Editor je implementovaný v jazyku C++ za použitia GUI frameworku Qt. Na vykresľovanie sa používajú hlavne komponenty QGraphicsSceneView a QGraphicsScene.

5.2 Ohraničenie miestností

Jednou z vlastností programu bude zobrazenie podrobností o miestnosti. Jednou z možností ako ich zobrazíť bude aj kliknutie na mapu kedy si používateľ kliknutím na miestnosť vyberie danú miestnosť. Program následne zobrazí informácie o danej miestnosti.

5.2.1 Analýza

Samotná mapa je forme bitmapového obrázka. Po kliknutí na mapu je potrebné zistiť na ktorú miestnosť používateľ vybral. Sú možné tri prístupy. Použitie už existujúcich bitmap ktoré kódujú číslo miestnosti do farby alebo definovať ohraničenie jednotlivých miestností formou polygónu a následne analyticky vypočítať v ktorom polygóne leží klik. Nevýhodou bitmap je ich dátová veľkosť. Preto sme sa rozhodli pre druhú možnosť analyticky zisťovať v polygónoch.

5.2.2 Editor

Do editora bude treba pridať možnosť taktiež definovať polygóny. Jednotlivé polygóny sa budú definovať klikaním na miesta kde majú ležať vrcholy. Definovanie polygónu sa ukončí dvojklikom.

5.2.3 Program

Na zistenie či sa nachádza klik v polygóne sme použili metódu počítania prieniku polpriamky s hranami polygónu. Táto jednoduchá metóda je založená na fakte že ak je bod vnútri akéhokoľvek polygónu tak počet prienikov polpriamky vedúcej tohto bodu a okrajov polygónu je nepárny. Je možná taktiež optimalizácia hľadania týchto prienikov pri vhodne zvolenom smere polpriamky. Ak smeruje rovnobežne s niektorou z osí dá sa vylúčiť prienik jednoduchým porovnaním hodnôt na bodov na tejto osi. Ak je napríklad polpriamka rovnobežná s osou X tak stačí porovnať či je x-ová súradnica v rozmedzí súradníc bodov ktoré tvoria hranu polygónu. Ak leží mimo tak prienik nemôže nastať.

5.3 Zobrazenie osobného rozvrhu

Pre zobrazenie rozvrhu z AIS treba implementovať parsovanie rozvrhu ktorý zobrazuje AIS vo formáte HTML. Je potrebné aby sa načítala HTML stránka z rozvrhom tá sa spracovala a následne zobrazil rozvrh v aplikácii.

5.3.1 Analýza

Spracovanie HTML stránky prebehne na strane servera keďže PhoneGap neposkytuje API na prácu s DOM. Pridáme teda modul do existujúceho servera v PHP. Ten vráti spracovaný rozvrh v JSON ktorý sa už bude dať jednoducho spracovať v PhoneGap.

5.3.2 Návrh

Pri požiadavke na zobrazenie rozvrhu si aplikácia zavolá PHP skript na serveri ktorému sa pošlú prihlasovacie údaje do AIS ktorý ich následne použije na prihlásenie sa do AIS a stiahne osobný rozvrh používateľa vo forme HTML stránky. Táto stránka sa následne zparsuje pomocou DOMDocument. Tento rozvrh sa následne vloží do poľa ktoré sa pomocou funkcie json_encode() zakóduje do JSON formátu a pošle klientovi. Ten to následne zobrazí.

5.4 Prevod bitmapových obrázkov do SVG

Pre ušetrenie miesta sme zvažovali možnosť použitia SVG obrázkov. Keďže mapy sú jednoliate plochy takže sú ľahko opísateľné formou vektorov a kriviek. Použili sme viacero programov na prevod z bitmapových obrázkov do vektorového formátu. Taktiež by sa dali takéto vektorové mapy prakticky neobmedzene zväčšovať bez straty kvality.

3dsMax – samotný 3dsMax ponúka na export aj EPS formát čo by mal byť vektorový. Nenašli sme ale spôsob ako ho previesť na SVG ktoré by bolo použiteľné na zobrazenie.

Autotrace – program ktorý dokáže okrem SVG prevádzať aj do veľa ďalších vektorových formátov. Jeho prevod ale nie je celkom presný vyzerá ako kreslený štetcom. Taktiež veľkosť výstupu je omnoho väčšia ako PNG. Kompresiou pomocou zip sa veľkosť dostane na porovnateľnú úroveň ako PNG s vysokým rozlíšením.

AdobeFlash – tu sme narazili na celkom dobrú kvalitu ale problém bolo zobrazenie na iPhone zariadeniach keďže nepodporujú Flash a taktiež prepojenie na zvyšok programu.

Nakoniec sme sa rozhodli použiť bitmapové obrázky keďže aj keď majú vysoké rozlíšenie tak je ich veľkosť prijateľná a sú najjednoduchšie zobraziteľné.

5.5 Zobrazenie mapy v HTML tabuľke

5.5.1 Zobrazenie mapy

Kvôli problémom so zobrazovaním obrázkov vo vysokej kvalite na platforme Android sme chceli zobraziť mapu ako obrázok, ktorý by bol rozdelený na viac menších častí, ktoré by boli zobrazené v tabuľke. Tento prístup sa ukázal ako neefektívny, keďže zobrazovací výkon na testovacom zariadení (Sony Xperia so systémom Android 2.2) bol nedostatočný, hlavne pri mierke inej ako 1:1 (t.j. ak sme obrázku nastavili rozmery odlišné od rozmerov samotného obrázka). Ďalším problémom boli problémy s posúvaním mapy, kde sme narazili na implementačné problémy a preto sme sa rozhodli použiť externú knižnicu (viac nižšie).

Mapa je teda finálne implementovaná ako html5 canvas, kde sú vykreslené jednotlivé dielčie obrázky (zvolili sme rozmer 1000x1000 pixelov, keďže táto veľkosť je pod hranicou, ktorá spôsobuje zníženie kvality obrázkov pri kompilácii PhoneGap aplikácie, t.j. kvalita ostane zachovaná).

Použitie canvasu sa ukázalo ako dobrá cesta s mnohými výhodami:

- celá mapa je jeden canvas a nie viac prekrývajúcich sa html elementov, čo umožňuje jednoduché posúvanie mapy (opäť vid' nižšie)
- približovanie/oddial'ovanie mapy je implementované jednoducho ako zmena mierky vykresľovania do canvasu (`canvas.scale()`)
- jednoduchá integrácia navigácie (cesta sa vykresľuje priamo do canvasu)
- jednoduché pridávanie prvkov do mapy, napríklad šípka lokalizujúca miestnosť
- dobrý výkon pri posúvaní/zoomovaní, keďže mapa je vykreslená na canvas, a teda odpadá potreba prepočítavať polohu viacerých elementov (pri posúvaní) a prepočítavať veľkosť obrázka (pri zoomovaní, resp. zobrazovaní obrázku v mierke inej ako 1:1)

5.5.2 Posúvanie mapy

Kvôli implementačným problémom s posúvaním mapy (problémy s výpočtami pozície pri zmene orientácie zariadenia (vertikálne vs. horizontálne), zoomovaní, zmene poschodia a pod.) sme sa rozhodli využiť externú knižnicu. Voľba padla na knižnicu iScroll [1], keďže ide o rozšírenú a otestovanú knižnicu s potrebnými vlastnosťami (multiplatformová, kompatibilná s jQuery Mobile).

Integrácia do aplikácie je jednoduchá, mape-canvasu sa pridá jeden wrapper, nad ktorým sa inicializuje objekt `iscroll`. Prakticky všetko ostatné obstará samotná knižnica.

5.5.3 Výber miestnosti

Pri kliknutí do mapy je potrebné určiť, či sa na mieste, na ktoré používateľ klikol, nachádza miestnosť, a ak áno, ktorá miestnosť to je.

Na určenie miestnosti je nutné najprv prepočítať súradnice kliknutia (ťuknutia) na obrazovke na súradnice na mape. Tu sa berie do úvahy:

- umiestnenie mapy na obrazovke (Y súradnica je zmenšená o výšku titlebaru aplikácie)
- posunutie mapy (zistí sa z objektu `iscroll`)
- mierka mapy

Po zistení súradnice na mape sa pomocou algoritmu [2] zistí, do ktorého polygónu ohraničujúceho

miestnosti súradnice patria. Na zrýchlenie používame bounding boxy.

Polygóny miestností sú uložené v súbore, kde sú zoradené podľa poschodí a názvu miestnosti.

Polygóny je možné editovať v editore navigačného grafu. Tieto údaje sa tiež používajú pri lokalizácii miestnosti na mape (za stred miestnosti momentálne považujeme stred bounding boxu).

5.6 Vyhľadávanie osôb a miestností

5.6.1 Vyhľadávanie

Na vyhľadávanie osôb a miestností používame existujúce API servera virtuálnej FIIT:

- info/search/room
- info/search/all

Pre účely použitia v PhoneGap aplikácii bolo nutné pridať nový formát výstupu JSONP [3], keďže formát JSON nie je možné získať z inej domény (PhoneGap aplikácia beží na rozdiel od klienta virtuálnej FIIT na telefóne a nie na rovnakej doméne ako server).

Pre zobrazenie výsledkov používame plugin jQuery UI Autocomplete [4].

5.6.2 Zobrazenie informácií o osobách, miestnostiach a predmetoch

Každý typ entity (osoba, miestnosť) má vlastnú obrazovku, kde sa zobrazujú informácie. ID entity sa zistí z GET parametra a pri zobrazení stránky sa Ajax-om načítajú informácie. Opäť používame existujúce API servera virtuálnej FIIT a opäť bolo nutné pridať formát JSONP:

- osoba
 - info/person/about
 - info/person/courses
 - info/person/timetable
- miestnosť
 - info/room/about
- predmet
 - info/course/timetable

Získané informácie sa následne už len zobrazia.

5.6.3 Offline mód

Použité API volania sa nevolajú priamo, ale cez wrapper, ktorý umožňuje transparentne spracovávať volania, a teda po implementácii uvedených volaní v mobilnom klientovi bude offline mód fungovať transparentne.

5.7 Lokalizácia používateľa

Pod lokalizáciou používateľa je treba rozumieť možnosť prečítať QR code v ktorom je zakódovaná GPS súradnica s aktuálnou polohou. Phonegap dokáže pomocou pluginov, pre jednotlivé operačné systémy mobilných zariadení, prečítať QR code. Pluginy využívajú knižnicu Zxing. Plugin sa pripojí k aplikácii vytvorenej pre phonegap.

5.7.1 Android

Požiadavky :

- Eclipse
- Android development tools
- Android SDK
- Phonegap

Krok za krokom pre „hello world“ aplikáciu sa nachádza priamo na stránke www.phonegap.com. S daným návodom nebol žiaden problém.

Pridávanie pluginu BarcodeScanner [5]:

1. Pridať „LibraryProject“ do eclipsu.
2. Treba ísť do projektových vlastností, sekcia android a zaškrtnúť „je knižnica“.
3. Vo vytváranej aplikácii a jej vlastnostiach, sekcia android treba pridať knižnicu.
4. Barcodescanner.js premiestniť do zložky www.
5. Vytvoriť zložku „com/phonegap/plugins/barcodescanner“ v zložke src a vložiť tam barcodescanner.java
6. Do zložky res/xml/plugins.xml vložiť

```
<plugin name="BarcodeScanner"
value="com.phonegap.plugins.barcodescanner.BarcodeScanner"/>
```

7. Do android manifestu vložiť riadok

```
<!-- ZXing activities --> <activity
android:name="com.google.zxing.client.android.CaptureActivity"
android:screenOrientation="landscape"
android:configChanges="orientation|keyboardHidden"
android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
android:windowSoftInputMode="stateAlwaysHidden"> <intent-filter> <action
android:name="com.phonegap.plugins.barcodescanner.SCAN"/> <category
android:name="android.intent.category.DEFAULT"/> </intent-filter> </activity>
<activity android:name="com.google.zxing.client.android.encode.EncodeActivity"
android:label="@string/share_name"> <intent-filter> <action
android:name="com.phonegap.plugins.barcodescanner.ENCODE"/> <category
android:name="android.intent.category.DEFAULT"/> </intent-filter> </activity>
```

Bod jedna a dva bolo treba preskočiť a vložiť danú knižnicu priamo do aplikácie inak sa po skompilovaní vyskytla chyba. Hlásenie bolo nenájdená funkcia scan().

Každý typ kódu (QR, Aztec, EAN, datamatrix, ...) mal tri package (balíčky). Napríklad datamatrix, datamatrix.decoder, datamatrix.detector ktoré zaberali zbytočne veľa pamäte. Ich vyhodenie ovplyvnilo iba zxing.MultiFormatReader - v prípade 1D čiarových kódov aj MultiFormatWriter a znížilo potrebnú pamäť takmer o polovicu. Pridaný súbor phonegap-1.4.1.js je nutný pre barcodescanner 67 ktorý registruje plugin do phonegapu.

5.8 Mapy mobilnej aplikácie

Základom mobilnej aplikácie pre navigáciu sú mapové plány všetkých poschodí, bolo potrebné tieto mapy vytvoriť.

Analýza

Aplikácia bude používať mapy ako základ navigácie. Tieto mapy je potrebné získať vo vhodnom formáte. Po dôkladnej analýze použitého frameworku PhoneGap sme sa zhodli, že najlepším formátom týchto máp bude bezstratový formát PNG..

Prvotne sme chceli používať mapy vo vektorovom formáte ale neskôr sme zistili, že obmedzenia tohto frameworku nám to nedovoľujú.

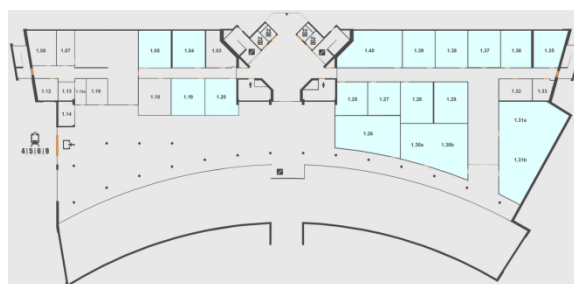
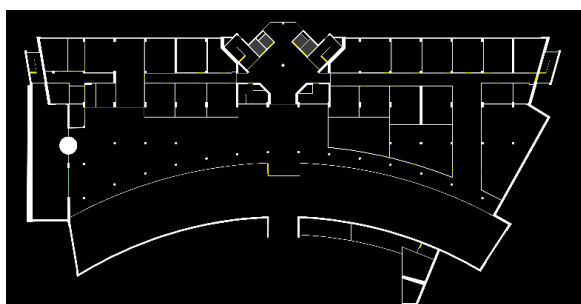
Export máp z prostredia 3DSMax

Keďže 3D verzia novej budovy FIIT je už namodelovaná vo formáte 3DSMax, je možné získať z modelu 2D podklady týchto máp. Prostredie 3DSMax podporuje export týchto máp vo vybranom formáte PNG..

Úprava bitmapových máp

Mapy po exporte z prostredia 3DSMax bolo potrebné graficky upraviť a odstrániť viacero zbytočných a prekážajúcich prvkov. Taktiež bolo potrebné niektoré prvky dorobiť. Jednotlivé úpravy boli vykonané pomocou voľne dostupného programu GIMP. Zoznam úprav je nasledovný:

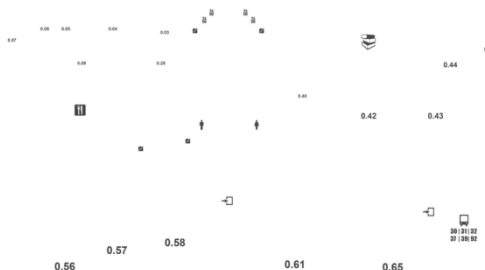
- zmena farby stien
- zmena farby dverí
- zmena farby pozadia
- úprava hrúbky stien
- odstránenie nosných pilierov
- pridanie čísiel miestností
- pridanie farieb podľa účelu miestnosti
- pridanie značiek pre špeciálne miestnosti (výťah, schodisko, WC...)



Obr. č. 29: Mapa poschodia 1 pred a po úprave

Taktiež pre lepšiu editáciu v budúcnosti a taktiež viac možností zobrazovania v rámci mobilnej aplikácie sme sa rozhodli, že mapy budú vytvorené vo vrstvách. Vrstvy budú vytvorené ako priehľadné obrázky vo formáte PNG. Prekrytím týchto vrstiev (obrázkov) získame kompletne mapy so všetkými prvkami. Zoznam vrstiev je nasledovný:

1. vrstva s modelom (plánom)
2. vrstva so zafarbením miestností
3. vrstva so špeciálnymi značkami a číslami miestností



Obr. č. 30: Vrstva so špeciálnymi ikonami a číslami miestností



Obr. č. 31: Vrstva so zafarbením miestností



Obr. č. 32: Vrstva s modelom (plánom) poschodia

5.9 Analýza aplikácie iTransit

5.9.1 Okno (záložka) departures (odchody autobusov)



Obr. č. 33: Odchody z najbližších zastávok

Popis okna

- v tomto okne sú zobrazené odchody z najbližších zastávok spolu so vzdialenosťou zastávky
- najbližšie zastávky sú detekované pomocou GPS
- vzdialenosť zastávky a používateľa je taktiež detekovaná pomocou GPS
- pri jednotlivých zastávkach sú vypísané spoje aj s príslušnými časmi odchodov
- v zátvorke pri čase príchodu je uvedený časový údaj, za ktorý má daná linka očakávaný príchod na zastávku

Zhodnotenie pre našu aplikáciu

- ako zoznam liniek použiť linky v okolí novej budovy FIIT (Mlynskej doliny)
- po kliknutí na danú linku bude používateľ presmerovaný na časový rozvrh danej linky namiesto vzdialenosti uviesť, o ktorú zastávku sa jedná (smer jazdy – konečnej stanice)

5.9.2 Okno (záložka) around me (Kde som?)



Obr. č. 34: Mapa

Popis okna

- v tomto okne je zobrazená mapa s aktuálnou pozíciou používateľa
- pozícia je detekovaná pomocou GPS
- mapa je vo forme vlozenej stránky Google maps
- červená šípka určuje fyzickú pozíciu používateľa, čierne značky určujú
- po kliknutí čiernej značky sa nám zobrazí okno so zoznamom liniek danej zastávky – po kliknutí na modrú šípku (viď obr. 34) nás aplikácia presmeruje na časový rozvrh danej linky
- mapa sa dá prepínať medzi tromi vzhlľadmi (viď obr. 34)

Zhodnotenie pre našu aplikáciu

- v našej aplikácii by stačilo použiť mapu pre okolie novej budovy FIIT (Mlynskej doliny)
- najvýhodnejšie by bolo použiť čo najprehľadnejšiu mapu (viď pravá časť obr. 34)
- detekcia pozície používateľa pomocou GPS je zbytočná, keďže aplikácia je určená hlavne pre pravidelných návštevníkov školy, navigáciu priamo na zastávku nepotrebujú

5.9.3 Okno (záložka) stops (zastávky)



Obr. č. 35: Zastávky



Obr. č. 36: Zoznam zastávok

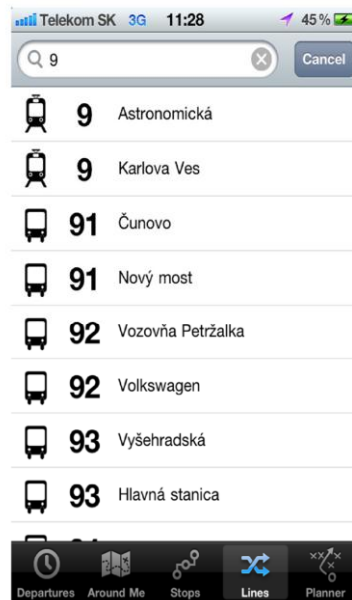
Popis okna

- okno pre vyhľadávanie zastávok pomocou mena (vid'. obr. 35)
- automatické obnovovanie výsledkov po zadaní každého nového znaku
- po kliknutí na danú zastávku sa zobrazí zoznam liniek danej zastávky (vid'. obr. 36)

Zhodnotenie pre našu aplikáciu

- keďže počet zastávok v okolí školy nie je veľký, postačuje použitie zoznamu vo forme spomenutej vyššie
- z dôvodu menšieho počtu zastávok, nie je potrebné implementovať funkciu vyhľadávania

5.9.4 Okno (záložka) lines (linky)



Obr. č. 37: Linky

Popis okna

- podobné ako vyhľadávanie zastávok
- namiesto mien zastávok je vyhľadávanie pre číslo linky
- po kliknutí na danú linku je používateľ presmerovaný na zoznam jednotlivých zastávok tejto linky
- automatické obnovovanie výsledkov po zadaní každého nového znaku

Zhodnotenie pre našu aplikáciu

- keďže počet liniek v okolí školy nie je veľký, postačuje použitie zoznamu vo forme spomenutej vyššie
- z dôvodu menšieho počtu liniek, nie je potrebné implementovať funkciu vyhľadávania

5.10 Zobrazenie cestovného poriadku

5.10.1 Zástavky mestskej verejnej dopravy v oblasti budovy FIIT

V oblasti budovy FIIT sa nachádzajú 2 zástavky mestskej verejnej dopravy. Najbližšou zastávkou je zastávka ZOO. Vzdialenejšou ale stále dostupnou zastávkou je zastávka Botanická záhrada.

5.10.2 Spoje prichádzajúce/odchádzajúce z jednotlivých zastávok

Zastávka ZOO (blízka zastávka):

- autobus 30 (smer Lamač)
- autobus 30 (smer Nový most)
- autobus 31 (smer Trnavské mýto)
- autobus 31 (smer Hlavná stanica)
- autobus 32 (smer Hlavná stanica)
- autobus 32 (smer Kuklovská)
- autobus 37 (smer Záhorská Bystrica)
- autobus 37 (smer Nový most)
- autobus 39 (smer Súhvezdná)
- autobus 39 (smer Cintorín Slávičie údolie)
- autobus 92 (smer Vozovňa Petržalka)
- autobus 92 (smer Volkswagen)
- autobus 192 (smer Vozovňa Petržalka)
- autobus 192 (smer Volkswagen)

Zastávka Botanická záhrada (vzdialenejšia zastávka):

- električka 4 (smer Pri kríži)
- električka 4 (smer Zlaté piesky)
- električka 5 (smer Pri kríži)
- električka 5 (smer Komisárky)
- električka 6 (smer Karlova Ves)
- električka 6 (smer Šafárikovo námestie)
- električka 9 (smer Karlova Ves)
- električka 9 (smer Astronomická)
- autobus 28 (smer Nová budova SND)
- autobus 28 (smer Opletalova)
- autobus 29 (smer Nový most)
- autobus 29 (smer Hrad Devín)
- autobus 32 (smer Hlavná stanica) – spoj dostupný z blízkej zastávky ZOO
- autobus 32 (smer Kuklovská) – spoj dostupný z blízkej zastávky ZOO

5.10.3 Dostupnosť údajov

Databáza iTransit:

- databáza mobilnej aplikácie iTransit
- dostupná v rámci fakulty FIIT
- ideálne riešenie – jednoduchý prístup, jednoduché získavanie údajov
- náš tím momentálne nemá prístup k tejto databáze

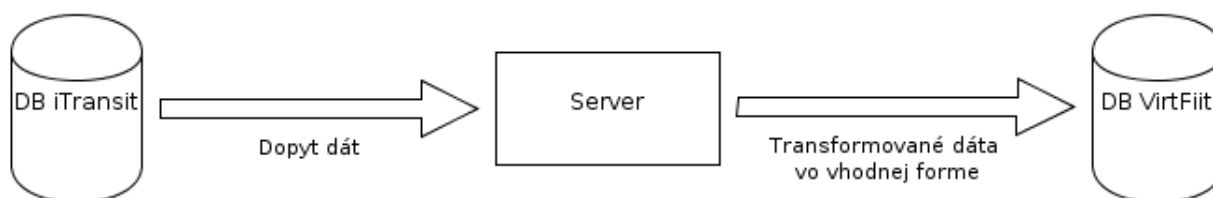
Web mestskej hromadnej dopravy:

- webová stránka spravovaná dopravným podnikom BA
- obsahuje grafikony všetkých liniek mestskej hromadnej dopravy
- záložné riešenie – nutnosť parsovať údaje, zmena web stránky môže znemožniť

funkcionalitu

5.10.4 Import dát do databázy VirtualFiit

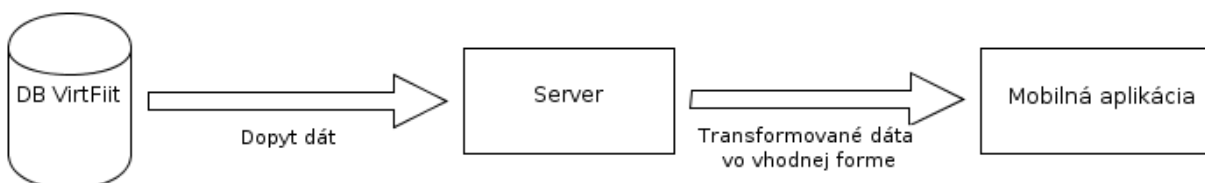
1. Dopyt dát z databázy iTransit (prípadne webu MHD) serverom
2. Transformácia dát do vhodnej formy serverom
3. Uloženie dát do databázy VirtFiit



Obr. č. 38: Import dát

5.10.5 Dopyt dát mobilnou aplikáciou

1. Dopyt dát z lokálnej databázy VirtFiit serverom
2. Transformácia dát do vhodnej podoby serverom
3. Prijatie a zobrazenie transformovaných dát mobilnou aplikáciou



Obr. č. 39: Dopyt dát mobilnou aplikáciou

5.10.6 Návrh tabuliek

Tabuľka Spoj

Názov stĺpcu	Typ	Popis
id_spoju	INT AUTO_INCREMENT PRIMARY KEY	Unikátny indentifikátor spoju
spoj	INT	Číslo linky (napr. 92)
smer	VARCHAR(255)	Smerovanie linky (napr. Nový most)
typ	VARCHAR(255)	Typ spoju (napr. autobus alebo električka)

Tabuľka Rozvrh

Názov stĺpcu	Typ	Popis
id_spoju	INT PRIMARY KEY	Identifikátor spoju
hodina	INT PRIMARY KEY	Časový záznam hodiny (napr. 14)
minuta	INT PRIMARY KEY	Časový záznam minúty (napr. 53)
rozvrh_vikend	INT PRIMARY KEY	Hodnota 1 / 0 – podľa toho, či časový údaj je platný cez víkend alebo cez bežný deň

5.10.7 Implementácia

Implementácia zobrazovania cestovného poriadku prebehla nasledovne:

1. rozšírenie importéra serveru o import údajov MHD vyparsovaných zo stránky imhd.sk
2. vytvorenie nových dopytov na server
3. zobrazenie údajov v mobilnej aplikácii

Na serveri bola API rozšírená o 2 nové dopyty

- mobile/mhd/search_timetable
 - dopyt vracia cestovný poriadok pre daný spoj vo formáte JSON
 - dopyt je odoslaný s dvoma parametrami
 - prvý parameter je číslo linky spoju
 - druhý parameter je číslo 0 alebo 1 – 0 pre cestovný poriadok pre pracovný deň a 1 pre cestovný poriadok pre voľný deň
 - príklad – mobile/mhd/search_timetable/92/0 vracia cestovný poriadok vo formáte JSON pre linku 92 cez pracovné dni
- mobile/mhd/get_buses
 - dopyt vracia zoznam všetkých liniek z databázy na serveri
 - dopyt je odoslaný bez parametrov
 - príklad – mobile/mhd/get_buses vracia zoznam všetkých liniek spolu s údajom začiatkovej zastávky, smerom jazdy a typom vozidla vo formáte JSON

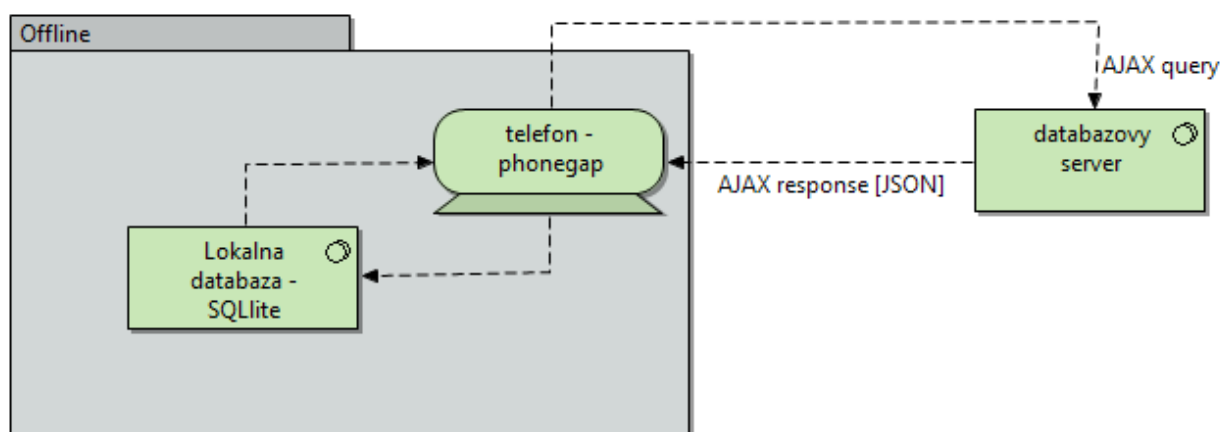
5.11 Analýza použitia databázy na mobilných zariadeniach

Phonegap priamo umožňuje vykonávať dotazy nad lokálnou databázou v mobilnom telefone[7].

Priamo podporované zariadenia sú:

- Android
- iPhone
- BlackBerry WebWorks (OS 6.0 a väčší)

Implementované riešenie minuloročného teamu obsahuje databázový server ktorý by sme mohli použiť pre naše potreby. Server na dotaz vo formáte Ajax odošle späť požadovanú odpoveď v správnom tvare vo formáte JSON.



Obr. č. 40: Navrhnuté riešenie

Na obrázku 1 je znázornené možné riešenie. Aplikácia by mala byť schopná v stave offline získavať dáta z lokálnej databázy. Táto časť navrhnutého riešenia bude riešená SQL dotazmi nad databázou SQLite v mobilnom telefóne.

V online režime, teda v režime pri ktorom bude prebiehať komunikácia priamo so serverom, bude komunikácia riešená podobne ako na desktopovej aplikácii.

5.12 Analýza možnosti použitia jedalen.stuba.sk

Jedalen.stuba.sk umožňuje používateľom prezerať si jedálne lístky viacerých jedáľenských zariadení v rámci STU. Internetová stránka zároveň umožňuje prihlásiť sa do systému (meno heslo z AIS) a poprípade si objednať menu. V našej aplikácii by sme radi využili iba prvú spomenutú funkciu zobrazenia jedálneho lístka.

Systém obsahuje jedálny lístok, maximálne dva týždne dopredu aj to nie pre všetky jedálne. Keďže v okolí FIIT STU a aj iných fakúlt pôsobí viacero jedální treba ponúknuť možnosť vybrať si jedáleň.

Stránka je napísaná v kóde ASP.

Úlohy parsovania informácií zo stránky:

- Treba zistiť ako poslať request na server pre konkrétny deň
- Zároveň aj pre konkrétne stravovacie zariadenie
- Parsovanie samotných informácií
 - Ponúkané menu
 - Miestami ponúkajú „minútku“
 - Cena

Všetky tieto údaje by sa uložili do databázy. Aktualizáciu bude treba robiť minimálne každé dva týždne.

Druhá možnosť získania dát:

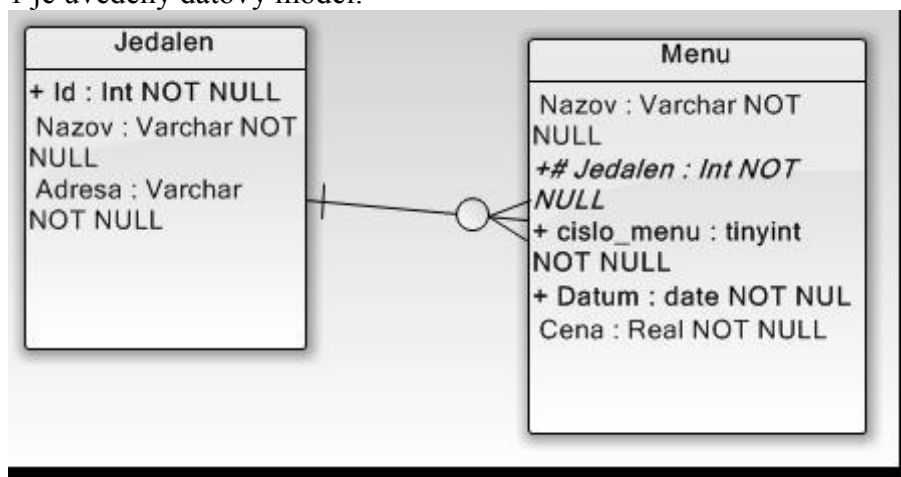
Prevádzka študentských jedální:

- Ing. Eva Beständigová: tel.: 0918/664 058, e-mail: eva.bestandigova@stuba.sk

Zatiaľ iba písomne potvrdili ochotu poskytnúť informácie.

5.12.1 Dátový model

Na obrázku č 1 je uvedený dátový model.



Obr. č. 41: Dátový model

5.12.2 Opis dátových entít

Tu sa nachádza opis dátových entít. Všetky dátové entity majú ako primárny kľúč stĺpec s menom Id typu int. Keďže SQLite neponúka ako dátový typ money treba ho nahradiť dátovým typom real [8] [9].

Jedáleň

Táto entita predstavuje stravovacie zariadenia z ktorých si používateľ môže vybrať.

- Nazov - Varchar- Meno jedálne napríklad „Výdajňa SvF“
- Adresa – Varchar - adresa jedálne napríklad „Staré Grunty 53, 842 47 Bratislava“

Menu

Táto entita obsahuje ponúkané menu pre konkrétny deň a jedáleň.

- Nazov_jedla – Varchar – Meno ponúkaného jedľa
- Cena – Real – cena ponúkaného menu
- Cislo_menu – tinyint – PK – číslo v položke menu (pre konzistenciu s webovou aplikáciou)
- Datum – date – PK – dátum kedy sa dané menu ponúka

5.13 Offline mód

Databáza

Pre funkčnosť offline módu je potrebné, aby sa dali dáta ukladať v mobilnom telefóne. Na ukladanie dát sme využili implementáciu HTML5 modulu WebSQL v prostredí PhoneGap [10]. Databáza je štruktúrou aj obsahom ekvivalentná databáze na serveri.

Po spustení aplikácie sa skontroluje, či databáza existuje a či obsahuje potrebné tabuľky. V prípade, že neobsahuje, štruktúra databázy aj údaje sa načítajú z konfiguračného adresára. Tento proces predstavuje inicializáciu databázy v telefóne a zabezpečuje, že offline režim funguje aj bez predchádzajúceho prístupu na internet.

Offline mód

Offline mód je implementovaný pomocou wrappera, ktorý zapuzdruje volania na server. V prípade, že je aktívny offline mód, sa namiesto volaní na server zavolá zodpovedajúca interná funkcia, ktorá je funkčne ekvivalentná metóde na serveri, ktorá spracúva dané volanie. Ak je offline mód neaktívny, volania sú spracované ako requesty na server cez internet.

Aktuálne implementované volania sú nasledovné:

- mobile/mhd/get_lines
- mobile/mhd/search_timetable
- info/course/timetable
- info/person/about
- info/person/courses
- info/person/timetable
- info/room/about
- info/search/room
- info/search/all

Prepínanie módov

Zapínanie a vypínanie offline módu je vyriešené voľbou v menu nastavení. Pri aktivovaní offline módu aplikácia ponúkne používateľovi možnosť stiahnuť aktuálnu verziu databázy zo servera. V prípade, že používateľ súhlasí, pošle sa na server request, ktorého odpoveď obsahuje všetky údaje zo serverovej databázy vo formáte JSON. Aplikácia spracuje uvedené dáta tak, že ich uloží do lokálnej databázy, kde sú následne dostupné pre offline volania. Ide o obdobný proces ako inicializácia databázy.

5.14 Možné rozšírenia mobilnej aplikácie v budúcnosti

V rámci prezentácie na súťaži TPCup bolo nášmu tímu dodaných niekoľko nápadov od porotcov ale aj bežných návštevníkov konferencie. Jednotlivé nápady boli poväčšine časovo náročné na implementáciu. Preto sme sa rozhodli spísať ich do zoznamu možných rozšírení v budúcnosti:

- navigácia založená na umiestnení štandardnej výbavy budov (hasiace prístroje, východy)
- aplikáciu rozšíriť pre použitie v obchodných strediskách, nemocniciach, ministerstvách
- v QR kóde ukladať URL tak, že si naša aplikácia by si z nej načítala miestnosť a zobrazila ju ale mobily bez našej aplikácie by boli presmerované na stránku aplikácie, kde by bol používateľovi ponúknuté stiahnutie
- zavesenie aplikácie na web, kde by fungovalo všetko okrem QR kódov
- rezervovanie miestnosti
- lokalizácia návštev podľa toho, ktorý QR kód sa odfoť
- ukladanie informácii (napr. otváracie hodiny) priamo do QR kódov

Taktiež nám boli dodané negatívne hodnotenie spísané do nasledujúceho zoznamu:

- navigácia vzhľadom na použitie vo vnútorných priestoroch je náročná, je potrebné premyslieť, aby bola použiteľná
- nie úplne zaujímavé spracovanie aplikácie, pomohlo by trocha osvieženia do GUI hlasová navigácia
- optimalizácia výťah verzus schodište
- špeciálny mód na požiadavku pre bezbariérový prístup a slabozrakých
- počet handicapovaných ľudí stúpa a stúpa aj ich integrácia do spoločnosti a komunikácie.

6 Zhodnotenie

Počas dvoch semestrov tímového projektu sa náš tím pokúšal zlepšovať v rámci tímovej práce. Zo začiatku to bolo pre všetkých členov zložité ale postupne sa začali zlepšovať všetky aspekty tímovej práce a hlavne komunikácie.

Nášmu tímu sa podarilo v zimnom semestri vylepšiť a odstrániť mnoho chýb z desktopovej 3D verzie aplikácie Virtuálna FIIT. Podarilo sa nám optimalizovať aplikáciu, odstrániť niekoľko chýb modelu a samotného pohybu v aplikácii a vylepšiť grafické rozhranie.

Na základe viacerých odporúčaní mnohých ľudí sme sa rozhodli, že v letnom semestri sa pokúsime vytvoriť mobilnú verziu tejto aplikácie pre čoraz populárnejšie smart telefóny. Aj napriek časovému obmedzeniu jedného semestra a odstúpeniu 4 členov tímu sa nám podarilo vytvoriť plne funkčnú mobilnú aplikáciu s množstvom funkcií. Niektoré funkcie sa nám z dôvodu nedostatku času nepodarilo implementovať. Tieto funkcie sme sa pokúsili zdokumentovať pre budúcu prácu na aplikácii. Taktiež sme vykonali používateľské testovanie a mnoho jednoduchších nápadov sa nám podarilo aj doimplementovať.

Práca na tímovom projekte dala všetkým členom mnoho skúseností, ktoré sa na iných predmetoch dajú získať len veľmi ťažko a zriedkavo. Medzi hlavné nadobudnuté skúsenosti patria tímová práca, komunikácia v tíme, práca viacerých ľudí na jednom produkte, riešenie krízových situácií a deadlinov, organizácia práce a času. Taktiež sa členovia zoznámili s mnohými technológiami, pričom s niektorými pracovali po prvýkrát. Medzi ne patria PHP, Javascript, HTML5, Phonegap, MySQL, Codeigniter, 3DS Studia Max. Členovia tímu na vývoj aktívne využívali podporné prostriedky ako Redmine a GIT.

Traja členovia tímu sa taktiež zúčastnili konferencie IIT.SRC 2012, kde prezentovali svoju aplikáciu v rámci súťaže TPCup. Táto jedinečná skúsenosť nám dala možnosť prezentovať a ukázať náš produkt ostatným študentom, zamestnancom a porotcom fakulty FIIT.

S mobilnou aplikáciou sme po ukončení predmetu spokojní, aj keď možných vylepšení a rozšírení je veľké množstvo. Dúfame, že aplikácia bude v budúcnosti v rámci tohto predmetu ešte vylepšená a jedného dňa sa dostane do stavu, že sa bude aktívne používať študentmi a zamestnancami v novej budove STU FIIT.

7 Zdroje

- [1] <http://cubiq.org/iscroll-4>
- [2] <http://jsfromhell.com/math/is-point-in-poly>
- [3] <http://en.wikipedia.org/wiki/JSONP>
- [4] <http://jqueryui.com/demos/autocomplete/>
- [5] <http://phonegap.com/start#android>
- [6] <https://github.com/phonegap/phonegap-plugins/blob/master/Android/BarcodeScanner/README.md>
- [7] http://docs.phonegap.com/en/1.0.0/phonegap_storage_storage.md.html#Database
- [8] <http://sqlite.org/datatype3.html>
- [9] <http://www.mail-archive.com/sqlite-users@sqlite.org/msg27247.html>
- [10] http://docs.phonegap.com/en/1.4.1/phonegap_storage_storage.md.html

A. Mobilná aplikácia - používateľská príručka

Používateľská príručka obsahuje návod na obsluhu aplikácie Virtuálna FIIT.

A.1 Ovládanie

Ovládanie je realizované pomocou dotykového displeja mobilného zariadenia.

- a) Stlačenie sa vykoná dotykom prsta na konkrétny ovládací prvok na displeji zariadenia.
- b) Pohyb v aplikácii sa vykoná podržaním prsta na obrazovke a následným posunutím prsta (Obr. A.1).



Obr. č. A.1: Posunutie prsta na obrazovke

Rozloženie ovládacích prvkov v aplikácii Virtuálna FIIT (Obr. 2) je nasledovné.

1. Informačný panel.
 - a. Podáva informáciu o aktuálnom zobrazení.
 - b. Je zobrazený na každej stránke aby používateľ vedel kde sa práve nachádza napr. poschodie.
2. Zobrazovací panel.
 - a. V tomto paneli sa zobrazujú vybrané údaje, obrazovky, alebo mapy.
3. Hlavné menu.
 - a. Slúži na prepínanie sa medzi obrazovkami a nastaveniami aplikácie.
4. Nastavenia.
 - a. Prístup k nastaveniam aplikácie.

A.2 Mapa

Mapa novej budovy FIIT sa zobrazuje po jednotlivých poschodiach (Obr. 4). Mapa reaguje na posun prsta a ťuknutie prsta.

- Posun prsta posúva mapu.
- Ťuknutie prsta zobrazí informácie o miestnosti.

Informácie o zobrazovanom poschodí sú zobrazené v informačnom paneli. Čísla miestností sú

vpísané priamo na mape. Prepínanie poschodí sa vykonáva tlačidlami

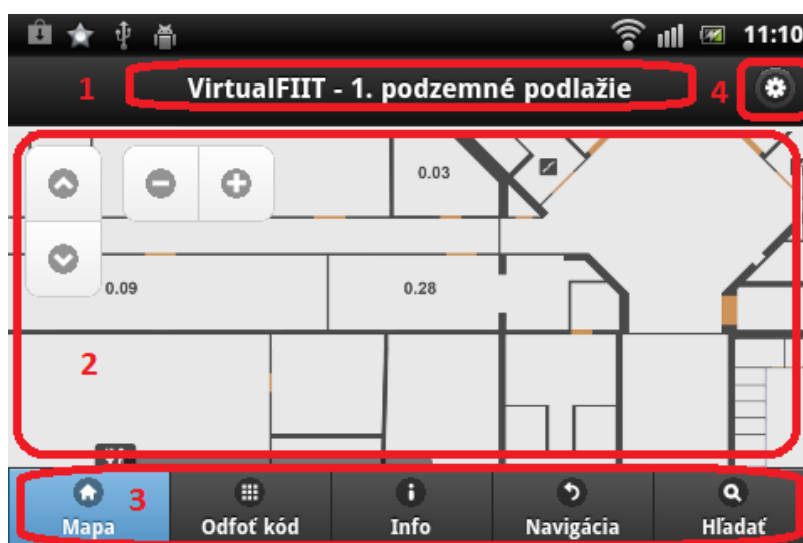


a

. Približovanie a vzdďalovanie sa vykonáva tlačidlami + a -.

A.3 Hlavná obrazovka

Hlavná obrazovka zobrazuje prvotne mapu. Pomocou hlavného menu(Obr. 3) sa môžete dostať k hľadaným informáciám, alebo vyhľadaniu konkrétnej miestnosti na mape a následným dotykom miestnosti na mape.



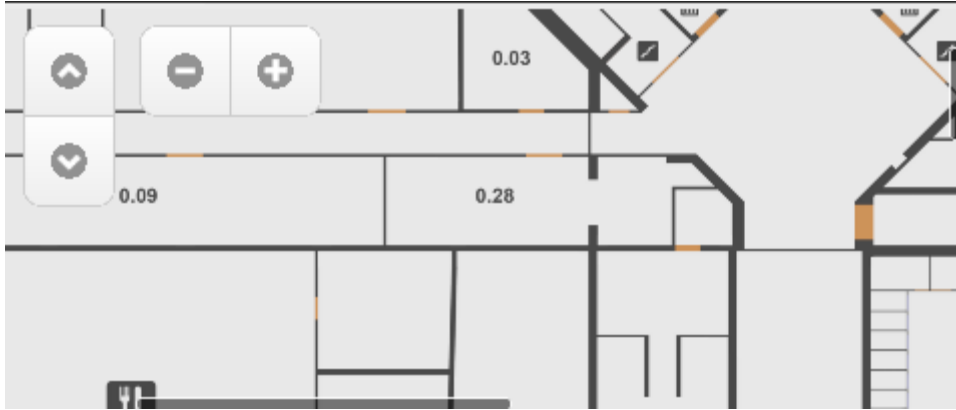
Obr. č. A.2: Rozloženie ovládacích prvkov

Na spodnej lište sú vyobrazené základné ovládacie prvky (Obr. A.1) „hlavné menu“.



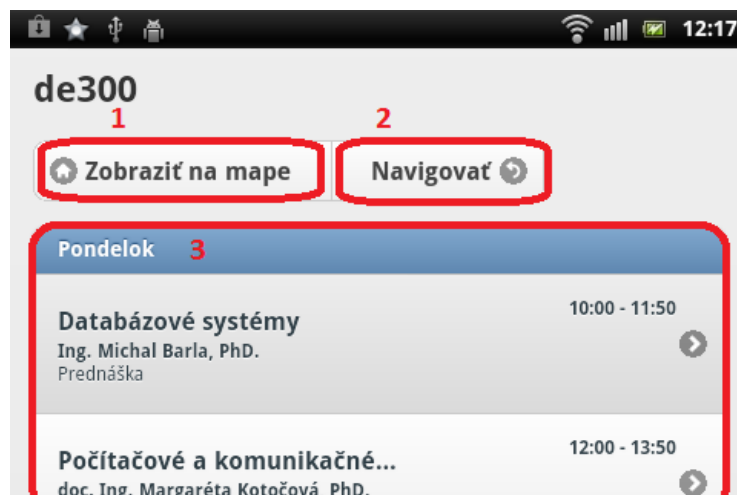
Obr. č. A.1: Tlačidlá

1. **Mapa** – Prvé tlačidlo Mapa slúži ako návrat na hlavnú stránku, na ktorej je vyobrazená mapa.
2. **Odfoť kód** – Táto položka v menu umožňuje používateľovi spustiť čítačku čiarových kódov.
3. **Info** – Táto položka v menu umožňuje prístup k väčšine informácií aplikácie.
 - a. Osobný rozvrh.
 - b. MHD.
 - c. fiit.stuba.sk.
 - d. Jedálny lístok.
 - e. AIS.
4. **Navigácia** – Slúži na prístup k menu navigácie.
5. **Hľadať** – Umožňuje vyhľadávanie.



Obr. č. A.4: Mapa

Informácie o miestnosti sa zobrazia po ťuknutí na miestnosť v zobrazení mapa, alebo po vyhľadani.



Obr. č. A.5: Miestnosť

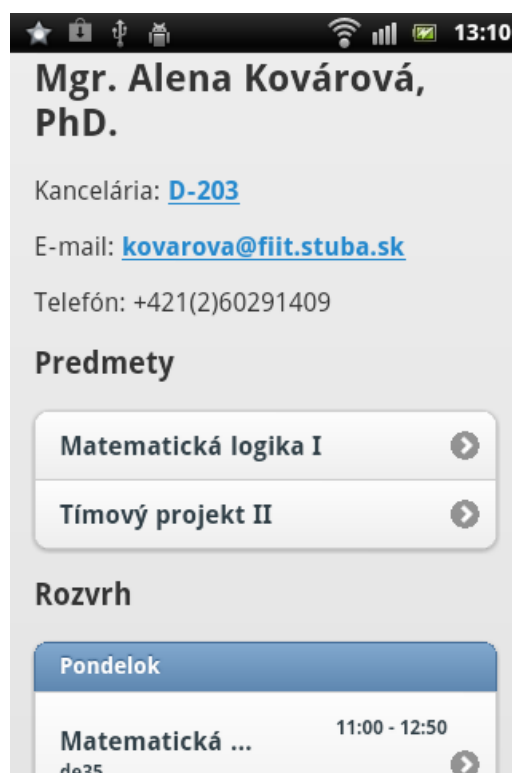
1. Zobrazí zvolenú miestnosť na mape.
2. Zobrazí navigáciu do danej miestnosti.
3. Zobrazuje info miestnosti.
 - a. Rozvrh – rozdelený do jednotlivých dní.
 - i. Názov predmetu.
 - ii. Prednášajúci / cvičiaci.
 - iii. Čas prednášky/ cvičenia.
 - b. Sídlo pracovníkov FIIT STU.

Informácie o predmete zobrazujú názov predmetu a cvičenia s prednáškami , ktoré sú zároveň rozdelené do konkrétnych dní. Po stlačení na „i“ sa zobrazia informácie o prednášajúcom, alebo cvičiacom. Po stlačení na „de300“ sa zobrazí informácia o miestnosti.



Obr. č. A.6: Informácie o predmete

Informácia o prednášajúcom, cvičiacom zobrazuje číslo kancelárie, e-mail, telefón, predmety a osobný rozvrh. Po stlačení čísla kancelárie sa zobrazí informácia o miestnosti v ktorej pracovník sídli. Email umožní poslať emailovú správu pomocou poštového klienta. Obrazovka umožňuje vybrať informácie o predmete a miestnostiach v ktorých je vykonávaná výučba.



Obr. č. A.7: Informácia o pracovníkovi fakulty

Sídlo pracovníkov fakulty. Namiesto zobrazenia predmetov a časov jednotlivých prednášok/cvičení zobrazí pracovníkov, ktorí majú sídlo v danej miestnosti.



Obr. č. A.8: Miestnosť D-203

A.4 Odfot' kód

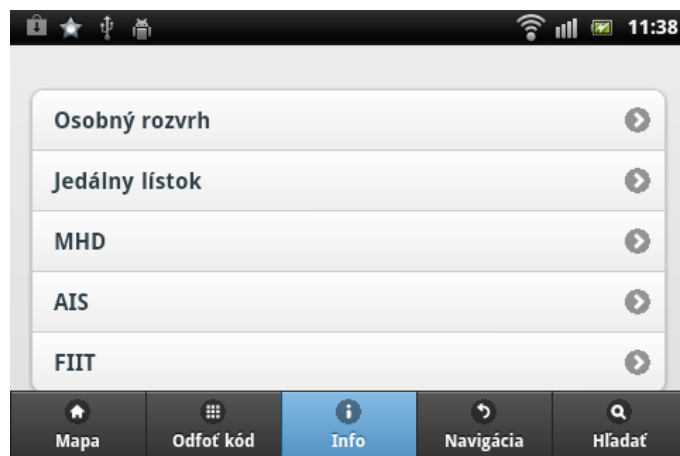
Aplikácia umožňuje pomocou čítačky QR kódov zistiť informácie o miestnosti. Zároveň umožňuje zistiť aktuálnu polohu. Stačí načítať QR kód pomocou fotoaparátu na mobilnom zariadení. Po načítaní aplikácia sama vypíše údaje, pričom nie je nutné stlačiť tlačidlo „odfot“.



Obr. č. A.9: Čítanie QR kódu

A.5 Info

Aplikácia umožňuje jednoduchý prístup k informáciám. Prístup k informáciám osobného rozvrhu, jedálneho lístka, MHD, informačného systému a fakultného webu (Obr. 10).



Obr. č. A.10: Info

A.5.1 Osobný rozvrh

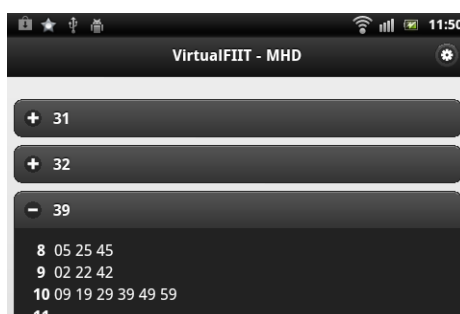
Zobrazí osobný rozvrh prihláseného používateľa (Obr. A.11). Používateľ sa môže prihlásiť cez nastavenia -> Prihlásenie do AIS.



Obr. č. A.11: Osobný rozvrh

A.5.2 MHD

Zobrazí príchody a odchody mestskej hromadnej dopravy v okolí fakulty (Obr. A.12). MHD je rozdelená podľa čísel spojov. Je potrebné poznať svoj spoj, nakoľko aplikácia nezobrazuje informácie o spoji (jednotlivé zastavenia).



Obr. č. A.12: MHD

A.5.3 FIIT

Odkaz priamo na internetovú stránku fakulty fiit.stuba.sk.

A.5.4 Jedálny lístok

Zobrazí jedálny lístok vybranej jedálne (Obr. 13).



Menu	Cena s DPH
1. Kuracie stehno pečené, ryža, šalát	3,62 €
Po 2. Koložvárska kapusta, zemiaky	3,62 €
3. Kuracia pečeň vyprážaná, zem. kaša, šalát	3,32 €
1. Vypráž. morč. prsia s nivou, zemiaky, šalát	3,62 €
Ut 2. Bravč. rebierko na rasci, ryža, šalát	3,62 €
3. Ryžový náryp s ovocím, kompót	3,32 €
1. Vypráž. bravč. karé čiernohorské, mast. zemiaky, šalát	3,62 €
St 2. Hanárka morč. prsia šnikované marhuľami	

Obr. č. A.13: Menu jedálne

A.5.5 AIS

Odkaz priamo na akademický informačný systém is.stuba.sk.

A.6 Navigácia

Aplikácia umožňuje navigovať z učebne/miestnosti do učebne/miestnosti. Stačí vložiť názov učebne, do ktorej potrebujeme navigovať, do okna „Do miestnosti“ a do okna „Z miestnosti“ z miestnosti, z ktorej chceme navigovať (Obr. A.14). V prípade, že do okna „Z miestnosti“ nič nevpíšete, bude použitá posledná pozícia načítaná aplikáciou, alebo východzia pozícia vstup do budovy.



VirtualFIIT - Navigácia

Z miestnosti:

Do miestnosti:

Navigovať

Zrušiť

Obr. č. A.14: Nastavenie navigácie

Po stlačení tlačidla „Navigovať“ aplikácia zobrazí mapu aj s vypočítanou trasou.

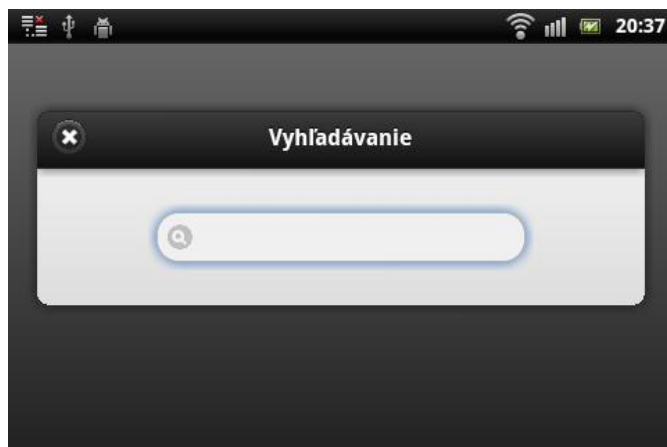
A.7 Hľadať

Umožňuje vyhľadávanie podľa viacerých faktorov (Obr. A.15).

- Názov miestnosti.
- Pracovník fakulty.

- Predmet.

Aplikácia sama ponúkne podľa zadaných slov hľadaný výraz.



Obr. č. A.15: Vyhládavanie

A.8 Nastavenia

V menu „Nastavenia“ sú nasledovné položky (Obr. A.16):

- Prihlásenie do AIS(Obr. A.17).
 - Umožňuje používateľovi prihlásiť sa do IS a získať podľa potreby osobný rozvrh (Obr. A.11).
- Prejsť do režimu offline.
 - Umožňuje používateľovi pracovať bez prístupu na internet.
 - S pýta sa či môže aktualizovať databázu. V prípade, že áno je potrebné pripojenie na internet.
 - Funkcionalita aplikácie ostáva zachovaná v prípade pravidelnej aktualizácie databázy.











Obr. č. A.16: Nastavenia



Obr. č. A.17: Prihlásenie do AIS

A.9 Vysvetlenie obrázkov na mape

Tabuľka 1 Vysvetlenie obrázkov na mape:

	Výťah
	Únikový východ
	Knižnica
	Východ, vchod do budovy
	Toalety
	Schody
	MHD
	Jedáleň

A.10 Farebné označenie miestností na mape

Mapy môžu obsahovať 3 základne zafarbenia miestností

- Bez zafarbenia (šedá farba)
 - Miestnosti bez zafarbenia nemajú bližšie určený účel
- Zelené zafarbenie
 - Farba označuje miestnosti, ktoré si primárne určené na prednášky
 - Zelená farba je rovnaká, ako farebné označenie prednášok v rozvrhoch v systéme AIS
- Tyrkysové zafarbenie
 - Farba označuje miestnosti, ktoré si primárne určené na cvičenia
 - Tyrkysová farba je rovnaká, ako farebné označenie cvičení v rozvrhoch v systéme AIS



Obr. č. A.18: Farby miestností na mape

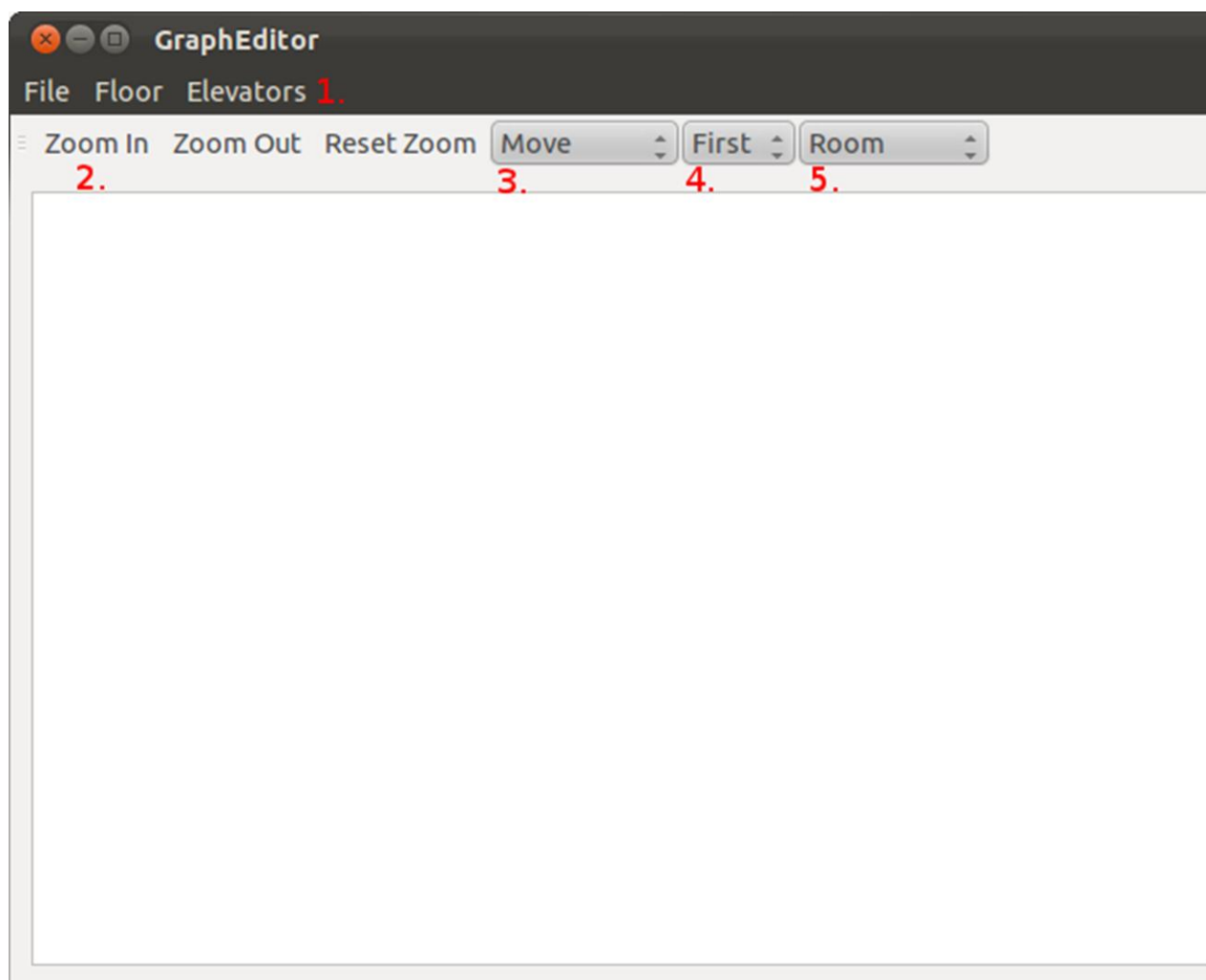
B. GraphEditor - používateľská príručka

Používateľská príručka na ovládanie editora na tvorbu navigačného grafu.

B.1 Hlavné okno

Takto vyzerá aplikácia hneď po spustení. Je vidno základné ovládacie prvky.

1. Hlavné menu
2. Ovládanie priblíženie zobrazenia
3. Výber módu editora
4. Výber poschodia



Obr. č. B.1: Hlavné okno aplikácie

5. Výber typu pridávaných bodov

B.2 Hlavné menu

V hlavnom menu sú tri položky *File*, *Floor* a *Elevators*.

B.2.1 Menu File

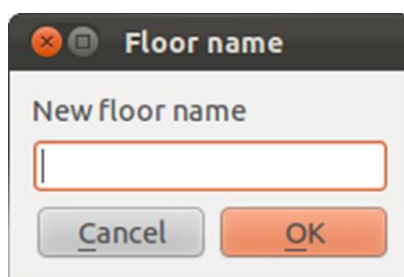
Toto menu slúži na otvorenie existujúceho súboru alebo uloženie rozpracovaného grafu.

- Open – načíta existujúci graf
- Open image – načíta podkladovú mapu aktuálnemu poschodiu
- Save – uloží graf do súboru
- Quit – ukončí aplikáciu POZOR nepýta sa na uloženie rozpracovanej práce

B.2.2 Menu Floor

Toto menu slúži na operácie s poschodiami

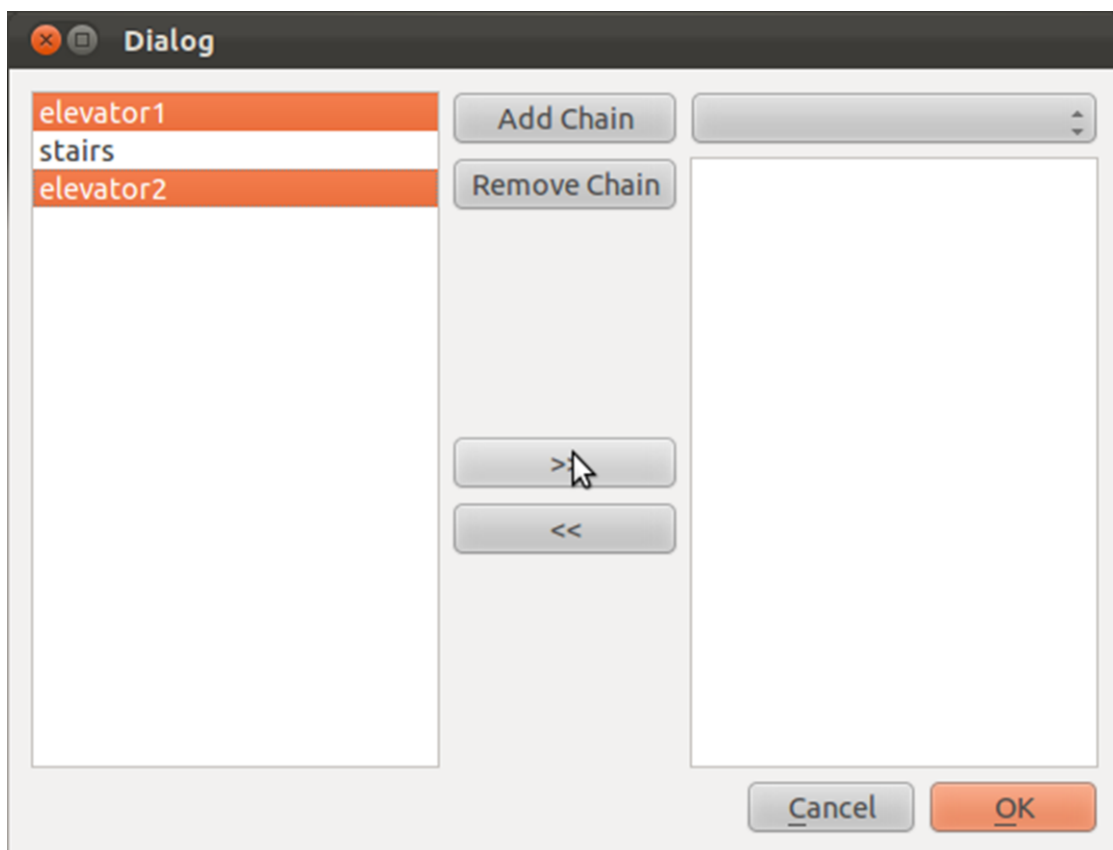
- Add floor – pridá ďalšie poschodie
- Remove floor – odstráni aktuálne poschodie
- Move up – posunie aktuálne poschodie o jednu pozíciu hore
- Move down – posunie aktuálne poschodie a jednu pozíciu dole



Obr. č. B.2: Pridávanie poschodia

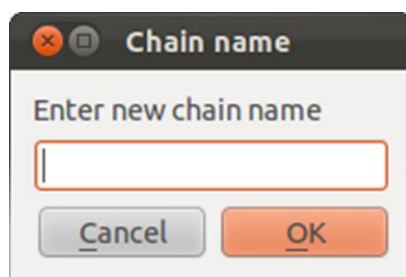
B.2.3 Menu Elevators

Vyvolá dialóg na upravovanie reťazí výtahov a schodísk. Napravo je zoznam všetkých výtahov a schodísk zo všetkých poschodí. Napravo je zoznam výtahov a schodísk ktoré patria do aktuálne vybranej reťaze.



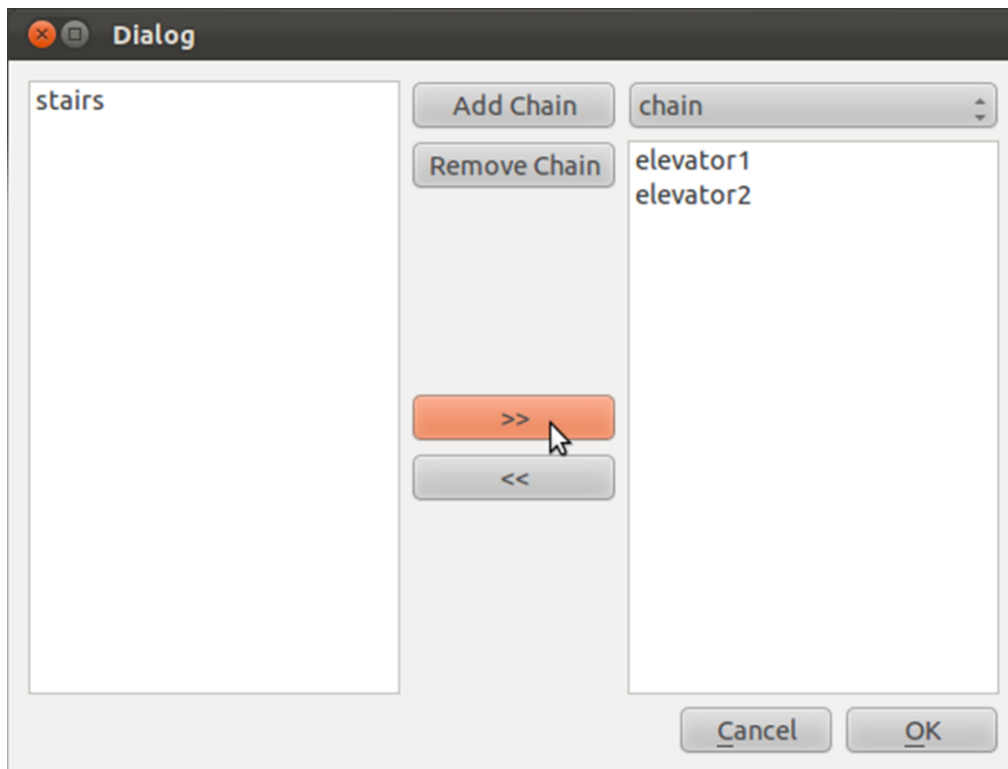
Obr. č. B.3: Dialóg na úpravu výťahových reťazi

Po kliknutí na tlačítko *Add chain* sa vyvolá dialóg na zadanie meno novej reťaze.



Obr. č. B.4: : Pridávanie reťaze

Po pridaní reťaze je možné vybrať položky naľavo a kliknutím na tlačítko >> ich pridať do reťaze. Druhým tlačítkom je ich zase možné odobrať.



Obr. č. B.5: Pridané výťahy do reťaze

B.3 *Módy* editora

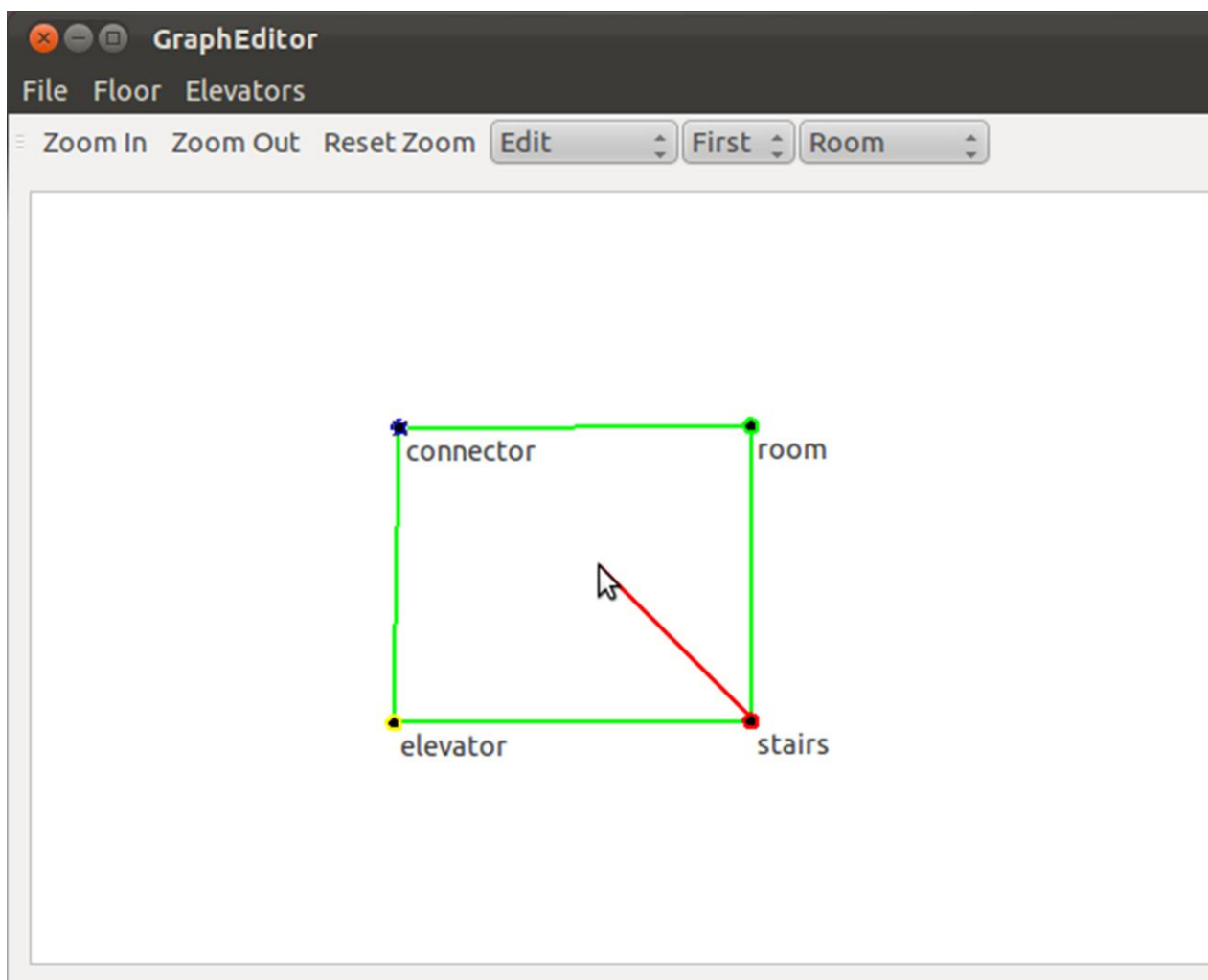
Editor pracuje v štyroch módoch: *Panning*, *Edit*, *Move* a *Room edit*.

B.3.1 Múd Panning

V tomto móde je možné posúvať aktuálne zobrazenú časť grafu. Treba kliknúť ľavým tlačítkom myši

B.3.2 Múd Edit

V tomto móde sa pridávajú nové body a hrany medzi nimi. Nové body sa pridávajú kliknutím ľavým tlačítkom myši. Typ novo pridávaných bodov je možné nastaviť pomocou výberu v panely nástrojov. Hrana sa vytvorí kliknutím na existujúci bod a za držania tlačítka potiahnutím na druhý bod a následným uvoľnením tlačítka myši.



Obr. č. B.6: Editovanie bodov a hrán

B.3.3 Múd Move

V tomto móde je možné presúvať body a polygóny. Taktiež je tu možné editovať popisok bodov a polygónov. Dvojklikom na existujúci popisok sa aktivuje editovanie textu popisok.

B.3.4 Múd Room edit

V tomto móde sa definujú ohraničenia miestností formou polygónov. Kliknutím sa začne definovať polygón a každé kliknutie pridá ďalší vrchol. Definovanie ukončíme dvojklikom kedy sa pridá posledný vrchol a polygón sa uzavrie.

Mobile Navigation in Indoor Spaces Using QR Codes

Igor AUFRICHT*, Samo FORUS†, Jozef KUJAN†,
Dušan POIZL†, Matej ŠKODA*, Juraj TÓTH†, Juraj VOLENTIER†

*Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
tim9@wx.sk*

Today, intelligent smartphones are used by many people. They have become versatile assistants. Integrated GPS is very common for them. It is useable only outdoors, therefore we must depend on other forms of navigation indoors. The last year project, created by the team SW7D, contains high quality web application for navigation and presentation of the new FIIT building in three dimensional space (the current version of Virtual FIIT project can be found online [1]). We searched for another ways to resolve the navigation in buildings effectively using mobile devices.

Inspired by projects realized at the Technical University in Graz [2] or the project of University de Vigo [3], we decided to use QR codes located at strategic points of the building. These codes will include information about their physical location in the building. This navigation is very simple, user friendly and consists of the following four steps: The user scans the nearest QR code with the camera of his mobile device. Application reads the current position from the code and asks the user to specify the endpoint of navigation (classroom number or name of an employee). Then the application displays a map with the shortest route from the actual to the final destination.

The special characteristic about our project is that it represents two different solutions joined in one application. The first part is the mobile navigation mentioned above. The second part is the information system that provides all relevant information not only for visitors but also for students and university staff. These data will be obtained directly from the academic information system (AIS). This includes information about schedules, names of employees along with their workplaces. Other information includes departures of buses as well as the current canteen menu. It is possible to get this information from other sources but we believe that such centralization is more convenient. These information are logically linked together so users can easily find them (e.g. the schedules are linked to the staff, the staff is linked to the rooms, etc.).

There is a wide variety of mobile operating systems for mobile devices which are also completely incompatible. Naturally, we want our application to be supported by many mobile devices so we searched for a fully multiplatform solution. We selected the open source framework PhoneGap. This framework allows us to create an application using the modern technology of HTML5 and JavaScript for the most used operation systems such as Android, iOS, Blackberry OS, WebOS, Windows Mobile or Symbian.

The important objective is to implement the application in a way that it could be easily deployed for other buildings through the exchange of data packages. These data packages must include a plan of building in the form of 2D model, the list of rooms, stairways and elevators along with their positions. This feature is designed in such a way that it can be easily deployed by people without technical and programming skills. The main data are stored locally in the application because of the data communication savings. The application will include support for updating the data via Wi-Fi or 3G.

* Master degree study programme in field: Software Engineering

† Master degree study programme in field: Information Systems

Supervisor: Mgr. Alena Kovárová, PhD., Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU, Bratislava

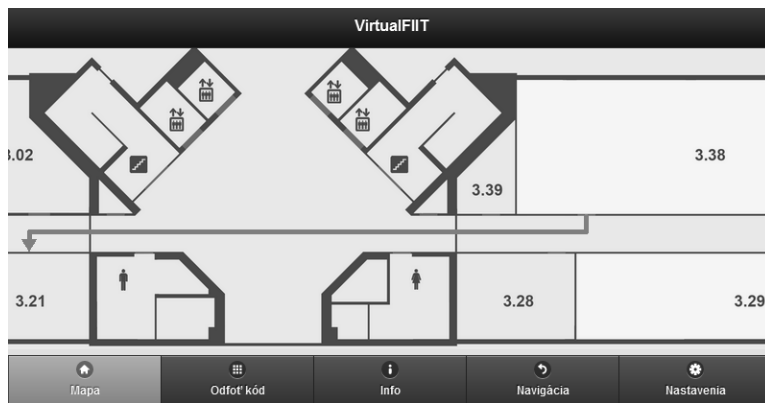


Figure 1. Example of navigation on a mobile device.

We believe that our team can create a successful mobile application using modern and progressive technologies. There are several ideas for new features that can be implemented in the future as well. Surely, we would like also the future teams or other fans all over the world to further develop the additional features for the application. There are many possible extensions such as lists of opening hours, helpful links, actual room occupancy, the number of free parking spaces, interactive 3D avatars, boards and chat. Unlimited are the future possibilities of using such a system. The extensions are, of course, not limited to the mobile version. The 3D desktop version can be a part of the interactive kiosks, which not only capture one's attention but also give an advice every time the visitor feels helpless.

References

- [1] Hlaváček, F., Hudec, J., Mešťaník, P., Novotný, M., Pečík, R., Polko, I., Palček, M.: *The Conjunction of Spatial Navigation and Information Systems through Virtual Reality*. IIT.SRC 2011, Bratislava, (2011). Project available at: <http://stavba.fiit.stuba.sk/virtfiit>
- [2] Mulloni, A., Wagner, D., Schmalstieg, D. and Barakonyi, I.: Indoor Positioning and Navigation with Camera Phones. *Pervasive Computing*, IEEE, (2009), vol. 8, pp. 22-31. [Online; accessed February 21, 2012]. Available at: http://wiki.fluidproject.org/download/attachments/6819626/indoor_positioning_camera.pdf
- [3] Costa-montenegro, E., González-castaño, F. J., Conde-lagoa, D., Barragáns-martínez, A.B., Rodríguez-hernández, P.S., and Gil-castiñeira, F.: QR-Maps: An efficient tool for indoor user location based on QR-Codes and Google maps. *Consumer Communications and Networking Conference (CCNC)*, IEEE, (2011), p. 928-932.

D. Projektový poster zo súťaže TPCup

Príloha obsahuje iba zmenšený poster. Poster v plnej veľkosti je dostupný na webovej stránke tímu 14G v sekcii na stiahnutie - <http://labss2.fiiit.stuba.sk/TeamProject/2011/team09is-si/nastiahnutie.html>

Mobile navigation in indoor spaces using QR codes

Motivation

- achieve navigation without GPS
- collect information from many systems
- provide help for new and lost visitors of buildings
- develop a multiplatform solution
- deploy application for any other building
- create a popular and used application
- add 2D mobile version into existing 3D desktop version

Navigation

- navigation from point A to point B

- easy, fast a user friendly input of start and end locations

Features

- indoor navigation and localization using QR codes
- information gathering from other systems (AIS, public transport website, Webkredit canteen system)
- intelligent link-up between navigation and information system together with intuitive searching
- local caching of data for usage without internet access
- easy deployment for other buildings (preferably STU) without the need of programing skills

Canteen menu

- daily canteen menu with prices
- information collected directly from the Webkredit system

QR reader

- localization by scanning of strategically placed QR codes around building
- easy and fast scanning with an integrated reader

Architecture

Technologies

Server	Application
GNU Linux Debian	PhoneGap
CodeIgniter	JavaScript
MySQL	HTML5
PHP	CSS3

Maps of the building

Student's personal schedule

- schedule downloaded directly from AIS
- local caching for offline use
- directly linked with student's courses

Public transport

- schedule for all public transport lines around the building
- highlighted nearest departure

Conclusions

- multiplatform application for mobile devices has been developed
- universal fitting modules and editor for enriched maps have been developed as well
- extensible in future by many functions such as opening hours, helpful links, actual room occupancy, the number of free parking spaces, interactive 3D avatars, chat etc.

STU ••• Faculty of Informatics and Information Technology Team 14G

••• Slovak University of Technology in Bratislava

••• F I I T •••

••• - Dr. Alena Kovárová -

Igor Aurficht - Samo Forus - Jozef Kujan - Dušan Poizl - Matej Škoda - Juraj Tóth - Juraj Volentier

Obr. č. D.1: Poster zo súťaže TPCup