

Slovenská technická univerzita

Fakulta informatiky a informačných technológií Ilkovičova 2, 842 16 Bratislava 4

Podporné nástroje pre navigáciu vo vnútri budov

IndoorNav

Dokumentácia k dielu

Vedúca tímu: Mgr. Alena Kovárová, PhD.

Členovia tímu: Bc. Ondrej Čerman, Bc. Matúš Demko, Bc. Juraj Marák, Bc. Samuel Pecár, Bc. Gabriel Takács, Bc. Miroslav Takács

Školský rok: 2015/2016

Obsah

1	Úvod	5
2	Slovník pojmov	6
3	Globálne ciele.....	7
3.1	Pre zimný semester.....	7
3.2	Pre letný semester.....	7
4	Celkový pohľad	8
4.1	Aplikácia Virtual FIIT	8
4.2	Navigácia v budovách	8
4.3	Aplikácia na úpravu plánu budovy.....	8
4.4	Modul na automatické rozmiestnenie beaconov	9
4.5	Diagram komponentov	9
4.6	Prípady použitia pre modul editora	11
4.7	Prípady použitia pre modul rozmiestňovania beaconov.....	12
4.8	Postup pri obsluhu modulu editora.....	13
4.9	Postup pri obsluhu modulu rozmiestňovania beaconov	15
5	Úprava Virtual FIIT.....	16
5.1	Oprava chýb	17
5.1.1	Zmiznutie prihlasovacieho okna po prihlásení.....	17
5.1.2	Miestnosť študijného oddelenia	17
5.1.3	Chybové hlásenie v detaile vyučovacích hodín.....	17
5.1.4	Nefunkčné načítanie dát z „iTranzit“ a jedálne	17
5.1.5	Tlačidlo späť v mapách	18
5.1.6	RSS správy	18
5.1.7	Kontrola pripojenia na internet pre načítaním MHD.....	18
5.1.8	Označenie miestností budovy FEI.....	18
5.2	Pridávanie nových funkcií.....	18
5.2.1	Zobrazenie rozlíšenia obrazovky vo funkcii „bug report“	18
5.2.2	Zobrazenie poznámky pod rozvrhom ako v AISE	19
6	Tvorba modulu pre editáciu mapy.....	20
6.1	Zobrazenie mapy v editore	20
6.2	Nasadenie prostredia na virtuálny server	20
6.3	Vytvorenie základného jadra modulu.....	20

6.4	Upload súboru	21
6.5	Zobrazenie mapy ako SVG	21
6.6	Načítanie obsahu súboru do aplikácie	22
6.7	Zobrazenie zoznamu vrstiev.....	22
6.8	Konverzia JSON súboru do formátu zobraziteľného na FE.....	22
6.9	Získanie informácií zo súboru DXF	22
6.10	Zobrazenie dát v podobe mapy.....	22
6.11	Zobrazovanie stavu procesu úpravy mapy	23
6.12	Zobrazovanie zoznamu vrstiev a zobrazovanie obsahu vrstiev.....	23
6.13	Problém s CORS	24
6.14	Problém s Gruntom	24
6.15	Vytvorenie homepage a informačnej stránky.....	25
6.16	Zachovať zoom a pozíciu mapy pri zmene vrstiev.....	25
6.17	Stiahnutie súboru svg	25
6.18	Indikátor aktivity - koliesko pri načítaní DXF	25
6.19	Oboznámenie s knižnicami OFP editor.....	26
6.20	Prenesenie konvertora => DXF worker.....	26
6.21	Vyriešiť automatizáciu v Grunte	26
6.22	Úprava čiary v mape.....	27
6.23	Optimalizácia výsledného SVG súboru.....	27
6.24	Analýza errorov pri úprave.....	28
6.25	Analyzovať free editory SVG	28
6.26	Analyzovať či Leaflet neumožňuje vylievačik alebo či neexistuje iný open-source nástroj.....	28
7	Tvorba modulu pre automatické rozmiestnenie beaconov	29
7.1	Podpora pre spracovanie paralelných požiadaviek.....	29
7.2	Získanie mierky mapy	29
7.3	Vyrobiť poschodie FEI.....	29
7.4	Aktualizácia mapy FIIT.....	29
7.5	Pridať beacony do mapy FIIT	30
7.6	Zistiť informácie od prof. Farkaša.....	30
7.7	Nájsť softvér na simuláciu rozmiestnenia beaconov	30
7.8	Import SVG	30
7.9	Transformácia koordinátov	31

7.10	Reprezentácia mriežky	31
7.11	Výpočet šírenia signálu do okolitých bodov	31
7.12	Aktualizovanie mapy FIIT	32
7.13	Selenium testy	32
7.14	Algoritmus rozmiestnenia beaconov	32
7.15	Vyhodnotenie rozmiestnenia algoritmom	33
7.16	Transformácia výsledku do svg mapy	33
7.17	Zobrazenie SVG mapy a beaconov	33
7.18	Unit testy	34
7.19	Oprava bugov	37
7.20	Úprava klienta pre propagation časť	37
7.21	Unit testy	37
7.22	Prezentácia na Tech Inno day	38
7.23	Úprava procesu editácie mapy	38
7.24	Vytvorenie postera	38
7.25	Prepočet jednotiek na metre	38
7.26	Príprava testovania používateľmi	39
7.27	Vytvorenie SVG pre každú chodbu na FIIT	39
7.28	Doplniť Wi-Fi do algoritmu	39
7.29	Doplniť progress bar do propagation	40
7.30	Update dokumentácie	40
7.31	Upratovanie v podkladoch k projektu	40
7.32	Prezentácia projektu na IIT.SRC	40
7.33	Vytvorenie inštalačnej príručky	40
7.34	Komplexný code review	40
7.35	Doplnenie chýbajúcich user stories	40
7.36	Vytvorenie používateľskej príručky	41
7.37	Testovanie prototypu používateľmi	41
	Príloha A Používateľská príručka	42
	Príloha B Používateľské testovanie	48
	Príloha C Inštalačná príručka	50

1 Úvod

Navigácia v priestoroch budov je v dnešnej dobe veľmi žiadaná. Existujú mnohé komplikované budovy, ako sú napríklad obchodné domy alebo administratívne budovy, v ktorých sa môže človek ľahko stratiť. Preto by bola veľmi nápomocná aplikácia, ktorá by umožnila zobrazit' mapu budovy, aktuálnu pozíciu v nej, prípadne aj cestu do určeného cieľa. Keďže v budovách nie je možné používať navigáciu pomocou GPS, je potrebné na tento účel použiť iné riešenie. Ako najvhodnejšie sa javí technológia Bluetooth LE beacon-ov.

Pri vývoji aplikácií zameraných na navigáciu vo vnútorných priestoroch je časovo veľmi náročnou činnosťou vytvorenie detailnej mapy.

Naším cieľom v prvom semestri je vytvoriť webovú aplikáciu, ktorá umožní používateľovi nahrať plány budovy, upraviť ich do požadovaného tvaru a exportovať. Aplikácia ďalej umožní pridávanie údajov pre miestnosti a udávanie polôh beacon-ov – zariadení používaných na lokalizáciu. Vytvorenie aplikácie, ktorá by dokázala tento čas skrátiť a celú činnosť tak zefektívniť, by uľahčilo prenositeľnosť aplikácie na ľubovoľnú ďalšiu budovu.

V tomto dokumente je okrem globálnych cieľov stanovených na zimný semester aj podrobný popis vyvíjanej aplikácie, jej celková architektúra a všetky jej moduly. V architektúre je najprv popísaný stav, v akom sme aplikáciu dostali, a potom aj stav v akom sa aplikácia aktuálne nachádza. Každý modul potom obsahuje zoznam používateľských príbehov a podrobný popis úloh, ktoré sa v rámci nich riešili.

Cieľom letného semestra je vytvorenie funkcionality automatického rozmiestnenie majáčikov na mape. Vhodné rozmiestnenie majáčikov zabezpečí kvalitné pokrytie budovy signálom, čím môžeme získať polohu človeka v nej presnejšie, pričom bude počet použitých majáčikov čo najmenší, čím sa ušetrí finančné prostriedky na zavedenie indoor navigácie v budove. Okrem toho sme ďalej pracovali na editore vytvorenom v zimnom semestri, zrefaktorovali sme jeho zdrojový kód a vytvorili sme testy.

V tomto dokumente je okrem globálnych cieľov aj podrobný popis vyvíjanej aplikácie, jej celková architektúra a všetky jej moduly. V architektúre je najprv popísaný stav, v akom sme aplikáciu dostali, a potom aj stav v akom sa aplikácia aktuálne nachádza. Každý modul potom obsahuje zoznam používateľských príbehov a podrobný popis úloh, ktoré sa v rámci nich riešili.

2 Slovník pojmov

Backend – serverová časť webovej aplikácie

Beacon – majáčik, zariadenie vysielajúce signál, fungujúce na technológii Bluetooth LE

Bluetooth LE – bezdrôtová technológia s nižšou spotrebou elektrickej energie

Cachovanie – ukladanie informácií do vyrovnávacej pamäte

Defaultne - prednastavene

DXF - formát výstupného súboru programu AutoCAD

Framework - softvérová štruktúra, ktorá slúži ako podpora pri programovaní a vývoji softvéru

Parsovanie - proces, pri ktorom sa vstupný text transformuje na určité dátové štruktúry

Promise - služba umožňujúca asynchrónne spracovanie

Request - požiadavka

SVG – formát vektorových obrázkov

Upload – prenos súboru na server

Redis – databáza na serveri s kľúčmi a k nim priradenými hodnotami

Subkomponent - menšia časť komponentu

3 Globálne ciele

3.1 Pre zimný semester

Prvý šprint bol venovaný opravám známych chýb v existujúcej aplikácii Virtual FIIT. Zvyšok zimného semestra sme vyhradili na tvorbu prototypu aplikácie na vytváranie mapy použiteľnej pre navigáciu v priestoroch budovy.

Plány budovy sú často zadané vo formáte nepoužiteľnom pre mobilnú aplikáciu slúžiacu na navigáciu. Naším cieľom je vytvoriť program, ktorým môže používateľ – ten kto chce pre svoju budovu poskytovať navigáciu, vytvoriť mapu s čo najmenšou námahou. Prvým krokom je nahrať súbor s plánmi budovy. Tie často obsahujú prvky, ktoré by na mape nemali byť zobrazené, a používateľ ich môže vyfiltrovať. Ďalším krokom je definovanie miestností.

Miestnosť je v najjednoduchšom prípade na mape zobrazená ako obdĺžnik, ale môže mať aj tvar polygónu. Miestnostiam môžu byť priradené atribúty, ako číslo miestnosti, názov, ktoré sú používateľovi k dispozícii. Ďalej sa na mapu pridávajú polohy a údaje o beacon-och, ktoré sú používané na lokalizáciu mobilného zariadenia. Výslednú mapu bude možné jednoducho uložiť, a tým bude pripravená na použitie v module pre navigáciu.

V neposlednom rade medzi naše ciele patrí získanie schopnosti práce v tíme. Na to, aby sme boli v tomto projekte úspešní, je potrebná dobrá tímová komunikácia a spolupráca.

3.2 Pre letný semester

Prvý šprint v letnom semestri sme venovali analýze existujúcich riešení zaoberajúcich sa šírením bluetooth signálu v priestore a algoritmu na optimálne rozmiestnenie beaconov.

Ďalšie šprinty venujeme na vytvorenie prototypu modulu na automatického rozmiestňovanie beaconov v budove. Tento modul bude použitý po vytvorení mapy v editore. Používateľ si v ňom zvolí miestnosti, v ktorých sa majú rozmiestniť beacony, a modul mu navrhne rozmiestnenie beaconov a ich počet. Rozmiestnenie navrhnuté algoritmom bude zabezpečovať dostatočné pokrytie signálom v miestnostiach, pričom bude počet použitých beaconov minimálny.

Okrem toho budeme ďalej pracovať na editore vytvorenom v prvom semestri. Chceme refaktorovať niektoré časti zdrojového kódu a vytvoriť automatické testy. Dôležitou časťou bude aj používateľské testovanie výsledného prototypu.

4 Celkový pohľad

4.1 Aplikácia Virtual FIIT

Naše práce v rámci tímového projektu začali oboznamovaním sa s aplikáciou Virtual FIIT. Túto aplikáciu sme “zdedili” od minuloročných tímov. Našou úlohou bolo hlavne oboznámiť sa s aplikáciou, použitými technológiami a následne opravovať niektoré známe chyby, ktoré sme buď sami našli, alebo nám ich niekto nahlásil v rámci funkcie ohlasovania chýb, ktorá sa v aplikácii nachádza.

Aplikácia Virtual FIIT je naprogramovaná prevažne v jazyku JavaScript a skladá sa z časti bežiackej na serveri a časti bežiackej na klientovi. Časť, ktorá beží na serveri využíva framework NodeJS, pričom časť aplikácie na klientskej strane je postavená na frameworku AngularJS. Okrem toho sa v tejto aplikácii samozrejme používajú jazyky HTML a SASS.

Vzhľadom na to, že v rámci tejto aplikácie sme zatiaľ iba opravovali niektoré chyby, prípadne sme v nej robili menšie úpravy a zmeny, sa architektúra aplikácie viditeľne oproti minuloročnej verzii nezmenila. Z tohto dôvodu na tomto mieste architektúru aplikácie neuvádzame.

4.2 Navigácia v budovách

Primárnym cieľom nášho tímu v rámci predmetu Tímový projekt je venovať sa téme navigácie vo vnútri budov. Do aplikácie Virtual FIIT chceme doprogramovať navigáciu v budove FIIT, ktorá bude založená na technológii beacon.

Systém navigácie v rámci budovy samozrejme potrebuje poznať, ako daná budova vyzerá - potrebuje poznať jej presný plán. Na tento účel sme sa rozhodli vyvinúť aplikáciu, ktorá umožňuje spracovanie plánu budovy a jej následný export do rôznych formátov (formát čitateľný pre človeka - SVG a aplikáciou čitateľný formát - JSON).

4.3 Aplikácia na úpravu plánu budovy

Pri inštalácii systému navigácie v rámci nejakej budovy je potrebné vypočítať správne rozmiestnenie zariadení typu beacon. Jedná sa o netriviálnu vec z viacerých dôvodov. Jednak potrebujeme mať presný plán budovy, na ktorom sú zaznačené nielen steny a miestnosti, ale tiež napríklad dvere.

Jednou z možností je použiť plán, podľa ktorého bola budova postavená. Takéto plány máme väčšinou k dispozícii vo formáte DWG. DWG je binárny súbor, ktorý je výsledkom kreslenia v programe AutoCAD. Zistili sme, že súbor vo formáte DWG vieme ľahko a rýchlo pomocou dostupných nástrojov prekonvertovať na formát DXF.

V tomto súbore sa však nachádza zväčša oveľa väčšie množstvo informácií a dát, ako je pre nás potrebné a žiaduce. Jedná sa o rôzne pomocné popisy, body, kvóty a pod., teda o elementy, ktoré sú potrebné pre staviteľov budovy, avšak pre nás i pre používateľov našej navigácie sú nepotrebné a mätúce. Z tohto dôvodu potrebujeme systém, ktorý nám, prípadne majiteľovi alebo manažérovi

budovy, do ktorej sa bude navigácia inštalovať, umožní nepotrebné elementy odstrániť, prípadne upraviť.

Možnosť úpravy máp je potrebná aj z viacerých iných dôvodov. Jedným z nich je, že budovy sú občas prerobené a tieto úpravy budov nie sú zakreslené na pôvodných plánoch. Ďalej je potrebné určiť názvy miestností, ktoré sa v plánoch nenachádzajú. V neposlednom rade je do plánov potrebné zakresliť vypočítané rozmiestnenie beacon-ov. Samotný výpočet rozmiestnenia beacon-ov budeme preberať od našich kolegov z FIIT, ktorí sa týmto výpočtom venujú v rámci ich diplomových prác.

System na úpravu plánu budovy musí umožňovať upravenú a spracovanú mapu exportovať a to jednak do súboru vo formáte SVG, prípadne PNG alebo JPEG kvôli zobrazenie v mobilnom zariadení, ale aj kvôli použitiu v samotnom systéme navigácie.

4.4 Modul na automatické rozmiestnenie beaconov

Pre aplikáciu pre navigáciu v budove je okrem mapy potrebné aj rozmiestniť zariadenia typu beacon v budove. Vstup pre tento modul polygón ohraničujúci chodbu, prípadne inú miestnosť. Používateľ si zvolí parametre dostupných beaconov a parametre určujúce požadovanú presnosť lokalizácie. Na dosiahnutie vyššej presnosti je potrebný väčší počet beaconov.

Algoritmus v ďalšom kroku navrhne optimálne rozmiestnenie beaconov v miestnostiach. Výstupom bude vrstva na mape obsahujúca pozície beaconov.

4.5 Diagram komponentov

Vytváraná aplikácia sa skladá z dvoch oddelených častí - klientskej časti a časti bežiacej na serveri(obr. č.1). Tieto dve časti navzájom intenzívne komunikujú a obidve sú nevyhnutné na fungovanie aplikácie.

Časť aplikácie, ktoré beží na serveri, je napísaná vo frameworku Node.js a je zodpovedná za spracovávanie požiadaviek, ktoré prichádzajú od klientskej časti aplikácie. Jej obsahom je niekoľko komponentov, ktoré vykonávajú rozličné úlohy, v závislosti od požiadavky. Jedným z komponentov je "DXF to SVG". Tento komponent sme naprogramovali my a jeho subkomponenty vykonávajú všetky základné operácie, ktoré vykonávame s DXF súborom:

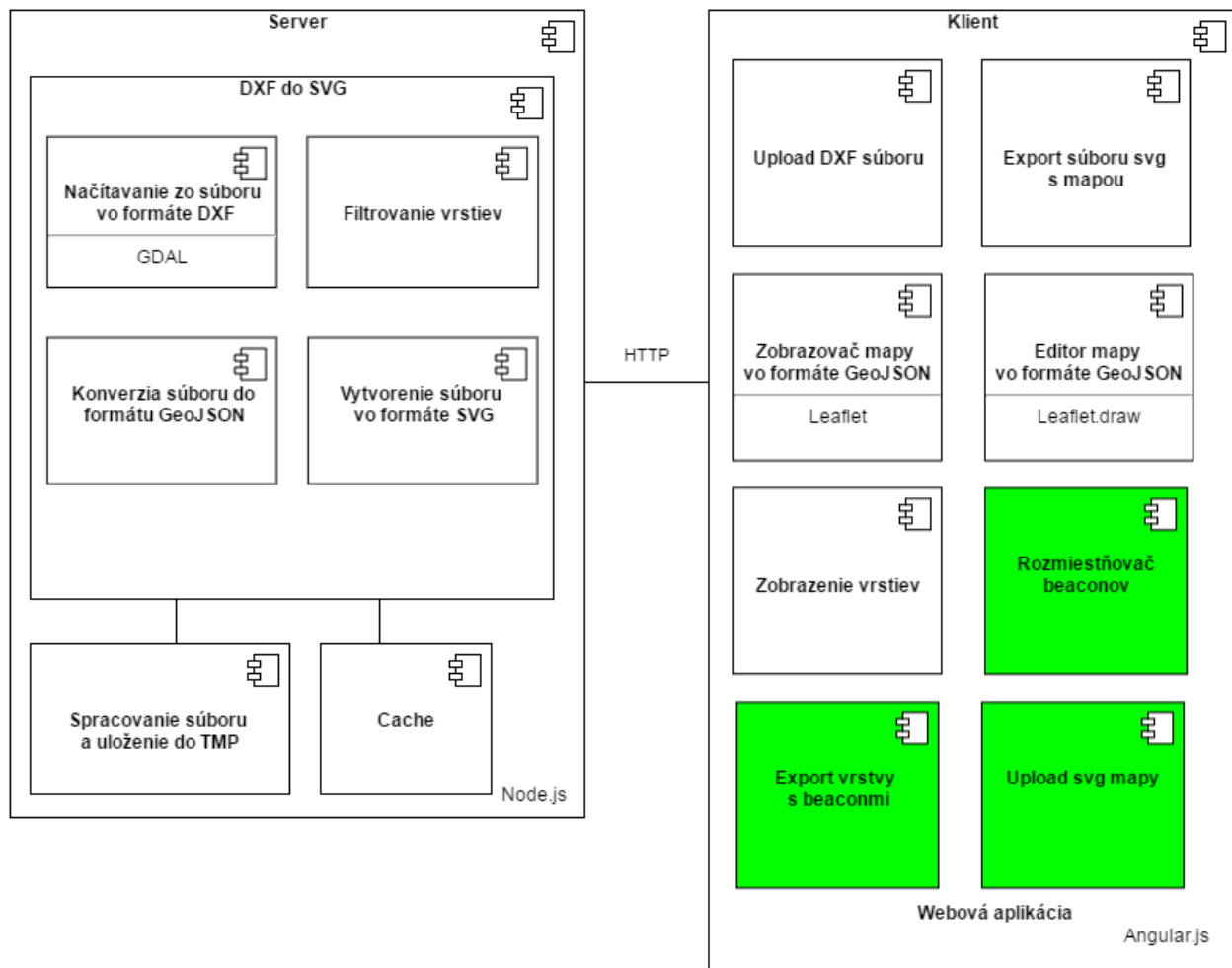
- Nahrávanie súboru vo formáte DXF - spracováva požiadavku klienta na upload DXF súboru. Tento súbor sa po nahraní na server uloží.
- Filtrovanie vrstiev - na základe zoznamu potrebných vrstiev, ktoré dostane v rámci požiadavky z klientskej časti aplikácie, tieto vyfiltruje a umožňuje vrátenie SVG súboru, ktorý obsahuje iba požadované vrstvy.
- Vytváranie súboru vo formáte SVG - umožňuje export súboru do formátu SVG

Okrem tohto hlavného komponentu obsahuje časť aplikácie bežiaca na serveri aj komponent na všeobecné spracovanie DXF súboru a jeho uloženie do dočasného úložiska na serveri. Nachádza sa tu tiež komponent cache, ktorý slúži ako vyrovnávací pamäť aplikácie a beží nad databázou Redis.

Druhá časť aplikácie je aplikácia bežiaci na klientovi. Táto časť aplikácie slúži na interakciu s používateľom - jej zodpovednosťou je načítavanie vstupu od používateľa, jeho validácia a prezentácia výsledkov spracovania serverom používateľovi. Obsahom klientskej časti aplikácie je viacero komponentov, pričom každý z týchto komponentov má vykonáva určitú časť procesu:

- Upload DXF súboru - umožňuje nahráť používateľov DXF súbor. Stará sa aj o validáciu nahraného súboru.
- Zobrazovač mapy vo formáte SVG - umožňuje zobrazit' obrázok vo formáte SVG.
- Zobrazovač zoznamu vrstiev - zobrazuje používateľovi zoznam vrstiev aktuálne spracovávaného obrázku.
- Editor mapy vo formáte SVG - umožňuje pridávanie elementov do SVG obrázku a jeho editáciu.
- Export súboru s svg mapou - umožňuje export upravenej mapy vo formáte SVG.
- Upload svg mapy – umožňuje používateľovi nahráť svg mapu
- Rozmiestňovač beaconov – navrhne používateľovi optimálne rozmiestnenie beaconov na základe zadaných parametrov
- Export vrstvy s beaconmi – umožňuje používateľovi stiahnuť svg mapu obohatenú o vrstvu s rozmiestnenými beaconmi

Táto časť aplikácie je napísaná prevažne v jazyku JavaScript a frameworku AngularJS.



Obr. č. 1: Diagram komponentov celej našej aplikácie na úpravu máp. Zelená farba označuje komponenty pridané v letnom semestri, všetky ostatné boli vytvorené v zimnom semestri.

4.6 Prípady použitia pre modul editora

Identifikovali sme tri základné prípady použitia (obr. č.2):

- Načítanie zdrojových dát
- Vytvorenie mapy
- Uloženie vytvorenej mapy

Prvý prípad použitia, s ktorým sa používateľ dostane do kontaktu je Načítanie zdrojových dát. S jeho pomocou používateľ môže načítať zdrojové dáta zo súboru vo formáte DXF do našej aplikácie. S týmito dátami bude následne pracovať.

Vytváranie mapy je pokryté druhým hlavným prípadom použitia, Vytvorením mapy. Tento prípad použitia sme rozdelili na niekoľko menších. Používateľovi umožňujeme správu zobrazovaných vrstiev (zobraziť len zvolené vrstvy, ostatné odstrániť), správu jednotlivých elementov na obrázku (manipuláciu s bodmi, čiarami, polygónmi, umožňujeme pridávať nové elementy i odstraňovať existujúce) a správu metadát ku jednotlivým elementom (napr. pridávanie číslovania miestností, umiestňovanie beacon-ov na mapu, ...).



Obr. č. 2: Diagram prípadov použitia pre modul editora.

4.7 Prípady použitia pre modul rozmiestňovania beaconov

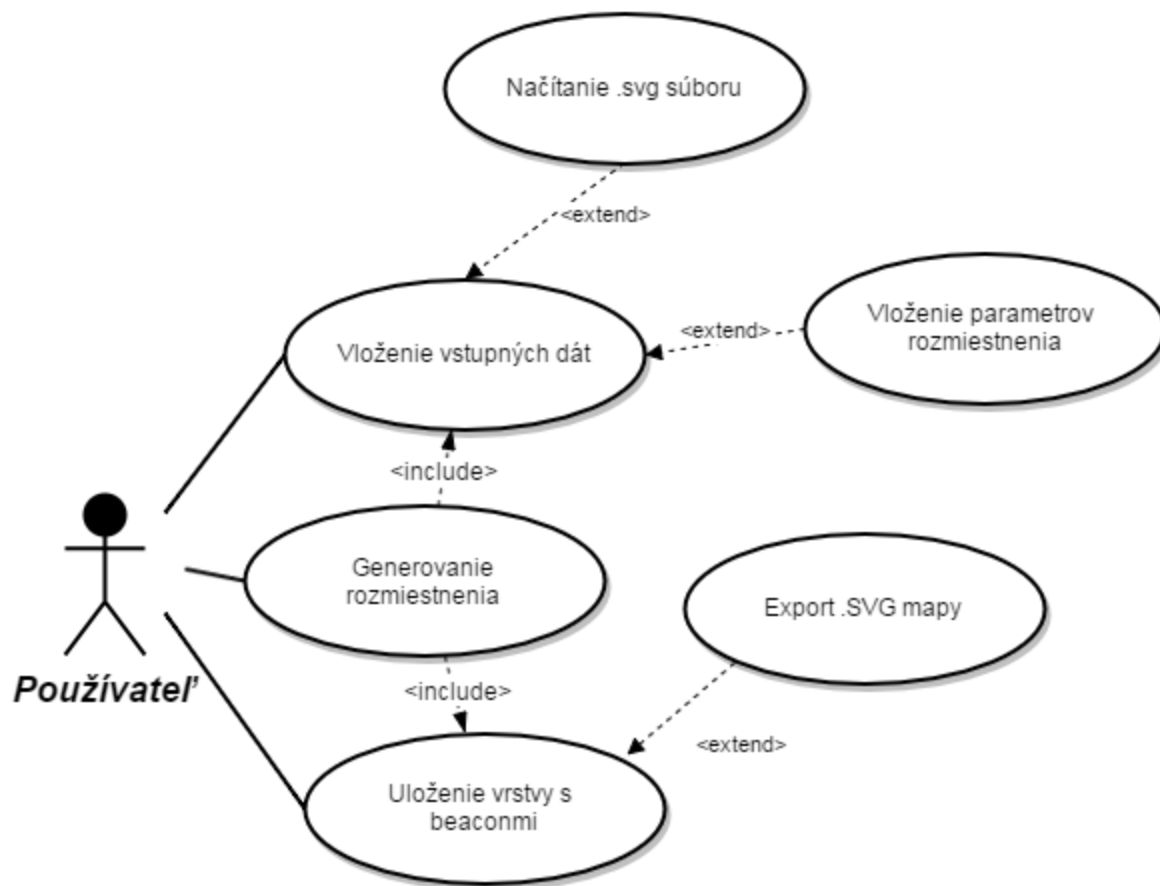
Pre modul na rozmiestňovanie beaconov sme identifikovali tri základné prípady použitia (obr. č.3):

- Vloženie vstupných dát
- Generovanie rozmiestnenia
- Uloženie vrstvy s beaconmi

Prvý prípad použitia, s ktorým sa používateľ dostane do kontaktu je Vloženie vstupných dát. Používateľ nahrá svg súbor, a zadá želané parametre pre rozmiestnenie beaconov. S týmito dátami sa bude následne ďalej pracovať.

Druhý prípad použitia zahŕňa automatické generovanie rozmiestnenia. Po kliknutí na tlačidlo sa používateľovi zobrazí odporúčané rozmiestnenie.

Posledný prípad použitia umožňuje používateľovi uložiť mapu s vrstvou vygenerovaného rozmiestnenia beaconov na jej ďalšie použitie.



Obr. č. 3: Diagram prípadov použitia pre modul rozmiestňovanie beaconov.

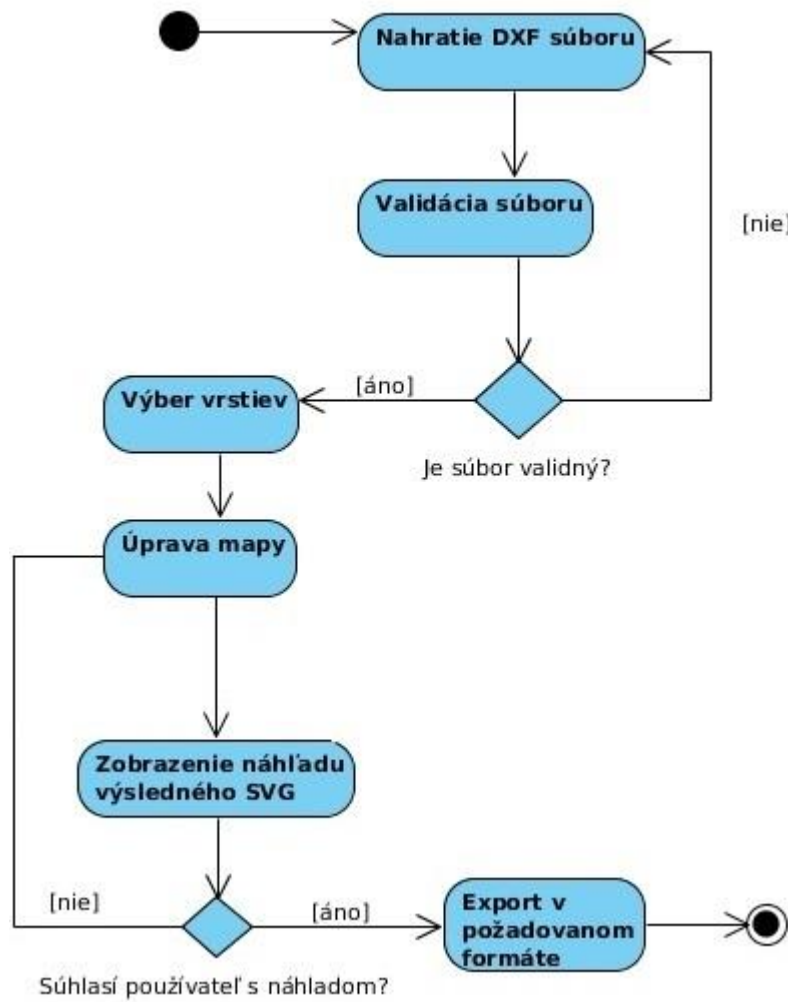
4.8 Postup pri obsluhu modulu editora

Proces úpravy mapy začína nahraním súboru vo formáte DXF do aplikácie (obr. č.4). Nahraný súbor je validovaný a v prípade, ak nie je schválený (pomýlil sa pri výbere súboru), používateľ sa vracia na začiatok procesu. Znovu nahráva súbor, ktorý musí byť v správnom formáte. Validácia pozostáva z kontroly prípony súboru. Pokiaľ je prípona správna, ale obsah súboru nezodpovedá súboru typu dxf, server vráti chybu.

Ak je súbor v poriadku, proces pokračuje výberom vrstiev, ktoré chce mať vo výslednom súbore zobrazené. Tento výber sa vykonáva kvôli veľkému počtu vrstiev a teda aj samotných dát, ktoré zväčša obsahuje nahraný DXF súbor. Mnohé z týchto vrstiev nie sú žiadané vo výslednom súbore. Na server sa odosiela zoznam vrstiev, ktoré sa majú používateľovi zobraziť, a server vráti svg s výslednými vrstvami.

Po výbere vrstiev má používateľ možnosť upraviť prvky na mape - môže pridávať prvky (steny, dvere, body) a označiť jednotlivé miestnosti alebo body (body reprezentujú napríklad beacon zariadenia). Úpravy na mape sú vykonávané na klientskej časti aplikácie. Po dokončení úprav sa používateľovi zobrazí náhľad výsledného obrázku. Používateľ náhľad skontroluje. Ak je mapu ďalej potrebné upravovať, proces sa vracia do fázy úpravy obrázku. V prípade, ak je používateľ s

náhľadom spokojný, proces pokračuje exportom mapy do požadovaného formátu (SVG, PNG, JPEG, JSON) a následne proces končí.



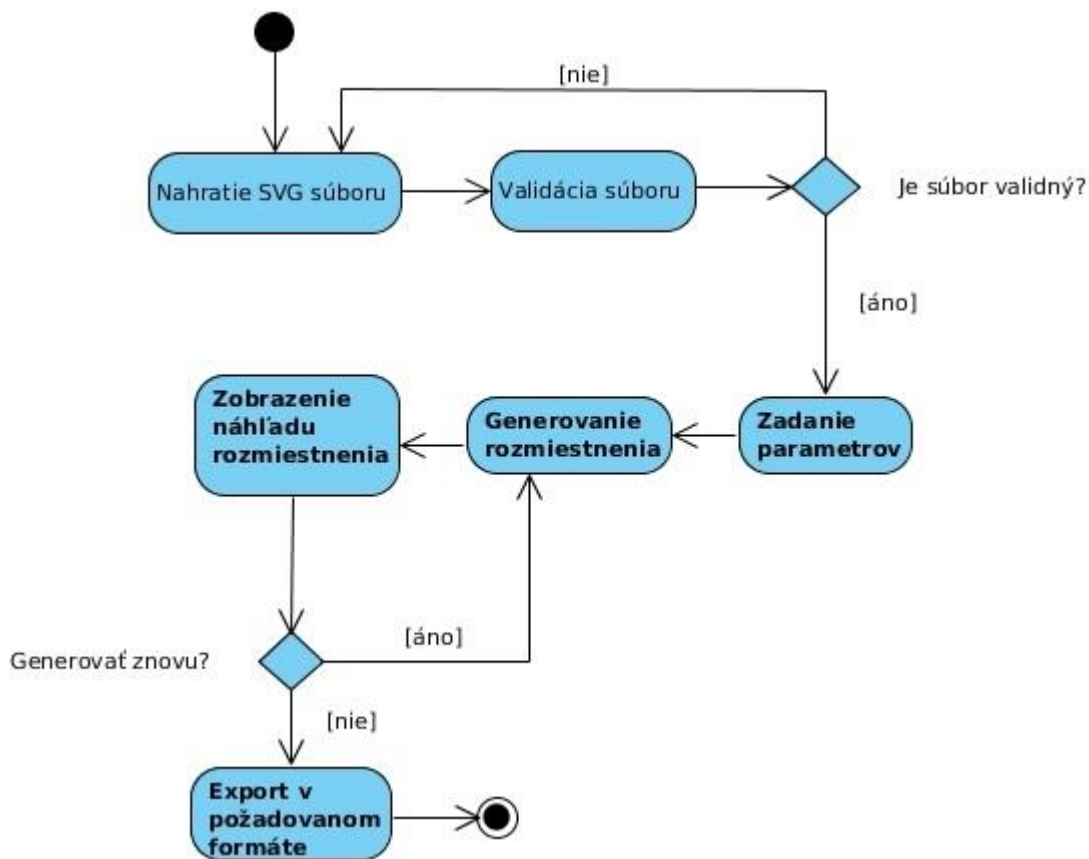
Obr. č. 4: Diagram činností modulu editora.

4.9 Postup pri obsluhu modulu rozmiestňovania beaconov

Proces rozmiestnenia začína nahraním súboru vo formáte svg do aplikácie (obr. č.5). Nahraný súbor je validovaný a v prípade, ak nie je schválený (nie je fomátu svg), používateľ musí nahráť nový súbor v správnom formáte.

V prípade validného súboru dostane používateľ možnosť zadať parametre pre algoritmus rozmiestnenia. Po zadaní týchto parametrov môže kliknúť na tlačidlo, ktoré vygeneruje optimálne rozmiestnenie. Toto rozmiestnenie sa následne graficky zobrazí na mape. Používateľ má možnosť znova vygenerovať rozmiestnenie, v prípade že sa mu aktuálne nepozdáva.

V opačnom prípade proces pokračuje exportom obohatenou o vrstvu beaconov a následne proces končí. Celý proces návrhu rozmiestnenia funguje na klientskej strane aplikácie a neprebíha komunikácia so serverom.



Obr. č. 5: Diagram činností modulu rozmiestňovanie beaconov.

5 Úprava Virtual FIIT

Prvý šprint bol venovaný úpravám aplikácie Virtual FIIT, na ktorej pracovali predchádzajúce tímy. Išlo o opravu chýb, ktoré sme objavili my alebo boli nahlásené funkciou nahlasovanie chýb v aplikácií, a pridávanie nových funkcionalít do aplikácie.

Každý testoval svoju pridelenú úlohu vo svojom prostredí. Keďže nie všetci ho máme rovnaké, uvádzame pre každého člena tímu to jeho:

- Ondrej Čerman:
 - Operačný systém Ubuntu Mate,
 - internetový prehliadač: Firefox,
 - monitor: 1920x1080,
 - mobil: ASUS Zenfone 2 ZE551ml,
- Matus Demko:
 - Operačný systém Linux Mint,
 - internetový prehliadač: Google Chrome XY,
 - monitor: 1920x1080,
 - tablet: Android 4.0.4, obrazovka 800x600,
- Juraj.Marák:
 - Operačný systém Ubuntu 14.04,
 - internetový prehliadač: Google Chrome,
 - monitor: 1920x1080,
 - mobil: Huawei P8 Lite,
- Samuel.Pecár:
 - Operačný systém Ubuntu 14.04,
 - internetový prehliadač: Chrome/Mozilla,
 - monitor: 1600x900,
 - mobil: Huawei P8 Lite,
- Gabriel.Takács:
 - Operačný systém OsX El Capitan,
 - internetový prehliadač: Google Chrome/Safari,
 - monitor: 2560x1600,
 - mobil: iPhone6,
- Miroslav Takács:
 - Operačný systém Linux Mint 17.2,
 - internetový prehliadač: Google Chrome 46.0.2490.86 m, Mozilla Firefox 34.0.5,
 - monitor 1366x768,
 - mobil: Android 4.2, obrazovka 480x800.

5.1 Oprava chýb

5.1.1 Zmiznutie prihlasovacieho okna po prihlásení

INDOORNAV-8, Ondrej Čerman

Analýza Po prihlásení bol stále viditeľný formulár na prihlásenie a po odhlásení bol viditeľný box s najbližšou hodinou namiesto formulára na prihlásenie.

Riešenie Problém bol v tom, že všetky zobrazenia sú automaticky ukladané do cache pamäte. Vyriešilo sa to vymazaním cache pamäte po prihlásení a zakázaním cache na stránke prihlásenia. Pri odhlasovaní vymazanie cache pamäte na tento problém nestačí, pretože boxy sa nachádzajú na práve zobrazovanej stránke a nedá sa žiadnym rozumným spôsobom tú stránku prekresliť (bez cache).

Našli sa dve riešenia: obnoviť stránku, alebo vypnúť cache na hlavnej stránke. Zvolilo sa riešenie č. 1, obnoviť stránku po odhlásení.

Testovanie Funkčnosť bola overená aj vo webovej aj v mobilnej verzii.

5.1.2 Miestnosť študijného oddelenia

INDOORNAV-10, Samuel Pecár, Miroslav Takács

Analýza Našli sme chybu pri zobrazení detailu o študijnom oddelení, kde sa nezobrazovalo číslo miestnosti a po kliknutí na túto voľbu došlo k presmerovaniu na hlavnú stránku.

Riešenie Manuálne sme vložili číslo miestnosti študijného oddelenia do časti, ktorá parsuje informácie z AIS. Nebolo možné spraviť automatické načítanie tejto informácie, nakoľko takáto informácia sa tam priamo a jednoznačne nezobrazuje.

Testovanie Po zobrazení detailu študijného oddelenia sa zobrazí číslo miestnosti študijného oddelenia a po kliknutí na túto možnosť je používateľ presmerovaný na mapu, kde sa farebne zobrazí študijné oddelenie.

5.1.3 Chybové hlásenie v detaile vyučovacích hodín

INDOORNAV-11, Gabriel Takács

Analýza Problém spočíval v tom, že načítavanie dát z AIS trvalo v niektorých prípadoch dlhšie, čo spôsobovalo, že namiesto reálnych dát sa v detaile vyučovacích hodín zobrazovali kučeravé zátvorky a nastávala chyba AngularJS.

Riešenie Keďže dáta boli väčšinou dostupné už v lokálnom úložisku prehliadača, nebolo potrebné dáta zakaždým naťahovať z backend-u. V prípade, ak je to možné, tak sa dáta načítavajú bez potreby požiadavky na backend.

Testovanie Funkčnosť som overil vo webovej aj mobilnej verzii aplikácie.

5.1.4 Nefunkčné načítanie dát z „iTranzit“ a jedálne

INDOORNAV-14, Samuel Pecár, Ondrej Čerman

Analýza Po nasadení na virtuálny server a pripojení k aplikácii nebolo možné načítať údaje o jedálňach ani spojeniach MHD.

Riešenie Zistili sme, že chyba je vo validácii požiadaviek prichádzajúcich na server. Táto funkcionálna bola na produkčnom serveri ohraničená znakmi pre komentovanie. Upravili sme

existujúcu validačnú funkciu a doplnili novú tak, aby bolo možné korektne skontrolovať prichádzajúce požiadavky.

Testovanie Po spustení aplikácie, ktorá sa pripája na virtuálny server je už funkčné zobrazenie údajov o jedálňach a spojeniach MHD.

5.1.5 Tlačidlo späť v mapách

INDOORNAV-17, Miroslav Takács

Analýza V mapách by tlačidlo späť malo používateľa vrátiť na predchádzajúcu stránku, nie na predchádzajúcu záložku (mapu).

Riešenie Po kliknutí na tlačidlo s číslom poschodia sa vymaže z histórie aktuálne zobrazené poschodie, takže tlačidlo späť vráti používateľa na predchádzajúcu stránku. Pri kliknutí na tlačidlo späť na mape okolia sa používateľ vráti späť o 2 navštívené stránky.

Testovanie Funkčnosť riešenia bola overená vo webovej verzii, tlačidlo späť vracia používateľa na stránku otvorenú pred oknom s mapami.

5.1.6 RSS správy

INDOORNAV-19, Samuel Pecár, Miroslav Takács

Analýza Našli sme chybu, kde po kliknutí na RSS správy v časti linky, nedošlo k presmerovaniu.

Riešenie V súbore, v ktorom sú uchovávané všetky odkazy chýbal odkaz na RSS správy.

Testovanie Po kliknutí na RSS správy v časti linky dôjde k presmerovaniu na stránku s RSS správami.

5.1.7 Kontrola pripojenia na internet pre načítaním MHD

INDOORNAV-22, Matúš Demko

Riešenie Pred načítaním dát kontrolujem pripojenie na internet funkciou `IsConnectedToInternet():Bool`. Kontrola funguje iba na zariadeniach (resp. nefunguje na internetových prehliadačoch). Kontrola prebieha pomocou typu pripojenia `window.connection.type`. Ak je tento typ `Connection.NONE` (nie je pripojenie na internet) ukáže sa používateľovi dialóg s upozornením.

Testovanie Funkčnosť bola overená vo webovej verzii.

5.1.8 Označenie miestností budovy FEI

INDOORNAV-10, Ondrej Čerman

Analýza Problém spočíva v tom, že miestnosti z FEI majú zlé (neznáme) číselné označenie v AIS. Správne označenie je však v časti názve miestnosti.

Riešenie Číslo miestnosti na FEI STU berieme z názvu miestnosti.

Testovanie Funkčnosť bola overená vo webovej verzii.

5.2 Pridávanie nových funkcií

5.2.1 Zobrazenie rozlíšenia obrazovky vo funkcii „bug report“

INDOORNAV-6, Matúš Demko

Riešenie Implementované jednoduchým pridaním objektu *screenResolution* ako parametra do funkcií *bugrep.send()* a *serverapi.sendBugreport()*. Objekt *screenResolution* obsahuje dva atribúty *width* a *height*.

Testovanie Funkčnosť bola overená vo webovej verzii.

Používateľský príbeh Vývojári aplikácie Virtual FIIT chcú vidieť na akom displeji bola chyba nahlásená aby mohli identifikovať o aký typ zariadenia ide.

5.2.2 Zobrazenie poznámky pod rozvrhom ako v AISE

INDOORNAV-9, Samuel Pecár

Analýza Zistili sme, že informácie, ktoré sa nachádzajú po rozvrhom ako poznámky ani číslo poznámky pri rozvrhu nie sú parsované.

Návrh Pre parsovanie tabuľky poznámok nebude potrebné výrazne zasahovať do funkcionality, iba doplniť v existujúcej funkcii parsovanie čísla poznámky a na záver aj parsovanie tabuľky poznámok.

Riešenie Doplnili sme parsovanie čísla poznámky počas parsovania rozvrhu. Následne zisťujeme, či niektorý z predmetov mal poznámku a ak áno, tak parsujeme tabuľku poznámok, ktorá sa tam inak nenachádza.

Zmenili sme spôsob odpovede na požiadavku, kedy sa JSON zobrazujúci rozvrh presunul o úroveň nižšie a na rovnakú úroveň sme pripojili tabuľku s poznámkami. Táto zmena si vyžadovala aj zmenu vo FE časti, aby sa správne zobrazil rozvrh a následne aj tabuľka s poznámkami.

Testovanie Zobrazenie rozvrhu aj s poznámkami bolo otestované vo webovej verzii aplikácie.

Používateľský príbeh Študent si chce pozrieť v rozvrhu, ktorý týždeň má cvičenie.

6 Tvorba modulu pre editáciu mapy

Prvý semester je venovaný vytváraniu modulu pre editáciu mapy. Tento modul má slúžiť na nahranie mapy na server a jej následnú konverziu na použiteľný formát. Ďalším krokom je stiahnutie skonvertovanej mapy a jej zobrazenie. Používateľ následne zruší označenie vrstiev, ktoré chce odfiltrovať. Potom nasleduje definovanie miestností a pozícií .

6.1 Zobrazenie mapy v editore

INDOORNAV-25, Samuel Pecár

Používateľský príbeh Používateľ, ktorým je v tomto prípade majiteľ alebo administrátor budovy, chce nahrať DXF súbor, ktorý bude môcť upravovať alebo ďalej spracovávať za účelom získania správneho rozmiestnenia beacon-ov a použitia takto upravenej mapy na zobrazenie v mobilnej aplikácii.

Analýza Cieľom je vytvoriť novú stránku, ktorá umožní nahranie súboru. Tento súbor bude potrebné odoslať na server, kde ho ďalší skript spracuje a následne zobrazí.

Návrh Vytvoríme klient-server aplikáciu, ktorá umožní nahranie DXF súboru na strane klienta a klienta, ktorý sa následne odošle a spracuje na strane servera. Po spracovaní server uloží extrahované vrstvy do cache a vloží do cookies id. Následne môže používateľ pomocou id v cookies a zoznamu vrstiev doťahovať do klientskej časti dáta o vrstvách, ktoré mu budú zobrazené.

Riešenie Vytvorili sme časť aplikácie, ktorá umožňuje nahrať DXF súbor, ktorý je odoslaný na server, kde je spracovaný. Používateľ môže požiadať o zobrazenie požadovaných vrstiev, ktoré sú doťahované zo servera. Aplikácia následne zobrazí tieto vrstvy vo forme mapy

Testovanie Na testovanie sme použili plán budovy FEI. Súbor bol po načítaní na server konvertovaný na zobraziteľný JSON formát, ktorý bol odoslaný a úspešne zobrazený na klientskej strane. Po zmene zobrazenia vrstiev bol plán budovy aktualizovaný. Zároveň sme na testovanie aplikácie použili automatizované testy.

6.2 Nasadenie prostredia na virtuálny server

INDOORNAV-27, Ondrej Čerman

Používateľský príbeh Používateľ aplikácie si nechce spúšťať vlastný server vždy keď chce vytvoriť novú mapu.

Analýza Bolo potrebné nasadiť celé prostredie na virtuálny server

Riešenie Na server bol vytvorený skript na sťahovanie zdrojových kódov a spúšťacích skriptov. Na server boli pridané testovacie úlohy.

Testovanie Testovanie prebehlo na našom virtuálnom serveri.

6.3 Vytvorenie základného jadra modulu

INDOORNAV-28, Samuel Pecár

Riešenie Na vytvorenie základného jadra sme použili express-generator, ktorý vygeneroval serverovú časť aplikácie a Yeoman, ktorý pomocou Angular generátora, vygeneroval základnú štruktúru klientskej časti aplikácie.

6.4 Upload súboru

INDOORNAV-29, Gabriel Takács

Používateľský príbeh Používateľ, ktorým je v tomto prípade majiteľ alebo administrátor budovy, chce nahrať DXF súbor, ktorý bude môcť upravovať alebo ďalej spracovávať za účelom získania správneho rozmiestnenia beacon-ov a použitia takto upravenej mapy na zobrazenie v mobilnej aplikácii.

Analýza Cieľom je vytvoriť novú stránku, ktorá umožní nahranie súboru. Tento súbor bude potrebné odoslať na server, kde ho ďalší skript spracuje a následne zobrazí.

Návrh Bude potrebné vytvoriť novú stránku (“obrazovku”), ktorá bude umožňovať nahrávanie súboru. Pri nahrávaní bude potrebné testovať, či je nahrávaný súbor správneho typu (podľa prípony, ktorá musí byť “.DXF”). Po nahraní súboru bude nasledovať presmerovanie na zobrazenie mapy.

Riešenie Vytvoril som novú obrazovku, ktorá pozostáva z UploadController-u (súbor *controllers/upload.js*) a upload view (*views/upload.html*). Na upload súborov som použil modul tretej strany *ng-file-upload*. Tento modul umožňuje jednoduchšie nahrávanie súborov.

Na validáciu typu nahrávaného súboru som napísal vlastnú funkciu, ktorá validuje typ súboru na základe jeho prípony. Presmerovanie na zobrazenie súboru som vytvoril pomocou promise, ktorá sa viaže na úspešné nahranie súboru.

6.5 Zobrazenie mapy ako SVG

INDOORNAV-30, Miroslav Takács

Používateľský príbeh Používateľ aplikácie chce vidieť interaktívne grafické zobrazenie plánu budovy, ktoré načítal z DXF súboru.

Analýza Na zobrazenie plánu budovy budeme používať knižnicu *d3-floorplan*. Cieľom je zobrazit' dáta získané v JSON súbore získaného zo servera.

Návrh Bude potrebné vytvoriť obrazovku, ktorá bude umožňovať zobrazenie mapy. Zobrazenie mapy bude prepojené s výberom vrstiev, ak používateľ zmení vrstvy ktoré sa majú zobrazit', pošle sa požiadavka na server a spätne získané dáta sa zobrazia.

Riešenie Do aplikácie bola importovaná knižnica *d3-floorplan*, ktorá umožňuje zobrazenie plánu budovy zadaného v JSON formáte. Túto knižnicu nebolo možné pridať pomocou nástroja bower, preto bola do projektu pridaná manuálne. Knižnica *d3* bola pridaná cez nástroj bower.

Na server sa odosiela zoznam vrstiev, ktoré má plán obsahovať. Následne pomocou promise sú tieto vrstvy zobrazené.

Testovanie Na testovanie sme použili plán budovy FEI. Súbor bol po načítaní na server konvertovaný na zobraziteľný JSON formát, ktorý bol odoslaný a úspešne zobrazený na klientskej strane. Po zmene zobrazenia vrstiev bol plán budovy aktualizovaný.

6.6 Načítanie obsahu súboru do aplikácie

INDOORNAV-31, Samuel Pecár

Riešenie Po načítaní súboru na strane servera, tento súbor spracovali pomocou modulu DXF-parsing. Následne sú tieto údaje odoslané do funkcií na konverziu formátu a na filtrovanie vrstiev a nakoniec uložené do cache.

6.7 Zobrazenie zoznamu vrstiev

INDOORNAV-32, Matúš Demko

Používateľský príbeh Používateľ aplikácie si chce vybrať iba niektoré vrstvy na zobrazenie.

Analýza Na zobrazenie zoznamu vrstiev je nutné nahráť súbor a prečítať z neho na serveri všetky vrstvy.

Návrh Vrstvy získané z nahratého súboru na serveri budú zobrazené v okne pre editovanie mapy.

Riešenie V obrazovke pre zobrazenie mapy bolo vytvorené miesto pre zobrazenie vrstiev. Vrstvy sú získavané metódou GET návrhového štýlu REST, ktorá vráti dáta vo formáte JSON.

Testovanie Na server bol nahraný súbor *fei.DXF*. Následne sa zobrazila nová obrazovka so všetkými vrstvami ktoré obsahuje tento súbor.

6.8 Konverzia JSON súboru do formátu zobraziteľného na FE

INDOORNAV-34, Samuel Pecár, Juraj Marák

Riešenie Po spracovaní obsahu súboru sú údaje skonvertované do formátu načítateľného pre modul d3-floorplan, ktorý používame na zobrazenie obsahu mapy. Následne sú rozdelené jednotlivé vrstvy a samostatne cachované do Redis databázy.

6.9 Získanie informácií zo súboru DXF

INDOORNAV-42, Samuel Pecár, Juraj Marák

Používateľský príbeh Používateľ chce mať podrobnejšie a čitateľnejšie zobrazené elementy.

Analýza Pre získanie podrobnejších a kvalitnejších informácií je potrebné využiť knižnicu GDAL.

Návrh Bude potrebné implementovať funkcionality modulu node-gdal, ktorý umožní spracovanie súboru vo formáte DXF a extrahuje objekty vo formáte GeoJSON.

Riešenie Knižnica node-gdal bola implementovaná do aplikácie a zároveň bol vytvorený modul, ktorý vytvorí z požadovaného súboru súbor vo formáte SVG, ktorý je zobraziteľný na klientskej strane aplikácie.

Testovanie Testovanie bolo uskutočnené pomocou testovacieho súboru, z ktorého bol vytvorený výstup vo formáte SVG. Tento výstup sa zhodoval s požadovanými informáciami v testovacom súbore DXF.

6.10 Zobrazenie dát v podobe mapy

INDOORNAV-43, Miroslav Takács, Matúš Demko

Používateľský príbeh Používateľ chce mať zobrazené dáta zo súboru DXF v podobe čitateľnej mapy, preto je potrebné použiť funkcie na zobrazenie mapy z projektu OFP viewer.

Analýza Na zobrazenie plánu budovy budeme používať knižnicu ofp-viewer. Cieľom úlohy je zobrazenie dát získaných zo servera.

Návrh Bude potrebné zmeniť obrazovku na zobrazenie mapy. Zobrazenie mapy bude prepojené s výberom vrstiev, ak používateľ zmení vrstvy, ktoré sa majú zobrazit', pošle sa požiadavka na server a spätne získané dáta sa zobrazia. Po zmene vrstiev bude vhodné zachovať mierku priblíženia a polohy na mape.

Riešenie Do aplikácie bol importovaný projekt ofp-viewer, pomocou ktorého je zobrazený plán budovy. Tento plán používateľ predtým odoslal na server ako dxf súbor a server mu odoslal zoznam vrstiev. Server následne odoslal používateľovi plán budovy s označenými vrstvami.

Na server sa odosiela zoznam vrstiev, ktoré má plán obsahovať. Následne pomocou promise sú tieto vrstvy zobrazené.

Testovanie Na testovanie sme použili plány budov FEI a FIIT. Súbor bol po načítaní na server konvertovaný, bol odoslaný a úspešne zobrazený na klientskej strane. Po zmene zobrazených vrstiev bol plán budovy aktualizovaný. Testovanie úspešne prebehlo na prehliadačoch Google Chrome a Mozilla Firefox.

6.11 Zobrazovanie stavu procesu úpravy mapy

INDOORNAV-44, Gabriel Takács

Používateľský príbeh Používateľ chce byť informovaný v akom stave procesu úpravy mapy sa nachádza, preto je potrebné zmeniť systém zobrazovania jednotlivých častí aplikácie.

Analýza Na zobrazovanie stavu procesu budeme používať nástroje AngularJS.

Návrh Na uloženie aktuálneho stavu aplikácie budeme používať AngularJS Factory.

Riešenie Aktuálny stav procesu úpravy mapy si jednoducho ukladáme pomocou AngularJS Factory appState a zobrazujeme ako "breadcrumbs" v hornej časti používateľského prostredia. Do appState budeme v budúcnosti ukladať celý stav aplikácie.

Testovanie Zobrazovanie aktuálneho stavu procesu úpravy mapy sme otestovali pri úprave mapy budovy FEI. Code review spravil Ondrej Čerman.

6.12 Zobrazovanie zoznamu vrstiev a zobrazovanie obsahu vrstiev

INDOORNAV-51, Gabriel Takács, Matúš Demko, Miroslav Takács

Používateľský príbeh Používateľ chce mať možnosť vybrať, ktoré z vrstiev súboru chce zobrazit', preto je potrebné mu v jednoduchom formulári zobrazit' zoznam vrstiev, ktoré sa po zvolení tlačidla "Zobrazit'", zobrazia v mape.

Analýza V druhej fáze procesu úpravy mapy zobrazíme bočný panel v pravej časti mapy. Tento panel bude štandardne vysúvateľný, avšak bude ho možné skryť, aby bola plocha viditeľnej mapy väčšia. Panel bude obsahovať zoznam vrstiev, ktoré mapa obsahuje. Zoznam bude zobrazený ako zoznam "checkbox" elementov. Používateľ bude mať možnosť zaškrtnúť vrstvy, ktoré chce mať na mape viditeľné. Štandardne budú zaškrtnuté všetky vrstvy.

Návrh Na realizáciu tejto úlohy použijeme najmä framework jQuery, samozrejme s využitím prostriedkov AngularJS na volanie servera. Pri každej zmene stavu (zaškrtnutí alebo odškrtnutí ktoréhokoľvek checkboxu) sa zavolá server, ktorý vráti URL novo-vygenerovaného SVG súboru, pričom tento súbor zobrazíme.

Riešenie Pri zobrazovaní zoznamu vrstiev sme nepoužili žiadne neštandardné alebo zaujímavé riešenie - zobrazujeme zoznam vrstiev, ktoré nám vráti server po nahrať mapy. Pri zmene hodnoty ktoréhokoľvek checkboxu sa zavolá metóda *getLayersListData*. Táto metóda volá server, ktorý vygeneruje nový SVG súbor v závislosti od zoznamu zaškrtnutých vrstiev a vráti nám URL tohto súboru. Tento SVG súbor následne zobrazujeme pomocou knižnice D3.

Testovanie Zobrazovanie aktuálneho stavu procesu úpravy mapy sme otestovali pri úprave mapy budovy FEI. Code review spravil Miroslav Takács.

6.13 Problém s CORS

INDOORNAV-52, Ondrej Čerman

Analýza CORS pravidla na serveri boli nastavené len na localhost, čo spôsobuje problém pri automatickom nasadení.

Riešenie Na serveri sa vytvoril whitelist pôvodcov. Po prístupe na server sa overí, či sedí pôvodca s tým vo whitelist-e. Pokiaľ áno, tak sa odošle správna hlavička CORS.

Testovanie Funkčnosť bola overená vo webovej verzii.

6.14 Problém s Gruntom

INDOORNAV-61, Ondrej Čerman

Problém Nefunguje príkaz grunt. Pri vytváraní 'dist' sa neinkludujú všetky javascripty. Requesty z FE sú napevno, a pri nasadzovaní na server je nutné manuálne prepisovať zdrojové kódy.

Analýza Grunt nefungoval korektne, pretože mu chýbali niektoré balíky. Tie však nešli nainštalovať cez *npm install* ani cez *npm install <balík>*, kvôli nevyriešeným závislostiam. Po dlhom čase sa zistilo, že ide o bug v npm3 a to aj v najnovšej verzii (v čase riešenia bugu). Nebol nájdený žiaden fix, ale upravením súboru *package.json* sa podarilo tento bug obísť. Po tomto sa grunt rozbehol ďalej, ale stále padal v tasku *jshint*. Problém je nekorektne vložená knižnica *d3.floorplan*. Z rovnakého dôvodu sa nezahŕňa do 'dist'. Tento problém sa ďalej neriešil, keďže *d3.floorplan* je v aplikácii pravdepodobne iba dočasne. Ďalší problém bol v tasku *imagemin*. Zistilo sa, že problém nie je v konfigurácii samotného tasku, ani programe *imagemin*, ale je to bug v balíku *grunt-contrib-imagemin*, konkrétne <https://github.com/gruntjs/grunt-contrib-imagemin/issues/208>

Riešenie Problémy s gruntom:

- Upravenie *package.json*.
- Requesty z FE: prepis requestov

Testovanie Funkčnosť bola overená vo webovej verzii.

6.15 Vytvorenie homepage a informačnej stránky

INDOORNAV-69, Samuel Pecár

Problém Pri otvorení webovej stránky je používateľ vyzvaný hneď k načítaniu súboru.

Analýza Je potrebné aby bola vytvorená domovská stránka a stránka, ktorá by informovala o projekte.

Riešenie Bola vytvorená domovská stránka a informačná stránka, ktorá informuje používateľa o projekte.

Testovanie Funkčnosť bola overená vo webovej verzii.

6.16 Zachovať zoom a pozíciu mapy pri zmene vrstiev

INDOORNAV-68, Miroslav Takács

Používateľský príbeh Používateľ chce zachovať posunutie a priblíženie mapy pri zmene vrstiev.

Riešenie Pred zmenou ktoréhokoľvek checkboxu sa zapamätá hodnota posunutia mapy a hodnota priblíženia. Pri načítaní nových vrstiev sa táto nová vrstva zobrazí, a obnovia sa hodnoty posunutia a zoomu.

Testovanie Zachovanie posunutia a priblíženia sme otestovali s mapou budovy FEI. Code review spravil Gabriel Takács.

6.17 Stiahnutie súboru svg

INDOORNAV-50, Ondrej Čerman

Používateľský príbeh Používateľ chce mať možnosť stiahnuť súbor s dátami mapy vo formáte svg.

Analýza Aktuálne server dokázal vytvárať z dxf geojson, ktorý slúži ako zdroj dát mapy, avšak základné funkcie pre tvorbu svg sú v serveri, pretože boli používané v prvotnom štádiu projektu.

Návrh Vytvorenie funkcie na serveri, ktorá prekonvertuje geojson do svg + vytvorenie prislúchajúceho rozhrania na frontende.

Riešenie Po úpravy mapy sa extrahuje geojson do zdieľanej premennej. Po výbere na export dát sa na server odošle geojson a informácie o rozmeroch mapy a server z týchto dát vygeneruje svg, ktoré odošle v base64 formáte. Po prijatí svg sa vytvorí odkaz na stiahnutie.

Testovanie Testovanie prebehlo na mapách basement a test.

6.18 Indikátor aktivity - koliesko pri načítaní DXF

INDOORNAV-65, Ondrej Čerman

Používateľský príbeh Používateľ chce mať informáciu, že prebieha načítavanie DXF.

Analýza Nahrávanie a spracovanie DXF na serveri trvá dlhú dobu, preto je potrebné, aby o tom bol používateľ informovaný. Okrem načítavania dxf môžu v budúcnosti pribudnúť ďalšie úlohy, ktoré budú dlho trvať.

Návrh Vytvorenie indikátoru aktivity pri čakaní na odpoveď servera.

Riešenie Bola vytvorená nová direktíva, ktorá zobrazí tzv. koliesko v prípade, že klient čaká na odoslané požiadavky.

Testovanie Testovanie prebehlo pri načítavaní DXF: fei, basement, test.

6.19 Oboznámenie s knižnicami OFP editor

INDOORNAV-49, Miroslav Takács

Analýza Na zobrazenie plánu budovy sme chceli použiť funkcie z projektu OFP editor. Tento editor umožňuje vytvárať a upravovať 2D objekty, a pridelovať im atribúty. Naším cieľom bolo analyzovať, či by bolo možné použiť funkcie tohto editora pre našu aplikáciu.

Pri analýze sme na začiatku odstránili nepotrebnú funkcionálnosť, napríklad vykresľovanie mapy na pozadí, nepotrebné tlačidlá a ich funkcie a iné. Ďalej sme analyzovali spôsob ukladania útvarov. Tu sme zistili, že tento formát nie je pre našu aplikáciu použiteľný, a jeho prerobenie by zabralo príliš veľa času. Táto úloha bola ukončená nájdením vhodnejšej knižnice Leaflet.draw, ktorá bola v aplikácii použitá.

6.20 Prenesenie konvertora => DXF worker

INDOORNAV-63, Juraj Marák

Používateľský príbeh Používateľ chce mať možnosť paralelného spracovávania viacerých súborov.

Analýza Spracovávanie súboru na serveri trvá určitú dobu a v súčasnosti spôsobuje blokovanie servera.

Návrh Navrhujem presunúť funkcionálnosť spracovania súboru do osobitného vlákna a umožniť tak paralelné spracovávanie viacerých požiadaviek.

Riešenie Bolo vytvorené nové vlákno, ktoré spracováva vstupný súbor.

Testovanie Vytvorené vlákno bolo otestované a je funkčné. Testovanie prebehlo pri načítavaní DXF: fei, basement, test.

6.21 Vyriešiť automatizáciu v Grunte

INDOORNAV-66, Ondrej Čerman

Problém Príkaz grunt build a iné grunt príkazy nefungujú správne.

Analýza Po pridaní sass a minifikácie výsledok grunt build obsahuje chyby. Iné príkazy ako grunt watch tiež nefungovali vždy správne.

Zistilo sa, že:

- boli odstránené niektoré tasky zo zoznamu grunt build či iných,
- niektoré tasky boli premiestnené na nesprávne miesto,
- niektoré nastavenia taskov kolidovali s taskami, ktoré sú generované pomocou useminPrepare.

Riešenie Úprava Gruntfile.js, premiestnenie css (po úprave bolo zbytočne “hlboko” v strome), kompilácia sass do css premiestnená do .tmp (keďže .tmp sa pôvodne využíval na css).

Bol vytvorený nasledujúci postup pre overenie či grunt build funguje správne:

- prejdite do priečinku s klientom,
- spustite grunt build - nemal by vyhodíť žiaden error,
- prejdite do priečinku dist a otvorte index.html vo webovom prehliadači,
- otvorte vývojárske nastavenia, prejdite do funkcií konzola a overte, že tam nie sú žiadne chyby týkajúce sa nenájdeného css, js či iného dôležitého komponentu.

6.22 Úprava čiary v mape

INDOORNAV-67, Matúš Demko

Používateľský príbeh Používateľ chce presunúť, zmeniť veľkosť alebo vymazať čiaru v načítanej mape.

Analýza Na splnenie tejto úlohy mali pôvodne byť použité knižnice z projektu OpenFloorPlan editor. Počas ich analýzy sme však zistil niekoľko nedostatkov. Následne boli analyzované a testované ďalšie možnosti implementácie úpravy čiary v mape- knižnica Leaflet.draw a editor svg-edit. Najlepšie výsledky sme dosiahli pri knižnici Leaflet.draw.

Návrh Na implementáciu úpravy čiary v mape bola zvolená knižnica Leaflet.draw.

Riešenie Pri implementácii nastali dva problémy:

- mapy spracovávané zo súborov budov FEI a FIIT sa nezobrazujú, avšak nenastala pri ich zobrazovaní ani viditeľná chyba,
- pri úprave niektorých vrstiev vzniká chyba v knižnici Leaflet.draw.

Prvý problém sa nám úspešne podarilo odstrániť- mapy mali nastavenú projekciu na príliš veľký faktor priblíženia a boli posunuté. Druhý problém však zatiaľ pretrváva a bude sa riešiť.

Testovanie Testovanie prebehlo pri načítavaní DXF: FEI, basement a súboru z budovy FIIT.

6.23 Optimalizácia výsledného SVG súboru

INDOORNAV-75, Gabriel Takács

Používateľský príbeh Používateľ chce, aby bol vygenerovaný SVG súbor čo najmenší.

Analýza Na optimalizáciu SVG súboru existuje viacero knižníc, ktoré som otestoval tak, že som do nich vložil nami generované SVG súboru. Najlepší výkon ponúkal <https://github.com/petercollingridge/SVG-optimiser.js>. Vybraný SVG optimalizátor však nie je použiteľnou knižnicou, ale hotovou JavaScript aplikáciou. Z tohto dôvodu je potrebné z neho vybrať a oddeliť potrebné komponenty.

Návrh Vytvoriť komponent na optimalizáciu výsledného SVG súboru. Súbor sa bude optimalizovať pred jeho stiahnutím.

Riešenie Vytvoril som AngularJS service “svgOptimizer”, ktorý používa funkcie zo súboru scripts/svg_optimizer.js. V tomto súbore sú vyextrahované funkcie z vybraného projektu na optimalizáciu SVG. Vzhľadom na to, že pri procese optimalizácie sa vynechávajú tagy <svg> a </svg>, tieto sú po optimalizácii prilepené späť k SVG reťazcu.

Testovanie Efektivitu optimalizácie som otestoval na DXF súboroch z FEI, pri ktorých optimalizátor zmenšil výsledný súbor o približne 42%. Optimalizovaný súbor som vizuálne porovnal s pôvodným súborom, pričom som nenašiel žiadne vizuálne zmeny.

6.24 Analýza errorov pri úprave

INDOORNAV-72, Miroslav Takács

Analýza Pri editovaní útvarov (miestností) v aplikácii nastávajú chyby a nie je možné editovať niektoré polygóny.

Riešenie Zistili sme presne kde chyba nastáva, ale neboli sme schopný bližšie identifikovať jej príčinu. Vyskúšali sme najznámejšie branche Leaflet.draw, ale v žiadnej z nich nebola táto chyba vyriešená.

6.25 Analyzovať free editory SVG

INDOORNAV-73, Matúš Demko

Analýza Cieľom bolo analyzovať editory SVG s bezplatnou licenciou, ktoré by poskytovali tieto funkcie: zjednodušenie kriviek, úprava metadát, editovanie čiar, oblastí a pod. prvkov SVG súboru. Po vyskúšaní editorov: Inkscape, libreOffice, gimp, sk1, synfig studio sme prišli k záveru, že najlepšie spĺňa požiadavky editor Inkscape. Poskytuje všetky spomínané funkcie na manipuláciu s prvkami SVG súboru. Pri hľadaní ďalších nástrojov sme našli iba také, ktoré neboli multiplatformové alebo nemali jednoduchú inštaláciu.

6.26 Analyzovať či Leaflet neumožňuje vylievačik alebo či neexistuje iný open-source nástroj

INDOORNAV-74, Juraj Marák

Analýza Hlavná vetva Leaflet.draw obsahuje niekoľko bugov a zároveň v nej absentuje niekoľko funkcionálít, ktoré by boli užitočné pre náš projekt. V rámci tejto úlohy sme analyzovali iné vetvy Leaflet.draw ako aj iné odnože Leafletu. Po skúmaní sme nenašli žiadnu vetvu, ktorá by nám poskytla nami požadovanú funkcionálnosť. Snažili sme sa o vyhľadanie iného open-source nástroja, ktorý by nám poskytol takúto funkcionálnosť a bolo ho možné integrovať do našej aplikácie, čo tiež nebolo úspešné.

7 Tvorba modulu pre automatické rozmiestnenie beaconov

Druhý semester je venovaný hlavne analýze a implementácií metódy na automatické rozmiestnenie beaconov v budove alebo miestnosti.

7.1 Podpora pre spracovanie paralelných požiadaviek

INDOORNAV-80, Samuel Pecár

Problém Pri pripojení viacerých používateľov naraz server úspešne spracuje len požiadavku na spracovanie súboru len jedného používateľa.

Analýza Riešením by mohlo byť použitie clusteru.

Riešenie Problém sme vyriešili použitím clusteru a redesignom časti kódu zodpovednej za spracovanie súboru

Testovanie Riešenie sme otestovali spustením úloh na spracovanie súborov v rovnakom čase na dvoch rôznych počítačoch.

7.2 Získanie mierky mapy

INDOORNAV-81, Samuel Pecár

Analýza Cieľom bolo analyzovať možnosti knižnice GDAL ohľadom získania metadát a získania mierky mapy. Zistili sme, že knižnica obsahuje funkcie na získanie metadát, ale súbory DXF neobsahujú informáciu o mierke mapy.

7.3 Vyrobit' poschodie FEI

INDOORNAV-82, Matúš Demko

Analýza Cieľom je zistiť ako dlho trvá vytvorenie mapy použiteľnej v aplikácii pre navigáciu vo vnútri budovy. Vstupom je súbor .DXF, ktorý obsahuje technický náčrt pôdorysu jedného poschodia budovy STU FEI.

Riešenie Vytvorenie mapy trvalo približne 1h 30m. Väčšinu času sme strávili úpravou vzhľadu hlavne pri stenách (musel som odstraňovať čiary a dvere ktoré tam boli nadbytočné, ale vrstva, v ktorej boli, by pomerne znemožnila tvoriť miestnosti "vylievačikom").

Vytvorenie miestnosti, chodieb a pridanie beaconov zabralo maximálne 30% z tohto času. Riešili sme aj menšie problémy mapy vygenerovanej nami (bola extrémne veľká).

7.4 Aktualizácia mapy FIIT

INDOORNAV-83, Miroslav Takács

Používateľský príbeh Používateľ chce mať dostupnú aktuálnu verziu mapy budovy.

Analýza Bude potrebné aktualizovať mapu FIIT. V aktuálnej chýbajú niektoré priečky, auly sú neklikateľné, niektoré miestnosti sú premenované.

Riešenie Auly boli nastavené na klikateľné, niektoré miestnosti boli premenované, podateľňa bola zrušená. Laboratóriám boli pridané atribúty *name_long* a *name_short*, ktoré obsahujú dlhý a krátky názov laboratórií (názvy nie sú kvôli ich dĺžke zobrazené priamo).

7.5 Pridať beacony do mapy FIIT

INDOORNAV-84, Gabriel Takács

Analýza Do mapy FIIT je potrebné pridať beacony.

Riešenie V programe Inkscape som pridal do mapy kruhy. Každý kruh reprezentuje jeden beacon. Ku každému beaconu som do XML pridal atribút "minor_id". ID a label som nastavil ako číslo beaconu.

7.6 Zistiť informácie od prof. Farkaša

INDOORNAV-85, Ondrej Čerman

Riešenie Od profesora Farkaša sme sa nedozvedeli žiadne relevantné informácie.

7.7 Nájsť softvér na simuláciu rozmiestnenia beaconov

INDOORNAV-86, Gabriel Takács

Analýza Hľadáme softvér na simuláciu rozmiestnenia beaconov.

Riešenie Našli sme nasledujúce programy:

- my.indoo.rs - Measurement Tool - sila beaconov sa nesimuluje, funguje ako nástroj na zobrazenie heatmapy podľa nameraných dát
- Ekahau HeatMapper - podobne, signál sa dá len namerať
- VisiWave - nevidel som možnosť na pridanie beaconov, len Wi-Fi
- Netspot - robí iba scanning existujúcich Wi-Fi
- <https://mibeacon.net/ios8-tracking-heat-map/> - vytvára heatmapu z existujúcich beaconov
- MeasurementTool - vyzeralo to nádejne, dlho sme sa s tým hrali, ale nakoniec to nerobilo to, čo sme chceli
- Urbana (ArcMap)
http://calhoun.nps.edu/bitstream/handle/10945/1292/04Dec_Boukraa.pdf?sequence=3, po čase som to vzdal, script padal aj na example dátach
- EMSlice - veľmi podrobné možnosti na nastavovanie, možnosť pridávania rôznych typov stien, simuluje šírenie signálu, neprehľadné a škaredé UI, program príliš sekal

Program EMSlice spĺňa väčšinu požiadaviek, avšak pri pridaní viacerých beaconov nezvláda spracovať tieto údaje. Niekedy aj prestane pracovať. Našli sme modul pre Matlab, ktorý simuluje šírenie signálu beaconov, avšak pri spustení výpočtu nastane výnimka.

7.8 Import SVG

INDOORNAV-87, Samuel Pecár

Analýza Je potrebné vytvoriť upload svg súboru, ktorý bude poskytnutý ďalším funkciám, ktoré extrahujú path požadovaného elementu.

Riešenie Vytvorili sme novú časť v aplikácii, ktorá bude umožňovať upload svg súboru, ktorý je odoslaný na server. Odpoveďou zo strany servera budú koordináty požadovaného elementu, ktoré extrahuje iná funkcia na serveri.

7.9 Transformácia koordinátov

INDOORNAV-88, Juraj Marák

Analýza Používateľ chce, aby sa vstupný SVG path polygónu chodby transformoval do koordinátovej sústavy, s ktorou sa dá ďalej pracovať. Je teda potrebné vytvoriť funkcie pre transformáciu svg path elementu do poľa koordinátov.

Riešenie Vytvorili sa funkcie pre transformáciu path elementu do poľa koordinátov, kde sú jednotlivé body extrahované z path elementu reprezentované jednorozmerným poľom json objektov Point. JSON point obsahuje koordináty X a Y daného bodu.

7.10 Reprezentácia mriežky

INDOORNAV-89, Ondrej Čerman

Analýza Pre výpočet rozmiestnenia beaconov využijeme algoritmus, ktorý z mriežky bodov (ktoré reprezentujú beacons) odoberá beacons, pokiaľ nenájde dobré riešenie. Preto je potrebné mriežku vytvoriť. Mriežka pritom musí pokrývať plochu, na ktorej majú byť rozmiestnené beacons.

Riešenie Vytvoril som metódu `gridGenerator.compute`, ktorá danú mriežku vytvorí. Zo vstupu (polygón predstavujúci miesto pre beacons) sa vypočítajú hraničné body a vytvorí sa 2D mriežka. Pre každý bod sa následne kontroluje, či je vo vnútri polygónu. Pokiaľ áno, v mriežke sa uložia súradnice. Pre kontrolu, či je bod vo vnútri polygónu slúži algoritmus založený na ray-casting, ktorý avšak nezaručuje správnosť pri testoch na hranách, preto sa hrany kontrolujú zvlášť.

7.11 Výpočet šírenia signálu do okolitých bodov

INDOORNAV-90, Gabriel Takács

Analýza Pre každý bod v mape (mriežke) potrebujeme určiť, že keď do tohto bodu umiestnime beacon, do ktorých všetkých okolitých bodov signál rozšíri. Radius (polomer) šírenia signálu musí byť možné nastaviť ako konštantu, pretože sa bude meniť (v závislosti na použitých beaconoch). Na určenie, či sa bod nachádza v dosahu beaconu bude potrebné použiť Pytagorovu vetu (predpokladáme šírenie signálu do kruhu). Vzhľadom na to, že výpočet môže prebiehať pre veľké množstvo bodov, je výpočet nutné aspoň čiastočne optimalizovať.

Riešenie Výpočet šírenia signálu do okolitých bodov som implementoval do existujúcej metódy `gridGenerator.compute` v súbore `propagation.js`. Táto metóda je používaná na vygenerovanie mriežky zo zadanej mapy. Implementoval som aj optimalizáciu, ktorá počíta s výslednou časovou náročnosťou $N * radius^2$, kde N je počet bodov v mriežke a `radius` je polomer šírenia signálu ($radius \ll N$). Touto optimalizáciou sa čas výpočtu značne skrátil, pretože pôvodná časová

náročnosť bola N^2 . Šírenie signálu vypisujem do pridaného textového poľa v tvare: “(x_zdroj, y_zdroj): (x_bod1, y_bod1) (x_bod2, y_bod2)...”. Správnosť algoritmu som overil ručne - nakreslil som si danú mriežku a overil som správnosť na niekoľkých náhodne vybraných bodoch (aj bodov na kraji mriežky, aj bodoch v strede mriežky).

7.12 Aktualizovanie mapy FIIT

INDOORNAV-91, Miroslav Takács

Aktualizovanie mapy FIIT

Používateľský príbeh Používateľ chce mať dostupnú aktuálnu verziu mapy budovy, a mať na mape zobrazené názvy laboratórií a niektorých dôležitých miestností.

Analýza Do mapy FIIT treba zapísať názvy ďalších miestností, a zakresliť nové priečky podľa plánov budovy od pána Špičku.

Riešenie Pomocou programu Inkscape boli do mapy zaznamenané názvy niektorých miestností. Do mapy boli pridané nové priečky a dvere.

7.13 Selenium testy

INDOORNAV-92, Matúš Demko

Analýza Treba testovať.

Riešenie Boli vytvorené používateľské testy pomocou knižnice Selenium automatizujúcej ovládanie webových prehliadačov (podporované sú takmer všetky - Firefox, Chrome, Safari, IE, Opera). Testy boli vytvorené dvoma spôsobmi:

nahrávaním činnosti pomocou prídavného modulu dostupného iba pre webový prehliadač Firefox naprogramovaním testu pomocou jedného z programovacích jazykov (zvolili sme jazyk Java), pre ktoré existuje knižnica Selenium (Java, C#, Python, Ruby alebo Javascript)

Spôsob č.1 je vhodný na rýchle jednoduché testovanie, avšak nie je vhodný pri veľkom množstve opakujúcich sa operácií (v našom prípade napríklad výber vrstiev). Na opakujúce sa činnosti je vhodnejšie použiť druhý spôsob, ktorý umožňuje použitie programatických cyklov a priradovanie jednotlivých webových elementov do objektov.

7.14 Algoritmus rozmiestnenia beaconov

INDOORNAV-93, Matúš Demko, Ondrej Čerman

Analýza Používateľ chce zadaním základných hodnôt (*hustota mriežky, dosah beaconov, minimálne pokrytie jedného bodu mriežky*) a stlačením jedného tlačidla vygenerovať rozmiestnenie beaconov v polygóne zo súboru ktorý nahral na server.

Riešenie Po transformácii koordinát z nahratého svg súboru a výpočte šírenia signálu do okolitých bodov, máme k dispozícii pole objektov “*Point*” v ktorom poznáme susedné body tohto bodu.

Hodnoty tohto pola sú následne zdublikované (hard copy) a skopírované do ďalších dvoch poli: listCopy - hard copy zoznamu bodov v poli,

beaconList - zoznam bodov mriežky, ktoré sú typu beacon (isBeacon=true)- na začiatku sú všetky body poľa beaconmi,
shuffledList- zabezpečí náhodné poradie vykonávania algoritmu.

Hard copy zabezpečí, že algoritmus bude možné vykonať viac krát za sebou s pôvodnými hodnotami bez opätovného vytvárania poľa a výpočtu šírenia signálu do okolitých bodov.

Nad novovzniknutými objektami sa vykoná navrhnutý algoritmus (pseudokód):

```
for 1 to 100 begin
```

```
  foreach(Point)
```

```
    removeBeacon(randomBeacon);
```

```
end.
```

Funkcia *removeBeacon()* sa snaží vymazať každý beacon, avšak iba v tom prípade že po jeho odstránení budú mať všetky susedné body dostatočné pokrytie(hodnota zadávaná pri spúšťaní automatického rozmiestnenia).

Táto funkcia bola aj optimalizovaná- pri každom odstránení beaconu je šanca že sa nevymaže (ak je pri stene - 98%, inak - 2%).

Po skončení odstraňovania beaconov s náhodným obnovením beaconu sa vykoná algoritmus ešte raz, avšak tentokrát už bez šance na vrátenie. Tým sa zabezpečí že sa odstránia všetky nadbytočné beacons.

7.15 Vyhodnotenie rozmiestnenia algoritmom

INDOORNAV-94, Samuel Pecár

Analýza Pre každé rozmiestnenie je potrebné zistiť počet beaconov aj ich dosah.

Riešenie Pre každé rozmiestnenie zavoláme funkciu, ktorá spočíta počet beaconov v rozmiestnení a následne aj pokrytie priestoru beaconmi, na základe ktorého sa zistí to najlepšie pokrytie, ktoré sa uchová pre ďalšie porovnania.

7.16 Transformácia výsledku do svg mapy

INDOORNAV-95, Juraj Marák

Analýza Používateľ chce, aby bolo možné výsledky algoritmu na rozmiestnenie beaconov zobrazit' na mape. Je preto potrebné vytvorit' funkciu na transformáciu koordinát z používanej sústavy na vykresliteľný formát.

Riešenie Navrhla sa funkcia na transformáciu koordinát, ktorá bola následne nevyužitá. Rozhodli sme sa pre úpravu existujúcej funkcionality na vykreslenie bodov do mapy. Táto funkcionality bola upravená tak, aby akceptovala na vstupe pole koordinát a transformovala ich do Leaflet mapy.

7.17 Zobrazenie SVG mapy a beaconov

INDOORNAV-96, Samuel Pecár

Analýza Najlepšie rozmiestnenie je potrebné vizualizovať v podobe mapy.

Riešenie Najlepšie vypočítané rozmiestnenie sme zobrazili v podobe mapy v Leaflete, kde je zobrazený aj pôvodný polygón aj rozmiestnené beacony.

7.18 Unit testy

INDOORNAV-97, Gabriel Takács, Miroslav Takács

Analýza Cieľom tejto úlohy je dodatočné vytvorenie jednotkových testov pre už hotové základné funkcie tak, aby bol kód aplikácie testovateľný.

Nižšie je uvedený zoznam všetkých relevantných metód v aplikácii:

Súbor	Metóda	Poznámka
converter.controller.js	convertFile	Nevieme, či má zmysel metódu testovať. Ak áno, extrahovať do modelu.
	getSVG	Extrahovať do modelu.
	handleLayers	
layers.controller.js	getLayers	Metóda pracuje s Redisom. Snáď bude možné otestovať. Extrahovať do modelu.
	getLayersData	
path.controller.js	parseSVGPath	Extrahovať do modelu.
fileHandler.js	handle	Nemá zmysel testovať
logger.js		Nemá zmysel testovať
svgBuilder.js	getJSONtoSVG	
	elementToSVG	
	lineStringToSVG	
	pointToSVG	
	multiLineStringToSVG	
	polygonToSVG	

Zoznam analyzovaných metód v klientskej časti aplikácie:

Súbor	Metóda	Poznámka
export.js	exportToSvg	Metóda controllera. Netestovať.
main.js	Nemá	Nie je čo testovať.
mapdisplay.js	init	Metóda controllera. Netestovať.
	getLayersData	
	getCheckedLayers	
	loadDataToMap	

mapedit.js	init	Oddeliť do modelu a testovať.
	saveMap	
	loadDataToMap	
propagation.js	submit	
	getGrid	
	arr2d	
	gridgenerator	
	removebeacon	
	isPointOnPolygonEdge	
	isPointOnLine	
isPointInside		
upload.js	submit	Metóda controllera. Netestovať.

Nakoľko sa aplikačná logika má nachádzať v modeloch, metódy controllerov nemá zmysel testovať. Naša aplikácia však obsahuje veľké množstvo aplikačnej logiky aj v controlleroch. Z tohto dôvodu je nutné logiku oddeliť do modelov a následne napísať jednotkové testy pre tieto modely. Funkcionalita controllerov je prakticky otestovaná pomocou Seleniových testov.

Na vytvorenie jednotkových testov je potrebné vybrať vhodný Javascript framework. Nižšie uvádzam zoznam frameworkov, ktoré som bral do úvahy ako možných kandidátov na vytvorenie jednotkových testov:

Karma

- zapne malý web server
- načíta stránku v minimalistickom prehliadači
- otestuje a vráti výsledky
- out-of-the-box solution

Mocha

- menší balík, ale pravdepodobne bude postačovať pre naše potreby

Jasmine

- podobne ako Mocha, menší testovací balík
- na rozdiel od Mocha má vlastnú assertion library, vlastné metódy a pod.

should.js

- assertion library

Riešenie Na vytvorenie jednotkových testov na serverovej časti sme sa rozhodli použiť framework Mocha spolu s assertion library should.js. Pre tieto frameworky sme sa rozhodli kvôli minimalistickej povahe a jednoduchosti balíka Mocha a balíka should.js.

Zoznam vytvorených metód, pre ktoré sme vytvorili testy:

Časť aplikácie	Trieda	Metóda	Poznámka
Backend	SvgBuilder	pointToSvg	
		polygonToSvg	
		lineStringToSvg	
		multiLineStringToSvg	
		geoJsonToSvg	
	SvgPathParser	parse	
	Connector	setKey	
getKey			
Frontend	gridGenerator	shuffle	
		arr2D	
	geometryUtils	isPointOnLine	
		isPointInside	
		isPointOnPolygonEdge	

Pri tvorbe jednotkových testov pre časť aplikácie, ktorá beží na serveri, som narazil na ďalšie vylepšenia v testoch, ktoré by bolo možné a vhodné spraviť:

- Test converter.js testuje metódu controllera Converter. Aplikačnú logiku z tohto controllera by som navrhol presunúť do modelu.

- V novo-vytvorenom modeli by bolo možné testovať aj metódu ConverterController::handleLayers, LayersController::getLayers, LayersController::getLayersData
- Takto oddelenú funkcionálnosť by bolo možné testovať ako celok

7.19 Oprava bugov

INDOORNAV-98, Matúš Demko

Bug Nefungovalo download SVG vo Firefoxe

Riešenie Nefungovalo kvôli tomu, že sťahovanie SVG funguje tak, že sa simuluje klik na odkaz. Keďže tento odkaz nebol v DOM, vo Firefoxe toto nefungovalo. Odkaz sa teraz dočasne pridá do DOM.

Bug V propagation nefungovali niektoré SVG

Riešenie Nefungovali SVG, ktoré nemali zadaný viewBox. Pri týchto SVG sa viewBox vypočíta z width a height.

7.20 Úprava klienta pre propagation časť

INDOORNAV-99, Juraj Marák

Analýza Používateľ chce, aby bol vzhľad podstránky Propagation upravený do prezentovateľnej podoby. Chce doplniť validácie nad vstupnými poľami formulára a zobrazovanie iba relevantných častí html stránky na základe kroku v ktorom sa práve nachádza.

Riešenie Navrhli a implementovali sa zmeny na propagation podstránke. Pridali sa validácie nad každým vstupným poľom, označenie chybných vstupov a rozdelenie stránky na jednotlivé kroky. Zároveň sa doplnila funkcionálnosť pre nastavenie vzhľadu výstupnej mapy.

7.21 Unit testy

INDOORNAV-100, Gabriel Takács, Miroslav Takács

Analýza Cieľom tejto úlohy je vytvorenie jednotkových testov pre funkcie analyzované v úlohe INDOORNAV-97, pre ktoré neboli vytvorené testy v úlohe aby bol kód aplikácie testovateľný.

Riešenie Zoznam vytvorených metód, pre ktoré sme vytvorili testy:

Časť aplikácie	Trieda	Metóda	Poznámka
Backend	layersHandler	saveLayers	Test napísaný pre metódu convertFile controllera Converter
		getLayersList	

		getLayersData	
Frontend	gridGenerator	deepCopyList	
		convertListToGrid	
		calculateNeighbors	

Na frontendovej časti aplikácie sa ešte nachádzajú funkcie, ktoré nie sú otestované. Niektoré z týchto funkcií ale nie sú deterministické, alebo obsahujú príliš komplexné vstupy a výstupy - narazili sme na chybu *circular structure error*.

Na backendovej časti aplikácie sme oddelili funkcionality z LayersControllera do modulu LayersHandler. Oddelené metódy sme zrefaktorovali a napísali sme pre ne jednotkové testy.

Napriek snahe otestovať metódu LayersHandler::saveLayers mimo controllera, nebolo to možné a túto metódu testujeme v rámci metódy convertFile controllera Converter.

7.22 Prezentácia na Tech Inno day

INDOORNAV-101, Miroslav Takács

Analýza Treba pripraviť a prezentáciu a zúčastniť sa na Tech Inno day, ktorá sa uskutoční 6. apríla 2016. Prezentácia prezentovaná na Tech Inno day je súčasťou priloženého CD média.

7.23 Úprava procesu editácie mapy

INDOORNAV-102, Samuel Pecár

Analýza Proces pre úpravu mapy z DXF je potrebné upraviť. Pokúsiť sa objaviť zdroj chyby needitovateľných čiar. Používateľ by mal mať možnosť výberu štýlu svg mapy.

Riešenie Používateľ má možnosť pri exporte editovanej mapy upraviť štýl mapy. Môže upraviť farbu obrysu, výplň ako aj hrúbku čiary. Snaha o objavenie zdroja chyby needitovateľných čiar v editore bola neúspešná, pretože chyba bude pravdepodobne priamo v knižnici Leaflet.Draw.

7.24 Vytvorenie postera

INDOORNAV-104, Ondrej Čerman

Analýza Pre prezentáciu IIT.SRC 2016 je potrebné vytvoriť poster vo formáte A1 reprezentujúci náš projekt

Riešenie Poster bol vytvorený v programe Inkscape. Poster je súčasťou CD média.

7.25 Prepočet jednotiek na metre

INDOORNAV-105, Matúš Demko

Analýza Používateľ chce používať intuitívne merné jednotky (metre) namiesto pixelov pri zadávaní hustoty a polomeru dosahu beaconov.

Riešenie Po nahratí súboru sa zo zadanej šírky chodby a šírky obrázku vypočíta počet pixelov na jeden meter. Touto hodnotou sú pre násobené hodnoty: hustota rozmiestnenia beaconov a polomer dosahu beaconov.

7.26 Príprava testovania používateľmi

INDOORNAV-106, Matúš Demko

Analýza Používateľ chce otestovať funkcionality vytvoreného prototypu.

Riešenie Bola vytvorená metodika pre testovanie používateľmi obsahujúca testovacie scenáre. Metodika pre testovanie je napísaná v prílohe B.

7.27 Vytvorenie SVG pre každú chodbu na FIIT

INDOORNAV-107, Miroslav Takács

Analýza Do aktuálnej mapy FIIT je potrebné pridať vrstvy obsahujúce polygóny chodieb. Do týchto polygónov bude algoritmus rozmiestňovať beacony, a budú použité v aplikácií na navigáciu v budove.

Riešenie Do svg súborov pre jednotlivé poschodia budovy FIIT boli pridané vrstvy si id chodby. V týchto vrstvách sa nachádzajú polygóny ohraničujúce chodby.

7.28 Doplniť Wi-Fi do algoritmu

INDOORNAV-108, Ondrej Čerman

Analýza Používateľ chce, aby v blízkosti Wi-Fi vysielateľov neboli umiestnené beacony. Je teda potrebné, aby aplikácia dokázala načítať informácie kde sa Wi-Fi vysielateľ nachádzajú a v ich blízkosti neumiestňovali beacony.

Riešenie V vstupnom súbore spolu s polygónom určujúci miestnosť sú zaznačené aj Wi-Fi vysielateľ - pomocou kruhov, ktoré majú triedu Wi-Fi, vid' http://img.h.us/wifi_3.svg. Stred kruhu určuje miesto Wi-Fi a polomer určuje dosah Wi-Fi signálu (miesto, v ktorom sa nebudú nachádzať beacony).

Tieto informácie načíta algoritmus pre výpočet polohy beaconov a po vytvorení plnej mriežky sa odstránia body, ktoré sú v dosahu Wi-Fi. Algoritmus počíta polohy beaconov ďalej bez zmeny.

Do mapy sa vykreslia beacony pomocou prvku circle. Kvôli chybe v leaflete bolo potrebné upraviť počítanie polomera kruhu.

Pri exporte do svg bolo potrebné rozlíšiť beacon a Wi-Fi (pri exporte z leafletu do geojsonu sú oba určené ako body), preto každý kruh pri exporte do geojsonu v časti properties má uložený polomer.

7.29 Doplniť progress bar do propagation

INDOORNAV-109, Juraj Marák

Analýza Používateľ chce vidieť postup generovania rozmiestnenia beaconov na propagation podstránke. Je teda nutné implementovať progress bar.

Riešenie Do podstránky bol doplnený progres bar ktorý indikuje aktuálny stav generovania rozmiestnenia beaconov. Okrem pridania samotného progres baru si riešenie vyžadovalo úpravu implementácie funkcie pre generovanie rozmiestnenia, a to tak, aby po ukončení každej iterácie funkcia oznamovala progres baru aktuálny stav.

7.30 Update dokumentácie

INDOORNAV-110

Riešenie Boli doplnené niektoré časti dokumentácie, konkrétne bol aktualizovaný Big Picture, Boli doplnené diagramy a upravené existujúce.

7.31 Upratovanie v podkladoch k projektu

INDOORNAV-111

Riešenie Súborny na našom úložisku na Google Drive boli kategorizované do priečinkov. Niektoré úlohy v Jire boli premenované, boli opravené preklepy a doplnené popisy.

7.32 Prezentácia projektu na IIT.SRC

INDOORNAV-112

Riešenie Prezentovali sme náš prototyp na konferencii IIT.SRC.

7.33 Vytvorenie inštalačnej príručky

INDOORNAV-113, Ondrej Čerman

Riešenie Bola vytvorená inštalačná príručka pre nasadenie aplikácie IndoorNav na server. Inštalačná príručka je súčasťou prílohy C.

7.34 Komplexný code review

INDOORNAV-114, Samuel Pecár, Juraj Marák

Riešenie Prešli sme zdrojové kódy aplikácie, odstránili zbytočné časti kódu, pridali chýbajúce komentáre a upravili logovanie do súborov.

7.35 Doplnenie chýbajúcich user stories

INDOORNAV-115, Miroslav Takács

Analýza Treba doplniť používateľské príbehy, ktoré boli prehliadnuté alebo sa na ne zabudlo.

Riešenie Do dokumentácie boli doplnené používateľské príbehy pre niektoré úlohy.

7.36 Vytvorenie používateľskej príručky

INOORNAV-116, Gabriel Takács

Riešenie Vytvorili sme používateľskú príručku k editoru. Používateľská príručka je dostupná online. Používateľská príručka je súčasťou prílohy A.

7.37 Testovanie prototypu používateľmi

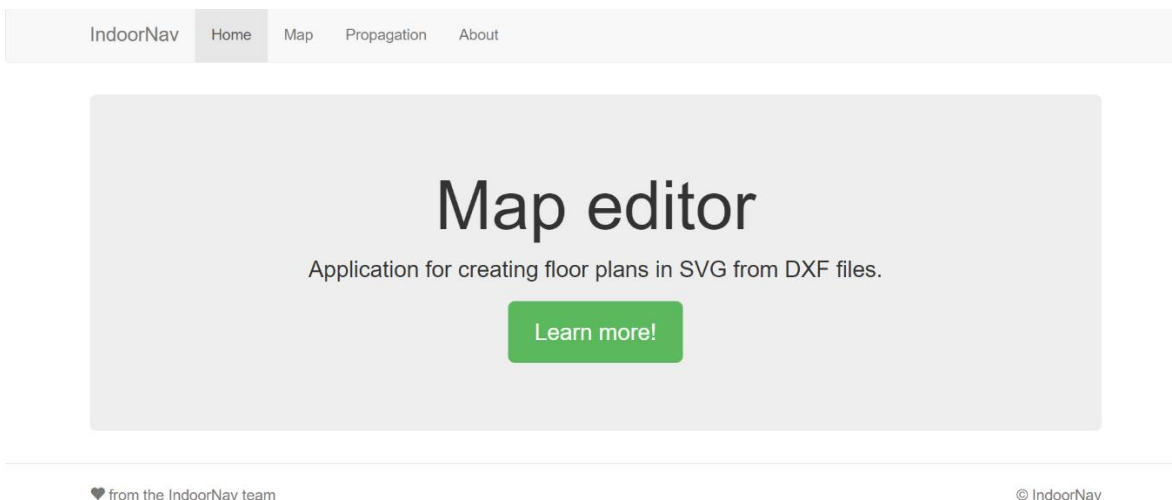
INDOORNAV-117, Matúš Demko

Riešenie Testovanie prebehlo s piatimi používateľmi - 2 vysokoškolákmi, 1 tínedžerom a 2 dospelými. Postup a výsledky testovania sú súčasťou prílohy B.

Príloha A Používateľská príručka

Aplikácia je dostupná online na <http://team10-15.studenti.fiiit.stuba.sk:9000/#/>. Úvodná stránka obsahuje názov a stručný popis aplikácie. V hornej časti stránky sa nachádza navigačný panel so štyrmi tlačidlami:

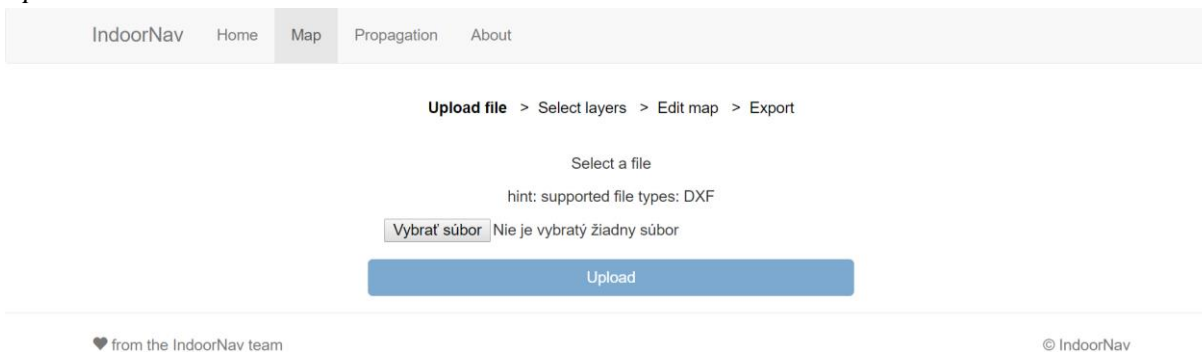
- Home - úvodná stránka
- Map - stránka pre editáciu mapy
- Propagation - stránka pre výpočet rozmiestnenia majáčikov
- About - informácie o tíme Indoornav a samotnej aplikácii.



Obrázok 1 Úvodná stránka aplikácie.

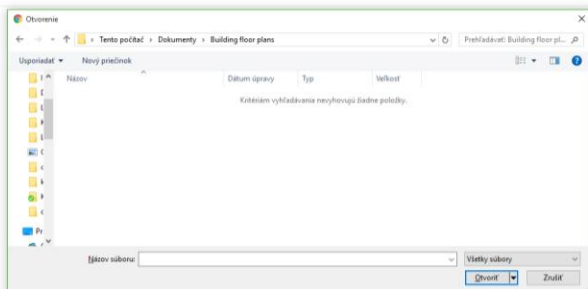
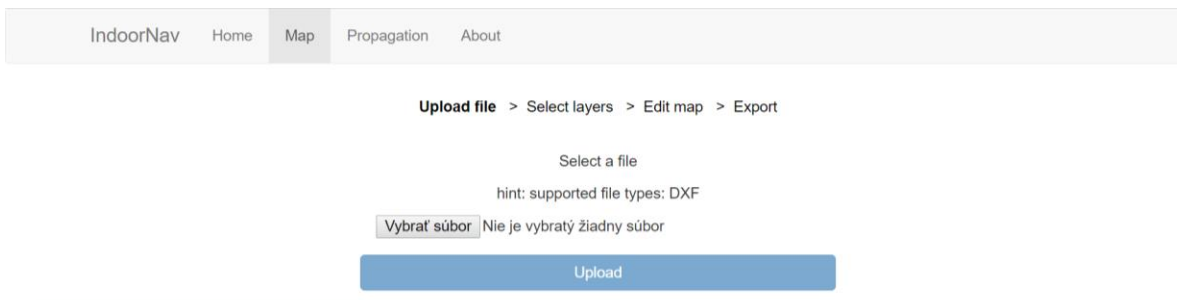
Editácia mapy

Aby bolo možné zahájiť editáciu mapy, je nutné najprv zvoliť v hornom navigačnom paneli možnosť *Map*.



Obrázok 2 Obrázovka nahrávania súboru pri editácii mapy.

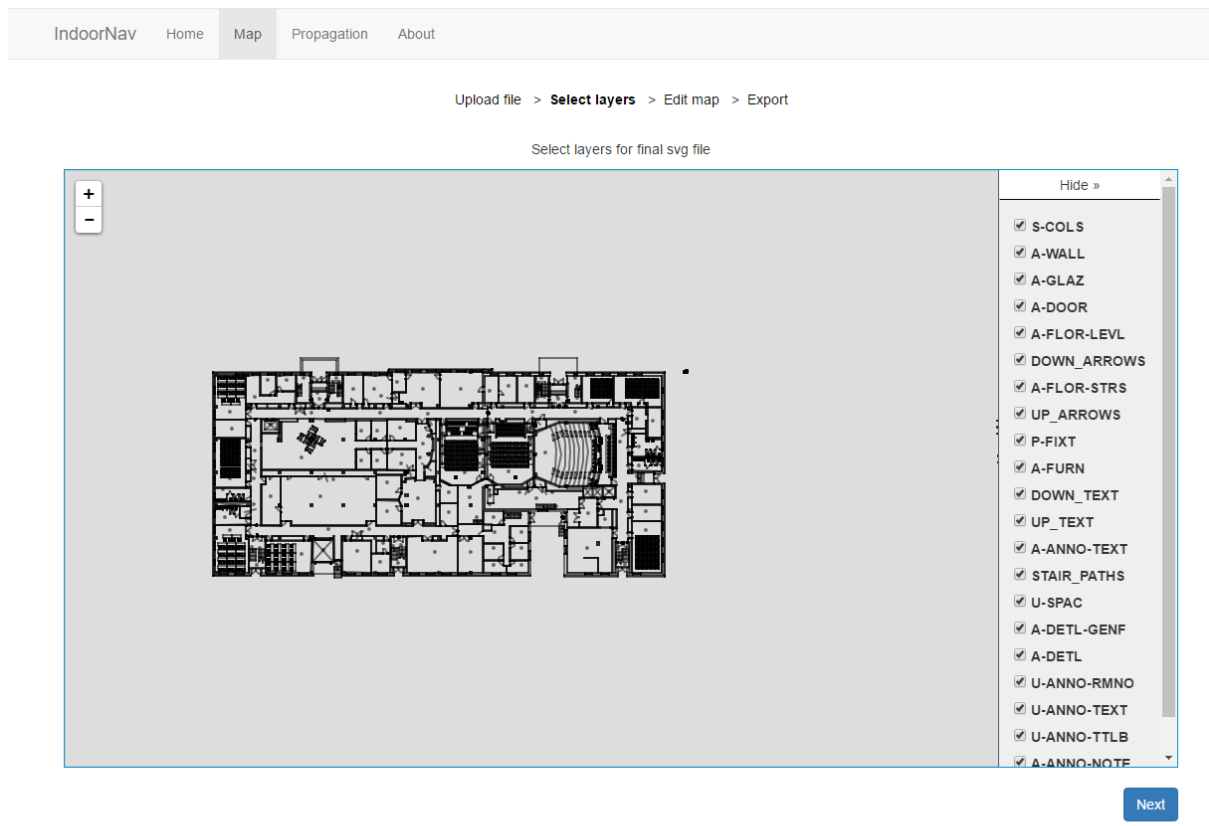
Pre nahratie súboru plánu poschodia budovy klikneme na tlačidlo *Vybrať súbor* (Obr. č. 2), ktoré nám otvorí dialóg prehľadávania priečinkov (Obr. č.3).



Obrázok 3. Dialóg prehľadávania priečinkov.

Po zvolení súboru, ktorý chceme upraviť do formy použiteľnej v navigácii, potvrdíme výber stlačením tlačidla *Otvoriť*. Po uzavretí dialógu spustíme nahrávanie súboru stlačením tlačidla *Upload*. Tento proces môže trvať niekoľko minút!

Po nahratí súboru na server sme automaticky prenesení na stránku odstraňovania existujúcich vrstiev z nahratého súboru – *Select layers* (Obr. č.4).



♥ from the IndoorNav team

© IndoorNav

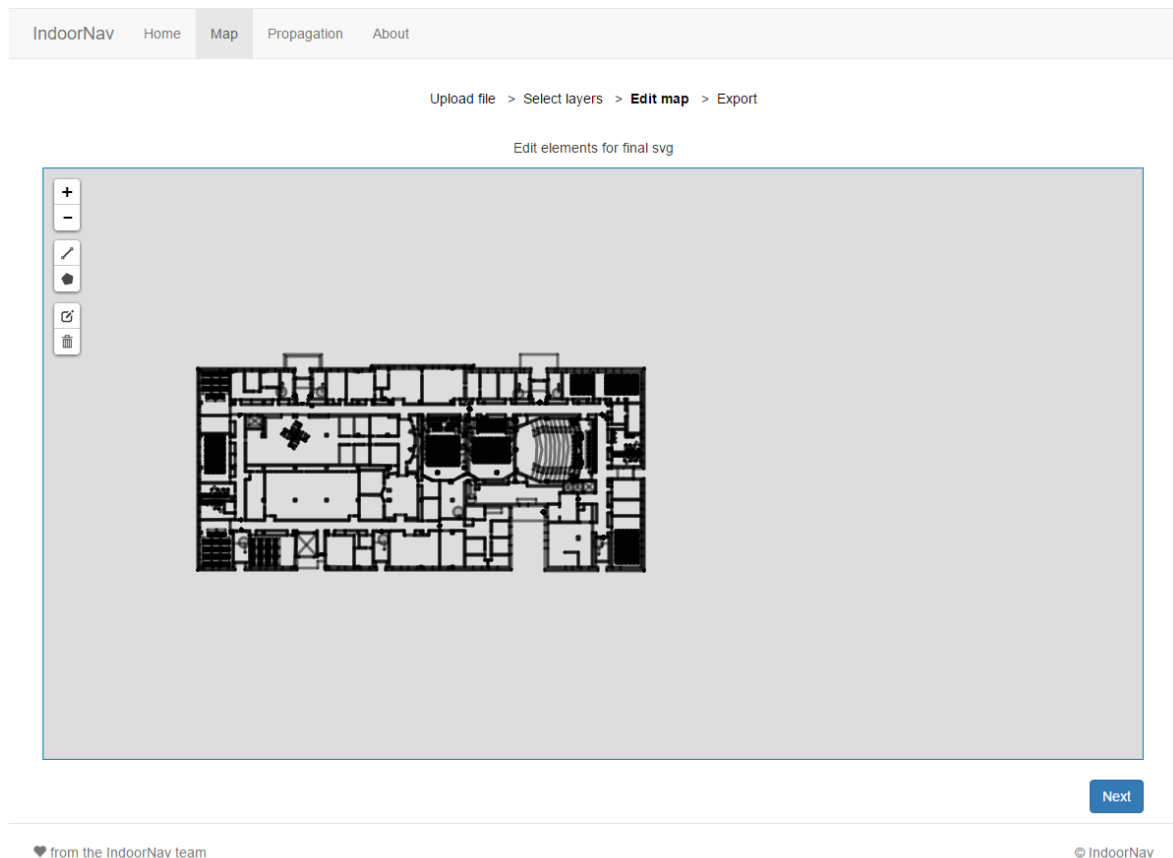
Obrázok 4. Obrazovka odstraňovania vrstiev.

Pre odstránenie ľubovoľnej vrstvy odznačíme zaškrtnutý box pri jej názve. Po kliknutí na box sa spustí vymazávanie vrstvy znázornené točiacim sa kolieskom.


Pre výber ďalšej vrstvy nemusíme čakať na dokončenie vymazávania a môžeme kludne pokračovať vo výbere.


Ak sme spokojní s výsledným počtom zobrazených vrstiev, môžeme prejsť na ďalšiu časť stlačením tlačidla *Next* v pravom dolnom rohu obrazovky.


V časti *Edit map* môžeme pridávať nové čiary, polygóny a odstraňovať existujúce čiary. Táto časť obsahuje na ľavej strane panel s nástrojmi. Obrázok sa dá približovať točením kolieskom myši alebo tlačidlami + a – na paneli nástrojov.




Obrázok 5. Obrázovka editácie mapy.

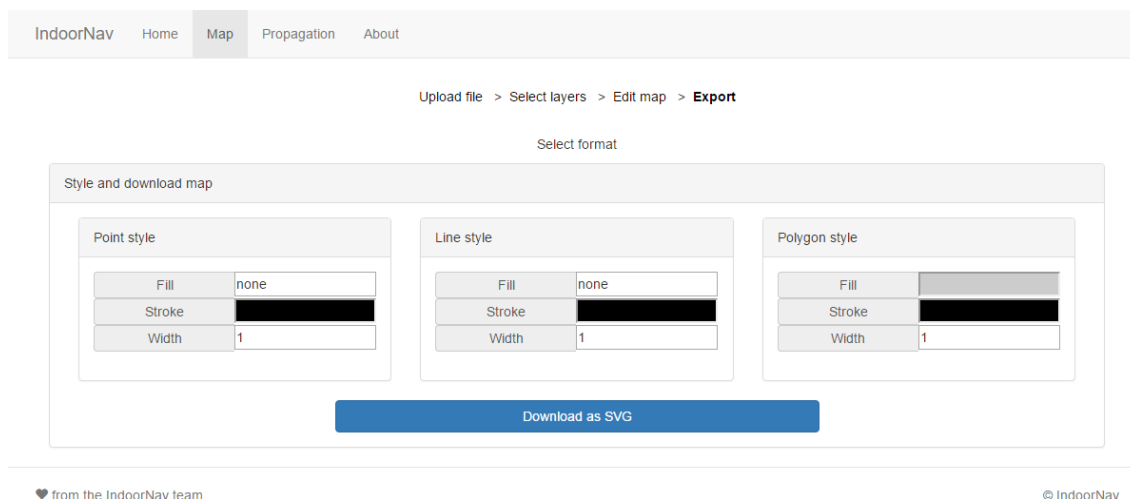
Pre pridanie novej čiary zvolíme nástroj pridania čiary (cesty) . Po aktivácii nástroja sa prvým kliknutím na plochu sa označí začiatkový bod čiary. Každým ďalším kliknutím na ľubovoľné miesto na obrázku sa čiara predlžuje. Pre potvrdenie vytvorenia čiary klikneme na posledný vytvorený bod čiary.

Pre pridanie nového polygónu zvolíme nástroj pridania polygónu . Tento nástroj sa správa rovnako ako nástroj vytvorenia čiary. Jedinou výnimkou je potvrdenie vytvorenia objektu, pretože polygón je nutné uzavrieť kliknutím na **začiatkový** bod polygónu.

Pre úpravu existujúcich čiar zvolíme nástroj editácie čiar . Po aktivácii nástroja editácie sa editovateľné čiary zafarbia na ružovo a zobrazia sa body, ktoré sa dajú posúvať. Po dokončení upravy čiar je nutné potvrdiť zmeny tlačidlom *Save* pri ikonke nástroja alebo zahodiť zmeny tlačidlom *Cancel*.

Pre odstránenie existujúcich čiar zvolíme nástroj odstraňovania čiar . Po aktivácii tohto nástroja stačí kliknúť na ľubovoľnú čiaru. Po dokončení odstraňovania čiar je nutné potvrdiť zmeny tlačidlom *Save* pri ikonke nástroja alebo zahodiť zmeny tlačidlom *Cancel*.

Pre ukončenie editácie mapy stlačíme tlačidlo *Next* v pravom dolnom rohu obrazovky.

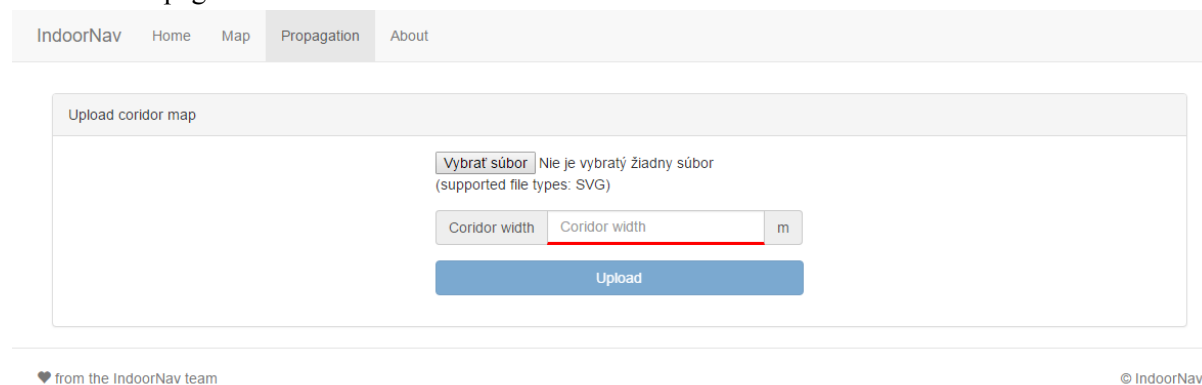


Obrázok 6 Obrazovka exportu výsledného obrázka.

V časti *Export* si môžeme prispôsobiť vzhľad výslednej mapy podľa požiadaviek. Následne stiahneme súbor pomocou tlačidla *Download as SVG*.

Výpočet rozmiestnenia majáčikov (propagation)

Ak chceme využiť možnosť výpočtu rozmiestnenia majáčikov, zvolíme v hornom navigačnom paneli možnosť *Propagation*.



Obrázok 7 Obrazovka Nahratia súboru pre výpočet rozmiestnenia.

Pre automatický výpočet rozmiestnenia je nutné nahráť jednotlivé súbory SVG chodieb poschodia. Po nájdení súboru (podobne ako pri výbere súboru pre editáciu) zadáme do poľa *Corridor width* šírku nahrávanej chodby (šírka celého obrázka chodby). Na nahratie súboru stlačíme tlačidlo *Upload*.

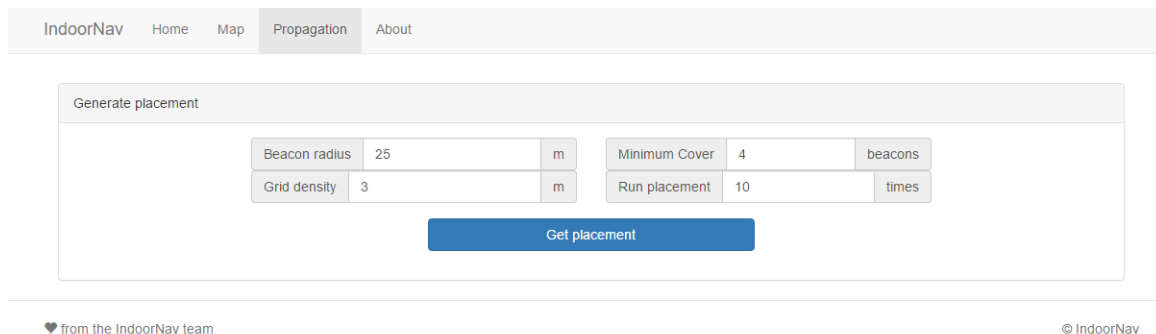
Následne zadáme požadované parametre rozmiestnenia:

Beacon radius - dosah majáčikov v metroch

Minimum cover - minimálny počet majáčikov v každom bode mriežky

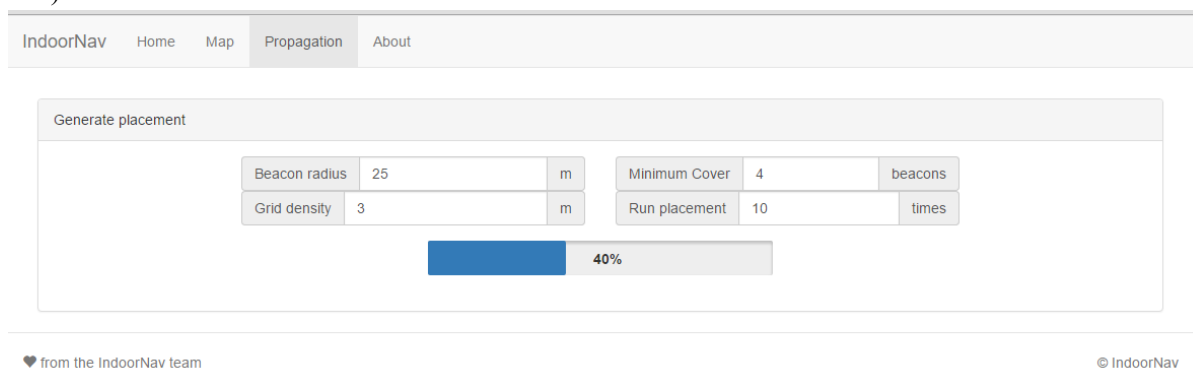
Grid density - hustota mriežky

Run placement X times - počet opakovaní výpočtu (čím vyššie číslo tým dlhší výpočet a väčšia šanca že sa nájde najlepšie rozmiestnenie)



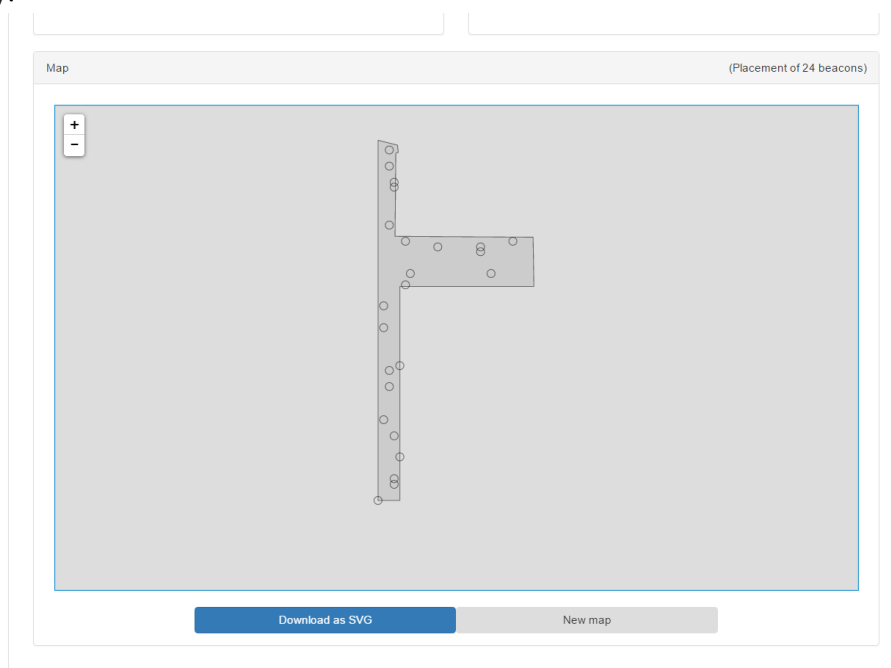
Obrázok 8. Obrazovka zadávania parametrov výpočtu rozmiestnenia.

Tlačidlo *Get placement* spustí výpočet rozmiestnenia znázornený plniacim sa pásikom s percentami (Obr. č. 9).



Obrázok 9. Prebiehajúci výpočet rozmiestnenia.

Po ukončení výpočtu sa premiestnime nižšie kde uvidíme výsledok výpočtu a finálne rozmiestnenie majáčikov. Pre stiahnutie súboru obsahujúceho majáčiky stlačíme tlačidlo *Download as SVG* v spodnej časti obrazovky.



Obrázok 10 Výsledné rozmiestnenie majáčikov.

Príloha B Používateľské testovanie

Postup pre testerov

Editácia mapy

1. Navštív <http://team10-15.studenti.fiit.stuba.sk:9000/#/>
2. Nahraj mapu (súbor 65-01.dxf), tento proces môže trvať niekoľko minút
3. Odstráň nepotrebné vrstvy (text, notes, šípky, nábytok, anotácie ...)
4. Odstráň stenu
5. Ulož zmeny
6. Pridaj stenu
7. Stiahni svg súbor novej mapy

Výpočet rozmiestnenia majáčikov

1. Navštív <http://team10-15.studenti.fiit.stuba.sk:9000/#/>
2. Vyber súbor chodby (fiit-1.4.svg) a zadaj jej šírku.
3. Nahraj súbor chodby.
4. Zadaj šírku chodby
5. Zadaj požadované parametre rozmiestnenia.
6. Spuť výpočet rozmiestnenia.
7. Stiahni svg súbor chodby s majáčikmi.

Výsledky testovania

Testovanie prebiehalo na prehliadači Google Chrome (verzia 50.0.2661.94 m) na online verzii aplikácie.

Používateľ 1

Aplikácia dlho nahrávala súbor mapy, chýbalo mi tam odpočítavanie zostávajúceho času. Tam kde bolo treba odškrtnúť čo nechcem mať na mape, som nerozumela skratkám a názvom. Nerozumela som prečo nemôžem použiť mapu, ktorú som si upravila v prvej časti. Polovica vecí bola po anglicky a polovica zas po slovensky a to sa mi zdá neprehľadné.

Používateľ 2

Chýbali mi tam bublinky ktoré by mi ukazovali interaktívne, čo mám spraviť a ako. Ani po prečítaní návodu som úplne nerozumel štyrom poliam pri výpočte umiestnenia majáčikov. Na stránke nie je návod na používanie. Pri odstraňovaní vrstiev sa niekedy zaseklo koliesko a nevedel som čo sa deje.

Používateľ 3

Páčilo sa mi že pri správne zvolených vrstvách, mi to pekne vyčistilo obrázok budovy. Páčilo sa mi že som pri výbere vrstiev nemusel čakať na dokončenie vymazávania vrstvy a mohol som pokračovať v odškrtavaní. Chýbajú informácie o majáčikoch na stránke.

Chýbajú vysvetlivky k parametrom rozmiestnenia.

Používateľ 4

Neintuitívne ovládanie editácie (zmiatol ma zoom pri nástrojoch).

Zaseklo sa koliesko avšak keď som klikal ďalej na vrstvy ale neodškrtovali sa. Po odseknutí sa následne odškrtili a to ma zmiatlo.

Chýbali mi prednastavene možnosti odstraňovania vrstiev(napríklad všetky vrstvy s textom).

Chýbali mi vopred stanovené rozsahy pri parametroch.

Používateľ 5

Polka po slovensky a polka po anglicky.

Pri zle zadanom parametri iba podčiarklo a nevypísalo chybu. Chybu vypísalo až po ukázaní myšou na červenú čiaru.

Bez návodu by som určite neprišiel na to že výpočet rozmiestnenia je pod tabom s názvom Propagation.

Príloha C Inštalačná príručka

Aktuálny stav

Aplikácia momentálne nie je nasadená na servery stavba. V čase písania dokumentácie sa len pracovalo na nasadení. Nedostali sme práva na server pre nasadenie našej aplikácie, zodpovednosť za nasadenie si zobral administrátor serverov, ktorý sa bude riadiť nižšie napísanou príručkou.

Kontaktná osoba pri nasadzovaní je Radoslav Pešek.

Inštalačná príručka aplikácie Indoornav

Príručka je primárne určená pre Ubuntu Server x64 14.04.

Predpoklady:

- nainštalovaný OS Linux
- voľné porty 9000, 8444
- existujúci používateľ indoornav s domácim priečinkom /home/indoornav

Inštalácia:

1. Nainštalovanie systémových závislostí:
 - Nastavenie prostredia
 - `su - root`
 - Nainštalovanie nodejs 4.x a npm
 - `curl -sL https://deb.nodesource.com/setup_4.x | sudo -E bash -`
 - `apt-get install -y nodejs`
 - Nainštalovanie Redis 2.8.4
 - `apt-get install redis-server`
 - Nainštalovanie Sass
 - `apt-get install ruby-full`
 - `gem install sass`
 - Nainštalovanie bower, grunt a http-server
 - `npm install --global bower grunt http-server`
2. Stiahnutie zdrojových kódov
 - Nastavenie prostredia
 - `su - indoornav`
 - Získanie kódov z repozitára
 - `git clone https://bitbucket.org/indoornav10/indoornav.git`
 - alebo prekopírovaním z iného zdroja do ~/indoornav
3. Inštalácia servera
 - Nastavenie prostredia
 - `cd ~/indoornav/server`
 - `mkdir logs`
 - Nainštalovanie závislostí
 - `npm install`
4. Inštalácia klienta
 - Nastavenie prostredia
 - `cd ~/indoornav/client`
 - Nainštalovanie závislostí

- `npm install`
 - `bower install`
 - Vygenerovanie produkčnej verzie
 - `grunt build`
- 5. Nastavenie automatického spúšťania
 - Nastavenie prostredia
 - `su - root`
 - `cd /etc/init.d`
 - Získanie spúšťacích skriptov stiahnutím z Internetu
 - `wget -O indoornav http://sda1.eu/etc/indoornav`
 - `wget -O indoornavclient http://sda1.eu/etc/indoornavclient`
 - alebo prekopírovaním z iného zdroja do `/etc/init.d/`
 - Nastavenie oprávnení
 - `chmod a+x indoornav`
 - `chmod a+x indoornavclient`
 - Nastavenie spúšťaní skriptov
 - `update-rc.d indoornav defaults`
 - `update-rc.d indoornavclient defaults`
- 6. Reštart servera