

Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava 4

# **Virtuálna FIIT**

## ViFiiTeam

Dokumentácia k dielu

**Vedúca tímu:**

Mgr. Alena Kovárová, PhD.

**Členovia tímu:**

Bc. Lukáš Cáder

Bc. Martin Dušek

Bc. Jaroslav Dzurila

Bc. Roland Gášpár

Bc. Martin Londák

Bc. Michal Ševčík

Bc. Matej Toma

**Školský rok:**

2013/2014

# Obsah

|     |   |    |
|-----|---|----|
| 1   | Úvod.....   | 6  |
| 1.1 | Účel a rozsah dokumentu .....   | 6  |
| 2   | Stanovenie globálnych cieľov projektu pre zimný semester.....   | 7  |
| 2.1 | Zjednotiť grafický dizajn.....  | 7  |
| 2.2 | Dotvoriť aplikačnú logiku .....   | 7  |
| 2.3 | Google Play .....   | 7  |
| 2.4 | Oboznámiť sa s novými technológiami .....   | 7  |
| 2.5 | Naučiť sa pracovať v tíme .....   | 8  |
| 3   | Úlohy pred začatím šprintov .....   | 9  |
| 3.1 | Optimalizácia máp budovy fakulty.....   | 9  |
| 4   | Opis príbehov 1. šprintu .....  | 10 |
| 4.1 | Prerozdelenie spojov MHD .....  | 10 |
| 4.2 | Zladenie jedálni .....  | 11 |
| 4.3 | Nová grafika hlavného menu.....   | 12 |
| 4.4 | Úprava približovania mapy vo webovej verzii máp.....  | 13 |
| 4.5 | Opraviť názvy odkazov .....   | 14 |
| 4.6 | Hladké scrollovanie .....   | 14 |
| 5   | Opis príbehov 2. šprintu .....  | 17 |
| 5.1 | Kontrola správnosti máp budovy.....   | 17 |
| 5.2 | Zoradenie spojov MHD podľa obľúbenosti .....  | 17 |
| 5.3 | Zoradenie spojov MHD podľa času.....  | 18 |
| 5.4 | Upraviť zobrazenie MHDčky aby bolo viac farebnejšie.....  | 19 |
| 5.5 | Nezobrazovať prázdny zoznam obľúbených spojov MHD.....  | 19 |
| 5.6 | Úprava dizajnu jedální.....   | 20 |
| 5.7 | Mapa okolia fakulty .....   | 22 |
| 5.8 | RSS správy .....  | 23 |
| 6   | Opis príbehov 3. šprintu .....  | 26 |
| 6.1 | Obnoviť zoznam obľúbených spojov pri kroku späť .....   | 26 |
| 7   | Ostatné úlohy 3. šprintu .....  | 27 |
| 7.1 | Konfigurácia webovej verzie virtfiit aby bolo možné sa pripojiť na vzdialený server (AIS, HladnyStudent, stuba, itransit) ..... | 27 |
| 7.2 | Nefunkčná história a zmiešaná obrazovka pri QR kódoch.....  | 28 |

|      |  |    |
|------|--|----|
| 8    | Opis príbehov 4. šprintu .....   | 30 |
| 8.1  | Doplnenie predčítavania dát pre jedálne a MHD .....                                    | 30 |
| 8.2  | Logovanie aktivity používateľa .....   | 31 |
| 8.3  | Pridanie informačného dialógu pri použití QR kódu.....                                 | 33 |
| 8.4  | Zoradenie obľúbených spojov MHD podľa časov odchodu.....                               | 35 |
| 9    | Ostatné úlohy 4. šprintu .....   | 37 |
| 9.1  | Čierna obrazovka pri využití funkcie skenovania QR kódu .....                          | 37 |
| 9.2  | Testovanie usability aplikácie .....   | 38 |
| 9.3  | Testovanie rozdelenia MHD.....   | 41 |
| 9.4  | Príprava ikon pre budúcu funkcionálnosť .....  | 43 |
| 9.5  | Chybné mapovanie najbližších odchodov MHD.....   | 45 |
| 9.6  | Nemožnosť pridať medzi obľúbené linku z rôznych zastávok v jednom smere.....           | 47 |
| 9.7  | Nesprávne zobrazovanie miestnosti na mobilných zariadeniach .....                      | 48 |
| 10   | Opis príbehov 5. šprintu .....   | 50 |
| 10.1 | Logovanie aktivity používateľa, pokračovanie.....                                      | 50 |
| 11   | Ostatné úlohy 5. šprintu .....   | 52 |
| 11.1 | Nástroj Visual Paradigm for UML .....  | 52 |
| 11.2 | Pridanie informácie kto je zodpovedný za miestnosť .....                               | 53 |
| 12   | Opis príbehov 6. šprintu .....   | 55 |
| 12.1 | Zobrazovanie SVG a PNG ikon na základe verzie Androidu.....                            | 55 |
| 12.2 | Vylepšiť aktualizáciu údajov pri MHD .....   | 56 |
| 12.3 | Implementácia posielania logov z aplikácie na server.....                              | 57 |
| 12.4 | Otváracie hodiny na študijnom oddelení .....   | 58 |
| 12.5 | Zistiť spôsob identifikácie verzie databázy a vytvoriť zobrazenie v sekcii ABOUT<br>59 |    |
| 12.6 | Vytvorenie pomoci pri vyhľadávaní .....  | 60 |
| 12.7 | Implementácia mapy okolia .....  | 62 |
| 13   | Ostatné úlohy 6. Šprintu.....  | 63 |
| 13.1 | Opravenie chyby zobrazovania Undefined v rozvrhu.....                                  | 63 |
| 13.2 | Analýza možnosti unikátnej identifikácie mobilného zariadenia .....                    | 63 |
| 13.3 | Zobrazuje sa starý rozvrh.....   | 64 |
| 14   | Opis príbehov 7. šprintu .....   | 65 |
| 14.1 | Možnosť vyhľadávania nad MHD .....   | 65 |

|      |   |    |
|------|---|----|
| 14.2 | Vytvorenie návrhov rozloženia odkazov .....                                     | 67 |
| 15   | Ostatné úlohy 7. šprintu .....  | 69 |
| 15.1 | Analýza sťahovania jedálnych lístkov a MHD .....                                | 69 |
| 16   | Opis príbehov 8. šprintu .....  | 71 |
| 16.1 | Pridanie možnosti nájsť miestnosť aj podľa názvu.....                           | 71 |
| 16.2 | Dorobenie klikateľnosti vo vyhľadávaní v MHD .....                              | 72 |
| 16.3 | Klikateľná mapa okolia.....   | 73 |
| 16.4 | Automatické načítavanie jedálni raz za deň.....                                 | 73 |
| 16.5 | Vážený graf slov z otázok a odpovedí .....                                      | 74 |
| 17   | Opis príbehov 9. Šprintu .....  | 75 |
| 17.1 | Vylepšenie dizajnu MHD a rozvrhov .....   | 75 |
| 17.2 | Nový koncept hlavného menu.....   | 77 |
| 17.3 | Analýza použitia knižnice scrollspy .....                                       | 78 |
| 17.4 | Implementácia kompasu do mapy.....  | 79 |
| 17.5 | Optimalizácia mapy okolia .....   | 81 |
| 17.6 | Analýza kompilácie aplikácie pre Windows Phone.....                             | 81 |
| 17.7 | Zmena spúšťania automatických aktualizácií .....                                | 82 |
| 17.8 | Analýza proximity search v rámci Elasticsearch .....                            | 83 |
| 18   | Ostatné úlohy 9. Šprintu .....  | 85 |
| 18.1 | Chýbajúce zastávky MHD .....  | 85 |
| 19   | Opis príbehov 10. Šprintu .....   | 86 |
| 19.1 | Zmena veľkosti ikon v hlavnom menu podľa rozlíšenia displeja .....              | 86 |
| 20   | Ostatné úlohy 10. Šprintu .....   | 87 |
| 20.1 | Nefunkčné vyhľadávanie z dôvodu nenačítania MHD .....                           | 87 |
| 20.2 | Chyba scrollspy v rozvrhu .....   | 88 |
| 20.3 | Notifikácia emailom o odoslaní logu .....                                       | 88 |
| 21   | Opis príbehov 11. Šprintu .....   | 90 |
| 21.1 | Komunikácia klient server pri odosielaní dopytu inteligentného vyhľadávania ... | 90 |
| 21.2 | Klientská časť inteligentného vyhľadávania.....                                 | 91 |
| 22   | Ostatné úlohy 11. Šprintu.....  | 93 |
| 22.1 | Chýbajúce zastávky MHD - Hl. stanica.....                                       | 93 |
| 22.2 | Veľké rozlíšenie ikon pri malom rozlíšení displeja v landscape mode .....       | 93 |
| 22.3 | Na mape sa nedá približovať mapu pomocou myšky .....                            | 94 |

|      |   |     |
|------|---|-----|
| 23   | Opis príbehov 12. Šprintu .....                                 | 96  |
| 23.1 | Rozvrh skúškového .....   | 96  |
| 23.2 | Serverová časť inteligentného vyhľadávania .....                | 97  |
| 24   | Ostatné úlohy 12. Šprintu .....                                 | 99  |
| 24.1 | Nesprávne rozlíšenie ikoniek v portrait mode pri 768x1024 ..... | 99  |
| 25   | Návody .....  | 100 |
| 25.1 | Inštalácia a spustenie aplikácie pre vývoj .....                | 100 |
| 25.2 | Inštalácia virtuálneho serveru .....                            | 102 |
| 25.3 | Nastavenie automatického update-u databázy .....                | 103 |
| 25.4 | Manuálne spustenie aktualizácie databázy .....                  | 105 |
| 25.5 | Spustenie a build aplikácie pre Windows Phone 8 .....           | 105 |
| 25.6 | Nasadenie aplikácie .....                                       | 107 |
| 26   | Celkový pohľad .....  | 109 |
| 26.1 | Opis produktu .....   | 109 |
| 26.2 | Architektúra a modely .....                                     | 111 |
| 26.3 | Ohraničenia .....   | 120 |
| 26.4 | Používateľská príručka .....                                    | 120 |
| 26.5 | Podrobné obrazovky produktu .....                               | 130 |

# 1 Úvod

---

Projekt Virtuálnej FIIT nie je na ničím novým. Náš tím pokračoval v rozrobenej práci staršieho tímu 14G, ktorý na mobilnej aplikácii pracoval v školskom roku 2011/2012. Dokument k inžinierskemu dielu obsahuje analýzy, návrhy a implementácie množstva user story podľa metodiky SCRUM s cieľom navrhnuť a realizovať riešenie, aké študenti, zamestnanci a návštevníci budovy FIIT potrebujú.

## 1.1 Účel a rozsah dokumentu

Dokument sa zaoberá analýzou, návrhom, implementáciou user story, testovaním a obsahuje návody na rozbehania aplikácie.

Dokument je výsledkom práce členov tímu ViFiiTeam - Bc. Lukáš Cáder, Bc. Martin Dušek, Bc. Jaroslav Dzurila, Bc. Roland Gášpár, Bc. Martin Londák, Bc. Michal Ševčík a Bc. Matej Toma v rámci predmetu Tímový projekt.

## **2 Stanovenie globálnych cieľov projektu pre zimný semester**

---

V našom tímovom projekte – Virtuálna FIIT sme identifikovali nasledujúce ciele, ktoré chceme počas zimného semestra dosiahnuť.

### **2.1 Zjednotiť grafický dizajn**

Nie všetky časti aplikácie sú graficky zladené do rovnakého štýlu a farebnej schémy. Ikony hlavného menu sú rôznorodého štýlu a jedálne majú úplne inú farebnú schému. Cieľom je úprava fontov, veľkostí, štýlov aby aplikácia všetky časti aplikácie vyzerali jednotne a ponúkali používateľovi čo najväčší komfort.

### **2.2 Dotvoriť aplikačnú logiku**

Ďalším dôležitým aspektom je dodať používateľom funkcionality, ktorú požadujú. Na základe dotazníkov a interných stretnutí tímu sme identifikovali funkcionálne požiadavky, ktorých naimplementovanie považujeme za jeden z hlavných cieľov pre zimný semester.

### **2.3 Google Play**

Ďalším cieľom je dať novú verziu na Google Play a dostať sa do štádia, kedy po každom šprinte sme schopný vyprodukovať novú verziu aplikácie pripravenú na nahranie na Google Play.

### **2.4 Oboznámiť sa s novými technológiami**

Technológie využívané v našom projekte sú pre všetkých členov tímu z časti nové. Prácou na tomto projekte sa oboznámime s úplne novými technológiami ako aj zlepšíme v niektorých známejších. Medzi využité technológie patrí napr. Typescript, Xjade, CSS.

## **2.5 Naučiť sa pracovať v tíme**

Pri väčších projektoch je nevyhnutná práca v tíme, a preto je nesmiernym prínosom získanie takýchto skúseností na predmete tímový projekt. Cieľom každého z nás je zhostiť sa danej roli v tíme zodpovedne a prispieť, k čo najlepšiemu fungovaniu tímu.



## 3 Úlohy pred začatím šprintov

---

### 3.1 Optimalizácia máp budovy fakulty

VIRTFIIT-28

#### 3.1.1 Analýza

Mapy poschodí v stave v akom boli dodané veľmi dlho načítavalo z dôvodu nadbytočného množstva bodov v mapách, ktoré vznikli transformáciou z png do svg formátu. Taktiež boli posunuté dvere a niektoré priečky v miestnostiach boli už neaktuálne.

#### 3.1.2 Návrh

Zarovnať všetky dvere na poschodiach tak, aby žiadne z nich nevyčnievali mimo steny chodby. Taktiež treba skontrolovať či sú totožné priečky v aplikácii s tými, ktoré sú vyobrazené na aktuálnych pôdorysoch.

#### 3.1.3 Implementácia

Pomocou programu Inkscape boli zarovnané umiestnenia dverí do stien a odstránené nadbytočné body, ktoré zbytočne zväčšovali čas načítania mapy. Následne potom prebehla kontrola priečok podľa aktuálne dostupných podkladov.

#### 3.1.4 Testovanie

Testovanie prebehlo pešou prechádzkou po poschodiach budovy, kde sa kontrolovali čísla na miestnostiach, aby sedeli s číslami miestností v aplikácii.

## 4 Opis príbehov 1. šprintu

---

### 4.1 Prerozdelenie spojov MHD

#### VIRTFIIT-56

Ako používateľ chcem prerozdeliť spoje MHD, lebo aktuálne sú pomiešané smery a výpis nie je pre mňa prívetivý a pôsobí chaoticky.

#### 4.1.1 Analýza

Pôvodný stav tejto funkcionality nebol veľmi používateľsky prívetivý. Zoznam odchodov MHD z najbližších zastávok ZOO a Botanická záhrada bol neprehľadný nakoľko voľným okom nebolo možné rozoznať skupinu smerov z jednej a druhej zastávky. Taktiež výpis smerov odchodov bol neprehľadný nakoľko v každom riadku figurovalo slovo „smer“ čo nepôsobilo vizuálne vhodne na človeka.

#### 4.1.2 Návrh

Návrh riešenia, ktoré by vnieslo poriadok medzi takúto neprehľadnosť bolo vizuálne rozdelenie smerov do dvoch skupín podľa smerodajných zastávok. Slovo „smer“ je potrebné nahradiť niečím iným čo by vyjadrovalo podstatu tohto slova.

#### 4.1.3 Implementácia

Pre dosiahnutie tohto cieľa bolo potrebné upraviť štruktúru konfiguračného súboru itransit.json, ktorý obsahuje schému liniek vo formáte JSON tak aby sa docielila lepšia manipulácia so súborom pri jeho vypisovaní. Okrem toho bolo potrebné mierne upraviť aj šablónu routes-list.xjade, ktorá ma za úlohu vypisovať dáta z tohto JSON súboru. Ako posledná vec, ktorú bolo potrebné vykonať bolo pridať pár pravidiel do accordion.less súboru pre naštýlovanie samotnej obrazovky.

#### 4.1.4 Testovanie

Testovanie prebehlo na telefóne Samsung Galaxy Ace GT-S5830i s verziou Androidu 2.3.6. Riešenie je plne funkčné a splňa požadovanú funkcionality.

## 4.2 Zladenie jedálni

### VIRTFIIT-61

Ako používateľ chcem mať jedálne v rovnakých farbách ako zvyšok aplikácie, lebo teraz to neladí s ostatnými časťami aplikácie a pôsobí neprofesionálne.

#### 4.2.1 Analýza

Obrazovka jedální a aj jedálnych lístkov sa graficky príliš líšila od zvyšku aplikácie, takisto boli tieto obrazovky vykreslené prostredníctvom elementu iFrame, ktorý spôsoboval rôzne problémy.

#### 4.2.2 Návrh riešenia

Riešením problému je vytvorenie nových obrazoviek pre zoznam jedální a jedálne lístky, ktoré sú graficky zladené so zvyškom aplikácie.

#### 4.2.3 Implementácia

Dáta ohľadom jedální sa získavajú z JSON súboru hladnystudent.json, dáta ohľadom jedálnych lístkov sa parsujú v reálnom čase prostredníctvom parsera. Požadovaná funkcionálna a dizajn obrazoviek sa docielil vytvorením a úpravou spolu 5 súboroch nachádzajúcich sa v /virtfiit/mobile/src/views/foodmenu. Hlavná funkcionálna je definovaná v typescript súbore foodmenuView.ts, dizajn sa upravoval v .xjade súboroch (foodmenu-view.xjade, /tpl/canteen-list.xjade, /tpl/menu.xjade) a štýly v súbore foodmenu-style.less.

#### 4.2.4 Testovanie

Testovanie prebiehalo priebežne počas vývoja danej funkcionality. Finálny produkt bol testovaný prostredníctvom napísaných testovacích scenárov. Testovalo sa na webovej verzii aplikácie a aj mobilnej.

Webová verzia:

server: nginx 1.4.3

prehliadač:

- Google Chrome – Version 30.0.1599.114
- Mozilla Firefox 25.0
- Internet Explorer 9

Mobilná verzia:

zariadenie: Sony Xperia U

operačný systém: Android 2.3.7

### Testovací scenár č.1 : Prezretie jedálneho lístku na rôzne dni

| Číslo kroku | Popis kroku   | Očakávaná akcia                           | Výsledok |
|-------------|---|---|----------|
| 1.          | Používateľ spustí aplikáciu   | Zobrazí sa hlavné menu aplikácie          | Ok       |
| 2.          | Používateľ vyberie niektorú zo zoznamu jedálni  | Zobrazí sa jedálny lístok na aktuálny deň | Ok       |
| 3.          | Používateľ preklikáva medzi jednotlivými dňami a kontroluje či zobrazené lístky korešpondujú s lístkami na portáli hladnýstudent.sk | Jedálne lístky sa zobrazujú správne       | Ok       |

### Testovací scenár č.2 : Používateľ nemá prístup na internet

| Číslo kroku | Popis kroku                                    | Očakávaná akcia                                       | Výsledok |
|-------------|--|---|----------|
| 1.          | Používateľ spustí aplikáciu                    | Zobrazí sa hlavné menu aplikácie                      | Ok       |
| 2.          | Používateľ vyberie niektorú zo zoznamu jedálni | Zobrazí sa hláška o nutnosti internetového pripojenia | Ok       |

## 4.3 Nová grafika hlavného menu

### VIRTFIIT-50

Ako používateľ chcem vidieť menu v ktorom ikony majú rovnaký štýl a sú rovnako výrazné.

#### 4.3.1 Analýza

Pôvodné ikony jasne vyjadrujú obsah avšak nemajú jednotnú formu. Nedodržiavajú spoločné zásady ani farebnú schému. Nijako nevyjadrujú príslušnosť k 1 aplikácií. Návrh riešenia

Riešením problému je vytvorenie nových obrazoviek pre zoznam jedální a jedálne lístky, ktoré sú graficky zladené so zvyškom aplikácie.

#### 4.3.2 Návrh

Vytvorenie nových ikon podľa stanovenej farebnej schémy.

Aplikácia je aplikáciou FIIT-ky, preto aj ikony musia dodržiavať farebnú schému FIIT, ktorá je modrá a biela. Podľa grafického manuálu STU je modrá farba v logu FIIT RGB = 0, 152,

220 resp. 0098dc. Ďalej bude použitá biela RGB = 255, 255, 255 resp. ffffff a v špeciálnom prípade pre odlíšenie od bielej šedá farba RGB = 230, 230, 230 resp. e6e6e6.

Ikony musia mať rovnaký štvorcový tvar s dĺžkou strany 96px.

### **4.3.3 Implementácia**

### **4.3.4 Testovanie**

Pri externom testovaní boli používatelia schopní orientovať sa bez problémov. Nové ikony sú dostatočne názorné.

## **4.4 Úprava približovania mapy vo webovej verzii máp**

### **VIRTFIIT-58**

Ako používateľ chcem, aby sa posúvanie a približovanie na mape vo webovej verzii správalo rovnako, ako google maps, lebo na toto ovládanie som zvyknutý.

### **4.4.1 Analýza**

Aplikácia Virtuálna FIIT funguje aj ako internetová stránka. Pri prezeraní máp budovy v mobilnej aplikácii sa ich dá posúvať a približovať pomocou dotykov. Toto približovanie a posúvanie však pri prezeraní aplikácie ako internetovej stránky nefunguje.

### **4.4.2 Návrh**

Návrhom riešenia je, že sa umožní približovanie/vzdialenie a posúvanie máp pomocou pohybov myšky a jej scrollovacieho tlačidla.

### **4.4.3 Riešenie**

Riešenie bolo dosiahnuté úpravou súboru mapView.ts. Tento súbor je typu typescript. úprava spočívala v tom, že sa zachytávajú udalosti (eventy) vykonané myšou, konkrétne to boli pohyb myšky a posúvanie scroll tlačidla. Po zachytení týchto udalostí sa zavolali funkcie pre priblíženie a pohyb mapy podľa potreby.

### **4.4.4 Testovanie**

Riešenie je funkčné a vykonáva očakávané reakcie. Riešenie bolo otestované v prehliadačoch Google Chrome a Mozilla Firefox. Riešenie by malo byť funkčné aj v ostatných

prehliadačoch ale v Internet Explorer len od verzie 9. V prehliadači Mozilla Firefox sa môže posúvanie máp vykonávať trhane, toto však je spôsobené tým ako prehliadač pracuje s formátom SVG a nie implementáciou tejto funkcie.

## **4.5 Opraviť názvy odkazov**

### **VIRTFIIT-65**

Ako používateľ chcem, aby názvy odkazov boli čitateľnejšie, lebo nerozumiem často krát ich významu.

#### **4.5.1 Analýza**

Pôvodný stav tejto funkcionality nebol veľmi používateľsky prívetivý. Zoznam všetkých odkazov v aplikácií bol neprehľadný. Niektoré odkazy obsahovali iba skratku a neobsahovali popis, iné zas obsahovali popis ale neobsahovali skratku, iné boli zle napísané.

#### **4.5.2 Návrh**

Návrh riešenia, bol jednoduchý. Bolo potrebné upraviť konfiguračný súbor na výpis odkazov aby bol gramaticky dobre napísaný.

#### **4.5.3 Implementácia**

Pre dosiahnutie tohto cieľa bolo potrebné upraviť štruktúru konfiguračného links.json. Takto urobenú zmenu už iba zostalo nahráť na server.

#### **4.5.4 Testovanie**

Cieľom testovania bolo zistiť, že či odkazy, ktoré ponúkame v aplikácií sú dostatočne zrozumiteľné pre všetkých. Cieľom testovania bolo taktiež zistiť, že či sú dané odkazy napísané gramaticky správne. Používateľ, ktorý kontroloval dané odkazy potvrdil správnosť a zrozumiteľnosť odkazov

## **4.6 Hladké scrollovanie**

### **VIRTFIIT-59**

Ako používateľ chcem, aby sa nezastavilo scrollovanie po obrazovke okamžite po pustení prsta, lebo to nezodpovedá štandardnému správaniu mobilných aplikácií na smartfónoch.

#### **4.6.1 Analýza**

V doterajšom riešení aplikácie nebolo implementované scrollovanie, ani cez externú knižnicu, ani nijak inak. Bol implementovaný iba dotyk prsta a jeho posúvanie scroll panelu. Keďže používatelia sú zvyknutí na možnosť scrollovania vo svojich telefónoch, je nutné túto vlastnosť implementovať.

#### 4.6.2 Návrh

Riešením problému je doprogramovanie už existujúcej touch funkcionality a jej doplnenie o Actionlistener typu “touchend”.

#### 4.6.3 Implementácia

Na všetky dotyky používateľa existujú mechanizmy – listenery – ktoré ich dokážu spracovať a vyhodnotiť, o aký špecifický druh dotyku išlo (touchstart, touchmove, touchend). Bolo potrebné doimplementovať funkcionality do Actionlistener-a touchend a to tak, že sa mu tam pridal posun po skončení dotyku ešte o definovaný počet pixelov. Všetko potrebné je definované a implementované v súbore touchEvent.ts.

#### 4.6.4 Testovanie

Testovanie prebiehalo priebežne počas vývoja danej funkcionality, ako aj finálneho produktu. Testovalo sa najskôr na webovej verzii aplikácie, kde sa zistilo, že browser neemuluje toucheventy. Následne sa teda testovalo na reálnom telefóne s Androidom 2.3.7.

Mobilná verzia:

zariadenie: Sony Xperia U

operačný systém: Android 2.3.7

Tabuľka 1 Scrollovanie v menu Odkazy

| Číslo kroku | Popis kroku  | Očakávaná akcia  |
|-------------|--|--|
| 1.          | Používateľ spustí aplikáciu na mobile                        | Zobrazí sa hlavné menu aplikácie   |
| 2.          | Používateľ vyberie menu Odkazy                               | Zobrazí sa obrazovka s nadefinovanými odkazmi  |
| 3.          | Používateľ prstom po obrazovke prejde nižšie a zodvihne prst | Scrollovanie bude ešte nejakú dobu pokračovať, aj keď prst už nie je na dotykovej ploche |

**Tabuľka 2 Scrollovanie v na obrazovke s rozbalenými MHD spojmi**

| Číslo kroku | Popis kroku  | Očakávaná akcia  |
|-------------|--|--|
| 1.          | Používateľ spustí aplikáciu na mobile                        | Zobrazí sa hlavné menu aplikácie   |
| 2.          | Používateľ vyberie menu MHD                                  | Zobrazí sa obrazovka so zastávkami   |
| 3.          | Používateľ klikne na čo najviac smerov zastávok              | Roztvoria sa menu pre jednotlivé smery s konkrétnymi linkami                             |
| 4.          | Používateľ prstom po obrazovke prejde nižšie a zodvihne prst | Scrollovanie bude ešte nejaké dobu pokračovať, aj keď prst už nie je na dotykovej ploche |



## 5 Opis príbehov 2. šprintu

---

### 5.1 Kontrola správnosti máp budovy

#### VIRFIIT-72

Ako používateľ chcem mať bezchybné mapy budovy, lebo by sa mohlo stať že by som nevedel aj s pomocou tejto aplikácie trafiť do miestnosti, do ktorej chcem.

#### 5.1.1 Analýza

Pôvodný stav tejto funkcionality nebol veľmi používateľsky prívetivý. Pri klikaní na jednotlivé miestnosti na 4.poschodí sa pri miestnostiach, ktoré boli skôr opravované sa nezobrazovali po kliknutí korektné čísla.

#### 5.1.2 Návrh

Prejsť peši všetky poschodia a presvedčiť sa, či čísla v aplikácií sedia s realitou. Taktiež je potrebné implementovať listener na kliknutia na čísla aby aj tie boli klikateľné.

#### 5.1.3 Implementácia

Pre dosiahnutie tohto cieľa bolo potrebné peši prejsť všetky poschodia, či sedia jednotlivé čísla dverí s realitou v aplikácií. Taktiež bolo potrebné elementu `<g>` nastaviť `pointer-events:none;`.

#### 5.1.4 Testovanie

Používateľ, ktorý testoval túto funkcionality skúšal postupne klikať na všetky čísla miestností, či zobrazujú správne číslo, ktoré im je priradené.

### 5.2 Zoradenie spojov MHD podľa obľúbenosti

#### VIRFIIT-89

Ako používateľ chcem, aby sa mi na vrchu obrazovky spojov MHD zobrazovali moje obľúbené spoje MHD a nemusel som ich hľadať v zozname.

#### 5.2.1 Analýza

Pôvodný stav tejto funkcionality bol taký, že odchody autobusov boli usporiadané iba podľa času kedy odchádzajú najbližšie spojenia.

#### 5.2.2 Návrh

Návrh riešenia spočíval v tom aby sa okrem klasického výpisu liniek MHD, zobrazovali aj tzv. obľúbené linky, tj. linky ktorými človek najčastejšie cestuje. Takéto linky by sa

používateľovi zobrazovali na vrchu obrazovky ako prvé a tak by nemusel strácať drahocenný čas klikaním na jednotlivé smery, aby zistil, kedy mu ide najbližšie autobus, alebo električka domov.

### **5.2.3 Implementácia**

Pre dosiahnutie tohto cieľa bolo potrebné upraviť `itransit.ts` model, do ktorého boli pridané funkcie na prácu s `Localstoragom`. Taktiež bola upravená trieda `transportView.ts` aby sa dalo plynulo prepínať medzi pridávaním a odoberaním obľúbenej linky. Miernou úpravou prešla aj šablóna `transport-view.xjade`, do ktorej bol pridaný prepínač na pridávanie a odoberanie linky z obľúbených položiek. Následne sa už iba upravila šablóna `routes-list.xjade` aby vedela vypisovať aj konkrétne obľúbené linky.

### **5.2.4 Testovanie**

Testovanie prebehlo na telefóne Samsung Galaxy Ace GT-S5830i s verziou Androidu 2.3.6. Testovaním sa zistilo, že daná funkcionálnosť bola implementovaná podľa návrhu. Vylepšenie tejto funkcionality do budúcnosti by som videl v tom, že sa bude dať pridať linka, ktorá na svojej trase má dve zastávky z ponúkaných.

## **5.3 Zoradenie spojov MHD podľa času**

### **VIRTFIIT-90**

Ako používateľ chcem, aby boli odchody spojov MHD usporiadané podľa toho, ktorý ide najskôr, lebo ma zaujímajú vždy najskoršie spoje.

#### **5.3.1 Analýza**

Pôvodný stav tejto funkcionality bol taký, že odchody autobusov boli usporiadané podľa čísiel liniek a nie podľa času kedy odchádzajú najbližšie spojenia.

#### **5.3.2 Návrh**

Návrh riešenia, ktoré by usporiadalo linky podľa času odchodu vyzeralo v prvej fáze jednoducho. Pôvodná myšlienka bola taká, že nad daným polom objektov pustím funkciu na utriedenie a bude to hotové. Neskôr sa však ukázalo, že si to bude vyžadovať hlbší zásah do kódu, pochopenia logiky výpisu a upravenie jestvujúcej štruktúry do mierne pozmenenej.

#### **5.3.3 Implementácia**

Pre dosiahnutie tohto cieľa bolo potrebné upraviť šablónu `routes-list.xjade`, ktorá vypisuje údaje na obrazovku. Okrem toho bolo potrebné upraviť aj samotný view `transportView.ts`, v ktorom som si predpripravil dáta aby boli už utriedené pred poslaním do šablóny.

### **5.3.4 Testovanie**

Testovanie prebehlo na telefónoch Samsung Galaxy Ace GT-S5830i s verziou Androidu 2.3.6 a HTC Wildfire S s Androidom vo verzii 2.3.3. Výsledky testov na obidvoch typoch zariadení boli úspešné. Odchody autobusov v dvoch hlavných kategóriách(Zoo a Botanická záhrada) boli usporiadané podľa času najbližších odchodov.

## **5.4 Upraviť zobrazenie MHDčky aby bolo viac farebnejšie**

### **VIRTFIIT-96**

Ako používateľ chcem, aby som vedel lepšie vizuálne rozdeľovať jednotlivé výpisy spojov MHD, lebo teraz to nie je pre mňa prehľadné.

#### **5.4.1 Analýza**

Pôvodný stav tejto funkcionality nebol veľmi používateľsky prívetivý. Zoznam odchodov MHD z najbližších zastávok ZOO a Botanická záhrada bol mierne neprehľadný nakoľko bol fádny lebo boli použité iba základné farby a to biela a čierna. Taktiež rozkliknuté smery obsahovali biely text a tak celkový dojem bol fádny.

#### **5.4.2 Návrh**

Návrh riešenia, ktoré by trochu oživilo obrazovku výpisu bolo pridanie farieb a transformácií textu aby bol vizuálne odlišený od zvyšného textu na obrazovku.

#### **5.4.3 Implementácia**

Bol pridaný rámik okolo zastávky Botanická zastávka a Zoo aby vizuálne odčlenil tieto dve skupiny smerov od seba. Taktiež sa smery jednotlivých zastávok vypisujú modrou farbou, ktorá je použitá aj v menu rámičku. Po rozkliknutí konkrétneho smeru sú čísla liniek a konečných zastávok napísané sice bielou farbou ale je použitý italica aby bol text odlišiteľnejší. Počet minút bol ofarbený červenou farbou aby vynikol na obrazovke. Pri používaní aplikácií na menších rozlíšeniach displeja ako je 320px sa aplikuje zmenšenie textu pre výpis liniek.

#### **5.4.4 Testovanie**

Testovanie prebehlo na vzorke približne desiatich ľudí z radov študentov, ktorý sa vyjadrili pozitívne ku farebným kombináciám, ktoré boli použité v sekcii MHD.

## **5.5 Nezobrazovať prázdny zoznam obľúbených spojov MHD**

### **VIRTFIIT-106**

Ako používateľ nechcem, aby sa zobrazoval prázdny zoznam obľúbených spojov MHD, lebo to zbytočne zaberá priestor na obrazovke, ktorý patrí normálnemu výpisu spojov.

### 5.5.1 Analýza

Pôvodná verzia obsahovala sekciu „Oblíbené linky“ aj keď žiadne neboli zatiaľ pridané medzi oblíbené.

### 5.5.2 Návrh

Pokiaľ si používateľ neuložil žiadnu linku medzi oblíbené tak by sa sekcia „Oblíbené linky“ nemala zobrazovať.

### 5.5.3 Implementácia

Bola pridaná podmienka výpisu do šablóny routes-list.xjade, ktorá zabezpečuje to, že keď v telefóne neexistujú žiadne uložené linky tak sa sekcia „Oblíbené linky“ nebude zobrazovať.

### 5.5.4 Testovanie

Testovanie prebehlo na telefóne HTC Wildfire S s Androidom vo verzii 2.3.3. Testovaním bolo potvrdené, že daná chyba bola odstránená.

## 5.6 Úprava dizajnu jedální

### VIRTFIIT-64, VIRTFIIT-93

Ako používateľ chcem zvýšiť kvalitu výpisu jedální, lebo obsahuje mnoho chýb a aj nezaujímavé zbytočné informácie o jedálňach na druhom konci mesta.

#### 5.6.1 Analýza

Obrazovka jedální potrebuje zvýšenie kvality po grafickej stránke i použiteľnej. Medzi hlavné problémy aktuálnej verzie patria:

- nevyhovujúci dizajn
- používateľ nevidí pri výbere jedální dokedy je daná jedáleň otvorená
- zoznam jedální obsahuje ekonomické jedálne, ktoré študenti FIIT nepotrebujú
- pri jedálnych lístkoch sa zobrazujú dni v anglickom jazyku
- tabuľka s jedlami sa nezobrazuje na mobilnom zariadení správne

O daných jedálňach je potrebné používateľovi zobrazit' viacero informácií, ako lokalitu danej jedálne, otváracie hodiny a iné.

#### 5.6.2 Návrh

Spraviť modernejší dizajn s využitím hlavnej farebnej palety aplikácie (čierna, šedá, modrá). V zozname jedální budú pridané otváracie časy na konkrétny deň, pri obrazovke konkrétnych jedální budú dni zobrazené po slovensky a prídruže informačné tlačidlo, po ktorého stlačení sa zobrazí obrazovka s otváracími hodinami, odkazom na mapu a domovskou stránkou danej jedálne.

### 5.6.3 Implemetácia

Zmena dizajnu sa docielila upravením štýlov jednotlivých elementov v súbore .less. Viacerým elementom sa pridal atribút width:100% aby sa prispôbili obrazovkám rôznych rozmerov, takisto sa upravili farby na hodnoty podľa potreby na black, grey a deepskyblue(#00a9e0). Úpravou canteen-list.xjade súboru sa pod názvy jedálni v zozname pridali otváracie hodiny na konkrétny deň. Prepínanie jedálnych bolo zmenené v súbore menu.xjade. Do hlavnej lišty aplikácie pri zobrazení obrazovky s jedálnym lístkom jedálne pribudne tlačidlo pre zobrazenie dodatočných informácií o tejto jedálni. Pre tento účel bol vytvorený nový info.xjade súbor, ktorý obsahuje výpis otváracích hodín a odkaz na domovskú stránku jedálne. Tieto dáta sa získavajú z JSON súboru, konkrétne ide o elementy: opening\_times, homepage. Na info obrazovke sa takisto zobrazí odkaz na google maps, kde sa zobrazí daná jedáleň na základe jej gps koordinátov. Tieto koordináty sú získané zo spomínaného JSON súboru a pridané do url daného odkazu. Správna funkčnosť zobrazovania a skrývania info obrazovky bola dosiahnutá úpravou hlavného typescript súboru jedálni.

### 5.6.4 Testovanie

Testovanie prebiehalo priebežne počas vývoja danej funkcionality jej vývojárom, otestovaný bol aj finálny produkt. Testovalo sa na webovej verzii aplikácie a aj mobilnej.

Webová verzia:

lokálny server: nginx 1.4.3

prehliadač:

- Google Chrome – Version 30.0.1599.114
- Mozilla Firefox 25.0
- Internet Explorer 9

Mobilná verzia:

zariadenie: Sony Xperia U

operačný systém: Android 2.3.7

Testovací scenár č.1 : Prezretie jedálneho lístku na rôzne dni

| Číslo kroku | Popis kroku                                    | Očakávaná akcia                           | Výsledok |
|-------------|--|---|----------|
| 1.          | Používateľ spustí aplikáciu                    | Zobrazí sa hlavné menu aplikácie          | Ok       |
| 2.          | Používateľ vyberie niektorú zo zoznamu jedálni | Zobrazí sa jedálny lístok na aktuálny deň | Ok       |
| 3.          | Používateľ preklikáva medzi jednotlivými dňami | Jedálne lístky sa zobrazujú správne       | Ok       |

## Testovací scenár č.2 : Prezretie informácií o jedálni

| Číslo kroku | Popis kroku  | Očakávaná akcia  | Výsledok |
|-------------|--|--|----------|
| 1.          | Používateľ spustí aplikáciu                                | Zobrazí sa hlavné menu aplikácie   | Ok       |
| 2.          | Používateľ vyberie niektorú zo zoznamu jedální             | Zobrazí sa jedálny lístok na aktuálny deň                                      | Ok       |
| 3.          | Používateľ klikne na tlačidlo „i“ v hornej lište aplikácie | Zobrazí sa obrazovka z otváracími hodinami, odkazom na mapu a domovskú stránku | Ok       |
| 4.          | Používateľ klikne na „zobrazit' na mape“                   | Otvorí sa nové okno v prehliadači so zobrazenou lokalitou na mape              | Ok       |
| 5.          | Používateľ mapu zatvorí a zobrazí domovskú stránku jedálne | Domovská stránka jedálne sa zobrazí v novom okne                               | Ok       |

## 5.7 Mapa okolia fakulty

### VIRTFIIT-112

Ako nový študent alebo návštevník chcem vidieť na aplikácii cestu k zastávkam MHD jedálňam a bufetom v okolí, lebo to okolie nepoznám

#### 5.7.1 Analýza

V okolí našej fakulty sa nachádza veľké množstvo budov, bufetov a zastávok MHD. Noví študenti a návštevníci sú často krát stratený a majú problém nájsť všetky dôležité miesta.

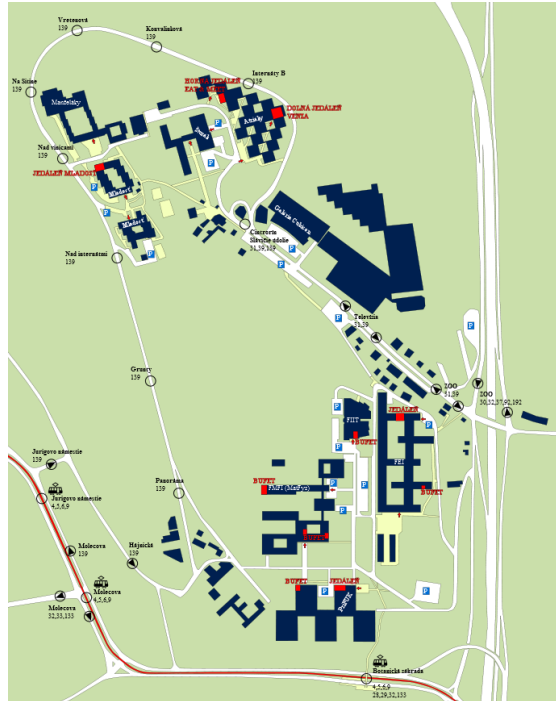
#### 5.7.2 Návrh

Návrh riešenia spočíva v pridaní mapy okolia, kde by boli zástavky električiek, autobusov ktoré z niekoľkoročných skúseností poznám, že ich študenti používajú, všetky jedálne a bufety na našej fakulte, okolitých fakultách a internátoch v Mlynskej doline.

#### 5.7.3 Implementácia

Implementácia spočívala vytvorení SVG mapy podľa satelitných snímok prevzatých z Google Maps. Kde sa nachádzajú jedálne, bufety a zástavky MHD som zistil zo stránky iMHD.sk,

domovských stránok jednotlivých jedální a z príručiek pre nových študentov susedných fakúlt. Na mape sú vyznačené navyše koľajnice električiek, chodníky a vstupy budov.



Obrázok 1 Ukážka výsledku vytvorenia mapy

## 5.7.4 Testovanie

Testovanie správnosti mapy prebiehalo osobne tým, že som navštívil miesta na mape.

## 5.8 RSS správy

### VIRTFIIT-91

Ako používateľ chcem mať v aplikácii po ruke najnovšie správy fakulty, lebo chcem sledovať najdôležitejšie informácie o nej.

### 5.8.1 Analýza

Fakulta FIIT poskytuje na svojich stránkach informácie o aktualitách na fakulte. Tieto informácie poskytuje aj vo forme XML dokumentu, kde ich všetky združuje na jednom mieste. Používatelia našej aplikácie určite budú mať záujem o tieto informácie v našej mobilnej aplikácii. Ide o aktuálne dianie a okrem názvu a opisu eventu sa v XML dokumente nachádza aj webový odkaz na stránku s viac informáciami.

## 5.8.2 Návrh

Riešením bude pridať do menu Odkazov podmenu pre RSS správy. Po kliknutí na ne sa otvorí nová obrazovka, kam sa načítajú stiahnuté RSS správy z webovej lokality. Tie budú obsahovať aj odkaz na externú stránku s viac informáciami.

## 5.8.3 Implementácia

Väčšina implementácie prebieha v priečinku views/links, kde bolo treba pozmeniť všetky súbory. Do .less sa pridala nová trieda pre nadpis novej RSS obrazovky. V súbore .xjade som pridal novú obrazovku, kde sa vykreslia správy. Súbor .ts obsahuje funkcionality, ktorá prepína medzi obrazovkami, zobrazuje a schováva ich. Taktiež sa v tom súbore načítajú správy a pošlú sa na rendering. V priečinku TPL som pridal .ts triedu pre samotný rendering.

Bolo nutné k tomu všetkému spraviť XML parser pre samotné XML dáta zo stránky, ktoré treba upraviť a pripraviť na zobrazenie.

## 5.8.4 Testovanie

Testovanie prebiehalo priebežne počas vývoja danej funkcionality, ako aj finálneho produktu. Testovalo sa hlavne na webovej verzii aplikácie (väčšina času). Občas sa to testovalo aj na reálnom telefóne s Androidom 2.3.7.

Webová verzia

vd'aka technológii nginx

v prehliadači Chrome

Mobilná verzia:

zariadenie: Sony Xperia U

operačný systém: Android 2.3.7

### Testovací scenár č. 1 : Zobrazenie RSS správ

| Číslo kroku | Popis kroku                           | Očakávaná akcia                               |
|-------------|---------------------------------------|---|
| 1.          | Používateľ spustí aplikáciu na mobile | Zobrazí sa hlavné menu aplikácie              |
| 2.          | Používateľ vyberie menu Odkazy        | Zobrazí sa obrazovka s nadefinovanými odkazmi |



|    |   |   |
|----|---|---|
| 3. | Používateľ sa dostane až dole v zozname a klikne na menu RSS správy | Zobrazí sa mu obrazovka s nadpisom RSS správy a s obsahom správ |
|----|---|---|

**Testovací scenár č. 2 : Získanie viac informácií o výseku z RSS správy**

| <b>Číslo kroku</b> | <b>Popis kroku</b>  | <b>Očakávaná akcia</b>   |
|--------------------|---|--|
| 1.                 | Používateľ spustí aplikáciu na mobile                               | Zobrazí sa hlavné menu aplikácie   |
| 2.                 | Používateľ vyberie menu Odkazy                                      | Zobrazí sa obrazovka s nadefinovanými odkazmi  |
| 3.                 | Používateľ sa dostane až dole v zozname a klikne na menu RSS správy | Zobrazí sa mu obrazovka s nadpisom RSS správy a s obsahom správ                                  |
| 4.                 | Používateľ klikne na ľubovoľnú správu zo zoznamu                    | Bude presmerovaný v prehliadači na príslušnú stránku na fiit.stuba.sk, kde nájde viac informácií |

## 6 Opis príbehov 3. šprintu

---

### 6.1 Obnoviť zoznam obľúbených spojov pri kroku späť

#### VIRTFIIT-121

Ako používateľ chcem, aby sa zoznam obľúbených spojov obnovil pri kroku späť, lebo sa mi teraz nezobrazujú zmeny, ktoré urobím pred stlačením kroku späť.

#### 6.1.1 Analýza

Pôvodný stav bol taký, že keď sa používateľ nachádzal na obrazovke odchodov konkrétnej linky, pridal si takúto linku medzi svoje obľúbené a použil naše tlačidlo späť tak sa jeho pridaná linka nenachádzala na obrazovke. Až po obnove stránky použitím ikonky domčeka a opätovného návratu do zoznamu liniek MHD sa zobrazila daná linka medzi obľúbenými.

#### 6.1.2 Návrh

Pri použití tlačidla „späť“ z obrazovky zoznam odchodov linky na celkový zoznam liniek je potrebné vykonať aktualizáciu obrazovky „celkový zoznam liniek“

#### 6.1.3 Implementácia

Na dosiahnutie aktualizácie obrazovky pri použití tlačidla späť z obrazovky „zoznam odchodov linky“ na obrazovku „celkový zoznam liniek“ som zmenil implementáciu tohto tlačidla aby vykonal znovu vykreslenie obrazovky.

#### 6.1.4 Testovanie

Testovanie prebehlo na telefóne Samsung Galaxy Ace GT-S5830i s verziou Androidu 2.3.6. Používateľ, ktorý vykonával testovanie skúšal striedavo pridávať a odoberať jednotlivé linky z obľúbených aby zistil, že či sa obrazovka so všetkými linkami MHD aktualizuje.

## 7 Ostatné úlohy 3. šprintu

---

### 7.1 Konfigurácia webovej verzie virtfiit aby bolo možné sa pripojiť na vzdialený server (AIS, HladnyStudent, stuba, itransit)

VIRTFIIT-55

#### 7.1.1 Analýza

Ak program bol spustený na webovom prehliadači, server (nginx) sa nedokázal pripojiť k inému resp. k jeho api. Tento problém bol spôsobený lebo nebol server správne nakonfigurovaný.

#### 7.1.2 Riešenie

V zložke /etc/nginx/sites-enabled/virtfiit sa pridali nasledujúce riadky:

```
location /api/hladnystudent/ {
    rewrite ^/api/hladnystudent(.*)$ // $1 break;
    proxy_pass http://hladnystudent.zones.sk;
}

location /api/fiit {
    rewrite ^/api/feedrzs/(.*)$ // $1 break;
    proxy_pass http://fiit.stuba.sk;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host fiit.stuba.sk;
}
```

Tieto isté prístupy sa vytvorili pre ais a itransit. Rewrite vždy určuje kam sa v programe snažíme pripojiť a vďaka proxy\_pass nám povolí prístup. Ak sa jedná o prihlásenie na api (ais, itransit) prostredníctvom autorizácie, pridáme nasledujúce dva riadky:

```
proxy_set_header Authorization $http_authorization;
proxy_pass_header Authorization;
```

### 7.1.3 Testovanie

Po pridaní nasledujúcich riadkov aplikácia pracovala správne bez chybových hlášok.

## 7.2 Nefunkčná história a zmiešaná obrazovka pri QR kódoch

VIRTFIIT-107

**Dátum reportu o chybe:**14.11.2013

**Sekcia:** Odfot' kód

**Zariadenie, na ktorom sa chyba deteguje:** Sony Xperia U

**Verzia OS Android:**2.3.7

Po kliknutí na možnosť Odfot' kód v hlavnom menu aplikácie sa spustí skener QR kódov. Po úspešnom načítaní QR kódu sa načítané údaje vyučujúceho prekrývajú s hlavným menu aplikácie. Takisto po kliknutí na miestnosť a stlačení tlačidla späť sa používateľ nevráti na obrazovku s kontaktnými údajmi ale na hlavnú obrazovku aplikácie.

### 7.2.1 Analýza

Aplikácia funguje princípom „Single Page“, čo znamená, že reálne je iba jedna stránka na ktorej sa skrývajú a odkrývajú jednotlivé elementy. Problém zo zmiešanou obrazovkou je spôsobený neskrýťím elementov hlavnej obrazovky. Čo sa týka problému s históriou, pri zobrazení načítaných údajov sa novo zobrazená stránka nevloží do histórie stránok.

### 7.2.2 Návrh

Problém so zmiešanou obrazovkou je možné vyriešiť zmenou priehľadnosti pozadia zobrazených údajov o vyučujúcom, a problém s históriou vytvorením novej metódy, ktorá do histórie aplikácie (zapamätaný stav zobrazených a skrytých elementov) vloží zobrazenú stránku.

### 7.2.3 Implementácia

Upravil som pozadie súboru timetable-view.less s priehľadného na čierne (background: black) a v súbore /src/lib/virtfiit/view.ts pridal metódu PushPath(path), ktorej argumentom je stav obrazovky, ktorý chceme uložiť do histórie. Túto metódu som následne zavolať po zobrazení obrazovky s načítanými údajmi s QR kódu.

## 7.2.4 Testovanie

Tabuľka 3 Odfotenie QR

| Číslo kroku | Opis kroku  | Očakávaná akcia                              | Výsledok |
|-------------|---|--|----------|
| 1.          | Používateľ klikne na Odfot' kód                             | Spustí sa snímač QR kódov                    | OK       |
| 2.          | Používateľ zosníma QR kód obsahujúci informácie o učiteľovi | Korektne sa zobrazia informácie o vyučujúcom | OK       |
| 3.          | Používateľ klikne na číslo miestnosti vyučujúceho           | Zobrazí sa daná miestnosť na mape            | OK       |
| 4.          | Používateľ klikne v aplikácii na tlačidlo späť              | Zobrazia sa informácie o vyučujúcom          | OK       |

## 8 Opis príbehov 4. šprintu

---

### 8.1 Doplnenie predčítavania dát pre jedálne a MHD

#### VIRTFIIT-124, VIRTFIIT-125

Ako používateľ chcem, aby sa dali jedálne lístky a odchody MHD prezerat' na niekoľko dní dopredu aj bez internetového pripojenia.

#### 8.1.1 Analýza

Aplikácia Virtuálna FIIT poskytuje možnosť prezerania si jedálnych lístkov vo vybraných jedálňach a tiež možnosť prezerania si odchodov MHD z dvoch najbližších zastávok od fakulty. Na ich prezeranie je však potrebné internetové pripojenie a bez neho tieto funkcionality nemajú veľkú použiteľnosť.

#### 8.1.2 Návrh

Návrhom riešenia je, že sa pri prezeraní MHD alebo jedálnych lístkov predčítajú do pamäte na niekoľko dní dopredu.

#### 8.1.3 Riešenie

Bolo naprogramované také riešenie, že ak sa prvýkrát začne prezerat' MHD a aplikácia má pripojenie na internet tak sa automaticky predčítajú spojenia na týždeň dopredu. Takto isto funguje aj predčítavanie jedálnych lístkov. Aby sa predčítali je však potrebné zvolit' konkrétnu jedáleň a len pre ňu sa predčíta lístok na týždeň dopredu. Ak chceme predčítat' lístky pre ďalšiu jedáleň tak si ju musíme zvolit' z menu. Počet dní, na koľko sa predčítajú lístky alebo MHD je pevne zadané. Tu je priestor na vylepšenie, že by si počet dní mohol používateľ sám zvolit'. V súbore „itransit.ts“ bola vytvorená funkcia „prefetchSchedules()“, ktorá predčítava odchody MHD a ukladá ich do „local storage“. Táto funkcia sa volá v súbore „transportView.ts“. Takým istým spôsobom sú spravené aj jedálne, kde sa vytvorila funkcia „prefetchSchedules()“ v súbore „hladnystudent.ts“ a táto funkcia sa volá zo súboru „foodmenuView.ts“.

#### 8.1.4 Testovanie

Riešenie je funkčné a vykonáva očakávané reakcie. Ak sa predčítajú jedálne lístky alebo MHD tak sa potom dajú prezerat' na týždeň dopredu aj bez internetového pripojenia.

## 8.2 Logovanie aktivity používateľa

### VIRTFIIT-129

Ako vývojár chcem vedieť, kde v aplikácii používateľ klikal a čo robil, lebo mi to pomôže pri reprodukcii nájdených chýb a ďalšom zlepšovaní aplikácie.

#### 8.2.1 Analýza

Používatelia aplikácie nájdu pri používaní ne jeden bug, či chybu iného charakteru. Častokrát je to iba náhodne a používateľ si nepamätá, ako by sa tá chyba dala zreprodukovať. Pribežným logovaním jeho krokov v aplikácii sa dá prispieť k ľahšiemu reprodukovaniu nájdených bugov a zrýchliť tak prácu vývojárov.

#### 8.2.2 Návrh

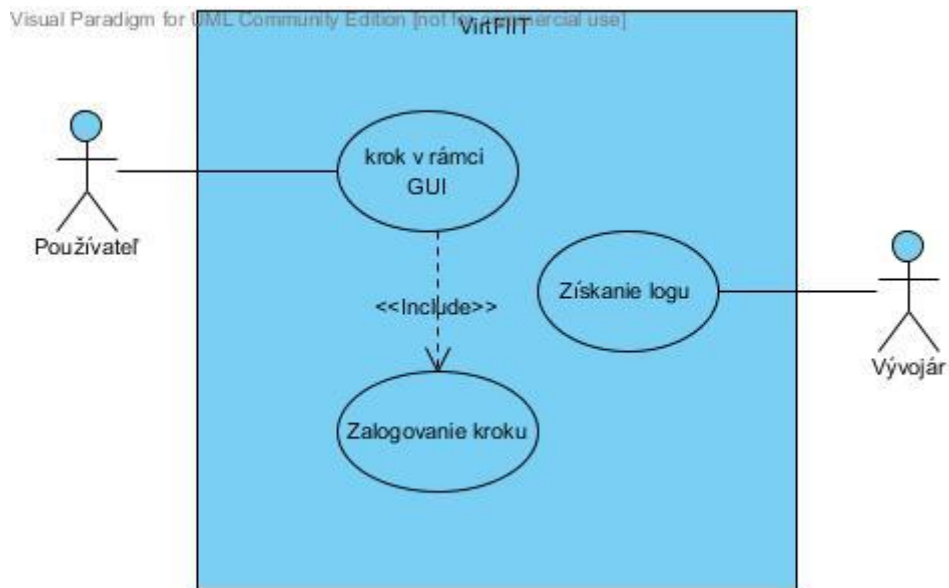
Vytvorí sa logovacia trieda v jazyku Typescript, tá bude zapisovať dáta do localstorage-u, kde sa budú uchovávať. Logovacie výpisy budú pridané do jednotlivých tried v rámci celej aplikácie, aby sa dala zrekonštruovať cesta používateľa v aplikácii.

#### 8.2.3 Implementácia

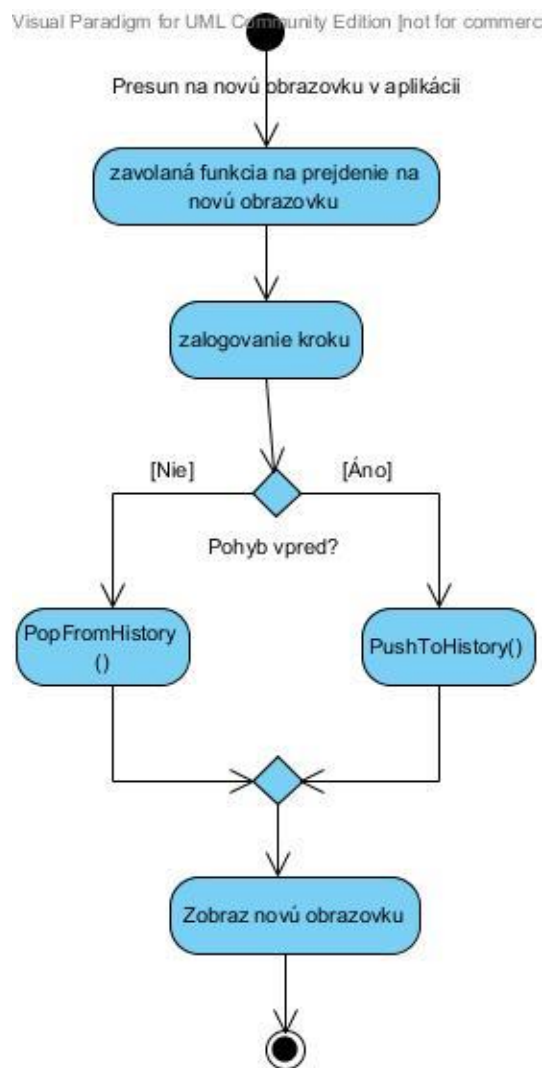
Bola vytvorená trieda `log.ts` a v nej metóda `log(level, message)`, ktorá zapíše argumenty spolu s aktuálnym dátumom a časom do localstorage-u. V každej cieľovej `.ts` triede sa vytvorí inštancia logu ( `var logger = Log.getInstance()` ) a vo vhodných miestach sa zavolá metóda `log()`, do ktorej sa ako argumenty dá informácia, kde v aplikácii sa práve nachádzame, a nejaká prídavná informácia.

#### 8.2.4 Testovanie

Overiť sa táto funkcionálnosť dá iba pre vývojárov, ktorí si dokážu pri behu aplikácie pozrieť výpis, čo sa aktuálne nachádza v localstorage-i.



Obrázok 2 Use case diagram logovania používateľovej aktivity



Obrázok 3 Activity diagram prechodu používateľa na novú obrazovku v rámci aplikácie



## 8.3 Pridanie informačného dialógu pri použití QR kódu

### VIRTFIIT-139

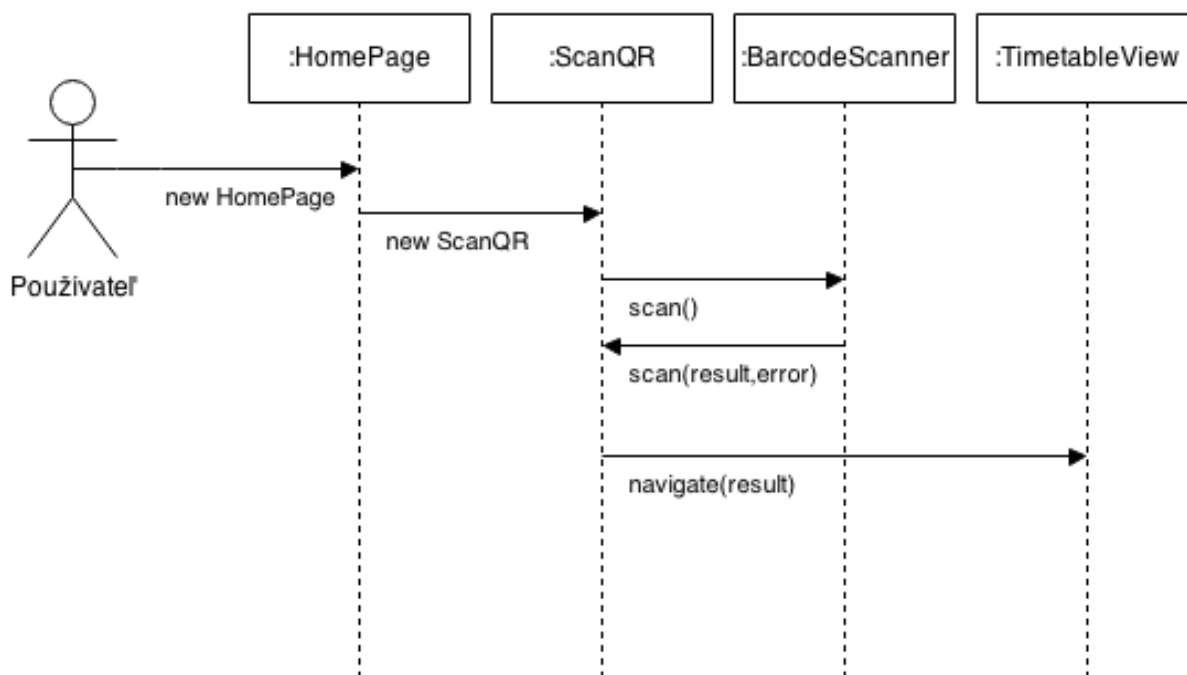
Ako používateľ, ktorý používa aplikáciu, chcem vedieť načo slúži možnosť Odfot' kód na hlavnej obrazovke aplikácie.

#### 8.3.1 Analýza

Mnoho používateľov pri používaní alebo testovaní aplikácie nevedelo načo slúži možnosť Odfot' kód. Používateľom musí byť ozrejmnený význam danej funkcionality.

#### 8.3.2 Návrh

Používateľovi sa pri prvom použití funkcie odfotenía QR kódu zobrazí informačný dialóg popisujúci načo tieto QR kódy slúžia.



Obrázok 4 Sekvenčný diagram naskenovania QR kódu

Informačný dialóg sa zobrazí pri vytvorení objektu triedy ScanQR.

### 8.3.3 Implementácia

V súbore app.ts bol v bloku kódu, ktorý sa vykoná po kliknutí na Odfot' kód pridaný nasledujúci kód:

```
if (! localStorage.getItem('app.qr-showed')) {  
    Dialogs.alert("Funkcia Odfot' kód slúži na snímanie QR kódov v budove fakulty.  
    Kódy rozmiestnené pri jednotlivých miestnostiach obsahujú informácie o pozícii danej  
    miestnosti, ako aj vyučujúcom v tejto miestnosti, jeho kontaktných údajoch a jeho  
    rozvrhu.")  
    .then(()=> { localStorage.setItem('app.qr-showed','true')  
                this.scanqr.scan() })  
}  
else this.scanqr.scan()
```

Ak sa v local storage zariadenia nenachádza príznak app.qr-showed znamená to, že používateľ funkciu zosnímania QR kódu ešte nepoužil. Po prvom spustení tejto funkcie sa používateľovi zobrazí informačný dialóg popisujúci význam QR kódov a do local storagu sa uloží príznak app.qr-showed. Funkcia scanqr.scan() vytvorí objekt snímača QR kódov.

### 8.3.4 Testovanie

Tabuľka 4 Funkcia odfot' kód

| Číslo kroku | Opis kroku                                      | Očakávaná akcia   | Výsledok |
|-------------|---|---|----------|
| 1.          | Používateľ klikne po prvý krát na Odfot' kód    | Zobrazí sa dialóg informujúci o význame QR kódov v aplikácii  | OK       |
| 2.          | Používateľ klikne na OK                         | Spustí sa snímač QR kódov                                     | OK       |
| 3.          | Používateľ stlačí hardvérové tlačidlo Späť      | Zobrazí sa informačná obrazovka                               | OK       |
| 4.          | Používateľ stlačí opäť hardvérové tlačidlo Späť | Zobrazí sa hlavná obrazovka                                   | OK       |
| 5.          | Používateľ klikne znova na možnosť odfot' kód   | Informačný dialóg sa už nezobrazí a spustí sa snímač QR kódov | OK       |

## 8.4 Zoradenie obľúbených spojov MHD podľa časov odchodu

### VIRTFIIT-126

Ako používateľ, ktorý každý deň používa MHD chcem aby obľúbené linky boli utriedené podľa času odchodu, lebo chcem vedieť kedy odchádza najbližšie linka.

#### 8.4.1 Analýza

Zistil som, že ľudia omnoho lepšie reagujú na MHD v aplikácií, keď sú časy ich obľúbených liniek zoradené podľa času odchodu najbližšej linky a nie podľa toho ako boli pridané do zoznamu obľúbených.

#### 8.4.2 Návrh

Záznamy, ktoré sa vyberú z localstoragu aplikácie a idú na výstup do šablóny routes-list.xjade, je potrebné pred ich vypísaním usporiadať podľa času najbližšieho odchodu v súbore transportView.ts.

#### 8.4.3 Implementácia

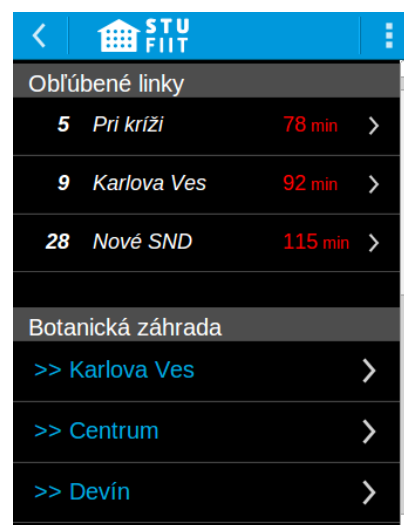
Bol implementovaný usporiadovací algoritmus v metóde sortFavouriteLinks(links) v súbore transportView.ts, ktorý zabezpečuje to, že keď sa nachádza medzi obľúbenými zastávkami MHD viac ako 1 záznam tak sa dané linky usporiadajú podľa času najbližšieho odchodu. Danú funkciu volá getAllFavourites a ako parameter mu posielajú obľúbené linky.

#### 8.4.4 Testovanie

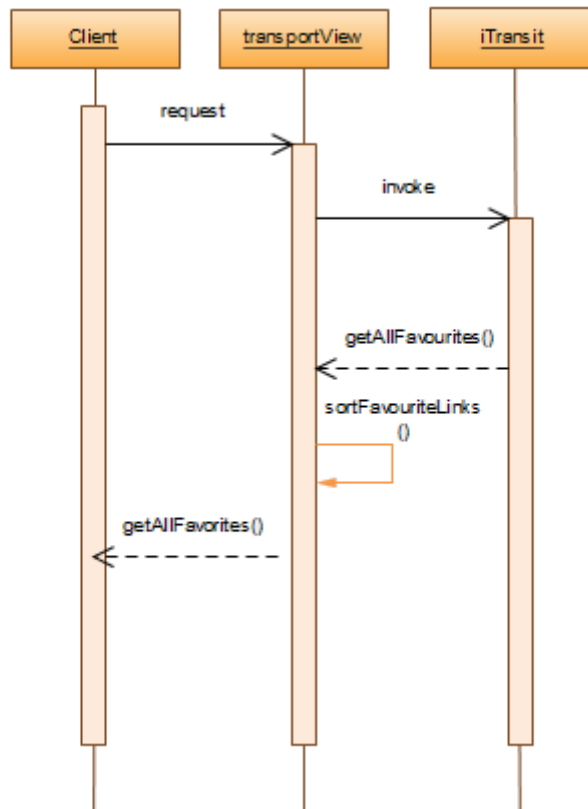
Testovanie prebehlo na telefónoch HTC Desire S s Android verziou 2.3.3 a HTC Galaxy Ace s Android verziou 2.3.6. Cieľom testu bolo preukázať, že pridávaním a odoberaním liniek do obľúbeného zoznamu linky zostávajú stále usporiadané podľa času odchodov.



Obrázok 5: Stav pred úlohou



Obrázok 6: Stav po úlohe



Obrázok 7: Usporiadanie obľúb. mhd podľa času

## 9 Ostatné úlohy 4. šprintu

---

### 9.1 Čierna obrazovka pri využití funkcie skenovania QR kódu

VIRTFIIT-63

**Dátum reportu o chybe:** 16.11.2013

**Sekcia:** Odfot' kód

**Zariadenie, na ktorom sa chyba deteguje:** Sony Xperia U

**Verzia OS Android:** 2.3.7

Po kliknutí na možnosť Odfot' kód v hlavnom menu aplikácie sa spustí skener QR kódov. Po stlačení hardvérového tlačidla späť alebo úspešnom rozpoznaní QR kódu sa namiesto presunu do hlavného menu resp. zobrazovania rozpoznaných informácií zobrazí čierna obrazovka.

#### 9.1.1 Analýza

Aplikácia využíva na skenovanie QR kódov plugin do phonegap-u BarcodeScanner. Problém bol identifikovaný nasledovne: po úspešnom načítaní kódu alebo po vystúpení z obrazovky snímača kódu sa nespustia metódy, ktoré sú zadané aby sa spustili pri týchto akciách. Je pravdepodobné že chyba nastala niekde v plugine resp. v komunikácii medzi aplikáciou a pluginom.

#### 9.1.2 Návrh

Spôsob ako takéto správanie úplne vyriešiť je buď skúsiť iné verzie snímača QR kódov, alebo počkať na vydanie novej verzie oficiálneho pluginu. Ako dočasné riešenie som sa rozhodol zameniť čiernu obrazovku za obrazovku s popisnou informáciou o stlačení hardvérového tlačidla späť pre pokračovanie.

#### 9.1.3 Implementácia

Pred samotným vytvorením skenovacieho objektu som pridal metódu, ktorá vykreslí zvolenú obrazovku, teda namiesto čiernej obrazovky sa po načítaní/nenačítaní kódu zobrazí informačná obrazovka. Túto obrazovku som vytvoril nasledovne:

- vytvorením súboru scanqrView.ts v priečinku /src/views/scanqr v ktorom je zadaná funkcionálnosť danej obrazovky, a pridaním odkazu na tento súbor do app.ts

- vytvorením súboru scanqr-view.xjade, ktorý je zodpovedný za vykreslenie obsahu danej obrazovky, a pridaním odkazu na tento súbor do index.xjade
- vytvorením súboru scanqr-view.less, ktorý zodpovedá za css štýlovanie obrazovky, a zadaním tohto súboru v style.less

### 9.1.4 Testovanie

Tabuľka 5 Neodfotenie QR

| Číslo kroku | Opis kroku  | Očakávaná akcia                 | Výsledok |
|-------------|---|---------------------------------|----------|
| 1.          | Používateľ klikne na Odfot' kód                   | Spustí sa snímač QR kódov       | OK       |
| 2.          | Používateľ stlačí hardvérové tlačidlo Spät'       | Zobrazí sa informačná obrazovka | OK       |
| 3.          | Používateľ stlačí opät' hardvérové tlačidlo Spät' | Zobrazí sa hlavná obrazovka     | OK       |

Tabuľka 2 Odfotenie QR

| Číslo kroku | Opis kroku  | Očakávaná akcia                             | Výsledok |
|-------------|---|---|----------|
| 1.          | Používateľ klikne na Odfot' kód                             | Spustí sa snímač QR kódov                   | OK       |
| 2.          | Používateľ zosníma QR kód obsahujúci informácie o učiteľovi | Zobrazí sa informačná obrazovka             | OK       |
| 3.          | Používateľ stlačí hardvérové tlačidlo Spät'                 | Zobrazia sa načítané informácie o učiteľovi | OK       |

## 9.2 Testovanie usability aplikácie

### VIRTFIIT-88

Sústavný proces počas vývoja aplikácie.

### 9.2.1 Analýza

Cieľom používateľských testov je overenie či sa aplikácia správa podľa očakávaní používateľa a či používateľ dokáže identifikovať a používať ponúkanú funkcionálnosť.

- Použitie vyhľadávania
- Označenie verzie
- Použitie rozvrhu
- Použitie máp
- Požitie jedální
- Použitie odkazov
- Použitie QR kódov

A overiť nápravu bugov.

### 9.2.2 Návrh

Navrhnuté sú zadania testovacích úloh:

- Vyhľadávanie
  - Vyhľadať podľa mena Mgr. Aleny Kovárovej jej kanceláriu na mape. Vyhľadať podľa mena Prof. Márie Bielikovej jej kanceláriu na mape. Porovnaj.
- Použitie rozvrhu
  - Zistiť z rozvrhu miesto kde sa bude konať prvá rozvrhová akcia vo štvrtok.
- Použitie máp
  - Zistiť z mapy nasledujúcu rozvrhovú akciu v danej miestnosti (-1.57).
- Požitie jedální
  - Zistiť najlepší jedálny lístok z jedálne Eat & Meet.
- Použitie odkazov
  - Zistiť cez odkazy otváracie hodiny študijného oddelenia.
- Použitie QR kódov
  - Načítať určený QR-kód (Bieliková\_Mária\_1871.vcf.png) z monitora. Zobraz na mape jej kanceláriu.
  - Načítať určený QR-kód (Kovárová\_Alena\_38836.vcf.png) z monitora. Zobraz na mape jej kanceláriu.
- Zistiť verziu aplikácie a databázy.

Na záver testu tester vyplní dotazník o spokojnosti. S možnosťami hodnotenia zle-1, slabé-2, dobre-3, lepšie-4 a výborne-5.

### 9.2.3 Implementácia

Každý test prebieha samostatne v prítomnosti správcu testu, ktorý zodpovedá za zaznamenanie priebehu testu, komunikáciu zadaní testovacích úloh testerovi. Tester je postupne oboznámený a plní zadania úloh. Správca pozoruje aplikované postupy a porovnáva ich s predpokladanými postupmi.

### 9.2.4 Testovanie

Tester pri testovaní mali len malé problémy orientovať sa v aplikácií. Noví používatelia aplikácie spočiatku využívali najmä zabudované funkčné tlačidlá smartfónov pre návrat späť. Experimentovaním však boli schopní nájsť funkcionality na navigovanie v aplikácií už pri druhom pokuse.

Tester boli schopní rýchlo zistiť zadané informácie.

Problém spôsobovalo najmä potreba pripojenia na internet pri využívaní služby jedálny lístok.

Tester hodnotili aplikáciu:

**Tabuľka 6 Priemerné hodnotenie aplikácie a funkcionality testerami**

| <b>Otázka:<br/>Ako hodnotíš...</b> | <b>Bodové hodnotenie</b> | <b>Slovné hodnotenie</b> |
|------------------------------------|--------------------------|--------------------------|
| Aplikáciu                          | 3                        | dobré                    |
| Skrolovanie                        | 4.5                      | lepšie                   |
| Hlavné menu                        | 5                        | výborné                  |
| Mapy                               | 4                        | lepšie                   |
| Jedálne                            | 5                        | výborné                  |
| Vyhľadávanie                       | 5                        | výborné                  |
| QR                                 | 5                        | výborné                  |
| Odkazy                             | 4                        | lepšie                   |
| Rozvrhy                            | 3                        | dobré                    |



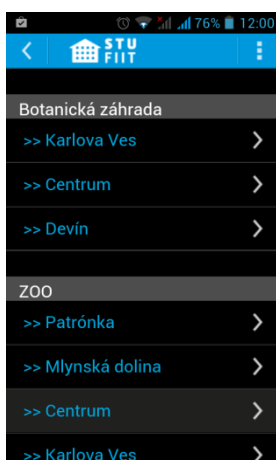
## 9.3 Testovanie rozdelenia MHD

### VIRTFIIT-87

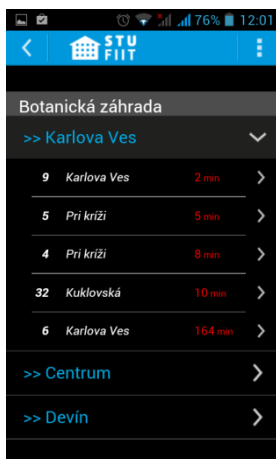
Zmena rozdelenia a usporiadania položiek v menu služby MHD vyžadovala overenie užívateľskými aplikačnými testami.

#### 9.3.1 Analýza

Cieľom používateľských testov je overenie či sa aplikácia správa podľa očakávaní používateľa a či používateľ dokáže identifikovať a používať ponúkanú funkcionality na vyhľadávanie spojov MHD.



Obrázok 8 MHD úroveň 1



Obrázok 9 MHD úroveň 2

| Line | 0  | 4  | 9  | 17 | 25 | 33 | 41 | 49 | 57 |
|------|----|----|----|----|----|----|----|----|----|
| 07   | 0  | 4  | 9  | 17 | 25 | 33 | 41 | 49 | 57 |
| 08   | 5  | 13 | 21 | 29 | 37 | 45 | 53 |    |    |
| 09   | 1  | 9  | 17 | 25 | 33 | 41 | 49 | 57 |    |
| 10   | 5  | 13 | 21 | 29 | 37 | 45 | 53 |    |    |
| 11   | 1  | 9  | 17 | 25 | 33 | 41 | 49 | 57 |    |
| 12   | 5  | 13 | 21 | 29 | 37 | 45 | 53 |    |    |
| 13   | 1  | 9  | 17 | 25 | 33 | 41 | 49 | 57 |    |
| 14   | 5  | 13 | 21 | 29 | 37 | 45 | 53 |    |    |
| 15   | 1  | 9  | 17 | 25 | 33 | 41 | 49 | 57 |    |
| 16   | 5  | 13 | 21 | 29 | 37 | 45 | 53 |    |    |
| 17   | 1  | 9  | 17 | 25 | 33 | 41 | 49 | 57 |    |
| 18   | 5  | 13 | 21 | 29 | 37 | 45 | 53 |    |    |
| 19   | 1  | 16 | 31 | 46 |    |    |    |    |    |
| 20   | 0  | 15 | 30 | 45 |    |    |    |    |    |
| 21   | 0  | 15 | 30 | 44 | 59 |    |    |    |    |
| 22   | 14 | 29 | 44 | 59 |    |    |    |    |    |
| 23   | 14 |    |    |    |    |    |    |    |    |

Obrázok 10 MHD úroveň 3

V rámci testu sa používateľ - tester, musí pokúsiť získať z aplikácie informáciu, o ktorej vieme postup ako ju získať. Testerov postup získavania informácie bude zaznamenaný a analyzovaný.

### 9.3.2 Návrh

Navrhnuté sú zadania testovacích úloh:

1. Vyhľadať najbližší spoj idúci zo Zoo na Hlavnú stanicu. Pridaj spoj medzi oblúbené. Vyhľadaj nasledujúci najbližší spoj idúci zo Zoo na Hlavnú stanicu. Pridaj spoj medzi oblúbené. Vyhľadať najbližší spoj idúci zo Zoo na kuklovskú. Pridaj spoj medzi oblúbené. Určiť z oblúbených spojov poradie v akom spoje pôjdu.
2. Vyhľadať linku 32 zo Zoo na Hlavnú stanicu a pridať ju medzi oblúbené. Vyhľadať linku 32 z Botanickej na Hlavnú stanicu pridať medzi oblúbené. Porovnať časy odchodov.  
Vyhľadať linku 32 zo Zoo na Kuklovskú a pridať ju medzi oblúbené. Vyhľadať linku 32 z Botanickej na Kuklovskú pridať medzi oblúbené. Porovnať časy odchodov.

Na záver testu tester vyplní dotazník o spokojnosti.

### 9.3.3 Implementácia

Každý test prebieha samostatne v prítomnosti správcu testu, ktorý zodpovedá za zaznamenanie priebehu testu, komunikáciu zadaní testovacích úloh testerovi. Tester je postupne oboznámený a plní zadania úloh. Správca pozoruje aplikované postupy a porovnáva ich s predpokladanými postupmi.

### 9.3.4 Testovanie

Tester pri testovaní MHD nemali problém orientovať sa v aplikácii. Boli schopný bez problémov a rýchlo zistiť zadané informácie a Službu hodnotili pozitívne.

## 9.4 Príprava ikon pre budúcu funkcionálnosť

### VIRTFIIT-127

Ako používateľ aplikácie a študent fakulty FIIT potrebujem rýchly a zrozumiteľný prístup do Akademického Informačného Systému (AIDS), k fakultnému RSS a aj iným zdrojom informácií.

#### 9.4.1 Analýza

Funkcionálnosť aplikácie bude pravdepodobne v budúcnosti preorganizovaná a bude rozšírená hlavná ponuka. Pre zachovanie jednotnosti dizajnu a pre vizualizáciu nového dizajnu je vhodné pripraviť ikony vopred.

Do hlavnej ponuky prídu ikony pre:

- AIS – Akademický informačný systém
- RSS – Novinky na fakulte
- Informácie – Informačné zdroje

#### 9.4.2 Návrh

Pre RSS a Informácie existujú všeobecne uznávané symboly, ktoré je potrebné použiť.



Obrázok 11 Ukážka štýlu ikon informačného obsahu. Zdroj: google.com

Pre ikony informácií je typické písmeno „i“.



Obrázok 12 Ukážka štýlu ikon RSS. Zdroj: google.com

RSS má svoje zaužívané logo.

Pre AIS neexistuje jedno zaužívané logo alebo symbol, preto bude stanovený vlastný. Vhodným symbolom je školský zošit so záložkami.

### 9.4.3 Implementácia

Ikony musia dodržiavať stanovenú farebnú schému a štýl aplikácie.



Obrázok 13 Ikona vytvorená pre Informácie



Obrázok 14 Ikona vytvorená pre RSS



Obrázok 15 Ikona vytvorená pre AIS

#### 9.4.4 Testovanie

Dizajn ikon bol testovaný predvedením na stretnutí tímu. Tím vyhodnotil ikony ako dostatočne výpovedné a dodržiajúce stanovený grafický štýl aplikácie.

### 9.5 Chybné mapovanie najbližších odchodov MHD

#### VIRTFIIT-83

**Dátum reportu o chybe:** 02.11.2013

**Sekcia:** MHD

**Zariadenie, na ktorom sa chyba deteguje:** VŠETKY

**Verzia OS Android:** VŠETKY

Keď som si dal zobrazit' najbližšie odchody zo zastávky Zoo tak linka č. 32 mala rovnaký čas odchodu ako zo zastávky Botanická záhrada. Podľa cestovného poriadku by mala linka č. 32 odchádzať zo zastávky Botanická záhrada o minútu neskôr ako zo zastávky Zoo na konečnú zastávku Kuklovska.

#### 9.5.1 Analýza

V aplikácií existuje problém, že keď si používateľ otvorí odchod linky č. 32 zo zastávky Zoo v smere do Karlovej vsi tak je totožný s odchodom rovnakej linky zo zastávky Botanická záhrada taktiež v smere do Karlovej vsi. Tieto údaje sú nesprávne. Linka č. 32 by mala zo zastávky Botanická záhrada odchádzať neskôr ako zo zastávky Zoo. Problém teda spočíva v mapovaní odchodov liniek. Tento problém sa však prejavuje iba na linke č. 32 a to z toho dôvodu, že jedine táto linka v našej aplikácií odchádza z dvoch zastávok v jednom smere a z dvoch zastávok v opačnom smere. Keď by sa v aplikácií nachádzalo viacero podobných liniek tak problém by bol viac viditeľnejší. Zlé mapovanie vzniklo preto, lebo jednotlivé linky boli málo špecifikované. Konkrétny cestovný poriadok sa vyberal len na základe *route\_id* čiže jedinečného identifikátora danej linky a *dir\_id* (identifikátor smeru - konečnej zastávky).

## 9.5.2 Návrh

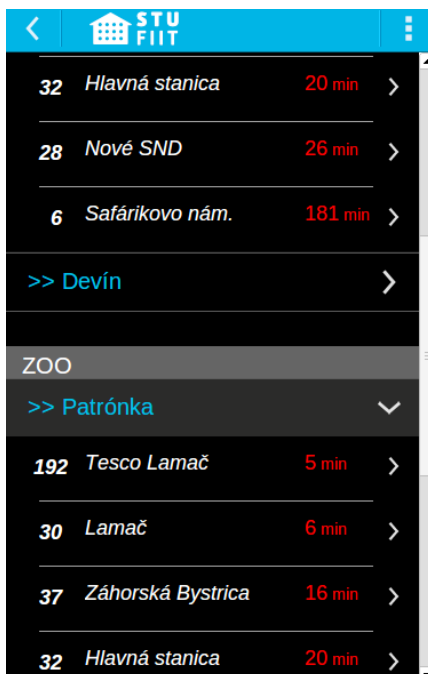
Návrh ako opraviť toto mapovanie by spočíval v tom aby sa cestovný poriadok pre danú linku nešpecifikoval len na základe *route\_id* a *dir\_id* ale aby bol zohľadnený aj *parent\_id* (identifikátor zastávky) pre daný spoj v konkrétnom smere.

## 9.5.3 Implementácia

Jednou z príčin chybného mapovania bolo zlé nastavenie konečných zastávok linky č. 32 v súbore *mobile/src/data/itransit.json*. Z tohto dôvodu boli zamenené cestovné poriadky v smere na Hlavnú stanicu so smerom na Kuklovskú. Keďže záznamy neboli úplne jednoznačné, tak bol pridaný identifikátor zastávky (*parent\_id*), ktorý sa načítava v metóde *getGroups()* v súbore *mobile/src/lib/virtfiit/itransit.ts*. Takto pripravené dáta potom stačilo už iba v súbore *mobile/src/views/transport/transportView.ts* správne namapovať. Bol pridaný atribút *parent\_id*, ktorý do pola *routes* na kľúč *schedule* pomocou funkcie *\_find(...)* mapuje jednotlivé cestovné poriadky pre konkrétnu linku (*route\_id*) na daný smer (*dir*) a z konkrétnej zastávky (*parent\_id*).

## 9.5.4 Testovanie

Testovaním sa preukázalo, že daná chyba bola opravená a že sa už ďalej neprejavuje. Testovanie prebehlo na telefónoch s nainštalovanou rôznou verziou operačného systému Android. Na všetkých nami otestovaných zariadeniach to fungovalo správne tak ako to zobrazuje obrázok 2.



Obrázok 17: Stav pred opravou



Obrázok 16: Stav po oprave

## **9.6 Nemožnosť pridať medzi obľúbené linku z rôznych zastávok v jednom smere**

**VIRTFIIT-128**

**Dátum reportu o chybe:** 21.11.2013

**Sekcia:** MHD

**Zariadenie, na ktorom sa chyba deteguje:** VŠETKY

**Verzia OS Android:** VŠETKY

Po kliknutí na ikonku hviezdičky na linke 32 zo zastávky Botanická záhrada smerom na Kuklovskú sa pridala do zoznamu obľúbených liniek. Keď som chcel však k nej pridať aj linku 32 zo zastávky Zoo v smere na Kuklovskú tak sa nič neudialo. Linka sa nepridala. Keď som klikol na jej odobranie zo zoznamu tak zostala stále v tom zozname.

### **9.6.1 Analýza**

Po bližšom skúmaní kódu som zistil, že dôvod prečo sa linka č.32 nepridávala bol ten, že linka zo zastávky Zoo nebola rozlíšiteľná od linky č. 32 zo zastávky Botanická záhrada. Z tohto dôvodu v lokálnej databáze bol uchovaný iba jeden záznam a nie ako by tomu malo v skutočnosti byť.

### **9.6.2 Návrh**

Spôsob ako by sa dal vyriešiť tento problém, spočíval v tom, že by sa pridal ďalší identifikátor na spresnenie z akej zastávky daná linka odchádza.

### **9.6.3 Implementácia**

Problém som vyriešil tak, že okrem identifikátora linky a smeru bol pridaný aj identifikátor zastávky z ktorej linka odchádza.

### **9.6.4 Testovanie**

Riešenie problému bolo otestované na telefóne HTC Wildfire S s Android verziou 2.3.3 a Samsung Galaxy ACE s Androidom 2.3.6. Testovaním sa ukázalo, že daný problém bol úspešne vyriešený.



Obrázok 18: Stav pred opravou



Obrázok 19: Stav po oprave

## 9.7 Nesprávne zobrazovanie miestnosti na mobilných zariadeniach

VIRTFIIT-122

Dátum reportu o chybe: 27.11.2013

Sekcia: MAPY

Zariadenie, na ktorom sa chyba deteguje: VŠETKY

Verzia OS Android: VŠETKY

Pri vyhľadani učiteľa a zobrazení jeho miestnosti sa prvý krát zobrazí miestnosť správne, no po tomto zobrazení sa už nezobrazí žiadna iná miestnosť a nedá sa prepínať ani medzi podlažiami.

### 9.7.1 Analýza

Správanie som overil na viacerých zariadeniach a všade bolo rovnaké.

### 9.7.2 Návrh

Opraviť som sa to rozhodol v zdrojovom kóde v súbore „./src/lib/virtfiit/mapCanvas.ts“ a to tak, že som si prešiel celý jednotlivé riadky za pomoci debugovacieho nástroja weinre.



### **9.7.3 Implementácia**

V kóde som našiel chybu pri metóde `deselectRoom`, kde bola zlá podmienka na zistenie či je miestnosť vybraná a taktiež bolo zlé odznačenie označenej miestnosti.

### **9.7.4 Testovanie**

Kód som skontroloval na 2 mobilných zariadeniach s verziami androidu (2.3.4 a 2.3.6) a následne sa dala aplikácia otestovať trom žiakom. Chyba sa viac už nevyskytla.

## 10 Opis príbehov 5. šprintu

---

### 10.1 Logovanie aktivity používateľa, pokračovanie

#### VIRTFIIT-129

Ako vývojár chcem vedieť, kde v aplikácii používateľ klikal a čo robil, lebo mi to pomôže pri reprodukcii nájdených chýb a ďalšom zlepšovaní aplikácie.

#### 10.1.1 Analýza

Používatelia aplikácie nájdu pri používaní nejaký bug, či chybu iného charakteru. Často je to iba náhodne a používateľ si nepamätá, ako by sa tá chyba dala zreprodukovať. Pribežným logovaním jeho krokov v aplikácii sa dá prispieť k ľahšiemu reprodukovaniu nájdených bugov a zrýchliť tak prácu vývojárov. Okrem toho pomáha testerom aplikácie pri vyhodnocovaní krokov, ktoré používateľ pri usability testovaní spraví.

#### 10.1.2 Návrh

Vytvorí sa logovacia trieda v jazyku Typescript, tá bude zapisovať dáta do localStorage-u, kde sa budú uchovávať. Logovacie výpisy budú pridané do jednotlivých tried v rámci celej aplikácie, aby sa dala zrekonštruovať cesta používateľa v aplikácii.

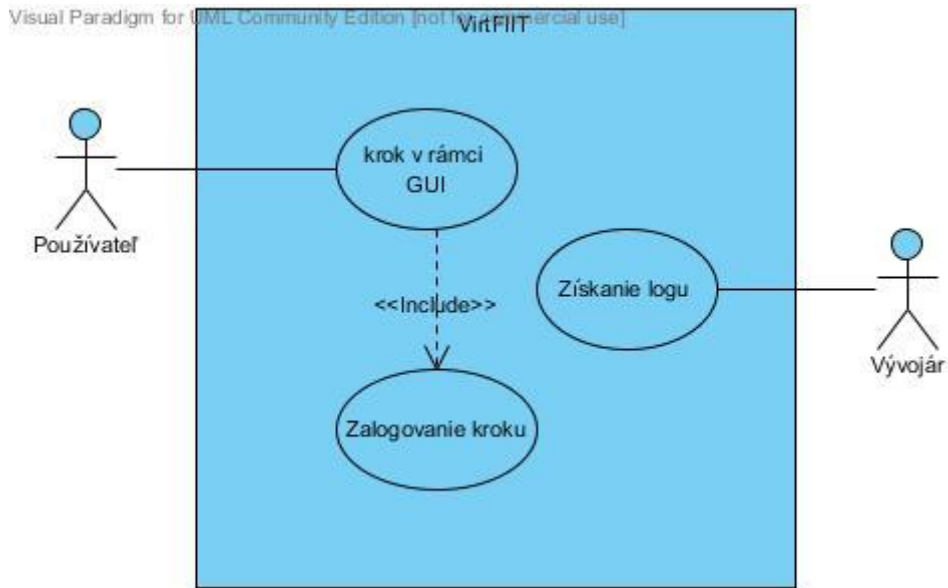
#### 10.1.3 Implementácia

V predchádzajúcom šprinte bola vytvorená trieda log.ts a v nej metóda log(level, message), ktorá zapíše argumenty spolu s aktuálnym dátumom a časom do localStorage-u. V každej cieľovej .ts triede sa vytvorí inštancia logu ( var logger = Log.getInstance() ) a vo vhodných miestach sa zavolá metóda log(), do ktorej sa ako argumenty dá informácia, kde v aplikácii sa práve nachádzame, a nejaká prídavná informácia.

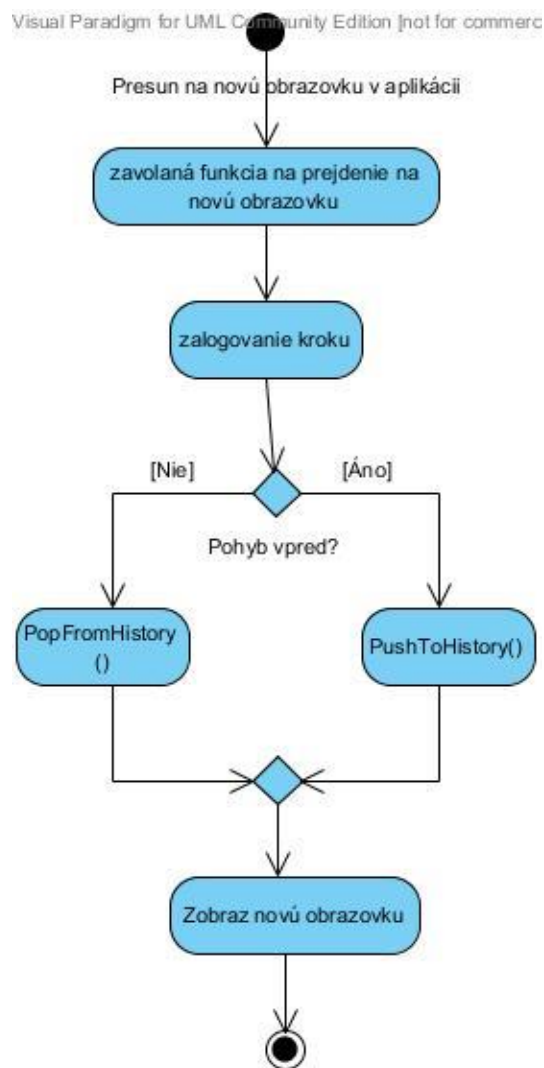
Rozsah tried, na ktoré sa logovanie rozšírilo, je iTranzit, jedálne, odkazy, tlačidlo Späť, externé URL odkazy, na ktoré sa kliklo, QR kódy a rôzne obrazovky v rámci AIS-u.

#### 10.1.4 Testovanie

Overiť sa táto funkcionálnosť dá iba pre vývojárov, ktorí si dokážu pri behu aplikácie pozrieť výpis, čo sa aktuálne nachádza v localStorage-i.



Obrázok 20 Use case diagram logovania používateľovej aktivity



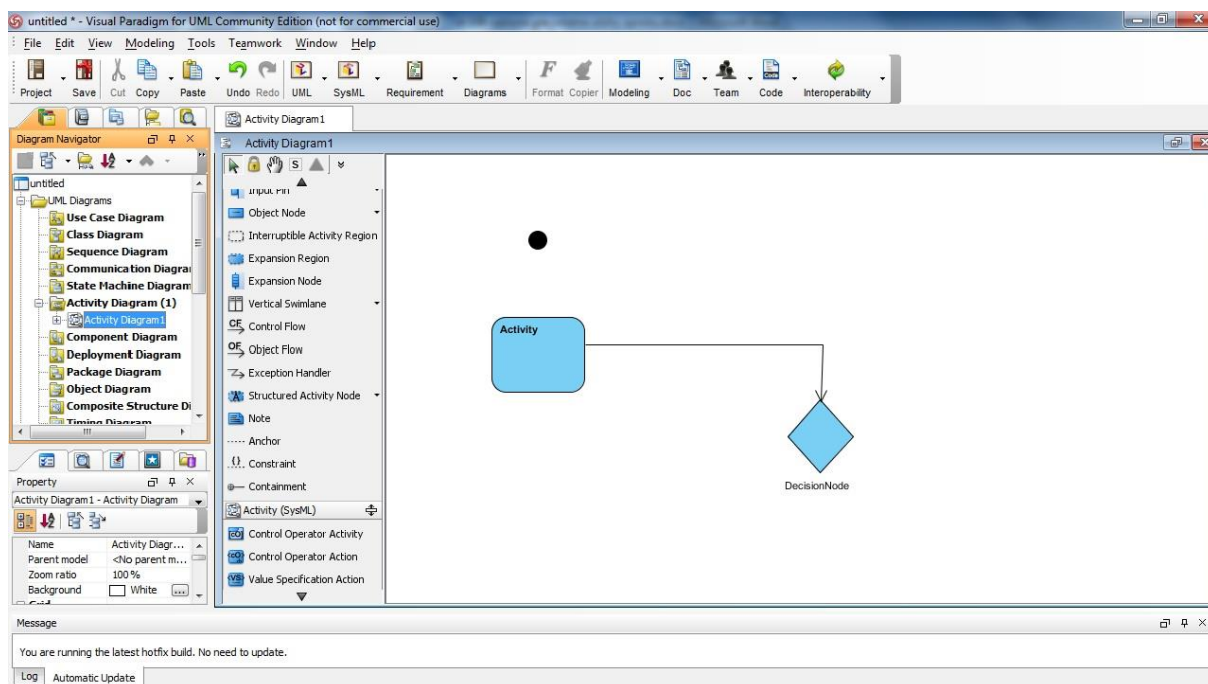
Obrázok 21 Activity diagram prechodu používateľa na novú obrazovku v rámci aplikácie

# 11 Ostatné úlohy 5. šprintu

## 11.1 Nástroj Visual Paradigm for UML

### 11.1.1 Opis nástroja

Visual Paradigm for UML je nástroj na tvorbu UML notácií. Je dostupný na stiahnutie zo stránky <http://www.visual-paradigm.com/>. Je možné v ňom vytvoriť všetky bežne používané UML diagramy. Jedná sa o platený softvér. Je k dispozícii aj verzia zadarmo pre nekomerčné účely. Treba si však zaregistrovať aj túto verziu a pri prvom spustení aplikácie pýta informácie ako meno, priezvisko a emailovú adresu. Po zaregistrovaní príde na adresu kľúč, ktorý treba zadať v samotnom programe. Aplikácia je už teraz plne využiteľná na tvorbu UML. Avšak ak sa chce z programu importovať nejaký diagram vo forme obrázku tak sa na ňom objaví vodotlač, ktorá oznamuje, že je ten program použiteľný len pre nekomerčné účely.



Obrázok 22 Ukážka programu Visual Paradigm

## **11.2 Pridanie informácie kto je zodpovedný za miestnosť**

### **VIRTFIIT-4**

#### **11.2.1 Analýza**

Po kliknutí na niektorú z miestností na mape sa na obrazovke zobrazujú len niektoré informácie. Medzi informácie, ktoré sa zobrazujú patria informácie o čísle miestnosti, popis o akú miestnosť ide, v akej budove sa miestnosť nachádza a rozvrh danej miestnosti. Tieto informácie sa zobrazujú v prípade ak sa jedná o niektorú z učební. Keď používateľ klikne na niektorú z miestností, ktorá je kancelária vyučujúceho tak sa zobrazia informácie: číslo miestnosti, názov budovy v ktorej sa miestnosť nachádza a o koho kanceláriu ide. Taktiež je možné obidva typy miestností zobrazit' na mape.

#### **11.2.2 Návrh**

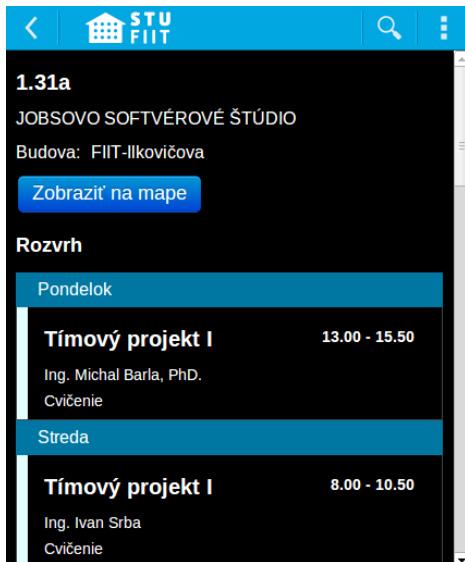
Bolo by vhodné keby sa na obrazovke, ktorá informuje o akú miestnosť ide zobrazovala informácia o tom, kto je správcou danej miestnosti. Táto informácia je užitočná v prípadoch keď nastal problém v danej miestnosti a používateľ nevie na koho sa má obrátiť.

#### **11.2.3 Implementácia**

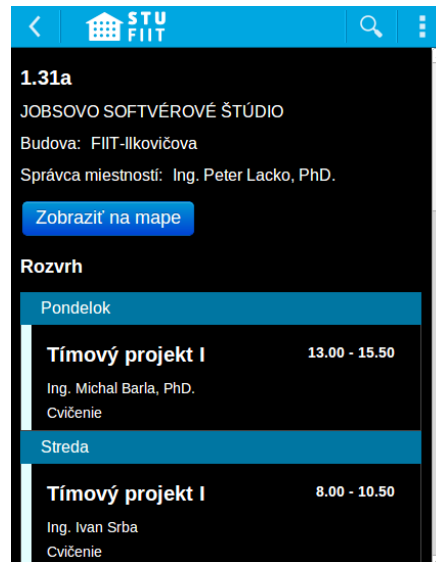
Doplnil som skript (run.js), ktorý sa nachádza v priečinku scripts/aisdump o pár riadkov kódu, aby z AISu sťahoval informáciu aj o tom kto je správcou miestnosti. Avšak nie každá miestnosť musí mať svojho správcu. Preto takáto informácia sa bude zobrazovať len pri niektorých miestnostiach. Taktiež som upravil šablónu (rooms.xjade) v priečinku src/views/ais/tpl, tak aby zobrazovala informáciu o tom kto je správcou danej miestnosti.

#### **11.2.4 Testovanie**

Testovanie prebehlo úspešne na rôznych verziách Androidu a na rôznych zariadeniach. Žiadne chyby neboli objavené počas testovania. Daná obrazovka sa správa korektne.



Obrázok 23: Stav úlohy pred



Obrázok 24: Stav úlohy po

## 12 Opis príbehov 6. šprintu

---

### 12.1 Zobrazovanie SVG a PNG ikon na základe verzie Androidu

#### VIRTFIIT-183

Ako používateľ chcem aby sa mi na zariadení s veľkou uhlopriečkou a jedným z najnovších Androidov zobrazovali vektorové ikony, lebo klasické rastrové sú rozmazané.

#### 12.1.1 Analýza

Doteraz naša aplikácia používala pri zobrazovaní ikon na zariadeniach iba typ PNG. Toto riešenie je ale nedostačujúce pre zariadenia s veľkou uhlopriečkou, kde sú zobrazené PNG rozťahnuté na potrebnú veľkosť a rozmazané.

#### 12.1.2 Návrh

Pre každú použitú ikonu sa spraví jej vektorový náprotivok. Do kódu aplikácie sa pridá test, či dané zariadenie/prehliadač podporuje vektorové súbory. Ak áno, zobrazí sa vektorová ikona, inak sa zobrazí tá klasická.

#### 12.1.3 Implementácia

Pre túto úlohu vypracoval Jaroslav Dzurilla vektorové ikony ku všetkým klasickým PNG ikonám použitých v aplikácii. Tieto ikony boli prekopírované do priečinkov, kde sídlia ich PNG verzie. Čo sa týka kódu, museli sa upraviť LESS súbory pre štýl obrazovky, ktorý zobrazoval PNG a to tak, že sa vytvorila nová CSS trieda, v ktorej sa nahradili cesty k PNG ikonám za cesty k SVG. Nakoniec sa musel upraviť Javascript-ový súbor Modernizr.js tak, že sa do neho pridal nový test, či zariadenie má podporu pre SVG. Ak áno, načítala sa CSS trieda pre SVG ikony.

#### 12.1.4 Testovanie

Testovanie prebehlo iba na webovej verzii aplikácii (cez Chrome), nakoľko som nemal k dispozícii iné zariadenie, ktoré by malo dostatočne nový Android, ktorý by podporoval SVG.

## **12.2 Vylepšiť aktualizáciu údajov pri MHD**

Niektorý používatelia, ktorý majú internetové pripojenie a nekliknú na MHD, tak sa im automatický nestiahnu údaje.

### **12.2.1 Analýza**

Keď ma používateľ internet a neklikne na MHD, tak nemá nové údaje MHD. Preto je potrebné vytvoriť automatické sťahovanie údajov, ktoré by pomohlo stiahnuť tieto údaje aj keď používateľ neklikne na linku MHD.

### **12.2.2 Návrh**

Každú pol hodinu sa bude kontrolovať či zariadenie má k dispozícii internetové pripojenie, a ak má tak stiahne nové údaje bez toho aby o tom používateľ vedel.

### **12.2.3 Implementácia**

V súbore `app.ts` som pridal metódu `public update()`, ktorá kontroluje internetové pripojenie a ak je dostupné, tak zvolá `prefetchSchedules` z triedy `Itransit`, ktorá automaticky postúhuje všetky údaje (táto istá metóda sa zvolá pri kliknutí na MHD). Aby sa táto kontrola volala, je použitá `java script` metóda `setInterval()`, v ktorej sú ako argumenty použité: funkcia na `update` a čas v milisekundách (toto spôsobí, že sa daná metóda spustí vždy po uplynutí zadaného času).

### **12.2.4 Testovanie**

V spomínanej metóde som si dal vypisovať `debug` výpis, ktorý mi potvrdil že táto metóda sa spúšťa.



## 12.3 Implementácia posielania logov z aplikácie na server

Pri používaní našej aplikácie sa v pozadí generuje história logov. Aby pre nás tieto logy mali zmysel, je potrebné ich uchovávať na servere v databáze.

### 12.3.1 Analýza

Dáta logov nie sú extra štruktúrované, takže nám postačí noSQL databáza (mongo db).

### 12.3.2 Návrh

Pri kliknutí na možnosť nahlásiť chybu, odoslať, sa cez ajax post odošlú logy na našu stránku [team06-13.ucebne.fiit.stuba.sk/save.php](http://team06-13.ucebne.fiit.stuba.sk/save.php) kde bude php skript, ktorý uloží odoslané dáta do databázy.

### 12.3.3 Implementácia

Ako prvé bolo treba nainštalovať mongo-db. Aby bolo možné prijímať tieto log správy, je taktiež potrebné nainštalovať php. Po nainštalovaní týchto dvoch vecí, je potrebné ich prepojiť, tj. treba nainštalovať modul do php aby dokázal komunikovať s databázou.

Po týchto základných inštaláciách treba na strane klienta vytvoriť odosielanie. V súbore /libs/log.ts v metóde sendLogs(msg) som vybral všetky logované správy z lokálnej databázy a vytvoril som jeden veľký stringu ktorý obsahoval aj odosielanú správu. Takto vytvorený string odosielam cez ajax POST na spomínanú linku [team06-13.ucebne.fiit.stuba.sk/save.php](http://team06-13.ucebne.fiit.stuba.sk/save.php).

Na strane servera sa nachádza php skript save.php ktorý získa pomocou GET[] informácie, a uloží ich do databázy (db.logy).

### 12.3.4 Testovanie

Odoslal som zopár logov, a skontroloval som cez wireshark či boli odoslané. Taktiež som pozrel v databáze či boli správne prijaté.

## **12.4 Otváracie hodiny na študijnom oddelení**

### **VIRTFIIT-169**

Ako používateľ chcem vedieť, kedy sú úradné hodiny na študijnom oddelení.

#### **12.4.1 Analýza**

Aplikácia zobrazuje na mape budovy miestnosť, kde sa nachádza študijné oddelenie. Doteraz sa však na túto miestnosť nedalo kliknúť a ani nikde v aplikácii sa nenachádzali informácie o úradných hodinách na študijnom oddelení.

#### **12.4.2 Návrh**

Návrhom riešenia je získanie údajov o otváracích hodinách na študijnom oddelení a tieto informácie vypisovať po kliknutí na miestnosť alebo po vyhľadani študijného oddelenia v aplikácii.

#### **12.4.3 Implementácia**

Pri implementácii bol upravený skript „run.js“, ktorý sa spúšťa na serveri a aktualizuje informácie dostupné v systéme AIS. Do tohto skriptu bol pridaný parser (funkcia „getStudyDepartment()“) na získanie informácií o študijnom oddelení. Po tom, ako sa tieto informácie stiahnu sa otváracie hodiny pridajú položky „description“ pre miestnosť „1.06“. Tiež bola upravená mapa prvého poschodia aby sa dalo kliknúť na miestnosť, kde sa nachádza študijné oddelenie.

Tieto údaje sa ukladajú v súbore „rooms.json“ a tiež v „search.json“, aby sa študijné oddelenie dalo v aplikácii vyhľadávať.

#### **12.4.4 Testovanie**

Implementácia bola otestovaná v prehliadači a aj na mobilnom zariadení. Neboli nájdené žiadne problémy alebo chyby.

## **12.5 Zistiť spôsob identifikácie verzie databázy a vytvoriť zobrazenie v sekcii ABOUT**

Aplikácia neposkytuje informáciu o verzii databázy rozvrhov tried.

### **12.5.1 Analýza**

Ak na servery sa nachádzajú nové dáta databázy, tak sa stiahnu. Toto sa overuje pomocou verzie databázy.

### **12.5.2 Návrh**

V `app.ts` sa inicializuje `aboutview` cez `conf`, ktorý obsahuje údaje o verzii databázy.

### **12.5.3 Implementácia**

V `aboutView.ts` sa pridá riadok na vypísanie verzie: `template.render(this.config)`, ktorý vypíše verziu.

### **12.5.4 Testovanie**

Na spustenom telefóne som skúsil či verzia je správna aj po update, a bola.

## 12.6 Vytvorenie pomoci pri vyhľadávaní

### VIRTFIIT-3

Ako používateľ chcem vedieť v čom všetkom viem v aplikácii vyhľadávať, lebo teraz to neviem a preto je pre mňa vyhľadávanie nepoužiteľné.

#### 12.6.1 Analýza

Pôvodný stav vyhľadávania bol taký, že nebolo vôbec známe nad akou množinou údajov sa vyhľadáva. Teda používateľ nevedel aké výsledky môže očakávať. Taktiež v prípade žiadnej zhody nevedel, že sa nič nenašlo.

#### 12.6.2 Návrh

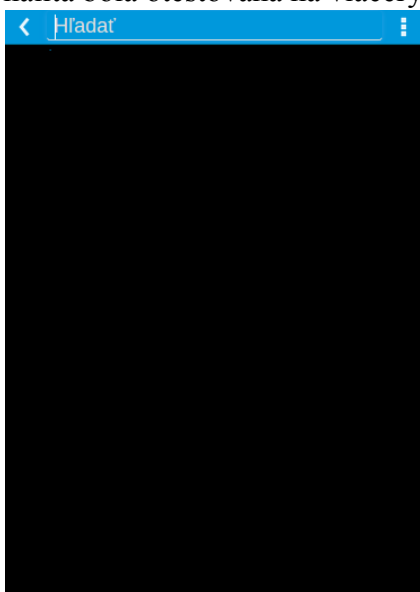
V obrazovke na vyhľadávanie sa pridá informácia o tom nad akou množinou údajov sa dá vyhľadávať. V prípade že sa nič nenájde treba o takomto stave vypísať informáciu do obrazovky s odkazom na študentského radcu.

#### 12.6.3 Implementácia

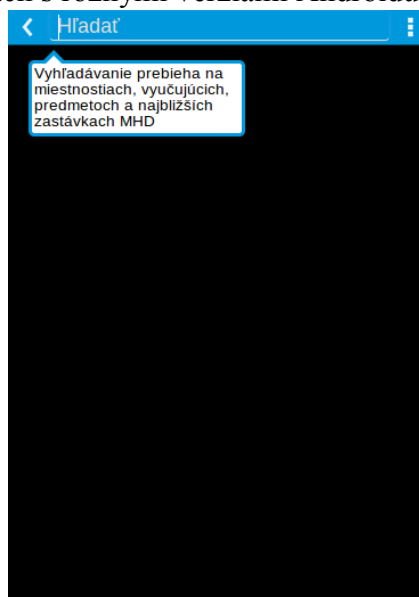
Pri otvorení obrazovky vyhľadávania sa zobrazuje obrázok bubliny v ktorom sa nachádza text o tom, že nad akou množinou údajov sa dá vyhľadávať. Táto bublina bola vytvorená pomocou CSS a teda je ľubovoľne škálovateľná. Ak sa nenájdu žiadne výsledky, ktoré by zodpovedali výsledkom hľadania tak sa zobrazí text, ktorý o tom upovedomí používateľa.

#### 12.6.4 Testovanie

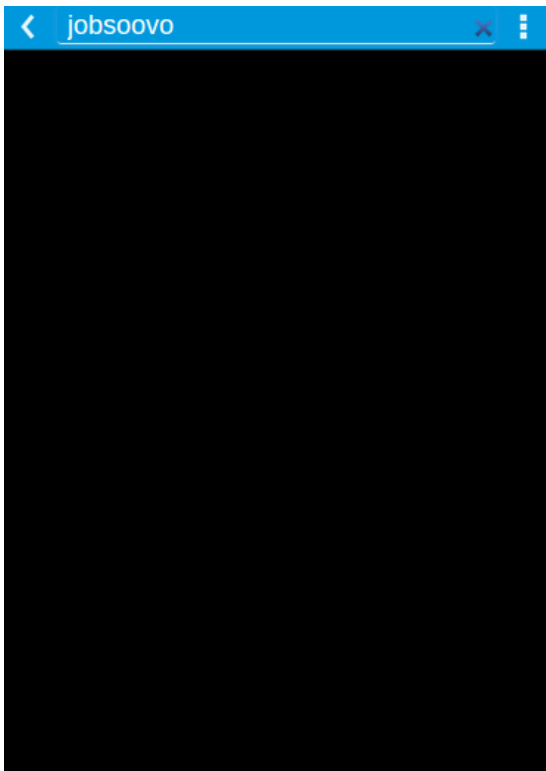
Funkcionalita bola otestovaná na viacerých zariadeniach s rôznymi verziami Androidu.



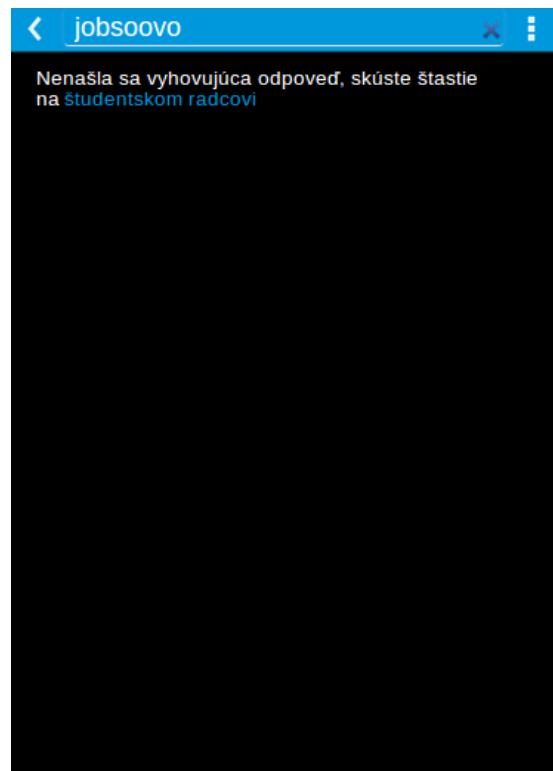
Obrázok 26 Obrazovka vyhľadávania pred



Obrázok 25 Obrazovka vyhľadávania po



Obrázok 28 Výsledky vyhľadávania pred



Obrázok 27 Výsledky vyhľadávania po

## 12.7 Implementácia mapy okolia

### VIRTFIIT-162

Ako používateľ sa chcem zorientovať aj v okolí fakulty, lebo som tu nový a nepoznám to.

#### 12.7.1 Návrh

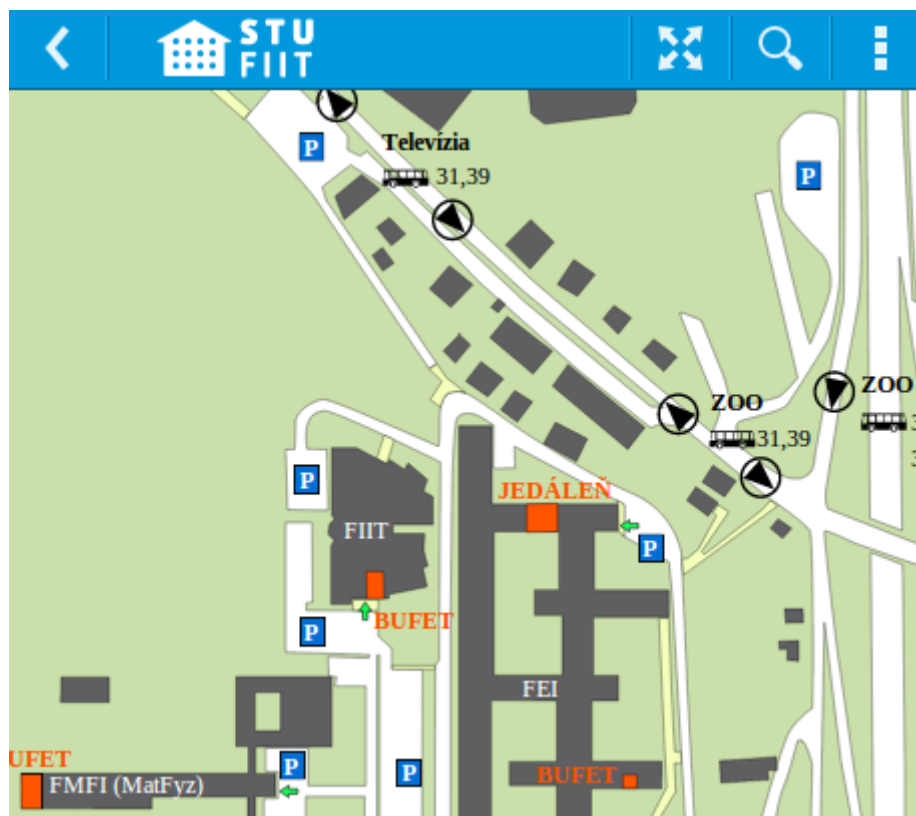
Tím navrhol, aby sa vytvorená mapa okolia z predošlých šprintov implementovala ako ďalšia obrazovka pri ktorej by sa dalo približovať a posúvať rovnako ako na mapách budovy.

#### 12.7.2 Implementácia

Implementácia vychádzala zo skopírovaného kódu z máp budovy, ktorý bol upravený pre potreby mapy okolia a očistený o nepotrebné veci, napríklad navigáciu medzi poschodiami

#### 12.7.3 Testovanie

Mapa bola otestovaná na zariadeniach Android 2.3.x, 4.x a v prehliadači Chrome. Vykazovala pomalé reakcie z dôvodu zložitosti mapy.



Obrázok 29 Mapa okolia

## 13 Ostatné úlohy 6. Šprintu

---

### 13.1 Opravenie chyby zobrazovania Undefined v rozvrhu

VIRTFIIT-172

#### 13.1.1 Analýza

Keď začal nový semester tak sme na základe odozvy od používateľov zistili, že niektorým sa v osobnom rozvrhu zobrazuje ako pomenovanie dňa slovo „undefined“. Neskôr sme zistili, že takýto problém je aj v rozvrhoch niektorých vyučujúcich. Hlbšou analýzou sme dospeli k názoru, že chyba nastala na strane AIS-u. problém bol v tom, že v AIS-e sa zmenila štruktúra akou sú vypisované rozvrhy na jednotlivé dni v týždni. Keďže náš skript, ktorý sťahuje informácie o vyučujúcich, predmetoch, miestnostiach je závislí na pevnej štruktúre AIS-u tak aj malá zmena štruktúry html na strane AIS-u vyvolá pri našom update problém v našej aplikácii. V tomto konkrétnom prípade bol problém v tom, že deň v týždni „Pia“ bol z nejakých príčin zmenený na „Pi“. Čo vyvolalo problém v našom parsery, keďže ten očakával ako vstup „Pia“ a nie „Pi“.

#### 13.1.2 Implementácia

Bol upravený updatový skript, ktorý sťahuje informácie z AIS-u tak aby namiesto „Pia“ parsoval dni v týždni na slovo „Pi“. Po tejto úprave a následnom spustení skriptu bol problém s „undefined“ odstránený.

#### 13.1.3 Testovanie

Testovanie prebehlo tak, že sa spustil manuálne skript, ktorý generuje databázu. Pri pokuse vyhľadať v databáze slovo „undefined“ sme zistili, že chyba bola napravená a teda rozvrh sa bude zobrazovať už korektne.

### 13.2 Analýza možnosti unikátnej identifikácie mobilného zariadenia

VIRTFIIT-158

#### 13.2.1 Analýza

Android má vo svojom API vlastné metódy na získavanie unikátnej identifikácie zariadenia (a dokonca má aj viac rôznych identifikátorov). Phonegap všetky tieto možnosti však nemá, iba jednu. Phonegap vie poskytnúť UUID zariadenia, na ktorom beží. Dostaneme sa k nemu nasledujúcim príkazom:

```
var version = window.device.uuid
```

## **13.3 Zobrazuje sa starý rozvrh**

**VIRTFIIT-153**

**Dátum reportu o chybe:** 18.2.2014

**Sekcia:** ROZVRHY

**Zariadenie, na ktorom sa chyba deteguje:** VŠETKY

**Verzia OS Android:** VŠETKY

Po vyhľadání učebne/profesora a kliknutí na rozvrh, sa zobrazujú staré informácie.

### **13.3.1 Analýza**

Po vyhľadání profesora/učebne sa prejavil problém starého rozvrhu.

### **13.3.2 Návrh**

Po kontrole som zistil, že skript ktorý vytvára automatický update, neprepísal verziu aplikácie.

### **13.3.3 Implementácia**

Prepísal som verziu aplikácie a všetko šlo.

### **13.3.4 Testovanie**

Chyba sa neprejavila už a rozvrhy boli správne.



## 14 Opis príbehov 7. šprintu

---

### 14.1 Možnosť vyhľadávania nad MHD

#### VIRTFIIT-190

Ako používateľ chcem vedieť vyhľadávať nad zastávkami MHD lebo chcem vedieť ktorou linkou z okolia sa viem dostať na dané miesto.

#### 14.1.1 Analýza

Doteraz bolo možné vyhľadávať iba nad miestnosťami, vyučujúcimi a predmetmi.

#### 14.1.2 Návrh

Je potrebné vedieť vyhľadávať aj nad zastávkami MHD liniek, ktoré prechádzajú cez zastávky ZOO a Botanická záhrada. Je potrebné upraviť súbor, ktorý obsahuje zaindexované hodnoty vyhľadávania o špecifické zastávky MHD liniek.

#### 14.1.3 Implementácia

Bol upravený *run.js* skript, ktorý sa spúšťa pri aktualizácii databázy. Keďže mená zastávok MHD sa môžu z času na čas zmeniť je teda potrebné aby sa to nachádzalo v tomto skripte. Boli doňho pridané funkcie, ktoré najprv zabezpečia stiahnutie všetkých liniek spolu so smermi pre každú linku. Následne potom sa vykoná stiahnutie všetkých zastávok a aj identifikátorov liniek bez smerov.

Zo stiahnutých zastávok sa vyfiltrujú iba tie, ktoré nás zaujímajú. V tomto prípade ide o zastávky ZOO a Botanická záhrada. Keď máme takto vyfiltrované zastávky, tak pustíme nad nimi funkciu, ktorá nám vráti zoznamy liniek, ktoré prechádzajú cez dané zastávky v oboch smeroch. Keďže výsledok bude obsahovať duplikáty liniek tak ich ešte prefiltrujeme od duplícít, aby sme získali unikátne identifikátory liniek, ktoré prechádzajú cez nami zvolené zastávky.

Keď už máme vyfiltrované linky od duplícít tak pomocou ďalšej funkcie získame identifikátory jednotlivých zastávok, ktoré daná linka obsluhuje v oboch smeroch.

Po získaní identifikátorov jednotlivých zastávok, ktoré daná linka obsluhuje spustíme parser, ktorý nám vytvorí JSON objekty, ktoré obsahujú nami požadované parametre, ako napríklad identifikátor zastávky, smer jazdy, meno zastávky, meno linky, identifikátor linky, meno zastávky cez ktorú daná linka v našom okolí prechádza (ZOO / Botanická záhrada) a identifikátor rodičovskej zastávky (ZOO / Botanická záhrada).

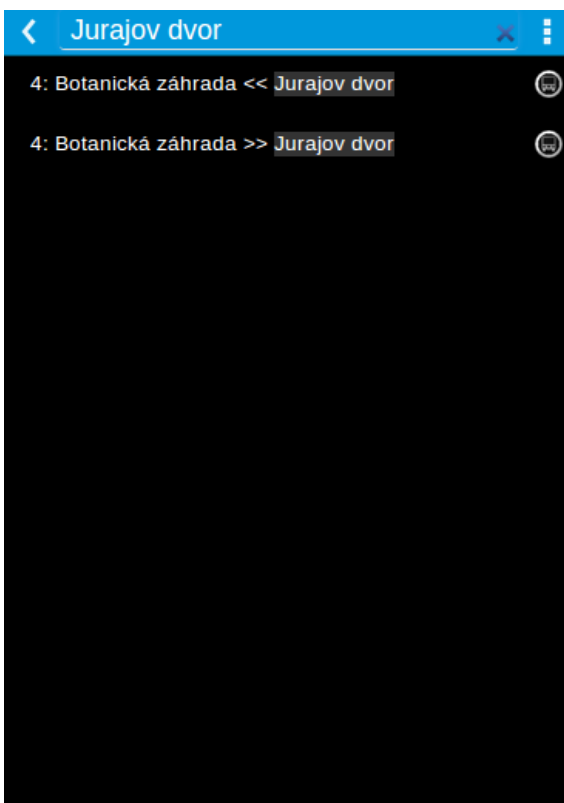
Keďže tieto informácie sú relevantné iba pre účely vyhľadávania v našej aplikácii, tak sa nikde v súborovej štruktúre samostatne nenachádzajú. Tieto informácie sa nachádzajú jedine v *search.json* súbore, ktorý slúži na účely vyhľadávania v našej aplikácii.

#### 14.1.4 Testovanie

Testovanie prebehlo na viacerých zariadeniach s nainštalovaným systémom Android, ale aj vo viacerých prehliadačoch na klasických počítačoch. Neboli spozorované žiadne závažne problémy.



Obrázok 31 Výsledky vyhľadávania MHD pred



Obrázok 30 Výsledky vyhľadávania MHD po

## **14.2 Vytvorenie návrhov rozloženia odkazov**

### **VIRTFIIT-191**

Ako používateľovi mi je pohodlnejšie navigovať odkazmi, ktoré sú vizuálne odlišné, ľahko identifikovateľné a ľahko naučiteľné.

#### **14.2.1 Analýza**

Do teraz boli odkazy zobrazované ako zoznam textových názvov organizácií, ktorým patrili stránky, na ktoré sa odkazy odkazovali.

#### **14.2.2 Návrh**

Zobrazovať odkazy ako zoznam grafických log organizácií, ktorým patria stránky, na ktoré odkazy smerujú pre jednoduchšiu a rýchlejšiu orientáciu v odkazoch.

#### **14.2.3 Implementácia**

Je potrebné vytvoriť grafický návrh zobrazujúci aplikáciu zobrazujúcu odkazy po nahradení textu logami. Je potrebné zaobstarať tieto logá z vhodných zdrojov a pripraviť ich formu pre použitie v aplikácii.

Pre potreby grafického návrhu je potrebné a dostatočné znázorniť len tú časť odkazov zoznamu, ktorá sa zobrazí na obrazovke.

Pre potrebu výskumu použiteľnosti so zámerom „fair use“ sa Logá extrahujú buď z cieľových stránok odkazov alebo z hlavných stránok organizácií prípadne z iných zdrojov zvolených podľa kvality dostupného loga. Pre použitie loga vo vlastnej aplikácii je potrebné si vypýtať povolenie a originálne logo od ich autorov alebo dotknutých organizácií pre súlad s autorskými zákonmi.

Po extrakcii sa logá škálujú na rovnaký rozmer. Škálované logá potom umiestniť a naranžovať na screenshot aplikácie.

#### **14.2.4 Testovanie**

Znázornené návrhy na obrázkoch 1 - 3 slúžia pre výskumné účely pre znázornenie možného budúceho výzoru odkazov.



Obrázok 32 Logá zarovnané na pravo



Obrázok 34 Logá rozšírené na šírku obrazovky v portrait móde



Obrázok 33 Logá zarovnané na stred

## 15 Ostatné úlohy 7. šprintu

---

### 15.1 Analýza sťahovania jedálnych lístkov a MHD

#### VIRTFIIT-193

Ako používateľ chcem aby sa mi jedálne lístky a MHD spoje načítali čo najrýchlejšie. Ak načítavanie trvá dlhšie chcem vedieť koľko ešte budem musieť počkať.

#### 15.1.1 Analýza

Jedálne fungujú na princípe že jedným ajax requestom sa získa jedálny lístok konkrétnej jedálne na určitý deň. Hneď po načítaní jedálneho lístka na dnešný deň sa zobrazí stránka a v pozadí sa sťahujú dáta na ostatné dni. MHD fungujú podobným spôsobom. Pri prvom spustení prebehnú 3 ajax requesty na zistenie statusu, získanie všetkých zastávok a trás, ktoré sa uchovávajú v cache, ale potom keď sa budú načítavať nové údaje prvý deň sa načíta prostredníctvom 2 ajax requestov (Zoo a Botanická) zobrazí sa stránka a ďalšie requesty bežia opäť v pozadí.

#### 15.1.2 Návrh

Je potrebné implementovať pomocné časové premenné na zistenie časov jednotlivých ajax requestov a určiť či je možné pridať do aplikácie “loading bar” element. Resp. či netreba zmeniť časový limit pre ajax requesty (tj. čas, kým sa automaticky nedajú do stavu fail)

#### 15.1.3 Implementácia

Do typescript súboru hladnystudent.ts zodpovedného za načítavanie jedálnych lístkov som do funkcie `private fetchMenu(canteenId, date)` kde sa spúšťa ajax request na načítanie jedálnych lístkov dal časovú premennú na začiatok funkcie a po úspešnom príchode ajax requestu som daný čas zaznačil. Podobným spôsobom som získaval časy pri MHD, kde som časovú premennú pridal do metódy `private ajax(urlParts: string[])`

#### 15.1.4 Testovanie

Pri načítavaní mhd a aj jedálnych lístkov nastavený na jednotlivé ajax requesty časový limit 3s. Testoval som časy requestov na wifi vo web verzii aj na mobile. Testovanie časov prebehlo cca 12krát.

Priemerný čas načítania jedálne: 650ms

Najlepší čas: 150ms

Najhorší čas: 2100ms

Priemerný čas načítania MHD spojov: 820ms

Najlepší čas: 350ms

Najhorší: 2300ms

Testovaním sa zistilo že nemá zmysel pridávať loading bar pre jedálne a aj MHD keďže po načítaní jedálneho lístka resp. spojov na 1. deň sa ostatné načítavajú v pozadí. Takisto podľa časov ajax requestov sa osvedčil časový limit nastavený na 3 sekundy ako vhodná voľba.

## 16 Opis príbehov 8. šprintu

---

### 16.1 Pridanie možnosti nájsť miestnosť aj podľa názvu

#### VIRTFIIT-199

Ako používateľ chcem vedieť nájsť miestnosť podľa názvu, lebo nepoznám jej číselné označenie.

#### 16.1.1 Analýza

Miestnosti v budove sa dali doteraz vyhľadávať len na základe ich číselného označenia čo nebolo veľmi používateľsky pohodlné.

#### 16.1.2 Návrh

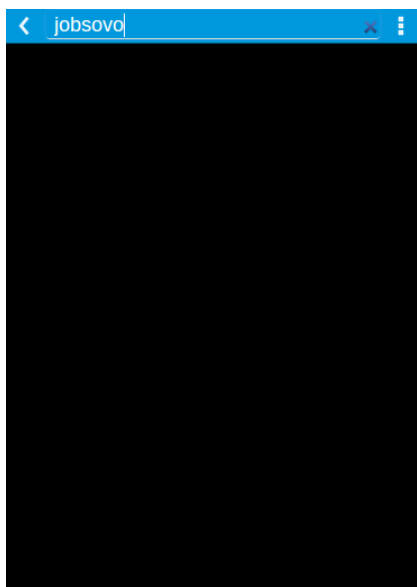
Je potrebné upraviť skript, ktorý zabezpečuje sťahovanie informácií o miestnostiach, tak aby vo výsledkoch vyhľadávania sa dali miestnosti vyhľadať aj podľa ich názvu. Je teda potrebné doplniť v mieste generovania výsledkov do mena miestnosti aj jej názov, ktorý sa nachádza v popise miestnosti.

#### 16.1.3 Implementácia

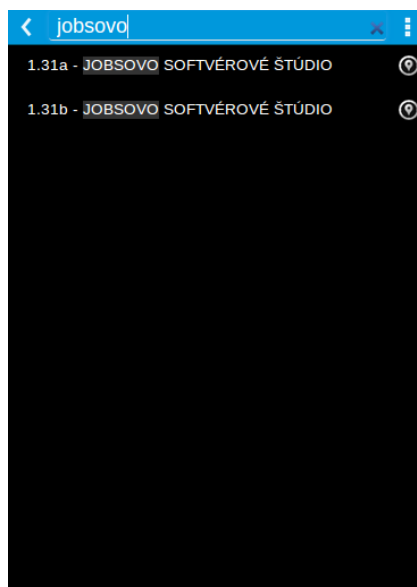
Upravil som skript *run.js*, ktorý sa nachádza v priečinku */scripts/aisdump*. Upravil som funkciu *parseRoom(window, id)*. Zmenil som atribút *name* návratového objektu aby v jeho hodnote bol aj popis miestnosti. Keďže niektoré miestnosti takýto popis nemajú bolo treba túto skutočnosť ošetriť testom na „undefined“.

#### 16.1.4 Testovanie

Testovanie prebehlo na viacerých zariadeniach s nainštalovaným systémom Android, ale aj vo viacerých prehliadačoch na klasických počítačoch. Neboli spozorované žiadne závažné problémy.



Obrázok 36 Výsledok vyhľadávania miestnosti  
nred



Obrázok 35 Výsledok vyhľadávania miestnosti po

## 16.2 Dorobenie klikateľnosti vo vyhľadávaní v MHD

### VIRTFIIT-207

Ako používateľ chcem aby keď vyhľadám zastávku MHD tak aby som na takýto výsledok vyhľadania vedel kliknúť, lebo nechcem sa preklikávať cez obrazovky do obrazovky MHD, kde by som dlho hľadal danú linku a jej odchod z blízkej zastávky.

#### 16.2.1 Analýza

Síce už funguje vyhľadávanie ale nefunguje klikanie na jednotlivé odkazy výsledkov vyhľadávania.

#### 16.2.2 Návrh

Umožniť používateľovi kliknúť na odkaz vo výsledku vyhľadávania aby po kliknutí sa mu zobrazili odchody danej linky na ktorú klikol.

#### 16.2.3 Implementácia

Aby sa dalo klikat' na jednotlivé výsledky vyhľadávania pre MHD bolo potrebné upraviť súbor *results.xjade* v priečinku */mobile/src/views/search/tpl*. Bolo v ňom pridané mapovanie typu „transit“, kde bola nastavená CSS trieda „nav-transit“, ktorá sa odkazuje na ikonku MHD vo výsledkoch hľadania a klikacia linka, ktorá má tvar „/transport/schedule/“. Taktiež musela byť upravená trieda *TransportView* v priečinku */mobile/src/views/transport*, ktorá zabezpečuje aj vykresľovanie obrazovky s odchodmi linky. Konkrétne musela byť upravená premenná „routes“, ktorá ma na starosti mapovanie URL. Fragment URL „Schedule“ pred samotným vykreslením je testovaný na „undefined“ a to z toho dôvodu, že keď používateľ prichádza z obrazovky vyhľadávania tak nie je vygenerovaná na pozadí obrazovka odchodov linky. Z tohto dôvodu sa musí spustiť manuálne funkcia *onShow*, ktorá sa volá automaticky keď je otvorená obrazovka s odchodmi liniek.

#### 16.2.4 Testovanie

Testovanie prebehlo na viacerých zariadeniach s rôznou verziou operačného systému Android.



## **16.3 Klikateľná mapa okolia**

### **VIRTFIIT-213**

Ako používateľ chcem sa chcem dostať na iné miesta v aplikácii cez mapu, ak spolu súvisia, lebo je to nepohodlné vracat' sa späť.

#### **16.3.1 Návrh**

Tím navrhol, aby sa z mapy dalo dostať na jedálne, cestovné poriadky MHD a na mapy budovy FIIT.

#### **16.3.2 Implementácia**

Pre implementáciu sa musela zmeniť mapa a pridať do nej neviditeľné mnohouholníky, ktoré sú stlačiteľné, keďže ak by boli stlačiteľné len elementy, ktoré sú viditeľné, je nízka pravdepodobnosť, že by ich používateľ trafil prstom na mobile na prvý krát.

#### **16.3.3 Testovanie**

Mapa bola otestovaná na zariadeniach Android 2.3.x, 4.x a v prehliadači Chrome.

## **16.4 Automatické načítavanie jedálni raz za deň**

Ak má používateľ na svojom mobilnom zariadení internet, je vhodné aby sa mu automatický stiahli dáta jedálni, ktoré si bude potom môcť prezerať bez toho aby mal internet.

#### **16.4.1 Analýza**

Používatelia ktorý sú pripojení na internet a používajú našu aplikáciu, nie vždy kliknú na možnosť jedálne a tak sa im jedálne lístky neaktualizujú. Ale tlačidlo home je používané veľmi často.

#### **16.4.2 Návrh**

Automatický update sa spusti pri stlačení home tlačidla a zistí sa, či už bol update vykonaný.

#### **16.4.3 Implementácia**

V triede homeView.ts som do metódy kliknutia na home button nastavil kontrolu automatického update-u. Je tam zadaná premenná canteencheck ktorá ma zo začiatku

hodnotu 1, ktorá znamená že ešte neboli update-ované. A tak sa spustí update a táto premenná sa vynuluje.

## **16.5 Vážený graf slov z otázok a odpovedí**

### **VIRTFIIT-206**

#### **16.5.1 Analýza**

Zo získaných záznamov otázok študentov a odpovedí na ne sa má zostaviť vážený graf pre jednotlivé slová na základe ich výskytu a relatívnej vzdialenosti medzi sebou. Pomôže nám to odpovedať na dotazy používateľov aplikácie v prirodzenom jazyku. Cieľom je nájsť relevantné odkazy, na ktoré sa používateľ pravdepodobne pýta.

#### **16.5.2 Návrh**

Vstupný dokument s otázkami a odpoveďami sa musí spracovať a zlematizovať. Výsledné lemy budú tvoriť základ databázy. Každý ďalší dopyt používateľa sa musí rovnako zlematizovať a potom nájsť podobnosť v už vytvorenej databáze. Graf by sa mal vytvoriť zo zlematizovaných slov a ováňovať podľa výskytu slov v tej-ktorej otázke, či odpovedi.

#### **16.5.3 Implementácia**

Graf sa mal pôvodne implementovať v jednom z vyšších programovacích jazykov a zobraziť pomocou nástroja na zobrazovanie grafov. Ale existuje nástroj Elasticsearch, ktorý umožňuje takú funkcionality, ktorú potrebujeme pre náš projekt. Preto sa nateraz od grafu upustilo a skúmal som možnosti Elasticsearch-u. Otázky a odpovede som upravil do formátu JSON a vložil do ES databázy. V nej sa dá vyhľadávať pomocou JSON-like dopytov v konzole (doinštalované rozšírenie). ES podporuje aj „proximity search“, ktorý potrebujeme na zobrazovanie dokumentov (otázok, či odpovedí), ktoré sú najpodobnejšie zadanému vstupu od používateľa.

## 17 Opis príbehov 9. Šprintu

---

### 17.1 Vylepšenie dizajnu MHD a rozvrhov

VIRTFIIT-228

VIRTFIIT-229

VIRTFIIT-230

Ako používateľ chcem aby bol dizajn obrazovky MHD a rozvrhov rovnaký, lebo je potom aplikácia v jednotnej farebnej škále.

#### 17.1.1 Analýza

Pravidelné testovanie s používateľmi ukázalo, že používatelia by si priali aby výpis rozvrhov bol ladený v podobnej farebnej škále ako je výpis na obrazovke MHD.

#### 17.1.2 Návrh

Navrhli sme aby sa modré rámičky, ktoré ohraničovali dni boli šedé a aby boli rovnako široké a zaoblené ako sú tie na obrazovke s MHD. Taktiež sme navrhli aby modrý obdĺžnik pomocou ktorého sa používateľ dostane na obrazovku konkrétnej miestnosti vyzeral ako tlačidlo.

#### 17.1.3 Implementácia

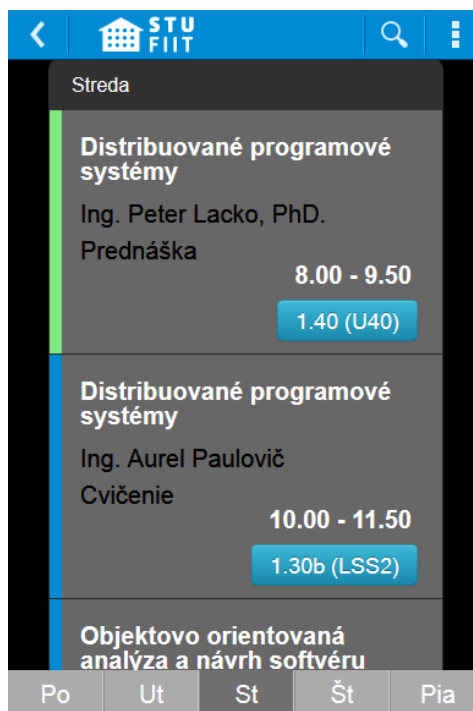
Navrhnuté zmeny sme implementovali pomocou použitia rovnakých CSS tried pre obidve obrazovky. Týmto sme dosiahli to aby dizajn obrazoviek bol totožný a aby prípadne zmeny v budúcnosti sa aplikovali na obidve obrazovky zároveň. Tlačidlo výberu miestnosti bolo zmenené použitím inej CSS triedy ako tomu bolo doteraz.

#### 17.1.4 Testovanie

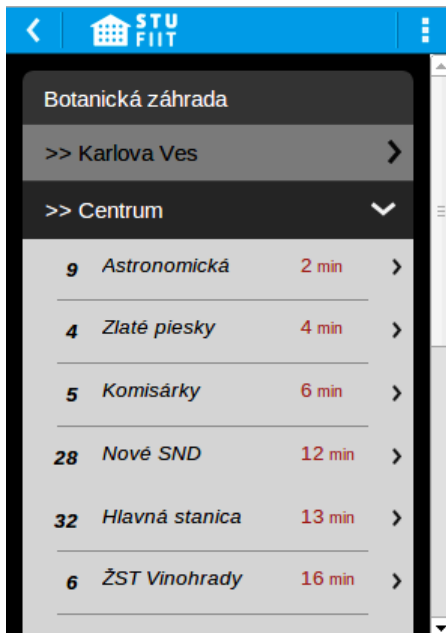
Nový vzhľad obrazoviek sme ukázali vzorke používateľov a boli s ním spokojný. Podľa ich slov nebol vôbec rušivý ale naopak bol prívetivý.



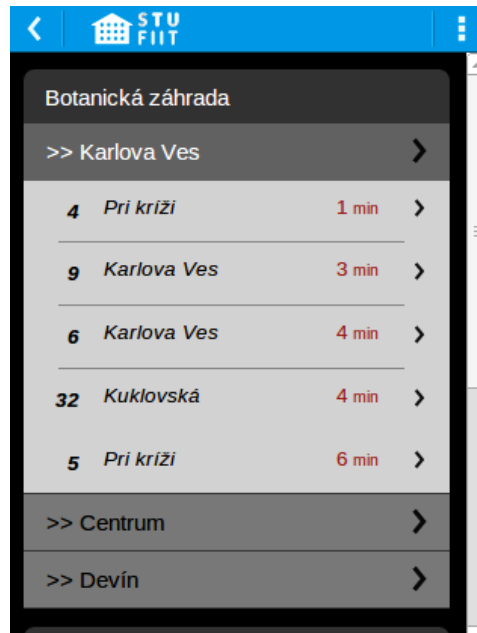
Obrázok 40: Rozvrhy pred úpravou



Obrázok 39: Rozvrhy po úprave



Obrázok 38: MHD pred úpravou



Obrázok 37: MHD po úprave

## 17.2 Nový koncept hlavného menu

### VIRTFIIT-231, VIRTFIIT-232

Ako používateľ chcem, mať prehľadné hlavné menu a rýchly prístup k informáciám.

#### 17.2.1 Analýza

Aplikácia Virtuálna FIIT poskytuje množstvo informácií pre používateľa. Je však dôležité, aby mal rýchly a jednoduchý prístup k týmto informáciám.

#### 17.2.2 Návrh

Návrhom je prerobenie hlavného menu tak aby sa na ňom nachádzali len tie najdôležitejšie informácie a ostatné sa pridajú do bočného vysúvacieho menu. Na hlavnej obrazovke zostanú odkazy pre osobný rozvrh študenta, vyhľadávane informácií, mhd spoje, jedálne lístky, mapa okolia a mapa budovy FIIT. Do bočného menu sa dajú odkazy na študijné oddelenie, harmonogram štúdia, rss správy z fakulty, odfotenie QR kódov, odkazy a informácie o aplikácii.

#### 17.2.3 Riešenie

Riešenie sa realizovalo odstránením zbytočných a pridaním potrebných odkazov do hlavného menu v súbore „home-view.xjade“. Bočné menu sa implementovalo v súboroch „index.xjade“, „app.ts“ a štýl sa nachádza v súbore „sidemnu.less“. Bočné menu sa dá vysunúť pomocou tlačidla v ľavom hornom rohu na domovskej stránke aplikácie alebo pomocou potiahnutia prsta na pravú stranu. Prst musí n obrazovke prejsť minimálnu vzdialenosť 120px na osi x. Táto hodnota sa dá zmeniť a je uložená v premennej „DISTANCE“, ktorá sa nachádza v súbore „app.ts“. Pre vzsúvanie menu bola tiež pridaná animácia pomocou jquery funkcie „animate“. Dĺžka animácie je 300ms a táto hodnota sa nachádza v premennej „ANIMATION\_TIME“ v súbore „app.ts“.

#### 17.2.4 Testovanie

Všetky odkazy sú umiestnené na hlavnej obrazovke tak ako bolo navrhnuté. Bočné menu je funkčné a obsahuje tie odkazy, ktoré má mať. Pri natívnej android aplikácii sa zistilo, že sa občas vyskytuje menšie zasekávanie pri animácii vysúvanie bočného menu. Tento problém sa však nevyskytuje pri verzii vo webovom prehliadači a predpokladáme, že problém je s technológiou „grunt“, ktorá kompiluje android verziu aplikácie.

## 17.3 Analýza použitia knižnice scrollspy

### VIRTFIIT-233

Ako používateľ chcem aby sa pri posúvaní menu na obrazovke rozvrhov automaticky označoval aktuálne zobrazený deň, lebo chcem byť vizuálne informovaný na ktorom dni sa nachádzam.

#### 17.3.1 Analýza

Analyzoval som možnosť použiť knižnicu Bootstrap resp. jeden jej plugin, ktorý by vedel zabezpečiť scrollspy. Použitie tohto pluginu sa ukázala ako zbytočná záťaž pre načítavanie zdrojov. Preto sme sa v konečnom dôsledku rozhodli nepoužiť tento plugin, ale vyrobiť si vlastnú odľahčenú verziu scrollspy.

#### 17.3.2 Návrh

Návrh samotného scrollspy pozostáva v tom, že sa nastavuje aktívny prvok pre element dňa (Pondelok, Utorok, ..., Piatok) keď „vdchádza“ do okna.

#### 17.3.3 Implementácia

Implementácia samotného scrollspy sa nachádza v metóde *renderTimetable* v súbore *timetableView.ts* v priečinku */src/views/timetable*.

## 17.4 Implementácia kompasu do mapy

### VIRTFIIT-246

Ako používateľ chcem vedieť môj smer cesty keď sa pohybujem, lebo teraz netuším či chodím po budove alebo okolí fakulty správnym smerom.

#### 17.4.1 Analýza

Analýza spočívala v odskúšaní pluginu cordova-plugin-device-orientation, či je vlastne možné a či sme schopní zistiť vo phonegap smerovanie mobilu podľa svetových strán. Po vyriešení problémoch s technológiou TypeScript sa mi to podarilo vyriešiť a mohol som prejsť k návrhu kompasu.

#### 17.4.2 Návrh

Návrh vychádzal z toho, že chceme naraz používať mapu aj kompas, čiže musia byť naraz vidieť na tej istej obrazovke. Vybral sa teda pre kompas ľavý horný roh.



Obrázok 41 Návrh grafiky kompasu

#### 17.4.3 Implementácia

Implementácia bola podľa návrhu bola dokončená aj pre mapu budovy aj pre mapu okolia. Stupne otočenia kompasu boli v budove zmenené o 90 stupňov, aby ukazovali správny smer. Kompas sa obnovuje každých 300 milisekúnd.



**Obrázok 42 Ukážka kompasu v mape budovy, kompas znázorňuje že človek pred vrátnicou ukazuje mobilom na študijné oddelenie**

#### **17.4.4 Testovanie**

Testovanie bolo vykonávané na zariadeniach s kompasom aj bez kompasu, na ktorých sa kompas skryl a mohli používať mapu tak, ako doteraz. Testovanie dopadlo úspešne a kompas aj keď oneskorene do par sekúnd vždy ukazoval smer, akým je zariadenie otočené k mape.



## 17.5 Optimalizácia mapy okolia

### VIRTFIIT-247

Ako používateľ chcem aby sa mapa rýchlejšie približovala a posúvala na mobile, lebo v súčasnom stave je to nedostatočne rýchle.

#### 17.5.1 Analýza

Z predošlých úloh optimalizácií SVG máp som zistil, že je potrebné:

- Odstrániť alebo nahradiť elementy, ktoré majú veľa bodov
- Odstrániť násobné nastavenie štýlov
- Vymazanie zbytočných bodov v elementoch

#### 17.5.2 Návrh

Navrhol som všetky odporúčané body z analýzy aplikovať na hotovú mapu, malé ikony zmeniť na rastre. Elementy, ktoré mali rovnakú farbu som navrhol zoskupiť a nastaviť im rovnaký štýl naraz na všetky.

#### 17.5.3 Implementácia

Výsledkom implementovania týchto zmien bolo zrýchlenie vyzobrazovania mapy a zmenšenie jej veľkosti na disku o približne 30%.

## 17.6 Analýza kompilácie aplikácie pre Windows Phone

### VIRTFIIT-252

Ako budúci používateľ aplikácie chcem aby išla na mojom Windows Phone zariadení, lebo ju chcem využívať.

#### 17.6.1 Analýza

Po otestovaní aplikácie na zariadení Windows Phone podľa návodu “Build a spustenie aplikácie pre Windows Phone 8“ zo sekcie návody, sa zistilo že aplikácia je na platforme Windows Phone 8 nefunkčná. Jedným z dôvodom je použitie knižnice Zepto.js, ktorá v aktuálnej verzii nie je kompatibilná s Windows Phone platformou.

#### 17.6.2 Návrh

Zepto.js knižnica bude nahradená odľahčenou verziou jQuery

### 17.6.3 Implementácia

Do aplikácie bol pridaný plugin jQuery a odstránený plugin Zepto.js, následne sa všade v kóde zmenili referencie zo Zepto na jQuery.

## 17.7 Zmena spúšťania automatických aktualizácií

### VIRTFIIT-253

Ako používateľ chcem, aby sa automatické aktualizácie robili bez toho, aby som musel niekam kliknúť. Chcem aby to všetko prebiehalo na pozadí raz za deň.

#### 17.7.1 Analýza

Automatické aktualizácie síce fungujú, ale zapínajú sa iba po kliknutí na home button.

#### 17.7.2 Návrh

Pre aktualizácie je vhodné aby sa spúšťali samé od seba, bez toho aby nejaká akcia od používateľa musela byť vytvorená. Toto možno dosiahnuť tak, že vždy keď sa človek vráti do aplikácie, a je dostupná aktualizácia, tak sa stiahne. Samozrejme musí mať aj internet. Čiže celý tento proces bude odštartovaný po obnovení aplikácie.

#### 17.7.3 Implementácia

V aplikácii, v triede app.ts sa pri spustení aplikácie zavolá metóda checkUpdates(), ktorá overí všetky dostupné aktualizácie. Toto zahŕňa (Cestovné poriadky, jedálne a novú verziu aplikácie). Takto prebehne prvá aktualizácia. Ak používateľ minimalizuje aplikáciu a potom ju obnoví, tak sa spustí znova aktualizácia vďaka pridanému listenerovi :

```
document.addEventListener("resume", ()=> this.checkUpdates(), false).
```

Samotná aktualizácia kontroluje, či už dnes bola aktualizácia vykonaná. Ak nebola tak sa neaktualizuje no ak nebola tak sa spustí proces aktualizácie. Dátum poslednej aktualizácie je uložený v local storage-i. K tomuto dátumu sa možno dostať pomocou príkazu:

```
localStorage.getItem('app.last-update')
```

Aktualizácia sa nachádza v app.ts a názov tejto metódy je checkUpdates().

#### 17.7.4 Testovanie

Testovanie prebehlo tak, že sa nastavila aktualizácia na každé 3 minúty. Po tom ako sa obnovila aplikácia, aktualizácia prebehla. Ak bola aplikácia v pozadí hodinu, tak po spustení sa spustila iba jeden krát, čo je správne.

## **17.8 Analýza proximity search v rámci Elasticsearch**

### **VIRTFIIT-254**

Ako používateľ chcem byť schopný vyhľadávať nad čo najväčšou bázou dát, lebo potrebujem rýchly prístup k informáciám.

#### **17.8.1 Analýza**

Používatelia musia byť schopní vyhľadávať v našej aplikácii tak, aby mali rýchly prístup do všetkých sekcií v aplikácii a taktiež k dátam, ktorými aplikácia (a náš vzdialený server) disponuje.

#### **17.8.2 Návrh**

Je potrebné upraviť terajšie vyhľadávanie tak, aby nebolo iba kontextové v rámci aplikácie (teda hľadanie medzi číslami miestností, osôb na FIIT, a tak ďalej), ale aj nad bázou prídavných dát, ktoré môžu mať výpovednú hodnotu pre používateľa.

#### **17.8.3 Implementácia**

Kontextové vyhľadávanie v rámci aplikácie už hotové je. Full-textové vyhľadávanie nad bázou dát robíme prostredníctvom open-source projektu Elasticsearch (ES), ktorý kombinuje no-SQL databázu s full-textovým vyhľadávaním. Najskôr bolo potrebné vytvoriť uzly a indexy v ES a potom zaindexovať dokumenty, ktoré budú prístupné v rámci vyhľadávania. Posledný bod je vytvoriť komunikáciu klient-server, ktorý spraví dopyt do ES a navráti dokument s najväčšou zhodou. Dopyt, na ktorý odpovedá ES musí byť správne štruktúrovaný.

#### 17.8.4 Testovanie

Po skúšaní rôznych dopytov pre proximity search v rámci ES som dospel k záveru, že pre naše potreby (nájsť zhodu čo najviac slov, v lepšom prípade aj v správnom poradí) je najlepší dopyt:

```
{
  "query": {
    "bool": {
      "must": {
        "match": {
          "title": "OTÁZKA"
        }
      },
      "should": {
        "match": {
          "title.shingles": " OTÁZKA "
        }
      }
    }
  }
}
```

Testované to bolo na viacerých dopytoch do ES databázy (prístup cez Sense rozhranie Marvel pluginu vo webovom prehliadači) a práve vyššie uvedený sa ukázal ako najúčinnjší.

## 18 Ostatné úlohy 9. Šprintu

---

### 18.1 Chýbajúce zastávky MHD

VIRTFIIT-237

**Dátum reportu o chybe:** 01.04.2014

**Sekcia:** MHD

**Zariadenie, na ktorom sa chyba deteguje:** VŠETKY

**Verzia OS Android:** VŠETKY

Keď som sa pokúšal vyhľadať autobus č. 92 zo zastávky ZOO tak takýto autobus z danej zastávky nebol nájdený.

#### 18.1.1 Analýza

Analýzou problému som zistil, že problém bol v tom, že v prvotnej implementácii algoritmu na generovanie dát som použil pre identifikovanie jednotlivých záznamov existujúce ID (`stop_id`). To by nebol až taký problém, ale pri indexovaní záznamom pre lepšie vyhľadávanie sa indexujú záznamy podľa ID. To znamená, že síce záznamy sa predtým vygenerovali správne ale tým, že záznamy mali rovnaké ID sa väčšia časť záznamov stratila.

#### 18.1.2 Implementácia

Upravil som skript `run.js`, ktorý generuje zastávky pre našu aplikáciu takým spôsobom, že som si vytvoril unikátne IDčko (`myIndependentUniqueID`), ktoré som priraďoval jednotlivým záznamom.

#### 18.1.3 Testovanie

Testovanie prebehlo na viacerých zariadeniach a daná chyba sa už neprejavuje.

## 19 Opis príbehov 10. Šprintu

---

### 19.1 Zmena veľkosti ikon v hlavnom menu podľa rozlíšenia displeja

#### VIRTFIIT-289

Ako používateľ chcem aby sa ikonky na hlavnej obrazovke primerane škálovali, lebo mi vadí, že na veľkom displeji vidím malé ikonky.

#### 19.1.1 Analýza

Analýzou sme zistili, že na veľkých displejoch malé ikonky v hlavnom menu nevyzerajú vôbec dobre. Je preto potrebné ich primerane škálovať.

#### 19.1.2 Návrh

Je potrebné ikonky v hlavnom menu primerane škálovať podľa displeja.

#### 19.1.3 Implementácia

Škálovanie ikon je zabezpečené pomocou media query, ktorá deteguje rôzne rozlíšenia zariadení a podľa toho nastavuje škálovací koeficient. CSS kód je možné nájsť v súbore *home-style.less*, ktorý sa nachádza v priečinku *src/views/home*.

#### 19.1.4 Testovanie

Rozlíšenie ikon sa testovalo na viacerých zariadeniach s rôznym rozlíšením.

## 20 Ostatné úlohy 10. Šprintu

---

### 20.1 Nefunkčné vyhľadávanie z dôvodu nenačítania MHD

VIRTFIIT-286

#### 20.1.1 Analýza

Ak používateľ nemá internetové pripojenie nefunguje možnosť vyhľadávania. Zobrazí sa hláška o nemožnosti načítania MHD a vráti používateľa do hlavného menu.

#### 20.1.2 Návrh

Ak používateľ nemá k dispozícii internetové pripojenie. Vyhľadávanie by malo ďalej normálne fungovať. Ak používateľ vyhľadá autobusový spoj, chybovú hlášku o nemožnosti zobrazenia danej linky z dôvodu neexistujúceho internetového pripojenia zobrazí až po kliknutí na daný výsledok hľadania.

#### 20.1.3 Implementácia

Upravil som súbory transportView.ts a searchView.ts. V súbore searchView.ts som odstránil zobrazenie chybovej hlášky a vrátenie obrazovky do hlavného menu ako aj zobrazenie chybovej hlášky pri nemožnosti načítaní údajov z iných dôvodov (server neodpovedá). Tým pádom nemožnosť všetky hlášky načítania MHD spravuje súbor transportView.ts, kde som pridal ešte kontrolnú podmienku, či je linka, ktorú chceme cez vyhľadávanie zobraziť načítaná.

#### 20.1.4 Testovanie

Vykonanú zmenu som testoval na mobilnom zariadení Nokia Lumia U a aj na webovej verzii.

Tabuľka 7 Vyhľadávanie bez internetu

| Číslo kroku | Opis kroku  | Očakávaná akcia   | Výsledok |
|-------------|---|---|----------|
| 1.          | Používateľ vypne internetové pripojenie na svojom zariadení                 | Zariadenie nemá prístup k internetu   | OK       |
| 2.          | Používateľ zapne aplikáciu Virtuálna FIIT a klikne na tlačidlo Vyhľadávanie | Používateľovi sa nezobrazí žiadna chybová hláška a dostane sa na obrazovku vyhľadávania | OK       |
| 3.          | Používateľ dá vyhľadať autobusový spoj alebo zastávku                       | Používateľovi sa zobrazí chybová hláška o nemožnosti zobrazenia                         | OK       |

|  |  |   |  |
|--|--|---|--|
|  | (napr. botanická) a klikne na nájdený výsledok z MHD | danej linky a presmeruje ho na hlavnú obrazovku aplikácie |  |
|--|--|---|--|

## 20.2 Chyba scrollspy v rozvrhu

### VIRTFIIT-295

#### 20.2.1 Analýza

Po kliknutí na deň na ktorý sa nedá nascrollovať sa automaticky označí deň ktorý je možné ako posledný nascrollovať.

#### 20.2.2 Návrh

Pokiaľ nie je možné nascrollovať sa na konkrétny deň malo by sa pohodlne dať naňho aj tak kliknúť aby v navigácií bol daný deň označený.

#### 20.2.3 Implementácia

Túto chybu som opravil tak, že som implementoval vo funkcii scrollspyu zámok, ktorý sa zapína a vypína podľa toho aký pohyb používateľ vykonáva na obrazovke. Či ide o scrollovanie alebo kliknutie na konkrétny deň.

#### 20.2.4 Testovanie

Opravené scrollspy sa testovalo na viacerých zariadeniach a monitoroch.

## 20.3 Notifikácia emailom o odoslaní logu

### VIRTFIIT-174

#### 20.3.1 Analýza

Odosielanie logov na server je úspešné. Avšak, ak sa nahlási chyba, je potrebné o nej čo najrýchlejšie vedieť. Preto treba nainštalovať a nakonfigurovať SMTP server, ktorý bude odosielať informáciu o novom logu prostredníctvom e-mailovej notifikácie.

#### 20.3.2 Návrh

Zvolený linuxový smtp server je postfix. Nato, aby tento server išiel, treba ho nakonfigurovať aby odosielať správy cez školský server mail.fiit.stuba.sk. Pre jazyk php existujú rôzne knižnice na odosielanie mailov. Implicitný príkaz mail, nie je postačujúci z dôvodu že sa nekontroluje či bol mail naozaj odoslaný.



### **20.3.3 Implementácia**

Po nainštalovaní serveru ho treba nakonfigurovať. Bolo treba upraviť súbor `/etc/postfix/main.cf`, v ktorom sa doplnil riadok `“relayhost = mail.fiit.stuba.sk”`, aby sa odosieli e-mailové správy cez školský mailový server a `“inet_protocols = ipv4”` aby sa používala klasická verzia ip adres. Aby bolo odosielanie ošetrené, používa sa knižnica PHPMailer pre php. Toto odosielanie mailov je v skripte `send.php`, ktorý slúži na uloženie logov na server. Takže počas ukladania logov sa odošle notifikačný mail.

### **20.3.4 Testovanie**

Odosielanie mailov bolo testované najskôr iba pri spustení skriptu manuálne na rôzne mailove adresy za použitia rôznych odosielateľov. Toto išlo správne, a tak sa otestovalo automatické odosielanie pri vložení logu. Škúsalo sa odoslanie viacerých logov, a na všetky prišiel notifikačný e-mail.

## 21 Opis príbehov 11. Šprintu

---

### 21.1 Komunikácia klient server pri odosielaní dopytu inteligentného vyhľadávania

#### VIRTFIIT-306

Ako používateľ chcem aby inteligentné vyhľadávanie odosielať moju žiadosť na server, kde sa potom spracuje, vyhodnotí a odošle naspäť vo forme odpovede.

#### 21.1.1 Analýza

Komunikácia bude prebiehať medzi: klientom a serverom, serverom a službou lematizér a následne server a klient pre odoslanie odpovede.

#### 21.1.2 Návrh

Aby toto vyhľadávanie išlo správne aj vo webovej verzii, treba upraviť konfiguráciu serveru NGINX a povoliť komunikáciu. Medzi serverom a lematizérom treba odomknúť porty cez, cez ktoré budú komunikovať.

#### 21.1.3 Implementácia

V konfiguračnom súbore `/etc/nginx/sites-enabled/virtfiit` sa pridal odstavec aby bolo možné komunikovať so serverom. Odomknutie portu sa spravilo v iptables príkazom:

„`sudo iptables -A INPUT -p tcp --dport 8267 -j ACCEPT`“ a „`sudo iptables -A INPUT -p tcp -dport 8200 -j ACCEPT`“, keďže na komunikáciu je potrebné mať otvorené dva porty. Medzi aplikáciou a serverom bol vytvorený ajax get na náš server, na stránku [www.team06-13.ucebne.fiit.stuba.sk/find.php](http://www.team06-13.ucebne.fiit.stuba.sk/find.php), kde sa spracoval odoslaný reťazec. Po spracovaní sa vykonal príkaz `echo` s výsledkom, ktorý sa na klientskej strane zobrazí na zariadení.

#### 21.1.4 Testovanie

Odsielalo sa mnoho dopytov pričom všetky boli úspešne. Dopyty boli sledované pomocou nástroja wireshark.

## 21.2 Klientická časť inteligentného vyhľadávania

### VIRTFIIT-284

Ako používateľ a študent FIIT chcem aby vyhľadávanie v aplikácii umožňovalo nájsť odpovede na všetky moje otázky ohľadom štúdia, pretože potrebujem tieto odpovede nájsť veľmi rýchlo a jednoducho.

#### 21.2.1 Analýza

Používateľ bude schopný zadať otázku (dopyt v prirodzenom jazyku) a na základe full-textového vyhľadávania sa tento dopyt vyhledá v databáze otázok a odpovedí od študentských radcov na FIIT.

#### 21.2.2 Návrh

Po zadaní dopytu do vyhľadávacieho okna aplikácie sa najskôr hľadá daný výraz použitím indexového vyhľadávania v lokálnej databáze aplikácie. Prehľadajú sa učitelia, miestnosti, autobusové zastávky, ak sa nič nenájde dopyt sa odošle na server s databázou otázok a odpovedí od študentských radcov.

#### 21.2.3 Implementácia

V priečinku mobile/src/lib/virtfiit som vytvoril nový súbor search.ts s triedou ElasticSearch na spracovanie poslania requestu na server a prijatia odpovede so servera. V súbore app.ts som pridal inicializáciu triedy ElasticSearch. V súbore searchView.ts som pridal volanie metódy ElasticSearch.ajax(msg), ktorá pošle request na server s inteligentným vyhľadávaním, kde parameter msg predstavuje daný používateľský dopyt. Následne som upravil výpis výsledkov v príslušnom .xjade súbore.

#### 21.2.4 Testovanie

Vyhľadávanie som testoval na mobilnom zariadení Nokia Lumia U a aj na webovej verzii.

Tabuľka 8 Inteligentné vyhľadávanie – klientická časť

| Číslo kroku | Opis kroku                               | Očakávaná akcia                   | Výsledok |
|-------------|--|-----------------------------------|----------|
| 1.          | Používateľ klikne na ikonku vyhľadávania | Zobrazí sa vyhľadávacia obrazovka | OK       |

|    |  |   |    |
|----|--|---|----|
| 2. | Používateľ zadá dopyt, ktorý nemá súvis s vyučujúcimi, miestnosťami alebo mhd zastávkami, napr. slovo „štipendium“ | Používateľovi sa zobrazí nájdená otázka a odpoveď obsahujúca slovo „štipendium“ alebo slovo veľmi podobné | OK |
|----|--|---|----|

## **22 Ostatné úlohy 11. Šprintu**

---

### **22.1 Chýbajúce zastávky MHD - Hl. stanica**

**VIRTFIIT-303**

**Dátum reportu o chybe:** 9.5.2014

**Sekcia:** MHD

**Zariadenie, na ktorom sa chyba deteguje:** VŠETKY

**Verzia OS Android:** VŠETKY

Ked' som klikol na vyhľadávanie a chcel som vyhľadať Hlavná stanica, nič mi to nenašlo

#### **22.1.1 Analýza**

Podrobnejším skúmaním som zistil, že chyba nastala pri poslednom spájaní vetiev v GITE, keď sa pôvodná implementácia stratila.

#### **22.1.2 Implementácia**

Opravil som to tak, že som v GITE vyhľadal poslednú funkčnú verziu a doplnil som kód do aktuálnej verzie, aby to fungovalo.

### **22.2 Veľké rozlíšenie ikon pri malom rozlíšení displeja v landscape mode**

**VIRTFIIT-296**

**Dátum reportu o chybe:** 29.4.2014

**Sekcia:** HOME

**Zariadenie, na ktorom sa chyba deteguje:** VŠETKY

**Verzia OS Android:** VŠETKY

Ked' som otvoril aplikáciu, tak ikonky na hlavnej obrazovke boli veľké

### **22.2.1 Analýza**

Analýzou sa zistilo, že CSS selektor dobre nezachytil rozlíšenie obrazovky.

### **22.2.2 Implementácia**

Upravil som hraničné hodnoty pre CSS media selektor, ktorý zabezpečuje výber správne naškálovaných ikoniek.

### **22.2.3 Testovanie**

Otestoval som rôzne rozlíšenia ikon v Google Chrome prehliadači, pomocou pluginu, ktorý zväčšoval a zmenšoval okno.

## **22.3 Na mape sa nedá približovať mapu pomocou myšky**

**VIRTFIIT-XXX**

**Dátum reportu o chybe:** 15.4.2014

**Sekcia:** MAPY

**Zariadenie, na ktorom sa chyba deteguje:** Webové prehliadače

**Verzia OS Android:** VŠETKY

Vo webových prehliadačoch sa nedalo približovať mapu pomocou scroll kolieska myšky.

### **22.3.1 Analýza**

Chyba bola vyskúšaná na viacerých prehliadačoch a na každom bol rovnaký výsledok. Zistilo sa, že chyba bola kvôli zmene knižnice „zepto.js“ na „jquery.js“.

### **22.3.2 Návrh**

Návrhom na opravu bolo prepísanie časti kódu, tak aby fungoval aj pod novou knižnicou „jquery.js“.

### **22.3.3 Implementácia**

Opravilo sa to tak, že sa zmenil spôsob ako sa čítali eventy pre scroll s myšou. To sa dosiahlo tak, že sa pridalo rozhranie „MouseEvent“ a časti kódu s eventmi „e“ sa zmenili na „e.originalEvent“, tak ako to vyžaduje knižnica „jquery.js“.

#### **22.3.4 Testovanie**

Riešenie bolo otestované na viacerých webových prehliadačoch a viacerých zariadeniach. Všetko fungovalo tak ako malo a chyba sa už nevyskytovala.

## 23 Opis príbehov 12. Šprintu

---

### 23.1 Rozvrh skúškového

#### VIRTFIIT-234

Ako používateľ chcem mať prístup ku rozvrhu skúškového v aplikácií, lebo chcem mať počas skúškového obdobia prehľad o svojich skúškach.

#### 23.1.1 Analýza

Na to aby sme mohli do aplikácie umiestniť rozvrh skúškového je potrebné získať od rozvrhára rozvrh skúšok v excel formáte. Taktiež je potrebné aby boli správne nastavené dátumy začiatku a konca semestra v *src/config.json*. Po získaní rozvrhu v excel formáte je potrebné z neho vygenerovať súbor *finals.json*. Viac informácií ako to urobiť je napísaných v Implementácií.

#### 23.1.2 Návrh

V aplikácií sa bude nachádzať rozvrh skúšok pod ikonkou rozvrhy. Počas študijnej časti semestra sa v tejto sekcii po prihlásení nachádza rozvrh konkrétnej prihlásenej osoby. Keď sa skončí študijná časť semestra a začne skúšková, obsah tejto sekcie sa vymení za rozvrh skúšok.

#### 23.1.3 Implementácia

##### NUTNÉ POŽIADAKY

Po získaní rozvrhu skúšok od rozvrhára je potrebné spustiť skript *run.js*, ktorý sa nachádza v *scripts/finals*. Informácia o tom ako sa tento skript a s akými parametrami sa spúšťa je bližšie popísané v samotnom súbore na začiatku. Po vytvorení súboru *finals.json* je potrebné ho nakopírovať do priečinku */data* v aplikácií.

##### REALIZÁCIA

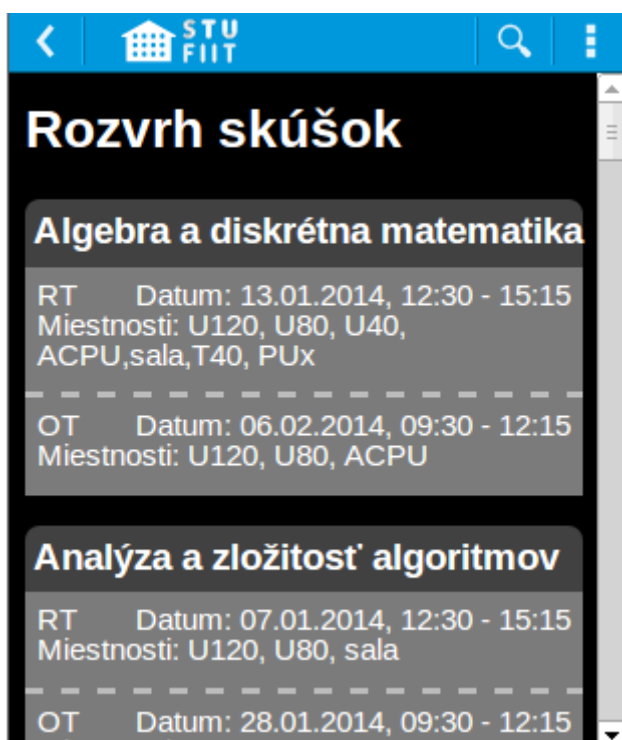
Po prihlásení sa do AISu sa overí či ešte je / nieje semester. To či je alebo nie je semester sa ukladá do localstoragu do premennej *is\_semester*. V kóde sa to overuje v *timetableView.ts*, v inicializácií. Buď sa volá metóda *showTimetable()* počas semestra alebo počas skúškového je to metóda *showFinals()*. Metóda *showFinals()* získa informácie z AISu o predmetoch, ktoré ma daný človek zapísané a vyfiltruje ich z celkového zoznamu skúškových predmetov. Taktiež je možné nepoužiť tento filter a ako parameter pre *renderFinals* dať všetky predmety (*finals.json*). metóda *renderFinals* ma za úlohu teda buďto



naformátovať všetky predmety alebo zavolať filtračnú metódu `this.ais.filterMyFinals` ktorá vyfiltruje predmety konkrétneho človeka a vráti ich už naformátované v správnom tvare. Predmety je potrebné naformátovať z toho dôvodu aby výstupné dáta boli lepšie čitateľné. Formátovanie predmetov prebieha v metóde `formatFinals`, ktorá sa nachádza v `src/lib/virtfiit/ais.ts`. Táto metóda doplní do informácií o predmetoch názov predmetu a pomocou metódy `formatTerms` v rovnakom súbore upraví dátum, začiatok a koniec skúšky a názvy miestností aby boli lepšie čitateľné.

### 23.1.4 Testovanie

Funkcionalitu som otestoval v prehliadači ale aj na tablete Prestigio. Rozvrh skúšok sa zobrazoval korektne.



Obrázok 43 Rozvrh skúškového

## 23.2 Serverová časť inteligentného vyhľadávania

### VIRTFIIT-333

Ako používateľ chcem byť schopný vyhľadávať nad čo najväčšou bázou dát, lebo potrebujem rýchly prístup k informáciám.

### **23.2.1 Analýza**

Elasticsearch (ES) bol úspešne vyskúšaný a otestovaný v lokálnej verzii s testovacími vstupmi. Treba toto riešenie preniesť na server a prepojiť s klientskou aplikáciou cez Internet tak, že definujeme vstupný bod pre PHP súbor bežiaci na serveri.

### **23.2.2 Návrh**

Na serveri vytvorím PHP súbor, ktorý sa postará o prijatie a spracovanie vstupnej otázky od používateľa z klientskej aplikácie. PHP následne začne komunikovať s ES a na základe hľadania nájde najpodobnejší dokument, ktorý navráti klientskej aplikácii na zobrazenie.

### **23.2.3 Implementácia**

Vytvorené „find.php“ prijme premennú typu string od klientskej aplikácie. Tomuto vstupu bude treba najskôr pridať diakritiku, ak chýba. Na to je potrebný slovník s množstvom slov, ktoré bude treba porovnať a ak sa slová zhodujú, doplní do pôvodného slova diakritiku tak, ako je na slove zo slovníka. Táto časť však na serveri ešte naimplementovaná nie je.

Následne sa tento upravený vstup zlematizuje poslaním do lematizéru na adrese [text.fiit.stuba.sk/lemmatizer](http://text.fiit.stuba.sk/lemmatizer). Výstupom je text, v ktorom sú slová premenené na (domnelé) základné tvary. Nie vždy to však funguje správne. Tento text je rozbitý na jednotlivé slová, ktoré sa ďalej spracúvajú.

Načíta sa zoznam stop-slov a ak sa nájde slovo z tohto zoznamu v zlematizovanom texte, tak sa odstráni. Zostatkové pole slov sa znova spojí do jedného stringu.

Po všetkých týchto úpravách je text pripravený ako dopyt do ES. Hľadá sa zhoda v zlematizovaných dokumentoch v databáze. Po nájdení vhodných dokumentov sa zoberie ID dokumentu s najväčšou zhodou a z ES databázy sa prinesú originálne texty práve na základe tohto ID. Otázka a odpoveď sa dajú dohromady do jedného stringu a odošlú sa ako odpoveď klientskej aplikácii.

### **23.2.4 Testovanie**

Správnu funkcionálnu som testoval priamo na PHP súbore v prehliadači (bez klientskej aplikácie), ktorému som podsunul text, ktorý by dostal z klientskej aplikácie. Všetko prebiehalo v poriadku, no v niektorých prípadoch sa stalo, že ES nerozpoznal špeciálny znak zo slovenskej abecedy. Veľakrát sa tiež stávalo, že lematizér nesprávne určil slovo a teda mu priradil zlý základný tvar.

## 24 Ostatné úlohy 12. Šprintu

---

### 24.1 Nesprávne rozlíšenie ikoniek v portrait mode pri 768x1024

VIRTFIIT-322

**Dátum reportu o chybe:** 13.5.2014

**Sekcia:** HOME

**Zariadenie, na ktorom sa chyba deteguje:** VŠETKY

**Verzia OS Android:** VŠETKY

Keď som otvoril aplikáciu, tak ikonky na hlavnej obrazovke boli malé.

#### 24.1.1 Analýza

Analýzou sa zistilo, že CSS selektor dobre nezachytil rozlíšenie obrazovky.

#### 24.1.2 Implementácia

Upravil som škálovací koeficient na 1.7 v src/views/home/home-style.less

#### 24.1.3 Testovanie

Rozlíšenie som otestoval v Google Chrome prehliadači, pomocou pluginu, ktorý zväčšoval a zmenšoval okno.

## 25 Návody

---

### 25.1 Inštalácia a spustenie aplikácie pre vývoj

Tento návod popisuje inštaláciu a konfiguráciu všetkých potrebných častí za účelom vytvorenia vhodného prostredia pre vývoj aplikácie Virtuálna FIIT. Znak \$ na začiatku riadkov značí použitý príkaz v termináli.

#### 25.1.1 Inštaláciu Git-u

Najskôr treba git nainštalovať

```
$ sudo apt-get install git
```

Potom je potrebné našu aplikáciu, ktorá sa nachádza na stránke Bitbucket vyexportovať

```
$ git clone https://skyrow@bitbucket.org/dorny/virtfiit.git
```

Tento náš server pridáme medzi známe servery nášho lokálneho gitu

```
$ git add remote server git@bitbucket.org:dorny/virtfiit.git
```

#### 25.1.2 Inštalácia nodeJS

NodeJS nainštalujeme bežným spôsobom

```
$ sudo apt-get install nodejs
```

Ak by však nastali nejaké problémy s nodejs, resp. nezhody vo verziách treba ho nainštalovať nasledovným spôsobom:

```
$ sudo apt-get update
```

```
$ sudo apt-get install python-software-properties python g++ make
```

```
$ sudo add-apt-repository ppa:chris-lea/node.js
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install nodejs
```

#### 25.1.3 Inštalácia a konfigurácia PhoneGap a Android SDK

PhoneGap nainštalujeme príkazom

```
$ sudo npm install -g phonegap
```

Treba stiahnuť android sdk zo stránky <http://developer.android.com/sdk/index.html> (stačí sdk tools only) a rozbaľiť do zvoleného priečinka. Následne treba nainštalovať jdk a ant.

```
$ sudo apt-get install openjdk-7-jre
```

```
$ sudo apt-get install openjdk-7-jdk
```

```
$ sudo apt-get install ant
```

Je prospešné nastaviť cestu v premennej PATH aby systém vedel prístupit' k android nástrojom odkiaľkoľvek. Cesta, ktorá je hrubým značí umiestnenie rozbaleného android sdk.

```
export PATH=$PATH:$HOME/android-sdk-linux/tools:$HOME/android-sdk-  
linux/platform-tools
```

Ďalej treba nainštalovať android emulátor

```
$ sudo apt-get install android-tools-adb
```

Pokiaľ máte 64-bitovú verziu Linuxu tak je potrebné nainštalovať aj nasledujúce balíky a ich závislosti:

```
$ sudo apt-get install lib32stdc++6 ia32-libs
```

Ak pri inštalácii týchto balíkov vypíše chybu, treba skontrolovať v update manageri Linuxu, či nechce nainštalovať nové aktualizácie, po ich nainštalovaní by mala inštalácia balíkov prebehnúť v poriadku.

Pozrite si zoznam nainštalovaných verzii Android SDK

```
$ android list targets
```

Emulátor zariadenia sa vytvorí nasledovným príkazom, kde „t“ značí vybrané API zo zoznamu vyššie a „n“ názov zariadenia

```
$ android create avd -n myAndroid -t android-18
```

Emulátor spustíme nasledovným spôsobom

```
$ emulator -avd myAndroid
```

#### 25.1.4 Build aplikácie

Prejdite do priečinku virtfiit (klonovaný z Bitbucket) a do priečinku mobile a spustite tieto príkazy:

```
$ sudo npm install -g grunt-cli
```

```
$ sudo npm install .
```

Prejdite do priečinku tools/xjade a spustite

```
$ sudo npm install .
```

Prejdite naspäť do priečinku mobile a nasledujúci príkaz, kde „t“ značí použité Android API:

```
$ android update project -p cesta_k_vasmu_sdk/virtfuit/mobile/platforms/android -t 'android-18'
```

Build aplikácie vykonáme príkazom

```
$ grunt
```

Pre inštaláciu aplikácie na mobilné zariadenie alebo spustený emulátor treba spustiť príkaz

```
$ grunt run
```

## 25.2 Inštalácia virtuálneho serveru

Za virtuálny server sme si zvolili OS ubuntu 12.04 Server pretože OS ubuntu má prevažná časť z nás. Po nainštalovaní som na virtuálny server dal našu webovú stránku a nainštaloval weiner – nástroj na debugovanie.

### 25.2.1 Inštalácia serveru

Cez VNC viewer po zadaní našich prihlasovacích údajov som sa pripojil na náš virtuálny server, kde už bol inštalovaný proces. Počas inštalácie som nastavil hlavne tieto časti: pri výbere miesta kam sa má systém nainštalovať: guided - user entire disk a pri type servera som zvolil možnosť LAMP. Po skončení tohto procesu sa mi nepodarilo OS rozchodiť a zopakoval som rovnaký postup trikrát a už to šlo. Ďalší problém bol po spustení OS ten, že po krátkom čase mi v terminálovom okne vypísalo dlhú kernelovú chybu a nedokázal som nič spraviť, tak som napísal žiadosť do redmine-u, aby to Ing. Peter Lacko reštartoval a všetko už šlo v pohode. Aby som nemusel používať stále VNC prístup, ktorý bol dosť nepohodlný, pomalý a často krát sa stalo že ma vypojoilo, som sa rozhodol nainštalovať ssh sever: \$sudo apt-get install openssh-server; \$sudo apt-get install openssh-client. Po tomto som už mohol začať používať server.

### 25.2.2 Vloženie stránky na server

Na to aby šla stránka, som skontroloval, či je nainštalovaný apache a cez scp protokol (port 22, keďže port na ftp je zakázaný) som preniesol dokumenty našej stránky do priečinku /var/www/. Po tomto už stránka išla.

### 25.2.3 Inštalácia debug nástroja „Weinre“ na server

Na to aby som mohol nainštalovať weinre, som musel najskôr nainštalovať node.js cez sled príkazov:

```
$sudo apt-get install python-software-properties python g++ make
```

```
$sudo add-apt-repository ppa:chris-lea/node.js
```

```
$sudo apt-get update
```

```
$sudo apt-get install nodejs
```

```
$sudo apt-get install npm
```

a potom som mohol nainštalovať weinre cez príkaz:

```
$sudo npm -g install weinre
```

Po tejto inštalácii som spustil príkaz `$weinre --boundHost=-all-` a príkaz `$screen`, v ktorom som spustil weinre (príkaz `weinre`) a cez „Ctrl+d“ som sa vrátil späť zo screen-u, pričom weinre stále bol spustený v pozadí.

#### 25.2.4 Testovanie

Vstup: Do webového prehliadača vložíme link:

<http://labss2.fiit.stuba.sk/TeamProject/2013/team06is-si/>

Výstup: Zobrazí sa naša webová stránka.

Vstup: Do webového prehliadača vložíme link:

<http://team06-13.ucebne.fiit.stuba.sk:8080/>

Výstup: Zobrazí sa webová stránka Weinre-u

### 25.3 Nastavenie automatického update-u databázy

Automatický update som robil pre server `stavba.fiit.stuba.sk`, kde bolo treba vždy manuálne prekopírovať súbory `courses.json`, `people.json`, `rooms.json` a `timetable.json`. Nebolo to príjemne a trvalo to vždy dosť času pri čakani kým sa stiahnu dáta a kým sa človek pripojí na server a vloží dáta. Zautomatizovanie som sa rozhodol spraviť prostredníctvom linuxového skriptu písaného v BASH-i.

#### 25.3.1 Skript pre stiahnutie novej verzie a prekopírovanie súborov

Skript ktorý stiahne údaje z aisu sa nachádza v `scripts/aisdump/run.js`. Aby bolo možné spustiť treba nainštalovať všetky závislosti v zložke `scripts/aisdump` pomocou príkazu `$npm install` a spustenie samotného príkazu cez `$node run.js`.

V skripte sa pred začatím sťahovania súborov prekopírujú všetky súbory zo zložky `data/` (kam sa stiahnu nové údaje) do zložky `data_old/`. Po tomto sa stiahnu nové údaje a príkaz `diff -q` zistí zmeny medzi všetkými súbormi s rovnakými názvami a vypíše len tie, ktoré sú rozdielne. Ak sa nájde súbor ktorý by/sú bol rozdielny/e tak sa spustí ďalší skript pre pripojenie na stavbu, a odoslanie súborov cez sftp protokol na server – odoslanie.

Prekopírovanie súborov je z lokálnej zložky `scripts/data/` do vzdialenej zložky na serveri `www/mobile/data/`.

### 25.3.2 Zautomatizovanie skriptu

Aby skript mohol byť spustený, spravil som si lokálnu kópiu našej aplikácie kde sa nachádza aj skript na sťahovanie dát na náš virtuálny server. Toto riešenie je dočasné a snaha je prekopírovať tento skript priamo na server stavba.fiit.stuba.sk, kde sa rieši prístup. No zatiaľ to je na našom virtuálnom serveri. Aby bolo možné zautomatizovať spúšťanie skriptu a odosielanie dát, bolo nutné nainštalovať crontab cez príkaz: `$sudo apt-get install cron`. Po nainštalovaní cron-u som mohol naplánovať spúšťanie skriptu každý pondelok o 00:00 pomocou príkazu: `$sudo crontab -e` (keď som tento program spustil prvý krát, tak som si vybral druhú možnosť pre zobrazovanie dokumentu – gedit). V novom textovom okne som pridal riadok:

```
00 00 * * 1 /home/vifiit/virtfiit/scripts/aisdump/chod
```

Čo znamená že skript sa spustí o polnoci (00:00 ľubovoľný dátum dňa - \*, ľubovoľný dátum mesiaca – 8, a druhý deň v anglickom kalendári (pondelok, číslovanie začína od 0 – nedeľa).

Po spustení skriptu sa do `/home/vifiit/virtfiit/scripts/aisdump/automatic_update/cas.txt` uloží čas kedy bol spustený skript a napíše či sa dáta odosielali aj na server.

### 25.3.3 Testovanie

Vstup: V utorok (tj. Po pondelku 00:00) si pozrieme dokument `cas.txt`;

Výstup: Bude sa v ňom nachádzať informácia o stiahnutí novej verzie dát



## 25.4 Manuálne spustenie aktualizácie databázy

V niektorých prípadoch je potrebné si manuálne vygenerovať údaje z databázy pre vlastné testovacie účely. Na to aby sme si vedeli takéto dáta sami vygenerovať je potrebné spustiť manuálne skript, ktorý sa o všetko postará.

Skript ktorý generuje celú DB sa nachádza v priečinku *virtfiit/scripts/aisdump*. Názov skriptu je *run.js*. Aby sme vedeli daný skript spustiť je potrebné vo vyššie uvedenej zložke spustiť príkaz *npm install*, ktorý stiahne a nainštaluje potrebné závislosti do počítača. Pred spustením skriptu je potrebné si vytvoriť vlastnú kópiu súboru *auth.sample.json*, ktorý pomenujeme *auth.json*. Doňho napíšeme naše prihlasovacie meno a heslo do AIS-u. Tieto údaje sa nikde neuchovávajú, sú použité výlučne na komunikáciu s AISom.

Samotný skript sa spúšťa cez konzolu príkazom *node run.js*. Trvá dlhšiu dobu kým celý skript zbehne, preto sa netreba báť, keď sa zdá, že konzola je „mŕtva“. Keď skript skončí tak vypíše informáciu o tom koľko predmetov, vyučujúcich a rozvrhov bolo stiahnutých.

Keď kvôli testovaniu nechceme vždy sťahovať všetky dáta tak si otvoríme súbor *run.js* v editore a v poli *conf* zmeníme premennú *true* na *false* pri tých premenných, ktoré nás momentálne nezaujímajú. V štandardnom nastavení sú všetky premenné (*courses*, *people*, *rooms*, *timetable*, *routes* nastavené na *true* a *vcard* na *false*.

*Courses* sa starajú o stiahnutie dát o predmetoch, *people* sťahuje informácie o ľuďoch na škole, *rooms* sa stará o miestnosti, *timetable* o rozvrhy, *routes* sa stará o stiahnutie zastávok MHD v okolí a *vcard* slúži na stiahnutie informácií o vyučujúcich vo formáte *vcard*.

### 25.4.1 Generovanie QR kódov pre vyučujúcich

Keď chceme vygenerovať QR kódy pre vyučujúcich je potrebné najprv otvoriť súbor *run.js* a premennú *vcard* v poli *conf* nastaviť na *true*. Ostatné premenné okrem *people* môžeme nastaviť pre rýchlosť na *false*, keďže nie sú pre nás potrebné.

Spustíme skript štandardným spôsobom cez *node run.js* a keď skript skončí tak sa nám v priečinku *virtfiit/scripts/aisdump* vytvorí priečinok *vcard*, ktorý obsahuje informácie o vyučujúcich vo formáte *vcard* z ktorého sa budú generovať QR kódy.

Aby sme vedeli generovať samotné QR kódy je potrebné si nainštalovať balík *qrencode*. Nainštalujeme ho pomocou príkazu *npm install qr*. Keď sa balík nainštaloval tak zostáva spustiť už iba skript *qr.js* pomocou príkazu *node qr.js*, ktorý vytvorí ďalší priečinok s názvom *qr*, ktorý bude obsahovať vygenerované QR kódy.

## 25.5 Spustenie a build aplikácie pre Windows Phone 8

**Prerekvizity:**

- Operačný systém Windows 8
- Vytvorený Microsoft účet (<https://login.live.com>) a Microsoft Windows Phone Developer účet (<https://dev.windowsphone.com/en-us> )
- Stiahnuté a nainštalované Windows Phone 8 SDK (<https://dev.windowsphone.com/en-us/downloadsdk>)
- Zaregistrované mobilné zariadenie s WP8 na vývoj

### Ako zaregistrovať mobilné zariadenie pre vývoj

1. Zapnite telefón a odomknite obrazovku
2. Uistite sa že čas a dátum na telefóne sú správne
3. Pripojte telefón prostredníctvom USB k počítaču
4. Na počítači spustíte program Windows Phone Developer Registration (program sa nainštaluje pri inštalácii WP8 SDK)
5. Kliknite na Register (ak sa bude vyžadovať meno a heslo, použite váš Microsoft účet)

### GIT, NodeJS

- Naištalujte si NodeJS (<http://nodejs.org/>)
- Naištalujte si GIT pre Windows (<http://msysgit.github.io/>)
  - Pri inštalácii vyberte recommended možnosti

### GIT Clone, NPM dependencies

- Spustíte cmd (command prompt), prejdite do priečinka kde chcete mať našu aplikáciu
  - Medzi priečinkami sa presúva klasicky `cd cesta`, obsah priečinka zobrazíte príkazom `dir` a medzi jednotlivými diskovými zväzkami napr. ak sa chceme dostať do zväzku G príkazom `G:`
- `git clone https://vasemeno@bitbucket.org/dorny/virtfiit.git`
- po naklonovaní virtfiit priečinka choďte do mobile a spustíte príkaz ***npm install*** .
- choďte do priečinka `/tools/xjade` a aj tu spustíte príkaz ***npm install*** .

### Windows Phone SDK Path

1. Kliknite na tento počítač a dajte vlastnosti
2. Vyberte pokročilé systémové nastavenia
3. Pod záložkou pokročilé kliknite na systémové premenné
4. Medzi systémovými premennými nájdite premennú **Path** a upravte ju tak že na koniec pridáte **C:\Windows\Microsoft.NET\Framework\v4.0.30319;**

### Windows Phone Local Build

- Choďte do priečinka `/virtfiit/mobile` a spustíte príkaz ***phonegap local build wp8***
- Zbuildované .xap takisto ako aj Visual Studio projekt (.sln) sa budú nachádzať v `/mobile/platforms/wp8`

- Buildnutú appku je možné do zariadenia nahrať buď prostredníctvom Application Deployment tool (Windows Phone SDK) alebo nahraním projektu do Visual Studio a následným spustením na zariadení.
- Spustenie na emulátore je možné cez Visual Studio ale je nutné mať Windows 8 Pro verziu alebo vyššie

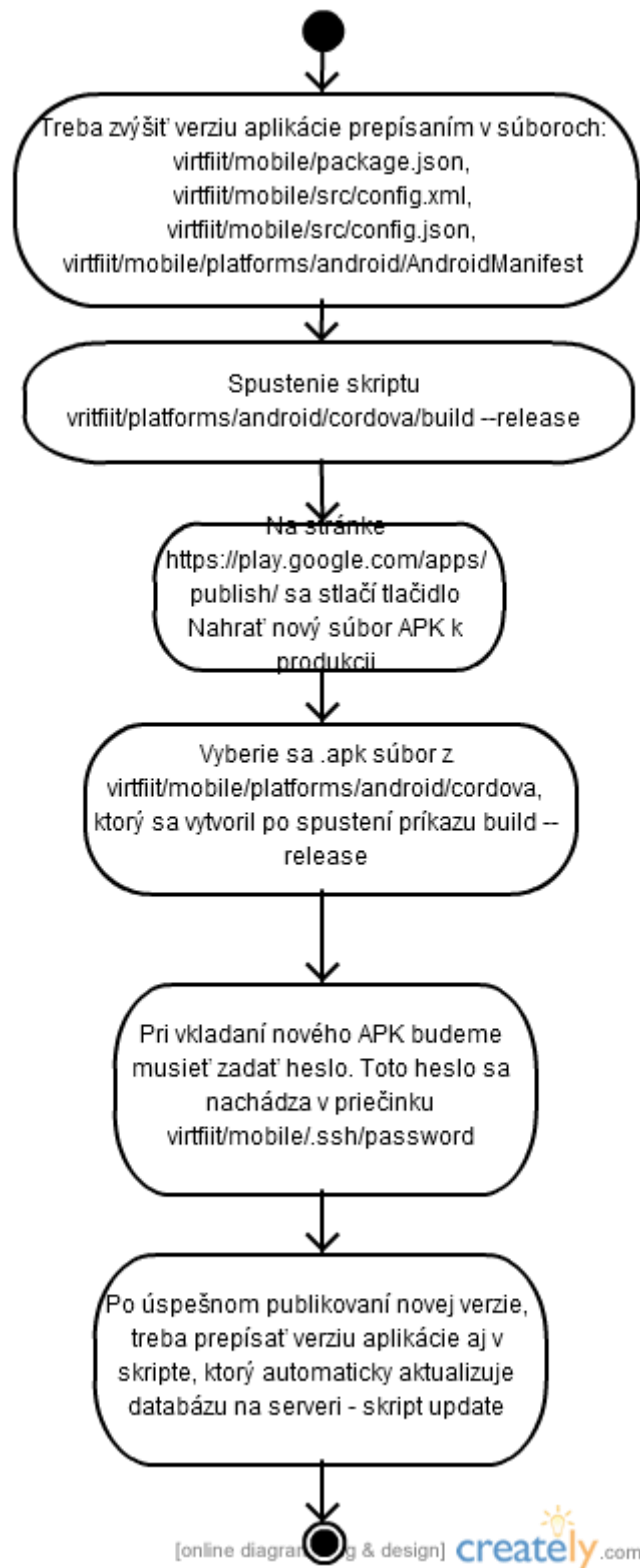
## **25.6 Nasadenie aplikácie**

### **25.6.1 Nasadenie na Google Play**

Aby sme mohli našu aplikáciu vložiť do Google Play, musíme v nej upraviť verzie aplikácie na vyššiu hodnotu. Po tom ako tieto verzie zvýšime, vytvoríme prostredníctvom príkazu build – release v príkazovom riadku podpísaný .apk súbor. Tento súbor môžeme vložiť na Google Play, pričom počas vkladania budeme musieť zadať heslo. Nová verzia aplikácie bude dostupná pre verejnosť za zopár hodín, čo treba brať do úvahy.

### **25.6.2 Nasadenie webovej verzie**

Pre webovú verziu je nasadzovanie jednoduchšie, stačí prekopírovať skompilované súbory z adresára mobile/www do adresára na servery, na ktorom je spustená produkčná verzia aplikácie. Na našom tímovom servery je to adresa var/virtfiit/www



Obrázok 44 Deployment diagram nasadenia aplikácie do Google Play

## 26 Celkový pohľad

---

### 26.1 Opis produktu

#### 26.1.1 Hlavná obrazovka

Na hlavnej obrazovke boli zmenené všetky ikonky, ktoré zobrazujú používateľovi príbuzný obsah. Z rôznorodej pestrofarebnosti boli zmenené do jednotného farebného odtieňu aby ladili s výzorom celej aplikácie. Výsledkom je šesť nových ikon, ktoré navrhol člen nášho tímu Jaroslav Dzurilla. On je aj hlavným grafikom v našom tíme a väčšina grafických prvkov, ktoré prinášame sú práve od neho. Ikonky, ktoré používame v aplikácii sme použili aj na tímovom plagáte. Na hlavnej obrazovke pribudla aj nová ikonka pre „home“ tlačidlo, ktorá má tvar typický pre návrat na základnú obrazovku. Pre túto ikonku bol použitý domček, na aký sú používatelia bežne zvyknutý z prehliadačov. Zároveň však bola zachovaná symbolika aplikácie a jej prepojenie so školou vo forme „bodiek“ vnútri domčeka a názvu STU FIIT vedľa neho.

#### 26.1.2 Mapy

Medzi prvé úlohy, ktoré sme riešili bolo úprava máp jednotlivých poschodí budovy FIIT. Na niektorých poschodiach sa podľa informácií zmenili číslovania miestností, pridali, alebo odstránili niektoré priečky v miestnostiach. Tieto zmeny bolo potrebné preniesť aj do aplikácie, lebo v opačnom prípade by mapy neaktuálne a pôsobili by nedôveryhodne na používateľa.

Jednotlivé poschodia sme si v tíme rozdelili a upravili podľa aktuálnych podkladov, ktoré sme mali k dispozícii, taktiež sme na jednotlivých mapách zarovnávali dvere do stien, lebo väčšina z nich sa nachádzala mimo nich. Tento posun vznikol pri transformácii máp z formátu PNG do formátu SVG, ktorý používame. Taktiež sme na jednotlivých poschodiach odstraňovali nadbytočné body v mapách, ktoré vznikli takouto transformáciou. Miestami na definovanie jednej steny boli namiesto štyroch bodov použité osem až viac bodov, čo spôsobovalo pomalé načítanie jednotlivých poschodí.

Taktiež sme do aplikácie pridali mapu širšieho okolia FIITky, ktorá zobrazuje priestor od internátov až po Botanickú záhradu so všetkými dôležitými objektmi. Na tejto mape je možné nájsť okolité budovy fakúlt, ktoré sa nachádzajú v okolí FIITky, zástavky MHD, stravovacie zariadenia. Internáty, ako aj označenie smerov do jednotlivých mestských častí.

Špeciálnou funkciou pre navigáciu v budove a jej okolí je kompas, ktorý pomáha v zorientovaní sa v budove. Každá mapa obsahuje v ľavom hornom rohu šípku, akým smerom sa práve pozeráte na objekty zobrazené na mape. Kompas nezobrazuje v každom prípade sever, lebo mapy nemáme otočené na sever z dôvodu lepšej prehľadnosti.

### 26.1.3 Rozvrhy

Rozvrhy sa od svojej pôvodnej verzie zmenili len málo. Zmenil sa spôsob akým používatelia dokážu s ním pracovať. V aktuálnej verzii rozvrhov už nie je potrebné posúvať sa ručne aby som našiel konkrétny deň v rozvrhu, ale boli implementované prepínače na jednotlivé dni v týždni. Používateľ teda môže ľahko a pohodlne prepínať medzi jednotlivými dňami v týždni. Pôvodná funkcionálna však zostala zachovaná a používa sa aj naďalej.

### 26.1.4 Odkazy

Medzi odkazy boli pridané nové prepojenia na YonBan, IMHD, a RSS správy. Ku odkazom, ktorým chýbal popis tak bol pridaný. Naopak, ku odkazom pri ktorých chýbala skratka tak bola doplnená. Popisy odkazov prešli gramatickou korekciou nakoľko niektoré z nich obsahovali závažné gramatické chyby.

RSS Správy dokážu zobrazovať aktuálne informácie o dianí na fakulte. Tieto RSS správy sa preberajú z fiit.stuba.sk.

### 26.1.5 MHD

MHD prešlo oproti pôvodnej verzii výraznou zmenou. Jednotlivé odchody MHD sa nezobrazujú tak ako tomu bolo pôvodne a to v chaotickom výpise kde formát výpisu vyzeral približne takto: „xxxxx smer Pri križi“. Takto formovaný výpis bol nahradený výpisom jednotlivých liniek rozdelených do dvoch základných skupín. Jednu skupinu tvoria linky, ktoré odchádzajú zo zastávky Botanická záhrada a druhú skupinu tvoria linky, ktoré odchádzajú zo zastávky Zoo. Slovo „smer“, ktoré pôsobilo vizuálne rušivo bolo nahradené znakmi „>>“, ktoré majú rovnaký význam ako slovo „smer“. Bola zmenená aj vizuálna podoba celého výpisu. Názvy skupín boli osadené do šedých rámečkov, na názvy smerov bola použitá modrá farba, na výpis minút nasledujúceho odchodu bola použitá červená farba. Na názvy konkrétnych liniek bola použitá kurzíva. Linky autobusov sú usporiadané v týchto zoznamoch podľa času najbližšieho odchodu niektorých z liniek.

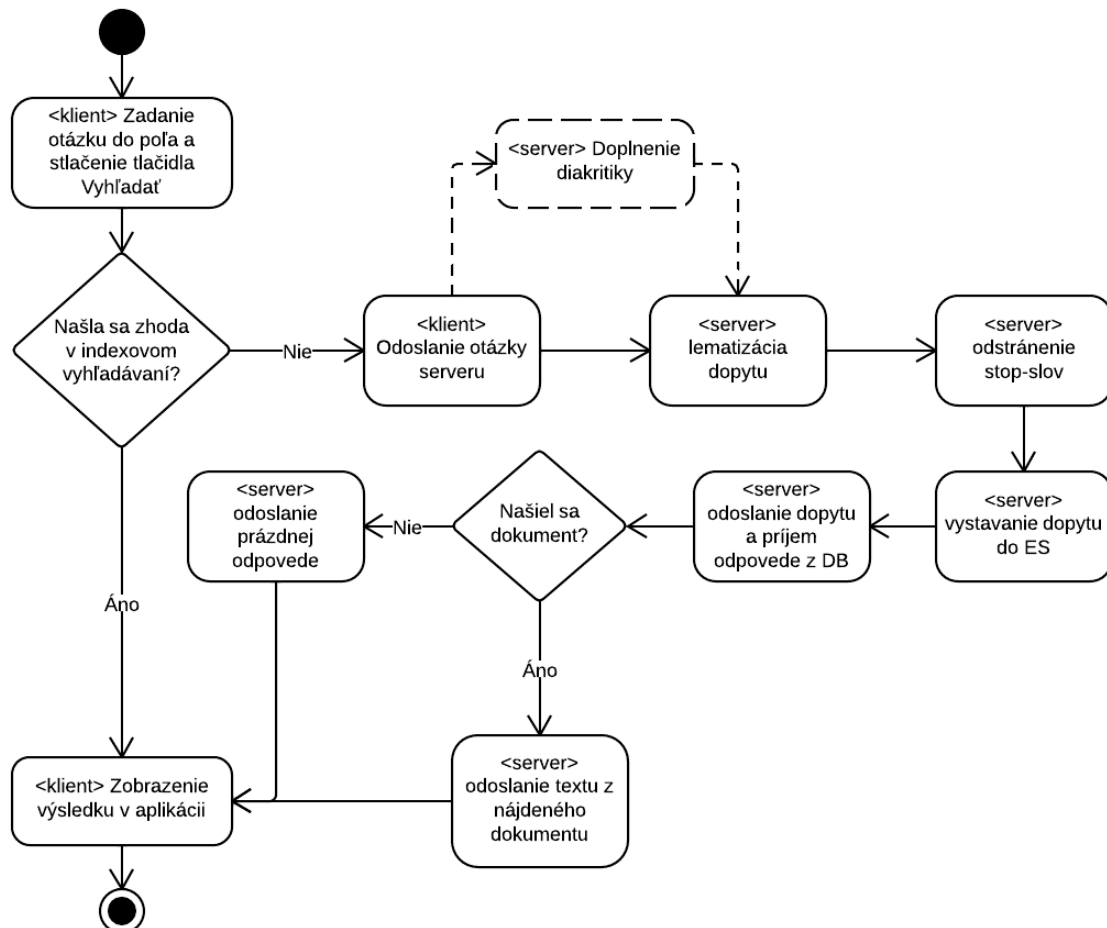
Ďalšou dôležitou vecou, ktorá sa nám podarila implementovať a ktorá vzišla z potrieb zákazníka bolo možnosť označovať jednotlivé linky ako obľúbené, čím sa pridajú do virtuálneho zoznamu a takéto linky sa budú prednostne zobrazovať vo výpise na vrchu ako prvé.

### 26.1.6 Jedálne

Jedálne ako ďalšia zo sekcií prešla výraznou obmenou. Vstavaný iFrame, ktorý sa nachádzal v pôvodnej verzii aplikácie bol nahradený za plnohodnotnú integráciu jedálnych lístkov. Je možné si vybrať jedálny lístok niektorej zo študentských jedální, ktoré sa nachádzajú v okolí. Pri každej z jedální je zobrazená hneď aj otváracia doba od kedy do kedy je otvorená. Taktiež je možné prepínať medzi jednotlivými dňami v týždni v ponuke jednotlivých jedální. V hornej navigačnej lište sa nachádza informačné tlačidlo, po ktorého stlačení sa zobrazí obrazovka s otváracími hodinami a odkazom na mapu a domovskú stránku jedálne.

## 26.1.7 Inteligentné vyhľadávanie

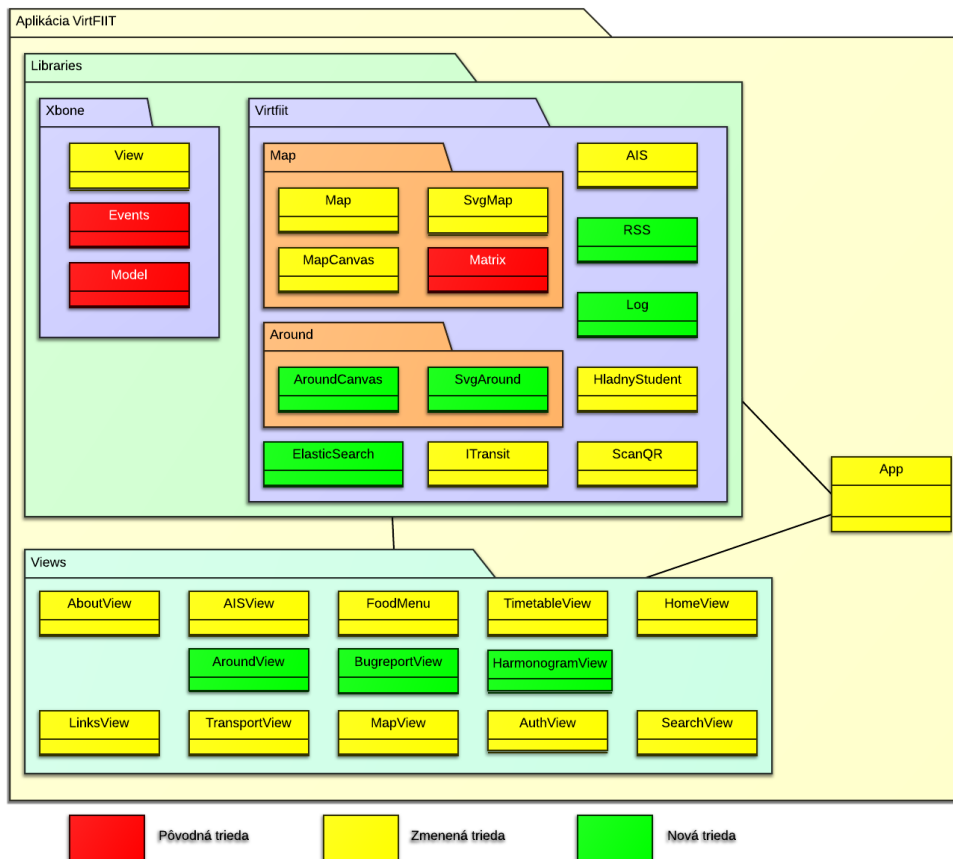
Inteligentné vyhľadávanie je postavené nad databázou iného projektu – Študentského radcu. Našou prácou bolo vytvorenie inteligentného vyhľadávania nad databázou otázok a odpovedí zložitým spracovaním otázky a špeciálnym vyhľadávaním cez ElasticSearch nad databázou pre čo najlepší výsledok.



Obrázok 45 Aktivita diagram inteligentného vyhľadávania. Komponent označený prerušovanou čiarou ešte nebol na serveri naimplementovaný, ale do budúcnosti sa s ním počíta.

## 26.2 Architektúra a modely

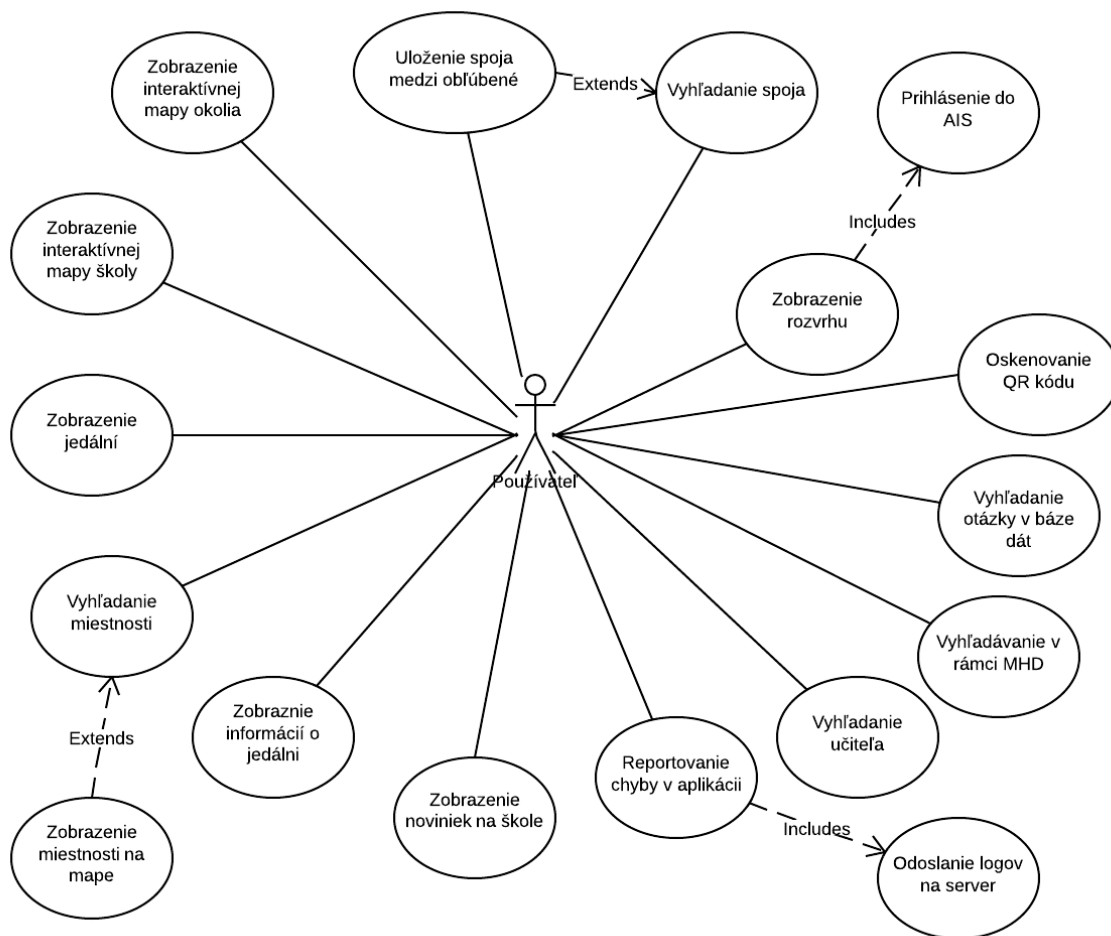
Aplikácia pozostáva zo spúšťača, modelov, viewov, templatov a ďalších podporných javascriptových knižníc, ktoré sú potrebné pre jej fungovanie. Základná architektonická štruktúra je vyobrazená na obrázku základná architektonická štruktúra.



**Obrázok 46 základná štruktúra tried**

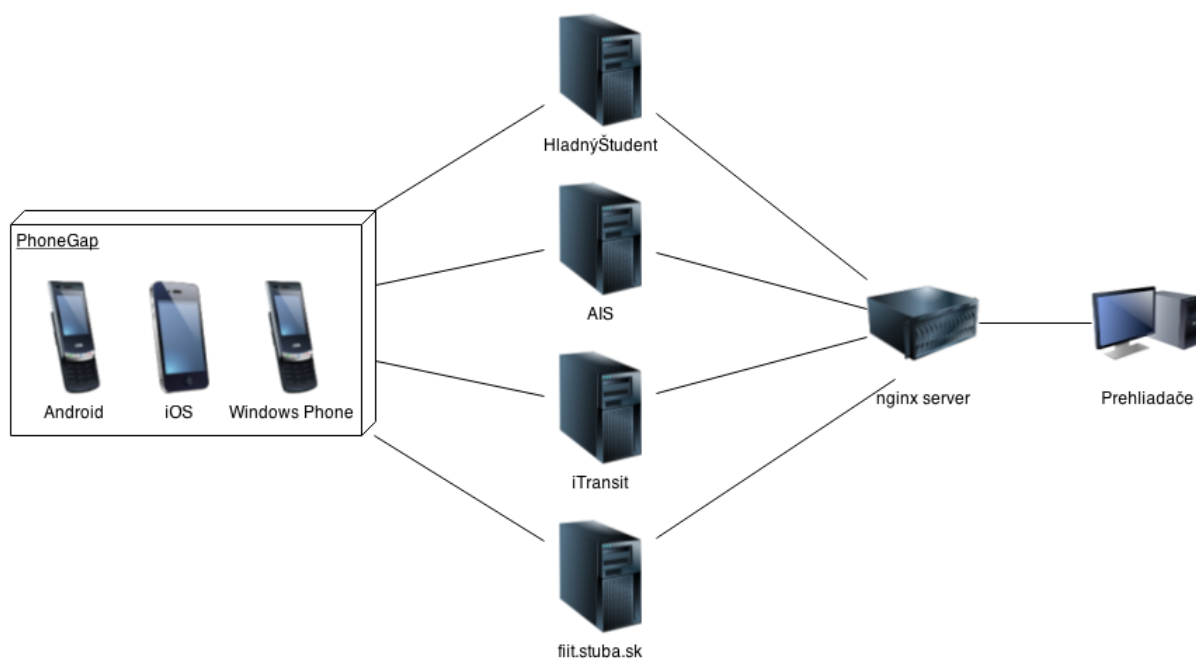
Trieda App je hlavným spúšťačom celej aplikácie. Niektoré z view-ov v balíku Views s modelmi Libraries aktívne spolupracujú počas celej doby spustenia aplikácie. View-y taktiež nepretržite komunikujú spolu s App, ktorý pôsobí aj ako usmerňovač.





**Obrázok 47 use case diagram pre používateľa aplikácie**

Na predošlom obrázku možno vidieť všetky akcie, ktoré môže používateľ vykonávať v našej aplikácii.

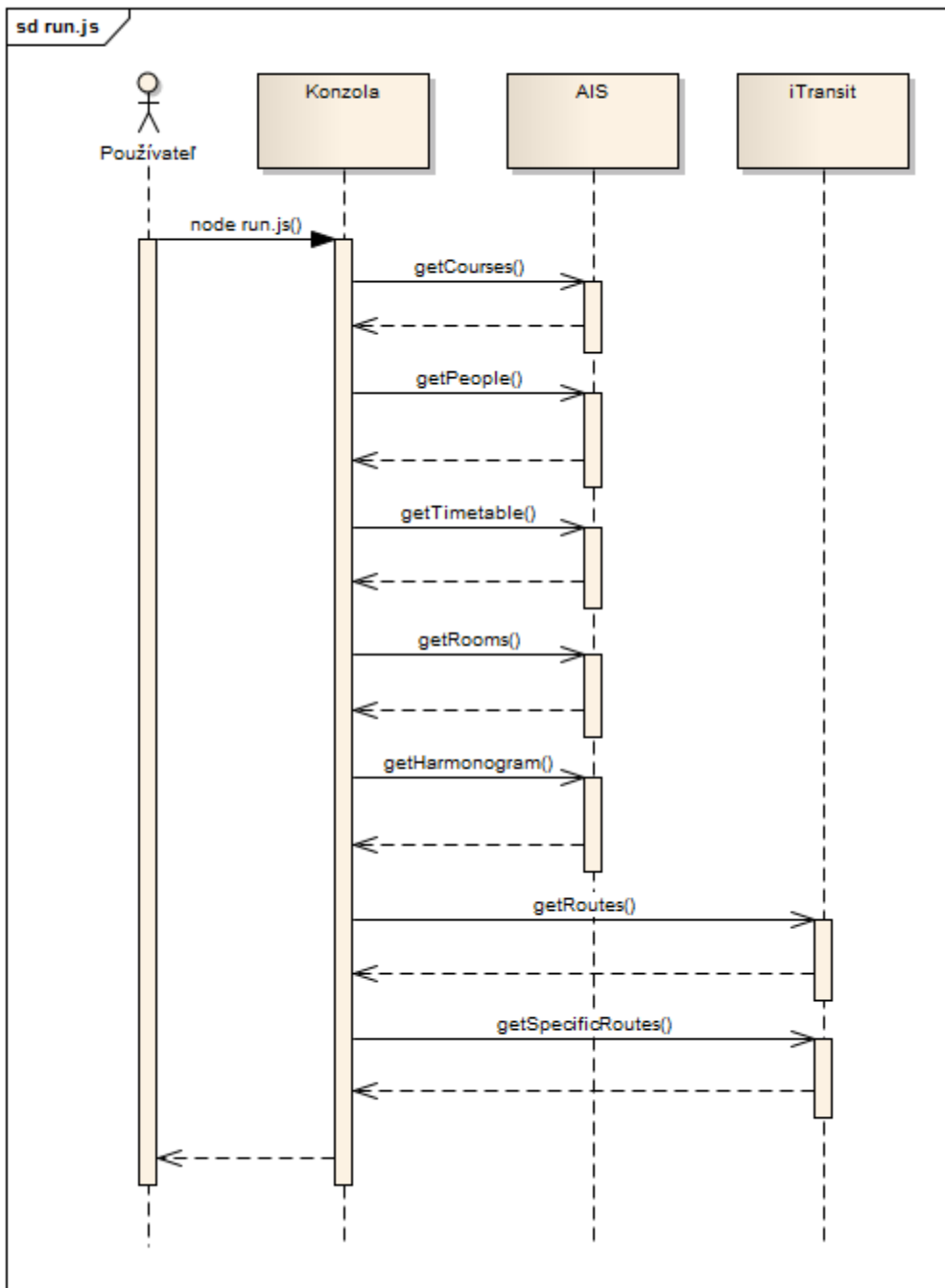


**Obrázok 48 technologická vrstva aplikácie**

Predchádzajúci obrázok opisuje technologickú vrstvu aplikácie. Zobrazuje jej mobilnú verziu, ktorá sa pripája na vzdialené servery aby získala dáta na priamo a jej webovú verziu ktorá sa pripája na náš nginx server ktorý sa následne pripája na tieto servery.

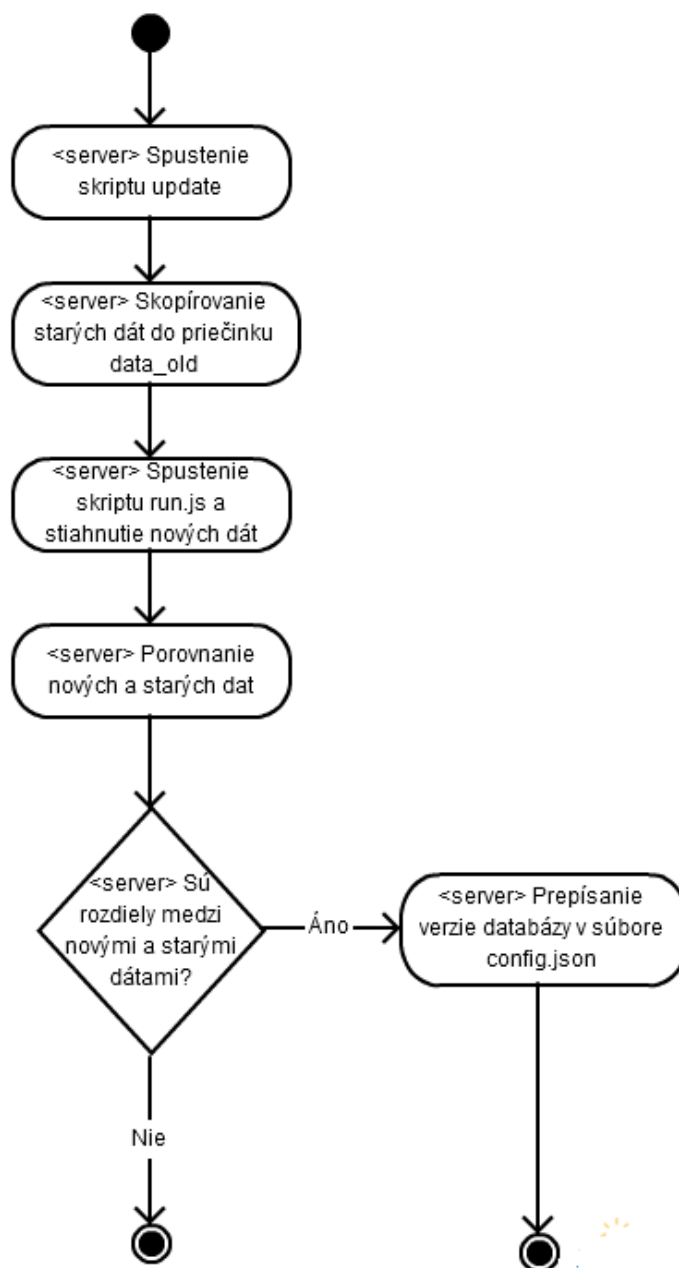
### 26.2.1 Aktualizácia dát v aplikácii

Pre aktualizáciu dát tretích strán pre našu aplikáciu je treba dáta predspracovať na našom servery a až následne odtiaľ si ich bude sťahovať aplikácia. Podrobný spôsob aktualizácie znázorňuje nasledujúci diagram.



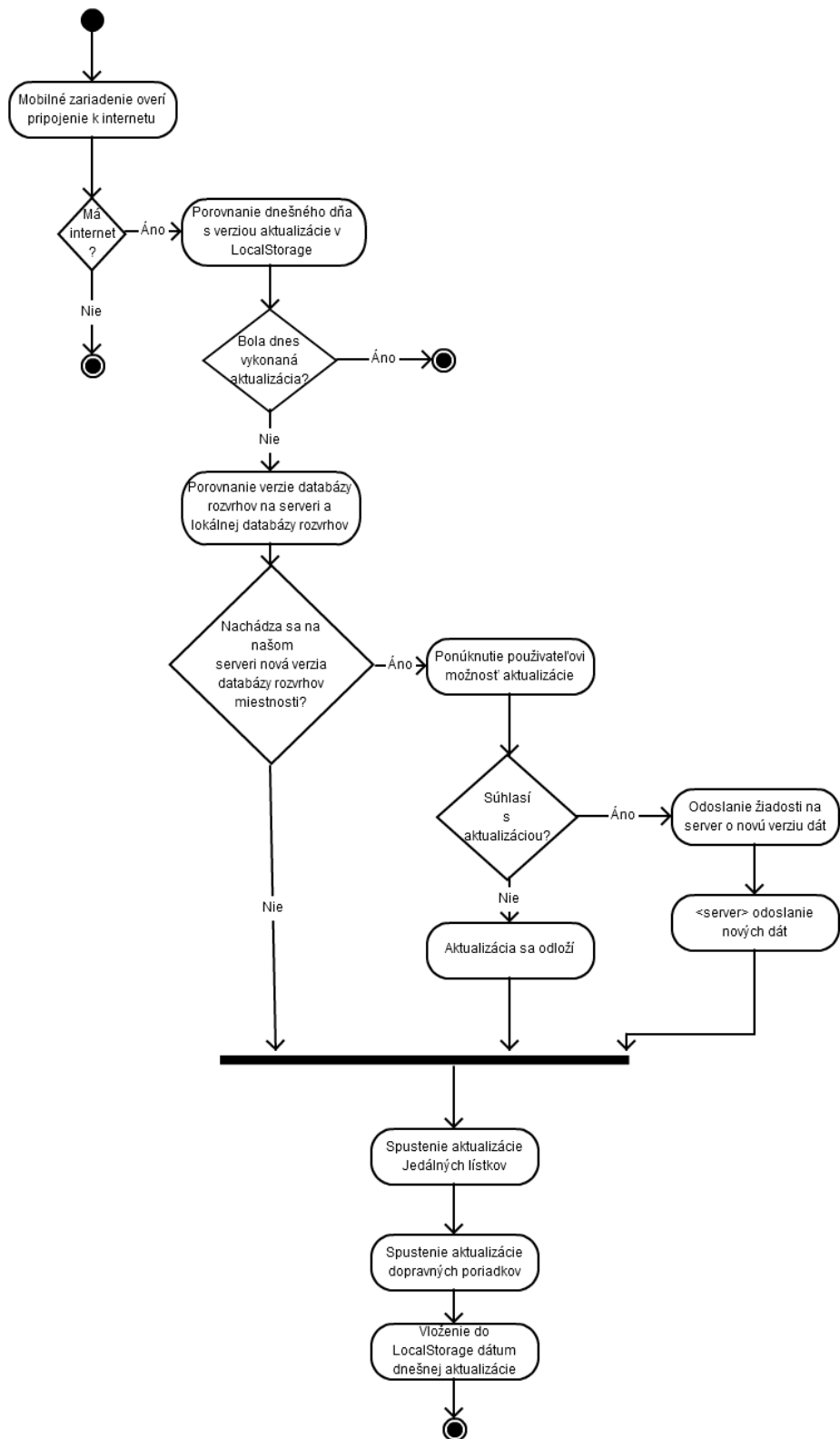
Obrázok 49 Predspracovanie dát tretích strán na servery

Tento proces sa spúšťa každý deň o 4:00 ráno. Jeho automatické spúšťanie je naplánované v crontab na našom tímovom servery



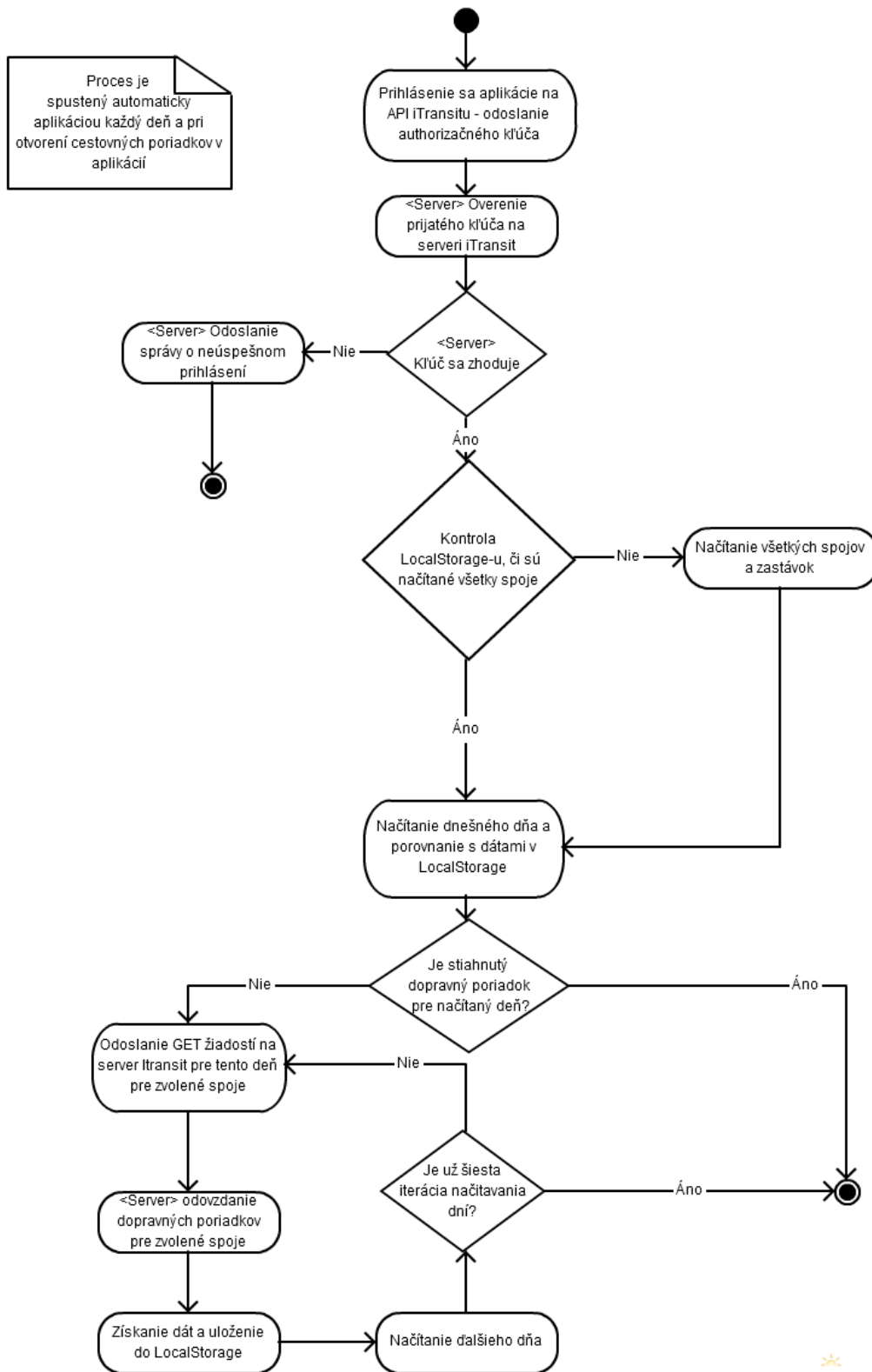
Obrázok 50 Aktivita diagram aktualizácie informácií o vyučujúcich, rozvrhov a miestnostiach

Aktualizácia rozvrhov miestnosti prebieha na našom serveri, spustením skriptu update. Tento skript načíta všetky miestnosti a následne odošle na server stavba.fiit.stuba.sk. S tohto serveru si nové dáta stiahne zariadenie.



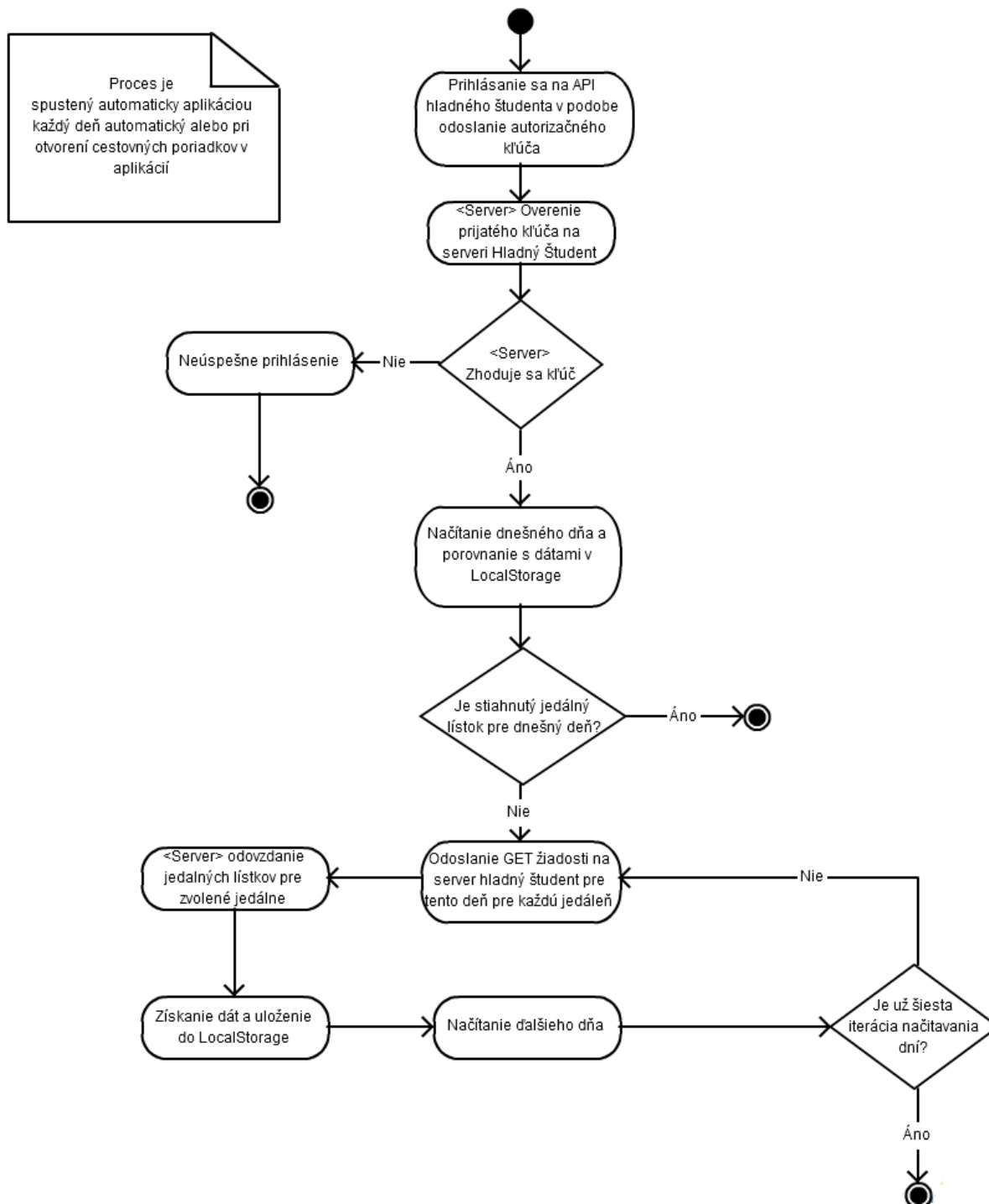
Obrázok 51 Aktivita diagram aktualizácie dát na zariadení

Na zariadení sa spúšťa aktualizácia pri zapnutom prístupe k internetu, inteligentne kontroluje či je potrebné aktualizáciu spúšťať, alebo nie. Toto spúšťanie prebieha každý deň aj keď je zariadenie spustené celý deň.



Obrázok 52 Aktivita diagram aktualizácie cestovných poriadkov

Cestovné poriadky sa sťahujú zo servera iTransit. Po prihlásení sa odošlú žiadosti o získanie cestovných poriadkov pre zadané linky a dátumy. Tieto dáta sa potom ukladajú do lokálnej databázy zariadenia.



Obrázok 53 Aktivita diagram aktualizácie jedálnych lístkov

Jedálne lístky sa sťahujú zo servera hladný študent. Po prihlásení sa odošlú žiadosti pre získanie jedálnych lístkov pre zadané jedálne a dni. Dáta sa taktiež ukladajú do lokálnej databázy zariadenia. Jedálne lístky aj cestovné poriadky sa sťahujú na šesť dní dopredu.

## **26.3 Ohraničenia**

### **26.3.1 Operačné systémy**

Aplikácia v aktuálnej verzii funguje na operačnom systéme Android od verzie 2.3.3. Taktiež sa dá používať webová verzia aplikácie. Žiadne iné operačné systémy zatiaľ nie sú podporované.

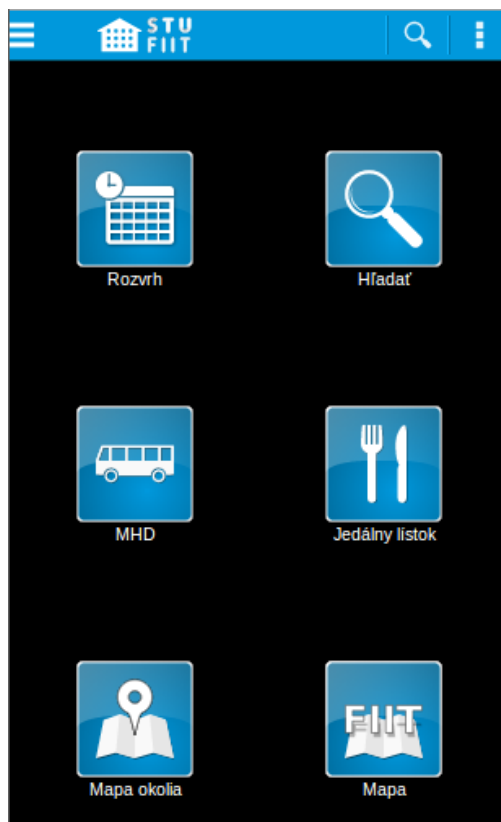
### **26.3.2 Zástavky MHD**

V časti MHD zobrazujeme len odchody zo zastávok Zoo a Botanická záhrada.

## **26.4 Používateľská príručka**

Po zapnutí aplikácie sa používateľ dostane do hlavného menu aplikácie. Tento bod tvorí centrálny uzol v aplikácii. Z neho sa dá dostať do všetkých základných sekcií, ktoré naša aplikácia ponúka. Ide o sekciu Rozvrh, Hľadať, MHD, Jedálny lístok, Mapa okolia a Mapa. Prístup k ďalšej funkcionalite sa nachádza vo vysúvacom menu do ktorého je možné sa dostať kliknutím do ľavej hornej časti (tri paličky) našej aplikácie na hlavnej obrazovke. Aplikácia taktiež obsahuje aj druhé menu (tri kocky).

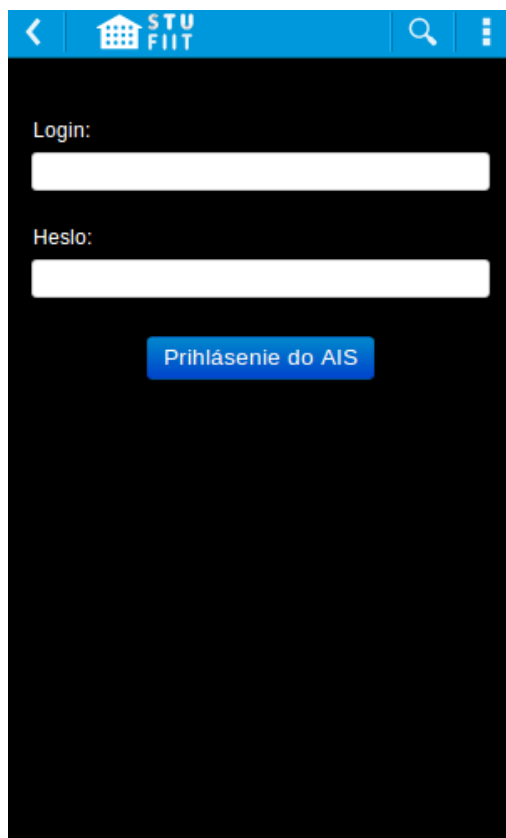




Obrázok 54 Hlavná obrazovka

### 26.4.1 Rozvrh

Pre zobrazenie svojho rozvrhu je potrebné sa prihlásiť pomocou prihlasovacích údajov do AIS. Po úspešnom prihlásení vidí používateľ svoj rozvrh v ktorom sa môže prostredníctvom dní preklikať.



Obrázok 55 Neprihlásený používateľ



Obrázok 56 Prihlásený používateľ

## 26.4.2 Vyhľadávanie

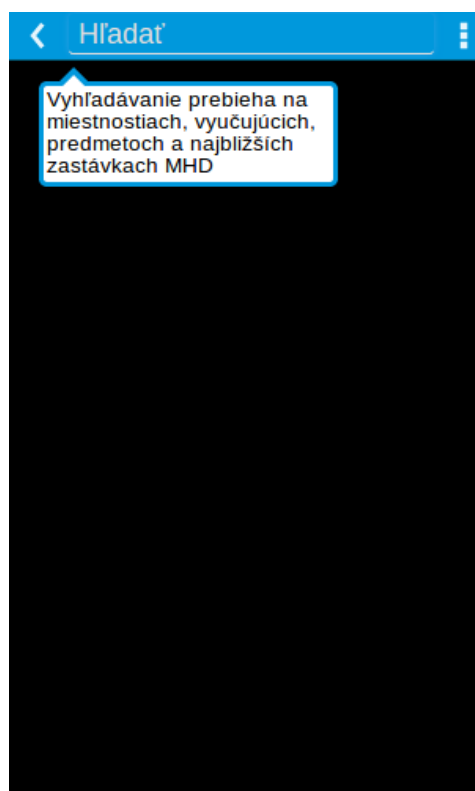
Ku vyhľadávaniu v našej aplikácii je možné sa dostať dvoma spôsobmi. Prvým spôsobom je prostredníctvom lupy, ktorá sa nachádza v hornej lište, alebo pomocou ikonky na hlavnej ploche. Vyhľadávať je možné v predmetoch, miestnostiach osobách, ktoré vyučujú na FIIT a zastávkach MHD spojov, ktoré odchádzajú zo zastávok Botanická záhrada a ZOO. Týka sa to iba denných spojov, nočné spoje nevidujeme. Vyhľadať je možné všetky tie zastávky, ktoré sa nachádzajú v smere jazdy pred danou zastávkou. Tj. zastávky, ktoré sa nachádzajú na trase pred nástupnou zastávkou (Zoo, Botanická záhrada) nie je možné vyhľadať. Po kliknutí na niektorú zo zastávok je používateľ presmerovaný do MHD sekcie kde sa mu otvorí daná linka v požadovanom smere.

Cieľom takéhoto vyhľadávania nie je v žiadnom prípade nahradenie aplikácie iTransit. Aj naďalej platí, že v aplikácii sa nachádzajú odchody spojov LEN zo zastávok Botanická záhrada a Zoo. Cieľom je sprístupniť možnosť zistiť či sa daná zastávka nachádza na trase niektorých z našich evidovaných spojov.

Vo výsledkoch vyhľadávania sa zobrazuje vždy iba prvých 20 záznamov.



Obrázok 57 Prázdny formulár na vyhľadávanie



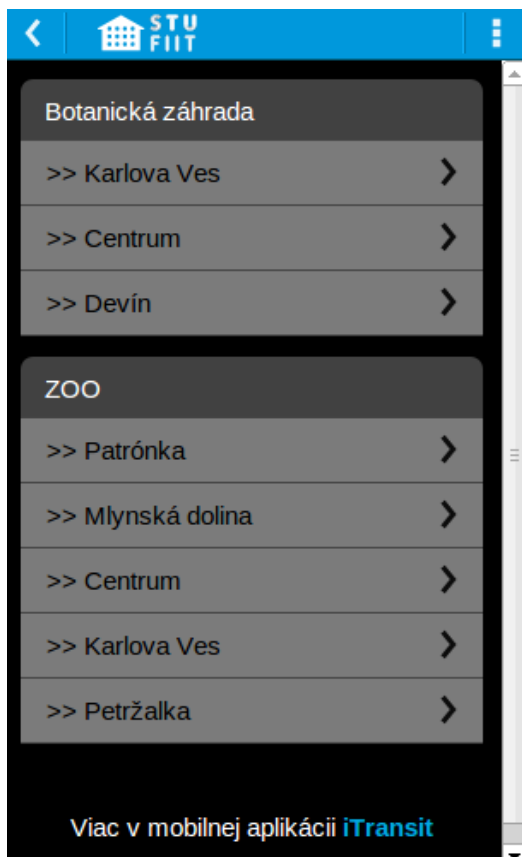
Obrázok 58 Výsledky vyhľadávania

## 26.4.3 MHD

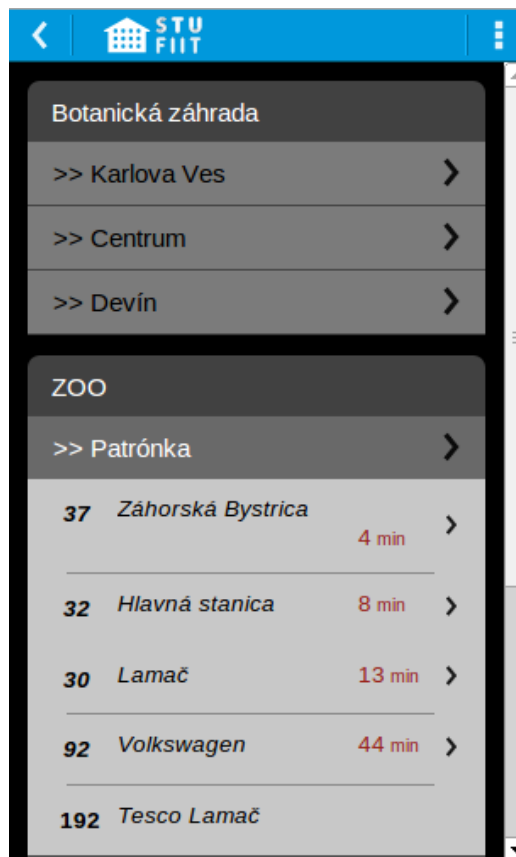
Prostredníctvom sekcie MHD je možné zistiť najbližšie odchody zo zastávok Zoo a Botanická záhrada. Po kliknutí na niektorý zo smerov sa zobrazia linky, ktoré z danej

zastávky idú na daný smer. O takejto linke je možné sa dozvedieť jej číslo, konečnú zastávku a počet minút do najbližšieho odchodu. Po kliknutí na konkrétnu linku sa zobrazia všetky odchody v daný deň.

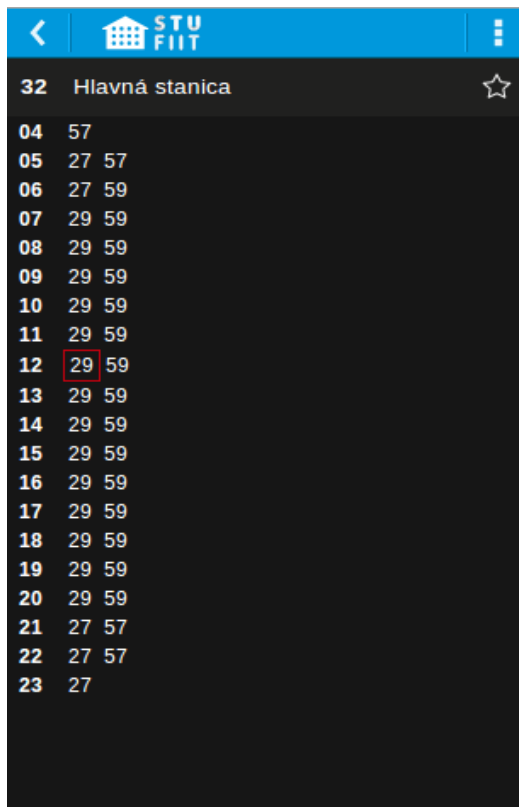
Každú linku MHD je možné uložiť si medzi svoje obľúbené linky. Obľúbené linky sú také linky, ktorými používateľ cestuje najčastejšie. Takéto linky sa používateľovi prednostne zobrazujú v tejto sekcii návrhu obrazovky. Linku je možné pridať medzi obľúbené jednoduchým kliknutím na obrázok hviezdičky, ktorá sa nachádza na obrazovke so všetkými odchodmi linky MHD v daný deň. Odoberanie prebieha rovnako. Pri pridávaní a odoberaní obľúbenej linky sa zmení len jej ikonka.



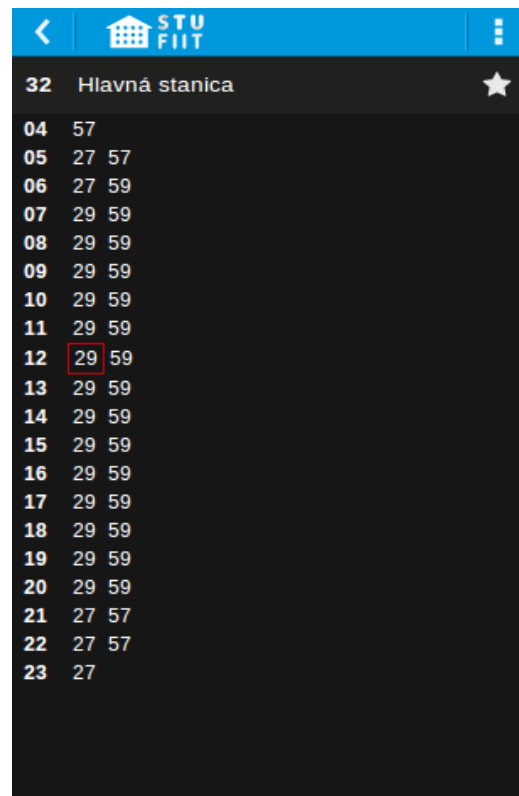
Obrázok 59 MHD sekcia



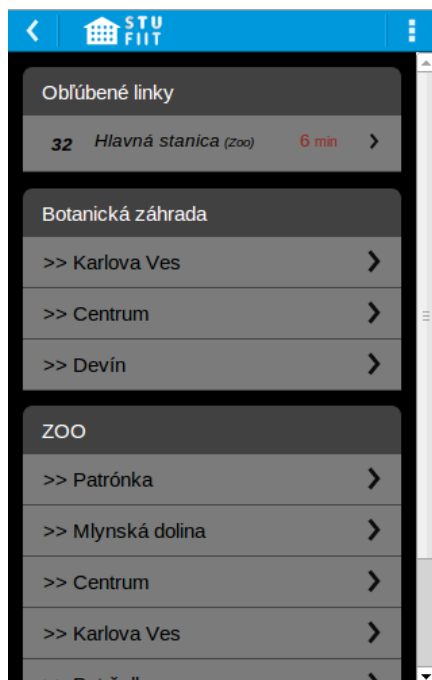
Obrázok 60 Rozkliknuté menu MHD



Obrázok 62 Odchody linky 32



Obrázok 61 Linka 32 označená ako obľúbená

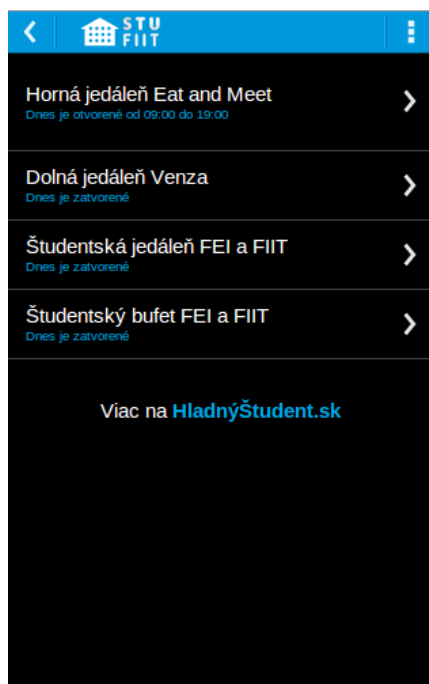


Obrázok 63 MHD Sekcia s obľúbenými linkami

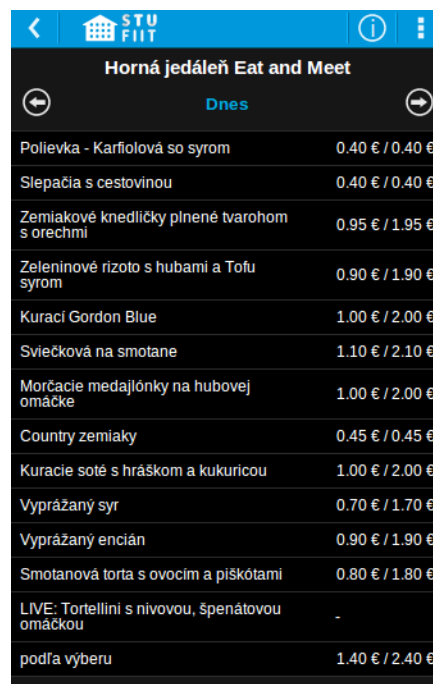
## 26.4.4 Jedálny lístok

V jedálnom lístku sa nachádzajú všetky študentské jedálne, v blízkom okolí FIIT. O každej jedálni vieme hneď zistiť odkedy dokedy je otvorená.

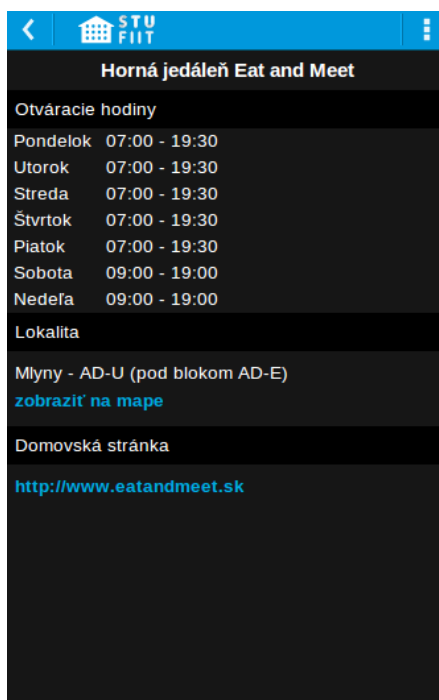
Po kliknutí na konkrétnu jedáleň sa nám otvorí aktuálna denná ponuka. Klikaním na šípky v hornej časti aplikácie sa vieme prepínať medzi jednotlivými dňami v týždni. V hornej lište sa nachádza aj informačné tlačidlo, ktoré keď zatlačíme sa nám ukáže jedáleň na mape.



Obrázok 64 Všetky jedálne



Obrázok 65 Jedálny lístok



Obrázok 66 Informácie o jedálni

### 26.4.5 Mapa okolia

V sekcii mapa okolia je vyobrazená Mlynská dolina so všetkými dôležitými orientačnými bodmi akými sú umiestnenie internátov, zastávok MHD, smerovanie ciest, bufety a jedálne, parkoviská.

V tejto sekcii sa nachádza aj kompas, ktorý ukazuje na smer ako sme vzhľadom na mapu otočený (neukazuje na sever – nie je to jeho funkcionalitou).



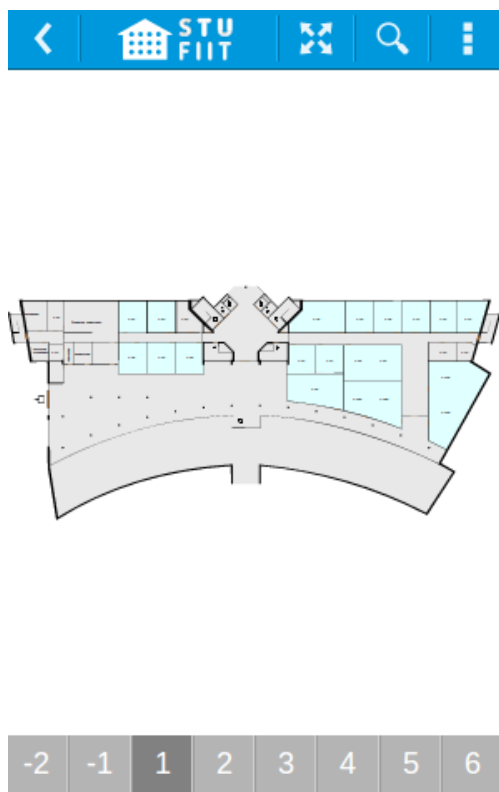
Obrázok 67 Mapa okolia

### 26.4.6 Mapa

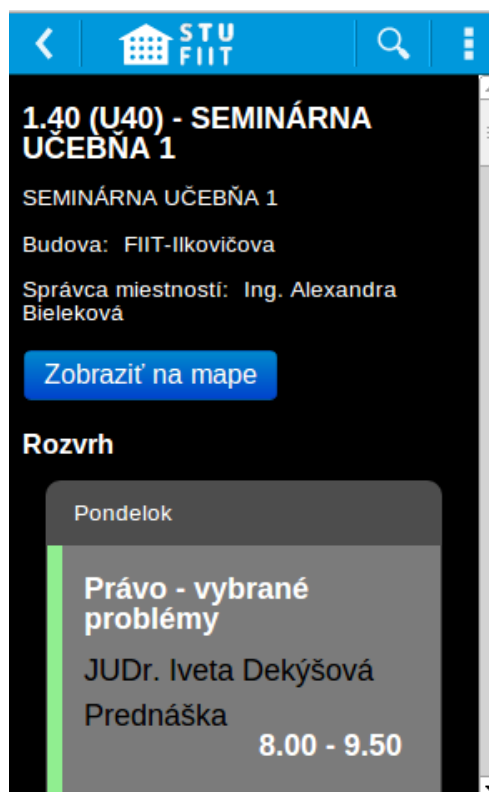
V sekcii mapa je možné prepínať sa medzi jednotlivými poschodiami budovy. Je možné kliknúť na jednotlivé miestnosti s číslami. Po kliknutí na miestnosť sa zobrazí buď rozvrh vyučovania v danej miestnosti, alebo sa zobrazí zoznam osôb, ktoré sedia v danej kancelárii.

Danú miestnosť je možné spätne zobraziť na mape. Po kliknutí na meno osoby, ktorá sedí v danej kancelárii sa otvorí vizitka tejto osoby a je možné z tejto vizitky zistiť mail, telefónny kontakt, predmety, ktoré učí a aktuálny rozvrh. Po kliknutí na miestnosť v rozvrhu sa zobrazí spätne mapa s vysvietenou miestnosťou, ktorú sme chceli zobraziť. Po kliknutí na predmet, ktorý učí daná osoba sa zobrazí celkový rozvrh predmetu pre daný semester.

Na mape sa nachádza aj kompas, ktorý ukazuje smer ktorým sme otočený.



Obrázok 68 Mapa poschodia

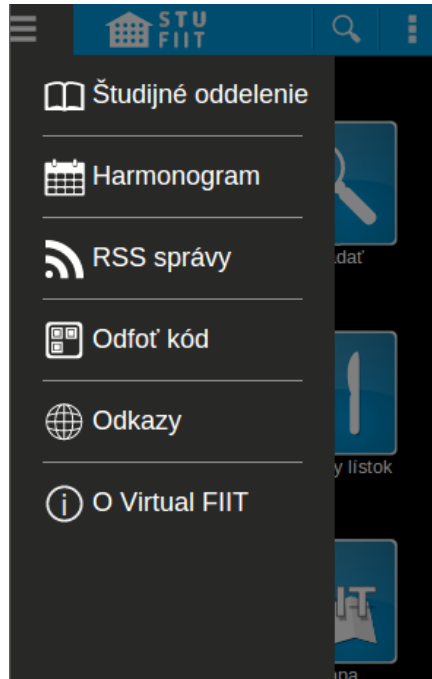


Obrázok 69 Vybraná miestnosť

### 26.4.7 Vysúvacie menu

Vysúvacie menu poskytuje prístup ku QR čítačke, zobrazeniu dôležitým odkazov, aktuálnemu daniu na fakulte prostredníctvom RSS kanála. Ďalej zobrazuje aktuálny školský harmonogram, informácie o štúdijnom oddelení a informáciám o aplikácií.





Obrázok 70 Vysúvacie menu

#### **26.4.7.1 Študijné oddelenie**

Táto sekcia zobrazuje informácie o tom kedy je otvorené študijné oddelenie.

#### **26.4.7.2 Harmonogram**

V tejto sekcii je možné nájsť aktuálny harmonogram na daný akademický rok.

#### **26.4.7.3 RSS správy**

Poskytuje používateľovi informácie o aktuálnom dianí na fakulte.

#### **26.4.7.4 Odfot' kód**

Odfot' kód slúži na odfovenie QR kódu, ktorými bude vybavená budova FIITky v blízkej budúcnosti. Takéto QR kódy bude možné nájsť na kanceláriách vyučujúcich. Takýmto jednoduchým spôsobom bude možné zistiť všetky základné informácie o danom vyučujúcom. QR kód bude obsahovať vizitku danej osoby.

#### **26.4.7.5 Odkazy**

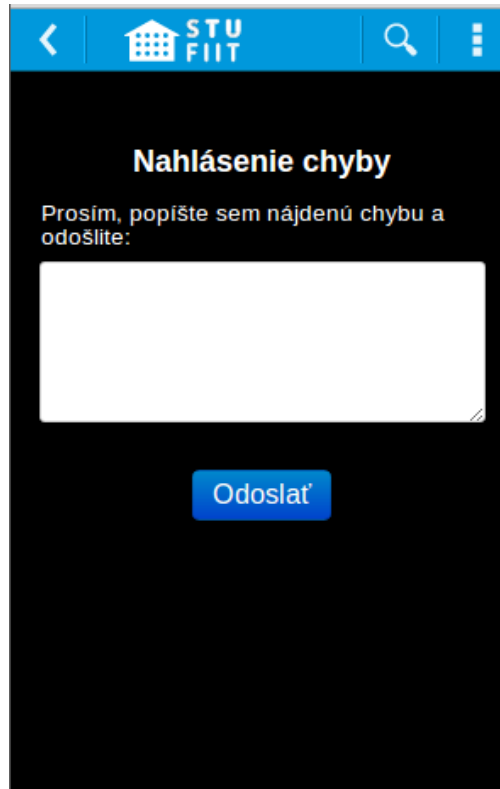
Slúžia ako centrálny repozitár na úschovu všetkých dôležitých odkazov, ktoré sa môžu študentovi, vyučujúcemu, alebo návštevníkovi fakulty zísť.

#### **26.4.7.6O Virtual FIIT**

V tejto sekcii je možné sa dočítať o všetkých tímoch, ktoré na tejto aplikácii spolupracovali pred nami. Taktiež sa tu nachádza aj aktuálna verzia aplikácie a kontaktný email pre nahlasovanie chýb, ktoré sa môžu vyskytnúť počas doby používania aplikácie.

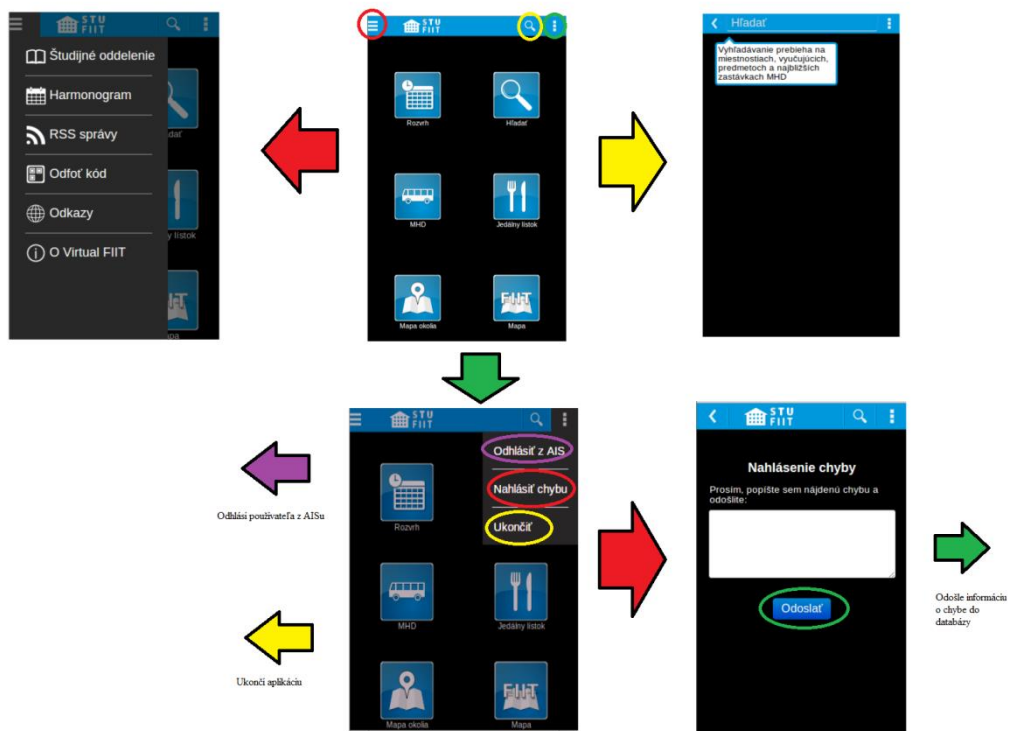
## 26.4.8 O aplikácií

Pomocou tlačidla troch kociek nad sebou v hornej lište sa nám ponúknu dve možnosti: Nahlásiť chybu a Ukončiť. Stlačením „Ukončiť“ ukončíme našu prácu s aplikáciou a aplikácia sa bezpečne vypne. Kliknutím na nahlásiť chybu sa nám zobrazí formulár, prostredníctvom ktorého je možné odoslať bližší popis chyby, ktorú používateľ našiel.

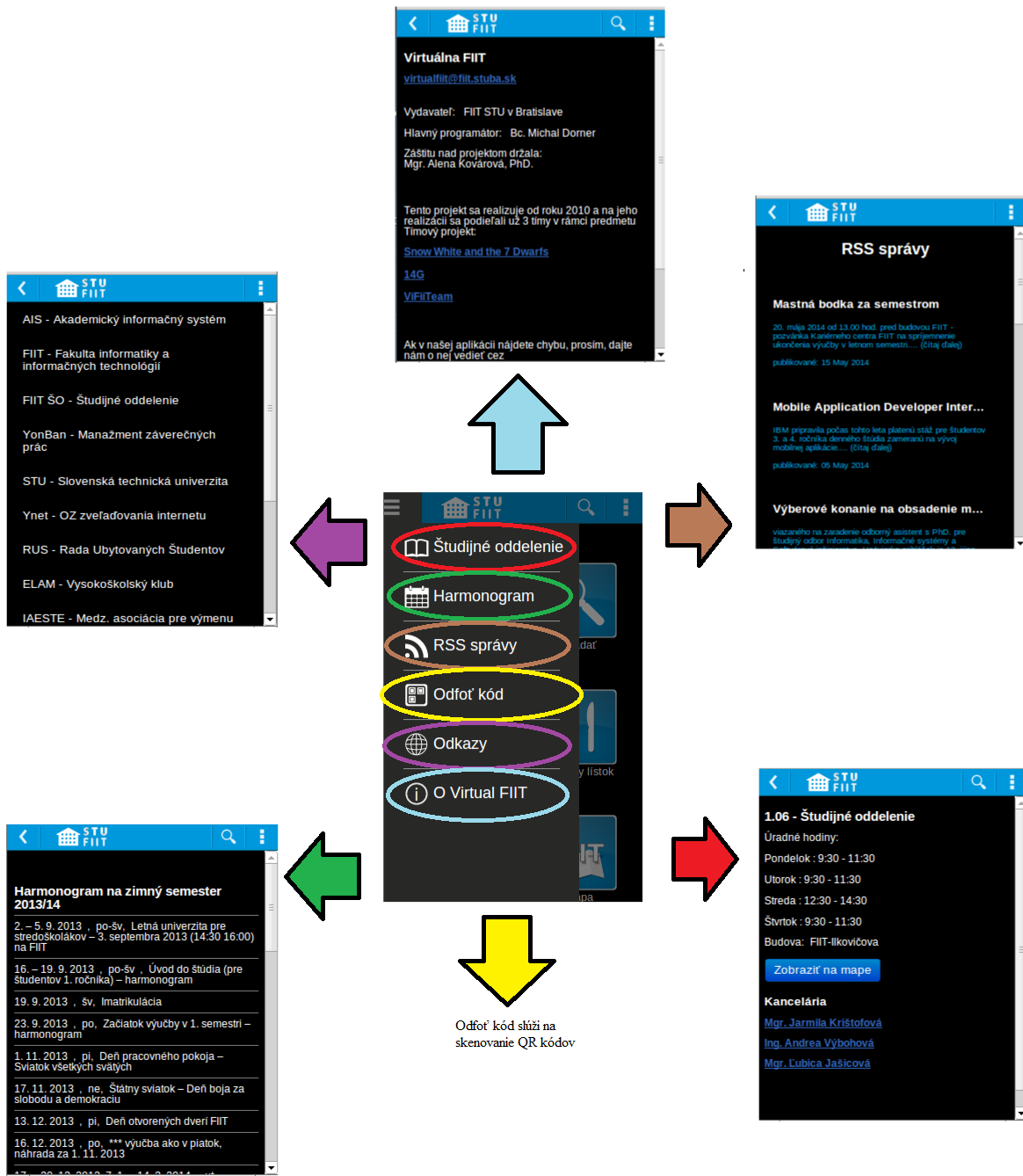


Obrázok 71 Formulár na odosielanie chýb

## 26.5 Podrobné obrazovky produktu



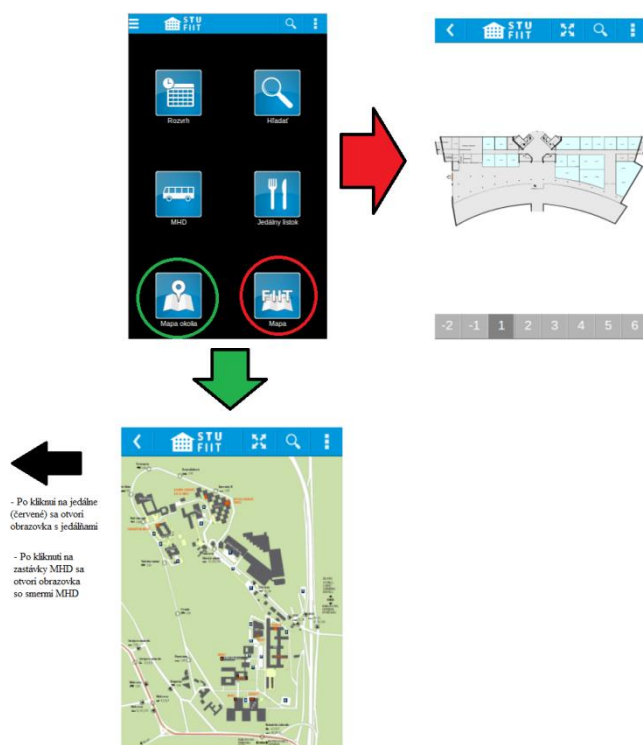
Obrázok 72 Diagram obrazoviek hlavného menu



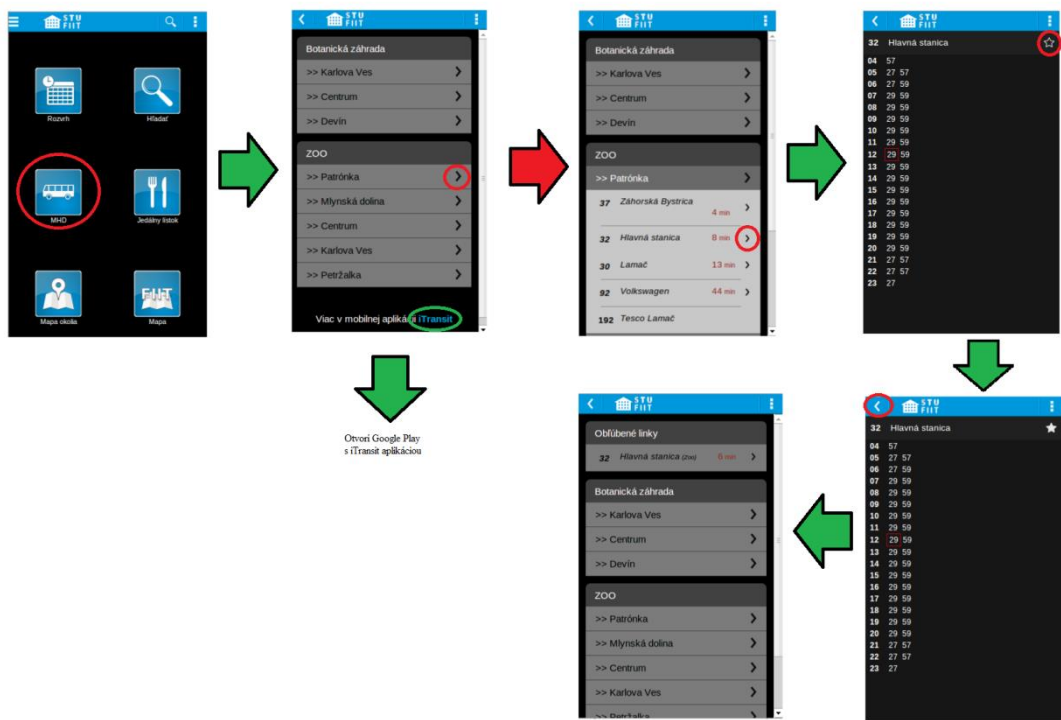
Obrázok 73 Diagram obrazoviek bočného menu



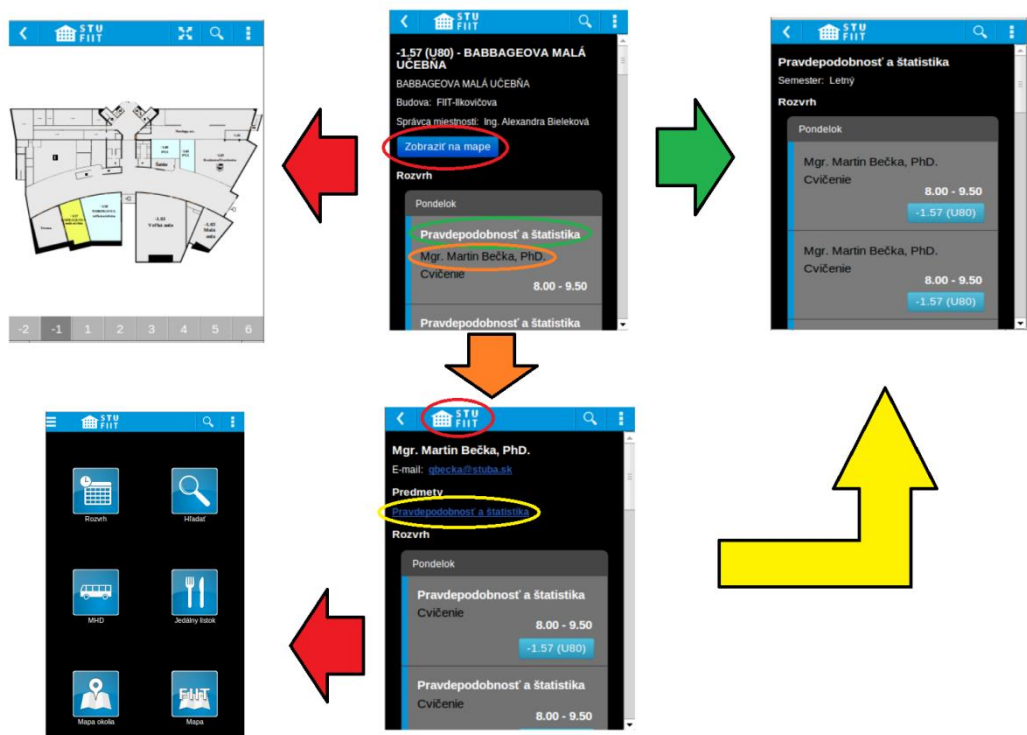
Obrázok 74 Diagram obrazoviek jedálnych lístkov



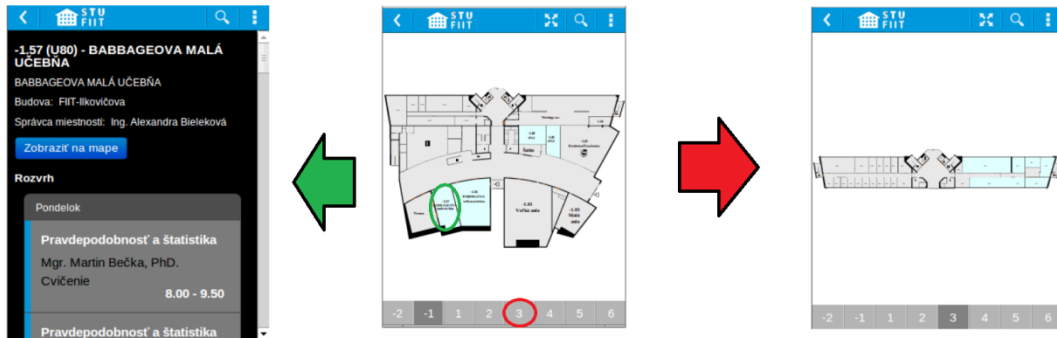
Obrázok 75 Diagram obrazoviek máp



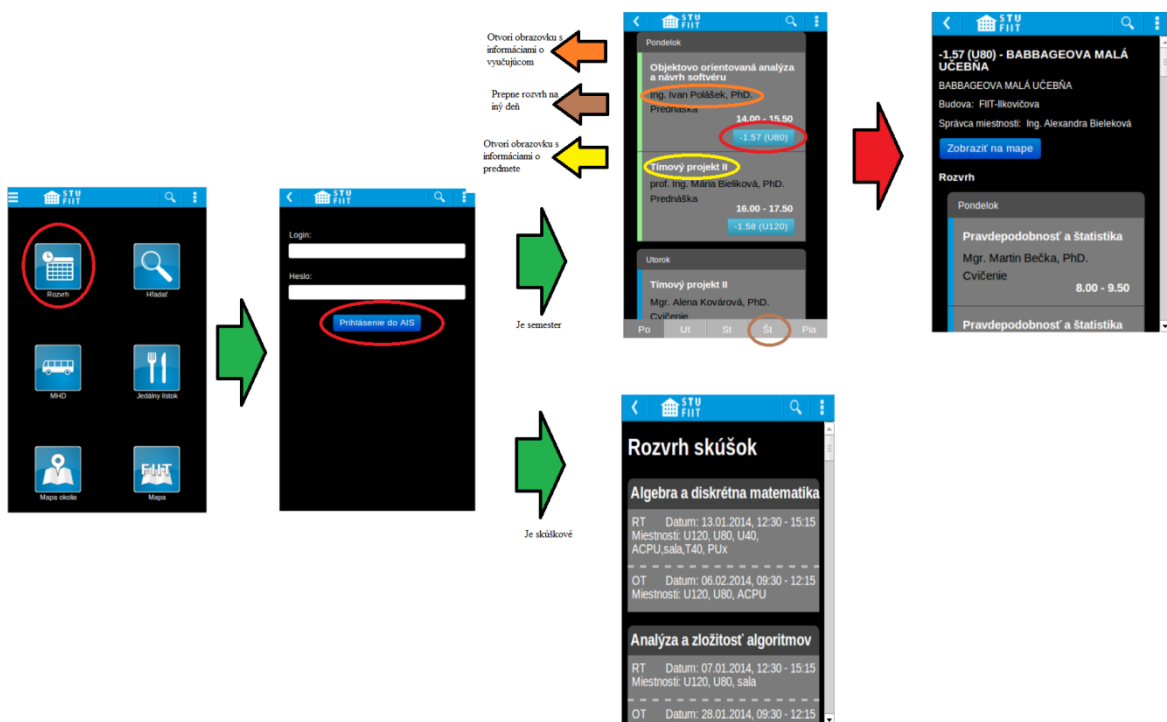
Obrázok 76 Diagram obrazoviek cestovných poriadkov



Obrázok 77 Diagram obrazoviek vyučujúcich



Obrázok 78 Diagram obrazoviek pre miestnosti



Obrázok 79 Diagram obrazoviek pre osobný rozvrh



Obrázok 80 Diagram obrazoviek pre vyhľadavanie