



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

## ZADÁNÍ DIPLOMOVÉ PRÁCE

<b>Název:</b>	Mobilní aplikace pro Dopravní podnik města Jihlavy a.s.
<b>Student:</b>	Bc. František Koumar
<b>Vedoucí:</b>	Ing. Petra Pavlíčková, Ph.D.
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Webové a softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce zimního semestru 2021/22

### Pokyny pro vypracování

Cílem práce je pro Dopravní podnik města Jihlavy a.s. (DPMJ) navrhnout a implementovat mobilní aplikaci, která nahradí stávající aplikaci pro zobrazování jízdních řádů a zároveň ji rozšíří o funkcionality dalších informačních portálů provozovaných DPMJ.

- 1) Proveďte rešerši obdobných aplikací.
- 2) Prozkoumejte možnosti vývoje aplikací pro mobilní zařízení.
- 3) Proveďte analýzu požadovaných funkcionalit.
- 4) Navrhněte a implementujte prototyp.
- 5) Aplikaci řádně otestujte.
- 6) Zhodnoťte a navrhněte případná budoucí vylepšení aplikace.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.  
děkan

V Praze dne 25. února 2020





**FAKULTA  
INFORMAČNÍCH  
TECHNologiÍ  
ČVUT V PRAZE**

Diplomová práce

**Mobilní aplikace pro  
Dopravní podnik města Jihlavy a.s.**

*Bc. František Koumar*

Katedra softwarového inženýrství

Vedoucí práce: Ing. Petra Pavlíčková, Ph.D.

28. května 2020



---

## Poděkování

Tímto bych rád poděkoval vedoucí mé práce paní Ing. Petra Pavlíčková, Ph.D. za trpělivost, ochotu a cenné rady, které napomohly vzniku této práce. Dále bych také rád poděkoval všem zaměstnancům Dopravního podniku města Jihlavy a jihlavského magistrátu za jejich ochotnou spolupráci a cenné připomínky.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 28. května 2020

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2020 František Koumar. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Koumar, František. *Mobilní aplikace pro Dopravní podnik města Jihlavy a.s.*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.



---

# Abstrakt

Tato práce se zabývá realizací mobilní aplikace pro Dopravní podnik města Jihlavy a.s. určené pro operační systém Android, která uživatelům umožní přistupovat k informacím, které jsou dopravním podnikem poskytovány veřejnosti, v rámci jedné aplikace, a zároveň realizací serverové části aplikace sloužící pro správu offline podkladů, které jsou mobilní částí využívány v rámci její funkcionality.

V rámci práce je provedena rešerše obdobných mobilních aplikací a s využitím metod softwarového inženýrství je vypracována analýza požadavků, na jejímž základě je navrhnout a implementován funkční prototyp aplikace, který je následně důkladně otestován.

V závěru práce je pak navrženo několik vylepšení, která by mohla být do budoucna do aplikace zapracována.

Výsledný funkční prototyp aplikace pak svojí funkcionalitou úspěšně nahrazuje stávající a již nevyhovující aplikaci pro vyhledávání v jízdních řádech a zároveň přináší novou funkcionalitu v podobě online mapy dopravy, možnosti zobrazení aktuálních odjezdů z dané zastávky či možnosti zobrazení aktualit o změnách v dopravě.

**Klíčová slova** jízdní řády, mobilní aplikace, Android, PHP

# Abstract

The subject of this thesis is realization of mobile application for Dopravní podnik města Jihlavy a.s., which will make it possible for its users to access information that is provided to public by Dopravní podnik města Jihlavy a.s. via single application, and also realization of server part of the application which will be used for administration of offline resources needed for the functionality of the mobile part.

A recherche of similar mobile application was carried out within the thesis, as well as analysis of requirements which was the base of subsequent design and implementation of a functional prototype of the application which was then thoroughly tested.

Finally a few possible improvements are suggested that might be implemented within the application in the future.

The final functional prototype of the application successfully replaces original and now unsuitable application used for searching within timetables and also introduces new functionality in the form of online map of public transport as well as possibility to display actual departures from given bus stop or news about current changes in public transport.

**Keywords** timetables, mobile application, Android, PHP

---

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Rešerše obdobných aplikací</b>	<b>5</b>
2.1 Aplikace JRm . . . . .	5
2.2 Jízdní řády IDOS . . . . .	7
2.3 Jízdní řády Seznam.cz . . . . .	8
2.4 PID Lítačka . . . . .	9
2.5 DPMBinfo . . . . .	10
2.6 DPMCB . . . . .	11
2.7 Moje PMDP . . . . .	12
2.8 Moje DPO . . . . .	12
2.9 Závěr . . . . .	13
<b>3 Vývoj aplikací pro mobilní platformy</b>	<b>15</b>
3.1 Responzivní webová aplikace . . . . .	15
3.2 Progresivní webová aplikace . . . . .	17
3.3 Nativní mobilní aplikace . . . . .	18
3.4 Závěr . . . . .	19
<b>4 Analýza</b>	<b>21</b>
4.1 Rozbor požadavků . . . . .	21
4.1.1 Funkční požadavky . . . . .	23
4.1.2 Nefunkční požadavky . . . . .	24
4.2 Rozdělení systému . . . . .	24
4.3 Případy Užití . . . . .	25
4.3.1 Správcovská část systému . . . . .	25
4.3.2 Klientská část systému . . . . .	27

4.4	Závěr . . . . .	28
<b>5</b>	<b>Výběr technologií</b>	<b>29</b>
5.1	Platforma rozhraní správcovské části . . . . .	29
5.2	Technologie správcovské části . . . . .	31
<b>6</b>	<b>Návrh uživatelského rozhraní</b>	<b>37</b>
6.1	Zásady tvorby uživatelského rozhraní . . . . .	37
6.2	Volba barevného schématu . . . . .	39
6.3	Správcovská část systému . . . . .	40
6.4	Klientská část systému . . . . .	43
<b>7</b>	<b>Návrh a implementace</b>	<b>47</b>
7.1	Správcovská část . . . . .	47
7.2	Klientská část . . . . .	52
<b>8</b>	<b>Testování</b>	<b>57</b>
8.1	Testování software obecně . . . . .	57
8.2	Správcovská část systému . . . . .	58
8.3	Klientská část systému . . . . .	59
8.4	Závěr . . . . .	60
<b>9</b>	<b>Další rozvoj aplikace</b>	<b>61</b>
	<b>Závěr</b>	<b>65</b>
	<b>Bibliografie</b>	<b>67</b>
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>71</b>
<b>B</b>	<b>Přílohy</b>	<b>73</b>
B.1	Případy užití . . . . .	73
B.1.1	Serverová část . . . . .	73
B.1.2	Klientská část . . . . .	81
B.1.2.1	Podčást jízdní řády . . . . .	81
B.1.2.2	Podčást online mapa dopravy . . . . .	83
B.1.2.3	Podčást aktuality . . . . .	86
B.2	Drátěné modely . . . . .	86
B.2.1	Správcovská část . . . . .	86
B.2.2	Klientská část . . . . .	93
<b>C</b>	<b>Obsah přiloženého DVD</b>	<b>97</b>

---

## Seznam obrázků

2.1	Záběry obrazovky aplikace JRm . . . . .	6
2.2	Záběry obrazovky aplikace Jízdní řády IDOS . . . . .	7
2.3	Záběry obrazovky aplikace Jízdní řády (Seznam.cz) . . . . .	8
2.4	Záběry obrazovky aplikace PID Lítačka . . . . .	9
2.5	Záběry obrazovky aplikace DPMBinfo . . . . .	11
2.6	Záběry obrazovky aplikace Moje DPO . . . . .	13
3.1	Ilustrace mobile-first principu [10] . . . . .	16
4.1	Důvody vzniku problémů u SW projektu . . . . .	22
4.2	Důvody selhání SW projektu . . . . .	22
4.3	Vývoj ceny opravení chyby v rámci vývojového procesu [20] . . . . .	23
5.1	TIOBE rating top 10. Data dostupná na [26] . . . . .	33
5.2	PYPL Index top 10. Data dostupná na [27] . . . . .	34
6.1	Výsledné barevné schéma klientské části . . . . .	39
6.2	Drátěný model obrazovky pro správu jízdních řádů . . . . .	40
6.3	Drátěný model obrazovky pro správu jízdních řádů - úprava položky . . . . .	41
6.4	Task diagram správcovské části . . . . .	42
6.5	Ukázka drátěných modelů klientské části 1 . . . . .	43
6.6	Ukázka drátěných modelů klientské části 2 . . . . .	44
6.7	Task diagram klientské části . . . . .	46
7.1	Databázové schéma správcovské části systému . . . . .	51
7.2	Databázové schéma SQLite databáze jízdních řádů . . . . .	51
7.3	Zjednodušený třídní model správcovské části systému . . . . .	52
7.4	Zjednodušený třídní model dotazů . . . . .	54



---

# Úvod

Informační technologie jsou v dnešní době využívány stále častěji a stěží bychom dnes hledali společnost, jejíž činnost by nebyla usnadněna využitím množství ať již navzájem provázaných či samostatných informačních systémů.

Hlavním účelem těchto systémů jak již bylo řečeno je především usnadnění vlastní činnosti společnosti, jejích procesů atd. Jejich využití nicméně v mnoha případech umožňuje společností rovněž sdílet se svými zákazníky či širokou veřejností informace, u kterých to dříve nebylo možné nebo přinejmenším velmi složité. Mnohdy jsou ovšem tyto informace sdíleny různými způsoby a prostředky, ačkoliv spolu souvisejí a uživatel tak musí namáhavě vyhledávat potřebné informace na mnoha různých místech.

Podobnou situaci je možné pozorovat i v rámci Dopravního podniku města Jihlavy a.s., který poskytuje svým cestujícím množství různých informací a to od aktualit na svém webovém portálu, přes jízdní řady v mobilní aplikaci až po možnost online sledování polohy jednotlivých vozů na mapě. Tyto informace jsou však dnes sdíleny prostřednictvím různých systémů, což je pro koncového uživatele značně nepraktické.

Dopravním podnikem aktuálně využívaná mobilní aplikace pro jízdní řady navíc trpí dlouhodobě mnohými nedostatky v podobě různých chyb ve funkcionality, častými pády aplikace, či nevhodně navrženého uživatelského rozhraní.

Právě proto vznikla myšlenka na realizaci nové aplikace, která by napravila nedostatky původní aplikace pro jízdní řady a zároveň by uživatelům umožnila přístup i k dalším zdrojům informací o hromadné dopravě jako např. ke zmíněné online mapě dopravy či aktualitám o změnách v dopravě z webového portálu.

A právě realizace této aplikace je hlavní náplní této práce.





---

## Cíl práce

Autor této práce si klade za cíl vypracovat funkční prototyp mobilní aplikace, která by nahradila Dopravním podnikem města Jihlavy a.s. aktuálně využívanou a již nevyhovující aplikaci umožňující uživatelům vyhledávání v jízdních řádech, a zároveň v rámci implementované aplikace zahrnout funkcionalitu dalších informačních systémů, prostřednictvím kterých dopravní podnik poskytuje další potřebné či zajímavé informace o městské hromadné dopravě, mezi něž patří například online mapa dopravy, jež umožňuje cestujícím sledovat aktuální polohu a zpoždění jednotlivých vozů.

V rámci práce pak autor dále rovněž navrhne vylepšení, která by do výsledné aplikace mohla být v budoucnu zapracována.



---

## Rešerše obdobných aplikací

Před začátkem vývoje je vhodné provést průzkum za účelem nalezení produktů obdobného typu, které jsou již na trhu k dispozici, a to především v případě komerčních aplikací, u kterých se jedná o praktickou nutnost vzhledem ke konkurenčnímu boji, kdy je nutné dopředu přesně určit potenciální konkurenci tak, aby bylo možné dopředu říci, zda-li má aplikace šanci na trhu uspět, či případně identifikovat nedostatky konkurenčních řešení, při jejichž vyřešení bude šance na úspěch aplikace vyšší.

Nemalou roli ovšem může takovýto průzkum hrát i v případě, že se jedná o vývoj nekomerční aplikace, jako je tomu v případě této práce.

V případě nekomerčních aplikací je rovněž vhodné se seznámit s nedostatky obdobných řešení a pokud je to možné se těmto nedostatkům vyvarovat. Takovýto průzkum rovněž umožňuje se případně inspirovat, co se týče řešení uživatelského rozhraní či zajímavé funkcionality, kterou je možné následně do vyvíjené aplikace zapracovat.

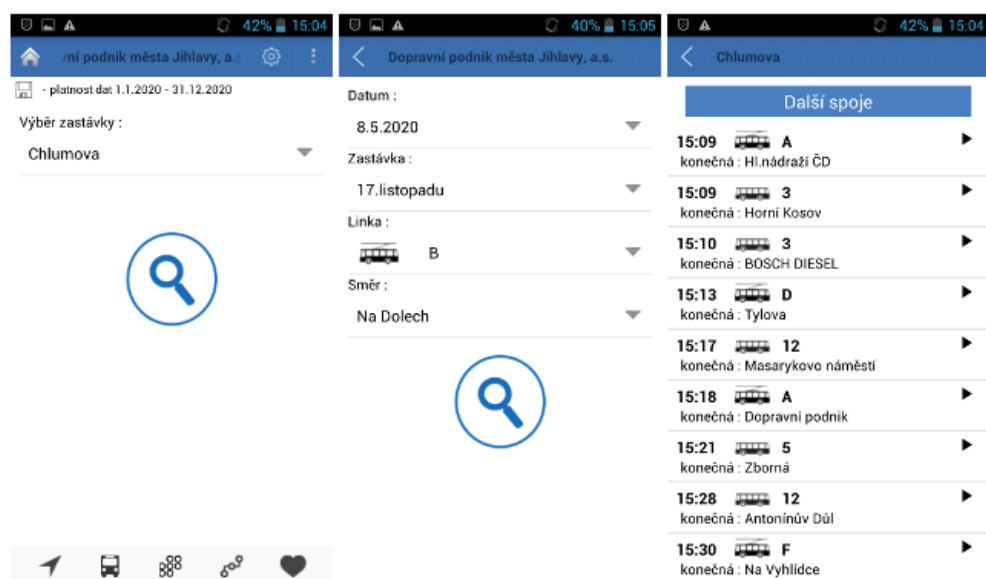
### 2.1 Aplikace JRm

Jako první bude představena aplikace JRm [1], která je aktuálně Dopravním podnikem města Jihlavy využívána pro prezentování jízdních řádů cestujícím v rámci jejich mobilních zařízení, a jejíž kompletní nahrazení je hlavním cílem této práce.

Jedná se o aplikaci softwarové společnosti FS Software, jejíž hlavní činností je vývoj software pro plánování a řízení dopravy, který je využíván mnohými dopravními podniky a to včetně DPMJ.

Aplikace umožňuje vyhledávat v rámci jízdních řádů odjezdy z jednotlivých zastávek, spojení mezi zastávkami, jízdní řády jednotlivých linek či jízdních řádů zastávek a to buď v online režimu, či po stažení potřebných dat rovněž offline.

## 2. REŠERŠE OBDOBNÝCH APLIKACÍ



Obrázek 2.1: Záběry obrazovky aplikace JRm

V aplikaci je rovněž uživateli umožněno uložení jím často vyhledávaných dotazů na jízdní řády do oblíbených položek, čímž uživatel získá rychlejší přístup ke kýženým informacím. Nicméně není z nějakého důvodu možné uložit jako oblíbený dotaz na odjezdy, což je z autorovi zkušenosti jeden z nejvyužívanějších typů dotazů pro uživatele, který MHD využívá častěji.

Aplikace ovšem trpí značným množstvím problémů, které ji činí využitelnou jen částečně, či v případě některých zařízení zcela nevyužitelnou.

Hlavními problémy aplikace jsou její časté pády, jakož to i částečná nefunkčnost některých částí aplikace, jakými jsou například vyhledávání spojení, či zobrazení dalších spojů na seznamu odjezdů.

V případě pokusu o přechod na vyhledávání spojení aplikace padá a jejich vyhledání tak není možné. Při vyhledávání odjezdů pak aplikace sice zobrazí seznam kýžených odjezdů, ale nejedná se o kompletní seznam, nýbrž jen o několik málo počátečních odjezdů, přičemž zobrazení dalších není možné vzhledem k nefunkčnosti příslušného tlačítka pro zobrazení dalších spojů.

U některých zařízení pak dochází k pádu aplikace již při jejím spuštění a její funkcionalitu tak na těchto zařízeních není možné využít.

Aplikace rovněž trpí nedostatky, co se týče návrhu uživatelského rozhraní. Některé položky uživatelského rozhraní, se kterými má koncový uživatel interagovat, jsou značně malé, což značně zhoršuje jejich použitelnost na dotykových zařízeních.

Nejllepší příklad tohoto nešvaru je možné vidět na obrázku 2.1 v pravé části, kde je vyobrazen seznam vyhledávaných odjezdů. Jak je možné vidět,

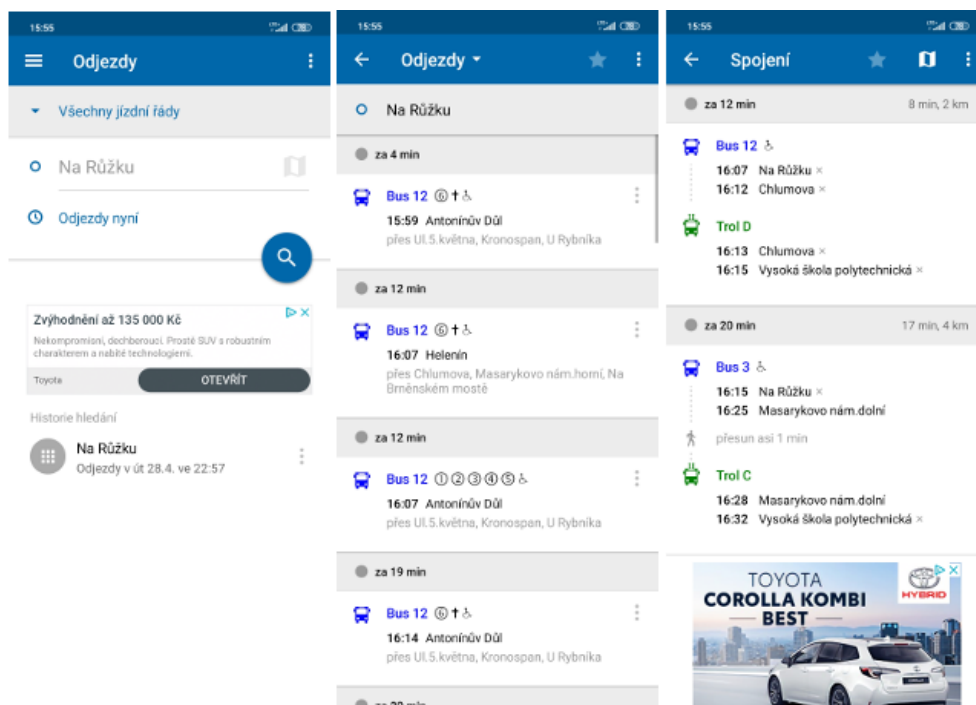
jednotlivé položky disponují vpravo šipkami, které indikují, že je na položku možné kliknout pro zobrazení detailu spoje. U moderní mobilní aplikace by uživatel očekával, že po kliknutí na libovolné místo na položce či případně do pravé části, kde se nachází ona šipka, bude zobrazen detail spoje. V tomto případě je však "klikatelná" část položky omezena pouze ne nejblíže okolí šipky a uživatel ji tak může snadno minout.

## 2.2 Jízdní řády IDOS

Jako druhá v pořadí bude představena aplikace Jízdní řády IDOS společnosti MAFRA a.s. [2]

Systém IDOS představuje asi nejznámější systém pro vyhledávání autobusových a vlakových spojení a to včetně MHD.

Jeho mobilní verze pak, kromě vyhledávání spojení umožňuje vyhledávat i odjezdy z jednotlivých stanic či zastávek.



Obrázek 2.2: Záběry obrazovky aplikace Jízdní řády IDOS

Na rozdíl od aplikace JRm pak aplikace IDOS nabízí navíc možnost zobrazení spojení na mapě, v níž je možné rovněž provést výběr zastávky respektive zastávek pro vyhledání odjezdů a spojení.

Uživatelsky zajímavou funkcionalitou je pak uchování historie dotazů, kdy je uživateli na obrazovkách pro vyhledávání spojení a odjezdů prezentováno

## 2. REŠERŠE OBDOBNÝCH APLIKACÍ

několik z naposledy realizovaných dotazů. Na druhou stranu však není přítomna možnost uložení oblíbených položek vyhledávání.

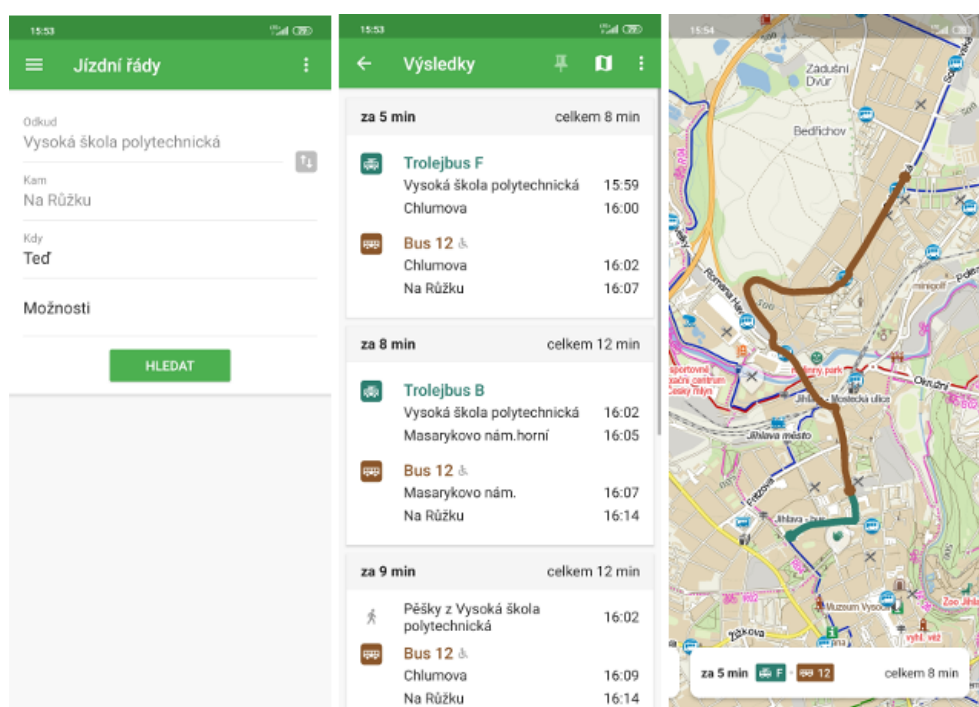
Další zajímavou funkcionalitou je poté možnost definovat pokročilé parametry hledání jako například možnost volby pouze přímých spojení či určení minimálního času pro přestup.

Ve výchozím stavu ovšem aplikace zobrazuje reklamy, jak lze vidět i na obrázku 2.2 a zbavit se jich dá pouze zakoupením placené verze aplikace.

Aplikace rovněž umožňuje vyhledávání pouze s připojením k internetu a nenabízí jakoukoli formu offline vyhledávání.

### 2.3 Jízdní řády Seznam.cz

Další obdobnou aplikací je aplikace Jízdní řády od společnosti Seznam.cz. [3]



Obrázek 2.3: Záběry obrazovky aplikace Jízdní řády (Seznam.cz)

Na rozdíl od předchozích aplikací umožňuje aplikace z dílen Seznamu vyhledávat pouze spojení a možnost vyhledat odjezdy zde zcela chybí.

Stejně jako v případě aplikace IDOS však nabízí upřesnění možností spojení, či ukládání historie dotazů. Navíc pak oproti IDOSu nabízí možnost uložit oblíbené položky.

Aplikace rovněž nenabízí možnost offline vyhledávání, ale na rozdíl od IDOSu je zde přece jenom přítomna určitá míra offline funkcionality, kdy

aplikace ukládá výsledky již realizovaných dotazů, které je následně možné zobrazit z historie dotazů, či opětovné realizace totožného dotazu v rámci určitého krátkého časového období, po který považuje aplikace předchozí výsledek za platný.

Uživatelské rozhraní aplikace, které je možné vidět na obrázku 2.3, je pak velmi podobné tomu z aplikace IDOS.

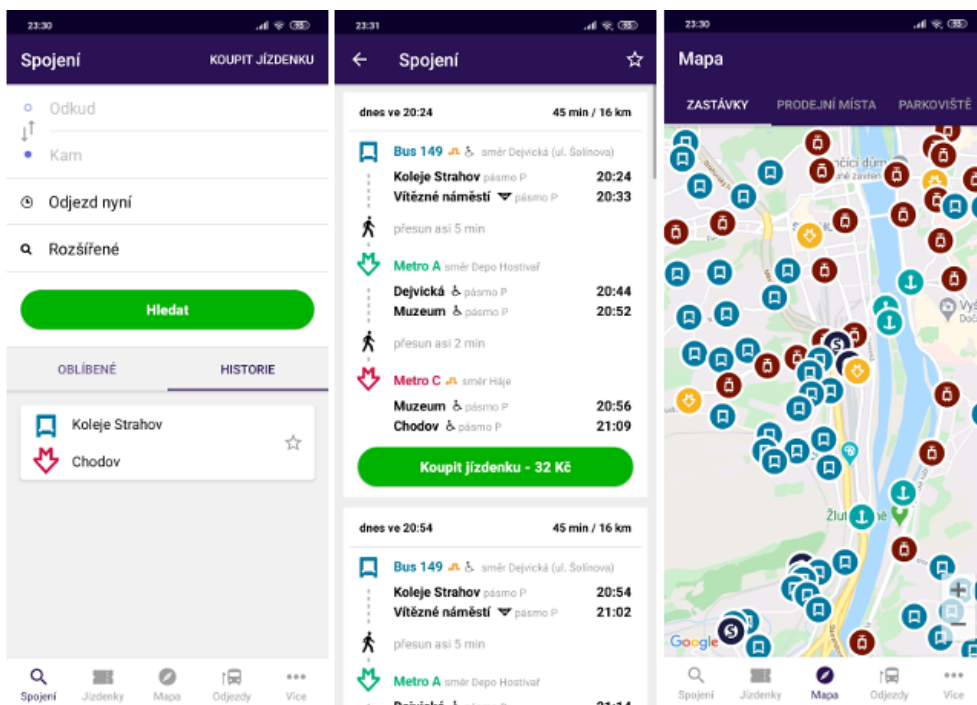
## 2.4 PID Lítačka

Aplikace PID Lítačka [4] je svou funkcionalitou o něco blíže aplikaci, jejíž realizací se zabývá tato práce. Jako taková představuje komplexní informační systém pro cestující, který nenabízí pouze možnost vyhledávání spojů či odjezdů, ale který umožňuje rovněž zobrazení dalších informací jako jsou informace o omezeních a výlukách MHD či další aktuality.

Aplikace umožňuje vyhledávání v jízdních řádech ve stejném rozsahu, jako tomu bylo u již zmíněných aplikací IDOS a Jízdní řády (Seznam.cz), tedy vyhledání spojů a odjezdů s možností volby pokročilých parametrů vyhledávání.

Rovněž je přítomno ukládání historie dotazů, jakožto i možnost uložení oblíbených položek.

Dále pak ani v případě této aplikace není přítomna možnost vyhledávání odjezdů a spojů bez připojení k internetu.



Obrázek 2.4: Záběry obrazovky aplikace PID Lítačka

Z hlediska uživatelského rozhraní autora práce zarazila absence stavového hlášení při kliknutí na hvězdičku u daného dotazu (obrázek 2.4 uprostřed), jejímž účelem je přidat dotaz k oblíbeným, které by indikovalo, že došlo k uložení dotazu do oblíbených položek. Po kliknutí sice dojde ke grafické změně hvězdičky, díky čemuž uživatel ví, že systém něco udělal, chybí ovšem už informace o tom, co vlastně udělal.

Autorovi práce pak přijde poněkud nešťastné řešení výběru objektů v mapě (obrázek 2.4 vpravo), kdy v situaci, kdy je vícero objektů velmi blízko sebe, či se dokonce překrývají, je vybrán objekt, který je nejbližší k bodu dotyku, přičemž se ovšem vzhledem k celkové blízkosti objektů nemusí nutně jednat o objekt, na který chtěl uživatel skutečně kliknout, a uživatel je tak nucen přiblížit mapu, aby kýžený objekt mohl bez problému vybrat.

Mnohem logičtějším řešením by dle autora bylo v případě, kdy nelze s jistotou určit, na který z objektů v okolí bodu dotyku chtěl uživatel kliknout, zobrazit možnost výběru z množiny objektů, které se v blízkosti bodu dotyku nachází.

### 2.5 DPMBinfo

Aplikace DPMBinfo [5] brněnského dopravního podniku představuje aplikaci, která je svou funkcionalitou nejbližší té, jejíž realizací se tato práce zabývá.

Stejně jako ostatní zmíněné aplikace umožňuje vyhledání spojení, místo klasického vyhledávání odjezdů však obsahuje pouze možnost vyhledat aktuální odjezdy z dané zastávky, která uživateli umožní rychle zjistit jaké spoje z dané zastávky v nejbližší době odjíždí a to včetně započítaného aktuálního zpoždění spojů. Na rozdíl od ostatních aplikací ale chybí možnost hledat odjezdy ze zastávky od uživatelem definovaného času a uživatel je tak nucen pro tuto možnost využít vyhledávání spojení, což může mírně nepříjemné pro uživatele, kteří již disponují určitou znalostí o linkách jedoucích na dané zastávce, a pro které je tudíž zadávání cílové zastávky při vyhledávání odjezdů nadbytečné.

Vyhledávání spojení je navíc v případě této aplikace řešeno poněkud nešťastně, kdy systém sice umožní uživateli zadat parametry dotazu v rámci mobilní aplikace, ale po stisknutí příslušného tlačítka pro vyhledání spojů však aplikace uživatele přesměruje na webové stránky IDOSu, který sice zobrazí spojení vyhledaná dle zadaných parametrů, ale v aplikaci webového prohlížeče a nikoliv v aplikaci samotné. Uživatel je tak případně nucen při používání aplikace zbytečně přepínat mezi aplikací a webovým prohlížečem.

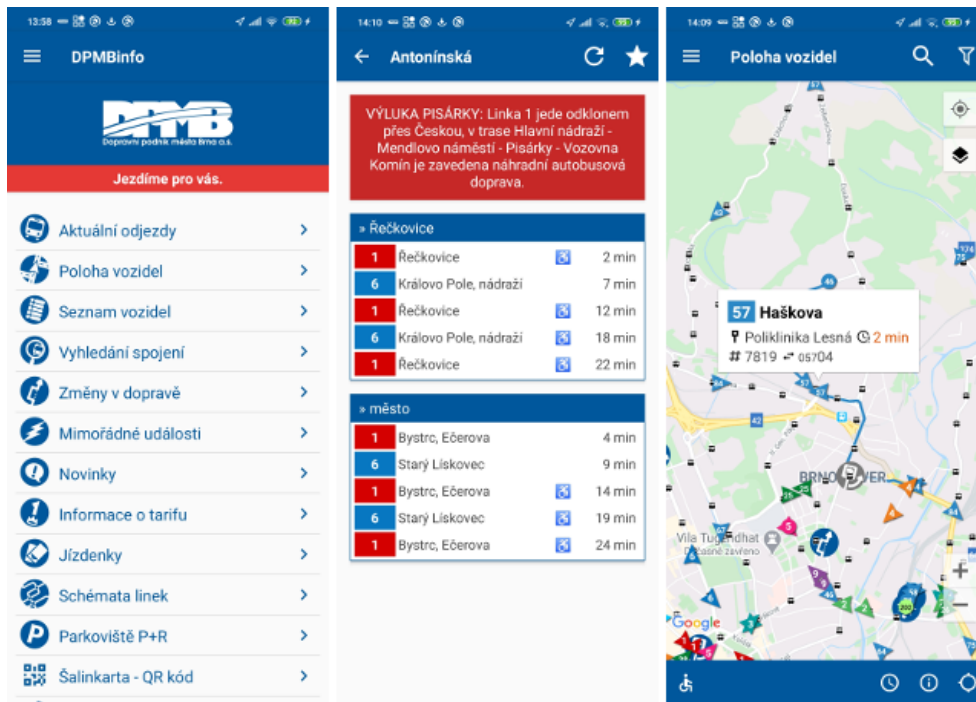
Integrací systému IDOS v této podobě rovněž odpadá jakákoliv možnost pro offline vyhledávání v jízdních řádech.

Aplikace rovněž disponuje mapou, ve které je mimo zastávek zobrazena i aktuální poloha jednotlivých vozů, u nichž je dále možné zobrazit jejich případné zpoždění, což je možné vidět na obrázku 2.5 vpravo.



Aplikace dále nabízí možnost zobrazení aktuálních změn v dopravě či mimořádných událostí.

Jako v případě aplikace PID Lítačka pak autorovi práce i v tomto případě chybí možnost výběru objektů po kliknutí na skupinu objektů v mapě, které jsou velmi blízko, či se dokonce překrývají.



Obrázek 2.5: Záběry obrazovky aplikace DPMBinfo

## 2.6 DPMCB

Další ze zkoumaných aplikací je aplikace českobudějovického dopravního podniku DPMCB. [6]

Tato aplikace, mimo běžné vyhledávání spojení a zobrazení aktuálních informací o dopravě, nabízí rovněž možnost zakoupení sms jízdenky a po přihlášení uživatele rovněž správu časového jízdného.

Poněkud zvláště je v případě této aplikace řešena hlavní obrazovka, na které je jediným interaktivním prvkem tlačítko pro zobrazení menu, přičemž by na ní autor práce očekával v ideálním případě uživateli nejvyužívanější funkcionalitu, tedy s nejvyšší pravděpodobností vyhledávání spojení.

Poněkud uživatelsky nepřívětivé je rovněž našeptávání zastávek ve formuláři pro vyhledávání spojení, který započne našeptávat hodnoty až po zadání třech a více písmen.

### 2.7 Moje PMDP

Dále bude představena aplikace Plzeňských městských dopravních podniků a.s. Moje PMDP [7], která nabízí srovnatelnou funkcionalitu jako již zmíněné aplikace PID Lítačka či DPMCB.

Jako jediná ze zkoumaných aplikací (s výjimkou aplikace JRm), nabízí Moje PMDP rovněž možnost zobrazení jízdních řádů jednotlivých zastávek bez připojení k internetu. Ani v tomto případě ale není umožněno offline vyhledávání odjezdů či spojení.

I v případě této aplikace pak není přítomna možnost výběru objektů po kliknutí na skupinu blízkých objektů v mapě, která již byla popsána u předcházejících rozborů aplikací PID Lítačka a DPMBinfo.

### 2.8 Moje DPO

Moje DPO [8] představuje aplikaci Dopravního podniku Ostrava, která je svou funkcionalitou velmi podobná již zmíněným aplikacím jiných dopravních podniků, jako Moje PMDP či PID Lítačka.

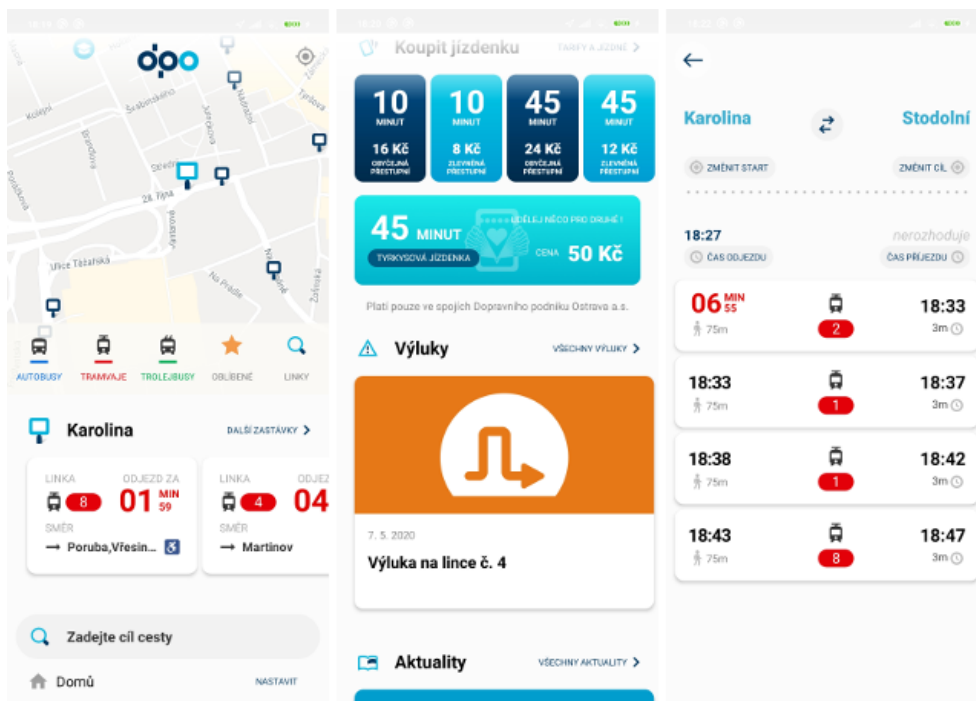
Jako jediná však pro využití svých funkcionalit vyžaduje registraci uživatele a to pro veškeré funkcionality aplikace bez ohledu na to, zda-li je přihlášení uživatele pro danou funkcionalitu skutečně zapotřebí (např. zakoupení jízdenky) či nikoliv.

Aplikace rovněž dle autorova názoru disponuje nejméně intuitivním uživatelským rozhraním v porovnání s ostatními aplikacemi zmíněnými v této kapitole (obrázek 2.6).

Příkladem může být vyhledávání spojů, které může uživatel provést přímo z hlavní obrazovky. Výběr počáteční a cílové stanice je ovšem zcela odlišný a ne zcela intuitivní. Zatímco způsob výběru cílové stanice je na první pohled zřejmý, výběr počáteční již nikoli. Při spuštění aplikace je již vybrána určitá zastávka a jediný způsob jak ji změnit v rámci hlavní obrazovky je výběr zastávky v zobrazené mapě. Tento způsob zadání ovšem předpokládá od uživatele znalost polohy této zastávky na mapě.

Další možností jak změnit počáteční zastávku je nejprve zadat cílovou zastávku, po čemž je uživatel přesměrován na obrazovku se seznamem výsledků dotazu, které předchází klasický formulář pro vyhledávání spojů, jaký je možné nalézt u většiny již zmíněných aplikací podobného typu, ve kterém je následně možné vybrat počáteční stanici stejným způsobem jako stanici cílovou. I tento postup je však dle autora pro koncového uživatele zcela neintuitivní.

Dalším z autorem identifikovaných problémů uživatelského rozhraní této aplikace je pak nevhodně zvolená barva oznamovací lišty respektive barva textu lišty, kdy jsou údaje uvedené na oznamovací liště telefonu v rámci spuštěné aplikace zcela nečitelné. Tuto skutečnost je možné pozorovat na příložených snímcích obrazovky na obrázku 2.6.



Obrázek 2.6: Záběry obrazovky aplikace Moje DPO

## 2.9 Závěr

V této kapitole bylo prozkoumáno několik aplikací, jež poskytují funkcionalitu podobnou té, kterou se zabývá tato práce.

V rámci tohoto průzkumu bylo identifikováno několik nedostatků v rámci uživatelského rozhraní jednotlivých aplikací a zároveň byly vyzdvíženy zajímavé funkcionality poskytované těmito aplikacemi jako například ukládání historie dotazů či možnost definování pokročilých parametrů při vyhledávání spojení a odjezdů.

Průzkum rovněž ukázal, že u většiny těchto aplikací chybí jakákoliv možnost vyhledávání v jízdních řádech bez připojení k internetu. Výjimku v tomto případě představuje aplikace JRm, která umožňuje plnohodnotné offline vyhledávání, a aplikace Moje PMDP, která nabízí možnost bez připojení k internetu zobrazit alespoň jízdní řády jednotlivých zastávek.



---

# Vývoj aplikací pro mobilní platformy

Než přejdeme k praktické části této práce, je vhodné se zamyslet, v jaké formě bude kýžená funkcionalita poskytnuta koncovému uživateli.

Jak již bylo avizováno dříve, je aplikace, jejíž implementací se tato práce zabývá, určena pro mobilní zařízení, konkrétněji zařízení s operačním systémem Android.

Existuje ovšem více možností jak pro tato zařízení požadovanou funkcionalitu zprostředkovat, přičemž jako nejčastější je možné považovat následující tři možnosti:

- responzivní webová aplikace,
- progresivní webová aplikace,
- nativní aplikace.

Z toho důvodu se autor této práce rozhodl prozkoumat výhody a nevýhody jednotlivých možností.

## 3.1 Responzivní webová aplikace

Responzivní webová aplikace představuje druh webové aplikace, která se odlišuje svou schopností adaptivně měnit svoje rozložení v závislosti na velikosti obrazovky cílového zařízení s pomocí speciálního CSS kódu nazývaného CSS media queries [9].

Responzivní webové aplikace tímto způsobem upravují svoje rozložení tak, aby jejich funkcionalita mohla být uživatelem jednoduše využita na odlišně velkých zařízeních.

### 3. VÝVOJ APLIKACÍ PRO MOBILNÍ PLATFORMY

---

Běžnou praxí je v případě takovýchto aplikací rozdělit cílová zařízení do několika kategorií v závislosti na velikosti jejich obrazovky a pro tyto kategorie následně navrhnout vhodné rozložení prvků aplikace. Typicky se jedná o kategorie

- desktop,
- tablet,
- mobilní zařízení.

V dnešní době je pak při návrhu rozložení aplikace či webu díky vzrůstající oblíbenosti mobilních zařízení upřednostňován princip *mobile-first*, kdy je nejprve navrženo rozložení pro malá mobilní zařízení, z něhož následně vychází rozložení další. To ilustruje následující obrázek 3.1.



Obrázek 3.1: Ilustrace *mobile-first* principu [10]

Pro využití funkcionality responzivní webové aplikace pak stačí téměř libovolný webový prohlížeč, což ve spojení se schopností adaptivně měnit rozložení, umožňuje spuštění aplikace na širokém spektru koncových zařízení.

Nevýhodou tohoto typu aplikací je ovšem nutnost kvalitního připojení k internetu, jelikož tento druh aplikací neposkytuje jakoukoli offline funkcionalitu. Navíc i v případě, že mobilní zařízení uživatele disponuje konektivitou k internetu, ale horší kvality, dochází k ovlivnění běhu aplikace v podobě delší odezvy, která ve výsledku může koncového uživatele odradit od využití aplikace.

Skutečnost, že veškerá funkcionalita aplikace je zprostředkována online, pak může rovněž vést k přenosu většího množství dat po síti, což může představovat problém především v situaci, kdy předpokládáme, že koncový uživatel využívá mobilní připojení k internetu, a to zvláště v rámci České republiky, kde jsou datové tarify pro mobilní zařízení stále poněkud dražší.

## 3.2 Progresivní webová aplikace

Progresivní webové aplikace neboli PWA představují relativně nový přístup k vývoji aplikací spustitelných na mobilních zařízeních.

Přesněji byl termín progresivní webová aplikace použit poprvé společností Google v roce 2015 k popsání aplikací využívajících výhod nových funkcí moderních webových prohlížečů, jako jsou service workers a webové aplikační manifesty, které umožňují uživatelům "upgradovat" webové aplikace o funkcionality z nativního operačního systému [11].

Aby mohla být aplikace nazvána progresivní webovou aplikací, měla by v sobě zahrnovat následující principy: [12]

- **Objektivitné:** aplikace je rozpoznatelná pomocí vyhledávacích enginů.
- **Instalovatelné:** aplikace je dostupná z home obrazovky zařízení.
- **Odkazovatelná:** je možné aplikaci sdílet pomocí URL.
- **Nezávislé na konektivitě:** funguje i offline či s nekvalitním připojením.
- **Progresivní:** aplikace je možné použít s libovolným prohlížečem, ačkoli u starších může její funkcionality být omezena.
- **Znovuzapojení uživatele:** je schopna notifikovat uživatele, kdykoli je k dispozici nový obsah.
- **Responzivní:** je možné ji použít na libovolném zařízení disponujícím obrazovkou a prohlížečem.
- **Bezpečné:** spojení mezi aplikací a zařízením je zabezpečeno tak, aby byl zamezen přístup třetích stran k citlivým informacím.
- **App-like:** uživatel má pocit, jako by využíval nativní aplikaci.

Z těchto principů již můžeme vyvodit některé výhody tohoto druhu aplikací.

Jednou z velkých výhod je, že PWA jsou instalovatelné a z pohledu uživatele se prakticky neliší od nativních aplikací. Není potřeba složitě umisťovat odkaz na webovou stránku na domovskou obrazovku zařízení, či hledat aplikaci v App store či Google play. Možnost instalace je uživateli jednoduše nabídnuta při navštívení webové stránky aplikace.

Rovněž oproti aplikaci zobrazené v prohlížeči chybí u PWA adresový řádek či další UI prohlížeče, což znamená, že aplikace může využít celý prostor obrazovky a při dodržení principu app-like, se tak na první pohled nedá rozlišit od aplikace nativní.

Uživatelské rozhraní a infrastruktura je kešována lokálně pomocí service worker, každé další načtení aplikace tedy načte jen potřebná data místo toho,

aby musela načíst všechno znovu. [11] To má za následek, že má aplikace z pohledu uživatele mnohem lepší odezvu než klasická webová aplikace, a zároveň je sníženo množství dat přenesených po síti.

PWA ovšem nemají plný přístup k hardware zařízení, jako tomu je u aplikací nativních, a vývojáři se tak musí rozloučit s možností využití některých funkcí zařízení.

Je například limitováno množství dat, které aplikace může kešovat pro offline využití a jež se liší v závislosti na použitém prohlížeči. [13]

V závislosti na prohlížeči se pak rovněž může lišit výčet funkcionalit, ke kterým má aplikace přístup. Příkladem může být dle [14] přístup k bluetooth zařízení, který není podporován v prohlížeči Firefox.

Problémem v kontextu této práce je pak integrace technologie progresivních webových aplikací s mapovými technologiemi společnosti Esri, které využívá Magistrát města Jihlavy a na nichž je postavena webová verze online mapy dopravy.

Vzhledem k webové povaze PWA by totiž bylo nutné využít ArcGIS API pro javascript této společnosti, v němž je sice možné využít offline mapové podklady, jak autor této práce zamýšlí, ale pouze s využitím offline knihoven, které již nejsou společností Esri aktivně vyvíjeny a udržovány, a z jejíž strany je doporučeno využít ArcGIS Runtime SDK pro příslušnou platformu. [15]

### 3.3 Nativní mobilní aplikace

Poslední možností, která zde bude představena, představují nativní aplikace.

Nativní aplikace je software, který je vyvinut pro specifickou platformu či zařízení s využitím technologií spjatých s danou platformou. Ve světě mobilních aplikací se v dnešní době jedná především o operační systém Android, na němž je pro vývoj aplikací využívána Java, a iOS, na kterém implementace realizována v jazyce Swift či Objective-C. [16]

Toto cílení na určitou platformu či zařízení umožňuje nativním aplikacím využívat veškerou funkcionalitu, kterou daná platforma nabízí a jež není možné využít v případě PWA či webové aplikace (například geofencing). [17]

Nativní aplikace jsou na tom rovněž lépe z výkonnostního hlediska a to díky skutečnosti, že běží přímo v rámci operačního systému zařízení a mohou tak přistupovat k hardwaru zařízení mnohem efektivněji, na rozdíl od PWA, které běží uvnitř prohlížeče, čímž může být způsobena horší odezva aplikace či větší spotřeba baterie. [18]

I nativní aplikace ovšem disponují určitými nevýhodami, kdy hlavní z nich představuje nutnost vývoje samostatné aplikace pro každou platformu pro niž chce vývojář svoji aplikaci poskytnout. Při potřebě vývoje pro více platform tak může dojít k výraznému zvýšení nákladů na vývoj oproti PWA, které jsou multiplatformní.



### 3.4 Závěr

Pro aplikaci, jejíž realizací se tato práce zabývá, připadají z dříve zmíněných možností pro poskytování funkcionality na mobilních zařízeních pouze poslední dvě možnosti, tedy progresivní webová či nativní aplikace, a to díky úmyslu v rámci aplikace poskytovat i offline funkcionality.

Ze zbylých dvou možností se jako slibný kandidát jeví PWA díky svojí multiplatformní povaze, díky níž by bylo možné funkcionality aplikace poskytnout většímu množství uživatelů, avšak kvůli existujícímu omezení na množství kešovaných dat a problematické integraci s technologiemi společnosti Esri při využití offline mapových podkladů se tato možnost jeví jako nevyhovující.

Při realizaci aplikace, jež je předmětem této práce, tak bude využita poslední možnost, tedy nativní aplikace.



---

# Analýza

Předtím než je možné přistoupit k dalším fázím vývojového procesu, je nejdříve nutné stanovit, jakou funkcionalitu má výsledná aplikace poskytovat, jakož i stanovit případné další vlastnosti, kterými má aplikace disponovat.

Toho je docíleno pomocí systematického sběru požadavků od zadavatele a případně rovněž od budoucích uživatelů vyvíjeného systému.

## 4.1 Rozbor požadavků

Určení požadavků na software představuje základní krok ve vývoji softwarového produktu. Jako takové požadavky na systém vymezují základní představu o produktu, definují, jakou funkcionalitu od výsledného software zadavatel očekává a jakými dalšími vlastnostmi má disponovat.

V prvním případě se jedná o takzvané funkční požadavky, které, jak již bylo zmíněno výše, definují očekávanou funkcionalitu výsledného systému.

V druhém případě se pak jedná o požadavky nefunkční, které definují další požadované vlastnosti systému, jakými jsou například dostupnost, odezva či bezpečnost systému.

Dále pak specifikace požadavků může obsahovat další omezení kladená na systém daná zadavatelem, například co se týče použitých technologií.

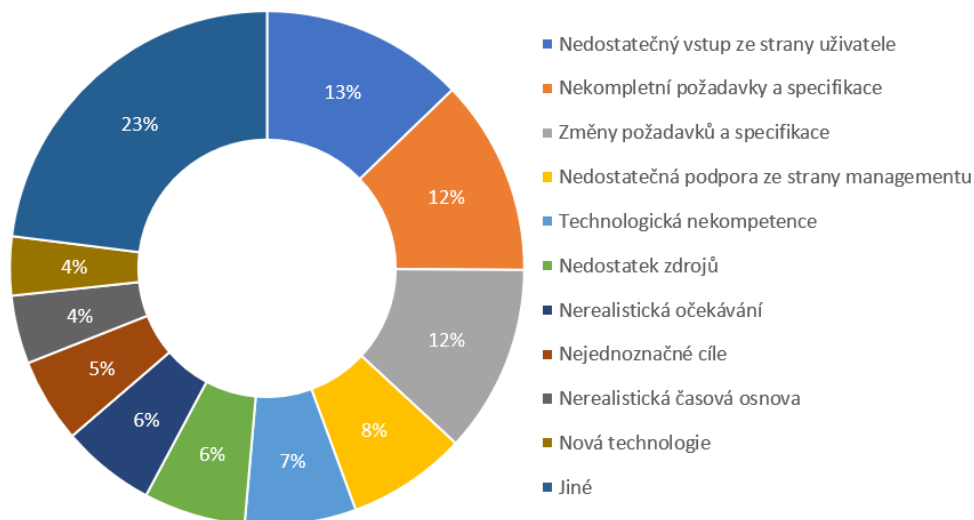
Jak již bylo řečeno, specifikace požadavků nebo také SRS (z anglického Software Requirments Specification) je prvotním krokem ve vývoji softwarového produktu a na jejím základě jsou pak založeny další části vývoje. Z toho důvodu je nutné této fázi vývoje věnovat zvýšenou pozornost, neboť nedostatky či chyby ve specifikaci mohou negativně ovlivnit průběh projektu či dokonce vést k jeho selhání.

Jak je možné vidět na následujících grafech, patří dle The Standish Group [19] nedostatky ve specifikaci požadavků dokonce k jednomu z hlavních příčin problémů u softwarového projektu, či dokonce jeho celkového selhání.

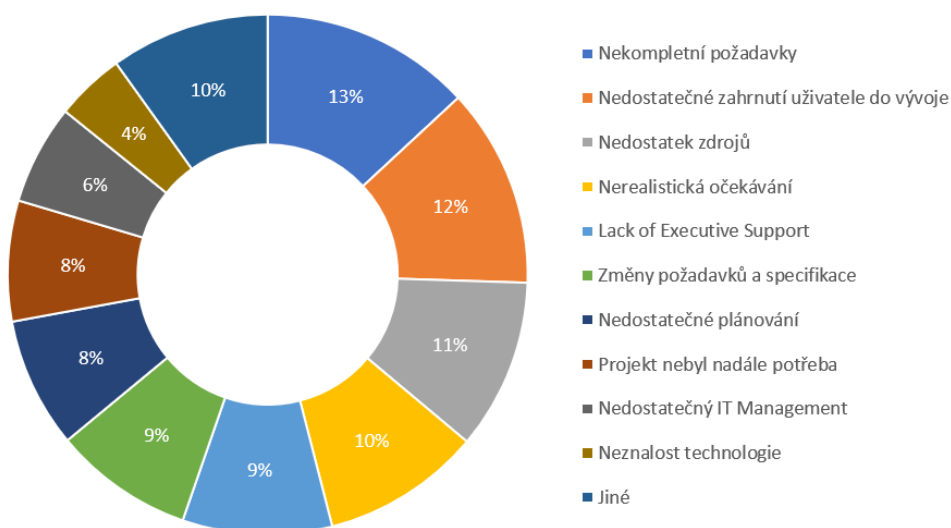
#### 4. ANALÝZA

Problémy jsou podle [19] způsobeny nekompletní specifikací až ve 12 % případů, selhání pak dokonce ve 13 %.

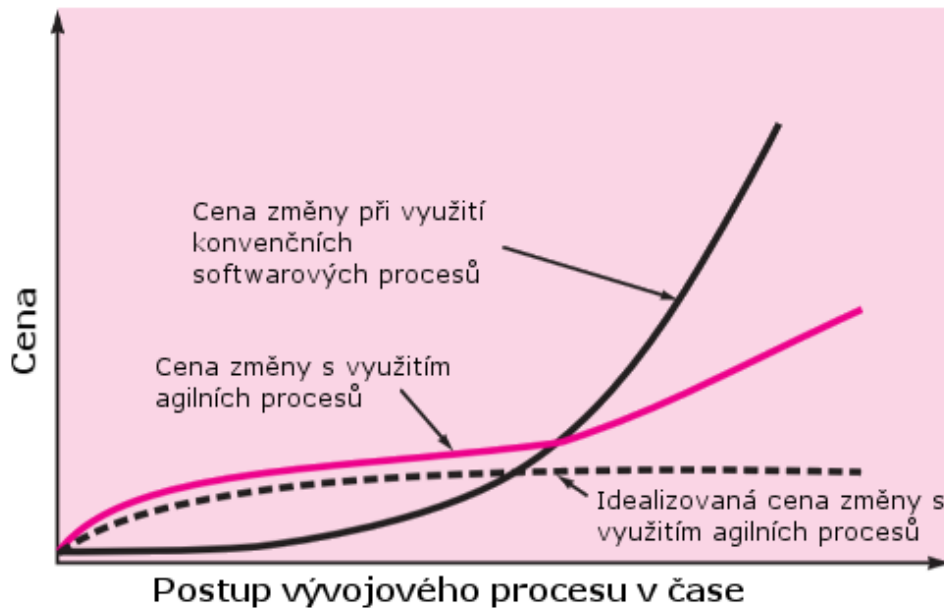
Z toho důvodu je nutné této fázi vývoje věnovat zvýšenou pozornost, jelikož v této fázi je nejjednodušší a zároveň i nejlevnější odstranit případné chyby, neboť jak projekt postupuje, je odstranění chyb stále náročnější, což je možné pozorovat i na grafu vyobrazeném na obrázku 4.3.



Obrázek 4.1: Důvody vzniku problémů u SW projektu



Obrázek 4.2: Důvody selhání SW projektu



Obrázek 4.3: Vývoj ceny opravení chyby v rámci vývojového procesu [20]

#### 4.1.1 Funkční požadavky

- **Zobrazení jízdních řádů:** Aplikace umožní uživateli vyhledávat odjezdy a spojení v rámci jízdních řádů a to bez připojení k internetu.
- **Uložení oblíbených dotazů:** Uživateli bude umožněno si v rámci aplikace uložit jím často používané dotazy na jízdní řády.
- **Integrace online mapy dopravy:** Aplikace bude implementovat funkcionalitu online mapy dopravy, kterou vyvinul a provozuje Magistrát města Jihlavy, a s pomocí které může uživatel sledovat aktuální polohu jednotlivých vozidel městské hromadné dopravy a jejich zpoždění, jakož i zobrazovat trasy jednotlivých linek MHD.
- **Integrace se systémem chytrých zastávek:** Aplikace bude obsahovat možnost zobrazení aktuálních odjezdů spojů, jež na vybraných zastávkách zobrazují takzvané chytré zastávky.
- **Propojení s webovým portálem DMPJ:** V aplikaci si uživatel bude moci zobrazit aktuality týkající se městské hromadné dopravy zveřejněné na webovém portálu DPMJ.
- **Notifikace na nové aktuality:** Aplikace bude uživatele notifikovat o nově dostupných aktualitách.

- **Notifikace na dostupnost nových offline podkladů:** Aplikace upozorní uživatele na dostupnost novější verze offline podkladů, tedy jízdních řádů a mapových podkladů.
- **Správa offline podkladů:** Výsledné řešení bude implementovat rozhraní pomocí něž bude správci umožněna správa offline podkladů (jízdni řády, mapové podklady, definice svátků), a určení datumu, kdy tyto podklady vstupují v platnost.
- **Vícejazyčnost:** Výsledná aplikace umožní zobrazení obsahu ve více jazycích tak, aby ji mohli využívat i turisté, či například návštěvníci každoročního mezinárodního dokumentárního festivalu, kteří mohou v posledních letech v průběhu festivalu využívat MHD zdarma a jimiž je z tohoto důvodu hojně využívána.

### 4.1.2 Nefunkční požadavky

- **Bezpečnost:** V části systému sloužící pro správu offline podkladů budou uložena citlivá data správců jednotlivých podkladů v podobě jejich emailových adres, které by mohly být zneužity (např. pro zasílání spamu), pokud by k nim útočník získal přístup. Systém by tedy měl být implementován s ohledem na bezpečnost v něm uložených dat.
- **Spolehlivost a dostupnost:** Správná funkcionality klientské části je závislá na dostupnosti offline podkladů, které jsou poskytovány serverovou částí systému. Pro zajištění spolehlivé funkcionality klientské části aplikace je tedy potřeba se v maximální míře vyhnout chybám v implementaci serverové části.

## 4.2 Rozdělení systému

Jak je patrné ze získaných požadavků, výsledný systém bude tvořen dvěma oddělenými částmi, které disponují odlišnými rozhraními a funkcionalitou. Konkrétně se jedná o část systému sloužící pro správu offline podkladů a následně o vlastní klientskou část systému v podobě mobilní aplikace určenou pro cestující.

### Správcovská část systému

Tato část systému, jak již bylo avizováno dříve, umožňuje správu jednotlivých offline podkladů v podobě definic jízdních řádů, svátků a mapových podkladů, které jsou touto částí systému následně poskytovány části klientské, která je využívá k realizaci své funkcionality.

Správcovská část systému rovněž umožňuje spravovat názvy linek, které bohužel nejsou dostupné v exportu jízdních řádů a je tedy nutné zadat v systému ručně mapování mezi identifikátorem linky a jejím názvem.

Dále tato část systému umožňuje spravovat pěší spojení, která slouží správci jízdních řádů k definování možnosti přechodu cestujícího mezi danými dvěma zastávkami. Takto definovaná spojení jsou pak v klientské části využívána při vyhledávání spojení.

Jako taková operuje správcovská část systému z velké části nad množinou dat, které se nemění příliš často (státní svátky, pěší spojení, názvy linek) s výjimkou jízdních řádů, u kterých může docházet ke změnám s vyšší frekvencí než u zbylých dat spravovaných touto částí systému.

V porovnání s klientskou částí aplikace využívanou cestujícími tak bude správcovská část využívána jen velmi málo.

### Klientská část

Klientská část systému je tvořena vlastní mobilní aplikací a jako taková poskytuje veškerou funkcionalitu zajímavou pro koncového uživatele v podobě cestujícího. Jako taková umožňuje především vyhledávání odjezdů a spojení v rámci jízdních řádů, zobrazení aktuálních odjezdů z dané zastávky, zobrazení online mapy dopravy a informací o aktuálních změnách v dopravě.

Ke zprostředkování této funkcionality pak využívá offline podklady, které jsou poskytnuty správcovskou částí systému.

## 4.3 Případy Užití

Případy užití neboli anglicky use cases představují další techniku zachycení funkcionality informačního systému. Jako takové zachycují tuto funkcionalitu z pohledu uživatele systému prostřednictvím strukturovaných scénářů, které detailně popisují, jakým způsobem je systém uživatelem používán za účelem dosažení určitého cíle.

Struktura jednotlivých scénářů příslušných případů užití pak v rámci této práce vychází z publikace pana Alistair Cockburna – Writing Effective Use Cases [21].

V rámci této kapitoly uvádí autor pouze stručný výčet jednotlivých případů užití, přičemž podrobně rozepsané případy užití je možné nalézt v přílohách této práce.

### 4.3.1 Správcovská část systému

V rámci této části aplikace byly celkově identifikovány následující tři typy uživatelů:

- správce systému,

- správce jízdnicích řádů,
- správce mapových podkladů.

Správci systému je umožněno provádět v systému veškeré dostupné úkony, přičemž oproti ostatním typům uživatelů disponuje především oprávněním pro manipulaci s uživateli.

Správčům jízdnicích řádů a mapových podkladů je pak umožněna manipulace s odpovídajícími offline podklady v systému.

#### **Případy užití společné všem uživatelům**

- **U1.0.1** Přihlášení
- **U1.0.2** Změna vlastního hesla

#### **Případy užití správce systému**

- **U1.1.1** Přidání nového uživatele
- **U1.1.2** Změna údajů uživatele
- **U1.1.3** Změna přístupových práv uživatele
- **U1.1.4** Odstranění uživatele

#### **Případy užití správce jízdnicích řádů**

- **U1.2.1** Nahraní nové verze jízdnicích řádů
- **U1.2.2** Odstranění verze jízdnicích řádů
- **U1.2.3** Úprava data vstoupení v platnost dané verze jízdnicích řádů
- **U1.2.4** Nahraní nové verze definice svátků
- **U1.2.5** Odstranění verze definice svátků
- **U1.2.6** Úprava data vstoupení v platnost dané verze definice svátků
- **U1.2.7** Přidání jména linky
- **U1.2.8** Odstranění jména linky
- **U1.2.9** Úprava jména linky
- **U1.2.10** Přidání pěšího spojení
- **U1.2.11** Úprava pěšího spojení
- **U1.2.12** Přidání pěšího spojení



#### **Případy užití správce mapových podkladů**

- **U1.3.1** Nahrání nové verze mapových podkladů
- **U1.3.2** Odstranění verze mapových podkladů
- **U1.3.3** Úprava data vstoupení v platnost dané verze mapových podkladů

#### **4.3.2 Klientská část systému**

Klientská část představuje vlastní mobilní aplikaci, kterou lze z hlediska její funkcionality rozdělit do tří logických celků, jimiž jsou:

- část týkající se jízdních řádů,
- část týkající se online mapy dopravy,
- a část týkající se aktualit o změnách v dopravě.

#### **Případy užití související s jízdními řády**

- **U2.0.1** Zobrazení aktuálních odjezdů
- **U2.0.2** Vyhledání odjezdů
- **U2.0.3** Vyhledání spojení
- **U2.0.4** Zobrazení oblíbených položek
- **U2.0.5** Přidání oblíbené položky
- **U2.0.6** Odstranění oblíbené položky
- **U2.0.7** Aktualizace offline podkladů pro jízdní řády
- **U2.0.8** Aktualizace offline definice svátků
- **U2.0.9** Zobrazení oblíbeného dotazu

#### **Případy užití související s online mapou dopravy**

- **U2.1.1** Zobrazení detailu objektu na mapě
- **U2.1.2** Zobrazení legendy/informací o mapě
- **U2.1.3** Filtrování linek
- **U2.1.4** Nastavení map do online režimu
- **U2.1.5** Nastavení map do offline režimu
- **U2.1.6** Aktualizace offline mapových podkladů

### Případy užití související s aktualitami

- **U2.2.1** Zobrazení aktuality

## 4.4 Závěr

Provedením analýzy byla ustanovena základní představa o funkcionalitě a struktuře vyvíjeného systému.

Ze získaných informací je patrné, že architektonickým základem systému bude model klient-server, kdy serverová část systému slouží k uchování offline podkladů, jejichž správa je umožněna rozhraním správcovské části systému.

Tyto offline podklady jsou pak následně poskytovány části klientské, která je využívá ke zprostředkování své funkcionality.

---

## Výběr technologií

Z předchozích kapitol už je zřejmé, jakou funkcionalitu bude výsledný systém poskytovat a jakým způsobem bude tato funkcionalita využívána. Je tedy možné přistoupit k rozhodování o technické podobě systému.

Jak již bylo uvedeno, bude výsledný systém rozdělen na dvě oddělené části – správcovskou a klientskou, přičemž u klientské již bylo rozhodnuto o její implementaci v podobě nativní mobilní aplikace. Tím je rovněž u této části systému prakticky rozhodnuto o použité technologii a to vzhledem ke skutečnosti, že vývoj nativních aplikací pro operační systém Android je možné realizovat pouze s využitím programovacího jazyka Java či případně jazyka Kotlin.

Autor práce se pak rozhodl v případě klientské části systému využít při implementaci jazyk Java a to především vzhledem k tomu, že již disponuje určitou znalostí tohoto jazyku, na rozdíl od jazyku Kotlin, se kterým se doposud nesešel.

Platforma a technologie klientské části systému jsou tedy již dané, zbývá ovšem rozhodnout, jakou platformu a technologie zvolit pro realizaci správcovské části.

### 5.1 Platforma rozhraní správcovské části

Z hlediska platformy rozhraní správcovské části systému existují z autorova pohledu dvě přijatelné možnosti pro jeho realizaci. Jsou jimi

- tlustý desktopový klient a
- tenký webový klient.

Možnost realizace správcovského rozhraní jako mobilní aplikace byla předem zavržena vzhledem k problémům, které by využití takové platformy přineslo, přičemž za hlavní z těchto problémů lze považovat vázanost takového

řešení na daný mobilní operační systém, vzhledem ke skutečnosti, že nelze předpokládat, že každý ze správců bude disponovat mobilním zařízením s operačním systémem, pro který by případně správcovské rozhraní v podobě mobilní aplikace bylo implementováno.

O využití příslušné platformy, která bude využita pro realizaci správcovského rozhraní pak autor práce rozhodl v závislosti na následujících kritériích.

**Dostupnost a pohodlí:** Důležitým faktorem ke zvážení je složitost zprovoznění systému koncovým uživatelem, jeho dostupnost na různých zařízeních, jakož i míra pohodlí s jakou je možné vykonávat jednotlivé úkony v rámci systému na dané platformě.

**Šíření aktualizací:** Toto kritérium slouží ke zhodnocení, jak složité je šíření nových verzí systému k jednotlivým uživatelům.

**Znalosti vývojáře:** V rámci tohoto kritéria budou zhodnoceny dosavadní zkušenosti autora s vývojem pro danou platformu a rovněž znalost technologií používaných při vývoji na dané platformě.

### Thustý desktopový klient

V případě tlustého desktopového klienta by výsledné správcovské rozhraní bylo tvořeno desktopovou aplikací instalovanou přímo na zařízeních jednotlivých správců offline podkladů, přičemž aplikační logika by byla z větší části obsažena přímo v rámci vlastního rozhraní a serverová část systému by sloužila pouze uložení dat a jejich sdílení s částí klientskou.

Jako takové představuje tato možnost v době, kdy je ze strany vývojářů software stále častěji a častěji využíván koncept software jako služby (SaaS – Software as a Service), představuje spíše zpátečnické řešení.

Díky nutnosti instalace je pro koncového uživatele složitější systém na svém zařízení zprovoznit a v případě, že uživatel chce využívat systém na více zařízeních, je nucen provést instalaci na všech těchto zařízeních. S touto skutečností je pak rovněž spojeno problematické šíření aktualizací, kdy je zapotřebí zajistit jejich šíření ke všem uživatelům, kteří systém používají.

Z hlediska pohodlí uživatele pak velké obrazovky, kterými desktopová zařízení disponují, umožňují přehledné zobrazení velkého množství informací, díky čemuž mohou uživatelé využívat funkcionalitu systému mnohem efektivněji než například na zařízeních mobilních.

Zkušenosti autora práce s vývoje pro tuto platformu jsou omezená na zkušenosti z různých školních projektů a jako takové nejsou úplně rozsáhlé.

### Tenký webový klient

Tenký webový klient představuje na druhou stranu moderní řešení, které je díky využití webových technologií platformě nezávislé a jeho funkcionalitu lze

tak využít na libovolném zařízení s webovým prohlížečem, který podporuje technologie použité při implementaci webového klienta.

Na rozdíl od desktopového klienta tak webový klient nevyžaduje zvláštní instalaci a vzhledem ke skutečnosti, že je veškerý kód aplikace přítomen pouze na příslušném serveru, není zapotřebí zajišťovat distribuci aktualizací k jednotlivým uživatelům systému.

Jak již bylo zmíněno výše, je webový klient nezávislý na platformě, jeho funkcionalitu lze tak může uživatel využít kromě desktopu i na svém mobilním zařízení, ačkoliv menší obrazovka takového zařízení mu zamezí využívat aplikaci se stejnou efektivitou, s jakou je možné ji využívat na větším zobrazovacím zařízení desktopu.

Tento problém lze částečně eliminovat uplatněním responzivního designu popsaného dříve, kdy je přeskupením prvků aplikace možné dosáhnout větší přehlednosti i na malé obrazovce. Množství informací, které je možné najednou zobrazit, je ale i v takovém případě stále omezené velikostí obrazovky zařízení.

Znalosti autora týkající se vývoje pro tuto platformu jsou mnohem rozsáhlejší než v případě desktopových aplikací, vzhledem ke skutečnosti, že se autor práce zabýval tvorbou webových stránek i mimo svoje školní povinnosti.

### Závěr

Jak je patrné z předešlého rozboru obou možností pro implementaci správcovského rozhraní, nedisponuje tlustý desktopový klient v případě vyvíjeného systému žádnou významnou výhodou oproti tenkému webovému klientu a v případě autorem sledovaných kritérií za ním zaostává ve všech případech.

Z toho důvodu je tenký webový klient vnímán autorem jako jasná volba pro realizaci správcovského rozhraní vyvíjeného systému.

## 5.2 Technologie správcovské části

Nyní když je stanovena platforma správcovské části systému, je možné přistoupit k výběru technologií, které budou využity při její implementaci, přičemž budou zvažovány technologie, s nimiž se autor práce setkal v rámci školních povinností či praxe.

Konkrétně budou zvažovány následující technologie:

1. C#,
2. Java EE,
3. PHP a
4. Smalltalk.

Při hodnocení jednotlivých technologií pak budou zkoumány především tyto parametry:

- míra s jakou je autor obeznámen s danou technologií,
- výše nákladů na vývoj a provoz aplikace,
- množství dostupných knihoven a frameworků,
- kvalita a dostupnost dokumentace a
- rozšířenost v rámci IT komunity.

### C#

C# představuje programovací jazyk z dílen společnosti Microsoft, který byl poprvé představen široké veřejnosti v roce 2000 jako součást frameworku .NET téže společnosti. [22]

Jako takový ho lze využít pro implementaci na množství platforem počínaje desktopovými aplikacemi přes webové aplikace až po aplikace mobilní (Xamarin).

Dlouhodobě se jedná o jazyk, který je v rámci IT komunity poměrně oblíbený, což ukazuje i jeho umístění na prvních příčkách TIOBE (graf 5.1) a PYPL indexu (graf 5.2).

C# a s ním spojená technologie .NET rovněž disponuje velmi kvalitní dokumentací přímo od Microsoftu a i uživatelská základna těchto technologií je dle [23] poměrně obstojná, čímž by mělo být usnadněno hledání řešení problémů, které se mohou během vývoje objevit.

Nevýhodou je ovšem v případě této technologie horší znalost autora z hlediska tvorby webových aplikací s využitím této technologie.

Další nevýhodou jsou pak vyšší náklady na hosting oproti například PHP, u kterého lze pořídit hosting poměrně levno.

### Java EE

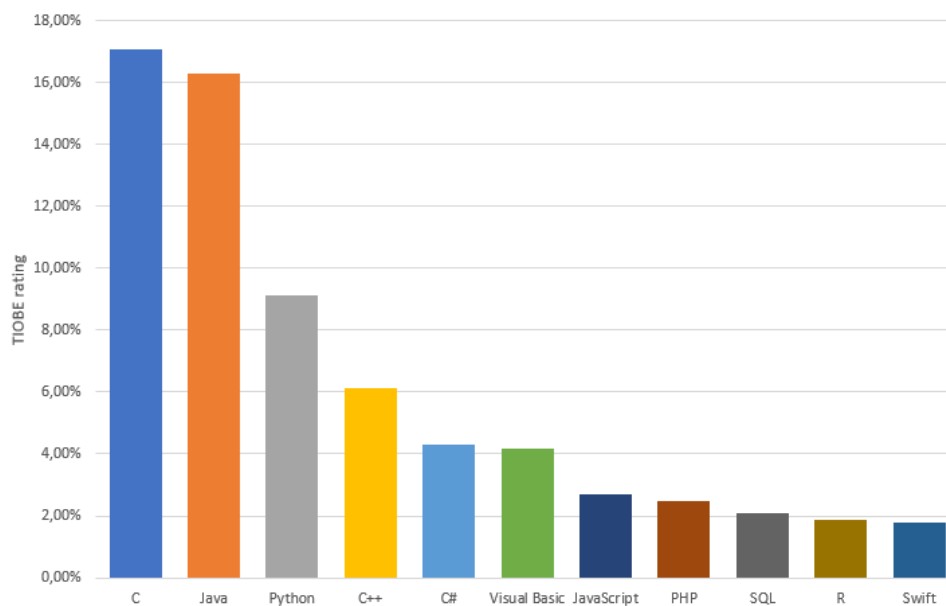
Java patří v dnešní době k jednomu z nejpobulárnějších programovacích jazyků, což dokládají její výsledky v rámci TIOBE a PYPL indexu, v nichž dosahuje hodnot 16,28% v případě TIOBE ratingu a 17,75% v rámci indexu PYPL.

Java EE, která je nadstavbou Java SE používanou pro vývoj webových aplikací, je pak dle [24] využívána u 3,5% webových aplikací, u nichž je známa použitá technologie. Toto číslo se zdá být poněkud zanedbatelné v porovnání s PHP, které je využito až u 78,4% webových aplikací [25], je ovšem nutné vzít také v potaz, že Java je využita především v případě aplikací, které vykazují mnohem vyšší traffic, než je tomu v případě aplikací využívajících PHP, a u nichž je tak kladen mnohem větší důraz na spolehlivost a škálovatelnost. [24]

Nevýhodou je ovšem u Java EE množství znalostí potřebných pro rychlý a kvalitní vývoj webových aplikací s využitím této technologie, kterými autor

této práce bohužel nedisponuje vzhledem ke skutečnosti, že se s touto technologií setkal spíše okrajově.

Další nevýhodou je pak vzhledem k poměru využití této technologie u webových aplikací výrazně menší komunita, což může mít za následek mnohem složitější dohledání řešení problémů, které mohou vyvstat v průběhu vývoje.



Obrázek 5.1: TIOBE rating top 10. Data dostupná na [26]

## PHP

Jak již bylo zmíněno dříve, je PHP využíváno jako hlavní server-side technologie až u 78,4% webových stránek a jako takové na webu zcela dominuje.

Ačkoliv značná část IT komunity chová k této technologii určitou zášť, jedná se dle obou indexů TIOBE i PYPL stále o jeden z nejpobulárnějších jazyků vůbec, kdy je v případě TIOBE momentálně umístěn na osmé příčce (graf 5.1) a v případě PYPL pak dokonce na příčce páté (graf 5.2). Je pravdou, že využití této technologie bylo v poslední době poněkud na ústupu na úkor např. jazyka Python, avšak se zlepšeními (hlavně z hlediska výkonu), které přinášejí novější verze 7.x, se stále jedná o poměrně solidní volbu.

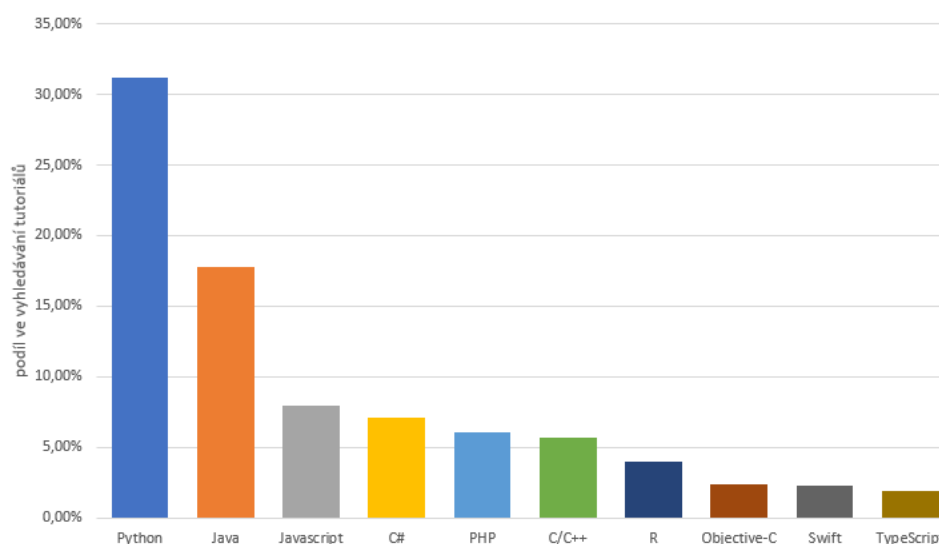
Díky širokému využití této technologie pak tato technologie disponuje rozsáhlou komunitou a není tak problémem najít množství návodů, rad či jiných materiálů, které mohou pomoci rychle řešit problémy, které vyvstanou v průběhu vývoje.

Velkou výhodou jsou pak v případě PHP zkušenosti autora s touto technologií, které výrazně převyšují ty u ostatních zmíněných technologií.

## 5. VÝBĚR TECHNOLOGIÍ

---

Výhodou je pak v rámci tohoto projektu rovněž možnost hostovat řešení implementované s pomocí PHP na existujícím PHP serveru DPMJ, čímž by došlo k ušetření nákladů na zřízení hostingu pro výslednou aplikaci, které by bylo nutné vynaložit v případě ostatních zmíněných technologií.



Obrázek 5.2: PYPL Index top 10. Data dostupná na [27]

### Smalltalk

Smalltalk, konkrétněji Pharo, představuje další programovací jazyk, se kterým se autor práce setkal v rámci studia, a který umožňuje implementaci webových aplikací (např. s využitím knihovny Zinc).

Jako takový představuje smalltalk obecně velmi zřídka používanou technologii a o to méně pak v případě webových aplikací. Ačkoliv je tak jeho komunita velmi malá, je poměrně aktivní a sdílá a řešení některých problémů se lze dobrat relativně rychle (např. s využitím kanálu Phara na platformě Discord).

Problémem je ovšem omezené množství knihoven a dokumentace, kdy je vývojář mnohdy nucen správný způsob využití API dané knihovny vyvodit z jejích testů a příkladů, které jsou ve většině případů zahrnuté v dané knihovně a jsou společně s ní nahrány do vývojového prostředí při její instalaci.

Autor práce pak, ačkoliv využil této technologie i mimo předměty, které se jí přímo zabývájí (BI-OOP, NI-MOP), disponuje jen velmi omezenou znalostí vývoje webových aplikací na této platformě.

Neznalost této technologie v rámci IT komunity pak rovněž představuje problém z dlouhodobého hlediska, kdy bude třeba nutné předat projekt do



údržby jinému vývojáři, který s největší pravděpodobností nebude disponovat jakoukoli znalostí jazyka smalltalk a platformy Pharo.

### Závěr

Jako první lze z výběru bezesporu vyřadit jazyk Smalltalk a to především díky jeho malé rozšířenosti v rámci IT komunity, jakož i menšímu množství dostupných knihoven a dokumentace.

Zbývající technologie, tedy C#, Java EE a PHP, představují všechny poměrně solidní volby pro vývoj webových aplikací, ovšem jak v případě C#, tak i v případě Javy, jsou znalosti autora této práce omezené a obou technologií by rovněž bylo zapotřebí vynaložení dalších nákladů na zřízení hostingu pro výslednou aplikaci.

Jako logickou volbou se tak jeví PHP, v jehož případě jsou zkušenosti autora znatelnější, a které díky jeho rozšířenosti a velikosti komunity disponuje obstojným množstvím knihoven poskytujících další funkcionalitu, jakož i nepřehledným množstvím návodů a rad, které mohou značně přispět k rychlému vývoji konečného řešení.



## Návrh uživatelského rozhraní

Návrh uživatelského rozhraní představuje velmi důležitou součást vývoje software, neboť ať poskytuje vyvíjený software sebelepší funkcionalitu, nebude ve výsledku koncovými uživateli využíván, pokud nebude rovněž disponovat uživatelským rozhraním odpovídající kvality.

Jako takový má tedy přímý vliv na potenciální úspěch či neúspěch daného software a je třeba mu věnovat zvýšenou pozornost a to zvláště v případě systémů, které jsou využívány velmi frekventovaně a u kterých je tak nutno zajistit co největší efektivitu provádění jednotlivých úkonů uskutečňovaných koncovým uživatelem.

### 6.1 Zásady tvorby uživatelského rozhraní

V průběhu let, kdy se odborná IT komunita zabývala návrhem uživatelského rozhraní, vzniklo množství různých doporučení, guidelines či takzvaných pravidel palce vycházejících z množství empirických studií, studie lidského vnímání či užití prostého selského rozumu, jejichž cílem je vybavit vývojáře sadou instrukcí a rad, které mu pomohou vytvořit kvalitní uživatelské rozhraní. [28]

Příkladem může být takzvané Nielsenovo desatero, s nímž se autor práce setkal v rámci předmětu MI-NUR, a jež se skládá z následujících deseti zásad. [29]

**Viditelnost stavu systému:** Systém by vždy měl informovat uživatele o aktuálním dění v systému a to v rozumném čase.

**Shoda mezi systémem a realitou:** Systém by měl využívat jazyk a terminologii srozumitelnou cílové skupině uživatelů a zároveň by měl zachovávat konvence reálného světa (např. ikonka odpadkového koše slouží pro odstranění položky).

**Minimální zodpovědnost (a stres):** Uživatelé při vykonávání úkonů v systému často chybují. Systém by tak v ideálním případě měl nabízet jednoduchý způsob, jak se vrátit do stavu před započítím nechtěného úkonu.

**Konzistence a dodržení standardů:** Systém by měl využívat konvencí dané platformy. Neměl by tedy vystavovat uživatele zbytečné nejistotě použitím pro danou platformu nekonvenčních ovládacích prvků a pojmů.

**Prevence chyb:** Systém by měl být navržen tak, aby pokud možno zabránil vzniku problémů. Toho může být dosaženo například včasnou kontrolou vstupních hodnot či vyžádáním potvrzení dané akce, které může pomoci tomu, aby si uživatel uvědomil, že buď danou akci nechce provést vůbec, nebo že při jejím zadávání někde udělal chybu.

**Kouknu a vidím:** Systém by měl klást minimální požadavky na paměť uživatele. Jednotlivé akce, které může uživatel na dané obrazovce systému provést, by měly být dobře viditelné a snadno dosažitelné. Systém by rovněž neměl uživatele nutit k zapamatování si informací z jím dříve vykonaných kroků.

**Flexibilita a efektivita:** Systém by měl obsahovat nástroje pro urychlení často vykonávaných úkonů (mapř. klávesové zkratky), které ač neviditelné a neznámé nezkušenému uživateli, mohou výrazně usnadnit práci zkušenému uživateli. Systém by tak měl být navržen s ohledem na skutečnost, že ho využívají jak zkušení, tak i nezkušení uživatelé.

**Minimalita:** Na aktuální obrazovce by měly být zobrazeny pouze potřebné informace. Jakákoli další zobrazená informace může zbytečně odvádět pozornost uživatele od těch skutečně podstatných.

**Smysluplné chybové hlášky:** Chybové hlášky by měly být vyjádřeny pomocí běžného jazyka, měly by přesně indikovat problém, jehož jsou následkem, a případně i navrhnout řešení onoho problému.

**Nápověda a dokumentace:** Ačkoliv by v ideálním případě měl být systém použitelný bez nápovědy či dokumentace, může být v některých případech nutné nápovědu do systému zahrnout. V takovém případě by měla být tato nápověda stručná a věcná a mělo by v ní být možné snadno vyhledávat, tak aby se uživatel dobral řešení nastalého problému pokud možno co nejrychleji.

Dodržením těchto zásad by pak měly být eliminovány alespoň základní prohřešky, kterých se vývojář může dopustit při návrhu uživatelského prostředí.

## 6.2 Volba barevného schématu

Důležitá je i správná volba barev použitých pro finální uživatelské rozhraní, která může mít výrazný vliv na použitelnost výsledné aplikace a na způsob, jakým je koncovým uživatelem vnímána.

Při tvorbě uživatelského rozhraní nesmí vývojář zapomenout, že existují určité kulturní konvence, co se týče významu jednotlivých barev, které se v různých kulturách mohou diametrálně lišit. Pokud tedy aplikace zamýšlí cílit na uživatele jiných kultur je záhodno prozkoumat tyto konvence a na základě toho upravit případnou volbu barev.

Systém, jehož vývojem se zabývá tato práce, slouží k poskytování informací a autor práce se tak rozhodl držet se rámcově klasického podání černé na bílém s využitím odstínů modré, která je v našich končinách vnímána jako barva klidu.

Kromě kulturních významů jednotlivých barev je pak rovněž nutné brát ohled na čitelnost textů obsažených v systému. Volba barev pozadí a textu ji totiž může značně ovlivnit a v případě špatně zvolených barev může klást zbytečně vysoké nároky na zrak uživatele, způsobovat mu únavu očí a celkově zhoršit použitelnost výsledného systému.

Z toho důvodu je nutné věnovat pozornost tomu, jaký kontrastní poměr zvolené barvy poskytují tak, aby byla zajištěna dobrá čitelnost.

Tento přístup lze ilustrovat na volbě barev klientské části, jejichž volba byla provedena s využitím tohoto nástroje pro výběr barev [30], přičemž kontrastní poměr zvolených barev byl ověřen s pomocí nástroje [contrast-ratio.com](https://contrast-ratio.com) [31].



Obrázek 6.1: Výsledné barevné schéma klientské části

Jako základní barva pozadí je v případě klientské části využita barva #FFFCF9 (Baby powder), u které je s použitím barvy #483D3F (Black coffee) jako hlavní barvy pro text zajištěn dostatečný kontrastní poměr (10,19) pro zajištění dobré čitelnosti textu v rámci aplikace. Dobrý kontrastní poměr (7,81) pak rovněž poskytuje kombinace hlavní barvy pozadí a barvy #024DA8 (Cobalt blue), která je v aplikaci využita jako barva horní lišty a pro zvýraznění některých textových prvků. U poslední zvolené barvy (#C9F0FF – Light blue), která slouží pro zvýraznění prvků v aplikaci, lze rovněž bez problému využít hlavní barvu textu a to se slušným kontrastním poměrem 8,63.

Při doporučeném kontrastním poměru 7:1 [32], by tak výsledné barevné schéma mělo zajistit relativně dobrou čitelnost textů v rámci celé aplikace.

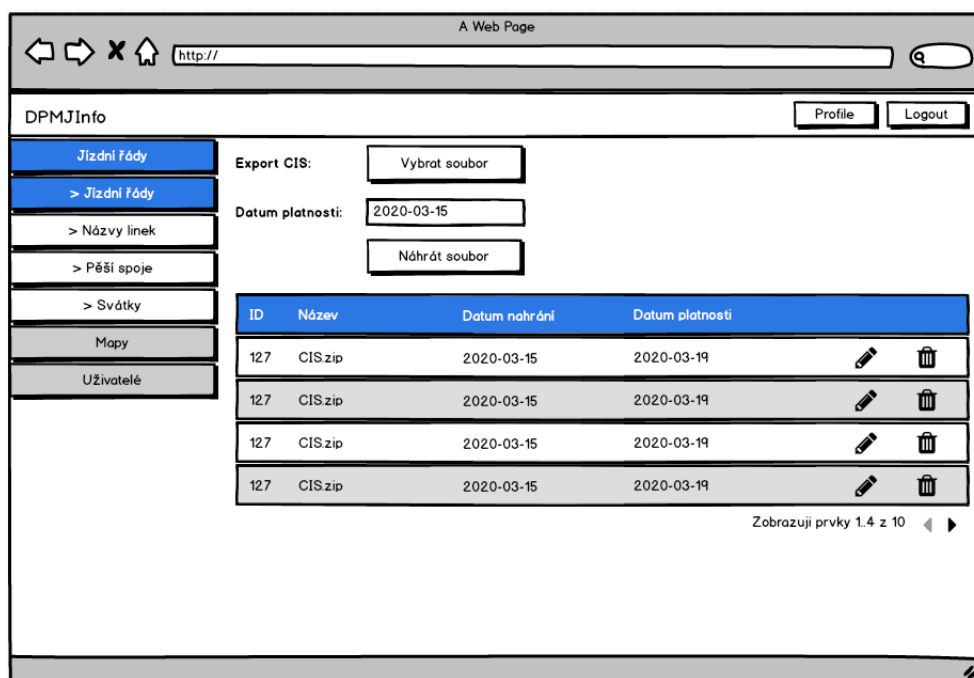
### 6.3 Správcovská část systému

Před započítím samotného návrhu uživatelského rozhraní v podobě takzvaných drátěných modelů, je zapotřebí nejdříve identifikovat jednotlivé úkony, které uživatel může v rámci systému vykonávat, jakož i jednotlivé obrazovky, které bude systém obsahovat, přičemž lze vycházet z případů užití získaných v rámci analýzy.

Jednotlivé identifikované úkony (zelené) a obrazovky (modré) lze vidět na diagramu 6.4.

Jak si čtenář povšimne domovská obrazovka není zbarvena modře, ačkoliv by intuitivně předpokládal, že se jedná o obrazovku. Je to z toho důvodu, že uživatel je systémem po přihlášení automaticky přesměrován na jednu z obrazovek, na kterou se z této virtuální domovské obrazovky dá dostat v rámci diagramu, a to v závislosti na oprávněních, která byla danému uživateli přiřazena.

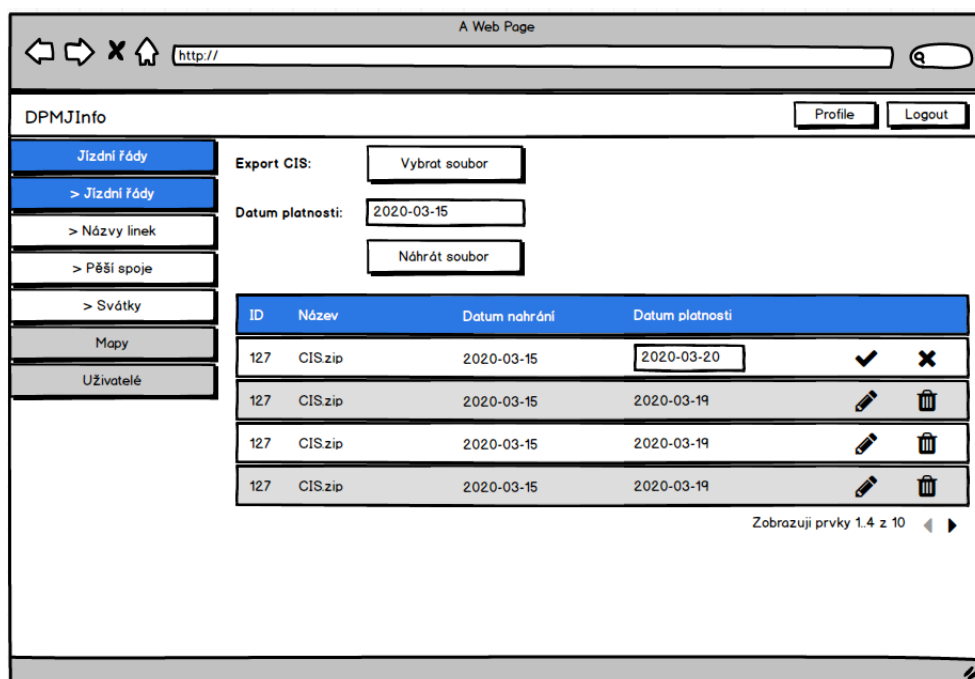
Jako příklad drátěného modelu je zde uvedena obrazovka pro správu jízdních řádů (obrázek 6.2), která bude v rámci výsledného systému využívána nejčastěji. Ostatní obrazovky jsou pak strukturou velmi podobné, s výjimkou obrazovky uživatelského profilu a obrazovky pro úpravu uživatele, které obsahují pouze formuláře pro úpravu příslušných hodnot. Drátěné modely ostatních obrazovek je pak možné najít v přílohách této práce.



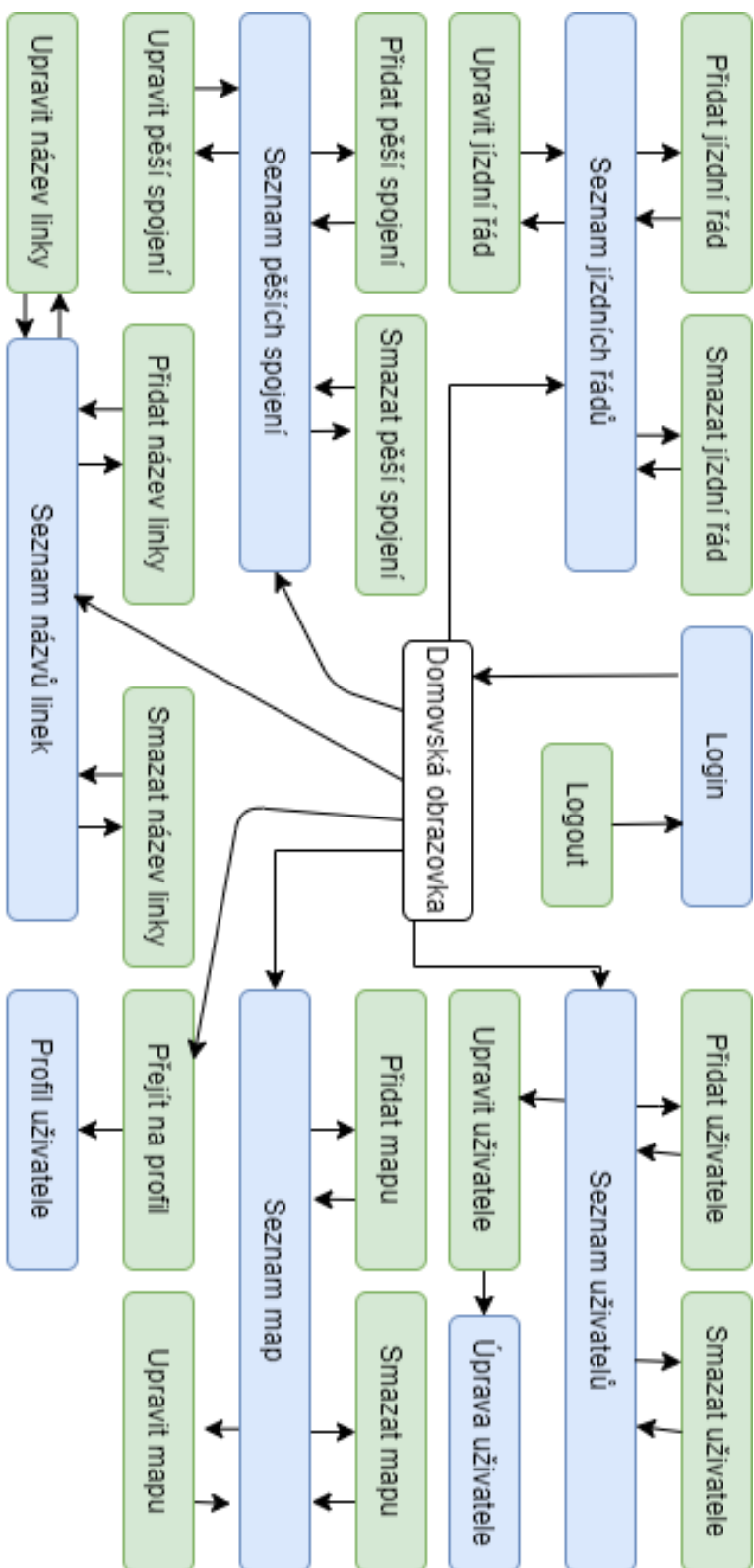
Obrázek 6.2: Drátěný model obrazovky pro správu jízdních řádů

### 6.3. Správcovská část systému

Jak je možné si povšimnout na drátěném modelu 6.3, snažil se autor práce navrhnout uživatelské rozhraní tak, aby pokud možno šlo snadno provést většinu úkonů souvisejících se správou dané kategorie položek v rámci jedné obrazovky. Nad samotný seznam položek tak byl vložen formulář pro vložení nové položky. Editace jednotlivých položek pak probíhá přímo v rámci seznamu položek, což je možné díky skutečnosti, že u většiny systémem spravovaných entit lze upravovat pouze jednu hodnotu (s výjimkou uživatelů). U jízdních řádů se například jedná o datum, kdy daný jízdní řád vstupuje v platnost.



Obrázek 6.3: Drátěný model obrazovky pro správu jízdních řádů - úprava položky



Obrázek 6.4: Task diagram správce části



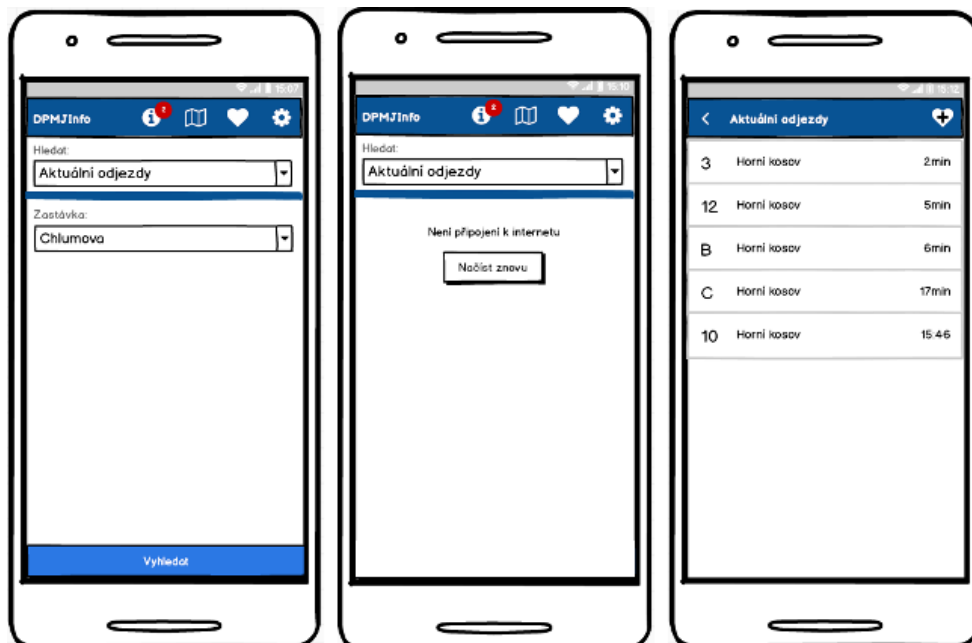
## 6.4 Klientská část systému

Stejně jako v případě správcovské části systému i u klientské části došlo nejdříve k identifikaci jednotlivých obrazovek a možných úkonů, přičemž výsledek tohoto kroku je vidět na diagramu 6.7.

Výsledný diagram je ovšem poněkud zjednodušen vzhledem k potřebě jeho zahrnutí do tohoto výtisku. Toto zjednodušení spočívá především v uvedení pouze jedné obrazovky pro zobrazení výsledků dotazů, přičemž by správně měla být pro každý druh dotazu, tedy vyhledání aktuálních odjezdů, odjezdů a spojení, uvedena samostatná obrazovka s příslušnými úkony, kdy by úkon zobrazení detailu spoje a navazující obrazovka byly správně uvedeny pouze ve spojení s obrazovkou s výsledky vyhledávání odjezdů, u níž je jako u jediné možné tento úkon provést.

Diagram se v tomto ohledu čirou náhodou blíží spíše finální implementaci, kdy je pro vykreslení výsledků dotazů použita stejná obrazovka, která díky využití polymorfismu umožňuje vykreslení výsledků různých druhů dotazů.

Jako v případě správcovské části pak bylo i při návrhu uživatelského rozhraní klientské části využito drátěných modelů, které disponují tou výhodou, že je lze v porovnání s hi-fi prototypy tvořit rychle a poměrně jednoduše a je tak možné bez větší námahy a zbytečně vynaložených nákladů měnit v průběhu vývoje na základě nově získaných poznatků či připomínek potenciálních uživatelů.



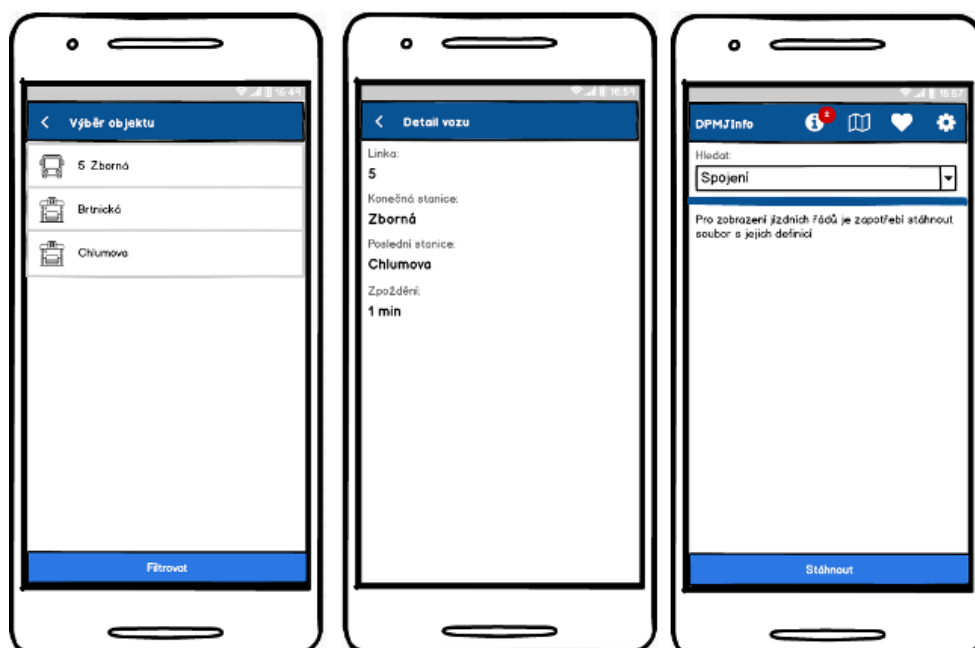
Obrázek 6.5: Ukázka drátěných modelů klientské části 1

## 6. NÁVRH UŽIVATELSKÉHO ROZHRANÍ

Na domovskou obrazovku, jejíž drátěný model je možné vidět na obrázku 6.5 vlevo, autor práce umístil samotné vyhledávání v jízdnicích řádech, které představuje centrální funkcionalitu aplikace a jež je z pohledu uživatele nejčastěji využívanou částí funkcionality výsledné aplikace. Z domovské obrazovky je pak rovněž umožněn přístup k dalším funkcionalitám aplikace, jako je zobrazení aktualit, online mapa dopravy či oblíbených položek, a to pomocí ikon v horní liště aplikace.

Mezi jednotlivými typy vyhledávání (aktuální odjezdy, odjezdy, spojení) může pak uživatel přepínat pomocí nabídky v horní části obrazovky, která je znatelně graficky oddělena od části "formulářové", kde je možné zadat jednotlivé parametry daného typu dotazu.

V rámci mapové části aplikace se pak autor práce pokusil vyřešit nedostatek, jímž disponovala většina obdobných aplikací, jež autor zkoumal, kdy aplikace při výběru objektů v mapě vždy dojde k výběru nejbližšího objektu od místa dotyku a to bez ohledu na ostatní objekty, které leží v jeho těsném okolí. Uživatel je tak v případě těchto aplikací nucen zdlouhavě manipulovat s mapou (přibližování a posun) tak, aby mohl s jistotou vybrat ze skupiny blízko ležících objektů ten jím kýžený.



Obrázek 6.6: Ukázka drátěných modelů klientské části 2

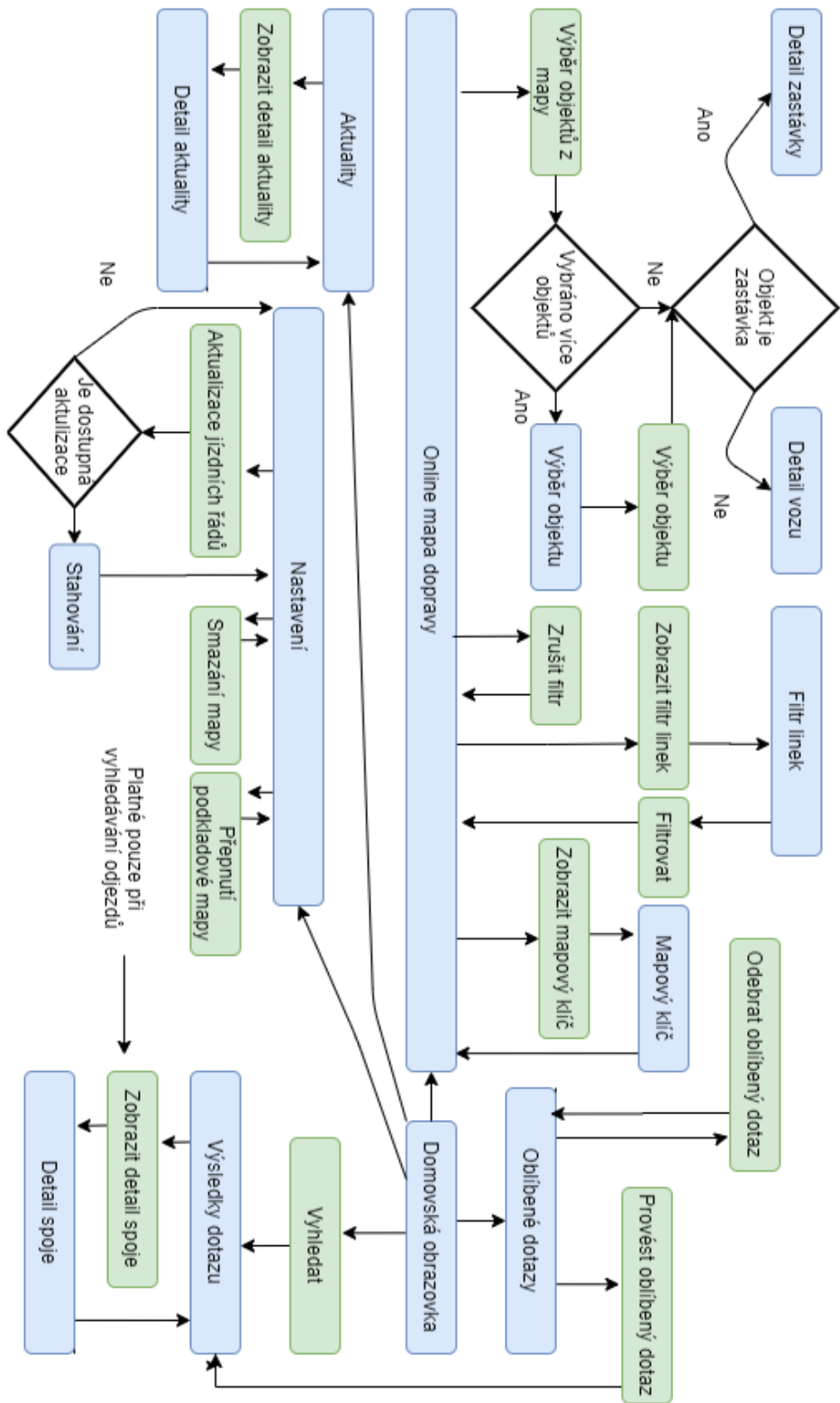
Tento nedostatek se autor práce pokusil vyřešit tak, že v případě, že je v těsné blízkosti dotyku přítomno více objektů, spíše než aby byl vybrán ten nejbližší, který nemusí být nutně tím, na který chtěl uživatel skutečně kliknout, zobrazí aplikace nabídku pro výběr z těchto objektů (drátěný model na

obrázku 6.6 vlevo), díky němuž se uživatel dostane k detailu kýženého objektu mnohem rychleji než u obdobných aplikací, u kterých by se musel v případě výběru nesprávného objektu vracet v rámci aplikace o krok zpět a následně manipulovat s mapou tak, aby mohl vybrat ten správný objekt.

Napraveny rovněž byly nedostatky uživatelského rozhraní původní aplikace týkajícího se problematického využití ovládacích prvků na dotykových zařízeních, kdy jsou v rámci různých seznamů v rámci aplikace (např. zmíněný výběr objektu z mapy) zobrazeny jednotlivé položky dostatečně velké, tak aby na ně bylo v případě potřeby možné snadno kliknout, přičemž je možné kliknout na libovolné místo položky a ne pouze na dedikovanou malou ikonu, jak tomu bylo v případě původní aplikace (viz rešerše obdobných aplikací).

Na obrázcích 6.5 a 6.6 je pak možné vidět i drátěné modely některých dalších obrazovek aplikace. Veškeré drátěné modely klientské části jsou pak součástí příloh této práce.

6. NÁVRH UŽIVATELSKÉHO ROZHRANÍ



Obrázek 6.7: Task diagram klientské části

---

# Návrh a implementace

## 7.1 Správcovská část

Jak bylo zjištěno v předcházejících kapitolách, správcovská část systému slouží ke správě jednotlivých offline podkladů, které jsou zapotřebí pro zajištění funkcionality části klientské. Jedná se celkově o tři typy offline podkladů: jízdní řády, mapové podklady a definice svátků.

### Jízdní řády

Jízdní řády představují nejdůležitější offline podklad, se kterým obě části systému pracují, přičemž správcovská část se zabývá jejich evidencí a zpracováním do finální formy, ve které jsou následně využívány v klientské části.

Jako takové jsou jízdní řády výstupem programu Grafikon, který umožňuje DPMJ definovat jednotlivé linky a následně spoje těchto linek ve formě takzvaných grafikonů, které představují grafy, u nichž jsou na svislé ose přítomny zastávky dané linky a na vodorovné ose je vyneseno čas, přičemž výsledný spoj je dán jako spojnice průsečíků daného času s příslušnou zastávkou.

Tento program umožňuje kromě jízdních řádů čitelných lidmi, které jsou vystaveny na jednotlivých zastávkách a webovém portálu DPMJ, exportovat rovněž jízdní řády hned v několika strojově čitelných formátech. Celkově systém umožňuje export v pěti různých formátech:

- ME2,
- BUSE,
- CIS,
- FS Mobil,
- Google.

ME2 a BUSE představují interně využívané exporty, kdy ME2 je využíván v rámci odbavovacího systému v jednotlivých vozech a formát BUSE pro provoz takzvaných chytrých zastávek, které na zabudovaném displeji zobrazují aktuální odjezdy spojů z dané zastávky a to včetně případného zpoždění.

Exporty CIS, FS Mobil a Google pak představují formáty, které jsou určeny pro využití externími aplikacemi, přičemž formát FS Mobil je proprietárním formátem, který je využíván původní aplikací pro zobrazování jízdních řádů JRm od společnosti FS Software s.r.o. Export pro Google, jak již název napovídá, je pak využíván službami společnosti Google.

Posledním možným formátem pro export je pak formát CIS, který je definován Ministerstvem dopravy ČR pro potřeby celostátního informačního systému o jízdních řádech, a k jehož vedení jsou smluvně vázáni všichni poskytovatelé veřejné dopravy.

Právě poslední zmíněný formát se pak autor práce rozhodl využít i v rámci nově implementovaného systému a to především z toho důvodu, že se jedná, jak je zmíněno výše, o státem definovaný jednotný formát, který jsou povinni vést všichni veřejní dopravci, díky čemuž bude případně možné znovupoužití větší části systému i pro jiné dopravce.

Export CIS ovšem disponuje nevýhodou v podobě absence názvů linek ve výsledném exportu, díky čemuž je nutné ve správcovské části systému umožnit zadání mapování mezi identifikátorem linky a jejím názvem.

Jízdní řády je pak v rámci systému možné obohatit o pěší spojení, která ručně zadává správce jízdních řádů. Pro potřebu jejich zadávání pak systém uchovává ve své databázi rovněž seznam zastávek, který je extrahován z exportovaných jízdních řádů při jejich nahrání do systému.

Export jako takový je do systému nahrán jako komprimovaný archiv (zip), přičemž při jeho nahrání dojde k extrakci potřebných dat a jejich transformaci pro využití klientskou částí do podoby SQLite databázového souboru. Jednotlivé verze jízdních řádů jsou pak evidovány systémem v rámci databázové tabulky uploads.

Správcovská část systému tak operuje nad dvěma různými databázovými schémata, přičemž první slouží k uložení jím evidovaných entit (uživatelské účty, evidence nahraných souborů, pěších spojů atd.), a jež je k vidění na obrázku 7.1.

Druhé databázové schéma (obrázek 7.2) je pak vystavěno nad daty extrahovanými z exportu jízdních řádů obohacených o definované názvy linek a pěší spoje, přičemž názvy linek jsou přidány přímo do tabulky linek a pěší spoje jsou pak reprezentovány samostatnou tabulkou.

V rámci tohoto schématu pak stojí za zmínku ještě tabulka graphDepartures, která obsahuje denormalizovaná data z ostatních tabulek a která byla vytvořena v návaznosti na značnou časovou náročnost SQL dotazu připravujícího tyto data pro využití v případě vyhledávání spojení, u kterého tak došlo k výraznému zlepšení odezvy systému.

## Definice svátků

Překvapivým problémem v rámci systému představuje určení státních svátků, jejichž definice je zapotřebí při vyhledávání v jízdních řádech v rámci klientské části vzhledem k odlišným jízdním řádům, podle nichž linky o státních svátcích jezdí.

Problém představují především pohyblivé státní svátky, jimiž jsou u nás Velký pátek a Velikonoční pondělí a u nichž není pevně dané datum, které se mění v závislosti na datu prvního jarního úplňku.

Původním autorovým nápadem bylo pro řešení tohoto problému využít veřejný google kalendář České státní svátky [33] a v rámci klientské části na začátku roku načíst a uložit data o svátcích nadefinovaných v tomto kalendáři s využitím příslušného API. Jak se ovšem ukázalo, obsahuje tento kalendář i dny, které nejsou uznanými státními svátky jako například Den svatého Valentína, Mezinárodní den žen či Den matek, díky čemuž je tento kalendář pro získání informací o státních svátcích nepoužitelný.

Autor práce se tak nakonec rozhodl pro zadávání nepohyblivých státních svátků prostřednictvím csv souboru, v němž jsou zaneseny data jednotlivých svátků. Velikonoční státní svátky jsou pak určeny pomocí algoritmu zvaného *computus* v rámci klientské části, díky čemuž není nutné definici státních svátků každý rok aktualizovat.

## Třídní model

V rámci backendu správcovské části systému mají na starosti většinu aplikační logiky systému třídy odvozené od třídy *BaseAPI*, tedy *UploadsAPI*, *LineNamesAPI*, *UserAPI* a *FootConnectionAPI*. Tyto třídy, jak již jejich název napovídá, definují API pro manipulaci s jednotlivými entitami v rámci systému. V rámci každé z těchto tříd jsou implementovány metody pro vykonání jednotlivých CRUD operací, které jsou spouštěny v závislosti na metodě příchozích HTTP požadavku, tedy *GET*, *POST*, *PUT* a *DELETE*.

Ve většině tříd pak dochází především k validaci jednotlivých vstupů a následná invocace některé z metod třídy *ServerAppDAO*, v rámci které je následně využito příslušné instance jedné z DAO tříd pro propagaci změn dané entity do databáze systému. Třída *ServerAppDAO* jako taková neposkytuje jakoukoli další funkcionalitu, avšak jako taková přináší výrazné zjednodušení závislostí v rámci systému a zároveň umožňuje snadný přístup k celé databázi prostřednictvím jediné třídy.

S jednotlivými DAO třídami, včetně *ServerAppDAO*, jsou pak úzce spjaty třídy odvozené od třídy *BasePDO*, která rozšiřuje systémovou třídu *PDO* jazyka PHP. Třída *BasePDO* jako taková definuje několik utility metod usnadňujících manipulaci s daty databáze. Z této třídy jsou následně odvozeny třídy *DefaultPDO* a *TestPDO*, přičemž *DefaultPDO* je v rámci systému využívána pro přístup jak k hlavní databázi správcovské části systému, tak i k přístupu

k SQLite databázím jednotlivých verzí jízdních řádů. TestPDO poté, jak již název napovídá, slouží pro připojení k testovací databázi v rámci provádění automatických testů, pro jejichž účely definuje metodu clearData, pro kompletní smazání aktuálního obsahu databáze, které je zapotřebí provést mezi jednotlivými testy.

Posledními zatím nezmíněnými třídami, které jsou vyobrazeny na zjednodušeném diagramu tříd 7.3, jsou třída AJAXResponse a do ní odvozené třídy, které jsou využívány API třídami k předání jimi získaných hodnot či případných chybových hlášení frontendu správcovské části a rovněž části klientské.

### Uživatelské rozhraní

Uživatelské rozhraní správcovské části systému je navázáno na backend pomocí AJAX požadavků, pro jejichž realizaci a následné zpracování navrácených hodnot je použita javascriptová knihovna jQuery [34], bez níž se dnes prakticky neobejde jakákoli webová stránka.

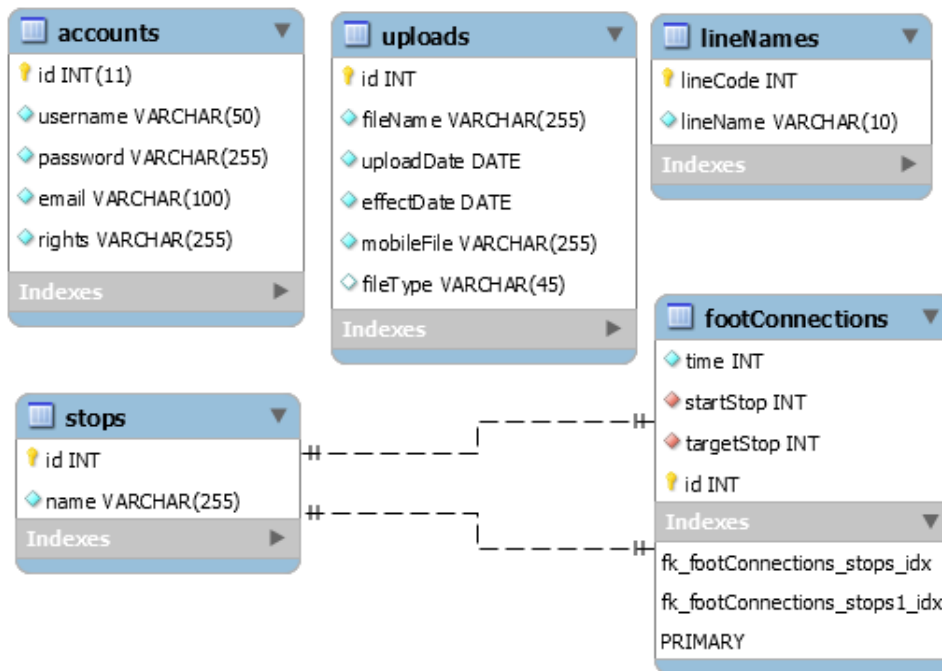
Kromě samotného jQuery je pak rovněž využito widgetu DatePicker z jQueryUI [35], který umožňuje uživateli snadný výběr data pomocí kalendáři podobného ovládacího prvku. Vývojáři pak přináší například možnost jednoduché restrikce dat, které může uživatel vybrat. Pro další zpříjemnění uživatelského zážitku je dále použit jQuery plugin Select2 [36], který je využit pro usnadnění zadávání zastávek při vytváření nového pěšího spojení díky možnosti vyhledávání v rámci položek nabízených k výběru.

Pro zobrazení hlášek o stavu systému je pak použit jQuery plugin Notify.js [37], pomocí něž je uživateli přehledně poskytována zpětná vazba k jednotlivým úkonům, které v systému provádí, a to ať již byl úkon úspěšně dokončen, či během jeho zpracování systémem došlo k chybě.

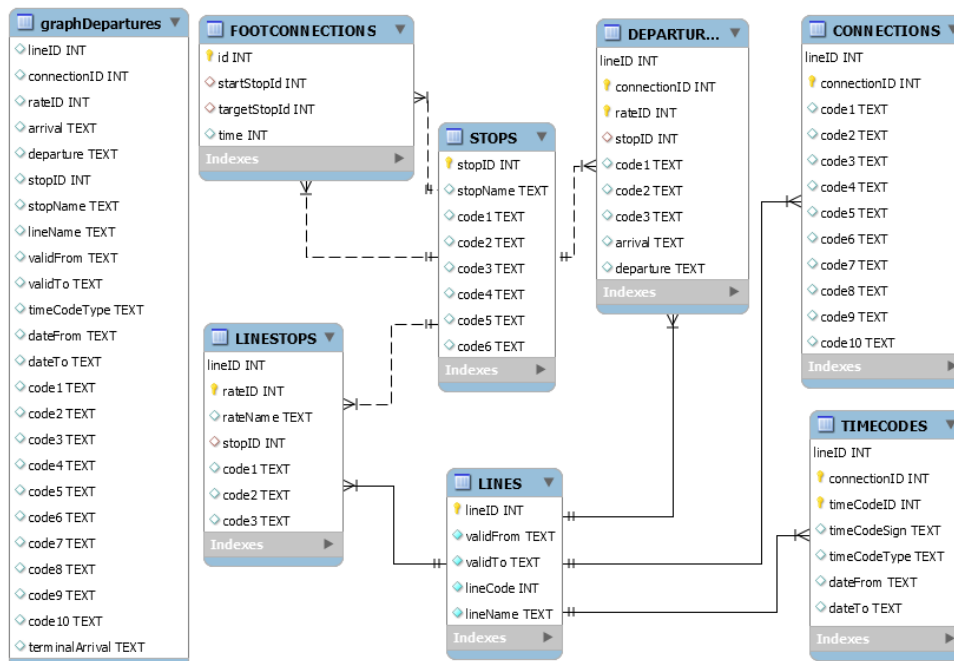
Poslední použitou javascriptovou technologií je plugin jQuery Validation Plugin [38], který umožňuje snadnou a efektivní validaci hodnot zadaných uživatelem před jejich odesláním na server. Uživatel je tak upozorněn na případnou chybu mnohem dříve.

Kromě javascriptových technologií je pak ještě využit grafický framework Bootstrap [39], s pomocí něž lze velmi rychle a jednoduše nadefinovat rozložení výsledného webu.

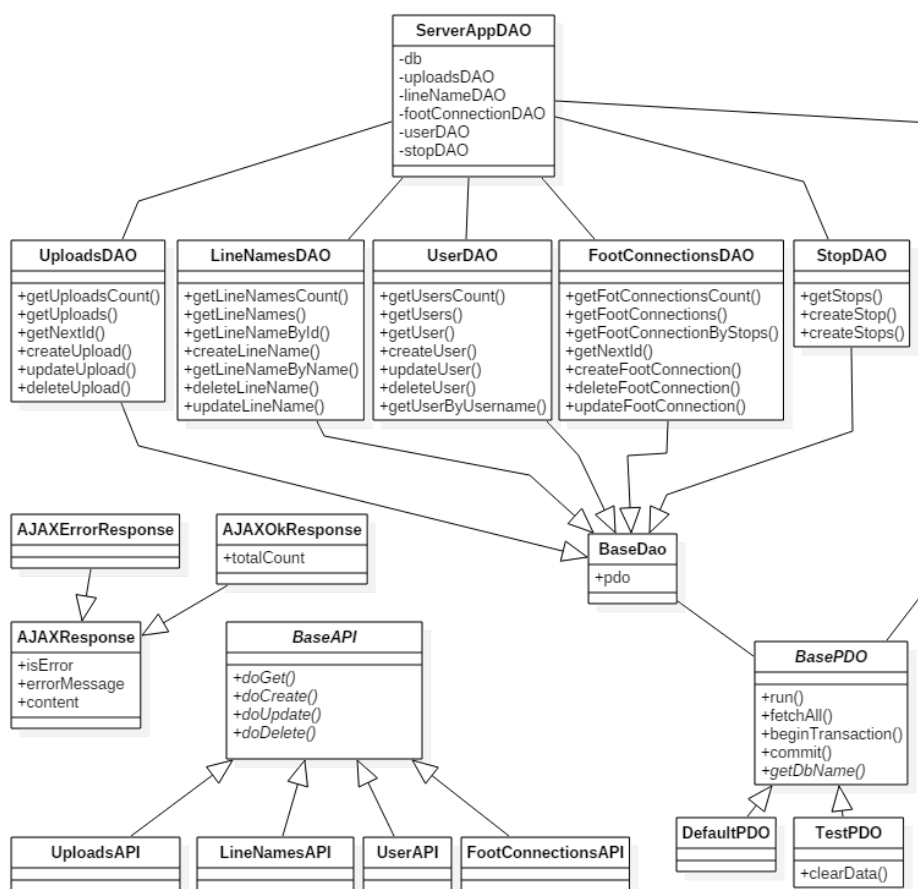




Obrázek 7.1: Databázové schéma správcovské části systému



Obrázek 7.2: Databázové schéma SQLite databáze jízdních řádů



Obrázek 7.3: Zjednodušený třídní model správcovské části systému

## 7.2 Klientská část

Klientská část systému sestává z vlastní nativní aplikace pro operační systém Android, jejíž implementace je realizována v jazyce Java s využitím Android Studio IDE, a jako taková nabízí tato aplikace následující tři typy funkcionality: zobrazení aktuálních odjezdů a vyhledávání v jízdních řádech, zobrazení aktuálních zpráv o změnách v dopravě a zobrazení online mapy dopravy.

### Aktuální odjezdy a vyhledávání v jízdních řádech

Funkcionalita spojená s jízdními řády představuje páteřní funkcionalitu celé aplikace. Nově implementovaná aplikace pak přináší v tomto ohledu oproti původně využívané aplikaci JRm možnost zobrazení aktuálních odjezdů z dané zastávky.

Tato nová funkcionální je umožněna díky zavedení chytrých zastávek, které jsou umístěny na několika vybraných zastávkách a jsou schopny zobrazovat aktuální odjezdy spojů z dané zastávky se započítáním případného zpoždění. Systém, který poskytuje data těmto několika chytrým zastávkám, je schopný poskytnout tyto odjezdy i pro zastávky, na kterých fyzicky chytrá zastávka umístěna není, díky čemuž je možné v rámci nově implementované aplikace zobrazit aktuální odjezdy pro libovolnou zastávku. Jako taková se tato funkcionální kvůli nutné komunikaci se systémem chytrých zastávek liší od zbytku funkcionality spojené s jízdními řádů, kterou je možné využívat i bez připojení k internetu.

Vyhledávání odjezdů a spojení je možné realizovat offline a to díky skutečnosti, že jsou do mobilního zařízení uživatele staženy definice jízdních řádů a svátků, které jsou poskytovány správcovskou částí systému. Výsledná aplikace pak vždy při spuštění zkontroluje (pokud je dostupné připojení k internetu) dostupnost aktuálnější verze těchto podkladů, načtež nabídne uživateli možnost jejich stažení.

Jak již bylo zmíněno v této kapitole v sekci zabývající se implementací správcovské části systému, je definice svátků reprezentována pomocí csv souboru obsahujícím data jednotlivých svátků a definice jízdních řádů je tvořena SQLite databázovým souborem, přičemž tato forma reprezentace jízdních řádů byla zvolena vzhledem k zabudované funkcionální Android aplikací, která umožňuje tento typ databáze v rámci aplikace využívat. Ve výsledku je tak možné pro vyhledávání odjezdů využít prostý SQL dotaz.

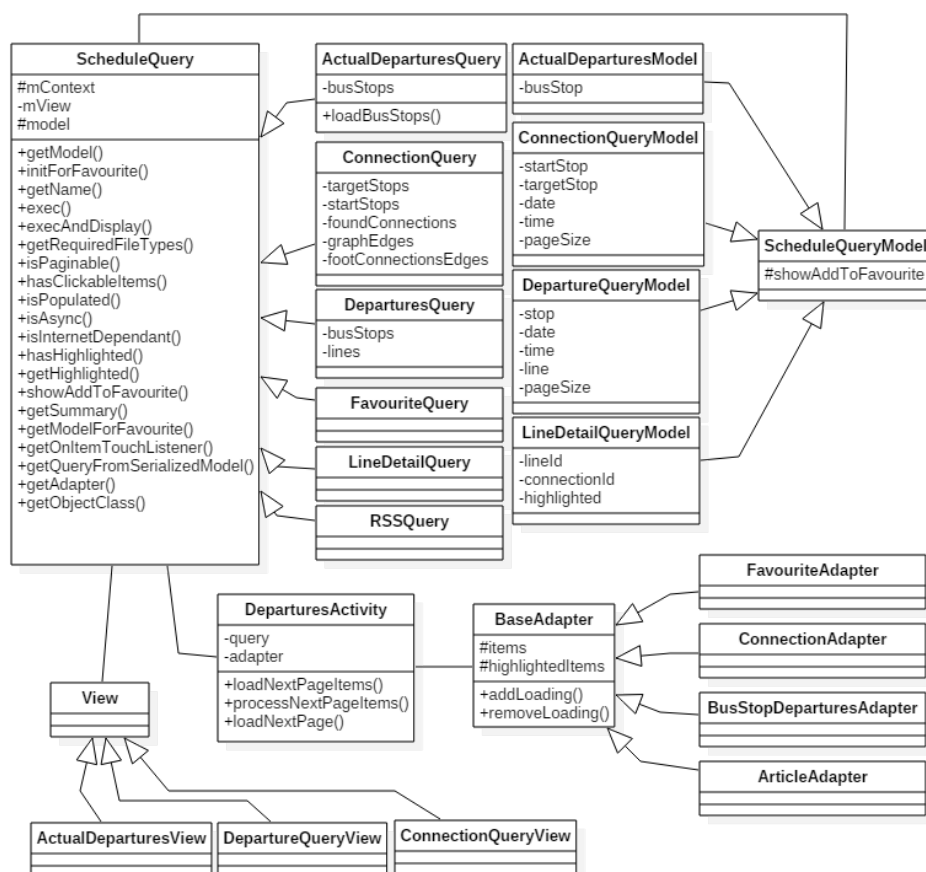
V případě vyhledávání spojení je pak situace poněkud odlišná vzhledem ke skutečnosti, že je nad jízdními řádů nutné spustit algoritmus pro nalezení nejkratšího spojení mezi danými zastávkami, kdy je nutné ve většině případů stejně načíst většinu dostupných dat. I v tomto případě lze však možnost dotazování pomocí SQL částečně využít, kdy je možné snadno vyfiltrovat z dat, nad kterými bude následně spuštěn vyhledávací algoritmus, ty spoje, jejichž čas odjezdu z dané zastávky je nižší než čas zadaný uživatelem.

S funkcionální jízdních řádů pak rovněž souvisí možnost uložení oblíbených dotazů, která umožňuje se rychleji dostat k výsledkům dotazů, které uživatel provádí častěji.

V rámci systému jsou jednotlivé dotazy reprezentovány podtřídami třídy `ScheduleQuery`, která v rámci svých atributů drží základní požadované hodnoty, jako je aktuální aplikační kontext, model daného dotazu či případně instanci pohledové třídy.

Jednotlivé modely dotazů pak schraňují parametry daného dotazu zadaného uživatelem a jako takové jsou oddělené od vlastní třídy dotazu kvůli potřebě předávání dotazů aktivitě `DeparturesActivity`, která se stará o zobrazení jejich výsledků. V rámci aplikací pro operační systém Android totiž bohužel není možné předávat objekty mezi jednotlivými aktivitami aplikace prostřednictvím referencí, nýbrž je možné předávat pouze serializovaná data pomocí tříd `Intent` a `Bundle`. Při předávání dotazu aktivitě `DeparturesActi-`

vity je tak předáván pouze serializovaný model dotazu, ze kterého je následně pomocí statické metody `getQueryFromSerializedModel` třídy `ScheduleQuery` vytvořena instance příslušné třídy dotazu, které jsou pomocí deserializovaného modelu přiřazeny původní parametry předávaného dotazu.



Obrázek 7.4: Zjednodušený třídní model dotazů

Další výhodou tohoto oddělení parametrů dotazu do serializovatelného objektu je pak možnost uložení serializovaného modelu do databáze při ukládání oblíbených dotazů uživatele, díky čemuž pro každý oblíbený dotaz stačí uložit do databáze jen dvě hodnoty: typ daného dotazu a serializovaný model obsahující příslušné parametry dotazu.

Výsledky všech typů dotazů jsou pak zobrazeny pomocí jediné aktivity `DeparturesActivity`. To je možné díky implementovanému polymorfismu u tříd dotazů, kdy jsou výsledky jednotlivých dotazů zobrazovány touto aktivitou na základě základě návratových hodnot metod, které jsou deklarovány v rámci třídy `ScheduleQuery` a jejichž implementace se liší u jejich jednotlivých po-

tomků dle jejich konkrétních potřeb. Příkladem mohou být například metody `isAsync` a `isPaginable`, přičemž první metoda určuje, jestli mají být výsledky dotazů načítány pomocí asynchronního procesu či nikoli. Druhá metoda pak udává, jestli dotaz poskytuje výsledky ve stránkované podobě, a jestli má tedy aktivita při skrolování výsledky umožnit načítání dalších stránek výsledku.

Nejdůležitější metodou pro potřeby vykreslení výsledků je pak metoda `getAdapter`, v rámci níž je aktivitě `DeparturesActivity` předán adaptér (`FavouriteAdapter`, `ConnectionAdapter`, `BusStopDeparturesAdapter`) používaný ovládacím prvkem `RecyclerView` systému Android k vykreslení jednotlivých položek získaných výsledků v podobě skrolovatelného seznamu. Jednotlivé adaptéry tedy jinak řečeno transformují data obsažená ve výsledku dotazu do jim příslušné grafické podoby.

## Aktuální zprávy o změnách v dopravě

Další z funkcionalit poskytovaných klientskou částí systému je možnost zobrazení aktualit o změnách v dopravě, která je v rámci výsledné aplikace implementována zatím pouze jako *proof of concept* vzhledem k aktuálně probíhajícímu projektu, jehož účelem je kompletní update webového portálu DPMJ a v rámci nějž dojde k nahrazení značně zastaralého CMS Mambo, na kterém je postavena aktuální funkcionalita portálu.

Momentálně je v rámci tohoto projektu řešena problematika uživatelského rozhraní a grafického vzhledu a není prozatím jasné, jaký systém bude využit při implementaci nového webu (předpokládá se využití systému Joomla či Wordpress).

Jako vhodným způsobem pro implementaci této funkcionality se pak jeví využití RSS kanálů, jejichž využití podporuje většina dnes běžně používaných systémů pro správu obsahu. Jako takový poskytuje RSS kanál dle nastavení systému určitý počet nejaktuálnějších článků či příspěvků přidávaných na daný web a to ve formě XML souboru, ve kterém jsou obsaženy základní informace o daných článcích, jakož i jejich obsah.

Většina systémů pro správu obsahu pak rovněž umožňuje definovat samostatné RSS kanály pro jednotlivé kategorie článků, díky čemuž je možné v rámci implementované aplikace načítat pouze články kýžené kategorie (v tomto případě se jedná o zmíněné aktuality o změnách v dopravě).

Pro samotné načítání článků z RSS kanálu je pak v rámci aplikace využita knihovna `RSS-parser` [40].

## Online mapa dopravy

Poslední část implementované mobilní aplikace je tvořena online mapou dopravy, která poskytuje funkcionalitu obdobnou její webové variantě, za jejíž implementací stojí odbor informatiky jihlavského magistrátu.

Při vlastní implementaci online mapy dopravy bylo využito nástrojů pro vývoj mapových aplikací společnosti Esri (ArcGIS Android SDK) a to vzhledem ke skutečnosti, že software této společnosti je základem geografického informačního systému spravovaného Magistrátem Města Jihlavy, čímž by měla být výrazně usnadněna integrace potřebných mapových podkladů týkajících se dopravy, které jsou v rámci tohoto systému vedeny, do finální mobilní aplikace. Na technologiích od společnosti Esri je pak vybudována i webová verze online mapy dopravy (ArcGIS javascript API).

Díky využití ArcGIS Android SDK pak rovněž bylo umožněno využití RESTového API, které je na pozadí využíváno webovou verzí online mapy dopravy, což umožnilo snadno načítat údaje o jednotlivých vozech MHD, jakož i načítání nadefinovaných geografických tvarů, jimiž jsou například trasy jednotlivých linek MHD.

ArcGIS Android SDK pak rovněž umožňuje využití offline mapových podkladů při zobrazování mapy, jejichž využití je jednou z prvotních pohnutek pro vznik této aplikace. Využitím podkladové mapy, která je uložena v rámci zařízení uživatele, totiž dochází ke snížení nároků aplikace co se týče nutných datových přenosů a v důsledku tedy na datový tarif koncového uživatele.

V rámci aplikace byla nakonec bohužel využita rastrová podkladová mapa s větší velikostí (cca 150 MB v případě výřezu blízkého okolí Jihlavy) oproti vektorové verzi (cca 70 MB při stejném rozsahu mapy), vzhledem k tomu, že se autorovi práce nepodařilo online mapu dopravy s využitím vektorové mapy zprovoznit.<sup>1</sup> Do budoucna ovšem rozhodně bude usilováno o její využití namísto rastrové varianty.

---

<sup>1</sup>Autor práce má podezření, že problém mohl být způsoben využitím emulátoru systému Android při vývoji mapové části namísto fyzického zařízení.

---

# Testování

Testování představuje nedílnou součástí vývoje software, jejímž účelem je zajištění toho, že výsledný software implementuje správně požadovanou funkcionálnitu a zároveň disponuje veškerými dalšími vlastnostmi vyplývajícími z požadavků zadavatele.

## 8.1 Testování software obecně

Testování software lze dělit na základě jeho povahy na dva základní typy – funkční a nefunkční testování. [41]

**Funkční testování** zkoumá funkční aspekty implementovaného software, tedy jestli funkcionalita odpovídá specifikaci daného software a zda poskytuje správné výsledky.

**Nefunkční testování:** zkoumá, zda systém splňuje na něj kladené nefunkční požadavky, jako jsou například požadavky na dostupnost, spolehlivost či bezpečnost výsledného systému.

Z hlediska přístupu lze následně rozdělit testování na tři typy: black box, white box a grey box. [42]

**Black box:** testování je prováděno bez znalosti vnitřní implementace daného systému.

**White box:** testování je založeno na znalosti interní implementace systému.

**Grey box:** testování kombinuje oba předešlé přístupy, kdy tester disponuje částečnou znalostí interní struktury systému.

Testování lze dále rovněž rozdělit do několika základních úrovní v závislosti na zaměření daného testu. Typicky jsou rozeznávány následující čtyři úrovně. [42]

**Jednotkové testy:** v rámci jednotkových testů je testována funkcionality jednotlivých jednotek či komponent systému nezávisle na ostatních jednotkách přítomných v systému.

**Integrační testy:** na této úrovni je testováno, zda jednotlivé komponenty systému fungují ve spojení s ostatními komponentami systému.

**Systémové testy:** v rámci systémových testů je testováno, zda systém jako celek funguje dle specifikovaných požadavků.

**Akceptační testy:** na této úrovni je zadavatelem testováno, zda systém odpovídá jeho požadavkům a zda je připraven pro nasazení do ostrého provozu.

### 8.2 Správcovská část systému

V rámci testování správcovské části na úrovni unit testů byly využity dva základní nástroje: PHPUnit a phpt skripty.

PHPUnit [43] představuje framework, který umožňuje vývojářům jednoduše definovat a provádět jednotkové testy a jedná se prakticky o jeden z nejvyužívanějších testovacích frameworků pro jazyk PHP.

S pomocí tohoto frameworku je pak otestována většina funkcionality systému (jednotlivá API umožňující CRUD operace, DAO objekty atd.).

Výjimkou je v tomto ohledu část funkcionality spojená s nahráváním souborů na server. V rámci aplikační logiky třídy UploadAPI totiž dochází k manipulaci s nahraným souborem pomocí zabudovaných funkcí PHP `move_uploaded_file` a `is_uploaded_file`. Tyto funkce totiž provádí kvůli bezpečnosti systému kontrolu, zda soubor, který je jim předán je skutečně souborem nahraným na server prostřednictvím HTTP požadavku, či zda-li se jedná o soubor, který byl na serveru již přítomen a s nímž by z bezpečnostních důvodů nemělo být manipulováno. Tato kontrola ovšem znamená, že v rámci jednotkových testů definovaných pomocí PHPUnit nelze pro testování nahrávání souborů na server využít přímé manipulace s magickou proměnou `FILES`, jelikož systém při následné kontrole vyhodnotí, že soubor ručně přidaný do proměnné `FILES` nepředstavuje nahraný soubor a tedy, že by s ním nemělo být manipulováno.

Z toho důvodu bylo nutné využití druhého nástroje v podobě phpt skriptů, které jsou využívány samotnými vývojáři jazyku PHP pro testování jeho funkcionality. Phpt skripty kromě samotného kódu definují i HTTP požadavek, který je následně systému předán před započítím vykonávání testovacího kódu. Díky tomu je možné simulovat nahrání souboru na server, který je následně testy zpracován již bez problému, neboť se jedná o soubor skutečně nahraný na server.

V případě testování nahrávání souboru u implementovaného systému, je pak tohoto využito k vytvoření dočasně nahraného souboru, jehož obsah



je před samotným provedením testovacího kódu nahrazen obsahem kýženého testovacího souboru. Tento postup je zvolen vzhledem k velikosti a binární povaze nahrávaných souborů, díky nimž není možné soubor definovat přímo v rámci phpt skriptu.

Kromě zmíněného jednotkového testování pak byla testována funkcionálnita systému rovněž ručně realizací jednotlivých případů užití specifikovaných v rámci kapitoly zabývající se analýzou požadavků na systém.

Veškeré testování pak probíhalo na zařízení s operačním systémem Windows a to s využitím nástroje XAMPP ve verzi 7.4.4.

## 8.3 Klientská část systému

V rámci testování klientské části systému bylo využito obdobného postupu jako v případě správcovské části, tedy využití jednotkových testů pro ověření správné funkcionality jednotek, u nichž není vyžadována interakce uživatele, a ruční testování vůči specifikovaným případům užití v případě funkcionality závislé na interakci uživatele.

Pro realizaci jednotkových testů je kromě klasického nástroje JUnit rovněž využita knihovna AndroidX Test, která výrazně usnadňuje a zrychluje provádění testů komponent, které pro svojí funkcionality vyžadují dostupnost aplikačního kontextu. V běžném případě by totiž takové testy bylo zapotřebí provádět v rámci emulátoru či fyzického zařízení, což je oproti klasickým jednotkovým testům výrazně pomalejší.

Kromě jednotkových a ručních testů provedených autorem pak rovněž proběhlo testování na malé skupině uživatelů. Na jeho základě bylo dodatečně upraveno zadávání zastávek u vyhledávání v jízdních řádech, kdy bylo zjištěno, že uživatelé překvapivě raději vyhledávají kýženou zastávku raději pomocí skrolování seznamu, než využitím vyhledávacího pole. V takovém případě pak bylo pro uživatele značně nepříjemné původní řazení zastávek v nabídce na základě množství z nich odjíždějících spojů, místo něž uživatelé očekávali klasické řazení podle abecedy, které bylo na základě těchto zjištění zapracováno do finální aplikace.

Uživatelské testování rovněž odhalilo odlišnou představu o způsobu fungování oblíbených dotazů. Původní myšlenkou bylo uložení daného dotazu s přesnými parametry, s jakými uživatel daný dotaz provedl s výjimkou data, jehož uložení by z uživatelského pohledu nedávalo příliš smysl. Uživatelé pak ovšem očekávali rovněž nezávislost uložení dotazu na původně zadaném času, tedy že při vyhledání uložení oblíbeného dotazu dojde k vyhledání na základě aktuálního času a nikoli času zadaného v rámci dotazu před jeho uložení do oblíbených položek.

### 8.4 Závěr

Ačkoliv v rámci testování klientské části bylo provedeno i uživatelské testování, bylo provedeno pouze na malé skupině uživatelů a celkově v rámci projektu převažuje spíše testování v podobě jednotkových a ručních testů prováděných autorem práce.

Jedním z prvotních zájmu autora do budoucnosti z hlediska testování je proto provedení testování na větší skupině uživatelů, na jehož základě by mohla být aplikace, její uživatelské rozhraní a funkcionality dále doladěna.

---

## Další rozvoj aplikace

Ačkoliv aplikace implementuje požadovanou funkcionalitu v podobě vyhledávání v jízdnicích řádech, ukládání oblíbených dotazů, zobrazování aktualit a online mapy dopravy, je rozhodně stále co vylepšovat.

Autor práce tak v této kapitole představí několik vybraných funkcionalit a úprav aplikace, jež by mohly být do budoucna v rámci aplikace implementovány a jež by mohly představovat přínos pro koncového uživatele a celkovou použitelnost aplikace.

### Historie vyhledávání

Historie vyhledávání je funkcionalita, která byla implementována u některých ze zkoumaných aplikací podobné té, jejíž vývojem se zabývá tato práce, a z autora pohledu by její implementace přinesla uživatelům další způsob, jak realizovat dotazy, které realizují často a které by se tudíž s největší pravděpodobností vyskytly i v historii vyhledávání ukládající několik dotazů.

Výhodou pro uživatele by pak bylo rozdělení historie dle jednotlivých typů dotazů, kdy by u jednotlivých obrazovek dotazů byla zobrazena historie pouze dotazů daného typu, čímž by se tato funkcionalita odlišovala od mechanismu oblíbených dotazů, který zobrazuje všechny typy dotazů nezávisle na jejich typu.

Zároveň by implementace takovéto funkcionality neměla být příliš náročná vzhledem k tomu, že mechanismus pro ukládání dotazů je již implementován v rámci funkcionality oblíbených položek.

### Pokročilé parametry vyhledávání

Pokročilé parametry rovněž představují funkcionalitu pozorovanou u obdobných aplikací a jako takové by byly pro uživatele rozhodně přínosem.

Užitečné by pokročilé parametry mohly být především u vyhledávání spojení, u kterého by pomocí nich bylo například možné omezit vyhledávání jen na přímá spojení či zakázat využití pěších přechodů mezi zastávkami (vhodné například pro uživatele s omezenou pohyblivostí).

Omezení vyhledávání odjezdů a spojení pouze na bezbariérové spoje, které v rámci pokročilých parametrů nabízí některé z obdobných aplikací, pak není v případě této aplikace zapotřebí a to díky skutečnosti, že ve vozovém parku DPMJ je momentálně jen několik zbývajících vozů, které nedisponují bezbariérovým vstupem, a které jsou využívány pouze jako náhradní v případě poruch a havárií. Zároveň by i tyto vozy měly být v létě kompletně nahrazeny novými bezbariérovými autobusy.

### Sběr statistik o užívání

Jedním z vylepšení systému, které by autor práce určitě do budoucna rád implementoval, je sběr informací o využívání aplikace koncovými uživateli, přičemž by byla sledována především frekvence s jakou uživatel využívá jednotlivé funkcionality systému.

Dále by bylo možné sledovat například informace o zařízení, na němž je aplikace nainstalována (například verze operačního systému).

Na základě takto získaných informací by pak mohly být založeny další úpravy aplikace. Například pokud by se ukázalo, že uživatelé využívají častěji vyhledávání odjezdů než aktuálních odjezdů, bylo by následně logické změnit pořadí, ve kterém jsou tyto dotazy uživateli prezentovány tak, aby vyhledávání odjezdů bylo zobrazeno dříve.

### Využití systémových upozornění

Výsledná aplikace umožňuje uživateli zobrazení aktualit o změnách v dopravě, přičemž při jejím spuštění dojde k načtení aktualit a pokud jsou dostupné nějaké, které uživatel ještě určitě nečetl (datum vydání aktuality je vyšší než poslední datum, kdy uživatel zobrazil aktuality), je uživatel na tuto skutečnost upozorněn zobrazeným počtem nových aktualit u ikony pro zobrazení aktualit. Uživatel tak musí nejdříve otevřít aplikaci, aby zjistil, že je nějaká nová změna v dopravě.

Tento nedostatek by bylo do budoucna vhodné napravit zobrazením systémového upozornění, které by informovalo uživatele o existenci nové aktuality.

Pomocí systémových notifikací by pak aplikace mohla upozornit uživatele rovněž na existenci novější verze některého z offline podkladů.

Takovouto funkcionalitu by bylo možné implementovat například s využitím Firebase Cloud Messaging (FCM) od společnosti Google a pluginu pro příslušný systém pro správu obsahu, který využítí takzvaných push notifikací využívajících FCM na daném CMS umožňuje.

---

## Možnost přizpůsobení pořadí oblíbených položek

Uživatelsky zajímavé vylepšení by mohlo představovat umožnění definice vlastního pořadí při zobrazení oblíbených položek.

Pro uživatele, který má uloženo jen několik málo oblíbených dotazů (cca 4 až 5), řazení nepředstavuje zas až takový problém, jelikož při běžné velikosti obrazovek dnešních mobilních zařízení budou prakticky všechny tyto dotazy zobrazeny najednou.

Při větším počtu uložených dotazů by však již mohla být možnost vlastního řazení pro uživatele velmi užitečná, jelikož by mu umožnila přesně určit, které položky chce zobrazit hned na začátku seznamu a pro jejichž dosažení nebude nucen skrolovat.

## Napojení aplikace na zamýšlený eshop

Posledním možným vylepšením, které zde bude uvedeno, je napojení na plánovaný eshop, díky němuž by bylo uživatelům umožněno zakoupení a zobrazení informací o časovém jízdě. Aplikace by rovněž díky napojení na eshop mohla uživatele předem varovat ohledně konce platnosti jím pořízeného časového jízdě.

Toto vylepšení je ovšem závislé na existenci zmíněného eshopu, jehož realizaci zatím DPMJ plánuje pouze výhledově, a z tohoto pohledu je tak zatím čistě hypotetické.



---

## Závěr

Cílem této práce bylo realizovat mobilní aplikaci pro Dopravní podnik města Jihlavy a.s., která by nahradila stávající aplikaci pro zobrazování a vyhledávání v jízdních řádech, a rozšířit její funkcionalitu zahrnutím funkcionality online mapy dopravy, možnosti zobrazení aktuálních odjezdů ze zastávky propojením se systémem chytrých zastávek či možnosti zobrazení aktualit o změnách v dopravě.

V práci pak bylo nejdříve přistoupeno k vypracování řešerše obdobných mobilních aplikací, v rámci níž bylo identifikováno několik zajímavých funkcionalit, které by mohly být do výsledné aplikace zahrnuty. Dále pak rovněž byly identifikovány nedostatky těchto aplikací zejména z pohledu uživatelského rozhraní, kterým by bylo záhodno se při implementaci aplikace vyhnout.

Následně autor přistoupil k porovnání jednotlivých možností vývoje aplikací pro mobilní zařízení (konkrétně responzivní webové aplikace, progresivní webové aplikace a nativní aplikace), kde autor zhodnotil výhody a nevýhody jednotlivých možností, na jejichž základě se rozhodl výslednou aplikaci realizovat v podobě nativní mobilní aplikace.

V následující kapitole jsou pak s využitím metod softwarového inženýrství identifikovány jednotlivé požadavky kladené na systém a pomocí případů užití je zachycena funkcionalita z pohledu uživatele systému. V rámci této kapitoly pak rovněž dochází k identifikování nutnosti rozdělení systému na dvě oddělené části – správcovskou část systému sloužící pro správu offline podkladů a klientskou část v podobě samotné mobilní aplikace určené pro koncové uživatele, která offline podklady poskytované správcovskou částí systému využívá v rámci své funkcionality.

Následně autor práce přistupuje k výběru vhodné technologie pro realizaci správcovské části, návrhu uživatelského rozhraní a samotnému návrhu a implementaci obou částí výsledného systému na základě získaných požadavků.

Implementovaný prototyp aplikace byl následně podroben testování s využitím jednotkových a ručních testů provedených autorem práce a v případě mobilní aplikace tvořící klientskou část systému bylo rovněž provedeno tes-

tování na malé skupině uživatelů, na jehož základě došlo k malým změnám uživatelského rozhraní a rovněž byl upraven způsob fungování oblíbených dotazů.

V závěrečné kapitole práce se pak autor zabývá možnými vylepšeními, které by bylo možné do budoucna v rámci aplikace implementovat.

V rámci práce pak bylo dosaženo původního cíle, kdy implementovaný prototyp aplikace implementuje potřebnou funkcionalitu a napravuje nedostatky původní aplikace JRm, jíž má v budoucnu nahradit.

Kvůli probíhajícímu projektu DPMJ, jehož účelem je vytvoření nového webového portálu, se ovšem autor práce rozhodl nepropojovat prototyp aplikace s aktuálním webovým portálem a to zejména vzhledem ke skutečnosti, že webový portál využívá značně zastaralý CMS Mambo. Ačkoliv je tedy funkcionality pro zobrazení aktualit v aplikaci implementována, je kvůli její demonstraci z tohoto důvodu prozatím napojena na jiný webový portál.

Kromě výsledného prototypu aplikace pak byla práce rovněž přínosem díky tomu, že v průběhu implementace a testování aplikace bylo autorem práce odhaleno několik chyb v definici jízdnic řádů v programu Grafikon, které díky tomu mohly být následně opraveny.



---

## Bibliografie

1. *Aplikace JRm* [online] [cit. 2020-05-11]. Dostupné z: <https://play.google.com/store/apps/details?id=jrm50.jrm>.
2. *Aplikace Jízdní řády IDOS* [online] [cit. 2020-05-11]. Dostupné z: <https://play.google.com/store/apps/details?id=cz.mafra.jizdnirady>.
3. *Aplikace Jízdní řády (Seznam.cz)* [online] [cit. 2020-05-11]. Dostupné z: <https://play.google.com/store/apps/details?id=cz.fhejl.pubtran>.
4. *Aplikace PID Lítačka* [online] [cit. 2020-05-11]. Dostupné z: <https://play.google.com/store/apps/details?id=cz.dpp.praguepublictransport>.
5. *Aplikace DPMBinfo* [online] [cit. 2020-05-11]. Dostupné z: <https://play.google.com/store/apps/details?id=cz.dpmb.dpmbinfo>.
6. *Aplikace DPMCB* [online] [cit. 2020-05-11]. Dostupné z: <https://play.google.com/store/apps/details?id=cz.dpmcb.dpmcb>.
7. *Aplikace Moje PMDP* [online] [cit. 2020-05-11]. Dostupné z: <https://play.google.com/store/apps/details?id=cz.programm.dpmp.mojepmdp>.
8. *Aplikace Moje DPO* [online] [cit. 2020-05-11]. Dostupné z: <https://play.google.com/store/apps/details?id=cz.dpo.app>.
9. SEBASTIEN. *Understanding apps: mobile, native or responsive* [online] [cit. 2020-02-26]. Dostupné z: <https://empireone.com.au/understanding-apps-mobile-native-or-responsive/>.
10. *Responsive design VS Mobile first* [online]. 2014 [cit. 2020-02-26]. Dostupné z: <https://www.imagro.nl/nl/actueel/responsive-design-vs-mobile-first/148>.
11. GATTERMAYER, Josef. *Proč a jak psát progresivní webové aplikace* [online]. 2017 [cit. 2020-02-27]. Dostupné z: <https://www.ackee.cz/blog/proc-a-jak-psat-progresivni-webove-aplikace/>.

12. *Introduction to progressive web apps* [online]. 2019 [cit. 2020-02-27]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Web/Progressive\\_web\\_apps/Introduction](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Introduction).
13. MUGHAL, zaffar. *Browsers support and limitation towards PWA* [online]. 2019 [cit. 2020-05-11]. Dostupné z: <https://medium.com/@zaffarabbasmughal/browsers-support-and-limitation-towards-pwa-2e6c58cd16d5>.
14. SANTONI, Muriel. *Progressive Web Apps browser support and compatibility* [online]. 2018 [cit. 2020-05-11]. Dostupné z: <https://www.goodbarber.com/blog/progressive-web-apps-browser-support-compatibility-a883/>.
15. *Esri offline-editor-js* [online] [cit. 2020-05-11]. Dostupné z: <https://github.com/Esri/offline-editor-js/blob/master/README.md>.
16. SACCOMANI, Pietro. *Native Apps, Web Apps or Hybrid Apps? What's the Difference?* [online] [cit. 2020-05-12]. Dostupné z: <https://www.mobiloud.com/blog/native-web-or-hybrid-apps/#5>.
17. SMARTAPPS. *Native application VS Progressive Web App: which one should you choose?* [online]. 2018 [cit. 2020-05-12]. Dostupné z: <https://medium.com/inside-smartapps/native-application-vs-progressive-web-app-which-one-should-you-choose-5eeaf6ee92d>.
18. ALVIN. *PWA vs Native App and How to Choose Between Them* [online]. 2018 [cit. 2020-05-12]. Dostupné z: <https://blog.magestore.com/pwa-vs-native-app/>.
19. GROUP, The Standish. *The CHAOS Report (1994)*. Dostupné také z: <http://www-cs-faculty.stanford.edu/~uno/abcde.html>.
20. PRESSMAN, Roger S. *Software Engineering: A Practitioner's Approach*. Clarendon Press, 2009. ISBN 0071267824. Dostupné také z: [http://dinus.ac.id/repository/docs/ajar/RPL-7th\\_ed\\_software\\_engineering\\_a\\_practitioners\\_approach\\_by\\_roger\\_s.\\_pressman.pdf](http://dinus.ac.id/repository/docs/ajar/RPL-7th_ed_software_engineering_a_practitioners_approach_by_roger_s._pressman.pdf).
21. COCKBURN, Alistair. *WRITING EFFECTIVE USE CASES* [online]. 2001 [cit. 2018-05-01]. ISBN 02-017-0225-8. Dostupné z: <http://alistair.cockburn.us/get/2465>.
22. *C# Language Specification* [online]. 2017 [cit. 2020-05-16]. Dostupné z: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-334.pdf>.
23. *Developer Survey Results 2019* [online] [cit. 2020-05-16]. Dostupné z: <https://insights.stackoverflow.com/survey/2019#technology>.
24. *Usage statistics of Java for websites* [online] [cit. 2020-05-16]. Dostupné z: <https://w3techs.com/technologies/details/pl-java>.

25. *Usage statistics of server-side programming languages for websites* [online] [cit. 2020-05-16]. Dostupné z: [https://w3techs.com/technologies/overview/programming\\_language](https://w3techs.com/technologies/overview/programming_language).
26. *TIOBE Index for May 2020* [online] [cit. 2020-05-16]. Dostupné z: <https://www.tiobe.com/tiobe-index/>.
27. *PYPL PopularitY of Programming Language* [online] [cit. 2020-05-16]. Dostupné z: <http://pypl.github.io/PYPL.html>.
28. *Design Guidelines* [online] [cit. 2020-05-17]. Dostupné z: <https://www.interaction-design.org/literature/topics/design-guidelines>.
29. NIELSEN, Jakob. *10 Usability Heuristics for User Interface Design* [online]. 1994 [cit. 2020-05-17]. Dostupné z: <https://www.nngroup.com/articles/ten-usability-heuristics/>.
30. *Coolers.do* [online] [cit. 2020-05-17]. Dostupné z: <https://coolers.co/024da8-483d3f-c9f0ff-fffcf9>.
31. *Contrast-ratio.com* [online] [cit. 2020-05-17]. Dostupné z: <https://contrast-ratio.com/>.
32. W3C. *Web Content Accessibility Guidelines (WCAG) 2.1* [online]. 2018 [cit. 2020-05-17]. Dostupné z: <https://www.w3.org/TR/WCAG/#contrast-enhanced>.
33. *Google kalendář České státní svátky* [online] [cit. 2020-05-27]. Dostupné z: <https://calendar.google.com/calendar/embed?src=cs.czech%23holiday%40group.v.calendar.google.com&ctz=Europe%2FPrague>.
34. *jQuery* [online] [cit. 2020-05-27]. Dostupné z: <https://jquery.com>.
35. *jQueryUI* [online] [cit. 2020-05-27]. Dostupné z: <https://jqueryui.com>.
36. *Select2* [online] [cit. 2020-05-27]. Dostupné z: <https://select2.org>.
37. *Notify.js* [online] [cit. 2020-05-27]. Dostupné z: <https://notifyjs.jpillora.com>.
38. *jQuery ValidationPlugin* [online] [cit. 2020-05-27]. Dostupné z: <https://jqueryvalidation.org>.
39. *Bootstrap* [online] [cit. 2020-05-27]. Dostupné z: <https://getbootstrap.com>.
40. *RSS-Parser* [online] [cit. 2020-05-20]. Dostupné z: <https://github.com/prof18/RSS-Parser>.
41. HOSSAIN, Akhtar. *Types of Software Testing* [online]. 2020 [cit. 2020-05-21]. Dostupné z: <https://hackr.io/blog/types-of-software-testing>.
42. RAJKUMAR. *What Is Software Testing – Definition, Types, Methods, Approaches* [online]. 2020 [cit. 2020-05-21]. Dostupné z: <https://www.softwaretestingmaterial.com/software-testing/>.

## BIBLIOGRAFIE

---

43. *PHPUnit* [online] [cit. 2020-05-27]. Dostupné z: <https://phpunit.de>.

## Seznam použitých zkratk

**CMS** Content Management System

**DPMJ** Dopravní podnik města Jihlavy a.s

**HTTP** Hypertext Transfer Protocol

**PWA** Progressive Web Application

**SaaS** Software as a Service

**SRS** Software Requirments Specification



## Přílohy

### B.1 Případy užití

#### B.1.1 Serverová část

##### U1.0.1 Přihlášení

**Předpoklady:** Uživateli byl správcem vytvořen účet a předány přihlašovací údaje.

**Cíl:** Zamezit neautorizovanému přístupu do systému.

**Hlavní scénář:**

1. Uživatel přistoupí na přihlašovací obrazovku aplikace
2. Uživatel vyplní požadované informace tj. jemu přidělené uživatelské jméno a heslo.
3. Jestliže jsou údaje správné, je uživateli umožněn vstup do systému.

**Alternativní scénáře:**

- 3.a Nejsou-li údaje poskytnuté uživatelem správné, systém tuto skutečnost prezentuje.

##### U1.0.2 Změna vlastního hesla

**Předpoklady:** Uživateli je přihlášen do systému.

**Cíl:** Uživateli bude umožněn vstup do systému s použitím nově zvoleného hesla.

- Uživatel - chce změnit jím používané heslo pro přístup do systému

**Hlavní scénář:**

1. Uživatel přistoupí na obrazovku uživatelského profilu
2. Uživatel vyplní nové heslo do dvou různých polí, kvůli jeho kontrole.
3. Jestliže jsou údaje v obou polích shodné, je heslo uživatele změněno.

### **Alternativní scénáře:**

- 3.a Nejsou-li údaje v polích shodné, je uživatel požádán o jejich opětovné zadání.

### **U1.1.1**

Přidání nového uživatele

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s uživateli.

**Cíl:** V systému bude zaevidován nový uživatel.

### **Hlavní scénář:**

1. Uživatel přejde na obrazovku pro správu uživatelů.
2. Uživatel vyplní požadované údaje – uživatelské jméno, e-mail a oprávnění nově vytvářeného uživatele.
3. Pokud jsou zadané údaje v pořádku, uloží systém záznam o uživateli.

### **Alternativní scénáře:**

- 3.a Pokud již existuje uživatel s daným uživatelským jménem, uživatel nebude vytvořen a systém tuto skutečnost indikuje.
- 3.b Pokud již existuje uživatel s danou e-mailovou adresou, uživatel nebude vytvořen a systém tuto skutečnost indikuje.

### **U1.1.2**

Změna údajů uživatele

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s uživateli v případě úpravy údajů jiného uživatele. Případně je přihlášen uživatel, jehož údaje jsou upravovány.

**Cíl:** V systému budou evidovány pozměněné údaje uživatele.

### **Hlavní scénář:**

1. Uživatel přejde na obrazovku pro úpravu údajů uživatele.
2. Uživatel vyplní kýžené údaje, které chce změnit (e-mail, heslo).
3. Systém uloží změny do databáze.



### U1.1.3

Změna přístupových práv uživatele

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s uživateli.

**Cíl:** Daný uživatel bude v rámci systému disponovat nově zadanými oprávněními.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku pro úpravu údajů uživatele.
2. Uživatel vybere příslušná oprávnění.
3. Systém přiřadí uživateli zvolená oprávnění.

### U1.1.4

Odstranění uživatele

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s uživateli.

**Cíl:** V systému nadále nebude evidován zvolený uživatel.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku pro správu uživatelů.
2. Uživatel smaže uživatele kliknutím na příslušné tlačítko.
3. Systém odstraní uživatele z evidence.

### U1.2.1

Nahrání nové verze jízdních řádů

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s jízdními řády.

**Cíl:** Do systému bude nahrána nová definice jízdních řádů.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku pro správu jízdních řádů.
2. Uživatel vybere soubor s definicí jízdních řádů, který má být nahrán na server, a datum vstupu v platnost vložené definice.
3. Pokud jsou zadané údaje v pořádku, systém nahraje soubor s definicí na server a zaeviduje o něm záznam.

**Alternativní scénáře:**

- 3.a Pokud na serveru již existuje definice jízdnicích řádů se stejným datem platnosti, systém soubor nezaeviduje a indikuje tuto skutečnost uživateli.
- 3.b Pokud uživatel zadal neplatný typ souboru, systém tuto skutečnost indikuje a daný soubor nezaeviduje.

### U1.2.2

Odstranění verze jízdnicích řádů

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s jízdnicími řády.

**Cíl:** V systému již nebude evidována daná definice jízdnicích řádů.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku pro správu jízdnicích řádů.
2. Uživatel smaže definici jízdnicích řádů kliknutím na příslušné tlačítko.
3. Systém odstaní danou definici z evidence.

### U1.2.3

Úprava data vstoupení v platnost dané verze jízdnicích řádů

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s jízdnicími řády.

**Cíl:** U dané definice jízdnicích řádů bude evidováno nově zadané datum vstupu v platnost.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku pro správu jízdnicích řádů.
2. Uživatel přejde do editačního módu daného záznamu.
3. Uživatel zadá nové datum vstupu v platnost.
4. Systém zaeviduje nově zadané datum vstupu v platnost.

**Alternativní scénáře:**

- 3.a Pokud již existuje záznam se stejným datem vstupu v platnost, systém změnu nezaeviduje a tuto skutečnost indikuje.

#### U1.2.4

Nahrání nové verze definice svátků

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s jízdními řády.

**Cíl:** Do systému bude nahrán soubor s novou definicí svátků.

Průběh je obdobou U1.2.1.

#### U1.2.5

Odstranění verze definice svátků

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s jízdními řády.

**Cíl:** V systému nebude nadále evidovaná daná definice svátků.

Průběh je obdobou U1.2.2.

#### U1.2.6

Úprava data vstoupení v platnost dané verze definice svátků

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s jízdními řády.

**Cíl:** U dané definice svátků bude evidováno nově zadané datum vstupu v platnost.

Průběh je obdobou U1.2.3.

#### U1.2.7

Přidání jména linky

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s jízdními řády.

**Cíl:** V systému bude evidováno nově zadané jméno linky.

#### Hlavní scénář:

1. Uživatel přistoupí na obrazovku pro správu názvů linek.
2. Uživatel zadá ID linky a její název.
3. Systém zaeviduje zadané jméno linky a rovněž zaeviduje název linky ve všech nadcházejících definicích jízdních řádů.

**Alternativní scénáře:**

- 3.a Pokud již existuje záznam s daným ID linky, systém záznam nezaeviduje.
- 3.b Pokud již existuje záznam s daným názvem linky, systém záznam nezaeviduje.

**U1.2.8**

Odstranění jména linky

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s jízdními řády.

**Cíl:** V systému nebude nadále evidován záznam o daném jménu linky.

**Hlavní scénář:**

1. Uživatel přistoupí na obrazovku pro správu názvů linek.
2. Uživatel smaže daný záznam kliknutím na příslušné tlačítko.
3. Systém vyřadí příslušnou položku z evidence.

**Alternativní scénáře:**

- 3.a Pokud je daná linka přítomna v libovolné z nadcházejících definic jízdních řádů, záznam o jméně linky nebude smazán a tato skutečnost je indikována uživateli.

**U1.2.9**

Úprava jména linky

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s jízdními řády.

**Cíl:** V systému bude u záznamu o jméně linky evidován nově zadaný název linky.

**Hlavní scénář:**

1. Uživatel přistoupí na obrazovku pro správu názvů linek.
2. Uživatel vstoupí do editačního módu dané položky.
3. Uživatel zadá nový název linky.
4. Systém zaeviduje nově zadaný název linky a propaguje změnu do všech nadcházejících definic jízdních řádů.

**Alternativní scénáře:**

- 3.a Pokud již existuje záznam s daným názvem linky, nebude systémem úprava názvu linky zaevidována.

### U1.2.10

Přidání pěšího spojení

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s jízdními řády.

**Cíl:** V systému bude evidováno nové pěší spojení.

#### Hlavní scénář:

1. Uživatel přejde na obrazovku pro správu pěších spojení.
2. Uživatel vyplní počáteční a koncovou zastávku pěšího spojení a rovněž čas potřebný k přechodu mezi vybranými zastávkami.
3. Systém zaeviduje dané pěší spojení a vloží jej do všech nadcházejících definic jízdních řádů.

#### Alternativní scénáře:

- 3.a Pokud již existuje spojení mezi danými zastávkami, nebude nový záznam vytvořen.

**Poznámka:** Pěší spojení je chápáno obousměrně, není tedy nutné zadávat záznam o pěším spojení v opačném směru.

### U1.2.11

Odstranění pěšího spojení

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s jízdními řády.

**Cíl:** V systému nebude nadále evidován záznam o daném pěším spojení.

#### Hlavní scénář:

1. Uživatel přejde na obrazovku pro správu pěších spojení.
2. Uživatel smaže daný záznam kliknutím na příslušné tlačítko.
3. Systém vyřadí příslušnou položku z evidence.
4. Systém smaže záznam o pěším spojení ze všech nadcházejících definic jízdních řádů.

### U1.2.12

Úprava pěšího spojení

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s jízdními řády.

**Cíl:** U pěšího spojení mezi danými zastávkami bude evidován nový čas potřebný k přechodu.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku pro správu pěších spojení.
2. Uživatel vstoupí do editačního módu dané položky.
3. Uživatel vloží novou hodnotu času potřebného pro přechod.
4. Systém zaeviduje nově vloženou hodnotu a aktualizuje ji u všech nadcházejících definic jízdních řádů.

### U1.3.1

Nahrání nové verze mapových podkladů

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s mapovými podklady.

**Cíl:** V systému bude evidována nová verze mapových podkladů.

Průběh je obdobou U1.2.1.

### U1.3.2

Odstranění verze mapových podkladů

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s mapovými podklady.

**Cíl:** V systému nebude nadále evidována daná verze mapových podkladů.

Průběh je obdobou U1.2.2.

### U1.3.3

Úprava data vstoupení v platnost dané verze mapových podkladů

**Předpoklady:** Je přihlášen uživatel s oprávněním pro manipulaci s mapovými podklady.

**Cíl:** U dané verze mapových podkladů bude evidováno nové datum vstupu v platnost.

Průběh je obdobou U1.2.3.

## **B.1.2 Klientská část**

### **B.1.2.1 Podčást jízdní řády**

#### **U2.0.1**

Zobrazení aktuálních odjezdů

**Předpoklady:** Zařízení je připojeno k internetu.

**Cíl:** Systém zobrazí seznam aktuálních odjezdů pro danou zastávku.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku pro vyhledání aktuálních odjezdů.
2. Uživatel zadá zastávku, pro níž mají být zobrazeny aktuální odjezdy.
3. Systém zobrazí aktuální odjezdy.

#### **U2.0.2**

Vyhledání odjezdů

**Předpoklady:** V aplikaci je přítomna definice jízdních řádů a svátků.

**Cíl:** Systém zobrazí seznam odjezdů pro danou zastávku.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku pro vyhledání odjezdů.
2. Uživatel zadá čas, datum, zastávku a volitelně linku, pro níž mají být zobrazeny odjezdy.
3. Systém zobrazí odjezdy dle zadaných parametrů.

#### **U2.0.3**

Vyhledání spojení

**Předpoklady:** V aplikaci je přítomna definice jízdních řádů a svátků.

**Cíl:** Systém zobrazí seznam spojení mezi danými zastávkami.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku pro vyhledání spojení.
2. Uživatel zadá počáteční a cílovou zastávku společně s časem a datem.
3. Systém zobrazí spojení dle zadaných parametrů.

#### **U2.0.4**

Zobrazení oblíbených položek

**Předpoklady:** –

**Cíl:** Systém zobrazí seznam oblíbených položek.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku pro zobrazení oblíbených položek.
2. Systém zobrazí seznam uložených oblíbených položek.

#### **U2.0.5**

Přidání oblíbené položky

**Předpoklady:** –

**Cíl:** Systém uloží daný dotaz na jízdní řády mezi oblíbené položky.

**Hlavní scénář:**

1. Uživatel realizuje dotaz na jízdní řády (vyhledání aktuálních odjezdů, odjezdů či spojení).
2. Systém zobrazí obrazovku s výsledky daného dotazu.
3. Uživatel přidá dotaz do oblíbených kliknutím na příslušné tlačítko.
4. Systém uloží dotaz s danými parametry mezi oblíbené položky.

**Alternativní scénáře:**

- 4.a Pokud již byla již daná položka uložena jako oblíbená, systém tuto skutečnost indikuje uživateli.

**Poznámka:** Uložit je možné dotaz na aktuální odjezdy, odjezdy a spojení.

#### **U2.0.6**

Odstranění oblíbené položky

**Předpoklady:** –

**Cíl:** V seznamu oblíbených dotazů nebude nadále evidována vybraná položka.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku pro zobrazení oblíbených položek.
2. Uživatel odstraní položku stisknutím příslušného tlačítka.
3. Systém odstraní danou položku ze seznamu oblíbených položek.



### U2.0.7

Aktualizace offline podkladů pro jízdní řády

**Předpoklady:** Je dostupná novější verze definice jízdních řádů.

**Cíl:** V systému bude k dispozici aktuální definice jízdních řádů.

**Hlavní scénář:**

1. Systém vyzve uživatele ke stažení nové definice jízdních řádů.
2. Uživatel potvrdí stažení nové definice jízdních řádů.
3. Systém stáhne novou definici jízdních řádů a nahradí jí tu stávající.
4. Systém umožní přístup k funkcionalitě, která definici vyžaduje.

### U2.0.8

Aktualizace offline definice svátků

**Předpoklady:** Je dostupná novější verze definice svátků.

**Cíl:** V systému bude k dispozici aktuální definice svátků.

Průběh je obdobou U2.0.7

### U2.0.9

Zobrazení oblíbeného dotazu

**Předpoklady:** –

**Cíl:** Systém zobrazí výsledky vyhledávání pro daný oblíbený dotaz.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku s oblíbenými dotazy.
2. Uživatel vybere kýžený oblíbený dotaz.
3. Systém vyhledá a zobrazí výsledky na základě typu a parametrů zvoleného oblíbeného dotazu.

## B.1.2.2 Podčást online mapa dopravy

### U2.1.1

Zobrazení detailu objektu na mapě

**Předpoklady:** –

**Cíl:** Systém zobrazí detailní informace o vybraném objektu.

**Hlavní scénář:**

1. Uživatel vybere objekt kliknutím do mapy.
2. Systém zobrazí obrazovku s detailními informacemi o objektu.

**Alternativní scénáře:**

- 2.a.1 Pokud je v blízkosti místa kliknutí více než jeden objekt, systém zobrazí seznam pro výběr z těchto objektů.
- 2.a.2 Uživatel vybere objekt ze zobrazeného seznamu.
- 2.a.3 Systém zobrazí obrazovku s detailními informacemi o objektu

**U2.1.2**

Zobrazení legendy/informací o mapě

**Předpoklady:** –

**Cíl:** Systém zobrazí legendu/informace o mapě.

**Hlavní scénář:**

1. Uživatel na obrazovce s online mapou dopravy klikne na tlačítko pro zobrazení informací.
2. Systém zobrazí legendu/informace o mapě.

**U2.1.3**

Filtrování linek

**Předpoklady:** Úspěšně načtené linky v mapě.

**Cíl:** Systém zobrazí na mapě pouze vybrané linky a vozidla těchto linek.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku filtru linek.
2. Uživatel vybere kýžené linky a výběr potvrdí.
3. Systém zobrazí na mapě pouze uživatelem vybrané linky a vozy těchto linek.

#### U2.1.4

Nastavení map do online režimu

**Předpoklady:** –

**Cíl:** Systém bude při zobrazení mapy využívat online mapové podklady.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku nastavení.
2. Uživatel přepne přepínač pro využití offline mapových podkladů do polohy vypnuto.
3. Pokud jsou na zařízení přítomny offline mapové podklady, nabídne systém uživateli možnost jejich smazání.
4. Uživatel potvrdí smazání offline mapových podkladů.
5. Systém smaže offline mapové podklady a při zobrazení online mapy dopravy bude využívat online mapové podklady.

**Alternativní scénáře:**

- 4.a.1 Uživatel odmítne smazání offline mapových podkladů.
- 4.a.2 Systém uchová mapové podklady, které budou moci být využity při opětovném přepnutí do offline režimu.

#### U2.1.5

Nastavení map do offline režimu

**Předpoklady:** –

**Cíl:** Systém bude při zobrazení mapy využívat online mapové podklady.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku nastavení.
2. Uživatel přepne přepínač pro využití offline mapových podkladů do polohy zapnuto.
3. Systém bude nadále při zobrazení online mapy dopravy bude využívat offline mapové podklady.

#### U2.1.6

Aktualizace offline mapových podkladů

**Předpoklady:** Je dostupná novější verze mapových podkladů.

**Cíl:** V systému bude k dispozici aktuální mapové podklady.

Průběh je obdobou U2.0.7

### B.1.2.3 Podčást aktuality

#### U2.2.1

Zobrazení aktuality

**Předpoklady:** Zařízení je připojeno k internetu.

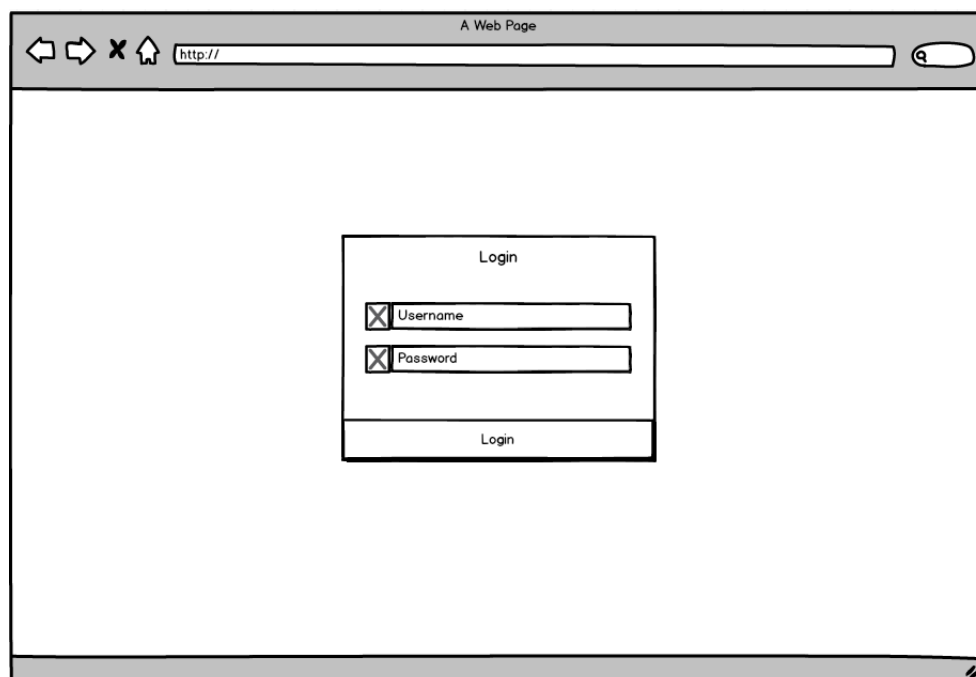
**Cíl:** Systém zobrazí danou aktualitu.

**Hlavní scénář:**

1. Uživatel přejde na obrazovku s aktualitami.
2. Systém zobrazí obrazovku se seznamem aktualit.
3. Uživatel vybere kýženou aktualitu.
4. Systém zobrazí vybranou aktualitu.

## B.2 Drátěné modely

### B.2.1 Správcovská část



## B.2. Drátěné modely

The screenshot shows a web browser window with the address bar containing "http://". The page title is "A Web Page". The application header is "DPMJInfo" with "Profile" and "Logout" buttons. A sidebar on the left contains a menu with items: "Jízdní řády", "> Jízdní řády", "> Názvy linek", "> Pěší spoje", "> Svátky", "Mapy", and "Uživatelé". The main content area has "Export CIS:" with a "Vybrat soubor" button and "Datum platnosti:" with a text input field containing "2020-03-15" and a "Náhrát soubor" button. Below this is a table with the following data:

ID	Název	Datum nahrání	Datum platnosti		
127	CIS.zip	2020-03-15	2020-03-19		
127	CIS.zip	2020-03-15	2020-03-19		
127	CIS.zip	2020-03-15	2020-03-19		
127	CIS.zip	2020-03-15	2020-03-19		

At the bottom right of the table area, it says "Zobrazují prvky 1.4 z 10" with navigation arrows.

This screenshot is identical to the one above, but the first row of the table has been updated. The "Datum platnosti" field for the first row now contains "2020-03-20" and is highlighted with a text input box. Additionally, the edit and delete icons for this row have been replaced with a checkmark and an 'X' icon, respectively, indicating a successful update.

ID	Název	Datum nahrání	Datum platnosti		
127	CIS.zip	2020-03-15	2020-03-20		
127	CIS.zip	2020-03-15	2020-03-19		
127	CIS.zip	2020-03-15	2020-03-19		
127	CIS.zip	2020-03-15	2020-03-19		

At the bottom right of the table area, it says "Zobrazují prvky 1.4 z 10" with navigation arrows.

## B. PŘÍLOHY

The screenshot shows a web browser window titled 'A Web Page' with a search bar containing 'http/'. The page content is for 'DPMJInfo' and includes a 'Profile' and 'Logout' button. A left sidebar contains navigation links: 'Jízdní řády', '> Jízdní řády', '> Názvy linek', '> Pěší spoje', '> Svátky', 'Mapy', and 'Uživatelé'. The main area has two input fields: 'Kód linky' with value '103' and 'Název linky' with value '3', and a 'Přidat linku' button. Below is a table with columns 'Kód linky' and 'Název':

Kód linky	Název		
201	A		
202	B		
203	C		
204	D		

At the bottom right, it says 'Zobrazují prvky 1.4 z 10' with navigation arrows.

This screenshot is identical to the one above, but the 'Název' field in the first row of the table (code 201) is highlighted in grey, and the value '5' is entered into it. The right-side icons for this row are a checkmark and an 'X' instead of edit and delete icons. The rest of the interface remains the same.

## B.2. Drátěné modely

A Web Page

http://

DPMJInfo Profile Logout

**Jízdní řády**

- > Jízdní řády
- > Názvy linek
- > **Pěší spoje**
- > Svátky
- Mapy
- Uživatelé

Ze:

Do zastávky:

Cas (min):

Ze zastávky	Do zastávky	Cas		
Brtnická	Chlumova	3		
Brtnická	Chlumova	3		
Brtnická	Chlumova	3		
Brtnická	Chlumova	3		

Zobrazují prvky 1.4 z 10

A Web Page

http://

DPMJInfo Profile Logout

**Jízdní řády**

- > Jízdní řády
- > Názvy linek
- > **Pěší spoje**
- > Svátky
- Mapy
- Uživatelé

Ze:

Do zastávky:

Cas (min):

Ze zastávky	Do zastávky	Cas		
Brtnická	Chlumova	<input type="text" value="5"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Brtnická	Chlumova	3		
Brtnická	Chlumova	3		
Brtnická	Chlumova	3		

Zobrazují prvky 1.4 z 10

## B. PŘÍLOHY

A Web Page

http://

DPMJInfo Profile Logout

**Jízdní řády**

> Jízdní řády

> Názvy linek

> Pěší spoje

**> Svátky**

Mapy

Uživatelé

Soubor se svátky:

Datum platnosti:

ID	Název	Datum nahrání	Datum platnosti		
127	CIS.zip	2020-03-15	2020-03-19		
127	CIS.zip	2020-03-15	2020-03-19		
127	CIS.zip	2020-03-15	2020-03-19		
127	CIS.zip	2020-03-15	2020-03-19		

Zobrazují prvky 1.4 z 10

A Web Page

http://

DPMJInfo Profile Logout

Jízdní řády

> Jízdní řády

> Názvy linek

> Pěší spoje

> Svátky

**Mapy**

Uživatelé

Soubor s mapou:

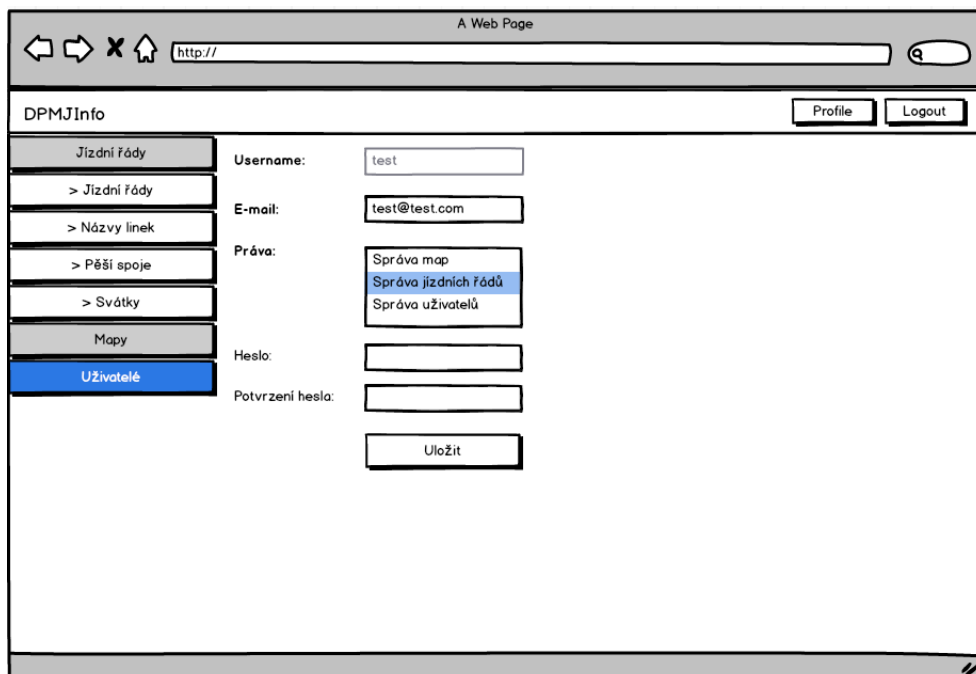
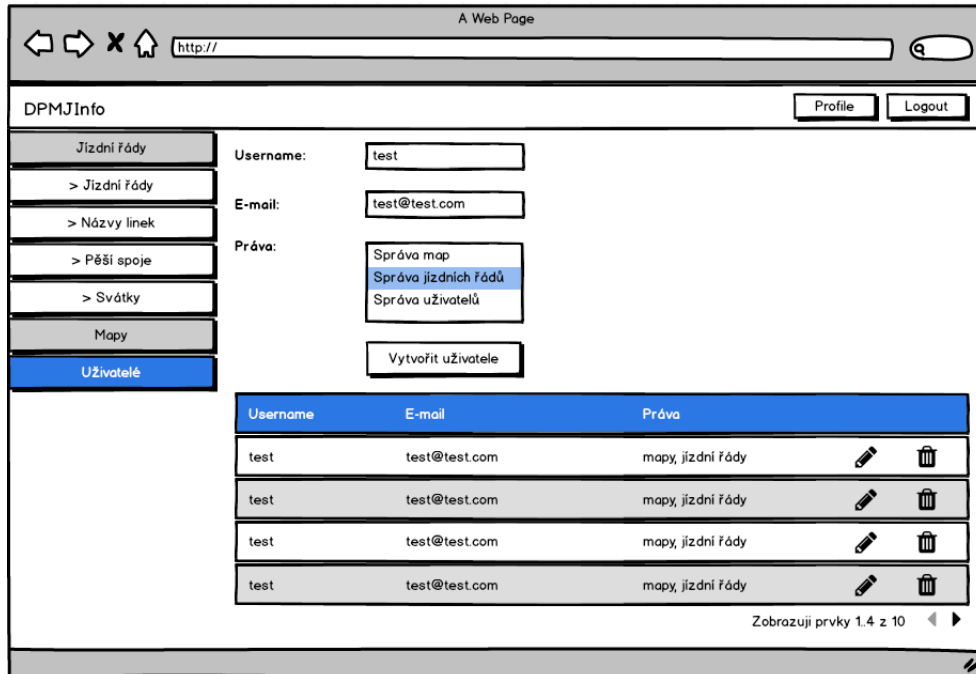
Datum platnosti:

ID	Název	Datum nahrání	Datum platnosti		
127	CIS.zip	2020-03-15	2020-03-19		
127	CIS.zip	2020-03-15	2020-03-19		
127	CIS.zip	2020-03-15	2020-03-19		
127	CIS.zip	2020-03-15	2020-03-19		

Zobrazují prvky 1.4 z 10



## B.2. Drátěné modely

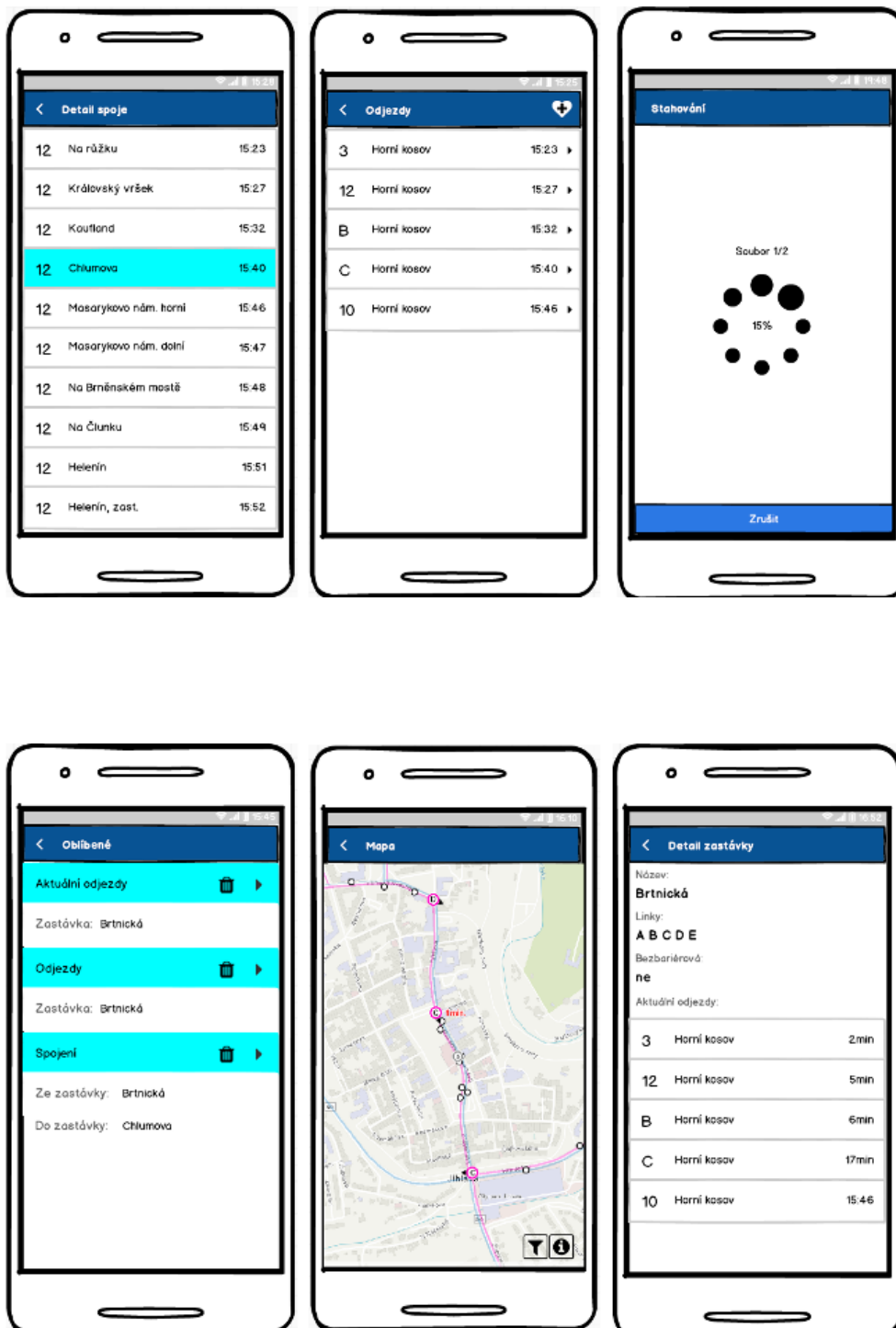


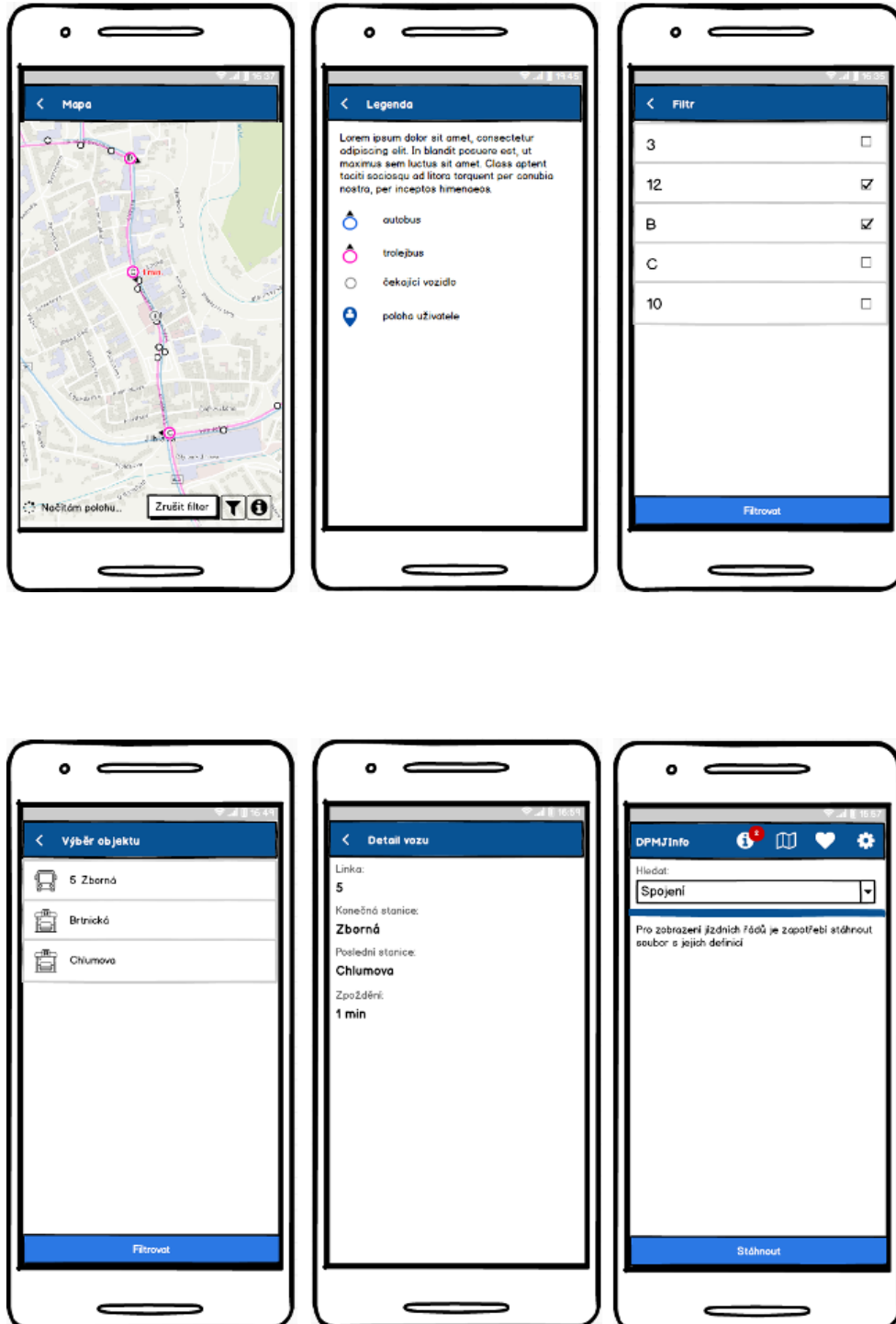
The image shows a web browser window titled "A Web Page". The address bar contains "http://". The page content is titled "DPMJInfo" and includes a "Profile" button and a "Logout" button in the top right corner. On the left side, there is a vertical sidebar menu with the following items: "Jízdní řády", "> Jízdní řády", "> Názvy linek", "> Pěší spoje", "> Svátky", "Mapy", and "Uživatelé" (highlighted in blue). The main content area contains a registration form with the following fields and labels: "Username:" with the value "test", "E-mail:" with the value "test@test.com", "Heslo:" (password), and "Potvrzení hesla:" (confirm password). There is an "Uložit" (Save) button at the bottom of the form.

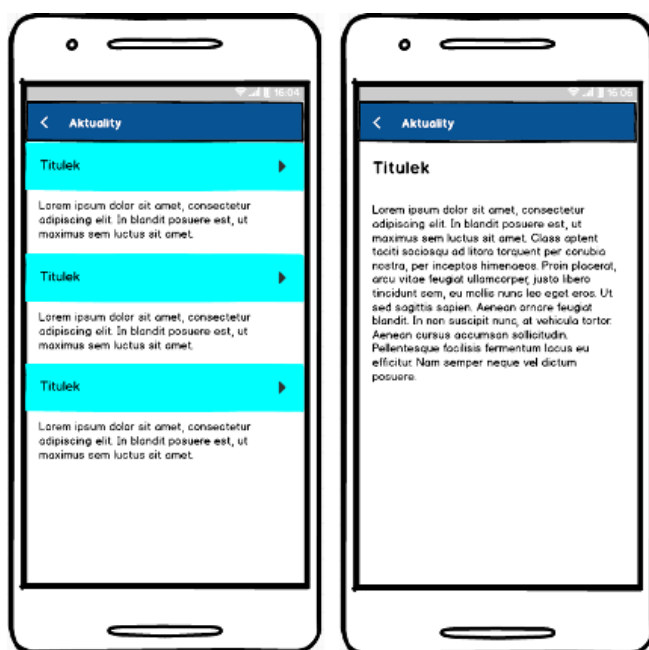
## B.2.2 Klientská část



## B. PŘÍLOHY







---

## Obsah přiloženého DVD

readme.txt.....	stručný popis obsahu DVD
src	
├─ dpmjinfoserver.....	zdrojové kódy serverové části aplikace
├─ DPMJInfo.....	zdrojové kódy klientské části aplikace
├─ thesis.....	zdrojová forma práce ve formátu $\text{\LaTeX}$
text.....	text práce
├─ thesis.pdf.....	text práce ve formátu PDF