

Dlouhodobá maturitní práce



Střední **průmyslová** škola
na Proseku

18-20-M/01 Informační technologie

Veřejná doprava v ČR – informace v mapě

Richard Mlejnek

Vedoucí: Vaculík Přemysl
Zaměření: Vývoj aplikací
2024

ZADÁNÍ PRAKTICKÉ MATURITNÍ PRÁCE

Žák: Richard Mlejnek
 Obor: 18-20-M/01 Informační technologie
 Školní rok: 2023/2024
 Téma práce: Vývoj webové aplikace
 Název práce: Veřejná doprava v ČR - informace v mapě
 Ved. práce: Ing. Přemysl Vaculík
 Oponent: bude jmenován ředitelem školy v souladu s vyhláškou 177/2009 Sb.

Termín odevzdání – řádný termín: **1. 3. 2024 do 12:00 hodin**
 Termíny odevzdání – opravné termíny: podzimní opravný termín: nejpozději poslední pracovní den v červnu do 12:00 hodin
 jarní opravný termín: nejpozději první pracovní den v březnu do 12:00 hodin

Délka obhajoby maturitní práce před maturitní komisí: 15 minut včetně doplňujících otázek

Poučení: Dle vyhlášky 177/2009 Sb. § 15 odst. 7 – Neodevzdá-li žák pro vážné důvody práci v termínu stanoveném podle odstavce 1 písm. b), omluví se písemně řediteli školy nejpozději v den stanovený pro odevzdání maturitní práce; uzná-li ředitel školy omluvu žáka, určí žákovi náhradní termín pro odevzdání maturitní práce. Pokud žák maturitní práci neodevzdá v termínu podle odstavce 1 písm. b) bez písemné omluvy s uvedením vážných důvodů nebo pokud mu omluva nebyla uznána, posuzuje se, jako by danou zkoušku vykonal neúspěšně.

Dle školského zákona č. 561/2004 Sb. § 79 odst. 7 - Profilová část maturitní zkoušky je veřejná s výjimkou zkoušek konaných formou písemné zkoušky a jednání zkušební maturitní komise o hodnocení žáka; zkoušky konané formou praktické zkoušky jsou neveřejné v případech, kdy je to nutné z důvodu ochrany zdraví, bezpečnosti práce a u zdravotnických oborů také z důvodu ochrany soukromí pacienta.

Pokyny pro vypracování:

- vypracování rešerší pro seznámení s řešenou tématikou V práci musí být vypracovány rešerše v takovém rozsahu, aby byla odůvodněna každá část návrhu od volby koncepce řešení až po volbu jednotlivých komponent, či návrh programu. Zároveň práce nesmí obsahovat rešerše nadbytečné, které s návrhem nesouvisí. Minimální počet rešerší jsou dvě.
- vytvoření technické dokumentace (technické řešení, požadavky, postup instalace, ...)
- vytvoření uživatelské dokumentace (ukázka použití, manuály, ...)
- plakát ve formátu A1 prezentující práci
- prezentace pro obhajobu práce

Požadavky:

- vypracování jednoduchého průzkumu trhu a stanovení základní požadavků (funkce, využití, ...), pokud je to možné a daná práce toto umožňuje
- automatické zpracování open dat
- vytvoření webové aplikace s mapovými podklady a informace o veřejné dopravě
- vytvoření a nasazení demo verze
- vytvoření API v Pythonu

Hodnocení:

- výsledná známka z maturitního projektu s obhajobou se skládá z hodnocení:
 - hodnocení v závěrečném posudku vedoucího maturitního projektu
 - hodnocení v závěrečném posudku oponenta maturitního projektu
 - hodnocení obhajoby maturitního projektu před maturitní komisí

Hodnocení práce – plagiátorství:

Odevzdané textové části práce budou posouzeny systémem na kontrolu plagiátů odevzdej.cz. V případě míry shody přesahující 15 % bude práce posouzena předmětovou komisí a výsledek posouzení bude poté předán k rozhodnutí maturitní komisí. Pokud se ukáže při hodnocení práce, či při samotné obhajobě, že je práce plagiátem, maturitní komise rozhodne, že práce bude hodnocena známkou nedostatečný.

Kritéria hodnocení maturitního projektu:

- samostatný a tvůrčí přístup k práci
- dodržování stanovených termínů
- prezentace dosažených výsledků projektu při konzultacích
- dodržení stanoveného rozsahu práce – minimálně pět citovaných zdrojů (nelze citovat web Wikipedia), alespoň dvě témata pro řešerši
- kvalita vypracovaných řešerši
- dodržení typografických pravidel
- kvalita provedení praktické části práce
- splnění pokynů k vypracování
- prezentace výsledků projektu (plakát) a schopnost obhajoby práce (prezentace)

Hodnotící kritéria praktické části maturitního projektu:

- dokumentace, popis API, uživatelská dokumentace, popis instalace a nasazení a popis ukázek (10)
- UI (responzivita, překrývání, barvy, text, ...) (20)
- funkčnost řešení (20)
- kvalita kódu a správné použití nástrojů a dokumentace (10)
- návrh architektury aplikace - normalizace a uml (databáze, OOP) (10)

Rozsah práce:

- minimální rozsah textové části práce (řešerše a popis praktického řešení) je 15 normostran textu (bez formálních částí – obsah, literatura atd.)
- minimální rozsah praktické části je stanoven pokyny k vypracování práce, tj. splněním cílů práce
- V případě zavedení distanční výuky, nebo nařízení karantény v předmětu Projekt, trvající déle než 45 kalendářních dnů (vč. období prázdnin), nemusí žák odevzdávat fyzický prototyp, který je součástí praktické části práce. V tomto případě bude při hodnocení kladen důraz na dokumentační část projektu (výkresy, modely, schémata, simulace, ...), podle které musí být prototyp realizovatelný.

Počet vyhotovení práce:

- maturitní práce bude odevzdána v elektronické podobě
- veškeré uložené textové dokumenty, včetně příloh (i fotodokumentace a videozáznamů), budou odevzdány v původním formátu (zdrojový formát např. .DOCX) i ve formátu .PDF.
- textová část dokumentu bude obsažena v jednom souboru a bude obsahovat všechny formální části (titulní strana, anotace, obsah atd.), pro zadání bude v dokumentu jedna nečíslovaná stránka
- elektronická verze práce (včetně prezentace) bude uložena na Google Classroom (přesné pokyny k odevzdání budou zaslány žákům na školní email, nebo prostřednictvím IS v průběhu února 2024).

V Praze dne 24. 10. 2023

.....
Ing. Lukáš Procházk
ředitel školy

převzal dne: 24.10.2023

podpis žáka: Mlejnský

Poděkování

Chtěl bych poděkovat panu Ing. Přemyslu Vaculíkovi za vedení a pomoc při tvorbě mé maturitní práce. Dále bych chtěl poděkovat mé rodině a kamarádům za pomoc, podporu a konstruktivní připomínky k mé maturitní práci.

Prohlášení

Prohlašuji, že jsem svou maturitní práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje (literaturu, projekty, SW, atd.), které jsou uvedené v seznamu literatury.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Abstrakt. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze, březen 2024

.....

Podpis autora práce

Abstrakt

Práce se zaměřuje na tvorbu webové aplikace, která zobrazuje v mapě informace o hromadné dopravě v české republice. Cílem je získat a uživatelsky přívětivě zobrazit data, které stát poskytuje o hromadné dopravě.

Klíčová slova: doprava, zastávky, stanice, vlaky, autobusy, jízdní řády, MHD, API

Abstract

This work focuses on the creation of a web application that shows information about public transport in the Czech Republic on a map. The work aims to obtain and display in a user-friendly environment the data that the state provides about public transport.

Keywords: transport, stops, stations, trains, buses, timetables, public transport, API

Obsah

Úvod	1	
1 Průzkum trhu	2	
1.1 Databáze a poskytovatelé dat.....	2	
1.1.1 CHAPS – CIS JŘ.....	2	
1.1.2 Datové sady PID.....	2	
1.2 Vyhledávače spojů.....	3	
1.2.1 Seznam Jízdní řády / Mapy.cz...	3	
1.2.2 IDOS.....	3	
1.2.3 Mapa PID.....	3	
2 Teoretická část	4	
2.1 Specifikace otevřených dat	4	
2.1.1 Definice otevřených dat v českém zákoně	4	
2.1.2 O otevřených datech.....	4	
2.1.3 Zveřejňovatel otevřených dat v ČR	4	
2.1.4 Podoba otevřených dat [3].....	4	
2.1.5 Národní katalog otevřených dat.	5	
2.1.6 Stupně otevřených dat	5	
2.1.7 RDF specifikace	6	
2.1.8 RDF Schéma.....	6	
2.2 Poskytovatelé dat o dopravě v ČR..	7	
2.2.1 CHAPS	7	
2.2.2 Dopravci	8	
2.2.3 Poskytovatelé zastávek.....	8	
2.3 Dopravci poskytující data	9	
2.3.1 PID [35].....	9	
2.3.2 IDOL [36].....	9	
2.3.3 IDS-JMK [37].....	9	
2.3.4 VDKHK [38].....	10	
2.3.5 Dopravní podnik měst Mostu a Litvínova [39]	10	
2.3.6 Jihočeský kraj [40]	10	
2.3.7 Statutární město Plzeň [41]	10	
2.3.8 DPO [42]	11	
2.3.9 Statutární město Děčín [43]	11	
2.4 Formát – JDF [14][15][16][17]	11	
2.4.1 VerzeJDF.....	12	
2.4.2 Zastávky	12	
2.4.3 Oznacníky (nepovinný).....	13	
2.4.4 Pevnýkod.....	13	
2.4.5 Dopravci.....	13	
2.4.6 Linky	14	
2.4.7 LinExt (nepovinný)	15	
2.4.8 Spoje.....	15	
2.4.9 SpojSkup (nepovinný).....	15	
2.4.10 Zaslinky.....	15	
2.4.11 Zasspoje.....	16	
2.4.12 Udaje (nepovinný).....	16	
2.4.13 Caskody.....	17	
2.4.14 Navaznosti (nepovinný)	18	
2.4.15 Altdop (nepovinný)	18	
2.4.16 Altlinky (nepovinný).....	19	
2.4.17 Mistenky (nepovinný)	19	
2.4.18 Rozdíly předchozích verze - 1.10 [16][17]	19	
2.4.19 Rozdíly předchozích verze – 1.9 [15][16]	20	
2.4.20 Rozdíly předchozích verze – 1.8 [14][15]	20	
3 Použité technologie	21	
3.1 Python.....	21	
3.2 HTML	21	
3.3 CSS	21	
3.4 JavaScript.....	22	
3.5 JQuery + JQueryUI.....	22	
3.6 Leaflet.....	22	

3.7	OpenStreetMap.....	23	4.3.5	System na porovnávání názvů .	36
3.8	API.....	24	4.3.6	Zpracovávání souborů s polohami zastávek	38
3.8.1	REST API	25	4.3.7	Zpracování souborů s trasami spojů a linek	39
3.8.2	GraphQL.....	25	4.3.8	API.....	40
3.9	Databáze [74][75]	26	4.4	Tvorba systému – webová aplikace	42
3.9.1	MySQL	26	4.4.1	Vzhled.....	42
3.9.2	PostgreSQL	26	4.4.2	Funkce aplikace	42
3.9.3	SQLite	27	4.4.3	Popis webové aplikace.....	42
4	Praktická část	28	5	Uživatelská dokumentace	45
4.1	UI a UX návrh webové aplikace....	28	5.1	Webová aplikace	45
4.2	Návrh databáze	30	5.1.1	Hlavní stránka.....	45
4.2.1	Dopravci (Operators)	31	5.1.2	Nastavení	46
4.2.2	Typy dopravy (TransportTypes)	32	5.1.3	Okno s hlavní zastávkou.....	46
4.2.3	Vozidla (TransportVehicle)	32	5.1.4	Okno s linkami.....	47
4.2.4	Linky (Lines)	32	5.1.5	Okno se spoji	47
4.2.5	Popis linky (LineDescription)..	32	5.1.6	Okno se seznamem zastávek spoje	47
4.2.6	Skupina spojů (TripsGroups)...	32	5.1.7	Okno se seznamem propojených zastávek	47
4.2.7	Spoje (Trips)	32	5.1.8	Okno s alternativní zastávkou..	47
4.2.8	Zastávky (Stations)	32	5.2	Aktualizace databáze.....	47
4.2.9	Sloupky / Označníky (Stops) ...	32	5.2.1	Instalace Pythonu.....	47
4.2.10	Trasy (Routes).....	33	5.2.2	Instalace závislostí.....	48
4.2.11	Pevné kódy (FixCodes).....	33	5.2.3	Spuštění skriptu	48
4.2.12	Pevné kódy linku (CodesInLines)	33	5.2.4	Ovládání skriptu.....	48
4.2.13	Pevné kódy spoje (CodesInTrips)	33	5.2.5	Kde se nachází databáze	48
4.2.14	Pevné kódy zastávky (CodesAtStation)	33	Závěr		1
4.3	Tvorba systému – API a databáze .	34	Zdroje		2
4.3.1	Databáze.....	34			
4.3.2	Získávání dat ze zdrojů	34			
4.3.3	Postup získání dat	35			
4.3.4	Porovnávání názvu - teorie	36			

Seznam obrázků

Obrázek 1 - (Upraveno) Znázornění síťového efektu – vzájemné propojení dat [83].....	6
Obrázek 2 - Příklad RDF schématu z W3C dokumentace [8].....	7
Obrázek 3 – Návrh vizuálu webové aplikace. V pozadí mapa s body, které znázorňují zastávky. V popředí vizualizace jednoho okna.....	29
Obrázek 4 - Znázorňuje, ze kterého okna půjde, které otevřít. Přerušovaná čára znázorňuje možnou změnu z Druhé zastávky na Hlavní zastávku.	29
Obrázek 5 – UML návrh databáze	31

Seznam tabulek

Tabulka 1 - Obsah jednoho .zip souboru v JDF formátu [17]	12
Tabulka 2 - Typy linky [17]	14
Tabulka 3 - Dopravní prostředky linky [17]	14
Tabulka 4 - Nepovolené kombinace Typů časových kódů [17].....	17
Tabulka 5 - Typy návaznosti, jejich popis a symbol [17].....	18

Úvod

Kdo hledá, jak se dostat pomocí veřejné dopravy do školy, práce nebo kam má zrovna namířeno, kromě papírových jízdních řádů může využít i elektronické jízdní řady nebo aplikace pomocí nich vytvořené, které dokážou najít rychle a efektivně spoje i s přestupy.

Nejpopulárnější vyhledávač spojů je bezpochyby IDOS (idos.idnes.cz), provozovaný mediální společností MAFRA, ale již existují i další alternativy. Pořád vcelku nové, ale již do širšího povědomí se dostávající, jsou Jízdní Řády od Seznamu (dříve Pubtran). Ty zprostředkovává technologický gigant Seznam.cz a hlavně díky jejich několikaletým soudním tahanicím byly v roce 2015 všechny tyto data o dopravě v ČR poskytnuty veřejně. [1]
[2]

Hlavně díky tomu mohlo autora, který o této skutečnosti četl, napadnout, že zatím neexistuje žádná služba, která by dokázala všechny tyto data pouze zobrazit v mapě, bez vyhledávání spojů. Autor původně zamýšlel udělat jednoduchý web na zobrazení míst, kam se dá dojet bez přestupu, avšak když zjistil, jaké všechny informace lze z různých zdrojů získat, rozhodl se vytvořit webovou aplikaci, ve které by mohly všechny tyto data být přehledně zobrazeny.

Cílem práce je tedy vytvořit co nejobsáhlejší a zároveň přehlednou mapu pro zobrazení dat o dopravě v ČR. Zde by se mohl uživatel podívat, kde se nachází nejen jeho nejbližší zastávky, ale také kam až může ze zastávky dojet, jakým spojem, kde spoj zastavuje, kdo ho provozuje a další veřejně dostupné informace.

1 Průzkum trhu

K první části této práce – databáze a získávání / práce s daty, jsou alternativou hlavně systémy, ze kterých budu data čerpat. Tyto systémy ale často neobsahují ucelené informace v lehce použitelném formátu, což se tato práce bude snažit změnit.

K webové aplikaci, kterou se snažíme vytvořit, neexistuje přesná alternativa. Existují ale aplikace, které se také zabývají veřejnou dopravou, převážně na vyhledávání spojů.

1.1 Databáze a poskytovatelé dat

1.1.1 CHAPS – CIS JŘ

CIS JŘ (Celostátní informační systém o jízdních řádech) shromažďuje a poskytuje data o veřejné dopravě z pověření státu, zároveň je provozovaný pod hlavičkou státní ČD (České dráhy). [79]

Data jsou veřejně zdarma k dispozici na jejich FTP serveru. Tam lze ale nalézt větší množství dat, než které jsou pro tento projekt využitelná. [10]

1.1.2 Datové sady PID

Otevřená data o veřejné dopravě v systému Pražské integrované dopravy i na území Středočeského kraje a části Libereckého kraje. Poskytuje je státní společnost ROPID (Regionální organizátor pražské integrované dopravy) ve spolupráci se státní IDSK (Integrovaná doprava Středočeského kraje).

Tyto data jsou zdarma ve standardu GTFS (General Transit Feed Specification), formát vytvořený a provozovaný Googlem. I přes obsáhlost tohoto formátu, má své nedostatky, jedním z nich je, že neobsahuje formát na zastávky, který je značně důležitý pro použití v této práci. [35]

1.2 Vyhledávače spojů

1.2.1 Seznam Jízdní řády / Mapy.cz

Jízdní řády od Seznamu jsou jednoduchý vyhledávač spojů za pomocí dat od Chaps.cz. Zároveň jsou tyto data integrována i do mapy.cz (taktéž od Seznamu) kde jsou zastávky zobrazeny v mapě a dají se zde vyhledávat aktuální spojení. [80]

1.2.2 IDOS

Jeden z prvních vyhledávačů spojů, používal data od CHAPSu ještě když nebyla veřejná (byl vytvořen CHAPSem). Na začátku jeden z nejlepších v Evropě. [81]

1.2.3 Mapa PID

Mapa od PIDu, která zobrazuje aktuální spoje (kde se nachází) na mapě. [82]

2 Teoretická část

2.1 Specifikace otevřených dat

Používá se i anglický termín Open Data.

2.1.1 Definice otevřených dat v českém zákoně

Citace ze zákona § 3a odst. 5 zákona č. 106/1999 Sb. o svobodném přístupu k informacím - znění od 01.04.2023: „Otevřenými daty se pro účely tohoto zákona rozumí informace zveřejňované způsobem umožňujícím dálkový přístup v otevřeném a strojově čitelném formátu, jejichž způsob ani účel následného využití není povinným subjektem, který je zveřejňuje, omezen a které jsou evidovány v národním katalogu otevřených dat.“ [3]

2.1.2 O otevřených datech

Otevřená data jsou strojově čitelná data, vytvořená za účelem dalšího použití v souvisejících systémech vytvořených třetími stranami. [5][9]

2.1.3 Zveřejňovatel otevřených dat v ČR

Otevřená data může zveřejňovat jakákoli státní instituce. Některé mají zákonu povinnost data zveřejňovat, ostatní je mohou zveřejňovat dobrovolně, pokud dodrží zákonné normy. [5][6]

2.1.4 Podoba otevřených dat [3]

- Otevřená data jsou volně dostupná bez poplatků nebo za odpovědný poplatek spojený s vytvořením, údržbou a zveřejněním dat.
- Všechny data by měla být aktuální.
- Poskytovatel nesmí omezovat použití dat. Použití dat musí ale být v souladu s platnými zákony České republiky.
- Data jsou zveřejňována v národním katalogu otevřených dat.
- Datové sady mají potřebnou dokumentaci, k jejich správnému použití.
- Jejich stažení nesmí být nijak omezené. Například dostupné bez předchozí registrace, neomezené na počet stažení anebo dostupné pro lidi, ale i roboty.
- Strojová podoba by měla být co nejsnadněji použitelná programátory.

2.1.5 Národní katalog otevřených dat

Katalog, který má zkratku NKOD, je soubor publikovaných dat. Byl vytvořen za účelem zjednodušení a zpřehlednění otevřených dat zveřejňovaných veřejnou správou ČR. [6]

2.1.6 Stupně otevřených dat

Tim Berners-Lee vytvořil schéma pro úrovně otevřených dat. Označuje se jako Timův 5-hvězdičkový plán. Všechny stupně mají společné, že jsou zveřejňovány na webu a mají otevřenou licenci pro další použití, komerční i nekomerční [4]

1 Hvězdička

Data mají libovolný formát a jsou jen těžce strojově čitelná, ale lze si je prohlédnout a pochopit pouhým okem. Například naskenovaný dokument nebo obrázek.

2 Hvězdičky

Data mají strukturovaný proprietární formát, lze je číst i strojově, ale obsahují i nadbytečné informace jako třeba formátování. Lze si je zobrazit a prohlédnout, ale se správným softwarem je lze i strojově přečíst. Například strukturovaná Excelovská tabulka.

3 Hvězdičky

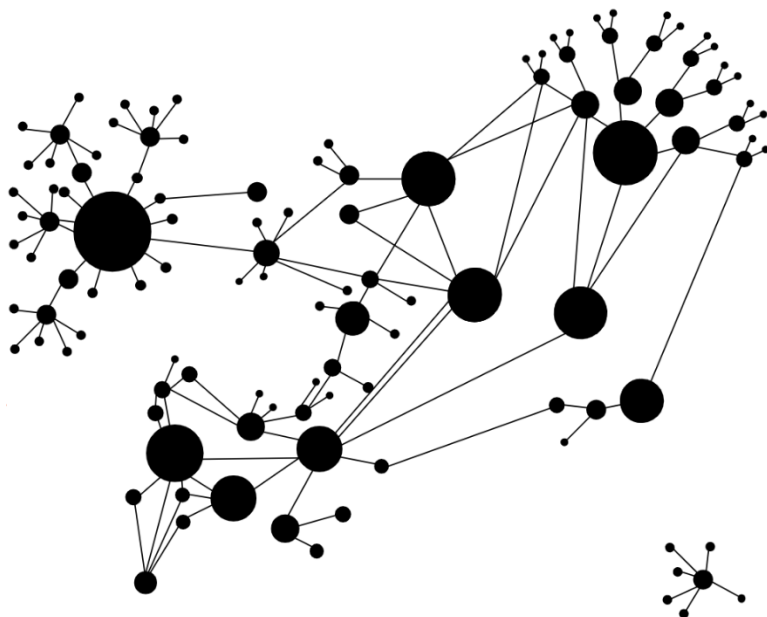
Data jsou v otevřeném formátu. Mohou tedy být strojově čitelné bez speciálního softwaru, ale špatně čitelná uživatelem, pokud nejsou zpracovány. Například CSV, XML nebo JSON.

4 Hvězdičky

Data jsou přímo na webu a lze na ně odkazovat odjinud. Lze získat a použít jen části dat. Například formát RDF.

5 Hvězdiček

Data jsou na webu a odkazují se na další data. Vytváří tak propojenou síť dat, takzvaný síťový efekt (**Obrázek 1**). Využívá se například formát RDF s odkazy na další data, která jsou buď také pěti hvězdičková nebo čtyř hvězdičková, tedy přístupná přímo na webu.



Obrázek 1 - (Upraveno) Znárodnění síťového efektu – vzájemné propojení dat [83]

2.1.7 RDF specifikace

Resource Description Framework (v češtině Systém na popis zdrojů) jsou specifikace pro popis dokumentu, který lze strojově číst. Specifikace byla vytvořena organizací World Wide Web Consortium (W3C). Usnadňuje propojení jednotlivých dat pomocí URI, těmi ale nemusí disponovat. [7][8]

2.1.8 RDF Schéma

RDF Schéma (zkráceně RDFS) je značkový jazyk založený na XML. Obsahuje a popisuje jednotlivé prvky dat, jejich vzájemné propojení a vztahy. Má hierarchii, tedy některé prvky jsou podřazeny jiným. Prvky mohou odkazovat na jiné data nebo webové stránky pomocí URI, tyto data nemusí být na stejném místě jako původní, ale i jinde na internetu. Například odborný článek o měrné jednotce v poskytovaných datech (**Obrázek 2**). [8][9]

EXAMPLE 133

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/1999/xhtml/datatypes/"
  xmlns:xh1id="http://www.w3.org/1999/xhtml/datatypes/"
  targetNamespace="http://www.w3.org/1999/xhtml/datatypes/"
  elementFormDefault="qualified"
>
  <xs:simpleType name="CURIE">
    <xs:restriction base="xs:string">
      <xs:pattern value="([\i-[:]][\c-[:]]*)?(\/[^\s/][^\s]*|[^\s/][^\s]*|[\s]?)" />
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
```

Obrázek 2 - Příklad RDF schématu z W3C dokumentace [8]

2.2 Poskytovatelé dat o dopravě v ČR

2.2.1 CHAPS

Hlavním zdrojem dat pro tento projekt je Celostátní informační systém o jízdních řádech (CIS JŘ), který spravuje společnost CHAPS, vlastněná dceřinou společností Českých drah. [10][11][13][18]

System CIS JŘ poskytuje jízdni řady v elektronické podobě a také seznam všech zastávek, které se v jízdni řádech objevují. Jízdní řady jsou poskytovány v jednotném datovém formátu (JDF), který je navržen a používán výhradně tímto systémem. Použití a strukturu formátu JDF určuje Ministerstvo dopravy, podle vyhlášky Ministerstva dopravy ČR č. 388/2000 Sb., o jízdni řádech veřejné linkové osobní dopravy. Nejnovější verze formátu je 1.11, verze, které se stále používají jsou: 1.8, 1.9, 1.10 a 1.11. [12][13][14][15][16][17][18]

Dokumentace je k dispozici na oficiálních stránkách. Formát 1.8 by se podle zákona neměl používat, ale stále ho někteří dopravci využívají. Dokumentace k formátu 1.8 na oficiálních stránkách CHAPSu ani Ministerstva dopravy není k dispozici. Jednotlivé verze, nejsou zpětně kompatibilní, což práci s nimi komplikuje. [13][14][15][16][17][44]

Zároveň CIS JŘ obsahuje i seznam všech zastávek, které se v JDF vyskytují. V tomto seznamu ale není poloha zastávek. Polohu zastávek, které jsou stěžejní pro zobrazení v mapě a bez nichž by projekt nemohl fungovat tak bohužel nelze získat přímo z CIS JŘ. Data z CIS JŘ jsou volně přístupná na FTP serveru: <http://ftp.cisjr.cz/> nebo na webové adrese: <https://portal.cisjr.cz/pub/>. [13][18][19][20]

Polohu zastávek lze získat od konkrétních dopravců v regionech.

2.2.2 Dopravci

V České republice je pro téměř každý kraj jiný organizátor dopravy. Organizátoři jsou sdruženi v České asociaci organizátorů veřejné dopravy (ČAO VD). [21]

Seznam dopravců v ČAO VD:

- JIKORD (Jihočeský koordinátor dopravy) [22]
- KIDS KK (Koordinátor integrovaného dopravního systému Karlovarského kraje) [23]
- KIDSOK (Koordinátor Integrovaného dopravního systému Olomouckého kraje) [24]
- koordinátor ODIS (Moravskoslezský kraj) [25]
- KORDIS JMK (Jihomoravský kraj) [26]
- KOVED ZK (Koordinátor veřejné dopravy Zlínského kraje) [27]
- KORID LK (Koordinátor veřejné dopravy Libereckého kraje) [28]
- POVED (Plzeňský Organizátor Veřejné Dopravy) [29]
- ROPID (Regionální organizátor Pražské integrované dopravy) [30]
- IDSK (Integrovaná doprava Středočeského kraje) [31]
- OREDO (Královéhradecký a Pardubický kraj) [32]

Dopravci, kteří nejsou členy ČAO VD:

- DÚK (Doprava Ústeckého kraje) [33]
- VDV (Veřejná doprava Vysočiny) [34]

2.2.3 Poskytovatelé zastávek

Protože CHAPS poskytuje v rámci CIS JŘ pouze seznam zastávek, bez souřadnic, je nutné polohu zastávek získat jinde. Někteří dopravci právě data o polohách zastávkách zveřejňují v Národním katalogu otevřených dat nebo na svých webových stránkách. [45]

Data poskytuje PID (Pražská integrovaná doprava), IDOL (Liberecký kraj), IDS-JMK (Integrovaný dopravní systém Jihomoravského kraje), VDKHK (Veřejná doprava Královéhradeckého kraje), Dopravní podnik měst Mostu a Litvínova, Jihočeský kraj, Statutární město Plzeň, DPO (Dopravní podnik Ostrava) a Statutární město Děčín. [35][36][37][38][39][40][41][42][43]

2.3 Dopravci poskytující data

2.3.1 PID [35]

Data, která poskytuje PID:

- Jízdní řády ve formátu GTFS
- Aktuální polohu vozů a zpoždění spojů v Realtime API
- Seznam zastávek. Rozšiřují informace o zastávkách, které se nevejdou do formátu GTFS ve formátu JSON nebo XML
- Prodejní místa lístků nebo průkazek (Lítačka) v systému PID
- Data o dalších geografických místech souvisejících s dopravou, například trasy spojů, vstupy do metra nebo tarifní pásma
- Novinky, výluky nebo mimořádně události v kanálu RSS

2.3.2 IDOL [36]

Data, která poskytuje IDOL:

- Autobusové zastávky ve formátu SHP, GML, KMZ
- Tarifní zóny ve formátu SHP, GML, KMZ
- Spoje autobusových linek ve formátu SHP

2.3.3 IDS-JMK [37]

Data, která poskytuje IDS-JMK:

- Jízdní řády ve formátu GTFS
- Trasy linek v různých formátech (GeoJSON, CSV, ...)
- Zastávky MHD v různých formátech (GeoJSON, CSV, ...)

2.3.4 VDKHK [38]

Data, která poskytuje VDKHK:

- Autobusové linky VDKHK v různých formátech (GeoJSON, CSV, ...)
- Vlakové linky VDKHK v různých formátech (GeoJSON, CSV, ...)
- Železniční stanice a zastávky VDKHK v různých formátech (GeoJSON, CSV, ...)
- Autobusové zastávky VDKHK v různých formátech (GeoJSON, CSV, ...)

2.3.5 Dopravní podnik měst Mostu a Litvínova [39]

Tento dopravní podnik poskytuje pouze data o polohách zastávek.

2.3.6 Jihočeský kraj [40]

Jihočeský kraj poskytuje pouze data o plochách zastávek ve formátech SHP, KML, GPKG.

2.3.7 Statutární město Plzeň [41]

Data, která poskytuje Plzeň:

- Vlakové koleje ČD na území Plzně v různých formátech (GeoJSON, ...)
- Tramvajové koleje na území Plzně v různých formátech (GeoJSON, ...)
- Polohu jednotlivých sloupků (označníků) zastávek v různých formátech (GeoJSON, ...)
- Přístřešky na zastávkách MHD v různých formátech (GeoJSON, ...)
- Trasy linek v různých formátech (GeoJSON, ...)
- Polohu zastávek v různých formátech (GeoJSON, ...)

2.3.8 DPO [42]

Data, která poskytuje DPO:

- Tramvajové koleje s depy a točnami v různých formátech (GeoJSON, ...)
- Pouze tramvajové koleje v různých formátech (GeoJSON, ...)
- Polohu zastávek v různých formátech (GeoJSON, ...)

2.3.9 Statutární město Děčín [43]

Statutární město Děčín poskytuje pouze data o zastávkách ve formátu SHP.

2.4 Formát – JDF [14][15][16][17]

V této části je popsán obsah nejnovější verze JDF – 1.11, následně jsou popsány odlišnosti starších verzí.

Na FTP serveru ve složce JDF se nachází .zip soubor obsahující .zip soubor pro každou jednu linku. Tento .zip soubor obsahuje 9 – 17 textových souborů (.txt) ve formátu CSV (**Tabulka 1**), tyto soubory jsou hromadně nazývány dávka. Jednotlivé sloupečky v jednom záznamu jsou odděleny čárkou a každý záznam končí středníkem a nový začíná na dalším řádku. Text je kódován ve středoevropském kódování – CP1250.

název souboru	popis souboru	stav
VerzeJDF	Verze jednotného datového formátu	povinný
Zastavky	Zastávky	povinný
Oznacniky	Označníky zastávek	nepovinný
Dopravci	Dopravci	povinný
Linky	Linky	povinný
LinExt	Linky v MHD (IDS)	nepovinný
Zaslinky	Zastávky linky	povinný
Spoje	Spoje	povinný
SpojSkup	Skupiny spojů	nepovinný
Zasspoje	Zastávky spoje	povinný
Udaje	Další údaje potřebné pro informování cestujících	nepovinný
Pevnykod	Pevný kód - seznam pevných kódů použitých na lince	povinný
Caskody	Časové kódy spojů	povinný
Navaznosti	Strojově zpracovatelné informace o možnosti přestupu	nepovinný
Altdop	Alternativní dopravci na lince nepovinný	nepovinný
Altlinky	Alternativní čísla linek nepovinný	nepovinný
Mistenky	Doplňující údaje o předprodeji místenek	nepovinný

Tabulka 1 - Obsah jednoho .zip souboru v JDF formátu [17]

2.4.1 VerzeJDF

Soubor obsahuje informace o použité verzi formátu JDF, místu, datu a identifikaci jedné dávky.

2.4.2 Zastávky

Obsahuje zastávky, které jsou použité v této dávce. Číslo zastávky, na které se vazbí ostatní soubory (**Oznacniky**, **Zasspoje** a **Zaslinky**). Celý název zastávky, ten se skládá ze tří hodnot: Název obce + „,“ + Část obce + „,“ + Bližší místo. Hodnoty Část obce a Bližší místo jsou nepovinné.

Pokud se zastávka nachází v České nebo Slovenské republice, tak se uvádí blízká obec. Dále obsahuje zkratku pro stát (CZ – Česká republika, SK – Slovenská republika). A může obsahovat až 6 různých pevných kódů, viz soubor **Pevnykod**.

2.4.3 Označníky (nepovinný)

Soubor nemusí být v dávce obsažen, obsahuje seznam označnicků na zastávkách. Označnick je jeden sloupek zastávky. Zastávky obvykle mývají dva sloupky na každý směr dopravního prostředku, ale mohou jich mít více nebo třeba jen jeden.

Mají vazbu na Číslo zastávky do souboru (**Zastávky**) a každý označnick má svůj kód podle kterého ho lze jednoznačně identifikovat v rámci jedné zastávky. Pokud je soubor v dávce přítomen, vazbí se na něj soubory – **Zasspoje** a **Zaslinky**. Může obsahovat Název, Směr/popis a Stanoviště, což je název označnicku pro veřejnost.

2.4.4 Pevnykod

Obsahuje doplňující informace k některým souborům v dávce (**Zastavky**, **Spoje**, **Zaslinky**, **Zasspoje** a **Altdop**). Soubor obsahuje Číslo pevného kódu, přes které se vazbí do dalších souborů a Označení pevného kódu.

2.4.5 Dopravci

V souboru jsou uvedeni dopravci, kteří provozují linky a spoje v dávce. Každý dopravce je rozlišen pomocí svého IČ (Identifikační číslo) a pokud jich je více pod stejným IČ, tak jsou dále rozlišení Rozlišením dopravce (pokud je dopravce jeden, sloupeček obsahuje jedničku – „1“).

U každého dopravce musí být uvedeno jeho Obchodní jméno, Druh firmy, který určuje, zda jde o právnickou nebo fyzickou osobu (1 – právnická osoba, 2 – fyzická osoba), sídlo a telefonní číslo do sídla. Pokud jde o fyzickou osobu musí být uvedeno i Jméno fyzické osoby. Dále může obsahovat DIČ (Daňové identifikační číslo), Telefon na dispečink, Telefon na informace, Fax, E-mail a webové stránky.

2.4.6 Linky

Obsahuje jednotlivé linky. V každé dávce je většinou jedna linka, soubor tedy obsahuje stejnou linku, které ale mají různou platnost JŘ. Linka je rozlišena Číslem linky a Rozlišením linky. Linka je navazbená na právě jednoho dopravce, přes IČ dopravce a Rozlišení dopravce. Musí obsahovat Název linky, Platnost JŘ (začátek i konec), Typ linky (**Tabulka 2**), Dopravní prostředek (**Tabulka 3**), jestli je to Výlukový jízdní řád (hodnota 0 nebo 1), zda má Seskupení spojů (hodnota 0 nebo 1), jestli jsou Použity označnický (hodnota 0 nebo 1) a zda je to Jednosměrný JŘ (hodnota 0 nebo 1). Může obsahovat Číslo licence a její platnost (začátek a konec).

Znak	Typ
A	Městská
B	Městská s obsluhou příměstských oblastí
N	Mezinárodní – s vyloučenou vnitrostátní dopravou
P	Mezinárodní – s povolenou vnitrostátní dopravou
V	Vnitrostátní – vnitrokrajská
Z	Vnitrostátní – mezikrajská
D	Vnitrostátní – dálková

Tabulka 2 - Typy linky [17]

Znak	Dopravní prostředek
A	Autobus
E	Tramvaj
L	Lanová dráha
M	Metro
P	Přívoz (plavidlo, loď)
T	Trolejbus

Tabulka 3 - Dopravní prostředky linky [17]

2.4.7 LinExt (nepovinný)

Vyskytuje se pouze u linek, které přísluší k městské hromadné dopravě a obsahuje informace o její příslušnosti. Je navazbena na linku přes Číslo linky a Rozlišení linky a obsahuje Pořadí v rámci linky, jak se mají zobrazovat za sebou, Kód dopravy, ten určuje příslušnost ke konkrétní MHD, Označení linky, název nebo číslo pro příslušnou MHD a Preferenci označení (hodnota 0 nebo 1), který smí být pouze jeden pro každou linka a je zvolen při vizualizaci JŘ linky.

2.4.8 Spoje

Soubor obsahuje spoje, které určují jednotlivé časy v jízdních řádech podle souboru Caskody. Je navazbený na linku přes Číslo linky a Rozlišení linky. Má Číslo spoje, které určuje směr, kterým linka jede, liché linky jsou standardní směr a sudé jsou ve směru zpátky. Pokud je u linky nastavený příznak Seskupení spojů, tak obsahuje kód skupiny spojů. Může obsahovat až 10 různých Pevných kódů, respektive vazeb do souboru Pevnykod.

2.4.9 SpojSkup (nepovinný)

Pokud je povoleno seskupování spojů u linky, tak každý spoj musí být v nějaké skupině. Soubor umožňuje seskupovat spoje a zobrazovat je pod společným nadpisem (například „Víkend“). Obsahuje Kód skupiny spojů, na který jsou navazbeny spoje ze souboru Spoje. Dále má Pořadí, podle kterého jsou vizualizovány v rámci linky, Název, pod kterým jsou prezentovány a mohou obsahovat Popis, což je jednořádková poznámka.

2.4.10 Zaslanky

Soubor udává seznam zastávek pro konkrétní linku a jejich pořadí. Je navazben na linku přes Číslo linky a Rozlišení linky. Dále obsahuje Tarifní číslo, které určuje pořadí zastávek a Číslo zastávky, vazba na zastávku. Může obsahovat Tarifní pásmo a Průměrnou dobu, ta určuje počet minut od první zastávky linky. A může obsahovat až tři Pevné kódy.

2.4.11 Zasspoje

Obsahuje seznam zastávek pro každý spoj. Je navazben na linku přes Číslo linky a Rozlišení linky, na konkrétní spoj linky pomocí Číslo spoje a jejich pořadí je určeno Číslem tarifním. Zároveň mají vazbu na zastávku (Číslo zastávky) a pokud je v lince nastaveno použití označnicků, tak se vazbí i na označnick (Kód označnicku).

U normálních spojů je vyplněn Čas odjezdu a může být vyplněn i Čas příjezdu. Výjimka konečné stanice, které neobsahuje Čas odjezdu, ale musí obsahovat Čas příjezdu. Pokud spoj zastávkou projíždí obsahují obě kolonky „|“ a pokud jede jinou trasou obsahuje „<“, jinak časy tvoří neklesající posloupnost. V případě, kdy spoj stojí v zastávce déle než 5 minut, musí být uveden i Čas příjezdu.

Je-li spoje provozován podmíněčně nebo na objednání, musí spoj obsahovat Čas příjezdu a Čas odjezdu, společně s Časem příjezdu min. (nejkratší trasa) a Časem odjezdu max. (nejdelší trasa) Čas příjezdu tvoří v tomto případě nejzazší možný čas příjezdu a Čas odjezdu tvoří nejdřívější možný čas odjezdu. Pokud spoj zastávkou projíždí nebo jí objíždí obsahuje zastávka „|“ (průjezd) nebo „>“ (objezd).

Obsahuje počet kilometrů z výchozí zastávky, pokud je vyplněn Čas příjezdu nebo Čas odjezdu, nebo pokud zastávkou projíždí (symbol „|“). Může obsahovat Číslo stanoviště a až tři Pevné kódy.

2.4.12 Udaje (nepovinný)

Obsahuje dodatečné informace k lince. Je navazben na linku pomocí Číslo linky a Rozlišení linky. A obsahuje Číslo údaje, které udává pořadí záznamů a Text, který obsahuje informaci pro jeden řádek a nesmí překročit 254 znaků. Celkový maximální počet znaků jsou 4000, veškeré znaky navíc budou ignorovány.

2.4.13 Caskody

Určuje časy a dny kdy je konkrétní spoj provozován. Je navazbený na spoj linky přes Číslo linky a Rozlišení linky (linka) a Číslo spoje (spoj). Každý spoj může mít více časových kódů, ty jsou rozlišeny pomocí Číslo časového kódu.

Konkrétní dny a jejich význam je určen Typem časového kódu, ten určí, zda v daném intervalu dnů jede (1), pro konkrétní dny (2) – na jeden záznam vždy jeden den (lze kombinovat s ostatními), pouze jede v konkrétních dnech (3) – na jeden záznam vždy jeden den (nelze! kombinovat s ostatními) nebo nejede (4), také umí rozlišit, pokud jede v lichých (7) nebo sudých (8) týdnech v časovém rozmezí nebo ve všechny liché (5) nebo sudé (6) týdny v období platnosti JŘ. Některé kombinace Typů časových kódů nejsou povolené (**Tabulka 4**) a tak se nesmí u jednoho spoje vyskytovat. Stejný Typ časového kódu se může vždy vyskytovat vícekrát a Typ časového kódu 3 nelze kombinovat s žádným jiným, kromě sebe sama. Další upřesnění, jako konkrétní dny v týdnu, jsou uvedeny v pevných kódech spoje.

Typy časových kódů jsou seskupeny podle Označení časového kódu pod negativní značku ta může být v intervalu 10-99 nebo prázdná. (Negativní značky nemusí být použity vzestupně, ale může být použito kterékoli číslo z intervalu). Negativní značka je uvedena u konkrétního spoje linky (stejná může být i u více než jednoho spoje) a její význam je zobrazen pod jízdním řádem. Pod jednou negativní značkou může být uvedeno více časových rozmezí provozu, ale musí dodržet povolené kódy (**Tabulka 4**).

Může obsahovat Datum od a Datum do, který určuje časové rozmezí dní, ve kterých je spoj provozován. Pro časové rozmezí jednoho dne stačí uvést Datum od. Také může obsahovat Poznámku s dodatečnými informacemi.

5,6,7,8	5,6	5,7,8	6,8	1,8
5,6,7	5,7	6,7,8	7,8	1,7,8
5,6,8	5,8	6,7	1,7	

Tabulka 4 - Nepovolené kombinace Typů časových kódů [17]

2.4.14 Navaznosti (nepovinný)

Soubor obsahuje data o propojených spojích, které na sebe vzájemně nebo jeden na druhého čekají. O jaký typ návaznosti se jedná určuje Typ návaznosti, který obsahuje malé nebo velké M / m (**Tabulka 5**) Obsahuje vazbu na konkrétní zastávku linku pomocí Číslo linky a Rozlišení linky (vazba na linku), Číslo spoje (vazba na spoj) a Číslo tarifní, které určuje konkrétní zastávku. Může obsahovat Číslo přestupní linky, Dobu čekání (v minutách) a Poznámku.

Samozřejmě ale může být navazben na jiný spoj v CIS JŘ přes Číslo zastávky přestupní linky, Kód označnicku přestupní linky, Číslo výchozí/koncové zastávky spoje přestupní linky, Kód výchozího/koncového označnicku spoje přestupní linky. Tyto údaje ale nemusí být vyplněny.

Typ návaznosti	Poznámka	Symbol dle přílohy č. 2 k vyhlášce
m	spoj (...) vyčká (v zastávce ...) na příjezd spoje ... linky ... / vlaku ... / lodní dopravy z ... nejvýše ... minut	△
M	na spoj (...) navazuje (v zastávce ...) spoj ... linky ... / vlak ... / lodní doprava do ...	▲

Tabulka 5 - Typy návaznosti, jejich popis a symbol [17]

2.4.15 Altdop (nepovinný)

Umožňuje přiřadit alternativního dopravce ke konkrétnímu spoji nebo pro celou linku. Je navazben na dopravce uvedeného v souboru Dopravci přes IČ dopravce a Rozlišení dopravce, na linku přes Číslo linky a Rozlišení linky a na konkrétní spoj přes Číslo spoje, pokud má být přiřazen ke všem spojům linky, Číslo spoje obsahuje 0.

Konkrétní dny jsou přiřazeny přes Pevný kód (povolené jsou pouze časové Pevné kódy), přes Typ časového kódu (Pouze hodnota 5 a 6) a Datum od a Datum do (pro jeden den stačí vyplnit Datum do).

Pomocí Pevného kódu lze nastavit jednotlivé dny v týdnu nebo všechny pracovní dny, případně neděli + státní svátky. Typ časového kódu je stejný jako v souboru Caskody, ale lze zde použít pouze možnost pro rozlišení lichých týdnů (5) nebo sudých týdnů (6), případně prázdnou hodnotu pro všechny týdny. Datum od a Datum do určuje rozmezí, ve kterém dopravce spoj / linku provozuje a nemusí být vyplněno (platí po celou dobu platnosti JŘ).

2.4.16 Altlinky (nepovinný)

Alternativní čísla linek, která se používají pro mezinárodní linky, určuje číslo linky pro konkrétní stát. Vazbí se do linky přes Číslo linky a Rozlišení linky a obsahuje Alternativní číslo linky a stát pro který alternativní číslo platí.

2.4.17 Místenky (nepovinný)

Obsahuje dodatečné informace ke spojům, které obsahují Pevný kód – „R“ (k jízdence je možné zakoupit místenku) nebo „#“ (spoj je možné použít jen s místenkou). Informace jsou užitečné pro zjednodušení cestování cestujícím a mohou obsahovat odlišnosti oproti běžné rezervaci místenek. Musí mít vazbu do linky přes Číslo linky a Rozlišení linky, společně s vazbou do spoje přes Číslo spoje. Je zde Tex informace.

2.4.18 Rozdíly předchozích verze - 1.10 [16][17]

V souboru Linky není uvedeno, zda se jedná o Jednosměrný JŘ a nemá povinně uváděný konec platnost JŘ.

V souboru Navaznosti není políčko na Poznámku.

V soubor Zasspoje je o jeden Pevný kód méně a neobsahuje Čas příjezdu min. a Čas příjezdu max.

2.4.19 Rozdíly předchozích verze – 1.9 [15][16]

Chybí soubor Oznamniky a s ním i hodnota Použití označníku v souboru Linky a Kód označníku v souboru Zasspoje.

Chybí soubor LinExt.

Chybí soubor SpojSkup a hodnota Seskupení spojů v souboru Linky a Kód skupiny spojů v souboru Spoje.

Chybí soubor Navaznosti.

Soubor VerzeJDF obsahuje pouze jednu hodnotu: Číslo verze JDF.

V souboru Dopravci není hodnota Rozlišení dopravce, dopravci se stejným IČem tak musí pro rozlišení používat upravené IČ. Hodnota Rozlišení dopravce také chybí v souborech s vazbou na dopravce (Linky, Altdop)

V souboru Linky nejsou hodnoty: Objízďkový JŘ, Seskupení spojů, Použití označníků. Také není uveden Dopravní prostředek a Rozlišení dopravce, neuvádí se ani Rozlišení linky, které chybí i v navazbených souborech (Spoje, Zaslinky, Zasspoje, Udaje, Caskody, Altdop, AltLinky, Mistenky).

V souboru Spoje není hodnota Kód skupiny spojů a Rozlišení linky.

V souboru Zaslinky není Rozlišení linky a Průměrná doba.

V souboru Zasspoje není Kód označníku, Číslo stanoviště a Rozlišení linky.

V souboru Udaje není Rozlišení linky.

V souboru Caskody není Rozlišení linky.

V souboru Altdop není Rozlišení dopravce a Rozlišení linky.

V souboru AltLinky není Rozlišení linky.

V souboru Mistenky není Rozlišení linky.

2.4.20 Rozdíly předchozích verze – 1.8 [14][15]

V souboru Zastavky je název hodnoty Blízká obec změněn na Okres a je povinný pouze pro Českou republiku (hodnota CZ v políčku Stát).

3 Použité technologie

3.1 Python

Python je interpretovaný programovací jazyk, je objektivě orientovaný a je vysokoúrovňový. Je zaměřený na jednoduchost, přehlednost a čitelnost. Může být rozšířen o další funkce přes knihovny (například Django, Flask nebo NumPy). Používá se na skriptování, automatizaci, datovou analýzu, strojové učení nebo tvorbu backendu. Existuje ve verzi Python 2.X a Python 3.X, tyto verze se od sebe liší syntaxí a menšími změnami ve funkčnosti kódu, ale nejsou zpětně kompatibilní. Obě verze mají dílčí pod verze, které jsou kompatibilní, ve vývoji je ale už jenom verze Python 3.X+. [46][47]

3.2 HTML

HTML, zkratka pro HyperText Markup Language (česky hypertextový značkovací jazyk), je značkovací jazyk používaný při tvorbě webových stránek, nejnovější verze je HTML5. Jeho „kód“ je tvořen takzvanými tagy, podobně jako XML, které určují, co se má zobrazit na webové stránce. Nelze v něm přímo programovat, vytvářet funkce, podmínky, proměnné, je určený pouze k specifikaci co zobrazit na webu, zároveň určuje jen minimálně jak se mají jednotlivé tagy zobrazovat. Pro vizuální úpravu se používá CSS. Logická část tvorby webu (funkce, podmínky, proměnné) se provádí na serveru za pomoci programovacích jazyků (nejčastěji PHP, Python, Node.js a další) a k uživateli se dostane jen hotový HTML soubor. [48][49][50]

3.3 CSS

CSS jsou kaskádové styly, popisují, jak zobrazovat soubory napsané v HTML nebo XML. Používají se k vizuální úpravě HTML nebo k tvorbě SVG. Pomocí podmínek určuje, jak vypadá konkrétní tag nebo tagy v HTML / XML. Obsahuje také možnost tvorby proměnných. [51][52][53]

3.4 JavaScript

JavaScript je skriptovací objektově orientovaný asynchronní programovací jazyk. Používá se nejenom k tvorbě dynamického obsahu na webu (frontend), ale může být použit i k tvorbě backendu (například Node.js). Pokud se jedná o frontend, tak běží na počítači uživatele, kde na základě jeho interakcí mění podobu webu (HTML a CSS) a lze tak pomocí něj vytvářet webové aplikace nebo dynamické weby. [54][55]

3.5 JQuery + JQueryUI

JQuery je JavaScriptová knihovna, určený pro jednodušší práci s JavaScriptem. Přidává mnoho funkcí a spoustu jiných vylepšuje (jako nové funkce, nativní JavaScriptové funkce lze stále používat). Veškeré funkce v JQuery jsou pouze na usnadnění tvorby programu a samy o sobě nepracují s vizuálem webu. [56][57]

JQueryUI je knihovna rozšiřující JQuery, která umožňuje jednoduchou tvorbu specifických vizuálních prvků webu, například přesouvání nebo zvětšování elementu nebo tvorbu různých menu. Má i vlastní design, vytvořený v CSS. [56][58]

3.6 Leaflet

Leaflet je JavaScriptová knihovna určená pro zobrazení map na webu (neposkytuje mapové podklady). Podporuje jenom zobrazení map jako obrázky, které získává z uživatelem definované API. Je uzpůsobená i k fungování na mobilních zařízeních. Má spoustu funkcí a může být rozšířena pomocí pluginů. Kromě zobrazení map do nich umí vkládat body, trasy či tvary podle souřadnic. [59][60]

3.7 OpenStreetMap

OpenStreetMap jsou mapy vytvořené komunitou lidí, jsou zdarma a každý je může pomáhat vylepšit. Celý projekt spravuje Nadace OpenStreetMap, která provozuje web i zajišťuje kontrolu úprav v mapě. Díky své velké komunitě obsahují data ze všech koutů planety a spoustu podrobných informací o konkrétních místech. [61][62]

Kromě samotného webu, kde jsou mapy, je v provozu i API s podklady. Z té lze čerpat mapové podklady pro zobrazení ve své aplikaci. Přístup k serveru je zdarma a není nijak zpoplatněný, ale OpenStreetMap může kdykoliv ukončit přístup konkrétní službě, pokud bude nadměrné využití z dané služby. [63][64]

3.8 API

API, anglická zkratka pro Application Programming Interface, je prostředí nebo způsob, kterým spolu mohou komunikovat různé platformy nebo soupis funkcí, metod nebo procedur na ovládání softwaru nebo hardwaru. [65][66][67][68]

V případě tohoto projektu se jedná o API ve formě prostředí pro komunikaci mezi jednotlivými částmi aplikace (databází a webovou aplikací). Pomocí API může webová stránka přistupovat k datům v databázi bez nutnosti uchovávání přihlašovacích údajů do databáze nezabezpečeně.

Pomocí požadavků na získání (GET) nebo uložení (POST) dat přistupuje web k databázi přes vrstvu, která požadavky zpracovává a výrazně omezuje / stěžuje útok na databázi s daty. Mezi další funkce, které lze přes API dělat je mazání (DELETE), úprava (PUT) nebo záplatování či částečná úprava (PATCH). Metoda PATCH není v původní specifikaci pro web, ale s modernizací webu vznikla, aby snížila množství dat, které putují po internetu. PATCH na rozdíl od PUT, který dělá zdánlivě to samé, umožňuje upravit (poslat po internetu) pouze část informací a tím snižuje tok dat přes internet, na rozdíl od metody PUT, která přepisuje informace, a tak posílá po síti i ty informace, které se v databázi nemění. [66][67][68]

Pro tvorbu toho typu API se používá buď REST API nebo GraphQL. [65]

3.8.1 REST API

REST API je rozhraní umožňující přístup k datům na serveru, aby API mohla být považována za REST (někdy také RESTful) musí splňovat několik zásad:

1. Je přístupná přes internet pomocí protokolu http a jednotlivé rozhraní API mají unikátní adresu.
2. Je rozdělen klient a server. Serverová část (poskytující a ukládající data do databáze) je odděleně od rozhraní klienta a umožňuje tak přístup více různým klientům ke stejným datům. To umožňuje jednodušší přenositelnost mezi různými platformami (web, počítačová nebo mobilní aplikace).
3. Každý dotaz je individuální a není propojován s ostatními (Stateless). Server neuchovává data o předchozích požadavcích klienta a všichni klienti mají vždy stejná data.
4. Má oddělené jednotlivé vrstvy, mezi kterými jsou data poskytována a žádná vrstva by neměla mít přístup k jiným než sousedním vrstvám.

Požadavky, nebo také odpovědi, z REST API jsou přes protokol http (stejně jako požadavky) a používané formáty jsou: SON (Javascript Object Notation), HTML, XLT, XML, nebo jednoduchý text. [69][70][71][72]

3.8.2 GRAPHQL

GRAPHQL je modernější API, díky své konstrukci umožňuje programátorovi získat z databáze / serveru přesně ty data která potřebuje. Nejenom že snižuje množství přenášených dat, ale zjednodušuje další využití dat.

Vrací opravdu jenom ty data, o které si programátor řekne a zároveň umožňuje získávat při jednom dotazu data z více zdrojů a tím i snížit množství dotazů na server. Všechny tyto aspekty vedou k tomu, že může být tento typ API rychlejší a méně náročný na prostředky serveru. [65][73]

3.9 Databáze [74][75]

3.9.1 MySQL

MySQL je databázový systém, který běží na serveru. Od roku 2010 je vyvíjený společností Oracle a je poskytován pod dvěma licencemi – základní zdarma a komerční s více funkcemi. MySQL je zaměřený na rychlost a spolehlivost, ale nemusí mít plnou podporu standardu SQL.

MySQL není tak složitý na nasazení, a proto je vhodný na webové aplikace. Dokáže zpracovávat více operací na zápis nebo čtení najednou, ale může mu dělat problém mnoho uživatelů zapisujících data ve stejnou chvíli.

3.9.2 PostgreSQL

PostgreSQL (někdy také Postgres) je pokročilý databázový systém určený pro použití na serveru. Jedná se o otevřený software a na jeho vývoji může spolupracovat téměř kdokoliv, má díky tomu velkou komunitu lidí, která ho stále udržuje. Je zaměřen na co největší podporu standardu SQL, podporuje 160 ze 179 funkcí pro splnění požadavků jádra SQL:2011 a množství volitelných funkcí.

Je vhodný na použití v systémech, kde je důležitá integrita dat a pro komplexní operace, protože dokáže rozložit zátěž na hardware a zároveň zpracovávat požadavky více klientů najednou. PostgreSQL je ale náročné na paměť počítače (pro každého klienta si alokuje 10MB paměti) a pomalejší na jednoduché požadavky.

3.9.3 SQLite

SQLite není běžný databázový systém. Na rozdíl od dvou předchozích je uložen v jednom souboru a ke svému fungování nepotřebuje běžet na serveru. Data z databáze ale nejsou sdílená napříč systémy a má je k dispozici pouze jeden uživatel. Většinou se využívá pro uchovávání dat pro lokální použití v zařízení (například mobilní aplikace), ale lze ho použít i na serveru. SQLite je pod licencí Public domain, což znamená že se autoři vzdaly všech svých autorských práv a software je tedy volně k užívání, kopírování, distribuci u úpravám pro všechny zdarma a bez omezení.

Hlavní výhodou SQLite je jeho jednoduché a rychlé nasazení a používání. Na serveru nemusí běžet žádná další služba na pozadí ke které program přistupuje, ale rovnou si může brát data ze souboru. Nevýhodou je, že do databáze může zapisovat jenom jeden klient najednou.

Pro tento projekt, kde data vytváří a ukládá jedna část programu a API je pouze čte, je tento databázový systém dostačující.

4 Praktická část

4.1 UI a UX návrh webové aplikace

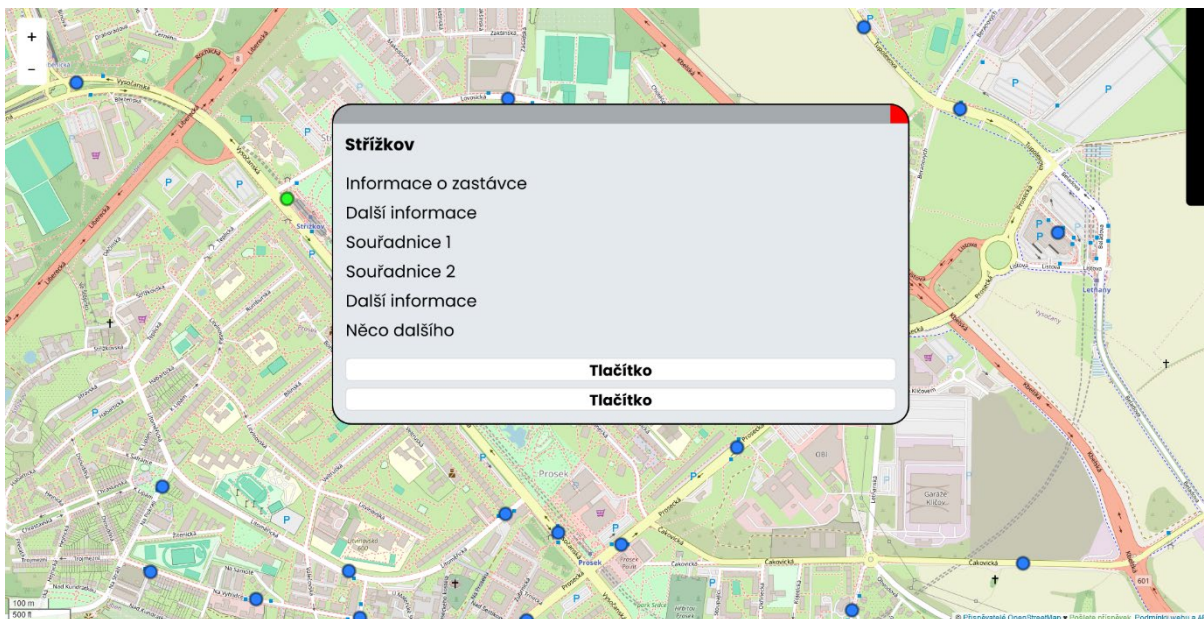
Webová aplikace bude zobrazovat mapové podklady s polohami zastávek, viz **(Obrázek 3)**. Informace o zastávkách, linkách, spojích a trasách budou zobrazeny ve virtuálních oknech webové stránky, jaké lze vidět na **(Obrázek 3)**. Využití oken bylo upřednostněno před využitím jednoho bočního panelu s informacemi z toho důvodu, že díky více oknům lze lépe strukturovat informace a zobrazovat je přehledněji. Každý typ okna může být zobrazen pouze jednou, aby nedocházelo k zahlcení pracovní plochy.

Při kliknutí na zastávku se zobrazí okno s informacemi o Hlavní zastávce. V tomto okně bude několik tlačítek s funkcemi: zaměřit zastávku (vycentrovat mapu na polohu zastávky), zvýraznit na mapě zastávky na které se dá dojet a otevřít okno se Seznamem linek či okno se Seznamem všech propojených zastávek (viz níže). Vizualizace propojení oken je znázorněna na **(Obrázek 4)**.

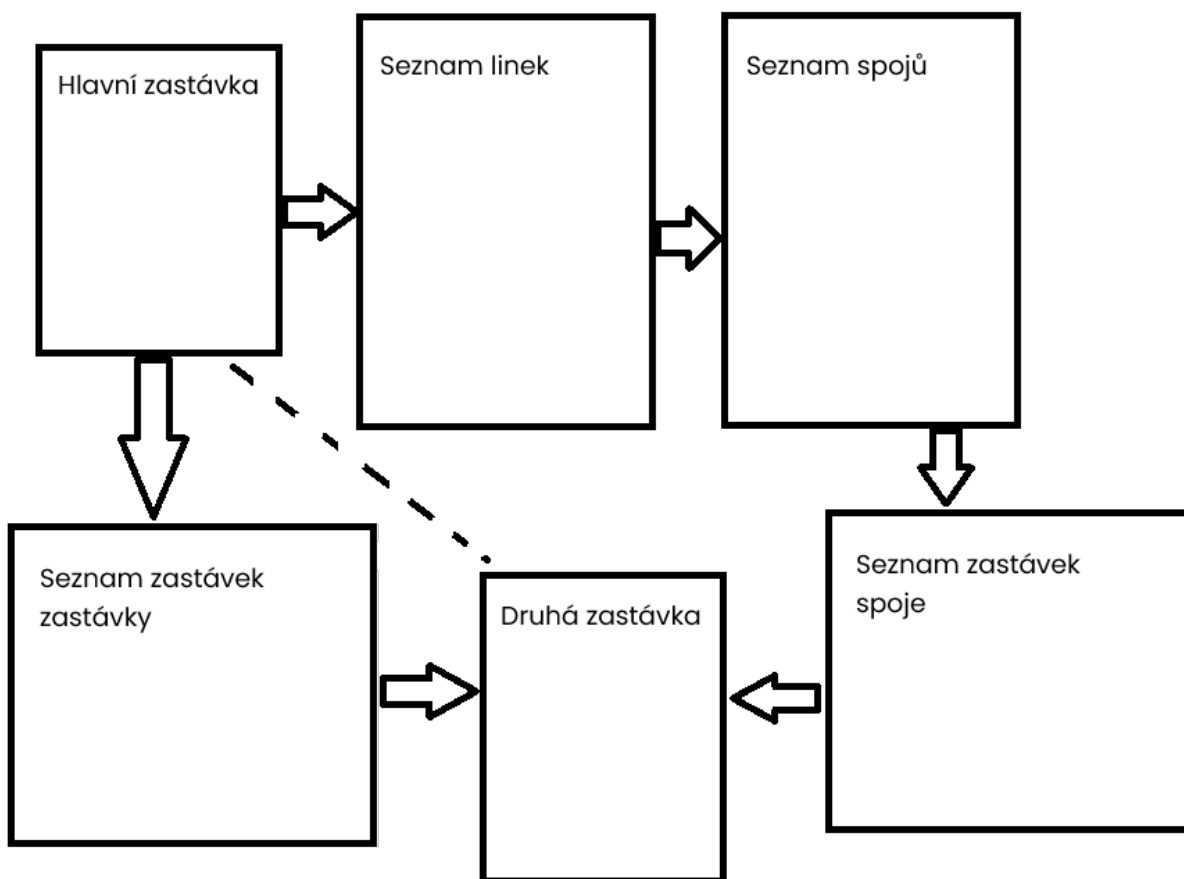
Z okna s Hlavní zastávkou půjde otevřít okno se Seznamem linek, které jezdí na zastávce. Z každé linky půjde zobrazit okno se Seznamem spojů přiřazených k lince. Z každého spoje půjde zobrazit okno se Seznamem zastávek, kterými spoj projíždí. Každá zastávka půjde zaměřit, pokud budou souřadnice zastávky k dispozici anebo zobrazit okno s informacemi o této zastávce (Druhá zastávka). (Popis okna s druhou zastávkou níže).

Z okna s Hlavní zastávkou také půjde otevřít okno se Seznamem všech propojených zastávek zobrazující všechny zastávky, na které se dá z dané zastávky dojet. Z tohoto seznamu půjde zastávka zaměřit nebo otevřít okno s informacemi o této zastávce.

Z okna s informacemi o Druhé zastávce půjde zaměřit zastávku anebo změnit zastávku na Hlavní zastávku.



Obrázek 3 – Návrh vizuálu webové aplikace. V pozadí mapa s body, které znázorňují zastávky. V popředí vizualizace jednoho okna.



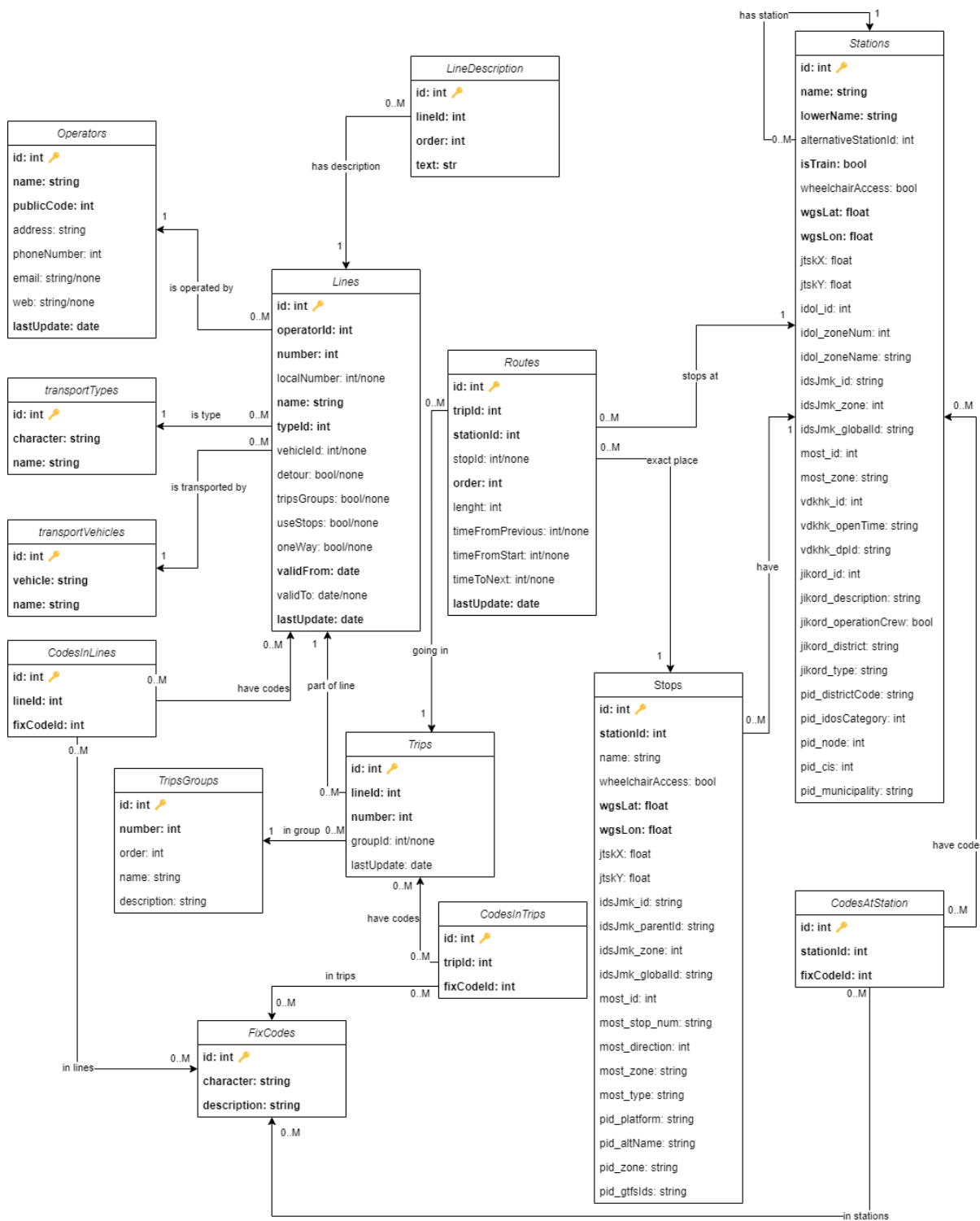
Obrázek 4 - Znárodňuje, ze kterého okna půjde, které otevřít. Přerušovaná čára znázorňuje možnou změnu z Druhé zastávky na Hlavní zastávku.

4.2 Návrh databáze

Databáze projektu obsahuje značnou část informací, které jsou v CIS JŘ, ale je ochuzená o některé pro projekt nedůležité informace. Vychází z podoby, ve které data uchovává CIS JŘ.

Databáze je ale rozšířená o některé tabulky, které jsou v CIS JŘ fixně nastavené. Umožňuje tak lepší škálovatelnost a případně jednodušší rozšíření o další informace (hlavně v podobě pevných kódů, typů dopravy a dopravních prostředků).

Všechny tabulky mají jako primární klíč id (číslo). Na obrázku (**Obrázek 5**) označen ikonkou klíče.



Obrázek 5 – UML návrh databáze

4.2.1 Dopravci (Operators)

Určuje provozovatele (dopravce) pro jednotlivé linky. Musí obsahovat název (name), IČ (publicCode) a čas poslední aktualizace (lastUpdate).

4.2.2 Typy dopravy (TransportTypes)

Specifikuje typ dopravy pro linku. Musí obsahovat znak (character) pod kterým je uveden ve specifikaci CIS JŘ a název (name), také ze specifikace CIS JŘ.

4.2.3 Vozidla (TransportVehicle)

Specifikuje dopravní prostředek, který jezdí na lince. Musí obsahovat vozidlo (vehicle) – znak, pod kterým je uveden v CIS JŘ a název (name), také ze specifikace CIS JŘ.

4.2.4 Linky (Lines)

Specifikuje informace o jedné lince. Musí obsahovat číslo linky (number), název linky (name) a datum poslední aktualizace (lastUpdate). Musí mít vazbu na **Dopravce** a **Typ dopravy**, může mít vazbu na **Vozidlo**.

4.2.5 Popis linky (LineDescription)

Dodatečné informace k lince. Musí obsahovat pořadí záznamů (order) a text informace (text). Musí obsahovat vazbu na **Linku**.

4.2.6 Skupina spojů (TripsGroups)

Seskupuje spoje do určitých okruhů. Musí obsahovat číslo skupiny (number).

4.2.7 Spoje (Trips)

Specifikuje informace o jednom spoji. Musí obsahovat číslo spoje a musí mít vazbu do **Linky**. Pokud to má linka povolené, tak může obsahovat vazbu do **Skupiny spojů**.

4.2.8 Zastávky (Stations)

Informace o jedné zastávce. Musí obsahovat název zastávky (name) a název malým (lowerName), jestli je zastávka vlaková (isTrain) a souřadnice (wgsLat, wgsLon). Může obsahovat informace o zastávkách z různých systémů. Také může mít vazbu na jinou zastávku, pokud je již zastávka se stejným názvem v databázi (alternativeStationId).

4.2.9 Sloupky / Označníky (Stops)

Informace o konkrétním sloupku zastávky, pouze pro některé zastávky. Musí obsahovat souřadnice (wgsLat, wgsLon). Musí mít vazbu na zastávku ke které náleží. Může obsahovat informace o sloupcích z různých systémů.

4.2.10 Trasy (Routes)

Tabulka Trasy obsahuje jednotlivé zastávky každého spoje. Musí mít pořadí (order), které určuje, jak jdou konkrétní zastávky linky za sebou a datum poslední aktualizace (lastUpdate). Dále musí mít vazbu na **Zastávku**, na které zastavuje a může mít vazbu na **Sloupek**.

4.2.11 Pevné kódy (FixCodes)

Pevné kódy určují dodatečnou informaci k zastávce, spoji nebo lince. Musí mít znak (character), pod kterým jsou uvedeny v CIS JŘ a popis (description).

4.2.12 Pevné kódy linku (CodesInLines)

Propojovací tabulka pro pevné kódy v linkách. Musí mít vazbu na jednu **Linku** a jeden **Pevný kód**.

4.2.13 Pevné kódy spoje (CodesInTrips)

Propojovací tabulka pro pevné kódy ve spojích. Musí mít vazbu na jeden **Spoj** a jeden **Pevný kód**.

4.2.14 Pevné kódy zastávky (CodesAtStation)

Propojovací tabulka pro pevné kódy na zastávkách. Musí mít vazbu na jednu **Zastávku** a jeden **Pevný kód**.

4.3 Tvorba systému – API a databáze

4.3.1 Databáze

V tomto projektu je využito databázového systému SQLite. Projekt má dvě databáze, jednu hlavní a jednu s pomocnou.

Hlavní databáze je určena pro uložení všech údajů potřebných k provozování API (viz Návrh databáze). Pomocná databáze obsahuje pouze tabulku se zastávkami (Stations) + sloupeček na párování stanic (match_column), ten obsahuje název stanice v hlavní databázi. Pomocná databáze je určena na zapamatování chybných názvu, které je potom schopný systém automaticky přiřadit na správné místo.

Obě databáze jsou generovány programem, jejich struktura je v souboru /db_settings/models.py, každá třída v souboru určuje jednu tabulku a určuje jednotlivé sloupečky a datový typ sloupečku a vazby mezi tabulkami. Pro generování pomocné databáze je použita pouze třída Stations.

Práci s databází (ukládání nebo získávání dat) se zabývá třída Database v souboru /db_scripts/database_operator.py. Pro práci s databází je použita knihovna sqlalchemy.

4.3.2 Získávání dat ze zdrojů

Jak již bylo uvedeno výše, hlavní zdroj informací pro tento projekt jsou data ze systému CIS JŘ. Tyto data, ale neobsahují údaje o poloze zastávek, a tudíž je zapotřebí tyto data získat z jiných zdrojů, a to konkrétně od regionálních dopravců. Ne všichni dopravci, ale tyto data poskytují, a proto jsou v projektu jenom zastávky z určitých oblastí.

Ačkoli CIS JŘ neposkytuje data o polohách zastávek, poskytuje alespoň jmenný seznam zastávek, které používá v poskytovaných jízdních řádech. Tento seznam byl využit jakožto výchozí bod pro získání souřadnic zastávek.

4.3.3 Postup získání dat

Data je nutné ukládat do databáze ve správném pořadí, aby mohly být vytvořeny vazby mezi jednotlivými tabulkami. První jsou do databáze uloženy všechny zastávky ze seznamu zastávek poskytovaného CIS JŘ, poté souřadnice k těmto zastávkám a někde i sloupky zastávek a nakonec linky, spoje a jejich trasy.

Aby mohly být zastávky správně přiřazeny musí mít stejné jméno v seznamu CIS JŘ a u konkrétního dopravce. Při testování ale bylo zjištěno, že jména jsou často odlišná, a proto bylo zapotřebí porovnat názvy a při dostatečné shodě zastávku aktualizovat na správném místě v databázi.

Odlišnosti, které byly u názvů zaznamenány a je zapotřebí jim přizpůsobit systém, jsou:

- Zkratky v názvu obcí (Jablonec nad Nisou -> Jablonec n.Nisou)
- Zkratky v názvu zastávek (náměstí -> nám.)
- Nadbytečný nebo nedostatečný počet čárek (Jablonec n.Nisou,Kamenná -> Jablonec n.Nisou,,Kamenná)
- Nemá uvedené město kde se nachází (Kino Radnice -> Jablonec n.Nisou,,Kino Radnice)
- Velká a malá písmena nejsou totožná (zš 5.května -> ZŠ 5.května)
- Nesourodé mezery (za čárkou nebo tečkou, kolem pomlčky)
- Dodatečné informace v kulatých nebo hranatých závorkách
- Překlepy (Újezd u Černé Hory, SPD -> Újezd u Černé Hory,SDP)

Další objevený problém byl duplicita některých zastávek.

4.3.4 Porovnávání názvu - teorie

Při porovnávání názvů bylo zapotřebí brát v potaz dříve zjištěné skutečnosti o chybném pojmenování zastávek napříč systémy, a proto by na tento typ porovnávání mohlo být vhodné použít strojové učení či prostou procentuální shodu názvů.

První zmíněná možnost by byla obtížná na vytvoření a autor nedisponoval dostatečnými znalostmi, aby byl schopný vyřešit problém tímto způsobem ve stanoveném čase. Procentuální shoda zase nebyla možná z důvodu značné odlišnosti při zkracování některých názvů.

Bylo tedy rozhodnuto postupně řešit jednotlivé problémy, dokud nebude nalezena příslušná stanice v databázi. Pokud systém stanici nenalezne, bude správce systémem upozorněn a bude mu nabídnuta možnost ručního přiřazení.

4.3.5 Systém na porovnávání názvů

(funkce `search_for_name` v souboru `/db_scripts/database_operator.py`)

Funkce nejdříve zkusí základní hledání s malými písmeny, pokud není nalezena, vyzkouší možné zkratky, které neobsahují tečky. Dále zkusí odstranit první čárku (v některých systémech jsou špatně pojmenována města). Nakonec zkusí nahradit poslední mezeru čárkou. Funkce vrací list se všemi najitými zastávkami. (Každé toto hledání provádí funkce `check_name` v souboru `/db_scripts/database_operator.py`)

(níže je popsána funkce `check_name`)

Nejednoduší na nalezení odlišnosti jsou malá a velká písmena. Databáze, na kterou je dotaz o existenci zastávky volán, porovnává názvy nezávisle na velikosti písmen, tento problém byl vyřešen zvoleným způsobem porovnávání. Tedy takové bylo zdání, dokud nebylo velkým písmem písmeno s diakritikou, malou alternativu k takovému písmu databáze nezná, a proto nemohla nalézt podobnost.

Bylo zapotřebí zvolit jiný způsob. Celý název tak je převeden na malý ještě před prvotním uložením do databáze a uložen do speciálního sloupečku (`lowerName` ve `Stations`). S tímto sloupečkem je pak porovnáván hledaný název. Kromě zmenšení na malá písmena je název i zbaven nadbytečných mezer, například za čárkou či tečkou nebo kolem pomlčky a jsou odstraněny kulaté i hranaté závorky (funkce `clear_name` a z ní volaná funkce `clear_bracket` v souboru `/db_scripts/doprava_global_data.py`).

Pokud není zastávka nalezena a je k dispozici prefix (město ve kterém se zastávky nachází), zkouší porovnání s přidáním prefixem (funkce `check_prefix_name` v souboru `/db_scripts/doprava_global_data.py`).

Další problém při porovnávání jsou zkratky, které se občas vyskytují. Protože by bylo náročné získat všechny zkratky s jejich významy, jsou získány zastávky pomocí dotazu na databázi. Dotaz jsou pouze první písmena jednotlivých slov. Poté jsou znak po znaku porovnávány a pokud obsahují a určené pozici jiný než hledaný znak či tečku, název se liší (funkce `check_shortcut_name` v souboru `/db_scripts/database_operator.py`).

Pokud naopak je zastávka nalezena původním dotazem a je pouze jedna je vyhodnocena jako hledaná zastávka a vrácena jako výsledek.

Pokud ale je vráceno více zastávek, porovná je znak po znaku, aby našel tu správnou (součástí funkce `check_name`).

4.3.6 Zpracovávání souborů s polohami zastávek

Každý dopravce má jinak strukturovaná data, která poskytuje, a také obsahují různé informace. Každý soubor má tak svoji třídu, která data stáhne, zpracuje a uloží do databáze. Základ těchto tříd je stejný a je děděný z třídy `DataScaper` (v souboru `/db_scripts/global_data_scraper.py`). Každý dopravce má tak pouze unikátní funkce na stažení dat a jejich rozdělení na zastávky a sloupky (označníky) (`scrap_data`, v každé třídě unikátní), zpracuje dat o zastávkách (`extract_station_data`, v každé třídě unikátní) a zpracuje dat o sloupcích (`extract_stop_data`, některé třídy neobsahují).

Hlavní třída `DataScaper`, ze které ostatní dědí, tak obsahuje cykly na procházení dat o zastávkách a sloupcích, jednotlivé řádky předává na zpracování funkcím `extract_station_data` a `extract_stop_data`, které vrací list se získanými informacemi o zastávce. Ty jsou pak uloženy do databáze, do sloupečků, které jsou určeny v proměnných `stations_keys` a `stops_keys`. Uložení do databáze je buď funkce `save_station` pro zastávky nebo `save_stops` pro sloupky (obě funkce v souboru `/db_scripts/database_operator.py`).

Pokud funkce `save_station` nalezne více možných zastávek se stejným jménem (funkce `search_for_name`) zvolí jednu ze zastávek jako hlavní a ostatním pak přidělí tuto hlavní stanici jako alternativní stanici v databázi (sloupeček `alternativeStationId`).

Pokud při vkládání zastávky není nalezena žádná shodná zastávka, a ani v pomocné databázi se nenalézá, je nabídnuta správci možnost přiřadit zastávky ručně nebo je nechat vložit do databáze (funkce v souboru `/db_scripts/st_manual_inserter.py`). Pokud jí správce přiřadí nebo uloží je zastávka uložena i do pomocné databáze, ze které při dalším spuštění skriptu jí automaticky přiřadí.

4.3.7 Zpracování souborů s trasami spojů a linek

Další ze zpracovávaných souborů jsou JDF soubory. Tyto soubory jsou uloženy na ftp serveru CIS JŘ v souboru zip. Soubory obsahují informace o linkách, spojích a trasách. Každá dávka je pak ve svém vlastním souboru. (Popis dávky viz výše)

Třída `CIS_JR_Data_Scraper` (v souboru `/db_scripts/cis_jr_lines_scraper.py`) zpracovává tyto data. Tato třída data ze serveru stáhne, rozzipuje a následně prochází jednotlivé zip soubory, jejichž obsah ukládá do databáze. Kromě obsahu JDF zipu, také uloží do databáze informace o fixních datech specifikovaných formátem JDF, na které se soubory odkazují (tabulky `TransportTypes`, `TransportVehicles` a `FixCodes`).

Třída umí pracovat s verzemi formátu JDF: 1.8, 1.9, 1.10 a 1.11. Používání jiných verzí nebylo zjištěno a program tak není kompatibilní s jinými verzemi formátu JDF.

Třída postupně prochází soubory a přidává je do databáze. Mezi soubory přidávanými do databáze je zachována vazba, která je ale upravena na vazbu přes klíče v databázi (sloupeček `id`).

Soubory, které jsou do databáze uloženy, jsou: vazby pevných kódů na ostatní soubory (soubor `Pevnykod.txt`), vazby na zastávky (`Zastavky.txt`), `Dopravci.txt`, `Linky.txt` a jejich alternativní názvy (`LinExt.txt`), `Spoje.txt`, trasy jednotlivých spojů (`Zaslinky.txt` a `Zasspoje.txt`) a poznámky o lince (`Udaje.txt`).

Některé další soubory se nepodařilo implementovat (`Oznacniky.txt` a `SpojSkup.txt`), protože se v žádné dávce nenacházeli a nemohla tak být zaručena funkčnost.

Ostatní soubory z dávky nejsou využívány.

4.3.8 API

Pro tento projekt, aby výsledný web mohl získaná data zobrazit, bylo zapotřebí vytvořit API, ze které bude moci web přistupovat k informacím v databázi. Smysl API u tohoto projektu je, aby poskytovala data pro další použití i ostatním programátorům a uživatelům internetu. Z tohoto důvodu API nebude disponovat zabezpečením proti vytěživání dat a data tak budou k dispozici všem bez omezení.

Pro projekt byla využita Rest API, knihovny fastapi. Jednotlivé rozhraní API jsou definovány v souboru /main.py. Přes API lze data pouze získat, API neumožňuje ukládat ani upravovat data v databázi.

Z API jdou získávat informace na základě klíče (id) z tabulek Stations, Lines a Trips nebo GeoJSON se souřadnicemi všech zastávek. API vrací data ve formátu JSON. Vracená data jsou validována přes příslušnou třídu v souboru /db_settings/schemas.py. Všechny dotazy na databázi jsou v souboru /db_settings/crud.py.

Tabulka Stations obsahuje zastávky a poskytované informace jsou:

- informace o konkrétní zastávce (Stations)
- seznam sloupků navazbených na zastávky (Stops)
- linky jezdící na zastávce (Lines)
- spoje jezdící na zastávce (Trips)
- trasy, ve kterých se vyskytuje zastávka – pouze jeden záznam se zastávkou, ne celá trasa (Routes)
- zastávky, na které lze z dané zastávky dojet – zastávky z tras linek, které na zastávce projíždí – vrací zastávku a čísla linek přes které lze na zastávku dojet
- souřadnice zastávky, na které lze z dané zastávky dojet (Stations)

Tabulka Lines obsahuje linky a poskytované informace jsou:

- informace o konkrétní lince (Lines)
- informace o konkrétní lince s navazbenými daty z tabulek Operators, TransportTypes a TransportVehicles +
- dopravce na lince (Operators)
- vozidlo, kterým je linka provozována (TransportVehicles)
- typ konkrétní linky (TransportTypes)
- popis u linky (LineDescription)
- spoje linky (Trips)
- seznam prvních a posledních zastávek spojů příslušné linky
- seznam zastávek, které linka obsluhuje (Stations)

Tabulka Trips obsahuje spoje a poskytované informace jsou:

- informace o konkrétním spoji (Trips)
- fixní kódy konkrétní linky (FixCodes)
- celou trasu linky (Routes)
- seznam zastávek, kterými spoj projíždí (Stations)

Z API také lze získat soubor se souřadnicemi všech zastávek ve formátu GeoJSON.

4.4 Tvorba systému – webová aplikace

Celá webová aplikace je vytvořená pomocí HTML, CSS a JavaScriptu s knihovnou JQuery, JQueryUI a Leaflet s přidáním modulem MarkerCluster na seskupování míst.

4.4.1 Vzhled

Jak již bylo uvedeno v UI a UX návrhu, stránka zobrazuje mapu a nad ní okna s informacemi. Tyto okna jsou přemístitelná a lze jim upravovat velikost. Aplikace také disponuje nastavením, kterým může uživatel změnit chování aplikace. Pro ikonu nastavení byla využita ikona z knihovny ikon Flaticon [76]

Vzhled webu je upravován styly CSS, které jsou v souboru `/lib/styles/mapy.css`. Stránka je uzpůsobená i na mobilní zařízení, kombinuje responzivní design přes CSS s adaptivní designem pomocí JavaScriptu.

4.4.2 Funkce aplikace

Webová stránka je koncipována, aby se url adresa v adresním řádku dynamicky měnila s akcemi uživatele. Při vložení takové url do nového okna nebo obnovení stránky se pak automaticky načte obsah podle url adresy.

Stránka si také pamatuje uživatelem definované nastavení a poslední pozici a velikost oken. Tyto informace jsou ukládány do uživatelova prohlížeče přes `localStorage`.

4.4.3 Popis webové aplikace

Kód je rozdělen do několika tříd, každá třída obstarává jiné funkce a je v samostatném souboru. Třídy, které obstarávají dynamické změny podoby webu, jsou ve složce `/lib/user-interface/`, třídy, které nemanipulují s obsahem webu, jsou ve složce `/lib/js-functions/` a hlavní soubor, který spouští celý program je `/lib/main.js`.

Okna na stránce jsou otvírána při změně url adresy, tlačítka v oknech mění url adresy a event handler změnu zachytí a provede příslušnou akci. Uživatelské nastavení se chová podobně, pokud se změní, vytvoří event, který jiná část programu zachytí a provede změnu na webu.

Hlavní soubor vytvoří instanci třídy Main, které dále obstarává fungování webu, a instanci třídy Setting, která se stará o uživatelské nastavení webu.

Třída Main (soubor `/lib/user-interface/main-ui.js`) obstarává fungování ostatních částí webu. Při inicializaci načte mapové podklady a body se zastávkami s pomocí třídy Map, mění url adresu pomocí třídy Hash, vytváří jednotlivá okna a jejich obsah s pomocí třídy Popup. Na začátku třída Main vytvoří instanci třídy Map, která načte mapu, poté přidá event handler na všechny tlačítka, která budou v oknech, a nakonec spustí proces na načtení zastávek do mapy. Třída také reaguje na změny url adresy a podle toho mění obsah webu.

Třída Setting (soubor `/lib/user-interface/setting.js`) obsahuje informace o uživatelském nastavení stránky, jak se má zobrazovat a jak má fungovat. Při inicializaci načte nastavení z paměti prohlížeče (`localStorage`) a vytvoří obsah pro popup s nastavením. Při změně nastavení vytvoří event o změně, aby příslušná část programu změnu provedla.

Třída Map (soubor `/lib/user-interface/map.js`) zobrazuje mapu a body v mapě. Při inicializaci načte mapové podklady. Na načítání mapových podkladů využívá API z `OpenStreetMap` a na zobrazení a ovládání mapy JavaScriptovou knihovnu `Leaflet`. Načítá body do mapy z `GeoJSONu`.

Třída Popup (soubor `/lib/user-interface/popup.js`) vytváří a mění popupy. Třída obsahuje pouze statické funkce a není jí potřeba inicializovat. Třída vytváří popupy, podle jména vrací objekt s popupem, zavírá a otvírá popupy (mění je na neviditelné) a upravuje jejich obsah (přidává, maže nebo přepisuje).

Třída Hash (soubor `/lib/js-functions/hash.js`) mění url adresu. Třída obsahuje pouze statické funkce a není jí potřeba inicializovat. Třída vrací, mění a maže hodnoty v url adrese za křížkem („#“).

Třída `TransportApi` (soubor `/lib/js-functions/transport-api.js`) obsahuje všechny rozhraní API, ze kterých lze získat data. Třída obsahuje pouze statické funkce a není jí potřeba inicializovat. Do rozhraní doplní id podle požadavku a vrátí plnou url adresu na data. Pro změnu API stačí změnit první řádek třídy (proměnná `apiUrl`).

Třída `Translator` (soubor `/lib/js-functions/translator.js`) vrací popis řádků k jednotlivým sloupečkům v tabulce `Stations`. Třída obsahuje pouze statické funkce a není jí potřeba inicializovat. Třída vrací popis a obsah pro jednotlivé sloupečky zastávky (`Stations`).

5 Uživatelská dokumentace

Demo aplikace se nachází na adrese <https://maturita.kamdojedu.cz>. Demo API běží na adrese <https://api-maturita.kamdojedu.cz>, výpis a testování rozhraní API lze na adrese <https://api-maturita.kamdojedu.cz/docs>, zde je automaticky vygenerovaná testovací stránka s přehledem všech rozhraní a možností je otestovat.

Demo API má již vygenerovanou databázi, kterou nelze přes webové rozhraní aktualizovat. Pro aktualizaci je potřeba na serveru spustit skript ručně. Aktualizaci lze otestovat na lokálním zařízení bez webového serveru.

5.1 Webová aplikace

Webová aplikace je navržena, aby zobrazovala co nejvíce informací najednou. Toho se snaží dosáhnout oknovacím prostředím na stránce, znamená to, že stránka zobrazuje různé informace v oknech, které lze zvětšovat, zmenšovat a přesouvat.

Pro používání webové aplikace na počítači je vyžadován operační systém Windows 10, Windows 11 nebo Linux, pro mobilní zařízení operační systém Android. Podporované prohlížeče jsou:

- Mozilla Firefox ve verzi 123.0 nebo vyšší
- Microsoft Edge ve verzi 121.0 nebo vyšší
- Google Chrome ve verzi 122.0 nebo vyšší

5.1.1 Hlavní stránka

Při otevření webové aplikace se zobrazí načítací obrazovka, pod kterou se načítají informace do mapy. Když jsou informace úspěšně načteny, zobrazí se mapa se zastávkami seskupenými do skupin. Pokud uživatel použije odkaz, který již obsahuje informace o minulé interakci zobrazí okna a zastávku podle informací v url adrese.

5.1.2 Nastavení

Vpravo nahoře na stránce se nachází nastavení, kterým může uživatel měnit chování a vzhled stránky.

1. Změnit obsah všech oken při změně zastávky
Pokud se změní hlavní zvolená zastávka, tak se změní i obsah oken s linkami a seznamem všech zastávek propojených s hlavní zastávkou. Zároveň zavře všechna ostatní okna.
2. Zakázat seskupování zastávek (není doporučeno, pomalé)
Zobrazí všechny zastávky bez seskupení. Vykreslování stránky může být pomalejší a může být omezen uživatelský komfort.
3. Povolit animace seskupování při přiblížení / oddálení (není doporučeno)
Zastávky se při přiblížení rozdělí ze skupiny na jednotlivé zastávky. Může vytvářet sekavé pohyby při přibližování a oddalování.
4. Zakázat animace zaměření zastávky
Při zaměření zastávky se mapa nebude pohybovat, ale změní místo, kde se nachází.
5. Zaškrtněte pro obnovení původní velikosti a pozice oken (obnoví stránku)
Odstraní z paměti prohlížeče uživatelem upravené velikosti a pozici oken, pro provedení změn znovu načte stránku.
6. Používat mobilní zobrazení
Přizpůsobí vzhled a funkčnost stránky pro mobilní zařízení, zapne se automaticky pokud prohlížeč detekuje, že byla stránka zobrazena na mobilním telefonu.
7. Nezobrazovat že je stránka v Beta provozu
Pokud je zaškrtnuto, tak se při dalším otevření nebo obnovení stránky nezobrazí informace o Beta provozu.

5.1.3 Okno s hlavní zastávkou

Při rozkliknutí bodu zastávky se zobrazí okno s informacemi o příslušné zastávce. Z tohoto okna lze otevřít okno s linkami nebo seznamem všech propojených zastávek (těch na které lze z příslušné zastávky dojet). Také má tlačítko na zaměření zastávky a zvýraznění všech propojených zastávek.

5.1.4 Okno s linkami

Okno zobrazuje seznam linek pro jednu zastávku. Při rozkliknutí jednoho záznamu se zobrazí informace o lince a možnost otevřít okno se spoji této linky. U linky jsou zároveň ikony, které se zobrazují, pokud je linka výluková, jednosměrná nebo s neplatným jízdním řádem.

5.1.5 Okno se spoji

Okno zobrazuje seznam spojů pro jednu linku, na jaké zastávce začíná a končí. Umožňuje zobrazit seznam zastávek pro danou linku.

5.1.6 Okno se seznamem zastávek spoje

Okno zobrazuje jmenný seznam zastávek pro jeden spoj v pořadí, v jakém je spoj projíždí. Zobrazuje čas z první zastávky a počet kilometrů. Pokud uvedenou zastávkou spoj projíždí je uveden znak „|“, pokud spoj zastávku objíždí je uveden znak „<“. Konkrétní zastávku lze zaměřit nebo zobrazit v okně s alternativní zastávkou.

5.1.7 Okno se seznamem propojených zastávek

Okno zobrazuje jmenný seznam zastávek, na které lze dojet spojem z vybrané zastávky. Zobrazuje také linky, kterými se lze na danou zastávku dostat a umožňuje konkrétní zastávku zaměřit nebo zobrazit v okně s alternativní zastávkou. Umožňuje také zvýraznit vypsané zastávky.

5.1.8 Okno s alternativní zastávkou

Okno s alternativní zastávkou zobrazuje stejné informace o zastávce jako okno s hlavní zastávkou. Umožňuje zastávku zaměřit, nebo ji změnit na hlavní zastávku.

5.2 Aktualizace databáze

Skript vyžaduje ke spuštění Python 3.11 s knihovnamy uvedenými v souboru requirements.txt. Skript na aktualizaci databáze běží 2 až 7 hodin, ale může i více. Níže je postup na spuštění a skriptu na lokálním počítači. Počítač musí mít přístup k internetu.

5.2.1 Instalace Pythonu

Minimální požadovaná verze Pythonu je 3.11. Python lze stáhnout z oficiálních stránek: <https://www.python.org/downloads/> [78]. Python je dostupný pro systémy Windows, Linux a macOS. [77][78]

5.2.2 Instalace závislostí

Příkazový řádek na příslušném systému:

```
python -m pip install -r requirements.txt
```

Výše uvedený příkaz instaluje potřebné knihovny přes Python správce knihoven „pip“. Automaticky nainstaluje dostupné knihovny ze souboru requirements.txt.

5.2.3 Spuštění skriptu

Příkazový řádek na příslušném systému, otevřený ve složce s projektem - složka „API_a_databaze“:

```
python update_db.py
```

Tento příkaz spouští Python skript na aktualizace. Skript postupně stáhne data z internetu a uloží do databáze. Do konzole vypisuje aktuální stav průběhu a požaduje akce po uživateli, pokud nalezne problém. Nejdříve stahuje a ukládá zastávky (trvá několik desítek minut) následně program ukládá data o linkách, spojích a jejich trasách (trvá několik hodin).

5.2.4 Ovládání skriptu

Pokud se skriptu nepodaří nalézt zastávku v databázi, požaduje po uživateli interakci. Skript vypíše souřadnice a název zastávky, odkaz na polohu na mapy.cz a seznam názvů, které jsou podobné.

Uživatel číslem zvolí příslušný název nebo může vybrat id z databáze se správným názvem. Do režimu pasní id se uživatel přepne napsáním id a potvrzení enterem. Pro zvolení příslušného čísla zmáčkněte enter, pokud zastávku nelze přiřadit, nechte pole volné a zmáčkněte enter.

Na konci vkládání zastávek bude uživatel vyzván, zda chce zastávky do databáze vložit nebo ne. Stisknutím enteru se data uloží, napsáním „n“ budou ignorována.

5.2.5 Kde se nachází databáze

Databáze s výsledkem se po spuštění skriptu bude nacházet ve složce „API_a_databaze/db/main_database.db“ a lze jí otevřít pomocí softwaru určeného k editování nebo prohlížení SQLite databází.

Závěr

Přínos vypracované práce je zejména v dalším možném využití informací získaných z veřejných zdrojů. Autor se domnívá, že práce kvalitně kombinuje, propojuje a zprostředkovává různé informace, které stát zveřejňuje.

Autor také vidí velký přínos ve zveřejňování těchto informací státem a doufá, že se nadále bude množství dostupných informací, nejenom co se dopravy týče, zvětšovat a bude tak umožněno tvořit další aplikace z těchto dat.

Množství informací zatím není dostatečné, aby mohl projekt pokrývat celou Českou republiku, a proto je soustředěn na města a kraje s nimi spojené, které poskytují potřebné informace (poloha zastávek MHD). Projekt tak zatím zobrazuje zastávky pouze pro Prahu, Středočeský kraj, Jihomoravský kraj, Liberecký kraj, města Most a Litvínov a železniční stanice v Královéhradeckém kraji.

V průběhu práce se vyskytl problém se špatným přiřazováním souřadnic k názvům zastávek, zvláště u těch pojmenovaných stejně, ale nacházejících se v jiném městě. Tento problém se autor snažil odstranit, ale i přesto jsou některé zastávky zatím na nesprávném místě.

Aplikace by také šla rozšířit o informace, kdy spoje na zastávkách jezdí, tím by vznikly plnohodnotné online jízdní řády, ale jelikož autor s tímto nápadem přišel až ke konci vývoje, již nebyl dostatek času na předělání či dotvoření potřebné části systému.

Zdroje

- [1] **Petrů, V.** (2022, 28. července). *Pro jízdenky už nemusíte za Babišem. IDOS ale z někdejšího monopolu teží dodnes.* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://denikreferendum.cz/clanek/34248-pro-jizdenky-uz-nemusite-za-babisem-idos-ale-z-nekdejsiho-monopolu-tezi-dodnes>
- [2] **Seznam Zprávy.** (2015, 29. dubna). *Seznam.cz spustil jízdní řády pro celé Česko.* Novinky. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.novinky.cz/clanek/internet-a-pc-seznam-cz-spustil-jizdni-rady-pro-cele-cesko-296805>
- [3] **Parlament České republiky.** (1999). *Zákon o svobodném přístupu k informacím.* In: *E-Sbírka zákonů.* 1999, https://www.e-sbirka.cz/sb/1999/106#par_3a-odst_5, § 3a odst. 5.
- [4] **Národní portál otevřených dat.** (2022, 28 října). *Informace o portálu.* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://opendata.gov.cz/informace:start>
- [5] **Ministerstvo vnitra České republiky.** (c2024). *Národní katalog otevřených dat na portálu veřejné správy.* Zpravodajství Ministerstva vnitra. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.mvcr.cz/clanek/zpravodajstvi-narodni-katalog-otevrenych-dat-na-portalu-verejne-spravy.aspx>
- [6] **5 Star Data.** (2015, 31 srpna). *5 Star Data: Otevřená data pro všechny.* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://5stardata.info/cs/>
- [7] **World Wide Web Consortium (W3C).** (2015, 17. března). *RDFa Core 1.1 (W3C Recommendation).* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.w3.org/TR/rdfa-core/>
- [8] **Stěněk, P.** (prosinec 2009). *Semantický web: Úvod do principů a technologií.* [Online]. [Citováno 2024, 27. února]. Dostupné z: <http://vse.stencek.com/semanticky-web/ch03s05.html>
- [9] **Open Data Handbook.** (n.d.). *What is Open Data?* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://opendatahandbook.org/guide/en/what-is-open-data/>
- [10] **CHAPS spol. s r.o.** (n.d.). *CHAPS spol. s r.o.: Vývoj dopravního software.* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.chaps.cz/>
- [11] **Sůra, J.** (2017, 30. srpna). *Zadlužené České dráhy půjčí 400 milionů své firmě na nákup Chapsu.* *Zdopravy.cz.* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://zdopravy.cz/zadluzene-ceske-drahy-pujci-400-milionu-sve-firme-na-nakup-chapsu-1093/>
- [12] **Lupa.cz.** (2014, 26. srpna). *Tomáš Chlebičan (Chaps): Data, která má Bileto a Seznam, pocházejí od nás.* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.lupa.cz/clanky/tomas-chlebnican-chaps-data-ktera-ma-bileto-a-seznam-pochazeji-od-nas/>
- [13] **CHAPS spol. s r.o.** (n.d.). *CIS JŘ: Celostátní informační systém o jízdních řádech.* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.chaps.cz/cs/products/CIS>
- [14] Dostupné z: **Katedra softwarového inženýrství MFF UK.** (n.d.). *JDF 1.8* [PDF]. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://ksp.mff.cuni.cz/h/ulohy/32/serial-jr/data/jdf-1.8.pdf>

- [15] **CHAPS spol. s r.o.** (n.d.). *JDF 1.9*. [PDF]. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.chaps.cz/files/cis/jdf-1.9.pdf>
- [16] **CHAPS spol. s r.o.** (n.d.). *JDF 1.10*. [PDF]. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.chaps.cz/files/cis/jdf-1.10.pdf>
- [17] **Ministerstvo dopravy České republiky.** (c2024). *Metodický pokyn č. 5: Jízdní řády veřejné dopravy*. [PDF]. [Online]. [Citováno 2024, 27. února]. Dostupné z: [https://mdcr.cz/getattachment/Dokumenty/Verejna-doprava/Jizdni-rady,-kalendare-pro-jizdni-rady,-metodi-\(1\)/Jizdni-rady-verejne-dopravy/metodicky-pokyn-cis-5.pdf.aspx](https://mdcr.cz/getattachment/Dokumenty/Verejna-doprava/Jizdni-rady,-kalendare-pro-jizdni-rady,-metodi-(1)/Jizdni-rady-verejne-dopravy/metodicky-pokyn-cis-5.pdf.aspx)
- [18] **Celostátní informační systém o jízdních řádech.** (c2024). *Zpracování jízdních řádů pro CIS JŘ - Přihlášení*. [Webová stránka]. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://cisjr.cz>
- [19] **Celostátní informační systém o jízdních řádech.** (c2024). *Seznam zastávek*. [CSV soubor]. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://portal.cisjr.cz/pub/seznamy/zastavky.csv>
- [20] **Celostátní informační systém o jízdních řádech.** (c2024). *Seznam stanic*. [CSV soubor]. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://portal.cisjr.cz/pub/seznamy/stanice.csv>
- [21] **Česká asociace organizátorů veřejné dopravy, z. s.** (c2019). *ČAO VD - [21] Česká asociace organizátorů veřejné dopravy*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://caovd.cz/>
- [22] **JKORD, spol. s r.o.** (c2023). *O nás | Jikord.cz*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.jikord.cz/>
- [23] **Koordinátor integrovaného dopravního systému Karlovarského kraje** (c2024). *Koordinátor integrovaného dopravního systému Karlovarského kraje, příspěvková organizace*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.idok.info/>
- [24] **Koordinátor Integrovaného dopravního systému Olomouckého kraje, p. o.** (n.d.). *Koordinátor Integrovaného dopravního systému Olomouckého kraje -*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.kidsok.cz/>
- [25] **Koordinátor ODIS s.r.o.** (n.d.). *Koordinátor ODIS s.r.o.*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.kodis.cz>
- [26] **Integrovaný dopravní systém Jihomoravského kraje** (c2020). *Integrovaný dopravní systém Jihomoravského kraje*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.idsjmk.cz/index>
- [27] **Integrovaný dopravní systém Zlínského kraje** (n.d.). *Domů • IDZK*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.idzk.cz/>
- [28] **KORID LK, spol. s r.o.** (c2018). *Home | KORID LK*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.korid.cz/>
- [29] **POVED s.r.o.** (c2024). *Integrovaná doprava Plzeňského kraje*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.idpk.cz/cz/>
- [30] **ROPID** (c2024). *Úvodní stránka | Pražská integrovaná doprava Pražská integrovaná doprava*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://ropid.cz>
- [31] **Integrovaná doprava Středočeského kraje, p.o.** (c2021). *Integrovaná doprava Středočeského kraje*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.idsk.cz/>

- [32] **OREDO s.r.o.** (c2024). *OREDO s.r.o. - Integrátor regionální dopravy*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.oredo.cz/>
- [33] **Ústecký kraj** (n.d.). *Doprava Ústeckého kraje: Ústecký kraj*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.kr-ustecky.cz/doprava-usteckeho-kraje/ms-275463/p1=275463/>
- [34] **Kraj Vysočina** (n.d.). *Veřejná doprava Vysočiny: Kraj Vysočina*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.kr-vysocina.cz/vdv>
- [35] **Ropid, o.p.s.** (c2024). *Otevřená data PID | Pražská integrovaná doprava Pražská integrovaná doprava*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://pid.cz/o-systemu/opendata/>
- [36] **Liberecký kraj** (n.d.). *Dopravní mapy Libereckého kraje - OpenData*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://dopravnimapy.kraj-lbc.cz/opendata/>
- [37] **Statutární město Brno** (c2023). *data.Brno*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://data.brno.cz/>
- [38] **Královéhradecký kraj** (c2024). *Data KHK*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.datakhk.cz/search?collection=Dataset&tags=VDKHK>
- [39] **Statutární město Most** (c2022). *Zastávky MHD | OpenData*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://opendata.mesto-most.cz/datasets/mestomost::zast%C3%A1vky-mhd/about>
- [40] **Jihočeský kraj** (n.d.). *Geoportál Jihočeského kraje - Zastávky veřejné dopravy (Doprava a silniční hospodářství)*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://geoportal.kraj-jihocesky.gov.cz/portal/mapy/doprava-a-silnicni-hospodarstvi/zastavky-verejne-dopravy>
- [41] **Město Plzeň** (c2023). *Opendata*. [Online]. [Citováno 2024, 27. února]. Dostupné z: https://opendata.plzen.eu/?tag_id=2&filter-filter%5Btag_id%5D=2
- [42] **Statutární město Ostrava** (c2024). *opendata | Mapový portál města Ostravy*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://mapy.ostrava.cz/data/otevrenata/opendata-info/>
- [43] **Magistrát města Děčín** (2019, 20. června). *Zastávky MHD - DPMD Děčín - Datové sady - opendata města Děčín*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://opendata.mmdecin.cz/dataset/zastavky-mhd-dpmd-decin>
- [44] **Ministerstvo dopravy České republiky** (c2024). *Ministerstvo dopravy ČR - Veřejná doprava*. [Online]. [Citováno 2024, 27. února]. Dostupné z: [https://mdcr.cz/Dokumenty/Verejna-doprava/Jizdni-rady,-kalendare-pro-jizdni-rady,-metodi-\(1\)/Jizdni-rady-verejne-dopravy](https://mdcr.cz/Dokumenty/Verejna-doprava/Jizdni-rady,-kalendare-pro-jizdni-rady,-metodi-(1)/Jizdni-rady-verejne-dopravy)
- [45] **Národní portál otevřených dat** (n.d.). *Datové sady - Národní katalog otevřených dat (NKOD)*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://data.gov.cz/datov%C3%A9-sady>
- [46] **Tagliaferri, L.** (2020, 18. prosince). *What is Python? | DigitalOcean* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.digitalocean.com/community/tutorials/what-is-python>
- [47] **Python Software Foundation** (c2024). *What is Python? Executive Summary | Python.org*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.python.org/doc/essays/blurb/>

- [48] **Codecademy** (2021, 26. dubna). *What is HTML: Common uses and defining features*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.codecademy.com/resources/blog/what-is-html/>
- [49] **Jakpsatweb.cz** (2021). *HTML příručka, přehled HTML tagů* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.jakpsatweb.cz/html/>
- [50] **GeeksforGeeks** (2023, 18. duben). *Frontend vs Backend - GeeksforGeeks*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.geeksforgeeks.org/frontend-vs-backend/>
- [51] **Mozilla Developer Network** (c2024). *What is CSS? - Learn web development | MDN*. [Online]. [Citováno 2024, 27. února]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS
- [52] **Jakpsatweb.cz** (2021). *CSS - Kaskádové styly*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.jakpsatweb.cz/css/>
- [53] **Mozilla Developer Network** (c2024). *CSS: Cascading Style Sheets | MDN*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [54] **Jakpsatweb.cz** (2021). *Javascript - úvod*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.jakpsatweb.cz/javascript/javascript-uvod.html>
- [55] **Mozilla Developer Network** (c2024). *JavaScript | MDN*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/javascript>
- [56] **Kinsta** (2023, 11. září). *What Is jQuery? A Look At the Web's Most-Used JavaScript Library*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://kinsta.com/knowledgebase/what-is-jquery/>
- [57] **jQuery Foundation** (n.d.). *jQuery*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://jquery.com/>
- [58] **jQuery Foundation** (n.d.). *jQuery*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://jqueryui.com/>
- [59] **Leaflet** (c2023). *Leaflet - a JavaScript library for interactive maps*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://leafletjs.com/>
- [60] **Swadia, S.** (2021, 9. prosince). *A Beginner's Guide to Creating a Map Using Leaflet.js — SitePoint*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.sitepoint.com/leaflet-create-map-beginner-guide/>
- [61] **OpenStreetMap** (n.d.). *OpenStreetMap*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.openstreetmap.org/about>
- [62] **OpenStreetMap** (n.d.). *What is OpenStreetMap? | OpenStreetMap*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://welcome.openstreetmap.org/what-is-openstreetmap/>
- [63] **OpenStreetMap** (n.d.). *OpenStreetMap*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.openstreetmap.org>
- [64] **OpenStreetMap Foundation** (n.d.). *Tile Usage Policy*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://operations.osmfoundation.org/policies/tiles/>
- [65] **Kod'ousková, B.** (2023, 14. červen). *Co je to API a jaké jsou možnosti jeho využití?* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.rascasone.com/cs/blog/co-je-api>

- [66] **Smejkal, P.** (2021, 25. dubna). *Co je API a k čemu slouží?* | Petr Smejkal. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.petrsmejkal.cz/clanky/co-je-api-a-k-cemu-slouzi/>
- [67] **Drahomír Hanák** (c2024). *Stopařův průvodce REST API*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.itnetwork.cz/programovani/nezarazene/stoparuv-pruvodce-rest-api>
- [68] **Sophia Iroegbu** (2022, 29. června). *How to Design an API – Application Programming Interface Best Practices*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.freecodecamp.org/news/design-an-api-application-program-interface/>
- [69] **Sievers, N.** (2018, 19. dubna). *Building an API - Get Started* | Swagger Blog. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://swagger.io/blog/api-development/how-to-build-an-api/>
- [70] **Gupta, L.** (2023, 12. prosince). *What is REST?: REST API Tutorial*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://restfulapi.net/>
- [71] **Red Hat** (2020, 8. května). *What is a REST API?* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>
- [72] **Jason Van Schooneveld** (n.d.). *Python and REST APIs: Interacting With Web Services – Real Python*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://realpython.com/api-integration-in-python/>
- [73] **Red Hat** (2019, 8. ledna). *What is GraphQL?* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.redhat.com/en/topics/api/what-is-graphql>
- [74] **Můčka, J.** (2022, 25. květen). *MySQL, PostgreSQL a SQLite: jaké jsou rozdíly mezi databázemi?* [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.master.cz/blog/mysql-postgresql-sqlite-databazove-systemy-rozdily/>
- [75] **ostezer a Drake, M.** (2022, 9. března). *SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems* | DigitalOcean. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>
- [76] **Flaticon** (c2024). *Vector Icons and Stickers - PNG, SVG, EPS, PSD and CSS*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.flaticon.com/>
- [77] **Nauč se Python** (c2024). *Instalace Pythonu – Windows*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://nauce.python.cz/lessons/beginners/install/windows/>
- [78] **Python Software Foundation** (c2024). *Download Python* | Python.org. [Webová stránka]. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.python.org/downloads/>
- [79] **Novinky.cz** (2023, 16. listopadu). *Ministerstvo dopravy skončuje s monopolem na data o jízdnicích řádech - Novinky*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.novinky.cz/clanek/internet-a-pc-ministerstvo-dopravy-skoncuje-s-monopolem-na-data-o-jizdnich-radech-210495>
- [80] **Seznam.cz** (c2024). *Vyhledání spojení • Jízdní řády Seznam.cz*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://www.seznam.cz/jizdnirady/>
- [81] **iDNES.cz** (n.d.). *IDOS • Všechny jízdní řády • Vyhledání spojení*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://idos.idnes.cz/>
- [82] **Pražská integrovaná doprava** (n.d.). *Mapa PID*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://mapa.pid.cz/>

- [83] **Esme González Pillado** (2019, 7. listopadu). *Demystifying Network Effects. The basics of network effects explained | by Esme González Pillado | Stories of Platform Design*. [Online]. [Citováno 2024, 27. února]. Dostupné z: <https://stories.platformdesigntoolkit.com/nfx-cc1dd3aba061>