VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

MULTIMODÁLNÍ DOPRAVNÍ PLÁNOVAČ JIHOMORAVSKÉHO KRAJE

MULTIMODAL TRANSPORT PLANNER FOR THE SOUTH MORAVIAN REGION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. JAN VÁCLAVÍK**
AUTHOR

VEDOUCÍ PRÁCE **Ing. JIŘÍ HYNEK, Ph.D.**
SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání diplomové práce

164673

Ústav:	Ústav informačních systémů (UIFS)
Student:	**Václavík Jan, Bc.**
Program:	Informační technologie a umělá inteligence
Specializace:	Informační systémy a databáze
Název:	**Multimodální dopravní plánovač Jihomoravského kraje**
Kategorie:	Informační systémy
Akademický rok:	2024/25

Zadání:

1. Prostudujte problematiku plánování dopravních spojení. Prozkoumejte přístupy různých měst v ČR a ve světě. Zaměřte se mimo jiné na multimodální dopravní plánování.
2. Studujte systémy a algoritmy související s problematikou vyhledávání tras a plánování dopravy.
3. Analyzujte požadavky města Brna a jeho obyvatel na multimodální plánování dopravních spojení, které kombinuje hromadnou a automobilovou dopravu.
4. Dle požadavků bodu 3 navrhněte systém pro multimodální plánování dopravních spojení v Jihomoravském kraji, který bude kombinovat veřejnou dopravu a automobilovou dopravu.
5. Navržený systém implementujte.
6. Výsledné řešení otestujte ve spolupráci s městem Brno.

Literatura:

- Lazúr, J. (2023). *Analýza a vizualizace dat hromadné dopravy města Brna*. Diplomová práce. Vedoucí práce: Ing. Jiří Hynek. Ph.D. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: https://www.vut.cz/studenti/zav-prace/detail/146479
- Ondrušková, M. (2024). *Analýza a vizualizace dopravních dat města Brna*. Diplomová práce. Vedoucí práce: Ing. Jiří Hynek. Ph.D. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: https://www.vut.cz/studenti/zav-prace/detail/153666
- Rodrigue, Jean-Paul & Comtois, Claude & Slack, Brian. (2016). The Geography of Transport Systems. 10.4324/9781315618159.
- Tufte, E. R. (2001). The Visual Display of Quantitative Information. Graphics Press
- Van Nes, R. (2002). Design of multimodal transport networks. *Civil Engineering. Delft Technical University, Delft*, *304*.

Při obhajobě semestrální části projektu je požadováno:
Body 1 - 4.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce:	**Hynek Jiří, Ing., Ph.D.**
Vedoucí ústavu:	Kolář Dušan, doc. Dr. Ing.
Datum zadání:	1.11.2024
Termín pro odevzdání:	21.5.2025
Datum schválení:	22.10.2024

Abstrakt

Cílem této práce je implementace multimodálního plánovače ve formě webové aplikace, která kombinuje automobilovou a hromadnou dopravu pomocí přestupních bodů, kde dochází k přechodu z auta na hromadnou dopravu. Aplikace je vhodná zejména pro uživatele, kteří denně dojíždějí do práce a chtějí spojit rychlost a flexibilitu auta s pohodlím a nízkými emisemi hromadné dopravy, případně bydlí v místě, kde jsou spoje hromadné dopravy málo frekventované. Aplikace byla implementována pomocí knihovny React a běhového prostředí Deno a byla otestována reálnými uživateli.

Abstract

The main goal of this thesis is to implement a multimodal planner in form of a web application that combines car and public transport by using transfer points where the transition from car to public transport happens. This application is particularly suitable for daily commuters who want to combine the speed and flexibility of a car with comfort and ecofriendliness of public transport or they live in a place with poor public transport connections. The application was implemented by React library and Deno runtime and was tested by real users.

Klíčová slova

multimodální doprava, veřejná doprava, automobilová doprava, dopravní plánovač, trasování, GTFS, vizualizace

Keywords

multimodal transport, public transport, car transport, trip planner, routing, GTFS, vizualization

Citace

VÁCLAVÍK, Jan. *Multimodální dopravní plánovač Jihomoravského kraje*. Brno, 2024. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Hynek, Ph.D.

Multimodální dopravní plánovač Jihomoravského kraje

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Jiřího Hynka, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Jan Václavík
19. května 2025

Poděkování

Děkuji Ing. Jiřímu Hynkovi, Ph.D. za cenné rady a vedení této práce. Děkuji také rodičům a kamarádům za podporu během psaní této práce.

Obsah

Kapitola 1

Úvod

Poptávka po mobilitě neustále roste, zejména v souvislosti s nárůstem počtu obyvatel ve městech. S tím však souvisí i zvyšující se počet automobilů, což vede k častým dopravním zácpám, nedostatku parkovacích míst a negativním dopadům na životní prostředí. Stále větší dopravní zatížení městských oblastí vyžaduje efektivnější způsoby plánování cest, které by pomohly optimalizovat pohyb osob a zároveň přispěly k udržitelnější mobilitě. Řešením by mohlo být použití hromadné dopravy, avšak ne všechna místa jsou vhodně pokryta hromadnou dopravou a ne vždy přijedou spoje hromadné dopravy včas. Nejvhodnějším řešením je proto zkombinovat automobilovou a hromadnou dopravu, neboť jedině tak lze využít rychlost a flexibilitu automobilu s pohodlím a ekologičností hromadné dopravy.

Cílem této práce je proto navrhnout a vytvořit dopravní plánovač, který bude využívat kombinaci automobilové a hromadné dopravy pomocí přestupních bodů, kterými budou železniční stanice a autobusová nádraží. Aplikace bude uživatelům nabízet nejvhodnější trasy s ohledem na čas příjezdu, počet přestupů a další faktory, které mohou ovlivnit komfort a efektivitu cestování. Aplikace bude velmi vhodná pro lidi, kteří dojíždějí do jiného města za prací, případně bydlí v místech, kde nejsou vhodná dopravní spojení. Integrací různých druhů dopravy do jednoho plánovacího nástroje může tento systém pomoci snížit dopravní zatížení měst, zlepšit plynulost dopravy a podpořit využívání veřejné dopravy jako efektivní alternativy k individuální automobilové dopravě.

Kapitola 2 se zabývá konceptem multimodální dopravy, vývojem moderní dopravy a současným trendům. Mezi ty patří koncept mobility jako služby, *Park and ride*, inteligentní dopravních systémy a také snížení uhlíkové stopy. Část kapitoly je rovněž věnována jízdním řádům a jejich reprezentacím a existujícími dopravními plánovači.

Kapitola 3 se zabývá vizualizací geografických dat. Popisuje, jaké mapy se mohou pro vizualizaci použít, rozebírá pojem geografický informační systém (GIS) a rovněž ukazuje, jak se dají využít body, čáry, plochy a pole k vizualizaci geografických dat. Součástí kapitoly je také pasáž o kódování souřadnic a formátech, které se používají pro ukládání a výměnu těchto dat.

Kapitola 4 analyzuje algoritmy, které se používají pro nalezení nejkratší cesty, ať už se jedná o algoritmy pro osobní dopravu nebo tu hromadnou, která pracuje i s jízdními řády. Algoritmy jsou doplněny obrázky, které lépe ilustrují jejich fungování.

Kapitola 5 je zaměřena na analýzu problému a potenciálních uživatelů. Rozebírá cíl práce a také analyzuje existující řešení a na základě této analýzy a uživatelského dotazníku jsou zformovány funkční a nefunkční požadavky na aplikaci.

V kapitole 6 dochází k detailnímu popisu návrhu celého systému. Jsou zde představena kritéria, která budou ovlivňovat kvalitu trasy, architektura systému, je zde rozebrána pro-

blematika přestupních bodů a nakonec se zde rovněž nachází návrh uživatelského rozhraní včetně mobilního zobrazení.

Předmětem kapitoly 7 je především detailní náhled na fungování celé aplikace. Popisuje jednotlivé komponenty a technologie použité při tvorbě řešení a zároveň rozebírá problémy a algoritmy, které byly využity.

Kapitola 8 je poté věnuje testování celé aplikace. Je pokryto uživatelské testování, které může odhalit dříve neodhalené chyby a zároveň přináší nápady na budoucí rozšíření aplikace. Objevuje se zde i testování výkonnosti, která analyzuje výkon aplikace s různými konfiguracemi. Poslední část je zaměřena na analýzu chování aplikace na jiných větších oblastech, než je Jihomoravský kraj.

Kapitola 2

Multimodální doprava

Doprava je multimodální tehdy, používá-li se v ní více než jeden druh dopravy [60, strana 114] a mezi těmito druhy dopravy je třeba vykonávat přestupy. V dnešním světě se nejedná o nic neobvyklého. Je-li například převáženo zboží z Evropy do USA, tak je nejdřív převezeno kamionem do přístavu, poté putuje lodí a nakonec opět dochází k přesunu kamionem, nebo letadlem do místa určení. Zde byla využita kombinace automobilové, lodní a případně letecké dopravy.

V poslední době se lze s tímto pojmem setkat i v kontextu přepravy cestujících, kdy k dosažení cíle použijeme více druhů dopravy. Nabízí se tedy použití automobilu, vlaku, autobusu, tramvaje, případně jízdního kola či chůze. Každý z těchto způsobů je v určitých situacích výhodnější než jiný, protože ne vždy je potřeba, aby byla cesta co nejrychlejší. Dalšími kritérii, která ovlivňují, který dopravní prostředek zvolíme, jsou dle [44] dostupnost, spolehlivost, bezpečnost, cena, integrace, pohodlí, velikost zavazadlového prostoru a přístupnost.

Výhody multimodální dopravy jsou takové, že dokážeme snížit rychlost přepravy a tím i efektivitu například tak, že můžeme lépe reagovat na nějaké geografické překážky. Tím, že můžeme využít různých způsobů dopravy také dojde ke snížení ceny, kterou přeprava bude stát [45].

Nevýhodou poté může být to, že narůstá komplexita přepravy, neboť v přepravě v podstatě jakéhokoliv nákladu je nutné provádět přesuny mezi jednotlivými dopravními prostředky. Další nevýhodou je, že jednotlivé druhy dopravy obsluhují různí provozovatelé a může se stát, že dopravní prostředek bude v určitou chvíli nedostupný. Občas také může dojít k prodloužení času přepravy [45].

2.1 Vývoj moderní dopravy

Článek [26] říká, že první větší revoluce v dopravě se stala v 19. století a byla jí stavba kanálů, která propojila moře a pevninské vodstvo, což umožnilo přepravu zboží a lidí do těchto míst. Éra kanálů trvala zhruba 100 let než je nahradila železniční doprava.

První železnice byly postaveny na začátku 19. století a kanály převyšovaly v rychlosti, produktivitě a rovněž i času přepravy. S železnicemi rovněž přišla éra páry, uhlí a oceli. Během této doby byla postavena obrovská síť železnic, jejichž hustota nadále rostla a nezměnila se ani v současnosti. Železniční doprava dominovala zhruba do 30. let 20. století, kdy se jako způsob dopravy začal využívat automobil, který svou rychlostí, flexibilitou a ce-

nou deklasoval vlaky, co se individuální dopravy týče. Zatímco celková délka kanálů klesá a délka železniční sítě se příliš nemění, celková velikost silniční sítě stále stoupá. A to i s příchodem letecké dopravy, která je sice rychlejší, ale není tak flexibilní, neboť je vázaná na letiště.

Z pohledu dopravy osob je dopravní vývoj velmi podobný, kdy stále více lidí využívá auto a letadlo například k delším cestám. Naopak zájem o železnice a lodní dopravu, které byly v historii populární, postupně upadá [26].

Okolo 60. let 20. století se do dopravní infrastruktury dostávají i informační technologie, které se zpočátku používaly pro hledání cest a plánování. S příchodem internetu přestávají vznikat aplikace nezávisle pro jednotlivé uživatele a firmy, ale vznikají webové aplikace, které využívá vícero subjektů [28].

2.2 Současnost a trendy v multimodální dopravě

V současnosti je podle [15] snaha přidat do multimodální dopravy nové dopravní prostředky, které dříve nebyly kvůli technologickému pokroku možné. Uvažuje se například o dronech, které by se mohly používat v komerční sféře, nebo samořídicí auta, která již v současné době existují a dochází k jejich intenzivnímu testování. Je také kladen důraz na vytváření aplikací v digitálním světě jako jsou například dopravní plánovače ať už s vestavěným nákupem jízdenky či nikoliv, což umožní pasažérům porovnávat různé preference a ceny, čímž dochází ke zvýšení efektivity celého dopravního systému.

Hlavním tématem v multimodální dopravě posledních let je rovněž udržitelnost a životní prostředí. Cílem je snížit používání fosilních paliv a využívat ekologické způsoby přepravy, což znamená použití ekologických paliv nebo zvýšit počet cestujících, kteří k přepravě používají hromadnou dopravu. Dle údajů z roku 2022 totiž 90 % silniční dopravy využívá benzin, nebo naftu jako palivo. Mezi další trendy patří mobilita jako služba, inteligentní dopravní systémy a *Park and Ride*, které jsou detailněji popsány níže.

2.2.1 Mobilita jako služba

Koncept mobility jako služby, z anglického *Mobility as a service* (*MaaS*), je relativně nový a jehož cílem je v jediné aplikaci vyhovět všem dopravním požadavkům potenciálních uživatelů [35]. Aplikace by měla umět vyhledávat různé trasy, rezervovat si místa a také by měla nabízet možnost celou trasu zaplatit, ať už jednorázově nebo si cesty předplácet pomocí měsíčního předplatného. Charakteristiky tohoto konceptu jsou podle [35] následující:

1. **Integrace různých druhů dopravy** – aplikace by měla motivovat uživatele k použití veřejné dopravy ať už se jedná o vlaky nebo autobusy, ale také alternativní druhy dopravy jako například sdílená kola, sdílená auta nebo taxi. Měla by uživateli umožnit výběr mezi těmito druhy dopravy podle jeho preferencí.

2. **Výběr tarifů** – aplikace umožní 2 druhy tarifů. Buď si uživatel každý měsíc předplácí určitý počet kilometrů nebo minut, které může využít k přepravě, nebo může využít koncept *pay as you go* a tedy zaplatí pouze za služby, které skutečně využil.

3. **Jediná platforma** – veškeré funkce, které jsou uživateli nabízeny, se musí nacházet v jediné aplikaci ať už na mobilu nebo v prohlížeči.

4. **Více aktérů** – aplikace je založena na interakcích mezi různými typy lidí, ať už to jsou zákazníci, provozovatelé jednotlivých druhů dopravy nebo provozovatelé aplikací třetích stran.

5. **Použití různých technologií** – k dosažení mobility jako služby jsou použity různá zařízení, tedy mobily a počítače a různé technologie jako třeba internetové připojení, GPS, platební brány a různé databázové systémy.

6. **Orientace na požadavek** – jestliže si zákazník zvolí trasu na základě preferencí, tak by aplikace měla nalézt řešení, které je pro něj s ohledem na preference nejlepší bez ohledu na to, že obecně mohou existovat i lepší řešení.

7. **Registrace** – pro ukládání preferencí a spravování předplatného je třeba v aplikaci disponovat vlastním uživatelským účtem.

8. **Personalizace** – aplikace nabízí uživatelům různá doporučení a dopravní řešení na základě jeho profilu, chování a jeho předešlých cestách.

9. **Přizpůsobení** – ta umožňuje si uživateli nastavit si různé preference, které může opakovaně využívat. Tímto se aplikace stává atraktivnější pro potenciální zákazníky.

Příkladem implementace této služby je například aplikace `Transit`[1], která dokáže ve vybraných městech naplánovat trasu s možností platby přímo v ní. Podporována jsou hlavně města v USA, Kanadě a Francii, ale také jednotlivá města v jiných zemích [65]. V České republice je to Česká Lípa [65].

2.2.2 Inteligentní dopravní systémy

Trendem posledních let se také stávají inteligentní dopravní systémy, což je soubor různých technologií a senzorů, které ze získaných dat dokážou podniknout takové kroky, které mohou dopravu jako takovou v dané oblasti zefektivnit. Například mohou na základě dat získaných v reálném čase pomoci při plánování a optimalizaci trasy a umožní tak řidičům vyhnout se místům z vysokou hustotou dopravy [62].

Inteligentní dopravní systémy lze podle [62] použít i pro kontrolu semaforů, díky nimž lze regulovat plynulost dopravy a rychlost vozidel a tím například snížit spotřebu paliva. Tyto systémy mohou pomoci i cestujícím, protože ti mohou v reálném čase vidět hustotu provozu v dané oblasti a také třeba zpoždění spojů hromadné dopravy a jejich přesnou polohu. Ve velkých městech často bývají problémy s parkováním, proto tyto systémy uživatelům pomáhají hledat parkoviště, případně jim umožní si parkovací místo rezervovat a zaplatit parkovné online.

V Evropě funguje projekt C-Roads[2], který se snaží v Brně, ale i jiných evropských městech implementovat inteligentní dopravní systémy, aby byla zajištěna efektivita dopravy. Jedná se o zařízení, která se nacházejí jak v dopravních prostředcích, tak i na dopravní infrastruktuře a tato zařízení spolu dokážou komunikovat. Jedním z využití tohoto systému je ve složkách integrovaného záchranného systému, kdy řidiči mohou vidět, že se za nimi nachází sanitka ještě dříve, než by ji mohli spatřit. Pokud se vozidlo záchranného systému nachází v blízkosti semaforu, tak mu může dát zelenou a tím mu umožnit rychlejší a bezpečnější průjezd [39]. Zároveň tento koncept může být použit ve vozidlech hromadné dopravy a semafory mohou upřednostnit směr, kterým prostředek jede.

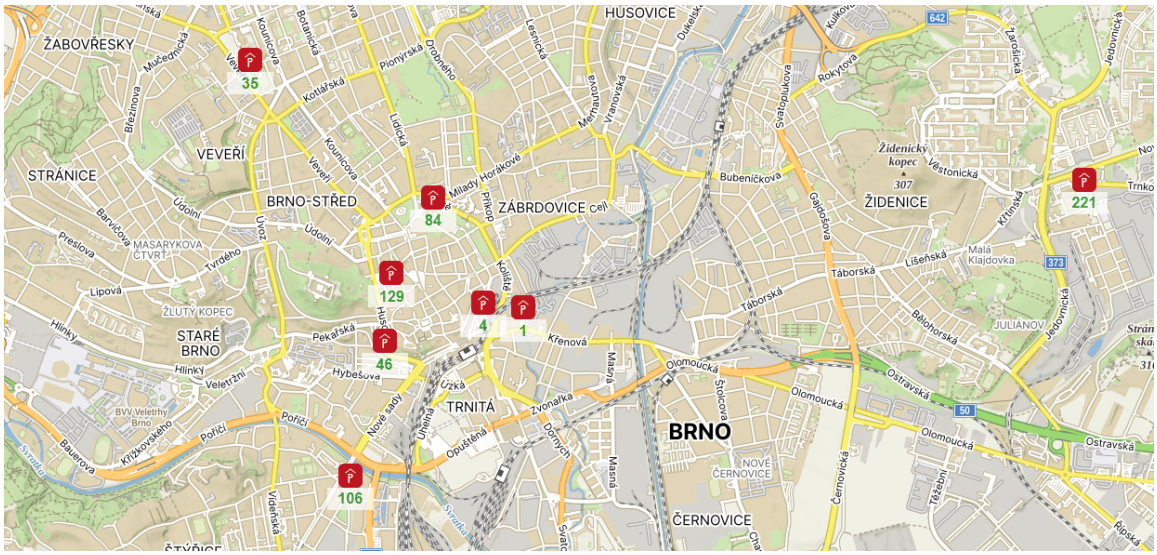
[1] https://transitapp.com/
[2] https://www.c-roads.eu/platform.html

2.2.3 Park and Ride

Dalším trendem, který se v poslední době rozmáhá, je koncept Park and Ride, neboli *„Zaparkuj a jeď“*, který tvoří kompromis mezi automobilovou dopravou a MHD. Zatímco automobilová doprava je zpravidla rychlejší, tak MHD je cenově výhodnější a ekologičtější. Ve velkých městech pak může být vysoký počet automobilů problém, neboť mají negativní vliv na životní prostředí, ve městě vznikají zácpy a také v nich bývá nedostatek parkovacích míst, tudíž nelze pohodlně zaparkovat. Koncept *Park and Ride* se tedy snaží městu ulehčit situaci a to tak, že zpravidla na okraji města je vybudováno velké množství parkovacích míst většinou ve formě parkovacích domů [47]. Tato místa mohou být zdarma, ale například v Brně jsou zpoplatněná a jsou stavěna blízko míst s velkou koncentrací zastávek, ze kterých je možno pokračovat pomocí MHD do centra města. Uživatel tak vůbec nemusí zajíždět do centra, ale může zaparkovat na jeho kraji v parkovacím domě a pro zbytek cesty využít hromadnou dopravu.

Park and Ride je vhodný pro lidi, kteří potřebují operovat v centru města, ale zároveň tam nechtějí jet autem kvůli zácpám a nemožnosti nalezení parkovacího místa. Tímto se ulehčí centru města od velkého množství vozidel, což vede i ke snížení ekologické zátěže [47]. Tímto může dojít i k nalákání více turistů, kteří budou chtít jet do centra, protože nemusí řešit, kde by v centru hledali parkovací místa. Nevýhodou může být to, že v blízkosti parkovacích domů dojde ke zvýšení hustoty dopravy ve snaze zaparkovat v parkovacím domě [47, 56].

Město Brno koncept *Park and Ride* podporuje a k datu 15. 11. 2024 se v něm nachází 9 míst, kde se dá zaparkovat[3]. Tento seznam se nachází na stránkách, kde Brno poskytuje otevřená data[4], která mohou být vhodná k dalšímu zpracování. Konkrétně u parkovacích domů jsou zaznamenány kromě jména a polohy také kapacita parkoviště a jeho obsazenost. Tyto informace jsou aktualizovány v reálném čase, takže je možné vidět, kolik je v dané chvíli na daném parkovišti volných míst. *Park and Ride* místa v Brně s počtem volných míst je zachycen na obrázku 2.1.



Obrázek 2.1: Parkovací domy v městě Brně s ukazateli počtů volných míst.

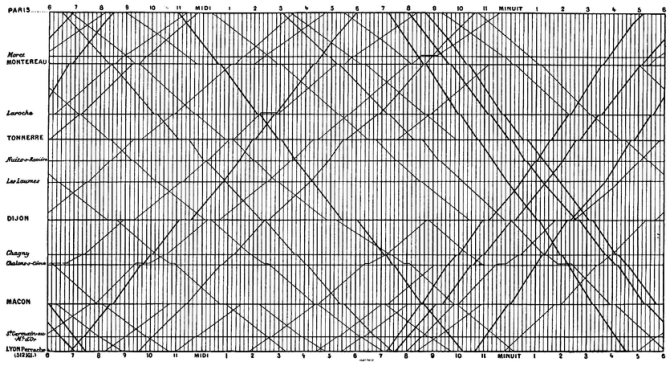
[3] https://data.brno.cz/datasets/821d12a2e7ca45db90b4e01ca2ebd93c_0/explore?showTable=true
[4] https://data.brno.cz/

2.3 Jízdní řády a jejich reprezentace

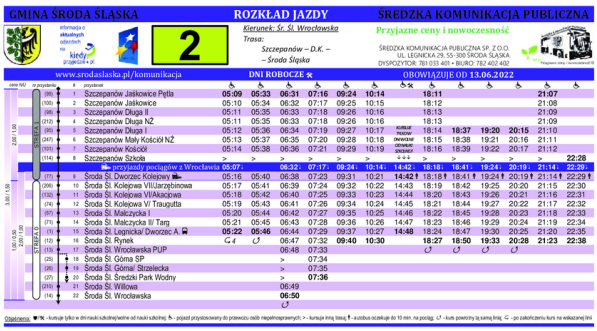
Aby multimodální doprava v dané oblasti fungovala efektivně a správně, tak je potřeba, aby byla nějak řízena. To se týká zejména hromadné dopravy, která není tak flexibilní a rovněž ji využívá mnoho lidí. Proto jsou zapotřebí jízdní řády, které obsahují informace o tom, kdy dané spoje jezdí a kde zastavují. Následující sekce se zabývají různými reprezentacemi jízdních řádů a pro koho jsou primárně určeny.

2.3.1 Analogová reprezentace

Do těchto reprezentací patří například jízdní řády na zastávce, které ukazují časy příjezdů spojů na danou zastávku. Případně se může jednat o informační tabule, které se nacházejí na nádražích. Stále také existují papírová vydání jízdních řádů v podobě knih, která mohou pokrývat jízdní řády celého kraje nebo okresu. Obrázky 2.2 a 2.3 ukazují reprezentací jízdního řádu v 19. století, respektive v moderní době.



Obrázek 2.2: Jízdní řád z 19. století z Paříže do Lyonu. Na horizontální ose je zobrazen čas, na vertikální jednotlivé zastávky. Čáry ukazují, odkud kam daný vlak jede a čím jsou čáry prudší oproti vodorovné ose, tím rychleji vlak jede [66]. Převzato z [34].



Obrázek 2.3: Ukázka moderního jízdního řádu. Převzato z [18].

2.3.2 Digitální reprezentace

Tato reprezentace není úplně vhodná pro samotné uživatele, pro něž může být nepřehledná, zato však mají pevnou strukturu, která je vhodná pro strojové zpracování, takže se hojně

používají například v dopravních plánovačích. Formátů jízdních řádů v digitální podobě je hned několik, avšak mezi nejznámější patří GTFS a NeTEx.

GTFS

Jihomoravský kraj a spoustu jiných regionů v České republice a ve světě zveřejňuje své jízdní řády ve formátu GTFS (*General Transit Feed Specification*). Používá je například Pražská integrovaná doprava[5] a na GTFS je dokonce postavené i celé Německo[6]. Jedná se o složku, která se skládá z následujících souborů (informace o nich byly převzaty z [50]):

- `agencies.txt` – obsahuje informace ohledně agentury či agentur, které jsou v daném regionu odpovědné za přepravu. Obsahuje jméno agentury, kontakty na ni a časovou zónu.
- `stops.txt` – zde se nacházejí informace o všech zastávkách, které se v daném dopravním systému nacházejí. Tento soubor obsahuje identifikátor zastávky, jméno, souřadnice, jestli zde zastavují vozidla umožňující bezbariérový nástup a výstup a identifikátor rodičovské zastávky.
- `routes.txt` – zde jsou zahrnuty veškeré linky, kterými se dá v daném dopravním systému přepravovat. Nachází se zde jejich identifikátor, identifikátor provozovatele, název linky a informace o tom, který dopravní prostředek je pro tuto linku využíván. Pro typ dopravního prostředku lze použít přímo čísla, která definuje dokumentace, například pro vlak, autobus nebo tramvaj, případně využít rozšířené typy tak, jak to má například Jihomoravský kraj, který dále rozděluje dopravní prostředky do dalších typů (například mezinárodní vlak, vysokorychlostní vlak, vlak převážející auta a jiné) [22].
- `trips.txt` – tento soubor již obsahuje informace o jednotlivých jízdách, tedy spojích, které využívají danou linku v určitou dobu. Obsahuje identifikátor linky a jízdy, dále název, který identifikuje, kam daný spoj jede, poté také směr, jestli je daný spoj bezbariérový a jestli do něj lze nastoupit s kolem. Důležitým atributem je také identifikátor služby, který odkazuje do souboru `calendar.txt`
- `stop_times.txt` – jedná se typicky o největší soubor, který již obsahuje samotné informace o časech příjedu a odjezdu. Nachází se zde identifikátor jízdy, identifikátor zastávky, čas příjezdu, čas odjezdu a pořadí zastávky pro danou jízdu.
- `calendar.txt` – tento soubor pro jednotlivé služby zobrazuje, ve které dny je v provozu (pro daný den hodnota 1) a pro které nikoliv (pro daný den hodnota 0) s datem počátku a datem konce. Toto je důležité, protože například některé linky nemusí jezdit o víkendu, případně mohou mít během něj jiný jízdní řád.
- `calendar_dates.txt` – obsahuje informace o výjimkách pro danou služby v daná data. V dané datum může být služba buď přidána, nebo odebrána.

GTFS definuje více dalších souborů, které jsou volitelné, ale Jihomoravský kraj používá pouze výše uvedené. Veškeré tyto soubory jsou však psány ve formátu `.txt`, ale jejich struktura odpovídá formátu `.csv`, tedy na prvním řádku se nachází seznam všech sloupců

[5] https://pid.cz/o-systemu/opendata/
[6] https://gtfs.de/en/

oddělených čárkami a na dalších řádcích se nacházejí záznamy, kde jsou čárkami oddělené jednotlivé atributy. Díky formátu zápisu a provázáním mezi soubory lze formát snadno strojově zpracovávat, případně z nich vytvořit databázi, kde každý soubor představuje jednu tabulku.

NeTEx

Jedná se o technický standard, který se zabývá sdílením dat ohledně veřejné dopravy a dat s nimi spojenými [1]. Data jsou ve formátu XML, což umožňuje snadné a rychlé strojové zpracování. NeTEx je rozdělen do šesti částí, z nichž každá se zabývá jiným aspektem hromadné dopravy [1].

1. topologie sítě hromadné dopravy,
2. jízdní řády,
3. informace o jízdném,
4. evropský profil o informacích o cestujících – určuje, které informace jsou relevantní pro poskytování informačních služeb pro cestující [9],
5. formát výměny alternativních druhů dopravy – specifikace výměny dat pro netradiční druhy dopravy (sdílená kola, sdílená auta, ...) [10],
6. evropský profil informací o přístupnosti cestujících – rozšiřuje evropský profil o informacích o cestujících o atributy týkající se přístupnosti vozidel pro osoby s postižením [11].

2.4 Existující dopravní plánovače

Tato sekce představuje různá řešení, která umožňují plánovat trasy s využitím více dopravních prostředků. Jedná se jak o české, tak i zahraniční aplikace.

2.4.1 Mapy Google

Aplikace Mapy Google[7] patří k nejpopulárnějším aplikacím svého druhu na světě a je dostupná jak v prohlížeči, tak i jako mobilní aplikace. Pro zobrazení map používá svůj vlastní podklad a je velmi známá svými funkcemi. Uživatelům nabízí detailní plánování tras pomocí auta, kola i hromadné dopravy. Pro auto dokáže najít nejvhodnější trasu i s ohledem na aktuální provoz.

Uživatelé rovněž díky ní mohou hledat zajímavá místa v okolí, či konkrétní služby jako například benzinové stanice nebo restaurace. Díky propojení celého Google ekosystému lze pro daná místa získat například i informace o otevírací době různých podniků nebo recenze na ně. Oproti jiným aplikacím podobného typu se Mapy Google pyšní svou funkcí *Street View*, která umožňuje virtuálně procházet vzdálená místa z pohodlí domova. To může být vhodné v situaci, kdy uživatel cestuje na místo, kde předtím nebyl, a může se s ním alespoň trochu seznámit dopředu. Výhodou rovněž je, že lze mapy v omezené míře používat i bez přístupu k internetu.

[7] https://www.google.com/maps/

Mezi nevýhody těchto map může podle [61] patřit to, že Google ukládá veškeré polohy uživatele i jeho historii, což může být problematické z hlediska soukromí. Mobilní aplikace je rovněž i poměrně energeticky náročná, takže se při jejím používání může rychleji vybít baterie. Problémem také může být to, že informace o různých bodech zájmu vyplňují samotní uživatelé a může se stát, že informace o nich mohou být zastaralé či dokonce zcela nepravdivé.

2.4.2 Mapy.cz

Alternativní aplikací k Mapám Google jsou Mapy.cz[8]. Jedná se o českou aplikaci od společnosti Seznam, která nabízí vlastní propracované mapové podklady založené na OpenStreetMap, které velmi dobře mapují cyklostezky a turistické trasy.

Je zde rovněž dostupné plánování trasy, které nabízí chůzi, kolo, auto, hromadnou dopravu a běžky, a umožňuje export tras do různých formátů. Aplikace rovněž umí během plánování zohlednit aktuální provoz a uzavírky. Tak jako u Map Google lze využít funkci Panorama, která funguje stejně jako *Street view*, avšak pouze s pokrytím pro Českou republiku.

Jejich mobilní aplikace umožňuje stáhnout mapy zemí dle výběru a mapy tak mohou fungovat i bez připojení k internetu. Nevýhodou této aplikace je však v nedávné době přidané předplatné, které sice nijak neomezuje běžné používání map, avšak limituje počet stažení map pro použití bez internetu, což je problém hlavně při jejich aktualizacích.

2.4.3 Citymapper

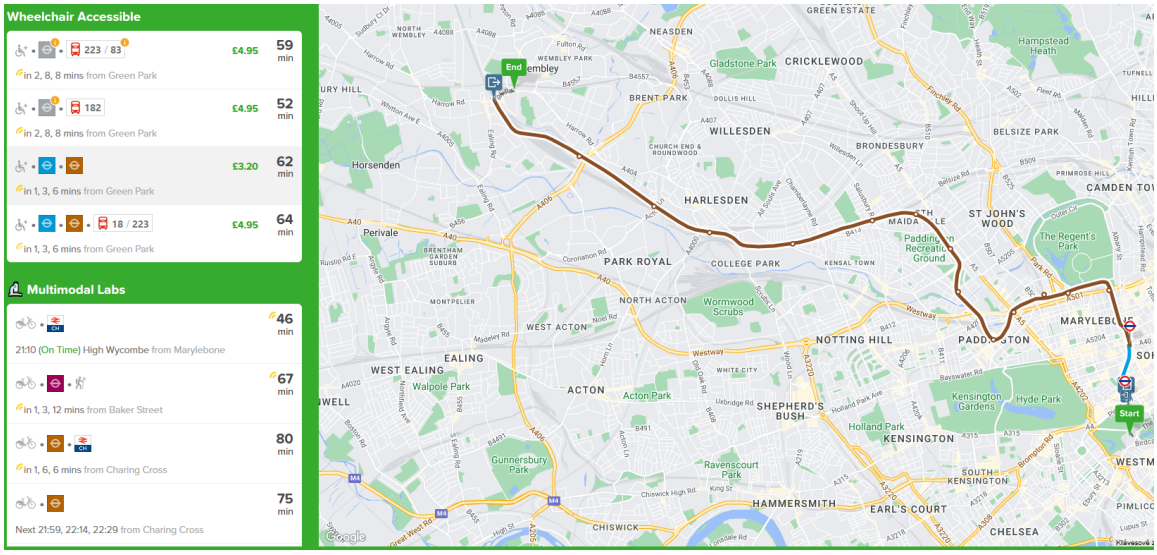
Jedná se o aplikaci[9], která je zdarma dostupná jak na webu, tak i na mobilních zařízeních s operačními systémy Android a iOS, která dokáže plánovat cesty a zobrazovat jejich trasy. Nabízí jak trasy, které využívají pouze jeden dopravní prostředek, tak i kombinaci dopravních prostředků. Dokáže zobrazit předpokládaný čas odjezdu, trasu i cenu. Zároveň také nabídne spoustu alternativních spojení například i s respektem k bezbariérovému přístupu [13].

Jedná se o velmi propracovanou aplikaci, která pracuje nad otevřenými daty, která dokáže naplno využít, což je výhodné například pro Londýn, kde se nachází mnoho způsobů přepravy jako například chůze, autobus, metro, auto a kolo. Problémem je však to, že aplikace podporuje pouze vybraná města a nedá se aplikovat na jakýkoliv region. Obrázek 2.4 ukazuje, grafické rozhraní plánovače.

2.4.4 IDOS

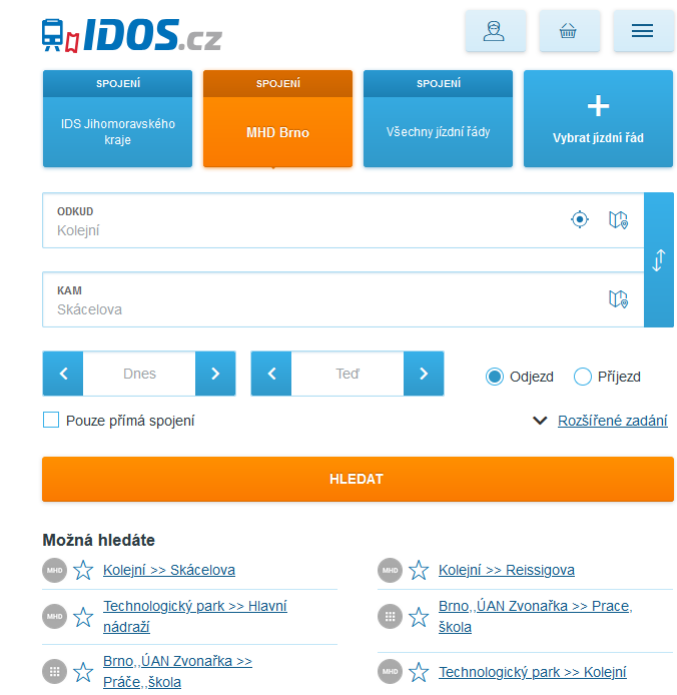
IDOS[10] je velmi populární aplikace, která umožňuje vyhledávání spojů, pokud jí zadáme počáteční a cílovou zastávku. Ve výchozím nastavení vyhledává spojení nezávisle na dopravním prostředku, ale lze upravit pro spoje využívající jeden konkrétní prostředek. Podporuje vyhledávání jízdních řádů po celé České republice a nabízí také jízdní řády pro městskou hromadnou dopravu vybraných měst. Je možné také zvolit průjezdní zastávky a zobrazit celou trasu na mapě. Aplikace také dokáže spočítat celkovou cenu jízdy a je skrze ní umožněno i jízdenky zakupovat.

[8] https://mapy.cz
[9] https://citymapper.com/
[10] https://idos.cz/



Obrázek 2.4: Snímek aplikace Citymapper při plánování cesty z Buckinghamského paláce na ulici High Rd, Wembley. Napravo na mapě se nachází jedna z možných tras a nalevo je zachycena možnost využití bezbariérového přístupu a multimodální dopravy.

Výhodou této aplikace je rovněž to, že dokáže pokrýt jízdní rády celé České republiky, nabízí mnoho nastavení a umí zobrazovat i aktuální zpoždění spojů. Nevýhodou je, že aplikace pracuje pouze s prostředky hromadné dopravy a neintegruje zde automobilovou dopravu. Na obrázku 2.5 se nachází ukázka uživatelského rozhraní této aplikace.



Obrázek 2.5: Ukázka aplikace IDOS.

Kapitola 3

Vizualizace geografických dat

Vizualizace je technika pro tvorbu obrázků, diagramů nebo animací s cílem snáze přenést informace koncovému uživateli, čehož by se dosahovalo hůře například s použitím slov. Vizualizace se využívají v mnoha odvětvích, mezi které kromě informatiky patří také medicína nebo strojírenství.

Standard [20] říká, že geografická data jsou takové informace, které jsou explicitně, nebo implicitně asociovány s lokací vzhledem k Zemi. Rozeznáváme 2 základní typy geografických dat, a sice vektorová a rastrová. Vektorová data jsou data sestávající ze základních primitiv jako jsou body a čáry a intenzivně se používají v mapových portálech jako například OpenStreetMap a Mapy Google. Body lze použít pro zobrazení měst a bodů zájmů, zatímco čáry pro zobrazení ulic nebo řek [49]. Rastrová data jsou poté tvořena pixely a typicky se využívají u topografických dat, satelitních snímků nebo při zkoumání různých oblastí [49].

3.1 Vizualizace prostorových primitiv

Velmi často je potřeba v geografických informačních systémech zobrazit nebo zvýraznit důležitá data, která se nacházejí na určitém místě na mapě. Základními kameny pro vizualizaci těchto dat jsou geografická primitiva jako jsou body, čáry, plochy a pole. V následujících částech je pro každé primitivum popsáno jeho využití a techniky, které se pro vizualizaci používají s cílem vizualizovat různé typy dat.

3.1.1 Body

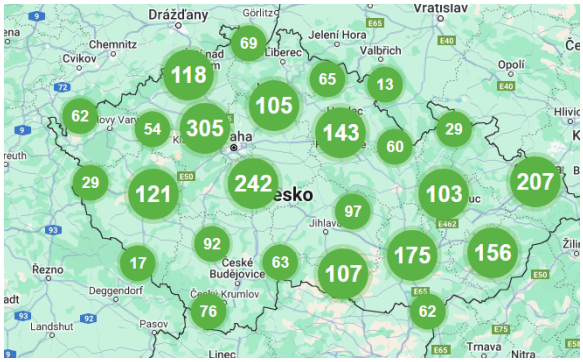
Bod lze pro vizualizaci použít v případě, že nás zajímá lokace jednoho konkrétního objektu. Různé body mohou mít různou důležitost a je vhodné je vizualizovat tak, aby uživatel věděl, který z bodů je na mapě výraznější, a tedy i důležitější [33]. Při zobrazení nějaké hierarchie mohou být méně důležité body menší a naopak důležitější body větší a výraznější jako tomu je na obrázku 3.1. Příkladem může být zobrazení měst, kde velikost bodu udává velikost města.

Bod lze také zobrazit pomocí ikony, jejíchž symbol může lépe ukázat, o jaká data se jedná. Ikony rovněž mohou mít různé velikosti, což ukazuje důležitost bodu, případně barvu pro různé typy dat. Například pro zobrazení lokace nějakého objektu je typicky používána ikona špendlíku [33].

Někdy se může stát, že je hustota bodů na daném místě příliš velká, což může vadit, když uživatel mapu oddálí, protože pak je pro něj celá vizualizace nepřehledná. Zároveň při velkém počtu bodů může docházet ke ztrátě výkonu a pohyb po mapě nemusí být

plynulý. Řešením je body shlukovat, kdy dochází k vytvoření jednoho reprezentativního bodu (shluku), který sdružuje všechny body pod sebou [19]. Shluky pak mohou být různě velké podle toho, kolik bodů pod sebou zastřešuje. To lze vidět na obrázku 3.2.

Obrázek 3.1: Ukázka velikosti bodu v závislosti na jeho důležitosti.



Obrázek 3.2: Mapa prodejen pro potravinovou sbírku. Každý shluk obsahuje počet bodů, které sdružuje. Po přiblížení by došlo k rozdělení jednotlivých shluků na menší části. Převzato z [19].

3.1.2 Čáry

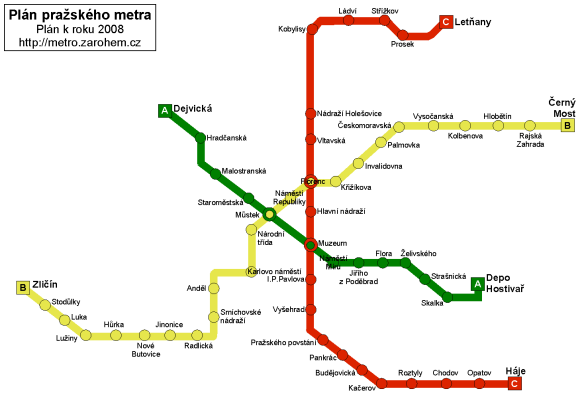
Čáry (tzv. *lines, polylines*) se používají pro zobrazení takových objektů, které se skládají z uspořádané posloupnosti bodů a nemá smysl je vizualizovat jednotlivě. Typicky se jedná o hranice států, řeky nebo cesty. Tak jako body, i čáry mohou být odlišeny podle důležitosti účelu například díky tomu, že jsou tlustší, mají jinou barvu nebo mají například jiný styl vykreslení, což ilustruje obrázek 3.3.

Pro snížení komplexity a celkové přehlednosti se ve speciálních případech používají čáry, kdy se každá hrana může ohnout pouze o určitý úhel (například o 45°, 60°, 90°). Tento přístup využívají mapy linek metra, kde nezáleží na geografické poloze jednotlivých stanic, ale spíše na času příjezdu z jedné do druhé [33], což lze vidět na obrázku 3.4.

3.1.3 Plochy a pole

Plochy se dají použít k vizualizaci všemožných uzavřených tvarů, ať už se jedná o rybníky, státy nebo celé kontinenty. Rovněž jsou vhodné pro vizualizaci skupin se stejnými nebo

Obrázek 3.3: Zachycení různých typů čar použitých pro odlišení dat. Frekventovanější silnice mají žlutou barvu a naopak ty méně frekventované bílou. Koleje jsou zde zobrazeny pomocí opakování černé a bílé barvy a přerušovaně jsou znázorněny menší turistické stezky.



Obrázek 3.4: Ukázka tras metra, kde se využívá ohyb čar pouze o určitý úhel. Převzato z [37].

podobnými daty, takže tak lze například rozlišit vodu od pevniny [33]. Jejich hranice jsou tvořeny posloupností čar spojených za sebou, čímž vzniká polygon.

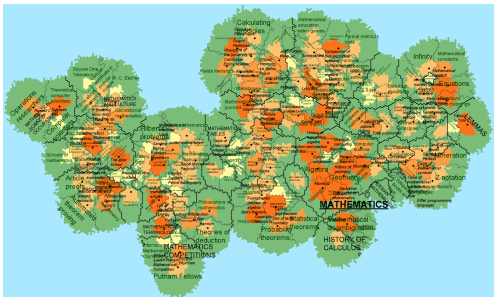
Nepravidelné plochy lze generovat přímo pomocí bodů, které lze spojit do polygonu nebo pomocí mnoha malých buněk pravidelného tvaru (nejčastěji šestiúhelníku), kdy je požadovaná plocha složena z těchto šestiúhelníků, která mají stejná, nebo podobná data (obrázek 3.5a) [33].

Pro vyšší čitelnost a lepší interpretaci dat může docházet ke zjednodušení plochy. Typicky dochází k převodu nepravidelných a členitých ploch na základní geometrické tvary [33]. Velikost vzhledem k ostatním plochám je však zachována, viz obrázek 3.5b.

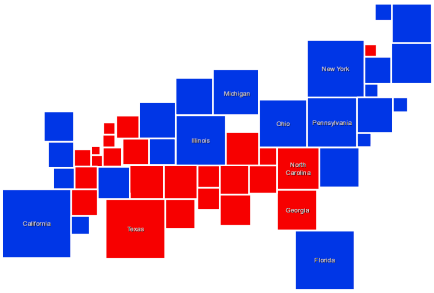
Pole se od plochy liší tím, že se v něm nacházejí spojitá data, která se v oblasti plynule mění. Pro vizualizaci lze použít barvy (viz obrázek 3.8a), kterými dojde k odlišení různých hodnot dat, nebo do ní přidat izolinie, které seskupí data se stejnými hodnotami a oddělí data s jinými [33]. Příkladem mohou být vrstevnice na horách, které sdružují body se stejnou nadmořskou výškou. Když jsou vrstevnice blízko sebe, znamená to, že má hora v daném místě větší sklon.

Specifickými technikami vizualizace polí je například roztažení, kdy je oblast roztažena více, či méně podle hodnot dat (obrázek 3.6b) [33]. Případně lze využít hustoty bodů,

kdy data s vyšší hodnotou budou charakterizovány větší koncentrací bodů u sebe, zatímco u jiných dat bude koncentrace bodů nižší (obrázek 3.6c) [33].

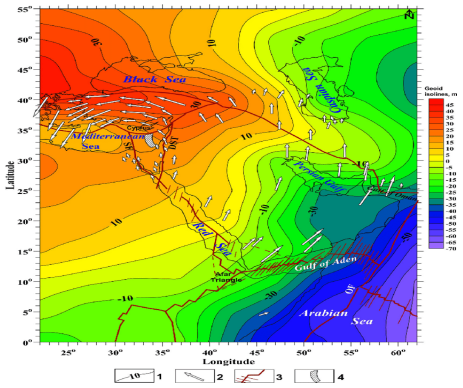


(a) Plochy – vytvoření nepravidelné struktury z menších pravidelných buněk. Převzato z [55].

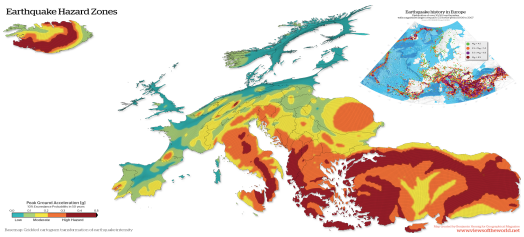


(b) Plochy – převod na základní geometrické tvary. Převzato z [24].

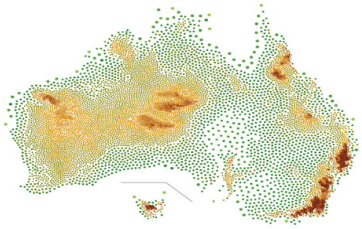
Obrázek 3.5: Ukázka vizualizace ploch.



(a) Pole – ukázka vrstevnic (izolinií). Převzato z [21].



(b) Pole – roztažení oblasti. Převzato z [70].



(c) Pole – body z vyšší hustotou představují data s vyššími hodnotami. Převzato z [27].

Obrázek 3.6: Ukázka vizualizace polí.

3.2 Ukládání geografických dat

Aby bylo možné geografická primitiva správně vizualizovat, tak je potřeba vědět, jak souřadnice zakódovat pro plochu, na kterou budou vizualizovány, čemuž se věnuje sekce 3.2.1.

Jakmile je toto splněno, tak je třeba tato primitiva převést do formátu, kterému rozumí vizualizační nástroje. Sekce 3.2.2 se zabývá nejčastěji používanými formáty.

3.2.1 Kódování souřadnic

Každé geografické primitivum je charakterizováno svými souřadnicemi, popisující jejich zeměpisnou šířku a délku. Tyto souřadnice se však mohou vztahovat k různým souřadnicovým systémům a rovněž mohou být různým způsobem zakódované třeba kvůli jejich velikosti.

Výběr souřadnicového systému

Jestliže jsou souřadnice daného objektu určeny k zobrazení na mapě, je třeba využít systém EPSG:3857 známý jako webová Mercatorova projekce, který mapuje povrch země na rovinu. Pokud jsou ale určena pro zobrazení na kouli, je využito EPSG:4326 (WGS84) [36]. Například souřadnice města Brna jsou v systému EPSG:4326 `[49.195061, 16.606836]`, kde první číslo reprezentuje zeměpisnou šířku a druhé zeměpisnou délku. V případě EPSG:3857 jsou poté souřadnice `[1848664.53957, 6308024.22914]`.

Přímé kódování

Tento způsob kódování je nejjednodušší, neboť nedochází k žádné konverzi souřadnic a jsou přenášeny tak, jak jsou. Typicky se jedná o pole polí, kde každá položka obsahuje bod ve formátu `[lat, lon]`, kde `lat` je zeměpisná šířka a `lon` zeměpisná délka. Avšak například formát `GeoJSON` vyžaduje souřadnice v obráceném pořadí, tedy `[lon, lat]`.

Kódování polyline

Pokud je sekvence souřadnic dlouhá, tak může zabírat hodně paměti. Pro tyto případy lze použít kódování `polyline`. Toto kódování převádí souřadnice na posloupnost znaků. Výhodou tohoto kódování je, že dochází ke kompresi souřadnic a zaberou tak méně paměti. Tomu ještě pomáhá to, že pokud je souřadnic více, tak se vždy ukládá pouze rozdíl mezi aktuální a předchozí hodnotou. Komprese je však ztrátová, tedy dochází ke ztrátě přesnosti souřadnic.

Například postup zakódování hodnoty -179.9832104 je dle [58] následující:

1. Vynásobení hodnoty číslem 10^5 a tedy i zaokrouhlení hodnoty – `-17998321`.
2. Konverze do binární formy. Kladná čísla jsou zapsána přímo a záporná musí být zapsána v doplňkovém kódu – `11111110 11101101 01011110 00001111`.
3. Posunutí o 1 bit doleva – `11111101 11011010 10111100 00011110`.
4. Jestliže původní číslo bylo záporné, probíhá inverze bitů – `00000010 00100101 01000011 11100001`.
5. Rozdělení bitů do pětic, tyto pětice se začínají tvořit zprava – `00001 00010 01010 10000 11111 00001`.
6. Obrácení pořadí těchto pětic – `00001 11111 10000 01010 00010 00001`.
7. Provedení operace OR každé z těchto pětic s hodnotou 32 (`100000`) pouze pokud se nejedná o poslední pětici. Pro každou neposlední pětici tedy přibyde jedničkový bit,

který dekodéru říká, že ještě není u konce a že následují ještě další pětice – `100001 111111 110000 101010 100010 000001`.

8. Převod každé ze šestic do desítkové soustavy – `33 63 48 42 34 1`.
9. Přičtení hodnoty 63 ke každé této hodnotě – `96 126 111 105 97 64`.
10. Konverze každé z těchto hodnot do znakové podoby podle ASCII – `‘õia@`.

Například souřadnice `[49.195061, 16.606836]` se pomocí polyline kódování zakódují jako `clgkHwojdB`. Pokud by došlo k zakódování posloupnosti `[[49.195061, 16.606836], [49.195061, 16.606836]]`, výsledkem bude `clgkHwojdB??`, tedy díky ukládání rozdílu mezi souřadnicemi se nový kód rozšířil o pouhé 2 znaky.

3.2.2 Formáty pro ukládání geografických dat

Pro zajištění kompatibility napříč různými programy byly vytvořeny formáty, které obsahují specifickým způsobem geografická data ukládají. V této sekci jsou popsány nejpopulárnější z nich, a sice `GeoJSON`, `GPX` a `KML`.

GeoJSON

Formát `GeoJSON` je založen na formátu `JSON` a obsahuje speciální údaje, pomocí kterých vizualizační nástroje dovedou zobrazit daný obsah na mapě. Příklad zápisu jednoho bodu se nachází ve výpisu 3.1. Následující informace o tvaru a parametrech tohoto formátu jsou převzaty z [8].

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {},
      "geometry": {
        "coordinates": [
          16.61992,
          49.20787
        ],
        "type": "Point"
      }
    }
  ]
}
```

Výpis 3.1: Příklad reprezentace jednoho bodu ve formátu GeoJSON.

Prvním důležitým parametrem je položka `features`, do které se dávají informace o primitivech, která mají být zobrazena. Jeden soubor ve formátu `GeoJSON` dokáže zobrazit více primitiv. Každá položka v parametru `features` obsahuje klíč `type`, který říká, o které primitivum se jedná. Od toho se rovněž odvíjí parametr `coordinates`. Typy jsou pro `GeoJSON` následující:

- `Point` – bod, jeho souřadnice je pole dvou bodů, kde na první pozici je zeměpisná délka a na druhé zeměpisná šířka.
- `MultiPoint` – kolekce bodů, v `coordinates` se nachází pole bodů.
- `LineString` – čára, souřadnice jsou ve stejném formátu jako `MultiPoint`.
- `MultiLineString` – kolekce čar, souřadnice jsou typu pole čar.
- `Polygon` – mnohoúhelník, který označuje hranice plochy, souřadnice jsou ve stejném formátu jako `LineString` s tím rozdílem, že `Polygon` musí sestávat alespoň ze 4 bodů a první bod musí být stejný jako poslední, aby se jednalo o uzavřenou plochu.
- `MultiPolygon` – kolekce polygonů, souřadnice jsou jako pole polygonů.

GPX

Jedná se o formát, který je založen na formátu XML používající se pro výměnu dat mezi aplikacemi a webovými službami na internetu [14]. Primitiva se zde podle [14] dělí do 3 skupin:

- `Waypoint` – reprezentuje bod, kromě informací o jeho poloze se zde mohou nacházet i další atributy jako třeba nadmořská výška, název bodu nebo jeho popis.
- `Route` – uspořádaný seznam bodů vedoucí k cíli. Díky posloupnosti bodů, které se chovají jako uzly dané cesty, lze uživateli ukázat směr, kterým se mají vydat.
- `Track` – jedná se opět o uspořádaný seznam bodů stejně jako `route`, ale tento prvek se používá v případě, že někdo danou cestu dokončil a nahrál například během jízdy na kole. Body uvnitř tohoto primitiva obsahují informace o poloze, nadmořské výšce a času průjezdu, z čehož lze poté získat průjezd danými body v čase a celkové převýšení na trase. Ukázka nahrané trasy lze vidět ve výpisu 3.2.

```
<trk>
    <name>Horní Bečva k~odbočce na Bílou</name>
    <type>road_biking</type>
    <trkseg>
        <trkpt lat="49.45534" lon="18.16266">
            <ele>411.40</ele>
            <time>2025-05-02T12:33:42.000Z</time>
        </trkpt>
        <trkpt lat="49.45533" lon="18.16269">
            <ele>420.20</ele>
            <time>2025-05-02T12:33:43.000Z</time>
        </trkpt>
      <trkpt lat="49.45532" lon="18.16272">
            <ele>420.80</ele>
            <time>2025-05-02T12:33:44.000Z</time>
        </trkpt>
    </trkseg>
```

```
</trk>
```

Výpis 3.2: Zjednodušená ukázka nahrané cyklotrasy. Trasa (`trk`) se skládá ze segmentů (`trkseg`), které v sobě obsahují seznam bodů (`trkpt`). Ty obsahují údaje o nadmořské výšce a času průjezdu.

KML

KML (*Keyhole Markup Language*) je rovněž založen na formátu XML, který se používá například pro zobrazení dat *Google Earth* [38]. Skládá se z podobných primitiv jako `GeoJSON`, která mohou obsahovat další metadata jako jméno nebo popis. Primitiva jsou podle [38] nejčastěji zabalena do prvku `Placemark`. Tento prvek obsahuje jméno a popis společně s primitivem, které charakterizuje jeho souřadnice. Nejčastější primitiva jsou `point`, `path`, `polygon`, jejichž význam je stejný jako pro formát `GeoJSON` popsaný v 3.2.2. Příklad zápisu bodu lze nalézt ve výpisu 3.3.

Těmto primitivům lze nastavit i vzhled pomocí prvku `Style`, kde jim lze nastavit například šířku nebo barvu. Dalším prvkem, který lze použít, je prvek `GroundOverlay`, díky kterému je možné zobrazovat různé ikony, případně celé obrázky.

```
<kml xmlns="http://www.opengis.net/kml/2.2">
    <Placemark>
        <name>Simple placemark</name>
        <description>Attached to the ground. Intelligently places itself
            at the height of the underlying terrain.</description>
        <Point>
            <coordinates>-122.08220,37.42229,0</coordinates>
        </Point>
    </Placemark>
</kml>
```

Výpis 3.3: Ukázka zápisu bodu ve formátu `KML`. Převzato z [38].

3.3 Mapa

Základním prvkem, na kterém celá vizualizace geografických dat staví, je mapa. Díky ní uživatel získá větší představu o tom, jak spolu více objektů, které se na ní nachází, souvisí. Lze odtud získat informace jako je vzdálenost mezi dvěma objekty, směr nebo velikost objektu [40]. Mapy, které se dají použít pro vizualizaci, se dají rozlišit do dvou skupin, kterými jsou referenční a tematické.

3.3.1 Referenční mapy

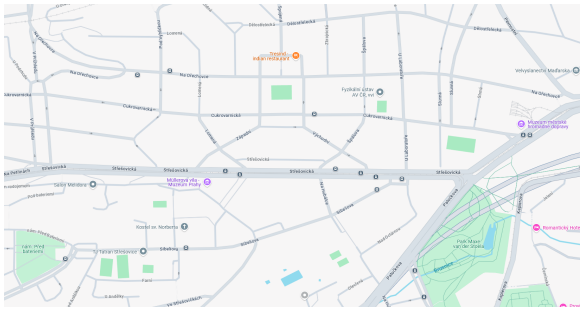
Tyto mapy zobrazují informace, které se týkají fyzických a politických charakteristik dané oblasti. Na mapě se tedy mohou nacházet hranice států, města, cesty, přírodní objekty jako například řeky, hory a mnohé další [32]. Do této skupiny spadají následující mapy:

- **Fyzické mapy** – tyto mapy jsou zaměřeny na přesné zobrazení krajiny v dané oblasti, to znamená, že jsou barevně odlišena místa s rozdílnou nadmořskou výškou. Čím tmavší barva, tím má daná oblast vyšší nadmořskou výšku [32]. Příklad fyzické mapy Evropy lze vidět na obrázku 3.7a,

- **Politické mapy** – tyto mapy obsahují informace o hranicích států, důležité cesty nebo vodní toky. Jsou to typicky i mapy, které se tisknou a ukazují lidem geografii nějaké oblasti [32]. Obrázek 3.7b ukazuje politickou mapu USA.
- **Mapy ulic, cest a dálnic** – tyto mapy kladou převážně důraz na přehledné zobrazení ulic, cest a dálnic a dominují digitálnímu světu. Mezi největší hráče patří mapy[1] od společnosti Google (obrázek 3.7c). V České republice jsou populární například Mapy.cz[2] od společnosti Seznam.

(a) Fyzická mapa. Převzato z [23].

(b) Politická mapa. Převzato z [57].



(c) Mapa ulic od společnosti Google.

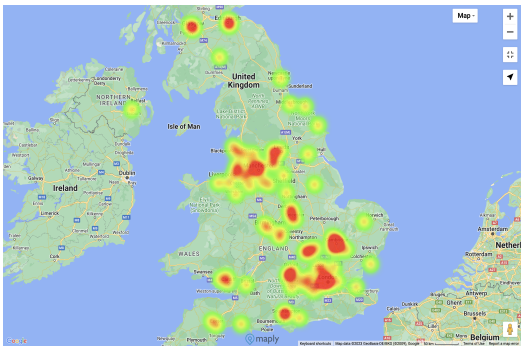
3.3.2 Tematické mapy

Tematická bývá většinou založena na mapě referenční, ale zobrazuje jevy, které se svými hodnotami v různých oblastech liší, a nejsou na referenční mapě viditelné [32]. Typicky se jedná o socioekonomické nebo přírodní jevy. V této skupině se opět nachází různé typy map jako například proporcionální symbolová mapa nebo choropletová mapa. Avšak nejdůležitější jsou následující:

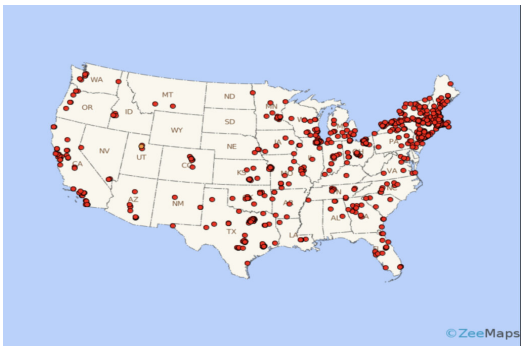
- **Teplotní mapy** – tyto mapy se používají k zobrazení hustoty něčeho. K indikaci se používají barvy a čím je barva tmavší, tím je hustota vyšší. Využití naleznou například v analýzách trhu nebo v meteorologii (obrázek 3.8a) [32].
- **Bodové mapy** – tyto mapy se rovněž mohou použít k vizualizaci hustoty, nicméně místo barev se používají body. Oblasti s vysokou hustotou se poznají tak, že se v nich

[1] https://www.google.com/maps
[2] https://mapy.cz/

nachází mnoho bodů, a jsou blízko u sebe. Výhodou tohoto typu mapy je, že umožňuje přesnější vizualizaci oproti barvám a zároveň je také vhodná pro lidi trpící barvoslepostí [32]. Obrázek 3.8b ukazuje bodovou mapu pro USA.



(a) Teplotní mapa. Převzato z [30].



(b) Bodová mapa. Převzato z [17].

3.4 Geografické informační systémy

Pokud je potřeba pracovat s geografickými daty, tak je nutné je nejprve získat, někam je uložit, udržovat je a v neposlední řadě na nich provést analýzy a přinést výstupy, které z těchto analýz vyplývají. Dříve bylo zpracování analogové, veškeré analýzy se prováděly manuálně a výstupy byly většinou v podobě papírových map [59]. Pro komplexnější analýzy však mohou být vyžadována data z různých zdrojů a mohou být v různých měřítkách a projekcích. Manuální konverze těchto dat by byla časově náročná a drahá.

K tomuto lze využít geografický informační systém (GIS), neboť mapy budou uloženy digitálně a jakákoliv konverze měřítka nebo projekce je prováděna pomocí programů, což je oproti manuálním konverzím rychlejší [59]. Existuje mnoho různých definic geografického informačního systému podle jeho schopností a účelu použití. Například dle [3] je geografický informační systém definován jako počítačový systém, který umožňuje získávání dat, jejich uložení a modifikaci, analýzu a prezentaci výstupů z těchto analýz. Jiná definice poté hovoří, že GIS je systém pro záznam, ukládání, kontrolu a kontrolu dat, jejich manipulaci, analýzu a zobrazení, přičemž tato data musí být prostorově vztažena k Zemi [59].

Cílem geografických informačních systémů je dle [59]:

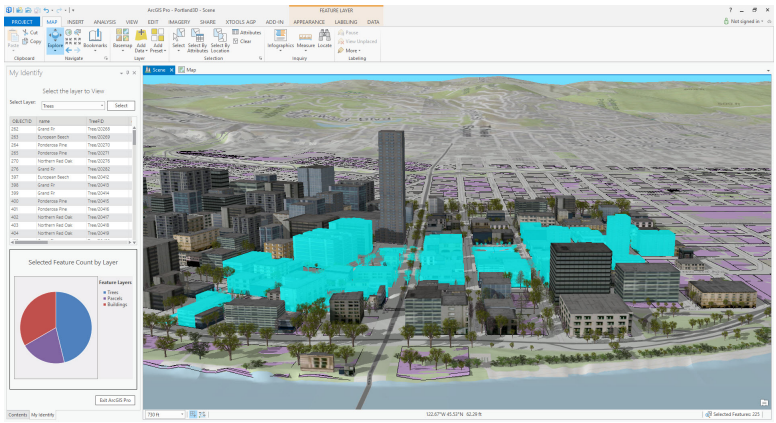
- maximalizovat efektivitu plánování a usnadnění rozhodování,
- poskytnout prostředky pro efektivní distribuci a zpracování dat,
- minimalizovat duplikaci dat,
- integrovat informace z různých datových zdrojů,
- umožnit provádění komplexních analýz a dotazů nad geografickými daty s cílem generovat nové informace.

Geografický informační systém být dle [40] použit k následujícím úlohám:

- **Identifikace** – schopnost získat informaci o objektu, ať je to třeba jeho jméno či jiné atributy, po najetí na určitou lokaci na mapě.

- **Lokace** – na základě různých dotazů je možno získat lokaci objektu, která je identifikována jeho souřadnicemi.
- **Určování trendů** – je možno analyzovat změny v průběhu času v dané oblasti.
- **Nalezení optimální cesty** – je možné získat cestu mezi 2 body s nejnižší cenou, ať už se jedná o silniční síť nebo třeba potrubní síť.
- **Hledání vzorů** – v geografických informačních systémech lze analyzovat různé vztahy mezi daty, které mohou pocházet z různých zdrojů. Například je možné analyzovat vztah mezi polohou továren a lokálním mikroklimatem.
- **Tvorba modelů** – je rovněž možné provádět různé predikce a plány do budoucna. Příkladem může být přizpůsobení stávající sítě hromadné dopravy, jestliže se ve městě postaví nová čtvrť.

Mezi nejznámější geografické informační systémy patří `ArcGIS`[3], což je profesionální program společnosti Esri, který umožňuje komplexní analýzu, zpracování a vizualizaci geografických dat. S `ArcGIS` lze vytvářet mapy, ať už 2D nebo 3D, provádět prostorovou analýzu, případně sdílet data mezi uživateli [2]. Obrázek 3.9 ukazuje, jak vypadá tvorba 3D mapy v `ArcGIS`.



Obrázek 3.9: 3D mapa v `ArcGIS`. Převzato z [63].

Konkrétně se dá `ArcGIS` použít například pro monitorování stavu cest a mostů nebo pro analýzu míst s vyšší kriminalitou, do kterých může být vysláno více policistů nebo tvorbu cyklomap pro lidi dojíždějící do práce [5].

[3] https://www.arcgis.com/index.html

Kapitola 4

Výpočet nejkratší cesty

Výpočet nejkratší cesty je složitý problém, jehož cílem je nalézt nejkratší cestu mezi startovacím a cílovým bodem. Tento problém je zpravidla reprezentován pomocí grafu, který se skládá z uzlů a hran, a uzly mohou, ale i nemusí být hranami propojené. Hranám lze přiřadit i váhy, které reprezentují jejich vlastnosti, například vzdálenost mezi dvěma uzly, které spojuje [46]. V kontextu dopravy lze graf vnímat například tak, že uzly reprezentují města a hrany reprezentují cesty, které mezi nimi vedou. Znalost nejkratší cesty mezi dvěma body je v dopravě velmi důležitá, neboť díky ní dochází k úspoře času a peněz. Tato kapitola se zaměřuje na algoritmy, které se používají v automobilové dopravě a v městské hromadné dopravě.

4.1 Výpočet nejkratší cesty pro automobilovou dopravu

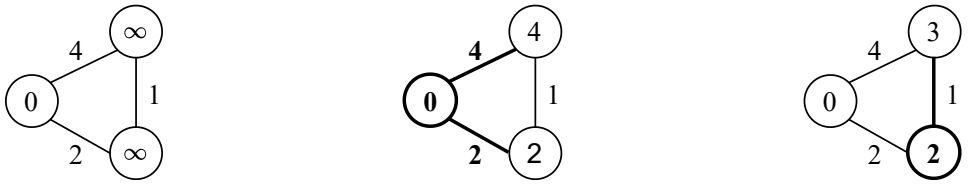
Tyto algoritmy mohou být kromě automobilu použity i pro jiné druhy dopravy, které se neřídí jízdním řádem. Lze je použít i například pro chůzi nebo jízdu na kole. Nejpopulárnějšími algoritmy pro tuto oblast je Dijkstrův algoritmus a algoritmus A*.

4.1.1 Dijkstrův algoritmus

Cílem tohoto algoritmu je nalézt cesty s nejnižší cenou, které vedou z počátečního bodu do všech ostatních. Na jeho vstupu se nachází graf. V případě silniční sítě může být hodnotícím kritériem dané hrany například její reálná délka. Algoritmus funguje podle [52] následovně (příklad se nachází na obrázku 4.1):

1. Nejprve dojde k označení počátečního bodu.
2. Dále se naleznou všechny body, které jsou k němu připojeny, a vypočítají se ceny všech hran, které k nim vedou. Poté dochází k označení uzlu, který má nejnižší cenu.
3. Dochází k výpočtu ceny, za kterou se lze dostat ze startovacího bodu do všech uzlů, které jsou spojeny s uzlem z kroku 1 a označení se ten, který má cenu z počátečního bodu nejnižší.
4. Opakuje se bod 3, dokud není označen cílový uzel.

Výstupem je poté celková cena spolu s uzly, které k danému cíli vedou [52]. Výhodou algoritmu je, že najednou získáme nejkratší cesty ke všem ostatním uzlům kromě startovacího. Nevýhodou však je, že je potřeba projít všechny uzly, což například pro mapu České



Obrázek 4.1: Ukázka Dijkstrova algoritmu. Start je z uzlu nalevo a jeho vzdálenost od startovacího uzlu je 0. U ostatních uzlů je vzdálenost od startu inicializována na ∞. Ve druhém kroku dochází k ohodnocení uzlů, na které vidí počáteční uzel. Vzdálenosti od počátků jsou tedy 4 a 2. V posledním kroku je však nalezena kratší cesta k vrchnímu uzlu přes uzel spodní. Nejkratší vzdálenost vrchního uzlu od počátku je tedy 3.

republiky může být velmi náročné, neboť složitost celého algoritmu je $O(n^2)$, kde n je počet uzlů v daném grafu [51, 52]. Navíc při běžném plánování tras dochází k výpočtu nejkratší cesty od startovacího bodu pouze k jednomu koncovému, tedy není nutné znát nejkratší cesty i k ostatním bodům.

4.1.2 A*

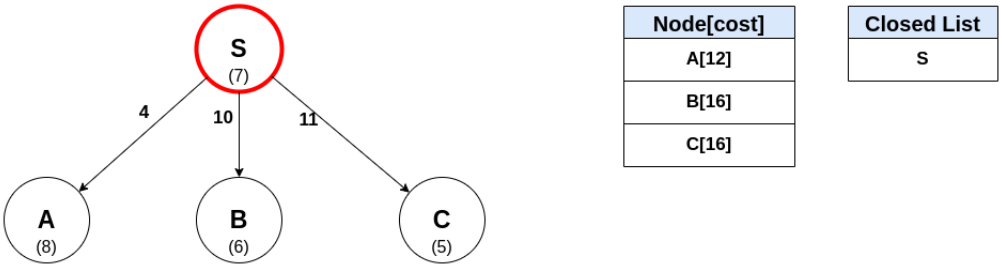
Na rozdíl od Dijkstrova algoritmu tento algoritmus hledá cestu s nejnižší cenou od jednoho počátečního do jediného koncového bodu. Dále se zde nepracuje pouze s cenou, kterou má daná hrana, a cenou, kterou stojí cesta z počátečního bodu do aktuálního, ale také s tzv. *heuristikou* [29]. Heuristika je určitá znalost daného problému, která vyhledává akce, vedoucí k efektivnímu řešení a vyhýbá se těm, které k efektivnímu řešení nevedou [43, strana 192]. V případě hledání nejkratší cesty je heuristikou například přímá vzdálenost mezi aktuálním a koncovým bodem [29]. V algoritmu A* se celková cena $f(n)$ vypočítá jako

$$f(n) = g(n) + h(n),$$

kde $g(n)$ je skutečná cena, kterou stálo dostat se z počátečního bodu do bodu n, a $h(n)$ je heuristická funkce, kterou lze aplikovat na každý uzel [29]. Algoritmus pak podle [29] funguje následovně (příklad se nachází na obrázku 4.2):

1. Nejprve se označí startovací uzel jako rozpracovaný a vypočte se jeho f.
2. Následuje volba nového uzlu n z rozpracovaných uzlů, který má f nejnižší. Pokud lze vybrat z více uzlů, je možné vybrat libovolný, ale jen když nejlepším uzlem není i cílový.
3. Jestliže je nejlepší uzel cílový, tak je označen jako navštívený a algoritmus je ukončen.
4. Z uzlu n jsou vygenerováni jeho potomci a pro každý z nich je vypočítáno f. Následně jsou tito následníci označeni jako rozpracovaní, pokud už náhodou nebyli navštíveni. Navštívený uzel je označen jako rozpracovaný pouze tehdy, je-li tento uzel potomkem n a jeho f je nižší než f v době, kdy byl označen jako navštívený. Poté dochází k návratu na krok 2.

K tomu, aby algoritmus poskytoval optimální řešení, je nutné zvolit heuristickou funkci takovou, aby $h(n) \leq p(n)$, kde $p(n)$ je skutečná vzdálenost z bodu n do cíle. Pokud tomu tak není, tak nalezené řešení nemusí být optimální [29].



Obrázek 4.2: Demonstrace algoritmu A*. Počáteční uzel S vygeneroval 3 uzly A, B a C. Každý z uzlů v sobě obsahuje hodnotu heuristické funkce $h(n)$ a každá hrana obsahuje cenu, za kterou se lze do daného uzlu dostat. Pro každý z uzlů se poté provede součet hodnot heuristické funkce a ceny příslušné hrany, která do něj vede, což ukazuje tabulka vpravo. Nejnižší součet má uzel A a proto nejkratší cesta povede přes něj. Algoritmus končí ve chvíli, kdy je vygenerován cílový uzel. Obrázek převzat z [64].

4.2 Algoritmy pro prostředky hromadné dopravy

V případě algoritmů pro hromadnou dopravu není cílem najít nejkratší cestu, neboť hromadná doprava jezdí ve většině případů po trasách, které jsou dané. Cílem je najít spoj, nebo posloupnost spojů takovou, která umožní dostat se z místa A do místa B za co nejkratší čas [6].

To, co algoritmy ze sekce 4.1 ignorují, jsou přechody na jiné spoje a také jízdní řády, které je třeba během plánování brát v úvahu, neboť spoje jezdí pouze v určité časy [6]. Výstupem těchto algoritmů je sekvence linek a přestupy mezi nimi společně s časy odjezdů a příjezdů tak, aby uživatel věděl kdy a kde nastoupit, vystoupit a přestoupit. Algoritmus rovněž může vrátit i trasu, kterou je třeba urazit, pokud ji lze získat.

4.2.1 Plánování tras veřejné dopravy na základě jízd

Jedním z algoritmů, který se používá pro efektivní výpočet nejefektivnější cesty z hlediska nejkratšího času příjezdu je algoritmus *Trip-Based Public Transit Routing*. Ten lze podle [69] rozdělit do několika fází.

1. Předzpracování – tento algoritmus nefunguje tak, že by se spustil celý při jednom požadavku a hned by vrátil výsledek. Nejprve musí dojít k předzpracování dat, aby mohly být výsledky na pozdější požadavky vypočteny rychleji. Prvním z kroků je výpočet množiny přestupů T, kdy se prochází všechny cesty a zastávky v nich a hledají se všechny cesty, na které se dá z dané zastávky na dané cestě přestoupit [69]. Tato množina je zpočátku velmi velká, takže dochází k jejich redukci a ve výsledné množině se poté nacházejí pouze přestupy, které jsou důležité a které umožňují nejrychlejší dobu příjezdu.

2. Dotaz pro nejkratší příjezd – tento dotaz může být pokaždé jiný a cílem je najít cestu takovou, že čas příjezdu bude co nejmenší, jestliže se na vstupu nachází počáteční

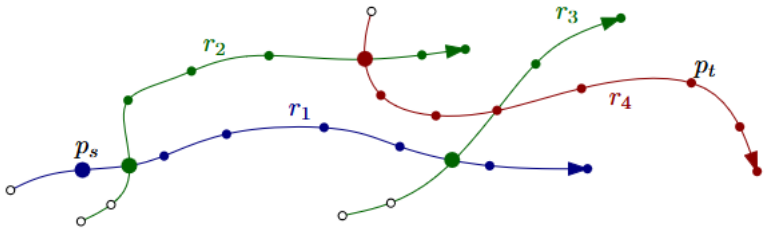
zastávka, cílová zastávka a čas odjezdu. Algoritmus poté vyhledá a zobrazí všechna řešení, která jsou Pareto-optimální. Paretova optimalita je stav, ve kterém nelze najít řešení, které by bylo lepší v jednom kritériu bez toho, aby bylo zhoršeno jiné kritérium [67]. Například, existuje-li řešení, v němž je určitý čas dojezdu a určitý počet přestupů, tak bude toto řešení nahrazeno novým pouze tehdy, bude-li nové řešení obsahovat méně přestupů a zároveň bude mít lepší čas dojezdu.

3. Profilový dotaz – ten je prováděn nad časovým intervalem doby odjezdu a pro každý čas v tomto intervalu se volá dotaz pro nejkratší příjezd. Hlavní myšlenkou je to, že pokud jsou nějaká dvě řešení stejně dobrá, vyhrává to, které má pozdější čas odjezdu.

4.2.2 RAPTOR

RAPTOR je akronym pro *Round-Based Public Transit Routing*. Tento algoritmus na rozdíl od předchozího nevyžaduje žádné předzpracování, takže se vše počítá najednou. Algoritmus pracuje v tzv. kolech [16]. Každé kolo představuje další přestup, který by musel být vykonán pro dosažení cílové zastávky. Takže nejprve se zkouší, zda je cílová zastávka dosažena po 0 přestupech (jestli existuje přímá cesta), poté po jednom přestupu, po dvou, atd.

Vstupem pro algoritmus je jízdní řád, počáteční zastávka, cílová zastávka a čas odjezdu [16]. Časy příjezdu na všech zastávkách jsou nastaveny na ∞, pouze u počáteční zastávky je nastaven na čas odjezdu. Při prvním průchodu ($k = 0$) se projdou všechny zastávky daného spoje a nastaví se jim hodnoty příjezdu na hodnoty, které odpovídají jízdnímu řádu. V následujících iteracích ($k \geq 1$) dochází k průchodu cest, které tyto zastávky protínají. Dochází však pouze k průchodu zastávek na dané cestě, které následují protínající zastávku, předchozí jsou ignorovány. Pokud se stane, že algoritmus znovu narazí na nějakou zastávku a zjistí, že se čas příjezdu zlepšil, tak je čas dojezdu upraven na tuto lepší hodnotu. Algoritmus končí v momentě, kdy nedojde k žádnému zlepšení času dojezdu [16]. Obrázek 4.3 názorněji ukazuje, jak celý algoritmus funguje.



Obrázek 4.3: Demonstrace algoritmu RAPTOR. Cílem je dostat se z p_s do p_t. V první iteraci dojde k průchodu cesty r_1. Ve druhé iteraci se algoritmus zaměří na cesty r_2 a r_3, neboť protínají cestu r_1 a je vyžadován pouze jeden přestup. Ve třetí iteraci dojde k analýze cesty r_4, kde se nachází cílová zastávka. V tomto případě lze využít jak cestu r_2, tak r_3. Vyhraje ta, pro níž má zastávka p_t nejnižší čas příjezdu. Algoritmus neprojde všechny zastávky na dané cestě, pouze ty, které jsou na nich vybarvené. Obrázek převzat z [16].

Základní varianta algoritmu RAPTOR se snaží minimalizovat pouze jedno kritérium, a to čas příjezdu. Byly vytvořeny i alternativy založené na základní verzi algoritmu, které pracují s dalšími kritérii. Takovým algoritmem je například McRAPTOR, kde Mc znamená *Multiple criteria*, který lze použít pro minimalizaci dalších kritérií jako je například cena přepravy [16]. Další alternativou je Range RAPTOR, kterému nemusí být zadán jeden čas

odjezdu, ale interval, ve kterém chceme odjet a algoritmus poté nalezne cestu s nejbližším časem příjezdu pro daný interval [16].

Kapitola 5

Analýza problému

Cílem této práce je vytvořit aplikaci, která nalezne nejlepší cestu z počátečního do cílového bodu pomocí kombinace automobilové a hromadné dopravy, která využije přestupních bodů k přechodu mezi těmito dvěma typy dopravy. Primárně je aplikace zaměřena na scénář, kdy je z počátečního do přestupního bodu použito auto a poté z přestupního bodu do cílového hromadná doprava. Aplikace však bude fungovat i pro případ, kdy uživatel někam jede a vrací se zpět. V tom případě aplikace vrátí trasy do cílové destinace i z ní.

V této kapitole nejprve dochází k analýze typů uživatelů a jejich potřeb, což popisuje sekce 5.1. V sekci 5.2 jsou popsány hlavně nevýhod existujících řešení z 2.4, který zabraňují tomu, aby byly využity pro tuto práci. Na základě této analýzy jsou v sekci 5.3 rozebrány funkční i nefunkční požadavky na aplikaci.

5.1 Analýza uživatelů

K analýze uživatelů a jejich zvyklostí byl použit dotazník, na který odpovědělo 33 respondentů ve věku od 20 do 27 let, kteří jsou převážně studenty. Dotazník se jich ptal na to, jak často cestují do práce, či do školy, jaké dopravní prostředky pro to využívají, případně jaké již existující dopravní plánovače používají. Kromě tohoto měli za úkol vyjádřit, jak je pro ně důležitá rychlost cesty, zpoždění, ekologická stopa a počet přestupů. Z dotazníku vyplynuly následující výsledky:

- 64 % respondentů cestuje několikrát týdně do práce nebo školy, ale pouze 18 % denně, což může být dáno tím, že mnoho respondentů jsou studenti, kteří nechodí do školy každý den (obrázek A.1).
- Většina respondentů (přes 80 %) využívá při cestování hromadnou dopravu a chůzi a pouze 10 % využívá pro cestu automobil (obrázek A.2).
- 58 % respondentů využívá při cestě více druhů dopravy (obrázek A.3).
- Z dotazníku také vyplynulo, že 76 % uživatelů používá aplikaci IDOS, 57 % Mapy Google a 33 % Mapy.cz pro plánování svých tras. Všichni tyto aplikace používají i na mobilních zařízeních (obrázek A.4).
- Drtivá většina uživatelů považuje rychlost cesty a zpoždění za velmi důležitá kritéria, počet přestupů je také důležitý, ale ne tolik jako předchozí 2 kritéria. Naopak ekologická stopa pro respondenty není ve většině případů vůbec důležitá (obrázek A.5).

- 61 % dotázaných uvedlo, že by chtěli, aby vznikla aplikace, která kombinuje automobilovou a hromadnou dopravu (obrázek A.6).

Na základě dotazníku a vlastní analýzy byly identifikovány 2 skupiny uživatelů, pro které aplikace může být užitečná, a sice dojíždějící a turisté.

5.1.1 Dojíždějící

Hlavní skupinou lidí, pro které je aplikace určená, jsou ti, kteří dojíždějí do Brna kvůli práci, případně i jiným záležitostem. Dostat se do Brna může být komplikované z důvodu velkého provozu ať už přímo v něm nebo na dálnici. Navíc kvůli vysokému provozu stráví uživatel nemálo času hledáním vhodného parkovacího místa, což pro něj přináší další diskomfort.

Tito lidé nutně nemusejí využívat pouze hromadnou dopravu pro přepravu z místa A do místa B, ale mohou pro část cesty využít své auto, poté ho zaparkovat blízko jednoho z přestupních bodů a odtud pokračovat hromadnou dopravou až do cílové destinace. Tím se sníží ekologická zátěž, neboť by se tímto snížila hustota automobilové dopravy a zlepšila se její plynulost. Rovněž se tím může snížit i čas cesty, neboť uživatel může bydlet na vesnici, kde spojů jezdí málo, a může se tak autem přiblížit do místa, kde spoje jezdí častěji. Důležitým požadavkem je nabídnout funkci, díky které může turista v rámci cesty tam naplánovat i cestu zpátky a tím i lépe naplánovat celý výlet.

Jelikož tato skupina uživatelů dojíždí pravidelně a potřebují být na daném místě včas, tak je pro ně důležité, aby aplikace uměla najít nejoptimálnější řešení z hlediska času dojezdu. Aplikace by také měla reagovat na to, že některé spoje mohou být často opožděné a nabízet tak trasy, které mají průměrné zpoždění nižší. Zároveň je důležité, aby aplikace byla jednoduchá a uživatel se v ní rychle zorientoval, případně aby mu aplikace ukázala, jak ji správně používat.

5.1.2 Turisté

Dalším typem uživatelů jsou turisté, kteří cílové město neznají, tudíž ho nemají zmapované a mohlo by pro ně být velmi komplikované najít parkovací místo a obecně se ve městě orientovat. Tito turisté poté mohou zaparkovat své auto někde dále od velkého města, kde není takový provoz a poté využít hromadnou dopravu k přepravě do turisticky zajímavých lokalit a poté opět využít hromadnou dopravu, aby se dostali k místu, kam zaparkovali své vozidlo.

Pro tento typ lidí, aby aplikace uměla nabízet i alternativní cesty, které sice mohou být delší, ale mohou být lepší z jiných hledisek, například zahrnují méně přestupů nebo jsou ekologičtější. Aplikace by měla rovněž všechny cesty, které připadají v úvahu, zobrazit na mapě včetně informací o odjezdech, příjezdech i přestupech.

5.2 Analýza existujících řešení

Všechna existující řešení z 2.4 jsou plnohodnotné dopravní plánovače, které nabízí mnoho užitečných funkcí, a používá je spoustu lidí. Avšak pro kombinaci auta a veřejné dopravy nejsou vhodná mimo jiné proto, že z existujících řešení pouze Mapy Google a Mapy.cz umí plánovat trasu pomocí auta.

Avšak ani tato řešení nejsou zcela vhodná, protože chce-li si uživatel v současnosti naplánovat trasu, ve které využije jak auto, tak hromadnou dopravu, musí tuto celkovou trasu rozdělit na dvě menší části, které musí naplánovat samostatně. Nejprve naplánuje

cestu autem z počátečního místa do místa, odkud chce pokračovat hromadnou dopravou, a odtud pak naplánovat trasu do cíle.

Tento přístup je značně nepohodlný, neboť uživatel musí trávit více času plánování dvou tras místo jedné. Pokud danou oblast nezná, tak přesně nemusí vědět, kde zaparkovat, a může tak vybrat místo, které nemusí být úplně vhodné.

Ať už aplikaci využijí dojíždějící nebo turisté, tak je velmi pravděpodobné, že pokud pojedou do cílové destinace, tak se budou vracet stejnou cestou zpět. U současných řešení se naplánování cesty tam a zpět musí udělat samostatně, a to i v případě, že uživatel zná dobu návratu. Nové řešení by tedy mělo umět vše naplánovat rovnou během plánování cesty tam.

5.3 Požadavky na aplikaci

Na základě analýzy uživatelů, jejich potřeb a analýzy existujících řešení lze sestavit následující funkční požadavky:

- Je nutné, aby aplikace umožňovala zobrazit mapu, neboť se na ní budou promítat vypočtené trasy a zároveň se kliknutím do mapy bude vybírat počáteční a koncový bod.
- Aplikace musí implementovat formulář, v němž bude možné specifikovat počáteční a koncový bod spolu s časem odjezdu.
- Dále se v ní musí nacházet možnost dodatečného nastavení parametrů, tedy umožnit výběr přestupního bodu ze seznamu, zvolit preferované prostředky hromadné dopravy a umožnit zaškrtnout možnost, která upřednostní trasy, které vyžadují nejméně přestupů.
- Aplikace musí implementovat funkci návratu zpět, díky níž si uživatel může v rámci plánování nastavit, kdy se chce vracet zpět a aplikace mu společně s cestami tam nabídne i cesty zpět.
- Aplikace musí umět zobrazit i alternativní trasy. Uživatel by neměl mít možnost vidět pouze trasu, která je nejrychlejší, ale také trasy, které jsou lepší z jiného hlediska, například obsahují nejmenší počet přestupů, případně mají nejlepší spotřebu.
- Aplikace musí umožnit nejen přepravu autem od začátku trasy k nejvhodnějšímu přestupnímu bodu, ale také možnost přepravit se autem z vhodné cílové zastávky do koncového bodu.

Zároveň z této analýzy plynou i nefunkční požadavky:

- Aplikace by měla být intuitivní a jednoduchá na používání, tedy uživatel, který aplikaci použije poprvé, by se měl schopen rychle zorientovat a v krátkém čase provést činnosti, které potřebuje. Zároveň pro uživatele, který pravidelně aplikaci používá, by nemělo zadání trasy a její nastavení příliš zdržovat.
- Aplikace by měla být přenositelná, ideálně fungovat v prostředí internetu, aby nebylo vyžadováno její stažení a instalace.
- Aplikace by rovněž neměla uživatele zdržovat. Měla by tedy být plynulá a nabídnout rychlý výpočet tras.

- Jelikož aplikaci mohou používat i lidé, kteří nemluví česky, tak by aplikace měla umožnit přepnutí do anglického jazyka.
- Z dotazníku vyplynulo, že všichni uživatelé používají konkurenční dopravní aplikace na mobilním zařízení, tudíž by tento požadavek by měla výsledná aplikace splňovat.

Kapitola 6

Návrh řešení

Tato kapitola se zabývá návrhem řešení, jehož hlavní náplní je plánování tras, které kombinují auto a městskou hromadnou dopravu tak, že nejprve uživatel cestuje autem z počátečního bodu do vypočteného přestupního bodu a odtud se pomocí hromadné dopravy dostane do cíle. Toto řešení může zjednodušit cestování hlavně uživatelům, kteří dojíždějí do práce nebo školy a rovněž i turistům. Na základě dotazníku bylo rozhodnuto, že řešení bude ve formě webové aplikace, protože všechna existující řešení nabízí připojení přes webové rozhraní.

Sekce 6.1 popisuje seznam kritérií, která rozhodují o tom, který přestupní bod jsou lepší než jiné. Sekce 6.2 poté ukazuje, jak vypadá architektura navrženého řešení a jak komponenty dané architektury mezi sebou komunikují. Návrh grafického uživatelského rozhraní se nachází v sekci 6.4 a sekce 6.5 shrnuje navrženou funkcionalitu, kterou je nutné implementovat.

6.1 Kritéria pro výpočet efektivní trasy

Jelikož se jedná o aplikaci, která plánuje trasy za použití automobilu a hromadné dopravy pro přepravu, je třeba definovat, jaká kritéria budou posuzována při výběru nejefektivnější trasy. Nestačí pouze nabízet trasy, které mají nejrychlejší dojezd, neboť by v mnoha případech vyhrál automobil. Ten bude efektivní, protože může vyjet kdykoliv a netýkají se ho žádná zpoždění a přestupy. Cílem je však také snížit dopady na životní prostředí, kde hromadná doprava vítězí, avšak rovněž není žádoucí nabízet trasy, které využívají pouze hromadnou dopravu.

Níže se nachází seznam kritérií, která se budou posuzovat při výběru nejlepší trasy a jejich alternativ:

- **Čas dojezdu** – byť jsou přestupní body vybírány podle více kritérií, stále se jedná o nejdůležitější kritérium, tudíž hlavním cílem aplikace bude nabízet nejkratší možné trasy z počátku do cíle.
- **Spotřeba** – jelikož je cílem aplikace více motivovat uživatele k používání hromadné dopravy, je jako kritérium použita i spotřeba, aby byly omezeny cesty, u kterých by z velké části bylo použito auto.
- **Zpoždění hromadné dopravy** – dalším z hodnotících kritérií je zpoždění spoje či spojů na dané trase. Cílem je omezit trasy, u kterých je zjištěno, že mají v daných hodinách zpoždění. To může prodloužit čas příjezdu do cílového bodu, ale rovněž

může uživatel kvůli zpoždění na jednom spoji ujet spoj, na který měl přestoupit, což celkovou dobu dojezdu ještě více prodlužuje.

- **Počet přestupů** – čím více přestupů musí uživatel na cestě podniknout, tím pro něj může být cesta samotná méně pohodlná a v případě zpoždění může navazující spoj propásnout. Cílem proto je vybírat řešení i s respektem k co nejméně přestupům.

Aby bylo možné přiřadit každému kritériu úroveň důležitosti a zároveň měnit jejich prioritu, bude každému z kritérií přiřazena určitá váha, jejichž velikosti se budou odvíjet od výsledku dotazníku. Kritéria, která jsou důležitá pro uživatele, budou mít váhu vyšší, zatímco méně důležitá nižší.

6.2 Architektura systému

Z analýzy uživatelů a celkových požadavků na řešení bylo rozhodnuto, že bude vytvořena webová aplikace. Schéma architektury se nachází na obrázku 6.1 a skládá se z následujících částí:

- **Klientská část aplikace** – jediná komponenta, se kterou uživatel interaguje. Obsahuje mapu pro zobrazení výsledků a pole pro zadání startovního bodu, cílového bodu, času odjezdu a možnosti dodatečného nastavení. Tyto informace poté putují na server pomocí REST API a jako odpověď dostane nejvhodnější trasy, které poté zobrazí na mapě.

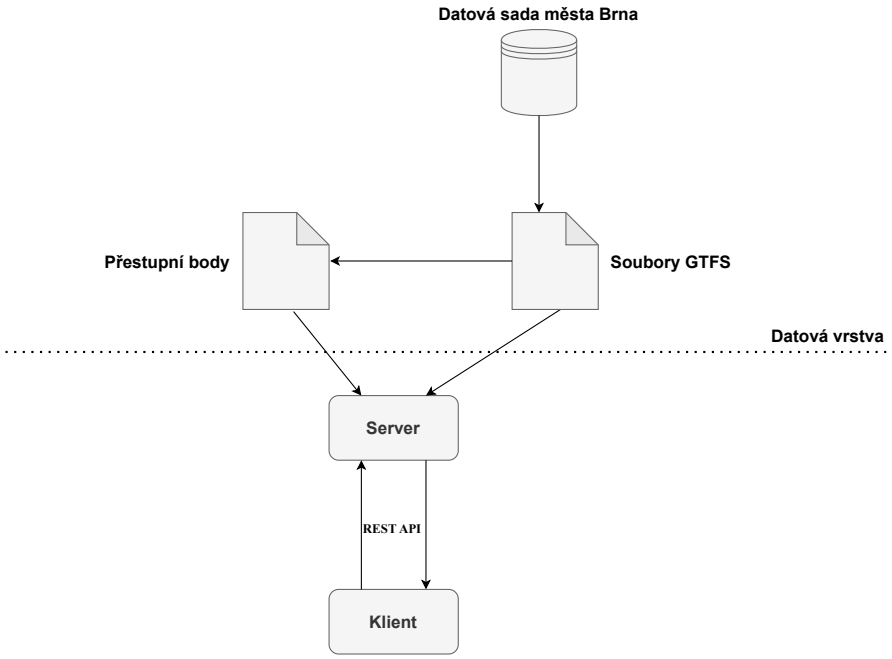
- **Serverová část aplikace** – tato část obsahuje dva koncové body, kdy jeden z nich vrací seznam přestupních bodů a druhý přijímá požadavky od klientské části obsahující veškeré informace, na základě nichž dojde k výpočtu optimálních tras a jejich odeslání klientu. Server periodicky získává přestupní body z datové vrstvy.

- **Datová část** – obsahuje seznam přestupních bodů, které budou poskytovány serveru, který je bude používat při výpočtech trasy a zároveň je posílat klientovi. Tato data budou periodicky aktualizována, ovšem v delších intervalech, protože se nepředpokládají změny v zastávkách. Přestupní stanice vznikají ze souborů GTFS, které se budou každý týden aktualizovat, neboť v této periodě je aktualizují i stránky otevřených dat pro Brno, ze kterých se také tyto soubory budou stahovat.

6.3 Přestupní body

Jelikož aplikace bude umožňovat kombinaci automobilu a hromadné dopravy, musí existovat bod, respektive body, kde dojde k přechodu z auta na hromadnou dopravu. Teoreticky je těchto bodů nekonečně mnoho, ale bylo by časově nereálné určit, který bod je ten správný. Proto bylo rozhodnuto, že jako validní přestupní body bude vnímána pouze určitá podmnožina. Tato podmnožina odpovídá všem zastávkám daného regionu, protože je od nich velmi jednoduché přestoupit na hromadnou dopravu.

Problémem však je vysoké množství zastávek, kterých je jen v Jihomoravském kraji více než 3000. Toto číslo však lze snížit, protože podmínkou pro to, aby byla zastávka klasifikována jako přestupní je, aby bylo možné v její blízkosti zaparkovat auto. Většinou, ale ne vždy toto splňují autobusová a vlaková nádraží.



Obrázek 6.1: Navržená architektura systému.

Aplikace bude z těchto přestupních bodů vybírat ty, které budou z hlediska kritérií ze sekce 6.1 nejvhodnější. Uživatel si však bude moct sám zvolit, který z přestupních bodů se použije.

6.4 Návrh uživatelského rozhraní

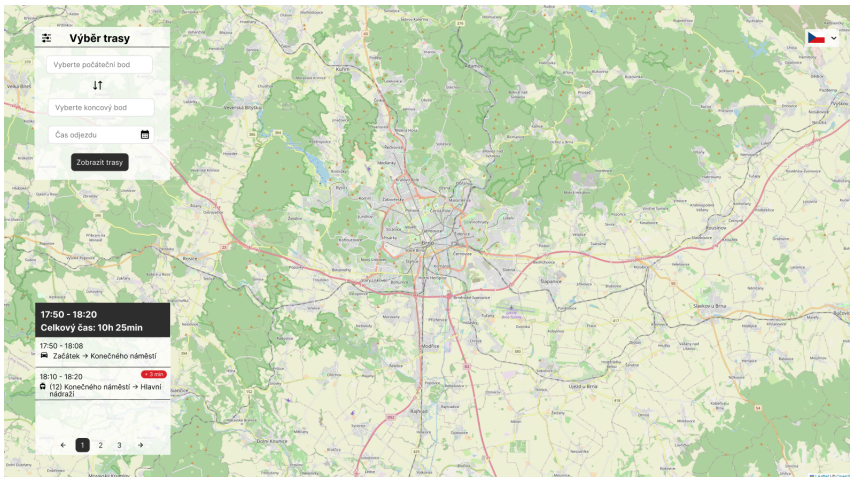
Jelikož podle dotazníku více než 50 % dotázaných uvedlo, že k plánování tras používají aplikaci Mapy Google, která v sobě obsahuje mapový podklad a nad tímto podkladem se poté nacházejí prvky, které umožňují aplikaci ovládat, například formulář pro plánování trasy. Tímto způsobem je navrženo i uživatelské rozhraní této aplikace.

6.4.1 Hlavní obrazovka

Na hlavní obrazovce se nachází referenční mapa Jihomoravského kraje, po které je možné se volně pohybovat, případně ji přibližovat a oddalovat. Po levé straně se nachází formulář, který je důležitý pro zadání parametrů nezbytných pro výpočet trasy. Formulář obsahuje pole pro volbu počátečního a koncového bodu společně s tlačítkem, které umožní prohození těchto dvou hodnot. Dále se zde nachází také pole pro specifikaci data a času odjezdu a v neposlední řadě tlačítko, které odešle požadavek na server.

Nalevo dole se poté nachází část, ve které se zobrazí informace o nalezených trasách. Levá část obsahuje sumarizační údaje pro danou trasu jako je celkový čas cesty, od kdy do kdy bude trvat a celkovou vzdálenost. Napravo od zvoleného řešení se nacházejí jednotlivé spoje tak, jak jdou po sobě s informací o tom, o jaký typ dopravy se jedná, čas předpokládaného nástupu a výstupu, názvy zastávek, číslo spoje a případné zpoždění, pokud ho daný spoj

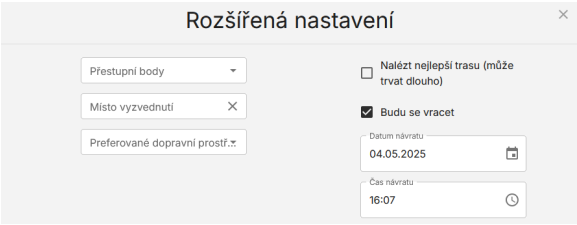
v danou dobu obvykle má. Na mapě se rovněž nachází trajektorie zvoleného řešení. Na obrázku 6.2 se nachází prvotní návrh hlavní obrazovky.



Obrázek 6.2: Prvotní návrh hlavní obrazovky.

6.4.2 Nastavení

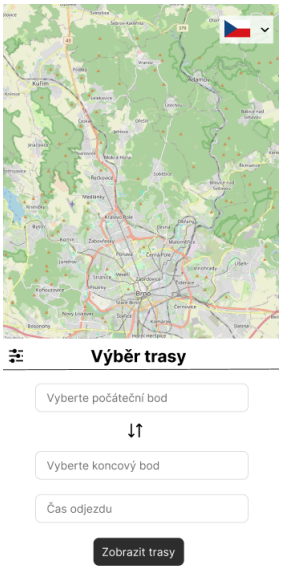
Pro dodatečné nastavení je použito separátní dialogové okno a nenachází se na hlavní obrazovce proto, aby nebyla omezena viditelnost mapy. V nastavení na obrázku 6.3 lze najít pole pro přesnou specifikaci přestupního bodu, který má být při výpočtu použit. Dále si zde může uživatel vybrat, které prostředky hromadné dopravy chce použít během přepravy a také specifikovat bod vyzvednutí. Poté se zde nachází možnost vynucení nalezení nejlepší trasy, která vyzkouší všechny přestupní body. Poslední položka umožňuje nastavení data a času návratu.



Obrázek 6.3: Ukázka dodatečného nastavení parametrů.

6.4.3 Mobilní zobrazení

V dnešní době však uživatelé používají nejen počítač, ale i chytré telefony, takže se rozhraní přizpůsobí, jestliže uživatel použije mobilní zařízení. Zde dochází k přesunu komponenty formuláře a informaci o trasách do dolní části obrazovky tak, aby uživatel nemusel mobilní zařízení přehmatávat. Po odeslání formuláře se místo něj zobrazí vypočtené trasy a po kliknutí na jednu z nich je obsah nahrazen detailem dané trasy. Z detailu na seznam tras a následně na formulář se lze vrátit pomocí šipky, která se nachází vlevo těsně nad zobrazovaným obsahem. Obrázek 6.4 ukazuje prvotní návrh mobilního zobrazení pro aplikaci. Skutečný vzhled mobilního zobrazení se poté nachází na obrázku 7.5.



Obrázek 6.4: Prvotní návrh mobilního zobrazení pro vstupní formulář.

6.5 Navržená funkcionalita

Na základě analýzy uživatelů, existujících řešení a požadavků na aplikaci byla navrženy následující funkce:

1. **Získání trasy kombinující auto a veřejnou dopravu** – hlavním výstupem aplikace musí být seznam různých řešení, které vedou k cíli. Každé řešení musí obsahovat detaily o jednotlivých spojích a také umět zobrazit trasu daného řešení na mapě.
2. **Výběr přestupního bodu** – přestupní bod může být vybrán automaticky podle toho, který nabídne lepší řešení. Pokud však uživatel nějaký bod preferuje, může si ze seznamu přestupních bodů zvolit ten, který preferuje.
3. **Zobrazení přestupních bodů** – kromě seznamu přestupních bodů budou všechny vidět i na mapě. Uživatel si bude moct po kliknutí na něj zobrazit jeho název a rovnou ho označit jako preferovaný přestupní bod.
4. **Zobrazení zpoždění spojů** – v rámci každého spoje musí bude vidět jeho průměrné zpoždění za několik předchozích dní. Kromě průměrného zpoždění bude rovněž možnost zobrazit zpoždění pro každý předcházející den samostatně.
5. **Funkce návratu zpět** – uživatel si bude moct během plánování cesty tam rovněž naplánovat i cestu zpět a server společně s cestami do cílové destinace vrátí i cesty do počátečního bodu.
6. **Funkce vyzvednutí** – uživatel si v nastavení bude rovněž moct zvolit místo vyzvednutí, kde ho někdo jiný vyzvedne a odtud budou společně cestovat do cílové destinace.

Pro zaručení správné funkcionality však musí být vyřešeny problémy, které by tuto funkcionalitu mohly omezovat. Tyto problémy jsou následující:

1. **Získání seznamu přestupních bodů** – z GTFS souborů je nutné získat podmnožinu přestupních bodů, které by byly pro uživatele při plánování k dispozici.

2. **Získání vhodné posloupnosti spojů** – ze zadaného počátečního a koncového bodu a datu a času odjezdu je potřeba získat posloupnost spojů, kterými je možné se spolehlivě dostat ze startu do cíle.

3. **Výběr vhodných přestupních bodů pro výpočet** – je potřeba implementovat algoritmus, který z dané množiny přestupních bodů vybere takové, které jsou nejvhodnější pro výpočet trasy.

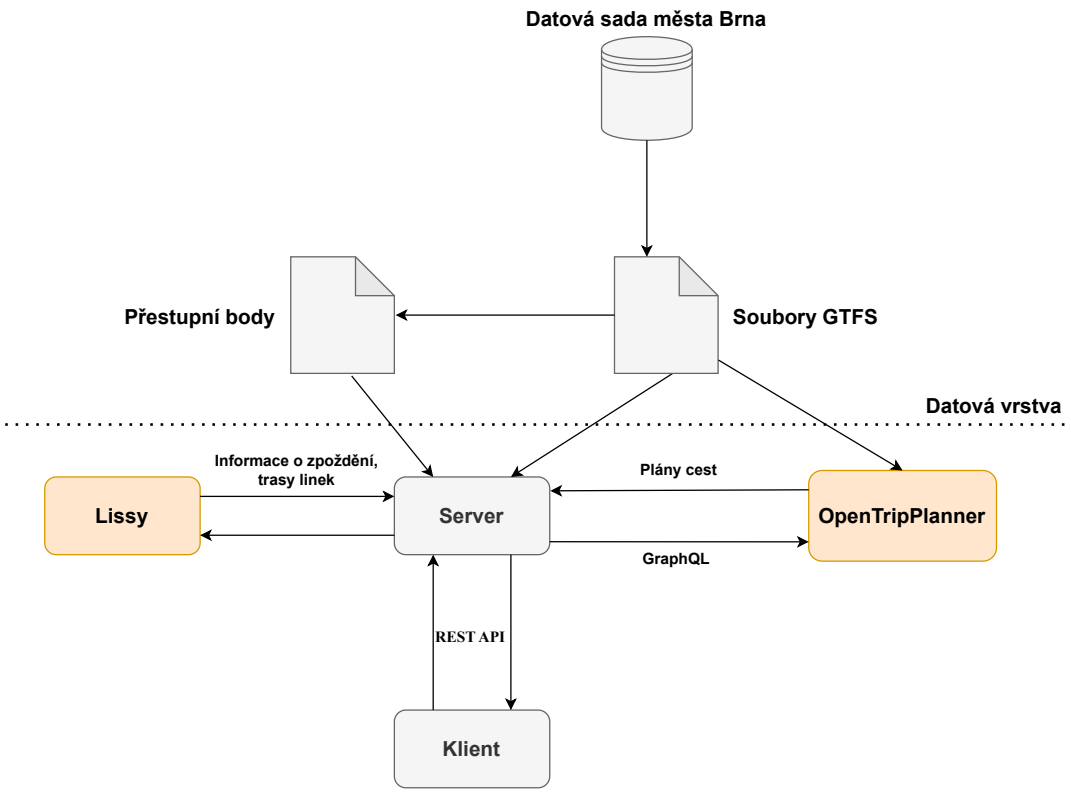
4. **Ohodnocení tras** – získané trasy je potřeba vhodně ohodnotit tak, aby byly uživatelům nabízeny co nejlepší trasy s ohledem na kritéria ze sekce 6.1.

Vzhledem k tomu, že přestupní body jsou jediná data, se kterými aplikace pracuje a která jsou pouze pro čtení, tak nebudou uchována v plnohodnotné databázi, ale v souborech. Dále serverová část aplikace bude obsahovat dva koncové body, z nichž jeden bude sloužit pro předávání přestupních bodů klientovi a na druhý z nich bude klient posílat požadavky na trasy a server je bude skrze tento bod vracet klientovi.

Kapitola 7

Implementace

Tato kapitola se zabývá implementací navržené funkcionality. Jsou zde popsány všechny komponenty a použité externí aplikace z obrázku 7.1, který rozšiřuje obrázek 6.1 o další části použité při implementaci. Dochází zde rovněž k popisu algoritmů, které byly použity pro hledání vhodných přestupních bodů a ohodnocení řešení.



Obrázek 7.1: Implementační architektura aplikace.

7.1 Datový model

Pro aplikaci není použita žádná databáze, neboť dat a vztahů mezi nimi není tolik, aby se vyplatilo použít plnohodnotný databázový systém, proto jsou veškerá data obsažena v souborech ve formátu `.csv`, kde jsou jednotlivé sloupce od sebe odděleny středníkem. Tyto soubory se používají jednak jako zdroj dat pro aplikaci, ale také pro komunikaci mezi serverem a pomocnými skripty.

Prvním ze zdrojů dat jsou soubory GTFS, které sice obsahují textové soubory, avšak ty svým charakterem `.csv` soubory připomínají. GTFS slouží jako primární zdroj dat pro získání přestupních bodů a také se předávají aplikaci `OpenTripPlanner` popsané v sekci 7.2.1, která je používá pro výpočty vhodných tras.

Dalším souborem je `transferStopsWithParkingLots.csv`, který již obsahuje všechny relevantní přestupní body získané z GTFS souborů doplněné o informaci, zdali se v okolí daného přestupního bodu nachází parkoviště, či nikoliv. Příklad obsahu souboru se nachází ve výpisu 7.1. Při spuštění aplikace je tento soubor zpracován serverem a data z něj posílá klientské aplikaci, která je poté zobrazí na mapě.

```
Babice nad Svitavou;49.276645;16.673012;U22101Z2;1
Blansko město;49.360996;16.639464;U22302Z2;1
Blansko, aut. st.;49.35381;16.646735;U12342Z10;1
Blatnice pod Sv. Antonínkem;48.941163;17.437706;U29403Z10;0
```

Výpis 7.1: Obsah souboru `transferStopsWithParkingLots.csv`. Každý přestupní bod obsahuje jméno, polohu, identifikátor a jestli se v okolí nachází parkoviště (1 – ano, 0 – ne).

Další soubory, které se již ale používají pro komunikaci mezi serverem a pomocným skriptem, jsou `candidates.csv` a `candidatesClusters.csv`, které se používají pro předání všech kandidátních přestupních bodů shlukovacímu skriptu (`candidates.csv`[1]) a seznam přestupních bodů, které byly vybrány jako středy jednotlivých shluků (`candidatesClusters.csv`). Sekce 7.2 popisuje detailněji, proč se v aplikaci shluky používají.

7.1.1 Získávání dat

Data se získávají pomocí skriptu jednou denně proto, aby existovala synchronizace s externími aplikacemi, které se také aktualizují jednou denně. Tento skript poté stáhne GTFS soubory a pomocí knihovny `pandas`[2] je zpracuje.

Přestupní body se vybírají ze všech vlakových nádraží a stanic, které lze snadno získat díky souboru `routes.txt` z GTFS. Součástí přestupních bodů jsou rovněž i větší autobusová nádraží, která z GTFS nelze jednoznačně získat. Proto je využito toho, že název autobusového nádraží v sobě typicky obsahuje podřetězce *„nádr.“*, *„nádraží“*, nebo *„aut. st.“*, tudíž se vybírají takové zastávky, které jej obsahují i s rizikem, že některá nádraží nemusela být nalezena.

Dalším krokem je jednotlivým přestupním bodům přiřadit informaci o tom, jestli v jejich okolí existuje nějaké parkoviště, na kterém se dá zaparkovat. K tomuto je použita

[1] Struktura souboru `candidates.csv` je až na drobné detaily totožná jako u `transferStopsWithParkingLots.csv`.

[2] https://pandas.pydata.org/

služba `overpass turbo`[3], která pomocí speciálního jazyka dokáže filtrovat OpenStreetMap data a dokáže vrátit například hranice různých oblastí nebo všechny výskyty místa daného typu v okolí. Zde je použit pro nalezení všech parkovišť, která se od daného přestupního bodu nacházejí do vzdálenosti 500 metrů. Jsou brána v úvahu veřejná parkoviště, placená i neplacená.

Jedná se o nejkomplikovanější část zpracování, protože se musí zpracovat okolo 200 přestupních bodů. Zároveň tuto službu používají lidé z celého světa a často se stane, že server nevrátí odpověď kvůli vysokému zatížení. Řešením by bylo mít vlastní instanci `overpass turbo`, na kterou by byly prováděny dotazy, což by ale vyžadovalo nemalé místo na disku, kam se veškeré mapy uloží. Současné řešení proto funguje tak, že přestupním bodům, pro které dotaz nebyl vykonán, je přiřazena stejná hodnota jako zastávkám, které v okolí žádná parkoviště nemají.

7.2 Server

Pro implementaci serverové části byl použito běhové `Deno`[4], což je rychlejší alternativa oproti populárnějšímu `NodeJS`. Pro zpracování požadavků byla použita knihovna `Hono`[5], která implementuje odlehčený HTTP server. K dispozici jsou 2 koncové body, a sice:

- `/api/transferStops`, který uživateli vrací všechny přestupní body,
- `/api/calculateTrips`, který přijímá počáteční a koncový bod, datum a čas a uživatelské preference a vrací trasy, které vedou ze startu do cíle a trasy, které vedou zpět (pokud si uživatel tuto možnost zvolí).

7.2.1 Výpočet trasy

Hlavním cílem aplikace je nalézt nejvhodnější trasy, které se ovšem musí nějak spočítat a rovněž musí respektovat jízdní řád, a tedy spoje by v dané trase na sebe měly navazovat. Dalším požadavkem je to, aby výpočet netrval příliš dlouho. Ke hledání samotných spojů a tras je proto použita externí aplikace, a sice `OpenTripPlanner 2`.

OpenTripPlanner 2

OpenTripPlanner 2 (v práci bude nadále použita zkratka OTP) je aplikace s otevřeným zdrojovým kódem, která umožňuje plánování cest zahrnující auto, jízdní kolo, chůzi i hromadnou dopravu.

Aplikace je napsána v jazyce Java a na jejím řešení staví plánovače v Norsku, Švédsku, některých státech USA a v některých evropských městech jako třeba Lipsko, Poznaň nebo Grenoble [54]. Je to způsobeno tím, že aplikace není vytvořena pro určitý region a je tedy nutné jí dodat jízdní řády ve formátu GTFS případně ve formátu NeTex. Pro správné zobrazení trasy je také třeba do aplikace nahrát část mapy ve formátu `.pbf`, pro kterou má plánování fungovat. Aplikace je primárně určená pro serverovou část, na kterou se klienti dotazují. Nicméně je v rámci ní dodáváno i jednoduché klientské rozhraní, které demonstruje, co aplikace umí a jak s ní pracovat. Nutno podotknout, že aplikace samotná toho umí mnohem více, než co nabízí klientská část.

[3] https://overpass-turbo.eu/
[4] https://deno.com/
[5] https://hono.dev/

Komunikace s OTP

Pro komunikaci s OTP není použito REST API, jako tomu bývá u běžných aplikací, ale GraphQL. Jedná se o dotazovací jazyk a komponentu běžící na straně serveru. Zde musí být k dispozici tzv. schéma, které se skládá z typů, které v sobě obsahují jednotlivé položky [25]. Výpis 7.2 ukazuje, jakým způsobem lze definovat schéma pro dotaz, postavu a planetu. Výpis 7.3 poté ukazuje, jak lze pro toto schéma vytvořit dotaz.

```
type Query {
    hero: Character
}

type Character {
    name: String,
    friends: [Character],
    homeWorld: Planet
}

type Planet {
    name: String,
    climate: String
}.
```

Výpis 7.2: Schéma pro GraphQL. Je možné použít jak předdefinované, tak i uživatelsky definované typy.

```
{
    hero {
        name
        friends {
            name
            homeWorld {
                name
                climate
            }
        }
    }
}.
```

Výpis 7.3: Dotaz v GraphQL. Tento vrátí jméno hrdiny a pro každého jeho přítele vrátí jméno spolu s názvem a klimatem jeho domovské planety. Převzato z [25].

Díky této struktuře lze v dotazu libovolně měnit pole podle toho, která potřebujeme. V REST API může existovat mnoho koncových bodů (tzv. *endpoints*), které vždy vrací data v pevně dané struktuře. Často se ale stane, že *endpoint* vrátí více dat, než je v danou chvíli potřeba, a někdy na druhou stranu je pro daný účel nutné kombinovat data z různých koncových bodů.

Výhodou GraphQL je, že zde existuje pouze jeden koncový bod, na nějž probíhají dotazy. Rovněž lze v dotazu specifikovat, které položky potřebujeme, ať už z jednoho, nebo i více

datových zdrojů. Nevýhodou je, že některé jednodušší dotazy mohou trvat déle než při REST API [48] a rovněž delší dotazy mohou být nečitelné.

Dotazy na trasy

Nejdůležitějším dotazem v OTP pro tuto aplikaci je dotaz `trip`, který na základě požadavků na trasu vrátí různé varianty přepravy s místa A do místa B. Parametrů, které lze do dotazu zadat, je více, ale tato práce využívá následující:

- `from` – souřadnice počátečního bodu,
- `to` – souřadnice koncového bodu,
- `dateTime` – datum a čas ve formátu ISO 8601,
- `modes` – obsahuje parametry:
  - `accessMode` – způsob dopravy z počátečního bodu na nejbližší zastávku,
  - `egressMode` – způsob dopravy z nejbližší zastávky do koncového bodu,
  - `directMode` – způsob dopravy mezi počátečním a koncovým bodem, pokud je použita přímá cesta,
  - `transportMode` – způsob přepravy v rámci jednotlivých spojů, lze zvolit i více dopravních prostředků.
- `numTripPatterns` – maximální počet řešení, které má OTP nalézt. Pro auto stačí jediné řešení, protože je nalezena jediná nejkratší cesta, avšak u přepravy hromadnou dopravou může existovat více možností, jak se do místa dostat. Výchozí hodnota je 12, která je také v aplikaci použita.
- `pageCursor` – funguje na principu stránkování výsledků. Pokud se stane, že není nalezena trasa pro určitý požadavek, tak se lze znovu dotázat se stejnými parametry a přidaným kurzorem, který podle typu kurozru vrátí předchozí, nebo následující výsledky. Reference na předcházející a následující stránku se vrací s každým dotazem na trasu bez ohledu na to, jestli je výsledek prázdný, či nikoliv.

Tento dotaz je použit ve dvou variantách, kdy v první je využito pouze auto, které slouží k přepravě z počátečního bodu do přestupního bodu. Druhá varianta je použita při přepravě z přestupního bodu do cíle, kdy je využita hromadná doprava. Pro nalezení trasy pro auto je použit algoritmus A*, neboť nejsou vyžadovány informace o jízdních řádech, a pro hromadnou dopravu OTP používá algoritmus RAPTOR [54].

Rychlost výpočtu je závislá na použitém počítači a na trase, ale obecně je výpočet pro auto rychlejší než pro MHD, kdy na procesoru `AMD Ryzen 7` byla rychlost pro auto řádově v nižších desítkách milisekund, zatímco pro hromadnou dopravu se rychlost pohybovala okolo nižších stovek milisekund. Detailnější analýzu výkonnosti lze najít v sekci 8.2. Vliv na rychlost má dle [54] i použití GTFS souboru `transfers.txt`, který definuje časy přestupů mezi některými vybranými spoji. Není doporučováno tuto funkci používat, neboť parametry rychlosti chůze a přestupů se dají nastavit v konfiguračních souborech pro OTP. Zároveň nepoužití této funkce o něco zlepší výkonnost.

Výstupem tohoto dotazu je pole všech řešení (tzv. *patterns*). Kde každé toto řešení vrací základní sumarizační údaje o trase jako je:

- čas odjezdu,
- čas příjezdu,
- celkový čas cesty,
- celková vzdálenost.

Každé toto řešení v sobě obsahuje pole spojů (tzv. *legs*), každý spoj v sobě obsahuje stejné sumarizační informace pro daný spoj spolu s informacemi o nástupní zastávce, výstupní zastávce, číslo linky a trasu samotnou zakódovanou do polyline. Kvůli informacím o zpoždění je také ještě potřeba vracet posloupnost všech zastávek daného spoje (ne jen těch, které jsou součástí řešení) společně s časy odjezdu.

Dotazy na informace o GTFS souborech

Jelikož jsou GTFS soubory důležitým prvkem pro fungování OTP, tak je nabízeno i rozhraní, pomocí kterého se lze dotazovat na informace o GTFS. Dokáže tedy vrátit například všechny linky, všechny zastávky v určité vzdálenosti od daného bodu, případně informace o provozovatelích. V této práci je toto API využito pro získání rodičovských zastávek pro přestupní body.

Analýza OTP

OpenTripPlanner 2 je velmi vhodný pro použití v této aplikaci, protože dokáže díky GTFS spolehlivě nalézat různé trasy, nabízí mnoho různých nastavení, které ani nejsou v této práci zmíněny, a je možné jej použít v podstatě pro jakýkoliv region, který nabízí jízdní řády ve formátu, který OTP podporuje. Díky tomu, že se jedná o aplikaci s otevřeným zdrojovým kódem, tak je možná ji i zdarma používat.

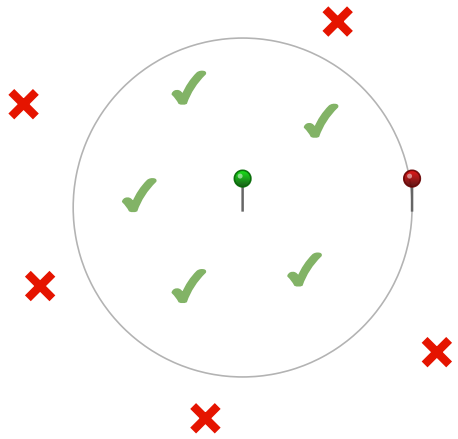
Nevýhodou OTP je vysoká paměťová náročnost, protože kvůli tomu, aby výpočty byly co nejrychlejší, se informace o GTFS a podkladové mapě ukládají přímo v RAM, takže dle [54] se doporučuje vyhradit pro tuto službu minimálně 2 GB této paměti. Pro Jihomoravský kraj se nejedná o takový problém, protože GTFS soubory nejsou rozsáhlé. Problém může být u jiných regionů, které mají GTFS soubory větší a zároveň mají větší podkladovou mapu.

7.2.2 Nalezení vhodného přestupního bodu

Hlavním problémem, který je třeba vyřešit pro nalezení nejvhodnějších tras, je nalezení vhodného přestupního bodu. Jako první možnost se nabízí vyzkoušet všechny body a zjistit, které z nich jsou nejlepší. Tento přístup lze použít pouze tehdy, jsou-li dotazy prováděny nad malým počtem bodů . Obecně však platí, že čím více bodů se vyzkouší, tím déle bude celý výpočet trvat.

Nejprve vždy dojde k odstranění přestupních bodů, které jsou příliš daleko a to tak, že zůstanou pouze takové, které jsou od počátečního bodu blíže než vzdálenost od počátečního bodu ke koncovému vzdušnou čarou. Jelikož jsou souřadnice bodů ve stupních, tak je pro výpočet vzdálenosti potřeba použít tzv. Haversinův vzorec pro výpočet vzdálenosti dvou bodů na kouli [12]. Obrázek 7.2 názorněji popisuje, které zastávky jsou zachovány a které odstraněny.

Může se stát, že se v zadané vzdálenosti nenachází žádný přestupní bod, což může být způsobeno krátkou vzdáleností mezi startem a cílem. V takovém případě se plánovač



Obrázek 7.2: Nejprve dojde k odstranění zastávek, které se nachází dále od počátečního bodu než je přímá vzdálenost mezi počátečním a koncovým bodem.

chová tak, že vrátí trasy mezi zadanými body, které využívají pouze hromadnou dopravu a nedochází k použití auta. Jestliže uživatel sám specifikuje přestupní bod, je trasa počítána pouze pro něj a nedochází k žádnému prořezávání.

Jestliže se nějaké přestupní body nacházejí v dané vzdálenosti od startu, tak se zkontroluje jejich počet. Experimentálně bylo určeno, že pokud je těchto bodů nejvýše 15, lze pro nalezení nejlepší trasy vyzkoušet všechny. Pokud jich je více, je pro určení kandidátních přestupních bodů použito shlukování. Když si uživatel zvolí, že chce provést výpočet pro všechny tyto body, tak se zkouší všechny body, které vyhovují vzdálenosti od startu, což ale znatelně zvyšuje čas výpočtu.

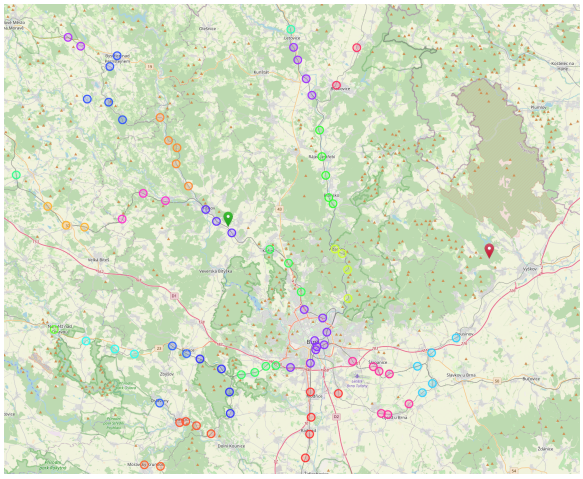
Shlukování přestupních bodů

Jestliže je přestupních bodů příliš mnoho, je potřeba nějakým způsobem zvolit takové, které se použijí pro výpočet. Podmínkou je, aby byly zvoleny takové body, které mohou zajistit různá řešení. Není tedy například žádoucí, aby byl vybrán bod A a bod B, který na bod A bezprostředně navazuje, protože se například nacházejí na stejné lince.

Shlukování probíhá pomocí skriptu napsaném v jazyce `Python`[6] za použití knihovny `scikit-learn`[7]. Tato knihovna byla použita kvůli jednoduchosti použití a také jejímu renomé. Jako shlukovací algoritmus byl zvolen algoritmus `KMeans`, který tvoří shluky kulovitého tvaru, které jsou poměrně dobře odděleny, a zároveň jsou rychlé na výpočet [7].

Základní počet shluků byl stanoven na 15 a jako alternativní počet shluků bylo nastaveno číslo $15 + \sqrt{n}$, kde n je počet přestupních bodů, které do skriptu vstupují. Shluky jsou poté ohodnoceny pomocí Davies-Bouldinova indexu, který hodnotí, jak jsou dané shluky dobré [7]. Podle této hodnoty je vybrán nejlepší počet shluků. Jelikož `KMeans` pracuje s fiktivními středy, tak je pro každý shluk vrácen přestupní bod, který se od daného středu nachází nejblíže. Příklad rozdělení přestupních bodů do shluků lze nalézt na obrázku 7.3.

[6] https://www.python.org/
[7] https://scikit-learn.org/stable/



Obrázek 7.3: Ukázka shlukování zastávek, které jsou uvažovány, je-li zadán počáteční a koncový bod. Přestupní body, které mají stejnou barvu, patří do jednoho shluku. Tyto body jsou pro demonstrační účely zvětšeny.

7.2.3 Získávání cest a informací o zpoždění

Jakmile je jasné, které přestupní body se budou používat, tak se může spustit plánování. To funguje tak, že se pro každý přestupní bod vypočte trasa autem ze startu do něj a posléze se vypočtou trasy využívající hromadnou dopravu, které vedou z přestupního bodu do cíle, a dojde ke spojení těchto tras.

Co se týče verze s automobilem, tak zde získání cesty zajišťuje OTP, které se řídí pouze podkladovou mapou, takže jsou-li získány informace o dopravě autem, tak je získána i jeho přesná trajektorie. U hromadné dopravy je toto také možné za podmínky, že se v GTFS nachází soubor `shapes.txt`, který definuje trasy pro jednotlivé spoje. GTFS pro Pražskou integrovanou dopravu tento soubor obsahuje, ale Jihomoravský kraj jím nedisponuje. Jestliže OTP nemůže tento soubor nalézt, spojuje navazující zastávky přímkou, což není vhodné pro vizualizaci zvláště tehdy, pokud jsou zastávky daleko od sebe. K získání cest a zpoždění lze pro Jihomoravský kraj využít aplikaci `Lissy`.

Lissy

Tato aplikace je projektem Juraje Lazúra, který vylepšil svou diplomovou práci zabývající se vizualizací zpoždění a cest pro hromadnou dopravu v Jihomoravském kraji [41]. Aplikace shromažďuje údaje o zpoždění a dokáže je pro jednotlivé dny zobrazit. Nejedná se však o získávání zpoždění v reálném čase. Postup k získání cest a zpoždění je následující:

1. Získání kalendářních dat, pro která jsou údaje k dispozici – zde dochází k získání rozsahu dat, pro který jsou údaje k dispozici, kdy nejpozdější datum je den před provedením dotazu, protože ke zpracování údajů o zpoždění za předchozí den dochází až v den následující. Součástí odpovědi jsou i data z tohoto rozsahu, pro která údaje nejsou dostupná, což může být způsobeno chybou na straně `Lissy`.

2. Získání linek pro daný rozsah dat – `Lissy` pro zadaný rozsah dat vrátí seznam linek, které obsahují informace o zpoždění. Jsou vráceny pouze takové linky, které tyto údaje obsahuje ve všech datech z daného rozsahu. Problém je to, že ne vždy jsou informace

o lince pro daný den dostupné, takže se stává, že čím více do historie se jde, tím méně linek aplikace vrátí.

3. Získání spojů pro danou linku – `Lissy` pro zadanou linku a zadaný rozsah dat vrátí informace o každém spoji, který obsahuje, a sice čas odjezdu a identifikátor cesty. Na základě tohoto identifikátoru pak lze dalším dotazem získat cestu pro daný spoj. Identifikátor cesty není jeden pro celou linku, protože každý spoj v ní obsažený může jet jinudy, což závisí například na denní době.

Bylo by neefektivní tyto informace získávat pokaždé, co by někdo chtěl vypočítat trasu, proto se získávají ihned po spuštění serveru a pak se až na požádání získávají údaje o cestách a zpožděních pro jednotlivé spoje.

Pro získání cest je použit identifikátor u daného spoje a `Lissy` vrátí cestu celého spoje v podobě pole, kde každá položka obsahuje posloupnost souřadnic. Platí, že první položka pole je použita k vytvoření cesty mezi zastávkou 1 a 2, druhá položka mezi zastávkou 2 a 3, a tak dále [42].

Jenže během trasy se velmi často stane, že je využita pouze část spoje (například pouze 3 zastávky), takže je potřeba z celé této cesty získat podcestu, která začíná v nástupní stanici a končí ve výstupní.

Pro získání zpoždění je nejprve potřeba zvolit správný spoj na základě času odjezdu z první zastávky. Identifikátor, který je součástí tohoto spoje, společně s rozsahem dat, pro které zpoždění požadujeme, je poté pro jejich získání.

Zpoždění jsou poté vrácena jako jeden velký objekt, kde jsou pro každé datum z rozsahu tato zpoždění zvlášť. Stejně jako v případě získání cest je první klíč použit pro zpoždění mezi zastávkou 1 a 2, druhý pro zastávku 2 a 3, atd. Každý segment mezi zastávkami je poté rozdělen na další segmenty, mezi nimiž se zpoždění měří [42]. Detailnější popis se nachází ve výpisu 7.4.

```
{
    "2025-04-08": {
        "0": {
            "1": 0,
            "3": 1,
        },
        "1": {
            "3": 2,
            "10": 3
        }
    },
    "2025-04-09": {
        "0": {
            "3": 0,
            "15": 0,
        },
        "1": {
            "1": 1,
            "7": 1
        }
```

```
        }
    }
```

Výpis 7.4: Příklad zpoždění pro 8. 4. 2025 a 9. 4. 2025. Pro 8. 4. existuje záznam pro segment mezi zastávkou 1 a 2 (první mezizastávkový segment, klíčem je index pole) a mezi zastávkou 2 a 3. Pro první mezizastávkový segment je zpoždění mezi subsegmentem 2 a 3 nula minut a mezi subsegmentem 4 a 5 jedna minuta. Stejně to funguje pro 9. 4.

Na rozdíl od cest není třeba v tomto případě provádět výřez, který odpovídá nástupní a výstupní zastávce, protože rozhodující je pouze zpoždění na výstupní zastávce, neboť to rozhoduje o času příjezdu a o tom, jestli cestující stihne navazující spoj, pokud existuje.

Nepříjemnou vlastností, se kterou se lze v při získávání zpoždění setkat, je fakt, že občas mohou chybět některé subsegmenty a dokonce i celé segmenty mezi zastávkami. V prvním případě se lze spokojit se subsegmentem, který má maximální klíč (nachází se nejblíže výstupní zastávce). Druhý případ je řešen tak, že se získá zpoždění z nejbližší zastávky, která informace o zpoždění obsahuje a zároveň předchází zastávce, pro je třeba zpoždění skutečně získat.

Dalším problémem je to, že `Lissy` je další aplikace, se kterou je nutné komunikovat, kvůli čemuž opět dochází zhruba ke 2,5-násobnému zpomalení celého výpočtu tras v závislosti na počtu spojů, které se v nich nacházejí, viz sekce 8.2.

7.2.4 Ohodnocení jednotlivých tras

Jakmile jsou získány všechny trasy s informacemi o zpožděních a cestách, lze je ohodnotit a na základě toho je seřadit od nejlepší po nejhorší.

V první části jsou vybrána ze všech řešení pouze ta, která jsou Pareto-optimální, protože OTP najde mnoho různých řešení, avšak velká část z nich je ve všech kritériích horší než jiná, tudíž tato řešení nejsou zajímavá. Pareto-optimální řešení se získávají podle kritérií popsaných v 6.1.

Normalizace kritérií

Poté, co zbydou pouze Pareto-optimální řešení, je potřeba hodnoty kritérií normalizovat, neboť různá kritéria mohou nabývat různých hodnot. Například celkový čas cesty se může pohybovat v rozmezí jednotek až stovek minut, počet přestupů bude ve většině případů menší než 10 a emise se mohou pohybovat od 0 do několika desítek až stovek tisíc gramů CO_2. Díky normalizaci se hodnota každého kritéria bude pohybovat v intervalu $\langle 0; 1 \rangle$. Pro výpočet je využita *Min-max* normalizace, jejíž vzorec je:

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}},$$

kde X_{max} je nejvyšší hodnota daného kritéria přes všechna řešení a X_{min} je nejmenší hodnota [31].

Váhovací algoritmus

Pareto-optimální řešení jsou poté ohodnocena váhovacím algoritmem a seřazena vzestupně podle této hodnoty. Celkové skóre řešení se vypočte jako:

$$s = \sum_{i=1}^{N} w_i * norm_i,$$

kde w_i je váha kritéria i a $norm_i$ je normalizovaná hodnota kritéria i a platí, že čím menší je hodnota s, tím je trasa lepší.

Váhy byly získány pomocí techniky AHP (*Analytical Hierarchy Process*), která dle [4] funguje tak, že dochází k porovnání každého kritéria s každým a pro každé toto porovnání je přiřazen koeficient důležitosti podle toho, jak moc je jedno kritérium důležité oproti jinému. Existují na výběr hodnoty 1, 3, 5, 7 a 9, kde 1 znamená, že kritérium A je stejně důležité jako kritérium B a 9 znamená, že kritérium A je velmi důležité oproti kritériu B. Hodnoty 2, 4, 6 a 8 se poté dají použít jako zjemnění přechodu mezi výše uvedenými hodnotami.

Důležité také je, že pokud je kritériu A přiřazena důležitost n oproti kritériu B, tak B bude mít oproti A důležitost $\frac{1}{n}$. Vyjádření důležitosti mezi jednotlivými kritérii lze pro přehlednost vyjádřit pomocí matice, která může pro kritéria C_1, C_2 a C_3 vypadat například takto:

$$\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \end{array} \begin{array}{c} \begin{array}{ccc} C_1 & C_2 & C_3 \end{array} \\ \begin{pmatrix} 1 & 3 & \frac{1}{5} \\ \frac{1}{3} & 1 & 8 \\ 5 & \frac{1}{8} & 1 \end{pmatrix} \end{array}.$$

Nenormalizované hodnoty vah se vypočtou jako geometrický průměr každého řádku matice, tedy pro příklad výše:

$$w_{C_1} = (1 * 3 * \frac{1}{5})^{\frac{1}{3}} = 0.84,$$
$$w_{C_2} = (\frac{1}{3} * 1 * 8)^{\frac{1}{3}} = 1.39,$$
$$w_{C_3} = (5 * \frac{1}{8} * 1)^{\frac{1}{3}} = 0.85.$$

Což po podělení celkovým součtem všech vah dává:

$$w_{C_1} = 0.27,$$
$$w_{C_2} = 0.45,$$
$$w_{C_3} = 0.28.$$

Pro výpočet vah pro tuto aplikaci označme kritéria ze sekce 6.1 jako T (celkový čas), E (emise), Z (zpoždění) a P (přestupy). Pro ně byla nastavena matice důležitosti takto:

$$\begin{array}{c} \\ T \\ E \\ Z \\ P \end{array} \begin{array}{c} \begin{array}{cccc} T & E & Z & P \end{array} \\ \begin{pmatrix} 1 & 5 & 3 & 7 \\ \frac{1}{5} & 1 & \frac{1}{5} & 3 \\ \frac{1}{3} & 5 & 1 & 7 \\ \frac{1}{7} & \frac{1}{3} & \frac{1}{7} & 1 \end{pmatrix} \end{array}.$$

Což po výpočtu geometrického průměru a normalizace dává váhy

$$w_T = 0.54,$$
$$w_E = 0.09,$$
$$w_Z = 0.31,$$
$$w_P = 0.05.$$

Tyto váhy byly určeny subjektivně na základě uživatelského dotazníku, z něhož vyplynulo, že pro většinu lidí je nejdůležitější celkový čas cesty. Naopak emise nejsou dle dotazníku vůbec důležité, avšak cílem aplikace je kvůli životnímu prostředí více preferovat hromadnou dopravu, proto mají 3. největší váhu.

Výhodou této metody oproti jiným, které pracují například s pořadím důležitosti pro kritéria, je, že lze měnit důležitosti mezi různými kritérii a tedy měnit jejich prioritu podle potřeby.

7.2.5 Bod vyzvednutí a návrat zpět

Jakmile dojde k ohodnocení všech tras a jejich seřazení na základě skóre, může dojít k dodatečným úpravám tras na základě nastavení uživatele.

První možností je nastavení bodu vyzvednutí, kdy si uživatel zvolí bod na mapě, v němž bude autem vyzvednut někým jiným a odvezen do cílové destinace. V tomto případě dojde nejprve k naplánování trasy z počátečního bodu do bodu vyzvednutí běžným způsobem a poté je ke každému řešení přidána další trasa z bodu vyzvednutí do cílové destinace, ve které je použito auto.

Druhou možností je, že si uživatel nastaví datum a čas návratu a aplikace kromě zobrazení cest tam zobrazí i cesty zpět. Takže plánování je prováděno v podstatě obráceně s tím, že už jsou známy přestupní body.

Nejprve je nutné získat všechny přestupní body, které se v daných řešeních nacházejí. Následně se přes OTP získají trasy, které vedou z cílové destinace do přestupního bodu s použitím hromadné dopravy. Pro tento případ je v OTP nastaveno, aby vrátil maximálně 5 řešení, protože následně nedochází k omezení jejich počtu jako se děje v sekci 7.2.4. Každé toto řešení je poté propojeno s trasou vedoucí z přestupního bodu zpět do bodu startovacího, a to s použitím auta.

7.3 Klientská část

Pro implementaci klientské části aplikace je použita knihovna `React`[8], díky níž lze vytvářet jednostránkové aplikace. Veškeré ikony, vstupy a seznamy jsou poté implementovány pomocí `React MUI`[9]. K vygenerování některých stylů byla použita umělá inteligence ChatGPT. Obrázek 7.4 ukazuje výsledný vzhled klientské části aplikace.

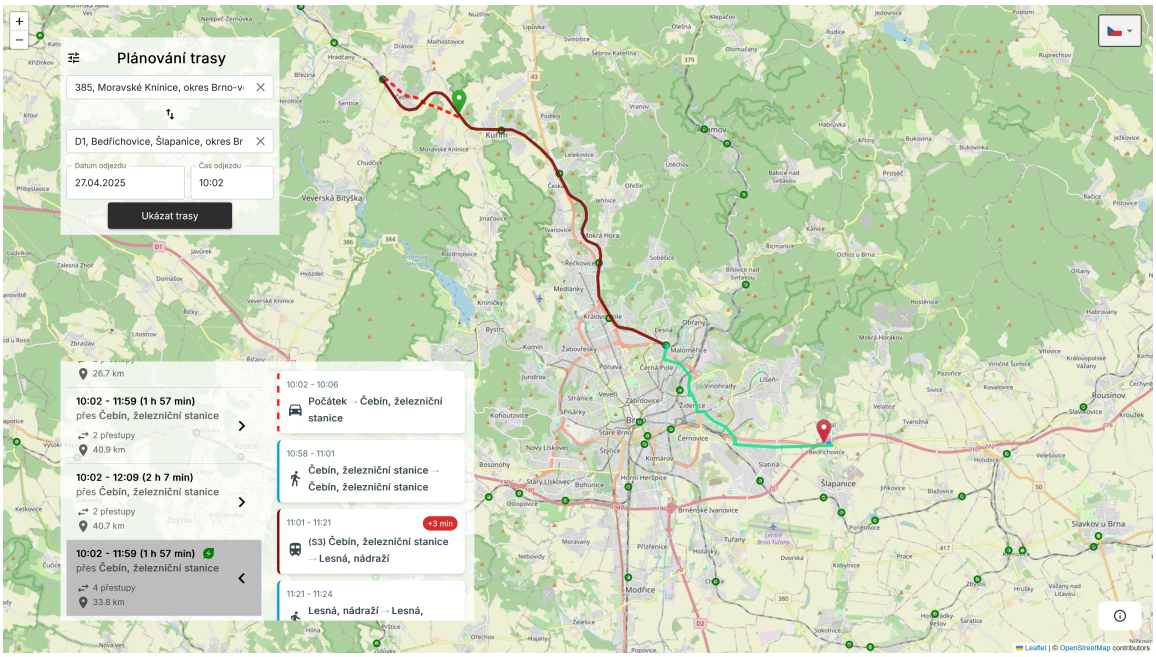
Pro práci s mapou je použita knihovna `React Leaflet`[10], což je rozšíření populárnější knihovny `Leaflet`[11], která umožňuje v aplikaci zobrazit mapu a je možné se v ní posouvat,

[8] https://react.dev/
[9] https://mui.com/material-ui/
[10] https://react-leaflet.js.org/
[11] https://leafletjs.com/

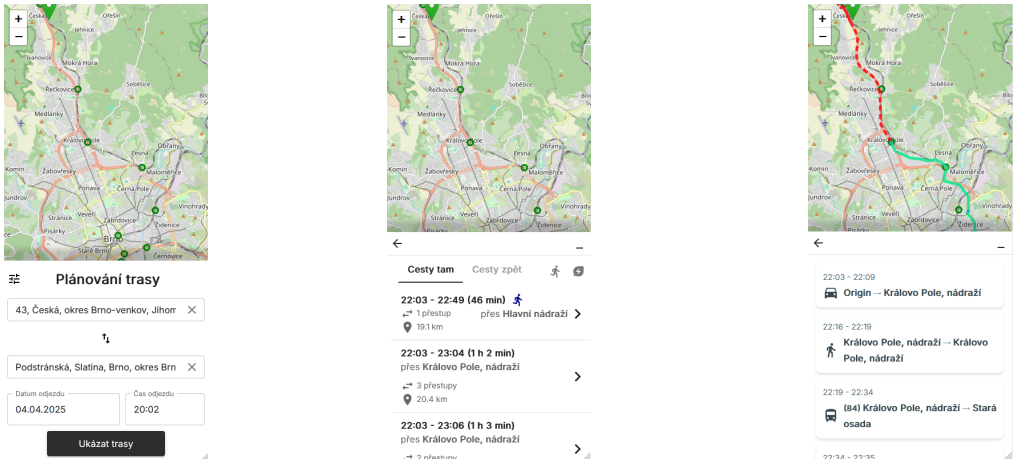
případně ji přibližovat a oddalovat. Jako podkladová mapa je zde použita ta od OpenStreetMap. Kromě interakce s mapou lze pomocí této knihovny do mapy vkládat různé objekty a ty se na základě svých souřadnic zobrazí na dané pozici na mapě.

Aplikace používá tyto objekty k vizualizaci následujících informací:

- **Uživatelské body** – pomocí ikony špendlíku jsou na mapě zobrazeny nejdůležitější body, jejichž polohu si volí sám uživatel. Dochází k vizualizaci počátečního bodu, který je na mapě zobrazen zelenou barvou, pro koncový bod je použita barva červená, pro přestupní bod barva oranžová a pro bod vyzvednutí je zvolena modrá barva. Ne vždy však dochází ke zobrazení všech bodů najednou, neboť povinný je pouze počáteční a koncový bod, ostatní jsou volitelné. Pro jednodušší změny tras lze po mapě přesouvat všechny body kromě přestupního a měnit tak jejich pozici.
- **Přestupní body** – po načtení aplikace se na mapě zobrazí veškeré dostupné přestupní body, jejichž počet se pohybuje okolo 200, což záleží na aktuálních GTFS souborech. Tyto body jsou zobrazeny jako zelené tečky a dávají uživateli informaci o tom, kde se dané body nacházejí. Kliknutím na daný bod lze zobrazit jeho název a jestli se v jeho okolí nachází nějaké parkoviště. Zároveň lze kliknutím na tlačítko bod natvrdo zvolit jako přestupní, nebo tuto možnost zrušit podle toho, jestli bod zvolen jako přestupní, či nikoliv.
- **Trajektorie** – při volbě každého řešení dojde k zobrazení jeho trajektorie na mapě. Trajektorie jsou všechny spočítány najednou při získání všech možných spojení a nepočítají se znovu při každém kliknutí na řešení. Trajektorie, které označují cestu autem z počátečního bodu do přestupního, jsou kvůli snazší identifikaci udělány čárkovaně a pro hromadnou dopravu jsou zobrazeny pomocí plné čáry. Jednotlivé prostředky hromadné dopravy jsou poté rozlišeny různými barvami opět pro zlepšení identifikace. Je podporována chůze, vlak, autobus, trolejbus, tramvaj a metro.



Obrázek 7.4: Hlavní obrazovka implementované klientské části aplikace.



Obrázek 7.5: Mobilní zobrazení, zleva: formulář pro plánování trasy, seznam vypočtených tras, detail trasy.

7.3.1 Formulář pro zadání trasy

Jediný prvek, který se po načtení aplikace kromě samotné mapy nachází, je formulář, který sám o sobě stačí k nalezení vhodné trasy. V aplikaci se rovněž nachází i pokročilé nastavení, které je popsáno v sekci 7.3.2. Hlavními prvky formuláře jsou dva vstupy, které určují počáteční a koncový bod. Volba probíhá tak, že uživatel klikne do jednoho ze vstupů, následně se změní kurzor indikující, že lze bod zvolit, a po kliknutí na mapu se informace o bodu ukáže v daném vstupu.

Pro lepší představu o tom, kde se daný bod nachází se primárně nepoužívají souřadnice, které jsou pro uživatele zpravidla nečitelné, ale zobrazí se adresa dané lokace. Pro získání adresy ze zeměpisných souřadnic je použita služba `Nominatim`[12]. V případě, že se adresu nepodaří získat, jsou zobrazeny zeměpisné souřadnice v pořadí zeměpisná šířka, zeměpisná délka.

7.3.2 Pokročilé nastavení

Kromě samotného základního formuláře, díky kterému lze zadat trasu, se v aplikaci nachází i pokročilé nastavení, které umožňuje uživateli ještě lépe specifikovat vlastnosti, které by trasy měly mít. Přímo v hlavním okně aplikace se nachází možnost přepnutí jazyka, který se na začátku nastaví podle jazyka prohlížeče uživatele. V současnosti je podporována pouze čeština a angličtina. Pro efektivní přepínání jazyků byla použita knihovna `i18next`[13].

Na další nastavení se může uživatel dostat ze vstupního formuláře, kde může uživatel nastavit přestupní bod, který se použije pro přestup z auta na hromadnou dopravu. Zde si ho může zvolit v případě, že hledá nějaký konkrétní, ale neví, kde na mapě se nachází. Tento seznam je provázán s přestupními body, které se nacházejí na mapě, tudíž změny provedené v tomto seznamu se projeví i na mapě a naopak.

Uživatel si rovněž může určit, které dopravní prostředky preferuje, může si vybrat i několik druhů najednou. Platí, že když žádný není zvolen, tak se mohou použít všechny druhy. Na výběr je z autobusu, vlaku, tramvaje, trolejbusu a metra. Tyto druhy dopravy jsou přímou součástí klientské části a lze je změnit pouze v kódu klienta.

[12] https://nominatim.openstreetmap.org/ui/search.html
[13] https://www.i18next.com/

Pokud se člověk nespokojí s tím, že aplikace může nalézt trasy, které nejsou úplně optimální, může si zaškrtnout možnost vyhledat optimální trasu, která vyzkouší všechny přestupní body, které nebyly odstraněny kvůli tomu, že jsou příliš daleko. Nevýhodou je však to, že nalezení těchto tras je zpravidla mnohem delší, neboť počet přestupních bodů, které musí být otestovány, může být o dost větší než v případě, že je tato možnost vypnuta. Důležité je rovněž zmínit, že zapnutí této možnosti nezaručuje nalezení lepšího řešení, než v případě standardního režimu, protože i ten může optimální řešení nalézt.

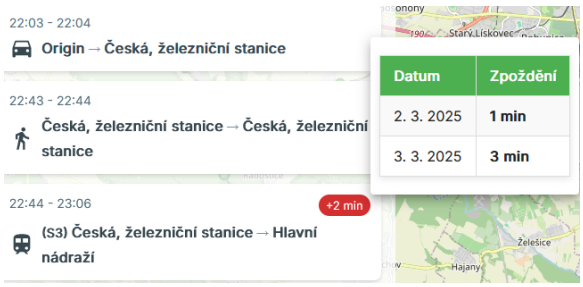
Jestliže daná osoba používá aplikaci například k dojíždění do práce a ví, že bude později vracet, může využít možnosti „Budu se vracet“. Po kliknutí zaškrtnutí této možnosti se objeví vstupy pro návratové datum a čas. Po odeslání požadavku server vrátí nejen vhodné cesty do cílové destinace, ale také cesty z destinace do počátečního bodu.

7.3.3 Zobrazení vypočtených tras

Po výpočtu se informace o jednotlivých trasách zobrazí v poli pod vstupním formulářem. Jestliže se nepodaří najít žádné trasy, je na to uživatel upozorněn varovným hlášením. Jednotlivé trasy se poté zobrazí ve formě seznamu, kde každá položka obsahuje základní sumarizační údaje. Kromě času odjezdu a příjezdu se zde rovněž nachází informace o tom, jak dlouho cesta trvá, jaká je její délka v kilometrech a kolik přestupů je třeba absolvovat. Pro každou vypočtenou trasu je rovněž ukázán i přestupní bod, od kterého by se mělo pokračovat hromadnou dopravou. Některé trasy mají i vedle času zobrazenou ikonu, které indikují, že je trasa nejrychlejší, případně je nejšetrnější k životnímu prostředí.

Po kliknutí na libovolnou trasu se napravo od seznamu objeví další seznam, který už obsahuje jednotlivé spoje daného trasy. Každý spoj obsahuje čas odjezdu a dojezdu, místo nástupu a výstupu a ikonu podle toho, o který druh dopravy se jedná.

Pokud existuje informace o zpoždění na daném spoji v daný čas, je informace o něm také zobrazena. Tato hodnota je průměrem všech zpoždění, které byly pro daný spoj v daném čase naměřeny v předchozích dnech. Pro detailní zobrazení, který den bylo jaké zpoždění, lze kliknout na prvek zobrazující průměrné zpoždění. Detailnější ukázka zobrazení zpoždění se nachází na obrázku 7.6.



Obrázek 7.6: Tabulka zobrazující vývoj zpoždění v různých dnech.

Nad seznamem tras lze poté vybírat mezi těmi trasami, které vedou od počátku do cíle a trasami, které vedou z cíle do počátku. Tyto možnosti se zobrazí pouze pokud existuje nenulový počet tras daného typu. Trasy jsou ve výchozím stavu seřazeny vzestupně podle skóre váhovacího algoritmu, avšak lze je seřadit podle rychlosti a podle spotřeby. V danou chvíli lze však řadit pouze podle jednoho parametru. K dispozici je řazení, vzestupně, sestupně i návrat do výchozího stavu.

Kapitola 8

Testování

Tato kapitola se zabývá testováním implementované aplikace, jejímž cílem je hlavně nalézt chyby, identifikovat možná rozšíření a body, které aplikaci zpomalují. Tato kapitola se zabývá uživatelským testováním a analýzou výkonnosti, díky které lze zjistit, jak je aplikace rychlá. Součástí kapitoly je také testování na jiných regionech s cílem ukázat, že lze aplikaci využít i pro jiné regiony. Poslední část se zabývá možnými budoucími rozšířeními.

8.1 Uživatelské testování

Uživatelské testování mělo za úkol zjistit, jak uživatelé s aplikací pracují, a také získat jejich zpětnou vazbu na použitelnost a další rozšíření aplikace. Testování probíhalo jak informovaně, kdy bylo uživateli řečeno, co má dělat, i neinformovaně, kdy uživateli byla řečena pouze hlavní funkce aplikace bez sdělení o tom, jak ji ovládat.

Jeden z prvních problémů nastal přímo u zadávání počátečního a koncového bodu, neboť uživatelé chtěli do vstupů přímo psát místo toho, aby bod vybrali po kliknutí na vstup přímo z mapy. Byla proto přidána informace o tom, jak body vybírat, což ale příliš nepomohlo, takže bude v budoucnu potřeba buď ještě explicitněji ukázat uživateli, jak výběr bodu provádět, nebo umožnit psaní adresy do vstupu, což by vyžadovalo využití služby `Nominatim`.

Díky uživatelskému testování se také podařily najít různé implementační chyby, které nebyly podchyceny během vývoje, například nefunkční získávání tras z mobilních zařízení s anglickou lokalizací kvůli nevhodnému formátu data. Všechny nalezené chyby tohoto typu byly později opraveny.

Práce byla prezentována na studentské konferenci Excel@FIT[1], kde si práci prohlédl Mgr. Jan Zvara, Ph.D. z oddělení dat, analýz a evaluací magistrátu města Brna, jehož aplikace zaujala, avšak bude třeba ještě další testování, aby se ukázalo, zda je aplikace pro Brno skutečně vhodná. Díky zpětné vazbě uživatelů z konference a testování vyplynula i potenciální rozšíření aplikace, kterými se zabývá sekce 8.4.

8.2 Výkonnostní testování

Aplikace nabízí různé konfigurace, které ovlivňují její výkon, ať už se jedná o počet shluků, vyzkoušení všech možných tras nebo použití aplikace `Lissy`. Níže se nacházejí tabulky s výsledky měření, kdy pro každou konfiguraci bylo provedeno 5 měření, z nichž byl vypočítán

[1] https://excel.fit.vutbr.cz/

průměr. Měření probíhalo s vypnutou i zapnutou aplikací `Lissy` a s lokální i produkční verzí OTP. Lokální verze běží na procesoru `AMD Ryzen 7` a produkční verze na `AMD Ryzen 5`, tudíž od produkční části je očekávaná nižší výkonnost i s ohledem na delší síťovou komunikaci. Výsledky těchto měření se nacházejí v tabukách 8.1 a 8.2.

	Lokální OTP	**Produkční OTP**
15 shluků	4,88 s	6,27 s
25 shluků	7,30 s	8,80 s
15 shluků s vracením	5,62 s	7,14 s
25 shluků s vracením	7,96 s	10,10 s
vyzkoušení všech (115) možností	25,74 s	32,78 s
předem zvolený přestupní bod	0.56 s	0.92 s

Tabulka 8.1: Výsledky měření výkonnosti se zapnutou aplikací `Lissy`.

	Lokální OTP	**Produkční OTP**
15 shluků	2,88 s	4,16 s
25 shluků	3,61 s	5,30 s
15 shluků s vracením	3,35 s	4,71 s
25 shluků s vracením	4,36 s	6,38 s
vyzkoušení všech (115) možností	10,13 s	16,09 s
předem zvolený přestupní bod	0.38 s	0.76 s

Tabulka 8.2: Výsledky měření výkonnosti s vypnutou aplikací `Lissy`.

Z tabulek lze vyčíst, že lokální instance OTP má nezanedbatelně vyšší výkonnost, než produkční verze, což souvisí s tím, že lokálně verze běží na silnějším procesoru a také s tím, že není potřeba komunikovat s žádným jiným počítačem. Avšak ukázalo se, že zásadní vliv na výkonnost má rovněž i `Lissy`, kdy je čas získání tras i 2x vyšší než v případě nepoužití této aplikace. Řešením tohoto problému by bylo použití lepšího procesoru, zlepšení algoritmu pro získávání přestupních bodů nebo zrychlení aplikace `Lissy`, případně přednačtení dat z ní. Rovněž také platí, že čím více přestupních bodů je ve výpočtu uvažováno, tím déle celý výpočet trvá.

8.3 Testování mimo Jihomoravský kraj

Jelikož OTP umožňuje načtení GTFS souborů i map pro libovolný region, tak lze aplikaci nasadit i do jiných oblastí, které tyto soubory poskytují. Pro otestování škálovatelnosti a celkové funkčnosti aplikace tak byly použity i jiné větší oblasti.

8.3.1 Středočeský kraj

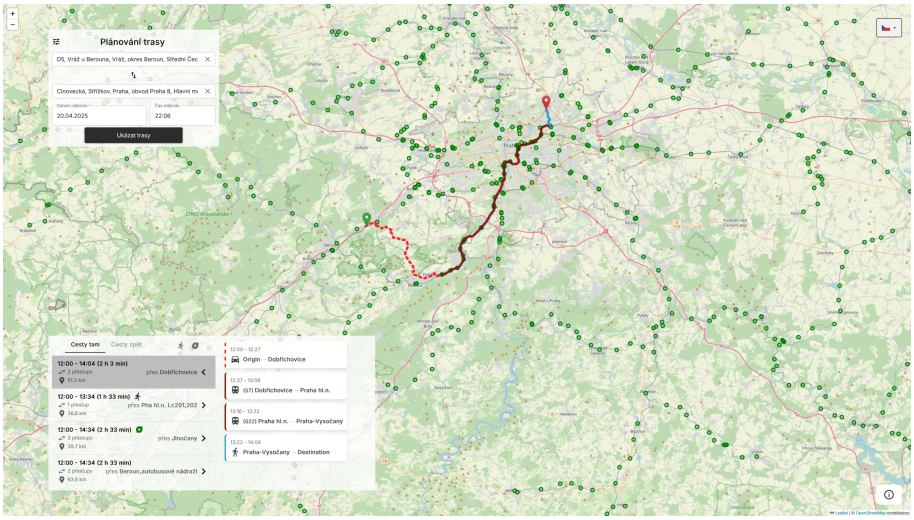
První vybranou oblastí se stal Středočeský kraj, protože ten díky Pražské integrované dopravě nabízí největší GTFS soubory v České republice, ale stále se jedná o relativně malou oblast, kterou lze otestovat na domácím počítači.

Ačkoliv se v tomto regionu nachází více než 900 přestupních bodů, tak bylo možné se plynule pohybovat po mapě GTFS soubory pro PID navíc obsahují `shapes.txt`, pomocí

kterého dokáže OTP vrátit přesné souřadnice trajektorie daného spoje, což lze vidět na obrázku 8.1.

Výpočty tras jsou obecně rychlejší než v případě Jihomoravského kraje, protože není použita aplikace `Lissy`, která by přesnou trajektorii vracela. Při větším počtu shluků však může být výpočet pomalejší hlavně kvůli vyššímu počtu přestupních bodů.

Během testování však vyvstala otázka, zdali je 15 shluků jako základní počet dostačující, neboť při takovém počtu přestupních bodů se může snadno stát, že shlukovací algoritmus nenalezne přestupní bod, který je lepší než jiné. Při navýšení základního počtu shluků však může dojít k prodloužení času výpočtu.



Obrázek 8.1: Ukázka použití aplikace pro Středočeský kraj s využitím GTFS Pražské integrované dopravy.

8.3.2 Švýcarsko

Jelikož GTFS soubory používají i celé státy, tak se nabízelo tuto aplikaci otestovat i na jednom z těchto států. Bylo zvoleno Švýcarsko, neboť GTFS soubory používá, a zároveň je dostatečně malé na to, aby k testu mohlo dojít i na domácím počítači. Jednalo se o rychlý experiment a nepředpokládá se, že by aplikace byla nasazena pro tak velkou oblast.

Zatímco pro Jihomoravský a Středočeský kraj stačilo pro chod OTP 2 GB RAM, tak pro Švýcarsko je nutno minimálně 16 GB RAM nejen kvůli větší mapě, ale také kvůli větším GTFS souborům. Je však potřeba zmínit, že švýcarské GTFS nepokrývá pouze Švýcarsko, ale také část Francie, Itálie nebo Německa.

Co se týká přestupních bodů, tak byly uvažovány pouze vlakové stanice, kterých i tak bylo okolo 3100. To už se trochu projevilo na výkonnosti na straně klienta a pohyb po mapě nebyl úplně plynulý. To by však šlo zajistit tak, že by docházelo k získávání pouze těch bodů, které by se v danou chvíli nacházely na obrazovce uživatele.

Díky shlukování však probíhaly výpočty podobně jako v případě Středočeského kraje. Bohužel švýcarské GTFS neobsahuje soubor `shapes.txt`, takže pro hromadnou dopravu byly sousední zastávky propojeny přímkou. Avšak podobně jako v případě Středočeského kraje by bylo pravděpodobně potřeba zvýšit počet shluků, které se vytvoří, nebo z daných shluků vyzkoušet více než jeden přestupní bod.

8.4 Možná rozšíření

Tyto nápady na možná rozšíření vznikly buď na základě uživatelského testování nebo v průběhu psaní této práce, ale zatím neexistují vhodné prostředky pro jejich implementaci.

- **Rozšířené informace o parkování** – pro daný přestupní bod aplikace v současnosti dokáže říct pouze, jestli se v jejím okolí nachází parkoviště, či nikoliv. Chybí jakékoliv informace o tom, kde přesně se parkoviště nacházejí, případně jaká je jejich cena, momentální kapacita nebo otevírací doba. Rozšíření by mělo tyto informace (pokud jsou dostupné) zobrazit po kliknutí na daný přestupní bod.
- **Vizualizace zpoždění a kolon pro automobilovou dopravu** – Magdaléna Ondrušková v [53] implementuje aplikaci, která dokáže získat historické informace o zácpách v jednotlivých ulicích Brna s použitím dat od společnosti Waze. Dříve tato data existovala pouze pro Brno, ale dochází k postupnému pokrytí celého Jihomoravského kraje. Tato data mohou být užitečná i pro tuto aplikaci, neboť by kromě informací o zpoždění hromadné dopravy mohla nabízet i informace o zácpách a předpokládané zpoždění pro auta, případně nabízet alternativní trasy, které zpoždění minimalizují.
- **Výpočet ceny přepravy** – Dalším z rozšíření může být výpočet ceny pro každou trasu jako to například nabízí aplikace od IDS JMK[2]. Odhadovanou cenu pro automobil lze spočítat pomocí vzdálenosti a průměrné spotřeby paliva, avšak pro hromadnou dopravu to není možné tak snadno, protože cena se kromě času a věku cestujícího odvíjí i od počtu zón, ve kterých daný spoj staví. Některé spoje však určitými zónami pouze projíždějí a tyto průjezdy se do ceny nepočítají. V tuto chvíli nelze spolehlivě detekovat, kterými zónami se pouze projíždí a kterými ne, proto to není v tuto chvíli není v aplikaci implementováno.

[2] https://www.idsjmk.cz/index

Kapitola 9

Závěr

Cílem této práce bylo vytvořit dopravní plánovač, který kombinuje automobilovou a hromadnou dopravu a který může pomoci dvěma hlavním typům uživatelů, a sice dojíždějícím a turistům. Přestup mezi těmito druhy dopravy probíhá v předdefinovaných přestupních bodech, kterými jsou železniční zastávky a autobusová nádraží. Hlavním úkolem tohoto plánovače je nalézt přestupní bod takový, aby cesta byla co nejefektivnější z pohledu celkového času cesty, zpoždění, množství emisí a počtu přestupů.

Aplikace detailně zobrazuje veškeré informace o trasách, ať už se jedná o časy odjezdů a příjezdů jednotlivých spojů, odkud kam jedou nebo počet přestupů. Spoje rovněž obsahují informaci o průměrném zpoždění a uživatel si může zobrazit vývoj zpoždění za poslední dny. Pro každou cestu je na mapě zobrazena detailní trajektorie pohybu, ze které lze vyčíst, kudy který spoj jede.

V rámci uživatelského nastavení lze zvolit přestupní bod použit při výpočtu a dále také specifikovat bod vyzvednutí, který lze použít v případě, že uživatele vyzvedne příbuzný nebo například taxi. Součástí nastavení je také možnost zvolit si dopravní prostředky, které budou použity k přepravě, a také funkce vracení, která dovoluje nastavení zpáteční cesty a která kromě cest tam vrátí i cesty zpět. V neposlední řadě lze aplikovat možnost nalezení nejlepšího řešení, která za cenu delšího času výpočtu vyzkouší všechny okolní přestupní body a nalezne ty nejvhodnější.

Aplikace byla nasazena na školním serveru[1], otestována uživateli a prezentována na konferenci Excel@FIT [68]. Některé problémy a chyby byly na základě těchto činností opraveny a zároveň se díky nim objevily i podněty na další rozšíření, například výpočet celkové ceny nebo analýza dopravní situace na trase pomocí Waze.

[1] https://dexter.fit.vutbr.cz/carpub

Literatura

[1] *Https://transmodel-cen.eu/index.php/netex/* online. Dostupné z: https://transmodel-cen.eu/index.php/netex/. [cit. 2025-05-15].

[2] *ArcGIS | Geospatial Platform – GIS Software for Business & Government* online. Dostupné z: https://www.esri.com/en-us/arcgis/geospatial-platform/overview. [cit. 2025-04-15].

[3] Aronoff, S. Geographic information systems: A management perspective. *Geocarto International*. Taylor & Francis, 1989, sv. 4, č. 4, s. 58–58. Dostupné z: https://doi.org/10.1080/10106048909354237.

[4] Barfod, M. a Leleur, S., ed. *Multi-criteria decision analysis for use in transport decision making*. 2. vyd. DTU Transport, 2014.

[5] Booth, B.; Mitchell, A. et al. *Getting started with ArcGIS*. Esri Redlands, CA, USA, 2001.

[6] Bozyiğit, A.; Alankuş, G. a Nasiboğlu, E. Public transport route planning: Modified dijkstra's algorithm. In: *2017 International Conference on Computer Science and Engineering (UBMK)*. 2017, s. 502–505.

[7] Burgetová, I. *Shluková analýza*. 2024. Texty k přednáškám z předmětu ZZN.

[8] Butler, H.; Daly, M.; Doyle, A.; Gillies, S.; Schaub, T. et al. *The GeoJSON Format* RFC 7946. RFC Editor, srpen 2016. Dostupné z: https://doi.org/10.17487/RFC7946.

[9] *Public transport – Network and Timetable Exchange (NeTEx) – Part 4: Passenger Information European Profile*. Standard. Brussels, Belgium: European Committee for Standardization, duben 2020.

[10] *Public transport – Network and timetable exchange (NeTEx) – Part 5: Alternative modes exchange format*. Standard. Brussels, Belgium: European Committee for Standardization, červen 2022.

[11] *Public transport – Network and timetable exchange (NeTEx) – Part 6: European Passenger Information Accessibility Profile*. Standard. Brussels, Belgium: European Committee for Standardization, červen 2024.

[12] Chopde, N. R. a Nichat, M. Landmark based shortest path detection by using A* and Haversine formula. *International Journal of Innovative Research in Computer and Communication Engineering*, 2013, sv. 1, č. 2, s. 298–302.

[13] Citymapper. *Citymapper – The Ultimate Transport App* online. Dostupné z: https://citymapper.com/. [cit. 2024-11-18].

[14] Citymapper. *GPX: the GPS Exchange Format* online. 2023. Dostupné z: https://www.topografix.com/gpx.asp. [cit. 2025-05-11].

[15] Commission, E.; Mobility, D.-G. for a Transport. *Transport in the European Union – Current trends and issues*. Publications Office of the European Union, 2024.

[16] Delling, D.; Pajor, T. a Werneck, R. Round-Based Public Transit Routing. In: *Proceedings of the 14th Meeting on Algorithm Engineering and Experiments (ALENEX'12)*. Proceedings of the 14th Meeting on Algorithm Engineering and Experiments (ALENEX'12). Society for Industrial and Applied Mathematics, Leden 2012. Dostupné z: https://www.microsoft.com/en-us/research/publication/round-based-public-transit-routing/.

[17] *Dot Density Maps* online. Dostupné z: https://www.zeemaps.com/world-of-maps/dot-density-maps/. [cit. 2024-12-19].

[18] Dąbrowski, A. a Klemba, S. Assessment of Rail Transport Conditions and of the Possibility to Increase the Role of Rail Transport in the Public Transport System of the Urban Functional Area of Olsztyn. *Problemy Kolejnictwa – Railway Reports*, Červen 2022, sv. 66, s. 79–96.

[19] Ekenes, K. *Five ways to visualize point density using the same dataset* online. Dostupné z: https://www.esri.com/arcgis-blog/products/js-api-arcgis/mapping/five-ways-to-visualize-point-density-using-the-same-dataset. [cit. 2025-04-29].

[20] *Geographic information – Procedures for item registration – Part 1: Fundamentals*. Standard. Geneva, Switzerland: International Organization for Standardization, květen 2016.

[21] Eppelbaum, L.; Ben Avraham, Z.; Youri, K.; Cloetingh, S. a Kaban, M. Combined Multifactor Evidence of a Giant Lower-Mantle Ring Structure below the Eastern Mediterranean. *Positioning*, Duben 2020, sv. 11, s. 11–32.

[22] *Extended GTFS Route Types* online. Dostupné z: https://developers.google.com/transit/gtfs/reference/extended-route-types. [cit. 2025-01-24].

[23] Feher, D. *Western Europe Physical Map* online. Dostupné z: https://www.freeworldmaps.net/europe/western/physical.html. [cit. 2024-12-19].

[24] Field, K. Cartograms. In: Wilson, J. P., ed. *The Geographic Information Science & Technology Body of Knowledge*. Q3 2017. University Consortium for Geographic Information Science, 2017. ISBN 9780892912674.

[25] *GraphQL | A query language for your API* online. Dostupné z: https://graphql.org/. [cit. 2024-12-02].

[26] Grubler, A. a Nakicenovic, N. *Evolution of Transport Systems: Past and Future*. IIASA Research Report. IIASA, Laxenburg, Austria, June 1991. Dostupné z: https://pure.iiasa.ac.at/id/eprint/3486/.

[27] Görtler, J.; Spicker, M.; Schulz, C.; Weiskopf, D. a Deussen, O. Stippling of 2D Scalar Fields. *IEEE Transactions on Visualization and Computer Graphics*, 2019, sv. 25, č. 6, s. 2193–2204.

[28] Harris, I.; Wang, Y. a Wang, H. ICT in multimodal transport and technological trends: Unleashing potential for the future. *International Journal of Production Economics*, 2015, sv. 159, s. 88–103. ISSN 0925-5273. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0925527314002837.

[29] Hart, P. E.; Nilsson, N. J. a Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 1968, sv. 4, č. 2, s. 100–107.

[30] *Create a Geographic Heat Map* online. Dostupné z: https://maply.com/geographic-heat-map. [cit. 2024-12-19].

[31] Henderi, H.; Wahyuningsih, T. a Rahwanto, E. Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. *International Journal of Informatics and Information Systems*, 2021, sv. 4, č. 1, s. 13–20.

[32] Hobart M. King, R. a Junkie, M. *Types of Maps: Political, Physical, Google, Weather, and More* online. Dostupné z: https://geology.com/maps/types-of-maps/. [cit. 2024-12-18].

[33] Hogräfer, M.; Heitzler, M. a Schulz, H.-J. The State of the Art in Map-Like Visualization. *Computer Graphics Forum*, 2020, sv. 39, č. 3, s. 647–674. Dostupné z: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14031.

[34] Hynek, J. *Impact of Subjective Visual Perception on Automatic Evaluation of Dashboard Design*. 2019. Disertační práce. Fakulta informačních technologií VUT Brno.

[35] Jittrapirom, P.; Caiati, V.; Feneri, A. M.; Ebrahimigharehbaghi, S.; Alonso González, M. J. et al. Mobility as a service: A critical review of definitions, assessments of schemes, and key challenges. *Urban Planning*. Cogitatio Press, 2017, sv. 2, č. 2, s. 13–25.

[36] Jurkowska, J. *Geographic Coordinate Systems 101: A Primer for Software Generalists* online. Dostupné z: https://8thlight.com/insights/geographic-coordinate-systems-101. [cit. 2025-01-30].

[37] K., K. *Mapy pražského metra* online. Dostupné z: https://metro.zarohem.cz/mapa.html. [cit. 2025-01-27].

[38] *KML Tutorial* online. Dostupné z: https://developers.google.com/kml/documentation/kml_tut. [cit. 2025-05-11].

[39] komunikace, B. *Brněnské komunikace a.s.* online. Dostupné z: https://www.bkom.cz/. [cit. 2024-11-23].

[40] Kraak, M. a Ormeling, F. *Cartography: Visualization of spatial data.* 3. vyd. Pearson Education, listopad 2010. ISBN 978-0-273-72279-3.

[41] Lazúr, I. J. *Analýza a vizualizace dat hromadné dopravy města Brna* online. Brno, 2023. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií, Ústav informačních systémů. Vedoucí práce Ing. Jiří Hynek, P. Dostupné z: http://hdl.handle.net/11012/211930. [cit. 2025-04-06].

[42] Lazúr, J. *Lissy API for External Use* online. Dostupné z: https://github.com/Jorgen98/Lissy/blob/main/README_API_EXTERNAL.md. [cit. 2025-04-08].

[43] Lenat, D. B. The nature of heuristics. *Artificial Intelligence*, 1982, sv. 19, č. 2, s. 189–249. ISSN 0004-3702. Dostupné z: https://www.sciencedirect.com/science/article/pii/0004370282900364.

[44] Litman, T. *Introduction to multi-modal transportation planning.* Victoria Transport Policy Institute Canada, 2017.

[45] Logisber. *What are the advantages and disadvantages of multimodal transport?* online. Dostupné z: https://logisber.com/en/blog/multimodal-transport-advantages-disadvantages. [cit. 2024-11-18].

[46] Magzhan, K. a Jani, H. M. A review and evaluations of shortest path algorithms. *Int. J. Sci. Technol. Res*, 2013, sv. 2, č. 6, s. 99–104.

[47] Memon, I. A.; Madzlan, N.; Talpur, M. A. H.; Hakro, M. R. a Chandio, I. A. A review on the factors influencing the Park-and-Ride traffic management method. *Applied Mechanics and Materials.* Trans Tech Publ, 2014, sv. 567, s. 663–668.

[48] Mera, A. Quiña; Fernandez, P.; García, J. M. a Ruiz Cortés, A. GraphQL: A Systematic Mapping Study. *ACM Comput. Surv.* New York, NY, USA: Association for Computing Machinery, Únor 2023, sv. 55, č. 10. ISSN 0360-0300. Dostupné z: https://doi.org/10.1145/3561818.

[49] MGISS. *The Different Types of GIS Data* online. Dostupné z: https://mgiss.co.uk/the-different-types-of-gis-data/. [cit. 2025-04-13].

[50] MobilityData. *General Transit Feed Specification* online. Dostupné z: https://gtfs.org/. [cit. 2024-11-27].

[51] Nha, V. T. N.; Djahel, S. a Murphy, J. A comparative study of vehicles' routing algorithms for route planning in smart cities. In: *2012 First International Workshop on Vehicular Traffic Management for Smart Cities (VTM).* 2012, s. 1–6.

[52] Noto, M. a Sato, H. A method for the shortest path search by extended Dijkstra algorithm. In: *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0.* 2000, sv. 3, s. 2316–2320 vol.3.

[53] Ondrušková, I. M. *Analýza a vizualizace dopravních dat města Brna* online. Brno, 2024. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií, Ústav informačních systémů. Vedoucí práce Ing. Jiří Hynek, P. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=267517. [cit. 2025-04-19].

[54] *OpenTripPlanner 2* online. 2025. Dostupné z: https://docs.opentripplanner.org/en/latest/. [cit. 2024-12-02].

[55] Pang, C.-I. a Biuk Aghai, R. P. Wikipedia world map: method and application of map-like wiki visualization. In: *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*. New York, NY, USA: Association for Computing Machinery, 2011, s. 124–133. WikiSym '11. ISBN 9781450309097. Dostupné z: https://doi.org/10.1145/2038558.2038579.

[56] Parkhurst, G. Influence of bus-based park and ride facilities on users' car traffic. *Transport Policy*, 2000, sv. 7, č. 2, s. 159–172. ISSN 0967-070X. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0967070X00000068.

[57] *What Is a Political Map? Understanding Its Purpose and Importance* online. Dostupné z: https://www.whiteclouds.com/what-is/what-is-a-political-map-understanding-its-purpose-and-importance/. [cit. 2024-12-19].

[58] *Encoded Polyline Algorithm Format* online. Dostupné z: https://developers.google.com/maps/documentation/utilities/polylinealgorithm. [cit. 2025-01-30].

[59] Raju, P. Fundamentals of geographical information system. *Satellite Remote Sensing and GIS Applications in Agricultural Meteorology*. Citeseer, 2006, sv. 103.

[60] Rodrigue, J.-P. *The geography of transport systems*. 5. vyd. Routledge, 2020. ISBN 9780429346323.

[61] Saif, J. *Google Maps Pros and Cons* online. Dostupné z: https://medium.com/@jamsaif/google-maps-pros-and-cons-d3ff8bb54126. [cit. 2025-04-19].

[62] Shaheen, S. a Finson, R. Intelligent transportation systems, 2013. Dostupné z: https://escholarship.org/uc/item/3hh2t4f9.

[63] Shaktawat, Y. S. *What is ArcGIS?* online. Dostupné z: https://geospatialworld.net/blogs/what-is-arcgis/. [cit. 2025-04-15].

[64] TECHBREAUX. *A* Search* online. Dostupné z: https://www.codecademy.com/resources/docs/ai/search-algorithms/a-star-search. [cit. 2025-01-02].

[65] *Transit – the best app for buses and trains* online. Dostupné z: https://transitapp.com/. [cit. 2025-04-22].

[66] Tufte, E. R. a Graves Morris, P. R. *The visual display of quantitative information*. Graphics press Cheshire, CT, 1983.

[67] Vlachopanagiotis, T.; Grizos, K.; Georgiadis, G. a Politis, I. Public Transportation Network Design and Frequency Setting: Pareto Optimality through Alternating-Objective Genetic Algorithms. *Future Transportation*, 2021, sv. 1, č. 2, s. 248–267. ISSN 2673-7590. Dostupné z: https://www.mdpi.com/2673-7590/1/2/15.

[68] Václavík, J. Multimodální dopravní plánovač Jihormoravského kraje. In: *Excel@FIT*. Fakulta informačních technologií VUT v Brně, Květen 2025, s. 2. Dostupné z: https://excel.fit.vutbr.cz/submissions/2025/028/28.pdf.

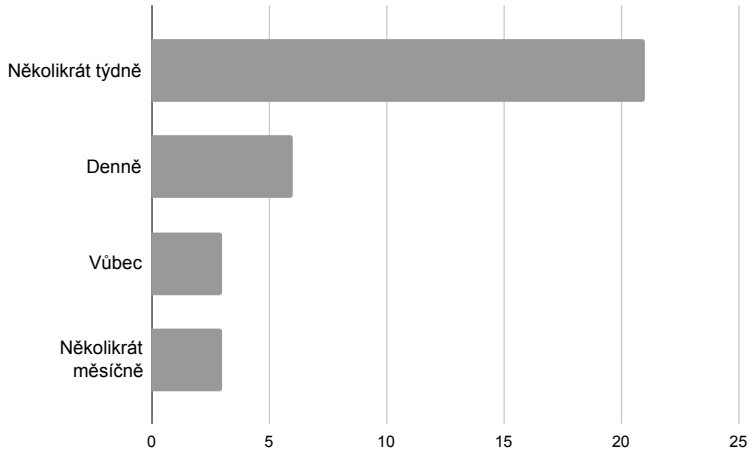
[69] Witt, S. Trip-Based Public Transit Routing. In: Bansal, N. a Finocchi, I., ed. *Algorithms - ESA 2015*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, s. 1025–1036. ISBN 978-3-662-48350-3.

[70] worldmapper. *Danger Zones: Mapping Europe's Earthquakes* online. Dostupné z: https://www.viewsoftheworld.net/?p=5652. [cit. 2025-01-27].

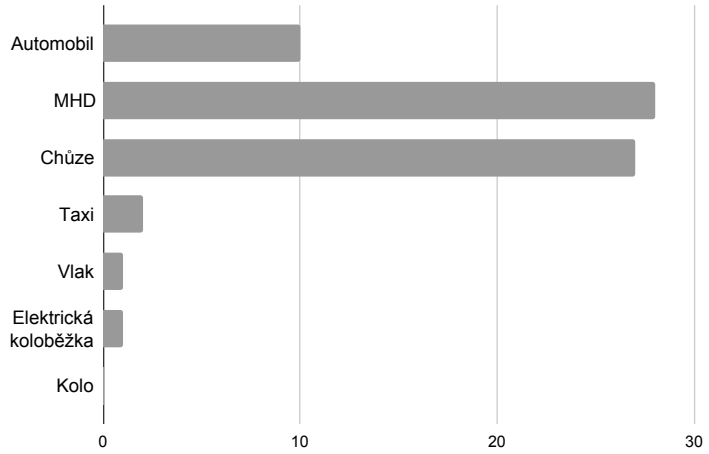
Příloha A

Výsledky uživatelského dotazníku

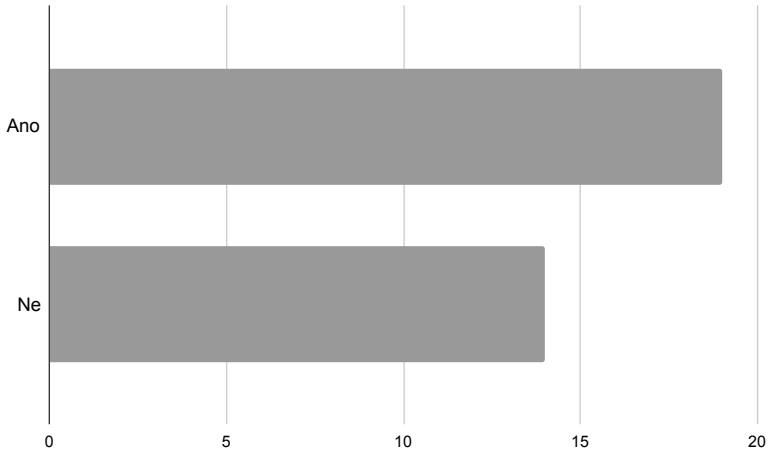
Níže se nacházejí grafy, které byly vyhotoveny v rámci dotazníku analyzující uživatele a jejich dopravní zvyklosti.



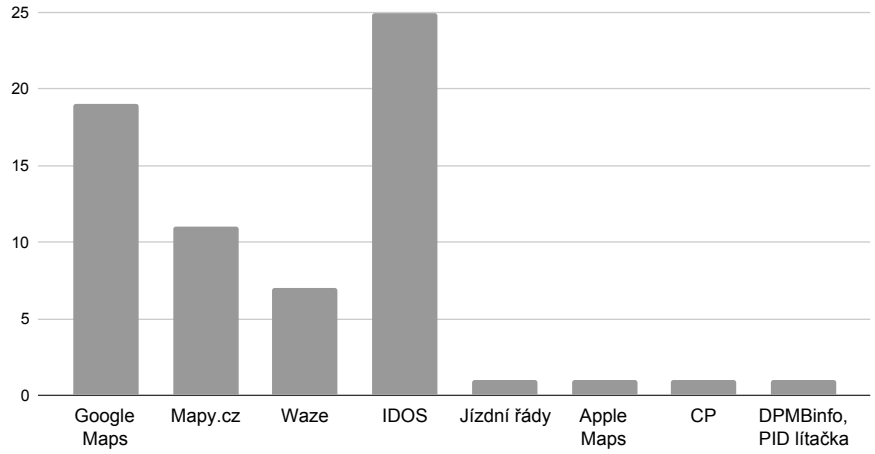
Obrázek A.1: Odpovědi na otázku, jak často lidé cestují do školy nebo do práce.



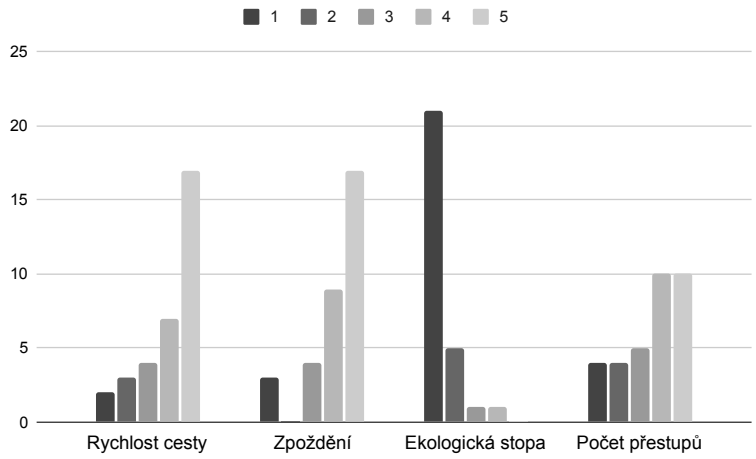
Obrázek A.2: Dopravní prostředky použity během cestování do školy nebo do práce.



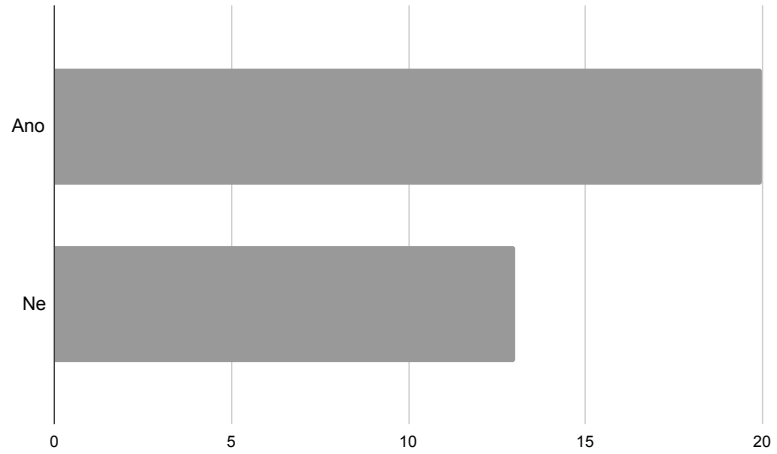
Obrázek A.3: Graf ukazující, jestli lidé během cestování používají více druhů dopravy, či nikoliv.



Obrázek A.4: Graf ukazující, které dopravní plánovače lidé nejčastěji používají.



Obrázek A.5: Graf ukazující, jak uživatelé vnímají důležitost jednotlivých kritérií během cestování (1 – nejméně důležité, 5 – velmi důležité).



Obrázek A.6: Výsledky ankety, zdali by uživatelé ocenili aplikaci, která kombinuje automobilovou a hromadnou dopravu.