

# STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

Internetová platforma pro vizualizaci aktuální  
polohy vozů MHD

Jakub Kacálek, Jan Volhejn  
Pardubický kraj

Pardubice, 2023

# STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

Internetová platforma pro vizualizaci aktuální  
polohy vozů MHD

Internet platform for visualizing the current  
location of public transport vehicles

**Autoři:** Jakub Kacálek, Jan Volhejn

**Škola:** DELTA - Střední škola informatiky a ekonomie, s.r.o.

**Kraj:** Pardubický kraj

**Konzultant:** Bc. Vladka Janů

# Prohlášení

Prohlašujeme, že jsme svou práci SOČ vypracovali samostatně a použili jsme pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašujeme, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemáme závažný důvod proti zpřístupnění této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Pardubicích dne 24. března 2023 .....

Jakub Kacálek, Jan Volhejn

## **Poděkování**

Chtěli bychom poděkovat Bc. Vladce Janů za odborný dohled a vedení technické stránky projektu a panu Ing. Jiřímu Formánkovi za značnou pomoc v komunikaci s dopravci. Dále bychom chtěli poděkovat všem, kteří se podíleli na testování platformy.

## **Abstrakt**

Práce SOČ dokumentuje vývoj platformy pro vizuální zobrazování pohybů vozidel hromadné dopravy. Platforma nabízí aktuální pohled na polohu hromadné dopravy pomocí webové a mobilní aplikace. Realizace projektu zahrnuje backendovou část a webový frontend.

## **Klíčová slova**

Programování; GraphQL; Progresivní webová aplikace; Yarn Workspaces; NextJS; Go programovací jazyk

## **Abstract**

This thesis documents development of a platform for visual display of public transport vehicles. The platform offers an up-to-date view of the location of public transport using web and mobile application. The project implementation includes a backend and a web frontend.

## **Keywords**

Programming; GraphQL; Progressive Web Application; Yarn Workspaces; NextJS; Go programming language

# OBSAH

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Teoretická část</b>	<b>3</b>
2.1	Inspirace projektu . . . . .	3
2.2	Pojmy . . . . .	3
2.3	Technologie . . . . .	5
<b>3</b>	<b>Realizace</b>	<b>9</b>
3.1	Architektura . . . . .	9
3.2	Backend . . . . .	10
3.3	Webová aplikace . . . . .	13
<b>4</b>	<b>Bezpečnost</b>	<b>21</b>
<b>5</b>	<b>Závěr</b>	<b>22</b>

# 1 ÚVOD

V dnešním rychlém světě je doprava důležitou součástí našeho každodenního života. Od dojíždění do práce až po cestování na dlouhé vzdálenosti se ve velké míře spoléháme na dopravní společnosti, které nás dopraví tam, kam potřebujeme. Jednou z největších výzev pro dopravní společnosti je ku příkladu informování cestujících o aktuální poloze jejich spoje. Bez aktualizací v reálném čase mohou cestující zažívat frustraci a nepříjemnosti, což vede k negativní zákaznické zkušenosti.

Mnoho dopravců disponuje informacemi o svých spojích, které využívají jak na svých dispečincích, tak na informačních tabulích na některých zastávkách. Tyto informace jsou však často nedostupné pro cestující. Tato nedostupnost informací o spojích je způsobena tím, že dopravci nemají jednotný způsob, jakým sdílí informace s cestujícími. Většina dopravců používá své vlastní webové stránky, které jsou často neaktualizované a nekompatibilní s mobilními zařízeními.

Cílem této práce je navržení a realizace webové platformy, která umožní dopravním společnostem sdílet informace a provozní upozornění s cestujícími. Tato platforma má potenciál zlepšit způsob, jakým cestující interagují s dopravními společnostmi a poskytnout jim přesné a aktuální informace, které jim pomohou činit informovaná rozhodnutí o jejich cestovních plánech.

## 2 TEORETICKÁ ČÁST

### 2.1 INSPIRACE PROJEKTU

Motivace k tomuto projektu vznikla absencí podobného řešení ve městě Pardubice.

#### 2.1.1 Existující řešení

Řešení pro zobrazování aktuální polohy již existuje v některých městech České republiky, tato řešení jsou však většinou uživatelsky nepřívětivá, nebo nezobrazují informace o spojení. Zde jsou některá řešení, u kterých jsme se inspirovali.

- <https://mapa.pid.cz/> – Praha a Středočeský kraj (PID)
- <https://www.mpvnet.cz/odis/map> – Liberecký kraj (IDOL)

### 2.2 POJMY

#### 2.2.1 WebSocket

WebSocket [1] je technologie pro obousměrnou komunikaci mezi webovým prohlížečem a serverem. Umožňuje otevření komunikačního kanálu mezi klientem a serverem, přes který mohou obě strany posílat a přijímat zprávy v reálném čase. To je velmi užitečné pro aplikace, které potřebují získávat aktualizace nebo zprávy od serveru bez nutnosti, aby klient musel ručně požadovat nové informace. WebSocket poskytuje efektivnější způsob komunikace než technologie jako HTTP [2], které vyžadují, aby klient a server navazovali nové spojení pro každou zprávu.



### 2.2.2 Framework

Framework je sada předpřipraveného kódu a nástrojů, které poskytují strukturu pro vývoj určitého typu aplikace nebo softwaru. [3] Je navržen tak, aby vývojářům ušetřil čas a námahu tím, že jim poskytne základní strukturu a společné funkce a umožní jim soustředit se na vytváření jedinečných funkcí aplikace. Frameworky lze použít pro vývoj frontendů i backendů a k dispozici je mnoho různých typů frameworků.

Při použití frameworku mohou vývojáři využívat předpřipravené moduly a knihovny, což pomáhá zkrátit dobu vývoje, zlepšit udržitelnost kódu a zvýšit celkový výkon a škálovatelnost aplikace. Frameworky také často přicházejí se sadou konvencí, osvědčených postupů a dokumentace, které pomáhají zajistit, aby byl kód uspořádaný a konzistentní, což může usnadnit jeho pochopení a údržbu ostatními vývojáři.

### 2.2.3 Frontend

Vývoj frontendu se týká tvorby uživatelského rozhraní a uživatelského prostředí webových stránek nebo aplikací. Jedná se o část vývoje webových stránek, která se zaměřuje na klientskou stranu neboli část aplikace, která je viditelná pro uživatele a komunikuje s ním. Frontendoví vývojáři používají jazyky jako HTML, CSS a JavaScript k vytvoření vizuálního rozvržení, designu a funkčnosti webových stránek nebo aplikace. Používají také frameworky, jako jsou React, Angular a Vue.js, aby uspořádali a strukturovali svůj kód, který je tak efektivnější a škálovatelnější.

### 2.2.4 Backend

Vývojem backendu se rozumí vytváření serverové části webových stránek nebo aplikací. Jedná se o část vývoje webu, která se zaměřuje na logiku a funkce, jež pohání aplikaci, a je zodpovědná za zpracování a manipulaci s daty a komunikaci s frontendem. Vývojáři backendu používají jazyky jako Java, NodeJS a Go, aby vytvořili logiku a funkčnost aplikace.

Využívají se frameworky jako Express pro NodeJS, nebo Chi pro Go, které usnadňují vývoj backendu a zvyšují jeho efektivitu a škálovatelnost.

### 2.2.5 Databáze

Databáze je systém pro ukládání a správu dat.

### 2.2.6 Cache

Cache neboli mezipaměť je termín, který označuje dočasné uložení dat, aby bylo možné k nim později přistupovat rychleji. Cache může zlepšit výkon aplikace, protože umožňuje ukládat data, která se často používají, do paměti a pak je při dalším použití získat rychleji.

### 2.2.7 Jednotný datový formát JDF

Jednotný datový formát JDF je sada pravidel vydaných v metodickém pokynu č.5 k organizaci celostátního informačního systému o jízdních řádech vydaném Ministerstvem dopravy ČR [4]. JDF udává strukturu souborů a jejich obsah. Je založen na formátu CSV (Comma Separated Values) – záznamově orientovaný formát dat s oddělovači (pole oddělena čárkou, záznamy odděleny středníkem a CRLF). Aktuálně existují tři verze JDF: JDF1.9, JDF 1.10 a JDF 1.11, každá využívaná různými dopravními podniky.

## 2.3 TECHNOLOGIE

### 2.3.1 JavaScript

JavaScript je programovací jazyk [5], který je používán především pro vývoj webových aplikací. Je to jazyk s dynamickým typováním, což znamená, že se datové typy proměňují v závislosti na hodnotách, které jsou jim přiřazeny. JavaScript může být použit pro tvorbu interaktivních webových stránek, včetně těch, které reagují na události, jako jsou kliknutí na tlačítka nebo přetažení objektů. Je také používán pro tvorbu mobilních aplikací a pro spouštění úloh na straně serveru pomocí technologie Node.js. JavaScript je jedním z nejčastěji používaných programovacích jazyků na webu.

### 2.3.2 TypeScript

TypeScript je programovací jazyk [6], který je postaven na JavaScriptu a přidává do něj statické typování. To znamená, že v TypeScriptu je nutné explicitně určit datové typy proměnných a funkcí, což může pomoci při vyhledávání chyb v kódu a umožňuje lepší předvídatelnost a spolehlivost programů. TypeScript je navržen tak, aby podporoval rozšiřitelnost a zjednodušil vývoj velkých aplikací. Jeho kód se kompiluje do JavaScriptu, takže může být použit na všech místech, kde je podporován JavaScript. TypeScript je používán pro vývoj webových aplikací, často spolu s frameworky jako Angular nebo React.js.

### 2.3.3 React.js

React.js je populární knihovna jazyka JavaScript pro vytváření uživatelských rozhraní. Umožňuje vývojářům vytvářet opakovaně použitelné, složitelné komponenty, které lze snadno vykreslovat a aktualizovat v reakci na změny dat nebo interakce s uživatelem. Jednou z klíčových vlastností Reactu je jeho schopnost efektivně aktualizovat uživatelské rozhraní v reakci na změny v podkladových datech, což se děje prostřednictvím procesu zvaného „virtuální rozdělení DOM“. To může vést k výraznému zvýšení výkonu, zejména u rozsáhlých a složitých aplikací. React také poskytuje užitečnou sadu vývojářských nástrojů a vývojářsky přívětivých funkcí, jako je JSX, syntaktické rozšíření jazyka JavaScript, které umožňuje vkládat do kódu jazyka JavaScript prvky podobné HTML. Popularita Reactu v posledních letech rychle roste a je široce používán v mnoha populárních webových stránkách a webových aplikacích, jako jsou Facebook, Instagram, Netflix a Airbnb [7]. React má také velkou a aktivní komunitu vývojářů, kteří přispívají k vývoji knihovny a vytvářejí mnoho knihoven a nástrojů třetích stran, které lze použít k vylepšení možností vývoje Reactu.

### 2.3.4 Leaflet

Leaflet je open source knihovna JavaScriptu [8], která slouží k vytváření interaktivních map. Umožňuje vývojářům snadno vytvářet přizpůsobitelné, výkonnostně nenáročné mapy, které lze vkládat do webových stránek a aplikací. Leaflet poskytuje jednoduché a intuitivní rozhraní API a také širokou škálu modulů pro rozšíření jeho funkčnosti. Podporuje také různé poskytovatele mapových podkladů, například OpenStreetMap a Mapbox, a lze jej integrovat s různými dalšími webovými technologiemi, například Bootstrap a ReactJS. Celkově je Leaflet výkonným nástrojem pro vytváření dynamických a interaktivních map, které mohou zlepšit uživatelské prostředí webových stránek nebo aplikací.

### 2.3.5 Go

Go je programovací jazyk, vyvinutý společností Google [9]. Je to staticky typovaný jazyk, a je navržen tak, aby byl efektivní a snadno použitelný. Go se často používá pro vývoj webových aplikací a nástrojů pro správu a automatizaci systémů. Jeho jednoduchost a rychlost kompilace jsou mezi vývojáři velmi oblíbené.

### 2.3.6 GraphQL

GraphQL [10] je dotazovací jazyk pro rozhraní API, který klientům umožňuje předvídatelným a efektivním způsobem požadovat přesně ta data, která potřebují. Na rozdíl od jazyka REST, který k získání všech dat potřebných pro konkrétní zobrazení nebo funkci vyžaduje několik koncových bodů, jazyk GraphQL umožňuje vývojářům získat všechna potřebná data jediným požadavkem. To může vést k výraznému zvýšení výkonu, zejména na mobilních zařízeních nebo při práci s velkým množstvím dat. Jazyk GraphQL navíc umožňuje větší flexibilitu ve struktuře vrácených dat, protože klient může požadovat přesně ta pole, která potřebuje, a nikoli pevnou sadu polí definovanou rozhraním API. To umožňuje efektivnější využití šířky pásma sítě a může zjednodušit kód klienta.

### 2.3.7 Redis

Redis je open source software, který ukládá data do paměti jako páry klíč-hodnota. Redis lze použít jako databázi, cache nebo zprostředkovatele zpráv. Redis podporuje různé datové struktury, například textové řetězce, seznamy, mapy, množiny a další. Redis je rychlý a škálovatelný pro aplikace v reálném čase, které potřebují nízkou latenci a vysokou propustnost. [11]

### 2.3.8 MongoDB

MongoDB je dokumentový databázový systém, který je navržený tak, aby byl snadno škálovatelný a přizpůsobitelný. Dokumentový systém je typ databázového systému, který ukládá data ve formě dokumentů, které jsou uloženy v kolekcích. Dokumenty jsou zpravidla uloženy ve formátu JSON, což umožňuje ukládat data ve struktuře, která je přímo použitelná pro kód aplikace. Dokumentové databáze jsou vhodné pro ukládání dat, která jsou složitá a nestrukturovaná, jako jsou data zpracovávaná v rámci webových aplikací. MongoDB je jedním z nejpoužívanějších dokumentových databázových systémů a je často používán pro vývoj webových aplikací.

### 2.3.9 Docker

Docker je platforma, která umožňuje vytvářet, spouštět a sdílet aplikace pomocí kontejnerů. Kontejnery jsou izolovaná prostředí, která společně balí kód a závislosti, což usnadňuje nasazení a provoz aplikací na různých platformách. [12]

### 2.3.10 PWA

Progresivní webové aplikace (anglicky Progressive Web Apps, PWA) jsou webové aplikace, které vypadají a načítají se jako běžné webové stránky. Navíc ale nabízí funkce běžně dostupné pouze nativním aplikacím, např. práci offline, push notifikace nebo přístup k hardwaru zařízení. PWA tak kombinují flexibilitu webu s možnostmi aplikací. [13]

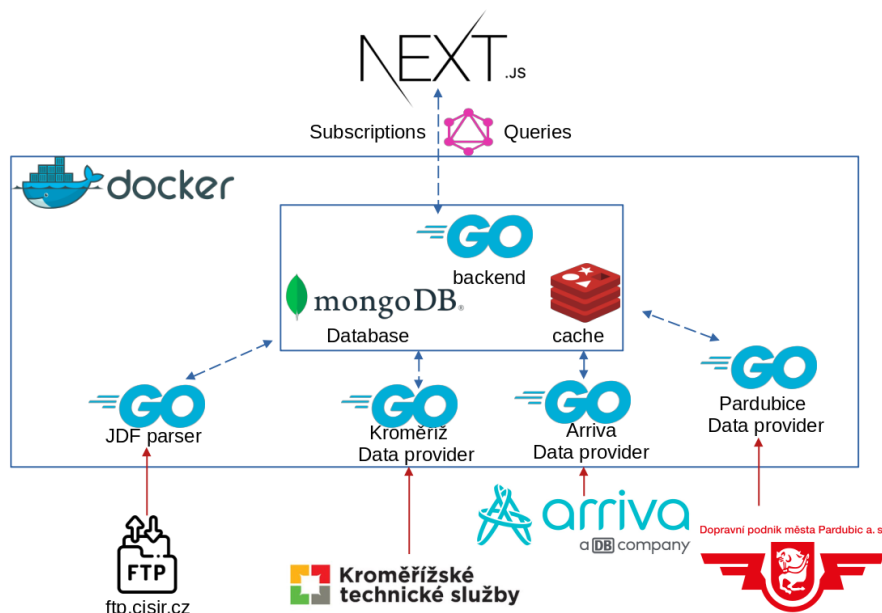
## 3 REALIZACE

Platforma má za účel vytvořit pro dopravní podniky jednoduchý způsob sdílení dat o polohách vozidel. Dopravní podniky většinou interně disponují těmito daty, využívají se na dispečincích, či k sledování zpoždění na informačních tabulích na jednotlivých zastávkách.

Služba nabízí pro dopravní podniky další způsob komunikace o aktuálním dění se svými cestujícími. Každý dopravce má možnost za pomoci webového portálu přidat provozní upozornění přímo do aplikace.

### 3.1 ARCHITEKTURA

Architektura platformy je tvořena z několika částí, které jsou navzájem propojeny. Uživatel interaguje s webovou aplikací, která komunikuje s backendem. Backend je tvořen z několika služeb, které komunikují s datovými zdroji dopravců.



Obrázek 3.1: Architektura aplikace

Tento projekt je rozdělen do dvou hlavních částí. První částí je frontend, vytvořený pomocí frameworku NextJS v programovacím jazyce TypeScript. Druhou částí je backend, který je napsán v programovacím jazyce GO.

## 3.2 BACKEND

Backendová část projektu má za úkol získat data od jednotlivých dopravních podniků, získat a zpracovat data o jízdách [15] a za pomoci GraphQL je zprostředkovat frontendové části.

### 3.2.1 Využité technologie

Backendová část projektu je napsaná v programovacím jazyce GO. Routování zajišťuje jednoduchá knihovna Chi-go. Hlavním endpointem celé aplikace je `"/query"`, za který odpovídá GraphQL, jenž se stará o dotazy a subscriptions. Díky využití GraphQL subscriptions je frontend schopen zobrazovat data v téměř reálném čase. Jako hlavní databázový systém byla využita databáze MongoDB 2.3.8, díky své flexibilitě a rychlé škálovatelnosti. Pro urychlení často-se opakujících operací, aplikace využívá Redis 2.3.7, rychle dostupnou cache.

### 3.2.2 Služby

Samotný backend využívá návrhovou architekturu mikroslužeb (microservice). Je rozdělen na několik menších částí, které mají za úkol různé operace. Podle jejich funkce je můžeme rozdelit do následujících kategorií:

- Microservice zabývající se aktualizací a zpracováním dat pro strojové čtení jízdých řádů. K jejímu spuštění dochází pravidelně každý den, pro zajištění aktuálních dat.
- Mikroservice zajišťující komunikaci s dopravci, jejich cílem je načítání dat o polohách vozidel v reálném čase. Data však bývají nejednotná a je třeba individuální implementace pro každý dopravní podnik.
- Služba obstarávající samotnou komunikaci s frontendovou částí. Pro aplikaci zajišťuje dostupnost správných dat.

### 3.2.3 Strojové čtení jízdnic řádů

Podle vyhlášky [14] má každý dopravní podnik povinnost zveřejňovat jízdnic řády svých spojů ve formátu, vhodném pro strojové čtení. Tyto data dopravní podniky publikují prostřednictvím Celostátního informačního systému o jízdnicích řádech (CISJŘ) [15]. Tento systém spravuje firma CHAPS spol. s r.o. [16]. Výstupem CISJŘ je veřejně přístupný ftp server na adrese 'ftp://ftp.cisjr.cz'. Data jsou zveřejňována ve formátu JDF 2.2.7 a jsou zázipována.

Pro účel aktualizace a zpracování těchto dat slouží samostatná microservice „*JDF Parser*“. Ze struktury a dat těchto souborů microservice vytvoří v databázovém systému MongoDB dokumenty jednotlivých spojení veřejné dopravy. Tento dokument obsahuje informace o příjezdech a odjezdech ze zastávek, jejich pevné kódy, typ vozidla a další. Jednotlivé spojení veřejné dopravy jsou jednoznačně identifikovatelná za pomoci kombinace čísla linky a čísla spoje.

### 3.2.4 Data dopravních podniků

Dopravní podniky zpravidla využívají data o aktuální poloze vozidel pouze pro interní použití na dopravních dispečincích. Tato data nejsou nijak upravována nařízenými nebo vyhláškami, což znamená že nejsou sjednocená pravidla mezi podniky, jak by data měla vypadat.

Z tohoto faktu vznikla nutnost implementace individuální microservice pro každý dopravní podnik. Úkolem těchto microservis je získání dat o aktuální poloze vozidel a jejich zpracování do jednotného formátu pro použití. Data jsou poté ukládána do datového úložiště Redis 2.3.7, které funguje jako cache 2.2.6, odkud je hlavní microservice čte.

Informace o poloze spoje jsou ukládána v kontextu celé trasy, což umožňuje cestujícím vidět přesnou trasu spoje, včetně informací o zastávkách.

### 3.2.5 Komunikace s frontendem

Poslední microservice má za úkol poskytovat všechna získaná data frontendové části. K tomuto úkolu bylo využito rozhraní GraphQL. Správné využití GraphQL zabraňuje přebytknému stahování velkého množství dat a pomáhá ke zjednodušení komunikace. [17]

Hlavními informacemi, které backend poskytuje frontendové části, jsou aktuální polohy vozidel. K tomuto účelu slouží subscription „*Get Vehicles*“, která každých 10 sekund pošle nová data.



Pro získání dodatečných informací o spoji slouží query “*GetConnectionDetail*“ a “*GetRoads*“, které potřebují dva argumenty, číslo linky “*line\_name*“ a číslo spoje “*connection\_number*“ k jedinečné identifikaci spoje.

K získání budoucích odjezdů ze zastávek slouží query „*GetStationConnections*“, která potřebuje jako argument “*IDzastávky*“ a vrací zpět spoje které v příštích 45 minutách projedou danou zastávkou.

### 3.2.6 Nasazení na server

Platforma je nasazena za pomoci kontejnerizace. Kontejnerizace nám zaručuje identické prostředí, které obsahuje pouze nezbytná data k běhu jednotlivých microservice.

Pro vytvoření kontejnerů se využívá Docker. Proces vytváření jednotlivých kontejnerů zahrnuje použití souboru „*Dockerfile*“, který přesně popisuje jednotlivé kroky k sestavení.

K následnému nasazení je využíván *docker-compose*, který umožňuje definovat služby (jednotlivé microservice, mongoDB, redis) a komunikaci mezi službami. Pro definování služeb se využívá soubor „*docker-compose.yaml*“, který poté dovoluje spravovat všechny kontejnery pomocí jediného příkazu.

## 3.3 WEBOVÁ APLIKACE

Frontendová část platformy, ve formě webové aplikace a aplikace mobilní, je hlavní částí platformy, která živá data o pohybu vozidel komunikuje k cestujícím.

Využívá přitom komunikace s Backendem za pomoci REST + GraphQL API, vše za využití protokolů HTTPS nebo WebSocket.

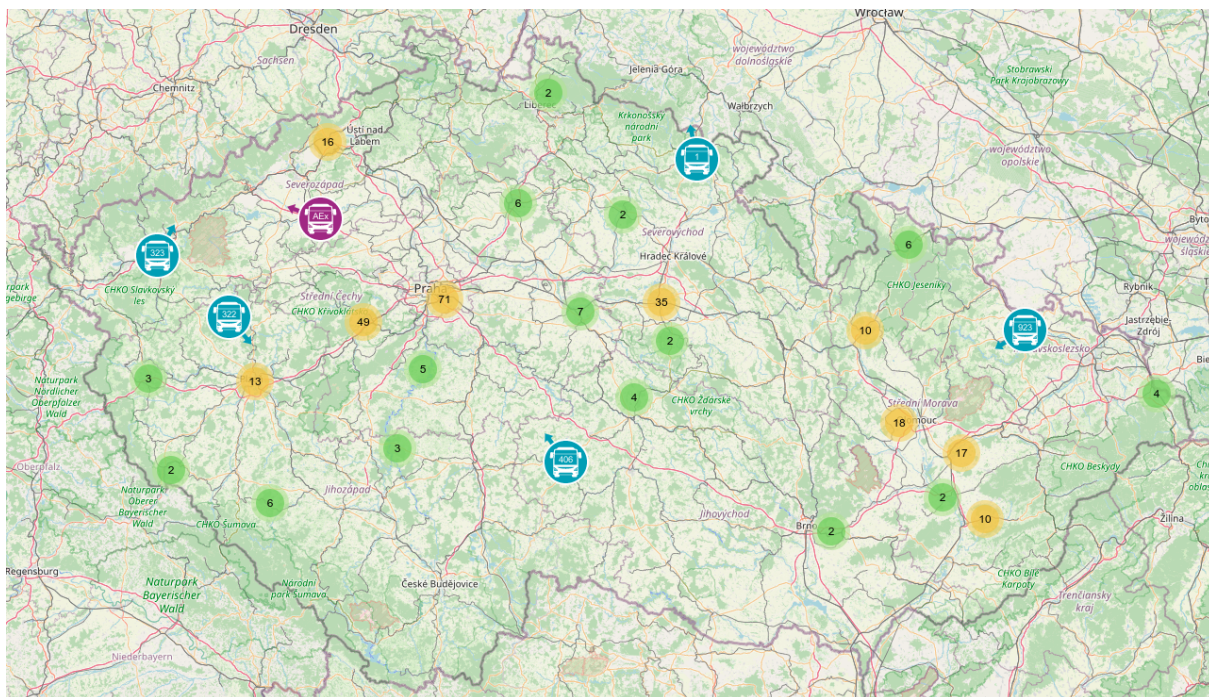
Webová aplikace využívá technologie NextJS pro statické generování aplikace na straně serveru (pro podporu SEO). Webová aplikace využívá za svého chodu celou řadou doplňujících dotazů přes internet pro dosažení interaktivity, takzvaných XHR requestů.

Vývojový model webové aplikace dopovídá programovacímu paradigma jednostránkové aplikace. [18]

### 3.3.1 Mapa

Na jednotné mapě se zobrazují všichni dopravci. Mapu lze ovládat na mobilních zařízeních pohyby prsty, nebo na stolních počítačích myší. Podle uživatelského pohledu aplikace vybere dopravce, které by měla zobrazovat, čímž redukuje objem přenášených dat. Do zařízení uživatele jsou přenášeny pouze data o vozidlech, které uživatel sleduje.

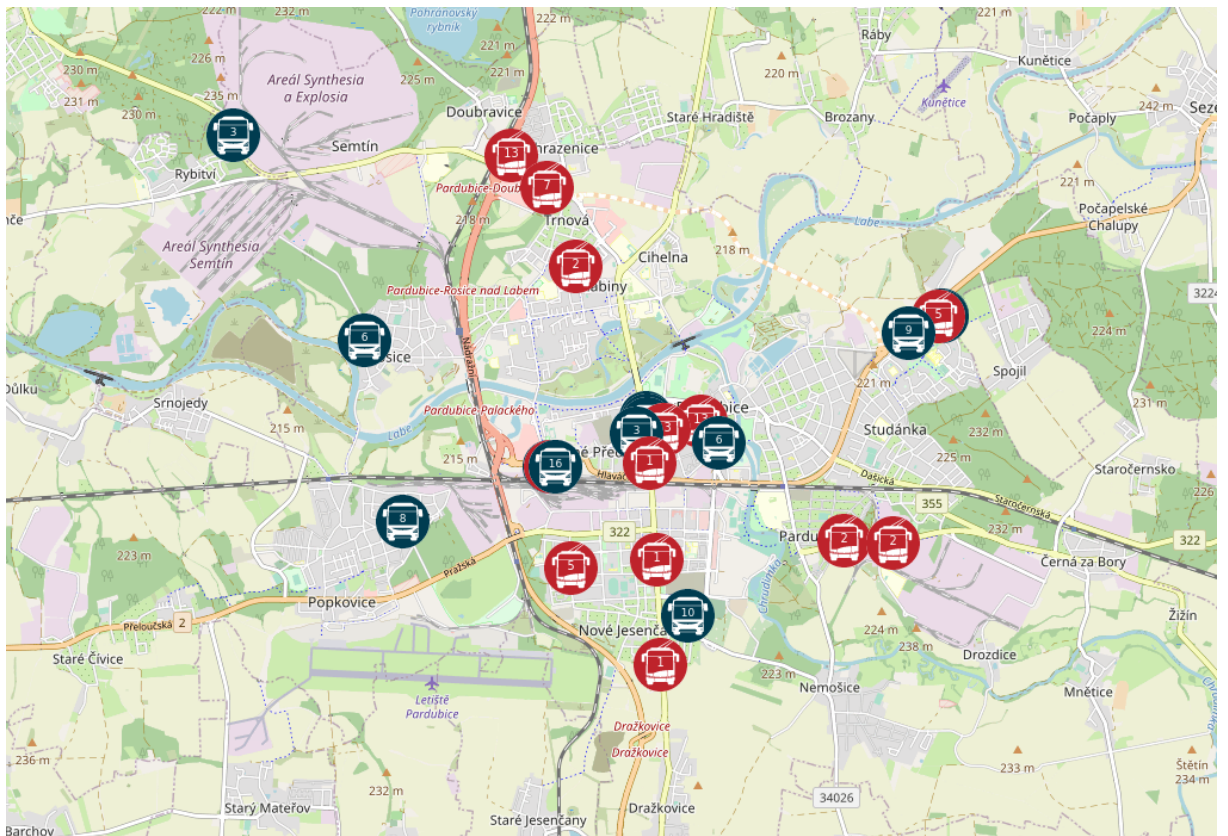
Pro přehlednost mapa implementuje funkci shlukování vozidel do bublin (viz obr. 3.2). Shlukování do bublin zajistí v aplikaci přehlednost a plní i funkci optimalizace výkonu mapy. Při vzdáleném pohledu je funkce plynulé animace pohybu vozidel vypnuta.



Obrázek 3.2: Ukázka shlukování vozidel na mapě

### 3.3.2 Zobrazování vozidel

K živému zobrazování dat se využívá GraphQL subscription „*GetVehicles*“, díky které z backendu automaticky přijde zpráva s aktuálními daty. Tato data však mívají rozestup 10 sekund, proto se animace plynulého přechodu dopočítává. Jednotlivá data vozidel jsou rozdělena podle dopravců, subscription se tedy upravuje podle toho, kam se uživatel zrovna dívá.



Obrázek 3.3: Ukázka části mapy

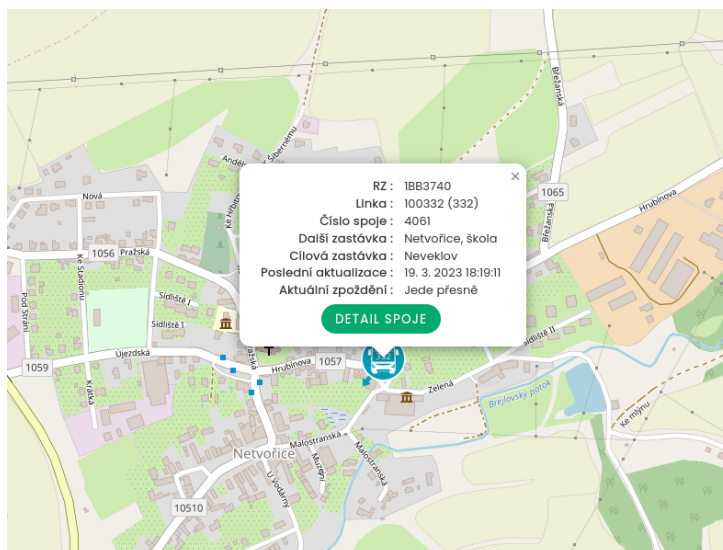
Pro dosažení plynulosti zobrazování využívá aplikace předepsaných praktik pro práci s frameworkem ReactJS [19]. Časování animace přenechává aplikace webovému prohlížeči uživatele, využívá totiž Webového rozhraní `Window.requestAnimationFrame()` [20], které zajišťuje vytváření nových snímků ve vhodné frekvenci.

### 3.3.3 Cache

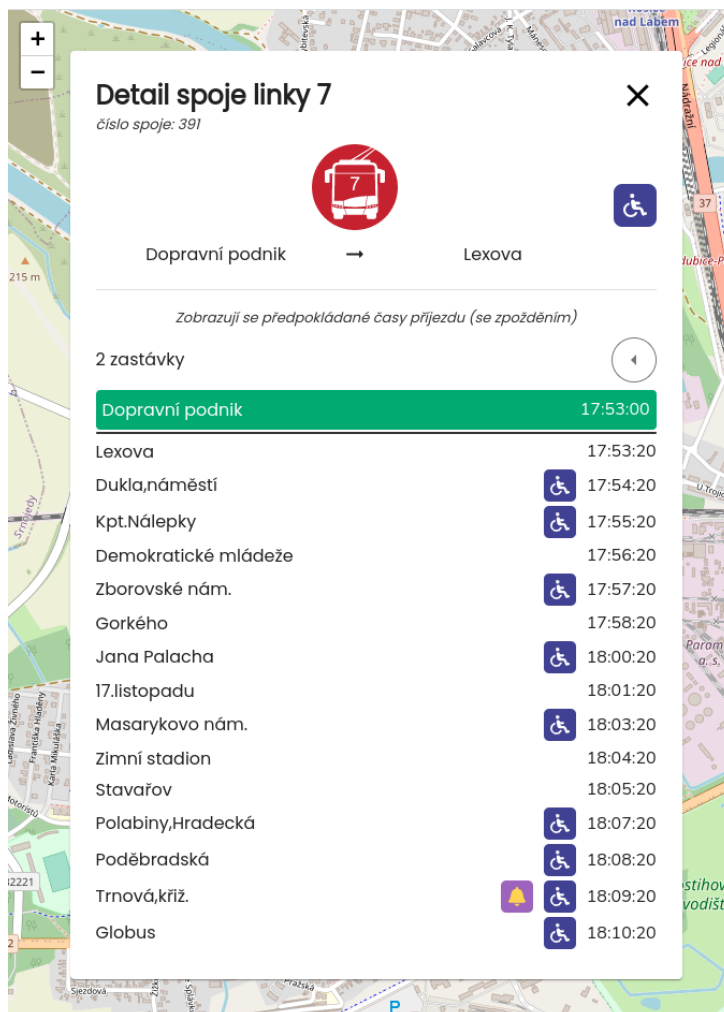
Informace o detailech spojení jsou na straně cestujícího ukládána do cache, aby již nebylo třeba tato data opakovaně načítat. Cache 2.2.6 na straně webové aplikace šetří uživatelům jejich datové připojení a zároveň snižují nároky na vytěžování serveru.

### 3.3.4 Detail spojení

Uživatel má po kliknutí na jedoucí spoj k dispozici základní informace o tomto spoji, např. číslo linky, zpoždění, další zastávka, nebo cílová zastávka (viz obr.3.4). Toto vše je vždy individuálně upravitelné pro každý dopravní podnik. Po rozkliknutí bližšího detailu o spojení se uživateli zobrazí dodatečné informace (viz obr.3.5).



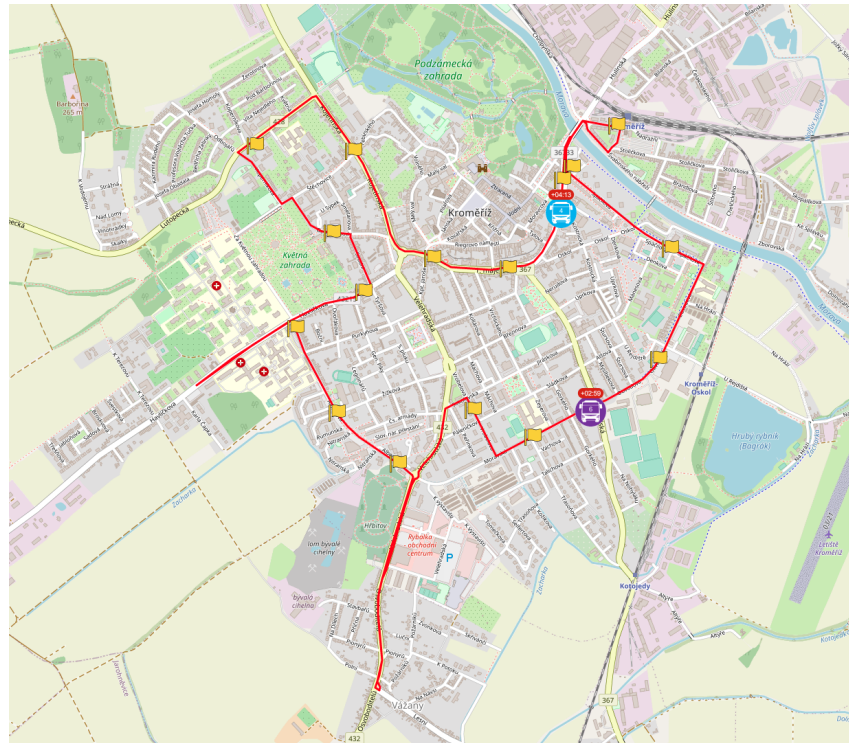
Obrázek 3.4: Detail linky 332



Obrázek 3.5: Rozšířený detail linky 7



Současně se zobrazením bližších informací o spoji, se zobrazuje také grafická reprezentace trasy, kudy daný spoj pojede. V tomto zobrazení je také možné vidět všechny zastávky na trase.



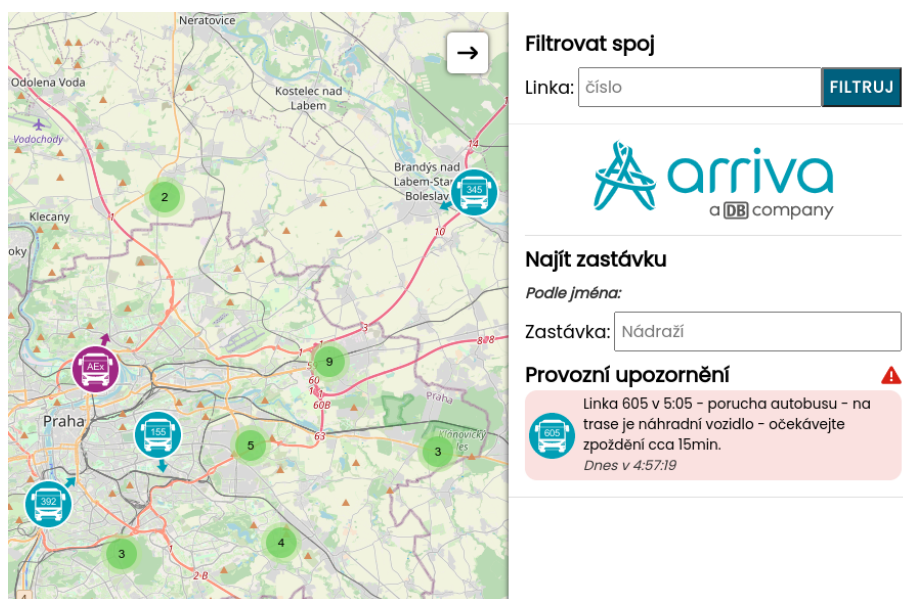
Obrázek 3.6: Detail trasy v Kroměříži

### 3.3.5 Provozní upozornění

Dopravci mohou přidávat do aplikace klienta provozní upozornění, které se zobrazí v případě, že se vyskytne nějaká situace, která může mít vliv na cestu cestujícího.

Za tímto účelem vzniklo uživatelské prostředí pro dispečink dopravce.

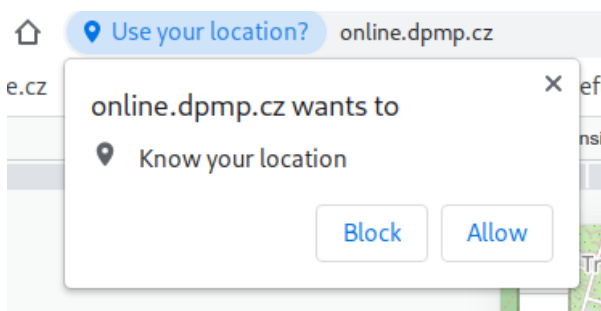
Aplikace pravidelně zjistí aktuální stav a zobrazí nejnovější informace.



Obrázek 3.7: Ukázka provozního upozornění

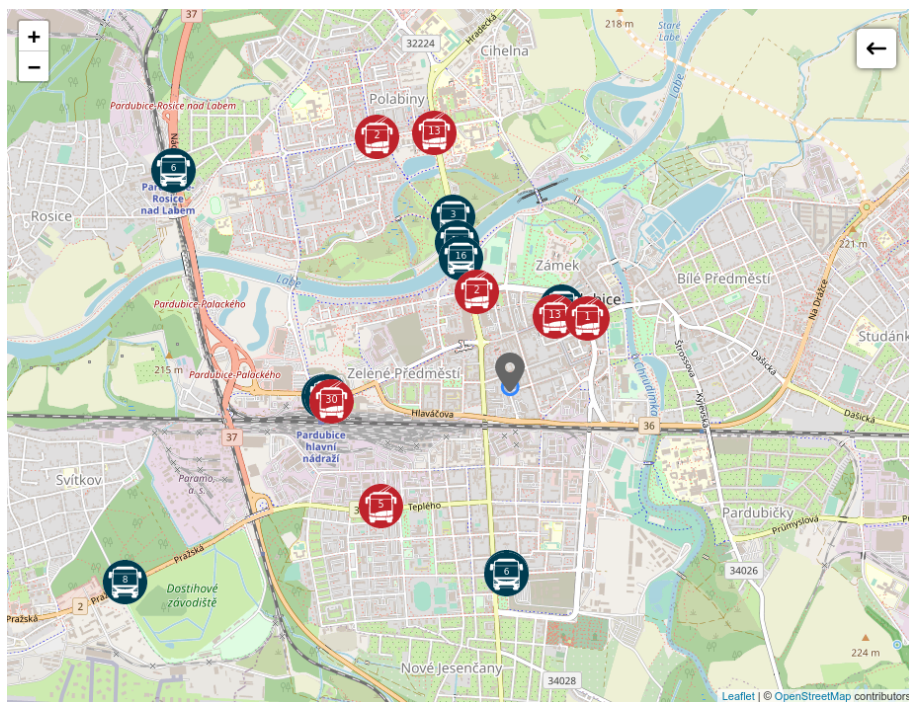
### 3.3.6 Práce s polohou uživatele

Aplikace využívá webového rozhraní geolokace pro získání aktuální polohy uživatele. V tomto kroku se aplikace řídí webovou mechanikou Permissions API [28]. Proces zahrnuje požádání uživatele polohu využít. (viz obr.3.8)



Obrázek 3.8: Žádost o povolení

S uděleným oprávněním může aplikace pro cestujícího zobrazit zastávky v jeho blízkosti.



Obrázek 3.9: Ukázka polohy uživatele



### 3.3.7 Monitorování návštěvnosti

Zajímavou funkcí pro dopravce je monitorování návštěvnosti aplikace za pomoci Google Analytics. Za běhu aplikace jsou při různých událostech (např. návštěva stránky) odesílány informace o době návštěvy uživatele do monitorovacího panelu. [21]

Pro získávání přesných dat o užitečných a oblíbených funkcích aplikace využíváme monitorování vlastních událostí, např. při otevření detailu spoje, nebo vyhledání detailu zastávky.

### 3.3.8 Aplikace a web

Aplikace využívá experimentálních funkcí technologie PWA 2.3.10 jako je například načítání GPS souřadnic uživatele, zobrazování notifikací, nastavení připomínky na příjíždějící spoj. Funkce PWA zároveň nabízí stahování map do zařízení, aby nemusely být při každém spuštění znovu stahovány.

Díky technologii PWA je webové aplikaci umožněno využít vnitřního uložení zařízení pro ukládání dat o mapách, a tím je možné ušetřit uživatelům další stahování a prodlevu před možností využívání aplikace.

## 4 BEZPEČNOST

Webové aplikace jsou v dnešní době rozšířeny do různých odvětví, včetně školství, zdravotnictví a samozřejmě hojně i do běžného života. Toto rozšíření však s sebou nese i rizika kybernetických hrozeb a útoků. Kyberzločinci neustále hledají zranitelná místa ve webových aplikacích, a proto je zabezpečení webových aplikací nezbytnou součástí každého procesu vývoje webových aplikací.

Bezpečností webových aplikací se zabývá projekt OWASP [22](Open Web Application Security Project) se svým seznamem Top 10 [23] nejzávažnějších bezpečnostních rizik webových aplikací. Tento seznam je pravidelně aktualizován, aby odrážel nejnovější trendy v oblasti bezpečnostních hrozeb pro webové aplikace.

Tento projekt se snaží o řešení těchto bodů. Zde jsou některá navržená řešení.

- **Kryptografické selhání** [24]: Backendová i frontendová část projektu využívají komunikace pomocí šifrovaného protokolu HTTPS. Certifikáty jsou podepsané certifikačními autoritami Let's Encrypt[25] a Google Trusted Services.
- **Injection** [26]: GraphQL se stará o validitu datových typů při komunikaci mezi frontendem a backendem. Při práci s uživatelským vstupem probíhá důležitá sanitizace, aby nebylo možné vytvořit NoSQL inject.
- **Magic Link Login** [27]: Uživatelé se nepřihlašují pomocí obvyklé kombinace jméno - heslo. Při přihlášení zadávají pouze svůj email. Na tento email je jim následně odeslán časově omezený klíč pro jednorázové přihlášení. Dochází k eliminaci hrozby uniknutí hesla, jelikož není nikde uloženo.

## 5 ZÁVĚR

Závěrem lze říci, že jsme úspěšně vyvinuli platformu, která umožňuje dopravním podnikům měst sdílet data o poloze v reálném čase. Díky využití veřejně dostupných dat jízdních řádů, se podařilo vytvořit spolehlivé a efektivní řešení pro sdílení údajů o poloze a o informacích o vozidlech veřejné dopravy.

Aktuálně je platforma dostupná pro dopravní podniky města Pardubice a.s. (na adrese <https://online.dpmp.cz>), a pro technické služby Kroměříž s.r.o (na adrese <https://kromeriz.mhdonline.cz/>).

V době psaní této dokumentace probíhá integrace nového dopravce Arriva do aplikace (na adrese <https://global.mhdonline.cz/>)

Plánem do budoucna je rozšiřovat dostupnost služby i pro další dopravní podniky v rámci České republiky.

Kromě toho byla platforma navržena s ohledem na škálovatelnost a rozšiřitelnost, což umožňuje budoucí růst a rozšíření. Celkově tato práce nejen splnila své cíle, ale také prokázala potenciál této platformy výrazně zlepšit komunikaci a spolupráci s dopravními podniky.

## LITERATURA

- [1] *The WebSocket Protocol* [online]. [cit. 2023-03-14]. Dostupné z <https://www.rfc-editor.org/rfc/rfc6455>
- [2] *Difference between HTTP and WebSocket (HTTP 2.0 )* [online]. <https://developerinsider.co/> [cit. 2023-03-19]. Dostupné z <https://developerinsider.co/difference-between-http-and-http-2-0-websocket/>
- [3] *Framework* [online]. [cit. 2023-03-10]. Dostupné z <https://codeinstitute.net/global/blog/what-is-a-framework/>
- [4] ČESKÁ REPUBLIKA. METODICKÝ POKYN Č.5 K ORGANIZACI CELOSTÁTNÍHO INFORMAČNÍHO SYSTÉMU O JÍZDNÍCH ŘÁDECH. In: . Ministerstvo dopravy – Odbor veřejné dopravy, 2014, ročník 2014, 11/2014-190-CIS/6. Dostupné také z [https://www.mdcr.cz/getattachment/Dokumenty/Verejna-doprava/Jizdni-rady,-kalendare-pro-jizdni-rady,-metodi-\(1\)/Jizdni-rady-verejne-dopravy/metodicky-pokyn-cis-5.pdf.aspx](https://www.mdcr.cz/getattachment/Dokumenty/Verejna-doprava/Jizdni-rady,-kalendare-pro-jizdni-rady,-metodi-(1)/Jizdni-rady-verejne-dopravy/metodicky-pokyn-cis-5.pdf.aspx)
- [5] *JavaScript* [online]. mozilla.org [cit. 2023-03-19]. Dostupné z <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [6] *TypeScript* [online]. <https://www.typescriptlang.org/> [cit. 2023-03-19]. Dostupné z <https://www.typescriptlang.org/>
- [7] *The Top Companies That Are Using React Js Services* [online]. [cit. 2023-03-19]. Dostupné z <https://www.bigscal.com/blogs/frontend-technology/top-companies-using-react-js-services-to-their-best/>
- [8] *Leaflet* [online]. [cit. 2023-01-25]. Dostupné z <https://leafletjs.com/>
- [9] *Go* [online]. [cit. 2023-02-19]. Dostupné z <https://go.dev/>
- [10] *GraphQL* [online]. [cit. 2023-03-15]. Dostupné z <https://graphql.org/>
- [11] *Redis* [online]. [cit. 2023-03-19]
- [12] *Docker* [online]. [cit. 2023-03-04]. Dostupné z <https://www.docker.com/>

- [13] *Wikipedie: Otevřená encyklopedie: Progresivní webové aplikace* [online]. c2023 [citováno 19. 03. 2023]. Dostupný z WWW: [https://cs.wikipedia.org/w/index.php?title=Progresivn%C3%AD\\_webov%C3%A9\\_aplikace&oldid=22399924](https://cs.wikipedia.org/w/index.php?title=Progresivn%C3%AD_webov%C3%A9_aplikace&oldid=22399924)
- [14] ČESKÁ REPUBLIKA. *VYHLÁŠKA ze dne 23. června 2014 o jízdách veřejné linkové dopravy*. In: Sběrka zákonů. 2014, částka 52, číslo 122. Dostupné také z <http://aplikace.mvcr.cz/sbirka-zakonu/ViewFile.aspx?type=z&id=27158>
- [15] *Celostátní informační systém o jízdách CIS JŘ* [online]. [cit. 2023-03-12]. Dostupné z <https://portal.cisjr.cz/>
- [16] *CHAPS: CIS JŘ* [online]. [cit. 2023-03-13]. Dostupné z <https://www.chaps.cz/cs/products/CIS>
- [17] *GraphQL is the better REST* [online]. [cit. 2023-03-26]. Dostupné z <https://www.howtographql.com/basics/1-graphql-is-the-better-rest/>
- [18] *HUSP/* [online]. [cit. 2023-03-23]. Dostupné z <https://huspi.com/blog-open/definitive-guide-to-spa-why-do-we-need-single-page-applications/>
- [19] *React JS Documentation* [online]. [cit. 2023-03-23]. Dostupné z <https://react.dev/learn/rendering-lists>
- [20] *MDN Web Docs* [online]. [cit. 2023-03-19]. Dostupné z <https://developer.mozilla.org/en-US/docs/Web/API/window/requestAnimationFrame>
- [21] *Google Cloud Docs* [online]. [cit. 2023-03-24]. Dostupné z <https://cloud.google.com/appengine/docs/flexible/integrating-with-analytics?tab=node.js#top>
- [22] *OWASP* [online]. [cit. 2023-03-19]. Dostupné z <https://owasp.org/>
- [23] *OWASP Top 10* [online]. [cit. 2023-03-19]. Dostupné z <https://owasp.org/Top10>
- [24] *Kryptografické selhání* [online]. [cit. 2023-03-19]. Dostupné z [https://owasp.org/Top10/A02\\_2021-Cryptographic\\_Failures/](https://owasp.org/Top10/A02_2021-Cryptographic_Failures/)
- [25] *Let's Encrypt* [online]. [cit. 2023-03-19]. Dostupné z <https://letsencrypt.org/>
- [26] *Testing for NoSQL Injection* [online]. [cit. 2023-03-24]. Dostupné z [https://owasp.org/www-project-web-security-testing-guide/latest/4-Web\\_Application\\_Security\\_Testing/07-Input\\_Validation\\_Testing/05.6-Testing\\_for\\_NoSQL\\_Injection](https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/07-Input_Validation_Testing/05.6-Testing_for_NoSQL_Injection)
- [27] *WorkOS* [online]. [cit. 2023-03-24]. Dostupné z <https://workos.com/blog/a-guide-to-magic-links>

- [28] *MDN Web Docs* [online]. [cit. 2023-03-25]. Dostupné z [https://developer.mozilla.org/en-US/docs/Web/API/Permissions\\_API/Using\\_the\\_Permissions\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Permissions_API/Using_the_Permissions_API)

## SEZNAM OBRÁZKŮ

3.1	Architektura aplikace . . . . .	9
3.2	Ukázka shlukování vozidel na mapě . . . . .	14
3.3	Ukázka části mapy . . . . .	15
3.4	Detail linky 332 . . . . .	16
3.5	Rozšířený detail linky 7 . . . . .	16
3.6	Detail trasy v Kroměříži . . . . .	17
3.7	Ukázka provozního upozornění . . . . .	18
3.8	Žádost o povolení . . . . .	18
3.9	Ukázka polohy uživatele . . . . .	19