VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

SYSTÉM PRO ZÍSKÁVÁNÍ A VIZUALIZACI DAT O VOZIDLE ZE SBĚRNICE OBD

SYSTEM FOR ACQUIRING AND VISUALIZING VEHICLE DATA FROM THE OBD BUS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JAN KROUTIL**
AUTHOR

VEDOUCÍ PRÁCE **Ing. PETR JOHN**
SUPERVISOR

BRNO 2025

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

161067

Ústav:	Ústav informačních systémů (UIFS)
Student:	**Kroutil Jan**
Program:	Informační technologie
Název:	**Systém pro získávání a vizualizaci dat o vozidle ze sběrnice OBD**
Kategorie:	Webové aplikace
Akademický rok:	2024/25

Zadání:

1. Prostudujte oblast internetu věcí (*Internet of Things, IoT*), prozkoumejte sběrnici On-Board Diagnostics (OBD). Zaměřte se primárně na verzi OBD2.
2. Prostudujte principy vizualizace dat na obrazovkách typu *dashboard* a klíčové ukazatele výkonu (*Key Performance Indicators, KPI*).
3. Analyzujte požadavky na systém agregující data – nejdříve na úrovni vozidla, a poté v centrálním systému.
4. Na základě analýzy navrhněte systém pro agregování dat na úrovni vozidla a jeho integraci do existujícího centrálního systému.
5. Navržené řešení implementuje a integrujte.
6. Proveďte zhodnocení zvoleného přístupu.

Literatura:

- Greengard, S.: The Internet of Things. MIT Press, 2015, ISBN 978-026-2527-736.
- JOHN, P. Optimising processes in IoT. Brno, 2024. PhD thesis proposal. Brno University of Technology, Faculty of Information Technology. Supervisor prof. Ing. Tomáš Hruška, CSc.
- Few, S.: Information Dashboard Design: The Effective Visual Communication of Data. Sebastopol, USA: O'Reilly, 2006, ISBN 978-059-6100-16

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:	**John Petr, Ing.**
Vedoucí ústavu:	Kolář Dušan, doc. Dr. Ing.
Datum zadání:	1.11.2024
Termín pro odevzdání:	14.5.2025
Datum schválení:	11.10.2024

Abstrakt

V dnešní době má již takřka každé auto OBD2 port. Tento port slouží k načtení dat z jednotky automobilu do chytrého telefonu či počítače a následně je možné tato data použít při diagnostice vozidla. Stávající diagnostické programy však umožňují zobrazit pouze aktuální data a většinou pouze z jednoho vozu, právě kontrolovaného. Mnoha servisům a jiným subjektům spravujícím flotily vozidel tak nezbývá nic jiného než manuálně načíst data z každého vozidla, což zabere nějaký čas a stojí peníze. Tato práce se zabývá zkoumáním použití dashboardů pro správnou vizualizaci dat z vozidla. Řeší korektní a přehledné zobrazení relevantních dat na jednom místě pro rychlé použití uživatelem. Dále řeší komunikaci s OBD2 portem a následný přesun a archivaci dat do uložiště a nakonec i samotné odeslání dat na server a zobrazení archivovaných dat přehledně pro koncového uživatele, který si je může zobrazit historicky pro jakékoliv spravované vozidlo.

Abstract

Nowadays, almost every car has an OBD2 port. This port is used to read data from the vehicle's unit to a smartphone or computer, and these data can then be used for vehicle diagnostics. However, existing diagnostic programs typically only display current data, usually from a single vehicle, the one being checked. Many service providers and other entities managing vehicle fleets have no choice but to manually read data from each vehicle, which takes time and costs money. This work focuses on exploring the use of dashboards for the proper visualization of vehicle data. It addresses the correct and clear display of relevant data in one place for quick use by the user. It also deals with communication with the OBD2 port and the subsequent transfer and archiving of data to storage and finally, it covers the sending of data to a server and the display of archived data in an easily understandable way for the end user, who can view it historically for any managed vehicle.

Klíčová slova

internet věcí, OBD2 konektor, dashboard, vizualizace dat, RIoT, Quasar Framework, webová aplikace

Keywords

internet of things, OBD2 port, dashboard, data visualization, RIoT, Quasar Framework, web app

Citace

KROUTIL, Jan. *Systém pro získávání a vizualizaci dat o vozidle ze sběrnice OBD*. Brno, 2025. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Petr John

Systém pro získávání a vizualizaci dat o vozidle ze sběrnice OBD

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Petra Johna. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Jan Kroutil
11. května 2025

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce panu Ing. Petru Johnovi za odborné vedení, cenné rady a vstřícnost při konzultacích během celé doby zpracování práce.

Obsah

1 Úvod **5**

2 Internet věcí **7**
2.1 Historie IoT 7
2.2 Inteligentní objekty (zařízení) 8
2.3 Komunikace v IoT 10
2.4 Architektura IoT 12
2.5 Praktické využití IoT 13
2.6 OBD (On-Board Diagnostics) 15

3 Dashboard **18**
3.1 Kategorie 18
3.2 Vizuální vnímání 19
3.3 Prvky dashboardu 22
3.4 Správný návrh dashboardu 26
3.5 Klíčové ukazatele výkonnosti (KPIs) 27

4 Analýza **29**
4.1 Cílová skupina 29
4.2 Proces 29
4.3 Požadavky 30
4.4 Existující řešení 30
4.5 Využití RIoT 35

5 Návrh **37**
5.1 Návrh na úrovni vozidla 38
5.2 Návrh uživatelského rozhraní 39

6 Implementace **46**
6.1 Implementace na úrovni vozidla 46
6.2 Implementace uživatelské aplikace 48

7 Zhodnocení přístupu **56**
7.1 Změny v průběhu vývoje 56
7.2 Demonstrace 58
7.3 Potenciální rozšíření 60

8 Závěr **61**

Literatura **62**

A Obsah externí přílohy **65**

B Testovací scénáře **66**

Seznam obrázků

2.1 Shrnutí šesti úrovní automatizace vozidel 15
2.2 OSI (Open Systems Interface) model 17
2.3 OBD II uzel 17

3.1 Vizualizace šesti barev, nahoře pozorováno člověkem bez zrakové vady a dole člověkem, který trpí deuteranopií. Převzato z [31]. 20
3.2 Srovnání klasického zpracování (vlevo) a předpozornostního (vpravo) 21
3.3 Příklady tvarové (gestalt) psychologie. Převzato z [22]. 22
3.4 Krabicový graf (vlevo) a Sparkline (vpravo) 24
3.5 Malé násobky (small multiples)[1] 26
3.6 Příklad vizualizace vyhodnocení KPI z platformy ACADA (Logimic [26]), převzato z bakalářské práce [7] 28

4.1 Příklad vizualizací dat Motopower MP69033 31
4.2 Zařízení Xentry Diagnosis Lite Pad 32
4.3 Autocom CARS software 33
4.4 Monitorovací služby firmy Commander 34
4.5 Car Scanner ELM OBD II 35

5.1 Schéma systému pro zpracování a vizualizaci dat z OBD II 37
5.2 Flow diagram cyklu získávání a zpracování dat z OBD 39
5.3 Návrh obrazovky dashboardu 41
5.4 Návrh obrazovky výběru cesty 42
5.5 Návrh okna editace/vytvoření skupiny parametrů 43
5.6 Návrh obrazovky nastavení 43
5.7 Příklad spojitého grafu s mezními hodnotami 44
5.8 Příklad bullet grafu 45

6.1 Příklad komunikace s využitím *python-OBD* 47
6.2 Struktura dat odesílaných na server pres MQTT 48
6.3 Dotaz na senzor pomocí GraphQL query 50
6.4 Vzhled implementované stránky výběru vozidla 50
6.5 Vzhled implementované stránky výběru trasy 51
6.6 Vzhled implementované stránky dashboardu 51
6.7 Zabarvení trasy na barvu prahu 52
6.8 Spojitý graf vlevo a bullet graf vpravo 53
6.9 Okno pro vytvoření a editaci skupin grafů 54
6.10 Okno pro editaci konfigurace grafu 54

7.1 Graf před možností editace (vlevo) a po přidání možnosti editace (vpravo) . 57

Kapitola 1

Úvod

V dnešní době digitálních technologií a internetu věcí (IoT) se stále více zařízení propojuje do rozsáhlých sítí, které umožňují jejich vzájemnou komunikaci a řízení na dálku. Tento rychlý rozvoj přináší nejen větší efektivitu a automatizaci procesů, ale také nové možnosti v oblastech, jako je průmysl, zdravotnictví nebo chytré domácnosti. Jedním z těchto zařízení jsou i moderní vozidla, která jsou vybavena komplexními řídicími jednotkami a komunikačními sběrnicemi, umožňujícími získávat rozsáhlé množství dat o jejich stavu a provozu. Tato data lze získat přes OBD (On-Board Diagnostics) rozhraní. Takřka většina vozidel v dnešní době takové rozhraní má. Toto rozhraní poskytuje přístup k diagnostickým informacím, jako jsou údaje o motoru, emise a další klíčové parametry.

Existuje mnoho programů pro načítání a zobrazení těchto informací, od drahých profesionálních nástrojů až po levné alternativy či dokonce bezplatná řešení. Většina ale má něco společného, tyto programy často nejsou moc uživatelsky přívětivé a nezkušený uživatel z nich dostane potřebné informace hodně těžko a složitě. Navíc, kromě několika výjimek, umožňují pouze čtení dat v reálném čase a neumožňují ukládání ani zaznamenávání hodnot pro zpětnou analýzu. To tedy znamená, že například mechanik, který spravuje flotilu 50 automobilů, musí při kontrole všech vozidel osobně vstoupit do kabiny každého z nich a data načíst přes datový kabel či bezdrátové připojení do svého počítače. V horším případě se při výskytu chyby k vozidlu musí nějakým způsobem dostat a tedy ztrácí peníze i svůj čas. Funkce zpětné vizualizace by navíc umožnila analyzovat například měsíc staré cesty vozidla a tím odhalit souvislosti mezi problémem a stavem vozu v daném momentu (stoupání nebo klesání, zatáčení, čí pouhé stání na volnoběh), které by normálně při kontrole vozu a případné testovací jízdě mechanik neodhalil. Díky kontinuálnímu zaznamenávání dat je navíc možné zachytit i okamžik, kdy došlo k určité závadě, která byla zaznamenána v řídicí jednotce. To umožní přesně určit, kdy se problém stal, jaký byl stav vozidla před tím a jaké podmínky panovaly v momentě závady, což výrazně zlepšuje a zjednodušuje diagnostiku a následné opravy.

Tato práce se zaměřuje na návrh a implementaci systému, který s využitím již existujících veřejně přístupných technologií pro získávání dat z OBD II rozhraní vozidla, jejich zpracování a uložení, umožní jejich následnou vizualizaci pomocí uživatelsky přívětivého dashboardu. Cílem je propojit technologie IoT s automobilovou diagnostikou, čímž se otevřou nové možnosti pro monitorování a analýzu vozidla. Vozidla budou ukládat načtená data v reálném čase přes OBD II rozhraní na lokální úložiště a po návratu z cesty automaticky v dosahu bezdrátové sítě odesílat na server. Server bude využívat již existující backend systému RIoT (viz. 4.5). Uživatel pak následně může k datům přistoupit a využít

vizualizace, které mu poskytují obrazovky typu dashboard a zjistit informace v konkrétním časovém rozsahu.

Kapitola 2 se zaměří na obecný koncept Internetu věcí (IoT) a jeho význam v kontextu moderní techniky, kde budou diskutovány klíčové technologické prvky umožňující propojení zařízení a výměnu dat. V této části bude rovněž popsán vývoj IoT a jeho dopady na různé průmyslové sektory. Následně se v druhé části této kapitoly zaměřím na OBD II (On-Board Diagnostics), kde se prozkoumají možnosti získávání diagnostických dat z vozidel a bude rozebrán význam OBD II standardu z hlediska údržby a monitorování technického stavu vozidel. Kapitola rovněž pokryje principy fungování OBD II systému a jeho roli v moderních vozidlech.

Kapitola 3 o dashboardu se zaměří na různé metody vizualizace dat s důrazem na přehlednou a intuitivní prezentaci informací uživatelům. Bude podrobně rozebráno, jak správně navrhnout dashboard tak, aby uživateli poskytoval potřebné informace efektivně a bez zbytečného přetížení. Součástí budou i klíčové ukazatele výkonu (KPI), které mají zásadní význam při hodnocení stavu vozidel, provozní efektivity a dalších důležitých parametrů systému.

V kapitole 4 se zaměřím na analýzu systémových požadavků, konkrétně na systém, který agreguje data z jednotlivých vozidel. Bude diskutováno, jak jsou data sbírána a jaký přínos má takový systém z hlediska efektivního spravování flotil vozidel. Následně se budu věnovat tomu, jak tato data integrovat a zpracovávat na centrální úrovni.

Návrh systému bude podrobně rozebrán v kapitole 5, kde se zaměřím na technické detaily a architekturu systému. Diskuse bude zahrnovat klíčové technologické volby, které ovlivní stabilitu a efektivitu systému. Následně v kapitole 6 shrnu, jak byla aplikace implementovnána a s použitím jakých technologií.

V kapitole 7 zhodnotím celý přístup, kde budou shrnuty dosažené výsledky, výhody a případné výzvy implementace navrhovaného řešení. Důraz bude kladen na to, jak systém přispívá ke zlepšení správy vozidel a jak může být dále optimalizován. Závěrečná kapitola 8 pak poskytne shrnutí celé práce, přičemž zhodnotím dosažené cíle a navrhnu možné oblasti dalšího výzkumu a rozvoje.

Kapitola 2

Internet věcí

Internet věcí (anglicky *Internet of Things*, zkráceně IoT) obecně představuje skupinu zařízení, která jsou spolu vzájemně propojená pomocí Internetu. Může se jednat o počítače, mobilní telefony, tablety, chytré hodinky, náramky a také různé senzory, čidla, žárovky a jiná chytrá zařízení. Tato zařízení musí mít unikátní identifikační číslo (UID) a IP [13] adresu.

Existuje mnoho různých definicí pojmu IoT a většina z nich sdílí jedno společné, vždy se jedná o fyzické objekty (zařízení), která interagují se s svým okolím a komunikují spolu mezi sebou nebo s počítači a jinými podobnými zařízeními. Například podle IBM [16] lze definovat IoT takto:

> Internet věcí označuje síť fyzických zařízení, vozidel, spotřebičů a dalších fyzických objektů, které jsou vybaveny senzory, softwarem a síťovým připojením, což jim umožňuje sbírat a sdílet data.

Nebo ve společné publikaci Somayya Madakama a dalších [27] je IoT definováno takto:

> Otevřená a komplexní síť inteligentních objektů, které mají schopnost se samoorganizovat, sdílet informace, data a zdroje, reagovat a jednat ve vztahu k situacím a změnám v prostředí.

Pro tuto práci je důležitá druhá definice Internetu věcí (IoT), která zdůrazňuje roli **inteligentních objektů** 2.2. Tato definice je klíčová, protože zohledňuje, že základním prvkem IoT nejsou pouze jednoduchá zařízení, která data sbírají a sdílejí, ale i složitější objekty, které aktivně reagují na změny ve svém okolí. Inteligentní objekty dokáží vyhodnotit různé podmínky, jako je například změna teploty, vlhkosti či přítomnost dalších zařízení, a na základě toho provádět adekvátní úpravy svého chování. Tento interaktivní prvek IoT je pro práci zásadní, neboť právě schopnost zařízení nejen monitorovat, ale i aktivně reagovat na prostředí, je základem pro efektivní využití IoT v mnoha oblastech, včetně průmyslových, dopravních a domácích aplikací.

2.1 Historie IoT

Technologie se vyvíjejí ohromujícím tempem a Internet věcí již dávno není žádnou novinkou. První zmínka se datuje do začátku osmdesátých let minulého století, kdy se programátoři Carnegie Mellon univerzity připojily k upravenému automatu Coca Coly a vždy před nákupem se podívali zda je dostupný určený nápoj a jestli je vychlazený [13].

Pojem Internet věcí byl poprvé použit v roce 1999. Použil ho Kevin Ashton [2], v té době výkonný ředitel Auto-ID Labs na MIT. Byl prvním, kdo popsal IoT během prezentace pro společnost Procter and Gamble, avšak definice IoT se v průběhu času vyvíjela. Kevin Ashton popisoval [1], že počítače a internet jsou téměř zcela závislé na lidské činnosti při získávání informací. Většina dat na internetu byla vytvořena lidmi prostřednictvím různých aktivit, jako je psaní nebo nahrávání. Problémem je omezená přesnost a pozornost lidí při sběru dat o reálném světě. Kdyby počítače mohly autonomně shromažďovat informace, mohli bychom lépe sledovat věci, minimalizovat plýtvání a optimalizovat náklady.

Kevin Ashton [2] věřil, že identifikace na rádiové frekvenci (RFID) je nezbytná pro Internet věcí. Díky ní by bylo možné snadno sledovat různé objekty, což by představovalo efektivní řešení pro správu skladových zásob. Každý objekt nebo zařízení by mělo RFID čip, QR kód nebo čárový kód a to by umožnilo počítačům je spravovat a sledovat v rámci nějakého skladu. První velké organizace, které využily Ashtonův způsob sledování zásob pomocí RFID a IoT, byly Walmart a Ministerstvo obrany Spojených států amerických, a to přibližně od roku 2002-2003.

V dalších letech se objevilo mnoho jiných užitečných způsobů využití Internetu věcí jako například chytrý zvonek, který umožnil uživateli vzdáleně sledovat, kdo se právě nachází před jeho vchodovými dveřmi, pomocí mobilního telefonu a upozornit ho notifikací, nebo také chytrá města (anglicky. Smart Cities), jež umožňují využít IoT pro snížení produkce odpadu nebo pro efektivní používání energie. Internet věcí je možné využít i pro řízení provozu nebo hledání volného parkování a tím zefektivnit dopravu ve městech a snížit dopravní zácpy v ulicích. Chytré budovy samy o sobě využívají senzory a automatizované procesy k řízení operací budovy, což zahrnuje klimatizaci, topení, ventilaci, bezpečnost, osvětlení a další systémy.

Průmyslový Internet věcí (IIoT) je rozšířením IoT, které využívá chytré senzory a aktuátory propojené s průmyslovými aplikacemi za účelem zvýšení efektivity a spolehlivosti. Vznik IIoT, přibližně od roku 2010, byl umožněn díky masivní kapacitě cloudu. Termín „Průmyslový Internet věcí“ byl poprvé použit společností GE (General Electrics) v roce 2012 [9].

Do roku 2013 se IoT stal nedílnou součástí našeho života díky integraci různých technologií, jako jsou mobilní telefony, hodinky, zdravotnické zařízení, ale například i motory letadel. V roce 2015 se chytré mobilní telefony staly důležitým komunikačním nástrojem v rámci IoT a byly využívány například v marketingu pro personalizované reklamy.

Automobily, včetně autonomních vozidel, používají mnoho technologií jako OBD a GPS a tím umožňují lehké zakomponování do IoT, například pro sledování údržby a podmínek řízení. Autonomní vozidla zase využívají cloud pro lepší rozhodování na základě aktuálních dat, třeba při vyhýbání se okolním vozidlům nebo přizpůsobení jízdy povrchu vozovky.

2.2 Inteligentní objekty (zařízení)

Internet věcí je v dnešní době velice rozšířený. Se zařízeními, která využívají tento koncept, se setkáváme každý den. Jedná se o mobilní telefony, chytré hodinky a chytré váhy s kontrolou správné stravy podle aplikace nebo třeba samotná čidla teploty v domácnosti, detektory kouře či správného okysličení místnosti. Je jich tolik, že je potřeba je nějak kategorizovat pro větší přehlednost. Bohužel neexistuje žádné správné nebo nejlepší kategorizování těchto zařízení. Mnoho publikací uvádí různá, někdy stejná jindy úplně jiná, rozdělení.

Zařízení součástí Internetu věcí lze kategorizovat na základě jejich rolí, tedy toho, co vykonávají. Autoři [11], González C. a spol. zařízení rozdělují například na čidla (sensors), akční členy (actuators) a chytrá zařízení (smart objects).

Čidla (sensors)

Čidla slouží k měření určitých parametrů. Může to být například teplota, vlhkost ve vzduchu nebo množství světla v místnosti [6]. Čidla lze právě podle vlastností, jež měří, dále dělit na menší podskupiny. U čidel je důležité, že jejich měření musí být velice přesné a samotné čidlo tedy nesmí ovlivňovat měřenou vlastnost. Přesnost čidla je jedním z nejdůležitějších parametrů a další možnost jak kategorizovat tato zařízení.

Dalšími neméně důležitými parametry jsou energetická náročnost a výrobní cena, které ovlivňují nákup takových zařízení a jejich následné využití. Nakonec i spolehlivost, kalibrace nebo doba nastavení mohou čidla dále dělit a silně ovlivnit, kde se použijí.

Čidla je možné kategorizovat i podle jejich aktivity, tedy na aktivní a pasivní, nebo podle vlastností, které mají ve svém okolí měřit (polohové, pohybové, teplotní nebo vlhkostní, akustické, plynové, silové, optické a magnetické senzory).

Akční členy (actuators)

Mohli bychom říci, že čidla pouze sledují své okolí a nijak do něj nezasahují, narozdíl od akčních členů, které ho aktivně ovlivňují. V rámci prostředí, ve kterém zařízení jsou, můžeme čidla považovat za pasivní prvky a akční členy za prvky aktivní. Akční členy tedy využívají elektrickou energii a převádí ji na energii mechanickou (pohyb). Tento pohyb může být například otevření okna nebo otočení ventilem topení. Akční členy tedy lze dále kategorizovat podle pohybu, který vykonávají (posun, rotace), nebo jak uvadí Mohamed [30] podle provozních principů na elektrické, hydraulické a pneumatické.

Chytrá zařízení (smart objects)

Chytrá zařízení spojují obě předchozí kategorie a s použitím vestavěného operačního systému vytváří zařízení obsahující čidla i akční členy. Tato zařízení tedy jsou mnohem atraktivnější pro použití, jelikož umožňují využít výhody obou již zmíněných kategorií. Chytrá zařízení mohou být součástí mnoha námi velice známých zařízení, jako jsou například chytré kávovary, televize nebo lednice.

Samozřejmě je lze také rozdělit do různých kategorií. Kategorizovat můžeme například podle úrovně inteligence zařízení (schopností). González García [11] uvádí tři kategorie:

- **úroveň inteligence** – práce s informacemi, oznámení o problémech a rozhodování
- **umístění inteligence** – inteligence součástí samotného objektu, inteligence sítě, nebo kombinovaná inteligence
- **agregace inteligence** – popisuje chytré objekty sestávající z několika částí a určuje, zda může daná část fungovat samostatně a zachovat si svou inteligenci.

Zařízení je také alternativně možné kategorizovat podle použitých komunikačních protokolů. Rozhodovalo by se tedy podle toho, jaký protokol dané zařízení používá ke komunikaci s jinými zařízeními v IoT systému. Takové dělení by umožnilo lépe porozumět vzájemné

interoperabilitě mezi zařízeními a optimalizovat jejich nasazení v různých prostředích. Rozdělení podle komunikačních protokolů by rovněž mohlo usnadnit výběr vhodné technologie pro konkrétní aplikace, zajistit vyšší efektivitu přenosu dat a přizpůsobit systémy specifickým požadavkům na zabezpečení a spolehlivost.

2.3 Komunikace v IoT

Systém internetu věcí je velmi rozsáhlý a pro jeho správné fungování jsou klíčové dvě základní složky. Samotnou podstatou jsou jednotlivá zařízení, která spolu komunikují. Druhou důležitou součástí systému IoT je komunikace mezi těmito zařízeními. Pro správnou komunikaci je nutné vybrat dobrý komunikační prostředek (technologii) a s tím spojený komunikační protokol. Nyní si v několika dalších odstavcích projdeme používané komunikační technologie a protokoly.

Komunikační technologie používané v IoT mohou být rozděleny do několika skupin podle různých kritérií. Například John [15] využívá rozdělení podle dosahu dané technologie na sítě s krátkým dosahem, sítě s dlouhým dosahem a mobilní sítě. Já jsem se rozhodl pro podobné rozdělení, vycházející z publikace [37], které dělí technologie podle typu na kabelové, já se budu zabývat hlavně Ethernetem a CANBus technologií, bezdrátové sítě PAN a LAN a mobilní sítě. Toto rozdělení jsem zvolil, protože umožňuje přehledné uspořádání jednotlivých technologií na základě způsobu přenosu dat, což hraje klíčovou roli v návrhu IoT systémů. Díky této kategorizaci lze lépe porozumět specifickým výhodám a omezením více přístupů, což napomáhá při rozhodování o vhodné technologii pro konkrétní aplikace s ohledem na požadavky na rychlost, spolehlivost a dosah přenosu.

Ethernet a IP

Ethernet, který je standardizován v rámci IEEE 802.3 [36], je klíčovou technologií používanou pro propojení IoT zařízení. Používá se primárně k propojení uzavřených sítí s okolním světem přes rozhraní IoT brány. Ethernet se postupně vyvíjel, aby podporoval vyšší přenosové rychlosti a větší vzdálenosti, a to díky přechodu od kroucených párů k optickým vláknům. Každé zařízení připojené k síti je identifikováno unikátní adresou na vrstvě MAC (Medium Access Control), což umožňuje přímé adresování zařízení pro odesílání a přijímání Ethernetových rámců. Tyto rámce mohou být rozděleny na fragmenty a následně kontrolovány na chyby. Pokud by byly rámce poškozené, jsou na cílovém místě zahazovány.

S nárůstem počtu zařízení připojených k síti přestalo být možné spravovat komunikaci pouze pomocí Ethernetových MAC adres. Původní adresní prostor, který měl pouze 256 adres (v roce 1974), byl vyčerpán během necelých 10 let. Proto byla zavedena IP verze 4 (IPv4), která pracuje na vyšší vrstvě (síťové) a podporuje adresní prostor 2^{32} adres. Tyto adresy jsou dlouhé 32 bitů a skládají se ze čtyř oktetů oddělených tečkou.

I přes zavedení IPv4 byl adresní prostor časem vyčerpán kvůli dalšímu rozmachu internetu, což vedlo v roce 1998 k zavedení IP verze 6 (IPv6). IPv6 poskytuje výrazně větší adresní prostor s podporou 2^{128} adres, přičemž každá IPv6 adresa je dlouhá 128 bitů. Adresy jsou rozděleny na dvě části: 64 bitů představuje prefix, který identifikuje síť a zařízení, a 64 bitů představuje Interface ID (IID), které identifikuje rozhraní sítě a musí být v rámci sítě unikátní.

Adresování je zásadní pro jednoznačnou identifikaci zařízení a směrování provozu k těmto zařízením a od nich. IPv4 původně rozdělil adresy na dvě části – identifikátor sítě a hostitele. Tento systém se ale rychle ukázal jako nedostatečný, protože počet zařízení připojených

k internetu rychle rostl. Problém byl vyřešen zavedením tříd adres, které rozdělovaly adresy podle délky síťového identifikátoru. Pevné třídy však vedly k plýtvání adresami, a proto byl zaveden systém CIDR (Classless Inter-Domain Routing), který umožňuje přidělovat IP adresy flexibilně podle potřeb jednotlivých sítí. Tato flexibilita je dobře přizpůsobena pro sítě s omezenými zařízeními, typicky pro IoT.

CANBus

Controller Area Network (CAN) bus je nízkonákladová síťová technologie [35], která slouží jako standard pro vnitřní komunikaci vozidel a umožňuje elektronickým řídicím jednotkám (ECU) komunikovat v reálném čase. I když je CANbus efektivní, byl navržen bez ohledu na bezpečnost, což činí vozidla zranitelnými vůči útokům.

CANBus, známý jako architektura ISO 11898, je sběrnice, která propojuje vestavěné komponenty, IO rozhraní a brány pro komunikaci s externími systémy. Systémy CAN zahrnují řadič a převodník, které umožňují odesílání a příjem informací mezi podsystémy. CAN poskytuje detekci chyb, ale neimplementuje zabezpečení v žádné vrstvě abstrakce definované v protokolu CAN. Komunikace probíhá pomocí diferenciální dvojice signálů CAN high a CAN low a může být nakonfigurována na standardní datové rychlosti 125 kbps, 500 kbps a 1 Mbps.

Bezdrátové sítě PAN a LAN

Bezdrátová komunikace je klíčová pro uživatele a IoT zařízení, protože umožňuje připojení k jiným sítím přes přístupový bod a jeho bránu (WiFi router, vysílač mobilního signálu). Bezdrátové technologie zahrnují různé typy sítí, jako jsou mobilní sítě (např. GSM, UMTS, LTE), bezdrátové LAN (WLAN) a PAN (WPAN), bezdrátové senzorové sítě (WSN) [37], satelitní komunikace nebo pozemní mikrovlnné sítě.

Standard IEEE 802.11 (WLAN) je široce využíván pro propojení IoT zařízení v lokálních sítích. Specifikuje fyzickou vrstvu (PHY) a vrstvu řízení přístupu k médiu (MAC), zatímco umožňuje interoperabilitu s vyššími vrstvami. Tento standard operuje v bezlicenčním pásmu 2,4 GHz a podporuje přenosové rychlosti až stovky Mbit/s. IEEE 802.11 může pracovat jak v infrastrukturní architektuře, tak i v ad-hoc sítích, což je vhodné například pro mobilní ad-hoc sítě (MANET) nebo sítě pro vozidla (VANET).

Standard IEEE 802.15.4 je navržen pro bezdrátové osobní sítě (WPAN) a je vhodný pro zařízení s nízkou spotřebou energie, jako jsou senzory a RFID tagy. Podporuje dva různé typy zařízení, a to zařízení s omezenou funkcionalitou (RFD), která typicky periodicky sbírají data a připojují se k zařízením s plnou funkcionalitou (FFD), která řídí síť a komunikují s RFD i dalšími FFD. PHY vrstva podporuje komunikaci v různých frekvenčních pásmech (např. 868/915 MHz a 2,4 GHz), zatímco MAC vrstva nabízí různé režimy přenosu (např. CSMA/CA).

IoT protokoly

Pro komunikaci v IoT je využíváno mnoho různých komunikačních protokolů, které se liší svým použitím. Zahrnul jsem zde podle mě nejpoužívanější a nejrozšířenější z nich, vycházející z publikace [37].

ZigBee: Založený na IEEE 802.15.4, podporuje IP komunikaci s minimálním paměťovým a energetickým nárokem. Síť dokáže pokrýt až 1024 uzlů a dosah 200 metrů. Podporuje

zabezpečení pomocí AES a nabízí síťové funkce jako skenování, vytvoření sítě, směrování a hledání zařízení.

LoRa: Proprietární technologie od Semtech pro energeticky úspornou komunikaci na velké vzdálenosti. Dosahuje pokrytí až 15 km a má vysokou citlivost, což umožňuje průchod signálu skrze budovy. Podporuje různé třídy zařízení, které se liší spotřebou a způsobem komunikace.

RFID: Technologie používající elektromagnetické pole pro sledování a identifikaci objektů. Aktivní RFID štítky mají vlastní zdroj energie, zatímco pasivní štítky vyžadují energii z čtečky. RFID našel široké uplatnění v IoT, například při sledování předmětů nebo osob.

Z-Wave: Nízkopříkonový protokol pro domácí automatizaci, pracuje v pásmu pod 1 GHz a umožňuje komunikaci až do vzdálenosti 100 m. Nabízí interoperabilitu zařízení na aplikační vrstvě a podporuje cloudový přístup přes bránu Z-Wave centrální jednotky.

NFC: Technologie na krátkou vzdálenost (do několika centimetrů) používaná např. pro bezkontaktní platby nebo výměnu malých datových souborů. Zařízení mohou fungovat ve třech režimech: emulace karty, čtení a zápis nebo peer-to-peer komunikace.

Bluetooth: Standard pro komunikaci na krátké vzdálenosti, který podporuje přenos dat mezi zařízeními. Energeticky úsporná verze Bluetooth LE je vhodná pro IoT aplikace díky nízké spotřebě energie.

Mobilní síť

Mobilní síť funguje na bázi WAN a skládá se z jednotlivých částí, kdy každá obsahuje stanici s vysílačem a přijímačem. Velikost těchto částí se pohybuje od stovek metrů až po 25 km, přičemž se překrývají. Mobilní zařízení, jako jsou chytré telefony a tablety, se musí při přechodu do jiné části přepojit na korespondující stanici. Mobilní zařízení volají jiná zařízení pomocí E.164 telefonních čísel. Ke komunikaci je nutné mít předplatné veřejné mobilní sítě.

Mezi takové technologie patří například LTE-M nebo NB–IoT [28], rozšiřující stávající funkčnost LTE, ale i klasické mobilní sítě, jež většina z nás používá každý den, a to 3G, 4G (LTE) nebo 5G. Mobilní sítě jsou pro IoT atraktivní hlavně kvůli dostupnosti po celém světě, i v místech se špatnou nebo žádnou infrastrukturou.

2.4 Architektura IoT

Systémy Internetu věcí (IoT) v dnešní době mohou být hodně rozsáhlé a často vytvářejí hodně složité a výkonnostně náročné sítě. Vzhledem ke složitosti těchto systémů je velice důležité použití správné a efektivní architektury pro zajištění optimálního výkonu a integrace v rámci rozsáhlého IoT systému. Architektury IoT systémů jsou často tvořeny jednotlivými vrstvami (layers), ale každá se může lišit jejich počtem.

Třívrstvá architektura

Jedním ze základních a nejjednodušších modelů podle [27] a [29] je třívrstvá architektura. Tato architektura se skládá z vrstvy vnímaní (perception layer), transportní vrstvy (transport layer) a aplikační vrstvy (application layer).

Součástí vrstvy vnímání jsou jednotlivá zařízení vybavená senzory pro interakci s okolím (snímač teploty, čidlo otevřených dveří). Tato zařízení získávají data ze svého okolí nebo naopak okolí ovlivňují (více v kapitole 2.2). Data získaná ve vrstvě vnímání jsou přenášena

do přenosové vrstvy, která zajišťuje použití správných protokolů a komunikačních metod a následně předává data aplikační vrstvě. Aplikační vrstva zpracovává přijatá data a dále s nimi provádí operace a zobrazuje je koncovým uživatelům. Jednoduše se jedná už o samotnou aplikaci, která je dostupná pro uživatele v jeho zařízení nebo na cloudu (například Google Home aplikace pro správu chytrých zařízení).

Vícevrstvé architektury

Existují i jiné architektury s jiným počtem vrstev, jako například čtyřvrstvá architektura často pojmenovávaná jako Architektura orientovaná na Služby (SoA) [25]. Ta využívá již nám známé vrstvy a přidává jednu navíc. Vrstva vnímání a transportní vrstva zůstávají stejné, ale navíc ještě přibývá vrstva služeb (service layer) a aplikační vrstva mění svou funkci.

Objevují se i architektury s více vrstvami. Například publikace [29] uvádí pětivrstvou architekturu. Architektury se mohou skládat i z šesti, sedmi a teoreticky i více vrstev. S přidáváním vrstev se na jednu stranu rozloží celý systém na více menších jednodušších a tedy se i zvyšuje přehlednost a možnost modifikovat pouze jednu vrstu (ne celý systém), ale zároveň jsou takové systémy složitější na pochopení, správu ale i návrh či tvorbu.

2.5 Praktické využití IoT

IoT zahrnuje široké spektrum aplikací, od běžných spotřebitelských zařízení, jako jsou chytré domácí spotřebiče, až po složitější průmyslové systémy. Tato technologie se stává klíčovou složkou moderní digitální transformace, protože umožňuje sběr dat v reálném čase, optimalizaci procesů a zvýšení efektivity v mnoha odvětvích.

V oblastech, jako je zdravotnictví, automobilový průmysl, chytré budovy a chytrá města, IoT zavádí automatizované systémy zvyšující bezpečnost a komfort. Ve zdravotnictví sledují chytré náramky [3] fyzickou aktivitu a vitální funkce pacientů, což umožňuje včasnou reakci lékařů. V automobilovém průmyslu autonomní vozidla využívají senzory k optimalizaci jízdních tras a prevenci nehod. V chytrých budovách automatizace osvětlení a vytápění šetří energii a zvyšuje komfort obyvatel. V chytrých městech inteligentní semafory optimalizují dopravu, čímž zlepšují plynulost a bezpečnost na silnicích. S rostoucí schopností senzorů a zařízení zpracovávat a analyzovat velké množství dat se stává IoT nepostradatelným nástrojem pro optimalizaci operací i na globální úrovni.

Tento technologický trend je základem pro další inovace v oblasti Industry 4.0 a chytrých automobilů, kde IoT pomáhá zvyšovat autonomii systémů, flexibilitu výroby a poskytuje nové úrovně bezpečnosti a komfortu. Pokud se zaměříme na Industry 4.0 a chytrá auta, jde především o integraci IoT do výrobních procesů, skladování, infrastruktury nebo automobilních technologií, což umožňuje dosáhnout nové úrovně automatizace, konektivity a inteligence.

Průmysl 4.0 (Industry 4.0)

Průmysl se od počátku industrializace měnil díky technologickým pokrokům. První průmyslová revoluce byla charakterizována mechanizací, druhá intenzivním využitím elektrické energie a třetí digitalizací. Nyní, s pokročilou digitalizací v továrnách a propojením klasické výroby s IoT a chytrými zařízeními, se objevuje nový termín – Průmysl 4.0. Jak uvádí publikace [23], vize budoucí výroby v rámci Průmyslu 4.0 zahrnuje modulární a efektivní

výrobní systémy, kde produkty samy řídí svůj výrobní proces. To umožňuje vyrábět individuální produkty v sériích o velikosti jednoho kusu za ekonomických podmínek, které jsou běžné u masové výroby. Tento přístup získal na významu v Německu díky doporučení na implementaci v rámci strategie high-tech 2020.

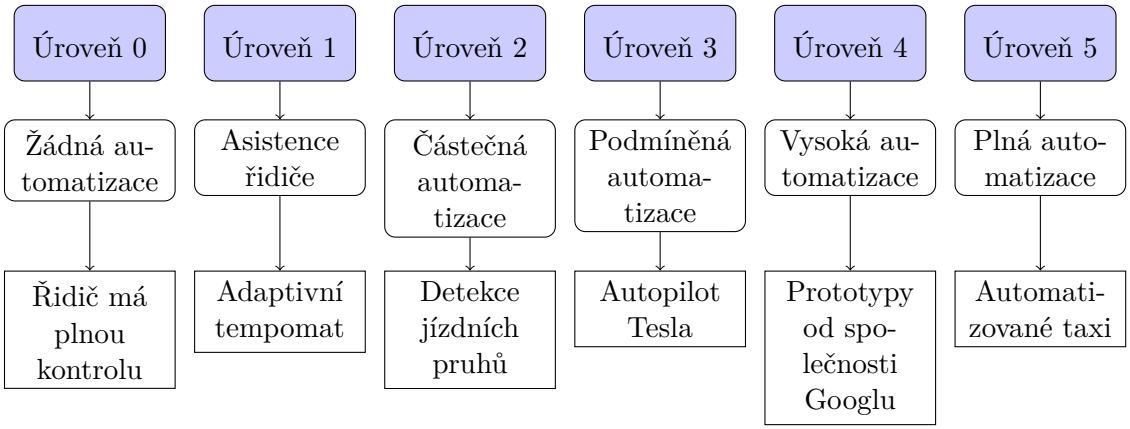
Koncept Průmyslu 4.0 je poháněn dvěma hlavními silami, a to poptávkovou silou (application-pull) a technologickým tlakem (technology-push). Poptávková síla zahrnuje zkrácení vývojových cyklů, individualizaci na míru, flexibilitu, decentralizaci a efektivitu zdrojů. Technologický tlak se projevuje mechanizací a automatizací, digitalizací a síťovým propojením, stejně jako miniaturizací technologií, které umožňují větší efektivitu a inovace v průmyslových procesech.

Termín Průmysl 4.0 zahrnuje širokou škálu konceptů, jako jsou chytré továrny, kyberfyzikální systémy a samoorganizace (Self-Organization). Chytré továrny jsou autonomní výrobní systémy, které jsou plně vybaveny senzory, akčními členy a autonomními systémy. Kyberfyzikální systémy spojují fyzickou a digitální úroveň, což je patrné například v preventivní údržbě, kde jsou procesní parametry mechanických součástí zaznamenávány digitálně pro sledování fyzického stavu součástí. Výrobní systémy se stávají čím dál tím více decentralizovanými. To je spojeno s rozkladem tradiční hierarchie ve výrobě a přechodem k decentralizované samoorganizaci. Moderní výrobní systémy jsou navrhovány tak, aby se přizpůsobovaly lidským potřebám, a důraz je kladen na společenskou odpovědnost firem, kde udržitelnost a efektivní využívání zdrojů hrají klíčovou roli pro budoucí úspěch.

Chytré automobily

Technologický pokrok neustále zrychluje a revoluce v této oblasti podporuje koncept automatizace v různých sektorech. Automatizace má za cíl eliminovat manuální práci a zlepšit uživatelskou zkušenost. Koncept Internetu věcí (IoT) se neomezuje pouze na chytré telefony a počítače, ale zahrnuje také například vozidla. IoT umožňuje propojení různých zařízení a rozšiřuje dynamickou mobilní komunikaci mezi automobily a jejich uživateli, což má za cíl poskytovat informace o dopravě v reálném čase, zlepšit pohodlí cestujících a zvýšit efektivitu dopravy.

Řidiči však čelí mnoha výzvám, jako je únava, rozptylování nebo hlučnost, což může vést k nepozornosti a následným nehodám. Důležitým bezpečnostním prvkem je i pravidelná údržba automobilů, která předchází selhání mechanických a elektrických systémů. Jak zmiňuje autor [34] v reakci na tyto problémy se výrobci automobilů zaměřují na vývoj pokročilých asistenčních systémů řidiče (ADAS), které zajišťují bezpečnost, komfort a efektivitu. ADAS jsou prvním krokem k autonomním vozidlům. Tato vozidla jsou podle autora výše zmíněné publikace klasifikována do šesti úrovní podle míry automatizace, přičemž každá úroveň nabízí různé schopnosti a neustálé zlepšování bezpečnosti a efektivity. Na obrázku 2.1 je znázorněna každá jednotlivá úroveň s popisem a příkladem v poslední buňce.



Obrázek 2.1: Shrnutí šesti úrovní automatizace vozidel

Úroveň 0 – Žádná automatizace

Vozidla na této úrovni vyžadují, aby řidiči plně ovládali všechny funkce. Systémy na této úrovni neprovádějí žádnou automatizaci řízení a poskytují pouze podpůrné systémy, které pomáhají řidičům při detekci objektů a událostí (OEDR), například systémy pro okamžité brzdění při nebezpečí kolize.

Úroveň 1 – Asistence řidiče

Úroveň 1 zahrnuje vozidla s určitou automatizací, která pomáhá řidiči. Systémy, jako jsou senzory pro nasazení airbagů a sledování tlaku v pneumatikách, přispívají k bezpečnosti. Řidič však musí tyto systémy aktivovat manuálně.

Úroveň 2 – Částečná automatizace

Na této úrovni vozidla mohou ovládat řízení, akceleraci a brzdění, ale pokud dojde k selhání automatického systému, řidič musí být připraven zasáhnout. Mezi funkce patří detekce jízdních pruhů a autonomní nouzové brzdění (AEB).

Další úrovně automatizace jsou podrobně popsány v publikaci [34]. V této práci se budeme dále zaměřovat na OBD standard, tedy hlavně na komunikaci s vozidlem na úrovni 0. Ta se soustředí na základní diagnostiku a podporu řidiče bez jakékoli automatizace řízení.

2.6 OBD (On-Board Diagnostics)

OBD představuje systém, který umožňuje monitorování a diagnostiku motoru a dalších klíčových komponent vozidla. Tento systém byl vyvinut v reakci na potřebu efektivního sledování emisí a zajištění ekologičtějších automobilů. První generace OBD, známá jako OBD I, byla zavedena v 80. letech 20. století a sloužila jako základní nástroj pro identifikaci problémů v motoru. OBD I umožnil mechanikům provádět diagnostiku pomocí jednoduchých čtecích zařízení, ale postrádal standardizaci a sofistikovanější funkce.

Na konci 20. století vedl rychlý rozvoj mikropočítačově řízených systémů k vytvoření nových standardů pro komunikaci mezi řídicími systémy a diagnostickými zařízeními. V roce 1996 byl v USA zaveden standard OBD II, který umožňuje různé komunikační protokoly

pro diagnostiku vozidel. Evropským ekvivalentem je EOBD (European On-Board Diagnostics) [39], který byl v Evropě zaveden v roce 2000 a původně platil pouze pro benzinové motory. Od roku 2003 se standard rozšířil i na dieselové motory a vozidla poháněná LPG. OBD II standard také zavedl jednotnost v komunikaci s diagnostickými zařízeními a normy pro kódování chyb, které umožňují jednotně označit určité problémy a chyby vozidel.

OBD II monitoruje prakticky všechny komponenty, které mohou ovlivnit výkon, chování a bezpečnost vozidla. Pokud je detekován problém, systém OBD II rozsvítí varovnou kontrolku na palubní desce, případně i upozorní řidiče akustických varováním nebo v nouzi vypne motor či nefunkční podsystém. Systém také ukládá důležité informace o zjištěné poruše, které technikovi umožní přesně najít problém pomocí speciálního nástroje (autodiagnostiky) a provést opravu. Informace z tohoto systému nejsou přístupné pro běžného provozovatele vozidla (například na infotainmentu) bez již zmíněné autodiagnostiky.

Většina výrobců vozidel implementuje vlastní platformy pro správu elektronických řídicích systémů, které většinou využívají jednoúrovňovou sběrnicovou topologii 2.6 založenou na standardu CAN (Controller Area Network) a dalších podsíťových systémech, jako je LIN (Local Interconnect Network). Tímto způsobem se zajišťuje efektivní komunikace mezi různými systémy vozidla, což přispívá k větší bezpečnosti, komfortu a ekologičnosti.

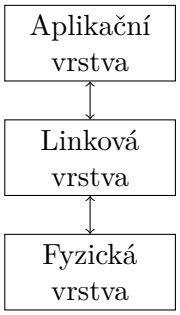
Důležitou součástí elektronického systému vozidla je diagnostický port (DLC – Data Link Connector), který nejen umožňuje diagnostikovat systémy vozidla, ale také umožňuje modifikaci stávajícího softwaru a blokování vybraných funkcí, například omezení otáček motoru při přepravě vozidla. Moderní mechatronické systémy často obsahují adaptivní algoritmy, které reagují na opotřebení různých komponent a přizpůsobují systém novým požadavkům. Přestože tato adaptabilita přináší výhody, může také skrývat chyby v adaptačním procesu, což představuje potenciální riziko.

Topologie OBD II

Síť OBD II je navržena s cílem propojit různé elektrické moduly vozidla pomocí otevřené architektury. Jak zmiňuje autor [10], otevřená architektura umožňuje přidávat nebo odstraňovat moduly (uzly) bez významného dopadu na ostatní části systému, ať už hardwarového nebo softwarového charakteru. V této síti jsou data přenášena mezi uzly za účelem eliminace nadbytečných senzorů a dalších prvků. Síť využívá metodu Carrier Sense Multiple Access (CSMA) s nedestruktivním řešením kolizí, což umožňuje upřednostnění důležitějších zpráv při případné kolizi.

Většina aplikací v automobilovém průmyslu dnes používá jednoúrovňovou topologii sítě, kde jsou všechny uzly propojeny jednou datovou sběrnicí. Topologie může být realizována pomocí více datových kabelů, ale pokud všechny uzly vysílají a přijímají data z jedné cesty, všechny uzly přijímají rámce současně a komunikace na každé sběrnici je identická, topologie zůstává jednoúrovňová. Nevýhodou této topologie je, že nelze zaručit časovou latenci dat, s výjimkou rámců s nejvyšší prioritou.

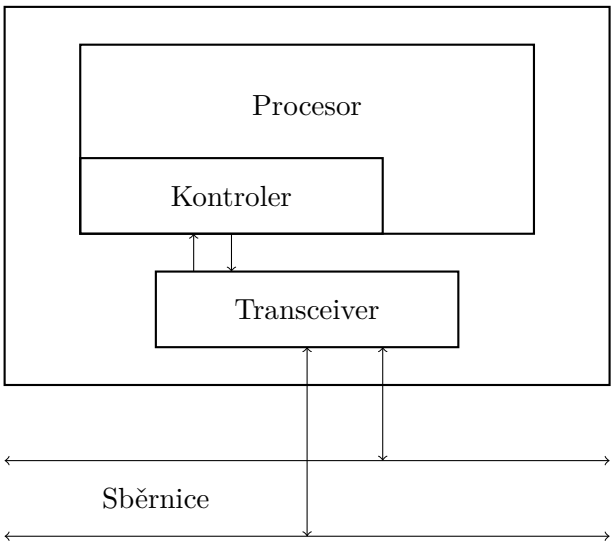
Síť OBD II se také řídí modelem OSI, který je zobrazen na obrázku 2.2. Ten je rozdělen do tří vrstev: fyzická vrstva, linková vrstva a aplikační vrstva. Fyzická vrstva se stará o fyzické přenosy dat mezi uzly, zatímco linková vrstva se zabývá převodem a kontrolou dat (kontrola případných chyb). Aplikační vrstva pak definuje, jakým způsobem bude komunikace a řízení probíhat na vyšší úrovni (komunikace se vstupními a výstupními zařízeními nebo uživateli).



Obrázek 2.2: OSI (Open Systems Interface) model

Rozhraní OBD II

Uzel OBD II systému zahrnuje tři klíčové komponenty: procesor, kontrolér OBD II a takzvaný transceiver (vysílač i přijímač v jednom). Procesor může mít podobu univerzálního mikroprocesoru, digitálního signálního procesoru nebo počítače. V některých případech mohou být kontrolér i transceiver integrovány do jednoho čipu s procesorem. Kontrolér OBD II funguje jako vyrovnávací paměť pro přenos zpráv. Umožňuje zprávám s vyšší prioritou přepsat ty s nižší prioritou a zajišťuje kompatibilní příjem i vysílání zpráv podle specifikací OBD II.



Obrázek 2.3: OBD II uzel

OBD II používá diferenciální přenos dat, což znamená, že zprávy jsou přenášeny pomocí dvou vodičů s opačnými signály, což zvyšuje odolnost vůči šumu a rušení z okolního prostředí. Tento způsob přenosu také snižuje elektromagnetické rušení (EMI), které může být generováno jednovodičovým přenosem. Vysílané zprávy jsou šířeny všem uzlům v síti, a každá zpráva je opatřena identifikátorem, který určuje její prioritu. Pokud dojde ke kolizi při přenosu dvou zpráv současně, zpráva s nižší prioritou se přeruší a vyšší prioritní zpráva se odešle bez přerušení.

Kapitola 3

Dashboard

Obrazovka typu dashboard představuje vizuální zobrazení nejdůležitějších informací potřebných k dosažení jednoho či více cílů přehledně uspořádaných na jediné obrazovce, aby mohly být sledovány a pochopeny hned na první pohled [8]. Podobně jako přístrojová deska v automobilu poskytuje kritické údaje pro řízení vozidla, dashboard poskytuje přehled klíčových ukazatelů a metrik, které pomáhají při rozhodování a řízení procesů.

Dashboardy jsou navrženy tak, aby poskytovaly rychlý přehled o aktuálním stavu, často prostřednictvím kombinace textu a grafických prvků, jako jsou různá měřidla, grafy a ukazatele. Hlavní předností je jejich schopnost přehledně vizualizovat klíčové informace v kontextu konkrétních cílů, což může zahrnovat různé typy dat z různých zdrojů. Dashboardy dnes nejsou určeny jen pro manažery nebo vedoucí pracovníky, ale využívají je i jednotlivci v nejrůznějších oblastech.

Klíčovým rysem dashboardů je jejich schopnost zobrazit vše podstatné na jedné obrazovce, čímž umožňují monitorování bez nutnosti procházet mezi více zobrazeními. Využití grafických prvků není jen pro estetiku, ale slouží k efektivnímu zpracování informací tak, aby bylo možné okamžitě identifikovat oblasti vyžadující pozornost. Kromě toho musí být dashboardy přizpůsobeny konkrétním potřebám uživatele nebo týmu, aby skutečně splnily svůj účel.

3.1 Kategorie

Dashboardy lze kategorizovat několika způsoby. I když může být tato snaha omezená a nedokonalá, má svůj význam, protože nám pomáhá zkoumat výhody a různé způsoby využití dashboardů. Few [8] považuje proces klasifikace za intelektuální cvičení, které umožňuje hlouběji proniknout do tématu. Přitom však neklade přehnaný důraz na jednu konkrétní metodu klasifikace a neprosazuje ji jako jedinou správnou.

Například podle proměnné „platforma“ můžeme dashboardy rozdělit na ty, které fungují v režimu klient/server a na ty, které běží v prohlížečích. Existuje mnoho proměnných, podle kterých lze dashboardy kategorizovat, a autor uvádí některé z nich, i když připouští, že tento seznam není vyčerpávající a představuje spíše jeho vlastní zkoumání možností dashboardů.

Dělení podle role

Jedním z nejpřínosnějších způsobů, jak kategorizovat dashboardy, je podle jejich role, tedy podle typu aktivit, které podporují. Few rozděluje dashboardy do tří hlavních rolí: strategické, analytické a operační. Tato klasifikace je užitečná, protože každá z těchto rolí ovliv-

ňuje vizuální design dashboardu. V jiné publikaci [33] je uvedeno stejné rozdělení, akorát se liší názvem jednotlivých kategorií (strategické, taktické, operační).

Strategické – Hlavním využitím dashboardů je dnes jejich strategická funkce. Tyto dashboardy poskytují manažerům rychlý přehled o zdraví a příležitostech v podniku. Zaměřují se na klíčové ukazatele výkonnosti a prognózy, které manažerům pomáhají rozhodovat o dlouhodobém směru společnosti. Vzhledem k tomu, že cílem není rychlá reakce na aktuální změny, ale spíše dlouhodobá orientace, strategické dashboardy obvykle nevyžadují data v reálném čase, ale spíše statické snímky, které jsou aktualizovány měsíčně, týdně nebo denně.

Analytické – Analytické dashboardy podporují hloubkovou analýzu dat a vyžadují více kontextu, jako jsou bohatší porovnání, delší historie a detailnější hodnotící mechanismy výkonnosti. Na rozdíl od strategických dashboardů zde často dochází k interakci s daty, například k procházení podrobností, což analytikům umožňuje zkoumat příčiny trendů a vyhodnocovat jejich dopady. Tyto dashboardy poskytují nejen přehled, ale i možnost hlouběji zkoumat vztahy mezi daty.

Operační – Dashboardy pro monitorování operací jsou navrženy tak, aby sledovaly dynamické a rychle se měnící aktivity. Klíčovým aspektem je okamžité povědomí o situaci a rychlá reakce, pokud dojde k odchylce od normálního stavu. Tyto dashboardy často obsahují velmi konkrétní a detailní informace, jako jsou čísla objednávek nebo umístění zboží ve skladu. Vzhledem k jejich povaze je důležitá interaktivita, která umožňuje rychlý přístup k podrobnostem, pokud je to potřeba.

Dělení podle využití

Dělení podle využití dat v rámci různých oblastí zahrnuje několik klíčových kategorií [8], přičemž každá z nich se zaměřuje na specifické metriky a příklady měření. V oblasti prodeje se například sleduje počet objednávek, tržby a prodané kusy, což pomáhá optimalizovat prodejní strategie. V marketingu se analyzuje podíl na trhu, úspěch kampaní a počet nových zákazníků, aby se zhodnotila efektivita marketingových aktivit. Finance se soustředí na příjmy, výdaje a výsledný zisk, což je zásadní pro řízení hospodářských výsledků. V technické podpoře se měří počet hovorů, vyřešené případy a spokojenost zákazníků, což přispívá k zlepšení zákaznického servisu. V oblasti výroby se sleduje počet výrobků, doba výroby a počet vadných kusů, což pomáhá zvyšovat efektivitu výroby. Lidské zdroje se zaměřují na spokojenost zaměstnanců a počet otevřených pozic, což je klíčové pro řízení lidského kapitálu. V oblasti informačních technologií se hodnotí doba běhu sítě a využití systémů, což je důležité pro zajištění bezproblémového chodu IT infrastruktury. Nakonec webové služby monitorují počet návštěvníků, zobrazení stránek a délku návštěv.

3.2 Vizuální vnímání

Vizuální vnímání hraje klíčovou roli v tom, jak uživatelé interpretují dashboardy a interagují s nimi. Dobře navržený dashboard využívá principy vizuálního vnímání k lepšímu porozumění a usnadnění rozhodování. Designové prvky, jako jsou barvy, rozvržení a vizuální hierarchie, ovlivňují způsob zpracování informací a mají dopad na efektivitu, s jakou mohou uživatelé získávat poznatky.

Barva a její vliv

Jak bylo zdůrazněno v úvodu, barva má významný vliv na vizuální vnímání. Volba barev na dashboardu ovlivňuje nejen estetickou přitažlivost, ale také efektivnost komunikace dat. Barvy mohou evokovat emocionální reakce a přitahovat pozornost k důležitým informacím. Efektivní použití kontrastních barev může zvýšit viditelnost důležitých metrik, což usnadňuje uživatelům rychlé identifikování trendů a anomálií. Naopak špatné barevné volby mohou vést k záměnám a nesprávné interpretaci dat, zejména u jedinců s poruchami barevného vidění.

Asi 10% populace [19, 8] trpí barvoslepostí, což je neschopnost rozlišovat určité barvy. Tato skupina lidí je z 95% složena z mužů. Je to hlavně tím, že geny ovlivňující správné vidění barev jsou uloženy v chromozomu X, který mají muži pouze jeden na rozdíl od žen, a tedy při jeho poškození nedojde k náhradě. Barvoslepost může být úplná, kdy lidské oko nedokáže vnímat jiné barvy než odstíny šedi, černou a bílou, nebo částečná. Částečná barvoslepost se dále dělí podle barev, které postižený nemůže vidět, do 3 tří hlavních skupin [24]: deuteranopie (neschopnost vnímat zelenou), protanopie (neschopnost vnímat červenou) a tritanopie (neschopnost vnímat modrou). Nejčastější je deuteranopie (7%), poté protanopie (2.7%) a tritanopie (0.02%). V dashboardech je proto důležité volit barevné schéma, které bere v úvahu tato omezení. Pokud vybíráme barvy pro vizualizaci dat z vozidla a předpokládáme, že cílovou skupinou budou převážně muži, vyhneme se použití červené a zelené. Nejlepším řešením je obecně používat různé odstíny pouze jedné barvy (například modré).

Obrázek 3.1: Vizualizace šesti barev, nahoře pozorováno člověkem bez zrakové vady a dole člověkem, který trpí deuteranopií. Převzato z [31].

Obecné rozvržení dashboardu

Uspořádání vizuálních prvků na dashboardu ovlivňuje, jak uživatelé prochází dashboard a zpracovávají informace. Jasné a logické rozvržení umožňuje uživatelům snadněji porozumět a rychleji reagovat na data. Důležité informace by měly být umístěny na prominentních místech (nahoře vlevo, nebo uprostřed), což usnadňuje jejich rychlé nalezení. Důležité je také zohlednit vizuální hierarchii, aby uživatelé mohli snadno rozlišit mezi klíčovými informacemi a sekundárními daty.

Předpozornostní zpracování (preattentive processing)

Správné využití předpozornostních vlastností může výrazně zkrátit čas potřebný k orientaci na dashboardu [8]. Předpozornostní zpracování umožňuje uživatelům rychle rozpoznat vzory a významné změny bez nutnosti věnovat pozornost každému jednotlivému prvku. Správné využití těchto vlastností, jako jsou barva, tvar a prostorová pozice, může zefektivnit proces smyslového zpracování a podpořit efektivní analýzu dat.

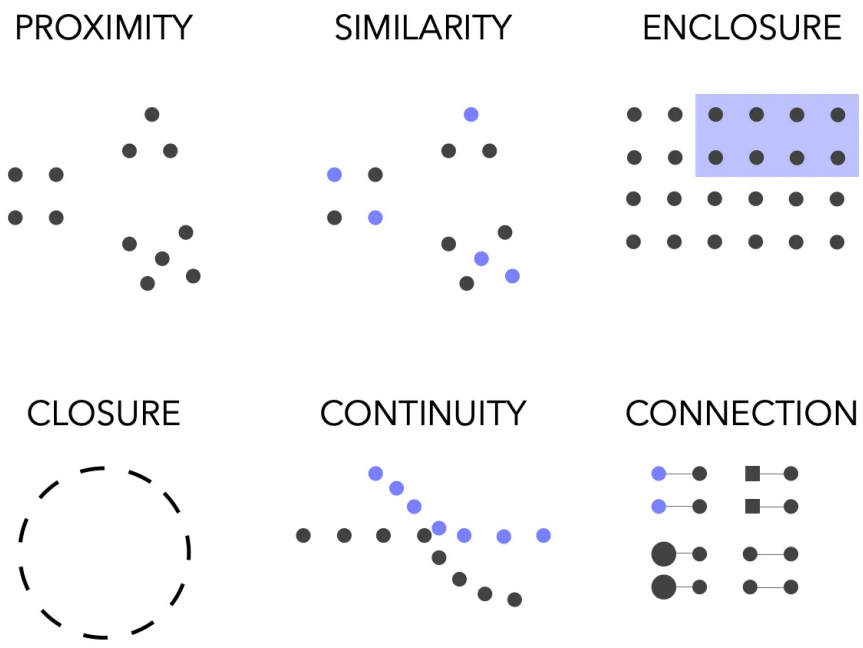
Obrázek 3.2: Srovnání klasického zpracování (vlevo) a předpozornostního (vpravo)

Smyslové utváření (Sensemaking)

Smyslové utváření je proces, při kterém pozorovatel reaguje na vizuální objekty, organizuje je a přidává jim význam. Baker a ostatní [5] tento proces definují jako *schopnost chápat komplexní informace, asimilovat je, vytvářet z nich pořádek a vyvinout mentální model situace jako předpoklad pro reakci na danou situaci.* Důležitým aspektem tohoto procesu je skutečnost, že pouze zlomek toho, na co se divák zaměřuje, je také objektem jeho pozornosti. Tento fakt odpovídá omezené kapacitě krátkodobé paměti, která uchovává objekty aktuálního zaměření pozornosti. Podle [8] se velikost krátkodobé paměti pohybuje mezi 3 a 9 položkami, ale existují i různé jiné interpretace, např. 3 – 5 položek podle [19].

Tvarová psychologie

Tvarová (gestalt) psychologie [20, 21] se zabývá tím, jak lidé vnímají a organizují vizuální informace. V rámci této psychologie byly formulovány zákony, které vysvětlují, jakým způsobem diváci uspořádávají objekty ve svém zorném poli. Zákon jednoduchosti stanovuje, že diváci interpretují vizuální informace v co nejjednodušší podobě. Zákon figura/pozadí se týká schopnosti oddělit figuru od pozadí, což umožňuje soustředit se na klíčové objekty. Zákon blízkosti popisuje tendenci diváků seskupovat objekty, které se nacházejí blízko sebe, zatímco zákon podobnosti určuje, že diváci seskupují objekty na základě jejich podobnosti, jako je barva nebo tvar. Zákon ohraničení říká, že objekty ohraničené rámečkem jsou vnímány jako celky, a zákon uzavření popisuje tendenci dokončovat neúplné objekty. Zákon kontinuity stanovuje, že diváci preferují objekty uspořádané v jednom směru za sebou, zatímco zákon spojení říká, že propojené objekty jsou vnímány jako související. Konečně, zákon symetrie ukazuje, že lidé mají tendenci preferovat symetrické objekty, které působí harmonicky a esteticky přívětivě.



Obrázek 3.3: Příklady tvarové (gestalt) psychologie. Převzato z [22].

3.3 Prvky dashboardu

Dashboard musí být sestaven pomocí vhodně zvolených a navržených zobrazovacích prostředků, aby bylo dosaženo jejich jedinečného potenciálu pro jasnou a okamžitou komunikaci [8]. Nejlepší způsob zobrazení dat vždy závisí na povaze informací, sdělení a potřebách cílové skupiny. Jediný dashboard obvykle zobrazuje různorodá data a vyžaduje několik typů zobrazovacích médií, přičemž každé médium je přiřazeno konkrétnímu typu dat.

Otázka, zda informace mají být zobrazeny jako text, grafika, nebo obojí, je velmi důležitá a ne vždy má jednoduchou odpověď. Text poskytuje přesnost, což je důležité zejména při sdělování kvantitativních údajů. Například při komunikaci aktuálních výdajů, jako je částka 487,321 Kč, je textové zobrazení plně dostačující, protože čísla jsou jednoznačná a jasně interpretovatelná. V tomto případě není nutné používat složité vizuální prvky, protože samotné zobrazení číselné hodnoty poskytuje všechny potřebné informace. Pro uživatele je okamžitě zřejmá hodnota, a to bez rizika nejednoznačnosti nebo nepřesného vnímání, které by mohly vzniknout u vizuálních prvků. Na druhou stranu, grafika přidává hodnotu při potřebě vizualizovat vztahy a trendy, které by byly v čistě textové podobě snadno přehlédnuty.

Efektivní dashboardy kombinují text a grafiku tak, aby poskytovaly smysluplné zobrazení dat s požadovanou mírou přesnosti a rychlostí vnímání. Každé měřítko nebo sada souvisejících měřítek by měla být prezentována tak, aby byla zajištěna co nejefektivnější komunikace, což vyžaduje správné sladění textu a grafických prvků.

Po výběru způsobu reprezentace dat je dalším krokem rozhodnutí, jak informace vizuálně organizovat. Je zásadní zvolit správné zobrazení, protože špatně navržený graf může zakrýt i jasně čitelná data. Na dashboardech se často používají známá zobrazovací média, jako jsou kvantitativní a jiné grafy, které při správném navržení mohou efektivně prezen-

tovat data i v malém prostoru. Při navrhování je důležité, aby zvolené médium co nejlépe odpovídalo typu zobrazovaných informací a zůstalo přehledné a čitelné. Mezi hlavní typy zobrazovacích prvků, které uvádí Few [8], patří grafy, obrázky, ikony, grafické objekty (drawing objects), text a organizační prvky, přičemž každé z těchto médií slouží jinému účelu na dashboardu a podporuje konkrétní druh komunikace s uživatelem.

Grafy

Vzhledem k převaze kvantitativních dat na dashboardech, není překvapivé, že většina používaných zobrazovacích médií spadá právě do kategorie grafů. Few [8], Bach a ostatní [4] uvádí několik kategorií grafů, které vychází z běžně používaných, ale některé nemusí být pro mnohé tolik známé:

Sloupcové grafy (horizontální a vertikální): Tento typ grafu umožňuje snadno porovnat hodnoty mezi různými kategoriemi pomocí délky obdélníkových sloupců. Například mohou být využity v obchodním dashboardu pro porovnání prodejů jednotlivých produktů během měsíce, přičemž delší sloupec indikuje vyšší prodej.

Složené sloupcové grafy (horizontální a vertikální): Umožňují zobrazit nejen celkovou hodnotu v jednotlivých kategoriích, ale i její složení. Například v rozpočtovém přehledu lze vidět celkové výdaje rozdělené na jednotlivé položky (jako nájem, marketing a platy) a jejich podíl na celkové částce.

Bodové grafy: Zobrazují vztah mezi dvěma kvantitativními proměnnými. Například v dashboardu pro správu skladu mohou body znázorňovat závislost mezi počtem objednávek a dobou doručení, což pomůže identifikovat vzory, jako je nárůst objednávek při delších časech doručení.

Spojité grafy: Spojité grafy jsou ideální pro znázornění změn v datech v průběhu času. Například ve výrobním prostředí mohou ukazovat nárůst nebo pokles produkce v závislosti na čase, což umožňuje sledovat sezónní trendy.

Kombinované sloupcové a spojité grafy: Kombinací sloupců a spojnic mohou tyto grafy porovnat rozdílné typy dat. Například sloupce mohou znázorňovat měsíční prodeje, zatímco spojnice ukazuje celkový trend, což umožňuje přehled o současném výkonu a jeho vývoji.

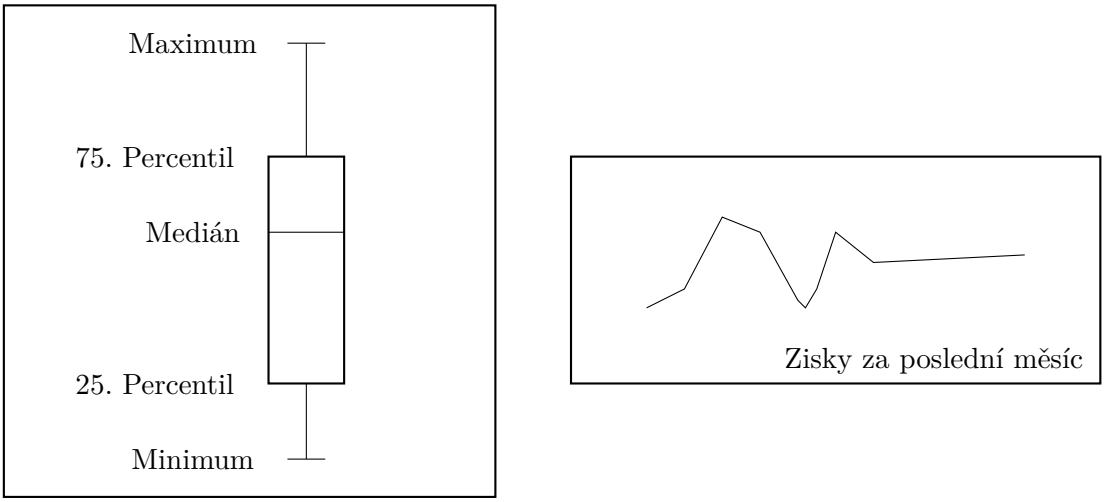
Stromové mapy: Využívají se k zobrazení hierarchických dat, například ve finančních dashboardech pro vizualizaci podílů jednotlivých položek na celkovém rozpočtu. Rozměr nebo barva oblastí znázorňují, jak velký podíl má každá položka v rámci celku.

Sparklines: V jiných publikacích také nazývány jako *minigrafy* [4], jsou kompaktní grafy vhodné tam, kde je potřeba ukázat trend v malém prostoru, jako například vedle tabulky s finančními údaji. Mohou ukázat vývoj akcií v průběhu krátkého časového úseku bez přetížení vizuálních informací.

Krabicový graf (Boxplot): Slouží k zobrazení rozložení dat a identifikaci extrémních hodnot. Například v analýze zákaznické spokojenosti může krabicový graf odhalit variabilitu v hodnocení různých produktů a zvýraznit extrémně negativní nebo pozitivní zpětnou vazbu.

Bullet grafy: Tento typ grafu je vhodný pro sledování pokroku vůči stanovenému cíli. V HR (lidské zdroje) dashboardu lze například zobrazit, jak blízko je tým k dosažení měsíčního cíle v počtu nových zaměstnanců, přičemž barevné zóny ukazují, jaký pokrok byl učiněn vůči očekávané hodnotě.

Gauge grafy: Použití tohoto typu grafu je trochu sporné. Few [8] ve své publikaci tvrdí, že jejich používání není vhodné, protože zabírají hodně místa na stránce. Kruhový tvar těchto grafů není praktický a místo jednoho kruhového grafu by prostor mohlo zabírat několik menších, například sloupcových nebo bullet grafů. Naopak publikace [4] gauge grafy doporučuje a používá. V kontextu této práce je použití gauge grafů efektivní a myslím si i v některých aspektech lepší než jiné dříve probírané typy. Z pohledu uživatele, tedy mechanika, bude určitě lepší a přirozenější číst některé hodnoty z měřáků (gauges), než například ze spojitého grafu.



Obrázek 3.4: Krabicový graf (vlevo) a Sparkline (vpravo)

Ikony

Ikony jsou jednoduché obrázky, které mají jasný a srozumitelný význam. Na dashboardu jich obvykle potřebujeme jen několik, přičemž užitečné ikony vyjadřují například: upozornění, růst nebo pokles, zapnuto nebo vypnuto.

Ikony upozornění informují o důležité informaci, zejména když je něco špatně a vyžaduje to pozornost. Aby ikona efektivně plnila tuto roli, měla by být jednoduchá a výrazná. Je lepší omezit úrovně výstrah na maximálně dvě a ideálně na jednu [8], protože jednoduchá ikona přitahuje pozornost mnohem účinněji než vícero různých výstrah. Populární schéma výstrah používá metaforu semaforu, ale pro barvoslepé osoby to může být problém, jak už jsem zmiňoval u správného výběru barev na obrázku 3.1. Lepším řešením je používat různé intenzity stejných barev.

Ikony růst a pokles vyjadřují jednoduchou zprávu, že hodnota vzrostla nebo poklesla ve srovnání s nějakým minulým bodem. Tyto ikony se běžně používají ve finančních dashboardech, například pro indikaci pohybu akcií nebo zisků. Standardním symbolem pro tyto účely je trojúhelník nebo šipka směřující nahoru nebo dolů. Pro lepší srozumitelnost by měly být barvy ikon výrazně odlišné, aby je i barvoslepí lidé rozpoznali.

Ikony zapnuto nebo vypnuto slouží k označení položek, které se od ostatních liší. Například při zobrazení podaných žádostí mohou tyto ikony efektivně signalizovat, které z nich jsou schváleny. Mezi běžně používané ikony patří značka zaškrtnutí, hvězdička a nebo křížek. Důležité je zvolit jednu ikonu a držet se jí pro zajištění konzistence, což zvyšuje přehlednost dashboardu.

Text

Na každém dashboardu, bez ohledu na jeho grafické zaměření, by měly být přítomny také textové informace. Text je nezbytný pro popisné štítky a může být užitečný i jinde, například pro zobrazení jediného měření bez porovnání. V takových případech text jasně a efektivně vyjadřuje údaj a je často lepší než grafické zobrazení.

Obrázky a videa

Zobrazování obrázků, jako jsou fotografie nebo diagramy, může být někdy užitečné, ale běžně je to zřídka nutné. Dashboardy pro specifické účely, například školení nebo vizualizaci nějakého postupu, mohou obsahovat obrázky pro zobrazení důležitých informací. Většinou ale použití obrázků rozptyluje uživatele a má za důsledek nepřehlednost celého dashboardu. V některých případech je ale jejich využití vhodné. Ještě spolu s videi mohou například sloužit pro zobrazení bezpečnostních kamer v dashboardu bezpečnostního systému.

Grafické objekty

Grafické objekty (angl. drawing objects) jsou užitečné pro uspořádání a propojení informací tak, aby byly jasné pro uživatele dashboardu. Například při zobrazení informací o nějakém procesu nebo postupu může být užitečné uspořádat jednotlivé události sekvenčně za sebou a ukázat cestu, kterou proces nebo postup prochází. Kružnice a obdélníky, nebo jiné tvary, mohou být efektivně využity k reprezentaci různých úkolů, zatímco čáry a šipky mohou znázorňovat vztahy a tok mezi nimi. Tento vizuální přístup usnadňuje pochopení hierarchických a procesních vztahů v systému. Například v rámci workflow diagramů mohou kruhy nebo obdélníky představovat jednotlivé kroky v procesu a šipky znázorňovat tok činností, což zajišťuje jasné sledování postupu. Podobně v Petriho sítích mohou být kružnice využity pro reprezentaci stavů a obdélníky pro události nebo přechody, přičemž čáry mezi nimi ilustrují vztah mezi stavy a akcemi, které mění systém.

Organizátory

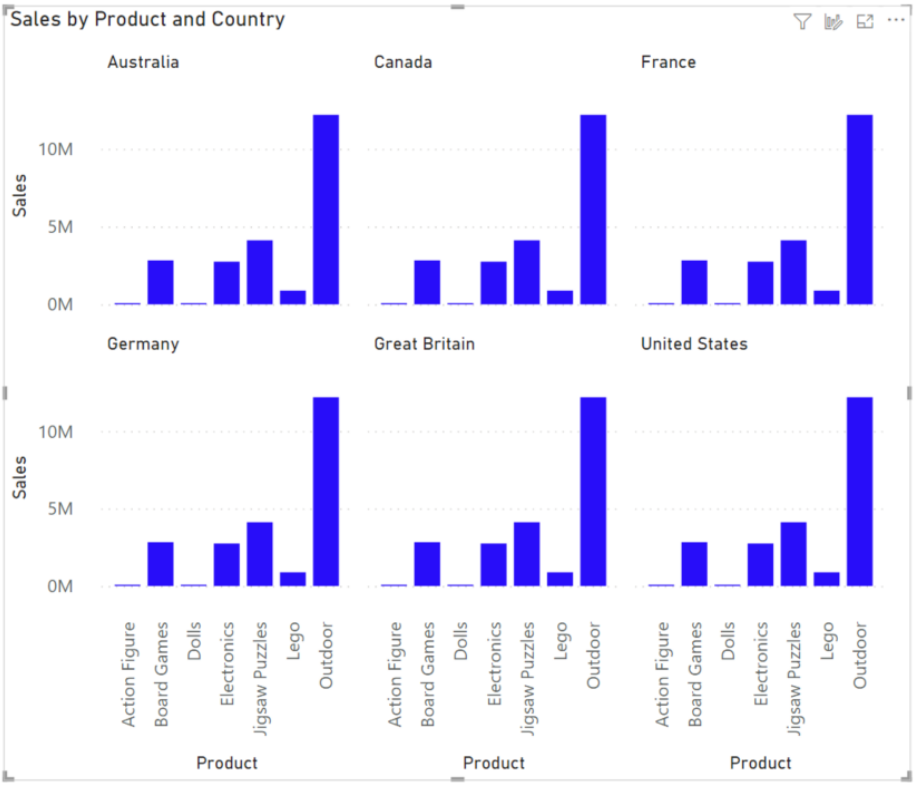
Informace je často potřeba uspořádat určitým způsobem, aby byly jasně komunikovány. Tři zvláštní způsoby organizace a uspořádání souvisejících informací, které se ukazují jako užitečné na dashboardech, zahrnují [8]: tabulky, prostorové mapy a malé násobky (small multiples) [38].

Tabulky organizují data do sloupců a řádků. Tento známý formát uspořádání je efektivní pro text, ale také může sloužit k uspořádání grafů, ikon a obrázků. Uložení různých typů dat do tabulkového formátu zjednodušuje srovnání a analýzu informací. Tabulky jsou užitečné, když je třeba prezentovat více souvisejících dat najednou.

Prostorové mapy představují specializovanou formu organizace dat, která spojuje informace s fyzickým prostorem. Umožňují zobrazit data v geografickém uspořádání, což je užitečné, když je důležité vidět, jak jsou měření umístěna v reálném světě. Příkladem je zobrazení zisků v různých regionech na mapě, což může odhalit vzory, které by jinak nebyly zřejmé.

Malé násobky (small multiples), jak je nazývá Edward Tufte, uspořádávají grafy do tabulkového formátu, skládajícího se z řady nebo sloupce souvisejících grafů. Například pokud je potřeba zobrazit zisky rozdělené podle regionů, malé násobky umožňují umístit více verzí stejného grafu vedle sebe, což usnadňuje srovnání. Efektivní organizátor pro malé

násobky by měl umožnit uživatelům vybrat proměnné, které se zobrazí na osách grafu, a automaticky uspořádat grafy do požadovaného formátu.



Obrázek 3.5: Malé násobky (small multiples)[1]

3.4 Správný návrh dashboardu

Při návrhu dashboardu není klíčové pouze vybrat vhodná vizualizační média pro zobrazení dat, ale rovněž je potřeba zaměřit se na celkový design a strukturu. Aby byl dashboard užitečný a snadno použitelný, musí splňovat specifické požadavky na přehlednost a efektivní zpracování informací. To znamená, že důraz musí být kladen nejen na to, jak data zobrazit, ale také jakým způsobem je organizovat, aby uživatelé mohli rychle a přesně interpretovat klíčové informace [8].

Jedním z hlavních principů návrhu dashboardu je správná organizace informací podle jejich významu a použití. Informace nemohou být náhodně rozmístěny. Je nutné, aby data byla seskupena na základě jejich funkce nebo způsobu použití. To znamená, že informace, které spolu souvisí nebo které slouží stejnému účelu, by měly být umístěny společně. Například data o výkonnosti jednotlivých oddělení by měla být seskupena tak, aby bylo snadné porovnávat je mezi sebou. Kromě toho, aby uživatelé nebyli vizuálně zahlceni, je důležité, aby byly jednotlivé skupiny dat jasně odděleny, a to pomocí jednoduchých vizuálních prostředků, jako je bílý prostor nebo tenké linky. Tyto vizuální prvky přispívají k lepší pře-

[1] Zdroj: https://learn.microsoft.com/en-us/power-bi/visuals/power-bi-visualization-small-multiples

hlednosti bez nadbytečného zdůraznění, které by mohlo odvádět pozornost od podstatných informací.

Dalším důležitým aspektem správného návrhu dashboardu je podpora smysluplných porovnání dat. Data by měla být prezentována způsobem, který umožňuje uživatelům snadno srovnávat různé informace a odhalovat vzorce či trendy. Například využití grafů, které jasně zobrazují změny v čase, nebo tabulek, které zjednodušují porovnání mezi různými kategoriemi, pomáhá uživatelům lépe pochopit kontext a provádět rozhodnutí na základě prezentovaných informací. Na druhou stranu by měl být kladen důraz i na to, aby nedocházelo k nesmyslným porovnáním. Například špatně navržené barevné schéma nebo nevhodné rozložení prvků může vést k chybným závěrům, pokud uživatelé omylem srovnávají nesouvisející informace.

Konzistence je dalším klíčovým faktorem při návrhu dashboardu [4]. Všechny prvky, které zobrazují stejné typy dat, by měly být vizuálně zpracovány stejně. Pokud jsou různé typy informací prezentovány odlišně, může to vést k nejasnostem a zpomalit práci uživatelů. Například časové řady by měly být v rámci celého dashboardu zobrazovány stejným typem grafu, aby uživatelé nemuseli přizpůsobovat své chápání dat na základě změn vizualizace. Konzistentní design prvků nejen usnadňuje práci s dashboardem, ale také minimalizuje riziko chyb při interpretaci dat.

Vedle funkčnosti je důležitý také estetický vzhled dashboardu [18]. Nejenže musí být vizuálně přitažlivý, ale musí zároveň poskytovat příjemný uživatelský zážitek a ne uživatele odrazovat. Přestože estetika by neměla převažovat nad funkčností, dobře navržený a vizuálně přitažlivý dashboard může přispět k tomu, že uživatelé budou s daty pracovat efektivněji a s menší frustrací. Při výběru barev se doporučuje používat tlumené tóny, které tolik nezatěžují oči, a jasnější barvy nechat pouze pro zvýraznění důležitých prvků. Kromě toho by měla být zajištěna vysoká kvalita zobrazených grafů a obrazů, aby byly čitelné a přehledné i na různých typech displejů. Důležitou roli hraje také výběr fontu – ten by měl být snadno čitelný a měl by být konzistentně používán napříč celým dashboardem.

Interaktivita je další nezbytnou vlastností moderních dashboardů. V mnoha případech dashboard slouží jako výchozí bod pro hlubší analýzu dat, a proto by měl umožňovat uživateli interakci s daty. To zahrnuje možnost filtrování informací, zobrazování detailů kliknutím na jednotlivé prvky nebo provádění dalších akcí, které jsou pro uživatele intuitivní. Interaktivní prvky by měly být navrženy tak, aby byly snadno přístupné a konzistentní ve své funkčnosti.

Před samotným spuštěním by měl být dashboard důkladně testován z hlediska použitelnosti. Prototyp dashboardu by měl být předložen uživatelům, aby bylo možné zjistit, zda splňuje jejich očekávání a potřeby. Testování použitelnosti je důležité pro identifikaci případných problémů s navigací nebo porozuměním dat a také pro zjištění, že interaktivní prvky fungují podle očekávání.

3.5 Klíčové ukazatele výkonnosti (KPIs)

Jak uvádí Few [8], v 90. letech 20. století se objevila nová metodika řízení, která zahrnovala identifikaci a využívání klíčových ukazatelů výkonnosti (KPI). Tato metodika byla představena Robertem S. Kaplanem a Davidem P. Nortonem v rámci konceptu Balanced Scorecard. To zapříčinilo velké rozšíření a zájem o systémy pro zobrazení výkonnosti, které umožňují organizacím efektivně řídit a optimalizovat své činnosti.

Koncept řízení výkonnosti je nezbytným nástrojem pro organizace, které chtějí zajistit, že se ubírají správným směrem k dosažení svých cílů a strategií [17]. Pro efektivní řízení

výkonnosti je klíčové, aby organizace měly jasné a měřitelné ukazatele výkonnosti, které jim umožňují sledovat a hodnotit jejich činnosti.

Ukazatele výkonnosti jsou specifické metriky (fyzické hodnoty), které slouží k měření a porovnání celkové výkonnosti organizace [12]. Klíčové ukazatele výkonnosti jsou poté ukazatele výkonnosti, které jsou důležité (klíčové) pro úspěšnost a dosažení určitých cílů organizace (podmnožina ukazatelů výkonnosti). Tyto ukazatele mohou zahrnovat různé aspekty, jako je kvalita produktů, náklady, finanční ukazatele, flexibilita, spolehlivost dodání, spokojenost zákazníků a zaměstnanců, bezpečnost, environmentální aspekty a možnosti učení a růstu [17]. Klíčové ukazatele výkonnosti však lze využít i jinak než jen na sledování výkonnosti organizace, jak ukazuje obrázek 3.6.

Podle D. Parmentera [32] se tyto ukazatele vyznačují několika specifickými vlastnostmi: jsou nefinanční (není jim přiřazena měna), měří se často (ideálně denně nebo týdně) a jsou zaměřeny na vrcholný management, který na jejich základě přijímá rozhodnutí. KPI by měly být jednoduché a srozumitelné pro všechny zaměstnance organizace a zároveň by měly mít jednoznačně přiřazenou odpovědnost konkrétnímu týmu, který zajišťuje případná nápravná opatření. Kromě toho mají KPI značný dopad na kritické faktory úspěchu organizace (Critical Success Factors, CSFs) a celkový úspěch organizace. Současně by měly být tyto ukazatele bezpečné, což znamená, že byly otestovány s cílem minimalizovat jejich potenciální negativní dopady a nezamýšlené důsledky, které by mohly vést k nežádoucím výsledkům.

Důležité je také mít na paměti, že mezi jednotlivými ukazateli výkonnosti může docházet ke vzájemnému ovlivňování. Například zvýšení jedné hodnoty může vést ke snížení jiné. Tato komplexnost vyžaduje, aby organizace pečlivě zvažovaly, jak jednotlivé ukazatele ovlivňují celkovou výkonnost a jak jsou vzájemně propojeny.

Obrázek 3.6: Příklad vizualizace vyhodnocení KPI z platformy ACADA (Logimic [26]), převzato z bakalářské práce [7]

Kapitola 4

Analýza

Analýza požadavků na systém byla založena na řízeném rozhovoru s vedoucím servisního oddělení pobočky společnosti CBA v Šumperku, která by o takový systém měla zájem. Společnost CBA je sdružení velkoobchodů a maloobchodních jednotek, které poskytují především prodej potravin. Společnost také nabízí catering a řeší logistiku spojenou s dovozem zboží. Zákazník se pohybuje v oboru automobilového průmyslu a tedy dobře rozumí potřebám spojeným s diagnostikou a monitorováním vozidel. Během rozhovoru byly identifikovány klíčové funkce, které by systém měl splňovat, aby vyhovoval požadavkům reálného provozu a efektivně podporoval každodenní činnosti zákazníka.

4.1 Cílová skupina

Cílovou skupinou tohoto systému jsou lidé odpovědní za opravu a správu silničních vozidel, ať už jde o soukromé majitele nebo komerční subjekty, jako jsou logistické a servisní společnosti. Mezi hlavní uživatele patří především servisní oddělení logistických firem a správci vozových parků, kteří potřebují mít neustálý přehled o technickém stavu svých vozidel. Klíčovým požadavkem pro tyto uživatele je možnost monitorovat stav flotily, předcházet problémům a minimalizovat náklady na opravy i neplánované odstávky. Je požadováno, aby systém umožnil efektivně sledovat velké množství vozidel z jednoho zařízení, což následně usnadňuje organizaci údržby a zvyšuje spolehlivost provozu celé flotily.

Při opravě vozidel často dochází k problému, že osoba servisující vozidlo nemá možnost se podívat zpátky do historie a zkontrolovat, co se přesně při jízdě s vozidlem stalo nebo před jakou dobou se daný problém vyskytl. Bylo by tedy vhodné, aby systém umožnil zaznamenávat a uchovávat data o sledovaném vozidle několik dní či týdnu zpět. Osoba, která vozidlo opravuje, by si pak jednoduše mohla najít jaký přesný stav vozidla byl v momentě, kdy se daná závada vyskytla, případně před či hned po výskytu problému.

4.2 Proces

První probíraným tématem rozhovoru byl postup, jak uživatel bude se systém pracovat a jak ho bude využívat. Uživatel, v našem případě nějaký mechanik nebo správce vozidla, umístí do všech vozidel zařízení pro získání dat. Vozidla pak budou vykonávat cesty a převážet zboží. Pokud se vyskytne problém s vozidlem, nebo se bude chtít mechanik podívat na jeho stav, tak si na svém osobním počítači otevře aplikaci pro přístup do systému a zobrazí si

vše potřebné z jakéhokoliv vozidla. Podle historie stavu jednotlivých součástí vozidla pak může odhalit a opravit nějaký problém nebo závadu daného vozidla.

4.3 Požadavky

Dalším probíraným tématem byly požadavky na výsledný systém. Hlavním požadavkem na systém je možnost sledovat a kontrolovat velké množství vozidel. Jedná se totiž o poměrně rozsáhlou flotilu a mít možnost všechna vozidla spravovat pomocí jednoho systému je velké ulehčení práce. Jinak by totiž mechanik musel při každé kontrole nastoupit do každého vozidla a kontrolovat jeho stav pomocí specializovaného diagnostického nástroje přímo z jeho kabiny. Takto bude moci například přes svůj osobní počítač zkontrolovat všechna vozidla, která jsou součástí systému a nebude muset do každého fyzicky nastoupit.

Je tedy požadováno, aby systém obsahoval dva hlavní typy zařízení. Prvním typem by byla zařízení pro ukládání (logování) dat a druhým zařízení pro zobrazení a analýzu dat. Zařízení pro ukládání dat by bylo umístěno ve vozidle a získávalo by data v průběhu jízdy a zařízením pro vizualizaci by byl například osobní počítač daného mechanika a zobrazoval by data z vozidla uživateli. Zařízení pro logování dat by bylo samozřejmě mnohem více (tolik kolik vozidel flotily) než zařízení pro vizualizaci (stačilo by teoreticky jedno).

Na úrovni zařízení pro logování dat by bylo požadováno agregovat data z jednotky pomocí diagnostického adaptéru (například ELM 327) s použitím rozhraní OBD II. Bylo by také potřeba použít nějaké zařízení, které bude mít malou spotřebu energie, a to umožní použít ho v automobilu a napájet z jeho baterie. Je požadováno, aby byl interval pro získávání dat vhodně zvolen, aby umožnil získat všechna relevantní data a nedošlo k vynechání (nezaznamenání) něčeho důležitého. Na druhou stranu interval nesmí být zvolen moc malý, protože by mohlo dojít k případnému zaplnění paměti a velké množství dat by se hůře zpracovávalo. Dalším důležitým požadavkem na toto zařízení je relativně velká paměť, v řádech GB. Historie dat bude uchovávána několik dní zpátky a je tedy třeba v závislosti na zvoleném intervalu získávání dat správně zvolit velikost paměti pro jejich uložení.

Na úrovni zařízení pro vizualizaci dat je požadováno data uživateli nějakým způsobem efektivně zobrazit a to pomocí webové aplikace, která poskytne přístup k datům prostřednictvím přehledného a interaktivního dashboardu. Aplikace musí běžet na standardních kancelářských noteboocích a to přesně ve všech používaných webových prohlížečích. V aplikaci chce mít uživatel přístup ke všem svým spravovaným vozidlům a všem cestám, které v poslední době vykonala. Dále chce mít možnost si konfigurovat jaké informace na dashboardu u vozidla uvidí a případně měnit jak daná data zobrazit a jaká vizualizační média použít. Chtěl by tedy mít nějaké skupiny parametrů, které mu umožní si vybrat, co se bude momentálně zobrazovat a jakým způsobem. Kromě toho je požadováno zobrazovat nějakým způsobem uživateli trasy daného vozidla (například mapou) a možnost se pohybovat v čase (historii) pomocí časové osy.

4.4 Existující řešení

Pro návrh řešení byla provedena analýza existujících produktů na trhu, které poskytují podobné funkce v oblasti OBD II systémů. Zaměřuji se zde na oficiální produkty automobilek a jiných firem, ale i na „hobby“ řešení či open source projekty. Produktů v této oblasti existuje mnoho, společnost CBA dokonce jeden z níže zmíněných produktů již používá a to

od firmy Commander. Dále využívají i diagnostické služby od firmy Texa[1], ale ani jedno ze zmíněných řešení nesplňuje požadavky, které na diagnostický systém mají.

Obecně v oblasti OBD II systémů existuje mnoho komerčně dostupných zařízení, která nabízejí základní diagnostiku vozidla. Některé z těchto produktů jsou navrženy především pro koncové uživatele a zaměřují se na jednoduchost použití. Tyto systémy však často omezují uživatele na předem definovaný výběr parametrů a nejsou vždy vhodné pro specifické aplikace vyžadující přizpůsobení. Mnohá z těchto zařízení jsou univerzální a fungují na většině vozidel s OBD II portem. Jejich ceny se pohybují v řádech stovek až tisíců korun.

Do této kategorie patří například OBD II scanner Motopower MP69033[2]. Tento diagnostický nástroj patří do kategorie univerzálních a umožňuje čtení chybových kódů pro rychlé zjištění příčiny problémů, jejich následné odstranění, zobrazení živých dat a informací o vozidle. Má širokou kompatibilitu s většinou vozidel vyrobených po roce 1996 v USA, po roce 2000 v EU a v Asii, včetně některých motocyklů. Zařízení má snadné ovládání, je kompaktní a zobrazuje data na 2,8 palcovém LCD displeji. Je ideální pro domácí diagnostiku bez nutnosti návštěvy autoservisu.

Obrázek 4.1: Příklad vizualizací dat Motopower MP69033

Pokročilejší OBD II řešení, určená pro profesionální využití, nabízejí široké možnosti integrace, avšak bývají cenově náročná a vyžadují vyšší technické znalosti při instalaci a konfiguraci. Taková zařízení se pohybují už ve vyšší cenových kategoriích, často v řádech desetitisíců korun. V této kategorii se setkáme například se scannerem Launch X431 PRO 5[3] pro vozy BMW. Spadají sem také diagnostická zařízení specifická pro jednotlivé

[1]Oficiální stránky: https://www.texa.com

[2]Oficiální obchod: https://motopowers.com/products/motopower-mp69033-obd2-scanner-universal-car-engine-fault-code-reader-can-diagnostic-scan-tool-for-all-obd-ii-protocol-cars-since-1996

[3]Dostupné: https://www.diagstore.cz/launch-x431-pro-5-c

značky automobilů, jako Xentry Diagnosis Lite Pad[4] pro Mercedes Benz. Tento systém poskytuje diagnostiku pro všechny řídicí jednotky všech modelových řad Mercedes Benz od roku 2004, včetně základní diagnostiky, kódování, parametrizace, čtení/vymazání chybové paměti, funkční diagnostiky a programování klíčů. Kromě zakoupení samotného zařízení také vyžaduje časové předplatné pro použití služeb.

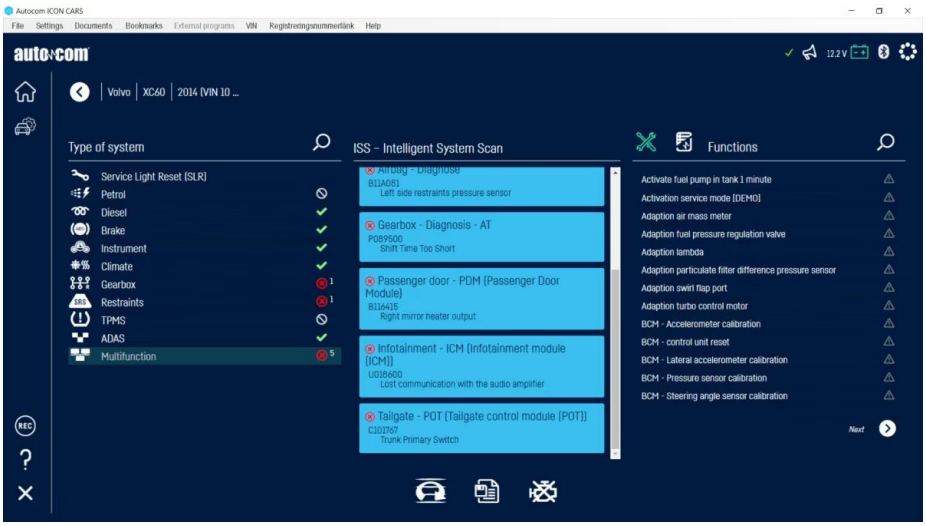
Obrázek 4.2: Zařízení Xentry Diagnosis Lite Pad

V oblasti OBD II systémů je důležité zmínit také diagnostické programy, které uživatelé využívají na běžných počítačích místo těch, jež jsou nainstalovány přímo na specializovaných diagnostických zařízeních. Tyto systémy určené pro stolní či přenosné počítače často poskytují širší možnosti analýzy a vizualizace dat. Tyto programy je ale potřeba na zařízení nainstalovat a konfigurovat, což může být pro některé uživatele složité.

V této kategorii diagnostických programů pro osobní počítače jsou oficiální programy od automobilek či jiných firem z oboru ale i amatérská řešení a projekty. Oficiální programy jsou většinou placené, ale často mnohem lépe zpracované než ty amatérské. Nabízí podporu pro většinu vozidel s OBD II portem a umožňují standardní diagnostiku, testování nebo kódování jednotek. Ceny těchto programů se pohybují v desítkách tisíc při jednorázové platbě, ale většina funguje formou předplatného. Patří mezi ně například Autocom CARS software[5].

[4]Dostupné: https://www.maverickdiagnostics.com/shop/oem-tools/xentry-diagnosis-lite-pad-tablet/
[5]Oficiální stránky: https://autocom.se/en/cars-software

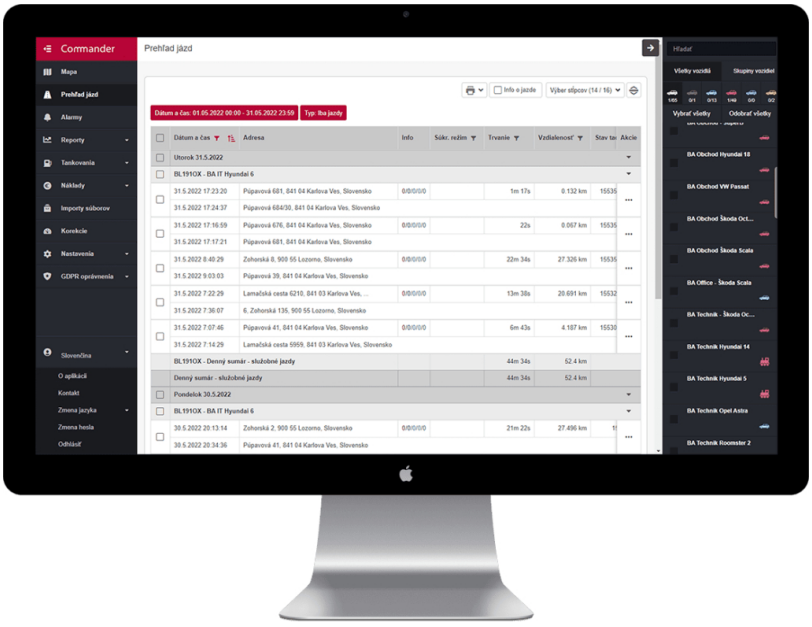
Autocom CARS software je diagnostický software určený pro profesionální servisy, který pokrývá širokou škálu vozidel, včetně osobních aut, lehkých nákladních vozidel a hybridních vozidel. S CARS je možno číst a mazat chybové kódy, provádět nastavení, aktivace a další funkce. Software obsahuje přístup k více než 40 000 systémům, což umožňuje diagnostiku většiny značek a typů vozidel. Funkce jako „System Scan“ umožňuje během několika minut prozkoumat všechny systémy ve vozidle a získat přehled o jeho aktuálním stavu, což šetří čas. Pro většinu OBD funkcí jsou k dispozici podrobné pokyny, které uživatelům usnadní diagnostiku i složitější úkoly. K dispozici je také funkce „Flight Recorder“ pro sledování problémů, které se vyskytují během jízdy, a možnost připojení k OBD pro provádění emisních diagnostik.



Obrázek 4.3: Autocom CARS software

Existují zde i programy pro specifické využití jako sledování polohy vozidla pomocí GPS, přístup k datům z jednotky na dálku nebo zabezpečení automobilů. Všechny tyto případy užití řeší například firma Commander[6]. Program od této firmy využívá momentálně právě společnost CBA pro správu svých vozidel. Umožňuje sledování vozidel pomocí GPS pro monitorování jejich pohybu a je dostupný pro různé typy vozidel, včetně osobních aut, dodávek, nákladních vozidel, autobusů, pracovních strojů, motocyklů a lodí. Systém umožňuje sledování aktuální polohy vozidel na mapě v reálném čase, registraci řidičů pomocí identifikačních klíčů, dále zaznamenává pohyby vozidel, výpočty jízdních parametrů a analýzu jízdního stylu řidičů. Také sleduje spotřebu, otáčky motoru, hladinu paliva a umožňuje propojením s tachografem kontrolu dodržování rychlosti.

[6]Oficiální stránky: https://commanderservices.eu/



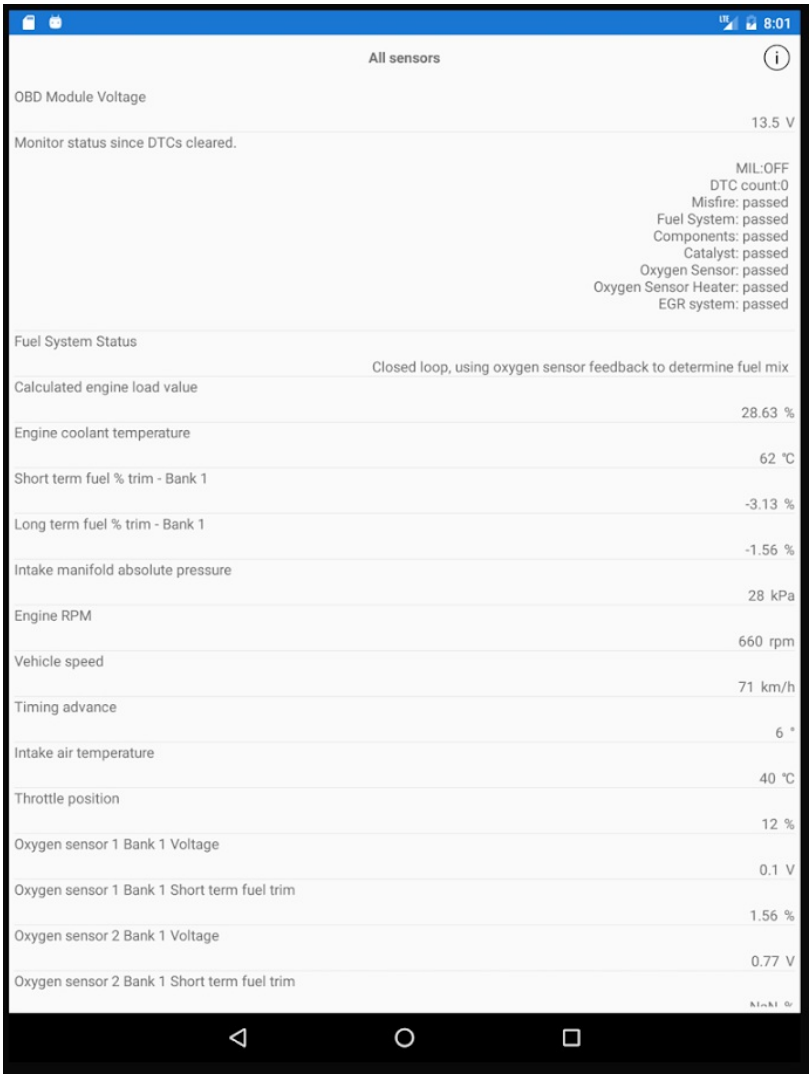
Obrázek 4.4: Monitorovací služby firmy Commander

Na druhou stranu amatérské programy nabízí levné nebo i bezplatné řešení pro někoho, kdo hledá program, který si chce sám konfigurovat a přizpůsobit. Existuje mnoho volně dostupných projektů, jako jsou telemetry-obd[7] nebo pyobd-pi[8]. Telemetry-obd je v mnoha aspektech stejné jako projekt popisovaný v této práci, kromě koncové vizualizace pomocí dashboardu.

Nakonec by bylo vhodné zmínit také mobilní aplikace pro diagnostiku vozidel. Tyto aplikace, dostupné pro chytré telefony a tablety, jsou většinou zdarma nebo za relativně nízkou cenu pohybující se v řádu desítek až stovek korun. Nabízejí uživatelům pohodlný přístup k základním diagnostickým funkcím, jako je čtení a mazání chybových kódů, sledování živých dat senzorů nebo kontrola emisního stavu vozidla. Nicméně jejich funkčnost bývá často omezená ve srovnání s profesionálními diagnostickými systémy, a to jak v hloubce analýzy, tak v podpoře specifických funkcí pro různé značky a modely automobilů. Tyto aplikace jsou však oblíbené mezi amatéry a majiteli vozidel, kteří potřebují rychlé a jednoduché řešení pro základní diagnostiku. Patří sem například Car Scanner ELM OBD II[9].

Jedná se o mobilní diagnostický nástroj pro vozidla, který využívá OBD II adaptér (WiFi, Bluetooth) k připojení k řídícímu systému vozidla. Nabízí širokou škálu funkcí, jako je zobrazení dat pomocí měřidel a grafů, čtení a resetování chybových kódů DTC, kontrolu připravenosti vozidla k emisní zkoušce a sledování senzorů. Aplikace podporuje vozidla kompatibilní s OBD II (většina vozidel od roku 2000, některé i od 1996) a obsahuje profily pro různé značky automobilů. Car Scanner je zdarma dostupný na Google Play, ale vyžaduje kompatibilní OBD II adaptér (ELM 327). Tento program jsem měl možnost i vyzkoušet a pro jednoduché činnosti je za mě dostačující.

[7] Github: https://github.com/thatlarrypearson/telemetry-obd
[8] Github: https://github.com/Pbartek/pyobd-pi
[9] Car Scanner: https://play.google.com/store/apps/details?id=com.ovz.carscanner



Obrázek 4.5: Car Scanner ELM OBD II

Když vše shrnu, po důkladné analýze dostupných konkurenčních řešení lze konstatovat, že aplikace popisovaná v této práci je opravdu jedinečná. Na trhu sice existují podobné aplikace, které pokrývají určité dílčí funkce, avšak žádné z nich neposkytují kompletní soubor funkcionalit či přístup, které tento systém nabízí. Tím se výrazně odlišuje od stávající konkurence a představuje jiný, můžu říci něčím i inovativní přístup k řešení dané problematiky.

4.5 Využití RIoT

Požadavkem na tuto aplikaci je její integrace do již existujícího RIoT systému. Tato aplikace bude tedy postavená nad RIoT backendem a bude ho využívat pro zpracování a získání dat pro vizualizaci. Bude však využívat pouze backend tohoto systému a frontend se v rámci této práce vytvoří kompletně nový.

RIoT je informační systém pro zpracování IoT dat v reálném čase a monitorování stavu na základě uživatelsky definovaných kritérií (KPIs) [7]. Architektura tohoto konkrétního

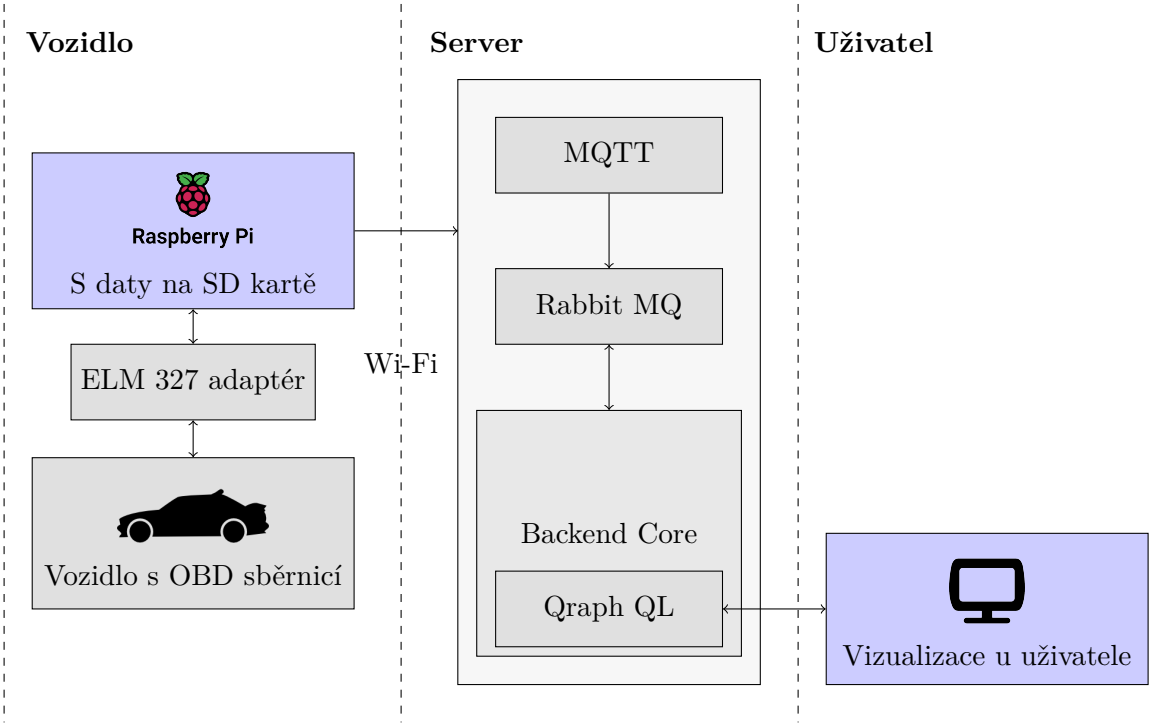
použitého RIoT systému je tříúrovňová. První úroveň, frontend, představuje klientskou stranu, která umožňuje interakci uživatelů se systémem. Uživatelé mohou prostřednictvím rozhraní odesílat požadavky na backend a zobrazovat výsledky zpracování dat. Na backendu probíhá samotné zpracování dat a vyhodnocování klíčových indikátorů výkonnosti. Backend je rozdělen na několik mikroslužeb, což umožňuje lepší škálovatelnost a flexibilitu při růstu požadavků.

První služba v rámci backendu je MQTT Preprocessor, který zajišťuje příjem a dekódování příchozích MQTT zpráv. Na základě těchto zpráv generuje interní zprávy, které jsou dále zpracovávány v systému. Tato služba je klíčová pro komunikaci s IoT zařízeními, které odesílají data do systému. Druhou službou je Message Processing Unit, která provádí samotné zpracování dat, vyhodnocování klíčových indikátorů výkonnosti a generování dalších systémových zpráv na základě zadaných požadavků. Třetí složkou backendu je Backend Core, který se stará o příjem výsledků zpracování dat, propagaci nových informací na frontend a správu relační databáze. Tento modul je zodpovědný za provádění CRUD operací, tedy tvorbu, čtení, aktualizaci a mazání dat, a zajištění konzistence dat v systému. Všechny části (služby) spolu komunikují pomocí technologie RabbitMQ.

Kapitola 5

Návrh

Na základě provedené analýzy je navržena integrace dané funkcionality (aplikace), která umožňuje průběžné získávání dat z OBD II sběrnice, jejich lokální ukládání a následné odesílání na server. Ze serveru budou data vizualizována prostřednictvím webové aplikace v prohlížeči počítače. Pro připojení k OBD II portu vozidla bude využíván ELM 327 adaptér, nebo jiný adaptér kompatibilní s OBD II sběrnicí. Pro sběr a odesílání dat se využije nějaké zařízení s malou spotřebou energie a možností komunikace přes Wi-Fi, jako je například Raspberry Pi. Data se průběžně ukládají do lokálního úložiště (SD karta). Tato celá část se bude odehrávat ve vozidle (viz. obrázek 5.1). V dosahu Wi-Fi se data odešlou na server, který se skládá z více RIoT projektů. Jeden z nich bude zpracovávat právě tato získaná data pro vizualizaci popisované aplikace. Klientská aplikace pak bude tato data zobrazovat přehledně pomocí dashboardu a využívat grafy a jiná zobrazovací média.



Obrázek 5.1: Schéma systému pro zpracování a vizualizaci dat z OBD II

5.1 Návrh na úrovni vozidla

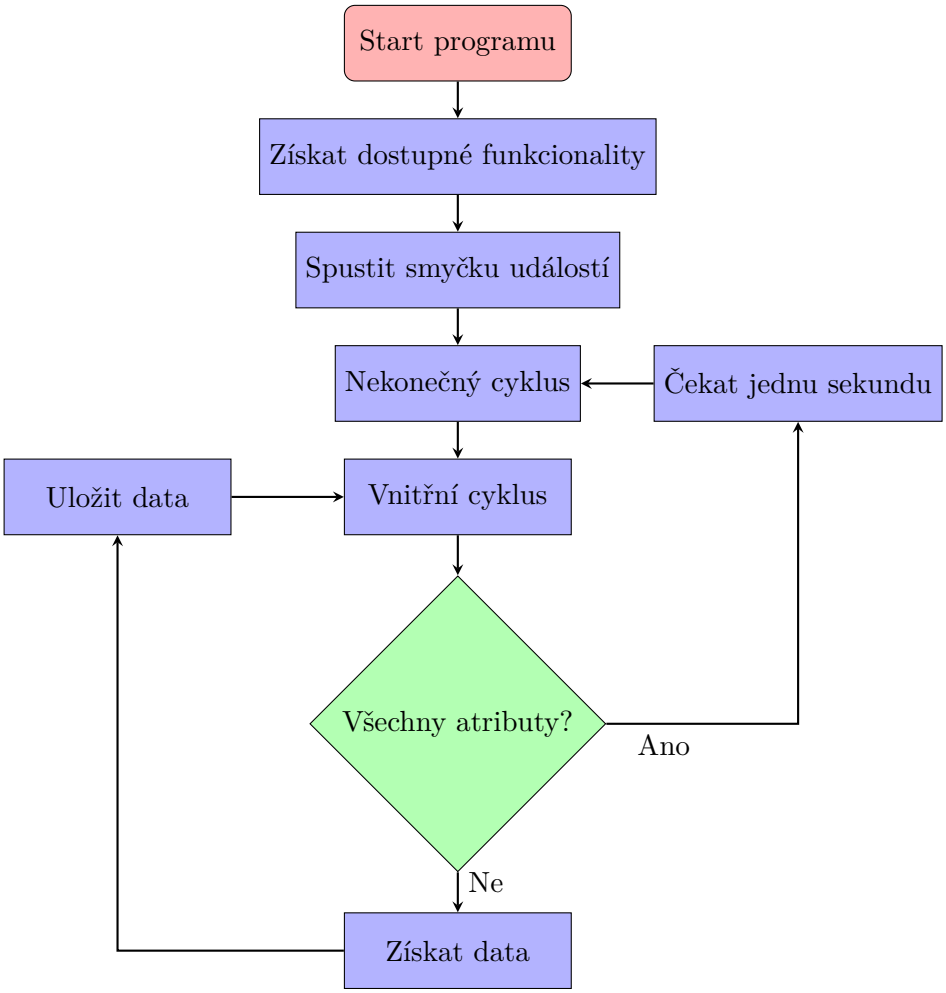
První část systému se soustředí na sběr dat z OBD II portu vozidla a jejich předběžné zpracování a odesílaní na server. Hlavním hardwarovým komponentem je zařízení Raspberry Pi, které bylo zvoleno pro svou dostatečnou výpočetní kapacitu, malou spotřebu energie, možnost připojení k internetu pomocí Wi-Fi a k OBD adaptéru pomocí Bluetooth. Raspberry Pi nedisponuje interní pamětí, proto je potřeba využít SD kartu pro uložení kódu programu a dat získaných z vozidla. Alternativně by bylo možné použít zařízení jako ESP32, které má také nižší spotřebu energie a dostatečný výkon pro jednodušší aplikace, jako je právě tato.

Komunikace s OBD II sběrnicí se realizuje prostřednictvím Bluetooth adaptéru, například ELM 327. Tento adaptér slouží k připojení k diagnostickému portu vozidla a následně umožňuje přenos dat do zařízení, jako je Raspberry Pi. Na úrovni vozidla je komunikace s OBD II rozhraním zajištěna pomocí zařízení Raspberry Pi, na kterém běží program pro získávání a zpracování dat. Pro implementaci této komunikace bylo rozhodnuto využít existující knihovnu kvůli složitosti a náročnosti celé implementace. Jinak by bylo nutné programovat komunikaci zařízení přímo s jednotkou přes OBD II protokol.

V návrhové fázi byl zvolen balíček *python-OBD*, který poskytuje komplexní a snadno použitelnou sadu funkcí pro čtení dat z OBD II portu. Mezi hlavní důvody této volby patří jeho podpora široké škály OBD II příkazů, pravidelná údržba a aktualizace, což zajišťuje jeho kompatibilitu s různými OBD II adaptéry a spolehlivé využití v širokém spektru aplikací. Knihovna také umožňuje snadnou integraci do Python aplikací a poskytuje rozhraní pro asynchronní čtení dat, což je výhodné při zpracování dat v reálném čase. Prostřednictvím python-OBD zařízení pravidelně získává diagnostická data, jako je rychlost vozidla, otáčky motoru, úroveň paliva, chybové kódy a další údaje. Tato data jsou následně zpracovávána a využívána pro další analýzy a vizualizace.

Obrázek 5.2 zobrazuje abstraktní popis cyklu zpracování dat z OBD a následné ukládání dat do úložiště. Tento diagram ilustruje proces, ve kterém je nejprve získán seznam dostupných funkcionalit z OBD. Poté se spustí smyčka událostí, která je nekonečný cyklus, jenž poběží, dokud se nevypne přívod energie do zařízení ve vozidle. V rámci tohoto cyklu se vnitřně iterují všechny dostupné funkcionality, přičemž pro každou z nich se získávají konkrétní data. Data jsou následně uložena do úložiště zařízení jako CSV soubor. Po každém kompletním vnitřním cyklu následuje pauza 1 sekundu, než proces začne znovu pro další funkcionalitu. Tento proces se opakuje, dokud je program aktivní (dokud je zařízení napájeno).

Data jsou ukládána průběžně do úložiště, tedy na SD kartu, v serializovaném formátu. Pokud by se data neserializovala, nemohla by být správně převedena do přenositelného formátu, což je nezbytné pro jejich odeslání mezi zařízeními. Serializace zajišťuje, že data mají jasnou strukturu, která umožňuje jejich bezpečný přenos a správné zpracování na cílové straně. Bez serializace zůstávají data v původní, nestrukturované podobě v paměti, což může vést k jejich neodeslání, poškození nebo nečitelnosti při přenosu. Jakmile tedy zařízení detekuje připojení k síti (Wi-Fi), jsou serializovaná data odeslána na server.



Obrázek 5.2: Flow diagram cyklu získávání a zpracování dat z OBD

5.2 Návrh uživatelského rozhraní

Na úrovni klientské aplikace bude nutno vyřešit efektivní a přehledné zobrazení relevantních informací pro uživatele aplikace (mechanika). Aplikace je navržena jako webová a bude přístupná z webového prohlížeče. Umožní uživateli přistoupit k datům jednotlivých vozidel a použít je k diagnostice a opravě závad. Aplikace bude načítat data z RIoT serveru a zobrazovat je pomocí dashboardu.

Aplikace je optimalizována primárně pro zobrazení na notebooku s displejem o velikosti 15,6 palců, přičemž rozhraní je navrženo tak, aby se přizpůsobovalo šířce obrazovky s cílem zamezit horizontální posouvání (horizontal scroll). Rozlišení aplikace a velikost jednotlivých elementů je přizpůsobena pro zobrazení na 125% zvětšení.

Návrh stránky dashboardu

Dashboard je navržen jako součást interaktivní webové aplikace a umožňuje přehlednou vizualizaci dat získaných z OBD II sběrnice. Design je zaměřen na přehlednost, jednoduchost ovládání a přizpůsobení potřebám uživatele, především mechaniků nebo techniků.

Jeho vzhled lze vidět na obrázku 5.3, kde očíslované části odpovídají číslům sekcí v této kapitole. Dashboard je rozdělen do tří hlavních sekcí:

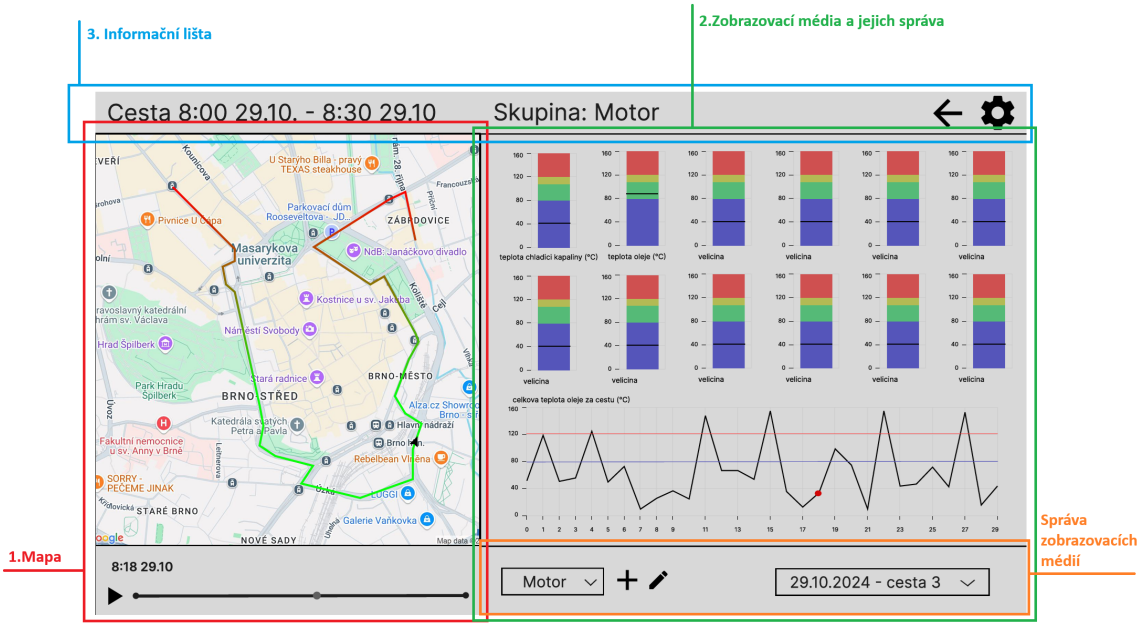
1. Mapa trasy: Levá část dashboardu obsahuje mapu, která zobrazuje trasu vozidla na základě nasbíraných GPS dat. Trasa je vizualizována pomocí křivky, která přesně kopíruje silnice a cesty v mapě (pokud vozidlo nejelo mimo cestu). Může mít více barev a to podle toho, zda v daný okamžik bylo s vozidlem vše v pořádku, nebo ne. Aktuální pozice vozidla, tedy pozice právě vizualizována na dashboardu, je vyznačena značkou na mapě. Značka je vždy umístěna na trase vozidla.

Pod mapou je umístěna časová osa, která umožňuje uživateli navigovat v čase záznamu dat. Osa vizualizuje časový úsek celé cesty, tedy od startu po konec. Obsahuje posuvník a tlačítko s funkcí přehrávání (*play*), které se po zmáčknutí změní na tlačítko pozastavení (*pause*) a obráceně, což umožňuje simulaci průběhu jízdy. Při přehrávání nebo posunutí posuvníku uživatelem se současně aktualizuje mapa i hodnoty podle zvoleného časového bodu. Uživatel má tedy možnost vybrat konkrétní bod a zobrazit detailní údaje, jako je rychlost, otáčky nebo teplota motoru v daném okamžiku.

2. Zobrazovací média a jejich správa: Pravá část dashboardu, z pohledu této práce nejdůležitější, je věnována grafům a jiným zobrazovacím médiím, které vizualizují různé klíčové ukazatele, jako jsou rychlost, otáčky motoru, spotřeba paliva nebo teplota chladicí kapaliny. Zobrazovací média jsou uspořádána ve skupinách, které lze podle potřeby přepínat. Vždy je zobrazena pouze jedna z nich. Jednotlivé skupiny jsou tvořeny uživatelem, který si je upraví podle své potřeby. Může se tedy rozhodnout, jaké parametry v nich chce zobrazit a jaké ne.

Spodní část pravé sekce, tedy pod samotnými grafy, obsahuje nástroje pro správu skupin zobrazených parametrů a výběr zobrazované cesty. Uživateli umožňuje vybrat stávající skupinu zobrazovacích médií, která odpovídá konkrétnímu účelu analýzy, například diagnostika motoru nebo analýza spotřeby paliva, dále upravit stávající parametry ve skupině, přidáním či odebráním konkrétních grafů a jiných médií z výběru, nebo vytvořit novou skupinu, která odpovídá specifickým požadavkům uživatele. Také zde lze vybrat záznam cesty, který bude dashboard vizualizovat. Uživatel může zobrazit seznam zaznamenaných cest a vybrat konkrétní trasu, kterou chce analyzovat. Cesta lze změnit také v horní části na liště, k čemuž se dostanu v další části.

3. Informační lišta: Informační lišta dashboardu slouží k navigaci, zobrazení informací o aktuální cestě, názvu skupiny parametrů a přístupu k nastavení aplikace. V pravé části lišty se nachází tlačítko pro návrat na úvodní stránku, které uživateli umožňuje snadno přejít zpět na výběr cest ze všech dostupných záznamů a zvolit tedy jinou cestu k analýze. Dále pravá část lišty obsahuje tlačítko pro přístup k nastavení, kde uživatel může upravit vizualizační parametry, například barevné schéma grafů nebo zvýraznění určitých dat, definovat filtrační kritéria či spravovat jiná relevantní nastavení. Levá část lišty zobrazuje klíčové informace o aktuálně zvolené cestě, jako je datum a čas zahájení a ukončení jízdy. Uprostřed se nachází název skupiny zobrazených parametrů.



Obrázek 5.3: Návrh obrazovky dashboardu

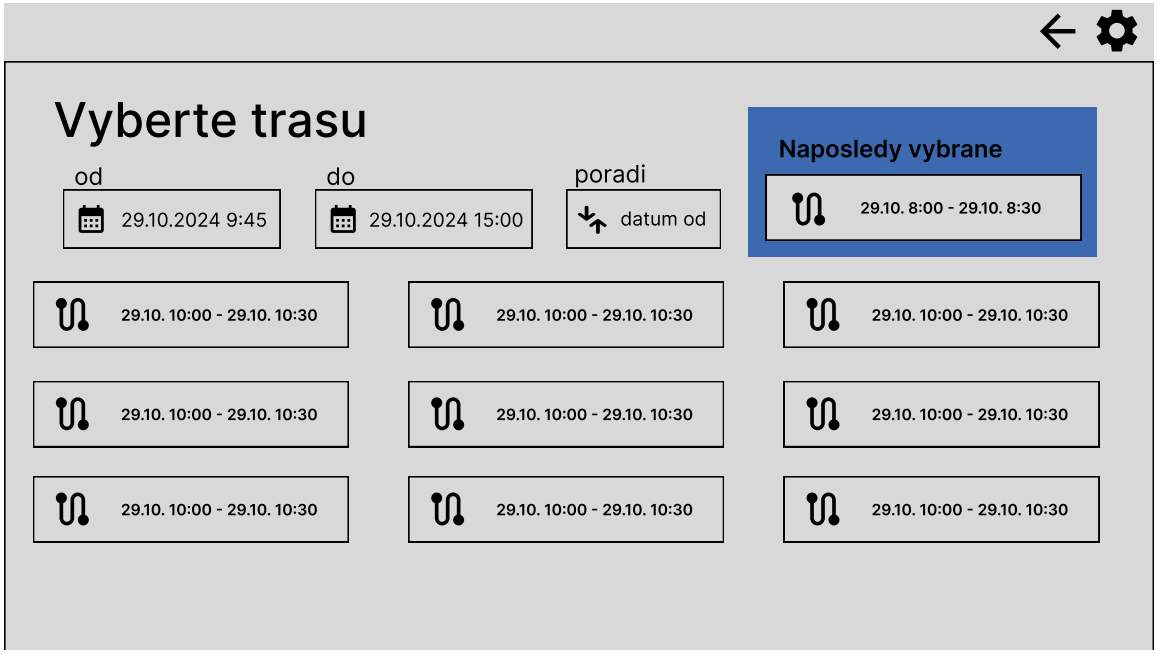
Návrh stránky výběru cesty

Stránka s výběrem cesty je přístupná přes tlačítko návratu (šipky) ze stránky s dashboardem. Tato stránka je také zobrazena hned po zapnutí aplikace a vybrání příslušného vozidla ze seznamu vozidel daného uživatele. Nabízí uživateli přehledný výběr cesty daného vozidla, kterou by chtěl vizualizovat. Lze ji vidět na obrázku 5.4.

Nahoře je stále dostupná informační lišta, na které se nachází pouze jedno tlačítko a to tlačítko nastavení. Přes něj se uživatel dostane do nastavení aplikace, jak bylo již zmíněno dříve. Zbytek stránky zobrazuje všechny dostupné cesty z historie, které lze zobrazit. Nahoře na stránce hned pod lištou je možno nastavit filtry a pořadí, ve kterém se záznamy zobrazí. Je zde možnost nastavit filtr dne a čas od a do, kdy se mají cesty zobrazovat, tedy je to řízeno časem, kdy byl záznam pořízen (reálný čas v době jízdy). Dále zde lze nastavit pořadí, ve kterém se cesty uživateli zobrazují. Pořadí může být podle délky nebo počátku cesty či konce cesty, ať už klesající nebo stoupající (například sestupně podle data a času počátku cesty).

Úplně vpravo vedle všech filtrů a pořadí je zvýrazněn záznam (cesta), který si uživatel naposledy vizualizoval. Po jeho vybrání se zobrazí vizualizace tohoto záznamu pomocí dashboardu. Tato funkce zjednodušuje navigaci v aplikaci a zvyšuje uživatelskou přívětivost. Uživatelé mohou například omylem přejít zpět na hlavní stránku nebo aplikaci restartovat, a proto je důležité poskytnout snadno dostupnou možnost návratu. Uživatel tedy po opětovném zapnutí aplikace jednoduše jen vybere naposledy vybranou cestu a nemusí ji hledat mezi tisíci jinými záznamy.

Zbytek stránky zabírají jednotlivé záznamy cest, které vozidlo vykonalo. Jeden záznam obsahuje vždy datum a čas zahájení cesty (nastartování) a datum a čas konce cesty (vypnutí motoru). Po vybrání jakéhokoliv záznamu se tento záznam vizualizuje pomocí stránky s dashboardem.

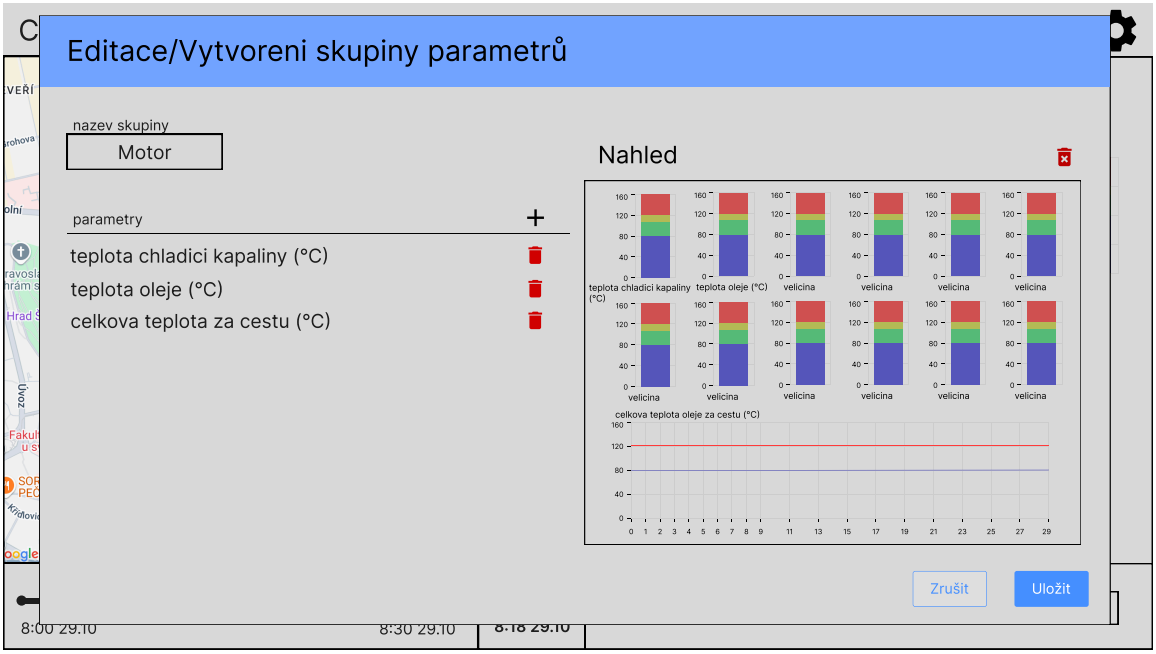


Obrázek 5.4: Návrh obrazovky výběru cesty

Návrh okna editace skupiny parametrů

Dialogové okno pro editaci skupiny parametrů na obrázku 5.5 je navrženo tak, aby uživatelům poskytovalo intuitivní rozhraní pro úpravu a vytvoření skupin parametrů získávaných z OBD rozhraní. V horní části okna vlevo se nachází název aktuálně upravované skupiny parametrů, který je zobrazen jako upravitelný text (tedy ho lze změnit). V dolní části okna vpravo jsou umístěna dvě tlačítka a to **Uložit**, které uživateli umožní uložit aktuální nastavení skupiny parametrů a zavřít okno, a **Zrušit**, které zahodí provedené změny a obnoví původní stav skupiny.

Levá část okna pod názvem skupiny obsahuje interaktivní seznam parametrů, kde každý parametr je zobrazen ve formě řádku. Parametr má název, který identifikuje parametr, a tlačítko pro odstranění ze seznamu. Nad seznamem parametrů se nachází tlačítko pro přidání nového parametru do skupiny. Kliknutím na toto tlačítko se otevře dialogové okno, které umožní vybrání nového parametru (zobrazovacího média) z dostupných a případnou prvotní konfiguraci. V pravé části okna je náhled výsledného dashboardu. Grafy a jiná zobrazovací média jsou zde vizualizována přesně tak, jak budou vizualizována na reálném dashboardu v aplikaci. Nad náhledem je červené tlačítko pro kompletní odstranění momentálně zobrazené skupiny parametrů.



Obrázek 5.5: Návrh okna editace/vytvoření skupiny parametrů

Návrh stránky nastavení

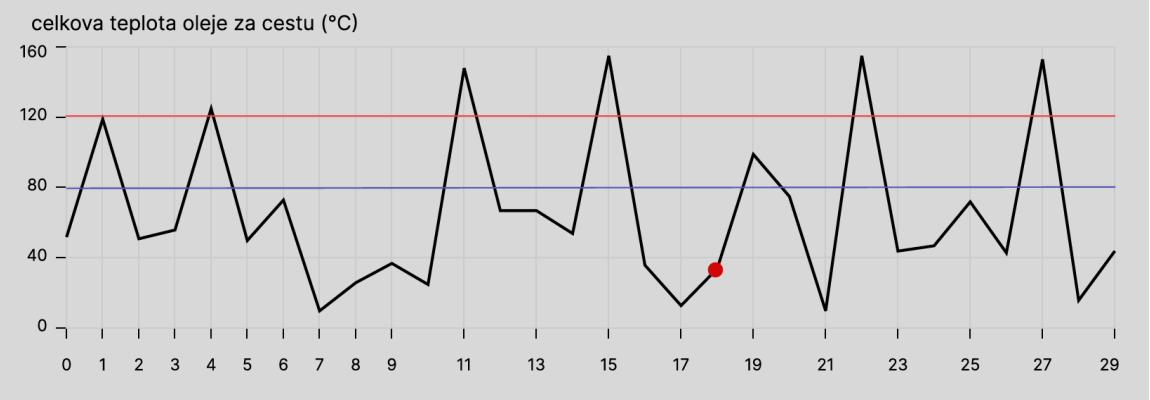
Stránka nastavení na obrázku 5.6 je navržena jako jednoduché uživatelské rozhraní umožňující přizpůsobení základních parametrů aplikace. V horní části stránky je umístěna informační lišta, která obsahuje pouze šipku pro návrat zpět na předchozí obrazovku (dashboard nebo výběr cesty). Pod ní se nachází hlavní sekce s dvěma nastaveními. Prvním je možnost upravit rychlost animace, kde uživatel může určit, jak rychle se má animace přehrávat, a to výběrem z přednastavených možností (1x, 2x, 0.5x, atd.). Druhé nastavení se týká kroku animace, což umožňuje uživateli nastavit velikost jednoho kroku animace v sekundách zadáním celého čísla do vstupního pole.

Obrázek 5.6: Návrh obrazovky nastavení

Použité grafy a vizualizační média

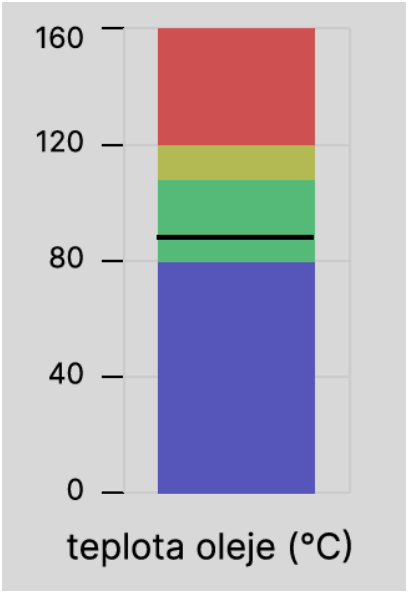
Na dashboardu pro vizualizaci dat z OBD byly zvoleny grafy a obecně média, která splňují požadavky uživatele na vysokou informační hodnotu a efektivní zobrazení i komplexních závislostí. Výběr grafů vycházel z analýzy uživatelských preferencí, která ukázala ochotu pracovat se sofistikovanějšími grafy a současně preferenci alternativních řešení namísto gauge grafů.

Pro zobrazení změn veličin (parametrů) v čase byly zvoleny spojité grafy, které umožňují intuitivní sledování trendů a výkyvů. Tento typ grafu byl zvolen pro zobrazení parametrů za nějaký časový úsek, protože umožňuje snadno identifikovat vzory a nečekané výkyvy v průběhu jízdy. Graf byl doplněn o tzv. thresholdy (prahy, mezní hodnoty), které rozdělují plochu grafu vertikálně na několik částí, což umožňuje snadno sledovat, zda některá veličina v určitém okamžiku překročila stanovené mezní hodnoty, nebo zda se pohybovala v očekávaném rozsahu, a tím usnadňuje identifikaci odchylek či potenciálních problémů.



Obrázek 5.7: Příklad spojitého grafu s mezními hodnotami

Bullet grafy byly zvoleny jako náhrada za gauge grafy, protože umožňují efektivní zobrazení různých parametrů s ohledem na stanovené mezní hodnoty, aniž by zabíraly příliš mnoho místa. Tento typ grafu je vhodný pro vizualizaci parametrů, jako je teplota oleje, kde je třeba sledovat aktuální hodnotu v kontextu několika zón – například normální teploty, varovné zóny a kritického stavu. Graf je rozdělen na několik částí, přičemž každá část má odlišnou barvu (např. zelená pro optimální hodnoty, žlutá pro varovné hodnoty a červená pro kritické úrovně), což umožňuje snadnou identifikaci aktuálního stavu vzhledem k těmto prahovým hodnotám.



Obrázek 5.8: Příklad bullet grafu

Pro zobrazení hodnot typu string (například stavů nebo textových popisů) a boolean hodnot (například zapnuto nebo vypnuto) z OBD systému bylo zvoleno prosté zobrazení hodnoty jako text. Boolean hodnoty jsou zobrazeny například jako text „Aktivní“ a „Neaktivní“ nebo „Zapnuto“ a „Vypnuto“ podle kontextu použití. Pro stringové hodnoty, jako jsou různé stavové zprávy nebo popisy, je hodnota jednoduše zobrazena tak, jak byla načtena z OBD portu.

Kapitola 6

Implementace

Na základě provedené analýzy a návrhu systému bylo přistoupeno k implementaci samotné webové aplikace, která slouží k vizualizaci dat získaných z OBD II sběrnice vozidla. Tato data jsou nejprve lokálně shromažďována a ukládána pomocí zařízení Raspberry Pi, které je připojeno k vozidlu prostřednictvím adaptéru ELM 327 (nebo jiného kompatibilního OBD II zařízení). Následně jsou v rámci dostupnosti Wi-Fi tato data odesílána na server, kde jsou dále zpracována. Pro implementaci cyklu získávání, zpracování a následného odesílání dat jsem vybral vysokoúrovňový, interpretovaný programovací jazyk Python. Rozhodl jsem se pro něj kvůli jednoduchosti a velkému rozsahu použitelnosti, tedy i pro tento projekt, dále také hlavně kvůli použití již existující Python knihovny pro čtení dat z OBD II sběrnice.

Pro zobrazení dat uživateli je implementována klientská část systému jako webová aplikace postavená na frameworku Quasar[1] využívajícího programovací jazyk TypeScript[2]. Aplikace běží přímo v internetovém prohlížeči a zajišťuje přehlednou prezentaci dat prostřednictvím dashboardu, který obsahuje například grafy, mapu a další vizualizační komponenty. Aplikace využívá již implementovaný systém RIoT pro ukládání, agregování a načítání dat o vozidlech.

6.1 Implementace na úrovni vozidla

Jak už jsem zmiňoval výše, pro implementaci procesů na úrovni vozidla jsem použil programovací jazyk Python. S jeho využitím jsem implementoval cyklus pro získávání dat, který je uložen v paměti Raspberry Pi a stará se o cyklické získávání dat z OBD II sběrnice, a také skript pro odesílání získaných dat z úložiště Raspberry Pi na server, nakonec i jednoduchý skript pro odesílaní gps dat na server. Skripty lze konfigurovat přes společný *.env* soubor, který umožňuje nastavit parametry chování jako je adresa serveru či umístění souboru CSV.

Cyklus pro získávání dat

Cyklus pro získávání, zobrazený na obrázku 5.2, využívá knihovnu *python-OBD*, která nabízí příkazy pro čtení dat ze sběrnice a umožňuje jednodušší správu a komunikaci se zařízením. Knihovna za uživatele řeší celou komunikaci a udržuje spojeni mezi zařízením a vozidlem otevřené, takže není potřeba se zaobírat samotnou komunikací přes OBD II protokol. Knihovna vytváří jednoduché rozhraní, přes které uživatel přistupuje k jednotlivým příkazům pro komunikaci s jednotkou vozidla.

[1] Quasar: https://quasar.dev
[2] TypeScript: https://www.typescriptlang.org

Nejprve je potřeba inicializovat spojení s použitím příkazu *obd.OBD()*. Následně je zjištěno, jaké hodnoty a tedy jaké parametry jednotka daného vozidla podporuje. Bez zjištění dostupnosti parametrů by mohlo dojít k vrácení prázdné odpovědi a tedy by se dotaz vykonával zbytečně a zatěžoval spojení. Kromě toho by se zbytečně ukládaly a zpracovávaly prázdné odpovědi. Knihovna totiž podporuje desítky příkazů a mnoho, především starších vozidel, s nimi není kompatibilních. Jde tedy hlavně o větší efektivitu a menší zátěž spojení mezi zařízením a vozidlem.

Po inicializaci spojení jsou data načítána v nekonečném cyklu s využitím Python plánovače, dokud se zařízení nevypne či nerestartuje. Plánovač (scheduler) v jazyce Python je softwarová komponenta, která umožňuje časové řízení spouštění funkcí nebo úloh. Plánovač umožňuje periodické spouštění funkcí v definovaných intervalech či časech. V rámci této práce je využit plánovač, který umožňuje jednoduchým způsobem nastavit periodické spouštění funkce pro načítání dat z OBD II sběrnice. V každém jednom cyklu (spuštění funkce) se pomocí sekundárního cyklu projdou všechny dostupné parametry a získají se jejich data pomocí korespondujících příkazů. Data se také případně správně formátují pro následné správné uložení do CSV souboru. Na obrázku 6.1 je jednoduchý příklad komunikace přes OBD II s využitím *python-OBD*.

```
import obd

connection = obd.OBD() # auto-connects to USB or RF port

cmd = obd.commands.SPEED # select an OBD command (sensor)

response = connection.query(cmd) # send command, parse response

print(response.value) # returns unit-bearing values (Pint)
print(response.value.to("mph")) # unit conversions
```

Obrázek 6.1: Příklad komunikace s využitím *python-OBD*

Data jsou průběžně ukládána do vyrovnávací paměti (buffer) pro dočasné uložení před zápisem do CSV. Po zaplnění určitého počtu řádků v dočasné paměti se data hromadně zapíší do CSV souboru. První řádek souboru tedy slouží jako hlavička obsahující názvy jednotlivých proměnných, zatímco každá další řádka reprezentuje jeden záznam získaný z vozidla v konkrétním časovém okamžiku s hodnotou *timestamp* navíc pro identifikaci.

Odesílaní dat na server

Po nasbírání diagnostických dat z OBD II sběrnice a jejich dočasném uložení ve formátu CSV probíhá následné zpracování a odeslání těchto údajů na server do RIoT aplikace pomocí již zmíněného protokolu MQTT 4.5 prostřednictvím zabezpečeného spojení přes WebSocket. Skript nejprve nalezne všechny dostupné CSV soubory (může jich být více kvůli několika odděleným startům vozidla) a následně je postupně po jednom načte. Načtená data jsou poté transformována do podoby slovníku (datové struktury typu klíč–hodnota), přičemž dochází také k převodu textových hodnot zpět na odpovídající datové typy (například čísla na datový typ `float` nebo `int`). Pro každý takto připravený záznam je vygenerován jedinečný identifikátor zprávy a sestaven objekt typu JSON, který kromě samotných dat

ze senzorů obsahuje také informace o zařízení, typu vozidla, identifikátoru VIN a časovém razítku. Příklad struktury takového objektu je na obrázku 6.2.

```
"devPars": {
    "timestamp": "2025-04-26T14:30:00Z",
    "SPEED": 80,
    "RPM": 2500,
    "FUEL_LEVEL": 65,
    "COOLANT_TEMP": 90,
    "THROTTLE_POSITION": 25
  }
```

Obrázek 6.2: Struktura dat odesílaných na server pres MQTT

Zprávy obsahují vždy pouze jeden řádek dat z CSV souboru a jsou odesílány samostatně, každá na specifické MQTT téma, jež je odvozeno z konfiguračních parametrů zařízení (soubor .env). Samotné odesílání probíhá prostřednictvím klienta knihovny *paho-mqtt*, který navazuje TLS zabezpečené spojení s MQTT brokerem. Po zpracování všech řádků může být CSV soubor dle nastavení buď trvale smazán, nebo zálohován do zvláštní složky s časovým razítkem v názvu souboru. Celý proces je následně ukončen odpojením MQTT klienta.

Odesílání GPS dat na server je řešeno podobně jako odesílání samotných dat z vozidla. Data se načítají z GPX[3] souboru a následně jsou po jednotlivých záznamech odesílána prostřednictvím MQTT protokolu. GPX soubor je vybrán pomocí jednoduchého uživatelského rozhraní postaveného na knihovně Tkinter, čímž je zajištěna základní interaktivita. Samotné zpracování GPX formátu probíhá s využitím knihovny gpxpy, která umožňuje extrakci zeměpisné šířky, délky, nadmořské výšky a času jednotlivých bodů trasy.

Pro každý bod trasy je vytvořen unikátní MQTT payload obsahující základní metadata, identifikátor zařízení (včetně VIN kódu) a samotné GPS souřadnice. Struktura odesílané zprávy je prakticky identická jako u telemetrických dat z OBD II portu, což umožňuje snadnou integraci a sjednocení zpracování dat na straně serveru. Každá zpráva je odesílána zvlášť pomocí MQTT klienta s použitím zabezpečeného připojení přes WebSocket a TLS. Celý proces probíhá automatizovaně – po načtení dat a jejich zpracování se jednotlivé zprávy publikují na specifikované MQTT téma, které zahrnuje identifikaci projektu, typu zařízení a konkrétního vozidla.

6.2 Implementace uživatelské aplikace

Uživatelská aplikace je realizována jako webová aplikace postavená na frameworku Quasar, jenž je založen na knihovně Vue.js[4]. Tento moderní vývojový rámec poskytuje sadu předpřipravených komponent a nástrojů, které zjednodušují vývoj vícestránkových či responzivních aplikací. Komponenty lze snadno konfigurovat prostřednictvím jejich atributů a případně dále přizpůsobovat pomocí kaskádových stylů (CSS), čímž se významně snižuje potřeba manuálního návrhu a implementace uživatelského rozhraní od základů.

[3] GPX (GPS Exchange Format) je XML formát pro ukládání a výměnu dat o geografické poloze
[4] Domovská stránka projektu Vue.js: https://vuejs.org

Vzhledem k tomu, že Quasar využívá principy reaktivního programování skrze Vue.js, je zajištěna konzistentní synchronizace datového modelu a tedy stavu systému s vizuální reprezentací uživatelského rozhraní. Při jakékoliv změně se aplikace automaticky překreslí, což zajišťuje velkou responzivitu. Vývojová logika komponent je implementována v jazyce TypeScript, který nadstavuje nad JavaScript silnější typový systém. Tím se zvyšuje spolehlivost kódu, zlepšuje přehlednost v rozsáhlejších projektech a umožňuje efektivnější detekce chyb již při překladu.

Architektura aplikace je komponentově orientovaná. Komunikace mezi komponentami je řešena pomocí deklarativních vazeb a vlastností komponent, bez nutnosti explicitní manipulace s objektovým modelem dokumentu (DOM z anglického Document Object Model) nebo implementace složitých přenosových struktur. Použitím tohoto přístupu dochází ke zjednodušení vývoje i údržby aplikace a zároveň ke zvýšení její rozšiřitelnosti.

V rámci implementace jsou dále využívány doplňkové balíčky, které rozšiřují funkcionalitu aplikace. Jedním z nich je *vue-i18n*[5], který zajišťuje podporu vícejazyčnosti. Tento balíček umožňuje snadno definovat jazykové mutace a dynamicky měnit jazyk rozhraní podle preferencí uživatele. Lokalizační soubory jsou spravovány ve formátu JSON a jednotlivé překlady jsou dostupné prostřednictvím reaktivní vazby přímo v komponentách. Mezi další používané balíčky patří například *Apollo Client*[6] pro komunikaci se serverem pomocí GraphQL nebo *Pinia*[7] pro správu aplikačního stavu.

Implementace komunikace se serverem

Aplikace slouží k reprezentaci dat pro uživatele, tedy je nutné, aby se data pro vizualizaci nějakým způsobem získávala. V předchozí sekci 6.1 jsem zmiňoval, že se data posílají do RIoT aplikace pomocí MQTT protokolu. Data jsou jednorázově odesílána na MQTT brokera a server si je poté zpracuje a uloží do InfluxDB (databáze). Pro načtení dat se využívá GraphQL[8]. Jedná se o pokročilý dotazovací jazyk pro API, který umožňuje přesně definovat, jaká data má server vrátit. Na rozdíl od REST API (Representational State Transfer API), kde jsou data často pevně strukturována, GraphQL umožňuje klientovi určit konkrétní požadované hodnoty, čímž se minimalizuje množství přenesených dat a zvyšuje se efektivita komunikace.

Pro zajištění komunikace mezi frontendovou aplikací a serverem je využíván nástroj Apollo Client, který slouží jako prostředník pro práci s GraphQL API. Aplikace zasílá dotazy a přijímá odpovědi pomocí protokolu HTTP, přičemž může být každý požadavek doplněn o doplňující hlavičky, například pro účely autentizace. Hlavička pro autentizaci obsahuje JWT token, který je dynamicky vkládán při každém požadavku, čímž je zajištěna bezpečnost přenosu dat. Pokud není JWT token přítomen, je uživatel odkázán na stránku pro autentizaci (Google Auth), kde získá nový JWT token pro komunikaci.

Konfigurace klienta zahrnuje specifikaci cílového endpointu, správu mezipaměti a nastavení případných middleware vrstev. Data jsou načítána pomocí GraphQL dotazů, které mohou být parametrizované a navržené tak, aby vracely pouze potřebné množství informací. Typicky se jedná o výběr posledních záznamů konkrétního zařízení, přičemž se definuje identifikátor zařízení, časové omezení, nebo způsob řazení výsledků. Na obrázku 6.3 je zobrazen jednoduchý dotaz na jeden senzor (vozidlo) v daném systému.

[5] vue-i18n: https://vue-i18n.intlify.dev
[6] Apollo Client: https://www.apollographql.com/docs/react
[7] Pinia: https://pinia.vuejs.org
[8] GraphQL: https://graphql.org

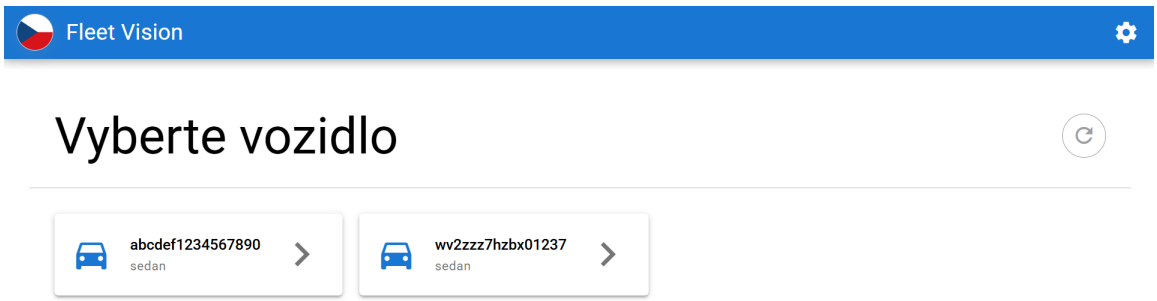
```
query StatisticsQuery($sensors: SimpleSensors, $request: StatisticsInput) {
  statisticsQuerySimpleSensors(sensors: $sensors, request: $request) {
    time
    deviceId
    data
  }
}
```

Obrázek 6.3: Dotaz na senzor pomocí GraphQL query

Kromě načítání dat umožňuje Apollo Client také správu lokálního stavu, aktualizaci dat prostřednictvím mutací a periodické opakování dotazů pomocí tzv. *pollingu*. Výhodou využití GraphQL v kombinaci s tímto nástrojem je možnost přesně specifikovat, jaká data má server vrátit, a tím snížit objem přenášených informací. Celý komunikační proces je tímto způsobem centralizován, bezpečný a snadno rozšiřitelný o další požadavky dle potřeb aplikace.

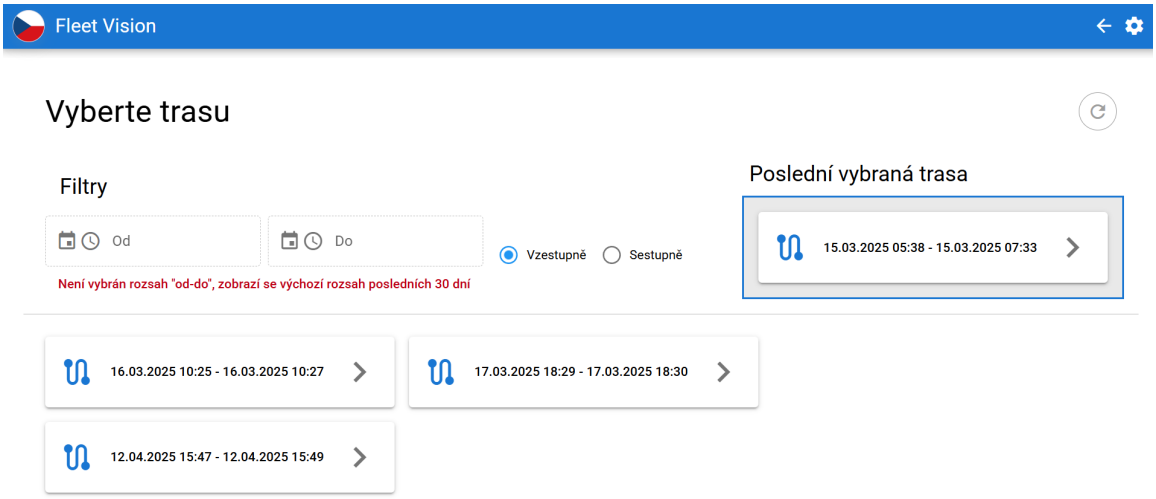
Implementace uživatelského rozhraní

Jak už jsem zmiňoval v návrhu, aplikace se skládá ze stránky výběru trasy, následně i samotného dashboardu a stránky nastavení. V porovnání s návrhem se výsledná aplikace moc neliší. Oproti návrhu je doplněna navíc stránka výběru vozidla. Stránka výběru vozidla umožní uživateli vybrat, jaké vozidlo chce zobrazit z vozidel načtených ze serveru přes GraphQL dotaz. Jedná se o jednoduchý seznam vozidel, ze kterého si uživatel, vybere jaké chce zobrazit. Stránka je zobrazena na obrázku 6.4.



Obrázek 6.4: Vzhled implementované stránky výběru vozidla

Stránka výběru trasy pak načte dostupné záznamy daného vozidla ze serveru podle zadaných parametrů časového rozsahu. Následně je agreguje do několika tras podle předdefinované hranice (2 minuty) mezi jednotlivými daty. Uživatel si výsledný seznam tras může seřadit sestupně nebo vzestupně podle data. Implementace této stránky je na obrázku 6.5.



Obrázek 6.5: Vzhled implementované stránky výběru trasy

Hlavní stránka typu dashboard pak uživateli zobrazuje všechna potřebná data pro diagnostiku vozidla. Levá strana obsahuje mapu dané trasy a ovládací panel s posuvníkem a nastavením dashboardu, jak už bylo popisováno v návrhu 5.2. Oproti návrhu bylo změněno pouze umístění nastavení dashboardu a zmenšena velikost mapy. Pro umístění nastavení pod mapu jsem se rozhodl z důvodu úspory místa a obecně pro ušetření prostoru pro samotný dashboard. Pravá strana pak obsahuje samotný dashboard s grafy, jež reprezentují konkrétní data. Nakonec stránka nastavení umožňuje měnit hodnoty jako rychlost a krok animace dashboardu (viz. 6.2), barvu prahu mapy (viz. 6.2) a resetovat celou aplikaci do výchozího stavu. Obrázek 6.6 zobrazuje implementovanou stránku dashboardu.

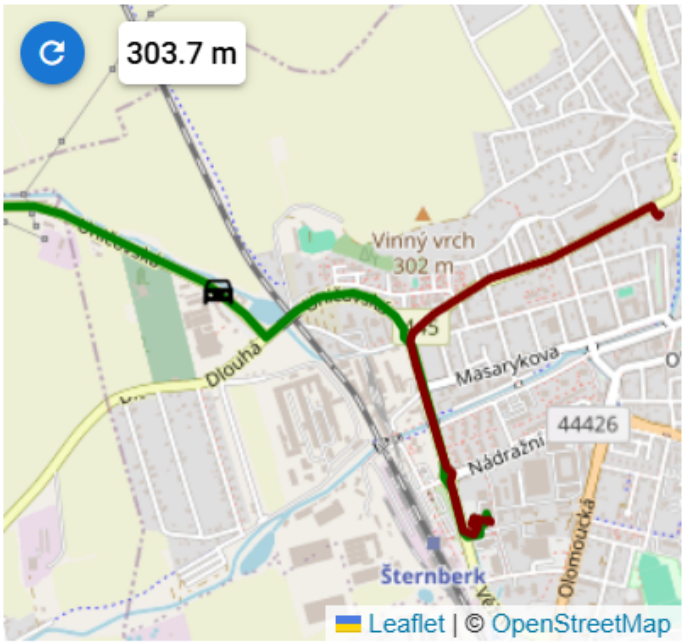
Obrázek 6.6: Vzhled implementované stránky dashboardu

Mapa

Mapa je implementována s využitím JavaScriptové knihovny Leaflet[9], která slouží k vykreslování interaktivních mapových podkladů v prostředí webových aplikací. Tato knihovna umožňuje zobrazit mapu včetně základní interakce, jako je posun, přiblížení nebo změna vrstvy. Do mapy je dále možné dynamicky přidávat prostorové prvky, jako jsou značky (markery), trasy (polylines) nebo oblasti (polygony). V rámci této práce je Leaflet využit pro vizualizaci pohybu vozidla na základě geografických dat, která jsou získávána ze serveru. Na základě těchto dat jsou na mapě zobrazovány jednotlivé body (geolokační pozice), které dohromady vytvoří celou projetou trasu vozidla.

Mapové podklady pro aplikaci jsou načítány z veřejně dostupného mapového systému OpenStreetMap[10], který poskytuje bezplatná kartografická data. Tato data jsou využívána jako podkladová vrstva pro Leaflet a umožňují tak zobrazení geografického kontextu. Mapová komponenta je integrována do aplikace pomocí knihovny Vue-Leaflet, jež využívá Leaflet a umožňuje jednodušší propojení mapy s reaktivní logikou aplikace vytvořené ve Vue.js (Quasar).

Na mapě se vždy vykresluje aktuální pozice vozidla v rámci celé trasy pomocí ikony automobilu. Pozice se vztahuje k aktuálnímu časovému okamžiku vybranému pomocí posuvníku pod mapou. Pokud není aktuální pozice známá, vykreslí se poslední známá pozice a ikona zabarví na červeno. Trasa je vykreslována také a to v jedné základní barvě (zelená). Pokud nějaká aktuálně zobrazená veličina na dashboardu přesáhla uživatelem definovaný práh v nějakém bodě trasy, korespondující bod se (případně více bodů) zabarví na uživatelem zvolenou barvu. Na obrázku 6.7 lze vidět zabarvení trasy na červeno při přesažení takového prahu. V případě úplné nedostupnosti jakýchkoliv geografických dat vozidla se mapa nezobrazí vůbec.



Obrázek 6.7: Zabarvení trasy na barvu prahu

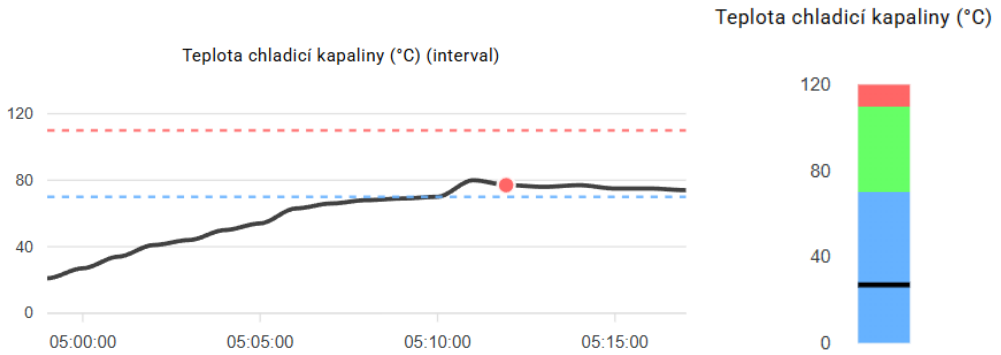
[9]Leaflet: https://leafletjs.com
[10]OpenStreetMap: https://www.openstreetmap.org

Zobrazení grafů

Dashboard s grafy představuje jednu z klíčových částí uživatelského rozhraní aplikace. Zobrazuje několik různých grafů a ukazatelů, které znázorňují konkrétní data, například rychlost, stav paliva, teplotu motoru nebo další parametry získané z řídicí jednotky vozidla. Stejně jako na mapě, i tady se data vztahují vždy k aktuálnímu časovému okamžiku vybranému pomocí posuvníku pod mapou. Pro implementaci grafického výstupu je využita knihovna ApexCharts[11], která poskytuje široké možnosti tvorby interaktivních a přehledných grafů v prostředí webové aplikace. Tato knihovna umožňuje vykreslování sloupcových, čárových, koláčových i dalších typů grafů a zároveň podporuje responzivní zobrazení a dynamické aktualizace dat.

Grafy jsou napojeny na reaktivní datový model aplikace, čímž je zajištěno, že se při změně dat automaticky aktualizuje i odpovídající vizualizace. Díky této vlastnosti je možné měnit pozici v rámci dané trasy pomocí posuvníku pod mapou a data se automaticky mění. Také je možné spustit u posuvníku animaci, která podle uživatelem definované rychlosti a kroku animace mění zobrazovaná data v rámci určité trasy (včetně pozice mapy).

Jak už jsem zmiňoval dříve v návrhu 5.2, grafy jsou dvou typů, a to bullet graf pro aktuální hodnotu na obrázku 6.8 vpravo, který zobrazuje přesnou hodnotu daného parametru v daném okamžiku, a graf spojitý na obrázku 6.8 vlevo, který ukazuje průběh dané hodnoty parametru po celou trasu včetně aktuální hodnoty v daném okamžiku. Grafy jsou seskupeny do několika skupin, které jsou v aplikaci již předdefinované, ale uživatel si je může upravit podle sebe, či změnit nebo úplně odstranit. Aktuálně vybraná skupina se zobrazí na dashboardu.

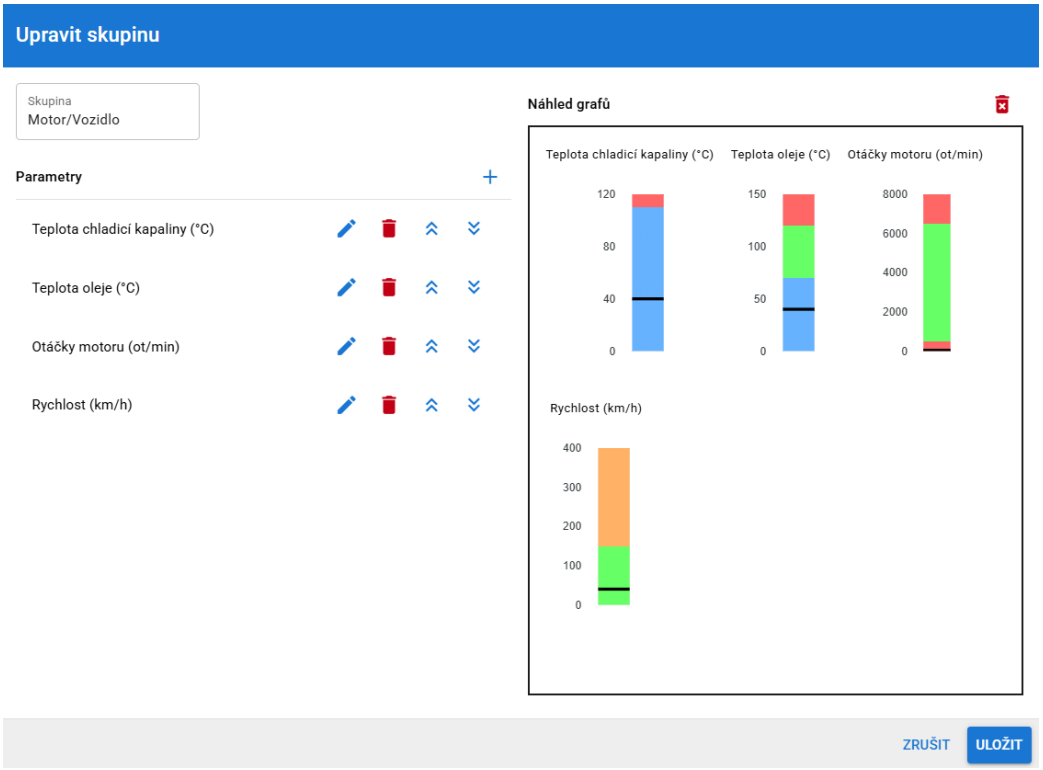


Obrázek 6.8: Spojitý graf vlevo a bullet graf vpravo

Editace grafů

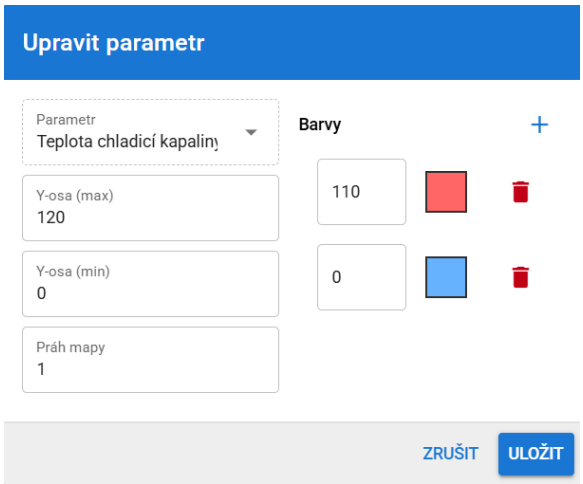
Skupiny grafů a zároveň i samotné grafy lze určitým způsobem upravovat a měnit. U skupin lze nastavit jejich název, definovat jaké grafy obsahují a v jakém pořadí se mají dané grafy ve skupině zobrazit. Obrázek 6.9 zobrazuje dialogové okno pro editaci nebo vytvoření skupiny grafů. Kromě výše zmíněných možností editace obsahuje i náhled jednotlivých grafů, aby uživatel věděl, jak přibližně definované grafy na výsledném dashboardu vypadají. Je zde možnost jednotlivé grafy i upravit a změnit jejich vlastnosti.

[11] ApexCharts: https://apexcharts.com



Obrázek 6.9: Okno pro vytvoření a editaci skupin grafů

Úprava jednotlivých grafů skupiny se provádí pomocí okna editace grafu (obecně parametru), které je zobrazeno na obrázku 6.10. Pro grafové parametry jsou zde možnosti úpravy vrchní a spodní hranice grafu, tedy jaký má graf rozsah na vertikální ose y. Dále nabídka poskytuje možnost nastavit již zmíněný práh mapy, který daný graf na dashboardu podbarví červenou barvu, pokud se přesáhne jeho hodnota, a zároveň obarví část trasy na mapě na barvu definovanou uživatelem, pokud v jejím rozsahu došlo k přesažení daného prahu.



Obrázek 6.10: Okno pro editaci konfigurace grafu

Další konfigurace grafu se liší podle typu grafu. Pokud se jedná o bullet graf pro aktuální hodnotu, je zde navíc možnost nastavit pozadí grafu a tím rozlišit různé rozsahy. Specifikuje se barva a spodní hranice rozsahu, přičemž automaticky se danou barvou obarví část grafu po další spodní hranici jiného rozsahu nebo vrchol grafu. Toto lze vidět na obrázku 6.8 vpravo. Pro grafy spojité, které zobrazují celý průběh daného parametru, je zde možnost nastavit jednu barvu grafu, jež obarví křivku daného grafu na zvolenou barvu. Dále je tu možnost definovat prahy grafu, které se vykreslí do výsledného grafu na dashboardu. Definuje se u nich název, pozice na vertikální ose y a barva daného prahu. Používají se pak k přehlednému rozdělení grafu na rozsahy, například jak tomu je na obrázku 6.8 vlevo.

Kapitola 7

Zhodnocení přístupu

Implementace řešení byla od začátku postupně testována. Nejprve byly jednotlivé části systému ověřovány samostatně, následně pak celý systém jako celek. Postupné testování bylo nezbytné zejména kvůli požadavku na kontinuální a spolehlivý sběr dat přímo ve vozidle. Aby bylo možné správně implementovat a následně otestovat vizualizaci ve webové aplikaci, bylo nutné nejprve zajistit, že samotný sběr dat probíhá bezchybně. Z tohoto důvodu proběhlo několik jízd, během kterých byla data zaznamenávána a následně využita pro vývoj a testování vizualizační části systému.

Aby bylo možné ověřit správnost a stabilitu sběru dat v reálných podmínkách, proběhlo několik testovacích jízd s několika vozidly a s využitím adaptéru ELM 327, během kterých se systém opakovaně zkoušel a ladil. GPS data byla získávána manuálně při jízdě vozidla pomocí chytrého telefonu a aplikace pro zaznamenání GPS dat *GPS Logger*[1].

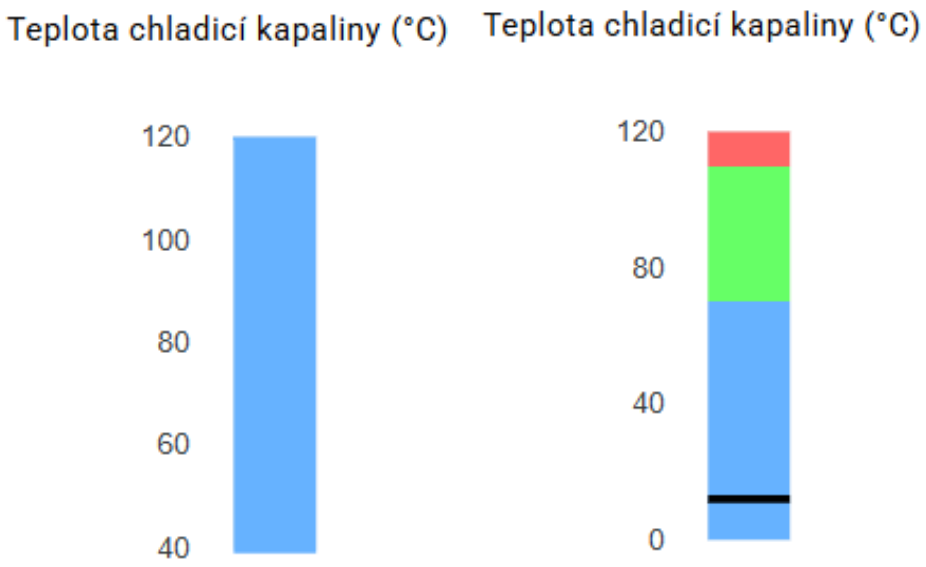
Během testování se postupně objevily určité problémy, například výpadky spojení, nekompletní datové záznamy nebo chyby v načítání některých parametrů. Tyto nedostatky byly v průběhu implementace průběžně identifikovány a opravovány. Část problémů byla způsobena omezeními použitého ELM adaptéru, jehož chování není možné přímo softwarově ovlivnit. Jako možné řešení se nabízí použití jiného adaptéru s lepší stabilitou a podporou. Díky postupnému testování a ladění se i přesto podařilo všechny zásadní chyby odstranit ještě před dokončením projektu, čímž se výrazně zvýšila spolehlivost výsledného řešení.

7.1 Změny v průběhu vývoje

V průběhu celého vývoje došlo k několika drobným, ale i poměrně zásadním změnám v aplikaci. První velkou změnou bylo přidání možnosti úpravy grafů pro jednotlivé parametry z OBD sběrnice. Původně se totiž uživateli po výběru grafu vůbec neumožňovalo upravit žádné jeho parametry a musel tedy mít graf ve stavu takovém, jak je předdefinovaný (graf na obrázku 7.1 vlevo). Bohužel se to jevilo jako nevhodné, protože například pokud byla reálná hodnota (aktuální) parametru mimo rozsah konkrétního grafu, nebyla vůbec viditelná na grafu a uživatel tedy neměl možnost zjistit její aktuální hodnotu. U spojitých grafů se tento problém projevoval křivkou zasahující mimo oblast grafu, případně úplně celou křivkou mimo rozsah grafu. Z tohoto důvodu tedy byla přidána možnost editace grafu, která umožnila upravit horní a dolní hranici grafu (osa y), například právě pokud jeho reálná hodnota byla mimo rozsah původního grafu.

[1] GPS Logger: https://play.google.com/store/apps/details?id=eu.basicairdata.graziano.gpslogger

Kromě úpravy hranic se přidala i možnost změny barvy grafu a u grafu pro aktuální hodnotu (bullet graf) i možnost definice více barev pro různé rozsahy grafu. Na obrázku 7.1 vpravo je vidět graf, který je tímto způsobem upraven a tudíž je přehlednější a díky změněnému rozsahu lze vidět i aktuální hodnotu, která na původním (7.1 vlevo) nebyla vidět. U grafů spojitých byla přidána možnost definovat tzv. prahy, které umožnily přidat do grafu vodorovné křivky pro rozdělení rozsahu na více částí (například pro označení rozsahu, ve kterém je vše v pořádku). Tyto úpravy byly provedeny s cílem zvýšit přehlednost dashboardu a zlepšit srozumitelnost zobrazených grafů. Z praktického testování vyplynulo, že uživatelé lépe vnímají grafy, které jsou rozděleny do dílčích rozsahů. Tyto rozsahy jim přehledně ukazují, jak si daný parametr vede, například zda je ve vyhovujícím, nebo nevyhovujícím stavu. Jednobarevné grafy bez dělení na dílčí části byly z hlediska srozumitelnosti hůře čitelné. Nakonec i samotná úprava a konfigurace grafů dává uživateli určitou svobodu a možnost si vizualizaci více přizpůsobit vlastní preferenci.



Obrázek 7.1: Graf před možností editace (vlevo) a po přidání možnosti editace (vpravo)

Dále také došlo k menší změně vizualizace trasy vozidla na mapě oproti původnímu návrhu. Původně měla být trasa vozidla vizualizována více barvami, ne pouze dvěma. To lze vidět na obrázku 5.3 v návrhu aplikace. Trasa měla měnit barvu podle závažnosti problému v daném bodě. Tedy například když by byl definován rozsah čtyř barev, a to červená pro vážné problémy, oranžová pro středně závažné, žlutá pro málo závažné a zelená pro žádné problémy, pokud by v daném bodě došlo k dvěma středně závažným a jednomu vážnému problému, bod by se zabarvil červeně. Nakonec jsem tuto funkcionalitu implementoval jen v omezeném rozsahu (viz. obrázek 6.7). Rozhodl jsem se tak nejen z důvodu omezených časových možností, ale především s ohledem na cílovou skupinu uživatelů. Výsledky uživatelského testování ukázaly, že i relativně jednoduché grafické rozhraní může být pro méně technicky orientované uživatele složité. Z tohoto důvodu byla upřednostněna nižší funkcionalita ve prospěch vyšší a lepší použitelnosti. Barvy jsou definovány pouze dvě a obarvení trasy se vztahuje pouze k aktuálně vybrané skupině grafů. Trasa je primárně obarvena jednou barvou a změní se na druhou pouze, pokud v daný okamžik jeden z parametrů skupiny přesáhne hranici definovanou uživatelem.

7.2 Demonstrace

V rámci ověření použitelnosti a funkčnosti vytvořeného systému proběhly dvě klíčové demonstrace. První byla realizována s expertem na dashboardy, druhá s uživatelem z praxe. Cílem bylo ověřit správnost a přehlednost vizualizací, intuitivnost ovládání systému, schopnost uživatele samostatně plnit praktické úkoly pomocí grafického rozhraní a identifikovat případné problémy nebo návrhy na zlepšení.

Demonstrace s expertem na dashboardy

První demonstrace byla provedena před Ing. Jiřím Hynkem, Ph.D., odborníkem na návrh a používání dashboardů v profesionálním prostředí. Ing. Jiří Hynek, Ph.D., se ve své disertační práci věnoval tématu vlivu subjektivního vizuálního vnímání na automatické hodnocení vzhledu rozhraní dashboardů [14]. Během demonstrace byla představena celá aplikace včetně možnosti výběru tras, zobrazování vizualizačních dashboardů a práce s mapou.

Expert hodnotil vizualizace pozitivně. Grafy, mapa i struktura dashboardů byly podle něj přehledné a vhodné pro použití. Připomínky směřovaly k procesu výběru skupin, kde doporučil zpřehlednění a přidání případného filtrování, aby byl výběr pro uživatele intuitivnější a více přehledný. Druhá připomínka se týkala přesnosti zobrazení trasy na mapě, kde bylo během demonstrace odhaleno, že problém spočívá v agregaci dat a jejich získávání po velkých intervalech (po jedné minutě). Tento nedostatek byl následně odstraněn úpravou metody získání dat, tedy úpravou intervalu na menší, a to na jednu sekundu. Zároveň se otevřela otázka, jak pracovat s chybějícími vzorky. Diskutovali jsme možnost jejich přibližného dopočítání podle plánované trasy, což by umožnilo trasu zobrazit i v místech, kde nebyla data dostupná.

Demonstrace s uživatelem z praxe

Druhá demonstrace proběhla s uživatelem z praxe, který nemá odborné znalosti v oblasti návrhu dashboardů, ale představuje typického koncového uživatele systému. Jednalo se o vedoucího mechanika šumperské pobočky společnosti CBA. V rámci demonstrace byly provedeny čtyři konkrétní testy zaměřené na běžné uživatelské operace.

První úkol spočíval v tom, zjistit, jaká byla teplota chladicí kapaliny dne 26. dubna 2025 v 8:15 pro vozidlo s VIN `WV2ZZZ7HZBX01237`. Uživatel neměl žádný problém s výběrem požadovaného vozidla, stejně tak výběr příslušné cesty ve správný den proběhl bez obtíží. Menší komplikace nastala při výběru skupin parametrů na dashboardu, kdy uživatel mylně předpokládal, že jednotlivé skupiny slouží k výběru konkrétních parametrů, nikoliv celých kategorií. Po správném výběru skupiny se následně setkal s dalším problémem, kdy se dashboard nevešel na obrazovku a byl tedy mírně posunutý směrem dolů (vertikální posun). Uživatel si tedy nevšiml názvu grafu, který hledal, protože název byl mimo obrazovku. To vedlo k určité nejistotě při identifikaci správného grafu, i přesto se mu ale nakonec podařilo daný údaj najít.

Druhý úkol měl za cíl zjistit, kde se přibližně nacházelo vozidlo s VIN `WV2ZZZ7HZBX01237` dne 26. dubna 2025 kolem 7:52 a jaká byla jeho nadmořská výška, případně jestli stoupalo nebo klesalo. Ani v tomto případě uživatel neměl problém s výběrem správného vozidla ani příslušné jízdy. Bez potíží se také zorientoval v mapě a rychle lokalizoval přibližnou polohu vozidla v daný čas. Potíže nastaly pouze při hledání informace o nadmořské výšce. Uživatel ji totiž intuitivně očekával na dashboardu mezi grafy, a nikoli v mapové části rozhraní, kde byla ve skutečnosti zobrazena. To vedlo ke krátkému zdržení, než si uživatel uvědomil, kde

se požadovaný údaj nachází. Přesto nakonec správně určil jak nadmořskou výšku, tak i to, zda vozidlo stoupalo či klesalo.

Třetím úkolem bylo zjistit, v jakém rozsahu se pohyboval průtok vzduchu (MAF senzor) během jízdy ráno dne 26. dubna 2025 pro vozidlo s VIN `WV2ZZZ7HZBX01237`. Uživatel měl zároveň identifikovat hodnotu, která vystřelila mimo běžný rozsah grafu kolem 8:00. Ani tentokrát neměl uživatel žádný problém s výběrem vozidla ani s nalezením příslušné jízdy. Při práci s dashboardem však narazil na nejasnost ohledně významu skupin s názvy obsahujícími „interval“. Nebylo mu jasné, co tato část názvu skupiny představuje a že obsahuje spojité grafy. Po objasnění této skutečnosti již postupoval bezchybně, až narazil na další drobný problém, kdy nemohl najít tlačítko pro úpravu zobrazení rozsahu grafu. Jakmile jej objevil, podařilo se mu graf správně upravit. Krátce ho však zmátla vizualizace grafu v režimu náhledu editace, kdy se domníval, že již vidí upravený graf s platným rozsahem, a nevšiml si, že šlo pouze o náhled, který neodráží reálnou hodnotu v daný moment. I přes tyto dílčí překážky se mu nakonec podařilo správně odhalit extrémní hodnotu a určit celý rozsah grafu.

Ve čtvrtém úkolu měl uživatel vytvořit novou skupinu s názvem „Kontrola motoru“. Do této skupiny měl nadefinovat tři parametry. Prvním parametrem bylo zatížení motoru, které mělo být definováno jako okamžitá hodnota zobrazovaná ve sloupcovém grafu a zároveň měla být změněna barva na zelenou. Druhým parametrem byly otáčky motoru, které měly být rovněž definovány jako okamžitá hodnota, přičemž bylo potřeba změnit rozsah na 0 až 6500 otáček za minutu a upravit barevné rozlišení tak, aby zelená barva odpovídala rozsahu 400 až 2500 otáček a červená barva zbytku rozsahu. Třetím parametrem byl tlak nasávaného vzduchu, který měl být definován jako spojitý graf (interval) s nastavením dolní hranice na 20 a horní hranice na 150, přičemž pro obě hranice měla být použita červená barva.

Ačkoli uživatel neměl žádné potíže s vytvořením nové skupiny, zpočátku se domníval, že parametry do ní přidává výběrem ze seznamu skupin (podobně jako v prvním testu), a nikoli přímo definováním konkrétních parametrů v editoru skupin. Uživatel nakonec zvládl úspěšně definovat parametry a změnit barvu u prvního parametru. Problém nastal až s nastavením rozsahu barev u parametru dalšího, jelikož mu nebylo zřejmé, že se barva přiřazuje pouze na základě spodní hranice a platí až do další definované hranice (tj. bez výslovného uvedení horního limitu). U třetího parametru se znovu objevil již dříve zmíněný problém s nepochopením významu části názvu parametru „interval“ (podobně jako u skupiny v předchozím testu). Uživateli nebylo jasné, že se jedná o typ spojitého grafu. Nakonec se potýkal i s problémem způsobeným automatickým překladem v rozšíření prohlížeče, které měnilo názvy některých prvků v uživatelském rozhraní, například výraz „Práh grafu“ byl chybně přeložen jako „Pražský graf“, a proto nemohl požadované nastavení snadno najít. Přes všechny tyto obtíže se mu však nakonec podařilo skupinu správně nadefinovat a úspěšně ji uložit.

Z celkového průběhu testování vyplývá, že se uživatel v aplikaci dokázal dobře orientovat a zvládal plnit zadané úkoly, i když narazil na několik dílčích nejasností. Jednalo se zejména o nepochopení některých pojmů (např. význam skupiny s názvem „interval“ jako spojitého grafu), či o obtíže spojené s vizuální interpretací dat (například barevné škály bez definované horní hranice nebo omezení způsobená rozvržením dashboardu). Pozitivně byla hodnocena obecná logika aplikace, způsob vizualizace dat i celkový vzhled rozhraní. Přes zjištěné problémy nezazněl žádný návrh na změnu vzhledu uživatelského rozhraní nebo funkčnosti aplikace. Uživatel uvedl, že by preferoval dostupný návod nebo stručný manuál, který by osvětlil základní ale i některé pokročilejší funkce či terminologii. Na základě této

zpětné vazby byl vytvořen stručný uživatelský manuál, který reflektuje nejčastější problémy a nedorozumění zjištěné během testování a měl by sloužit jako doplněk pro budoucí uživatele při práci s aplikací.

7.3 Potenciální rozšíření

Jak už jsem zmiňoval výše v sekci 7.1, aktuální verze aplikace definuje pouze dvě barvy pro obarvení trasy vozidla na mapě. Výchozí barva trasy, pokud je vše v pořádku, je pevně definována a nastavená na zelenou a pouze sekundární barvu lze měnit. Jako jedno z potenciálních budoucích rozšíření bych umožnil definovat více barev pro obarvení trasy, a tím lépe rozlišit závažnost problému v jejích jednotlivých částech. Zároveň bych přidal možnost pro uživatele měnit všechny barvy včetně té původní a jiných barev pevně definovaných v aplikaci. To by uživatelům umožnilo si aplikaci ještě více přizpůsobit podle svých představ. Jedná se ale pouze o kosmetické úpravy, které zatím nejsou nutné.

Zvládání většího množství dat představuje pro aplikaci určitou výzvu, která si do budoucna nejspíše vyžádá menší optimalizaci. V aktuální verzi aplikace probíhá zpracování dat přímo na straně klienta, což se při testování ukázalo jako plně dostačující pro aktuální použití aplikace. V budoucnu u rozsáhlejšího použití by se však mohlo projevit menší zpomalení, zejména při vykreslování mapy, kde jsou poziční data načítána v sekundovém rozlišení, zatímco ostatní parametry jsou agregovány po minutách. Možným řešením do budoucna je optimalizace zpracování dat na klientské straně nebo jejich asynchronní zpracování, případně přesun části logiky na server. Optimalizace však aktuálně není nutná, protože aplikace i v současné podobě poskytuje plně funkční a dostatečně přívětivou uživatelskou zkušenost.

Dalším možným rozšířením je vylepšení získávání dat ve vozidle a automatizace konfigurace zařízení. Aktuální implementace používá blokující volání, což může způsobit zpoždění mezi dotazy. V budoucnu by bylo možné využít asynchronní přístup pro potenciální zrychlení získávání dat. Dále se plánovalo vytvoření konfiguračních skriptů pro automatické nastavení zařízení, což by usnadnilo nasazení systému a snížilo manuální zásahy. Tato vylepšení nejsou nezbytná, ale mohou zlepšit uživatelskou zkušenost v budoucnu.

Kapitola 8

Závěr

Cílem této bakalářské práce bylo navrhnout a realizovat systém pro sběr, agregaci, ukládání a vizualizaci dat z palubních diagnostických systémů (OBD) vozidel. Na základě průzkumu současných řešení a identifikace jejich omezení, jako je nedostatečná podpora pro zobrazení historických dat nebo absence centralizované správy vozidel, byl navržen nový systém, který klade důraz na snadnou rozšiřitelnost, spolehlivý sběr dat v offline režimu a možnost správy vozidel centrálně přes jedno zařízení.

Součástí řešení je komponenta běžící na zařízení Raspberry Pi, která pravidelně získává diagnostická data z OBD rozhraní vozidla, uchovává je v lokálním úložišti ve formátu CSV a při dostupném Wi-Fi připojení je odesílá do centrálního systému. Data jsou následně vizualizována prostřednictvím webové aplikace, která umožňuje uživatelům přehledně sledovat historii dat, stav jednotlivých vozidel a vyhodnocovat klíčové indikátory výkonnosti (např. spotřebu, zatížení motoru nebo teplotní odchylky).

Při návrhu systému byl kladen důraz na jednoduchost použití a možnost využití aplikace uživateli, kteří nemají velké zkušenosti v oblasti nových technologií, složitých na naučení. Byla rovněž ověřena funkčnost celého systému v reálných podmínkách s více vozidly.

Ačkoliv je systém plně funkční a splňuje stanovené cíle, během vývoje a testování se objevilo pár praktických problémů, které se ale v průběhu vývoje podařilo vyřešit. Jedním z hlavních problémů byla samotná komunikace s OBD zařízením prostřednictvím ELM 327 adaptéru. Tento adaptér se v praxi ukázal jako nespolehlivý, protože často docházelo k výpadkům komunikace, zpožděným odpovědím nebo potřebě opakovaného připojování, což značně komplikovalo pravidelné načítání dat.

Výstupy této práce mohou poskytnout užitečný základ pro další rozvoj v oblasti monitorování vozidel, telemetrie a datové vizualizace. Systém může posloužit jako praktické řešení pro vývojáře a výzkumníky, kteří se zaměřují na zpracování dat z vozidel a jejich aplikaci pro prediktivní údržbu, optimalizaci provozu nebo vývoj aplikací pro správu vozového parku. Systém pro získávání a vizualizaci dat o vozidle ze sběrnice OBD pod názvem *Fleet Vision* je dostupný ve službě GitHub[1] a webová aplikace veřejně nasazená pro testování[2]. Projekt je pod licencí MIT[3].

[1] GitHub: https://github.com/pocketix/fleet-vision
[2] Fleet Vision: https://dexter.fit.vutbr.cz/fleet-vision
[3] MIT licence: https://choosealicense.com/licenses/mit

Literatura

[1] Ashton, K. *The Internet of Things*. 1999. Prezentováno na konferenci o IoT, Procter & Gamble, USA.

[2] Ashton, K. et al. That 'internet of things' thing. *RFID journal*. Hauppauge, New York, 2009, sv. 22, č. 7, s. 97–114.

[3] Azzawi, M. A.; Hassan, R. a Bakar, K. A. A. A review on Internet of Things (IoT) in healthcare. *International Journal of Applied Engineering Research*, 2016, sv. 11, č. 20, s. 10216–10221.

[4] Bach, B.; Freeman, E.; Abdul Rahman, A.; Turkay, C.; Khan, S. et al. Dashboard design patterns. *IEEE Transactions on Visualization and Computer Graphics*. IEEE, 2022, sv. 29, č. 1, s. 342–352.

[5] Baker, J.; Jones, D. a Burkman, J. Using visual representations of data to enhance sensemaking in data exploration tasks. *Journal of the Association for Information Systems*, 2009, sv. 10, č. 7, s. 2.

[6] Bedi, G.; Venayagamoorthy, G. K. a Singh, R. Internet of Things (IoT) sensors for smart home electric energy usage management. In: IEEE. *2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*. 2016, s. 1–6.

[7] Bureš, M. *Systém pro zpracování dat z chytrých zařízení*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Hynek, Ph.D.

[8] Few, S. *Information dashboard design: The effective visual communication of data*. O'Reilly Media, Inc., 2006.

[9] Franek, J. Optimalizace sít'ových topologií: 3-tier vs. Spine-and-Leaf. online, 2023. Dostupné z: https://www.academia.edu/download/111350234/zaverecna_prace_4_.pdf.

[10] Godavarty, S.; Broyles, S. a Parten, M. Interfacing to the on-board diagnostic system. In: IEEE. *Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000. 52nd Vehicular Technology Conference (Cat. No. 00CH37152)*. 2000, sv. 4.

[11] González, C.; Daniel, G. a Cueva Lovelle, M.-L. C. P. G.-B. J. M. A review about Smart Objects, Sensors, and Actuators. *International Journal of Interactive Multimedia and Artificial Intelligence*, 2017, sv. 4, č. 3. ISSN 1989-1660. Dostupné z: http://www.ijimai.org/journal/sites/default/files/files/2016/08/ijimai20174_3_1_pdf_15525.pdf.

[12] Gosselin, M. An empirical study of performance measurement in manufacturing firms. *International journal of productivity and performance management*. Emerald Group Publishing Limited, 2005, sv. 54, 5/6, s. 419–437.

[13] Greengard, S. *The Internet of Things*. Cambridge, Massachusetts: MIT Press, 2015. 13–15 s. The MIT Press Essential Knowledge series. ISBN 9780262527736.

[14] Ing. Hynek Jiří. *Impact of subjective visual perception on automatic evaluation of dashboard design*. Brno, 2019. PhD thesis. Brno University of Technology, Faculty of Information Technology. Vedoucí práce prof. Ing. Tomáš Hruška, CSc.

[15] Ing. John Petr. *Optimising processes in IoT*. Brno, 2024. PhD thesis proposal. Brno University of Technology, Faculty of Information Technology. Vedoucí práce prof. Ing. Tomáš Hruška, CSc.

[16] International Business Machines Corporation ("IBM"). What is the Internet of Things (IoT)? *IBM* online. Dostupné z: https://www.ibm.com/topics/internet-of-things. Path: Home; Topics; Internet of Things.

[17] Ishaq Bhatti, M.; Awan, H. M. a Razaq, Z. The key performance indicators (KPIs) and their impact on overall organizational performance. *Quality & Quantity*. Springer, 2014, sv. 48, s. 3127–3143.

[18] Janes, A.; Sillitti, A.; Succi, G. et al. Effective dashboard design. *Cutter IT journal*, 2013, sv. 26, č. 1, s. 17–24.

[19] Johnson, J. *Designing with the mind in mind: simple guide to understanding user interface design guidelines*. Morgan Kaufmann, 2020.

[20] Koffka, K. Perception: an introduction to the Gestalt-Theorie. *Psychological bulletin*. Psychological Review Company, 1922, sv. 19, č. 10, s. 531.

[21] Köhler, W. An aspect of Gestalt psychology. *The Pedagogical Seminary and Journal of Genetic Psychology*. Taylor & Francis, 1925, sv. 32, č. 4, s. 691–723.

[22] Kuznetsova, A. Gestalt Principles in Data Visualization. online, Listopad 2023. Dostupné z: https://nastengraph.medium.com/gestalt-principles-in-data-visualization-a4e56e6074b5.

[23] Lasi, H.; Fettke, P.; Kemper, H.-G.; Feld, T. a Hoffmann, M. Industry 4.0. *Business & information systems engineering*. Springer, 2014, sv. 6, s. 239–242.

[24] Lillo, J.; Moreira, H. a Spilberger, C. Color blindness. *Encyclopedia of applied psychology*. Elsevier Academic Press, Oxford, 2004, s. 411–420.

[25] Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H. et al. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal*, 2017, sv. 4, č. 5, s. 1125–1142.

[26] Logimic. Platforma pro vaše zařízení. *ACADA Iot Platforma* online. 2021. Dostupné z: https://www.logimic.com/cs/platforma/. Path: Home; ACADA Iot Platforma.

[27] Madakam, S.; Ramaswamy, R. a Tripathi, S. Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications* online, 2015, sv. 3, č. 5, s. 164–173. ISSN 2327-5219, 2327-5227. Dostupné z: https://doi.org/10.4236/jcc.2015.35021. [cit. 2024-17-09].

[28] Mangalvedhe, N.; Ratasuk, R. a Ghosh, A. NB-IoT deployment study for low power wide area cellular IoT. In: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. Valencia, Spain: IEEE, Září 2016, s. 1–6. ISBN 978-1-5090-3254-9. Dostupné z: Availableat:http://ieeexplore.ieee.org/document/7794567/.

[29] Miao Wu, Ting-Jie Lu, Fei-Yang Ling, Jing Sun, Hui-Ying Du. Research on the architecture of Internet of Things. In: *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)* online. Chengdu, China: IEEE, Srpen 2010, sv. 5, s. V5–484–V5–487. Dostupné z: https://doi.org/10.1109/ICACTE.2010.5579493.

[30] Mohamed, K. S. *The Era of Internet of Things: Towards a Smart World*. 1. vyd. Springer Cham, 2019. ISBN 978-3-030-18132-1 978-3-030-18133-8. Dostupné z: http://link.springer.com/10.1007/978-3-030-18133-8.

[31] Monigatti, L. Why Your Data Visualizations Should Be Colorblind-Friendly: Especially if You Are Trying to Convince Men. *Towards Data Science* online, Červenec 2022. Dostupné z: https://towardsdatascience.com/is-your-color-palette-stopping-you-from-reaching-your-goals-bf3b32d2ac49.

[32] Parmenter, D. *Key performance indicators: developing, implementing, and using winning KPIs*. John Wiley & Sons, 2015.

[33] Rahman, A. A.; Adamu, Y. B. a Harun, P. Review on dashboard application from managerial perspective. In: IEEE. *2017 International Conference on Research and Innovation in Information Systems (ICRIIS)*. 2017, s. 1–5.

[34] Shah, K.; Sheth, C. a Doshi, N. A survey on iot-based smart cars, their functionalities and challenges. *Procedia Computer Science*. Elsevier, 2022, sv. 210, s. 295–300.

[35] Siddiqui, A. S.; Gui, Y.; Plusquellic, J. a Saqib, F. Secure communication over CANBus. In: IEEE. *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2017, s. 1264–1267.

[36] Spurgeon, C. E. *Ethernet: the definitive guide*. "O'Reilly Media, Inc.", 2000.

[37] Stiller, B.; Schiller, E.; Schmitt, C.; Ziegler, S. a James, M. An overview of network communication technologies for IoT. *Handbook of Internet-of-Things*. Springer Int, 2020, sv. 12.

[38] Tufte, E. R. Envisioning information. *Optometry and Vision Science*. LWW, 1991, sv. 68, č. 4, s. 322–324.

[39] Wierzbicki, S. Evaluation of on-board diagnostic systems in contemporary vehicles. *Diagnostyka*. Polska Akademia Nauk. Polskie Towarzystwo Diagnostyki Technicznej PAN, 2011, 3 (59, s. 35–40.

Příloha A

Obsah externí přílohy

- Tento dokument ve formátu PDF.
- manual.pdf – manuál pro uživatelskou aplikaci ve formátu PDF.
- latex – zdrojové soubory dokumentu.
- source/car – Skripty pro zpracování dat na úrovni vozidla.
- source/frontend – Quasar aplikace.
- source/README.md – popis repozitáře.

Příloha B

Testovací scénáře

Testovací scénář 1: Cílem testu bylo ověřit, zda je možné prostřednictvím uživatelského rozhraní zjistit hodnotu teploty chladicí kapaliny pro konkrétní vozidlo na základě zadaného VIN a přesného časového údaje v rámci jedné jízdy.

Testovací scénář 2: Test byl zaměřen na ověření, zda lze zjistit přibližnou geografickou polohu vozidla v daný čas a současně určit nadmořskou výšku spolu s informací o směru jízdy (stoupání/klesání).

Testovací scénář 3: Úkolem testu bylo prověřit, zda lze z grafu správně odečíst rozsah průtoku vzduchu zaznamenaného senzorem MAF během určité jízdy, včetně identifikace hodnoty vybočující z běžného rozsahu.

Testovací scénář 4: Cílem testu bylo ověřit schopnost uživatele vytvořit novou skupinu parametrů v rozhraní a správně nakonfigurovat vizualizaci tří specifických datových parametrů dle zadaných požadavků na typ zobrazení, rozsahy a barevné rozlišení.