

# **Webová aplikace pro zpracování signálů**

## **A Signal Web Application**

Michal Kmit'

---

Bakalářská práce  
2017

 Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2016/2017

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal Kmit'**  
Osobní číslo: **A14576**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Informační technologie v administrativě**  
Forma studia: **prezenční**

Téma práce: **Webová aplikace pro vizualizaci signálů**  
Téma anglicky: **A Web Application for Signal Visualisation**

Zásady pro vypracování:

1. Popište základní typy grafů, které jsou vhodné pro vizualizaci signálů.
2. Seznamte se s existujícími knihovnamy v JavaScriptu, užívanými pro vykreslování grafů.
3. S ohledem na reálná data a požadavky na vizualizaci signálů proveďte analýzu a porovnání vybraných knihoven.
4. Vytvořte vizuální webové aplikace založené na vybraných knihovnách, které budou splňovat požadavky stanovené s ohledem na použití v průmyslové praxi.
5. Zdrojové kódy aplikace doplňte podrobným a srozumitelným komentářem tak, aby se daly v budoucnu snadno upravovat nebo rozšiřovat.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. SUEHRING, Steve. JavaScript: krok za krokem. Brno: Computer Press, 2008. Krok za krokem (Computer Press). ISBN 978-80-251-2241-9.
2. FLANAGAN, David. JavaScript: kompletní průvodce. 2. aktualiz. vyd. Praha: Computer Press, 2002. Všechny cesty k informacím. ISBN 80-722-6626-8.
3. Tvorba efektivních grafů: Doporučení pro diplomovou práci [online]. Brno [cit. 2017-01-05]. Dostupné z: [http://is.muni.cz/th/362075/ff\\_m/tvorba-efektivnich-grafu.pdf](http://is.muni.cz/th/362075/ff_m/tvorba-efektivnich-grafu.pdf). Masarykova univerzita.
4. BOSTOCK, Mike. D3.js: Data-Driven Documents. Data Driven Documents: JavaScript Library [online]. 2015 [cit. 2017-01-05]. Dostupné z: <https://d3js.org/>
5. JQuery. ČÁPKA, David. IT Network [online]. [cit. 2017-01-05]. Dostupné z: <http://www.itnetwork.cz/javascript/jquery-zaklady>

Vedoucí bakalářské práce:

**Ing. Pavel Pokorný, Ph.D.**

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

**3. února 2017**

Termín odevzdání bakalářské práce:

**30. května 2017**

Ve Zlíně dne 3. února 2017



doc. Mgr. Milan Adámek, Ph.D.  
*děkan*



Ing. Miroslav Matýšek, Ph.D.  
*ředitel ústavu*

**Jméno, příjmení: Michal Kmit'**

**Název bakalářské/diplomové práce: Webová aplikace pro zpracování signálů**

**Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 16. 5. 2017

.....  
podpis diplomanta

## **ABSTRAKT**

Cílem zadané práce je prostudovat vybrané knihovny pro práci s grafy za pomoci programovacího jazyku JavaScript. V teoretické části zadání jsou popsány základní vlastnosti těchto knihoven, kde jsou vyzdviženy jejich funkce pro vykreslování a práce s nimi. Dále v této části, jsou popsány grafy, které budou reprezentovat výstup práce. Následně je vytvořena analýza porovnání popsaných knihoven. Dle nejlépe dostupných informací a porovnání knihoven bude na tomto základě vybrána nejvhodnější a použita v praktické části. Výstupem práce bude webové rozhraní v HTML pro vizualizaci signálů v zobrazující podobě grafů.

Klíčová slova: JavaScript, knihovna, graf, vizualizace, HTML

## **ABSTRACT**

The aim of thesis is to study selected libraries for working with graphs using the JavaScript programming language. In the theoretical part, the basic features of these libraries are described, where their functions for rendering and working with them are highlighted. Further, in this section, charts are described that will represent the output of the work. Next step is comparison analysis of the described libraries. According to the best available information and comparison of libraries, this will be the most appropriate and used in the practical part. The output of the work will be web interface in HTML for visualization of signals in graphical form.

Keywords: JavaScript, library, visualization, HTML

Děkuji Ing. Pavlu Pokornému Ph.D. za pomoc při vedení bakalářské práce. Mé poděkování patří též p. Lukáši Duffkovi za spolupráci při získávání údajů pro výzkumnou část práce.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

## OBSAH

ÚVOD.....	9
<b>I TEORETICKÁ ČÁST.....</b>	<b>10</b>
<b>1 PREZENTACE DAT .....</b>	<b>11</b>
1.1 GRAFICKÁ PREZENTACE .....	11
1.1.1 Spojnicový graf .....	11
1.1.2 Sloupcový graf .....	12
1.1.3 Výšečový graf .....	13
1.1.4 Histogram .....	13
<b>2 VYBRANÉ KNIHOVNY PRO VYKRESLOVÁNÍ GRAFŮ .....</b>	<b>15</b>
2.1 POUŽITÉ TECHNOLOGIE .....	15
2.1.1 JavaScript .....	15
2.2 PHP.....	16
2.3 JQUERY .....	17
2.4 VSTUPNÍ DATA .....	17
2.4.1 Formát JSON.....	17
2.4.2 Formát CSV .....	18
2.5 TECHNOLOGIE VYKRESLOVÁNÍ .....	18
2.5.1 VML.....	18
2.5.2 SVG.....	18
2.5.3 Canvas .....	19
2.6 SROVNÁNÍ SVG VS CANVAS .....	19
2.7 D3 – DATA DRIVEN DOCUMENTS.....	20
2.8 N3-CHARTS.....	22
2.9 GOOGLE CHARTS .....	23
2.10 CHARTJS.....	24
2.11 AMCHARTS.....	24
2.12 FLOT .....	26
2.13 ZING CHART.....	27
2.14 MXGRAPH .....	28
2.15 SROVNÁNÍ VYBRANÝCH KNIHOVEN.....	29
<b>II PRAKTICKÁ ČÁST .....</b>	<b>31</b>
<b>3 HLAVNÍ POŽADAVKY .....</b>	<b>32</b>
<b>4 VÝSTUPNÍ ANALÝZA VYBRANÝCH KNIHOVEN.....</b>	<b>33</b>

4.1	D3.....	33
4.2	N-3 CHARTS.....	34
4.3	GOOGLE CHARTS .....	36
4.4	CHARTJS.....	37
4.5	AMCHARTS .....	39
4.6	FLOT .....	40
4.7	ZINGCHART.....	41
4.8	MXGRAPH.....	43
4.9	VSTUPNÍ DATA .....	43
4.10	UKLÁDÁNÍ KONFIGURACE GRAFU.....	45
<b>5</b>	<b>VÝSTUPNÍ APLIKACE .....</b>	<b>46</b>
5.1	POPIS KONFIGURACE: .....	47
5.2	PLUGIN PRO ZMĚNU BARVY SIGNÁLU .....	48
5.3	RYCHLOST ODEZVY KNIHOVEN .....	49
<b>6</b>	<b>SROVNÁNÍ WEBOVÝCH APLIKACÍ .....</b>	<b>50</b>
6.1	AMCHARTS .....	50
6.2	GOOGLE CHARTS .....	50
6.3	ZINGCHART.....	50
6.4	PROTOTYP D3 .....	50
	<b>ZÁVĚR .....</b>	<b>51</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>52</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>55</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>56</b>
	<b>SEZNAM TABULEK.....</b>	<b>57</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>58</b>



## ÚVOD

Sběr a zpracování dat v dnešní době má veliký rozmach, než kdy dříve, proto se začínají využívat stále více nástroje, které dokáží reprezentovat informace do čitelné podoby. Jednou z vhodnějších cest pro vizualizaci vybraných informací je grafická reprezentace v podobě grafů, které slouží pro lepší rozhodování při statistických záměrech. Všechny zpracované informace jsou ukládány do informačních systémů v různých podobách, proto je nutné data roztřídit a ucelit do formátu, které ulehčí cestu při vizualizaci. Práce se zabývá vizualizací dat v podobě grafů výhradně liniové podoby. První část práce popisuje jednotlivé zobrazovací techniky v podobě typových grafů, se kterými se běžně setkáme na Internetu nebo v určených aplikacích.

Další obsah teoretické části je věnována konkrétním knihovnám pro vizualizaci dat. Následný popis dostupných technologií pro knihovny je v další části srovnání a shrnutí funkcionalit, které vychází z vybraných knihoven. Poslední část teorie je zaměřena na způsob vykreslování, použitých technologií a datových formátů. Vykreslování nabízí různé způsoby zobrazení obsahu na webové stránce podle požadavků. Aby bylo možné vizualizovat dílčí průběhy, je nutné načíst zdroj dat v obsažené podobě. Pro vypracování webové aplikace jsou použity technologie HTML5, JavaScript a framework jQuery usnadňující tvorby konkrétních funkcí.

Praktická část práce je zaměřena na analýzu vybraných knihoven, které jsou otestovány z reálných průmyslových dat, které byly pořízeny měřením při průběhu tlaku pneumatiky na optický senzor. Z testovaných knihoven budou vybrány nejvhodnější, které dokáží splnit nejvíce požadavků pro sestavení webové aplikace a následně budou interagovat s uživatelem. Vybrané knihovny budou umět vykreslovat jednotlivé hodnoty z měření tak, aby měly čitelnou podobu. Výsledná aplikace může být použita v jiných oblastech zpracování.

## I. TEORETICKÁ ČÁST

## 1 PREZENTACE DAT

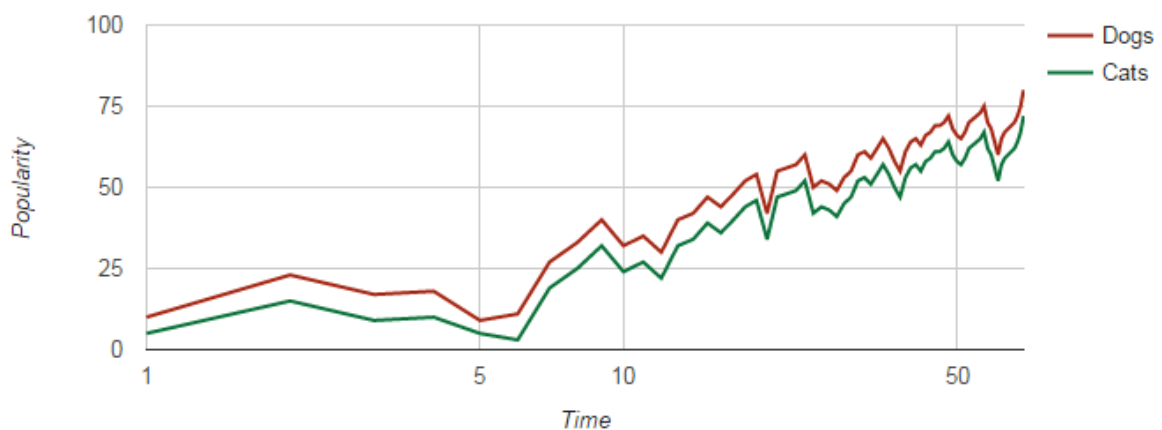
### 1.1 Grafická prezentace

Pokud je velké kvantum dat, tak je potřeba zobrazit v čitelné podobě s jednoduchým výstupem. Grafický výstup je závislý na druhu získaných informací, proto je potřeba zvolit vhodnou podobu zobrazení. V tomto směru je nejlepší volba použít grafy, které vyobrazí velké množství informací na požadovaný výstup. Samotná reprezentace nemusí být v podobě grafu, ale i pomocí mapových myšlenek, EPC diagramů a mnoho dalších metod. [1]

V běžném užití se můžeme setkat přímo s programy, které slouží pro tvorbu grafů v případě volně stažitelné aplikace Graph. Na druhou stranu pokud potřebujeme více vstupních údajů nebo jiné požadavky je možné použít z některých existujících tabulkových procesorů například Excel, který nabízí mnohem obsáhlejší práci. V odkázané literatuře z předchozího odstavce můžeme shlédnout další typy grafů, které mají své specifika pro vstupní informace, tudíž každý graf má svůj účel. Následující kapitola je zaměřena na grafy, které jsou v oblasti zpracování velkého množství dat významné.

#### 1.1.1 Spojnicový graf

Tento graf byl zvolen jako hlavní požadavek průběhu měření tlaku na optický senzor pneumatik, protože splňuje nejlepší vizualizační výstupy veličin ve sledovaném čase. Prvním požadavkem je výpis velkého množství údajů, které graf musí vhodně vykreslit, zvolením požadovaných velikostí os, velikost mřížky a jednoduchý přehled legend průběhů, které si můžeme ve výsledku volit, pokud bude výstup mít více průběhů.



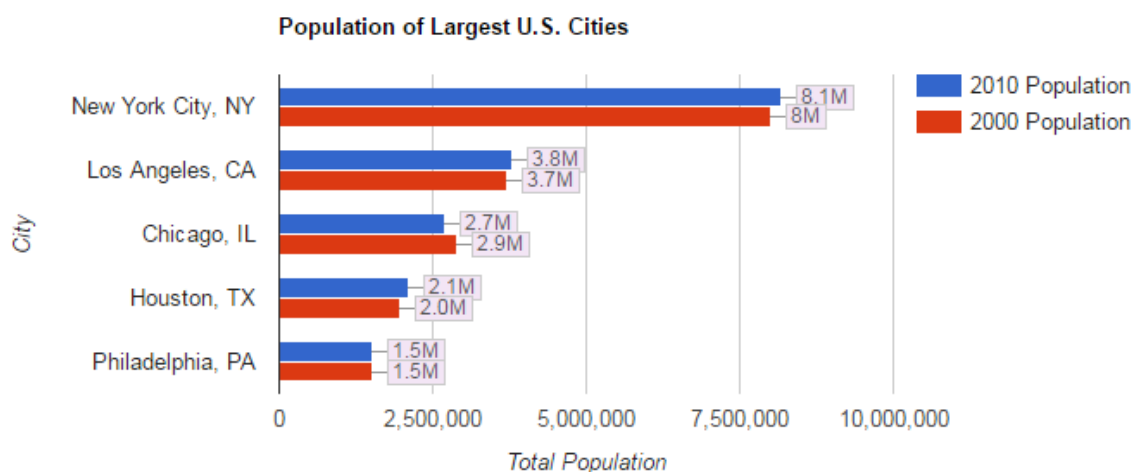
Obrázek č. 1 - Ukázka spojnicového grafu [4]

Patří mezi velice jednoduché grafy jako nástroj technické analýzy, protože pro jeho reprezentaci je požadováno sestavení průběhu s velkým množstvím dat. Jednotlivé kroky vytvoří výslednou spojitou čáru v grafu. Díky jednoduchosti a přehlednosti můžeme vidět jako nevýhodu sémantiky informací s porovnáním ostatních grafů. [6]

Soubor vložených datových hodnot nebo bodů, které zobrazují jednoduché vnímání přírůstku nebo pokles v určitém průběhu, nejčastěji v podobě časového intervalu. Tento postup můžeme také aplikovat při zápisu to sloupcového grafu, kde ovšem spojnicové body postrádáme. Díky spojnicím můžeme vidět podrobně informaci vyobrazení na ose jednoho kroku. [7]

### 1.1.2 Sloupcový graf

Společně s výšečnými a spojnicovými grafy patří mezi nejvíce používané grafy. Tento graf vyobrazuje časové změny v daném období nebo porovnává vybrané kvantitativní položky. Výsledek grafu můžeme pozorovat v podobě sloupců v požadované šířce a výšce zobrazených os. Interpretace údajů mohou vyobrazeny na zvoleném typu sloupců, které mohou být vertikální nebo horizontální. Graf je konstruován tak, že výška jednotlivého sloupce je přímo úměrná cílové skupiny informací v ose Y. Osa X znázorňuje pouze názvy cílových kategorií bez statistických údajů. Mezery mezi jednotlivými sloupci představují diskretnost hodnot v představované ose. Pokud máme různé hodnoty v konkrétní kategorii, tak lze tyto údaje vrstvit vedle sebe nebo i za sebou. [2] [3]

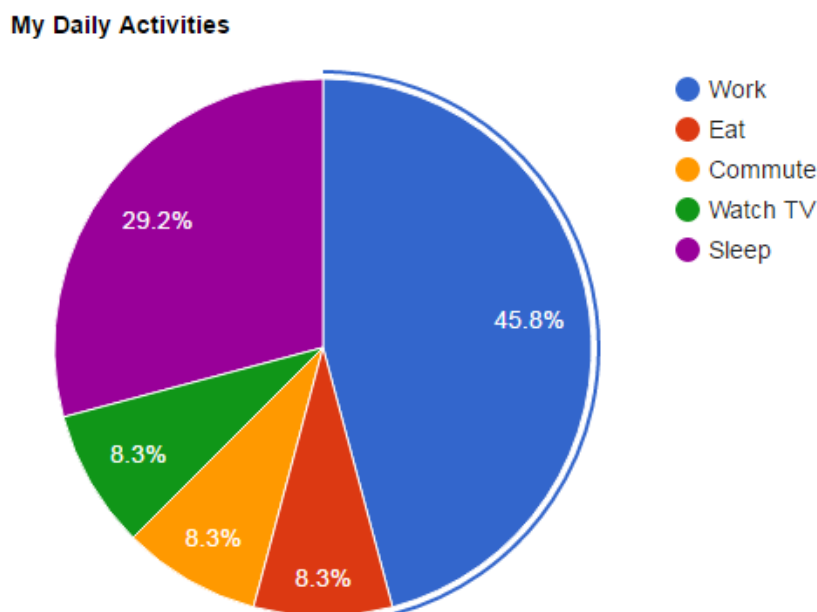


Obrázek č. 2 - Ukázka horizontálního sloupcového grafu [4]

### 1.1.3 Výsečový graf

Graf, který je často označován za koláčový nebo segmentový, který slouží jako nástroj ve statistice. Okolo kruhu je definovaná proměnná, která je graficky znázorněna ve směru hodinových ručiček nebo proti směru. Proměnné, jež jsou závislé na vstupních datech, obvykle reprezentací procent, jsou vyobrazeny v obloukovém tvaru. Velikost oblouku je závislý na vstupní kvantitě informací.

Každá část oblouku představuje konstrukci radiálních přímk ze svých končících bodů do středu grafu. Výsledkem je klínový segment, přičemž nezávislá proměnná může dosáhnout konečných diskrétních hodnot. V závislém kvadrantu obvykle nabývá hodnot od 0% do 100%. Výsledný graf je čitelný, pokud obsahuje malé množství kategorií, v opačném případě ztrácí význam přehlednosti dat. [8]

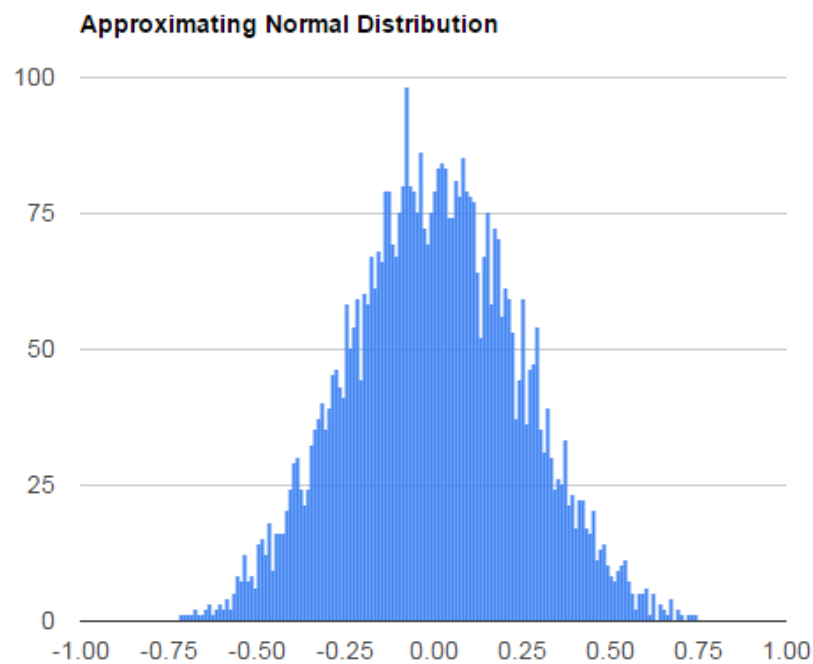


Obrázek č. 3 - Ukázka výsečového grafu [4]

### 1.1.4 Histogram

Představuje grafické znázornění číselné řady v daném intervalu. Vyjadřuje odhad pro rozdělení pravděpodobností jedné spojité proměnné, která představuje kvantitativní počet. Pro vyhotovení histogramu je nutný proces sestavení tzv. bin (datové předzpracování informací pro snížení chyb v pozorování), následně je celý obsah rozdělen na intervaly, kde každý interval má přiřazenou hodnotu. Intervaly jsou řazeny systematicky za sebou.

Po sobě jdoucí intervaly jsou označovány jako proměnná v čase. Sousední intervaly nemají mezi sebou mezery, čímž představují kontinuální proměnnou. Histogramy zobrazují hrubou představu distribuci dat či odhad četnosti neboli výskytu. Tento nástroj představuje jeden aplikaci ze sedmi pro řízení kontroly kvality. Využití histogramu oproti sloupcovému grafu je v návaznosti intervalů v řadě, sloupcový může obsahovat mezery, které mohou značit rozdíly v kategoriích. [9]



Obrázek č. 4 - Znáznornění histogramu [4]

## 2 VYBRANÉ KNIHOVNY PRO VYKRESLOVÁNÍ GRAFŮ

V programovacích jazycích bývá knihovna označována jako předpřipravený celek procedur, funkcí. Tímto se usnadňuje práce programátora, který může použít již připravený zdrojový kód, který dále je možný sdílet mezi více programy a lze ho libovolně upravovat, pokud je open source. Knihovna poskytuje rozhraní API, které poskytuje volání funkcí z knihovny. Můžeme se setkat s několika druhy knihoven v závislosti požadovaných funkcí a výstupů. Pro obecnou grafickou reprezentaci dat se obvykle využívají knihovny DirectX, OpenGL nebo SDL, v oblasti webových aplikací to bývá SVG, Canvas, případně WebGL. Tato práce je zaměřena na webové aplikace vizualizace grafů. V této oblasti tvorby aplikací dominuje JS (JavaScript). V následujících kapitolách jsou uvedeny nejpoužívanější JavaScriptové knihovny, které se používají v oblasti vizualizace grafů. Nejprve budou popsány technologie, které jsou použité v bakalářské práci a následně vybrané knihovny. [10]

### 2.1 Použité technologie

V dnešní době se můžeme setkat s mnoha způsoby tvorby dynamického obsahu stránek z pohledu programování. Na programovací jazyky pro web se můžeme dívat ze dvou pohledů. První je ze strany klienta a používá se ke kontrole dat přímo ve webovém prohlížeči uživatele, kde nejběžněji je JS nebo AJAX technologie. Tato možnost řešení je lepší v směru rychlosti přístupu a výkonu na úkor menší bezpečnosti. Když se podíváme ze strany serveru, tak všechny vykonané operace jsou prováděny přímo na serveru, tudíž data se musí posílat. Nejvíce používané skriptovací jazyky na straně serveru je PHP, Python nebo Pearl.

V této práci je použit skriptovací jazyk JavaScript, pro svou jednoduchost a robustnost, čímž v dnešní době našel nejlepší uplatnění. Součástí JS mohou být rozšiřující frameworky AngularJS nebo jQuery, které usnadňuje implementaci složitějších funkcí a otevírá další možnosti programování složitějších aplikací. [12][13]

#### 2.1.1 JavaScript

JavaScript je moderní programovací jazyk se základy objektového přístupu. Základní jádro bylo původně vloženo do NetScape Navigatoru, Internetu Explorer, Googlu a dalších webových prohlížečů. Pro editaci vzhledu programovatelných komponent bylo přidáno

okno prohlížeče a jeho obsah. Význam je ten, že JS umožnit vložit do HTML obsah, který nemusí být nutně statický nýbrž dynamický. Ve výsledku tvoří komunikaci mezi uživatelem, serverem a řídí obsah webu.

Jádro JS svými příkazy a konstrukcí připomíná jazyk C++. Syntaxí nekončí vše, tento jazyk je bez typové hodnoty a to znamená, že není nutné definovat typ proměnné. Objekty v JS mají velkou shodu s jazykem Pearl. Tento jazyk není kompilovatelný, čímž není nutné mít nástroje pro tvorbu zdrojových kódů. K vytvoření kódu postačí obyčejný poznámkový blok. Tento způsob tvorby JS aplikací je virtuálně limitovaný. V praxi to znamená, že užití např. Visual Studia si usnadníme práci, při tvorbě webových stránek. [14] [15]

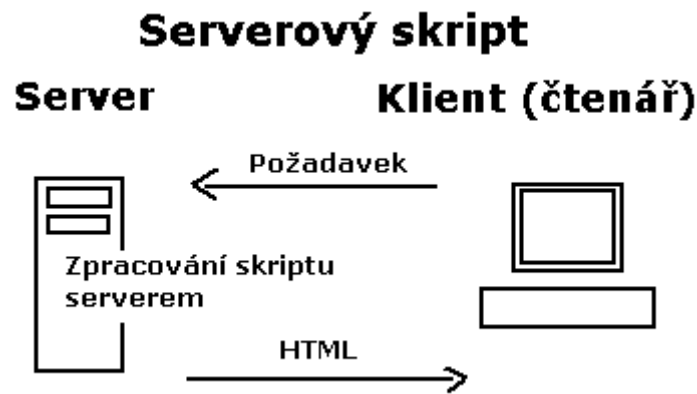
Ze strany klienta můžeme JS charakterizovat, že překladač je vložen do webového prohlížeče. Tento způsob prezentace je nejběžnější, se kterým se můžeme setkat. Klientský jazyk kombinuje možnost zpracování skriptů interpretem JS s objektovým modelem dokumentů DOM – definovaným webovým prohlížečem. Soulad obou technologií umožní synergický efekt vzájemného doplňování. Klientský JS poskytuje distribuci spustitelného obsahu v dynamické podobě HTML stránek. [15]

## 2.2 PHP

Skriptovací jazyk sloužící pro tvorbu dynamické webové stránky, podporuje také objektově orientované programování. Syntaxe je velice podobná jazyku C a tím je zaručena jednoduchost při programování. Jazyk je zcela multiplatformní, což je velká výhoda a vývoj může probíhat nezávisle na jakémkoliv systému nebo hostovaném serveru. PHP pracuje na straně serveru, kde uživatel vytvoří požadavek, který je sestaven a následně odeslán na server v čistém HTML kódu.

Zdrojový kód nelze zobrazit na webové stránce, protože je odesílán na server a tím je zaručena větší bezpečnost oproti spouštění kódu na straně klienta, které je běžné u JavaScriptu. PHP si velice rozumí s technologií Apache, která zajišťuje komunikaci s klientem. Další výhodou je snadné připojení k databázovým systémům jako je MySQL. V jazyku PHP je možné vytvořit redakční systémy. Eshopy, informační systémy nebo dynamické osobní portfolio. Nevýhody jsou, že jazyk není kompilovaný, ale interpretovaný, dále ten kdo má přístup k serveru, může nahlédnout na veškerý obsah skriptů. [34] [35]





Obrázek č. 5 – Ukázka zpracování PHP

## 2.3 jQuery

JS knihovna, která má jednoduchou syntaxi a odstraňuje nekompatibilitu mezi prohlížeči a tím funguje zcela na všech platformách. Zjednodušuje práci s obsahem webu, protože používá CSS selektory, které se používají v běžném CSS stylu, kde selektory zjednodušují práci tím, že si volíme přesně ten prvek, který chceme editovat. Zdrojová knihovna má dva způsoby načítání. V prvním případě můžeme knihovnu stáhnout jako soubor a umístit do cílového adresáře. Druhá možnost je spojení přes Google, kde je knihovna nahrána a můžeme k ní přistupovat a ukládá se do vyrovnávací paměti prohlížeče, čímž se urychluje načítání, kratší odezva a zvyšuje efektivnost. [22]

## 2.4 Vstupní data

Pro výstupní vizualizace grafů je potřeba nahrát vstupní data, které mohou být ve formátech JSON nebo CSV. Každý z těchto formátů má specifické vlastnosti a požadavky pro práci. Nejprve je potřeba vybrat vhodné data, ke zpracování dle priorit zobrazení.

### 2.4.1 Formát JSON

Jeden z nejvíce používaných formátů pro výměnu datových souborů mezi počítači na webu. Formát není omezen na žádnou platformu, a tudíž je zcela kompatibilní na všech systémech a prohlížečích. Původní struktura vychází ze základu programovacích jazyků C, C++, Java, Python. Syntaxe zdrojového kódu lze převést do jazyků, které byly vyjmenovány, díky této vlastnosti se stává velice silným nástrojem pro rychlé převody dat na základě požadované platformy. Díky jednoduchosti formátu, lze data rovnou nahrát do základních datových typů, protože neobsahuje složitou strukturu. Pokud tato aplikace není možná je vhodné použít pro implementaci nástroje XPath, XSLT, které v XML dokáží urychlit práci. [30]

### 2.4.2 Formát CSV

Označení formátu pro soubory, sloužící pro přenos dat mezi odlišnými aplikacemi. Typ těchto souborů je uložen v digitální podobě uspořádaných v tabulkách. Záznam tabulky tvoří řádky a sloupce, které jsou odděleny čárkou, kde jeden řádek představuje jednu hodnotu. Dnes jsou standardem pro programy, které umožní export a pracují s databázovými systémy, kde nalezneme strukturované data. Opačný proces v podobě importu dat z CSV do databáze. Díky jednoduchosti má podporu na všech platformách. Programy umožňující s CSV jsou Microsoft Excel, NetBeans IDE nebo OpenOffice.org Calc. [31]

## 2.5 Technologie vykreslování

Je taková technologie, která umožní vykreslovat do stránek HTML pokročilejší grafiku. V minulosti bylo potřeba používat statické obrázky, pluginy typu Flash, anebo formát VML. V dnešní době pro vykreslování slouží nejznámější bitmapové metody pro, které se nazývají SVG a Canvas. Každý formát se od sebe liší vlastnostmi a je potřeba zvolit nejvhodnější za daným účelem. [29]

### 2.5.1 VML

Zastaralý grafický vektorový formát založený na XML, tento jazyk se velice podobá HTML. Pomocí vektorů, křivek můžeme pracovat s grafickým obsahem za použití formátování CSS stylů. VML je od firmy Microsoft a tudíž je proprietární – uzavřený, tudíž je vázaný pouze na aplikace od Microsoftu. Manipulace s obsahem je prováděna přímo ve zdrojovém kódu stránky. Celý obsah je interpretován jako text, takže není bitmapový obrázek, tím umožní vyhledávání a kopírování. Prohlížeče podporující dynamický obsah umí s VML vytvářet interaktivní grafy a animace. Vlastnosti popsaného formátu jsou velice podobné SVG. [28]

### 2.5.2 SVG

Vektorový formát, který je založen na XML. Své možnosti uplatní v obsahu, který může být statický, dynamický, animovatelný včetně interakce, díky vyjmenovaným vlastnostem je velice flexibilní. Stylování a editace je možné za pomoci CSS stylů, HTML i dynamicky s SVG DOM. SVG je možné zapsat za pomoci elementu *object* do webového obsahu. Úpravu lze provádět přímo pomocí JS. Výhoda vektorové grafiky je tak, že je

založena na vykreslování křivek, tudíž rozlišení obrázku je nezávislé na velikosti v obsahu a tím je zachována kvalita při přiblížení. Matematický zápis struktury SVG umožňuje rychlé transformace a animace obrázků i v reálném čase. Nevýhoda formátu, že při složitějším obsahu dokument rychlost vykreslování klesá. [29]

### 2.5.3 Canvas

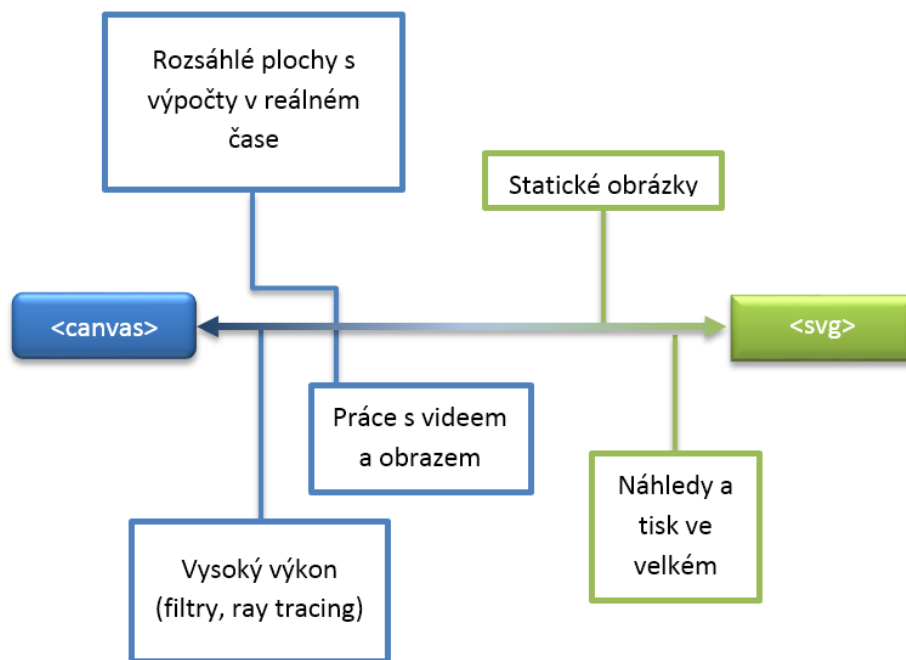
Jednoduché API založené na JS, které je schopno vykreslovat programově. Do webové stránky umí vykreslovat za pomoci definice objektu, který se nazývá Canvas. Kreslení probíhá do 2D výstupu. Nakreslené tvary neobsahují DOM a vše je podobě bitmapové grafiky. Pro opakované vykreslování složitějších objektů se nebude v paměti kumulovat složitost výstupního formátu. Jednotlivé operace je možné cílit přímo na konkrétní pixely a tím je možné vytvořit bitmapové výstupní formáty JPEG, PNG, které je prohlížeč schopen zobrazit a uložit jako výsledný obrázek. V případě animací tento formát není příliš vhodný, protože při každé změně pixelu je potřeba obrázek znovu překreslit. Vhodné zobrazení nachází u interaktivních grafů nebo textu. [29]

## 2.6 Srovnání SVG vs Canvas

První rozdíl mezi způsoby renderování je, že SVG používá křivky při zobrazení, v případě velkých objektů nemá problém rychlého zpracování. Pokud obsah stránek má mnoho objektů, tak se výkon snižuje, objekty fungující na DOM (Document Object Model) navíc zvyšují režii načítání. Veškerá manipulace a interakce probíhá pomocí zdrojového skriptovací kódu a CSS, takže je veškerá dynamika pod kontrolou, avšak je za potřebí více programování oproti Canvas. Přístup k SVG dokumentům je mnohem snadnější, protože je založen na jazyku XML, který umí zpracovat webový prohlížeč. SVG nalézá uplatnění u textů, velkých map a diagramů. V případě užití složitějších obsahů stránek je vhodná kombinace obou technologií SVG a Canvas. Obnova stránek probíhá okamžitě a není potřeba znova opětovné vykreslování obsahu, protože všechno je založeno na vektorech, kde proběhne pouze přepočítání.

Canvas je nejvíce vhodný způsob pro zobrazování velkého počtu objektů v menším měřítku. Využití je vhodné u her a fraktálů, kde je přímý přístup k pixelům, neobsahuje DOM dokumenty, obsah je vykreslen v pixelech. Optimální výkon je dán pomocí pixelů i s větším obsahem obrázků, výkon klesá pouze s rostoucím rozlišením. Přímý přístup

k pixelům v bitmapové podobě, výsledek je možný uložit do nabízených formátů JPG a PNG. Canvas je vhodný pro animace v pozadí. [29] [32]



Obrázek č. 6 – Porovnání technologií Canvas a SVG [32]

## 2.7 D3 – Data Driven Documents

Velká JS knihovna, ze které vychází další knihovny, které mají specifické požadavky pro konkrétní účel plnění. D3 je zaměřena na zpracování dokumentů obsahu dat, která mají svůj specifický význam. Umí oživit data na základě technologie HTML, SVG a CSS, kde je kladen důraz na požadavky moderních prohlížečů, čímž ztrácí podporu u starších prohlížečů. Nativita pro určitou platformu tzv. proprietární rámec, který zajišťuje optimální výkon komponent knihovny řízeným manipulací dat s DOM (Document Object Model). [16]

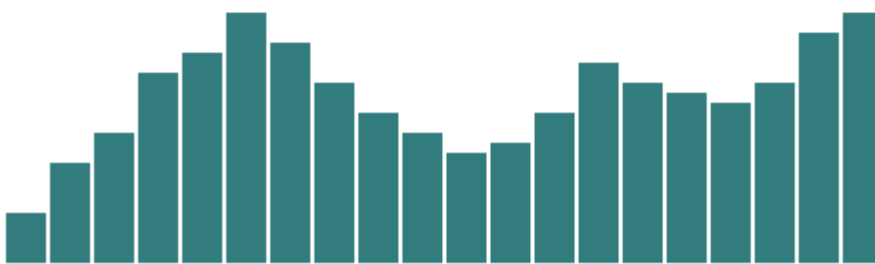
DOM umožňuje nahrát jakoukoliv strukturu dat, kterou je možné aplikovat za pomoci úprav přímo do cílového dokumentu. Možný příklad transformace v D3 je vygenerování HTML kódu do tabulky z pole čísel nebo za použití stejných dat lze vytvořit interaktivní SVG graf s hladkými přechody. Knihovna není monolitický framework, který by se pokoušel poskytnout myslitelné funkce a místo toho řeší jádro problému tím, že efektivně zpracovává dokumenty na základě údajů.

Dynamika knihovny, která se podobá jQuery nebo Prototype, kde jednotlivé atributy, styly lze modifikovat pomocí údajů v D3 a nikoliv jen jednoduchých konstant. Přes veškerou jednoduchost jsou robustním nástrojem *d3.geo.path* funkce, která pracuje se zeměpisnými souřadnicemi do datové struktury SVG.

Obsah knihovny je složen z mnoha funkcí, které můžeme používat opakovaně jako grafické primitiva, oblastní grafy, liniové nebo koláčové. Výhodou knihovny je velká pružnost při načítání větších objemů dat, možností úpravou jen parametrů, které opravdu chceme, snižujeme režii na webový prohlížeč a tím je zachován grafický výkon vykreslování i při vysokém počtu snímků.

```
var dataset = [ 5, 10, 13, 19, 21, 25, 22, 18, 15, 13,
               11, 12, 15, 20, 18, 17, 16, 18, 23, 25 ];

d3.select("body").selectAll("div")
  .data(dataset)
  .enter()
  .append("div")
  .attr("class", "bar")
  .style("height", function(d) {
    var barHeight = d * 5;
    return barHeight + "px";
  });
```



Obrázek č. 7 – vizualizace za pomoci D3 s ukázkou sloupcového grafu [19]

## 2.8 N3-Charts

Představuje volně otevřenou knihovnu pod licencí Open Source a vychází ze základu D3.js. Hlavní struktura vychází z AngularJS, kde je kladen důraz na jednoduchost. Instalace hlavního souboru knihovny může být z lokálního počítače nebo jako internetový odkaz. Hlavní využití je v liniových grafech, které lze zobrazit v různých velikostech zařízení od počítače přes tablet až po mobilní rozložení. Dále obsahuje podporu vedlejších grafů typu oblastních, koláčových, řadových sloupců, formáty vykreslování nachází podporu typu Canvas i SVG.

Výstup grafů lze exportovat do grafických formátů JPEG, PNG nebo možnost manuální převedení do publikačního formátu PDF. Knihovna podporuje potřebné funkce vykreslování os, popisky, mřížky, reakce na události, umístění do kontejneru včetně přesunování. Samozřejmostí je úprava v CSS stylech, úprava výchozích funkcí, přidávání nových funkcionalit a také podpora aplikací z třetích stran. Podporu nalezne u všech známých prohlížečů Google, Firefox, Safari a Internet Explorer. [17]

```
$scope.options = {
  margin: {top: 5},
  series: [
    {
      axis: "y",
      dataset: "numerical",
      key: "val_0",
      label: "A line series",
      color: "hsla(88, 48%, 48%, 1)",
      type: ["line"],
      id: "mySeries0",
      visible: true
    }
  ],
  axes: {x: {key: "x"}}
};
```

Obrázek č. 8 - kód pro liniový graf z N3-Charts [17]

## 2.9 Google Charts

Představuje volně dostupnou službu pro tvorbu grafů od firmy Google. Tato JS knihovna obsahuje velkou nabídku grafů včetně tvorbu spojnicových, výsečových, sloupcových. Knihovna Google Charts je multiplatformní, čímž tvoří sofistikovaný nástroj pro tvorbu i pokročilých webových aplikací, zaručuje také zpětnou kompatibilitu na starších prohlížečích. Vykreslování grafů je založeno na HTML, SVG nebo VML technologii. Google Charts obsahuje nespočet předpřipravených grafů, kde stačí pouze měnit parametry, a tím docílíme požadované změny.

Zdroj předpřipravených objektů a funkcí je získán ze vzdálené databáze na internetu, znamená to, že není potřeba mít uložené lokální soubory knihovny na disku. Grafy jsou vysoce interaktivní, kde za pomoci události se mohou připojit ke složitějším dashboardům. Třída *DataTable* umožní naplnit údaje do grafů, které můžeme lehce měnit na jiné typy, abychom si je přizpůsobily ideálnímu výsledku. Třída *DataTable* umožňuje vyhledávání funkcí, jejich filtraci, které můžeme implementovat do databáze, či našeho webu. [11]

```
<script>
//Instance třídy naplněna daty
var data = new google.visualization.DataTable(),
    data.addColumn('string', 'Popis'); data.addColumn('number', 'Počet');
    data.addRows(
        [
            ['data1', 3],
            ['data2', 1]
        ]
    );
//Instance pro vykreslení grafu
var chart = new google.visualization.PieChart(document.getElementById('chart_div1'));
    chart.draw(data, options);
</script>
//Vykreslení grafu
<div id="chart_div" style="width:400; height:300"></div>
```

Obrázek č. 9 - Ukázka kódu pro vykreslení grafu v Google Charts [11]

## 2.10 ChartJS

Knihovna, která umí vykreslit jednoduché grafy s použitím technologie HTML5. Výstup je založen na grafickém vykreslování Canvas, čímž si zaslouhuje i zpětnou kompatibilitu se staršími prohlížeči. Obsahuje celkem šest základních grafů, čímž je paměťová náročnost knihovny malá. Každý graf je umístěn ve své vlastní třídě, která odděluje základní strukturu ostatních grafů. Díky rozdělení si uživatel zavolá pouze žádaný graf a nemusí se načítat zbytek knihovny. Dále je zde kladen důraz na jednoduchost a rychlost. Grafy se dokážou přizpůsobit požadované velikosti stránky nebo cílového kontejneru umístění. Uplatnění nachází v jednoduchých webových stránkách bez použití složitějších efektů nebo přechodů. Licence knihovny je zcela otevřená – Open Source, čímž zaslouhuje volnou úpravu jádra zdrojového kódu dle svých potřeb. [20]

```
var myChart = new Chart(ctx, {
  type: 'bar',
  data: {
    labels: ['Item 1', 'Item 2', 'Item 3'],
    datasets: [
      {
        type: 'bar',
        label: 'Bar Component',
        data: [10, 20, 30, 40, 50],
      },
      {
        type: 'line',
        label: 'Line Component',
        data: [1, 2, 3, 4, 5],
      }
    ]
  }
});
```

Obrázek č. 10 - Vykreslení liniového grafu [21]

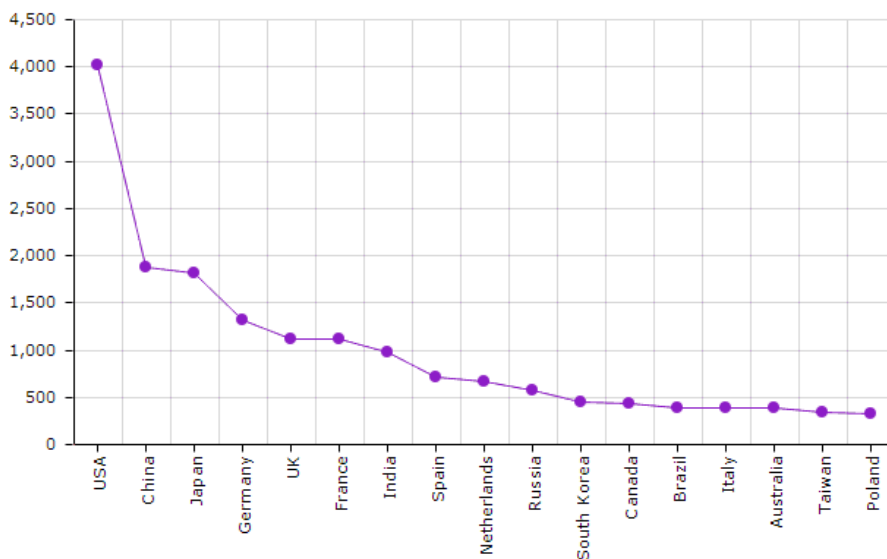
## 2.11 amCharts

Komerční knihovna s velkou nabídkou funkcí, kde je možné pracovat s licencí zadarmo pro vlastní účely. Kompatibilita je zaručena ve všech moderních webových platformách včetně několika starších. Jednoduše si poradí i se složitějšími vizualizacemi velkého počtu vstupních dat včetně kombinací více grafů v jednom. Dynamika knihovny umožňuje zobrazování grafů i v reálném čase bez nutnosti opětovného načítání celé stránky pomocí API, ve kterém lze upravovat vlastnosti.



Knihovna amCharts dokáže načítat data externí v podobě CSV a JSON formátů včetně předzpracování dat, možnost znovu načítání je také možná. Na druhé straně má velkou možnost exportu od bitmapových obrázků JPG a PNG, vektorových formátů SVG, PDF, xls (MS Excel) a CSV. Knihovnu je možné dále rozšířit implementací podporných pluginů, čímž přidává další vlastnosti při reálném nasazení.

K dispozici je plno mnoho použitelných funkcí v podobě umístění grafů, přibližování cílové oblasti, úprava a přepínání legend, bohaté vykreslování dat do HTML přes různé úrovně kombinací grafů. Interaktivita je natolik vysoká, že je možnost grafy přizpůsobit i velice malým rozměrům např. v mobilním zařízení, když jsou vypnuty komponenty legend, popisků os a zarovnání. Jakékoliv změny velikost jsou okamžitě vypočítány a provedeny ve velmi krátkém čase, změny os, rozsah vzdáleností komponent i v případě formulářových vstupů. [23]

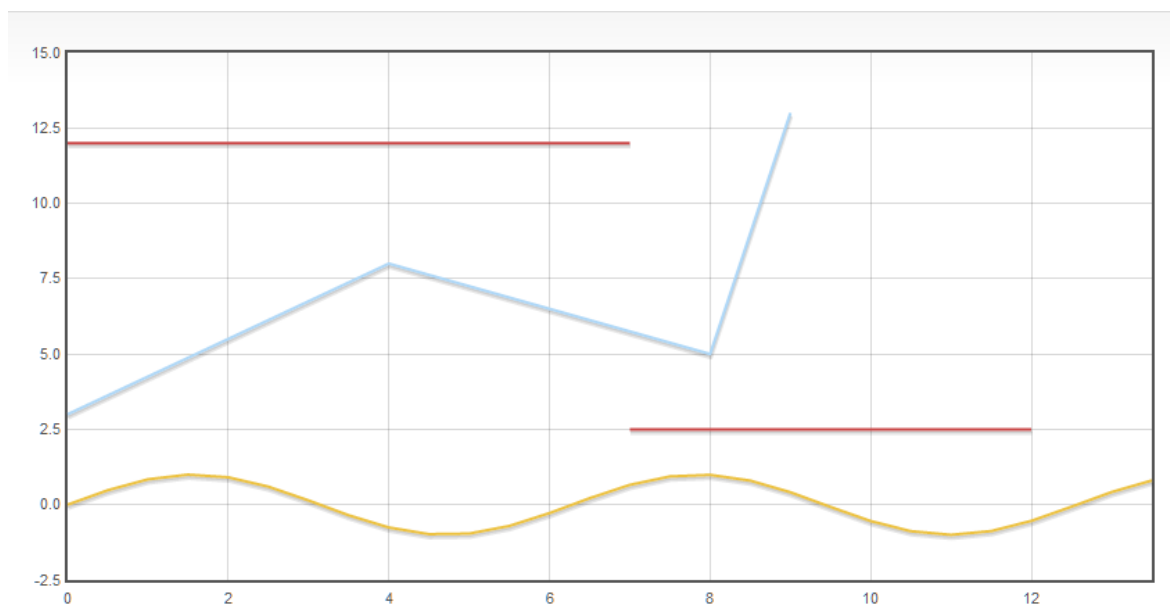


Obrázek č. 11 - Liniový graf z knihovny amCharts [23]

## 2.12 Flot

Knihovna čistě založená na technologii JS na základě jQuery zaměřená na jednoduché použití a interakci s uživatelem bez složitých úprav nebo implementací. Grafický výstup je založen na základě Canvas(u) s technologií HTML5 a nalezneme podporu u všech nových prohlížečů. V případě starších prohlížečů je možné použít emulátor exCanvas nebo FlashCanvas. Flot je licencován pod jménem Apache Licence - MIT, kde je možnost otevřeného kódu tzv. Open Source pro vlastní potřeby a úpravy.

Pro vstupní data používá formát JSON, export grafů je také možný v bitmapových grafických formátech JPEG nebo PNG. Pluginy lze také použít pro rozšíření funkcí i zjednodušení úprav kódů. Knihovna obsahuje základní grafy liniové, sloupcové, koláčové, vyplňování oblastí, různé modifikace v podobě více os a grafů v jednom obsahu, možnost přiblížení konkrétní části grafu je také možná. [24]

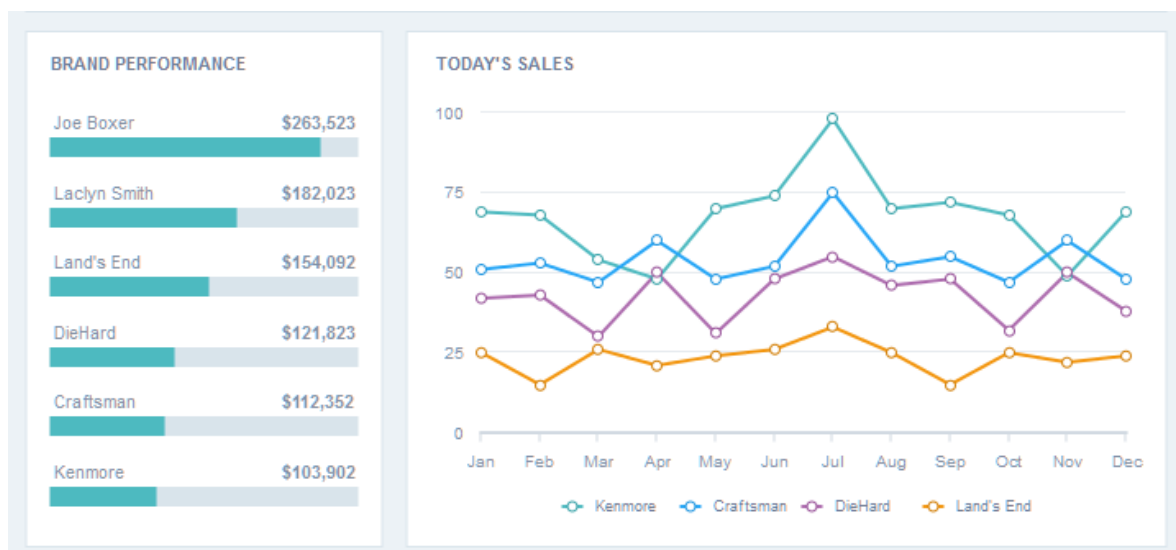


Obrázek č. 12 - Liniový graf knihovny Flot [24]

## 2.13 Zing Chart

Komerční knihovna založena na čistém JS s uzavřeným zdrojovým kódem sloužící pro vykreslování interaktivních grafů při objemných datech s podporujícími vstupními formáty JSON a JS Array. Návrh struktury je takový, aby dokázal zaručený výkon při zpracování výstupů a pokročilých konfigurací. ZingChart poskytuje několik druhů licencí pro zakoupení, kde rozšiřuje další možnosti užití včetně technické podpory.

Technologie vykreslování je SVG nebo Canvas, u starších verzí prohlížečů je nutné použít VML, čímž je zaručena zpětná kompatibilita. Knihovna zahrnuje spolupráci s dalším programovacími jQuery, AngularJS nebo PHP wrapper jazyky včetně frameworků usnadňující snadnější implementaci složitějších funkcí. Software obsahuje integrace demo verze, včetně frameworků např. od Microsoftu C# .NET nebo JavaScriptové frameworky Ember a Backbone, grafy mohou také číst data přímo z databáze MySQL. Knihovna obsahuje nepřeberné množství grafů od základních liniových, sloupcových, koláčových a po animované dashboardy, včetně různých API. [27]

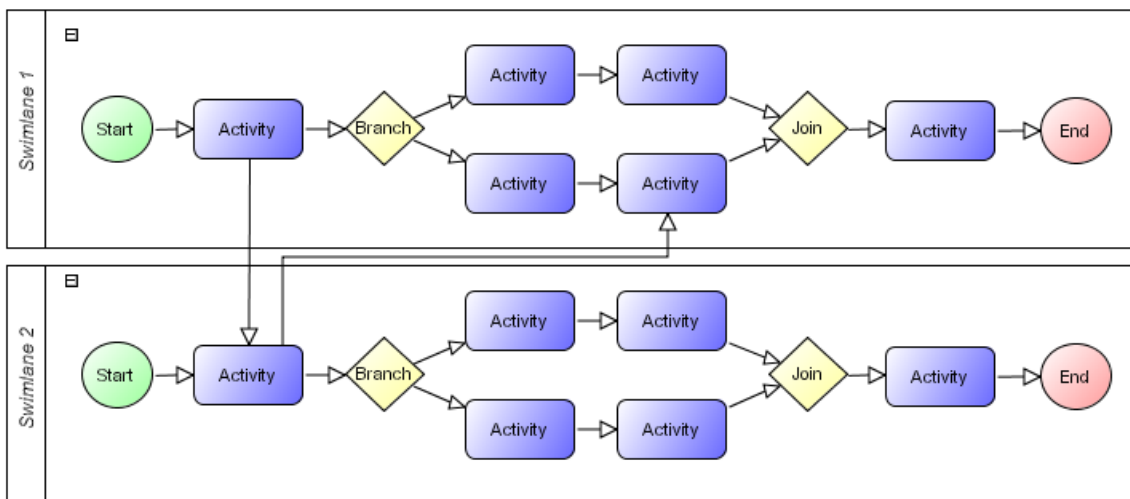


Obrázek č. 13 - Ukázka grafového výstupu dat z knihovny ZingChart [27]

## 2.14 Mxgraph

Rozsáhlá knihovna, se kterou se můžeme také setkat pod názvem JGraph sloužící pro tvorbu interaktivních diagramů a plně běží na straně klienta, čímž je kladen důraz na rychlé zpracování a renderování prvků. Technologie vykreslování používá Canvas, HTML pro zobrazení jednotlivých prvků, umí načítat a uchovávat diagramy. Pro výměnu dat obsahuje podporu značkovacího jazyku XML (Extended Markup Language). Podporu nachází u všech moderních webových prohlížečů včetně mobilních systémů, zpětná kompatibilita starších verzí je také garantována bez menších obtíží implementace. V případě žádného připojení je možné pracovat na straně klienta v offline módu, kde není potřeba backend nebo web server.

Tato knihovna je používána několika světovými společnostmi, protože je zaručena technická podpora. Mxgraph zahrnuje zcela funkční online GraphEditor, který slouží pro tvorbu přednastavených diagramů, kde se vývojář může zaměřit na detaily aplikací od samého počátku. Tento software je pod licencí Open Source a free software. Praktickou ukázkou editoru je možné nalézt v odkazu literatury. [25] [26]



Obrázek č. 14 – Ukázka z modelování digramů MxGraph [35]

## 2.15 Srovnání vybraných knihoven

Legenda jednotlivých popisků:

1. Rok vydání – první vydání knihovny
2. Poslední aktualizace – datum poslední stabilní aktualizace
3. Technologie vykreslování – způsob vykreslování obsahu na pozadí
4. Obsah grafů – počet typů, které jsou obsaženy ve struktuře knihovny
5. Vstupní formát dat – typ formátu načítání vstupních dat
6. Licence – typ vázané licence
7. Otevřený kód – možnost úpravy kódu a šíření

Tabulka č. 1 - srovnání vybraných knihoven

	Rok vydání	Poslední aktualizace	Technologie vykreslování	Obsah grafů	Vstupní formát dat	Licence	Otevřený kód
D3	2011	2017	SVG	8+	JSON CSV	BSD	pod BSD
Google Charts	2008	2016	SVG/VML	18	JSON JS API	Svobodný software	Open Source
ChartJS	2013	2017	Canvas	6	JSON CSV	MIT	pod MIT
amCharts	2006	2017	SVG	13	JSON CSV	Svobodný software	Open Source
N3-Charts	2013	2016	SVG/Canvas	3	JSON/CSV	Svobodný software	pod MIT
Flot	2007	2014	Canvas	8	JSON	MIT	pod MIT
ZingChart	2009	2017	SVG Canvas/VML	20+	JSON JS Array	Komerční užití	NE
Mxgraph	2005	2016	SVG	žádné	JS API	Svobodný software	Open Source

Všechny vybrané knihovny jsou ukázkou konkrétních implementačních řešení do moderních prohlížečů při rekonstrukci dat do vizuální podoby, jak grafy, dashboardy ale i mapy mohou být součástí. Knihovny se od sebe liší, jakým způsobem vykreslují obsahy stránek z nabízených technologií SVG nebo Canvas. Ve starších prohlížečích se můžeme setkat s formátem VML, který se již příliš nepoužívá, protože nárokuje na zpětnou kompatibilitu, která není příliš u moderních knihoven zaručena.

Pro vykreslování grafů je nejvhodnější technologie Canvas, protože graf je vykreslen jako celek a je vhodný pro prezentaci dat i v reálném čase. Grafická jednoduchost grafů umožní využít lepší výkonový potenciál. Pro větší rozlišení se hodí formát SVG, kde je lepší nasazení u map bez ztráty kvality při přibližování.

Podstatný vliv na funkcionalitu má cílová licence knihovny, kdy její zakoupení přidá větší rozsah funkcí. Komerční licencování přináší široké řešení zpracování dat, vizualizací včetně složitějších implementací přesných požadavků. Bezplatné zpracování přináší mnohé výhody, jednoduchá struktura, úpravy dle vlastní potřeby. Většina komerčních knihoven má uzavřený kód, čímž není možné přizpůsobovat detailní požadavky na míru.

Vstupní požadavek dat je podmíněn formáty JSON, CSV nebo v prosté tabulkové uspořádání jak je známo z Microsoft Excel. Mimo zmíněné formáty je možné přistupovat za pomoci speciálního JS API, které představuje jednotné zpracování pro všechny grafické komponenty. Již u většiny jmenovaných knihoven je přístup k rozhraní ze serveru ve formě URL (Uniform Resource Locator) odkazu. [33]

## **II. PRAKTICKÁ ČÁST**

### 3 HLAVNÍ POŽADAVKY

#### Implementace do prostředí

1. Knihovna musí fungovat ve všech soudobých prohlížečích a na všech platformách
2. Knihovna musí být schopná fungovat v prohlížečích na mobilních zařízeních Apple a Android
3. Knihovna nesmí vyžadovat instalaci externího softwaru do prohlížeče, jako např. Flash či Java Applet

#### Vstupní požadavky

4. Knihovna musí umožňovat načtení vzorku naměřených signálů z formátu JSON nebo CSV (hodnoty odděleny středníkem)
5. Knihovna musí podporovat obsluhu uživatelských akcí vlastními událostmi (pro klient-ská rozšíření)

#### Základní funkce

6. Graf musí být možné zobrazit v libovolném měřítku na obou osách
7. Graf musí podporovat interaktivní výběr regionu pro zvětšení (zoom)
8. Grafu musí mít „tooltip“ zobrazující aktuální měřenou hodnotu
9. Graf musí být možné lokalizovat do více jazyků
10. Graf musí mít možnost hledání hodnot, popř. rozsahu od do
11. Graf musí reagovat rychle a pružně i při načtení většího objemu dat
12. Graf musí mít možnost zobrazení mřížky
13. Graf musí být možné umístit do kontejnerů různé velikosti
14. Graf musí mít možnost konfiguračně změnit základní nastavení, zejména:
  - a. Barvy signálu
  - b. Popisy os
  - c. Velikost mřížky

#### Pokročilé funkce

15. Graf musí umožňovat export do CSV pouze viditelných data
16. Graf musí být možné ovládat kolečkem na myši (zoom)



## 4 VÝSTUPNÍ ANALÝZA VYBRANÝCH KNIHOVEN

Výstupní analýza popisuje veškeré funkční i nefunkční požadavky, které byly splněny, otestovány i případně nesplněné funkce. Požadavky jsou rozděleny od kompatibility v prostředí, základní funkcionality a pokročilé funkce. Knihovny byly otestovány na dodaném vzorku z průmyslové praxe v textovém formátu, který byl potřeba upravit do vhodného formátu, dle vstupních dat knihovny.

### 4.1 D3

1. Požadavek: prostředí, které knihovna podporuje, všechny aktuální platformy a webové prohlížeče
2. Požadavek: podporuje mobilní zařízení platformy Android a Apple
3. Požadavek: knihovna nevyžaduje instalaci externího software
4. Požadavek: načítání vstupních dat formátu JSON a CSV
5. Požadavek: podporuje vlastní uživatelské události
6. Požadavek: graf lze zobrazit v libovolných měřítkách
7. Požadavek: podporuje interaktivní výběr oblasti pro zoom
8. Požadavek: funkce zobrazující tooltip aktuální měřené hodnoty zobrazuje
9. Požadavek: lokalizace je možná do více jazyků
10. Požadavek: zobrazení hodnot pomocí filtru vyžaduje vlastní implementaci
11. Požadavek: graf je pružný a rychlý při větším objemu dat
12. Požadavek: lze zobrazit i vypnout mřížku
13. Požadavek: podporuje základní uchopení grafu funkcí „drag“, pro umístění do kontejnerů vyžaduje instalaci žádané funkce a implementaci
14. Požadavek: změna barvy signálu, názvy os a velikost mřížky je možné uživatelsky měnit
15. Požadavek: vyžaduje vlastní implementaci funkce pro export do CSV
16. Požadavek: graf lze přiblížit pomocí kolečkem na myši

Velice komplexní knihovna, z jejíž jádra vychází mnoho dalších. Tím, že obsahuje velké množství konfigurací, nemá v základní nabídce žádný graf ani nenabízí přednastavené funkce s atributy pro snadnější sestavení aplikace. Knihovna vyžaduje vytvoření vlastních funkcí, čímž jsou požadovány větší znalosti v programování a práce s grafickým obsahem stylů. Díky komplexnosti a otevřenosti knihovny je možné vytvořit vlastní řešení funkcí na míru do posledního detailu. D3 zcela podporuje soudobé prohlížeče včetně mobilních zařízení, na starší verze není brán zřetel se zpětnou kompatibilitou. Instalace externího nebo podpůrného softwaru není vyžadována.

Podpora uživatelských akcí vlastními událostmi pro rozšíření je k dispozici. Rozšíření je možné pomocí instalací dodatečných funkcí nebo zcela vlastní implementací žádané funkcionality. Zobrazování funkce tooltip aktuální měřené hodnoty je k dispozici bez omezení zobrazení. Lokalizace je k dispozici do více jazyků. Hledání hodnot v zadaném rozsahu vyžaduje vlastní instalaci funkce nebo vytvoření vlastní implementace řešení. Graf je schopen rychle zpracovávat informace ve větším objemu v řádech desítek milisekund, zobrazené data mohou mít zobrazenou nebo skrytou mřížku. Přesouvání objektu grafu je možné v základní verzi pomocí funkce drag, nýbrž vkládání do kontejnerů je nutné vytvořit. Změna konfigurace vlastností grafu názvu os, barva signálu a změna mřížky je možná. Ukládání dat z grafu do formátu CSV vyžaduje vlastní řešení funkce. Přibližování obsahu pomocí kolečka myši je podporováno.

Knihovna D3 není příliš vhodným kandidátem pro demonstrativní ukázkou webové aplikace, protože je příliš obecná pro splnění požadavků a vyžaduje ve větším množství vlastní cílená řešení. Díky své univerzálnosti se hodí pro komplexní řešení velkých grafických aplikací vyžadující přesné nasazení. Základní graf D3 byl do práce zahrnut jako ukázkou prototypové verze. [16]

## 4.2 N-3 Charts

1. Požadavek: prostředí, které knihovna podporuje, jsou všechny aktuální webové prohlížeče a operační systémy, minimální verze IE 8 je vyžadována
2. Požadavek: mobilní zařízení je podporováno pouze u Android
3. Požadavek: knihovna nevyžaduje instalaci externího software
4. Požadavek: načítání vstupních JSON a CSV dat je potřeba přes knihovnu AngularJS

5. Požadavek: omezené uživatelské události, vyžaduje vlastní řešení
6. Požadavek: graf lze zobrazit v různých měřítkách
7. Požadavek: interaktivní výběr oblasti pro zoom je podporován
8. Požadavek: funkce zobrazující tooltip aktuální měřené hodnoty zobrazuje
9. Požadavek: lokalizace je pouze v anglickém jazyku
10. Požadavek: zobrazení hodnot v rozsahu filtrem vyžaduje vlastní implementaci
11. Požadavek: graf je pružný i rychlý při větším objemu dat
12. Požadavek: mřížku lze zobrazit i vypnout
13. Požadavek: pro vkládání grafu do kontejnerů je nutno vytvořit alternativní funkci a následně implementovat
14. Požadavek: změna barvy signálu, názvy os, zobrazení velikosti mřížky je možné uživatelsky měnit
15. Požadavek: vyžaduje vlastní implementaci exportu viditelných dat do CSV
16. Požadavek: graf lze přiblížit kolečkem na myši

Jednoduchá knihovna vycházející z D3, která vyžaduje framework AngularJS, bez kterého nemůže fungovat. Funkčnost je zaručena na všech současných prohlížečích a operačních systémech včetně mobilních kromě systémů od Apple. Instalace vedlejšího softwaru třetích stran není vyžadována. Pro vstupní načítání dat je nutné použít AJAX, funkce, která dokáže získat ze serveru nebo zadaným odkazem data, aniž by byla potřeba opětovně načítat stránku s obsahem. Následné načtení vstupních údajů je potřeba převést do formátu JSON nebo CSV, větší objem informací je schopna rychle načíst v řádech milisekund.

Manipulace N3-Chart uživatelskými událostmi není obsažena, a proto je nutná vlastní implementace. Graf umožňuje editace názvů os, barvy signálu a změna velikosti mřížky, která může být viditelná nebo skrytá. Pro zobrazení rozsahu vybraných dat je nutné vytvořit a následně implementovat vlastní řešení. Manipulace grafu v oknech různých velikostech není k dispozici, protože knihovna nepodporuje v základní verzi responsibilitu obsahu, proto vyžaduje vytvoření žádané funkce a následné implementace do samotného kódu grafu.

Export viditelných dat do formátu JSON není k dispozici a je zapotřebí vlastní vytvoření požadované funkce. Interakce s grafem pomocí scroll tlačítka na myši pro přibližování interaktivní oblasti je plně podporována. N3-Charts vyžaduje znalost

AngularJS včetně AJAX a může být jednoduchým, ale mocným vizualizačním nástrojem statistických dat, chybějící funkce je možné jednoduše vytvořit za pomoci frameworku AngularJS. V základní verzi knihovna není schopna pokrýt všechny stěžejní požadavky a není příliš vhodný kandidát pro vytvoření webové aplikace, protože musí vycházet z AngularJS a nemá zcela zásadní požadované funkce. [17]

### 4.3 Google Charts

1. Požadavek: prostředí, které knihovna podporuje všechny aktuální webové prohlížeče kromě Safari, minimální verze IE 8 a novější je vyžadována
2. Požadavek: prohlížení je možné na mobilních zařízeních pod systémy Android a Apple
3. Požadavek: knihovna nevyžaduje instalaci externího softwaru
4. Požadavek: načítání je možné z formátu JSON a CSV
5. Požadavek: podporuje vlastní uživatelské události
6. Požadavek: nová verze knihovny dovoluje neomezené zobrazení měřítka i mřížky
7. Požadavek: podporuje interaktivní výběr oblasti pro zoom
8. Požadavek: funkce tooltip zobrazující vybranou měřenou hodnotu
9. Požadavek: podporuje lokalizaci do více jazyků
10. Požadavek: graf je schopen zobrazit hodnoty ve vybraném rozsahu (filtru)
11. Požadavek: při větším objemu dat je načítání knihovna v řádu sekund > 10. tisíc záznamů se rychlost vykreslování snižuje
12. Požadavek: mřížku lze zobrazit i vypnout
13. Požadavek: podporuje přesun do kontejnerů, ale je žádané uzavřít celý dashboard do jednoho celku
14. Požadavek: změna barvy signálu, názvy os, zobrazení velikosti mřížky je možné uživatelsky měnit
15. Požadavek: graf umí exportovat viditelné data do CSV
16. Požadavek: graf lze přibližovat pomocí kolečka na myši

Google Charts splňuje většinu požadavků a je vhodná pro vizualizaci dat. Nevýhodou může být, že knihovnu nelze použít v offline režimu, protože je hostována na serverech Google, kde má přístup k hlavním komponentám pro vykreslování. Knihovna funguje na všech platformách, včetně mobilních zařízení a vyšší verze Internet Explorer 8

je také podporována bez instalace externího softwaru. Vstupní data v podobě CSV nebo JSON je nutné dát do požadované podoby, aby bylo možné načítat data.

Uživatelská interakce řízená událostmi je plně přístupná ve velkém rozsahu. Poslední velká aktualizace Google Charts API přinesla zásadní změny, odstraňuje omezení rozlišení nebo množství zpracování dat čerpajících z URL odkazů. Lokalizace do jiných jazyků je podporována. Graf umí zobrazit určitou oblast hledání dat s kombinací interaktivního zoomu, který lze přiblížit pomocí kolečka na myši. Funkce tooltip není limitována a zobrazuje měřené hodnoty v plném zobrazení bez nutnosti přibližování. Kvůli hostování jádra knihovny vzdáleně z URL může trvat zpracování o několik setin sekund déle.

Průběhy grafu mohou být prokládány mřížkou, kterou je možné zobrazit nebo skrýt. Přesun grafu do kontejnerů vyžaduje složitější vytvoření funkce, protože dashboard se skládá z dílčích komponent a není jednotný jako u jiných knihoven, filtr rozsahu hodnot nebo exportování dat jsou v HTML odděleny. Uživatelská API umožňuje měnit barvu signálu, názvy os a velikost mřížky. Data je možné uložit ve vybraném rozsahu do formátu CSV. Události grafu je možné zpracovávat pro další komunikace s aplikací. Google Chart splňuje většinu požadavků a je vhodný pro webovou aplikaci vzhledem k širokým možnostem konfigurace API a zcela rozsáhlou dokumentací. Větší nevýhoda je nepružnost velkého objemu dat v určité době kvůli vzdálenému hostingu knihovny. Oficiální přidávání dalších funkcionalit vyžaduje kontrolu a schválení ze strany Google i když je pod licenci open source. Uživatelé nemají přístup k jádru knihovny kvůli zmíněnému hostingu. [14]

#### 4.4 ChartJS

1. Požadavek: všechny prohlížeče podporující technologii canvas, IE nepodporuje
2. Požadavek: prohlížení je možné na mobilních zařízeních běžící pod systémy Android a Apple
3. Požadavek: knihovna nevyžaduje instalaci externího software
4. Požadavek: vstupní načítání dat je umožněno přes formát JSON nebo CSV
5. Požadavek: podporuje vlastní uživatelské události
6. Požadavek: graf lze zobrazit v libovolném měřítku
7. Požadavek: umožňuje interaktivní výběr oblasti zoomem
8. Požadavek: zobrazuje tooltip aktuálně měřené hodnoty

9. Požadavek: lokalizace pouze v anglickém jazyku
10. Požadavek: vyžaduje vlastní vytvoření dané funkcionality hledání hodnot v rozsahu (filtr)
11. Požadavek: graf je pružný i rychlý při větším objemu dat v desetinách milisekund
12. Požadavek: mřížku lze vypnout a zapnout
13. Požadavek: vyžaduje instalaci pluginu pro zpřístupnění manipulace s grafem do kontejneru
14. Požadavek: změna barvy signálu, názvy os, zobrazení velikosti mřížky je možné uživatelsky měnit
15. Požadavek: export viditelných dat je možné v případě za složitější implementace vlastní funkce
16. Požadavek: vyžaduje instalaci pluginu pro zpřístupnění přibližování pomocí kolečka na myši

JSCharts není příliš vhodná knihovna pro základní zpracování dat, protože nesplňuje stěžejní požadavky. Knihovna vyžaduje instalace dalších komponent, tzv. pluginů, umožňující zpřístupnění dalších funkcí, které je dále potřeba vytvořit a vložit do těla kódu grafu. Kompatibilita je zaručena u všech aktuálních prohlížečů podporující pouze Canvas, výjimky IE, instalace externího software nepožaduje. Vykreslování grafu může být také SVG podle potřeby. Nevýhoda může být omezený výběr typu grafů, kterých je pouze šest. Grafy nalézají podporu u mobilních zařízení a jsou zcela responsivní obsahu, tudíž se zobrazují v libovolném měřítku.

Vstupní data lze načítat v podobě JSON nebo CSV, které umožňují zobrazit tooltip aktuální měřené hodnoty. Některé nástroje knihovny mají limitované nastavování parametrů a následnou kontrolu nad nimi. Vstupní rozsah dat není omezen, graf je schopen zpracovat velký objem vstupních dat v jedné chvíli. ChartJS obsahuje většinu hlavních funkcionalit, které je možné rozšířit pomocí dodatečných pluginů např. přibližování kolečkem na myši nebo umístování grafů do oken různých velikostí s kombinací jQuery. Funkce, které knihovna neobsahuje je názorné hledání hodnot v určitém rozsahu, které lze vytvořit pomocí vlastní implementace. Knihovna je projektem Open Source komunity a je stále ve vývoji, čímž má volný přístup do zdrojového kódu. Mřížka vlastní několik nastavení mezi, které patří nastavení velikosti a její zobrazení. Uživatelské změny názvu os a barva signálu je proveditelná. Zachytávání událostí z grafu je dispozici pro okolní komunikaci s prostředím. Základní verze ChartJS umí uložit jako obrázek, v případě

exportu viditelných dat do CSV nebo jiného formátu je nutné naprogramovat vlastní funkci. [21]

## 4.5 AmCharts

1. Požadavek: prostředí, které knihovna podporuje, jsou všechny aktuální webové prohlížeče a operační systémy, minimální verze IE 8 a novější je vyžadována
2. Požadavek: prohlížení je možné na mobilních, tabletech zařízení běžící na systémech Android a Apple
3. Požadavek: knihovna nevyžaduje instalaci externího softwaru
4. Požadavek: vstupní načítání dat je umožněno přes formát JSON a CSV
5. Požadavek: podporuje vlastní uživatelské události
6. Požadavek: graf lze zobrazit v různých měřítkách
7. Požadavek: podporuje interaktivní výběr regionu pomocí filtru i zoom
8. Požadavek: funkce tooltip zobrazující aktuálně měřenou hodnotu
9. Požadavek: podporuje lokalizaci do více jazyků
10. Požadavek: graf je schopen zobrazení hodnot ve vybraném rozsahu (filtru)
11. Požadavek: odezva při načítání vstupních dat je rychlá v řádech milisekund
12. Požadavek: mřížku lze zobrazit i vypnout
13. Požadavek: vkládání do kontejnerů pomocí skriptu jQuery
14. Požadavek: změna barvy signálu, názvy os, zobrazení velikosti mřížky je možné uživatelsky měnit
15. Požadavek: graf podporuje export do více formátů, export vybraných viditelných dat nelze provést, protože knihovna požaduje zakoupení licence pro vlastní rozšiřování
16. Požadavek: graf lze přibližovat pomocí kolečka na myši

Prvním vhodným kandidátem, který splňuje téměř všechny stěžejní požadavky je knihovna AmCharts v základním rozšíření. Plný přístup lze získat zakoupením licence za cílem komerčního užití pro možné rozšíření dle vlastních požadavků. Podpora je garantovaná na všechny moderní prohlížeče a mobilní zařízení. Instalace pluginů třetích stran není vyžadována a knihovna běží samostatně na webové stránce. Obsluha knihovny uživatelskými událostmi je možná. Konfigurace změn názvu os, velikosti mřížky nebo změna barvy signálu včetně hledání rozsahu hodnot je k dispozici.

AmCharts byla otestována na dodaném vzorku měřených dat a knihovna obstála pružné reakce načítání většího objemu dat v milisekundách i případné aktualizace konfigurace uživatelem ve stanovených požadavcích. Vstupní data ve formátu JSON nebo CSV je schopna načíst i případně exportovat. Export do formátu CSV je možný jen pro všechny data v grafu. Výsledná podoba grafu je liniového typu, který znázorňuje průběh změny veličiny v časovém okamžiku, měřítko se zcela přizpůsobují velikosti grafu i žádaná mřížka. Pomocí vlastního skriptu z jQuery je možné přesouvat graf do kontejnerů. Při velkém obsahu hodnot v grafu se mřížka nezobrazuje korektně. Výchozí technologie vykreslování je SVG, která je zaměřena na pružnost zobrazování obsahu prvků grafu. Případné zakoupení plné verze AmCharts lze provádět vlastní úpravy na míru nebo případně další rozšiřující implementace funkcionalit. Jediná větší nevýhoda knihovny je poměrně vysoká cena za pořízení licence. [23]

## 4.6 Flot

1. Požadavek: knihovna podporuje všechny webové prohlížeče a operační systémy, v případě nižší verze než IE 8 je nutné použít vykreslování exCanvas
2. Požadavek: prohlížení je možné na systémech Android a Apple
3. Požadavek: knihovna nevyžaduje instalaci externího software
4. Požadavek: vstup dat za pomoci funkce AJAX a následná extrakce do podoby JSON
5. Požadavek: vlastní uživatelské události jsou podporovány, rozšíření funkcí je možné pomocí pluginů
6. Požadavek: graf lze zobrazit v libovolném měřítku
7. Požadavek: podporuje interaktivní výběr oblasti zoomem
8. Požadavek: funkce tooltip zobrazující aktuální měřenou hodnotu
9. Požadavek: podporuje lokalizace do více jazyků, češtinu nezahrnuje
10. Požadavek: graf je schopen zobrazit data ve vybraném rozsahu (filtr)
11. Požadavek: odezva načítání vstupních dat je rychlá a pružná
12. Požadavek: mřížku lze zobrazit i vypnout
13. Požadavek: umožňuje vkládání do kontejnerů různých velikostí
14. Požadavek: změna barvy signálu, názvy os a velikost mřížky je možné uživatelsky měnit
15. Požadavek: přidání exportu dat do formátu CSV vyžaduje vytvoření funkce pomocí knihovny jQuery



16. Požadavek: graf lze přibližovat pomocí kolečka na myši

Jednoduchá grafová knihovna, která vyžaduje spolupráci s knihovnou jQuery. V případě, že neobsahuje základní klientské rozšíření je nutná vlastní implementace chybějících funkcionalit nebo instalaci rozšiřujících pluginů, kterých je velké množství, všechny pluginy je možné najít v odkazu na reference literatury. Podporu nachází u všech současných webových prohlížečů včetně mobilních zařízení bez problému. V případě starších verzí prohlížečů je nutné použít technologii exCanvas pro vykreslování obsahu.

Flot nevyžaduje instalaci softwaru třetích stran, čímž je nezávislý a jednoduchý. Zpracování dat je nejprve nutné provést přes asynchronní javascriptové volání AJAX, které nahraje potřebná data ze souboru nebo URL (Uniform Resource Locator), které následně je potřeba upravit do JSON podoby. Vstupní data je možné zobrazit v libovolném měřítku. Knihovna nemá problémy s větším objemem dat při vstupu ani při reálném zpracování v čase. Výsledná data při zobrazení mohou obsahovat mřížku, kde je možné měnit velikost podle potřeby nebo ji vypnout. Graf se dokáže přizpůsobit velikosti okna, do kterého je vložen pomocí frameworku jQuery.

Uživatelská konfigurace dokáže měnit vlastnosti grafu, názvy os, barvu signálu, export viditelných dat je nutný implementovat nejlépe přes knihovnu jQuery. Zoom má podporu v základním rozšíření vybraných oblastí dat nebo pomocí kolečka na myši. Flot splňuje základní uživatelské funkce, ale je nutné přidat stěžejní pluginy, aby dokázal splnit hlavní požadavky pro webovou aplikaci. [24]

## 4.7 ZingChart

1. Požadavek: knihovna podporuje všechny webové prohlížeče a operační systémy
2. Požadavek: prohlížení je možné na systémech Android a Apple
3. Požadavek: knihovna nevyžaduje instalaci externího softwaru
4. Požadavek: vstupní formát JSON a CSV
5. Požadavek: podporuje vlastní uživatelské události
6. Požadavek: graf lze zobrazit v libovolném měřítku
7. Požadavek: podporuje interaktivní výběr oblasti zoomu
8. Požadavek: funkce tooltip zobrazující aktuální měřenou hodnotu
9. Požadavek: podporuje lokalizace do více jazyků
10. Požadavek: graf je schopen zobrazit data ve vybraném rozsahu (filtr)

11. Požadavek: odezva načítání objemných vstupních dat je pomalejší
12. Požadavek: pouze vodící čáry, nelze měnit mřížku
13. Požadavek: umístění grafu do kontejneru je možné pomocí skriptu jQuery
14. Požadavek: změna barvy signálu, názvy os je možné uživatelsky měnit, velikost mřížky nelze měnit
15. Požadavek: graf nemá možnost export dat do CSV
16. Požadavek: graf nemá možnost přibližování kolečkem na myši

ZingChart dokázala splnit pouze nadpoloviční požadavky a některé stěžejní nebyla schopna pokrýt v základní verzi díky své jednoduchosti. Knihovna nabízí mnoho grafických nastavení do nejmenších detailů podle potřeby i za pomoci CSS stylů bez nutných instalací doplňků. Kompatibilita je zaručena u každého novějšího prohlížeče včetně mobilních zařízení Android a Apple. Aplikace běží samostatně a nevyžaduje externí instalaci softwaru. Vstupní formát je umožněn načítání v podobě JSON nebo CSV. Knihovna má obsáhlou API, které umožní řízení uživatelských událostí ve velkém měřítku. Vykreslení obsahu dosahuje libovolného měřítka, zobrazená data mají možnost zobrazit tooltip neboli aktuálně měřenou hodnotu. Pokud je potřeba zobrazit hodnotu je nutné hodně přiblížit oblast pomocí kolečka na myši. Hodnoty mohou být zobrazeny v určitém rozsahu, které umožňuje interaktivní zoom oblasti pro lepší náhled. Lokalizace má podporu do více jazyků včetně češtiny.

Načítání většího objemu dat knihovna reaguje pomaleji, včetně interaktivních událostí s grafem. Práce s mřížkou není možná, změna velikosti nebo jiných vlastností. Přesun celého grafu do vybraných kontejnerů lze provést za pomoci skriptu z jQuery. Uživatelská konfigurace vlastností zahrnuje pouze změnu názvu os a barvy signálu. Výstupní data nemohou být exportovány do podoby CSV, funkce zoom kolečkem na myši pro přiblížení není podporována. Pro vytvoření vlastních funkcí je nutné zakoupit platnou licenci, která je poměrně drahá, avšak za cenu přístupu ke zdrojovému kódu jádra knihovny. Rozšiřující funkce je možné vytvářet pomocí Angular a jQuery wrapperů, které mohou být následně integrovány jako součást knihovny. Knihovna je ovšem vhodným kandidátem pro vytvoření webové aplikace jako názorná ukázka jednoduché manipulace s API a vytváření dílčích funkcí. [27]

## 4.8 MxGraph

JavaScriptová knihovna vyžadující na straně serveru jazyk Java pro uvedení do chodu. Knihovna není vhodná jako demonstrační ukázka webové aplikace, protože její primární účel je tvorba digramů a interaktivních map. Tvorba grafů vyžaduje složitější implementaci a hlavně znalosti jazyku Java. Výsledné digramy mohou sloužit jako UML (Unified Modeling Language) pro business modelování. Alternativní řešení pro MxGraph je například Google Charts, která má mnohem jednodušší tvorbu a manipulaci. Tvorba digramů probíhá ve vybraném HTML kontejneru, do kterého se vkládají buňky a následné spojnice propojují jednotlivé prvky. Knihovna je kompatibilní na všech platformách včetně mobilních zařízení. Dokumentace pro tvorbu aplikace je v odkazu na literaturu obsahující reference a názorné ukázky. [35]

## 4.9 Vstupní data

Prvním krokem bylo obdržení naměřených hodnoty z tlaku, který působil na optický senzor z průmyslové praxe. Informace jsou obsaženy v textovém dokumentu a první požadavek je převést obsah do formátu JSON.

Ukázka reálných dat ze souboru je zobrazeno v obrázku č. 15. Pro konverzi do správného formátu JSON bylo potřeba vytvořit skript, který převede textový soubor do požadovaného výstupu. Skript je vytvořen v prostředí PHP (Hypertext Preprocessor). Následující obrázek č. 16 zobrazuje zdrojový kód skriptu.

```
1706043693.219217: DAC value: 3000, ADC value: 406 (0.4956V)
1706043693.231481: DAC value: 3000, ADC value: 413 (0.5042V)
1706043693.242824: DAC value: 3000, ADC value: 409 (0.4993V)
1706043693.253953: DAC value: 3000, ADC value: 413 (0.5042V)
1706043693.265034: DAC value: 3000, ADC value: 406 (0.4956V)
```

Obrázek č. 15 - Obdržený vzorek dat v textovém formátu

```

$lines = file('data.txt'); //uložení počtu řádků do proměnné
$outputArray = []; //výstupní pole
//Cyklus pro procházení řádků
foreach ($lines as $line_num => $line) {

    $values = explode(" ",$line); //rozdělení řádku, mezi mezerami

    //naformátování textového dokumentu do požadované podoby
    $formattedArray =
    [
        "time" => $values[0], //časová známka
        "dac" => $values[3], //stejnoseměrný proud
        "adc" => $values[6], //střídavý proud
        "voltage" => substr($values[7],1,6) //funkce substr ořezá závorky
    ];

    //předání výsledného formátu do výstupního pole
    array_push($outputArray,$formattedArray);
}

```

Obrázek č. 16 - Skript pro převod do JSON formátu

Výsledná podoba na obrázku č. 17 formátu JSON, jednotlivé data jsou v uvozovkách oddělené dvojtečkou, jednotlivé řádky obsažené v poli jsou odděleny čárkou. Načítání dat je prováděno z externího souboru s daty do dvourozměrného pole, ze kterého jsou načítány zvolené hodnoty. Měřené hodnoty DAC, ADC a volty jsou rozděleny do jednorozměrného pole se, kterým probíhá manipulace.

```

{"time":"1706043693.231481:", "dac":"3000,", "adc":"413", "voltage":"0.5042"},
{"time":"1706043693.242824:", "dac":"3000,", "adc":"409", "voltage":"0.4993"},
{"time":"1706043693.253953:", "dac":"3000,", "adc":"413", "voltage":"0.5042"},
{"time":"1706043693.265034:", "dac":"3000,", "adc":"406", "voltage":"0.4956"},

```

Obrázek č. 17 – validní JSON formát

Každá knihovna má jiné požadavky na vstupní podobu dat, proto bylo nutné provést další úpravy, které jsou vyobrazeny na obrázku č. 18 a 19. Knihovna AmChart nevyžaduje nutnou úpravu.

```

//roztřídění měřených veličin ze souboru dat
for(var i = 0; i < chartData01.length; i++)
{
    adc [i] = chartData01[i]["adc"];
    dac [i] = chartData01[i]["dac"];
    voltage [i] = chartData01[i]["voltage"];
}

```

Obrázek č. 18 – ukázka cyklu pro roztřídění měřených informací do polí

Dalším postupem bylo jednotlivé vkládání hodnot do grafu způsobem řádek a sloupec pomocí cyklu. Názorná ukázka na obrázku č. 19 vkládání údajů do knihovny Google Charts, u této knihovny bylo zvolen menší počet cyklů než je maximální velikost kvůli rychlejší odezvě grafu. ZingChart si žádá způsob vstupních dat z předchozího obrázku polí.

```
var data = new google.visualization.DataTable();
data.addColumn('number', 'Průběh');
data.addColumn('number', 'Volty');

for (var i = 0; i < 5000; i++)
{
    data.addRow([
        [i, voltage[i]]
    ])
}
```

Obrázek č. 19 – ukázka sestavení dat v Google Charts

## 4.10 Ukládání konfigurace grafu

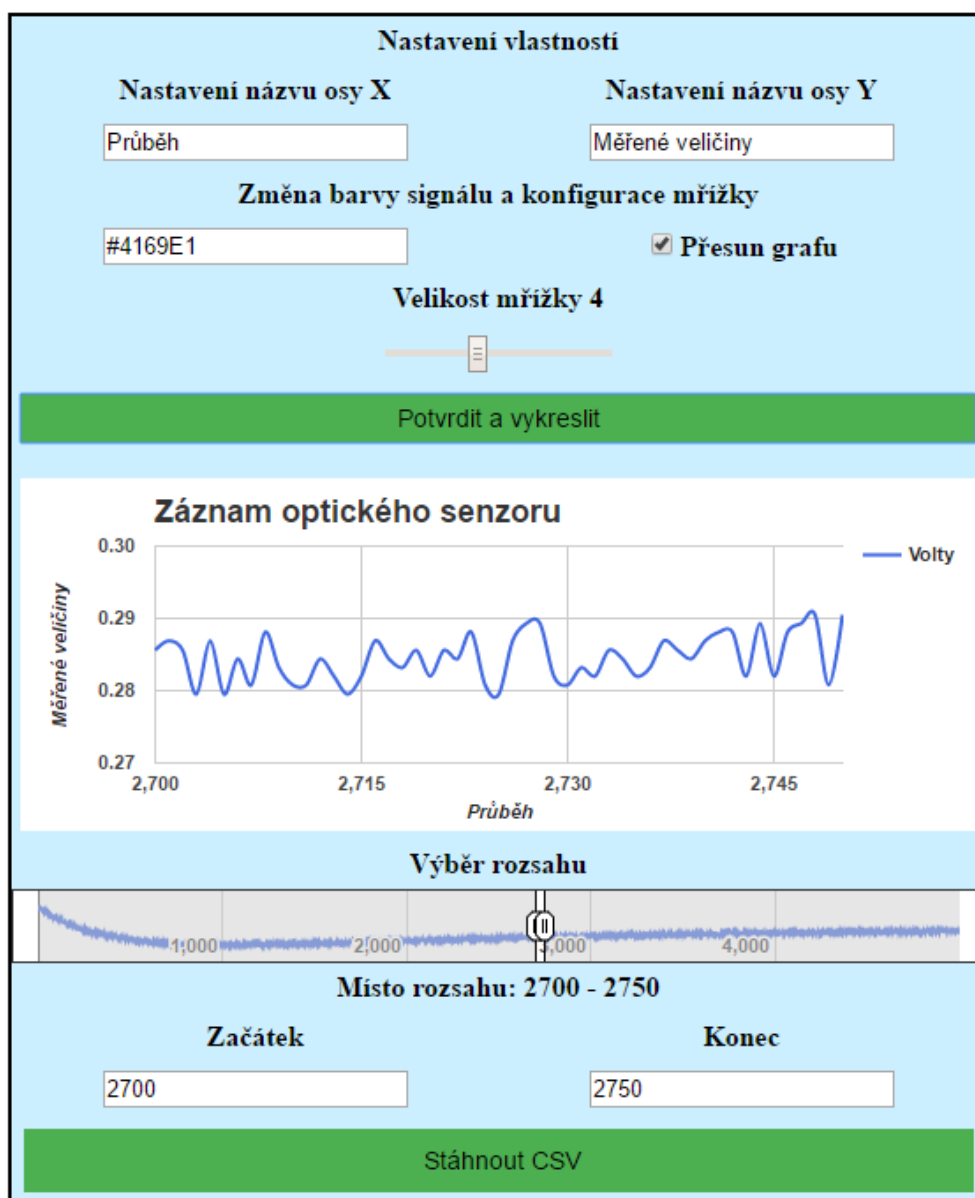
U webových stránek obecně platí, že pokud uživatel opustí záložku, tak jsou dočasná data smazány. Jsou dva způsoby jak ukládat parametry, aby prohlížeč pamatoval a to v podobě cookies nebo session, česky se můžeme setkat s výrazem relace. Protože Google Chrome nepovoluje práci s cookies, tak je použít session. Důvod proč používat session z důvodu větší bezpečnosti a je k dispozici pro ukládání většího objemu dat. Obecná ukázka místního ukládání konfigurace grafů do relace je vyobrazeno na obrázku č. 20. Na začátku stránky zavoláme `reloadoptions()`; a tím je provedeno uložení proměnných.

```
//funkce pro počáteční uživatelskou konfiguraci grafu
function saveoptions()
{
    var value = [axisX,axisY,colorsignal,gridsize];
    localStorage.setItem("options", JSON.stringify(value));
}
//funkce pro uložení jednotlivých uživatelských nastavení
function reloadoptions()
{
    var save = JSON.parse(localStorage.getItem("options"));
    axisX = save[0];
    axisY = save[1];
    colorsignal = save[2];
    gridsize = save[3];
}
```

Obrázek č. 20 – funkce pro ukládání do relace

## 5 VÝSTUPNÍ APLIKACE

Poslední část práce prezentuje ukázkou vybraných knihoven AmCharts, ZingChart, Google Charts a D3 v aplikační podobě, která obsahuje rozhraní dle zadaných požadavků. Základní layout rozhraní pro všechny knihovny je téměř shodný podle požadavků funkcionalit. Vzhled aplikace je vytvořen nástroji CSS stylů. Ukázka API je znázorněna na obrázku č. 21 z prostředí knihovny Google Charts.

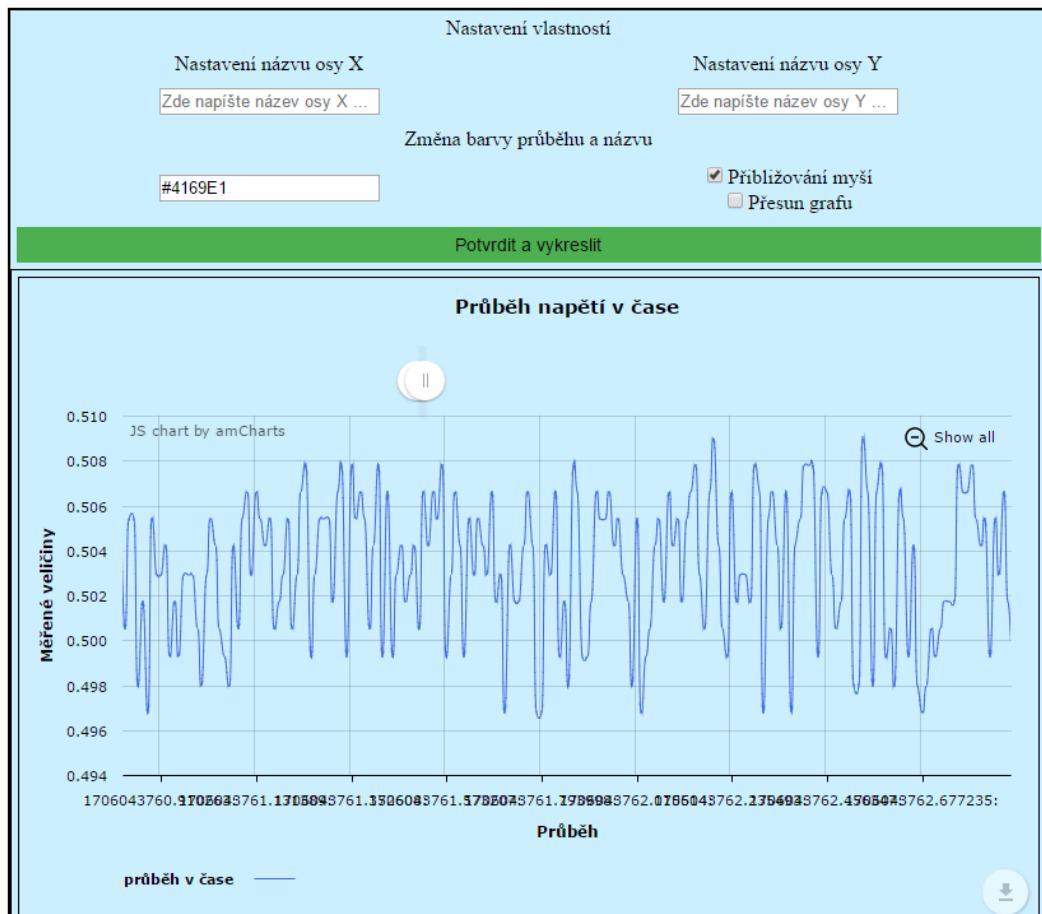


Obrázek č. 21 - ukázka rozhraní knihovny Google Charts

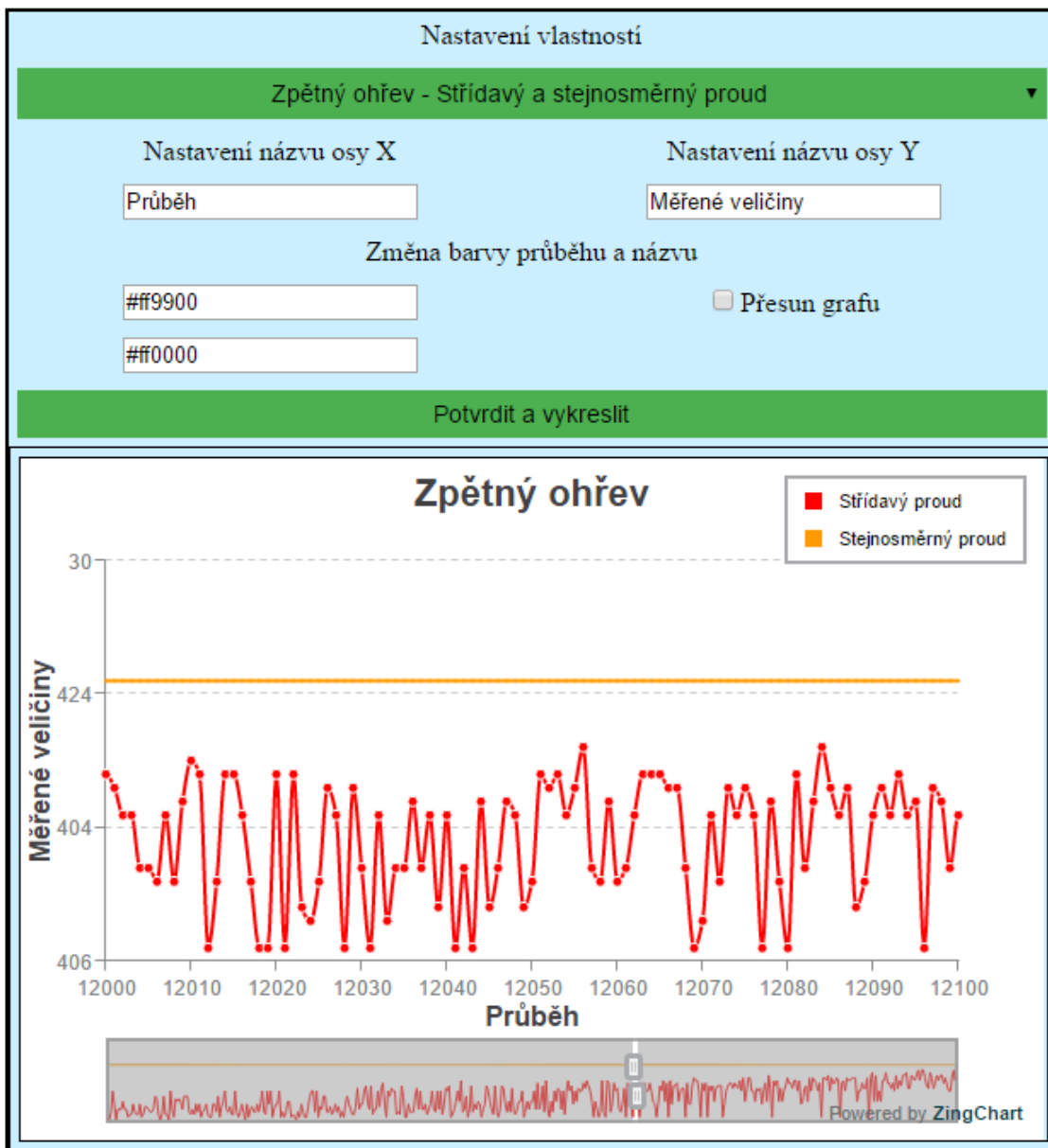
## 5.1 Popis konfigurace:

Webové aplikace obsahují obsluhu uživatelských událostí, aby bylo možné upravovat konfiguraci grafu. Níže je možné shlédnout obecný výčet funkcí pro všechny nastavení grafů. Všechny textové vstupy jsou ošetřeny proti chybám, pokud nejsou pole vyplněna, tak se zapíše prvotní konfigurace jako při prvním spuštění stránky.

1. Nastavení názvu osy X
2. Nastavení názvu osy Y
3. Změna barvy – volba pro vykreslený signál
4. Volba přiblížení – zapnutí a vypnutí přibližování pomocí kolečka na myši
5. Velikost mřížky – změna velikosti zobrazené mřížky
6. Potvrdit a vykreslit – tlačítko pro provedení zadané konfigurace a opětovné vykreslení obsahu grafu
7. Stáhnout CSV – volba pro stažení dat z grafu do formátu CSV
8. Volba graf – přepínání grafů pro jiné zobrazení dat
9. Zoom – volba pro přibližování
10. Přesun grafu - nutné vypnout, pokud je potřeba interakce s vnitřní API grafu



Obrázek č. 22 – Ukázka rozhraní knihovny AmCharts



Obrázek č. 23 – Ukázka rozhraní knihovny ZingChart

## 5.2 Plugin pro změnu barvy signálu

Všechny prohlížeče nemají implementaci pro vstup (input), HTML barvy, proto bylo nutné nahradit volenou možnost jQuery pluginem. Plugin se nazývá color pick (výběr barvy), který obsahuje základní barvy konfigurace. V hlavní stránce grafu jsou odkazy pro soubory pro plugin a také css styly. Spouštěcí funkce pro paletu barev je znázorněna na obrázku č. 24.



```
<!-- změna barvy signálu-->
<script src="colorpick/js/bootstrap-colorpicker.min.js"></script>
<script src="colorpick/js/bootstrap-colorpicker-plus.js"></script>
<script type="text/javascript">
  $(function(){
    var signal = $('#colorsignal');
    signal.colorpickerplus();
    signal.on('changeColor', function(e,color){
      if(color==null)
        $(this).val('transparent').css;
      else
        $(this).val(color).css;
    });
  });
});
```

Obrázek č. 24 – Skript pro spuštění palety barev tzv. color pick [18]

### 5.3 Rychlost odezvy knihoven

Při každém spuštění aplikace je provedeno měření rychlosti zpracování kódu. Měření je započato na začátku zdrojového kódu a ukončeno na konci. Bylo provedeno opakované měření rychlosti, celkem 10x a následné zprůměrování času odezvy. Tabulka č. 2 znázorňuje měřené rychlosti v milisekundách.

Tabulka č. 2 - rychlost odezvy knihoven

D3	46 ms
AmCharts	206 ms
GoogleCharts	328,9 ms
ZingChart (přepínání)	790,7 ms
ZingChart	1451,1 ms

## 6 SROVNÁNÍ WEBOVÝCH APLIKACÍ

Poslední kapitola je zacílena na vybrané oblasti splněných požadavků aplikací. Popisuje nejvíce vhodné knihovny pro zpracování dat a jednoduchosti.

### 6.1 AmCharts

Nejvhodnější knihovna pro velkém zpracování je AmCharts, která dokázala splnit 95% požadavků. Obsahuje další možnosti, které nejsou v požadavcích a mohli by být vhodně užity. Knihovna umí nad rámec ukládat sama sebe do grafických formátů JPEG a PNG. Exportování pouze všech dat z grafu ovšem do formátu JSON, CSV a XLSX. Rychlost zpracování dat je na druhém prvním místě a je vhodná pro jiné zpracování oblasti velkého objemu dat.

### 6.2 Google Charts

Druhé místo obsazuje knihovna od firmy Google, která svou jednoduchostí a velkých možností dokázala splnit také 95% požadavků. Byly splněny všechny požadavky, ovšem při velkém objemu dat nedokáže aplikace rychle reagovat a má zpoždění při vykreslení. Z druhého pohledu aplikace umí exportovat formát CSV pouze vybraných dat, jak bylo požadováno.

### 6.3 ZingChart

Poslední místo obsadila ZingChart s možností přepínání dvou grafů pro zobrazení více druhů dat a pro svou jednoduchost. Požadavky naplňuje na 80%. Knihovna postrádá ukládání do formátu CSV, dokáže uložit výstup v podobě grafických formátů PDF, SVG nebo je možnost tisku. Načítání a přepínání mezi grafy není pružné jak u knihovny Google nebo AmCharts.

### 6.4 Prototyp D3

Dodatečná knihovna D3, která má velice komplexní rozsahu je ve velmi základní kostře pro ukázkou webové aplikace. D3 neobsahuje objekty s atributy, které by dokázaly usnadnit práci pro kompletní všech požadavků. Knihovna vyžaduje komplexní znalosti programování a manipulaci s objekty. Vykreslování komponent je nezávislé na zbytku celého grafu, čímž jsou větší možnosti dílčích úprav na míru.

## ZÁVĚR

Práce je orientovaná na webové aplikace za cílem prezentování dat v grafové podobě za použití technologie HTML5 a JavaScript. Cílem práce bylo prostudování nejvíce používaných grafů při grafické prezentaci statistických dat. Pro vizualizaci byly evaluovány nejvíce užívané grafy pro zobrazování informačních dat. Součástí vizualizace dat bylo vybrat a popsat vybrané JavaScriptové knihovny pro vykreslování grafů. Některé knihovny mohou být dále rozšířeny frameworky, které přidávají další možnosti užití. Následně byly vlastnosti knihoven porovnány pro sestavení aplikací v praktické části, kde jsou vyzdvíženy jejich silné a slabé stránky, které sloužily pro rozhodnutí vytvoření vizuálních webových aplikací.

Druhým krokem bylo provedení podrobné analýzy vybraných knihoven za účelem splnit, co nejvíc stěžejních požadavků. Testování probíhalo na základě funkčních a nefunkčních požadavků, které měly být splněny tak, aby byla možná demonstrace grafů s pokrytím maximální funkcionality, které knihovny umožní ve svých možnostech. Vstupní data byly použity z průmyslové praxe, následně upraveny pomocí převodního skriptu do vhodné podoby, aby bylo možné, co nejlépe prezentovat data v liniových grafech. Každá knihovna vyžadovala jinou vstupní podobu dat, proto bylo nutné dále data cíleně rozřadit tak, aby byly vhodné pro čtení.

Poslední část práce tvoří praktická ukázka webových aplikací, které reprezentují vložené data z vybraných měření optického senzoru. Výstupní aplikace jsou porovnány z pohledu funkcionalit a všech konfiguračních vstupů. Aplikace má posílené nefunkční požadavky, které ošetřují uživatelské vstupy, Implementace aplikace je navržena tak, aby pokryla možnosti vstupních požadavků na knihovnu, které nejsou omezeny licencí. Vybrané knihovny byly zvoleny za účelem, aby dokázaly fungovat na všech soudobých platformách a zařízeních. Struktura kódu byla navrhována s ohledem na jednoduchou čitelnost, a aby bylo možné provádět další rozšíření za konkrétním užitím v reálném nasazení.

## SEZNAM POUŽITÉ LITERATURY

- [1] ČERNÝ, Michal. *Grafická prezentace informací ve znalostním managementu* [online]. 2011 [cit. 2017-05-09]. Dostupné z: <https://ikaros.cz/graficka-prezentace-informaci-ve-znalostnim-managementu>
- [2] Bar Charts. *Universite of Leicester* [online]. Velká Británie [cit. 2017-05-09]. Dostupné z: <http://www2.le.ac.uk/offices/ld/resources/numerical-data/bar-charts>
- [3] ČÍHAŘ, Jiří. *Excel - Grafy - Kombinace sloupcového a spojnicového grafu* [online]. [cit. 2017-05-09]. Dostupné z: <http://www.dataspectrum.cz/pages/glossary/wizzards/chart3.htm>
- [4] Chart Gallery. *Google Charts* [online]. Mountain View, Kalifornie, USA: Google, 2011 [cit. 2017-05-09]. Dostupné z: <https://developers.google.com/chart/interactive/docs/gallery>
- [6] Digitální knihovna Západočeské univerzity v Plzni. *Digitální knihovna* [online]. Plzeň [cit. 2017-05-09]. Dostupné z: <https://otik.uk.zcu.cz/bitstream/handle/11025/12106/VetrovskaZuzanaBP.pdf?sequence=1>
- [7] MAREK, Tomáš. *Efektivní vizualizace dat se zaměřením na základní typy grafů: Effectiveness of data visualization with a focus on basic graph types* [online]. Brno, 2015 [cit. 2017-05-09]. Diplomová práce. Masarykova Univerzita. Vedoucí práce Jan Boček.
- [8] *Pie Graph* [online]. 2011 [cit. 2017-05-09]. Dostupné z: <http://whatis.techtarget.com/definition/pie-graph-or-pie-chart>
- [9] Freedman, D. Pisani, R. and Purves, R. 1998 [cit. 2017-05-09]. *Statistics* (Third edition). W.W.Norton
- [10] *Maths a Stats Support Centre: Katedra aplikované matematiky a informatiky Ekonomicko-správní fakulty Masarykovy univerzity* [online]. Masarykova Univerzita [cit. 2017-05-09]. Dostupné z: <http://mathstat.econ.muni.cz/media/12557/graf1.pdf>
- [11] *Google Charts: Using Google Charts* [online]. Mountain View, Kalifornie, USA, 2011 [cit. 2017-05-09]. Dostupné z: <https://developers.google.com/chart/interactive/docs/>
- [12] KOSEK, Jiří. *Skriptování na straně serveru a klienta: Tvorba webových stránek a aplikací* [online]. 2012 [cit. 2017-05-09]. Dostupné z: <http://www.kosek.cz/vyuka/4iz228/prednasky/skriptovani.pdf>
- [13] *AngularJS* [online]. Londýn, Velká Británie: Google, 2010 [cit. 2017-05-09]. Dostupné z: <https://angularjs.org/>
- [14] SUEHRING, Steve. *JavaScript: krok za krokem*. Brno: Computer Press, 2008. Krok za krokem (Computer Press). ISBN 978-80-251-2241-9.
- [15] FLANAGAN, David. *JavaScript: kompletní průvodce*. 2. aktualiz. vyd. Praha: Computer Press, 2002. Všechny cesty k informacím. ISBN 80-722-6626-8.

- [16] BOSTOCK, Mike. *D3: Data Driven Documents* [online]. 2011 [cit. 2017-05-09]. Dostupné z: <https://d3js.org/>
- [17] *N3-Charts* [online]. 2013 [cit. 2017-05-09]. Dostupné z: <https://github.com/n3-charts>
- [18] Bootstrap Colorpicker Plus - jQuery Plugin: Bootstrap plugins. *JQuery Rain* [online]. 2016 [cit. 2017-05-09]. Dostupné z: <http://www.jqueryrain.com/2016/02/bootstrap-colorpicker-plus-jquery-plugin/>
- [19] MURRAY, Scott. *Making a bar chart: D3 tutorials* [online]. 2015 [cit. 2017-05-09]. Dostupné z: <http://alignedleft.com/tutorials/d3/making-a-bar-chart>
- [20] VIERA, Sara. *Java Script: Easily Create Stunning Animated Charts with Chart.js* [online]. 2013 [cit. 2017-05-09]. Dostupné z: <http://www.webdesignerdepot.com/2013/11/easily-create-stunning-animated-charts-with-chart-js/>
- [21] *Chart.js: API Documentation* [online]. 2013 [cit. 2017-05-10]. Dostupné z: <http://www.chartjs.org/docs/#line-chart-introduction>
- [22] ČÁPKA, David. *Java Script: Základy jQuery* [online]. 2013, 14 [cit. 2017-05-10]. Dostupné z: <http://www.itnetwork.cz/javascript/jquery-zaklady>
- [23] ČÁPKA, David. *AmCharts: Java Scripts Charts and Maps* [online]. 2013 [cit. 2017-05-10]. Dostupné z: <https://www.amcharts.com/>
- [24] LAURSEN, Ole, SCHNUR, David, ed. *Flot: Attractive JavaScript plotting for jQuery* [online]. 2009 [cit. 2017-05-10]. Dostupné z: <http://www.flotcharts.org/>
- [25] *MxGraph: JavaScript diagramming library* [online]. 2005 [cit. 2017-05-10]. Dostupné z: <https://www.jgraph.com/javascript-graph-visualization-library.html>
- [26] *MxGraph: GraphEditor* [online]. [cit. 2017-05-10]. Dostupné z: <https://jgraph.github.io/mxgraph/javascript/examples/grapheditor/www/index.html>
- [27] *ZingChart: All Your Dataviz Needs In One Library* [online]. 2009 [cit. 2017-05-10]. Dostupné z: <https://www.zingchart.com/>
- [28] GRIMMICH, Šimon. *Interval.cz: Vector Markup Language* [online]. 2004 [cit. 2017-05-10]. Dostupné z: <https://www.interval.cz/clanky/vml-vector-markup-language/>
- [29] MALÝ, Martin. *Zdroják: SVG, nebo Canvas? Vyberte si* [online]. 2010 [cit. 2017-05-10]. Dostupné z: <https://www.zdrojak.cz/clanky/svg-nebo-canvas-vyberte-si/>
- [30] LHOTKA, Ladislav. *Zdroják: Nové standardy pro JSON* [online]. 2013 [cit. 2017-05-10]. Dostupné z: <https://www.root.cz/clanky/nove-standardy-pro-json/>
- [31] VANČO, Michal. *Podpora pro práci s CSV soubory v NetBeans IDE* [online]. Brno, 2009 [cit. 2017-05-10]. Dostupné z: [https://is.muni.cz/th/207479/fi\\_b/BP.pdf](https://is.muni.cz/th/207479/fi_b/BP.pdf). Bakalářská práce. Masarykova Univerzita. Vedoucí práce Petr Adámek.
- [32] ANTONNIETA PERNA, Marie. *HTML and CSS: Canvas vs. SVG: Choosing the Right Tool for the Job* [online]. 2016 [cit. 2017-05-10]. Dostupné z: <https://www.sitepoint.com/canvas-vs-svg-choosing-the-right-tool-for-the-job/>

- [33] STOKLÁSKA, Kamil. *Webové knihovny pro vykreslování specifických dashboardů pro technologie v dopravě* [online]. Zlín, 2016 [cit. 2017-05-10]. Diplomová práce. Univerzita Tomáše Bati. Vedoucí práce Pavel Pokorný.
- [34] *PHP: Historie a budoucnost* [online]. 2004 [cit. 2017-05-10]. Dostupné z: [http://www.linuxsoft.cz/article.php?id\\_article=171](http://www.linuxsoft.cz/article.php?id_article=171)
- [35] *MxGraph: User manual - JavaScript* [online]. 2007 [cit. 2017-05-10]. Dostupné z: <https://jgraph.github.io/mxgraph/docs/manual.html>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

API	Application Programming Interface
JS	JavaScript
HTML	HyperText Markup Language
DOM	Document Object Model
VML	Vector Markup Language
SVG	Scalable Vector Graphics
JSON	JavaScript Object Notation
MIT	Massachusetts Institute of Technology
XML	Extended Markup Language
PHP	Hypertext Preprocessor
IE	Internet Explorer
AJAX	Asynchronous JavaScript and XML
URL	Uniform Resource Locator
UML	Unified Modeling Language

**SEZNAM OBRÁZKŮ**

Obrázek č. 1 - Ukázka spojnicového grafu [4] .....	11
Obrázek č. 2 - Ukázka horizontálního sloupcového grafu [4] .....	12
Obrázek č. 3 - Ukázka výsečového grafu [4] .....	13
Obrázek č. 4 - Znázornění histogramu [4] .....	14
Obrázek č. 5 – Ukázka zpracování PHP .....	17
Obrázek č. 6 – Porovnání technologií Canvas a SVG [32] .....	20
Obrázek č. 7 – vizualizace za pomoci D3 s ukázkou sloupcového grafu [19] .....	21
Obrázek č. 8 - kód pro liniový graf z N3-Charts [17] .....	22
Obrázek č. 9 - Ukázka kódu pro vykreslení grafu v Google Charts [11] .....	23
Obrázek č. 10 - Vykreslení liniového grafu [21] .....	24
Obrázek č. 11 - Liniový graf z knihovny amCharts [23] .....	25
Obrázek č. 12 - Liniový graf knihovny Flot [24] .....	26
Obrázek č. 13 - Ukázka grafového výstupu dat z knihovny ZingChart [27] .....	27
Obrázek č. 14 – Ukázka z modelování digramů MxGraph [35] .....	28
Obrázek č. 15 - Obdržený vzorek dat v textovém formátu .....	43
Obrázek č. 16 - Skript pro převod do JSON formátu .....	44
Obrázek č. 17 – validní JSON formát .....	44
Obrázek č. 18 – ukázka cyklu pro roztřídění měřených informací do polí .....	44
Obrázek č. 19 – ukázka sestavení dat v Google Charts .....	45
Obrázek č. 20 – funkce pro ukládání do relace .....	45
Obrázek č. 21 - ukázka rozhraní knihovny Google Charts .....	46
Obrázek č. 22 – Ukázka rozhraní knihovny AmCharts .....	47
Obrázek č. 23 – Ukázka rozhraní knihovny ZingChart .....	48
Obrázek č. 24 – Skript pro spuštění palety barev tzv. color pick [18] .....	49



## SEZNAM TABULEK

Tabulka č. 1 - srovnání vybraných knihoven.....	29
Tabulka č. 2 - rychlost odezvy knihoven.....	49

## SEZNAM PŘÍLOH

1. Univerzální HTML kód webové aplikace.
2. Elektronická verze bakalářské práce a zdrojové kódy aplikací.

## PŘÍLOHA P I: UNIVERZÁLNÍ HTML KÓDY APLIKACE

```
<table>
  <tr>
    <td colspan="3">Nastavení vlastností</td>
  </tr>
  <tr>
    <td>Nastavení názvu osy X</td>
    <td>Nastavení názvu osy Y</td>
  </tr>
  <tr>
    <td>
      <input type="text" id="axisX" maxlength="30" placeholder="Zde napíšete název
osy X ...">
    </td>
    <td>
      <input type="text" id="axisY" maxlength="30" placeholder="Zde napíšete název
osy Y ...">
    </td>
  </tr>
  <tr>
    <td colspan="2"> Změna barvy průběhu a názvu</td>
  </tr>
  <tr>
    <td> <input type="text" id="colorsignal" value="#4169E1" maxlength="16">
      <td> <input type="checkbox" onClick="EnableDrag(this.value);" checked>Přesun
grafu</td>
  </tr>
  <tr>
    <td colspan="2"><input onclick="chartoptions()" type="button"
class="button" value="Potvrdit a vykreslit"></td></tr>
  <tr>
    <td colspan="2" id="container" class="container">
      <div id="chartdiv"></div>
    </td>
  </tr>
  <tr>
    <td colspan="2" id="container" class="container"> </td>
  </tr>
  <tr>
    <td colspan="2" id="container" class="container"> </td>
  </tr>
</table>
```

## **PŘÍLOHA P II: ELEKTRONICKÁ VERZE**

Zdrojové kódy aplikací a publikace bakalářské práce v PDF

Aplikace:

AmCharts/index.html

GoogleCharts/index.html

ZingChart/index.html

D3/index.html

Text:

fulltext.pdf