

**ČESKÉ VYSOKÉ  
UČENÍ TECHNICKÉ  
V PRAZE**

**FAKULTA  
STROJNÍ**



**BAKALÁŘSKÁ  
PRÁCE**

**2018**

**ONDŘEJ  
BAUMRT**

**České vysoké učení technické v Praze**  
**Fakulta strojní**

**Ústav přístrojové a řídicí techniky**  
**Obor: Informační a automatizační technika**

**Rozšíření webové aplikace**  
**EasyMeasurement o pokročilé uživatelské**  
**funkce**

**BAKALÁŘSKÁ PRÁCE**

Vypracoval: Ondřej Baumrt  
Vedoucí práce: Ing. Matouš Cejnek  
Rok: 2018

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Baumrt** Jméno: **Ondřej** Osobní číslo: **438273**  
Fakulta/ústav: **Fakulta strojní**  
Zadávající katedra/ústav: **Ústav přístrojové a řídicí techniky**  
Studijní program: **Strojirenství**  
Studijní obor: **Informační a automatizační technika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Rozšíření webové aplikace EasyMeasurement o pokročilé uživatelské funkce**

Název bakalářské práce anglicky:

**EasyMeasurement web application extension with advanced user functions**

Pokyny pro vypracování:

Účelem práce je rozšířit nástroj pro záznam hodnot naměřených při výuce v laboratořích. Předmětem této práce jsou rozšíření:

1. zobrazení grafů z uložených hodnot přes webové rozhraní
2. vytvoření webového formuláře pro hromadné zadání hodnot
3. rozšíření API o možnost přijímat více hodnot z jednoho požadavku

Seznam doporučené literatury:

Pilgrim, Mark - Ponořme se do Python(u) 3 (ISBN 978-80-904248-2-1)  
Bendoraitis, Aidas - Web Development with Django Cookbook (ISBN 978-1-78588-677-5)

Jméno a pracoviště vedoucí(ho) bakalářské práce:


**Ing. Matouš Cejnek, U12110.3**

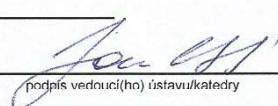
Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

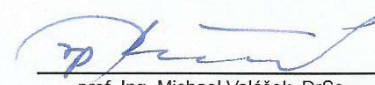
Datum zadání bakalářské práce: **18.04.2018**

Termín odevzdání bakalářské práce: **15.06.2018**

Platnost zadání bakalářské práce: \_\_\_\_\_

  
Ing. Matouš Cejnek  
podpis vedoucí(ho) práce

  
podpis vedoucí(ho) ústavu/katedry

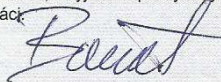
  
prof. Ing. Michael Valášek, DrSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

**26. 4. 2018**

Datum převzetí zadání



Podpis studenta

### **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně s tím, že její výsledky mohou být dále použity podle uvážení vedoucího bakalářské práce jako jejího spoluautora. Souhlasím také s případnou publikací výsledků diplomové práce nebo její podstatné části, pokud budu uveden jako její spoluautor.

V Jirkově dne 4. 6. 2018.

.....  
Ondřej Baumrt

### **Poděkování**

Děkuji Ing. Matouši Cejnkovi za připomínky a podněty při vedení této bakalářské práce a současně i rodině a přátelům za pomoc a podporu během celého studia.

Ondřej Baumrt

**Název práce:** Rozšíření webové aplikace EasyMeasurement o pokročilé uživatelské funkce

**Autor:** Ondřej Baumrt

**Obor:** Informační a automatizační technika

**Druh práce:** Bakalářská práce

**Vedoucí práce:** Ing. Matouš Cejnek  
Ústav přístrojové a řídicí techniky Fakulty strojní  
České vysoké učení technické v Praze

**Abstrakt:** Tato bakalářská práce je rozšířením stávající aplikace EasyMeasurement. Je zde popsána možnost vytvoření grafů v HTML včetně popsaní tvorby několika knihoven, které slouží tomuto účelu. Další částí je možnost uložení většího množství dat v rámci jednoho požadavku. Také je zde uvedena možnost tvorby dynamických formulářů spolu s funkcí exportu do přenositelného formátu, opět včetně popisu několika možností využití knihoven. Vše je také reálně aplikováno přímo na dané aplikaci.

**Klíčová slova:** Django, EasyMeasurement, framework, HTML, Chart.js, JavaScript, Jinja2, LaTeX, Python

**Title:** EasyMeasurement web application extension with advanced user functions

**Author:** Ondřej Baumrt

**Abstract:** This bachelor thesis extends an existing application EasyMeasurement. There are described several possibilities of creating charts in HTML including libraries which can be used to this purpose. Next part is an option for uploading bigger amount of data within one request. There is also mentioned possibility of creating dynamical forms with an export function to a portable document format with a description of exporting libraries as well. All of these points are applied on the given application.

**Keywords:** Chart.js, Django, EasyMeasurement, framework, HTML, JavaScript, Jinja2, LaTeX, Python

# Obsah

Přehled použitých zkratk	1
Úvod	3
Teoretická část	5
1 Programovací jazyk Python	5
1.1 Framework Django	6
1.1.1 Struktura projektu	6
1.1.2 Databáze	8
1.1.3 Internetové protokoly	13
1.2 Rozdíly Pythonu mezi verzí 2 a 3	15
2 Přídavné nástroje	16
2.1 Skriptovací jazyk JavaScript	16
2.1.1 Knihovna jQuery	16
2.1.2 Knihovna Bootstrap	17
2.2 Tvorba grafů	18
2.2.1 Python	19
2.2.2 JavaScript	20
2.2.3 Srovnání knihoven pro tvorbu grafů	29
2.3 Knihovny pro export	30
2.3.1 Knihovna ReportLab	30
2.3.2 Knihovna xhtml2pdf	30
2.3.3 Knihovna WeasyPrint	31
2.3.4 Knihovna Jinja2	31
Praktická část	33
3 Tvorba grafů	33
3.1 Definice funkcí pro grafy	33
3.2 Vytvoření HTML šablon	34
3.2.1 Šablony pro výběr typu grafů	34
3.2.2 Šablony pro výběr měření a veličin	34
3.2.3 Šablony pro zobrazení grafů	35
4 Formuláře	40
4.1 Šablona	40
4.2 Generování tabulky	42
4.3 Ukládání dat do databáze	43
4.4 Generování grafu	44
4.5 Exportování	44
4.5.1 Generování PDF dokumentu	45
5 Ukládání více dat	47
6 Změna vzhledu	49
6.1 EasyLaboratory	50
6.2 Přechod na Python 3	51
Závěr	53
Seznam použité literatury	55
Seznam použitých obrázků	59
Přílohy	60

## Přehled použitých zkratk

- **API** – Application Programming Interface, rozhraní pro programování aplikací, str. 16;
- **CSS** – Cascading Style Sheet (kaskádové styly), jazyk pro popis způsobu zobrazení jednotlivých elementů v HTML, str. 16;
- **CSV** – Comma-separated Value (hodnoty oddělené čárkami), souborový formát pro výměnu tabulkových dat, str. 7;
- **DRY** – Don't reapeate yourself (Neopakuj se), princip snižování počtu duplicitních informací, str. 7;
- **EPS** – Encapsulated PostScript, formát obrázků pro rastrovou, vektorovou i kombinovanou grafiku, str. 19;
- **FTP** – File Transfer Protocol, protokol pro přenos souborů mezi počítači, str. 13;
- **GUI** – Graftic User Interface (grafické uživatelské prostředí), uživatelské rozhraní, které umožňuje ovládat počítač pomocí interaktivních grafických ovládacích prvků, str. 19;
- **HTML** – HyperText Markup Language (Hypertextový značkovací jazyk), název značkovacího jazyka používaného pro tvorbu webových stránek, str. 7;
- **HTTP** – HyperText Transfer Protocol, internetový protokol určený pro výměnu dokumentů HTML, str. 13;
- **HTTPS** – HyperText Trasfer Protocol Secured, zabezpečený protokol HTTP, str. 14;
- **IDE** – Integrated Development Enviroment (Vývojové prostředí), str. 6;



- **NASA** – National Aeronautics and Space Administration (Národní úřad pro letectví a kosmonautiku), str. 6;
- **NoSQL** – databázový koncept, kde se užívají jiné prostředky než tradiční relační databáze, str. 8;
- **PDF** – Portable Document Format (přenosný formát dokumentů), souborový formát užívaný pro ukládání dokumentů nezávisle na hardwaru i softwaru, str. 4;
- **PHP** – HyperText Preprocessor (Hypertextový preprocesor), skriptovací programovací jazyk, str. 5;
- **PNG** – Portable Network Graphic (přenosná síťová grafika), grafický formát určený pro bezztrátovou kompresi rastrové grafiky, str. 19;
- **SFTP** – SSH File Transfer Protocol, protokol a program pro bezpečný přenos souborů, str. 15;
- **SQL** – Structured Query Language (Strukturovaný jazyk dotazu), dotazovací jazyk užívaný pro práci s daty v relačních databázích, str. 8;
- **SVG** – Scalable Vector Graphic (škálovatelná vektorová grafika), formát souboru, který popisuje dvourozměrnou vektorovou grafiku pomocí XML, str. 19;
- **TSV** – Tab-Separated Value (hodnoty oddělené tabulátory), formát pro ukládání a čtení málo strukturovaných dat, str. 21;
- **URL** – Uniform Resource Locator (jednotná adresa zdroje), řetězec znaků sloužící ke specifikaci umístění zdrojů informací, str. 7;
- **XML** – eXtensible Markup Language (rozšířitelný značkovací jazyk), značkovací jazyk pro jednoduché vytváření konkrétních jazyků pro různé účely a různé typy dat, str. 2.

# Úvod

Aplikace EasyMeasurement byla vytvořena v roce 2017 jako bakalářská práce pana Vojtěcha Kadlece. Jejím úkolem bylo zjednodušit zadávání a práci s daty, které bylo možné naměřit na Fakultě strojní ČVUT v Praze, ale i v reálné praxi.

Již tehdy uměla základní operace jako: vytvořit měření s danými veličinami, které měly určité hodnoty. Tato měření bylo možné upravovat a mazat, k veličinám bylo možné přidat jejich název i jednotky a hodnoty vždy obsahovaly časové razítko. Dokonce bylo možné jednotlivé veličiny i se všemi jejich hodnotami vyexportovat s příponou *.xls*, tedy formátem, se kterým úspěšně v prostředí Windows pracuje tabulkový editor Microsoft Office Excel. Samozřejmostí je také smazání daných hodnot. Další nespornou výhodou aplikace je skutečnost, že bylo možné poslat hodnoty s datovým razítkem pomocí skriptu v jakémkoli programovacím jazyce za přispění jednoduchého pořadí odesílaných hodnot. Všechny tyto informace, a ještě mnohem další lze najít právě v bakalářské práci [0-1] pana Vojtěcha Kadlece.

Tato práce má na práci pana Kadlece navázat a aplikaci EasyMeasurement dle možností zdokonalit. Primárními úkony této práce jsou:

Zobrazení naměřených hodnot pomocí grafu, kdy je možné tato data filtrovat podle data naměření. To hlavně z důvodu lepší přehlednosti při naměření většího množství dat u dané veličiny. Je také možné graficky zobrazit dvě veličiny z daného měření, a to hlavně z důvodu možnosti srovnání a například ověření správné funkčnosti snímače.

Dalším, neméně důležitým úkonem této bakalářské práce je možnost přijímání většího množství dat v rámci jednoho požadavku. To ulehčí práci nejen serveru, na kterém aplikace EasyMeasurement v budoucnosti poběží, čímž zlepší rychlost běhu aplikace, ale i danému uživateli, který může odeslat tisíc nebo i více hodnot najednou. To vše za použití jednoduchého formátu.

Posledním úkolem této práce je, jak jinak než, zjednodušení práce, v tomto případě studentovi, který vypracovává protokol měření v laboratorních pracích. Za součinnosti lektora dané laboratorní úlohy je možné vytvořit jednoduchý formulář naměřených hodnot, kam student již jen vypíše naměřená data, nechá si sestrojít graf dle svých požadavků a toto všechno si nechat vyexportovat do formátu PDF, který lze snadno přenést na vlastní zařízení a vytisknout.

V kapitolách 1 a 2 byly použity části semestrálních projektů autora z letního semestru 2016/2017 [0-2] a zimního semestru 2017/2018 [0-3].

# Teoretická část

## 1 Programovací jazyk Python

*„Python je vysokoúrovňový skriptovací programovací jazyk“* [3]. Tento jazyk, vyvinutý v roce 1991 nizozemským programátorem Guidem van Rossumem, je využíván díky své široké aplikovatelnosti, ale i své jednoduchosti, jelikož umožňuje kód sepsat do menšího počtu řádků ve srovnání například s C++ či Javou. Další výhodou je, že se jedná o tzv. open-source projekt – čili je volně šiřitelný a kupříkladu v Linuxu se objevuje již v základní instalaci. [1]

Podobně jako C++, Java či PHP se jedná o jazyk vyšší úrovně, což znamená, že programovat v nich je jednodušší (méně času k napsání) a existuje menší pravděpodobnost vytvoření chyby během psaní programu. Jsou přenosné, což znamená, že je lze spustit na různých počítačích, což je u programů psaných v jazycích nižší úrovně poměrně velký problém, neboť se programy často musí upravovat pro různá zařízení. Proto se jazyky nižší úrovně využívají jen ve specializovaných aplikacích. [2]

Problémem ovšem je, že zařízení umějí číst pouze jazyky nižší úrovně. Proto se pro realizaci aplikací v jazycích vyšší úrovně používá dvou programů: překladačů, které čte program v jazycích vyšší úrovně a v krátkých časových intervalech je střídavě čte a realizuje; a kompilátorů, který program přečte a přeloží najednou předtím, než ho spustí. Při využití kompilátorů se program v jazycích vyšší úrovně nazývá zdrojový kód a přeložený kód je strojový kód, či cílový kód. [2]

## 1.1 Framework Django

„Django je svobodný a open source webový aplikační framework napsaný v Pythonu. Webový framework je sada komponent, která pomáhá vyvíjet webové stránky rychleji a snadněji“ [3]. Django byl vytvořen za účelem pomoci autorům aplikace jejich práci co nejvíce urychlit, současně při dosažení co největší bezpečnosti, díky varováním na mnohé bezpečnostní chyby [4]. Toho je dosaženo velmi konzistentní datovou strukturou a následnou dobrou organizací [5].

Django bylo spuštěno 21. července 2005 a od té doby se stalo nejvíce používaným frameworkem psaným v Pythonu. Poslední stabilní aktualizace byla vydána 2. dubna letošního roku, jedná se o verzi 2.0.4. [6]

Pokud je definován základní pracovní postup lze vytvořit opravdu impozantní projekt [5]. Django je v současné době velmi využíván. Mezi aplikace, které jej využívají, lze zařadit sociální síť *Instagram*, internetový prohlížeč *Mozilla Firefox*, či webové stránky americké *NASA* [7].

### 1.1.1 Struktura projektu

V podstatě se jedná o knihovnu, ve které většina rozhodování závisí pouze na programátorovi aplikace. [8]

Nejvyšší úroveň v Django je projekt, který je charakterizován souborem `settings.py`, čili se jedná o modul v Pythonu a je společný pro všechny aplikace v Django. Díky jednoduchosti vytváření aplikací skriptem je postačující pro tvorbu textový editor s podporou zvýrazňování syntaxe (avšak ani ta není, pro zkušenější programátory, nutná), není tedy nutné používat IDE. [8]

Nejprve je nutné zkopírovat script `django-admin.py` do adresáře projektu. Poté je nutné vytvořit projekt příkazem `python django-admin.py startproject (název projektu)`. V tomto bodě se vytvoří adresář v pracovním adresáři s názvem projektu a bude obsahovat následující soubory: [8]

- *manage.py*, který má na starosti vlastní správu projektu čili „vytvoření základní struktury nové aplikace; vytvoření databázového schématu; spouštění testovacího webového serveru; a spouštění jednotlivých testů“ [8].
- *settings.py*, což je konfigurační skript, který je společný pro všechny aplikace v projektu. Obsahuje zejména: „konfiguraci databázového připojení; konfiguraci připojení k SMTP serveru pro snadné posílání emailů; konfiguraci tzv. Middleware; nastavení časového pásma a lokále aplikace a registraci aplikací do projektu“ [8].
- *uris.py*. což je globální konfigurace URL [8].
- *\_\_init\_\_.py*, díky čemuž je adresář projektu zároveň balíčkem Pythonu [8].
- Součástí jsou také složky *static* a *media*, které být použity mohou, ale nemusí. Do složky *static* je vhodné umístit například statické JavaScriptovské skripty či kaskádové styly, do složky *media* například nahrané CSV soubory.

Výhodou je, že konfigurační program se vygeneruje již okomentovaný, čímž velmi usnadní práci i naprostým začátečníkům. V tuto chvíli je již v provozu příkaz pro spuštění testovacího serveru a po přístupu k webu již lze vidět úvodní stránku. [8]

Důležité je, že aby aplikace mohla být vytvořena, je nutné mít založený projekt. Taktéž jedna aplikace může být součástí více projektů. [8]

Mimo výše zmíněných pythonovských programů jsou součástí každé aplikace další. Mezi ně patří například:

- *admin.py*, což je program sloužící jako administrační stránka aplikace. V podstatě je zbytečné ji upravovat, jelikož automaticky generovaná stránka je dostačující pro většinu aplikací [5];
- *forms.py*, který obsahuje HTML formuláře, které jsou používané částěji a bylo by tedy nevhodné je v rámci přístupu DRY psát opakovaně [5];

- *models.py*, kde autor může najít databázové modely čili vlastně tabulky, ze kterých se pak využívají data ve vlastní aplikaci [5].
- *urls.py* obsahuje v případě aplikace jednotlivé URL adresy, které se poté hromadně importují do globálních URL adres projektu [5].
- *views.py* je nejspíše ten nejdůležitější soubor v celé aplikaci. Obsahuje totiž předepsané funkce psané v Pythonu a určují tedy, co která stránka aplikace bude vytvářet [5].
- Poslední, a neméně důležitou součástí, je složka *templates*, která obsahuje veškeré HTML soubory čili provádí samotné zobrazení dat v internetovém prohlížeči [5].

### 1.1.2 Databáze

Databáze je velmi důležitou součástí každé aplikace psané v Django. Jedná se o seřazený soubor dat či informací, v tomto konkrétním případě dostupným pomocí aplikace [9]. Existují různé způsoby ukládání dat, díky kterým můžeme rozlišovat tzv. databázové modely. Konkrétně se jedná o:

- hierarchickou databázi;
- síťovou databázi;
- relační databázi [10].

Největší význam má relační databázový model, jelikož je nejpoužívanější a pro cíl této práce je nejlépe využitelný. Byl vytvořený roku 1970 Dr. Edgarem Frankem Coddem, který taktéž definoval dvanáct pravidel, kdy pracuje databáze jako relační. [10,11]

Dva hlavní druhy databází, které jsou v této aplikaci použitelné, jsou: tzv. SQL databáze – relační model; a NoSQL databáze – nerelační model.

Dle odborníků se zdvojnásobí objem světových dat každé dva roky, s čímž také spočívá fakt, že, aby se nárůst v oblasti big data zachytil a bylo

možné ho skladovat, je nutné rozšiřovat kapacity. Ještě v době nedávné se většina světových dat ukládala v papírových složkách, nicméně nyní je možné data efektivně ukládat díky IT technologiím. Díky tomu lze přeplněné kanceláře plné stojanů plných složek nahradit „plochými“ soubory. [12]

### 1.1.2.1 SQL databáze

SQL databáze, které se dají zařadit mezi databáze relační, ačkoli mezi ně přímo nepatří, jsou velmi hojně využívány v běžných databázových aplikacích. SQL znamená „Structured Query Language“, v překladu strukturovaný dotazovací jazyk. Vznikl z jazyku SEQUEL („Structured English Query Language“ – strukturovaný anglický dotazovací jazyk), který v 70. letech 20. století vyvinula společnost IBM, konkrétně Donald D. Chamberlain a Raymond F. Boyce. [13]

V dnešní době, se MySQL, což je implementace SQL, využívá v takových aplikacích, jako například: sociální sítě Facebook a Twitter, vyhledávač Google, či dokonce YouTube. [12]

#### 1.1.2.1.1 Příkazy jazyka SQL

Mezi výhody SQL databází patří jistě unifikovanost jazyka. Díky tomu můžeme příkazy tohoto jazyka rozdělit do čtyř skupin:

- *Příkazy pro manipulaci s daty* (DML – Data Manipulation Language); které slouží ke čtení, ukládání či odstraňování dat. Do této skupiny můžeme zařadit příkazy SELECT, INSERT, UPDATE či DELETE [14].
- *Příkazy pro definici dat* (DDL – Data Definition Language), sloužící pro popsání datových struktur tabulek, indexů či pohledů a jejich editaci. K těmto příkazům můžeme přiřadit: CREATE, ALTER, DROP [14].
- *Příkazy pro řízení* (DCL – Data Control Language), které slouží pro řízení přístupových práv a ovládání transakcí. Zde mohou být jmenovány příkazy: GRANT, REVOKE, BEGIN, COMMIT nebo ROLLBACK [14].



- *Ostatní či speciální příkazy.*

#### 1.1.2.1.2 Manažeři SQL databází

Jedním z manažerů užívaných pro správu SQL databází je **Oracle Database**. Jedná se o velmi využívaný v oblasti databázových řešení. Je využíván například ve státní správě. Mezi jeho výhody patří například, že tento klient patří mezi standardy databázových systémů. Také ho lze využít i na těch nejzatíženějších systémech. [14]

Avšak na druhou stranu, jedná se o licencovaný produkt, jehož ceny nejsou velmi nízké. Liší se počtem procesorů, na které jsou využívány a ceny se pohybují v řádech stovek až desetitísiců dolarů [15]. Taktéž jeho hardwarové požadavky jsou vysoké. [14]

Dalším využívaným klientem je kupříkladu **SQL Server**. Tento klient je vydáván přímo korporací *Microsoft*, z čehož automaticky vyplývá první nevýhoda – je prakticky nemožné ho migrovat na jiný operační systém. Taktéž vysoká rozšiřitelnost operačního systému Windows přináší poměrně vysokou bezpečnost rizika. [14]

Ovšem je cenově příznivější než *Oracle Database* a další výhodou je, že je dobře integrovatelný v operačních systémech Windows. Taktéž ve spojení s některými jinými softwary společnosti *Microsoft* je možné ještě více snížit náklady na *SQL Server*. [14]

Posledním zde zmíněným manažerem je **MySQL Server**. Jedná se o nejrozšířenější open-source relační databázový systém. S tímto faktem také souvisí jeho výhody: jeho základní verze není nijak licencována, jde velmi dobře spustit na většině počítačů a velkém množství platform. [14]

Také s faktem, že se jedná o open-source software, souvisí i jeho nevýhody. Jedna z nich je, že může být snazší proniknout do systému, či že mu chybí některé funkce, které jsou v jiných klientech dostupné. [14]

### 1.1.2.2 NoSQL databáze

NoSQL databáze jsou nerelačním typem databází, které podporují distribuovanou architekturu. Jejich výhodou je, že jsou volně šiřitelné. NoSQL znamená v jednom z využívaných překladů „nejen SQL,“ čili jsou schopny dotazování nejen v jazyce SQL. Tento druh databází nabízí velmi dobrou možnost správy dat na tzv. „real-time webech“, čili aplikacích, kde je nutné, či přinejmenším dobré zpracovávat dotazy apod. v reálném čase čili okamžitě. To také souvisí se skutečností, že tyto databáze mohou pracovat s velkým objemem dat. [16]

NoSQL databáze nejsou relační, což v podstatě znamená, že nemají žádnou tabulku, čímž umožňují rychlejší správu dat. I díky tomu jsou více flexibilní. Zároveň se jedná o tzv. open-source software. Taktéž jejich tvorba a správa je velmi levná, tudíž jsou velmi vhodné pro aplikace neziškových společností, či v případech, kdy je na aplikaci pouze velmi omezený finanční rozpočet, s čímž také velmi úzce souvisí malá náročnost na použitý hardware. [16]

Další nezkrotnou výhodou je možnost využití tzv. „MapReduce“, což je „*programovací model pro zpracování velkých množin dat pomocí paralelního zpracování*“ [17]. Díky tomu je velmi dobře škálovatelný [12].

Jako poslední významnou výhodou, kterou lze zmínit, je potřeba nevytvářet podrobný databázový model, s čímž opět koresponduje snadnost těchto databází a nerelační povaha této databáze. S touto skutečností jde ruku v ruce skutečnost, že tvorba NoSQL databází je velmi časově úsporná. [12]

Nevýhody vyplývají ze skutečnosti, že tato metoda je relativně velmi nová. Jedna z nich je špatná definovatelnost komunity NoSQL. To souvisí se skutečností, že tato komunita je poměrně mladá a nevyzrálá čili koncoví uživatelé nejsou ještě tolik zkušení, jako například v případě MySQL. [12]

Další nevýhodou je fakt, že v NoSQL zatím není dostatečné množství nástrojů, které vytvářejí přehledy pro analýzu a testování výkonu. [12]

Poslední nevýhodou, kterou lze definovat jako velmi zásadní je skutečnost, že v NoSQL je nízká míra standardizace. To lze opět srovnat s SQL databázemi, které nabízejí unifikovaný jazyk dotazu. [12]

#### 1.1.2.2.1 Modely NoSQL databází

Prvním modelem je *model s klíčovými hodnotami*. Jedná se o nejméně složitý model NoSQL databází. V principu uchovává data způsobem bez schématu, který je tvořen z oindexovaných klíčů a jejich hodnot. Tato metoda nabízí vysokou škálovatelnost (vhodný pro big data databáze), výkon a flexibilitu při téměř žádné náročnosti. Příklady těchto databází jsou například:

- Azure,
- LevelDB,
- Riak,
- Apache's Cassandra. [18]

Dalším je *model se sloupcovým uložištěm*. Také nazýván model se širokým sloupcovým uložištěm. Ukládá data v tabulce jako sloupce (na rozdíl od SQL databází, která uchovává data v řádcích). Ovšem se nejedná pouze o inverzní tabulku. Díky tomuto způsobu ukládání nabízí model se sloupcovým uložištěm vysoký výkon, škálovatelnost, avšak flexibilita již není tak značná jako v prvním případě, stejně tak jako složitost, která je vyšší. Příkladem těchto databází mohou být softwary:

- Apache's HBase,
- BigTable,
- HyperTable. [18]

Třetím typem je *dokumentová databáze*. Tento model má koncepci klíč-hodnota. Každý dokument užívá vlastní unikátní klíč, který se užívá

k jeho načtení. Je vhodný pro správu dat, které jsou dokumentově orientovaná, avšak stále strukturovaná. Vyznačuje se vysokým výkonem, proměnlivou, avšak vysokou škálovatelností a vysokou flexibilitou. Je opět složitější než model s klíčovými hodnotami. Databáze užívající tento model jsou například:

- Apache's CouchDB,
- MongoDB. [15]

Posledním typem je *model s grafovou databází*. Tento model je málo využívaný kvůli jeho obtížnosti. Je ovšem velmi flexibilní, což je klíčový znak NoSQL databází. Jeho výkonnost a škálovatelnost je proměnlivá. Je využíván v případech, kdy jsou data nejlépe prezentovatelná jako grafy. Příklady těchto databází jsou:

- Polyglot,
- Neo4J. [15]

### 1.1.2.3 Srovnání typů databází

Během tvorby databáze je nutné počítat s jejími nevýhodami, kupříkladu v případě NoSQL malou standardizací jazyku. Ačkoli je možné, že budou výsledkem měření užívané „nestandardní“ výstupy, na něž by bylo vhodnější použít databázi NoSQL, jazyk SQL zvládá nepřeberné množství různých typů dat a lze ho, maximálně s drobným zásahem programátora využít na téměř veškeré výstupy měření. Z tohoto hlediska by se tedy za vhodnější databázový typ dalo považovat SQL. Dále i z důvodu, že aplikace již běží za pomoci SQL databáze je velmi vhodné ji využívat i nadále.

### 1.1.3 Internetové protokoly

K přenosu dat mezi počítači, popřípadě mezi počítačem a serverem existuje několik různých protokolů, které se liší jejich funkcí. Jedná se tudíž o velmi důležitou součást této aplikace. Existuje několik „vrstev“ protokolů, a to sice: *Aplikační a prezentační vrstva* (protokoly HTTP, FTP, DNS,

POP3 a další), *transportní vrstva* (protokoly TCP, UDP), *síťová vrstva* (protokoly IP, ICMP) a *linková vrstva* (Ethernet, WiFi či Bluetooth). Pro užití v této aplikaci je, jak již z názvu vyplývá, první vrstva, proto dále bude zmíněna hlavně ona. [19]

### 1.1.3.1 Aplikační a prezentační vrstva

Jedná se o nejpočetnější vrstvu. Jejím cílem je zajistit komunikaci mezi programy. Také je to vrstva, se kterou se běžní uživatelé setkávají nejčastěji. [19]

Prvním z důležitých protokolů této vrstvy je HTTP. *„Jedná bezstavový protokol, který funguje způsobem dotaz-odpověď. Uživatel pošle serveru dotaz ve formě čistého textu, obsahujícího označení požadovaného dokumentu, informace o schopnostech prohlížeče apod. Server následně odpoví taktéž v textové podobě. Jednotlivé dotazy mezi sebou nemají žádnou spojitost“* [19]. Samotný HTTP je velmi užívaný, nicméně není šifrovaný, což ovlivňuje bezpečnost jeho užití. Existuje ovšem i šifrovaná verze HTTPS. Pro použití v příkladech, jako je aplikace, která je cílem práce, je jeden z nejvyužívanějších. [19]

Zabezpečený HTTP čili HTTPS je nutné použít kdykoli v případech, kdy se od uživatele přenášejí citlivá data. Jedná se například o: obchodování na e-shopech, provádění transakcí platební kartou nebo přenášení citlivých údajů (e-mailové adresy, doručovací adresy, a jiné). [20]

HTTPS používá protokol HTTP společně s protokolem SSL. Ve své podstatě zajišťuje autentizaci a důvěrnost přenášených dat. Byl vyvinut v roce 1994 společností Netscape Communication. [20]

Protokol FTP slouží pro přenášení souborů prostřednictvím sítě. Je velmi oblíbený pro svou jednoduchost, stejně tak jako možnost užití bez nutnosti instalace speciálního softwaru (ačkoliv se to doporučuje), ale jeho velkou nevýhodou je nemožnost šifrovat přenášené soubory, stejně tak jako například hesla. Také se jedná o dvouportový přístup (TCP porty 20 a 21). [19, 21]

Jeho další výhodou je, že umožňuje různé nastavení přístupu, například pro mazání, čtení či nahrávání souborů. Je postupně nahrazován protokolem SFTP. [21]

Ačkoli by to bylo, dle zkratky, nasnadě, tento protokol nemá nic společného s protokolem FTP kromě názvu („protokol pro přenos souborů“). [21]

Jeho výhodou je bezesporu možnost šifrovat přenášená data, což zajišťuje vyšší bezpečnost než při používání FTP, stejně tak jako použití jednoho standardního portu či velké množství klientů. Naopak, pro část uživatelů, je nevýhodou to, že musí instalovat speciální software (balíček OpenSSH). [21]

## 1.2 Rozdíly Pythonu mezi verzí 2 a 3

Od roku 2008 je k dispozici nová verze – Python 3.0, která nastolila velkou změnu v chápání některých funkcí a zápisů. Vývoj Pythonu se dá považovat za evoluční, s čímž také jde ruku v ruce značná nevýhoda – není zpětně kompatibilní s verzemi 2.X. Jediným mezistupněm je verze 2.7. V současné době je nejaktuálnější verze Pythonu verze 3.6.1, která je v provozu od 21. března 2017. [1]

Python 3.0 je první aktualizací jazyka, která není zpětně kompatibilní. Je také známá pod názvy „Python“ 3000 nebo „Py3k“. V této aktualizaci se nachází více novinek, než je obvyklé, ale většina z nich jsou pouze kvůli odstranění známých chyb jazyka. Jedná se o například: [22, 23]

Print se nyní chová jako funkce čili již se nejedná o pouhé „prohlášení“. V Pythonu 3.0 je tedy nutné užití závorek, například: `print("výsledek je", 8*8)`; oproti dřívějšímu zápisu: `print "výsledek je", 8*8`. [22, 23]

Další viditelnou aktualizací je kupříkladu práce s celými čísly. Ve starších verzích platilo při zápisu:  $5/2 = 2$ , stejně jako  $5//2 = 2$ . Ve verzi 3.X již platí, že:  $5/2 = 2.5$  a  $5//2 = 2$ . [22, 23]

## 2 Přídavné nástroje

Aplikace potřebuje ke svému správnému chodu ještě dodatečné nástroje. Jelikož několik z nich je již v původní aplikaci implementováno, budou zde zmíněny jen ty, které byly nutné pro pokročilé funkce, kterými se tato práce zabývá.

### 2.1 Skriptovací jazyk JavaScript

JavaScript je vedle HTML a CSS jedním z nejdůležitějších jazyků pro tvorbu webu. Programuje vlastní chování dané webové stránky. Výhodou JavaScriptu je, že má velké množství předprogramovaných knihoven, které vývojáři mohou jeho práci velmi usnadnit. [24]

#### 2.1.1 Knihovna jQuery

Jednou z nich je i knihovna *jQuery*, která byla vytvořena lednu 2006 Johnem Resingem. Jedná se o open-source knihovnu, jejíž používání je svobodné pod MIT licencí. [25]

Ačkoliv obsahuje velké množství funkcí, je malá a poměrně rychlá. Dle autorů dokáže díky API, které je snadné k použití, a navíc funguje v mnoha prohlížečích, například přechod a manipulaci s daty, zpracovat události či animace. [26]

Má mírně odlišnou syntaxi ve srovnání s klasickým JavaScriptem. Kupříkladu k práci s jQuery se používá znak dolaru – „\$“, což je ve skutečnosti zvolená funkce tak, aby bylo nutné psát co nejméně kódu. [27]

Taktéž u práce s HTML tagy, která je u JavaScriptu poměrně zdlouhavá, nabízí velké zjednodušení práce. Zatímco u JavaScriptu, pokud chceme pracovat s některým objektem zadaným jeho ID, musí vývojář napsat `document.getElementById("ID_objektu")` u knihovny jQuery postačí jen `"#ID_objektu"`. [27]

## 2.1.2 Knihovna Bootstrap

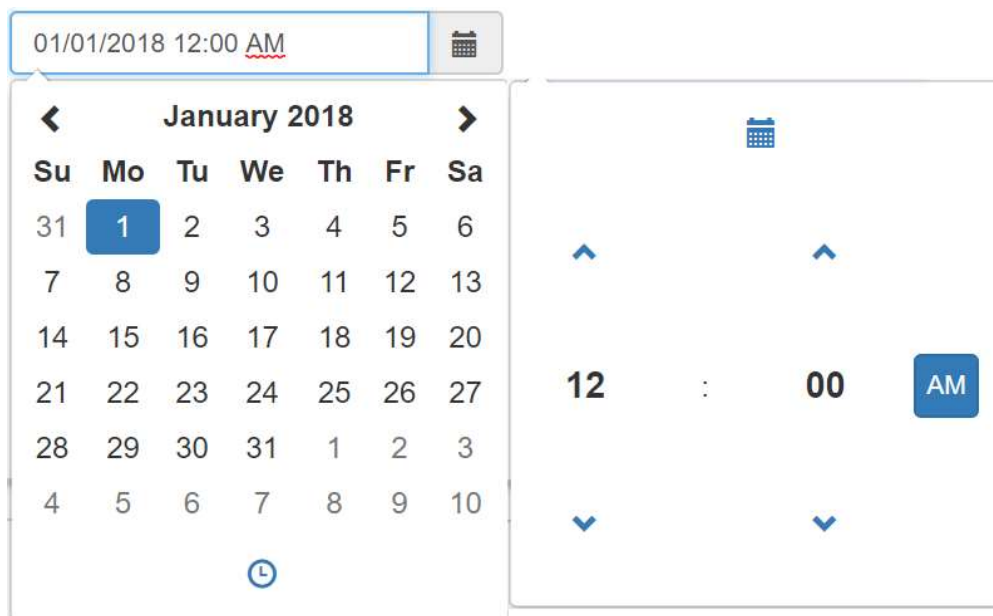
Bootstrap byl vyvinut Markem Ottem a Jacobem Thorntonem na sociální síti Twitter. Jednalo se o framework, který podporoval konzistenci mezi jednotlivými nástroji. V podstatě sjednotil velké množství různých knihoven, které byly velmi náročné na údržbu, do jedné. Byl vydán v srpnu 2011 jako Twitter Bootstrap a již o šest měsíců později se stal nejoblíbenějším vývojářským projektem na GitHubu. Nyní je dostupný, stejně jako jQuery, jako open-source pod MIT licenci. [28]

Jak z výše uvedeného vyplývá, jedná se o velmi obsáhlou knihovnu, která nespočívá jen v samotném JavaScriptu, ale jsou v něm například implementovány nepřehledně obsáhlé kaskádové styly. Ty již jsou v aplikaci integrovány. Minimálně pro některé vývojáře je nevýhodou, že je postavený právě na knihovně jQuery a potřebuje tedy pro svou úplnou funkčnost její implementování do aplikace. [29]

### 2.1.2.1 Bootstrap DateTime Picker

Jednou z knihoven implementovaných v Bootstrapu je *DateTime Picker* [30]. Jedná se v podstatě o knihovnu, která dokáže klasický HTML tag *input type="text"* přeformátovat na vstup, který, pokud na něj návštěvník webové stránky klikne, rozbalí kalendář, ve kterém lze vybrat požadované datum a čas. Existuje mnoho různých způsobů programování takového, jaké vývojář požaduje a který je vhodný pro použití v té dané oblasti na světě.





Obrázek 2.1 - Ukázka použití Bootstrap 3 DateTime Pickeru

### 2.1.2.2 Bootstrap Slider

Ačkoliv HTML5 zná posuvníky, konkrétně pomocí tagu `input type="range"`, jedná se pouze o posuvník, kde se dá nastavit pouze jedna hodnota. To by při zamýšleném použití v této aplikaci, pro filtraci dat dle množství, bylo velmi nedostačující a musely by být použity posuvníky dva.

Zde přichází dobrá možnost s knihovnou *Bootstrap Slider* [31]. Podobně jako u DateTime Pickeru se jedná o knihovnu, která dokáže přeformátovat jednoduchý `input type="text"`. Nicméně u tohoto je možné implementovat dvouhodnotový posuvník s tím, že jeho výstupem je dvoumocné pole. Samozřejmě za pomoci Bootstrap kaskádových stylů je možné vytvořit vzhled, který vývojáři vyhovuje. Pro zjednodušení zapsání kódu je samozřejmě možné využít jQuery.

## 2.2 Tvorba grafů

Pokud chce vývojář používat grafické zobrazení dat v aplikaci, která běží v Django, existují v podstatě dvě možnosti, jak toho dosáhnout. První

možností je připravit graf přímo v Pythonu a pomocí HTML ji pouze zobrazit, druhou pak vytvořit až při samotném zobrazování v HTML.

## 2.2.1 Python

Při zobrazování grafů v Pythonu se dostáváme k určitým, a pro některé aplikace poměrně zásadním, omezením. Jedním z nich může být kupříkladu nemožnost interaktivity daného grafu ve webovém rozhraní, což je v dnešní době, dá se říci, standardem.

Jako běžná volba pro grafické zobrazování dat v Pythonu funguje knihovna *Matplotlib*.

V prvopočátku má vývojář na výběr ze dvou API. Jedno z nich je zaměřené objektově, druhé obsahuje mnoho funkcí a, podobně jako v MATLABu, globální proměnné. [32]

*„Matplotlib v podstatě umí:*

- *vytvářet 2D a 3D grafy,*
  - *vytvářet popisky v LATEXu,*
  - *kontrolovat nastavení všech grafických prvků v grafu,*
  - *vytvořit kvalitní grafický výstup, včetně formátů PNG, PDF, SVG, EPS;*
  - *interaktivní práci s grafy (přibližování, posouvání atd.) – díky GUI“*
- [32]

Použití knihovny *Matplotlib* je názorně ukázáno na příkladu níže. Společně s touto knihovnou je použita i knihovna *numPy*<sup>1</sup>. Jedná se o dva obrázky, přičemž první reprezentuje zapsaný kód v Pythonu, zatímco druhý přímo grafickou reprezentaci jednoduchého grafu funkce  $y = x^3$ .

---

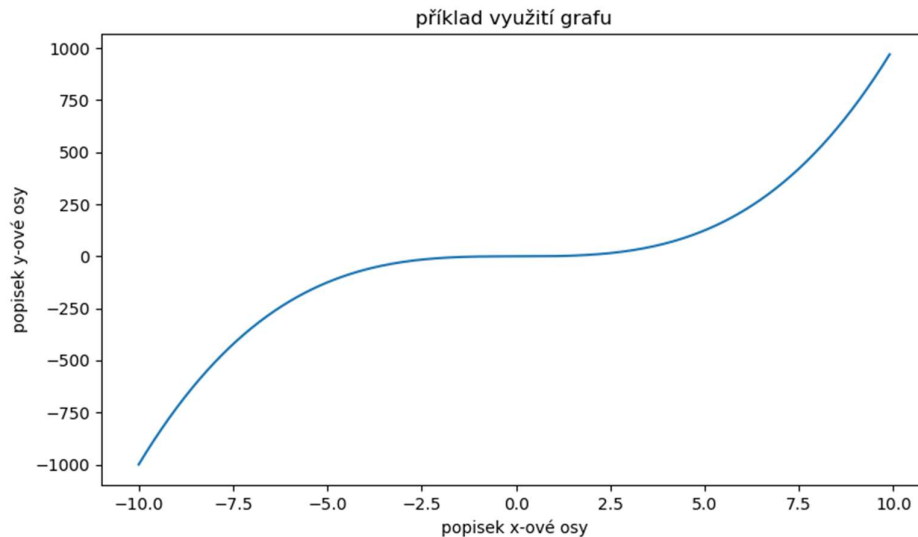
<sup>1</sup> „**NumPy** je základní knihovna pro vědce a analytiku, kteří pracují s Pythonem. Definiuje typ pro  $n$ -rozměrné homogenní pole (nejčastěji čísel) a API pro práci s takovým polem.“ [33]

```

>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> x=np.arange(-10, 10, 0.1)
>>> y=x**3
>>> fig = plt.figure()
>>> axes=fig.add_axes([0.1, 0.1, 0.8, 0.8])
>>> axes.plot(x,v)
>>> axes.set(xlabel='popisek x-ové osy', ylabel='popisek y-ové osy', title='příklad využití grafu')
>>> plt.show

```

**Obrázek 2.2 - Ukázka kódu k vytvoření grafu v Pythonu**



**Obrázek 2.3 - Ukázka grafu vytvořeného pomocí knihovny Matplotlib v Pythonu**

Podobně, jako tento čárový graf, se dají dělat i například grafy sloupcové, bodové, schodové či i graf s výplní mezi dvěma křivkami, stejně tak jako histogramy, či polární grafy. [32]

Podobným způsobem se dají tvořit i trojrozměrné grafy, nicméně se předpokládá, že v této aplikaci nebude nezbytně nutné je použít.

## 2.2.2 JavaScript

Pro vytváření grafů v HTML se běžně používají knihovny psané v JavaScriptu. Dnes je jich k dostání celá řada, mnohé z nich jsou open source. Jejich užití je různě složité a samotné knihovny jsou různě aktualizovány a udržovány. Pro tuto aplikaci je vhodné volit knihovnu, která je stabilní a udržována, což níže uvedené knihovny jistě splňují.

### 2.2.2.1 Knihovna D3.js

*D3.js*, neboli *Data-Driven Documents* (datově řízené dokumenty), je jednou z dnes nejvyužívanějších knihoven. Opět se jedná o open-source software. Pro vykreslování grafů používá HTML, SVG a CSS. Mezi její výhody patří kupříkladu: výkonné přechody, vysoká flexibilita funkce „Enter()“ a „Exit()“, nebo syntaxe velmi podobná jQuery. [34]

Má ovšem i nevýhody. Jednou z nich je například nepodpora starších prohlížečů, jako je například Internet Explorer 8. Další nevýhodou, která je velmi významná při rychlejší tvorbě aplikace, nebo při kratším „deadlinu“, je fakt, že na rozdíl od většiny JavaScriptových knihoven pro tvorbu grafů, neobsahuje žádné přednastavené grafy. Vše si tudíž musí programátor vytvořit sám. [34]

Mezi aplikace, které v minulosti velmi využívaly D3.js, patří například web New York Times, Uber nebo Weather.com. V současnosti je využíván kupříkladu některými burzami s kryptoměnami. [34]

Tento příklad byl vytvořen podle příkladu použití čárového grafu v aplikaci SidePoint.com [35]. Je tu pominuto nastavení kaskádových stylů, a to konkrétně pro nastavení čar a popisků os.

1. Vytvoření SVG „kreslicího plátna“ pro vykreslení grafu a importování d3.js knihovny:

```
<svg width="1000" height="500" id="test"></svg>  
<script src="https://d3js.org/d3.v3.min.js"></script>
```

*Obrázek 2.4 - 1. část ukázky kódu pro vytvoření grafu pomocí D3.js*

2. Následuje vložení dat. Tento způsob jde nahradit importem ze souboru formátu TSV:

```
var lineData = [{
  'x': 0,
  'y': 13
}, {
  'x': 20,
  'y': 23
}, {
  'x': 40,
  'y': 13
}, {
  'x': 60,
  'y': 40
}, {
  'x': 80,
  'y': 0
}, {
  'x': 100,
  'y': 69
}, {
  'x': 110,
  'y': 23
}
];
```

*Obrázek 2.5 - 2. část ukázky kódu pro vytvoření grafu pomocí D3.js*

3. Dalším krokem je vytvoření os x a y. Element „domain“ v tomto případě značí minimum a maximum osy. Dá se nahradit funkcí, která nastavuje rozsah automaticky dle dat. Je zde definována šířka, výška a okraje grafu. Osa y zde musí být orientována doleva. Okraje jsou definovány tak, aby se nedotýkaly okrajů prostoru SVG:

```

var vis = d3.select("#test"),
    WIDTH = 1000,
    HEIGHT = 500,
    MARGINS = {
      top: 20,
      right: 20,
      bottom: 20,
      left: 50
    },
    xRange = d3.scale.linear().range([MARGINS.left, WIDTH - MARGINS.right]).domain([0,120]),

    yRange = d3.scale.linear().range([HEIGHT - MARGINS.top, MARGINS.bottom]).domain([0,100]),

    xAxis = d3.svg.axis()
      .scale(xRange)
      .tickSize(5)
      .tickSubdivide(true),

    yAxis = d3.svg.axis()
      .scale(yRange)
      .tickSize(5)
      .orient("left")
      .tickSubdivide(true);

vis.append("svg:g")
  .attr("class", "x axis")
  .attr("transform", "translate(0," + (HEIGHT - MARGINS.bottom) + ")")
  .call(xAxis);

vis.append("svg:g")
  .attr("class", "y axis")
  .attr("transform", "translate(" + (MARGINS.left) + ",0)")
  .call(yAxis);

```

*Obrázek 2.6 - 3. část ukázky kódu pro vytvoření grafu pomocí D3.js*

4. Předposledním krokem je koordinace os a jejich transformace do kreslicího plátna. Poslední řádek v tomto skriptu značí způsob interpolace dat. Je zvolena „basis“, která se nejvíce podobá vykreslování grafů z bodů dle studia na Fakultě strojní:

```

var lineFunc = d3.svg.line()
  .x(function (d) {
    return xRange(d.x);
  })
  .y(function (d) {
    return yRange(d.y);
  })
  .interpolate('basis');

```

*Obrázek 2.7 - 4. část ukázky kódu pro vytvoření grafu pomocí D3.js*

5. Závěrečným krokem je nastavení atributu d z SVG cesty na souřadnice, které nám určila funkce. Dále je zde nastavena barva čáry, její šířka a vyplnění:

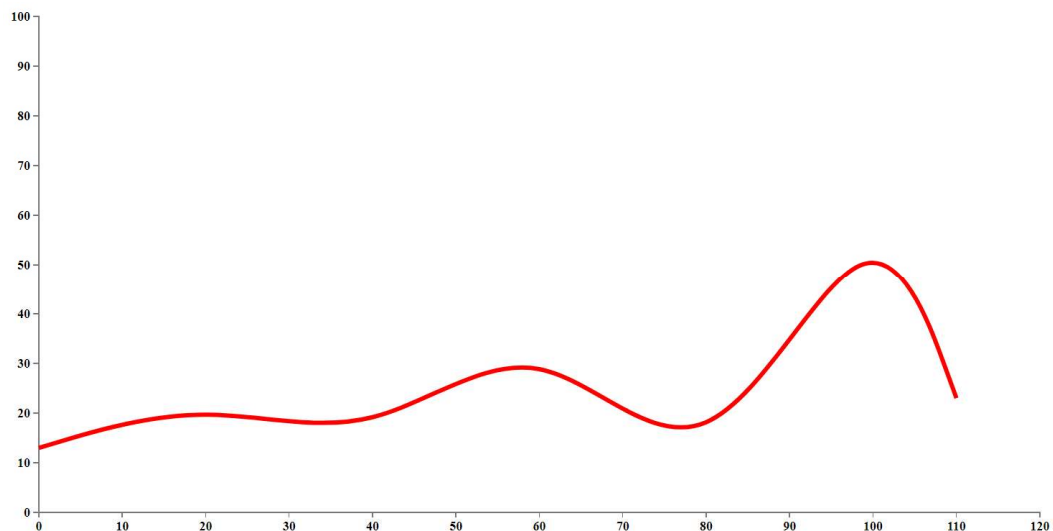
```

vis.append("svg:path")
  .attr("d", lineFunc(lineData))
  .attr("stroke", "red")
  .attr("stroke-width", 4)
  .attr("fill", "none");

```

Obrázek 2.8 - 5. část ukázky kódu pro vytvoření grafu pomocí D3.js

Po sepsání tohoto kódu se vygeneruje velmi jednoduchý graf. Ačkoli bylo sepsání kódu srovnatelné, ne-li těžší než při tvoření grafu pomocí knihovny Chart.js, je tento graf velmi jednodušší, bez interaktivity.



Obrázek 2.9 - Ukázka grafu vytvořeného pomocí knihovny D3.js

### 2.2.2.2 Knihovna Chart.js

Další knihovnou je Chart.js. Tato knihovna nabízí mnohé možnosti tvorby grafů. Jeho výhodou je jednoduchost nebo poměrně dobrá interaktivita. Pro vykreslování používá tag *canvas*, který je součástí *HTML 5*. Mezi jeho další plusy patří i to, že nabízí alternativy i pro starší prohlížeče, jako kupříkladu *Internet Explorer 7*, či jeho osmá generace. Taktéž nabízí dobré zobrazení i pro chytré telefony a tablety. Jedná se o open-source software, který je šířen pod MIT licencí. [34]

Nabízí šest různých základních typů grafů, které dokáže i kombinovat:

- čárový,

- sloupcový (jak horizontální, tak vertikální),
- síťový,
- výsečový,
- bublinový a
- graf polární oblasti. [36]

Podobně jako v předchozím případě, i u Chart.js je níže vypracovaný příklad. Je ovšem vypracován podrobněji, jelikož se jedná o vysoce vhodnou knihovnu pro použití v aplikaci. Byl vytvořen podle vzoru na blogu Tobiaše Ahlina [37].

1. Integrovaní knihovny Chart.js do aplikace:

```
<script src="C:\Users\User\node_modules\chart.js\dist\docs\gitbook\gitbook-plugin-chartjs\Chart.js"></script>
```

*Obrázek 2.10 - 1. část ukázky kódu pro vytvoření grafu pomocí Chart.js*

Tento skript je možné nahradit i URL adresou, pokud je k dispozici.

2. Vytvoření canvasu, pro vložení grafu:

```
<canvas id="testovaciGraf" width="200" height="100"> </canvas>
```

*Obrázek 2.11 - 2. část ukázky kódu pro vytvoření grafu pomocí Chart.js*

3. Vložení samotného skriptu pro vytvoření grafu:

Tento graf se pomocí ID vnoří do výše zmíněného elementu canvas. Obsahuje několik důležitých částí, jako typ grafu (type: 'line'), data (včetně barvy čáry a popisku) nebo popisků os.



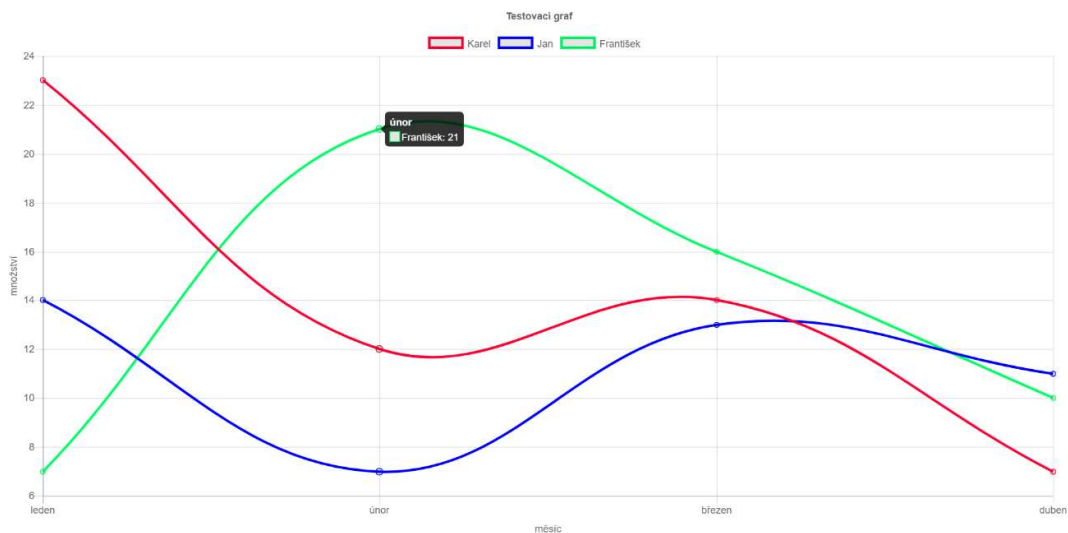
```

<script>
new Chart(document.getElementById("testovacíGraf"), {
  type: 'line',
  data: {
    labels: ['leden', 'únor', 'březen', 'duben'],
    datasets: [
      {
        data: [23,12,14,7],
        label: "Karel",
        borderColor: "#F03",
        fill: false
      },
      {
        data: [14,7,13,11],
        label: "Jan",
        borderColor: "blue",
        fill: false
      },
      {
        data: [7,21,16,10],
        label: "František",
        borderColor: "#00FF66",
        fill: false
      }
    ]
  },
  options: {
    title: {
      display: true,
      text: 'Testovací graf'
    },
    scales: {
      xAxes: [
        {
          display: true,
          scaleLabel: {
            display: true,
            labelString: 'měsíc'
          }
        }
      ],
      yAxes: [
        {
          display: true,
          scaleLabel: {
            display: true,
            labelString: 'množství'
          }
        }
      ]
    }
  }
});
</script>

```

**Obrázek 2.12 - 3. část ukázky kódu pro vytvoření grafu pomocí Chart.js**

V závěru už pouze vznikne graf, který je interaktivním v tom slova smyslu, že dokáže při najetí na jedno z dat ukázat jeho příslušnost k čáře a jeho pozici na x-ové, potažmo y-ové ose.



**Obrázek 2.13 - Ukázka grafu vytvořeného pomocí knihovny Chart.js**

### 2.2.2.3 Knihovna GoogleCharts

Google Charts je výhodná knihovna pro ty, kteří chtějí mít grafy vytvořené pomocí JavaScriptu velmi jednoduše. Nabízí velké množství přednastavených grafů, a to i „nezvyklé“ z pohledu JavaScriptových knihoven, jako „Geo Charts“ (grafy, které na mapě světa zobrazují data podle států, či podle jiných komponentů map), nebo kalendáře, kde lze poměrně jednoduše udělat například diář. [34]

Podobně jako D3.js, Google Charts vykresluje grafy pomocí prvku HTML5, SVG. Tato kombinace nabízí dobrou přenositelnost mezi různými prohlížeči i mobilními zařízeními s operačními systémy iOS či Android. Díky VML jsou podporovány i starší prohlížeče. [34]

Značnou nevýhodou je, že ačkoliv se jedná o software s licencí zdarma, není open-source. V návaznosti s tím, podle licenčního ujednání společnosti Google, není možné umístit jejich JavaScriptové soubory na vlastní server. Toto je nevýhodou hlavně pro společnosti, které mají citlivá data. [38]

Následující příklad byl vytvořen za pomoci aplikace Google Charts [39], kde je téměř dokonalý výukový materiál, možnosti změn apod.:

Skripty pro tvorbu grafu jsou velmi jednoduché, navíc v aplikaci Google Charts je velmi dobrá odezva. Skripty jsou dva, první pro importaci knihovny Google Charts, druhý pro samotné vytvoření grafu. Oba jsou vnořeny v hlavičce.

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
  google.charts.load('current', {'packages':['corechart']});
  google.charts.setOnLoadCallback(graf);

  function graf() {
    var data = google.visualization.arrayToDataTable([
      ['Měsíc', 'František', 'Jan', 'Karel'],
      ['leden', 23, 24, 12],
      ['únor', 15, 48, 60],
      ['březen', 55, 7, 14],
      ['duben', 5, 12, 55],
      ['květen', 48, 23, 13]
    ]);

    var options = {
      title: 'Testovací graf',
      titleTextStyle: {fontSize: 20, bold: true},
      curveType: 'function',
      legend: { position: 'right', textStyle: {bold: true} },
      animation: {startup: true, duration: 2000, easing: 'inAndOut'}
    };

    var chart = new google.visualization.LineChart(document.getElementById('testovaci_graf'));

    chart.draw(data, options);
  }
</script>
```

**Obrázek 2.14 - 1. část ukázky kódu pro vytvoření grafu pomocí Google Charts**

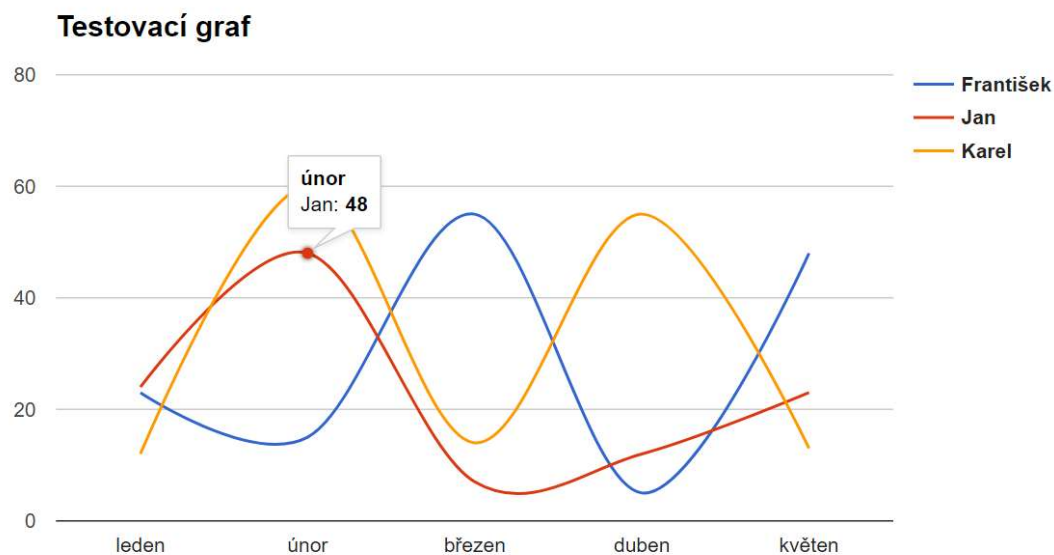
Data grafu jsou poměrně jasná, jsou zadávána podle náležitosti k hodnotě na horizontální ose. V nastavení vlastností grafu je většina hodnot téměř zbytečná, nicméně si myslím, že minimálně titulek (element Title) a legenda (element Legend), jsou navýsost vhodná. Element curveType nastavuje „zaoblení“ hran čáry tak, aby bylo možné interpretovaná data brát jako přesnější. Také v tomto nastavení lze nastavit spojnicí trendu. Animace už plní pouze efektní roli. [39]

Jak je popsáno výše, Google Charts používá pro vykreslování prvek SVG. V tomto případě je ovšem nahrazen jednoduchým elementem DIV. Jedná se o jediný nutný prvek pro graf v těle HTML stránky.

```
<div id="testovaci_graf" style="width: 900px; height: 500px"></div>
```

**Obrázek 2.15 - 2. část ukázky kódu pro vytvoření grafu pomocí Google Charts**

Tento poměrně jednoduchý skript zvládne vytvořit efektní i efektivní vizuální zobrazení dat:



Obrázek 2.16 - Ukázka grafu vytvořeného pomocí knihovny Google Charts

### 2.2.3 Srovnání knihoven pro tvorbu grafů

Pro zobrazení dat v této aplikaci se, dle názoru autora a dle dnešních poměrně vysokých nároků na nejen efektivitu, ale i efektnost, spíše nehodí knihovna *Matplotlib* obsažená v Pythonu. Také nepříliš ideální je JavaScriptová knihovna *D3.js*, a to sice z důvodu, že neobsahuje přednastavené grafy a tím pádem je to velké zdržení i při minimální změně zobrazování.

Na druhou stranu knihovny *Google Charts* i *Chart.js* jsou pro tvorbu dat poměrně jednoduché, a navíc i poměrně sofistikované. Proto by se grafy v této aplikaci mohly tvořit jednou z těchto dvou knihoven. Pro *Chart.js* hovoří lepší organizace dat při importování do grafu a možnost umístění knihovny na vlastním serveru. *Google Charts* disponuje možná jednodušší strukturou skriptu, ve které je lepší organizace. Také automaticky definuje barvu čáry u čárového grafu, což je možná také usnadnění. Ovšem z důvodu umístění na serverech Google se jeví jako potenciálně riziková s ohledem na soukromí naměřených dat.

## 2.3 Knihovny pro export

### 2.3.1 Knihovna ReportLab

Jedná se o knihovnu, která byla založena již v roce 2000. Poskytuje možnost tvorby dynamických PDF dokumentů při dosažení vysoké rychlosti tvorby a zároveň složitosti. [42]

Nabízí velké množství různých druhů PDF dokumentů ve spojení s vývojářskou pomocí při jejich tvorbě. S tím se ovšem váže finanční náročnost (od 1 275 britských liber ročně [42]). Je možné ovšem, společně s použitím dalších knihoven, jako *xhtml2pdf*, *html5lib* a *pypdf* vygenerovat PDF i bez zakoupení této licence. [42]

Její značnou nevýhodou je vyšší složitost při tvorbě bez zakoupené licence a zásahu vývojářů společnosti *ReportLab*.

Mezi aplikace knihovny *ReportLab* patří například NASA, Hewlett-Packard, webové stránky seychelského turismu nebo samotná Wikipedie. [42]

### 2.3.2 Knihovna xhtml2pdf

Tato knihovna je již zmíněna výše, nicméně ji lze využít i samostatně, a to pomocí definování funkce *render\_to\_pdf*. Na rozdíl od výše zmíněného *ReportLab* je tato knihovna k dispozici zcela zdarma. [43]

Její nevýhodou je fakt, že nelze, za použití běžně složitých funkcí, vygenerovat PDF dokument z pouhé části HTML stránky. Další ne zcela pozitivní skutečností je fakt, že pro správné naformátování stránky a funkčnost generování je nutné využít kaskádové styly přímo vepsané do HTML šablony, což při použití takového množství definic kaskádových stylů v této aplikaci není dobré, hlavně z důvodu velmi špatné přehlednosti. [43]

Poslední nevýhodou, která je důležitá a měla by zde být zmíněna, je nemožnost jiného formátování generovaného PDF souboru oproti samotné HTML stránce. [43]

### 2.3.3 Knihovna WeasyPrint

Třetí možnou knihovnou, která by mohla být vhodná pro tvorbu PDF dokumentů v této aplikaci, je knihovna *WeasyPrint*. V současné době je možné ji využívat pod BSD licenci<sup>2</sup> a je plně funkční pro Python ve verzích 2.7 a 3.4 a novějších. [45, 46]

Na rozdíl od výše zmíněných má tato knihovna tu výhodu, že může u formátované HTML stránky změnit kaskádové styly přímo ve pythonovské funkci. Nicméně podobně jako u předchozích knihoven je při běžné složitosti není možné vygenerovat PDF pouze z části webové stránky, a musela by tudíž vzniknout „mezistránka“ s vygenerovaným odkazem. [45]

### 2.3.4 Knihovna Jinja2

Poslední zde zmíněnou knihovnou je *Jinja2*. Jedná se v podstatě o šablonovací systém vydaný vývojářskou skupinou *Pocoo* a je také šířena pod BSD licenci. V současné době běží ve verzi 2.10. [47]

Její naprostou výhodou je fakt, že je přímo tvořena v syntaxi frameworku Django. Mezi další výhody patří například: rozšiřitelnost (pomocí filtrů a pluginů), vysoká rychlost použití či snadná použitelnost a přehlednost. [47, 48]

Funguje v podstatě tak, že vývojář vytvoří šablonu v LaTeXu, do které pomocí Djanga (a v podstatě Pythonu) přidává proměnné. Práce s proměnnými i šablonami je v podstatě totožná jako při použití klasických

---

<sup>2</sup> „**BSD licence** je licence pro svobodný software, mezi kterými je jednou z nejsvobodnějších. Umožňuje volné šíření licencovaného obsahu, přičemž vyžaduje pouze uvedení autora a informace o licenci, spolu s upozorněním na zřeknutí se odpovědnosti za dílo.“ [44]

HTML šablon. Díky tomu, že je obsah přidáván do LaTeXové šablony, neexistuje zde problém s odlišným formátování generovaného dokumentu s webovou stránkou. Vývojář si jednoduše určí, jaké formátování stránky požaduje. [47]

Tuto knihovnu využívá například *Mozilla Firefox* či sociální síť *Instagram*. [47]

# Praktická část

## 3 Tvorba grafů

Pro vytvoření podkladů pro samotné vykreslování bylo zapotřebí několikerych skutečností.

### 3.1 Definice funkcí pro grafy

Tyto funkce se, stejně jako ostatní pro chod aplikace, nachází v souboru *views.py*. Bylo zapotřebí vytvořit tyto funkce. V závorce jsou uvedeny argumenty funkce, kde za zmínku stojí, že *idMeasurement* značí identifikační číslo daného měření a *idQuantity* identifikační číslo dané veličiny:

- ***displayChart (request)*** – tato funkce pouze načte stránku, kde lze vybrat vytvoření grafu „měření“ nebo grafu jednotlivé veličiny. Odkazuje na *chooseMeasChart* a *chooseMeasurement*;
- ***chooseMeasChart (request)*** – tato funkce vytvoří seznam všech měření daného uživatele pro pozdější vytvoření grafu „měření“;
- ***chooseQuMe (request, idMeasurement)*** je funkce, která tvoří seznam všech veličin pro vybrání první veličiny pro graf „měření“;
- ***chooseQu2Me (request, idMeasurement, idQuantity1)*** provádí podobnou činnost jako *chooseQuMe*, avšak vybírá již druhou veličinu pro graf a obsahuje tudíž jako argument první veličinu;
- ***measurementChart (request, idMeasurement, idQuantity1, idQuantity2)*** – funkce, která dává podklady pro vytvoření grafu „měření“ (v podstatě se jedná o srovnání dvou veličin z jednoho měření, předdefinované na stejný čas pořízení dat);



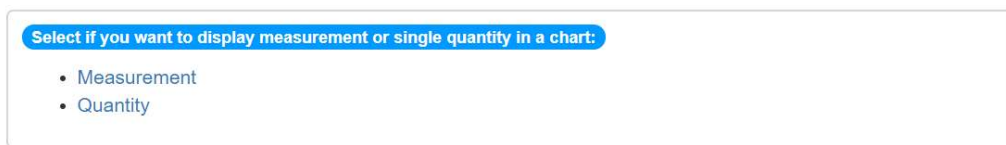
- *chooseMeasurement (request)* – tato funkce vytvoří seznam všech měření daného uživatele pro pozdější vytvoření grafu veličiny;
- *chooseQuantityChart (request, idMeasurement)* provádí výpis všech veličin z vybraného měření, ze kterých lze udělat graf. Je v podstatě obdobou funkce *chooseQuMe*;
- *quantityChart (request, idMeasurement, idQuantity)* je poslední funkcí, která byla vytvořena speciálně za účelem vytvoření grafů. Provádí v podstatě podobnou funkci jako *measurementChart*, nicméně, jak již název vypovídá, vytváří podklady pro graf veličiny.

## 3.2 Vytvoření HTML šablon

Tyto šablony jsou, opět stejně jako u funkcí, uloženy ve složce *templates* (v překladu šablony) a vlastně naprosto korespondují s funkcemi uvedenými v kapitole 4.1. Níže jsou uvedeny použité typy HTML podkladů.

### 3.2.1 Šablony pro výběr typu grafů

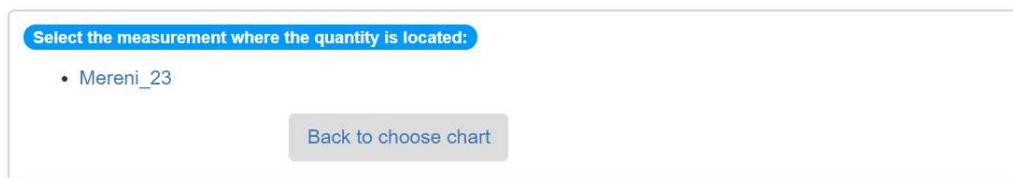
Zde se nachází pouze jednoduchý zápis v HTML, který vytváří seznam dvou odkazů, ze kterých lze vybrat typ grafu čili graf měření či graf jediné veličiny. Jedná se o podklad *displayChart.html*. Také stejně, jako ostatní šablony, pouze rozšiřuje základní stránku *base.html*.



Obrázek 3.1 - Zobrazení šablony pro výběr typu grafu

### 3.2.2 Šablony pro výběr měření a veličin

V podstatě se jedná o šablony *chooseMeasurement.html* a *chooseMeasChart.html*. Jejich úkolem je zobrazit vygenerovaná měření daného uživatele.



Obrázek 3.2 - Výstup šablony *chooseMeasurement.html*

Pro výběr veličiny, kterými se zabývají šablony *chooseQuantityChart.html*, *chooseQuMe.html* a *chooseQu2Me.html*, jsou použity velmi podobné přístupy, pouze jsou jinak pojmenovány proměnné.

### 3.2.3 Šablony pro zobrazení grafů

Vlastní tvorba grafů byla na počátku testována knihovnamy *Google Charts* a *Chart.js*. O obou, a dalších způsobech tvorby grafů, je významnější zmínka v teoretické části této bakalářské práce, konkrétně v kapitole 2.2. Po testování bylo shledáno, že z hlediska bezpečnosti a importu dat je vhodnější využít knihovnu *Chart.js*. Díky této knihovně, která je open-source a zároveň poměrně obsáhlá, je možné využít velkou škálu typů grafů.

Tento fakt ve spojení se skutečností, že zřejmě nelze určit „dokonalý“ a univerzální typ grafu, zapříčinil to, že je zde implementováno několik typů grafů: čárový, sloupcový, radarový pro oba druhy grafů a horizontální sloupcový pro grafy jednotlivých veličin.

Již dříve bylo zmíněno, že graf měření není přímo grafickým zpracováním celého měření, ale je vytvořen pouze pro srovnání dvou veličin daného měření, u kterých se zároveň předpokládá, že byly změřeny ve stejném čase. Podobně by se dal implementovat i graf pro zobrazení tří či více hodnot, nicméně by úměrně k tomu rostla obtížnost šablon a funkcí.

Během odladování zobrazování grafů byl zjištěn nedostatek použité knihovny. Spočívá ve skutečnosti, že typ x-ové osy je primárně nastavena na „category“, což v podstatě znamená, že se na ní počítá s typem string. Tato skutečnost nemusí být notně problémem, jelikož při použití v této aplikaci jsou na x-ovou osu položeny data naměření dané hodnoty.

Může se to jevit jako výhoda, že všechny hodnoty jsou od sebe ve stejné vzdálenosti. Nicméně při měřeních, kde záleží na čase a sleduje se vývoj měření. Pokud shodou okolností nebude po určitý čas snímač, který odesílá data, fungovat, může tato skutečnost způsobit nepoužití notné části grafu.

Bylo by vhodné na tomto místě poznamenat, že vydavatel knihovny *Chart.js* již pracuje na změně typu os a změna na číslo by měla být implementována v jedné z příštích aktualizací. Vzhledem k relativně snadné syntaxi grafu nebude pro budoucího správce aplikace velkým problémem typ osy změnit.

Níže jsou uvedeny příklady grafů pro oba způsoby.



**Obrázek 3.3 - Příklad typů grafů pro graf jedné veličiny**



**Obrázek 3.4 - Příklad typů grafu pro graf měření**

Je velice vhodné a výhodné, že lze kliknutím na jednu z veličin v legendě tuto veličinu dočasně smazat z grafu a tím si dočasně prohlédnout v detailnějším pohledu například veličinu, která má mnohem nižší hodnoty než druhá.

Co se týče vzhledu samotné HTML stránky, obsahuje kromě kreslicího plátna, kde je vykreslen graf, který vizte na obrázcích 4.3 a 4.4, lze zde najít povícero velmi důležitých součástí:

- pole pro výběr typu grafu realizovaný pomocí HTML5 tagu *select*,
- informace o datu a čase pořízení první a poslední hodnoty,
- pole pro filtraci a
- pochopitelně potvrzovací tlačítko na vytvoření grafu, které je při načtení stránky provázeno popiskem „Create Chart“ a po vytvoření grafu „Refresh Chart“.

Toto potvrzovací tlačítko je zde nikoliv z důvodu, že by bylo nemožné aktualizovat graf automaticky při jakékoli změně v reálném čase, ale protože je pro uživatele pohodlnější mít veškerý děj pod kontrolou a jelikož při změně například typu grafu by bylo nevhodné opakovaně (a při špatném používání i každou sekundu) znovu načítat graf. Toto by totiž znamenalo vyšší využití dat (což by například při užití mobilního hotspotu

či mobilního zařízení mohlo být i finančně náročné) a chybové načítání animace při obnovení grafu.



Obrázek 3.5 - Vzhled části stránky se zobrazením grafu

### 3.2.3.1 Filtrace dat

Zde se jedná o velmi důležité pole celé webové stránky, která zobrazuje graf.

V prvním návrhu aplikace byla tato činnost realizována pomocí Bootstrap Slideru (vizte kapitolu 2.1.2.2) a to sice takovým způsobem, že se do posuvníku načetlo pořadí poslední hodnoty a uživatel si poté mohl vybírat pořadová čísla, mezi kterými budou zobrazena data (například nastavil na posuvníku hodnoty 2 a 5 a zobrazil se mu tedy druhý až pátý záznam).

Tento způsob byl shledán jako nepříliš vyhovující pro reálné využití z důvodu neúplné komfortnosti a špatné ovladatelnosti (například uživatel ví, že pro něj důležitá část měření se udála mezi 1. lednem a 10. únorem, ale netuší, jaká pořadová čísla tyto záznamy mají).

Proto bylo přistoupeno k druhému způsobu možnosti filtrování dat, a sice za pomoci Bootstrap DateTime Pickeru (vizte kapitolu 2.1.2.1). Tento způsob je poměrně vhodný, nicméně i on má jisté nedostatky.

Je nutné použít dva „pickery“, aby bylo možné dosáhnout rozpětí od – do. Primární nastavení lokality pro tyto tagy je pro Spojené státy americké („en-US“) čili formát MM/DD/RRRR (kde M značí měsíc, D den, R rok za použití dvanáctihodinového formátu), což má určité výhody, které budou

zmíněny v následujících odstavcích, nicméně pro lidi žijící v České republice je tento formát matoucí (zejména pokud je den a měsíc stejný – například 4. dubna, 10. října apod.).

Proto byl využit formát, který je průnikem klasicky českého a „univerzálního“ formátu. Konkrétně se jedná o lokalizaci pro Spojené království s formátem DD/MM/RRRR s použitím 24hodinového formátu.

Bohužel se s tímto formátem (a ke stejnému problému by došlo i při použití klasické české lokalizace) se pojí značná nepříjemnost. Jelikož je poté potřeba výstup z „pickerů“, který je typu *string* (textový řetězec) užít jako datum pro další operace, nastává problém s převodem.

V JavaScriptu je zabudována funkce *new Date()* pro převod řetězce (*stringu*) do formátu data. Ta je primárně ve formátu *rok, měsíc, den (, hodina, minuta, sekunda, milisekunda)*, přičemž údaje v závorce jsou nepovinné. Dokonce si funkce poradí i s americkou lokalizací data, bohužel s britskou nikoliv.

Kvůli tomu je nutné údaj z „pickeru“ rozdělit pomocí trojího použití funkce *split()*, která v závorce obsahuje údaje o znaku (popřípadě písmena nebo čísla), podle kterého se řetězec rozděluje.

Bylo by vhodné zde poznamenat, že po nastavení data v alespoň jednom se nad „pickery“ objeví informace o tom jaká data byla vybrána a sice JavaScriptem vygenerovaným textem: „Data was measured from/to/between 1/1/2018 00:00 (and 12/2/2018 11:23)<sup>3</sup>“.

---

<sup>3</sup> Údaje v závorce jsou použity v případě, že jsou nastaveny data na obou „pickerech“ a tím pádem je jako předložka *mezi* (between)

## 4 Formuláře

Tato část aplikace má za úkol zjednodušit studentům sepisování naměřených hodnot během předmětů na Fakultě strojní ČVUT v Praze. Důvodem, proč k tomuto bylo přistoupeno je ten, že na některých předmětech je stále vyžadováno vypracování alespoň části ručně.

Tato část aplikace EasyMeasurement může právě vypracování protokolu studentovi velmi ulehčit, ale naopak je vyžadována větší součinnost vyučujícího.

Po vyučujícím je zde vyžadováno, aby jako přihlášený uživatel nastavil šablonu pro dané měření (společně s několika údaji, jako veličinami či jejich názvy) a vybral si, zda chce, aby studenti ukládali svá naměřená data do databáze.

Student, který nemusí být do aplikace přihlášen, pak již pouze vypíše své naměřené hodnoty a vybere si, jestli chce vygenerovat graf a případně z jakých veličin ho utvořit. Tuto šablonu je pak možné vyexportovat do formátu PDF.

### 4.1 Šablona

Samotná šablona je generována dle zadání vyučujícího, a sice vyplněním následujícího formuláře při tvorbě na stránce *New Template*.

Your name of template and its number of measurements:

Please, mind that this feature was designed to accept a maximum of 9 quantities!

Name of template:

Number of measurements:

Number of quantities:

Save data to DB: Yes ▼

Create

**Obrázek 4.1 - Formulář "New Template"**

Údaje o počtu měření a veličin jsou důležité z důvodu, aby bylo dosaženo naprosto správné funkce šablony při ukládání dat do databáze a exportu.

Kromě stránky, která generuje vlastní tabulku pro vytvoření dat obsahuje „podaplikace“ *EasyLaboratory* ještě webové stránky pro editaci vlastní šablony měření.

Samotná stránka pro vyplňování dat studentem nese název *Edit Data* a má následující vzhled:



**Name(s):**

**Measured data:**

Quantity 1: T1 [°C]	Quantity 2: T2 [°C]	Quantity 3: T3 [°C]
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

**Selected quantities to chart:**

X-axis:  Y-axis:

**Export data:**

Export with chart  Export without chart

*Obrázek 4.2 - Webová stránka pro vyplňování dat*

Tato stránka má několik částí, které jsou vždy pro něco důležité. První část je pro vyplnění jména, které slouží pro identifikaci naměřených dat v případě ukládání do databáze a zároveň pro následný export do PDF, kde je vyplněno i jméno autora.

## 4.2 Generování tabulky

Druhou částí je tabulka, která je vygenerovaná podle výše zmíněného principu. Je tvořena několika inputy, jejichž počet je vždy určen jako součin počtu měření a počtu veličin.

Tyto inputy jsou nastaveny jako číselné, vzhledem ke skutečnosti, že se jedná o obvyklý výstup ve strojařských aplikacích a že je logičtější

z nich generovat graf. Samozřejmě je možné zásahem vývojáře tuto skutečnost změnit.

Tato tabulka je ukončená tlačítkem „Save to DB“, čili „uložit do databáze“, které je aktivní nebo neaktivní podle zadání vyučujícího. Více o ukládání do databáze vizte v kapitole 4.3.

### 4.3 Ukládání dat do databáze

Jak je popsáno výše, pro uložení do databáze jsou zapotřebí první dvě části stránky, a to sice jméno autora a samotná načtená data. Do databáze se ukládá pomocí odeslání formuláře tlačítkem *Save to DB*.

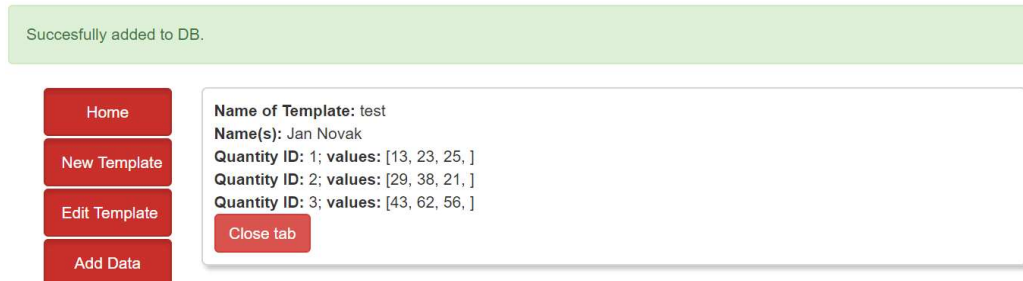
Nejprve je nutné každou hodnotu správně určit k dané veličině. Toto se děje skrytým polem inputu, kam je Pythonem automaticky zasláno ID dané veličiny.

Druhým polem v databázi je zde, taktéž poměrně důležité pole, a to sice číslo měření. Toto pole vzniká až v pythonovské funkci a má spíše informativní povahu.

Třetím polem je vlastní hodnota veličiny. Tato hodnota je posílána z výše zmíněných inputů typu číslo, kde je přesná identifikace určena „řetězcovým součtem“ čísla řádku a čísla sloupce, tudíž je každá hodnota určena naprosto spolehlivě.

Posledním polem v databázi je identifikační pole každé hodnoty. Skládá se z několika samostatných textových řetězců, které jsou spojeny pomocí dvojtečkového operátoru. Prvním řetězcem je vždy název měření. Druhým je čas nahrání do databáze s přesností na mikrosekundy. Jelikož je nahrávání do databáze uskutečněno for-cyklem, jsou hodnoty pro každé další měření odlišné. Posledním řetězcem je jméno měřícího, které bylo vyplněno v první části stránky *Edit Data*. Je možno ho vyplnit i s mezerou, jelikož tato nečiní v databázi žádné problémy.

Po odeslání dat k uložení do databáze vyskočí záložka s dvojnásobným potvrzením uložení do databáze, jednak zprávou o samotném uložení a dále zprávou o uložených datech. Celkem má tato stránka následující vzhled:



**Obrázek 4.3 - Vzhled potvrzení o uložení do database**

Po kliknutí na tlačítko *Close tab* je tato záložka opětovně zavřena.

## 4.4 Generování grafu

Třetí částí stránky pro vkládání dat je oblast tvorby grafu. Student si zde může vybrat, jakou veličinu (či počet měření) vložit na horizontální osu, stejně tak i vertikální osu.

Zde se opět lze setkat s nevýhodou typu horizontální osy. Jak je již řečeno výše, je možné při další aktualizaci knihovny *Chart.js* změnit toto nastavení.

Další možností pro změnu typu grafu je ho později změnit na bodový, nicméně v současné době nelze předpokládat, s jakými bude nakládáno výsledky, tudíž je nejspíše v současné době nejlepší ponechat klasický čárový graf.

## 4.5 Exportování

Poslední důležitou částí je oblast pro exportování dat. Student si zde může vybrat, zdali chce vygenerované PDF včetně grafu, či bez něj. Šablony jsou blíže popsány v kapitole 4.5.1.

Vygenerované dokumenty se od sebe liší jen a pouze přítomností oddílu grafu a jím samotným.

Jedním z poměrně důležitých faktorů v rámci generování dokumentu je úprava jednotek pro schůdnost aplikace. Jelikož v podmínkách Fakulty strojní ČVUT v Praze je možné, že naměřená data budou v „nestandardních jednotkách“, které neodpovídají kódování aplikace, musí být změněna. Konkrétně je toto aplikováno pro stupně a stupně Celsia, kdy znak „°“ je nahrazen anglickým výrazem „deg“.

Pro exportování je užíván formulář, který je téměř celý skrytý. Při každé změně je kopírován obsah viditelného formuláře primárně užívaného pro ukládání do databáze obohacený o viditelné pole výběru (HTML tag `input type="radio"`) a neviditelný „obraz“ HTML tagu `canvas`, kde je zobrazen graf, a sice převod samotného grafu do URI adresy v kódování `base64` ve formátu PNG a údaje o velikosti grafu. Tyto údaje jsou v pythonské funkci ve `views.py` dekodovány zpět do obrázku, který je neprodleně uložen do sekce `media` v serverové části aplikace pod jménem `chart.png`.

V druhé části je načtena systémová cesta k tomuto souboru a uložena jako proměnná do LaTeXové šablony.

Obrázek se při dalším použití funkce automaticky přepisuje.

#### 4.5.1 Generování PDF dokumentu

Jak je popsáno výše, uživatel si může vybrat ze dvou možností čili s grafem či bez grafu pomocí přepínače.

Realizace vlastního generování je prováděna knihovnou `Jinja2`, která umožňuje načtení LaTeXové šablony, která je obohacena o prvek proměnných psaných v Pythonu úplně stejně jako v případě HTML šablon čili následující podobou: `{{ název_proměnné }}`. Jediným nebezpečím tohoto doplňování je fakt, že samotné doplňování určitých částí LaTeXového

kódu je prováděno do složených závorek, proto musí být mezi proměnnou v Pythonu a složenými závorkami LaTeXového kódu umístěna mezera.

To obvykle nedělá v žádném případě problém, jelikož mezera může být snadno přehlédnuta, ne však v případě vkládání obrázku. Ten, jak je výše popsáno, je vkládán pomocí absolutní systémové cesty (z důvodu omezení chybovosti). Po načtení této cesty do proměnné a vnoření do LaTeXové šablony je zde mezera, která brání správnému přečtení, konkrétně: `(mezera)C:/.../chart.png(mezera)`, kde *C*: pochopitelně značí diskové uložení a v oblasti tří teček je vlastní cesta (zde pochopitelně není uvedena, jelikož zatím aplikace běží na zkušebním serveru v počítači a tudíž se u každého uživatele liší). Proto je nutné vložit do části pro vkládání grafu před a za proměnnou prázdný znak, který tvoří část mezi složenými závorkami LaTeXu a proměnnou. Prázdný znak je realizovaný LaTeXovým kódem `\empty`.

Kromě ne úplně ideálního vkládání cesty k obrázku je použit „klasický“ LaTeXový kód, který je obohacen o balíček pro kódování UTF-8 a grafický balíček, který je použit při vkládání obrázku.

Zde závěrečným, avšak ve vlastní aplikaci primárním krokem, je název vlastního dokumentu. Je zde použita sekvence `measureddata`, která je obohacena názvem šablony (respektive názvu měření), uvedeným za podtržítkem.

Vlastní vzhled exportovaného dokumentu je následující:

# Measured data - test

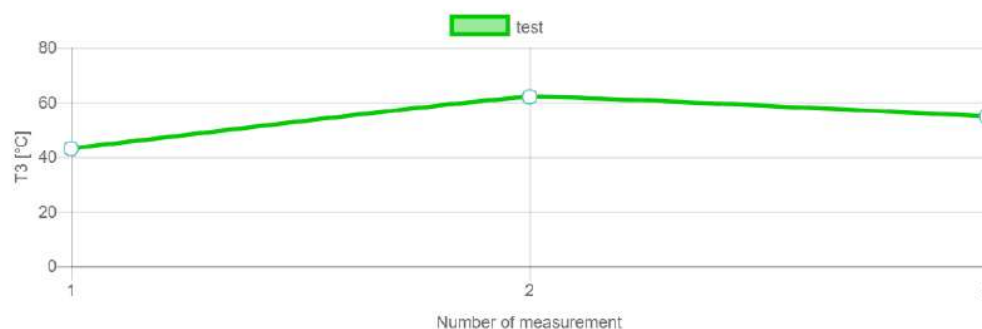
Jan Novak

May 20, 2018

## 1 Data

T1 [degC]	T2 [degC]	T3 [degC]
13	29	43
23	38	62
25	21	56

## 2 Chart



Obrázek 4.4 - Vzhled exportovaného dokumentu

Jediným rozdílem při využití dokumentu exportovaného bez grafu je neexistence druhé sekce – grafu.

## 5 Ukládání více dat

Z důvodu, že přidávání hodnot po jedné, přestože to lze dělat skriptem, který může pouštět robot, je relativně dost pomalé, a navíc při měřeních, která mají výstup kupříkladu deset hodnot během jedné sekundy, a tudíž by přidávání jednotlivých hodnot zbytečně zatěžovalo server, bylo přistoupeno k tomu, že jedním z úkolů této bakalářské práce je vytvořit prostředí pro ukládání vícero hodnot v rámci jednoho požadavku (requestu).

Po dohodě s vedoucím této práce bylo dospěno k názoru, že ideální formát na posílání strukturovaných dat je formát CSV (Comma Separated Value – hodnoty oddělené čárkami). Tento formát je univerzální a dokáže ho vygenerovat například i tabulkový editor Office Excel od společnosti Microsoft, či ho lze napsat i v jednoduchém textovém editoru (jako například Microsoft Notepad).

Formát CSV jde vytvořit z klasického listu v prostředí Excel pomocí příkazu *Uložit jako* a následně vybráním jednoho formátu CSV, který bude odpovídat správnému běhu aplikace (Microsoft Office Excel nabízí čtyři možnosti uložení CSV souboru).

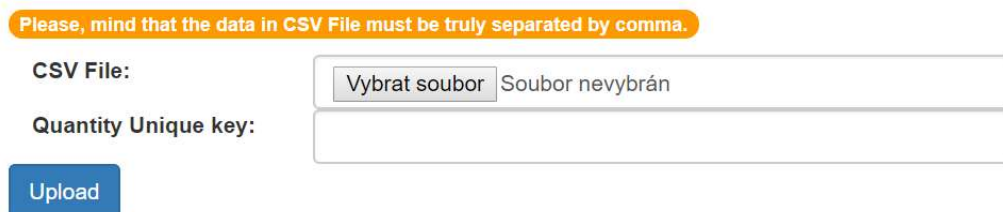
Je nutné upozornit uživatele, aby hodnoty opravdu oddělovaly čárkami, což výše zmíněný Excel v základním nastavení nedělá (odděluje hodnoty středníky). Samozřejmě, po dohodě s uživateli (kteří raději používají Excel), je možné nastavit ve *views.py* ve funkci, která zajišťuje tento upload, nastavit rozdělování středníky, přičemž tato změna by se týkala pouze jednoho řádku dané funkce ve *views* (jedná se o funkci *upload*), kde by se nastavila funkce *split()* na hodnotu argumentu *","*. Jak je ovšem řečeno výše, tato změna je možná po dohodě uživatelů s budoucím správcem aplikace.

Samotné nahrání dat je realizováno pomocí formuláře s metodou „POST“, který ve spojení Django s CSRF Tokenem vyžaduje přihlášení uživatele. Není ověřováno, že daná veličina, ke které jsou přidávány hodnoty, je součástí měření daného uživatele, a to z prostého důvodu.

Kromě samotného pole pro nahrávání CSV souboru je, jako druhé pole, požadováno napsání unikátního kódu dané veličiny. Tento kód je možné zobrazit pouze u uživatele, kterému „patří“ měření či přečtením údaje v databázi. Samozřejmě, že je možné, že v průběhu určitého technického měření, potřebuje nahrát data na server jiný uživatel. I z tohoto důvodu není ověřována příslušnost daného měření, a tudíž i veličiny, k uživateli.

Uspořádání daného souboru určeného k nahrání je velmi jednoduchý: *hodnota\_ veličiny (čárka) čas\_ pořízení\_ hodnoty*, přičemž je vhodné časy pořízení zaznamenávat alespoň na celé sekundy, ale je možné přijmout i detailnější rozlišení (například na milisekundy).

Samotný vzhled daného formuláře je následovný:



Please, mind that the data in CSV File must be truly separated by comma.

CSV File:  Soubor nevybrán

Quantity Unique key:

Obrázek 5.1 - Vzhled formuláře pro nahrání dat

Lze si zde povšimnout upozornění na oddělení dat čárkami, aby bylo možné co nejvíce předejít chybovému hlášení aplikace.

## 6 Změna vzhledu

Jakožto první operaci vývoje aplikace EasyMeasurement v rámci této bakalářské práce, ačkoliv to nebylo předmětem zadání, byla estetická změna vzhledu aplikace. Konkrétně se jedná o změnu vzhledu menu, jelikož původní menu bylo barevně poměrně „křiklavé“.

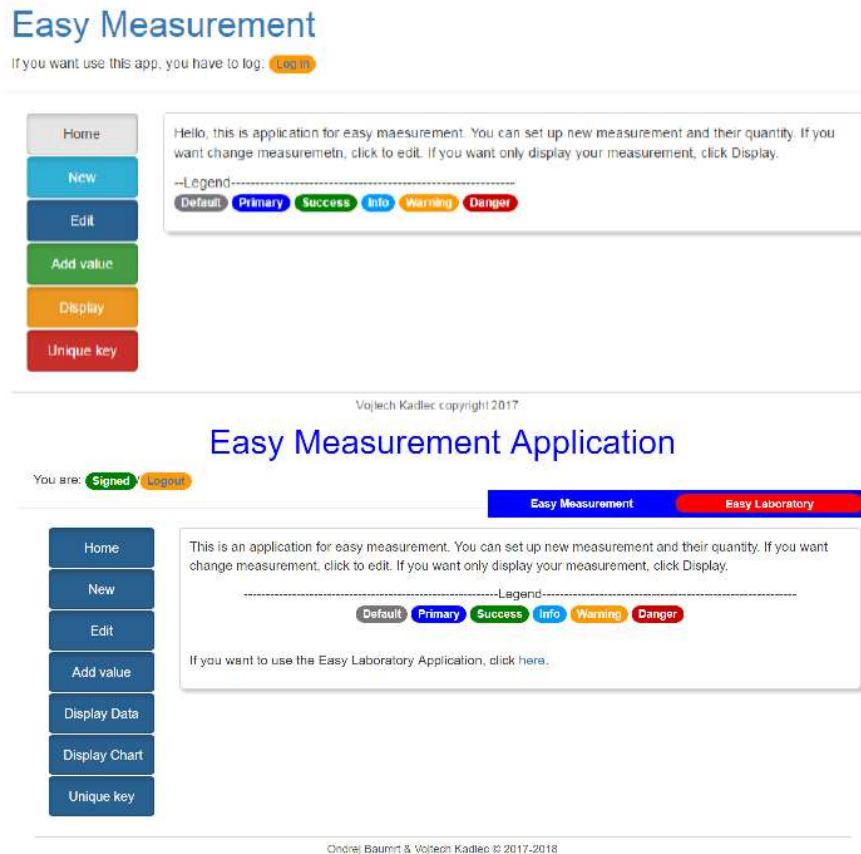
Také v rámci úkolů této bakalářské práce bylo do původního menu přidána položka *Display Chart* pro zobrazení grafů (více o zobrazování grafů vizte v páté kapitole. Následkem toho, čistě z přehlednosti, byla původní položka *Display*, která zobrazovala naměřená data, přejmenována na *Display Data* kvůli lepší orientaci budoucích uživatelů aplikace.

Další v úpravě vzhledu je zarovnání názvu a drobná korekce barvy. Poslední částí úpravy je přidání „podaplikace“ EasyLaboratory, o níž konkrétně je kapitola 6.1 této bakalářské práce.



Závěrečnou kosmetickou změnou je přidání jména autora této bakalářské práce do tiráže.

Na obrázku 6.1 lze vidět rozdíly před a po této kosmetické změně, kdy původní vzhled je v horní části obrázku a nový ve spodní části:



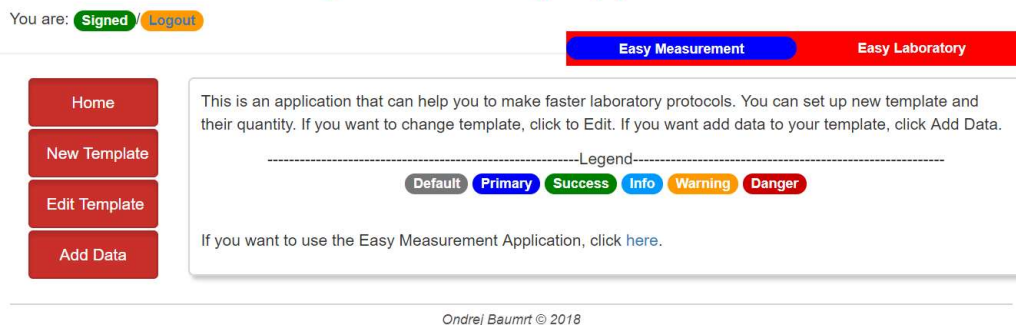
Obrázek 6.1 - Porovnání vzhledu původního vzhledu aplikace s novým

## 6.1 EasyLaboratory

Tato část aplikace byla primárně určena pro zapisování hodnot z formulářů, o kterých je více řečeno v páté kapitole. Jako „podaplikace“ byla vytvořena z důvodu potřeby jiného menu a také jiných databázových tabulek. Pro rozlišení této části, je na menu i titulek použita červená barva.

Vzhled EasyLaboratory vizte na obrázku 6.2:

# Easy Laboratory Application



Obrázek 6.2 - Vzhled základní obrazovky "podaplikace" EasyLaboratory

## 6.2 Přechod na Python 3

Částí, která byla nutná před vlastním řešením zadání této bakalářské práce, je převod aplikace do modernější verze Pythonu a sice do jeho třetího vydání. Do této kapitoly jsou také zařazeny chyby původní aplikace. Obvykle se jedná o chyby gramatické, nicméně se zde objevily i chyby faktické. Problémy kompatibility mezi druhou a třetí verzí Pythonu jsou blíže popsány v kapitole 1.2. Jsou stále vydávány nové verze Pythonu 3, proto je v této kapitole označen jako *Python 3.X*.

Úvodem je zde vhodné poznamenat, že pro funkčnost aplikace je nutné doinstalovat knihovnu *xlwt*, která je použita v původní aplikaci pro export dat. Tato knihovna je dostupná pomocí PyPI<sup>4</sup> [40].

Mezi chyby vyvstalé při převodu aplikace z Pythonu 2.7 patřilo například: špatná syntaxe příkazu *print()*, nutná úprava argumentů v určitých funkcích. Dalším, poměrně závažným, problémem při běhu aplikace v Pythonu 3.X, byla nutnost nahrazení `__str__()` v souboru *models*, který v Pythonu 2.7 funguje bez problémů, tagem `__repr__()`, který je univerzální pro všechny verze Pythonu a funguje tedy i v Pythonu 3.X. Za pomoci odstraněním těchto chyb patří velké poděkování vedoucímu práce.

<sup>4</sup> Python Package Index (PyPI) je uložisko softwaru pro programovací jazyk Python. [41]

Poté bylo nutné odstranit problém, že se data do databáze ukládala ve formátu *b'data'*, kde *data* značí poslaná data. Byl způsoben typem kódování utf-8. Byl odstraněn smazáním kódu `.encode('utf-8')`, avšak je možné, že s použitím Pythonu 2.7 fungoval kód bez tohoto problému.

Dále se zde vyskytovaly chyby pravopisné, které pochopitelně nemají vliv na funkci aplikace, nicméně jejich přítomnost je degradující pro práci po vzoru „forma kazí obsah“.

Bylo také nutné změnit funkci `views`, která se zabývá potvrzením uložení nového měření. Chybou zde byl fakt, že změnou `__str__` na `__repr__` aplikace zobrazuje celý řádek, a ne pouze jeho název. To způsobovalo, že se místo názvu objevil následující řetězec: „*Measurement object (x)*“, kde *x* značí ID číslo vlastního měření.

Poslední vážnější chybou zde bylo špatné pojmenování funkce v šabloně `Display Measurement`, která má zobrazovat naměřená data.

## Závěr

Cílem této bakalářské práce bylo implementovat a zprovoznit tři pokročilé funkce do aplikace EasyMeasurement, která byla vytvořena a obhájena jako bakalářská práce pana Vojtěcha Kadlece.

Prvním bodem bylo obohacení aplikace o možnost zobrazování dat grafem. Toto bylo implementováno za pomoci knihovny Chart.js, která, ačkoli zdaleka není dokonalá, se po rešerši v této oblasti jevila jako nejideálnější. Grafy jsou implementovány ve dvou verzích, jednak pro zobrazování jedné veličiny, poté jako srovnání vývoje dvou veličin (v aplikaci pojmenováno jako graf měření). Obě tyto verze jsou plně funkční a také obohaceny o možnost filtrování dat podle data pořízení. Velmi zajímavou možností je v obou verzích dočasně „odstranit“ jednu linii veličiny, a tudíž si detailněji prohlédnout zbylou. V práci již zmíněnou nevýhodou je typ horizontální osy, který je implementován jako „kategorie“. To znamená, že jsou všechny hodnoty na vodorovné ose stejně daleko od sebe.

Druhým bodem bylo implementování jednoduššího zaznamenávání dat spojená s exportováním těchto hodnot. Toto může velmi pomoci studentům se zaznamenáváním dat naměřených během laboratorních prací. Je zde možnost uložit hodnoty do databáze, což může pomoci vyučujícímu pro kontrolu těchto dat. Jak je již výše řečeno, je možnost tyto hodnoty exportovat do formátu PDF, který je z mého hlediska nejpoužitelnější pro přenos na různá zařízení. Výsledný dokument je velmi jednoduchý, kromě názvu a údajích o autorovi a datu pořízení je zde tabulka s naměřenými daty a dle volby uživatele i vygenerovaný graf, který může být tvořen jakoukoli naměřenou veličinou či pořadovým číslem měření.

Závěrečným bodem zadání bylo obohacení aplikace o možnost vložení většího množství dat v rámci jednoho požadavku. Opět jsem zde použil zřejmě nejlepší formát pro vkládání strukturovaných dat, a sice dokument s příponou CSV. Data jsou přiřazována k dané veličině pouze na ověření dle unikátního identifikačního řetězce dané veličiny.

Nad zadání bylo vhodné učinit několik dalších kroků. Prvním krokem bylo převedení verze Pythonu na nejnovější verzi 3.6. Poté, speciálně pro účel tvoření formulářů naměřených hodnot, byla vytvořena „podaplikace“ EasyLaboratory. V závěru byl již jen mírně upraven vzhled samotné aplikace.

Při tvorbě této práce jsem se setkal s několika obtížemi. Primární byla absolutně nedokonalá znalost programovacího jazyku Python ve spojení s frameworkem Django. V této věci mi byla nápomocna skutečnost, že jsem nemusel vytvářet kompletně novou aplikaci, ale mohl jsem rozšiřovat již stávající o, alespoň pro mne, zajímavé funkce, nicméně přechod aplikace z původního Pythonu ve druhé verzi na třetí vydání bylo mnohdy přitěžující skutečností. Dalším nápomocným faktorem byla znalost programovacího jazyku Java, který má sice odlišnou syntaxi, ale způsob implementace problému je téměř totožný.

Další obtíže mnohdy činila kombinace čtyř programovacích jazyků, které jsou zde použity, konkrétně se jedná o samotný Python obohacený o HTML, JavaScript a LaTeX. Není vždy jednoduché zkombinovat tyto jazyky, aby docílily společného cíle, ale mnohdy je to i nápomocno (kupříkladu kombinace Pythonu s JavaScriptem je velmi ulehčující).

Dle mého názoru je aplikace zcela funkční a může být již použita k plnému užívání v rámci určitých předmětů na Fakultě strojní Českého vysokého učení technického v Praze.

Tato aplikace může být obsahem tvoření dalších prací, jelikož je zde stále co vylepšovat. Některý z dalších studentů může upravit styl vytváření grafů (hlavně změnu typu horizontální osy, která je přislíbena v jedné z příštích aktualizacích knihovny Chart.js). Dále je možné kupříkladu vytvořit sofistikovaný způsob zadávání dat studenty pomocí unikátně přiděleného odkazu, vytvořit způsob, kdy by se data posílaly vysokou frekvencí (jako stream videa), nebo vytvořit měřicí kartu pro mikropočítač nebo program, která by sama automaticky posílala data přímo do aplikace.

## Seznam použité literatury

- [0-1] KADLEC, Vojtěch. *Serverová aplikace pro bezpečný sběr dat*. Praha, 2017. Bakalářská práce. Fakulta strojní ČVUT v Praze. Vedoucí práce Ing. Vladimír Hlaváč.
- [0-2] BAUMRT, Ondřej. *Webová aplikace pro ukládání výsledků laboratorních prací během výuky ve frameworku Django*. Praha, 2017. Semestrální projekt. Fakulta strojní ČVUT v Praze. Vedoucí práce Ing. Matouš Cejnek.
- [0-3] BAUMRT, Ondřej. *Webová aplikace pro ukládání výsledků laboratorních prací během výuky ve frameworku Django*. Praha, 2018. Semestrální projekt. Fakulta strojní ČVUT v Praze. Vedoucí práce Ing. Matouš Cejnek.
- [1] Python. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikipedia Foundation, 2001-. [cit. 2018-04-16]. Dostupné z: [https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- [2] ELKNER, Jeffrey, Allen B. DOWNEY, Chris MEYERS a Jaroslav KUBIÁŠ. *How to think like a computer scientist – Learning with Python* [online]. Second edition. Sebastopol (CA): O'Reilly Media, 2015 [cit. 2018-04-16]. ISBN 978-0971677500. Dostupné z: <http://howto.py.cz/index.htm>
- [3] *Co je Django?* [online]. Django Girls Tutorial, 2018 [cit. 2018-04-30]. Dostupné z: <https://tutorial.djangogirls.org/cs/django/>
- [4] *The web framework for perfectionists with deadlines* [online]. Django Software Foundation, 2018 [cit. 2018-04-30]. Dostupné z: <https://django-project.com/>
- [5] BENDORAITIS, Aidas. *Web Development with Django Cookbook*. Second Edition. Birmingham: Packt Publishing, 2016. ISBN 978-1-78588-677-5.
- [6] Django (web framework). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-04-16]. Dostupné z: [https://en.wikipedia.org/wiki/Django\\_\(web\\_framework\)](https://en.wikipedia.org/wiki/Django_(web_framework))
- [7] BOGDANOV, Vladimir. Top 10 sites built with Django Framework. *LinkedIn.com* [online]. 2015 [cit. 2018-04-16]. Dostupné z: <https://www.linkedin.com/pulse/top-10-sites-built-django-framework-vladimir-bogdanov>
- [8] Django. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-04-16]. Dostupné z: <https://cs.wikipedia.org/wiki/Django>
- [9] *Databáze* [online]. ABC.cz, 2017 [cit. 2018-04-30]. Dostupné z: <http://slovník-cizich-slov.abz.cz/web.php/slovo/databaze>
- [10] *Databáze* [online]. Misha, 2010 [cit. 2018-04-30]. Dostupné z: <http://www.databaze.chytrak.cz/modely.htm>
- [11] BÍLA, Jiří. *Database and Knowledge-base Systems*. (přednáška) Praha: Fakulta strojní ČVUT v Praze, 2012
- [12] *The key differences between SQL and NoSQL DBs* [online]. Monitis, 2017 [cit. 2018-05-10]. Dostupné z: <http://www.monitis.com/blog/cc-in-review-the-key-differences-between-sql-and-nosql-dbs/>
- [13] SQL. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-10]. Dostupné z: <https://en.wikipedia.org/wiki/SQL>

- [14] PETŘÍK, Tomáš. *Volba databázového systému na základě požadavků projektu IS/ICT* [online]. Praha, 2008 [cit. 2018-04-30]. Dostupné z: <https://vskp.vse.cz/id/1147179>. Bakalářská práce. Fakulta informatiky a statistiky, Vysoká škola ekonomická v Praze. Vedoucí práce Ing. Dušan Chlapek.
- [15] *Oracle Technology Global Price List* [online]. Oracle, 2017 [cit. 2018-05-05]. Dostupné z: <http://www.oracle.com/us/corporate/pricing/technology-price-list-070617.pdf>
- [16] RYCHLÝ, Marek. *NoSQL databáze* (přednáška) Brno: Fakulta informačních technologií VUT v Brně. 15. října 2013. Dostupné z: <http://www.fit.vutbr.cz/~rychly/public/docs/slides-nosql-databases/slides-nosql-databases.print.pdf>
- [17] MapReduce. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-05]. Dostupné z: <https://cs.wikipedia.org/wiki/MapReduce>
- [18] *SQL vs. NoSQL Databases: What's the Difference?* [online]. UpWork.com, 2017 [cit. 2018-05-05]. Dostupné z: <https://www.upwork.com/hiring/data/sql-vs-nosql-databases-whats-the-difference/>
- [19] VAŇKOVÁ, Jana. *Sítě pod drobnohledem: Internetové protokoly* [online]. RVP, 2011 [cit. 2018-05-06]. Dostupné z: <http://clanky.rvp.cz/clanek/c/G/13905/site-pod-drobnohledem-internetove-protokoly.html/>
- [20] *Protokol HTTPS* [online]. SSSL.cz, 2017 [cit. 2018-05-06]. Dostupné z: <https://www.sssl.cz/https.html>
- [21] KRAČMÁŘ, Petr. *Jak nahradit FTP pomocí SFTP a zamknout uživatele* [online]. Root.cz, 2010 [cit. 2018-04-16]. Dostupné z: <https://www.root.cz/clanky/jak-nahradit-ftp-pomoci-sftp-a-zamknout-uzivatele/>
- [22] PILGRIM, Mark. *Ponořme se do Python(u) 3*. Praha: CZ.NIC, 2010. ISBN 978-80-904248-2-1.
- [23] *What's New In Python 3.0* [online]. Python Software Foundation, 2009 [cit. 2018-05-06]. Dostupné z: <https://docs.python.org/3.0/whatsnew/3.0.html>
- [24] *JavaScript Tutorial* [online]. W3Schools, 2018 [cit. 2018-05-07]. Dostupné z: <https://www.w3schools.com/js/default.asp>
- [25] JQuery. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-07]. Dostupné z: <https://cs.wikipedia.org/wiki/JQuery>
- [26] *JQuery* [online]. The jQuery Foundation, 2018 [cit. 2018-05-07]. Dostupné z: <https://jquery.com>
- [27] *1. díl – Úvod do jQuery* [online]. IT Network, 2018 [cit. 2018-05-07]. Dostupné z: <https://www.itnetwork.cz/javascript/jquery-zaklady/javascript-tutorial-funkcionalni-programovani-a-jquery-webova-kalkulacka>
- [28] Bootstrap. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-07]. Dostupné z: <https://cs.wikipedia.org/wiki/Bootstrap>

- [29] *Bootstrap – The most popular HTML, CSS and JS library in the world* [online]. MIT, 2018 [cit. 2018-05-07]. Dostupné z: <https://getbootstrap.com/>
- [30] *Bootstrap 3 Datepicker* [online]. GitHub, 2018 [cit. 2018-04-19]. Dostupné z: <https://eonasdan.github.io/bootstrap-datetimepicker/>
- [31] *Slider for Bootstrap Examples Page* [online]. Seiyria.com, 2018 [cit. 2018-04-19]. Dostupné z: <http://seiyria.com/bootstrap-slider/>
- [32] URBAN, Jakub a Jan PIPEK. *Matplotlib – grafy v Pythonu* [online]. Pythonic.eu, 2017 [cit. 2018-05-07]. Dostupné z: <http://pythonic.eu/fjfi/posts/matplotlib.html>
- [33] VIKTORIN, Petr a Miro HRONČOK. *NumPy* [online]. Nauč se Python!, 2017 [cit. 2018-05-07]. Dostupné z: <http://nauce.python.cz/lessons/intro/numpy/>
- [34] RAHMAN, Syed Fazle. *16 JavaScript Libraries for Creating Beautiful Charts* [online]. Sitepoint Pty., 2015 [cit. 2018-05-07]. Dostupné z: <https://www.sitepoint.com/15-best-javascript-charting-libraries/>
- [35] RAJ, Jay. *16 JavaScript Libraries for Creating Beautiful Charts* [online]. Sitepoint Pty., 2015 [cit. 2018-05-07]. Dostupné z: <https://www.sitepoint.com/creating-simple-line-bar-charts-using-d3-js/>
- [36] *Charts* [online]. Chart.js, 2016 [cit. 2018-05-07]. Dostupné z: <http://www.chartjs.org/docs/latest/charts/>
- [37] AHLIN, Tobias. *10 Chart.js example charts to get you started* [online]. Tobias Ahlin Blog, 2017 [cit. 2018-05-07]. Dostupné z: <http://tobiasahlin.com/blog/chartjs-charts-to-get-you-started/>
- [38] SINGHAL, Vaibhav. *20 best JavaScript charting libraries* [online]. The Next Web B.V., 2018 [cit. 2018-05-07]. Dostupné z: <https://thenextweb.com/dd/2015/06/12/20-best-javascript-chart-libraries/>
- [39] *Charts* [online]. Google Developers, 2018 [cit. 2018-05-07]. Dostupné z: <https://developers.google.com/chart/>
- [40] *Xlwt* [online]. Python Software Foundation, 2018 [cit. 2018-04-19]. Dostupné z: <https://pypi.org/project/xlwt/>
- [41] *PyPI – the Python Package Index* [online]. Python Software Foundation, 2018 [cit. 2018-04-19]. Dostupné z: <https://pypi.org/>
- [42] *ReportLab – Complete Hosted Solutions – ReportLab.com* [online]. ReportLab, 2018 [cit. 2018-05-13]. Dostupné z: <https://www.reportlab.com/>
- [43] *HTML Template to PDF in Django* [online]. Coding for Entrepreneurs, 2018 [cit. 2018-05-13]. Dostupné z: <https://www.codingforentrepreneurs.com/blog/html-template-to-pdf-in-django/>
- [44] BSD Licence. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-13]. Dostupné z: [https://cs.wikipedia.org/wiki/BSD\\_licence](https://cs.wikipedia.org/wiki/BSD_licence)
- [45] NARETLA, Swetha. *Using Python WeasyPrint Generate HTML To PDF In Django* [online]. MicroPyramid, 2017 [cit. 2018-05-13]. Dostupné z: <https://micropyramid.com/blog/generate-pdf-files-from-html-in-django-using-weasyprint/>



- [46] *WeasyPrint* [online]. WeasyPrint, 2018 [cit. 2018-05-13]. Dostupné z: <https://weasyprint.org/>
- [47] *Jinja2* [online]. Pocoo, 2014 [cit. 2018-05-13]. Dostupné z: <http://jinja.pocoo.org/>
- [48] Jinja2. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-13]. Dostupné z: <https://cs.wikipedia.org/wiki/Jinja2>

## Seznam použitých obrázků

Obrázek 2.1 - Ukázka použití Bootstrap 3 DateTime Pickeru.....	18
Obrázek 2.2 - Ukázka kódu k vytvoření grafu v Pythonu .....	20
Obrázek 2.3 - Ukázka grafu vytvořeného pomocí knihovny Matplotlib v Pythonu .....	20
Obrázek 2.4 - 1. část ukázky kódu pro vytvoření grafu pomocí D3.js ..	21
Obrázek 2.5 - 2. část ukázky kódu pro vytvoření grafu pomocí D3.js .	22
Obrázek 2.6 - 3. část ukázky kódu pro vytvoření grafu pomocí D3.js.	23
Obrázek 2.7 - 4. část ukázky kódu pro vytvoření grafu pomocí D3.js .	23
Obrázek 2.8 - 5. část ukázky kódu pro vytvoření grafu pomocí D3.js.	24
Obrázek 2.9 -Ukázka grafu vytvořeného pomocí knihovny D3.js.....	24
Obrázek 2.10 - 1. část ukázky kódu pro vytvoření grafu pomocí Chart.js .....	25
Obrázek 2.11 - 2. část ukázky kódu pro vytvoření grafu pomocí Chart.js .....	25
Obrázek 2.12 - 3. část ukázky kódu pro vytvoření grafu pomocí Chart.js.....	26
Obrázek 2.13 - Ukázka grafu vytvořeného pomocí knihovny Chart.js .	27
Obrázek 2.14 - 1. část ukázky kódu pro vytvoření grafu pomocí Google Charts .....	28
Obrázek 2.15 - 2. část ukázky kódu pro vytvoření grafu pomocí Google Charts .....	28
Obrázek 2.16 - Ukázka grafu vytvořeného pomocí knihovny Google Charts .....	29
Obrázek 3.1 - Zobrazení šablony pro výběr typu grafu .....	34
Obrázek 3.2 - Výstup šablony chooseMeasurement.html .....	35
Obrázek 3.3 - Příklad typů grafu pro graf jedné veličiny .....	36
Obrázek 3.4 - Příklad typů grafu pro graf měření .....	37
Obrázek 3.5 - Vzhled části stránky se zobrazením grafu.....	38
Obrázek 4.1 - Formulář "New Template" .....	41
Obrázek 4.2 - Webová stránka pro vyplňování dat.....	42
Obrázek 4.3 - Vzhled potvrzení o uložení do database.....	44
Obrázek 4.4 - Vzhled exportovaného dokumentu .....	47
Obrázek 5.1 - Vzhled formuláře pro nahrání dat.....	49
Obrázek 6.1 - Porovnání vzhledu původního vzhledu aplikace s novým.....	50
Obrázek 6.2 - Vzhled základní obrazovky "podaplikace" EasyLaboratory .....	51

## **Přílohy**

Jako příloha této práce je samotná aplikace, která je k této práci připojena v elektronické podobě.