

**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

**FAKULTA
STROJNÍ**



**BAKALÁŘSKÁ
PRÁCE**

2017

**VOJTĚCH
KADLEC**

České vysoké učení technické v Praze

Fakulta strojní

Ústav přístrojové a řídicí techniky

Obor: Informační a automatizační technika

Serverová aplikace pro bezpečný sběr dat

BAKALÁŘSKÁ PRÁCE

Vypracoval: Vojtěch Kadlec

Vedoucí práce: Ing. Vladimír Hlaváč

Konzultant: Ing. Matouš Cejnek

Rok: 2017

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kadlec** Jméno: **Vojtěch** Osobní číslo: **420553**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav přístrojové a řídicí techniky**
Studijní program: **Strojírenství**
Studijní obor: **Informační a automatizační technika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Serverová aplikace pro bezpečný sběr dat

Název bakalářské práce anglicky:

Server application for save data collection

Pokyny pro vypracování:

1. Navrhnout univerzální model SQL databáze pro ukládání dat z různých experimentů ? každý uživatel může vytvořit více experimentů, každý obsahující více sledovaných hodnot.
2. Vytvořit a zdokumentovat API pro ukládání dat do databáze přes HTTP ? uživatel by měl být schopný přes toto API posílat do databáze automatizovaně výsledky pomocí libovolných nástrojů.
3. Vytvořit webové rozhraní pro zobrazení, úpravy a export získaných dat. Zabezpečení tohoto přístupu by mělo odpovídat současným požadavkům na bezpečnost.

Seznam doporučené literatury:

Gourley, David, and Brian Totty. HTTP: the definitive guide. " O'Reilly Media, Inc.", 2002.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Vladimír Hlaváč, U12110.3

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


Ing. Matouš Cejnek, U12110.3

Datum zadání bakalářské práce: **19.04.2017**

Termín odevzdání bakalářské práce: **16.06.2017**

Platnost zadání bakalářské práce: _____


Podpis vedoucí(ho) práce

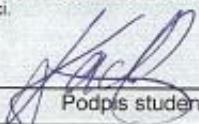

Podpis vedoucí(ho) ústavu/katedry


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.

19. 4 2017
Datum převzetí zadání


Podpis studenta

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literatura, projekty, SW atd.) uvedené v této práci.

Ve Veselí nad Lužnicí dne 8.6.2017

Vojtěch Kadlec

Poděkování

Děkuji Ing. Matouši Cejnkovi za jeho konzultace při tvorbě této bakalářské práce, za jeho podněty a připomínky, které ji obohatily a za jeho pomoc při tvorbě webové aplikace.

Dále jak je dobrým zvykem, děkuji své rodině a přátelům, kteří mi poskytli nezbytnou podporu během celého studia.

Vojtěch Kadlec

Název práce:

Serverová aplikace pro bezpečný sběr dat

Autor: Vojtěch Kadlec

Obor: Informační a automatizační technika

Druh práce: Bakalářská práce

Vedoucí práce: Ing. Vladimír Hlaváč
Ústav přístrojové a řídicí techniky
České vysoké učení technické v Praze

Konzultant: Ing. Matouš Cejnek

Abstrakt: Tato bakalářská práce je zaměřená na seznámení čtenáře s webovým frameworkem Django. Zároveň v ní je zahrnuta praktická ukázka, z které je zřejmé, jak framework pracuje a komunikuje s okolním virtuálním světem. Aplikace EasyMeasurement by měla usnadnit uživateli jednoduchý a bezpečný sběr dat, přičemž dokáže komunikovat s externím měřicím zařízením za účelem dlouhodobého nebo intenzivního sběru.

Klíčová slova: Django, framework, Python, Html, Css

Title:

Server application for save data collection

Author: Vojtěch Kadlec

Abstract: This bachelor's thesis introduces basic principles of using the web Framework Django. In the practical part of this work there is included an example of the framework function, and its communication with virtual world. The EasyMeasurement application should help the user to easily and safely collect the data from the external measuring device in order to obtain a large amount of data from long time periods.

Key words: Django, framework, Python, Html, Css

Obsah

1. Úvod.....	1
2. Django.....	3
2.1 Historie Djanga	4
2.2 Proč Django	5
2.2.1 Objektový přístup k databázím	5
2.3 Bezpečnost v Django.....	6
2.3.1 Cross-site scripting (XSS).....	6
2.3.2 Cross site request forgery (CSRF).....	7
2.3.3 SQL injection ochrana	8
2.3.4 Clickjacking ochrana	8
2.4 Aplikace Django, databáze a server.....	9
2.4.1 Jak instalovat.....	9
2.4.2 Databáze.....	10
2.4.3 Server	10
2.5 Struktura Djanga.....	11
2.5.1 Projekt.....	11
2.5.2 Aplikace.....	12
3. Přidružené nástroje a jazyky.....	15
3.1 HTTP: Internetový mediální kurýr.....	15
3.2 Python	16
3.2.1 Proč Python	17
3.3 MySQL	17
3.4 CSS.....	18
3.5 HTML.....	19
3.6 Bootstrap.....	22
3.7 Sublime Text 2	23
4. Webová aplikace EasyMeasurement	24
4.1 Popis projektu.....	25
4.1.1 Prostředí aplikace	26

4.1.2 Šablona – base.html	27
4.2 Databáze.....	28
4.3 Databázové modely.....	29
4.4 Funkce definované pro aplikaci	30
4.5 Sběr dat z externího zařízení	32
4.5.1 Formulace http requestu.....	33
4.6 Budoucnost aplikace	34
5. Závěr.....	35
Seznam použitých zdrojů.....	36
Seznam použitých zkratek.....	398
Seznam obrázků.....	39
Seznam tabulek.....	39
Ukázky vzhledu aplikace	40

1. Úvod

V dnešní době je takřka nemožné si představit, že by svět fungoval bez internetu. Jedná se o naprosto nepostradatelnou součást našeho světa. Díky této celosvětové počítačové síti, můžou mezi sebou komunikovat jak lidé, tak i různé stroje. Budeme-li se bavit o nás, jako o lidech, pak asi nejvíce používanou formou naší komunikace přes internet bude systém World Wide Web, známý spíše pod zkratkou „web“. Bylo by ovšem chybou považovat „web“ a internet za jednu a tatáž věc, jak je zvykem mnoha lidí. Internet je globální systém vzájemně propojující počítačové sítě na celém světě, zatím co World Wide Web je aplikace, která na internetu funguje. Každý z nás si může takový „web“ vytvořit, kde může publikovat jakékoliv informace.

Zpočátku byly „weby“ tvořené jako statické prezentace určitých informací, ať už nějaké filmy či pro osobní potřebu. Postupem času se z webu stal nástroj pro uživatele ke sdílení informací. To způsobilo vznik mnoha nástrojů k usnadnění tvorby webových aplikací a jedním z nich je i právě framework Django.

Ukázku, na které chci demonstrovat funkce Djanga a dalších open-source programů, jsem nazval EasyMeasurement. Aplikace by měla usnadnit uživateli jednoduché zpracování naměřených hodnot. Funkcemi této aplikace bude vytváření jednotlivých měření a k nim přidělování určitých veličin a hodnot. Vkládání hodnot je možné přímo ve webovém rozhraní, nebo z jiného externího zařízení. Z toho vyplývá, že je možné dlouhodobé měření v rámci dnů, či měsíců, v čemž shledávám podstatnou výhodu této aplikace. Všechna data jsou ukládána do databází. Následně je možné měření a veličiny

upravovat, zobrazovat nebo naměřené hodnoty exportovat pro jejich další zpracování.

Pro využívání této aplikace bude nutné přihlášení, ověření uživatele, kterému budou přihlašovací údaje přiděleny administrátorem aplikace. Samotná registrace pak není možná. Stejně tak i data posílaná z externího zařízení musí nejprve projít ověřením pro daného uživatele.

Framework Django jsem zvolil z důvodu jeho flexibility při tvorbě takové aplikace a pro usnadnění práce programátora. Navíc je to světově používaný framework, který se stále vyvíjí a je v dnešním počítačovém světě bezpečný.

2. Django

Následující kapitola čerpá především z dvou webových zdrojů, které se přímo zabývají samotným frameworkem Django. Prvním z nich je „djangoproject.cz“ [1] a druhým je „zdroják.cz“ [2].

Pokud jste již někdy programovali nějakou aplikaci, konkrétně nějaký „web“, určitě jste se dostali do fáze, kdy jste museli určitou část kódu opakovaně psát, spíše kopírovat, protože se musela vyskytovat v každé stránce, kterou aplikace obsahovala. To by ovšem ještě nebyl takový problém, až do doby, kdy jste potřebovali v této části provést nějakou změnu. V tuto chvíli se takový systém programování projevuje jako velice neefektivní, nešikovný. V důsledku takových a podobných situací začali vznikat frameworky. Jedním z nich je i Django, které se pro řešení takové situace hodí.

Django bylo vyvíjeno už od roku 2003 jako součást redakčních systémů kansaského deníku Lawrence Journal-World. O dva roky později byl zdrojový kód otevřen a poskytnut široké veřejnosti pod liberální BSD¹ licenci. Co se týče názvu, je odvozen od uměleckého jména jazzového kytaristy Djanga Reinhardta, jenž byl označován za průkopníka žánru. Framework Django se snaží, podobně jako Reinhardt, o inovaci a perfekcionismus. [2]

Django je vyspělý webový framework napsaný v jazyce Python, který podporuje rychlý vývoj a čisté, pragmatické konstrukce. Django bylo vyvinuto před několika lety v rychle

¹ **BSD** (Berkeley Software Distribution) je licence pro svobodný software. Umožňuje volné šíření licencovaného obsahu, přičemž vyžaduje pouze uvedení autora a informace o licenci, spolu s upozorněním na zřeknutí se odpovědnosti za dílo.

se měnícím novinářském prostředí. Mělo pomoci zvládat dva velké úkoly: intenzivní plnění termínů v redakci a dodržovat přísné požadavky vývojářů — autorů frameworku. [1]

Jednou velkou předností Djanga je administrační rozhraní, které je automaticky generováno. Samotné administrační rozhraní pak umožňuje správu dat v databázích nebo úpravu webových stránek. Optimalizovaný a bezpečný přístup k datům v databázích je samozřejmostí. Po mém krátkém seznámení s Djangem můžu konstatovat, že se jedná o propracovaný šablonovací systém, díky kterému nemusím opakovaně psát stejný kód.

2.1 Historie Djanga

Django bylo vyvinuto Web development týmem kansaského deníku Lawrence v USA. Bylo to na podzim v roce 2003, kdy weboví programátoři v Lawrence Journal-world Newspapers, Adrian Holovaty a Simon Willison spojili jednotlivé webové aplikace a následně je zobecnili tak, že vznikla první verze frameworku Django. Avšak před tím než k tomuto spojení došlo se Adrian a Simon zabývali tím, jak ze statických stránek udělat nástroj, díky kterému bude možné provádět rychlé změny v závislosti na nových událostech, které se ve světě odehrávají. Proto začali vyvíjet jednotlivé aplikace pro specifické weby a když je pak spojili, vznikl onen framework.

V následujících letech se k vývojovému týmu připojil Jacob Kaplan-Moss. Pod dvou letech vývojáři rozhodli, že jejich framework je již použitelný pro veřejnost a tak se rozhodli ho vydat jako open-source pod licencí BSD. Jak již bylo zmíněno, jméno bylo odvozeno od uměleckého jména jazzového kytaristy Djanga Reinhardta, jenž byl označován za průkopníka žánru. Framework Django se snaží, podobně jako Reinhardt, o inovaci a perfekcionismus.

Dnes na frameworku pracuje tisíce lidí, ať už samotní vývojáři nebo uživatelé, kteří ho používají pro své vlastní dynamické webové aplikace. Proto podoba frameworku jak ji známe dnes, již není pouze práce zmíněných prvních vývojářů, ale pochází od spolupráce komunity, která díky Djangu vznikla. [3]

2.2 Proč Django

Jak již bylo zmíněno, Django si zakládá na programovacím principu DRY², díky kterému se programátoři mohou vyvarovat opakování kódu. Protože každý projekt, který je v Django vytvořen má stejnou základní strukturu, je pro ostatní programátory jednoduché se vyznat v cizích projektech. To se například hodí i v situaci, kdy si nevíte jako autor rady s vlastním problémem. Při dotazech na ostatní vývojáře to pak umožňuje snadnější komunikaci.

Dalšími výhodami jsou:

- Architektonický vzor MTV³
- Automaticky generované administrační rozhraní
- Výkonný šablonovací systém
- Podpora kešování
- Generování formulářů z databázového modelu, automatická validace
- Odlehčený webserver pro vývoj
- Škálovatelná architektura
- Komentářový systém
- Generování RSS
- Autentizace uživatelů

2.2.1 Objektový přístup k databázím

Object-Relation-Mapping je přístup k databázím, díky kterému pracujeme pouze jenom s objekty a funkcemi Pythonu. ORM implementované v Django transformuje objekty na dotazy a zároveň se snaží o jejich optimalizaci. Je nám ovšem ponechána možnost psát i čisté SQL. To můžeme uvítat v případě potřeby psát složitější příkazy.

² **DRY** (Do not Repeat Yourself) – „Pragmatický princip DRY nám říká, že každá dílčí znalost musí mít v systému jedinou, jednoznačnou, směrodatnou reprezentaci. Znamená to, že se nás Django snaží přimět k takovému rozvržení práce, v němž se zamezí rutinnímu opakování kódu. Bohatě se využívá vlastností objektově orientovaného programování a automatického generování kódu.“ [2]

³ **MTV** (Model-Template-View). „Zjednodušeně řečeno máme databázový model, z nějž získává data pohled (view), který je posílá (X)HTML šabloně. Tak je oddělena definice dat, jejich zpracování a prezentace.“ [2]

2.3 Bezpečnost v Django

Používáním Django šablon chráníme aplikaci proti většině útokům, které jsou na ni mířeny. Je však důležité porozumět tomu, kdy je taková ochrana pro aplikaci přínosem a naopak, kdy aplikaci zase limituje.

2.3.1 Cross-site scripting (XSS)

„Cross-site scripting (XSS) je metoda narušení WWW stránek využitím bezpečnostních chyb ve skriptech (především neošetřené vstupy). Útočník díky těmto chybám v zabezpečení webové aplikace dokáže do stránek podstrčit svůj vlastní javascriptový kód, což může využít buď pouze k poškození vzhledu stránky, jejímu znefunkčnění anebo dokonce k získávání citlivých údajů návštěvníků stránek, obcházení bezpečnostních prvků aplikace a phishingu⁴.“ [4]

XSS útoky můžeme rozdělit do tří typů:

- **Dom based** nebo **lokální**. Taková metoda je využívána při napadení statických stránek a jedná se o neošetřené přenesení proměnné z URL⁵ odkazu do Javascriptu⁶.

⁴ *„Phishing (někdy převáděno do češtiny jako rybaření) je podvodná technika používaná na Internetu k získávání citlivých údajů (hesla, čísla kreditních karet apod.) v elektronické komunikaci. K nalákání důvěřivé veřejnosti komunikace předstírá, že pochází z populárních sociálních sítí, aukčních webů, on-line platebních portálů, úřadů státní správy nebo od IT administrátorů.“ [14]*

⁵ **URL (Universal Resource Locator)** – *„Standardizovaný formát lokalizace zdrojů na internetu. Nejznámější a nejrozšířenější typ URI, nemá ovšem funkci trvalého identifikátoru. URL slouží pro různé typy služeb internetu (HTTP, FTP, Telnet, Gopher atd.). Specifikuje jméno a typ zdroje, hostitelský počítač, adresář, kde lze zdroj nalézt, a přenosový protokol potřebný pro použití zdroje. Tentýž zdroj může být prezentován na internetu na více adresách a přístupný pod různými protokoly.“ [17]*

⁶ *„JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape. Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, často vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků.“ [15]*

- **Non-persistent** nebo **reflected**. Je založen na úpravě části URL, která se interpretuje do webové stránky jako její součást. Například metodou GET nebo POST, které popíšu v dalších částí této práce. (r = request.POST["něco"])
- **Persistent, stored** nebo **second-order**. Dochází k přímé a trvalé modifikaci stránky. Například můžu vložit do komentáře nějakého webu vlastní Javascript, který se spolu s komentářem uloží do databáze. Následně, když jiní uživatelé si tento komentář zobrazí, provede se zároveň i script, který jsem použil.

K tomu, aby byla naše aplikace psaná v Django vzdorovala těmto typům útoků, existují jisté tagy, které zajistí právě onu ochranu. Jestliže chceme v Django přenést informaci nějaké proměnné z views do templates viz. MTV, použijeme v templates její definovaný tvar například {{title}}. Takto přenesená proměnná je ovšem napadnutelná výše zmíněnými typy útoků. Pokud ovšem k naší proměnné přidáme tag „save“ -> {{title|save}}, Django se už samo postará o ochranu proměnné.

2.3.2 Cross site request forgery (CSRF)

„CSRF je typ útoku na webovou aplikaci nebo službu. Zkratka CSRF (Cross-Site Request Forgery) znamená „podvržení požadavku mezi různými stránkami“. Někdy se také můžeme setkat i s dalšími termíny jako XSRF, Cross-Site Reference Forgery, Session Riding nebo Confused Deputy attacks.“ [5]

Django má vestavěnou ochranu proti většině typů CSRF útoků. V případě, že ji chcete využít, musíte ji povolit v nastavení a poté ji použít. Nicméně, stejně jako u ostatních zobecněných technik i zde existují určitá omezení. Například, subdomény⁷ v rámci nějakého webu mohou nastavit cookies⁸ na straně klienta pro celou doménu.

⁷ *„Subdoména je rozšíření doménového názvu o další úroveň. Je ve tvaru např. eshop.firma.cz, kde eshop je název domény III. řádu. Subdomény je možné vytvářet u domén II. řádu.“ [18]*

⁸ *„Cookies jsou krátké textové soubory vytvářené webovým serverem a ukládané v počítači prostřednictvím prohlížeče. Když se později vrátíte na stejný web, prohlížeč pošle uloženou cookie zpět a server tak získá všechny informace, které si u vás předtím uložil.“ [16]*

Nastavením cookies a za použití odpovídajícího tokenu⁹, můžou pak subdomény obejít ochranu CSRF. Jediný způsob jak se tomu vyhnout, je zajistit to, aby subdomény byly kontrolovány důvěryhodnými uživateli (nebo zakázat nastavení cookies).

2.3.3 SQL injection ochrana

SQL¹⁰ injection je typ útoku, kdy je útočník schopen spustit libovolný SQL kód v databázi. To může mít za následek vymazání či únik dat z databáze. Používáním Django queryset bude výsledné SQL správně načítáno driverem pro podkladové databáze. Avšak Django umožňuje vývojářům psát syrové SQL příkazy nebo provádět přímé úpravy v databázi. Tyto funkce by měly být využívány co nejstřídměji a co nejopatrněji, aby nedošlo k možnému úniku dat. [6]

2.3.4 Clickjacking ochrana

„Clickjacking je způsob útoku na uživatele webových stránek, při kterém uživatel nějakou činností na zdánlivě neškodné stránce (např. kliknutím na tlačítko či obrázek) spustí akci, kterou nepředpokládal. Tato technika využívá zranitelnost, která se vyskytuje ve většině stávajících prohlížečů na všech platformách.“ [7]

Django obsahuje Clickjacking ochranu v podobě X-Frame-Options middleware¹¹, který v podpoře prohlížeče může zabránit otevření stránky uvnitř rámu. Django ovšem nabízí i možnost vypnutí této ochrany.

X-Frame-Options sděluje prohlížeči, zda je dovoleno příslušný obsah zobrazit uvnitř rámu. *„Pokud tedy např. e-shop všechny potenciálně nebezpečné stránky posílá s touto hlavičkou a útočná stránka se pokusí načíst je do rámu, v rámu se objeví jen upozornění prohlížeče, že požadovaný obsah nemůže být z bezpečnostních důvodů zobrazen.“* [8]

⁹ „**Token** - elektronický klíč, který slouží pro ověření identity uživatele.“ [19]

¹⁰ „**SQL** je specializovaný programovací jazyk, který se používá ve vhodném prostředí buď uživatelsky nebo interaktivně k okamžitému řešení úloh (nejčastěji dotazy), nebo se jeho příkazy vkládají do hostitelského jazyka.“ [22]

¹¹ „**Middleware** je software, který umožňuje propojit softwarové komponenty a aplikace v distribuovaném prostředí. Jedná se tedy o integrační nástroj, který zajišťuje přenos dat mezi různými komponentami technické části informačního systému.“ [20]

2.4 Aplikace Django, databáze a server

Django je vyspělý webový framework, který je složen z Pythonovských¹² knihoven. Jak již bylo řečeno výše, jedná se o open-source projekt. A právě díky dostupnosti jeho zdrojových kódů může Django běžet na každém operačním systému, který podporuje Python.

2.4.1 Jak instalovat

Protože Django funguje díky Pythonu, je nutností, ještě před samotnou instalací Djanga, mít nainstalovaný Python. Python je možné stáhnout z jeho oficiálního webu a taktéž se jedná o open-source. Jakou vybrat verzi? V tuto dobu je dostupná již verze 3.6.1 s kterou by neměl být problém. Oproti dlouhodobě spolehlivé verzi 2.7.13 jsou verze 3.x bezpečnější, což byl i jeden z důvodů upgradu Pythonu.

Django je možné instalovat rovnou ze zdrojových kódů nebo za pomoci binárních balíčků.

Pokud bychom vybrali možnost instalace ze zdrojových kódů, pak se jedná o jednoduchý způsob. Stačí nakopírovat potřebné knihovny na místo, kde je Python může používat. Následně se musí přidat spouštěcí cesty do systémových proměnných.

Mnoho distributorů ovšem dnes poskytuje instalační balíček Djanga jako součást jejich „appstoru“. Výhodou je samostatné provedení instalace s možností automatického nainstalování závislostí, jako například databáze adapterů, které Django vyžaduje. Obvykle jsou tyto balíčky založeny na nejnovější verzi Djanga, pokud ale chcete používat vývojovou verzi Djanga, musíte postupovat dle pokynů pro instalaci vývojové verze z oficiálního Djanga Git repositáře. [9]

Pro uživatele, kteří používají operační systém Windows, existují taktéž instalační balíčky. Jak jsem však psal výše, než spustíte instalační balíček, musíte nejprve nainstalovat Python. Poslední dobou se však objevují kompletní balíčky, které obsahují Django s Pythonovým webovým serverem a databází.

¹² **Python** je vysokoúrovňový skriptovací programovací jazyk, který v roce 1991 navrhl Guido van Rossum. Nabízí dynamickou kontrolu datových typů a podporuje různá programovací paradigmatata, včetně objektově orientovaného, imperativního, procedurálního nebo funkcionálního.

2.4.2 Databáze

Pokud hodláte vyvinout obsáhlejší webovou aplikaci, pak se jistě neobejdete bez databáze, ve které budete shromažďovat veškerá data, s kterými budete pracovat. Django nás v tomto směru nikterak neomezuje a umožňuje nám používat téměř veškeré známé typy databází. Velikou výhodou je také to, že Django sjednocuje práce se všemi databázemi do jednotného rozhraní, díky kterému odpadá znalost programování SQL. Jak již už bylo ale zmíněno, Django nám nebrání ani v tom psát vlastní „syrové“ SQL příkazy. Do seznamu podporovaných databází můžeme zahrnout databáze typu SQLite, MySQL, PostgreSQL a Oracle. Pro další typy databází se vyvíjí jejich podpora. Django ovšem nabízí i možnost databáze nevyužívat, ale to vede spíše k nevýhodám jako například, že nebudete moci využívat webového administrátorského rozhraní.

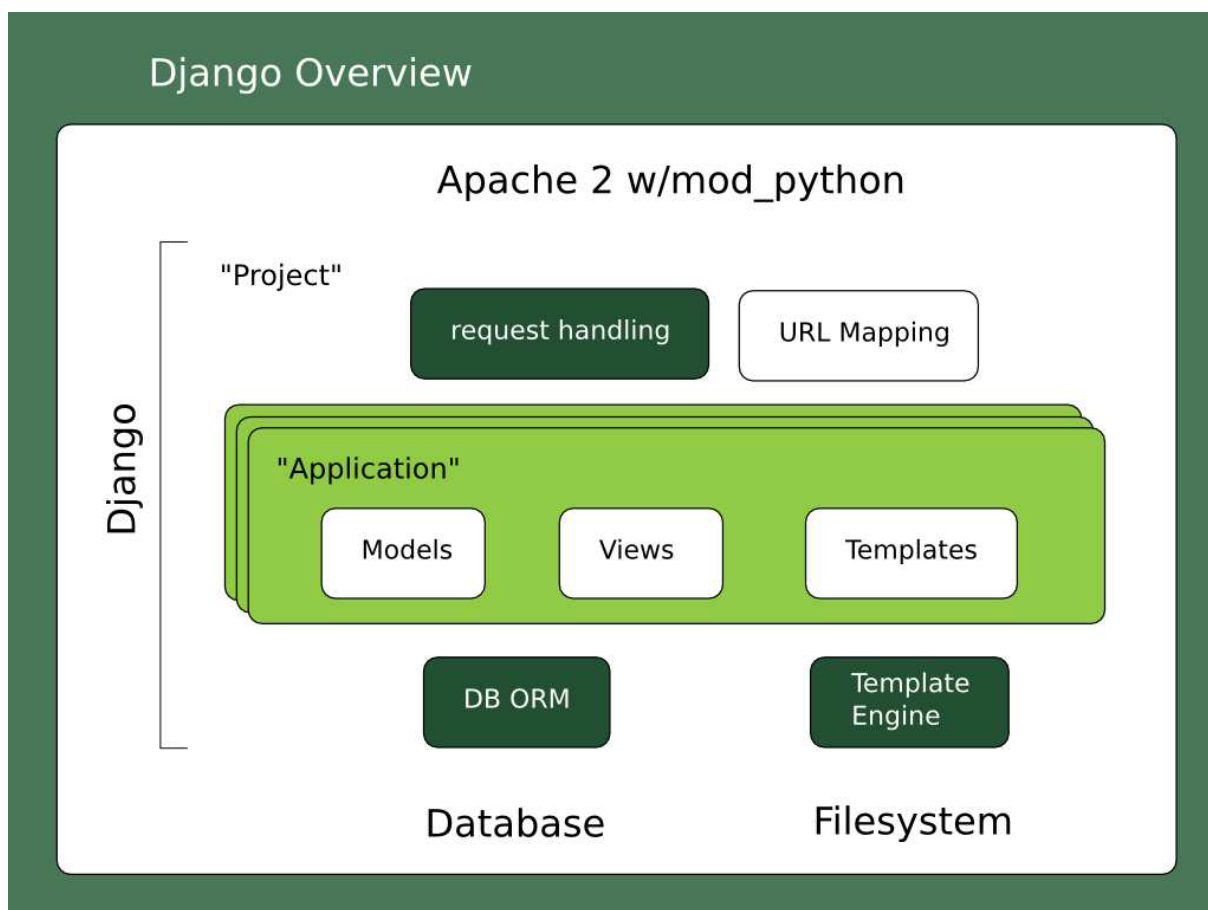
2.4.3 Server

V této části nám Django nabízí odlehčenou verzi webového serveru. Jedná se spíše o vývojový server, který je odlehčený a díky tomu i rychlý. Proto se tedy i hodí pro testování vyvíjené aplikace. Mezi možné změny nastavení můžeme zahrnout pouze nastavení portu nebo změnu IP adresy. Další změny nastavení nejsou možné. Pokud ale vezmeme v potaz, že se server instaluje s Djangem a že se dá spustit jediným příkazem, jedná se nevhodnější volbu pro začínající vývoj.

Pokud by ale naše požadavky na server byly náročnější a potřebovali bychom výkonnější server, není nic jednoduššího než Django nainstalovat na takový server, který nám bude vyhovovat. Podmínkou pro správné fungování Djanga je ovšem serverem poskytnutá podpora Pythonu.

2.5 Struktura Django

Django má v základu již vytvořenou strukturu toho, jak mají všechny projekty vypadat, proto není dovoleno vývojářům strukturu vytvářet nebo měnit. Pro jednodušší představu jak tato struktura vypadá, nám poslouží obrázek 2.1, který je převzat ze serveru excess.org.



Obrázek 2.1 – Struktura projektu

2.5.1 Projekt

Jak napovídá obrázek, založení projektu je nepostradatelnou součástí budoucí aplikace. Jako má dům své základy tak i aplikace potřebuje základ, který nabízí vytvořený projekt. K tomu, abychom mohli vytvořit projekt slouží příkaz `django-admin.py startproject mysite`, který stačí zadat do příkazové řádky. Takto vznikne projekt s názvem `mysite` umístěný do adresáře, který si určíme taktéž v příkazové řádce.

Vytvořené soubory příkazem `startproject` právě tvoří již zmíněný základ. Jsou to soubory nacházející se v adresáři s názvem `mysite`.

- `__init__.py`: Prázdný soubor, který říká Pythonu, že adresář má být považován za Python balíček.
- `manage.py`: Utilita pro příkazový řádek, která vám umožňuje projekt spravovat. Možnosti které vám nabízí se dají jednoduše zobrazit použitím příkazu `python manage.py help`. Pokud tedy chce použít jednu z možností, které nám nabízí, stačí použít příkaz ve tvaru `python manage.py [vybraná možnost]`. Mimo jiné, pomocí tohoto souboru můžeme spustit lokální vývojový server příkazem `runserver` nebo Pythonovský shell příkazem `shell`.
- `settings.py`: Zde se nachází v zásadě veškeré nastavení projektu. Je tedy velice důležitý. Django má v základě nastavenou databázi SQLite, které je v mnoha ohledech dostačující. Pokud ovšem chceme použít jinou formu databáze, pak právě zde se musí změnit její nastavení. Dále se zde nastavují cesty ke statickým souborům, kterými může být například stylový soubor CSS¹³ nebo obrázek. Další nastavení není třeba měnit nebo nějak upravovat, protože vše je již vývojáři šikovně nastaveno.
- `urls.py`: Deklarace URL adres pro Django projekt, dalo by se mluvit o jakém si „rejstříku“ tvořeného webu, aplikace. Každý odkaz, adresa v naší aplikaci zde musí mít záznam, aby byla dostupná. Záznam, který nás odkazuje na určité funkce a který také po jeho zavolání předává parametry GET nebo POST dotazu.

2.5.2 Aplikace

Nový projekt je založen a nyní můžeme vytvořit první aplikaci. Minimálně jedna aplikace v projektu být musí. Chceme-li ale rozsáhlejší projekt, nic nám nebrání v tom, abychom měli v jednom projektu více aplikací. Informace mezi nimi je posléze možné navzájem sdílet. Aplikací rozumějme část programu, která má na starosti určitou úlohu. Může to být například aplikace pro správu uživatelů. Taková aplikace se bude starat o jejich

¹³ **CSS** „(Cascading Style Sheets) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents.“ [23]

přihlášení, odhlášení či registraci do webové aplikace, o změnu hesla nebo o úpravu osobních údajů.

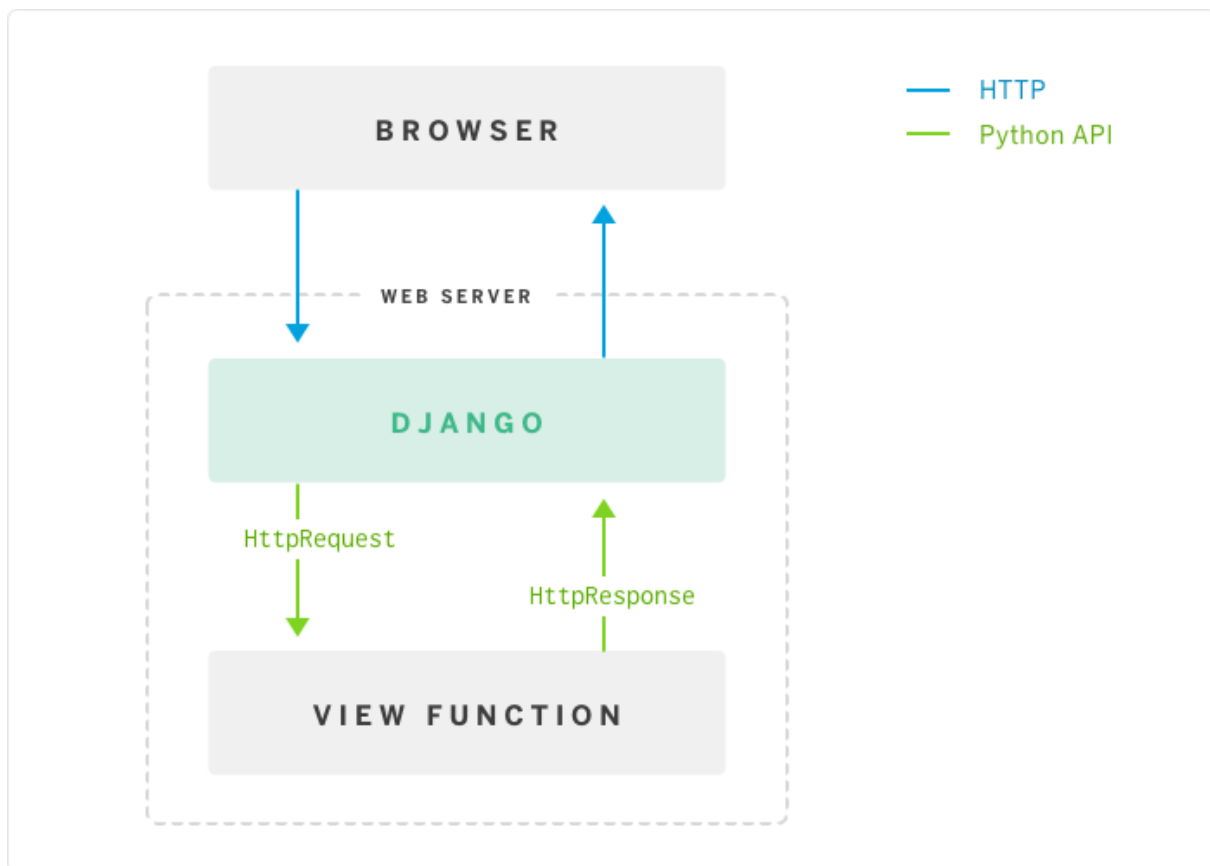
Dané pravidlo na to, jak by se měl projekt členit do jednotlivých aplikací neexistuje. Zde to závisí pouze na samotném vývojáři. Doporučení je však takové, svůj projekt rozdělit do několika aplikací, které pak mezi sebou spolupracují. Na druhou stranu není doporučeno mít na každou funkci vlastní aplikaci nebo naopak všechno nasypat do jedné obří aplikace.

Jak aplikaci vytvořit? Je to neméně podobné jako vytvoření projektu. V příkazové řádce si otevřeme adresář, který nám vytvořil projekt a zde použijeme příkaz `python manage.py [název aplikace]`. Tento příkaz automaticky vytvoří nový adresář s názvem, který jsme zadali v příkazu pro vytvoření nové aplikace, tedy `[název aplikace]`. Adresář v základu obsahuje soubory `__init__.py`, `models.py`, `views.py`. Aby Django ale o vytvořené aplikaci vědělo, musí se do výše popsaného souboru `settings.py`, do proměnné `INSTALLED_APPS` aplikace přidat.

Soubor `__init__.py` znovu popisovat nebude neb zastává stejnou funkci jako v případě projektu. Následně tedy popíšu zbylé soubory.

- **models.py:** Tento soubor v podstatě definuje schéma databáze. Pythonovské třídy zapsané v tomto souboru definují jednotlivé tabulky a atributy třídy pak definují její sloupce. Každý model je pak odvozený z třídy `django.db.models.Model`. Pokud máme všechny třídy pro aplikaci připravené, musí se podle nich vytvořit tabulky v databázi. Následujícím příkazem `python manage.py sql [název aplikace]` se vytvoří SQL příkazy `CREATE TABLE` pro aplikaci. Jestliže je vše v pořádku, dalším příkazem `python manage.py syncdb` se takto vytvoří tabulky v databázi. Django do každé tabulky přidává sloupec s primárním klíčem, kde není definován.
- **views.py:** Do tohoto souboru se definují funkce, jejichž výsledky se prezentují prostřednictvím šablon. V jednoduchosti řečeno, `views.py` obstarává spolupráci mezi jednotlivými částmi Djanga. Funkce, které jsou ve `views` definovány se volají pomocí souboru `urls.py`. Stejně tak, pokud se jako uživatelé dotazujeme na nějaká data z databáze, pak právě ve `views` musí být nastavena funkce, která

vytáhne žádaná data z databáze a pak je přes šablonu zobrazí nám, uživateli. Viz Jednoduché schéma na obrázku 2.2.



Obrázek 2.2 – Schéma views (převzato od Heroku.com)

- **forms.py:** Pokud má aplikace pracovat s formuláři použitými na webové stránce, musí být v tomto souboru, který je dodatečně vytvořen, každý z nich definovaný. Funguje to na stejném principu jako u souboru models.py. Jednotlivé třídy definují formuláře a jejich atributy pak jednotlivé kolonky, které obsahují. Pokud se na takto definovaný formulář odkážeme pomocí views, pak samo Django v templates odvodí validovaný formulář.

3. Přidružené nástroje a jazyky

V této kapitole bych chtěl seznámit čtenáře se softwarem a jazyky, které jsou použity v aplikaci EasyMeasurement. Většina informací použitých v této kapitole pochází z oficiálních stránek jednotlivých projektů.

- HTTP
- Python
- MySQL
- CSS
- HTML
- Boozstrap
- Sublime Text 2

3.1 HTTP: Internetový mediální kurýr

„Biliony JPEG obrázků, HTML stránek, textových souborů, MPEG videí, WAV audio souborů, Java appletů, a mnoho dalších proudí skrze Internet každý den. HTTP hromadně přenáší tyto informace rychle, výhodně a spolehlivě z webových serverů z celého světa do našich počítačů, webových prohlížečů. Protože HTTP používá spolehlivé protokoly datových přenosů, je garantováno, že vaše data nebudou zničena nebo pomíchána při přenosu, dokonce i když k vám přijdou z druhé strany zeměkoule. Toto je dobré pro vás jako uživatele, protože můžete přijmout informace bez starosti o vaše soukromí. Spolehlivý přenos je také dobrý pro vás jako pro vývojáře internetových aplikací, protože se

nemusíte strachovat o to, že by HTTP komunikace byla zničena, z duplikována nebo zkreslena. Můžete se zaměřit na programování rozlišujících detailů vaší aplikace bez strachování o chyby a slabosti Internetu.“ [10]

3.2 Python

„Jazyk Python začal vznikat v roce 1989 ve výzkumném ústavu v Amsterdamu. Při jeho zrodu stál Guido van Rossum a je vidět, že u návrhu dostatečně přemýšlel. Vznikl promyšlený jazyk, který je stále ve vývoji. Jméno dostal podle pořadu BBC Monty Python's Flying Circus. V současné době běží na mnoha platformách (Linux, Win, Mac, WinCE, OS/2, Java). Stejně tak programy v něm napsané lze na těchto systémech téměř vždy spouštět bez úprav.“ [11]



Obrázek 3.1 – Python

Python je čistý objektový a výkonný programovací jazyk. Je vyvíjen jako open-source, což znamená, že si každý může přečíst zdrojový kód. Python je dynamický interpretovaný jazyk, to znamená, že se kód překládá až za běhu. Důsledkem toho se některé případné chyby projeví až při spuštění programu. Na druhou stranu je díky tomu samotný kód jednodušší.

Python spolehlivě běží na nejběžněji používaných operačních systémech jakou jsou například Windows, UNIX nebo MacOS X.

Zdrojové kódy jsou volně poskytnuty ke stažení na oficiálním webu <http://python.org/> pod licencí Python Software Foundation License, která se v mnoha ohledech podobá licenci BSD.

3.2.1 Proč Python

Python jazyk je ideální pro začínající programátory. Jeho syntaxe je „jednoduchá a čistá“. Podporuje tři programovací paradigma a to procedurální, funkcionální a objektové. Další výhodou je to, že jeho jádro a složitější funkce jsou naprogramovány v jazyce C, což se projevuje na rychlosti výpočtů programů. Jak již bylo řečeno, Python, než vyhodnotí nějakou chybu v kódu, má snahu program spouštět, což může být motivující pro začátečníky. Většina ostatních programovacích jazyků se totiž nespustí, dokud nejsou všechny chyby vyřešeny.

3.3 MySQL

Jak již bylo řečeno výše, v rámci vývoje aplikace, bude nutné používat databázi. Django podporuje mnoho typů databází z nichž jedna je databáze typu MySQL. MySQL AB je softwarová společnost, která vyvinula tento databázový systém. Byla založena Michaelem Wideniusem, Davidem Axmarkem a Allan Larssonem v roce 1995. Dnes je to jedna z nepoužívanějších databází vůbec. Jedná se o tzv. relační databázi.



Obrázek 3.2 - MySQL

Myslí se tím, že je založena na jednotlivých tabulkách, z nichž každá obsahuje položky jednoho typu. Například jedna z tabulek s názvem uživatelé bude obsahovat informace o uživateli, další tabulka s názvem články pak bude obsahovat informace o článcích. Taková databázová tabulka může vypadat jako tabulka v Excelu viz tabulka 3.1 – uživatelé.

Jméno	Příjmení	Datum narození	Počet článků
Josef	Novák	26/5/1993	11
Tomáš	Zima	12/4/1989	1
Pavel	Kopřiva	30/1/1988	23

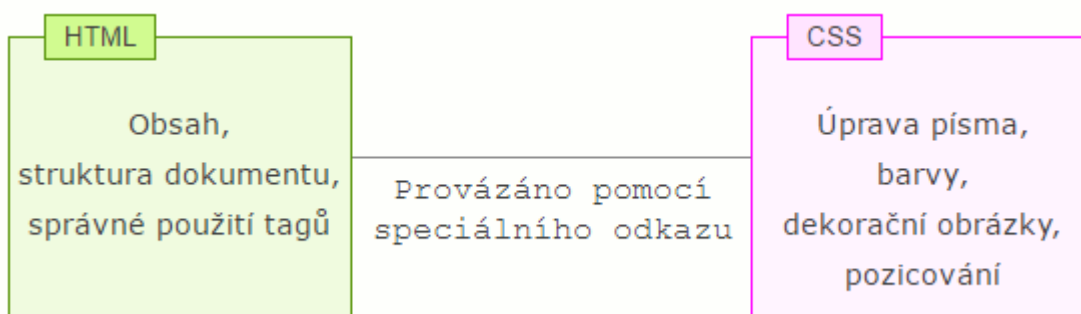
Tabulka 3.1 - Uživatelé

Z toho plyne, že jednotlivé položky se ukládají do řádků. Sloupce pak označují atributy, které položky mají. Protože databáze MySQL je typovaná, což znamená, že každý sloupec ukládá data jednoho typu, musí se stanovit pro každý sloupec určitý typ. Typy jsou pak například číslo, znak, krátký text, dlouhý text atd. Při návrhu tabulky by se nemělo zapomínat na sloupec do kterého se ukládá identifikační klíč. Pomocí těchto klíčů se pak dají jednotlivé tabulky spojovat.

3.4 CSS

„Obecně řečeno, CSS je nějaký zápis, který určuje vzhled (barvy, dekorační obrázky, rozmístění prvků) HTML dokumentu. CSS je zkratka slov Cascading Style Sheets. Do češtiny bychom to přeložili jako Tabulky kaskádových stylů. V praxi se pak častěji používají tyto názvy: Kaskádové styly, styly, CSS soubor(y).“ [12]

Soubor, který obsahuje kaskádové styly, má příponu css. K tomu, aby se ale projevilo jeho potenciál, ho musíme spojit se souborem s příponou html. Viz obrázek 3.3 – Propojení HTML a CSS. Obrázek je převzat ze serveru pestujemeweb.cz



Obrázek 3.3 – Propojení HTML a CSS

```
i 1 <div class="ukazkovyBox">Obsah html souboru</div>
2
HTML
CSS
1 .ukazkovyBox {
2   color:white;
3   font-size:20px;
4   background:green;
5   padding:10px;
6   max-width:200px;
7   border:solid 2px black;
8   border-radius:10px;
9   box-shadow:5px 5px 10px black;
10 }
```

Obsah html souboru

Obrázek 3.4 – Ukázka CSS

Jednoduše řečeno, HTML se stará o obsah webové stránky a o formu se stará právě CSS. Forma a obsah by měly být striktně odděleny. Pokud tedy používáme formátování pomocí CSS neměli bychom v samotném HTML nijak zasahovat do formátování. CSS jazyk byl navržen organizací W3C. První návrh pocházel od autora Hakov Wium Lie. Postupně byly vydány tři verze CSS1, CSS2 a CSS3. Jak již bylo řečeno, hlavní myšlenkou byla možnost oddělit vzhled dokumentu od jeho struktury obsahu.

3.5 HTML

V dnešní době snad každý, kdo se aspoň trochu zajímá o počítače, ví, čemu se tato část práce bude věnovat. HTML (HyperText Markup Language) je jazyk, který se používá pro tvorbu webových stránek. Stará se o popis toho, co webová stránka obsahuje. To znamená, že uspořádává vše od textu přes odkazy až po obrázky a další prvky, které webová stránka může obsahovat. Dalo by se říci, že se jedná o technologii, bez které by web nemohl existovat.

„První definici jazyka HTML vytvořil v roce 1991 Tim Berners-Lee jako součást projektu WWW, který měl umožnit vědcům zabývajícím se fyzikou vysokých energií komunikaci a

sdílení výsledků výzkumu po celém světě. Ne náhodou proto celý projekt vznikl v CERNu (Centre Européenne pour la Recherche Nucléaire, Evropské centrum jaderného výzkumu), který leží na švýcarsko-francouzských hranicích nedaleko Ženevy. Jednalo se o výsledek dvouletého projektu, který měl vyřešit problémy se sdílením informací ve velké instituci, jakou CERN bezesporu je. Umožňovala text rozčlenit do několika logických úrovní, použít několik druhů zvýraznění textu a zařadit do textu odkazy a obrázky.

Berners-Lee při návrhu HTML nepředpokládal, že by autoři webových stránek museli tento jazyk znát. První verze WWW softwaru byla napsána pro operační systém NextStep a obsahovala jak prohlížeč, tak i integrovaný editor webových stránek. Software vyvinutý v CERNu včetně specifikací jazyka HTML byl veřejně uvolněn v roce 1991. Když Marc Anderssen se svými kolegy z NCSA (National Center for Supercomputing Applications) psal dříve populární prohlížeč Mosaic, považoval za příliš obtížné implementovat do programu rovnou i editor HTML. Díky tomuto rozhodnutí a tomu, že ne každý provozuje na svém počítači NextStep, je dnes nutné, aby autoři profesionálních stránek znali jazyk HTML.” [13]

Na následujícím obrázku 3.5 – *HTML příklad* bych chtěl jednoduchým příkladem demonstrovat, jak vypadá webová stránka v html kódu a jak ji zobrazuje internetový prohlížeč.

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2 <html>
3   <head>
4     <meta http-equiv="content-type" content="text/html; charset=utf-8">
5     <title>Ukázková stránka</title>
6   </head>
7   <body>
8     <h1>Nadpis 1</h2>
9     <p>
10      Lorem Ipsum je demonstrativní výplňový text používaný v tiskařském a
11      knihařském průmyslu. Lorem Ipsum je považováno za standard v této
12      oblasti už od začátku 16. století, kdy dnes neznámý tiskař vzal
13      kusy textu a na jejich základě vytvořil speciální vzorovou knihu.
14      Jeho odkaz nevydržel pouze pět století, on přežil i nástup elektronické
15      sazby v podstatě beze změny. Nejvíce popularizováno bylo Lorem Ipsum v
16      šedesátých letech 20. století, kdy byly vydávány speciální vzorníky s
17      jeho pasážemi a později pak díky počítačovým DTP programům jako Aldus
18      PageMaker.
19     </p>
20   </body>
21 </html>
22
23

```

Nadpis 1

Lorem Ipsum je demonstrativní výplňový text používaný v tiskařském a knihařském průmyslu. Lorem Ipsum je považováno za standard v této oblasti už od začátku 16. století, kdy dnes neznámý tiskař vzal kusy textu a na jejich základě vytvořil speciální vzorovou knihu. Jeho odkaz nevydržel pouze pět století, on přežil i nástup elektronické sazby v podstatě beze změny. Nejvíce popularizováno bylo Lorem Ipsum v šedesátých letech 20. století, kdy byly vydávány speciální vzorníky s jeho pasážemi a později pak díky počítačovým DTP programům jako Aldus PageMaker.

Obrázek 3.5 – HTML příklad

3.6 Bootstrap

Bootstrap je jeden z nejpoužívanějších nástrojů, který usnadňuje formátování HTML, CSS a JavaScript aplikací. Pokud se nechcete trápit úpravou webových stránek, ať už pomocí HTML, CSS nebo JS, je Bootstrap ideální možností, jak zdlouhavé programování formátu webu obejít. Bootstrap obsahuje velké množství knihoven, na které se můžete odkázat pouhým odkazem, který vložíte do hlavičky webové stránky. Následně pak můžete pomocí jednotlivých tříd ovlivňovat samotné části webové stránky. Jedná se například o schématické rozložení dané stránky, návrh tabulky, formát formulářů nebo jednotlivých tlačítek.



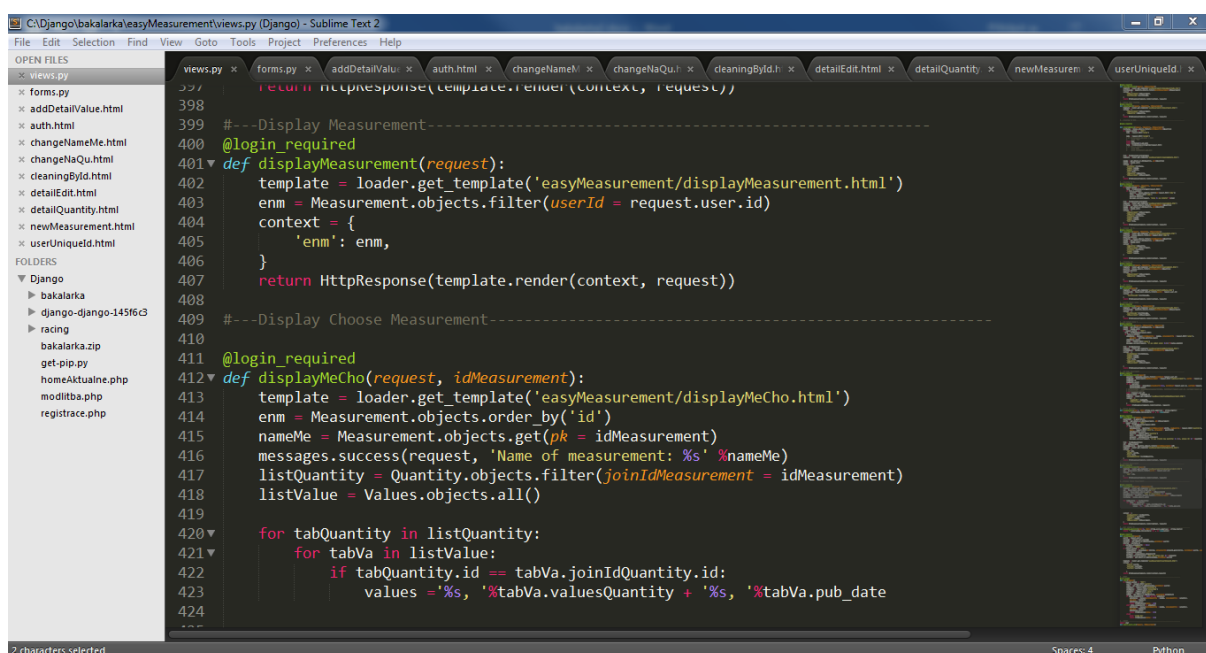
Obrázek 3.6 - Bootstrap

Výhodou Bootstrapu je neustálé inovování knihoven. Díky tomu se nemusíme bát, že by se použité programovací kódy postupem času staly neefektivními pro naše webové stránky.

Bootstrap byl vytvořen na Twitteru v polovině roku 2010 autory @mdo a @fat. Před tím, než byl Bootstrap vydán jako open-source framework byl znám spíše jako Twitter Blueprint. Datum původního vydání se datuje na pátek 19. srpna 2011.

3.7 Sublime Text 2

Pro programování v jazyce Python a jiných typů jazyků existuje mnoho softwarů, z kterých si můžete vybrat. Mě osobně oslovil svou jednoduchostí a přehledným uživatelským rozhraním software Sublime Text. Sublime Text podporuje většinu známých programovacích jazyků s nimiž se můžeme setkat. Nelze ovšem říci, že se jedná o nejlepší software, protože zde se jedná spíše o subjektivní názor samotného programátora, o to, jak jaký software mu vyhovuje. Více informací o Sublime Text lze nalézt na oficiálních stránkách softwaru. <https://sublimetext.com/>



Obrázek 3.7 – Sublime Text

4. Webová aplikace EasyMeasurement

Tato kapitola se věnuje webové aplikaci EasyMeasurement, která byla vytvořena za pomoci frameworku Django a dalších přidružených programů, jazyků, které jsou popsány v předchozí kapitole č. 3.

V mnoha ohledech by se mohlo zdát, že taková aplikace, jejíž funkce byla již nastíněna v úvodu, je zbytečná, protože snad každý student či laborant je schopný si v excelu nebo podobném programu vytvořit jednoduchou tabulku, do které může zaznamenávat hodnoty svého měření. Pokud by ovšem chtěl zaznamenávat například dlouhodobé měření, excel již není šťastnou volbou. Stejně tak i pokud by se jednalo o měření náročné na rychlost zaznamenávání měřených hodnot. V tuto chvíli se webová aplikace EasyMeasurement nabízí jako vhodnější a bezpečná varianta. Aplikace umožňuje ukládat data posílaná prostřednictvím url odkazu. Stačí tedy nastavit nějaké zařízení, které by bylo schopné zformulovat a odeslat tvar url odkazu (popsáno níže), do kterého by se uložila naměřená hodnota. Webová aplikace pak už sama rozpozná, o jaké se jedná měření, měřenou veličinu a hodnotu uloží se správným klíčem do databáze, kterou je samozřejmě možné zpětně zobrazit, upravit či exportovat do excelové tabulky.

Z hlediska bezpečnosti musí mít každý uživatel poskytnuty přihlašovací údaje přímo od hlavního administrátora aplikace, jinak není možné aplikaci používat. To platí i v případě používání ukládání dat pomocí url odkazu. Url odkaz musí obsahovat mimo samotnou naměřenou hodnotu a adresu serveru, kam se má poslat identifikační údaje s unikátním klíčem, který si uživatel může vygenerovat po přihlášení do aplikace. Tímto je tedy zamezeno cizímu vniknutí do „soukromého“ měření.

4.1 Popis projektu

V tuto dobu je možné si aplikaci EasyMeasurement vyzkoušet na serveru s odkazem <http://vojtechkadlec.pythonanywhere.com>. Jako přihlašovací údaje použijte následující:

Username: **demoUser**

Password: **demo1230**

Jak jste si mohli všimnout na odkazu, aplikace běží na doméně třetího řádu. Tu poskytuje serverový hostitel Pythonanywere.com. Dle mého názoru se jedná o jedno z nejjednodušších a nejefektivnějších řešení, jak publikovat své aplikace tvořené v Pythonu, posléze v Django. Tento hostitel již v základu poskytuje funkčnost Pythonu i Django, takže my jako uživatelé se nemusíme starat o jejich instalaci a uvedení do provozu. Nám stačí pouze nakopírovat naši aplikaci na server a nastavit pár parametrů pro její správné fungování.

Vraťme se ovšem zpět k popisu aplikace. Z jednotlivých střípků, které jsou rozesety v celé práci už asi máte představu o čem a jak aplikace funguje, nicméně rád bych v této podkapitole znovu kompletně shrnul, jak aplikace vypadá a funguje. Smyslem aplikace EasyMeasurement je jednoduchý a bezpečný sběr dat. Jak jsem již psal v úvodu této kapitoly, aplikace vyžaduje přihlášení neboli ověření uživatele. Bez přihlášení do aplikace není možné ji používat ani v omezené formě. Jediné, kam se dostanete, je úvodní stránka. Aplikace prozatím neumožňuje registraci uživatele, proto každého nového uživatele musí do databáze přidat hlavní administrátor, správce aplikace. Po přihlášení se zpřístupní záložky **New, Edit, Add value, Display a Unique key**. Jednotlivé názvy záložek samy napovídají, k čemu jsou určeny, přesto však stručně popíšu, co od každé čekat.

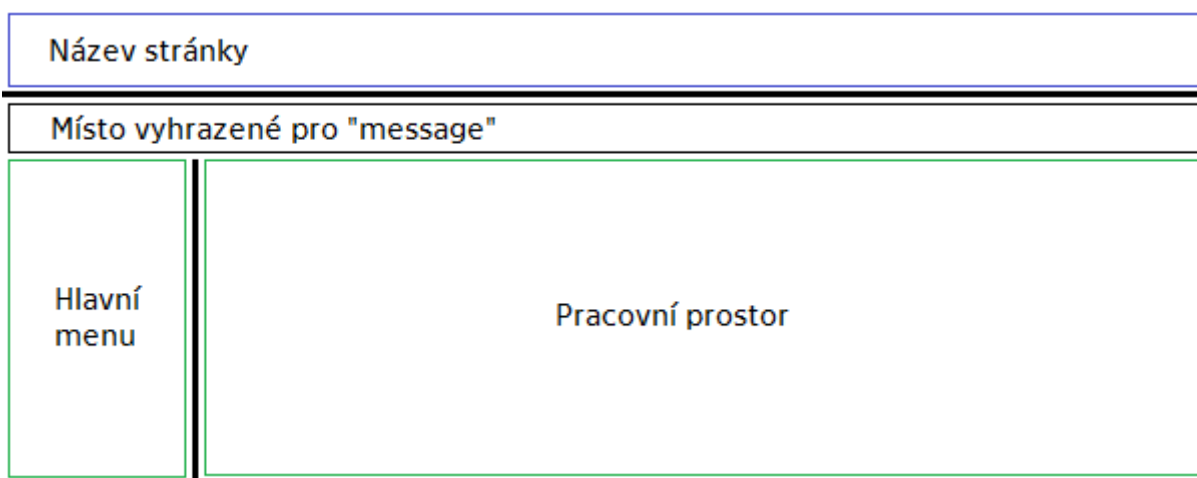
- **New:** Tato záložka vás vybídne vytvořit nové měření s následující možností přiřadit jednotlivé veličiny, které chcete měřit.
- **Edit:** Tato záložka umožňuje upravovat vytvořená měření i s již naměřenými hodnotami. Prvně se vás aplikace zeptá, které měření chcete upravovat. Následně pak můžete vybrat z jednotlivých úprav jako například: Změna názvu měření, smazání měření, přidání nové veličiny nebo její úprava, a nakonec mazání

měřených hodnot. To je rozděleno do tří kategorií, čišťení dle časového záznamu, čišťení dle vlastního ID hodnoty a kompletní čišťení všech hodnot dané veličiny.

- **Add value:** Po vybrání měření a veličiny můžete přidávat jednotlivé hodnoty.
- **Display:** Zde stačí vybrat pouze měření, které chcete zobrazit. Aplikace pak pro každou veličinu s hodnotami z vybraného měření vytvoří vlastní tabulky. Chcete-li exportovat naměřená data, v této záložce je ta možnost. V seznamu veličin musíte vybrat tu, kterou chcete exportovat a pak tlačítek Export potvrdit export. Automaticky se začne stahovat xls (formát souborů tabulkových listů MS Excelu) soubor s názvem vybraného měření.
- **Unique key:** Tato záložka je důležitá, pokud chcete odesílat data do aplikace z externího zařízení. Dozvíte se zde jaké máte User ID a Unique key. Tyto hodnoty musí obsahovat url odkaz, aby aplikace umožnila uložit data do databáze. Unikátní klíč je zde možné měnit.

4.1.1 Prostředí aplikace

Vzhled webové aplikace je velice jednoduchý a pro uživatele nenáročný. Je optimalizována na rozlišení rovná 1024x768 a větší. Výšku daného rozlišení překročí jen na stránkách, kde jsou zobrazeny naměřené hodnoty. Jak je možné vidět na obrázku 4.1, v hlavičce je umístěn název aplikace a informace o vašem stavu, zda jste nebo nejste přihlášení. Hned pod hlavičkou se zobrazují tzv. message, resp. zpráva o tom, co bylo nebo naopak nebylo provedeno. V levé části stránky je umístěno hlavní menu s výše popsanými záložkami. V pravé a větší části se nachází „pracovní prostor“. V něm jsou zobrazovány záložky.



Obrázek 4.1 – Prostředí aplikace

4.1.2 Šablona – base.html

Pro vhodné zpracování výše uvedeného prostředí je vhodné použít šablonovacího systému Django. Ten spočívá ve vytvoření základního souboru, chcete-li šablony, `base.html`. Ten obsahuje základní rozvržení webové stránky, v případě této aplikace dle obrázku 4.1 – Prostedí aplikace. Jak takový soubor vypadá a funguje je znázorněno na následující ukázce. Zde použijte zkrácenou verzi, než které je využito v aplikaci.

```
<html>
  <head>
    <title>EasyMeasurement</title>
  </head>
  <body>
    <div class="page-header ">
      <h1>Easy Measurement</h1>
    </div>
    <div class="messages">
      Zde se zobrazují message
    </div>
    <div class="content container">
      <div class="col-md-2">
        Zde je umístěno hlavní menu
      </div>
      <div class="col-md-10">
        {% block content %}
          Zde je umístěn pracovní prostor
        {% endblock %}
      </div>
    </div>
  </body>
</html>
```

Jedná se o jednoduchý kód v jazyce HTML, který je ovšem obohacen o klíčové značky, které dělají tento soubor šablonou. Jde o značky `{% block content %}` a `{% endblock %}`. Do tohoto místa se budou nahrávat jednotlivé stránky, na které se uživatel bude dotazovat. Pro správné fungování je ale zapotřebí dopsat podobné značky do jednotlivých stránek, které se mají zobrazovat v šabloně.

```
{% extends "base.html" %}
  {% block content %}
```

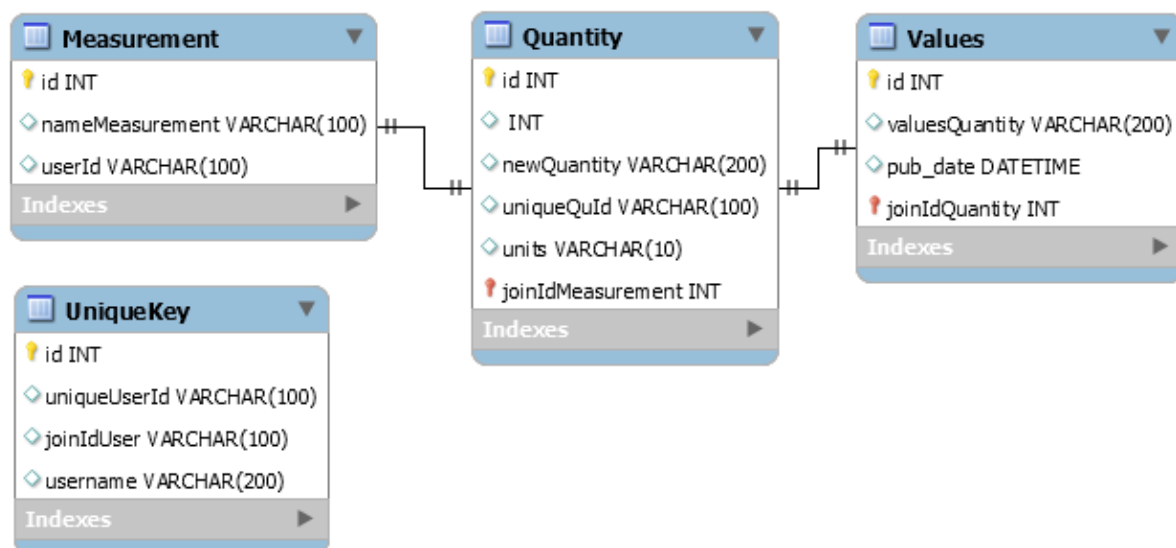
Obsah stránky.

```
{% end block %}
```

Django však nabízí mnoho dalších značek, které je možné využít v souborech HTML. V případě aplikace EasyMeasurement jsem ještě využil značky `{% if parametr %}{% endif %}`, `{% for parametr_x in parametr_y %}{% endfor %}` nebo `{{parametr}}`. V prvním případě se jedná o klasický „if“, podmíněný příkaz. Pokud se `parametr` rovná hodnotě `False`, blok v „ifu“ nebude proveden. V druhém případě se jedná o velice známý `for` cyklus. Do parametru `x` se postupně nahrává parametr ze seznamu, z parametru `y`. V třetím případě se jednoduše vypíše hodnota parametru.

4.2 Databáze

Aplikace EasyMeasurement je napojená na databázi MySQL. Při samotném začátku, kdy je aplikace tvořena, Django definuje několik základních tabulek, které jsou společně s tabulkami navrženými programátorem vytvořené za použití příkazu `syndb`, jak již bylo popsáno v druhé kapitole. Já zde uvedu pouze tabulky, které jsem navrhl v rámci vývoje aplikace. Jde tedy o tabulky s názvy `Measurement`, `Quantity`, `Values` a `UniqueKey`. Na obrázku 4.2 je zobrazeno schéma tabulek a jejich závislosti.



Obrázek 4.2 – Schéma databáze

4.3 Databázové modely

Uspořádání databáze, jak je popsáno v předchozí podkapitole, vychází v základu podle souboru `models.py`. Jak soubor funguje zde popisovat nebudu, protože jsem již tak učinil v podkapitole 2.5.2 na straně 13. Místo toho tady popíšu jednotlivé třídy, které jsou použity v aplikaci.

```
class Measurement(models.Model):
    nameMeasurement = models.CharField(max_length=100)
    userId = models.CharField(max_length=100)
    def __str__(self):
        return self.nameMeasurement
```

Třída `Measurement` má definovány dva parametry a to `nameMeasurement`, který vytvoří sloupec tabulky `Measurement` v databázi, kam se budou ukládat názvy měření a `userId`, který taktéž vytvoří sloupec v tabulce `Measurement`, kam se budou ukládat ID uživatelů. Poslední dva řádky slouží k tomu, že když například ve views zavoláme určitou metodou objekt z tabulky `Measurement`, vrátí nám název měření.

```
class Quantity(models.Model):
    joinIdMeasurement = models.ForeignKey(Measurement,
    on_delete=models.CASCADE)
    newQuantity = models.CharField(max_length=200)
    uniqueQuId = models.CharField(max_length=100)
    units = models.CharField(max_length=10)
    def __str__(self):
        return self.newQuantity
```

Třída `Quantity` má definovány čtyři parametry. Parametr `joinIdMeasurement` spojuje veličinu s daným měřením. Parametr `newQuantity` představuje místo pro název veličiny, `uniqueQuId` představuje místo pro unikátní klíč veličiny a `units` představuje místo pro jednotky. Poslední dva řádky mají stejný význam jako v případě třídy `Measurement`.

```

class Values(models.Model):
    joinIdQuantity = models.ForeignKey(Quantity,
    on_delete=models.CASCADE, null=True, blank=True)
    valuesQuantity = models.CharField(max_length=200)
    pub_date = models.DateTimeField('Date Measurement')

```

Třída **Values** má definovány tři parametry. Parametr **joinIdQuantity** spojuje naměřenou hodnotu s danou veličinou. Parametr **valuesQuantity** představuje místo pro naměřenou hodnotu. Parametr **pub_date** představuje místo pro časový záznam, kdy byla měřená veličina zaznamenána.

```

class uniqueKey(models.Model):
    uniqueUserId = models.CharField(max_length=100, null=True)
    joinIdUser = models.CharField(max_length=100)
    username = models.CharField(max_length=200)
    def __str__(self):
        return self.uniqueUserId

```

Třída **uniqueKey** má definovány tři parametry. Parametr **uniqueUserId** představuje místo pro unikátní klíč uživatele. Parametr **joinIdUser** představuje místo pro ID uživatele. Parametr **username** představuje místo pro uživatelské jméno uživatele. Poslední dva řádky mají stejný význam jako v případě třídy **Measurement**.

4.4 Funkce definované pro aplikaci

V této podkapitole budou shrnuty použité funkce aplikace, které se nachází v souboru **views.py**.

- **Index(request)**: Tato funkce pouze načte hlavní stránku aplikace.
- **newMeasurement(request)**: Tato funkce načte odpovídající formulář ze souboru **forms.py**. Zároveň se stará o jeho částečnou validaci.
- **addNewMeasurement(request)**: Tato funkce vytvoří nové měření.
- **editMeasurement(request)**: Tato funkce vytvoří vyfiltrovaný seznam měření aktuálně přihlášeného uživatele.
- **detailEdit(request, measurement_id)**: Tato funkce stáhne z databáze informace potřebné k úpravě jednotlivých měření, veličin a hodnot. Pokud funkce

zaznamená v requestu metodu POST, vymaže vybrané měření. Na tuto funkci se totiž odkazuje metodou POST jen požadavek k smazání měření. Parametr `measurement_id` nese informaci o vybraném měření.

- `changeNameMe(request, idMeasurement)`: Tato funkce načte odpovídající formulář ze souboru `forms.py`. Zároveň se stará o jeho částečnou validaci. Pokud funkce zaznamená v requestu metodu POST, změní jméno měření dle hodnot poslaných právě metodou POST z formuláře.
- `chooseQuantity(request, idMeasurement)`: Funkce vytvoří seznam veličin vybraného měření.
- `editQuantity(request, idQuantity, idMeasurement)`: Funkce stáhne potřebné informace z databáze pro následující operace.
- `changeNaQu(request, idQuantity, idMeasurement)`: Tato funkce načte odpovídající formulář ze souboru `forms.py`. Zároveň se stará o jeho částečnou validaci. Pokud funkce zaznamená v requestu metodu POST, změní jméno veličiny dle hodnot poslaných právě metodou POST z formuláře.
- `deleteQu(request, idQuantity, idMeasurement)`: Funkce vymaže vybranou veličinu i s naměřenými hodnotami.
- `chooseQuantityRe(request, idMeasurement)`: Funkce vytvoří seznam veličin vybraného měření.
- `chooseTypeCl(request, idQuantity, idMeasurement)`: Funkce stáhne potřebné informace z databáze pro následující operace.
- `cleaningByDate(request, idQuantity, idMeasurement)`: Tato funkce načte odpovídající formulář ze souboru `forms.py`. Zároveň se stará o jeho částečnou validaci. Pokud funkce zaznamená v requestu metodu POST, vymaže naměřené hodnoty podle časového intervalu, který je odeslán z formuláře metodou POST.
- `cleaningById(request, idQuantity, idMeasurement)`: Funkce je stejného charakteru jako funkce `cleaningByDate` s rozdílem mazání hodnot dle jejich ID.
- `cleaningByAll(request, idQuantity, idMeasurement)`: Funkce je stejného charakteru jako funkce `cleaningByDate` s tím rozdílem, že smaže všechny hodnoty dané veličiny

- `addValue(request)`: Tato funkce vytvoří vyfiltrovaný seznam měření aktuálně přihlášeného uživatele.
- `detailChooseMe(request, idMeasurement)`: Funkce vytvoří seznam veličin vybraného měření.
- `addDetailValue(request, idQuantity, idMeasurement)`: Tato funkce načte odpovídající formulář ze souboru `forms.py`. Zároveň se stará o jeho částečnou validaci. Pokud funkce zaznamená v requestu metodu POST, přidá novou hodnotu měření dané veličině.
- `detailQuantity(request, idMeasurement)`: Tato funkce načte odpovídající formulář ze souboru `forms.py`. Zároveň se stará o jeho částečnou validaci. Pokud funkce zaznamená v requestu metodu POST, přidá novou veličinu daného měření.
- `id_generator(size=10, chars=string.ascii_uppercase+string.digits)`: Funkce, generátor náhodných znaků v rozsahu deseti řádů.
- `displayMeasurement(request)`: Tato funkce vytvoří vyfiltrovaný seznam měření aktuálně přihlášeného uživatele.
- `displayMeCho(request, idMeasurement)`: Tato funkce vytváří seznamy veličin a jejich hodnot daného měření.
- `uniqueId_generator(size=20, chars=string.ascii_uppercase + string.digits)`: Funkce, generátor náhodných znaků v rozsahu dvaceti řádů.
- `userUniqueId(request)`: Tato funkce ukládá nově vygenerovaný unikátní klíč.
- `robot(request)`: Tato funkce slouží k rozpoznání a ověření dat, která jsou poslána jako součást http requestu. Právě na tuto funkci se musí odkazovat externí zařízení, jak jsem již psal výše v úvodu.
- `export_users_xls(request, idMeasurement)`: Pomocí této funkce je možné vyexportovat naměřená data do excelového souboru (.xls).

4.5 Sběr dat z externího zařízení

Jednou z výhod této aplikace je, dle mého názoru, právě možnost neustálého sběru dat z externího zařízení. I zde se jedná o bezpečnou část aplikace. Externí zařízení musí odesílat data v určitém formátu a musí projít ověřením ke kterému slouží unikátní klíč uživatele. Ten je generován jednoduchou funkcí `random.choice(chars)`. Unikátní klíč obsahuje dvacet znaků, konkrétně velkých písmen. To znamená, že máme-li počet prvků

(počet písmen v Anglické abecedě) 26 a z nich vybíráme kombinace 20 prvků, dostaneme až $1,9928 \cdot 10^{28}$ možných kombinací viz rovnice 4.1

$$X = 26^{20} = 1,9928 \cdot 10^{28}, \quad (4.1)$$

kde X je počet kombinací.

Šance že by teda někdo odhadnul takový klíč je velice malá. Pro zjednodušení ukládání hodnot pod určité veličiny též využívám generování určitého klíče nebo spíše ID. To znamená, že každá veličina při jejím vytvoření zároveň obdrží vlastní ID. Toto číslo má pouze deset znaků, taktéž velká písmena. Použijeme-li stejný vzorec jako u rovnice 4.1, získáme $1,4116 \cdot 10^{14}$ možných kombinací ID. Takový počet veličin nebude jen tak dosažen. V každém případě, je ale možné kdykoli počet kombinací navýšit například tím, že do počtu prvků přidáme další znaky.

4.5.1 Formulace http requestu

Aby funkce `robot` ve `views.py` rozpoznala http request, musí mít následující tvar¹⁴

```
time = datetime.datetime.now() #Je důležité ukládat časový záznam  
kdy byla hodnota změřena, nikoli, kdy byla přijata aplikací!  
requests.post('http://vojtechkadlec.pythonanywhere.com/easyMeasurement/robot/', data = {  
    'userId': '1',  
    'uniqueKey': 'OT5K73CBWNXWE8L6NW9L',  
    'uniqueIdQu': 'GPLV8D120P',  
    'value': namerena_hodnota,  
    'time': time,  
})
```

Funkce, která odešle http request se může lišit, záleží na použitém programovacím jazyku. Co by však mělo zůstat stejné je odkaz na server a parametry obsažené v parametru `data`. Z příkladu je jasné, jaké informace se mají doplňovat za jednotlivé parametry.

¹⁴ Příklad vychází z programového jazyka Python, což není podmínkou. Důležité je zachovat názvy parametrů obsažených v parametru `data`

4.6 Budoucnost aplikace

Aplikace EasyMeasurement zřejmě nebude fungovat pro širší veřejnost, je určena spíše pro soukromé experimenty jako například laboratorní měření prováděné na univerzitách. Ovšem nic nebrání tomu, aby byla aplikace používána v širším měřítku. V takovém případě by nebylo špatné aplikaci vylepšit o pár funkcí navíc. V záložce display by mohla být například možnost zobrazit naměřená data v podobě grafu a také by mohla přibýt další záložka s možností registrace uživatele.

Co se týká samotného designu aplikace, i zde by se mohlo zapracovat na drobných úpravách, které jsem opomněl z důvodu věnování se na samotnou funkčnost aplikace.

Nyní je aplikace napsaná v anglickém jazyce. Pro usnadnění práce při jejím používání, by mohla být aplikace přeložena například do českého nebo německého jazyka, mezi kterými by si mohl uživatel vybírat.

Pro omezení útoku „hrubou silou“ (brute force attack), tedy pro pokus o rozluštění určité šifry bez znalosti klíče k dešifrování, by se mohla v budoucnu navýšit velikost ID veličin a unikátních klíčů uživatelů. Momentálně je však velikost dostačující a je velmi nepravděpodobné, že by ID a klíče byly uhádnuty. Dalším zabezpečením proti takovému útoku by mohlo být automatické zablokování IP adresy, z které by bylo prováděno brute force testování klíčů.

V rámci kontroly robotů, externích zařízení, by se mohla aplikace rozrůst o nějaký sledovací systém, který by sledoval jejich chování. Například nepravdělné odesílání dat nebo jejich vynechávání. Součástí takového systému by mohlo být i nastavení počtu naměřených hodnot, a to z důvodu útoku, kdy by byl například uhádnut unikátní klíč či ID veličiny. V takovém případě by útočník nemohl zahltit server, na kterém aplikace běží.

5. Závěr

Cílem této práce bylo navrhnout a vytvořit webovou aplikaci pro ukládání dat z různých experimentů a to za pomoci frameworku Django. Tento cíl byl, dle mého názoru, úspěšně splněn. Webová aplikace nyní běží na serveru `vojtechkadlec.pythonanywhere.com` a je připravena k používání. Není však podmínkou, aby aplikace běžela jen na tomto serveru. Pokud by měla být aplikace používána pro veřejné nebo soukromé účely, stačí nakopírovat její zdrojové soubory na server nebo počítač, který podporuje python a Django.

Aplikace EasyMeasurement v tuto chvíli poskytuje vytvořeným uživatelům možnost založit nová měření nebo existující editovat spolu s veličinami a jejich hodnotami. Hotová naměřená data je možné exportovat do excelové tabulky. Pro uživatele, kteří chtějí využít možnosti ukládání dat do databáze přes HTTP, je vytvořen jednoduchý návod v podkapitole 4.5 Sběr dat z externího zařízení.

Framework Django je založen na programovacím jazyku Python a dokáže spolupracovat s několika typy databázových systémů. Pro vývoj aplikace, jakou je EasyMeasurement, je Django skutečně vhodný nástroj, který v mnoha ohledech ušetří programátorovi práci a čas. Pro programátory, kteří se s Djangem nesetkali, není těžké pochopit, jak Django funguje a jak se s ním pracuje. Je to dáno už jen tím, že na internetu existuje veliké množství zdrojů (především oficiální dokumentace), z kterých je možné čerpat.

Seznam použitých zdrojů

- [1] M. Valoušek, „Dokumentace,“ 2005-2017. [Online]. Available: <http://www.djangoproject.cz/>. [Přístup získán 8 4 2017].
- [2] P. Dvořák, „Úvod a instalace,“ 14 8 2009. [Online]. Available: <https://www.zdrojak.cz/clanky/django-uvod-a-instalace/>. [Přístup získán 8 4 2017].
- [3] M. Django, „Introducing django,“ 2016. [Online]. Available: <http://djangobook.com/introducing-django/>. [Přístup získán 8 4 2017].
- [4] Kasman, „XSS Úvodní stránka,“ 11 5 2016. [Online]. Available: <http://www.kasman.sk/xss>. [Přístup získán 22 5 2017].
- [5] J. Pejša, „Co je Cross-Site Request Forgery a jak se mu bránit,“ 2008. [Online]. Available: <https://www.zdrojak.cz/clanky/co-je-cross-site-request-forgery-a-jak-se-branit/>. [Přístup získán 15 4 2017].
- [6] Django project, „Security,“ 2005-2017. [Online]. Available: <https://docs.djangoproject.com/en/1.10/topics/security/>. [Přístup získán 8 4 2017].
- [7] PBWCZ, „Clickjacking,“ 18 9 2016. [Online]. Available: <http://pbwcz.cz/Technika%20hackeru/Jednotlive%20techniky/clickjacking.html>. [Přístup získán 22 5 2017].
- [8] Mozilla wiki, „Security, CSP,“ 5 8 2012. [Online]. Available: <https://wiki.mozilla.org/Security/CSP>. [Přístup získán 15 4 2017].
- [9] Django project, „misc-distributions,“ 2017. [Online]. Available: <https://docs.djangoproject.com/en/dev/misc/distributions/#misc-distributions>. [Přístup získán 8 4 2017].
- [10] D. Gourley a B. Totty, „HTTP: the definitive guide,“ Sebastopol, O'Reilly, 2002, pp. 3,4.
- [11] J. Švec, Létající cirkus, http://ii.iinfo.cz/r/old/data/letajici_cirkus.pdf, 2003.

- [12] Pestujemeweb, „Co je css,“ 2017. [Online]. Available: <http://www.pestujemeweb.cz/obsah/css/co-je-css.php>. [Přístup získán 29 4 2017].
- [13] J. Kosek, „Historie a vývoj html,“ 1997-2014. [Online]. Available: <http://htmlguru.cz/uvod-historie.html>. [Přístup získán 15 4 2017].
- [14] Globtrex, „INFORMACE O BEZPEČNOSTI,“ 2017. [Online]. Available: <https://www.globtrex.com/cz/informace-o-bezpe%C4%8Dnosti>. [Přístup získán 22 5 2017].
- [15] D. Staňková, „JavaScript,“ [Online]. Available: <http://jakpsatwww.webnode.cz/javascript/>. [Přístup získán 22 5 2017].
- [16] Adaptic, „Cookies,“ 2017. [Online]. Available: <http://www.adaptic.cz/znalosti/slovnicek/cookies/>. [Přístup získán 8 4 2017].
- [17] D. N. k. ČR, „KTD - Úplné zobrazení záznamu,“ 2014. [Online]. Available: http://aleph.nkp.cz/F/?func=direct&doc__number=000000568&local__base=KTD. [Přístup získán 8 4 2017].
- [18] Hosting, „Co je to subdomena,“ 2017. [Online]. Available: <http://www.hosting-zdarma.cz/napoveda-co-je-to-domena-i-ii-iii-radu-nebo-subdomena>. [Přístup získán 15 4 2017].
- [19] It-slovník, „Token,“ 2017. [Online]. Available: <https://it-slovník.cz/pojem/token>. [Přístup získán 15 4 2017].
- [20] Managementmania, „middleware,“ 19 10 2016. [Online]. Available: <https://managementmania.com/cs/middleware>. [Přístup získán 15 4 2017].
- [21] MySQL, „Doc,“ 2017. [Online]. Available: <https://dev.mysql.com/doc/>. [Přístup získán 15 4 2017].
- [22] Sadowski, „SQL-PRVN,“ [Online]. Available: <http://books.fs.vsb.cz/SQLReference/Sadovski/SQL-PRVN.HTM>. [Přístup získán 29 4 2017].
- [23] W3, „CSS,“ 2017. [Online]. Available: <https://www.w3.org/Style/CSS/>. [Přístup získán 29 4 2017].

Seznam použitých zkratek

- BSD licence - Berkeley Software Distribution, str. 3
- DRY - Don't repeat yourself, str. 5
- ORM - Object-Relation-Mapping, str. 5
- RSS - Rich Site Summary, str.5
- MTV - Model-Template-View, str. 5
- XSS - *Cross-site scripting*, str. 6
- WWW - World Wide Web, str. 6
- URL - Universal Resource Locator, str. 6
- CSRF - *Cross-Site Request Forgery*, str. 7
- SQL - *Structured Query Language*, str. 8
- CSS - *Cascading Style Sheets*, str. 12
- HTML - HyperText Markup Language, str. 15
- JPEG - *Joint Photographic Experts Group*, str. 15
- MPEG - *Moving Picture Experts Group*, str. 15
- WAV - *Waveform audio file format*, str. 15
- HTTP - Hypertext Transfer Protocol, str. 15
- ID - IDentification, str. 33

Seznam obrázků

- Obrázek 2.1 – Struktura projektu
- Obrázek 2.2 – Schéma views (převzato od Heroku.com)
- Obrázek 3.1 – Python
- Obrázek 3.2 – MySQL
- Obrázek 3.3 – Propojení HTML a CSS
- Obrázek 3.4 – Ukázka CSS
- Obrázek 3.5 – HTML příklad
- Obrázek 3.6 – Bootstrap
- Obrázek 3.7 – Sublime Text
- Obrázek 4.1 – Prostředí aplikace
- Obrázek 4.2 – Schéma databáze
- *Obrázek A.1 – Ukázka úvodní stránky*
- *Obrázek A.2 – Ukázka editační stránky*
- *Obrázek A.3 – Ukázka display stránky*

Seznam tabulek

- Tabulka 3.1 – Uživatelé

Příloha A

Ukázky vzhledu aplikace

Easy Measurement

If you want use this app, you have to log: [Log In](#)



Hello, this is application for easy measurement. You can set up new measurement and their quantity. If you want change measurement, click to edit. If you want only display your measurement, click Display.

--Legend-----

Default

Primary

Success

Info

Warning

Danger

Obrázek A.1 – Ukázka úvodní stránky

Easy Measurement

Your are: **Signed** / **Logout**

Home

New

Edit

Add value

Display

Unique key

Choose your edit in **Temperature**

--- Measurement -----

Change the name of measurement

Delete measurement

--- Quantity -----

Add new quantity

Edit quantity

--- Values -----

Remove values

Vojtech Kadlec copyright 2017

Obrázek A.2 – Ukázka editační stránky

Easy Measurement

Your are: [Signed](#) / [Logout](#)

Name of measurement: Temperature

- Home
- New
- Edit
- Add value
- Display
- Unique key

List value for quantity: [Export](#)

T2 [K], Unique ID: 7WNGM5

25	2017-05-02 06:07:12
36	2017-05-02 06:07:14

T3 [K], Unique ID: V203NF

	Time of Measurement
45	2017-05-02 06:07:22
69	2017-05-02 06:07:24

Obrázek A.3 – Ukázka display stránky