

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra technologií a měření

DIPLOMOVÁ PRÁCE

Elektronická prezentace experimentu

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. David KRUML**

Osobní číslo: **E16N0013P**

Studijní program: **N2612 Elektrotechnika a informatika**

Studijní obor: **Komerční elektrotechnika**

Název tématu: **Elektronická prezentace experimentu**

Zadávací katedra: **Katedra technologií a měření**

Z á s a d y p r o v y p r a c o v á n í :

Cílem je vytvoření části informačního systému, přebírajícího aktuální data z reálného technického experimentu.

1. Prostudujte rozhraní pro komunikaci s přístroji.
2. Vytvořte jeho programovou obsluhu s ukládáním dat do databáze.
3. Prostudujte možnosti zobrazení dat na webové stránce, výstupy tabulkové s filtrací, grafické, download dat, atd.
4. Vytvořte program pro efektivní prezentaci dat z experimentu pro webový server.
5. Popište vlastnosti vytvořeného systému, prostudujte a sepište jeho slabá místa a možné důvody poruchy.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah kvalifikační práce: **40 - 60 stran**

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

Seznam odborné literatury:

- 1. PHP & MySQL: Novice to Ninja: The Easy Way to Build Your Own Database Driven Website; Kevin Yank; SitePoint; 2012; ISBN 145719192X, 9781457191923**

Vedoucí diplomové práce:

Ing. Richard Linhart, Ph.D.

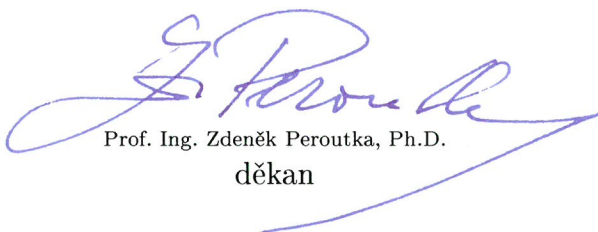
Katedra aplikované elektroniky a telekomunikací

Datum zadání diplomové práce:

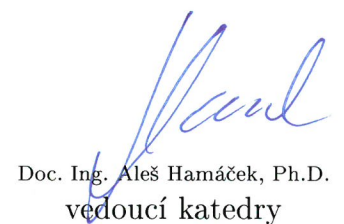
5. října 2018

Termín odevzdání diplomové práce:

30. května 2019


Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan




Doc. Ing. Aleš Hamáček, Ph.D.
vedoucí katedry

V Plzni dne 5. října 2018

Abstrakt

Předkládaná diplomová práce je zaměřena na vytvoření části informačního systému přebírající aktuální data z technického experimentu s názvem „VZLUsat-1“. Přebíraná data jsou informačním systémem zobrazována v podobě grafů, tabulek s možností filtrace apod. Součástí je i administrátorské rozhraní pro operátory kvůli možnosti nastavení projektu, databáze a výstupů. V informačním systému je počítáno i s možností napojení na databázi dalšího/dalších technických projektů – případných následovníků projektu „VZLUsat-1“. Součástí práce je nejen samotné řešení, ale i návrh informačního systému s jeho následnou dokumentací a analýza předchůdce tohoto systému, který lze nalézt na adrese <https://www.pilsencube.zcu.cz/vzlusat1/>. V analýze jsou zhodnoceny silné a slabé stránky současného systému, ze kterých poté vychází návrh systému nového – realizovaného v poslední části této diplomové práce.

Klíčová slova

MySQL, PHP, Vzlusat, webová aplikace, grafy, informační systém, zobrazení dat, webový server, Nette Framework, družice, pozemní stanice, backend, frontend.

Abstract

The main purpose of this diploma thesis is to create part of an information system taking over data from technical experiment called „VZLUsat-1“. The data are projected as graphs, tables and tables with possibility of filtering data. Part of this program is also an admin interface for operators of the ground station. Information system is developer to be connected to the another database also. The thesis consists of the developed program, its proposal with documentation and also analysis of currently used webpage placed on URL: <https://www.pilsencube.zcu.cz/vzlusat1/>. In analysis are discussed advantages and disadvantages of the current program. This analysis is used to create proposal of new program after that.

Key words

MySQL, PHP, Vzlusat, web application, graph, information system, data projection, web server, Nette Framework, satellite, ground station, backend, frontend.

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

.....

podpis

V Plzni dne 26.5.2019

David Kruml

Poděkování

Tímto bych rád poděkoval vedoucímu diplomové práce Ing. Richardovi Linhartovi, Ph.D. za cenné profesionální rady, připomínky a metodické vedení práce.

Obsah

OBSAH	5
ÚVOD	6
SEZNAM SYMBOLŮ A ZKRATEK	7
1 ANALÝZA SOUČASNÉ APLIKACE	8
1.1 POPIS APLIKACE	8
1.2 SILNÉ A SLABÉ STRÁNKY SOUČASNÉHO ŘEŠENÍ.....	9
2 NÁVRH NOVÉ APLIKACE	11
2.1 CÍLE NAVRŽENÉHO ŘEŠENÍ.....	11
2.1.1 Cíle z pohledu návštěvníka webu.....	11
2.1.2 Cíle z pohledu operátora	12
2.1.3 Cíle s ohledem na MySQL databázi.....	13
2.2 FRONTEND – VZHLED A ZOBRAZENÍ DAT	14
2.2.1 Pozadí.....	14
2.2.2 Hlavička a patička.....	14
2.2.3 Menu.....	15
2.2.4 Zobrazení dat.....	16
2.3 BACKEND – VZHLED A MODULY	20
2.3.1 Vzhled administračního rozhraní.....	20
2.3.2 UserModule	21
2.3.3 AdminModule	22
2.4 NETTE FRAMEWORK	24
2.5 DATABÁZE	25
3 REALIZACE NAVRŽENÉHO ŘEŠENÍ	26
3.1 ADRESÁŘOVÁ STRUKTURA WEBOVÉ APLIKACE.....	26
3.1.1 Adresář LOG	27
3.1.2 Adresář SQL	27
3.1.3 Adresář TEMP.....	27
3.1.4 Adresář WWW	28
3.1.5 Adresář APP.....	29
3.2 FRONTEND.....	34
3.2.1 Frontend - vzhled.....	34
3.2.2 Frontend – zobrazení dat.....	37
3.3 BACKEND.....	43
3.3.1 Backend – přihlášení	44
3.3.2 Backend – vzhled rozhraní.....	45
3.3.3 Backend – tvorba položek v menu.....	46
3.3.4 Backend - uživatelé.....	48
3.3.5 Backend - databáze.....	50
3.3.6 Backend – návštěvníci	51
3.3.7 Backend – storage, homepage	52
3.4 VÝHODY A NEVÝHODY APLIKACE, SROVNÁNÍ S PŘEDCHOZÍM WEBEM	53
3.4.1 Backend	53
3.4.2 Zobrazení pro mobilní zařízení.....	54
3.4.3 Napojení na více databázi	54
3.4.4 Zajištění běhu databáze	55
3.4.5 Zobrazení dat na webové stránce	56
3.4.6 Fotogalerie	58
ZÁVĚR	59
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	1

Úvod

Předkládaná práce je zaměřena na vytvoření části informačního systému, přebírajícího aktuální data z reálného technického experimentu.

Práce se skládá ze tří částí. V první části je analyzován původní stav aplikace. Je zde přiblížen technický experiment, toky informací, propojení komponent, princip a platformy, na kterých jsou realizované jednotlivé části informačního systému. V této části práce jsou popsány slabé a silné stránky současného informačního systému.

Z analýzy v první části práce vychází část druhá – návrh nového řešení. Tato část se zaměřuje na návrh nového řešení aplikace. Bere ohled především na vylepšení slabých stránek na zachování kladných vlastností původního řešení. Dále jsou v této části vybrány platformy pro programování, které jsou následně porovnány s původním řešením. V návrhu je počítáno i s optimalizací webové aplikace z hlediska spolehlivosti a zobrazení pro mobilní zařízení, s ohledem na zachování komplexnosti a důležitých parametrů aplikace.

Dalším parametrem, který návrh zohledňuje, je i možnost budoucího napojení aplikace na databázi navazujícího technického projektu. To obnáší navržení a vytvoření programové struktury tak, aby bylo možné v budoucnu aplikaci efektivně spravovat za pomoci co nejmenšího množství zásahů operátora.

Ve třetí části této práce je vytvořena samotná webová aplikace. Aplikace je tvořena na základě předchozí analýzy, tedy tak aby přinášela zlepšení popsaných funkcí. V rámci realizace je také sepsána dokumentace, jaké možnosti aplikace přináší, jak s aplikací pracovat a jaké ovládací prvky jsou k dispozici pro návštěvníka stránek. V rámci této části je i hlouběji popsáno administrativní rozhraní určené pro operátory. Kromě popisu backend a frontend uživatelského rozhraní aplikace je zde popsána i programová část – adresářová struktura aplikace, routování, připojení k databázi, konfigurační soubory, a coding standard týkající se namespace, modulů, objektů aj. Tato část je uvedena zejména s ohledem na možnosti budoucích úprav a rychlejší pochopení struktury aplikace pro programátora.

Seznam symbolů a zkratek

<i>AJAX</i>	Technologie umožňující překreslit pouze vybranou část webové stránky
<i>CSS</i>	Programovací jazyk využívaný pro úpravu grafické podoby HTML elementů
<i>HTML</i>	Značkovací programovací jazyk, který je používán pro psaní webových aplikací
<i>IDE</i>	Prostředí pro vývoj aplikace programátorem
<i>FTP</i>	Protokol starající se o přenos souborů
<i>ACL</i>	Seznam povolených přístupů – ke kterým souborům uživatel smí či nesmí přistupovat
<i>Backend</i>	Administrátorské rozhraní aplikace – přístupné pouze přihlášenému uživateli
<i>Frontend</i>	Rozhraní aplikace určené pro všechny návštěvníky webové stránky – nevyžaduje přihlášení
<i>Grid</i>	Tabulka pro zobrazení dat s možností filtrace, řazení a stránkováním
<i>MySQL</i>	Programovací jazyk určený pro přístup k databázím

Analýza současného stavu

1 Analýza současné aplikace

V rámci analýzy původního řešení webové aplikace je třeba zmínit jeho kladné stránky a následně tyto stránky promítnout i do nové verze aplikace. Následně bude zmíněno, v jakých partiích je třeba aplikaci zlepšit, co bude potřeba přidat a do jaké míry toto ovlivní vnímání operátora či uživatele. Na současné stránce byla provedena analýza několika dílčích částí. Jedná se zejména o informační hodnotu stránek ve spojitosti s přehledností, dále funkční stránku aplikace, logický návrh jednotlivých komponent, umístění komponent, zabezpečení programu, struktura databáze, připojení k databázi a vzhled. Z analýzy budou následně vyvozeny požadavky na nový program.

1.1 Popis aplikace

Úvodem této diplomové práce je třeba nastínit základní popis pozemní stanice a způsobu, jakým komunikuje s družicí VZLUsat-1. Pozemní stanice se nachází v budově Fakulty elektrotechnické Západočeské Univerzity v Plzni. Na střeše budovy je umístěna anténa, která slouží ke komunikaci s družicí. K přenosu dat mezi družicí a pozemní stanicí dochází vždy při přeletech družice nad horizontem. K výpočtu předpokládaného času přeletu slouží predikční software „Predict“. Na základě reálných dat o přeletech je následně předpověď korigována. Na základě této předpovědi je pomocí rotátoru otáčena i anténa na střeše pozemní stanice. Průběžně dochází ke snaze o kontakt s družicí, kontrole jeho stavu a plánování nahrávání dat. V době přeletu jsou vykonány upload příkazy a download příkazy, ty mohou být ovlivněny požadavky operátora stanice. Data jsou ukládána do MySQL databáze, následně jsou zobrazována pomocí webové stránky umístěné na adrese <https://www.pilsencube.zcu.cz/vzlusat1/>. Vzhledem k vytížení databáze jsou data duplikována do druhé databáze. Z druhé databáze jsou teprve poskytována webové aplikaci. Je tak zajištěn bezproblémový chod hlavní databáze.

Současná webová stránka je realizována pomocí 3 základních programovacích jazyků. Jedná se o PHP pro logickou část aplikace, HTML pro vzhled a jazyk MySQL pro komunikaci s databází. Účelem webu je poskytovat informace o přenesených datech a samotném projektu pozemní stanice pro širokou veřejnost (vědecké týmy, radioamatéry, laiky) a možnost nastavovat vybrané parametry pro operátora pozemní stanice.

1.2 Silné a slabé stránky současného řešení

Mezi kladné stránky současné aplikace patří struktura databáze. Jednotlivá data jsou logicky členěna do tabulek, tabulky jsou přehledné, jsou velmi podrobně zpracovány datové typy sloupců. Počet sloupců v jednotlivých tabulkách není příliš vysoký, což umožňuje, aby byla aplikace robustní.

Vysoký počet sloupců v rámci jedné tabulky by následně mohl způsobovat nepřehlednost pro uživatele, který do databáze přistupuje či pro programátora. Těmto problémům se návrh současné aplikace vhodně vyhnul. Na struktuře databáze lze nalézt jen drobné nešvary. Nejsou příliš používány indexy, primární klíče a cizí klíče nad nejčastěji prohledávanými sloupci. Takto opatřené sloupce mají prioritu při prohledávání pomocí jazyka MySQL, vyhledávání je tudíž rychlejší. Druhým nalezeným nedostatkem je způsob inkrementace záznamů v databázi. U několika tabulek chybí sloupec, který by měl zápis takzvaného „auto-inkrementu“ nad unikátním identifikátorem. U tabulek, kde je toto definováno, je mnohdy unikátním identifikátorem nevhodný datový typ sloupce. Nejčastěji se jedná o typ *datetime*. Při prohledávání databázové tabulky je z výkonových důvodů vhodné mít definovaný 1 sloupec jako prostý unikátní identifikátor s nastaveným auto-incrementem. Ideálně datového typu *integer*. Procházení tabulek přes typ *integer* je vždy rychlejší, navíc je výše zmíněnými parametry zajištěna přehlednost a unikátnost záznamu. Jako další bod vhodný k vylepšení lze zmínit připojení k databázi. To je v současné práci realizováno poněkud složitým a nepřehledným kódem. Připojení k databázi bude třeba definovat a realizovat pomocí objektů, ideálně na 1 místě v kódu, aby byla zajištěna snazší možnost propojení na více databází (tato problematika bude rozebrána v dalších partiích práce).

Informační hodnota stránky současného programu je na velmi dobré úrovni, boxy se zobrazovanými daty působí přehledně, jsou v nich uvedena důležitá data, návštěvník není zatěžován nepodstatnými informacemi. Tato vlastnost by rozhodně měla být zachována. Jediným nešvarem v rámci tohoto bodu jsou formuláře na úvodní straně, které umožňují operátorovi pomocí checkboxů ovlivňovat data v databázi. Ačkoliv jsou pole znepřístupněna běžnému návštěvníkovi, stránka pak působí nevěrohodně. Může tak lákat k napadení a ovlivnění dat v databázi pomocí různých technik, například cross-site scriptingu. Tyto části stránek bude potřeba skrýt běžnému uživateli.

Z hlediska funkčního je třeba stránkám vytknout pravidelné aktualizování celého webu a přesměrování na úvodní stránku. Toto je prováděno především kvůli aktuálním obrázkům ze střechy pozemní stanice zobrazující anténu využívanou ke komunikaci s družicí. Tato funkčnost není nezbytná a navíc příliš obtěžuje uživatele. Příkladem budiž situace, kdy uživatel přejde na jinou než domovskou podstránku webu, prohlíží si zde data a zničehonic je automaticky přesměrován na domovskou stránku. Tato vlastnost je nežádoucí, matoucí a spíše odrazuje uživatele od opětovné návštěvy stránek. Absence odlišné URL adresy pro jednotlivé podstránky je taktéž nevhodná. Není možnost odkazovat na konkrétní stránky webu, prohlížeče nemohou indexovat web na základě obsahu aj. V následující verzi bude třeba implementovat routování URL adresy.

Posledním zhodnoceným parametrem je vzhled aplikace a dat. V desktopové verzi je aplikace přehledná. Ale absence mobilní verze webu je jeho velkým nedostatkem. Zvláště v dnešní době, kdy vysoký podíl návštěvnosti stránek je tvořen návštěvami z přenosných zařízení. Implementace responzivního zobrazení webu bude jedním z nosných pilířů této diplomové práce.

S vzhledem souvisí i grafická prezentace dat. V rámci jedné z podstránek jsou implementovány grafy. Tato podstránka se poměrně dlouho načítá. Pomalé načítání je způsobeno dobou načtení nástroje pro zobrazení grafů (tato doba odezvy bude přibližně stejná u všech nástrojů, takže ji webu nelze vyčítat). Dalším faktorem, který způsobuje dlouhou odezvu stránky, je načítání 6 grafů zároveň a také načítání dat z databáze už při prvním načtení podstránky. Zejména druhý a třetí dílčí čas lze výrazně ovlivnit. Zobrazením pouze jednoho grafu na stránce s možností výběru vykreslené závislosti veličin bude pro uživatele pohodlnější a přehlednější. Zároveň načítání dat až na požadavek uživatele bude působit lépe. V mobilním zařízení navíc nebude návštěvník ochuzen o data bez vlastního přičinění a vědomí. Posledním diskutovaným parametrem grafů budou samotná data. V současné verzi programu není možné zpětně zobrazovat historická data. Tento aspekt bude třeba taktéž vyřešit. Samozřejmě s ohledem na neomezení chodu MySQL databáze. Tento bod bude v práci rozebrán v návrhu nového řešení.

Poslední komponentou, která bude hodnocena, je fotogalerie. Obrázky umístěné ve fotogalerii nelze zvětšit. Jsou vidět pouze jejich miniatury. V rámci realizace nového webu bude potřeba implementovat i možnost zobrazení obrázků v plné velikosti.

2 Návrh nové aplikace

V následující části této diplomové práce bude navržen program, který bude implementovat poznatky z předcházející analýzy současného stavu webové stránky. Důraz bude kladen na komplexnost řešení, spolehlivost a zabezpečení aplikace a výběr vhodných platform pro realizaci jednotlivých komponent aplikace. V rámci návrhu bude důležité definovat cíle, kterých by mělo výsledné řešení dosahovat. Ty budou vycházet z předešlé analýzy. Bude potřeba nejen zachovat dostačující prvky a výhody původní ho řešení, ale i optimalizovat jeho nedostatky. V rámci zachování dostačujících prvků budou navržena jejich vylepšení při zachování výkonových požadavků na rychlost webu.

2.1 Cíle navrženého řešení

Při návrhu nového řešení aplikace je třeba rozdělit cíle do následujících kategorií. Těmi jsou cíle pro běžného návštěvníka webu, operátora a zajištění bezproblémového chodu MySQL databáze.

2.1.1 Cíle z pohledu návštěvníka webu

Běžný návštěvník webové stránky má v dnešní době několik základních požadavků. Mezi tyto základní požadavky patří především přehlednost webové stránky, její rychlost, responzivita, dostupnost a především informační hodnota. Uživatel požaduje dostat informace, které si žádá, za vyvinutí co nejmenší množství úsilí. Uspokojit uživatele ve všech uvedených parametrech zároveň přináší vysoké nároky na programátora. Je potřeba správně rozvrhnout algoritmy jednotlivých funkcí, vybrat platformy, pomocí nichž lze cílů dosáhnout a nakonec všechny platformy propojit v jeden funkční celek.

Aplikace by měla umět zobrazit data přenášená z družice VZLU_{sat-1}. Zobrazení by mělo být přehledné, jednoduché, s co nejnižšími požadavky na rychlost odezvy. V rámci této práce bude potřeba zobrazovat data několika způsoby. Většina způsobů je již využita v původní verzi webové aplikace. Nicméně jejich provedení je značně omezené – především z výkonových důvodů. Mezi způsoby zobrazení dat patří jednoduché HTML tabulky pro zobrazení jednotek záznamů z databázových tabulek, složitější tabulky typu grid s možnostmi filtrace a řazení záznamů. Pro zobrazování tabulek typu grid bude důležité vybrat jejich vhodnou implementaci. V gridech budou načítány řádově tisíce až

desetitisíce záznamů, bude tedy potřeba implementovat variantu, která podporuje stránkování a postupné načítání pouze té stránky záznamů, kterou si uživatel přeje zobrazit. Ideálním případem je načítání dalších stránek pomocí technologie zvané AJAX [1], která umožňuje překreslit jen vybranou část webové stránky bez nutnosti zatěžovat server požadavkem na překreslení kompletní stránky. Využití AJAX technologie bude pohodlné i pro oko uživatele. Dále bude potřeba zobrazovat data pomocí grafů. Graf jedna z podob zobrazení dat, které nejlépe vyjadřují trendy hodnot fyzikálních veličin. Původní web tuto zobrazovací techniku již využívá, avšak s výraznými omezeními. Bude potřeba vylepšit vykreslování grafů nejen z pohledu možnosti zobrazovat historická data, ale i z pohledu možností a nástrojů k úpravě výsledné křivky. Určitě bude potřeba implementovat metodu, která umí přibližovat či oddalovat oblast grafu a následně přepočítat hodnoty na osách X a Y do vhodného měřítka.

Dalším parametrem, který bude návštěvníka zajímat, bude možnost zobrazit na web na mobilních zařízeních. Současná podoba webové aplikace zobrazuje v mobilním zařízení stejnou verzi jako na desktopovém zařízení. To způsobuje obtíže pro uživatele při zobrazování v mobilním zařízení. Zejména protože k zobrazení kompletní stránky je potřeba přejíždět po displeji či stránku nadměrně oddalovat a přibližovat. Je tedy potřeba prozkoumat pokročilé techniky programovacího jazyka CSS, které s responzivitou HTML elementů umí pracovat a následně navrhnout řešení, které tyto techniky jazyka CSS implementuje.

2.1.2 Cíle z pohledu operátora

Operátor webových stránek má požadavky shodné s běžným uživatelem. Tedy responzivita, přehlednost, rychlost a zobrazení široké škály dat. Oproti běžnému uživateli má ovšem na starost správu webové aplikace. Ta by měla být především jednoduchá, nenáročná na čas a podporovat výše zmíněné parametry. Dalším požadavkem správce webové stránky je, aby stránka příliš nezatěžovala MySQL databázi z hlediska jejího výkonu. Tento požadavek bude stěžejní a bude se objevovat napříč celou prací. Databáze je totiž příjemcem dat z družice VZLUsat-1 a pro tyto účely musí být vždy v pohotovosti. Nesmí nastat stav, kdy požadavky uživatelů na veřejně přístupné části webu zatíží databázi natolik, že bude vyřazena z provozu. Bude proto potřeba navrhnout účinná omezení uživatelských požadavků na zobrazení dat z databáze.

Cílem práce by rozhodně mělo být i přemístění formulářových a zobrazovacích částí webu, které jsou dostupné pouze operátorovi „na pozadí“, aby ho nemohli vidět běžní návštěvníci stránek. Jedním z důvodů je i předcházení možností napadení aplikace (co uživatel nevidí, to ho neláká). Dalším parametrem, který je důležitý z pohledu operátora, je možnost napojení aplikace na jinou databázi. V současné době totiž probíhají práce na novém technickém projektu, který by měl přinést následníka družice VZLUsat-1. To přináší i požadavek na budoucí zobrazení dat z této družice. Bude tedy třeba navrhnout a vytvořit program tak, aby nejen umožňoval napojení na více databází, ale především, aby této funkce bylo dosaženo jednoduše, bez zbytečných obtíží a úprav v kódu programu.

Všechny výše zmíněné podmínky vedou k nutnosti navrhnout backend aplikace. V rámci tohoto rozhraní bude možné provádět změny, editovat, nastavovat důležité stavy do databáze, zkrátka spravovat celý web. Správcovské rozhraní bude navíc potřeba skrýt za uživatelské přihlášení. Bude tak vystavena pomyslná bezpečnostní zeď pro správce webu. Toto řešení pochopitelně vede k nutnosti návrhu administrace uživatelských přístupů a následné kontrole přístupů oprávněných uživatelů do jednotlivých částí správcovského rozhraní. Bude tedy také potřeba navrhnout a implementovat ACL část aplikace pro backend.

2.1.3 Cíle s ohledem na MySQL databázi

Databáze MySQL má stejně jako jiné databázové systémy své limity. Je potřeba respektovat tyto limity, ale zároveň posunout schopnosti grafického zobrazování dat na vyšší úroveň (oproti současné webové stránce). K tomu bude potřeba zajistit kontrolu aktivity návštěvníků webu, jejich požadavků na databázi. Následně bude důležité mít možnost aktivitu uživatelů regulovat vhodnými omezeními. K regulaci aktivity uživatelů bude ideální využít správcovské rozhraní, a poskytnout tuto funkčnost danému administrátorovi. Zároveň bude potřeba přizpůsobit databázové tabulky požadavkům na přístup k řádově tisícům až desetitisícům záznamů najednou. Bude taktéž nutné jednoduše a unikátně identifikovat každého návštěvníka stránek, aby neodcházelo ke zbytečným komplikacím pro administrátory webu.

2.2 Frontend – vzhled a zobrazení dat

Pro dosažení cílů stanovených v předchozím bodě této diplomové práce je potřeba nejdříve udělat grafický návrh veřejně dostupné části aplikace. Ten bude sestávat ze dvou částí, tedy z verze pro stolní zařízení a verze pro mobilní zařízení. Obě části by měly mít podobné grafické prvky, aby vzhled aplikace vypadal jednotně. Je tedy třeba stanovit vzhled pozadí, hlavičky, patičky, hlavní nabídky a obsahu jednotlivých podstránek.

2.2.1 Pozadí

Pozadí webu je vhodné udělat jednoduše, bez výrazných a kontrastních grafických prvků, které neodvádí pozornost od obsahu webových stránek. Přesto by ale bylo dobré, aby pozadí nezapadlo do „šedého průměru“ a mělo alespoň nějaký grafický prvek, který mu přidá moderní vzhled. Ideální variantou pro takové případy je využit barevného přechodu v kombinaci s obrázkem na pozadí. S ohledem na výše zmíněné nekontrastní barvy vyžaduje pozadí podobné odstíny barev. K tématům družice, oběžné dráhy aj. se nabízí využití tmavých barev působících příjemně pro oko, bez následného odvádění pozornosti od zobrazených dat. Barevný přechod tedy bude vytvořen lineární ve vertikálním směru začínající od černé barvy, přecházející přes odstíny tmavě zelené až tmavě modré, končící znovu barvou černou, aby stránky vypadaly symetricky. Zvolenými přechodovými barvami jsou (uvedeno v hex kódech) #000000, #212121, #037263, #212121, #000000. Vertikální směr barevného přechodu vypadá velmi dobře zejména při scrollování v mobilních zařízeních. K náhledu přechodu při vývoji bude využita online služba Gradient Editor ColorZilla [2].

2.2.2 Hlavička a patička

Do záhlaví stránky bude umístěn obrázek s tematikou vesmíru. Pro tyto účely byl vybrán banner, který je na současné verzi webové stránky pozemní stanice VZLUsat-1. Díky svým tmavým okrajům zapadne do vzhledu stránek a dokreslí jejich moderní vzhled. Obrázek bude překrývat nadpis webu „VZLUsat Ground Station“. Text nadpisu bude, stejně jako veškerý text na veřejně dostupné části stránek, v bílé barvě pro dosažení dostatečné čitelnosti. V mobilní verzi webu bude tento obrázek skryt. Důvodem skrytí je šetření dat uživatelů. Načtení a zobrazení stránky totiž mimo jiné vyžaduje i stažení všech obrázků na ní zobrazených. Tato vlastnost a následné urychlení načítání uleví zejména návštěvníkům stránek, nebudou tak nuceni obrázek zobrazovat na každé podstránce.

Design patičky bude jednotný pro celý frontend. Bude vycházet z navrženého pozadí webu. V tom je konec stránky vykreslen jako přechod v odstínech černé barvy. Nabízí se tedy udělat celou patičku jako horizontální černý pruh fixní výšky. Protože v hlavičce webu bude umístěn obrázek a menu pro běžného návštěvníka, je vhodné umístit tlačítka pro vstup do administrace a přihlášení uživatele spíše do patičky webu, aby nezpůsobovala nepřehlednost hlavičky webu. Tlačítko pro přihlášení bude umístěno v pravém rohu patičky. Pro přihlášeného uživatele se bude na místě přihlašovacího tlačítka zobrazovat jeho jméno a příjmení. Tlačítko pro vstup do administrace webu bude umístěno v levém rohu patičky.

2.2.3 Menu

V hlavní nabídce webu bude 5 záložek. Všechny záložky budou odpovídat původnímu webu, s tím rozdílem, že některé záložky budou přesunuty do administrace stránek. Záložky, které v menu zůstanou, jsou: „COMM Info“, „Predict“, „WOD OBC“, „Radio amateur“ a „About us“. Vzhledem k obrázkovému banneru umístěnému v hlavičce webu je nasnadě vybrat horizontálně orientované menu. To by ovšem působilo problémy při zobrazení v mobilním zobrazení. Bude tak nutné vzhled menu pro mobilní zařízení upravit. V tomto případě bude orientace menu změněna na vertikální. Efektu změny vzhledu a orientace lze dosáhnout pomocí CSS techniky zvané **@media**. Ta umožňuje nastavit odlišné kaskádní stylování pro různé šířky displeje. Pokud tedy šířka displeje bude menší než 768 pixelů, bude zobrazeno vertikálně orientované menu, v opačném případě bude menu orientováno horizontálně.

Důležité je také stanovit barvy hlavní nabídky. V desktopových zařízeních se bude jednat o lineární horizontálně orientovaný přechod odstínů zelené barvy, které budou podobné barvám pozadí webu. Nicméně pro zvýraznění hlavní nabídky bude barva více kontrastní. Konkrétně se bude jednat o tyto odstíny (uvedeno v hex kódech): #377a61, #1f7f5c, #037263, #00705f, #1f7f5c, #387c63. V mobilním zobrazení bude nabídka ve výchozím nastavení skryta, aby nenutila uživatele nadbytečně posunovat stránku. Místo ní bude v horní části stránky zobrazeno tlačítko, takzvaný „Burger button“. Po kliknutí na tlačítko se zobrazí vertikální nabídka překrývající většinu displeje zařízení. Nepřekrývající část stránky bude prosvítat přes šedou průhlednou clonu. Samotné menu bude v tomto případě taktéž šedé (hex kód #383838), jeho okraj bude stínován.

2.2.4 Zobrazení dat

Je potřeba připravit šablony pro zobrazování dat. K tomuto účelu bude vhodné využít třídu **box** z balíku CSS služeb Admin LTE 2 [3]. Tato třída je responzivní a zobrazuje veškerá data černým textem na čisté bílé pozadí. Pro účely této práce bude třída přetížena a budou jí přidány následující 2 kaskádní styly: *border-radius: 2px;* a *overflow-x: auto;*. První styl zajistí boxu zaoblené rohy, druhý mu zajistí přidání vodorovné lišty pro přetažení v případě příliš široké oblasti dat. Bude tak zamezeno přetečení dat mimo oblast zobrazeného boxu. Následně bude přetížena i podřazená třída **box-header**, které bude přidán kaskádní styl *font-weight: bold;*. Tím bude dosaženo zvýraznění textů v záhlaví boxů.

Pro umístění boxů vedle sebe v zobrazení pro stolní počítače a seřazení boxů pod sebe v mobilním zobrazení stránek bude využito CSS techniky **@media**. S tím rozdílem, že tentokrát bude využita zprostředkovaně, přes HTML & CSS framework Bootstrap 3 [4]. Třídy definované v tomto frameworku v kombinaci s třídami definovanými ve službě Admin LTE2 [3] poskytnou dostatečné zázemí pro dosažení responzivity kompletního webu. Mezi hlavní využívané třídy bude patřit **row**, **col-sm-*** či **text-center**. Prvně jmenovaná třída seřadí všechny o 1 úroveň podřízené HTML elementy do řádku, druhá jmenovaná třída řádek rozděljuje na 12 stejně širokých částí a třetí třída zarovná text v podřízeném HTML elementu na střed. Kombinace první a druhé zmíněné třídy zajistí v mobilních zařízeních responzivní zobrazení a seřazení jednotlivých podřízených HTML elementu s třídou **box** pod sebe. Tím bude zajištěno dostatečného komfortu pro návštěvníky stránek.

V boxech budou zobrazována data pomocí 3 základních technik. První technikou je standardní HTML tabulka stylovaná pomocí Bootstrap tříd **table**, **table-***, **col-*-***. Tato tabulka bude použita pouze pro zobrazení jednotlivých záznamů z databáze. V případě potřeby zobrazit řádově tisíce až desetitisíce záznamů bude použita tabulka typu grid. Vybrána byla z důvodu náročnosti operace na výkon a také kvůli možnosti filtrování, řazení a postupného načítání dat za pomoci stránkování. K tomu lze využít 2 známé služby. Nextras/Datagrid [5] a Ublaboo/Datagrid [6]. Obě služby poskytují dostatečné grafické prvky, možnosti filtrace + řazení + stránkování, neustále probíhající vývoj a navíc širokou komunitu uživatelů. Navíc je při filtraci či řazení podporována technologie AJAX, která

umožňuje překreslit jen výřez stránky obsahující tabulku typu grid. Není tak potřeba zatěžovat webový server požadavkem na překreslení celé webové stránky. Široká komunita uživatelů je důležitá především při implementaci služby. Poskytuje totiž pomocí diskuzních fór a souvisejících stránek dostatečnou podporu pro uživatele a řešení případných problémů s implementací. Stále probíhající vývoj zakládá předpoklad, že vývoj služby neustane v mrtvém bodě a služba bude aktualizována i v dalších vydávaných verzích programovacího jazyka PHP. Pro tuto diplomovou práci byla zvolena služba Ublaboo/Datagrid [6], především pro její snazší implementaci, širší komunitu uživatelů a lepší výkonové parametry.

Posledním způsobem zobrazení dat návštěvníkům budou grafy. V rámci výše stanovených požadavků je potřeba implementovat platformu, která zvládá obsluhovat velké množství dat, je schopna je zpracovávat v online režimu a zároveň poskytuje širokou paletu nástrojů pro úpravu grafu uživatelem. Tyto požadavky značně omezují výběr platformy. Vzhledem k výkonovým požadavkům a nutnosti obsahovat širokou paletu nástrojů je třeba vybrat platformu postavenou na javascriptu. Příliš časté posílání dotazů do databáze při každé nástrojové úpravě by běh programu výrazně zpomalovalo (nehledě na podmínky spojené se zajištěním bezproblémového chodu databáze). Je tedy nutné k datům do databáze přistoupit pouze jednou, následně je udržovat pomocí kódu (objekty, session, ...) a nástrojové úpravy provádět pomocí javascriptu. Tím je částečně vyřešen problém s výkonem databáze. Ale dále nastává problém s výkonem samotného programu. Desetitisícové soubory dat (pole, kolekce) je třeba předávat javascriptové platformě. K předání dat je ideální zvolit formát *json*, který je podporován jak javascriptem, tak PHP. Podle uvedených požadavků byla nalezena platforma PlotLy [7]. Všechny požadavky splňuje v maximálním možném rozsahu. Je postavena na javascriptu, přebírá data ve formátu *json*, poskytuje širokou paletu nástrojů od přiblížení vybraného výřezu grafu, přes možnost posunu oblasti grafu, oddálení a přiblížení celé oblasti grafu, přepočítávání hodnot na osách X a Y v reálném čase, možnosti automaticky zaměřit kompletní oblast dat grafu či možnosti pořídit fotografii celé oblasti grafu a stáhnout si ji do svého zařízení. Uvedené vlastnosti budou poskytovat vysoký komfort pro uživatele a výrazně tak vylepší uvedené funkčnosti z předchozího řešení webové aplikace zobrazující data z družice VZLUsat-1.

Ke grafům je potřeba navrhnout ještě způsob konverze a výběru dat pro zobrazení. Volba dat bude čistě na uživateli. Bude zobrazen formulář, který poskytne návštěvníkovi možnost vybrat si databázi, ze které chce data zobrazovat. Tato možnost souvisí s požadavkem operátora na budoucí připojení aplikace k více databázím. Dále si uživatel může zvolit z nabídky sledovaných fyzikálních veličin. Zobrazen mu bude vždy graf právě 1 zvolené veličiny. Tím bude dosaženo přehlednosti dané podstránky. V minulé verzi webové aplikace je omezeno zobrazení dat jen na poslední uplynulé dny. Pomocí formuláře si uživatel bude moci navolit rozmezí dnů, mezi kterými chce graf vykreslit. Bude tak možné dohledat zpětně i historické trendy sledovaných fyzikálních veličin. Rozmezí dnů bude vhodně omezeno. Bude tomu tak především z výkonových důvodů vyplývajících z požadavku zajistit bezproblémový běh MySQL databáze pro potřeby záznamu dat přicházejících z družice.

Je taktéž třeba stanovit omezení požadavků návštěvníku webu na databázi. Tato omezení bude moci spravovat operátor webové stránky pomocí jednoduchých nastavení v administraci webu. Budou zaznamenávány požadavky uživatelů na zobrazení grafů. U každého požadavku bude zaznamenán datum a čas provedení požadavku. Uživatelé budou identifikováni pomocí unikátní IP adresy svého zařízení. Data o návštěvnicích a jejich požadavcích budou ukládána do databáze pro účely správy webu. Budou rozdělena do 2 tabulek: *visitor* a *visitor_log*. V první tabulce bude pouze IP adresa uživatele, počet povolených požadavků na vykreslení grafů a aktivita uživatele. Dále zde bude sloupec stanovující počet minut, za které může uživatel provést daný počet požadavků. Po uplynutí tohoto počtu minut od prvního požadavku vyprší požadavku expirace a bude povolen požadavek nový. Výchozí nastavení databázové tabulky bude stanoveno na 25 požadavků za 30 minut. Pokud uživatel překročí tyto omezující parametry, jeho požadavek nebude proveden a uživatel bude vyzván k odeslání požadavku později. Sloupec pro aktivitu uživatele bude typu *tinyint* a jeho hodnota bude udávat, zda je daná IP adresa blokována či nikoliv. Možnost zablokovat IP adresu bude výhradně v režii operátora. V druhé tabulce budou logovány veškeré požadavky návštěvníků. Tedy bude obsahovat již zmíněný datum a čas požadavku. K propojení s IP adresou bude sloužit ID uživatele propojené cizím klíčem s tabulkou *visitor*. Operátor tak bude mít možnost pomocí tabulky typu grid filtrovat kompletní historii požadavků návštěvníka a podle jejího charakteru rozhodovat o úpravě parametrů pro danou IP adresu, případně o její zablokování. Tento nástroj dá správci webu do ruky dostatečné podklady a záruky pro bezproblémový chod databáze.

Dalším parametrem vybíraným parametrem ve formuláři pro tvorbu grafů bude vzhled grafu a zobrazené křivky. Uživatel bude mít možnost volitelně zadat šířku a barvu hodnot grafu, popis vodorovné a svislé osy, pojmenovat graf, zvolit zda chce zobrazit legendu a vybrat styl křivky. Těmito parametry bude plněno json pole, které bude následně předáno šabloně s javascriptem spouštějícím vykreslení grafu.

V rámci veřejné části aplikace bude vylepšena ještě jedna komponenta, kterou je fotogalerie. V původní verzi webové stránky jsou obrázky příliš malé, nelze je zvětšit. V nové aplikaci bude implementována možnost zvětšit si obrázek. Ke zvětšení obrázku dojde po kliknutí na obrázek. Následně bude otevřeno modální okno, které zobrazí obrázek v plné velikosti. Musí být pamatováno na mobilní zařízení, proto by neměly být zobrazeny najednou všechny obrázky v plné velikosti, ale vždy jen 1. Případné ostatní obrázky budou zobrazeny pouze jako miniatury. Tím bude zajištěno šetření stahovaných mobilních dat návštěvníkem stránky a zároveň k jejímu rychlejšímu načítání. Bude tak splněn nejen požadavek návštěvníka na přehlednost, ale i ostatní požadavky zmíněné v přecházející části této práce.

2.3 Backend – vzhled a moduly

V této části práce bude rozebrán návrh administračního rozhraní webu. Návrh lze rozdělit na následující dílčí části – vzhled, zobrazovaná data a jednotlivé moduly vytvořené v rámci práce.

2.3.1 Vzhled administračního rozhraní

Vzhled rozhraní pro operátora by měl splňovat základní požadavky na program vyplývající z úvodní analýzy a uvedených cílů práce. Základní vzhled rozhraní bude realizován pomocí platformy Admin LTE 2 [3]. Ta poskytuje uživateli vhodné a přehledné prostředí podporující CSS techniku **@media**. Administrační část webu tak bude splňovat požadavek na responzivitu. Záhlaví rozhraní je tvořeno vodorovným pruhem. V levé části záhlaví je umístěn název projektu a tlačítko pro skrytí hlavní nabídky. V pravé části záhlaví je odkaz na frontend aplikace a odkaz typu „dropdown“ nabízející uživateli další možnosti. V této rozbalovací nabídce bude možnost odhlášení uživatele a změny hesla. V levé části stránky je umístěno vertikálně orientované menu, které umožňuje skrytí pro případ zobrazení v mobilním zařízení. V rámci tohoto bloku bude využita víceúrovňová hlavní nabídka, pomocí níž budou vhodně seříděny jednotlivé záložky. Toto přinese uživateli přehlednost všech záložek. Záložky budou seříděny na základě modulů, vyplývajících z adresářové struktury programu.

Samotný obsah jednotlivých podstránek se nebude příliš lišit od veřejně dostupné části této webové aplikace. Data budou zobrazována pomocí filtrovacích tabulek typu grid, případné změny dat, nastavení hodnot v databázi bude prováděno pomocí formulářů. Rozdílem oproti původní aplikaci bude přesunutí těchto formulářů do administrační části webu, tedy přidání pomyslné bezpečnostní zdi v podobě přihlášení.

2.3.2 UserModule

V adresářové struktuře budou 3 moduly: „UserModule“, „PublicModule“ a „AdminModule“. Dva z nich budou obsahem administrační části webu. V PublicModule bude kompletní návrh veřejně dostupné části webové aplikace vytvořené v rámci této práce. V následující části práce bude rozebrán návrh a význam uživatelského modulu. Uživatelský modul bude zodpovědný za několik funkcí. Mezi tyto funkce patří přihlášení, odhlášení či změna hesla uživatele, vytvoření přístupu novému uživateli, editace přístupu již existujícímu uživateli, vytvoření seznamu zdrojů a přidělení zdrojů jednotlivým uživatelům.

Přihlašovací stránka bude obsahovat jednoduchý formulář vyžadující vyplnit heslo, uživatelské jméno a nepovinný checkbox pro pamatování údajů. Pozadí přihlašovací stránky bude obsahovat fotografii antény umístěné na střeše budovy FEL převzatou z původní webové stránky. Nepovinný checkbox prodlouží délku přihlášení (před jeho vypršením) v případě neaktivity uživatele na webu. Výchozím nastavením bude 30 minut, po zaškrtnutí bude doba expirace nastavena na 14 dní. Přihlašovací jméno a heslo bude ověřováno oproti záznamu v databázi. Po úspěšném přihlášení bude uživatel přesměrován do administračního rozhraní webu. V hlavní nabídce tohoto rozhraní bude odkaz na seznam uživatelů a seznam rolí. Seznam uživatelů zobrazí v tabulce typu grid všechny uživatele s přístupem do rozhraní. Výchozím nastavením filtrů budou pouze aktivní uživatelé. Aktivním uživatelem je myšlen uživatel s aktivním účtem, účet totiž bude možné deaktivovat. Ověření aktivity účtu bude samozřejmě probíhat už při přihlašování uživatele. Deaktivovaným účtům bude přístup do rozhraní zamítnut. Jednotlivé účty bude možné editovat, případně zakládat nové pomocí jednoduchého formuláře. Zakládání nových účtů nebude probíhat automatizovaně pomocí odeslání automatického potvrzovacího emailu. Tato vlastnost je u dané aplikace nežádoucí. Zakládání nových uživatelských účtů bude zodpovědností a prací pro operátora webu s přístupem do příslušného zdroje.

Při zakládání účtu bude potřeba přidat uživateli práva na přístup k jednotlivým zdrojům. Tato možnost bude realizována pomocí rolí. Jedna role bude může obsahovat více zdrojů. Jednotlivé role bude možné založit a přiřadit jim odpovídající zdroje. Součástí této práce bude i založení vzorových rolí a alespoň jednoho uživatelského přístupu pro pohodlí budoucího správce této webové stránky.

2.3.3 AdminModule

Tento modul bude spravovat zbývající záložky a hlavní stránku administračního rozhraní. Na hlavní stránce rozhraní bude zobrazen formulář s checkboxy sloužícími k nastavení příznaků (flags) v databázové tabulce *cmd_flags*. Na původním webové stránce je tento formulář viditelný na homepage, což působí poměrně nevěrohodně. Ač je formulář zabezpečený proti změně dat, může působit vyzývavě směrem k možnému napadení webu.

Další záložkou z této aplikace přesunutou do administrace bude „STOR info“. V této sekci budou pomocí filtrovacích tabulek typu grid zobrazeny stavy automatického stahování dat z družice při přeletech. Stavy jsou uloženy v databázi současné aplikace. Zde budou stavy i nadále, s tím rozdílem, že v kódu aplikace pro ně budou založeny konstanty pro možnosti rychlejší a jednodušší úpravy kódu. V jednotlivých gridech bude možné filtrovat vybrané stavy pomocí selectboxů.

V hlavní nabídce bude také záložka umožňující vytvořit databázi. Položka nebude tvořit strukturu databáze, půjde jen o vytvoření záznamu o databázi, ke které je možné připojit tuto aplikaci. Daný záznam bude obsahovat název databáze, popis a stav. Pomocí popisu bude snazší identifikovat, o jakou databázi se jedná (název databáze nemusí být vždy jednoznačný). Stav databáze bude možné nastavit na aktivní či neaktivní. Tímto bude operátor regulovat, ke kterým databázím bude aplikace přistupovat a tím pádem, z jakého balíku dat si bude moci návštěvník vykreslovat grafy na veřejné části těchto webových stránek. Tím dostane správce stránek kontrolu nad aplikací a zobrazovanými daty včetně možnosti jejich ovlivnění bez nutnosti provádět velké programové změny.

Poslední částí hlavní nabídky v administrativním rozhraní tohoto programu bude záložka „Návštěvy“. V této části bude mít správce stránek možnost sledovat, z jakých IP adres jsou odesílány požadavky na vykreslování grafů ve veřejné části webu. Jednotlivým adresám bude moci ručně nastavit omezující parametry zmíněné v předchozí části návrhu této aplikace. Počet požadavků a čas bude možné měnit jak „ve prospěch bezproblémového chodu MySQL databáze“, tak „ve prospěch návštěvníka stránek“. Bude zde zahrnuta také možnost zablokovat vybranou IP adresu. V takovém případě nebudou požadavky z této IP adresy vykonány. Součástí AdminModule bude i možnost filtrovat jednotlivé požadavky návštěvníků. Bude možnost vyfiltrovat například požadavky IP adresy ABC mezi dny 01. 12. 2019 a 05. 12. 2019 a následně je seřadit od nejstaršího k nejnovějšímu. Tato funkcionality dá operátorovi dostatečný přehled a podklad pro případné omezení požadavků z dané IP adresy.

2.4 Nette Framework

Pro účely naprogramování této aplikace je potřeba zvolit vhodný nástroj, který umí pracovat se všemi výše uvedenými platformami. Zmíněný nástroj by tedy měl být postavený na programovacím jazyce PHP, měl by podporovat značkový jazyk HTML v kombinaci s javascriptem a kaskádními styly. Dalším důležitým parametrem bude propojení s jazykem MySQL při komunikaci z databází. Podstatná taktéž bude bezpečnost webu, přihlašovacích údajů uživatelů a s tím spojený implementovaný šablonovací systém. Mezi požadavky musíme přidat i vysokou rozšířenost nástroje mezi uživateli a aktivní neboli stále probíhající vývoj. Všechny tyto parametry může splňovat hned několik frameworků. Pro účely této práce byl vybrán Nette Framework [8]. Rozhodla zejména velká komunita uživatelů a s tím související dostupnost velkého množství řešených problémů a odladěných a prověřených doplňků používaných v kombinaci s tímto frameworkem.

Nette je založeno na softwarové architektuře MVC [9]. Tato architektura odlišuje skripty a kódy psané v jednotlivých programovacích jazycích do 3 základních bloků. Model se stará především o logiku aplikace. Přistupuje k databázi, provádí výpočty, přitom není existenčně závislý na ostatních 2 částech. Jen pro ně vykonává takzvanou „černou práci“. Ostatním částem pouze poskytuje vypočtená či získaná data. Taktéž od nich data přijímá, aby s nimi mohl dále naložit. K příjmu či poskytování dat používá vlastní metody, kterými lze data ovlivňovat. Druhá vrstva – „View“ – se stará pouze o to, aby zobrazila návštěvníkovi stránky. Je založena na šablonovacím systému, který má předem nadefinovaný vzhled (HTML, CSS, javascript, ...). Ten se může měnit jen pomocí dat předávaných do šablony třetí vrstvou aplikace. Poslední vrstvou aplikace je „Controller“. Ten se stará o komunikaci s uživatelem. Přijme požadavek od uživatele, zpracuje ho a následně předá výsledná data šabloně (tedy druhé vrstvě). Ke zpracování požadavků si nevolá nic jiného než první zmíněnou vrstvou – model. Ta za něj provede výpočtovou část a vrátí mu data. Následně se controller podle typu požadavku rozhodne, které šabloně má předat data k vykreslení, zavolá ji a šablona se postará o zbytek.

Nette Framework používá vlastní šablonovací systém Latte [10]. Ten podporuje escapování HTML znaků, čímž eliminuje cross-site scripting (bezpečnostní riziko pro webové aplikace). Controllerem je v rámci Nette frameworku takzvaný Presenter. Jedná se o objekt, který dělá přesně to, co by controller dělat měl. Připojení k databázi bude realizováno pomocí volně dostupného doplňku Kravcik/Core [11]. Jedná se o doplněk dědicí od obdobných tříd v podání uestla/YetORM [12]. Repozitáře budou sloužit k definici databázové tabulky, k níž chceme přistupovat. Dále v nich bude definováno, jakou entitou mají být obsluhovány požadavky na danou databázovou tabulku (bude potřeba definovat kompletní namespace dané entity). Entita bude třída, která si v základu vystačí pouze s anotací. V ní budou definovány sloupce tabulky, ke kterým chceme přistupovat a jejich datové typy. Objekt entity následně bude požadovaná data vracet právě v těchto datových typech. Tento systém slouží zejména k zpřehlednění kódu při přístupu k databázi. Následně stačí pouze nadefinovat připojení ke správné databázi pomocí objektu `\Nette\Database\Context` [13] a přístup k datům je zajištěn. Díky tomu bude ulehčeno i zajištění přístupu k více databázím požadované operátory a uvedené v předchozích partiích tohoto textu. Připojení bude možné spravovat z jednoho místa v kódu, což vše značně usnadní.

2.5 Databáze

Z analýzy struktury databázových tabulek vyplývá, že není potřeba zásadně měnit její základy. Bude tedy potřeba pouze připravit vylepšení výkonu při přístupu do vybraných tabulek, ze kterých budou zobrazována data. Sloupcům v tabulkách budou nadefinovány indexy, primární klíče, případně i cizí klíče. To způsobí rychlejší prohledávání MySQL databázové tabulky. Dále bude potřeba vytvořit a přidat tabulky potřebné k vybraným výše popsaným partiím administrace webu. Databáze následujícího technického projektu, která bude připojena v budoucnu tak může převzít strukturu vytvořenou v rámci této práce a nebude tak potřeba provádět ani změny v ukládání dat přicházejících z družice. Připojení k více databázím bude realizováno pomocí konfiguračního souboru, jehož struktura bude popsána v následující části této diplomové práce, tedy v realizaci navrženého řešení. Správu připojených databází bude možné provádět v zabezpečeném veřejně nepřístupném administračním rozhraní webové aplikace vytvořené v rámci této diplomové práce.

3 Realizace navrženého řešení

V této části práce je popsána základní adresářová struktura aplikace včetně jejího významu. Následuje dokumentace frontend a backend části webové aplikace s popisem k čemu slouží jednotlivé části a jak je používat. Zároveň bude ve třetí části této diplomové práce popsáno, v čem jsou výhody a nevýhody vytvořeného programu. Tyto partie budou taktéž porovnány s původní verzí aplikace.

3.1 Adresářová struktura webové aplikace

Program je dělen do složek dle odpovídajících základní struktury Nette Sandboxu [9]. Adresář „vzlusat“ obsahuje složky `/app/`, `/log/`, `/sql/`, `/temp/`, `/vendor/` a `/www/`. Dále jsou zde soubory `.htaccess`, `.gitignore`, `composer.json`, `composer.lock` a `readme.md`.

Soubor `.gitignore` obsahuje informace o souborech a složkách, které jsou ignorovány při případném nahrávání projektu do online repozitáře. `Readme.md` je připraven pro přidání dokumentace do repozitáře. Soubor [14] `.htaccess` je použit k tomu, aby bylo možné při příchodu na web přesměrovat kořenovou složku serveru na kořenovou složku projektu (tj. složku `/www/`) bez nutnosti kontaktovat správce serveru. `Composer.json` obsahuje (jak už název souboru napovídá) `json` pole se závislostmi projektu. Především je v něm uveden samotný Nette Framework, ve kterém je celá aplikace naprogramována. Uvedené závislosti jsou do projektu implementovány přes stejnojmenný program (Composer [15]). Composer se stará i o kompatibilitu jednotlivých doplňků vzájemně mezi sebou, ale i s verzí PHP použitou pro vývoj a běh programu. Po kontrole parametrů Composer stáhne a umístí doplňky do složky `/vendor/`. Informace o stažených verzích jednotlivých balíčků a vyžadovaných knihoven jsou obsaženy v souboru `composer.lock`.

3.1.1 Adresář LOG

Do adresáře */log/* jsou zaznamenávány chybové hlášky, logy. Logy jsou generovány pomocí ladícího nástroje zvaného Tracy [16]. Jedná se o nedílnou součást Nette Frameworku, která pomáhá nejen při psaní programu k debuggování chyb v kódu, ale následně se i za běhu programu na webovém serveru stará o správu informací o chybách programu, které následně ukládá do již zmíněné složky */log/*. Správa webové aplikace je tak jednodušší, pokud dojde k chybě, je odeslán automatický email na adresu uvedenou programátorem. Email obsahuje stručné shrnutí chyby. Následně se programátor přes FTP klienta podívá na server dané aplikace a ze složky */log/* si stáhne kompletní logovanou hlášku včetně odesílaných dat, která chybu způsobila. Tuto chybu pak následně může jednoduše replikovat u sebe v IDE, odstranit její příčiny a změny v projektu následně nahrát na server.

3.1.2 Adresář SQL

Ve složce */sql/* je umístěna databázová struktura projektu (ve formátu **.sql*). Obsah složky je nahrán v MySQL databázi na příslušném databázovém serveru. Do této databáze jsou za běhu programu nahrávána a ukládána data, případně jsou z ní programem zobrazena návštěvníkům webové stránky.

3.1.3 Adresář TEMP

Adresář */temp/* obsahuje tzv. „cache“ a „session“. Jedná se o složky, které pomáhají k rychlejšímu načítání aplikace za jejího běhu. Fungují jako jakási paměť pro opakované načítání dat. Ve složce */temp/cache/* je načtena databázová struktura, ACL, šablony pro zobrazení dat a další opakovaně načítané soubory, třídy, objekty apod. Pokud uživatel přijde na určitou stránku webu (odešle požadavek na server), tzv. RobotLoader [17] se stará o procházení adresářové struktury aplikace a vyhledání objektů, tříd, rozhraní a skriptů, které jsou potřeba ke splnění požadavku a odeslání odpovědi uživateli. RobotLoader má skripty dopředu načtené a indexované v cache, což mu zjednoduší práci – podle indexů potřebné skripty vyhledá a následně může dojít k jejich spuštění. Takto je ušetřen čas, který by byl stráven procházením kompletní adresářové struktury aplikace a uživatel dostane odpověď webu dříve.

Ve složce */temp/sessions/* jsou uloženy relace uživatelů, kteří navštívili webovou stránku [18][19]. Relace (neboli) session je tvořena unikátním identifikátorem návštěvníka webu a daty spjatými s uživatelem. Aplikace vytvořená v této diplomové práci používá session například pro uložení dat o přihlášeném uživateli.

3.1.4 Adresář WWW

Adresář */www/* je jediným veřejně přístupným adresářem webové aplikace vytvořené v rámci této práce. Obsahuje velmi důležitý soubor – *index.php*. Tento soubor se stará o 2 věci – zjistí ze souboru */app/bootstrap.php* nastavení aplikace (bude vysvětleno později) a následně spustí celou Nette aplikaci. Soubor proto musí být umístěn ve veřejné složce, do které je uživatel přesměrován výše zmíněným souborem */.htaccess*, aby bylo možné aplikaci spustit. Adresář dále obsahuje například *favicon.ico*, což je ikona, která se uživateli zobrazí na záložce webového prohlížeče, *.htaccess* soubor, který se stará o zapnutí používání HTTPs protokolu. Ve složce */www/webtemp/* jsou načtené javascriptové (**.js*) soubory a soubory stylů (**.css*), které již byly použity. Stejně jako u cache jsou zde připraveny pro rychlejší načítání opakovaně požadovaných skriptů. V adresáři */www/fonts* jsou uloženy fonty písma používané v této aplikaci, v adresáři */www/images* jsou uloženy veškeré obrázky tohoto webu. Složky */www/js* a */www/css* obsahují veškeré javascriptové soubory a soubory stylů použité při tvorbě webové aplikace.

3.1.5 Adresář APP

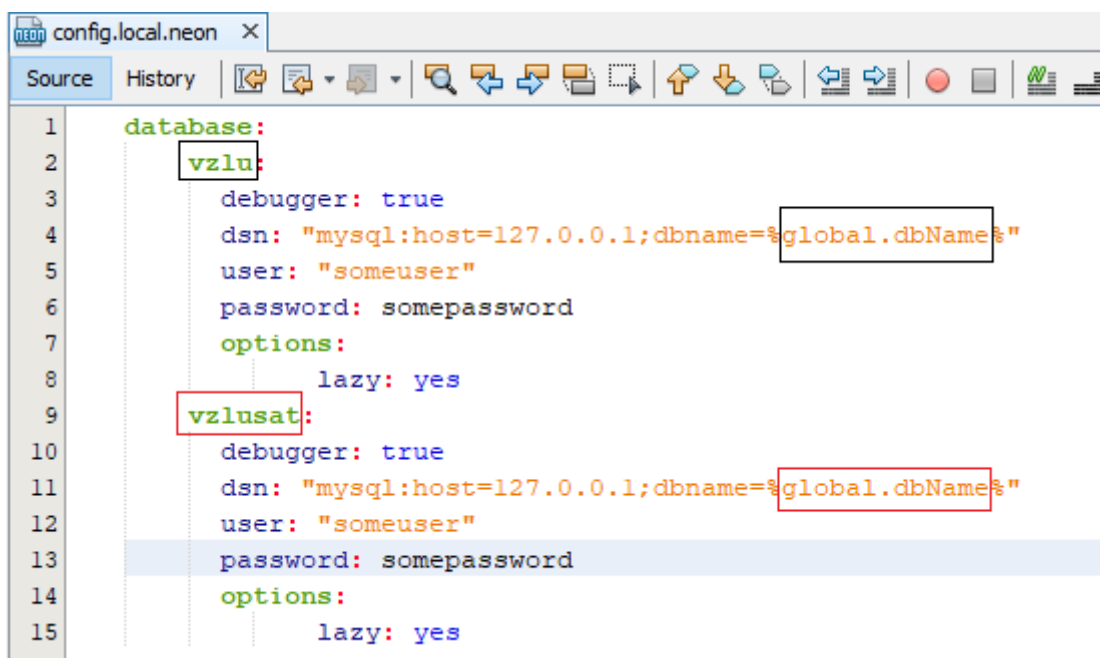
Adresář */app/* je nejdůležitější součástí celé struktury. Je v něm obsažena vrchní vrstva aplikace vytvořené v rámci této práce. Je rozdělen na moduly, přičemž každý modul má velmi podobnou strukturu. Unifikace struktury modulů zajišťuje přehlednost pro programátora, lepší orientaci v kódu. Případné budoucí úpravy kódu či přidávání dalších modulů tak může navázat na současnou strukturu. Kromě rychlejšího „proniknutí“ do projektu toto programátorovi přinese i určitou kostru, kterou může využít jako vzor a odpadne mu tak velká část práce s návrhem modulu. Na začátek je potřeba vysvětlit význam skriptu obsaženého v již jednou zmíněném souboru */app/bootstrap.php*. Tento soubor je načten už ve spouštěčím */www/index.php* a obsahuje důležité informace pro start celé aplikace. Tzv. třída *Configurator* [20] v tomto souboru automaticky generuje *containers* [20] se závislostmi a službami. V souboru *bootstrap.php* je uvedeno jaké konfigurační soubory mají být načteny, jestli má být aplikace spuštěna v produkčním nebo ve vývojovém režimu, jaké složky jsou určeny pro ukládání cache či logů a jsou zde nastaveny parametry objektu *configurator*.

Nyní bude rozebrán adresář */app/*. Složka */app/config/* zahrnuje konfigurační soubory této vrstvy aplikace. Všechny konfigurační soubory jsou zapisovány ve formátu Neon [21]. *Global.neon* obsahuje globální parametry nadefinované zejména pro administrátorské rozhraní aplikace. Důležité jsou zejména parametry *projectName* a *dbName*. Název projektu se uživateli zobrazuje v layoutu u všech podstránek administrátorského rozhraní (backend). Parametr *dbName* nastavuje jméno databáze, ke které se bude aplikace připojovat, pokud uživatel odešle požadavek na přístup a zobrazení dat obsažených v databázi.

Config.neon nastavuje, do jakého adresáře jsou ukládány sessions a zároveň jakou mají platnost (v tomto případě je nastavena na 14 dní, po kterých platnost session vyprší). Dále jsou zde uvedeny chybové hlášky pro formuláře. Dalším významným parametrem je definice rozšíření – *Extension* [22]. Jako jediná *Extension* je v aplikaci vyžadována *WebloaderExtension*, která se stará o nahrávání javascriptových souborů a souborů stylů. Není tak potřeba přidávat do HTML kódu tyto soubory pomocí tagů *<link>* či *<skript>*. A toto se postará *Webloader*. Dále v sekci *application* je definován tzv. *ErrorPresenter*, který se stará o zobrazení chybové stránky uživateli ve chvíli, kdy na webu dojde k nějaké

chybě. Významný parametr je taktéž *mapping*. Pomocí tohoto parametru Nette automaticky vytváří instance speciálních tříd, takzvaných Presenterů, ve chvíli, kdy jsou v aplikaci volány pomocí tohoto zjednodušeného mapování. Odpadá tak nutnost definovat třídy na daném místě kódu ručně pomocí klíčového slova *new* a názvu třídy s kompletní namespace. Posledním a nezbytným nastavením uvedeným v tomto souboru je služba Router. Ta se stará o tvar URL adresy (takzvané routování) – na základě výše popsaného mappingu tato třída generuje URL adresu do požadovaného tvaru, odpovídajícímu adresářové struktuře webu - tedy */Module/Presenter/action*. V konfiguračním souboru *webloader.neon* jsou definovány jednotlivé soubory stylů a javascriptů, které mají být v aplikaci použity [23]. Soubory jsou zde navíc definovány a logicky děleny v balících (basic, homepage, public, login, default) a následně jsou v právě využívaných objektech presenterů volány přímo tyto balíky souborů, aby byly vždy dostupné jen ty soubory, které jsou momentálně potřeba k odpovědi na požadavek uživatele. Aplikace tak předchází načítání souborů, které v daném čase nejsou potřeba, a následnému zpomalení běhu programu.

Config.local.neon obsahuje, jak už je z názvu souboru jasné, místní nastavení aplikace. Tj. tento soubor je odlišný pro vývoj na lokálním PC a pro běh programu na webovém serveru. Jsou zde obsaženy informace pro připojení k databázi. Pod klíčem *dsn* je uvedena adresa databázového serveru a název databáze. Název databáze je plněn globálním parametrem *dbName* uvedeným v konfiguračním souboru *global.neon*. Pod klíči *user* a *password* je uvedeno přihlašovací jméno a heslo pro přístup do databáze.



```
1 database:
2   vzlu:
3     debugger: true
4     dsn: "mysql:host=127.0.0.1;dbname=%global.dbName%"
5     user: "someuser"
6     password: somepassword
7     options:
8       lazy: yes
9   vzlusat:
10    debugger: true
11    dsn: "mysql:host=127.0.0.1;dbname=%global.dbName%"
12    user: "someuser"
13    password: somepassword
14    options:
15      lazy: yes
```

Obr. 1 - Config.local.neon pro 2 databázová připojení – barevně označené parametry vyžadují shodný řetězec.

Posledním konfiguračním souborem je *services.neon*. V tomto souboru jsou obsaženy všechny třídy vytvořené v rámci této práce, které nejsou registrovány a načítány automaticky na základě dědičnosti. Jedná se tedy o **Repository*, **Factory*, **Model* či **Manager* třídy. Jejich zaregistrováním jako služby není potřeba vytvářet jejich instance v místě, kde jsou potřeba. Služby je potřeba zapsat do konstruktoru tříd, které s nimi pracují a následně jsou jejich instance vytvářeny automaticky a především pouze tehdy, kdy jsou služby aktivně využívány. Toto výrazně usnadňuje programátorskou práci a navíc urychluje běh aplikace.

Ve složce */app/router/* se nachází třída *RouterFactory*. Ta kompletně zajišťuje routování URL adresy. Využívá k tomu objektu `\Nette\Application\Routers\RouteList`, do kterého přidává instance objektu `\Nette\Application\Routers\Route` [24]. To umožňuje vytvořit více rout a tím pádem nastavit i více mappingů. URL adresa je v tomto projektu mappována pouze jako */ModuleName/PresenterName/*. Toto je dostačující i vzhledem k tomu, že není potřeba řešit otázky SEO apod.

V adresáři `/app/presenters/` jsou 2 soubory významné pro tuto webovou aplikaci. `@public.latte` a `Presenter.php`. Soubor `Presenter.php` obsahuje v rámci této práce vytvořenou abstraktní třídu `\App\Presenter\Presenter`. Tento objekt je předkem pro všechny ostatní presentery v rámci této aplikace. Jsou v něm vytvořeny komponenty `Css`, `BasicCss`, `Js`, `BasicJs`, které za pomoci `Webloaderu` načítají soubory stylů a javascriptové soubory. Dále zde můžeme najít metodu zajišťující zobrazení a vypsaní všech `flashMessages`, metodu `findLayoutTemplateFile()`, která se stará o rozpoznání správné layout šablony obsahující HTML kód. V rámci tohoto projektu jsou použity 2 odlišné layouty – pro backend a pro frontend. Výše zmíněná metoda se rozhoduje pro správný layout na základě presenteru. Pokud je aktuálně používán presenter instancí `\App\Presenter\Presenter`, jedná se o frontend rozhraní a je zobrazena šablona </app/presenters/templates/@public.latte>. Pokud se jedná o instanci presenteru `\App\User\AuthPresenter` (presenter, který je zodpovědný za přihlášení uživatele apod.), je zobrazena šablona </app/UserModule/templates/@admin.latte>. Toto zajišťuje odlišný vzhled stránek pro administraci a frontend.

Dále je aplikace členěna do třech základních modulů – `AdminModule`, `PublicModule` a `UserModule`. `PublicModule` je zaměřen výhradně na frontend část webové aplikace. Do šablony `@public.latte` je zde vložena šablona </app/PublicModule/templates/@layout.latte>. V ní je obsaženo menu pro frontend, nezbytná cookie lišta, obrázek zobrazený v záhlaví webu, patička s tlačítkem pro přihlášení uživatele a především je do této šablony dynamicky vkládán obsah do definovaného bloku `{define #bodyContent}{/define}`. Je tak naplno využito výhod šablonovacího systému Latte [10]. V tomto modulu jsou taktéž obsaženy: formulář pro zobrazování grafů, grida pro filtrování dat z databáze Vzlusat, komponenty zobrazující záznamy z databáze či obrázky z fotogalerie, příslušné presentery zajišťující zobrazení dat uživateli, entity se strukturou databázových tabulek, repozitáře zajišťující propojení entity se správnou tabulkou v databázi a rovněž šablony příslušící danému presenteru. `UserModule` je převážně součástí backend části aplikace. Je zodpovědný za přihlašování uživatelů, ověření jejich hesla, uživatelského jména a jeho ACL. Realizace ověření přístupů je zajištěna pomocí rolí přiřazených konkrétnímu uživateli. V administraci webu je možné vytvořit novou roli, přiřadit ji libovolnému uživateli a kontrolovat tak přístup k jednotlivým položkám správcovského rozhraní. Dále tento modul umožňuje založení nového uživatele pro přístup do správcovského rozhraní.

AdminModule je zodpovědný za všechny zbývající funkce administračního rozhraní. Je v něm možné přidat přístup k další databázi, případně přístup k libovolné databázi deaktivovat. Je zde možnost nastavit jednotlivým IP adresám navštěvujícím frontend část aplikace zobrazující grafy maximální počet požadavků, které smí odeslat za určitou dobu. Tuto dobu zde lze také manuálně nastavit, případně je možné požadavky určité IP adresy blokovat, pokud se k tomu operátor spravující webovou aplikaci rozhodne. Je možné i filtrovat jednotlivé požadavky všech IP adres, aby měl operátor dostatek podkladů pro toto rozhodnutí. Na hlavní stránce administrace je možné nastavit stavy pro komunikaci s družicí uložené v databázové tabulce *cmd_flags*. V záložce *Storage* je možné zjistit stavy přenosů dat vycházející z databázových tabulek *s1p41*, *s1p34*, *s1p41*, *s4p2*, *s4p3*, *s6p2*.

Výše uvedené moduly UserModule a AdminModule taktéž obsahují konfigurační soubory *config.neon*. Tyto soubory nastavují položky v menu správcovského rozhraní. Názvy všech klíčů v menu jednoznačně říkají, co položka znamená – například *title* obsahuje titulek zobrazený v menu, *module* název modulu, *presenter* název presenteru nebo *action* pro zobrazení správné šablony presenteru – vše pro jednoduchou tvorbu URL adresy.

3.2 FRONTEND

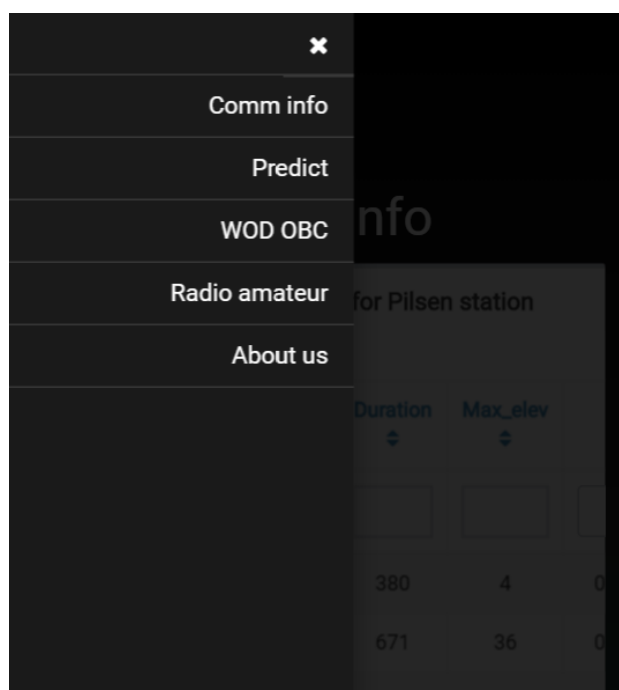
Hlavní částí webové stránky vytvořené v rámci této práce je uživatelské rozhraní pro běžného uživatele, takzvaný frontend. Jeho popis lze rozdělit na popis grafického rozhraní a popis zobrazovaných dat.

3.2.1 Frontend - vzhled

Pro vzhled této části aplikace byla navržena jednoduchá grafická šablona. Šablona sestává ze záhlaví, horizontálního menu, obsahu dané podstránky a patičky webu. Jako pozadí webu byl zvolen barevný přechod od odstínů černé (hex kód #000000) přes tmavě zelenou (hex kód #037263) až po návrat k barvě černé (hex kód #000000). K tvorbě šablony a obecně většiny grafických prvků byl použit grafický HTML framework Bootstrap 3 [4]. V záhlaví webu je na pozadí vložen obrázkový banner [25] ve formátu *.png*. Horizontální menu bylo zvoleno zejména pro jeho přehlednost na desktopových zařízeních zobrazující webové stránky. Z návrhu programu uvedeném v předchozí části práce vyplývá potřeba přizpůsobení webové stránky pro mobilní zařízení. Bylo proto potřeba naprogramovat menu tak, aby při zobrazení v těchto zařízeních došlo ke sbalení celé nabídky do jednoho tlačítka. K tomu bylo použito čistě CSS metod [26]. Po kliknutí na toto tlačítko v mobilním zařízení je zobrazena vertikálně orientovaná nabídka položek z menu.

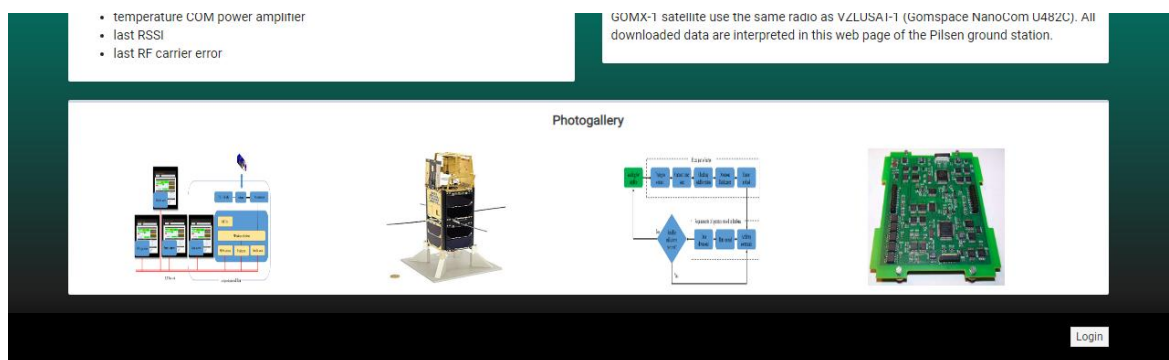


Obr. 2 - Vzhled menu pro stolní počítače

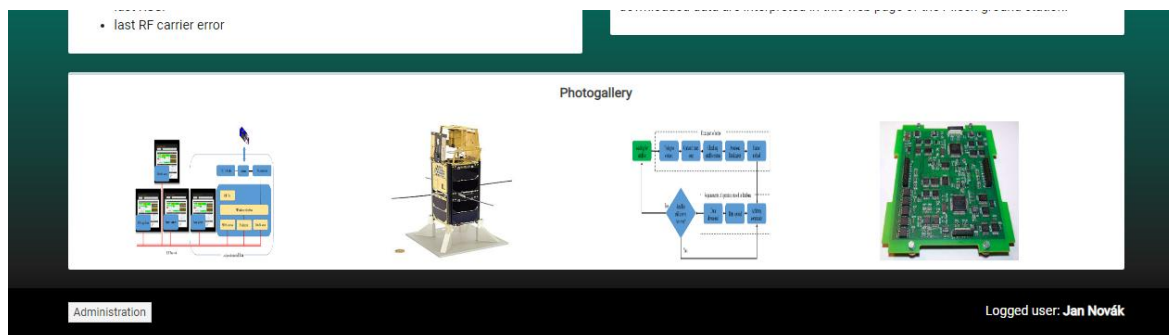


Obr. 3 - Vzhled menu pro mobilní zařízení

Veškerá zobrazovaná data, tabulky, grafy a obrázky jsou vloženy do bílých responzivních boxů. Pokud by velikost zobrazovaného obsahu přesahovala samotný box, je nastaveno zobrazení vodorovné scrollovací lišty (pomocí CSS vlastnosti *overflow-x: auto;*) k posunu obsahu v rámci boxu. Je tak zajištěno správné zobrazení boxů na všech zařízeních bez nutnosti oddalování stránek na mobilních zařízeních. Jak už bylo zmíněno, boxy a jejich umístění v HTML kódu bylo navrženo tak, aby zajišťovaly maximální responzivitu nejen obsahu vloženého do boxu, ale i samotného boxu. Pokud tedy v desktopové verzi aplikace můžeme nalézt více boxů v jedné horizontální rovině, v mobilním zobrazení jsou boxy seřazeny pod sebe. Tohoto efektu je docíleno pomocí Bootstrap tříd *col-sm-** rozdělující vybraný nadřazený element stránky horizontálně na 12 stejných libovolně kombinovatelných dílů [27]. V patičce webu je umístěno tlačítko pro přihlášení uživatele. Přihlášený uživatel zde vidí vypsané své přihlašovací jméno a tlačítko pro vstup do administrační části webu.



Obr. 4 - Vzhled patičky pro běžného návštěvníka



Obr. 5 - Vzhled patičky pro přihlášeného uživatele

3.2.2 Frontend – zobrazení dat

O zobrazení dat ve veřejné části aplikace se starají jednotlivé presentery obsažené ve složce */PublicModule/presenter/*. Presentery dědí od *PublicPresenteru* vytvořeného taktéž v tomto adresáři. *PublicPresenter* předává potomkům static proměnou *\$image*, která obsahuje CSS kód s cestou k banneru zobrazenému v hlavičce webu. Dále jsou zde do šablony nadefinovány proměnné *\$user* a *\$loggedUser*, jejichž obsah je využit k zobrazení obsahu patičky webu. Není tak nutné definovat tento kód v každém potomkovi *PublicPresenteru* zvlášť. V následujících částech textu této práce budou probrány jednotlivé položky v menu veřejně přístupné části webu a související podstránky zobrazované příslušnými presentery.

3.2.2.1 Menu - COMM info

CommunicatePresenter zobrazuje uživateli informace a data o přeletech družice *Vzlusat*, její pozici, data o fyzikálních parametrech jednotlivých komponent družice (teplota, napětí, proud) nebo například data o naposledy odeslaných požadavcích a obdržených odpovědích družice. K těmto informacím lze přejít pomocí položky „COMM info“ umístěné v menu. *CommunicatePresenter* vytváří instance čtyř základních komponent dědicích od *\Nette\Application\UI\Control*, které jsou umístěny v podadresáři */component/*. Jedna komponenta vytvářející instanci *\Ublaboo\DataGrid\DataGrid* je umístěna v podadresáři *grid*. Jednotlivé komponenty získávají data z databáze, případně je upravují pro následné umístění do HTML kódu a předávají data šabloně. Šablony podporující šablonovací formát *latte* [10] následně vykreslí data v požadovaném formátu. V případě zobrazení posledních záznamů z databáze byly použity jednoduché HTML tabulky, pro zobrazení kompletního obsahu databázové tabulky s názvem *dk_list* byla vybrána tabulka s možností filtrace a řazení záznamů, takzvaný grid.

3.2.2.2 Menu – predict, pass info

Následující položkou v menu je „Predict“. O zobrazení dat se zde stará PredictPresenter. Na této podstránce jsou k vidění 2 základní boxy obsahující výstupy typu grid. Na této podstránce jsou uvedena kompletní data o přeletech družice. Jak už název presenteru napovídá, 1 grid obsahuje predikované přelety pro následující zhruba 3 týdny. Druhý grid obsahuje historii uskutečněných přeletů družice. Jsou zde uvedeny datumy a časy přeletů, doba trvání přeletu či maximální elevace.

3.2.2.3 Menu - WOD OBC

WodPresenter zobrazuje obsah menu položky „WOD OBC“. Na této podstránce je možné vykreslovat grafy fyzikálních parametrů komponent družice. O volbu parametrů grafu se stará třída *GraphForm* umístěná ve podsložce */form/*. Formulář má celkem 12 polí, které je možné vyplnit. Povinné položky jsou uvedeny v sekci „Graph Data“, nepovinné se nacházejí v sekci „Graph Layout“. V povinných položkách je potřeba vybrat databázi, data a rozmezí dnů, během kterých došlo k naměření a uložení dat do databáze. Výběr databáze vychází z jednoho ze základních požadavků na program stanovených v návrhu této aplikace. Tato možnost počítá s připojením webové aplikace na více databází (způsob připojení bude popsán později). Po připojení aplikace na databázi budoucího následovníka družice VZLUsat-1 bude možné vybrat, ze které databáze bude graf naplněn daty. Databázi si uživatel bude moci zvolit pouze pomocí 1 jednoduchého selectboxu. Selectbox „Data“ dává uživateli možnost zvolit, jaký fyzikální parametr chce v grafu zobrazit. Bude zobrazena jeho závislost v čase. Poslední 2 povinné položky formuláře „From“ a „To“ dávají uživateli možnost vybrat si rozmezí datumů mezi kterými byla data naměřena.

The screenshot shows a web interface titled "WOD OBC" with a sub-header "Choose & style your graph". The interface is split into two main sections: "GRAPH DATA" and "GRAPH LAYOUT".

GRAPH DATA:

- Database *: vzlusat - přístup do DB vzl
- Data *: Battery voltage [V]
- From *: 29.04.2019
- To *: 01.05.2019

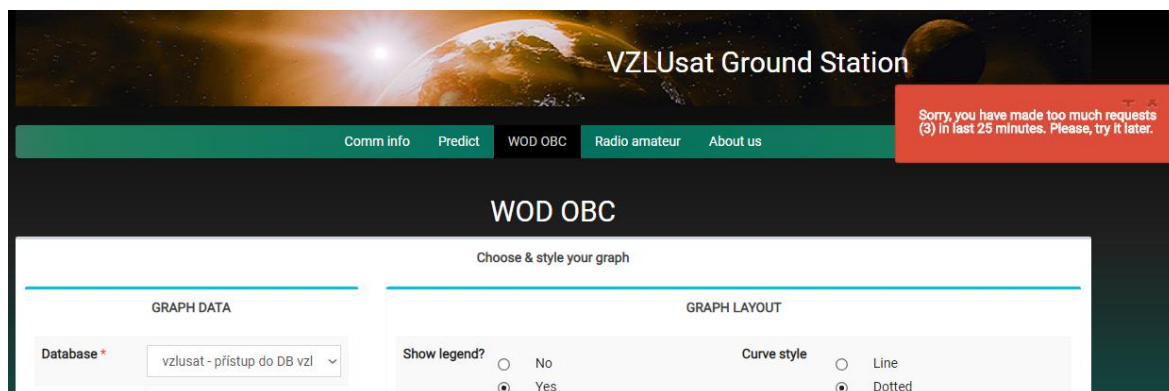
GRAPH LAYOUT:

- Show legend?: Yes, No
- Curve style: Dotted, Line
- Graph title: Battery Voltage
- Curve title: Voltage
- X axis title: Date
- Curve color: Green
- Y axis title: Voltage [V]
- Curve width: 1

At the bottom of the form is a button labeled "Render graph".

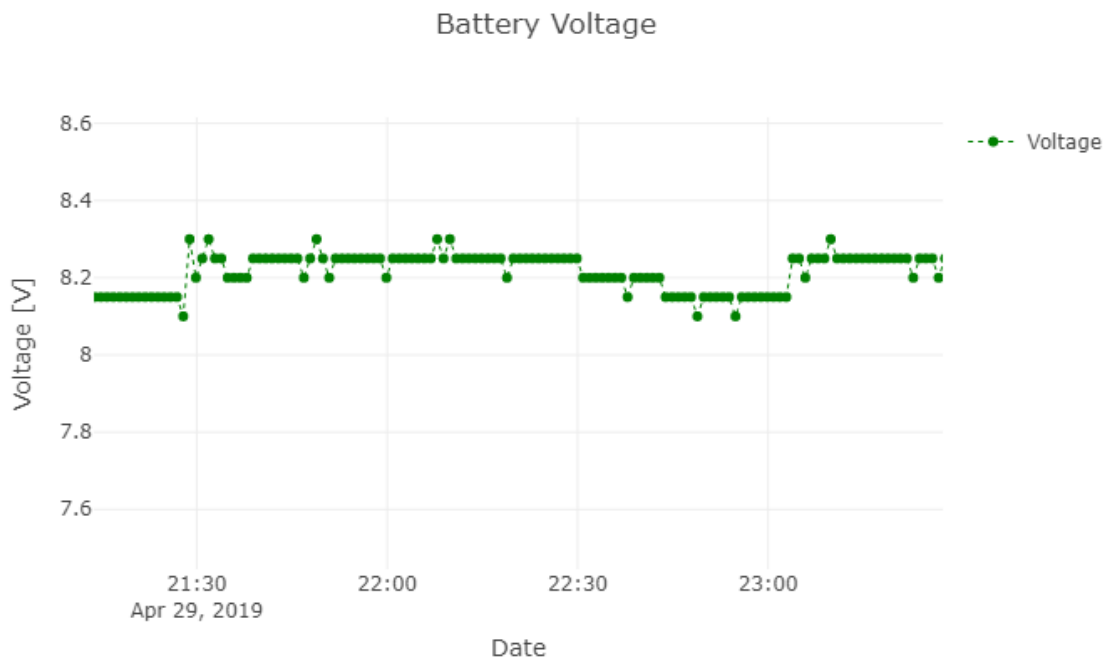
Obr. 6 - Formulář pro tvorbu grafů

Zde se dostáváme k omezení grafů vyplývajícím z návrhu aplikace. Databáze nesmí být zatěžována vysokým počtem požadavků, aby bylo možné neustále ji používat pro ukládání dat přicházejících z družice. Proto je rozmezí datumů nastaveno maximálně na 8 dní. Při frekvenci záznamu dat 1 záznam za minutu se jedná maximálně o 11 520 záznamů, k nimž je do databáze přistupováno během 1 požadavku. Tato hodnota sama o sobě nepředstavuje významné omezení pro rychlost MySQL databáze. Je třeba počítat s tím, že se nacházíme na veřejné části webu, kde může být v 1 moment více návštěvníků odesílajících požadavky na vykreslení grafů. Proto je zde nastavena hranice maximálního počtu požadavků za určitý časový interval pro návštěvníka. Návštěvník je identifikován pomocí IP adresy svého zařízení. Výchozí nastavení těchto hodnot je 25 požadavků za 30 minut. Požadavky návštěvníků webu jsou logovány v tabulce *visitor_log*. Návštěvníci jsou zaznamenáváni v tabulce *visitor*. Každé IP adrese je možné individuálně nastavit maximální počet požadavků i počet minut, za které tyto požadavky smí provést. Dále je zde možnost zablokovat požadavky pocházející z určité IP adresy. Tyto parametry lze jednoduše nastavit v administračním rozhraní webu (v další části práce budou tyto funkčnosti programu detailně rozebrány a popsány). Po vypršení časového limitu nejstarší z počtu požadavků je uživateli zpřístupněn další požadavek (nejstaršímu požadavku vyprší platnost).



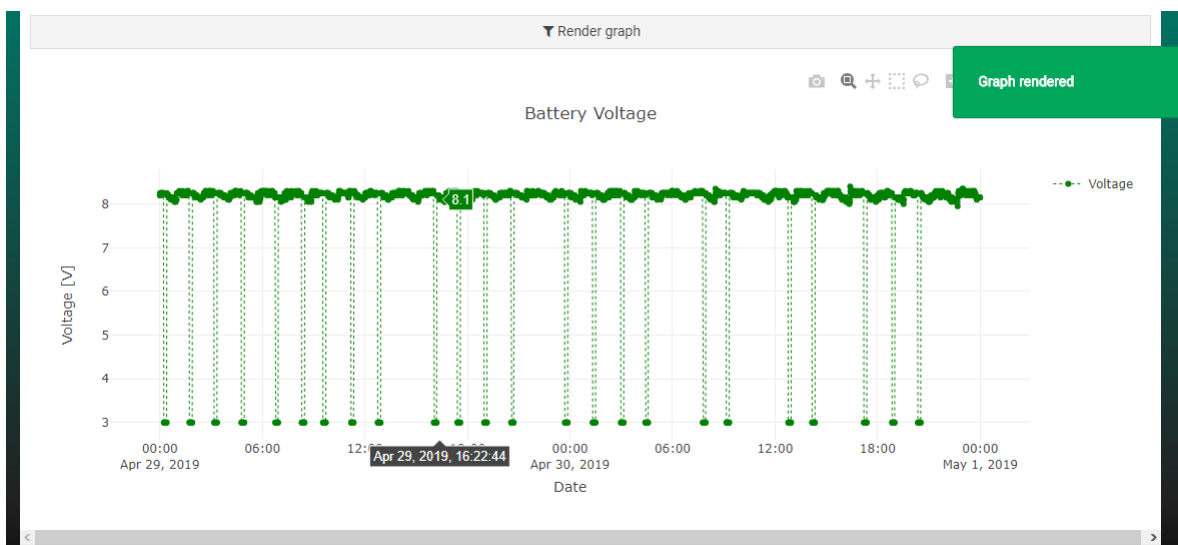
Obr. 7 - Zamítnutí provedení požadavku uživatele

Omezení formuláře je zajištěna správná doba odezvy databáze při zápisu dat z družice a může být rozebrána další část formuláře pro zobrazení grafů – Graph Layout. Tato sekce formuláře je zodpovědná za vzhled grafu. Radiolisty „Show legend“ a „Curve style“ určí, zda má být zobrazeny legenda křivky a zda má být křivka tečkovaná či plná. „Graph title“ dává uživateli možnost napsat si název grafu dle vlastní potřeby. Obdobnou funkčnost zajišťují i položky „X axis title“, „Y axis title“ a „Curve title“ – tedy popisky osy X, Y a vykreslené křivky. Pomocí „Color“ a „Width“ lze zvolit barvu a tloušťku vykreslené křivky. Formulář je potřeba odeslat pomocí tlačítka „Render graph“. Po odeslání formuláře dochází nejdříve k logování požadavku návštěvníka stránek. Pokud požadavek projde přes výše uvedená omezení, je odeslán dotaz do MySQL databáze, který zprostředkuje požadovaná data. Data pro graf a pro vzhled grafu jsou ze success metody formuláře předána do komponenty `\Wod\Component\Graph`. Zodpovědností této třídy je, aby byla data seříděna do statického asociativního pole, které je následně pomocí vestavěné PHP funkce `json_encode()` převedeno do formátu `json` a předáno do šablony. Zároveň je ke komponentě připojen javascript zapsaný v souboru `graphJs.latte`. Nyní je potřeba graf vykreslit. To probíhá pomocí volně dostupného nástroje Plot.ly [7]. Ve `startup()` metodě `WodPresenteru` je připojen javascriptový balík (pro webloader komponentu `js` již zmiňovanou v předchozí části práce) obsahující soubor `plotlyjs-bundle.js`. Jedná se o javascriptovou knihovnu starající se o vykreslení grafu a jeho další vlastnosti od zoomování dat až po printscreen grafu ve formátu `.png`. Knihovna je volána a spouštěna skriptem napsaným ve zmíněném souboru `graphJs.latte`. Graf je vykreslen pod formulářem. Když na graf najedeme kurzorem, zobrazí se v jeho pravém horním rohu nabídka ikon. Ta přináší možnost živě pracovat s grafem – stáhnout printscreen zobrazeného okna grafu, přiblížit určitou oblast grafu, posunout, přiblížit či oddálit celý graf nebo pomocí „autoscale“ resetovat data grafu do původního nastavení.



Obr. 8 - Printscreen grafu pořízený pomocí nástrojového vybavení grafů

Při této práci s grafem jsou zároveň přepočítávány hodnoty na osách, tak aby odpovídaly měřítku aktuálně zobrazených dat. Pokud najedeme kurzorem myši na určitý vykreslený bod, je zobrazen tooltip (popisek) s hodnotami bodu na osách X a Y. Možnost takto „zaměřit“ jednotlivé body dává uživateli podrobnější náhled na data získávaná z družice Vzlsat.



Obr. 9 - Vykreslení grafu a následné zaměření 1 konkrétního bodu

3.2.2.4 Menu – radio amateur

Další položkou v menu je „Radio amateur“. Tato podstránka je určena zejména pro radioamatéry. Jsou zde uvedeny informace o frekvenci pro rádiové spojení s družicí, použité modulaci aj. Na této podstránce je také implementována vylepšená fotogalerie. Miniatury obrázků a fotek jsou zobrazeny v responzivním boxu, což zajišťuje nižší požadavek na objem dat předávaných ze serveru na stranu uživatele při načtení podstránky. Stránka se tak načítá rychleji. Po kliknutí na miniaturu se zobrazí modální okno. V modálním okně je fotka/obrázek zobrazena v plné velikosti. Navíc je zde možnost přechodu na další obrázky pomocí navigace. Uvedené obrázky jsou převzaty z původní verze webové aplikace pro zobrazování dat z družice VZLUsat-1 [25].

3.2.2.5 Menu – about us

Na uvedené podstránce lze nalézt informace o projektu Vzlusat, o skupině vědeckých pracovníků, která se projektem zabývá aj. Uvedené informace jsou převzaty z původní verze webu VZLUsat-1.

3.3 BACKEND

Pro správu webové aplikace je využíváno administrační uživatelské rozhraní. Do rozhraní má přístup pouze přihlášený uživatel. Z veřejně přístupné části webových stránek se do backend části aplikace lze dostat pomocí tlačítka „Login“ umístěného v patičce stránky. Uživatel bude po stisku tlačítka přesměrován na přihlašovací stránku. V levém horním rohu je umístěno tlačítko pro návrat zpět na frontend. Na pozadí přihlašovací stránky je vložena fotografie antény přijímající signál z družice umístěné na pozemní stanici na střeše budovy Fakulty Elektrotechnické v Plzni [25]. Samotné přihlášení uživatele vyžaduje vyplnění pouze 2 údajů v přihlašovacím formuláři. Checkbox „Pamatovat si moje údaje“ je nepovinný. Po zaškrtnutí checkboxu a odeslání formuláře, pokud dojde k úspěšnému přihlášení uživatele, je nastavena expirace přihlášení uživatele v případě jeho neaktivity na webu na 14 dní. Pokud uživatel nechá pole nezaškrtnuté, provede úspěšné přihlášení a následně nebude aktivně procházet webovou stránku, bude expirace jeho přihlášení 60 minut.

3.3.1 Backend – přihlášení

Přihlašovací stránkou se dostáváme do části aplikace, za kterou je zodpovědný `UserModule` [28]. Nyní je potřeba zmínit, co znamená „úspěšné přihlášení“. Ve třídě `LoginForm` dojde po odeslání formuláře k několika úkonům. Po nastavení expirace objektu typu `\Nette\Security\User`, předanému třídě přes konstruktor, je volána metoda objektu s názvem `login()`. Parametry této metody jsou **nickname** a **password**. Metoda je volána v **try** bloku a ověřuje přihlašovací údaje uživatele v databázové tabulce `user`. V případě nesprávných údajů je „vyhozena“ výjimka `\Nette\Security\AuthenticationException`. Uživatel musí v takovém případě zadat znovu přístupové údaje, tentokrát správné. V případě správně zadaných údajů je objektu nastaven stav uživatele `logged` a vyplněn seznam rolí, ke kterým je uživatel oprávněn. Jednotlivé role obsahují soupis všech zdrojů, ke kterým je uživatel oprávněn (ACL). Je tak zajištěno, že uživatel má možnost zobrazit pouze zdroje a akce, ke kterým má oprávnění. V případě, kdy by se uživatel pokoušel přistoupit k jiným zdrojům, dojde k volání `ErrorPresenteru`. Ten zajišťuje volání chybových šablon. Rozhodnutí, kterou chybovou šablonu zvolí, závisí na instanci vyhozené výjimky a s ní souvisejícího chybového kódu. Po přihlášení je uživatel přesměrován na zdroj `:Admin:Homepage:`. Tento zdroj zajistí vytvoření instance `presenteru` `Homepage` v modulu `Admin` a daný `presenter` zobrazí domovskou stránku administrace webu.

3.3.2 Backend – vzhled rozhraní

Vzhled webového rozhraní je tvořen vertikálně orientovanou nabídkou, hlavičkou a samotným obsahem podstránek. Rozhraní je implementováno z volně dostupné služby Admin LTE 2 [3]. Hlavička webu zde obsahuje několik komponent. V levém horním rohu je uveden název projektu a jeho verze. V obou případech se jedná o globální parametry zapsané v konfiguračním souboru *global.neon*. Ty jsou takto dynamicky předávány napříč celou aplikací, aby je nebylo třeba stále definovat. Jejich definice je uvedena právě na jednom místě v tomto konfiguračním souboru. Nevznikají tak problémy s úpravou kódu v budoucnu – tímto je problémům předcházeno. Jak už bylo zmíněno, globální parametry jsou předávány do šablony – v tomto případě jde o šablonu *@admin.latte*. Vpravo od názvu projektu je tlačítko pro „sbalení“ vertikálního menu. V pravém horním rohu je v hlavičce přidán odkaz na frontend aplikace a Jméno a Příjmení uživatele. Po kliknutí na jméno uživatele je zobrazena rozbalovací nabídka typu **dropdown**.

V této nabídce lze nalézt 2 odkazy. První odkaz přesměruje uživatele na formulář pro změnu hesla. Tento formulář je sám o sobě velice prostý. Vyžaduje po uživateli zadání původního hesla, následně nového a potvrzení nového hesla.

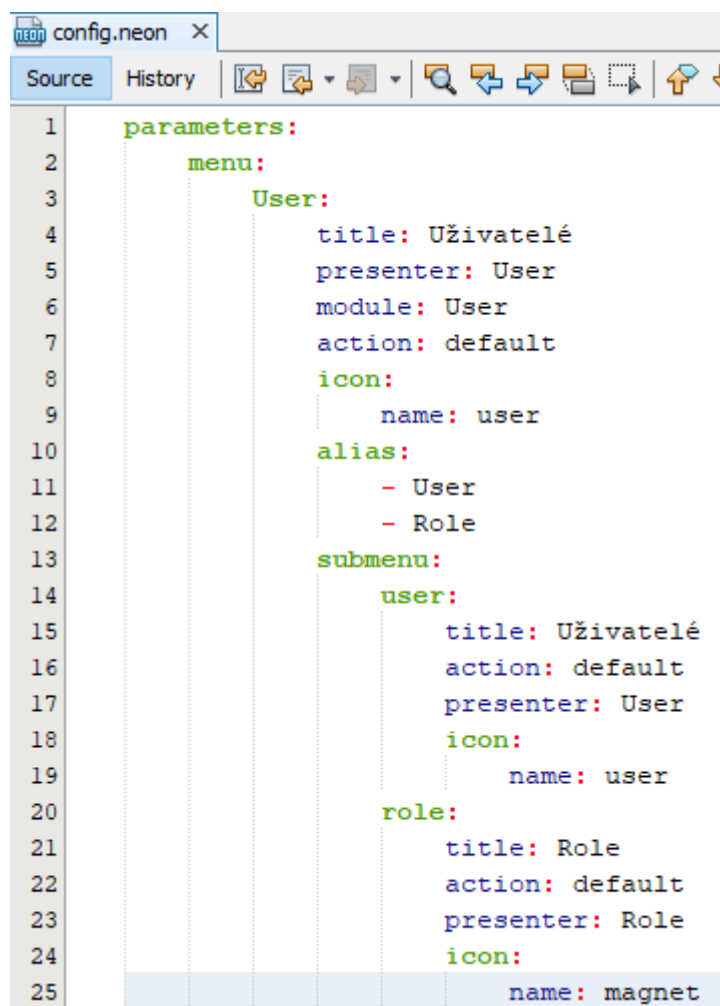
Heslo má nastavená pravidla, která jsou formulářem kontrolována. Jedním z pravidel je, že heslo nové se nesmí shodovat s heslem původním. V takovém případě je uživatel vyzván k opětovnému vyplnění formuláře. Naopak pole „Nové heslo“ a „Potvrzení nového hesla“ se shodovat musí. Posledním pravidlem kontrolovaným formulářem je nutnost minimální délky 8 znaků, přičemž minimálně 1 znak musí být číselný, 1 znak musí být malé písmeno a 1 znak musí být velké písmeno. Znaků mohou být pouze alfanumerické. Pokud některému z pravidel uživatel nevyhoví, je aplikací vyzván k opětovnému přihlášení uživatele. Druhým odkazem v **dropdown** nabídce je odhlášení uživatele. Po úspěšném odhlášení je uživatel přesměrován na přihlašovací obrazovku. Odtud má již zmíněné možnosti – znovu se přihlásit nebo přejít na frontend aplikace.

Poslední částí tohoto rozhraní je vertikální menu. K menu je třeba zmínit, že je dvouúrovňové. Každá položka tedy může obsahovat vlastní submenu. Třída, která se stará o zobrazení menu a získání dat pro naplnění nabídky se nachází AdminModule a její název je `\Admin\Component\Menu`. V této třídě je zásadní metoda `generate()`, která načte proměnou `$this->parameters` a parsuje ji do asociativního pole. Ve zmíněné proměnné jsou uloženy všechny parametry uvedené v konfiguračních souborech typu *neon*. Následně jsou z parametrů vybrány ty, které mají jako nadřazený klíč „menu:“ či „submenu:“.

3.3.3 Backend – tvorba položek v menu

Na tomto místě je třeba zmínit způsob tvorby menu a submenu v konfiguračních souborech. V AdminModule a UserModule jsou umístěny konfigurační soubory *config.neon*. Zápis menu konfiguračních souborů velmi připomíná zápis asociativního pole [21] [28]. Všechny sekce jsou odsazeny pomocí tabulátorů. Případné nahrazení mezerami je odchyceno výjimkou. Důležitá je sekce **parameters**: přes dvojtečku je přístupováno k další sekci. Tou je **menu**:. Následuje klíč pro 1. úroveň menu. Následují povinné položky podřazené první úrovni nabídky. „Title“ obsahuje text, který uvidí návštěvník stránek. Název presenteru a název modulu zajistí pozdější zobrazení správné podstránky. Následuje „icon“ – tato hodnota určí ikonku zobrazenou v menu u titulku. Všechny ikony použité na tomto webu pocházejí z volně dostupné databáze FA Icons 4.7 [29]. Následují aliasy presenterů. Tato sekce není povinná. Alias se postará o zobrazení submenu v rámci jeho podstránek. Zjednodušeně řečeno – pokud je v 1. úrovni menu zvolena možnost „Uživatelé“ a na 2. úrovni menu je zvolena například možnost „Role“ při chybějícím zápisu aliasů v konfiguračním souboru, je zobrazena podstránka „Seznam rolí“, ale 2. úroveň nabídky bude sbalena. V případě zápisu aliasů presenterů obsažených v dané sekci menu do konfiguračního souboru je zobrazena i druhá úroveň nabídky. Následuje sekce **submenu**: - do této podřazené sekce jsou zapsány všechny položky 2. úrovně menu. Každá podsekce má svůj klíč, ke kterému jsou přiřazeny podřazené položky.

Tyto položky jsou obdobné jako u první úrovně menu. Jedná se o „title“, „presenter“ a „icon“. Jedinou položkou navíc je zde „action“. Do té je potřeba napsat název šablony, kterou má presenter zobrazit. Pokud by byla nějaká položka pouze jednoúrovňová (tedy neobsahovala klíč pro submenu), je potřeba „action“ zapsat již v sekci **menu**. Tedy o 1 úroveň výše. To zajistí vybrání správné šablony. V případě zápisu action do obou úrovní nabídky je upřednostněna akce presenteru vyžadující zobrazení šablony z druhé úrovně menu. V obou úrovních nabídky lze zapsat neomezený počet položek. Tento zápis menu v administraci umožňuje snadnou editaci, návaznost na strukturu MVC [9] a možnost budoucího rozšíření aplikace dle potřeb její obsluhy.



```
1 parameters:
2     menu:
3         User:
4             title: Uživatelé
5             presenter: User
6             module: User
7             action: default
8             icon:
9                 name: user
10            alias:
11                - User
12                - Role
13            submenu:
14                user:
15                    title: Uživatelé
16                    action: default
17                    presenter: User
18                    icon:
19                        name: user
20                role:
21                    title: Role
22                    action: default
23                    presenter: Role
24                    icon:
25                        name: magnet
```

Obr. 10 - Definice položek v menu pomocí konfiguračního souboru

3.3.4 Backend - uživatelé

V následující části práce budou rozebrány jednotlivé sekce v menu, které poskytuje UserModule správcovského rozhraní vytvořeného programu. Detailně tedy budou zmíněny seznam a tvorba uživatelských přístupů a seznam tvorba rolí pro uživatele. Tyto funkčnosti zajišťují jednoduchou správu webovou aplikací pro jejího operátora.

3.3.4.1 Založení uživatele

V seznamu uživatelů jsou zobrazeny záznamy obsažené v databázové tabulce *user*. Seznam uživatelů je zobrazen v grid tabulce. Je tedy umožněno filtrovat a řadit záznamy dle potřeb operátora. Filtrace a řazení ve všech tabulkách typu grid je v rámci tohoto projektu ukládáno do session uživatele. Hodnoty filtrace a řazení jsou tak zapamatovány pro opakované návštěvy grida. Svázáním se session je zajištěno pamatování si nastavení filtrace a řazení pro každého návštěvníka zvlášť. Nedochozí tedy k situacím, kdy si 1 uživatel vyfiltruje určitá data a jiný uživatel mu dané parametry filtrace přepíše svými parametry.

V seznamu uživatelů jsou zobrazeny základní sloupce z tabulky *user*. Jedná se o uživatelské jméno, příjmení, jméno a stav uživatele. Stav uživatele je možné měnit ze stavu **aktivní** na stav **neaktivní**. Jako výchozí nastavení filtru je zvoleno zobrazení pouze aktivních uživatelů. Uživatel může měnit stav pouze jiným uživatelům. Sám sobě tuto akci provést nelze, došlo by totiž k chybě v ověření identity uživatele. Pro úspěšné přihlášení je totiž potřeba, aby byl stav uživatele nastaven na hodnotu aktivní. Stav uživatele je v programu přidán zejména kvůli možnosti blokovat již nepoužívané uživatelské přístupy. Lze ho měnit pomocí šedivého tlačítka s ikonou dvojité šipky v každém řádku grida. Druhou akcí grida je možnost editovat uživatelský účet. Tato možnost je zobrazena po kliknutí na modré tlačítko s ikonou tužky. Stejně jako velké zelené tlačítko „Přidat uživatele“ v pravé horní části podstránky nás odkáže na formulář pro tvorbu/editaci uživatele. O tom, zda se ve formuláři mají předvyplnit data anebo se jedná o založení nového uživatele, rozhoduje akce presenteru. Pokud je URL adresa zakončena řetězcem **/create*, jedná se o založení nového uživatele. Pokud je v URL adrese uveden řetězec **/edit/**, jedná se o editaci stávajícího uživatele. Za akcí „edit“ je uvedeno ID uživatele používané v databázi. Toto je následně použito pro přístup k záznamu uživatele z databáze.

Záznam je použit pro předvyplnění formuláře. Formulář se skládá z 6 polí. Je třeba zmínit minimálně heslo a role. Heslo je standardně zakódováno do textového řetězce. Zakódovaný řetězec je následně uložen do databáze. Uživatel má jistotu, že heslo je zabezpečené proti zneužití a zároveň se nemusí starat o nic dalšího. K přihlášení potřebuje pouze své uživatelské jméno a heslo, o následné zakódování či dekodování řetězce se postará autentifikační metoda objektu `\Nette\Security\User` [30]. Výběr role zajišťuje uživateli přístup ke zdrojům nadefinovaným v rámci role.

3.3.4.2 Založení role

Pokud v hlavní nabídce přejdeme do sekce „Role“, zobrazí se nám seznam rolí. Role jsou zobrazeny v tabulce typu grid. Jednotlivé záznamy jsou v databázi uloženy v tabulce `user_role`.

Každý záznam obsahuje 5 sloupců. Kromě ID (unikátní identifikátor) jsou to název role, popis role, akce a stav. Stav role umožňuje roli aktivovat či deaktivovat. Název role je důležitý při tvorbě uživatele. Je zde zobrazen operátorovi, aby bylo snáze identifikovatelné, k jakým zdrojům zajišťuje přístup. Název role je použit i v kódu, neměl by tedy obsahovat diakritiku, mezery ani jiné znaky, které nespádají do kategorie alfanumerických. Pro detailní rozlišení významu rolí slouží sloupec „description“ – popis role. Zde může operátor použít libovolné znaky a řetězce, pomocí kterých popíše, k čemu daná role slouží a k jakým zdrojům dává uživateli přístup. Posledním sloupcem je „action“. V akci je vypsán seznam zdrojů ve formátu `json`, ke kterým může uživatel s touto rolí přistupovat. V rámci této práce byly nadefinovány 3 základní role – `admin`, `basic` a `common_user`.

Uživatel s rolí **admin** má v rámci celého projektu neomezená práva. Může přistupovat ke všem zdrojům, lze jeho roli kombinovat s libovolnou další rolí a nedojde tak k žádné změně.

Role **basic** je pouze podpůrnou rolí pro ostatní role. Zajišťuje uživateli přístup na přihlašovací stránku, odhlašovací stránku, formulář pro změnu hesla a na homepage administračního rozhraní. Tato role byla nadefinována především pro jednodušší správu pro operátora. Ten tak nemusí při zakládání nových rolí vybírat zdroje, které už jsou nadefinované v roli **basic**. Postačuje, když uživateli přidá roli **basic** v kombinaci s nově vytvořenou rolí a přístup ke zdrojům je zajištěn. Tento návrh taktéž předchází problémům s vícenásobným přidáním jednoho zdroje v různých rolích kombinovaných u 1 uživatele.

Role **common_user** zajišťuje uživateli přístup ke třem zbývajícím položkám v menu administračního rozhraní – „Databáze“, „Návštěvy“ a „Storage“. Z návrhu rolí vyplývá, že je potřeba tuto roli kombinovat s podpůrnou rolí **basic**. Uživatel s kombinací těchto 2 rolí je oprávněn přistupovat v administraci do všech modulů, kromě UserModule. Do uživatelského modulu je dle výše zmíněného návrhu rolí oprávněn přistupovat pouze uživatel s rolí **admin**. Je tak zajištěna selektivita práv uživatelů.

3.3.5 Backend - databáze

Pokud uživatel vybere v menu položku „Databáze“, je mu zobrazen seznam záznamů z tabulky *database*. Nyní je třeba vysvětlit význam této tabulky. Už při návrhu aplikace bylo zmíněno, že je potřeba počítat s pozdějším napojením tohoto programu na odlišnou databázi. V současné době je projekt napojen na MySQL databázi technického projektu VZLU_{sat-1}. Z ní přistupuje k datům shromažďovaným v rámci tohoto projektu a zobrazuje uživatelům veškeré tabulkové či grafické výstupy. V budoucnu by mělo dojít k realizaci projektu navazujícího na VZLU_{sat-1}, tedy na zprovoznění družice, která půjde takřikajíc ve stopách současného VZLU_{sat-1}. Ten bude mít svojí vlastní MySQL databázi oddělenou od současného experimentu. Proto bylo potřeba, aby byla tato webová aplikace vytvořena s ohledem na budoucí připojení na více databází. Popis definice databázového spojení již byl zmíněn v předchozích částech práce (odstavec 3.1.5, popis konfiguračního souboru `config.local.neon`). Připojení k databázi je zajištěné i z rozhraní aplikace. Je tak možné dopředu nadefinovat konfigurační soubor a připojení k databázi „oživit“ až s odstupem času ve chvíli, kdy to bude operátor vyžadovat.

Záznam v tabulce lze nastavit jako neaktivní. K tomu lze využít akce v gridu nastavené na šedivé tlačítko s ikonou dvojité šipky. Deaktivování záznamu způsobí, že se ve veřejně přístupné části webu v podstránce WOD OBC nebude zobrazovat v nabídce výběru databáze pro plnění data do vykreslovaných grafů. Toto propojení dává operátorovi možnost pomocí několika kliknutí spravovat přístupy k databázím pro tento projekt. Operátor tím zároveň získává vyšší kontrolu i nad požadavky návštěvníků stránky. Záznam v gridu databází je možné rovněž editovat či přidat nový. K tomu slouží modré tlačítko s ikonou tužky respektive zelené tlačítko v pravé horní části podstránky. Tato tlačítka uživatele přesměrují na formulář pro tvorbu záznamu. Je třeba zde uvést přesný název databáze. Jedná se o úplné znění databáze na databázovém serveru. Do poznámky Uživatel může vyplnit krátký popis, o jakou databázi se jedná, s jakým projektem souvisí apod. Tím zpřehlední výběr ze selectboxu návštěvníkům webu při zadávání databáze grafů.

3.3.6 Backend – návštěvníci

Tato položka má v menu 2 úrovně. Na druhé úrovni si uživatel může vybrat seznam návštěvníků nebo logy návštěvníků. Proč je tato sekce v práci zahrnuta bylo už ve zkratce popsáno v předcházející části práce. Nyní bude problematika vysvětlena spíše z pohledu správce webu.

V rámci návrhu aplikace bylo stanoveno, že je potřeba příliš nezatěžovat databázi požadavky návštěvníků, aby byl zajištěn bezproblémový průběh ukládání dat přicházejících z družice. Proto byla stanovena omezení zajišťující stanovený hladký chod MySQL databáze. Jedním z omezení je stanovení maximální počtu požadavků na databázi během vykreslování grafů ve veřejně přístupné části webu. Počet požadavku je vztažen k unikátní IP adrese zařízení, ze kterého požadavek přichází a k časovému intervalu, za který je možné požadavky uskutečnit. Výchozím nastavením je 25 požadavků za 30 minut. Časový interval i počet požadavků lze měnit pro každou IP adresu zvlášť. Případně je vytvořena i možnost zablokovat požadavky z vybrané IP adresy trvale. Právě k tomu slouží administrace návštěvníků webu.

Seznam návštěvníků je tabulkou typu grid. Zobrazeny jsou záznamy z databázové tabulky *visitor*. V seznamu návštěvníků jsou uvedeny všechny IP adresy, ze kterých byl odeslán požadavek na vykreslení grafů. Pomocí šedivého tlačítka lze změnit stav IP adresy a dané zařízení tak trvale zablokovat či odblokovat. Ve výchozím nastavení jsou zobrazeny pouze IP adresy, které jsou aktivovány. Pomocí modrého tlačítka je možné danou IP adresu editovat. Editací je v tomto případě myšlena pouze změna počtu požadavků a počtu minut, za které lze požadavky provést. Samotnou IP adresu nelze editovat, což vychází z povahy toho, že se jedná o identifikátor spjatý se zařízením, nikoliv s touto webovou aplikací.

Další položkou v submenu jsou „Logy návštěv“ zobrazená pomocí tabulky typu grid. Přináší pouze náhled na všechny logované požadavky na vykreslení grafů z databázové tabulky *visitor_log*. Je zde uvedena IP adresa, datum a čas požadavku. Správce si tak zde může vyfiltrovat jednotlivé návštěvníky, seřadit si jejich požadavky v určitém rozmezí datumů a získat tak podklady k editaci IP adresy v seznamu návštěvníků. Tím je administrátorovi dán do rukou dostatečný nástroj pro správu webu a ochranu databáze v případě nouze. Nebo případná možnost rozšíření přístupových práv, uzná-li to za vhodné a bezpečné vzhledem k výše zmíněným omezením a parametrům vybrané MySQL databáze.

3.3.7 Backend – storage, homepage

V této části administrace je převzato zobrazení sekce „STOR info“ z původní aplikace. V ní byla tato část zahrnuta ve veřejné sekci, což u dat, která mohou ovlivňovat pouze operátoři, není žádoucí. Jsou zde zahrnuta data z databázových tabulek *s1p52*, *s1p41*, *s1p34*, *s1p58*, *s4p3*, *s4p2* a *s6p2*. Výstupem jsou tabulky typu grid. Grida zobrazují stavy automatického stahování jednotlivých chunků družice. Ty jsou kontrolovány a stahovány na základě paměti předchozích stavů požadavků v rámci družice, aby nedocházelo ke stahování již stažených dat. V těchto gridech je možné filtrovat jednotlivé nadefinované stavy.

Po přihlášení uživatele nebo po kliknutí na název projektu v administraci je zobrazena podstránka obsahující další část převzatou z původní verze aplikace – „COMM info“. Konkrétně box „Communication state“. V tomto boxu má operátor webu možnost přepínat stavy jednotlivých checkboxů. Tato část byla opět přesunuta do administrace, aby byla skryta běžnému návštěvníkovi stránek. Celá podstránka se skládá z jednoduchého formuláře s jednotlivými checkboxy, které je možné zaškrtnout či nechat prázdné a po odeslání formuláře tak změnit požadované stavy.

3.4 Výhody a nevýhody aplikace, srovnání s předchozím webem

V této části práce bude porovnán současný stav aplikace s aplikací vytvořenou v rámci této diplomové práce. Budou rozebrány jednotlivé výhody, nevýhody a slabiny programů.

3.4.1 Backend

Správa původní webové aplikace je prováděna operátory pozemní stanice. Vzhled stránek je shodný pro běžného návštěvníka webu i pro operátora. Toho může být zneužito při pokusu o napadení stránek. Jednou z technik, které k tomu slouží je takzvaný cross-site scripting, s jehož pomocí lze měnit obsah stránek nebo dokonce odeslat jiné hodnoty do formulářů [31]. Této technice lze zabránit escapováním hodnot. To zajišťuje automaticky šablonovací systém latte. Dalším způsobem je zabezpečit přístup k formulářům s nastavením důležitých stavů a dat v databázi přihlášením uživatele. Tento způsob pochopitelně není samospásný. Ale v kombinaci se silou šablonovacího systému latte dává operátorovi webu dostatečnou jistotou, že nedojde k napadení aplikace. Dalším důvodem, proč je dobré mít na webu administraci je, že jednotný vzhled uživatele zkrátka láká. Když návštěvník vidí checkbox s možností nastavit důležitá data, leckdy ho napadne checkbox zaškrtnout. Nadanějšímu programátorovi situace nedá spát a pokusí se změnit hodnoty i přes vyskakovací okno typu „confirm“, které mu sdělí, že změna hodnoty je dostupná pouze operátorovi. Skrytí těchto formulářů do administrace tedy působí i psychologicky. Další výhodou aplikace je možnost založit přístup do administrace více uživatelům. Následně je možné stanovit uživateli přístup jen k určitým položkám v menu administrace. To přináší možnost selektivně nastavit úroveň práv jednotlivým uživatelům. Tyto funkčnosti jsou zajištěny uživatelským modulem.

3.4.2 Zobrazení pro mobilní zařízení

Dalším důležitým parametrem je (nejen) v dnešní době responzivní design webu. Ta přináší pohodlí nejen běžnému návštěvníkovi, který je zvyklý pohybovat se na internetu nejen ze stolního počítače, ale i ze zařízení přenosného charakteru. Mobilní zařízení mají samozřejmě širokou škálu velikosti displejů, rozlišení apod. S tímto bohužel původní web neumí příliš pracovat. Ve všech zařízeních je zobrazován stejně, uživatel je nucen v mobilním provedení přejíždět prstem či využít přiblížení/oddálení pro zobrazení kompletní podstránky. Pohodlí pro návštěvníka ovlivňuje i chuť vracet se na stránky v budoucnu. Responzivní vzhled programu vytvořeného v rámci diplomové práce je plusem i pro samotné operátory. Bude pro ně jistě snazší upravit nastavení webu například na cestách a nebudou muset toto odkládat až do kanceláře či složitě brouzdat na desktopové verzi webu přes mobilní zařízení.

3.4.3 Napojení na více databází

Jedním z požadavků na program byla možnost napojení na jinou databázi. Především z důvodů dosluhující družice VZLU_{sat-1} a přípravy nové družice je potřeba, aby byla aplikace připravena zobrazovat data i z jiné aplikace. S využitím objektového programování a dědičnosti umožňující definovat připojení k databázi na jednom místě a následně pouze používat službu, která toto obstará, byl připraven konfigurační soubor s parametry pro připojení k databázi. V kombinaci s administračním rozhraním umožňujícím aktivovat připojení k databázi je tak operátorovi dán dostatečný nástroj pro pozdější přesun k jiné databázi, případně k připojení více databází. Tato možnost dává i běžnému návštěvníkovi veřejně dostupné části stránek možnost vybrat si, ze které z aktivních databází bude čerpat data pro vykreslení grafů.

3.4.4 Zajištění běhu databáze

Omezením požadavků na databázi je zajištěn chod databáze bez možnosti jejího pravidelného přetěžování. Pokud by nějaký návštěvník zkoušel databázi vytížit nebo zkrátka jen příliš prahnul po zobrazení dat, bude odkázán, aby požadavky vyzkoušel později. Počet požadavků a čas prodlevy je navíc možné měnit individuálně pro každou IP adresu přes správcovské rozhraní. Případně je možné vybranou IP adresu blokovat trvale. Tato možnost dává operátorovi do ruky další zbraň, jak může regulovat provoz například v době přeletů družice, kdy je potřeba rychlá odezva databáze. Navíc má operátor k dispozici i historii požadavků, aby měl relevantní podklady pro případné omezení provozu IP adres.

Jako negativní stránku nově vytvořeného programu je potřeba zmínit, že tato omezení požadavků na databázi nejsou samospásná. Jsou velmi účinná při zvýšeném počtu požadavků jednotlivých uživatelů za určitý časový limit. Pokud ale budou na web přistupovat řádově stovky uživatelů v jeden moment, může docházet ke zpomalování běhu. Nicméně na základě současné návštěvnosti webu a velmi nízké pravděpodobnosti, že takto vysoké množství uživatelů bude využívat zobrazování grafů v rozmezí pár sekund, lze předpokládat, že k tomuto jevu téměř nebude docházet.

3.4.5 Zobrazení dat na webové stránce

Data jsou na jednotlivých podstránkách zobrazována různými způsoby. Především jsou využity tabulky typu grid. Ty umožňují filtrovat a řadit data přesně dle požadavků uživatele. Velkou změnou prošlo i zobrazování grafů. V původní verzi webu bylo na podstránce zobrazeno několik boxů s grafy, navíc byl rozsah dat poměrně omezený, především z výkonových důvodů. Současná verze aplikace zobrazuje pouze 1 graf na podstránce, ten je tak větší a poskytuje uživateli větší přehled. Dále nezobrazuje data ihned při příchodu na podstránku, ale až při odeslání formuláře uživatelem. Tím je zajištěno rychlejší první načtení podstránky, jelikož ta nemusí ihned požadovat data z databáze. Výhodou je taktéž formulář, který dovoluje nastýlovat zobrazenou křivku, přidat jí vlastní popisek, pojmenovat graf či osy, zobrazit legendu apod. Tím je dosaženo variability grafů pro uživatele. Pokud si například bude chtít uživatel připravit graf s popisky v cizím jazyce pro prezentaci před cizojazyčným obecnstvem, může si vše nadefinovat dle potřeb svých či potřeb posluchačů. Tato pole jsou nepovinná, čímž je dosaženo vyššího komfortu pro uživatele.

Ohledně volby dat je nutné zmínit, že uživatel má omezenou možnost na rozmezí pouze 8 dní. Toto se může zdát jako malé číslo. Je ale potřeba pamatovat na skutečnost, že v průběhu tohoto času je zaznamenáno zhruba 11 000 záznamů do databáze. Přístup k takto vysokému množství záznamů je už sám o sobě náročný na výkon. Nicméně jedná se o kompromis, který by měl zároveň uspokojit potřeby většiny návštěvníků při zachování bezproblémového běhu MySQL databáze. Již bylo zmíněno, že data je možné navolit z různých databází (po dokončení budoucí družice). Ohledně dat je ještě třeba zmínit, že návštěvníci nejsou ochuzeni počtem grafů oproti předchozí verzi webu. Sledovanou fyzikální veličinu, kterou požaduje uživatel zobrazit, si může zvolit pomocí selectboxu. To také zajišťuje přehlednost webu. Uživateli nejsou zobrazovány redundantní grafy, které si v aktuální moment nepřeje vidět. Nezatěžuje se tedy těmito požadavky zbytečně databáze a uživatel má graf zobrazen komfortně přes celý obsah boxu. Ostatní fyzikální parametry si může prohlédnout později.

Tímto byly nastíněny některé nevýhody a omezení v zobrazování grafů. Pro srovnání s původní verzí webu je třeba uvést, jak zobrazení grafů funguje v této aplikaci. Původní program je taktéž omezen rozmezím datumů. Jsou zobrazována data za poslední den. Data není možné změnit, tedy není možné se podívat například na den předchozí či rok zpátky. Zpětné zobrazení dat nová aplikace umožňuje, rozmezí dnů je navíc vyšší (8 dnů). Grafů je zde zobrazeno několik najednou, což může snižovat přehlednost aplikace. Navíc jsou grafy „plněny“ daty již při prvním příchodu na stránku, což způsobuje pomalejší načtení podstránky.

Dalším parametrem je takzvané „oživení grafů“. Původní web po najetí kurzorem myši na křivku grafu je zobrazen titulek s hodnotami vybraného bodu na osách grafů. Nová verze aplikace vytvořená v rámci této práce též zobrazuje titulek s hodnotami ve zvoleném bodě. Přináší ovšem vylepšení ve větším množství nástrojů pro prozkoumání grafu. Je zde možnost přiblížit či oddálit celý graf, přiblížit pouze uživatelem vybraný výřez grafu, posunout sledovanou oblast grafu či možnost autoskle, která provede automatické zobrazení celé oblasti dat. Při všech těchto operacích je zajištěna rychlá odezva a přepočtení hodnot na osách grafu podle momentálně zobrazeného měřítka. Navíc je zde zahrnuta i možnost „Download plot as png“. Tento nástroj vystřihne z obrazovky právě zobrazenou oblast křivky v grafu a stáhne ho uživateli do zařízení. Návštěvník si tak může jednotlivé grafy a jejich nastavené filtrace dat jednoduše uložit k sobě do PC či mobilu pro pozdější využití.

3.4.6 Fotogalerie

V rámci jedné z podstránek veřejně dostupné části práce nabízí vylepšenou možnost zobrazení obrázků pomocí fotogalerie. Jednotlivé obrázky jsou zobrazeny samostatně v modálním okně ve větší velikosti, což původní verze aplikace neumožňuje. Navíc je možné v modálním okně přepínat mezi jednotlivými obrázky. Tato funkčnost přináší větší pohodlí a přehlednější zobrazení obrázků pro návštěvníka při zachování rychlého načtení podstránky.

Celkově tak lze říci, že aplikace přináší pro běžného návštěvníka stránek nejen výhody v podobě pohodlného zobrazení na všech zařízeních. Přináší i lepší a četnější možnosti zobrazení dat, tabulek a grafů doprovázené zvýšením přehlednosti zobrazení. Operátorovi poskytuje obdobné pokroky a navíc přidává přehlednější a jednodušší správu aplikace pomocí správcovského rozhraní, ale i přehled o požadavcích na databázi a možnosti regulace požadavků s ohledem na zajištění bezproblémového běhu MySQL databáze při vylepšeném zobrazování dat.

Závěr

V rámci této diplomové práce jsem provedl analýzu webové aplikace zobrazující data z reálného technického experimentu VZLUsat-1. Na základě analýzy jsem vytvořil návrh nové aplikace. Podle tohoto návrhu jsem posléze program i vytvořil a zdokumentoval.

V rámci analýzy předchozí verze programu jsem nejdříve prostudoval rozhraní a platformy, na jejichž základě je program vytvořen. Následně jsem vyhodnotil jeho kladné a záporné stránky. Z kladných stránek byla zmíněna především informační hodnota webové stránky, struktura databáze a přehlednost webu. Tyto parametry byly přeneseny i do návrhu a realizace nové aplikace. Webové stránky měly mezery především v chybějící verzi zobrazení pro přenosná zařízení, obtěžující aktualizaci s přesměrováním na domovskou stránku, v zobrazování grafů a jednotném vzhledu stránek pro operátora i běžného návštěvníka. Mezi drobné nedostatky byla zahrnuta i fotogalerie s příliš malými rozměry obrázků a absence routování URL adresy, které způsobují sníženou přívětivost programu pro uživatele.

Návrh nového řešení jsem založil na předešlé analýze. Kladné stránky původního řešení jsem se snažil zachovat, v některých případech jsem navrhnul vylepšení. To se týká zejména nadefinování indexů a klíčů do databázových tabulek. Ty byly definovány zejména pro urychlení procházení databáze. V této části práce jsem taktéž navrhnul uživatelské rozhraní pro operátory s přihlášením a možností definovat si přístupy k jednotlivým zdrojům pomocí uživatelských rolí. Do tohoto rozhraní byly přesunuty formuláře určené pouze pro operátory. Tím byla zvýšena bezpečnost aplikace a dat v databázi. V rámci návrhu jsem vybral platformu podporující routování URL adres, přehledné členění kódu pomocí architektury MVC a je postavená na kombinaci programovacích jazyků PHP, HTML, CSS či MySQL. Platformou je takzvaný Nette Framework.

Byl navržen i vzhled celé aplikace. Vzhled bylo třeba rozdělit na 2 základní části – veřejně přístupnou a administrační. Následně byl vybrán vhodný nástroj pro zobrazování grafů pomocí jazyka javascript – Plot.ly. Byla navržena i vylepšení fotogalerie, aby návštěvníci stránek mohli zobrazit obrázky v plné velikosti.

Posledním navrženým vylepšením je databázové spojení. Nejdříve bylo navrženo databázové spojení realizované pomocí objektů, čímž je umožněno řídit připojení k databázi pouze z jednoho místa kódu. Poté bylo navrženo připravení aplikace na budoucí připojení k více databázím – to je snáze realizovatelné díky připojení k databázi pomocí vhodného objektu. Tento požadavek vzešel od operátora současné verze webu. Dalším požadavkem operátora bylo zajistit bezproblémový chod databáze. Aby návštěvník svými požadavky například na vykreslení grafů příliš nevytěžoval databázi, což by mohlo způsobit zpomalení jejího běhu. Proto jsem navrhnul omezení počtu požadavků každého návštěvníka spravované samotným operátorem.

V poslední části jsem naprogramoval kompletní aplikaci včetně jejího popisu. Zahrnul jsem vylepšenou fotogalerii, vytvořil mobilní zobrazení webu přepínané pomocí CSS techniky @media. Ve veřejně přístupné části webu jsem přidal i omezení uživatelských požadavků na zobrazování grafů. Správce tak má možnost zablokovat či omezit požadavky pocházející z určité IP adresy. K tomu mu jako nástroj poslouží možnost prohlížení historie požadavků IP adres. Následně jsem připravil možnost připojení k více databázím. V budoucnu si tak návštěvníci webu budou moci vykreslovat grafy nejen z databáze experimentu VZLUsat-1. Samotné grafy prošly velkou proměnou. Nyní je možné zobrazovat data i zpětně upravovat vzhled grafů, zoomovat vybrané oblasti, odečítat hodnoty v bodě či stáhnout a uložit si daný graf k sobě do počítače.

V administrativním rozhraní jsem vytvořil uživatelský modul umožňující vytváření uživatelských účtů (pro případ více operátorů), přihlášení, možnost přidělit uživateli přístup jen k vybraným zdrojům či sledovat historii požadavků návštěvníků stránek. Dále jsem do rozhraní přesunul formuláře sloužící operátorům, jak bylo zmíněno v návrhu aplikace.

Seznam literatury a informačních zdrojů

- [1] "Ajax programming." [Online]. Available: [https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming)).
- [2] "Colorzilla editor." [Online]. Available: <https://www.colorzilla.com/gradient-editor/>.
- [3] "Admin LTE 2." [Online]. Available: <https://adminlte.io/themes/AdminLTE/index2.html>.
- [4] "Bootstrap 3." [Online]. Available: <https://getbootstrap.com/docs/3.3/>.
- [5] "Nextras/Datagrid." [Online]. Available: <https://componette.com/nextras/datagrid/>.
- [6] "Ublaboo/Datagrid." [Online]. Available: https://ublaboo.org/datagrid/?examplesGrid-per_page=10&examplesGrid-filter%5Bstatus%5D%5B0%5D=1.
- [7] "Plot.ly." [Online]. Available: <https://plot.ly/~stejul/69/python-javascript-php/#/data>.
- [8] "Nette Framework." [Online]. Available: <https://nette.org/cs/>.
- [9] "MVC architecture." [Online]. Available: <https://doc.nette.org/cs/2.4/presenters>.
- [10] "Nette Latte." [Online]. Available: <https://latte.nette.org/cs/>.
- [11] "Kravcik/Core." [Online]. Available: <https://github.com/kravcik/core>.
- [12] "Uestla/YetORM." [Online]. Available: <https://github.com/uestla/YetORM>.
- [13] "Nette Context." [Online]. Available: <https://api.nette.org/2.4/Nette.Database.Context.html>.
- [14] "Htaccess." [Online]. Available: <https://www.jakpsatweb.cz/server/htaccess.html>.
- [15] "Composer." [Online]. Available: <https://getcomposer.org>.
- [16] "Tracy." [Online]. Available: <https://tracy.nette.org/cs/>.
- [17] "Nette autoloading." [Online]. Available: <https://doc.nette.org/cs/2.3/auto-loading>.
- [18] "Nette sessions." [Online]. Available: <https://doc.nette.org/cs/2.4/sessions>.
- [19] "Nette Framework sessions." [Online]. Available: <https://www.zdrojak.cz/clanky/nette-framework-sessions/>.
- [20] "Nette Configurator." [Online]. Available: <https://doc.nette.org/cs/2.2/configuring#toc-configurator>.
- [21] "Neon." [Online]. Available: <https://github.com/nette/neon/releases>.
- [22] "Nette extension." [Online]. Available: <https://doc.nette.org/cs/2.2/di-extensions>.
- [23] "Webloader." [Online]. Available: <https://componette.com/janmarek/webloader/>.
- [24] "Nette routing." [Online]. Available: <https://doc.nette.org/cs/3.0/routing>.
- [25] "VZLUsat-1." [Online]. Available: <https://www.pilsencube.zcu.cz/vzlsat1/>.
- [26] "Responsive pure css off canvas-hamburger menu." [Online]. Available: <https://medium.com/@heyoka/responsive-pure-css-off-canvas-hamburger-menu-aebc8d11d793>.
- [27] "Bootstrap 3 grid options." [Online]. Available: <https://getbootstrap.com/docs/3.4/css/#grid-options>.
- [28] "Modul-IS." [Online]. Available: <https://modul-is.cz>.
- [29] "FA icons." [Online]. Available: <https://fontawesome.com/v4.7.0/icons/>.
- [30] "Nette @ api.nette.org." [Online]. Available: <https://api.nette.org/2.4/Nette.Security.User.html>.
- [31] "Co je XSS a jak mu předcházet." [Online]. Available: <https://www.zdrojak.cz/clanky/co-je-xss-jak-mu-predchazet/>.