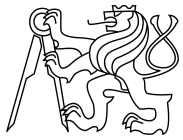
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Vizualizační komponenta do CMS Drupal - verze 2.0
Student:	Bc. Jan Píša
Vedoucí:	Ing. Jindřich Fiedler
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

- Analyzujte současné řešení vizualizační komponenty do CMS Drupal a proveďte revizi, redesign a reimplementaci této komponenty s ohledem na požadovanou funkcionalitu.
- Proveďte revizi systému tvorby uživatelských obrazovek/dashboardů a šablonovacího systému těchto dashboardů.
- Proveďte revizi systému modulárních rozšíření/widgetů se zaměřením na usnadnění tvorby a začlenění nových widgetů.
- Analyzujte a navrhněte nové UI pro práci s dashboardy a konfiguraci widgetů s důrazem na intuitivní a snadné ovládání.
- Proveďte analýzu, návrh a implementaci nových widgetů zaměřených na vizualizaci dat v tabulkách, grafech a mapách dle požadavků zadavatele.
- Otestujte novou verzi komponenty v prostředí vedoucího práce/zadavatele.

Použité technologie: PHP, MySQL, jQuery, JSON, CMS Drupal.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 18. ledna 2016

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra softwarového inženýrství

Diplomová práce

Vizualizační komponenta do CMS Drupal - verze 2.0

Bc. Jan Píša

Vedoucí práce: Ing. Jindřich Fiedler

4. května 2016

Poděkování

Tímto chci poděkovat vedoucímu práce Ing. Jindřichu Fiedlerovi za trpělivost, cenné rady a celkové usměrnění při tvorbě mé diplomové práce. Dále chci poděkovat oponentovi práce Ing. Michalu Valentovi Ph.D. za konzultaci a užitečné rady.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 4. května 2016 .....

České vysoké učení technické v Praze
Fakulta informačních technologií
© 2016 Jan Píša. Všechna práva vyhrazena.
Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Píša, Jan. *Vizualizační komponenta do CMS Drupal - verze 2.0.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Tato práce analyzuje současné řešení vizualizační komponenty a následně navrhuje provedení změn na základě této analýzy. Proveden je také návrh nového uživatelského rozhraní, které vyhovuje standardům moderních webových aplikací. Součástí práce je i popis nejzajímavějších problému řešených během implementace a následného uživatelského testování. Hlavním přínosem práce je rozsáhlá analýza problematiky uživatelských obrazovek, která může být použita při návrhu a implementaci podobných systémů.

Klíčová slova uživatelské obrazovky, vizualizace dat

Abstract

This master's thesis analyses the existing solution of the visualization component and subsequently proposes modifications based on this analysis. These modifications follow current trends in dashboarding. New user interface that meets the standards of modern Web applications is also designed. Furthermore, the paper contains description of the most interesting problems solved during implementation and subsequent usability testing. The main benefit of this paper remains extensive analysis of the dashboarding issue. This analysis can be used to design and develop similar applications.

Keywords dashboards, data visualisation

Obsah

Úvod **1**

1 Cíle a požadavky projektu **3**
1.1 O společnosti FIEDLER AMS 3
1.2 Popis současného stavu 3
1.3 Rozbor cílů projektu 5
1.4 Konzultace zadání se zadavatelem 6

2 Analýza a návrh **7**
2.1 Analýza současného řešení vizualizační komponenty 7
2.2 Koncept uživatelských obrazovek a trendy v této oblasti 15
2.3 Využití existujícího řešení uživatelských obrazovek 20
2.4 Technologie pro reimplementaci vizualizační komponenty 22
2.5 Systém tvorby uživatelských obrazovek 23
2.6 Systém modulárních rozšíření 24
2.7 Systém rozložení obrazovek 26
2.8 Vizualizace dat pomocí grafů 27
2.9 Analýza případů užití 33
2.10 Analýza požadavků 37
2.11 Doménový model 42
2.12 Návrh uživatelského rozhraní 44

3 Implementace **49**
3.1 ORM 49
3.2 Uživatelské obrazovky 49
3.3 Oblasti 49
3.4 Vstupní brána 50
3.5 Bloky obrazovky 50
3.6 Problémy řešené při implementaci 51

4 Testování **53**
4.1 Uživatelské testy 53
4.2 Akceptační testy 59

Závěr **61**

Literatura **63**

A Seznam použitých zkratek **67**

B Slovníček pojmů **69**

C Instalační příručka **71**
C.1 Úvod 71
C.2 Instalace CMS Drupal 7 71
C.3 Instalace komponenty 71
C.4 Před použitím 72

D Uživatelská příručka **73**
D.1 Úvod 73
D.2 Práce s oblastmi 73
D.3 Práce s obrazovkami 73
D.4 Práce s výchozími obrazovkami 76

E Uživatelské testy **79**
E.1 Dotazník před testem 79
E.2 Slovníček pojmů 79
E.3 Registrovaný uživatel 79
E.4 Registrovaný uživatel s právy editace 80
E.5 Dotazník po testu 80

F Obsah přiloženého CD **81**

Seznam obrázků

1.1 Původní aplikace 4
1.2 Nová aplikace využívající CMS Drupal 4

2.1 Přehled dostupných obrazovek spolu s jejich šablonami 8
2.2 Výběr typu nového bloku 10
2.3 Úprava pozice bloků na obrazovce 11
2.4 Úprava nastavení bloku 13
2.5 Příklad přehledové obrazovky 17
2.6 Příklad analytické obrazovky 17
2.7 Příklad hybridní obrazovky 18
2.8 Příklad častého způsobu výběru bloku 19
2.9 Příklad častého způsobu nastavení bloku 19
2.10 Ukázka nástroje Dashing 21
2.11 Ukázka nástroje RazorFlow 21
2.12 Ukázka nástroje ClicData 22
2.13 Diagram znázorňující hierarchii rozšíření 25
2.14 Ukázka knihovny C3js 29
2.15 Ukázka knihovny NVD3 30
2.16 Ukázka knihovny Plotly 32
2.17 Diagram případů užití 38
2.18 Diagram doménového modelu 43
2.19 Návrh obrazovky 44
2.20 Návrh obrazovky v režimu úprav 45
2.21 Návrh konfigurace bloku 46

3.1 Ukázka obrazovky aplikace 50

4.1 Pohlaví zůčastněných testerů 53
4.2 Úroveň znalosti angličtiny zůčastněných testerů 54
4.3 Věkové skupiny zůčastněných testerů 54
4.4 Rozložení výsledků metody SUS 56

4.5 Výsledky metody SUS 56

D.1 Formulář pro vytvoření nové obrazovky 74
D.2 Obrazovka v režimu úprav 75
D.3 Nastavení bloku 75
D.4 Přehled výchozích obrazovek 77

Seznam tabulek

2.1 Tabulka výskytu rozvržení 27

Úvod

Práce se zabývá analýzou, návrhem, implementací a testováním vizualizační komponenty pro CMS Drupal. Hlavním zaměřením komponenty je zobrazení naměřených telemetrických dat formou uživatelských obrazovek. Mezi další funkce aplikace se řadí hromadná správa vytvořených obrazovek a jejich organizace do stromové struktury oblastí.

Hlavním přínosem komponenty je možnost sestavit si vlastní pohledy na naměřená data. Dále je také připraven komplexní systém modulárních rozšíření, v rámci kterého jsou následně jednotlivá rozšíření implementována. Tato rozšíření umožňují vykreslení požadovaných dat ve formě textu, tabulky, grafu nebo mapy.

V rámci rozsáhlé analýzy je zhodnoceno současné řešení komponenty. Zkoumán je také koncept a trendy v oblasti uživatelských obrazovek. Na základě těchto poznatků je dále proveden nový návrh systému uživatelských obrazovek, systému modulárních rozšíření i nový návrh uživatelského rozhraní. Mimo jiné je provedeno srovnání několika nejvhodnějších JavaScriptových knihoven s otevřenou licencí pro tvorbu dynamických grafů.

Další část této práce se věnuje zajímavým problémů řešeným během implementace a popisu základních součástí aplikace.

Provedeno je také uživatelské testování použitelnosti. Na základě výsledků tohoto testování jsou definovány problémy uživatelského rozhraní a návržena řešení těchto problémů.

Kapitola 1

Cíle a požadavky projektu

Tato kapitola pojednává o úvodu do problematiky této práce. Zadání práce je zde detailně rozebráno a vysvětleno.

1.1 O společnosti FIEDLER AMS

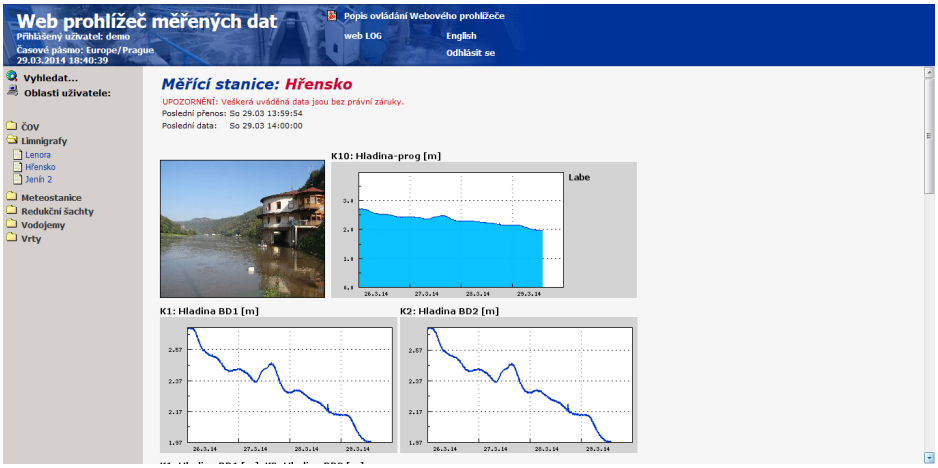
Společnost FIEDLER AMS[1] se již 25 let zabývá výrobou a prodejem telemetrických stanic a vývojem kompletních řešení pro sběr i analýzu dat s návazností na řízení procesů v rámci průmyslových provozů. Mezi vyráběná zařízení se řadí hladinoměry, průtokoměry, pH-metry, oxymetry, konduktometry, meteorologické i limnigrafické stanice.

V současné době má společnost FIEDLER AMS ve správě několik tisíc telemetrických stanic, které jsou umístěny nejen na území České republiky, ale také v mnoha dalších státech po celém světě. Mezi významné odběratele služeb patří mimo jiné Český hydrometeorologický ústav, který využívá limnigrafických stanic pro monitorování hladin a průtoků českých řek.[2]

1.2 Popis současného stavu

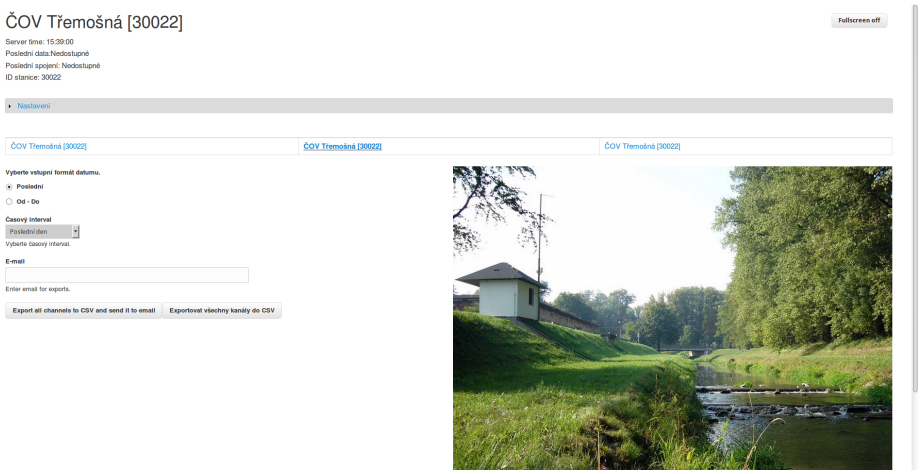
V současné době je v ostrém provozu využívána webová aplikace pro prezentaci dat psaná v jazyce PHP. Ukázka této aplikace je na obrázku 1.1. Tato aplikace nabízí zobrazení statických grafů a tabulek s nastavením časových parametrů. Dále je implementována možnost exportovat data do několika formátů. Mimo jiné také nabízí zobrazení několika stavů měřící stanice a informace o datový přenosech stanice.

Aplikace pokrývá většinu zákaznických požadavků, nicméně existují i požadavky, kterým aplikace momentálně není schopna vyhovět. Úprava stávající aplikace byla shledána jako neefektivní řešení z důvodu zastaralé architektury. Jako řešení byl zvolen vývoj nové aplikace, postavené na základech CMS Drupal 7.



Obrázek 1.1: Původní aplikace

Tato nová aplikace využívající CMS Drupal je aktuálně dostupná v testovacím provozu pro vybrané klienty a v běžném provozu na několika lokalitách. Podstatnou součástí této aplikace je vizualizační komponenta vyvíjená v rámci mé bakalářské práce. Ukázku této komponenty zobrazuje obrázek 1.2.



Obrázek 1.2: Nová aplikace využívající CMS Drupal

Vzhledem ke změnám některých požadavků v rámci rychlého vývoje a nevhodnému návrhu systému modulárních rozšíření, jsem se rozhodl v rámci mé diplomové práce provést analýzu, revizi, redesign a reimplementaci této komponenty.

1.3 Rozbor cílů projektu

V této části jsou podrobně rozebrány jednotlivé části zadání.

Analyzujte současné řešení vizualizační komponenty do CMS Drupal a proveďte revizi, redesign a reimplementaci této komponenty s ohledem na požadovanou funkcionalitu. Bude provedena analýza současné vizualizační komponenty pro CMS Drupal a následně provedena revize, redesign a reimplementace této komponenty s ohledem na funkční požadavky aplikace, současné webové standardy a trendy v oblasti webových technologií. Zvláštní důraz bude kladen na přehledné uživatelské rozhraní a pozitivní uživatelskou zkušenost.

Proveďte revizi systému tvorby uživatelských obrazovek / dashboardů a šablonovacího systému těchto dashboardů. Bude provedena analýza současného řešení tvorby dashboardů a šablonovacího systému v porovnání k současným trendům v oblasti dashboardů. Na základě výsledků této analýzy bude proveden nový návrh systému dashboardů a jeho následná implementace.

Proveďte revizi systému modulárních rozšíření / widgetů se zaměřením na usnadnění tvorby a začlenění nových widgetů. Bude provedena analýza současného řešení widgetů a proveden nový návrh tohoto systému v rámci zvolených technologií. Hlavním cílem této revize bude usnadnění vývoje nových rozšíření a jejich následného začlenění do systému dashboardů.

Analyzujte a navrhněte nové UI pro práci s dashboardy a konfiguraci widgetů s důrazem na intuitivní a snadné ovládání. Bude provedena analýza současného uživatelského rozhraní s ohledem na nejčastěji prováděné uživatelské scénáře. Následně bude navrženo nové uživatelské rozhraní, které bude dodržovat principy a standardy pro návrh moderního a intuitivního uživatelského rozhraní.

Proveďte analýzu, návrh a implementaci nových widgetů zaměřených na vizualizaci dat v tabulkách, grafech a mapách dle požadavků zadavatele. Bude provedena analýza současného řešení vizualizace dat v tabulkách, grafech a mapách. Následně bude proveden návrh a implementace widgetů pro tyto typy vizualizace dat.

Otestujte novou verzi komponenty v prostředí vedoucího práce / zadavatele. Po dokončení implementace komponenty proběhne akceptační a uživatelské testování v prostředí pracoviště zadavatele.

1.4 Konzultace zadání se zadavatelem

Uživatelské obrazovky budou v rámci aplikace využívány k sestavení vizualizace dat z několika druhů měřících stanic za účelem analýzy aktuálního stavu měřených veličin či zpětné analýzy a reportování. Hlavními uživateli budou pracovníci zákaznických společností zadavatele, kteří budou nastavovat uživatelské obrazovky pro své stanice a následně sledovat aktuální i historicky naměřené hodnoty. Dalšími uživateli budou přímo pracovníci zadavatele, kteří budou využívat rozhraní aplikace pro nastavení obrazovek k jakékoliv stanici v rámci zákaznické podpory a dále nastavení výchozích uživatelských obrazovek pro vybrané kategorie. Veřejně přístupné obrazovky budou určeny pro širokou veřejnost se zájmem o stavy některých měřených veličin, například stavy vodních toků.

Uživatelská obrazovka by měla obsahovat nastavitelné rozvržení bloků ve formě tabulek, grafů a map s možností snadného budoucího rozšíření o další druhy. V rámci návrhu systému modulárních rozšíření je potřeba zohlednit také budoucí zapojení nově vyvíjené JavaScriptové aplikace, která je určena ke vzdálené konfiguraci stanic a zobrazení aktuálních dat v rámci technologických obrazovek.

Při práci s tvorbou modulárních obrazovek je potřeba připravit možnost snadného kopírování již vytvořených obrazovek a dále možnost hromadné úpravy obrazovek. Tyto možnosti by měly co nejvíce usnadnit vytváření více totožných obrazovek a jejich následné hromadné úpravy.

Aplikace by po provedení revize měla být dostupná v rámci modulu pro CMS Drupal 7, způsobem, kterým ji bude možné začlenit do již existujícího ekosystému ostatních modulů. Dále by měla být schopna fungovat jako samostatně oddělitelná komponenta, například pro účely lokálních dispečinků.

Kapitola 2

Analýza a návrh

Tato kapitola zahrnuje rozsáhlou analýzu současného řešení i koncept a trendy v oblasti uživatelských obrazovek. Dále je proveden průzkum již existujících částečných nebo úplných řešení vyhovujících zadaným požadavkům. Součástí kapitoly je také porovnání otevřených knihoven pro tvorbu dynamických grafů. Závěr kapitoly je věnován analýze případů užití, doménového modelu a návrhu nového uživatelského rozhraní.

2.1 Analýza současného řešení vizualizační komponenty

2.1.1 Systém tvorby uživatelských obrazovek

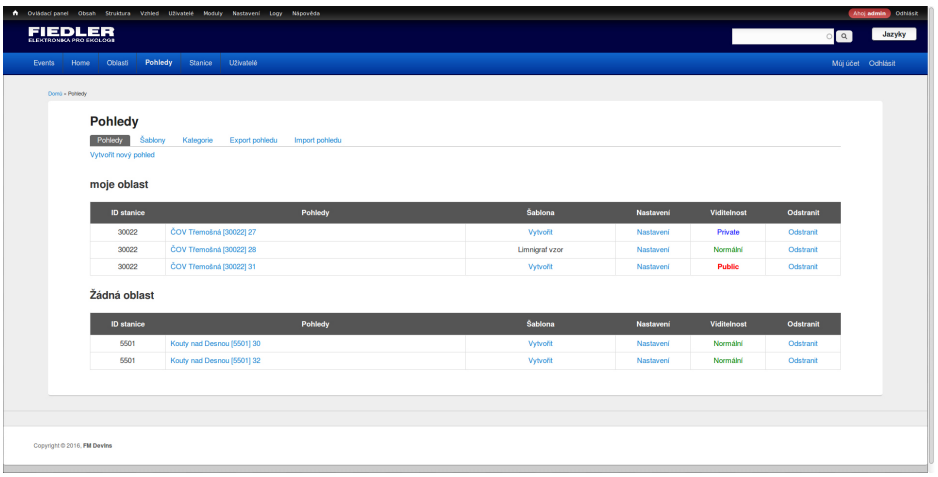
Základ systému tvorby uživatelských obrazovek převážně vychází z konceptu těchto obrazovek v rámci mé bakalářské práce. V systému došlo během dalšího vývoje k několika drobným změnám, které jsou v rámci této kapitoly zmíněny a vysvětleny.

2.1.1.1 Původní koncept

Uživatelské obrazovky jsou rozděleny do skupin se stejným rozložením a nastavením bloků. Při tvorbě nové obrazovky si uživatel může vybrat do které skupiny bude nově vytvořená obrazovka přiřazena. Po vytvoření této obrazovky je její rozložení i nastavení bloků shodné s celou skupinou. Při úpravě obrazovky uživatel dostává na výběr, zda chce výslednou změnu aplikovat na celou skupinu obrazovek, či zda touto změnou upravenou obrazovku od skupiny oddělí. Pokud je obrazovka od skupiny oddělena, je jí také přiřazena nová samostatná skupina.

2.1.1.2 Provedené změny

V rámci horšího pochopení výše zmíněného konceptu z pohledu uživatelů, byla provedena změna, po které nově vytvořené a oddělené obrazovky standardně nepatří do žádné skupiny. Při úpravě takovéto obrazovky nedochází k duplikaci provedených změn na žádnou další obrazovku. Uživatelé také dostali možnost z tohoto typu obrazovek vytvořit novou skupinu. Dále byla pro každou skupinu určena řídící obrazovka, která určuje rozvržení a nastavení celé skupiny. Úpravou jiné než řídící obrazovky je upravená obrazovka ze skupiny oddělena. Ukázku přehledu obrazovek zobrazuje obrázek 2.1.



Obrázek 2.1: Přehled dostupných obrazovek spolu s jejich šablonami

2.1.1.3 Související součásti systému

Systém uživatelských obrazovek je také úzce spjatý s komponentou pro kategorizaci jednotlivých stanic do stromové struktury oblastí. V této struktuře je pro každou stanici uživateli nabízen odkaz na první dostupnou uživatelskou obrazovku. Pokud žádná obrazovka pro danou stanici neexistuje, je nabízena výchozí obrazovka pro typ (účel) této stanice. Nastavení výchozích obrazovek je pak součástí uživatelského rozhraní pracovníků společnosti zadavatele.

2.1.1.4 Výhody

Hlavní výhodou aktuálního systému uživatelských obrazovek je možnost hromadné úpravy všech obrazovek v rámci skupiny skrze úpravu řídící obrazovky. Mezi další pozitiva patří možnost zveřejnění skupiny (šablony) pro všechny uživatele.

2.1.1.5 Nevýhody

Nevýhodou tohoto systému je horší uchopitelnost pro méně zkušené uživatele. Dalším problémem může být změna veřejné skupiny (šablony), kde se mohou vyskytnou případy žádoucí i nežádoucí aktualizace obrazovek. Chybí zde i možnost znovu sloučit oddělenou obrazovku s již existující skupinou.

2.1.1.6 Zhodnocení

Celkově je tento systém vhodný spíše pro pokročilé uživatele, kteří mají ve správě řádově desítky stanic se stejným zaměřením, a tedy významně ocení hromadnou správu shodných uživatelských obrazovek. Dále se nabízí několik možných rozšíření systému, které by přinesli novou funkcionalitu a usnadnění některých úkonů v rámci aplikace. Mezi tyto změny patří možnost definovat více stanic v rámci jedné uživatelské obrazovky nebo možnost zobrazení více obrazovek pro jednu stanici uvnitř stromu oblastí. Následně by bylo možné zobrazení více výchozích obrazovek pro stanice, které jsou využívány k více než jednomu účelu. Spolu s těmito změnami je potřeba provést nový a komplexnější návrh systému obrazovek, který bude umožňovat snadné pochopení aplikace z pohledu běžných uživatelů a zároveň efektivní práci z pohledu uživatelů pokročilých.

2.1.2 Systém modulárních rozšíření

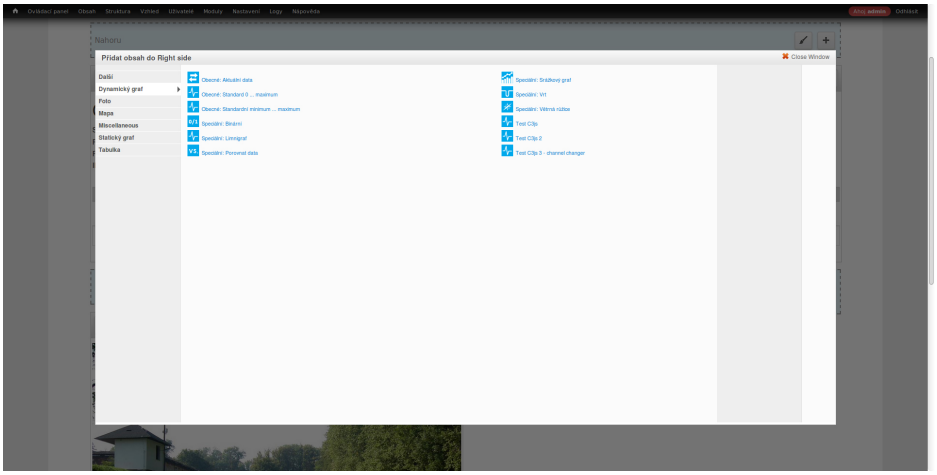
Modulární rozšíření v rámci současné vizualizační komponenty jsou začleněny do systému jako samostatné moduly pro CMS Drupal. Vnitřní struktura modulu je navržena tak, aby umožňovala snadné automatické načítání skrze systém pluginů Ctools, jenž je využíván modulem Panels. Jeden modul může obsahovat i více těchto Ctools pluginů, které reprezentují různé varianty daného rozšíření. Obrázek 2.2 zobrazuje výběr z přehledu pluginů Ctools.

2.1.2.1 Struktura

Uvnitř složky modulárního rozšíření se vždy nachází hlavní soubor samostatného modulu a soubor s meta informacemi o tomto modulu. Dále je zde složka, kde pro každý plugin existuje konfigurační soubor. Součástí každého z těchto konfiguračních souboru je pole, které obsahuje základní informace o pluginu, název funkce pro vykreslení nastavení pluginu a také název funkce pro vykreslení obsahu tohoto pluginu. Dále obsahuje seznam použitých proměnných a jejich výchozí hodnoty. Krom výše zmíněných souborů může plugin obsahovat libovolné knihovny, potřebné pro vizualizaci a další funkce pluginu.

2.1.2.2 Výhody

Výhodou tohoto systému modulárních rozšíření je naprostá nezávislost jednotlivých modulů. Integrace nových rozšíření probíhá plně skrze systémovou



Obrázek 2.2: Výběr typu nového bloku

architekturu CMS Drupal, díky které je modul po instalaci a povolení automaticky začleněn do nabídky dostupných modulů.

2.1.2.3 Nevýhody

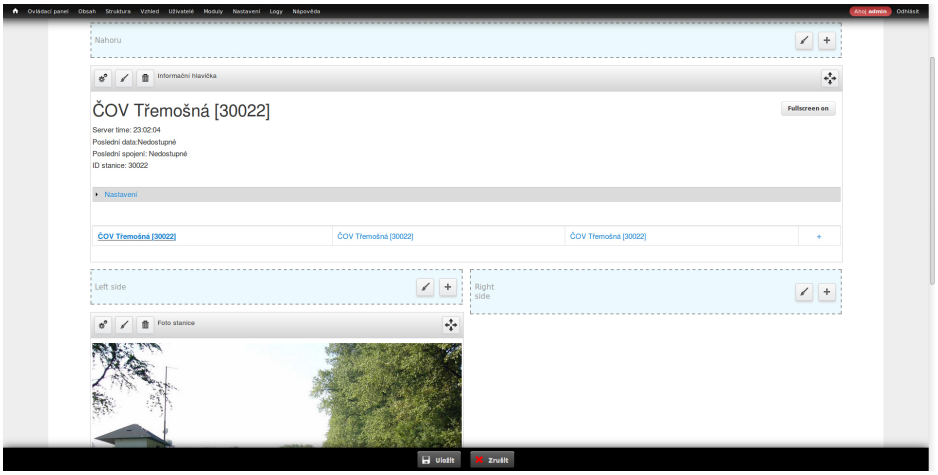
Nevýhodou současného provedení je fakt, že konfigurace rozšíření je jak součástí seznamu základních hodnot, tak součástí funkce pro vykreslení formuláře s těmito hodnotami. Další nevýhodou jsou opakující sekvence kódu napříč většinou modulů, namísto jednotné knihovny, která by zahrnovala funkce společné pro více modulů.

2.1.2.4 Zhodnocení

Systém modulárních rozšíření plně využívá výhod systému modulů CMS Drupal a systému pluginů Ctools. Prostor pro vylepšení systému je především v lepším návrhu struktury konfigurace, který by vedl k možnosti automaticky generovat formuláře. Další výrazné zjednodušení vývoje nových rozšíření by přineslo jednotná knihovna pro práci s těmito rozšířeními.

2.1.3 Uživatelské rozhraní

Uživatelské rozhraní aplikace využívá standardních komponent CMS Drupal 7 jako je field, fieldgroup, filedset nebo tabs. V editačním módu uživatelské obrazovky převažují základní stavební prvky modulu Panels In-Place Editing. Ukázka obrazovky editačním módu se nachází na obrázku 2.3.Celá aplikace používá předpřipravnou responzivní grafická šablonu.



Obrázek 2.3: Úprava pozice bloků na obrazovce

2.1.3.1 Heuristická analýza

Pro zhodnocení uživatelského rozhraní byla použita Nielsenova heuristická analýza. [3]

Viditelnost stavu systému Uživatelské rozhraní systému ve většině případů poskytuje odpovídající zpětnou vazbu, nicméně vyskytují se i případy, kdy systém neinformuje dostatečně o stavu uživatelských požadavků. Toto se děje například při požadavku uživatele na opakované načtení dat pro dynamické grafy.

Propojení systému a reálného světa Systém obrazovek navázaných k jednotlivým stanicím a řazení těchto stanic do stromu oblastí odpovídá topologii v reálném světě. K lepšímu propojení by mohlo přispět hojnější využití vhodných ikon.

Uživatelská kontrola a svoboda Uživatel má v každém kroku práce se systémem možnost zrušení provedených změn a návratu do původního stavu, zároveň je při pokusu o opuštění editace obrazovek žádán o potvrzení odchodu a zahození neuložených změn. Svoboda pohybu mezi jednotlivými obrazovkami je však částečně omezena pokud se na aktuálně prohlížené obrazovce nenachází rozšíření s odkazem na ostatní obrazovky pro aktuální stanici.

Standardizace a konzistence Názvosloví v rámci aplikace je konzistentní, ale v některých pojmech se odlišuje od běžného názvosloví spojeného s doménou uživatelských obrazovek, popřípadě názvosloví původní aplikace.

Prevence chyb Ve většině formulářů je použita validace uživatelsky vložených hodnot, bohužel se tato validace provádí až po odeslání formuláře. Vhodnějším řešením by byla validace na straně klienta, popřípadě zavedením typovaných vstupů, které zcela zamezí vložení neplatných hodnot.

Rozpoznání namísto vzpomínání Většina potřebných informací je uživateli neustále k dispozici. Prostor pro zlepšení je v rámci nastavení jednotlivých rozšíření, kde v posledním kroku není k dispozici informace o typu aktuálně zvoleného rozšíření.

Flexibilní a efektivní použití Systém uživatelských obrazovek je efektivní z hlediska vyžití hromadných úprav. Plné využití potenciálu tohoto systému však vyžaduje dobré porozumění ze strany uživatele. Méně efektivní je pak proces výběru a nastavení nového rozšíření, který se skládá ze tří kroků, které by bylo vhodné sloučit do jednoho.

Estetický a minimalistický design Design aplikace lze považovat za minimalistický, bohužel nikoliv za estetický.

Pomoc uživatelům, pochopit, poznat a vzpamatovat se z chyb Interpretace chyb vůči uživateli je v mnoha případech příliš strohá a není možné ji využít k jasnému pochopení chybové události.

Nápověda a návody V rámci nastavení jednotlivých rozšíření jsou používány popisy jednotlivých částí nastavení, nápověda však zcela chybí.

2.1.3.2 Zpětná vazba uživatelů

V rámci zpětné vazby měli uživatelé připomínky k horší přehlednosti konfigurace modulárních rozšíření. Dále také hůře hodnotili samotný výběr typu modulárního rozšíření. Kladně naopak hodnotili rozhraní pro přidávání a manipulaci s rozšířeními v rámci obrazovky.

2.1.3.3 Zhodnocení

Uživatelské rozhraní aplikace není příliš propracované a obsahuje mnoho nedostatků z hlediska základních principů návrhu uživatelského rozhraní. Přestože je v aplikaci použita responzivní grafická šablona, uživatelská obrazovka, spolu s jejími prvky, není zcela připravena pro zobrazení na displeji mobilních zařízení. Nový návrh uživatelského rozhraní je stěžejní ke zlepšení použitelnosti aplikace.

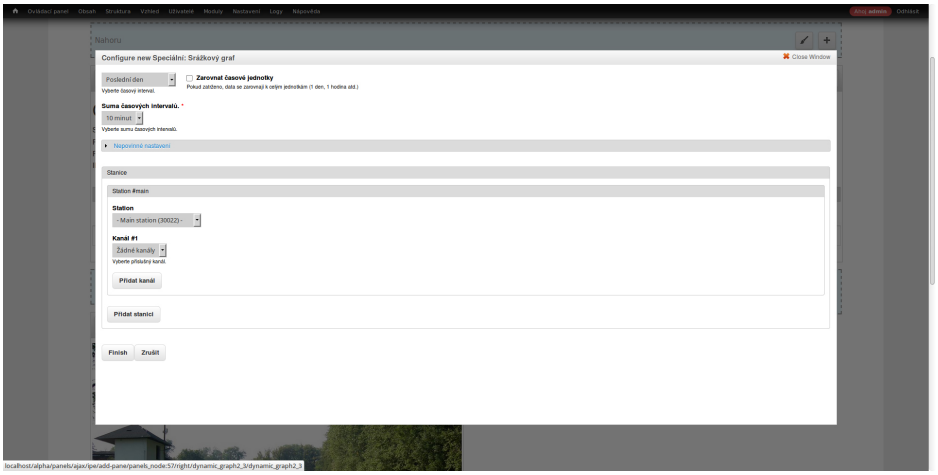
2.1.4 Dostupná modulární rozšíření

2.1.4.1 Tabulky

Mezi rozšíření spadající to kategorie tabulek patří běžná datová tabulka zobrazující data za vybrané období z jednoho nebo více kanálů. Dále je implementována agregační tabulka, která zobrazuje minimum, maximum, průměr a sumu z hodnot za vybrané období. Další speciální tabulky se používají pro zobrazení událostí, aktuálních dešťových srážek a dalším účelům. Všechny z těchto rozšíření využívají knihovny CMS Drupal pro tvorbu HTML tabulek. Kromě vlastního vykreslení zvládá knihovna základní stránkování a řazení podle jednoho sloupce. Tyto akce však vyžadují nové načtení aktuální stránky.

2.1.4.2 Grafy

Rozšíření pro vykreslení dat formou grafů lze rozdělit na grafy statické, které jsou generovány jako obrázky ve formátu PNG a dynamické, které se vykreslují až ve webovém prohlížeči uživatele pomocí JavaScriptu. Poslední verze rozšíření dynamických grafů využívá open source grafovou knihovnu C3js, která je postavena nad knihovnou D3js. Na obrázku 2.4 je zobrazena ukázka úpravy nastavení rozšíření.



Obrázek 2.4: Úprava nastavení bloku

2.1.4.3 Mapy

Pro zobrazení naměřených dat v kontextu s polohou měřících stanic je využita knihovna Leaflet, která poskytuje mapové vrstvy optimalizované pro mobilní zařízení. Leaflet poskytuje základní API pro práci s objekty a událostmi na mapě.

Kromě základního rozšíření pro zobrazení polohy stanic, je implementováno také rozšíření vhodné pro dispečinky monitorující stanice v rámci určitého území. Mimo jiné je v rámci mapových rozšíření dostupná také velmi zjednodušená verze technologické obrazovky, kde mapový podklad nahrazuje schéma sledované části provozu.

2.1.4.4 Ostatní

Mezi ostatní dostupná rozšíření se řadí nástroj pro export naměřených dat do různých formátů a rozšíření pro zobrazení základních informací o stanici nebo odkazů na ostatní dostupné obrazovky.

2.1.4.5 Zhodnocení

Dostupná modulární řešení představují funkční nástroje pro většinu uživatelských požadavků. Nicméně s přihlédnutím k dnešnímu velmi rychlému vývoji nových webových technologií, je na místě provést porovnání ostatních řešení, která splňující klíčové požadavky a zároveň přináší vylepšení použitelnosti.

2.1.5 CMS Drupal 7

Současná aplikace postavená na základech CMS Drupal 7 využívá výhody stabilního jádra tohoto systému, snadné implementace vlastních modulů s možností nezávislé komunikace modulu skrze celý systém formou hooků. Dále je aktivně využívána část systému pro správu uživatelů spolu s nastavením práv a práce se základními entitami CMS Drupal. Vizualizační komponenta pak využívá modul Panels určený pro správu a organizaci obsahu, který zároveň poskytuje rozhraní pro začlenění a správu pluginů Ctools a dalších systémových komponent Drupalu.

Přestože je CMS Drupal 7 velice modulární, je jeho struktura zaměřena především pro práci s obsahem například pro účely blogu, fóra, eshopu nebo online zpravodajství. Pro ostatní účely lze nalézt mnoho vhodných modulů, které poskytují Drupalu další široké spektrum funkcionalit. Nicméně velice specifické aplikace, jako je například vizualizační komponenta vyvíjená v rámci tohoto projektu, naráží na některá systémová, popřípadě technologická omezení, která zpomalují nebo omezují další vývoj.

Z pohledu celé aplikace je CMS Drupal 7 dobrou volbou, nicméně pro složitější komponenty je vhodné zvážit využití dalších řešení. Vzhledem ke komplexnosti systému modulů CMS Drupal, je možné pro jednotlivé komponenty využít i dalších knihoven nebo frameworků, které jinak nezasahují do ostatních součástí systému.

2.1.6 Celkové zhodnocení

Z předešlé analýzy současného řešení vizualizační komponenty vyplývá, že je vhodné provést nový a intuitivnější návrh systému uživatelských obrazovek, který zachová možnost efektivní hromadné správy obrazovek a zároveň začlení možnosti použití více stanic pro jednu obrazovku a více výchozích obrazovek pro jednu stanici. Za účelem úspornějšího vývoje a integrace nových rozšíření je vhodné provést nový návrh konfigurace, ze kterého půjde snadným způsobem generovat potřebné formuláře pro nastavení rozšíření. Uživatelské rozhraní je potřeba zcela přepracovat, aby splňovalo základní principy pro rozhraní moderní webové aplikace. Dále je potřeba provést analýzu technologií, které mohou přispět ke zlepšení použitelnosti modulárních rozšíření.

Před provedením jednotlivých částí redesignu, je potřeba také zvážit použití jiných nástrojů, které by usnadnili vývoj vizualizační komponenty. Podstatné je, aby použitý nástroj umožnil jak instalaci formou modulu v rámci CMS Drupal 7, tak i samostatnou instalaci pro specifické potřeby zákazníků.

2.2 Koncept uživatelských obrazovek a trendy v této oblasti

2.2.1 Definice

Modulární obrazovka neboli dashboard má několik definic. Pro účel této práce jsem zvolil následující: „Vizuální zobrazení nejdůležitějších informací, které jsou potřebné pro dosažení jednoho nebo více cílů na jediné obrazovce monitoru tak, aby bylo umožněno snadné sledování.“[4]

Z definice vyplývá několik podstatných tvrzení, které jsou směrodatné pro návrh rozhraní modulárních obrazovek:

- Modulární obrazovka by měla obsahovat především důležité informace.
- Modulární obrazovka by měla umožnit snadné sledování informací.
- Modulární obrazovka by měla být uzpůsobena ke sledování jednoho i více cílů.

2.2.2 Trendy v oblasti uživatelských obrazovek

Koncept uživatelských obrazovek je velice úspěšný, především z důvodu rychlého zobrazení klíčových informací. Jejich použití je možné najít v každém odvětví, ve kterém je třeba analyzovat určitá data v aktuálním okamžiku. V poslední době se jejich využití objevuje také v osobním životě například pro účely sledování stavu chytré domácnosti nebo také pro sledování zpravodajského dění pomocí kombinace zpráv z RSS kanálů a příspěvků na sociálních sítích. Spolu s každoročním výrazném nárůstem objemu dostupných dat, bude také narůstat význam a počet uživatelských obrazovek.[5]

2.2.3 Základní rozdělení uživatelských obrazovek dle druhu využití

Analýzou většího počtu uživatelských obrazovek lze definovat 3 typy obrazovek dle druhu využití.

Prvním typem jsou obrazovky přehledové, jejichž obsahem bývají zřetelně viditelné informační výstupy, které lze vizuálně vyhodnotit řádově v několika vteřinách. Takto zobrazované informace jsou vždy již ve finální podobě a poskytují podklady pro okamžitá rozhodnutí. Jsou využívány například jako přehled denních bilancí v rámci manažerských přehledů nebo v rámci kontroly aktuálního stavu sprintu v během vývoje pomocí agilní metodiky SCRUM. Typicky obsahují velmi jednoduché grafy nebo zobrazení stavu, popřípadě hodnoty, pomocí textu, ikony či obrázku.

Druhý typ jsou obrazovky analytické, které obsahují podstatně více informací, jež mohou být ve zcela původní nebo již agregované podobě. Jsou určeny pro delší analýzu a hledání případných souvislostí v rámci zobrazených informací. Jsou využívány například jako součást DSS. Typicky obsahují složitější interaktivní grafy a mapy, či tabulky s možností různého řazení a filtrování.

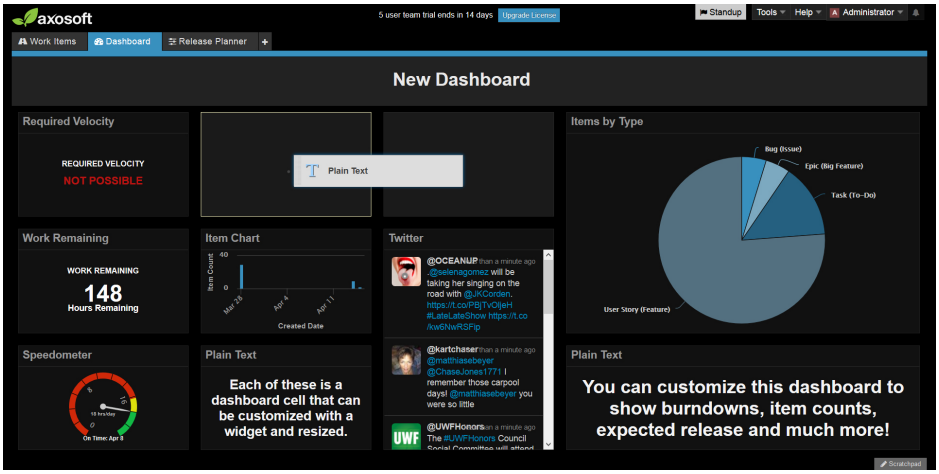
Posledním typem jsou obrazovky hybridní, které kombinují oba výše zmíněné přístupy, například z důvodu různých požadavků v rámci jednoho systému obrazovek, či přímo v rámci jedné obrazovky. Umožňují tak zobrazení klíčových indikátorů ve viditelné části obrazovky a dále zobrazení podrobnějšího přehledu dat ve zbytku obrazovky.

2.2.4 Systém organizace obrazovky podle typu

Během analýzy obrazovek se ukázalo, že s účelem obrazovky se váže také způsob jakým je obrazovka definována.

2.2.4.1 Přehledové obrazovky

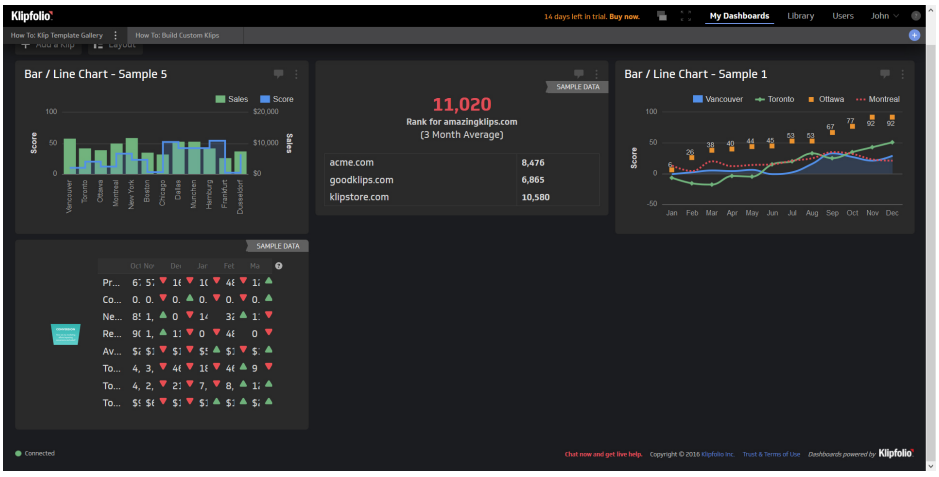
Pro přehledové obrazovky je typické, že všechny části jsou viditelné současně, z čehož plyne, že obrazovka je navrhována na celou výšku i šířku displeje. Z tohoto důvodu je obrazovka rozdělena pomocí čtvercové popřípadě obdélníkové mřížky, která má předem definovaný počet polí na výšku i na šířku. Bloky jsou poté rozmisťovány tak, že zabírají jedno nebo více sousedních polí. Výhodou je již zmiňované plné obsazení displeje, nevýhodou pak je horší zobrazení širokých mřížek na úzkých displejích, kde dojde k přílišnému zúžení obdélníků mřížky. Problém lze řešit fixováním poměru stran jednotlivých bloků na úkor pokrytí celé plochy. Dalším nedostatkem je omezený počet bloků v rámci mřížky. Ukázka přehledové obrazovky se nachází na obrázku 2.5.



Obrázek 2.5: Příklad přehledové obrazovky[6]

2.2.4.2 Analytické obrazovky

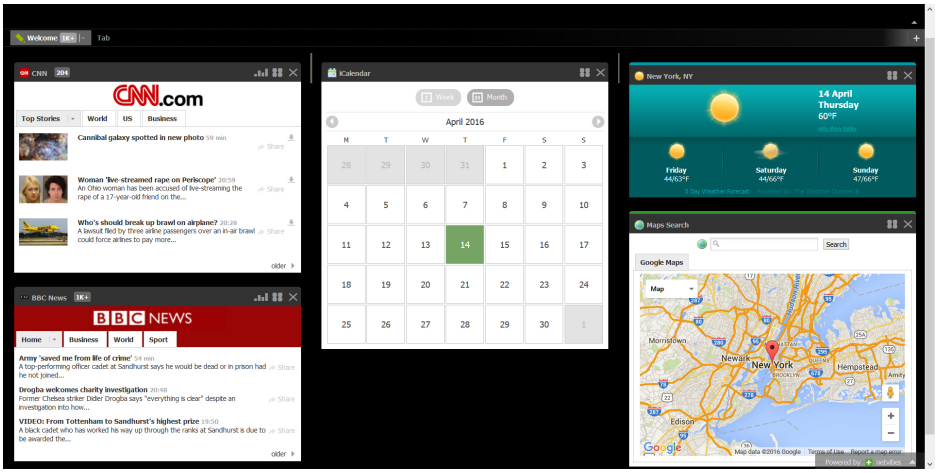
Analytické obrazovky jsou navrhovány, aby vyplnili celou šířku displeje. Výška obrazovky pak může být libovolná. Tyto požadavky nejlépe splňuje systém, ve kterém je základem obrazovky šablona řádek a sloupců. Každý řádek pak obsahuje libovolný, avšak předem daný, počet sloupců. Každý sloupec, jakožto samostatný region, obsahuje libovolný počet bloků u kterých je uloženo také jejich pořadí v daném regionu. Výhodou tohoto řešení je neomezený počet vložených bloků. Nevýhodou pak může být nutnost posunu na stránce pro nalezení požadovaných informací. Ukázka analytické obrazovky se nachází na obrázku 2.6.



Obrázek 2.6: Příklad analytické obrazovky[7]

2.2.4.3 Hybridní obrazovky

Hybridní obrazovky kombinují oba výše uvedené systémy organizace. Ukázka hybridní obrazovky se nachází na obrázku 2.7.



Obrázek 2.7: Příklad hybridní obrazovky[8]

2.2.5 Systém přidání a nastavení bloku

Vhodných systémů pro přidání a nastavení bloků je několik a dají se vzájemně dobře kombinovat. I zde je vidět určitá provázanost s typem obrazovky, nicméně již nejde zcela o přesné kombinace.

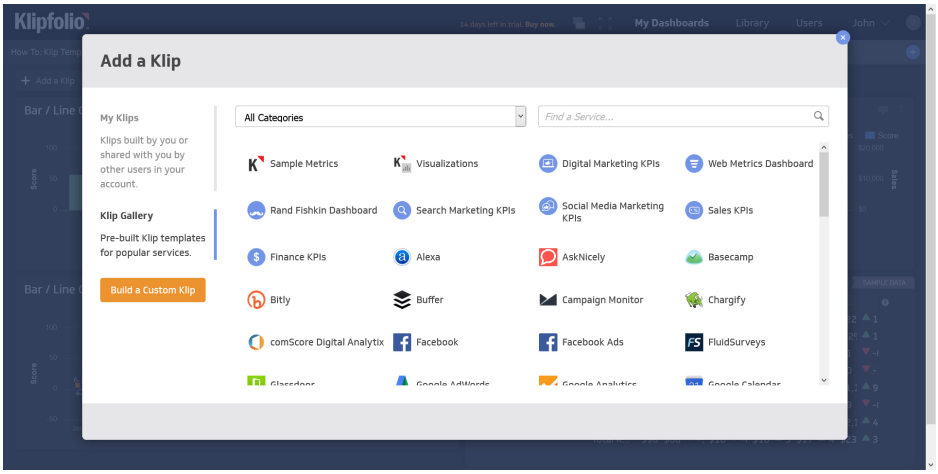
2.2.5.1 Přidání nového bloku

Základní způsob přidaní nového bloku obrazovky probíhá skrze tlačítko „přidat“, jehož stisknutí je následováno zobrazením dialogového okna s nabídkou dostupných bloků. Často jsou také bloky řazeny do kategorií nebo je dostupné filtrování bloků podle textového vstupu. Tento přístup je zobrazen na obrázku 2.8.

Podobnou variantou je pak automatické přidaní abstraktního bloku do části obrazovky a následný výběr konkrétního typu přímo uvnitř přidaného bloku. Další možností je nabídka jednotlivých bloků, která je součástí obrazovky. Z této nabídky lze bloky do obrazovky vložit přetažením bloku.

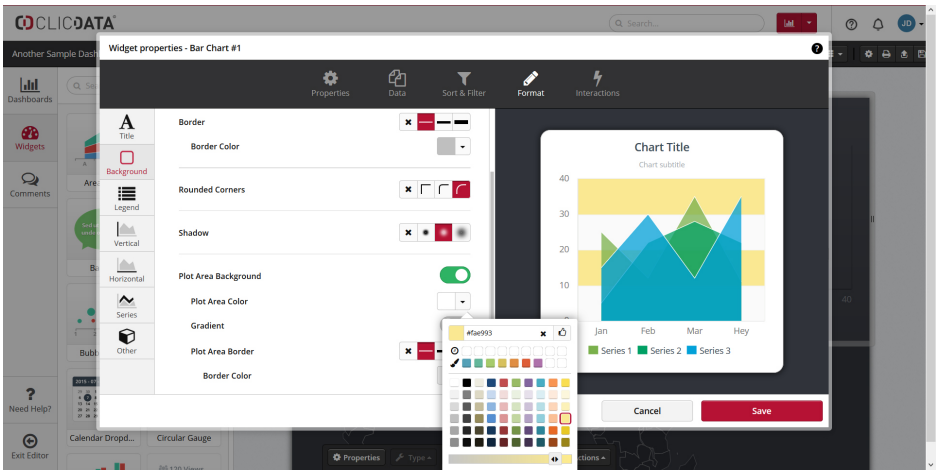
2.2.5.2 Nastavení bloku

Nastavení libovolného bloku lze buďto přímo navázat na proces jeho výběru nebo ho oddělit do samostatného kroku. Oddělení nastavení může přinést větší flexibilitu pro práci zkušenějšího uživatele, také však může vést k opomenutí



Obrázek 2.8: Příklad častého způsobu výběru bloku[7]

nastavení některých bloků. Samotný formulář s nastavením se ve většině případů zobrazuje v rámci dialogového okna, jak je vidět na obrázku 2.9.



Obrázek 2.9: Příklad častého způsobu nastavení bloku[9]

2.2.6 Vybrané vlastnosti obrazovky

Z analýzy konceptu a trendů uživatelských obrazovek byly vybrány tyto klíčové vlastnosti, které by měla vizualizační komponenta implementovat. Konkrétní výběr je také krátce zdůvodněn.

Základní typ obrazovky: Hybridní Zákaznické požadavky na uživatelské obrazovky společnosti FIEDLER AMS se mohou výrazně lišit formou použití.

Pro dispečinky jsou podstatné převážně přehledové obrazovky, avšak z pohledu vedení či správy a řešení problémů naopak dává větší smysl použití analytických obrazovek.

Systém organizace obrazovky: Šablona s řádky a sloupci Vzhledem k různým požadavkům na využití obrazovek se může podstatně lišit počet bloků, které je potřeba v rámci obrazovky zobrazit. Šablona s řádky a sloupci není nijak limitována výškou a umožňuje tak, narozdíl od mřížky, zobrazení libovolného počtu bloků.

Systém přidání nového bloku: Abstraktní blok s vnitřním výběrem Výhodou této varianty je velká flexibilita v uživatelské práci s obrazovkou. Některým uživatelů může více vyhovovat prvotní rozložení struktury bloků s následným výběrem a nastavením, jiným uživatelům může být příjemnější postupné vkládání a nastavení bloků.

Systém nastavení bloku: Volná návaznost, dialogové okno Volná návaznost nastavení nabízí možnost výběru z více různých dialogů pro nastavení shodného bloku. Toho lze využít například pro rozdělení nastavení na základní a pokročilé nebo také pro nastavení s různými výchozími hodnotami.

2.3 Využití existujícího řešení uživatelských obrazovek

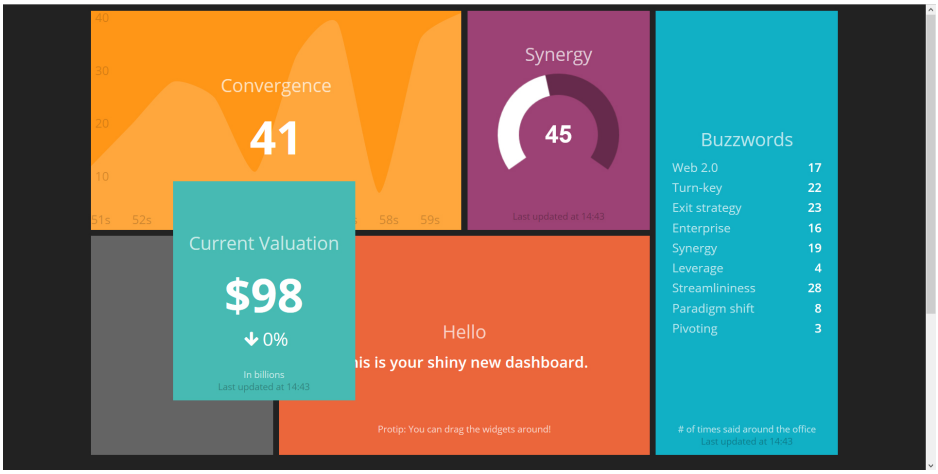
Kvůli nadcházejícím rozhodnutím v oblasti vývoje vizualizační komponenty bude analyzována existence částečného nebo kompletního řešení modulárních obrazovek, začlenitelných ve formě požadované komponenty nebo její části.

2.3.1 Dashing

Nástroj Dashing je zajímavou Open Source knihovnou pro tvorbu uživatelských obrazovek ve webovém prohlížeči. Jeho komunita na GitHubu poskytuje také velké množství již předpřipravených rozšíření. Nevýhodou je velmi strohá dokumentace, která je částečně kompenzována několika tutoriály. Bohužel pro projekt vizualizační komponenty jej nelze využít vzhledem k tomu, že je dostupný pouze pro programovací jazyk Ruby. Ukázka obrazovky vytvořené nástrojem Dashing se nachází na obrázku 2.10.

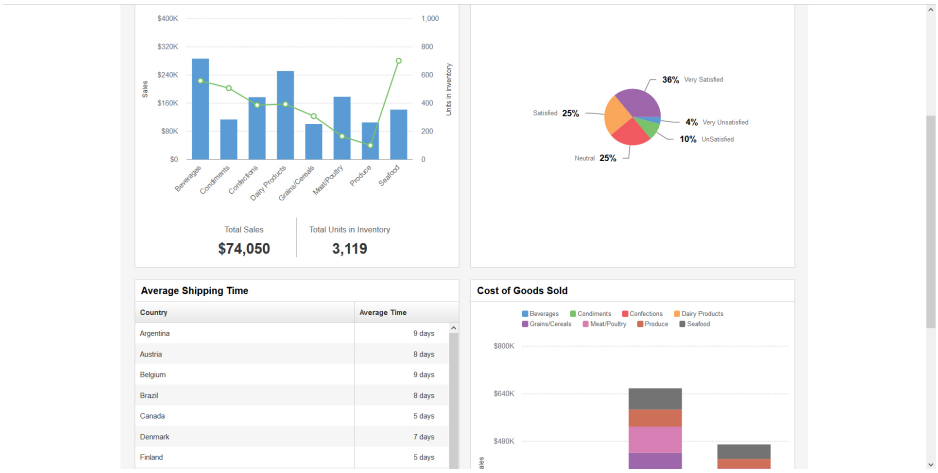
2.3.2 RazorFlow

RazorFlow je knihovnou, která se soustředí na vývoj uživatelských obrazovek. Celkově je složen ze 2 částí, které lze použít současně nebo odděleně. První z nich je psaná v JavaScriptu a zaměřuje na zobrazení i úpravu obrazovek.



Obrázek 2.10: Ukázka nástroje Dashing[10]

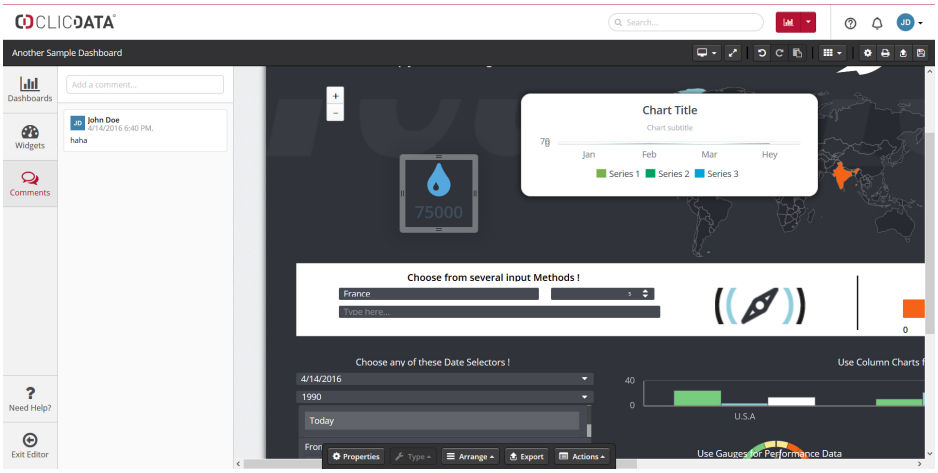
Druhá část kopíruje API JavaScriptové části, tak aby bylo možné ho využít v rámci PHP. Dokumentace je na první pohled dobře zpracovaná, ale při detailnějším pohledu jsou zde některé kapitoly zpracované pouze z části. RazorFlow poskytuje také vlastní fórum, které je však velice neaktuální. Při nahlédnutí do repozitáře vše nasvědčuje tomu, že byl vývoj tohoto projektu ukončen. Dalším problémem využití této knihovny je nejasně definovaná licence, jejíž text na oficiálním webu se liší od textu v samotném repozitáři. Ukázka obrazovky vytvořené nástrojem RazorFlow se nachází na obrázku 2.11.



Obrázek 2.11: Ukázka nástroje RazorFlow[11]

2.3.3 ClicData

Zatímco předchozí řešení byla dostupná formou knihoven, ClicData poskytuje tvorbu obrazovek formou webové služby. Obrazovky aplikace ClicData poskytují možnost načítání dat z mnoha různých datových zdrojů. Načtená data pak lze pomocí přehledné aplikace zobrazit formou rozšíření v rámci obrazovky. Výslednou obrazovku lze následně zobrazit jako vložený prvek webové stránky nebo publikovat formou veřejného odkazu. Slabým místem je aktuálně dostupný počet těchto rozšíření, bez možnosti definice vlastních rozšíření. Dalším podstatnou překážkou je cena produktu, která pro neomezený počet přístupů a uživatelů začíná na částce 1400 amerických dolarů za měsíc. Ukázka obrazovky vytvořené nástrojem ClicData se nachází na obrázku 2.12.



Obrázek 2.12: Ukázka nástroje ClicData[9]

2.3.4 Zhodnocení

Vzhledem k tomu, že ani jedno z těchto řešení není možné rozumně začlenit jako požadovanou vizualizační komponentu nebo její část, je potřeba provést nový návrh a implementaci komponenty.

2.4 Technologie pro reimplementaci vizualizační komponenty

2.4.1 Výhody použití frameworku

Využití frameworku při vývoji komplexnějších aplikací pomáhá šetřit čas strávený vývojem generických částí jako je například systém autentizace uživatelů. Obecně také napomáhá k abstrakci nižších vrstev programovacího jazyka. Celkově se tak lze více soustředit přímo na business logiku aplikace. Další výhodou

je fakt, že většina frameworků navádí k dodržování standardů při psaní kódu a osvědčených postupů pro danou platformu.

2.4.2 Je Drupal framework?

Přestože se CMS Drupal 7 zaměřuje především na správu a publikaci uživatelského obsahu, sdílí také jisté znaky frameworku. Z toho důvodu je někdy označován jako content management framework neboli CMF. Na rozdíl od většiny tradičních frameworků však Drupal 7 neposkytuje například typickou MVC architekturu a je téměř celý vystavěn procedurálně.[12]

V současné době je však již dostupný CMS Drupal verze 8, který se výrazně přibližuje konceptu plnohodnotného frameworku. Přichází nově například s objektovým přístupem, jmennými prostory, MVC architekturou nebo novým formátem konfiguračních souborů. Většinu těchto principů přijímá z již existujícího frameworku Symfony, jehož části jsou zakomponovány do jádra nové verze systému.[13]

2.4.3 Zhodnocení

Pro implementaci vizualizační komponenty byl po důkladném zvážení zvolen framework Symfony, který pokrývá všechny významné oblasti vývoje moderní webové aplikace. Výrazným faktorem výběru byla také skutečnost, že nová verze CMS Drupal ve velké míře adaptuje principy právě tohoto frameworku a po provedení migrace zbylých modulů aplikace na Drupal verze 8 bude docíleno konzistence celého projektu.

2.5 Systém tvorby uživatelských obrazovek

V této části je na základě předchozí analýzy proveden nový návrh systému uživatelských obrazovek. Tento návrh respektuje výběr zvolených technologií a klíčové aspekty uživatelských obrazovek z kapitoly 2.2.6.

Systémem tvorby uživatelských obrazovek je chápán jako systém přístupu k uživatelským obrazovkám, rozhraní pro jejich správu a úpravu uspořádání bloků těchto obrazovek.

V rámci systému obrazovek je potřeba zohlednit organizační strukturu obrazovek, která vychází ze stromové struktury stanic a oblastí zmíněné v kapitole 2.1.1.3. Nově definovaná stromová struktura oblastí může rámci jedné konkrétní oblasti obsahovat další podoblasti nebo obrazovky. Vnitřní oblasti stromu reprezentují územní celky nebo jejich části. Koncové oblasti pak reprezentují jednotlivé stanice. Oblasti také obsahují seznam stanic, které se k dané oblasti vážou.

Kromě kategorizace do stromové struktury je oblastem přidána také možnost přiřazení dynamických značek neboli „tagů“. Pro každou takovou značku je pak možné definovat výchozí obrazovku, která bude v rámci oblastí s touto

značkou zobrazena. Tato část systému vylepšuje původní systém výchozích obrazovek.

Jednotlivé uživatelské obrazovky obsahují, podobně jako oblasti, seznam dostupných stanic a dále jednotlivé bloky neboli „widgety“, jejichž rozložení je definováno šablonami řádek a sloupců. Uživatel si může pomocí kombinace jednotlivých bloků poskládat vlastní obrazovku, kterou může následně kopírovat na ostatní obrazovky.

V rámci hromadné správy obrazovek je navržen nový dvoufázový systém kopírování obrazovek. V první fázi má uživatel možnost propojení více obrazovek do jedné skupiny. Takto propojené obrazovky mohou mít libovolné rozvržení i nastavení. Ve druhé fázi pak jsou uživateli během akce kopírování nabídnuty obrazovky ze skupiny propojených obrazovek. Výhodou rozdělení kopírování do 2 kroků je zjednodušení při opakovaném výběru obrazovek.

Za účelem snadnějšího kopírování obrazovek nejsou v rámci bloků uchovávány reálná id stanic, ale jsou použity zástupné identifikátory platné pouze v rámci konkrétní obrazovky. Reálné id stanice je získáváno z nastavení obrazovky až ve chvíli, kdy widget žádá o načtení dat.

Kromě kopírování jedné obrazovky je možné kopírovat i celou koncovou oblast, což umožní ještě rychlejší a efektivnější tvorbu nových oblastí a obrazovek. Navíc při takovémto kopírování koncových oblastí dojde také k automatickému propojení nově vzniklých obrazovek do původních skupin propojených obrazovek.

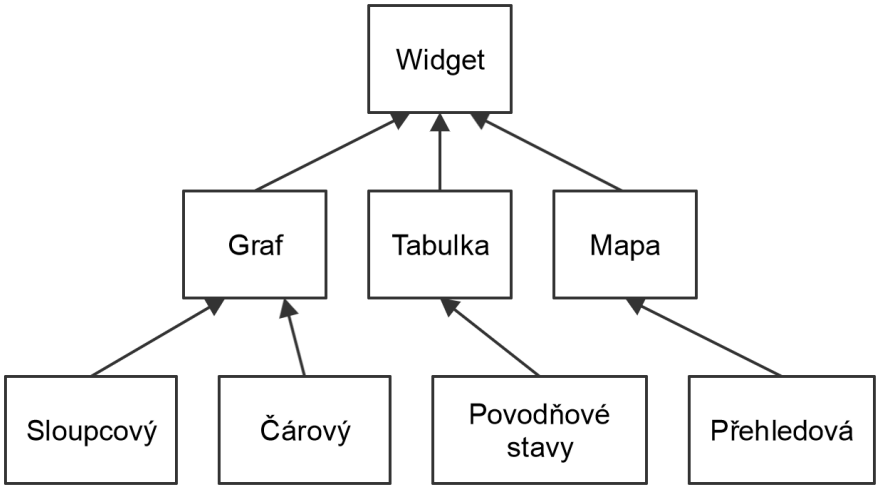
2.6 Systém modulárních rozšíření

Systém modulárních rozšíření je navržen tak, aby umožnil rychlý vývoj nových rozšíření, během kterého je možné využít co největší část dříve implementovaných funkcí, ale zároveň dodržuje nezávislost jednotlivých rozšíření.

2.6.1 Využití dědičnosti

Kvůli docílení nezávislosti a zároveň vysoké znovu použitelnosti kódu je systém modulárních rozšíření rozdělen do třech generací tříd typu controller. Rodičem je třída abstraktního rozšíření nazvaná WidgetController, od které pak dědí třídy na úrovni typu (kategorie) rozšíření, tedy například ChartController. Třetí generací jsou pak již přímo třídy podtypu rozšíření, například ColumnChartController. Celá hierarchie je znázorněna na diagramu 2.13.

Takto navržená hierarchie tříd umožňuje v rámci rozšíření využívat obecné metody shodné pro všechna rozšíření a dále také metody specifické pro konkrétní typ (kategorii) rozšíření. Třídy podtypu rozšíření pak využívají vlastních metod, takže je splněn i druhý požadavek na nezávislost.



Obrázek 2.13: Diagram znázorňující hierarchii rozšíření

2.6.2 Struktura rozšíření

Kromě controlleru obsahují jednotlivá rozšíření také složku views, ve které se nacházejí 2 varianty twig šablony pro hlavičku, tělo a patičku zobrazovaného bloku. První varianta je určena pro stav úpravy bloku, druhá varianta pak pokrývá samotné zobrazení bloku v rámci vizualizace dat. Dále obsahují složku configs, ve které se nalézá vždy minimálně 1 konfigurace rozšíření. Mimo to mohou ještě obsahovat složku s překlady a libovolné knihovny nebo další soubory potřebné pro správné zobrazení bloku.

2.6.3 Formát konfigurace

Jako formát pro uložení konfigurace je použit formát YAML, který je standardním formátem pro uložení konfigurací frameworku Symfony i nového CMS Drupal 8. Výhodou formátu YAML je jeho snadná čitelnost i pro člověka.[14]

2.6.4 Struktura konfigurace

Struktura konfigurace rozšíření zahrnuje typ i nastavení. Nastavení pak dále rozděleno na 2 části. První z nich jsou informace o jednotlivých položkách nastavení. Druhá část pak zahrnuje omezení, která slouží k validaci či skrytí položky nastavení.

```
widget:
  type: graph
  subtype: limnigraph
  settings:
```

```
    defaultLabel: Limnigraph
    defaultDescription: New limnigraph.
    customSettings:
      refreshTime:
        info:
          type: intervalpicker
          units:
s~label: Refresh time
          description: Set your refresh time.
          help: This parameter defines ...
        restrictions:
          required: true
          min: 0
          max:
          regexp:
          validate: function
          visible: false
          visibleIf: spa1
          invisibleIf:
          disabled: false
          disabledIf:
          enabledIf:
          options: []
          defaultValue: 60
      spa1:
        ...: ...
  allowedDataSources:
  - type: channels
    restrictions:
      minChannels: 0
      maxChannels: 16
      channelTypes:
      - water_level
      - gauge
```

2.7 Systém rozložení obrazovek

Na základě analýzy z bakalářské práce, která se zabývala rozložením obrazovek v původním systému je možné stanovit minimální požadavky na dostupná rozložení řádek a sloupců. V rámci bakalářské práce bylo prozkoumáno 20 obrazovek v původním systému, na kterých bylo určeno 5 typů řádkových rozložení a 2 typy sloupcových rozložení. Výsledky jsou prezentovány v tabulce

2.1. Sloupec „Rozložení řádek“ reprezentuje počet bloků na řádce sloupec „Rozložení sloupců“ nabízí minimalizovanou verzi rozložení.[15]

Tabulka 2.1: Tabulka výskytu rozvržení

Rozložení řádek	**Výskyt**	**Rozložení sloupců**	**Výskyt**
1 ... 1 ... 1 ... 1 ...	70%	1	70%
2 ... 1 ... 1 ... 1 ...	15%	2 .. 1	30%
2 ... 2 ... 1 ... 1 ...	5%		
2 ... 2 ... 2 ... 1 ...	5%		
2 ... 2 ... 2 ... 2 ...	5%		

Z tabulky 2.1 vyplývá potřeba rozložení celkově o dvou řádcích, kde v prvním bude vložen 1 sloupec o šířce 100% a ve druhém budou vloženy 2 sloupce, každý o šířce 50%. Tento požadavek je velmi triviální a v kontextu s dnešním širokým spektrem používaných displejů v mnoha případech také nedostatečný. Z toho důvodu je potřeba definovat flexibilní způsob reprezentace layoutů.

Jako způsob reprezentace, který pokrývá veškeré možné layouty, je zvoleno 2D pole, kde první rozměr reprezentuje řádky layoutu a druhý rozměr reprezentuje poměry jednotlivých sloupců v daném řádku.

S ohledem na použití mobilních zařízení, je třeba počítat s automatickým přepnutím do mobilního layoutu. Ten obsahuje pouze jeden řádek s jedním sloupcem o šířce 100%.

2.8 Vizualizace dat pomocí grafů

2.8.1 Statické grafy

Po konzultaci se zadavatelem bylo rozhodnuto, že statické grafy formou PNG obrázků nebudou součástí nové vizualizační komponenty. Implementace těchto grafů jako rozšíření uživatelské obrazovky nemá význam, vzhledem k tomu, že podpora a povolení JavaScriptu v prohlížeči bude nutná pro správný chod celé aplikace.

2.8.2 Dynamické grafy

V současné chvíli existuje mnoho JavaScriptových knihoven, které slouží k vykreslování dynamických grafů. Většina těchto knihoven je pro komerční použití zpoplatněna. Z důvodu možné instalace vizualizační komponenty na servery nebo počítače klientů, je navíc potřeba počítat se zakoupením OEM licence, která umožňuje její použití jako součásti komerčního produktu. V rámci mé bakalářské práce jsem porovnával 3 placené knihovny, u kterých jsem zjišťoval cenu a podmínky OEM licence. Při následném jednání o nákupu vybrané grafové knihovny, bylo ze strany prodejce předložena nabídka, která neodpovídala původní komunikaci a zahrnovala pouze variantu OEM licence s omezeným

počtem instalací. Z tohoto důvodu jsou v rámci této práce porovnány 3 nejvhodnější JavaScriptové knihovny zaměřené na tvorbu dynamických grafů, avšak s otevřenou licencí. Ohled bude brán na požadované typy grafů, funkcionalitu, spolehlivost, přehlednost dokumentace a velikost komunity.[16]

Požadované typy grafů

- Lineární graf s nastavením minima a maxima
- Lineární graf s možností invertovat osu y
- Lineární graf kombinovaný se sloupcovým grafem
- Lineární krokový graf
- Limnigraf
- Větrná růžice

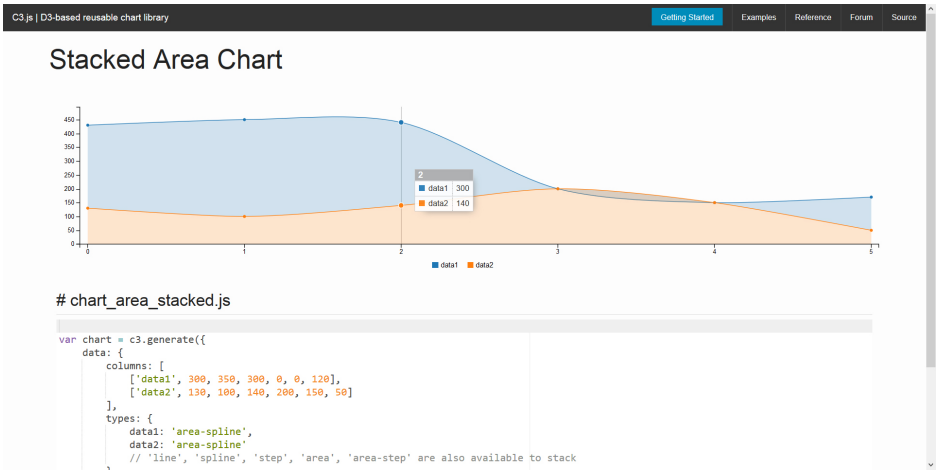
Požadovaná funkcionalita

- Interaktivní skrývání a odkrývaní datových kanálů
- Možnost přiblížení
- Posun v grafu
- Podpora časové řady s automatickým výpočtem popisů časové osy
- Podpora dat ve formátu JSON
- Automatická aktualizace dat
- Responzivita

2.8.2.1 C3js

Základ grafové knihovny C3js je JavaScriptová knihovna D3js, která se zaměřuje na vizualizaci dat a následnou manipulaci s touto vizualizací. K samotnému vykreslení ve webovém prohlížeči jsou použity technologie HTML, SVG a CSS. Aktuálně je dostupná stabilní verze číslo 0.4.10 a pre-release verze 0.4.11-rc4 v rámci MIT licence. Ukázka grafu vytvořeného poocí knihovny C3js se nachází na obrázku 2.14.

Dokumentace C3js je složena pouze z tutoriálů a příkladů, které však pokrývají veškeré vlastnosti knihovny. Komunita kolem knihovny je řádově v počtu desítek členů, kteří komunikují především skrze komunikační kanál Gitter. Vývojem se zabývá původní tvůrce knihovny, s občasnými příspěvky dalších vývojářů.



Obrázek 2.14: Ukázka knihovny C3js[17]

Požadované typy grafů

- Lineární graf s nastavením minima a maxima: **ano**
- Lineární graf s možností invertovat osu y: **ano**
- Lineární graf kombinovaný se sloupcovým grafem: **ano**
- Lineární krokový graf: **ano**
- Limnigraf: **ano**
- Větrná růžice: **ne**

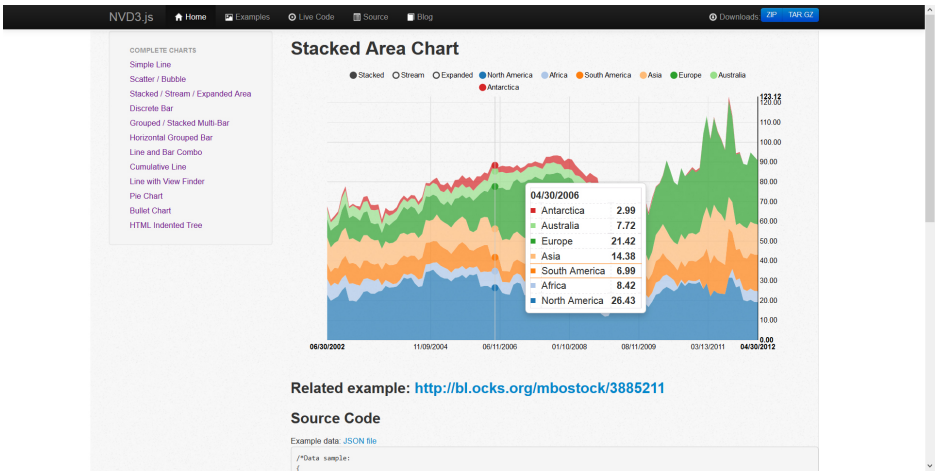
Požadovaná funkcionalita

- Interaktivní skrývání a odkrývaní datových kanálů: **ano**
- Možnost přiblížení: **ano**
- Posun v grafu: **ano**
- Podpora časové řady s automatickým výpočtem popisů časové osy: **ne**
- Podpora dat ve formátu JSON: **ano**
- Automatická aktualizace dat: **ano**
- Responzivita: **ano**

Kromě požadovaných druhů grafů navíc obsahuje graf míry a koláčový graf. Do nestandardních funkcí patří sada transformačních animací mezi jednotlivými typy grafů a načítání dat přímo z url. Naopak zde chybí graf větrné růžice a automatický výpočet popisů časové osy. Nevýhodou je také velice pomalé vykreslování pro větší objemy dat.

2.8.2.2 NVD3

Grafová knihovna NVD3 je také postavena na základech knihovny D3js. Navíc tato knihovna dodržuje konvence pro zapouzdření opakovaně použitelných grafů, které vydal sám autor knihovny D3js Mike Bostock. Aktuálně je dostupná stabilní verze číslo 1.8.2 v rámci licence Apache verze 2.0. Ukázka grafu vytvořeného poocí knihovny NVD3 se nachází na obrázku 2.15.



Obrázek 2.15: Ukázka knihovny NVD3[18]

Dokumentace knihovny NVD3 je velice přehledná a krom výčtu vlastností k jednotlivým typům grafů také obsahuje praktické příklady grafů s určitými vlastnosti. Komunita kolem knihovny je řádově o velikosti několika stovek členů, kteří komunikují především skrze systém repozitáře GitHub. Hlavním vývojem se zabývá několik vývojářů společnosti Novus, která sama využívá této knihovny pro potřeby svých řešení z oblasti manažerských obrazovek.

Požadované typy grafů

- Lineární graf s nastavením minima a maxima: **ano**
- Lineární graf s možností invertovat osu y: **ano** (nepřímo)
- Lineární graf kombinovaný se sloupcovým grafem: **ano**
- Lineární krokový graf: **ne**

- Limnigraf: **ano**
- Větrná růžice: **ne**

Požadovaná funkcionalita

- Interaktivní skrývání a odkrývaní datových kanálů: **ano**
- Možnost přiblížení: **ano**
- Posun v grafu: **ano**
- Podpora časové řady s automatickým výpočtem popisů časové osy: **ano**
- Podpora dat ve formátu JSON: **ano**
- Automatická aktualizace dat: **ano**
- Responzivita: **ano**

Knihovna celkově obsahuje mnoho typů grafů, které se výrazně liší svými vlastnostmi. Mezi dostupnými typy však není ani krokový graf, ani graf větrné růžice. Výhodou je naopak podpora automatických popisů časové osy.

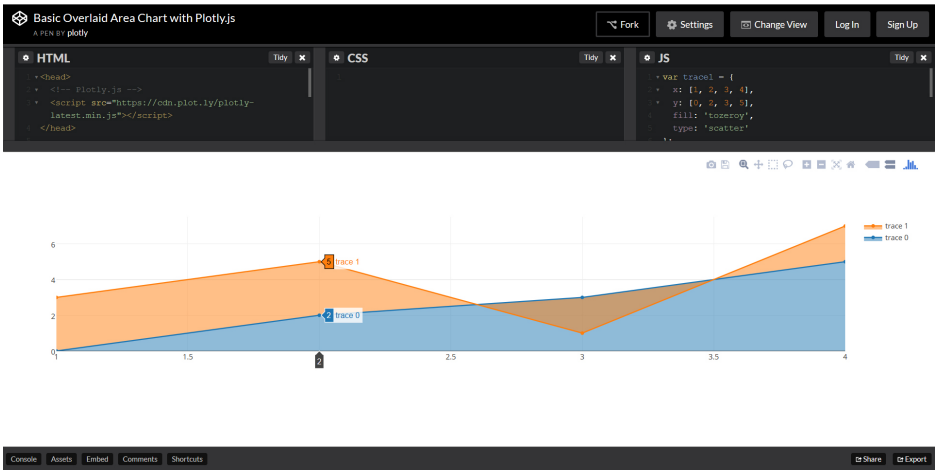
2.8.2.3 Plotly

I grafová knihovna Plotly leží na základech D3js. Kromě toho také využívá knihovnu Stack.gl pro práci s 3D objekty a technologií WebGL obecně. Kromě webové rozhraní pro JavaScript nabízí Plotly také rozhraní pro Python, Matlab a statistický software R. Plotly.js je aktuálně dostupná ve verzi 1.9 v rámci MIT licence. Ukázka grafu vytvořeného poocí knihovny Plotly se nachází na obrázku 2.16.

Dokumentace Plotly.js je zcela vyčerpávající, neboť obsahuje od úvodních tutoriálů, přes kompletní API s interaktivními příklady, až po přehledně zpracovaný přehled základních vlastností v tisknutelné podobě. Komunita kolem Plotly se vzhledem k nedávnému uvolnění licence teprve formuje, ale již nyní se na komunitním fóru přímo od tvůrců Plotly nachází stovky příspěvků.

Požadované typy grafů

- Lineární graf s nastavením minima a maxima: **ano**
- Lineární graf s možností invertovat osu y: **ano**
- Lineární graf kombinovaný se sloupcovým grafem: **ano**
- Lineární krokový graf: **ano**
- Limnigraf: **ano**
- Větrná růžice: **ano**



Obrázek 2.16: Ukázka knihovny Plotly[19]

Požadovaná funkcionalita

- Interaktivní skrývání a odkrývaní datových kanálů: **ano**
- Možnost přiblížení: **ano**
- Posun v grafu: **ano**
- Podpora časové řady s automatickým výpočtem popisů časové osy: **ano**
- Podpora dat ve formátu JSON: **ano**
- Automatická aktualizace dat: **ano**
- Responzivita: **ano**

Plotly.js poskytuje všechny typy požadovaných grafů a zároveň splňuje také všechny požadavky na funkcionalitu. Kromě toho poskytuje další desítky typů grafů, několik typů vektorových map a navíc také podporu 3D grafů. Velkou výhodou je taky rozhraní pro práci s událostmi a uživatelskými interakcemi v grafech. Užitečná může být také možnost zadávání textů ve formátu LaTeX.

2.8.2.4 Zhodnocení

Z výběru těchto 3 knihoven je jasným vítězem knihovna Plotly, která splňuje požadavky na všechny potřebné typy grafů i požadovanou funkcionalitu, poskytuje nejpřehlednější dokumentaci s mnoha tutoriály a zaručuje velice stabilní produkt od společnosti, která na základech této knihovny staví své vlastní aplikace. Kromě těchto faktorů je také velkou výhodou rostoucí komunita.

2.9 Analýza případů užití

2.9.1 Persony

K získání seznamu účastníků byla provedena analýza budoucích uživatelů aplikace. Pro tento účel byli zvoleni uživatelé z odvětví vodohospodářství.

2.9.1.1 Obyvatel obce

Tento člověk v aplikaci hledá například informace o aktuálním výšce hladiny nebo o objemu průtoku místních vodních toků. Není členem žádné vnitřní struktury. Z pohledu aplikace jde o anonymního uživatele. Má přístup pouze k veřejným obrazovkám.

2.9.1.2 Pracovník zodpovědný za provozu v čistírně odpadních vod

Tento člověk v aplikaci hledá podrobné přehledy aktuálního stavu chodu čerpadel nebo výšku hladiny, popřípadě objem kyslíku v nádrži. Je členem vnitřní struktury organizace čistírny odpadních vod. Z pohledu aplikace jde o přihlášeného uživatele, který má přístup k zobrazení určité skupiny obrazovek.

2.9.1.3 Vedoucí pracovník čistírny odpadních vod

Tento člověk v aplikaci hledá podrobné zobrazení aktuálního stavu provozu, stejně jako dlouhodobé přehledy. Dále vyžaduje možnost nastavení těchto přehledů pro potřeby sebe i svých podřízených. Je členem vnitřní struktury organizace pro čistírny odpadních vod. Z pohledu aplikace jde o přihlášeného uživatele, který má přístup k zobrazení, tvorbě a úpravě vlastních obrazovek.

2.9.1.4 Pracovník zajišťující podporu na straně poskytovatele

Tento člověk v aplikaci hledá možnost kompletního nastavení obrazovek a výchozích obrazovek. Je členem vnitřní struktury organizace poskytovatele aplikace. Z pohledu aplikace jde o přihlášeného uživatele, který má přístup ke všem obrazovkám v systému, které může upravovat pro potřeby zajištění technické podpory vůči zákazníkovi.

2.9.1.5 Demo uživatel

Tento člověk v aplikaci hledá možnost prohlédnout si funkcionalitu aplikace. Není členem žádné vnitřní struktury. Z pohledu aplikace jde o přihlášeného uživatele, který má přístup ke skupině stanic a zobrazení jejich obrazovek.

2.9.2 Seznam účastníků

Z pohledu seznamu účastníků lze sloučit personu demo uživatele a pracovníka zodpovědného za provoz v čistírně odpadních vod do jedné role.

- Neregistrovaný uživatel
- Registrovaný uživatel
- Registrovaný uživatel s právy editace
- Administrátor
- Hlavní administrátor

2.9.3 Scénáře případů užití

U1 Přihlášení

U1.1 Přihlášení

1. Případ užití začíná vstupem uživatele na domácí stránku aplikace s úmyslem se přihlásit.
2. Uživatel zadá své uživatelské jméno do textového pole s nápisem „Username“.
3. Uživatel zadá své heslo do textového pole s nápisem „Password“.
4. Uživatel stiskne tlačítko s ikonou přihlášení.
5. Aplikace zobrazí uživatelské jméno přihlášené osoby.

U2 Zobrazení obrazovky

U2.1 Zobrazení veřejné obrazovky

1. Případ užití začíná kliknutím na odkaz veřejné obrazovky s úmyslem prohlédnout si dostupné informace z určité stanice nebo skupiny stanic.
2. Aplikace zobrazí vybranou obrazovku.

U2.2 Zobrazení obrazovky

1. Případ užití začíná úmyslem prohlédnout si dostupné informace z určité stanice nebo skupiny stanic.
2. Include (Přihlášení)
3. Uživatel vyhledá požadovanou obrazovku ve stromové struktuře oblastí a stiskne její odkaz.
4. Aplikace zobrazí vybranou obrazovku.

U3 Vytvoření obrazovky

U3.1 Vytvoření nové obrazovky

1. Případ užití začíná úmyslem vytvořit novou obrazovku.
2. Include (Přihlášení)
3. Uživatel vyhledá ve stromové struktuře přehled oblasti, do které chce novou obrazovku vložit a stiskne odkaz přehledu.
4. Aplikace zobrazí stránku s přehledem vybrané oblasti.
5. Uživatel stiskne tlačítko „+ Dashboard“.
6. Aplikace zobrazí formulář pro vytvoření obrazovky.
7. Uživatel vyplní potřebná pole formuláře a stiskne tlačítko „Create dashboard“.
8. Aplikace zobrazí nově vytvořenou obrazovku.

U3.2 Propojení obrazovky

1. Případ užití začíná úmyslem propojit vybranou obrazovku.
2. Include (Zobrazení obrazovky)
3. Uživatel stiskne tlačítko „Connect“.
4. Aplikace zobrazí nabídku obrazovek, se kterými je možné obrazovku propojit.
5. Uživatel stiskne tlačítko „Save dashboard connections“.
6. Aplikace opět zobrazí vybranou obrazovku s textem o úspěšném uložení propojených obrazovek.

U3.3 Kopírování obrazovky

1. Případ užití začíná úmyslem kopírovat vybranou obrazovku.
2. Include (Zobrazení obrazovky)
3. Uživatel stiskne tlačítko „Copy“.
4. Aplikace zobrazí nabídku možností kopírování a obrazovky, na které je možné vybrané změny kopírovat.
5. Uživatel stiskne tlačítko „Copy template to selected dashboards“.
6. Aplikace opět zobrazí vybranou obrazovku s textem o úspěšném kopírování.

U4 Zobrazení obrazovky

U4.1 Změna nastavení obrazovky

1. Případ užití začíná úmyslem změnit nastavení vybrané obrazovky.
2. Include (Zobrazení obrazovky)
3. Uživatel stiskne tlačítko „Settings“.
4. Aplikace zobrazí formulář pro nastavení obrazovky.

5. Uživatel stiskne tlačítko „Save dashboard settings“.
6. Aplikace opět zobrazí vybranou obrazovku s textem o úspěšném uložení nastavení.

U4.2 Změna obsahu obrazovky

1. Případ užití začíná úmyslem změnit obsah vybrané obrazovky.
2. Include (Zobrazení obrazovky)
3. Uživatel stiskne tlačítko „Edit“ u bloku.
4. Aplikace zobrazí editační mód obrazovky.
5. Uživatel provede změny jednotlivých bloků a stiskne tlačítko „Uložit“.
6. Aplikace opět zobrazí vybranou obrazovku s textem o úspěšném uložení provedených změn.

U4.3 Smazání obrazovky

1. Případ užití začíná úmyslem smazat vybranou obrazovku.
2. Include (Přihlášení)
3. Uživatel vyhledá ve stromové struktuře přehled oblasti, do které patří obrazovka, kterou chce smazat a stiskne odkaz přehledu.
4. Aplikace zobrazí stránku s přehledem vybrané oblasti.
5. Uživatel stiskne tlačítko ve sloupci „Delete“ v řádce vybrané obrazovky.
6. Aplikace zobrazí dotaz na potvrzení této akce.
7. Uživatel potvrdí své rozhodnutí stisknutím tlačítka „Ok“.
8. Aplikace znovu zobrazí přehled oblasti spolu s textem o úspěšném smazání obrazovky.

U5 Vytvoření oblasti

U5.1 Vytvoření oblasti

1. Případ užití začíná úmyslem vytvořit novou oblast.
2. Include (Přihlášení)
3. Uživatel vyhledá ve stromové struktuře přehled oblasti, do které chce přidat novou oblast a stiskne odkaz přehledu.
4. Aplikace zobrazí stránku s přehledem vybrané oblasti.
5. Uživatel stiskne tlačítko „+ Group“.
6. Aplikace zobrazí formulář pro vytvoření nové oblasti.
7. Uživatel vyplní formulář a stiskne tlačítko „Create group“.
8. Aplikace znovu zobrazí přehled vybrané oblasti s textem o úspěšném vytvoření.

U6 Úprava oblasti

U6.1 Změna nastavení oblasti

1. Případ užití začíná úmyslem změnit nastavení oblasti.
2. Include (Přihlášení)
3. Uživatel vyhledá ve stromové struktuře přehled oblasti, u které chce změnit nastavení a stiskne odkaz přehledu.
4. Aplikace zobrazí stránku s přehledem vybrané oblasti.
5. Uživatel stiskne tlačítko „Settings“.
6. Aplikace zobrazí formulář pro nastavení oblasti.
7. Uživatel provede úpravy a stiskne tlačítko „Save group settings“.
8. Aplikace znovu zobrazí přehled vybrané oblasti s textem o úspěšném uložení nastavení.

U6.2 Přiřazení výchozí obrazovky k tagu oblasti

1. Případ užití začíná úmyslem nastavit novou výchozí obrazovku pro oblasti s vybraným tagem.
2. Include (Přihlášení)
3. Uživatel vyhledá ve stromové struktuře přehled výchozích oblasti a stiskne odkaz přehledu.
4. Aplikace zobrazí stránku s přehledem vybrané oblasti.
5. Uživatel stiskne tlačítko ve sloupci „Settings“ vybrané výchozí obrazovky.
6. Aplikace zobrazí formulář pro nastavení obrazovky.
7. Uživatel vybere tag, ke kterému bude obrazovka přiřazena jako výchozí a stiskne tlačítko „Save dashboard settings“.
8. Aplikace znovu zobrazí přehled oblasti výchozích obrazovek.

2.10 Analýza požadavků

V této části je provedena analýza funkčních, nefunkčních a systémových požadavků.[20]

2.10.1 Funkční požadavky

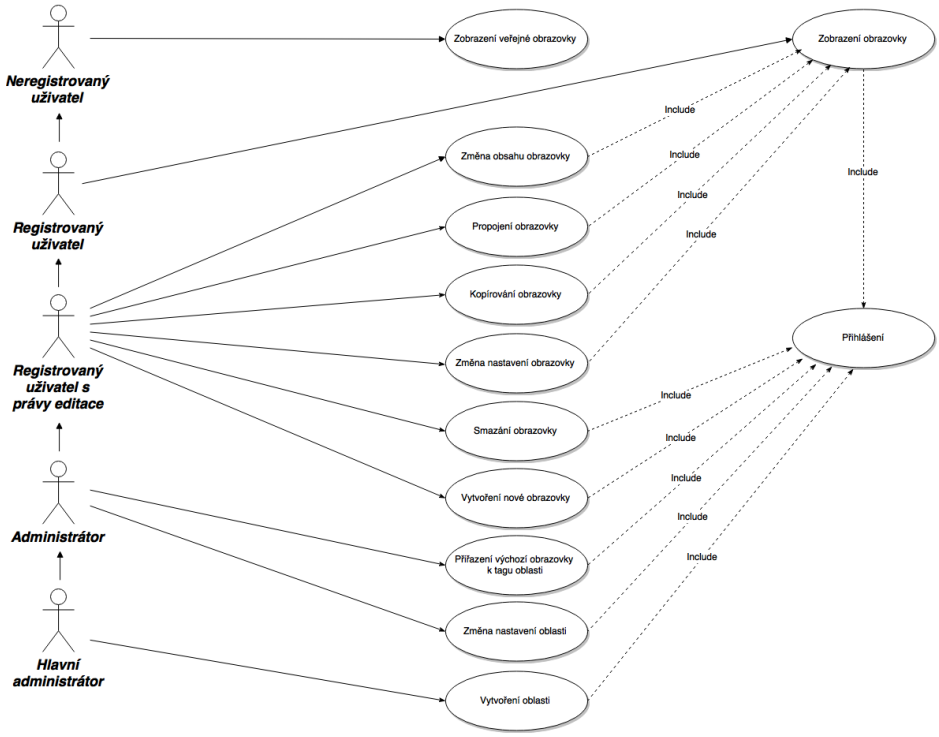
2.10.1.1 Uživatelské role

R1 Neregistrovaný uživatel

- Anonymní uživatel bez přihlašovacích údajů

R2 Registrovaný uživatel

- Registrovaný uživatel s vlastními přihlašovacími údaji
- Předek: R1 Neregistrovaný uživatel



Obrázek 2.17: Diagram případů užití

R3 Registrovaný uživatel s právy editace

- Registrovaný uživatel s vlastními přihlašovacími údaji, který je autorizován k úpravám vlastních obrazovek
- Předek: R2 Registrovaný uživatel

R4 Administrátor

- Registrovaný uživatel s vlastními přihlašovacími údaji, který je autorizován k většině úkonů v rámci aplikace
- Předek: R3 Registrovaný uživatel s právy editace

R5 Hlavní administrátor

- Registrovaný uživatel s vlastními přihlašovacími údaji, který je autorizován ke všem úkonům v rámci aplikace
- Předek: R4 Administrátor

2.10.1.2 Požadavky

F1 Zobrazení obrazovky

Požadovaná role: R1 Neregistrovaný uživatel

F1.1 Zobrazení veřejné obrazovky

F1.1.1 Systém zobrazí požadovanou veřejnou obrazovku.

F1.1.2 Jednotlivé bloky obrazovky zobrazí svůj obsah.

F1.1.3 Možnosti pro úpravu obrazovky se nezobrazí.

F1.2 Zobrazení obrazovky

F1.2.1 Systém zobrazí požadovanou obrazovku oprávněnému uživateli.

F1.2.2 Jednotlivé bloky obrazovky zobrazí svůj obsah.

F1.3 Zobrazení výchozí obrazovky

F1.3.1 Systém zobrazí požadovanou obrazovku uživateli s oprávněním ke zobrazení oblasti výchozí obrazovky.

F1.3.2 F1.2.2 Jednotlivé bloky obrazovky zobrazí svůj obsah.

F1.3.3 Možnosti pro úpravu obrazovky se nezobrazí.

F2 Správa obrazovky

Požadovaná role: R3 Registrovaný uživatel s právy editace

F2.1 Vytvoření obrazovky

F2.1.1 Systém zobrazí formulář s nastavením nové obrazovky.

F2.1.2 Formulář bude obsahovat výběr stanic obrazovky.

F2.1.3 Formulář bude obsahovat název a popis obrazovky.

F2.1.4 Formulář bude obsahovat tagy obrazovky.

F2.2 Nastavení obrazovky

F2.2.1 Systém zobrazí formulář s nastavením obrazovky.

F2.2.2 Formulář bude obsahovat výběr stanic obrazovky.

F2.2.3 Formulář bude obsahovat název a popis obrazovky.

F2.2.4 Formulář bude obsahovat tagy obrazovky.

F2.2.5 Formulář bude obsahovat nastavení viditelnosti obrazovky.

F2.3 Vytvoření a výběr bloku

F2.3.1 Systém zobrazí nový blok.

F2.3.2 Systém zobrazí formulář pro výběr typu bloku.

F2.3.3 Po zvolení bloku zobrazí systém možnosti nastavení bloku.

F2.4 Nastavení bloku

F2.4.1 Systém zobrazí formulář s nastavením bloku.

F2.4.2 Formulář bude obsahovat název a popis bloku.

F2.4.3 Formulář bude obsahovat další nastavení v závislosti na vybraném typu bloku.

F2.5 Propojení obrazovky

F2.5.1 Systém zobrazí formulář pro propojení dostupných obrazovek.

F2.5.2 Formulář bude obsahovat výběr propojených obrazovek.

F2.6 Kopírování obrazovky

F2.6.1 Systém zobrazí formulář pro kopírování obrazovky.

F2.6.2 Formulář bude obsahovat výběr částí, které mají být zkopírovány.

F2.6.3 Formulář bude obsahovat výběr obrazovek, které budou přepsány kopií.

F3 Správa výchozích obrazovek

Požadovaná role: R4 Administrátor

F3.1 Vytvoření výchozí obrazovky

F3.1.1 Systém zobrazí formulář s nastavením nové výchozí obrazovky.

F3.1.2 Formulář bude obsahovat název a popis obrazovky.

F3.1.3 Formulář bude obsahovat tagy obrazovky.

F3.2

F3.2.1 Systém zobrazí formulář s nastavením výchozí obrazovky.

F3.2.2 Formulář bude obsahovat název a popis obrazovky.

F3.2.3 Formulář bude obsahovat tagy obrazovky.

F3.2.4 Formulář bude obsahovat nastavení příslušnosti obrazovky k tagu oblasti.

F4 Správa oblastí

Požadovaná role: R5 Hlavní administrátor

F4.1 Vytvoření libovolné oblasti

F4.1.1 Systém zobrazí formulář s nastavením nové oblasti.

F4.1.2 Formulář bude obsahovat výběr rodičovské oblasti.

F4.1.3 Formulář bude obsahovat název a popis oblasti.

F4.1.4 Formulář bude obsahovat tagy oblasti.

Požadovaná role: R4 Administrátor

F4.2 Vytvoření vnořené oblasti

F4.2.1 Systém zobrazí formulář s nastavením nové vnořené oblasti.

F4.2.2 Formulář bude obsahovat název a popis oblasti.

F4.2.3 Formulář bude obsahovat tagy oblasti.

F4.3 Úprava nastavení oblasti

F4.3.1 Systém zobrazí formulář s nastavením oblasti.

F4.3.2 Formulář bude obsahovat název a popis oblasti.

F4.3.3 Formulář bude obsahovat tagy oblasti.

2.10.2 Nefunkční požadavky

N1 Aplikace bude funkční v prostředí pracoviště zadavatele

N2 Aplikace bude integrovatelná do CMS Drupal verze 7

N3 Aplikace bude využívat databázi MySQL

N4 Aplikace bude pracovat s daty ve formátu JSON a YAML

N5 Prostředí aplikace bude uživatelsky přehledné

N6 Aplikace bude podporována moderními internetovými prohlížeči

2.10.3 Systémové požadavky

2.10.3.1 Systémové požadavky na straně serveru

Doporučeným webovým serverem pro Drupal verze 7 je Apache HTTP Server verze 2 a vyšší. Alternativami k serveru Apache jsou servery NGINX verze 0.7 a vyšší a Microsoft IIS verze 5 a vyšší. Doporučená verze PHP je 5.3 a vyšší. V krajních případech je možné použít verzi 5.2.5. Standardní instalace Drupalu verze 7 vyžaduje 15 - 60 MB diskového prostoru, v závislosti na počtu instalovaných modulů.

Doporučenou databází pro Drupal verze 7 je MySQL verze 5.0.15 a vyšší s rozšířením PHP Data Objects. Dalšími možnostmi jsou PostgreSQL verze 8.3 a vyšší s rozšířením PHP Data Objects nebo SQLite verze 3.3.7 a vyšší. Databáze Microsoft SQL Server a Oracle jsou podporovány po doinstalování potřebných modulů.[21]

Symfony pak vyžaduje pro svůj běh PHP verze 5.5.9 a vyšší. Dále vyžaduje povolení direktivy JSON a ctype. Posledním požadavkem je nastavení časové zóny v rámci konfiguračního souboru php.ini

2.10.3.2 Systémové požadavky na straně klienta

Drupal 7 podporuje většinu standardních webových prohlížečů. Zde je jejich oficiální výčet:

- Internet Explorer verze 8 a vyšší
- Firefox verze 5 a vyšší
- Opera verze 12 a vyšší
- Safari verze 5 a vyšší
- Google Chrome

Podpora a povolení JavaScriptu v prohlížeči je pro funkčnost vizualizační komponenty nutná.

2.11 Doménový model

V rámci doménového modelu na diagramu 2.18 jsou znázorněny entity a vztahy mezi nimi.

2.11.1 Settings

Entita Settings slouží obecně k uložení nastavení jednotlivých komponent.

2.11.2 Category

Entita Category udržuje informace o stromové struktuře oblastí a dashboardů. Krom toho také zaznamenává stanice dostupné v dané oblasti.

2.11.3 Dashboard

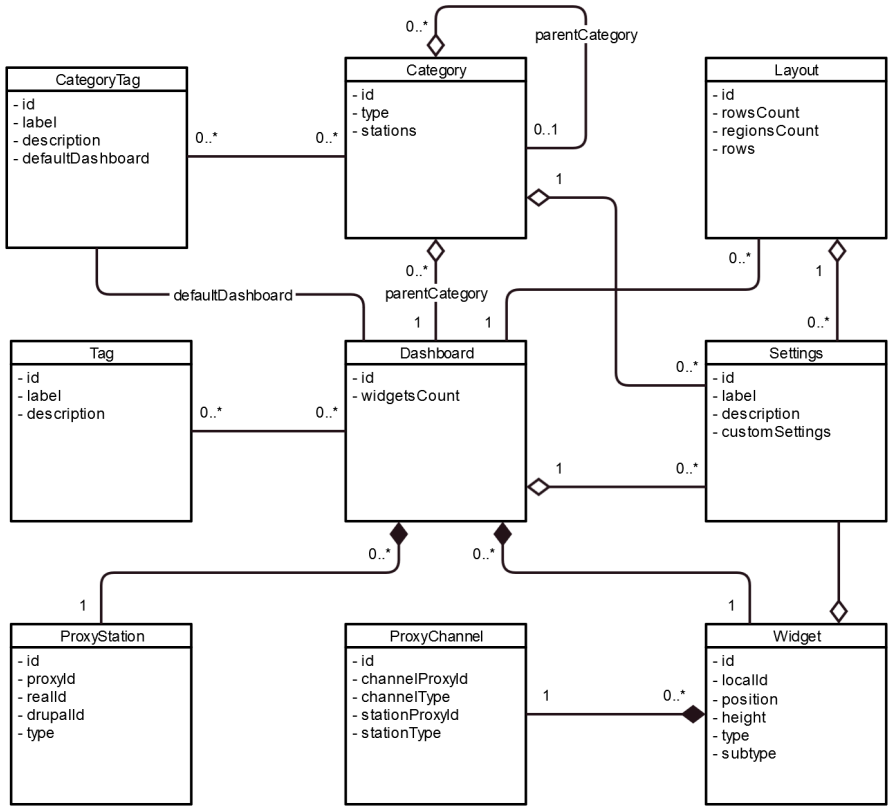
Entita Dashboard je ústřední entitou celé aplikace a propojuje jednotlivé ostatní komponenty do logického celku.

2.11.4 Layout

Entita Layout udržuje informaci o šabloně s řádky a sloupci, která určuje rozložení bloků v rámci dashboardu.

2.11.5 Widget

Entita Widget udržuje informace o své pozici v šabloně, o svém typu podtypu a dále udržuje vazbu na své vnitřní nastavení.



Obrázek 2.18: Diagram doménového modelu

2.11.6 Tag

Entita Tag slouží jako dynamický štítek určité skupiny dashboardů.

2.11.7 CategoryTag

Entita CategoryTag slouží podobně jako entita Tag k dynamické definici určité skupiny, tentokrát však skupiny oblastí. Kromě toho sebou také nese vazbu na výchozí dashboard, který se uživateli zobrazí v prázdné oblasti s tímto štítkem.

2.11.8 ProxyStation

Entita ProxyStation slouží jako prostředník ve spojení mezi dashboardem a konkrétní stanicí na serveru. Pro každý dashboard je definována jedna nebo více proxy stanic, kterým mohou být při kopírování přepsány pouze měnící se identifikátory pro získání potřebných dat.

2.11.9 ProxyChannel

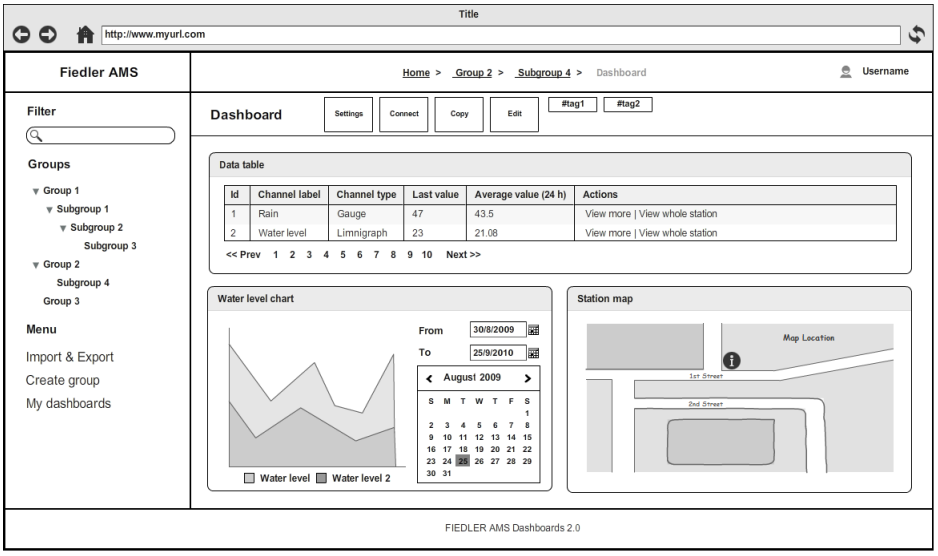
Podobně jako entita ProxyStation, slouží Entita ProxyChannel jako prostředník, tentokrát však vůči dashboardu. Finální cesta pro načtení potřebných dat je získána až kombinací entity ProxyChannel a entity ProxyStation. Tento systém zajišťuje nezávislost widgetů na konkrétních stanicích dashboardu.

2.12 Návrh uživatelského rozhraní

2.12.1 Schématický návrh rozhraní

2.12.1.1 Uživatelská obrazovka

Při schématickém návrhu uživatelské obrazovky byly zohledněny požadavky vyplývající z případů užití aplikace. Rozložení vychází z responzivního vzoru „Off Canvas“, který využívá možnosti skrytí levé části pro zařízení s menším rozlišením displeje. Hlavním cílem návrhu bylo intuitivní a kompaktní uživatelské rozhraní pro potřeby celé aplikace se snahou o zobrazení pouze podstatných informací a maximalizaci prostoru pro vizualizaci dat. Schématický návrh obrazovky se nachází na obrázku 2.19.

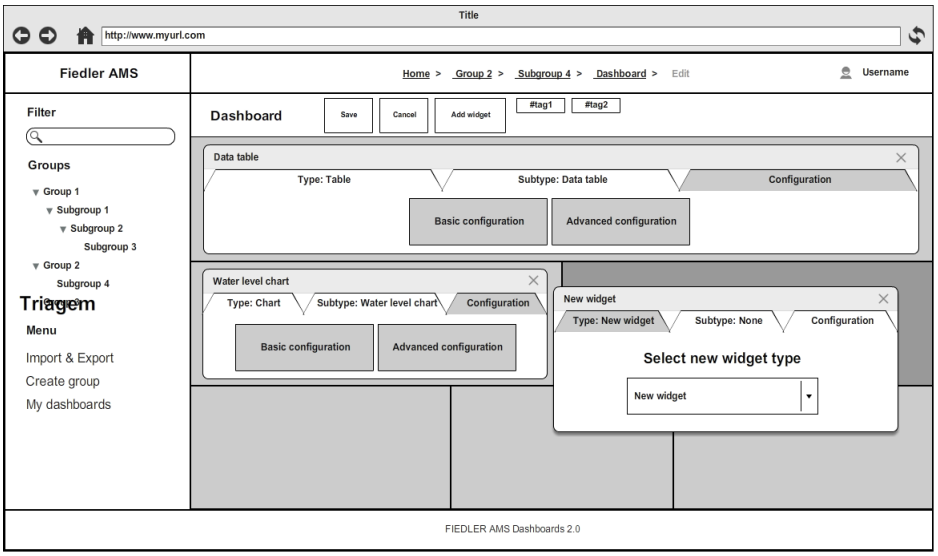


Obrázek 2.19: Návrh obrazovky

2.12.1.2 Editační mód uživatelské obrazovky

Návrh editačního módu uživatelské obrazovky vychází z kapitoly 2.2 a 2.5. Kromě konceptu výběru rozšíření také naznačuje možnost interaktivní mani-

pulace s jednotlivými bloky. Návrh editačního módu obrazovky se nachází na obrázku 2.20.



Obrázek 2.20: Návrh obrazovky v režimu úprav

2.12.1.3 Konfigurace rozšíření

Návrh konfigurace vychází rovněž z kapitoly 2.2, konkrétně pak především z její části 2.2.5. V rámci schématu 2.21 je zobrazen návrh základního nastavení tabulkového rozšíření.

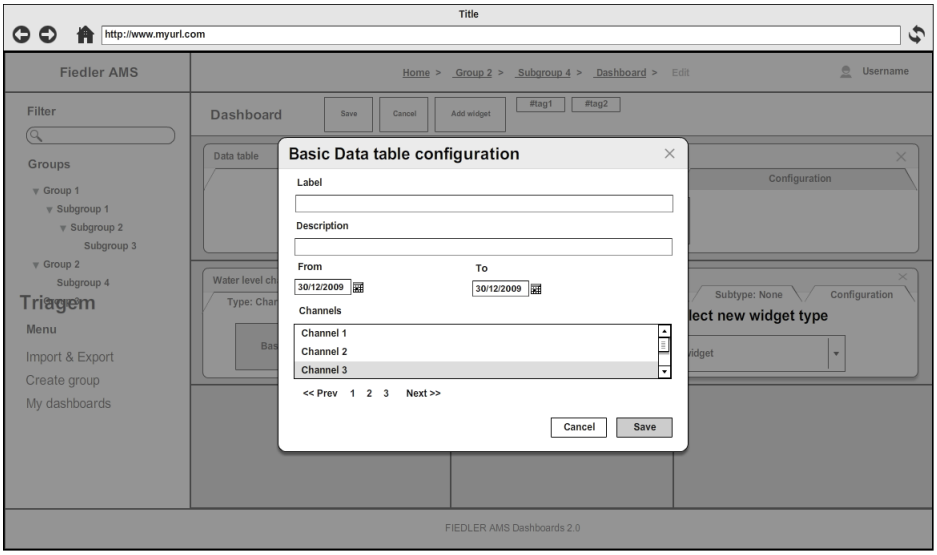
2.12.2 Výběr nástrojů pro tvorbu uživatelského rozhraní

Uživatelská rozhraní webových aplikací prošla během vývoje webu celou řadou inovací a změn. V historii velmi často docházelo k velkým nekonzistencím v uživatelských rozhraních, kdy každá společnost implementovala vlastní přístup nebo dokonce více různých přístupů. Vzniklý problém vedl ke snaze určitého sjednocení uživatelskému rozhraní v rámci webových aplikací.

Mezi osvědčené postupy při vývoji uživatelského rozhraní se také přidalo použití nástrojů, které respektují základní principy a všeobecné shody v oblasti moderního rozhraní webových aplikací. Z tohoto důvodu je podstatné analyzovat a vybrat vhodný nástroj pro implementaci uživatelského rozhraní vizualizační komponenty.

2.12.2.1 Bootstrap

Vizuální framework Bootstrap vychází z práce Marka Otta a Jacoba Thorntona, kteří začali s jeho vývojem pro interní sjednocení vizuálních nástrojů



Obrázek 2.21: Návrh konfigurace bloku

v rámci sociální sítě Twitter. Bootstrap je v podstatě sada nástrojů pro vývoj moderního responzivního uživatelského rozhraní. Dokumentace je rozdělena do několika specifických částí, které zahrnují všechny možnosti formou příkladů. Bootstrap je plně podporován v rámci prohlížeče Firefox od verze 44, Chrome od verze 48, IE od verze 8 a Safari od verze 8.[22]

2.12.2.2 Material Design

Za knihovnou Material Design, stojí společnost Google. Hlavním cílem knihovny je vývoj systému pro tvorbu uživatelského rozhraní, který poskytuje jednotnou uživatelskou zkušenost napříč všemi platformami. Přehledná dokumentace poskytuje dostatečné množství informací a příkladů. Material Design v porovnání s Bootstrapem podporují dřívější verze většiny prohlížečů. Výjimku tvoří pouze IE, který je podporován až od verze 10.[23]

2.12.2.3 Ostatní nástroje pro tvorbu uživatelského rozhraní

Existuje celá řada dalších nástrojů, které pokrývají různé skupiny prvků, vzhledů a funkcí uživatelského rozhraní. Jejich uživatelská základna ani možnosti využití však nedosahují srovnatelných kvalit v porovnání s výše uvedenými nástroji.

2.12.2.4 Zhodnocení

Jako nástroj pro tvorbu uživatelského rozhraní je vybrán Bootstrap. Hlavním důvodem je velká rozšířenost a s tím spojená dostupnost mnoha užitečných

rozšíření. Částečnou roli hraje také fakt, že je tato knihovna použita v rámci ostatních projektů společnosti FIEDLER AMS.

Kapitola 3

Implementace

Tato kapitola pojednává o způsobu implementace vyvíjené aplikace. V závěru kapitoly je zmíněno několik zajímavých implementačních problémů.

3.1 ORM

V rámci implementace datového modelu komponenty je použito objektově relační mapování ve formě ORM knihovny Doctrine. Výhodou použití ORM je lepší abstrakce při práci s relační databází. Nevýhodou ORM je horší výkon spojený s režií mapování dotazů.

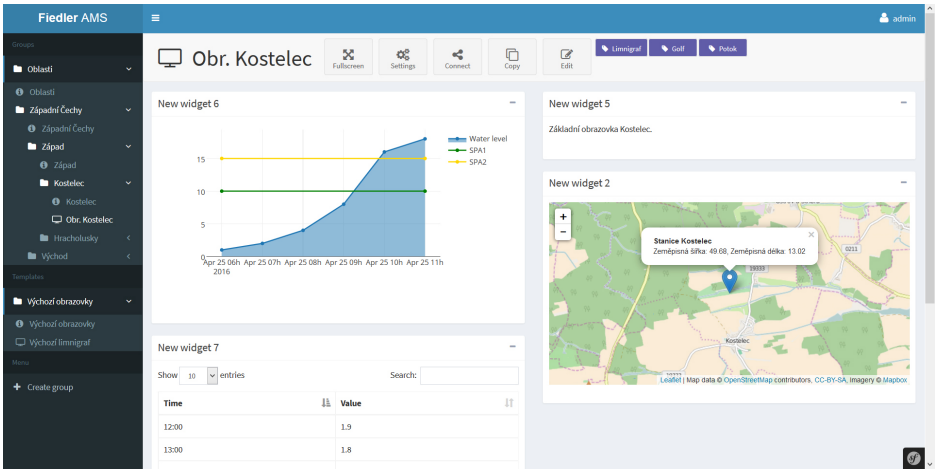
3.2 Uživatelské obrazovky

Podstatnou částí samotných obrazovek je jejich vykreslení v běžném režimu i v režimu úprav. Pro usnadnění uživatelské interakce s obrazovkou a jejími bloky je použita JavaScriptová třída obrazovky, která zaštiťuje veškeré operace spojené se zobrazením jednotlivých bloků, změnou jejich pořadí nebo jejich typu. Ukázka obrazovky v běžném režimu je zobrazena na obrázku 3.1.

Jako vstupní parametr nového objektu obrazovky je použita konfigurace obrazovky ve formátu JSON. Tato konfigurace obsahuje id obrazovky, kvůli následnému uložení obrazovky, počet bloků obrazovky, kvůli výpočtu id nových bloků, a dále již jen přehled konfigurací jednotlivých bloků. Ty sebou dále nesou své lokální id (v rámci obrazovky), své globální id (v rámci celé komponenty), dále také svou pozici, výšku, svůj typ, své nastavení a přehled kanálů, které aktuálně využívají.

3.3 Oblasti

Pro přístup k obrazovkám je použit strom oblastí, který rekurzivně vykresluje podoblasti a obrazovky patřící do dané oblasti. Tento strom je součástí levého



Obrázek 3.1: Ukázka obrazovky aplikace

panelu, který je dostupný jak v normální, tak i v minimalizované verzi pro rozšíření zobrazovacího prostoru.

3.4 Vstupní brána

V rámci potřeby asynchronního přístupu k datům byla jako součást komponenty implementována třída vstupní brány pro tyto požadavky. Součástí konceptu vstupní brány je také třída, která zapouzdřuje odpověď obsahující požadovaná data a další informace.

V současné chvíli zprostředkovává vstupní brána data pro zobrazení modulů a jejich nastavení. Do budoucna je však počítáno s rozšířením pro načítání dat ze stanic a kanálů skrze vzdálené REST API. Další možné využití existuje v oblasti komunikace s ostatními moduly Drupalu.

3.5 Bloky obrazovky

Správu modulů komponenty zajišťuje třída Modules, která se stará o dostupná modulární rošířeními. V současné době implementuje pouze metodu pro získání dostupných rozšíření. Do budoucna však bude rozšířena o možnosti povolení čí zakázání jednotlivých rozšíření pro vybrané uživatele nebo skupiny uživatelů. Případně také o instalaci nebo aktualizaci nových rozšíření skrze uživatelské rozhraní

V současné době jsou implementovány následující 4 rozšíření, které lze v rámci uživatelských obrazovek zobrazit.

3.5.1 Mapa

Blok mapa momentálně implementuje zobrazení mapy s fixní polohou stanice prostřednictvím knihovny LeafletJS. Jako mapové podklady jsou použity Open street maps.

3.5.2 Dynamický graf

Blok dynamický graf aktuálně zobrazuje několik typů dynamických grafů pro pevně zvolené časové období.

3.5.3 Tabulka

Blok tabulka zobrazuje tabulku dvojic času a hodnoty naměřené v daném čase pro vybraný kanál.

3.5.4 Popis obrazovky

Blok popis obrazovky zobrazuje uživatelem nastavený popis obrazovky.

3.6 Problémy řešené při implementaci

3.6.1 Asynchronní načítání obsahu

Během implementace se vyskytl problém s asynchronním načítání HTML obsahu. Tento problém byl v reprezentaci víceřádkového JavaScriptového řetězce. Nativně jsou totiž víceřádkové řetězce v JavaScriptu oddělovány buďto oddělením jednotlivých řádků řetězce a jejich následným spojením pomocí operátoru „+“ nebo znakem „\“ na konci každého řádku. Vzhledem k tomu, že požadovaný HTML obsah nebyl ve formátu víceřádkového řetězce, bylo potřeba tuto nekompatibilitu vyřešit. Pro tento účel byla navržena 3 řešení.

První z nich zahrnovalo doplnění všech takovýchto HTML souborů o znak JavaScriptového konce řádku. Toto řešení je velice nepraktické hned ze 2 důvodů. Za prvé je velice pracné a náročné udržovat takovéto HTML soubory, navíc by nebylo možné jejich zobrazení jinou formou než právě pomocí řetězce v JavaScriptu. Toto řešení bylo vyhodnoceno jako nevyhovující.

Jako druhé možné řešení bylo navrženo vytvoření parseru těchto HTML souborů, který by prováděl konverzi souborů do JavaScriptových řetězců. Toto řešení by bylo relativně jednoduché na implementaci, ale z hlediska další práce se soubory velice nesystémové.

Jako vhodné bylo shledáno až třetí řešení, které využívá běžných nástrojů jazyků PHP i JavaScript pro práci se soubory ve formátu JSON. Na straně PHP tak dojde k zakódování HTML souboru právě do formátu JSON, který je na straně JavaScriptu následně dekódován a vykreslen.

3.6.2 Formulářový prvek s tlačítkem přidat

Při implementaci formulářového prvku pro zadání tagů se vyskytl problém spojený s přidáváním nových prvků formuláře pomocí JavaScriptové funkce. Takto přidané prvky nebyli ze strany validace na straně serveru akceptovány, kvůli bezpečnostnímu systému formulářů Symfony.

Po delším hledání bylo nalezeno řešení v podobě nastavení vlastnosti formuláře pro povolení přidávání nových polí na straně klienta. Toto řešení sebou přináší nutnost dodatečné validace dat na straně serveru. Konkrétní řešení pak vypadá takto:

```
$form = $this->createFormBuilder(
    $data,
    array('allow_extra_fields' => true),
);
```

3.6.3 Načítání modulů z vlastní složky

Během vývoje systému modulů se vyskytl problém s voláním šablony jednotlivých modulů mimo standardní složku.

Řešením problému je definice další cesty v souboru routing.yml v rámci základní konfigurace frameworku Symfony. Konkrétní příklad cesty pro vykreslení pak nabývá tohoto tvaru:

```
@Prefix/custom/path/to/file
```

3.6.4 Přenos stavu aplikace mezi obrazovkami

V rámci uživatelského rozhraní komponenty se vyskytuje několik součástí obrazovky, které nabývají více stavů. (Například standardní a minimalizovaný levý ovládací panel.) Vzhledem k tomu, že HTTP protokol sebou nepřenáší stav aplikace, je potřeba přenesení těchto stavů zajistit jiným způsobem.

Tento problém byl vyřešen implementací JavaScriptové třídy PageState, která zahrnuje metody pro změny jednotlivých stavů. Samotná informace o stavu je přenášena ve formě GET parametrů.

Kapitola 4

Testování

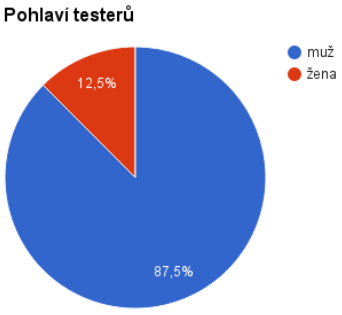
Tato kapitola pojednává o uživatelském a akceptačním testování aplikace. Je zde uveden přehled zjištěných problému a návrh možností jejich řešení.

4.1 Uživatelské testy

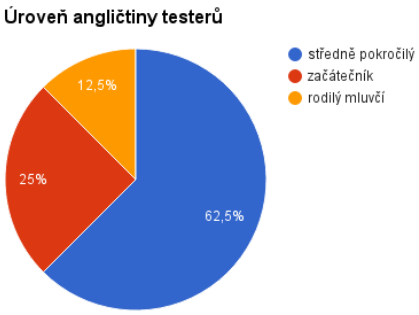
4.1.1 Testované role

Pro testy použitelnosti byly vybrány 2 z 5 uživatelských rolí definovaných v části s případy užití v této práci - Registrovaný uživatel a Registrovaný uživatel s právy editace. Tyto dvě role byly vybrány z důvodu největšího pokryti případů užití. Zbývající role se od vybraných dvou, z pohledu případů užití, liší minimálně.[24]

Přehled informací o účastnících testování je zobrazen v rácmci grafů 4.1, 4.2 a 4.3.



Obrázek 4.1: Pohlaví zůčastněných testerů



Obrázek 4.2: Úroveň znalosti angličtiny zúčastněných testerů

Obrázek 4.3: Věkové skupiny zúčastněných testerů

4.1.2 Registrovaný uživatel

Uživatelských testů v roli Registrovaný uživatel se zúčastnili čtyři muži a jedna žena s různou úrovní znalosti anglického jazyka, ve věkovém rozsahu 18-34 let. Někteří z nich se s původní aplikací nesetkali, ale všichni vyhovují vybraným personám. Zadání celého testu včetně jednotlivých úkolů je dostupné v příloze této práce.

Pro tento scénář bylo vytvořeno několik různě zanořených oblastí se třemi koncovými oblastmi třech různých stanic. Každá stanice obsahoval jednu obrazovku s několika bloky. Formou zjišťovacích úkolů byla testována orientace uživatele ve vytvořených obrazovkách a porozumění jejich obsahu.

V první části měli uživatelé za úkol určit datum a čas posledního spojení a posledních dat přenesených do databáze. V rámci této části nedocházelo k výrazným problémům. Z uživatelské zpětné vazby vyplynula pouze horší

orientace mezi těmito dvěma pojmy.

Ve druhé části měli uživatelé za úkol z bloku mapa určit polohu stanice. Tento úkol nečinil žádné problémy a uživatelé intuitivně pracovali s blokem mapy.

Ve třetí části měli uživatelé za úkol zjistit datum a čas, během kterého došlo k překročení 1. povodňového stupně. Tento úkol vedl k interaktivní práci s dynamickým grafem, kterou většina uživatelů po krátkém pátrání zvládla. Při snaze o přiblížení grafu uživatelé zaměňovali funkci přiblížení s funkcí označení úseku dat.

4.1.3 Registrovaný uživatel s právy editace

Uživatelských testů se zúčastnili tři muži se základní a středně pokročilou znalostí anglického jazyka, ve věkové skupině 25-34 let. Většina z nich již pracovala s původní aplikací. Zadání celého testu včetně jednotlivých úkolů je dostupné v příloze této práce.

V první části měli uživatelé za úkol vytvořit novou obrazovku s libovolnými bloky a určeným rozložením a názvem stanice. Při přesunu bloků do určeného rozložení, uživatelé v první chvíli nerozeznali možnost pohybu bloků po obrazovce. Přesunutí jednotlivých bloků se snažili dosáhnout skrze nastavení bloku. Dalším poznatkem při pozorování bylo nevhodné zobrazení krokového výběru bloku v rámci výběru jeho typu. Jednotlivé kroky zde byly vnímány jako záložky.

V druhé části měli uživatelé za úkol propojit obrazovky do skupiny. Tato část se setkala s relativně velkým nepochopením jejího účelu. Uživatelé neměli jasnou představu, čeho danou operací dosáhnou.

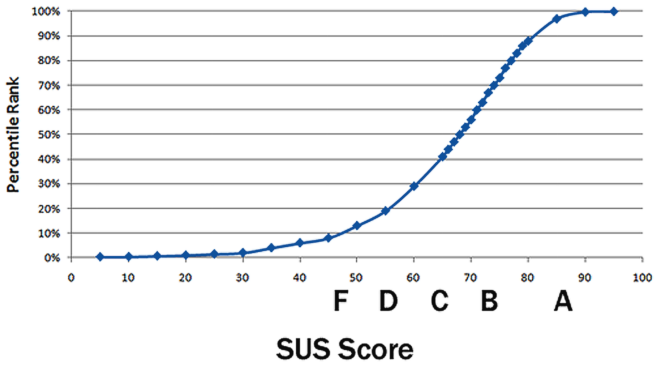
Ve třetí části měli uživatelé za úkol zkopírovat obrazovku, kterou dříve vytvořili na obrazovky již existující v rámci systému. Zde docházelo k různému pochopení akce kopírování obrazovky.

4.1.4 System Usability Scale

System Usability Scale neboli SUS je univerzální metoda hodnocení uživatelských testů. Autor této metody je John Brooke, který ji v roce 1986 vytvořil pro hodnocení uživatelských rozhraní aplikací v rámci terminálu VT100. SUS se postupem času ukázala být na tolik všestranná a objektivní, že se z ní později stal průmyslový standard.

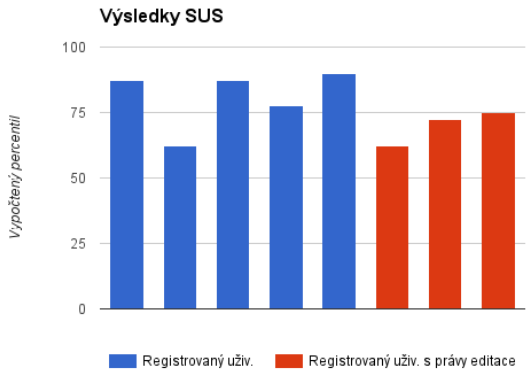
Metoda se skládá z 10 tvrzení, která jsou subjektivně hodnocena uživatelem na stupnici pravdivosti s rozsahem 1 - 5. Formulář s těmito 10 tvrzeními je uživateli předložen k vyplnění bezprostředně po použití testované aplikace. Následné vyhodnocení probíhá vypočtením bodové hodnoty každého tvrzení. Lichá tvrzení jsou kladná a jejich hodnota je určena jako x - 1, kde x je hodnota vyplněná uživatelem. Sudá tvrzení jsou záporná a jejich hodnota je určena jako

5 - x, kde x je hodnota vyplněná uživatelem. Po sečtení hodnot všech tvrzení je součet vynásoben koeficientem 2,5 a je určen výsledný percentil.[25]



Obrázek 4.4: Rozložení výsledků metody SUS

Formulář SUS byl po použití aplikace předložen všem uživatelům. Je nutné zdůraznit, že vzhledem k tomu, že výsledná hodnota vyhodnocených dotazníků není procentuální hodnotu úspěšnosti ale percentilem. Proto je pro interpretaci tohoto výsledku nutné znát průměrný percentil SUS, který dosahuje hodnoty 68. Lepší orientaci v hodnocení poskytuje také graf 4.4, který znázorňuje rozložení výsledků SUS. Výsledky jednotlivých uživatelů jsou znázorněny v grafu 4.5. Průměrem výsledků pro roli registrovaný uživatel byla vypočtena nadprůměrná hodnota 81. Průměrem výsledků pro roli registrovaný uživatel s právy editace byla vypočtena průměrná hodnota 70. Průměrem všech výsledných hodnot byl vypočten celkový mírně nadprůměrný percentil 76,875.



Obrázek 4.5: Výsledky metody SUS

Hodnota SUS vypovídá o dobré uživatelské přívětivosti aplikace. I přes tento výsledek je však nutné zabývat se změnami uživatelského prostředí apli-

kace, které byly analyzovány v rámci testů použitelnosti. Některé problémy byly rozpoznány v rámci testovaných scénářů přímo, jiné byly identifikovány při následném rozhovoru o průběhu testování s jednotlivými uživateli.

4.1.5 Navrhované změny uživatelského prostředí

4.1.5.1 Vizuální rozlišení dat

Popis problému: Uživateli trvalo déle rozlišení rozdílu mezi posledním odesláním dat stanice a posledním připojením stanice.

Navrhované řešení: Pro lepší orientaci v těchto informacích je vhodné přidat ikony reprezentující datový přenos a libovolné připojení.

4.1.5.2 Funkce dynamického grafu

Popis problému: Uživatel si nebyl vědom všech dostupných funkcí při práci s grafem.

Navrhované řešení: Lze přidat nápovědu, která bude stručně a přehledně vysvětlovat jednotlivé funkce. Podobná nápověda se může stát standardní součástí všech dostupných bloků.

4.1.5.3 Organizace bloků v editačním módu

Popis problému: Uživatel hledal možnost přesunu bloků v rámci jeho nastavení.

Navrhované řešení: Zde se naskytuje možnost přidání změny umístění do nastavení bloku nebo zvýraznění možnosti bloky přesouvat.

4.1.5.4 Kroky při výběru bloku

Popis problému: Během výběru bloku obrazovky se uživatel pokoušel přepínat mezi jednotlivými kroky nastavení s předpokladem, že jde o záložky.

Navrhované řešení: Pro lepší orientaci v tomto výběru je nutné zvýraznit, že jde o posloupnost jednotlivých kroků.

4.1.5.5 Propojení obrazovek

Popis problému: Uživatel nedokázal identifikovat účel akce propojení obrazovek.

Navrhované řešení: Je potřeba doplnit nápovědu, která by vysvětlovala jednotlivé dostupné akce menu a jejich důsledky.

4.1.5.6 Pojem kopírování obrazovky

Popis problému: Uživatel nedokázal správně určit výsledek akce kopírování.

Navrhované řešení: Je potřeba pojem kopírování obrazovky lépe specifikovat, například jako „Hromadná úprava obrazovek“ nebo „Přepsání obrazovek“.

4.1.5.7 Stromová struktura oblastí

Popis problému: Uživatel zmínil, že se mírně ztrácí ve stromové struktuře oblastí.

Navrhované řešení: Pro lepší orientaci je možné, aby pravá část vždy zobrazovala přehled aktuálně vybrané oblasti. Další zlepšení může přinést filtrování oblastí a obrazovek na základě textového vstupu.

4.1.5.8 Jméno obrazovky formou odkazu

Popis problému: Uživatel očekával, že název obrazovky povede na její zobrazení.

Navrhované řešení: Je potřeba přidat odkaz ke jménu obrazovky.

4.1.5.9 Vylepšení formulářových prvků

Popis problému: Uživatel očekával větší pohodlí v rámci formulářů aplikace. Například zmínil automatické doplňování rozepsaných tagů nebo předvyplnění stanice oblasti při vytváření obrazovky.

Navrhované řešení: Je potřeba provést analýzu formulářových prvků pro zlepšení uživatelské zkušenosti.

4.1.5.10 Omezená zpětná vazba

Popis problému: Uživatel očekával větší zpětnou vazbu od aplikace.

Navrhované řešení: Aplikace může přidat více zpětných vazeb pro uživatele formou informativních zpráv či změnou některých vizuálních prvků.

4.1.5.11 Vizualizace propojených obrazovek

Popis problému: Uživatel požadoval vizualizaci propojených obrazovek.

Navrhované řešení: Aplikace může implementovat přehled uživatelem definovaných skupin obrazovek s možností jejich dalších hromadných úprav.

4.1.6 Zhodnocení

V rámci testů použitelnosti byly odhaleny některé nedostatky uživatelského rozhraní aplikace, které budou v budoucnu odstraněny implementací změn navržených v této kapitole. V rámci vyhodnocení metody SUS byl zjištěn dobrý stav uživatelského rozhraní a celkové použitelnosti aplikace.

4.2 Akceptační testy

4.2.1 Podmínky testů

Scénáře akceptačních testů vycházejí z případů užití definovaných v této práci. Samotné testování probíhalo v místě pracoviště zadavatele.

4.2.2 Testování v místě pracoviště zadavatele

Po dokončení implementace proběhlo akceptační testování komponenty v místě pracoviště zadavatele. V rámci tohoto testování bylo zjištěno několik chyb, které byly ohodnoceny podle závažnosti. Nejzávažnější chyby byly hodnoceny písmenem A. Středně závažné chyby byly hodnoceny písmenem B. Méně závažné chyby byly hodnoceny písmenem C. Následuje jejich seznam.

Chyba: Při zobrazení výchozí obrazovky není pověřenému uživateli nabídnuta možnost úpravy a změny nastavení obrazovky. Tato možnost je zobrazena pouze v přehledu výchozích obrazovek.
Závažnost: C

Chyba: Při kopírování obrazovky nejsou správně filtrovány dostupné propojené obrazovky.
Závažnost: B

Chyba: Při výběru typu bloku dojde ke skrytí hlavičky bloku.
Závažnost: C

Chyba: Při přesunu některých bloků do oblastí s rozložením do 3 sloupců nedochází ke správnému ukotvení bloku.
Závažnost: C

Chyba: Při přesunu bloků v rámci jednoho regionu v některých případech nedochází ke změně pořadí bloků po uložení změn.
Závažnost: C

4.2.3 Zhodnocení

V rámci akceptačního testování bylo zjištěno 5 chyb. Z toho 4 chyby byly hodnoceny jako méně závažné a 1 chyba byla hodnocena jako středně závažná. Zjištěná chyba ohodnocena jako středně závažná byla odstraněna. Ostatní zjištěné chyby budou odstraněny v rámci budoucího vývoje. Implementace vizualizační komponenty byla akceptována.

Závěr

V rámci této práce byla provedena rozsáhlá analýza předchozí verze vizualizační komponenty, která spolu s analýzou konceptu a trendů v oblasti uživatelských obrazovek položila dobrý základ pro nový návrh systému uživatelských obrazovek a systému modulárních rozšíření. Navrženo bylo také nové uživatelské rozhraní aplikace, které dodržuje principy rozhraní moderní webové aplikace. Následně byla komponenta dle návrhu implementována a otestována v prostředí zadavatele.

Dále je provedeno vyhodnocení splněných požadavků definovaných zadáním této práce.

Proveďte revizi systému tvorby uživatelských obrazovek / dashboardů a šablonovacího systému těchto dashboardů. V kapitole 2.1.1 byla provedena analýza současného řešení tvorby uživatelských obrazovek a šablonovacího systému v porovnání se současnými trendy v oblasti uživatelských obrazovek. Na základě výsledků této analýzy byl v kapitole 2.5 proveden nový návrh systému obrazovek. Dále byla provedena implementace tohoto návrhu, jejíž detaily jsou popsány v kapitole 3.

Proveďte revizi systému modulárních rozšíření / widgetů se zaměřením na usnadnění tvorby a začlenění nových widgetů. V kapitole 2.1.2 byla provedena analýza současného řešení widgetů. Následně byl v kapitole 2.6 proveden nový návrh tohoto systému v rámci zvolených technologií.

Analyzujte a navrhněte nové UI pro práci s dashboardy a konfiguraci widgetů s důrazem na intuitivní a snadné ovládání. V kapitole 2.1.3 byla provedena analýza současného uživatelského rozhraní s ohledem na nejčastěji prováděné uživatelské scénáře. Následně bylo v kapitole 2.12 navrženo nové uživatelské rozhraní, které dodržuje principy a standardy pro návrh moderního a intuitivního uživatelského rozhraní.

Proveďte analýzu, návrh a implementaci nových widgetů zaměřených na vizualizaci dat v tabulkách, grafech a mapách dle požadavků zadavatele. V kapitolách 2.1.4 byla provedena analýza widgetů zaměřených na vizualizaci dat v tabulkách, grafech a mapách. V kapitole 2.8.2 byla provedena analýza dostupných open source řešení pro vizualizaci dat v grafech. Následně byla provedena implementace widgetů zobrazujících tabulky, grafy a mapy dle požadavků zadavatele, jejíž detaily jsou popsány v kapitole 3.

Otestujte novou verzi komponenty v prostředí vedoucího práce / zadavatele. Po dokončení implementace komponenty proběhlo akceptační a uživatelské testování v prostředí pracoviště zadavatele, jejichž výsledky jsou popsány v kapitolách 4.1 a 4.2.

Celkově mi práce na tomto projektu přinesla mnoho nových zkušeností v oblastech analýzy, návrhu, implementace a testování webových aplikací. V rámci programování jsem se seznámil s vývojem aplikace za pomoci frameworku Symfony. Dále jsem získal cenné zkušenosti spojené s objektovým návrhem aplikace a s návrhem jejího datového modelu. Jako další přínos vnímám také zkušenosti získané při návrhu uživatelského rozhraní a následném uživatelském testování.

Literatura

[1] Fiedler.company: FIEDLER AMS [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.fiedler.company

[2] PÍŠA, J.: *Vizualizacní komponenta do CMS Drupal*, kapitola 1. Praha: České vysoké učení technické v Praze, 2014, str. 3, [cit. 2016-05-01]. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/pisajan1_2014bach.pdf

[3] NIELSEN, J.: *Usability Engineering*. Boston: Academic Press, 1993, ISBN 0-12-518405-0, [cit. 2016-05-01].

[4] FEW, S.: *Information dashboard design: the effective visual communication of data*, kapitola 1. Sebastopol: O'Reilly, 2006, ISBN 978-0-596-10016-2, str. 34, [cit. 2016-05-01]. Dostupné z: http://oreilly.de/catalog/infodashboard/chapter/ch01.pdf

[5] PÍŠA, J.: *Vizualizacní komponenta do CMS Drupal*, kapitola 2. Praha: České vysoké učení technické v Praze, 2014, str. 9, [cit. 2016-05-01]. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/pisajan1_2014bach.pdf

[6] Axosoft.com: Axosoft [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.axosoft.com

[7] Klipfolio.com: Klipfolio [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.klipfolio.com

[8] Netvibes.com: Netvibes [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.netvibes.com

[9] Clicdata.com: Clicdata [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.clicdata.com

[10] Dashing.io: Dashing [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.dashing.io

[11] RazorFlow.com: RazorFlow [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.razorflow.com

[12] CMF Wikipedia: the free encyclopedia. [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://en.wikipedia.org/wiki/CMF

[13] Drupal.org: Drupal 8 [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.drupal.org/8

[14] Yaml.org: YAML [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.yaml.org

[15] PÍŠA, J.: *Vizualizacní komponenta do CMS Drupal*, kapitola 2. Praha: České vysoké učení technické v Praze, 2014, str. 25, [cit. 2016-05-01]. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/pisajan1_2014bach.pdf

[16] PÍŠA, J.: *Vizualizacní komponenta do CMS Drupal*, kapitola 2. Praha: České vysoké učení technické v Praze, 2014, str. 27, [cit. 2016-05-01]. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/pisajan1_2014bach.pdf

[17] C3js.org: C3js [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.c3js.org

[18] NVD3.org: NVD3 [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.nvd3.org

[19] Plot.ly: Plotly [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.plot.ly

[20] PÍŠA, J.: *Vizualizacní komponenta do CMS Drupal*, kapitola 3. Praha: České vysoké učení technické v Praze, 2014, str. 43, [cit. 2016-05-01]. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/pisajan1_2014bach.pdf

[21] PÍŠA, J.: *Vizualizacní komponenta do CMS Drupal*, kapitola 2. Praha: České vysoké učení technické v Praze, 2014, str. 33, [cit. 2016-05-01]. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/pisajan1_2014bach.pdf

[22] Getbootstrap.com: Bootstrap [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.getbootstrap.com

[23] Google: Materialize [online]. 2016, [cit. 2016-05-01]. Dostupné z: http://www.materializecss.com

[24] PÍŠA, J.: *Vizualizacní komponenta do CMS Drupal*, kapitola 5. Praha: České vysoké učení technické v Praze, 2014, str. 53, [cit. 2016-05-01]. Dostupné z: `https://dip.felk.cvut.cz/browse/pdfcache/pisajan1_2014bach.pdf`

[25] SAURO, J.: Measuring Usability With The System Usability Scale (SUS) [online]. 2014, [cit. 2016-05-01]. Dostupné z: `http://www.measuringusability.com/sus.php`

Příloha **A**

Seznam použitých zkratek

3D Three-dimensional

AMS Automatic Monitoring Systems

API Application Programming Interface

CMF Content Management Framework

CMS Content management system

CSS Cascading Style Sheets

DSS Decission Support System

GSM Global System for Mobile

HTTP Hypertext Transfer Protocol

HTML HyperText Markup Language

IE Internet Explorer

IIS Internet Information Services

JSON JavaScript Object Notation

MIT Massachusetts Institute of Technology

MVC Model View Controller

OEM Original Equipment Manufacturer

ORM Object-relational mapping

PNG Portable Network Graphics

PHP PHP: Hypertext Preprocessor

REST Representational State Transfer

RSS Rich Site Summary

SQL Structured Query Language

SUS System Usability Scale

SVG Scalable Vector Graphics

UI User Interface

URL Uniform Resource Locator

WebGL Web Graphics Library

YAML YAML Ain't Markup Language

Příloha B

Slovníček pojmů

controller část návrhového vzoru MVC, která vykonává logiku aplikace

dynamický graf graf, který umožňuje interakci se zobrazenými daty

dashoard uživatelská obrazovka

formát konfigurace formát, který definuje povinné i nepovinně parametry jednotlivých rozšíření

framework softwarová struktura, která slouží jako podpora při programování a vývoji nebo organizaci softwarových projektů

hook funkce, která definuje určité rozhraní jednotlivých modulů a umožnuje jejich nezávislou komunikaci

konduktometr zařízení určené pro měření měrné elektrické codivosti

layout definice rozložení oblastí bloků v rámci obrazovky

limnigrafická stanice měřící stanice, která slouží k pozorování vodního stavu

oxymetr zařízení určené pro měření obsahu kyslíku

parser nástroj pro textovou analýzu

plugin dodatečné rozšíření aplikace

statický graf graf, který neumožňuje interakci se zobrazenými daty

tag textová reprezentace určitého označení

telemetrická stanice měřící stanice, která odesílá naměřená data přes GSM spojení do vzdálené databáze

twig šablonovací systém frameworku Symfony

widget blok uživatelské obrazovky

Příloha C

Instalační příručka

C.1 Úvod

V této příručce se nachází návod na instalaci vizualizační komponenty ve formě modulu pro CMS Drupal 7.

C.2 Instalace CMS Drupal 7

Nejdříve je nutné provést instalaci samotného CMS Drupal verze 7, kterou lze získat stažením z oficiálních webových stránek nebo pomocí nástroje Drush.

Příkazy potřebné pro instalaci pomocí Drush:

```
drush dl drupal-7.x

drush site-install standard
--account-name=admin
--account-pass=123
--db-url=mysql://DbUser:Pass@localhost/DbName
```

C.3 Instalace komponenty

Dalším krokem je instalace modulu komponenty a následné povolení tohoto modulu. To lze provést buďto skrze uživatelské rozhraní CMS Drupal nebo opět pomocí nástroje Drush.

Příkazy potřebné pro instalaci pomocí Drush:

```
git clone GitUser@bitbucket.org/Repository/fm_vision_v2

drush -y en fm_vision_v2
```

C.4 Před použitím

Před prvním použitím aplikace, je třeba se přihlásit v rámci komponenty jako hlavní administrátor a vytvořit kořenovou oblast stromové struktury oblastí a dále oblast pro zařazení výchozích obrazovek.

Příloha D

Uživatelská příručka

D.1 Úvod

V této příručce se nachází manuál pro správné použití aplikace.

D.2 Práce s oblastmi

D.2.1 Vytvoření nové oblasti

1. V levé nabídce vyberte oblast, ve které chcete oblast vytvořit a stiskněte přehled této oblasti.
2. Stiskněte položku vrchního menu „+ Group“.
3. Vyplňte potřebná nastavení a stiskněte tlačítko „Create group“.

D.2.2 Úprava nastavení oblasti

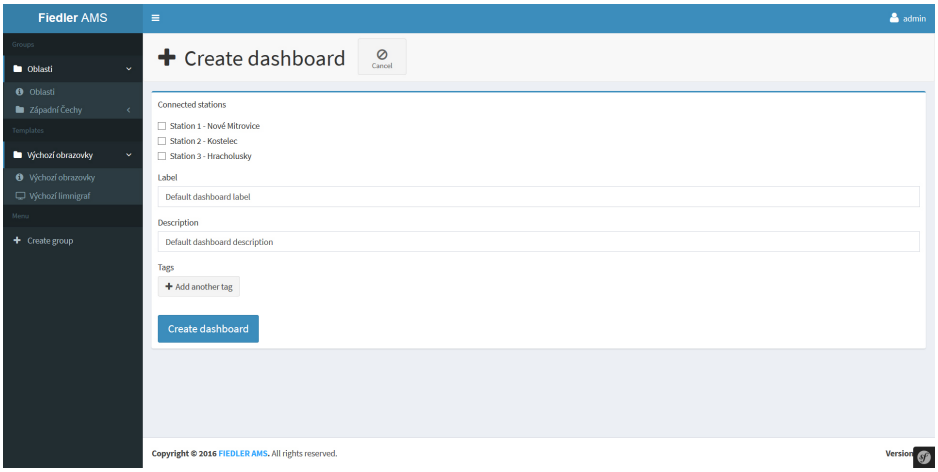
1. V levé nabídce vyberte oblast, ve které chcete, ve které chcete změnit nastavení a stiskněte přehled této oblasti.
2. Stiskněte položku vrchního menu „Settings“.
3. Vyplňte potřebná nastavení a stiskněte tlačítko „Save group settings“.

D.3 Práce s obrazovkami

D.3.1 Vytvoření nové obrazovky

1. V levé nabídce vyberte oblast, ve které chcete obrazovku vytvořit a stiskněte přehled této oblasti.
2. Stiskněte položku vrchního menu „+ Dashboard“.

3. Vyplňte potřebná nastavení a stiskněte tlačítko „Create dashboard“.



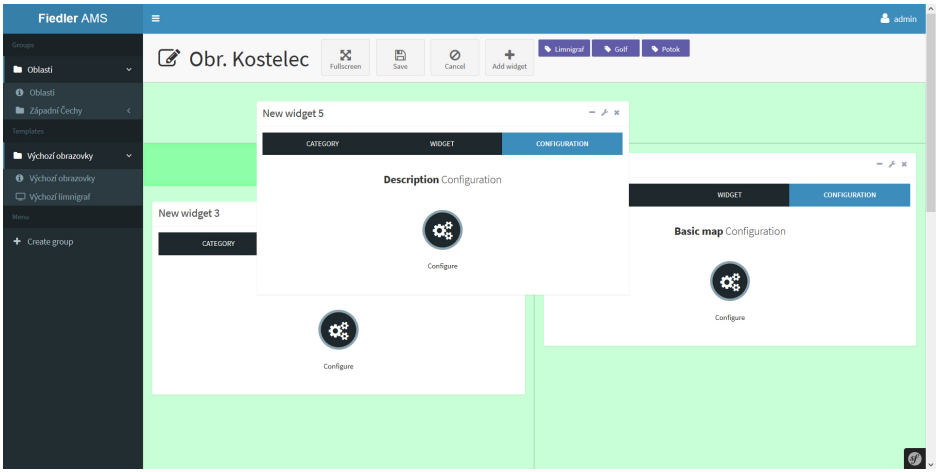
Obrázek D.1: Formulář pro vytvoření nové obrazovky

D.3.2 Úprava obrazovky

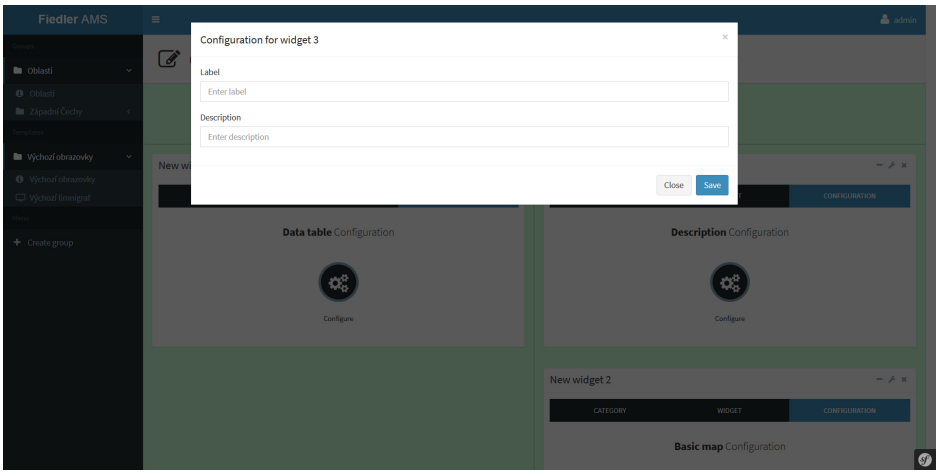
1. Přejděte na stránku obrazovky, kterou chcete upravit.
2. Stiskněte položku vrchního menu „Edit“.
3. Pro přidání nových bloků stiskněte položku vrchního menu „Add widget“.
4. Po přidání bloku provedete výběr typu v rámci vloženého bloku.
5. Dále je možné upravit nastavení bloku pomocí tlačítka „Configure“.
6. Bloky je možné libovolně přesunovat v rámci obrazovky do viditelných regionů.
7. Pro smazání bloku stiskněte křížek v pravé části bloku.
8. Pro uložení provedených úprav stiskněte položku vrchního menu „Save“.

D.3.3 Změna nastavení obrazovky

1. Přejděte na stránku obrazovky, které chcete změnit nastavení.
2. Stiskněte položku vrchního menu „Settings“.
3. Vyplňte potřebná nastavení a stiskněte tlačítko „Save dashboard settings“.



Obrázek D.2: Obrazovka v režimu úprav



Obrázek D.3: Nastavení bloku

D.3.4 Propojení obrazovky

1. Přejděte na stránku obrazovky, kterou chcete kopírovat.
2. Stiskněte položku vrchního menu „Connect“.
3. Vyberte možnosti propojení a stiskněte tlačítko „Save dashboard connections“.

D.3.5 Kopírování obrazovky

1. Přejděte na stránku obrazovky, kterou chcete kopírovat.
2. Stiskněte položku vrchního menu „Copy“.

3. Vyberte možnosti kopírování a stiskněte tlačítko „Copy template to selected dashboards“.

D.3.6 Smazání obrazovky

1. V levé nabídce vyberte oblast, ze které chcete obrazovku smazat a stiskněte přehled oblasti.

2. Stiskněte tlačítko ve sloupci „Delete“ v řádce obrazovky, kterou chcete smazat.

D.3.7 Zobrazení obrazovky v režimu celé obrazovky

1. Přejděte na stránku obrazovky, kterou chcete zobrazit v režimu celé obrazovky.

2. Stiskněte položku vrchního menu „Fullscreen“.

3. Pro ukončení režimu celé obrazovky stiskněte položku vrchního menu „Exit fullscreen“ nebo klávesu „Escape“.

D.4 Práce s výchozími obrazovkami

D.4.1 Vytvoření nové výchozí obrazovky

1. V levé nabídce vyberte oblast výchozích obrazovek a stiskněte přehled této oblasti.

2. Stiskněte položku vrchního menu „+ Dashboard“.

3. Vyplňte potřebná nastavení a stiskněte tlačítko „Create dashboard“.

D.4.2 Úprava výchozí obrazovky

1. V levé nabídce vyberte oblast výchozích obrazovek a stiskněte přehled této oblasti.

2. Stiskněte tlačítko ve sloupci „Edit“ v řádku obrazovky, kterou chcete upravit.

3. Pro přidání nových bloků stiskněte položku vrchního menu „Add widget“.

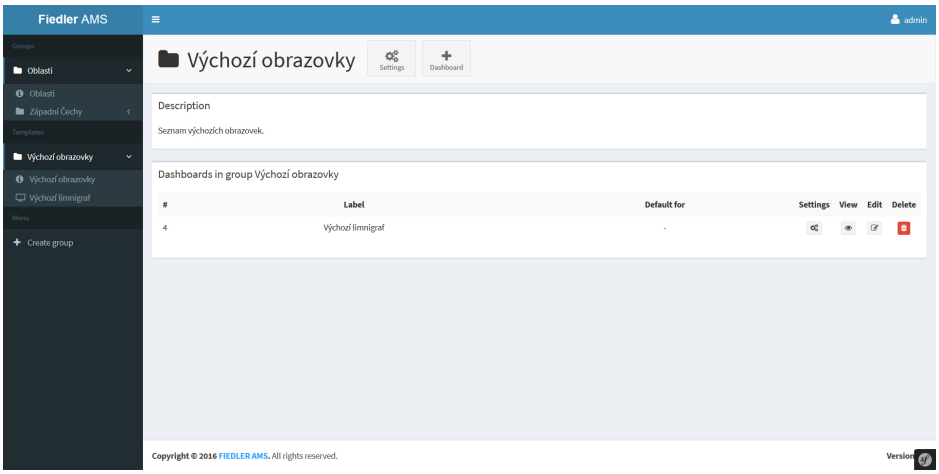
4. Po přidání bloku provedete výběr typu v rámci vloženého bloku.

5. Dále je možné upravit nastavení bloku pomocí tlačítka „Configure“.

6. Bloky je možné libovolně přesunovat v rámci obrazovky do viditelných regionů.

7. Pro smazání bloku stiskněte křížek v pravé části bloku.

8. Pro uložení provedených úprav stiskněte položku vrchního menu „Save“.



Obrázek D.4: Přehled výchozích obrazovek

D.4.3 Změna nastavení výchozí obrazovky

1. V levé nabídce vyberte oblast výchozích obrazovek a stiskněte přehled této oblasti.

2. Stiskněte tlačítko ve sloupci „Settings“ v řádku obrazovky, kterou chcete upravit.

3. Vyplňte potřebná nastavení a stiskněte tlačítko „Save dashboard settings“.

Příloha E

Uživatelské testy

E.1 Dotazník před testem

Pohlaví: muž / žena
Úroveň anglického jazyka: začátečník / středně pokročilý / pokročilý / rodilý mluvčí
Věková skupina: 18-24 / 25-34 / 35-44 / 45-55 / více než 55

E.2 Slovníček pojmů

Obrazovka = dashboard
Blok obrazovky = widget

E.3 Registrovaný uživatel

E.3.1 Informace o datových přenosech

Určete datum a čas posledního spojení stanice Nové Mitrovice s databází.
Určete datum a čas posledních naměřených dat stanice Nové Mitrovice.

E.3.2 Informace o poloze stanice

Určete z mapy adresu stanice Kostelec.

E.3.3 Informace o stavu vody

Určete v jakém čase překročila výška hladiny měřeného toku hranici 1. povodňového stupně v grafu stanice Kostelec.

E.4 Registrovaný uživatel s právy editace

E.4.1 Vytvoření nové obrazovky

Vytvořte novou obrazovku s libovolným názvem pro stanici Nové Mitrovice. Obrazovka by měla obsahovat alespoň 3 různé widgety. Využijte rozložení do 2 sloupců.

E.4.2 Propojení obrazovek

Propojte Vámi vytvořenou obrazovku se dvěma již existujícími obrazovkami jiných stanic.

E.4.3 Kopie obrazovky

Použijte všechna nastavení Vámi vytvořené obrazovky z úkolu 1 na propojené obrazovky z úkolu 2.

E.5 Dotazník po testu

Ohodnoťte jednotlivá tvrzení 1 - 5 body. 1 bod znamená „Rozhodně nesouhlasím“, 5 bodů znamená „Rozhodně souhlasím“.

1. Myslím si, že bych tuto aplikaci rád/a intezivně využíval/a

2. Myslím si, že aplikace je zbytečně komplexní

3. Myslím si, že je snadné tuto aplikaci používat

4. Myslím si, že bych potřeboval/a konzultaci s technickou podporou, abych mohl/a tuto aplikaci používat

5. Myslím si, že různé funkce jsou do aplikace dobře integrované

6. Myslím si, že tato aplikace je příliš nekonzistentní

7. Myslím si, že většina lidí by se rychle naučila tuto aplikaci používat

8. Myslím si, že tato aplikace je příliš těžkopádná na použití

9. Cítil/a jsem se velice sebevědomě při používání této aplikace

10. Musel/a jsem se dozvědět příliš mnoho informací před použitím této aplikace

Příloha F

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `images` ..... obrázky potřebné k vytvoření výsledného PDF
  - `DP_Pisa_Jan_2016.bbl` ..... soubor s obsahem použité literatury
  - `DP_Pisa_Jan_2016.tex` ..... zdrojová forma práce ve formátu LaTeX
- `text`
  - `DP_Pisa_Jan_2016.pdf` ..... text práce ve formátu PDF