České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra softwarového inženýrství
Obor: Aplikace softwarového inženýrství

Elektronická úřední deska pro Magistrát města Děčín

Electronic Official Board for the Municipality of Děčín

BAKALÁŘSKÁ PRÁCE

Vypracoval: Vít Barták
Vedoucí práce: Ing. Michal Moc
Rok: 2021

České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra softwarového inženýrství

Akademický rok 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	**Vít Barták**
Studijní program:	**Aplikace přírodních věd**
Obor:	**Aplikace softwarového inženýrství**
Název práce česky:	**Elektronická úřední deska pro Magistrát města Děčín**
Název práce anglicky:	**Electronic Official Board for the Municipality of Děčín**

Pokyny pro vypracování:

1. Proveďte rešerši dostupných řešení.
2. Na základě provedené rešerše navrhněte vlastní aplikaci.
3. Naprogramujte navrženou aplikaci.
4. Otestujte naprogramovanou aplikaci.
5. Napište uživatelskou příručku na ovládání aplikace.

Doporučená literatura:

[1] DAVID, M. *HTML5: Designing Rich Internet Applications. 2nd ed.* Burlington: Focal Press, 2013. ISBN 978-0-240-82076-7.

[2] STANÍČEK, P. *CSS Kaskádové styly: kompletní průvodce.* Praha: Computer Press, 2003. ISBN 80-722-6872-4.

[3] VRÁNA, J. *1001 tipů a triků pro PHP.* Brno: Computer Press, 2010. ISBN 978-80-251-2940-1.

Jméno a pracoviště vedoucího práce:

Ing. Michal Moc
Katedra softwarového inženýrství, Fakulta jaderná a fyzikálně inženýrská, ČVUT v Praze

Jméno a pracoviště konzultanta:

Mgr. Roman Žaloudek
Statutární město Děčín, Magistrát města Děčín, Mírové náměstí 1175/5, 405 38 Děčín IV

Michal Moc
.....
vedoucí práce

Datum zadání bakalářské práce: 16. 10. 2020

Termín odevzdání bakalářské práce: 7. 7. 2021

Doba platnosti zadání je dva roky od data zadání.

.....
garant oboru

.....
vedoucí katedry

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská
-1-

.....
děkan

V Praze dne 16. 10. 2020

Prohlášení

Prohlašuji, že jsem bakalářskou práci na téma *Elektronická úřední deska pro Magistrát města Děčín* vypracoval pod vedením vedoucího práce samostatně a použil jsem pouze zdroje (literární nebo softwarové) uvedené v této práci.

V Děčíně dne ....

Vít Barták

Poděkování

Děkuji vedoucímu práce panu Ing. Michalu Mocovi za odborné vedení, rady a odezvu při vypracování této práce. Dále chci poděkovat Magistrátu města Děčín za spolupráci jmenovitě panu Mgr. Romanu Žaloudkovi.

Vít Barták

Název práce:
Elektronická úřední deska pro Magistrát města Děčín

Autor: Vít Barták

Studijní program: Aplikace přírodních věd
Obor: Aplikace softwarového inženýrství
Druh práce: Bakalářská práce

Vedoucí práce: Ing. Michal Moc
Katedra softwarového inženýrství, Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze
Konzultant: Mgr. Roman Žaloudek
Statutární město Děčín, Magistrát města Děčín, Mírové náměstí 1175/5, 405 38 Děčín IV

Abstrakt: Cílem této bakalářské práce je návrh a implementace softwarové úřední desky pro Magistrát města Děčín. Výsledkem je prototyp webové aplikace, která slouží jako softwarová část v elektronické úřední desce. Cílová platforma pro provoz této aplikace je jednodeskový počítač. V analytické kapitole se práce nejprve zabývá úřední deskou z pohledu zákona, jsou zde definovány vlastnosti elektronické úřední desky, dále kapitola zpracovává nynější řešení města Děčín a následně analýzu existujících řešení na českém trhu. Větší část práce pak řeší návrh a implementaci této webové aplikace v ohledu ke specifikovaným vlastnostem. Ke konci práce je představen prototyp elektronické úřední desky, jehož provoz je testován na Magistrátě města Děčín. V tomto prototypu slouží naprogramovaná aplikace jako hlavní, softwarová část. V závěru je pak celá práce zhodnocena a jsou navržena případná zlepšení.
Klíčová slova: elektronická úřední deska, Fat-Free Framework, webová aplikace, jednodeskový počítač

Title:
Electronic Official Board for the Municipality of Děčín

Author: Vít Barták

Abstract: The aim of this bachelor thesis is the design and implementation of a software official board for the Municipality of Děčín. The outcome is a prototype web application, which serves as the software part in an electronic official board. The target operating platform of this application is a single-board computer. In the analytical chapter the thesis deals at first with how the law looks at official boards, attributes of an electronic official board are also specified here. The analytical chapter then deals with current solution of the Děčín City and existing solutions on the Czech market are analyzed. The bigger part of this thesis deals with the design and implementation of said web application with regards to the specified attributes. By the end of the thesis a prototype of electronical official board is introduced, the operation of which is tested by the Municipality of Děčín. In this prototype the programmed web application serves as a main software component. In the conclusion the whole work is evaluated and possible improvements are proposed.
Key words: electronic official board, Fat-Free Framework, web application, single-board computer

Obsah

Úvod **11**

1 Analýza současných řešení **13**
1.1 Úřední deska z pohledu zákona 13
1.2 Rozdíl mezi DÚD a EÚD 13
1.3 XML feed 14
1.4 Současné řešení MM Děčín 14
1.5 Firmy zabývající se výrobou a prodejem elektronických úředních desek 15
1.6 Analýza ceny a technické specifikace desek 16

2 Návrh webové aplikace **21**
2.1 Funkce elektronické úřední desky a požadavky na aplikaci 22
2.2 Návrh serverové části 24
2.2.1 Uživatelé a jejich práva 25
2.2.2 Třídy a diagram tříd 27
2.2.3 Kontroléry 28
2.2.4 ORM modely 28
2.3 Návrh databáze 29
2.4 Návrh uživatelského rozhraní 30
2.4.1 Občanské UI 30
2.4.2 Administrátorské UI 31
2.5 Návrh hardwarové části 31

3 Implementace webové aplikace **33**
3.1 Použité technologie 33
3.2 Fat-Free Framework 35
3.2.1 Soubory a složky 36
3.2.2 Přesměrování a práva 37
3.3 Kontroléry 37
3.3.1 MainController.php 37
3.3.2 Index.php 38
3.3.3 User.php 39
3.3.4 Content.php 40
3.3.5 Board.php 43
3.3.6 ImageY.php 44
3.3.7 Stats.php 44
3.4 Databáze a ORM třídy 44

3.5 Uživatelské rozhraní 46
3.5.1 Společné soubory 46
3.5.2 Občanské rozhraní 46
3.5.3 Správcovské rozhraní 48

4 Nasazení a testování prototypu 49
4.1 Nasazení na jednodeskový počítač 49
4.2 Instalační manuál 50
4.2.1 Linux (produkční prostředí) 50
4.2.2 Windows (vývojové prostředí) 53
4.3 Prototyp EÚD na Magistrátě města Děčín 53
4.4 Webová digitální úřední deska 55

Závěr 57

Literatura 59

Přílohy 61

A Obsah přiloženého CD 61

B Obrázky 63

Úvod

Ve snaze o digitalizaci společnosti se objevují různá řešení pro usnadnění komunikace mezi státem a občany. Jedním z nich je elektronická úřední deska, ta umožňuje jednosměrnou komunikaci s občanem ať už přes internet nebo lokálně – většinou přímo na zdi úřadu. Úředníkům z městských nebo obecních úřadů elektronická deska navíc usnadňují práci při vyvěšování a správě dokumentů, ať jsou to vyhlášky nebo různá oznámení. V kontrastu s vyvěšováním tištěných listin do zasklené tabule se dá správa úřední desky provádět přes počítač vzdáleně.

Obec je ze zákona povinna provozovat veřejně přístupnou úřední desku, vznikají tak různé nároky na implementaci elektronické úřední desky. Z těchto nároků plynou nevýhody elektronických úředních desek oproti těm klasickým, hlavními jsou pořizovací a provozní cena elektronických desek. Vysoká cena vyplývá z nároků na provedení desky, ta musí být robustní a odolná proti povětrnostním vlivům, musí být neustále v provozu apod. Elektronické desky jsou vyráběné specializovanými výrobci, kteří mají na výrobě a provozu desky samozřejmě zisk. Vzhledem k tomu, že města a obce mají rozpočet složený z peněz daňových poplatníků, je otázka nákladů na pořízení a provoz elektronických úředních desek vcelku důležitá. Peníze z rozpočtu mohou jít do důležitější infrastruktury, než jsou úřední desky.

Tato bakalářská práce se v několika kapitolách zabývá návrhem open-source, rozšiřitelné webové aplikace, která při vhodném nasazení poslouží městu Děčín jako nová elektronická úřední deska. Využitím této aplikace pro postavení elektronické úřední desky by město Děčín mohlo ušetřit hned v několika rovinách. Náklady jako jsou licenční poplatky a cena za softwarové vybavení desky jsou u tohoto řešení nulové.

V první kapitole rozebírám pohled, jakým se český zákon dívá na úřední desky. Jsou zde specifikované určité vlastnosti, které mají elektronické úřední desky. Je zde vysvětlen rozdíl mezi digitální úřední deskou a elektronickou úřední deskou, tyto dva pojmy se však v mé práci mohou zaměňovat. V průběhu kapitoly porovnávám různé elektronické úřední desky od českých výrobců. Zajímá mě především jejich cena a technická specifikace, okrajově i softwarové vlastnosti. Ke konci kapitoly navrhuji několik způsobů, jakými by se daly snížit náklady na pořízení a provoz elektronické úřední desky.

V druhé kapitole se zaměřuji na návrh webové aplikace, která má vlastnosti elektronické úřední desky specifikované v první kapitole. Webová aplikace je nenáročná a může běžet na jednodeskovém počítači. Uvádím zde požadavky a funkce takové aplikace a pomocí UML a ERA diagramů ji vizualizuji. Zaměřuji se na uživatele a jejich práva v rámci elektronické úřední desky. Také rozebírám PHP třídy (a jejich

metody), které tvoří backend celé digitální úřední desky. Nakonec jsou představeny dva typy uživatelského rozhraní pro občany a úředníky. Částečně se kapitola zabývá i návrhem hardwarové části elektronické úřední desky.

Po návrhu aplikace následuje její implementace, ta je popsána v kapitole třetí. Zde představuji a zdůvodňuji použité technologie, pomocí kterých jsem výslednou aplikaci naprogramoval. Speciálně se zaměřuji na Fat-Free Framework, který slouží jako páteř celé webové aplikace. Rozebírám podrobněji všechny funkce vlastních PHP tříd (kontrolérů), které aplikaci ovládají. Na to navazuji implementací ORM datových modelů a uživatelského rozhraní. Druhá a třetí kapitola se prolínají, obě popisují tu samou aplikaci jen na jiné úrovni abstrakce.

V poslední kapitole je sepsán instalační manuál pro mou aplikaci, pomocí kterého jsem zprovoznil ve spolupráci s Magistrátem města Děčín prototyp elektronické úřední desky. Prototyp je dotykový a chybí mu již pouze ochranný kovový kryt, ve kterém by byla obrazovka chráněna proti povětrnostním a jiným vlivům. Krátce se zde zmiňuji i o webové úřední desce, která v tuto chvíli běží na hlavní stránce magistrátu města Děčín. Webová úřední deska obsahuje ořezaný kód hlavního programu.

Kapitola 1

Analýza současných řešení

1.1 Úřední deska z pohledu zákona

O funkci úřední desky se zmiňuje zákon hned několikrát. Pro potřeby této práce je nejdůležitější první odstavec v § 26 správního řádu:

(1) Každý správní orgán zřizuje úřední desku, která musí být nepřetržitě veřejně přístupná. Pro orgány územního samosprávného celku se zřizuje jedna úřední deska. Obsah úřední desky se zveřejňuje i způsobem umožňujícím dálkový přístup.[5]

Z tohoto paragrafu plynou důležité informace týkající se samotné implementace elektronické úřední desky. Musí být neustále veřejně dostupná, tím vzniká nutnost provozovat desky někde, kam se veřejnost dostane vždy – například venku nebo v nezamčené budově.

Z tohoto zákona vznikají různé nároky na technické provedení elektronické úřední desky – dostupnost i při výpadku internetu, provedení odolné vandalismu apod. Jak dále zjistíme, výrobci těchto desek dodávají zařízení, která těmto nárokům vyhovují. Dále správní řád nařizuje dostupnost přes dálkový přístup – v dnešní době to může znamenat přístup k desce přes webové stránky.

1.2 Rozdíl mezi DÚD a EÚD

Digitální úřední deskou (DÚD) můžeme chápat program, který zastává funkci úřední desky. To může být například webová stránka provozovaná příslušným magistrátem.

Elektronická úřední deska (EÚD) naproti tomu je fyzické elektronické zařízení většinou sestávající alespoň z počítače, obrazovky a ochranného krytu. Elektronická úřední deska nahrazuje klasickou papírovou úřední desku a má oproti ní několik výhod i nevýhod. Hlavní výhodou je jednoduchá správa oznámení a vyhlášek úřadem a jednoduchý přístup a ovládání pro občany. Nevýhodou je její provozní a pořizovací cena, která se může vyšplhat do statisícových částek. Na elektronické úřední desce běží digitální úřední deska jako obslužný program. Jak už bylo řečeno, cílem této

bakalářské práce je návrh a implementace programu (digitální úřední desky), který poběží v rámci fyzické elektronické úřední desky.

1.3 XML feed

Město Děčín, jako jiná města, provozuje spisovou službu. Tato spisová služba produkuje pro funkci úředních desek dedikovaný XML feed (jeho funkce je podobná RSS feedu), ten je pro implementaci softwaru úřední desky klíčový. XML feed úřední desky je veřejně dostupný[1]. Když se v této práci zmiňuji o XML souboru, XML feedu apod. myslím tím právě tento soubor, který je ve všední dny několikrát aktualizován. DÚD používá tento XML soubor jako centrální zdroj dat pro funkci úřední desky.

1.4 Současné řešení MM Děčín

Magistrát města Děčín v současné době provozuje tři elektronické úřední desky. Dvě jsou umístěny vedle sebe na vnější straně budovy magistrátu na adrese *Mírové nám. 1175/5, Podmokly, 405 38 Děčín*. Třetí deska se nachází uvnitř druhé budovy magistrátu na adrese *28. října 1155/2, 405 02 Děčín*. Desky jsou napojeny na intranet magistrátu a zároveň tak na již zmíněný XML feed.

Na obrázku 1.1 lze vidět, že jedna deska je připevněna na zdi níže než druhá. Je to kvůli přístupu pro občany na invalidním vozíku. Toto řešení je poněkud nehospodárné, místo dvou desek by totiž stačila jedna, která by měla obrazovku s větší úhlopříčkou. Tato obrazovka by byla polohována na výšku. Pak by přístup pro invalidy mohl řešit software. Tímto způsobem to řeší i čeští výrobci elektronických úředních desek. Pokud bude město provozovat pouze dvě desky stejného typu (jednu na každé straně města), náklady na pořízení a provoz úředních desek se sníží o třetinu.

Podle zákona 500/2004 [5] musí být úřední deska přístupná i *vzdáleně*. Město Děčín mělo minimálně dvě různé webové stránky (na jednu byl přístup z hlavní webové stránky města Děčín), které sloužily jako webové úřední desky. V současnou chvíli nemám na tyto zmíněné webové úřední desky přístup. Děčínské webové úřední desky se svým vzhledem nepodobaly hlavní webové stránce, ze které na ně bylo odkazováno. Jelikož hl. webová stránka města Děčín[2] běží na redakčním systému Joomla, naskytlo se řešení v podobě Joomla rozšíření. Joomla rozšíření již integruje desku přímo do hlavní stránky. Toto rozšíření je vedlejším produktem při vývoji hlavní webové aplikace v této bakalářské práci a dále se o něm zmiňuji v kapitole 4.

[1] Dostupné na této adrese http://77.95.47.242/ost/xml/export.php?command=ud

[2] Webové stránky MM Děčín https://www.mmdecin.cz

Obrázek 1.1: Úřední desky na budově děčínského magistrátu

1.5 Firmy zabývající se výrobou a prodejem elektronických úředních desek

Na českém trhu se výrobou a prodejem elektronických úředních desek zabývá mnoho firem, zde jsou některé z nich:

- BSS Praha s.r.o.[3]
- DigiDay Czech s.r.o.[4]
- Digitální panely s.r.o.[5]
- GROUP24 INNOVATIONS s.r.o.[6]
- Ki-Wi Digital s. r. o.[7]
- SPOJMONT Brno, spol. s r.o.[8]

[3]http://www.bsspraha.cz/
[4]https://www.digitalninosice.cz/
[5]https://www.digitalnipanely.cz/
[6]https://www.obec24.cz/
[7]https://www.ki-wi.cz/
[8]http://www.venkovni-lcd.cz/

Za spolupráce s panem Mgr. Žaloudkem z MM Děčín jsem získal cenové nabídky od těchto firem:

- DigiDay Czech s.r.o.
- Digitální panely s.r.o.
- GROUP24 INNOVATIONS s.r.o.
- Ki-Wi Digital s. r. o.

Na tomto vzorku cenových nabídek provedu analýzu ceny a technické specifikace nabízených desek. Všechny cenové nabídky jsou dostupné v PDF souborech na přiloženém CD, reprezentováno v příloze A. Následující tabulku 1.1 a část kapitoly zakládám na těchto PDF souborech.

1.6 Analýza ceny a technické specifikace desek

Vlastnosti elektronických úředních desek, které jsou pro účely porovnání *hlavní*:

- Cena
- Paušální poplatky (měsíční popřípadě roční za správu, za licenci)
- Odolnost panelu (proti povětrnostním podmínkám a vandalismu)
- Úhlopříčka obrazovky
- Hardwarové vybavení počítače

Cena je zde za elektronickou desku jako fyzické zařízení, vynechávám například cenu za dopravu, instalaci, školení apod. – tyto vedlejší náklady se liší od výrobce. Cenu za softwarové vybavení desky a paušální poplatky kategorizuji zvlášť. Důležitou vlastností desek je také termoregulace, většina těchto desek (venkovní modely) dokáže operovat v mrazech i ve vysokých teplotách. Řešení od KiWi například pracuje v teplotách od -30 °C do +50 °C.

Lze také kategorizovat *ostatní* vlastnosti, například:

- Dotyková obrazovka
- Režim při výpadku internetu
- Invalidní režim
- Vzdálená správa
- Dotyková klávesnice

- Úspora energie (úsporný režim)

- Statistika provozu (například návštěvnosti)

- Servis (popř. pozáruční servis)

Ostatní vlastnosti mají do různé míry všechny srovnávané desky. Firmy dodávají funkcionálně velmi podobný produkt (i když cenově různý), který se liší především v detailech (např. vzhled uživatelského rozhraní).

Některé z ostatních vlastnosti bych mohl také nazvat implicitními, jelikož nepřímo vyplývají už ze zákona (režim při výpadku internetu). Většina těchto vlastností je navíc implementována softwarově, nemá tudíž smysl je porovnávat z cenového hlediska, to hlavně proto, že softwarové řešení v této bakalářské práci je zdarma a pořizovací a licenční poplatky za software tak odpadají. Dá se také tvrdit, že tyto vlastnosti jsou obsaženy už v ceně desky. Porovnám desky pouze podle hlavních vlastností:

Elektronické úřední desky od firmy DigiDay						
Úhlopříčka	**Umístění**	**Odolnost**	**HW**	**Cena za desku**	**Cena za SW**	**Poplatky**
43"	Vnitřní (H)	5mm, IP55	V1	118 500 vč. DPH	-	-
	Vnější (H)	7mm, IP66 kov		218 500 vč. DPH	-	-
65"	Vnější (V)			258 500 vč. DPH	-	-

Elektronické úřední desky od firmy Digitální panely s.r.o.						
Úhlopříčka	**Umístění**	**Odolnost**	**HW**	**Cena za desku**	**Cena za SW**	**Poplatky**
43"	Vnější (H)	6mm, kov	V1	109 800 bez DPH	30 000 bez DPH	2000 bez DPH
55"				138 941 bez DPH		
65"				168 112 bez DPH		
43"	Vnější (V)			169 900 bez DPH		
55"				179 900 bez DPH		
65"				210 900 bez DPH		

Elektronické úřední desky od firmy Ki-Wi Digital s.r.o						
Úhlopříčka	**Umístění**	**Odolnost**	**HW**	**Cena za desku**	**Cena za SW**	**Poplatky**
23"	Vnitřní (H)	ocelový plech	V3	až 60 400 bez DPH	V ceně	až 7200 ročně bez DPH
55"	Vnější (V)	tvrzené sklo, IP65, kov	V2	179 900 bez DPH	50 000 bez DPH	až 6000 ročně (volitelné)

Elektronické úřední desky od firmy GROUP24 INNOVATIONS s.r.o.						
Úhlopříčka	**Umístění**	**Odolnost**	**HW**	**Cena za desku**	**Cena za SW**	**Poplatky**
55"	Vnější (V)	neuvedeno	V4	389 000 bez DPH	V ceně	V ceně
75"				499 000 až 649 000 bez DPH		

Tabulka 1.1: Elektronické desky různých výrobců

Vysvětlivka k tabulce 1.1:

- Úhlopříčka obrazovky v palcích
- Umístění:
  - Venku nebo uvnitř budovy
  - **V**ertikální nebo **H**orizontální poloha obrazovky
- Odolnost:
  - *6mm*: Tloušťka tvrzeného skla v milimetrech
  - *IP65*: Stupeň ochrany krytem (stupeň krytí)
  - *kov*: Materiál šasi
- HW: **V**arianty počítačového vybavení desky
  - V1: Intel, 128 GB SSD, 8 GB RAM, WiFi, LAN, Windows 10
  - V2: Intel Pentium G4400T (2,9 GHz), 128 GB SSD, 4 GB RAM, WiFi, LAN, Windows 10 IoT Enterprise
  - V3: Intel Core i3-8100 (3,6 GHz), 128 GB SSD, 4 GB RAM, LAN, Windows 10 IoT Enterprise
  - V4: Intel Core i3, 128 GB SSD, 4 GB RAM, WiFi, LAN, Windows 10

Z technické specifikace plyne, že desky jsou až na drobnosti totožným produktem. Některé desky napříč výrobci dokonce obsahují totožné počítače.

Elektronické úřední desky tuzemských firem se pohybují v rozsahu od 60 400 Kč do 649 000 Kč (bez slevy). V extrémním případě, kdy by město pořídilo tři nové desky za 649 tisíc korun, částka by se vyšplhala k bezmála dvěma milionům. Když posoudíme cenu vnějších desek okolo průměru ceny, může se cena tří desek vyšplhat zhruba od 600 tisíc po 750 tisíc (za fyzická zařízení). K této částce musíme připočíst ostatní náklady v řádech deseti tisíců (software, montáž apod.). To jsou relativně vysoké částky za zařízení, které je ve své podstatě obrazovka napojená na počítač v kovovém krytu. Nepochybně je tento kryt robustní a dokáže čelit povětrnostním podmínkám i vandalům (jak lze vidět na obrázku 1.1).

Nenavrhuji snižovat náklady na zřízení a provoz v ohledu k ochrannému krytu, ale vzhledem k ostatním složkám celého systému. Dle mého názoru jsou počítače v těchto deskách předimenzované a stačil by jednodeskový počítač typu Raspberry Pi[9] – tím by se ušetřily i provozní náklady. Deska nevykonává nijak paměťově nebo výpočetně náročné úkony, pouze zobrazuje data v grafickém uživatelském rozhraní. Pokud by se toto grafické rozhraní prokázalo jako zvláště náročné na grafický procesor, existují také jednodeskové počítače (například NVIDIA Jetson)[10], které jsou pro práci s grafikou uzpůsobené.

Ušetřit prostředky se dá i na pořizovací ceně softwaru a na práci okolo provozu desky, za kterou si výrobce účtuje poplatky. V pozdější části této práce představím prototyp elektronické desky, který se snaží snížit tyto náklady. Desku by si mohlo město Děčín sestavit samo a samo ji i provozovat. Takto se ušetří peníze daňových poplatníků a zbude víc prostředků na podstatnější výdaje, než jsou úřední desky.

Desky obsahují licencovaný software a některé za příplatek částečně provozuje podnik, který desku vyrobil. Při analýze nákladů na pořízení a provoz je tedy na místě myslet i na to, že provoz stojí nejen elektřinu, ale i periodické poplatky (například roční poplatek za vzdálenou správu od Ki-Wi). Těmto poplatkům se dá vyhnout použitím open-source softwaru vytvořeného na míru městu Děčín a také pomocí self-hostingu.

[9] https://www.raspberrypi.org/products/raspberry-pi-4-model-b/
[10] https://www.nvidia.com/cs-cz/autonomous-machines/jetson-store/

Kapitola 2

Návrh webové aplikace

V této kapitole popisuji návrh webové aplikace, která slouží jako softwarová část v systému elektronické úřední desky. Svou aplikaci jsem pracovně nazval `yadb` (Yet Another Digital Board). Rozebírám zde nejen funkcionalitu úřední desky ale i návrh serverové části, databázové části i grafického rozhraní a požadavky na ně. Částečně zde rozeberu i návrh hardwarové části elektronické úřední desky, přestože kompletní hardwarové řešení není cílem této práce. Používám anglické názvosloví pro jména tříd, tabulek, funkcí proměnných apod. a to především z estetických důvodů (také je to běžnou praxí).

První podkapitola se věnuje požadavkům na webovou aplikaci a funkcím, které by obecně měla mít elektronická úřední deska.

Druhá podkapitola podrobněji řeší problematiku serverového návrhu aplikace. Jsou zde popsány různé návrhy *sítě* úředních desek (dvě a více v jednom městě) a jejich výhody a nevýhody. V této podkapitole řeším i role, které mají uživatelé v mé aplikaci. Z těchto rolí vyplývá systém práv, který každému druhu uživatelů zpřístupňuje určitou funkcionalitu a možnosti použití aplikace.

Ve třetí podkapitole se zabývám návrhem jednoduché databáze, která dostatečně pokrývá zmíněné nároky aplikace.

V předposlední podkapitole řeším problematiku dvojího uživatelského rozhraní. Deska má dva hlavní typy uživatelů, *občany* a *úředníky*, každý interaguje s jinou částí aplikace, a proto je zapotřebí dvojího uživatelského rozhraní.

Nakonec trochu odbočím do hardwarové problematiky a proberu aspekty hardwarové části elektronické úřední desky.

Chtěl bych zde vyjasnit určitý jev, který by mohl mást čtenáře. Tato aplikace, ačkoliv je webová, není určena pro provoz na internetu, je určena pro lokální provoz na jednodeskovém počítači. K implementaci jsem však zvolil programovací jazyky pro tvorbu webových stránek. Výsledná aplikace je určena pro provoz na zařízení podobném tomu na obrázku 1.1, pro její běh je potřeba webového prohlížeče, nikoliv ale internetu (stroj může být připojen na intranet magistrátu).

Občané nebudou mít přístup na aplikaci `yadb` z internetu, musí přijít k fyzické elektronické desce, kde bude tato aplikace dostupná. Aplikace, která na této elektronické úřední desce běží je digitální úřední deska – má webová aplikace, oba termíny se však dají v kontextu této práce vzájemně nahrazovat. Z internetu mají občané přístup na ořezanou verzi mé aplikace na stránce magistrátu města Děčín (webovou digitální úřední desku). Pro správu elektronické úřední desky se k ní úředník připojí přes prohlížeč buď v rámci lokální sítě nebo přes internet (záleží na implementaci z 2.2).

2.1 Funkce elektronické úřední desky a požadavky na aplikaci

Hlavní funkcí úřední desky je umožnění komunikace mezi magistrátem města a občanem. Magistráty měst informují své občany skrze vyhlášky a různé jiné dokumenty. Okolo elektronické úřední desky existují dva typy uživatelů: občané města, kteří desku používají pro získávání informací a úředníci, kteří na desku vyvěšují dokumenty. Elektronická úřední deska usnadňuje oproti klasické přístup k těmto informacím z obou stran. Úředníkům i lidem šetří čas a při použití vzdáleného přístupu k úřední desce (přes internet) nemusí občan nikam chodit. Vyvěšování dokumentů je prováděno pomocí spisové služby.

Ovládání elektronické úřední desky by mělo být pro oba typy uživatelů co možná nejintuitivnější. V nejlepším případě by ani jedna strana neměla použít návod. Občané tak po několika kliknutích na dotykové obrazovce desky zjistí hledané informace. Na elektronické desce mají tu výhodu, že mohou dokumenty filtrovat, mohou vyhledávat pomocí dotazů nebo si dokumenty mohou stáhnout do svého zařízení.

Elektronická úřední deska funguje autonomně, není potřeba se o ní starat v nejlepším případě vůbec. Má webová aplikace je přímo napojená na spisovou službu města Děčín a na desce se automaticky aktualizují vyvěšené a svěšené dokumenty. Úřední deska také může obstarávat funkčnost interaktivního informačního centra nejen pro turisty. Otázkou zde je, zda někdo v době chytrých telefonů takovou funkcionalitu vůbec využije.

Pro případ, že by městu Děčín mé řešení nevyhovovalo, aplikaci lze rozšířit nebo předělat díky tomu, že je open–source a využívá pouze open–source frameworky a knihovny. Dají se tak například přidat rozšíření a z desky udělat informační kiosek.

Navrhuji pro funkci elektronické úřední desky použít webovou aplikaci typu MVC. Webové aplikace mají výhodu toho, že jsou relativně jednoduché na programování, mají velmi široké možnosti užití a funkcionality – existuje pro ně velké množství různých open–source knihoven. Dalším plusem je to, že HTML, CSS a JS tvoří dohromady velmi kvalitní balík jazyků pro tvorbu uživatelského rozhraní, které je v mé aplikaci podstatnou částí.

Alternativně by se digitální úřední deska dala naprogramovat například v C++ s použitím knihovny Qt. Otázkou je, zda by rychlost, kterou C++ přináší oproti

webové aplikaci opravdu odůvodnila ztrátu jednoduché tvorby uživatelského rozhraní za použití různých pluginů a knihoven, kterých ve svém programu využívám mnoho.

Moje webová aplikace `yadb` splňuje tyto požadavky:

- Dostupnost při výpadku internetu
- Autonomní funkcionalita
- Intuitivní dotykové uživatelské rozhraní
- Přístupnost pro invalidy a seniory
- Vzdálená správa
- Open–source
- Rozšiřitelnost
- Nenákladný provoz
- Statistika provozu

Dostupnost při výpadku internetu je důležitou vlastností elektronické úřední desky. Z kapitoly 1 je jasné, že tato vlastnost je nutností každé elektronické úřední desky a souvisí se zákonem, který udává, že deska musí být nepřetržitě veřejně přístupná.

Autonomní funkcionalita – je praktické, aby deska v daných intervalech aktualizovala svůj obsah sama od sebe. Město Děčín provozuje spisovou službu, která je mimo jiné i zdrojem stále zmiňovaného XML feedu s daty úřední desky. Tento XML feed slouží mé aplikaci jako hlavní zdroj informací, které deska zpracovává a poskytuje občanům.

Intuitivní dotykové ovládání – dá se očekávat, že desku budou využívat různé demografické skupiny občanů a je tedy na místě zajistit, aby bylo ovládání elektronické úřední desky co možná nejintuitivnější. V ideálním případě natolik, že nebude potřeba vůbec použít návod.

S tím souvisí vlastnost **přístupnost pro invalidy a seniory** – pro zpřístupnění používání desky pro úplně všechny je nutné myslet i na vozíčkáře, pro které může být nepraktické používat přibližně dvou metrovou úřední desku. Je potřeba softwarově zajistit režim, který jim práci s deskou usnadní. Starší občané naproti tomu mohou mít například horší zrak a je tudíž potřeba mít vše na desce dostatečně čitelné.

Vzdálená správa – ovládací počítač je uzavřený v kovovém krytu v elektronické úřední desce, která se může nacházet venku. Nepřipadá tak v úvahu, že by kvůli běžné správě desky bylo potřeba k ní fyzicky přijít a ovládat jí, natož pak přes dotykový panel. Pro úředníky a správce je v mém programu zavedeno správcovské rozhraní, na které se dá dostat přes internetový prohlížeč a lze přes něj spravovat různé funkce desky. Navrch běžné správy úřední desky se může správce také připojit

pomocí běžných protokolů pro vzdálené připojení, jakými jsou například `SSH` nebo `RDP`. Aplikace by však měla fungovat způsobem, že nebude nutné pro její správu používat nic jiného než webové správcovské rozhraní.

Open–source a rozšiřitelnost – open–sourcing aplikace pro veřejnou správu obecně přináší mnoho výhod. Jednou z hlavních je to, že je taková aplikace zcela zdarma a město se vyhne licenčním i pořizovacím nákladům. Dalšími bonusy oproti proprietárnímu kódu jsou možnosti objektivního posouzení veřejností. Open–sourcing také přináší rozšiřitelnost ve formě nejrůznějších patchů a oprav, například optimalizace rychlosti zpracování dat, úprava uživatelského rozhraní nebo třeba objevení bezpečnostních chyb a jejich oprava.

Nenákladný provoz – elektronická úřední deska běží v ideálním případě dvacet čtyři hodin denně. Je tedy vhodné myslet i na energetické nároky. Použitím jednodeskového počítače typu NVIDIA Jetson namísto klasického desktopu se sníží spotřeba elektřiny na zlomek. Dále je dobré implementovat způsob, který desku převede do režimu, ve kterém spotřebovává méně elektřiny než běžně – například ztlumení jasu obrazovky.

Statistika provozu – pro magistrát města může být důležitá informace o tom, kolik lidí jejich úřední desku vůbec používá. Dále může být důležité kontrolovat i systémové informace o provozu počítače v desce, jestli se například nepřehřívá.

2.2 Návrh serverové části

Přemýšlel jsem nad návrhem *sítě* úředních desek pro případ, že by jich město provozovalo hned několik na různých místech. Přišel jsem s třemi návrhy, každý má výhody i nevýhody.

- *Jedna deska jeden server* – na každé elektronické úřední desce běží instance `yadb`. Tato instance je napojena na spisovou službu města a v rámci lokální sítě úřadu jí mohou úředníci spravovat. Tento návrh je referenční a implementuje všechny požadavky na elektronickou desku z kapitoly 1. Řešení by bylo nevhodné, kdyby chtěl jeden úřad provozovat několik desek, bylo by pak nutné každou z nich konfigurovat a spravovat samostatně kvůli odděleným serverům.

- *Jeden úřad jeden server* – v síti každého úřadu by byl provozován server, na kterém by běželo API, které by obsluhovalo volání jednotlivých desek. Výhodou je zde společná konfigurace všech připojených desek zároveň – desky by se chovaly jako totožné kopie. Nevýhodou je nutnost provozu dalšího serveru. Má aplikace by se po několika změnách dala předělat na toto řešení – webová aplikace může buď běžet lokálně na desce nebo na dedikovaném serveru. V současnou chvíli je toto řešení (bez změn zdrojového kódu) možno zprovoznit tak, že se na deskách bude skrze prohlížeč zobrazovat jedna instance `yadb` běžící na serveru, ale úředníci přijdou o možnost sledovat stav ovládacího počítače desky (což není zvláště důležité).

- *Jedno město jeden server* – toto řešení je velmi podobné tomu předchozímu s tím rozdílem, že server by neběžel lokálně ale někde na internetu, šlo by využít i možnosti pořízení VPS od třetí strany. Na tento server by mohly být napojeny desky v celém městě. Nevýhodou je nedostupnost při výpadku internetu, ale dlouhodobější výpadky internetu jsou v dnešní době spíše vzácností.

Město Děčín v tuhle chvíli provozuje tři různé elektronické úřední desky s tím, že dvě se dají nahradit jednou a tím pádem by na každé straně města byla jen jedna elektronická úřední deska. Má aplikace se proto drží návrhu *jedna deska jeden server*.

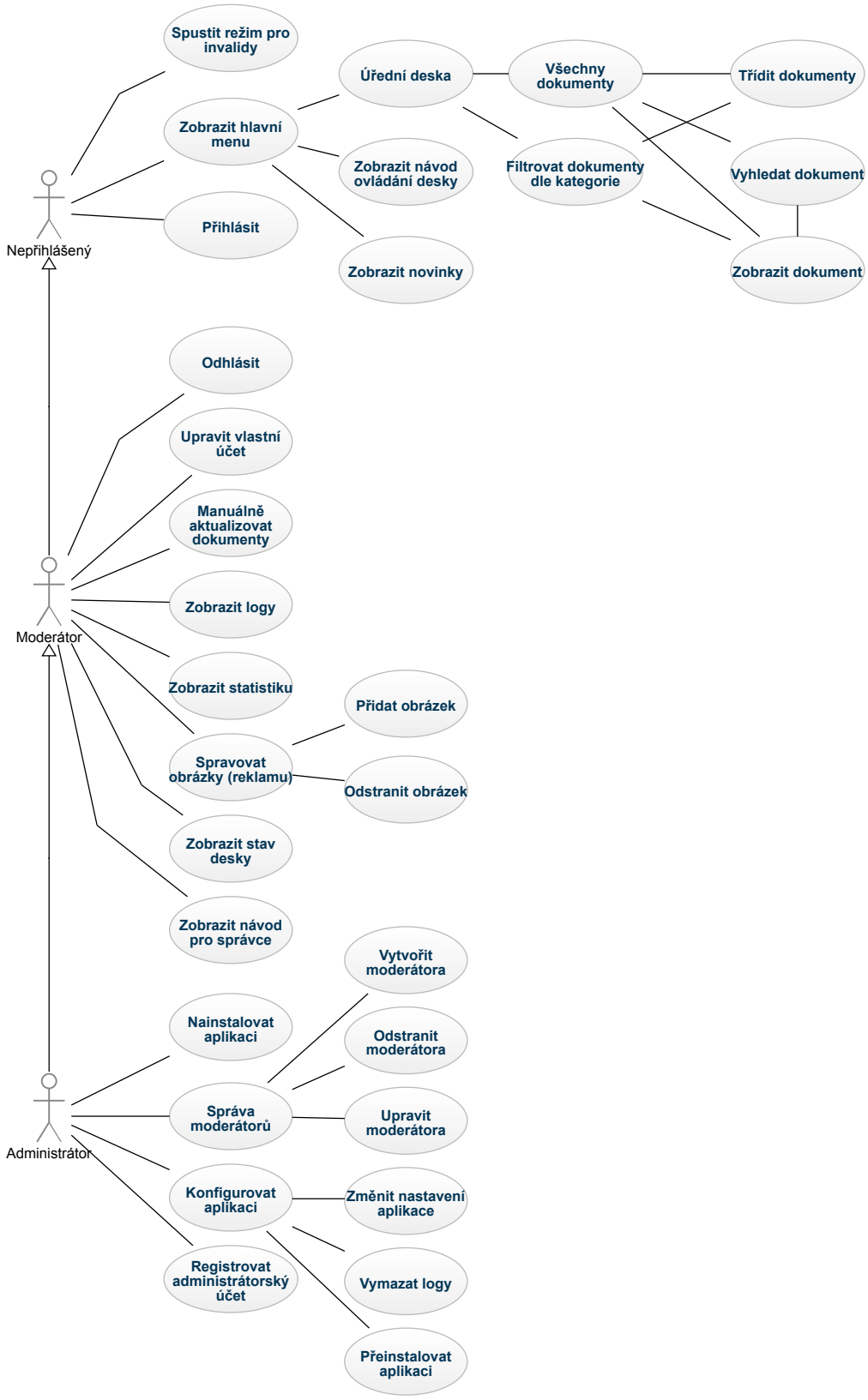
2.2.1 Uživatelé a jejich práva

V aplikaci *yadb* je několik typů uživatelů, každý z nich má různá práva a možnosti, jak aplikaci používat. Možnosti použití desky uživateli jsou podrobněji znázorněny v diagramu užití (obr. 2.1).

Základním uživatelem je nepřihlášený (občan), tento uživatel není v mé aplikaci nijak specifikován (nemá reprezentaci v databázi). Takový je výchozí stav mé aplikace. Občan by neměl mít možnost se jakkoliv dostat do správcovské části aplikace. Tomu zabraňuji pomocí přístupových práv a faktu, že se z prostředí dotykového panelu nemá jak *prokliknout* do správcovské části. Nepřihlášený uživatel interaguje s dotykovým panelem elektronické úřední desky a *konzumuje* její obsah. Může listovat, filtrovat a vyhledávat dokumenty na úřední desce. Také má možnost číst si novinky města a v budoucnu by mohl desku využívat jako obecné informační centrum.

V aplikaci jsou dva správcovské typy uživatelských účtů. Jedním je *moderátor* obsahu (úředník) a druhým je *administrátor*. Moderátor má menší práva než administrátor. Jeho hlavním úkolem je kontrola stavu desky a správa obsahu (hlavně reklamy).

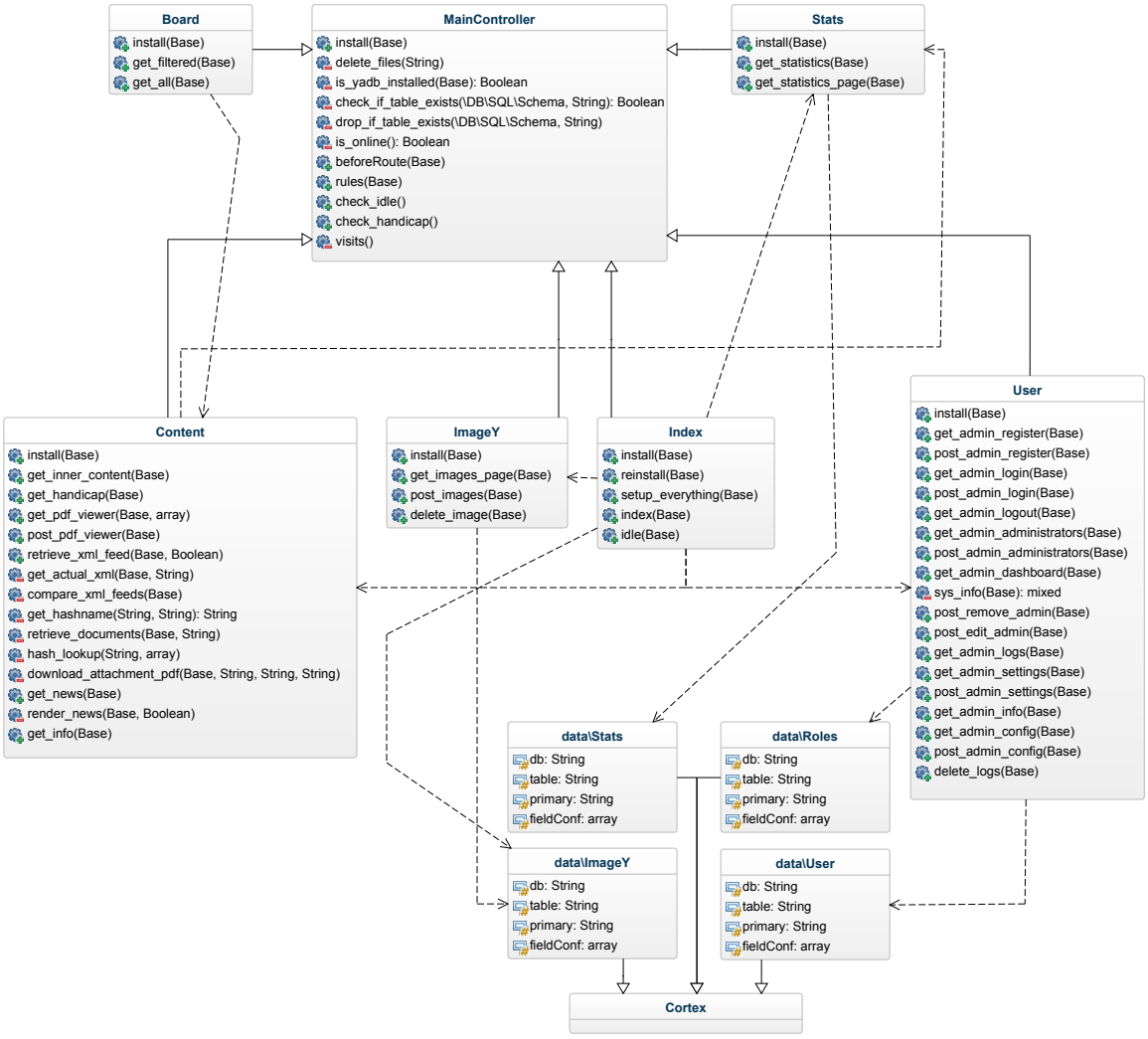
Administrátor je nadstavbou nad moderátorem a může zasahovat do funkcionality desky (nesprávným zásahem jí může vyřadit z provozu). Tento uživatelský účet by měl používat jeden hlavní správce desky, který je počítačově gramotný a je si vědom toho, čeho může změnou konfigurace desky dosáhnout. Kromě vytváření a správy uživatelských účtů moderátorů má tedy administrátor ještě navíc přístup ke konfiguraci desky.



Obrázek 2.1: Diagram užití

2.2.2 Třídy a diagram tříd

Tato podkapitola částečně přesahuje do kapitoly 3. V mé aplikaci se vyskytují dva typy tříd. Prvním typem tříd jsou *kontroléry*, které ovládají běh aplikace, vykreslování šablon a podobně (podrobněji v kapitole 3). Druhým typem jsou třídy *modelové*, které implementují databázové ORM modely (mají stejný název jako kontroléry, ale začínají označením `data`. V diagramu tříd lze vidět, že všechny kontroléry dědí z předka `MainController`. Třídy jsou mezi sebou závislé a to ve smyslu, že v rámci tříd existují instance sourozeneckých tříd. Podobně jsou kontroléry závislé na *modelových* třídách, tyto jsou používané pro komunikaci s databází. Zmíněná závislost je v diagramu tříd (obr. 2.2) zakreslena přerušovanou čárou.



Obrázek 2.2: Diagram tříd

2.2.3 Kontroléry

Tyto třídy odpovídají vrstvě `controller` v architektuře MVC. Zajišťují interakci s databází skrze datové ORM modely, mění informace v databázi a vykreslují příslušné šablony uživatelského rozhraní. Je v nich implementovaná logika, které zajišťuje aktuální obsah na desce a automatizovaný chod desky.

Hlavní třídou, ze které dědí všechny ostatní je `MainController`. V této třídě se nachází užitečné metody pro chod webové aplikace. Jsou zde kontrolovány přístupová práva uživatelů, kontrola a změna stavu programu a jiné pomocné metody jako kontrola připojení aplikace k internetu.

Třída `Index` zajišťuje instalaci a přeinstalaci celého programu, přechod do klidového režimu a vykreslení hlavní (vstupní) stránky.

Jednoduchá třída `Board` zajišťuje sestavení a vykreslení tabulky, která slouží jako úřední deska. Zpracovává aktuální XML soubor ze spisové služby (který je uložený lokálně) a dle něj sestavuje již zmíněnou HTML tabulku. V této tabulce jsou vyvěšeny všechny vyhlášky a informace na desce včetně příslušných příloh v podobě PDF dokumentů.

Další z jednodušších tříd je třída `Stats`, ta zajišťuje vykreslování statistik návštěvnosti pro moderátory nebo administrátory.

`ImageY` je třída, která obstarává rozhraní pro nahrávání a mazání obrázků. Tyto obrázky mohou být propagační nebo reklamní materiál, který město Děčín může na desce zobrazovat. Obrázky rotují na obrazovce desky, když je v klidovém režimu a nikdo jí zrovna nepoužívá.

Srdcem backendu úřední desky je třída `Content`. Ta, jak už je z názvu zřejmé, implementuje zobrazování a správu obsahu na desce. Je zde implementovaný hlavní algoritmus, který zajišťuje aktualizaci XML souboru (ze spisové služby) a stahování čerstvých PDF příloh. V kapitole 3 je podrobněji popsáno, jakým způsobem se dokumenty a XML soubor na desce aktualizují. Mezitím co ostatní třídy *slouží* především moderátorům desky, tato třída je centrální pro občana. Mimo další je zde implementováno vykreslování novinek, které jsou stahovány z RSS feedu města Děčín.

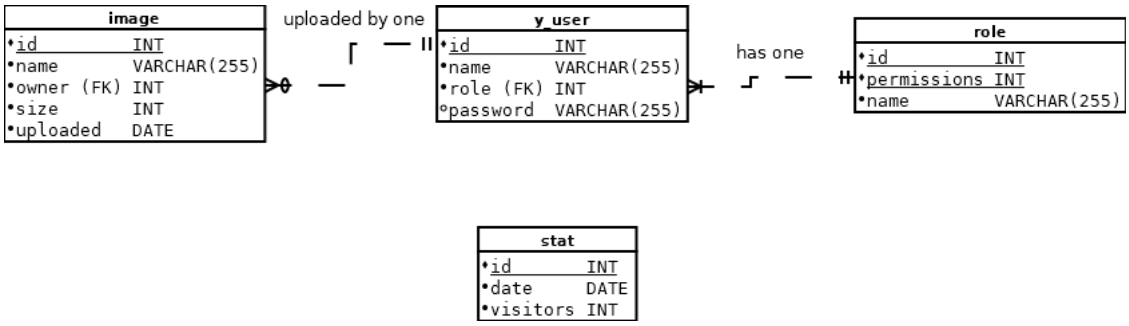
Největší třídou je třída `User`. Název třídy je trochu zavádějící, jelikož zároveň s metodami pro správu uživatelských účtů implementuje i metody pro vykreslování správcovské části webové aplikace. Implementuje také metody, díky kterým se dá monitorovat stav desky, měnit konfigurace apod. Naproti třídě `Content` je tato třída důležitá pro administrátory a moderátory.

2.2.4 ORM modely

Všechny *modelové* třídy dědí z rodičovské třídy `Cortex` (ORM plugin do Fat-Free Framework). Třídy jsou mapovány na stejnojmenné tabulky v databázi. V těchto třídách jsou stanovena integritní omezení a kardinalita mezi tabulkami databáze. Modelové třídy obstarávají funkcionalitu vrstvy `model` v architektuře MVC.

2.3 Návrh databáze

Pro funkci mé elektronické úřední desky je zapotřebí databáze. Databáze je znázorněna pomocí ERA diagramu na obr. 2.3. Každá tabulka odpovídá některé z *modelových* ORM tříd.



Obrázek 2.3: ERA diagram

Databáze je velmi jednoduchá a to proto, že ji využívám především pro správu uživatelů a jejich práv. Hlavní data na digitální úřední desce (dokumenty, vyhlášky atd.) jsou zajišťována pomocí XML souboru pocházejícího ze spisové služby. Zde jsou vlastnosti a integritní omezení databáze:

- **y_user** – uživatelský účet správce.
  - id – primární klíč, automaticky inkrementovaný.
  - name – unikátní přihlašovací jméno správcovského účtu.
  - role – práva správcovského účtu, cizí klíč do tabulky role, uživatel má jednu roli.
  - password – heslo od uživatelského účtu správce.
- **role** – role správce (přístupová práva), jednu roli má 1 až N uživatelů.
  - id – primární klíč, automaticky inkrementovaný.
  - permissions – primární klíč, číselné vyjádření práv.
  - name – unikátní slovní název role.
- **image** – reklamní obrázky
  - id – primární klíč, automaticky inkrementovaný.
  - name – název obrázku, unikátní, totožné se jménem souboru na disku.
  - owner – cizí klíč do tabulky y_user, 0 až N obrázků může být vlastněno jedním uživatelem.
  - size – velikost obrázku v `Kb`.
  - uploaded – datum nahrání obrázku.

- **stat** – ukládání návštěvnosti, tabulka bez rodiče.
  - id – primární klíč, automaticky inkrementovaný.
  - date – datum záznamu návštěvnosti.
  - visitors – počet návštěvníků v daný den.

2.4 Návrh uživatelského rozhraní

Jak jsem již zmiňoval, tato implementace digitální úřední desky má dvojí charakter uživatelského rozhraní (zkráceně UI). Jedna strana UI je pro zobrazování informací a úřední desky pro občana na dotykovém panelu EÚD. Na druhé straně proti tomu implementuji UI pro konfiguraci a kontrolu stavu desky pro úředníky (moderátory a administrátory). Obě UI mají jednoduchý intuitivní vzhled, který se drží grafického návrhu hlavní webové stránky MM Děčín (stejný font, barevné schéma). Vzniká tak celistvost vzhledů napříč digitálními prezentacemi města Děčín. Jako hlavní font používám ve své aplikaci `Fira Sans`, o kterém se zmiňuje *manuál vizuální identity* města Děčín, který mi byl zaslán panem Mgr. Žaloudkem z MM Děčín. Město Děčín také vyhovělo mé žádosti použít městský znak v mém programu, proto se v uživatelském rozhraní několikrát objevuje. Mezi oběma UI však existují rozdíly, ty spočívají především v tom, že občanské UI je určené pro vertikální dotykový panel.

2.4.1 Občanské UI

Uživatelské rozhraní pro občany je určené pro běh na širokoúhlé dotykové obrazovce, proto se musí držet určitých vzhledových pravidel. Písmo musí být dostatečně čitelné pro všechny demografické skupiny, tlačítka musí být dostatečně veliká, aby je bylo možné jednoduše stisknout. Občanské UI má také intuitivní dopředný charakter, aby občan nemusel dlouho tápat nad tím, jakým způsobem desku vůbec použít. Na obrázcích B.1,B.2 a B.3 v příloze B lze vidět obyčejný postup, jakým se občan dostane ve třech kliknutí k hledanému dokumentu. Lze si také všimnout, že je všechno situované na výšku, to mimo jiné zpřehlední čtení PDF dokumentů. Na obrázku také můžeme vidět dotykovou klávesnici, která je podstatná pro vyhledávání specifických dokumentů.

Aktivovaný režim pro invalidy lze vidět na obrázku B.4 v příloze B. Podobným způsobem je *režim invalida* implementován na elektronické úřední desce od firmy Ki-Wi Digital s. r. o. Je to velmi jednoduchý způsob, jakým vyřešit přístup pro vozíčkáře. Ve své podstatě se všechen obsah posune o polovinu obrazovky níž a vytvoří se tak větší místo pro *scrollování* i ze sedící pozice. Má implementace posouvá v režimu invalida ovládací prvky pro procházení tabulkou dokumentů níž. Po kliknutí na ikonku vyhledávání se otevře dotyková klávesnice a automaticky se posune doprostřed stránky.

2.4.2 Administrátorské UI

Administrátorské rozhraní nemá žádné speciální požadavky, je to klasická webová stránka pro zobrazení na desktopovém počítači nebo notebooku. Na obrázku B.5 v příloze B představuji ukázku hlavní stránky správcovského rozhraní. Jako u občanského UI jsem se snažil o co největší intuitivnost a jednoduchost a tak se ke každé podstatné funkci úřední desky dá dostat v několika kliknutích. Moderátoři aplikace zde mohou monitorovat stav desky, nahrávat na ní reklamu apod.

Administrátorovi se pak odemyká více možností především konfigurace desky a správa uživatelských účtů. Pro případ, že bude kdokoliv v UI ztracen, jsem připravil návod na ovládání úřední desky jak pro občana tak i pro úředníky a oba ho mohou nalézt ve svém UI.

2.5 Návrh hardwarové části

V kapitole 1 jsem se již zmiňoval o tom, jaké vlastnosti mají elektronické úřední desky tuzemských výrobců. V ohledu na hardwarovou implementaci to jsou:

- Odolnost proti vandalismu
- Odolnost proti povětrnostním vlivům (proti vodě a teplotním změnám)
- Dotyková obrazovka
- Kovový kryt a tvrzené sklo
- Širokoúhlá obrazovka
- Ovládací počítač

Na návrhu českých výrobců není moc co měnit, snad kromě ovládacího počítače. Pro běh mé webové aplikace je dostačující jednodeskový počítač typu Raspberry Pi nebo NVIDIA Jetson Nano. NVIDIA Jetson Nano mi byl poskytnut mým vedoucím práce a na tento počítač jsem první verzi svého programu nainstaloval (více v kapitole 4). Výhodou Jetsonu je jeho cena, která se pohybuje okolo dvou až tří tisíc korun. Za tuto cenu dostáváme kompletní počítač, který dokáže provozovat Linuxovou distribuci (defaultně Ubuntu). Pro mou aplikaci je dostatečné, pokud ovládací počítač rozběhne Apache a MySQL server plus webový prohlížeč. SSD disky a Intel procesory jsou pro aplikaci tohoto typu zbytečným luxusem a jejich odebráním nepřijdeme o potřebný výkon. Dalším bonusem jednodeskového počítače je malá energetická náročnost.

Tato bakalářská práce se primárně zabývá návrhem a implementací softwarové části elektronické úřední desky, ale pro finální produkt, který bude sloužit občanům v budoucnu je potřeba myslet i na to, že takové elektronické zařízení je potřeba sestavit. Zde je nevýhoda toho, že si konstrukci desky musí město zařídit svépomocí. Musí se navrhnout rozměry kovové skříně, elektrické rozvody, termoregulace, odolnost proti

vodě apod. To vše se pak musí smontovat (svařit, přišroubovat) dohromady a zapojit před budovou magistrátu.

Kapitola 3

Implementace webové aplikace

V této kapitole objasním, jakým způsobem jsem implementoval webovou aplikaci, která splňuje požadavky z podkapitoly 2.1. Tato aplikace zastává funkci digitální úřední desky, která po zkompletování s *dotykovou obrazovkou* (blíže v kapitole 4) tvoří základ pro elektronickou úřední desku. Vysvětlím, jakým způsobem webová aplikace funguje. Nejprve však představím použité technologie, zaměřím se při tom blíže na Fat-Free Framework. Následně rozeberu implementaci obslužných metod v kontrolérech a implementaci databáze a datových modelů. Na to naváži implementací uživatelského rozhraní a detailněji rozpracuji i algoritmus, kterým deska získává aktuální informace a PDF dokumenty.

3.1 Použité technologie

Pro implementaci aplikace `yadb` jsem zvolil jako základ programovací jazyky PHP 7, HTML 5, CSS 3 a Javascript. Tyto jazyky dohromady tvoří kvalitní základnu pro uživatelsky orientované aplikace a existuje pro ně obří množství open–source knihoven a frameworků. Jsou to velmi univerzální jazyky, které lze použít v širokém spektru aplikací. Má aplikace je uživatelsky orientovaná a tudíž se čtveřice nejčastějších webových jazyků prezentovala jako vhodná volba. Ze zkušenosti vím, že se v nich dá tvořit moderní uživatelské rozhraní, které je responzivní, reaktivní a dynamické.

Aplikace není nijak výpočetně náročná, proto jsem nezvažoval implementaci v C++ apod. Při implementaci v C++ bych se připravil o benefit využití mnoha javascriptových knihoven jako je JQuery nebo DataTables. Čímž bych se zároveň připravil i o čas, jelikož bych musel implementovat části, jako je třeba dotyková klávesnice, sám.

Rozděluji použité technologie na *frontend* (uživatelské rozhraní) a *backend* (serverovou část aplikace). Při implementaci webové aplikace jsem primárně vycházel ze zdrojů (dokumentací na webových stránkách), na které je odkazováno v tabulce 3.1 u jednotlivých knihoven a frameworků. Dále jsem čerpal z PHP dokumentace[19], W3Schools[24] a StackOverflow[23]. O různých knihovnách a pluginech se budu zmiňovat v průběhu popisu implementace mé aplikace, vyzdvihnul jsem pouze Fat-Free

Framework, na kterém celá aplikace stojí. V tabulce 3.1 prezentuji výčet všech frameworků, pluginů a knihoven, které jsem pro sestavení výsledné aplikace použil.

Název	Použití	Licence
Frontend		
Animate.css[1]	Animace	MIT
blockUI[15]	Blokování UI	MIT
Bootstrap[3]	CSS, responzivita	MIT
Bootstrap Icons[3]	Ikony	MIT
Chart.js[4]	Graf návštěvnosti	MIT
DataTables[6]	Úřední deska	MIT
Font Awesome[8]	Ikony	CC BY 4.0, MIT
GoogleFonts[13]	Písma	Různé (open–source)
JQuery[14]	Uživatelská interakce, AJAX, pluginy	MIT
MD Bootstrap[16]	Material design pro Bootstrap	MIT
PDF.js[18]	Prohlížeč PDF dokumentů	Apache License 2.0
Popper.js[3]	Popups	MIT
qrcode.js[21]	QR kód	MIT
simple-keyboard[22]	Dotyková klávesnice	MIT
Backend		
Apache[2]	HTTP Server	Apache Licence
Fat-Free Framework (F3)[7]	Micro–framework, základ backendu	GPL-3.0
F3 Access[9]	Uživatelská práva	GPL-3.0
F3 Schema Builder[12]	Nastavení databáze	GPL-3.0
F3 Cortex[10]	ORM modely	GPL-3.0
F3 Flash[11]	Varování, oznámení	GPL-3.0
MySQL[17]	Databáze	GPL-2.0

Tabulka 3.1: Použité technologie

3.2 Fat-Free Framework

Fat-Free Framework, zkráceně F3, je open–source PHP micro–framework, který je zaměřený na minimalismus a jednoduché použití přičemž obsahuje důležité funkce, které se dají standardně očekávat u kvalitního PHP frameworku. Různé nástroje obsažené ve frameworku dovolují tvořit komplexní aplikace.[7] Framework je rozšiřitelný množstvím pluginů, kterých ve své práci využívám hned několik. Fat-Free Framework jsem si vybral z důvodu, že jsem s ním dobře seznámen a použil jsem ho již dříve pro několik projektů. Jeho *lightweight* filozofie také vyhovuje cílové platformě mé aplikace, kterou je jednodeskový počítač.

Mimo jiné obsahuje F3 vlastní šablonovací engine, který má jednoduchou syntaxi a dovoluje tvořit obsahově dynamické stránky. Také směrovací metody velmi usnadňují práci při vývoji v tomto frameworku. F3 podporuje různé typy databází a při použití patřičných pluginů i ORM mapování. Není povinností využívat MVC architekturu, ale v základním nastavení je jí framework nakloněn. Této architektury jsem se držel při vývoji i já.

Nevýhod, se kterými jsem se při vývoji v tomto frameworku setkal, je několik. Framework je poměrně neznámý a nemá tudíž široké využití jako například Laravel nebo Nette. S tím přichází i nedostatek návodů a diskuzí okolo chyb a bugů na různých fórech. Na větších frameworcích také pracuje více lidí a jejich vývoj je tím pádem rychlejší a reaguje na objevené nedostatky lépe. Několikrát při vývoji se mi stalo, že očekávaná funkcionalita frameworku nefungovala, jak by měla a musel jsem jí obejít. Framework bych pro větší projekt nepoužil, ale pro potřeby této práce je dostatečný.

3.2.1 Soubory a složky

Všechny frameworky, pluginy a knihovny jsou staženy lokálně pro zachování plné funkčnosti elektronické úřední desky v režimu bez internetu. Vyhnul jsem se použití CDN. Držel jsem se základního uspořádání souborů a složek v F3. Zde pro představu uvádím některé soubory a souborovou hierarchii celého projektu.

```
yadb-release ..................................... Kořenový adresář.
├── app
│   ├── yadb ..................................... Třídy s kontroléry.
│   │   ├── data ................................. ORM modely.
│   │   │   ├── Board.php
│   │   │   ⋮
│   │   │   └── User.php
│   │   ├── Board.php
│   │   ⋮
│   │   └── User.php
│   ├── config ................................... Konfigurační soubory.
│   ├── lib ...................... Adresář s Fat-Free Frameworkem.
│   ├── logs ..................................... Logy.
│   ├── scripts .................................. Bash skripty.
│   ├── tmp ...................................... Dočasné soubory.
│   ├── ui ....................................... Uživatelské rozhraní.
│   │   ├── css
│   │   ├── icons
│   │   ├── img
│   │   ├── js ................................ JavaScriptové knihovny.
│   │   ├── pdf ....................... PDF dokumenty z úřední desky.
│   │   ├── pdfjs ..................... Knihovna pro zobrazení PDF.
│   │   ├── simple_keyboard ................... Dotyková klávesnice.
│   │   ├── xml ................................ XML soubory.
│   │   ├── admin_dashboard.html ..... HTML šablony.
│   │   ⋮
│   │   └── scripts.html
│   └── uploads ..................... Soubory nahrané uživatelem.
├── .htaccess
├── composer.json
├── config.ini ..................... Konfigurace proměnných F3.
├── db.ini ....................... Konfigurace databáze.
├── index.php ..................... Vstupní stránka F3.
├── installation_manual.txt ..... Instalační manuál v textové podobě.
├── README.md
└── routes.ini ................. Definice routes neboli přesměrování.
```

3.2.2 Přesměrování a práva

V F3 se dá v souboru `routes.ini` vypsat a pojmenovat URL adresy pro přesměrování. Syntaxe je následující:

```
1 GET /admin = \yadb\User->get_admin_login
```

Kde místo `GET` může být i jiné z dotazovacích HTTP metod. `/admin` je definice relativní URL cesty a `\yadb\User->get_admin_login` definuje, jaká obslužná metoda kterého kontroléru se má zavolat. V programu `yadb` používám přes třicet takových *rout*.

Speciálně cesty pro správcovskou část aplikace mají předponu `/admin/`, kterou mohou navštívit pouze přihlášení správci. Nepřihlášený uživatel se tak do této části nedostane. Zda má daný uživatel právo navštívit danou cestu řeší funkce `rules`, která je implementovaná v `MainController.php` a využívá pro svou funkci knihovnu F3 Access. Před každým přesměrováním je tato funkce volána a je určeno, zda má uživatel příslušná práva. Některé cesty navíc nesmí navštívit moderátorské účty, ale pouze účty administrátorské (jedná se o cesty spojené s konfigurací a přeinstalací programu).

3.3 Kontroléry

V této části popíšu metody kontrolérových tříd a jejich roli v provozu úřední desky. Podrobněji se zaměřím na hlavní aktualizační algoritmus, pomocí kterého získávám aktuální úřední PDF dokumenty ze serverů města. Většina veřejných metod v kontrolérech slouží pro zpracování *rout*. Privátní jsou většinou pomocné metody, které daný kontrolér využívá pro vyprodukování výsledné šablony. Všechny metody jsou zdokumentovány ve zdrojovém kódu pomocí *tagů* definovaných na stránce phpDocumentor.[20]

3.3.1 MainController.php

Rodičovský kontrolér, ze kterého dědí všechny ostatní kontroléry, obsahuje pomocné metody, které jsou využívány napříč ostatními kontroléry.

`install`: metoda, kterou dědí všechny ostatní kontroléry, každý jí implementuje jinak. V tuto chvíli je prázdná, protože při instalaci není potřeba žádné interakce s tímto kontrolérem. Do budoucna by jí mohlo být potřeba.

`delete_files`: metoda, která rekurzivně vymaže složku a soubory v ní. Toho je využito například při přeinstalaci celé aplikace.

`drop_if_table_exists`: smazání databázové tabulky, pokud již existuje. Potřeba při přeinstalaci aplikace.

`check_if_table_exists`: metoda, která kontroluje, zda specifikovaná tabulka existuje.

`is_online`: zjištění, zda je aplikace připojená k internetu nebo ke specifikovanému serveru. V případě, že je aplikace bez internetu nebo bez spojení se zdrojovým serverem s PDF dokumenty, neproběhne aktualizace a použije se lokální *neaktualizovaná* verze PDF dokumentů.

`beforeRoute`: implementace funkce z F3. Tato metoda se volá pokaždé, když je uživatel přesměrován na některou z cest specifikovaných v `routes.ini`. V této metodě volám metody, které kontrolují přístupová práva a zda je aplikace v nečinném režimu a nebo v režimu pro invalidy.

`rules`: pomocí pluginu F3 Acces kontroluji, zda má uživatel dostatečná přístupová práva pro přístup na specifikovanou cestu. V základu jsou všechny cesty povoleny. Pro nepřihlášené uživatele zakazuji všechny cesty s předponou `/admin/*`. Navíc zde ručně vybírám některé cesty, do kterých má přístup pouze administrátor ale ne moderátor.

`is_yadb_installed`: kontroluji, zda byla na daném stroji aplikace již nainstalována pomocí cesty `/install`. Před tím, než je aplikace schopná provozu je nejprve nutno zavolat cestu `/install`, která nastaví databázi, přístupová práva, konfigurační soubory apod.

`check_idle`: kontrola, zda je aplikace v nečinném režimu (*idle*).

`check_handicap`: kontroluje, zda je aplikace v režimu invalida. Díky této metodě se následně změní uživatelské rozhraní, aby vyhovovalo přístupu z invalidního vozíku.

`visits`: pomocí této metody zvyšuji návštěvnost. Je volána pokaždé, kdy je z nečinného režimu aplikace přesměrována na hlavní menu. Režim nečinnosti se spustí po patnácti minutách, kdy na aplikaci nikdo neklikne. Předpokládám tedy, že když někdo desku probudí z nečinnosti, je to nový návštěvník.

3.3.2 Index.php

Tento kontrolér je vstupní bránou do aplikace.

`install`: volaná z metody `setup_everything`. Tato metoda volá metody `install` jednotlivých tříd, které je potřeba *nainstalovat*. Těmi jsou v tuto chvíli třídy `Stats, ImageY, User`. Slovem instalovat je zde myšleno nastavit správně databázové tabulky pro funkci datových ORM modelů.

`reinstall`: v případě, že je již aplikace `yadb` nainstalována, může ji přeinstalovat administrátor pomocí `/reinstall` z administrátorského menu konfigurace. Tato metoda vymaže dynamické složky (/uploads, /ui/pdf atd.). Smaže soubor /config/installed.txt, který dokazuje, že je aplikace již nainstalována. Zapíše do logu, že byla aplikace přeinstalována a zavolá metodu `setup_everything`.

`setup_everything`: důležitá metoda, která se volá pomocí adresy `/install`. Zkontroluje, zda je aplikace nainstalována, pokud ano, neudělá nic. Pokud je aplikace

nenainstalovaná, vytvoří explicitně složky a cesty a dá jim patřičná práva (kvůli Linuxu a uživateli www-data). Vytvoří konfigurační soubory se základními hodnotami. Nakonec zavolá aktualizaci dat na desce (XML souboru a PDF dokumentů).

`index`: metoda volaná relativní cestou `/`. Nastaví obsah stránky na pohyblivé obrázky (reklama, nečinný stav) a vykreslí tuto stránku.

`idle`: metoda která uvede aplikaci do nečinného stavu. Volána javascriptovým skriptem na cestě `/idle`, který jí volá každých patnáct minut. Vymaže proměnnou `SESSION`.

3.3.3 User.php

Obsahově největší kontrolér. Obstarává veškerou funkcionalitu, která je potřeba pro správcovskou stranu elektronické úřední desky. Obsahuje celkem devatenáct metod na pět seti řádcích, proto jejich popis zkrátím. Jsou to metody pro vykreslování všech šablon správcovského rozhraní. Dále zajišťuje přihlašování, odhlašování, registraci, mazání a nastavení uživatelských účtů. Pro administrátorský účet navíc přináší možnost konfigurovat aplikaci (přeinstalace, změna obsahu konfiguračních souborů). Všechny URL cesty jsou v souboru `routes.ini` začínající předponou `/admin/` a jsou podobné názvům metod.

`install`: nastavení databázových tabulek `y_user` a `role` za využití F3 Schema Builder. Nastaví sloupce tabulky podle specifikace v ORM třídách `data\User.php` a `data\Roles.php`. Navíc vytvoří základní role Admin a Moderator a nastaví jim přístupovou úroveň.

`get_admin_register`: vykreslí šablonu s registračním formulářem pro administrátorský účet.

`post_admin_register`: zpracuje registrační formulář, ověří uživatelské vstupy (shodu hesla se jménem, obsazenost jména apod.). Pro registraci administrátorského účtu je potřeba přístup k souboru /config/secret.ini, který je vygenerován při instalaci aplikace.

`get_admin_login`: vykreslí šablonu s přihlašovacím formulářem do správcovské části.

`post_admin_login`: zpracuje přihlašovací formulář, ověří uživatelské vstupy (shodu haše hesla s databází apod.). Přihlásí uživatele, uloží jeho ID, jméno a roli do `SESSION`.

`get_admin_logout`: odhlásí správce. Vymaže ho ze `SESSION`.

`get_admin_administrators`: vykreslí šablonu s tabulkou správců. Moderátor zde může vidět všechny správce. Administrátor je může upravovat a mazat, nebo vytvářet nové moderátorské účty.

`post_admin_administrators`: zpracovává formulář pro vytváření nových moderátorských (úřednických) účtů administrátorem. Vstupy jsou ošetřeny.

`get_admin_dashboard`: vykreslí hlavní stránku správcovského uživatelského rozhraní. Zde lze monitorovat současný stav jednodeskového počítače (teplota, zatížení procesoru apod.).

`sys_info`: pomocná funkce pro dekódování JSON souboru do kterého v intervalech (cronem) ukládá jednodeskový počítač informace o svém stavu. Script, který cron volá je v souboru `scripts/system_info.sh`. Je to BASH skript a pro různé počítače nemusí fungovat stejně.

`post_remove_admin`: zpracovává smazání správce hlavním administrátorem.

`post_edit_admin`: zpracovává formulář pro úpravu účtu moderátora hlavním administrátorem.

`get_admin_logs`: vykreslí stránku, na které může správce procházet logy. V mé aplikaci jsou tři typy logů: `user.log` pro logování aktivity správců, `error.log` pro logování chyb a `system.log` pro aplikační hlášení. Do logů je zapisováno ve spoustě funkcí napříč kontroléry.

`get_admin_settings`: vykreslí stránku s formulářem pro úpravu vlastního (přihlášeného) účtu.

`post_admin_settings`: zpracovává formulář pro úpravu vlastního účtu.

`get_admin_info`: vykreslí stránku s uživatelskou příručkou pro správce.

`get_admin_config`: vykreslí stránku s formulářem pro konfiguraci nastavení aplikace. Tuto funkci může volat jen hlavní administrátor (přes cestu `/admin/config`). V tuto chvíli zde lze nastavit URL adresu, ze které se má stahovat XML soubor. Dále se tu nastavuje URL adresa, na které se nachází PDF dokumenty a RSS adresa s novinkami. Prakticky se zde dají nastavit i intervaly, po kterých by se měla deska aktualizovat a restartovat. Nepodařilo se mi zprovoznit plugin F3 Cron, tudíž jsou tyto intervaly nastaveny v crontabu `staticky` (více v kapitole 4). Dají se změnit, ale né skrze webové rozhraní (v nynější verzi programu).

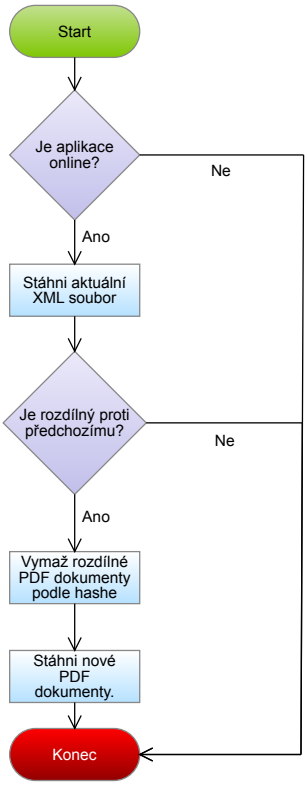
`post_admin_config`: zpracuje úpravu konfiguračních souborů a intervalů.

`delete_logs`: vymaže logy. Voláno z rozhraní konfigurace. Logy nejsou vymazány při přeinstalaci programu, je potřeba je vymazat manuálně.

3.3.4 Content.php

Druhá největší třída má za hlavní funkci obstarat aktuální XML soubor a stažení PDF dokumentů. Toho dosahuje v několika metodách, které implementují algoritmus, ten je zjednodušeně zobrazen na obr. 3.1. Algoritmus je navržen tak, aby každá aktualizace úřední desky trvala krátce a aby se stahovaly pouze nové nebo změněné PDF dokumenty. Naproti tomu stojí přístup *smazat vše a stáhnout vše znovu*, při němž zabrala aktualizace několik minut. Dále je algoritmus navržen tak, aby šetřil SD kartu na jednodeskovém počítači. Kdyby se totiž při každé aktualizaci stahovalo vše znova, byla by SD karta rychleji opotřebovaná. To není vhodné z hlediska životnosti jednodeskového počítače zabudovaného v elektronické úřední desce. Třída

také získává a vykresluje novinky z městského RSS (pouze v *online* režimu) a obecně obstarává zobrazování obsahu pro uživatele (občana).



Obrázek 3.1: Algoritmus pro zpracování dat a dokumentů

XML soubor stahovaný ze spisové služby města Děčín obsahuje stovky položek, které reprezentují jednotlivá oznámení vyvěšená na úřední desku. Takto je reprezentována například veřejná vyhláška:

```
<item>
    <TYP>Verejne vyhlasky</TYP>
    <VYVESENO>2021-04-29</VYVESENO>
    <STAZENO>2021-05-17</STAZENO>
    <POPIS>Pisemnost - Renata Fujerikova, vice viz priloha.</POPIS>
    <ZDROJ>Financni urad pro Ustecky kraj</ZDROJ>
    <PRILOHY>
      <item>
        <ID>U1829856</ID>
        <NAME>21690.pdf</NAME>
        <SIZE>251290</SIZE>
        <DATETIME>2021-04-29 12:05:18</DATETIME>
        <DESCRIPTION></DESCRIPTION>
      </item>
    </PRILOHY>
</item>
```

Metody kontroléru `Content.php`, v nichž je algoritmus implementován jsou tyto:

`install`: v tuto chvíli prázdná metoda, kontrolér pro svou funkci nepotřebuje interakci s databází, tudíž nepotřebuje *nainstalovat* žádné tabulky.

`get_inner_content`: voláno URL cestou `/inner_content` kliknutím kamkoliv na obrazovku v klidovém režimu. Vykreslí šablonu hlavního menu, ze kterého může uživatel (občan) dále pokračovat v používání desky. Dále přepne desku do aktivního režimu a zavolá metodu `visits()`.

`get_handicap`: voláno pomocí `/handicap` z tlačítka ze spodní dotykové lišty. Zapne režim invalida.

`get_pdf_viewer`: vykreslí šablonu s prohlížečem PDF souborů.

`post_pdf_viewer`: po kliknutí na vybranou přílohu (PDF dokument) v tabulce úřední desky vykreslí tento dokument do prohlížeče PDF.

`retrieve_xml_feed`: vstupní bod pro aktualizaci obsahu na desce. Voláno cronem nebo manuálně na adrese `/update`. Z konfiguračních souborů načte URL adresu serveru, na který spisová služba publikuje XML soubor a PDF soubory. Pokud není aplikace připojena k tomuto serveru, zapíše tuto skutečnost do logu a přesměruje aplikaci zpět do hlavního menu.

Pokud je aplikace připojena k specifikovanému serveru, pokračuje na další krok z obr. 3.1. Stáhne aktuální XML soubor pomocí metody `get_actual_xml` ve které ho porovná se starým (již staženým) XML souborem voláním metody `compare_xml_feeds`. Nakonec zavolá metodu `retrieve_documents` a tím stáhne nové nebo změněné PDF soubory.

`get_actual_xml`: stáhne ze serveru spisové služby XML soubor. Pokud ještě neexistuje složka /ui/xml/, vytvoří ji a uloží do ní tento soubor (po instalaci aplikace). Pokud v této cestě již existuje soubor `main_xml.xml`, uloží nový soubor jako `main_xml_new` a zavolá porovnání souborů `compare_xml_feeds`. Pokud v cestě soubor neexistuje, uloží ho pod názvem `main_xml.xml`.

`compare_xml_feeds`: metoda nejprve načte starý a nový XML soubor do proměnných. V dvou for cyklech zpracuje každý XML soubor zvlášť, získá z nich pomocí funkce `get_hashname` haše do polí `old_hashes` a `new_hashes`. Tyto haše jsou generovány z `ID` a `DATETIME` každé z příloh. Vznikne tak jednoznačný identifikátor dané přílohy. Pole jsou porovnána a vymažou se soubory s haši, které jsou ve starém XML souboru ale né v novém.

Nutnost hašovat názvy příloh vzniká z toho, že když úředník nebo úřednice ve spisové službě upraví již vyvěšený dokument, algoritmus založený na jménech souborů (nebo `ID`) by změnu nerozpoznal a tudíž by dokument neaktualizoval. `DATETIME` je poslední čas, kdy byl daný dokument upraven a ve spojení s `ID` tak vznikne pro každou verzi dokumentu unikátní identifikátor a tím pádem algoritmus pozná, že má tento nově upravený soubor stáhnout. Tag `DATETIME` byl do spisové služby úřadem přidán speciálně pro tuto funkci v `yadb`.

`retrieve_documents`: metoda naskenuje složku /ui/pdf a načte všechny uložené haše (názvy souborů) do pole. Porovná haše příloh z XML souboru a stáhne pouze ty, které se na disku nevyskytují ale v XML souboru ano. Zde je ukázka kódu metody `retrieve_documents`:

```
function retrieve_documents(\Base $base, $url)
    {
        $xml = simplexml_load_file($base->get('ROOT') . "/ui/xml/main_xml.xml");
        $dir = $base->get('ROOT') . "/ui/pdf/";
        if (!file_exists($dir)) {
            mkdir($dir, 0777, true);
        }
        $filehashnames = scandir($dir);

        if ($xml->item) {
            foreach ($xml->item as $item) {
                foreach ($item as $element => $value) {
                    if ($element === "PRILOHY") {
                        $pcount = 1;
                        foreach ($value->item as $attachment => $it) {
                            $hashname = $this->get_hashname($it->ID, $it->DATETIME);
                            if (!$this->hash_lookup($hashname, $filehashnames)) {
                             $this->download_attachment_pdf($base, $it, $hashname, $url);
                            }
                            $pcount++;
                        }
                    }
                }
            }
        }
    }
```

`hash_lookup`: slouží pro vyhledání haše v daném poli.

`download_attachment_pdf`: stáhne jeden konkrétní soubor, tato metoda je volána z `retrieve_documents`.

`get_news`: voláno pomocí `/news` z tlačítka v hlavním menu. Metoda načte RSS URL adresu z konfiguračního souboru. Pokud je aplikace online, stáhne RSS soubor z městského RSS feedu a vykreslí šablonu, na které si občané mohou přečíst novinky. Pokud je aplikace offline, novinky nezobrazí.

`get_info`: vykreslí návod na použití elektronické úřední desky pro nepřihlášeného uživatele (občana). Voláno pomocí cesty `/info` z tlačítka z hlavního menu.

3.3.5 Board.php

Tento kontrolér obsahuje mimo prázdnou metodu `install` dvě důležité metody. Slouží k sestavení grafické šablony s úřední deskou, následně je tato deska vykreslena uživateli.

`get_filtered`: metoda, která je volána z hlavního menu pomocí cesty `/filtered`. V metodě je vyřešená logika filtrace různých typů PDF dokumentů na úřední desce. Když například uživatel v hlavním menu zvolí, že chce zobrazit pouze *dotace*. Filtrovat se dá deset různých typů dokumentů, tyto typy jsou definovány zdrojovým XML souborem, který reprezentuje datovou základnu pro úřední desku. Nastaví tedy filtr a pak pomocí `get_all` vykreslí vyfiltrovanou úřední desku.

`get_all`: z hlavního menu je tato metoda zavolaná tlačítkem s cestou `/all`. V metodě je nejprve načten lokálně uložený XML soubor, kde je úřední deska datově vyjádřena. XML soubor je procházen pomocí for cyklů a z dat v něm je sestavována interaktivní HTML tabulka, která slouží jako digitální úřední deska. Tuto tabulku

lze vidět na obr. B.2 v příloze B. Vygenerovaná úřední deska je pak prezentována uživateli.

Tabulka v hlavičkách obsahuje jednotlivé tagy z XML souboru (typ, vyvěšeno/staženo, popis atd.). Jednotlivé položky jsou pak každá na svém řádku, s tím že obsahují odkaz na PDF přílohy, které mohou mít i sto stran. Podrobněji je grafické uživatelské rozhraní popsáno v sekci 3.5 této kapitoly.

3.3.6 ImageY.php

Jednoduchá třída pro správu obrázků (reklamy).

`install`: nastavení databázové tabulky `image` za využití F3 Schema Builder. Nastaví sloupce tabulky podle specifikace v ORM datové třídě `data\ImageY.php`.

`get_images_page`: voláno z administrátorské navigační lišty (cesta `/admin/images`). Vykreslení stránky pro správu (nahrávání a mazání) obrázků.

`post_images`: metoda volaná z administrační stránky s obrázky zpracovává nově nahraný obrázek (cesta `/admin/images/upload`). Kontroluje jeho velikost, šířku výšku, orientaci a pokud splňuje předepsaná kritéria, je nahrán na server a zobrazen s ostatními obrázky v klidovém režimu.

`delete_image`: metoda, která smaže obrázek (volána tlačítkem v tabulce s obrázky, cesta `/admin/images/delete`).

3.3.7 Stats.php

`install`: nastavení databázové tabulky `stat` za využití F3 Schema Builder. Nastaví sloupce tabulky podle specifikace v ORM datové třídě `data\Stats.php`.

`get_statistics`: voláno AJAXem ze správcovské stránky s návštěvností (cesta `/admin/statistics`). Načte z databázové tabulky `stat` informace o návštěvnosti v různých obdobích. Měří se denní, měsíční, roční a celková návštěvnost která je následně vykreslena na stránce se statistikou do Chart.js grafu.

`get_statistics_page`: voláno kliknutím na tlačítko správcovské lišty pomocí cesty `/admin/statistics_page`. Vykreslí správcovskou stránku s Chart.js grafem.

3.4 Databáze a ORM třídy

Pro implementaci databáze jsem použil MySQL (respektive MariaDB). Jak bylo řečeno v sekci 2.3, databáze je malá a obsahuje tabulky pro správu uživatelů, obrázků a návštěvnosti. Většina dat totiž pochází z externího zdroje (XML souboru spisové služby). Pro komunikaci s databází používám ve svém programu objektivně relačně mapovací modely (ORM), které komunikaci usnadňují. Vlastní modely mám implementované v několika třídách, které mají stejný název jako třídy kontrolerů. Všechny

dědí ze třídy F3 Cortex, což je plugin do F3, který zpřístupňuje funkcionalitu ORM. Každý model je mapovaný na některou z databázových tabulek a pomocí několika F3 Cortex funkcí komunikuji s danou tabulkou.

F3 Cortex přináší jednoduchý způsob, jakým lze ORM modely definovat. Zde je ukázka definice třídy `data\User.php`:

```
1  class User extends \DB\Cortex
2  {
3      protected $db = 'DB';
4      protected $table = 'y_user';
5      protected $primary = 'id';
6
7      protected $fieldConf = [
8          'name' => [
9              'type' => 'VARCHAR256',
10             'nullable' => false,
11             'unique' => true
12         ],
13         'role' => [
14             'belongs-to-one' => '\yadb\data\Roles'
15         ],
16         'password' => [
17             'type' => 'VARCHAR256',
18             'nullable' => false
19         ],
20         'image' =>[
21             'has-many' => array('\yadb\data\ImageY','owner')
22         ]
23     ];
24 }
```

V poli `$fieldConf` se nastavují atributy dané tabulky, takto se vyhnu přímému používání SQL dotazů. Lze pozorovat, že například atribut `name` je typu `VARCHAR256`, nelze ho vynulovat a je unikátní. Zavoláním metody `data\User::setup()` se podle této konfigurace nastaví databázová tabulka `y_user`. Další výhodou použití F3 Cortex místo čistých SQL dotazů je snadná správa kardinality vztahu mezi tabulkami. Zde například `has-many` u atributu `image` znamená, že tabulka `user` je s tabulkou (modelem) `image` ve vztahu N:1. V definici třídy `data\ImageY` je pak zapsán vztah opačně (`belongs-to-one`). Tímto způsobem se dají modelovat všechny známé typy kardinality.

Stejným způsobem, jako je implementována třída `data\User.php`, jsou implementovány třídy: `data\Roles.php`, `data\ImageY.php` a `data\Stats.php`. Každá s vlastní konfigurací dle specifikace v části 2.3.

3.5 Uživatelské rozhraní

Vzhledem k tomu, že je má aplikace k použití obyčejnými občany (narozdíl od programátorské knihovny), strávil jsem s vytvářením uživatelského rozhraní poměrně dost času. Trojice *webových jazyků* se velmi osvědčila a výsledné uživatelské rozhraní je velmi funkční, jak pro desktop (správcovské) tak pro širokoúhlou dotykovou obrazovku (občanské). V této části povrchně popíšu každou stránku, na kterou může uživatel mé aplikace narazit. HTML stránky jsou vykreslovány po zavolání některé z cest jako F3 šablony. Soubory se šablonami správcovského rozhraní mají předponu `admin_*`.

Hlavní font je v aplikaci `yadb` *Fira Sans*[1]. Je to ten samý font, o kterém se zmiňuje *manuál vizuální identity*, který mi byl zaslán Mgr. Žaloudkem a který je použit na hlavní stránce města Děčín. Také používám stejné odstíny barev, které jsou použity na hlavní stránce magistrátu. Dodržuji tak celistvý vzhled vizuální identity města Děčín.

3.5.1 Společné soubory

Mezi společné šablony obou typů rozhraní se řadí:

`header.html`: zde importuji většinu CSS a některé JS knihovny a soubory. Toto je hlavička HTML dokumentu, mezi ní a `footer.html` vykresluji všechen dynamický obsah.

`footer.html`: patička, pouze značí konec HTML dokumentu.

`scripts.html`: zde na jednom místě importuji většinu JS knihoven a pluginů včetně některých vlastních JS skriptů. Tento soubor vkládám nad patičku, takže je v každé vykreslené šabloně.

3.5.2 Občanské rozhraní

Když se s deskou patnáct minut nemanipuluje, přejde do nečinného stavu. Toto je základní stav mé aplikace, kdy se na desce zobrazuje stránka `carousel.html`, zde se zobrazují v rotaci obrázky (reklama) nahraná správci. Využil jsem Bootstrap komponentu Carousel (kolotoč). Nad obrázky je text s logem Děčína, který oznamuje kolemjdoucím, že tento přístroj je úřední deska.

Po dotyku obrazovky se uživatel dostane do hlavního menu `menu.html`, zde je mu prezentován rozcestník dalších možností (obr. B.1 v příloze B). Mimo to uživatel poprvé vidí spodní dotykové menu, na kterém lze zapnout režim pro invalidy, vrátit se na domovskou stránku, zapnout vyhledávání a zjistit čas. Uživatel má v hlavním menu na výběr v tuto chvíli ze čtyř možností.

[1] Fira Sans https://fonts.googleapis.com/css2?family=Fira+Sans&display=swap

- Všechny dokumenty úřední desky – zobrazení kompletní úřední desky se všemi typy dokumentů.

- Filtrovat dokumenty – menu pro výběr filtrace dokumentů úřední desky podle typu.

- Novinky – zobrazí RSS novinky z webu města Děčín (`news.html`).

- Návod na používání úřední desky – zobrazí návod na používání úřední desky (`info.html`).

Nejdůležitější z těchto možností je úřední deska, proto se na ní zaměřím podrobněji. Po kliknutí na tlačítko *Všechny dokumenty úřední desky* se po chvilce načítání zobrazí úřední deska (`board.html`). Pro animační efekt zde využívám knihovnu `Animate.css`. Stránka s úřední deskou sestává z několika částí.

Vyhledávací pole se po rozkliknutí rozbalí na dotykovou klávesnici `simple-keyboard`. Vyhledávání je reaktivní a ihned po napsání i jediného písmene prohledá celou úřední desku a zobrazí filtrovaný výsledek. Do vyhledávacího pole lze psát datum, typ, nebo libovolný řetězec znaků, úřední deska zobrazí nejbližší shodu. To se hodí, když občan tuší, co hledá.

Jako úřední deska zde slouží dynamická tabulka (knihovna `DataTables`), kterou lze ovládat pomocí dvou modrých šipek nebo čísel stránky. Tabulka je vizuální reprezentací XML souboru ze spisové služby (aktuální stažené kopie). V tabulce jsou seřazeny všechny oznámení zpočátku dle data vyvěšení. Občan může filtrovat pomocí šipek v hlavičce tabulky dle jednotlivých sloupečků (typ, vyvěšení/stažení, popis, zdroj, přílohy).

Ve chvíli, kdy občan nalezne hledanou vyhlášku nebo jinou vyvěšenou písemnost, může rozkliknout tlačítko `Příloha`. V tu chvíli se mu zobrazí příloha (PDF dokument) vybrané vyhlášky v `PDF.js` prohlížeči dokumentů (`pdf_viewer.html` lze vidět na obr. B.3 přílohy B). Změní se i spodní dotykové menu: objeví se tlačítko zavřít, čímž se občan vrátí zpět na úřední desku. Také se zde objeví QR kód (`qrcode.js`), který po naskenování odkazuje na URL adresu právě prohlíženého dokumentu. Občan si tak může dokument stáhnout do telefonu.

Zdrojový kód knihovny `PDF.js` jsem musel upravit a to ve dvou ohledech. Jelikož je zdroj XML externí k mé aplikaci, neovlivním, co se v něm objeví a vzhledem k tomu, že se v PDF dokumentech vydaných magistrátem často objevují URL odkazy, bylo potřeba je odstranit. Občan by se skrze ně mohl prokliknout mimo aplikaci `yadb` a to je nevhodné pro chod této desky. Proto každý PDF dokument před zobrazením skenuji a odstraňuji z něj všechny odkazy. Poté jsem musel odstranit některá tlačítka v panelu možností PDF prohlížeče. Například tlačítko tisku, nebo tlačítko stažení a otevření souboru, přes ně by se občan mohl dostat do prohlížeče souborů jednodeskového počítače a jistými způsoby například spustit příkazovou řádku, nebo desku jinak softwarově poničit.

3.5.3 Správcovské rozhraní

Správcovské rozhraní je desktopová webová stránka s několika šablonami (obr. B.5, příloha B). V souborech `admin_login.html` a `admin_register.html` jsou implementovány přihlašovací a registrační formuláře pro správce. Po přihlášení do aplikace se správce dostane na hlavní stránku správy `yadb`: `admin_dashboard_home.html` (lze vidět na obr. B.5 v příloze B). Zde se správci zobrazí navigační lišta, ze které se dostane na všechny ostatní stránky správy. Také se mu ukáže současný stav desky (teplota, zatížení procesoru apod.). Pomocí tlačítek může desku manuálně aktualizovat, konfigurovat a nebo si jí zobrazit z pohledu občana, tak může zjistit, jestli je s ní něco v nepořádku.

Záložka *správci* (`admin_dashboard_administrators.html`) je důležitá pro hlavního administrátora, ten zde může vytvářet moderátorské účty, mazat je a upravovat je. Moderátoři (úředníci) zde mohou prohlížet účty ostatních moderátorů.

Na stránce *reklama na desce* (`admin_dashboard_images.html`) mohou moderátoři přidávat reklamu na úřední desku pomocí nahrávacího dialogu. Mohou zde vidět již nahrané obrázky a mazat je.

Stránka *logy* (`admin_dashboard_logs.html`) je vhodná pro zjišťování toho, co se v aplikaci dělo. Jsou zde zobrazeny tři logy, uživatelský, systémový a chybový. Je zřejmé, co každý z logů zaznamenává. Logování je uskutečněno pomocí F3 `Log` třídy napříč celou aplikací.

Ve *statistice* (`admin_dashboard_stats.html`) je zobrazena návštěvnost, denní, měsíční, roční a celková. Je vykreslena do Chart.js grafu. Díky této stránce lze zjistit, zda elektronickou úřední desku někdo vůbec využívá. V době chytrých mobilních telefonů a velkého pokrytí sítí 4G je možné, že deska zůstane nevyužitá a bude sloužit pouze jako splnění zákonné povinnosti.

Administrátor aplikace `yadb` může konfigurovat různé parametry aplikace skrze rozhraní konfigurace (`admin_dashboard_config.html`). Konfigurační soubory jsou uloženy ve složce `config`. Dá se nastavit počet hodin, kdy se má restartovat aplikace, počet minut, po kterých se má aplikace aktualizovat. Tyto dvě možnosti jsou v tuto chvíli nefunkční a toto je potřeba nastavovat přes crontab. Nepodařilo se mi bohužel zprovoznit plugin F3 Cron, který zpřístupňuje cronjoby přímo v aplikaci. Dále může administrátor mimo URL s RSS feedem nastavit URL, na které je XML soubor s informacemi pro úřední desku a URL, ze které se stahují PDF přílohy. Tato možnost tu je pro případ, že by se v budoucnu přesunula spisová služba na jiný server.

Na stránce *o programu* (`admin_dashboard_info.html`) popisuji, jakým způsobem má aplikaci používat správce. Je to uživatelská příručka pro pracovníky magistrátu.

V *nastavení* (`admin_dashboard_user_settings.html`) může správce změnit své uživatelské jméno nebo heslo.

Kapitola 4

Nasazení a testování prototypu

V této kapitole ukážu způsob, jakým nasadit aplikaci na jednodeskový počítač, aby sloužila jako autonomní elektronická úřední deska. S touto částí je spojen i instalační manuál. Aplikaci lze nasadit na libovolný Linux nebo Windows (testovány Ubuntu a Windows 10), přičemž Linux jsem využil pro provoz a Windows pro testování a vývoj. Představím prototyp elektronické úřední desky, který byl sestaven ve spolupráci s děčínským magistrátem a vedoucím této práce. Nakonec zmíním Joomla modul, který posloužil jako základ současné webové verze digitální úřední desky města Děčín.

4.1 Nasazení na jednodeskový počítač

Vzhledem k tomu, že je má aplikace webová a je nenáročná na výpočty a na grafiku, rozhodl jsem se použít pro ovládací počítač elektronické úřední desky jednodeskový počítač. Takový jednodeskový počítač je řádově levnější než plně desktopové počítače, které používají čeští výrobci elektronických úředních desek ve svých zařízeních. Byl mi zapůjčen NVIDIA Jetson Nano[1], který se při testování prokázal jako dostatečný pro běh mé aplikace.

Na Jetsonu běží operační systém založený na Ubuntu. Při instalaci své aplikace na Jetson jsem postupoval dle instalačního manuálu v podkapitole 4.2.1. Tento způsob instalace by se neměl zásadně lišit na rozdílných linuxových distribucích. Instalační manuál se také nachází v textové podobě v hlavním repozitáři (yadb-release) na přiloženém CD pro snadnější kopírování příkazů. Instalační manuál předpokládá, že máte čistě nainstalovanou linuxovou distribuci založenou na Debianu. Při instalaci je potřeba základní znalosti práce s příkazovou řádkou Linuxu. Příkazy pro příkazovou řádku jsou zvýrazněny červeně a každý příkaz je ukončen středníkem.

[1] https://developer.nvidia.com/embedded/jetson-nano-developer-kit

4.2 Instalační manuál

4.2.1 Linux (produkční prostředí)

Instalace a nastavení Apache

```
sudo apt-get update;
sudo apt-get install apache2;
sudo a2enmod rewrite; # Zapnutí mod rewrite pro Apache.
sudo a2enmod headers; # Kvůli F3.
sudo cp /etc/apache2/sites-available/000-default.conf
/etc/apache2/sites-available/yadb.conf;
sudo nano /etc/apache2/sites-available/yadb.conf; # Sem vložte
následující text (VirtualHost) a uložte.
```

```
<VirtualHost *:80>
        ServerName yadb.local
        ServerAlias yadb
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/yadb-release/
        ErrorLog ${APACHE_LOG_DIR}/yadb-error.log
        CustomLog ${APACHE_LOG_DIR}/yadb-access.log combined
</VirtualHost>
```

```
# Nastavit apache, aby četl .htaccess v yadb.
sudo nano /etc/apache2/apache2.conf;
# Nalezněte, kde se v souboru apache2.conf nachází následující a
změňte AllowOverride None na AllowOverride All a uložte.
```

```
<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride All # <——— Změnit z None na All.
        Require all granted
</Directory>
```

Instalace PHP

```
sudo add-apt-repository ppa:ondrej/php;
sudo apt-get update;
sudo apt-get install php7.4 php7.4-common php7.4-mysql php7.4-cgi
php7.4-cli php7.4-fpm php7.4-mbstring php7.4-curl php7.4-xmlrpc
php7.4-gd php7.4-zip php7.4-pdo-sqlite php7.4-xml;
sudo systemctl restart apache2;
sudo nano /etc/hosts; # Na konec souboru vložit: 127.0.0.1 yadb.local
```

Instalace a nastavení databáze (MySQL)

```
sudo apt install mysql-server;
sudo mysql_secure_installation; # Postupujte podle pokynů.
systemctl status mysql.service; # Otestování, jestli MySQL běží.
```

Vytvoření databáze (defaultně yadb)

```
      mysql -u root -p ;
Prostředí MySQL
      mysql> CREATE DATABASE yadb;
      mysql> exit
```

Alternativně lze pro vytvoření databáze použít PHPMyAdmin:

```
sudo apt-get install phpmyadmin php-mbstring php-gettext -y;
chromium-browser ‘‘localhost/phpmyadmin’’; # Pokud máte přístupné
desktopové prostředí.
```

Instalace webové aplikace

1. Zkopírujte složku **yadb-release** do **/var/www/html/** z přiloženého CD.
2. Nastavte vlastníka složky yadb-release na uživatele www-data:
`sudo chown -R www-data:www-data /var/www;`
3. Nastavte spustitelnost souborů ve složce scripts (skript je příklad pro Jetson):
`cd /var/www/html/yadb-release/scripts; sudo chmod +x system_info.sh;`
4. `sudo chmod -R 777 /var/www/html/yadb-release;`
5. Nastavte informace v souboru `yadb-release/db.ini` na shodné s databází (username a password).
6. Otevřete stránku v prohlížeči: **http://yadb.local/install** tímto se aplikace nainstaluje (prvotní nastavení trvá několik minut).
   Registrace administrátorského účtu -> tajné heslo v /config/secret.ini.
   Přihlášení.
   yadb.local/admin – přístup do správcovské části.
   yadb.local/ – přístup do občanské části.
6.1. Pokud nemáte prohlížeč Chrome, nainstalujte ho:
`sudo apt install chromium-browser;`

Následující kroky můžete vynechat, pokud chcete aplikaci pouze testovat.
7. Cronjoby, plánované restartování a aktualizace souborů na desce.
`sudo crontab -e` # Vložte následujících pět řádků a uložte:

Aktualizace informací o provozu desky.
* * * * * cd /var/www/html/yadb-release/scripts; /bin/bash ./system_info.sh
Aktualizace dokumentů každých /xx minut.
*/15 * * * * curl yadb.local/update
Restartování desky každých /xx hodin.
1 */8 * * * sleep 60 && /sbin/reboot

8. Nastavte automatické přihlašování po spuštění.
8.1 Nastavte, ať se systém nezamyká a neuspává se.
8.2 Nastavte kioskový režim webového prohlížeče při spuštění:
`chromium-browser -kiosk ‘‘http://yadb.local’’;`
v XFCE4 lze tuto funkci nastavit v Session and Startup.

4.2.2 Windows (vývojové prostředí)

1. Instalace XAMPP
https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/7.4.8/
Při instalaci zaškrtněte Apache a MySQL.

2. Vytvořte databázi 'yadb' v MySQL CLI nebo PHPMyAdmin (bin-czech-ci kódování).
3. Do souboru \xampp\apache\conf\extra\httpd-vhosts.conf vložte:

```
<VirtualHost *:80>
        ServerAdmin webmasterlocalhost
        DocumentRoot "C:/xampp/htdocs/yadb-release"# podle vašeho umístění
         ServerName yadb.local
        ErrorLog "logs/yadb-error.log"
        CustomLog "logs/yadb-access.log"common
</VirtualHost>
```

4. Otevřete soubor hosts jako administrátor v textovém editoru:
C:\windows\system32\drivers\etc\hosts
Přidejte na nový řádek:
127.0.0.1 yadb.local

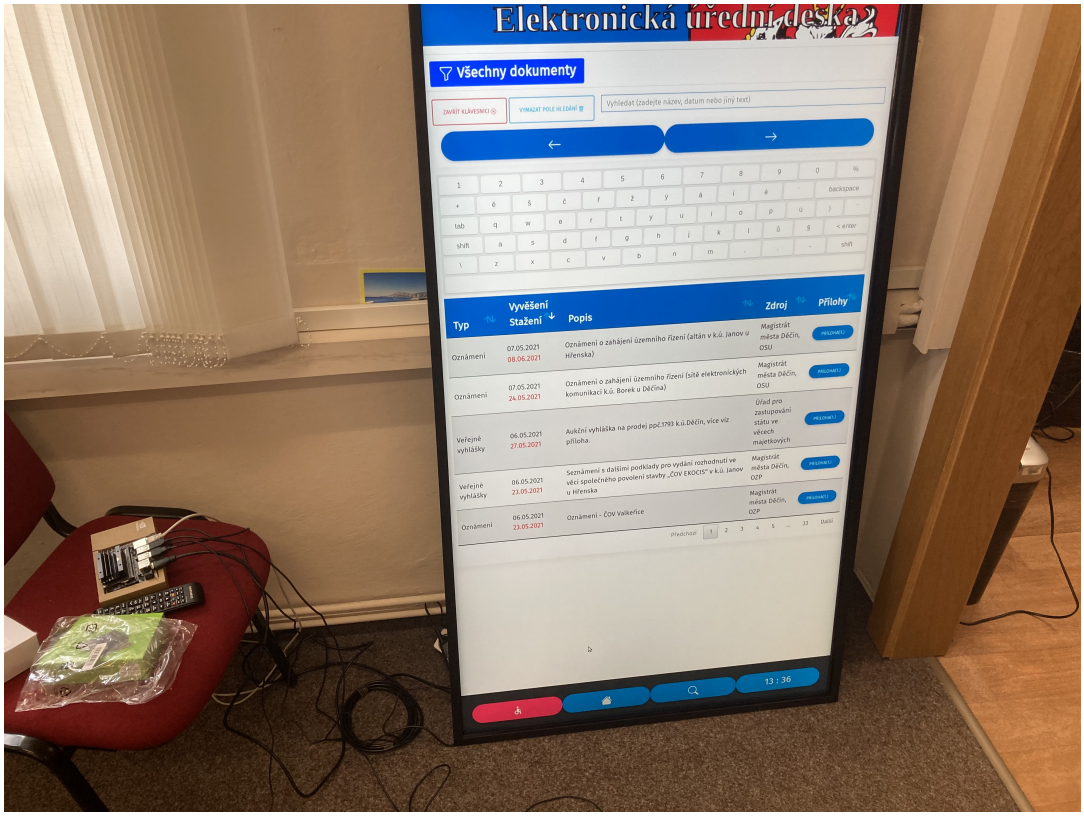
5. Do složky /htdocs/ ve složce /xampp/ vložte složku yadb-release z přiloženého CD.
6. Zapněte XAMPP Control panel a v něm Apache a MySQL.
7. Otevřete stránku v prohlížeči: **http://yadb.local/install** tímto se aplikace nainstaluje (prvotní nastavení trvá několik minut).
    Registrace administrátorského účtu -> tajné heslo v /config/secret.ini.
    Přihlášení
    yadb.local/admin - přístup do správcovské části.
    yadb.local/ - přístup do občanské části.

4.3 Prototyp EÚD na Magistrátě města Děčín

Po instalaci dle instalačního manuálu byl ve spolupráci s děčínským magistrátem použit Jetson v prototypu úřední desky. Tento prototyp sestává z NVIDIA Jetson Nano, 4K 65" obrazovky a dotykového rámečku[2]. Bylo mi řečeno, že obrazovka (televize) stála třicet pět tisíc korun, rámeček stál okolo čtrnácti tisíc korun a Jetson Nano přibližně tři tisíce. To je dohromady něco přes padesát tisíc korun. To znamená, že aby se cenově tento prototyp přiblížil k nejlevnější 65" desce od českých výrobců,

[2]http://www.kobe.cz/hlavni-nabidka/irda-dotykove-ramy/dotykovy-ram-na-lcd-displej-65-multitouch-usb.html

zbývá v rozpočtu ještě přibližně sto šedesát tisíc korun. To by mělo stačit na pokrytí nákladů sestavení železného krytu, termoregulace a elektroinstalace celého zařízení. Bonusem je to, že kromě nákladů na provoz už se za toto řešení nic platit nemusí. Software (`yadb`) je kompletně zdarma a bez licenčních a provozních poplatků.



Obrázek 4.1: Dotykový prototyp elektronické úřední desky na Magistrátě města Děčín

Při testování aplikace na úřadě jsem zjistil, že dotykový rámeček z Číny podporuje gesta podobně jako chytrý telefon. Dá se tak pomocí známých gest například přiblížit uživatelské prostředí nebo PDF dokumenty, což může pomoci lidem se zhoršeným zrakem. Ve chvíli, kdy deska přejde zpět do nečinného režimu se toto přiblížení restartuje a deska je znovu plně použitelná (kdyby někdo zanechal desku v přiblíženém stavu).

Také jsem zjistil, že čínský dotykový rámeček se po zapojení chová inverzně. Je tedy potřeba při každém spuštění Jetsonu zavolat tento příkaz, který nastaví správnou orientaci dotykového rámečku (relativně k vertikální televizi):

```
xinput set-prop 6 "Coordinate Transformation Matrix"-1 0 1 0 -1 1 0 0 1
```

Jelikož je televize 4K, je potřeba stáhnout rozlišení, aby byl text čitelný. Toho jsem dosáhl pomocí tohoto příkazu (který se také volá při každém zapnutí desky):

```
xrandr -output HDMI-0 -scale 0.5x0.5
```

4.4 Webová digitální úřední deska

Jak bylo řečeno v kapitole 1, běží hlavní webová stránka MM Děčín na CMS Joomla. Naprogramoval jsem tedy Joomla modul, který posloužil jako základ pro nynější podobu webové úřední desky[3]. Nová webová úřední deska už je integrovaná přímo do webové stránky magistrátu (na rozdíl od předchozího řešení), dodává tak občanům lepší uživatelský zážitek a graficky ladí s již existující webovou stránkou magistrátu - přebírá její CSS styly. Modul je okleštěnou verzí hlavní webové aplikace a obsahuje až na drobné rozdíly totožný zdrojový kód jako hlavní aplikace `yadb`, nebudu se jím proto zabývat dopodrobna. Zdrojový kód tohoto modulu je dostupný na přiloženém CD (obsah CD v příloze A).

[3]Webová úřední deska https://www.mmdecin.cz/uredni-deska

Závěr

Cílem práce bylo prozkoumat existující řešení elektronických úředních desek a na základě toho navrhnout a implementovat vlastní řešení elektronické úřední desky pro město Děčín. Bylo prozkoumáno několik řešení současných českých výrobců elektronických úředních desek a na základě jejich produktů byly specifikovány vlastnosti a požadavky elektronické úřední desky obecně.

V rámci bakalářské práce byla navrhnuta a implementována webová aplikace (digitální úřední deska), která byla následně nasazena a testována v prototypu elektronické úřední desky. Webová aplikace je zcela open–source a tím pádem je i nadále rozšiřitelná kýmkoliv, včetně úředníků děčínského magistrátu. Splňuje nutné podmínky pro úřední desky a je schopna autonomního provozu. Cíle práce byly tudíž splněny.

Předpokládám, že digitální úřední deska navržená v této práci by mohla v případě plného nasazení městem Děčín ušetřit některé výdaje. Licenční, provozní a pořizovací náklady obslužného programu v elektronických úředních deskách od tuzemských výrobců díky této práci odpadají. Dále se dá ušetřit na předimenzovaných ovládacích počítačích, které lze v elektronické úřední desce nahradit jednodeskovým počítačem.

Deska je přístupná všem lidem: písmo by mělo být dostatečně čitelné a ovládání intuitivní. K tomu má v sobě zabudovaný mechanismus, kterým se zpřístupní i lidem na invalidním vozíku. Uživatelské rozhraní je totiž optimalizované pro dotykový panel usazený na výšku (který měří okolo dvou metrů). Desku mohou úředníci vzdáleně spravovat a konfigurovat a zároveň kontrolovat statistiku provozu. Digitální deska je určena pro *self–hosting* děčínským magistrátem. Odpadají tak další náklady (za vzdálenou správu), které někteří tuzemští výrobci desek účtují.

Při implementaci bylo využito dobře známých a široce používaných programovacích jazyků (a technologií) pro webové prostředí, mělo by tak rozšiřování programu ostatními lidmi probíhat hladce.

Program si *převzalo* IT oddělení děčínského magistrátu a nadále testují prototyp, který bude snad v budoucnu uveden do ostrého provozu. Pro to je však nutno sestavit ochranný kovový kryt a zprovoznit v něm elektroinstalaci, zajistit odolnost proti povětrnostním podmínkám apod. Tato snaha je nad rámec mé bakalářské práce, která se primárně zabývá obslužným programem elektronické úřední desky. Další kroky ve zlepšení tohoto programu bych vedl směrem k zobecnění programu, rozšíření funkcionality a předělání uživatelského rozhraní.

Program je v tuto chvíli dělaný na míru městu Děčín, ale v budoucnu by se dal zobecnit pro libovolné město. Města po celé České republice by mohla přejít na toto (nebo podobné) *self–hosting* řešení úředních desek a ušetřit napříč republikou miliony korun. Snaha standardizovat elektronické úřední desky v Česku by nejspíše volala po existenci větší komunity lidí, která by se o tento projekt starala.

Má digitální úřední deska by nemusela sloužit jen jako *úřední deska*, ale přinejmenším jako informační kiosek, kde by se dala procházet alespoň webová stránka města Děčín. Podobné informační kiosky jsou zabudovány do elektronických úředních desek tuzemských výrobců.

Zdrojový kód uživatelského rozhraní je vcelku nepřehledný a možná by se vyplatilo celé uživatelské rozhraní předělat do nějakého singlepage frameworku, jako je například Vue.js. Také by bylo dobré opravit některé nedostatky tohoto programu, především nastavení intervalů restartování a aktualizace desky, které se v tuto chvíli nedají nastavit přes webové rozhraní.

Literatura

[1] Animate.css. *Animate.css* [online]. [cit. 2021-5-9]. Dostupné z: https://animate.style/

[2] Apache HTTP Server Project. *Apache: The Number One HTTP Server On The Internet* [online]. The Apache Software Foundation, 2021 [cit. 2021-5-9]. Dostupné z: https://httpd.apache.org/

[3] Bootstrap. *Bootstrap* [online]. 2021 [cit. 2021-5-9]. Dostupné z: https://getbootstrap.com/

[4] Chart.js. *Chart.js: Simple yet flexible JavaScript charting for designers & developers* [online]. 2021 [cit. 2021-5-9]. Dostupné z: https://www.chartjs.org/

[5] ČESKÁ REPUBLIKA. Zákon č. 500/2004 Sb.: Správní řád In: *Sbírka zákonů České republiky*. 2004, částka 174/2004, číslo 500. Dostupné také z: https://www.zakonyprolidi.cz/cs/2004-500

[6] DataTables. *DataTables* [online]. Scotland: SpryMedia, 2021 [cit. 2021-5-9]. Dostupné z: https://datatables.net/

[7] Fat-Free Framework. *Fat-Free Framework: A powerful yet easy-to-use PHP micro-framework designed to help you build dynamic and robust web applications - fast!* [online]. [cit. 2021-5-9]. Dostupné z: https://fatfreeframework.com

[8] Font Awesome. *Font Awesome* [online]. Fonticons, 2021 [cit. 2021-5-9]. Dostupné z: https://fontawesome.com/

[9] F3-access. *Access: Route access control for the PHP Fat-Free Framework* [online]. [cit. 2021-5-9]. Dostupné z: https://github.com/ikkez/f3-access

[10] F3-cortex. *Cortex: A general purpose Data-Mapper for the PHP Fat-Free Framework* [online]. [cit. 2021-5-9]. Dostupné z: https://github.com/ikkez/f3-cortex

[11] F3-flash. *Flash: Add simple Flash messages and Flash Keys to PHP Fat-Free Framework* [online]. [cit. 2021-5-9]. Dostupné z: https://github.com/ikkez/f3-flash

[12] F3-schema-builder. *SQL Schema Builder: An extension for creating and altering SQL database tables.* [online]. [cit. 2021-5-9]. Dostupné z: https://github.com/ikkez/f3-schema-builder

[13] Google Fonts. *Google Fonts* [online]. 2021 [cit. 2021-5-9]. Dostupné z: https://fonts.google.com/

[14] JQuery. *JQuery* [online]. 2021 [cit. 2021-5-9]. Dostupné z: https://jquery.com/

[15] JQuery BlockUI Plugin. *JQuery BlockUI Plugin* [online]. 2013 [cit. 2021-5-9]. Dostupné z: https://malsup.com/jquery/block/

[16] Material Design for Bootstrap. *Material Design for Bootstrap* [online]. 2021 [cit. 2021-5-9]. Dostupné z: https://mdbootstrap.com/

[17] MySQL. *MySQL* [online]. Oracle Corporation, 2021 [cit. 2021-5-9]. Dostupné z: https://www.mysql.com/

[18] PDF.js. *PDF.js: A general-purpose, web standards-based platform for parsing and rendering PDFs.* [online]. Mozilla, 2021 [cit. 2021-5-9]. Dostupné z: https://mozilla.github.io/pdf.js/

[19] PHP: Documentation. *PHP* [online]. 2021 [cit. 2021-5-9]. Dostupné z: https://www.php.net/docs.php

[20] PhpDocumentor Manual. *PhpDocumentor Manual: phpDocumentor tags* [online]. 2011 [cit. 2021-5-22]. Dostupné z: https://manual.phpdoc.org/HTMLframesConverter/default/

[21] QRCode.js. *QRCode.js* [online]. [cit. 2021-5-9]. Dostupné z: https://davidshimjs.github.io/qrcodejs/

[22] Simple-keyboard. *Simple-keyboard: Javascript Virtual Keyboard - Customizable, responsive and lightweight* [online]. [cit. 2021-5-9]. Dostupné z: https://hodgef.com/simple-keyboard/

[23] Stack Overflow. *Stack Overflow* [online]. 2021 [cit. 2021-5-9]. Dostupné z: https://stackoverflow.com/

[24] W3Schools. *W3Schools* [online]. 2021 [cit. 2021-5-9]. Dostupné z: https://www.w3schools.com/

Příloha A

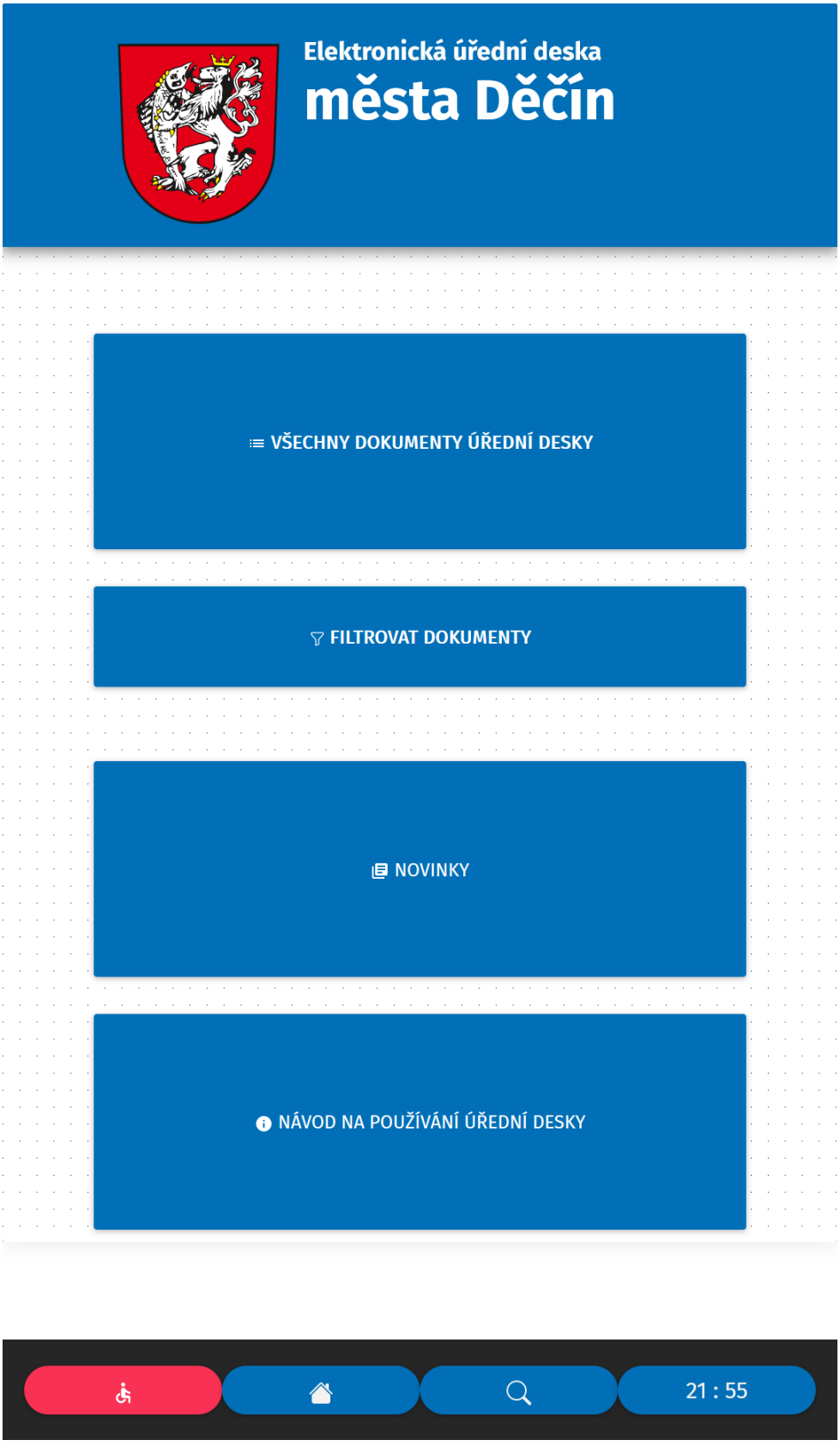
Obsah přiloženého CD

```
nabidky ..............Obsahuje PDF soubory cenových nabídek firem.
  digitalni_nosice_sro
  digitalni_panely_cz
  kiwi
  obec_24
yadb-release ..............Obsahuje zdrojový kód hlavní aplikace.
yadb-joomla .....Obsahuje zdrojový kód základu webové úřední desky.
bakalarska_prace_bartak.pdf ............PDF verze bakalářské práce.
```

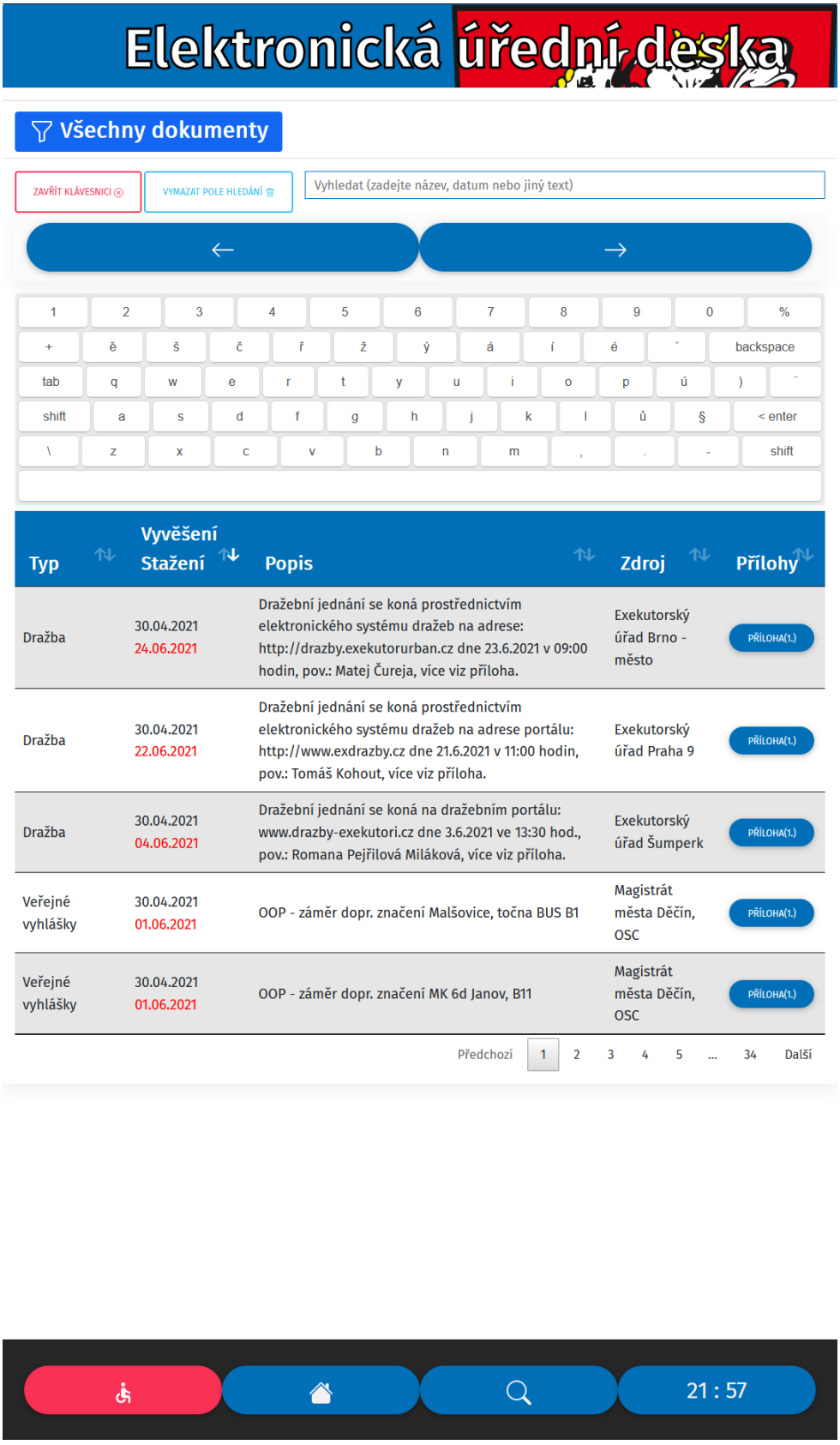
Příloha B

Obrázky

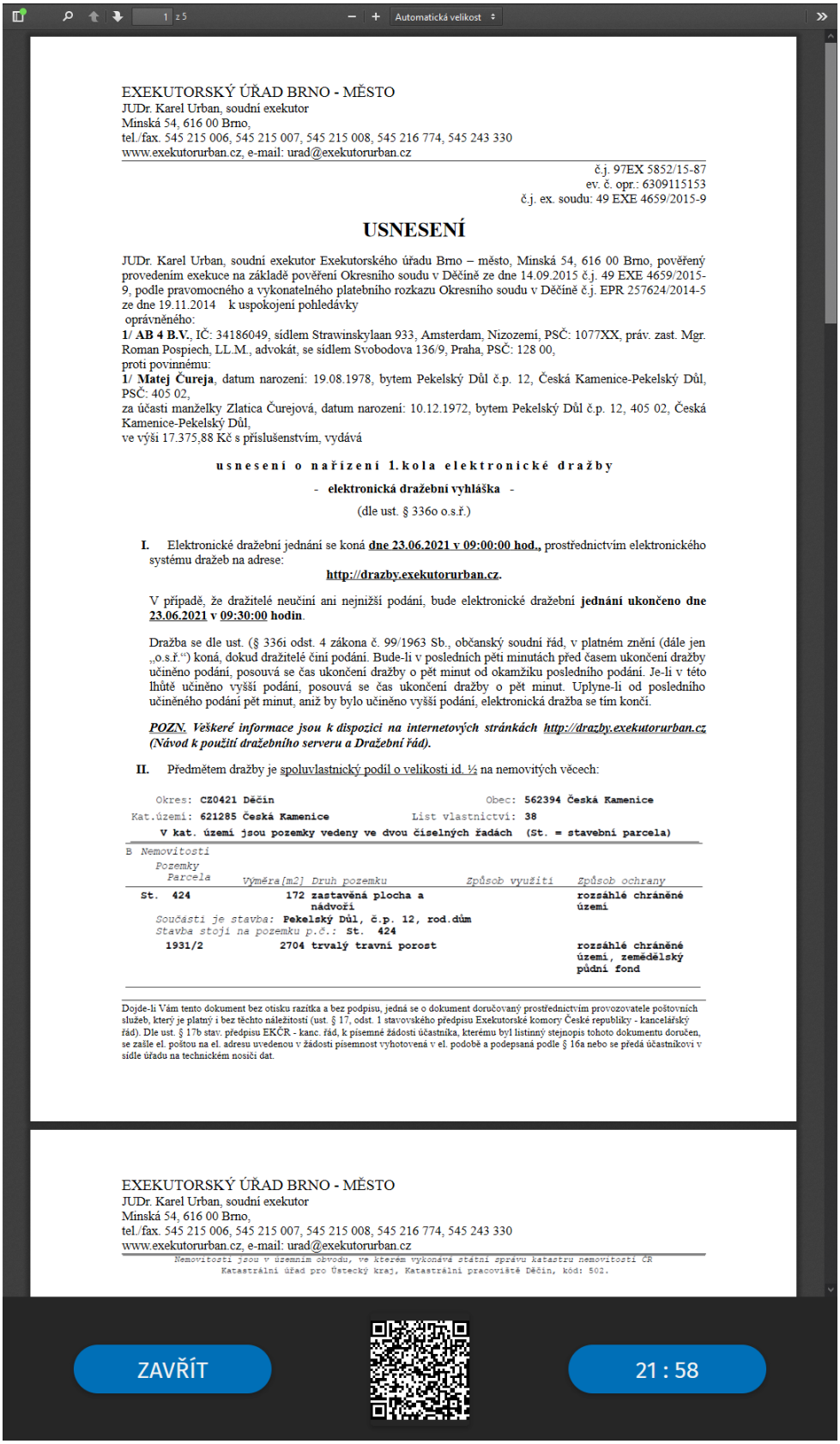
V této příloze lze nalézt obrázky, které jsou pro svou velikost nevhodné pro přímý vklad do hlavního textu práce. Jsou to výstřižky obrazovky zobrazující uživatelské rozhraní aplikace.



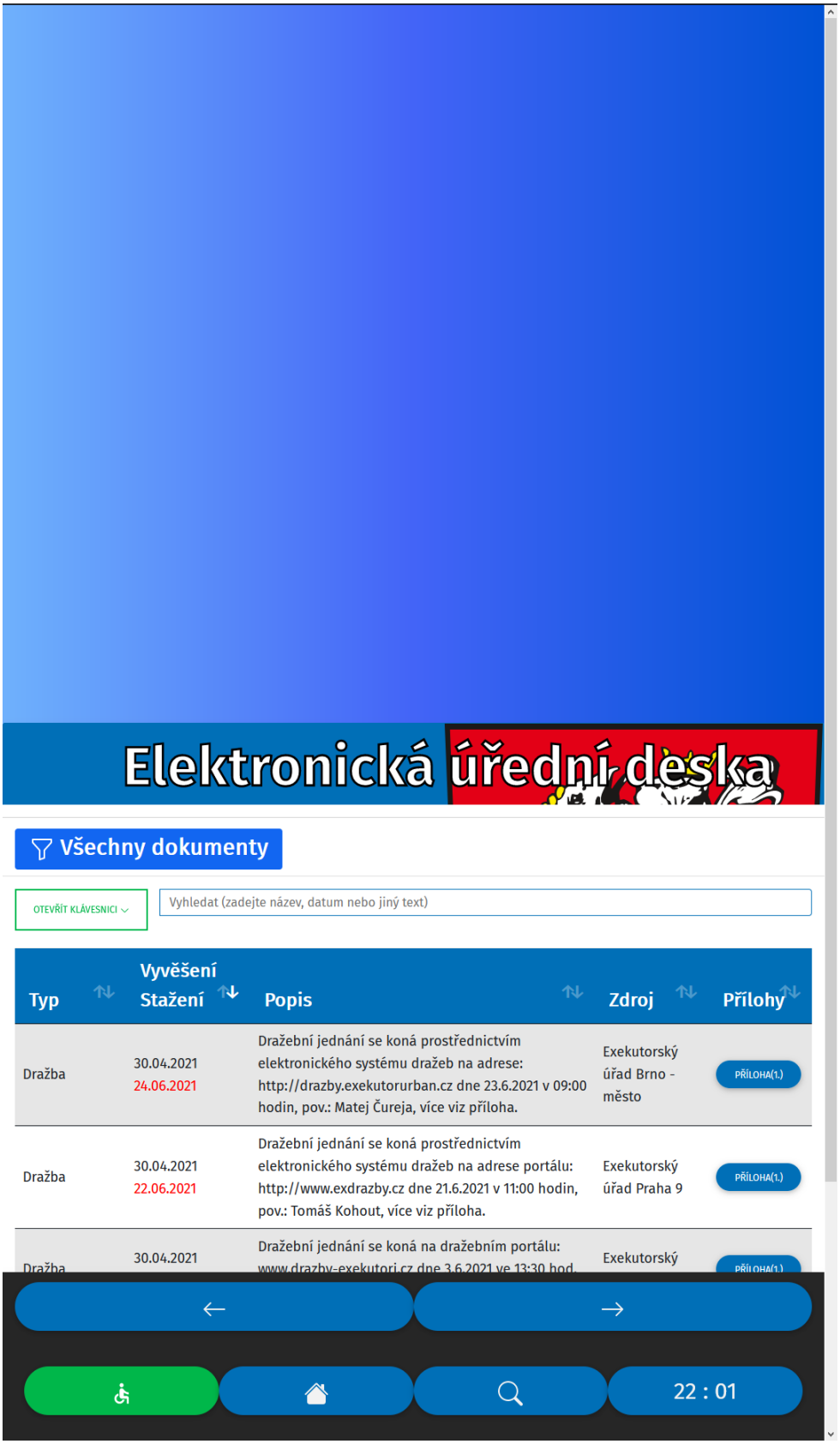
Obrázek B.1: Obyčejné procházení deskou – hlavní menu



Obrázek B.2: Obyčejné procházení deskou – úřední deska



Obrázek B.3: Obyčejné procházení deskou – PDF dokument



Obrázek B.4: Režim pro invalidy

Obrázek B.5: Správcovská část aplikace