

# Návrh a implementace modulů pro open-source platformu MagicMirror2 na Raspberry Pi

Richard Fous

---

Bakalářská práce  
2019



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2018/2019

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Richard Fous**

Osobní číslo: **A16345**

Studijní program: **B3902 Inženýrská informatika**

Studijní obor: **Informační a řídicí technologie**

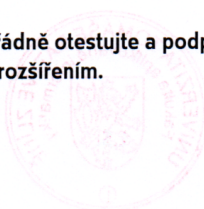
Forma studia: **prezenční**

Téma práce: **Návrh a implementace modulů pro open-source platformu MagicMirror2 na Raspberry Pi**

Téma anglicky: **The Design and Implementation of Modules for the MagicMirror2 Open-Source Platform for Raspberry Pi**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Popište manuální způsob instalace MagicMirror2 platformy na Raspberry Pi jako doplněk k automatické instalaci.
3. Analyzujte existující moduly pro platformu MagicMirror2 a navrhnete rozšíření tohoto seznamu o další funkční bloky.
4. Implementujte vybrané moduly na základě vývojové dokumentace platformy MagicMirror2.
5. Vytvořené moduly řádně otestujte a podpořte vývoj GitHub projektu MagicMirror2 jeho rozšířením.



Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. UPTON, Eben a Gareth HALFACREE. Raspberry Pi user guide. 4th edition. Chichester, West Sussex, UK: John Wiley, 2016. ISBN 978-1119264361.
2. PIPINELLIS, Achilleas. GitHub Essentials. Birmingham: Packt Publishing, 2015. ISBN 978-1783553716.
3. FLANAGAN, David. JavaScript: the definitive guide. 6th ed. Sebastopol, CA: O'Reilly, 2011. ISBN 978-0596805524.
4. MagicMirror2 Module Development Documentation [online]. [cit. 2018-11-19]. Dostupné z: <https://github.com/MichMich/MagicMirror/tree/master/modules>
5. 3rd Party Modules. MagicMirror2 [online]. GitHub [cit. 2018-11-19]. Dostupné z: <https://github.com/MichMich/MagicMirror/wiki/3rd-party-modules>
6. Electron Documentation. Electron [online]. [cit. 2018-11-19]. Dostupné z: <https://electronjs.org/docs/all>

Vedoucí bakalářské práce:

**Ing. Luboš Spaček**  
Ústav řízení procesů

Datum zadání bakalářské práce:

**21. prosince 2018**

Termín odevzdání bakalářské práce:

**15. května 2019**

Ve Zlíně dne 21. prosince 2018

doc. Mgr. Milan Adámek, Ph.D.  
*děkan*



prof. Ing. Vladimír Vašek, CSc.  
*ředitel ústavu*

### Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářské práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky. Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen přípouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně 6.5.2019

Richard Fous, v. r.

.....  
podpis autora



## **ABSTRAKT**

Cílem této bakalářské práce je navrhnout, vytvořit a implementovat rozšiřující moduly pro platformu MagicMirror2 a seznámit čtenáře s postupem instalace této platformy. Platforma MagicMirror2 je určena pro využití s počítači Raspberry Pi a operačním systémem Linux. V teoretické části práce jsou popsány jednotlivé prvky, které jsou důležité pro pochopení dané problematiky. V praktické části jsou popsány způsoby instalace platformy společně s návrhem a implementací nových modulů.

Klíčová slova: MagicMirror2, Raspberry Pi, Raspbian, JavaScript, HTML, CSS

## **ABSTRACT**

The aim of this bachelor thesis is to design, construct and implement extension modules for MagicMirror2 platform and introduce its further installation process. MagicMirror2 platform utilisation is proposed for Raspberry Pi computers with Linux operating system. The theoretical part of this study provides a description of the individual features important for the understanding of the overall issue. Several platform installation approaches, as well as a proposal and new modules implementation, are described in the practical part of the study.

Keywords: MagicMirror2, Raspberry Pi, Raspbian, JavaScript, HTML, CSS

Chtěl bych zde poděkovat vedoucímu práce panu Ing. Luboši Spačkovi za odbornou pomoc, cenné rady a za pomoc při kompletaci práce. Také bych chtěl poděkovat své rodině za veškerou podporu a pomoc při psaní této bakalářské práce a v průběhu celého studia.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

## OBSAH

ÚVOD .....	10
<b>I TEORETICKÁ ČÁST .....</b>	<b>10</b>
<b>1 PLATFORMA RASPBERRY PI .....</b>	<b>12</b>
1.1 MODELOVÁ ŘADA .....	12
1.1.1 Model A/B .....	12
1.1.2 Model A+/B+ .....	13
1.1.3 Raspberry Pi 2 B .....	13
1.1.4 Raspberry Pi 3 B/B+ .....	14
1.1.5 Raspberry Pi 4 .....	14
1.1.6 Raspberry Pi Zero .....	14
1.2 PŘIPOJENÍ PERIFERÍÍ .....	15
1.3 PŘÍSLUŠENSTVÍ .....	15
<b>2 OPERAČNÍ SYSTÉM RASPBIAN .....</b>	<b>17</b>
2.1 TVORBA INSTALAČNÍHO MÉDIA .....	17
2.1.1 NOOBS .....	18
2.1.2 Ostatní Linuxové Distribuce .....	18
2.2 INSTALACE OS RASPBIAN NA RASPBERRY PI .....	21
2.3 KONFIGURACE OS RASPBIAN .....	21
2.4 ZÁKLADNÍ INFORMACE PRO PRÁCI S OS RASPBIAN .....	22
2.4.1 Správce souborů .....	23
2.4.2 Správce doporučených programů .....	23
2.4.3 Systémové nastavení .....	23
2.4.4 Vypnutí zařízení .....	24
2.5 PŘEDINSTALOVANÝ SOFTWARE .....	24
<b>3 OPEN SOURCE SOFTWARE .....</b>	<b>26</b>
<b>4 GITHUB .....</b>	<b>27</b>
4.1 TVORBA REPOZITÁŘE .....	27
4.2 PŘÍKAZY .....	27
<b>5 WEBOVÉ TECHNOLOGIE .....</b>	<b>29</b>
5.1 HTML .....	29
5.2 CSS .....	30
5.3 JAVASCRIPT .....	30
5.3.1 DOM .....	31

5.3.2	JQuery.....	31
5.3.3	Node.js.....	31
5.3.4	Electron.....	32
5.4	API.....	32
5.5	CHROME DEVTOOLS.....	33
<b>6</b>	<b>PLATFORMA MAGICMIRROR2.....</b>	<b>34</b>
6.1	VZNIK PLATFORMY.....	34
6.1.1	Využité technologie.....	36
6.2	GITHUB MM2.....	36
6.2.1	Moduly.....	36
6.2.2	Dokumentace.....	37
6.3	FÓRUM.....	37
<b>II</b>	<b>PRAKTICKÁ ČÁST.....</b>	<b>37</b>
<b>7</b>	<b>INSTALACE A KONFIGURACE PLATFORMY MAGICMIRROR2 ...</b>	<b>39</b>
7.1	INSTALACE.....	39
7.1.1	Automatická instalace.....	39
7.1.2	Manuální instalace.....	39
7.2	MAGICMIRROR2 JAKO SERVER.....	40
7.3	KONFIGURACE.....	41
7.3.1	Automatické spuštění pomocí PM2.....	41
7.3.2	Spořič obrazovky.....	42
7.3.3	Natočení obrazovky.....	43
7.3.4	Konfigurační soubor.....	43
7.4	INSTALACE A KONFIGURACE MODULŮ.....	44
7.5	AKTUALIZACE.....	45
<b>8</b>	<b>ANALÝZA A NÁVRH MODULŮ.....</b>	<b>46</b>
<b>9</b>	<b>IMPLEMENTACE.....</b>	<b>47</b>
9.1	MODUL MMM-CZECHPOST.....	47
9.1.1	CSS.....	55
9.1.2	Překlady.....	56
9.1.3	Testování a Github.....	56
9.2	MODUL MMM-NAME DAY.....	57
9.2.1	CSS.....	63
9.2.2	Překlady.....	63
9.2.3	Testování.....	64



9.3	GitHUB.....	64
9.3.1	Dokumentace.....	65
<b>ZÁVĚR</b> .....		<b>66</b>
<b>SEZNAM POUŽITÉ LITERATURY</b> .....		<b>67</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK</b> .....		<b>70</b>
<b>SEZNAM OBRÁZKŮ</b> .....		<b>71</b>

## ÚVOD

Open source technologie jsou v dnešní době hojně využívány a jejich popularita každým dnem roste. Jednou z nejpobulárnějších platform vyžívaných v kombinaci s open source technologiemi je platforma Raspberry Pi, která je mezi uživateli oblíbená zejména kvůli své nízké ceně. V dnešní době zažívají velký rozmach také webové technologie, které můžeme najít téměř na každém kroku. Platforma MagicMirror2 vznikla jako open source projekt, který kombinuje webové technologie s výrobkem, bez kterého se v dnešní době člověk neobejde, se zrcadlem. Tato platforma vznikla za účelem zjednodušení každodenního života a nyní je mezi uživateli velmi populární.

Platforma MagicMirror2 je veřejnosti zpřístupněna pomocí projektu GitHub, kde je možné také nalézt seznam rošířujících modulů. Zobrazování informací na platformě je prováděno pomocí modulů, které vytvořili ostatní uživatelé této platformy. Každý modul je volně dostupný jako open source, a je tedy možné si jej přizpůsobit dle vlastních požadavků.

Tato bakalářská práce se zabývá návrhem a implementací rošířujících modulů a také podrobným popisem postupu instalace platformy MagicMirror2. V teoretické části je nastíněna problematika jednotlivých částí. V praktické části je popsán podrobný postup implementace nových modulů spolu se všemi možnými způsoby instalace této platformy.

# I. TEORETICKÁ ČÁST

## 1 Platforma Raspberry Pi

Raspberry Pi je malý a levný počítač o velikosti platební karty, který na rozdíl od standardních počítačů nebo notebooků obsahuje všechny potřebné komponenty na jedné tištěné desce. Má široké spektrum možných využití, lze jej využít pro automatizaci, jako náhradu tradičního počítače, učební pomůcku pro programování nebo automatizaci, ale také jako domácí server.

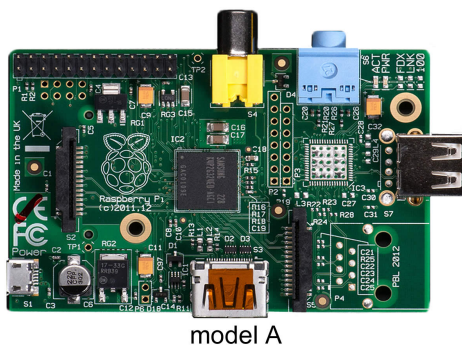
Platforma Raspberry Pi byla vytvořena za účelem rozšíření podpory výuky o počítačích po celém světě. Její tvůrci, kteří se sjednotili, aby vytvořili neziskovou organizaci Raspberry Pi Foundation netušili, jak rychle nabude na popularitě. V roce 2012 vydali prvních 1000 kusů počítačů, které byly ihned vyprodány. V následujících letech prodali milióny počítačů po celém světě. Tyto počítače si našli cestu do domácností, škol, kanceláří, datových center ale také i do továren. [1]

Všechny modely Raspberry Pi mají jednu společnou věc, jsou kompatibilní, tzn. program vytvořený pro jeden model bude fungovat na všech ostatních modelech. Je také možné vzít nejnovější operační systém Raspberry Pi a spustit ho na úplně první verzi Modelu B. Systém bude pracovat pomaleji, ale bude fungovat. [1]

### 1.1 Modelová řada

#### 1.1.1 Model A/B

Původní Raspberry Pi modely nesly názvy Model A (Obr 1.1) a Model B. Oba modely obsahovaly stejný Broadcom BCM2835 SoC, ale lišily se ve specifikacích. Model A disponoval 256MB paměti RAM a jedním USB portem. Model B disponoval 512MB RAM paměti, dvěma USB porty a na rozdíl od modelu A také portem pro Ethernet. Video výstup byl u obou modelů zajištěn pomocí HDMI nebo kompozitního videa s využitím RCA phono konektoru. Na rozdíl od novějších modelů obsahovaly modely A a B pouze 26 vstupně-výstupních pinů a pro úložiště využívaly standartní SD kartu.[2]

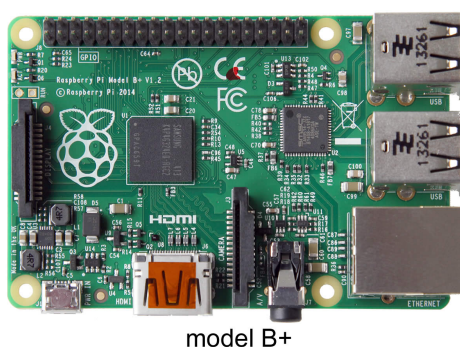


Obr. 1.1 Raspberry Pi model A [3]



### 1.1.2 Model A+/B+

Modely A a B byly velmi populární, ale byly rychle nahrazeny novým návrhem desky zvaným Plus. Z tohoto návrhu poté vznikly modely A+ a B+ (Obr 1.2), u kterých došlo ke zvýšení počtu vstupně-výstupních pinů na 40, přechodu na microSD úložiště a vylepšení některých funkcí oproti modelům A a B. U modelu A+ došlo ke zmenšení velikosti desky a zvýšení paměti RAM na 512 MB, zatímco u modelu B+ zůstala velikost RAM paměti stejná, došlo k navýšení počtu USB portů na 4. U obou modelů zůstal zachován video výstup HDMI, bylo přidáno rozhraní DSI a RCA Phono konektor byl spojen s 3,5 mm audio konektorem do jednoho 3,5mm konektoru. [2] [3]

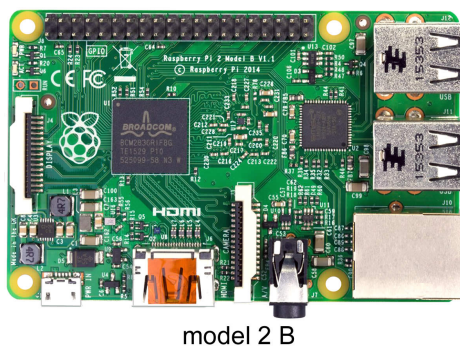


model B+

Obr. 1.2 Raspberry Pi model B+  
[3]

### 1.1.3 Raspberry Pi 2 B

Raspberry Pi 2 B (Obr 1.3) je nástupcem úspěšného modelu B+. Došlo k nahrazení BCM2835 SoC za BCM2836 SoC, který obsahuje 4 jádra. Tento nový SoC je 4-krát až 8-krát rychlejší než předchozí. Paměť RAM byla zvýšena na 1024 MB. Rozložení desky zůstalo až na pár drobností stejné jako u předchozího modelu. [2]



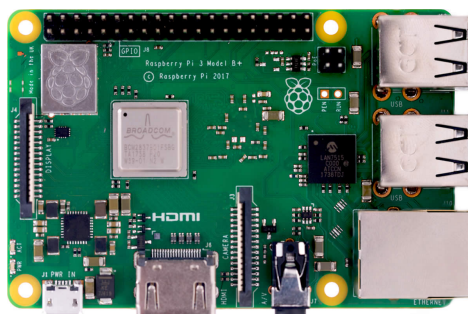
model 2 B

Obr. 1.3 Raspberry Pi model 2 B  
[3]

#### 1.1.4 Raspberry Pi 3 B/B+

Hlavní změnou u modelu 3 B bylo zabudování WiFi 802.11 b/g/n a Bluetooth 4.1 LE. Dále došlo ke změně procesoru z Broadcom BCM2836 SoC na BCM2837 SoC, který přinesl 64-bit podporu. [3]

Raspberry Pi 3 B+ (Obr 1.4) je nejnovějším zařízením z rodiny Raspberry Pi. Broadcom BCM2837 Soc byl nahrazen novějším BCM2837B0 SoC, u kterého došlo k navýšení frekvence procesoru na 1.4 GHz. Vylepšení se dočkal i WiFi modul, který nyní podporuje i standart IEEE 802.11ac. Změny nastaly i u Bluetooth modulu, který se zde nyní nachází ve verzi 4.2. Rozložení desky u modelů B a B+ zůstalo stejné jako u předchozích modelů. [2] [3]



model 3 B+

Obr. 1.4 Raspberry Pi model 3 B+ [3]

#### 1.1.5 Raspberry Pi 4

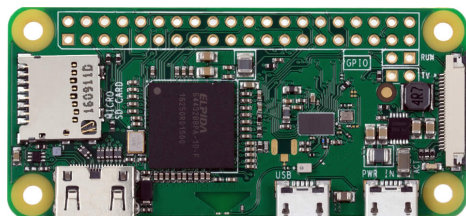
O nástupci Raspberry Pi 3 B+ bylo prozatím uvolněno pouze pár informací. Nástupce ponese název Raspberry Pi 4, velikostně nebude větší než model 3 B+ a cena se bude pohybovat kolem 35\$. Další novinkou by měl být přechod na čip s 28nm technologií, jelikož starší 40nm technologie nedokáže pracovat s více než 1 GB RAM. Vydání Raspberry Pi 4 je odhadováno na rok 2020. [4]

#### 1.1.6 Raspberry Pi Zero

Jedná se o nejmenší a nejlevnější Raspberry Pi. Je založeno na komponentech využitých pro Raspberry Pi model B+, obsahuje tedy Broadcom BCM2835 SoC a 512 MB RAM paměti. Kvůli své malé velikosti a také nízké ceně nedisponuje Ethernet portem a USB porty byly nahrazeny jedním micro USB portem. Velký HDMI port byl nahrazen mini-HDMI portem, 3,5mm AV konektor a rozhraní DSI byly zcela odstraněny. Dále také bylo odstraněno rozhraní CSI, které umožňovalo připojení externí kamery. [2] [3]

V roce 2016 byla vydána nová verze Raspberry Pi Zero nazvaná V1.3 ve které bylo

přidáno rozhraní CSI, jinak vše ostatní zůstalo stejné. V následujícím roce vyšla opět nová verze nazvaná W (Obr 1.5), která oproti verzi 1.3 obsahuje navíc modul pro WiFi 802.11 b/g/n a Bluetooth 4.1 LE. [3]



model Zero W

Obr. 1.5 Raspberry Pi Zero W [3]

## 1.2 Připojení periferií

Raspberry Pi bylo vytvořeno tak, aby bylo možné jej jednoduše a v co nejkratším čase sestavit a začít využívat, ale jako u všech počítačů je jeho funkčnost závislá na externích komponentech zvaných periferie. Mezi základní periferie potřebné pro využívání Raspberry pi patří [1]:

- Síťový adaptér s micro USB konektorem a výstupním proudem 2.5 A.
- MicroSD karta s nainstalovaným OS (viz kapitola 2.1).
- Klávesnici a myš s USB konektory.
- Monitor a k němu příslušný kabel s případnou redukcí.

K Raspberry Pi je možné napřímo připojit monitor, který disponuje HDMI konektorem. V ostatních případech je zapotřebí využít vhodný převodník nebo redukcí. Při nutnosti použití převodníku z HDMI na VGA není zapotřebí vlastnit externě napájený převodník, jelikož HDMI port na Raspberry Pi disponuje dostatečně velkým proudem pro interní napájení převodníku. [1]

Připojení síťového adaptéru by měl být poslední krok před spuštěním systému. Jelikož Raspberry Pi nemá fyzické tlačítko pro spuštění systému, tak po připojení do sítě dojde k automatickému zapnutí systému. [1]

## 1.3 Příslušenství

Nadace Raspberry Pi se nezabývá pouze vývojem a prodejem počítačů, ale nabízí také vlastní řadu příslušenství [2]:

- Univerzální síťový adaptér, který je určen pro všechny modely Raspberry Pi.

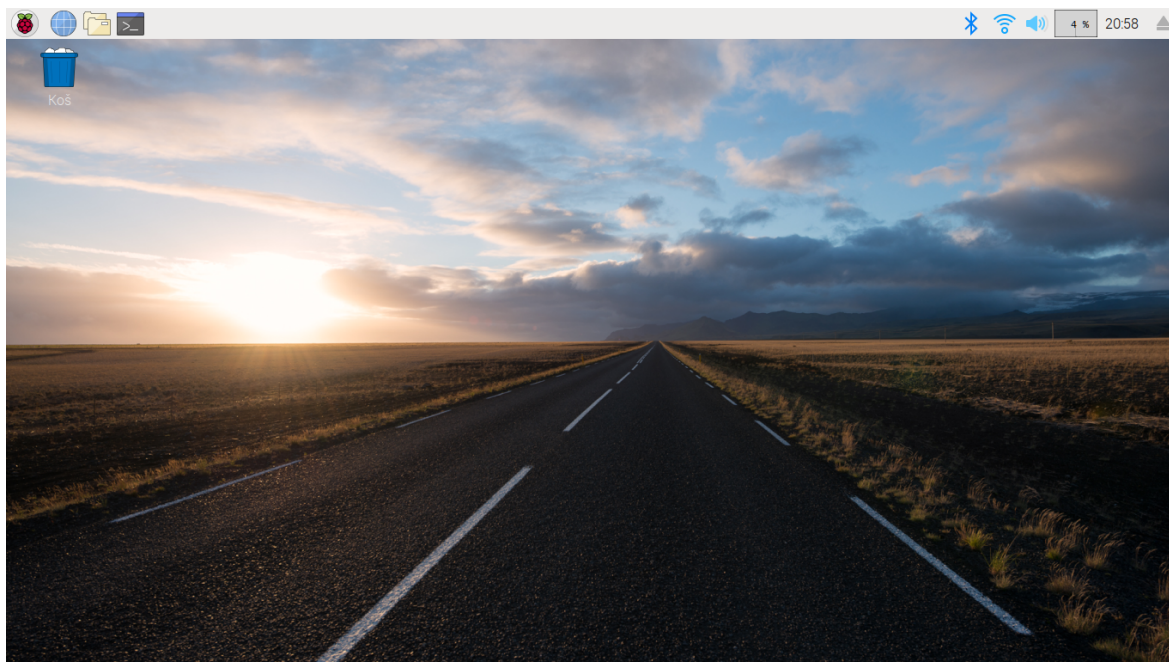
- 
- Externí WiFi adaptér, který je možné připojit pomocí USB portu a je určen pro všechny modely, které neobsahují WiFi.
  - Sada adaptérů pro Raspberry Pi Zero, která obsahuje redukce z micro USB na USB a z mini-HDMI na HDMI.
  - Kamerový modul, který je možné k Raspberry Pi připojit pomocí rozhraní CSI. Tento modul dokáže nahrávat videozáznamy a pořizovat fotografie.
  - Dotykový display - Tento display je k Raspberry Pi připojen pomocí rozhraní DSI.
  - Sense HAT - multifunkční deska, která rozšiřuje Raspberry Pi o poziční senzor (gyroskop, akcelerometr, magnetometr) a o senzory teploty, tlaku a vlhkosti. Dále obsahuje 64 LED, které slouží jako výstup a 5-ti poziční ovladač.



## 2 Operační systém Raspbian

Raspbian je upravená verze populární linuxové distribuce Debian. Debian je jedna z nejdéle fungujících, nejstabilnějších a nejlépe odladěných distribucí linuxu, což z ní dělá perfektního kandidáta pro použití s Raspberry Pi. Raspbian přebírá klíčové funkce Debianu, ke kterým byly následně přidány speciální funkce a software tak, aby bylo využívání Raspbianu co nejjednodušší. Raspbian obsahuje grafické prostředí zvané Lightweight X11 Desktop Environment (Obr 2.1), které má podobnou funkčnost jako prostředí např. v OS Windows nebo Mac OSX. [1] [2]

Jedním z hlavních kritérií při vývoji Raspbianu bylo, aby OS zabíral co nejméně místa na microSD kartě. Z tohoto důvodu Raspbian obsahuje jen pár předinstalovaných programů, které umožňují například prohlížení internetu a programování v Pythonu. Případná instalace dodatečných programů je možná pomocí manažeru balíčků atp.[2]



Obr. 2.1 Raspbian OS

### 2.1 Tvorba instalačního média

Nadace Raspberry Pi prodává oficiální balíčky, ve kterých se nachází kromě Raspberry Pi i SD karta s předinstalovaným NOOBS. S takto předinstalovanou kartou je možné rovnou začít s instalací OS na Raspberry Pi a není nutné toto bootovací zařízení vytvářet. Pokud je zapotřebí vytvořit bootovací microSD kartu, je možné si vybrat ze dvou možností. Je možné nainstalovat NOOBS (viz kapitola 2.1.1), který výrazně ulehčí postup instalace OS, ale NOOBS obsahuje pouze pár předinstalovaných OS. Druhou

možností je vytvořit bootovací médium bez NOOBS přímo s vybraným OS (viz kapitola 2.1.2), tato možnost je o něco složitější. Seznam podporovaných OS je možné najít na webové stránce [www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads). [2]

### 2.1.1 NOOBS

K instalaci NOOBS bude zapotřebí prázdná microSD karta, redukce na SD kartu a počítač, který umí číst SD karty. Dále je potřeba stáhnout samotný NOOBS, který je možné nalézt na webové adrese [rpf.io/downloads](http://rpf.io/downloads), zde je zapotřebí kliknout na odkaz NOOBS se znakem Raspberry Pi a poté kliknout na tlačítko Download ZIP u části "NOOBS Offline and network install". [1]

Nyní je zapotřebí připojit microSD kartu do počítače a naformátovat ji. Uživatelé OS Windows a MacOS mohou využít program pro formátování SD karet, který je dostupný na webové adrese [rpf.io/sdcard](http://rpf.io/sdcard). Po stažení programu je zapotřebí jej nainstalovat. Pokud je program nainstalovaný a microSD karta vložená v počítači, je možné začít s formátováním. Prvním krokem je v programu SD Card Formatter vybrat v políčku "Select card" kartu, kterou chceme naformátovat. V seznamu, který se objeví pod tímto polem, je zapotřebí zkontrolovat, zdali je vybrána správná microSD karta. Po zkontrolování je nutné kartu pojmenovat vepsáním "NOOBS" do políčka "Volume label". Posledním krokem je kliknutí na tlačítko Format, které naformátuje kartu. Po dokončení formátování již program nebude nadále zapotřebí, a je tedy možné jej uzavřít. [1]

Uživatelé OS linux mohou využít program na správu disků, který se nachází v jimi využívané distribuci. V tomto programu je zapotřebí smazat všechny existující oddíly na kartě a poté vytvořit jeden nový oddíl ve formátu VFAT. [1]

V dalším kroku je zapotřebí zkopírovat stažené soubory na microSD kartu. Stažený NOOBS soubor je zabalen v archivu, který obsahuje vše potřebné pro spuštění NOOBS. Stažený archiv je zapotřebí otevřít a vybrat všechny položky pomocí klávesové zkratky CTRL+A nebo Command+A na MacOS. Takto vybrané soubory je poté možné pomocí levého tlačítka myši přetáhnout do adresáře microSD karty. Tím se soubory extrahují a přepokopírují na microSD. Po dokončení kopírování je možné microSD kartu odpojit z počítače a připojit ji do Raspberry, kde se při prvním zapnutí spustí instalační průvodce NOOBS. [1]

### 2.1.2 Ostatní Linuxové Distribuce

Manuální instalace je komplikovanější procedura oproti instalaci NOOBS, ale v některých případech se jedná o lepší řešení, jelikož NOOBS nemusí obsahovat preferovanou distribuci Linux nebo může obsahovat zastaralé verze. [2]

Jedním z prvních kroků instalace je výběr distribuce nebo jiného OS, které bude nejlépe odpovídat požadavkům uživatele. Pokud uživatel nebude spokojen s vybraným OS, je možné kdykoliv OS změnit pomocí opětovné instalace. Seznam všech dostupných OS pro Raspberry Pi je možné nalézt na webové adrese [www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads). Nadace Raspberry Pi nabízí možnost stažení jednotlivých OS pomocí BitTorrentu. Jedná se o malé soubory, které je možné využít s BitTorrent programy pro stáhnutí dat od ostatních uživatelů. Aby tedy bylo možné tyto soubory stáhnout, je zapotřebí BitTorrent program, jako např. uTorrent, který funguje s operačními systémy Windows, OS X a Linux a je dostupný z [www.utorrent.com/downloads](http://www.utorrent.com/downloads). [2]

Distribuce pro Raspberry Pi jsou dodávány v jedné archivu. Tento archiv je nutné rozbalit, aby bylo možné s ním dále pracovat. Archiv obsahuje 2 soubory. Jeden ze souborů je hash soubor s příponou `.sha1`, který slouží k překontrolování, zda při stahování nedošlo k chybě a zda jsou staženy všechny soubory. Druhý soubor s příponou `.img` obsahuje kopii obsahu SD karty, kterou vytvořil vydavatel této distribuce tak, aby jí Raspberry Pi rozumělo. Tento soubor je zapotřebí nahrát na microSD kartu. [2]

## Linux

Při tvorbě instalačního média na platformě linux je možné využít příkazu `dd` pro zápis obsahu `.img` souboru na microSD kartu. Tento příkaz je zadáván pomocí příkazového řádku, v Linuxu zvaného "terminál". Pomocí následujících kroků je možné vytvořit instalační médium [2]:

1. Otevřete terminál.
2. Připojte prázdnou microSD kartu pomocí redukce do počítače.
3. Do terminálu vepište příkaz **sudo fdisk -l**, který zobrazí seznam všech připojených disků. Podle velikosti vaší microSD karty najděte a označte si název zařízení - `/dev/sdX`, kde X identifikuje vaši microSD kartu.
4. Pomocí **cd** příkazu přejděte do složky, kde se nachází vyextrahovaná požadovaná distribuce.
5. Příkaz **sudo dd if=názevImageSouboru.img of=/dev/sdX bs=2M** vepište do terminálu. Tento příkaz slouží k zápisu `.img` souboru na microSD kartu. Nahraďte `názevImageSouboru.img` názvem souboru, který byl vyextrahován ze staženého archivu. Tento krok trvá delší dobu. Po ukončení nahrávání se v terminálu zobrazí hláška, která vás bude informovat o úspěšném vytvoření.

## OS X

Postup tvorby instalačního média u OS X je podobný jako u OS Linux [2]:

1. V seznamu aplikací otevřete složku Utilities. V této složce poté otevřete aplikaci Terminál.
2. Připojte prázdnou microSD kartu pomocí redukce do počítače.
3. Do terminálu vepište příkaz **diskutil list**, který zobrazí seznam všech připojených disků. Podle velikosti vaší microSD karty najdete a poznačte si název zařízení - /dev/diskX, kde X identifikuje vaši microSD kartu.
4. Pokud se microSD karta automaticky připojila a její ikona se objevila na ploše, je nutné ji odpojit pomocí příkazu **diskutil unmountdisk /dev/diskX**.
5. Pomocí **cd** příkazu přejděte do složky, kde se nachází vyextrahovaná požadovaná distribuce.
6. Příkaz **sudo dd if=názevImageSouboru.img of=/dev/diskX bs=2M** přepište do terminálu. Tento příkaz slouží k zápisu .img souboru na microSD kartu. Nahraďte názevImageSouboru.img názvem souboru, který byl vyextrahován ze staženého archivu. Tento krok trvá delší dobu. Po ukončení nahrávání se v terminálu zobrazí hláška, která vás bude informovat o úspěšném vytvoření.

## Windows

V OS Windows je postup tvorby instalačního média odlišný od předchozích OS. V tomto OS nenajdeme příkaz dd, a proto je nutné využít volně dostupné aplikace Image Writer for Windows z webové stránky <https://sourceforge.net/projects/win32diskimager>. Dále je nutné pokračovat pomocí následujících kroků [2]:

1. Stažený program pro tvorbu instalačního média je nutné extrahovat z archivu do složky.
2. Připojte prázdnou microSD kartu pomocí redukce do počítače.
3. Dvojklikem otevřeme soubor Win32DiskImager.exe a klikem na modrou ikonu složky otevřeme průvodce výběrem souboru.
4. Nalezneme námi extrahovaný .img soubor a klikneme na tlačítko Open.
5. Z menu dostupných zařízení vybereme naformátovanou microSD kartu a klikneme na tlačítko Write, který započne zápis na microSD kartu.



## 2.2 Instalace OS Raspbian na Raspberry Pi

Po vložení microSD karty s čerstvě nainstalovaným NOOBS a zapnutí Raspberry Pi se zobrazí obrazovka načítání s logem Raspberry Pi, která po pár minutách zmizí. Po tomto úvodním načtení se zobrazí menu NOOBS, ve kterém je možné zvolit OS, který bude nainstalován. Originální verze NOOBS obsahuje defaultní OS: Raspbian a Debian Linux, který je speciálně upraven pro použití s Raspberry Pi. Pokud je Raspberry Pi připojeno k internetu, je možné dodatečně stáhnout jiný OS. [1]

K započítí instalace je nutné zvolit požadovanou verzi OS Linux pomocí políčka, které se nachází v levé části seznamu. Po zvolení OS se nám objeví možnost startu instalace v horním menu, která byla doposud zašedlá. Po kliknutí na tuto ikonu vyskočí varovná zpráva, která uživatele varuje, že započítím instalace dojde k přepsání dosavadních dat na microSD kartě kromě NOOBS, který zůstane nedotčený. Po potvrzení této skutečnosti pomocí tlačítka **Yes** dojde k instalaci vybraného OS. Instalace OS může trvat v rozmezí 10 až 30 minut v závislosti na rychlosti microSD karty. Průběh instalace je zobrazován ve spodní části NOOBS menu. Po dokončení instalace vyskočí okno s tlačítkem **OK**, které uživatele informuje o dokončení instalace a nutnosti restartu zařízení. Po stisknutí tlačítka **OK** dojde k restartování Raspberry Pi. Po restartu zařízení se na obrazovce objeví černá obrazovka s rychle se měnícím textem. V tomto momentě dochází k prvotnímu samonastavení, kdy OS zjišťuje volné místo na microSD kartě a jak s ním bude v budoucnu zacházet. Po dokončení tohoto procesu se na obrazovce objeví základní obrazovka nainstalovaného OS a tím je proces instalace dokončen. Nyní je možné začít s konfigurací zařízení (viz kapitola 2.3). [1]

Pokud uživatel zvolil manuální instalaci OS, je postup téměř totožný s tím rozdílem, že po prvotním spuštění Raspberry Pi dojde přímo ke spuštění instalačního průvodce dané distribuce. Při této instalaci bude po uživateli požadováno vytvoření uživatelského účtu a dalších nezbytných věcí. Po dokončení instalace opět dojde k restartu zařízení a poté k načtení OS. [1]

## 2.3 Konfigurace OS Raspbian

Po instalaci a prvotním restartu se na ploše objeví okno, ve kterém je možné provést konfiguraci OS. Na úvodní obrazovce je uživatel vítán do OS Raspbian, kde je možné kliknout na tlačítko **Next** pro pokračování v konfiguraci. V následujícím okně dochází k nastavení země, jazyka a časového pásma. Po nastavení těchto tří možností je opět možné kliknout na tlačítko **Next**, které uživatele posune na konfiguraci hesla. [1]

Zde si uživatel může zvolit heslo pomocí kterého se bude přihlašovat do OS. Pokud uživatel nevyplní žádné heslo, zůstane defaultně nastavené heslo "raspberrypi". Ovšem je doporučeno defaultní heslo změnit, aby bylo zamezeno přístupu nepovolaných osob do

systemu. Po vepsání a ověření nového hesla je možno pokračovat kliknutím na tlačítko **Next**. [1]

V následujícím kroku dochází k nastavení připojení k síti WiFi. Pokud je již zařízení připojené k síti pomocí Ethernetu, nebo si uživatel nepřeje být připojen k internetu, je možné tento krok přeskočit pomocí tlačítka **Skip**. V opačném případě je možné v seznamu vybrat jméno sítě, která patří uživateli a poté kliknout na tlačítko **Next**. Zde bude uživatel požádán o zadání hesla pro připojení k této síti. [1]

V předposledním kroku konfigurace je možné zkontrolovat, zdali nejsou dostupné aktualizace systému. Pokud uživatel není připojen k internetu a nebo si nepřeje provést kontrolu aktuálnosti systému, je možné tento krok přeskočit pomocí tlačítka **Skip**. Ovšem je doporučeno udržovat OS na nejnovější verzi, jelikož s každou novou verzí dochází k opravám chyb, které mohou mít zásadní dopad na bezpečnost a stabilitu systému. Pro provedení této kontroly je nutné kliknout na tlačítko **Next**. Stažení aktualizací může trvat až několik minut. [1]

Po stažení aktualizací dojde k přesunu na poslední stranu konfigurace, kde je po uživateli požadováno restartování zařízení pomocí tlačítka **Reboot**. Po tomto restartu je dokončena základní konfigurace a uživatel může zařízení plně využívat. [1]

## 2.4 Základní informace pro práci s OS Raspbian

Při spuštění Raspberry Pi dojde k načtení plochy (Obr 2.1). V horní části obrazovky se nachází menu, pomocí kterého je možné přistupovat k jednotlivým funkcím operačního systému.

V pravé části tohoto menu se nachází ikona Bluetooth, která po rozkliknutí umožní uživateli připojení k jinému bluetooth zařízení. Vedle se nachází ikona, která zobrazuje intenzitu signálu připojené WiFi sítě. Pokud je uživatel připojen pomocí ethernetu, zobrazí se místo ikony signálu ikona dvou proti sobě směřujících šipek. Nastavení úrovně hlasitosti lze provést pomocí následujícího tlačítka s ikonou reproduktoru, kde se po rozkliknutí objeví posuvník hlasitosti. Vedle ikony hlasitosti se nachází ukazatel vytíženosti procesoru. Další položkou na tomto menu je zobrazení aktuálního času, kde po kliknutí na tento čas dojde k zobrazení zmenšeného kalendáře, který ukazuje aktuální datum. Poslední položkou je možnost zobrazení a bezpečného odpojení připojených externích paměťových zařízení. [1]

V levé části horního menu se nachází ikona Raspberry Pi, která po rozkliknutí zobrazí roztríděný seznam všech dostupných programů a také tlačítko pro bezpečné vypnutí počítače. V této liště se dále nachází zkratky pro rychlejší spuštění programů. Pořadí umístění těchto zkratk je možné upravit delším stiskem levého tlačítka myši na příslušné ikoně a poté za stálého držení levého tlačítka tuto ikonu přesunout. Stej-

ným způsobem je možné přidávat nové zkratky ze seznamu programů. Pro odstranění některé zkratky je nutné na příslušné ikoně kliknout pravým tlačítkem myši a v zobrazené nabídce zvolit možnost odstranění. Vedle těchto zkratek se nachází hlavní panel, ve kterém se zobrazují aktuálně běžící programy. [1]

#### 2.4.1 Správce souborů

Správce souborů umožňuje uživateli přistupovat ke všem souborům, které se nacházejí na microSD kartě, ale také na ostatních externích úložných zařízeních. Při spuštění správce souborů se zobrazí domácí složka, která obsahuje další podsložky, jako například dokumenty, stažené soubory, muzika, obrázky a videa. [1]

Správce souborů je rozdělen na dvě části. V levé části se nachází hlavní složky a případné externí úložiště. V pravé části se nachází jednotlivé soubory a podsložky.

Soubory z externích úložišť mohou být jednoduše překopírovány do Raspberry Pi, stejně tak je možné překopírovat soubory z Raspberry Pi na externí úložiště. Kopírování je možné provést dvěma způsoby. První možnost je otevřením druhého okna a přetažením požadovaných souborů z jednoho okna do druhého. Druhá možnost je klikem pravého tlačítka myši na souboru nebo složce vyvolat menu a poté vybrat možnost kopírovat nebo vyjmout. V cílovém adresáři je poté možné stejným způsobem otevřít vyskakovací menu a zvolit možnost vložit. Namísto využití tlačítek myši je možné využít klávesových zkratk `CTRL+C` pro zkopírování nebo `CTRL+X` pro vyjmutí a poté v cílovém adresáři vložit soubory zkratkou `CTRL+V`. [1]

#### 2.4.2 Správce doporučených programů

Správce doporučených programů obsahuje seznam dostupných programů, které je možné doinstalovat nebo odinstalovat. Tento správce vyžaduje připojení k internetu, jelikož programy, které se zde nacházejí, jsou stahovány z internetu. V levé části tohoto správce se nachází seznam kategorií, do kterých jsou jednotlivé programy rozděleny. V pravé části se nachází seznam programů. Pokud se vedle nějakého programu nachází ikona křížku, je již tento program nainstalován. Odkliknutím této ikony je možné program odinstalovat. [1]

Instalaci nového programu je možné provést zakliknutím výše zmiňované ikony a poté kliknutím na tlačítko OK spustit instalaci. Po dokončení instalace se zobrazí okno informující o dokončení instalace. [1]

#### 2.4.3 Systémové nastavení

Nastavení systému se nachází v záložce nastavení pod ikonou Raspberry Pi na horní liště. Okno nastavení je rozděleno do čtyř částí. V záložce "Systém" je možné změnit

heslo k přístupu do systému, jméno, pod kterým bude zařízení vystupovat v lokální síti a mnoho dalších. Většina možností v této záložce by ovšem měla zůstat na výchozích hodnotách. [1]

Další částí je záložka "Rozhraní", kde se nachází větší množství možností, ale všechny tyto možnosti jsou ve výchozím nastavení vypnuté. Většina těchto nastavení je určena pro rozšiřující moduly, které je možné k zařízení připojit a proto není doporučeno s nimi manipulovat. Výjimkou jsou pouze možnosti "Secure Shell"(SSH), "Virtual Network Computer"(VNC) a "Remote GPIO". Pomocí SSH je možné se k Raspberry Pi připojit z jiného počítače v síti s využitím SSH klienta. VNC umožňuje připojení jiného počítače v síti k Raspberry Pi, připojený počítač může toto zařízení vzdáleně ovládat. "Remote GPIO"slouží k vzdálenému ovládání vstupně-výstupních pinů na Raspberry Pi. [1]

Předposlední záložkou v nastavení je záložka "Výkon", která slouží k nastavení množství paměti dodávané grafickému procesoru a u některých modelů je možné zde i přetaktovat procesor pro dosažení vyššího výkonu. Pokud uživatel nepotřebuje vyšší výkon, je doporučeno tato nastavení ponechat na původních hodnotách. [1]

Poslední záložkou je "Lokalizace", kde je možné změnit nastavení jazyka, možnosti zobrazování času, jednotek a rozložení klávesnice. [1]

#### 2.4.4 Vypnutí zařízení

Stějně jako všechny ostatní počítače i Raspberry Pi uchovává potřebná data ve volatilní paměti, která potřebuje neustálý přísun napájení. Proto je velice důležité nevypínat počítač odpojením napájecího kabelu, ale pomocí tlačítka **Shutdown** neboli vypnout, které se nachází pod ikonou Raspberry Pi na horní liště. Tímto způsobem je zajištěno, že nedojde k žádné ztrátě důležitých souborů, které by mohly být odstraněny a OS by nemusel při příštím zapnutí naběhnout. Úplné vypnutí zařízení je možné poznat tak, že zelená dioda na zařízení přestala svítit. Poté je možné odpojit zdroj napájení. [1]

Pro opětovné zapnutí je nutné znovu připojit zařízení ke zdroji napájení.

## 2.5 Předinstalovaný software

Raspbian v základu obsahuje předinstalované programy, které jsou potřebné jak pro klasické využívání počítače, tak i pro základní programování. Jedním z programů pro programování je Geany, který je určen pro různé programovací jazyky a je populární volbou mezi uživateli. Dále se zde nachází program pro grafické programování Scratch, který je určen pro mladší uživatele. [2]

Pro rozvoj výuky programování mezi mladými uživateli se zde nachází upravená verze populární hry Minecraft zvaná "Minecraft Pi", která umožňuje uživatelům ovlá-

dat hru pomocí programů, které napsali např. v jazyce Python. [2]

Pro rozšíření produktivity se zde nachází balíček programů "Libreoffice base", který obsahuje základní programy pro psaní textů, tvorbu prezentací a tabulek. Dalším důležitým programem je "Chromium Web Browser", který je založen na populárním prohlížeči Chrome a slouží k prohlížení webu. Mezi předinstalované programy lze nalézt i program "PDF Viewer", který slouží k prohlížení PDF dokumentů a "Text editor", který slouží např. k zapisování poznámek. [2]

### 3 Open source software

Open source software je software s veřejně dostupným zdrojovým kódem, který může kdokoliv využít, upravit nebo vylepšit. Tento software je vyvíjen pod některou z mnoha licencí Open Source Initiative (OSI). Získání této licence může být poskytováno zdarma, ale také může být zpoplatněné. [5]

Zdrojový kód je část programu, kterou většina uživatelů nikdy neuvidí. Programátoři, kteří mají přístup ke zdrojovému kódu, ho mohou různými způsoby upravovat a tím jsou schopni přidat různá vylepšení nebo opravit chyby, které se v kódu nachází. [5]

Mezi open source software patří např. většina Linuxových distribucí, LibreOffice, Mozilla Firefox a Thunderbird, ale také webové technologie jako např. PHP, Node.js a Bootstrap. [5]

## 4 GitHub

GitHub je webový projekt, který umožňuje vývojářům ukládat a spravovat jejich zdrojové kódy pomocí open source technologie Git. [6]

Git je systém pro správu verzí softwaru. Při vývoji programů nebo aplikací dochází neustále k různým úpravám kódu a k vytváření nových verzí. Git je prostředek, který umožňuje snazší spolupráci vývojářů při tvorbě programů, jelikož každý vývojář si může stáhnout aktuální verzi kódu, provádět úpravy a také nahrát nejnovější verzi. Každý vývojář může sledovat nově provedené změny a stáhnout si je. [6]

Git pracuje s příkazovým řádkem, a proto není vhodný pro začátečníky. Z tohoto důvodu vznikl Github, který zpřístupnil technologii Git i méně zdatným uživatelům. GitHub je pro svoji jednoduchost využíván i pro ostatní typy projektů, jako například psaní knih. [6]

### 4.1 Tvorba repozitáře

Pro tvorbu repozitáře je nutné mít vytvořený účet, který je možné vytvořit na webové stránce <https://github.com/join>. Po přihlášení se v pravém horním rohu objeví ikona plus, pomocí které je možné přidat repozitář. [7]

Prvním krokem je vepsání jména repozitáře, pomocí kterého bude následně vytvořen webový odkaz pro tento repozitář. V dalším kroku je možné přidat popis repozitáře, který bude ostatní uživatele informovat čím se tento repozitář zabývá. Dále je možné zvolit, jestli repozitář bude volně přístupný, nebo privátní. K repozitáře je také možné připojit README soubor, který obsahuje důležité informace týkající se například instalace programu nebo řešení případných problémů. K repozitáři je také možné připojit licenci, která bude stanovovat za jakých podmínek mohou ostatní uživatelé program využívat. [7]

### 4.2 Příkazy

Níže uvedený seznam obsahuje nejdůležitější příkazy, které jsou využívány při práci s platformou GitHub a ukázky jejich využití [8]:

- `git add` - Umožňuje specifikovat soubory k nahrání.
  - ◊ `git add nazevSouboru.txt`
- `git branch` - Slouží k vytvoření nové větve. Vytvoří kopii aktuální větve, včetně všech provedených změn.
  - ◊ `git branch nova_vetev`



- `git clone` - Slouží k naklonování již existujícího repozitáře.
  - ◊ `git clone webova_stranka`
- `git commit` - Slouží k porovnání již existujících souborů v repozitáři s těmi, které mají být nahrány.
  - ◊ `git commit -a "Informace o commitu"`
- `git config` - Umožňuje nastavení emailu, přezdívky, formátu souborů atd.
  - ◊ `git config user.email "email@email.cz"`
- `git fetch` - Umožňuje stažení všech souborů ze vzdáleného repozitáře, které se nenacházejí v lokálním repozitáři.
  - ◊ `git fetch origin`
- `git checkout` - Umožňuje přepínání mezi větvemi.
  - ◊ `git checkout -b <nazev_vetve>`
- `git init` - Slouží k vytvoření nového lokálního repozitáře.
- `git log` - Výpis historie daného repozitáře.
- `git merge` - Slouží ke spojení větví.
  - ◊ `git merge <nazev_vetve>`
- `git pull` - Slouží ke sloučení již existujícího vzdáleného repozitáře s lokálním.
- `git push` - Umožňuje odeslání souborů do repozitáře.
  - ◊ `git push origin master`
- `git remote` - Umožňuje připojení ke vzdálenému repozitáři, který může být uložen např. na serveru.
  - ◊ `git remote -v`
- `git status` - Zobrazí seznam všech souborů, ve kterých byly provedeny změny.

## 5 Webové technologie

Pojem webové technologie značí několik na sobě závislých programovacích jazyků, které jsou v dnešní době potřebné pro tvorbu moderních dynamických webových stránek. Každá technologie je sama o sobě dosti omezená, a proto většinou vyžaduje několik dalších technologií pro správnou funkčnost. [9]

### 5.1 HTML

Hypertext Markup Language (HTML) je název značkovacího jazyka, který je využíván pro tvorbu webových stránek. HTML jazyk je tvořen pomocí značek a jejich atribut. Webový prohlížeč obdrží HTML dokument z webového serveru nebo z lokálního úložiště a zobrazí dokument jako webovou stránku. [10]

Jednotlivé názvy značek a vlastnosti jim příslušící se píší do úhlových závorek. Většina HTML je párových, přičemž koncová značka je stejná jako úvodní, ale před názvem obsahuje lomítko. [10]

HTML dokument má předepsanou strukturu, kterou je důležité dodržovat [10] :

- Deklarace typu dokumentu - pomocí značky `<DOCTYPE html>!` je prohlížeči sděleno, že otevíraný dokument je ve formátu HTML.
- Kořenový element : Začíná a ukončuje celý dokument. Je označen značkami `<html>` a `</html>`.
- Hlavička dokumentu: Obsahuje data, která se vztahují k celému HTML dokumentu jako například titulek stránky, skripty a využití styly. Využívá značek `<head>` a `</head>`.
- Tělo dokumentu: Značky `<body>` a `</body>`. Obsahuje vše, co má být vykresleno na výsledné webové stránce.

Značky lze dělit na tři základní skupiny [10] :

- Strukturální značky - dodávají dokumentu strukturu např.: odstavce (`<p>`) nebo nadpisy (`<h1>` až `<h6>`).
- Popisné značky - popisují povahu obsahu prvku. Například nadpis (`<title>`) nebo adresa (`<address>`).
- Stylistické značky - určují vzhled při výsledném zobrazení ve webovém prohlížeči. Postupně ale tyto značky začínají nahrazovat kaskádové styly.

## 5.2 CSS

Kaskádové styly (CSS) je název jazyka, pomocí kterého je možné upravovat vzhled elementů napsaných v jazyce HTML. Pomocí kaskádových stylů je možné upravovat rozložení stránky, barvy, fonty atp.

Kaskádové styly je možné zapisovat dvěma způsoby. Jedním ze způsobů je zápis přímo do HTML kódu a druhým je zápis do externího souboru s příponou `.css`, který je poté možné načíst v HTML dokumentu pomocí značky `<link>`. První způsob zápisu je vhodný pro malé projekty, kde bude využito pouze malé množství kaskádových stylů. Při větším počtu stylů je lepší zapisovat tyto styly do výše zmiňovaného `.css` souboru. [11]

Zápis kaskádových stylů se dělí na 2 části. První částí je selektor, který definuje HTML značku. Druhá část obsahuje deklaraci s příkazy. Výsledný zápis je poté v následujícím tvaru *selektor {vlastnost:hodnota;}*. V deklaraci pro jeden selektor může být i více vlastností, které se oddělují středníkem. Dále je také možné určit stejné vlastnosti více selektorům, kde každý selektor je oddělen čárkou. [11]

V případě, že se v HTML dokumentu nachází více značek jednoho druhu a je požadováno nastavit každé značce jinou vlastnost, je možné jednotlivým značkám přiřadit různé třídy (class) nebo identifikátory (id). Pomocí těchto tříd nebo identifikátorů je možné poté v CSS souboru upravit vlastnosti pouze pro zvolenou HTML značku. Hlavní rozdíl mezi třídou a identifikátorem je ten, že identifikátor lze použít v celém kódu pouze jednou, zatímco stejnou třídu je možné využít v několika HTML značkách. Při volání třídy v CSS dokumentu se místo selektoru píše tečka a název třídy; při volání identifikátoru se místo selektoru píše "#" a poté název identifikátoru. [11]

## 5.3 Javascript

JavaScript je dynamický skriptovací jazyk určený pro web. Jedná se o multiplatformní a objektově orientovaný jazyk, který běží na straně klienta. Je podporován téměř všemi internetovými prohlížeči jako Google Chrome, Mozilla Firefox, Safari, Internet Explorer, Opera, Edge atd. Dále je také podporován většinou mobilních prohlížečů pro chytré telefony. [12]

Pojem JavaScript může být zavádějící, jelikož sdílí část svého jména s jiným programovacím jazykem zvaným Java. Jedinou společnou věcí mezi nimi je podobná syntaxe, jinak se jedná o dva naprosto různé programovací jazyky. V případě JavaScriptu už se nejedná pouze o skriptovací jazyk, ale přerostl do plnohodnotného programovacího jazyka, který definuje nové funkce pro tvorbu obrovských projektů. [13]

Hlavním využitím JavaScriptu je rozšíření webových stránek tak, aby byly co nejvíce

uživatelsky přívětivé. Jedná se o dynamické aktualizování obsahu webových stránek, vylepšení uživatelského prostředí jako například animace, 2D a 3D grafika, interaktivní mapy, přehrávače videí atp. [12]

Stejně jako u kaskádových stylů je možné JavaScript využít dvěma způsoby. Prvním způsobem je vepsání přímo do HTML kódu a druhým je vytvoření samostatného souboru s příponou `.js`, který je poté vložen do HTML kódu pomocí značky `<script>`.

### 5.3.1 DOM

Document Object Model (DOM) je multi-platformní a jazykově nezávislé API, které zpracovává HTML a XML dokumenty jako logický strom, kde každý uzel tohoto stromu je objekt, který reprezentuje část daného dokumentu. DOM umožňuje programátorům přístup a modifikaci obsahu, struktury nebo stylu dokumentu a jeho částí. [14]

Každý webový prohlížeč měl své vlastní specifické rozhraní pro přístup a manipulaci s HTML elementy pomocí JavaScriptu. Tyto webové prohlížeče nebyly mezi sebou kompatibilní, a to vedlo k mnoha problémům při tvorbě webových stránek. Tato nekompatibilita přivedla členy skupiny W3C k myšlence vytvoření jednoho standartizovaného prostředí, které by bylo spustitelné ve všech webových prohlížečích a také by nebylo vázáno pouze na jeden programovací jazyk. [14]

### 5.3.2 JQuery

Jedná se o JavaScript knihovnu, která byla vytvořena za účelem zrychlení a usnadnění manipulace s Javascriptem. Jedná se o multiplatformní knihovnu, která je dostupná zdarma. Jedná se o jednu z nejpobulárnější a nejvíce využívaných JavaScriptových knihoven na trhu. [15]

Syntaxe JQuery byla navržena tak, aby ulehčila práci s HTML dokumenty, DOM objekty, tvorbou animací a vývojem Ajax aplikací. Primární zaměření bylo urychlení práce s DOM objekty, a proto byly implementovány funkce, které výrazně zrychlily hledání, výběr a manipulaci s DOM elementy. [15]

### 5.3.3 Node.js

Jedná se o open source framework, který umožňuje spuštění JavaScriptového kódu mimo webový prohlížeč. Hlavní využití nachází při tvorbě síťových programů, jako například webových serverů. Na rozdíl od jazyka PHP je Node.js asynchronní, tedy dokáže zpracovávat více funkcí naráz a nemusí čekat na dokončení předchozího příkazu, jako je tomu u PHP. [16]

Node.js je postaven na V8 JavaScript enginu, který je základem webového prohlížeče Google Chrome. Node.js využívá vysokého výkonu enginu V8, a proto je schopný

dosahovat vysokých rychlostí při zpracovávání JavaScriptového kódu. [17]

### 5.3.4 Electron

Electron je open source framework, který byl vyvinut společností GitHub. Umožňuje tvorbu multiplatformních aplikací s grafickým uživatelským rozhraním (GUI) pomocí komponentů, které byly původně vytvořeny pro tvorbu webových aplikací. Využívá prostředí Node.js pro backend a Chromium (open source webový prohlížeč vyvíjený společností Google) jako frontend. [18] [19]

Vývoj Elektronu započal v roce 2013 a v roce 2014 byl uvolněn jako open source. Jedná se o velmi populární framework, který je využíván pro tvorbu aplikací, jako například Discord, Skype, Slack a Twitch.tv. [18]

## 5.4 API

API je rozhraní pro programování aplikací. Jedná se o sbírku funkcí, tříd či protokolů některé knihovny nebo jiného programu, které může vývojář využít při tvorbě aplikace. API určuje, jak budou jednotlivé funkce knihovny volány. [20]

Web API běžící na straně serveru je programovací rozhraní, pomocí kterého je možná výměna dat s webovým serverem. Funguje na principu žádost-odpověď, kde pomocí různých metod žádá server o informace, a server poté odpoví odesláním informací ve formátu JSON nebo XML. Mezi jednotlivé metody žádosti patří [21]:

- GET - slouží pouze pro příjem informací a tedy nedochází k úpravám textu na serveru.
- POST - slouží k vytvoření nové kolekce mezi ostatními kolekcemi na serveru.
- PUT - slouží primárně k úpravě již existujících kolekcí na serveru.
- DELETE - slouží k odstranění existujícího zápisu.
- PATCH - slouží k částečné úpravě již existujícího zápisu.

Server při odesílání odpovědi odesílá také informaci o stavu. Při úspěšném provedení žádosti všech metod s výjimkou metody POST je odeslána stavová informace 200 (OK) a naopak, pokud nastala chyba, je odeslána informace o stavu 404 (Not Found). U metody POST je při úspěšném vytvoření odeslána stavová informace 201 (Created) a u metod PUT a PATCH může dojít k odeslání stavové informace 204 (No Content), pokud upravovaná kolekce nebyla nalezena na serveru. [21]

## 5.5 Chrome DevTools

Chrome DevTools je sada pro webové vývojáře, která je vestavěna do webového prohlížeče Google Chrome. DevTools umožňuje editovat webové stránky a rychle diagnostikovat případné problémy. Přístup k DevTools pro úpravu DOM nebo CSS je možný pomocí klávesových zkratk `Control+Shift+C` (Windows, Linux, Chrome OS) nebo `Command+Option+C` (Mac OS). [22]

Po otevření DevTools se zobrazí několik panelů, které nabízejí různé funkce pro práci [22]:

- Elements - slouží pro provádění změn v DOM a CSS.
- Console - umožňuje spuštění JavaScript přímo z konzole a výpis zpráv do konzole.
- Sources - obsahuje seznam všech souborů, které byly využity pro spuštění webové stránky nebo aplikace a umožňuje jejich editaci.
- Network - zobrazuje aktivitu na síti.
- Performance - umožňuje zjištění výkonu webové stránky nebo aplikace, za účelem optimalizace rychlosti.
- Memory - zobrazuje aktuální stav využití paměti a dokáže zjistit úniky paměti.
- Application - slouží k zobrazení všech zdrojů, které jsou načítány, jako například Web SQL databáze, cookies, obrázky, fonty a CSS.
- Security - slouží k zabezpečení webové stránky nebo aplikace, jako například zjištění problémů s certifikáty atp.

## 6 Platforma MagicMirror2

MagicMirror2 (Obr 6.1) je open source modulární platforma inteligentního zrcadla. S rostoucím počtem instalovatelných modulů umožňuje platforma MagicMirror2 přeměnit zrcadlo na chodbě nebo v koupelně na personálního asistenta. Platforma byla vytvořena Michaelem Teeuwem, který je také tvůrcem originální platformy MagicMirror. [23]



Obr. 6.1 MagicMirror2 [24]

### 6.1 Vznik platformy

Myšlenka vytvoření platformy MagicMirror vznikla v roce 2014, když se Michael Teeuwem procházel se svojí přítelkyní nákupním centrem, kde si všiml zrcadla, na kterém byl pověšen svítící nápis a rozhodl se, že vytvoří něco na podobném principu, ale lepšího. K výrobě prvního inteligentního zrcadla (Obr 6.2) bylo zapotřebí zajistit zrcadlo, monitor, Raspberry Pi a spoustu volného času. Bylo zapotřebí sehnat zrcadlo, které by se chovalo jako normální zrcadlo, pokud je pozadí za ním černé, a zároveň by se chovalo, jako normální průhledné sklo, když se za ním objeví informace na monitoru. Volba padla na polopropustné zrcadlo, které funguje na stejném principu jako zrcadla

na policejních stanicích. Jedná se o čiré sklo, které je potaženo velmi tenkou a skoro průhlednou vrstvou hliníku. Výsledkem je povrch, který odráží část dopadajícího světla a zbytek propouští. Pokud na jednu stranu zrcadla dopadají světelné paprsky a druhá strana je zatemněná, chová se jako zrcadlo, ale pokud jsou obě strany zrcadla osvětleny, chová se jako normální průhledné sklo. [24]



Obr. 6.2 MagicMirror2 zrcadlo [25]

Dalším krokem bylo zajistit monitor, který bude mít výstupy a vstupy orientovány směrem dolů a zároveň bude dostatečně velký. Po odstranění plastového krytu monitoru bylo možné jej vložit do podomácky vyrobeného rámu, který zároveň obsahoval polo-propustné zrcadlo. V předposledním kroku bylo zapotřebí zakomponovat do zrcadla počítač Raspberry Pi a napájecí kabely pro monitor a Raspberry tak, aby z výsledného produktu vedl pouze jeden napájecí kabel. V posledním kroku bylo zapotřebí nainstalovat OS Raspbian na Raspberry Pi a vytvořit vhodné prostředí. [24]

První verze platformy se stala mezi uživateli velice populární a mnoho uživatelů se snažilo tuto aplikaci vylepšit. Autor původního projektu ale brzy zjistil, že tato verze není důmyslně propracovaná a je nutné ji úplně přepracovat. V závislosti na tomto rozhodnutí vznikla nová verze. Druhá verze platformy nazvaná MagicMirror2 vznikla



v roce 2016 spoluprací autora původní verze zrcadla a stovek dalších nadšenců, kterým se tento nápad zalíbil. [24]

V říjnu roku 2016 se platforma MagicMirror2 umístila na první pozici v seznamu 50-ti nejlepších projektů vytvořených s využitím Raspberry Pi v magazínu The MagPi, který je oficiálním magazínem nadace Raspberry Pi. [24]

### 6.1.1 Využití technologie

Pro vývoj první verze MagicMirror bylo využito webserveru Apache a webového prohlížeče Chromium, který byl spuštěn v Kiosk módu, který zajišťoval přechod do zobrazení na celou obrazovku při každém restartu Raspberry Pi. Hlavními požadavky autora bylo, aby zrcadlo umělo zobrazovat aktuální počasí, aktuální zprávy, komplimenty a aktuální čas. Při vývoji byly využity JavaScriptové knihovny JQuery, Moment.js, FeedToJson a iCal Parser. Design vzhledu celého zrcadla se odvíjel od designu společnosti Apple. [24]

Verze MagicMirror2 je postavena na open source frameworku Electron, který umožňuje uživatelům udržovat platformu na nejnovější verzi bez ztráty konfigurace. Aplikace obsahuje vestavěný web server a také velké množství dokumentace, která pomáhá vývojářům s tvorbou nových modulů. [24]

## 6.2 Github MM2

Všechny potřebné soubory pro využívání MagicMirror2 se nacházejí na GitHub stránce projektu, společně s rozšiřujícími moduly a dokumentací pro využívání platformy.

### 6.2.1 Moduly

Ve verzi MM2, která se nachází na GitHub stránce projektu jsou předinstalovány výchozí moduly (Obr 6.1), jako například modul pro zobrazení aktuálního počasí v nastavené lokalitě, modul pro zobrazení nejnovější zpráv z internetu pomocí RSS kanálu, modul pro zobrazení aktuálního data a času a modul pro zobrazení kalendáře. [26]

Díky popularitě této platformy je dostupný nespočet modulů, které byly tvořeny členy MM2 komunity a jejich seznam je dostupný na GitHub stránce projektu, kde je rozřazen do následujících kategorií [26]:

- Vývoj / jádro MagicMirror - Obsahuje moduly jako např. MMM-connection-status, který zobrazuje stav připojení k internetu nebo MMM-ModuleScheduler, který umožňuje skrýt a zobrazit moduly v závislosti na vytvořeném časovém plánu.

- Finance - Moduly jako např. MMM-Bitcoin zobrazující aktuální cenu virtuální měny Bitcoin.
- Zprávy / Náboženství / Informace - Obsahuje např. MMM-Wiki, který zobrazuje náhodné úryvky z Wikipedie.
- Přeprava / Cestování - Obsahuje moduly pro motoristy jako např. modul MMM-GoogleMapsTraffic, který zobrazuje aktuální stav provozu na silnicích.
- Hlasové ovládání - Moduly umožňující ovládání zrcadla pomocí hlasových příkazů např. MMM-GoogleAssistant.
- Počasí - Obsahuje moduly pro předpověď počasí, např. MMM-DarkSkyForecast.
- Sport - Moduly zobrazující aktuální informace o dění ve sportu, např. MMM-NHL.
- Nástroje / IOT / Integrace - Moduly jako například MMM-EmbedYoutube, pro zobrazování youtube videí.
- Zábava / Různé - Například MMM-Instagram, který slouží pro zobrazování aktuálních fotek ze sociální sítě Instagram.
- Zdraví - Momentálně obsahuje pouze jeden modul MMM-Nightscout, který slouží diabetikům ke sledování stavu cukru v krvi.

### 6.2.2 Dokumentace

K platformě MagicMirror2 je dostupná přehledná a srozumitelná dokumentace, která obsahuje základní informace o platformě, postup instalace platformy na Raspberry Pi a mnoho dalších informací spojených s provozem platformy. [23]

Pro případné vývojáře se zde také nachází dokumentace pro vývoj modulů. V dokumentaci je popsán základní postup tvorby, jako například umístění souborů, popis tvorby hlavních spouštěcích souborů a také popis jednotlivých funkcí, které je možné ke tvorbě modulů využít. [27]

### 6.3 Fórum

Na fóru MagicMirror2, které je z větší části řízeno nadšenci této platformy, je možné najít řešení různých problémů, které mohly nastat při využívání platformy, ale také spoustu dalších věcí, které mají co do činění s touto platformou. Nachází se zde například vlákno, kde uživatelé mohou dávat inspiraci vývojářům pro vytvoření nových modulů. [28]

## II. PRAKTICKÁ ČÁST

## 7 Instalace a konfigurace platformy MagicMirror2

### 7.1 Instalace

#### 7.1.1 Automatická instalace

Automatickou instalaci platformy MagicMirror2 je možné provést pouze na modelech Raspberry Pi 2 a Raspberry Pi 3. Ostatní modely nejsou v tento moment dostupné, jelikož Electron, který je využit pro platformu MagicMirror2 tyto modely nepodporuje. Tímto způsobem je možné také nainstalovat platformu i na model Raspberry Pi 1, ale je nutné platformu spustit v režimu pro server (viz kapitola 7.2). K provedení instalace je také nutné využít nejnovější plné verze Raspbianu a je doporučeno nevyužívat Lite verzi. [23]

Spuštěním následujícího kódu v terminálu Raspbianu dojde k instalaci platformy [23]:

```
bash -c "$(curl -sL https://raw.githubusercontent.com/MichMich  
↪ /MagicMirror/master/installers/raspberry.sh)"
```

Při instalaci se může objevit hláška "Unable to install dependencies!", která uživatele informuje o tom, že instalátor není schopen nainstalovat balíčky, které jsou nutné pro správné fungování platformy. Tento problém je možné vyřešit instalací starší verze Electronu pomocí příkazu [29]:

```
sudo npm install -g electron@3.0.13
```

Instalátor se v průběhu instalace zeptá, zdali je vyžadováno využití manažera procesů PM2, který umožňuje platformě okamžité zapnutí v případě restartu Raspberry Pi. Po stisku kláves Y (pro potvrzení), nebo N (pro zamítnutí) bude instalace dále pokračovat. V případě zamítnutí je možné manažera PM2 doinstalovat později (viz kapitola 7.3.1).

V poslední fázi instalace je možné zvolit, zdali bude zobrazován spořič obrazovky. Spořič obrazovky je možné vypnout stisknutím klávesy Y (pro potvrzení), nebo ponechat zapnutý pomocí klávesy N (pro zamítnutí). Toto nastavení je opět možné později změnit (viz kapitola 7.3.2).

#### 7.1.2 Manuální instalace

Manuální instalace platformy je vhodná například v případě, že automatická instalace selhala, nebo pokud při aktualizaci došlo k chybě a MagicMirror2 přestal fungovat. Manuální instalace je ve většině případů rychlejší než snaha o vyřešení problémů s již existující instalací. K úspěšnému provedení instalace je nutné dodržovat následující kroky. V prvním kroku je nutné stáhnout a nainstalovat nejnovější verzi frameworku Node.js pomocí následujících příkazů [23]:

```
curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -  
sudo apt install -y nodejs
```

V dalším kroku je nutné naklonovat repozitář MagicMirror pomocí příkazu:

```
git clone https://github.com/MichMich/MagicMirror
```

Po úspěšném naklonování repozitáře je nutné vstoupit do repozitáře MagicMirror pomocí příkazu:

```
cd MagicMirror/
```

a poté spustit instalaci aplikace pomocí příkazu:

```
npm install
```

Instalace `npm install` zabere hodně času, většinou bez jakékoliv odezvy na konzoli. V případě modelu Raspberry Pi 3 instalace trvá v průměru 10 minut, u modelu Raspberry Pi 2 může instalace trvat až 25 minut. Je nutné nechat instalaci dokončit bez přerušení, jinak může dojít k zamrznutí celého systému a bude nutné znovu nainstalovat celý OS a opakovat instalaci platformy MagicMirror2. [23]

Po úspěšném dokončení instalace je možné aplikaci spustit, ale s největší pravděpodobností se po spuštění objeví hláška "Please create a config file", která uživatele informuje o chybějícím konfiguračním souboru. K vytvoření tohoto souboru je zapotřebí otevřít správce souborů, který je nainstalován v OS a přechodem do složky MagicMirror. V této složce se nachází složka config, která obsahuje soubor `config.js.sample`. Je zapotřebí vytvořit kopii tohoto souboru a přejmenovat ji na `config.js`. Po vytvoření a přejmenování této kopie je možné spustit aplikaci pomocí příkazu [23]:

```
npm start
```

Případně je možné aplikaci spustit se zapnutým DevTools pomocí příkazu:

```
npm start dev
```

## 7.2 MagicMirror2 jako server

Aplikace spuštěná jako server umožňuje využití jiného webové prohlížeče pro zobrazení dat, a není tedy nutné využívat výchozí prohlížeč. Před spuštěním je nutné změnit síťové nastavení aplikace v konfiguračním souboru. V možnosti `address` je nutné nastavit hodnotu "0.0.0.0", aby mohl server naslouchat na všech rozhraních a ne jenom v režimu localhost. Do pole u položky `ipWhitelist` je nutné přidat IP adresu klienta nebo

klientů, kteří budou tento server využívat. V této položce je možné nastavit prázdné pole, což umožní připojení všem klientům v dané síti. [23]

Spuštění MagicMirror2 jako serveru je možné provést pomocí následujícího příkazu [23]:

```
node serveronly
```

Po úspěšném spuštění serveru se mohou klienti, kteří mají nainstalovanou aplikaci MagicMirror2 připojit k tomuto serveru pomocí následujícího příkazu, kde je nutné dosadit IP adresu a port serveru, ke kterému se klient připojuje [23]:

```
node clientonly --address --port
```

## 7.3 Konfigurace

### 7.3.1 Automatické spuštění pomocí PM2

Manažer procesů PM2 umožňuje udržovat aplikace neustále spuštěné. V případě platformy MagicMirror2 je možné tohoto manažera využít pro automatické spuštění platformy při restartování OS. Pokud při automatické instalaci byla zamítnuta instalace manažera procesů PM2, nebo bylo využito manuální instalace, je možné tohoto manažera doinstalovat a donastavit. V případě automatické instalace platformy a potvrzení instalace PM2 je již manažer nainstalován a nastaven, a není tedy nutné dále postupovat v tomto návodu. [30]

V prvním kroku je nutné nainstalovat manažera na Raspberry Pi pomocí příkazu [30]:

```
sudo npm install -g pm2
```

Dále je nutné umožnit spuštění manažera při každém načítání OS. Manažer obsahuje pomocníka, kterého je možné otevřít pomocí příkazu [30]:

```
pm2 startup
```

V konzoli se nyní objeví příkaz, který je nutné spustit. Dále je zapotřebí vytvořit skript, který umožní propojit platformu s tímto manažerem. Tento skript je vhodné umístit mimo složku MagicMirror, aby při aktualizacích nedocházelo k problémům. Následující příkazy zajistí přechod do kořenového adresáře OS a následně vytvoří a otevrou nový textový soubor `mm.sh`. [30]

```
cd ~  
nano mm.sh
```

Do souboru `mm.sh` je nutné vložit následující řádky a poté uložit změny pomocí klávesových zkratk `CTRL-O` a `CTRL-X`. [30]

```
cd ~/MagicMirror
DISPLAY=:0 npm start
```

Tento nově vytvořený a upravený soubor je poté nutné převést na spustitelný skript pomocí příkazu [30]:

```
chmod +x mm.sh
```

Nyní je vše připraveno pro spuštění platformy pomocí příkazu [30]:

```
pm2 start mm.sh
```

Posledním krokem je uložení aktuálně běžících stavů PM2 [30]:

```
pm2 save
```

Nyní je již vše nastavené a při restartu počítače by měla platforma automaticky naběhnout.

Následující příkazy je možné využít pro správu platformy [30]:

```
pm2 stop mm      # pro ukončení
pm2 restart mm   # pro restartování
pm2 logs mm      # pro zobrazení záznamů výkonů
pm2 show mm      # pro zobrazení informací o procesu MagicMirror
```

### 7.3.2 Spořič obrazovky

Pro vypnutí spořiče obrazovky je nutné otevřít soubor autostart [31]:

```
sudo nano /etc/xdg/lxsession/LXDE-pi/autostart
```

Do tohoto souboru je zapotřebí přidat následující řádky, a poté pomocí klávesových zkratk CTRL-O a CTRL-X, tento soubor uložit a uzavřít. [31]

```
@xset s noblank
@xset s off
@xset -dpms
```

Dále je nutné otevřít soubor `lightdm.conf` [31]:

```
sudo nano /etc/lightdm/lightdm.conf
```

A pod položku `[SeatDefault]` přidat následující řádek.

```
xserver-command=X -s 0 -dpms
```

V případě, že se v souboru tato položka nenachází, je nutné ji vytvořit. [31]

### 7.3.3 Natočení obrazovky

Nastavení natočení obrazovky je možné provést otevřením souboru `config.txt` [31]:

```
sudo nano /boot/config.txt
```

A poté vložení následujících řádků [31]:

```
display_rotate=1  
avoid_warnings=1
```

Změnu úhlu natočení je možné provést změnou čísla u možnosti `display_rotate`. 0 pro normální zobrazení, 1 pro natočení o 90 stupňů, 2 pro natočení o 180 stupňů a 3 pro natočení o 270 stupňů. Pomocí kláves `CTRL-O` a `CTRL-X` je nutné soubor uložit, a poté pomocí následujícího příkazu restartovat počítač: [31]

```
sudo reboot
```

### 7.3.4 Konfigurační soubor

Při instalaci je vytvořen konfigurační soubor, který obsahuje výchozí nastavení platformy. Tento soubor se nachází ve složce `MagicMirror/config` pod názvem `config.js`. Po otevření tohoto souboru pomocí programu na úpravu kódu, jako například Geany, je možné upravovat následující možnosti [23]:

- `port` - Obsahuje informace o portu, na kterém bude server `MagicMirror2` pracovat. Výchozí hodnota je 8080.
- `address` - Jedná se o IP adresy síťových rozhraní, na kterých budou přijímány připojení. Výchozí hodnota je `localhost`, která zabraňuje zobrazení vestavěného web-serveru ostatním zařízeními v síti. Pro umožnění připojení ostatních zařízení v síti k tomuto web-serveru je možné změnit hodnotu na `0.0.0.0`.
- `ipWhitelist` - Obsahuje seznam všech IP adres, kterým je umožněno připojení k tomuto web-serveru. Výchozí hodnota je `["127.0.0.1", "::ffff:127.0.0.1", "::1"]` a umožňuje připojení pouze z `localhostu`. Zde je možné přidat další adresy, kterým bude umožněn přístup. Rozsah IP adres je možné zadávat pomocí masky podsítě jako například `["127.0.0.1", "127.0.0.1/24"]`, nebo přímo `["127.0.0.1", ["192.168.0.1", "192.168.0.100"]]`. Pro povolení přístupu všem IP adresám je možné nastavit hodnotu prázdného pole `[]`.
- `zoom` - Umožňuje změnu velikosti textu, který je zobrazován. Výchozí hodnota je 1.0



- `language` - Nastavení jazyku platformy. Možnosti jsou například `en` (angličtina), `fr` (francouzština), `cs` (čeština) atd. Výchozí hodnotou je `en`. Některé elementy platformy nemusí obsahovat lokalizaci pro daný jazyk.
- `timeFormat` - Formát času. Možné hodnoty jsou `12` a `24`. Výchozí hodnota je `24`.
- `units` - Jednotky, které jsou využívány ve výchozích modulech počasí, ale také v některých uživatelských modulech. Možnosti jsou `metric` (metrické) a `imperial` (imperiální). Výchozí hodnota je `metric`.
- `modules` - Pole obsahující seznam aktivních modulů (viz kapitola 7.4).
- `electronOptions` - Volitelné pole obsahující nastavení frameworku Electron.
- `customCss` - Cesta k `custom.css` souboru. Výchozí hodnota je `css/custom.css`

#### 7.4 Instalace a konfigurace modulů

Popis instalace každého modulu je obsažen v dokumentaci pro daný modul. Většinu modulů je možné nainstalovat pomocí příkazu `git pull` a odkazu na danou GitHub stránku, případně doinstalováním závislých funkcí pomocí `npm install`. Po stažení souborů modulu je nutné modul aktivovat v konfiguračním souboru. Každý modul má opět své vlastní možnosti a dokumentace k tomuto modulu by měla obsahovat ukázkou výchozího nastavení konfiguračního souboru. [23]

Následující možnosti je možné přidat nebo upravovat v hlavním konfiguračním souboru `config.js` [23]:

- `module` - Zde je nutné vložit název modulu, který bude zobrazován.
- `position` - Určuje pozici modulu na obrazovce. Mezi možné hodnoty patří: `top_bar`, `top_left`, `top_center`, `top_right`, `upper_third`, `middle_center`, `lower_third`, `bottom_left`, `bottom_center`, `bottom_right`, `bottom_bar`, `fullscreen_above` a `fullscreen_bellow`. Toto pole není povinné, ale některé moduly jej mohou vyžadovat.
- `classes` - Další třídy, které jsou posílány do modulu. Toto pole je nepovinné.
- `header` - Slouží pro zobrazení hlavičky nad modulem. Toto pole je nepovinné.
- `disabled` - Pomocí parametru `true` je možné zakázat zobrazení modulu. Toto pole je nepovinné.
- `config` - Objekt, který obsahuje konfigurační vlastnosti modulu. Více informací je možné zjistit v dokumentaci k žádanému modulu.

## 7.5 Aktualizace

V případě uvolnění nové aktualizace platformy se v okně aplikace objeví informační tabulka, která hlásí nově dostupnou verzi. Pro provedení aktualizace je nutné v terminálu spustit následující příkaz [23]:

```
git pull && npm install
```

Tento příkaz provede stažení nejnovější verze z GitHubu a tuto verzi poté nainstaluje.

## 8 Analýza a návrh modulů

Seznam všech modulů je dostupný na webové stránce: <https://github.com/MichMich/MagicMirror/wiki/3rd-party-modules>. Seznam je rozdělený do kategorií, které reprezentují různé způsoby využití (viz kapitola 6.2.1).

V každé kategorii se nacházejí moduly, které nějakým způsobem usnadňují život a šetří čas. Platforma MagicMirror2 byla vytvořena tak, aby fungovala jako chytré zrcadlo, které bude viset na stěně. Je tedy nutné při vývoji modulů myslet na fakt, že většinu času nebude k platformě připojena myš ani klávesnice. Většina modulů z výše uvedených kategorií pracuje pouze s informacemi uvedenými v konfiguračním souboru a tyto informace není možné měnit bez využití myši a klávesnice. Jedním z kritérií při návrhu je tedy vytvořit modul, který by pro případné vkládání informací nevyžadoval připojení klávesnice a myši k platformě.

Další kritérium při vývoji je schopnost modulu zobrazovat informace v českém jazyce, jelikož mezi dostupnými moduly se momentálně nenachází jediný modul, který by disponoval českým překladem. Je tedy nutné využít API, které je schopné vracet informace i v českém jazyce.

Využitelnost modulu je dalším klíčovým kritériem při vývoji. Modul by tedy měl být vytvořen tak, aby jej uživatel mohl využívat nejlépe každý den a aby se nejednalo pouze o modul na jedno použití.

Posledním kritériem je dostupnost API, které musí být veřejně dostupné a jeho využívání musí být bezplatné. Volba tedy padla na API od České pošty, které je dostupné na webové stránce <https://b2c.cpost.cz> a na API pro zobrazování jmenin, dostupné z <https://api.abalin.net>.

Na základě výše zmíněných API tedy vzniknou dva moduly:

1. Modul, který bude využívat informace o zásilkách, které jsou přepravovány Českou poštou. Číslo balíku, u kterých bude požadováno zobrazení na platformě, budou modulu předávána pomocí dokumentu vytvořeného na platformě Google Docs. V případě, že nebudou dostupné žádné zásilky k zobrazení, bude na obrazovce zobrazena informace o otevírací době pobočky pošty. Tato pobočka musí být specifikována v konfiguračním souboru modulu.
2. Modul pro zobrazování jmenin, který bude v závislosti na nastavení v konfiguračním souboru platformy zobrazovat jmeniny v daném státě.

Oba moduly budou zobrazovat informace v českém jazyce, případně v dalších jazycích.

## 9 Implementace

Pro vytvoření obou modulů bylo využito šablony, která je dostupná na GitHub stránce projektu. Tato šablona byla poté upravena tak, aby odpovídala jednotlivým požadavkům. Vývoj následujících modulů probíhal přímo na Raspberry Pi s OS Raspbian a v programu pro editaci kódu Geany.

### 9.1 Modul MMM-CzechPost

Na začátku celého kódu se nachází pole, které obsahuje definici výchozích proměnných.

```
defaults: {
  updateInterval: 5*60*1000, //every 5 minutes
  initialLoadDelay: 3000,
  retryDelay: 3000,
  lang: config.language,
  packagesUrl: "",
  postCode: "76001",
  showPostInfo: "all", //today, all, none
  showPackageInfo: "all", //latest, all
  tableClass: "small",
  maxNumOfShownPackages: 1,
}
```

Proměnná `updateInterval` je určena pro stanovení časového intervalu po jehož uplynutí dojde k aktualizaci dat v modulu. `initialLoadDelay` a `retryDelay` - slouží ke specifikování doby čekání při načítání a při opětovném pokusu o stažení dat z API. Pomocí proměnných `packagesUrl`, `postCode`, `showPostInfo`, `showPackageInfo`, `tableClass` a `maxNumOfShownPackages` mohou uživatelé ovládat chování tohoto modulu.

```
getScripts: function(){
  return ['https://ajax.googleapis.com/ajax/libs
  ↪ /jquery/1.9.1/jquery.min.js']
},
```

Funkce `getScripts` je vestavěna v platformě MagicMirror a jejím využitím je možné specifikovat další skripty, které jsou potřebné pro vývoj modulu, ale nenacházejí se na platformě. V tomto modulu je nutné využít knihovnu `JQuery`, která nám později poslouží pro zasílání čísel balíků do modulu.

```
start: function() {
  Log.info("Starting module: " + this.name);
  this.packages = [];
  this.packagesData = [];
  this.table;
  this.loaded = false;
  this.scheduleUpdate(this.config.initialLoadDelay);
  this.updateTimer = null;
},
```

Ve funkci `start`, která je základním prvkem všech modulů, dochází k definování globálních proměnných, které bude později nutné využívat v různých funkcích. Po první definici je volána funkce `scheduleUpdate` s hodnotou proměnné `initialLoadDelay`.

```

scheduleUpdate: function(delay) {
    var nextLoad = this.config.updateInterval;
    if (typeof delay !== "undefined" && delay >= 0) {
        nextLoad = delay;
    }
    var self = this;
    clearTimeout(this.updateTimer);
    this.updateTimer = setTimeout(function() {
        if(self.config.packagesUrl !== ""){
            self.getPackages();
        }else if(self.config.postCode !== ""){
            self.loaded = true;
            self.updateDom();
        }else{Log.error("Post code not set!");
            return;}
    }, nextLoad);
},

```

Pomocí funkce `scheduleUpdate` je možné nastavit interval aktualizace informací, které jsou zobrazovány na obrazovce. Nachází se zde podmínka, ve které dochází k volání dalších funkcí, v závislosti na informacích nastavených ve výchozích proměnných popř. v konfiguračním souboru. V případě, že není zadána adresa, která obsahuje čísla zásilek pro zobrazení, kontroluje program zdali je zadáno poštovní směrovací číslo pošty, která by měla být zobrazena. Pokud není zadána ani jedna informace, dojde k vypsání chybové hlášky na konzoli a k ukončení modulu. Je-li zadána adresa obsahující čísla zásilek, dojde k volání funkce `getPackages`. V případě, že není zadána webová adresa, ale je zadáno PSČ, je volána funkce `getDom`.

```

getPackages: function(){
    var self = this;
    var retry = true;
    self.packages = [];
    var dataRequest = new XMLHttpRequest();
    var url = this.config.packagesUrl;
    $.get(url,function(response) {
        try{
            parser = new DOMParser();
            xmlDoc= parser.parseFromString(response,"text/xml");
        }
        catch(e){
            Log.log(e.message);
            return;
        }
        var metas = xmlDoc.getElementsByTagName("meta");
        for(var i=0; i<metas.length;i++){
            if(metas[i].getAttribute("property") ==
                ↪ "og:description"){
                var res =
                    ↪ metas[i].getAttribute("content");
                self.packages =
                    ↪ res.replace(/\s+/g,",").split(",");
            }
        }
        self.getPackagesData();
    });
},

```

Aby bylo možné zadávat čísla zásilek do zrcadla bez nutnosti využití klávesnice a myši u

platformy MagicMirror2, byla vytvořena následující funkce nazvaná `getPackages`. Tato funkce využívá veřejně sdíleného dokumentu vytvořeného na platformě Google Docs. Tento dokument je poté převeden do XML formátu. Google Docs - ukládá veškerý obsah sdíleného dokumentu v meta značce, a proto je poměrně lehké tyto informace získat. Tyto informace jsou poté uloženy do pole a je zavolána funkce `getPackagesData`.

```
getPackagesData: function(){
if(this.packages.length > 0){
    this.packagesData = [];
    for(var x=0; x<this.packages.length;x++){
        var self = this;
        var retry = true;
        var dataRequest = new XMLHttpRequest();
        if(this.config.lang === "cs"){
            var url =
                "https://thingproxy.freeboard.io/fetch/" +
                "https://b2c.cpost.cz/services
                /ParcelHistory/getDataAsJson?idParcel=" +
                this.packages[x];
        } else{var url =
            "https://thingproxy.freeboard.io/fetch/" +
            "https://b2c.cpost.cz/services
            /ParcelHistory/getDataAsJson?idParcel=" +
            this.packages[x] + "&language=en";}
        dataRequest.open("GET",url,false);
        dataRequest.onreadystatechange = function() {
            if(this.readyState === 4){
                if(this.status >= 200 && this.status <
                    ↪ 400){
                    self.packagesData.push(JSON.parse
                    ↪ (this.response));
                }else {Log.error(self.name + ": Could
                    ↪ not load packages.");}
                if (retry) {
                    self.scheduleUpdate((self.loaded) ? -1
                    ↪ : self.config.retryDelay);
                }
            }
        };
        dataRequest.send();
    }
    this.loaded = true;
    this.updateDom();
}else if(this.config.postCode === ""){
    Log.error("No parcel numbers entered!");
    Log.error("No post code entered!");
    return;
}else{this.loaded = true;
    this.updateDom();
    Log.error("No parcel numbers entered! Showing
    ↪ only post info.");}
},
```

Pro získávání dat z API České pošty slouží funkce `getPackagesData`. Je zde využíváno funkce `XMLHttpRequest`, která umí komunikovat s různými API a získat od nich data ve formátu JSON nebo XML. V první řadě je zapotřebí vytvořit proměnnou, která bude ve formátu `XMLHttpRequest`. Poté pomocí metody `.open` dojde k nastavení parametrů, které jsou nezbytně nutné pro otevření spojení mezi klientem a serverem. Mezi tyto

parametry patří způsob získávání dat ze serveru, webová adresa API a možnost `true`, nebo `false`, která určuje, zdali API žádost bude provedena synchronně (`false`), nebo asynchronně (`true`). K připojení k tomuto API je využit proxy server, jelikož se zde vyskytuje problém s CORS, který neumožňuje využívat toto API v aplikacích, ale pouze ve webových prohlížečích. Požadavek na server je odeslán pomocí metody `.send` a výsledná odpověď je poté analyzována metodou `.onreadystatechange`. Server oznámí status 200 pokud došlo k úspěšnému připojení a odeslání dat, v případě neúspěchu je do konzole vypsána chybová hláška a modul provede znovu žádost na server. Po úspěšném získání dat jsou tato data přeparsována do globální proměnné `packagesData` a poté je zavolána funkce `getDom`.

```
getDom: function(){
    var wrapper = document.createElement("div");
    wrapper.className = "czechPostWrapper";
    if (!this.loaded) {
        wrapper.innerHTML = this.translate("LOADING");
        wrapper.className = "dimmed light small";
        return wrapper;
    }
    if(this.config.packagesUrl != "" && this.packages.length > 0){
        this.table = document.createElement("table");
        this.table.className = this.config.tableClass;
        var row = document.createElement("tr");
        this.table.appendChild(row);
        var infoCell = document.createElement("th");
        infoCell.className = "infoTest";
        infoCell.setAttribute("colspan", "2");
        infoCell.innerHTML = this.translate("INFO");
        row.appendChild(infoCell);
        var row = document.createElement("tr");
        row.className = "emptyRow";
        this.table.appendChild(row);
    }
}
```

Funkce `getDom` slouží k vytvoření HTML elementů. V první podmínce modul kontroluje pomocí proměnné `loaded`, zda byly provedeny všechny potřebné úkony. V následující podmínce je kontrolováno, zdali je zadána webová adresa sdíleného Google Docs dokumentu a zdali tento dokument obsahuje nějaký text. V případě, že jsou obě podmínky splněny, dochází k vytvoření hlavní tabulky, která bude obsahovat všechny informace, a také dochází k vytvoření hlavního nadpisu tabulky, který uživatele informuje, o jaký modul se jedná.

```
for(var y=0; y < this.packages.length;y++){
    var row = document.createElement("tr");
    this.table.appendChild(row);
    var idCell = document.createElement("th");
    idCell.className = "packageId";
    idCell.setAttribute("colspan", "2");
    idCell.innerHTML = this.translate("PACKAGE_INFO") +
        ↪ this.packagesData[y][0].id;
    row.appendChild(idCell);
    var row = document.createElement("tr");
    this.table.appendChild(row);
    var priceCell = document.createElement("td");
```

```

priceCell.className = "price";
priceCell.setAttribute("colspan", "2");
if(this.packagesData[y][0].attributes.dobirka != 0){
    priceCell.innerHTML = this.translate("PRICE") +
        this.packagesData[y][0].attributes.dobirka + "
        ⇔ CZK";}
row.appendChild(priceCell);
var row = document.createElement("tr");
this.table.appendChild(row);
if(this.packagesData[y][0].attributes.stored != ""){
    var storedCell = document.createElement("td");
    storedCell.className = "stored";
    storedCell.setAttribute("colspan", "2");
    storedCell.innerHTML = this.translate("STORED") +
        ⇔ this.packagesData[y][0].attributes.ulozeniDo;
    row.appendChild(storedCell);
}
}

```

Po vytvoření hlavní tabulky a nadpisu dojde ke spuštění cyklu, u kterého je počet opakování stanoven počtem získaných odpovědí u vložených zásilek. Při každém otočení cyklu je do hlavní tabulky zapsáno číslo balíku, a pokud se jedná o dobírku, je zde zobrazena i cena dobírky. Pokud byla zásilka uložena na poště, v tabulce se zobrazí i datum, do kterého je zásilka uložena.

```

var postCode = null;
var postName = null;
if(this.config.showPackageInfo === "all" && this.packages.length <=
    ⇔ this.config.maxNumOfShownPackages ){
    for(var x = 0; x < this.packagesData[y][0].states.state.length;
        ⇔ x++){
        var row = document.createElement("tr");
        this.table.appendChild(row);
        var dateCell = document.createElement("td");
        dateCell.className = "packageDate";
        dateCell.innerHTML =
            ⇔ this.packagesData[y][0].states.state[x].date;
        row.appendChild(dateCell);
        var textCell = document.createElement("td");
        textCell.className = "packageText";
        Log.log(this.packagesData);
        if(this.packagesData[y][0].states.state[x].id === "81" ||
            ⇔ this.packagesData[y][0].states.state[x].id === "91"){
            postCode =
                ⇔ this.packagesData[y][0].states.state[x].postcode;
            postName =
                ⇔ this.packagesData[y][0].states.state[x].postoffice;
            textCell.innerHTML =
                this.packagesData[y][0].states.state[x].text +
                ⇔ "___" +
                ⇔ this.packagesData[y][0].states.state[x].postoffice;
        }else {textCell.innerHTML =
            ⇔ this.packagesData[y][0].states.state[x].text;}
        row.appendChild(textCell);
    }
    if(postCode != ""){
        var row = document.createElement("tr");
        row.className = "emptyRow";
        this.table.appendChild(row);
        this.showPostOfficeInfo(postCode, postName);
    }
}

```



Při každém průchodu výše uvedeného cyklu dochází ke kontrole, jakým způsobem mají být zobrazovány informace o zásilkách. Pomocí proměnné `showPackagesInfo` je možné nastavit, zdali mají být zobrazovány všechny informace o zásilkách, nebo bude zobrazena pouze nejaktuálnější informace. Dále je možné pomocí proměnné `maxNumOfShownPackages` limitovat maximální počet zásilek, které budou v jeden moment zobrazeny na obrazovce, pokud bude vybrána možnost zobrazení všech informací.

V případě, že je vybrána možnost zobrazení všech informací a počet zásilek je menší než hodnota uvedená u proměnné `maxNumOfShownPackages`, začne se provádět cyklus, který do tabulky vypíše všechny dostupné informace o stavu zásilky. V průběhu tohoto cyklu dochází ke kontrole id. čísla stavu zásilek a pokud se hodnota tohoto id rovná číslu 81 (zásilka uložena na poště) nebo 91 (zásilka doručena), jsou do proměnných `postCode` a `postName` zapsány údaje o názvu pošty a o PSČ, které jsou poté využity pro zobrazení otevírací doby pošty, na které se zásilka nachází.

```
}else { var row = document.createElement("tr");
        this.table.appendChild(row);
        var single = this.packagesData[y][0].states.state.length - 1;
        var dateCell = document.createElement("td");
        dateCell.className = "packageDate";
        dateCell.innerHTML =
            ↪ this.packagesData[y][0].states.state[single].date;
        row.appendChild(dateCell);
        var textCell = document.createElement("td");
        textCell.className = "packageText";
        if(this.packagesData[y][0].states.state[single].id === "81" ||
            ↪ this.packagesData[y][0].states.state[single].id === "91"){
            postCode =
                ↪ this.packagesData[y][0].states.state[single].postcode;
            postName =
                ↪ this.packagesData[y][0].states.state[single].postoffice;
            textCell.innerHTML =
                ↪ this.packagesData[y][0].states.state[single].text + "---" +
                ↪ this.packagesData[y][0].states.state[single].postoffice;
        }else {textCell.innerHTML =
            ↪ this.packagesData[y][0].states.state[single].text;}
        row.appendChild(textCell);
        if(postCode != null){
            var row = document.createElement("tr");
            row.className = "emptyRow";
            this.table.appendChild(row);
            this.showPostOfficeInfo(postCode,postName);
        }
    }
    var row = document.createElement("tr");
    row.className = "emptyRow";
    this.table.appendChild(row);
}
return this.table;
```

V případě nesplnění výše uvedené podmínky o způsobu zobrazování zásilky, je vypsána informace pouze o posledním stavu zásilky a opět dochází ke kontrole, zdali se id posledního stavu nerovná hodnotě 81 nebo 91, a v případě že ano, je volána funkce `showPostOfficeInfo`.

Při průběhu for cyklu je mezi jednotlivé zásilky umístěn prázdný řádek tabulky, který tyto jednotlivé zásilky oddělí v tabulce. Po proběhnutí for cyklu dojde k jeho ukončení a celá tabulka je odeslána ke zobrazení na obrazovce.

```

}else {
    this.table = document.createElement("table");
    this.table.className = this.config.tableClass;
    this.showPostOfficeInfo(this.config.postCode, "");
    return this.table;
}
},

```

Pokud není splněna původní podmínka, ve které je kontrolováno, zdali je zadána webová adresa sdíleného Google Docs dokumentu a zároveň zdali tento dokument obsahuje nějaký text, je volána funkce showPostOfficeInfo s parametrem PSČ zadaného v konfiguračním souboru.

```

showPostOfficeInfo: function(postCode, postName){
if(this.config.showPostInfo != "none"){
    var temp = this.getPostCodeData(postCode);
    var row = document.createElement("tr");
    this.table.appendChild(row);
    var postNameCell = document.createElement("th");
    postNameCell.className = "postInfo";
    postNameCell.setAttribute("colspan", "2");
    if(postName === ""){
        postNameCell.innerHTML = this.translate("OPENING_HOURS") +
        ↪ temp[0].attributes.name;
    }else{postNameCell.innerHTML = this.translate("OPENING_HOURS")
    ↪ + postName;}
    row.appendChild(postNameCell);
if(this.config.showPostInfo === "all" && this.packages.length <=
↪ this.config.maxNumOfShownPackages){
    for(var d=0; d<7; d++){
        var row = document.createElement("tr");
        this.table.appendChild(row);
        var dayCell = document.createElement("td");
        dayCell.className = "openingDay";
        dayCell.innerHTML = temp[0].openingHours.day[d].name;
        row.appendChild(dayCell);
        var hoursCell = document.createElement("td");
        hoursCell.className = "openingHours";
        if(temp[0].openingHours.day[d].since3 === null &&
        ↪ temp[0].openingHours.day[d].to3 === null){
            if(temp[0].openingHours.day[d].since2 === null &&
            ↪ temp[0].openingHours.day[d].to2 === null){
                if(temp[0].openingHours.day[d].since1 != null
                && temp[0].openingHours.day[d].to1 !=
                ⇔ null){
                    hoursCell.innerHTML =
                    temp[0].openingHours.day[d].since1
                    + " - " +
                    ⇔ temp[0].openingHours.day[d].to1;
                }else{hoursCell.innerHTML =
                ↪ this.translate("CLOSED");}
            }else{hoursCell.innerHTML =
            temp[0].openingHours.day[d].since1 + " - " +
            temp[0].openingHours.day[d].to1 + " " +
            ⇔⇔⇔ temp[0].openingHours.day[d].since2 + " - " +
            ⇔⇔⇔ temp[0].openingHours.day[d].to2}
        }
    }
}
}

```

```

}else{hoursCell.innerHTML = temp[0].openingHours.day[d].since1
+ " - " + temp[0].openingHours.day[d].to1 + " " +
temp[0].openingHours.day[d].since2 + " - " +
temp[0].openingHours.day[d].to2 + " " +
⇕⇕⇕⇕
temp[0].openingHours.day[d].since3 + " - " +
⇕⇕⇕⇕
temp[0].openingHours.day[d].to3}
row.appendChild(hoursCell);
}

```

Funkce `showPostOfficeInfo` je volána rovnou při načítání modulu, pokud není zadána webová adresa Google Docs dokumentu, a nebo může být volána při vypisování informací o zásilkách. V konfiguraci tohoto modulu je možné nastavit skrytí otevírací doby poboček pošt. K tomuto účelu slouží proměnná `showPostInfo`. Pokud není u této proměnné nastavena hodnota `none`, funkce `showPostOfficeInfo` začne zobrazovat informace na základě vložených parametrů. V případě, že proměnná `showPostInfo` nabývá hodnoty `all`, je zobrazena kompletní otevírací doba.

```

}else {
var n = new Date();
var d = n.getDay() - 1;
var row = document.createElement("tr");
this.table.appendChild(row);
var dayCell = document.createElement("td");
dayCell.className = "openingDay";
dayCell.innerHTML = temp[0].openingHours.day[d].name;
row.appendChild(dayCell);
var hoursCell = document.createElement("td");
hoursCell.className = "openingHours";
if(temp[0].openingHours.day[d].since3 === null &&
⇕ temp[0].openingHours.day[d].to3 === null){
if(temp[0].openingHours.day[d].since2 === null &&
⇕ temp[0].openingHours.day[d].to2 === null){
if(temp[0].openingHours.day[d].since2 != null
&& temp[0].openingHours.day[d].to2 !=
⇕⇕ null){
hoursCell.innerHTML =
temp[0].openingHours.day[d].since1
+ " - " +
⇕⇕ temp[0].openingHours.day[d].to1;
}else{hoursCell.innerHTML =
⇕ this.translate("CLOSED");}
}else{hoursCell.innerHTML =
temp[0].openingHours.day[d].since1 + " - " +
temp[0].openingHours.day[d].to1 + " " +
⇕⇕⇕⇕
temp[0].openingHours.day[d].since2 + " - " +
⇕⇕⇕⇕
temp[0].openingHours.day[d].to2}
}else{hoursCell.innerHTML = temp[0].openingHours.day[d].since1
+ " - " + temp[0].openingHours.day[d].to1 + " " +
temp[0].openingHours.day[d].since2 + " - " +
temp[0].openingHours.day[d].to2 + " " +
⇕⇕⇕⇕
temp[0].openingHours.day[d].since3 + " - " +
⇕⇕⇕⇕
temp[0].openingHours.day[d].to3}
row.appendChild(hoursCell);
}
}
},

```

V případě, že proměnná `showPostInfo` nabývá hodnoty `latest`, dojde k zobrazení otevírací doby pouze pro daný den. Odpověď z API serveru může obsahovat až 3 rozmezí otevírací doby pro jeden den a je tedy nutné eliminovat prázdné možnosti pomocí podmínek.

### 9.1.1 CSS

Styl zobrazování modulu je specifikován v souboru `MMM-CzechPost.css`.

```
.packageText{
    text-align: center;
    font-weight: bold;
}
.packageId{ font-size: 1.25em;}
.price{ font-size: 1.15em; }
.stored{ font-size: 1.15em; }
.postInfo{
    font-size: 1.15em;
    border-bottom: 1px solid white;
}
.openingHours{ text-align: center; }
.czechPostWrapper{ font-size: 0.75em; }
.stored{ border-bottom: 1px solid white; }
.infoText{
    border-bottom: 1px solid white;
    font-size: 1.35em;
}
.emptyRow{
    height: 20px;
    background-color: black;
}
```

Výše uvedené prvky byly upraveny tak, aby byl výsledný text přehledný a lehce čitelný. Velikosti písma jsou zde měněny pomocí jednotek `em`, které mění svoji velikost v závislosti na výchozím fontu elementů.

```
.MMM-CzechPost .xsmall {
    font-size: 12px;
    line-height: 17px;
}
.MMM-CzechPost .small {
    font-size: 15px;
    line-height: 20px;
}
.MMM-CzechPost .medium {
    font-size: 20px;
    line-height: 25px;
}
.MMM-CzechPost .large {
    font-size: 25px;
    line-height: 30px;
}
.MMM-CzechPost .xlarge {
    font-size: 30px;
    line-height: 35px;
    letter-spacing: -1px;
}
```

Ve výchozím css souboru, který je využit u platformy `MagicMirror2`, existují předvytvořené třídy, které upravují velikosti tabulek. Tyto třídy byly upraveny tak, aby lépe

pasovaly do celkového designu modulu.

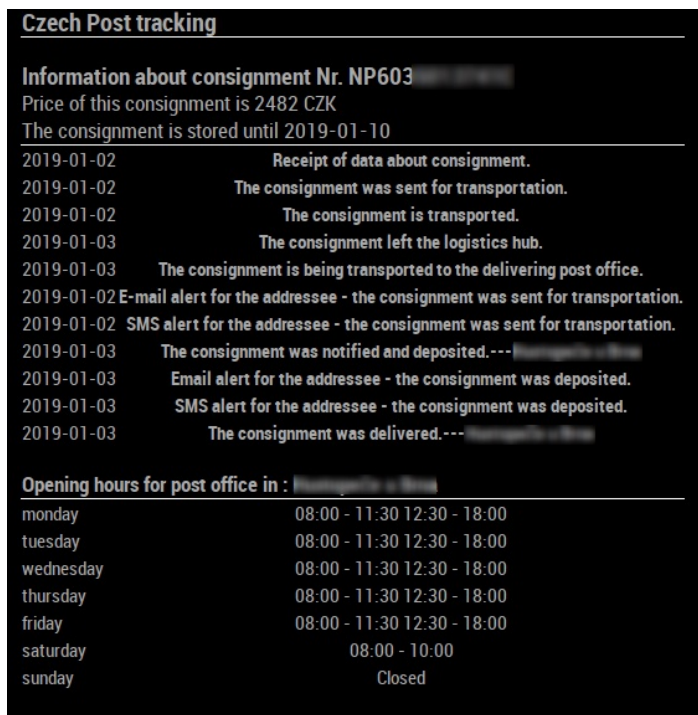
### 9.1.2 Překlady

Pomocí funkce `getTranslations`, která se nachází v hlavním JS souboru, je možné do modulu přidávat překlady. Tyto překlady se nacházejí ve složce `translations`. K jednotlivým přeloženým frázím je poté možné přistupovat pomocí klíčových slov. O tom, do jakého jazyka bude modul přeložen, rozhoduje jazyk, který je nastavený v hlavním konfiguračním souboru pod položkou `language`. V případě, že modul nedisponuje tímto jazykem, je jako hlavní jazyk modulu nastaven anglický jazyk.

```
{  
    "LOADING": "Načítání &hellip;",  
    "INFO": "Česká pošta - sledování zásilek",  
    "PACKAGE_INFO": "Informace o zásilce číslo ",  
    "PRICE": "Cena zásilky je ",  
    "STORED": "Zásilka je uložena do ",  
    "OPENING_HOURS": "Otevírací doba : ",  
    "CLOSED": "Zavřeno"  
}
```

### 9.1.3 Testování a Github

Při testování modulu bylo nutné vyzkoušet všechny možné konfigurace tak, aby při používání tohoto modulu nemohlo dojít k fatální chybě. Po provedení těchto nezbytných testů a opravení nalezených chyb je možné modul nahrát na platformu Github.



Czech Post tracking	
<b>Information about consignment Nr. NP603</b>	
Price of this consignment is 2482 CZK	
The consignment is stored until 2019-01-10	
2019-01-02	Receipt of data about consignment.
2019-01-02	The consignment was sent for transportation.
2019-01-02	The consignment is transported.
2019-01-03	The consignment left the logistics hub.
2019-01-03	The consignment is being transported to the delivering post office.
2019-01-02	E-mail alert for the addressee - the consignment was sent for transportation.
2019-01-02	SMS alert for the addressee - the consignment was sent for transportation.
2019-01-03	The consignment was notified and deposited.---
2019-01-03	Email alert for the addressee - the consignment was deposited.
2019-01-03	SMS alert for the addressee - the consignment was deposited.
2019-01-03	The consignment was delivered.---
<b>Opening hours for post office in :</b>	
monday	08:00 - 11:30 12:30 - 18:00
tuesday	08:00 - 11:30 12:30 - 18:00
wednesday	08:00 - 11:30 12:30 - 18:00
thursday	08:00 - 11:30 12:30 - 18:00
friday	08:00 - 11:30 12:30 - 18:00
saturday	08:00 - 10:00
sunday	Closed

Obr. 9.1 Modul MMM-CzechPost

## 9.2 Modul MMM-NameDay

Modul pro zobrazování jmenin, funguje na podobném principu, jako modul MMM-CzechPost. Na začátku kódu se nachází pole, které obsahuje výchozí definici proměnných.

```
defaults:{
    mode: "today", //today, tomorrow, yesterday, namedays,
    ↪ getdate,
    country: "", //us, cz, sk, pl, fr, hu, hr, se, at, it,
    ↪ de, es, none
    day: "", //1-31
    month: "", //1-12
    name: "",
    wrapperSize: "0.75em",
    textCellSize: "1.1em",
    updateInterval: 5*60*1000, //every 5 minutes
    initialLoadDelay: 3000,
    retryDelay: 5000,
    lang: config.language,
    tableClass: "small",
  },
```

Proměnná `mode` slouží ke specifikování dat, která mají být navrácena z API. Tato proměnná musí obsahovat jednu z následujících možností, aby byla zaručena správná funkčnost modulu.

- `today` - Zobrazení jmenin pro aktuální den.
- `tomorrow` - Zobrazení jmenin pro následující den.
- `yesterday` - Zobrazení jmenin předchozího dne.
- `namedays` - Zobrazení jmenin v závislosti na zadaném datu. V tomto případě je nutné, aby byly vyplněny proměnné `day` a `month`.
- `getdate` - Zobrazení data jmenin na základě vloženého jména. Je nutné, aby byla vyplněna proměnná `name`

Proměnná `country` slouží ke specifikování státu, pro který mají být jmeniny zobrazeny. Vyplnění této proměnné není důležité. V případě, že není tato proměnná vyplněna, jsou zobrazovány informace pro všechny státy, uvedené v seznamu níže. Dále je zde možné zadat jeden nebo více států a to pomocí specifického kódu státu. V případě zadávání více států, je nutné tyto státy zadat jako pole (do hranatých závorek). Mezi podporované možnosti patří: `us` (Spojené státy americké), `cz` (Česká republika), `sk` (Slovenská republika), `pl` (Polská republika), `fr` (Francouzská republika), `hu` (Maďarská republika), `hr` (Chorvatská republika), `se` (Švédské království), `at` (Rakouská republika), `it` (Italská republika), `de` (Spolková republika Německo) a `es` (Španělské království).



Další možností je `default`, která nastaví stát v závislosti na jazyce modulu, který je specifikován v hlavním konfiguračním souboru platformy.

Dále je možné pomocí proměnných `wrapperSize` a `textCellSize` měnit velikost textu, který bude zobrazován na obrazovce. Proměnná `updateInterval` umožňuje nastavení intervalu pro obnovení informací na obrazovce. V případě, že při odesílání API požadavku dojde k chybě, je po uplynutí časového intervalu, který je specifikovaný pomocí proměnné `retryDelay` opět proveden požadavek na server.

```
start: function() {
    Log.info("Starting module: " + this.name);
    this.names = [];
    this.loaded = false;
    this.scheduleUpdate(this.config.initialLoadDelay);
    this.updateTimer = null;
    var url;
    if(this.config.country === "default"){
        if(this.config.lang === "en"){
            this.config.country = "us";
        }else if(this.config.lang === "cs"){
            this.config.country = "cz";
        }else if(this.config.lang === "sv"){
            this.config.country = "se";
        }else if(Array.isArray(this.config.country)==false){
            this.config.country = this.config.lang;
        }
    }
},
```

Funkce `start` slouží k inicializaci modulu. V průběhu této funkce dochází k definování globálních proměnných. Pokud je v proměnné `country` nastavena možnost `default`, dojde k nastavení státu v závislosti na jazyce platformy. V některých případech je možné, že zkratka jazyka není shodná s kódem požadovaného státu, a proto zde dochází k případnému upravení kódu státu. Také zde dochází k volání funkce `scheduleUpdate`, která slouží k automatickému obnovování modulu. Ve funkci `scheduleUpdate` dále dochází k volání funkce `getData`.

```
getData: function(){
    var self = this;
    var retry = true;
    var dataRequest = new XMLHttpRequest();
    var url = "https://thingproxy.freeboard.io/fetch/" +
        ↪ this.getURL();
    dataRequest.open("GET",url,false);
    dataRequest.onreadystatechange = function() {
        if(this.readyState === 4){
            if(this.status >= 200 && this.status < 400){
                self.names = JSON.parse(this.response);
            } else {
                Log.error(self.name + ": Could
                    ↪ not load name days.");
            }
            if (retry) {
                self.scheduleUpdate((self.loaded) ? -1 :
                    ↪ self.config.retryDelay);
            }
        }
    }
}
```

```

    }
};
    dataRequest.send();
    this.loaded = true;
    this.updateDom();
},

```

Funkce `getData` slouží k získání dat pomocí metody `XMLHttpRequest`. Je zde volána funkce `getURL`, která vytvoří webový odkaz, který je poté využit pro spojení se serverem. Po úspěšném vytvoření webového odkazu dochází k otevření spojení modulu se serverem pomocí metody `.open`, která obsahuje parametry `GET`, proměnnou obsahující webovou adresu serveru a parametr `false`. Parametr `GET` specifikuje metodu, která bude využita pro získání dat ze serveru a parametr `false` značí, že stažení dat ze serveru bude probíhat synchronně s během kódu, tedy kód bude dále pokračovat až po stažení dat. Pokud spojení se serverem z nějakého důvodu selhalo, dochází k opětovnému pokusu o připojení k serveru. Po úspěšném získání dat, jsou tato data vložena ve formátu `JSON` do globální proměnné `names` a je volána funkce `getDom`, která se stará o vykreslování informací na obrazovku.

```

getURL: function(){
    var baseUrl = "https://api.abalin.net/get/";
    if (this.config.mode === "today" || this.config.mode ===
        ↪ "tomorrow" || this.config.mode === "yesterday"){
        if(this.config.country !== "" &&
            ↪ Array.isArray(this.config.country)==false){
            url = baseUrl + this.config.mode + "?country="
                ↪ + this.config.country;
        }else { url = baseUrl + this.config.mode}
    } else if (this.config.mode === "namedays"){
        if(this.config.day !== "" && this.config.month !== ""){
            if(this.config.country !== "" &&
                ↪ Array.isArray(this.config.country)==false) {url =
                ↪ baseUrl + this.config.mode + "?day=" +
                ↪ this.config.day + "&month=" + this.config.month +
                ↪ "&country=" + this.config.country ;
            }else {url = baseUrl + this.config.mode +
                ↪ "?day=" + this.config.day + "&month=" +
                ↪ this.config.month;}
        }else {Log.error(self.name + ": Month or day not
            ↪ inserted!!");
            return;}
    } else if (this.config.mode === "getdate"){
        if(this.config.name !== "" && this.config.country !==
            ↪ ""){
            url = baseUrl + this.config.mode + "?name=" +
                ↪ this.config.name + "&calendar=" +
                ↪ this.config.country;
        }else {Log.error(self.name + ": Country not
            ↪ inserted!!");
            return;}
    } else {this.hide(1000, {lockString:this.identifier});
        return;}
    return url;
},

```



Webová adresa, ze které budou získávána data, je vytvořena v závislosti na možnostech zvolených v konfiguraci. V případě, že je u proměnné `mode` zvolena možnost `today`, `tomorrow` nebo `yesterday`, dochází ke kontrole, zdali je zadán kód státu v proměnné `country`. Není li v proměnné `country` zadán žádný kód země, obsahuje webová adresa pouze parametr zvolený v proměnné `mode`, a tím pádem dojde k zobrazení jmenin pro všechny státy. U možností `namedays` a `getdate` je dále kontrolováno, zdali jsou zadány všechny potřebné parametry pro tyto možnosti. V případě, že nějaký parametr není zadán, nebo je zadán chybně, dochází k vypsání chybové hlášky do konzole. V případě, že jsou všechny parametry zadány správně, je výsledná adresa uložena do proměnné `url` a odeslána zpět do funkce `getData`, kde je poté využita k získání dat z API.

```
getDom: function(){
    var wrapper = document.createElement("div");
    wrapper.className = "nameDayWrapper";
    wrapper.style.fontSize = this.config.wrapperSize;
    var countriesCodes = ["us", "cz", "sk", "pl", "fr", "hu",
        ↪ "hr", "se", "at", "it", "de", "es"];
    var skip = false;
    if (this.config.mode === "") {
        wrapper.innerHTML = this.translate("SET_CORRECT_MODE")
            ↪ + this.name + ".";
        wrapper.className = "dimmed light small";
        return wrapper;
    }
    if (!this.loaded) {
        wrapper.innerHTML = this.translate("LOADING");
        wrapper.className = "dimmed light small";
        return wrapper;
    }
}
```

Funkce `getDom` slouží k vykreslení získaných dat na obrazovku. Ze začátku je vytvořen hlavní element, který bude obsahovat všechny informace a také dochází k vytvoření proměnné `countriesCodes`, která obsahuje kódy všech dostupných zemí. V případě, že je zadán špatný mód pro zobrazení, dojde k vypsání chybové hlášky na obrazovku. Na obrazovce je také zobrazován text načítání, který zmizí v momentě, kdy jsou získána data z API serveru.

```
if(this.config.mode === "getdate"){
    var table = document.createElement("table");
    table.className = this.config.tableClass;
    var row = document.createElement("tr");
    table.appendChild(row);
    var textCell = document.createElement("td");
    textCell.className = "textCell";
    textCell.setAttribute("colspan", "2");
    textCell.style.fontSize = this.config.textCellSize;
    textCell.innerHTML = this.translate("NAMEDAY_GETDATE");
    row.appendChild(textCell);
    for(var x = 0; x < this.names.results.length; x++){
        var row = document.createElement("tr");
        table.appendChild(row);
        var countryCell = document.createElement("td");
        countryCell.className = "date";
```

```

        countryCell.innerHTML =
            this.names.results[x].day + "." +
            ↪ this.names.results[x].month;
        row.appendChild(countryCell);
        var namesCell = document.createElement("td");
        namesCell.className = "name";
        namesCell.innerHTML =
            ↪ this.names.results[x].name;
        row.appendChild(namesCell);
    }
    ↪ return table;
}

```

Funkce `getDom` je rozdělena na 3 hlavní podmínky. Ve výše uvedené podmínce dochází ke kontrole, zdali je zvolen mód `getdate`. Je-li podmínka splněna, dochází k vytvoření tabulky, která bude obsahovat informace pro vykreslení na obrazovku. Dále je vytvořen řádek tabulky, který obsahuje přeložený nadpis tabulky, a poté jsou vytvořeny další řádky tabulky, obsahující informace ze získaného JSON dokumentu. Po vytvoření je tabulka zobrazena na obrazovku.

```

if(this.config.country === "" ||
    ↪ Array.isArray(this.config.country)==true || skip == true){
    var table = document.createElement("table");
    table.className = this.config.tableClass;
    if(this.config.mode === "today"){
        var text = this.translate("NAMEDAY_TODAY_TABLE");
    } else if(this.config.mode === "tomorrow")
        {var text = this.translate("NAMEDAY_TOMORROW_TABLE");
    } else if(this.config.mode === "yesterday")
        {var text = this.translate("NAMEDAY_YESTERDAY_TABLE");
    } else if(this.config.mode === "namedays"){
    var text =
        this.translate("NAMEDAY_NAMEDAYS_TABLE").replace("$DAY$"
            ↪ ,this.config.day).replace("$MONTH$",this.config.month);
    }
    var row = document.createElement("tr");
    table.appendChild(row);
    var textCell = document.createElement("th");
    textCell.className = "textCell";
    textCell.setAttribute("colspan","2");
    textCell.style.fontSize = this.config.textCellSize;
    textCell.innerHTML = text;
    row.appendChild(textCell);
    if (Array.isArray(this.config.country)==true &&
        ↪ this.config.country.length > 0){
        showCountries = this.config.country;
    }else{var showCountries = countriesCodes;}
    countries = this.translate("COUNTRIES").split(",");
    for(var i = 0; i < showCountries.length ; i++){
        if(countriesCodes.indexOf(showCountries[i]) >
            ↪ -1){
            var countryIndex = countriesCodes.
                ↪ indexOf(showCountries[i]);
            var show = "name_" +
                ↪ countriesCodes[countryIndex];
            var row= document.createElement("tr");
            table.appendChild(row);
            var countryCell =
                ↪ document.createElement("td");
            countryCell.className = "country";
            countryCell.innerHTML =
                ↪ countries[countryIndex];

```

```

        row.appendChild(countryCell);
        var namesCell =
        ↪ document.createElement("td");
        namesCell.className = "name";
        namesCell.innerHTML =
        ↪ this.names.data[show];
        row.appendChild(namesCell);
    }
    else {Log.error(this.translate("ARRAY_ERROR")
    ↪ .replace("$WRONG_COUNTRY$", showCountries[i]));}
    }
    return table;
}

```

V případě, že není zadán žádný stát, nebo je zadáno více států, dochází k provedení výše uvedené podmínky. Je vytvořena nová tabulka, která v prvním řádku obsahuje nadpis, který je vytvořen v závislosti na zadaném módu a jazyce. V dalším kroku je kontrolováno, zdali je v proměnné `country` zadáno pole. Dále je také vytvořena proměnná `countries`, která obsahuje přeložené názvy států. V následujícím cyklu dochází k vypsaní požadovaných států do tabulky. Pokud je specifikováno více států, dochází ke zjištění indexů jednotlivých států, aby vykreslené informace odpovídaly překladům. Je-li jeden z kódů států zadaných v poli špatný, je tento kód ignorován. Výsledná tabulka je poté zobrazena na obrazovce.

```

if(this.config.country != "" &&
    countriesCodes.indexOf(this.config.country) > -1 &&
    ↪ Array.isArray(this.config.country)==false){
    var show = "name_" + this.config.country;
    if(this.config.mode === "today"){
    var message = this.translate("NAMEDAY_TODAY")
    ↪ .replace("$NAME$", this.names.data[show]);
    } else if(this.config.mode === "tomorrow")
    {var message = this.translate("NAMEDAY_TOMORROW")
    ↪ .replace("$NAME$", this.names.data[show]);
    } else if(this.config.mode === "yesterday")
    {var message = this.translate("NAMEDAY_YESTERDAY")
    ↪ .replace("$NAME$", this.names.data[show]);
    } else if(this.config.mode === "namedays"){
    var message = this.translate("NAMEDAY_NAMEDAYS")
    .replace("$DAY$", this.config.day)
    ↪ .replace("$MONTH$", this.config.month)
    ↪ .replace("$NAME$", this.names.data[show]);
    }
    wrapper.innerHTML = message;
    return wrapper;
} else {
    Log.error(this.translate("COUNTRY_ERROR"));
    skip = true;
}
},

```

V případě, že je zadán pouze jeden kód státu, dochází k provedení výše uvedené podmínky. Zde je zjišťován mód zvolený v proměnné `mode`. Na základě zjištěného módu dochází k získání potřebného překladu a pomocí metody `.replace` dochází k nahra-

zení části textu informacemi o jmeninách. V případě zadání špatného kódu státu, je vypsána chybová hláška na konzoli.

### 9.2.1 CSS

Soubor CSS, který je přiložen k tomuto modulu, obsahuje úpravy přednastavených velikostí tabulek tak, aby nedocházelo k případným chybám při zobrazování. Dále je zde také specifikována velikost mezery, která upravuje odsazení vypsáných jmen v tabulce.

```
.name{
    padding-left: 5%;
}
.MMM-NameDay .xsmall {
    font-size: 12px;
    line-height: 17px;
}
.MMM-NameDay .small {
    font-size: 15px;
    line-height: 20px;
}
.MMM-NameDay .medium {
    font-size: 20px;
    line-height: 25px;
}
.MMM-NameDay .large {
    font-size: 25px;
    line-height: 30px;
}
.MMM-NameDay .xlarge {
    font-size: 30px;
    line-height: 35px;
    letter-spacing: -1px;
}
```

### 9.2.2 Překlady

Všechny možné překlady jsou specifikovány pomocí funkce `getTranslations`, která se nachází v hlavním JS souboru. Jazyk, který je nastavený v hlavním konfiguračním souboru platformy specifikuje, který překlad bude využit pro tento modul. V případě, že překlady, které jsou dostupné pro tento modul, neobsahují daný jazyk, je jako výchozí využit anglický jazyk. Překlady jsou v kódu aplikovány pomocí metody `translate` a klíčového slova. Jelikož je slovosled jednotlivých jazyků rozdílný, je zde využito parametrů `$NAME$`, `$DAY$`, `$MONTH$` a `$WRONG_COUNTRY$`, které jsou v hlavním kódu nahrazeny potřebnými údaji a tím umožňují přehlednější a jednodušší zápis v hlavním kódu. K nahrazení je využito metody `.replace`. Níže je uveden příklad překladu do českého jazyka.

```
{
  "LOADING": "Načítání &hellip;",
  "SET_CORRECT_MODE" : "Nastavte správný mód v config souboru modulu: ",
  "NAMEDAY_GETDATE" : "Zobrazení data podle zadaného jména a kalendáře.",
  "NAMEDAY_TODAY" : "Dnes má svátek $NAME$",
}
```

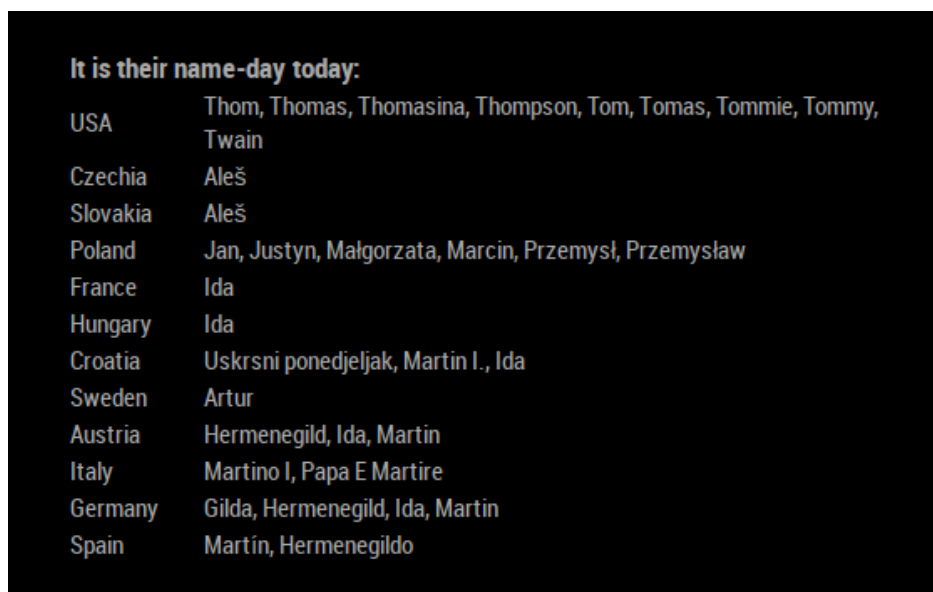
```

"NAME_DAY_TOMORROW" : "Zítřa má svátek $NAME$",
"NAME_DAY_YESTERDAY" : "Včera měl/a svátek $NAME$",
"NAME_DAY_NAMEDAYS" : "Dne $DAY$/ $MONTH$ má svátek $NAME$",
"NAME_DAY_TODAY_TABLE" : "Dnes mají svátek: ",
"NAME_DAY_TOMORROW_TABLE" : "Zítřa mají svátek: ",
"NAME_DAY_YESTERDAY_TABLE" : "Včera měli svátek: ",
"NAME_DAY_NAMEDAYS_TABLE" : "Dne $DAY$/ $MONTH$ mají svátek: ",
"COUNTRY_ERROR" : "Zadaná země není platná!! Zobrazuji všechny možné
↳ země."
"COUNTRIES" :
    "USA, Česko, Slovensko, Polsko, Francie, Maďarsko, Chorvatsko, Švédsko,
    ↳ Rakousko, Itálie, Německo, Španělsko",
"ARRAY_ERROR" : "Země $WRONG_COUNTRY$ není dostupná."
}

```

### 9.2.3 Testování

Modul byl řádně otestován tak, aby při řádném využívání nedocházelo k chybám. V případě, že dojde k chybě, je vypsána příslušná chybová hláška do konzole. Po provedení menších oprav bylo možné modul nahrát na GitHub.



Obr. 9.2 Modul MMM-NameDay

## 9.3 GitHub

Při nahrávání na platformu Github je nutné vytvořit repozitář (viz kapitola 4.1), který bude poté spojen s daným modulem. Po vytvoření tohoto repozitáře je možné modul nahrát. V prvním kroku je nutné provést inicializační nastavení složky s modulem pomocí příkazu `git init`. Po inicializaci je nutné vybrat soubory, které budou nahrány do repozitáře. Pomocí příkazu `git status` je možné zjistit stav souborů v této složce. Příkazem `git add` je možné vybrat soubory, které budou nahrány. Po vybrání souborů je možné vytvořit tzv. commit (příkaz `git commit`), který slouží k porovnání již existujících verzí.

tujících souborů v repozitáři s těmi, které mají být do repozitáře nahrány. Po vytvoření commitu je možné jej odeslat do repozitáře příkazem `git push`.

Modul MMM-CzechPost (Obr 9.1) je dostupný na webové adrese: <https://github.com/RichieCZ/MMM-CzechPost> a modul MMM-NameDay (Obr 9.2) je dostupný na webové adrese: <https://github.com/RichieCZ/MMM-NameDay>. Oba moduly je také možné nalézt v seznamu modulů třetích stran na GitHub stránce platformy MagicMirror2.

### 9.3.1 Dokumentace

V posledním kroku je nutné vytvořit dokumentaci, která bude potencionální uživatele informovat o možnostech nastavení a využívání těchto modulů. Vývojová dokumentace modulů pro platformu MagicMirror2 vyžaduje, aby vývojáři ke svému modulu vytvořili dokumentaci, která bude obsahovat základní popis modulu, informace o různých způsobech nastavení, instalační manuál a také fotku modulu.

Dokumentace pro modul MMM-CzechPost, která obsahuje postup instalace a seznam všech možných konfigurací je dostupná na webové adrese: <https://github.com/RichieCZ/MMM-CzechPost/blob/master/README.md>

Kompletní dokumentace pro modul MMM-NameDay se nachází na webové adrese: <https://github.com/RichieCZ/MMM-NameDay/blob/master/README.md>

## ZÁVĚR

Cílem bakalářské práce bylo navrhnout a implementovat nové rozšiřující moduly pro platformu MagicMirror2 a popsat způsob manuální instalace této platformy na zařízení Raspberry Pi jako doplněk k automatické instalaci.

V teoretické části se nachází popis jednotlivých prvků, které jsou potřebné k pochopení dané problematiky. V první kapitole se nachází popis platformy Raspberry Pi, který obsahuje informace o všech momentálně dostupných modelech tohoto zařízení, seznam dostupného příslušenství a základní popis postupu připojení periférií. Následující kapitola obsahuje popis operačního systému Raspbian. V této kapitole se nachází popis postupu instalace operačního systému Raspbian na zařízení Raspberry Pi a základní informace o tomto operačním systému. Třetí kapitola je věnována základnímu popisu pojmu open source. Ve čtvrté kapitole se nachází popis platformy GitHub, včetně seznamu základních příkazů. Předposlední kapitola popisuje jednotlivé webové technologie, které jsou využívány u platformy MagicMirror2. V poslední kapitole je popsána platforma MagicMirror2, její historie a základní přehled hlavních částí této platformy.

Praktická část se nejprve věnuje popisu všech možných způsobů instalace platformy MagicMirror2. Dále se v tomto popisu nachází informace o konfiguraci této platformy, společně s postupem instalace nových modulů a návodem pro aktualizaci platformy. Druhým bodem je analýza již existujících modulů a návrh nových modulů, které by podpořily vývoj GitHub projektu MagicMirror2 jeho rozšířením. V poslední části je podrobně popsán vývoj těchto modulů, které získávají data z veřejně dostupných API, a poté vykreslují získané informace na obrazovku zrcadla. Výsledkem jsou dva moduly, jeden nazvaný MMM-CzechPost, který slouží k zobrazování informací o zásilkách, přepravovaných Českou poštou. Druhý modul nese název MMM-NameDay, a ten slouží k zobrazování jmenin, dle zadaných parametrů a je schopen zobrazovat informace i pro některé cizí státy. V závěrečné fázi této části bylo provedeno testování a doladění těchto modulů, byla vytvořena dokumentace a výsledné moduly byly nahrány na GitHub a přidány do seznamu modulů na GitHub stránce platformy MagicMirror2.

## SEZNAM POUŽITÉ LITERATURY

- [1] HALFACREE, Gareth. *The official Raspberry Pi beginner's guide : how to use your new computer*. Cambridge, UK: Raspberry Pi Press, 2016. ISBN 978-1912047680.
- [2] UPTON, Eben a Gareth HALFACREE. *Raspberry Pi user guide*. 4th edition. Chichester, West Sussex, UK: John Wiley, 2016. ISBN 978-1119264361.
- [3] MBTECHWORKS. Raspberry Pi Model Comparison Table. *MBTechWorks - Raspberry Pi Articles, Examples, Guides and Programming* [online]. 2018 [cit. 2019-03-27]. Dostupné z: <https://www.mbtechworks.com/hardware/raspberry-pi-model-comparison.html>
- [4] PILTCH, Avram. Raspberry Pi 4: Release Date, Specs, Price, Everything We Know. *Tom's Hardware* [online]. 2019 [cit. 2019-03-29]. Dostupné z: <https://www.tomshardware.com/news/raspberry-pi-4-everything-we-know,38539.html>
- [5] What is open source software? | *Opensource.com*. *Opensource.com* | *Opensource.com* [online]. Copyright ©2019 Red Hat, Inc. [cit. 03.04.2019]. Dostupné z: <https://opensource.com/resources/what-open-source>
- [6] What Is GitHub? A Beginner's Introduction to GitHub. *Kinsta Managed WordPress Hosting - Powered By Google Cloud* [online]. Copyright © Kinsta Ltd. All rights reserved. [cit. 03.04.2019]. Dostupné z: <https://kinsta.com/knowledgebase/what-is-github/>
- [7] PIPINELLIS, Achilleas. *GitHub Essentials*. Birmingham: Packt Publishing, 2015. ISBN 978-1783553716.
- [8] Basic GIT Commands. *The Hosting Platform Made For You - Go Online With Hostinger* [online]. Copyright © 2004 [cit. 30.04.2019]. Dostupné z: <https://www.hostinger.com/tutorials/basic-git-commands>
- [9] The Basics of Web Technologies. *alphadevx.com* [online]. Copyright © 2001 [cit. 03.04.2019]. Dostupné z: <http://www.alphadevx.com/a/7-The-Basics-of-Web-Technologies>
- [10] Hypertext Markup Language. *Wikipedie: Otevřená encyklopedie* [online]. Copyright © 2019 [cit. 03.04.2019]. Dostupné z: [https://cs.wikipedia.org/w/index.php?title=Hypertexty\\_Markup\\_Language](https://cs.wikipedia.org/w/index.php?title=Hypertexty_Markup_Language)



- [11] Tvorba webu - CSS - kaskádové styly. *Tvorba webu* [online]. Dostupné z: <http://tvorba-webu.webtvor.net/css.php>
- [12] What Is JavaScript and How Does It Work?. *MakeUseOf - Technology, Simplified* [online]. Copyright © 2019 MakeUseOf. All Rights Reserved. [cit. 03.04.2019]. Dostupné z: <https://www.makeuseof.com/tag/what-is-javascript/>
- [13] FLANAGAN, David. *JavaScript: the definitive guide*. 6th ed. Sebastopol, CA: O'Reilly, 2011. ISBN 978-0596805524.
- [14] Document Object Model. *Wikipedie: Otevřená encyklopedie* [online]. Copyright © 2017 [cit. 03.04.2019]. Dostupné z: [https://cs.wikipedia.org/w/index.php?title=Document\\_Object\\_Model](https://cs.wikipedia.org/w/index.php?title=Document_Object_Model)
- [15] JQuery. *Wikipedia: The Free Encyclopedia* [online]. Copyright © 2019 [cit. 03.04.2019]. Dostupné z: <https://en.wikipedia.org/w/index.php?title=JQuery>
- [16] Node.js. *Wikipedia: The Free Encyclopedia* [online]. Copyright © 2019 [cit. 05.04.2019]. Dostupné z: <https://en.wikipedia.org/w/index.php?title=Node.js>
- [17] Introduction to Node.js. Introduction to Node.js [online]. Copyright © 2019 [cit. 05.04.2019]. Dostupné z: <https://nodejs.dev>
- [18] Electron (software framework). *Wikipedia: The Free Encyclopedia* [online]. Copyright © 2019 [cit. 05.04.2019]. Dostupné z: [https://en.wikipedia.org/w/index.php?title=Electron\\_\(software\\_framework\)](https://en.wikipedia.org/w/index.php?title=Electron_(software_framework))
- [19] Electron Documentation. *Electron* [online]. [cit. 2018-11-19]. Dostupné z: <https://electronjs.org/docs/all>
- [20] API. *Wikipedie: Otevřená encyklopedie* [online]. Copyright © 2018 [cit. 06.04.2019]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=API>
- [21] HTTP Methods [ RESTful APIs Verbs ] – REST API Tutorial. *What is REST – Learn to create timeless RESTful APIs* [online]. Dostupné z: <https://restfulapi.net/http-methods/>
- [22] Chrome DevTools. *Google Developers* [online]. Copyright © 2019 [cit. 06.04.2019]. Dostupné z: <https://developers.google.com/web/tools/chrome-devtools/>
- [23] GitHub - MichMich/MagicMirror: MagicMirror<sup>2</sup>. *The world's leading software development platform* · GitHub [online]. Copyright © 2019 [cit. 06.04.2019]. Dostupné z: <https://github.com/MichMich/MagicMirror>

- 
- [24] Xonay Labs | Michael Teeuw. *Xonay Labs / Michael Teeuw* [online]. Dostupné z: <https://michaeltteeuw.nl>
- [25] Front door mirror | MagicMirror Forum. *Home / MagicMirror Forum* [online]. Dostupné z: <https://forum.magicmirror.builders/topic/768/front-door-mirror>
- [26] 3rd Party Modules. *MagicMirror2* [online]. GitHub [cit. 07.04.2019]. Dostupné z: <https://github.com/MichMich/MagicMirror/wiki/3rd-party-modules>
- [27] *MagicMirror2 Module Development Documentation* [online]. [cit. 07.04.2019]. Dostupné z: <https://github.com/MichMich/MagicMirror/tree/master/modules>
- [28] Home | MagicMirror Forum. *Home / MagicMirror Forum* [online]. Dostupné z: <https://forum.magicmirror.builders>
- [29] Complete Raspberry Pi Magic Mirror Tutorial | Magic Mirror Central. *Magic Mirror Central | The #1 Website for Smart Mirror Making* [online]. Copyright © 2019 [cit. 16.04.2019]. Dostupné z: <https://www.magicmirrorcentral.com/complete-raspberry-pi-magic-mirror-tutorial/>
- [30] Auto Starting MagicMirror · MichMich/MagicMirror Wiki · GitHub. *The world's leading software development platform · GitHub* [online]. Copyright © 2019 [cit. 17.04.2019]. Dostupné z: <https://github.com/MichMich/MagicMirror/wiki/Auto-Starting-MagicMirror>
- [31] Configuring the Raspberry Pi · MichMich/MagicMirror Wiki · GitHub. *The world's leading software development platform · GitHub* [online]. Copyright © 2019 [cit. 17.04.2019]. Dostupné z: <https://github.com/MichMich/MagicMirror/wiki/Configuring-the-Raspberry-Pi>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

SoC	System-on-chip - systém na čipu
RAM	Random Access Memory - paměť s přímým přístupem
USB	Universal Serial Bus - univerzální sériová sběrnice
SD	Secure Digital
HDMI	High-Definition Multimedia Interface
VGA	Video Graphics Array
DSI	Display Serial Interface
CSI	Camera Serial Interface
NOOBS	New Out Of the Box Software
OS	Operating System - Operační systém
LED	Light-Emitting Diode - elektroluminiscenční dioda
GPIO	General-purpose input/output
PDF	Portable Document Format
SSH	Secure Shell
VNC	Virtual Network Computer
OSI	Open Source Initiative
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
XML	Extensible Markup Language
W3C	World Wide Web Consortium
DOM	Document Object Model - objektový model dokumentu
API	Application Programming Interface - rozhraní pro programování aplikaci
JSON	JavaScript Object Notation
SQL	Structured Query Language
MM2	MagicMirror2
CORS	Cross-origin resource sharing
PSČ	Poštovní směrovací číslo

**SEZNAM OBRÁZKŮ**

Obr. 1.1	Raspberry Pi model A [3]	12
Obr. 1.2	Raspberry Pi model B+ [3]	13
Obr. 1.3	Raspberry Pi model 2 B [3]	13
Obr. 1.4	Raspberry Pi model 3 B+ [3]	14
Obr. 1.5	Raspberry Pi Zero W [3]	15
Obr. 2.1	Raspbian OS	17
Obr. 6.1	MagicMirror2 [24]	34
Obr. 6.2	MagicMirror2 zrcadlo [25]	35
Obr. 9.1	Modul MMM-CzechPost	56
Obr. 9.2	Modul MMM-NameDay	64