

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# DIPLOMOVÁ PRÁCE

Knihovní systém katedry



2021

Vedoucí práce:  
doc. Mgr. Jan Outrata, Ph.D.

Karel Panchártek

Studijní obor: Aplikovaná informatika,  
prezenční forma

## **Bibliografické údaje**

Autor: Karel Panchártek  
Název práce: Knihovní systém katedry  
Typ práce: diplomová práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2021  
Studijní obor: Aplikovaná informatika, prezenční forma  
Vedoucí práce: doc. Mgr. Jan Outrata, Ph.D.  
Počet stran: 59  
Přílohy: 1 CD/DVD  
Jazyk práce: český

## **Bibliographic info**

Author: Karel Panchártek  
Title: Department library system  
Thesis type: master thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2021  
Study field: Applied Computer Science, full-time form  
Supervisor: doc. Mgr. Jan Outrata, Ph.D.  
Page count: 59  
Supplements: 1 CD/DVD  
Thesis language: Czech

## Anotace

*Systém vznikl pro usnadnění a zpřehlednění evidence výpůjček v knihovně Katedry informatiky Univerzity Palackého. Systém umožňuje evidenci výpůjček primárně pomocí samoobslužného výpůjčního místa. Systém poskytuje rozhraní pro administrátora i možnost úpravy uživatelských profilů. Práce se zabývá celkovým vznikem systému včetně výběru hardwaru, instalací operačního systému, nastavením operačních systémů, zálohováním databází i samotným vývojem aplikace. Text obsahuje návod pro uživatele a administrátora. Dále obsahuje popis implementace využitelný i pro případné budoucí úpravy systému.*

## Synopsis

*The system was created to facilitate and clarify the records of borrowings in the library of the Department of Informatics of Palacký University. The system allows the registration of borrows primarily using a self-service borrow point. The system provides an interface for the administrator and the ability to edit user profiles. The thesis deals with the overall the creation of the system, including the selection of hardware, installation of the operating system, setting up operating systems, backing up databases and the development of the application itself. The text contains help for users and administrators. It also contains a description of the implementation usable for possible future modifications of the system.*

**Klíčová slova:** samoobslužná, knihovna, PHP, Javascript, HTML, CSS

**Keywords:** self-service, library, PHP, Javascript, HTML, CSS

Rád bych poděkoval za vedení a pomoc vedoucímu práce, kterým je pan doc. Mgr. Jan Outrata, Ph.D. Dále bych rád poděkoval Katedře informatiky jako instituci za poskytnutí vybraného hardwaru k vývoji systému. Za rychlou přípravu potřebné infrastruktury i pomoc s testováním bych rád poděkoval panu Ivu Fridrichovi. Děkuji.

*Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.*

datum odevzdání práce

podpis autora

# Obsah

<b>1</b>	<b>Úvod</b>	<b>8</b>
<b>2</b>	<b>Zpřesnění pojmů</b>	<b>9</b>
2.1	Role uživatelů systému . . . . .	9
2.2	Pojmy netýkající se rolí uživatelů . . . . .	9
<b>3</b>	<b>Původní evidence výpůjček</b>	<b>9</b>
<b>4</b>	<b>Předpokládaná podoba systému před implementací</b>	<b>10</b>
4.1	Odlišnosti systému od běžné knihovny . . . . .	11
<b>5</b>	<b>Technologie</b>	<b>11</b>
5.1	Scanner čárových kódů . . . . .	11
5.2	Dotyková obrazovka . . . . .	12
5.3	Raspberry Pi . . . . .	12
5.4	Server . . . . .	12
5.5	Síťová architektura . . . . .	13
5.6	Použité softwarové technologie . . . . .	13
<b>6</b>	<b>Instalace systému</b>	<b>13</b>
6.1	Postup pro RPI . . . . .	13
6.2	Postup pro veřejný server . . . . .	14
<b>7</b>	<b>Uživatelská nápověda pro výpůjční místo</b>	<b>14</b>
7.1	Obecné informace o aplikaci . . . . .	15
7.2	Přihlášení uživatele . . . . .	15
7.3	Katalog knih . . . . .	16
7.4	Detail knihy . . . . .	16
7.5	Nahlásit chybu . . . . .	16
7.6	Zadání výpůjčky knihy . . . . .	16
7.7	Ukončení výpůjčky knihy . . . . .	17
7.8	Historie výpůjček . . . . .	19
7.9	Odhlásit . . . . .	19
<b>8</b>	<b>Uživatelská nápověda pro nastavení uživatelského profilu</b>	<b>19</b>
8.1	Obecné . . . . .	20
8.2	Katalog . . . . .	20
8.3	Detail knihy . . . . .	20
8.4	Přihlášení uživatele . . . . .	20
8.5	Nastavení profilu . . . . .	20
8.6	Připravený seznam . . . . .	21
8.7	Historie výpůjček . . . . .	22
8.8	Nahlásit chybu . . . . .	22
8.9	Odhlásit . . . . .	22

<b>9</b>	<b>Návoděda pro rozhraní hlavního knihovníka</b>	<b>23</b>
9.1	Obecné informace o rozhraní . . . . .	23
9.2	Úvodní stránka . . . . .	23
9.3	Knihy . . . . .	23
9.4	Uživatelé . . . . .	25
9.5	Výpůjčky . . . . .	25
9.6	Nahlásit chybu . . . . .	27
<b>10</b>	<b>Vstupní data</b>	<b>27</b>
<b>11</b>	<b>Databáze</b>	<b>28</b>
11.1	Zrcadlení databáze . . . . .	28
11.2	Obsah databáze . . . . .	31
11.3	Řešení problému rozdílných záznamů v databázích . . . . .	36
<b>12</b>	<b>Popis skriptů</b>	<b>36</b>
12.1	Složky a skripty pro výpůjční místo (RPI) . . . . .	37
12.2	Složky a skripty pro veřejný server . . . . .	43
<b>13</b>	<b>Zálohování, synchronizace a obnovení po havárii</b>	<b>51</b>
13.1	Obnovení na veřejném serveru . . . . .	51
13.2	Obnovení na výpůjčním místě . . . . .	51
<b>14</b>	<b>Zabezpečení systému</b>	<b>52</b>
<b>15</b>	<b>Hromadné načtení obrázků obálek</b>	<b>52</b>
<b>16</b>	<b>Možná rozšíření a změny</b>	<b>53</b>
<b>17</b>	<b>Diskuse</b>	<b>54</b>
	<b>Závěr</b>	<b>56</b>
	<b>Conclusions</b>	<b>57</b>
	<b>A Obsah příloženého disku</b>	<b>58</b>
	<b>Literatura</b>	<b>59</b>

## Seznam obrázků

1	Výpůjční místo – úvodní strana . . . . .	18
2	Výpůjční místo – obrazovka po přihlášení . . . . .	18
3	Výpůjční místo – katalog . . . . .	19
4	Editace údajů v nastavení profilu . . . . .	22
5	Výchozí stránka po přihlášení hlavního knihovníka . . . . .	26
6	Rozhraní hlavního knihovníka – správa kategorií . . . . .	26

# 1 Úvod

Vzniklý systém umožňuje evidenci výpůjček v knihovně Katedry informatiky Univerzity Palackého. Systém je zjednodušeně rozhraním k databázi. Toto rozhraní má dvě části. První část je poskytována samoobslužným výpůjčním místem. Druhá část je poskytována administrátorským rozhraním a editací uživatelských profilů přístupných skrze webové rozhraní na veřejné doméně.

Administrátorské rozhraní a rozhraní pro evidenci uživatelských účtů je poskytováno serverem s veřejnou IP adresou. Tento server budu dále nazývat veřejný server, popř. pouze server.

Administrátorské rozhraní je určeno pro hlavního knihovníka, popř. hlavní knihovníky. Hlavní knihovník je termín, který budu používat pro tuto roli. V systému je přihlašovací jméno `admin`. Na tento účet není možno evidovat výpůjčky. Tento účet poskytuje přehled o stavu databázi. Databáze jsou dvě a jejich vzájemná spolupráce bude vysvětlena dále. Dále toto rozhraní umožňuje zobrazit chyby hlášené uživateli, exportovat tabulky databáze do CVS souborů, prohlížení, tvorbu a editaci záznamů o knihách, prohlížení, tvorbu a editaci záznamů o kategoriích knih, prohlížení, tvorbu a editaci záznamů o uživatelích. Prohlížení všech výpůjček i výpůjček pro vybranou knihu nebo vybraného uživatele. Vytvoření a ukončení výpůjčky pro uživatele. Umožňuje i nahlášení chyby, což v případě administrátorského rozhraní je pro účel vytvoření poznámky.

Rozhraní pro editaci uživatelských profilů je podobné administrátorskému rozhraní. Služby jsou však poskytovány jen danému přihlášenému uživateli. Je poskytováno také veřejným serverem. Umožňuje prohlížení výpůjček. Neumožňuje však jejich vytvoření nebo ukončení. To je pro uživatele umožněno pouze na výpůjčním místě. Čili při fyzické přítomnosti uživatele a ideálně i knihy v knihovně. Dále umožňuje prohlížení záznamů o knihách. Umožňuje přípravu seznamu knih k vypůjčení, který může být zobrazen na výpůjčním místě. Umožňuje nastavení uživatelského profilu a nahlášení chyby, resp. poslání zprávy hlavnímu knihovníkovi.

Dále toto webové rozhraní poskytuje služby i nepřihlášenému uživateli. Může prohlížet katalog a detail jednotlivých knih.

Na veřejném serveru je také dostupná nápověda pro běžné uživatele i hlavního knihovníka.

Samoobslužné výpůjční místo, dále často označováno jako výpůjční místo, je jedno. Pouze zde je běžný uživatel schopen zavést do systému záznam o výpůjčce a vrácení knihy. Toto výpůjční místo má dotykovou obrazovku, která je připojena na počítač Raspberry Pi (dále také RPI), který dále komunikuje s veřejným serverem. K RPI je připojen skener čárových kódů.

Výpůjční místo slouží primárně k evidenci záznamů o výpůjčce. Knihy by měli být primárně identifikovány pomocí čárových kódů, které je možné skenovat skenerem čárových kódů. Je však možné i knihu najít v katalogu. To buď pomocí řazení sloupců s údaji nebo zadáním klíčových slov. Za klíčová slova jsou považovány některé údaje o knize. Konkrétně název, autor, rok vydání, katego-



rie i hodnoty ve sloupci nazvaném info, který nejčastěji odpovídá vydavatelství. Výpůjční místo umožňuje získat informaci o aktuální dostupnosti knihy, popř. kdo má knihu půjčenou. Je zde také možnost zobrazit nápovědu k výpůjčnímu místu.

Celý systém je založen na důvěře, stejně jako původní evidence výpůjček. Do knihovny mají přístup pouze zaměstnanci katedry. Ve výchozím nastavení nejsou na výpůjčním místě vyžadována hesla uživatelů. Uživatelé ale mají možnost si vyžadování hesla nastavit. U zaměstnanců se předpokládá, že nemají motivaci k útokům na tento systém, proto je i zaváděno co nejméně omezení a je snaha, aby systém byl co nejjednodušší a uživatelé měli možnost si knihu vypůjčit a vrátit v co nejméně krocích.

## 2 Zpřesnění pojmů

### 2.1 Role uživatelů systému

**Běžný uživatel** – uživatel, který není hlavním knihovníkem.

**Hlavní knihovník** – uživatel, který má oprávnění k měnění nastavení celého systému. V systému má účet s loginem `admin`. Je použit termín hlavní knihovník, protože ostatní uživatelé, kteří mají možnost si vypůjčovat knihy jsou vlastně zároveň i knihovníky ve smyslu možnosti vydávat knihy, zadávat výpůjčky do systému, vracet knihy do regálu, navrhopvat změny umístění knih, upozorňovat na nesrovnalosti reálného a evidovaného stavu.

**Administrátor** – administrace systému jako celku, operačního systému, nastavení skriptů, přístupových práv automatického přihlášení po spuštění systému atp.

### 2.2 Pojmy netýkající se rolí uživatelů

**RPI** – Raspberry Pi – počítač, který na kterém běží část systému pro výpůjční místo.

**Veřejný server, popř. pouze server** – počítač, který poskytuje webové rozhraní pro administrátora a uživatele. Je dostupný na veřejné IP adrese.

**Haš, hašování** a další tvary z tohoto základu. Bohužel jsem nenanešel, jaký správný tvar tohoto slova a slov odvozených pro české texty. V různých zdrojích se vyskytují i jiné tvary např. „hešování“, „hashování“. Používám tedy haš, hašování.

## 3 Původní evidence výpůjček

Do knihovny mají mít přístup pouze zaměstnanci katedry. Původní evidence knih byla v souboru ve formě tabulky. K tomuto souboru mají zaměstnanci katedry přístup. V tomto souboru jsou knihy, které jsou zaměstnancům katedry k dispozici. Soubor má přes 1600 řádků. V souboru jsou sloupce, ve kterých jsou

uvedeny následující údaje o knihách: název, autor, ISBN, info, signatura, kategorie, poznámka. Sloupec info obsahuje nejčastěji informaci o vydavatelství a rok vydání knihy. Umístění knihy je dáno kategorií knihy. Umístění kategorií knihy bylo možno najít v dalším souboru s tabulkou nebo je také k nahlédnutí v tištěné podobě přímo v knihovně. Kategorií je 61. Umístění kategorie je trojrozměrné. Je dáno policí, řadou a sloupcem v polici. Častější je stav, kdy je kategorie umístěna ve více buňkách.

Informaci, zda-li je kniha vypůjčena není zaměstnanec schopen zjistit před návštěvou knihovny. Není-li kniha vypůjčena, měl by zde knihu najít. V každé knize by měl být přítomen lístek knihy s údaji o knize. Knihu si zaměstnanec vypůjčí tak, že lístek z knihy spojí kancelářskou sponkou s lístkem s jeho jménem. V polici, kde by se kniha měla nacházet tedy zůstanou místo knihy tyto lístky. Takto ostatní zaměstnanci, poznají, že kniha je vypůjčená a kdo si knihu půjčil.

Pokud vše funguje jak má, tak je tento systém funkční. Slabinou je snad jen to, že zaměstnanec není schopen zjistit, jestli kniha v knihovně je před návštěvou knihovny. Najít umístění kategorie přímo v knihovně mezi takovým počtem kategorií nemusí být zcela rychlé. Pak jsou zde další slabiny, které tento systém ohrožují. Ponechané lístky se lehce mohou ztratit, přesunout. Evidence knih není jednotná. Jedním z problémů je i opomenutí zanechání lístků místo knihy v buňce. Nicméně tento problém zůstane u samoobslužného systému vždy, pokud nebude nějak technicky omezen. Například čipy, rámem ve dveřích a popřípadě ještě kamerovým záznamem. Taková omezení jsou ale v tomto případě nadbytečná. Obecně tento systém je a může být založen na důvěře. Mimo jiné se primárně počítá s tím, že uživatelé nebudou k zadání výpůjčky vyžadovat ochranu heslem.

## 4 Předpokládaná podoba systému před implementací

Vypůjčení knih bude probíhat v knihovně na počítači, ke kterému bude připojen dotykový displej, skener čárových kódů, klávesnice a popř. i myš. Tímto počítačem bude ideálně Raspberry Pi. Tento počítač bude spojen kabelem přímo se serverem, který bude poskytovat rozhraní, které bude umožňovat dálkový přístup do katalogu. Vypůjčení a vrácení knihy by mělo proběhnout v co nejméně krocích.

U každé knihy bude evidováno, v jaké je kategorii a kde je fyzicky kniha v knihovně umístěna. V knihovně jsou fyzicky přítomné otevřené skříně na knihy. Ve skříních jsou přítomny police, které jsou definovány souřadnicemi. Např. skříň vlevo police D3.

Hlavní knihovník by měl mít možnost při přidávání knihy načíst informace z online databáze.

Databáze by měla být dostupná na RPI i veřejném serveru i při ztrátě konektivity.

Vzdálené webové rozhraní pro administrátora. Vzdálené webové rozhraní pro uživatele k prohlížení katalogu.

## 4.1 Odlišnosti systému od běžné knihovny

Knihovny, které využívají nějaký software k evidenci většinou volí vlastní kódy, kdy každý jednotlivý svazek má vlastní kód. Pro čtení kódů bývají využita různá zařízení, podle volby nosiče kódu. Volen bývá nejčastěji čárový, QR nebo chip. Výpůjčky knih do systému zavádí knihovník, zaměstnanec knihovny. Stejně tak knihy jsou vráceny na místo knihovníkem. Některé knihovny umožňují vrácení knih do sběrných boxů. I potom ale výpůjčku musí v systému ukončit knihovník a zařadit knihu do police.

Ve vznikajícím systému bude důležité, aby systém byl samoobslužný, jednoduchý a bylo cílem vyhnout se přidělování kódů jednotlivým knihám. Také je rozdíl ve velikosti knihovny. Knihovna obsahuje asi 1600 svazků.

## 5 Technologie

### 5.1 Scanner čárových kódů

Vybraný scanner čárových kódů se po připojení k počítači chová jako automatická klávesnice. Vysílá kódy kláves jako poslední klávesu vysílá kód klávesy `enter`. Pro čísla jsou použity kódy kláves horní řady klávesnice. Tato řada je při českém rozložení využita převážně pro písmena s diakritikou. Takto vzniklý řetězec znaků je tedy přímo nepoužitelný. Skeneru je možno přepínat rozložení klávesnice, avšak jen na několik nejčastějších rozložení. České rozložení chybí. Je také možné skener přepnout na to, aby znaky vysílal pouze z numerické klávesnice. Na numerické klávesnici, ale není znak `x`, který se v ISBN-10 může vyskytovat. Řešením by mohlo být i přepnutí na anglické rozložení klávesnice. Na výpůjčním místě však bude připojena i klávesnice a je vhodné aby byla v českém rozložení. Rozložení klávesnice je na výpůjčním místě možno přepínat mezi českou a anglickou klávesnicí (klávesová zkratka `ctrl + shift`).

Problém s převodem českých znaků a rozlišením skenování od běžného vstupu z klávesnice je vyřešen ve skriptu `include/scanner.js`. Bylo zjištěno, že klávesy jsou ze skeneru vysílány s latencí do 200 milisekund. Nejrychlejší písáři na klávesnici mají do 1200 úhozů za minutu. Což je latence 50 milisekund. Avšak tohoto jsou schopni písáři dosáhnout při psaní na klávesnici všemi deseti prsty a asi ne za podmínek, jaké budou na výpůjčním místě. Dále kódy ISBN mají deset nebo třináct znaků. Na konci proudu skener posílá kód klávesy `13`, což odpovídá klávese `enter`. Proto skript rozlišuje zadání mezi deseti a čtrnácti klávesami, které byly zadány před klávesou `enter` a rozdíl mezi jednotlivými klávesami je do 200 milisekund. Pravděpodobnost, že by došlo běžným psaním ke splnění těchto podmínek je tedy naštěstí malá. Po rozpoznání, že došlo k naskenování kódu je řetězec znaků předán funkci, která převede české znaky na

čísla a zavolá funkci, kterou již bude obsahovat skript, do kterého je tento skript (`include/scanner.js`) vložen. Tato funkce již dále určí, jak naskenovaný čárový kód v daném skriptu obsloužit.

Nastavení skeneru se provádí pomocí čárových kódů, kterou jsou v návodu ke skeneru. Návod odpovídající tomuto skeneru je přiložen na disku. Nastavení oproti továrnímu, které jsem využil, bylo nastavení posílání pouze malých znaků abecedy. Nastavení si skener pamatuje i po odpojení od napájení, takže není potřeba jej případně opětovně nastavovat. Skener po naskenování kódu pípne. Pokud by to bylo žádoucí, lze toto výchozí nastavení změnit.

## 5.2 Dotyková obrazovka

Dotyková obrazovka byla zvolena s ohledem na dostatečnou velikost, pro možnost dostatečně velkého písma a i kompatibilitou s RPI. Po připojení nebyl problém s rozpoznáním událostí a jsou běžně rozpoznávány události odpovídající kliknutím myši. Proto nebylo nutné skripty dotykové obrazovce nijak přizpůsobovat.

## 5.3 Raspberry Pi

Raspberry Pi bylo dostupné přímo na katedře. Konkrétně jde o model `Raspberry Pi 4 Model B 4GB RAM`. Tato informace může být důležitá pro případnou instalaci celého systému. Jako operační systém bylo zvoleno `Ubuntu 20.04`<sup>[1]</sup> s grafickým rozhraním `Xfce4`<sup>[2]</sup>, někdy je používáno i označení `Xubuntu 20.04`. Avšak instalován musel být nejdříve čistý operační systém, čili `Ubuntu 20.04` a grafické rozhraní bylo instalováno až následně.

Použití RPI však sebou nese určitá omezení. Je poznat, že je méně výkonný než stolní počítač. Například se při testování ukázalo, že při načtení katalogu, který má asi 1500 položek má při renderování stránky větší latenci, než stolní počítač, kde je toto téměř neznatelné. A dále je u operačního systému poznat, že je to vedlejší architektura, takže případné problémy s balíky nejsou řešeny jako u dnes majoritní architektury `x86-64`.

## 5.4 Server

Hardware pro server byl na požádání přidělen správcem sítě i s nainstalovaným operačním systémem. Měl jsem možnost volby operačního systému a také zde jsem vybral `Ubuntu 20.04`. U veřejného serveru není grafické rozhraní. Dle informace správce sítě je tento server virtualizován. Avšak od tohoto jsem odstíněn a na implementaci tato informace neměla vliv. Hardwarové požadavky pro běh systému bych z dnešního pohledu při očekávaném zatížení hodnotil jako malé.

## 5.5 Síťová architektura

Veřejný server je připojen do sítě internet a má veřejnou IP adresu. Je tedy dostupný ze sítě internet. V budoucnu je však možné, že bude správcem sítě omezen přístup pouze z adres katedry informatiky. Veřejný server má ještě druhé síťové rozhraní, které je již v rámci další sítě s privátními adresami. Z tohoto rozhraní by měl být veden přímý kabel k výpůjčnímu místu. Měla by to tedy být privátní síť se dvěma počítači, kterou mám považovat za bezpečnou. Výpůjčnímu místu je skrze veřejný server umožněn přístup do sítě internet.

## 5.6 Použité softwarové technologie

Jako operační systém pro oba počítače byl zvolen Ubuntu 20.04. Tento operační systém je výhodný mimo jiné i kvůli vcelku dlouhé podpoře a bude podporován do roku 2025. Později ale již nebude možné jej udržovat aktuální. Jako grafické rozhraní operačního systému na RPI bylo zvoleno prostředí Xfce4. Operační systém umožňuje jednoduše nainstalovat LAMP server, který obsahuje PHP interpret i použitou MySQL[3] databázi. Skripty jsou psané v jazyce PHP 7.4[4]. PHP skripty generují HTML[5] stránky. Vzhled HTML stránek je ovlivněn i CSS[6]. V HTML stránkách je využit Javascript. Stránky jsou zobrazovány pomocí prohlížeče. Na výpůjčním místě byl jako výchozí prohlížeč zvolen Chromium, protože umožňuje spuštění přímo ve fullscreen módu. Pro webové rozhraní by mělo být možno použít libovolný prohlížeč, protože byla snaha tvořit validní HTML kód. Nejvíce vyzkoušeným prohlížečem pro toto rozhraní byl Firefox 90. Webové rozhraní primárně není určeno pro použití mimo desktop, avšak i při zkoušení v prohlížeči mobilního telefonu by se dalo označit za použitelné. Pro zálohování jsou využity i nástroje poskytované operačním systémem: BASH, OpenSSH, SSHFS, rsync, find, mysqldump, aj. K parsování obsahu cizích HTML stránek byl využit nástroj Xpath[7]. Technologie jsem zvolil především na základě dostupnosti dokumentace a znalostech, byť v některých případech počátku ne příliš hlubokých.

## 6 Instalace systému

Detailní popis zavedení systému je na přiloženém disku. Zde jsou v podkapitolách nastíněny pouze body. Při předání práce však systém již běží. V knihovně je výpůjční místo s dotykovou obrazovkou a skenerem čárových kódů. Veřejný server je dostupný na adrese knihy.inf.upol.cz.

### 6.1 Postup pro RPI

- Instalace operačního systému
- Nastavení firewallu

- Aktualizace systému
- Instalace balíčků
- Odinstalování `xfce4-screensaver`
- Přidání uživatele knihovna
- Úprava ikon na ploše a v panelu prostředí `Xfce4`
- Nastavení automatického startu aplikace
- Nastavení automatického přihlášení
- Kopírování skriptů a nastavení přístupových práv
- Nastavení MySQL
- Import databáze
- Nastavení SSH spojení se serverem
- Aktivace `crontab`

## 6.2 Postup pro veřejný server

Na veřejném serveru už byl nainstalován operační systém.

- Nastavení firewallu
- Aktualizace systému
- Instalace balíčků
- Přidání uživatelů
- Kopírování skriptů a nastavení přístupových práv
- Nastavení MySQL
- Import databáze
- Nastavení HTTPS přístupu
- Nastavení vynucení HTTPS

## 7 Uživatelská nápověda pro výpůjční místo

Pro představu jak vypadá uživatelské rozhraní jsou ke konci kapitoly vloženy obrázky. Více obrázků uživatelského rozhraní je možno nalézt na přiloženém disku.

## 7.1 Obecné informace o aplikaci

### Ukončení aplikace

Aplikace běží v prohlížeči, který se spouští ve fullscreen módu. Jednou možností ukončení je klávesová zkratka alt + F4. Další možností je ukončení fullscreen módu a zavření okna. Ukončení režimu přes celou obrazovku nebo případné opětovné spuštění je možné pomocí klávesy F11.

### Spuštění aplikace

Aplikace se spustí automaticky po spuštění systému ve fullscreen módu (přes celou obrazovku). Pokud je potřeba aplikaci spustit ručně, tak je zástupce pro spuštění přítomen na ploše. Další možností je otevřít prohlížeč a zadat adresu: localhost. Přepnutí aplikace do fullscreen módu jde pomocí klávesové zkratky prohlížeče F11.

### Oznámení po provedené uživatelské akci

U některých uživatelských akcí je zobrazen výsledek. Oznámení je nastaveno tak, aby bylo při dolním okraji obrazovky a příliš nerušilo další uživatelské akce. Oznámení je psáno světlým písmem na tmavém pozadí. Je možné jej zavřít kliknutím kdekoli v obrazovce aplikace. Oznámení samo zmizí po pěti vteřinách. Pokud má oznámení více řádků, nemusí být zobrazeno celé. Je to často spíše doplňková informace. Vzhledem ke krátkému času zobrazení nemusí být kompletní. Nekompletní je ale jen v případě půjčení knih, kdy je zobrazen počet půjčených nebo nepůjčených knih. V detailu oznámení je možné přecíst, kterých knih se to týká. Pokud by byl zájem si oznámení přecíst celé nebo znovu, je možnost jej zobrazit pomocí tlačítka `Zobrazit detail oznámení`. Toto tlačítko se objeví v pravém dolním rohu.

### Nastavení uživatelského profilu

Nastavení uživatelského profilu je možné na adrese knihy.inf.upol.cz. Je možné, že přístup na server bude umožněn jen z počítačů Katedry informatiky.

## 7.2 Přihlášení uživatele

K přihlášení je potřeba se dotknout jména na dotykové obrazovce. Jméno může být přítomno na první obrazovce v záložce `Častí uživatelé`, pokud je v uživatelském profilu aktivní volba `Častý` a `Aktivní`. Pokud je aktivní volba `Aktivní` a není aktivní volba `Častý`, tak je uživatel na obrazovce v záložce `Další uživatelé`. Pokud není aktivní volba `Aktivní`, není uživatel na těchto seznamech vůbec a nemůže se tedy na výpůjčním místě přihlásit. Tyto volby si je uživatel schopen nastavit sám v nastavení uživatelského profilu. Může je nastavit i hlavní knihovník. Po dotyku na jméno je uživatel dotázán na heslo pokud má

v uživatelském profilu aktivní volbu `Požadovat heslo`. Pokud je tato volba neaktivní je rovnou přihlášen. Tato volba nemá vliv na požadavek na heslo na veřejném serveru. Tam je heslo vyžadováno vždy.

### 7.3 Katalog knih

Do katalogu knih je možné nahlížet i bez přihlášení. Je zde možnost knihy vyhledávat psaním do textového pole `Vyhledat . . .`. Psaním vyhledávaného textu dochází k zužování výběru. Vyhledat knihu je možno podle všech zobrazených sloupců kromě sloupce, který ukazuje počet knih. V katalogu je možno knihy třídit podle některých sloupců dotykem na název sloupce. Z katalogu je možné nahlížet na detail knihy dotykem na její název.

### 7.4 Detail knihy

Na detail knihy je možno se podívat dotykem na název knihy v libovolném seznamu knih. Seznamy knih jsou katalog, tabulka vypůjčených knih, tabulka knih k půjčení, historie výpůjček. Dále je možno se na detail knihy podívat naskenováním knihy odkudkoli, kromě záložek `Nahlásit chybu` a `Zrychleně`. Pokud je uživatel přihlášen a knihu má půjčenou, může ji zde vrátit. Pokud je k dispozici kniha k půjčení, může ji přidat na seznam knih k vypůjčení.

### 7.5 Nahlásit chybu

Zde má uživatel možnost nahlásit chybu, poslat vzkaz hlavnímu knihovníkovi. Pokud je uživatel přihlášen, tak je vyplněno i jeho přihlašovací jméno. Toto je možno smazat. Nebo naopak u nepřihlášeného uživatele doplnit. Pole ISBN je po naskenování čárového kódu vyplněno jen tímto kódem. Jakékoli předešlé úpravy pole jsou přemazány. Je možno pole ISBN i vyplnit ručně, anebo nevyplňovat vůbec. Do pole text je možno skenovat ISBN kódy libovolně. Do textu se napíše hlášení, vzkaz hlavnímu knihovníkovi. Ne všechna hlášení se musí týkat konkrétní knihy.

### 7.6 Zadání výpůjčky knihy

Běžný uživatel si knihu může vypůjčit pouze u výpůjčního pultu v knihovně. Uživatel má více možností, jak do systému zavést záznam o vypůjčení. V nastavení uživatelského profilu má možnosti voleb, které ovlivňují průběh vypůjčení. Na průběh vypůjčení má vliv volba `Zrychlené půjčování`, podle které mohou nastat dva případy. Uživatel se přihlásí (vizte `Přihlášení uživatele`). Při aktivní volbě `Zrychlené půjčování` je uživatel po přihlášení ihned na obrazovce pod záložkou `Zrychleně`. Jinak je pod záložkou označenou jeho jménem.

V záložce `Zrychleně` je možno skenovat čárové kódy knih. Pokud má uživatel knihu půjčenu, je vrácena. Pokud ji půjčenu nemá, je vypůjčena. Takto může pokračovat dokud se neodhlásí, popř. nevybere jinou záložku.



V záložce označené jménem uživatele probíhá vypůjčení přes tvorbu seznamu knih k vypůjčení. Po naskenování kódu je zobrazen detail knihy. Zde si uživatel může knihu přidat na seznam k vypůjčení, pokud je kniha k dispozici. Po přidání na seznam knih k vypůjčení může pokračovat dál a skenovat další čárové kódy. Může odebrat jednotlivé knihy nebo všechny knihy ze seznamu. Knihy lze na seznam přidávat i bez skenování a to přes katalog nebo odjinud, odkud je možno otevřít detail knihy. V případě, že uživatel zadá volbu `Půjčit` u dané knihy na seznamu knih k vypůjčení je kniha vypůjčena. V případě, že zvolí volbu `Půjčit` vše jsou vypůjčeny všechny knihy ze seznamu.

Seznam knih k vypůjčení je možno vytvořit i v uživatelském profilu. Jak se tento seznam tvoří je popsáno v návodu k uživatelského profilu. Na výpůjčním místě v záložce označené jménem uživatele tlačítko `Načíst` připravený seznam - počet připravených knih: X. Tato volba způsobí načtení připraveného seznamu. Pokud byl dosud vytvořen nějaký seznam knih k vypůjčení, je tento původní seznam smazán a nahrazen připraveným seznamem. I po vypůjčení knih není připravený seznam modifikován. Modifikován může být jen v uživatelském profilu.

I přes nastavení v uživatelském profilu je možné na výpůjčním místě používat oba způsoby vypůjčení. Pro zrychlené půjčování stačí zvolit záložku `Zrychleně`. Naopak pro tvorbu seznamu zvolit záložku označenou jménem uživatele.

### **Zkráceně:**

**Zrychlené půjčování:** Přihlášení -> skenování knihy/knih

**Půjčování s tvorbou seznamu:** Přihlášení -> výběr z katalogu/skenování -> detail knihy -> přidat na seznam k půjčení -> půjčit knihu/půjčit všechny knihy v seznamu

**Půjčování pomocí připraveného seznamu:** Přihlášení -> tlačítko `Načíst` připravený seznam -> (editace seznamu) -> půjčit knihu/půjčit všechny knihy v seznamu

## **7.7 Ukončení výpůjčky knihy**

Běžný uživatel může knihu vrátit pouze u výpůjčního pultu v knihovně. Je více možností jak knihu/knihy vrátit. Po přihlášení uživatel vidí tabulku s vypůjčenými knihami. V této tabulce může přímo vracet jednotlivé knihy volbou `Vrátit` v řádcích tabulky. Nebo pomocí tlačítka `Vrátit` vše vrátit všechny knihy v seznamu vypůjčených knih. Pokud se uživatel nachází pod záložkou `Zrychleně`, může vypůjčenou knihu vrátit naskenováním jejího kódu. Další možností je knihu vrátit přes detail knihy, kam se uživatel může dostat naskenováním kódu knihy jinde než v záložkách `Zrychleně` nebo `Nahlásit chybu`. K detailu knihy se dá dostat i přes výběr knihy v katalogu či jiném seznamu knih.

<a href="#">Častí uživatelé</a>	<a href="#">Další uživatelé</a>	<a href="#">Katalog</a>	<a href="#">Nahlásit chybu</a>	<a href="#">Nápověda</a>
Jiří Balun	Miroslav Kolařík	Jan Laštovička	Jan Tříška	
Eduard Bartl	Alena Kolovrátníková	Tomáš Masopust	Tomáš Urbanec	
Radim Bělohávek	Jan Konečný	Tomáš Mikula	Jiří Valůšek	
Ivo Fridrich	David Kočíř	Petr Osička	Arnošt Večerka	
Radek Janoščík	Petr Krajča	Jan Outrata	Roman Vyjídaček	
Petr Jančar	Michal Krupka	Martin Trnečka	Jiří Zaccpal	
Jakub Juračka	Tomáš Kühř	Markéta Trnečková		

Obrázek 1: Výpůjční místo – úvodní strana

<a href="#">Outrata</a>	<a href="#">Katalog</a>	<a href="#">Zrychlené</a>	<a href="#">Historie výpůjček</a>	<a href="#">Nahlásit chybu</a>	<a href="#">Nápověda</a>	<a href="#">Odhlásit</a>
-------------------------	-------------------------	---------------------------	-----------------------------------	--------------------------------	--------------------------	--------------------------

### Vypůjčené knihy

Název	Autor	Umístění	Půjčeno od	
Computer networks :a system approach	Larry L. Paterson and Bruce S. Davie	Vpravo E3, G3	2021-08-11 23:21:39	<a href="#">Vrátit</a>
Introduction to data compression	Khalid Sayood	Vpravo D4	2021-08-11 23:21:59	<a href="#">Vrátit</a>
Handbook of data compression	David Salomon, Giovanni Motta ; with contributions by David Bryant	Vpravo D4	2021-08-11 23:22:21	<a href="#">Vrátit</a>

[Vrátit vše](#)

### Seznam knih k půjčení:

Název	Autor	Umístění	Půjčit	Odstranit ze seznamu
A course in cathogorical data analysis	Thomas Leonard with contributions by Orestis Papasouliotis	Střed F6	<a href="#">Půjčit</a>	<a href="#">Odstranit ze seznamu</a>
A course in model theory	Katrin Tent, Martin Ziegler	Střed D6	<a href="#">Půjčit</a>	<a href="#">Odstranit ze seznamu</a>
A course in model theory :an introduction to contemporary mathematical logic	Bruno Poizat ; translated by Moses Klein	Střed D6	<a href="#">Půjčit</a>	<a href="#">Odstranit ze seznamu</a>

[Půjčit vše](#) [Odstranit vše](#) [Načíst připravený seznam - počet připravených knih: 3](#)

Obrázek 2: Výpůjční místo – obrazovka po přihlášení

## Katalog knih katedry informatiky PŘF UP

Vyhledat ...						
Název	Autor	Rok	Signatura	Kategorie	Umístění	Kopíí volných/celkem
<a href="#">.NET Core in Action</a>	Dustin Metzgar	2018		Programovací jazyky	Vpravo E2, F2	0/1
<a href="#">*OS Internals.</a>	Jonathan Levin	2019	m2/1564-1a (přf)	Operační systémy	Vpravo D5, F5	1/1
<a href="#">*OS Internals.</a>	Jonathan Levin	2019	m2/1564-3a (přf)	Operační systémy	Vpravo D5, F5	1/1
<a href="#">3D printed science projects.</a>	Joan Horvath, Rich Cameron	2017	M2/1626 (přf)	Ostatní	Vlevo G1, H1	1/1
<a href="#">A book of abstract algebra</a>	Charles C. Pinter	2010	m1/3072 (přf)	Algebra	Střed C2	1/1
<a href="#">A Book of Abstract Algebra: Second Edition</a>	Charles C Pinter	2010	M1/3072a (přf)	Algebra	Střed C2	1/1

Obrázek 3: Výpůjční místo – katalog

### Zkráceně:

**První možnost:** Přihlášení -> vrácení knihy/knih ze seznamu vypůjčených knih

**Další možnost:** Po přihlášení v záložce Zrychleně Skenování knihy (Systém půjčenou knihu vrátí. Nepůjčenou půjčí.)

## 7.8 Historie výpůjček

Zobrazuje historii výpůjček. I zde je možnost knihy vracet.

## 7.9 Odhlásit

Odhlásí uživatele. Uživatel je automaticky odhlášen po deseti minutách. I tak je ale vhodné se odhlašovat po ukončení práce.

## 8 Uživatelská nápověda pro nastavení uživatelského profilu

Pro představu jak vypadá nastavení uživatelského profilu je ke konci kapitoly vložen obrázek. Více obrázků je možno nalézt na přiloženém disku.

## 8.1 Obecné

Veřejný server je dostupný na adrese `knihy.inf.upol.cz`. Je možné, že přístup na server bude umožněn jen z počítačů Katedry informatiky. Veřejný server poskytuje služby uživatelům k nastavení uživatelského profilu a přípravy seznamu knih, které si chce vypůjčit. A dále poskytuje rozhraní pro administrátora – hlavního knihovníka k nastavení systému.

U některých uživatelských akcí je zobrazen výsledek. Oznámení je nastaveno tak, aby bylo při dolním okraji obrazovky a příliš nerušilo další uživatelské akce. Oznámení je psáno světlým písmem na tmavém pozadí. Je možné jej zavřít kliknutím kdekoli v obrazovce aplikace. Oznámení samo zmizí po pěti vteřinách. Pokud má oznámení více řádků, nemusí být zobrazeno celé. Je to často spíše doplňková informace. Vzhledem ke krátkému času zobrazení nemusí být kompletní. Nekompletní je ale jen v případě půjčení knih, kdy je zobrazen počet půjčených nebo nepůjčených knih. V detailu oznámení je možné přečíst, kterých knih se to týká. Pokud by byl zájem si oznámení přečíst celé nebo znovu, je možnost jej zobrazit pomocí tlačítka `Zobrazit detail oznámení`. Toto tlačítko se objeví v pravém dolním rohu.

## 8.2 Katalog

I nepřihlášený uživatel může procházet katalog knihovny. Přihlášený uživatel má navíc možnost z katalogu přidávat knihy do připraveného seznamu knih pomocí volby `přidat` v posledním sloupci katalogu. Je zde možnost knihy vyhledávat psaním do textového pole `Vyhledat . . .`. Psaním vyhledávaného textu dochází k zužování výběru. Vyhledat knihu je možno podle všech zobrazených sloupců kromě sloupce, který ukazuje počet knih. V katalogu je možno knihy třídit podle některých sloupců kliknutím na název sloupce.

## 8.3 Detail knihy

I nepřihlášený uživatel může zobrazit detail knihy a získat informaci o počtu svazků a jejich dostupnosti.

Přihlášený uživatel navíc u půjčených svazků uvidí, kdo je má půjčené. Nepůjčené svazky má možnost si přidat do připravovaného seznamu knih.

## 8.4 Přihlášení uživatele

Proběhne po zadání jména a hesla. Po přihlášení získá uživatel informace o právě půjčených knihách.

## 8.5 Nastavení profilu

Na této obrazovce uživatel vidí informaci o aktuálním nastavení profilu. Tlačítko `Zobrazit výpůjčky` zde má stejnou funkci jako záložka `Historie`

výpůjček. Je možné zde změnit přihlašovací jméno a heslo. Přihlašovací jméno (login) musí být rozdílný od přihlašovacích jmen ostatních uživatelů. Pokus o změnu na existující přihlašovací jméno bude neúspěšný. Není možné měnit jméno uživatele, to může pouze hlavní knihovník.

Dále jsou zde možná nastavení chování výpůjčního místa. Volba `Požadovat heslo` - pokud je volba aktivní uživatel je po dotyku na své jméno na obrazovce na výpůjčním místě vyzván k zadání hesla. Ve výchozím nastavení je tato volba neaktivní. Uživatelé mají možnost si požadavek na heslo zapnout a budou poté při přihlašování na výpůjčním místě po dotyku na své jméno vyzváni k zadání hesla. Lehce se ale může stát, že při nepozornosti se omylem někdo může přihlásit na jiný účet. Na druhou stranu zadávání hesla zdržuje. Systém je založen na důvěře. Je možné knihu fyzicky odnést a záznam neudělat, takže motivace k záměrnému zavádění falešných záznamů je malá. Proto má uživatel možnost si zde vybrat. Volba `Požadovat heslo` nemá vliv na nutnost zadání hesla zde na webovém rozhraní k úpravě účtu. Zde je vyžadováno vždy.

Volba `Zrychlené půjčování` - pokud je tato volba aktivní je uživatel na přihlašovacím místě po přihlášení přesměrován na obrazovku `Zrychleně` odkud si může knihy půjčovat nebo vracet skenováním bez dalšího potvrzování. I při neaktivní volbě má uživatel možnost se do záložky `Zrychleně` dostat. A naopak i při neaktivní volbě si může půjčovat přes tvorbu seznamu knih k půjčení.

Volba `Častý a Aktivní`. Pokud jsou obě volby aktivní je jméno uživatele na přihlašovacím místě na první obrazovce, tedy záložce `Častí uživatelé`. Pokud je aktivní volba `Aktivní` a není aktivní volba `Častý`, tak je uživatel na obrazovce v záložce `Další uživatelé`. Pokud není aktivní volba `Aktivní`, není uživatel na těchto seznamech vůbec a nemůže se tedy na výpůjčním místě přihlásit.

## 8.6 Připravený seznam

Tato funkce umožňuje vytvořit seznam knih, který je možno načíst na výpůjčním místě. V této záložce je možno prohlédnout si, které knihy na seznamu jsou a knihy z něj odstraňovat. Kniha se na seznam přidává přímo z katalogu pomocí volby `přidat` v posledním sloupci katalogu. Je možno přidat i knihu, která je právě vypůjčena. Není však možno mít knihu na seznamu vícekrát, než je celkový počet kopií. Knihu je možno na seznam přidat i z detailu knihy. Odtud však není možno knihu na seznam přidat, pokud je vypůjčena. Použití připraveného seznamu na výpůjčním místě je možno dotykiem na tlačítko `Načíst připravený seznam - počet připravených knih:X`. Pokud byl dosud na výpůjčním místě vytvořen seznam knih k vypůjčení, je tento původní seznam smazán a nahrazen připraveným seznamem. I po vypůjčení knih není připravený seznam modifikován. Modifikován může být jen z nastavení uživatelského profilu.

Jan Outrata Katalog Nastavení profilu Připravený seznam (0) Historie výpůjček Nahlásit chybu nápověda Odehlásit

**Profil**

Jméno: Jan Outrata  
 Login: outrata  
 Požadovat heslo: ne  
 Zrychlené půjčování: ne  
 Častý: ano  
 Aktivní: ano

[Zobrazit výpůjčky](#)

**Změna profilu**

Login:  [změnit login](#)  
 Heslo:  [změnit heslo](#)  
 Heslo znovu:  [nastavit požadování hesla](#)  
 Požadovat heslo:  [nastavit zrychlené půjčování](#)  
 Zrychlené půjčování:  [nastavit častost](#)  
 Častý:  [nastavit aktivnost](#)  
 Aktivní:

Obrázek 4: Editace údajů v nastavení profilu

## 8.7 Historie výpůjček

Zobrazí historii výpůjček. Na tuto stránku je možno se dostat kliknutím na záložku **Historie výpůjček** nebo kliknutím na tlačítko **Zobrazit výpůjčky** v záložce **Nastavení profilu**. Oproti výpůjčnímu místu je zde navíc i informace, kdo výpůjčku do systému zadal. Výpůjčku a vrácení může zadat jen uživatel na výpůjčním místě nebo hlavní knihovník, zde označen jako **Admin**, na veřejném serveru.

## 8.8 Nahlásit chybu

Zde má uživatel možnost nahlásit chybu, poslat vzkaz hlavnímu knihovníkovi. Pokud je uživatel přihlášen, tak je vyplněno i jeho přihlašovací jméno. To je možno smazat. Nebo naopak u nepřihlášeného uživatele doplnit. I když se primárně nepočítá s přítomností skeneru u tohoto rozhraní. Může být použit i zde. Pole ISBN je po naskenování čárového kódu vyplněno jen tímto kódem. Jakékoli předešlé úpravy pole jsou přemazány. Je možno pole ISBN i vyplnit ručně, anebo nevyplňovat vůbec. Do pole text je možno skenovat ISBN kódy libovolně. Do textu se napíše hlášení, vzkaz hlavnímu knihovníkovi. Ne všechna hlášení se musí týkat konkrétní knihy.

## 8.9 Odhlásit

Dojde k odhlášení přihlášeného uživatele.

## 9 Náповěda pro rozhraní hlavního knihovníka

Pro představu jak vypadá rozhraní pro hlavního knihovníka jsou ke konci kapitoly vloženy obrázky. Více obrázků je možno nalézt na přiloženém disku.

### 9.1 Obecné informace o rozhraní

Toto rozhraní je dostupné komunikací s veřejným serverem. Je dostupné i na výpůjčním místě. V tom případě je potřeba přepnout z uživatelské aplikace a v běžném prohlížeči zadat adresu `knihy.inf.upol.cz`, jako kdekoli jinde.

U některých uživatelských akcí je zobrazen výsledek. Oznámení je nastaveno tak, aby bylo při dolním okraji obrazovky a příliš nerušilo další uživatelské akce. Oznámení je psáno světlým písmem na tmavém pozadí. Je možné jej zavřít kliknutím kdekoli v obrazovce aplikace. Oznámení samo zmizí po pěti vteřinách. Pokud má oznámení více řádků, nemusí být zobrazeno celé. Je to často spíše doplňková informace. Vzhledem ke krátkému času zobrazení nemusí být kompletní. Nekompletní je ale jen v případě půjčení knih, kdy je zobrazen počet půjčených nebo nepůjčených knih. V detailu oznámení je možné přecíst, kterých knih se to týká. Pokud by byl zájem si oznámení přecíst celé nebo znovu, je možnost jej zobrazit pomocí tlačítka `Zobrazit detail oznámení`. Toto tlačítko se objeví v pravém dolním rohu.

### 9.2 Úvodní stránka

Po přihlášení jsou zobrazeny informace o stavu knihovny. Dostupnosti databází a porovnání databází. Pokud je rozdíl v záznamech výpůjček mezi primární a sekundární databází je zobrazeno hlášení o chybě. Řešení tohoto problému je popsáno v textu diplomové práce v sekci `Řešení problému rozdílných záznamů v databázích`.

Dále jsou zde tlačítka pro export tabulek do CVS souborů. Jsou zde logy – hlášené chyby od uživatelů s možností je mazat či označit za vyřešené. V poslední části jsou záznamy ze souborů s logy. Tyto soubory jsou vytvářeny pokud dojde k rozdílným výsledkům dotazů, které mění nebo vytváří záznamy v tabulkách. V tom případě je možné, že je rozdíl mezi databázemi. Řešení tohoto problému je popsáno v textu diplomové práce v sekci `Řešení problému rozdílných záznamů v databázích`.

V poslední části je možnost změnit heslo pro účet admin.

### 9.3 Knihy

Katalog knih. Z této stránky je možno se dostat na stránku přidání nové knihy, správy kategorií knih a na detail vybrané knihy.

## Přidat novou knihu

Knihu jde přidat ručním vyplněním polí a kliknutím na tlačítko Přidat knihu. Další způsob je vyplnění pole ISBN a kliknutí na tlačítko Načti data o knize, pokud data o knize byla nalezena budou pole předvyplněna. ISBN zde je možno načíst pomocí skeneru. V době tvorby práce byl k dispozici pouze jeden skener. Takže je možnost knihy přidávat pomocí administrátorského rozhraní na výpůjčním místě. Nebo si skener na chvíli odnést na místo, kde budou knihy přidávány.

K přidání knihy je třeba vyplnit pole ISBN, Název, Počet svazků a vybrat kategorii knihy. Umístění knihy je určenou vybranou kategorií. Zde se vyplňuje pouze pokud je cílem mít u této knihy jiné umístění, než je umístění kategorie. Např. k určení přesné buňky v polici v rámci kategorie. K přehlednosti je však velmi vhodné vyplnit i nepovinná pole. ISBN musí být jedinečné. Nelze vložit knihu s ISBN, které je v databázi již evidováno. ISBN je potřeba uvádět bez pomlček a nejlépe toto pole vyplnit pomocí naskenování ISBN.

Po přidání knihy není ještě aktualizován obrázek na výpůjčním místě. Na výpůjčním místě bude aktualizován až po spuštění skriptu `mysql_dump_Backups/backup.sh` na výpůjčním místě. Skript se na výpůjčním místě spouští každý den. Nicméně pokud by byl požadavek, aby byly obrázky aktualizovány ihned, je třeba skript na výpůjčním místě spustit ručně.

## Spravovat kategorie

Na této stránce lze spravovat kategorie knih. Lze kategorii přidat, smazat změnit název a umístění. Pokud je změněn název kategorie, projeví se změna u všech knih, které tuto kategorii mají přiřazenou. Pokud je změněno umístění, změní se umístění u všech knih v kategorii, kromě knih, které mají umístění kategorie přepsáno vlastním umístěním. Kategorii lze smazat jen pokud není přiřazena žádná knize. Toto rozhraní neumožňuje hromadné změny kategorií ve smyslu rozdělení kategorie nebo sloučení kategorií. Pokud by byl zájem takové operace provádět lze to provést jen přímo nad databázemi. V tom případě je velmi vhodné přečíst si tu část diplomové práce, která popisuje, jak je databáze postavena a jak jsou databáze zrcadleny.

## Detail knihy

Na této stránce je možné prohlédnout si zaznamenané informace o knize a případně je měnit. Je to jediné místo, odkud hlavní knihovník může přiřadit výpůjčku knihy. Je zde možné i výpůjčku knihy ukončit. Pokud na úvodní stránce je signalizován rozdíl mezi databázemi, je lépe výpůjčku nezadávat. Pokud byl tento rozdíl způsoben dočasnou ztrátou konektivity, je možné, že z výpůjčního místa ještě nedorazili informace do sekundární databáze.

Pokud je zde u knihy přepsáno umístění, nebo i pokus o uložení současného. Dojde u knihy k přepsání umístění kategorie. Změny umístění kategorie už tedy



nebudou mít vliv ve výpisech aplikace na tuto knihu. Zpětné přiřazení umístění kategorie je možné změnou umístění na prázdný řetězec.

Pokud nesouhlasí řádek Počet volných exemplářů dle tabulky knihy a Počet volných exemplářů dle počtu výpůjček je možno a vhodné tento stav opravit stisknutím tlačítka Opravit.

Na této stránce je i možnost získání informací z Google pomocí tlačítka Načti data o knize z Google. Po načtení jsou získané informace předvyplněny v polích, pokud jsou načtené informace rozdílné od informací z databáze, jsou psány červeným písmem. Nepředpokládá se hromadná změna údajů, proto ponechány změny po řádcích. Tlačítko Načti data o knize z Google aktualizuje i obrázek obálky, pokud již není uložen obrázek ve formátu JPEG. Pokud byl uložen obrázek v jiném formátu je ponechán na disku.

Při aktualizaci obrázku nedojde ihned k aktualizaci na výpůjčním místě. Na výpůjčním místě bude aktualizován až po spuštění skriptu `mysql_dump_Backups/backup.sh` na výpůjčním místě. Skript se na výpůjčním místě spouští každý den. Nicméně pokud by byl požadavek, aby byly obrázky aktualizovány ihned, je třeba skript na výpůjčním místě spustit ručně.

## 9.4 Uživatelé

Na této stránce je přehled uživatelů umožňující jejich správu. Z této stránky je možno se dostat na stránku přidání nového uživatele a detail vybraného uživatele. Uživatelské jméno (login) musí být jedinečné. Uživatelé jsou řazeni abecedně podle uživatelského jména.

### Přidat uživatele

Kromě vyplnění textových polí s údaji o uživateli. Jsou zde možnosti nastavení chování účtu na výpůjčním místě. Jak tyto volby ovlivňují chování účtu je popsáno v části Nastavení profilu v návodu pro uživatele.

### Detail uživatele

Na této stránce je možné prohlédnout si zaznamenané údaje o uživateli a případně je měnit. Všechny tyto údaje si může uživatel měnit i sám, jak je popsáno v části Nastavení profilu v návodu pro uživatele. Je zde ale navíc možnost měnit jméno uživatele. Tato možnost je ponechána pouze pro hlavního knihovníka z důvodu dohledatelnosti, komu je, popř. byla kniha půjčena. Změna jména se projeví i u všech výpůjček včetně uzavřených. Uživatele nelze odebrat. Také kvůli tomu, aby bylo dohledatelné komu byla kniha půjčena.

## 9.5 Výpůjčky

Na této stránce je možnost zobrazení historie výpůjček. Výpůjčky jsou rozděleny podle roku, ve kterém byla výpůjčka zahájena. Je možné použít filtr na pouze

[Admin](#) [Knihy](#) [Uživatelé](#) [Výpůjčky](#) [Naháší chybu](#) [Nápověda](#) [Odhlásit](#)

## Knihovna katedry informatiky PŘF UP

### Stav

Počet uživatelů: 32  
 Počet aktivních uživatelů: 31  
 Počet titulů: 1402  
 Počet výpůjček: 12  
 Počet aktivních výpůjček: 12

### Dostupnost databází

Spojení s primární databází: ok  
 Spojení se sekundární databází: ok

### Porovnání tabulek výpůjčky v primární a sekundární databázi

Počet řádků v primární databázi: 12  
 Počet řádků v sekundární databázi: 12  
 Počet řádků je stejný.

Kontrola stavu u posledního sta výpůjček: Stav výpůjček v obou databázích jsou stejné.

### Exporty do CSV souborů

[Exportovat knihy do CSV](#) [Exportovat uživatele do CSV](#) [Exportovat výpůjčky do CSV](#) [Exportovat hlášené chyby do CSV](#) [Exportovat kategorie knih do CSV](#)

### Hlášené chyby

[Zobrazit](#) | [vyřešeno](#)

Id	Čas	Login	ISBN	Text		
1	2021-08-11 23:15:56	admin	443192082	chybné ISBN	<a href="#">Označit jako vyřešeno</a>	<a href="#">Vymazat</a>
2	2021-08-11 23:15:56	admin	444192082	chybné ISBN	<a href="#">Označit jako vyřešeno</a>	<a href="#">Vymazat</a>
3	2021-08-11 23:15:56	admin	1726641	chybné ISBN	<a href="#">Označit jako vyřešeno</a>	<a href="#">Vymazat</a>
4	2021-08-11 23:15:56	admin	4443029743	chybné ISBN	<a href="#">Označit jako vyřešeno</a>	<a href="#">Vymazat</a>
5	2021-08-11 23:15:56	admin	4433029743	chybné ISBN	<a href="#">Označit jako vyřešeno</a>	<a href="#">Vymazat</a>

Obrázek 5: Výchozí stránka po přihlášení hlavního knihovníka

[Admin](#) [Knihy](#) [Uživatelé](#) [Výpůjčky](#) [Naháší chybu](#) [Nápověda](#) [Odhlásit](#)

## Kategorie knih

### Přidat novou kategorii

Název kategorie:

Umístění:

[Přidat kategorii](#)

Název kategorie	Umístění	Počet knih v kategorii	Změnit název kategorie	Změnit umístění kategorie	Smazat kategorii *
Algebra	Sřeď C2	17	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Algebra uspořádání a svazy	Sřeď F2	14	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Algoritmické obchodování	Vpravo E7	3	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Algoritmy	Vlevo E3, E4, F3, F4	44	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Analýza	Sřeď C1, D1; Vlevo D7	10	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Aproximační algoritmy	Vlevo C3	13	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Rinformatika	Vpravo I-5	16	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Databáze (prakticky)	Vlevo G5	23	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Databáze (teorie)	Vlevo CS, D5, E5	73	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Datové struktury, randomized alg.	Vlevo D3	10	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
diskrétní matematika	Sřeď D2, E2	22	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Distribuované algoritmy	Vlevo G3, H3	31	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Doležání dat	Vlevo E7, F7, G7, H7	70	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Expertní systémy	Vlevo H6	6	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Faktorní analýza, PCA	Vlevo D7	19	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Filozofie	Sřeď B6	21	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Grafika	Vpravo D3	17	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
GUI	Vpravo F1	9	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	
Hardware	Vpravo I-2	4	<a href="#">Změnit název</a>	<a href="#">Změnit umístění</a>	

Obrázek 6: Rozhraní hlavního knihovníka – správa kategorií

aktivní výpůjčky v daném roce - tlačítko `Pouze aktivní`. Tlačítko `Všechny aktivní výpůjčky` zobrazí všechny aktivní výpůjčky nezávisle na době zahájení výpůjčky. Z této stránky se dá dostat zpět pomocí tlačítka `Všechny výpůjčky` nebo kliknutím na záložku `Výpůjčky`. Na této stránce a jejích modifikacích je možné ukončit výpůjčku knihy.

## 9.6 Nahlásit chybu

Chybu v tomto případě hlásí hlavní knihovník sám sobě. Funkce je ale ponechána, protože může být využita jako poznámka. Také je možné, že přístup k účtu hlavního knihovníka bude mít více lidí a daný problém umí vyřešit někdo jiný.

## 10 Vstupní data

Vstupní data byla doručena v souborech typu CVS. Tato data bylo nutno převést do databáze. Pro hromadnou konverzi nebyl sestavován žádný automatický nástroj, protože je potřeba pouze jednorázově při inicializaci. Navíc bylo i potřeba ruční doladění některých nesrovnalostí nebo netypických stavů. Převod byl proveden pomocí tabulkového a textového editoru. Data byla převáděna na součást SQL dotazů. Vstupní data i jejich hotový převod je možno nalézt na příloženém disku. Po jejich otestování byl vytvořen ještě inicializační soubor `knihovna.sql`, který je výchozí primární i sekundární databází pro systém. I tento soubor je možné nalézt na disku.

U těchto vstupních dat si zaslouží další komentář evidence knih. V knihovně je přibližně 1600 svazků, pokud jsou sloučeny kopie téže knihy jedná se o asi 1400 titulů.

Ze vstupních dat bylo patrné, že všechny knihy mají uveden název, kategorii a počet. Ne všechny knihy mají uvedeny údaje o autorovi, ISBN, signatuře, roku vydání a údaje ve sloupci `info`, ten ale chyběl jen u jedné. Sloupec `info` většinou udává informaci o vydavatelství. Podle těchto údajů byla nastavena omezení na údaje, které je nutné uvést při přidávání další knihy. Navíc je nutné uvést i ISBN. Omezení jsou realizována na úrovni databáze a skriptů obsluhujících formuláře. Údaj ISBN nebyl u všech knih, tento údaj, je však používán k rozlišení knihy skenerem a také jako primární klíč databázové tabulky knihy. Je třeba uvést, že ač je použito označení ISBN, tak ne vždy to skutečně ISBN musí být. Může to být i jiný kód, který je použit pro identifikaci knihy. U chybějících knih bylo ISBN, vygenerováno a to podle předpisu: jedenácti-místné číslo, které začíná 777. Čili i zde se nejedná o ISBN. Pokud nejsou ISBN na knihách přítomna nebo nejsou v podobě čárového kódu, je možné tyto kódy vytisknout a na knihy nalepit.

ISBN bylo nutné převést na tvar bez pomlček. Je to z více důvodů. Skener vysílá tvar bez pomlček. V databázi by mohli vznikat duplicity, protože tvary s pomlčkami a bez jsou rozdílné. Navíc by bylo i kompilovanější vyhledávání. Nejspíše by se tyto problémy daly vyřešit, ale pokud by byl požadavek pomlčky

zachovat, tak bych za nejvhodnější řešení považoval evidovat ještě jeden sloupec nazvaný např. `ISBN_original`.

Problém jsem spatřoval v možnosti výskytu znaku `X`. Databáze je naštěstí case insensitive. Proto velikost `X` v primárním klíči nehraje roli.

Vstupní data však bohužel ne vždy odpovídají skutečnosti. Při testování v knihovně bylo odhaleno, že v evidenci neodpovídají některá ISBN skutečnosti. Čili, že naskenovaný kód z knihy neodpovídá evidovanému kódu. Tento problém bude muset být řešen až za běhu knihovny. A i kvůli tomu byla do aplikace přidána možnost měnit i ISBN. Naštěstí se nestávalo, že by knihu nebylo možno nalézt v katalogu zadáním klíčových slov.

Ve vstupních datech byla i informace o tom, v jaké kategorii je kniha evidována. Kategorie mají rozsah daný mapou kategorií. Uvedení rozsahu v aplikaci by však mohlo vést k nejednoznačné interpretaci. Např. umístění `Vlevo C1-D1` znamená buňky `C1` a `D1` nebo buňky `C1`, `C2`, `C3`, `C4`, `C5`, `C6`, `C7`, `C8` a `D1`? Proto byly buňky umístění kategorie taxativně vyjmenovány.

## 11 Databáze

### 11.1 Zrcadlení databáze

Výhodou systému je to, že je možné půjčovat knihy i při ztrátě konektivity mezi výpůjčním místem a serverem. Stejně tak je stále možné prohlížet záznamy z webového rozhraní.

Pro zajištění takového chování je potřeba mít dvě stejné databáze. Obecně pro řešení takového problému je potřeba, aby byla jen jedna databáze k zápisu. Taková databáze se nazývá primární. A druhá, která je kopií primární databáze je sekundární databází. V případě zrcadlení není možné, aby byl umožněn zápis do dvou databází, protože by mohlo docházet ke kolizím. Řešení kolizí obecně nemusí být jednoduché.

Pro řešení této vlastnosti vyšlo nejlépe řešení takové, kdy primární databáze bude na výpůjčním místě a sekundární na serveru poskytujícím webové rozhraní.

Při návrhu řešení jsem bral do úvahy i to, že na výpůjčním místě je RPI, které používá jako úložiště paměťovou kartu, která má omezený počet zápisů. Nicméně, pokud je požadavek mít stejné databáze na obou místech, tak počet zápisů je stejný. Dále databáze je postavena tak, že zápisy obecně přibývají, než že by se měnily. Pokud by však paměťová karta selhávala v krátkých intervalech, bylo by lépe použít klasický, magnetický pevný disk nebo primární databázi přesunout jinam.

Řešení, kdy je primární databáze na výpůjčním místě, umožňuje, aby i při ztrátě konektivity bylo na výpůjčním místě možno zadat výpůjčku nebo vrácení. Samozřejmě při ztrátě konektivity nemohou být aktuální data na obou místech. Proto při takové situaci nebudou aktuální data o výpůjčkách na veřejném serveru, což není takový problém, jako když by nešli zadávat výpůjčky a vrácení. Při ztrátě konektivity veřejného serveru vůči primární databázi není možno z veřejného

serveru provádět nad databázemi dotazy, které databáze mění. Je však zachována možnost data prohlížet.

Při ztrátě konektivity dochází na výpůjčním místě k tvorbě fronty s databázovými dotazy, které proběhli nad primární databází, ale ještě nebylo možné je provést nad databází sekundární. Po obnovení spojení je fronta vyprázdněna.

V návrhu a implementaci se i podařilo vyřešit možnost kolizí. Je to dáno tím, že výpůjční místo mění jen obsah tabulky výpůjčky a tabulku pro hlášení chyb. Z Veřejného serveru je možno změnami zasahovat do všech tabulek. U tabulky log to příliš nevadí. Protože z výpůjčního místa mohou záznamy pouze přibývat. Došlo by tak pouze ke změně pořadí záznamů v tabulkách, což pro běh není podstatné. Problém byl u tabulky výpůjčky, kdy jedinečná identifikace byla ponechána na databázi a tak zůstávala hypotetická možnost kolize. Bylo by to v situaci, kdy dojde k obnovení spojení, a byl požadavek z veřejného serveru na zadání výpůjčky. Zadat výpůjčku na veřejném serveru může jen hlavní knihovník. Ten je po přihlášení informován o stavu databází. V takovém okamžiku by tedy byl informován o tom, že databáze nejsou synchronizované. O této skutečnosti je napsáno varování i v návodu. I když je pravděpodobnost takového stavu je malá a nastal by i přes varování, byl ještě návrh doplněn o vyloučení této možné kolize. A to tak, že čísla výpůjček jsou přidělovány přímo PHP skripty. Čísla jsou tvořena z data, času až po desetiny sekund a jako poslední číslo je použito číslo 0 pro vznik výpůjčky na výpůjčním místě nebo číslo 1 pro vznik výpůjčky na veřejném serveru. Takže i v případě, že by nastala výše popsaná situace, tak ke kolizi nedojde.

Řešením mohlo být i to, že by hlavnímu knihovníkovi v případě detekce bylo zakázáno výpůjčky zadávat. Tato možnost ale zůstala ponechána. Je to z toho důvodu, že kdyby se stalo, že dojde k poškození zrcadlení, může i nadále hlavní knihovník zadávat výpůjčky. Systém by v případě poškozeného zrcadlení korektně fungoval pouze s primární databází, což je dostatečné. Na čtení sekundární se přepíná jen v případě, že dojde ke ztrátě spojení s primární databází. To se nejspíše často stávat nebude.

V tomto případě, kdy se kolize zdají být vyloučeny, by teoreticky mohlo být povoleno i měnit záznamy sekundárnímu serveru v případě výpadku primární databáze. Nicméně by se potom mohlo stát, že by nebyl výpadek konektivity řešen delší dobu, co už by mohl být problém. Pak by se mohlo stávat, že například by bylo změněno heslo na veřejném serveru, ale na výpůjčním místě ještě ne. Nezobrazily by se knihy připravené v seznamu a podobně.

Pro samotné zrcadlení databází je možno přímo využít nastavení poskytovaných přímo databázovým serverem. Od takového řešení mě však odradili diskuse na webových fórech, na kterých se psalo, že dochází k selhání takového zrcadlení. A i sám bych věřil, že se tak stát může. Například při rozdílných verzích databázových systémů při aktualizacích. Kvůli riziku takového problému jsem tedy zvolil vlastní implementaci zrcadlení. Mimo jiné v něm spatřuji i výhodu toho, že při případném narušení zrcadlení, stačí na sekundární server nahrát aktuální verzi primární databáze, popřípadě jinak dosáhnout stejných databází na

obou serverech. Pokud jsou databáze stejné, není už potřeba nic znovu nastavovat.

Toto zrcadlení je zajišťováno skriptem `s_db_connect.php`. Skript je stejný pro veřejný server i výpůjční místo. Je možné i databáze zaměnit a mít primární databázi na veřejném serveru. Při pokusech s takto nastavenou architekturou jsem žádnou chybu nenašel. Bylo ale vhodné implementovat automatické vyprázdnění fronty dotazů pro sekundární databázový server na veřejném serveru. V této verzi jsou pokusy o vyprázdnění fronty prováděny při uživatelské akci, u které se poprvé načítají data z databáze v daném sezení.

Ve skriptu `s_db_connect.php` jsou implementovány čtyři funkce. Dvě funkce po dotazy, které databáze nemění a dvě, které databáze mění. V každé z těchto dvou skupin je funkce, u které není nebo je potřeba ošetřit parametry, kvůli ochraně před XSS a SQL injection útoky. Funkcím jsou předávány požadavky od ostatních skriptů, které jsou tímto od této problematiky odstíněny voláním jedné z těchto čtyř funkcí.

Funkce, které databáze nemění se nejprve dotazují na primárním serveru. Pokud je spojení v pořádku, je vrácena odpověď. Pokud není spojení na primární server, tak již není primární databáze pro dané sezení zkoušena, protože neúspěšné pokusy o spojení s databází trvají. Je informován uživatel, není možno provádět změny a je přepnuto na sekundární databázi. Pokud by nebylo spojení ani na druhou databázi, je uživatel odhlášen, přesměrován na výchozí stránku a informován o problému.

Funkce které databázi modifikují vyžadují spojení na primární databázi. Pokud spojení není, nebo během sezení nebylo, ani se nepokoušejí dotazy vykonávat. Uživatel je o tomto informován. Při spojení s primární databází je dotaz proveden a vzápětí je proveden nad sekundární databází. Pokud by nebylo spojení na sekundární databázi je zakázáno další zkoušení sekundární databáze během sezení. Dotaz a stejně tak i případné následující modifikující dotazy během sezení jsou ukládány do fronty. O ztrátě konektivity na sekundární server není běžný uživatel informován. Pokus o vyprázdnění fronty je proveden při každém prvním pokusu o spojení s databází v daném sezení. Na výpůjčním místě takový pokus způsobuje načtení úvodní stránky. Úvodní stránka zároveň i odhlašuje uživatele, takže její načtení je novým sezením. Vzhledem k tomu, že je na výpůjčním místě nastaveno odhlášení, resp. přesměrování na úvodní stránku po deseti minutách od načtení stránky a je ponecháno i pro úvodní stránku, dochází k takovému pokusu co deset minut.

Podle návrhu se počítá s tím, že primární databáze bude na výpůjčním místě. Je to kvůli tomu, aby šlo i půjčovat v případě ztráty konektivity. Implementace však poskytuje možnost záměny umístění primární a sekundární databáze. Po případě mít i jinou architekturu. Včetně umístění databází mimo výpůjční místo i veřejný server. Například při požadavku mít více výpůjčních míst nebo požadavku mít databázový server mimo výpůjční místo, kvůli častým selháním paměťové karty. Definice hodnot potřebných k přístupu k databázím je ve skriptech `db_connect.php`. Pokud by však nebyla primární databáze na výpůjčním

místě, nebyla by možnost zadat výpůjčku v případě ztráty konektivity. Teoreticky by bylo možno i vcelku malými změnami toto umožnit, avšak na výpůjčním místě by musela být alespoň sekundární databáze.

## 11.2 Obsah databáze

Byť bude systém nasazen, tak stále se jedná o studentskou práci. Proto je pravděpodobné, že budoucí zásahy bude dělat někdo jiný. Proto byla snaha vytvořit databázi co nejjednodušší, ideálně čitelnou člověkem. V databázi jsou názvy tabulek a sloupců v českém jazyce bez diakritiky a mezer. Toto je rozdíl oproti PHP a BASH skriptům, které jsou napsány v jazyce anglickém. Český jazyk v databázi je použit z důvodu zhoršení čitelnosti a odhadu názvů pro případné útočníky, byť je v systému pamatováno na útoky typu XSS a SQL injection.

V této části je výčet tabulek, popis jejich funkce a popis sloupců tabulek. Na konci popisu sloupce je i popsáno nastavené omezení. Samotné SQL příkazy pro vznik databáze jsou k dispozici na příloženém disku.

Pokud je někde uvedeno, že se jedná o cizí klíč, tak je nastaven na chování, kdy při změně v tabulce původu dojde ke změně v tabulce, ze které je odkazováno. Smazání v tabulce původu je zakázáno.

### **kategorie**

Tabulka obsahuje názvy kategorií a jejich umístění. Sloupce:

#### **nazev\_kategorie**

Primární klíč, maximálně sto znaků.

#### **umisteni**

Maximálně sto znaků.

### **knihy**

Tabulka obsahuje informace o knihách. U knih je uvedena kategorie přímo. Není odkaz do tabulky kategorie. Tím dochází k redundanci informace, ale zase je tabulka lépe čitelná pro člověka. Na druhou stranu, pokud by byla provedena neopatrná změna kategorie knihy přímo v databázi, nebylo by knize již nadále přidělováno správné umístění kategorie. Nepočítá se ale s tím, že by byly prováděny změny databáze přímo. Například by to ale mohlo být potřeba při hromadném rozdělení kategorie a u takové operace by mohl být uživatel pozorný. Sloupce:

#### **isbn**

Primární klíč. Jednoznačná identifikace knihy. ISBN je uvedeno bez pomlček. Nemusí jít nutně ISBN, ale nejčastěji tomu tak je. Ne všechny knihy mají ISBN. A může být využit i jiný kód, který se například může lépe skenovat. Pokud knihy

ISBN nebo čárový kód nemají nebo je tento údaj nepoužitelný může být přidělen vlastní číselný kód a tento i ideálně nalepen jako čárový na knihu. Je vhodné se pro takový kód držet metodiky, aby bylo poznat, že je vlastní. Při inicializace databáze byla generována číselná řada kódů, které začínají 777 a celkově jsou jedenáctimístné. Např. 7770000005. Jak popsáno, je to metodika a databáze ani systém nemá nastaveno omezení. Nicméně bylo by vhodné se nějaké metodiky držet. Je důležité neuvádět pomlčky. Při použití pomlček by mohli být zaváděny do databáze duplicity a aplikace by nerozpoznalo kód po naskenování. Maximálně 30 znaků.

**nazev**

Název knihy. Maximálně tisíc znaků.

**autor**

Autor knihy. Maximálně tisíc znaků.

**info**

Doplňkové informace o knize. Nejčastěji vydavatel. Maximálně tisíc znaků.

**rok**

Rok vydání knihy. Maximálně 4 znaky. Zvoleny znaky, protože u čísel je neuvedení považováno za nulu.

**signatura**

Signatura popř. signatury u více svazků. Maximálně tisíc znaků.

**kategorie**

Kategorie. Měla by se shodovat s některou kategorií z tabulky `kategorie`. Systém neumožňuje, aby se toto neshodovalo, ale při provádění dotazů přímo nad databází by se toto stát mohlo. Následkem by bylo, že nebyla možná spolupráce s tabulkou `kategorie`. Tzn. neprojevila by se změna kategorie při přejmenování kategorie jako celku a nebyla by správně přiřazeno umístění knihy dle kategorie. Maximálně sto znaků.

**umisteni**

Umístění knihy. Obvykle zde je NULL. Tzn., že knize je programem přiřazeno umístění dle kategorie. Pokud je zde nějaká hodnota, je v systému zobrazena tato hodnota. Maximálně sto znaků.

**svazku\_celkem**

Počet svazků celkem. Hodnota je důležitá k tomu, aby bylo možno zjistit, zda je nějaký svazek volný. Číslo (MySQL datový typ SMALLINT).



**svazku\_volnych**

Počet volných svazků. Derivovaná hodnota. Hodnota je spočítána z celkového počtu svazků a počtu aktivních výpůjček knihy. Resp. při půjčení, vrácení knihy dochází ke změně tohoto čísla. Pokud by všechny dotazy neproběhli správně, může zde být špatná hodnota. Toto se dá napravit z detailu knihy v rozhraní pro hlavního knihovníka. I když je to hodnota dopočítaná, bylo zvolena její evidence i u knihy z důvodu možnosti zobrazení přímo v katalogu. Pokud by byl dotaz přes více tabulek při výpisu katalogu, trvalo by to déle. Hodnotou je číslo (MySQL datový typ SMALLINT).

**obrazek**

Pokud je uložen obrázek obálky, je zde přípona tohoto obrázku. Pokud obrázek uložen není, je zde NULL. Maximálně 10 znaků.

**log**

Tabulka k hlášení chyb uživateli. Záznamy do této tabulky může vytvořit přihlášený uživatel, nepřihlášený uživatel, hlavní knihovník i systém. Sloupce:

**id\_logu**

Automaticky přidělené číslo logu.

**nick**

Přihlašovací jméno, login uživatele, který log vytvořil. V aplikaci je možno tuto hodnotu editovat, může zde být tedy hodnota, která neodpovídá skutečnému uživateli. Maximálně 100 znaků.

**isbn**

V aplikaci editovatelná položka, nemusí odpovídat ISBN zavedenému v tabulce knihy. A je to i docela možné, protože jednou z možných chyb je i to, že ISBN skutečné a v databázi neodpovídají. Maximálně 30 znaků.

**text\_logu**

Volně napsaný text logu, hlášení chyby. Text.

**cas**

Čas, kdy byl log vytvořen. MySQL datový typ DATETIME.

**vyřešeno**

Poznámka pro hlavního knihovníka, jestli už hlášení vyřešil. Takový log se pak již ve výchozím zobrazení v aplikaci nezobrazuje. MySQL datový typ BOOLEAN.

## **priprava**

Tabulka k vytvoření předpřipravených seznamů pro uživatele. V aplikaci jsou řádky tabulky přiřazeny uživateli. Navrženo takto, aby se nevytvářela tabulka pro každého uživatele. Je to doplňková funkce a předpokládaný počet uživatelů 40. I kdyby bylo 100 uživatelů a každý měl 100 záznamů, tak je to stále „jen“ deset tisíc řádků.

## **id\_pripravy**

Databázi automaticky přidělené číslo.

## **isbn**

ISBN odpovídající knihy. Cizí klíč z tabulky knihy. Maximálně 30 znaků.

## **id\_vypujcitele**

Databázi přidělené číslo uživatele. Cizí klíč k tabulce uživatele. Číslo.

## **uzivatele**

Tabulka s údaji o uživatelných. Sloupce:

## **id\_uzivatele**

Databázi přidělené číslo. Primární klíč.

## **jmeno**

Jméno a příjmení. Jméno, příjmení popř. další jména, příjmení a tituly nejsou rozděleny. Pořadí je důležité, protože podle příjmení, resp. řetězce za první mezerou dochází k abecednímu řazení na přihlašovacích obrazovkách na výpůjčním místě. Maximálně sto znaků.

## **nick**

Přihlašovací jméno uživatele (login). V tabulce nesmí být více stejných přihlašovacích jmen. Maximálně sto znaků.

## **heslo**

Heslo v dvojité hašované podobě. První je proveden sha-512 haš u klienta Javascriptem a odeslán ve formuláři. Po přijetí na server je k tomuto haši přidán řetězec a je znovu proveden sha-512. Hodnotu přidaného řetězce je možno zjistit ze zdrojových kódů. Maximálně tisíc znaků.

## **pozadovat\_heslo**

Hodnota určuje, zda-li je požadováno heslo při přihlašování na výpůjčním místě. MySQL datový typ BOOLEAN.

**zrychlene\_pujcovani**

Hodnota určuje, zda-li je na výpůjčním místě uživatel rovnou přesměrován do záložky Zrychleně. MySQL datový typ BOOLEAN.

**aktivni**

Hodnota určuje, zda-li je možno se na výpůjčním místě přihlásit. MySQL datový typ BOOLEAN.

**casty**

Hodnota určuje, jestli je uživatel na první obrazovce na výpůjčním místě. MySQL datový typ BOOLEAN.

**vypujcky**

Tabulka eviduje výpůjčky uživatelů. Sloupce:

**id\_vypujcky**

PHP skripty přidělené číslo. Číslo je řetězec určený datem a časem. Poslední číslice určuje odkud byla výpůjčka zadána. 0 pro výpůjční místo. 1 pro veřejný server. Primární klíč.

**isbn**

ISBN odpovídající tabulce výpůjčky. Cizí klíč. Maximálně 30 znaků.

**id\_vypujcitele**

Odpovídá záznamu v tabulce uživatelů, sloupec `id_uzivatele`. Cizí klíč, číslo.

**pujceno\_od**

Začátek výpůjčky. MySQL datový typ DATETIME.

**pujceno\_do**

Konec výpůjčky. Pokud je NULL, kniha nebyla vrácena, výpůjčka je aktivní. MySQL datový typ DATETIME.

**vypucku\_zadal**

Odpovídající záznam v tabulce uživatele, sloupec `id_uzivatele`. Uvedeno, jen pokud je výpůjčka zadána z účtu `admin`, jinak výpůjčku zadal sám uživatel a zůstává NULL. Cizí klíč, číslo.

**vraceni\_zadal**

Odpovídající záznam v tabulce uživatele, sloupec `id_uzivatele`. Uvedeno, jen pokud je vrácení zadáno z účtu `admin`, jinak vrácení zadal sám uživatel a zůstává NULL. Cizí klíč, číslo.

## 11.3 Řešení problému rozdílných záznamů v databázích

### Detekce

I když je systém navržen tak, aby odolal i ztrátě konektivity mezi počítači, je možné, že dojde k rozdílu mezi databázemi a tím selhání celého zrcadlení.

Z rozhraní pro administrátora lze rozdílné databáze detekovat na úvodní stránce administrátorského rozhraní.

V sekci porovnání tabulek výpůjčky v primární a sekundární databázi. Pokud je zde rozdíl, tak databáze nejsou shodné a je vhodné postupovat podle návodu níže.

V sekci hlášené chyby. Zde se může objevit hlášení Rozdíl mezi počtem nevrácených v tabulce vypujcky a svazku\_volnych v tabulce knihy. Tento rozdíl se dá napravit z administrátorského rozhraní v detailu knihy. A to kliknutím na tlačítko Opravit v sekci Změna údajů řádek Počet volných exemplářů dle tabulky knihy.

Dále je na této stránce sekce Záznamy ze souboru s logy. Soubory s logy se tvoří pokud je detekován rozdílný výsledek modifikujících dotazů. V tomto případě mohou, ale nemusí být databáze rozdílné. Je vhodné databáze porovnat a potom teprve se snažit o nápravu. Avšak porovnání může být stejně časově náročné, jako náprava, takže je na zvážení, co je v danou chvíli výhodnější. Po nápravě nebo zjištění, že je vše v pořádku, je potřeba soubory ručně smažat. Cesta k souborům s logy je `/var/www/html/logs/logs.txt`. Soubory mohou být jak na RPI, tak na veřejném serveru, což je ale patrné z administrátorského rozhraní.

### Řešení

Nejpravděpodobněji k tomuto problému došlo v tabulce výpůjčky. Pokud je problém v sekundární databázi a tato situace je pravděpodobnější, může systém pracovat dál na primární databázi. Nicméně pokud by selhala i primární databáze, může dojít ke ztrátě dat. Ztráta dat by však měla být maximálně za posledních 24 hodin, protože systém každý den vytváří zálohu primární databáze.

Řešení tohoto problému není implementováno v administrátorském rozhraní a je k němu vhodná znalost MySQL databáze.

V době, kdy se bude problém řešit je potřeba systém zastavit, znemožnit další změny. Vybrat databázi, která je v pořádku. Pravděpodobně to bude primární databáze na RPI. A tuto databázi znovu nahrát na databázový server, u kterého se předpokládá, že má chybné záznamy. Nejsnazší způsob se jeví pomocí příkazu `mysqldump`.

## 12 Popis skriptů

Při tvorbě skriptů bylo snahou, aby skripty byly snadno čitelné pro případné budoucí změny. Skripty jsou psány v anglickém jazyce. Komentáře v jazyce čes-

kém. Obecně jsou skripty rozděleny na první část, která pracuje s databází. A druhou část, prezentační, která zajišťuje zobrazení výsledků. Obecně je HTML kód pro výpis připraven v první části, který je v prezentační části vypsán. Vznikl sice tzv. špagetový kód, na druhou stranu je vše na jednom místě.

Skripty používají sdílené proměnné `$_SESSIONS`. Požadavky si nejčastěji předávají typem `POST`, občas je ale využít i požadavek `GET`. Ten je využíván v situacích, kdy předání požadavku od skriptu není vyvoláno uživatelem, ale skriptem samotným.

V nastavení systému se počítá, že skripty jsou ve složce `/var/www/html`. A to jak na výpůjčním místě, tak na veřejném serveru. Dále popsané cesty k dalším složkám a skriptům jsou relativní k této výchozí složce.

## 12.1 Složky a skripty pro výpůjční místo (RPI)

### Struktura adresáře `var/www/html`

- book\_details.php
- borrow\_book.php
- catalog.php
- cover\_imgs
- extra
  - vlevo.jpeg
  - vpravo.jpeg
  - stred.jpeg
- help.php
- history.php
- how\_many\_lines\_logs\_has.php
- include
  - announcement.php
  - db\_connect.php
  - error\_reporting.php
  - logout\_after\_time.php
  - menu.php
  - scanner.js
  - styl.css
- index.php
- loggedin.php
- loggedin\_quick\_borrow-return.php
- login.php
- logs
  - logs.txt
- mysql\_dump\_Backups
  - crontab
  - daily
  - weekly

monthly  
backup.sh  
other\_active\_users.php  
report\_error.php  
return\_book.php  
secondary\_db\_pending\_query  
waitingQueries

### **book\_details.php**

Skript má za úkol zobrazit informace evidované o knize. Pokud má kniha uložen i obrázek obálky, je také zobrazen.

### **borrow\_book.php**

Skript má za úkol půjčit vybranou nebo vybrané knihy. Skript ověřuje, zda-li je výpůjčku možno zadat a po ověření provede příslušné změny v databázi. Skript mění záznam nebo záznamy v databázové tabulce výpůjčky a to tak, že vloží řádek s identifikací uživatele, knihy a datem výpůjčky. Tím je výpůjčka považována za zahájenou. Dále upravuje sloupec s volnými exempláři v tabulce knihy. Nakonec je ještě provedena kontrola, jestli odpovídá počet volných knih celkovému počtu svazků poníženému o počet aktivních výpůjček. Pokud je zde chyba, je automaticky poslána zpráva hlavnímu knihovníkovi. Skript po provedení přeměruje zpět na skript, který vypíše půjčené knihy. Podle aktuální situace je to skript `loggedin.php` nebo `loggedin_quick_borrow-return.php`

### **catalog.php**

Skript má za úkol zobrazit katalog knih. Knih je evidováno asi 1400, proto jsou vypsány na jedné stránce. I při testování na počtu 3000 je načtení stále dostatečně rychlé. Skript pro úsporu dotazů u každé knihy na umístění v kategorii provede pouze jeden dotaz na získání tabulky kategorie, kterou si uloží při běhu do asociativního pole. Je zde umožněno knihy řadit podle všech sloupců kromě sloupce zobrazujícího počty svazků a umístění. Skript umožňuje v klientské části vyhledávat v katalogu pomocí Javascriptu tvorbou CSS selektorů. Z tohoto skriptu je možno přejít na skript `book_details.php` pro vybranou knihu.

### **cover\_imgs**

Složka obsahující obrázky obálek knih. Složka je aktualizována obsahem z veřejného serveru jednou denně pomocí skriptu `mysql_dump_Backups/backup.sh`.

## **extra**

Složka obsahuje obrázky topografie knihovny. Obrázky jsou využity ve skriptu `help.php`.

## **help.php**

Skript zobrazí uživatelskou nápovědu. Text nápovědy je obsažen přímo v textu skriptu. Na stránce vytvořené tímto skriptem se zobrazují i obrázky umístění kategorií. Obrázky jsou uloženy ve složce `extra`. Názvy souborů obrázků: `stred.jpeg`, `vlevo.jpeg`, `vpravo.jpeg`.

## **history.php**

Skript zobrazuje historii výpůjček pro daného uživatele. Umožňuje odeslat požadavek k vrácení jednotlivých knih i všech vypůjčených knih.

## **how\_many\_lines\_logs\_has.php**

Skript není využit v rozhraní výpůjčního místa. Využívá ho veřejný server pro rozhraní hlavního knihovníka. Slouží pro zjištění kolik řádků je v souboru `logs/logs.txt`. Vypisuje i obsah souboru. Na veřejném serveru však výpis obsahu není využit.

## **include**

Složka obsahuje vkládané skripty. V této složce jsou některé soubory shodné se soubory na veřejném serveru. Jsou to soubory:

- `announcement.php`
- `db_connect.php`
- `error_reporting.php`
- `scanner.js`

## **announcement.php**

Skript zobrazuje hlášení o provedených uživatelských akcích. Zobrazuje světlý text na tmavém pozadí při dolním okraji obrazovky. Zobrazuje tlačítko, které oznámení znovu zobrazí v místě, kam je skript vložen. Toto znovu zobrazené hlášení již působí jako součást stránky. Skript musí být vložen v rámci HTML elementu `body`. Obvykle je vkládán na konec elementu `body`. Po zobrazení tlačítko zobrazené hlášení může opět skrýt.

### **db\_connect.php**

Skript zajišťuje spojení s databází. Je zde nastavení spojení na databáze. Přebírá dotazy a provádí je dle potřeby nad primární, sekundární nebo oběma databázemi. Tímto skriptem je řešeno zrcadlení databází. Do skriptu jsou i nesystematicky vloženy funkce pro hašování hesla a přesměrovávání skriptů. Je to soubor, u kterého je velmi vhodné omezit přístup, protože obsahuje hesla a předpis hašování. Bližší popis architektury zrcadlení databází je k nalezení v textu diplomové práce.

### **error\_reporting.php**

Skript umožňuje hlášení chyb při vývoji. V produkčním kódu bude hlášení chyb vypnuto. Zde je ale případně možno pro ladění hlášení chyb zapnout.

### **logout\_after\_time.php**

Skript způsobuje automatické odhlášení po deseti minutách. Zde případně možno čas k odhlášení přenastavit.

### **menu.php**

Skript pro prezentační část aplikace. Zajišťuje vypsání správných záložek, položek menu pro různé kategorie uživatelů. V tomto případě pro nepřihlášené a přihlášené uživatele. Skript také obsahuje zákaz vyvolání kontextového menu. To je menu, které je vyvoláno po stisknutí druhého tlačítka myši a lze je vyvolat i gestem na dotykové obrazovce. Protože je skript vložen do všech skriptů, které mají za úkol vygenerovat stránku a je v rámci HTML elementu body tak je do skriptu je vložen ještě skript `logout_after_time.php`.

### **scanner.js**

Skript pro ovládání a napojení skeneru do aplikace. Rozeznává skenování od psaní na klávesnici. Po rozeznání skenování volá funkci `isbnScanned(isbn)`. Tuto funkci už obsahují skripty, do kterých je tento skript vložen. Bližší popis návrhu je k nalezení v textu diplomové práce v části `Scanner` čárových kódů.

### **styl.css**

Definice CSS stylů

### **index.php**

Výchozí skript. Zobrazuje tabulku uživatelů. Zobrazují se zde uživatelé, kteří mají nastavenou volbu `Aktivní` a `Častý`. Skript načte uživatele z databáze. Seřadí je podle příjmení. Za příjmení se považuje řetězec za první mezerou. Poté uživatele umísťuje do sloupců. Skript slouží i odhlášení uživatele. Po dotyku na jméno dochází k odeslání požadavku skriptu `login.php`.



## **logged\_in.php**

Výchozí skript po přihlášení uživatele, pokud nemá nastavenou volbu Zrychlené půjčování. Zobrazuje aktuální výpůjčky, a je zde možnost tvorby seznamu knih k vypůjčení. Pokud je skript načten je sem uživatel vrácen po zbytek sezení po zadání nebo ukončení výpůjčky.

## **logged\_in\_quick\_borrow-return.php**

Výchozí skript pro uživatele, pokud má nastavenou volbu Zrychlené půjčování. Skript po naskenování knihy rozhodne, jestli knihu půjčit nebo vrátit. Knihu vrátí, v případě, že ji má uživatel půjčenou. Pokud ji půjčenou nemá, tak se pokusí zadat výpůjčku. Pokud by uživatel chtěl půjčit více než jednu kopii dané knihy musí to udělat přes tvorbu seznamu. Pokud je skript načten je sem uživatel vrácen po zbytek sezení po zadání nebo ukončení výpůjčky.

## **login.php**

Skript přebírá požadavek na přihlášení od skriptu `index.php` nebo `other_active_users.php`. V případě, že uživatel nemá volbu Požadovat heslo rovnou uživatele přihlásí a z tohoto skriptu se nezobrazí nic. Pokud uživatel má aktivní volbu Požadovat heslo, je vyzván k zadání hesla. Skript hašuje heslo v klientské i serverové části. Serverová část využívá funkce `hashPassword` ze skriptu `include/db_connect.php`. Po přihlášení je uživatel přesměrován na skript `logged_in.php` nebo `logged_in_quick_borrow-return.php` podle volby Zrychlené půjčování.

## **logs**

Složka k uložení logů.

## **logs.txt**

Pokud soubor není přítomen, je vše v pořádku. Pokud existuje a obsahuje nějaké záznamy, je vhodné si je přečíst a popř. na ně reagovat. Logy by se zde měli tvořit pokud není dostupná primární databáze. Záznamy se také tvoří pokud výsledek dotazu nad primární a sekundární databází je rozdílný. V takovém případě je potřeba postupovat podle části Řešení problému rozdílných záznamů v databázích.

## **mysql\_dump\_Backups**

Složka obsahuje skript k vytváření záloh primární databáze a dále obsahuje i zálohy primární databáze samotné. Zálohy jsou ve složkách `daily`, `weekly`, `monthly`. Vzhledem k velikosti databáze při inicializaci cca 250 kilobajtů, je možné uchovávat záloh více.

### **crontab**

Soubor k nastavení nástroje `cron`. Je zde možno definovat, v který čas se provádí denní zálohy.

### **daily**

Složka pro denní zálohy.

### **weekly**

Složka pro týdenní zálohy.

### **monthly**

Složka pro měsíční zálohy.

### **backup.sh**

Skript pro vyvážení záloh a synchronizace složky s obrázky obálek knih. Skript se přes SSHFS připojí část souborového systému veřejného serveru. Vytvoří na serveru i v této složce denní zálohu databáze. Pokud je daný den v týdnu vytvoří týdenní zálohu. Pokud je daný den v měsíci vytvoří měsíční zálohu. Zálohy se tvoří do výše popsaných složek, stejné složky jsou i na serveru. Den v týdnu a den v měsíci je možno nastavit na začátku souboru. Ve výchozím nastavení je dnem pro týdenní zálohu pátek. Den pro měsíční zálohu desátý den v měsíci. Po vytvoření záloh skript odstraní zálohy staré. Denní zálohy ponechává po dobu 14 dní. Týdenní zálohy po dobu 12 týdnů. Měsíční zálohy nemaže. Skript nastaví přístupová práva na zálohy 600, takže je může číst jen uživatel, který skript spustil. Je to správcovský účet, takže by nepovolání neměli vidět zálohy. Zálohy totiž mimo jiné obsahují haše hesel a další možné citlivé informace.

Potom skript synchronizuje pomocí nástroje `rsync` obrázky obálek z veřejného serveru. Obrázky obálek, tak opět vlastní správce, což je vyhovující, protože server má přístup ke čtení.

### **other\_active\_users.php**

Skript zobrazuje tabulku uživatelů. Zobrazují se zde uživatelé, kteří mají nastavenou volbu `Aktivní` a nemají aktivní volbu `Častý`. Po dotyku na jméno dochází k přesměrování na skript `login.php`. Skript načte uživatele z databáze a seřadí je podle příjmení. Za příjmení se považuje řetězec za první mezerou. Poté uživatele umísťuje do sloupců.

### **report\_error.php**

Skript slouží k nahlášení chyby, resp. poslání vzkazu hlavnímu knihovníkovi.

## **return\_book.php**

Skript má za úkol vrátit danou knihu půjčenou daným uživatelem. Skript mění záznam v databázové tabulce výpůjčky a to tak, že přidá datum vrácení na řádku dané výpůjčky. Tím je výpůjčka považována za ukončenou. Dále upravuje sloupec s volnými exempláři v tabulce knihy. Nakonec je ještě provedena kontrola, jestli odpovídá počet volných knih celkovému počtu svazků poníženému o počet aktivních výpůjček. Pokud je zde chyba, je automaticky poslána zpráva hlavnímu knihovníkovi. Skript po provedení přesměruje na skript `logged_in.php` nebo `logged_in_quick_borrow-return.php` podle aktuálního nastavení.

## **secondary\_db\_pending\_query**

V této složce se vytváří soubor `waitingQueries`, pokud není přítomen, je vše v pořádku.

**waitingQueries** Soubor s frontou dotazů, které mění záznamy v databázi. Tyto dotazy již proběhly v primární databázi, ale ještě neproběhly v sekundární. Soubor vytváří a maže skript `include/db_connect.php` pokud není spojení na sekundární databázi a došlo ke změně primární databáze. Ve výchozím nastavení k tomuto dojde při zadání výpůjčky a vrácení knihy. Při každém odhlášení uživatele dojde k pokusu o spojení se sekundární databází a vykonání požadovaných změn. K automatickému odhlášení uživatele podle výchozího nastavení dochází co 10 minut a to i v případě, že není žádný přihlášen.

## **12.2 Složky a skripty pro veřejný server**

### **Struktura adresáře `var/www/html`**

```
cover_imgs
extra
  isbnDone
  ISBNs
  s_covers_inicial.php
  update_db_covers_records.php
include
  announcement.php
  db_connect.php
  error_reporting.php
  scanner.js
  s_menu.php
  s_styl.css
index.php
logs
  logs.txt
```

s\_add\_new\_book.php  
s\_add\_new\_user.php  
s\_admin.php  
s\_book\_details.php  
s\_borrow\_book\_admin.php  
s\_borrows.php  
s\_catalog.php  
secondary\_db\_pending\_query  
    waitingQueries  
s\_get\_details\_from\_google.php  
s\_help\_admin.php  
s\_help.php  
s\_choose\_name.php  
s\_loggedin.php  
s\_logout.php  
s\_manage\_groups.php  
s\_prepare\_book\_list.php  
s\_report\_error.php  
s\_return\_book\_admin.php  
s\_settings.php

### **cover\_imgs**

Složka obsahující obrázky obálek knih. Názvy souborů jsou důležité. Jsou ve formátu <ISBN>.<přípona>. ISBN musí odpovídat ISBN uvedenému v databázi. Povolené formáty obrázků jsou: JPEG, PNG, GIF, BMP. Obrázky jsou aplikací vkládány pouze z administrátorského rozhraní, proto je primární složka zde. Obdobná složka na RPI se synchronizuje s touto jednou denně. Obrázky je možno vložit i nakopírováním a databázi informovat spuštěním skriptu `extra/s_covers_inicial.php`.

### **extra**

Obsah složky není pro běh aplikace nutný. Jsou zde skripty a soubory pro inicializaci obrázků obálek, které jsou stahovány pomocí Google.

### **isbnDone**

Soubor nemusí být přítomen, je vytvářen skriptem `s_covers_inicial.php` při stahování obrázků obálek knih. Vypsány jsou ISBN, která již byla zkoušena. Jestli úspěšně nebo neúspěšně nemá na zapsání ISBN do souboru vliv.

### **ISBNs**

Soubor nemusí být přítomný. Tento soubor je naplněn ISBN, která čte skript

`s_covers_inicial.php` a zkouší stáhnout obrázek. Formát: ISBN oddělena odřádkováním.

### **s\_covers\_inicial.php**

Samotný skript pro stahování obrázků. Očekává soubor ISBNs, ve výše popsaném formátu.

### **update\_db\_covers\_records.php**

Skript projde složku `cover_imgs` a podle obsahu aktualizuje záznamy v databázi. Podle záznamů v databázi se pokouší aplikace obrázky najít a zobrazit. Tento skript se může použít i při prostém nakopírování obrázků do složky `cover_imgs`. U takto nakopírovaných obrázků je však třeba dodržet tvar názvu, který je `<ISBN>.<přípona>`.

### **include**

Složka obsahuje vkládané skripty. V této složce jsou některé soubory shodné se soubory serverové části aplikace. Jsou to soubory:

- `announcement.php`
- `db_connect.php`
- `error_reporting.php`
- `scanner.js`

### **announcement.php**

Skript zobrazuje hlášení o provedených uživatelských akcích. Zobrazuje světlý text na tmavém pozadí při dolním okraji obrazovky. Zobrazuje tlačítko, které oznámení znovu zobrazí v místě, kam je skript vložen. Toto znovu zobrazené hlášení již působí jako součást stránky. Skript musí být vložen v rámci HTML elementu `body`. Obvykle je vkládán na konec elementu `body`. Po zobrazení tlačítko zobrané hlášení může opět skrýt.

### **db\_connect.php**

Skript zajišťuje spojení s databází. Je zde nastavení spojení na databáze. Přebírá dotazy a provádí je dle potřeby nad primární, sekundární nebo oběma databázemi. Tímto skriptem je řešeno zrcadlení databází. Do skriptu jsou i nesystematicky vloženy funkce pro hašování hesla a přesměrování skriptů. Je to soubor, u kterého je velmi vhodné omezit přístup, protože obsahuje hesla a předpis hašování. Bližší popis architektury zrcadlení databází je k nalezení v textu diplomové práce.

### **error\_reporting.php**

Skript umožňuje hlášení chyb při vývoji. V produkčním kódu bude hlášení chyb vypnuto. Zde je ale případně možno pro ladění hlášení chyb zapnout.

### **scanner.js**

Skript pro ovládání a napojení skeneru do aplikace. Rozeznává skenování od psaní na klávesnici. Po rozeznání skenování volá funkci `isbnScanned(isbn)`. Tuto funkci už obsahují skripty, do kterých je tento skript vložen. Bližší popis návrhu je k nalezení v textu diplomové práce v části `Scanner` čárových kódů.

### **s\_menu.php**

Skript pro prezentační část aplikace. Zajišťuje vypsání správných záložek, položek menu pro různé kategorie uživatelů. V tomto případě pro nepřihlášené, přihlášené běžné uživatele a hlavního knihovníka

### **s\_styl.css**

Definice CSS stylů

### **index.php**

Výchozí skript umožňující přihlášení a poskytující rozcestník pro nepřihlášené uživatele.

### **logs**

Složka k uložení logů.

### **logs.txt**

Pokud soubor není přítomen, je vše v pořádku. Pokud existuje a obsahuje nějaké záznamy, je vhodné si je přečíst a popř. na ně reagovat. Logy by se zde měli tvořit pokud není dostupná primární databáze. Záznamy se také tvoří pokud výsledek dotazu nad primární a sekundární databází je rozdílný. V takovém případě je potřeba postupovat podle části, která se tomuto problému věnuje.

### **s\_add\_new\_book.php**

Skript slouží k přidání nové knihy. Použití skriptu je jen pro uživatele `admin` – hlavního knihovníka. Umožňuje načtení dat ze skriptu `s_get_details_from_google.php` a po jejich zaslání zpět i zobrazení a případné uložení. Skript na začátku provede případné přidání knihy, pokud byl poslán příslušný požadavek. Pokud byla kniha přidána zobrazí její údaje na konci stránky. Pokud přišla data ze skriptu `s_get_details_from_google.php`, jsou načítána až prezentační části pomocí funkce `checkAndWrite($variableName)`. Pokud je na Google nalezen obrázek, je ihned uložen, pokud ještě není uložen

obrázek JPEG pod daným ISBN. Ten by ale před načtením ještě uložen být neměl, protože by muselo dojít k pokusu o přidání stejné knihy. Formulář skriptem vytvořené stránky nelze odeslat pomocí klávesy `enter`, protože kód této klávesy je vysílán skenerem na konci čárového kódu.

### **s\_add\_new\_user.php**

Skript slouží k přidání nového běžného uživatele. Klientská část kontroluje formulář a hašuje heslo. Serverová část mj. opět hašuje heslo, využívá přitom funkce `hashPassword` ze skriptu `include/db_connect.php`. Dále serverová část komunikuje s databází. Při požadavku nového uživatele přidá a rovnou i zobrazí údaje právě přidaného uživatele.

### **s\_admin.php**

Výchozí skript po přihlášení pro uživatele `admin` – hlavního knihovníka. Skript zobrazuje přehled počtů záznamu v databázi. Porovnává tabulku vypůjčky v primární a sekundární databázi. A to tak, že porovná počty řádků v tabulkách primární a sekundární databáze. U posledního sta výpůjček porovnává i stav výpůjček. K této kontrole využívá funkce `checkLastHundredOfBorrows()`, která je implementována ve skriptu `include/db_connect.php`. Funkce vrací objekt se zjištěnými údaji, který je dále zpracován a interpretován v tomto skriptu (`s_admin.php`).

Tento skript zobrazuje a umožňuje manipulaci s logy z databázové tabulky `log`. Do této tabulky jsou ukládána hlášení uživatelů (záložka `Nahlásit chybu`) a také některé logy vytvářené systémem.

Skript dále umožňuje exportovat jednotlivé tabulky do CSV souborů. Formát CSV souborů je oproti standardnímu CSV modifikován. Hodnoty jsou oddělené tabelátorem, řádky odřádkováním. CSV soubor s uživateli neobsahuje uživatelská hesla. Haše hesel lze zobrazit pouze přímo v databázi, popř. v souborech záloh databáze. CSV soubor s výpůjčkami neobsahuje původní hodnoty z tabulky, ale čísla uživatelů jsou přeložena na jména uživatelů. ISBN jsou přeložena na základní, člověkem čitelné údaje o knihách.

Další část skriptu načítá údaje ze souborů s logy, uloženými na serveru. Výpůjčního místa se dotáže, jestli nějaké logy má uložené a výsledek zobrazí a interpretuje. Skript umožňuje i změnit heslo pro účet `admin`. Využívá funkci pro hašování ze skriptu `include/db_connect.php`.

### **s\_book\_details.php**

Skript slouží k výpisu údajů o knize a v případě uživatele `admin` i pro změnu údajů o knize. Skript může využít služeb skriptu `s_get_details_from_google`. Pokud byla kniha skriptem nalezena a informace získané se liší s údaji uloženými v databázi, jsou údaje v poli pro změny napsány červeným písmem.

Pokud skript `s_get_details_from_google` nalezne obrázek obálky, aktualizuje ihned obrázek obálky, pokud již není uložen obrázek ve formátu JPEG. Pokud byl uložen obrázek v jiném formátu je ponechán na disku. Ve skriptu je i zakomentovaný řádek, který po odkomentování způsobí smazání obrázku v jiném formátu.

### **s\_borrow\_book\_admin.php**

Skript umožňuje vypůjčit knihu běžnému uživateli prostřednictvím účtu `admin`. Průběh výpůjčky zadávané z účtu `admin` je ten, že ve skriptu `s_book_details.php` je zadán požadavek na půjčení, který je předán skriptu `s_choose_name.php`, kde je určeno komu kniha bude půjčena. Skript `s_choose_name.php` potom předává obě tyto informace skriptu `s_borrow_book_admin.php`. Skript ověřuje, zda-li je výpůjčku možno zadat a po ověření provede příslušné změny v databázi. Po zadání výpůjčky ověřuje soulad evidence volných kopií knihy.

### **s\_borrows.php**

Skript slouží k zobrazení historie výpůjček. U výpůjček je zobrazeno i to, kdo výpůjčku zadal. Uživatel `admin` má možnost vidět všechny výpůjčky. Aktivní výpůjčky zde má možnost ukončovat. Má možnost si zobrazit výpůjčky omezené na daného uživatele. K tomuto slouží tlačítko `Zobrazit výpůjčky v detailu uživatele` (skript `s_settings.php`). Má možnost si zobrazit výpůjčky omezené na danou knihu. K tomu slouží tlačítko `Zobrazit výpůjčky v detailu knihy` (skript `s_book_details.php`). Pokud jsou zobrazeny všechny výpůjčky, jsou rozděleny do stránek podle roku začátku výpůjčky. V tomto zobrazení má možnost si zobrazit jen aktivní výpůjčky daného roku. A nakonec je i možnost zobrazit všechny aktivní výpůjčky nezávisle na roce. Běžný uživatel vidí pouze svou historii výpůjček. Oproti zobrazení historie na výpůjčním místě nemá možnost knihy zde vracet a má možnost vidět, kdo výpůjčku do systému zadal.

### **s\_catalog.php**

Skript má za úkol zobrazit katalog knih. Knih je evidováno asi 1400, proto jsou vypsány na jedné stránce. I do počtu 3000 je načtení stále dostatečně rychlé. Skript pro úsporu dotazů u každé knihy na umístění v kategorii provede pouze jeden dotaz na získání tabulky kategorie, kterou si uloží při běhu do asociativního pole. Je zde umožněno knihy řadit podle všech sloupců kromě sloupce zobrazující počty svazků, umístění a ISBN. Skript umožňuje v klientské části vyhledávat v katalogu pomocí Javascriptu tvorbou CSS selektorů. Z tohoto skriptu je možno přejít na skript `s_book_details.php` pro vybranou knihu. Běžný uživatel má navíc možnost přímo z katalogu přidávat knihy do připraveného seznamu knih. Tento požadavek je obsluhován skriptem `s_prepare_book_list.php`.



## **secondary\_db\_pending\_query**

V této složce se vytváří soubor `waitingQueries`, pokud není přítomen, je vše v pořádku.

## **waitingQueries**

Soubor s frontou dotazů, které mění záznamy v databázi. Tyto dotazy již proběhly v primární databázi, ale ještě neproběhli v sekundární. Soubor vytváří a maže skript `include/db_connect.php` pokud není spojení na sekundární databázi a došlo ke změně primární databáze. Ve výchozím nastavení, je vznik tohoto souboru málo pravděpodobný, protože na veřejném serveru se nachází sekundární databáze, na kterou by mělo být spojení.

## **s\_get\_details\_from\_google.php**

Skript k získání dat o knize ze služby Google. Informace o knize mohou být až na třech stránkách. Data se dají stáhnout již z první stránky. Pokud tam nejsou mohou být ještě na druhé, a pokud ani tam ještě nejsou mohou být až na třetí. Naštěstí z první stránky vede odkaz na druhou stránku a z druhé stránky na třetí. Data jsou načítána pomocí parsování nástrojem `Xpath`. Bohužel nefungovalo jednoduché získání selektorů pomocí prohlížeče a kopírování cesty `Xpath` u prvku, ale musel být nalezen nějaký jiný selektor. K tomu bylo potřeba znalosti `Xpath` a trpělivosti, protože bylo potřeba vyzkoušet více možností. Hledání začíná na první stránce. Definice výchozí stránky pro hledání je ve funkci `firstPageInformations($isbn)` v proměnné `$googleLinkStarts`. V této proměnné je řetězec `https://www.google.com/search?tbm=bks&q=isbn%3A`. Tento řetězec musí pokračovat ISBN, které již neobsahuje pomlčky. Nepovažuji za účelné skript parametrizovat, protože pokud bude odkaz změněn, budou pravděpodobně změněny i `Xpath` cesty k prvkům a tak stejně bude muset být projít a předělán celý skript. V tomto případě tento skript bude spíše inspirací pro další skript. Po skončení skriptu nalezené informace předává skriptu, který o ně požádal. Jsou to skripty `s_book_details.php` a `s_add_new_book.php`. Informace jsou předány v požadavku typu GET. Pokud skript nalezne obrázek obálky knihy a ještě obrázek není uložen ve složce `covers_imgs` ihned ho tam uloží. Avšak záznam do databáze o existenci obrázku udělá až následující skript. V případě skriptu `s_add_new_book.php` je to uživatelskou akcí přidání knihy. V případě skriptu `s_books_details.php` je toto provedeno automaticky.

## **s\_help\_admin.php**

Skript zobrazí nápovědu pro uživatele `admin`, hlavního knihovníka. Text nápovědy je obsažen přímo v textu skriptu.

### **s\_help.php**

Skript zobrazí nápovědu pro běžného uživatele, ať přihlášeného nebo nepřihlášeného. Text nápovědy je obsažen přímo v textu skriptu.

### **s\_choose\_name.php**

Skript slouží k výběru uživatele při použití účtu admin. Stává se to ve dvou situacích. Při vytváření výpůjčky je tomuto skriptu předán požadavek od skriptu s\_book\_details.php a dále je předán požadavek s ISBN a číslem uživatele skriptu s\_borrow\_book\_admin.php. Druhou situací je výběr uživatele k nahlédnutí či změně nastavení. V tomto případě je požadavek s číslem uživatele předáván skriptu s\_settings.php.

### **s\_loggedin.php**

Skript vytváří výchozí stránku po přihlášení běžného uživatele. Zobrazuje uživatelem vypůjčené knihy.

### **s\_logout.php**

Skript slouží k odhlášení uživatele.

### **s\_manage\_groups.php**

Skript pro hlavního knihovníka. Slouží k editaci kategorií knih a jejich umístění.

### **s\_prepare\_book\_list.php**

Skript pro běžné uživatele. Skript slouží k přidání a odebrání knihy ze seznamu knih, který je poté možno zobrazit i na výpůjčním místě. Skript přebírá požadavky na přidání knihy od skriptů s\_book\_details.php a s\_catalog.php. Po přidání knihy do seznamu je uživatel přesměrován na skript s\_catalog.php.

### **s\_report\_error.php**

Skript slouží k nahlášení chyby, resp. poslání vzkazu hlavnímu knihovníkovi.

### **s\_return\_book\_admin.php**

Skript slouží k ukončení výpůjčky hlavním knihovníkem. Skript zajistí potřebné změny v databázi k danému číslu výpůjčky. Po ukončení výpůjčky ověřuje soulad evidence volných kopií knihy.

### **s\_settings.php**

Skript slouží k zobrazení a změně nastavení uživatelského účtu. Změnu jména uživatele může provést jen hlavní knihovník.

## 13 Zálohování, synchronizace a obnovení po havárii

Synchronizace primární a sekundární databáze je popsána v části popisující databázi.

Zálohování primární databáze je prováděno každý den. Jeho provedení je zajištěno skriptem `mysql_dump_Backups/backup.sh`.

Synchronizace obsahu složky `cover_imgs` je také zajištěna skriptem `mysql_dump_Backups/backup.sh`.

Detailnější popis funkcí skriptu `mysql_dump_Backups/backup.sh` najdete u popisu tohoto skriptu. Skript je spouštěn na výpůjčním místě jedenkrát denně uživatelem, který administruje systém a jeho pravidelné spouštění zajišťuje nástroj `cron`.

Obnovení po havárii je možné jak u veřejného serveru, tak u výpůjčního místa na RPI. Systém je postaven tak, aby obnovení nebylo příliš složité.

### 13.1 Obnovení na veřejném serveru

Kompletní návod k instalaci od bodu s funkčním operačním systémem je na přiloženém disku. Zde jsou stručně shrnuty kroky které jsou potřeba k obnovení systému.

K běhu systému byl použit operační systém Ubuntu 20.04. Je vhodné použít stejný nebo příbuzný operační systém. Pro jiný operační systém je možné, že bude potřeba více úprav k znovuobnovení systému.

Na veřejném serveru po instalaci operačního systému je potřeba nakopírovat složky se skripty určenými pro veřejný server do složky, kde jsou webovým serverem očekávány. V případě operačního systému Ubuntu 20.04 s nainstalovaným LAMP serverem je to složka `/var/www/html`, je třeba nastavit přístupová práva. Poté nastavit přístupy k databázi a nahrát stejnou verzi databáze, která je na výpůjčním místě.

### 13.2 Obnovení na výpůjčním místě

Na výpůjčním místě běží systém na RPI. Tento systém má tu výhodu, že je možné jej zkopírovat celou prostou kopií paměťové karty. Poté stačí jen nahrát aktuální verzi databáze, která je shodná s databází na veřejném serveru a systém může běžet dál. Je třeba ale takovou zálohu mít již připravenou dopředu. Při pádu z důvodu selhání paměťové karty a nepřipravené záložní karty s nakopírovaným systémem je potřeba při obnově postupovat podobně, jak je popsáno u veřejného serveru.

Pokud by došlo k pádu, kvůli selhání RPI a karta byla v pořádku. Stačí danou kartu dát do jiného RPI. Nejspíše bude potřeba, aby to byl stejný model RPI. Je třeba pamatovat na to, že se tímto změní MAC adresa a patřičně nastavit síť.

Detailnější popis zálohy, obnovení i instalace celého systému je k dispozici na přiloženém disku.

## 14 Zabezpečení systému

I přesto, že není primárním cílem práce věnovat se zabezpečení systému, přijde mi vhodné se o něm zmínit. PHP skripty mají ošetřené vstupy proti útokům typu XSS a SQL injection. Dále jsou hašována hesla již na klientovi a jsou tedy odesílána jako haš. Spojení s veřejným serverem je pomocí HTTPS protokolu, proto se dá komunikace uživatelů s veřejným serverem považovat za bezpečnou.

Systém je zabezpečen i z pohledu přístupových práv. Ke skriptům a údajům ze záloh primární databáze, kde mohou být přístupové údaje k databázi i účtům uživatelů, může přistupovat jen správce s patřičnými oprávněními v operačním systému.

Komunikace na lokální síti je nešifrovaná. Po průzkumu paketů komunikace s MySQL serverem bylo zjištěno, že přihlašovací jméno do MySQL je posíláno nešifrovaně, heslo je šifrováno, resp. hašováno. Jak silný je haš hesla jsem se nedozvěděl, ale řetězec se mi nezdál nijak dlouhý. Dá se tedy usuzovat, že je použita nějaká slabší hašovací funkce. Dotazy do databáze jsou posílány nešifrovaně. Tzn. login a heslo k autentizaci uživatele (knihovního systému) lze z paketu vyčíst. I když heslo je již hašováno. Pokud by tedy byla lokální síť o více uzlech, byly by možné útoky typu MITM. Tento problém je možné řešit nastavením šifrované komunikace, kterou MySQL podporuje. Dalším řešením by bylo šifrovat komunikaci již na linkové eventuálně síťové vrstvě. Např. pomocí IPsec nebo virtuální privátní sítě. Již od počátku se počítalo s přímým kabelem. I tak je možné fyzicky vypojení kabelu na výpůjčním místě a vedením přes počítač útočníka odchytit heslo k databázi a poté s ní manipulovat. Dalším rizikem by bylo rozšíření této sítě o další uzly nesouvisející se systémem. Motivace k takovým útokům je však malá, protože stejně přístup k výpůjčnímu místu není nijak omezen a jde škodit i přímo přes výpůjční místo.

Ohrožením bezpečnosti je i to, že aplikace běží ve webovém prohlížeči. Tzn., že na výpůjčním místě je možné se dostat i k požadavkům předchozích uživatelů. Čili pokud by měli účet chráněný heslem, tak je možné se dostat k požadavku na přihlášení. Zde je již heslo hašované. Není sice prozrazeno heslo jako takové, ale i s jeho hašem, který se odesílá na server je možné proniknout do účtu vytvořením requestu už s hašem hesla. Nepřišel jsem na jiné řešení, než prohlížeč po sezení zavřít a aplikaci spustit znovu. Naštěstí je systém založen na důvěře ve výchozím nastavení není heslo požadováno.

## 15 Hromadné načtení obrázků obálek

Bylo využito pouze při inicializaci. Nebyl tedy důvod dělat uživatelsky přívětivou verzi. Přijde mi však vhodné postup popsat, kdyby tato jednorázová akce byla

potřeba někdy v budoucnu.

Obrázky jsou získávány pomocí služby Google, která je schopná najít knihu podle ISBN. ISBN musí být ve tvaru bez pomlček. Ve složce serverové části aplikace je skript s cestou `extra/s_covers_inicial.php`. Tento skript očekává soubor ISBNs ve kterém jsou po řádcích ISBN bez pomlček. Postupně se dotazuje a stahuje obrázky do složky s obrázky obálek, čili `cover_imgs`. Skript vytváří soubor `extra/isbnDone`, ve kterém jsou vypsány již proběhlé dotazy. Skript nemusí zvládnout všechny položky najednou. Při počtu knih do 1500 musel být spuštěn dvakrát. Kde skript skončil je poznat ze souboru `extra/isbnDone`. V souboru `extra/ISBNs` pak tedy stačí umazat hotové řádky a pokračovat. Pokud by bylo potřeba tento skript využít v budoucnu, tak je velmi pravděpodobné, že bude položek méně.

Při načítání obrázků obálek došlo k omezení služeb poskytovaných Goolem. Ze stránky, která se zobrazila jsem pochopil, že byl detekován robot a došlo k přerušení poskytování služeb. Je tedy možné, že použití tohoto softwaru porušuje smluvní podmínky poskytování služeb Google. Rozbor tohoto problému je nad rámec práce. Dá se konstatovat, že technicky takovéto stahování je možné. Další použití tohoto skriptu je na zvážení hlavního knihovníka nebo administrátora.

Je třeba nezapomenout, že uživatel, který spouští skripty potřebuje patřičný přístup do složky `cover_imgs` a k souborům `extra/isbnDone` a `extra/ISBNs`. V případě Apache serveru je to uživatel `www-data`.

Po načtení obrázků je potřeba upravit záznamy v databázích. To se provede spuštěním skriptu `extra/update_db_covers_records.php`. Skript sám projde složku `cover_imgs` a provede adekvátní změny v databázi.

## 16 Možná rozšíření a změny

Možnými rozšířeními současného systému by mohly být notifikace pro hlavního knihovníka i vypůjčitele. Toto bylo navrhováno i v zadání. Od implementaci takové funkcionality jsem ale upustil. Tato knihovna funguje jinak, než klasické veřejné knihovny a notifikace by nejspíše více obtěžovali než byly přínosem.

Možným přínosem by mohla být rezervace knihy. Mohlo by to být jednodušší než kontaktovat kolegu přímo.

Možnou změnou by byla změna grafického rozhraní na výpůjčním místě ze současného Xfce4 na Gnome. Výhodou by byla možnost rozšíření o virtuální klávesnici a také toto rozhraní neobsahuje balík `xfce4-screensaver`, ve kterém je pro architekturu ARM-64 chyba a musel být odinstalován. Nevýhodou by na druhou stranu byly větší hardwarové nároky a tím možnost prodloužení odezvy aplikace.

Dalším možným rozšířením, které mě napadá je využití mobilních telefonů. Při lehkém průzkumu jsem zjistil, že lze naskenovat čárový kód i pomocí fotoaparátu mobilního telefonu. Znamenalo by to ale určitý zásah do architektury systému a zvýšení počtu výpůjčních míst. Další výzvou by bylo napojení aplikace na fotoaparát, což by mohlo znamenat, že by bylo jednodušší vyvinout aplikaci pro

mobilní telefon, než upravit stávající webové prostředí. Výhodou by byla možnost zadat výpůjčku na jiném místě než u počítače se skenerem.

## 17 Diskuse

Během vývoje jsem sám uvažoval nad několika cestami, takže k diskusi je jistě více témat, než která zde zmíním. A i na konci vidím několik nedostatků, ze kterých se můžu do budoucna poučit.

První věcí k diskusi je volba stylu psaní PHP skriptů. Při psaní jsem nevyužil objektově orientovaného přístupu a i mi to tak přijde v pořádku, protože tento projekt mi na využití objektově orientovaného programování ani nepřišel vhodný.

Nevyužil jsem architektury MVC nepoužil framework ani šablonovací systém. Ač se mi myšlenka MVC architektury i použití šablonovacích systémů líbí, rozhodl jsem se pro použití čistého PHP bez těchto nadstaveb. Důvodů je více. Učit se pracovat s frameworkem by bylo pravděpodobně více náročné, než napsat skripty celé sám. Frameworky poskytují i funkcionality, které nemusí být potřeba a mohou tedy být i systémově více náročné. Udržovat frameworky aktualizované může být pro správce náročnější. Frameworky mohou zlepšit, ale také zhoršit zabezpečení systému. Také si myslím, že je jednodušší udělat případné úpravy bez nutnosti znát daný framework.

Bohužel jsem se tím ale nevyhnul vzniku tzv. špagetového kódu a ne zcela dodržel zásadu DRY (do not repeat yourself). Ač jsem na začátku měl záměr, aby byl kód přehledný, na jednom místě. Tak se zamýšlím, jestli skutečně je výsledek přehledný pro někoho jiného, kdo by měl případně něco opravovat či doplňovat. Sám na to přijdu až s odstupem času. Plné nedodržení DRY mě naštěstí tolik problémů při implementaci neudělalo. V některých místech mi přišlo vhodnější kód zopakovat, než například kvůli jedné funkci, která se vyskytuje ve dvou skriptech vytvářet zvláštní soubor, nebo kvůli tomu do skriptů zbytečně vkládat, co v nich není potřeba. Předpokládám, že v praxi se spíše používá plné dodržení DRY. Je to jiný přístup, a nejspíše o něm budu v budoucnu více přesvědčen a budu jej dodržovat plně. Vzhledem k tomu, že systém bude nasazen na dvou počítačích s rozdílnými rolemi, tak v tomto případě dalo vyhnout opakování kódu jen těžko. Zde jsem se zase snažil, alespoň takové soubory mít pohromadě a využívat je ideálně celé. K zamyšlení je i vytváření obecnějších funkcí k tvorbě tabulek. To byl problém, který jsem vnímal spíše ke konci implementace. Tabulky sice pokaždé vypisují něco jiného, ale nějaké zjednodušení by nejspíše možné bylo. Otázkou je, jak moc by to práce ušetřilo nebo přidalo. K ušetření by mohlo dojít při uvědomění si tohoto někde z počátku implementace.

K diskusi by byl vývoj i z pohledu softwarového inženýrství, kdy na začátku nebylo zcela jasné, jak systém vlastně bude vypadat, jak bude vyhovovat a jaká data se vlastně budou evidovat. Toto se naneštěstí u vývoje softwaru vyskytnout může. Dá se tomuto pomoci vytvořením prototypu, což jsem i využil. Ale bohužel se i tak stalo, že se bylo potřeba systém přizpůsobovat v různých fázích vývoje. A tak se i například projeví, že funkcionality, která byla plánována jen

k obsluhu jedné události byla potřeba pro událost další, což zase vrací otázku ke stylu psaní. V těchto chvílích bylo patrné, že některé funkcionality mohly být napsány obecněji kvůli znovupoužitelnosti.

Poučením do budoucna je myslet při vývoji na to, že je vhodnější mít jeden konfigurační soubor. Toto jsem si uvědomil při popisování skriptu `logout.php`, kde je například nastavení odhlášení na 10 minut. To už byl kód hotový a nechtěl jsem dělat další zásah.

Dalším poučením je nerozdělení jména a příjmení. Byť jsem nad tím zpočátku při návrhu databáze přemýšlel, tak jsem pro to neviděl důvod. Jména byla uložena v tvaru příjmení jméno. Vše fungovalo a přišlo mi to jako příjemné zjednodušenější. Později ale byl požadavek zobrazovat jméno příjmení v opačném pořadí a už jsem potom musel řešit tento problém.

K diskusi je i vzhled systému. Víím, že nejsem příliš dobrý ani citlivý v tomto směru. Naštěstí to snad nebude mít na funkci vliv a vzhledem i k snad přehlednému stylování by nemusel být problém případně vzhled poupravit. Některé části uživatelského rozhraní mohou být pro někoho zpočátku matoucí. Uživatelské prostředí je pojato místy netradičně kvůli zadání. Od původní myšlenky rozlišovat seznamy na výpůjčním místě zvlášť pro vrácení a zvlášť pro půjčení bylo upuštěno z důvodu, aby bylo potřeba co nejméně kroků k půjčení a vrácení knih. Bylo nastíněno, že velmi pravděpodobný scénář je vrácení a půjčení knihy v jedné návštěvě a bylo by nadbytečné vracet knihy v jednom seznamu a poté otevírat druhý seznam na půjčování. Byl tedy kladen větší důraz na to, aby bylo vše možné provést v co nejméně krocích, oproti intuitivnosti ovládní.

K diskusi je i problém s testováním. Obvykle by se takový systém vyvíjel v týmu, kde se s pohledem někoho jiného dají chyby snadněji odhalit. Ač jsem systém nejednou zkoušel i přímo na katedře s pomocí zaměstnanců, kterým za toto ještě jednou děkuji. Nemůže toto testování odpovídat hloubce testování u profesionálních zakázek. Snad se ale zásadní chyby podařilo odstranit a po nasazení nebude již třeba (příliš) zasahovat.

## Závěr

Byl vytvořen netypický samoobslužný systém pro evidenci výpůjček knih z příruční knihovny Katedry informatiky Katedry informatiky Univerzity Palackého. Systém se skládá ze dvou částí. Jednou je výpůjční místo s dotykovou obrazovkou a skenerem čárových kódů, které je umístěno v knihovně Katedry informatiky. Druhou částí je webové rozhraní pro administraci systému a nastavení uživatelských profilů. Systém zpřehledňuje evidenci knih a výpůjček v knihovně Katedry informatiky Univerzity Palackého. Systém je možno obnovit po pádu bez ztráty dat. I při běhu je zajištěn proti ztrátě dat z důvodu ztráty konektivity. Při ztrátě konektivity je schopen dále poskytovat služby. Systém obsahuje nápovědu pro běžné uživatele i správce knihovny – hlavního knihovníka.



## Conclusions

An atypical self-service system was created for the records of borrowings in the library of the Department of Informatics of Palacký University. The system has two parts. One is a borrowing point with a touch screen and a barcode scanner, which is located in the library of the Department of Informatics. The second part is a web interface for system administration and user profile settings. The system makes easier the records of books and borrows in the library of the Department of Informatics of Palacký University. The system can be restored after a crash without data loss. At runtime is secured against data loss due to connection loss. In case connection loss, it is able to continue to provide services. The system contains help for regular users and library administrator.

## A Obsah přiloženého disku

### **data/**

Data k inicializaci MySQL databáze pro knihovní systém.

### **doc/**

Text práce ve formátu PDF. Obsahuje soubory potřebné pro vygenerování PDF dokumentu textu.

### **instructions/**

Textové soubory, které popisují instalaci i případné obnovení systému. Návod k použitému skeneru [8]

### **pictures/**

Obrázky uživatelského rozhraní systému.

### **src/**

Kompletní adresářová struktura systému.

### **src/knihovna\_rpi/**

Část systému pro výpůjční místo.

### **src/knihovna\_server/**

Část systému pro veřejný server.

### **readme.txt**

Popis obsahu disku.

## Literatura

- [1] Úvodní stránka operačního systému Ubuntu <https://ubuntu.com/>
- [2] Úvodní stránka grafického rozhraní XFCE <https://www.xfce.org/>
- [3] Úvodní stránka MySQL databáze <https://www.mysql.com/>
- [4] Úvodní stránka skriptovacího jazyka PHP <https://www.php.net/>
- [5] Specifikace HTML standardu <https://html.spec.whatwg.org/multipage/>
- [6] W3C CSS domovská stránka <https://www.w3.org/Style/CSS/>
- [7] XPath 3.1 W3C Recommendation 21. 3. 2021  
<https://www.w3.org/TR/2017/REC-xpath-31-20170321/>
- [8] Manuál pro použití skeneru čárových kódů  
<https://aidc.honeywell.com/CatalogDocuments/00-02544%20Rev%20K%202-11.pdf>

Odazy na zde uvedené webové stránky jsou platné k 14. 8. 2021