

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



**Diplomová práce**

**Aplikace pro půjčování nářadí**

**Bc. David Prchlík**

© 2021 ČZU v Praze



## ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. David Prchlík

Systémové inženýrství a informatika  
Informatika

Název práce

**Aplikace pro půjčování nářadí**

Název anglicky

**Application for tool sharing**

---

### Cíle práce

Diplomová práce se zabývá návrhem a vývojem aplikace pro půjčování nářadí mezi lidmi. V aplikaci je možné vyhledat nářadí a vypůjčit si jej za předem určenou částku a na určitou dobu od někoho, kdo tuto aplikaci také používá. Hlavním cílem práce je návrh a implementace aplikace pro mobilní zařízení se systémem Android a její ekvivalent v podobě webové aplikace. Dílčím cílem bude popsat použité technologie a postupy.

### Metodika

Bude navržena struktura a vzhled aplikace pro mobilní zařízení se systémem Android a pro webovou aplikaci. Dále bude navržena struktura databáze, s kterou budou pracovat obě aplikace. Obě aplikace a databáze budou následně implementovány podle návrhu.

Aplikace pro mobilní zařízení se systémem Android bude implementována v prostředí Android Studio v programovacím jazyce Java. Webová aplikace bude implementována v prostředí NetBeans. Back-end bude implementován v jazyce PHP a front-end v HTML a CSS. Databáze bude implementována v dotazovacím jazyce SQL.

Výsledné aplikace budou na závěr otestovány a budou navrženy případné možnosti jejich dalšího rozvoje.

## Doporučený rozsah práce

60-80 stran

## Klíčová slova

Webová aplikace, Android aplikace, Java, PHP, Android, Půjčování náradí

---

## Doporučené zdroje informací

Android Developers [online]. [cit. 2019-06-06]. Dostupné z: <https://developer.android.com/>  
GILMORE, W J. – POKORNÝ, J. *Velká kniha PHP 5 a MySQL : kompendium znalostí pro začátečníky i profesionály*. Brno: Zoner Press, 2011. ISBN 978-80-7413-163-9.  
LACKO, Ľ. *Vývoj aplikací pro Android*. Brno: Computer Press, 2015. ISBN 978-80-251-4347-6.

---

## Předběžný termín obhajoby

2020/21 ZS – PEF (únor 2021)

## Vedoucí práce

Ing. Jiří Brožek, Ph.D.

## Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 19. 2. 2020

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 19. 2. 2020

**Ing. Martin Pelikán, Ph.D.**

Děkan

V Praze dne 31. 03. 2021

### **Čestné prohlášení**

Prohlašuji, že svou diplomovou práci "Aplikace pro půjčování nářadí" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31. 3. 2021

---

## **Poděkování**

Rád bych touto cestou poděkoval vedoucímu práce Ing. Jiřímu Brožkovi, Ph.D. za cenné rady, věcné připomínky, trpělivost, čas a ochotu, kterou mi v průběhu zpracování práce věnoval na konzultacích a v e-mailové korespondenci.

# Aplikace pro půjčování nářadí

## Abstrakt

V této diplomové práci je popsán vývoj webové aplikace a aplikace pro Android, která pomáhá jedněm uživatelům vypůjčit potřebné nářadí od druhých uživatelů, kteří toto nářadí nabízí k vypůjčení za malý poplatek.

V teoretické části jsou popsány technologie použité pro vytvoření těchto aplikací. Nejprve jsou popsány technologie použité pro webovou aplikaci, mezi které patří ve front-endové části HTML a CSS a v back-endové části PHP, Apache a MariaDB. V další části jsou popsány technologie použité pro vývoj Android aplikace, mezi které patří Java, objektově orientované programování, nástroje XML a JSON a http knihovna Volley. Na závěr této části jsou popsány použité programy a nástroje potřebné pro vytvoření této práce.

Ve vlastní práci je již popsán celý vývoj obou aplikací. Nejprve je navržena a vytvořena databáze. Dále je navrženo uživatelské rozhraní webové aplikace. Po návrhu uživatelského rozhraní byla provedena implementace a jsou ukázány a popsány části zdrojových kódů. Následně je podobně navrženo uživatelské rozhraní, provedena a popsána implementace Android aplikace. Na závěr je provedeno testování a jsou sepsány návrhy na další vylepšení.

**Klíčová slova:** Webová aplikace, Android aplikace, Java, PHP, Android, Půjčování nářadí, NetBeans, Android Studio

# Application for tool sharing

## Abstract

This diploma thesis describes the development of web application and application for Android, which helps users to borrow needed tools from other users, who offers this tool to lend for a small fee.

The theoretical part describes technologies used to create these applications. First, there are described used technologies for the web application, which include HTML and CSS in the frontend part and PHP, Apache and MariaDB in the backend part. The next section describes technologies used to develop Android application, including Java, object oriented programming, XML and JSON tools, and http Volley library. At the end of this section, there are described used programs and tools to create these applications.

The practical part describes the whole development process of both applications. First, a database is designed and created. Futhermore, the user interface of the web application is designed. After the design of the user interface, an implementation has been performed and parts of the source code are shown and described. Subsequently, the user interface is similarly designed and the implementation of the Android application is performed and described. At the end, testing is performed and there are written some suggestions for further improvements.

**Keywords:** Web application, Android application, Java, PHP, Android, Tool sharing, NetBeans, Android Studio



# Obsah

<b>1 Úvod .....</b>	<b>13</b>
<b>2 Cíl práce a metodika .....</b>	<b>14</b>
2.1 Cíl práce.....	14
2.2 Metodika.....	14
<b>3 Teoretická východiska .....</b>	<b>15</b>
3.1 Webová aplikace .....	15
3.1.1 Frontend.....	16
3.1.1.1 HTML.....	16
3.1.1.2 CSS.....	17
3.1.2 Backend .....	18
3.1.2.1 PHP .....	19
3.1.2.2 MariaDB a SQL.....	20
3.1.2.3 Apache.....	21
3.2 Android.....	22
3.2.1 Java.....	23
3.2.2 OOP.....	24
3.2.3 XML .....	24
3.2.4 JSON .....	25
3.2.5 Volley .....	26
3.3 Použité nástroje a programy .....	26
3.3.1 XAMPP .....	26
3.3.1.1 PhpMyAdmin .....	27
3.3.2 NetBeans.....	27
3.3.3 Android Studio.....	28
3.3.3.1 Android Emulator .....	28
3.3.4 Postman .....	29
<b>4 Vlastní práce .....</b>	<b>30</b>
4.1 Databáze .....	31
4.1.1 Tabulky.....	31
4.1.2 Atributy.....	32
4.2 Návrh webové aplikace .....	34
4.2.1 Titulní stránka .....	35
4.2.2 Přihlášení .....	36
4.2.3 Registrace a změna údajů o uživateli .....	37

4.2.4	Výsledky vyhledávání.....	38
4.2.5	Detail náradí .....	39
4.2.6	Výpůjčka .....	40
4.2.7	Profil uživatele a detail o uživateli .....	41
4.2.8	Přidání a úprava náradí .....	43
4.3	Implementace webové aplikace.....	44
4.3.1	Části kódu obsažené na všech stránkách.....	44
4.3.2	Titulní stránka.....	46
4.3.3	Výsledky vyhledávání.....	47
4.3.4	Registrace .....	49
4.3.5	Přihlášení.....	50
4.3.6	Detail náradí .....	50
4.3.7	Detail uživatele .....	51
4.3.8	Přidání náradí.....	52
4.3.9	Úprava profilu a náradí .....	53
4.3.10	Změna hesla.....	54
4.4	Návrh Android aplikace .....	55
4.4.1	Titulní stránka.....	55
4.4.2	Přihlášení.....	57
4.4.3	Registrace a změna údajů o uživateli.....	58
4.4.4	Výsledky vyhledávání.....	59
4.4.5	Detail náradí .....	60
4.4.6	Výpůjčka .....	61
4.4.7	Profil uživatele a detail o uživateli .....	62
4.4.8	Přidání a úprava náradí .....	63
4.5	Implementace Android aplikace.....	64
4.5.1	PHP skripty.....	64
4.5.2	Ukázka XML .....	65
4.5.3	Třída pro správu přihlašování.....	66
4.5.4	Menu .....	67
4.5.5	Titulní stránka.....	68
4.5.6	Přihlášení.....	69
4.5.7	Výpis vyhledávání .....	70
4.5.8	Ostatní stránky.....	72
<b>5</b>	<b>Výsledky a diskuse.....</b>	<b>73</b>
5.1	Testování aplikací.....	73
5.2	Návrhy na další vylepšení .....	74
<b>6</b>	<b>Závěr.....</b>	<b>75</b>

<b>7 Seznam použitých zdrojů .....</b>	<b>76</b>
<b>8 Přílohy .....</b>	<b>78</b>

## Seznam obrázků

Obrázek 1: Statická stránka (zdroj: <a href="https://www.javatpoint.com/website-static-vs-dynamic">https://www.javatpoint.com/website-static-vs-dynamic</a> ) .....	15
Obrázek 2: Dynamická stránka (zdroj: <a href="https://www.ausomdigitalsolutions.com/dynamic-web-design-coimbatore">https://www.ausomdigitalsolutions.com/dynamic-web-design-coimbatore</a> ) .....	16
Obrázek 3: Ukázka HTML .....	17
Obrázek 4: Abstraktní databázová vrstva (zdroj: Velká kniha PHP 5 a MySQL: kompendium znalostí pro začátečníky i profesionály.) .....	21
Obrázek 5: Apache server (zdroj: <a href="https://www.researchgate.net/figure/Workflow-of-apache-web-server_fig4_277816114">https://www.researchgate.net/figure/Workflow-of-apache-web-server_fig4_277816114</a> ) .....	22
Obrázek 6: Ukázka XML .....	25
Obrázek 7: Ukázka JSON .....	26
Obrázek 8: ER diagram .....	32
Obrázek 9: ER schéma .....	32

## Seznam tabulek

Tabulka 1: Uživatelé .....	33
Tabulka 2: Náradí .....	33
Tabulka 3: Výpůjčky .....	33

## Seznam drátových modelů

Drátový model 1: Titulní stránka webové aplikace .....	36
Drátový model 2: Stránka pro přihlášení webové aplikace .....	37
Drátový model 3: Stránka pro registraci webové aplikace .....	38
Drátový model 4: Stránka s výsledky vyhledávání webové aplikace .....	39
Drátový model 5: Stránka s detailem náradí webové aplikace .....	40
Drátový model 6: Stránka s výpůjčkou webové aplikace .....	41
Drátový model 7: Stránka s profilem uživatele webové aplikace .....	42
Drátový model 8: Stránka s přidáním náradí webové aplikace .....	43
Drátový model 9: Titulní stránka Android aplikace .....	56
Drátový model 10: Stránka pro přihlášení Android aplikace .....	57
Drátový model 11: Stránka pro registraci Android aplikace .....	58
Drátový model 12: Stránka s výsledky vyhledávání Android aplikace .....	59
Drátový model 13: Stránka s detailem náradí Android aplikace .....	60
Drátový model 14: Stránka výpůjčky Android aplikace .....	61
Drátový model 15: Stránka s profilem uživatele Android aplikace .....	62
Drátový model 16: Stránka pro přidání náradí Android aplikace .....	63

## Seznam ukázek zdrojového kódu

Zdrojový kód 1: Začátek session a převzetí konfigurace .....	44
Zdrojový kód 2: Konfigurace .....	44
Zdrojový kód 3: Začátek každé stránky .....	45
Zdrojový kód 4: Hlavička.....	46
Zdrojový kód 5: Formulář pro vyhledávání .....	47
Zdrojový kód 6: Podmínka vyhledávání a provedení dotazu .....	48
Zdrojový kód 7: Podmínka pro výpis řádku vyhledávání .....	48
Zdrojový kód 8: Výpis řádku vyhledávání.....	48
Zdrojový kód 9: Pole pro heslo .....	49
Zdrojový kód 10: Kontrola hesla .....	49
Zdrojový kód 11: Uložení uživatele do databáze .....	50
Zdrojový kód 12: Zápis přihlášeného uživatele do session.....	50
Zdrojový kód 13: SQL dotaz pro výpis náradí podle ID.....	50
Zdrojový kód 14: Kontrola vlastníka náradí .....	51
Zdrojový kód 15: Výpis výpůjček .....	51
Zdrojový kód 16: SQL dotaz na vyhledání vlastníka z výpůjček.....	52
Zdrojový kód 17: Zobrazení mapy .....	52
Zdrojový kód 18: Kontrola obrázku .....	53
Zdrojový kód 19: Uložení obrázku na server .....	53
Zdrojový kód 20: Aktualizace dat v databázi.....	54
Zdrojový kód 21: Změna hesla .....	54
Zdrojový kód 22: Funkce pro výpis uživatele .....	64
Zdrojový kód 23: Uložení výsledku dotazu do JSON .....	65
Zdrojový kód 24: Výsledek funkce výpis uživatele .....	65
Zdrojový kód 25: XML textového pole .....	66
Zdrojový kód 26: Funkce přihlášení .....	66
Zdrojový kód 27: Funkce přihlášen .....	67
Zdrojový kód 28: Zobrazení menu .....	67
Zdrojový kód 29: Výběr položky z nabídky.....	68
Zdrojový kód 30: Definice formuláře .....	68
Zdrojový kód 31: Přidání položek do nabídky .....	68
Zdrojový kód 32: Předání dat pro vyhledávání .....	69
Zdrojový kód 33: Začátek funkce přihlásit .....	69
Zdrojový kód 34: Reakce na odpověď skriptu .....	70
Zdrojový kód 35: Přiřazení proměnných k parametrům .....	70
Zdrojový kód 36: Pole v odpovědi skriptu.....	71
Zdrojový kód 37: Výpis výsledků vyhledávání.....	71

# 1 Úvod

Každý člověk potřebuje čas od času provést doma menší či větší opravy, případně si dům či byt vylepšit např. poličkou nebo pověšením nového obrazu. Jindy může jít naopak o rozsáhlejší rekonstrukci např. koupelny. Pokud se člověk rozhodne tyto práce realizovat sám svépomocí, může mu v tom bránit především dostatečná vybavenost nářadí. Ne každý totiž nutně nářadí pro danou práci vlastní. Pořízení kompletního nářadí může být ekonomicky náročnější, zejména pokud neplánuje toto nářadí dlouhodobě používat i v budoucnu, například řezačka na dlaždičky nemusí být úplně dobrá investice pro jednorázovou rekonstrukci. Další možností je si nářadí od někoho vypůjčit, ale ne každý, má dostatek známých, kteří vlastní např. bourací kladivo nebo výše zmíněnou řezačku na dlaždičky.

V dnešní době se většinu činností snažíme ulehčit pomocí počítačů a chytrých zařízení. Aplikace popsaná v této práci usnadní komunikaci mezi uživateli, nabízející nářadí s uživateli poptávajícími. Uživatelé si mohou pomocí počítače nebo chytrého telefonu jednoduše vyhledat potřebné nářadí, které nabízí jiný uživatel k vypůjčení. Za malý poplatek tak můžeme danou práci vykonat a neutratit tolik peněz za nové nářadí nebo složitě hledat známého, který dané nářadí vlastní a je ochotný jej půjčit.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Diplomová práce se zabývá návrhem a vývojem aplikace pro půjčování náradí mezi lidmi. V aplikaci je možné vyhledat náradí a vypůjčit si jej za předem určenou částku a na určitou dobu od někoho, kdo tuto aplikaci také používá. Hlavním cílem práce je návrh a implementace aplikace pro mobilní zařízení se systémem Android a její ekvivalent v podobě webové aplikace. Dílčím cílem bude popsat použité technologie a postupy.

### **2.2 Metodika**

Bude navržena struktura a vzhled aplikace pro mobilní zařízení se systémem Android a pro webovou aplikaci. Dále bude navržena struktura databáze, s kterou budou pracovat obě aplikace. Obě aplikace a databáze budou následně implementovány podle návrhu.

Aplikace pro mobilní zařízení se systémem Android bude implementována v prostředí Android Studio v programovacím jazyce Java. Webová aplikace bude implementována v prostředí NetBeans. Back-end bude implementován v jazyce PHP a front-end v HTML a CSS. Databáze bude implementována v dotazovacím jazyce SQL.

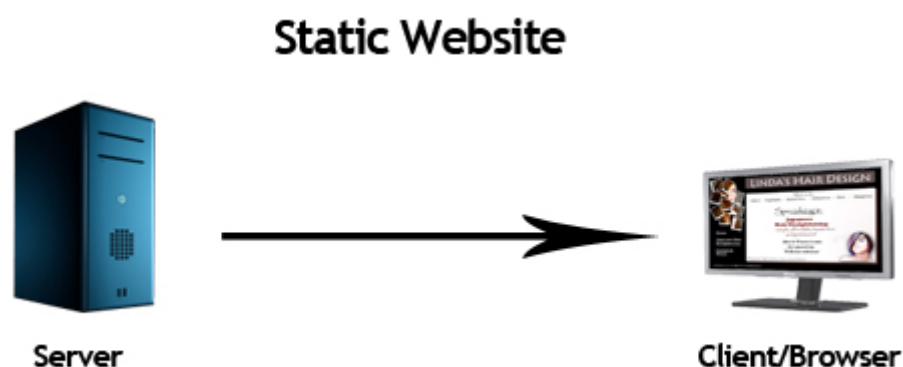
Výsledné aplikace budou na závěr otestovány a budou navrženy případné možnosti jejich dalšího rozvoje.

## 3 Teoretická východiska

### 3.1 Webová aplikace

Webová aplikace se skládá z kombinace statických a dynamických webových stránek. Webová stránka je dokument, který se zobrazuje pomocí internetového prohlížeče. Po zadání URL adresy se uživateli zobrazí data ve formě hypertextu. Kód je vytvořen pomocí značkovacího jazyka HTML nebo XHTML. Napsaný kód lze dále upravovat a stylovat pomocí kaskádových stylů, neboli CSS.

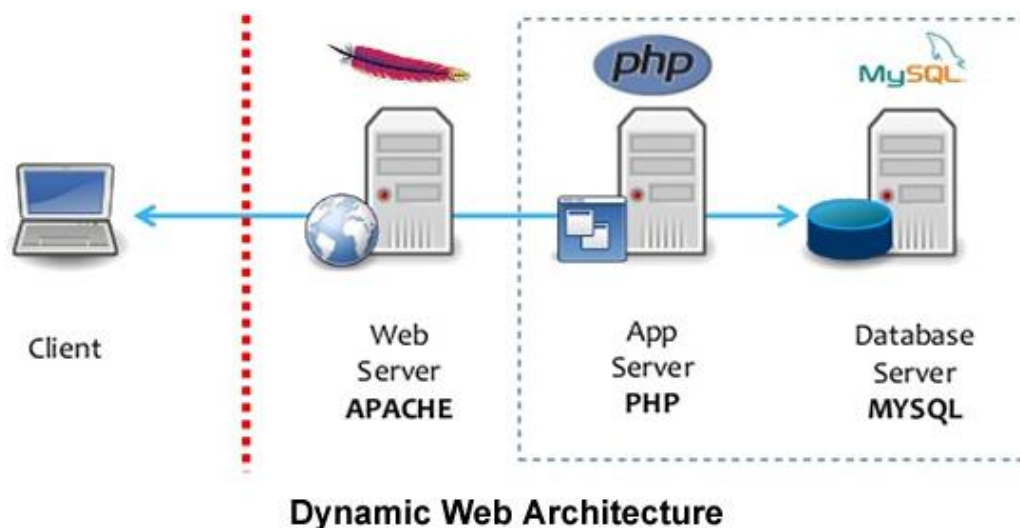
Statická webová stránka je neměnná. Zobrazuje se tedy v internetovém prohlížeči přesně tak, jak je uložena. To zahrnuje webové stránky se stejnými informacemi, které jsou napsány ve značkovacím jazyce HTML a jsou uloženy na webovém serveru. Tyto stránky jsou zobrazeny stejně pro každého uživatele. Statické webové stránky nepotřebují žádné programování nebo databázi, proto jsou levnější a je mnohem jednodušší je vytvořit. Jsou základní formou webových stránek a jsou vhodné pro malé weby. Web s velkým množstvím statických stránek a mnoha informacemi je velmi časově náročný na údržbu.



Obrázek 1: Statická stránka (zdroj: <https://www.javatpoint.com/website-static-vs-dynamic>)

Dynamická webová stránka obsahuje informace a data, která se mohou měnit. Data se mohou například měnit podle uživatele, časové zóny nebo jazyka země, ve které se uživatel nachází. Vyžaduje programování například v PHP nebo ASP.NET a většinou je spojena s databází. Data jsou uložena v databázi nebo systému pro správu obsahu. Pokud je v databázi změněna nějaká informace, zobrazí se tyto změny i na stránce. Když uživatel navštíví dynamickou webovou stránku, jeho prohlížeč pošle data na server. Na základě

těchto dat server v reálném čase vygeneruje HTML stránku. Zobrazí například potřebná data z databáze pouze pro přihlášené uživatele. Dynamické webové stránky jsou dražší a složitější na vytvoření. Jsou ale oproti statickým webovým stránkám mnohem jednodušší na údržbu, mají více funkcí, dynamicky mění data na stránce a podporují CRUD operace (zapisovat, číst, měnit a mazat data z databáze). [1]



Obrázek 2: Dynamická stránka (zdroj: <https://www.ausomdigitalsolutions.com/dynamic-web-design-coimbatore>)

### 3.1.1 Frontend

Přeložení slova frontend do češtiny je přední část. Frontend je část webu, kterou vidí uživatel ve svém prohlížeči. Jedná se tedy o text a jeho stylizaci, celkovou grafickou podobu stránky a rozložení jednotlivých položek. Frontend by měl uživatele zaujmout a pomoci uživateli, aby se na stránce dobře orientoval. Čistě frontendové statické stránky jsou napsané ve značkovacím jazyce HTML a jsou stylizovány pomocí kaskádových stylů CSS. [2]

#### 3.1.1.1 HTML

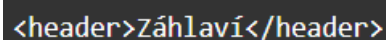
HTML je zkratka pro HyperText Markup Language. Je to hypertextový značkovací jazyk, který se používá k vytváření obsahu webové stránky. Do obsahu webové stránky patří například obrázky, tabulky, texty, multimédia a mnoho dalších. Celou stránku lze



pomocí HTML strukturovat do sekcí a odstavců. HTML je tedy základním stavebním kamenem webových stránek. Hypertextové odkazy odkazují na další webové stránky nebo propojují webové stránky mezi sebou. Při práci s HTML se používají jednoduché konstrukce v podobě prvků (elementů), značek (tagů) a atributů pro vytvoření struktury stránky. Pomocí prvků, neboli elementů, se vyjadřuje, jak se bude uživateli obsah prvku prezentovat. Prvky jsou tvořené z otevírací značky, obsahu a uzavírací značky. Většina elementů obsahuje atributy, které doplňují další informace o obsahu. [3]

Nejnovější verze HTML je HTML5. V této verzi bylo přidáno několik nových funkcí jako je například podpora pro audio nebo video a mnoho nových značek (tagů).

Na obrázku č. 3 lze vidět element záhlaví. Skládá se z otevírací značky (<header>), obsahu (Záhlaví) a uzavírací značky (</header>).



```
<header>Záhlaví</header>
```

**Obrázek 3: Ukázka HTML**

### 3.1.1.2 CSS

CSS je zkratka pro Cascading Style Sheet. Jsou to kaskádové styly, pomocí kterých se upravuje nejen grafický vzhled HTML dokumentu ale také rozložení jednotlivých prvků. Pomocí CSS lze například měnit barvu textu, font, mezery mezi odstavci, velikosti prvků, obrázky nebo pozadí stránky. Lze také upravit rozložení a velikosti prvků pro různá zařízení a velikosti obrazovky a spoustu dalších efektů. Soubor s příponou css je propojen s HTML dokumentem. Různé css soubory mohou upravit HTML dokument a naopak k různým HTML dokumentům lze připojit pouze jeden css soubor a zajistit tak stejný vzhled pro více stránek. Kvůli tomuto je ale nutné dodržovat určitou strukturu HTML dokumentu. Kvůli zrakově hendikepovaným osobám je nutné dodržovat strukturu stránky, protože kaskádové styly by mohly celou stránku přeházet a následně by nemusel fungovat správný překlad

CSS bylo vytvořeno a je upravováno skupinou lidí z W3C. První verze byla vydána v roce 1996. Nejnovější verzí je CSS3, která přidala mnoho nových funkcí pro úpravu webových stránek jako jsou například stíny, kulaté rohy, nové barvy, fonty a mnoho dalších. [4]

### 3.1.2 Backend

V počítačovém světě backend odkazuje na jakoukoliv část webových stránek nebo programu, které uživatelé nevidí. V programovací terminologii je backend vrstva přístupující k datům. Většina webových stránek v dnešní době jsou dynamické, což znamená, že se generují v reálném čase. Dynamická stránka obsahuje jeden nebo více skriptů, které se provedou na webovém serveru pokaždé, když je ke stránce přistupováno. Tyto skripty generují na webové stránce data, která jsou poslána internetovému prohlížeči uživatele. Vše co se stane předtím, než se stránka zobrazí v internetovém prohlížeči, je součástí backendu. Mezi příklady backendových procesů patří například spouštění skriptů pro vygenerování HTML kódu, přístup k datům z databáze pomocí SQL dotazů, šifrování dat, stahování a nahrávání souborů a zpracování vstupů uživatele přes JavaScript. Všechny předchozí příklady, až na poslední, jsou zpracovávány na straně serveru, jsou to tedy server-side procesy. JavaScript se zpracovává na straně uživatele, je to tedy client-side proces a běží v internetovém prohlížeči. JavaScript může být považován za backendový nebo frontendový proces, záleží zda kód postihuje uživatelské rozhraní nebo ne. Backend a frontend spolu spolupracují, aby vytvořili maximální uživatelský komfort. Data, která jsou generována na backendu, jsou posílána na frontend a jsou tak prezentována uživateli. [5]

Pro získání dat, která jsou potřebná zobrazit na webové stránce a tedy vygenerování HTML kódu, se používají skriptovací jazyky. Ve skriptovacím jazyce se zapisují skripty, což jsou série příkazů, které mohou být provedeny bez toho, aniž by musely být zkompileovány. Skriptovací jazyky používají program, který se nazývá interpreter. Tento program slouží pro překlad příkazů, které jsou rovnou interpretovány ze zdrojového kódu a nevyžadují krok kompilace. Programovací jazyky naopak musí být zkompileovány, tedy musí přeložit příkazy do strojového kódu předtím, než jsou tyto příkazy spuštěny. [6]

Mezi nejpoužívanější skriptovací platformy patří PHP a ASP.NET. ASP.NET rozšiřuje .NET platformu s nástroji a knihovnami přímo pro sestavování webových aplikací. Jedná se o framework, který zpracovává webové požadavky ve více programovacích jazycích, jako jsou například C#, F# a Visual Basic. Tato vývojářská platforma byla vytvořena společností Microsoft a je vydána jako open-source software, tedy software s otevřeným zdrojovým kódem. Uživatelé si tak mohou tento kód upravovat. [7]

### 3.1.2.1 PHP

PHP je rekurzivní zkratka pro Hypertext Preprocessor (původně Personal Home Page). PHP je skriptovací jazyk pro generování HTML dokumentů. Kód může být vložen do HTML dokumentu webové stránky. Když je přistupováno na PHP stránku, kód je přečten nebo analyzován serverem, na kterém je tato stránka uložena. Výsledkem PHP funkce na stránce je typicky vrácen HTML kód, který může být přečten internetovým prohlížečem uživatele. Protože je PHP kód transformován do HTML kódu předtím, než je stránka načtena, nemohou si uživatelé zobrazit PHP kód webové stránky. Díky této transformaci jsou PHP stránky bezpečné a mohou tak přistupovat k databázím a dalším zabezpečeným informacím. Mnoho ze syntaxe PHP je zapůjčeno z jiných jazyků jako jsou C, Java a Perl. Na druhou stranu má ale PHP mnoho unikátních vlastností a specifických funkcí. Hlavním cílem tohoto skriptovacího jazyka je dovolit webovým vývojářům psát dynamické webové stránky rychle a jednoduše. PHP je ale také výborným prostředkem pro vytvoření databázově řízených webových stránek. [8]

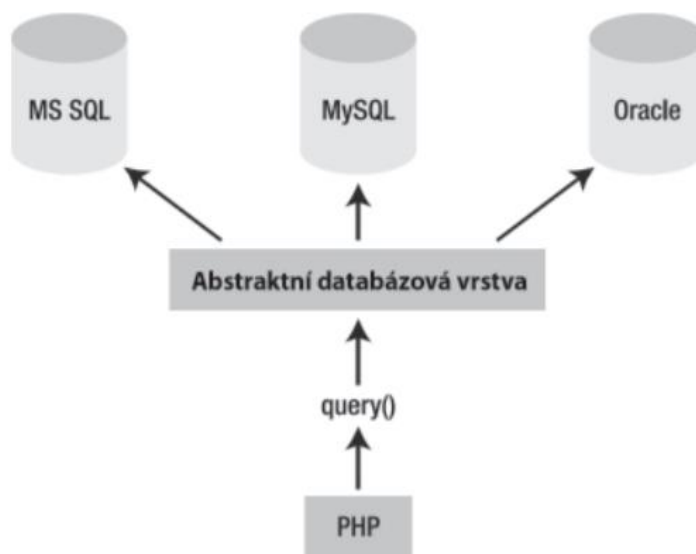
První verzi PHP vytvořil Rasmus Lerdorf v roce 1994. Nyní je nejnovější verze 8.0. Tato verze byla vydána v roce 2020 a přinesla spoustu nových funkcí. Mezi nově přidané funkce patří například nové spojovací typy, pojmenované argumenty, porovnávací výrazy, atributy a mnoho dalších. [9]

### 3.1.2.2 MariaDB a SQL

V této práci byl použit databázový systém MariaDB. Vedle MySQL se jedná o jednu z nejpoužívanějších open-source SQL relačních databází. Mezi další populární databázové systémy se řadí například Microsoft SQL Server a Oracle. MariaDB byla vytvořena původními vývojáři MySQL. Byla navržena pro rychlost, spolehlivost a jednoduchost používání. Je to základní typ MySQL databázového systému, který má většina Linuxových distribucí ve svých uložištích. Tento databázový systém ale používají i uživatelé Windows a MacOS. Je především používán jako náhrada za MySQL databázový systém v balíčku AMP. Tento balíček se používá pro kompletní správu webových aplikací. Obsahuje webový server Apache, databázový systém MariaDB a skriptovací jazyk PHP. Vývoj tohoto databázového systému začal v roce 2009 kvůli získání MySQL společností Oracle. Provádí ale měsíční sloučení s MySQL kódem aby bylo zaručeno, že bude MariaDB obsahovat relevantní opravy chyb, které byly přidány do MySQL. [10]

Pro komunikaci s databázovým systémem se používá dotazovací jazyk. V této práci byl použit jazyk SQL. SQL je zkratka pro Structured Query Language, tedy strukturovaný dotazovací jazyk. Díky tomuto jazyku je možné používat nejen CRUD operace, což jsou vytváření, čtení, úprava a mazání záznamů z databáze, ale i vytvářet, upravovat a mazat tabulky. Mezi tabulkami se vytváří vazby pomocí primárních a cizích klíčů. [11]

Existuje velké množství databázových systémů. Většina respektuje stanovené standardy SQL. I přes velkou podobnost příkazů, může být komunikace s těmito databázemi velmi odlišná. Proto bývají aplikace často spjaté s databázemi a nutí tak uživatele, aby si nainstalovali a pracovali nebo případně se naučili pracovat s danou databází. Kvůli tomuto problému vyvinuli programátoři abstraktní databázovou vrstvu, která odděluje logiku aplikace od logiky komunikace s databází. Příkazy, které souvisí s databází, prochází tímto rozhraním a právě díky tomu může aplikace používat jedno z několika databázových řešení. Ovšem pouze za předpokladu, že databáze podporuje schopnosti, které aplikace požaduje a že daná abstraktní databázová vrstva podporuje daný ovladač, který umí pracovat s danou databází. PHP používá abstraktní databázovou vrstvu PDO, což je PHP Data Object. [12]

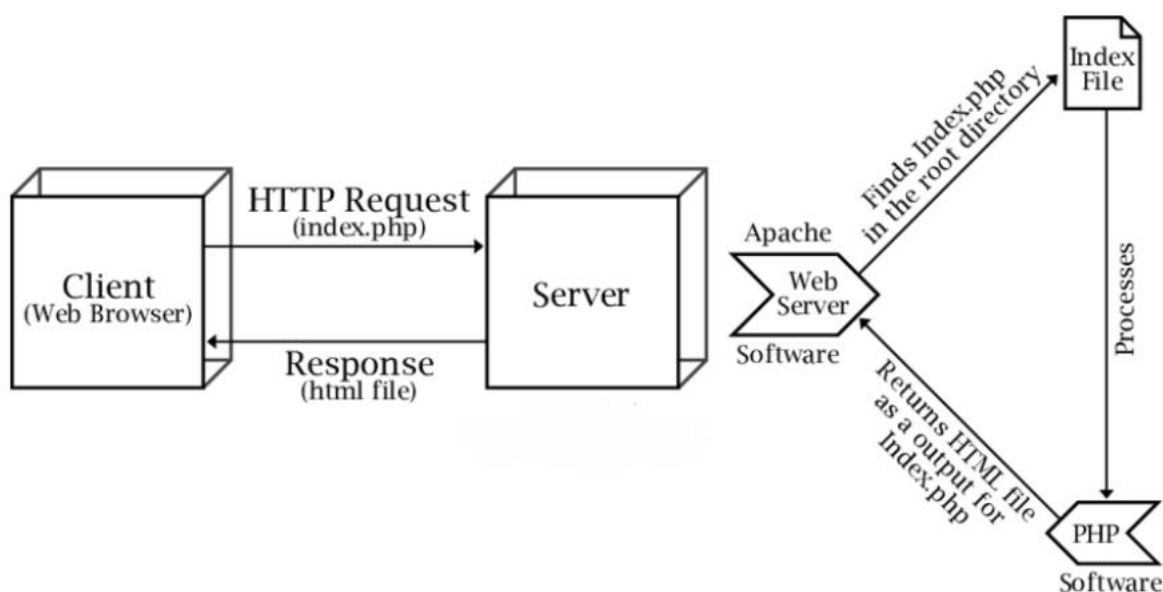


**Obrázek 4: Abstraktní databázová vrstva (zdroj: Velká kniha PHP 5 a MySQL: kompendium znalostí pro začátečníky i profesionály.)**

### 3.1.2.3 Apache

Apache je open-source software pro webový server, na kterém běží okolo 40% webových stránek na celém světě. Oficiální název je Apache HTTP Server a vyvíjí jej společnost Apache Software Foundation. Díky tomuto webovému serveru mohou vlastníci webových stránek zpřístupnit své stránky na internet. Patří mezi nejstarší a nejspolehlivější webové servery s první vydanou verzí v roce 1995. Apache je funkční napříč platformami, takže funguje jak na Unix serverech, tak Windows serverech. Mezi další webové servery se řadí například Nginx, Tomcat a další. Mezi hlavní nevýhody tohoto softwaru pro webové servery patří problémy s výkonem na hodně vytížených serverech a příliš velké množství možností pro konfiguraci, které vedou k chybám v zabezpečení.

Apache není fyzický server, ale je to software, který běží na serveru. Hlavním úkolem je zajistit spojení mezi serverem a prohlížečem uživatele. Když chce uživatel vstoupit na stránku, zadá doménu do adresního řádku v internetovém prohlížeči. Následně webový server předá požadované soubory. Server a klient komunikují skrz HTTP protokol a software Apache je zodpovědný za hladkou a zabezpečenou komunikaci mezi těmito dvěma stroji. [13]



Obrázek 5: Apache server (zdroj: [https://www.researchgate.net/figure/Workflow-of-apache-web-server\\_fig4\\_277816114](https://www.researchgate.net/figure/Workflow-of-apache-web-server_fig4_277816114))

## 3.2 Android

Android je open-source operační systém založený na jádru Linuxu. Je určený primárně pro mobilní zařízení jako jsou chytré telefony a tablety. Vývoj tohoto operačního systému provádí společnost Google pod konsorciem firem Open Handset Alliance. Díky skutečnosti, že je Android open-source, si mohou výrobci mobilních zařízení upravit operační systém na míru zařízení a doplnit nebo upravit další funkce. Většina těchto zařízení se prodává se zabudovanými aplikacemi, čímž podporují aplikace třetích stran. Vývojáři mohou vytvořit aplikace pomocí zdarma přístupného Android SDK (Software Development Kit). Aplikace pro tento operační systém jsou většinou psané v programovacím jazyce Java a běží v JVM (Java Virtual Machine), která je optimalizována pro mobilní zařízení. Uživatelé mohou stahovat aplikace primárně z Google Play. [14]

První verze operačního systému Android byla vydána v roce 2008. Společnost Google používá pro pojmenování jednotlivých verzí názvy sladkostí jako je například verze 5.0 Lollipop. Nejnovější verzí operačního systému je aktuálně verze 11. Tato aktualizace přinesla nové funkce jako je například plovoucí okno, nahrávání obrazovky, lepší zabezpečení a mnoho dalších. Android je v dnešní době nejpoužívanější operační

system mezi mobilními telefony. Jeho zastoupení na trhu činí přibližně 70%. Hlavním konkurentem Androidu je operační systém iOS od společnosti Apple. [15][16]

Jedna z největších výhod tohoto operačního systému je však zároveň i jeho nevýhoda. Jedná se o otevřenost této platformy a možnost úprav, které mohou provádět výrobci i uživatelé. Tyto úpravy se týkají nejen konfigurace nebo widgetů, ale i firmwaru. Nejvíce aplikací je dostupných právě pro platformu Android. Část těchto aplikací ovšem dosahuje pochybné kvality. Proces schvalování aplikací není totiž tak přísný jako u operačních systémů iOS nebo Windows. Zařízení s Androidem dodává hodně firem. Díky tomu je vývoj nových zařízení dynamičtější například oproti společnosti Apple, kde se o vývoj hardwaru stará pouze jedna firma. Nastává ale problém, protože aplikace běží na různých zařízeních s různými procesory a rozlišením obrazovky, což znamená různý komfort uživatelského rozhraní. Proto také v těchto případech nevydává aktualizace Google, ale výrobci zařízení. [17]

### 3.2.1 Java

Java je objektově orientovaný programovací jazyk a počítačová platforma vydaná v roce 1995 společností Sun Microsystems. Java je primárně vyvíjena pro použití na distribuovaném prostředí Internetu. Původně byla vyvíjena pro set-top boxy a ruční zařízení, ale později začala být populární volbou pro vytváření webových aplikací. Byla vyvíjena tak, aby vypadala podobně jako programovací jazyk C++ s jednodušším použitím a s objektově orientovaným modelem. Programy vytvořené v programovacím jazyce Java obsahují třídy, které jsou používány k vytvoření objektů a metod. Java je ale mnohem více striktní než C++, neboť významové proměnné a funkce musí být explicitně definovány. To znamená, že zdrojový kód může vytvářet chyby nebo výjimky mnohem jednodušeji než ostatní programovací jazyky. [18][19]

Na rozdíl od Windows spustitelných souborů (exe) nebo aplikací Macintosh (app), neběží Java programy přímo na operačním systému. Místo toho programy Javy interpretuje Java Virtual Machine (JVM), která může běžet na více platformách. To znamená, že všechny programy, vytvořené v programovacím jazyce Java, jsou multiplatformní a mohou tak běžet na různých platformách. Windows, Macintosh i Linux podporuje Java Runtime

Environment (JRE), která již obsahuje JVM a mohou tak spustit všechny programy napsané v Javě. [19]

### 3.2.2 OOP

OOP je zkratka pro objektově orientované programování. Je to programovací metodologie založená na objektech, namísto pouhých funkcích a procedur. Objekty jsou organizovány do tříd, které dovolují objektům vytvářet společné skupiny. Objekt v OOP programovacím jazyce odkazuje na určitý typ nebo instanci třídy. Každý objekt má podobnou strukturu jako ostatní objekty ve třídě, ale mohou mu být přidány individuální vlastnosti. Objekt také může zavolat funkci nebo konkrétní metodu pro daný objekt.

Objektově orientované programování umožňuje vývojářům lépe organizovat a vytvářet programy. Jednotlivé objekty mohou být změněny bez toho, aniž by pozměnily další části programu. Mohou se tak lépe aktualizovat bez změny celého programu. Je proto mnohem výhodnější používat objektově orientované programování pro velké programy, neboť se velikost programů čím dál více zvětšuje. Mezi nejznámější a nejpoužívanější OOP programovací jazyky patří například Java, PHP, C#, C++ a mnoho dalších. [20]

### 3.2.3 XML

V této práci je XML používáno primárně pro rozložení jednotlivých prvků na obrazovce v Android aplikaci. XML je zkratka pro Extensible Markup Language. Je to jednoduchý a velmi flexibilní textový formát. XML je používáno pro definování formátu dokumentů tak, aby jej mohly číst ostatní aplikace, které jsou s XML kompatibilní. Lze jej využívat společně s HTML stránkami. Používá se i na vytváření značkovacích jazyků pro nějakou konkrétní aplikaci. Může například popisovat data, která mohou být přístupná v momentě načtení webové stránky. Narozdíl od HTML značek, které jsou využívány pro zobrazení dat, XML značky identifikují data a jsou využívány pro ukládání a organizování dat. S XML je možné vytvořit databázi informací bez nutnosti použití databáze. XML primárně používají webové aplikace, ale mohou být využity v mnoha dalších programech. Například jako v této práci v Android Studiu k uchování dat jednotlivých položek na



obrazovce. Na následujícím obrázku je ukázáno, jak vypadá struktura ukládaných dat v jazyce XML. [21][22]

```
<nářadí>
  <id>1</id>
  <název>Vrtačka</název>
  <cena>3000</cena>
  <kategorie>elektrické</kategorie>
</nářadí>
```

Obrázek 6: Ukázka XML

### 3.2.4 JSON

JSON je v této práci používán pro posílání dat mezi PHP skriptem a Android aplikací. Je to zkratka pro JavaScript Object Notation. Je to obdobně jako XML nástroj pro strukturování dat. Tento formát je jednoduchý jak pro vývojáře na psaní, tak i pro stroje je jednoduché tento formát dělit a generovat. Pochází z programovacího jazyka JavaScript, ale je podporován v knihovnách většiny programovacích jazyků. Je velice často používán k posílání dat mezi webovým klientem a webovým serverem. [23][24]

JSON je vytvořený pro dvě struktury. První strukturou je dvojice klíče a hodnoty. V různých programovacích jazycích jsou realizovány jako objekt, struktura, záznam, slovník a další. Druhou strukturou jsou uspořádaná kolekce hodnot. V různých programovacích jazycích jsou realizovány jako pole (array), vektor, sekvence nebo kolekce (list). Jsou to univerzální datové struktury, které podporuje většina moderních programovacích jazyků. Na následující ukázce je zobrazena stejná struktura jako na obrázku č. 5 přepsána do JSON formátu. [24]

```
{
  "nářadí": {
    "id": 1,
    "název": "vrtačka",
    "cena": 3000,
    "kategorie": "elektrické"
  }
}
```

Obrázek 7: Ukázka JSON

### 3.2.5 Volley

Volley je knihovna HTTP pro Android aplikace, díky které je vytváření sítí jednodušší a hlavně rychlejší. Tato knihovna byla vytvořena společností Google v roce 2013. Byla vytvořena, protože v Android SDK chyběly síťové třídy, které jsou schopné pracovat bez zásahu do uživatelské zkušenosti (UX). Knihovna se stará o provedení a zachycení síťových žádostí. Pomáhá tak vývojáři ušetřit čas od psaní stejného kódu síťových volání a mezipaměti pořád dokola. V této práci byla tato knihovna použita pro správu žádostí, které byly zasílány pro práci s databází. [25]

## 3.3 Použité nástroje a programy

Pro vytvoření a správu Apache serveru, na kterém běží databázový systém MariaDB byl použit program XAMPP. Na tomto programu zároveň běží nástroj PhpMyAdmin, ve kterém je možné spravovat databázi. Pro psaní HTML, CSS a PHP kódu byl použit nástroj NetBeans. Ve vývojovém prostředí Android Studio byla napsána Android aplikace v programovacím jazyce Java a vzhled aplikace v jazyce XML. Součástí tohoto vývojového prostředí je i emulátor operačního systému Android pro zobrazení výsledné aplikace. V poslední řadě byl použit program Postman pro kontrolu výsledků PHP funkcí po zaslání žádosti.

### 3.3.1 XAMPP

XAMPP je open-source multiplatformní nástroj, díky kterému je možné vytvořit a spravovat webový server Apache, databázový systém MariaDB, PHP a další doplňky.

Každé písmeno z názvu tohoto nástroje značí jednu z jeho funkcí: X pro multiplatformní (cross-platform), A pro Apache, M pro MySQL, jedno P pro PHP a druhé pro Pearl. XAMPP pomáhá vývojářům testovat webové stránky na lokálním serveru na vlastních počítačích předtím, než jsou uvedeny na hlavní server. Hlavní výhodou tohoto nástroje je již zmíněná multiplatformnost, díky které mohou uživatelé různých operačních systémů jako jsou Windows, Linux a Mac, využívat tento nástroj. Je též velmi oblíbený díky skutečnosti, že je nástroj open-source a je pro všechny uživatele zcela zdarma. Obsahuje také jednoduchý nástroj pro vytváření a úpravu databáze, jejich tabulek a dat, který se jmenuje PhpMyAdmin. [26]

### 3.3.1.1 PhpMyAdmin

PhpMyAdmin je softwarový nástroj, který je zcela zdarma pro všechny uživatele. Je napsán v jazyce PHP a byl vytvořen pro administraci MySQL databáze na webových stránkách. Nástroj podporuje široký rozsah operací na MySQL a MariaDB databázích. Mezi tyto operace patří například správa databáze, tabulek, sloupců, uživatelů, práv, klíčů, vztahů mezi tabulkami a jiné. Tyto operace se provádějí v uživatelském rozhraní, ale uživatel má možnost provést jakýkoliv SQL příkaz. [27]

### 3.3.2 NetBeans

NetBeans je open-source integrované vývojové prostředí (IDE). Používá se pro vývoj aplikací primárně v programovacích jazycích Java, Php a C++, ale podporuje i mnoho dalších programovacích jazyků. NetBeans se také často označuje jako platforma modulárních komponent, která se používá pro vývoj Java desktopových aplikací. Vývojové prostředí NetBeans je napsáno v programovacím jazyce Java. Díky Java virtuálnímu stroji (JVM) běží na většině operačních systémů včetně Windows, Linux a Mac OS. NetBeans se velmi často využívá ve školách a univerzitách pro výuku informačních technologií.

NetBeans spravuje několik funkcí a komponent, mezi které patří například uživatelské nastavení, knihovny, prostor pro ukládání, vývojové nástroje, frameworky a mnoho dalších. NetBeans používá komponenty, které se také nazývají moduly. Tyto moduly umožňují vývoj softwaru. Moduly se dynamicky instalují a umožňují uživateli

stáhnout si další funkce a aktualizace. Součástí tohoto nástroje je JavaME, který slouží k vývoji mobilních aplikací. Ale od doby, kdy se Android stal populárním, se více rozmohlo vývojové prostředí Android Studio a JavaME se přestal používat. Opakovatelná použitelnost NetBeans frameworků zjednodušuje vývoj Java Swing desktopových aplikací. Díky čemuž poskytují možnosti rozšíření platformy i vývojářům třetích stran. [28][29]

### 3.3.3 Android Studio

Android Studio je oficiální integrované vývojové prostředí pro vývoj aplikací pro operační systém Android. Je založeno na vývojovém prostředí IntelliJ IDEA, od které si přidružilo své nástroje pro úpravu kódu a vývojářské nástroje. Pro plnou podporu vývoje aplikací s operačním systémem Android používá Android Studio vytvářecí systém založený na Gradle, emulátor s mnoha funkcemi, kódové šablony a integraci z hostitelské platformy GitHub. Podpora kódových šablon umožňuje vývojáři jednoduše vyhledat ukázky kódu a importovat jej do vlastního projektu. Každý projekt v Android Studiu má jeden nebo více modulů se zdrojovým kódem nebo zdrojovými soubory. Mezi tyto moduly patří aplikační moduly Android, knihovny a moduly Google App Engine.

Android Studio používá funkci Instant Push, která vkládá kód a zdrojové změny do běžící aplikace bez nutnosti restartování. Editor kódu pomáhá vývojáři se psaním kódu, nabízí jeho dokončování, refactoring a analýzu kódu. Při psaní kódu poskytuje Android Studio návrhy v rozbalovacím seznamu. Aplikace vytvořené v Android Studiu jsou kompilovány do formátu APK pro předání do obchodu Google Play. [30][31]

#### 3.3.3.1 Android Emulator

Android Emulator simuluje zařízení s operačním systémem Android na počítači. Díky čemuž je možné testovat aplikace na velkém množství zařízení a úrovně Android API bez potřeby vlastnit každé toto zařízení ve fyzické podobě. Tento emulátor poskytuje téměř všechny schopnosti reálného zařízení Android. Je možné simulovat příchozí hovory a textové zprávy, specifikovat, kde se zařízení nachází, simulovat rotaci a další hardwarové senzory, simulovat různé rychlosti připojení, přístup do obchodu Google Play a mnoho dalších. Testování vyvíjené aplikace může být s emulátorem mnohem jednodušší a rychlejší než s fyzickým zařízením. Například přenášení dat do emulátoru je rozhodně

rychlejší a pohodlnější, než připojovat zařízení přes USB. Emulátor má předdefinované konfigurace různých chytrých telefonů, tabletů, televizí a nositelných zařízení jako jsou například chytré hodinky. [32]

#### **3.3.4 Postman**

Postman je API (rozhraní pro programování aplikací) vývojový nástroj, který pomáhá stavět, testovat a modifikovat tato rozhraní. Téměř všechny potřebné funkce, které by vývojář potřeboval jsou zahrnuty v tomto nástroji. Používá jej více než pět milionů vývojářů každý měsíc, aby si ulehčili vyvíjení svých aplikací. Má možnost provést různé typy HTTP žádostí jako jsou například žádosti GET, POST, PUT a PATCH. Zachovává tak prostředí pro pozdější použití a převod API kódu pro různé jazyky jako je například JavaScript. V této práci byl nástroj Postman použit při testování HTTP žádosti POST pro odeslání dat a nástroj poté vypsál výsledek této žádosti. [33]

## 4 Vlastní práce

V této kapitole je popsán celý postup vývoje aplikací, které umožňují snadné půjčování náradí mezi lidmi. V těchto aplikacích je možné nabídnout náradí na vypůjčení a ostatní uživatelé si jej mohou vyhledat, prohlédnout detail a případně vypůjčit. Pro vypůjčení a nabídnutí náradí je potřebné, aby měl daný uživatel vytvořený účet. Na profilu si může uživatel upravovat svá data a informace o nabízeném náradí.

Ve vlastní práci je nejprve navržena a vytvořena databáze, ve které jsou uchována všechna data, se kterými pracují obě aplikace. Návrh databáze obsahuje také tabulky, sloupce a vztahy mezi těmito tabulkami. Přístupováno do databáze je pomocí PHP. Pomocí PHP přistupuje do databáze i aplikace pro Android a data bude aplikaci předávat ve formě JSON.

Obě aplikace jsou následně navrženy a implementovány. Návrh uživatelského rozhraní začíná titulní stránkou a následně se dělí na další stránky, které jsou k titulní stránce propojeny. Mezi tyto stránky patří přihlášení, registrace, výsledky vyhledávání, profil a přidání náradí. Tyto stránky dále odkazují na detail náradí, detail uživatele, změnu hesla a další. Návrh uživatelského rozhraní každé stránky nebo obrazovky obsahuje krátký popis, use case, scénář a ukázkou designu jednotlivých stránek nebo obrazovek. Ukázky designu jednotlivých stránek jsou zobrazeny ve formě wireframů, neboli drátěných modelů.

K implementaci obou aplikací jsou využity všechny technologie, které jsou popsány v předchozí kapitole. V implementaci jsou zobrazeny a popsány jednotlivé ukázky zdrojových kódů. Implementace jednotlivých aplikací je rozdělena do segmentů. Mezi tyto segmenty patří registrace a přihlašování, se kterými se pojí i zabezpečení. V těchto segmentech bude také popsán princip session pro webovou aplikaci a podobný princip ve správě přihlášení i pro aplikaci pro Android. Mezi další segmenty patří jednotlivé stránky, mezi které se řadí například výsledky vyhledávání, detaily, profil, výpůjčky a další. Některé stránky jsou přístupné pouze pro přihlášené uživatele. Například pouze přihlášený uživatel může zobrazit svůj profil a tím pádem může upravovat informace o svém náradí,

případně ohodnotit výpůjčku. Nepřihlášený uživatel též nemá možnost vidět detailní podmínky výpůjčky včetně kontaktů na majitele náradí.

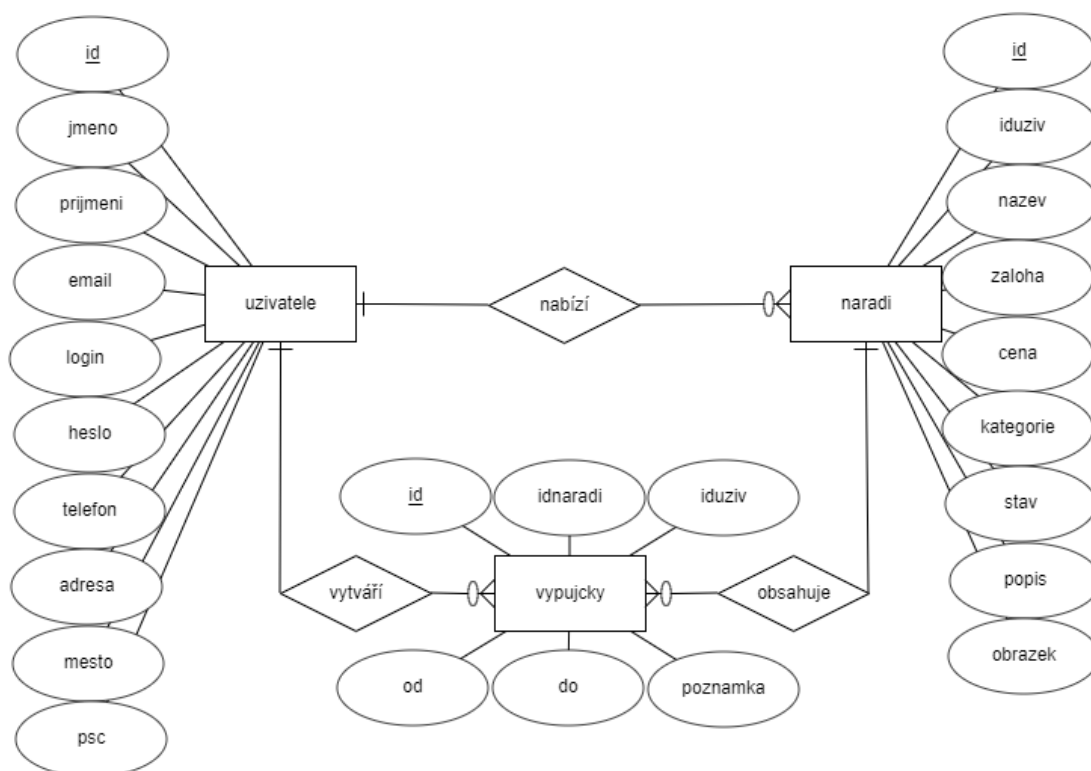
Na závěr je provedeno testování obou aplikací a jsou navrženy další možnosti vývoje a vylepšení aplikací. Jsou zde popsány jednotlivé testované subjekty, výsledky jejich testů a případné připomínky. Návrhy pro další vylepšení aplikací jsou vytvořeny na základě provedených testování. Tyto návrhy se věnují uživatelskému rozhraní a funkcím, které tyto aplikace nabízí.

## **4.1 Databáze**

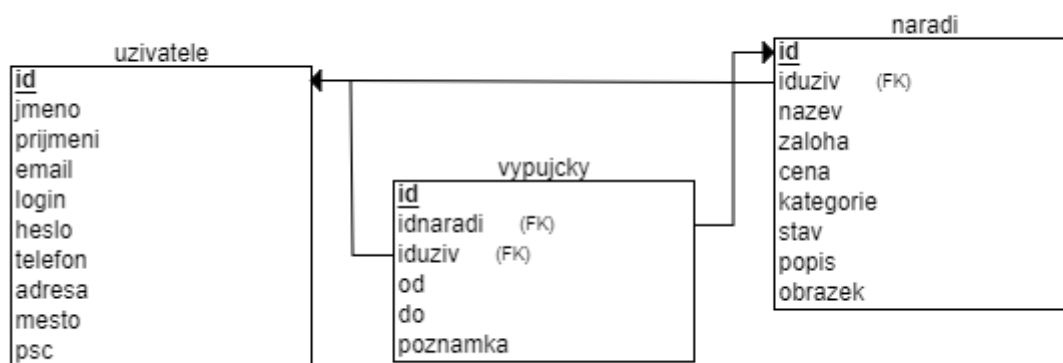
Pro uložení dat, se kterými pracují obě aplikace, byl použit databázový systém MariaDB. Tento databázový systém běží na webovém serveru Apache. Pro práci s daty uloženými v databázi je použit skriptovací jazyk PHP, který generuje HTML kód na webových stránkách a posílá data pro Android aplikaci. Databázový systém MariaDB, webový server Apache a PHP jsou obsaženy v nástroji XAMPP. Tento nástroj obsahuje nástroj PhpMyAdmin, ve kterém je pomocí uživatelského rozhraní vytvořena databáze a tabulky.

### **4.1.1 Tabulky**

Databáze obsahuje tři tabulky. První tabulka má název „uzivatele“, slouží pro uložení dat o registrovaných uživateli. Druhá tabulka s názvem „naradi“ obsahuje data o náradí, které uživatelé nabízí k vypůjčení dalším uživatelům. Poslední tabulkou je tabulka „vypujcky“, která obsahuje data o výpůjčkách těchto náradí. Na následujícím entity relationship diagramu a schématu jsou zobrazeny tyto tři tabulky, jejich atributy a vztahy mezi tabulkami. Entity relationship diagram a schéma byly vytvořeny ve webové aplikaci ERD Plus dostupné na adrese <https://erdplus.com/>.



Obrázek 8: ER diagram



Obrázek 9: ER schéma

#### 4.1.2 Atributy

V následujících tabulkách jsou detailně popsány atributy databázových tabulek.



## Uživatelé

Název atributu	Typ	Velikost	Popis
id	Integer	11	Primární klíč, automaticky číslovaný, identifikační číslo
jmeno	Varchar	20	Jméno uživatele
prijmeni	Varchar	20	Příjmení uživatele
email	Varchar	30	E-mailová adresa
login	Varchar	20	Uživatelské jméno
heslo	Varchar	50	Zašifrované heslo podle uživatelského jména
telefon	Varchar	13	Telefonní číslo
adresa	Varchar	30	Adresa bydliště
mesto	Varchar	20	Město
psc	Varchar	6	Poštovní směrovací číslo

Tabulka 1: Uživatelé

## Náradí

Název atributu	Typ	Velikost	Popis
id	Integer	11	Primární klíč, automaticky číslovaný, identifikační číslo
iduziv	Integer	11	Cizí klíč, identifikační číslo uživatele, který vlastní náradí
nazev	Varchar	50	Název náradí
zaloha	Integer	11	Vratná záloha za půjčení náradí
cena	Integer	11	Cena za den vypůjčení
kategorie	Varchar	30	Kategorie, do které náradí patří
stav	Varchar	20	Stav náradí
popis	Varchar	300	Detailní popis náradí
obrazek	Varchar	40	Název obrázku s příponou

Tabulka 2: Náradí

## Vypůjčky

Název atributu	Typ	Velikost	Popis
id	Integer	11	Primární klíč, automaticky číslovaný, identifikační číslo
idnaradi	Integer	11	Cizí klíč, identifikační číslo půjčovaného náradí
iduziv	Integer	11	Cizí klíč, identifikační číslo uživatele, který si náradí vypůjčuje
od	Date		Datum, od kterého si uživatel náradí vypůjčuje
do	Date		Datum, do kterého si uživatel náradí vypůjčuje
poznámka	Varchar	100	Poznámka k výpůjčce nebo zpráva pro vlastníka náradí

Tabulka 3: Vypůjčky

## 4.2 Návrh webové aplikace

Hlavním úkolem uživatelského rozhraní je, aby uživatel zvládl požadovaný úkol co nejjednodušeji, nejrychleji, a aby se na stránce neztrácel. Proto je kladen největší důraz při návrhu uživatelského rozhraní na jednoduchost a přehlednost. Pokud by se uživatel na stránce ztrácel nebo by mu přišla orientace na stránce moc složitá, případně by nebyl schopný dokončit úkol, kvůli kterému na stránku přišel, stránku by opustil a raději hledal nějakou jinou alternativu.

Při návrhu uživatelského rozhraní každé stránky je nejprve popsáno, co stránka obsahuje a k jakému účelu slouží. Dalším bodem každé stránky je use case, což je případ použití, který se píše z pohledu uživatele. Popisuje, co uživatel na stránce dělá a následně co od stránky očekává, že udělá. Dále je popsán scénář, který je psaný z pohledu systému. Scénář popisuje, jak se systém chová po nějaké události. Na závěr uživatelského rozhraní každé stránky je zobrazen wireframe. Wireframe je drátový model, pomocí kterého je možné zobrazit, jak bude vypadat design uživatelského rozhraní před implementací. Tohoto návrhu se bude držet implementace, aby výsledná aplikace vypadala tak, jak bylo naplánováno. Všechny drátové modely při návrhu uživatelského rozhraní webové aplikace byly vytvořeny v nástroji Mockplus, dostupný na adrese: <https://www.mockplus.com/>.

Každá stránka obsahuje základní možnosti v pravém horním rohu pro přihlášené uživatele nebo případně možnost přihlášení, či registrace. Přihlášený uživatel vidí své uživatelské jméno a má možnost přidat náradí na zapůjčení. Mezi další možnosti patří zobrazení vlastního profilu a možnost odhlášení ze stránky. Use case a scénář těchto možností je napsán pouze na titulní stránce. Na každé stránce je také možnost se vrátit na titulní stránku pomocí kliknutí na logo stránky. Use case a scénář této možnosti je napsán na stránce přihlášení. U drátových modelů stránek, kde uživatel nemusí být přihlášený, jsou zobrazeny možnosti pro nepřihlášené uživatele, tedy přihlášení a registrace. Na stránkách, kde musí být uživatel přihlášený, jsou naopak zobrazeny možnosti pro přihlášené uživatele, tedy zobrazení uživatelského jména a tlačítek pro přidání náradí, zobrazení profilu a odhlášení.

### 4.2.1 Titulní stránka

Titulní stránka je většinou tou první, kterou uživatel vidí. Je tedy nutné, aby byla přehledná. Pokud se uživateli nelíbí titulní stránka nebo se v ní ztrácí, existuje velká šance, že stránku opustí. Titulní stránka obsahuje základní rozcestník na další stránky. Primárně se tato stránka zaměřuje na vyhledávání náradí, které uživatel vyhledává. Proto je na stránce zobrazený formulář, pomocí kterého může požadované náradí vyhledat.

Use case – Uživatel očekává možnosti registrace a přihlášení. Po přihlášení na stránku očekává uživatel možnosti pro přidání náradí, zobrazení vlastního profilu a možnost odhlášení. Dále uživatel očekává zobrazení formuláře pro vyhledávání náradí, který obsahuje název náradí, maximální zálohu, maximální cenu za den, výběr kategorie a stavu. Po vyplnění alespoň jednoho políčka z formuláře a kliknutí na tlačítko vyhledat, uživatel očekává přesměrování na stránku s výsledky vyhledávání s parametry, které zadal.

Scénář – Pokud uživatel není přihlášený, systém nejprve zobrazí tlačítka pro registraci a přihlášení. Pokud je uživatel přihlášený, zobrazí systém uživatelské jméno a tlačítka přidat náradí, zobrazit profil a odhlásit. Následně systém zobrazí formulář, který obsahuje textové pole pro název náradí, maximální zálohu, maximální cenu za den a dvě výběrové pole pro výběr kategorie a stavu. Dále systém zobrazí tlačítko vyhledat, které přesměruje uživatele se zadanými daty na stránku s výsledky vyhledávání.

[Přihlásit](#)[Registrovat](#)

# Půjčovna nářadí

## Vyhledávání

Název nářadí:	<input type="text"/>
Maximální záloha:	<input type="text"/>
Maximální cena za den:	<input type="text"/>
Kategorie:	<input type="text"/>
Stav:	<input type="text"/>
	<input type="button" value="Vyhledat"/>

### Drátový model 1: Titulní stránka webové aplikace

#### 4.2.2 Přihlášení

Stránka s přihlášením by měla být jednoduchá a velmi podobná jako na ostatních stránkách, aby uživatel ihned věděl, co se po něm požaduje. Stránka obsahuje jednoduchý formulář pro zapsání uživatelského jména a hesla.

Use case – Uživatel očekává možnost pro vrácení na titulní stránku. Dále uživatel očekává zobrazení formuláře, který obsahuje pole pro zápis uživatelského jména a hesla. Po vyplnění formuláře a kliknutí na tlačítko *přihlásit* uživatel očekává přesměrování na titulní stránku.

Scénář – Systém nejprve zobrazí logo, které po stisknutí přesměruje uživatele na titulní stránku. Dále systém zobrazí formulář s textovými poli pro zapsání uživatelského jména a hesla. V poslední řadě systém zobrazí tlačítko *přihlásit*, které přesměruje uživatele na titulní stránku, za předpokladu, že je uživatelské jméno a heslo správné.

Přihlásit

Registrovat

## Půjčovna nářadí

### Přihlášení

Jméno:

Příjmení:

### Drátový model 2: Stránka pro přihlášení webové aplikace

#### 4.2.3 Registrace a změna údajů o uživateli

Na stránce registrace jsou zobrazena všechna textová pole, která jsou potřebná pro registraci uživatele. Tato pole jsou: jméno, příjmení, e-mail, uživatelské jméno, heslo, heslo znovu, telefon, adresa, město a PSČ. Stránka pro úpravu údajů o uživateli vypadá velmi podobně, jen neobsahuje pole heslo, heslo znovu a uživatelské jméno. Při úpravě údajů jsou všechna textová pole již předvyplněna údaji o uživateli, která jsou nyní uložena v databázi.

Use case – Uživatel očekává zobrazení formuláře, který obsahuje všechna pole potřebná pro registraci uživatele. V případě úpravy údajů očekává uživatel předvyplnění textových polí. Po vyplnění formuláře uživatel očekává zobrazení možnosti pro odeslání formuláře a přesměrování na titulní stránku, případně na stránku profilu.

Scénář – Systém nejprve zobrazí formulář s textovými poli, která jsou potřebná pro registraci uživatele. V případě úpravy údajů o uživateli systém předvyplní všechna textová pole s daty z databáze. Dále systém zobrazí tlačítko pro potvrzení. Po stisku tlačítka, za

předpokladu, že jsou všechna pole správně vyplněna, systém přesměruje uživatele na titulní stránku, případně na stránku profilu.

Přihlásit

Registrovat

## Půjčovna nářadí

### Registrace

Jméno:	<input type="text"/>
Příjmení:	<input type="text"/>
E-mail:	<input type="text"/>
Uživatelské jméno:	<input type="text"/>
Heslo:	<input type="text"/>
Heslo znovu:	<input type="text"/>
Telefon:	<input type="text"/>
Adresa:	<input type="text"/>
Město:	<input type="text"/>
PSČ:	<input type="text"/>
	<input type="button" value="Registrovat"/>

### Drátový model 3: Stránka pro registraci webové aplikace

#### 4.2.4 Výsledky vyhledávání

Na této stránce se vypisují výsledky vyhledávání dle požadavků uživatele z titulní stránky. Jednotlivá nářadí jsou vypsána v obdélnících, které obsahují základní informace o nářadí a jeho obrázek. Po kliknutí na tyto obdélníky se zobrazí detailní informace o nářadí.

Use case – Uživatel očekává zobrazení výsledků vyhledávání ve formě obdélníků se základními informacemi o nářadí a s obrázkem. Po kliknutí na zvolené nářadí očekává uživatel přesměrování na stránku s detaily o vybraném nářadí.

Scénář – Systém zobrazí výsledky vyhledávání dle požadavků uživatele, které byly poslány z předchozí stránky. Systém zobrazí obdélníky, do kterých vypíše základní

informace o nářadí a jeho obrázek. Po kliknutí na obdélník systém přesměruje uživatele na stránku s detaily o vybraném nářadí.



**Drátový model 4: Stránka s výsledky vyhledávání webové aplikace**

#### 4.2.5 Detail nářadí

Na této stránce jsou vypsány všechny informace o nářadí i s obrázkem. Dále je zde vypsána historie výpůjček daného nářadí, samozřejmě za předpokladu, že toto nářadí již bylo zapůjčeno.

Use case – Uživatel očekává zobrazení všech informací o nářadí včetně jeho obrázku. Pokud má nářadí nějaké výpůjčky, očekává uživatel zobrazení dat těchto výpůjček. Dále uživatel očekává možnost zobrazení údajů o vlastníkovvi nářadí a lokalitě. Uživatel očekává také možnost vypůjčit si vybrané nářadí. V případě, že je uživatel vlastníkem tohoto nářadí, očekává uživatel naopak místo předchozích možností především možnost pro úpravu údajů o nářadí.

Scénář – Systém zobrazí všechny informace o náradí i s obrázkem. Pokud má toto náradí nějaké výpůjčky, systém zobrazí tyto výpůjčky. Následně pokud není přihlášený uživatel vlastník, systém zobrazí tlačítka pro zobrazení základních údajů o vlastníkoví náradí a na vypůjčení vybraného náradí. Tato tlačítka přesměrují uživatele na stránku s detaily o uživateli, případně na stránku s výpůjčkou. Pokud je přihlášený uživatel vlastník náradí, systém zobrazí tlačítko pro úpravu údajů o náradí. Po kliknutí na toto tlačítko systém přesměruje uživatele na stránku s úpravou informací o náradí.

Přihlásit Registrovat

## Půjčovna náradí

### Detail náradí

- Název náradí
- Záloha
- Cena na den
- Kategorie
- Stav
- Popis

Zobrazit uživatele a lokalitu Vypůjčit

### Výpůjčky

Od - Do

#### Drátový model 5: Stránka s detailem náradí webové aplikace

### 4.2.6 Výpůjčka

Tato stránka obsahuje krátký formulář pro zadání dat, pro které si chce uživatel náradí vypůjčit a zprávu pro majitele náradí.

Use case – Uživatel očekává zobrazení formuláře, který obsahuje datum, od kterého si chce uživatel náradí vypůjčit, dále datum, do kterého si chce uživatel náradí



vypůjčit a textové pole pro zaslání zprávy majiteli. Dále uživatel očekává možnost potvrzení této výpůjčky a přesměrování na stránku s detailem náradí.

Scénář – Systém zobrazí formulář s políčky od kdy a do kdy si chce uživatel náradí vypůjčit a textové pole pro zapsání zprávy majiteli náradí. Dále systém zobrazí tlačítko vypůjčit. Po stisknutí tlačítka systém přesměruje uživatele na stránku s detailem o náradí.

Přihlášený uživatel

## Půjčovna náradí

Výpůjčka

Od:

Do:

Zpráva pro majitele:

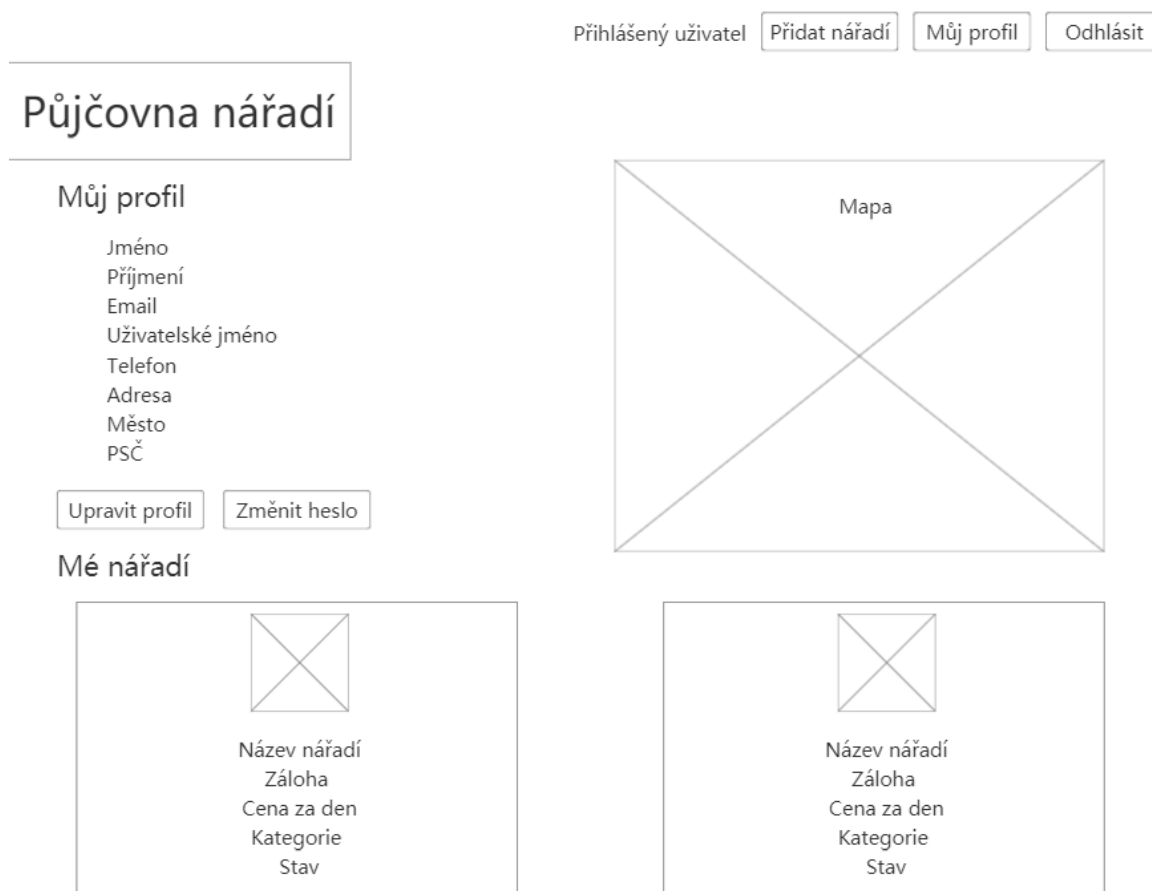
### Drátový model 6: Stránka s výpůjčkou webové aplikace

#### 4.2.7 Profil uživatele a detail o uživateli

Na této stránce je zobrazená většina informací o uživateli. Na pravé straně je zobrazená mapa s bydlištěm uživatele. Uživatel má možnost na svém profilu změnit své údaje nebo heslo. Ve spodní části je vypsáno náradí, které daný uživatel vlastní. Stránka je velmi podobná stránce s detailem o uživateli. Na rozdíl od profilu uživatele zde nejsou možnosti pro změnu údajů a změnu hesla. Mapa ukazuje pouze přibližnou polohu bydliště uživatele, pokud nemá uživatel od tohoto uživatele nic vypůjčeného.

Use case – Uživatel očekává zobrazení většiny informací o uživateli. Následně uživatel očekává zobrazení mapy, na které je ukázaná poloha bydliště. Dále pokud se jedná o profil přihlášeného uživatele, očekává uživatel možnosti pro změnu údajů a změnu hesla. Uživatel též očekává zobrazení všech nářadí, které vlastní a možnost si je po kliknutí prohlédnout. V případě detailu o uživateli očekává uživatel přibližnou polohu vlastníka nářadí, pokud nemá žádné nářadí vypůjčené.

Scénář – Systém zobrazí většinu informací o uživateli. Pokud se jedná o profil přihlášeného uživatele, zobrazí systém tlačítka pro změnu údajů a změnu hesla. Pokud se jedná o uživatele, který není vlastníkem profilu a nemá nic vypůjčené, zobrazí systém mapu s přibližnou polohou bydliště uživatele. V opačném případě zobrazí systém mapu s přesnou polohou bydliště uživatele. Dále systém zobrazí podobně jako u výsledků vyhledávání nářadí, které uživatel vlastní. Po stisknutí tlačítka systém přesměruje uživatele na příslušnou stránku.



**Drátový model 7: Stránka s profilem uživatele webové aplikace**

#### 4.2.8 Přidání a úprava náradí

Tato stránka obsahuje formulář se všemi potřebnými údaji o přidávaném náradí. Stránka s úpravou náradí je velmi podobná a předvyplňuje data do políček z databáze.

Use case – Uživatel očekává zobrazení formuláře s textovými poličky pro zadání názvu náradí, zálohy, ceny za den a popisem náradí. Dále uživatel očekává možnosti výběru z nabídky kategorií a stavu. V případě úpravy údajů o náradí očekává uživatel předvyplnění všech předchozích polí. Dále uživatel očekává možnost pro výběr obrázku. Následně uživatel očekává možnost pro potvrzení tohoto formuláře a přesměrování na titulní stránku.

Scénář – Systém nejprve zobrazí formulář, který obsahuje textové pole pro zápis údajů o náradí a výběrové pole pro výběr kategorie a stavu náradí. Pokud se jedná o úpravu informací o náradí, systém předvyplní všechna políčka daty z databáze. Dále systém zobrazí tlačítka pro výběr obrázku a potvrzení formuláře. Po kliknutí na tlačítko s výběrem obrázku systém zobrazí průzkumník souborů a po kliknutí na tlačítko potvrdit systém přesměruje uživatele na titulní stránku, pokud jsou všechny údaje správně zadány.

Přihlášený uživatel

### Půjčovna náradí

#### Přidat náradí na vypůjčení

Název náradí:	<input type="text"/>
Záloha:	<input type="text"/>
Cena na den:	<input type="text"/>
Kategorie:	<input type="text"/>
Stav:	<input type="text"/>
Popis:	<input type="text"/>
Obrázek:	<input type="button" value="Vybrat soubor"/>
	<input type="button" value="Vypůjčit"/>

#### Drátový model 8: Stránka s přidáním náradí webové aplikace

## 4.3 Implementace webové aplikace

Tato část se věnuje implementaci webové aplikace. Jsou zde postupně zobrazeny ukázky zdrojového kódu, který je následně popsán. Implementace je rozdělena do segmentů, které obsahují jednotlivé stránky, či důležité funkční prvky jako je zabezpečení, se kterou se pojí přihlašování a session. Pro uchování rozsahu práce v rozumném rozmezí, jsou zdrojové kódy, které jsou podobné, ukázány pouze jednou. Na každé stránce je ale uvedeno, že se zde podobný zdrojový kód nachází a je popsáno, v čem se liší. Celý zdrojový kód je obsažen v příloze práce.

### 4.3.1 Části kódu obsažené na všech stránkách

Většina ukázaných částí zdrojového kódu v této části je přebírána pomocí funkce `include`.

```
session_start();  
include("konfigurace.php");
```

**Zdrojový kód 1: Začátek session a převzetí konfigurace**

Na každé stránce se nejprve spustí session pomocí funkce `session_start()`. Pomocí této funkce je možné sdílet informace v proměnných mezi stránkami. Do těchto proměnných se přistupuje pomocí proměnné `$_SESSION`. Tato funkce se nejčastěji používá k přihlašování. Lze si například uložit informaci, zda je uživatel přihlášen a jeho uživatelské jméno. Dále se tato funkce používá například v online obchodech pro ukládání položek do nákupního košíku. Pomocí funkce `include` je převzatý skript *konfigurace.php*.

```
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "pujcovna";  
$conn = new mysqli($servername, $username, $password, $dbname);
```

**Zdrojový kód 2: Konfigurace**

Každá stránka obsahuje skript konfigurace, protože pracuje s databází. Nejdříve se nastaví proměnné s přihlašovacími údaji do databáze, které jsou výchozí z programu XAMPP. Následně se vytvoří nová instance. Pomocí tohoto krátkého skriptu se přistupuje do databáze.

```
<html>
  <head>
    <?php include("hlava.php"); ?>
  </head>
  <body>
    <div id="wrapper">
      <header>
        <p>
          <?php
            include("hlavicka.php");
          ?>
        </p>
        <a href="index.php">
          <div id="logo">
            <h1>Půjčovna náradí</h1>
          </div>
        </a>
      </header>
      <div id="page">
```

**Zdrojový kód 3: Začátek každé stránky**

Na začátku každé stránky je nejprve skript s názvem *hlava.php*. Tento skript obsahuje metadata stránky jako jsou klíčová slova, popis a kódování. Obsahuje také odkaz na CSS soubor s kaskádovými styly pro úpravu vzhledu stránek. Elementy *div* jsou používány pro úpravu určitých částí stránek. Každá stránka obsahuje kliknutelné logo, které vrátí uživatele na titulní stránku. V hlavičce každé stránky se přebírá skript *hlavicka.php*.

```

if (isset($_SESSION["prihlasen"])) {
    if ($_SESSION["prihlasen"] === true) {
        echo $_SESSION["login"] . "
            <a href=\"pridat.php\">Přidat nářadí</a>
            <a href=\"profil.php\">Můj profil</a>
            <a href=\"odhlaseni.php\">Odhlásit</a> ";
    } else {
        echo "<a href=\"prihlaseni.php\">Přihlásit</a>
            <a href=\"registrace.php\">Registrovat</a>";
    }
} else {
    $_SESSION["prihlasen"] = false;
    include 'hlavicka.php';
}

```

#### Zdrojový kód 4: Hlavička

V hlavičce se používá session pro zobrazení odkazů pro přihlášené, či nepřihlášené uživatele. V první řadě se zkontroluje, zda je proměnná nastavená a dále se zkontroluje, zda tato proměnná obsahuje hodnotu true. Pokud ano, vypíše se odkazy pro přihlášeného uživatele, v opačném případě se vypíše odkazy pro nepřihlášeného uživatele.

#### 4.3.2 Titulní stránka

Titulní stránka obsahuje formulář pro vyhledávání nářadí. Tento formulář posílá vyplněná data pomocí metody get stránce *vyhledavani.php*. Tato metoda posílá vyplněná data přes url adresu. Formulář obsahuje jednoduchá textová políčka a nabídku pro zvolení kategorie a stavu.

```

<h2>Vyhledávání</h2>
<div id="blok">
  <form action="vyhledavani.php" method="get"
    enctype="multipart/form-data">
    <div class="box" id="box1">
      Název nářadí: <input type="text" name="nazev"><br><br>
      Maximální záloha: <input type="text" name="zaloha">
      <br><br>
      Maximální cena na den: <input type="text" name="cena">
      <br><br>
      Kategorie: <select name="kategorie">
        <option></option>
        <option>Elektrické nářadí</option>
        <option>Zahradní technika</option>
        <option>Čistící technika</option>
        <option>Ruční nářadí</option>
        <option>Příslušenství</option>
        <option>Obráběcí stroje</option>
        <option>Pneumatické nářadí</option>
        <option>Stavba</option>
        <option>Motorové stroje</option>
        <option>Svářecí technika</option>
        <option>Jiné</option>
      </select><br><br>
      Stav: <select name="stav">
        <option></option>
        <option>Nové</option>
        <option>Lehce opotřebené</option>
        <option>Velmi opotřebené</option>
      </select><br><br>
      <input type = "submit" value = " Vyhledat " /><br>
    </div>
  </form>
</div>

```

Zdrojový kód 5: Formulář pro vyhledávání

### 4.3.3 Výsledky vyhledávání

Stránka s výsledky vyhledávání dostává zadaná data z titulní stránky, podle kterých vypíše výsledky vyhledávání.

```

if (!empty($_GET["nazev"]) || !empty($_GET["zaloha"]) ||
    !empty($_GET["cena"]) || !empty($_GET["kategorie"]) ||
    !empty($_GET["stav"])) {
$sql = "SELECT * FROM naradi";
$result = $conn->query($sql);
$count = 0;
while ($row = $result->fetch_assoc()) {

```

**Zdrojový kód 6: Podmínka vyhledávání a provedení dotazu**

V podmínce se nejprve zkontroluje, zda je alespoň jedno políčko zadané. Pokud ano, pošle se SQL dotaz databázi a výsledek dotazu se uloží do proměnné. Proměnná *count* slouží pro kontrolu, zda se vypsalo alespoň jeden řádek v následující podmínce. Cyklus se opakuje pro každý řádek z výsledku SQL dotazu.

```

if ((empty($_GET["nazev"]) ||
false !== (strpos($row['nazev'], $_GET["nazev"]))) &&
(empty($_GET["zaloha"]) || $row['zaloha'] <= $_GET["zaloha"]) &&
(empty($_GET["cena"]) || $row['cena'] <= $_GET["cena"]) &&
(empty($_GET["kategorie"]) || $row['kategorie'] == $_GET["kategorie"]) &&
(empty($_GET["stav"]) || $row['stav'] == $_GET["stav"])) {

```

**Zdrojový kód 7: Podmínka pro výpis řádku vyhledávání**

Tato podmínka kontroluje, zda zadaný název alespoň z části odpovídá názvu uloženému v daném řádku z databáze, nebo zda je záloha nebo cena menší, než zadané hodnoty. Dále kontroluje, zda zvolená kategorie nebo stav odpovídá uloženým hodnotám v řádku z databáze. Uživatel nemusí zadat všechny hodnoty, proto je tato podmínka splněna, pokud všechny zadané hodnoty odpovídají hodnotám v daném řádku z databáze a ostatní tak mohou být prázdné.

```

echo "<a href=\"detail.php?id=" . $row["id"] .
    "\"><div id=\"vbox\"> <img src=\"uploads\\" . $row["obrazek"] .
    "\" height=\"100\"><br>Název náradí: " . $row["nazev"] .
    "<br>Záloha: " . $row["zaloha"] .
    "<br>Cena na den: " . $row["cena"] . "<br>Kategorie: " .
    $row["kategorie"] . "<br>Stav: " . $row["stav"] . "</div></a>";
$count++;

```

**Zdrojový kód 8: Výpis řádku vyhledávání**



Pokud je tedy podmínka splněna, vypíše se v obdélníku náradí s obrázkem a základními informacemi o náradí. Celý tento obdélník slouží jako odkaz, který uživatele přesměruje na detail o vybraném náradí a pošle id náradí. Dále se proměnná *count* zvedne o jednu hodnotu. Tato proměnná je dále použita pouze pro kontrolu, zda se vypsalo alespoň jeden řádek.

#### 4.3.4 Registrace

Stránka s registrací obsahuje rozsáhlý formulář. Většina políček z formuláře jsou velmi podobná. Pro lepší přehlednost je zobrazeno pouze políčko s heslem.

```
Heslo: <input type="password" name="heslo"><span class="error">*<?php echo $hesloer; ?></span><br><br>
```

##### Zdrojový kód 9: Pole pro heslo

Proměnná *hesloer* obsahuje chybu, pokud nastala, především díky špatně zadanému heslu. Hvězdička označuje pole, které je povinné zadat.

```
if (strlen($_POST["heslo"]) < 6) {  
    $hesloer = "Heslo musí mít minimálně 6 znaků."  
} elseif ($_POST["heslo"] <> $_POST["heslo2"]) {  
    $hesloer = "Hesla se musí shodovat."  
} else {  
    $heslo = test_input($_POST["heslo"]);  
    $cheslo = crypt($heslo, $login);  
}
```

##### Zdrojový kód 10: Kontrola hesla

Tato podmínka kontroluje, zda má heslo alespoň 6 znaků a zda se shoduje s kontrolním heslem, zadaným v následujícím políčku. Pokud heslo nevyhovuje, je nastavená proměnná s chybou hesla, které se vypisuje ve formuláři. Následně se heslo zkontroluje, zda neobsahuje nějaké řetězce, které by mohly narušit zabezpečení. V poslední řadě se heslo zašifruje podle uživatelského jména.

```
$sqlk = "INSERT INTO uzivatele (jmeno, prijmeni, email, login, heslo,
telefon, adresa, mesto, psc) VALUES ('" . $jmeno . "', '" .
$prijmeni . "', '" . $email . "', '" . $login . "', '" .
$cheslo . "', '" . $telefon . "', '" . $adresa . "', '" .
$mesto . "', '" . $psc . "')";";
```

#### Zdrojový kód 11: Uložení uživatele do databáze

Pokud byla všechna pole zadaná správně, uloží tento příkaz uživatele do databáze.

#### 4.3.5 Přihlášení

Stránka s přihlášením obsahuje pouze krátký formulář s políčky pro uživatelské jméno a heslo. Pokud se uživatelské jméno a zašifrované heslo podle uživatelského jména shodují, zapíše se do session, že je uživatel přihlášen. Jeho uživatelské jméno a identifikační číslo uloží pomocí následujícího kódu. Případné odhlášení nastaví hodnotu *prihlasen* na false.

```
$_SESSION["prihlasen"] = true;
$_SESSION["login"] = $login;
$_SESSION["id"] = $row['id'];
```

#### Zdrojový kód 12: Zápis přihlášeného uživatele do session

#### 4.3.6 Detail náradí

Na této stránce jsou zobrazeny všechny informace o daném náradí. Nejprve je proveden následující SQL dotaz.

```
$sql = "SELECT * FROM naradi where id=\"" . $_GET["id"] . "\"";
```

#### Zdrojový kód 13: SQL dotaz pro výpis náradí podle ID

Výsledek tohoto dotazu je pouze jedno náradí, které odpovídá identifikačnímu číslu poslanému z předchozí stránky. Výsledek je vypsan pomocí funkce echo podobně jako na předchozí stránce.

```

if ($row["iduziv"] == $_SESSION["id"]) {
    echo "<br><a href=\"upravanaradi.php?id=" . $row["id"] .
        "\">Upravit nářadí</a>";
} else {
    echo "<br><a href=\"detailuzivatele.php?id=" . $row["iduziv"] .
        "\">Zobrazit uživatele a lokalitu</a><br><br>
        <a href=\"vypujcka.php?id=" . $row["id"] .
        "\">Vypůjčit</a><br><br>";
}

```

#### Zdrojový kód 14: Kontrola vlastníka nářadí

Tato podmínka kontroluje, zda je přihlášený uživatel vlastníkem zobrazovaného nářadí. Pokud je vlastníkem, zobrazí se odkaz pro úpravu nářadí, který zároveň posílá id nářadí. V opačném případě se zobrazí odkazy pro zobrazení uživatele, který posílá id uživatele a pro vypůjčení nářadí, který posílá id nářadí.

```

$sqlv = "SELECT * FROM vypujcky where idnaradi=\"" . $_GET["id"] . "\"";
$resultv = $conn->query($sqlv);
if ($resultv->num_rows > 0) {
    echo "<h3>Výpůjčky</h3>";
    while ($rowv = $resultv->fetch_assoc()) {
        echo "<br>Od: " . $rowv["od"] . " - Do: " . $rowv["do"];
        if ($_SESSION["prihlasen"]) {
            if ($row["iduziv"] == $_SESSION["id"]) {
                echo " Zpráva: " . $rowv["poznamka"];
            }
        }
    }
}

```

#### Zdrojový kód 15: Výpis výpůjček

Následně se provede SQL dotaz, který najde všechny výpůjčky daného nářadí podle id. Pokud výsledek tohoto dotazu obsahuje alespoň jeden řádek, vypíše se všechny výpůjčky daného nářadí.

### 4.3.7 Detail uživatele

Stránka s detaily o uživateli zobrazí uživatele podle identifikačního čísla, které bylo poslané z předchozí stránky. Tato stránka se dělí na tři části. V první části jsou vypsány informace o uživateli, který vlastní dané nářadí. Pokud má přihlášený uživatel od daného uživatele vypůjčené nějaké nářadí, má zobrazeny všechny informace. V opačném případě nevidí kontaktní informace. V druhé části je zobrazena mapa s přibližným nebo přesným

místem bydliště. Opět záleží, zda má přihlášený uživatel vypůjčené nějaké nářadí od daného uživatele. V poslední části je vypsáno všechno nářadí, které daný uživatel vlastní. Poslední část je velice podobná stránce s výsledky vyhledávání, s tím rozdílem, že se v SQL dotazu na tabulku nářadí vyhledává pouze podle identifikačního čísla uživatele.

Stránka profilu přihlášeného uživatele vypadá velmi podobně jako stránka s detaily uživatele. V první části obsahuje všechny informace o uživateli a v druhé části je na mapě zobrazena přesná lokace bydliště uživatele. Třetí část je zobrazena stejně. Pouze místo identifikačního čísla poslaného ze stránky detail nářadí, se na profilu pracuje s identifikačním číslem uloženým v session. Profil obsahuje navíc dvě tlačítka, která odkazují na změnu údajů a změnu hesla.

```
$sqlx = "SELECT u.id FROM naradi AS n LEFT JOIN vypujcky AS v
ON v.idnaradi=n.id LEFT JOIN uzivatele AS u
ON n.iduziv=u.id WHERE v.iduziv=" . $_SESSION["id"];
```

**Zdrojový kód 16: SQL dotaz na vyhledání vlastníka z výpůjček**

Pomocí tohoto dotazu je možné zjistit identifikační číslo vlastníka nářadí, a zda má přihlášený uživatel vypůjčené nějaké nářadí od daného uživatele. Díky tomuto opatření lze alespoň z části chránit vlastníka nářadí a zobrazit jen některé údaje.

```
echo "<iframe width=\"500\" height=\"350\" src=\"https://maps.google.com/
maps?q=" . $mapa . "&output=embed\"></iframe>";
```

**Zdrojový kód 17: Zobrazení mapy**

Mapa je zobrazena pomocí aplikace Google Maps. Proměnná *mapa* obsahuje lokaci, kterou má Google Maps zobrazit. Lokace může být buď přesná adresa vlastníka, nebo poštovní směrovací číslo. Lokace na mapě se nastavuje podle již výše uvedených důvodů.

#### 4.3.8 Přidání nářadí

Na tuto stránku se mohou dostat pouze přihlášení uživatelé. Stránka obsahuje formulář s textovými políčky a nabídkami pro výběr kategorie a stavu, podobně jako na

titulní stránce. Tato stránka navíc obsahuje textové pole pro zadání popisu náradí, kontrolu vstupů a uložení obrázku.

Stránka výpůjčka se velmi podobá této stránce a stejně jako tato stránka se podobá stránce s registrací. Tyto stránky se liší pouze vstupy, které uživatel zadává a které mají za úkol se uložit do databáze.

```
if (empty($_FILES["file"])) {
    $obrazek = "Obrázek je povinný";
} else {
    $obrazek = $_FILES['file']['name'];
    $pripona = pathinfo($obrazek, PATHINFO_EXTENSION);
    $cesta = getcwd() . '/uploads/' . $_FILES['file']['name'];
    if ($pripona != "jpg" && $pripona != "png" && $pripona != "jpeg"
        && $pripona != "gif") {
        $obrazek = "Podporujeme pouze JPG, JPEG, PNG a GIF.";
    }
}
```

#### Zdrojový kód 18: Kontrola obrázku

Obrázek se posílá pomocí vstupu typu file, tedy souboru. Nejprve se zkontroluje, zda byl obrázek vybrán. Poté se připraví obrázek k odeslání na server a zkontroluje se, zda má obrázek podporovanou příponu. Pokud alespoň v jednom případě obrázek neodpovídá kritériím, vypíše se chyba ve formuláři.

```
move_uploaded_file($_FILES['file']['tmp_name'], $cesta);
```

#### Zdrojový kód 19: Uložení obrázku na server

Pokud jsou všechny položky formuláře správně zadány, obrázek se uloží na serveru a do databáze se uloží název obrázku s příponou.

### 4.3.9 Úprava profilu a náradí

Stránky s úpravou profilu a úpravou náradí jsou velmi podobné stránkám s registrací a přidáním náradí. Obsahují velmi podobné formuláře bez některých polí. Na těchto stránkách je nejprve nalezen daný uživatel nebo náradí podle identifikačního čísla v databázi. Hodnoty z databáze jsou předvyplněny do atributu *value* v textových polích. Všechny hodnoty, které uživatel zadal, jsou opět testovány.

```
$sqlk = "UPDATE naradi SET nazev='" . $nazev . "', zaloha='" .
    $zaloha . "', cena='" . $cena . "', kategorie='" . $kategorie
    . "', stav='" . $stav . "', popis='" . $popis .
    "' WHERE id='" . $id . "'";
```

#### Zdrojový kód 20: Aktualizace dat v databázi

Pokud jsou všechny potřebné hodnoty zadány správně, jsou následně aktualizována i data v databázi pomocí příkazu update. Zdrojový kód je ukázaný pouze pro úpravu náradí. Pro úpravu profilu je zdrojový kód velmi podobný a liší se pouze v názvu tabulky, sloupců a proměnných.

#### 4.3.10 Změna hesla

Stránka se změnou hesla obsahuje pouze formulář, který obsahuje 3 textová pole pro zápis starého hesla, nového hesla a zopakování nového hesla. Staré heslo se musí shodovat s heslem uloženým v databázi.

```
if ($_POST['nove1'] == $_POST['nove2']) {
    $noveheslo = crypt($_POST['nove1'], $_SESSION["login"]);
    $sqlu = "UPDATE uzivatele SET heslo='" . $noveheslo . "'
    WHERE login='" . $_SESSION["login"] . "'";
```

#### Zdrojový kód 21: Změna hesla

Pokud jsou nová hesla stejná, zašifruje se heslo podle uživatelského jména. Jelikož formulář neobsahuje uživatelské jméno, musí být převzato ze session. Následně se zašifrované heslo aktualizuje v databázi na řádku, který se nalezne podle uživatelského jména.

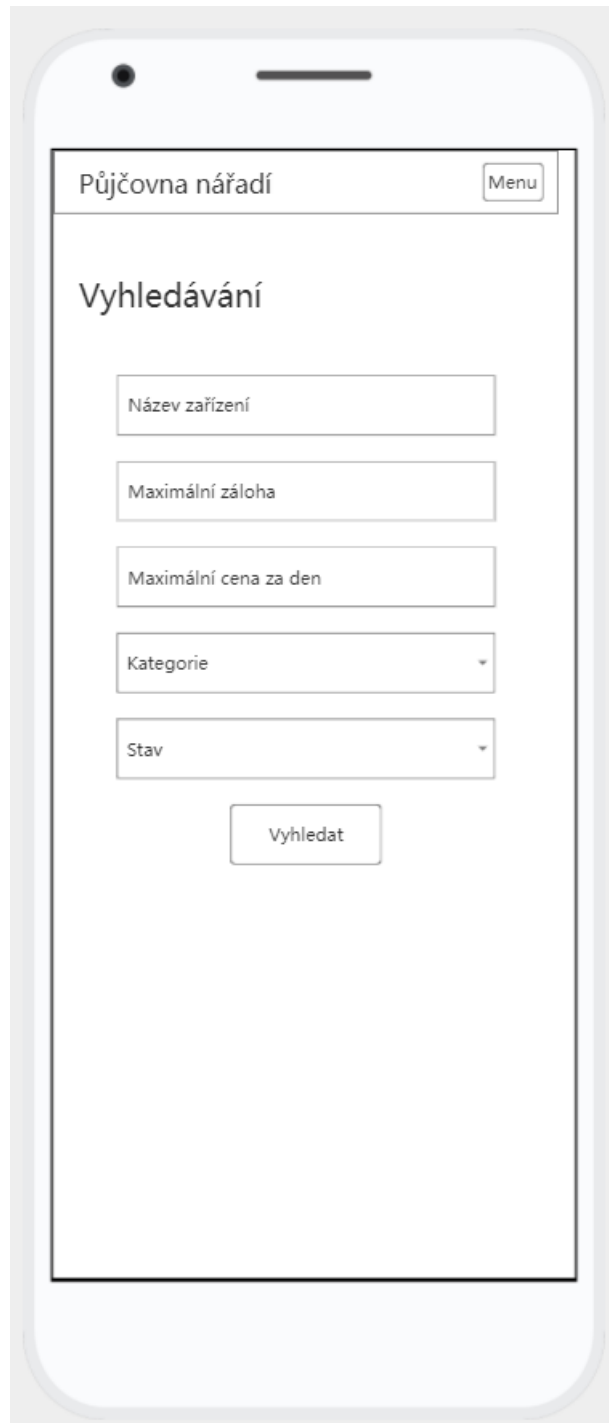
## 4.4 Návrh Android aplikace

Hlavní důraz při návrhu uživatelského rozhraní pro Android aplikaci je kladen na jednoduchost a přehlednost. Zároveň musí být dodrženy určité standardy pro aplikace pro mobilní dotyková zařízení. To například znamená, že funkční prvky na obrazovce nesmí být moc malé, aby je uživatel mohl bez obtíží stisknout a nestiskl omylem jiné prvky. Jelikož je Android aplikace velmi podobná webové aplikaci, jednotlivé návrhy obrazovek obsahují pouze popis a drátový model bez případů použití a scénářů.

Všechny stránky obsahují v pravém horním rohu tlačítko menu. Po stisknutí tohoto tlačítka se zobrazí nabídka, která je různá pro přihlášené a nepřihlášené uživatele. Nepřihlášený uživatel má v této nabídce možnosti na přihlášení a registraci. Přihlášený uživatel uvidí možnosti pro přidání náradí na výpůjčku, zobrazení profilu a odhlášení. Tyto možnosti jsou schované pod jedním tlačítkem na rozdíl od webové aplikace, kde jsou zobrazené přímo na stránce.

### 4.4.1 Titulní stránka

Titulní stránka obsahuje formulář pro vyhledávání. Uživatel může vyhledávat po zadání alespoň jednoho políčka z tohoto formuláře. Po stisknutí tlačítka vyhledat, mu aplikace zobrazí stránku s výsledky vyhledávání. Uživatel může zadat část názvu hledaného náradí, maximální zálohu, maximální cenu za den vypůjčení nebo vybrat z nabídky kategorii či stav.

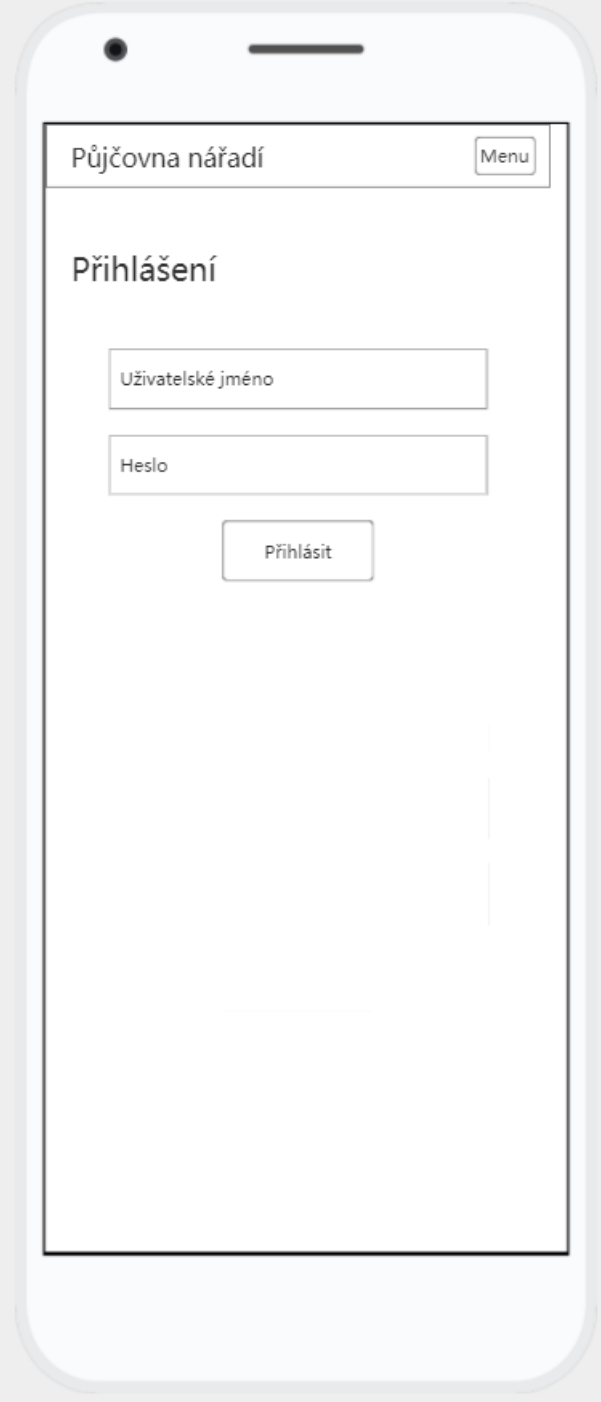


**Drátový model 9: Titulní stránka Android aplikace**



#### 4.4.2 Přihlášení

Stránka s přihlášením obsahuje jednoduchý formulář s políčky pro zadání uživatelského jména a hesla.



Půjčovna náradí Menu

### Přihlášení

Uživatelské jméno

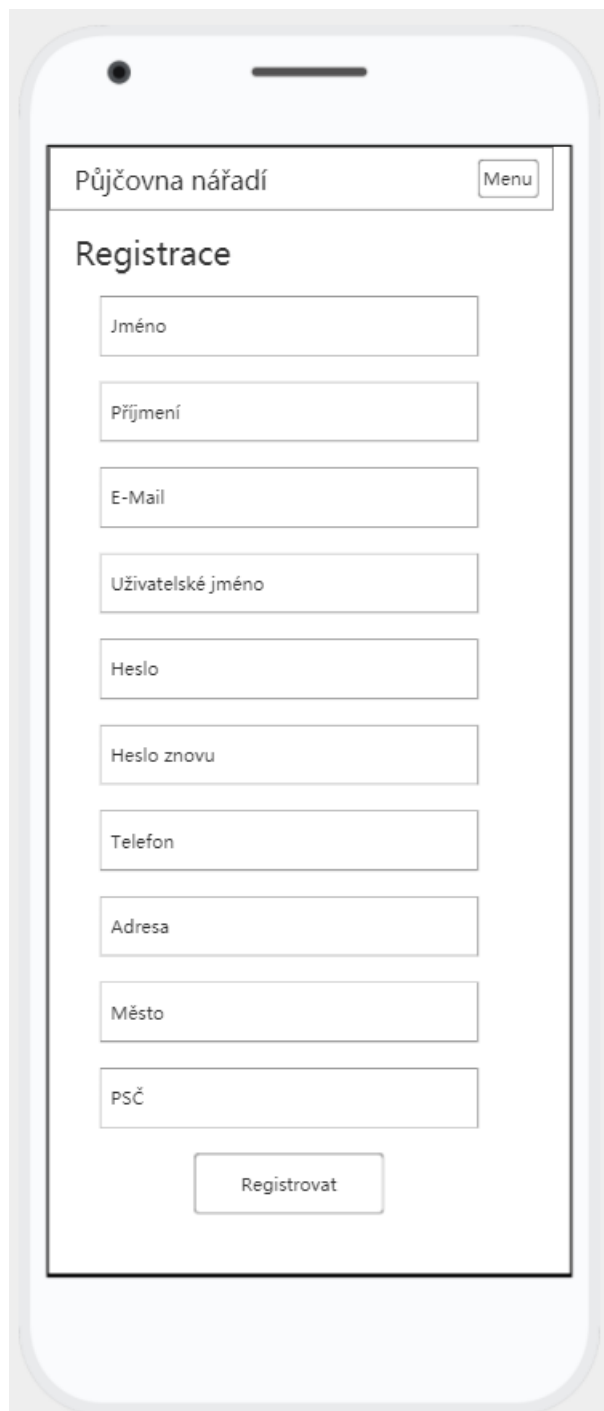
Heslo

Přihlásit

**Drátový model 10: Stránka pro přihlášení Android aplikace**

#### 4.4.3 Registrace a změna údajů o uživateli

Stránka s registrací obsahuje formulář s políčky, do kterých uživatel zadá všechny potřebné údaje. Stránka se změnou údajů je velmi podobná stránce s registrací. Jen neobsahuje políčka pro vyplnění hesel a uživatelského jména. Na změnu hesla existuje speciální stránka, přičemž uživatelské jméno je neměnné.

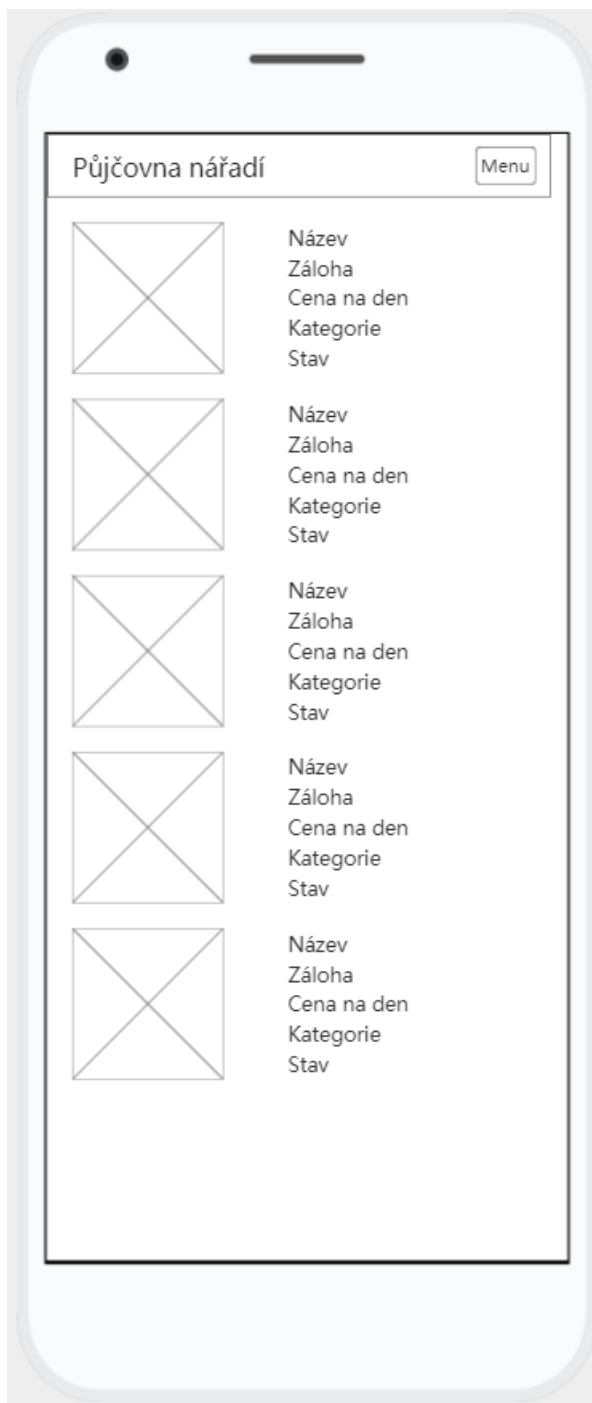


The image shows a wireframe of a registration form on an Android device. The form is titled "Půjčovna náradí" (Tool Rental) and has a "Menu" button in the top right corner. The main heading is "Registrace" (Registration). The form contains the following input fields: "Jméno" (Name), "Příjmení" (Surname), "E-Mail", "Uživatelské jméno" (Username), "Heslo" (Password), "Heslo znovu" (Password again), "Telefon" (Phone), "Adresa" (Address), "Město" (City), and "PSČ" (Postal Code). A "Registrovat" (Register) button is located at the bottom of the form.

**Drátový model 11: Stránka pro registraci Android aplikace**

#### 4.4.4 Výsledky vyhledávání

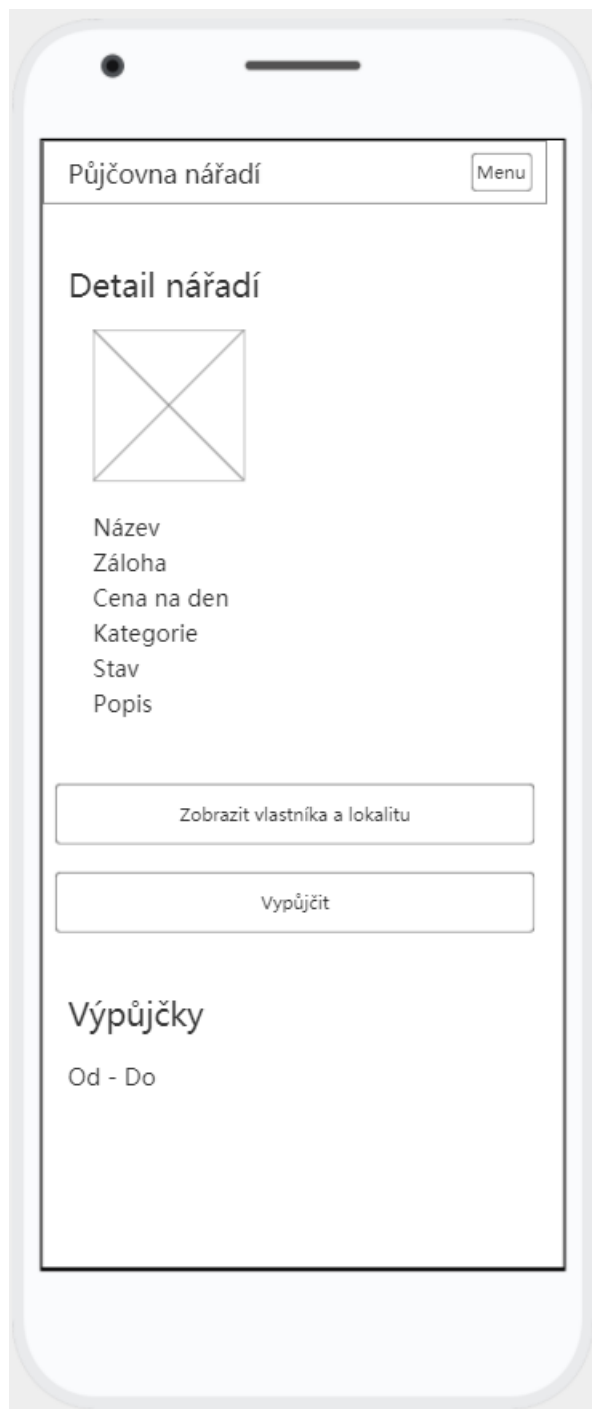
Stránka s výsledky vyhledávání zobrazí všechna nářadí, která splňují kritéria zadaná uživatelem na titulní stránce. V levé části je zobrazen obrázek a v pravé části základní údaje o nářadí. Jednotlivá nářadí jsou vypsaná pod sebou a po stisku vybraného nářadí je uživatel přesměrován na stránku s detailními informacemi o nářadí.



Drátový model 12: Stránka s výsledky vyhledávání Android aplikace

#### 4.4.5 Detail nářadí

Stránka s detailem nářadí obsahuje všechny údaje o nářadí včetně obrázku. Vlastník nářadí má možnost upravit informace o nářadí. Ostatní uživatelé mohou zobrazit údaje o vlastníkovi a lokaci nebo si mohou nářadí rovnou vypůjčit. Dále je na této stránce vidět historie výpůjček tohoto nářadí.



**Drátový model 13: Stránka s detailem nářadí Android aplikace**

#### 4.4.6 Výpůjčka

Tato stránka obsahuje krátký formulář pro zadání od kdy a do kdy si chce uživatel dané náradí vypůjčit. Dále obsahuje textové pole pro zadání zprávy majiteli náradí.

Půjčovna náradí Menu

### Výpůjčka

Od (YYYY-MM-DD)

Do (YYYY-MM-DD)

Zpráva pro majitele

Vypůjčit

**Drátový model 14: Stránka výpůjčky Android aplikace**

#### 4.4.7 Profil uživatele a detail o uživateli

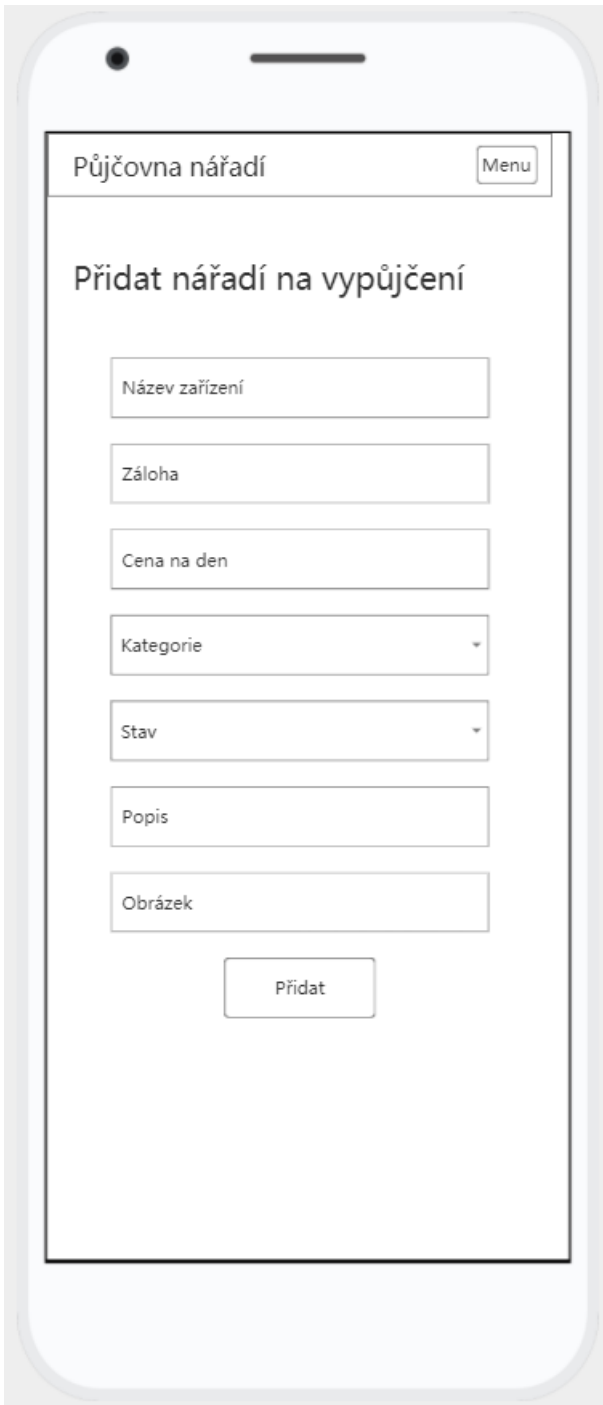
Na profilu uživatele je zobrazena většina jeho údajů. Uživatel zde má možnost upravit profil nebo změnit heslo. Dále tato stránka obsahuje výpis nářadí, které uživatel vlastní. Stránka s detailem o uživateli je velmi podobná, jen neobsahuje možnosti pro úpravu profilu a změnu hesla.



Drátový model 15: Stránka s profilem uživatele Android aplikace

#### 4.4.8 Přidání a úprava nářadí

Stránka s přidáním nářadí obsahuje formulář s políčky, do kterých uživatel zadá informace o přidávaném nářadí. Stránka s úpravou nářadí předvyplní políčka informacemi z databáze.



The screenshot shows a mobile application interface for adding equipment. At the top, there is a header bar with the text "Půjčovna nářadí" on the left and a "Menu" button on the right. Below the header, the main title "Přidat nářadí na vypůjčení" is centered. The form consists of several input fields: "Název zařízení" (text), "Záloha" (text), "Cena na den" (text), "Kategorie" (dropdown menu), "Stav" (dropdown menu), "Popis" (text), and "Obrázek" (text). At the bottom of the form is a "Přidat" button.

**Drátový model 16: Stránka pro přidání nářadí Android aplikace**

## 4.5 Implementace Android aplikace

V této části práce je popsána implementace Android aplikace. Jsou zde zobrazeny ukázky zdrojového kódu, které jsou následně popsány. Jelikož nelze z Android aplikace napřímo komunikovat s databází, je nutné použít PHP skripty, které provádí operace s databází a aplikaci posílají zpět data v JSON formátu. Celá implementace je rozdělena do segmentů. Mezi tyto segmenty patří ukázka PHP skriptů, krátká ukázka rozvržení stránky v XML, použité třídy a poté aktivity.

Pro zajištění správného chodu síťových operací, je vytvořena třída žádosti. Tato třída se stará o frontu jednotlivých žádostí, díky které zajistí správný chod. Pro správu této fronty žádostí je použita http knihovna Volley, díky které je možné posílat žádosti metodou *POST* a získat tak data z databáze.

### 4.5.1 PHP skripty

PHP skripty jsou velmi podobné těm, které jsou již popsány v implementaci webové aplikace. Je zde proto zobrazena jen ukázka pro výpis uživatele a výsledek tohoto skriptu, který se posílá zpět aplikaci.

```
public function vypsatiUzivatele($id) {
    $sql = $this->con->prepare("SELECT * FROM uzivatele WHERE id = ?");
    $sql->bind_param("i", $id);
    $sql->execute();
    return $sql->get_result()->fetch_assoc();
}
```

#### Zdrojový kód 22: Funkce pro výpis uživatele

Tato funkce provede SQL dotaz, který vybere daného uživatele podle identifikačního čísla, které je zadáno při volání této funkce. Funkce vrací celý výsledek tohoto dotazu.



```

$odpoved = array();
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $db = new operace();
    $uzivatel = $db->vypsatUzivatele($_POST['id']);
    $odpoved['jmeno'] = $uzivatel['jmeno'];
    $odpoved['prijmeni'] = $uzivatel['prijmeni'];
    $odpoved['email'] = $uzivatel['email'];
    $odpoved['login'] = $uzivatel['login'];
    $odpoved['telefon'] = $uzivatel['telefon'];
    $odpoved['adresa'] = $uzivatel['adresa'];
    $odpoved['mesto'] = $uzivatel['mesto'];
    $odpoved['psc'] = $uzivatel['psc'];
}
echo json_encode($odpoved);

```

**Zdrojový kód 23: Uložení výsledku dotazu do JSON**

Nejprve se deklaruje proměnná, která obsahuje pole, které se na závěr vypíše ve formátu JSON. Android aplikace je schopna posílat žádosti s metodou typu *POST*. Následně se vytvoří instance, která obsahuje předchozí funkci. Této funkci je posláno identifikační číslo, které bylo obdrženo od aplikace. Dále se přiřadí potřebné proměnné. Na závěr se odpověď přeloží do JSON formátu a vypíše se.

```

{"jmeno":"Lucie","prijmeni":"Novakova",
"email":"lnovakova@gmail.com","login":"novakova",
"telefon":"723159756","adresa":"Starodvorska 33",
"mesto":"Praha","psc":"165 00"}

```

**Zdrojový kód 24: Výsledek funkce výpis uživatele**

Výsledkem předchozí funkce je tedy ukázaný text ve formátu JSON. Tato data byla získána pomocí programu Postman. Tímto programem bylo, pomocí *POST* metody, posláno identifikační číslo pátého zaregistrovaného uživatele.

#### 4.5.2 Ukázka XML

V následující ukázce je zobrazeno XML textového pole pro zadání názvu náradí na titulní stránce pro vyhledávání. Pro textová pole, do kterých může uživatel zadávat informace, se používá element `EditText`. Pro výpis informací na obrazovku se používá

element TextView. Tyto dva elementy společně s tlačítkem (Button) jsou velmi podobné a využívají se velmi často. Proto je zde ukázáno pouze jedno textové pole.

```
<EditText
    android:id="@+id/textNazev"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:ems="10"
    android:hint="Název náradí"
    android:inputType="text"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView" />
```

#### Zdrojový kód 25: XML textového pole

Atributy, které se nejvíce mění, jsou id, hint, inputType a poslední atribut. Do id se zapisuje název pole, pomocí kterého se k tomuto poli přistupuje při programování. Hint znamená nápovědu, značí tedy uživateli, co má do daného políčka napsat. InputType se mění podle toho, co uživatel do daného pole zadává. Může se jednat například o čísla nebo písmena. Poslední atribut značí id předchozího elementu, který má toto pole následovat.

### 4.5.3 Třída pro správu přihlašování

Jelikož v programovacím jazyce Java neexistuje přímo session, je vytvořena třída, která využívá podobnou funkci pro sdílení dat v celé aplikaci a obsahuje základní funkce pro správu přihlašování. Třída obsahuje funkce pro přihlášení uživatele, zjištění, zda je uživatel přihlášen, odhlášení, získání identifikačního čísla přihlášeného uživatele a podobnou funkci pro získání uživatelského jména přihlášeného uživatele.

```
public boolean prihlaseni(String id, String login){
    SharedPreferences sharedPreferences =
        kontext.getSharedPreferences(nazevSpravy, Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString(idUzivatele, id);
    editor.putString(LoginUzivatele, login);
    editor.apply();
    return true;
}
```

#### Zdrojový kód 26: Funkce přihlášení

Tato funkce přijímá argumenty id a login. SharedPreferences nahrazuje funkci z PHP session. Nejprve se vytvoří nová preference s názvem. Dále se vytvoří editor, pomocí kterého se mohou přidávat data do této preference. Následně se zapíše identifikační číslo a uživatelské jméno přihlašovaného uživatele a aplikují se změny. Pro odhlášení se smaže editor a aplikují se změny.

```
public boolean prihlasen(){
    SharedPreferences sharedPreferences =
        kontext.getSharedPreferences(nazevSpravy, Context.MODE_PRIVATE);
    if(sharedPreferences.getString(LoginUzivatele, null) != null){
        return true;
    }
    return false;
}
```

#### Zdrojový kód 27: Funkce přihlášen

Pro zjištění, zda je uživatel přihlášen se opět vytvoří proměnná pro práci s preferencemi. Dále se zkontroluje pomocí funkce *getString*, zda je v preferenci uložené uživatelské jméno. Pokud ano, funkce vrátí pravdu, v opačném případě nepravdu. Funkce pro získání identifikačního čísla a uživatelského jména pouze vrací tyto hodnoty pomocí funkce *getString*.

#### 4.5.4 Menu

Menu lze zobrazit pomocí tlačítka v pravém horním rohu. Nabídka, která se po stisku tlačítka zobrazí je rozdílná pro přihlášené a nepřihlášené uživatele.

```
public boolean onCreateOptionsMenu(Menu menu) {
    if(SpravaPrihlaseni.getInstance(getApplicationContext()).prihlasen()){
        getMenuInflater().inflate(R.menu.menu2, menu);
        return true;
    } else {
        getMenuInflater().inflate(R.menu.menu, menu);
        return true;
    }
}
```

#### Zdrojový kód 28: Zobrazení menu

Funkce, která vytvoří nabídku, nejprve zkontroluje, zda je uživatel přihlášený. Podle toho zobrazí jedno nebo druhé menu.

```
switch(item.getItemId()){
    case R.id.menuPrihlaseni:
        startActivity(new Intent(this, PrihlaseniActivity.class));
        break;
```

#### Zdrojový kód 29: Výběr položky z nabídky

Po výběru položky z nabídky se spustí další funkce. Tato funkce zjistí, kterou položku uživatel stisknul. Například nepřihlášený uživatel stiskl možnost přihlásit. Následně se spustí daná aktivita.

### 4.5.5 Titulní stránka

Titulní stránka obsahuje formulář pro vyhledávání. Tento formulář obsahuje textové pole a výběrové nabídky pro zvolení kategorie a stavu náradí.

```
textNazev = (EditText) findViewById(R.id.textNazev);
textZaloha = (EditText) findViewById(R.id.textZaloha);
textCena = (EditText) findViewById(R.id.textCena);
spinnerKategorie = (Spinner) findViewById(R.id.spinnerKategorie);
spinnerStav = (Spinner) findViewById(R.id.spinnerStav);
buttonVyhledat = (Button) findViewById(R.id.buttonVyhledat);
buttonVyhledat.setOnClickListener(this);
```

#### Zdrojový kód 30: Definice formuláře

V této části kódu se přiřadí jednotlivá políčka formuláře k proměnným. Tlačítku se nastaví kontrola stisknutí.

```
String[] stavList = new String[]{"Stav", "Nové", "Lehce opotřebené",
    "Velmi opotřebené"};
ArrayAdapter<String> adapterStav = new ArrayAdapter<>(this,
    android.R.layout.simple_spinner_dropdown_item, stavList);
spinnerStav.setAdapter(adapterStav);
```

#### Zdrojový kód 31: Přidání položek do nabídky

Spinner představuje nabídku, ze které si může uživatel zvolit nějakou možnost. Do pole, které obsahuje řetězce, se zadají jednotlivé položky nabídky. Dále se připraví pole, aby jej bylo možné zadat do spinneru. Následně se nastaví hodnoty pole přímo do nabídky. Stejným způsobem se přidávají možnosti i do nabídky kategorií.

```

public void onClick(View v) {
    Intent intent = new Intent(getApplicationContext(), VyhledavaniActivity.class);
    intent.putExtra("nazev", textNazev.getText().toString().trim());
    intent.putExtra("zaloha", textZaloha.getText().toString().trim());
    intent.putExtra("cena", textCena.getText().toString().trim());
    intent.putExtra("kategorie",
        spinnerKategorie.getSelectedItem().toString().trim());
    intent.putExtra("stav", spinnerStav.getSelectedItem().toString().trim());
    startActivity(intent);
}

```

**Zdrojový kód 32: Předání dat pro vyhledávání**

Po stisknutí tlačítka vyhledat se deklaruje nová aktivita, do které se posílají i data z formuláře. Z textových polí se získá text, který uživatel zadal. Z nabídek se nejdříve získá položka, kterou uživatel vybral a konvertuje se na řetězec. Následně se spustí tato aktivita se zadanými daty.

#### 4.5.6 Přihlášení

Stránka s přihlášením obsahuje jednoduchý formulář, do kterého zadává uživatel uživatelské jméno a heslo. Nejprve se přiřadí všechna políčka proměnným, podobně jako na titulní stránce. Tentokrát byl přidán ProgressDialog, který oznamuje uživateli, co právě systém dělá. Po stisku tlačítka *přihlásit* se spustí funkce *přihlásit*.

```

final String login = textLogin.getText().toString().trim();
final String heslo = textHeslo.getText().toString().trim();
progressDialog.show();
StringRequest stringRequest = new StringRequest(
    Request.Method.POST,
    Soubory.urlPrihlaseni,
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            progressDialog.dismiss();
        }
    }
);

```

**Zdrojový kód 33: Začátek funkce přihlásit**

Nejprve se deklarují proměnné, do kterých se nastaví text z vyplněných políček. Dále se zobrazí dialog, který oznámí uživateli, že probíhá přihlašování. Následně se deklaruje nová žádost s danou metodou a url adresou, která je deklarovaná ve třídě soubory. Tato žádost bude na závěr předána do fronty. Dále se přepíše funkce, která reaguje na odpověď od PHP skriptu.

```

try {
    JSONObject obj = new JSONObject(response);
    if(!obj.getBoolean("chyba")){
        SpravaPrihlaseni.getInstance(getApplicationContext())
            .prihlaseni(
                obj.getString("id"),
                obj.getString("login")
            );
        Toast.makeText(getApplicationContext(), obj.getString("zprava"),
            Toast.LENGTH_LONG).show();
        startActivity(new Intent(getApplicationContext(),
            ProfilActivity.class));
        finish();
    }else{
        Toast.makeText(getApplicationContext(), obj.getString("zprava"),
            Toast.LENGTH_LONG).show();
    }
} catch (JSONException e) {
    e.printStackTrace();
}

```

#### Zdrojový kód 34: Reakce na odpověď skriptu

Nejprve se deklaruje objekt, který obsahuje odpověď ve formátu JSON. Pokud přihlášení proběhlo úspěšně, proběhne přihlášení uživatele, vypíše se zpráva a uživatel je přesměrován na aktivitu profilu. Pokud ale PHP skript pošle nějakou chybu, vypíše pouze příčinu této chyby. Pokud nastane nějaká chyba při komunikaci se serverem, vypíše se chyba z knihovny Volley.

```

protected Map<String, String> getParams() throws AuthFailureError {
    Map<String, String> params = new HashMap<>();
    params.put("login", login);
    params.put("heslo", heslo);
    return params;
}

```

#### Zdrojový kód 35: Přiřazení proměnných k parametrům

Tato funkce přiřadí proměnné, které obsahují hodnoty z textových polí, k parametrům, které se posílají PHP skriptu. Na závěr už se jen celá žádost přidá do fronty žádostí.

### 4.5.7 Výpis vyhledávání

Výpis vyhledávání funguje podobně jako přihlašování. Nejprve se uloží data poslaná z titulní stránky do proměnných pomocí funkce *getIntent* a *getStringExtra*.

Následně se opět vytvoří žádost. V této žádosti se ale reakce na odpověď velmi liší, protože se vypisuje pole vyhledaných náradí.

```
JSONArray array = new JSONArray(response);
for(int i = 0; i < array.length(); i++){
    JSONObject obj = array.getJSONObject(i);
    TableRow radek = new TableRow(getApplicationContext());
    TableRow.LayoutParams parametry = new
        TableRow.LayoutParams(TableRow.LayoutParams.WRAP_CONTENT);
    radek.setLayoutParams(parametry);
    TextView vse = new TextView(getApplicationContext());
    TextView index = new TextView(getApplicationContext());
    index.setText(obj.getString("id"));
    index.setVisibility(View.INVISIBLE);
    ImageView obrazek = new ImageView(getApplicationContext());
```

#### Zdrojový kód 36: Pole v odpovědi skriptu

Odpověď PHP skriptu je pole ve formátu JSON. Z tohoto důvodu je místo objektu deklarováno pole, do kterého je uložena tato odpověď. Následně je vytvořen cyklus, který projíždí celé pole a končí na posledním záznamu. V každém cyklu se vypíše řádek tabulky. Každý řádek obsahuje text, který obsahuje základní informace o náradí, obrázek náradí a neviditelný text s identifikačním číslem, které je nutné pro pozdější zobrazení detailu o náradí.

```
new VyhledavaniActivity.DownloadImageFromInternet(obrazek)
    .execute("http://10.0.1.29/Naradi/uploads/"+obj.getString("obrazek"));
vse.setText("Název: "+obj.getString("nazev)+"\nZáloha: "
    +obj.getString("zaloha")+"\nCena na den: "
    +obj.getString("cena")+"\nKategorie: "
    +obj.getString("kategorie")+"\nStav: "
    +obj.getString("stav")+"\n");
vse.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(),
            DetailActivity.class);
        intent.putExtra("IdNaradi",
            index.getText().toString().trim());
        startActivity(intent);
    }
});
radek.addView(obrazek);
radek.addView(vse);
displayLinear.addView(radek,i);
```

#### Zdrojový kód 37: Výpis výsledků vyhledávání

Pro zobrazení obrázku byla použita funkce, která stáhne obrázek z url adresy na internetu a zobrazí jej. Url adresa se skládá s přesné adresy serveru, složky a názvu obrázku s příponou. Dále jsou vypsány základní informace o náradí. Na celý text je možné kliknout. Po stisknutí je uživatel přesměrován na aktivitu s detailem vybraného náradí. Této aktivitě je odesláno identifikační číslo vybraného náradí. Dále je vše vypsáno na řádek a pokračuje další cyklus. Následně už stránka obsahuje jen chybu od Volley a přiřazení proměnných k parametrům podobně jako na stránce s přihlášením.

#### **4.5.8 Ostatní stránky**

Ostatní stránky pracují na velmi podobných principech, které byly již představeny. Každá stránka nejprve připojí jednotlivá políčka na stránce k deklarovaným proměnným. Dále jsou deklarovány proměnné, které obsahují data z políček, které zadal uživatel, nebo se posílají z předchozí stránky. Následně se obdobně vytvoří žádost. Odpovědi na tyto žádosti se na různých stránkách liší v políčkách, do kterých se zapisuje a daty, které jsou do políček zapisována. Pokud se pouze zapisuje do databáze, a vše se zapsalo správně, spustí se nová aktivita. Stránka obsahuje také funkci pro výpis chyby komunikace se serverem pomocí knihovny Volley a přiřazení proměnných potřebným parametrům.

Některé stránky obsahují více žádostí. Mezi tyto stránky patří například profil uživatele. Na profilu jsou nejprve vypsány informace o uživateli a druhá žádost obsahuje výpis vlastněného náradí. Další takovou stránkou je stránka s detailem náradí. Na této stránce jsou nejprve vypsány informace o náradí a následně výpůjčky tohoto náradí. V poslední řadě mezi tyto stránky patří úpravy profilu a náradí. Do první žádosti patří předvyplnění jednotlivých políček daty z databáze. Druhá žádost obsahuje uložení upravených informací o uživateli či náradí.



## 5 Výsledky a diskuse

### 5.1 Testování aplikací

Obě aplikace byly v průběhu implementace testovány autorem práce. Většina chyb byla při implementaci odstraněna. Po dokončení implementace obou aplikací, bylo provedeno testování třemi subjekty. Tyto subjekty dostaly úkoly, které prověří funkčnost celých aplikací. Jelikož se mají aplikace zastupovat, dostali uživatelé stejné úkoly pro webovou aplikaci i Android aplikaci. Těmito úkoly byla registrace a přihlášení do aplikace, přidání náradí na vypůjčení, úprava profilu a údajů o vystaveném náradí. Následně měly testované subjekty za úkol vyhledat požadované náradí a vypůjčit si jej. Hlavním cílem tohoto testování bylo odhalení chyb. Dalším cílem tohoto testování bylo zjistit, zda jsou všechny prvky uživatelského rozhraní logicky rozmístěné. Testovací subjekty byly v průběhu testování pod dohledem autora a po dokončení testování byl proveden rozhovor s každým testovacím subjektem, kde testovací subjekty sdělily, co se jim líbilo nebo naopak nelíbilo, co jim nešlo a další důležité připomínky. Všechny tyto chyby a připomínky jsou též sepsány v návrzích pro další vylepšení.

Všechny testovací subjekty zvládly dané úkoly celkem bez problému. Všechny testovací subjekty našly všechny funkční prvky ve webové aplikaci velmi rychle, protože byly na místě, na které jsou zvyklí. Android aplikace měla podobný výsledek, jeden testovací subjekt nebyl tolik obeznámený s mobilními aplikacemi a našel možnosti menu až po malé chvíli. Při rozhovoru s testovacími subjekty byly podány připomínky ke vzhledu webové aplikace. Byl doporučený modernější design. Další připomínkou byly již vytvořené výpůjčky, testovací subjekty by uvítaly další možnost při vyhledávání a to především zadání, zda se má ve výsledcích vyhledávání vypisovat aktuálně vypůjčené, a tudíž aktuálně nedostupné, náradí. Dále byla navržena možnost pro zobrazení všech výpůjček uživatele, nejen výpůjček konkrétního náradí.

## 5.2 Návrhy na další vylepšení

Všechny aplikace se mohou dále vyvíjet a vylepšovat. Proto jsou zde sepsány návrhy na další vylepšení naimplementovaných aplikací. Některé návrhy pochází od testovaných subjektů po provedeném testování a další byly přidány od autora práce.

Mezi návrhy, které byly poznamenány po testování, patří vylepšení vyhledávání. Testovací subjekty by uvítaly možnost eliminování aktuálně vypůjčených náradí ve výsledcích vyhledávání. Tuto funkci by bylo možné vytvořit pomocí zaškrtačacího tlačítka před vyhledáváním. Pokud by bylo toto tlačítko zaškrtnuto, zkontrolovaly by se výpůjčky náradí a pokud by bylo v dobu vyhledávání vypůjčeno, nebylo by do výsledků vyhledávání započítáno. Tento problém by mohl být vyřešen i přidáním políčka pro datum do vyhledávání. Do tohoto políčka by poté uživatel zadal, od jakého data požaduje vypůjčení náradí.

Dalším návrhem od testovacích subjektů byla možnost zobrazení všech výpůjček uživatele. V současné době podporuje aplikace pouze zobrazení výpůjček na stránce vybraného náradí. Mohly by se tedy všechny výpůjčky uživatele vypisovat v profilu uživatele.

Hlavním návrhem na další vylepšení od autora je prémiový účet. Tato funkce by pomohla s financováním aplikací. Uživatelé, kteří by se chtěli více zviditelnit, by si mohli zaplatit prémiový účet a tím zviditelnit svá vystavovaná náradí na vypůjčení. Dalším návrhem je pomocí Google Maps vypočítávat vzdálenost od uživatele, který si chce náradí vypůjčit, k danému náradí. Díky této funkci by se mohlo snadno vyhledávat podle vzdálenosti. Dalším návrhem je podpora online plateb, díky kterým mohou aplikace lépe kontrolovat platby a případně strhávat nějakou malou částku za využití této služby. V poslední řadě je doporučeno zkrácení, vylepšení a formátování zdrojového kódu a také vylepšení designové stránky aplikací.

## 6 Závěr

V teoretické části byly popsány všechny použité technologie. Tato kapitola byla rozdělena do tří částí. V první části byly popsány technologie, které zahrnovaly webové aplikace. Nejprve byl popsán rozdíl mezi statickými a dynamickými stránkami. Dále byly popsány technologie pro frontendovou část webové aplikace, které zahrnovaly značkovací jazyk HTML a kaskádové styly CSS. Mezi backendové technologie patří skriptovací jazyk PHP, webový server Apache a databázový systém MariaDB společně s dotazovacím jazykem SQL. V druhé části byly popsány technologie pro Android aplikaci. V této části byl nejprve popsán programovací jazyk Java a objektově orientované programování. Následně byly popsány nástroje XML a JSON a na závěr této části http knihovna Volley. V poslední části byly představeny všechny použité nástroje a programy. Nejprve byl představen balíček nástrojů XAMPP, který obsahuje webový server Apache, databázový systém MariaDB a nástroj PhpMyAdmin. Dále byl pro zápis PHP skriptů použit nástroj NetBeans. Následně bylo představeno vývojové prostředí pro vývoj Android aplikací, Android Studio a vestavený emulátor zařízení s operačním systémem Android. V poslední řadě byl představen nástroj Postman, který byl použit pro testování výstupů.

V praktické části byly všechny použité technologie využity. Byl zde popsán celý proces vývoje aplikací. V první části byla navržena a vytvořena databáze. Byly zde popsány všechny vytvořené tabulky, jejich vztahy a atributy. V druhé části byla navržena webová aplikace. Byly zde popsány stránky, vytvořeny případy použití a scénáře a pomocí drátového modelu, neboli wireframe, bylo ukázáno rozmístění všech prvků na stránkách. V další části byl popsán celý proces implementace webové aplikace. Celý tento proces je zobrazen pomocí ukázek zdrojového kódu a popisu těchto ukázek. V další části byla navržena Android aplikace. Jelikož se tyto aplikace mají zastupovat a jsou tedy velmi podobné, jsou zde pouze popsány jednotlivé obrazovky a vytvořeny drátové modely. Následně byla provedena implementace obdobně jako u webové aplikace. Na závěr bylo provedeno testování. Proces testování a následné návrhy na další vylepšení aplikací jsou popsány ve výsledcích a diskusi.

## 7 Seznam použitých zdrojů

1. The difference between dynamic and static websites. H&C [online]. [cit. 2021-03-21]. Dostupné z: <https://www.hughesandco.ca/blog/the-difference-between-dynamic-and-static-websites>
2. Frontend. Shoptet [online]. [cit. 2021-03-21]. Dostupné z: <https://www.shoptet.cz/slovník-pojmu/frontend/>
3. Značkovací jazyk (HTML). VeV-Va [online]. [cit. 2021-03-21]. Dostupné z: <https://web.vavyskov.cz/znackovaci-jazyk.html>
4. What is CSS? Tutorialspoint [online]. [cit. 2021-03-21]. Dostupné z: [https://www.tutorialspoint.com/css/what\\_is\\_css.htm](https://www.tutorialspoint.com/css/what_is_css.htm)
5. Backend. TechTerms [online]. [cit. 2021-03-21]. Dostupné z: <https://techterms.com/definition/backend>
6. What is a scripting language? Career karma [online]. [cit. 2021-03-21]. Dostupné z: <https://careerkarma.com/blog/what-is-a-scripting-language/>
7. What is ASP.NET? Microsoft [online]. [cit. 2021-03-21]. Dostupné z: <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet>
8. PHP. TechTerms [online]. [cit. 2021-03-21]. Dostupné z: <https://techterms.com/definition/php>
9. PHP releases. PHP [online]. [cit. 2021-03-21]. Dostupné z: <https://www.php.net/>
10. What is MariaDB? TecMint [online]. [cit. 2021-03-21]. Dostupné z: <https://www.tecmint.com/what-is-mariadb-how-does-mariadb-work/>
11. SQL - Overview. Tutorialspoint [online]. [cit. 2021-03-21]. Dostupné z: <https://www.tutorialspoint.com/sql/sql-overview.htm>
12. GILMORE, W J. – POKORNÝ, J. Velká kniha PHP 5 a MySQL : kompendium znalostí pro začátečníky i profesionály. Brno: Zoner Press, 2011. ISBN 978-80-7413-163-9.
13. What is Apache?. Hostinger Tutorials [online]. [cit. 2021-03-21]. Dostupné z: <https://www.hostinger.com/tutorials/what-is-apache>
14. Android. TechTerms [online]. [cit. 2021-03-21]. Dostupné z: <https://techterms.com/definition/android>
15. Android 11. Android [online]. [cit. 2021-03-21]. Dostupné z: <https://www.android.com/android-11/>
16. Android - Overview. Tutorialspoint [online]. [cit. 2021-03-21]. Dostupné z: [https://www.tutorialspoint.com/android/android\\_overview.htm](https://www.tutorialspoint.com/android/android_overview.htm)
17. LACKO, Ľ. Vývoj aplikací pro Android. Brno: Computer Press, 2015. ISBN 978-80-251-4347-6.
18. What is Java and why is it important? Code institute [online]. [cit. 2021-03-21]. Dostupné z: <https://codeinstitute.net/blog/what-is-java/>
19. Java. TechTerms [online]. [cit. 2021-03-21]. Dostupné z: <https://techterms.com/definition/java>

20. OOP. TechTerms [online]. [cit. 2021-03-21]. Dostupné z:  
<https://techterms.com/definition/oop>
21. XML - Overview. TutorialsPoint [online]. [cit. 2021-03-21]. Dostupné z:  
[https://www.tutorialspoint.com/xml/xml\\_overview.htm](https://www.tutorialspoint.com/xml/xml_overview.htm)
22. XML. TechTerms [online]. [cit. 2021-03-21]. Dostupné z:  
<https://techterms.com/definition/xml>
23. What is JSON?. InfoWorld [online]. [cit. 2021-03-21]. Dostupné z:  
<https://www.infoworld.com/article/3222851/what-is-json-a-better-format-for-data-exchange.html>
24. Introducing JSON. JSON [online]. [cit. 2021-03-21]. Dostupné z:  
<https://www.json.org/json-en.html>
25. Volley library in Android. GeeksforGeeks [online]. [cit. 2021-03-21]. Dostupné z:  
<https://www.geeksforgeeks.org/volley-library-in-android/>
26. XAMPP tutorial. JavaTpoint [online]. [cit. 2021-03-21]. Dostupné z:  
<https://www.javatpoint.com/xampp>
27. About. PhpMyAdmin [online]. [cit. 2021-03-21]. Dostupné z:  
<https://www.phpmyadmin.net/>
28. What is NetBeans? – for beginner. CodeJava [online]. [cit. 2021-03-21]. Dostupné z:  
<https://www.codejava.net/ides/netbeans/what-is-netbeans-for-beginner>
29. NetBeans. Techopedia [online]. [cit. 2021-03-21]. Dostupné z:  
<https://www.techopedia.com/definition/24735/netbeans>
30. Android Studio. SearchMobileComputing [online]. [cit. 2021-03-21]. Dostupné z:  
<https://searchmobilecomputing.techtarget.com/definition/Android-Studio>
31. Features overview. Android Developers [online]. [cit. 2021-03-21]. Dostupné z:  
<https://developer.android.com/studio/features>
32. Run apps on the Android Emulator. Android Developers [online]. [cit. 2021-03-21].  
Dostupné z: <https://developer.android.com/studio/run/emulator>
33. Introduction to Postman for API development. GeeksforGeeks [online]. [cit. 2021-03-21]. Dostupné z: <https://www.geeksforgeeks.org/introduction-postman-api-development/>

## **8 Přílohy**

V přiloženém zip souboru na CD jsou dostupné exportované soubory csv a sql z nástroje PhpMyAdmin, NetBeans projekty, které obsahují zdrojový kód webové aplikace a část pro komunikaci s databází Android aplikace a Android Studio projekt se zdrojovým kódem Android aplikace.