



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

WEBOVÉ ROZHRANÍ REPOZITÁŘE DEBIAN

DEBIAN REPOSITORY WEB INTERFACE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Marek Murgaš

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Petr Sysel, Ph.D.

BRNO 2023

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Marek Murgaš

ID: 221563

Ročník: 3

Akademický rok: 2022/23

NÁZEV TÉMATU:

Webové rozhraní repozitáře Debian

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se programovacím jazykem PHP a knihovnamy Nette pro tvorbu webových aplikací. Na základě získaných zkušeností navrhnete a realizujete webové rozhraní repozitáře systému Debian. Vytvořený kód bude řešen modulárně, aby ho bylo možné začlenit do dalších aplikací. V rámci semestrální práce vytvoříte základ aplikace umožňující práci s balíčky (získání informací z balíčku, vytvoření seznamu balíčku Release, atd.). V rámci navazující diplomové práce aplikaci rozšíříte tak, aby podporovala autorizaci heslem nebo certifikátem, podpis seznamu balíčku nebo jednotlivých balíčku. Dále zpřehledněte uživatelské rozhraní.

DOPORUČENÁ LITERATURA:

- [1] SKLAR, David a Jan POKORNÝ. PHP 7: praktický průvodce nejrozšířenějším skriptovacím jazykem pro web. Brno: Zoner press, 2018, 368 s. ISBN 978-80-7413-363-3
- [2] STEPHENS, Ryan K., Ronald R. PLEW a Arie JONES. Naučte se SQL za 28 dní. Brno: Computer Press, 2010, 728 s. ISBN 978-80-251-2700-1.

Termín zadání: 6.2.2023

Termín odevzdání: 26.5.2023

Vedoucí práce: doc. Ing. Petr Sysel, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Bakalárska práca sa zaoberá implementáciou webového rozhrania pre repozitár systému Debian. Cieľom mojej práce bolo vytvorenie aplikácie, ktorá bude schopná získavať informácie o balíčkoch a generovať príslušné indexové súbory za účelom simulovania štruktúry repozitára systému Debian. Práca sa v teoretickej časti zaoberá bližším predstavením technológií, ktoré boli použité pri implementácii webového rozhrania. Ďalej je tu opísaná samotná štruktúra repozitára Debian. Súčasťou tejto časti je aj opis štruktúry balíčkov Debian a informácií, ktoré sú potrebné na vygenerovanie zoznamov balíčkov. Nachádza sa tu aj predstavenie spôsobov akými je možné podpisovať balíčky alebo vygenerované zoznamy. Praktická časť sa venuje implementácii webového rozhrania a predstaveniu užívateľského rozhrania a funkcionalít, ktoré ponúka.

Kľúčové slová

Debian; PHP; Nette; Latte; HTML; SQLite; Bootstrap; webové rozhranie; repozitár systému Debian; softwarové balíčky

Abstract

The theme of this bachelor's thesis is implementation of web interface for Debian repository. The main goal of this bachelor's thesis was to create application, which will be able to get information from compressed debian packages and generate corresponding index files for the purpose of simulating Debian repository structure. The theoretical part of this thesis concentrates on introduction of technologies used for implementation of web interface. Further, it deals with description of Debian repository structure. Theoretical part also includes description of Debian packages structure and information needed for generating index files. Additionally, there is also introduction of possible ways for signing Debian packages. Practical part of this thesis consists of description of web interface implementation and presentation of user interface.

Keywords

Debian; PHP; Nette; Latte; HTML; SQLite; Bootstrap; web interface; Debian repository; software packages

Bibliografická citácia

MURGAŠ, Marek. *Webové rozhraní repozitáře Debian*. Brno, 2023. Dostupné také z: <https://www.vut.cz/studenti/zav-prace/detail/151203>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Petr Sysel.

Vyhlásenie autora o pôvodnosti diela

Meno a priezvisko študenta:	Marek Murgaš
VUT ID študenta:	221563
Typ práce:	Bakalárska práca
Akademický rok:	2022/23
Téma záverečnej práce:	Webové rozhranie repozitára Debian

Vyhlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedúceho záverečnej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia S 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

V Brne dňa:

podpis autora

Pod'akovanie

Ďakujem vedúcemu práce doc. Ing. Petrovi Syslovi, Ph.D. za odborné vedenie a cenné pripomienky, ktoré mi dopomohli k vypracovaniu práce.

V Brne dňa:

podpis autora

Obsah

1. POUŽITÉ TECHNOLOGIE.....	11
1.1 PROGRAMOVACÍ JAZYK PHP.....	11
1.2 ARCHITEKTÚRA MVC.....	11
1.2.1 <i>Presenter</i>	11
1.2.2 <i>Model</i>	13
1.2.3 <i>Šablóna</i>	13
1.3 NETTE FRAMEWORK.....	14
1.3.1 <i>Výhody Nette Frameworku</i>	14
1.3.2 <i>Nevýhody Nette Frameworku</i>	15
1.4 LATTE.....	15
1.5 DATABÁZA SQLITE.....	15
1.6 BOOTSTRAP.....	16
2. REPOZITÁR SYSTÉMU DEBIAN	17
2.1 DEBIAN BALÍČKY	17
2.1.1 <i>Štruktúra balíčku DEB</i>	17
2.2 ŠTRUKTÚRA REPOZITÁRA.....	21
2.3 VYTVORENIE REPOZITÁRA	23
2.4 PODPISOVANIE BALÍČKOV	23
3. IMPLEMENTÁCIA WEBOVÉHO ROZHRAŇIA.....	26
3.1 ŠTRUKTÚRA DATABÁZY	26
3.2 ŠTRUKTÚRA APLIKÁCIE.....	28
3.2.1 <i>DebianModule</i>	29
3.2.2 <i>Modely</i>	29
3.2.3 <i>Presentery</i>	31
3.2.4 <i>Šablóny</i>	32
3.3 AUTENTIZÁCIA A AUTORIZÁCIA	35
3.4 UŽÍVATELSKÉ ROZHRAŇIE	37
3.4.1 <i>Prihlásenie do užívateľského rozhrania</i>	37
3.4.2 <i>Administrátorský panel</i>	38
3.4.3 <i>Užívateľský panel</i>	39
3.4.4 <i>Návod na vloženie balíčka do repozitára</i>	40
3.4.5 <i>Návod na vytvorenie a úpravu distribúcie</i>	41
3.4.6 <i>Navigácia v repozitári</i>	43
4. ZÁVER.....	50

ZOZNAM OBRÁZKOV

Obrázok 1 : Životný cyklus presentera [6].....	12
Obrázok 2 : Obsah súboru Control	18
Obrázok 3 : Príklad štruktúry repozitára Debian	21
Obrázok 4 : Obsah súboru _gpgbuilder	24
Obrázok 5 : Výstup príkazu dpk-sig -verify.....	25
Obrázok 6 : Štruktúra databáze.....	27
Obrázok 7 : Štruktúra aplikácie	28
Obrázok 8 : Štruktúra adresára templates v DebianModule	33
Obrázok 9 : Štruktúra adresára templates	33
Obrázok 10 : Prihlasovací formulár.....	37
Obrázok 11 : Admin Panel.....	38
Obrázok 12 : Možné chybové hlásenia pri úprave tabuľky Manage Users.....	39
Obrázok 13 : Users Panel.....	40
Obrázok 14 : Obsah stránky Packages.....	41
Obrázok 15 : Chybové hlásenie pri vložení súboru v nesprávnom formáte.....	41
Obrázok 16 : Obsah stránky Distributions	42
Obrázok 17 : Vkladanie balíčkov do distribúcie s názvom TestDistribution.....	42
Obrázok 18 : Domovská stránka repozitára Debian	43
Obrázok 19 : Obsah adresára Debian.....	43
Obrázok 20 : Obsah adresára dists	44
Obrázok 21 : Obsah podadresára s názvom distribúcie	44
Obrázok 22 : Obsah súboru Release v podadresári distribúcie	45
Obrázok 23 : Obsah súboru Release.gpg	45
Obrázok 24 : Obsah podadresára komponenty main	46
Obrázok 25 : Obsah podadresára architektúry amd64	46
Obrázok 26 : Obsah súboru Packages v podadresári architektúry amd64	47
Obrázok 27 : Obsah súboru Release v podadresári architektúry amd64.....	47
Obrázok 28 : Obsah podadresára main	48
Obrázok 29 : Obsah podadresára so začiatočným písmenom distribúcie	48
Obrázok 30 : Obsah podadresára s názvom softwarového balíčku.....	49

ZOZNAM TABULIEK

Tabulka 1 : Význam tabuliek.....	26
Tabulka 2 : Autorizačné role a ich význam	36

ZOZNAM VÝPISOV

Výpis 1: Ukážka kódu funkcie slúžiacej na rozbalenie balíčka.....	31
Výpis 2: Vykresľovanie distribúcií v rozbalovacej ponuke šablóny distSelect.latte.....	34
Výpis 3: Ukážka kódu funkcie overujúcej platnosť certifikátu.....	35
Výpis 4: Autorizácia užívateľa pomocou metódy isInRole().....	36

ÚVOD

Bežný spôsob distribúcie softwaru v Linuxe je pomocou balíčkov vo formáte rpm alebo deb. Väčšina balíčkov je súčasťou oficiálnych distribúcií repozitárov alebo softwarových repozitárov tretích strán. Existujú však prípady kedy chceme nainštalovať len zopár samostatných balíčkov. Je možné použiť lokálne nástroje na inštaláciu balíkov ale môže sa stať, že balíček nebude možné nainštalovať kvôli závislostiam a vám neostáva nič iné ako nainštalovať všetky závislosti manuálne. Tento proces však môže byť časovo náročný. Riešením je vytvorenie vlastného repozitára, do ktorého budú vkladané balíčky.

Cieľom tejto bakalárskej práce je vytvoriť webové rozhranie repozitára systému Debian s využitím frameworku Nette. Webové rozhranie by malo užívateľovi poskytnúť možnosť nahráť balíček do distribúcie, získať informácie o balíčkoch a vygenerovať príslušné indexové súbory. Aplikácia by mala podporovať autentizáciu užívateľov pomocou hesla alebo platného certifikátu. Taktiež by mala umožňovať podpisovanie jednotlivých balíčkov alebo vygenerovaných zoznamov.

Práca sa v prvej kapitole zaoberá bližším predstavením technológií, ktoré boli použité pri implementácii webového rozhrania. Druhá kapitola sa sústreďuje na priblíženie štruktúry repozitára Debian a procesu podpisovania balíčkov Debian. Súčasťou tejto kapitoly je opis štruktúry balíčkov Debian a informácií, ktoré sú potrebné na vygenerovanie zoznamov balíčkov. Ďalej sú tu predstavené nástroje slúžiace na vytvorenie a údržbu repozitára. Tretia kapitola je venovaná samotnej implementácii webového rozhrania. Je tu predstavená štruktúra aplikácie, spôsob autentizácie a autorizácie užívateľov a návody na využitie funkcionalít, ktoré ponúka užívateľské rozhranie.

1. POUŽITÉ TECHNOLOGIE

1.1 Programovací jazyk PHP

PHP (PHP: Hypertext Preprocessor) je populárny open source skriptovací jazyk, ktorý podporuje objektové aj procedurálne programovanie. PHP slúži na vytváranie dynamických webových stránok na strane servera alebo nezávislých aplikácií s grafickým rozhraním. Prebieha to tak, že klient zadá požiadavku a zo serveru dostane odpoveď vo forme HTML. [1] Dôležitou vlastnosťou PHP je nezávislosť na platforme. To znamená, že rozdiely v rôznych operačných systémoch nájdeme len v niekoľkých systémovo závislých funkciách, takže vo väčšine prípadov by sme mali byť schopný prenášať skripty medzi operačnými systémami bez toho, aby sme pred tým vykonali akékoľvek úpravy. Medzi hlavné výhody PHP patrí už zmienená kompatibilita naprieč platformami. Ďalším plusom je podpora rôznych štandardných systémov správy databázy ako napríklad MySQL a SQLite. Človeka, ktorý uvažuje nad využitím PHP, môže presvedčiť obrovská podpora komunity, ktorá sa vytvorila okolo tohto skriptovacieho jazyka a predovšetkým jednoduchá naučiteľnosť jazyka. Okrem toho je PHP pomerne bezpečný, pretože k šifrovaniu užívateľských dát je k dispozícii mnoho hashovacích funkcií. [3]

História PHP siaha do roku 1994 kedy programátor Rasmus Lerdorf navrhol niekoľko skriptov v jazyku Perl, ktoré mu umožňovali zistiť, koľko návštevníkov čítalo obsah jeho webových stránok. Túto sadu skriptov pomenoval Personal Home Page. V roku 1997 vyšlo PHP 2.0 inak nazývané Personal Home Page – Form Interpreter (PHP/FI). Nasledujúca verzia PHP 3.0 predstavila trhu revolučné rozšírenie. Týmto rozšírením bolo prepojenie databázy so schémami obsahujúcimi objekty, teda obdoba súčasného objektovo orientovaného programovania. Hlavne vďaka svojej flexibilitě, popularita PHP za posledné roky drasticky vzrástla a urobila z neho najpoužívanejší programovací jazyk pre tvorbu webových aplikácií.[2] Pri implementácii aplikácie bola použitá verzia PHP 8.1.12.

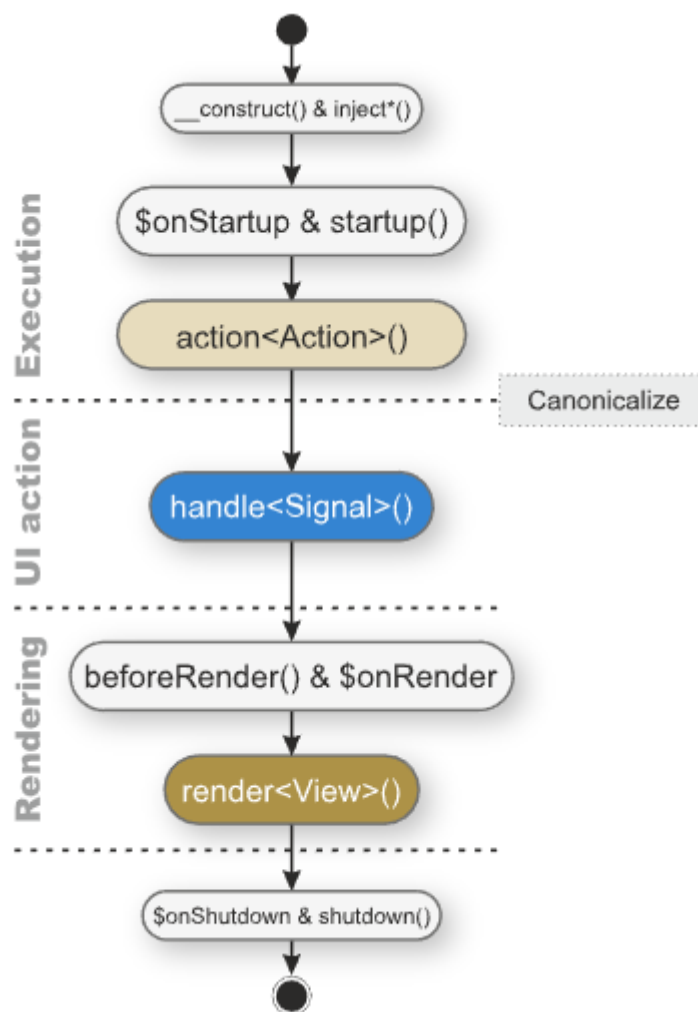
1.2 Architektúra MVC

Nette využíva softwarovú architektúru MVP (Model view presenter), ktorá funguje na rovnakom princípe ako MVC (Model view controler). Aplikácia sa delí na 3 vrstvy, ktoré sa starajú o riadenie, logiku a výstup.

1.2.1 Presenter

Presenter je komponent, ktorý slúži na komunikáciu s užívateľom a v prípade potreby komunikuje s modelom. Presenter prijímané parametre od klienta predáva ďalej modelom, z ktorých získa dáta využívané šablónami. Výsledkom tohto procesu je

HTML stránka. Presenter sa riadi životným cyklom, počas ktorého môžeme ovplyvniť beh aplikácie. Špeciálne metódy, z ktorých sa skladá životný cyklus presentera, sú volané sekvenčne zhora nadol. Existencia týchto metód nie je podmienkou a je možné vytvoriť prázdny presenter, ktorý nebude obsahovať ani jednu z nasledujúcich metód. [6]



Obrázok 1 : Životný cyklus presentera [6]

- **___construct()** – konštruktor nie je úplne súčasťou životného cyklu, pretože je volaný pri vytváraní objektu ale je dôležitou súčasťou predávania závislostí. [6]
- **startup()** – slúži na inicializáciu presenteru. Väčšinou sa tu registrujú služby definované v systémovom DI (Dependency Injection) kontajneri alebo overujú užívateľské oprávnenia. Je vyžadované aby bol v metóde volaný predok `parent::startup()`. [6]

- **action<Action>()** – táto metóda je volaná ešte pred samotným vykreslením šablóny, takže na základe výsledku môžeme ďalej rozhodovať o prípadnom presmerovaní na inú šablónu. [6]
- **handle<Signal>()** – metóda slúži na spracovanie prijatých signálov, ktoré je možné obdržať od rôznych komponentov alebo AJAXových požiadaviek. [6]
- **beforeRender()** – volá sa pred každou metódou `render<View>`. Slúži napríklad na konfiguráciu šablóny alebo predávanie premenných. [6]
- **render<View>()** – využíva sa na prípravu šablóny a jej následné vykreslenie. Väčšinou v nej predávame šablónu dáta získané z modelov. [6]
- **afterRender()** – volá sa za každou metódou `render<View>` a využíva sa len vo výnimočných prípadoch. [6]
- **shutdown()** – táto metóda je vykonaná automaticky na konci životného cyklu a nie je potrebné ju volať explicitne [6]

1.2.2 Model

Model tvorí rozhranie pre prácu s dátami. Predstavuje prvok aplikácie, ktorý je zodpovedný za logiku. Logikou aplikácie sa myslia rôzne výpočty alebo práca s databázou. Jednotlivé modely reprezentujú entity, ktoré by mali byť odrazom reálneho sveta. Akákoľvek akcia užívateľa by mala byť reprezentovaná akciou modelu. [7] Súvisiace modely a presentery sa ďalej združujú do modulov. Moduly predstavujú logické celky, z ktorých sa aplikácia skladá. Je to skupina presenterov, šablón a komponentov určených pre špecifickú časť webu. [22]

1.2.3 Šablóna

Šablóny slúžia na prezentáciu stavu aplikácie koncovému užívateľovi. Pohľad nemá vyvolávať žiadnu zmenu v aplikácii a preto by mal byť oslobodený od logiky aplikácie. Nette využíva šablónovací systém Latte. Šablóna musí byť uložená v adresári, ktorý je pomenovaný rovnako ako presenter. Tento presenter potom volá vykreslenie danej šablóny. Užitočnou vlastnosťou šablón je ich dedičnosť pomocou blokov.

Koncept dedičnosti šablón Latte funguje na podobnom princípe ako dedičnosť tried v PHP. Nadradená šablóna slúži ako rodič, z ktorého dedia ostatné podriadené šablóny. [8]

1.3 Nette Framework

Nette Framework je open source framework vytvorený na tvorbu webových aplikácií v jazyku PHP. Je napísaný v PHP a plne využíva objektovo orientované programovanie. Podporuje moderné technológie a postupy ako napríklad AJAX, DRY, KISS, MVC a znovupoužitelnosť kódu. Vznik frameworku sa datuje do roku 2004 a po štyroch rokoch vývoja bola v roku 2008 vydaná prvá verzia. Pôvodným autorom Nette Frameworku je skúsený český PHP vývojár David Grudl a o jeho budúcnosti a ďalšom rozvoji rozhoduje organizácia Nette Foundation. [4] Jeho licencia vychádza z BSD, ktorá patrí k tým najľahčejším. Nette je vyvíjaný v českom programátorskom prostredí vďaka tomu sa okolo neho vytvorila jedna z najaktívnejších komunít českých PHP vývojárov. Nette tiež používa veľa významných tuzemských spoločností ako napríklad Mladá fronta, Slevomat a mnoho ďalších. Výsledok testu uverejneného na serveri Root ukázal, že Nette je jedným z najvýkonnejších frameworkov. [5]

1.3.1 Výhody Nette Frameworku

- **Silný validačný jazyk** – Nette uľahčuje vývojárom vytváranie formulárov v aplikáciách. Framework sám o sebe poskytuje zabezpečenie proti útokom typu Cross Site Scripting, Cross-Site Request Forgery alebo UTF-8 attack. [4]
- **Routovanie** - Nette umožňuje navrhovať štruktúru URL nezávisle na zvyšku aplikácie. Je to docielené obojstranným routovaním medzi URL a akciou presenteru. Na rozdiel od väčšiny frameworkov je možné nastaviť tvar URL ako poslednú vec v celej aplikácii. [4]
- **Eliminácia a ladenie chýb** – Nette obsahuje nástroj nazývaný „laděnka“ (tracy), ktorý programátorovi prehľadne zobrazí chyby vzniknuté počas kompilácie kódu. Chyby je navyše možné nechať automaticky posielať na email. Jedná sa o nástroj vývojového prostredia a preto je potrebné využiť `NetteLoader` a zapnúť debug mód. [4]

- **Autoloading** – V Nette je možné nechať automaticky nahrávať súbory nezávisle na adresárovej štruktúre s využitím komponentu `RobotLoader`. Tieto súbory sú potom indexované a ukladané do cache. [4]

1.3.2 Nevýhody Nette Frameworku

- **Ručné premazávanie cache** – Framework neponúka automatické vymazávanie cache a preto je nutné občas premazať cache ručne. [4]
- **Dokumentácia** – Nette disponuje dokumentáciou, ktorá nie je konzistentná, neobsahuje všetko potrebné a nepokrýva všetky oblasti. Autor sa však snaží kompenzovať tento nedostatok organizovaním školení a neformálnych stretnutí priaznivcov. [4]

1.4 Latte

Súčasťou frameworku Nette je šablonovací systém Latte, ktorý prekladá šablóny na optimalizovaný kód. Vďaka tomuto prekladu má rovnakú rýchlosť ako PHP kód. Latte je označovaný ako najbezpečnejší šablonovací systém pre PHP. Na obranu pred Cross-Site Scripting (XSS) využíva kontextovo senzitivne escapovanie a kontrolu odkazov. [9] Systém Latte bol využitý na implementáciu šablón v aplikácii.

Medzi výhody šablonovacieho systému Latte patria:

- **intuitívna syntax**
- **možnosť vytvárať bloky**
- **dedičnosť**
- **možnosť použiť makrá**
- **zabezpečenie pred zraniteľnosťami**

1.5 Databáza Sqlite

Na rozdiel od veľkých databáz ako MySQL/MariaDB, PostgreSQL, je vhodnejšie SQLite označiť skôr ako knihovnu nástrojov. Databáza SQLite patrí k druhu takzvanej vloženej databáze. Nejedná sa teda o samostatný program ale funguje ako samostatný proces v rámci spustenej aplikácie. Výhodou je, že sa využíva len počas behu aplikácie a nevyužíva systémové prostriedky, keď to nie je potrebné. [11]

Každá databáza sa ukladá na disk vo forme obyčajného súboru s príponou .db. Pri návrhu SQLite sa vývojári riadili základnou myšlienkou, ktorou bola prenositeľnosť . Pre prenos projektu stačí skopírovať aplikáciu so súborom s databázou a spustiť bez dodatočného inštalovania služby. SQLite je dnes možné používať na systémoch Windows, Linux, BSD, MAC OS X, či na komerčných systémoch Unixu ako sú Solaris, HP-UX a AIX5. Formát súboru databáze je binárne kompatibilný naprieč všetkými podporovanými systémami. Ďalšou z výhod je jednoduchosť, ktorá spočíva v jednoduchej údržbe, správe a absencii akejkoľvek inštalácie či konfigurácie. [10] Databáza SQLite bola využitá pri implementácii modelov.

1.6 Bootstrap

Bootstrap je najpopulárnejší open-source framework slúžiaci na tvorbu responzívnych webov. Využíva návrhárské šablóny založené na HTML, CSS (Cascading Style Sheets) a Java-Scripte, slúžiace na úpravu vzhľadu komponentov rozhrania. Medzi tieto komponenty patria napríklad tlačidlá, navigácia alebo formuláre. [16] Veľkou výhodou Bootstrap frameworku je jeho rozsiahla dokumentácia. To znamená, že kód písaný v súlade s bootstrapom je ľahko pochopiteľný aj pre ostatných užívateľov. Zmyslom frameworku je predovšetkým jeho jednoduchá použiteľnosť. Užívateľ ovplyvňuje len vzhľad danej komponenty a všetko ostatné za neho rieši samotný bootstrap. Pri implementácii aplikácie bola použitá najnovšia verzia frameworku Bootstrap 5. [17]

2. REPOZITÁR SYSTÉMU DEBIAN

2.1 Debian balíčky

Balíčky distribúcie Debian sú vo všeobecnosti komprimované súbory s preddefinovanou vnútornou štruktúrou. Obsahujú všetky súbory potrebné na implementáciu súvisiacich príkazov a vlastností. [12]

Rozlišujeme dva druhy Debian balíčkov:

- **Binary** – balíčky tohto typu obsahujú spustiteľné súbory, konfiguračné súbory, knižnice a iné dáta. Tieto balíčky sú charakteristické príponou `.deb`. Názov súboru `.deb` dodržiava konvenciu `<PackageName>_<VersionNumber>-<DebianRevisionNumber>_<DebianArchitecture>.deb`. V balíčku sa nachádza súbor `control`, ktorý obsahuje metadáta o balíčku. Binárne balíčky je možné rozbaľiť pomocou Debian utility `dpkg`. [12]
- **Source** – balíčky typu source pozostávajú zo súborov s príponami `.dsc` a `.gz`. Súbor `.dsc` obsahuje informácie slúžiace na opis balíčku source. Súbor `.orig.tar.gz` predstavuje nemoďifikovaný balíček source vo formáte `tar`, ktorý podlieha kompresii pomocou metódy `gzip`. Ďalším zo súborov obsiahnutých v balíčku source je `.diff.gz`, ktorý obsahuje zmeny oproti originálnemu balíčku source. Source balíčky je možné rozbaľiť pomocou utility `dpkg-source`. [12]

2.1.1 Štruktúra balíčku DEB

- **debian-binary** – textový súbor obsahujúci číslo verzie formátu balíčku, aktuálna verzia je 2.0
- **data.tar.gz** – `tar` archív, ktorý obsahuje inštalovateľné súbory
- **control.tar.gz** – `tar` archív, ktorý obsahuje meta-informácie o balíčkoch [21]

Archív `Control` môže obsahovať nasledovné súbory [12] [21] :

- **control** – obsahuje základné meta-informácie ako napríklad názov balíčku, verzia, architektúra, udržiavateľ, závislosti a stručný popis balíčka.
- **md5sums** – obsahuje zoznam kontrolných súčtov MD5 všetkých súborov v archíve.

- **conffiles** – obsahuje zoznam súborov, s ktorými by sa malo zaobchádzať ako s konfiguračnými súbormi
- **preinst** – skript, ktorý beží pred inštaláciou balíčka
- **postinst** – skript, ktorý beží po inštalácii balíčka
- **prerm** – skript, ktorý beží pred odstránením balíčka
- **postrm** – skript, ktorý beží po odstránení balíčka
- **config** – voliteľný skript, ktorý podporuje konfiguračný mechanizmus `debconf`
- **shlibs** – obsahuje zoznam závislosti zdieľaných knižníc

Súbor `control` obsahujúci meta-informácie je dôležitou súčasťou archívu a preto by bolo vhodné prestavenie štruktúry tohto súboru a význam jednotlivých parametrov.

```
Package: zlib1g-dbg
Source: zlib
Version: 1:1.2.8.dfsg-5
Architecture: amd64
Maintainer: Mark Brown <broonie@debian.org>
Installed-Size: 210
Depends: zlib1g (= 1:1.2.8.dfsg-5)
Provides: libz-dbg
Section: debug
Priority: extra
Homepage: http://zlib.net/
Description: compression library - development
zlib is a library implementing the deflate compression method found
in gzip and PKZIP. This package includes debugging symbols for use
when debugging applications which use zlib.
Build-Ids: 908b5a955d0a73fb8d31e0f927d0cdba810cb300
```

Obrázok 2 : Obsah súboru Control

Popis parametrov [13] :

- **Package** – názov binárneho balíčku. Syntax názvu binárnych balíčkov podlieha rovnakým pravidlám ako syntax balíčkov `source`.
- **Source** – obsahuje názov balíčka `source`. Názov sa skladá z malých písmen, čísel a znakov (+), (-) a (.). Je odlišný ako samotný názov balíčka.
- **Version** – obsahuje číslo verzie pôvodného balíčka.

- **Architecture**- špecifikuje pre aký typ hardwaru bol balíček skompilovaný. Typickými architektúrami sú napríklad `i386`, `amd64`, `armhf`, `sparc` atd. Špeciálnymi prípadmi sú architektúry `all` a `any`. Architektúra `all` indikuje, že balíček je nezávislý na architektúre a `any` indikuje, že balíček je možné zostrojiť nad akoukoľvek architektúrou.
- **Maintainer** – predstavuje meno a emailovú adresu udržiavateľa balíčku. Parameter by mal byť vo formáte `Meno udržiavateľa<email>`.
- **Installed-Size** – uchováva veľkosť balíčku.
- **Depends** – zoznam balíčkov, ktoré sú potrebné pre fungovanie tohto konkrétneho balíčka. Balíček nebude nainštalovaný pokiaľ nebudú všetky balíčky zapísané v `Depends` správne nakonfigurované.
- **Provides** – obsahuje zoznam poskytnutých virtuálnych balíčkov. Virtuálny balíček je generický názov akejkoľvek skupiny balíčkov, ktoré vykonávajú podobnú funkcionality. Debian disponuje mechanizmom, ktorý umožňuje administrátorovi systému nastaviť jeden z balíčkov ako preferovaný v prípade, že viac balíčkov poskytuje rovnaký virtuálny balíček.
- **Section** – špecifikuje kategóriu balíčku na základe softwaru, ktorý inštaluje.
- **Priority** – určuje dôležitosť balíku. Môže nadobúdať hodnoty `required`, `important`, `standard`, `optional`, `extra`. [12]
 - **required** – balíčky, ktoré sú nevyhnutné pre správne fungovanie systému. Toto zahŕňa nástroje na opravu chýb, preto je dôležité tieto balíčky neodstraňovať. Takéto jednanie môže viesť k poškodeniu systému.
 - **important** – balíčky, ktoré by sa mali nachádzať na každom unixoidnom operačnom systéme.
 - **standard** – balíčky, ktoré sú štandardom na každom Linuxovom systéme. Súčasťou sú nástroje na odosielanie emailov a sťahovanie súborov z archívnych serverov.
 - **optional** – balíčky, ktoré môžu byť nainštalované alebo nevyžadujú špeciálne požiadavky.

- **extra** – balíčky, ktoré sú v konflikte s ostatnými vyššími prioritami alebo majú špeciálne požiadavky, ktoré ich robia nevhodnými pre kategóriu `optional`.
- **Homepage** – URL webstránky, na ktorej sa nachádza pôvodný zdroj balíčku a dodatočná dokumentácia.
- **Description** – popis binárneho balíčku skladajúci sa z dvoch častí, krátkeho a dlhého opisu.

2.2 Štruktúra repozitára

Debian repozitár je tvorený setom balíčkov usporiadaných do špeciálnej stromovej štruktúry. Môže obsahovať aj dodatočné súbory obsahujúce indexy a kontrolné súbory balíčkov. Repozitár musí obsahovať aspoň jeden adresár s nejakým Debian balíčkom a dvoma špeciálnymi indexovými súbormi: `Packages.gz` pre binárne balíčky a `Sources.gz` pre balíčky so zdrojovými kódmi. [14]

```
(repository root)
|
+-dists
|
|  -stable
|  |  -main
|  |  |  -binary-amd64
|  |  |  -binary-armhf
|  |  |  -binary-...
|  |  +-source
|  |  -contrib
|  |  |  -binary-amd64
|  |  |  -binary-armhf
|  |  |  -binary-...
|  |  +-source
|  +-non-free
|  |  -binary-amd64
|  |  -binary-armhf
|  |  -binary-...
|  +-source
|
+-pool
```

Obrázok 3 : Príklad štruktúry repozitára Debian

Koreňový adresár repozitára obsahuje adresár `dists`, ktorý v sebe zahrňuje adresár každého vydania. Debian repozitár je tvorený niekoľkými vydania. Tieto vydania nesú pomenovanie po postavách z filmov *Toy Story*. Vydanie je rozdelené na niekoľko komponentov, medzi ktoré patria `main`, `contrib`, a `non-free`. Tieto komponenty indikujú licenčné podmienky softwaru, ktorý obsahujú. Repozitár sa ďalej delí na jednotlivé architektúry na základe typu hardwaru, pre ktorý boli balíčky skompilované. Tieto podadresáre majú formát `binary-<arch>`. V nich sa nachádzajú textové súbory `Packages`, ktoré obsahujú meta-informácie o balíčkoch. Súčasťou adresára s názvom komponenty sú taktiež súbory `Sources`. [15]

Samotné balíčky sa nachádzajú v podadresári `pool` v rámci koreňového adresára. Adresár `pool` je opäť členený na komponenty a v tých sa nachádzajú podadresáre pomenované podľa softwarového balíčku. V týchto podadresároch sa nachádzajú balíčky s príponou `.deb`. Dodatočné podadresáre pomenované podľa začínajúceho písmena názvu balíčka slúžia na elimináciu príliš veľkého počtu súborov v rámci jedného adresára. Zahŕtenie adresára veľkým počtom súborov vedie k zníženiu výkonnosti množstva tradičných systémov. [15]

Súbory Release

Súbory `InRelease` sa nachádzajú v adresári `dists/<distribution>/InRelease`. Mali by obsahovať meta-informácie o distribúcii a kontrolné súčty indexových súborov, podľa možnosti podpísané OpenPGP signatúrou. Na starších klientoch sa môžeme stretnúť so súborom `dists/<distribution>/InRelease` bez podpisu. [14]

Medzi povinné parametre, ktoré by mal súbor `Release` obsahovať patria:

- **Suite and/or Codename**
- **Architectures**
- **Components**
- **Date**
- **SHA256**

Súbory Packages

Súbory `dists/<distribution>/<component>/<architecture>/Packages` sú nazývané indexy binárnych balíčkov. [14] Skladá sa z niekoľkých paragrafov s definovaným formátom. Medzi dodatočné parametre patria:

- **Filename (povinné)**
- **Size (povinné)**
- **MD5sum, SHA1, SHA256 (odporúčané)**
- **Description-md5 (voliteľné)**

Pokiaľ sú nasledujúce parametre obsiahnuté v súbore `control`, musí o nich existovať aj záznam v súbore `Packages` :

- **Depends**
- **Installed-Size**
- **Multi-Arch**

2.3 Vytvorenie repozitára

Vytvorenie a údržbu repozitára nám uľahčuje niekoľko jednoduchých nástrojov. Užívateľ má na výber z pomerne veľkého množstva nástrojov a preto sa bude nasledujúca časť zaoberať bližším predstavením niekoľkých z nich. Všetky nástroje na tvorbu indexov repozitára sú zdokumentované v manuálových stránkach. [15]

- **dak (Debian Archive Kit)** - dak je kolekcia nástrojov určených na správu balíčkových repozitárov Debian. Bol navrhnutý na správu veľkých repozitárov a množín dát. Práve vďaka jeho komplexnosti nie je vhodný na zaobchádzanie s osobným repozitárom. Keďže nebol určený na používanie mimo systému Debian, nie je dostatočne zdokumentovaný.
- **dpkg-scanpackages a dpkg-scansources** - `dpkg-scanpackages` slúži na skenovanie balíčkov s príponou `.deb` a generuje súbor `Packages`. Nástroj `dpkg-scansources` skenuje súbory s príponou `.dsc` a výsledkom je súbor `Sources`.
- **apt-ftparchive-** generuje komplexné indexy balíčkov, ktoré umožňujú plnohodnotné využívanie repozitára. Je supersetom `dpkg-scanpackages` a `dpkg-scansources`.
- **reprepro** - V minulosti známy ako `mirrorer`. Ukladá súbory do adresára `pool`. Balíčky a kontrolné súčty súborov sú ukladané do súboru Berkeley DB database, takže nie je potrebný žiadny dodatočný databázový server.

2.4 Podpisovanie balíčkov

Podpisovanie balíčkov je dôležitou súčasťou bezpečnostnej infraštruktúry. Umožňuje príjemcovi dát overiť, že nedošlo k nežiadúcej manipulácii s dátami po tom čo bol balíček podpísaný. Takýmto spôsobom sme schopný predísť útokom *man-in-the-middle*, ktoré by mohol využiť útočník na modifikáciu dát obsiahnutých v balíčkoch. Debian štandardne neposkytuje podpísané balíčky, avšak poskytuje podporu pre OpenPGP overovanie podpisov podpísaných balíkov. Namiesto toho sú balíčky overované podpísaním súboru `Release`, ktorý v sebe obsahuje metadáta. Medzi týmito metadátami sú obsiahnuté kontrolné súčty jednotlivých súborov v repozitári.

Takýmto spôsobom je zabezpečená autentickosť súborov v repozitári. [18]

V prípade podpisovania jednotlivých balíčkov existujú dve implementácie a to pomocou nástrojov `debsigs` alebo `dpkg-sig`. Podpisy vytvorené týmito nástrojmi nie sú medzi sebou kompatibilné, čo znamená, že nie je možné overiť podpis vytvorený druhým nástrojom. [19] Pri implementácii podpisovania balíčkov v repozitári bol zvolený nástroj `dpkg-sig`, pretože je užívateľsky prívetivejší ako jeho náprotivok.

dpkg-sig

Nástroj `dpkg-sig` nie je podporovaný softwarom `dpkg`, takže balíčky podpísané týmto nástrojom musia byť manuálne skontrolované pomocou `dpkg-sig`. V rámci podpisovania je v `ar` archíve balíčku vygenerovaný súbor s niekoľkými užitočnými informáciami:

- **Verzia nástroja `dpkg-sig`, ktorá vyhotovila podpis**
- **Informácie o podpisovateľovi**
- **Rola**
- **Kontrolné súčty, veľkosti súborov a názvy súborov obsiahnutých v balíčku Debian**

Tento súbor je pomenovaný na základe vybranej role. Pokiaľ nie je špecifikovaná pri vytváraní podpisu, tak je pridelená predvolená rola s názvom `builder`. V takomto prípade bude mať vygenerovaný súbor názov `_gpgbuilder`. [19]

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Version: 4
Signer:
Date: Tue Apr 25 13:11:39 2023
Role: builder
Files:
    3cf918272ffa5de195752d73f3da3e5e 7959c969e092f2a5a8604e2287807ac5b1b384ad 4 debian-binary
    3cc6e0bfe0b03123eb3704bae580b470 c8051ce599831c1312481da9aaced9fb423038cd 1792 control.tar.gz
    9ad5f93de942022bd5a03376aa3adc01 2f67638271e003ad93d75cb24176e252e982e6d4 71504 data.tar.xz
-----BEGIN PGP SIGNATURE-----

iQGzBAEBAgAdFiEEVRbguAaK5f3mT/3R09Yft83ZuKGFAmRICcsACgkQ09Yft83Z
uKjnKQwAxI9VRNvMkbC2eRZhULUJJ3tmJw00K/ipKL68yNOEVzxq0zWiXmKn2qHE
7VmdYu58mSxw2IwRQvUvV82kWkY9Q2OXODwBBM5n36prKIUgoIhZ3VuB1v4+JGUG
eXwp0ciGw7HxPUMhqn+XAJJNrd54hwqI3lrYH7W1eg020pyg0lqGHSyY6ZqBnPr
qnDEELe/r0cEI9FVnSTHsfAY0deuEjeDTf4hs/NgZCV1BuuMLe7KcoRpcW8xn0W7
zYLnLLR5Bo0egDfuv8hYLMkLmKUHeghUQpkcd/oabbxYPzCKVQmCjnfOIBiUcJIA
49Xk/k0rJUBmQayLAtcgtr7EBXbnwi7F0bJcVfWaV4YQ8to40Lhhg6NAzXGSwZ0
F4RweJ+Qe2MJ5m3sAWFMQk+bygnLa8m2PoJntzM4ICKPbRxoAmkaE/7tM4hmAu9m
484RioQ90I0THc5VgPSTSHpGc0N8KfY2tuctXcE6iIE/Q3lEx2rRkMebx7W4zv0J
i8aIynda
=Uceq
-----END PGP SIGNATURE-----
```

Obrázok 4 : Obsah súboru `_gpgbuilder`

Podpis balíku je vykonaný pomocou príkazu:

```
dpkg-sig --sign package_1.0.7_amd64.deb
```

Overenie podpisu je možné vykonať pomocou príkazu:

```
dpkg-sig --verify package_1.0.7_amd64.deb
```

V prípade správnosti podpisu je výstup príkazu:

```
└─$ dpkg-sig --verify python3-babelfish_5.3-1_all(5).deb  
Processing python3-babelfish_5.3-1_all(5).deb ...  
GOODSIG _gpgbuilder 5516E0B8068AE5FDE64FFDD1D3D61F4FCDD9B8A8 1682442699
```

Obrázok 5 : Výstup príkazu dpk-sig –verify

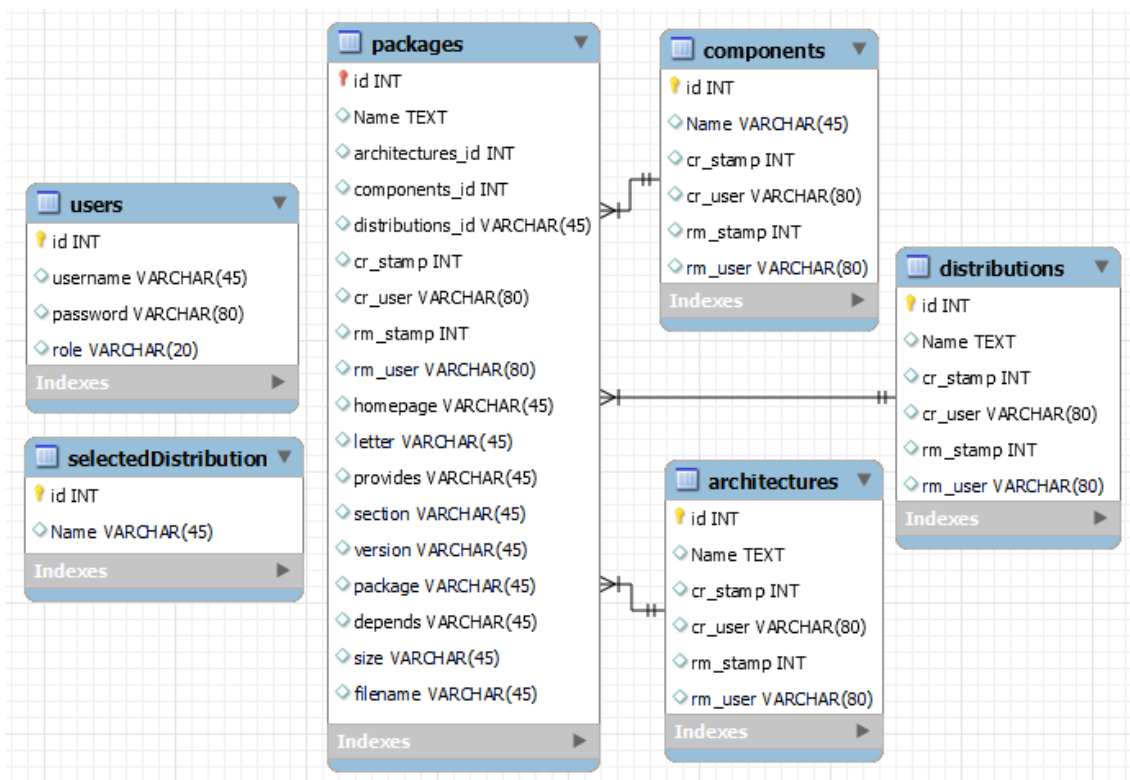
3. IMPLEMENTÁCIA WEBOVÉHO ROZHRAINIA

3.1 Štruktúra databázy

Nevyhnutnou súčasťou vývoju každej aplikácie je vytvorenie databázy, ktorá sa postará o správu dát akumulovaných počas jej využitia. Dáta sa do systému ukladajú pomocou systému SQLite a sú rozdelené do tabuliek. Kapitola obsahuje vysvetlenie významu jednotlivých tabuliek, ktoré sa nachádzajú v databáze. Súčasťou kapitoly je aj fyzický model pre databázový systém. Model popisuje atribúty entít a vzťahy medzi jednotlivými entitami.

Tabulka 1 : Význam tabuliek

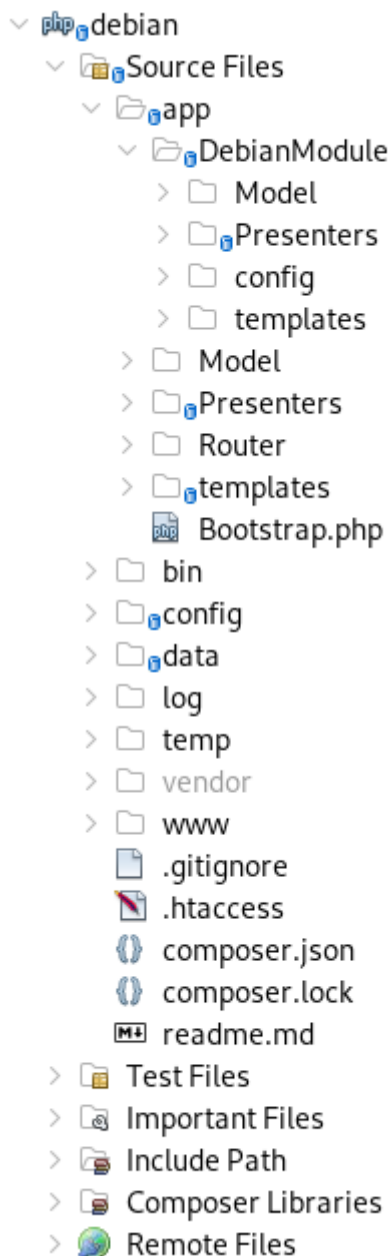
Názov tabuľky	Použitie
packages	Obsahuje všetky dôležité informácie o balíčkoch. Parametre sú získané extrahovaním informácií zo súboru control. Tieto parametre sú bližšie popísané v kapitole 2.1.1.
architectures	Obsahuje informácie o type hardwaru, pre ktorý bol balíček skompilovaný.
components	Obsahuje názvy všetkých komponentov, na ktoré sa delí repozitár Debian. Tieto komponenty indikujú licenčné podmienky softwaru, ktorý je v nich obsiahnutý.
distributions	Obsahuje názvy všetkých distribúcií, do ktorých je možné vkladať balíčky.
selectedDistribution	Pomocná tabuľka, ktorá slúži na dynamické uchovávanie názvu distribúcie. Názov distribúcie sa mení na základe užívateľského výberu distribúcie dostupnej v rozbaľovacej ponuke.
users	Obsahuje záznamy všetkých existujúcich užívateľov. Medzi uchovávané dáta patria ID užívateľa, užívateľské meno, heslo a užívateľská rola. Na základe týchto parametrov sa vytvorí užívateľská identita.



Obrázok 6 : Štruktúra databáze

3.2 Štruktúra aplikácie

Dokumentácia nette framework disponuje doporučenou adresárovou štruktúrou. Implementácia aplikácie vychádzala z týchto doporučení. V tejto podkapitole je predstavený DebianModule a jednotlivé objekty, ktoré sa v ňom nachádzajú. Štruktúra aplikácie je zobrazená na nasledujúcom obrázku.



Obrázok 7 : Štruktúra aplikácie

3.2.1 DebianModule

Moduly slúžia na združenie častí aplikácie s podobnou funkcionalitou. Projekt momentálne obsahuje jeden modul s názvom `DebianModule`, ktorý predstavuje jadro aplikácie. Tento modul obsahuje všetky modely, presentery a šablóny, ktoré sa využívajú na vytvorenie a správu repozitára.

3.2.2 Modely

Modely predstavujú logiku aplikácie. V tejto časti sú predstavené modely obsiahnuté v `DebianModule`.

AdminManager – obsahuje metódy slúžiace na správu užívateľov a prácu s tabuľkou `users`, ktorá je tu inicializovaná. Súčasťou modelu je metóda `getUsersIdentity` pomocou ktorej je možné získať užívateľskú identitu na základe užívateľského mena. Táto metóda je využitá pri prihlásení užívateľa pomocou platného certifikátu. Medzi metódy upravujúce tabuľku `users` patria:

- **updateUsername** – aktualizácia užívateľského mena
- **updatePassword** – zmena užívateľského hesla
- **updateGroup** – zmena autorizačnej skupiny
- **addNewUser** – pridanie nového užívateľa
- **removeUser** – odstránenie užívateľa

DistributionManager – stará sa o úkony spojené so správou distribúcií. Dochádza tu k inicializáciám tabuliek `distributions` a `selectedDistribution`. Medzi metódy upravujúce tabuľku `distributions` patria:

- **addDistribution** – pridanie novej distribúcie
- **addPackageToDistribution** – priradenie balíčka ku konkrétnej distribúcii
- **removeDistribution** – odstránenie distribúcie

Nachádzajú sa tu aj metódy ovplyvňujúce štruktúru adresára `dists`. Medzi tieto metódy patria:

- **distsStructure** – vytvorenie štruktúry adresára `dists` so všetkými patričnými podadresármi
- **clearPackagesIndicies** – odstránenie všetkých indexových súborov vygenerovaných pre konkrétnu distribúciu

PackageManager – slúži k správe balíčkov. Dochádza v ňom k inicializácii tabuľky `packages`. Medzi metódy upravujúce tabuľku `packages` patria:

- **addPackage** – pridanie vloženého balíčka do tabuľky `packages`
- **removePackage** – odstránenie balíčka z databázy

Nachádzajú sa tu metódy upravujúce štruktúru repozitára. Medzi tieto metódy patria:

- **poolStructure** – vytvorenie štruktúry adresára `pool` so všetkými patričnými podadresármi
- **generateIndicies** – generovanie indexových súborov
- **deleteIndicies** – odstránenie vygenerovaných indexových súborov, v ktorých sa nachádzal odstránený balíček
- **regenerateIndicies** – opätovné vygenerovanie indexových súborov, v ktorých sa nachádzal odstránený balíček

Package – objekt slúžiaci na získanie informácií o balíčkoch a následnému generovaniu indexových súborov. Súčasťou objektu sú `getter`y pre jednotlivé atribúty balíčkov. Obsahuje taktiež dôležité metódy z hľadiska funkcionality repozitára. Medzi tieto metódy patria:

- **load** – slúži na rozbalenie balíčka `.deb`.
- **parse** – slúži na získanie informácií zo súboru `control`.
- **metaData** – zabezpečuje vygenerovanie indexových súborov `Packages`
- **releaseData** – zabezpečuje vygenerovanie indexových súborov `Release` v podadresári s názvom distribúcie
- **renderReleaseFile** – zabezpečuje vygenerovanie indexových súborov `Release` v podadresároch jednotlivých architektúr
- **signRelease** – podpisuje súbor `Release` vygenerovaný pomocou metódy `releaseData`
- **signPackage** – slúži na podpísanie jednotlivých balíčkov vkladných do repozitára
- **gzCompressFile** – slúži na komprimovanie balíčka

Výpis 1: Ukážka kódu funkcie slúžiacej na rozbalenie balíčka

```
public function load(string $path) {
    $this->holder = $this->path;
    $this->path = exec("mktemp -d", $out, $err);
    if (false === $this->path || 0 < $err) {
        throw \RuntimeException("Cannot decompress package");
    }
    chdir($this->path);
    $control = exec("ar -t '$path' | grep -i 'control'");
    exec("ar -x '$path' -output = '$this->path' '$control'");
    exec("tar -xf '$control'");
}
```

ArchitectureManager – slúži k správe architektúr využívaných v repozitári. Vytvára tabuľku `architectures`.

ComponentsManager – slúži k správe komponentov. Dochádza tu k inicializácii tabuľky `components`.

3.2.3 Presentery

O zaistenie predávania dát z modelovej vrstvy do príslušných šablón sa starajú nasledujúce presentery:

AdminPresenter – zabezpečuje úkony spojené s prihlasovaním užívateľov. Prijíma parametre od klienta a predáva ich ďalej modelu `AdminManager`, ktorý porovná poskytnuté parametre so záznamami v databáze. Súčasťou presentera je aj metóda `generateClicked`, ktorá je volaná po kliknutí na tlačidlo `Generate New Key` v rámci Administrátorského panelu. Po zavolaní tejto metódy dôjde k vygenerovaniu nového GPG kľúča. Proces prihlasovania pomocou certifikátu je zabezpečený pomocou metód:

- **hasValidCert** – kontroluje či užívateľ poskytol platný certifikát
- **actionCertificateSignIn** – v prípade že užívateľ poskytol platný certifikát, porovná sa `Common Name` poskytnuté pri vystavení certifikátu so záznamami užívateľských mien. V prípade že dôjde k zhode, použije sa identita daného užívateľa na prihlásenie do webového rozhrania.

DistrosPresenter – je zodpovedný za získavanie dát potrebných na vykreslenie šablón `default.latte` a `distsSelect.latte`. Dochádza tu k volaniu metódy `distsStructure`, ktorá sa postará vytvoreniu dátovej štruktúry `dists`.

UsersPresenter – dochádza tu k získaniu dát potrebných pre vykreslenie šablóny `usersPanel.latte`.

HomepagePresenter – stará sa o zabezpečenie prístupu k domovskej stránke repozitára. V prípade že užívateľ nie je prihlásený, bude presmerovaný na prihlasovaciu stránku.

DistsPresenter – zabezpečuje sťahovanie `indices` súborov poskytnutím štandardnej cesty.

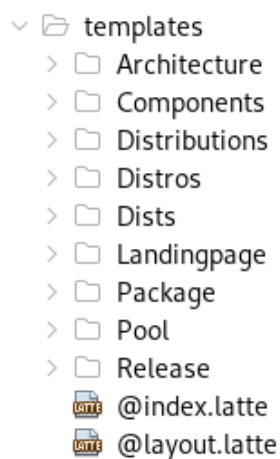
PoolPresenter – slúži na zabezpečenie štandardnej cesty, ktorá je vyžadovaná pri sťahovaní balíčkov s využitím nástroja `apt`.

LandingpagePresenter – zabezpečuje dáta pre vykreslenie šablóny `default.latte`.

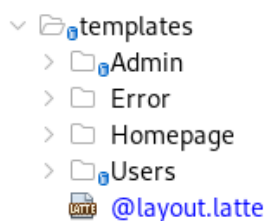
PackagePresenter – získava dáta z objektov `Package` a `PackageManager` a distribuuje ich do šablóny `default.latte`. Nachádza sa tu metóda `loadPackages`, ktorá je volaná po vložení balíčka do repozitára. Táto metóda sa stará o získanie informácií o balíčku. Extrahované informácie predstavujú argumenty predávané metóde `addPackage`, ktorá sa nachádza v `PackageManager`. Následne dôjde k vygenerovaniu indexových súborov pomocou metódy `generateIndices` a zavolaniu metódy `poolStructure`, ktorá sa postará o vytvorenie štruktúry adresára `pool`. Ďalšou metódou obsiahnutou v tomto presenteri je metóda `actionRemovePackage`, ktorá zabezpečuje odstránenie balíčka z repozitára. Po tom čo je balíček odstránený dôjde k opätovnému vygenerovaniu indexových súborov pomocou metódy `regenerateIndices`.

3.2.4 Šablóny

Štruktúra adresára `template` je tvorená podadresármi patriacimi jednotlivým presenterom. V tejto kapitole sú bližšie predstavené šablóny obsiahnuté v týchto podadresároch. Nasledujúce obrázky popisujú štruktúru adresára `templates`.



Obrázok 8 : Štruktúra adresára templates v DebianModule



Obrázok 9 : Štruktúra adresára templates

Obsah adresára `templates`:

@layout.latte – je hlavnou šablónou užívateľského rozhrania

@index.latte – je šablónou, ktorá je vykreslená pri prechádzaní štruktúrou repozitára Debian

Obsah jednotlivých podadresárov:

Distros – nachádza sa tu šablóna `default.latte`, ktorá vykresľuje formuláre slúžiace na správu distribúcií. Ďalej sa tu nachádza šablóna `distsSelect.latte`, ktorá sa stará o vykreslenie formulára pre vkladanie balíčkov do distribúcií.

Výpis 2: Vykresľovanie distribúcií v rozbalovacej ponuke šablóny distsSelect.latte

```
<select class="form-select" n:name="distributions">
  <option value="" selected disabled>
    Please select Distribution
  </option>
  {foreach $dists as $item}
  <option>{$item}</option>
  {/foreach}
</select>
```

Landingpage – nachádza sa v ňom šablóna `default.latte`, ktorá vykresľuje obsah podadresára `debian`.

Package – obsahuje šablónu `default.latte`, ktorá vykresľuje formulár zabezpečujúci vkladanie balíčkov do repozitára. Ďalej sa stará o vykreslenie tabuľky `Available Packages`, ktorá obsahuje záznamy o všetkých dostupných balíčkoch.

Dists – obsahuje šablónu `default.latte`, ktorá sa stará o vykreslenie podadresárov a indexových súborov nachádzajúcich sa v adresári `dists`.

Pool – obsahuje šablónu `default.latte`, ktorá zabezpečuje vykreslenie balíčkov a samotných podadresárov, ktoré sa nachádzajú v adresári `pool`.

3.3 Autentizácia a autorizácia

V aplikácii bola implementovaná podpora autentizácie pomocou prihlasovacieho formulára alebo poskytnutím platného certifikátu. Autentizačnými údajmi sú užívateľské meno a heslo. Pri prihlásení sú porovnané prihlasovacie údaje poskytnuté užívateľom s údajmi obsiahnutými v tabuľke `users`. V prípade zhody je užívateľ prihlásený a je mu pridelená príslušná užívateľská identita.

Proces prihlasovania pomocou certifikátu zahŕňa kontrolu platnosti poskytnutého certifikátu. O platnosti certifikátu rozhoduje metóda `hasValidCert`, ktorá zisťuje či certifikát poskytnutý užívateľom obsahuje všetky dôležité parametre. Kombinácia sériového čísla certifikátu a vystaviteľa certifikátu garantuje unikátnu identifikáciu poskytnutého certifikátu. [20] Dôležitou premennou je `SSL_CLIENT_VERIFY`, ktorá v prípade platnosti certifikátu vráti hodnotu `SUCCESS` a v prípade, že certifikát nebol poskytnutý vracia hodnotu `NONE`. Ak je certifikát poskytnutý užívateľom platný, porovná sa `Common Name` zadaný pri vystavení certifikátu so záznamami užívateľských mien uložených v databáze. V prípade, že dôjde k zhode, použije sa identita daného užívateľa na prihlásenie do webového rozhrania.

Výpis 3: Ukážka kódu funkcie overujúcej platnosť certifikátu

```
public function hasValidCert()
{
    if (!isset($_SERVER['SSL_CLIENT_M_SERIAL'])
        || !isset($_SERVER['SSL_CLIENT_VERIFY'])
        || $_SERVER['SSL_CLIENT_VERIFY'] !== 'SUCCESS'
        || !isset($_SERVER['SSL_CLIENT_I_DN']))
    ) {
        return false;
    }

    return true;
}
```

Autorizácia užívateľov je zabezpečená definovaním rôznych úrovní prístupu pre užívateľov. Každému užívateľovi je po prihlásení priradená rola, ktorá je súčasťou užívateľskej identity. Jednotlivým rolám sú priradené rôzne práva v systéme. Užívateľom je možné priradiť nasledujúce role:

Tabulka 2 : Autorizačné role a ich význam

Rola	Význam
Admin	Má najvyššie privilégia a prístup ku všetkým funkcionalitám
Developer	Disponuje právom pridávať balíčky do repozitára a vytvárať nové distribúcie. Užívatelia v roli developer majú prístup do užívateľského panelu.
User	Má právo sledovať obsah webového rozhrania.
Guest	Jedná sa o predvolenú rolu priradenú užívateľom, ktorí sú prihlásení bez užívateľského účtu. Rovnako ako užívatelia s rolou user, majú právo sledovať obsah webového rozhrania.

Prístup k jednotlivým funkcionalitám webového rozhrania je kontrolovaný pomocou autorizačného kritéria, ktorým je metóda `isInRole`. Tá je súčasťou objektu triedy `Nette\Security\User`. Metóda vracia rolu, v ktorej vystupuje prihlásený užívateľ a volá sa pri inicializácii každého presentera zabezpečujúceho funkcionality webového rozhrania. V prípade dostačujúcich privilégií je užívateľ pripustený k využitiu príslušných funkcionalít. Autorizácia užívateľa pred vstupom do Administrátorského panelu vyzerá nasledovne:

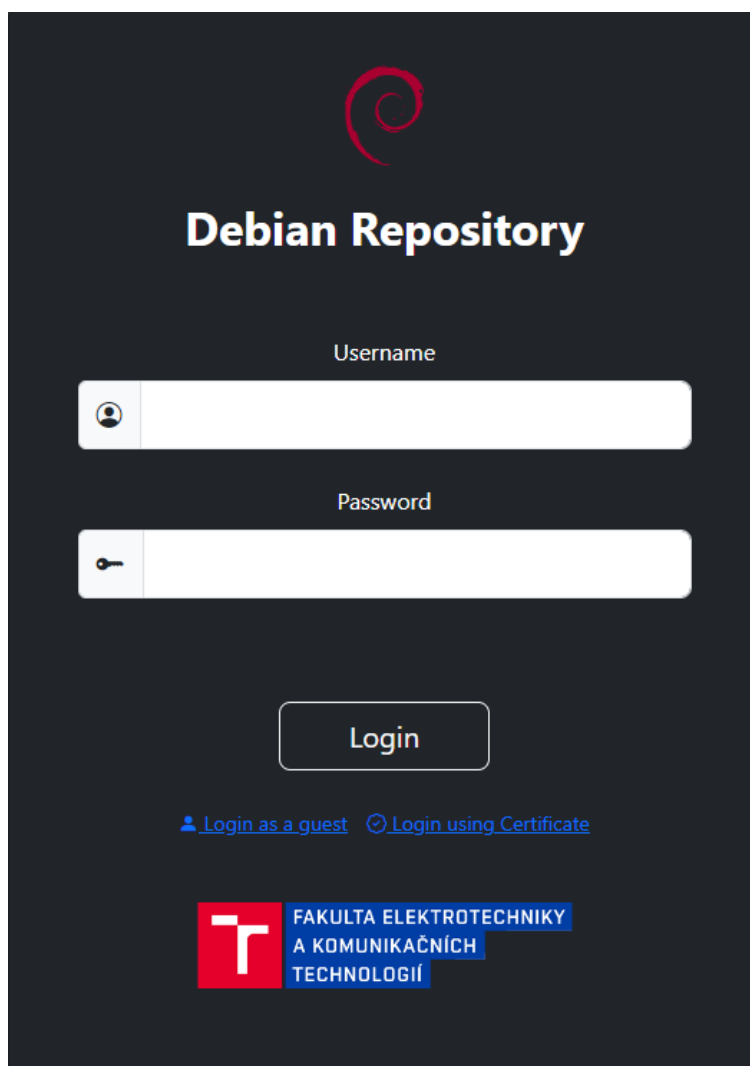
Výpis 4: Autorizácia užívateľa pomocou metódy `isInRole()`

```
if (!$this->user->isInRole('admin')) {
    $this->error('Forbidden', 403);
}
```

3.4 Užívateľské rozhranie

3.4.1 Prihlásenie do užívateľského rozhrania

Po prejení na stránku webového rozhrania je užívateľ presmerovaný na prihlasovaciu stránku. Užívateľ má možnosť prihlásiť sa do svojho účtu pomocou prihlasovacích údajov alebo na základe platného certifikátu. V prípade, že nedisponuje užívateľským účtom, môže využiť možnosť `Login as a guest`, ktorá mu umožní vstup do webového rozhrania s obmedzenými privilégiami. Prihlásenie na základe certifikátu je možné kliknutím na odkaz `Login using Certificate`.



Debian Repository

Username

Password

Login

[Login as a guest](#) [Login using Certificate](#)

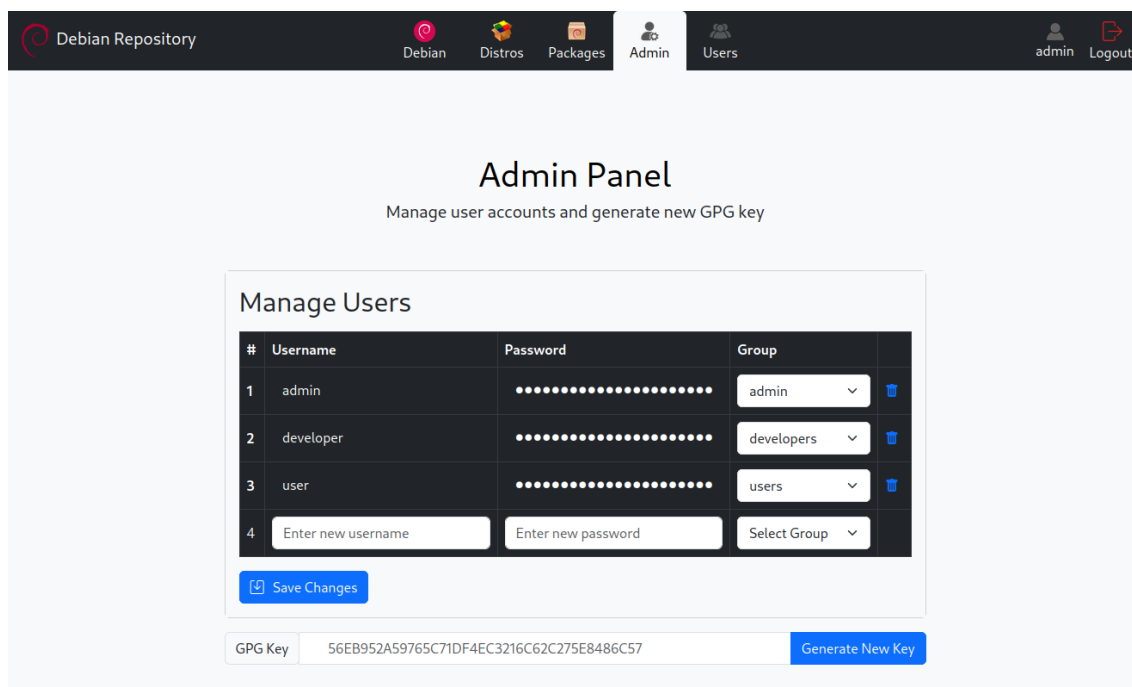
T FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH
TECHNOLOGIÍ

Obrázok 10 : Prihlasovací formulár

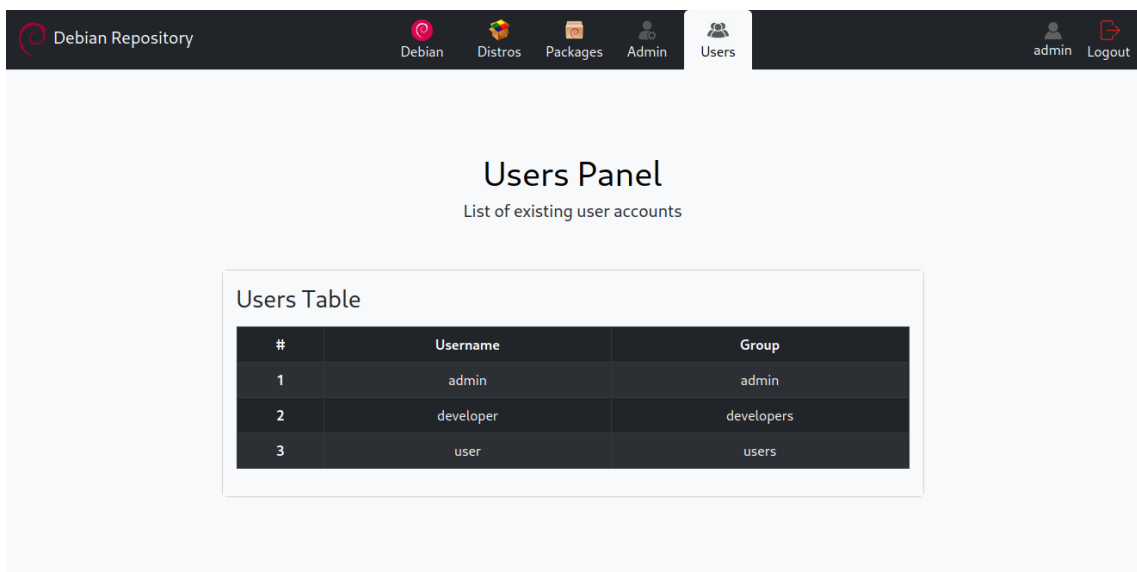
3.4.2 Administrátorský panel

Administrátorský panel slúži na správu webového rozhrania a je dostupný len pre užívateľov zo skupiny `admin`. Vstup do administrátorského panela je možný cez navigačnú lištu kliknutím na tlačidlo `Admin`. Administrátor má možnosť pridávať nové užívateľské účty alebo upravovať a odstraňovať existujúce účty. Všetky úkony týkajúce sa užívateľských účtov sa vykonávajú v rámci tabuľky s názvom `Manage Users`. Tá obsahuje zoznam užívateľských mien, hesiel a rozbaľovaciu ponuku s autorizačnými skupinami. Po vykonaní želaných zmien v tabuľke je potrebné zmeny potvrdiť pomocou tlačidla `Save Changes`. V prípade, že počas úpravy užívateľských účtov nastala chyba, užívateľ je o tom oboznámený pomocou chybových hlásení.

Ďalšou funkcionalitou, ktorá sa nachádza v administrátorskom paneli je vygenerovanie nového GPG kľúča, využívaného na podpisovanie balíčkov. Nový kľúč je možné vygenerovať kliknutím na tlačidlo `Generate New Key`.



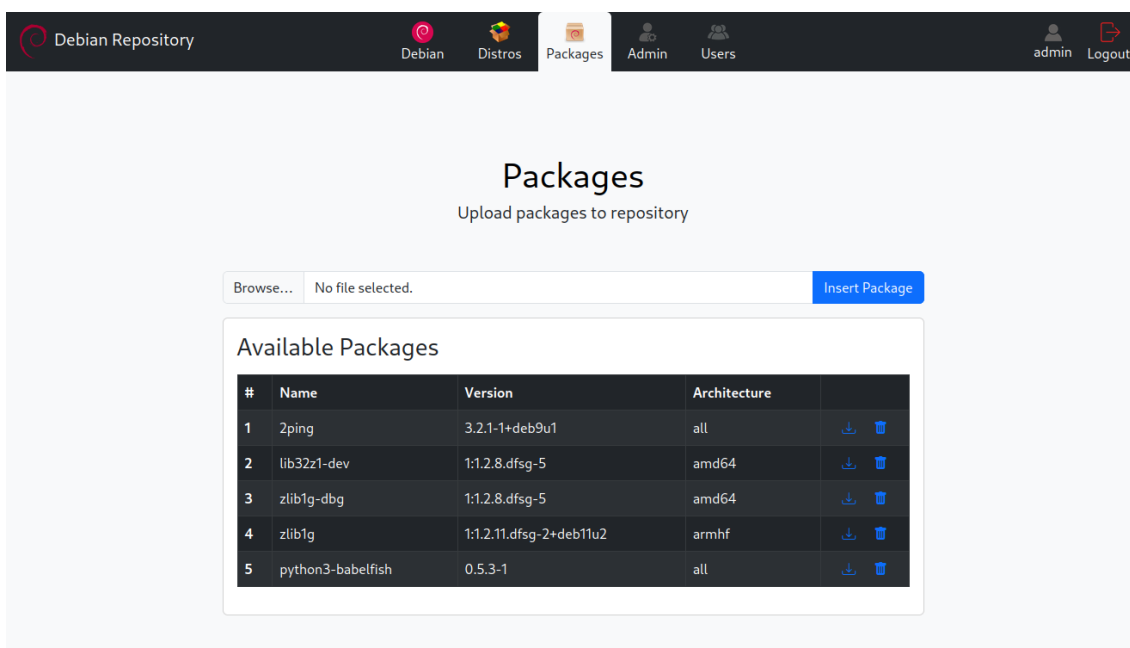
Obrázok 11 : Admin Panel



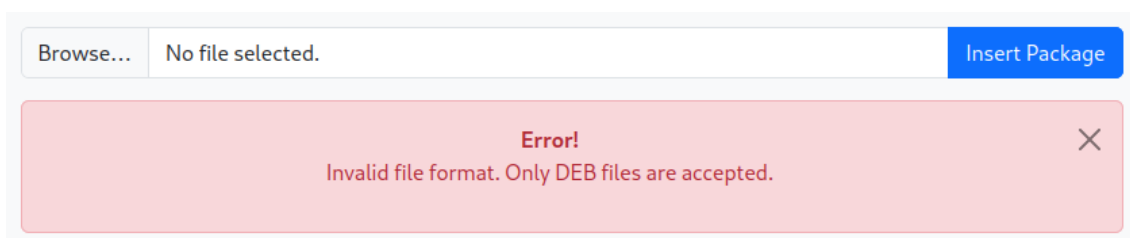
Obrázok 13 : Users Panel

3.4.4 Návod na vloženie balíčka do repozitára

Pre vloženie balíčku do repozitára je potrebné v rámci navigačnej lišty kliknúť na tlačidlo `Packages`. Užívateľ je presmerovaný na stránku `Packages`, ktorá slúži na správu balíčkov. Obsahom tejto stránky je `uploadForm`, pomocou ktorého je možné vkladať balíčky vo formáte `.deb`. Po vložení je balíček zobrazený v tabuľke s názvom `Available Packages`. Táto tabuľka obsahuje základné informácie o balíčku, medzi ktoré patria názov, verzia a architektúra balíčku. Pri každom balíčku sa nachádzajú dve ikony pomocou, ktorých môže užívateľ stiahnuť balíček lokálne do svojho zariadenia alebo odstrániť balíček z repozitára.



Obrázok 14 : Obsah stránky Packages

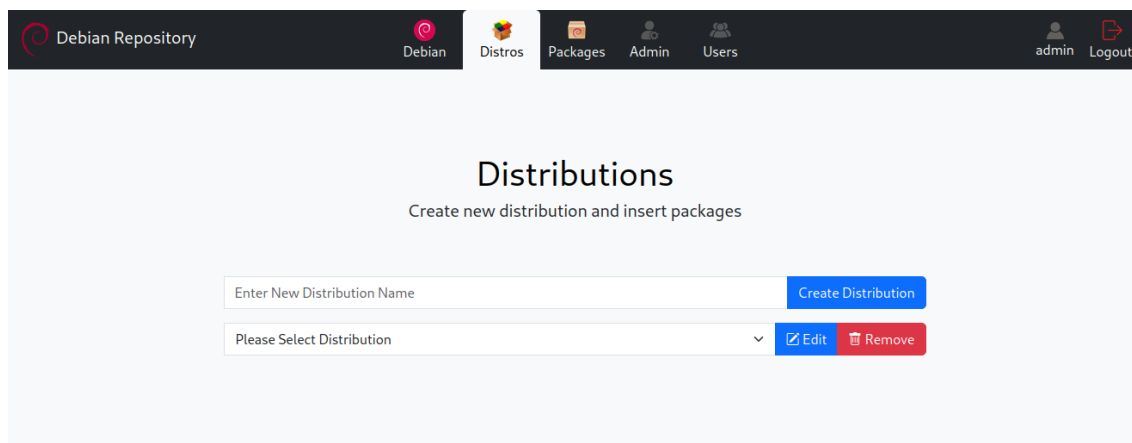


Obrázok 15 : Chybové hlásenie pri vložení súboru v nesprávnom formáte

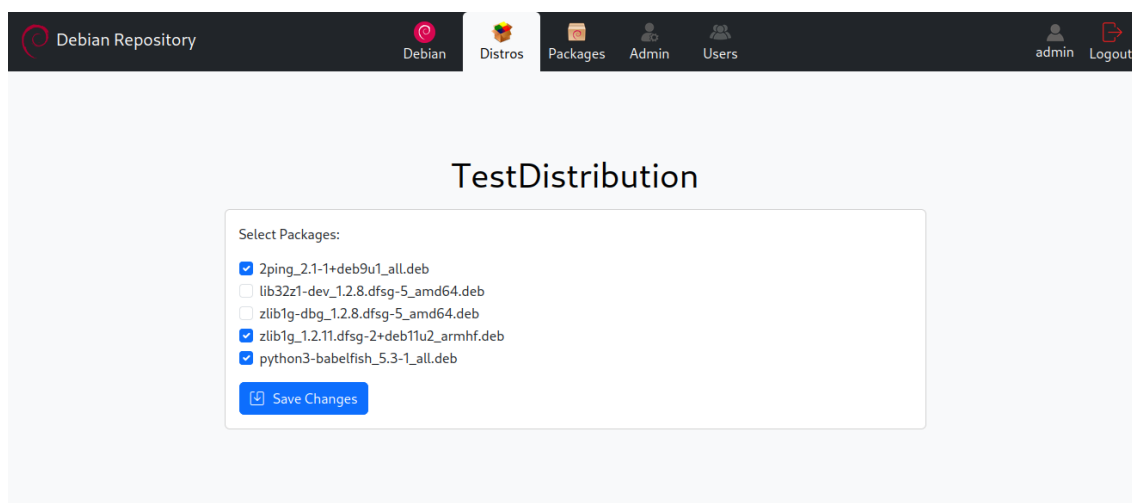
3.4.5 Návod na vytvorenie a úpravu distribúcie

Vytvorenie novej distribúcie je možné po kliknutí na tlačidlo `distros` v navigačnej lište. Po kliknutí je užívateľ presmerovaný na stránku `Distributions`. Užívateľ má možnosť vytvoriť novú distribúciu alebo upraviť a vymazať existujúcu distribúciu. Pre vytvorenie novej distribúcie je potrebné poskytnúť názov novej distribúcie a kliknúť na tlačidlo `Create Distribution`. Následne má užívateľ možnosť výberu spomedzi existujúcich distribúcií v rozbaľovacej ponuke. Po výbere distribúcie je možné prejsť k upravovaniu distribúcie pomocou tlačidla `Edit` alebo vybranú distribúciu odstrániť kliknutím na tlačidlo `Remove`. Po kliknutí na tlačidlo `Edit` je užívateľ presmerovaný na stránku obsahujúcu názov upravovanej distribúcie a výber spomedzi dostupných

balíčkov, ktoré zatiaľ nie sú priradené žiadnej distribúcii. Pri každom voľnom balíčku sa nachádza zaškrťavacie políčko, ktoré indikuje výber balíčku pre vloženie do upravovanej distribúcie. Po dokončení výberu musí užívateľ potvrdiť zmeny kliknutím na tlačidlo `Save Changes` a je presmerovaný späť na stránku `Distributions`.



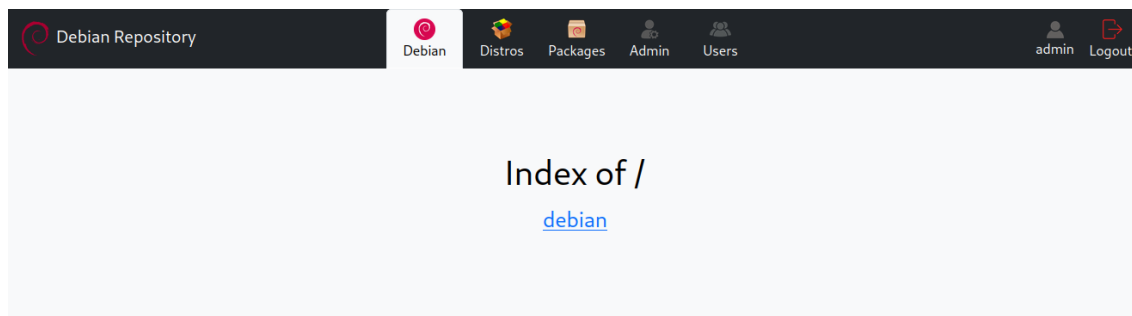
Obrázok 16 : Obsah stránky Distributions



Obrázok 17 : Vkladanie balíčkov do distribúcie s názvom TestDistribution

3.4.6 Navigácia v repozitári




Po prihlásení je užívateľ presmerovaný na domovskú stránku, ktorá obsahuje odkaz s názvom `debian`, pomocou ktorého môže užívateľ vstúpiť do samotného repozitára. Táto stránka je dostupná aj po kliknutí na tlačidlo `Debian`, nachádzajúce sa v navigačnej lište.



Obrázok 18 : Domovská stránka repozitára Debian

Po kliknutí na odkaz je užívateľ presmerovaný na stránky webového rozhrania, ktoré simulujú štandardnú štruktúru repozitára Debian. Každá stránka repozitára obsahuje dynamický nadpis, ktorý informuje užívateľa o jeho pozícii v repozitári. Domovská stránka repozitára obsahuje základné adresáre `dists` a `pool`.

Index of /debian

<u>Name</u>	<u>Last modified</u>	<u>Size</u>
 Parent Directory		-
 dists	2023-05-13 09:44	4 kB
 pool	2023-05-13 09:44	4 kB



Apache Server at ftp.debian.org Port 80

Obrázok 19 : Obsah adresára Debian

Adresár `dists`

Adresár `dists` obsahuje názvy všetkých vytvorených distribúcií. V podadresári s názvom distribúcie nájdeme komponentu `main` a súbory `Release` a `Release.gpg`, ktorý predstavuje podpis súboru `Release`.





Index of /dists

Name	Last modified	Size
 Parent Directory		-
 TestDistribution	2023-05-20 21:22	4 kB

Apache Server at ftp.debian.org Port 80

Obrázok 20 : Obsah adresára dists

Index of /dists/TestDistribution

Name	Last modified	Size
 Parent Directory		-
 main	2023-05-22 17:20	4 kB
 Release	2023-05-22 17:54	744 B
 Release.gpg	2023-05-22 17:54	659 B

Apache Server at ftp.debian.org Port 80

Obrázok 21 : Obsah podadresára s názvom distribúcie

```

1 Origin: repos
2 Label: PHP repos
3 Codename: murgas
4 Date: Sun May 14 16:39:00 CEST 2023
5 Architectures: amd64 armhf all
6 Components: main contrib non-free
7 Description: Apt repository for project x
8 MD5sum:
9 9a666943a790193a6d8707bddf2747f4 1481 binary-amd64/Packages
10 93b3f629917a948d55fe80c920ff4cf9 626 binary-amd64/Packages.gz
11 7b2057757c5674443b05a745e7ed364d 81 binary-amd64/Release
12 18234653bfb7412ed0e6c29af627d10c 1098 binary-armhf/Packages
13 aba76735f8f7d98960d7ae175259e2d9 545 binary-armhf/Packages.gz
14 8fefe90b110faf81ab486be1be353026 81 binary-armhf/Release
15 5e7d58e21813a61ae9031fc65f2e5b2e 778 binary-all/Packages
16 8726e217087a7cec5aa1a50effdc3bdb 433 binary-all/Packages.gz
17 407c242b87eeb3e9288bc457c487e8d8 79 binary-all/Release

```

Obrázok 22 : Obsah súboru Release v podadresári distribúcie

```



1 -----BEGIN PGP SIGNATURE-----
2
3 iQGzBAABCgAdFiEE15jvAyjApfW0I4axzrQzHI9Fu80FAmRg8oQACgkQzrQzHI9F
4 u81+lgv/ZUDuIe7QlgERb0q7K6neQVyMsh2ZZ0FcRiGlQw/cSTlN8j2EWuCTGfXq
5 c465Pb6rbhIJitkAQNKbJQ5/pgxHjQvPvUNcp8I4XrG201GL5Zw0WJHGoiFws9jv
6 chRLFQcQYETHyCnQQZcoLX2qsGkP0EcuPvHtVco5YVLP0Qf++tMLHfhSAM+P+d0h
7 8GrpoWnKTF0hMSuV2o7MLKDasydL4sX8DbUBbWNREwKS/n4vEQ0ItvUuRst5f0iG
8 Tb9NEqAGTL8YivsFJUlwFCZP0a0qke9goDxT+aAWg0ew5cC93RKHfp7YyDdLU0eV
9 GOvb5l34xgiyVcGXBKh/h22cLWrZW590r9WSZTaylg3U8rl3x6LAPzxqrdhGo09z
10 KyuKnRMcWLuJP5Kd8VX/yn6LFC/iDLPhxjLmDEorJY81/k7zIFAUuv50wzPCZhNr
11 V8y1kMomZ5SS8W1R/1A7y9LaefN1vL/T5fd0IjHhBD28EBzdI6UyBqroelthZXG/
12 UDu56PsA
13 =bbsX
14 -----END PGP SIGNATURE-----

```

Obrázok 23 : Obsah súboru Release.gpg

V podadresári komponentu `main` nájdeme zoznam podporovaných architektúr, pre ktoré boli balíčky skompilované.

Index of /dists/TestDistribution/main




Name	Last modified	Size
 Parent Directory		-
 binary-armhf	2023-05-22 17:54	4 kB
 binary-all	2023-05-22 17:54	4 kB
 binary-amd64	2023-05-22 17:54	4 kB

Apache Server at ftp.debian.org Port 80

Obrázok 24 : Obsah podadresára komponenty main

Pokiaľ sa v distribúcii nachádzajú balíčky skompilované pre vybranú architektúru, nájdeme v podadresári danej architektúry stiahnuteľné indexy binárnych balíčkov Packages a indexové súbory Release. Súbor Packages je možné stiahnuť aj v komprimovanej forme.

Index of /dists/TestDistribution/main/binary-amd64

Name	Last modified	Size
 Parent Directory		-
 Packages	2023-05-22 17:54	782 B
 Packages.gz	2023-05-22 17:54	440 B
 Release	2023-05-22 17:54	81 B

Apache Server at ftp.debian.org Port 80

Obrázok 25 : Obsah podadresára architektúry amd64

```

1 Package: python3-babelfish
2 Version: 0.5.3-1
3 Installed-Size: 310
4 Maintainer: Debian Python Modules Team <python-modules-team@lists.alioth.debian.org>
5 Architecture: amd64
6 Depends: python3-pkg-resources, python3:any (≥ 3.3.2-2~)
7 Provides:
8 Section: python
9 Priority: optional
10 MD5sum: 3eb16a4689238b9bbfb6afe597dde983
11 Filename: pool/main/p/python3-babelfish/python3-babelfish_0.5.3-1_all.deb
12
13 Package: 2ping
14 Version: 3.2.1-1+deb9u1
15 Installed-Size: 149
16 Maintainer: Ryan Finnie <ryan@finnie.org>
17 Architecture: amd64
18 Depends: init-system-helpers (≥ 1.18~), python:any (<< 2.8), python:any (≥ 2.7.5-5~)
19 Provides:
20 Section: net
21 Priority: optional
22 MD5sum: 6ba3dd1473b9f9992bf249072476c75a
23 Filename: pool/main/2/2ping/2ping_3.2.1-1+deb9u1_all.deb

```

Obrázok 26 : Obsah súboru Packages v podadresári architektúry amd64

```

1 Component: main
2 Origin: Debian
3 Label: Debian
4 Architecture: amd64
5 Description:

```






Obrázok 27 : Obsah súboru Release v podadresári architektúry amd64

Adresár pool

Adresár `pool` je rozdelený na podadresár s názvom komponenty `main`, podadresáre so začiatočnými písmenami jednotlivých balíčkov a názvami balíčkov. V podadresári s názvom balíčku nájdeme samotný balíček, ktorý je možný stiahnuť kliknutím na odkaz.

V adresári `pool` sa ukladajú balíčky, ktoré sú ďalej členené do podadresárov na základe prvého písmena v ich mene. Balíčky s rovnakým začiatočným písmenom sú ďalej rozdelené podľa názvu softwarového balíčku. V týchto podadresároch nájdeme vkladané balíčky s príponou `.deb`. Štruktúru adresára `pool` a podadresárov, ktoré sa v ňom nachádzajú bližšie približujú nasledujúce obrázky.




Index of /pool/main

<u>Name</u>	<u>Last modified</u>	<u>Size</u>
 Parent Directory		-
 l	2023-05-22 17:16	4 kB
 p	2023-05-22 17:16	4 kB
 z	2023-05-22 17:16	4 kB
 z	2023-05-22 14:10	4 kB

Apache Server at ftp.debian.org Port 80

Obrázok 28 : Obsah podadresára main

Index of /pool/main/z

<u>Name</u>	<u>Last modified</u>	<u>Size</u>
 Parent Directory		-
 zlib1g	2023-05-22 17:16	4 kB
 zlib1g-db	2023-05-22 17:16	4 kB

Apache Server at ftp.debian.org Port 80

Obrázok 29 : Obsah podadresára so začiatočným písmenom distribúcie

Index of /pool/main/z/zlib1g-dbg

<u>Name</u>	<u>Last modified</u>	<u>Size</u>
 Parent Directory		-
 zlib1g-dbg_1.2.8.dfsg-5_amd64.deb	2023-05-09 18:22	181 kB

Apache Server at ftp.debian.org Port 80

Obrázok 30 : Obsah podadresára s názvom softwarového balíčku

4. ZÁVER

Cieľom Bakalárskej práce bolo vytvorenie webového rozhrania repozitára systému Debian. Pred začatím implementácie webového rozhrania som sa bližšie zoznámil s jazykom PHP a frameworkom Nette. Keďže to bolo moja prvá skúsenosť s frameworkom Nette, musel som zo začiatku venovať veľa času vyhľadávaniu informácií, ktoré nie sú úplne dostatočne vysvetlené v dokumentácii. Akonáhle sa však užívateľ prenesie týmito začiatkami, odmenou sú všetky výhody, ktoré Nette framework prináša. Prvým krokom pri implementácii webového rozhrania bolo preštudovanie možností rozbaľovania balíčkov a spôsobov získania informácií zo súborov, ktoré sú v ňom obsiahnuté. Následne som vytvoril databázu, ktorá uchováva informácie získané z balíčkov, užívateľské identity existujúcich užívateľov a zoznamy architektúr, komponentov a distribúcií. Po vytvorení tabuliek nasledovalo vytvorenie presenterov a šablón zabezpečujúcich vytvorenie štruktúry repozitára a vygenerovanie príslušných indexových súborov. Aplikácia bola ďalej rozšírená o užívateľské rozhranie, ktoré umožňuje užívateľovi vkladať balíčky do repozitára, vytvárať distribúcie a vkladať balíčky do konkrétnych distribúcií. Balíčky sú po vložení do repozitára podpísané a podpísaný je taktiež zoznam balíčkov nachádzajúcich sa v jednotlivých distribúciách. Kliknutím na odkaz nachádzajúci sa na domovskej stránke, dôjde k presmerovaniu užívateľa na stránky simulujúce štruktúru repozitára Debian. Súčasťou užívateľského rozhrania je administrátorský panel slúžiaci na správu užívateľských účtov a generovanie GPG kľúča, využitého pri podpisovaní balíčkov. Nachádza sa tu aj užívateľský panel, ktorý pomocou prehľadnej tabuľky poskytuje informácie o užívateľských účtoch. V aplikácii bola implementovaná podpora autentizácie pomocou prihlasovacieho formulára alebo poskytnutím platného certifikátu. Autorizácia užívateľov bola zabezpečená definovaním rôznych úrovní prístupu pre užívateľov.

LITERATURA

- [1] PHP: What is PHP?. [online] [cit. 2022-27-11]. Dostupné z:
<https://www.php.net/manual/en/intro-what-is.php>
- [2] GILMORE, W. J. *Velká kniha PHP a MySQL 5: kompendium znalostí pro začátečníky i profesionály*. Vyd. 1. [i.e. 2. vyd.]. Brno: Zoner Press, 2007. Encyklopedie webdesignera. ISBN 978-80-86815-53-4.
- [3] WIKIPEDIA: PHP. [online]. [cit. 2022-27-11]. Dostupné z:
<https://cs.wikipedia.org/wiki/PHP>
- [4] WIKIPEDIA: Nette_Framework. [online]. [cit. 2022-27-11]. Dostupné z:
https://cs.wikipedia.org/wiki/Nette_Framework
- [5] ZDROJAK: Nette Framework: zvyšte svoji produktivitu. [online]. [cit. 2022-27-11]. Dostupné z: <https://zdrojak.cz/clanky/nette-framework-zvyste-svoji-produktivitu/>
- [6] NETTE: Presenters [online]. [cit. 2022-27-11]. Dostupné z:
<https://doc.nette.org/cs/application/presenters>
- [7] NETTE: Model [online]. [cit. 2022-27-11]. Dostupné z:
<https://doc.nette.org/cs/quickstart/model>
- [8] NETTE: Templates [online]. [cit. 2022-27-11]. Dostupné z:
<https://doc.nette.org/cs/application/templates>
- [9] NETTE: Latte [online]. [cit. 2022-27-11]. Dostupné z:
<https://latte.nette.org/cs/>
- [10] SQLITE: Features [online]. [cit. 2022-27-11]. Dostupné z:
<https://www.sqlite.org/features.html>
- [11] WIKIPEDIA : SQLite [online]. [cit. 2022-27-11]. Dostupné z:
<https://cs.wikipedia.org/wiki/SQLite>
- [12] DEBIAN: Manuals [online]. [cit. 2022-27-11]. Dostupné z:
<https://www.debian.org/doc/manuals/debian-faq/pkg-basics.en.html>

- [13] LINUX.DIE: Deb-control [online]. [cit. 2022-27-11]. Dostupné z: <https://linux.die.net/man/5/deb-control>
- [14] DEBIAN: Repository-howto [online]. [cit. 2022-27-11]. Dostupné z: <https://www.debian.org/doc/manuals/repository-howto/repository-howto.en.html#release>
- [15] DEBIAN: DebianRepository [online]. [cit. 2022-27-11]. Dostupné z: <https://wiki.debian.org/DebianRepository>
- [16] WIKIPEDIA: Bootstrap [online]. [cit. 2023-15-5]. Dostupné z: <https://cs.wikipedia.org/wiki/Bootstrap>
- [17] HTML-FACTORY: Proč používat bootstrap [online]. [cit. 2023-15-5]. Dostupné z: <https://www.html-factory.cz/clanek/proc-pouzivat-bootstrap/>
- [18] DEBIAN: Manuals [online]. [cit. 2023-15-5]. Dostupné z: <https://www.debian.org/doc/manuals/securing-debian-manual/deb-pack-sign.en.html>
- [19] PACKAGECLOUD: GPG sign and verify deb packages and APT repositories [online]. [cit. 2023-15-5]. Dostupné z: <https://blog.packagecloud.io/how-to-gpg-sign-and-verify-deb-packages-and-apt-repositories/>
- [20] POSTGRESQL: Docs [online]. [cit. 2023-15-5]. Dostupné z: <https://www.postgresql.org/docs/9.1/sslinf.html>
- [21] WIKIPEDIA: Deb (file format) [online]. [cit. 2023-15-5]. Dostupné z: [https://en.wikipedia.org/wiki/Deb_\(file_format\)](https://en.wikipedia.org/wiki/Deb_(file_format))
- [22] NETTE: Modules [online]. [cit. 2023-15-5]. Dostupné z: <https://doc.nette.org/en/application/modules>

ZOZNAM SYMBOLOV A SKRATIEK

Skratky:

PHP	Hypertext Preprocessor
HTML	Hypertext Markup Language
DI	Dependency Injection
GPG/GnuPG	GNU Privacy Guard
CSS	Cascading Style Sheets
MVP	Model view presenter
MVC	Model view controller
SQL	Structured Query Language
UTF	Unicode Transformation Format
URL	Uniform Resource Locator
MD5	Message-digest algorithm
SHA	Secure Hash Algorithm

SEZNAM PŘÍLOH

PŘÍLOHA A - ZDROJOVÝ KÓD APLIKÁCIE.....	55
---	----

Příloha A - Zdrojový kód aplikace

Zdrojový kód aplikace je propojený samostatně k této práci (soubor priloha-A.zip).