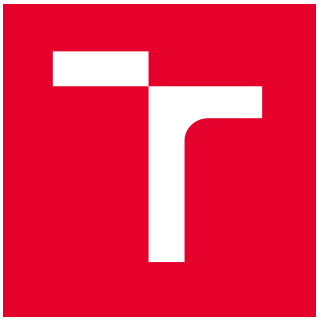


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

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 — **Aidana Kurmanova**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Petr Sysel, Ph.D.**
SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Studentka: Aidana Kurmanova ***ID:*** 227247

Ročník: 3 ***Akademický rok:*** 2023/24

NÁZEV TÉMATU:

Webové rozhraní repozitáře Debian

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se programovacím jazykem PHP a knihovnami Nette pro tvorbu webových aplikací. Na základě získaných zkušeností navrhněte a realizujte webové rozhraní repozitáře systému Debian. Vytvořený kód bude řešen modulárně, aby ho bylo možné začlenit do dalších aplikací. Aplikace by měla umožňovat základní práci s balíčky - import nového balíčku, zobrazení informací o balíčku, vytvoření seznemu balíčku Release, atd. Aplikace by měla podporovat podepsané seznamy balíčků, přihlašování heslem nebo certifikátem, vytvoření několika vydání s různými verzemi stejného balíčku.

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: 5.2.2024 ***Termín odevzdání:*** 28.5.2024

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

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

ABSTRAKT

Bakalářská práce se zaměřuje na implementaci webového rozhraní pro repozitář Debian a správu bezpečnostních certifikátů. Cílem práce je vytvořit modulární webovou aplikaci pro repozitář Debian. Aplikace repozitáře je navržena tak, aby splňovala základní funkce: importovala balíčky a zobrazovala informace o balíčcích.
Tato práce popisuje princip vytváření webových aplikací pomocí knihovny Nette, strukturu repozitáře Debian a další technologie používané při implementaci aplikace. Praktická část práce je věnována implementaci webového rozhraní a také seznámení s uživatelským rozhraním a jeho funkčností.

KLÍČOVÁ SLOVA

Webová aplikace, repository, Debian, Bootstrap, SQLIte, certifikát, podpis certifikátu, certifikační autorita, PHP, Nette Framework, autorizace.

ABSTRACT

The bachelor thesis focuses on the implementation of a web interface for the Debian repository and the management of security certificates. The aim of the work is to create a modular web application for the Debian repository. The repository application is designed to perform basic functions: import packages and display package information.
This thesis describes the principle of creating web applications using the Nette library, the structure of the Debian repository and other technologies used in the implementation of the application. The practical part of the work is devoted to the implementation of the web interface, as well as familiarization with the user interface and its functionality.

KEYWORDS

Web application, repository, Debian, Bootstrap, SQLIte, certificate, Signature, Certificate Authority, PHP, Nette Framework, authorization.

KURMANOVA, Aidana. *Webové rozhraní repozitáře Debian*. Bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2024. Vedoucí práce: doc. Ing. Petr Sysel, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Aidana Kurmanova
VUT ID autora:	227247
Typ práce:	Bakalářská práce
Akademický rok:	2023/24
Téma závěrečné práce:	Webové rozhraní repozitáře Debian

Prohlašuji, že svou závěrečnou práci jsem vypracovala samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autorky*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Ráda bych poděkovala vedoucímu bakalářské práce panu Ing. Petru Syslovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Úvod

Repozitáři softwaru obsahující binární balíčky, metadata a informace o závislostech pro instalaci a správu programů na Linuxu jsou repozitáři balíčků Linux. Každá distribuce Linuxu má své vlastní oficiální úložiště balíčků, které obsahují balíčky speciálně určené pro tuto distribuci. Použití svazku repozitáře umožňuje jednoduchou, centralizovanou metodu instalace a odinstalace programů a poskytuje pohodlný způsob, jak odesílat aktualizace.

Cílem této bakalářské práce je vytvořit webové rozhraní pro repozitář Debian pomocí platformy Nette framework. Získané teoretické znalosti byly použity k vývoji webové aplikace. Pro práci byl použit systém správy databází SQLite, byla vytvořena struktura databáze. Bylo prozkoumáno konstrukční schéma Model-View-Controller (MVC).

Cílem práce je také vývoj dříve vyvinuté aplikace pro správu bezpečnostních certifikátů.

První kapitola se zabývá teoretickými aspekty potřebnými pro vývoj aplikace: certifikáty a jejich použití, struktura repozitáře Debian, informace pro vytváření balíčku.

Druhá kapitola práce se podrobně zabývá potřebnými technologiemi: programovací jazyk PHP, Nette Framework, databáze SQLite, šablonový systém Latte a front-end framework Bootstrap.

Nakonec závěrečná kapitola slouží k prezentaci implementace aplikace, přezkoumání její struktury a prokázání výsledků.

1 Teoretická část

1.1 Certifikáty a jejich použití

Webové stránky se zabezpečeným připojením zabraňují úniku dat, zejména pokud jsou požadovány osobní údaje. Secure Socket Layer (SSL) je technologie, která poskytuje zabezpečené připojení HTTPS pomocí šifrování dat. Certifikát SSL je ve skutečnosti digitální podpis, který potvrzuje pravost webu. Použití certifikátu nám umožňuje chránit jak vlastníka webu, tak jeho zákazníky. Certifikáty jsou založeny na asymetrické kryptografii. Použijí se páry klíčů: veřejný (public key), který šifruje data, a jeho odpovídající soukromý (private key), který je dešifruje. Pokud lze veřejný klíč distribuovat bez omezení, pak soukromý klíč zůstává v tajnosti[8].

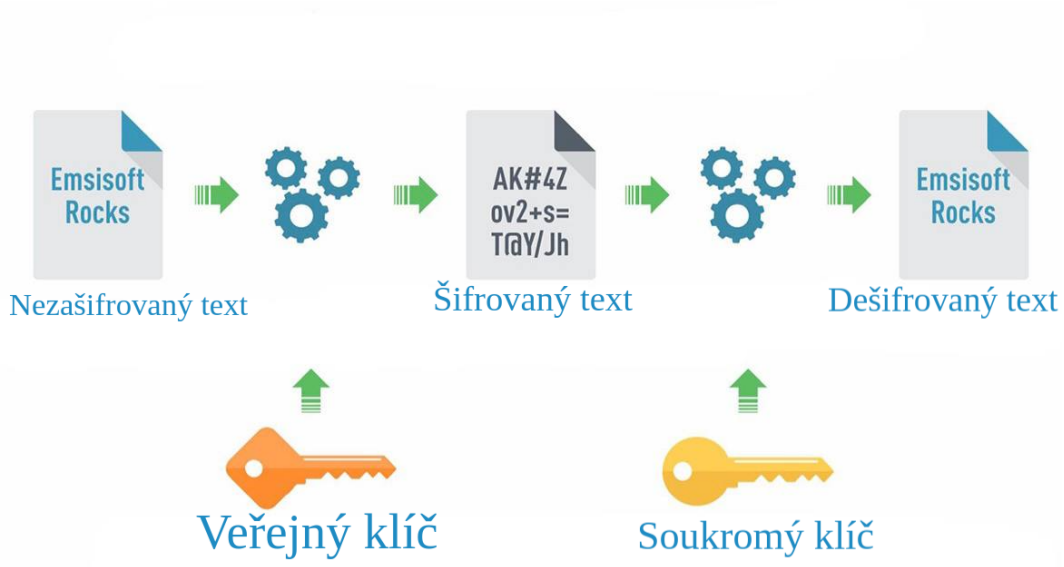
Obr. 1.1: Model elektronického podpisu

Je důležité zachovat přístup k soukromému klíči. Klíč je generován buď před procesem CSR (Certificate Signing Request) - požadavkem na vytvoření certifikátu, nebo s ním, pokud to zařízení umožňuje. Vytvoření dotazu CSR znamená vytvoření soukromého klíče: jeden požadavek na vytvoření certifikátu odpovídá jednomu požadavku na vytvoření klíče. Soukromý klíč neukládá certifikační autorita, ale samotný vlastník certifikátu. Umístění soukromého klíče závisí na druhu serveru, který je použit. Například při použití serveru Apache je soukromý klíč v hlavním konfiguračním souboru - `httpd.conf` nebo `apache2.conf`. Knihovna SSL na Apache ve výchozím nastavení ukládá klíče do adresáře `/usr/local/ssl`.

Výhodou používání asymetrických kryptosystémů je jejich poskytnutí možnosti vytváření elektronických digitálních podpisů. Digitální podpis umožňuje příjemci

zprávy ověřit autentičnost zdroje informací a ověřit, zda byly informace změněny, když byly na cestě. Digitální podpis je tedy prostředkem autentizace a kontroly integrity dat. Prostřednictvím elektronického podpisu může klient potvrdit autorství přenášeného elektronického dokumentu. Podpis je nejčastěji umístěn na otisku (hash) dokumentu, protože dokument může být často příliš velký. Hash v kryptografii se používá jako souhrn zprávy (message digest). Pro výpočet hash se používají kryptografické hashovací funkce. To zaručuje identifikaci jakýchkoliv změn v dokumentu. Elektronický digitální podpis používá protokoly jako S/MIME, PGP (Pretty Good Privacy), OpenPGP, IKE (Internet Key Exchange)[8].

Asymetrické šifrování využívající páry klíčů, veřejné a tajné, je pomalejší. Tento typ šifrování používají protokoly TLS (transport layer security) a SSL (Secure Sockets Layer). Tyto protokoly používají asymetrický šifrovací algoritmus ve fázi "handshake"- nastavení připojení. Spolehlivost šifrování závisí na délce klíče, složitosti řešení algoritmu, který je základem šifrování. Uvedené protokoly používají asymetrické šifrování pro autentizaci, zejména na internetu: webové prohlížeče, zasílání zpráv, e-mail a IP telefonie. Protokol SSL zajišťuje bezpečnost komunikace vytvořením zabezpečeného kanálu mezi klientem a serverem. Díky šifrování je kanál důvěryhodný, soukromý a ověřený[9].



Obr. 1.2: Asymetrický šifrovací model

Hlavním rozdílem mezi elektronickým podpisem a asymetrickým šifrováním je výběr klíče při šifrování a dešifrování. Pokud je při elektronickém podpisu zpráva podepsána pomocí soukromého klíče a ověření podpisu se provádí pomocí veřejného klíče, pak v asymetrickém šifrování - naopak. Zpráva je šifrována pomoci veřejného

klíče a dešifrována soukromém klíčem.

Proces ověřování se skládá z následujících kroků:

1. Uživatel zadá adresu URL serveru v prohlížeči. Požadavek klienta na webovou stránku je předán serveru.

2. Server obdrží požadavek a odešle svůj certifikát serveru klientovi.

3. Prohlížeč klienta zkontroluje, zda jeho úložiště certifikátů neobsahuje certifikát od certifikační autority, která vydala certifikát serveru.

4. Pokud je nalezen certifikát CA, prohlížeč certifikát potvrdí ověřením podpisu na certifikátu serveru pomocí veřejného klíče, který je k dispozici v certifikátu CA.

5. Pokud je ověření úspěšné, prohlížeč přijme certifikát serveru, to je považuje jej za platný.

6. Je generován symetrický šifrovací klíč, klient jej šifruje pomocí veřejného klíče serveru. Zašifrovaný klíč je vrácen serveru.

7. Server dešifruje klíč pomocí vlastního soukromého klíče serveru. Pro počítač je nyní sdílený šifrovací klíč, který lze použít k zabezpečení připojení mezi nimi.

CA - certifikační autorita přijímá žádosti o certifikát, ověřuje žádosti, vydává certifikáty a zveřejňuje aktuální stav ověření vydaných certifikátů, takže každý, kdo používá certifikát, má informace o certifikátu. Hlavním úkolem certifikační autority je ověřit pravost šifrovacích klíčů pomocí certifikátů elektronického podpisu. Jedná se o úložiště kryptografických klíčů uživatelů ve formě digitálních certifikátů, které obsahují: sériové číslo, jméno certifikační autority, datum vypršení platnosti, jméno vlastníka, jeho veřejné klíče a elektronický podpis. Certifikační autorita má právo poskytovat certifikáty veřejného klíče, přijímat a ověřovat informace a vytvářet certifikát.

1.2 Repozitář systémů Debian

Debian je distribuce Linuxu, která se skládá ze svobodného, Open Source softwaru vyvinutého komunitou Debian.

Repozitář Debianu je sada binárních nebo zdrojových Debian balíčků umístěných v přizpůsobeném stromu složek a dodávaných s různými infrastrukturními soubory. Klientské počítače se mohou připojit k repozitáři, stahovat a instalovat balíčky pomocí nástroje pro správu balíčků založeného na Apt-based PackageManagement. Účelem využití repozitáře je usnadnit vytváření prostředí. Existují jak oficiální repozitáře od vývojářů OS, tak třetích stran od jiných vývojářů. Oficiální obvykle obsahují verze základních aplikací, které se vyznačují větší stabilitou. A u třetích stran lze najít více, možná méně stabilních, ale novější verzí programů[10].

Hlavní součásti repozitáře Debian:

1. Binární balíčky (`.deb`): tyto balíčky obsahují kompilované a připravené k instalaci programy a knihovny.

2. Zdrojové balíčky: tyto balíčky obsahují zdrojový kód softwaru a soubory pro jeho sestavení.

Hlavní adresář repozitáři obsahuje `dists`, kde má každá distribuce svůj vlastní katalog. Repozitáři Debian se skládá z několika vydání, z nichž každá obsahuje několik komponent: primární ("main"), "contrib"a "non-free". Každá z těchto komponent obsahuje licenční podmínky softwaru. Repozitář je také rozdělen do podadresářích pro různé architektury ve formátu `<arch>`. Obsahují textové soubory s informacemi o metapaketech. Komponenta "components"také obsahuje původní soubory `Sources`.

Samotné balíčky jsou v podadresáři "Pool", kde se `pool` directory opět rozdělí na komponenty, z nichž každý má podadresáře zvané v těchto podadresářích a rozšiřující balíčky `.deb`. Existují také další adresáře pojmenované po indexu písmen, aby se zabránilo přetížení hlavních adresářů velkým množstvím souborů, což může v některých systémech vést ke snížení výkonu[11].

V uvedeném příkladu distribuce `stable`, `contrib` a `non-free` jsou uvedeny v katalogu `dists/`. V každé distribuci je sekce `main`, která ukládá binární balíčky. Adresář `pool/`obsahuje binární balíčky seskupené podle jejich názvů. Každý adresář balíčků v části `pool/main/` obsahuje skutečné soubory `.deb` soubory spolu se soubory kontrolního součtu.

```
/.....................................................kořenový adresář repozitáře
├── dists
│   ├── stable
│   │   └── main
│   │       ├── binary-amd4
│   │       │   ├── Packages
│   │       │   ├── Packages.gz
│   │       │   ├── Release
│   │       │   └── Release.gz
│   │       └── binary-...
│   ├── contrib
│   │   └── main
│   │       ├── binary-amd4
│   │       ├── binary-armhf
│   │       └── binary-...
│   └── non-free
│       └── main
│           ├── binary-amd4
│           ├── binary-armhf
│           └── binary-...
└── pool
    └── main
        └── package/
```

Adresář `dists` obsahuje většinu metadat, včetně všech souborů packages.gz (které obsahují seznam balíčků) a Release.gpg / Release, který podepisuje balíček.

Adresář `pool` obsahuje skutečné soubory `.deb`. Příkladem organizace je `/pool/[section]/[letter]/[group]/packagename.deb`. Tedy skutečné umístění balíčku je `/pool/main/z/zlib1g-dbg/zlib1-dbg_amd64.deb`, protože subversion je skupina balíčku zlib1g-dbg-subversion a `z` je první písmeno subversion.

Vytváření a správa repozitáře se zjednodušuje pomocí různých nástrojů. Pomocí těchto nástrojů je možné vytvořit a spravovat Debian, včetně přidávání nových balíčků, aktualizace stávajících a udržování Relevance indexů[17].

1. `dpkg`: toto je hlavní nástroj pro správu balíčků v Debianu. Používá se k instalaci, mazání a správě balíčků `.deb`.

2. `Apt`: je to nástroj pro práci s balíčky Debian. `Apt` nám umožňuje spravovat závislosti, aktualizovat balíčky a hledat nové balíčky k instalaci.

3. `dpkg-scanpackages`: tento nástroj se používá ke skenování adresáře s balíčky a vytváření indexového souboru `Packages.gz`, který obsahuje informace o všech dostupných balíčcích.

4. `dpkg-scansources`: podobně jako `dpkg-scanpackages` tento nástroj prohledá adresář s původními soubory balíčků a vytvoří indexový soubor `Sources.gz` obsahující informace o dostupných zdrojových balíčcích.

5. `GnuPG`: k zabezpečení repozitáře a podpisu metadat je nutné použít `GnuPG` k vytvoření a ověření digitálního podpisu balíčků a indexových souborů.

6. `reprepro`: je to nástroj, který usnadňuje správu repozitáře Debian. Poskytuje uživatelsky přívětivé rozhraní pro přidávání, mazání a aktualizaci balíčků v úložišti a vytváření indexů.

Vytvoření repozitáře Debian zahrnuje klíčové kroky, jako jsou:

1. Příprava balíčků: balíčky musí být zkompilovány a zabaleny do formátu `.deb` splňující standardy Debian.

2. Vytváření indexových souborů: pomocí nástrojů `dpkg-scanpackages` a `dpkg-scansources` se vytvářejí soubory `Packages.gz` a `Sources.gz`, které obsahují metadata o balíčcích.

3. Podpis úložiště: pro zajištění bezpečnosti je úložiště podepsáno pomocí `GnuPG`. To umožňuje uživatelům ověřit pravost balíčků.

4. Uspořádání struktury adresářů: balíčky a indexové soubory jsou umístěny v příslušných podadresářích podle výše uvedené struktury.

Pro správu repozitáře Debian je třeba provest následující úkoly[17]:

1. Aktualizace balíčků: pravidelné přidávání nových verzí balíčků a mazání starších verzí.

2. Udržování bezpečnosti: zajištění bezpečnosti balíčků pravidelnou aktualizací digitálních podpisů a kontrolou přístupu k úložišti. Jak již bylo zmíněno, soubory

v repozitáři Debian se podepisují pomocí digitálních klíčů vytvořených pomocí nástrojů, jako je GPG. Každý balíček a metadata úložiště jsou podepsána pomocí soukromého klíče a uživatelé pak mohou ověřit pravost těchto souborů pomocí příslušného veřejného klíče.

3. Monitorování a podpora: sledování stavu repozitáře, zpracování zpětné vazby od uživatelů a řešení problémů, které se objeví. Při instalaci nebo aktualizaci paketů z repozitáře Debian kontrola integritu paketů je nutná.

4. Dokumentace: je nutné poskytnout uživatelům podrobnou dokumentaci o úložišti, včetně pokynů k instalaci, použití a zabezpečení.

2 Použité technologie

2.1 Jazyk PHP

PHP je programovací jazyk, který je nejběžnější v oblasti vývoje webu. PHP se provádí na serveru. Programy napsané v PHP přijímají data od uživatelů webu, zpracovávají je, komunikují s databázemi a poté na web vracejí zpracované informace. Hlavním důvodem pro použití jazyka PHP v tomto projektu je orientace jazyka na práci se serverovými databázemi, jako je sqlite na webovém serveru.

V této práci používáme verzi 8.0 jazyka PHP.

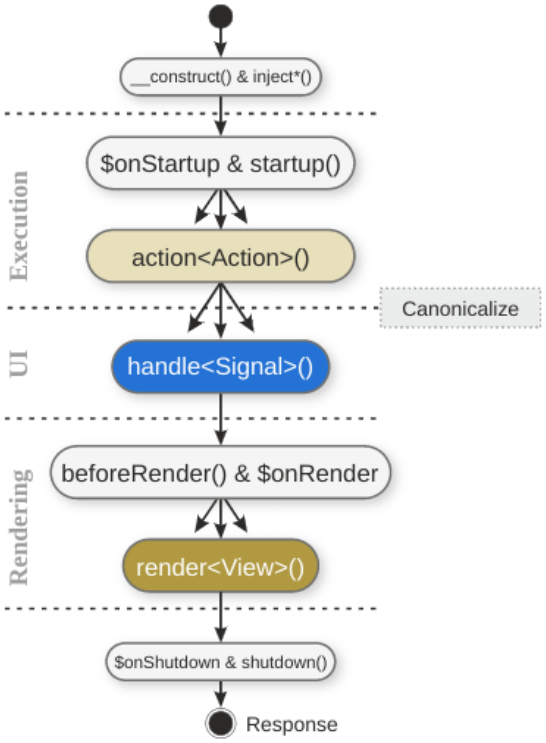
2.2 Objektový model MVC

Knihovna nám umožňuje použít návrh vzoru MVC (Model-View-Presenter). Jedná se o tři různé úrovně práce: Model je spojen pouze s Prezenterem, pracuje s daty aplikace, Pohled (View) je zodpovědný za návrh aplikace pomocí šablon, Prezenter se projevuje úrovní mezi nimi: odesílá požadavky na úroveň modelu a odešle odpověď na úroveň pohledu.

Šablona Latte poskytuje zabezpečení aplikaci, takže model certifikátu je v bezpečném adresáři: SecureModel. Model certifikátu obsahuje metody, které umožňují práci se samotným certifikátem: výstup informací o vlastnostech, jako je název nebo vydavatel, stažení certifikátu, jeho podepsání a vytvoření. Funkce jsou psány v jazyce PHP pomocí knihovny Nette a její třídy *SmartObject*, která rozšiřuje schopnosti jazyka PHP. Při psaní modelu certifikátu se používají hlavní funkce OpenSSL v jazyce PHP: `openssl-x509-parse` umožňuje získat z certifikátu pole s daty o něm, `openssl-cms-sign` umožňuje podepsat certifikát, `openssl-get-privatekey` slouží k získání soukromého klíče. Hlavním úkolem prezentera je předat jakékoli požadavky, například přesměrování nebo HTML stránka, a předat odpověď na požadavek[3].

Obrázek 2.1 ukazuje metody, kterými lze vyvolat shora dolů, pokud existují[7]. Na začátku je tedy proveden konstruktor, který slouží k vytvoření závislosti databáze a serveru. Následující metoda se používá ke kontrole uživatelských práv a inicializaci vlastností. Metoda `action<Action>(args...)` připravuje data na šablonu, která je dále vykresluje. Metoda `handle<Signal>(args...)` zpracovává signály-interaktivní komponenty aplikace. Dále metoda `beforerender ()`, která slouží ke konfiguraci šablony. Následující metoda přenáší data pro vykreslení. A provádí se konec cyklu prezentátora.

Pro aktuální informace o databázi na serveru slouží třída modelu *CertificateFacade*, obsahuje výstupní funkci aktuálních databázových záznamů a konstruktor pro



Obr. 2.1: Životní cyklus Prezentera

komunikaci databáze a serveru. Současně v šabloně webové stránky se záznamy databáze vytvoříme závislost s třídou *CertificateFacade*, čímž také zaregistrujeme třídu v kontejneru DI. Aby tato třída mohla aktualizovat data, musí se stát službou.
V konfiguračním souboru `services.neon` přidáme sekci Služby, v ní popíšeme cestu k naší třídě. Třída *CertificateFacade* je zaregistrována v kontejneru DI, zatímco kontejner vytvoří instanci a předá ji třídě sloužící pro zobrazení webové stránky. Model *PackageManager* slouží ke správě balíčků a také k nastavení databáze. Zde jsou inicializovány tabulky potřebné v databázi, stejně jako funkce správy balíčků, jako je odstranění paketu ze struktur Packages, pool, vytváření indexových souborů.

2.3 Knihovna Nette

Knihovna Nette představuje základ pro vytváření interaktivních webových aplikací. Knihovna používá jazyk PHP. Nette Foundation je developerem knihovny ve spolumajitelství s českým programátorem a původním autorem - Davidem Grudlem. Důvody použití této konkrétní knihovny jsou lehké psaní kódu, komponenta MVC (Model-View-Controller), díky níž je snadné vytvořit objektový model aplikace, podpora jazyka PHP, serializace a konfigurace dat ve formátu NEON, šablonovací sys-

tém Latte, pro zjednodušenou práci s třídou Presenter.

Knihovna nabízí velký systém šablon, bohatý katalog nástrojů pro nastavení, ochranu před zranitelnostmi a efektivní vrstvu databáze. Jedná se o rámec nové generace s podporou HTML, kvalitní dokumentací a uživatelsky přívětivým designem.

V práci používáme aktuální verzi Nette-4.0. Tato verze je nejvhodnější pro verzi 8.0 PHP.

2.4 Šablonovací systém Latte

Šablonovací systém Latte je zaměřen na jazyk PHP pro účinnou ochranu proti zranitelnosti, jako je Cross-Site Scripting (XSS). Se zranitelností XSS může útočník přidat nativní kód na dynamickou webovou stránku. Tímto způsobem je útočník schopen obejít omezení vstupu a získat přístup k datům. Tato chyba zabezpečení může být použita ve phishingu, nahrazení webové stránky důvěryhodnou.

Latte je šablonový systém nové generace, rozumí struktuře prvků a jazyku HTML. Rozpoznává značky, atributy, různé jednotlivé atributy a zpracovává data. Tento systém poskytuje ochranu před zranitelnostmi skriptování mezi weby. Latte používá mechanismy dědičnosti, které mohou znovu použít opakující se struktury, čímž zvyšují výkon práce. Dědičnost tříd a objektů obvykle komplikuje práci, ale v případě Latte funguje definice dědičnosti vzorů snadno a dokonale. Je dokonce možné použít víceúrovňové dědičnosti.

Latte zahrnuje typový systém, který má velký význam ve vývoji aplikací. Deklarace typu parametrů, proměnných a tříd usnadňuje psaní kódu.

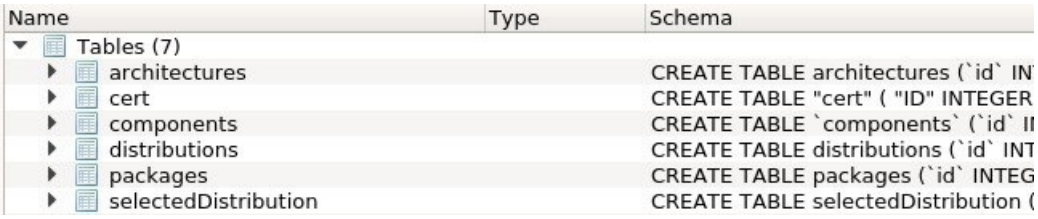
Aby byl zajištěn omezený přístup k filtrům, funkcím a metodám, používá Latte režim “Sandbox”. Tento režim sleduje přístup k prostředkům aplikace a určuje možnosti šablony. Následující obrázek ukazuje příklad šablony Latte.

```
{block content}

<p><a n:href="Homepage:default">← Back to list of certificates</a></p>
<a n:href="Edit:edit $certificate->id">Edit certificate</a>

{block title}<h1>{$certificate->CN}</h1>{/block}

<div class="certificate">{$certificate->Issuer}</div>
<div class="certificate">{$certificate->Valid From}</div>
<div class="certificate">{$certificate->Valid Until}</div>
<div class="certificate">{$certificate->Authority}</div>
<div class="certificate">{$certificate->Private Key}</div>
```

Obr. 2.2: Příklad šablony Latte na stránce aplikace

2.5 Databáze SQLite

Pro lepší výkon a dostupnost se používá SQLite. Jedná se o velmi snadno použitelnou databázi, kde jsou operace čtení a zápisu prováděny o 35% rychleji než v systému souborů. Databázi není nutné instalovat nebo konfigurovat, stačí nainstalovat knihovny SQLite na osobní počítač. Pro práci s databází byl také použit prohlížeč - DB Browser for SQLite. Virtuální nástroj nám umožňuje vytvářet databázový soubor, vytvářet a upravovat tabulky, upravovat a přidávat záznamy, importovat a exportovat databázové soubory. Databáze je přenosná. Několik procesů může být spojeno se stejným souborem aplikace, mohou číst a psát, aniž by se navzájem rušily. Databázi SQLite lze také používat se všemi programovacími jazyky bez problémů s kompatibilitou.



Obr. 2.3: DB Browser for SQLite

Pro připojení databáze k webovému serveru jsme použili konfiguraci Neon. Konfigurační soubor uvádí cestu k souboru, přihlašovací jméno a heslo databáze. Adresář `/config/` tedy obsahuje celou konfiguraci databáze. Aby webová stránka zobrazovala data naší databáze, vytvoříme na úrovni prezentera soubor PHP, obsahující konstruktor, který umožňuje výstup dat okamžitě na webovou stránku, stejně jako metodu pro odesílání dat do šablony. Tato šablona zase změní data do kódu HTML.

Databáze byla použita jako implementace modelů repozitářů Debian a bezpečnostních certifikátů. Struktura databáze, kterou jsme vytvořili, je znázorněna na obrázku 2.3[19].

2.6 Bootstrap

Framework Bootstrap je sada nástrojů pro vytváření a práci s webovými aplikacemi a weby v oblasti HTML, CSS, Javascript. Obsahuje šablony formulářů, navigační bloky a další součásti webového rozhraní. V této práci byl framework použit jako šablona pro prezentační část naší aplikace. Například navigační panel je pevný nástroj pro návrh. Framework lze také použít k navrhování dialogových oken, tabulek, médií a jako nástroj pro popis písem[20].

3 Implementace webové aplikace

3.1 Struktura aplikace

3.1.1 Databáze

Pro správu dat byla v rámci projektu vytvořena jednotná databáze. Popis každé tabulky obsažené v databázi je uveden v následující tabulce.

Názvy tabulek	**Popis tabulek**
architectures	informace o počítačové platformě, na které chce uživatel vytvořit Debian
cert	informace o parametrech bezpečnostních certifikátů
components	informace o komponentách, které slouží k ukládání balíčků s plně volnými licencemi odděleně od ostatních.
distributions	názvy releasů, do kterých lze balíčky vložit
packages	informace o parametrech balíčku
selectedDistrubutions	dynamické ukládání názvů vydaných verzi

3.1.2 Moduly

Webová aplikace je rozdělena do dvou modulů: Debian a Secure, pro snadnou správu, jelikož projekt má dvě oblasti pro správu. Tyto moduly, které fungují jako jádro aplikace, zahrnují modely, prezentory a šablony.

V adresáři `DebianModule/Model/` byly vytvořeny modelové třídy:

1. *ArchitectureManager.php* model vytváří tabulku `architectures` v databázi a slouží ke správě použitých architektur;

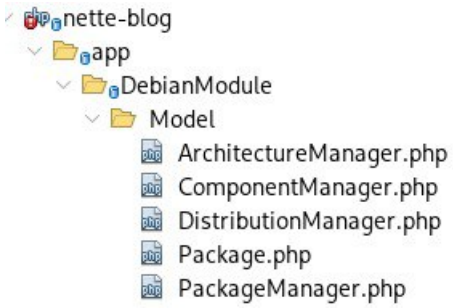
2. *ComponentManager.php* generuje tabulku `components` v databázi a správuje komponenty;

3. *DistributionManager.php* inicializuje tabulky `distributions`, `selectedDistribution` a spravuje `distributory`. V rámci tohoto modelu jsou definovány funkce pro přidání a odebrání distribuce, přidání balíčku do distribuce a vytvoření adresáře dists s potřebnými podadresáři;

4. *Package.php* je pro informace o balíčcích a vytváření indexových souborů obsahuje následující funkce: rozbalení balíčku `.deb` (*load*), extrahování informací

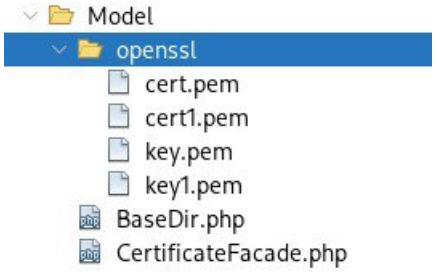
ze souboru control (*parse*), generování indexových souborů (*renderReleaseFile*) a generování indexových souborů s názvem distribuce (*releaseData*). Kromě toho model obsahuje tři hlavní metody související s verzemi v podadresářích: *signRelease* pro podpis souboru vydání, *signPackage* pro podpis konkrétních balíčků a *gzCompressFile* pro kompresi velikostí paketů;

5. *PackageManager.php* - je model pro správu balíčků. Generuje tabulku `packages` pro databázi. Díky popsaným funkcím lze balíček přidat do tabulky, odstranit balíček z databáze, vygenerovat, odstranit a obnovit indexové soubory a vytvořit adresář `pool` s potřebnými podadresáři.



Obr. 3.1: Modely v modulu Debian

V adresáři `SecureModule/Model/` byly vytvořeny modelové třídy: *BaseDir.php*, *CertificateFacade.php* a `openssl` adresář, který slouží k ukládání certifikátů a soukromých klíčů.



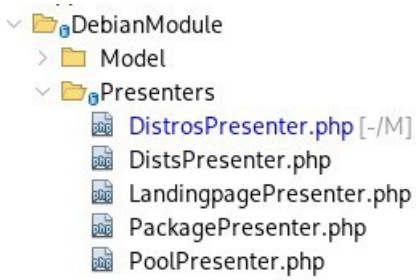
Obr. 3.2: Modely v modulu Secure

CertificateFacade.php slouží pro aktuální informace o certifikátech v databázi na serveru. Obsahuje výstupní funkci aktuálních databázových záznamů a konstruktor pro komunikaci databáze a serveru. Současně v šabloně webové stránky se záznamy databáze vytvoříme závislost s třídou *CertificateFacade*, čímž také zaregistrujeme třídu v kontejneru DI. Aby tato třída mohla aktualizovat data, musí se stát službou. V konfiguračním souboru services.neon přidáme sekci Služby, v ní popíšeme cestu

k třídě. Třída *CertificateFacade* je zaregistrována v kontejneru DI, zatímco kontejner vytvoří instanci a předá ji třídě sloužící pro zobrazení webové stránky.

3.1.3 Presentery

V procesu implementace aplikace byly vytvořeny presentery, které zajišťují komunikaci s modelem a interakci s uživatelem. Jsou také odděleny dvěma moduly: Debian a Secure.



Obr. 3.3: Prezentery modulu Debian

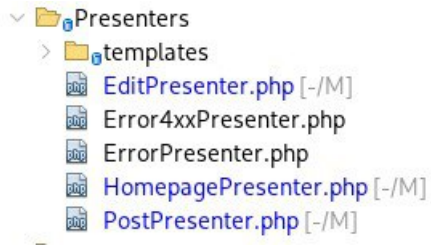
Modul Debian zahrnuje následující presentery:

1. *DistrosPresenter*: Tento presenter získává data, která budou následně zobrazena ve výchozím šabloně default.latte, a obsahuje metodu pro vytvoření adresáře `dists`.
2. *HomepagePresenter*: presenter se stará o zobrazení domovské stránky aplikace.
3. *DistsPresenter*: presenter slouží k načítání souborů.
4. *PoolPresenter*: Tento presenter určuje cílovou cestu pro načtení balíčku.
5. *LandingpagePresenter*: presenter slouží k vykreslení šablony default.latte.
6. *PackagePresenter*: Tento presenter využívá data z modelů Package a PackageManager a distribuje je do šablony default.latte. Metoda `loadPackage` umožňuje získat informace o balíčku a předat je metodě `addPackage`, která umožňuje přidat informace do repozitáře.

Modul Secure obsahuje následující prezentery:

1. *EditPresenter.php*: umožňuje upravovat data certifikátu. V tomto prezenteru jsou oznámeny funkce `createComponentCertificateForm`, které umožní vytvořit nové možnosti; `renderEdit`, který je navržen tak, aby hledal certifikát podle čísla ID; a `CertificateFormSucceeded`, který ukládá nové parametry do databáze.
2. *Error4xxPresenter.php* a *ErrorPresenter.php*: tyto prezentery se používají ke zpracování a zobrazení chyb.
3. *HomepagePresenter.php*: poskytuje domovskou stránku, která může obsahovat prvních 10 hodnot tabulky `cert`.

4. *PostPresenter.php*: tento prezenter je schopen prokázat samostatný certifikát požadované hodnoty ID. Zde je popsána funkce `renderShow`, která umožňuje komunikaci s tabulkou `cert` a výstup hodnoty ID.



Obr. 3.4: Prezentery modulu Secure

3.1.4 Šablony

Adresář šablon obsahuje podadresáře pro jednotlivé prezentátory a sdílenou šablonu *@layout.latte*, která se zobrazuje na všech stránkách celé aplikace.

Adresář templates pod modulem Debian obsahuje:

1. *Distros* obsahuje dvě šablony: default.latte pro zobrazení forem distribuce a také distsSelect.latte pro zobrazení forem příloh balíčků v distribucích;

2. *LandingPage* zobrazuje podadresáře repozitáře;

3. *Package* slouží k zobrazení podrobnosti balíčků a vykreslení tabulek dostupných balíčků;

4. *Dists* je zodpovědný za vykreslování podadresářů a indexových souborů adresáře `dists`;

5. *Pool* je zodpovědný za vykreslování podadresářů a indexových souborů adresáře `pool`.

Adresář templates pod modulem Secure obsahuje:

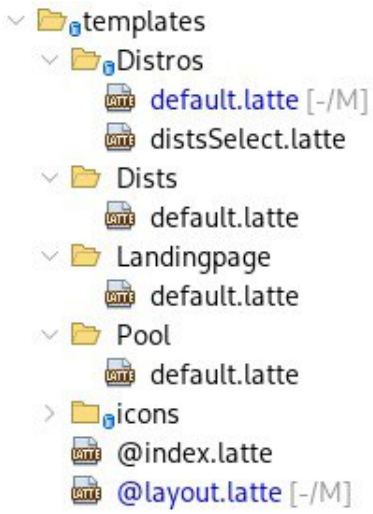
1. *Certificate* zobrazuje data tabulky certifikátů seřazená podle parametrů;

2. *Edit* obsahuje dvě šablony: create.latte, který umožní vytvořit nový certifikát a uložit jej do databáze, a edit.latte pro úpravu určitých parametrů certifikátu;

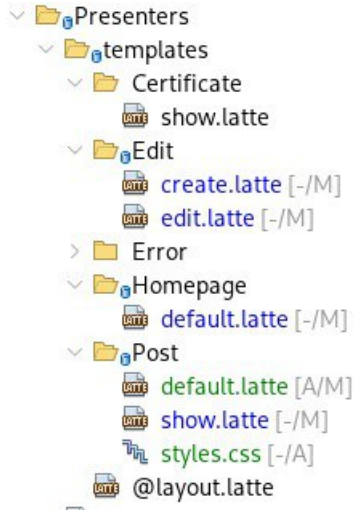
3. *Homepage* je zodpovědný za zobrazení domovské stránky;

4. *Post* obsahuje dvě šablony: default.latte a show.latte, stejně jako CSS-styl. Obě šablony plní funkci zobrazení konkrétního certifikátu;

5. *@layout.latte* je běžná šablona pro aplikaci. Obsahuje navigační panel Bootstrap, který se zobrazuje na všech stránkách webové aplikace.



Obr. 3.5: Šablony adresáře templates v modulu Debian

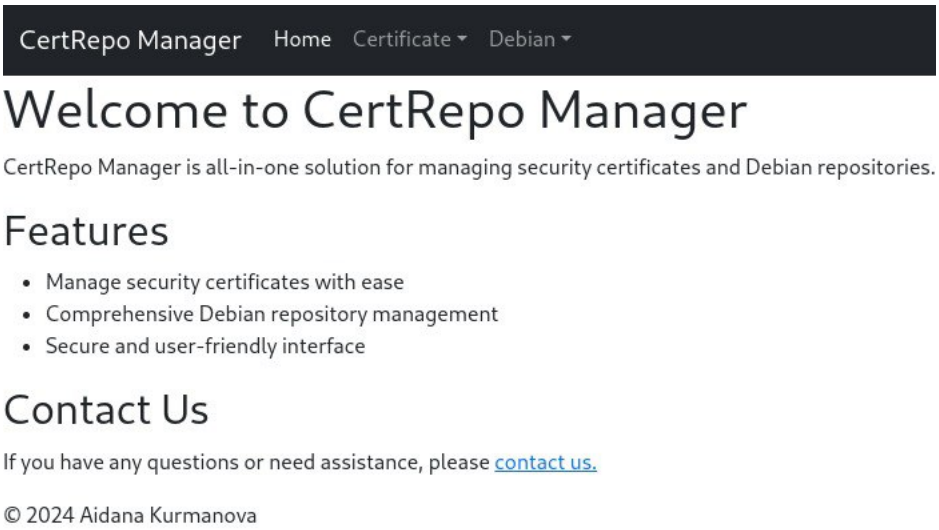


Obr. 3.6: Šablony adresáře templates v modulu Secure

3.2 Uživatelské rozhraní

Spuštěním aplikace pomocí příkazu `php -s localhost:8000 -t www` v adresáři `nette-blog` se uživateli zobrazí domovská stránka. Tato stránka podrobně popisuje funkce aplikace, uvádí kontaktní e-mailovou adresu pro zpětnou vazbu a poskytuje navigační panel. Tlačítko `Home` umožní vrátit se na domovskou stránku z jakékoli části aplikace. Rozbalovací nabídky `Certificate` a `Debian` nám poskytují možnost manipulovat s bezpečnostními certifikáty a repozitáře Debian.

Při přechodu do okna `Add Certificate` má uživatel možnost přidat nastavení certifikátu do formulářů webové stránky. Kliknutím na tlačítko `Save and Submit`



Obr. 3.7: Zobrazení Bootstrap na webové stránce

se data uloží do databáze a zobrazí se ve webové aplikaci.

1. *Common Name*: Toto pole označuje název domény nebo název zdroje, pro který byl certifikát vydán.

2. *Issuer*: Je to organizace nebo entita, která vydala certifikát.

3. *Valid From* (datum zahájení akce): označuje datum a čas, ze kterého se certifikát stává platným. *Valid Until* (Datum ukončení platnosti): označuje datum a čas, do kdy je certifikát platný.

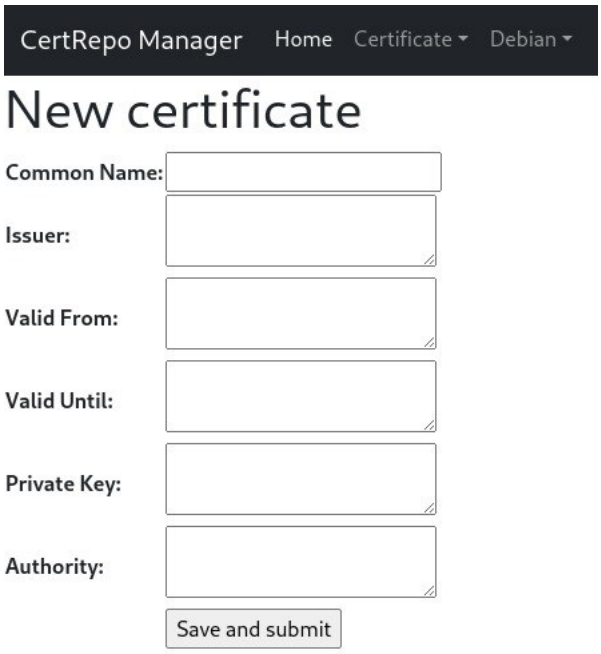
5. *Private Key*: Jedná se o tajný kryptografický klíč spojený s certifikátem, který se používá k dešifrování dat zašifrovaných pomocí veřejného klíče.

6. *Authority*: Jedná se o důvěryhodný orgán, který vydává certifikáty po ověření subjektu, který o certifikát žádá.

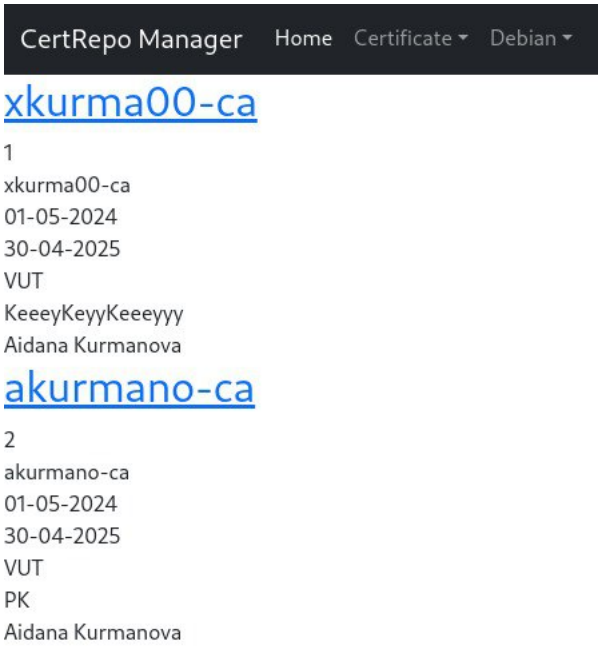
Když přejdete do okna `Show Certificates`, zobrazí se data tabulky `cert` naší databáze. Velkým písmem jsou zvýrazněna pole `Common Name` certifikátu, následují jeho ID, vydavatel, datum zahájení a ukončení platnosti, certifikační autorita a klíč. Po kliknutí na `Common Name` se uživatel přesune na stránku samostatného certifikátu, kde je k dispozici možnost upravovat data, přidaní nového certifikátu nebo se vrátit do obecného seznamu.

Po kliknutí na stránku `Debian Repositories` je uživatel přesměrován na webové rozhraní, které napodobuje standardní strukturu úložiště Debian. Stránka úložiště obsahuje hlavní adresáře dists a pool.

Pro vytvoření nové distribuce přejděte na stránku `Distribuce` kliknutím na stejnojmenné tlačítko v navigačním panelu. Na této stránce můžete vytvořit novou distribuci nebo upravit a odstranit stávající distribuce. Pro vytvoření nové distribuce

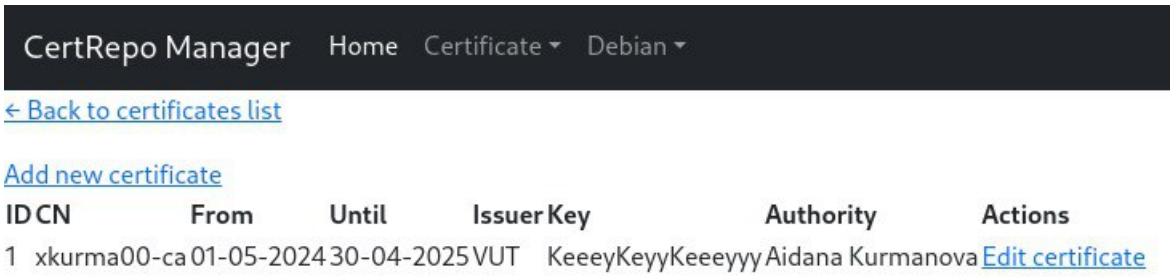


Obr. 3.8: Možnost vytvoření certifikátu na webu

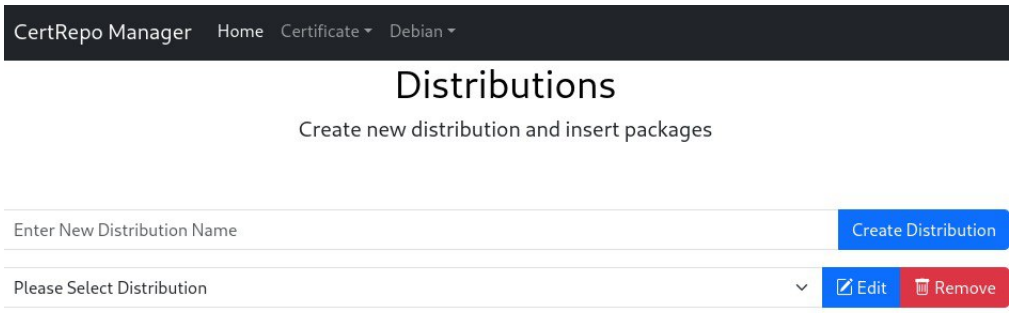


Obr. 3.9: Ukázka stávajících certifikátů

je třeba zadat její název a kliknout na tlačítko `Create Distribution`. V menu je možné vybrat jednu z existujících distribucí. Tlačítko `Edit` slouží k úpravě vybrané distribuce a tlačítko `Remove` k jejímu odstranění.

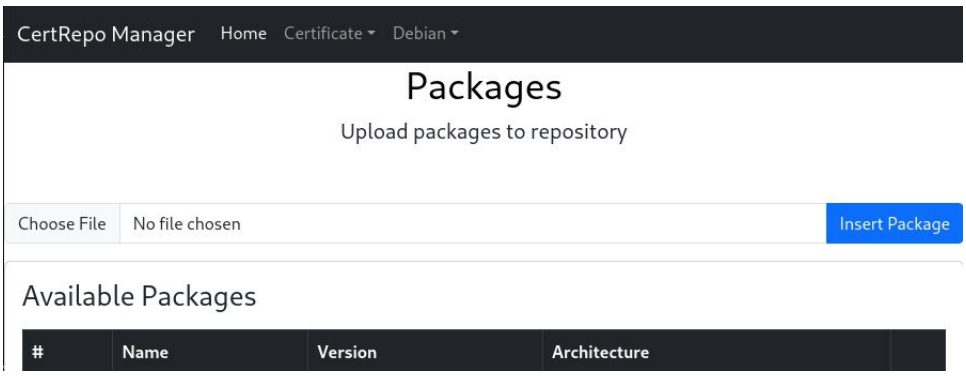


Obr. 3.10: Úprava charakteristik certifikátu



Obr. 3.11: Možnost vytvoření distribuce

Pro přidání balíčku do repozitáře na navigačním panelu vyberte záložku `Debian` a klikněte na tlačítko `Packages`. Po přesměrování na příslušnou stránku vyplňte formulář `uploadForm` a nahrajte balíčky ve formátu `.deb`. Po úspěšném nahrání se balíček zobrazí v tabulce s názvem `Available packages`. Tabulka obsahuje základní informace o balíčku, včetně jeho názvu, verze a architektury.

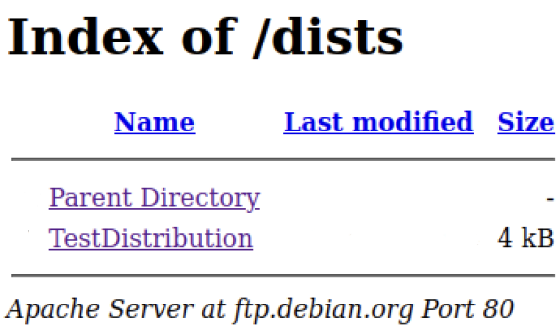


Obr. 3.12: Možnost nahrát balíček

Vybráním možnosti úpravy distribuce `Edit` je možné přidat balíček k distribuci.

Obr. 3.13: Vkládání balíčků do distribuce

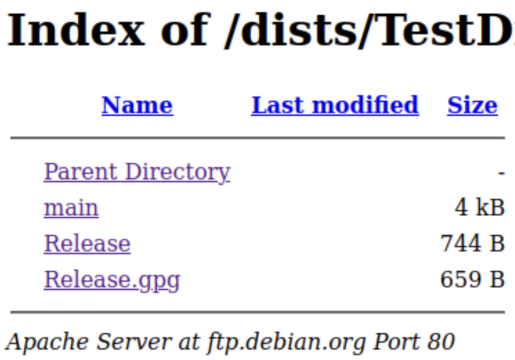
Adresář `dists` obsahuje názvy všech vytvořených distribucí. Podadresář s názvem distribuce obsahuje hlavní složku a také soubory vydání, včetně *Release* a *Release.gpg*, což je podpis souboru vydání.



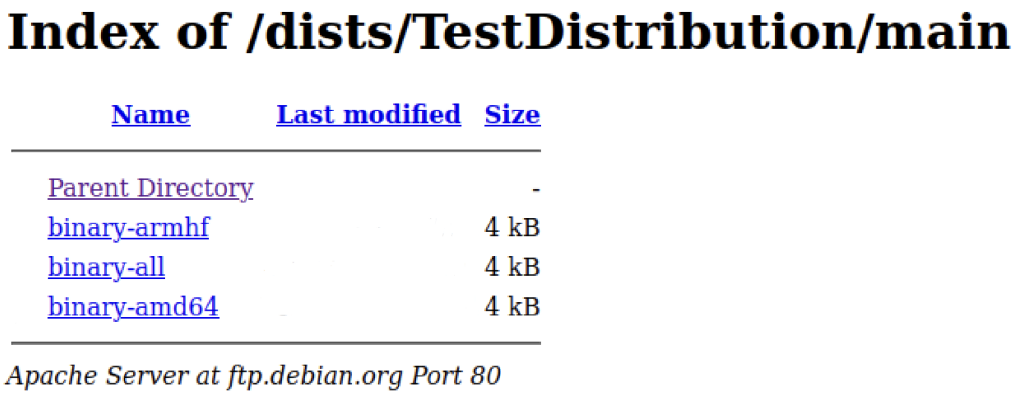
Obr. 3.14: Zobrazení adresáře /dists

Podadresář hlavní součásti (main) obsahuje seznam podporovaných architektur, pro které byly balíčky zkompilovány. Stahovatelné binární indexy balíčků a soubory indexu *Release* jsou uloženy v podadresáři této architektury. Při vytváření souboru *Release* je také vytvořen jeho digitální podpis *Release.gpg*, které je generováno pomocí kryptografických algoritmů. To umožňuje ověření a integritu souboru Release při jeho stažení z úložiště Debian. Soubory balíčků lze také stáhnout v komprimované podobě.

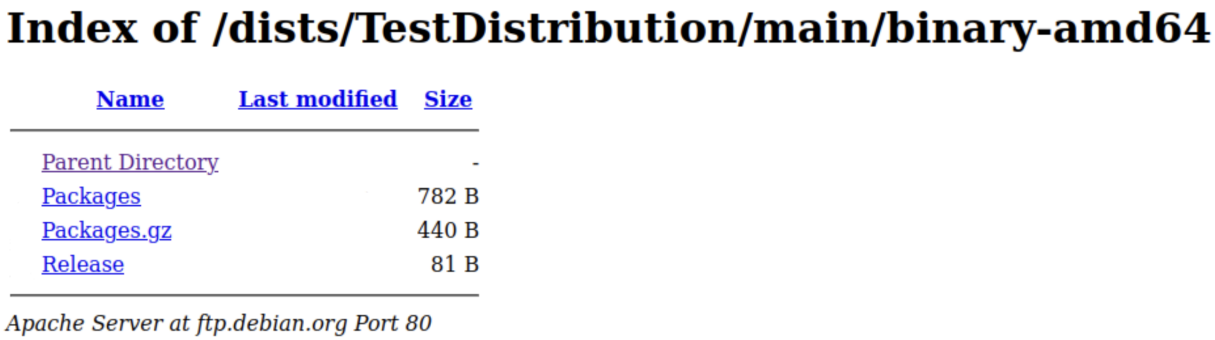
Adresář `pool` je rozdělen na podadresář s názvem `main`, stejně jako podadresáře



Obr. 3.15: Zobrazení adresáře /dists/TestDistribution



Obr. 3.16: Zobrazení adresáře /main

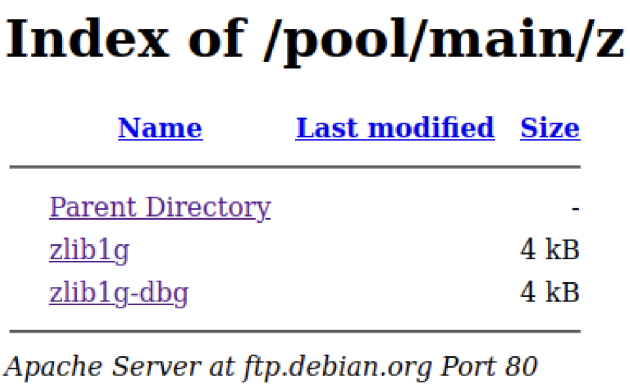


Obr. 3.17: Zobrazení balíčků v /dists

s počátečními písmeny každého balíčku a názvy samotných balíčků. V podadresáři s názvem balíčku je umístěn samotný balíček. Nejprve v adresáři `pool` uvidíme

názvy podadresář s jedním písmenem. Tímto písmenem začíná název samostatného balíčku.

Tyto podadresáře obsahují vestavěné rozšiřující balíčky `.deb`. Níže je uveden příklad struktury katalogu `pool` a jeho podadresářů.

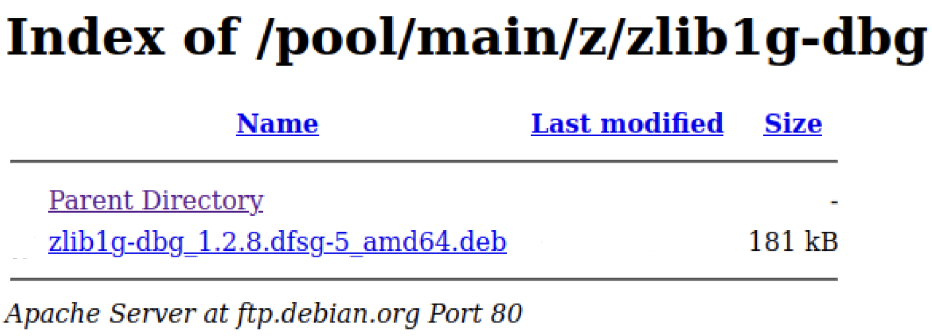


Index of /pool/main/z

Name	Last modified	Size
Parent Directory		-
zlib1g		4 kB
zlib1g-dbg		4 kB

Apache Server at ftp.debian.org Port 80

Obr. 3.18: Zobrazení adresáře /pool



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

Name	Last modified	Size
Parent Directory		-
zlib1g-dbg_1.2.8.dfsg-5_amd64.deb		181 kB

Apache Server at ftp.debian.org Port 80

Obr. 3.19: Zobrazení balíčků v /pool

Závěr

Cílem bakalářské práce bylo vytvořit webové rozhraní pro repozitář Debian. Kromě toho tato práce zahrnuje vývoj webové aplikace s funkcemi správy bezpečnostních certifikátů, které zahrnují vytváření, ukládání do databáze, operace modifikace a mazání. K realizaci tohoto cíle bylo zamýšleno použít programovací jazyk PHP jako hlavní vývojový nástroj a také framework Nette, aby byla zajištěna efektivní a strukturovaná implementace funkčnosti aplikace.

Pro vývoj byl prozkoumán repozitář linuxových distribucí, možnosti rozbalování balíčků, struktura a vytváření balíčků. Podívali jsme se na bezpečnostní certifikáty, jejich podpis a koncept certifikační autority. Byly analyzovány výhody knihoven pro psaní aplikací, jako je Latte Templated System a jazyk NEON určený k serializaci dat. Byly také studovány základní funkce OpenSSL v programovacím jazyce PHP a databáze SQLite.

Získané teoretické znalosti byly použity k vývoji webové aplikace. Byla vytvořena struktura databáze SQLite pro ukládání informací o distribučních balíčcích a certifikátech. Interaktivní komponenty byly navrženy tak, aby udržely záznamy v databázi aktuální. Byly vytvořeny webové stránky pro uživatelské rozhraní.

Vytvořená aplikace obsahuje domovskou stránku a navigační panel, který může uživatele přesměrovat na stránky správy bezpečnostních certifikátů a správy úložiště Debian.

Literatura

[1] *Aplikovaná kryptografie - Laboratorní cvičení, Ing. Lukáš Malina, Ph.D., Ing. Marek Sikora.* 54 stran. Brno: Fakulta elektrotechniky a komunikačních technologií, VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, 2019. Dostupné z URL: <https://moodle.vut.cz/pluginfile.php/173454/mod_page/content/17/TAKR_skripta_labiny_2019.pdf?time=1573665612606/>

[2] *The Joy of Cryptography, Mike Rosulek.* 286 stran. Corvallis, Oregon: Oregon State University, 2021. Dostupné z URL: <https://joyofcryptography.com/>

[3] *Configuring Nette, Nette documentation [online], David Grudl. Dostupné z URL:* <https://doc.nette.org/en/configuring>, 2022.

[4] *PHP documentation, Cryptography Extensions: OpenSSL [online], the PHP Group. Dostupné z URL:* <https://www.php.net/manual/en/intro.openssl.php>, 2021-2022.

[5] *Cryptography, digital certificates, PKI, "Methods and means of protecting computer information"[online], V.S. Solomykov. Dostupné z URL:* <http://www.msiit.ru/x/miszki/index.html>, 2019.

[6] *SSL Complete Guide - HTTP to HTTPS, Bogdan Stashchuk.* ISBN: 1839211504, 9781839211508, Packt Publishing, 2019.

[7] *Presentery Nette, Nette documentation [online], David Grudl. Dostupné z URL:* <https://doc.nette.org/cs/application/presenters>, 2022.

[8] *Jak funguje elektronický podpis [online], Astral. Dostupné z URL:* <https://astral.ru/info/elektronnaya-podpis/obshchie-voprosy/kak-rabotaet-elektronnaya-podpis/>, 2022.

[9] *Symetrické a asymetrické šifrování: jen o komplikovaném [online], Otus. Dostupné z URL:* <https://otus.ru/nest/post/726/>, 2022.

[10] *DebianRepository, Wiki [online]. Dostupné z URL:* <https://wiki.debian.org/DebianRepository>, 2022.

[11] *The Debian GNU/Linux FAQ, [online], Debian documentation. Dostupné z URL:* <https://www.debian.org/doc/manuals/debian-faq/index.en.html>, 2022.

[12] *Přečtěte si o repozitáři balíčků Linuxu, [online]. Dostupné z URL:* <https://pq.hosting/help/instructions/402-rasskazyvaem-o-repozitorii-paketov-linux.html>, 2023.

[13] *Get started with Bootstrap, [online], Bootstrap documentation. Dostupné z URL:* <https://getbootstrap.com/docs/5.3/getting-started/introduction/>, 2023.

[14] *deb-control(5) - Linux man page, [online], Linux-die-net. Dostupné z URL:* <https://linux.die.net/man/5/deb-control>, 2022.

[15] *Debian Packaging Tutorial, [online], Lucas Nussbaum. Dostupné z URL:* <https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.en.pdf#page=7>, 2024.

[16] *Debian package structure, [online], Wiki FreePascal. Dostupné z URL:* <https://wiki.freepascal.org/Debian_package_structure>, 2022.

[17] *How to Create a Simple Debian Package, [online], Baeldung. Dostupné z URL:* <https://www.baeldung.com/linux/create-debian-package>, 2024.

[18] *deb (file format), [online], Wikipedia. Dostupné z URL:* <https://en.wikipedia.org/wiki/Deb_(file_format)>, 2024.

[19] *DB Browser for SQLite wiki, [online], Justin Clift. Dostupné z URL:* <https://github.com/sqlitebrowser/sqlitebrowser/wiki>, 2024.

[20] *Proč používat bootstrap, [online], HTML factory. Dostupné z URL:* <https://www.html-factory.cz/clanek/proc-pouzivat-bootstrap/>, 2019.

Seznam symbolů a zkratek

CA Certifikační autorita – Certificate authority

CSR Žádost o podpis certifikátu – Certificate signing request

CSS Kaskádové styly– Cascading Style Sheets

HTML Hypertextový Značkovací Jazyk – HyperText Markup Language

HTTPS Hypertext Transfer Protocol Secure

IKE Internetová Výměna Klíčů – Internet Key Exchange

MVC Model–view–controller

PGP Docela Dobré Soukromí – Pretty Good Privacy

PHP Osobní domovská stránka – Personal Home Page

S/MIME Zabezpečené/Víceúčelové Rozšíření Internetové Pošty – Secure/Multipurpose Internet Mail Extensions

SSL Vrstva bezpečných socketů – Secure Sockets Layer

TLS Transport Layer Security

URL Uniform Resource Locator

GPG/GnuPG GNU Privacy Guard