Slovenská technická univerzita v Bratislave

Fakulta elektrotechniky a informatiky

**Zadanie 4: Implementácia správy používateľských hesiel**

Úvod do počítačovej bezpečnosti

Vypracovali: Jakub Šimek

Nikola Zarembová

Eduardo Milan Martinez

Boris Hvozda

Patrícia Hulinová

# Úlohy

## Úloha1

Cieľom štvrtého zadania je oboznámiť sa s problematikou správy používateľských hesiel a implementovať aplikáciu, ktorou otestujete získané poznatky v praxi. K dispozícii sú zdrojové kódy kostry takejto aplikácie, do ktorej je potrebné doimplementovať potrebnú funcionalitu. Odporúča sa využívať štandardné API z Javy.

Úloha by mala nadväzovať na doterajší web projekt pre šifrovanie súborov. Aktuálne viete šifrovať súbory symetrickou šifrou a šifrovací kľúč uchovávať ako súčasť šifrovacieho súboru. Cieľom tohto zadania je implementovať prihlasovanie do vašej aplikácie pre viac užívateľov. Podstatnú časť riešenia bude tvoriť správa užívateľských hesiel a ich bezpečné uchovávanie. Zaroveň implementovaním prihlasenia úžívateľov môžete vytvoriť efektívnejšie riešenie pre správu verejných a súkromných kľúčov úžívateľov, ktoré sú potrebné pre šifrovanie súborov. Vysledná aplikácia by teda mala umožňovať bezpečné zdieľanie súborov medzi dvoma úžívateľmi. V tejto časti zadania je však dôraz položený na bezpečnú implementáciu správy hesiel.

## Úloha 2

Cieľom druhej časti štvrtého zadania je doprogramovať k prvej časti zadania funkcie na kontrolu zložitosti hesla (t.j. aby boli použité veľké a malé písmená, číslice, minimálna dĺžka hesla, či sa heslo nenachádza v slovníku). Pre kontrolu hesla voči slovníku, (teda či sa na heslo neda použiť jednoduchý slovníkový útok) sa odporúča využívať už existujúce knižnice napr. vt-password alebo Passay, popr. iné. Samozrejme môžete vytvoriť aj vlastné riešenie na kontrolu pomocou slovníka. Táto kontrola je pre druhú časť zadania povinná! Následne je potrebné spísať správu o riešení zadania (ku každej podúlohe krátku analýzu problému, zvolený postup riešenia a dosiahnutý výsledok).

# Čo sme použili

## Jazyky

* Back-end: Java framework Spring Boot
* Front-end: Angular 8 framework

## Knižnica

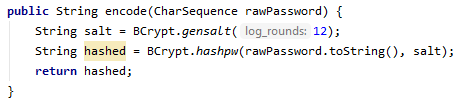
* org.springframework.security
* org.passay

## Podúlohy

### Generovanie saltu a hashov

Na generáciu saltu sme použili BcryptPasswordEncoder z knižnice springframework.security.crypto. Pomocou metódy *gensalt()* vygenerujeme randomný salt. Číselný argument nám špecifikuje počet opakovaní hashovania hesla. Dá sa špecifikovať v hodnotách od 4 do 32. Defaultne je nastavená hodnota 10.

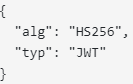
Metóda [*hashpw*](https://docs.spring.io/spring-security/site/docs/current/api/org/springframework/security/crypto/bcrypt/BCrypt.html#hashpw-java.lang.String-java.lang.String-)*(java.lang.String password, java.lang.String salt)* slúži na zahashovanie hesla. Takéto heslo sa následné ukladá do databázy.



### Autentifikácia

Na prevedenie autentifikácie sme zvolili JWT – Json Web Tokens. JWT je štandard (RFC 7519) pre bezpečný prenos informácií. Naša aplikácie používa API endpointy a tento typ autentifikácie je výhodný z viacerých dôvodov. Token nie je nutné uchovávať v databáze. Taktiež sa nemusíme zaoberať žiadnymi *‘sessions’*.

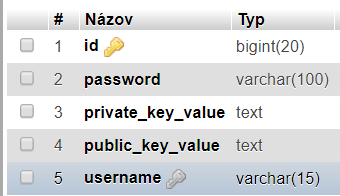
Token sa skladá z troch častí:

* Header – pozostáva z dvoch častí: typ tokenu, hashovací algoritmus
* Payload – obsahuje požiadavky: predmet, autorita, čas expirácie tokenu..
* Signature – zaisťuje, že token nebude zmenený počas prenosu

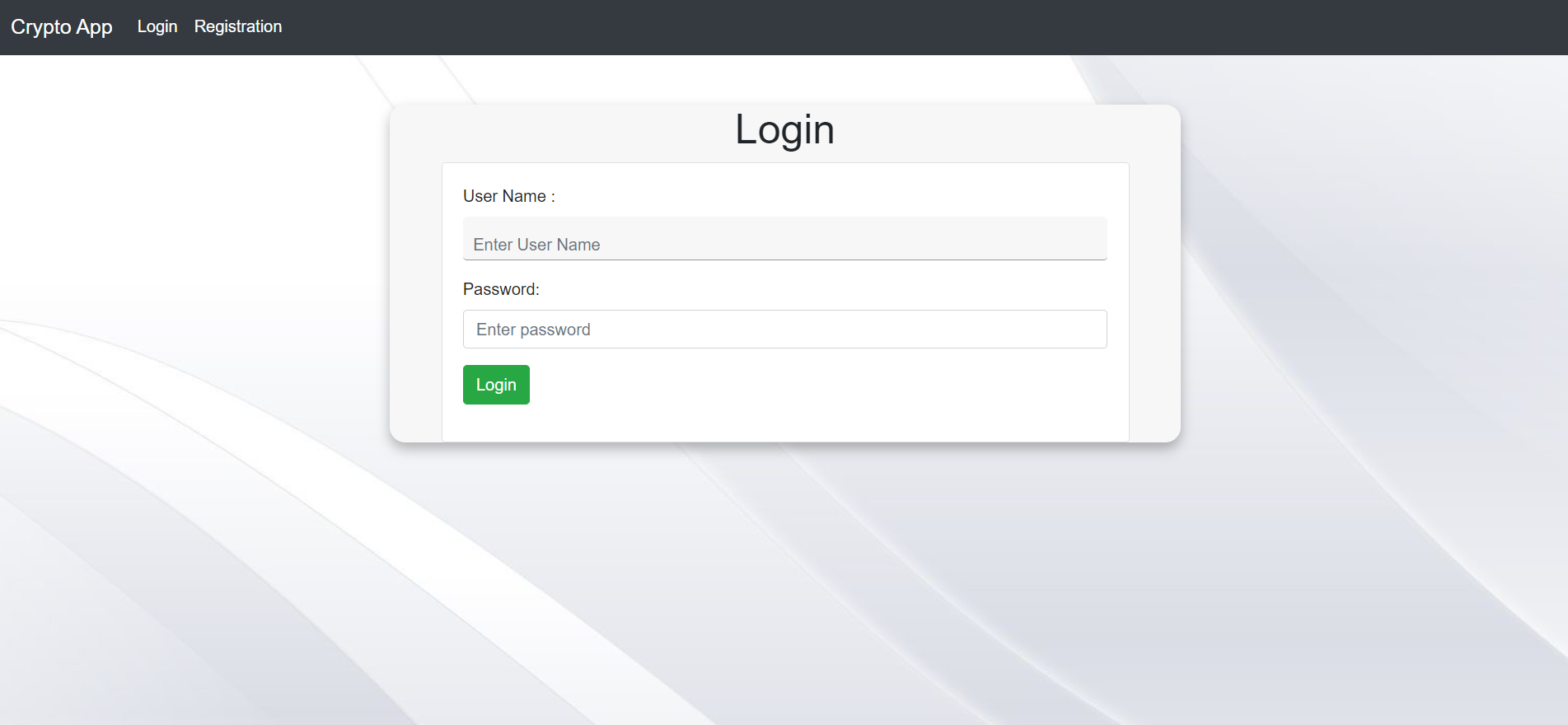
Token sa po prihlásení používateľa pošle ako odpoveď. A pri komunikácii opačným smerom sa server pozrie do autorizačného headera, či token existuje. Ak token existuje a je platný, používatelia majú prístup k zabezpečeným zdrojom/cestám. Obsah headera by mal vyzerať nasledovne. Platnosť tokenu je stanovená na 10minút. Po uplynutí tejto doby už nemá user prístup k endpointom.

### Ako sa aplikácia používa

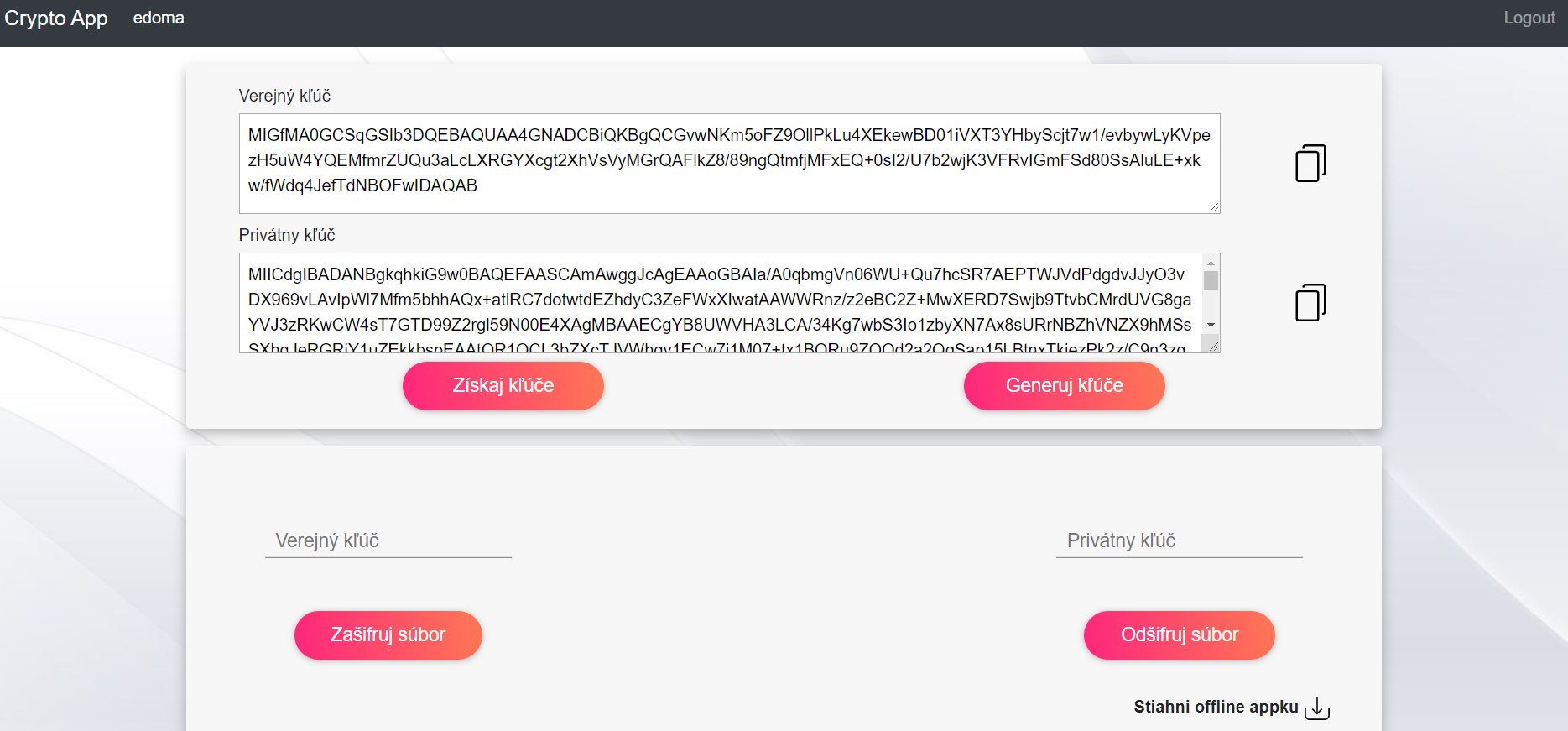
Na začiatku sa na našej webovej aplikácii možnosť prihlásenia pre používateľa. Ak používateľ ešte nemá účet, môže sa pomocou menu prepnúť do podstránky pre registráciu. Po prihlásení sa mu zobrazí hlavná stránka šifrovacej aplikácie. Po pridaní autentifikácie sa teraz do databázy ukladá pár kľúčov (public/private) priradený pre každého používateľa. Používateľ môže teda okrem generovania nových hesiel aj získať už existujúce heslá z databázy.



Obrázok - Tabuľka používateľov



Obrázok - Login => TODO



Obrázok - Hlavná stránka

Spodná lišta slúži na šifrovanie súborov, ktoré bolo implementované v predchádzajúcom zadaní.