Úvod do počítačovej bezpečnosti

Zadanie 4 - Implementácia správy používateľských hesiel

Cieľom zadanie bolo oboznámiť sa s problematikou správy používateľských hesiel a implementovať aplikáciu, ktorou si získané poznatky otestujeme v praxi.

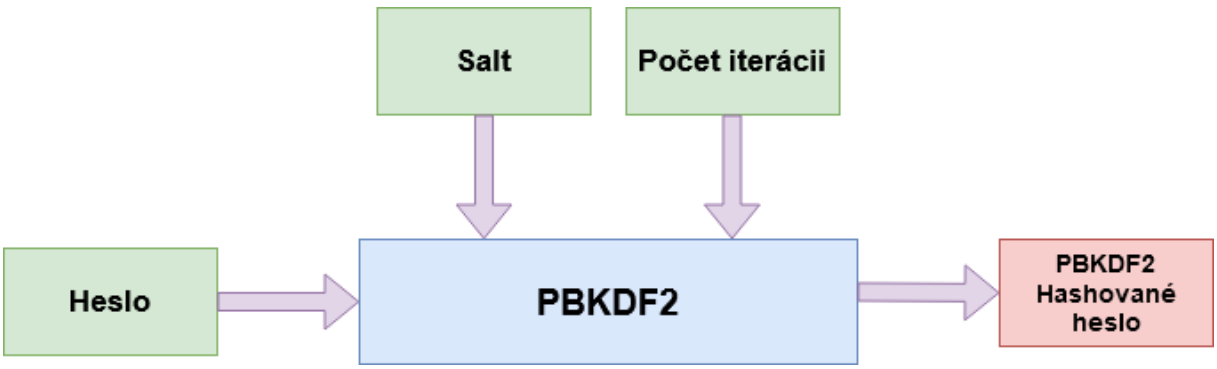
A. Špecifikácia

- Programovací jazyk: Java
- Knižnice: `java.security` (security framework)
  `javax.crypto` (kryptografické operácie)

B. Hashovací algoritmus

Bezpečne hashované heslo je šifrovaná sekvencia znakov získaná po použití určitého hashovacieho algoritmu na heslo od používateľa. Na to aby bolo možné ochrániť heslá samotný hash nemusí stačiť a preto sa pri generovaní hashu používa salt. Salt je náhodný reťazec špecifikovanej dĺžky, ktorý sa používa ako ďalší vstup do hashovacej funkcie a pripája sa k heslu pred jeho zaheshovaním. Po aplikácii saltu majú rovnaké hesla vždy rozdielne hashe čo značne stážuje útoky.

Ako algoritmus v mojej aplikácii som použil PBKDF2 (Password-Based Key Derivation Function 2). PBKDF2 je kryptografická funkcia, ktorá generuje hash požadovanej dĺžky z hesla užívateľa, pričom aplikuje salt. Cieľom PBKDF2 algoritmu je dosiahnuť aby bola hashovacia funkcia dostatočne pomalá na to aby zabránila útokom, ale stále dostatočne rýchla na to, aby používateľovi nespôsobovala znateľné oneskorenie. Algoritmus aplikuje pseudo náhodnú funkciu opakovane na základe špecifikovaného počtu iteracií, ktorý určuje ako pomalá hashovacia funkcia bude.



C. Spustenie aplikácie

Aby bolo možné aplikáciu spustiť je potrebné mať na zariadení nainštalovanú JAVU. Aplikáciu je možné spusti dvojklikom na .jar súbor.

D.Používateľská príručka

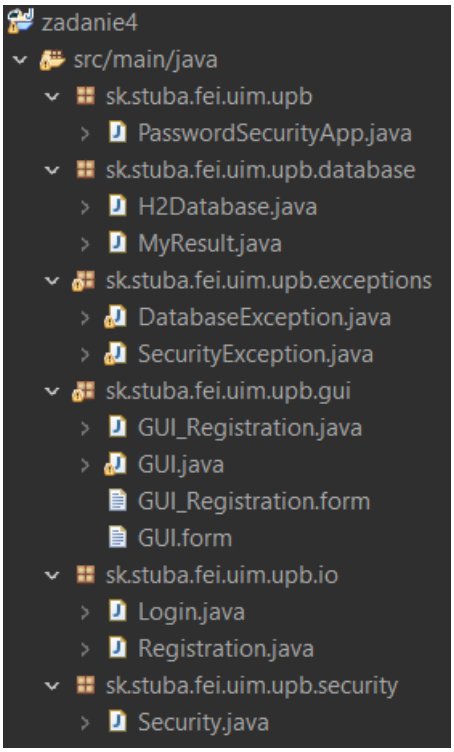
1. Návod na prepojenie databázy s aplikáciou

Aplikácia využíva embedded H2 databázu. Aplikácia hľadá predvolenú databázu v home adresári, preto je potrebné do neho (C:\Users\<meno>\database) nakopírovať priložený priečinok **`database`**. Tento priečinok obsahuje databázu **`myDB.mv.db`**, do ktorej sa ukladajú používateľské prihlasovacie udajú.

2. Návod na importovanie projektu do IDE

Aplikácia bola vyvíjaná v Eclipse IDE pričom bola vytvorená ako Gradle projekt, pre prípad, že by ste chceli otvárať kód aplikácie v IDE a nejako ho meniť, odporúčam aplikáciu importovať ako „existing gradle project“ aby náhodou nevznikali nejaké komplikácie.
V prípade, že by ste si chceli nanovo vygenerovať .jar stačí použiť gradle task „`jar`“, ktorý celý kód skompiluje vybuilduje a zabalí do `zadanie4.jar`, takto vygenerovaný súbor je potom možné nájsť v `zadanie4\build\libs` adresári a spustiť ho.

3. Štruktúra projektu



4. Stručný popis hlavných tried:

Trieda	Funkcionalita
Registration	Riadenie registrácie nového používateľa
Login	Riadenie prihlasovania používateľov
Security	Generovanie a kontrola hashovaných hesiel
H2Database	Spojenie a vytváranie dotazov na DB
GUI	Grafické používateľské rozhranie aplikácie
GUI_Registration	Grafické používateľské rozhranie registrácie

Registrácia nového užívateľa (Registration)

- Trieda vykonáva kontrolu duplicitnej registrácie používateľov s rovnakým používateľským meno a stará sa o registráciu nových používateľov

Prihlásenie nového užívateľa (Login)

- Trieda riadi prihlasovanie už existujúci požívateľov a vykonáva kontrolu korektného zadania používateľského mena a hesla.

Generovanie hashovaných hesiel (Security)

- Trieda riadi generovanie saltu, hashovanych hesiel a tiež spätnú kontrolu hesiel pri prihlasovaní
- Na hashovanie je použitý **PBKDF2** algoritmus so saltom veľkosti 128 bitov
- Na to aby bol salt naozaj náhodný bola využitá trieda `java.security.SecureRandom`, ktorá poskytuje kryptograficky silný generátor náhodných čísel **(úloha 1)**
- **PBKDF2** algoritmus obsahuje predvolenú funkcionalitu na oneskorenie **(úloha 3)**
- Algoritmus aplikuje pseudo náhodnú funkciu opakovane na základe špecifikovaného počtu iteracií. Tato funkcionalita zaručuje spomalenie hashovacej funkcie a tiež stážuje útočníkom uhádnutie hesla, pretože na to aby ho vedeli odhaliť je potrebné uhádnuť aj samotný počet iteracií, ktorý bol použitý pri jeho generovaní
- Metóda `PBEKeySpec(char[] password, byte[] salt, int iterationCount, int keyLength)` vytvorí kľuč, ktorý je následne použitý na vygenerovanie hashovaného hesla pomocou `SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1")` **(úloha 2)**

Aplikačná databáza (H2Database)

- Aplikácia využíva offline embedded H2 databázu uloženú v súbore **`myDB.db`**
- Trieda riadi vytváranie spojenia s databázou ako aj jeho zatváranie
- V triede sa tiež nachádzajú metódy na vyberanie dát a ich pridávanie do databázy