SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

PassLock

**PROJEKT IZ KOLEGIJA „Sigurnost informacijskih sustava“**

Varaždin, 2018.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

**Fabijan Josip Kraljić**

**Jakov Kristović**

**GitHub repozitorij:** [**github.com/jkristovi/PassLock**](https://github.com/jkristovi/PassLock)

PassLock

**PROJEKT IZ KOLEGIJA „SIgurnost informacijskog sustava“**

Mentor:

Doc dr. sc. Petra Grd

Varaždin, prosinac 2018.

Sadržaj

[1. Ideja projekta 1](#_Toc531790614)

[2. Slične aplikacije 2](#_Toc531790615)

[3. Korištene tehnologije 3](#_Toc531790616)

[3.1 Visual Studio 2017 (C#, .NET Framework) 3](#_Toc531790617)

[3.2 SQLite 4](#_Toc531790618)

[3.3 SHA256 hash algoritam 4](#_Toc531790619)

# Ideja projekta

Ideja za ovu aplikaciju je došla iz toga da na kolegiju Sigurnost informacijskih sustava se mora napraviti nekakav praktičan projekt. Sam kolegij je jako zanimljiv, te se moglo jako puno toga naučiti o (ne)sigurnosti stvari i sustava koje svakodnevno koristimo i kojima smo okruženi. Štoviše, naučili smo i da koliko god se zaštitimo, nikad nismo u potpunosti sigurni, te da je najviše što možemo učiniti sa naše strane to da smanjimo sve rizike koje možemo primijetiti na minimum, koristeći neke od mjera opreza.

Nadahnuti tom idejom i misli, odlučili smo se napraviti jednostavnu aplikaciju koja bi mogla svakome od nas povećati mjere opreza, odnosno smanjiti konkretne rizike na jednu nižu razinu – odlučili smo se napraviti **PassLock** aplikaciju.

Svrha ove aplikacije je povećanje sigurnosti na internetu, odnosno korištenje lozinki koje je teško probiti korištenjem nekih tipičnih 'Dictionary Attack' ili sličnih brute force metoda.

Aplikacija omogućuje korisniku izradu popuno prenosive (eng. portable, transferable) SQLite baze podataka koja bi bila zaključana željenom šifrom od minimalno 8 znakova.

Nadalje, korisnik može unositi željene lozinke koje će koristiti pri prijavi u društvene mreže i ostale servise koje svakodnevno koristi, pritom birajući željeni broj znakova pri generiranju lozinke koja će biti pohranjena u SHA256 formatu.

Korisniku je također omogućena i daljnja izmjena generirane lozinke, gdje može mijenjati i dodavati posebne znakove, velika slova (koja su uvjet kod izrade lozinki na pojedinim online servisima), ponovno je kriptirati, brisati i tako dalje. Također, ukoliko to želi, korisnik može i promijeniti lozinku za otvaranje same baze.

Time smo zapravo naše lozinke koje svakodnevno koristimo osigurali ne samo od tipičnih Dictionary Attack metoda, već i organizirali na jedno mjesto što nam daje mogućnost da ih ne moramo pamtiti konstanto, već samo tu jednu koja bi ih sve štitila.

# Slične aplikacije

# Korištene tehnologije

U sljedećem poglavlju ćemo ukratko opisati tehnologije koje smo koristili prilikom izrade PassLock aplikacije, kao i reći nešto više o samom algoritmu SHA256, koji je korišten pri kriptiranju lozinki.

## 3.1 Visual Studio 2017 (C#, .NET Framework)

![Slika na kojoj se prikazuje isječak crteža

Opis je automatski generiran]() Integrirano razvojno okruženje Visual Studio, kreativna je platforma koja se može koristiti za uređivanje, ispravljanje i izgradnju koda, a zatim i objavljivanje aplikacije, izrađena od strane Microsofta. Visual Studio je bogat program koji se može koristiti za mnoge aspekte razvoja softvera.

Slika 1. Microsoft Visual Studio

Osim standardnog urednika i debuggera koji većina IDE-ova pruža, Visual Studio uključuje kompajlere, alate za dovršavanje koda, grafičke dizajnere i još mnogo toga za olakšavanje procesa razvoja softvera. Neke od važnih značajki koje nudi su:

* **Refactoring** – *uključuje operacije kao što je inteligentno preimenovanje varijabli, vađenje jedne ili više linija koda u novu metodu, promjene redoslijeda parametara metode i još mnogo toga.*
* **IntelliSense** – *pojam za skup značajki koje prikazuju informacije o vašem kodu izavno u uređivaču i, u nekim slučajevima, za vas piše i male dijelove koda.*
* **Quick Actions** – *značajka koja podcrtava potencijalne probleme koje možete riješiti odmah, bez čekanja da se pogreška otkrije tijekom builda ili dok program već radi.*
* **Call Hierarchy** – *značajka koja prikazuje metode koje pozivaju odabranu metodu, može biti jako korisno kada razmišljate o izmjeni ili uklanjanju metode, ili kada pokušavate pronaći potencijalnu grešku.*
* **Code Lens** – *značajka koja vam pomaže da pronađete reference na svoj kod, promjene u kodu, povezane greške, radnje stavke, recenziju koda i testove jedinica, sve bez napuštanja urednika.*
* **Peek Definition** – *značajka koja vam prikazuje definiciju metode ili tipa podataka, bez potrebe da otvarate zasebne datoteke.*

## 3.2 SQLite

## 3.3 SHA256 hash algoritam