LockMaster – Maksim Momcilovic und Damian Sommer

Inhaltsverzeichnis

[Informieren 2](#_Toc131401002)

[Arbeitsform 2](#_Toc131401003)

[Beschreibung 2](#_Toc131401004)

[Ziele 2](#_Toc131401005)

[Anforderungsanalyse 3](#_Toc131401006)

[Planen 4](#_Toc131401007)

[Testfälle 4](#_Toc131401008)

[Zeitplan 5](#_Toc131401009)

[MockUPs 6](#_Toc131401010)

[Entscheiden 6](#_Toc131401011)

[Legende 6](#_Toc131401012)

[Präferenzmatrix 6](#_Toc131401013)

[Tabelle Technologien Auswahl 6](#_Toc131401014)

[Entscheidung 6](#_Toc131401015)

[Realisieren 7](#_Toc131401016)

[Probleme 7](#_Toc131401017)

[Kontrollieren / Testen 7](#_Toc131401018)

[Testfall-Protokoll 9](#_Toc131401019)

[Testumgebung 10](#_Toc131401020)

[Testfazit 10](#_Toc131401021)

[Auswerten 10](#_Toc131401022)

# Informieren

## Arbeitsform

* Gruppenarbeit
* Projektmitarbeiter:
  + Damian Sommer 🡪 Docs, Testing
  + Maksim Momcilovic 🡪 Developement, Testing
* Datenversionisierung-Software:
  + GitHub
* Technologien:
  + Tauri TS
  + React TS
  + Vite
  + Rust
* Projektmanagement-Methode: IPERKA

## Beschreibung

Der Passwort Generator soll Passwörter nach eigenen Vorgaben erstellen. Das bedeutet, man hat eine Auswahl von verschiedenen Templates, beispielsweise soll ein Sonderzeichen, mehrere Zahlen, kleine und grosse Buchstaben vorhanden sein.

Die Passwörter sollen anschliessend verschlüsselt in einem JSON-File zum zugehörigen Dienst zugeordnet werden.

Dieses Projekt hat die Eingrenzung, dass das gesamte Programm mit allen Passwörtern auf allen Computer Systemen funktionieren sollte.

Ziele

* Ein Programm schreiben, welches Passwörter nach eigenen Anforderungen generiert.
* Die generierten Passwörter sollen in einem JSON-File verschlüsselt abgespeichert werden.
* Die Passwörter sollen im JSON-File verschlüsselt angezeigt werden, wenn man es ohne das Masterpasswort öffnen will.
* Eine Desktop App für alle OS.

## Arbeitsjournal

Siehe «Arbeitsjournal» Excel.

## Anforderungsanalyse

|  |  |  |
| --- | --- | --- |
| **Nummer** | **Funktional? Qualität? Rand?** | **Beschreibung** |
| 1 | Funktional | Das Programm soll Passwörter generieren können. |
| 2 | Funktional | Der Benutzer soll das Passwort mit Grossbuchstaben konfigurieren können. |
| 3 | Funktional | Der Benutzer soll das Passwort mit Kleinbuchstaben konfigurieren können. |
| 4 | Funktional | Der Benutzer soll das Passwort mit Nummern konfigurieren können. |
| 5 | Funktional | Der Benutzer soll das Passwort mit Sonderzeichen konfigurieren können. |
| 6 | Funktional | Der Benutzer soll die Länge des Passworts zwischen 3 und 20 Zeichen frei wählen können. |
| 7 | Funktional | Alle systemrelevanten Daten werden in einem JSON-File gespeichert. |
| 8 | Funktional | Die Passwörter sollen verschlüsselt im JSON-File abgespeichert werden müssen. |
| 9 | Funktional | Der Benutzer soll so lange ein Passwort generieren können, bis es ihm gefällt. |
| 10 | Funktional | Das JSON-File soll mit einem Masterpasswort geschützt sein. |
| 11 | Funktional | Der Benutzer soll alle Einträge im JSON-File nach Eingabe seines Masterpassworts unverschlüsselt einsehen können. |
| 12 | Funktional | Der Benutzer soll die abgespeicherten Passwörter nicht mehr abändern können. |
| 13 | Funktional | Der Benutzer soll einzelne Einträge im JSON-File löschen können. |
| 14 | Funktional | Der Benutzer soll nach Einträgen im JSON-File suchen können. |
| 15 | Qualität | Das Programm soll nur auf Dark-Mode ausgerichtet sein. |
| 16 | Rand | Die Applikation soll auf Windows funktionieren. |
| 17 | Rand | Die Applikation soll auf MacOS funktionieren. |
| 18 | Rand | Die Applikation soll auf Linux funktionieren. |
| 19 | Rand | Die Applikation soll Tauri TS verwenden. |
| 20 | Rand | Die Applikation soll Vite verwenden. |
| 21 | Rand | Die Applikation soll React TS verwenden. |
| 22 | Rand | Die Applikation soll Rust verwenden. |
| 23 | Rand | Das Projekt soll bis am 08.05.2023 fertig sein. |
| 24 | Funktional | Copy-To-Clipboard für Passwörter |

# Planen

## Testfälle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test-Nr. | Anf-Nr. | Voraussetzung | Eingabe | Erwartete Ausgabe |
| 1 | 1, 2 | Das Programm ist gestartet.  Man befindet sich auf der Seite «Create Password» | Man drückt auf die Checkbox, «big letters».  Man drückt auf den Button «generate password» | Ein Passwort mit nur grossen Buchstaben mit der Standard-Länge von 3 wird erstellt |
| 2 | 1, 3 | Das Programm ist gestartet.  Man befindet sich auf der Seite «Create Password» | Man drückt auf die Checkbox, «small letters».  Man drückt auf den Button «generate password» | Ein Passwort mit nur kleinen Buchstaben mit der Standard-Länge von 3 wird erstellt |
| 3 | 1, 4 | Das Programm ist gestartet.  Man befindet sich auf der Seite «Create Password» | Man drückt auf die Checkbox, «numbers».  Man drückt auf den Button «generate password» | Ein Passwort mit nur Nummern mit der Standard-Länge von 3 wird erstellt |
| 4 | 1, 5 | Das Programm ist gestartet.  Man befindet sich auf der Seite «Create Password» | Man drückt auf die Checkbox, «special characters».  Man drückt auf den Button «generate password» | Ein Passwort mit nur Speziellen Zeichen mit der Standard-Länge von 3 wird erstellt |
| 5 | 1, 6 | Das Programm ist gestartet.  Man befindet sich auf der Seite «Create Password» | Man drückt auf die Checkbox, «big letters».  Man drückt auf den Button «generate password»  Man gibt eine 20 an der Stelle der 3 ein. | Ein Passwort mit nur grossen Buchstaben mit der Länge von 20 wird erstellt |
| 6 | 10 | Die App ist geöffnet | Klick auf «Passwords» | Alle Passwörter werden mit dem Login unverschlüsselt angezeigt. |
| 7 | 7 | Irgendein Computer. Es wurden schon Passwörter erstellt und abgespeichert. | Man öffnet den Pfad für Appdaten.  Beispiel Windows; /AppData/Local/LockMaster | Man sieht ein JSON-File im Fileexplorer. |
| 8 | 14 | Es wurde schon ein Password mit dem «Test» Login erstellt. Man befindet sich auf der Seite «Show Passwords». Cursor und Eingabe in der Suchleiste positioniert, im Dropdown ist «Login» selektiert. | Eingabe des Wort «test» in | Es kommt nur der Eintrag zum Wort «test» zum vorschein. |
| 9 | 13 | Es wurde schon ein Password mit dem «Test» Login erstellt. Man befindet sich auf der Seite «Show Passwords». | Klick auf das Rote «X» Icon | Password «test» wird gelöscht |
| 10 | 24 | Es wurde schon ein Password mit dem «Test» Login erstellt. Man befindet sich auf der Seite «Show Passwords». | Klick auf das blaue «Clipboard-Copy» Icon  «CTRL + V» Tastenkombination | Passwort von «test» wird in einem beliebigen Inputfeld das Text einnimmt eingefügt. |

## Zeitplan

Siehe File: «Zeitplan.xsxl»

## MockUPs

<https://www.figma.com/file/xosMA1IbXJByGbAX6AOqRt/LockMaster?node-id=0%3A1&t=lXW45ve16lKplog0-1>

# Entscheiden

|  |  |
| --- | --- |
| Legende |  |
| Performance | 1 |
| Komplexität | 2 |
| Sicherheit | 3 |
| Startgeschwindigkeit | 4 |
| Grösse | 5 |

## Präferenzmatrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | Anzahl Vergleiche |
| 1 |  | 1 | 1 | 4 | 1 | 4 |
| 2 |  |  | 3 | 4 | 3 | 3 |
| 3 |  |  |  | 3 | 5 | 2 |
| 4 |  |  |  |  | 4 | 1 |
| 5 |  |  |  |  |  | 0 |
| Nennungen | 3 | 0 | 3 | 3 | 1 | 10 |
| Rang | 1 | 3 | 1 | 1 | 2 |  |
| Prozent | 30 | 0 | 30 | 30 | 10 | 100 |

## Tabelle Technologien Auswahl

Auwahlsmöglichkeiten:

* Tauri
* Electron

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Tauri | | Electron | |
| Kriterien | Gewichtung | TN | GTN | TN | GTN |
| Performance | 30 | 2 | 60 | 1 | 30 |
| Komplexität | 0 | 1 | 0 | 2 | 0 |
| Sicherheit | 30 | 2 | 60 | 1 | 30 |
| Startgeschwindigkeit | 30 | 2 | 60 | 1 | 30 |
| Grösse | 10 | 2 | 20 | 1 | 10 |
| Gesamtnutzen | 100% | 9 | 200 | 6 | 100 |

## Entscheidung

Anhand unserer Entscheidungsmatrix haben wir und dazu entschieden, dass wir für unsere Applikation Tauri und nicht Electron verwenden werden da es für unseren Use-Case Optimal geeignet ist.

# Realisieren

Hier möchte ich ein paar Codesnippets von Problemen zeigen, die nicht auf Anhieb lösbar waren und ein wenig komplexer waren. Dazu beschreibe ich noch wo es was im Code macht.

# Kontrollieren / Testen

## Testfall-Protokoll

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test-Nr** | **Testfall-Nr** | **Datum** | **Bemerkung** | **Resultat** | **Durchgeführt** |
| 1 | 1 | 07.04.2023 | Funktioniert mit allen Kombinationen (Länge + Präferenzen) | OK | Maksim Momcilovic |
| 2 | 2 | 07.04.2023 | Funktioniert mit allen Kombinationen (Länge + Präferenzen) | OK | Maksim Momcilovic |
| 3 | 3 | 07.04.2023 | Funktioniert mit allen Kombinationen (Länge + Präferenzen) | OK | Maksim Momcilovic |
| 4 | 4 | 07.04.2023 | Funktioniert mit allen Kombinationen (Länge + Präferenzen) | OK | Maksim Momcilovic |
| 5 | 5 | 07.04.2023 | Funktioniert mit allen Kombinationen (Länge + Präferenzen) | OK | Maksim Momcilovic |
| 6 | 6 | 07.04.2023 | / | OK | Maksim Momcilovic |
| 7 | 7 | 07.04.2023 | / | OK | Maksim Momcilovic |
| 8 | 8 | 07.04.2023 | / | OK | Maksim Momcilovic |
| 9 | 9 | 07.04.2023 | / | OK | Maksim Momcilovic |
| 10 | 10 | 07.04.2023 | / | OK | Maksim Momcilovic |

## Testumgebung

**Device**: HP Z Book 15 G3, 512Gb SSD, 16GB Ram, I7 6700HK 2.6 ghz  
**OS**: Windows 10 Pro  
**Tested Software**: LockMaster

## Testfazit

100% aller Tests aus dem Protokoll haben funktioniert. Die App ist somit einwandfrei und bereit zum Deployment.

# Auswerten

## Reflexion über die Arbeit

Im Verlaufe des Projektes gab es nahezu keine Probleme, bis auf den Fakt das wir unsere Files nicht dynamisch verschlüsseln können, was somit die Anforderung das es auf allen OS laufen kann bereitstellen würde. Hätten wir das implementieren wollen, müssten wir erstmal unsere eigene Rust-Library dafür schreiben müssen und das hätte schlicht und einfach den zeitlichen Rahmen des Projektes gesprengt. Das nächste mal wollen wir definitiv die Anforderungen detaillierter Besprechen und abklären ob wir wirklich auch alles bieten können. Das haben wir uns in diesem Fall nicht detailliert genug angeschaut und das wollen wir in Zukunft anders machen.

Wir konnten eine neue Technologie kennenlernen und viele neue Dinge lernen. Zudem haben wir eine interessante Mischung aus TypeScript (React) und Rust zusammengecodet die einen tatsächlichen Nutzen hat und funktioniert. Die App funktioniert einwandfrei. Wir konnten uns gut organisieren und die Arbeit gerecht aufteilen mit Tools wie Git auf GitHub und teilweise MS Teams Calls.

## Schlusswort

Dieses Projekt war für uns ein Riesenspaß und konnte gut mit moderaten und spannenden Hindernissen gelöst werden. Im verlaufe dieses Projekts, haben wir herausgefunden was das Framework «Tauri» eigentlich zu bieten hat. Mit Rust als Sprache der Zukunft von komplexen Programmierproblemen und mit nahe zu unendlichen Möglichkeiten und Performance, ist es für uns eine Ehre bereits jetzt in der Blütezeit dieser Sprache mit ihr zu arbeiten und erste Erfahrungen zu sammeln. Damit wissen wir auch das wir in die richtige Richtung ansteuern was unsere Zukunft anbelangt.