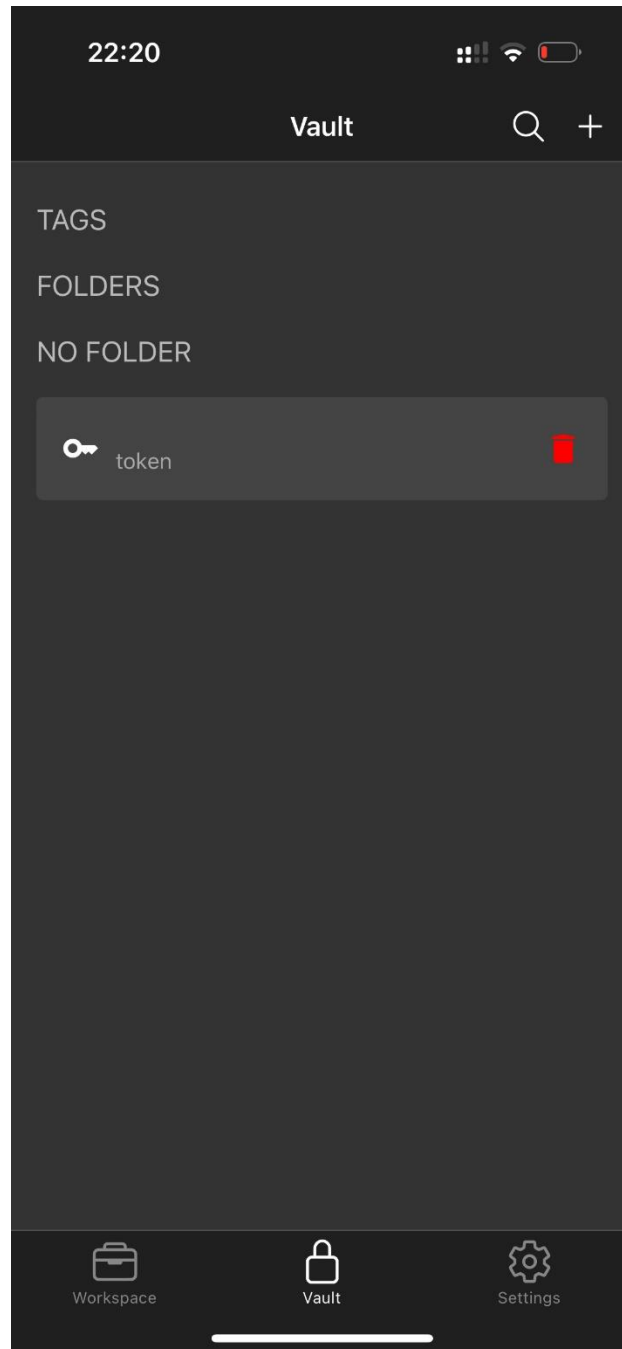


## MEINE APP

### Tokenward

Version 1.0.0, Datum: 6. Juni 2024



# ABSTRACT (KURZBESCHREIBUNG)

Tokenward ist eine Passwort-Management Applikation, die spezifisch dafür designed ist, um auch Access Tokens wie z.B. (Git, Figma, Github) etc. sicher zu speichern und zu verwalten.

In der heutigen digitalen Welt sind Access Tokens von entscheidender Bedeutung, da sie eine sichere Authentifizierung und Autorisierung in zahlreichen Anwendungen ermöglichen. Allerdings stellt die Verwaltung dieser Tokens eine Herausforderung dar, insbesondere für Entwickler und Teams, die täglich mit mehreren Tokens arbeiten müssen.

Tokenward hat das Ziel, eine benutzerfreundliche und sichere Anwendung zu entwickeln, die es den Nutzern ermöglicht, ihre Access Tokens wie auch ihre Passwörter in einer Applikation zentralisiert zu speichern, zu organisieren und zu verwalten. Die Applikation wird Funktionen wie sichere Verschlüsselung, einfache Integrationen mit beliebten Entwicklungsplattformen und eine intuitive Benutzeroberfläche bieten.

# USER STORIES

## Alle Anforderungen im ÜK:

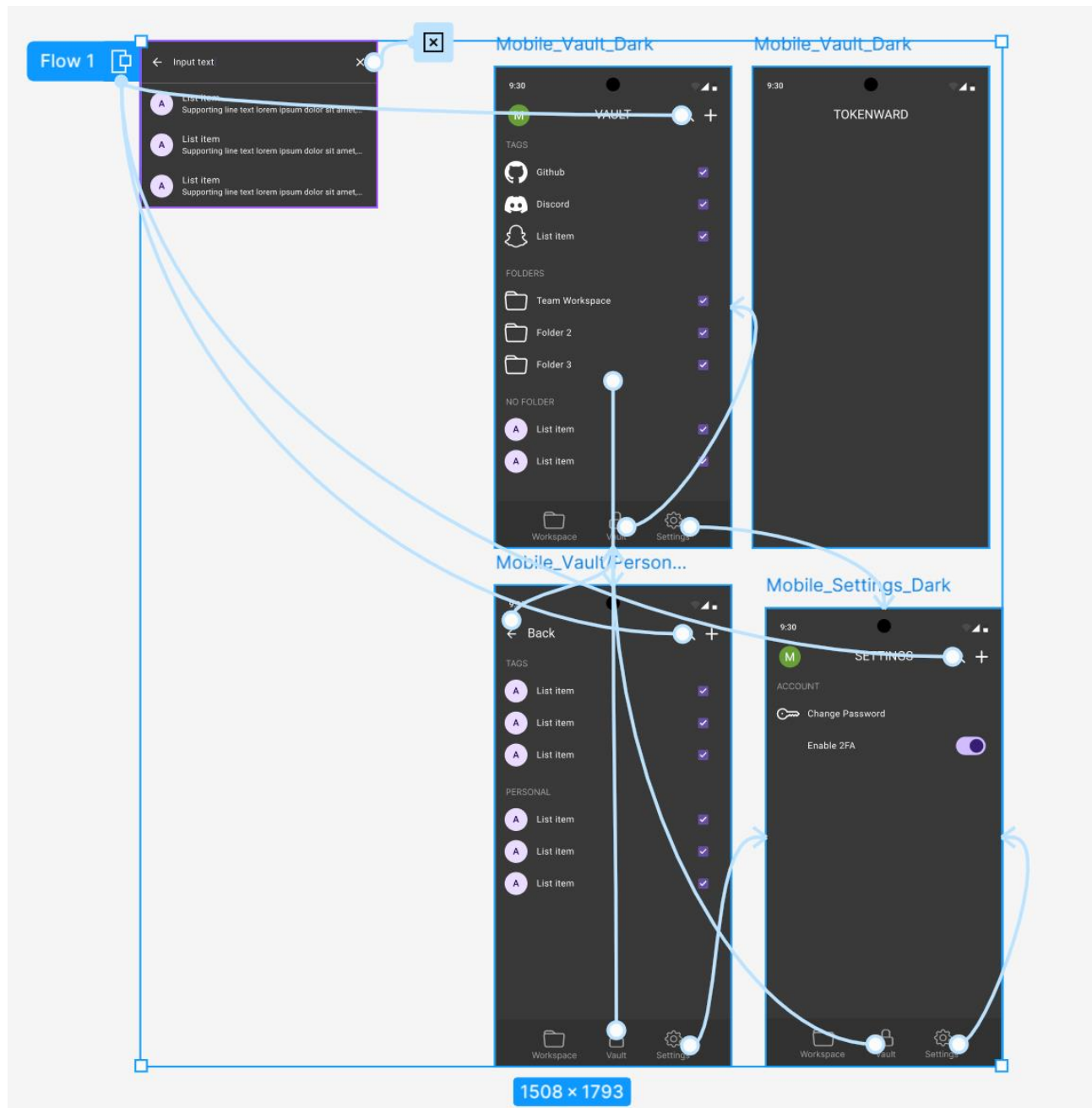
- Als Benutzer möchte ich meine Tokens sicher verschlüsselt abspeichern und von überall auf diese zugreifen können.
- Als Benutzer möchte ich eine schöne Schnittstelle haben, um Tokens zu verwalten.
- Als Benutzer möchte ich ein Konto erstellen können, indem ich meine E-Mail und ein sicheres Passwort angebe, um Zugang zur Anwendung zu erhalten.
- Als Benutzer möchte ich die Option haben, meine E-Mail-Adresse zu ändern, um sicherzustellen, dass mein Konto geschützt ist.

## Alle Anforderungen nach dem ÜK

- Als Benutzer möchte ich mein Konto über eine E-Mail-Verifizierungsmethode bestätigen, indem ich einen Verifizierungscode erhalte und diesen in die Anwendung eingeben kann.
- Als Benutzer möchte ich die Möglichkeit haben, mein Konto dauerhaft zu löschen, falls ich die Anwendung nicht mehr verwenden möchte.
- Als Benutzer möchte ich die Möglichkeit haben, mich bei Bedarf von allen Geräten abzumelden, um die Sicherheit meines Kontos zu gewährleisten.
- Als Benutzer möchte ich Benachrichtigungen erhalten, wenn es verdächtige Aktivitäten auf meinem Konto gibt, um potenzielle Sicherheitsbedrohungen zu erkennen.
- Als Benutzer möchte ich die Applikation auf dem Handy, PC und als Web-app verwenden können.
- Als Benutzer möchte ich «Team-Workspace» erstellen können, um Personen in ein «Team» einzuladen und mit denen Tokens zu teilen für Projekte etc.

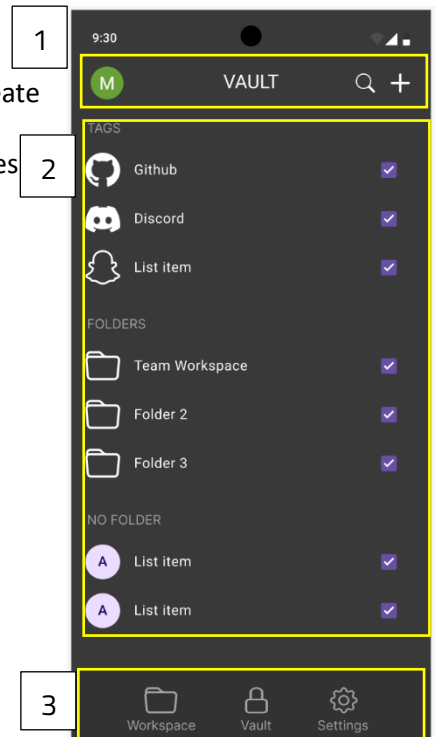
# MOCKUPS

An dieser Stelle den Prototypen deiner App beschreiben. Füge einen Printscreen deiner Screens ein und beschreibe jeden Screen einzeln.



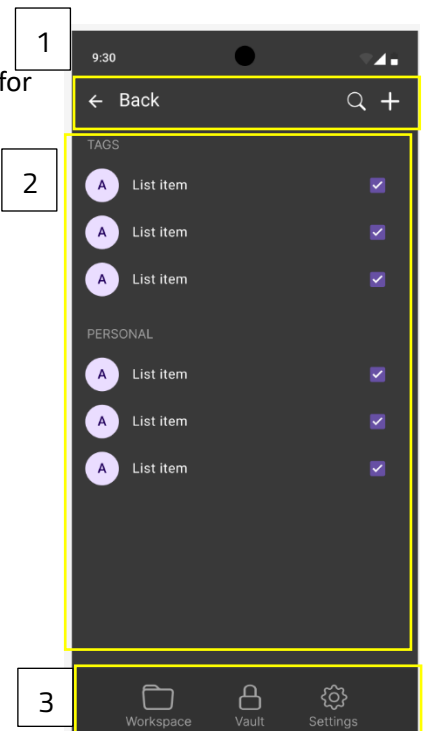
## Vault Page

- 1) The Top Navbar is used to either logout, Search for a specific Item or create a new Item(Password/Folder/Tag)
- 2) In this section you can view Items by specific tags, folder or view the ones with no folder.
- 3) Here you can navigate to either the workspace or the settings section.



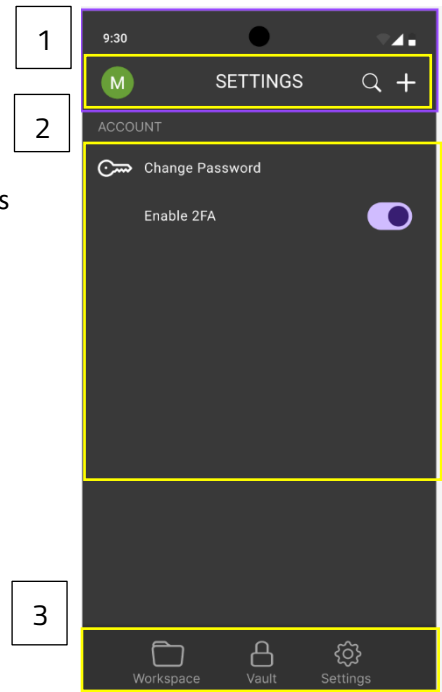
## Folder View

- 1) The Top Navbar is used to either go back to the main vault screen, Search for a specific Item or create a new Item(Password/Folder/Tag)
- 2) Here you can view the Items.
- 3) Here you can navigate to either the workspace or the settings section.



## Settings Page

- 1) The Top Navbar is used to either logout, Search for a specific Item or create a new Item(Password/Folder/Tag)
- 2) Here you can change all the settings.
- 3) Here you can navigate to either the workspace or the settings section.



# TECHNISCHE REALISIERUNG

Die mobile App besteht aus verschiedenen Komponenten, die zusammenarbeiten, um eine reibungslose Benutzererfahrung zu bieten. In diesem Abschnitt werden die technischen Details der App-Entwicklung beschrieben, einschließlich der Komponenten, der Datenverwaltung und -speicherung sowie der Datenquellen. Komplizierte Abläufe werden durch UML-Diagramme veranschaulicht.

## Komponenten der App

Die Hauptkomponenten der App sind in verschiedenen Verzeichnissen und Dateien organisiert:

1. **app/**: Enthält die Hauptlogik der App, einschließlich der Bildschirme und Navigation.
2. **components/**: Wiederverwendbare UI-Komponenten.
3. **config/**: Konfigurationsdateien für verschiedene Umgebungen.
4. **lib/**: Bibliotheken und Hilfsfunktionen.
5. **assets/**: Statische Ressourcen wie Bilder und Icons.
6. **CustomColors.js**: Definiert benutzerdefinierte Farben für die App.
7. **.env**: Umgebungsvariablen für die Konfiguration.
8. **app.config.js**: App-Konfigurationsdatei.
9. **babel.config.js**: Babel-Konfigurationsdatei für das Transpilieren von JavaScript.
10. **package.json**: Listet die Abhängigkeiten und Skripte der App auf.
11. **LICENSE**: Lizenzinformationen für die App.

## Datenverwaltung und -speicherung

Die App nutzt verschiedene Methoden zur Datenverwaltung und -speicherung:

- **Lokale Speicherung:** Die App speichert einige Daten lokal auf dem Gerät des Benutzers, um eine Offline-Funktionalität zu gewährleisten. Dies erfolgt über AsyncStorage oder ähnliche Technologien.
- **Externe APIs:** Die App ruft Daten von externen APIs ab, um aktuelle und dynamische Inhalte anzuzeigen. Diese Daten werden über RESTful API-Aufrufe abgerufen und verarbeitet.
- **Konfigurationsdateien:** Umgebungsabhängige Konfigurationen werden in .env-Dateien gespeichert und zur Laufzeit geladen.

## Datenquellen

Die Hauptdatenquellen der App sind:

- **Externe APIs:** Diese liefern dynamische Inhalte, die regelmäßig aktualisiert werden. Die API-Endpunkte und Schlüssel werden in der Konfigurationsdatei config/ definiert.
- **Lokale Daten:** Statische Inhalte und benutzerdefinierte Einstellungen werden lokal auf dem Gerät gespeichert.

# TESTING

## Test 1: Tokens sicher verschlüsselt speichern und abrufen

**ID:** ST-01

**Anforderungen:** Als Benutzer möchte ich meine Tokens sicher verschlüsselt abspeichern und von überall auf diese zugreifen können.

**Vorbedingungen:** Benutzer hat ein gültiges Konto und ist eingeloggt.

**Ablauf:**

1. Benutzer öffnet die App und navigiert zum Token-Speicherbereich.
2. Benutzer fügt einen neuen Token hinzu und speichert diesen.
3. Benutzer loggt sich aus und erneut ein.
4. Benutzer überprüft, ob der gespeicherte Token verfügbar und korrekt verschlüsselt ist.

**Erwartetes Resultat:** Der Token ist sicher gespeichert, verschlüsselt und kann nach dem erneuten Einloggen abgerufen werden.

## Test 2: Benutzerfreundliche Schnittstelle

**ID:** ST-02

**Anforderungen:** Als Benutzer möchte ich eine schöne Schnittstelle haben, um Tokens zu verwalten.

**Vorbedingungen:** Benutzer hat ein gültiges Konto und ist eingeloggt.

**Ablauf:**

1. Benutzer öffnet die App und navigiert zur Token-Verwaltungsoberfläche.
2. Benutzer prüft, ob alle Funktionen leicht zugänglich und verständlich sind.
3. Benutzer fügt, bearbeitet und löscht Tokens über die Schnittstelle.

**Erwartetes Resultat:** Die Benutzeroberfläche ist intuitiv und ermöglicht eine einfache Verwaltung der Tokens.

## Test 3: Konto erstellen

**ID:** ST-03

**Anforderungen:** Als Benutzer möchte ich ein Konto erstellen können, indem ich meine E-Mail und ein sicheres Passwort angebe, um Zugang zur Anwendung zu erhalten.

**Vorbedingungen:** Keine, Benutzer startet die App zum ersten Mal.

**Ablauf:**

1. Benutzer öffnet die App und navigiert zur Registrierungsseite.
2. Benutzer gibt seine E-Mail-Adresse und ein sicheres Passwort ein.
3. Benutzer bestätigt die Eingaben und erstellt das Konto.

**Erwartetes Resultat:** Das Konto wird erfolgreich erstellt und der Benutzer kann sich einloggen.



## Test 4: E-Mail-Adresse ändern

**ID:** ST-04

**Anforderungen:** Als Benutzer möchte ich die Option haben, meine E-Mail-Adresse zu ändern, um sicherzustellen, dass mein Konto geschützt ist.

**Vorbedingungen:** Benutzer hat ein gültiges Konto und ist eingeloggt.

**Ablauf:**

1. Benutzer navigiert zu den Kontoeinstellungen.
2. Benutzer wählt die Option, die E-Mail-Adresse zu ändern.
3. Benutzer gibt die neue E-Mail-Adresse ein und bestätigt die Änderung.
4. Benutzer loggt sich aus und versucht, sich mit der neuen E-Mail-Adresse einzuloggen.

**Erwartetes Resultat:** Die E-Mail-Adresse wird erfolgreich geändert und der Benutzer kann sich mit der neuen E-Mail-Adresse einloggen.

1. Benutzer bestätigt die Eingaben und erstellt das Konto.

**Erwartetes Resultat:** Das Konto wird erfolgreich erstellt und der Benutzer kann sich einloggen.

## Test 4: E-Mail-Adresse ändern

**ID:** ST-04

**Anforderungen:** Als Benutzer möchte ich die Option haben, meine E-Mail-Adresse zu ändern, um sicherzustellen, dass mein Konto geschützt ist.

**Vorbedingungen:** Benutzer hat ein gültiges Konto und ist eingeloggt.

**Ablauf:**

1. Benutzer navigiert zu den Kontoeinstellungen.
2. Benutzer wählt die Option, die E-Mail-Adresse zu ändern.
3. Benutzer gibt die neue E-Mail-Adresse ein und bestätigt die Änderung.
4. Benutzer loggt sich aus und versucht, sich mit der neuen E-Mail-Adresse einzuloggen.

**Erwartetes Resultat:** Die E-Mail-Adresse wird erfolgreich geändert und der Benutzer kann sich mit der neuen E-Mail-Adresse einloggen.

# TESTPROTOKOLL

In diesem Kapitel wird die Testausführung protokolliert. Welche Tests wurden wann von wem mit welchem Resultat durchgeführt.

Hier ein Beispiel eines Eintrages:

ID	Person / Datum	Erfolgreich	Bemerkungen
ST-01	D. Muster 18.06.2023	Ja	Der Testfall war erfolgreich, der Testperson 1 ist jedoch aufgefallen, dass es in der angezeigten Fehlermeldung noch einen Rechtschreibfehler gibt.
ST-01	Mitja Kurath 06.06.2024	Ja	Der Testfall war erfolgreich, keine Probleme festgestellt.
ST-02	Leon Höfferer 06.06.2024	Ja	Die Benutzeroberfläche ist intuitiv und alle Funktionen sind leicht zugänglich
ST-03	Leon Höfferer 06.06.2024	Ja	Konto wurde erfolgreich erstellt, keine Probleme festgestellt.
ST-04	Mitja Kurath 06.06.2024	Nein	E-Mail-Adresse wurde nicht erfolgreich geändert wegen fehlender Verifikation Möglichkeit der E-Mail.

# FAZIT

## Was lief gut/schlecht?

**Gut:** Die Entwicklung verlief insgesamt reibungslos, die meisten Features konnten ohne grössere Probleme implementiert werden.

**Schlecht:** Einige Integrationstests verliefen nicht wie geplant und erforderten zusätzliche Debugging-Zeit. Darüber hinaus hatten wir Probleme mit der Implementierung von Firebase, was dazu führte, dass nicht alle gewünschten Funktionen für die Applikation fertiggestellt werden konnten.

## Wie seid ihr mit dem Endergebnis zufrieden?

Insgesamt sind wir sehr zufrieden mit dem Endergebnis.

## Was habt ihr gelernt?

Wir konnten unser Wissen über React-Native erweitern und lernten auch viel über Firebase/Firestore

War alles vorhanden oder was fehlte noch?

Ein paar Funktionen wie erweiterte Benachrichtigungen und detaillierte Logs fehlen noch und sollen in zukünftigen Versionen implementiert werden.